

Pre-Lab:

- Short introduction video (10:55) and Part 3 fully working video (1:48) is on LMS. Please watch it.
- In this lab we'll be looking at:
 - o Ladder design
 - o Use of Timers
 - o Design / Use of HMIs
 - o State Machine Design

Lab. Work:

Start a New Project

Pick:

- BME P58 3020
- BME XBP 0800

Part 1: One Button Start Stop

For Part 1 you have one input and one output.

Input: PushButton1 (EBOOL) (This is a push button, by default it is Momentary and NO)

Output: SystemOn (EBOOL)

Functionality:

- At start-up all variables are false. (i.e. SystemOn = 0 & PushButton1 = 0)
- When PushButton1 is pressed momentarily, SystemOn should be activated (i.e. SystemOn=1)
- Then, when PushButton1 is pressed again momentarily, SystemOn should be deactivated (i.e. SystemOn=0)
- And the process would repeat if button is pressed again

Reminder:

- Feel free to introduce your own variables upon need
- This is part of the question you had on midterm (Smart Car Park Question)

Here is the timing diagram of the required functionality:



Create an Animation Table named "Part1" and include both of the variables in it.

For Part 2 you have two inputs and one output.

Input:

- light_on (EBOOL) (This is a push button, by default it is Momentary and NO)
- light_flash (EBOOL) (This is a push button, by default it is Momentary and NO)

Output: street_light (EBOOL)

HMI:

- Your HMI, should have 2 push buttons, and one image for the street lamp
- When light is on, lamp should be visible, when light is off lamp should be invisible
- (Pre-lab video during class will make this part more clear)



Figure 1 – Part 2 Sample HMI (feel free to be more creative & artistic & colorful than my design)

Functionality:

- When light_on is pressed, street_light should be on (no latching required)
- When light_flash is pressed, street_light should flash at 1Hz (no latching required)
- Feel free to give different text descriptions to your buttons
- Feel free to introduce your own variables upon need

For your reference:

• Clock Memory Bits

Bit	Symbol	Function	Timebase	Initial State	Platforms	Quantum	Premium
%34	T0.00S	Function	Timebase 10 ms	-	M340: Yes M680: Yes M680 Safety: Yes	Quantum: Yes Momentum: Yes	Premium: Yes Altium: Yes
		Initial State	-	-			
		Platforms					
		Quantum					
		Premium					
		Altium					
		An internal timer regulates the change in status of this bit. It is asynchronous in relation to the PLC cycle. Graph:					
%35	T0.01S	Function	Timebase 100 ms	-	M340: Yes M680: Yes M680 Safety: Yes	Quantum: Yes Momentum: Yes	Premium: Yes Altium: Yes
		Initial State	-	-			
		Platforms					
		Quantum					
		Premium					
		Altium					
		Same as %34					
%36	T0.1S	Function	Time base 1 s	-	M340: Yes M680: Yes M680 Safety: Yes	Quantum: Yes Momentum: Yes	Premium: Yes Altium: Yes
		Initial State	-	-			
		Platforms					
		Quantum					
		Premium					
		Altium					
		Same as %34					
%37	T0.1M	Function	Time base 1 min	-	M340: Yes M680: Yes M680 Safety: Yes	Quantum: Yes Momentum: Yes	Premium: Yes Altium: Yes
		Initial State	-	-			
		Platforms					
		Quantum					
		Premium					
		Altium					

Create an Animation Table named “Part2” and include all the variables in it.

Part 3: Single Traffic Light & Night Mode (HMI Design – State Machine Design)Input:

- start_traffic_light (EBOOL) (This is a push button, by default it is Momentary and NO)
- stop_traffic_light (EBOOL) (This is a push button, by default it is Momentary and NO)
- night_mode (EBOOL) (This is a push button, by default it is Momentary and NO)

Output:

- red_light (EBOOL) (this is the red light, when it is 1 that means red light is on, 0 when it is off)
- yellow_light (EBOOL) (this is the yellow light, when it is 1 that means yellow light is on, 0 when it is off)
- green_light (EBOOL) (this is the green light, when it is 1 that means green light is on, 0 when it is off)
- traffic_lights_on (EBOOL) (this variable becomes a 1 when the traffic light system is enabled, 0 otherwise)

HMI:

- Your HMI, should have 3 push buttons, one traffic light (3 color indicator) and one indicator light (single color)
- (Pre-lab video during class will make this part more clear)

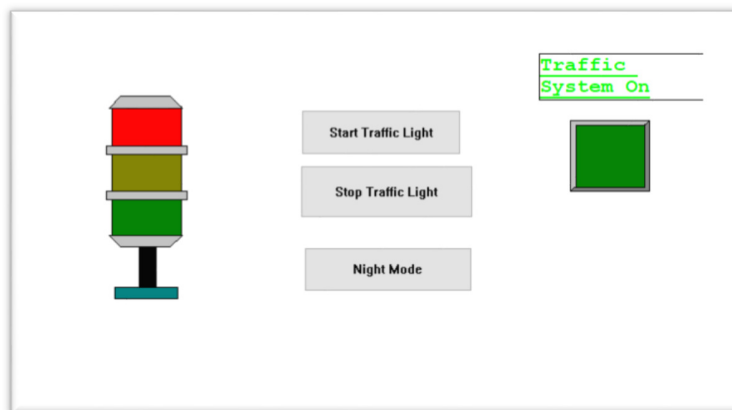


Figure 2 - Part 3 Sample HMI (feel free to be more creative & artistic & colorful than my design)

Functionality:

- Upon start, the traffic light system is disabled and all lights are off
- When start_traffic_light is pressed momentarily (latching is required)
 - Traffic light system is enabled
 - When the system is enabled traffic_lights_on indicator on HMI should be visible
 - The following sequence should be happening on the lights:
 - Only Red_light is on for 5 seconds, then
 - Only Yellow_light is on for 3 seconds then
 - Only Green_light is on for 6 seconds then
 - The whole cycle repeats
- When stop_traffic_light button is pressed momentarily (latching is required)
 - Traffic light system is disabled
 - All the lights are off
 - Traffic_light_on indicator is not visible

- At any point in time (either when system is active or not) when night_mode is pressed:
 - o Yellow_light flashes at 1Hz
 - o Red and Green lights are off
 - o Traffic_light_on indicator is visible
- Feel free to give different text descriptions to your buttons
- Feel free to introduce your own variables upon need
- **Use state machine design approach for this Part**

Here is the timing diagram of regular operation:

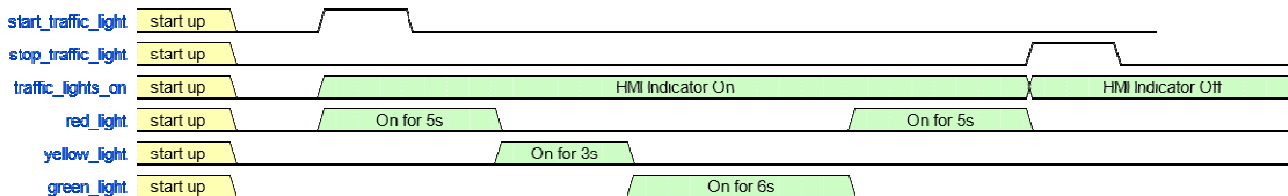


Figure 3 - Timing Diagram of Regular Operation

Here is the timing diagram of night mode operation for two cases:

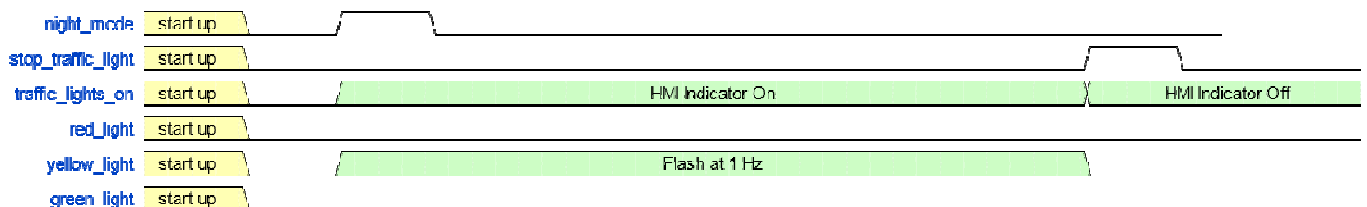


Figure 4 - Timing Diagram of Night Mode Operation when system is initially "Not Active"

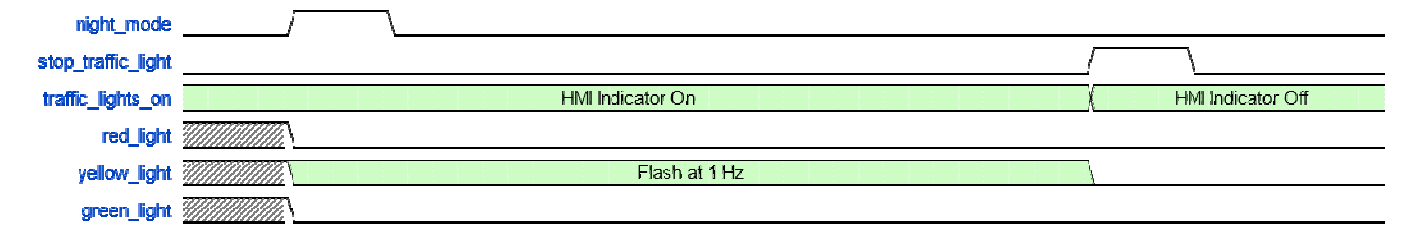


Figure 5- Timing Diagram of Night Mode Operation when system is initially "Active"

Create an Animation Table named "Part3" and include all the variables in it.