Report For Lab 3

Connor McCullough

EEN 312

20 February 2013

**Objective:**

Implement code for a functioning set of stoplights, that when receiving input cues, will cycle through the different light colors as a real stoplight would. This code will be implemented using subroutines, the call function, and the jump function.

**Equipment Used:**

Software:

* Text editor and 8051 ASM assembler
* Step debugger to execute program one instruction at a time
* Data memory, accumulator, code memory

**Flow Chart:**

**P1.2=0**

**P1.3=0**

**CALL TEMPO**

**CALL TEMPO**

P2.2=1 set?

P2.1=1set?

**P1.1=0**

**P1.3=0**

**CALL TEMPO**

**CALL TEMPO**

**CALL TEMPO**

**P1.0=0**

**P1.4=0**

**P1.1=0**

**P1.3=0**

**P1.1=0**

**P1.3=0**

**CALL TEMPO**

**CALL TEMPO**

**P1.0=0**

**P1.3=0**

**P1.0=0**

**P1.3=0**

**P1.2=0**

**P1.3=0**

**CALL TEMPO**

**CALL TEMPO**

**P1.0=0**

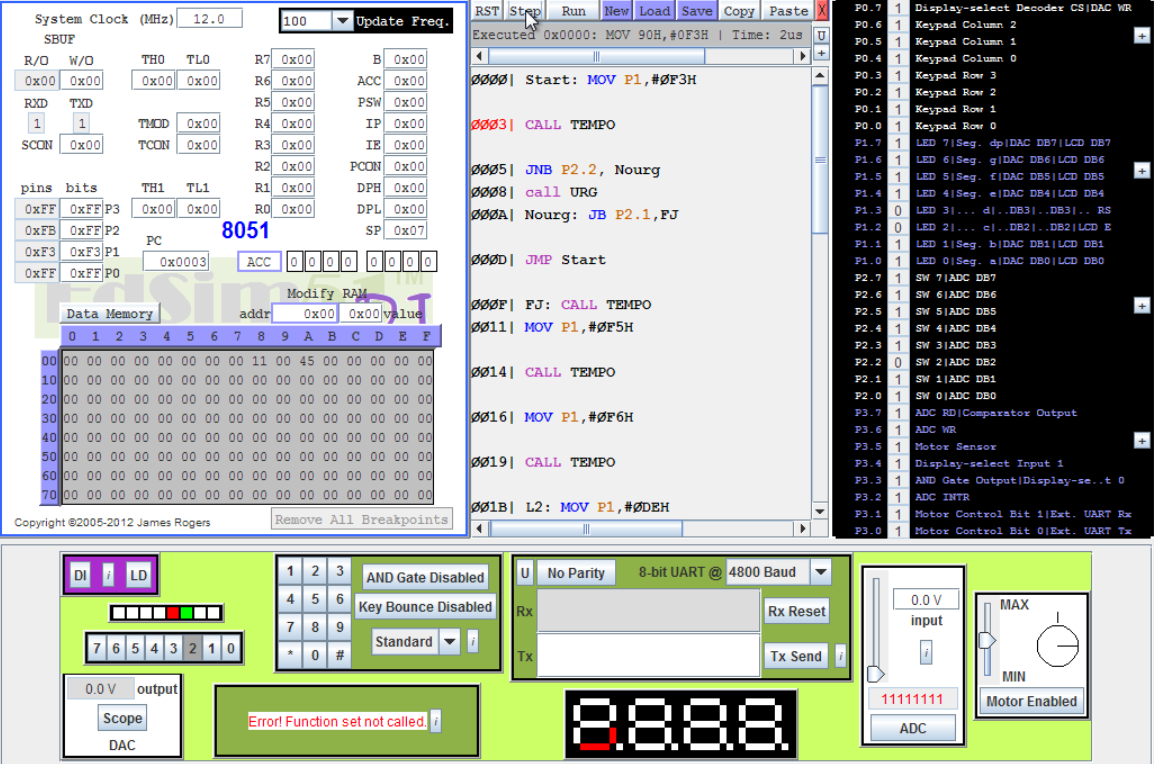
**P1.5=0**

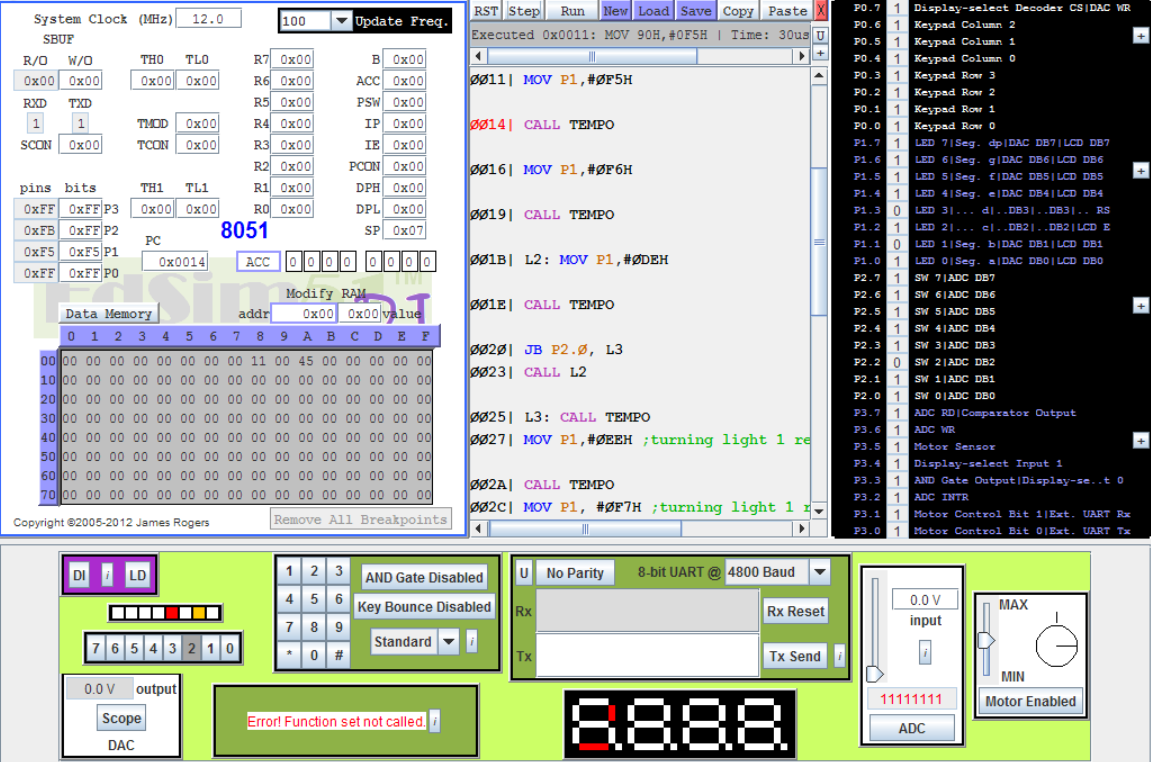
**CALL TEMPO**

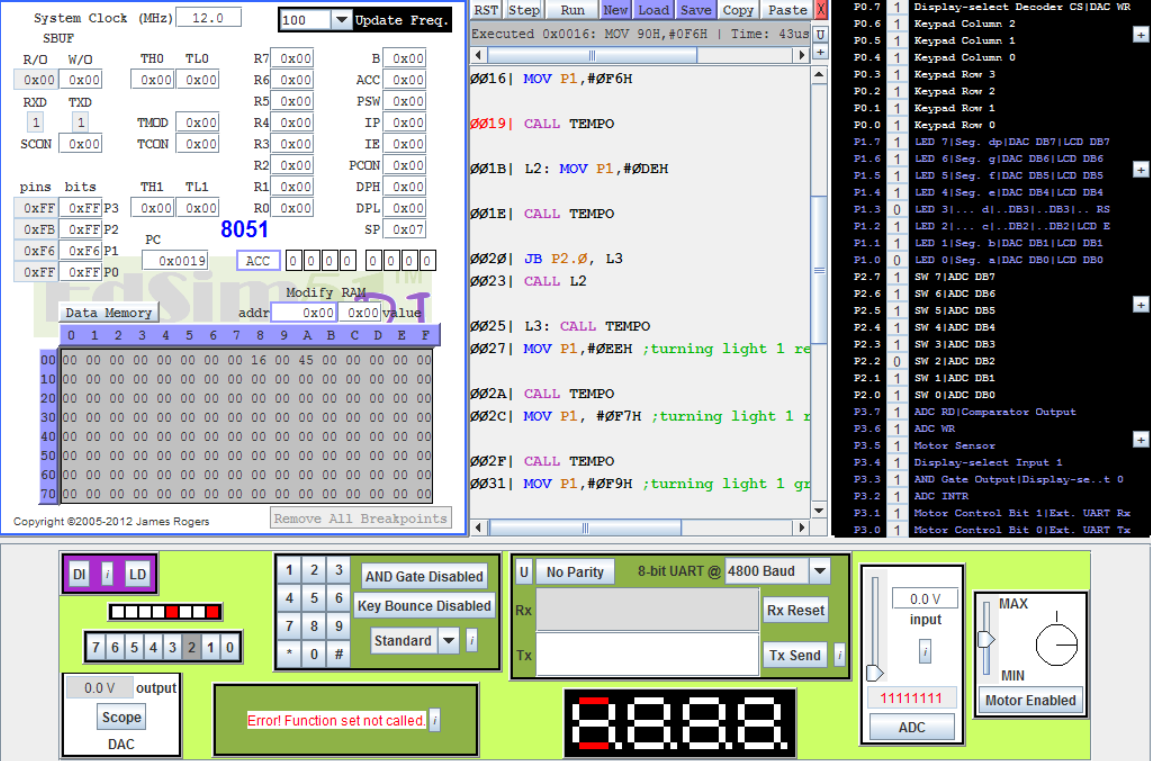
P2.0=1set?

**Results:**

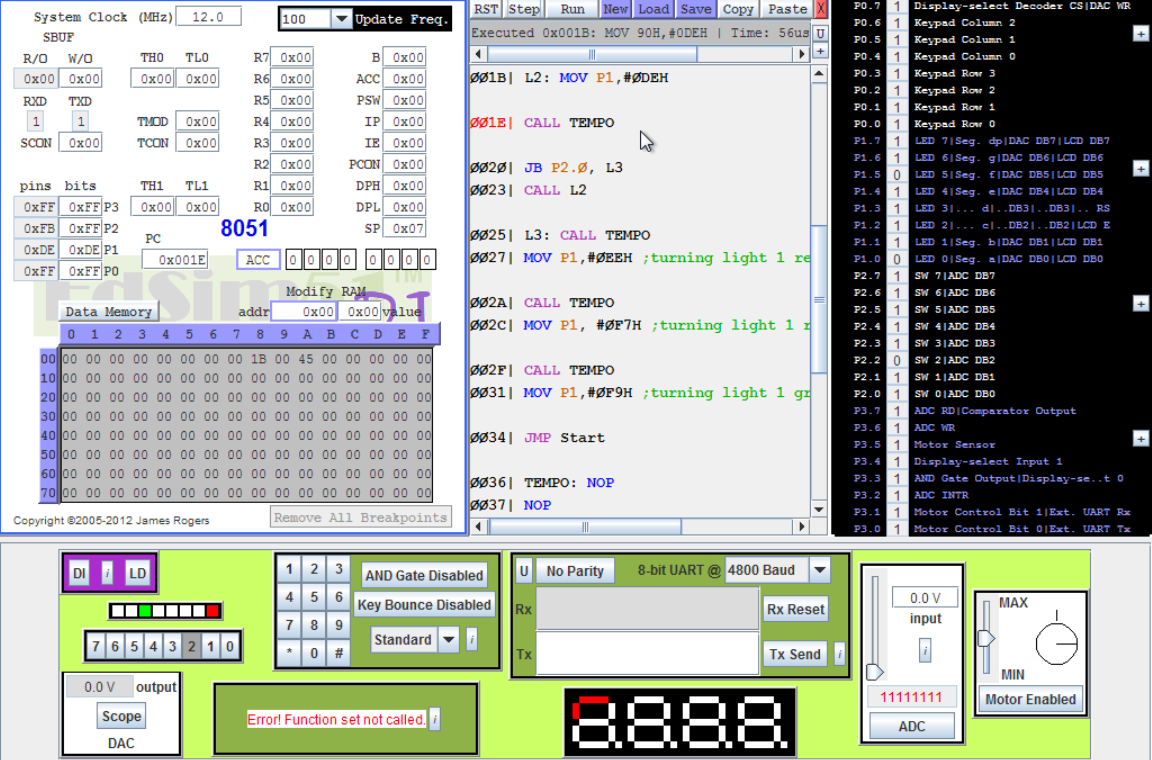
P2.1=1:

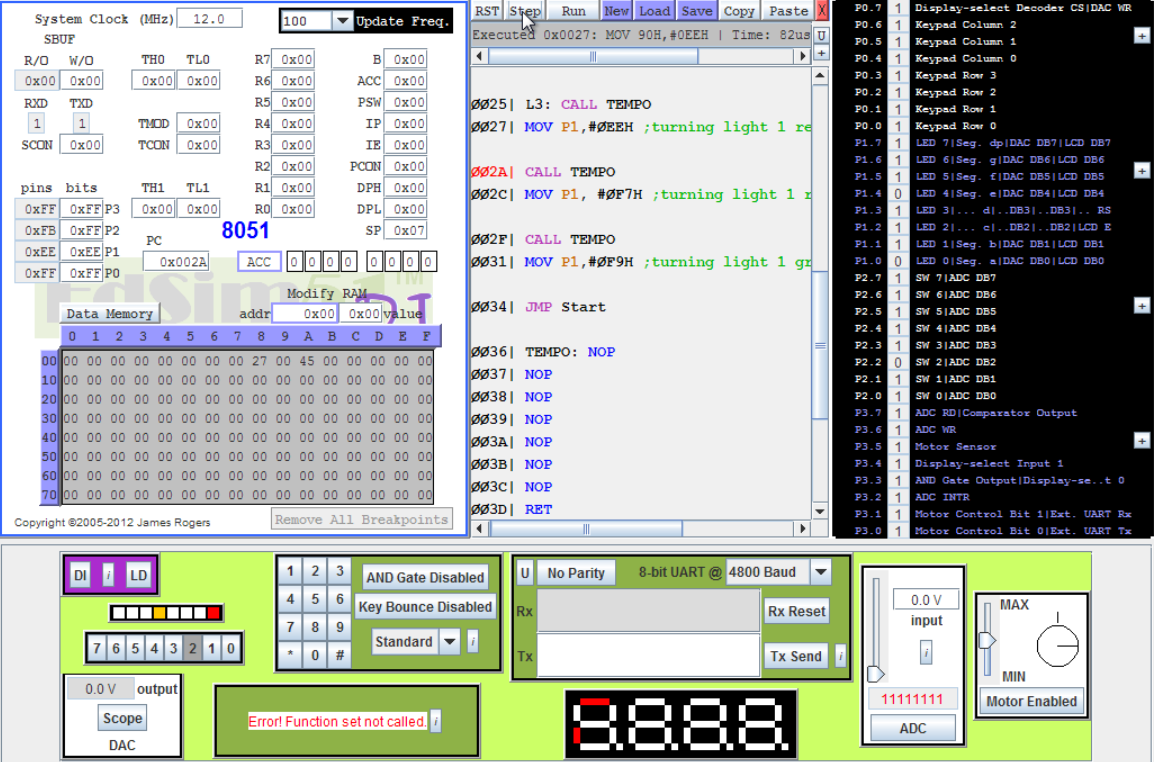


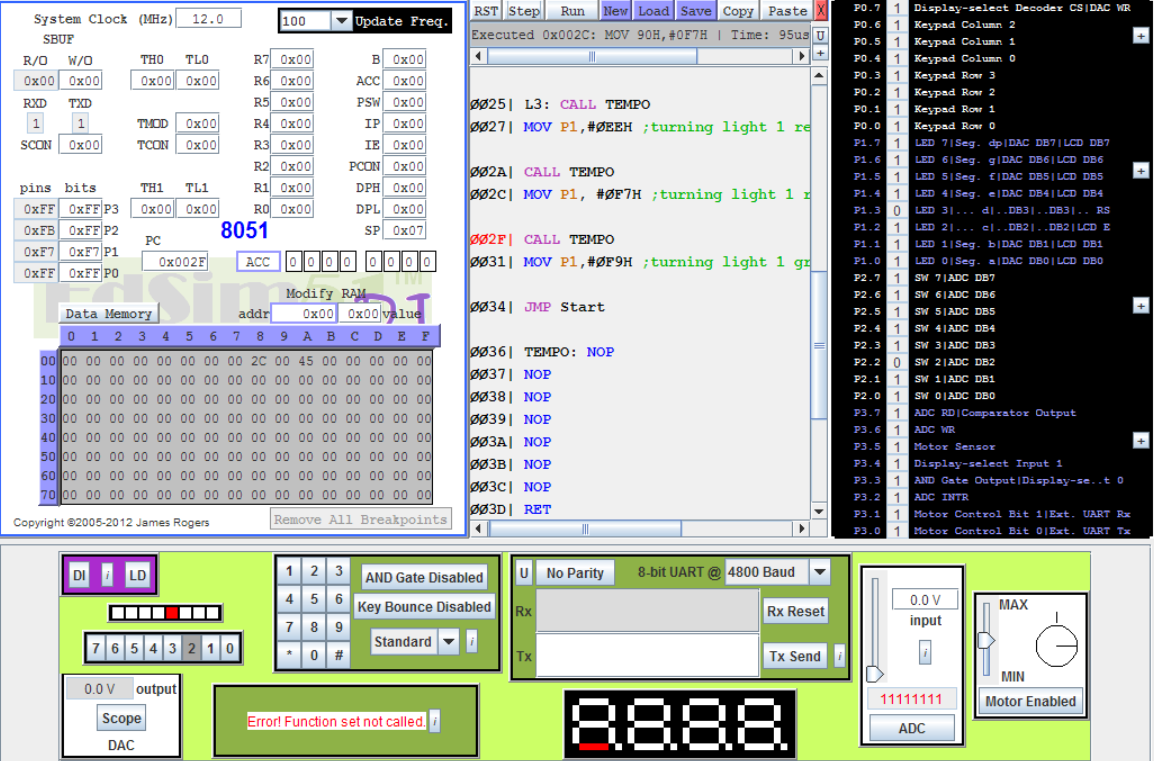




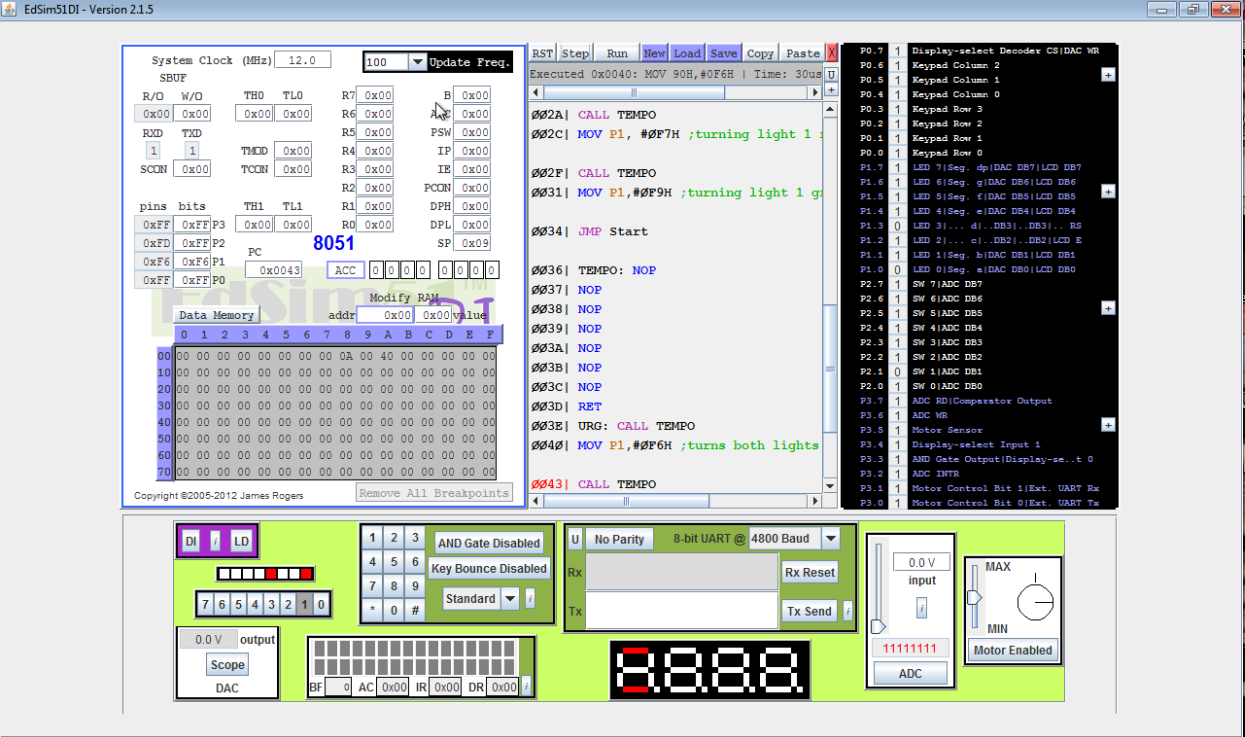
P2.0=1:

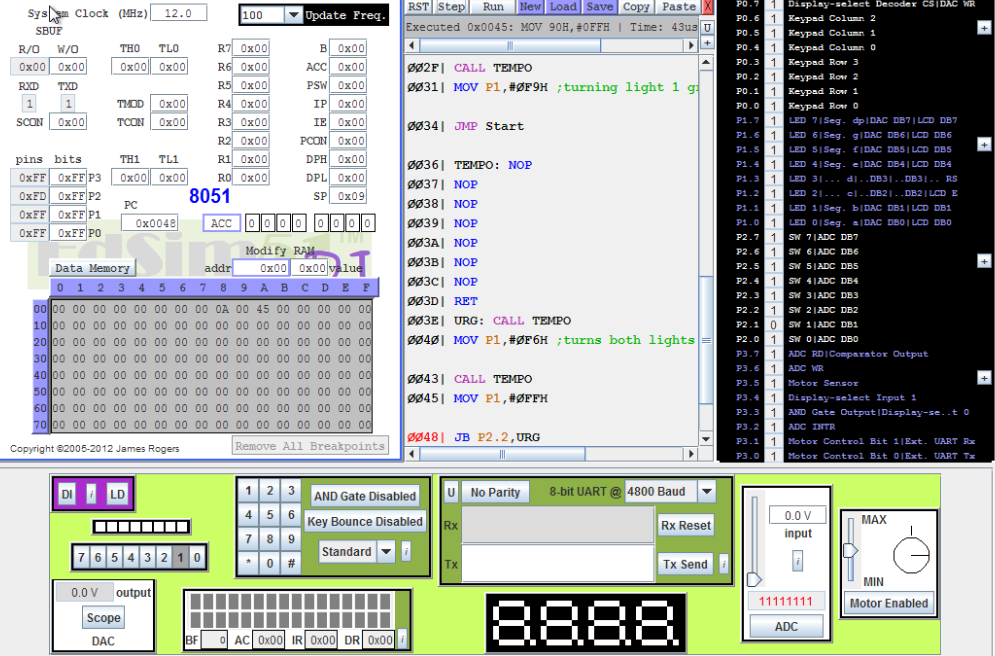






P2.2=1:





**Conclusion:**

This lab taught the importance of subroutines and calls, and how using them can greatly reduce the amount of coding necessary. There are many functions that will be have to be called constantly, in this case almost every other line, such as the Tempo function. Once there is input information, conditional statements will need to be implemented using the JMP function. For continuously looping functions such as this one, there will need to be jumps back to the start.

**Code:**

Start: MOV P1,#0F3H ; p1.2=0 and p1.3=0, L1=Green and L2=Red

CALL TEMPO ; Call delay subroutine

JNB P2.2, Nourg ; If 2.2=0, urgent switch not on, jump to Nourg  
call URG ; If 2.1=1, urgent switch is on, jump to URG

Nourg: JB P2.1,FJ ; If P2.1=1, jump to FJ to change Light 1 to Yellow

JMP Start ; Jump back to start to keep light 1 Green and Light 2 Red

FJ: CALL TEMPO  
MOV P1,#0F5H ; Set P1.1=0, P1.3=0, Light 1 is Yellow, Light2 is Red

CALL TEMPO

MOV P1,#0F6H ; Set P1.0=0, P1.3=0, Light 1 is Red, Light 2 is red

CALL TEMPO

L2: MOV P1,#0DEH ; Set P1.0=0, P1.3=0, Light 1 = Red, Light 2 = Green

CALL TEMPO

JB P2.0, L3 ; Jump to next step if P2.0 = 1, if 0 keep Light 1 Red and Light 2 Green  
CALL L2

L3: CALL TEMPO  
MOV P1,#0EEH ; P1.0=0, P1.4=0, Light 1 = Red and Light 2 = Yellow

CALL TEMPO  
MOV P1, #0F7H ; P1.0=0, P1.3=0, Light 1 = Red and Light 2 = Red

CALL TEMPO  
MOV P1,#0F9H ; P1.2=0, P1.3=0, Light 1 = Green and Light 2 = Red

JMP Start ; Go back to beginning of cycle

TEMPO: NOP ; Time delay routine  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
RET

URG: CALL TEMPO  
MOV P1,#0F6H ;P1.0=0, P1.3=0, Turns both lights red

CALL TEMPO  
MOV P1,#0FFH ; All bits set to 1, Turns both lights off

JB P2.2,URG ; Jump back to beginning if P2.2 is still set  
RET ; If P2.2 is no longer set, return to previous line

End