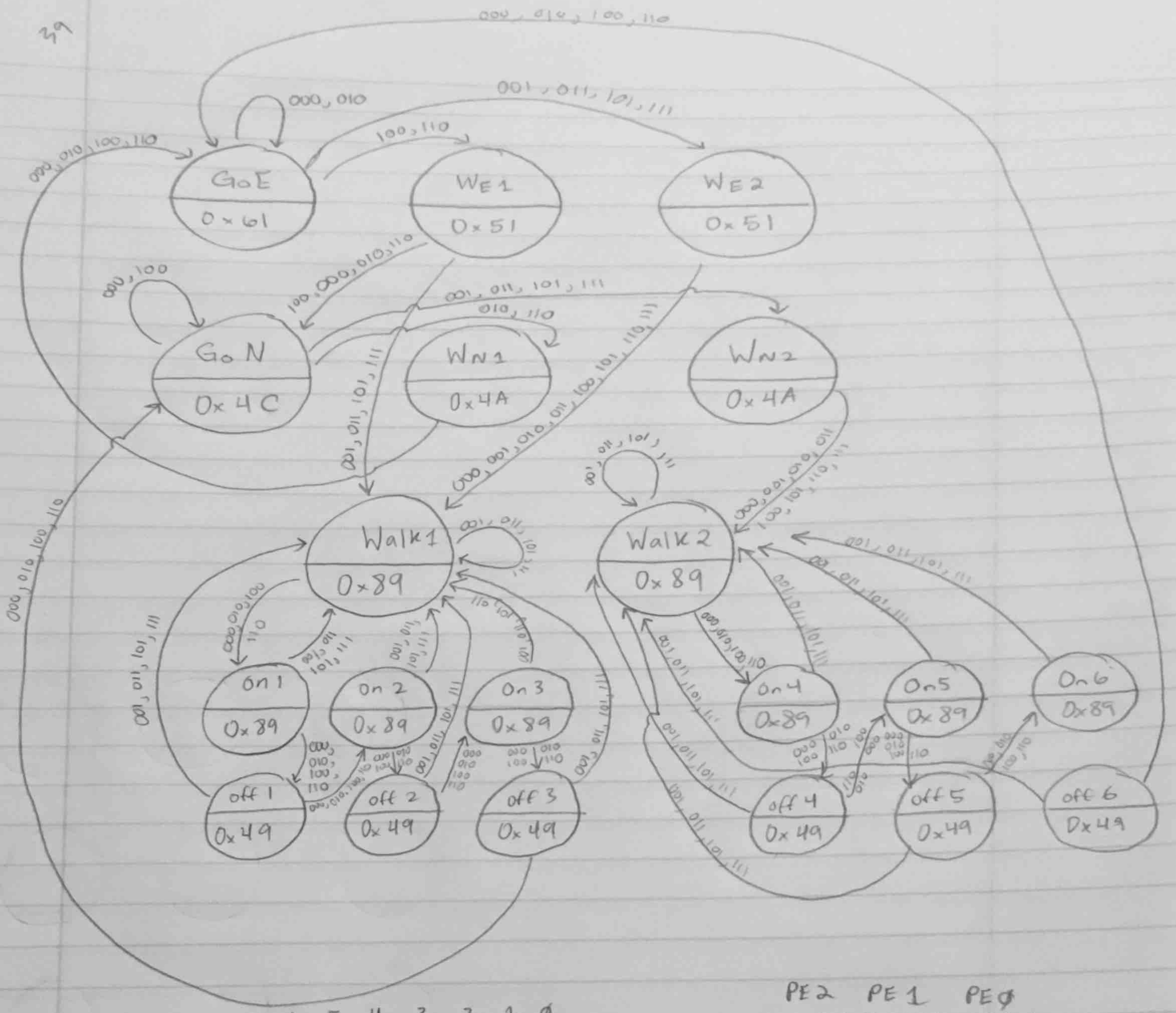


7	6	5	4	3	2	1	Ø
GW	YW	RW	GS	YS	RS	X	X

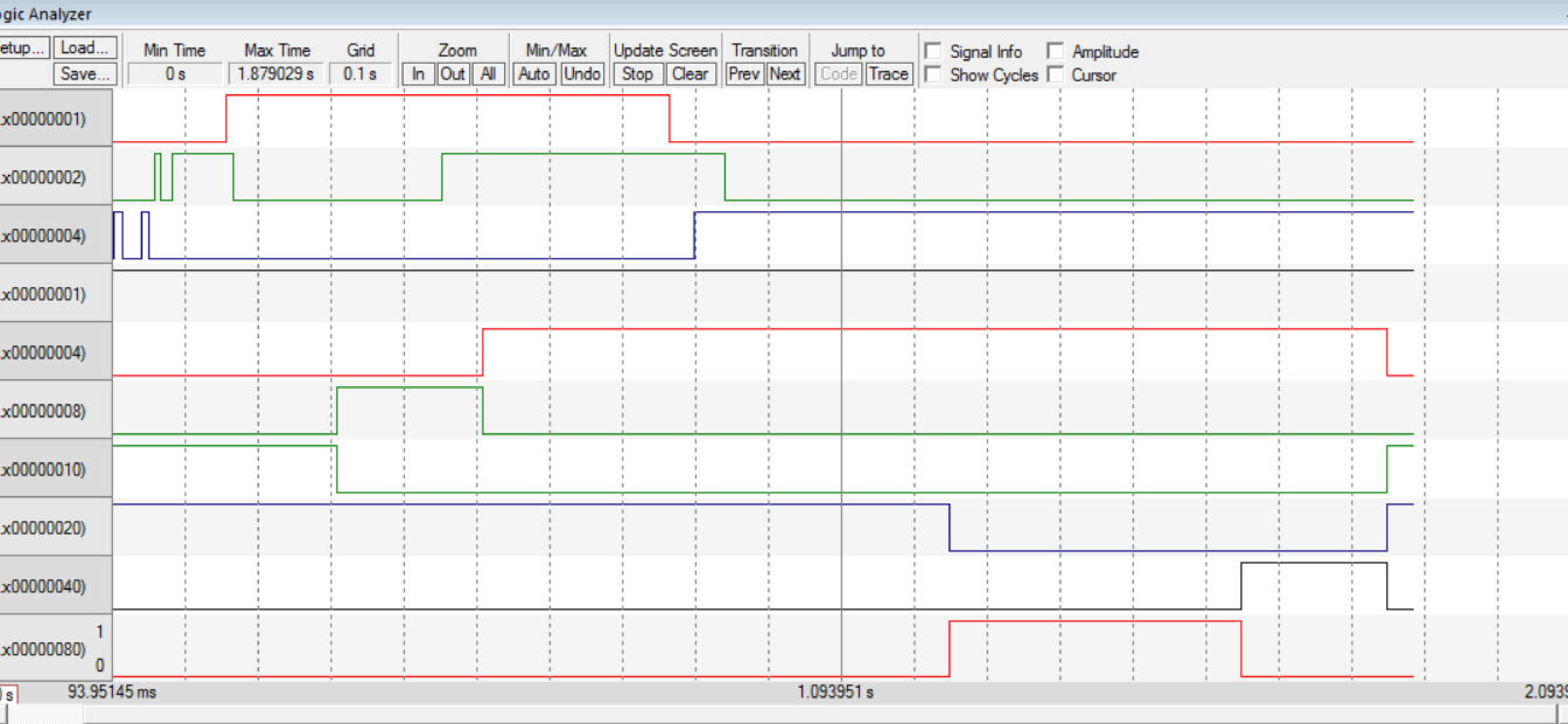
PORT A

Input

PE2	PE1	PEØ
South	West	Walk

Output

Walk	DW	GW	YW	RW	GS	YS	RS
PF3	PF1	PA7	PA6	PA5	PA4	PA3	PA2

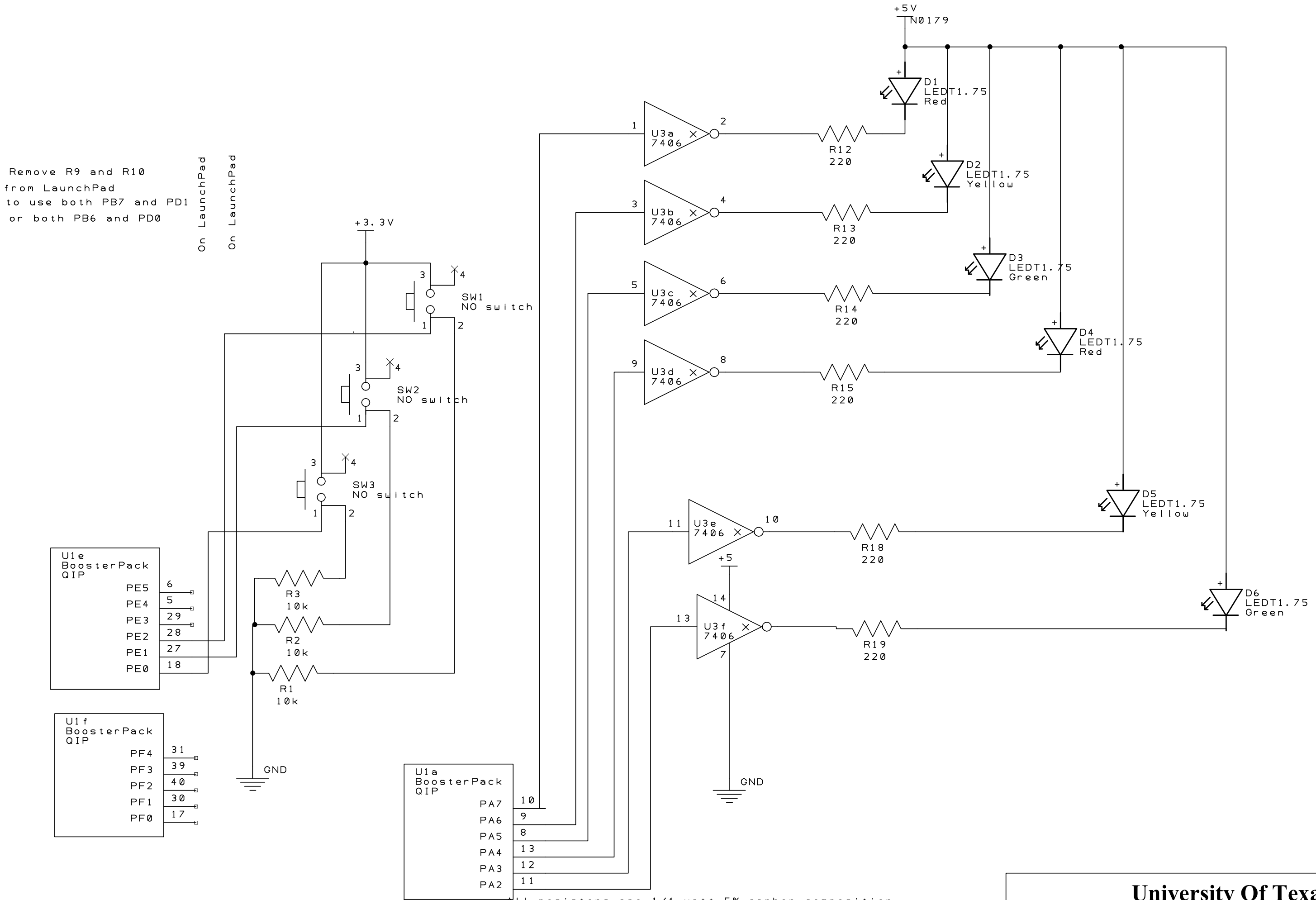


Min Time 0 s

Max Time 1.879029 s

Grid 0.1 s

Remove R9 and R10
from LaunchPad
to use both PB7 and PD1
or both PB6 and PD0



All resistors are 1/4 watt 5% carbon composition
C1 is ceramic Z5U
Switches are www.BGMicro.com SWT1043
Red LEDs. T1 3/4. 20mA Digikey 160-1087-ND
Yellow LEDs. T1 3/4. 20mA Digikey 160-1088-ND
Green LEDs. T1 3/4. 20mA Digikey 160-1089-ND
Slide pot. Bourns SSHA20B20300 www.AllElectronics.comSP-20K

University Of Texas At Austin

Schematic Name: EK-TM4C123GXL

Name(s): Kaela Todd & Megan Cooper

Date: March 9, 2016 **Semester:** Spring 2016

A1
frame

12

Out =

PF		PA					
Walk	DW	GW	YW	RW	GS	YS	RS
PF3	PF1	7	6	5	4	3	2

PE
Inputs

S	West	Walk
~~0~~	~~1~~	~~2~~

1010
$A

INPUTS PORTE 2 1 0

	Output	Time	000	001	010	011	100	101	110	111
GoE	0x61	20	GoE	WE2	GoE	WE2	WE1	WE2	WE1	WE2
WE1	0x51	10	GoN	Walk1	GoN	Walk 1	GoN	Walk1	GoN	Walk1
Walk → WE2	0x51	10	Walk1	Walk1	Walk1	Walk1	Walk1	Walk1	Walk1	Walk1
GoN	0x4C	20	GoN	WN2	WN1	WN2	GoN	WN2	WN1	WN2
WN1	0x4A	10	GoE	Walk2	GoE	Walk2	GoE	Walk2	GoE	Walk2
WN2	0x4A	10	Walt 2	Walk2	Walk2	Walk2	Walk 2	Walk2	Walk 2	Walk2
Walk1	0x89	20	On1	Walk1	On1	Walk1	On1	Walk1	On1	Walk1
Walk2	0x89	20	On4	Walk 2	On4	Walk2	On 4	Walk 2	On 4	Walk2
On1	0x89	2	Off1	Walk1	Off1	Walk1	Off 1	Walk1	Off 1	Walk1
Off1	0x49	2	On2	Walk1	On2	Walk1	On2	Walk 1	On2	Walk 1
On2	0x89	2	Off2	Walk1	Off2	Walk1	Off 2	Walk 1	Off2	Walk 1
Off2	0x49	2	On3	Walk1	On3	Walk 1	On3	Walk 1	On3	Walk 1
On3	0x89	2	Off3	Walk1	Off3	Walk1	Off3	Walk1	Off3	Walk 1
Off3	0x49	2	GoN	Walk1	GoN	Walk 1	GoN	Walk1	GoN	Walk 1
On4	0x89	2	Off4	Walk2	Off4	Walk 2	Off4	Walk2	Off4	Walk 2
Off 4	0x49	2	On5	Walk2	On5	Walk 2	On5	Walk 2	On5	Walk 2
On5	0x89	2	Off5	Walk2	Off5	Walk 2	Off5	Walk 2	Off5	Walk 2
Off5	0x49	2	On6	Walk2	On6	Walk 2	On6	Walk 2	On6	Walk 2
On6	0x89	2	Off6	Walk2	Off6	Walk 2	Off6	Walk 2	Off6	Walk 2
Off6	0x49	2	GoE	Walk2	GoE	Walk 2	GoE	Walk 2	GoE	Walk 2

```
// ***** 0. Documentation Section *****
// TableTrafficLight.c for (Lab 10 edX), Lab 5 EE319K
// Runs on LM4F120/TM4C123
// Program written by: Kaela Todd and Megan Cooper
// Date Created: 1/24/2015
// Last Modified: 3/2/2016
// Section Wed 4-5pm
// Lab number: 5
// Hardware connections
// Hardware Specifications
// east/west red light connected to PB5
// east/west yellow light connected to PB4
// east/west green light connected to PB3
// north/south facing red light connected to PB2
// north/south facing yellow light connected to PB1
// north/south facing green light connected to PB0
// pedestrian detector connected to PE2 (1=pedestrian present)
// north/south car detector connected to PE1 (1=car present)
// east/west car detector connected to PE0 (1=car present)
// "walk" light connected to PF3 (built-in green LED)
// "don't walk" light connected to PF1 (built-in red LED)

// ***** 1. Pre-processor Directives Section *****
#include <stdint.h>
#include "TExaS.h"
#include "tm4c123gh6pm.h"
#include "SysTick.h"


#define GPIO_PORTE_DIR_R        (*((volatile uint32_t *)0x40024400))
#define GPIO_PORTE_AFSEL_R      (*((volatile uint32_t *)0x40024420))
#define GPIO_PORTE_DEN_R        (*((volatile uint32_t *)0x4002451C))
#define GPIO_PORTE_AMSEL_R      (*((volatile uint32_t *)0x40024528))
#define GPIO_PORTE_PCTL_R       (*((volatile uint32_t *)0x4002452C))

#define SYSCTL_RCGC2_R          (*((volatile uint32_t *)0x400FE108))
#define SYSCTL_RCGC2_GPIOE      0x00000010  // port E Clock Gating Control
#define SYSCTL_RCGC2_GPIOB      0x00000002  // port B Clock Gating Control

#define GPIO_PORTA_DATA_R       (*((volatile uint32_t *)0x400043FC))
#define GPIO_PORTA_DIR_R        (*((volatile uint32_t *)0x40004400))
#define GPIO_PORTA_AFSEL_R      (*((volatile uint32_t *)0x40004420))
#define GPIO_PORTA_DEN_R        (*((volatile uint32_t *)0x4000451C))
#define GPIO_PORTA_AMSEL_R      (*((volatile uint32_t *)0x40004528))

#define GPIO_PORTF_DATA_R       (*((volatile uint32_t *)0x400253FC))
#define GPIO_PORTF_DIR_R        (*((volatile uint32_t *)0x40025400))
#define GPIO_PORTF_AFSEL_R      (*((volatile uint32_t *)0x40025420))
#define GPIO_PORTF_DEN_R        (*((volatile uint32_t *)0x4002551C))
#define GPIO_PORTF_AMSEL_R      (*((volatile uint32_t *)0x40025528))

#define SYSCTL_RCGCGPIO_R       (*((volatile uint32_t *)0x400FE608))
#define PA72                    (*((volatile uint32_t *)0x400043F0))
#define PF31                    (*((volatile uint32_t *)0x40025028))
#define PF1                     (*((volatile uint32_t *)0x40025008))
#define PF3                     (*((volatile uint32_t *)0x40025020))



// ***** 2. Global Declarations Section *****

// FUNCTION PROTOTYPES: Each subroutine defined
void DisableInterrupts(void); // Disable interrupts
void EnableInterrupts(void);  // Enable interrupts
void cDelay(unsigned short count);
extern void Delay(void);

// ***** 3. Subroutines Section *****
void cDelay(unsigned short count){
    while(count){
      Delay();
      count--;
    }
}

// Linked data structure
struct State {
  unsigned long Out;
  unsigned long Time;
  unsigned long Next[8];
};
typedef const struct State STyp;
#define goN     0
#define waitN1  1
#define waitN2  2
#define goE     3
#define waitE1  4
#define waitE2  5
#define walk1   6
#define walk2   7
#define on1     8
#define on2     9
#define on3     10
#define on4     11
#define on5     12
#define on6     13
#define off1    14
#define off2    15
#define off3    16
#define off4    17
#define off5    18
#define off6    19
STyp FSM[20] = {
 {0x4C, 20, {goN,waitN2,waitN1,waitN2,goN,waitN2,waitN1,waitN2}},  //20
 {0x4A, 10, {goE,walk2,goE,walk2,goE,walk2,goE,walk2}},            //10
 {0x4A, 10, {walk2,walk2,walk2,walk2,walk2,walk2,walk2,walk2}},    //10
 {0x61, 20, {goE,waitE2,goE,waitE2,waitE1,waitE2,waitE1,waitE2}},  //20
 {0x51, 10, {goN,walk1,goN,walk1,goN,walk1,goN,walk1}},            //10
 {0x51, 10, {walk1,walk1,walk1,walk1,walk1,walk1,walk1,walk1}},    //10
 {0x89, 20, {on1,walk1,on1,walk1,on1,walk1,on1,walk1}},            //20
 {0x89, 20, {on4,walk2,on4,walk2,on4,walk2,on4,walk2}},            //20
 {0x89,  2, {off1,off1,off1,off1,off1,off1,off1,off1}},
 {0x89,  2, {off2,off2,off2,off2,off2,off2,off2,off2}},
 {0x89,  2, {off3,off3,off3,off3,off3,off3,off3,off3}},
 {0x89,  2, {off4,off4,off4,off4,off4,off4,off4,off4}},
 {0x89,  2, {off5,off5,off5,off5,off5,off5,off5,off5}},
 {0x89,  2, {off6,off6,off6,off6,off6,off6,off6,off6}},
 {0x49,  2, {on2,on2,on2,on2,on2,on2,on2,on2}},
 {0x49,  2, {on3,on3,on3,on3,on3,on3,on3,on3}},
 {0x49,  2, {goN,walk1,goN,walk1,goN,walk1,goN,walk1}},
 {0x49,  2, {on5,on5,on5,on5,on5,on5,on5,on5}},
 {0x49,  2, {on6,on6,on6,on6,on6,on6,on6,on6}},
 {0x49,  2, {goE,walk2,goE,walk2,goE,walk2,goE,walk2}}
 };


unsigned long S;      // index to the current state
unsigned long Input;
uint16_t delay;

int main(void){

  SysTick_Init();

  TExaS_Init(SW_PIN_PE210, LED_PIN_PB543210); // activate grader and set system clock to 80 MHz

  volatile unsigned long delay;
  //SYSCTL_RCGC2_R |= 0x31;           // 1) A E F
  //delay = SYSCTL_RCGC2_R;

  //Turn on clock for Ports A, E, and F
  SYSCTL_RCGCGPIO_R   |= 0x31;
  delay = SYSCTL_RCGCGPIO_R;
```

```c
    //Initialize PortE
    GPIO_PORTE_DIR_R    &= ~0x07;
    GPIO_PORTE_AFSEL_R  &= 0x00;
    GPIO_PORTE_DEN_R    |= 0x07;
    //GPIO_PORTE_AMSEL_R  &= 0x00;
    //Initialize PortA
    GPIO_PORTA_DIR_R    |= 0xFC;
    GPIO_PORTA_AFSEL_R  &= ~0xFC;
    GPIO_PORTA_DEN_R    |= 0xFC;
    //GPIO_PORTA_AMSEL_R  &= 0x00;
    //Initialize PortF
    GPIO_PORTF_DIR_R    |= 0x0A;
    GPIO_PORTF_AFSEL_R  &= ~0x0A;
    GPIO_PORTF_DEN_R    |= 0x0A;
    //GPIO_PORTF_AMSEL_R  &= 0x00;

    S = goN;


    EnableInterrupts();
    while(1){
      PA72    =  ((FSM[S].Out & 0x3F)<<2);             // Set Traffic LEDs
      PF3     =  ((FSM[S].Out & 0x80)>>4);             // Set Walk LEDs
      PF1     =  ((FSM[S].Out & 0x40)>>5);             // Set Don't Walk LEDs
      cDelay(FSM[S].Time);                              // Implement State Delay
      Input = (GPIO_PORTE_DATA_R & 0x07);              // Read Sensors
      S = FSM[S].Next[Input];
    }
  }


```

```
            AREA   |.text|, CODE, READONLY, ALIGN=2
            THUMB
            EXPORT Delay

Delay       PUSH {R8,R4}
            MOV  R8, #50000
wait1       SUBS R8, #1
            BNE  wait1
            MOV  R8, #50000
wait2       SUBS R8, #1
            BNE  wait2
            MOV  R8, #50000
wait3       SUBS R8, #1
            BNE  wait3
            MOV  R8, #50000
wait4       SUBS R8, #1
            BNE  wait4
            MOV  R8, #50000
wait5       SUBS R8, #1
            BNE  wait5
            MOV  R8, #50000
wait6       SUBS R8, #1
            BNE  wait6
            MOV  R8, #50000
wait7       SUBS R8, #1
            BNE  wait7
            MOV  R8, #50000
wait8       SUBS R8, #1
            BNE  wait8
            POP  {R8,R4}
            BX   LR

            ALIGN      ; make sure the end of this section is aligned
            END        ; end of file
```

```c
// SysTick.h
// Runs on LM4F120
// Provide functions that initialize the SysTick module, wait at least a
// designated number of clock cycles, and wait approximately a multiple
// of 10 milliseconds using busy wait.  After a power-on-reset, the
// LM4F120 gets its clock from the 16 MHz precision internal oscillator,
// which can vary by +/- 1% at room temperature and +/- 3% across all
// temperature ranges.  If you are using this module, you may need more
// precise timing, so it is assumed that you are using the PLL to set
// the system clock to 50 MHz.  This matters for the function
// SysTick_Wait10ms(), which will wait longer than 10 ms if the clock is
// slower.
// Daniel Valvano
// October 25, 2012

/* This example accompanies the book
   "Embedded Systems: Real Time Interfacing to Arm Cortex M Microcontrollers",
   ISBN: 978-1463590154, Jonathan Valvano, copyright (c) 2012
   Program 2.11, Section 2.6

 Copyright 2012 by Jonathan W. Valvano, valvano@mail.utexas.edu
    You may use, edit, run or distribute this file
    as long as the above copyright notice remains
 THIS SOFTWARE IS PROVIDED "AS IS".  NO WARRANTIES, WHETHER EXPRESS, IMPLIED
 OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.
 VALVANO SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL,
 OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
 For more information about my classes, my research, and my books, see
 http://users.ece.utexas.edu/~valvano/
 */


// Initialize SysTick with busy wait running at bus clock.
#define NVIC_ST_CTRL_R      (*((volatile unsigned long *)0xE000E010))
#define NVIC_ST_RELOAD_R    (*((volatile unsigned long *)0xE000E014))
#define NVIC_ST_CURRENT_R   (*((volatile unsigned long *)0xE000E018))
void SysTick_Init(void){
  NVIC_ST_CTRL_R = 0;               // disable SysTick during setup
  NVIC_ST_CTRL_R = 0x00000005;      // enable SysTick with core clock
}
// The delay parameter is in units of the 80 MHz core clock. (12.5 ns)
void SysTick_Wait(unsigned long delay){
  NVIC_ST_RELOAD_R = delay-1;  // number of counts to wait
  NVIC_ST_CURRENT_R = 0;       // any value written to CURRENT clears
  while((NVIC_ST_CTRL_R&0x00010000)==0){ // wait for count flag
  }
}
// 10000us equals 10ms
void SysTick_Wait10ms(unsigned long delay){
  unsigned long i;
  for(i=0; i<delay; i++){
    SysTick_Wait(800000);  // wait 10ms
  }
}


```