CPE403 – Advanced Embedded Systems

# Design Assignment #5

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Xianjie Cao

Email: [caox2@unlv.nevada.edu](mailto:caox2@unlv.nevada.edu)

Github Repository link (root): <https://github.com/c1029324620/Flat-White.git>

Youtube Playlist link (root): <https://www.youtube.com/playlist?list=PLY90fbcjLcrnosJGw9U__oC1jxRqwp8w8>

**Follow the submission guideline to be awarded points for this Assignment.**

Submit the following for all Assignments:

1. In the document, for each task submit the modified or included code (from the base code) with highlights and justifications of the modifications. Also include the comments. If no base code is provided, submit the base code for the first task only.
2. Create a private Github repository with a random name (no CPE/403, Lastname, Firstname). Place all labs under the root folder TIVAC, sub-folder named Assignment1, with one document and one video link file for each lab, place modified c files named as asng\_taskxx.c.
3. If multiple c files or other libraries are used, create a folder asng1\_t01 and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) with startup\_ccs.c and other include files, c) text file with youtube video links (see template).
5. Submit the doc file in canvas before the due date. The root folder of the github assignment directory should have the documentation and the text file with youtube video links.
6. Organize your youtube videos as playlist under the name “cpe403”. The playlist should have the video sequence arranged as submission or due dates.
7. Only submit pdf documents. Do not forget to upload this document in the github repository and in the canvas submission portal.
8. Code for Tasks. for each task submit the modified or included code (from the base code) with highlights and justifications of the modifications. Also include the comments. If no base code is provided, submit the base code for the first task only. Use separate page for each task.

Collector main.c Code:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Includes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#include** <xdc/std.h>

**#include** <xdc/runtime/Error.h>

**#include** <xdc/runtime/System.h>

**#include** <ti/sysbios/BIOS.h>

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

**#include** <ioc.h>

**#include** "sys\_ctrl.h"

**#include** "ti\_drivers\_config.h"

**#include** <inc/hw\_ccfg.h>

**#include** <inc/hw\_ccfg\_simple\_struct.h>

/\* Header files required for the temporary idle task function \*/

**#include** <ti/drivers/Power.h>

**#include** <ti/drivers/power/PowerCC26XX.h>

**#include** <aon\_rtc.h>

**#include** <prcm.h>

/\* Header files required to enable instruction fetch cache \*/

**#include** <ti/sysbios/hal/Hwi.h>

**#include** "cpu.h"

**#include** "collector.h"

**#ifdef** NV\_RESTORE

**#include** "macconfig.h"

**#ifdef** ONE\_PAGE\_NV

**#include** "nvocop.h"

**#else**

**#include** "nvocmp.h"

**#endif**

**#endif**

**#include** <string.h>

**#ifdef** OSAL\_PORT2TIRTOS

**#include** "macTask.h"

**#include**"rom\_jt\_154.h"

**#else**

**#include** "api\_mac.h"

**#include** "icall.h"

**#endif**

**#if** defined(RESET\_ASSERT)

**#include** "csf.h"

**#endif**

**#ifndef** USE\_DEFAULT\_USER\_CFG

**#include** "mac\_user\_config.h"

/\* MAC user defined configuration \*/

macUserCfg\_t macUser0Cfg[] = MAC\_USER\_CFG;

**#endif** /\* USE\_DEFAULT\_USER\_CFG \*/

**#ifndef** CUI\_DISABLE

**#include** "cui.h"

**#endif**

**#ifdef** USE\_ITM\_DBG

**#include** "itm.h"

**#endif**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Constants

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Assert Reasons \*/

**#define** MAIN\_ASSERT\_MAC 3

**#define** MAIN\_ASSERT\_HWI\_TIRTOS 4

**#define** RFC\_MODE\_BLE PRCM\_RFCMODESEL\_CURR\_MODE1

**#define** RFC\_MODE\_IEEE PRCM\_RFCMODESEL\_CURR\_MODE2

**#define** RFC\_MODE\_ANT PRCM\_RFCMODESEL\_CURR\_MODE4

**#define** RFC\_MODE\_EVERYTHING\_BUT\_ANT PRCM\_RFCMODESEL\_CURR\_MODE5

**#define** RFC\_MODE\_EVERYTHING PRCM\_RFCMODESEL\_CURR\_MODE6

/\* Extended Address offset in FCFG (LSB..MSB) \*/

**#define** EXTADDR\_OFFSET 0x2F0

**#if** defined(DeviceFamily\_CC13X2) || (DeviceFamily\_CC26X2)

**#define** APP\_TASK\_STACK\_SIZE 1536

**#else**

**#define** APP\_TASK\_STACK\_SIZE 960

**#endif**

**#define** APP\_TASK\_PRIORITY 1

**#define** SET\_RFC\_MODE(mode) HWREG( PRCM\_BASE + PRCM\_O\_RFCMODESEL ) = (mode)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

External Variables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**extern** ApiMac\_sAddrExt\_t ApiMac\_extAddr;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Global Variables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Task\_Struct appTask; /\* not static so you can see in ROV \*/

**static** uint8\_t appTaskStack[APP\_TASK\_STACK\_SIZE];

**#ifdef** OSAL\_PORT2TIRTOS

**static** uint8\_t \_macTaskId;

**#endif**

/\*

When assert happens, this field will be filled with the reason:

MAIN\_ASSERT\_HWI\_TIRTOS or MAIN\_ASSERT\_MAC

\*/

uint8 Main\_assertReason = 0;

**#ifdef** NV\_RESTORE

mac\_Config\_t Main\_user1Cfg =

{ 0 };

**#endif**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Local Variables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Used to check for a valid extended address \*/

**static** **const** uint8\_t dummyExtAddr[] =

{ 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };

**extern** **void** **Board\_init**(**void**);

**#ifdef** NV\_RESTORE

**#ifdef** ONE\_PAGE\_NV

/\* NVOCOP load API pointers \*/

**static** **void** NVOCOP\_loadApiPtrs(NVINTF\_nvFuncts\_t \*pfn)

{

// Load caller's structure with pointers to the NV API functions

pfn->initNV = &NVOCOP\_initNV;

pfn->compactNV = &NVOCOP\_compactNV;

pfn->createItem = NULL;

pfn->deleteItem = &NVOCOP\_deleteItem;

pfn->readItem = &NVOCOP\_readItem;

pfn->writeItem = &NVOCOP\_writeItem;

pfn->writeItemEx = NULL;

pfn->getItemLen = NULL;

}

**#endif**

**#endif**

/\*!

\* @brief Reads the IEEE extended MAC address from the CCFG

\* @param addr - Extended address pointer

\*/

**static** **inline** **void** **CCFGRead\_IEEE\_MAC**(ApiMac\_sAddrExt\_t addr)

{

uint32\_t macAddr = (( HWREG(

CCFG\_BASE + CCFG\_O\_IEEE\_MAC\_0 ) &

CCFG\_IEEE\_MAC\_0\_ADDR\_M ) >>

CCFG\_IEEE\_MAC\_0\_ADDR\_S );

**memcpy**(addr, (uint8\_t \*)&macAddr, (APIMAC\_SADDR\_EXT\_LEN / 2));

macAddr = (( HWREG(

CCFG\_BASE + CCFG\_O\_IEEE\_MAC\_1 ) &

CCFG\_IEEE\_MAC\_1\_ADDR\_M ) >>

CCFG\_IEEE\_MAC\_1\_ADDR\_S );

**memcpy**(addr + (APIMAC\_SADDR\_EXT\_LEN / 2), (uint8\_t \*)&macAddr,

(APIMAC\_SADDR\_EXT\_LEN / 2));

}

/\*!

\* @brief Fill in your own assert function.

\*

\* @param assertReason - reason: MAIN\_ASSERT\_HWI\_TIRTOS or

\* MAIN\_ASSERT\_MAC

\*/

**void** **Main\_assertHandler**(uint8\_t assertReason)

{

Main\_assertReason = assertReason;

**#if** defined(RESET\_ASSERT)

Csf\_assertInd(assertReason);

/\* Pull the plug and start over \*/

SysCtrlSystemReset();

**#else**

Hwi\_disable();

**while**(1)

{

/\* Put you code here to do something if in assert \*/

}

**#endif**

}

/\*!

\* @brief Main task function

\*

\* @param a0 -

\* @param a1 -

\*/

Void **appTaskFxn**(UArg a0, UArg a1)

{

/\* The following code encapsulated in TI\_154STACK\_FPGA flag is used for

\* internal FPGA evaluation of the 15.4 Stack and should not be used with

\* TI hardware platforms. \*/

**#ifdef** TI\_154STACK\_FPGA

/\* FPGA build disables POWER constraints \*/

Power\_setConstraint(PowerCC26XX\_IDLE\_PD\_DISALLOW);

Power\_setConstraint(PowerCC26XX\_SB\_DISALLOW);

IOCPortConfigureSet(IOID\_20, IOC\_PORT\_RFC\_GPO0, IOC\_STD\_OUTPUT);

IOCPortConfigureSet(IOID\_18, IOC\_PORT\_RFC\_GPI0, IOC\_STD\_INPUT);

// configure RF Core SMI Command Link

IOCPortConfigureSet(IOID\_22, IOC\_IOCFG0\_PORT\_ID\_RFC\_SMI\_CL\_OUT, IOC\_STD\_OUTPUT);

IOCPortConfigureSet(IOID\_21, IOC\_IOCFG0\_PORT\_ID\_RFC\_SMI\_CL\_IN, IOC\_STD\_INPUT);

**#endif**

**#ifndef** OSAL\_PORT2TIRTOS

/\* Initialize ICall module \*/

ICall\_init();

**#endif**

/\* Copy the extended address from the CCFG area \*/

CCFGRead\_IEEE\_MAC(ApiMac\_extAddr);

/\* Check to see if the CCFG IEEE is valid \*/

**if**(**memcmp**(ApiMac\_extAddr, dummyExtAddr, APIMAC\_SADDR\_EXT\_LEN) == 0)

{

/\* No, it isn't valid. Get the Primary IEEE Address \*/

**memcpy**(ApiMac\_extAddr, (uint8\_t \*)(FCFG1\_BASE + EXTADDR\_OFFSET),

(APIMAC\_SADDR\_EXT\_LEN));

}

**#ifdef** NV\_RESTORE

/\* Setup the NV driver \*/

**#ifdef** ONE\_PAGE\_NV

NVOCOP\_loadApiPtrs(&Main\_user1Cfg.nvFps);

**#else**

NVOCMP\_loadApiPtrs(&Main\_user1Cfg.nvFps);

**#endif**

**if**(Main\_user1Cfg.nvFps.initNV)

{

Main\_user1Cfg.nvFps.initNV( NULL);

}

**#endif**

**#ifdef** OSAL\_PORT2TIRTOS

/\* Initialize the application \*/

Collector\_init(\_macTaskId);

**#else**

ICall\_createRemoteTasks();

/\* Initialize the application \*/

Collector\_init();

**#endif**

/\* Kick off application - Forever loop \*/

**while**(1)

{

Collector\_process();

}

}

/\*!

\* @brief TIRTOS HWI Handler. The name of this function is set to

\* M3Hwi.excHandlerFunc in app.cfg, you can disable this by

\* setting it to null.

\*

\* @param excStack - TIROS variable

\* @param lr - TIROS variable

\*/

xdc\_Void **Main\_excHandler**(UInt \*excStack, UInt lr)

{

/\* User defined function \*/

Main\_assertHandler(MAIN\_ASSERT\_HWI\_TIRTOS);

}

/\*!

\* @brief HAL assert handler required by OSAL memory module.

\*/

**void** **assertHandler**(**void**)

{

/\* User defined function \*/

Main\_assertHandler(MAIN\_ASSERT\_MAC);

}

/\*!

\* @brief "main()" function - starting point

\*/

**int** **main**(**void**)

{

Task\_Params taskParams;

**#ifndef** USE\_DEFAULT\_USER\_CFG

macUser0Cfg[0].pAssertFP = assertHandler;

**#endif**

/\*

Initialization for board related stuff such as LEDs

following TI-RTOS convention

\*/

Board\_init();

**#if** !defined(POWER\_MEAS) && !defined(CUI\_DISABLE)

/\* Initialize CUI UART\*/

CUI\_params\_t cuiParams;

**CUI\_paramsInit**(&cuiParams);

**#ifdef** MT\_CSF

cuiParams.manageUart = **false**;

**#endif** //MT\_CSF

// One-time initialization of the CUI

// All later CUI\_\* functions will be ignored if this isn't called

**CUI\_init**(&cuiParams);

**#endif**

**#ifdef** OSAL\_PORT2TIRTOS

\_macTaskId = macTaskInit(macUser0Cfg);

**#endif**

/\* Configure task. \*/

Task\_Params\_init(&taskParams);

taskParams.stack = appTaskStack;

taskParams.stackSize = APP\_TASK\_STACK\_SIZE;

taskParams.priority = APP\_TASK\_PRIORITY;

Task\_construct(&appTask, appTaskFxn, &taskParams, NULL);

**#ifdef** USE\_ITM\_DBG

ITM\_config itm\_config =

{

48000000,

ITM\_6000000

};

ITM\_initModule(itm\_config);

ITM\_enableModule();

**#endif** /\* USE\_ITM\_DBG \*/

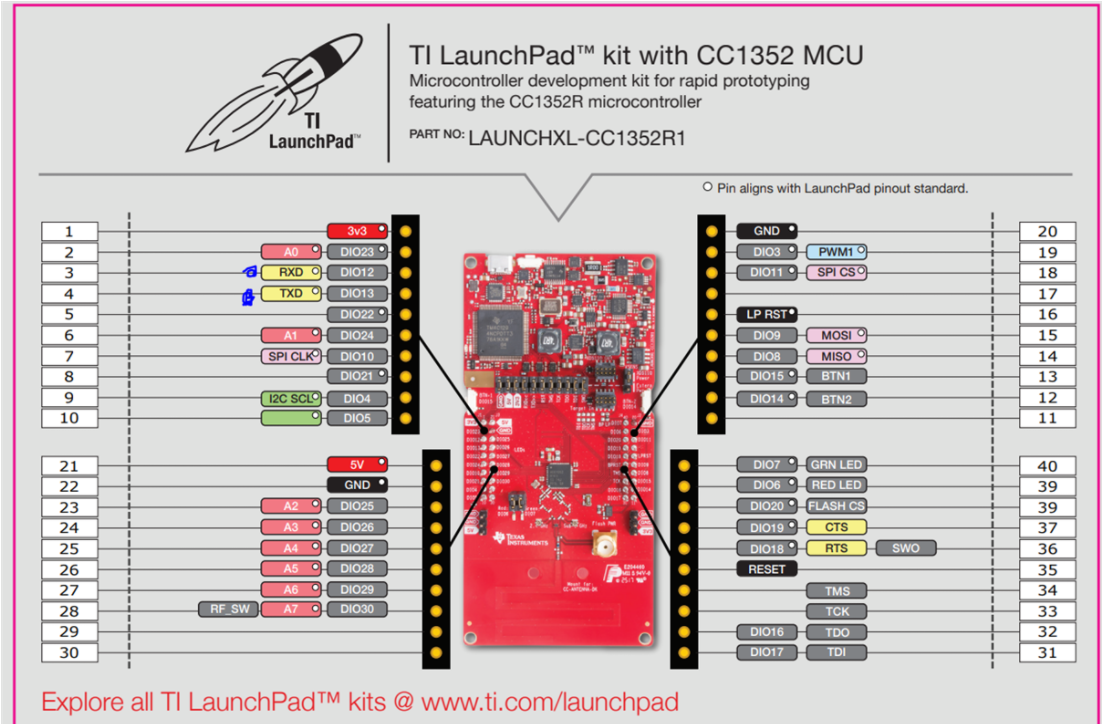
BIOS\_start(); /\* enable interrupts and start SYS/BIOS \*/

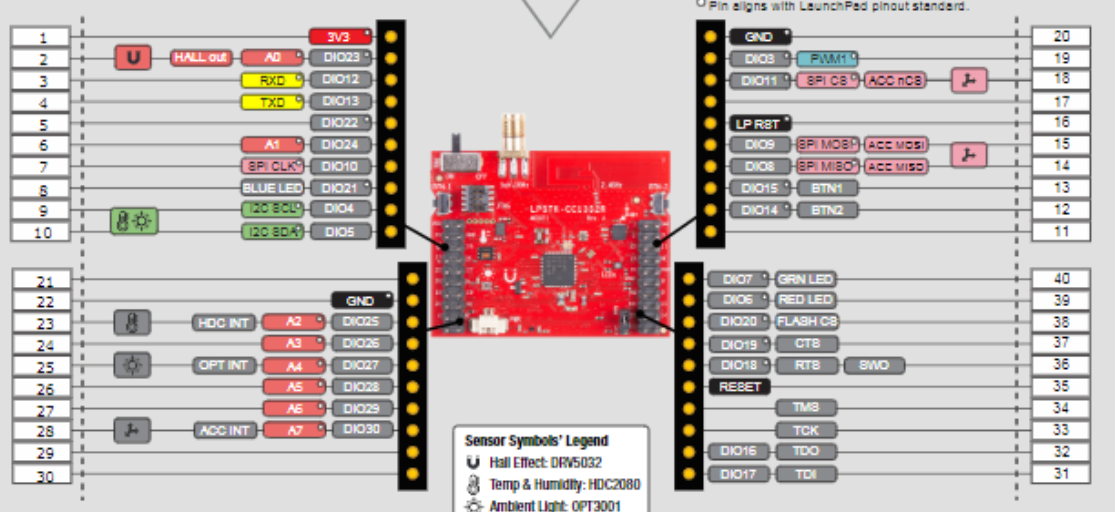
**return** (0);

}

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

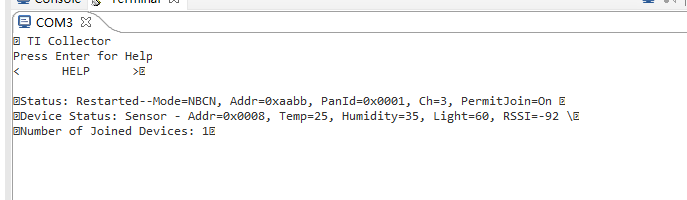
Components used: CC1352R1 Launch Pad and LPSTK, Code composer studio.

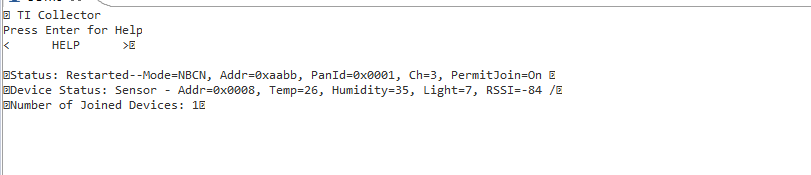




1. Screenshots of the IDE, physical setup, debugging process - Provide screenshot of successful compilation, screenshots of registers, variables, graphs, etc.

**LPSTK TI 15.4:**

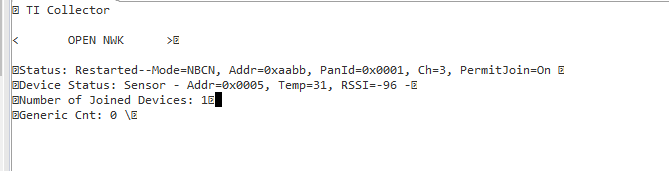




**Sensor and Collector – Project zero:**



**Adding a New Sensor:**



1. Declaration

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

“This assignment submission is my own, original work”.

Xianjie Cao