ENMT211 ELEVATOR PROJECT - USEFUL INFORMATION

Control Flags

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

Program Structure

- Break your program into meaningful sections, each one handling a particular job.
- Use Symbols in your program to represent **ALL** variables.
- Counters and Timers do not need symbols.
- Symbols are provided in the global Symbol table for all the PLC inputs and outputs.

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 motor speed is limited to between 0 (`#0000`) and 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 carriage is stopped.
- Each elevator has a different hold value, printed on the side of the elevator.
- If the elevator is moving when your program expects it to be stopped, you're probably using the incorrect hold value.

Floors and Doors

- The elevator **must** be stopped on a floor before the doors will open.
- Every floor has a limit switch. Use these to determine if the carriage is at a floor.
- 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).

Encoder Instructions

- The encoder uses High Speed Counter 0.
- The correct settings are used in the template file.
- Use the `PRV` instruction for reading the encoder value.
- The `INI` instruction can be used to set the encoder value to an arbitrary value.

Programming Issues

- 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.