A Simple Air Conditioning System

The HVAC systems is presumed to consist of three computational elements in addition to the mechanical and electrical elements. These three computational elements are: the HVAC controller proper, which controls the actuators and responds to the sensors; the human interaction system, which includes the buttons and display; and finally the HVAC control settings element, which manages the goal variables used by the other two elements. It is the control settings element which the focus of this project.

Abbreviations

HVAC-CON : the HVAC controller element

HID : the human input and display element

HCS : the HVAC control settings element

Assumptions:

Only one external event is processed at a time.

Asynchronous processing is assumed. All internal transitions which enable must have fired before any new external events will be processed.

Functional Specification:

There is ambiguity in what is to be done with the FanUp (FanDown) events regarding the manual-mode and the fan-speed-current variable. When in a non-manual mode does the event only cause a change to manual-mode or does it also cause a fan-speed-current change? I have decided that the initial event should not increment (decrement) the fan-speed-current but restore the previous manually entered value. This would produce a reminder to the operator before updating the fan-speed-current. A pair of additional variable is introduced to store the manually entered temperature and fan speed, manual-temp and manual-fan.

A further ambiguity is the factory initial value for manually entered goal-temp and fan-speed-current. I choose to use AUTO-F for the default manual-fan and MAXT for the default manual-temp.

The provided specifications are revised and restated equivalently as follows:

Power

The HID produces power-press events which toggles the HCS between power-on and power-off states

While in the power-off states all events are ignored except the power-press event

During power-off all variables and their values are maintained; in particular the goal-temp and fan-speed-current variables are maintained for use by the HID and the HVAC-CON

Input Mode

There are three mutually exclusive input modes: manual, auto and turbo

The auto-mode may only be entered by an auto-mode event from the HID

The turbo-mode may only be entered by a turbo-mode event from the HID

The manual-mode may be entered by either a FanUp or FanDown event from the HID

The factory default is auto-mode

The entering power-on results in returning to the most recently active mode.

Temperature (goal-temp)

The HID produces TempUp and TempDown events

The HCS maintains a goal-temperature (goal-temp) which is used by the HVAC-CON and the HID.

The goal-temp is an integer (as are all associated variables and constants)

The goal-temp must be between MINT and MAXT (MINT < goal-temp < MAXT)

MINT and MAXT are constants.

In manual-mode: the TempUp event results in the manual-temp incrementing by 1 degree C and the goal-temp being set to it.

In either manual-mode or turbo-mode, the TempDown event results in the manual-temp decrementing by 1 degree C and the goal-temp being set to it.

The turbo-mode event results in the goal-temp being set to MAXT

Upon entering the manual-mode the goal-temp is restored to manual-temp

Fan Speed (fan-speed-current)

The HID produces FanUp and FanDown events.

The HCS maintains an air flow speed via a fan in a current fan speed (fan-speed-current) variable

which is used by the HVAC-CON and the HID.

The fan-speed-current is an integer (as are all associated variables and constants)

The fan-speed-current must be between MINF and MAXF (MINF < fan-speed-current < MAXF)

MINF and MAXF are constants.

The FanUp event results in the fan-speed-current incrementing by 1 unit, only if in manual-mode

The FanDown event results in the fan-speed-current decrementing by 1 unit, only if in manual-mode

The auto-mode event results in the fan-speed-current being set to AUTO-F

AUTO-F is a constant between MINF and MAXF (MINF <= AUTO-F <= MAXF)

The turbo-mode event results in the fan-speed-current being set to MAXF

Upon entering the manual-mode the fan-speed-current is restored to manual-fan

Design Approach

There are many ways to approach the model for this system. It could be modeled as a flat FSM resulting in a model that is easier to argue correctness. I have chosen to build a model that reflects the re-factored specifications as outlined above, this bias introduces an AND state over the fan, temperature and input.

The model is robust if there is no non-determinism, this can be done by incorporating priority, and similar techniques. In this project model robustness is achieved by making strong restrictions. First are the assumptions of single event super-steps, exhausting all internal events before receiving a new external event and no time dependency (i.e. no timeout or schedule functions). Next, prevent variable and event conflict by simple argument, primarily by avoidance.

In order to avoid a variable conflict, it is sufficient that it is not be possible to update a variable from more than one transition. This can be accomplished by only updating a variable on a single transition. It is also sufficient if the variable may be updated on a set of exclusive transitions, that is, the transitions emanate from the same state.

In order to void event conflict, an event should only trigger one transition. If an event triggers more than one transition there must be strong arguments that the two transitions can not be in conflict.

The DisplayFan and DisplayTemp functions are imitated by a pair of Simulink Display objects. The buttons are each imitated by a manual switch and a pair of constants. This produces an edge event which is interpreted as indexed events.