**Morgan State University**

Department of Electrical and Computer Engineering

EEGR 417 - Microprocessors Applications

Mr. Paterne Sissinto

**Project # 04: Traffic Light Controller**

***Traffic Light Controller Code***

/\*------------------------------------------------------------------------------------------------------------

Name: Seth Yeboah

Class: EEGR 417 Microprocessors Application

Project #1

11/3/2015

Traffic Light Controller

-------------------------------------------------------------------------------------------------------------\*/

//-----------------------------------------------------------------------------

// BIOS header files

// ----------------------------------------------------------------------------

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\inc\tm4c123gh6pm.h"

#include <xdc/std.h>

#include <xdc/runtime/System.h>

#include <ti/sysbios/BIOS.h>

#include <ti/sysbios/knl/Task.h>

#include <xdc/runtime/Log.h> //needed for any Log\_info() call

#include <xdc/cfg/global.h> //header file for statically defined objects/handles

//----------------------------------------------------------------------------

// TivaWare Header Files

//----------------------------------------------------------------------------

#include <stdbool.h>

#include <stdint.h>

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\inc/hw\_gpio.h"

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\inc/hw\_memmap.h"

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\inc/hw\_types.h"

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\driverlib/gpio.h"

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\driverlib/pin\_map.h"

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\driverlib/rom.h"

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\driverlib/rom\_map.h"

#include "C:\TI\TivaWare\_C\_Series-2.1.1.71\driverlib/sysctl.h"

#include "pinout.h"

//-----------------------------------------------------------------------------

// Globals

//-----------------------------------------------------------------------------

volatile unsigned long FallingEdges = 0;

//-----------------------------------------------------------------------------

// Prototypes

//-----------------------------------------------------------------------------

void PinoutSet(void);

void PortCInt\_Init(void);

void goNorth\_Init(void);

void goEast\_Init(void);

void GPIOPortC\_Handler(void);

void delay(int n);

void NormalSequence(void);

void goNorth(void);

void goEast(void);

//---------------------------------------------------------------------------

// main()

//---------------------------------------------------------------------------

void main(void)

{

PinoutSet();

PortCInt\_Init();

goNorth\_Init();

goEast\_Init();

BIOS\_start(); // start BIOS scheduler

}

void PinoutSet(void)

{

// Enable Peripheral Clocks

MAP\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOB);

MAP\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOD);

// Configure GPIO Inputs

MAP\_GPIOPinTypeGPIOInput(GPIO\_PORTB\_BASE, GPIO\_PIN\_4);

// Configure GPIO Outputs

MAP\_GPIOPinTypeGPIOOutput(GPIO\_PORTB\_BASE, GPIO\_PIN\_0);

MAP\_GPIOPinTypeGPIOOutput(GPIO\_PORTB\_BASE, GPIO\_PIN\_1);

MAP\_GPIOPinTypeGPIOOutput(GPIO\_PORTB\_BASE, GPIO\_PIN\_2);

MAP\_GPIOPinTypeGPIOOutput(GPIO\_PORTB\_BASE, GPIO\_PIN\_3);

MAP\_GPIOPinTypeGPIOOutput(GPIO\_PORTD\_BASE, GPIO\_PIN\_0);

MAP\_GPIOPinTypeGPIOOutput(GPIO\_PORTD\_BASE, GPIO\_PIN\_1);

MAP\_GPIOPinTypeGPIOOutput(GPIO\_PORTD\_BASE, GPIO\_PIN\_2);

}

void delay(int n)

{

SysCtlDelay(n);

}

void NormalSequence(void)

{

PinoutSet();

int i = 0;

while(1)

{

if(i == 0){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is RED

delay(30000000); // ~3seconds delay

i++;

}

if(i == 1){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_2, 4); // D is GREEN

delay(150000000); // ~15seconds delay

i++;

}

if(i == 2){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_1, 2); // D is YELLOW

delay(60000000); // ~6seconds delay

i++;

}

if(i == 3){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is RED

delay(30000000); //~3seconds delay

i++;

}

if(i == 4){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_2, 4); // B is GREEN

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is RED

delay(150000000);

i++;

}

if(i == 5){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_1, 2); // B is YELLOW

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is RED

delay(60000000);

}

i = 0;

}

}

//---------------------------------------------------------------------------

// GPIO\_portC\_init() for Pedestrian Crossing

//---------------------------------------------------------------------------

void PortCInt\_Init(void)

{

SYSCTL\_RCGC2\_R |= 0x00000004; // (a) activate port C

FallingEdges = 0; // (b) initialize the counter

GPIO\_PORTC\_DIR\_R &= ~0x10; // (c) make PC4 input

GPIO\_PORTC\_DEN\_R |=0x10; // ( ) enable digital pin

GPIO\_PORTC\_IS\_R &= ~0x10; // (d) PC4 is edge sensitive

GPIO\_PORTC\_IBE\_R &= ~0x10; // ( ) PC4 not both edge sensitive

GPIO\_PORTC\_IEV\_R &= ~0x10; // ( ) PC4 falling edge event

GPIO\_PORTC\_ICR\_R = 0x10; // (e) Clear flag4

GPIO\_PORTC\_IM\_R |= 0x10; // (f) ARM Interrupt on PC4

NVIC\_PRI0\_R = (NVIC\_PRI0\_R &0xFF00FFFF) |0x00A00000; // (g) Priority 5

NVIC\_EN0\_R = 4; // (h) Enable interrupt 2 in NVIC

}

void North\_South\_Pedestrian\_Interrupt(void)

{

PinoutSet();

int i;

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is RED

for(i = 0; i < 10; i++)

{

GPIOPinWrite(GPIO\_PORTB\_BASE,GPIO\_PIN\_3, 0);

delay(3000000);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_3, 8);

delay(3000000);

}

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_3, 0);

GPIO\_PORTC\_ICR\_R |= 0x11; /\* clear the interrupt flag before return \*/

}

//---------------------------------------------------------------------------

// GPIO\_PortA Initialization for North-South Traffic

//---------------------------------------------------------------------------

void goNorth\_Init(void)

{

SYSCTL\_RCGC2\_R |= 0x00000001; // (a) activate port C

FallingEdges = 0; // (b) initialize the counter

GPIO\_PORTA\_DIR\_R &= ~0x10; // (c) make PA4 input

GPIO\_PORTA\_DEN\_R |=0x10; // ( ) enable digital pin

GPIO\_PORTA\_IS\_R &= ~0x10; // (d) PA4 is edge sensitive

GPIO\_PORTA\_IBE\_R &= ~0x10; // ( ) PA4 not both edge sensitive

GPIO\_PORTA\_IEV\_R &= ~0x10; // ( ) PA4 falling edge event

GPIO\_PORTA\_ICR\_R = 0x10; // (e) Clear flag4

GPIO\_PORTA\_IM\_R |= 0x10; // (f) ARM Interrupt on PC4

NVIC\_PRI0\_R = (NVIC\_PRI0\_R &0xFF00FFFF) |0x00A00000; // (g) Priority 5

NVIC\_EN0\_R = 4;

}

void goNorth(void)

{

PinoutSet();

int i = 0;

for(i = 0; i < 3; i++)

{

if(i == 0){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is RED

delay(30000000); // ~3seconds delay

i++;

}

if(i == 1){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_2, 4); // D is GREEN

delay(50000000); // ~5seconds delay

i++;

}

if(i == 2){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_1, 2); // D is YELLOW

delay(10000000); // ~3seconds delay

}

}

GPIO\_PORTA\_ICR\_R |= 0x11; /\* clear the interrupt flag before return \*/

}

//---------------------------------------------------------------------------

// GPIO\_PortE Initialization for East-West Traffic

//---------------------------------------------------------------------------

void goEast\_Init(void)

{

SYSCTL\_RCGC2\_R |= 0x10; // (a) activate port E

FallingEdges = 0; // (b) initialize the counter

GPIO\_PORTE\_DIR\_R &= ~0x10; // (c) make PE4 input

GPIO\_PORTE\_DEN\_R |=0x10; // ( ) enable digital pin

GPIO\_PORTE\_IS\_R &= ~0x10; // (d) PE4 is edge sensitive

GPIO\_PORTE\_IBE\_R &= ~0x10; // ( ) PE4 not both edge sensitive

GPIO\_PORTE\_IEV\_R &= ~0x10; // ( ) PE4 falling edge event

GPIO\_PORTE\_ICR\_R = 0x10; // (e) Clear flag4

GPIO\_PORTE\_IM\_R |= 0x10; // (f) ARM Interrupt on PC4

NVIC\_PRI0\_R = (NVIC\_PRI0\_R &0xFF00FFFF) |0x00A00000; // (g) Priority 5

NVIC\_EN0\_R = 4;

}

void goEast(void)

{

PinoutSet();

int i = 0;

for(i = 0; i < 3; i++)

{

if(i == 0){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is RED

delay(30000000); // ~3seconds delay

i++;

}

if(i == 1){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_2, 4); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is GREEN

delay(50000000); // ~5seconds delay

i++;

}

if(i == 2){

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, 0);

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_1, 2); // B is RED

GPIOPinWrite(GPIO\_PORTD\_BASE, GPIO\_PIN\_0, 1); // D is YELLOW

delay(10000000); // ~3seconds delay

}

}

GPIO\_PORTE\_ICR\_R |= 0x11; /\* clear the interrupt flag before return \*/

}