CPE403 – Advanced Embedded Systems

Design Assignment 02

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Do Le

Email: led2@unlv.nevada.edu

Github Repository link (root): https://github.com/DoVietLe/AES

Youtube Playlist link (root):
https://www.youtube.com/playlist?list=PLFfzhLPj7fvOz1lm2Vd9DevkHetoyvRQ6

1. 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.

   main.c Program

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/tm4c123gh6pm.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/sysctl.h"
#include "driverlib/interrupt.h"
#include "driverlib/gpio.h"
#include "driverlib/timer.h"
#include "driverlib/uart.h"
#include "driverlib/adc.h"
#include "utils/uartstdio.c"
#define MATH_TYPE FLOAT_MATH        // Makes sure that IQmath uses floating
point math.
#include "IQmath/IQmathLib.h"
#include "i2c.h"

// UART Stuff
#define BAUD_RATE 115200
#define GPIO_PA0_U0RX 0x00000001
#define GPIO_PA1_U0TX 0x00000401

// Addresses
#define SLAVE_ADDRESS 0x40
#define V_OBJ_ADDRESS 0x00
```

```c
#define T_AMB_ADDRESS 0x01
#define CONFIGURATION_ADDRESS 0x02
#define MANU_ID_ADDRESS 0xFE
#define DEVICE_ID_ADDRESS 0xFF

// Stuff for calculation of temperature
#define S0 6.0e-14
#define A1 1.75e-3
#define A2 -1.678e-5
#define T_REF 298.15
#define B0 -2.94e-5
#define B1 -5.7e-7
#define B2 4.63e-9
#define C2 13.4
#define MSB_VAL 156.25e-9           // Value of least significant bit in
voltage of V_OBJ register value.

// Makes sure that all constants are stored as IQ values.
const _iq20 qS0 = _IQ20(S0);
const _iq20 qa1 = _IQ20(A1);
const _iq20 qa2 = _IQ20(A2);
const _iq20 qtRef = _IQ20(T_REF);
const _iq20 qb0 = _IQ20(B0);
const _iq20 qb1 = _IQ20(B1);
const _iq20 qb2 = _IQ20(B2);
const _iq20 qc2 = _IQ20(C2);
const _iq20 qMSB = _IQ20(MSB_VAL);
_iq20 s, Vos, fVobj, qvOBJ, qtAMB, qTemperature, qTDiff;

uint16_t vOBJ;
uint16_t tAMB;
uint16_t deviceId, manuId;

void setupTimer() {
    uint32_t loadVal;

    // Calculates the cycle values for a 5s delay.
    loadVal = SysCtlClockGet();

    TimerConfigure(WTIMER0_BASE, TIMER_CFG_PERIODIC);
    TimerLoadSet(WTIMER0_BASE, TIMER_A, loadVal);
}

void setupUART() {
    GPIOPinConfigure(GPIO_PA0_U0RX);
    GPIOPinConfigure(GPIO_PA1_U0TX);
    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
    UARTClockSourceSet(UART0_BASE, UART_CLOCK_PIOSC);
    UARTStdioConfig(0, 115200, 16000000);
}

void setupTimerInterrupt() {
    IntEnable(INT_WTIMER0A);
    TimerIntEnable(WTIMER0_BASE, TIMER_TIMA_TIMEOUT);
}

int main(void) {
    // Sets up the system clock.
    SysCtlClockSet(SYSCTL_SYSDIV_5 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    FPULazyStackingEnable();
    FPUEnable();

    // Enables peripherals.
    SysCtlPeripheralEnable(SYSCTL_PERIPH_WTIMER0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);

    initI2C1();

    // Sets up peripherals.
    setupTimer();
    setupUART();

    // Sets up interrupts.
    setupTimerInterrupt();
    IntMasterEnable();

    // Enables stuff.
    TimerEnable(WTIMER0_BASE, TIMER_A);

    //I2C1_Send16(SLAVE_ADDRESS, CONFIGURATION_ADDRESS, 0b01111000011<<6);

    UARTprintf("Starting...\n\r");

    //I2C1_Send16(SLAVE_ADDRESS, CONFIGURATION_ADDRESS, (1<<15));
    while (1)
    {
    }
}

void timerhandler(void) {
    // Clears interrupt flag.
    TimerIntClear(WTIMER0_BASE, TIMER_TIMA_TIMEOUT);

    float tempKelvin, tempCelsius, tempFahrenheit;

    // Used to verify communication with the device.
    //deviceId = I2C1_Read16(SLAVE_ADDRESS, DEVICE_ID_ADDRESS);
    //manuId = I2C1_Read16(SLAVE_ADDRESS, MANU_ID_ADDRESS);

    // Used to calculate the temperature.
    vOBJ = I2C1_Read16(SLAVE_ADDRESS, V_OBJ_ADDRESS);
    tAMB = (I2C1_Read16(SLAVE_ADDRESS, T_AMB_ADDRESS) >> 2);
//*
    // Calculates the temperature and voltage.
    qvOBJ = _IQ20mpy(_IQ20( (float)vOBJ ), qMSB);                  //
Calculates the voltage in Volts.
    qtAMB = _IQdiv32(_IQ20( (float)tAMB )) + _IQ20(273.15);         //
Calculates the temperature in Kelvin.

    // Determines object temperature.
    qTDiff = qtAMB - qtRef;
    s = qS0*(_IQ20(1.0) + qa1*qTDiff + qa2*qTDiff*qTDiff);
    Vos = qb0 + qb1*qTDiff + qb2*qTDiff*qTDiff;
    fVobj = (qvOBJ - Vos) + qc2*(qvOBJ - Vos)*(qvOBJ - Vos);
    qTemperature = _IQ20sqrt(_IQ20sqrt(qtAMB*qtAMB*qtAMB*qtAMB + fVobj/s));
    tempKelvin = _IQ20toF(qTemperature);
    tempCelsius = _IQ20toF(qTemperature - _IQ20(273.15));
    tempFahrenheit = _IQ20toF( _IQ20mpy(_IQ20(1.8), qTemperature - 
_IQ20(273.15)) + _IQ20(32.0) );

//*/

    // Displays the temperature.
    UARTprintf("(V_OBJ = %d, T_AMB = %d)\n\r"
            "Temperature: %dK\n\r"
            "             %dC\n\r"
            "             %dF\n\r\n\r",
            vOBJ, tAMB, (uint32_t)tempKelvin, (uint32_t)tempCelsius, 
(uint32_t)tempFahrenheit);
}
```

i2c.c Program

```
_#include <stdbool.h>
#include <stdint.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 "i2c.h"

void initI2C1(void)
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_I2C1);

    GPIOPinConfigure(GPIO_PA6_I2C1SCL);
    GPIOPinConfigure(GPIO_PA7_I2C1SDA);

    GPIOPinTypeI2CSCL(GPIO_PORTA_BASE, GPIO_PIN_6);
    GPIOPinTypeI2C(GPIO_PORTA_BASE, GPIO_PIN_7);

    I2CMasterInitExpClk(I2C1_BASE, SysCtlClockGet(), false);
    //clear I2C FIFOs
    HWREG(I2C1_BASE + I2C_O_FIFOCTL) = 80008000;
}

//sends an I2C command to the specified slave
uint16_t I2C1_Read16(uint8_t slave_addr, uint8_t pointer_reg)
```

```
    uint16_t data;
    uint16_t RxData;
    I2CMasterSlaveAddrSet(I2C1_BASE, slave_addr, false);
    I2CMasterDataPut(I2C1_BASE, pointer_reg);
    I2CMasterControl(I2C1_BASE, I2C_MASTER_CMD_BURST_SEND_START);
    while(I2CMasterBusy(I2C1_BASE));
    I2CMasterSlaveAddrSet(I2C1_BASE, slave_addr, true);
    I2CMasterControl(I2C1_BASE, I2C_MASTER_CMD_BURST_RECEIVE_START);
    while(I2CMasterBusy(I2C1_BASE));
    //MSB first
    data = I2CMasterDataGet(I2C1_BASE);

    RxData = (uint16_t)(data << 8);
    I2CMasterControl(I2C1_BASE, I2C_MASTER_CMD_BURST_RECEIVE_CONT);
    while(I2CMasterBusy(I2C1_BASE));
    //LSB later
    data = I2CMasterDataGet(I2C1_BASE);
    RxData |= (uint16_t)(data);
    I2CMasterControl(I2C1_BASE, I2C_MASTER_CMD_BURST_SEND_FINISH);
    while(I2CMasterBusy(I2C1_BASE));

    return RxData;
}
```

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

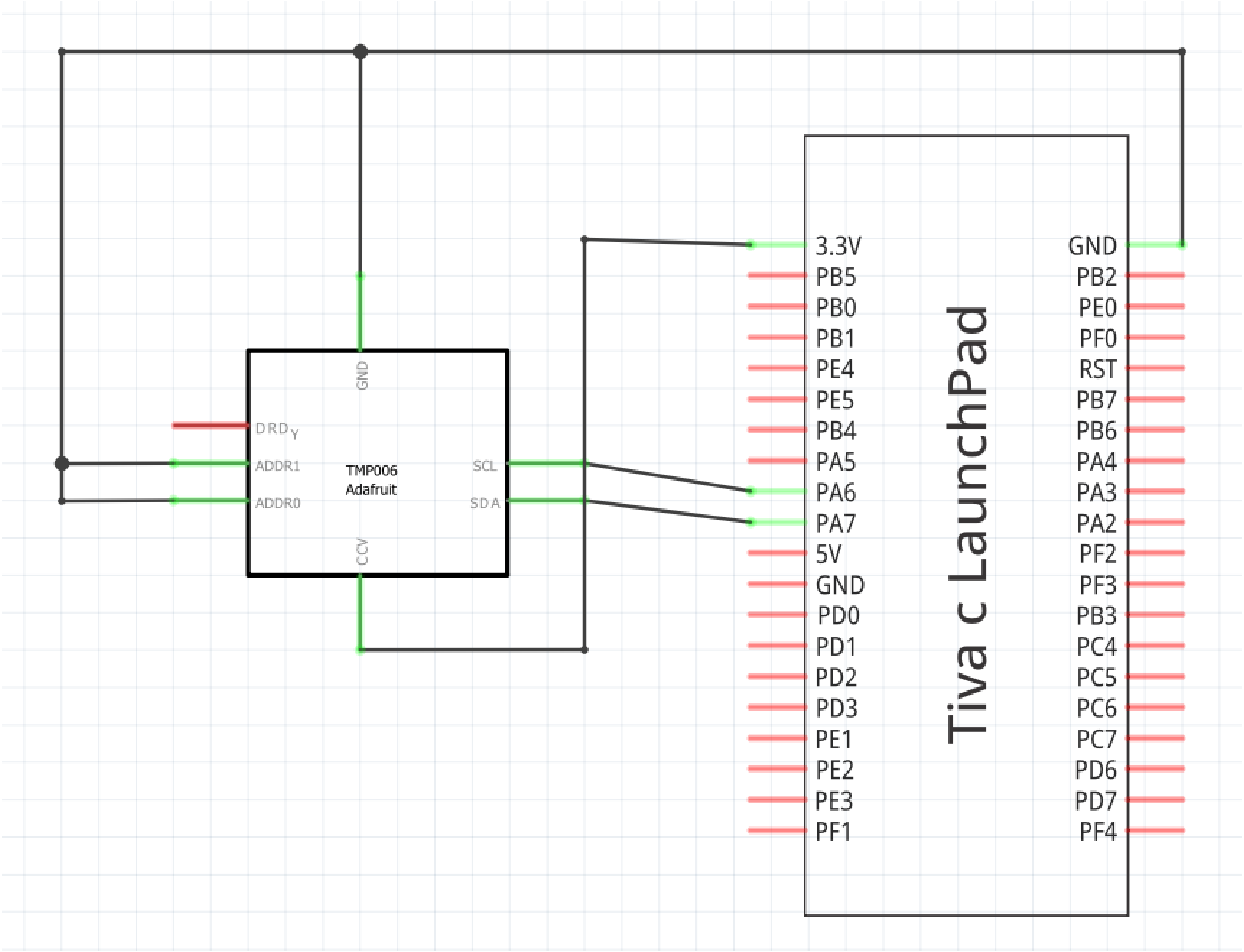
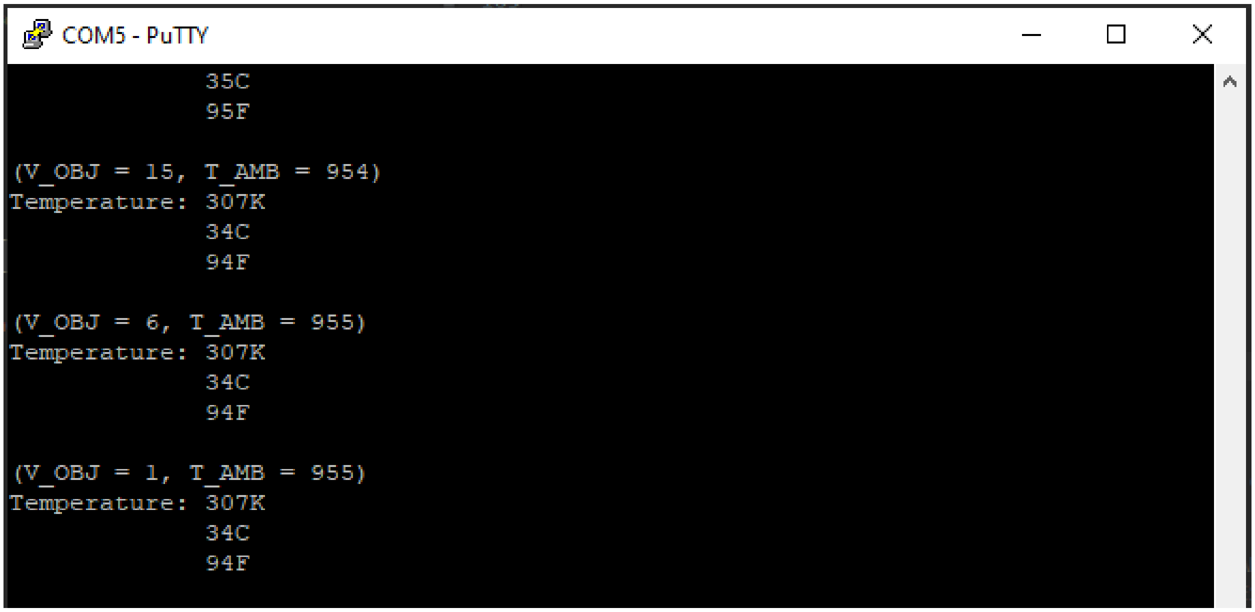


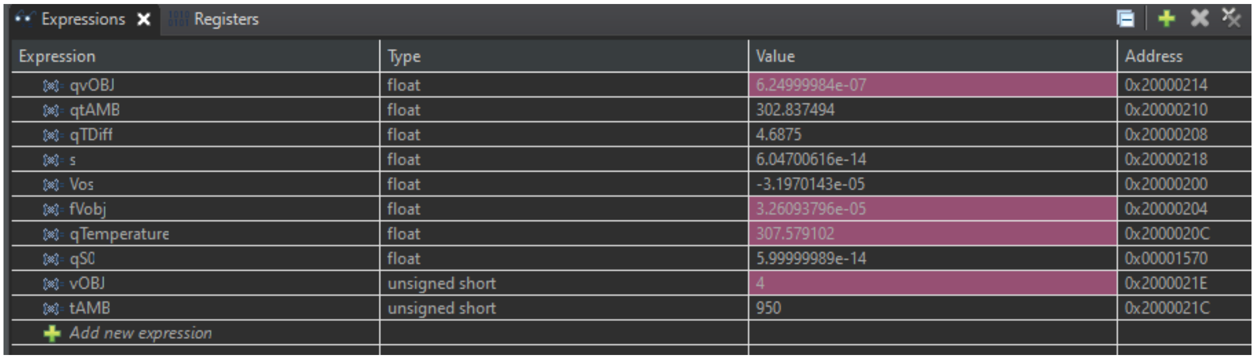
Figure 1: Block Diagram

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



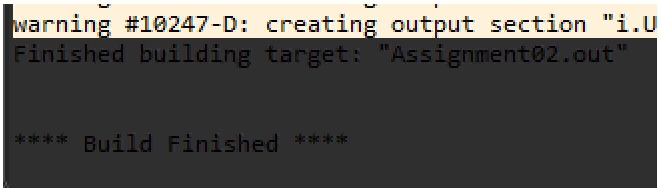
```
              35C
              95F

(V_OBJ = 15, T_AMB = 954)
Temperature: 307K
              34C
              94F

(V_OBJ = 6, T_AMB = 955)
Temperature: 307K
              34C
              94F

(V_OBJ = 1, T_AMB = 955)
Temperature: 307K
              34C
              94F
```

Figure 5: Measurement of the temperature of my hand



Expression	Type	Value	Address
qvOBJ	float	6.24999984e-07	0x20000214
qtAMB	float	302.837494	0x20000210
qTDiff	float	4.6875	0x20000208
s	float	6.04700616e-14	0x20000218
Vos	float	-3.1970143e-05	0x20000200
fVobj	float	3.26093796e-05	0x20000204
qTemperature	float	307.579102	0x2000020C
qS0	float	5.99999989e-14	0x00001570
vOBJ	unsigned short	4	0x2000021E
tAMB	unsigned short	950	0x2000021C
Add new expression			

Figure 4: Real time look at values in variable



```
warning #10247-D: creating output section "i.U
Finished building target: "Assignment02.out"


**** Build Finished ****
```

Figure 3: Successful build of program

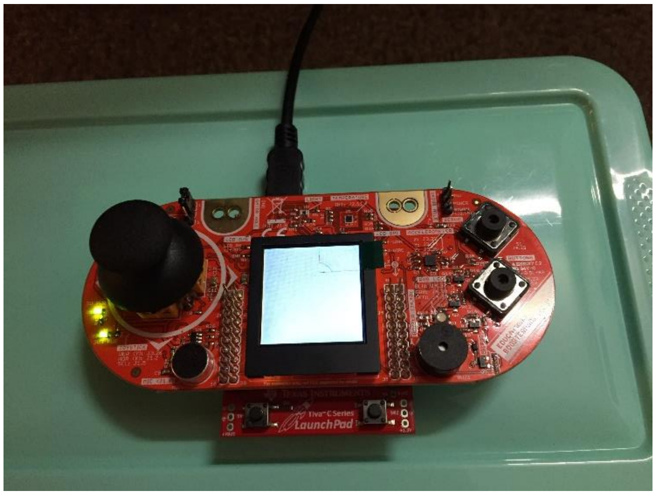


Figure 2: Hardware setup

4. Declaration

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

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