ELEVATOR PROJECT - USEFUL INFORMATION

Program Structure

- Break your program into meaningful sections, each one handling a particular job.
- Use Symbols in your program to represent **ALL** variables.
- Symbols are provided in the global Symbol table for all the PLC inputs and outputs. See the PLC IO sheet for more details.

Running the Elevator

Control Flags. *(VERY IMPORTANT)*

- **SYSTEM_ENABLE** must be set to **ON** for the physical elevator to respond to any commands.
- **SIM_ELE_ENABLE** must be set to **OFF** while using the physical elevator, or the simulated elevator will cause conflicts.

Motor Control

- **WINCH_MOTOR** is the symbol name used to set the speed of the elevator carriage. Use the **MOV** instruction to write to this location (e.g. MOV #94C WINCH_MOTOR will set the motor speed to #94C)
- The range of motor speeds are 0 (#0000) to 6000 (#1770). Make sure that your motor speed output cannot exceed this range.
- 0 (#0000) is the maximum speed **UP**. 6000 (#1770) is the maximum speed **DOWN**.
- **_HOLD_VALUE** is the motor speed value where the elevator carriage is stopped.
- Each elevator has a different hold value:

ELEVATOR	_HOLD_VALUE (Dec)	(Hex)
1	2931	#0B73
2	2886	#0B46
3	2921	#0B69
4	3189	#0C75

- Using the incorrect hold value will result in the elevator moving when your program expects it to be stopped.

Floors and Doors

- The elevator **must** be stopped on a floor before the doors will open.
- The DOOR_OPEN or DOOR_SHUT signals must be pulsed more than one PLC cycle (i.e. do not use differentiation) to activate the door movement sequence.
- The doors will not stop halfway through a cycle. Once the doors start moving, they will finish opening or closing, regardless of the control signal state.
- LIMIT_OPEN and LIMIT_SHUT are used to signify the completion of the door cycle, and the current door state (open or shut).
- Each floor has an individual limit switch. Use these to determine if the carriage is at a floor.

Encoder Instructions

- Use the PRV instruction for reading the encoder value. The encoder is using High Speed Counter 0 on all the elevators.
- The INI instruction can be used to set the encoder value to an arbitrary value.

Programming Issues (Data types, Memory and Conflicts)

- If an output is set in two places in your program, then the latter rung will overwrite the earlier rung.

- The encoder count value is 2 words (32 bits). `LONG` type instructions must be used when dealing with encoder data. Examples instructions are for moves (`MOVL`), maths (`+L`, `-L`, `*L`, `/L`), and comparisons (`>L`, `=L`, `<=L`, `<>L`, etc.)
- When using double or long variables (2 or 4 words), make sure your program is not using conflicting memory. For example, a `DINT` referenced at `W4` and an `INT` at `W5` are both using address `W5`. A `DINT` uses two words, in this case both `W4` and `W5`.
- The PLC compiler will not stop you using bits or integers that overlap (e.g. the bit `W0.06` and the integer `W0` are both using the same bit `W0.06`, and will interfere with each other).
- The Address Reference Tool (Alt + 4) can be used to search for conflicting variable memory usage.
- Make sure you check the Instruction Reference for each instruction you use. Some instructions, such as division (`/` & `/L`), have outputs than might be different than you expect.
- Do **not** put loops in your function blocks. Loops may never exit, so no other code will run.