CPE403 – Advanced Embedded Systems

# Design Assignment #2

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>

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.

**Task 1:**

**#include** <stdbool.h>

**#include** <stdint.h>

**#include** <stdio.h>

**#include** <stdarg.h>

**#include** "inc/hw\_i2c.h"

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "inc/hw\_gpio.h"

**#include** "driverlib/i2c.h"

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

**#include** "driverlib/gpio.h"

**#include** "driverlib/pin\_map.h"

**#include** "utils/uartstdio.h"

**#include** "driverlib/uart.h"

**#include** "math.h"

**#include** "IQmath/IQmathlib.h"

**#define** TMP006\_ADDR (0x40)

**#define** TMP006\_ID (0xFF)

**#define** M\_ID (0xFE)

**#define** TMP006\_COMMAND\_BIT (0x01)

**#define** V\_OBJ (0x00)

**#define** T\_DIE (0x01)

**#define** TMP006\_CONFIG (0x02)

**#define** ONE 1.0

**#define** LSB 1.5625e-7

**#define** S0 6e-5

**#define** A1 1.75e-3

**#define** A2 -1.678e-5

**#define** B0 -2.94e-5

**#define** B1 -5.7e-7

**#define** B2 4.63e-9

**#define** C2 13.4

**#define** T\_REF 25.13

uint16\_t XOR = 0xFFFF;

uint16\_t TMP006\_Voltage;

uint16\_t TMP006\_Temp;

**double** temp;

**double** TDIE;

**double** F;

**double** S;

**void** **ConfigureUART0**(**void**)

//Configures the UART to run at 115200 baud rate

{

// ENABLE PERIPHERAL UART 0

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_UART0);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOA);

// ENABLE GPIO PORT A,FOR UART

**GPIOPinConfigure**(GPIO\_PA0\_U0RX); // PA0 IS CONFIGURED TO UART RX

**GPIOPinConfigure**(GPIO\_PA1\_U0TX); // PA1 IS CONFIGURED TO UART TX

**GPIOPinTypeUART**(GPIO\_PORTA\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1);

**UARTClockSourceSet**(UART0\_BASE, UART\_CLOCK\_PIOSC);

**UARTStdioConfig**(0, 115200, 16000000);

}

**void** **I2C\_Init**()

{

// Enable I2C0 peripheral

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_I2C0);

**SysCtlPeripheralReset**(SYSCTL\_PERIPH\_I2C0);

// Enable GPIO Port B to be used for I2C0

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOB);

**GPIOPinConfigure**(GPIO\_PB2\_I2C0SCL);

**GPIOPinConfigure**(GPIO\_PB3\_I2C0SDA);

// Configure the pin muxing for I2C0 functions on Port B2 and B3

**GPIOPinTypeI2CSCL**(GPIO\_PORTB\_BASE, GPIO\_PIN\_2);

**GPIOPinTypeI2C**(GPIO\_PORTB\_BASE, GPIO\_PIN\_3);

**I2CMasterInitExpClk**(I2C0\_BASE, **SysCtlClockGet**(), **false**);

// Set write mode

**UARTprintf**("I2C Master communication with Slave Address\n");

**SysCtlDelay**(10000);

**UARTprintf**("I2C Init complete!\n");

**SysCtlDelay**(5000000);

}

**void** **I2C0\_Send16**(uint8\_t slave\_addr, uint8\_t pointer\_reg, uint16\_t TxData)

{

uint8\_t data;

**I2CMasterSlaveAddrSet**(I2C0\_BASE, slave\_addr, **false**);

**I2CMasterDataPut**(I2C0\_BASE, pointer\_reg);

**I2CMasterControl**(I2C0\_BASE, I2C\_MASTER\_CMD\_BURST\_SEND\_START);

**while**(**I2CMasterBusy**(I2C0\_BASE));

// MSB First

data = (uint8\_t)((TxData >> 8) & 0x00FF);

**I2CMasterDataPut**(I2C0\_BASE, data);

**while**(**I2CMasterBusy**(I2C0\_BASE));

//LSB Later

data = (uint8\_t)(TxData & 0x00FF);

**I2CMasterDataPut**(I2C0\_BASE, data);

**I2CMasterControl**(I2C0\_BASE, I2C\_MASTER\_CMD\_BURST\_SEND\_FINISH);

**while**(**I2CMasterBusy**(I2C0\_BASE));

}

uint16\_t **I2C\_Read**(uint8\_t slave\_addr, uint8\_t pointer\_reg)

{

uint8\_t UpperByte = 0;

uint8\_t LowerByte = 0;

uint16\_t Data = 0;

// Specify Slave device address to write to

**I2CMasterSlaveAddrSet**(I2C0\_BASE, slave\_addr, **false**);

// Send Register address on Slave device

**I2CMasterDataPut**(I2C0\_BASE, pointer\_reg);

// Initiate send of register address from Master to Slave

**I2CMasterControl**(I2C0\_BASE, I2C\_MASTER\_CMD\_SINGLE\_SEND);

**while**(**I2CMasterBusy**(I2C0\_BASE));

// Set read mode

**I2CMasterSlaveAddrSet**(I2C0\_BASE, slave\_addr, **true**);

// Get first byte from slave and ack for more

**I2CMasterControl**(I2C0\_BASE, I2C\_MASTER\_CMD\_BURST\_RECEIVE\_START);

**while**(**I2CMasterBusy**(I2C0\_BASE));

UpperByte = **I2CMasterDataGet**(I2C0\_BASE);

// Get second byte from slave and nack for complete

**I2CMasterControl**(I2C0\_BASE, I2C\_MASTER\_CMD\_BURST\_RECEIVE\_CONT);

**while**(**I2CMasterBusy**(I2C0\_BASE));

LowerByte = **I2CMasterDataGet**(I2C0\_BASE);

Data = (UpperByte<<8) | LowerByte;

**return** Data;

}

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

{

**SysCtlClockSet**(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

ConfigureUART0();

I2C\_Init();

**double** IQ\_NUM\_Mul\_Const = **pow**(2, -24);

\_iq24 v\_obj, t\_die, s0, a1, a2, t\_ref, b0, b1, b2, c2, s, t\_die\_sub\_t\_ref, v\_os, f\_v\_obj, v\_obj\_sub\_v\_os, one;

\_iq24 lsb;

s0 = \_IQ24(S0);

a1 = \_IQ24(A1);

a2 = \_IQ24(A2);

t\_ref = \_IQ24(T\_REF);

b0 = \_IQ24(B0);

b1 = \_IQ24(B1);

b2 = \_IQ24(B2);

c2 = \_IQ24(C2);

lsb = \_IQ24(LSB);

one = \_IQ24(ONE);

**while**(1)

{

TMP006\_Voltage = I2C\_Read(TMP006\_ADDR, V\_OBJ);

**if**(TMP006\_Voltage > 0x7FFF) //full scale from 0 to 7FFFh, if the reading is negative, get its absolute value.

{

//UARTprintf("negative\n");

TMP006\_Voltage = (TMP006\_Voltage ^ XOR) + 1;

}

//TMP006\_Voltage = TMP006\_Voltage;

**UARTprintf**("I2C V\_OBJ: %d\n", TMP006\_Voltage);

TMP006\_Temp = (I2C\_Read(TMP006\_ADDR, T\_DIE)>>2)/32 ; //Converting the integer temperature result of the TMP006

//to physical temperature is done by rightshifting the last two LSBs

t\_die = \_IQ24(TMP006\_Temp); //followed by a divide-by-32 of TREG to obtain the physical temperature result in degreesCelsius. T

v\_obj = \_IQ24(TMP006\_Voltage);

v\_obj = **\_IQ24mpy**(v\_obj, lsb);

**UARTprintf**("I2C T\_DIE: %d C\n", TMP006\_Temp);

t\_die\_sub\_t\_ref = t\_die - t\_ref;

s = **\_IQ24mpy**(s0,(**\_IQ24mpy**(a1, t\_die\_sub\_t\_ref) + **\_IQ24mpy**(a2, (**\_IQ24mpy**(t\_die\_sub\_t\_ref, t\_die\_sub\_t\_ref))) + one));

v\_os = **\_IQ24mpy**(t\_die\_sub\_t\_ref, b1) + **\_IQ24mpy**(b2,(**\_IQ24mpy**(t\_die\_sub\_t\_ref, t\_die\_sub\_t\_ref))) + b0;

v\_obj\_sub\_v\_os = v\_obj - v\_os;

f\_v\_obj = v\_obj\_sub\_v\_os + **\_IQ24mpy**(c2,**\_IQ24mpy**(v\_obj\_sub\_v\_os, v\_obj\_sub\_v\_os));

//Convert to double.

F = f\_v\_obj \* IQ\_NUM\_Mul\_Const;//convert to double

S = s \* IQ\_NUM\_Mul\_Const;

TDIE = t\_die \* IQ\_NUM\_Mul\_Const;

temp = **pow**((**pow**(TDIE, 4.0) + F/S), 0.25);

**SysCtlDelay**(5000000);

}

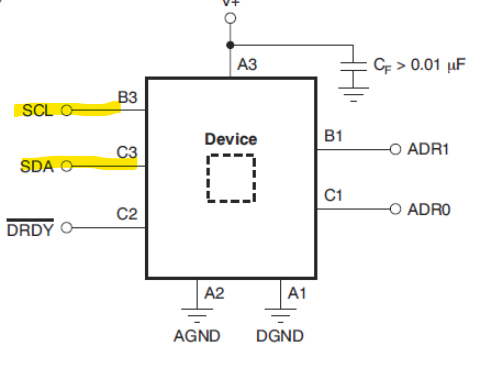
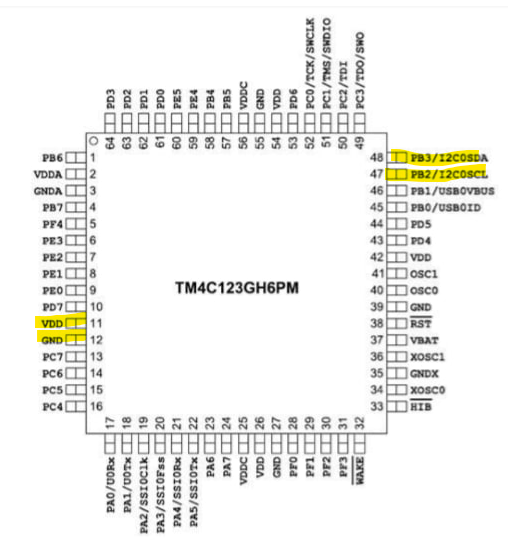
}

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

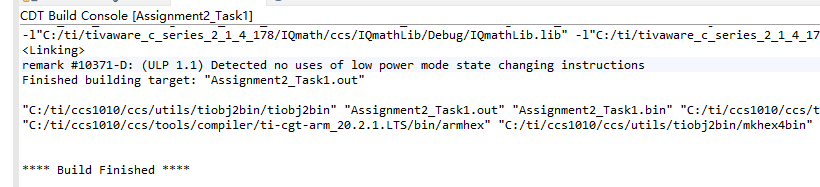
Components used: TM4C123GXl, BoosterPack MKII, TMP006 sensor, Jump wires, code composer studio.

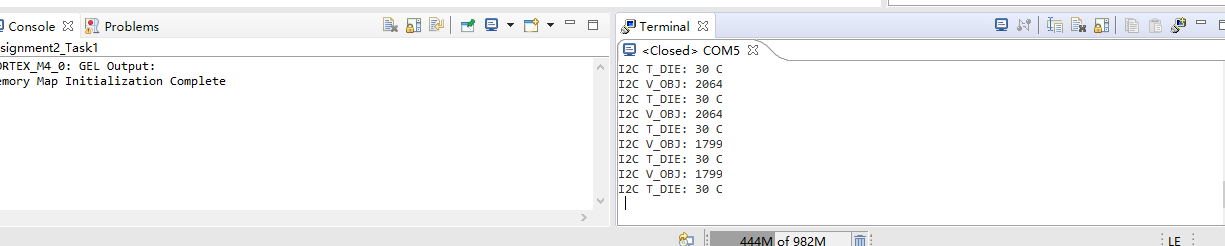
Pins used: 3.3V, 5V, GND, PB2, PB3.

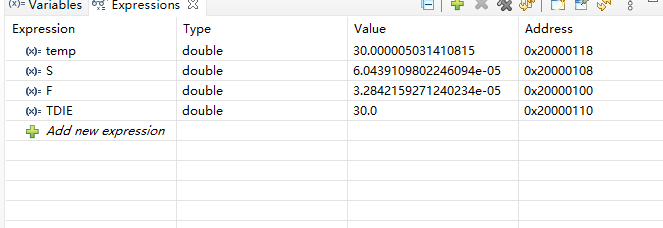
Interface: I2C



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









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