**Embedded Systems and Design Report**

**Final Report**

Traffic Light Controller

A report submitted in part fulfilment Design Lab course

**2nd Year (3rd Semester) in ECE**

**Supervisor:** Dr. Deepak Nair



|  |
| --- |
| **Designer:** Devendra singh shekhawat(18uec155) |
|
|  |

**Department of Electronics and Communication Engineering**

**The LNMIIT, Jaipur, Rajasthan, India**

July 04, 2021

**Declaration**

This report has been prepared on the basis of our work and designs developed under Embedded system and design Course. Where other published and unpublished source materials have been used, these have been acknowledged.

Students Name:

|  |
| --- |
| Devendra Singh Shekhawat – 18uec155 |

Date of Submission:

Signature of Supervisor:

Name of Supervisor: Dr. Deepak Nair

Table of Contents

[Abstract 4](#_Toc24225311)

[Project Specification 5](#_Toc24225312)

[Chapter 1: Introduction 6](#_Toc24225313)

[1.1 Objectives 6](#_Toc24225314)

[1.2 Basic Operations 6](#_Toc24225315)

[Chapter 2: Hardware Description and Working 11](#_Toc24225321)

[Chapter 3: Design and Implementation 13](#_Toc24225322)

[3.1 Source Code 13](#_Toc24225323)

[3.2 Circuit Diagram 15](#_Toc24225324)

[Chapter 4: Simulation and Result 17](#_Toc24225326)

[Chapter 5: Cost of Project 20](#_Toc24225327)

[Chapter 6: Conclusion 21](#_Toc24225328)

[Bibliography 22](#_Toc24225329)

Abstract

This project is all about the “Traffic Light Controller” circuits. Traffic of cars and pedestrians are controlled by using simple components like LEDs, resistors, switches and Tiva Launchpad. This project can be considered as getting started on TIVA C series TM4C123GXL launchpad from Texas Instruments. This launchpad has TM4C123GH6PM microcontroller and its details are [here](http://www.ti.com/tool/ek-tm4c123gxl).

Project Specification

The project consists of three traffic signals: West, South and Walk. These signals are controlled by three switches (sensors in reality). The working of each LEDs are: -

* Green: Go
* Yellow: Warn (turns on when there is change from green to red only)
* Red: Stop

# Introduction

The normal function of traffic lights requires more than sight control and coordination to ensure that traffic and pedestrians move as smoothly, and safely as possible. A variety of different control systems are used to accomplish this, ranging from simple clockwork mechanisms to sophisticated computerized control and coordination systems that self-adjust to minimize delay to people using the junction. The block diagram of the system is shown in Figure 1. The main objectives of the study include-

* To introduce and learn about Tiva C Board.
* To apply the C programming language systematically.
* To experiment on the compatibility of the working relation among the microcontroller and other components used.

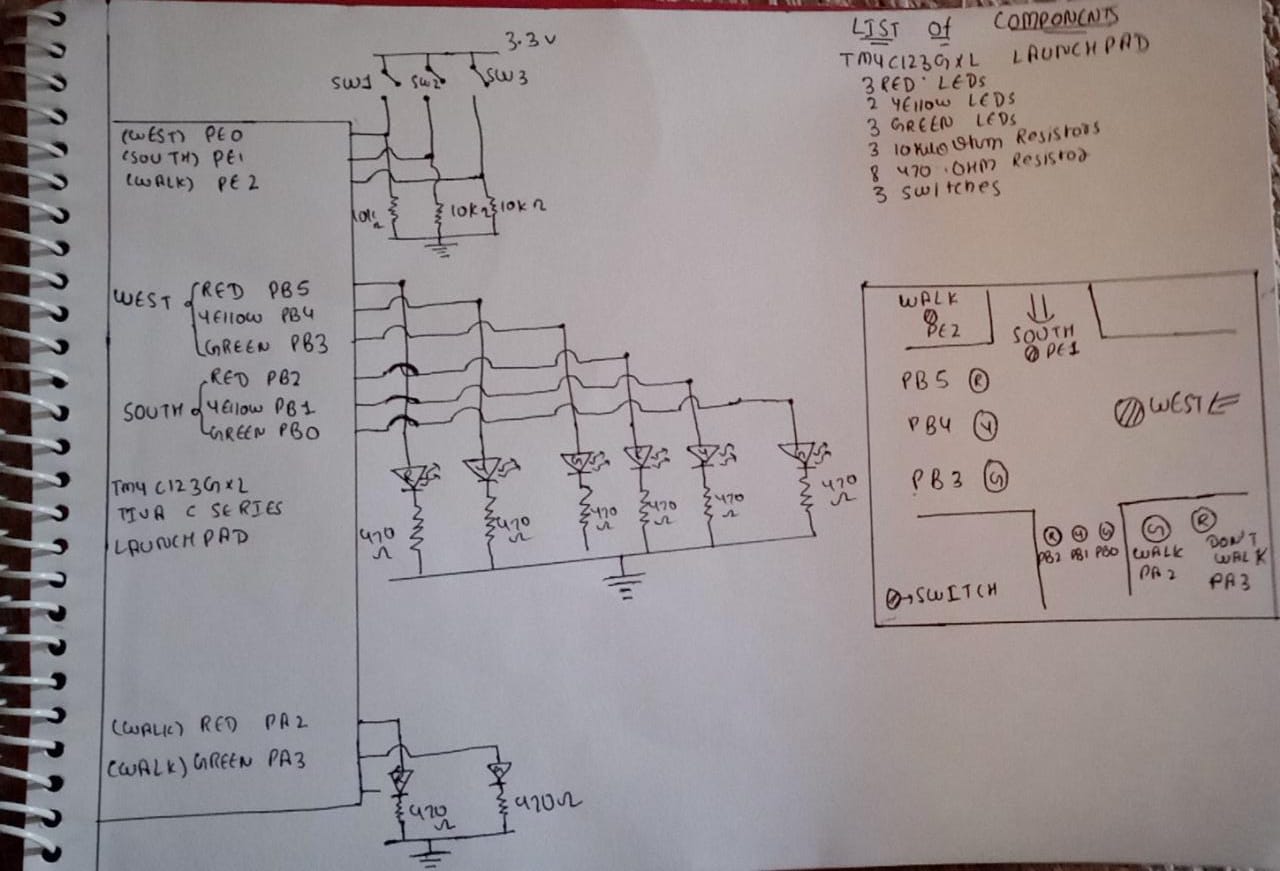


Figure 1: Block Diagram for Embedded systems and Design.

## Objectives

The objectives of the project are:

* The TLC must control all the signals.
* It must consider all the situation.
* It should control all the traffic properly.

## Basic Operations

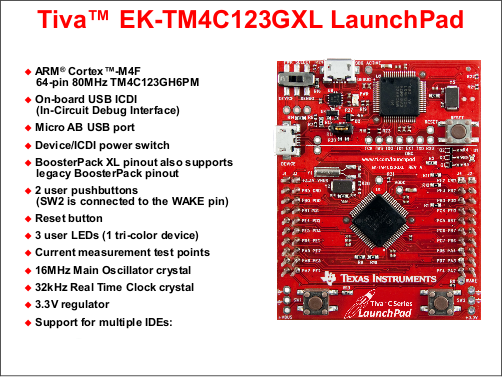
The basic operations of Traffic Light Controller are as follows:

* It would consider west and south signals.
* It also has Instruction to pedestrians too (Go and Stop) with the signals for vehicles.

# Chapter 2: Hardware Description

The required hardware and detail description of components used is discussed below: -

## TIVA C Series TM4C123G Launchpad from Texas Instruments



### Description

The TM4C123G LaunchPad Evaluation Kit is a low-cost evaluation platform for ARM Cortex-M4F based microcontrollers from Texas Instruments. The design of the TM4C123G LaunchPad highlights the [TM4C123GH6PM](http://shukra.dese.iisc.ac.in/emsys/tivac/ek-tm4c123gxl/TM4C123GH6PM_Microcontroller_Data_Sheet.pdf) microcontroller with a USB 2.0 device interface and hibernation module.

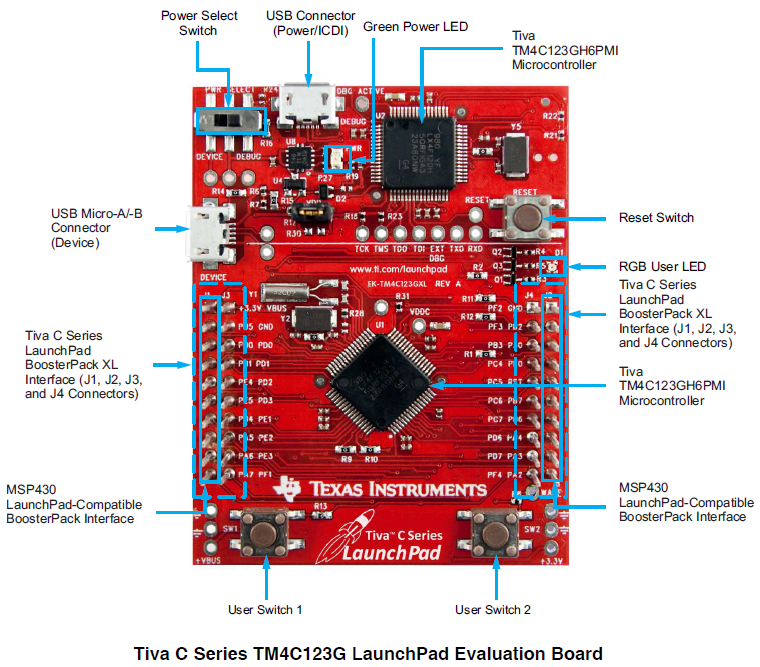
The EK-TM4C123GXL also features programmable user buttons and an RGB LED for custom applications. The stackable headers of the TM4C123G LaunchPad BoosterPack™ XL Interface make it easy and simple to expand the functionality of the TM4C123G LaunchPad when interfacing to other peripherals with Texas Instruments MCU BoosterPack.

### Features

The ARM Cortex-M4F Based MCU TM4C123G LaunchPad Evaluation Kit (EK-TM4C123GXL) offers these features:

* High Performance TM4C123GH6PM MCU:
  1. 80MHz 32-bit ARM Cortex-M4F CPU
  2. 256KB Flash, 32KB SRAM, 2KB EEPROM
  3. Two Controller Area Network (CAN) modules
  4. Dual 12-bit 2MSPS ADCs, motion control PWMs
  5. 8 UART, 6 I2C, 4 SPI
* On-board In-Circuit Debug Interface (ICDI)
* USB Micro-B plug to USB-A plug cable.
* Preloaded RGB quick-start application
* Tiva C Series TM4C123G LaunchPad -- [Readme First](http://shukra.dese.iisc.ac.in/emsys/tivac/ek-tm4c123gxl/Tiva_C_Series_TM4C123G_LaunchPad_Readme_First.pdf)

### Tiva C Series TM4C123G LaunchPad Evaluation Board



# Working Principle

The working is described with respect to the inputs (LSB=West Switch, Middle bit=South Switch and MSB=Walk Switch). This is as follows:

* **Input 000** (Go\_West): This is the Initial stage which is initialized in the code. The car will go to the west with green LED of west signal ON and Red LEDs ON for both south and walk traffic signals. (You can keep any one of the below states as initial state).
* **Input 001** (Go\_West): The West Switch/Sensor is active. Same output as that of initial stage.
* **Input 010** (Go\_South): The South Switch/Sensor is active. The Yellow LED of West traffic Signal will turn ON (Wait\_West) for few seconds to warn the cars that the Red LED will be turn ON and the west bound cars have to stop. As soon as the Red LED of West traffic signal turns ON, the Green LED of South traffic signal turns ON allowing south bound cars to go. The red LED of walk traffic signal is ON.
* **Input 011**: The South and West Switch/Sensor are active. The Green LED of south signal will turn OFF and the Yellow LED will turn ON (Wait\_South) for few seconds to warn the south bound cars. As soon as the Red LED of South signal turns ON, the Green LED of West signal turns ON. After few seconds, again yellow and then red LED of West signal turns ON and correspondingly green LED of South signal will be ON. This goes on repeating till the input is 011. The Walk signal is Red.
* **Input 100** (Walk\_Ped): The Walk Switch/Sensor is active. The Red LED of both West and South signals will turn ON. Green LED will be ON to allow Pedestrians to cross the intersection.
* **Input 101**: The Walk and West Switch/Sensor are active. In this case, Pedestrians can cross for some time and West bound cars can go for some time and the cycle is repeated. The transition from green to red is via yellow LED of West signal. In case of Walk signal, the flashing of Red LED (Red LED flashes two times) indicates that Pedestrians should hurry to be safe as the Red LED will be turn ON and Green LED of West signal will be ON. The South signal is Red.
* **Input 110**: The Walk and South Switch/Sensor are active. Same working as that of Input 101. Read South instead of West in above bullet point! The West signal is Red.
* **Input 111**: This input is an interesting case. Here all Switches/Sensors are active. Pedestrians will get some time to cross the road. West bound cars as well as South bound cars will get some time to cross the roads. Microcontroller is smart enough to avoid accident.

### Finite State Machine Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number given to states | Name of  the states | Output to Six Traffic Lights for Cars | | | | | |  | Output to two  Traffic Lights for Pedestrians | |  | Time in units of 10ms | Inputs  MSB: Walk Sensor (PE2) Middle bit: South Sensor (PE1) LSB: West Sensor (PE0) | | | | | | | |
| Next States | | | | | | | |
| West | | | South | | | Don’t Walk | Walk | I/P 000 | I/P 001 | I/P 010 | I/P 011 | I/P 100 | I/P 101 | I/P 110 | I/P 111 |
| PB5  (R) | PB4  (Y) | PB3  (G) | PB2  (R) | PB1  (Y) | PB0  (G) | Hex | PA3  (Y) | PA2  (R) | Hex |
| 0 | Go\_West | Low | Low | High | High | Low | Low | 0x0C | Low | High | 0x04 | 150 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | Wait\_West | Low | High | Low | High | Low | Low | 0x14 | Low | High | 0x04 | 50 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 2 |
| 2 | Go\_South | High | Low | Low | Low | Low | High | 0x21 | Low | High | 0x04 | 150 | 2 | 3 | 2 | 3 | 3 | 3 | 3 | 3 |
| 3 | Wait\_South | High | Low | Low | Low | High | Low | 0x22 | Low | High | 0x04 | 50 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 |
| 4 | Walk\_Ped | High | Low | Low | High | Low | Low | 0x24 | High | Low | 0x08 | 150 | 4 | 5 | 5 | 5 | 4 | 5 | 5 | 5 |
| 5 | Hurry\_On\_Ped1 | High | Low | Low | High | Low | Low | 0x24 | Low | High | 0x04 | 50 | 6 | 6 | 6 | 6 | 4 | 6 | 6 | 6 |
| 6 | Hurry\_Off\_Ped1 | High | Low | Low | High | Low | Low | 0x24 | Low | Low | 0x00 | 50 | 7 | 7 | 7 | 7 | 4 | 7 | 7 | 7 |
| 7 | Hurry\_On\_Ped2 | High | Low | Low | High | Low | Low | 0x24 | Low | High | 0x04 | 50 | 8 | 8 | 8 | 8 | 4 | 8 | 8 | 8 |
| 8 | Hurry\_Off\_Ped2 | High | Low | Low | High | Low | Low | 0x24 | Low | Low | 0x00 | 50 | 0 | 0 | 2 | 0 | 4 | 0 | 2 | 0 |

# Design and Implementation

## Source Code

*// \*\*\*\*\* 0. Documentation Section \*\*\*\*\**

*// Runs on TM4C123*

*// Index implementation of a Moore finite state machine to operate a traffic light.*

*//20 April 2021*

*//Reference: http://users.ece.utexas.edu/~valvano/*

*// "don't walk" red light connected to PA2*

*// "walk" yellow light connected to PA3*

*// 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)*

*// \*\*\*\*\* 1. Pre-processor Directives Section \*\*\*\*\**

*#include "TM4C123GH6PM.h"*

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

*// FUNCTION PROTOTYPES: Each subroutine defined*

*void SysTick\_Init(void); // initialize SysTick Timer*

*void SysTick\_Wait(unsigned long delay1);*

*void SysTick\_Wait10ms(unsigned long delay1); // waiting time*

*struct State{*

*unsigned long TrafficLights\_Cars; //6 bits output to Six LEDs for Cars*

*unsigned long TrafficLights\_Ped; //2 bits output to two Leds for Pedestrians*

*unsigned long Time; //delay in 10ms units*

*unsigned long Next[8]; //next states for inputs 0,1,2,3,4,5,6,7*

*};*

*typedef const struct State STyp;*

*#define Go\_West 0 //name of the states*

*#define Wait\_West 1*

*#define Go\_South 2*

*#define Wait\_South 3*

*#define Walk\_Ped 4*

*#define Hurry\_On\_Ped1 5*

*#define Hurry\_Off\_Ped1 6*

*#define Hurry\_On\_Ped2 7*

*#define Hurry\_Off\_Ped2 8*

*//Next[8] array represents 8 inputs and the next state transition*

*//The first array element represents 000 input, the second represents 001 input and so on upto 111 (eighth input)*

*//Input bits representation- MSB=Walk Sensor,Middle bit=South Sensor and LSB=West Sensor*

*//FSM[9]={*

*//{ouput to Six signal LEDs (TrafficLights\_Cars),output to two Pedestrian LEDs (TrafficLights\_Ped),Time in 10ms units,Next[8] array elements(next states)},*

*// ..............*

*STyp FSM[9]={*

*{0x0C,0x04,100,{0,0,1,1,1,1,1,1}}, //state 1*

*{0x14,0x04,50 ,{2,2,2,2,4,4,4,2}}, //state 2*

*{0x21,0x04,100,{2,3,2,3,3,3,3,3}}, //state 3*

*{0x22,0x04,50 ,{0,0,0,0,4,4,4,4}}, //state 4*

*{0x24,0x08,100,{4,5,5,5,4,5,5,5}}, //state 5*

*{0x24,0x04,25 ,{6,6,6,6,4,6,6,6}}, //state 6*

*{0x24,0x00,10 ,{7,7,7,7,4,7,7,7}}, //state 7*

*{0x24,0x04,25 ,{8,8,8,8,4,8,8,8}}, //state 8*

*{0x24,0x00,10 ,{0,0,2,0,4,0,2,0}} //state 9*

*};*

*unsigned long S; //index to the current state*

*unsigned long Input;*

*// \*\*\*\*\* 3. Subroutines Section \*\*\*\*\**

*int main(void) {*

*volatile unsigned long delay;*

*SysTick\_Init();*

*SYSCTL->RCGC2=0x13; //activate clock for port A,E,B*

*delay=SYSCTL->RCGC2; // no need to unlock port B,E,A*

*GPIOB->AMSEL=0x00; //disable analog on port B*

*GPIOE->AMSEL=0x00; //disable analog on port E*

*GPIOA->AMSEL=0x00; //disable analog on port A*

*GPIOB->PCTL=0x00000000; //enable regular GPIO*

*GPIOE->PCTL=0x00000000; //enable regular GPIO*

*GPIOA->PCTL=0x00000000; //enable regular GPIO*

*GPIOB->DIR=0x3F; //outputs on PB0-5*

*GPIOE->DIR=0x00; //inputs on PE0-2*

*GPIOA->DIR=0x0C; //outputs on PA2 and PA3*

*GPIOB->AFSEL=0x00; //disable alternate function*

*GPIOE->AFSEL=0x00; //disable alternate function*

*GPIOA->AFSEL=0x00; //disable alternate function*

*GPIOB->DEN=0x3F; //enable digital I/O on PB0-5*

*GPIOE->DEN=0x07; //enable digital I/O on PE0-2*

*GPIOA->DEN=0x0C; //enable digital I/O on PA2 and PA3*

*S=Go\_West;*

*while(1){*

*GPIOB->DATA=FSM[S].TrafficLights\_Cars; //set car signal lights*

*GPIOA->DATA=FSM[S].TrafficLights\_Ped; //set walk signal lights*

*SysTick\_Wait10ms(FSM[S].Time);*

*Input=GPIOE->DATA; //read sensors*

*S=FSM[S].Next[Input];*

*}*

*}*

*void SysTick\_Init(void){*

*SysTick->LOAD = 0; // disable SysTick during setup*

*SysTick->VAL = 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 delay1){*

*SysTick->LOAD = delay1-1; // number of counts to wait*

*SysTick->VAL = 0; // any value written to CURRENT clears*

*while((SysTick->LOAD&0x00010000)==0){ // wait for count flag*

*}*

*}*

*// 800000\*12.5ns equals 10ms*

*void SysTick\_Wait10ms(unsigned long delay1){*

*unsigned long i;*

*for(i=0; i<delay1; i++){*

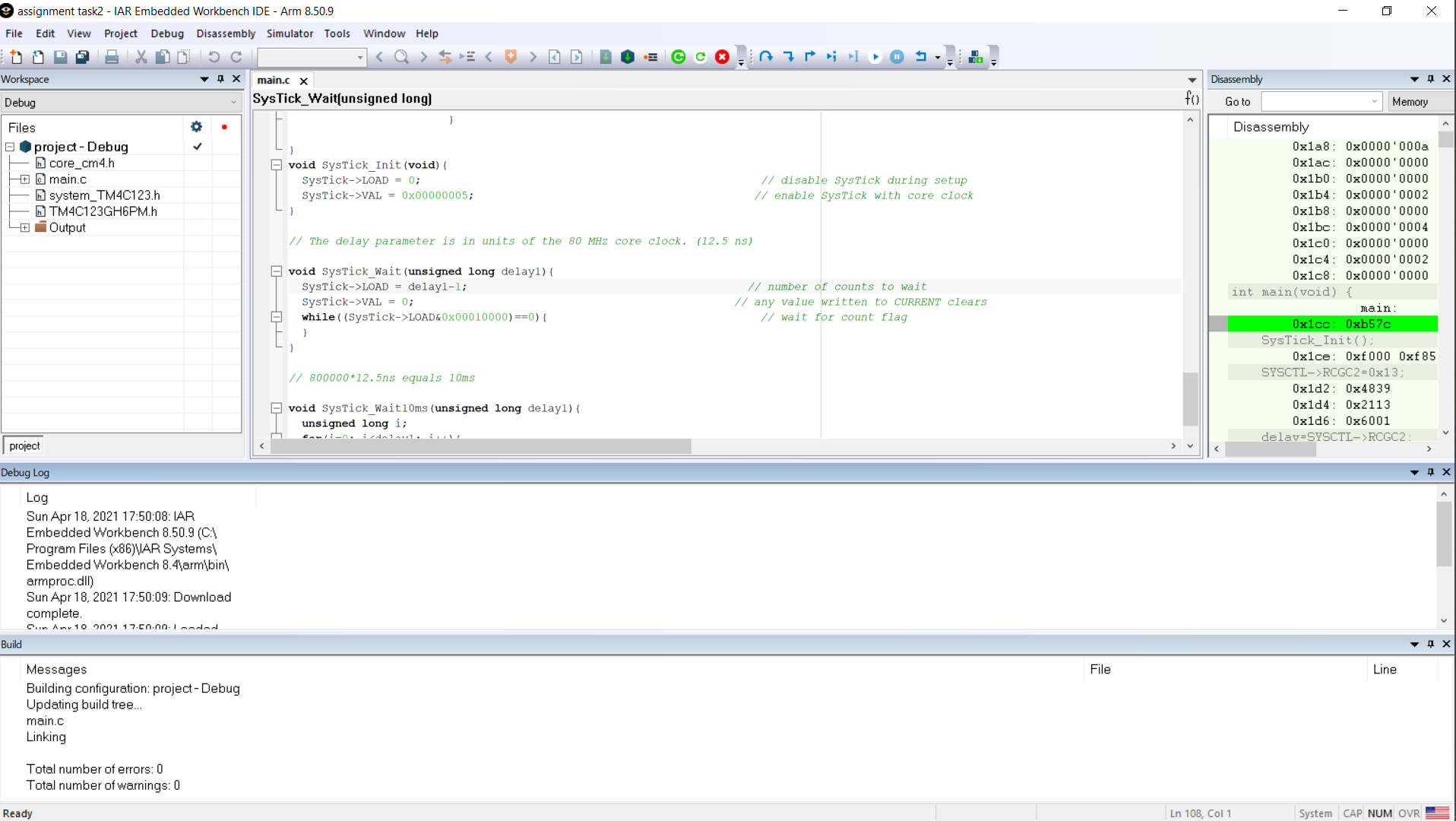
*SysTick\_Wait(800000); // wait 10ms*

*}*

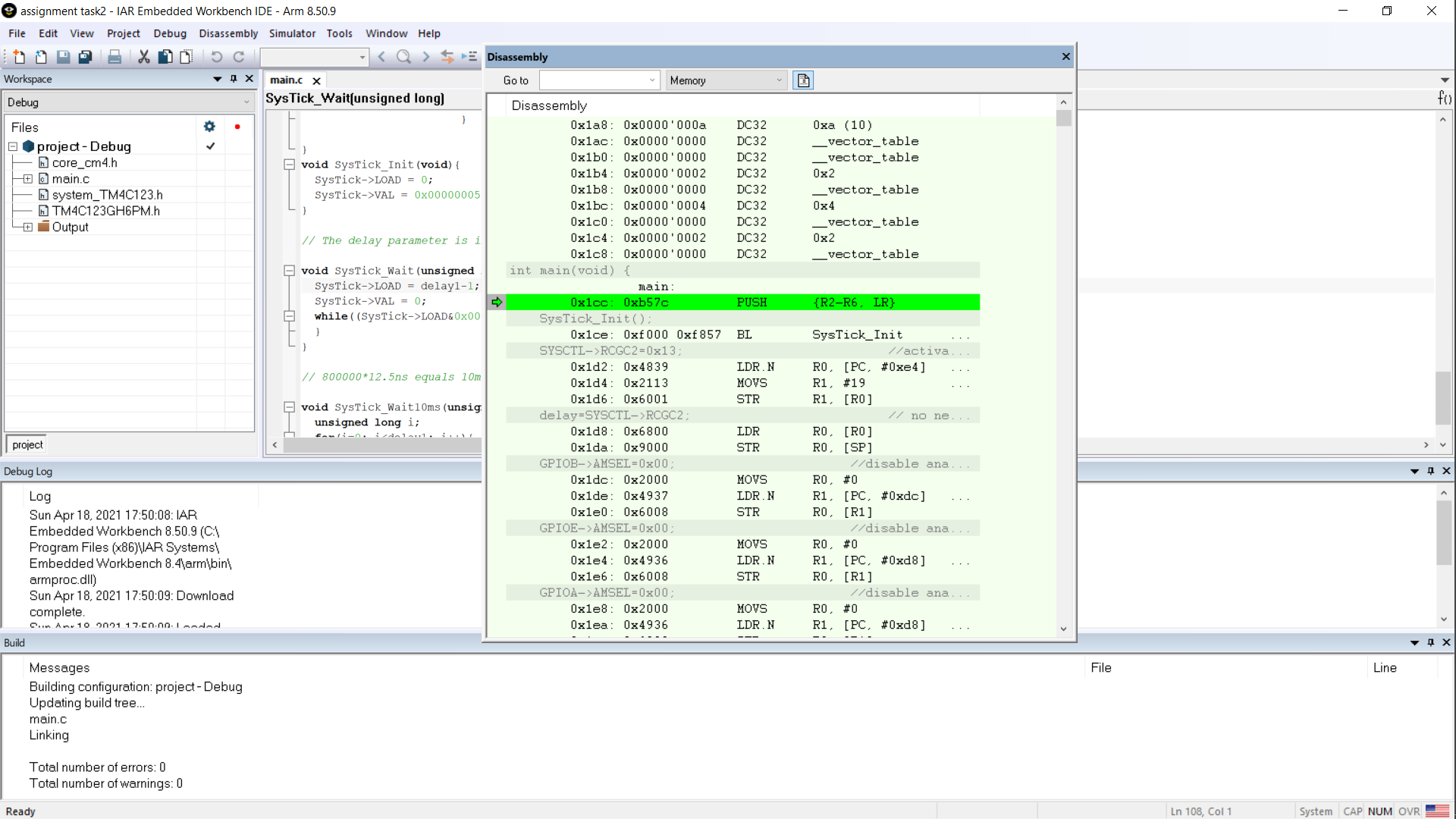
*}*

# Simulation and Result

* 1. Overall result: -



* 1. Disassembly: -



# Cost of Project

|  |  |  |
| --- | --- | --- |
| **Name of Component** | **Number of Component** | **Cost of Component** |
| TIVA C TM4C123GH | 1 | Rs.1134 |
| LED (generic) | 8 | Rs.10 |
| SparkFun Pushbutton switch 12mm | 3 | Rs.135 |
| Resistor 10k ohm | 3 | Rs.2 |
| Resistor 475 ohm | 8 | Rs.8 |
| Male/Male Jumper Wires | 15 | Rs.64 |
| Breadboard (generic) | 1 | Rs. 65 |
| **TOTAL** Rs.1418 | |

# Conclusion

In this project, a smart traffic control system (STCS) for the infrastructure in the smart city is proposed, which can be widely applied for the intelligent transportation system in smart city applications after some changes in it. The major components in the proposed STCS include TIVA-C Board, then the data can be sent online through WIFI or Bluetooth to make it IOT project too. It supports various smart city ITS applications including eco-driving, pre-time signal control,

The TM4C123 is the core of this work, where we discuss little bit about the system architecture, middleware, peripheral hardware modules, and control algorithm.

We can take this on higher level too just by adding IOT module, more sensors, etc.

In the future, real-time traffic information fusion for Adaptive Traffic Signal Control (ATSC) by integrating multiple data sources such as loop detector and smart AVI is to be designed.

Bibliography

1. Nellore, K.; Hancke, G. A survey on urban traffic management system using wireless sensor networks. Sensors **2016**, 16, 157.
2. Lee,W.H.; Chou, C.M.; Kuo, Y.H.; Hung,W.L.; Chang, D.W. Development of a Roadside Unit Middleware for Vehicular Ad-Hoc Network Based Intelligent Transportation Systems. J. Internet Technol. **2017**, 18, 1431–1439.
3. Ahn, S.; Choi, J. Internet of vehicles and cost-e\_ective tra\_c signal control. Sensors **2019**, 19, 1275.
4. Lee,W.H.; Li, J.Y. An Eco-Driving Advisory System for Continuous Signalized Intersections by Vehicular Ad Hoc Network. J. Adv. Transp. **2018**, 2018, 5060481.
5. Google and TI website for more ideas and details.