Garage Door State Machine

Zach Campanella - Emily Filmer
SWEN.564
3/29/2018

Introduction

Our garage door state machine project simulates, through hardware, the status of a garage door in terms of its position and components at any given moment in time. The system functions as an average garage door system, controlled by the user with a single remote control button. The expected functionality is observed as follows.
Button Press:

- If door is fully closed, the door will start opening.
- If door is fully open, the door will start closing.
- If in the process of opening or closing, the door will stop.
- If door is stopped, the door will start opening or closing, opposite of the direction it was previously moving.

The garage door controller system will receive a flag when the door has reached a full-open position, and a full-closed position. This trigger is simulated with the fullOpen/fullClosed events. The expected functionality is observed as follows:
Full-Open Reached/Full-Closed Reached:

- Door stops.

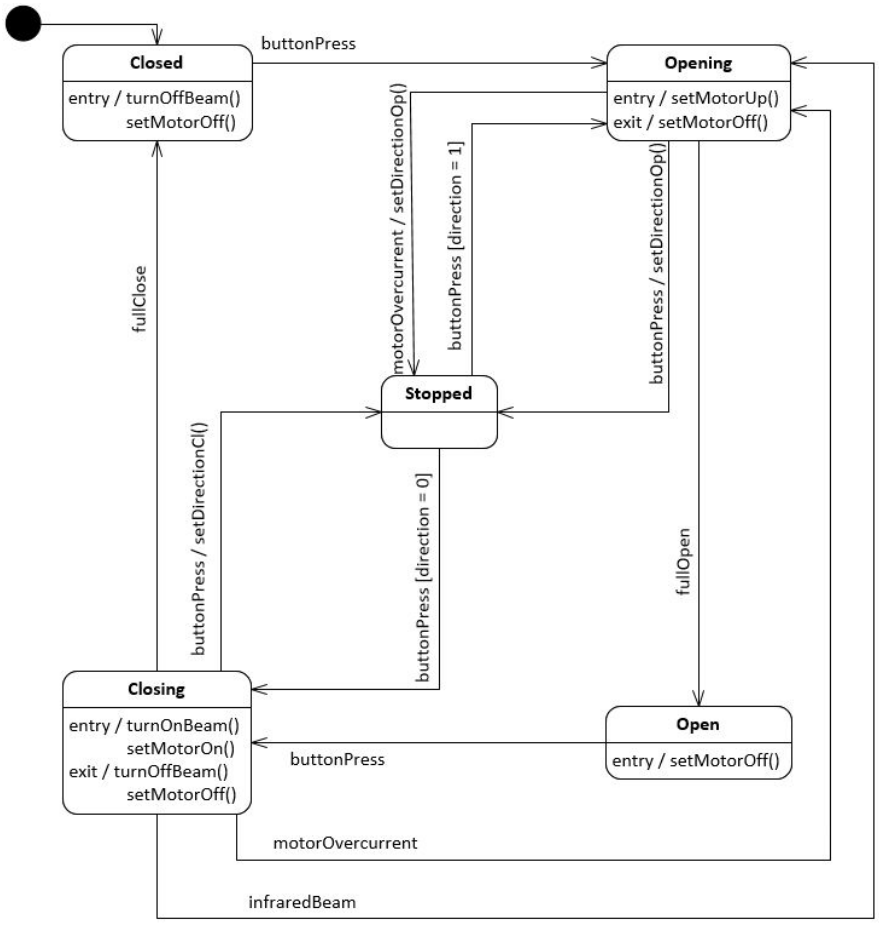
The garage door incorporates an infrared beam, which will act as another input to the garage door controller system. The expected functionality is observed as follows.
Infrared Beam Tripped:

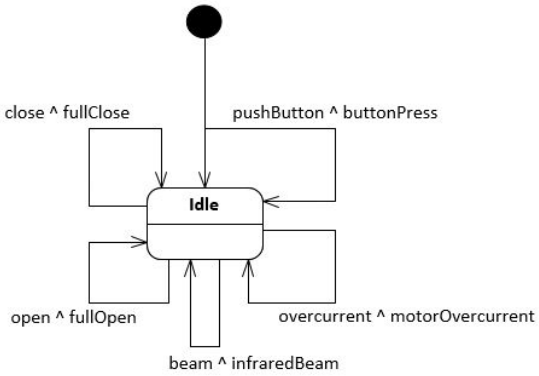
- If door is closing, the door will start opening.

The garage door also monitors that the motor is working properly, providing another input to the garage door controller system, the motor overcurrent flag. The expected functionality is observed as follows.
Motor Overcurrent:

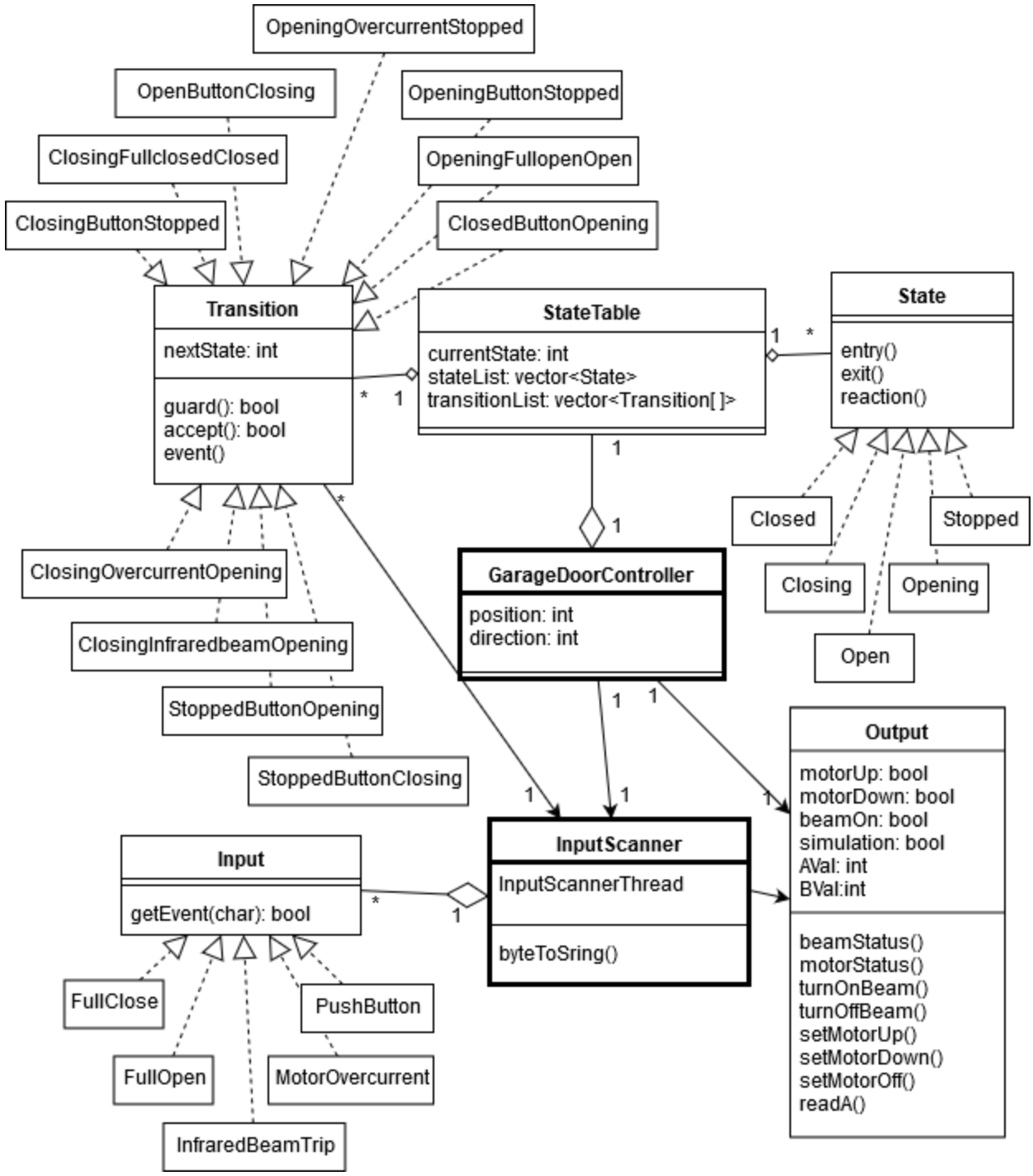
- If the door is closing, the door will start opening.
- If the door is opening, the door will stop.

State Machine





Class Diagram



Changes for Hardware Update

For the most part, the overall design of the system stayed the same. A few events and guards were updated, and Athena GPIO inputs/outputs were integrated. The relationship and interactions between classes were not necessary to change.

The hardware utilizes signals for the garage door being fully open or fully closed, as opposed to the simulation utilizing a time counter for determining if the door is fully open or closed. This change required updating the related Transitions. The guard() methods to check position before moving to state Closed from Closing and Open from Opening were removed for hardware mode. By default they return True when called. This guard was replaced with an Event requirement. This is reflected in the implementation of the accept() method. In hardware mode, the accept() method checks that the FullOpen/FullClosed event has been triggered.

The Output class contains internal methods (beamStatus and motorStatus) to produce the system's output. These methods contain two conditional behaviors, one for the simulation (printing output message on screen) if simulation==True, and another for the hardware implementation if simulation==False. The hardware implementation sets the GPIO pins high or low for output, instead of printing on screen messages. The system receives input on the DIO A pins and outputs on the DIO B pins.

Debouncing was added to the InputScanner class to protect the system from being overloaded from a bouncing input. Once an input is received, the InputScanner will not accept another input of the same type for the next 300 milliseconds. It was determined through testing the system that 300 milliseconds was long enough to prevent extra input but short enough to allow a user to operate the system unimpeded.

Discussion of Hardware Issues

There were a couple minor issues with the development the garage door opener. The largest issue was the bouncing input from the FPGA. The buttons on the FPGA aren't perfect, so when a button is pressed it usually outputs multiple times before a person has an opportunity to release their finger from the button. Debouncing was implemented to nullify this issue. The other issue for development was when a program would continue running on the Athena board despite appearing to have been stopped. These programs would attempt to output from the board along with the program that was attempting to be tested. This caused some issues with debugging until a power cycle on the Athena board was performed.