ECSE 421 - Embedded Systems

Lab #2 Report: Discrete Systems

Exercise 1: Timer Functions

Task: Binary Counter Demo Video *or* *421_L2_E1_BinaryCounter.MOV*

Question: According to Arduino's documentation, `micros()` will overflow after approximately 70 minutes. Use this to determine the number of bits used by `micros()`. Show your calculations.

$$70\,minutes\ =\ 70\ *\ 60\ *\ 10^{6}\,\mu s = 42 \times 10^{8} \approx 2^{32}$$

From the calculation, we can say that `micros()` uses 32 bits.

Exercise 2: The Smart Home

Question: Is this system best modelled as a Mealy machine or Moore machine? Explain your choice.

- The system is best modelled as a **Mealy machine**
- Reason: Lots of output actions are dependent on the input signals and not the state, for example:
    - The light bulb is turned on when there's motion in "No Light" state, and it is turned off after an amount of time when there's no more motion.
    -
- Exception: the temperature control system which designs are identical in both Moore and Mealy (see state diagrams below)

Task: Create a state diagram for the above specifications. Using your diagram, create the state-transition logic table. Include the current states, as well as the inputs and outputs.

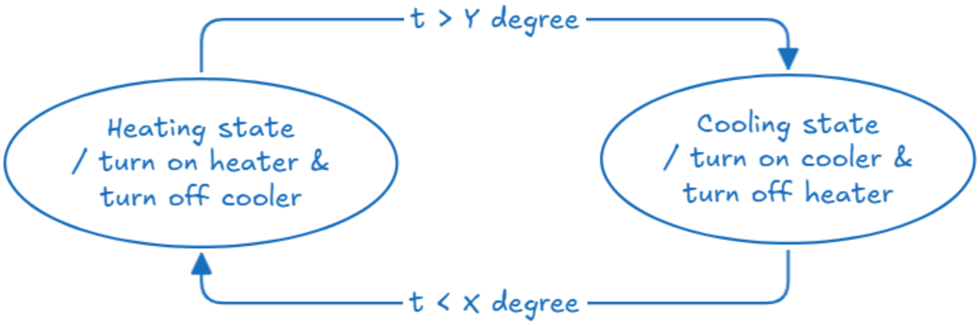
t > Y degree
Heating state / turn on heater & turn off cooler
Cooling state / turn on cooler & turn off heater
t < X degree

Figure 1: Moore machine for the Temperature Monitor

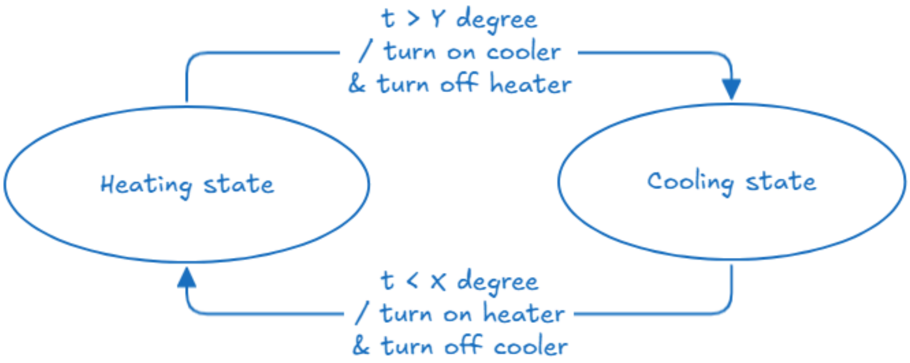
t > Y degree / turn on cooler & turn off heater
Heating state
Cooling state
t < X degree / turn on heater & turn off cooler

Figure 2: Mealy machine for the temperature monitor

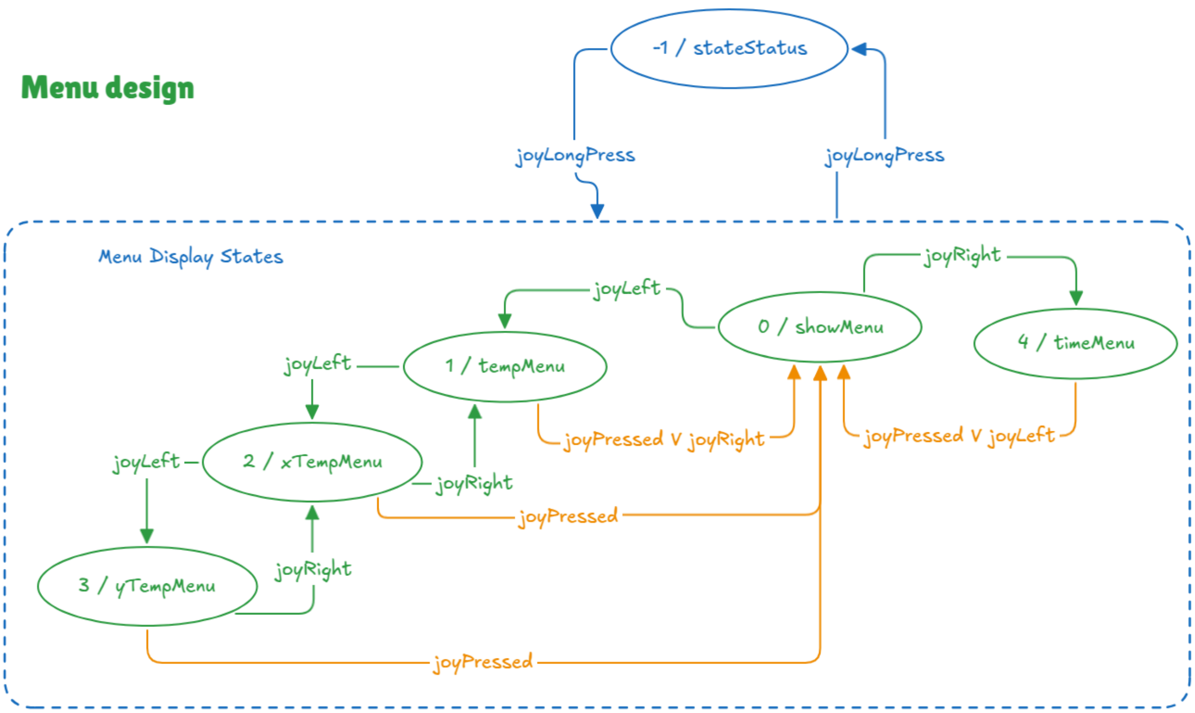

Figure 3: Menu Display state machine, implemented in Moore style

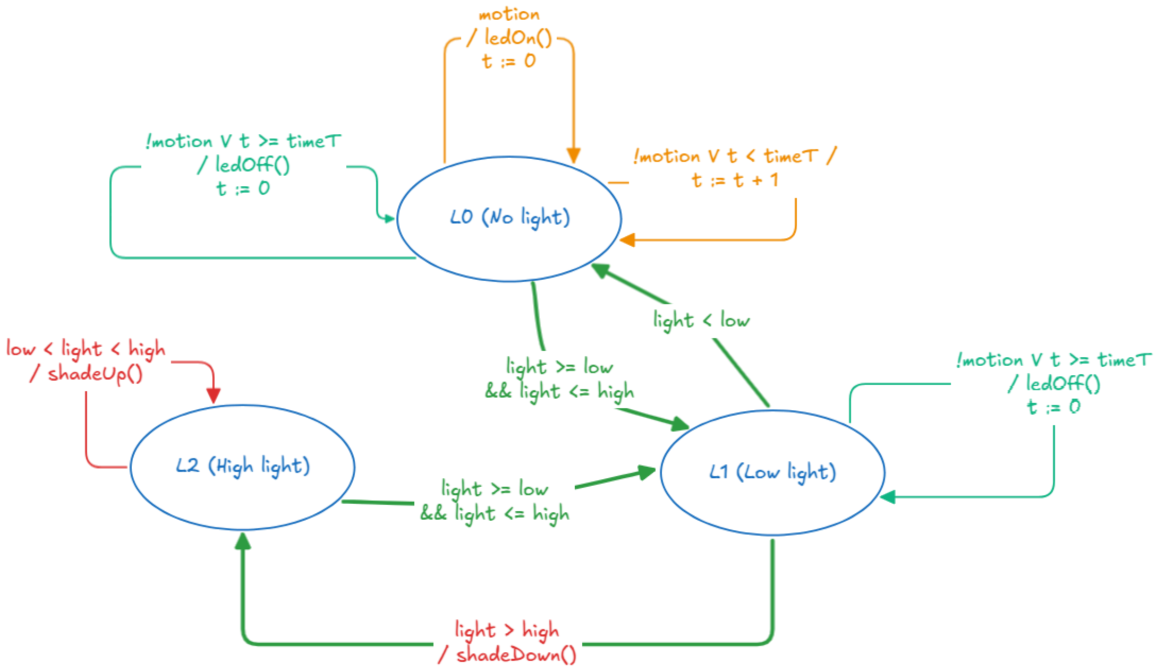

Figure 4: State machine for Indoor Light control, implemented in Mealy

Task: Design the smart home controller in Arduino using switch cases, which should be representative of each state in your state diagram. We recommend creating functions to organize your code.

Using State machine from Figure 4, [I created the demo here]() or 421_L2_E2_SmartHome.MOV

Exercise 3: Door Security System

Task: To begin, draw the state models for each component.

To view the entire state machine, you can view it here on [Excalidraw - State Machines]()
or at URL https://excalidraw.com/#json=hr_7Xb6aLc5B-YTKG7xYr,n8D1Dae0TjlM6HKDBqaWxA

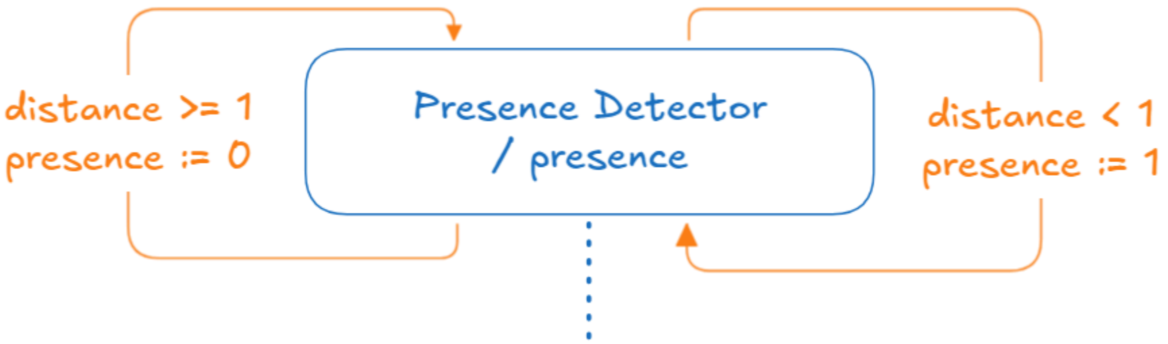

Figure 3.1: Presence detection subsystem, Moore machine design

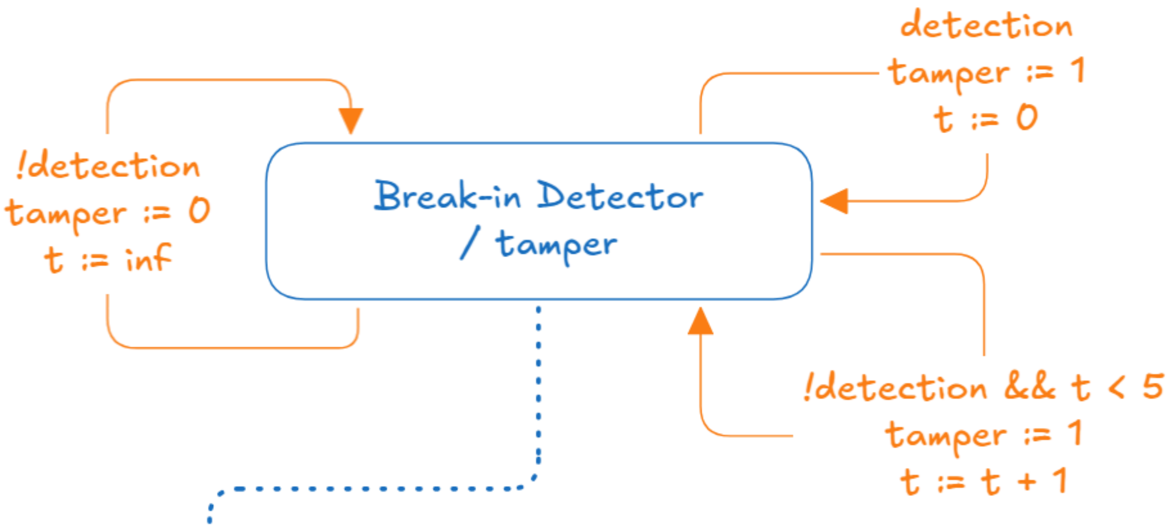

Figure: 3.2: Break-in/Tamper detection subsystem, Moore and extended machine design

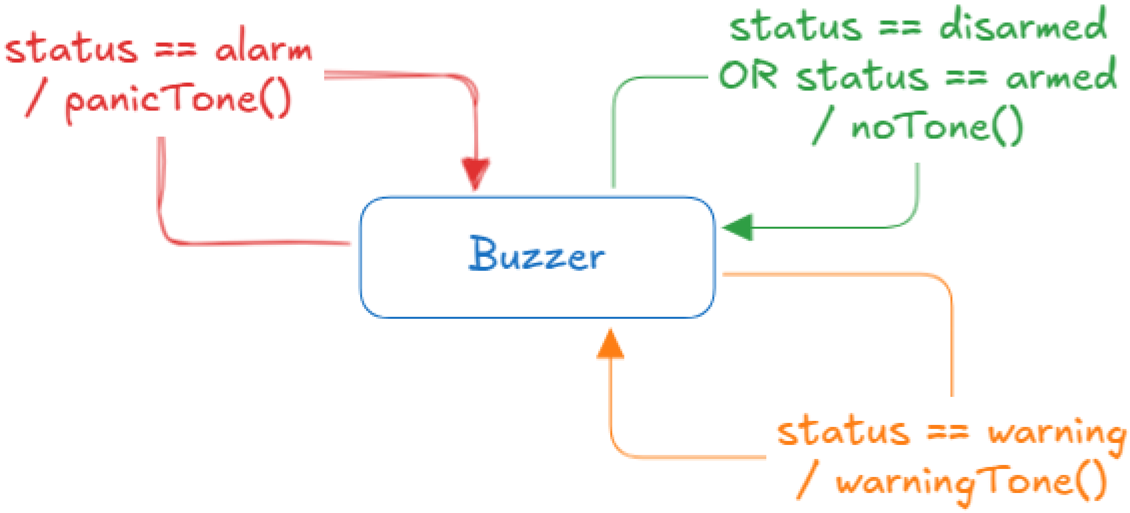

Figure 3.3: Buzzer subsystem, Mealy style state machine

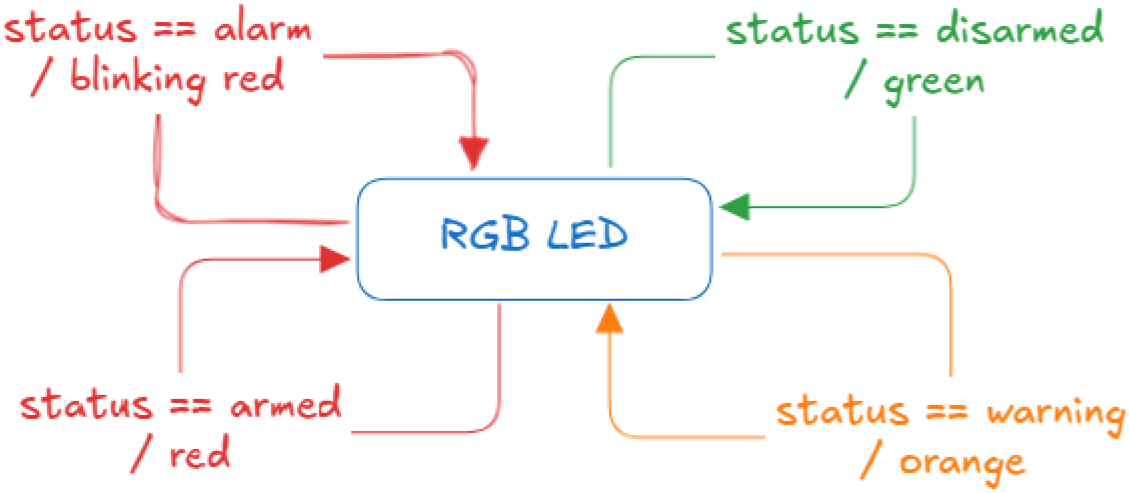

Figure 3.4: RGB LED subsystem, Mealy style state machine

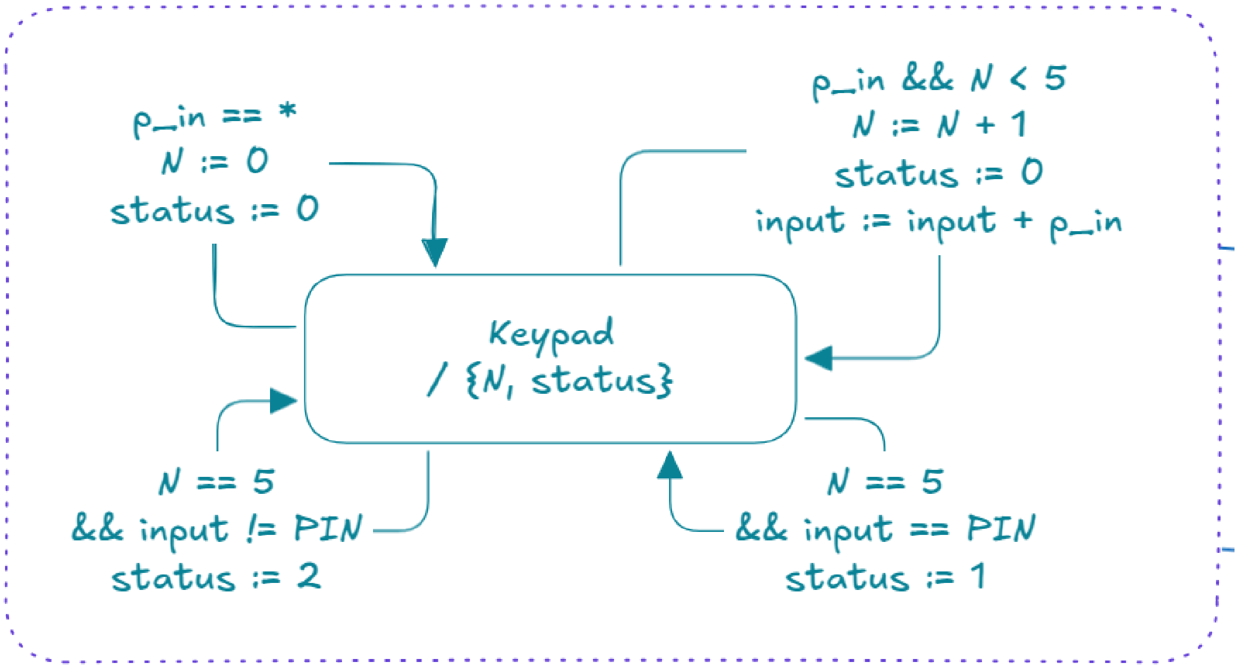

Figure 3.5: Keypad subsystem design, extended Moore machine

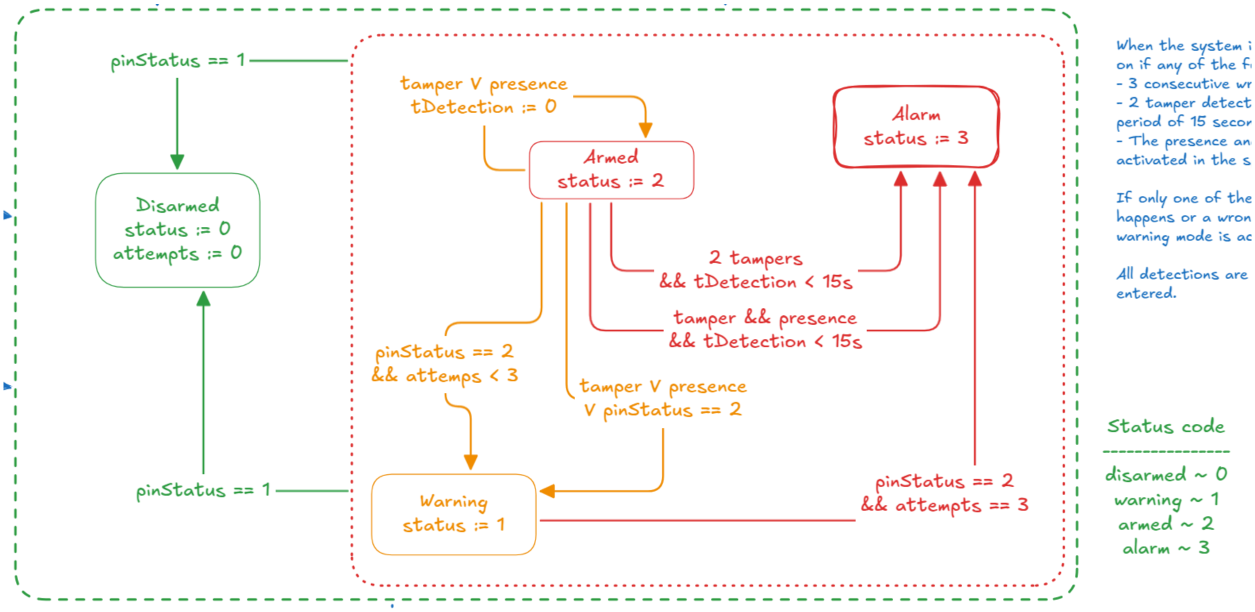

Figure: 3.6: System behaviour, modelled with an extended state machine. Note that the control system only manages the status of the system and sends that signal to the LED and buzzer subsystems.

Task: Using your Arduino and available sensors, design the controller as given in the specifications: The demo video is here! or 421_L2_E3_HomeSecurity.MOV