**Documentation:**

**Introduction:**

The smart security system project aims to create a reliable and efficient security solution using IoT (Internet of Things) technology. The system utilizes a PIR (Passive Infrared) sensor to detect motion, a GSM (Global System for Mobile Communications) module to send SMS alerts, and an Arduino Uno microcontroller for processing and control.

The primary purpose of the project is to provide real-time monitoring of unauthorized movements in a specific area and alert the user via SMS. This system can be deployed in homes, offices, or any location where enhanced security is required.

**Hardware Setup:**

**PIR Sensor Connection:**

PIR Sensor VCC to Arduino 5V

PIR Sensor GND to Arduino GND

PIR Sensor OUT to Arduino digital pin (e.g., Pin 2)

GSM Module Connection:

GSM Module VCC to Arduino 5V

GSM Module GND to Arduino GND

GSM Module TX to Arduino RX (Pin 0)

GSM Module RX to Arduino TX (Pin 1)

**Arduino Sketch:**

#include <SoftwareSerial.h>

#define PIR\_PIN 2

#define GSM\_RX 0

#define GSM\_TX 1

SoftwareSerial gsm(GSM\_RX, GSM\_TX);

void setup() {

pinMode(PIR\_PIN, INPUT);

Serial.begin(9600);

gsm.begin(9600);

}

void loop() {

int pirState = digitalRead(PIR\_PIN);

if (pirState == HIGH) {

Serial.println("Motion detected!");

sendSMS("Alert: Motion detected!");

delay(10000); // Delay to avoid sending multiple messages

}

}

void sendSMS(String message) {

gsm.println("AT+CMGF=1"); // Set SMS mode to text

delay(1000);

gsm.print("AT+CMGS=\"+1234567890\"\r"); // Replace with recipient's phone number

delay(1000);

gsm.println(message);

delay(1000);

gsm.println((char)26); // End of message character

delay(1000);

}

**Arduino Sketch Explanation:**

**Libraries:**

#include <SoftwareSerial.h>

This line includes the SoftwareSerial library, which allows serial communication on other digital pins of the Arduino board.

**Pin Definitions:**

#define PIR\_PIN 2

#define GSM\_RX 0

#define GSM\_TX 1

These #define statements define the pin numbers used for the PIR sensor input (PIR\_PIN), and the RX and TX pins for communication with the GSM module (GSM\_RX and GSM\_TX).

**SoftwareSerial Object Initialization:**

SoftwareSerial gsm(GSM\_RX, GSM\_TX);

This line initializes a SoftwareSerial object named gsm, specifying the RX and TX pins for communication with the GSM module.

**Setup Function:**

void setup() {

pinMode(PIR\_PIN, INPUT);

Serial.begin(9600);

gsm.begin(9600);

}

**In the setup() function:**

pinMode(PIR\_PIN, INPUT); sets the PIR\_PIN as an input pin, as it will be reading the state of the PIR sensor.

Serial.begin(9600); initializes serial communication with the Arduino IDE Serial Monitor at a baud rate of 9600.

gsm.begin(9600); initializes serial communication with the GSM module at a baud rate of 9600.

**Loop Function**:

void loop() {

int pirState = digitalRead(PIR\_PIN);

if (pirState == HIGH) {

Serial.println("Motion detected!");

sendSMS("Alert: Motion detected!");

delay(10000); // Delay to avoid sending multiple messages

}

}

**In the loop() function:**

digitalRead(PIR\_PIN) reads the state of the PIR sensor connected to the PIR\_PIN. If motion is detected, the state will be HIGH.

If motion is detected (pirState == HIGH), it prints "Motion detected!" to the Serial Monitor using Serial.println().

It then calls the sendSMS() function to send an SMS alert.

To prevent multiple alerts for continuous motion, it includes a delay of 10 seconds (delay(10000)).

**sendSMS Function:**

void sendSMS(String message) {

gsm.println("AT+CMGF=1"); // Set SMS mode to text

delay(1000);

gsm.print("AT+CMGS=\"+1234567890\"\r"); // Replace with recipient's phone number

delay(1000);

gsm.println(message);

delay(1000);

gsm.println((char)26); // End of message character

delay(1000);

}

The sendSMS() function is responsible for sending an SMS alert.

It sends AT commands to the GSM module via gsm.println() to configure SMS mode (AT+CMGF=1), set the recipient's phone number, and send the message.

It includes delays between commands to ensure proper communication with the GSM module.

This sketch continuously monitors the PIR sensor for motion. When motion is detected, it prints a message to the Serial Monitor and sends an SMS alert using the GSM module.

**Testing:**

**To test the system:**

Upload the Arduino sketch to the Arduino Uno.

Power up the Arduino Uno and GSM module.

Move in front of the PIR sensor to trigger motion detection.

You should receive an SMS alert on the configured phone number.

**Troubleshooting:**

Ensure all connections are correct.

Verify that the GSM module has a valid SIM card with sufficient credit.

Check for any errors in the Arduino sketch.

Integration:

**Integrating the system into a real-world security setup involves:**

Mounting the PIR sensor in a strategic location to cover the desired area.

Placing the GSM module in a secure location with access to a reliable power source.

Configuring the system to send alerts to the appropriate phone number(s).

**Conclusion:**

In conclusion, the smart security system provides an effective solution for monitoring and securing premises against unauthorized access. Future enhancements could include:

Adding more sensors for comprehensive monitoring (e.g., door/window sensors).

Implementing a web-based interface for remote monitoring and control.

Integrating with other IoT devices for enhanced automation and security features