#include <stdio.h>

#include <LPC17xx.h> // Include the LPC17xx header file for LPC1768 microcontroller

#include <string.h> // Include the string header file for string operations

// Defining Constants

#define LED\_Pinsel 0xff // Define LED\_Pinsel as 0xff for pins P0.4 to P0.11 (LEDs)

#define TRIGGER\_PIN (1 << 15) // Define TRIGGER\_PIN as (1 << 15) for pin P0.15 (Trigger Pin)

#define ECHO\_PIN (1 << 16) // Define ECHO\_PIN as (1 << 16) for pin P0.16 (Echo Pin)

// Variable Declarations

char ans[20] = ""; // Declare an array to store the output message

int temp, temp1, temp2 = 0; // Declare integer variables to store temporary values

int flag = 0, flag\_command=0; // Declare flags for control purposes

int i, j, k, l, r, echoTime = 5000; // Declare integer variables for iteration and echo time measurement

float distance = 0; // Declare a floating-point variable to store the calculated distance

//Function Declarations

void lcd\_wr(void); // Declare a function to write data to the LCD

void port\_wr(void); // Declare a function to write data to ports

void delay(int r1); // Declare a function to introduce delay

void timer\_start(void); // Declare a function to start a timer

float timer\_stop(); // Declare a function to stop a timer and return elapsed time

void timer\_init(void); // Declare a function to initialize a timer

void dealy\_in\_US(unsigned int microseconds); // Declare a function to introduce microsecond delay

void dealy\_in\_MS(unsigned int milliseconds); // Declare a function to introduce millisecond delay

//Functions

void dealy\_in\_US(unsigned int microseconds) {

LPC\_TIM0->TCR = 0x02; // Reset Timer

LPC\_TIM0->PR = 0; // Set prescaler to 0

LPC\_TIM0->MR0 = microseconds - 1; // Set match register for microseconds

LPC\_TIM0->MCR = 0x01; // Interrupt on match

LPC\_TIM0->TCR = 0x01; // Enable timer

while ((LPC\_TIM0->IR & 0x01) == 0); // Wait for interrupt flag

LPC\_TIM0->TCR = 0x00; // Stop the timer

LPC\_TIM0->IR = 0x01; // Clear the interrupt flag

}

void dealy\_in\_MS(unsigned int milliseconds) {

dealy\_in\_US(milliseconds \* 1000); // Convert milliseconds to microseconds and call dealy\_in\_US

}

void timer\_init(void) {

LPC\_TIM0->CTCR = 0x0; // Timer mode

LPC\_TIM0->PR = 11999999; // Set prescaler for 12MHz

LPC\_TIM0->TCR = 0x02; // Reset Timer

}

void timer\_start(void) {

LPC\_TIM0->TCR = 0x02; // Reset Timer

LPC\_TIM0->TCR = 0x01; // Enable timer

}

float timer\_stop() {

LPC\_TIM0->TCR = 0x0; // Disable timer

return LPC\_TIM0->TC; // Return elapsed time

}

void delay(int r1) {

for (r = 0; r < r1; r++); // Loop to introduce delay

}

void port\_wr() {

int j;

LPC\_GPIO0->FIOPIN = temp2 << 23; // Set port pins based on temp2 value

if (flag\_command == 0) {

LPC\_GPIO0->FIOCLR = 1 << 27; // Clear port pin 27

} else {

LPC\_GPIO0->FIOSET = 1 << 27; // Set port pin 27

}

LPC\_GPIO0->FIOSET = 1 << 28; // Set port pin 28

for (j = 0; j < 50; j++); // Delay

LPC\_GPIO0->FIOCLR = 1 << 28; // Clear port pin 28

for (j = 0; j < 10000; j++); // Delay

}

void lcd\_wr() {

temp2 = (temp1 >> 4) & 0xF; // Extract higher 4 bits of temp1

port\_wr(); // Call port\_wr

temp2 = temp1 & 0xF; // Extract lower 4 bits of temp1

port\_wr(); // Call port\_wr

}

// Main Program

int main() {

int ledflag = 0;

int command\_init[] = {3, 3, 3, 2, 2, 0x01, 0x06, 0x0C, 0x80}; // Initialize LCD

SystemInit(); // Initialize system

SystemCoreClockUpdate(); // Update system clock

timer\_init(); // Initialize timer

LPC\_PINCON->PINSEL0 &= 0xfffff00f; // Interface LEDs P0.4-P0.11

LPC\_PINCON->PINSEL0 &= 0x3fffffff; // Interface TRIG P0.15

LPC\_PINCON->PINSEL1 &= 0xfffffff0; // Interface ECHO P0.16

LPC\_GPIO0->FIODIR |= TRIGGER\_PIN | 1 << 17; // Direction for TRIGGER pin and Buzzer

LPC\_GPIO1->FIODIR |= 0 << 16; // Direction for ECHO PIN

LPC\_GPIO0->FIODIR |= LED\_Pinsel << 4; // Direction for LED

LPC\_PINCON->PINSEL1 |= 0; // Interface LCD

LPC\_GPIO0->FIODIR |= 0XF << 23 | 1 << 27 | 1 << 28; // Direction For LCDs

flag\_command = 0; // Initialize flag\_command

for (i = 0; i < 9; i++) { // Loop for LCD initialization

temp1 = command\_init[i]; // Set temp1 to current command

lcd\_wr(); // Call lcd\_wr

for (j = 0; j < 30000; j++); // Delay

}

i = 0; // Initialize i

flag = 1; // Initialize flag

LPC\_GPIO0->FIOCLR |= TRIGGER\_PIN; // Clear TRIGGER\_PIN

while (1) {

LPC\_GPIO0->FIOSET = 0x00000800; // Output 10us HIGH on the TRIGGER pin

LPC\_GPIO0->FIOMASK = 0xFFFF7FFF; // Set FIOMASK to mask unused pins

LPC\_GPIO0->FIOPIN |= TRIGGER\_PIN; // Set TRIGGER\_PIN

dealy\_in\_US(10); // Introduce 10us delay

LPC\_GPIO0->FIOCLR |= TRIGGER\_PIN; // Clear TRIGGER\_PIN

LPC\_GPIO0->FIOMASK = 0x0; // Reset FIOMASK

while (!(LPC\_GPIO0->FIOPIN & ECHO\_PIN)) {

// Wait till ECHO PIN becomes high

}

timer\_start(); // Start timer

while (LPC\_GPIO0->FIOPIN & ECHO\_PIN); // Wait till ECHO PIN becomes low

echoTime = timer\_stop(); // Stop timer and store elapsed time

distance = (0.00343 \* echoTime) / 2; // Calculate distance in cm

sprintf(ans, " Distance: %.3f", distance); // Format distance into string

flag\_command = 1; // Set flag\_command to 1

i = 0; // Reset i

flag\_command = 0; // Reset flag\_command

temp1 = 0x01; // Set temp1 to 0x01

lcd\_wr(); // Call lcd\_wr

flag\_command = 1; // Set flag\_command to 1

while (ans[i] != '\0') { // Loop until end of string

temp1 = ans[i]; // Set temp1 to current character

lcd\_wr(); // Call lcd\_wr

for (j = 0; j < 30000; j++); // Delay

i++; // Increment i

}

if (distance < 20) { // If distance is less than 20 cm

LPC\_GPIO0->FIOSET = LED\_Pinsel << 4; // Set LED pins

LPC\_GPIO0->FIOSET = 1 << 17; // Set Buzzer pin

} else { // If distance is greater than or equal to 20 cm

LPC\_GPIO0->FIOCLR = LED\_Pinsel << 4; // Clear LED pins

LPC\_GPIO0->FIOCLR = 1 << 17; // Clear Buzzer pin

}

delay(88000); // Introduce delay

}

}