**OOFB: outside of function block**

How the scheduler works:

* Made up of five main states
  + Idle
  + Up
  + Down
  + Selective Up
  + Selective Down
* After initialising, begins on floor 5 in the idle state
* Utilises program reading top-to-bottom for destination selection

**Idle**

* Has five primary rungs within its own function block
  + One for cases on each floor, all five are similar but different for a reason
* Each rung triggers when the carriage is “idle” (not moving, at a floor, doors are shut)
* Checks from the bottom floor up for a floor bit that is “on”
* Each check is paired with a NC contact for if it isn’t on (basically heaps of if-elses) to move to the next, so that it breaks out of the check once one is found, therefore multiple floors can’t be selected.
* Once a floor is found, one of five things will happen:
  + If the button is for the floor itself, the destination will be re-set to that floor and the elevator **remains in idle state**
  + If the button is above the floor:
    - If an internal or up button, destination is set and elevator **moves to** **up state**
    - If a down button, destination is set and elevator **moves to selective down state**
  + If the button is below the floor:
    - If an internal or down button, destination is set and elevator **moves to down state**
    - If an up button, destination is set and elevator **moves to selective up state**

**Up**

* Two main sections
  + Between floors
  + At a floor
* Utilises program reading top-to-bottom for destination selection

Between floors:

* Four rungs, for each different case of being “between floors”
* When a button is pressed that applies to the current state (i.e. is above the current position, and between the position and the destination) then the destination is switched to be the closer floor

At a floor:

* OOFB acceleration control will stop the carriage at a floor
* When at the floor, trigger a bit to start the OOFB door control, which when completed will trigger a timer for 3 seconds to allow time for the doors to be reopened if needed
* After the 3 seconds, depending on the floor, checks will be done to see if there are any more applicable floors to go to:
  + This runs top-floor-down, as being at the top floor means the elevator automatically **moves to idle state**
  + If not, elevator **moves to idle state**
  + If yes, elevator **remains in up state**

**Down**

* Much the same as up, still uses program reading top-to-bottom
* Between floors exactly the same except checking for closer floors down from the position
* Same at-floor process except next-floor check runs bottom-floor-up instead of top-floor-down as being at the bottom floor will mean elevator automatically **moves to idle state**
* If next floor found, elevator **remains in down state**

**Selective Up**

* This is where shit gets funky
* If a floor is selected that is *below the current floor but wants to go up*, then the elevator will *only go to that floor* before selecting any other destinations
* Essentially, the elevator is an “up elevator” the whole time despite moving downwards, it is just getting to its starting point
* Once the elevator arrives at its destination, it **moves to the up state**

**Selective Down**

* The same thing but triggers if a floor is selected that is *above the current floor but wants to go down*
* The elevator will only go to that floor before selecting new destinations, any other buttons pressed while it’s moving will not affect the destination
* Once the elevator arrives at its destination, it **moves to the down state**