

Generation II RCT User Manual

Introduction

The GenII RCT is an improved version of Sonicor's original RCT system; a two axis material transport system designed to carry product through a multi-station cleaning process with minimal (if any) operator intervention.

RCT systems have the potential to greatly improve process uniformity and operator safety while reducing labor when used with larger, multi-station cleaning systems.

Controls are accessed via a color touch-screen display.

GenII systems are fully user programmable.

The GenII is capable of being programmed by a software novice. A more sophisticated user will find additional features that make more powerful programs possible.

Programs running in the GenII can not only control the RCT's motion but can also interact with external devices (sensors, switches, pumps, etc.) to supervise an entire automated process.

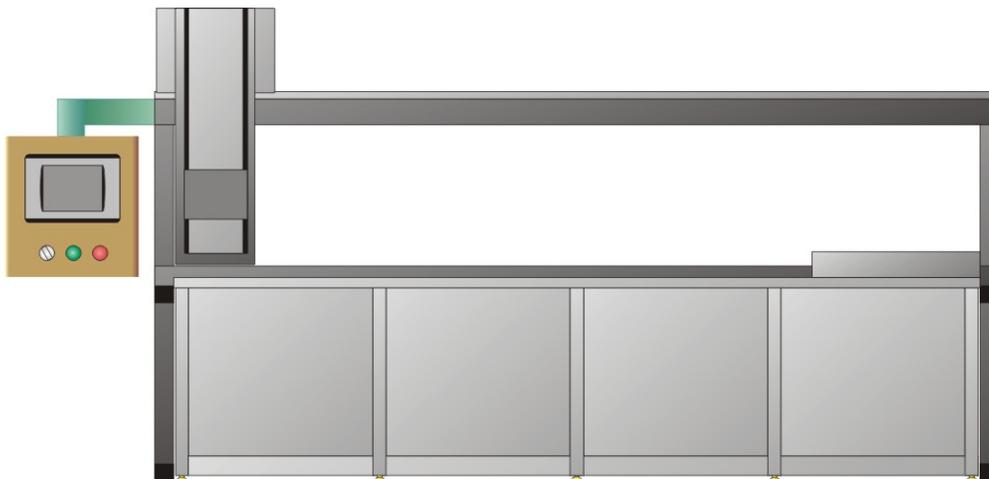
Almost without exception, programs written for use on an older GenI RCT will run unchanged on a GenII RCT.

The programming environment in the GenII has been improved to simplify writing and debugging programs. Additional instructions have been added to the GenI instruction set to make programming easier. A suite of new debugging tools are included in the GenII to simplify program development.

What It Is

A RTC system consists of an elevated horizontal track that is positioned above and behind a multi-station process.

On the track rides a trolley with an arm that extends over the process stations.



Hanging from the end of the arm are a set of hooks that are designed to be able to engage handles on baskets. Baskets carry product to be processed in the system.

The trolley is able to run back and forth along the track carrying baskets of product to (program) specified locations.

The trolley's arm is able to move up and down to move baskets in and out of the different stations used by the process.

The GenII's control system runs a program that defines how, where, and when the trolley moves.

The control system can also interact with other parts of the system to monitor sensors and turn ON/OFF pumps, ultrasonics, etc.

How Does It Work?

There is a motor to drive the trolley on each of the two (vertical and horizontal) axes.

The RCT determines the current position of the trolley by using two rotary optical encoders – one for the vertical position and the other for the horizontal position. The encoders produce an electrical pulse each time the trolley moves a fixed distance, along with information about which direction the trolley is moving. The RCT controller counts the pulses to keep track of relative motion of the trolley. Two “HOME” sensors are used to determine when the trolley is at the “start of travel” or “HOME” location. (HOME is located at the greatest height above the starting end of the system. The “starting end” is usually the left side but can be specified to be the right side at the time of purchase.)

Every time the RCT is turned ON its perception of “where it is” may not be accurate until it is allowed to move to its HOME position.

After being HOMEd an RCT may be commanded to move to user specified positions with excellent repeatability.

Locations in the system are specified in terms of “COUNTS”. Counts correspond to position information from the vertical and horizontal encoders. On most systems a “count” corresponds to approximately 3/8” (10mm).

Specifications:

Power Requirements:

- 105-125v A.C.
- 60hz
- 2 amps.
- Internal battery provides program retention for up to 5 years.

Environmental:

- Temperature, 50-110f
- RH, 2-95% non-condensing

Program Storage:

- Up to 4 user programs in internal storage.
- Up to 400 steps per user program.
- Optional removable storage is available.

I/O:

- Inputs: 12 optically isolated inputs for 12-24v D.C.
- Outputs: 8 hard contact, N.O. relay outputs. Up to 1 amp per output. Up to 240v output. (Note that external “snubbing” may be required for reactive loads.) (Note: Common is shared with the 24v input supply.)
- Additional, optional I/O options can be provided.

Mechanical:

- 150 lbs / 70kg maximum load.
- Repeatability approximately 0.375” (10mm).
- Track length and height are optional at the time of purchase.

The Programming Language

Overview

The GenII was designed to be programmed by a user without much (if any) programming experience because most facilities that have use for an RCT type device do not keep programmers on staff.

Someone with the ability to think logically and explain a process as a series of sequential steps can usually write a program to make the GenII do what they need it to do.

Someone with a bit of programming background will find additional features that make more powerful programs possible.

Commands like “GO_X (100)” and “GO_Y (75)” are almost self-explanatory and are the basis of the RCT programming language.

Program steps are performed by the GenII in the order that they appear in the program EXCEPT when flow control commands (GO_SUB, RETURN, GOTO, IF_IN_ON, and IF_IN_OFF) are used to change the flow of the program (the order of program instruction execution). Flow control allows for conditional (do this *only* if that...) and structured programming.

Only one instruction is run at a time.

Debugging tools are provided to make writing and testing user programs easier. These tools include:

- Single-step execution of programs. When the RCT is set into this mode the program stops after each instruction is executed and waits for the user to allow the next instruction to be run.
- Running programs with “no motion”. i.e. skip all GO_X, GO_Y, GO_LOCN, AGITATE, and HOME instructions but run all other instructions.
- View inputs and program FLAGS during program operation.
- Defeat motor “stall” detection.

Program editing tools have been improved and now include:

- Insert new instruction after the current instruction.
- Delete the current instruction.
- Insert a new instruction in place of the current instruction.
- Show a particular program step (by entering it's line number).
- Increment/Decrement the current instruction step to show the next or previous step.
- Save current program to a specified memory location.
- Recall program from a specified memory location (non-destructively).
- Erase current program.

Note:

- With the new system there are four memory locations in which to store programs – and an additional, separate location for the CURRENT working program.
- Only the CURRENT program may be RUN.
- Only the CURRENT program may be EDITED.
- The user may RECALL a stored program from one of the four memories into the CURRENT program.
- The user may STORE the CURRENT program in any of the four memory locations.
- A stored program may be recalled from a memory location, modified, and saved in a different memory location without changing the original stored program. This ability makes it easier to create variations on a standard program.
- GenI systems managed their stored programs differently. The GenII gives the user more flexibility in writing and editing programs.

Instruction Set

A device's instruction set is the collection of commands (instructions) that the device will respond to in a program.

The descriptions below show the RCT instructions with their arguments. Arguments are the parameters that must be entered with an instruction so that the instruction can be performed.

In this document arguments are shown in parentheses for clarity. Ie. "Instruction (ARGUMENT)".

In the instruction GO_X (1234) the (1234) is the argument – it tells the GO_X command that the programmer wants the RCT to travel to the location 1234 counts from HOME on the horizontal axis (X axis).

Note:

When a programmer enters a new instruction on the ENTER PROGRAM screen by pressing the button for the instruction, the display will change to show:

- A short description of what the instruction does.
- What parameters (arguments) are required by the instruction. (If any.)
- How the required parameters are used by the instruction. (Briefly.)

The entry of out-of-range arguments will not be permitted.

Instructions

NO-OP

A NO-OP instruction does NOTHING. It is simply used as a place holder. NO-OP is short for NO OPeration.

In the RCT GenII a user cannot insert a NO-OP instruction into a program.

When the current program is CLEARED the first (and only) instruction in the program will automatically be a NO-OP that is placed there by the system. The NO_OP will be overwritten by the user as a new program is entered.

Note:

If a user should want to use null instruction in a program, an unused LABEL instruction will serve the same purpose.

HOME

A HOME instruction returns the trolley to its HOME location.

The HOME location is always “fully UP” and “fully over” to the starting location.

Note:

Most systems have the home location at the LEFT side of the machine but some special systems are available with the home location at the RIGHT side of the machine. LEFT or RIGHT home location may be specified at the time of purchase.

HOME is always executed by:

- First moving the trolley UP to the TOP-most position. (First.)
- Then moving OVER to the home location. (Second.)

When performing the HOME command the RCT system does not use its position encoders to determine its location. It will move the trolley towards the HOME sensors until the sensor positions are reached. (Always moving Vertical first and then Horizontal).

When a HOME command is completed, the vertical and horizontal encoder COUNTS are automatically reset to ZERO (0). This procedure “calibrates” the position of the trolley.

Note:

Most systems have the home location at the LEFT side of the machine but some special systems are available with the home location at the RIGHT side of the machine. LEFT or RIGHT home location may be specified at the time of purchase.

Note:

The system will only operate one axis (motor) at a time.

Note:

During a HOME operation the trolley may appear to “bounce” off of the HOME location. This is intentional. The reversed motion increases accuracy by reducing “overshoot”.

Note:

If the user enters a GO_X (0) or a GO_Y (0) the RCT controller will instead execute a HOME_X or HOME_Y internal instruction to re-calibrate the RCT position.

GO_X (XXXX)

(XXXX = 0..9999)

The GO_X (XXXX) specifies that the trolley should move horizontally (on the X axis) to the absolute location XXXX counts away from the HOME location.

The MAXIMUM value for XXXX is 9999.

The MINIMUM value for XXXX is 0.

The higher the XXXX number, the further away from the HOME location the trolley will travel.

Each encoder count represents approximately 0.375" or 10mm of travel for the trolley. For example, specifying a move of GO_X (16) will move the trolley to a location about 6" (0.375" x 16) away from the horizontal home.

The trolley will move to the specified location by moving forward or backward as required in order to reach the position called for in the command. GO_X (100) is NOT saying to move the trolley 100 counts from where it is, it is saying to move the trolley to the position 100 counts away from the *home* location regardless of its *current* location.

For example:

- If the trolley is at location X = 500 and is commanded to GO_X (1000) it will move away from HOME.
- If the trolley is at location X = 1600 and is commanded to GO_X (1000) it will move towards HOME.
- In both cases, the final position after the move will be 1000 counts from HOME.

Note:

Systems are setup to have a minimum distance that they will move. Once a motor is turned ON there is a minimum distance it can move before it can stop. If a command requires the RCT to move less than this minimum distance it will ignore the move.

In most systems the minimum move is set to "3". In most systems the both the X and Y axes have the same minimum move.

For example. If a RCT trolley is at location X= 114 and it is commanded to GO_X(115) the trolley will not move because if it were to move it would surely overshoot the target.

In order to determine the exact count at a particular location to use for a GO_X command, a user can use the MANUAL JOG controls that are on the MANUAL SCREEN to position the trolley where it will need to be and then note down the indicated counts at that position.

The MANUAL SCREEN also contains a manual version of the GO_X (XXXX) program command so that the user may move directly to a know location or test a measured location before using it in a

program.

Note:

It is possible to use the GO_X command to tell the trolley to *try* to move THROUGH part of the system (a tank wall for example). Of course the trolley cannot pass through solid metal and therefore there will be a crash. In the event of a crash the RCT will detect that it has become unable to move and will shut down the motor, cancel program operation, and indicate a STALL error.

It is the responsibility of the operator/programmer to be sure that the RCT is not commanded to crash. Crashing may result in damage to the system, the product being processed in the system, and/or injury to operators even though the automatic system will shut itself down and there are mechanisms in the system that are designed to limit the force that the trolley can exert.

Once the system has detected a STALL error it will not move until the operator manually clears the STALL.

Immediately after a STALL has occurred the system's knowledge of its location may be inaccurate. A manual HOME command should be used to re-calibrate the system.

Note:

If the user enters a GO_X (0) or a GO_Y (0) the RCT controller will execute a HOME_X or HOME_Y internal instruction to re-calibrate the RCT position.

GO_Y (YYYY)

(YYYY = 0..9999)

The GO_Y (YYYY) specifies that the trolley should move vertically (on the Y axis) to the absolute position of YYYY counts.

The MAXIMUM value for YYYY is 9999.

The MINIMUM value for YYYY is 0.

The greater the YYYY number the further *down* from the HOME (top) location the trolley will travel.

Each encoder count represents approximately 0.375" or 10mm. For example, specifying a move of GO_Y (16) will move the trolley to a location about 6" (0.375" x 16) down from the vertical home. The trolley will move to the specified location by moving forward or backward as required in order to reach the position called for in the command.

For example:

- If the trolley is at location Y = 200 and is commanded to GO_X (500) it will move down to location Y = 500.
- If the trolley is at location Y = 600 and is commanded to GO_X (500) it will move UP to location Y = 500.
- In both cases, the final position after the move will be 500 counts from HOME.

Note:

Systems are setup to have a minimum distance that they will move.

Once a motor is turned ON there is a minimum distance it can move. If the RCT is asked to move less than this distance it will ignore the move.

In most systems the minimum move is set to "3". In most systems the both the X and Y axes have the same minimum move.

For example. If a RCT trolley is at location Y= 94 and it is commanded to GO_Y(95) the trolley will not move because if it were to move it would surely overshoot the target.

In order to determine the exact count at a particular location on the system so as to be able to use it for a GO_Y command a user can use the MANUAL JOG controls that are on the MANUAL SCREEN to move the trolley to where it will need to be and then note down the indicated counts at that position.

The MANUAL SCREEN also contains a manual GO_Y (YYYY) command so that the user may move directly to a known location or test a measured location before using it in a program.

Note:

It is possible to use the GO_Y command to tell the trolley to *try* to move THROUGH part of the system (a tank bottom, for example). Of course the trolley cannot pass through solid metal and therefore there will be a crash. In the event of a crash the RCT will detect that it has become unable to move and will shut down the motor, cancel program operation, and indicate a STALL error.

It is the responsibility of the operator/programmer to be sure that the RCT is not commanded to crash. Crashing may result in damage to the system, the product being processed in the system, and/or injury to operators even though the automatic system will shut itself down and there are mechanisms in the system that are designed to limit the force that the trolley can exert.

Once the system has detected a STALL error it will not move until the operator manually clears the STALL.

Immediately after a STALL has occurred the system's knowledge of its location may be inaccurate. A manual HOME command should be used to re-calibrate the system.

Note:

If the user enters a GO_X (0) or a GO_Y (0) the RCT controller will execute a HOME_X or HOME_Y internal instruction to re-calibrate the RCT position.

Label (ZZ)

(ZZ = 0..99)

Labels are used to mark locations in a program. Labels are used by flow control instructions as a destination – a place to GOTO.

Flow control commands that use LABELs include: GO_SUB, GOTO, IF_IN_ON, and IF_IN_OFF.

The RCT supports LABELS from 00 to 99 (a total of 100).

When the program reaches a LABEL (ZZ) instruction while running a program it does nothing. And then it moves on to the next program step.

If the programmer wants to "jump" to a new location – that is, move to an out-of-sequence location in a program, a GOTO (ZZ) command is used.

When the program encounters a GOTO (ZZ) command it stops and searches the program for a “LABEL (ZZ)” instruction. When it finds the LABEL it begins executing commands starting at the LABEL (ZZ) location.

If a program tries to find a LABEL (ZZ) and there is no such label in the program, program execution stops and an error is shown.

A programmer cannot enter two identical LABELs in the same program.

Why use LABELs when programs have line numbers associated with each instruction?
Program steps are numbered, always starting at 1.

As each new program step is entered by a programmer the system automatically assigns the next available line number to the new step.

When a program step is *inserted* into the middle of a program the line number of every step after the new step is incremented by 1. (All following instructions are “bumped” up by one to “make room” for the new instruction.)

This means that the line number of a particular location in a program can not be counted on to remain the same after the program is edited.

LABEL commands always remain unchanged, even when instructions are inserted or deleted. Using LABELs insures program references will continue to work correctly after edits.

IF_IN_OFF (II, LL)

(II = 0..00, LL = 0..99)

IF_IN_OFF is one of the two conditional branch commands that the RCT uses. (The other conditional branch command is IF_IN_ON).

By “conditional” we mean that the branch (GOTO) only occurs if a criteria is met. In this case, if a particular IN is OFF.

When the RCT encounters an IF_IN_OFF (II, LL) command, it:

- Tests input #II (allowed to be 01..99) to see if it is OFF.
- If it finds that INPUT #II is OFF the program branches (jumps; equivalent to a GOTO) to the LABEL (LL) and continues execution at the LABEL.
- If it finds that INPUT #II is ON the instruction does nothing and the program moves to the next instruction in the program.
- If the program does not contain a LABEL (LL) it encounters an ERROR and program execution is terminated.

A standard system has 12 inputs (1..12). The IF_IN_OFF allows the program to test to see if an input is OFF.

Note:

An input is viewed as ON if 12-24 volts D.C. is applied to it. An input is OFF if 0v D.C. is applied to it.

Voltages between 0 and 11v will produce uncertain results and should be avoided.

“INPUTs” numbered above 12, in most RCT systems, are not actually INPUTs; they are FLAGS. FLAGS are memory locations that can be SET to ON or CLEARED to OFF by program instructions (or manual intervention). FLAGS are viewed by the IF_IN_OFF and IF_IN_ON instructions as though they are actual INPUTs.

Note:

Conditional branch statements can also test the state of FLAGS. An RCT FLAG is a memory location that stores either an OFF or an ON.

Both FLAGS and OUTPUTs can be turned ON or OFF by the same statements. A FLAG may be turned ON by a PER_ON (PP) instruction. A FLAG may be turned OFF by a PER_OFF (PP) instruction. “PP” stipulates which FLAG (or OUTPUT) to turn ON or OFF. (“PER” refers to peripheral.)

FLAGS and OUTPUTs can also be turned ON and OFF by the TIMED_PER (PP) command.

IF_IN_ON (II,LL)

(II = 0..00, LL = 0..99)

IF_IN_ON is one of the two conditional branch commands that the RCT uses. (The other conditional branch command is IF_IN_OFF).

By “conditional” we mean that the branch (GOTO) only occurs if a criteria is met. In this case, if a particular IN is ON.

When the RCT encounters an IF_IN_ON (II,LL) command, it:

- Tests input II(allowed to be 01..99) to see if it is ON.
- If it finds that INPUT #II is ON the program branches to the LABEL (LL) and continues program execution at the LABEL.
- If it finds that INPUT #II is OFF the instruction does nothing and the program moves on to the next instruction in the program.
- If the program does not contain a LABEL (LL) it encounters an ERROR and program execution is terminated.

A standard system has 12 inputs (1..12). The IF_IN_ON allows the program to test to see if an input is ON.

Note:

An input is viewed as ON if 12-24 volts D.C. is applied to it. An input is OFF if 0v D.C. is applied to it.

Values between 0 and 11v will produce uncertain results and should be avoided.

“INPUTs” numbered above 12, in most RCT systems, are not actually INPUTs they are FLAGS. FLAGS are memory locations that can be SET to ON or CLEARED to OFF by program instructions (or

manual intervention). FLAGS are viewed by the IF_IN_OFF and IF_IN_ON instructions as though they are the same as actual INPUTS.

Note:

Conditional branch statements can also test the value of FLAGS. An RCT FLAG is a memory location that stores either an OFF or an ON.

Both FLAGS and OUTPUTs can may be turned ON or OFF by the same statements. A FLAG may be turned ON by a PER_ON (PP) instruction. A FLAG may be turned OFF by a PER_OFF (PP) instruction. "PP" stipulates which FLAG to turn ON or OFF.

Flags can also be turned ON and OFF by the TIMED_PER (PP) command.

GOSUB (LL)

(LL = 0..99)

GOSUB is a call to a subroutine that starts with LABEL (LL) and ends with a RETURN command.

GOSUB (LL) causes the program to perform the following actions:

- Stop executing instructions at the current program location.
- Store the current program location (the line number of the GOSUB instruction) in its internal memory.
- Search the program for LABEL (LL)
- If LABEL (LL) is found program execution begins at the LABEL. If LABEL (LL) is not found an error is flagged and program execution is terminated.
- The program continues to execute in the normal sequence, starting at the new location, until a RETURN instruction is reached.
- When a RETURN instruction is reached, the RCT will "recall" the location of the calling GOSUB instruction and will begin executing instructions at the step immediately after the calling GOSUB instruction.

Subroutines may be "nested" up to 30 levels deep. That means that there are sufficient system resources to permit a subroutine to call a subroutine, and that subroutine may call another subroutine... a total of 30 times. On the 31st nested subroutine call the RCT will indicate an error and the program will be terminated.

Care must be taken to see that a RETURN instruction is not encountered *before* a GOSUB. A RETURN will attempt to transfer program control to the location following the calling GOSUB – except if there wasn't a calling GOSUB there isn't any stored location to return to. So the RCT indicates an error and terminates the program.

Recursion (a subroutine calling itself) is permitted but not recommended.

Subroutines can simplify programs and add "structure".

Subroutines should be located in the program where the ONLY way to reach the subroutine is by way of the GOSUB command. This is often accomplished by placing all subroutines AFTER an END statement.

Program example:

If a program performs the same function in several location, say, for example, it opens a sliding door before dropping a basket in a station, the programmer has two options. He may insert the same block of instructions for opening the door at each location in the program where the door needs to be opened. Or he may write one block of code for opening the door as a subroutine and call that subroutine (via a GOSUB command) from every location in the program that needs to open the sliding door. The RETURN instruction that terminates the subroutine (MUST TERMINATE EVERY SUBROUTINE) brings the program right back to the place that called the subroutine once the subroutine has completed... and the program proceeds.

There are several advantages to the using a GOSUB in the above example.

- The length of the program is reduced because there needs only be one copy of the code that is used to open the sliding door.
- Should there ever be a need to make a change to how the door is opened there is only one place in the program that will have to be edited.
- Programs become more readable and hence more maintainable.

RETURN

Return from subroutine. RETURN must always be the last instruction in a subroutine.

The RETURN command is ONLY used in conjunction with the GOSUB command.

The GOSUB command tells the program to transfer execution to the beginning of a subroutine; from there normal, sequential program operation continues in the subroutine until a RETURN instruction is encountered.

When the program encounters a RETURN instruction it:

- Looks up the location of the GOSUB command that called the subroutine.
- Begins executing instructions starting with the instruction immediately following the calling GOSUB command.

Care must be taken to see that a RETURN instruction is not encountered BEFORE a GOSUB. In this (improper) case the RETURN will attempt to transfer control to the stored location (that following the calling GOSUB) – except there wasn't any GOSUB so there isn't any stored location. In this case the RCT will indicate an error and terminate the program.

PER_ON (PP)

(PP = 0..99)

The PER_ON instruction is used to turn ON the peripheral output specified by the argument (PP).

The standard GenII RCT has 12 (1..12) outputs (N.O. relay contacts). All outputs are set to OFF when

the system is first turned-on and also when a program starts to run. Outputs are also temporarily turned OFF when the PAUSE button is used.

The PER_ON command will turn ON (close the relay contacts) on the specified output.

If the argument (PP) is above the maximum number for physical outputs (12) the PER_ON command will turn ON a system FLAG #PP.

Note:

Flags are memory locations that may be turned ON and OFF to allow the program to keep track of a process status.

For example, a program may turn a flag ON immediately after placing a basket in a tank and it may turn the same flag OFF immediately after removing a basket from a tank. Another part of the program can then test the flag to determine if there is a basket in the tank.

Program examples:

PER_ON may be used to turn ON the ultrasonic generator in a cleaning tank.

PER_ON may be used to turn ON a FLAG so the program can “remember” that there is now a basket located in TANK#1.

PER_OFF (PP)

(PP = 0..99)

The PER_OFF instruction is used to turn OFF the peripheral output specified by the argument (PP).

The standard GenII RCT has 12 (1..12) outputs (N.O. relay contacts). All outputs are set to OFF when the system is first turned ON and when a program starts to run. Outputs are also temporarily turned OFF when the PAUSE button is used.

The PER_OFF command will turn OFF (open the relay contacts) on the specified output.

If the argument (PP) is above the maximum number for physical outputs (12) the PER_OFF command will turn OFF the (PP) system FLAG.

Note:

Flags are memory locations that may be turned ON and OFF to allow the program to keep track of a process status.

For example, a program may turn a flag ON immediately after placing a basket in a tank and it may turn the same flag OFF immediately after removing a basket from a tank. Another part of program can then test the flag to determine if there is a basket in the tank.

Program example:

PER_OFF may be used to turn OFF the ultrasonic generator in a cleaning tank.

PER_OFF may be used to turn OFF a FLAG so the program can “remember” that there is a basket located in TANK#1 and therefor TANK#1 is not available to be used by another basket.

TIMED_PER (PP,TT)

(PP = 0..32, TT = 0..99)

When a program encounters a TIMED_PER (PP,TT) instruction it turns on the specified output or flag (PP = 0..32) for (TT=0..99) time intervals.

The default time interval is 30 seconds – systems can be provided with different time intervals if needed. The maximum timing period for a standard system is 99 intervals of 30 seconds or 49.5 minutes.

The TIMED_PER instruction:

- Operates with all outputs (1..12) and flags up to #32. (PP = 1..32 inclusive.)
- Provides timing independent of other program operation. As long as the program is running, the TIMED_PERs will remain ON, regardless of what the rest of the program is doing, until the TIMED_PERs time-out when they are turned OFF.
- If the operator PAUSEs the program all running TIMED_PERs are temporarily PAUSED (timing is suspended) and the associated outputs (but not FLAGS) are turned OFF for the duration of the PAUSE. When the operator cancels the PAUSE, outputs are turned back ON and all TIMED_PER timers resume from where they were suspended.
- All TIMED_PER operation is canceled when a program ends. This includes timers that have not completed (have not timed-out).
- The TIMED_PER function takes no notice if an output or FLAG is changed (either manually or by another program command). It is NOT suggested that outputs or FLAGS controlled by a TIMED_PER be manipulated in any other way while the TIMED_PER function is running. The results are predictable, but likely to be undesirable.

Program examples:

The TIMED_PER may be used to turn on the ultrasonic cleaning in a tank (via an output) for a fixed time without preventing the RCT from moving baskets to and from other tanks. If the TIMER command were used, all RCT actions would wait until the TIMER completed (or was user canceled).

The TIMED_PER may be used to set a flag ON for a fixed interval to indicate how long a basket should be allowed to remain in a tank. The program may test that flag, at any convenient point in the process, to see if it is time to remove the basket from the tank.

TIMER (TTTT)

(0..9999)

The TIMER command stops any other program functions (except TIMED_PER operation) until (TTTT) seconds have elapsed or the “CANCEL TIMER” button on the RUN screen is pressed.

When a TIMER instruction is encountered in a running program a countdown timer will appear on the RUN screen showing the time remaining until the TIMER command will be completed. The program will not proceed to the next instruction until the TIMER interval of TTTT seconds has elapsed or the “CANCEL TIMER” button that appears on the screen is pressed.

The maximum time interval of 9999 seconds = 166.65 minutes = 2.75 hours.

The TIMER is used when the programmer wants to temporarily stop all other program activity for a

specific period of time.

For example, if a basket has just been lifted out of a cleaning tank and there is cleaning solution dripping off of the basket, a TIMER instruction may be used to allow a “drip-off” delay.

In this example it does not matter that nothing else will be done by the RCT while the TIMER is running. The RCT arm is occupied and it cannot do anything else.

Note:

Timers do NOT interfere with TIMED_PER operation.

WAIT

The WAIT command in a program causes the RCT to stop executing instructions and WAIT until the operator acknowledges by pressing the “CONTINUE” button. The “CONTINUE” button appears on RUN screen ONLY when a WAIT command is executed.

The purpose of the WAIT instruction is to allow the program to interact with its user.

When a program reaches a point that requires user interaction it can use a WAIT instruction to signal the user. The program will then stop and wait, suspending all operations except TIMED_PERs, until the user presses the CONTINUE button.

The blue CONTINUE button will only appear on the RUN screen when the WAIT instruction is being executed.

Note:

A program will wait *forever* until the CONTINUE button is pressed.

If an inattentive operator ignores a current WAIT, the program will patiently wait forever for a response.

Program Example

A user's process requires the machine operator to manually inspect the contents of a basket when a basket is lifted out of the cleaning tank.

The program will lift the basket out of the cleaning tank and then execute a WAIT instruction.

When the operator sees that the program is WAITing (displayed on the RUN screen) he can inspect the contents of the basket and then press the CONTINUE button.

Once the CONTINUE button has been pressed the program can resume the process.

GOTO (LL)

(LL = 0..99)

GOTO (LL) is an unconditional branch to LABEL (LL).

When the GOTO (LL) command is encountered in a program the controller will:

- Search the program to find LABEL (LL).
- If the LABEL is found the program will begin to execute instructions, starting at the LABEL.
- If the LABEL is not found the program will indicate an error and terminate the program.

Program example.

If a programmer wants to make an infinite loop:

```
instruction A
LABEL 01
instruction B
instruction C
instruction D
GOTO 01
instruction E
```

The result would be that the program would execute instructions A..D on the first time through. Then the program would loop back to LABEL 01 and forever would repeat instructions B,C, and D. Instruction E would never be performed. Instruction A would only be performed on the first iteration.

The only way to exit the loop (as written above) is to press the “CANCEL PROGRAM” button on the RUN SCREEN.

GO_LOCN (A,ZZ)

(A = 0..1), ZZ = 0..99)

The GO_LOCN (A,ZZ) command instructs the trolley to go to LOCATION (ZZ). LOCATION (ZZ) consists of a predefined pair of X and Y coordinates. The coordinates are defined by the programmer. The “A” parameter specifies whether to move to the X (0)coordinate first or the coordinate Y (1) first.

The GO_LOCN (A,ZZ) command requires that the memory LOCATION ZZ have a pair of (X and Y) coordinates stored in it. The default locations are 0,0 (HOME) until a programmer enters his own preferences. Locations revert to 0,0 when the LOCATION array is CLEARED by the programmer.

The GO_LOCN (A,ZZ) command will is equivalent to either:

- GO_X followed by a
- GO_Y (A=0)

or

- GO_Y followed by a
- GO_X (A=1)

...depending on the programmer specified value for “A”.

Note:

The “A” parameter in the GO_LOCN (A, ZZ) command specifies whether to move to the Y coordinate first and then the X coordinate (A=0) or the X coordinate first and then the Y coordinate (A= 1).

Which “A” is appropriate is usually determined by the path the trolley must take to avoid hitting the cleaning

system. (I.e. not running into the side of a tank or other structure.)

The selection of moving on the X axis first or the Y axis first is made by the programmer at the time the GO_LOCN (ZZ) instruction is entered.

- If A= 0: GO_X first, then GO_Y
- If A= 1: GO_Y first, then GO_X

ZZ refers to the memory location that contains the XXXX and YYYY coordinates for the moves. ZZ may be 0..99 for a total of 100 pairs of coordinates that may be stored.

The programmer is allowed to define a series of “LOCATIONS” within his system using the ENTER LOCATIONS SCREEN. Once the locations are defined it becomes easier to write a program that moves baskets of product from location to location.

In addition, if it becomes necessary to change the locations that the program goes to, the program itself does not need to be changed at every place in the program that the GO_LOCN command is used. Just the definitions of the LOCATIONS will need to be changed. This would not be the case if GO_X () and GO_Y() commands were used.

Allowing the programmer to specify which axis moves first makes programming easier and prevents collisions.

Programming Example:

The programmer wants to:

- Drop a basket off in station
- Do something else.
- Pick basket from station

Define Location 1 as above DROP point.

Define Location 2 as at DROP point

Define Location 3 as above PICK point.

Define Location 4 as PICK point

GO_LOCN (1,01)

GO_LOCN (1,04)

...Do other stuffing...

GO_LOCN (1,03)

GO_LOCN (1,02)

END

When the program reaches the END instruction program execution is terminated.

It is not necessary to place an END instruction at the end of a program. When a program reaches the last instruction it will automatically terminate.

Placing subroutines after an END instruction guarantees that the program will not enter the subroutine group without the use of a GOSUB. (Or, inappropriately, a GOTO, IF_IN_ON, or IF_IN OFF.)

Flow control instructions (GOTO, GOSUB, IF_IN_ON, and IF_IN_OFF) can “reach around” an END instruction to get to LABELs.

There may be multiple END instructions in a program.

Program Example

A programmer wants to write a program that runs until a sensor informs an input that there are no more baskets of product to run through the system.

The “DONE” sensor is connected to INPUT #1.

If the sensor is OFF there are no more baskets waiting to be run. If it is ON there are still baskets waiting to be processed.

```
LABEL 01  
instruction  
instruction  
instruction
```

```
IF IN_ON 01, 02  
END  
LABEL 02
```

```
instruction  
instruction  
GOTO 01
```

In the above code the program continues to perform the instructions in the loop (starting at LABEL 01 and looping from GOTO 01 until INPUT #1 is OFF then the program terminates at the END statement.

The IF_IN_ON instruction “jumps over” the END statement until the input indicates that the switch is OFF.

REPEAT (ZZZZ)

(ZZZZ = 0..9999, 0 = none, 9999 = continuous)

The REPEAT instruction causes the program to run from the start (the first instruction in the program)

to the point in the program where the REPEAT instruction is located ZZZZ times.

ZZZZ may be 0..9999.

The value of 9999 is interpreted to mean, “REPEAT forever”.

The value of 0 is interpreted to mean, do nothing.

The purpose of the REPEAT instruction is to allow the controller to perform a series of operations a fixed (ZZZZ) number of times.

When ZZZZ iterations are completed the program can pass through the REPEAT instruction and continues to run whatever program, if any, is located beyond the REPEAT instruction.

There should only be **ONE** REPEAT instruction in a program. The results of using multiple REPEAT instructions are predictable, repeatable, and undesirable.

NOTE:

The REPEAT instruction may NOT be placed within a subroutine.

Placing the REPEAT instruction within a subroutine will cause a program to terminate when the REPEAT instruction is encountered.

It is not possible to execute the REPEAT instruction properly, as described above, when it occurs within a subroutine.

AGITATE (NNNN)

(NNNN = 0..9999)

The AGITATE (NNNN) command causes the trolley arm to move down a short distance and then back up again NNNN times.

NNNN can be any number between 0 and 9999.

When used in a cleaning process AGITATE is often used to “shake off” excess liquid when a basket is removed from from a tank or to “slosh” the basket in the liquid.

AGITATE always moves down 20 counts and then up 20 counts in each cycle. 20 counts corresponds to about 7 ½” or 200mm.

It is important to be sure that there is room under the basket to move down the 20 counts when the AGITATE instruction is encountered. Without proper clearance there will be a crash.

The AGITATE command will not operate in the NO_MOTION test mode. (No motion-related commands will operated when in the NO_MOTION test mode.)

SCREENS

The various major functions of the RCT control system are accessed through various display screens on the terminal.

Note:

Screens will show both information and control buttons.

All *buttons* will be shown as “3-dimensional” with “beveled” edges.

All informational *displays* will have a “flat” image.

Some buttons may be “hidden” until needed to reduce screen clutter.

Some buttons may be “greyed out” when they are not available to be used.

Some buttons may be BOTH buttons (accepting “presses”) and informational (showing information).

Some multiple displays may be grouped by a beveled frame. This does not indicate the displays are buttons.

Available screens include:

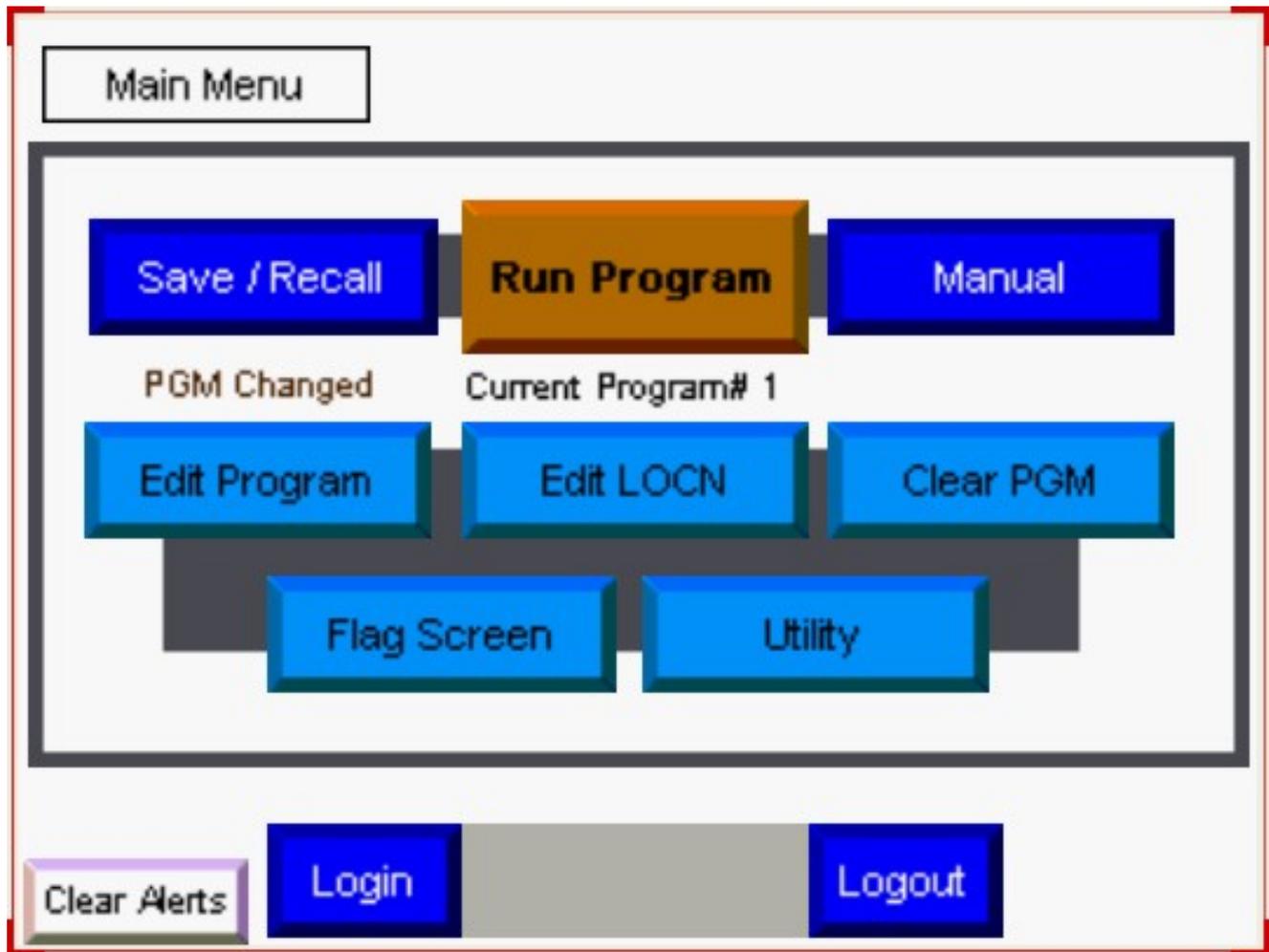
- **Main Menu Screen.** Provides access to all major functions. Always visible at turn-on.
- **Run Screen.** Screen must be visible to run the current program.
- **Manual Controls Screen.** Provides access to manual motion controls. Jog (up, down, left, right), GO_X, GO_Y, HOME_X, HOME_Y, and HOME.
- **Edit Program Screen.** Edit programs and/or get access to the Clear Program screen.
- **Edit Location Screen.** Enter, edit, clear, store LOCATIONS.
- **Clear Program Screen.** Clear (erase) the current program.
- **Change/View Flags.** View flags and inputs. Change flags/outputs.
- **Utility Screen.** Controls Single-step mode, No Motion mode, No Stall mode, Access display terminal configuration (system level maintenance only). Access flag screen. Access Screen 2. Access Change Password Screen.
- **Change Password Screen.** A user with a password may change their password.
- **Screen 2.** Direct access to any other screen. System maintenance only.

MAIN MENU SCREEN

The starting point for running an RCT system is the “MAIN MENU” screen.

When the system is first turned on the MAIN MENU is displayed (after initialization is completed).

The MAIN SCREEN (shown below) provides access to most of the major operating screens.



Most of the buttons on the MAIN MENU screen are used to access other screens. The exception are the buttons on the bottom of the screen that control access to restricted screens (User Login / Logout).

If a user has not logged in and attempts to use a restricted screen the screen will request a USER NAME and PASSWORD. If the password information is not supplied the user will not be given access to the requested screen.

Note:

Systems can be provided with passwords associated with any screen to limit access by unauthorized personnel.

Or, systems can be provided with no passwords if a system is to be located in a secure location that unqualified users can not reach.

By default, screens that can affect programs and their operation (editing and debugging) are password protected.

Screen Buttons

RUN PROGRAM

The most commonly used control on this screen should be the RUN PROGRAM button. Pressing this button will bring a user to the RUN PROGRAM screen from which the RCT can be commanded to run the currently selected program.

Note:

The RUN PROGRAM button does NOT actually start the user program. It brings the user to the page where the controls that can START and interact with a program are located.

SAVE RECALL

The SAVE-RECALL button brings the user to a screen that allows the current program to be saved to one of the four memory location or where a program stored in one of the four memory locations can be recalled as the CURRENT program.

MANUAL

The MANUAL button gives access to a screen with the MANUAL operating controls for the system. On this screen are the JOG, HOME, GO_X, and GO_Y commands.

EDIT PROGRAM

The EDIT PROGRAM button brings the user to a screen that allows editing of the current program. (The CURRENT program is the program which can be recalled from memory, edited, run, and saved to memory.)

EDIT LOCATION

The EDIT LOCATION button brings the user to a screen where the LOCATIONS used by the GO_LOCN command can be entered, edited, or cleared and the current position of the trolley may be saved to a “location” memory.

CLEAR PROGRAM

The CLEAR PROGRAM button brings the user a screen that can erase the CURRENT program.

FLAG SCREEN

The FLAG SCREEN button brings the user to a screen where FLAGS and INPUTs may be inspected and FLAGS and OUTPUTs may be changed. The FLAG screen may be accessed from the RUN screen while the program is running.

UTILITY

The **UTILITY** button brings the user to a screen where debugging modes (single-step, no-stall, no-motion) can be selected, user passwords may be changed, the OIT configuration can be changed, the **FLAG** screen can be accessed.

LOGGED USER

The gray band between the **LOGIN** and **LOGOUT** buttons shows the **USERNAME** of the currently logged user.

If there is no currently logged user then the gray band remains blank.

CLEAR ALERT

The **CLEAR ALERT** button appears in the lower left corner of the screen.

The **CLEAR ALERT** button is used to clear communication error alerts shown by the display terminal. This button should not need to be used unless there is a problem with the system or the system controller program is being changed.

(A similar **CLEAR ALERT** button appears on the “**UTILITY**” screen.)

LOGIN / LOGOUT

At the bottom of the screen are two password related buttons.

In some systems screens are protected by passwords. If a user attempts to access a protected screen he will be asked to provide a **USER NAME** and **PASSWORD** to proceed. Entering an invalid **USER NAME** and **PASSWORD** combination will prevent the user from reaching the protected screen.

Instead of waiting for the system to ask for a password to enter a screen, a user may elect to enter a valid **USER NAME** and **PASSWORD** when he begins to work with the system. Once a user has entered a valid ID he may continue to access protected screens until the unit is turned **OFF** or he presses the **LOG OUT** button.

LOGIN

Pressing the **LOGIN** button will bring up a window asking for the operator's **USERNAME** and **PASSWORD**.

Once a valid **USERNAME** and **PASSWORD** have been entered the logged in **USERNAME** will appear in the gray area between the **LOGIN** and **LOGOUT** buttons.

When entering **USERNAME** and **PASSWORD**

- **USERNAME** and **PASSWORD** are case sensitive. “**ABCD**” and “**abcd**” are **NOT** the same.
- Pressing the on-screen “**ESC**” button cancels the entry process.
- The **TAB** button (←/ →) moves from **USERNAME** window to **PASSWORD** window.
- The **ENTER** key is used to complete the entry.

LOGOUT

The LOGOUT button cancels the last LOGIN.

Pressing the LOGOUT button cancels the last LOGIN. The system is left with no user logged in.

For most installations it is recommended that users, especially privileged users (SUPERvisors) LOGOUT before leaving the machine unattended.

IMPORTANT:

Users with supervisory IDs (IDs that give full access to the RCT functions) should remember to LOG OUT when they leave the RCT.

Failing to do so may unintentionally give unqualified users supervisory access to the system.

Turning OFF power automatically performs a LOGOUT.

Screen Indicators.

LAST SAVED or RECALLED

The indicator in the center of the screen, below the “Run Program” button, shows the number of the last program that was saved or recalled from a memory.

PROGRAM CHANGED

To the right of the “last-program” indicator is a “Program Changed” indicator.

This will be visible only if the current program has been changed/edited after being recalled or after being saved. It is a reminder that the CURRENT program is NO LONGER exactly the same as the program stored in the memory location shown.

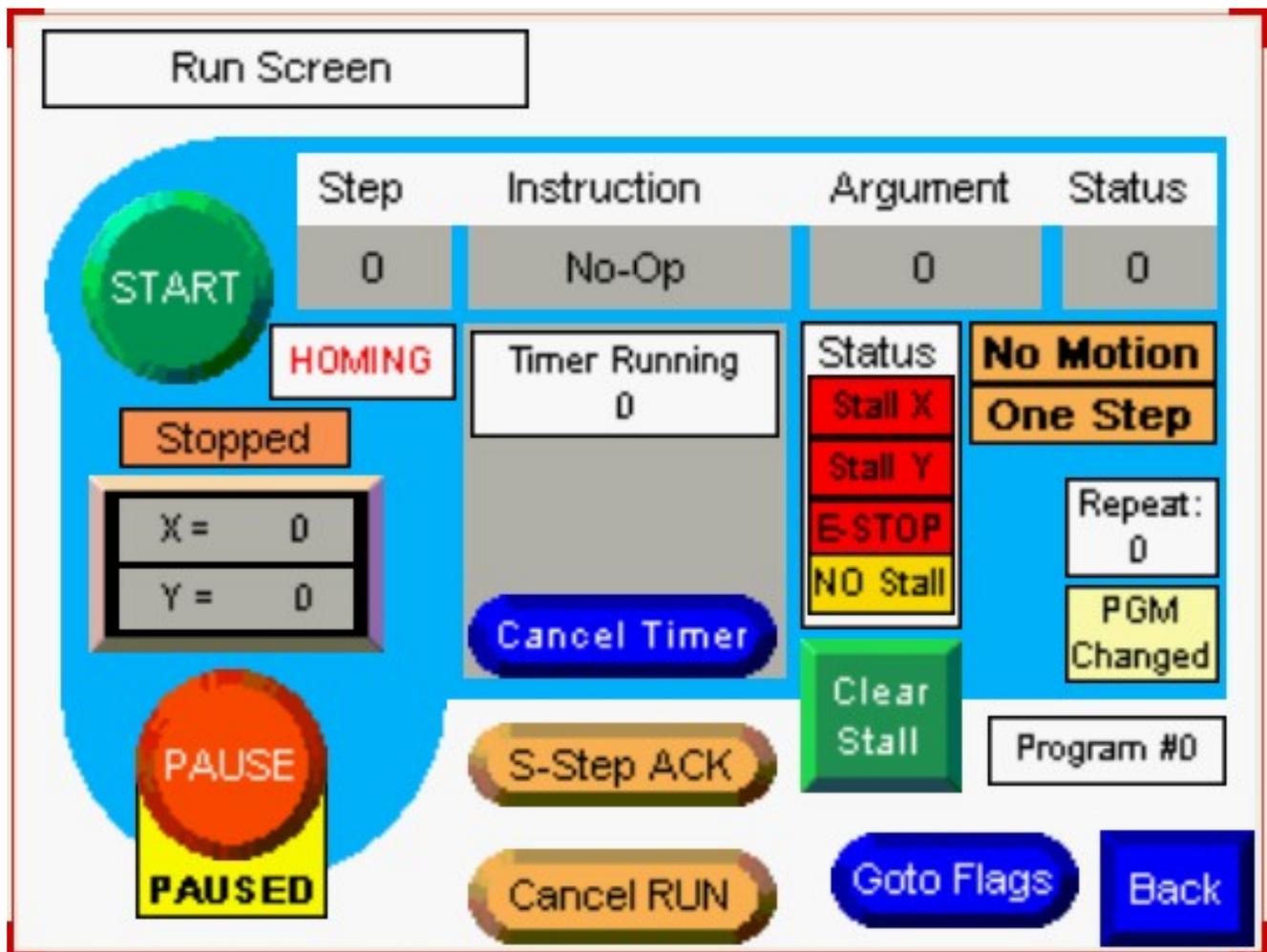
RUN SCREEN

The RUN SCREEN is accessed from the MAIN MENU.

The RUN SCREEN allows a user to start a program running and, if necessary, interact with the program.

The screen image below shows the RUN SCREEN with most of its controls and indicators visible. The only objects that are not visible are “behind” other objects and become visible when needed.

Buttons and indicators are only visible when appropriate; at other times they are hidden from view.



Screen Buttons

RUN

The RUN button starts the current user program.

If the button is GREY it is unavailable – when a program is already running.

PAUSE

The PAUSE button will pause a running program, that includes temporarily stopping all motion, and temporarily turning OFF all program controlled outputs.

The flashing, yellow “PAUSED” indicator below the button is only visible when the PAUSE function is active.

The PAUSE button toggles – press and it turns ON, press again and it turns OFF.

Important!

The PAUSE button is NOT a replacement for the E-STOP button and should never be used to prevent motion when the motion could possibly injure an operator or otherwise cause damage.

It is very unlikely that the PAUSE function will fail in such a way as to allow injury BUT it is possible and therefore should never be trusted to provide safety. The E-STOP button(s) remove power to the motors so that driven motion should be impossible.

CONTINUE

The CONTINUE button is used by an operator to acknowledge a current WAIT instruction.

When a running program encounters a WAIT instruction it stops further instruction execution until the CONTINUE button is pressed.

The program will wait indefinitely for a press of the CONTINUE button.

NOTE:

The CONTINUE button is not visible in the screen image shown because it is located “under” the “Timer Running” indicator.

During normal program operation the display will show either a description of the current program step, the “Timer Running” indicator, or the CONTINUE button. Only one can (or needs to be) show at a time.

CANCEL

The CANCEL TIMER button only appears when the TIMER function is running in a program.

Pressing the CANCEL TIMER will cancel the program's TIMER operation and cause the program to move on to its next step without completing the timing function.

After the TIMER has been canceled, the CANCEL TIMER button and the TIMER countdown display will disappear from the screen.

The CANCEL TIMER button has no direct effect on TIMED_PER instructions.

S_STEP ACK

The S_STEP ACK button (“single step acknowledge”) is only visible and active when the RCT is running a program in the “SINGLE STEP” test mode.

When running a program in the SINGLE STEP MODE the program runs a single step of the program

and then waits for the user to press the S-STEP ACK button.

When the S-STEP ACK button is pressed the program runs the next instruction and then waits...

The SINGLE STEP MODE is used for developing / debugging programs.

The control to turn ON/OFF the SINGLE STEP MODE appears on the UTILITY SCREEN.

An ORANGE “One Step” indicator appears on the right side of the RUN SCREEN when the SINGLE STEP mode is active.

CANCEL RUN

The CANCEL RUN button is used to completely stop (terminate) a running program.

The CANCEL RUN button is only visible when the current program is running.

The CANCEL RUN button may not stop a program immediately – it is checked by the program between instructions.

The CANCEL RUN button should NEVER be used in place of the E-STOP button when safety is involved. The E_STOP button should remove all power from the motors IMMEDIATELY while the CANCEL RUN function may involve some delays.

CLEAR STALL

The CLEAR STALL button is only visible when a motor stall has been detected.

If a stall has occurred the running program will terminate, a “STALL X” or “STALL Y” indicator will show, all motion will be inhibited, and the CLEAR STALL button will show.

Pressing the CLEAR STALL button will reset a current STALL and re-enable normal system operation.

Important:

The underlying cause of a STALL should be corrected before using the system.

The cause may be as simple and obvious as an unbalanced basket that toppled over but it may also indicate a less obvious system failure.

GOTO FLAGS

The GOTO FLAGS button provides access to the VIEW / CHANGE FLAGS screen.

The GOTO FLAGS button turns OFF the current RUN screen and turns ON the VIEW / CHANGE FLAGS screen.

Once in the VIEW / CHANGE FLAGS screen a user may view FLAGS and INPUTS or change FLAGS and OUTPUTS.

The VIEW / CHANGE FLAGS screen is useful during program debugging. The screen is accessible

during normal and SINGLE STEP operation.

The BACK button on the VIEW / CHANGE FLAGS screen returns to the calling screen. In this case, the RUN SCREEN.

BACK

The BACK button, when visible, is located in the lower right hand corner of this screen. It turns OFF this screen and returns to the previous screen (usually the MAIN MENU screen).

The BACK button is only visible when a program is NOT running. It is not appropriate to leave the RUN screen when the current program is running so the button is hidden when the program is running.

Screen Indicators

CURRENT STEP

Across the top of the screen is a bar that shows the current program step number and information about the step.

Included in the displayed information is: The step number of the current instruction, the name of the instruction, any argument or arguments used by the instruction, and the current RUN status (as a number code).

HOMING

The HOMING indicator appears when the RCT is performing a HOME function. (In the process of moving to full UP then full OVER.)

The RCT must have been HOMEd before its control system can accurately position the trolley.

If the RCT has not been homed after being turned ON and before a program is run, the RCT will automatically perform a HOME operation. The indicator will show that a HOME function is being performed.

Stopped / RUNNING

This indicator shows the current RUN status of the RCT.

The status can show “RUNNING” (a program is currently running) or “Stopped” (a program is not currently running).

This indicator is always visible on this page.

X-Y Position Indicator

Two line display shows the X and Y coordinates of the trolley's current position.

Note:

The X-Y position indicator may show slightly delayed position information due to the communications delays between the display and the controller. The delay is normal. The controller always knows where the trolley is.

Note:

The X-Y Position Indicator groups the X and Y position indicators within a beveled box for clarity. These indicators have no button functions.

Instruction Description

The gray box below the displayed INSTRUCTION shows, briefly, what the instruction does.

The Instruction Description may be covered by the "Timer Running" indicator or the CONTINUE button (associated with the WAIT instruction).

System Status Indicators

A rectangular box under the ARGUMENT window shows various elements of the system's status.

The system status indicators show:

- Stall X. A STALL error has occurred on the X axis. The system requires the "CLEAR STALL" button be pressed to clear the error before any motion or program may be used.
- Stall Y. A STALL error has occurred on the Y axis. The system requires the "CLEAR STALL" button be pressed to clear the error.
- E-STOP. The E-stop button has been pressed. The E-STOP must be turned OFF and the SYSTEM ENABLE button on the control panel must be pressed to cancel this error.
- NO Stall. Indicates that the "NO STALL" mode has been selected on the UTILITY page. No STALL errors are possible until the "NO STALL" mode is turned OFF.

Stall X, Stall Y and E-STOP each indicate that a serious error has occurred, the program has been terminated and all motion is inhibited until suitable action has been taken by the operator.

The "NO STALL" mode turns off STALL (X and Y) detection. This mode is intended for use when the system is being installed and setup. The "NO STALL" mode should not be used during normal operation.

No Motion

The No Motion indicator shows when the NO MOTION mode has been selected on the UTILITY page.

The NO MOTION mode allows programs to be run (for testing and debugging) without executing any of the motion commands that may appear in the program. All other instructions will run normally.

NO MOTION may be used in conjunction with ONE STEP to test and debug programs. This combination allows the operator to walk through a program and follow its logic without having to wait for the trolley to move.

NO MOTION suppresses the following commands:

- Go_X
- Go_Y
- Go_LOCN
- HOME
- AGITATE

One Step

The One Step indicator shows when the ONE STEP mode has been selected on the UTILITY page.

The ONE STEP mode is a diagnostic tool. When ONE STEP mode is active the RCT will run a single step in the program and then stop and wait for the S-STEP ACK button to be pressed. When the S-STEP ACK button is pressed the program will run another step and then wait for the... In this way the effect of each step in a program can be clearly observed and its effects on program operation accessed.

ONE STEP may be used in conjunction with NO MOTION to test and debug programs. This combination allows the operator to walk through a program and follow its logic without having to wait for the trolley to move.

REPEAT

The REPEAT indicator is active when a REPEAT instruction is currently in effect.

The REPEAT indicator shows how many REPEATs remain before the REPEAT instruction is completed.

The REPEAT XXXX instruction specifies that all steps between the beginning of the program and the REPEAT instruction are to be performed XXXX times before the program can progress beyond the location of the REPEAT instruction.

The REPEAT instruction may NOT be placed within a subroutine. Program execution will be terminated if the REPEAT instruction occurs within a subroutine.

PGM CHANGED

The PGM CHANGED indication means that the current program is no longer identical to the stored program. This condition may be intentional or unintentional, hence the indicator.

The PGM Changed indicator becomes visible if the program was edited after being recalled from or saved to memory.

Program #X

The Program #X indicator shows which program was last recalled from or saved to memory.

The system supports 4 memory locations in which the CURRENT program may be stored. (X may be

1 through 4, inclusive.)

If the CURRENT program has been modified or freshly entered the “PGM Changed” indicator will be visible to show that the CURRENT program is not the same as the last program memory location accessed.

The Program #X indicator is always visible on the RUN screen.

EDIT PGM SCREEN

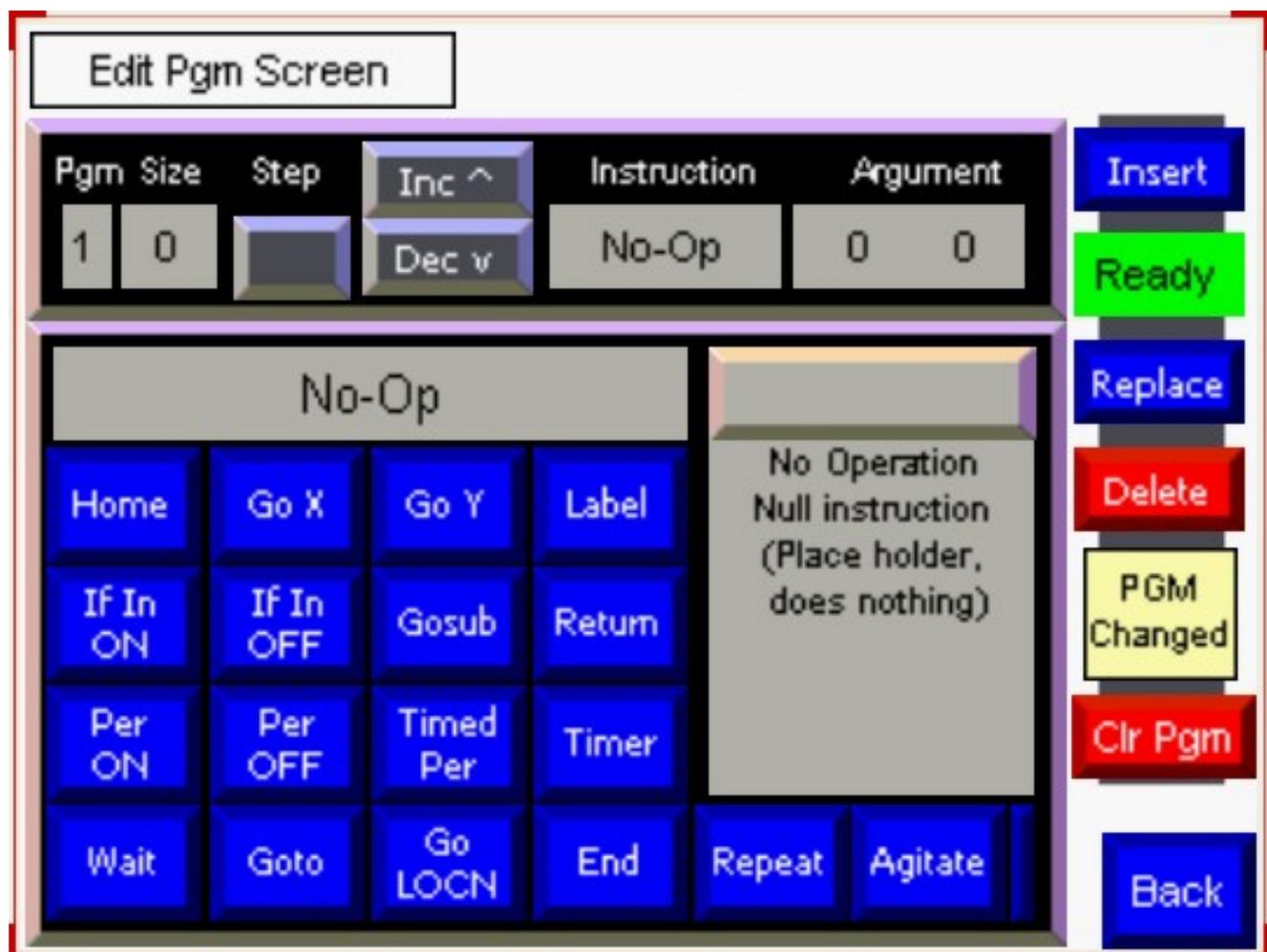
The EDIT PGM SCREEN contains the controls and indicators that are necessary to create, edit, or clear (delete) the CURRENT program.

The EDIT PGM SCREEN (shown below) has its screen objects (buttons and indicators) grouped into three section.

The TOP section shows information about the CURRENT program and allows viewing individual instructions already in the CURRENT program.

The BOTTOM section allows the user to select, compose, and edit an instruction to be placed into the CURRENT program. Instructions that have been assembled may be INSERTED immediately after the displayed instruction or may REPLACE the currently displayed instruction.

The RIGHT section has the controls and indicators needed to Insert, Replace, and Delete instructions in the program and to clear (delete) the entire program.



TOP SECTION

The TOP SECTION of the EDIT PGM SCREEN (enclosed in a beveled frame) has controls and indicators are necessary to select and view CURRENT program instructions.

NOTE:

Recall that the CURRENT program is the designation for the program that can be edited, run, saved, or cleared (deleted).

Pgm

The PGM indicator show which program has been most recently recalled from, or saved to memory.

NOTE:

- There are four (1..4) memory locations in which to store programs or from which to recall programs.
- The program being edited is always the CURRENT program.

SIZE

The SIZE indicator shows how many steps (total) there are in the CURRENT program.

NOTE:

- The maximum size of a program is 400 steps.
- The minimum size of a program is 1 step.
- When a program is CLEARed the system inserts a NO-OP (null) instruction in order to have at least one instruction.

STEP

This is both an indicator and a button.

The indicator shows which step in the CURRENT program is being displayed (1..400).

As an indicator, this display works with the INC and DEC buttons to allow the user to select step forward (INC) or backwards (DEC) through the program steps. Button presses that attempt to view program steps beyond the ends of the program are ignored.

Pressing the indicator allows the user to enter the number of the program step in the CURRENT program to view. Entries beyond the length of the current program will not be accepted.

INC^

The INC^ button increments the program step being displayed on the STEP indicator. Press this button to show the next step in the program.

Pressing the INC^ when the STEP shown is the last instruction in the program will have no effect. (Users are prevented from INCrementing beyond the end of the program.)

DECv

The DECv button decrements the program step being displayed on the step indicator. Press this button to show the previous step in the program.

Pressing the DECv button when the program step being displayed in STEP is the first instruction in the program will have no effect. (Users are prevented from DECrementing beyond the beginning of the program.)

Instruction

The INSTRUCTION display shows the type of instruction in the current step of the program being edited. The instruction types include:

- HOME
- GO X
- GO Y
- LABEL
- IF IN ON
- IF IN OFF
- GOSUB
- RETURN
- PER ON
- PER OFF
- TIMED PER
- TIMER
- WAIT
- GOTO
- GO LOCN
- END
- REPEAT
- AGITATE

Argument

The Argument indicator shows the argument(s) section of the instruction being displayed.

Instructions may have none, one or two arguments (specific to the instruction's requirements) and the ARGUMENT indicator will show the argument(s) (if any) associated with the current instruction.

BOTTOM SECTION

Directly below the Top Section is the Bottom Section.

The Bottom section is an area enclosed in a beveled frame consisting of individual buttons for each type of instruction, a window to show the type of instruction chosen, a window to show the instruction's argument(s), button or buttons to enter arguments (if any are required for the particular instruction), and a window to tell what the instruction does and how the argument(s) are used by the instruction.

The Bottom Section is used to “assemble” an instruction in order to be able to include the instruction in the CURRENT program.

INSTRUCTION

The instruction display is a window in the upper left side of the BOTTOM SECTION that displays the instruction that is being assembled.

Instructions must be assembled before being included in the CURRENT program (via INSERT or REPLACE commands)

To build an instruction:

- Select the type of instruction to build by pressing one of the blue buttons in the LOWER SECTION.
- Adjacent to the INSTRUCTION display is the ARGUMENT display. None, one, or two arguments will be automatically displayed depending on the type of instruction selected.
- If an argument is displayed in the window, press its displayed button to be allowed to enter the value for the argument. Out of range entries will be rejected.
- The display window below the ARGUMENT display will show a brief description of the instruction and its arguments (as a reminder).

ARGUMENT(S)

The display / buttons alongside the INSTRUCTION display show any arguments appropriate to the selected instruction and allow the user to enter the values.

To enter the value of a displayed ARGUMENT press the argument's button and use the keypad that appears in order to enter the appropriate value. Out of range entries will be rejected.

INSTRUCTION DESCRIPTION

The gray window below the ARGUMENT(S) will show a brief description of the instruction selected for entry and its associated argument(s) (if any).

RIGHT SECTION

The right section has controls and indicators that are used to INSERT, REPLACE, and DELETE instructions in the CURRENT program.

It also contains a button that call up the CLEAR PROGRAM SCREEN where the CURRENT program can be deleted.

Indicators show insert/replace status (READY / BUSY) and whether the CURRENT program has changed since it was last saved or recalled.

INSERT

The INSERT button takes the instruction displayed in the lower section (the one most recently assembled) and “inserts” it into the program immediately after the instruction step displayed in the upper section IF the current step is not NO_OP and there are fewer than 400 steps in the CURRENT program.

Pressing the INSERT button:

- Checks to see if the program is less than 400 steps. If it is 400 steps, nothing is done.
- Changes the READY / BUSY indicator to show BUSY.
- Inserts the new instruction in the current program AFTER the instruction displayed in the upper section.
- Changes the READY / BUSY indicator to show READY.
- Increments the program step in the UPPER SECTION to show the just-inserted instruction.

Note:

The INSERT button will be “grayed out” if the current step is NO_OP.

If the program step is NO-OP the REPLACE button must be used.

READY / BUSY

The INSERT and DELETE functions can take a couple of seconds to complete. The READY / BUSY indicator shows an amber backed BUSY while the process is underway and a green backed READY when the process is completed.

No additional INSERT, DELETE or REPLACE commands may be issued while the system shows BUSY.

REPLACE

Pressing the replace button causes the RCT to replace the instruction being shown in the upper section with the new instruction that has been prepared in the lower section.

The existing instruction is lost and the new instruction takes its place in the CURRENT program.

NOTE:

When the CURRENT program has just been cleared (all steps have been deleted) the system will place a NO-OP instruction in instruction #1 because programs must consist of at least one instruction.

The NO-OP instruction may not be DELETED while it is the ONLY instruction – programs must consist of at least one instruction. (Actually, if the last remaining instruction in a program is deleted it is replaced with a NO-OP instruction. So if a NO-OP instruction is the last remaining instruction in a program and it is deleted – it is replaced with another NO-OP instruction.)

A new instruction will REPLACE the NO-OP.

It is not possible to INSERT a new instruction after a NO-OP – the NO-OP must be REPLACED.

The NO-OP may not be deleted.

Clr Pgm

The Clr Pgm button will take the user to the CLEAR PROGRAM SCREEN. A separate screen is used for deleting a program to make it more difficult to accidentally delete a program.

On the clear program page the user will be warned that there is no way to UNDO a CLEAR

PROGRAM command once issued.

The CLEAR PROGRAM SCREEN has a large red CLEAR PROGRAM button that may be used to actually clear the program. The button will change from RED to WHITE while pressed.

When the user has completed using the CLEAR PROGRAM screen the small, blue, "BACK" button will return the user to the EDIT PGM SCREEN.

BACK

The small, blue, "BACK" button in the lower right corner will return the user to the previous screen. (Usually the MAIN MENU.)

Note:

No information is lost when the BACK button is used to depart the EDIT PGM SCREEN.

The operator may re-enter the EDIT PGM SCREEN at a later time and will find the program unchanged unless the operator CLEARS the program from a different screen.

The BACK button may be pressed at any time.

MANUAL CONTROLS SCREEN

The MANUAL CONTROLS screen (shown below) contains the controls and indicators used to manually control the motion of the trolley.

The screen contains controls for:

- Jog UP, Jog DN, Jog LT, Jog RT
- HOME
- HOME X
- HOME Y
- GO_X
- GO_Y
- PAUSE
- Cancel Automatic Motion
- Clear Stall
- Move to the FLAG SCREEN
- BACK (return to calling screen).

The screen contains indicators for:

- X location
- Y location
- Pause Active (paused)
- Stall X
- Stall Y
- E-Stop
- No Stall Mode
- Location to GO_X
- Location to GO_Y
- NO Motion Mode.

IMPORTANT

Manual jog controls require that the RCT display communicate with the RCT controller in real time.

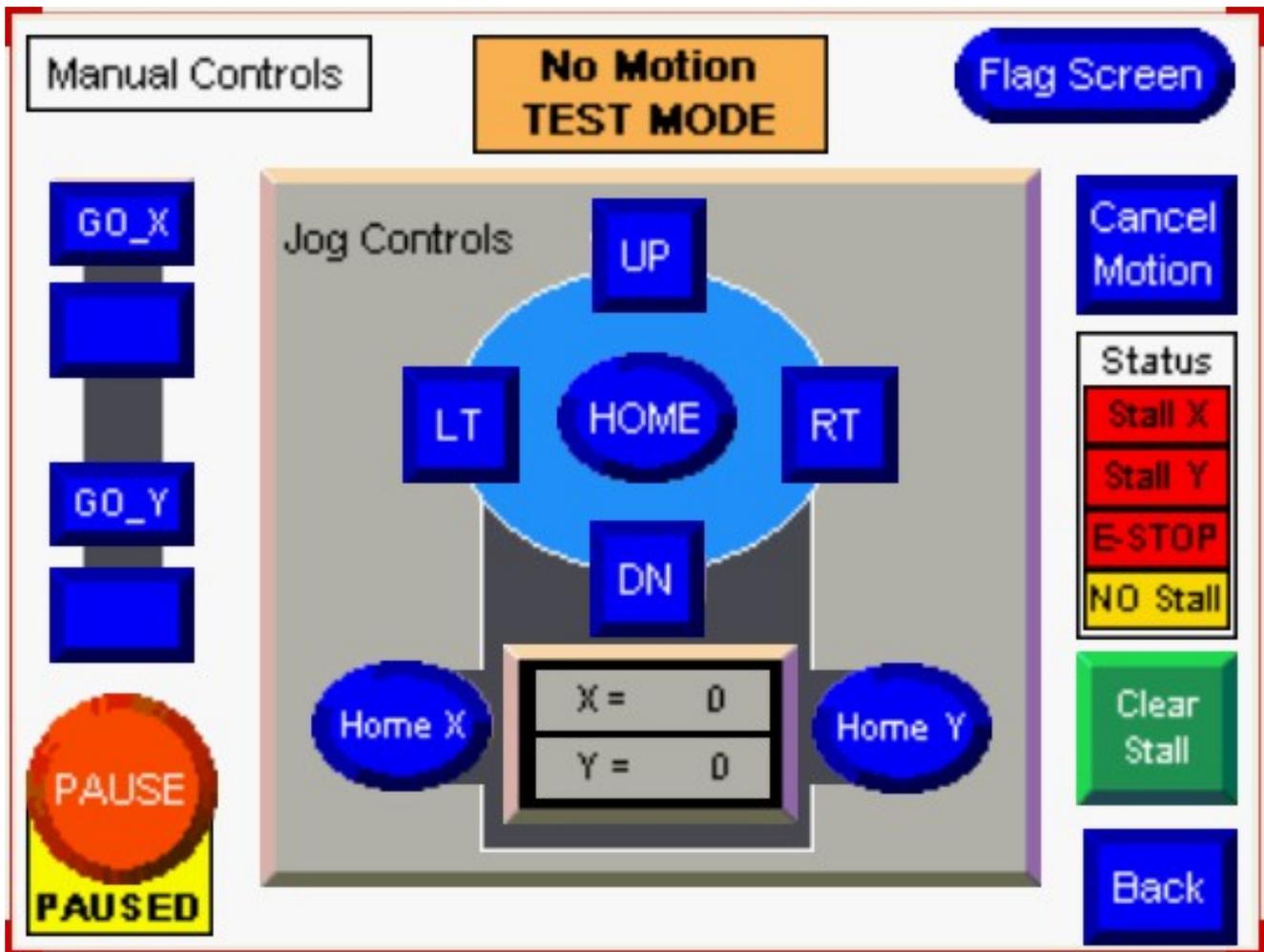
The communications between the display and the controller tends to be a little slow. It can take a noticeable fraction of a second between the time that a JOG button is pressed and the time that trolley motion begins.

It also takes a noticeable fraction of a second between the time that a JOG button is released and the time that trolley motion ceases.

This delay can be annoying when trying to make exact moves with the trolley.

With a little of experience (anticipating the delays) a user can usually get the trolley to move as desired.

If the coordinates of a location are known to the user the manual GO_X and GO_Y button commands can be more satisfying to use.



Screen Buttons

Note:

Most systems have a horizontal HOME location to the LEFT.

If the HOME location is LEFT the jog LT button will not function when the trolley is at the LEFT most travel location (HOME).

If the HOME location is RIGHT the jog RT button will not function when the trolley is at the RIGHT most travel location (HOME).

In all systems, the jog UP button will not work when the trolley is at the vertical home (TOP) position.

Important!

It is possible to jog the trolley so as to "crash" into part of the system (a tank, end of travel, etc.)

It is the operator's responsibility to avoid crashes and the possible damage they can cause.

UP

The UP button will jog the trolley in the UP direction.

The UP button will not work if:

- The trolley is at the vertical HOME location.
- The system is PAUSED.
- The system has encountered a stall (vertical or horizontal).
- The system has been E-STOPped.
- The system is in NO_MOTION mode.

DN

The DN button will jog the trolley in the DOWN direction.

The DN button will not work if:

- The system is PAUSED.
- The system has encountered a stall (vertical or horizontal).
- The system has been E-STOPped.
- The system is in NO_MOTION mode.

LT

The LT button will jog the trolley in the LEFT direction.

The LT button will not work if:

- The trolley is at the left horizontal HOME location. (On machines with a left horizontal home.)
- The system is PAUSED.
- The system has encountered a stall (vertical or horizontal).
- The system has been E-STOPped.
- The system is in NO_MOTION mode.

RT

The RT button will jog the trolley in the RIGHT direction.

The RT button will not work if:

- The trolley is at the right horizontal HOME location. (On machines with a right horizontal home.)
- The system is PAUSED.
- The system has encountered a stall (vertical or horizontal).
- The system has been E-STOPped.
- The system is in NO_MOTION mode.

HOME

The blue oval HOME button performs the automatic HOME function.

- The trolley first travels UP to the TOP (vertical HOME location).
- The trolley then travels over to the END (horizontal HOME location).

Motion will be inhibited:

- The PAUSE button will cause HOMING to PAUSE until the system is UN-PAUSED.

- The CANCEL MOTION button will completely cancel a current HOME operation at any time during the HOME process.
- A current STALL (X of Y) will prevent HOME operation.
- A current E-STOP will prevent HOME operation.
- If the system is in NO_MOTION mode HOME operation will be prevented.

HOME_X

The blue oval HOME_X button performs an automatic home function on the X (horizontal) axis only. The Y axis remains unchanged.

Motion will be inhibited:

- The PAUSE button will cause HOME_X to PAUSE until the system is UN-PAUSED.
- The CANCEL MOTION button will completely cancel a current HOME_X operation.
- A current STALL (X of Y) will prevent HOME_X operation.
- A current E-STOP will prevent HOME_X operation.
- The system is in NO_MOTION mode.
- If the system is in NO_MOTION mode HOME_X operation will be prevented.

HOME_Y

The blue oval HOME_Y button performs an automatic home function on the Y (vertical) axis only. The X axis remains unchanged.

Motion will be inhibited:

- The PAUSE button will cause HOME_Y to PAUSE until the system is UN-PAUSED.
- The CANCEL MOTION button will completely cancel a current HOME_Y operation.
- A current STALL (X of Y) will prevent HOME_Y operation.
- A current E-STOP will prevent HOME_Y operation.
- If the system is in NO_MOTION mode HOME_Y operation will be prevented.

GO_X

The GO_X button works in conjunction with the “TO_X” indicator/button.

Pressing the GO_X button cause the trolley to move to the location XXXX shown in the “TO_X” indicator / entry button.

The GO_X command is used to move the trolley to a specific location. This can be to test a location for use in a program or to simply position the trolley in a known position.

Note:

The system MUST have been HOMED at least once since power-on for the GO_X command to work correctly.

If the system has not been homed it will not know “where it is” and the GO_X command can have unpredictable results.

Motion will be inhibited:

- The PAUSE button will cause GO_X to PAUSE until the system is UN-PAUSED.

- The CANCEL MOTION button will completely cancel a current GO_X operation.
- A current STALL (X of Y) will prevent GO_X operation.
- A current E-STOP will prevent GO_X operation.
- If the system is in NO_MOTION mode GO_X operation will be prevented.

GO_Y

The GO_Y button works in conjunction with the “TO_Y” indicator/button.

Pressing the GO_Y button cause the trolley to move to the location YYYY shown in the “TO_Y” indicator / entry button.

The GO_Y command is used to move the trolley to a specific location. This can be to test a location for use in a program or to simply position the trolley in a known position.

Note:

The system MUST have been HOMED at least once since power-on for the GO_Y command to work correctly.

If the system has not been homed it will not know “where it is” and the GO_Y command can have unpredictable results.

Motion will be inhibited:

- The PAUSE button will cause GO_Y to PAUSE until the system is UN-PAUSED.
- The CANCEL MOTION button will completely cancel a current GO_Y operation.
- A current STALL (X or Y) will prevent GO_Y operation.
- A current E-STOP will prevent GO_Y operation.
- If the system is in NO_MOTION mode GO_Y operation will be prevented.

PAUSE

The PAUSE button will cause ALL motion to be suspended and all outputs to be temporarily set to OFF.

The PAUSE button toggles:

- Press to PAUSE.
- Press again to unPAUSE.

A yellow indicator flag will appear at the bottom of the PAUSE button when PAUSE is active.

FLAG SCREEN

The blue FLAG SCREEN button in the upper right corner of the screen will call the FLAG SCREEN to replace the MANUAL SCREEN.

Controls on the FLAG SCREEN will allow a user to view all INPUTs and FLAGs and to edit all OUTPUTs and FLAGs.

When the user has completed using the FLAG SCREEN he may return to the MANUAL SCREEN by pressing the blue BACK button located in the lower right hand corner of the FLAG SCREEN.

The FLAG SCREEN will allow a user to monitor system system inputs (is there a basket in the load station?), control outputs (open the sliding door over the dryer station) and monitor program operation (is there a flag indicating a basket in station #2).

CANCEL MOTION

The CANCEL MOTION button may be used to cancel any automatic motion command issued on the MANUAL SCREEN.

CANCEL MOTION will terminate the HOME, HOME_X, HOME_Y, GO_X, and GO_Y commands.

CANCEL MOTION is unlike PAUSE in that PAUSE prevents all motion (and inhibits all OUTPUTS) temporarily and CANCEL MOTION cancels any current automatic motion command immediately.

STALL CLEAR

The STALL CLEAR button will clear any current STALL_X or STALL_Y conditions.

The STALL CLEAR button will only be visible when a STALL condition exists.

STALLs are shown by the red indicators in the STATUS display.

STALL CLEAR cannot clear a E-STOP condition. E-STOP conditions are cleared by releasing the E-STOP button that was pressed.

Screen Indicators

X Position Indicator.

The X POSITION indicator shows the current X count.

The X POSITION and Y POSITION indicators are located in the beveled box at the bottom of the screen.

The X POSITION should be accurate if the system has been HOMED after having been turned ON.

Each count represents about 3/8" (10mm) displacement from the X HOME location.

The number shown in the X position indicator may be used to define a LOCATION or specify a destination in a GO_X command.

Note:

When preparing to write a program that moves baskets the user must determine the counts of the locations within the system they want the RCT to move to.

The controls on the manual page will allow a user to position the RCT trolley where they will want the program to

move the trolley.

Once the trolley is in the required position a user may copy down the X and Y positions (counts) from the position indicators and use these position to write their program.

Note:

The X Position Indicator on the display must communicate with the controller to get the position to display. The delays inherent in this communications can result in the display being slightly “behind” the motion of the trolley.

Y Position Indicator.

The Y POSITION indicator shows the current Y count.

The X POSITION and Y POSITION indicators are located in the beveled box at the bottom of the screen.

The Y POSITION should be accurate if the system has been HOMED after having been turned ON.

Each count represents about 3/8” (10mm) displacement from the Y HOME location. Y HOME is at the top of travel.

The number shown in the Y position indicator may be used to define a LOCATION or specify a destination in a GO_Y command.

Note:

When preparing to write a program that moves baskets the user must determine the counts of the locations within the system they want the RCT to move to.

The controls on the manual page will allow a user to position the RCT trolley where they will want the program to move the trolley.

Once the trolley is in the required position a user may copy down the X and Y positions (counts) from the position indicators and use these position to write their program.

Note:

The X Position Indicator on the display must communicate with the controller to get the position to display. The delays inherent in this communications can result in the display being slightly “behind” the motion of the trolley.

STATUS Indicators

The STATUS window on the right side of the screen shows several different status indicators:

- STALL X shows when there has been a stall detected during X motion. Once a stall occurs all motion is inhibited until the situation is remedied and the CLEAR STALL button is pressed.
- STALL Y shows when there has been a stall detected during Y motion. Once a stall occurs all motion is inhibited until the situation is remedied and the CLEAR STALL button is pressed.
- E-STOP shows when an E-STOP button has been pressed or the system has not been enabled. To restore operation the E-STOP button must be reset and the hardware (not screen) SYSTEM ENABLE button must be pressed.
- NO Stall shows when the NO STALL MODE has been selected on the UTILITY screen. The NO STALL MODE instructs the system to ignore situations that would normally trigger a STALL. This function must be used with great care to prevent operator injury and system

damage.

GO_X Position Indicator

The GO_X Position indicator shows the destination of the manual GO_X command.

GO_X Position indicator is located directly below the manual GO_X button.

Pressing the GO_X Position indicator will allow the user to enter a destination for the next GO_X manual command.

NO MOTION TEST MODE

The NO MOTION TEST MODE indicator at the top of the screen shows that the NO MOTION mode has been selected in the UTILITY screen.

The NO MOTION TEST MODE indicator is only visible when the mode is active. At other times the indicator is "hidden".

The NO MOTION TEST MODE suppresses all automatic motion. Its primary intent is to allow programs to be run without having to wait for the trolley to move and is usually used when programs are being tested.

In this screen it prevents the GO_X, GO_Y, HOME, HOME_X, and HOME_Y controls from working.

SAVE / RECALL PGM SCREEN

The SAVE/RECALL PGM SCREEN allows a user to:

- Save the CURRENT program to any of the four (1..4) memory locations.
- Recall the contents of any one of the four (1..4) memory locations into the CURRENT memory.

It is important to note:

- When the CURRENT program is saved into a memory location, whatever contents were previously in the memory location will be overwritten.
- When a program is recalled from a memory location into the CURRENT program the previous contents of the CURRENT program will be overwritten.

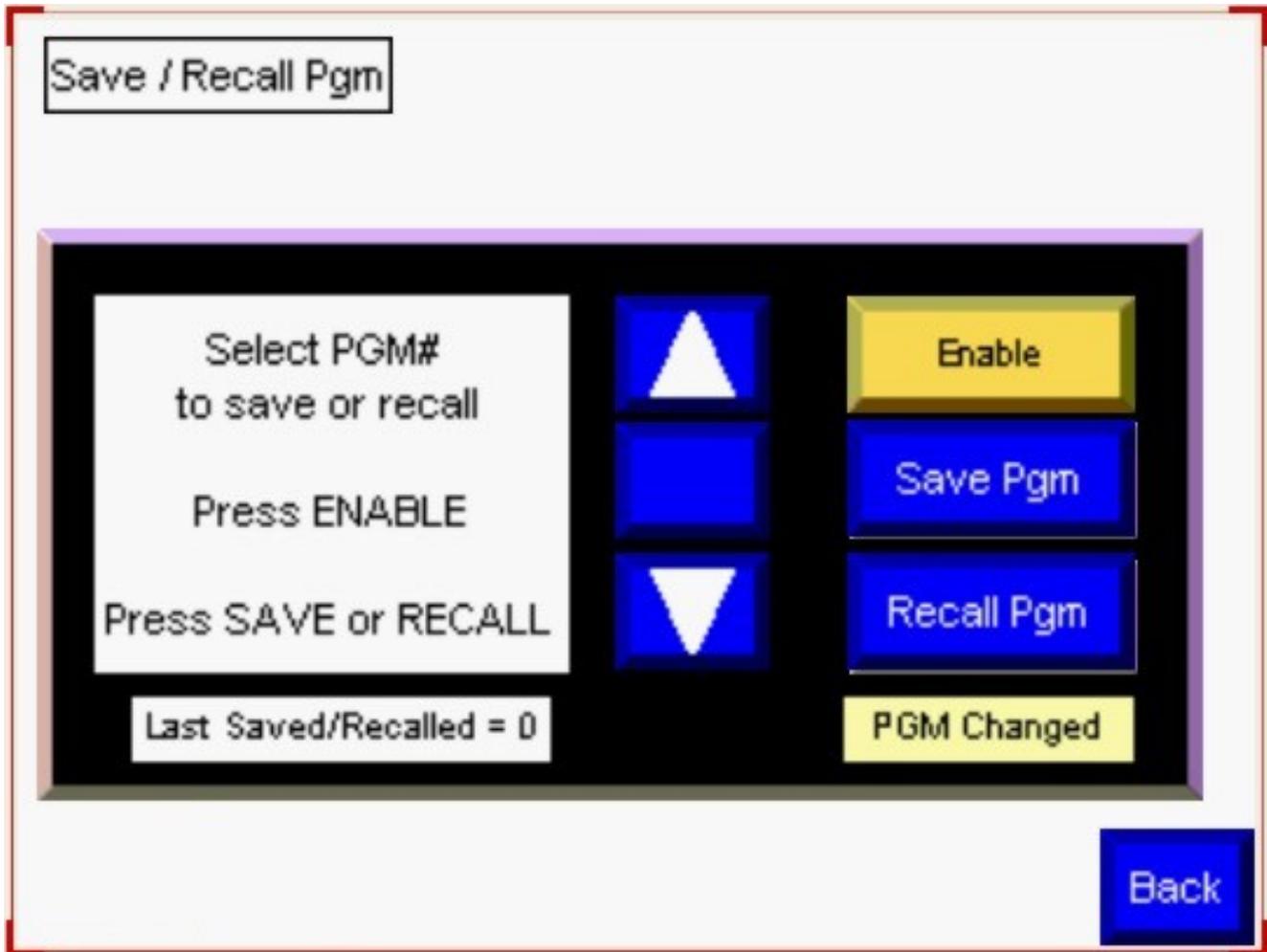
Recall that system memory is broken down into five program storage locations.

- The CURRENT program location is the only location that can be directly EDITED, CLEARED, or RUN.
- The four (1..4) memory locations are used to store programs for future use.

The screen is setup so that it requires a user to perform several actions to save or recall a program. This is intentional. It is to prevent the user from inadvertently losing a program.

The screen shows a description of the steps required to SAVE or RECALL a program.

- Select the memory location to save/recall using the UP/DN arrows OR press the box showing the current number and key in the new number on the keypad that displays.
- Press the ENABLE key. This will cause the “faded” SAVE PGM and RECALL PGM buttons to become available for use. The buttons will change to dark blue when ready.
- Press the SAVE PGM to save the CURRENT program in the memory location selected.
- OR press the RECALL PGM to recall a program from the memory location selected into the CURRENT program location.



Screen Buttons

Note;

The large white box on the left hand side of the screen includes instructions to save or recall a program.

Note:

This screen is setup so that a user will not accidentally lose a program.

It intentionally requires several actions to perform a SAVE or RECALL.

UP ARROW

The UP ARROW key increments the *memory location* that will be used to store or recall a program.

The current *memory location* to use for store or recall is shown on the indicator/button located between the UP ARROW and the DOWN ARROW.

The allowable range of memory locations is 1..4.

The system will not allow the selection of out of range locations.

DOWN ARROW

The DOWN ARROW key increments the *memory location* that will be used to store or recall a program.

The current *memory location* to use for store or recall is shown on the indicator/button located between the UP ARROW and the DOWN ARROW.

The allowable range of memory locations is 1..4.

The system will not allow the selection of out of range locations.

MEMORY LOCATION

The MEMORY LOCATION button / indicator shows the current memory location that will be used to save or recall a program.

Pressing the MEMORY LOCATION BUTTON will call up an entry screen that can be used to key in a new memory location for saving or recalling a program. This is an alternative to using the UP ARROW and DOWN ARROW buttons.

The allowable range of memory locations is 1..4.

The system will not allow the selection of out of range locations.

SAVE PGM

The SAVE PGM button will save the contents of the CURRENT program in the memory location (1..4) shown on the MEMORY LOCATION indicator.

The previous contents of the memory location will be overwritten and can not be retrieved.

RECALL PGM

The RECALL PGM button will recall the contents of the memory location shown on the MEMORY LOCATION indicator into the CURRENT program.

The previous contents of the memory location will be overwritten and may not be retrieved.

BACK

The small blue BACK button in the lower right corner will return to the previous screen. Usually the previous screen is the MAIN MENU screen.

Screen Indicators

LAST SAVED/RECALLED

This indicator shows the number of the memory location used for the last save or recall operation.

This indicator is located on the lower left side of the screen, below the screen instructions.

PGM CHANGED

This indicator is visible when the CURRENT program is no longer the same as the memory location it was last saved to or recalled from.

This indicator is located below the RECALL PGM button.

MEMORY LOCATIONS

This combined indicator/button shows the MEMORY LOCATION to which the CURRENT program will be stored or from which a program will be recalled to the CURRENT program.

This indicator is located in the center of the screen, between the UP ARROW and the DOWN ARROW.

Pressing this indicator calls up a numeric entry screen that allows selecting the MEMORY LOCATION as an alternate to using the UP ARROW and DOWN ARROW.

VIEW/CHANGE FLAGS SCREEN

The VIEW/CHANGE FLAGS SCREEN may be used to view any FLAG or INPUT and change any FLAG or OUTPUT.

The screen is broken into two sections.

Upper Sections:

- The upper section allows viewing any of the FLAGS or INPUTs (1..99).
- The display shows FLAGS or INPUTs in groups of 10.
- UP and DOWN arrows allow a user to select the next or previous group of 10.
- INPUTs (and FLAGS) are shown as “1” when ON or SET and “0” when OFF or CLEARED.

Lower Section:

- The lower section allows viewing and setting of any of the FLAGS or OUTPUTs.
- The UP and DOWN arrows may be used to select a FLAG or OUTPUT to view or change.
- The select indicator (between the UP and DOWN arrows) is also a button that may be pressed to pick a FLAG or OUTPUT to view or change.
- The content indicator (between the SET and CLEAR buttons) shows the current state of the selected FLAG or OUTPUT.
- The SET button turns ON=1 the selected FLAG or OUTPUT.
- The CLR button turns OFF =0 the selected FLAG or OUTPUT.

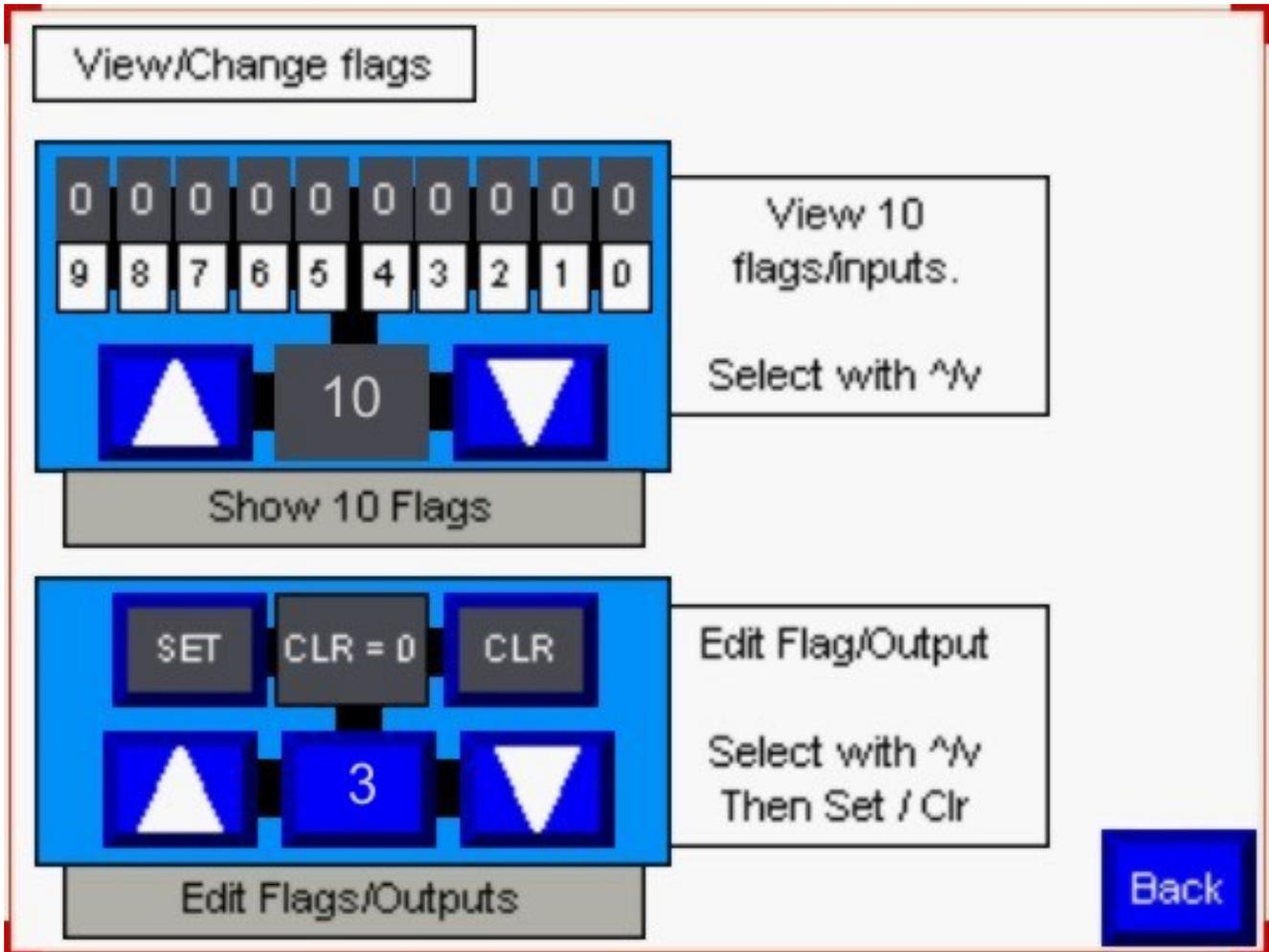
Notes:

- All FLAGS are volatile (are cleared at power-on and when a program is run) EXCEPT 13..16. Flags 13..16 remain as set through “program start” and “power OFF” of the system.
- FLAG 99 controls the system beeper. FLAG 99 = ON causes a continuous beep until turned OFF.
- All OUTPUTS are volatile (are cleared at power-on and when a program is run).

Note:

The lower section of the VIEW/CHANGE FLAGS SCREEN allows a user to view and edit any FLAG or OUTPUT.

It is not possible to edit an INPUT. (The RCT editor does not provide an equivalent to the PLC “force” functionality.)



Upper Section

Buttons

UP/DOWN Arrows

The ^UP and vDOWN arrows increment (UP arrow) and decrement (DOWN arrow) the number shown in the BASE indicator by 10. The BASE indicator is located between the UP and DOWN arrows.

The minimum is 0 and the maximum is 90.

The number on the indicator shows the “base” for the 10 FLAG / INPUT status indicators shown directly above.

If the BASE indicator shows “20” then the indicators above it show the state of the 10 FLAGS 20..29.

Upper Section Indicators

BASE

Located directly between the UP and DOWN buttons is the BASE indicator.

The BASE indicator shows which group of 10 FLAGS (or INPUTS) are being displayed in the indicators above.

INPUT STATE INDICATORS

The 10 INPUT STATE indicators show the current ON/OFF state of the 10 FLAGS or INPUTS selected by the BASE indicator.

Each of the black indicator windows shows the either “1” (=ON) or “0” (=OFF) depending on the current state of the particular FLAG or INPUT.

The white (unchanging) indicator windows below each of the INPUT STATE indicators show which FLAG or INPUT, within the group of 10 selected, is shown in the INPUT STATE indicator above.

Lower Section

Buttons

UP/DOWN Arrows

The UP / DOWN arrows in the lower section increment and decrement the number in the selection indicator between them.

The selection indicator determines which FLAG or OUTPUT the state indicator above it shows.

SELECTION INDICATOR BUTTON

The SELECTION INDICATOR BUTTON shows which FLAG or OUTPUT whose state has been selected to be viewed in the STATE INDICATOR located directly above the SELECTION INDICATOR BUTTON.

Pressing the SELECTION INDICATOR BUTTON brings up a keypad that allows the user to directly select the FLAG or OUTPUT being viewed in the state indicator above it. This number may also be changed by using the adjacent UP and DOWN arrow buttons.

Allowable inputs are 1..99. Numbers out of this range will be rejected.

SET Button

The SET button allows the user to force the flag or output state displayed in the FLAG / OUTPUT

indicator to be forced to a “1” (=ON=SET).

CLR Button

The CLR button allows the user to force the flag or output state displayed in the FLAG / OUTPUT indicator to be forced to a “0” (=OFF=CLear).

BACK Button

Pressing the blue BACK button in the lower right hand corner of the screen returns the user to the calling screen – whatever screen the user was on when the VIEW/CHANGE FLAGS button was pressed.

The calling screen may be the MAIN MENU, MANUAL CONTOLS, RUN SCREEN, UTILITY SCREEN,

Lower Section Indicators.

SELECTION INDICATOR BUTTON

This display on the SELECTION INDICATOR BUTTON shows which FLAG / OUTPUT is being displayed for editing on the FLAG / OUTPUT, directly indicator above.

FLAG / OUTPUT Indicator

The FLAG/OUTPUT indicator shows the current state “0” (=CLR = OFF) or “1” (=SET=ON) of the FLAG or OUTPUT selected by the SELECTION INDICATOR BUTTON.

Note:

The state of the FLAG or OUTPUT displayed can be edited using the adjacent SET and CLR (clear) buttons.

Enter Locations Screen

The Enter Locations Screen (accessed from the MAIN MENU screen) allows the user to enter the LOCATIONS (X-Y coordinate pairs) used by the GO_LOCN (A,ZZ) program command.

The GO_LOCN (A,ZZ) command instructs the RCT to move the trolley to the location defined by the X and Y coordinates stored in the ZZ (0..99) location.

The ENTER LOCATIONS SCREEN allows the user to either key in a pair of X and Y coordinates to be used as the ZZ location...

OR

The ENTER LOCATIONS SCREEN allows the user to store the current trolley X and Y position as the ZZ location coordinates.

The ENTER LOCATIONS SCREEN also allows the user to CLEAR (erase the contents) of ALL entered X,Y coordinate pairs stored in the LOCATION memory (0..99). After being CLEARed all LOCATIONS will be (X=0, Y=0).

Note:

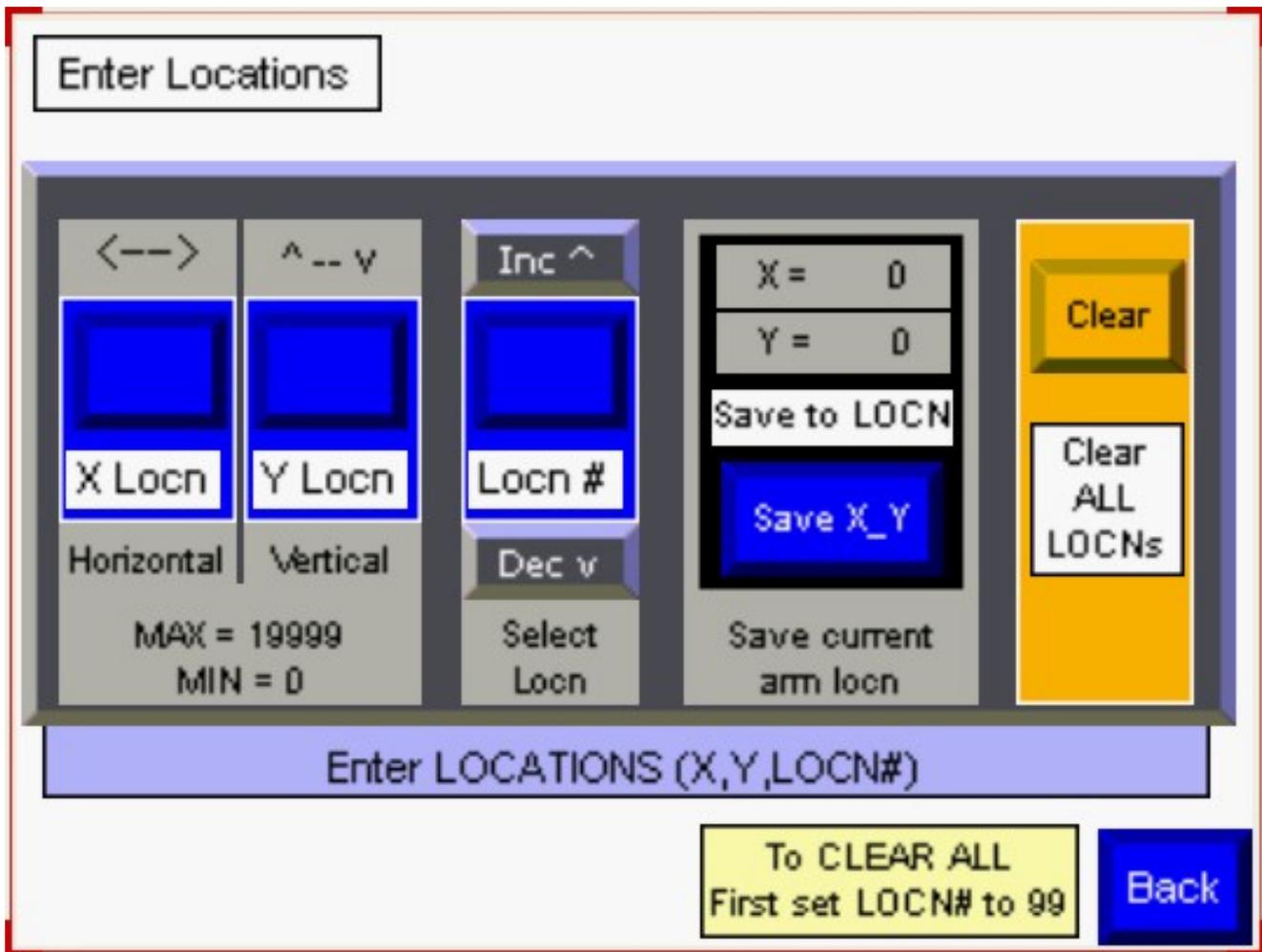
To avoid accidental erasure of the LOCATION data, the CLEAR button is *always* inhibited *except* when the LOCN# entry window shows "99". Only when the LOCN# window shows "99" will the CLEAR button be active.

The CLEAR button will be "GRAYed out" when it is not active (inhibited).

Note:

The "A" parameter in the GO_LOCN (A, ZZ) command specifies whether to move to the Y coordinate first and then the X coordinate (A=0) or the X coordinate first and then the Y coordinate (A= 1).

Which "A" is appropriate is determined by the path the trolley must take to avoid hitting the cleaning system. (I.e. not running into the side of a tank or other structure when moving from one LOCATION to another.)



Screen Buttons

LOCN#

The LOCN# button is a combined button indicator. The number on the face of the button shows which location is currently being shown/edited.

LOCN#s can range from 0..99 for a total of 100 user defined locations.

To change the LOCN# being displayed the user may:

- Press the LOCN# button. This will call up a numeric entry window with a keypad. Key in the desired LOCN#. Out of range entries will be rejected and the current value will remain unchanged. The range of entries is 0..99.
- Press the INC or DEC buttons located above (INC) and below (DEC) the LOCN# button/indicator. The user is prevented from using INC at 99 or DEC at 0.

When a new LOCN# is selected by either means described above, the X LOCN and Y LOCN button/displays will update to show the stored values for the current LOCN#.

Note:

When entering new X and Y values for a LOCN:

- Always begin by setting the LOCN# of the location.
- Once the LOCN# had been entered, the values of the X LOCN and the Y LOCN may be entered or edited.
- Alternatively, the SAVE_X_Y button may be used to store the current trolley location(X and Y) directly into the displayed LOCN#'s memory. There is no need to key in X and Y numbers when this entry method is used.
- When both the X LOCN and the Y LOCN have been entered the process is done – the entered X and Y numbers are already stored in memory and ready for use by the program. (There is no “STORE” button.)

X LOCN

The X LOCN button/indicator shows the current X coordinate of the LOCATION shown in the LOCN# button/indicator.

Pressing the X LOCN button/indicator will allow the user to enter a new value for the X coordinate of the current LOCATION (LOCN#). The allowable range is 0..19999.

Out of range entries will be rejected.

Note:

If the Save_X_Y button is pressed to save the current trolley location into the LOCN# location, the X LOCN and Y LOCN button/indicators will update to show the coordinates of the current location of the trolley.

Y LOCN

The Y LOCN button/indicator shows the current Y coordinate of the LOCATION shown in the LOCN# button/indicator.

Pressing the Y LOCN button/indicator will allow the user to enter a new value for the Y coordinate of the current LOCATION (LOCN#). The allowable range is 0..19999.

Out of range entries will be rejected.

Note:

If the Save_X_Y button is pressed to save the current trolley location into the LOCN# location, the X LOCN and Y LOCN button/indicators will update to show the coordinates of the current location of the trolley.

CLEAR

The CLEAR button clears ALL (0..99) stored locations to X=0, Y=0.

The CLEAR button normally shows gray indicating that it is not available for use. Pressing the CLEAR button when it is gray will produce no response.

The CLEAR button only works when the LOCATION is set to 99. There is a note at the bottom of the screen as a reminder that the user can only clear all locations by setting the LOCN# to 99 before pressing the CLEAR button.

Note:

Pressing the CLEAR button erases the contents of ALL 100 location memories.

Once memory has been CLEARed there is NO way to recover the previous contents of the LOCATION memories.

BACK

The blue BACK button in the lower right hand corner of the screen will return the screen from which the ENTER LOCATIONS screen was called. Usually the MAIN screen.

Screen Indicators

X LOCN

The X LOCN indicator/button shows the X value associated with the current LOCN# (location).

Y LOCN

The Y LOCN indicator/button shows the Y value associated with the current LOCN# (location).

LOCN #

The LOCN # indicator/button shows the current LOCATION (0..99). The contents of the indicated LOCATION are shown in the X LOCN and Y LOCN indicator/buttons.

X =

Y =

The X=? and Y=? indicators located above the Save_X_Y button show the current location of the trolley.

Pressing the SAVE_X_Y button will transfer these X and Y values into the location currently selected by the LOCN # control.

Note:

The purpose of the Save_X_Y control is to allow a user to:

- Jog the trolley into a specific position.
- Save the X and Y coordinates of that position into a LOCATION without having to transcribe the numbers into the X LOCN and Y LOCN entry windows.

This may be an easier and less error pron way to get positions into a program.

Clear Program Screen

The Clear Program Screen is used to irretrievably erase the contents of the CURRENT program.

The CURRENT program is the program that may be RUN, edited, saved to memory, and recalled from memory.

The purpose of having a separate screen for the single Clear Program button is to make it less likely that a user will inadvertently erase a program.

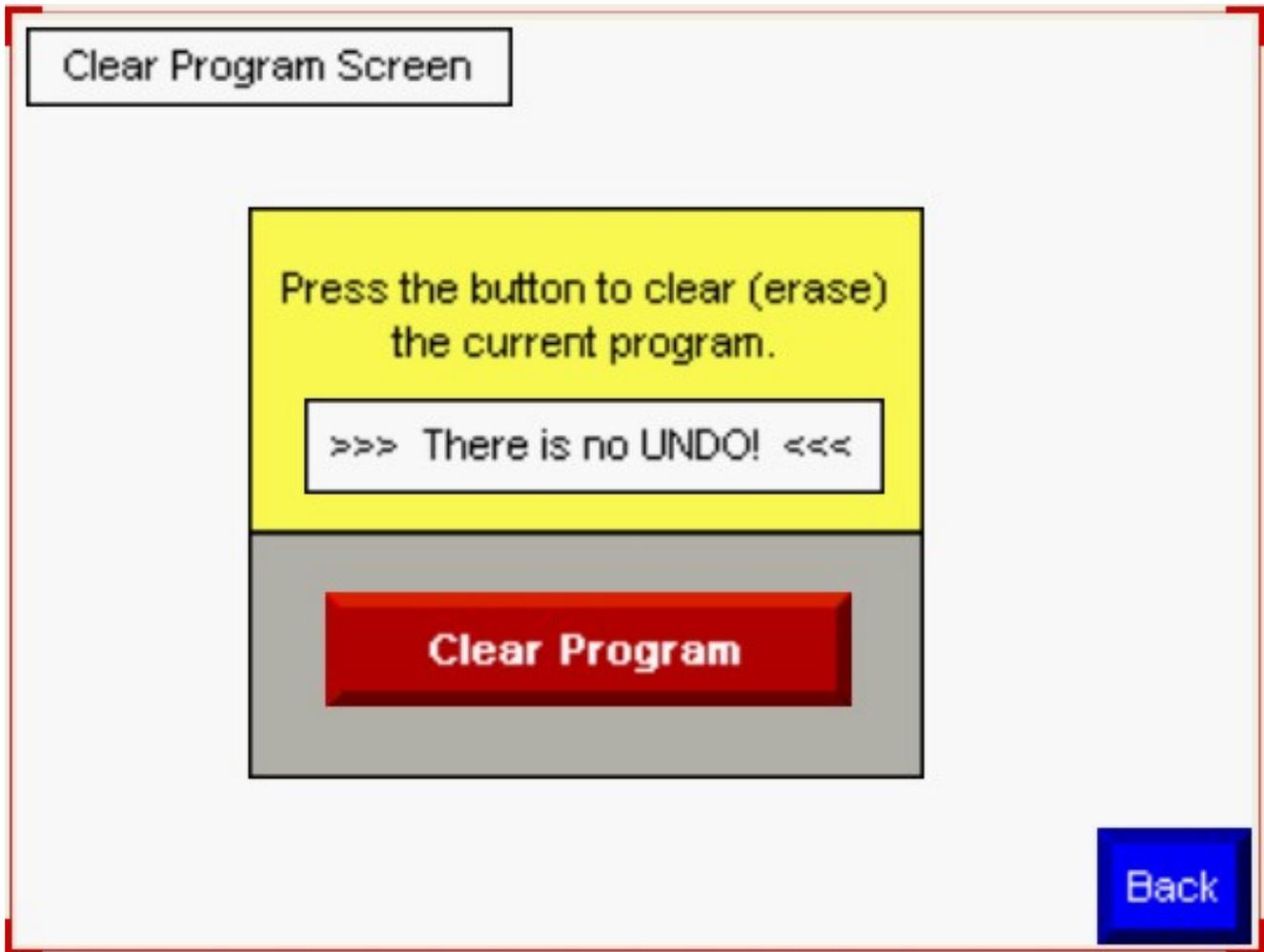
The process of CLEARing a program usually begins on the EDIT PGM SCREEN when the user presses the red “Clr Pgm” button.

Instead of immediately clearing the CURRENT program, the button takes the user to the “Clear Program” Screen.

On the Clear Program Screen there is:

- A note that says, “Press the button to clear (erase) the current program”.
- A flashing warning that, “THERE IS NO UNDO”.
- A red button labeled, “CLEAR PROGRAM”.

Pressing the CLEAR PROGRAM button will clear the CURRENT program.



Screen Buttons

Clear Program

Pressing the RED “Clear program” button will irretrievably clear (erase) the CURRENT program.

While the button is pressed, it will change from RED to WHITE to indicate the button press has been accepted.

The contents of the CURRENT program will be immediately erased.

There is no way the user can UN-ERASE a program.

BACK

Pressing the blue “Back” button will return the user to the whatever screen called the Clear Program screen.

The most common screens to return to will be the MAIN SCREEN and the ENTER PGM SCREEN.

Change Password Screen

The Change Password Screen allows users with passwords to change their password.

When the system is delivered there are two user level passwords in place. One for OPERATORS and one for SUPERVISOR / MAINTENANCE.

If neither is logged in, the system reverts to its default operating mode. In the default operating mode a user is restricted to running an existing RCT program. All other functionality requires the entry of a valid USERNAME and associated PASSWORD.

A user who knows a current password may use the Change Password Screen to change to another password.

Passwords provide access to the different screens.

A user with no assigned user name and password is considered a DEFAULT user and is generally restricted to the most basic operations.

A user with an OPERATOR password can reach the screens needed to operate but not program the RCT.

A user with SUPER (supervisory) password can reach all screens.

The selection of screen access may be modified by the factory at the time of purchase. A purchase may select password names and which screens they access. Multiple user names and passwords may be installed by the factory.

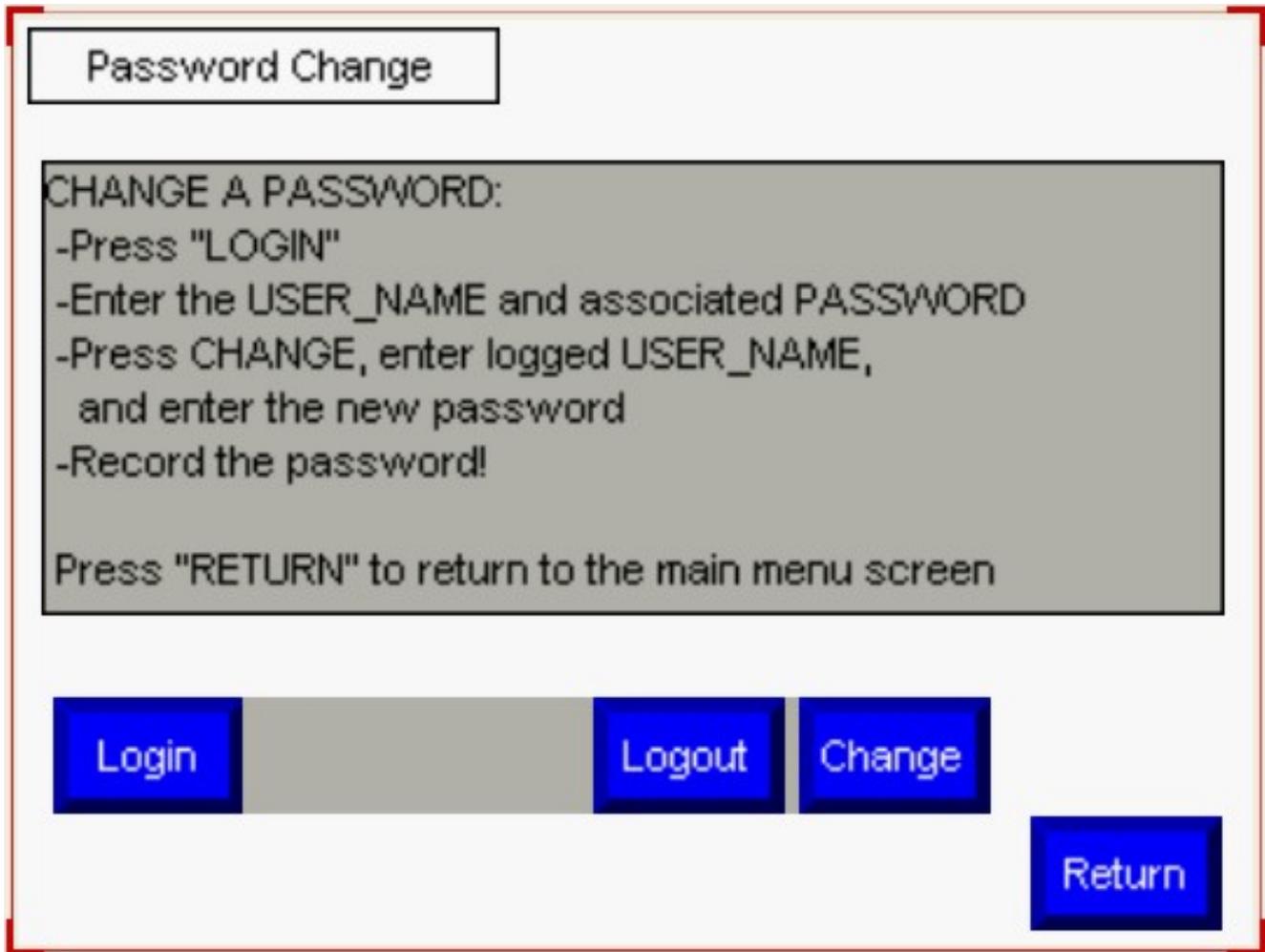
Note:

All USERNAMES and PASSWORDS are case sensitive. "ABCD" is not the same as "abcd".

Note:

Field editing the system to change user names and privilege is possible with an Ethernet equipped Windows PC. But it is not recommended.

The basic instructions for changing a password appear on the Change Password Screen.



Screen Buttons

LOGIN

The LOGIN button calls up a password entry window that may be used to enter a USERNAME and PASSWORD. Once a matching USERNAME and PASSWORD have been entered a user may choose to CHANGE the password or LOGOUT.

It is not necessary to for the user to LOGIN if they have already logged in on another page unless they wish to LOGIN with a different identity.

Only one USER may be logged in at a time.

Note:

A logged-in supervisor may change any level of password.

Note:

All USERNAMES and PASSWORDS are case sensitive. "ABCD" is not the same as "abcd".

LOGOUT

Pressing the LOGOUT button requests that the system log out any user who has logged in.

There is no need to re-enter the USERNAME or PASSWORD to logout. Simply pressing the LOGOUT button will cancel the most recent LOGIN.

Use LOGOUT to cancel a LOGIN when the operator leaves the area to prevent unauthorized access.

CHANGE

The CHANGE button permits a user to change their own password.

Passwords may be up to 16 characters long and may contain letters and numbers.

Passwords are case sensitive. "CDEF" is not the same as "cdef".

To change the OPERATOR password:

- Enter the screen *having already logged in* as SUPER. (MUST be logged in).
- Press CHANGE.
- Enter "OPERATOR" in the "User" window
- Use the TAB key to move to the "new password" entry window.
- Enter the new password in the "New password" window. Complete the entry with the "ENTER" key.
- Use the TAB key to move to the "confirm password" window.
- Re-enter the new password in the "confirm password" window. Complete the entry with the ENTER key.
- Optionally: Log out.

To change the SUPER password, follow the same procedure only key "SUPER" into the "User" window.

Note:

A logged-in supervisor may change any level of password.

Screen Indicators

Logged In Indicator

The gray space between the LOGIN and LOGOUT buttons is an indicator that shows the USERNAME of the currently logged in user.

If there is no currently logged in user the indicator will be blank showing a blank, gray background between the LOGIN and LOGOUT buttons

UTILITY SCREEN

The UTILITY SCREEN provides access to the various utility functions.

IMPORTANT

Functions that are accessible on the UTILITY SCREEN can impact system safety.

The utility screen provides access to diagnostic features that allow a user to operate the system in ways that might be unsafe or confusing for normal operation.

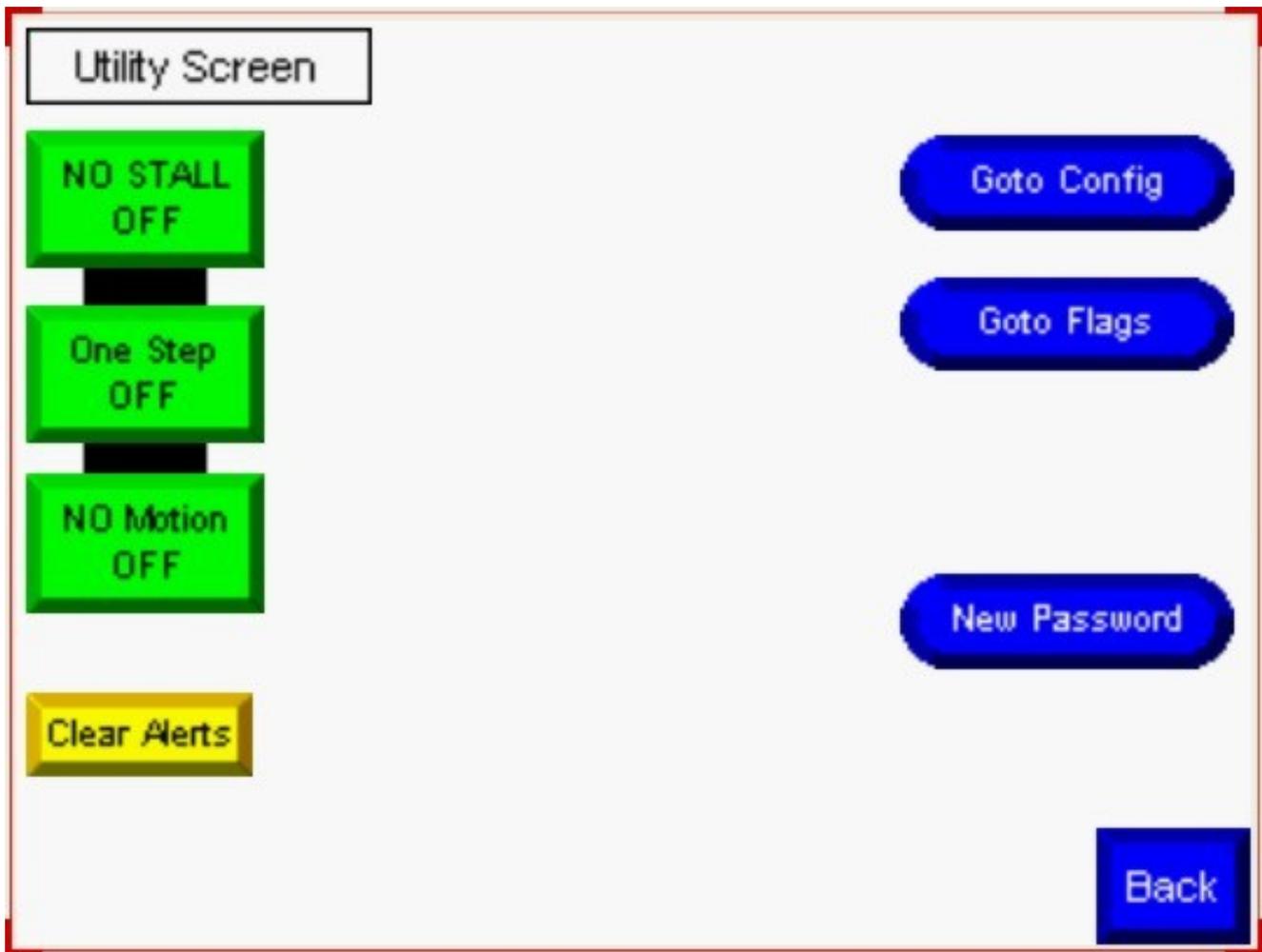
IMPORTANT

A system should never be left unattended when UTILITY SCREEN diagnostic functions are enabled.

Operators unfamiliar with the diagnostic functions should never have access to the machine when these functions are active.

The UTILITY SCREEN provides access to:

- NO-STALL Mode. When this diagnostic mode is active, the stall detect function for RCT motion is disabled. The RCT will ***not*** take note that its motion is being blocked by an obstacle such as a person, basket, or tank wall. Injury and damage may occur if this is used incorrectly.
- ONE STEP Mode. This diagnostic mode causes a RCT user program, when RUN, to execute one single step in the program and then wait for the operator. When the operator presses the “S-Step Ack” button the program will execute its next instruction and then wait...
- NO MOTION Mode. In this mode, a user's program may be run without any motion of the RCT trolley. All program commands that would usually result in motion (like GO_X) are skipped.
- CLEAR ALERTS. Used when system maintenance may result in a lack of communications between the PLC and the OIT resulting in on-screen “alerts”.
- Goto Config. This provides the required access to the OIT hardware configuration screen. Access to this screen is NOT recommended. Access is required by the OIT for OIT setup and maintenance and should NOT be required by users.
- GOTO Flags. Go to the VIEW / CHANGE FLAGS Screen.
- New Password. Permits the supervisor (SUPER password holder) to change any passwords used by the system to restrict access.



Screen Buttons

No-Stall OFF

The NO STALL button toggles to enable/disable the NO STALL mode.

The NO STALL mode is only intended for use by authorized maintenance personnel who are attempting to diagnose a problem involving the RCT stall detect system. THERE IS NO OTHER VALID USE FOR THIS FEATURE.

IMPORTANT

The NO STALL mode is potentially dangerous to the user and the system.

The NO STALL mode defeats the “stall detect” system that detects when the motion of the RCT trolley has been blocked and then stops all RCT motion.

Using the system for normal operation in NO STALL mode is potentially hazardous to operators, materials being processed, and the RCT itself.

Pressing the NO STALL button will cause the button to change from GREEN to RED and the word

OFF will change to ON.

Pressing the NO STALL button again will cause the button to change from RED to GREEN and the word ON will change to OFF.

There is also a NO STALL flag that will appear in the “STATUS” indicator windows that is displayed on the RUN SCREEN and the MANUAL CONTROL SCREEN. When NO STALL is active an amber “NO Stall” indicator becomes active on each of these screens.

When NO STALL mode is active the system will no longer detect when its motion is being blocked and will no longer shut down motion when a STALL is detected.

The NO STALL mode is only intended for use by authorized maintenance personnel who are attempting to diagnose a problem involving the RCT stall detect system. There is NO OTHER VALID use for this feature.

ONE STEP

Pressing the ONE STEP button turns ON or OFF the ONE STEP MODE.

Pressing the button changes the GREEN button to RED and changes the OFF legend to ON.

Pressing the button again changes the RED button to GREEN and changes the ON legend to OFF.

(The button “toggles” OFF → ON → OFF → ON →)

When the ONE STEP mode is active an orange “One Step” indicator flag appears on the RUN SCREEN.

The ONE STEP mode is used for developing, testing, and debugging user programs.

When ONE STEP mode is ON and a program is run, the RCT will execute one program instruction and stop. The RCT will not move on to the next instruction in the user program until the user presses the “S-Step Ack” button that appears on the RUN SCREEN.

In this way, a user may see what each step in the user program causes the system to do. Often this makes it easier to find problems in newly written programs.

The ONE STEP mode may be used with the NO MOTION mode for program debugging.

Note:

While it is not dangerous to leave the system in ONE STEP mode when testing is done, it may confuse an operator who is not familiar with the mode.

NO MOTION

The NO MOTION mode button toggles the NO MOTION operating mode.

Pressing the NO MOTION button will change the button from GREEN to RED and the OFF indicator will be replaced with the ON indicator.

Pressing the NO MOTION button again will change the button from RED to GREEN and the ON indicator will be replaced with the OFF indicator.

(The button “toggles” OFF → ON → OFF → ON →)

Orange “No Motion” indicators on the MANUAL and RUN screens appear when the NO MOTION mode is active to avoid confusion.

When the NO MOTION mode is active the RCT will not cause any motion of the trolley – either in automatic (program controlled) or manual (jog) modes. All motion related instructions (GO_X, GO_Y, HOME, AGITATE, and GO_LOCN) will not be executed when programs are run.

The intent of the NO MOTION mode is to allow faster testing and debugging of programs. The user need not wait for time consuming trolley moves when a program is being run. Once the user is confident that all “locations” in the program are correct there is no further reason to wait for actual motion when testing a program.

The NO MOTION mode may be used in conjunction with the ONE STEP mode.

Note:

While it is not dangerous to leave the system in ONE STEP mode when testing is done, it may confuse an operator who is not familiar with the mode.

GOTO CONFIG

The GOTO CONFIG button provides direct access to the OIT (display screen) setup parameters.

IMPORTANT

Users should NOT have to access the OIT configuration screen except when the hardware is being changed.

Changing OIT setup parameters is never necessary at any other time.

Changing OIT setup parameters can result in a RCT that cannot be used. And it may be difficult to change parameters back once improperly changed.

Once the user exits the CONFIG menu he will be returned to the UTILITY SCREEN.

GOTO FLAGS

The GOTO FLAGS button provides access to the VIEW / CHANGE FLAGS screen.

The GOTO FLAGS button turns OFF the current UTILITY screen and turns ON the VIEW / CHANGE FLAGS screen.

Once in the VIEW / CHANGE FLAGS screen a user may view FLAGS and INPUTS or change FLAGS and OUTPUTS.

The VIEW / CHANGE FLAGS screen is useful during program debugging. The screen is accessible during normal and SINGLE STEP operation.

The BACK button on the VIEW / CHANGE FLAGS screen returns to the calling screen. In this case, the UTILITY SCREEN.