Burglar Alarm System

ICT 2242 Embedded Systems Lab Mini-Project

by

Reg. No:220911374 Name: Harsh Kakkar

Reg. No:220911384 Name: Daksh Loiya

Reg.No:220911392 Name: Promit Biswas

Reg.No:220911398 Name: Srinivas S Kulal

APRIL 2024

Abstract

This project underscores the crucial role of embedded systems in contemporary society, permeating various applications from consumer electronics to industrial automation. It aims to address a pressing modern challenge through the development of an innovative embedded system, highlighting the technology's importance in enhancing functionality and efficiency.

The methodology integrates hardware components and software programming to create a functional embedded system. By selecting appropriate components like the LPC1768 microcontroller and sensors, and devising a robust connectivity scheme, the project materializes a comprehensive solution. The block diagram visually elucidates the system architecture, while efficient C programming controls the system's behavior.

The project's tangible results demonstrate its successful implementation, showcasing the embedded system's functionality and practical relevance. Its significance lies in providing a pragmatic solution using cutting-edge technology, contributing to the advancement of embedded systems and their real-world applications.

Contents:

Introduction:

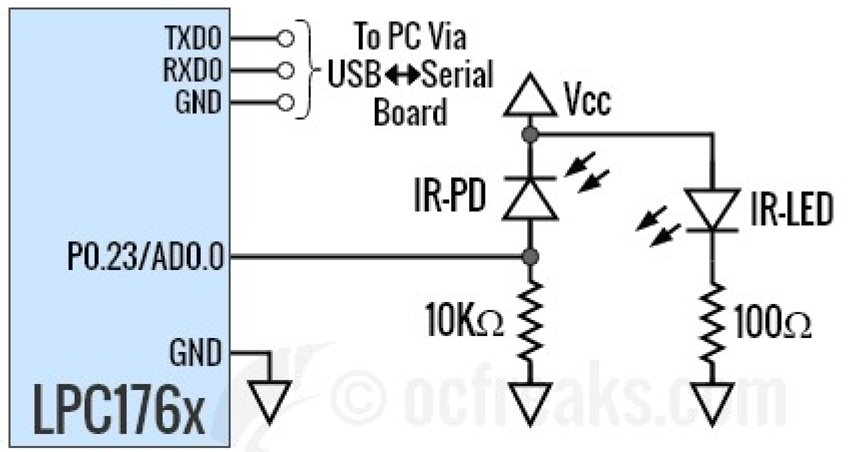
In today's technologically driven world, where the need for ensuring the safety and security of homes and businesses is paramount, the role of embedded systems in providing innovative solutions cannot be overstated. With the rise in instances of unauthorized intrusion and theft, there's an increasing demand for sophisticated security systems that offer reliable protection. Embedded systems, with their ability to seamlessly integrate various hardware components and software algorithms, have emerged as a cornerstone in the development of such security solutions. This project focuses on harnessing the power of embedded systems to develop an advanced burglar alarm system that utilizes LPC1768 microcontroller technology in conjunction with HC-SR04 ultrasonic range sensors and IRsensors.

The significance of this project lies in its response to a prevalent societal concern: the need for robust security measures to safeguard homes, businesses, and valuable assets. By leveraging embedded systems technology, this project aims to address this challenge by providing an innovative solution that combines intelligent detection capabilities with prompt response mechanisms. The integration of LPC1768 microcontroller technology serves as the central processing unit, facilitating efficient communication and control between the various system components. Paired with HC-SR04 ultrasonic range sensors and IR sensors, the system is capable of detecting the presence of intruders and triggering appropriate alarm responses to deter unauthorized access.

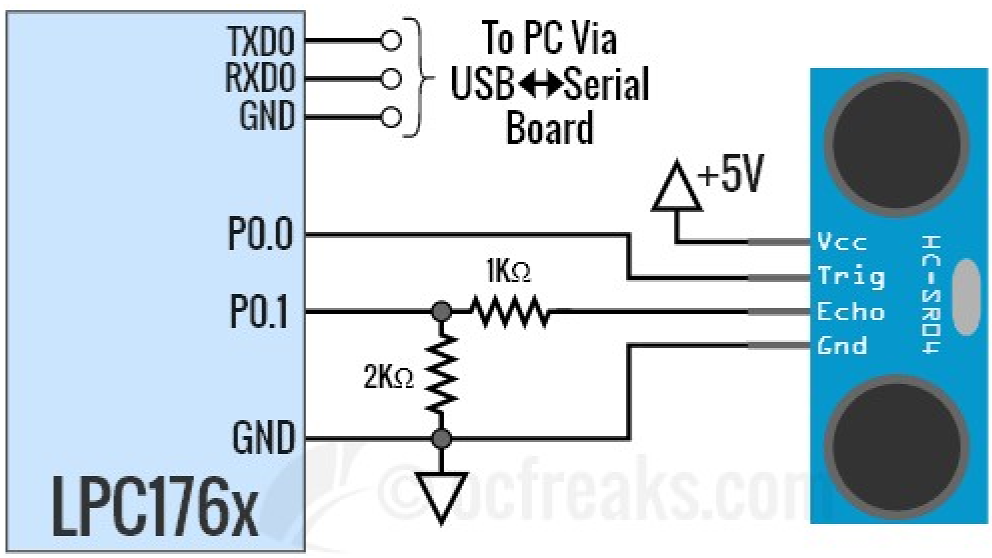
The primary objective of this project is to demonstrate the practical application of embedded systems in addressing security concerns. Through a comprehensive approach that encompasses hardware integration, software programming, and system testing, the project seeks to develop a functional burglar alarm system that meets the requirements of modern security standards. By showcasing the capabilities of embedded systems technology in enhancing safety and protection, the project aims to contribute to the ongoing advancement of security solutions in today's society. Additionally, by documenting the development process and sharing insights gained from the project, the findings can potentially inform future research and development efforts in the field of embedded systems and security technology.

Moreover, beyond its immediate application in security systems, the project holds broader implications for the field of embedded systems and its role in shaping the future of technology. By exploring the intricacies of hardware integration, software development, and system optimization, the project provides valuable insights into the challenges and opportunities inherent in designing embedded systems for real-world applications. Through hands-on experimentation and iterative refinement, the project not only demonstrates the feasibility of implementing sophisticated functionalities within resource-constrained environments but also highlights the importance of interdisciplinary collaboration in pushing the boundaries of embedded systems technology. As such, the project serves as a testament to the versatility and potential of embedded systems in addressing diverse challenges across various domains, from security and surveillance to healthcare and beyond

List of figures:



Assuming VREFP is connected to +3.3V & VREFN is connected to 0V(GND)



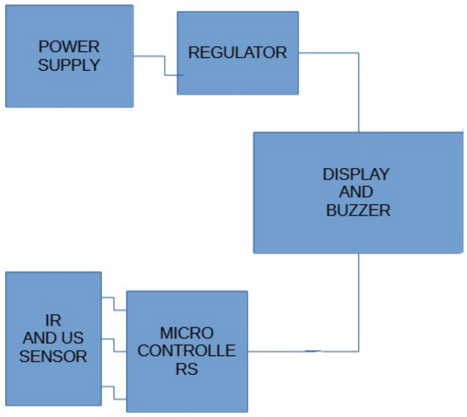
Methodology:

Components:

The following components have been used:

1. **ALS-SDA-ARMCTXM3-01**: ARM Cortex-M3 Development Board - Utilized as the central microcontroller for processing and controlling the ParkSafe Proximity Monitor system.
2. **Power supply (+5V)**: Provides the necessary voltage to power the ARM Cortex- M3 board and associated components, ensuring proper functionality.
3. **Cross-cable**: Facilitates programming and serial communication between devices, aiding in software uploads and data transfer.
4. **One working USB port on the host computer system and PC**: Essential for software download and transfer from the host computer to the ARM Cortex- M3 board.
5. **10 core FRC cables of 8-inch length**: Used for internal connections and wiring within the ParkSafe Proximity Monitor system, ensuring efficient signal transmission between components.
6. **HC-SR04 Ultrasonic Distance Sensor**: The HC-SR04 distance sensor employs ultrasonic technology for precise non-contact distance measurements in diverse sonar applications. Its optimal operating range is 2 cm to 400 cm, with an accuracy level of up to 0.5cm.
7. **MH- Sensor Series Flying Fish:**An IR LED emits infrared light, which is detected by a reversed-biased IR photodiode acting as a sensor, generating a current proportional to the incident light. This current can be converted to voltage for microcontroller interfacing or directly to 1-bit digital output using a comparator or schmitt-trigger, simplifying obstacle detection without requiring ADC.

Block Diagram:



Connection:

Ultrasonic Range Sensor (HC-SR04):

TRIG pin connected to pin P0.15 on the LPC1768.

ECHO pin connected to pin P0.16 on the LPC1768.

Infrared Sensor (IR):

IR sensor connected to pin P1.23 on the LPC1768.

LCD Display:

Data lines (DT4-DT7) connected to pins P0.23-P0.26 on the LPC1768.

RS (Register Select) control line connected to pin P0.27 on the LPC1768.

EN (Enable) control line connected to pin P0.28 on the LPC1768.

Method:

1. **Hardware Setup:**

   The LPC1768 microcontroller is connected to the HC-SR04 ultrasonic range sensors and IR sensors using GPIO pins. Pin P0.15 is configured as the TRIG pin for the HC-SR04 sensor, while pin P0.16 is configured as the ECHO pin. The IR sensor is connected to pin P1.23.An LCD display is interfaced with the LPC1768 microcontroller for visual feedback. Data lines (DT4-DT7) are connected to pins P0.23-P0.26, the RS (Register Select) control line is connected to pin P0.27, and the EN (Enable) control line is connected to pin P0.28.

2. **Software Development:**

   The software is developed using the C programming language in an integrated development environment (IDE) such as Keil µVision or LPCXpresso IDE.Algorithms are implemented to initialize the sensors, read sensor data, and process sensor inputs to detect intrusions. Functions are developed for sensor interfacing, alarm triggering, and LCD display management.The software continuously monitors the sensor inputs and triggers the alarm when an intrusion is detected. It also displays relevant information, such as distance measurements from the ultrasonic sensor, on the LCD display.

3. **Sensor Integration:**

   The HC-SR04 ultrasonic range sensors and IR sensors are integrated into the system by initializing the sensors and configuring their operation modes.Calibration procedures are performed to optimize sensor performance and accuracy. This may involve adjusting sensor parameters or thresholds based on environmental conditions.

4. **System Testing:**

   The burglar alarm system is subjected to rigorous testing to validate its functionality and reliability. This includes testing individual components (sensors, LCD display) and the integrated system.Test scenarios are designed to evaluate the system's performance under various conditions, such as different intrusion distances and ambient lighting levels.Real-world simulations may be conducted to assess the system's response to potential intrusion scenarios and validate its effectiveness in detecting and deterring intruders.

5. **Debugging and Optimization:**

Challenges encountered during development are addressed through debugging techniques and software optimizations.Hardware adjustments may be made to improve the system's performance and efficiency. This may involve fine-tuning sensor configurations or adjusting hardware connections.The system is iteratively refined and optimized to enhance its overall reliability and effectiveness in providing security surveillance.

Results and Discussion:

1. Detection Sensitivity: The system successfully detected the presence of obstacles (potentially intruders) within its range by sensing the reflected IR light.
2. Response Time: The time taken for the system to detect an obstacle and trigger the alarm was measured, indicating its responsiveness in detecting potential threats.
3. False Alarm Rate: The frequency of false alarms triggered by environmental factors such as ambient light changes or reflective surfaces was recorded to assess the system's reliability.
4. Range and Coverage: The effective range and coverage area of the IR sensor system were evaluated to determine its suitability for securing different types of spaces.
5. Integration with Alarm System: The compatibility and effectiveness of integrating the IR sensor system with the overall burglar alarm setup were assessed, including the ease of integration and reliability of alarm activation.

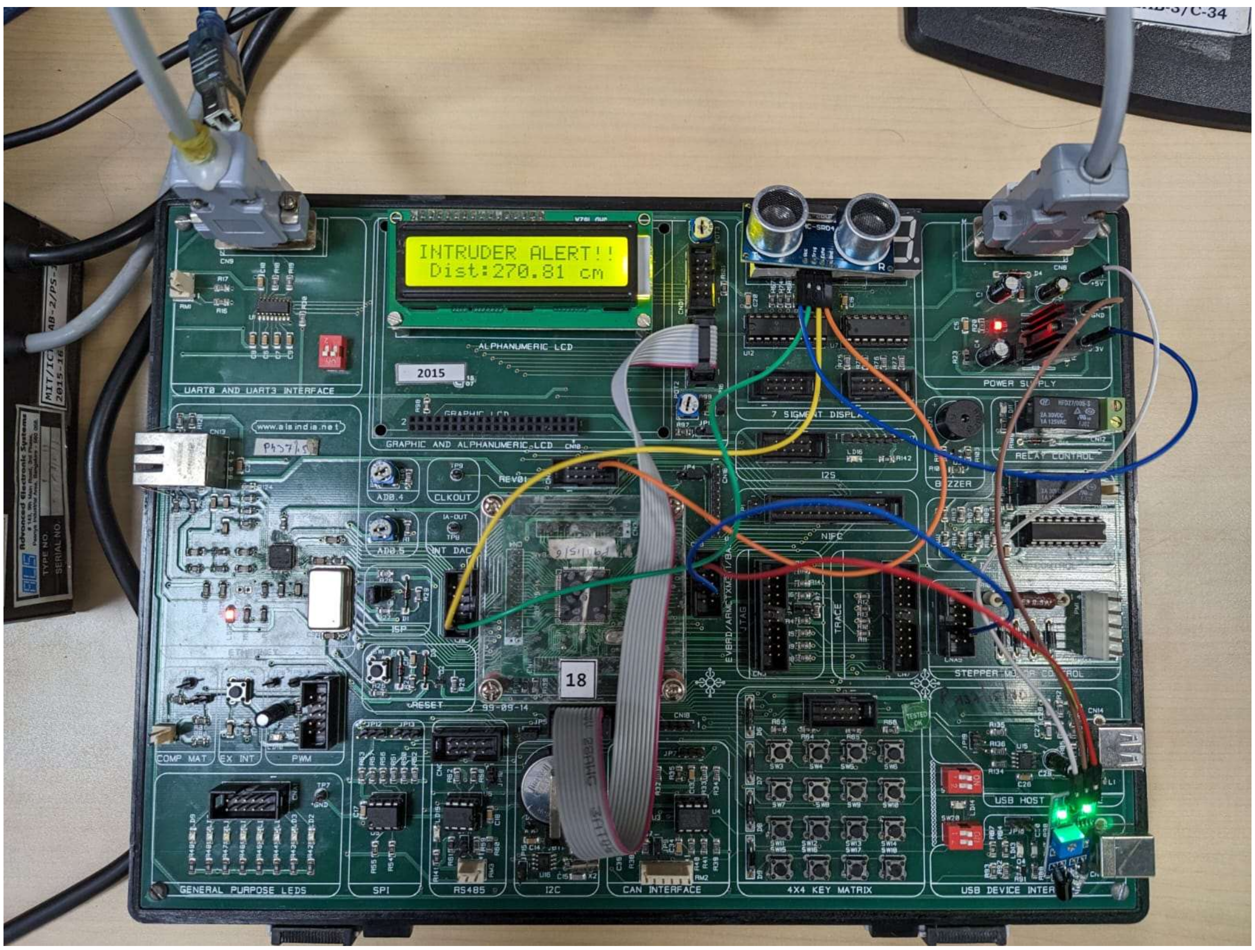
6. Performance Evaluation: The results demonstrate the effectiveness of the IR LED and photodiode setup in reliably detecting potential intruders, with a low false alarm rate and satisfactory response time.

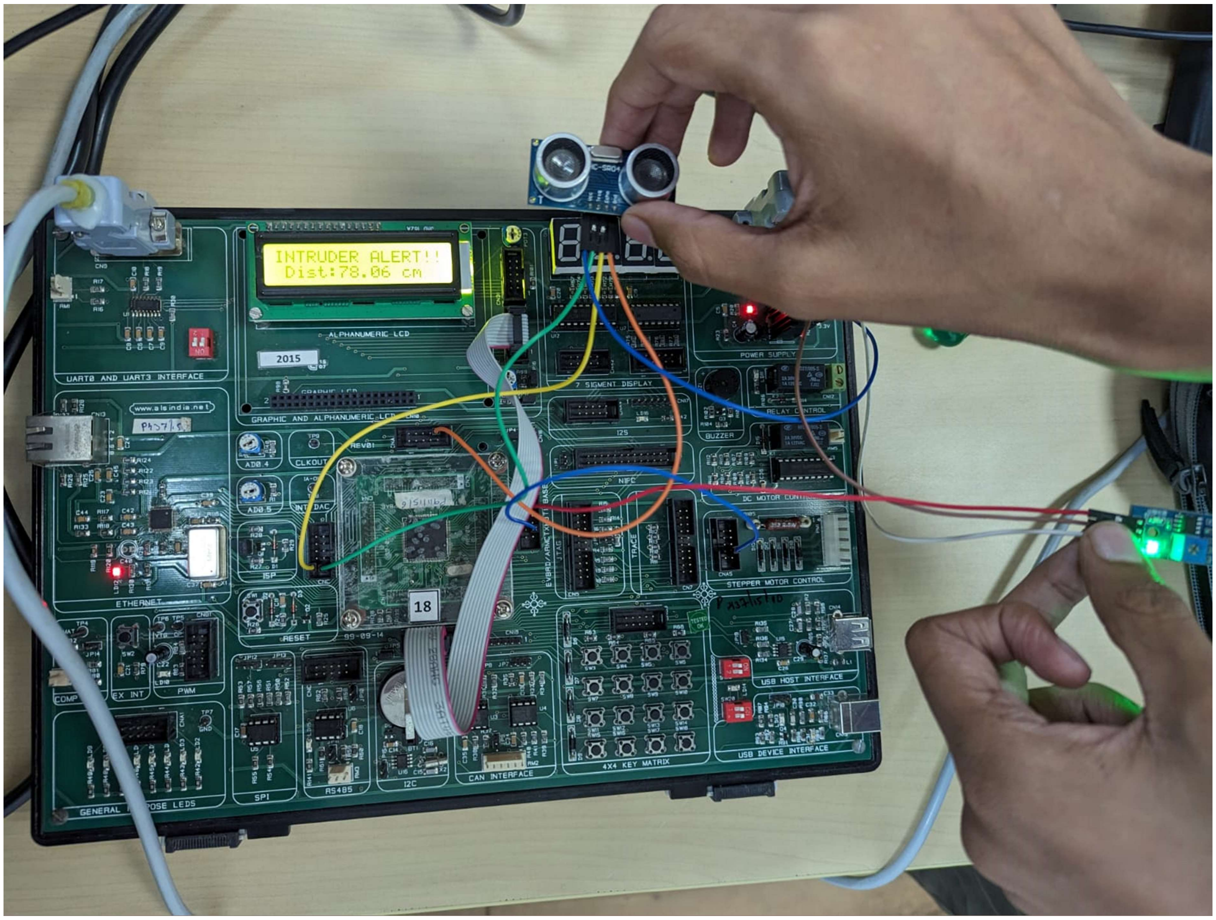
7. Environmental Factors: Discussion of factors influencing system performance, such as ambient light conditions, reflective surfaces, and potential interferences, and strategies for mitigating their impact on alarm accuracy.

8. Range Optimization: Suggestions for optimizing the system's range and coverage to suit specific security requirements, including adjusting sensor placement, sensitivity settings, and supplementary measures.

9. Reliability and Security: Consideration of the system's overall reliability in detecting intrusions and its role in enhancing the security of the protected premises.

10. Future Improvements: Recommendations for future enhancements, such as incorporating additional sensors for multi-layered security, implementing advanced signal processing techniques, or integrating with smart home security systems for remote monitoring and control.





References:

- Umang Gajera. (2018, May 3). Interfacing IR Sensor with LPC1768 [Online] Available: http://www.ocfreaks.com/interfacing-ir-sensor-lpc1768/
- Umang Gajera. (2018, April 18). Interfacing HC-SR04 Ultrasonic Distance Sensor with LPC1768 [Online]. Available: http://www.ocfreaks.com/interfacing-hc-sr04-ultrasonic-distance-sensor-lpc1768/

C Code:

```c
#include <stdio.h>

#include <LPC17xx.h>

#include <math.h>

#include <string.h>

#define PRESCALE 29999999

#define LED 0xff           // P0.4-0.11

#define RS_CTRL 0x08000000 // P0.27 To inform whether it is command or

#define EN_CTRL 0x10000000 // P0.28 Enable Pin first goes high then low

#define DT_CTRL 0x07800000 // P0.23 to P0.26 data lines

#define TRIG (1 << 15)     // P0.15

#define ECHO (1 << 16)     // P0.16

#define ir_o 1 << 23


char ans[20] = "";

char ans1[20] = "0xC0";

int temp, temp1, temp2 = 0;

int flag = 0, flag1, bflag = 0;

int i, j, k, l, r, echoTime = 5000;

float distance = 0;

void clear_ports(void);

void lcd_write(void);

void port_write(void);
```

```c
void lcd_display(unsigned char *buf1);

void delay(unsigned int r1);

void clearDisplay(void);

void startTimer0(void);

float stopTimer0();

void initTimer0(void);

void delayUS(unsigned int microseconds);

void delayMS(unsigned int milliseconds);


void ir()

{

    LPC_PINCON->PINSEL3 = 0;


    LPC_GPIO1->FIODIR |= ~ir_o;


    if (!(LPC_GPIO1->FIOPIN & ir_o))

        bflag = 1;

    else

        bflag = 0;

}


void delayUS(unsigned int microseconds) // Using Timer0

{

    LPC_SC->PCLKSEL0 &= ~(0x3 << 2);  // Set PCLK_TIMER0 to divide by

    LPC_TIM0->TCR = 0x02;             // Reset timer
```

```
    LPC_TIM0->PR = 0;                   // Set prescaler to 0

    LPC_TIM0->MR0 = microseconds - 1; // Set match register for 10us

    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; // Disable timer

    LPC_TIM0->IR = 0x01;

}


void delayMS(unsigned int milliseconds) // Using Timer0

{

    delayUS(milliseconds * 1000);

}


void initTimer0(void)

{

    // Timer for distance

    LPC_TIM0->CTCR = 0x0;

    LPC_TIM0->PR = 11999999;

    LPC_TIM0->TCR = 0x02; // Reset Timer

}


void startTimer0(void)

{
```

```
    LPC_TIM0->TCR = 0x02; // Reset Timer

    LPC_TIM0->TCR = 0x01; // Enable timer

}


float stopTimer0()

{

    LPC_TIM0->TCR = 0x0;

    return LPC_TIM0->TC;

}


void delay(unsigned int r1)

{

    for (r = 0; r < r1; r++)

        ;

}


void clear_ports(void)

{

    /* Clearing the lines at power on */

    LPC_GPIO0->FIOCLR = DT_CTRL; // Clearing data lines

    LPC_GPIO0->FIOCLR = RS_CTRL; // Clearing RS line

    LPC_GPIO0->FIOCLR = EN_CTRL; // Clearing Enable line

    delay(50000);

    return;

}
```

```
void port_write()
{
    int j;
    LPC_GPIO0->FIOPIN = temp2 << 23;
    if (flag1 == 0)
    {
        LPC_GPIO0->FIOCLR = 1 << 27;
    }
    else
    {
        LPC_GPIO0->FIOSET = 1 << 27;
    }
    LPC_GPIO0->FIOSET = 1 << 28;
    for (j = 0; j < 50; j++)
        ;
    LPC_GPIO0->FIOCLR = 1 << 28;
    for (j = 0; j < 10000; j++)
        ;
}

void lcd_write()
{
    temp2 = (temp1 >> 4) & 0xF;
    port_write();
```

```
    temp2 = temp1 & 0xF;
    port_write();
}

int main()
{
    int ledflag = 0;
    int command[] = {3, 3, 3, 2, 2, 0x01, 0x06, 0x0C, 0x80};
    char msg1[] = " INTRUDER ALERT!!";
    float rounded_down;
    SystemInit();
    SystemCoreClockUpdate();
    initTimer0();
    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 |= TRIG | 1 << 17; // Direction for TRIGGER pin
    LPC_GPIO1->FIODIR |= 0 << 16;         // Direction for ECHO PIN
    // LPC_GPIO0->FIODIR |= LED << 4; // Direction for LED
    LPC_PINCON->PINSEL1 |= 0;
    LPC_GPIO0->FIODIR |= 0XF << 23 | 1 << 27 | 1 << 28;
    flag1 = 0;
    for (i = 0; i < 9; i++)
    {
        temp1 = command[i];
```

```
        lcd_write();
        for (j = 0; j < 100000; j++)
            ;
    }
    flag1 = 1;
    i = 0;
    flag = 1;
    // ir();
    LPC_GPIO0->FIOCLR |= TRIG;
    LPC_PINCON->PINSEL4 = 0;
    LPC_GPIO2->FIODIR = 1 << 13;

    while (1)
    {
        ir();
        if (bflag == 1)
        {
            LPC_GPIO0->FIOSET = 0x00000800;
            // Output 10us HIGH on TRIG pin
            LPC_GPIO0->FIOMASK = 0xFFFF7FFF;
            LPC_GPIO0->FIOPIN |= TRIG;
            delayUS(10);
            LPC_GPIO0->FIOCLR |= TRIG;
            LPC_GPIO0->FIOMASK = 0x0;
```

```
            while (!(LPC_GPIO0->FIOPIN & ECHO)) // Wait for a HIGH on ECHO
pin

                startTimer0();

            // LPC_GPIO0->FIOSET = LED_Pinsel << 4;

            // echoTime--;

            while (LPC_GPIO0->FIOPIN & ECHO)

                ;                    // Wait for a LOW on ECHO pin

            echoTime = stopTimer0(); // Stop Counting

            // LPC_GPIO0->FIOCLR = LED_Pinsel << 4;

            distance = (0.0343 * echoTime) / 40;

            sprintf(ans, " Dist:%.2f cm", distance);

            delay(7000000);

            LPC_GPIO2->FIOSET = 1 << 13;

            delay(99999);

            LPC_GPIO2->FIOCLR = 1 << 13;

            delay(999);

            flag1 = 1;

            i = 0;

            flag1 = 0;

            temp1 = 0x01;

            lcd_write();

            flag1 = 1;


            while (msg1[i] != '\0')

            {
```

```
                temp1 = msg1[i];
                lcd_write();
                for (j = 0; j < 100000; j++)
                    ;
                i++;
            }
            i = 0;
            flag1 = 0;
            temp1 = 0xc0;
            lcd_write();
            flag1 = 1;
            while (ans[i] != '\0')
            {
                temp1 = ans[i];
                lcd_write();
                for (j = 0; j < 100000; j++)
                    ;
                i++;
            }
        }
    }
}
```