Assignment 1: Finite State Machines

BACKGROUND

A Finite State Machine (FSM) represents the design of a system as a graph, with "state" (nodes) and "transitions" (directed edges). Transitions between states are triggered by external event such as an input to the system, or an internal event such as timer expiring. Each edge is labelled with the trigger of the state transition that the edge represents. Read the instructions for Laboratory 1 and the course slides to get more information on FSMs. You can debug it effectively based on the design, and further even make the computer code *look like* the design, making it easier to reason about and extend. I recommend using Moore FSMs in which the output of the system is only a function of the current state with input such as buttons and sensor values determining when the machine changes state. An alternative is to use so-called Mealy FSMs in which the output is a function of both the input and current state. Some software systems can be described more compactly as Mealy FSMs but the diagram is much more complicated.

Below we will describe a simple elevator controller subsystem. Thinking about the possible states the elevator might be in, we need to consider its location, the status of the call buttons on the floors, and the floor selection buttons inside the elevator, along the following lines:

Location of elevator

- location = {Ground Floor, First Floor, Moving}

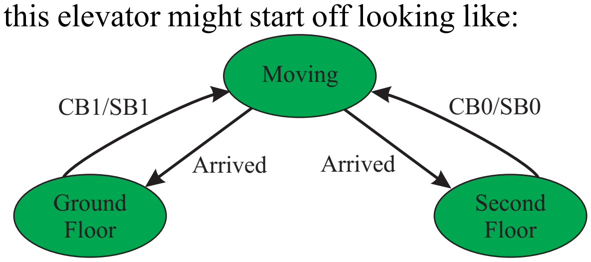
On each floor, 0 and 1, there are call buttons with lights

- call buttons: CB0, CB1 = {true, false}
- call lights: CL0, CL1 = {on, off}

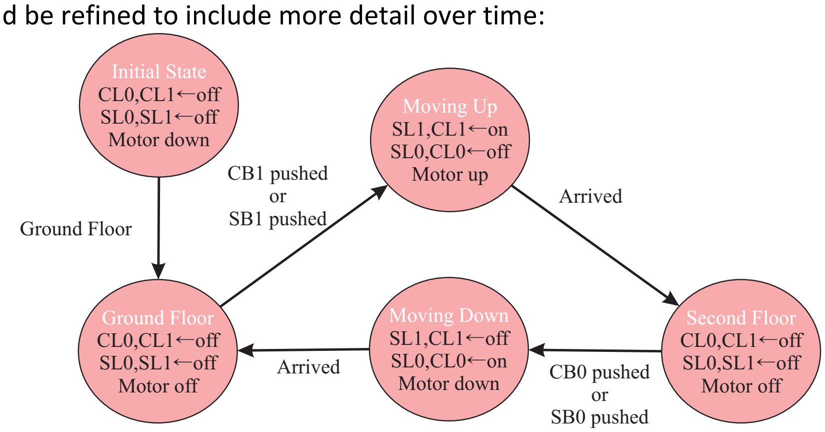
Inside the elevator, there are selection buttons with lights

- selection buttons: SB0, SB1 = {true, false}
- selection lights: SL0, SL1 = {on, off}

A really simple FSM for this elevator might start off looking like:



The diagram could be refined to include more detail over time:



PRE-LAB 1: A TRAFFIC LIGHT

In this assignment, you will work through an example FSM for traffic lights at an intersection. As you will see, describing the behavior of a control system, such as a streetlight controller, as a FSM is fairly simple. This description ensures that you have described the behavior of the system in all possible states. It is easier to see if the control policy of some states is unspecified than if just natural language was used to describe the system. Software, Computer, and Electrical engineers use an extension of FSMs to describe software controllers using a set of graphical conventions known as the Unified Modeling Language to describe the behaviour of the software objects within the systems that they design (http://en.wikipedia.org/wiki/Unified_Modeling_Language).

The traffic light model follows the pattern used by most lights in Victoria, where the lights cycle through green, yellow and red. The system will progress through sequences of light changes. Assume that in our current state, the first direction's light is green and the second direction's light is red. If a sensor on the street for the second direction detects incoming traffic, it transmits a signal to the light controller. In response, the controller will change the light in the main direction to yellow, provided the green light has been on for a sufficient duration. The controller will wait for any cars within the intersection to move out of the intersection and then change the first direction light to red and the second direction's light to green.

WHAT YOU NEED TO DO FOR THE PRELAB:

In your first lab you will be picking up your VEX kit. Fire it up and check it out!

1. Read the RobotC help files (hhttp://www.robotc.net/support/vexrobotics/) to find out how to install RobotC on your VEX controller and access basic RobotC functionality.
2. Generate a FSM diagram that described the behaviour of the traffic light controlled by the code in the file "TrafficLight.c". It can be hand drawn, but should clearly capture the states and transitions within this implementation of a traffic light controller.

Bring this to your second lab. It can be a simple sketch, but put your name and student number on the top of the page, so that we can give you some lab credit for thinking about this in advance!