

**Program: MCTA**

***Course Code:***

***Course Name:***

***Examination Committee***

**Dr. Sherif Hammad**

**Ain Shams University**

**Faculty of Engineering**

**International Credit Hours Engineering Programs (I-CHEP)**

**Spring Semester – 2020**

RESEARCH & PROJECT SUBMISSIONS

**Student Personal Information for Group Work**

|  |
| --- |
| **Student Names:** |
| Fill in |
| Fill in |
| Fill in |
| Fill in |
| Fill in |

|  |
| --- |
| **Student Codes:** |
| Fill in |
| Fill in |
| Fill in |
| Fill in |
| Fill in |

**Plagiarism Statement**

I certify that this assignment / report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they are books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment / report has not been previously been submitted for assessment for another course. I certify that I have not copied in part or whole or otherwise plagiarized the work of other students and / or persons.

**Signature/Student Name:**

**Date:**

Table of Contents

[1. Abstract 5](#_Toc42206339)

[2. Introduction 5](#_Toc42206340)

[3. Objective 6](#_Toc42206341)

[4. Description 6](#_Toc42206342)

[5. List of used peripherals 7](#_Toc42206343)

[5.1. GPIO port F 7](#_Toc42206344)

[5.2. GPIO port A (UART) 7](#_Toc42206345)

[5.3. GPTM (timer0A) 8](#_Toc42206346)

[5.4. SYSTICK 8](#_Toc42206347)

[6. Initialization 8](#_Toc42206348)

[6.1. GPIO PORT F 8](#_Toc42206349)

[6.2. Timer0A 8](#_Toc42206350)

[6.3. GPIO PORT A 8](#_Toc42206351)

[6.4. UART0 8](#_Toc42206352)

[6.5. SYSTICK 8](#_Toc42206353)

[7. List of interrupts 9](#_Toc42206354)

[7.1. UART interrupt 9](#_Toc42206355)

[RX\_handler 9](#_Toc42206356)

[7.2. GPIO Interrupt 9](#_Toc42206357)

[Button\_Fall 9](#_Toc42206373)

[Button\_Rise 9](#_Toc42206374)

[7.3. Timer0A interrupt 10](#_Toc42206375)

[Timer0\_handler 10](#_Toc42206377)

[7.4. Systick interrupt 10](#_Toc42206378)

[Systick\_handler 10](#_Toc42206379)

[8. Functions 11](#_Toc42206380)

[8.1. UARTprintf function 11](#_Toc42206381)

[8.2. Inc timer function 11](#_Toc42206382)

[8.3. Mode function 11](#_Toc42206383)

[9. Flow chart 12](#_Toc42206384)

[9.1. Main 13](#_Toc42206385)

[9.2. Initialization 14](#_Toc42206386)

[9.3. Interrupt handler functions 15](#_Toc42206387)

[UART 15](#_Toc42206388)

[GPIOF (switches SW0 and SW4) 16](#_Toc42206389)

[Systic and GPTM (Timer 0A) interrupts 17](#_Toc42206390)

# Abstract

# Introduction

Embedded system here

## Inclusions

An important topic to mention is the #includes that needed to be in the project so that the project runs. We were using the tivaware, so the following includes were used:

#include <stdint.h>

#include <stdbool.h>

#include "hw\_memmap.h"

#include "driverlib/debug.h"

#include "driverlib/gpio.h"

#include "driverlib/sysctl.h"

#include "tm4c123gh6pm.h"

#include "driverlib/uart.h"

#include "driverlib/pin\_map.h"

#include "driverlib/systick.h"

#include "driverlib/interrupt.h"

#include "driverlib/timer.h"

This includes are provided by Texas Instruments company as files in the Tiva C driver library. These files include many API functions and quick initializations and shortcuts to many functions instead of the register level programming. gpio.h, timer.h systick.h and uart.h were used for API functions related to the GPIO, timer0A, systick timer and UART0 respectively. interrupt.h, hw\_memmap.h, pin\_map.h were used for any interrupt and some pin locations definitions respectively. Tm4c123gh6pm.h was used for some built-in definitions.

# Objective

The project objective is to develop a microwave embedded application. To represent some inputs and displays, the pc serial terminal (putty) will be used. The microwave should have 2 operating modes (microwave and grill process).This modes should be displayed on the screen. A blue LED and a Green LED will be ON if the microwave or grill modes are operated, respectively. A counter of minutes and seconds that starts initially from zero will be displayed. When the user presses specific keys the counter will increments either by 10 seconds or 1 minute or 30 minutes. If switches 0 and 1 are pressed, respectively, the counter will start decrementing or stops counting. When the counter reaches zero, the Red LED will blink 5 seconds, and “DONE” message will be displayed in the screen. IF the switch 1 is pressed for 3 seconds, the counter will be zero again and the screen to the start mode menu. The user is then able to choose the operating mode again. Power saving design, with little code in the main file, should be approached. Interrupts must be used for buttons.

# Description

At the start of the application a start mode menu will appear. This mode menu will let the user choose the mode of operation either microwave or grill process. The GPIO interrupt will be disabled so that any switches pressed does not affect anything in this menu. Any button except ‘m’ or ‘g’, for microwave and grill respectively, will not affect the process. This is achieved by a switch case conditions in the UART interrupt handler function.

|  |  |
| --- | --- |
| If ‘m’ is pressed | If ‘g’ is pressed |
| * Switch flag so that (1,2,3) buttons can function and the (m, g) buttons does not function * Blue LED is ON * Print “microwave process” | * Switch flag so that (1,2,3) buttons can function and the (m, g) buttons does not function * Green LED is ON * Print “grill process” |

For both cases the timer will be shown in the screen with a zero value. If (1,2,3) button is pressed an increment of 10 seconds,1minute and 30 minutes, respectively, to the counter will happen. For example, if the user enters 1 then 2 then 3, the timer will be 31 minutes and 10 seconds as shown in the figure.

Once any of the (1, 2, 3) buttons is pressed and timer no longer is 00:00, the GPIO (switch) interrupt is enabled. This way, when switch 0 is pressed, the counter starts, and will start to decrement. In addition, the Systick timer starts and with the help of the Systick handler with each second timeout the counter decrements a second. The UART interrupt is disabled so that the user cannot increment the counter while a process is going. If switch 1 is pressed, the counter stops (by stopping the Systick timer) and the UART will be enabled to let the user increment the counter, if he wishes. If switch 1 is pressed for 3 seconds, timer0A interrupt handler will do the following:

* Switches off all LEDs
* Switch the menu flag
* Return to the start mode menu
* Reset the counter to 00:00
* Enable UART interrupt
* Disable GPIO interrupt

If the counter decrements to zero , which means the process ended .A Red LED will blink for 5 seconds and “DONE” will be displayed in the PC serial. Finally, the system will return to the start mode menu for the whole process to start again.

# List of used peripherals

## GPIO port F

The 5 pins where used, which are the 3 RGB LEDs and the two switches (switch 0 and 1).

## GPIO port A (UART)

Pin A0 was used as a receiver, and Pin A1 was used as transmitter.

## GPTM (timer0A)

Timer0A 32 bit, Periodic down, no presale, 3sec reload. Used in measuring the 3 seconds for resetting the timer.

## SYSTICK

Loaded with 1 sec, it is used so that with each timeout, the microwave counter decrement 1 sec, by using Systick interrupt handler function.

# Initialization

## GPIO PORT F

First, we enabled the clock for PORT F. We set the LEDs as outputs and the switches as inputs. We enabled the digital enable for the 5 pins. We enabled the Pull up Resistor for the switches.

## Timer0A

We enabled the clock for timer 0. We disabled the timer 0A to make the configurations (sandwich).We enabled the stall for the timer, and set the timer as periodic down counting. The reload time was set as 3 seconds

This was used to check if the user has pressed the switch for 3 seconds or not.

## GPIO PORT A

Enabled clock for port A

## UART0

We enabled the clock for UART. We set pinA0 as RX (receiver) and pinA1 as TX (transmitter). We enabled the alternative functions for both pins. We set the configuration for serial communication as (9600 baud rate, word length 8 bits, 1 stop bit and no parity). We disabled the FIFO so that the interrupt works for every character pressed by the user.

## SYSTICK

We set the reload period as 1 second and its priority to the least priority (7). This was used to decrement the counter in the display each second.

# List of interrupts

## UART interrupt

This was used so that when the user presses (m, g, 1, 2 and 3) buttons, specific tasks happens.

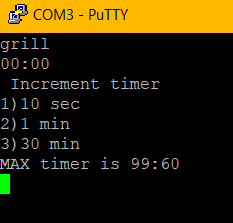
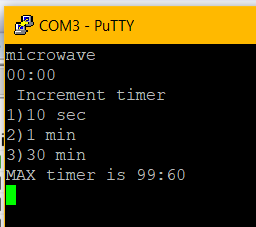
We enabled the receiver interrupt and set the RX\_handler function as the handler function.

Figure "m" is pressed

Figure "g" is pressed

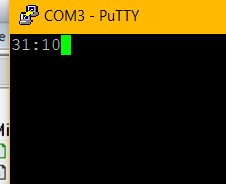


Figure all increment buttons are pressed once

### RX\_handler

This checks the availability of a character. If it receives any character, a (switch-case) with different conditions will do variable requirements. The interrupts clears at the end of the function.

## GPIO Interrupt

We used two handler functions for the GPIO interrupt of the switches (0, 1):

1. Button\_Fall
2. Button\_Rise

We set the handler function to be Button\_Fall, if the switch is pressed (setting the interrupt type as Falling\_Edge\_Trigger).If the user presses the switch, the handler function (Button\_Fall) is called.



### Button\_Fall

First, the interrupt is cleared, and the interrupt type is changed to rising\_edge\_trigger.The handler function is set to be Button\_Rise so that the next interrupt is called when the user release the switch.Timer0A is enabled and the counting starts. If the user releases the switch before 3 seconds, the Button\_Rise is handler function is entered.

### Button\_Rise

Second, the controller checks that the reason for the interrupt is pin\_4 (switch 1).if this is the case, “stopping operation” is printed in the PC screen, and the flag is switched so that the user can use increment buttons.

If pin\_0 (switch 0) is the reason for the interrupt, the Timer0A will be disabled, ” starting operation” is printed in the PC screen. Systick timer is enabled so that the microwave timer starts to decrement .Moreover, the UART interrupt will be disabled so that if the user accidentally enters any keyboard entry ,the system is not affected.

For both cases, the GPIO interrupt will be cleared Interrupt type will be changed into Falling\_Edge\_Trigger again (switch is pressed), and the handler function will be set to Button\_Fall.

## Timer0A interrupt

We enabled the timeout interrupt for timer0A.The handler function is set as Timer0\_handler.if the reload time (3 seconds) is reached, the handler function is called.

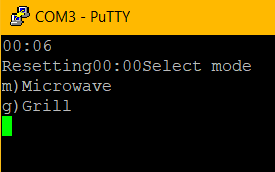


Figure output when user holds for 3 seconds



### Timer0\_handler

First, the interrupt is cleared. Then GPIO interrupt is disabled because if the user presses the switch for 3 seconds and releases, we don’t want the Button\_Rise interrupt handler function to be called. However, we have to set the GPIO interrupt type as Falling\_Edge\_Trigger and the handler functions as Button\_Fall so that when the user presses the switch the interrupt is called correctly.

If the switch is pressed for 3 seconds, the Systick is disabled to stop counting down. The LEDs are switched OFF, Resetting” is printed in the PC screen and the microwave timer is set back to zero. Moreover, the system gets back again to the start screen of the microwave. The UART interrupt is enabled and the GPIO interrupt is disabled so that the user enters (M, G) buttons to choose the operation again, and the switch presses don’t affect the system. Also, the Flag is set to neglect the entries (1-2-3) that increments the timer.

## Systick interrupt

We enabled the timeout interrupt for Systick timer. The handler function is set as Systick\_handler. If the reload time (1 second) is reached, the handler function is called.

### Systick\_handler

Figure timer 40 seconds

Figure timer decrementing

First the interrupt is cleared automatically once the compiler enters the handler function. (According to Tivaware datasheet). Then 1 sec is decremented from the microwave timer count. Then clear display screen and print the new timing to the display using UART.

Another function of the systick handler is to check that the overall timer is timedout ,and when this happens ,the following output is printed and the red LED blinks :



Figure output after timeout

# Functions

We used the (UARTprintf / inctimer / mode) functions in our application.

## UARTprintf function

This function was taken from uartstdio utility. It is used to print string and also can print values as strings example.

Uartprintf (“%d”, timer)

Timer here is a uint\_32t to show microwave timer seconds.

## Inc timer function

It shows the following output

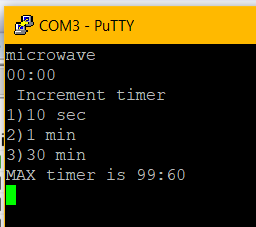


Figure output of Inctimer function

It switches the flag so that (1, 2, 3) buttons will function but the (m, g) buttons will not function.

## Mode function

It switches the flag so that (m, g) buttons functions and (1, 2, 3) buttons does not function. Also, it opens the start mode menu. Finally, it disables the GPIO interrupt so that if the user pressed any switch the application is not affected in the start menu.

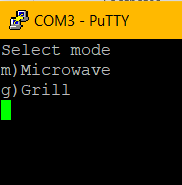
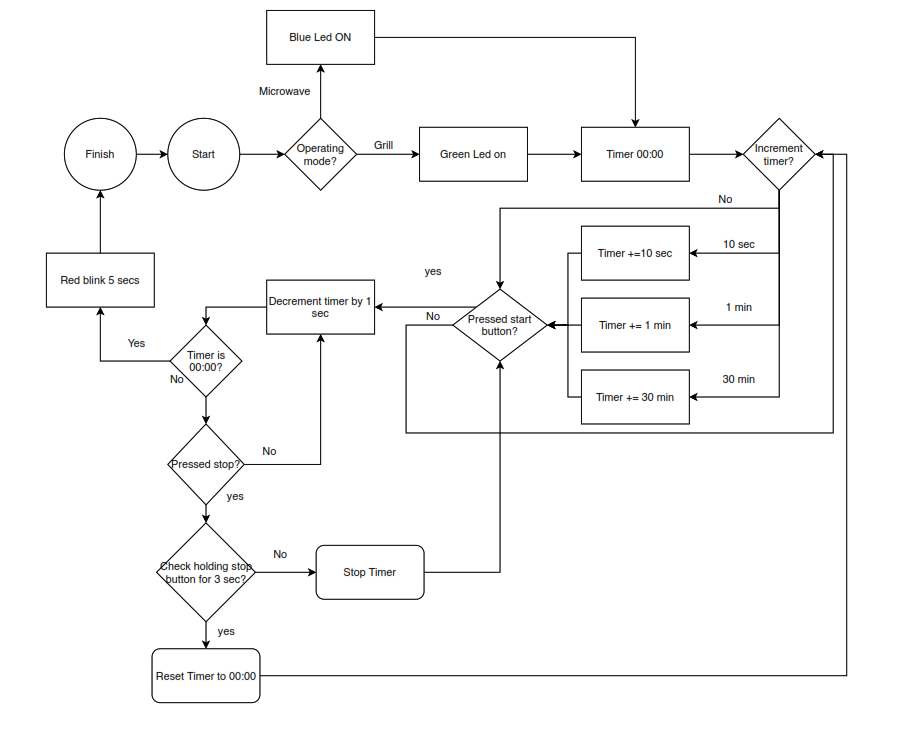


Figure start page

# Flow chart

A basic flowchart was made as a first step to identify the objectives and function required of the embedded system.



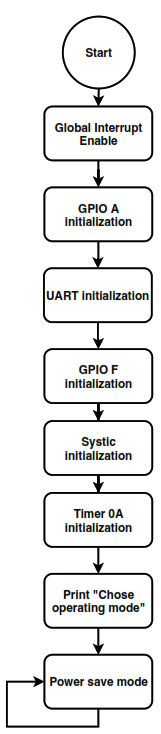
The Embedded system followed the interrupt procedures with no polling techniques and used many peripheral presents in the TIVA C microcontroller.

*M time in the Flowcharts refer to microwave timer that is set by increment command of user.*

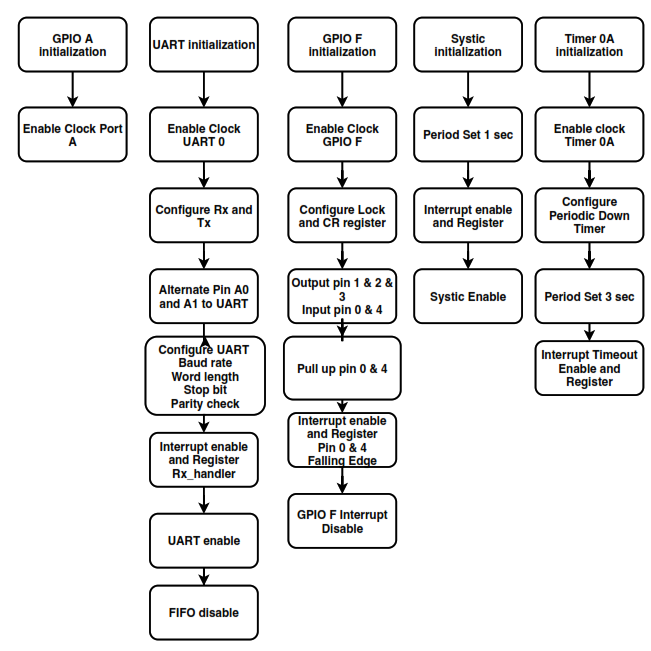
We divided the Flow chart to mainly 3 parts

* Main
* Initialization
* Interrupt handlers’ function

## Main

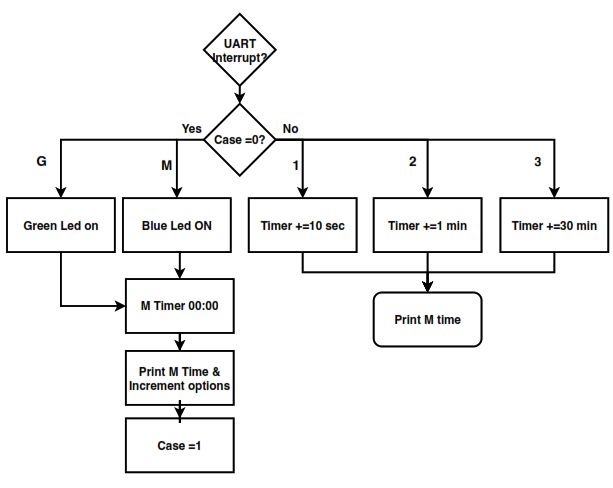


## Initialization

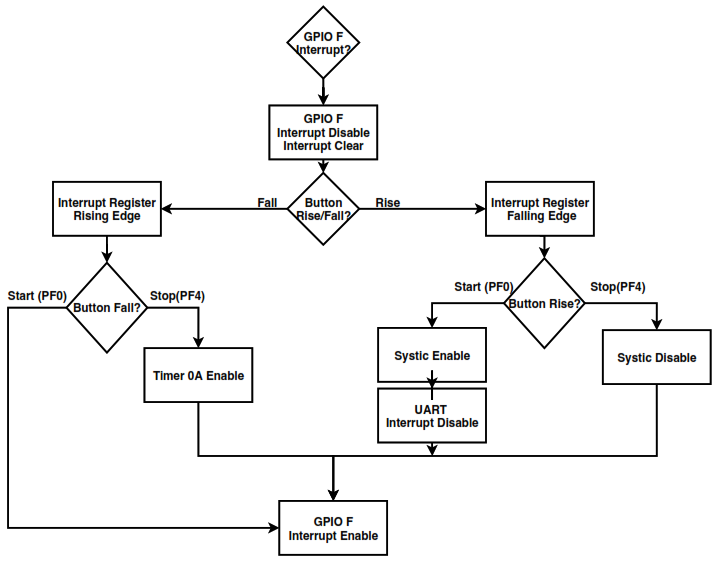


## Interrupt handler functions

### UART



### GPIOF (switches SW0 and SW4)



### Systic and GPTM (Timer 0A) interrupts

