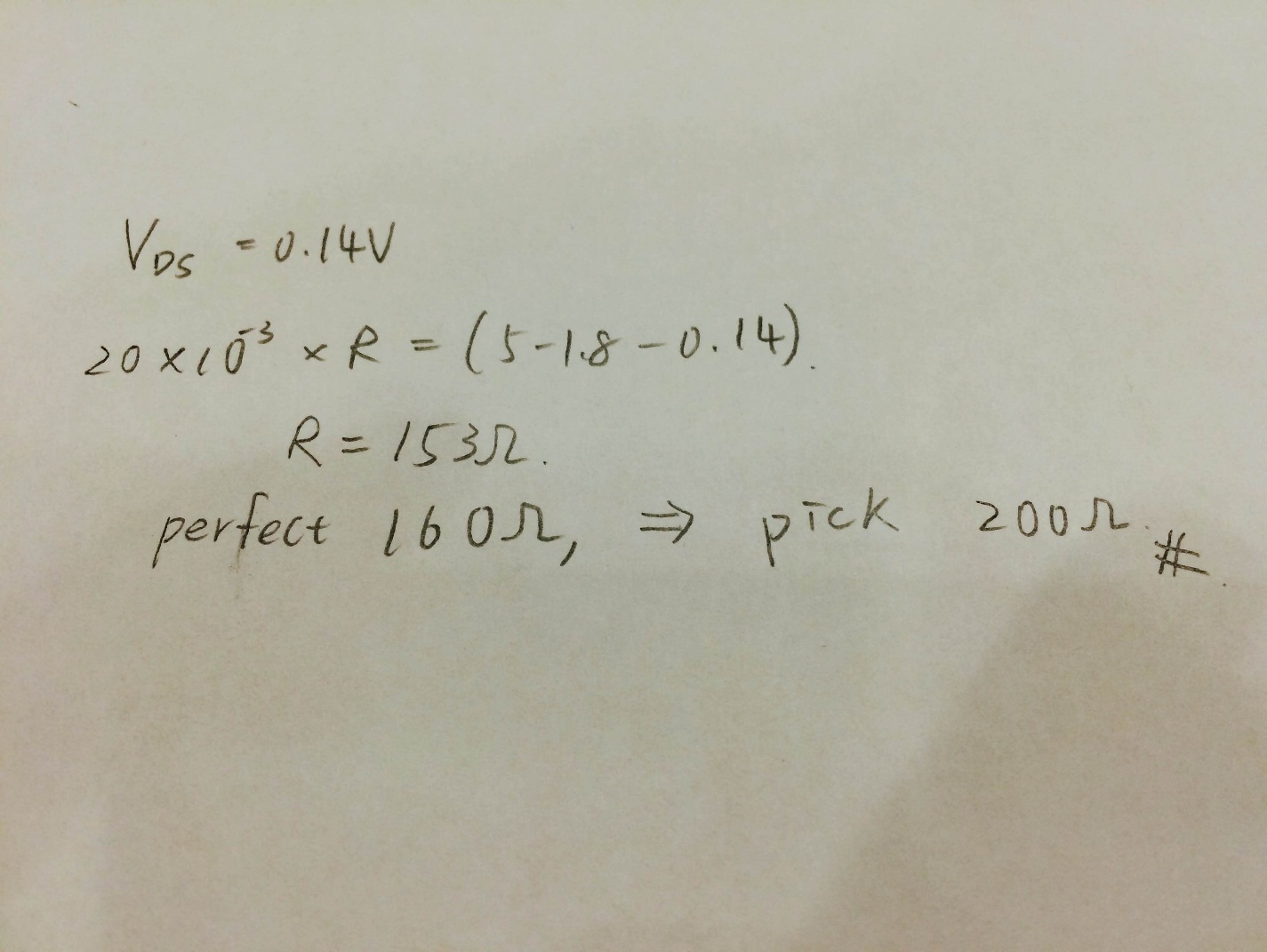
ME218A Lab Assignment #3 pcwu1023 Pei-Chen, Wu

Pre-lab section:

0.1)





0.2)

HWREG(SYSCTL\_RCGCGPIO) |= SYSCTL\_RCGCGPIO\_R5;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_DEN) |= GPIO\_PIN\_4;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_DIR) |= GPIO\_PIN\_4;

0.3)

void problem\_two\_four(){

HWREG(SYSCTL\_RCGCGPIO) |= SYSCTL\_RCGCGPIO\_R5;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_DEN) |= GPIO\_PIN\_4;

HWREG(GPIO\_PORTF\_BASE+GPIO\_O\_DIR) |= GPIO\_PIN\_4;

while(true){

HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) |= BIT4HI; HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) &= BIT4LO;

}

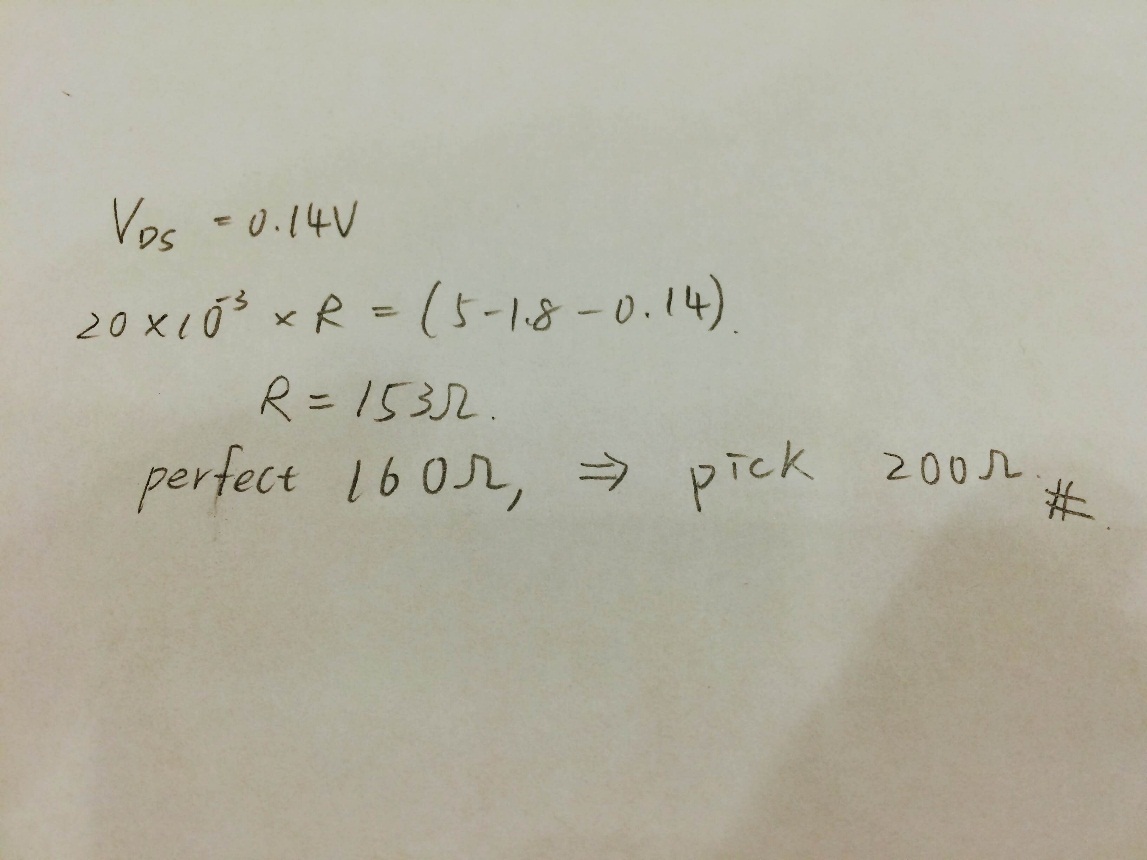
}

0.4)

I combine the Pseudo-Code to the name and comments of each function in the actual code in part3.

* Part 1: Setting up some debugging hardware





* Part 2: Your First & Second Embedded C program

2.2)

void problem\_two\_two(){

HWREG(SYSCTL\_RCGCGPIO) |= SYSCTL\_RCGCGPIO\_R5;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_DEN) |= GPIO\_PIN\_4;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_DIR) |= GPIO\_PIN\_4;

while(true){

HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) = BIT4HI;

HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) = BIT4LO;

}

}

2.3)

* V peak to peak = 3.594V;
* F = 3.066MHz;
* Period of HI time = 148ns
* Period of LOW time = 184ns

I revered the BIT4HI and BIT4LO, and the period of HI and LOW time reverses. The reason might be the while loop. The code needs to go back to the beginning of while loop.

2.4)

void problem\_two\_four(){

HWREG(SYSCTL\_RCGCGPIO) |= SYSCTL\_RCGCGPIO\_R5;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_DEN) |= GPIO\_PIN\_4;

HWREG(GPIO\_PORTF\_BASE+GPIO\_O\_DIR) |= GPIO\_PIN\_4;

while(true){

HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) |= BIT4HI; HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) &= BIT4LO;

}

}

2.5)

* Frequency: 2.186MHz
* the period of the HI time: 162ns
* the period of the LO time: 292ns

The difference between the period of HI and LOW time comes from the while loop. Otherwise, we can observe that the frequency decreases. The reason of this result might because of the HWREG has to be called twice when using bitwise operator.

2.6)

//Using while loop to delay the program. The function represents delay for 10 uS.

void delay\_uS(unsigned int num\_10\_uS){

int i=0;

while(i <= 1000\*num\_10\_uS) i++;

}

void problem\_two\_five(){

HWREG(SYSCTL\_RCGCGPIO) |= SYSCTL\_RCGCGPIO\_R5;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_DEN) |= GPIO\_PIN\_4;

HWREG(GPIO\_PORTF\_BASE+GPIO\_O\_DIR) |= GPIO\_PIN\_4;

while(true){

HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) |= BIT4HI;

delay\_uS(1); //delay for 10uS

HWREG(GPIO\_PORTF\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) &= BIT4LO;

delay\_uS(1); //delay for 10uS

}

}

* Part 3: Controlling An LCD Display

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

LCD.c file

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

#include <stdint.h>

#include <stdbool.h>

#include <stdio.h>

#include <string.h>

#include "ES\_Port.h"

#include "termio.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "inc/hw\_gpio.h"

#include "inc/hw\_sysctl.h"

#include "driverlib/gpio.h"

#define clrScrn() printf("\x1b[2J")

#define ALL\_BITS (0xff<<2)

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

Delay for few 10uS.

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

static void delay\_uS(unsigned int num\_10\_uS){

int i=0;

while(i <= 1000\*num\_10\_uS) i++;

}

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

Initializing micro-controller

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

static void Register\_Data\_Pins(){

HWREG(SYSCTL\_RCGCGPIO) |= SYSCTL\_RCGCGPIO\_R2; //port C

HWREG(GPIO\_PORTC\_BASE + GPIO\_O\_DEN) |= 0xf0; //pin 4,5,6,7

HWREG(GPIO\_PORTC\_BASE + GPIO\_O\_DIR) |= 0xf0; //pin 4,5,6,7

HWREG(SYSCTL\_RCGCGPIO) |= SYSCTL\_RCGCGPIO\_R1; //port B

HWREG(GPIO\_PORTB\_BASE + GPIO\_O\_DEN) |= 0xc0; //pin 6,7

HWREG(GPIO\_PORTB\_BASE + GPIO\_O\_DIR) |= 0xc0; //pin 6,7

}

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

Setting RS pin of LCD

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

static void Set\_RS\_bit(int state){

switch(state){

case 1: HWREG(GPIO\_PORTB\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) |= BIT6HI; break; //Setting RS pin to HI.

case 0: HWREG(GPIO\_PORTB\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) &= BIT6LO; break; //Setting RS pin to LOW.

}

}

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

Setting Enable pin of LCD

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

static void Set\_E\_bit(int state){

switch(state){

case 1: HWREG(GPIO\_PORTB\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) |= BIT7HI; break; //Setting E pin to HI.

case 0: HWREG(GPIO\_PORTB\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) &= BIT7LO; break; //Setting E pin to HI.

}

}

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

Using timer to write wait function.

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

static void Wait(uint16\_t Time){ //Passing required time period.

uint16\_t T1 = ES\_Timer\_GetTime(); //Get initial time.

while(true){

uint16\_t T2 = ES\_Timer\_GetTime(); //Get current time.

if((T2-T1) > Time) break; //if current time > required time, then break the while loop.

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Pulse function. Using delay function to keep the output pin HI for about 100uS.

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

static void Enable\_pulse(){

HWREG(GPIO\_PORTB\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) |= BIT7HI;

delay\_uS(10); //delay about 100uS

HWREG(GPIO\_PORTB\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) &= BIT7LO;

}

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

Input different BITS pattern to initialize or put char to LCD.

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

static void Set\_Bits(uint32\_t hex\_int){

HWREG(GPIO\_PORTC\_BASE+(GPIO\_O\_DATA+ALL\_BITS)) = hex\_int;

}

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

Move cursor from leftmost to rightmost.

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

static void Move\_Cursor(){

Set\_Bits(0x10); //Setting move cursor BITS pattern

Enable\_pulse();

Set\_Bits(0xc0);

Enable\_pulse();

Wait(5); //wait for 5mS.

}

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

The upper function can only move cursor to next one position, so using for loop to move cursor to rightmost.

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

void Cursor\_To\_RightMost(){

for(int i=0; i<=23; i++) Move\_Cursor();

}

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

Initialization of LCD. Since the lower four bits is connected to ground, I manually set every group of bits to upper four bits.

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

void LCDInit(){

\_HW\_Timer\_Init(ES\_Timer\_RATE\_1mS);

TERMIO\_Init();

clrScrn();

Register\_Data\_Pins(); //Register pins.

Set\_RS\_bit(0); //make sure RS and Enable pin are LOW

Set\_E\_bit(0);

Wait(150);

//Following steps are the procedure for initializing LCD

Set\_Bits(0x30);

Enable\_pulse();

Wait(5);

Set\_Bits(0x30);

Enable\_pulse();

Wait(1);

Set\_Bits(0x30);

Enable\_pulse();

Wait(1);

Set\_Bits(0x20); //Change interface from 8bits to 4bits mode

Enable\_pulse();

Wait(1);

//Set interface length

Set\_Bits(0x20);

Enable\_pulse();

Set\_Bits(0x00);

Enable\_pulse();

Wait(1);

Set\_Bits(0x00); //Display on/off

Enable\_pulse();

Set\_Bits(0x80);

Enable\_pulse();

Wait(1);

Set\_Bits(0x00); //Clear Display

Enable\_pulse();

Set\_Bits(0x10);

Enable\_pulse();

Wait(5);

Set\_Bits(0x00); //Set cursor move direction

Enable\_pulse();

Set\_Bits(0x70);

Enable\_pulse();

Wait(1);

Set\_Bits(0x00); //Enable Display/Cursor

Enable\_pulse();

Set\_Bits(0xf0);

Enable\_pulse();

Wait(3);

Cursor\_To\_RightMost(); //Move Cursor/Shift Display

}

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

Put char to LCD.

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

void LCDputchar(char source){

//Write a char to the display at the cursor position

Set\_RS\_bit(1);

char upper\_bits = source & 0xf0;

char lower\_bits = source & 0x0f;

lower\_bits <<= 4; //Move lower bits upper

Set\_Bits((uint32\_t)upper\_bits); //Type cast to uint32\_bits

Enable\_pulse();

Set\_Bits((uint32\_t)lower\_bits);

Enable\_pulse();

Wait(5); //Wait for 5mS for the instruction to complete

Set\_RS\_bit(0);

}

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

LCD.h file

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

#ifndef LCD\_H

#define LCD\_H

void LCDInit();

void LCDputchar(char source);

#endif

* Appendix : Enter a string from terminal to LCD (for fun)

#include <stdint.h>

#include <stdbool.h>

#include <stdio.h>

#include <string.h>

#include "LCD.h" //Include the LCD.h written by myself

#include "ES\_Port.h"

#include "termio.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "inc/hw\_gpio.h"

#include "inc/hw\_sysctl.h"

#include "driverlib/gpio.h"

#define clrScrn() printf("\x1b[2J")

#define ALL\_BITS (0xff<<2)

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

Wait for few mS

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

static void Wait(uint16\_t Time){

uint16\_t T1 = ES\_Timer\_GetTime();

while(true){

uint16\_t T2 = ES\_Timer\_GetTime();

if((T2-T1)>Time) break;

}

}

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

Delay for few 10 uS

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

static void delay\_uS(unsigned int num\_10\_uS){

int i=0;

while(i <= 1000\*num\_10\_uS) i++;

}

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

Get a string from terminal.

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

void get\_input\_sentence(char string[]){

printf("Enter a vacabulary:");

scanf("%s", string);

printf("\n\r\n");

printf("%s", string);

}

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

After the user enter a string from terminal, the following function will put the each char of string to LCD.

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

void interact\_display(char string[]){

while(true){

for(int i=0; i < strlen(string); i++){

LCDputchar(string[i]);

Wait(500);

}

for(int i=0; i<=(24-strlen(string)); i++){

LCDputchar(' ');

Wait(500);

}

}

}

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

Display some words written in the source code

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

void display\_default(){

while(true){

LCDputchar('C');

Wait(800);

LCDputchar('O');

Wait(800);

LCDputchar('N');

Wait(800);

LCDputchar('G');

Wait(800);

LCDputchar('R');

Wait(800);

LCDputchar('A');

Wait(800);

LCDputchar('T');

Wait(800);

LCDputchar('U');

Wait(800);

LCDputchar('L');

Wait(800);

LCDputchar('A');

Wait(800);

LCDputchar('T');

Wait(800);

LCDputchar('I');

Wait(800);

LCDputchar('O');

Wait(800);

LCDputchar('N');

Wait(800);

LCDputchar('!');

Wait(800);

LCDputchar('!');

Wait(800);

LCDputchar(' ');

Wait(800);

LCDputchar(' ');

LCDputchar('Y');

Wait(800);

LCDputchar('A');

Wait(800);

LCDputchar('!');

Wait(800);

LCDputchar(' ');

Wait(800);

}

}

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

The function allows user manually select two modes, the one is enter a string from terminal, the other is display a default string. The robustness test is included in the function, too.

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

int select\_mode(){

int mode = 0;

while(true){

printf("Select mode ([1] manually type a vacabulary. [2] Display):");

printf("\n\r\n");

scanf("%d", &mode);

if(mode==1 || mode==2) return mode;

else printf("Please enter number 1 or 2 \n");

}

}

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

Display of LCD based on the selected mode.

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

void Display(char string[]){

int mode = select\_mode();

switch(mode){

case 1: get\_input\_sentence(string); interact\_display(string); break;

case 2: printf("Display!!"); display\_default(); break;

}

}

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

Main function to run the whole prgram

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

int main()

{ char string[50] = {"\0"};

LCDInit();

Display(string);

return 0;

}