

# CPE403 – Advanced Embedded Systems

---

## Design Assignment 0

---

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Angelo Nolasco

Email: Nolasco@unlv.nevada.edu

Github Repository link (root): [https://github.com/AngeloNol/Design\\_Assignments](https://github.com/AngeloNol/Design_Assignments)

Youtube Playlist link (root): [CPE 403 DAs](#)

---



## 1. Code for Tasks

```
//*****
*****

//
// blinky.c - Simple example to blink the on-board LED.
//
// Copyright (c) 2012-2017 Texas Instruments Incorporated. All rights reserved.
// Software License Agreement
//
// Texas Instruments (TI) is supplying this software for use solely and
// exclusively on TI's microcontroller products. The software is owned by
// TI and/or its suppliers, and is protected under applicable copyright
// laws. You may not combine this software with "viral" open-source
// software in order to form a larger program.
//
// THIS SOFTWARE IS PROVIDED "AS IS" AND WITH ALL FAULTS.
// 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. TI SHALL NOT, UNDER ANY
// CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
// DAMAGES, FOR ANY REASON WHATSOEVER.
//
// This is part of revision 2.1.4.178 of the EK-TM4C123GXL Firmware Package.
//
//*****

#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h"
#include "driverlib/debug.h"
#include "driverlib/gpio.h"
```

```
#include "driverlib/sysctl.h"
```

```
//*****  
*****
```

```
//
```

```
//! \addtogroup example_list
```

```
//! <h1>Blinky (blinky)</h1>
```

```
//!
```

```
//! A very simple example that blinks the on-board LED using direct register
```

```
//! access.
```

```
//
```

```
//*****  
*****
```

```
//*****  
*****
```

```
//
```

```
// The error routine that is called if the driver library encounters an error.
```

```
//
```

```
//*****  
*****
```

```
#ifdef DEBUG
```

```
void
```

```
__error__(char *pcFilename, uint32_t ui32Line)
```

```
{
```

```
    while(1);
```

```
}
```

```
#endif
```

```
//*****  
*****
```

```
//
```

```
// Blink the on-board LED.
```

```

//
//*****
*****

int
main(void)
{
    volatile uint32_t ui32Loop;

    //
    // Enable the GPIO port that is used for the on-board LED.
    //
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);

    //
    // Check if the peripheral access is enabled.
    //
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOF))
    {
    }

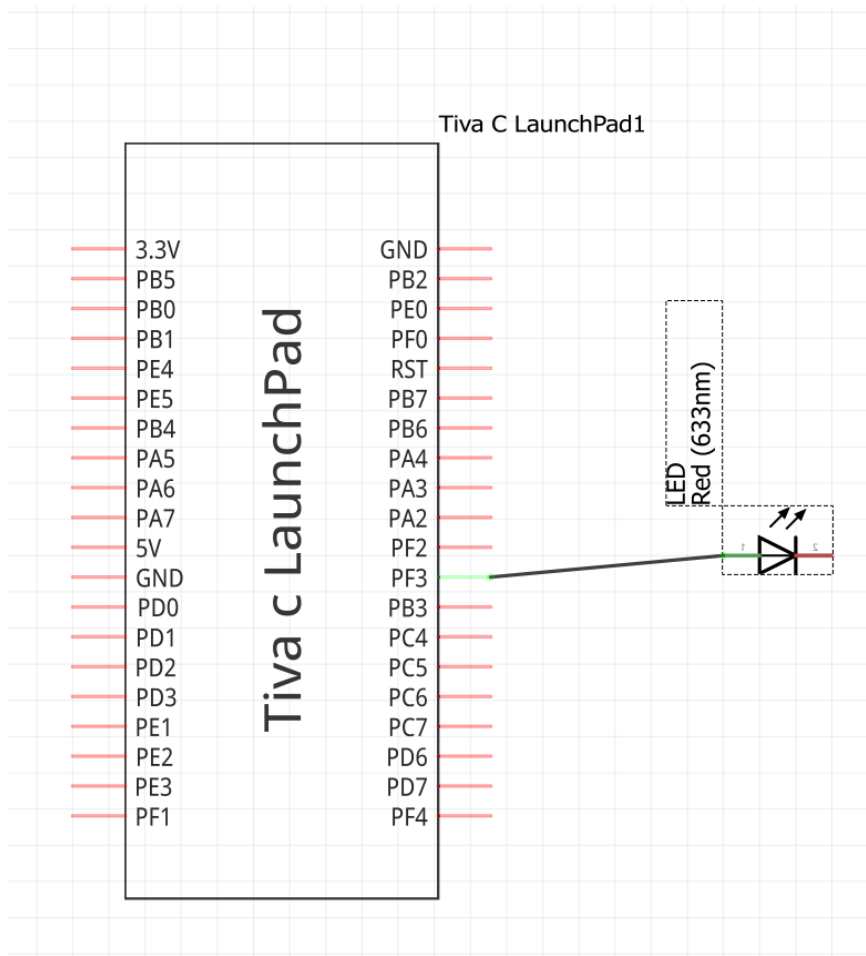
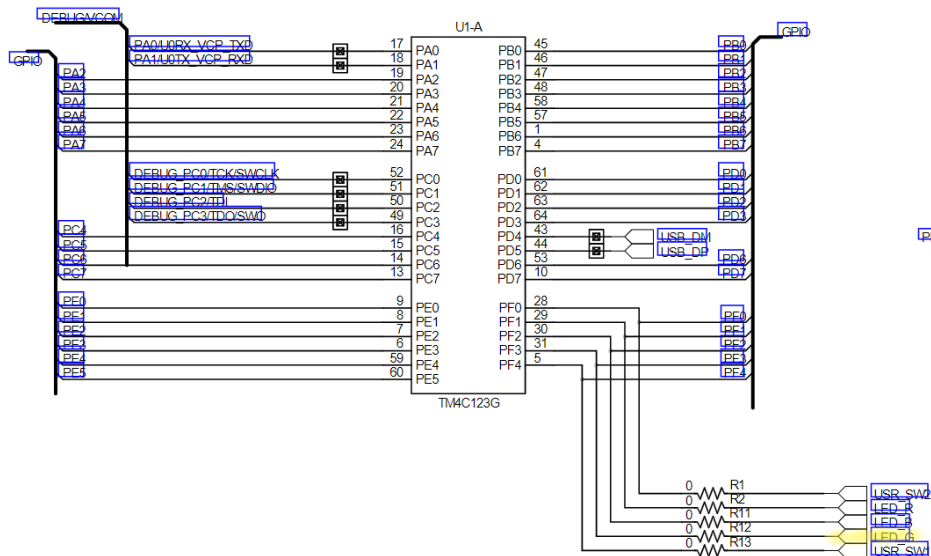
    //
    // Enable the GPIO pin for the LED (PF3). Set the direction as output, and
    // enable the GPIO pin for digital function.
    //
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_3);

    //
    // Loop forever.
    //
    while(1)
    {

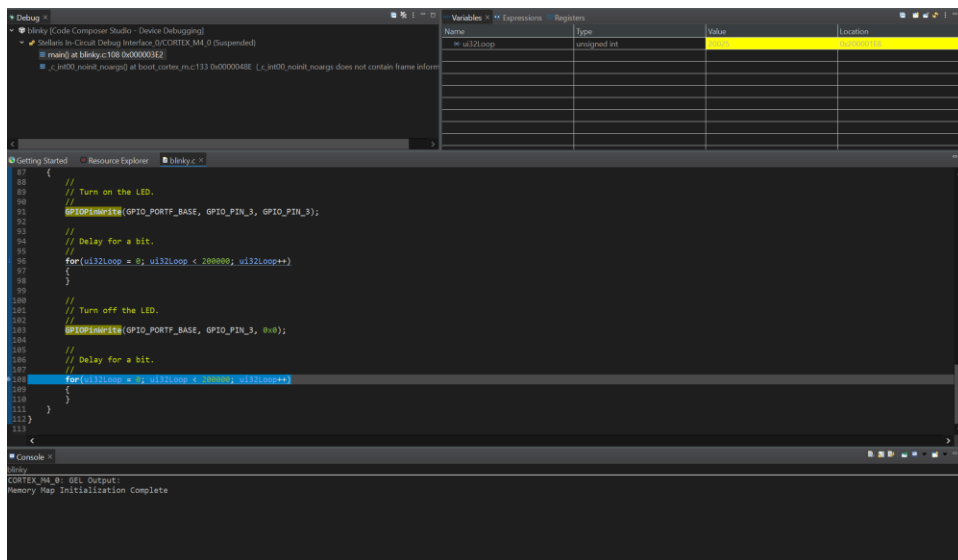
```

```
//  
// Turn on the LED.  
//  
GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, GPIO_PIN_3);  
  
//  
// Delay for a bit.  
//  
for(ui32Loop = 0; ui32Loop < 200000; ui32Loop++)  
{  
}  
  
//  
// Turn off the LED.  
//  
GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, 0x0);  
  
//  
// Delay for a bit.  
//  
for(ui32Loop = 0; ui32Loop < 200000; ui32Loop++)  
{  
}  
}  
}
```

2. Block diagram and/or Schematics showing the components, pins used, and interface.



### 3. Screenshots of the IDE, physical setup, debugging process



4. Declaration

I understand the Student Academic Misconduct Policy -  
<http://studentconduct.unlv.edu/misconduct/policy.html>

"This assignment submission is my own, original work".  
Angelo Nolasco