Traffic Light FSM

Gavin Fielder

Group: Gavin Fielder, Michael Musick

EECE 344-02: Digital Systems Design

26 February 2018

Objective

The objective of this labs is to design an implement a traffic light state machine using linked list data structures, and to use PCBArtist to design the circuit.

Basic Theory

A traffic light program for the Warner and 1st street T-intersection will be implemented using a finite state machine. There are four inputs: 1) A traffic sensor on 1st St, 2) a traffic sensor on Warner St, 3) A walk request button on 1st St, and 4) a walk request button on Warner St. There are 8 outputs: 2 traffic lights each with red, yellow, and green, and 2 walk signals, 1 across each street. The state machine will be designed with as many states necessary to produce 1) safety (no cross traffic, yellow before red, etc), 2) All requested traffic/passage serviced in reasonable time frames. In order to implement time day, each state will have a time delay field that allows the program to delay the specified time upon transition to that state.

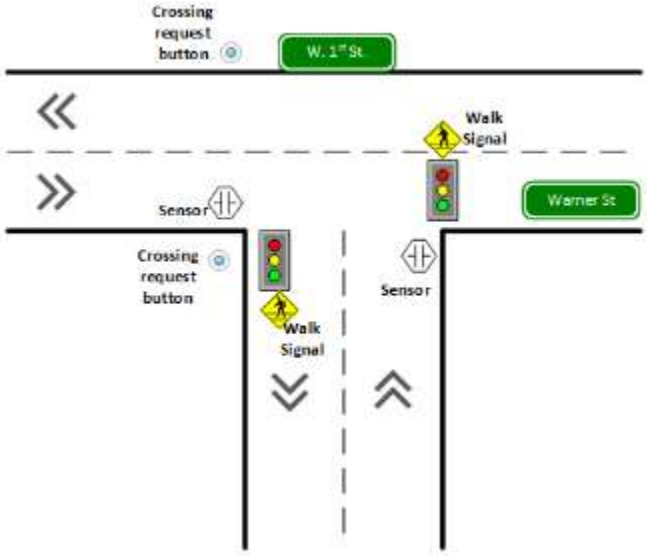

Figure 1

Traffic Light Controller Requirements

- If Traffic is coming through Warner St. and no traffic in 1st st. (SW1=1, SW2=0) then S1=Green, S2=Red.
- If Traffic is coming through 1stSt and no traffic in Warner st. (SW1=0, SW2=1), S1=Red, S2=Green.
- No traffic on any road (SW1=0, SW2=0), gives priority to Warner st.
- To change from Green to Red implement a yellow light for 2seconds (wait state for each of the two lights)
- Green lights should last for 7sec.
- If cars are coming in all directions (SW1=1, SW2=1), cycle through all the states.
- If walk request is initiated on Warner street (B1=1, B2=0), traffic on Warner street and 1ststreet should be stopped (S1=Red, S2=Red) and the walk signal for Warner streets should go ON for 4 sec.

- If walk request is initiated on 1ststreet only (B1=0, B2=1), then S1=Green, S2=Red and the walk signal for 1ststreet should go ON for 4 sec.
- If no walk requests were received, the walk signal should be OFF on both streets.

Procedure

1. Draw the circuit using PCBArtist (See Figure 2)
2. Design the finite state machine
    a. Determine the number of states needed (We decided 7)
    b. Design the state transition graph (See Figure 4)
    c. Set up the state table (See Figure 5)
3. Write controller software
    a. Implement the state structure
    b. Implement the states/state table
    c. Write functions to initialize Systick and any ports needed
    d. Write a function that uses Systick to delay a specified number of seconds
    e. Write main program
4. Implement Circuit according to design
5. Test the traffic light controller by simulating inputs

Figures

Circuit Diagram

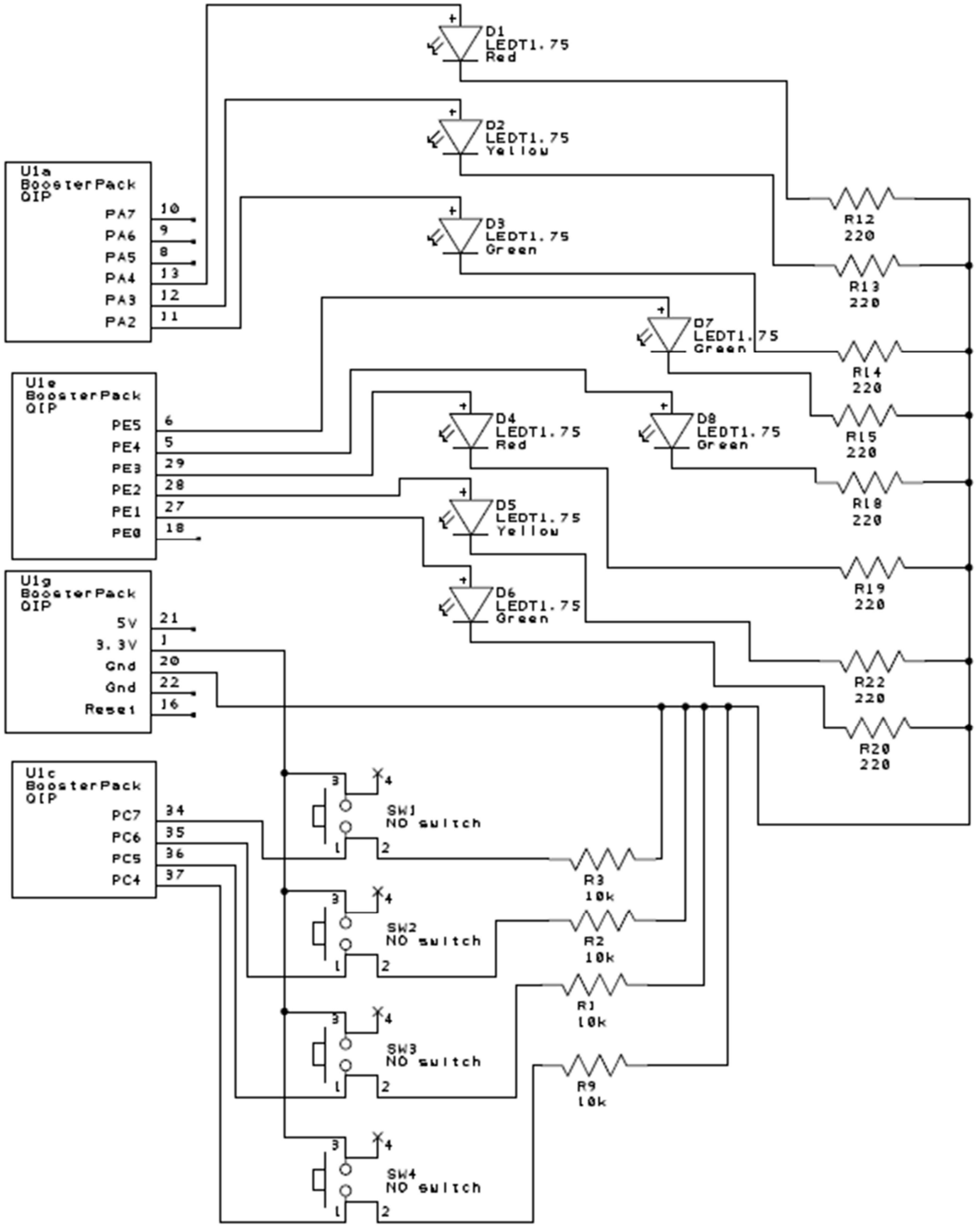

Figure 2: Circuit diagram in PCBArtist

Input Conditions

- PC4: First street walk button
- PC5: Warner street walk button
- PC6: First street traffic sensor
- PC7: Warner street traffic sensor

Using this, PC4-7 create a 4-bit number, with PC4 being the low bit and PC7 the high bit. With this, there are 16 input conditions (0-F)

PC7 Warner traffic	PC6 First traffic	PC5 Warner Walk	PC4 First Walk	Condition
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Figure 3: Input Condition Codes

Outputs

- PA2: 1st street green light
- PA3: 1st street yellow light
- PA4: 1st street red light

- PE1: Warner street green light
- PE2: Warner street yellow light
- PE3: Warner street red light
- PE4: First walk light
- PE5: Warner walk light

State Transition Graph

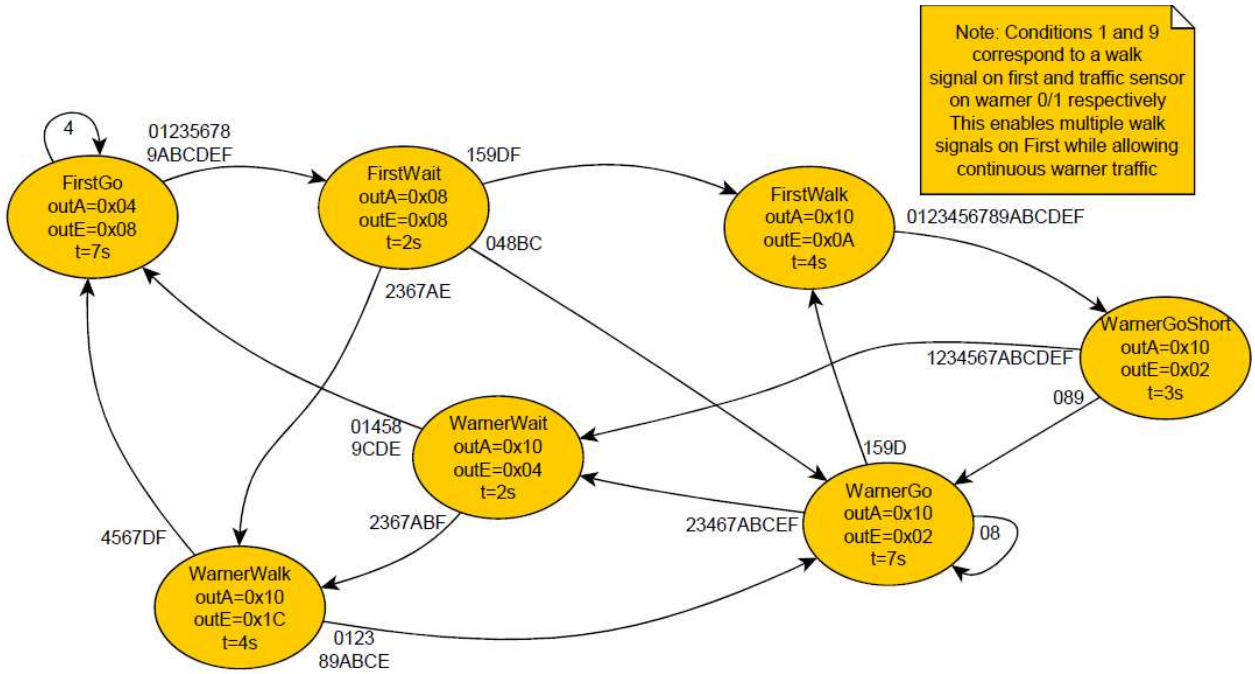

Figure 4: State Transition Graph

State Table

State name >>		WarnerGo	WarnerWait	FirstGo	FirstWait	WarnerWalk	FirstWalk	WarnerGoShort
Delay (s)		7	2	7	2	4	4	3
Output	PortA	0x10	0x10	0x04	0x08	0x10	0x10	0x10
	PortE	0x02	0x04	0x08	0x08	0x38	0x12	0x02
Next	***0***	WarnerGo	FirstGo	FirstWait	WarnerGo	WarnerGo	WarnerGoShort	WarnerGo
	1	FirstWalk	FirstGo	FirstWait	FirstWalk	WarnerGo	WarnerGoShort	WarnerWait
	2	WarnerWait	WarnerWalk	FirstWait	WarnerWalk	WarnerGo	WarnerGoShort	WarnerWait
	3	WarnerWait	WarnerWalk	FirstWait	WarnerWalk	WarnerGo	WarnerGoShort	WarnerWait
	4	WarnerWait	FirstGo	FirstGo	WarnerGo	FirstGo	WarnerGoShort	WarnerWait
	5	FirstWalk	FirstGo	FirstWait	FirstWalk	FirstGo	WarnerGoShort	WarnerWait
	6	WarnerWait	WarnerWalk	FirstWait	WarnerWalk	FirstGo	WarnerGoShort	WarnerWait
	7	WarnerWait	WarnerWalk	FirstWait	WarnerWalk	FirstGo	WarnerGoShort	WarnerWait
	8	WarnerGo	FirstGo	FirstWait	WarnerGo	WarnerGo	WarnerGoShort	WarnerGo
	9	FirstWalk	FirstGo	FirstWait	FirstWalk	WarnerGo	WarnerGoShort	WarnerGo
	A	WarnerWait	WarnerWalk	FirstWait	WarnerWalk	WarnerGo	WarnerGoShort	WarnerWait
	B	WarnerWait	WarnerWalk	FirstWait	WarnerGo	WarnerGo	WarnerGoShort	WarnerWait
	C	WarnerWait	FirstGo	FirstWait	WarnerGo	WarnerGo	WarnerGoShort	WarnerWait
	D	FirstWalk	FirstGo	FirstWait	FirstWalk	FirstGo	WarnerGoShort	WarnerWait
	E	WarnerWait	FirstGo	FirstWait	WarnerWalk	WarnerGo	WarnerGoShort	WarnerWait
	F	WarnerWait	WarnerWalk	FirstWait	FirstWalk	FirstGo	WarnerGoShort	WarnerWait

Figure 5: State Table

State Table (indexed as implemented in code)

		Delay	Output		Next													
State name			PortA	PortE	***0***	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***A***	***B***	***C***	***D***	***E***	***F***
0	WarnerGo	7	0x10	0x02	0	5	1	1	1	5	1	1	0	5	1	1	1	5	1	1
1	WarnerWait	2	0x10	0x04	2	2	4	4	2	2	4	4	2	2	4	4	2	2	2	4
2	FirstGo	7	0x04	0x08	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3
3	FirstWait	2	0x08	0x08	0	5	4	4	0	5	4	4	0	5	4	0	0	5	4	5
4	WarnerWalk	4	0x10	0x38	0	0	0	0	2	2	2	2	0	0	0	0	0	2	0	2
5	FirstWalk	4	0x10	0x12	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
6	WarnerGoShort	3	0x10	0x02	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Figure 6: State table as implemented in code

Implementation

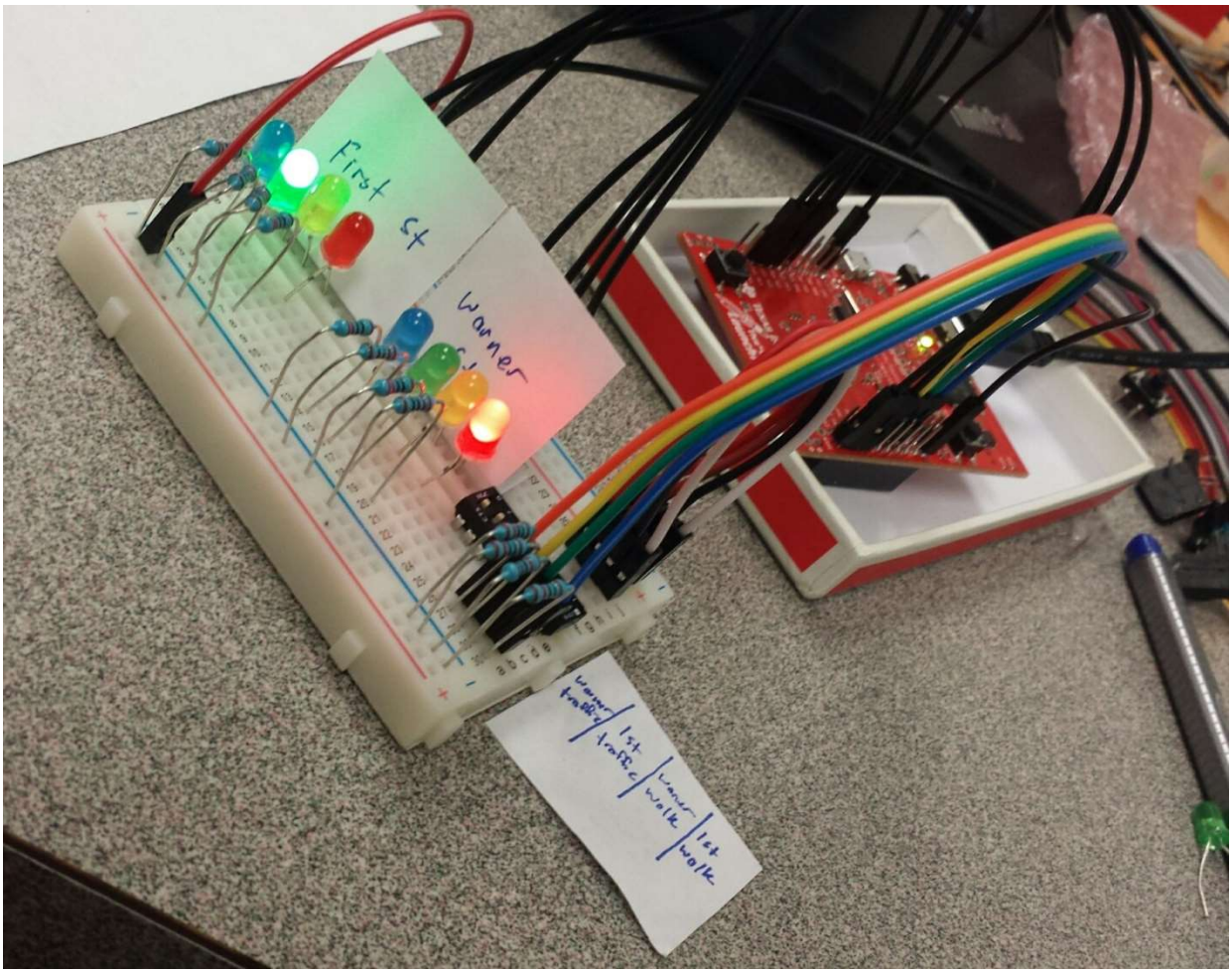

Figure 7: Photo of circuit showing FirstGo

Discussion Questions within Lab Instructions

Under the assumption that there are two crosswalks across Warner, we decided that when a Warner walk signal is given (state WarnerWalk), the walk signal for First street should also be on. This is consistent with the given requirements while allowing any pedestrians across First St. in a state where no traffic could safely flow through the First St. traffic light anyway, since there is no passage either right or left (and there is no straight) from First St.

Conclusion

Initially we designed our state machine with 6 states. FirstWalk, however, output a green light to Warner traffic, and as such that green light had a 4 second delay rather than a 7 second delay. To fix this problem, we added a 7th state, WarnerGoShort, that outputs a green light to Warner traffic for 3 seconds. FirstWalk always transitions to WarnerGoShort, so as to ensure that green light is a 7 second light.

This experiment required more time and effort in design than it did in implementation. The time and effort spent on design made the actual implementation go very quickly, to the extent that the implementation could be considered a negligible part of the experiment. I feel the success of this process method has demonstrated not only the usefulness of design, but also how various tools (PCBArtist, graph/chart software) can enable a development process that is dynamic and has a quick time to turn over when making changes to rectify failures identified during testing.

Our testing took two forms:

- (Uncontrolled) Random input switching and judging whether the resulting behavior is reasonable.
- (Controlled) Testing each condition to see how the states loop while on a single condition.

The controlled testing of each condition certainly does not test each combination and transition of states (and as such is not completely rigorous), but it did allow us to identify problems so that we could make changes to our state transition table. By testing this way, I believe we eliminated the possibilities of specific traffic having an unreasonable wait time or getting stuck completely, but it seems possible that there could be a specific sequence of inputs that extends waiting time for traffic. From all the tests we conducted, however, the controller works as expected.