

# CSE 312

# Microprocessor Based Systems

# Project 2 Report

|  |  |
| --- | --- |
| Name | Section |
| Omar Nasr El-Dean Mohamed | 2 |
| Omar Hesham Hanafy | 2 |
| Fatma Abd-El Moniem Mohamed | 2 |
| Kareem Salah Metwaly | 2 |
| Mohamed Erfan El Sayed | 3 |
| Nada Ihab Ahmed Mohamed | 3 |

2018/2019

Cairo, Egypt

## Table of contents

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| 1. Table of contents……………………………………………………………………………………… | 2 |
| 1. Project’s structure……………………………………………………………………………………. | 3 |
| 1. Used Peripherals……………………………………………………………………………………. | 4 |
| 1. MCAL Layers…………………………………………………………………………………………… | 4 |
| 1. Functions’ Description……………………………………………………………………………. | 5 |
| 1. Project’s Photo………………………………………………………………………………………. | 11 |
| 1. Video’s Link……………………………………………………………………………………………. | 11 |

## Project’s Structure

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In this project, three subsystems –which can be used in a smart home- are implemented using two launch pads -Tm4c123gh6pm- commonly known as Tiva c, the two Tivas communicate together using UART communication protocol.

The 1st Tiva is connected to:

* A stepper motor which can rotate 30 degrees clock wise or anti clock wise according to the order sent to it from the 2nd Tiva when one of the switches are pressed, for example if the left switch is pressed, the motor will rotate 30 degrees anti clock wise and so on.
* A Temperature sensor that is built in the Tiva, which is used to measure the Tiva’s Processor temperature.
* An LED that changes its degree of illumination according to the PWM signal sent to it from the 2nd Tiva using a potentiometer.

The 2nd Tiva is connected to:

* Two switches that are used to rotate the stepper motor as mentioned before, using UART.
* An LCD that is used to display the temperature measured form the 1st Tiva’s temperature sensor.
* A potentiometer that is used to change the 1st Tiva’s LED light intensity.

## Used Peripherals

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The used protocols are:

* UART communication protocol that is used to establish communication between the two launch pads.
* Pulse width modulation (PWM) that is used to change the light intensity of the LED according to a certain voltage.
* Analog to digital convertor (ADC) used in Temperature reading and potentiometer reading.

## MCAL Layers

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 

MCAL layers are divided into:

* Microcontroller Driver
* PLL Driver
* Communication Drivers
* UART Driver
* Internal Drivers
  + DIO Driver
  + PORT Driver
  + ADC Driver
  + PWM Driver
* Device Drivers
* LCD Driver
* Stepper Motor Driver
* Temperature Sensor
* Button driver

## Functions’ Description

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Microcontroller Driver:

1. Port Driver

|  |  |
| --- | --- |
| Function name | Port\_Init(uint8) |
| Input | Uint8 Index of the port to be initialized |
| Return | Void |
| Description | Enable specific Port and determine its alternative function. |

|  |  |
| --- | --- |
| Function name | Port\_SetPinDirection(uint8,uint8,Port\_PinDirectionType) |
| Input | Uint8 Port index, Uint8 pins’ mask, enum Port\_PinDirectionType pins’ level |
| Return | Void |
| Description | Set the direction of pins if it’s input (or) output. |

|  |  |
| --- | --- |
| Function name | Port\_SetPinPullUp(uint8,uint8,uint8 ) |
| Input | Uint8 Port index, Uint8 pins’ mask, Uint8 enable pull up |
| Return | Void |
| Description | If it enable you control the state of internal resistance –it is using for switch connection-to be pull-up. |

|  |  |
| --- | --- |
| Function name | Port\_SetPinPullDown(uint8,uint8,uint8 ) |
| Input | Uint8 Port index, Uint8 pins’ mask, Uint8 enable down up |
| Return | Void |
| Description | If it enable you control the state of internal resistance –it is using for switch connection-to be pulldown. |

1. PLL Driver

|  |  |
| --- | --- |
| Function name | PLL\_Init() |
| Input | Void |
| Return | Void |
| Description | Initialize system’s clock according to predefined system Divisor |

### Communication Driver:

1. UART Driver

|  |  |
| --- | --- |
| Function name | UARTX\_Init() |
| Input | Void |
| Return | Void |
| Description | Enable UART Module by enabling the UART clock, set the baud rate and other UART properties. |

|  |  |
| --- | --- |
| Function name | Uint8\_t UARTX\_Available(void) |
| Input | Void |
| Return | Uint8 Availability status |
| Description | Checks the FIFO of the UART to check that data arrived or not. |

|  |  |
| --- | --- |
| Function name | Uint8\_t UARTX\_Read(void) |
| Input | Void |
| Return | Uint8 Data |
| Description | Returns the value in the data register of UART. |

|  |  |
| --- | --- |
| Function name | void UARTX\_Write(Uint8\_t data) |
| Input | Data |
| Return | Void |
| Description | Write data on the UART data register. |

|  |  |
| --- | --- |
| Function name | void UARTX\_Read\_String(char c[]) |
| Input | String or char[] |
| Return | Void |
| Description | Concatenates Characters received by UART until it received null character to form a string. |

### I/O Driver:

1. DIO Driver

|  |  |
| --- | --- |
| Function name | DIO\_ReadPort(uint8,uint8) |
| Input | Uint8 Port index, Uint8 pin |
| Return | Level |
| Description | This function return the value of specific pin in any Port. |

|  |  |
| --- | --- |
| Function name | DIO\_WritePort(uint8,uint8,uint8) |
| Input | Uint8 Port index, Uint8 pin, level |
| Return | void |
| Description | This function is used to set a specific value –LOW or HIGH- for specific pin in any Port. |

1. ADC Driver

|  |  |
| --- | --- |
| Function name | Void ADC0\_Init(void) |
| Input | Void |
| Return | Void |
| Description | Initialization function that enables ADC clock, enable software trigger conversion, get input from channel 0, takes 1 sample at a time, enable ADC sequencer, and other ADC properties. |

|  |  |
| --- | --- |
| Function name | Void ADC0\_SS3\_In(uint16\_t \*data) |
| Input | Void |
| Return | Void |
| Description | This function starts conversion, wait for conversion until it’s completed, read conversion results, then clears completion flag. |

1. PWM Driver

|  |  |
| --- | --- |
| Function name | Blue\_PWM1\_Init() |
| Input | Void |
| Return | Void |
| Description | Enable PWM, determine its clock, control properties of PORTF and set other PWM properties. |

|  |  |
| --- | --- |
| Function name | BlueLED\_SetDutyCycle( Uint8\_t Duty cycle) |
| Input | Uint8 Duty cycle |
| Return | void |
| Description | Set value to its compare register according the value of duty cycle. |

### Device Drivers:

1. LCD Driver:

|  |  |
| --- | --- |
| Function name | LCD\_Init() |
| Input | Void |
| Return | void |
| Description | Enable (PORTE and PORTB) and set direction of their pins. |

|  |  |
| --- | --- |
| Function name | LCD\_CMD(char) |
| Input | The character that is typed in the LCD |
| Return | void |
| Description | Set values of (RS, RW, E) of LCD and set command in PORTB which control the LCD. |

1. Stepper Motor Driver

|  |  |
| --- | --- |
| Function name | void STEPPER\_MOTOR\_voidInit(void); |
| Input | Void |
| Return | void |
| Description | Enable PORTB and set direction of its pins  -PB0,PB1,PB2,PB3- |

|  |  |
| --- | --- |
| Function name | void STEPPER\_MOTOR\_MOVE\_30(int); |
| Input | Takes an integer value that determines whether to rotate clock or anti clock wise |
| Return | Void |
| Description | Control the stepper motor to rotate 30 degree by delay function and control PORTB DATA register. |

1. Temperature sensor Driver:

|  |  |
| --- | --- |
| Function name | void Temp\_Init(void) |
| Input | Void |
| Return | Void |
| Description | Initialize ADC and ADC sequencer through Pin E3 and choose Temperature sensor as ADC source. |

|  |  |
| --- | --- |
| Function name | uint16\_t Temp\_Read(); |
| Input | Void |
| Return | Uint16 Temperature Reading |
| Description | Returns temperature through the ADC of E3. |

1. Button Driver

|  |  |
| --- | --- |
| Function name | Button\_Init(uint8,uint8,Button\_Type ) |
| Input | uint8 port index, uint8 pins’ mask, enum Button\_Type |
| Return | Void |
| Description | Enable the PORTF and control the internal resistance type. |

|  |  |
| --- | --- |
| Function name | bool Button\_IsPressed(uint8,uint8,Button\_Type ) |
| Input | uint8 port index, uint8 pins’ mask, enum Button\_Type |
| Return | bool |
| Description | Check the state of the button. |

## Project’s Photo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 

## Video’s Link

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<https://www.youtube.com/watch?v=7RAbdAztdfc>