**Lab 2: Finite State Machine**

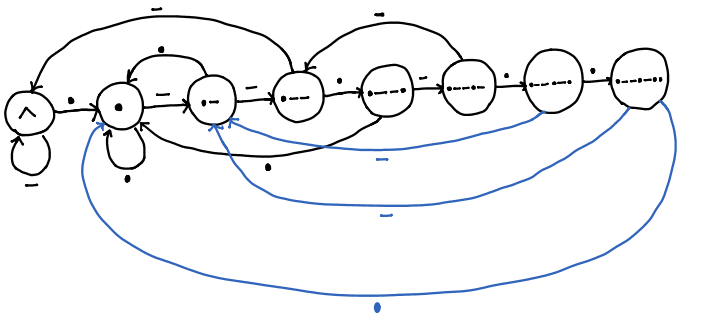


Figure 1: State Diagram for Detecting 'dot-dash-dash-dot-dash-dot-dot'. ‘^’ means the start of the input.

In lab 2, I have implemented a finite state machine that determines if the user enters the sequence 'dot-dash-dash-dot-dash-dot-dot'. The finite state diagram is outlined above in Figure 1. At first I have tried to use a falling-edge GPIO interrupts to update the state machine but the button bounces when I release it. To solve that problem, I have implemented a timer to count how long the button has been pressed. Timer0 starts every 20 milliseconds and counts up whenever the button is pressed. If the counter is above 5 (100 milliseconds), the debouncing algorithm classifies it as a dot. If it is above 15 (300 milliseconds), it is considered as a dash. When the button is released, the state machine updates if the button press is above 100 milliseconds and resets the counter to 0. This is one way to implement button debouncing, which is then used to drive an finite state machine.