

Workshop III

Finite State Machines

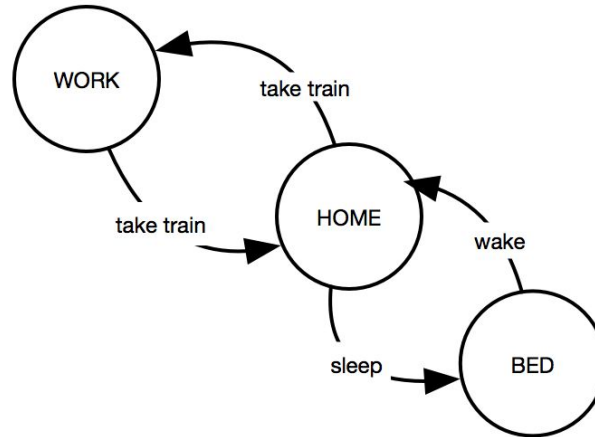
SECTION I

What is a Finite State Machine?

Finite-state machine (FSM)

- A model that is used to design and represent control flow in systems with a finite number of states.
- Also known as a finite-state automaton (FSA)

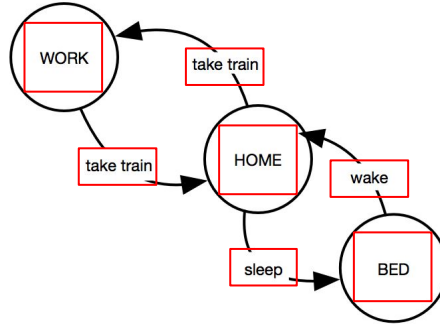
Ex.



Structure of an FSM

- FSMs are made up of states and transitions:
 - States represent the information stored by the system
 - Denoted by circles
 - Transitions represent the movement between states, and occur when a certain input or condition is satisfied.
 - Denoted by arrows, and their conditions are often written along the arrow

Ex.



How can an FSM be used in embedded systems?

- FSMs are a great way to implement the control flow of a system into code in an intuitive and efficient manner.
- FSMs can be implemented in numerous programming languages
 - We will implement FSMs using C/C++ through control statements (ex. if statements and switch statements) and enumerated types

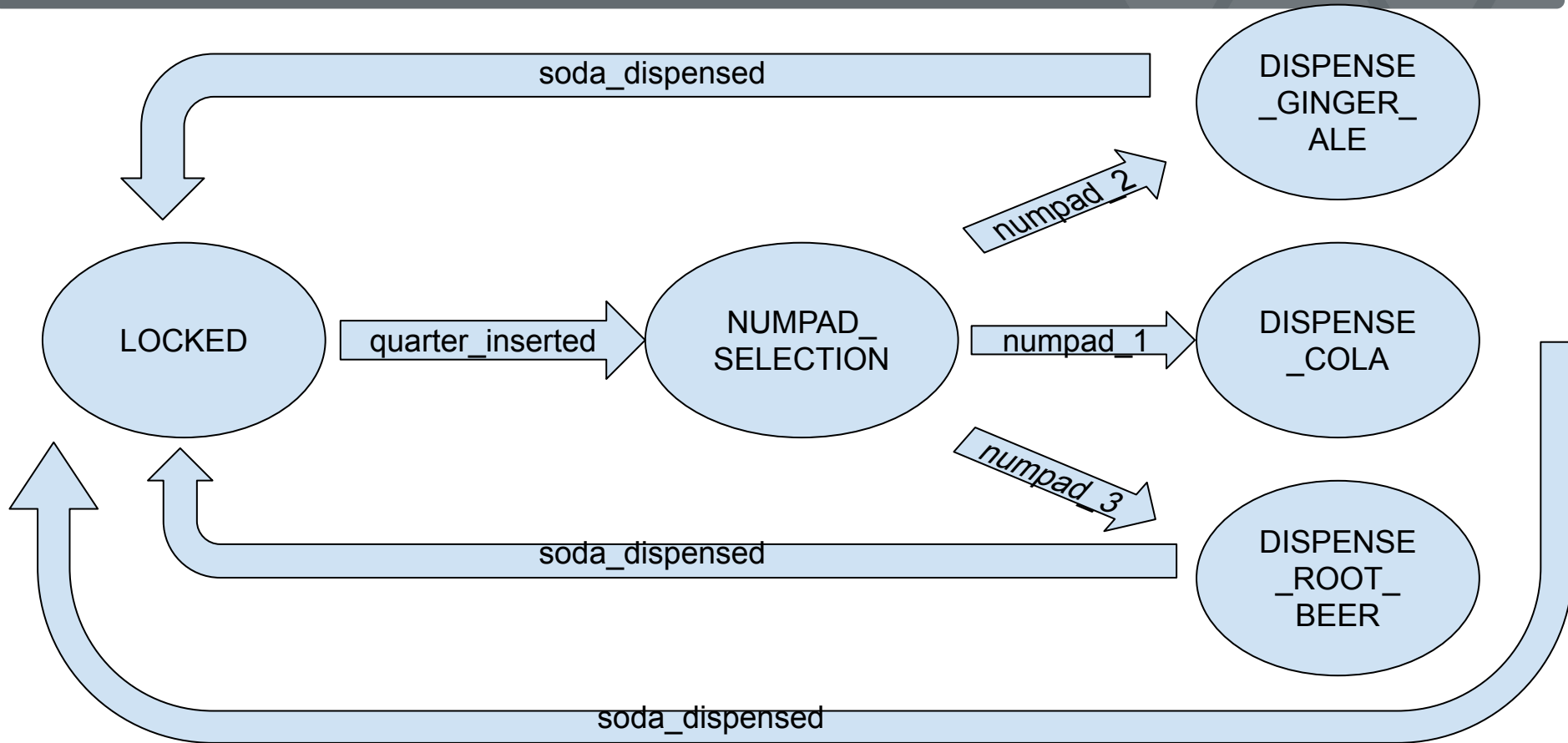
SECTION II

Making a Finite State Machine: Vending Machine Example

Requirements of the Vending Machine

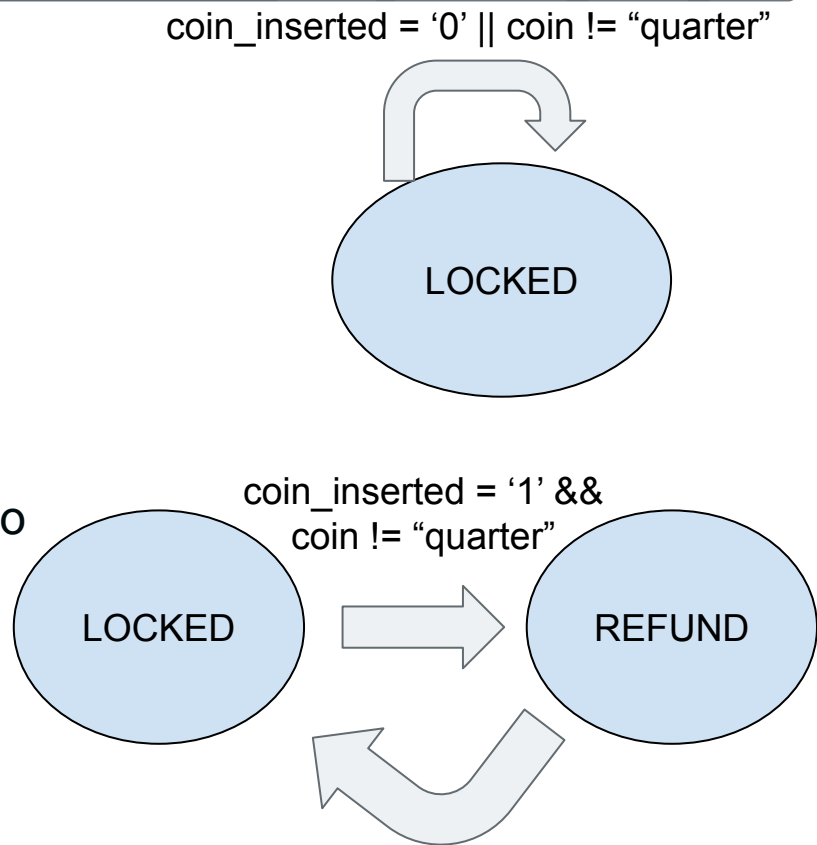
- We want to create a vending machine that accepts only quarters.
- It has a numpad with 3 buttons (1, 2, 3).
- The machine is locked until money is inserted.
- When the user inserts a quarter, they may select an option from the numpad.
- After making their selection, the machine dispenses the selected drink (cola, ginger ale, or root beer).
- Then the machine locks again.

Diagram of the Vending Machine



Bettering the Machine

- The user now enters coins that are not quarters.
 - We want the vending machine to do nothing until they enter a quarter.
 - How do we show this?
- Now, when they enter a coin that is not a quarter.
 - We want to return that coin, then return to the locked state.
 - How would the FSM change now?



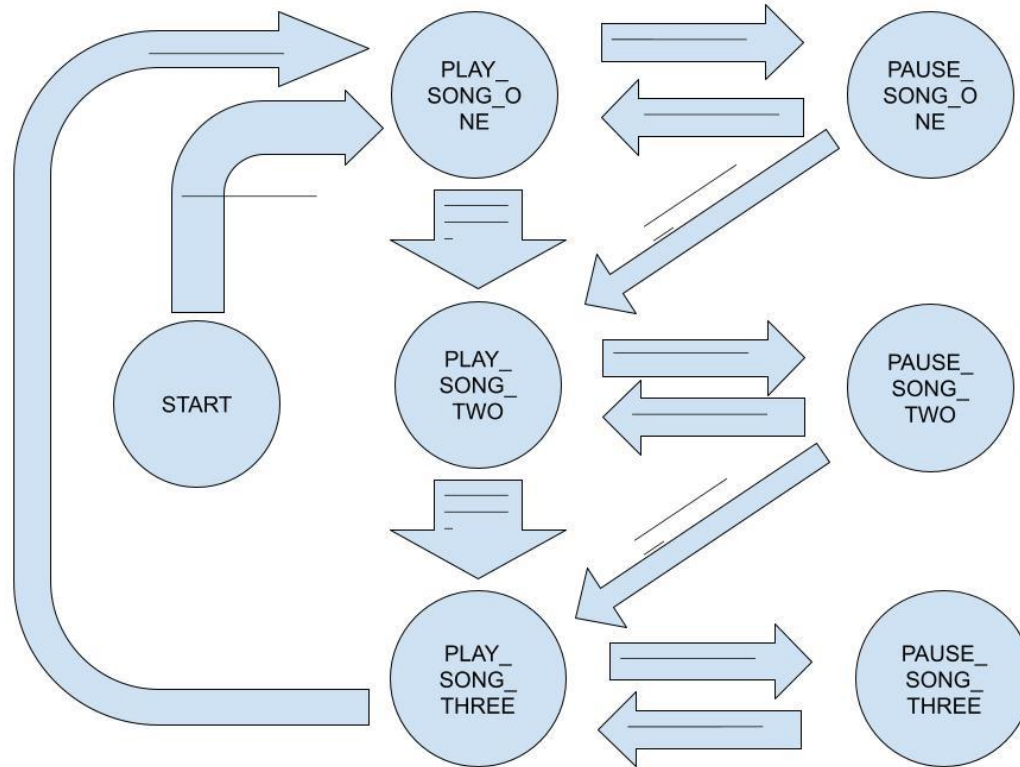
SECTION IV

Building an FSM for iPoduino

iPoduino Requirements

- The player must play 3 songs and have two buttons.
- One of the player's buttons must toggle the music between "Play" mode and "Pause" mode.
 - In "Play" mode, the player must playback a song. It should resume the song that was previously paused if the "Skip/Next Song" button was not pressed.
 - In "Pause" mode, no song should be played.
- The second button should be the "Skip/Next Song" button, which interrupts the current song and plays a new song.
 - Each press of this button must cycle through all songs
 - If the player was in "Pause" mode when the "Skip/Next Song" button is pressed, the player must immediately enter "Play" mode.

Diagram of iPoduino



SECTION V

Coding an FSM for iPoduino

"Life is like a finite state machine — sometimes you're in the 'happy' state, other times you're in the 'confused' state, and occasionally, you find yourself in the 'error' state. Just keep transitioning!"

-Danny Devito

Famous Misquotes

FAIR USE DISCLAIMER

Copyright Disclaimer under section 107 of the Copyright Act 1976, allowance is made for “fair use” for purposes such as criticism, comment, news reporting, teaching, scholarship, education and research.

Fair use is a use permitted by copyright statute that might otherwise be infringing.

Non-profit, educational or personal use tips the balance in favor of fair use.

CC BY-NC-SA 4.0

This work by the Institute of Electrical and Electronics Engineers, UC Irvine Branch, is licensed under CC BY-NC-SA 4.0