German University in Cairo

Advanced Mechatronics Engineering

MCTR903

Assignment 1

Submitted by:

1.	Abdelrahman Saad	43-16084	T-22
2.	Ahmed Zaghloul	43-15854	T-27
3.	Bishoy Atef	43-16090	T-30
4.	John Gameel	43-15785	T-30
5.	Mona Elboughdady	43-16697	T-21

Submitted to:

1. Prof. Ayman A. El-Badawy
2. M.Sc Catherine Elias
3. M.Sc Lobna Tarek
4. M.Sc Dalia Mamdouh
5. M.Sc. Ahmed Fahmy

Table of content

Problem1 Solution

Part 1:

A) Mealy Machine

States: $Q = \{q_1, q_2, q_3\}$

- q_1: First floor
- q_2: Second floor
- q_3: Third floor.

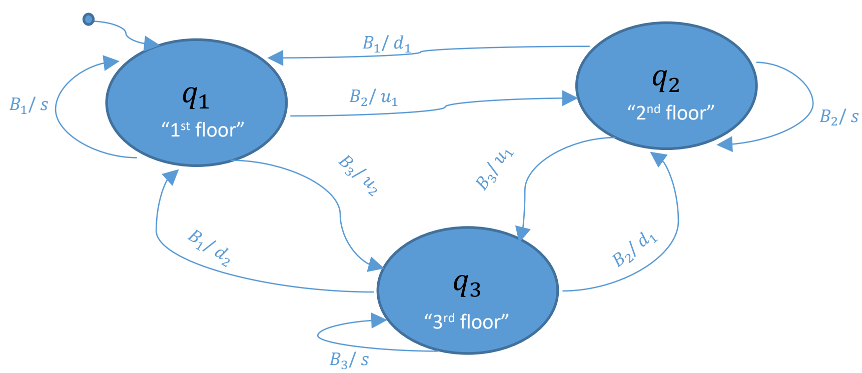
Inputs: $\Sigma = \{B_1, B_2, B_3\}$

- B_1: Go to the 1st floor
- B_2: Go to the 2nd floor
- B_3: Go to the 3rd floor

Outputs: $y = \{u_1, u_2, d_1, d_2, s\}$

- u_1: Go up one floor
- u_2: Go up two floors
- d_1: Go down one floor
- d_2: Go down one floor
- s: Do nothing

State Diagram:



State transition table:

Input / State	B_1	B_2	B_3
q_1	q_1	q_2	q_3
q_2	q_1	q_2	q_3
q_3	q_1	q_2	q_3

B) Moore Machine

States: $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$

- q_1: First floor
- q_2: Second floor
- q_3: Third floor
- q_4: 1st to 2nd
- q_5: 2nd to 1st
- q_6: 1st to 3rd
- q_7: 3rd to 1st
- q_8: 2nd to 3rd
- q_9: 3rd to 2nd.

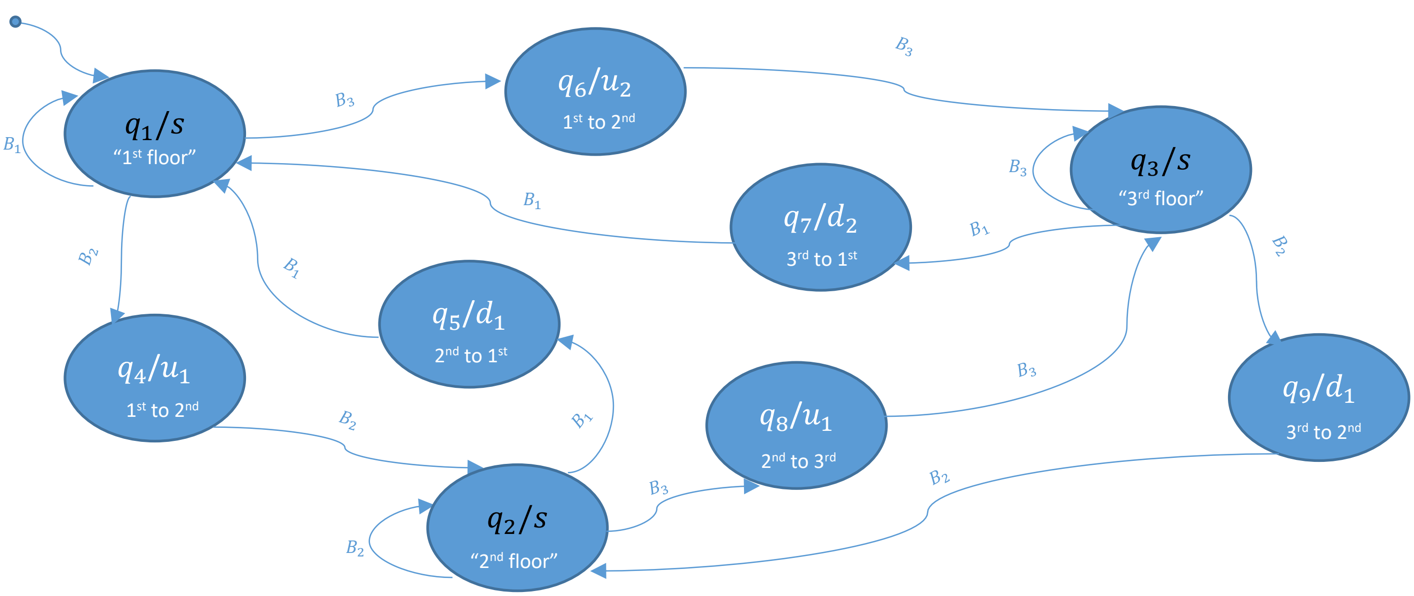
Inputs: $\Sigma = \{B_1, B_2, B_3\}$

- B_1: Go to the 1st floor
- B_2: Go to the 2nd floor
- B_3: Go to the 3rd floor

Outputs: $y = \{u_1, u_2, d_1, d_2, s\}$

- u_1: Go up one floor
- u_2: Go up two floors
- d_1: Go down one floor
- d_2: Go down one floor
- s: Do nothing

State Diagram:



State transition table:

Self-transition is assumed in undefined states

Input / State/output	B_1	B_2	B_3
q_1/s	q_1	q_4	q_6
q_2/s	q_5	q_2	q_8
q_3/s	q_7	q_2	q_3
q_4/u_1	$-$	q_2	$-$
q_5/d_1	q_1	$-$	$-$
q_6/u_2	$-$	$-$	q_3
q_7/d_2	q_1	$-$	$-$
q_8/u_1	$-$	$-$	q_3
q_9/d_1	$-$	q_2	$-$

Simplified solution

States: $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

- q_1: First floor
- q_2: Second floor
- q_3: Third floor
- q_4: Going up one floor
- q_5: Going down one floor
- q_6: Going up two floors
- q_7: Going down two floors

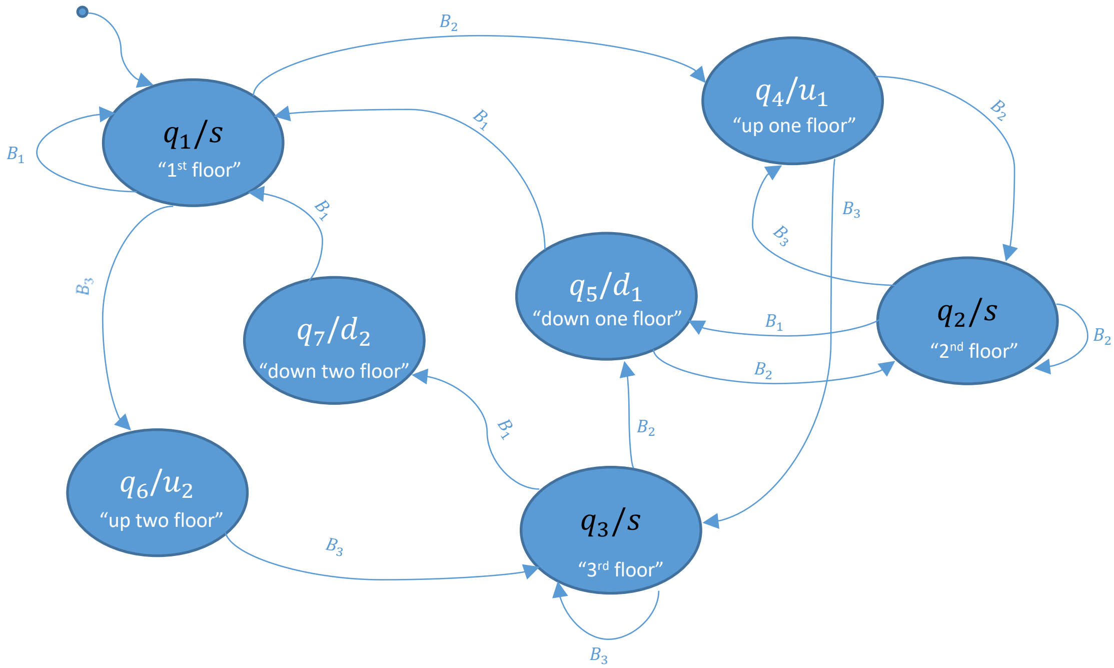
Inputs: $\Sigma = \{B_1, B_2, B_3\}$

- B_1: Go to the 1st floor
- B_2: Go to the 2nd floor
- B_3: Go to the 3rd floor

Outputs: $y = \{u_1, u_2, d_1, d_2, s\}$

- u_1: Go up one floor
- u_2: Go up two floors
- d_1: Go down one floor
- d_2: Go down one floor
- s: Do nothing

State Diagram:



State transition table:

Input / State/output	B_1	B_2	B_3
q_1/s	q_1	q_4	q_6
q_2/s	q_5	q_2	q_4
q_3/s	q_7	q_5	q_3
q_4/u_1	q_4	q_2	q_3
q_5/d_1	q_1	q_2	q_5
q_6/u_2	q_6	q_6	q_3
q_7/d_2	q_1	q_7	q_7

Part2: Simulink and MATLAB code

Mealy machine MATLAB function

```
function [state,y] = fcn(B1,B2,B3)
persistent current_state %state variable that persistes with
every function call
if(isempty(current_state)) %initializing the state variable
during first function call
    current_state = 1;
end
switch current_state
    case 1  %state q1 represents being in the first floor
        if(B2) %B2, the second floor button, is pressed
            current_state = 2;
            y = "u1"; %elevator goes up 1 floor
        elseif(B3) %B3, the third floor button, is pressed
            current_state = 3;
            y = "u2"; %elevator goes up 2 floors
        else %B1, the first floor button, is pressed
            current_state = 1;
            y = "s"; %elevator remains in place
        end
    case 2  %state q2 represents being in the second floor
        if(B1)
            current_state = 1;
            y = "d1"; %elevator goes down 1 floor
        elseif(B3)
            current_state = 3;
            y = "u1"; %elevator goes up 1 floor
        else
            current_state = 2;
            y = "s"; %elevator remains in place
        end
    otherwise %otherwise state q3 represents being in the third
floor
        if(B1)
            current_state = 1;
            y = "d2"; %elevator goes down 2 floors
        elseif(B2)
            current_state = 2;
            y = "d1"; %elevator goes down 1 floor
        else
            current_state = 3;
            y = "s"; %elevator remains in place
        end
end
state = current_state; %output current state
end
```

Moore machine MATLAB function

```
function [state,y] = fcn(B1,B2,B3)
persistent current_state %state variable that persistes with
every function call
if(isempty(current_state)) %initializing the state variable
during first function call
    current_state = 1;
end
switch current_state
    case 1 %state q1 represents being in the first floor
        if(B1) %B1, the first floor button, is pressed
            current_state = 1;
        elseif(B2) %B2, the second floor button, is pressed
            current_state = 4;
        elseif(B3) %B3, the third floor button, is pressed
            current_state = 6;
        end
    case 2 %state q2 represents being in the second floor
        if(B1)
            current_state = 5;
        elseif(B2)
            current_state = 2;
        elseif(B3)
            current_state = 8;
        end
    case 3 %state q3 represents being in the third floor
        if(B1)
            current_state = 7;
        elseif(B2)
            current_state = 9;
        elseif(B3)
            current_state = 3;
        end
    case 4 %state q4 represents the transition state (first
floor to second floor)
        if(B2)
            current_state = 2;
        end
    case 5 %state q5 represents the transition state (second
floor to first floor)
        if(B1)
            current_state = 1;
        end
    case 6 %state q6 represents the transition state (first
floor to third floor)
        if(B3)
            current_state = 3;
        end
```

```
case 7 %state q7 represents the transition state (third floor to
first floor)
        if(B1)
            current_state = 1;
        end
    case 8 %state q8 represents the transition state (second
floor to third floor)
        if(B3)
            current_state = 3;
        end
    case 9 %state q9 represents the transition state (third
floor to second floor)
        if(B2)
            current_state = 2;
        end
end
switch current_state %choose output depending on only the
current state
    case 1
        y = "s"; %elevator remains in place
    case 2
        y = "s";
    case 3
        y = "s";
    case 4
        y = "u1"; %elevator goes up 1 floor
    case 5
        y = "d1";
    case 6
        y = "u2"; %elevator goes up 2 floors
    case 7
        y = "d2"; %elevator goes down 2 floors
    case 8
        y = "u1";
    otherwise
        y = "d1"; %elevator goes down 1 floor
end
state = current_state; %output current state
end
```

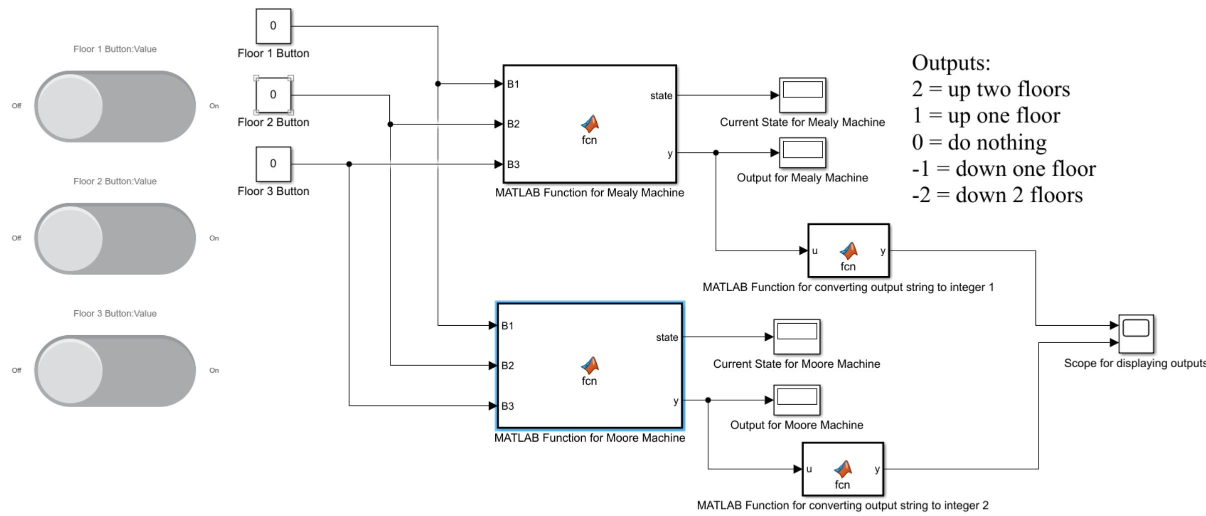
Modified Moore machine MATLAB function

```
function [state,y] = fcn(B1,B2,B3)
persistent current_state %state variable that persistes with
every function call
if(isempty(current_state)) %initializing the state variable
during first function call
    current_state = 1;
end
switch current_state
    case 1 %state q1 represents being in the first floor
        if(B1) %B1, the first floor button, is pressed
            current_state = 1;
        elseif(B2) %B2, the second floor button, is pressed
            current_state = 4;
        elseif(B3) %B3, the third floor button, is pressed
            current_state = 6;
        end
    case 2 %state q2 represents being in the second floor
        if(B1)
            current_state = 5;
        elseif(B2)
            current_state = 2;
        elseif(B3)
            current_state = 4;
        end
    case 3 %state q3 represents being in the third floor
        if(B1)
            current_state = 7;
        elseif(B2)
            current_state = 5;
        elseif(B3)
            current_state = 3;
        end
    case 4 %state q4 represents the transition state (first
floor to second floor)
        if(B2)
            current_state = 2;
        elseif(B3)
            current_state = 3;
        elseif(B1)
            current_state = 4;
        end
    case 5 %state q5 represents the transition state (second
floor to first floor)
        if(B1)
            current_state = 1;
        elseif(B2)
            current_state = 2;
```

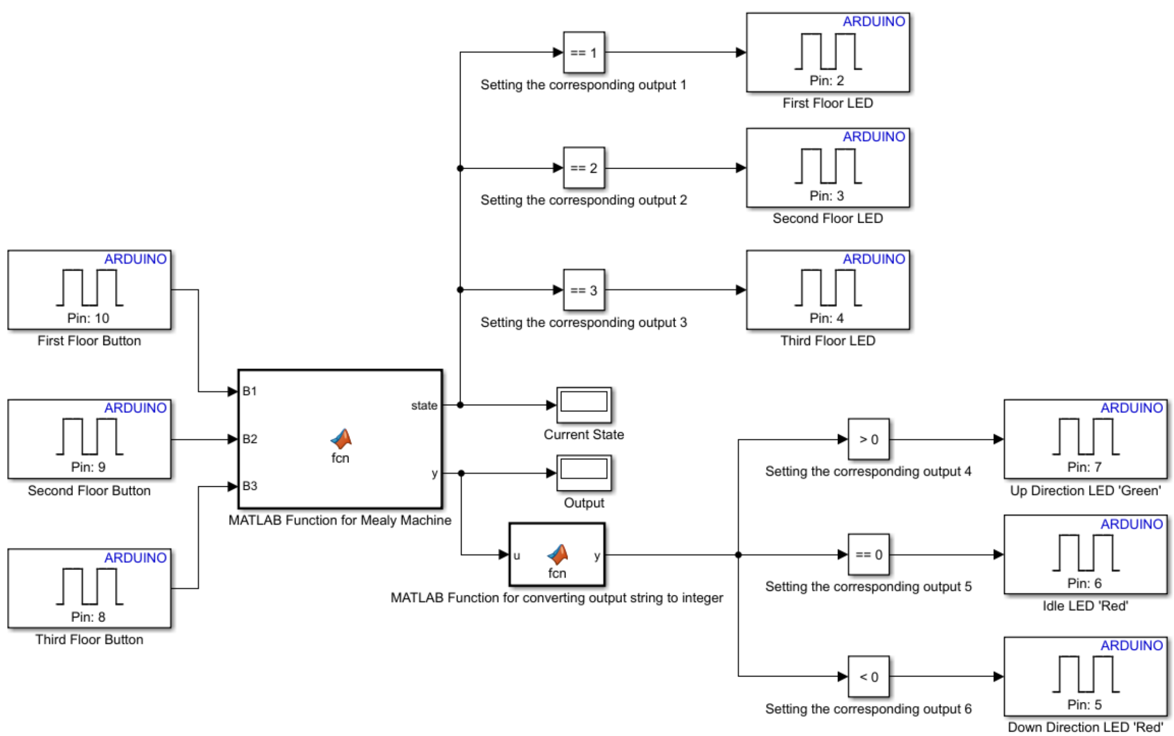
```
        elseif(B3)
            current_state = 5;
        end
    case 6 %state q6 represents the transition state (first
floor to third floor)
        if(B3)
            current_state = 3;
        elseif(B1)
            current_state = 6;
        elseif(B2)
            current_state = 6;
        end
    case 7 %state q7 represents the transition state (third
floor to first floor)
        if(B1)
            current_state = 1;
        elseif(B2)
            current_state = 7;
        elseif(B3)
            current_state = 7;
        end
end
switch current_state %choose output depending on only the
current state
    case 1
        y = "s"; %elevator remains in place
    case 2
        y = "s";
    case 3
        y = "s";
    case 4
        y = "u1"; %elevator goes up 1 floor
    case 5
        y = "d1";
    case 6
        y = "u2"; %elevator goes up 2 floors
    otherwise
        y = "d2"; %elevator goes down 2 floors
end
state = current_state; %output current state
end
```

Simulations can be stepped manually using Simulink controls or using pacing option in Simulink to show the state transition. Discrete solvers were not used as the output was still too fast as the simulation was still running far too fast.

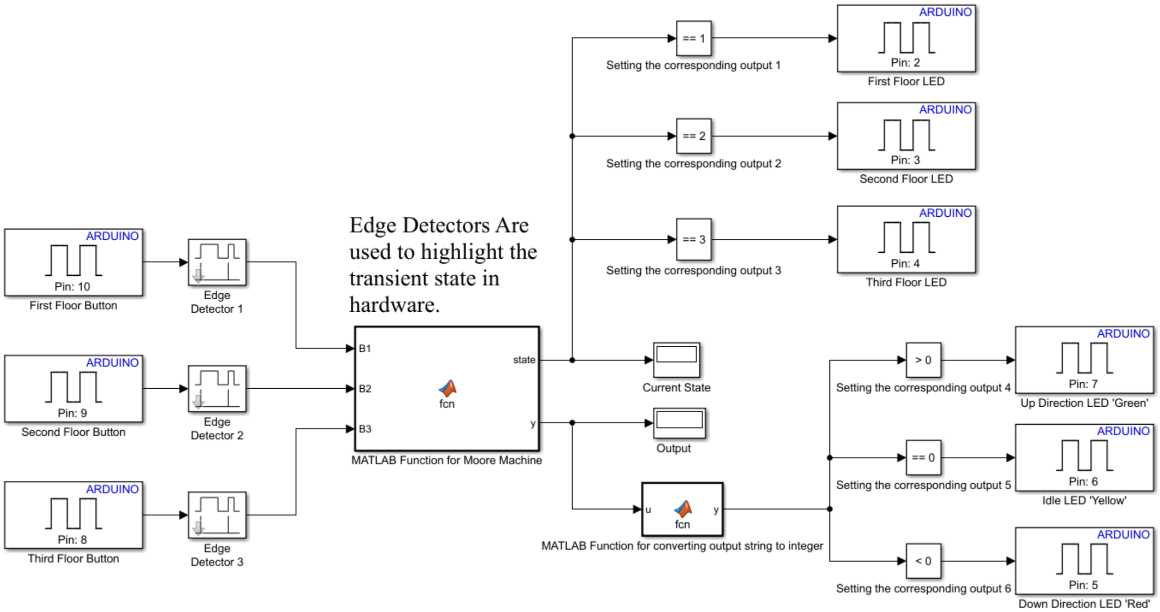
Simulink model for Mealy and Moore machines with switches and a scope



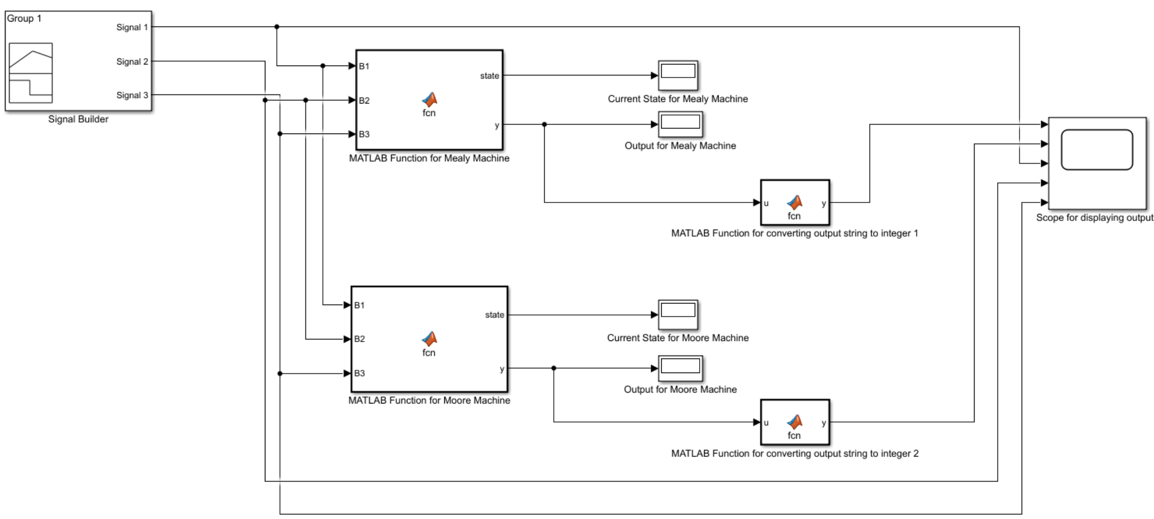
Simulink model of the Mealy machine for Arduino



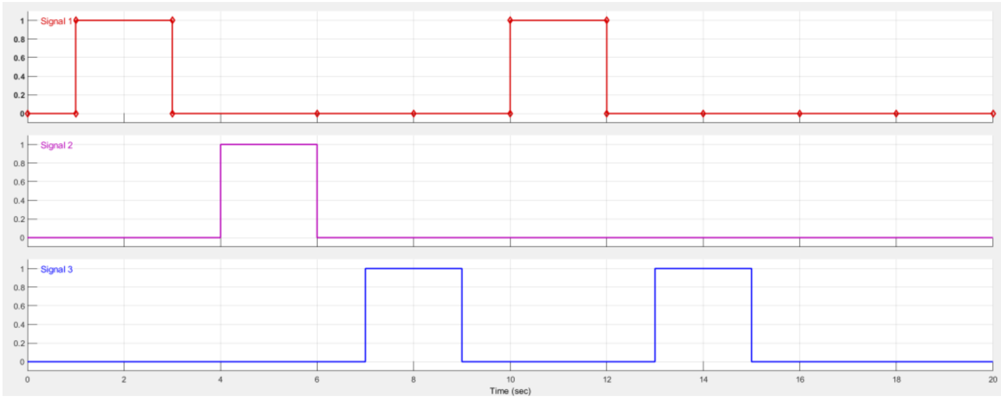
Simulink model of the Moore machine for Arduino



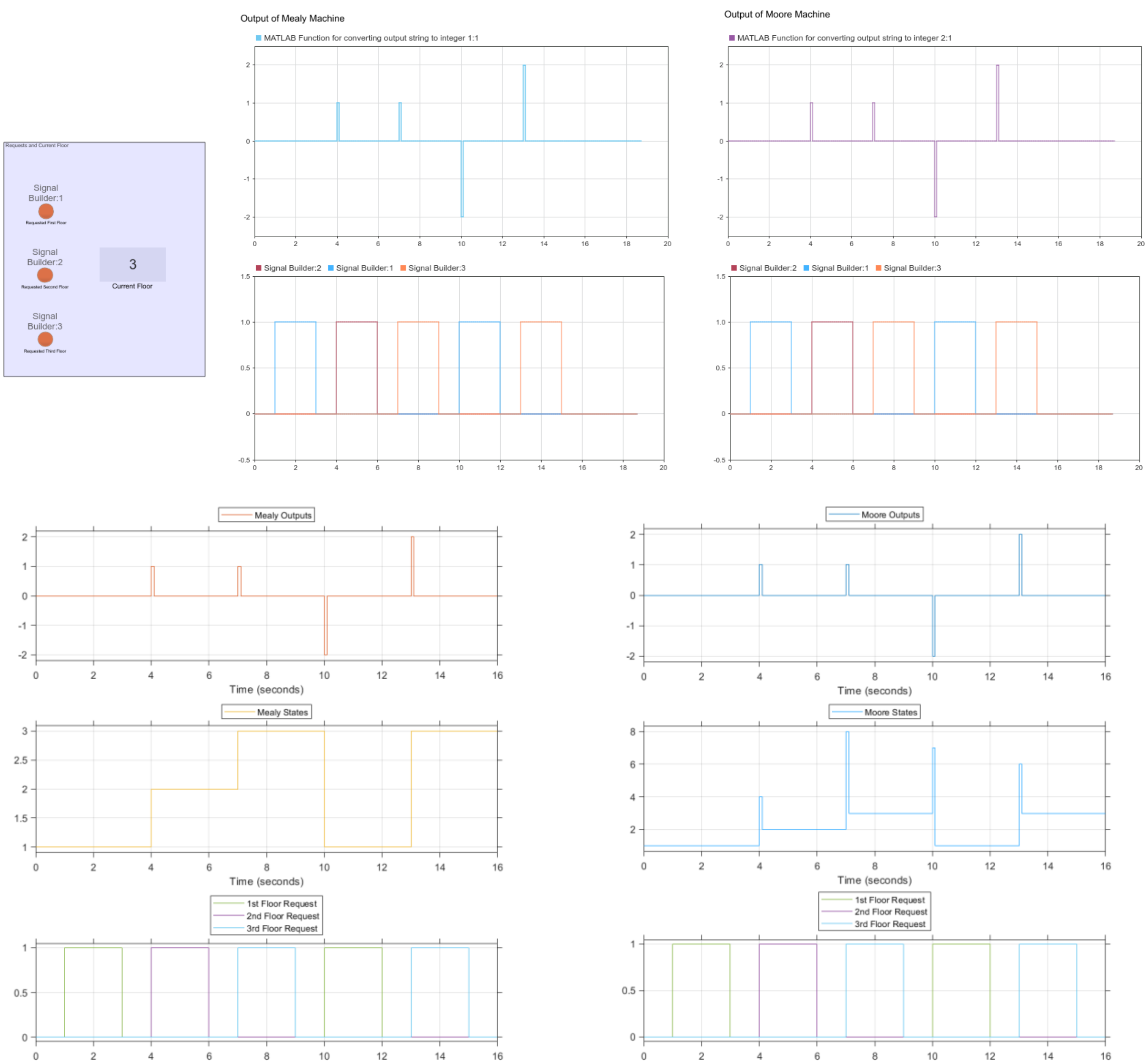
Part3: Using switches or signal builders to change the input



Signals used in the signal builder



Part 4: Plots showing triggering inputs and how this would change the states and the outputs.

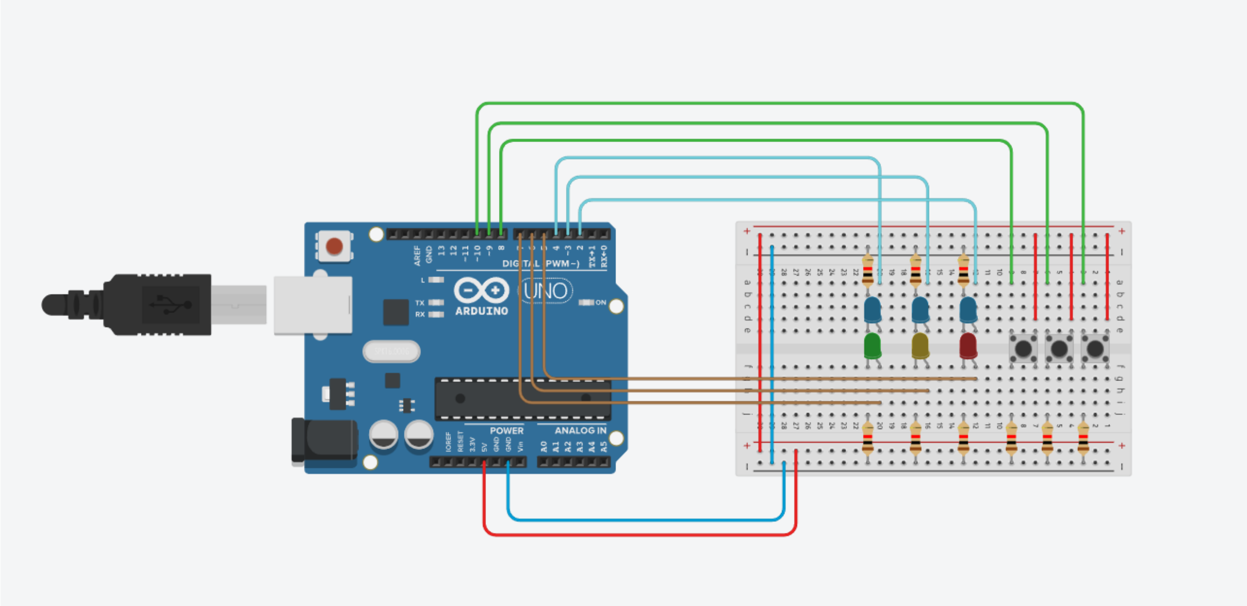


As can be seen in the previous graph the Mealy machine reaches the required floor state as soon as it gets an input while the Moore machine goes through a transient state and if the input to the machine is still the same it moves from this transient state to the required state. Going to the transient states is emphasized in the hardware implementation by using edge detection as can be seen in the hardware model for the Moore machine.

Part 5: Uploading the code on Arduino UNO

The code is uploaded using the Simulink support package for Arduino hardware.

Part 6. Arduino hardware and output.



The result of running the code on the hardware can be seen in the attached video.

Part 7. Comparison between the Mealy and Moore machines.

The Mealy and Moore machine both achieve the task successfully. The Mealy machine has less states than both Moore machines. Mealy machine consists of 3 states while Moore and modified Moore machines consist of 9 and 7 states respectively. The Mealy machine outputs the actuator action as soon as it gets a request and reaches the required floor after one transition. The Moore machine transitions first to a transient state which outputs the actuator action after which if the input to the machine remains the same it transitions to the required floor. This means that Mealy machine requires one transition and Moore requires two transitions to reach requested floor in this case.

8. Comments on performance.

The Mealy machine has less states which requires less coding and overall, less checks in the code while both Moore machines contain more states which requires more checks from the CPU in a simple system, such as this, this is not apparent. Another aspect is transitions Mealy machines require less transitions to reach the requested floor in this case while the difference is not apparent from a time perspective in MATLAB's simulation, stepping the simulation makes it apparent that this is the case. Both machines have their pitfalls as there is no mention of time in the FSM and sending commands to the actuators is not guarded by sensor signals that make sure that the elevator truly reached its destination.