Security System Implementation Using PIR Sensor

-Hari Narayanan M

BTech. Information Technology

From

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

November 2023

Index

1. Abstract

Security and access control are paramount in today's rapidly evolving technological landscape. Integrating embedded systems into security mechanisms has become crucial for safeguarding space and data. This project focuses on developing a robust security system utilising an LPC 1768 microcontroller featuring a PIR (Passive Infrared) sensor for motion detection and a 4x4 keyboard for password authentication. The significance lies in addressing the growing need for reliable and user-friendly security solutions, offering an accessible yet effective means of access control.

The system's design involves interfacing the LPC 1768 microcontroller with a PIR sensor to detect motion. Upon motion detection, the system prompts the user to input a password via the 4x4 keyboard. The methodology integrates C/C++ programming language, utilising the appropriate libraries for efficient code development. This approach ensures seamless communication between the various components, allowing for swift and accurate processing of sensor inputs and user authentication.

The results demonstrate a functional security system where correct password input triggers the rotation of a stepper motor (which, in real life, can be a gate or a door), signifying authorised access. Conversely, an incorrect password triggers the activation of a buzzer, alerting of unauthorised attempts. The project concludes by highlighting the successful integration of multiple components into a cohesive security framework. It emphasises the system's reliability in authenticating users and its potential for application in various security-sensitive environments, ensuring both accessibility and defence against unauthorised access.

2. Introduction

Objective:

This project aims to design and implement a comprehensive security system using the LPC 1768 microcontroller, PIR sensor, 4x4 keyboard interface, stepper motor, and buzzer. The primary aim is to create an access control mechanism that responds to motion detection by prompting the user to input a password through the keyboard. The system activates the stepper motor upon correct password entry, indicating authorised access, while incorrect attempts trigger the buzzer, signalling unauthorised entry. This project combines hardware interfacing and software development to establish a reliable, user-friendly security solution. The overarching goal is to showcase the practical integration of embedded systems for real-time security applications, emphasising functionality and user interaction.

Scope:

This project's scope extends across multiple domains, encompassing hardware integration, software development, and practical application in embedded systems and security technologies. From a hardware perspective, it involves interfacing a diverse set of components—LPC 1768 microcontroller, PIR sensor, 4x4 keyboard, stepper motor, and buzzer—requiring a thorough understanding of their functionalities and seamless integration. On the software front, the scope includes programming the microcontroller in C/C++ through Keil µVision IDE and Flash Magic, leveraging the embedded platform, and employing relevant libraries to enable effective communication and control between the interconnected hardware elements. Moreover, the project's practical scope involves demonstrating the system's functionality in real-time scenarios, emphasising its utility in access control and security applications. The project's reach extends to potential enhancements, such as incorporating additional security features, expanding compatibility with other devices, and optimising the system's performance and reliability. Overall, the project's scope spans hardware-software integration, practical implementation, and potential avenues for further refinement, positioning it as a comprehensive exploration of embedded security systems.

Project Description:

This project implements a security system using LPC 1768 microcontroller, PIR sensor, 4x4 keyboard, stepper motor, and buzzer. Upon motion detection, users input a password via the keyboard; correct entry triggers the stepper motor, indicating authorised access, while incorrect attempts activate the buzzer. Combining hardware integration, C/C++ programming, the Keil µVision IDE and the Flash Magic platform, the system showcases a robust and user-friendly approach to access control. Its practical application lies in real-time security scenarios, emphasising reliability and potential for future enhancements.

Hardware Requirements:

1. LPC1768 Development Board
2. Passive Infrared Sensor (HC-SR501)
3. LCD Display
4. 4x4 interface Keyboard
5. Buzzer (for alarm)

6. FRC and Female-to-Female Wires
7. Power Supply

Software Requirements:

Language: Embedded C
Software: Keil uVision, Flash Magic

3. Working Principle:

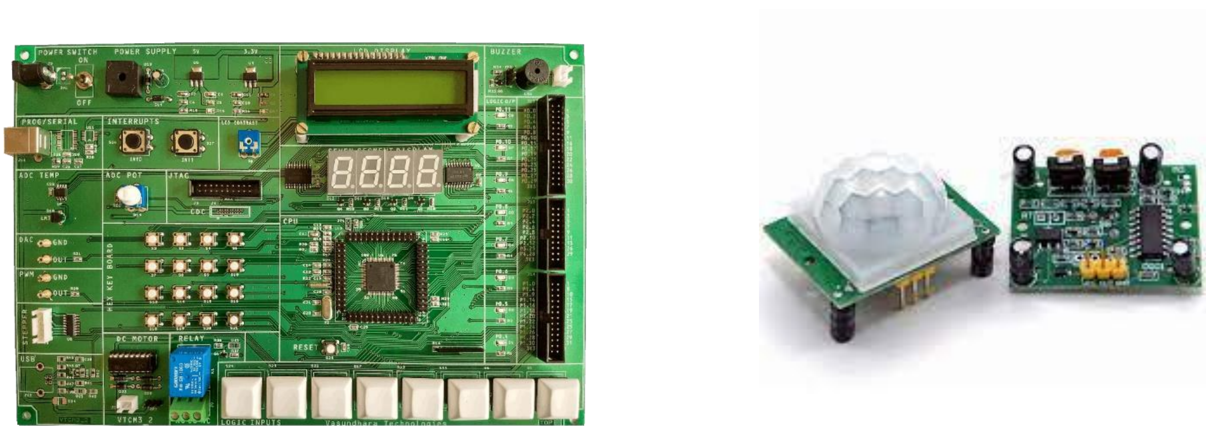

Fig 1. Lpc1768 kit and the PIR Sensor (HC-SR501)

Lpc1768 Microcontroller:

LPC1768 is a microcontroller based on the ARM Cortex-M3 architecture, and NXP Semiconductors manufactures it. It is part of the LPC1700 series of microcontrollers designed for embedded applications. [2] **Processor Core:**
Utilises an ARM Cortex-M3 processor core for high performance and low power consumption, handling the execution of instructions.

Peripherals:
Integrates on-chip peripherals (e.g., UART, SPI, I2C, GPIO, timers) for external devices and general-purpose input/output operations.

Memory:
Utilises Flash memory for program storage and SRAM for data storage during program execution.

Clock System:
It features a sophisticated clock system for precise control of the processor clock frequency, which is crucial for meeting performance requirements and minimising power consumption.

Communication Interfaces and Development Tools:
Supports various communication interfaces (e.g., UART, SPI, I2C) for serial communication and is programmed and debugged using development tools like Keil uVision and Flash Magic.

PIR Sensor (Passive Infrared Sensor):

A PIR (Passive Infrared) sensor is a motion sensor that detects infrared radiation emitted by objects in its field of view and returns a digital output. It's commonly used in security systems, lighting controls, and various automated applications. Here are some applications of the same:

Infrared Radiation Detection:
IR sensors are designed to detect the presence of infrared radiation emitted by objects based on their temperature. This makes them particularly useful for sensing the heat emitted by living organisms or other warm objects.

Motion Detection:

- One of the primary applications of PIR sensors is in motion detection. These sensors can detect changes in the infrared radiation pattern within their field of view, making them effective for triggering alarms or activating devices when motion is detected.[1]

Proximity Sensing:

- PIR sensors are often used for proximity sensing. By measuring the reflection of emitted infrared light, these sensors can determine the distance between the sensor and an object, enabling applications such as touchless switches or automatic faucets.[1]

Object Detection:

- IR sensors can detect the presence or absence of objects in their detection range. This is useful in applications like industrial automation and robotics for object recognition and handling.[1]

Energy Efficiency:

- PIR sensors contribute to energy efficiency by enabling automatic lighting controls. Lights can be turned on or off based on motion detection, reducing energy consumption in spaces when lighting is unnecessary.

Burglar Alarms and Security Systems:

- IR sensors play a crucial role in security systems for detecting intruders. When integrated into burglar alarm systems, they trigger alarms or activate surveillance cameras upon detecting unauthorised motion.[1]

Integration with Microcontrollers:

- IR sensors can be interfaced with microcontrollers to process and interpret the detected signals.

This allows for more advanced control and automation in various applications.

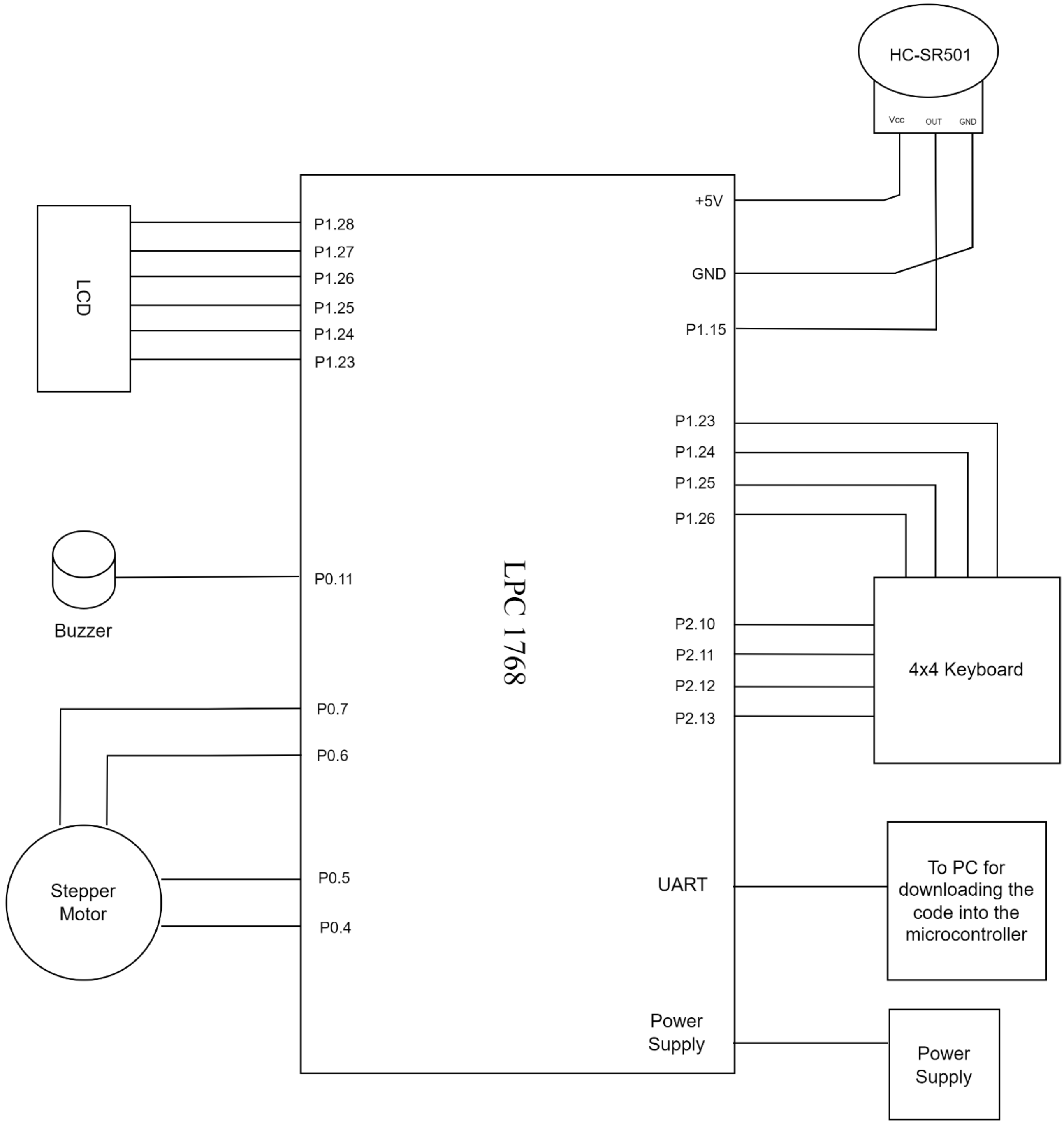

Fig 2. Circuit Diagram used for the project

4. Embedded C Working Code:

```
#include <LPC17xx.h>
#include <stdio.h>
#include <string.h>
int temp1,temp2,flag1,i,j,k,l,m;
int row,x,col;
void clock_wise(void);
void anti_clock_wise(void);
unsigned long int var1,var2;
int pass_len=0,sense=-1;
int command[] = {0x30,0x30,0x30,0x20,0x28,0x01,0x06,0x0C,0x80};
char pass[32];
long int seven[4][4]={{0,1,2,3},{4,5,6,7},{8,9,0xa,0xb},{0xc,0xd,0xe,0xf}};
char message[]="Welcome";
char yes[]="YES !!";
char no[]="NO !!";
char clear[]="                ";
char enter_pass[]="Enter Password";


void port_write(){
	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<30000;j++);
}
void lcd_write(){
	temp2=(temp1>>4)&0xF;
	port_write();
	if( ! (flag1==0 && (temp1==0x20 || temp1 == 0x30) )){
		temp2 = temp1& 0xF;
		port_write();
	}
}
void clock_wise(void) //Stepper motor
{
var1 = 0x00000080; //For Clockwise
for(m=0;m<=3;m++) // for A B C D Stepping
{
//var1 = var1<<1; //For Clockwise
//var2 = ~var1;
//var2 = var2 & 0x000000F0;
```

```
	LPC_GPIO0->FIOMASK=0XFFFFFF0F;
LPC_GPIO0->FIOPIN = var1 ;	//~var1;
	LPC_GPIO0->FIOMASK=0X00000000;
//LPC_GPIO0->FIOSET = var1;
//LPC_GPIO0->FIOCLR = var2;
	var1=var1>>1;
for(k=0;k<60000;k++); //for step speed variation
}
}

void keyboard(){
	while(1){
		//if((LPC_GPIO0->FIOPIN)& (1<<4)){
			int keypressed=0;
			int br=0;
			flag1=0;
			for( i=0;i<9;i++){ temp1 = command[i]; lcd_write();}
			flag1=1;
			i=0;
			while(enter_pass[i] != '\0'){
					if(i==16){
						flag1=0;
						temp1=0xC0;
						lcd_write();
					}
					flag1=1;
					temp1=enter_pass[i];
					lcd_write();
				for(row=0;row<4;row++)
				{
					LPC_GPIO2->FIOPIN=1<<(10+row);
					for(k=0;k<50;k++);
					x=(LPC_GPIO1->FIOPIN>>23)&0XF;
					if(x)
					{
						if(x==1)
							col=0;
						else if(x==2)
							col=1;
						else if(x==4)
							col=2;
						else if(x==8)
							col=3;
						sprintf(pass, "%ld", seven[row][col]);
						keypressed=1;
						break;
					}
					for(k=0;k<=50;k++);
				}
				if(keypressed){
					br=1;
```

```
                    break;
                }
                for(k=0;k<500000;k++);
                i++;
            }
            if(br){
                break;
            }
        //}
    }
    flag1=0;
    temp1=0x01;
    lcd_write();
    for(i=0;i<1000;i++);
    i=0;
    while(pass[i] != '\0'){
        if(i==16){
            flag1=0;
            temp1=0xC0;
            lcd_write();
        }
        flag1=1;
        //temp1=pass[i];
        temp1=42;
        lcd_write();
        for(k=0;k<500000;k++);
        i++;
    }
    pass_len=1;
    while(1){
        long int number;
        int keypressed=0;
        for(row=0;row<4;row++)
        {
            LPC_GPIO2->FIOPIN=1<<(10+row);
            for(k=0;k<50;k++);
            x=(LPC_GPIO1->FIOPIN>>23)&0XF;
            if(x)
            {
                if(x==1)
                    col=0;
                else if(x==2)
                    col=1;
                else if(x==4)
                    col=2;
                else if(x==8)
                    col=3;
                for(k=0;k<5000;k++);
                number = seven[row][col];
                if(number==0xf){
                    keypressed=1;
```

```
                    break;
                }
                if(number>=0 && number<=9){
                    sprintf(pass+pass_len,"%ld",number);
                }
                else if(number>=0xa && number <=0xf){
                    char hexChars[]="ABCDEF";
                    char hexChar = hexChars[number - 10];
                    sprintf(pass+pass_len,"%c",hexChar);
                }
                pass_len++;
                flag1=1;
                //temp1=pass[pass_len-1];
                temp1=42;
                lcd_write();
                if(pass_len==16){
                    flag1=0;
                    temp1=0xC0;
                    lcd_write();
                }
            }
            for(k=0;k<=500000;k++);
        }
        if(keypressed){
            break;
        }
    }
    flag1=0;
    temp1=0x01;
    lcd_write();
    i=0;
    while(pass[i] != '\0'){
        if(i==16){
            flag1=0;
            temp1=0xC0;
            lcd_write();
        }
        flag1=1;
        temp1=pass[i];
        lcd_write();
        for(k=0;k<500000;k++);
        i++;
    }
    if(strcmp(pass,"1234")==0){
        flag1=0;
        temp1=0x01;
        lcd_write();
        i=0;
        while(yes[i] != '\0'){
            if(i==16){
                flag1=0;
```

```
				temp1=0xC0;
				lcd_write();
			}
			flag1=1;
			temp1=yes[i];
			lcd_write();
			for(k=0;k<500000;k++);
			i++;
			for(j=0;j<20;j++) // 20 times in Clock wise Rotation
				clock_wise();
			for(k=0;k<65000;k++);
		}
	}
	else{
		flag1=0;
		temp1=0x01;
		lcd_write();
		i=0;
		while(no[i] != '\0'){
			if(i==16){
				flag1=0;
				temp1=0xC0;
				lcd_write();
			}
			flag1=1;
			temp1=no[i];
			lcd_write();
			for(k=0;k<800000;k++);
			i++;
		}
		for(k=0;k<3;k++){
			LPC_GPIO0->FIOSET=1<<11;
			for(l=0;l<1000000;l++);
			LPC_GPIO0->FIOCLR=1<<11;
			for(l=0;l<900000;l++);
		}
	}
}
int main(){
	SystemInit();
	SystemCoreClockUpdate();
	LPC_PINCON->PINSEL1 = 0;
	LPC_PINCON->PINSEL0=0;
	LPC_PINCON->PINSEL3=0;
	LPC_PINCON->PINSEL4=0;
	LPC_GPIO0->FIODIR=0XFFFF7FFF ; // lcd
	LPC_GPIO2->FIODIR=0XF<<10; // keyboard row
	// ------------- SENSOR -------------
	LPC_PINCON->PINSEL3 = 0; //SENSOR
	LPC_GPIO0->FIODIR |= 0xFF<<4;
	LPC_GPIO1->FIODIR = 0;
```

```
        //---------------END-----------------
        flag1=0;
        for( i=0;i<9;i++){ temp1 = command[i]; lcd_write();}
while(1){
redirect:
        flag1=0;
        temp1=0x01;
        lcd_write();
        i=0;
        while(message[i] != '\0'){
                if(i==16){
                        flag1=0;
                        temp1=0xC0;
                        lcd_write();
                }
                flag1=1;
                temp1=message[i];
                lcd_write();
                for(k=0;k<500000;k++);
                i++;
        }
        //while(1){
        sense=LPC_GPIO0->FIOPIN&1<<15;
        if(sense){
                for(k=0;k<5000;k++);
                LPC_GPIO0->FIOSET=0xF<<4;
                for(k=0;k<5000;k++);
                keyboard();
                goto redirect;
        }
        else{
                LPC_GPIO0->FIOCLR=0xFF<<4;
        }
        for(k=0;k<10;k++);
}
}
```

5. Demonstration Of the Security System using LPC1768:

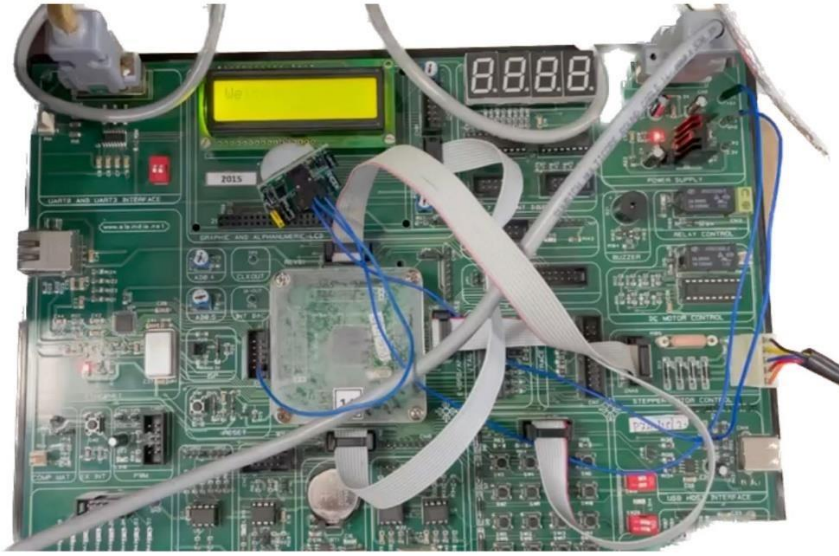

Fig 3(a). Working Project on LPC 1768

Fig3(b). Stepper Motor, which is connected to LPC 1768

Fig 3. Complete Working Project

6. Conclusion:

In conclusion, developing and implementing the security system using the LPC 1768 microcontroller, PIR sensor, 4x4 keyboard interface, stepper motor, and buzzer have yielded a robust and functional access control mechanism. This project successfully showcased the seamless integration of hardware components and software programming, culminating in an effective solution for real-time security applications.

The system's ability to detect motion, prompt for password authentication, and respond accordingly with stepper motor activation for authorized access or buzzer alert for unauthorized attempts highlights its reliability and practicality. Moreover, the project's exploration of the embedded platform, C/C++ programming, and hardware interfacing served as a comprehensive learning experience, demonstrating the potential for embedded systems in addressing contemporary security challenges.

While this project has achieved its predefined objectives by creating a working prototype, its significance extends beyond the prototype's functionality. The project serves as a stepping stone for further advancements, encouraging the exploration of enhanced security features, remote access capabilities, and seamless integration with evolving technologies. Its scalability, adaptability, and potential for future enhancements underscore its relevance in the ever-evolving landscape of embedded security systems. Ultimately, this endeavour contributes to the ongoing pursuit of creating sophisticated yet accessible solutions to ensure security and access control in diverse environments.

7. Future Scope:

The project on IR sensor interfaced with the LPC1768 microcontroller for motion detection presents several avenues for future enhancements and expansions:

1. **Enhanced Security Features**: Implementing multiple user profiles or more sophisticated authentication methods like biometrics (fingerprint or facial recognition) to bolster security.
2. **Remote Access**: Integrating wireless communication modules (like Wi-Fi or Bluetooth) to enable remote monitoring and control of the security system via a smartphone or web interface.
3. **Data Logging and Analytics**: Adding a feature to log access attempts, providing insights into potential security breaches and user activity patterns.
4. **Smart Home Integration**: Integrating the security system into a broader smart home ecosystem, allowing it to interact with other smart devices or systems.
5. **Advanced Intrusion Detection**: Integrate machine learning algorithms to differentiate between benign and suspicious movements, reducing false alarms and enhancing the system's accuracy.
6. **Secure Communication Protocols**: Implement encryption methods to secure data transmission between the components, ensuring confidentiality and integrity of sensitive information.
7. **Redundancy and Fail-Safe Mechanisms**: Develop backup systems or alternative authentication methods in case of component failure or hacking attempts, ensuring uninterrupted security.
8. **Integration with Home Automation**: Extend the project to integrate with home automation systems, allowing the security system to trigger lights, cameras, or alert other devices in case of unauthorised access.
9. **Cloud Integration for Data Storage and Analysis**: Utilize cloud services for storing access logs and conducting a deeper analysis of access patterns, potentially predicting and preventing security breaches.
10. **Accessibility Features**: Implement features for disabled users, ensuring inclusivity in accessing and controlling the security system.

8. References:

[1] https://www.ourpcb.com/pir-sensors.html

[2] https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/generalpurpose-mcus/lpc1700-arm-cortex-m3/512-kb-flash-64-kb-sram-ethernet-usb-lqfp100package:LPC1768FBD100