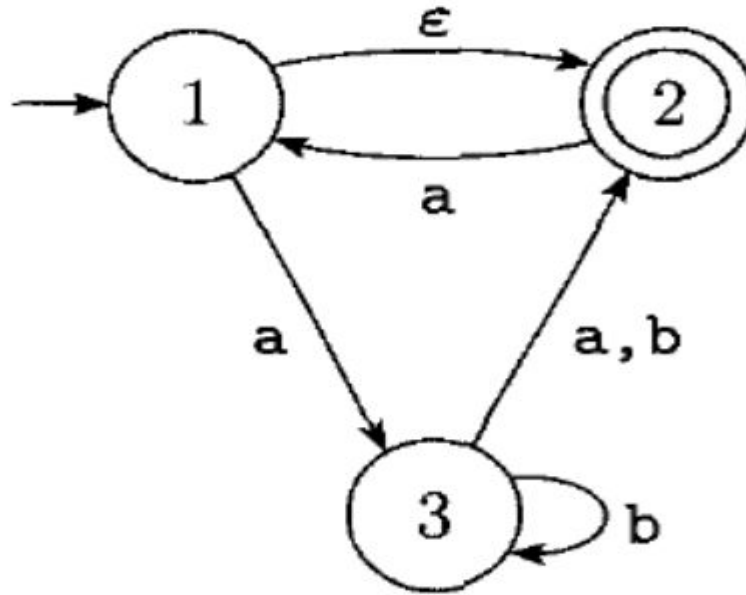


# NFA Interface

By Kellen Hurley



# What are NFAs?

An NFA is a type of finite state machine consisting of five things:

- A finite set of states  $Q$
- A finite alphabet  $\Sigma$  (usually  $\{0, 1\}$ )
- A transition function  $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$
- A set of accept states  $F \subseteq Q$
- A start state  $Q_0 \in Q$

# What are NFAs?

NFAs take strings as input and either “accept” or “reject” them based on whether the NFA finished processing the string in an accept state or not. A string is accepted by an NFA if it finished processing the string on an accept state. It should also be noted that the NFA always begins processing at the start state.

# What are NFAs?

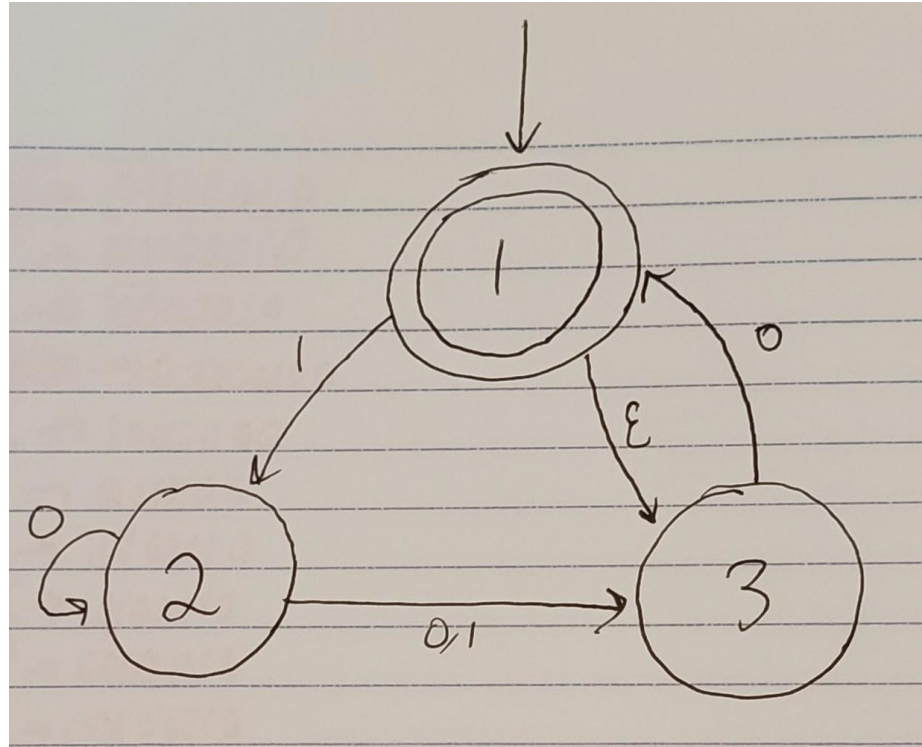
NFAs move from one state to the next based on which character is read in from the string. It should be noted that an NFA can be in multiple states at once since they are non-deterministic.

# What are NFAs?

NFAs can also have what are called “epsilon transitions” which are transitions the NFA may take by consuming no characters from the string.

# Example

Let's try:  
100 and 101



# The Project

In my project, I define 3 NFAs on the MSP430 using structs. The system then looks for a 1, 2, or 3 to be pressed on the keypad to select which NFA the system will work with. Then, the user enters a binary string (up to 16 characters), followed by the “#” key as a delimiter to let the system know the full string has been inputted.

# The Project

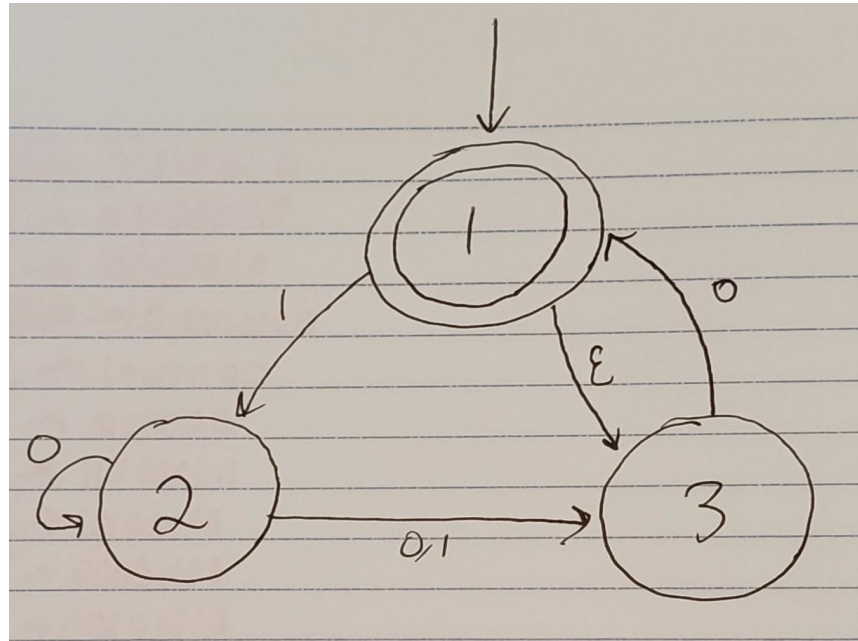
The system then does a bunch of work to determine whether or not inputted string was accepted or rejected by the NFA that was selected. It does this by considering all possible paths through the NFA given the string and if any path ends at an accept state, the string is accepted. If there are any epsilon transitions, the system uses a concept known as the “epsilon closure” to handle these cases.



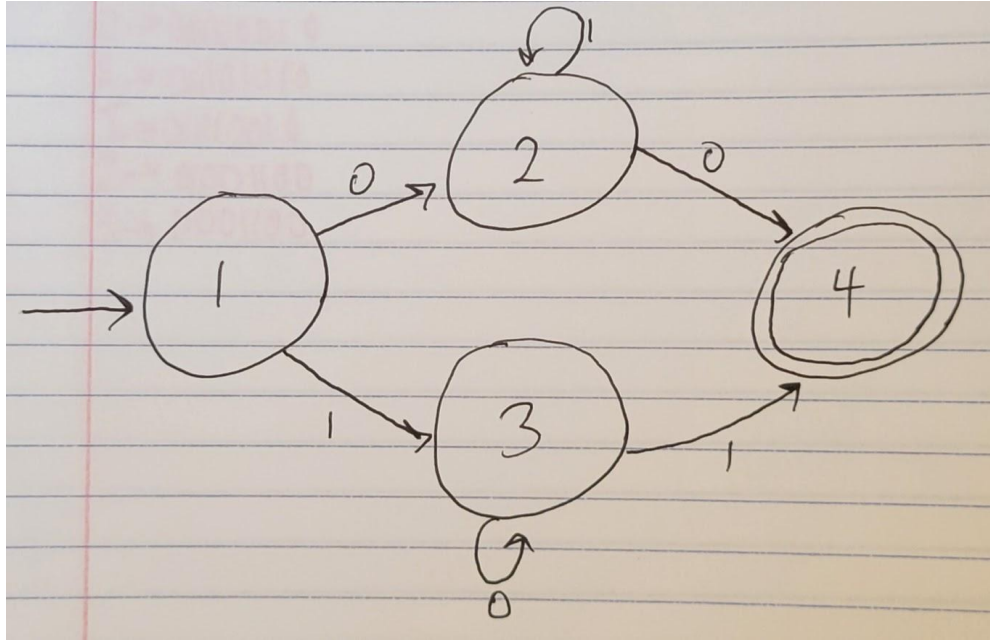
# The Project

Once acceptance of the string has been determined, it displays the selected string on 2 LED light bars and displays the selected NFA, and the acceptance status of the string on the LCD screen. From here, the user can touch a capacitive touch sensor to display the equivalent regular expression corresponding to the NFA they selected. When the user would like to try a new string or NFA, they can simply press the “\*” key to reset the system.

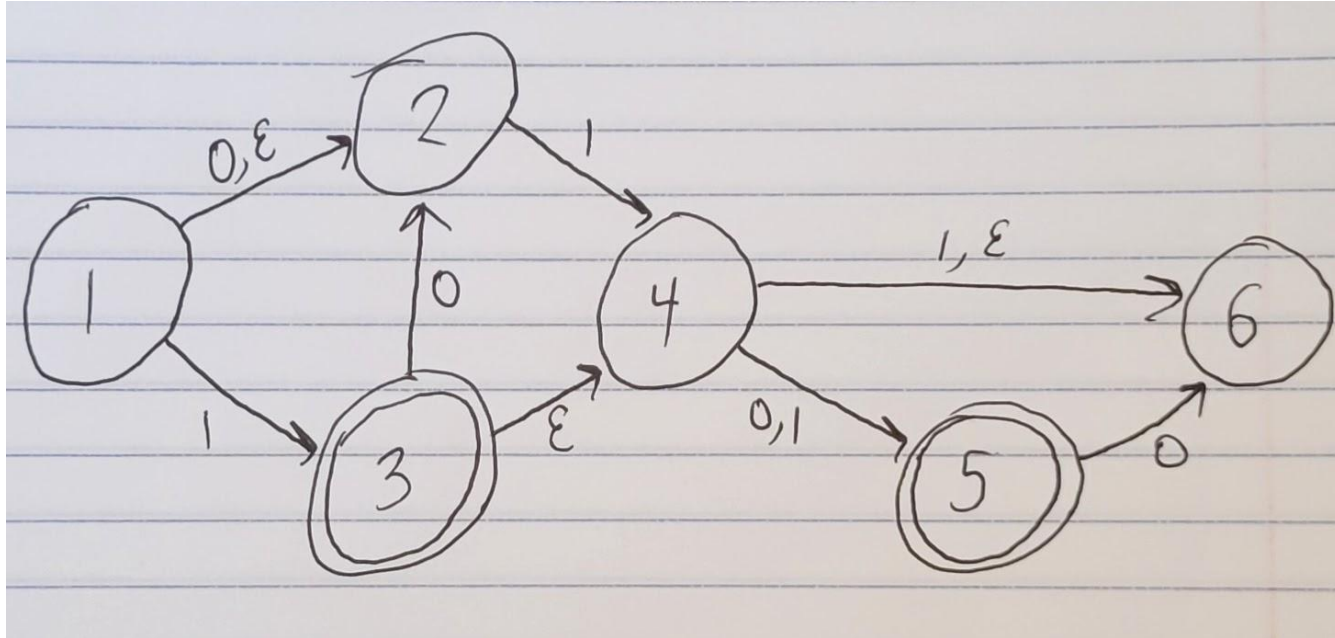
# NFA 1



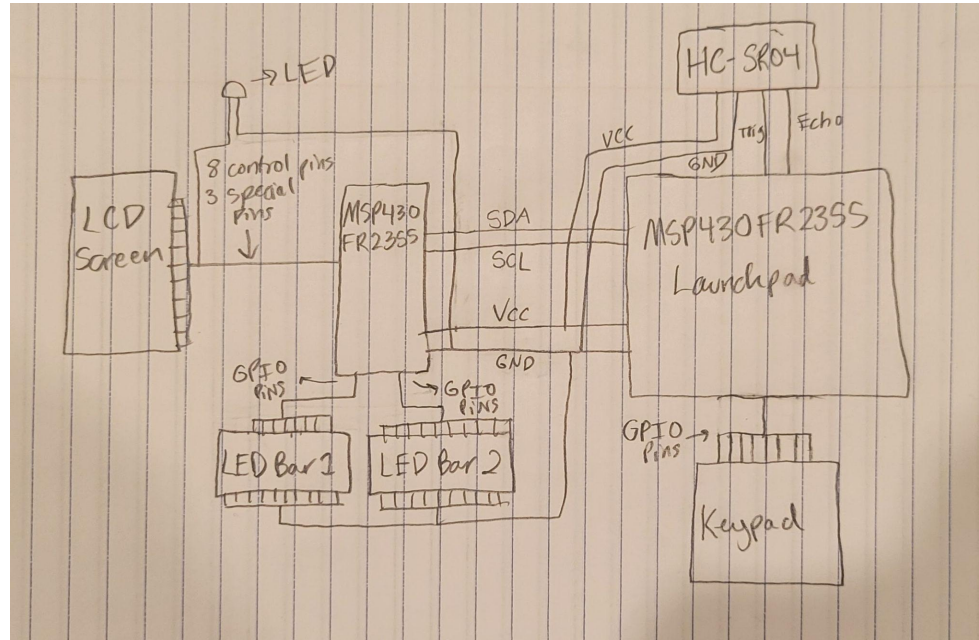
## NFA 2



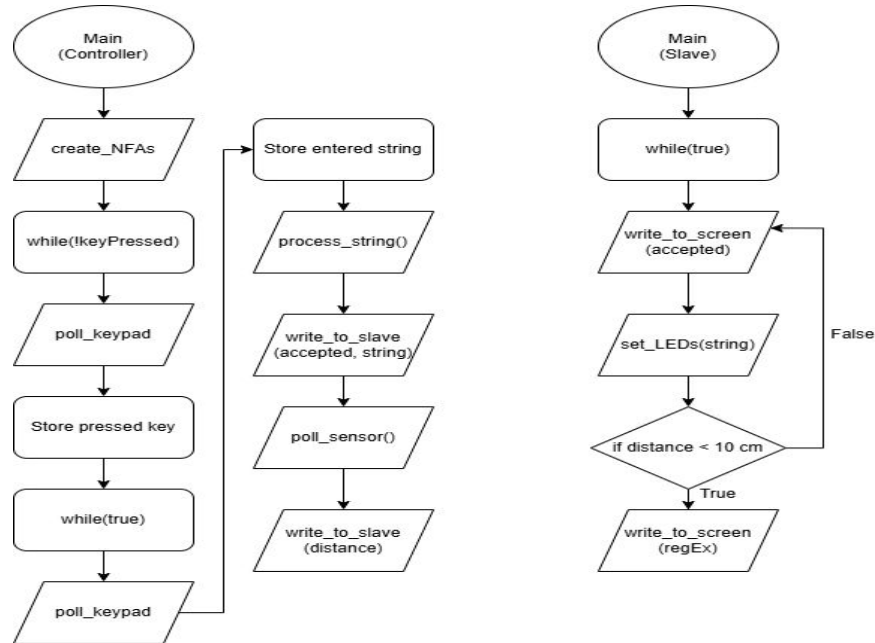
## NFA 3



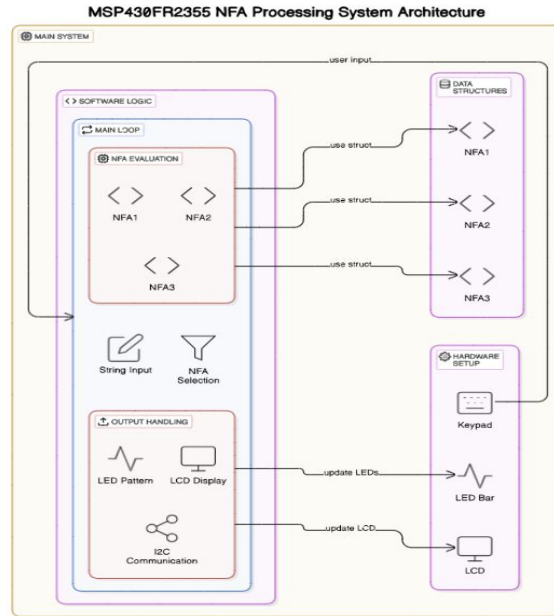
# Circuit Diagram



# Flow Diagram



# Software Architecture Diagram



DEMO