Project One

TRAFFIC LIGHT

Julie Kim | CECS 347 | March 3, 2017

Introduction:

This project design a simbolic traffic light by interfacing LED and switches on breadboad with microcontroller used in TM4C123. TM4C123 give access to General Purpose I/O ports, PLL, and Systick to generated time traffic light controlled by switches. There are two sides of traffic light, green, yellow and read, each on East and North and two light for pedestrain. The flow of the simbolic traffic is controlle by Moor state machine to ensure each member of the three direction have a fair chance to cross the street without heating each other.

Operation:

Three push buttons are used to indicate the present of cars and pedestrains. The first two button indicate the present of car coming from either East or North. When car travel from Northe side, North greed LED lights up, East and Pedestrain LED are on to give way. Pedestrain and East LED functions in the same way when either of its push button is pushed. When another push button is push while the previous LED is green, this previous green LED will complete 6 seconds before it is off and the yellow is on for the next 2 second before it off and it red LED is on. For pedestrain green LED is on for 8 second; the red LED will flash for interval of half second when detecting the push button from the coming of car either from East or West. The green LED is than on for East or North side only when the pedestrain light is solid red.

One push button is pushed to change the direction either from East to North or from East to Pedestrain. When two button are pushed at the same time, the flow of the traffic take turn between the two direction as long as the two button are pressed. Three button pushed to create flow of three direction of traffic flow in a round robbin manner by giving each direction, East, North and Pedestrain, a fair chance to cross the street. The cycle flow as triangle as long as the the three button are pressed.

Theory:

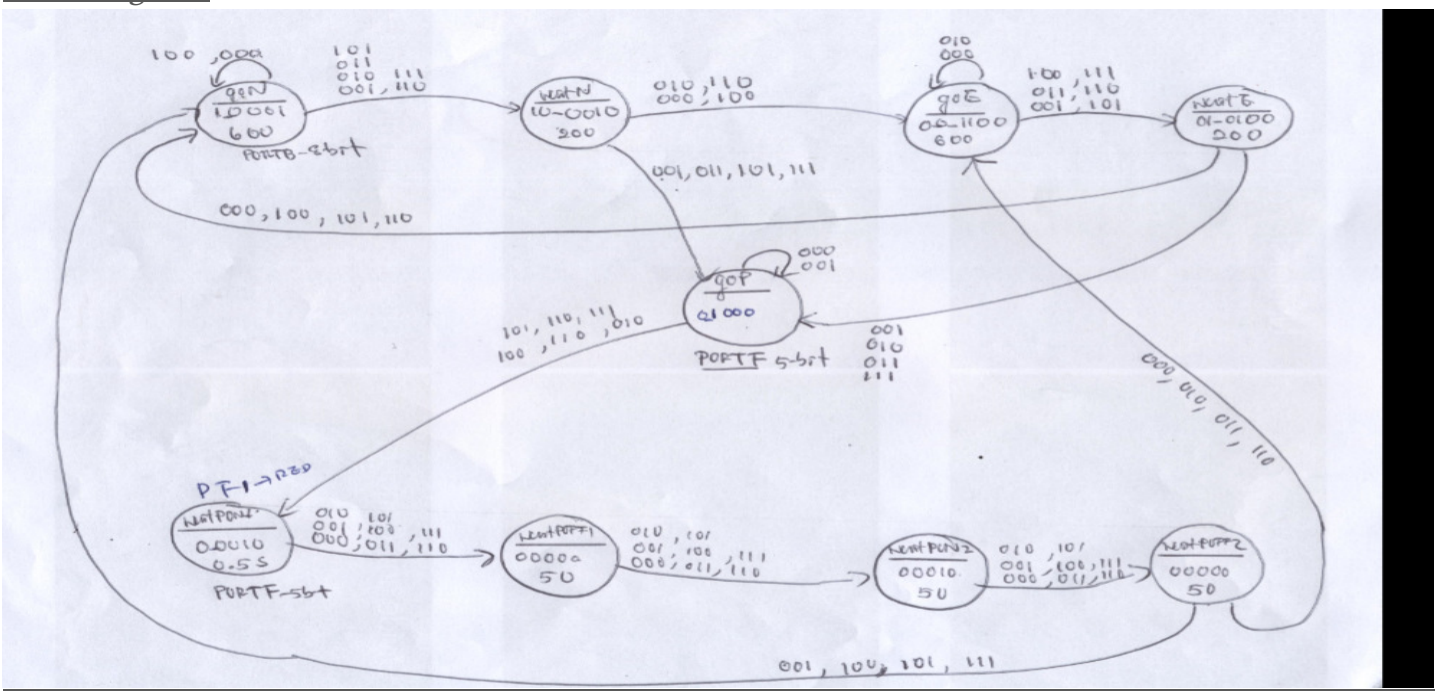
Tiva C TM4C123 microcontroller gives access to input and output port known as GPIO port which can be used as input as well as output port. Sensor is taken in through port E, push button, to indicate the coming of car from East, North or from Pedestrain. The output port, port B and F turn on the LED. PLL and Systick provided by the microcontroller allows the flexibility of chosing desired frequency. 80 MHz is chosed for this project. This frequency is used to generate real time of delay when switching from one output port to another. The

functionality of TM4C123 provide convenient way to design such simple implementation for educationaly purpose.

State Machine Table:

	000	001	010	011	100	101	110	111
goN	goN	waitN	waitN	waitN	goN	waitN	waitN	waitN
waitN	goE	goP	goE	goP	goE	goP	goE	goP
goE	goE	waitE	goE	waitE	waitE	waitE	waitE	waitE
waitE	goN	goP	goP	goP	goN	goN	goN	goP
goP	goP	goP	waitP_On1	waitP_On1	waitP_On1	waitP_On1	waitP_On1	waitP_On1
waitP_On1	waitP_Off1	waitP_Off1	waitP_Off1	waitP_Off1	waitP_Off1	waitP_Off1	waitP_Off1	waitP_Ofi
waitP_Off1	waitP_On2	waitP_On2	waitP_On2	waitP_On2	waitP_On2	waitP_On2	waitP_On2	waitP_On2
waitP_On2	waitP_Off2	waitP_Off2	waitP_Off2	waitP_Off2	waitP_Off2	waitP_Off2	waitP_Off2	waitP_Off2
waitP_Off2	goE	goN	goE	goE	goN	goN	goE	goN

State Diagram:



Software Code:

Lab3.c | SysTick.c

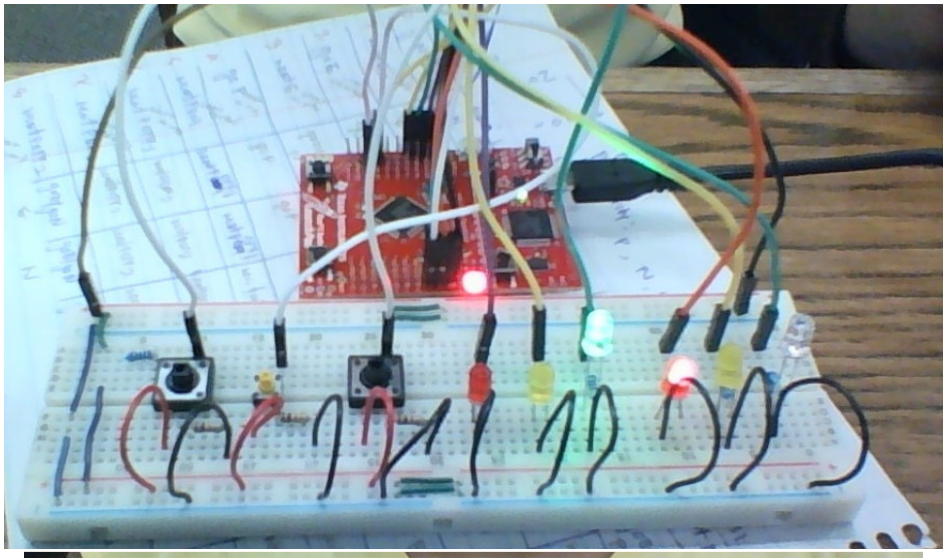
```
#include "SysTick.h"
#include "PLL.h"

#define GPIO_PORTB_DATA_R       (*((volatile unsigned long *)0x400053FC))
#define LIGHT_B                 (*((volatile unsigned long *)0x400053FC))
#define GPIO_PORTB_DIR_R        (*((volatile unsigned long *)0x40005400))
#define GPIO_PORTB_AFSEL_R      (*((volatile unsigned long *)0x40005420))
#define GPIO_PORTB_PUR_R        (*((volatile unsigned long *)0x40005510))
#define GPIO_PORTB_DEN_R        (*((volatile unsigned long *)0x4000551C))
#define GPIO_PORTF_LOCK_R       (*((volatile unsigned long *)0x40025520))
#define GPIO_PORTB_CR_R         (*((volatile unsigned long *)0x40005524))
#define GPIO_PORTB_AMSEL_R      (*((volatile unsigned long *)0x40005528))
#define GPIO_PORTB_PCTL_R       (*((volatile unsigned long *)0x4000552C))
#define SYSCTL_RCGC2_R          (*((volatile unsigned long *)0x400FE108))

#define GPIO_PORTE_DATA_R       (*((volatile unsigned long *)0x400243FC))
#define SENSOR                  (*((volatile unsigned long *)0x400243FC))
#define GPIO_PORTE_DIR_R        (*((volatile unsigned long *)0x40024400))
#define GPIO_PORTE_AFSEL_R      (*((volatile unsigned long *)0x40024420))
#define GPIO_PORTE_PUR_R        (*((volatile unsigned long *)0x40024510))
#define GPIO_PORTE_DEN_R        (*((volatile unsigned long *)0x4002451C))
#define GPIO_PORTF_LOCK_R       (*((volatile unsigned long *)0x40025520))
#define GPIO_PORTE_CR_R         (*((volatile unsigned long *)0x40024524))
#define GPIO_PORTE_AMSEL_R      (*((volatile unsigned long *)0x40024528))
#define GPIO_PORTE_PCTL_R       (*((volatile unsigned long *)0x4002452C))

#define GPIO_PORTF_DATA_R       (*((volatile unsigned long *)0x400253FC))
#define LIGHT_F                 (*((volatile unsigned long *)0x400253FC))
#define GPIO_PORTF_DIR_R        (*((volatile unsigned long *)0x40025400))
#define GPIO_PORTF_AFSEL_R      (*((volatile unsigned long *)0x40025420))
#define GPIO_PORTF_PUR_R        (*((volatile unsigned long *)0x40025510))
#define GPIO_PORTF_DEN_R        (*((volatile unsigned long *)0x4002551C))
#define GPIO_PORTF_LOCK_R       (*((volatile unsigned long *)0x40025520))
#define GPIO_PORTF_CR_R         (*((volatile unsigned long *)0x40025524))
#define GPIO_PORTF_AMSEL_R      (*((volatile unsigned long *)0x40025528))
#define GPIO_PORTF_PCTL_R       (*((volatile unsigned long *)0x4002552C))

#define goN          0
#define waitN        1
#define goE          2
#define waitE        3
#define goP          4
#define wP_On1       5
#define wP_On2       7
#define wP_Off1      6
#define wP_Off2      8
```

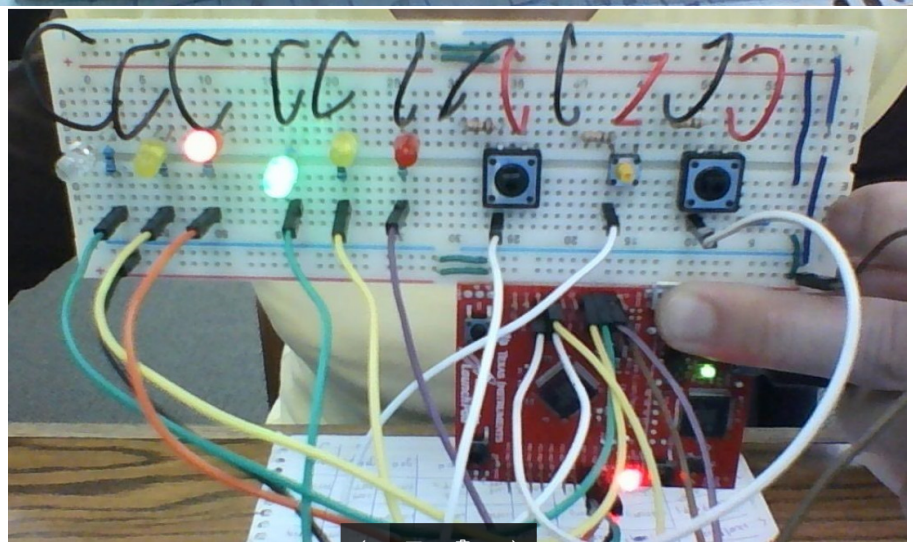
```
47
48  // Linked data structure
49 struct State {
50    unsigned long Out_Light;
51    unsigned long Out_p;
52    unsigned long Time;
53    unsigned long Next[8];
54  };
55  typedef const struct State STyp;
56 STyp FSM[9]={
57                      //000      001       010       011       100       101       110       111
58   {0x21, 0x02, 600,{goN,      waitN,    waitN,    waitN,    goN,      waitN,    waitN,    waitN}},
59   {0x22, 0x02, 200,{goE,      goP,      goE,      goP,      goE,      goP,      goE,      goP}},
60   {0x0C, 0x02, 600,{goE,      waitE,    goE,      waitE,    waitE,    waitE,    waitE,    waitE}},
61   {0x14, 0x02, 200,{goN,      goP,      goP,      goP,      goN,      goN,      goN,      goP}},
62   {0x24, 0x08, 600,{goP,      goP,      wP_On1,   wP_On1,   wP_On1,   wP_On1,   wP_On1,   wP_On1}},
63   {0x24, 0x02, 50, {wP_Off1,  wP_Off1,  wP_Off1,  wP_Off1,  wP_Off1,  wP_Off1,  wP_Off1,  wP_Off1}},
64   {0x24, 0x00, 50, {wP_On2,   wP_On2,   wP_On2,   wP_On2,   wP_On2,   wP_On2,   wP_On2,   wP_On2}},
65   {0x24, 0x02, 50, {wP_Off2,  wP_Off2,  wP_Off2,  wP_Off2,  wP_Off2,  wP_Off2,  wP_Off2,  wP_Off2}},
66   {0x24, 0x00, 50, {goE,      goN,      goE,      goE,      goN,      goN,      goE,      goN}}
67  };
68  void PortB_Init(void);
69  void PortE_Init(void);
70  void PortF_Init(void);
71  unsigned long Input;
72  unsigned long S;
73
```

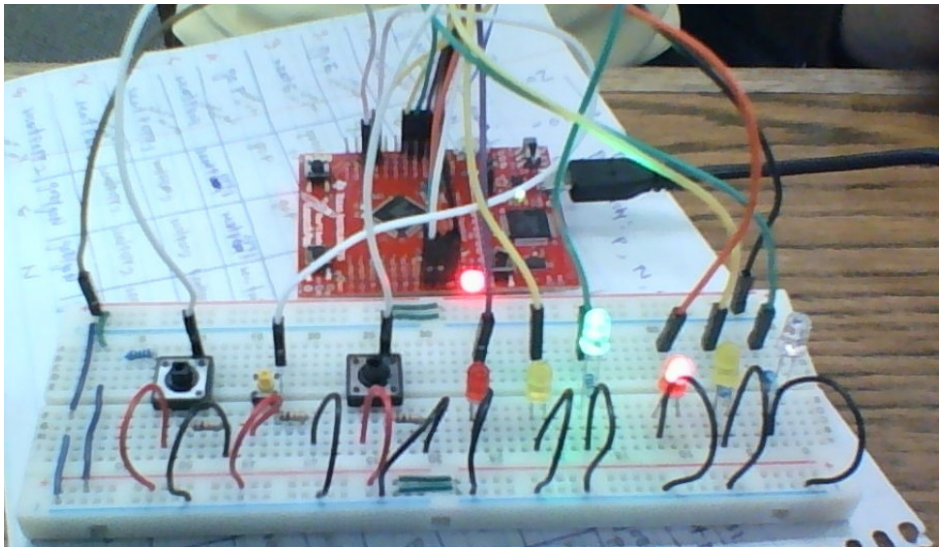
```
73
74 int main(void){
75    volatile unsigned long delay;
76  //  unsigned long S;                  // index to the current state
77  //  unsigned long Input;
78    PortB_Init();
79    PortE_Init();
80    PortF_Init();
81    PLL_Init();                       // set system clock to 80 MHz
82    SysTick_Init();                   // initialize SysTick timer
83    S = goN;
84    while(1){
85      LIGHT_B = FSM[S].Out_Light;     // TRAFFIC
86      LIGHT_F = FSM[S].Out_p;         // PEDESTRAINT LIGHT
87      SysTick_Wait10ms(FSM[S].Time);
88      Input = SENSOR;                 // read sensors
89      S = FSM[S].Next[Input];
90    }
91 }
92 void PortB_Init(void){
93    volatile unsigned long delay;
94    SYSCTL_RCGC2_R |= 0x00000002;     // 1) B clock
95    delay = SYSCTL_RCGC2_R;           // delay
96    GPIO_PORTB_CR_R = 0x3F;           // allow changes to PB5-0
97    GPIO_PORTB_AMSEL_R = 0x00;        // 3) disable analog function
98    GPIO_PORTB_PCTL_R = 0x00000000;   // 4) GPIO clear bit PCTL
99    GPIO_PORTB_DIR_R = 0x3F;          // 5) PB5 - 0 output
100   GPIO_PORTB_AFSEL_R = 0x00;        // 6) no alternate function
101   GPIO_PORTB_DEN_R = 0x3F;          // 7) enable digital pins PF4-PF0
```

```
101     GPIO_PORTB_DEN_R = 0x3F;          // 7) enable digital pins PF4-PF0
102   }
103   void PortE_Init(void) {
104     volatile unsigned long delay;
105     SYSCTL_RCGC2_R |= 0x00000010;      // 1) E clock
106     delay = SYSCTL_RCGC2_R;            // delay
107     GPIO_PORTE_CR_R = 0x07;            // allow changes to PE2-0
108     GPIO_PORTE_AMSEL_R = 0x00;         // 3) disable analog function
109     GPIO_PORTE_DIR_R = 0x00;           // 5) PE as input
110     GPIO_PORTE_AFSEL_R = 0x00;         // 6) no alternate function
111     GPIO_PORTE_DEN_R = 0x07;           // 7) enable digital pins PE2-PE0
112   }
113   void PortF_Init(void) {
114     SYSCTL_RCGC2_R |= 0x00000020;             // 1) F clock
115     GPIO_PORTF_LOCK_R = 0x4C4F434B;           // 2) unlock PortF PF0
116     GPIO_PORTF_CR_R = 0x1F;                   // allow changes to PF4-0
117     GPIO_PORTF_AMSEL_R = 0x00;                // 3) disable analog function
118     GPIO_PORTF_PCTL_R = 0x00000000;           // 4) GPIO clear bit PCTL
119     GPIO_PORTF_DIR_R = 0x0E;                  // 5) PF4,PF0 input, PF3,PF2,PF1 output
120     GPIO_PORTF_AFSEL_R = 0x00;                // 6) no alternate function
121     GPIO_PORTF_PUR_R = 0x11;                  // enable pullup resistors on PF4,PF0
122     GPIO_PORTF_DEN_R = 0x1F;                  // 7) enable digital pins PF4-PF0
123   }
124
```

Picture of Hardware Connection:







Conclusion:

The project has been very educational for student to learn how to interface C program software to hardware by the provided functionality of the TM4C123.

Hardware Schematic: