**BT22CIV101 NAMIT LOHARKAR**

**EXPERIMENT NO -5**

**TITLE** **:**

Designing an anti-theft security system using capacitive touch sensors for the locker such that owner gets informed if a theft gets detected using Bluetooth Classic module of ESP 32.

**AIM:**

Design an anti-theft security system using capacitive touch sensors for the locker. The person needs to enter a security code of 6-digits i.e. “975301” to unlock a locker within 30 seconds. If the code entered by a person is correct, the locker opens. If the code is wrong or not entered within 30 seconds an alarm buzzes continuously. Inform the owner about the thief. Only owner can stop the buzzing of alarm by pressing a button.

**APPRATUS:**

1)computer or laptop with installed ARDUINO IDE

2)ESP 32 microcontroller with board

3)connecting wires (female to female or male to female)

4)buzzer

5)push button

6) led for checking and convenience

7)USB cable for supply to ESP 32

**THEORY :**

The ESP32 is a powerful microcontroller with built-in Wi-Fi and Bluetooth capabilities. Bluetooth Classic, one of the protocols supported by ESP32, enables wireless communication over short distances. In this system, Bluetooth is used to notify the owner on their mobile device in the event of a security breach.

Capacitive touch sensors detect the presence of human touch by sensing changes in capacitance. When placed on the surface of a locker, the sensors detect if someone tries to open it. If an unauthorized touch is detected, the system triggers the alert mechanism.

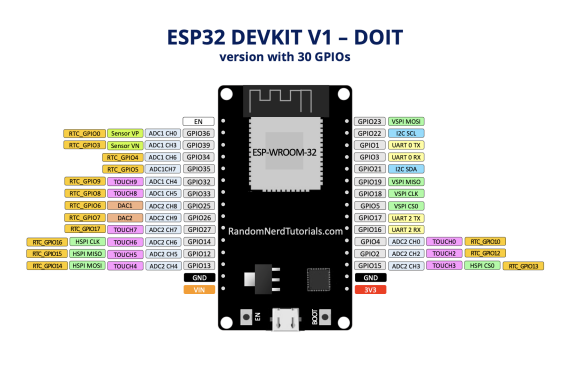
**Components:**

1. Buzzer: It produces a sound alarm when unauthorized access is detected.

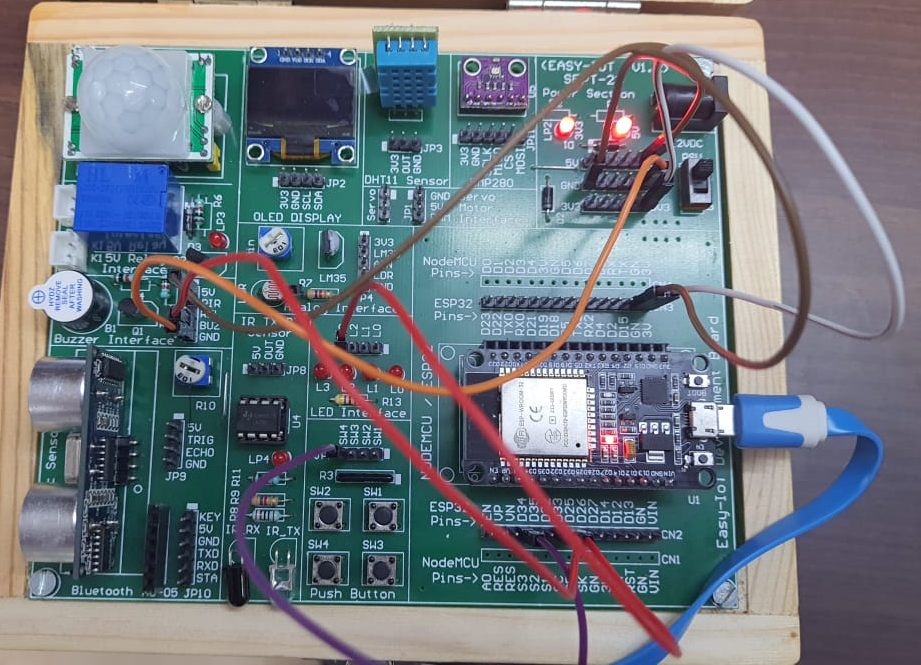
2. Pushbutton: Used to arm or disarm the system manually.

3. ESP32: Serves as the brain of the system, managing sensor input, Bluetooth communication, and alarm activation.

**ESP 32 PIN DIAGRAM:**



**CIRCUIT DIAGRAM:**



**PROCEDURE:**

1. Enter the correct code that will open the lock.  
2. Now make string variable that will store the code entered by the user.  
3. Next declare and assign the touch pin, buzzer pin and button pin.  
5. Now develop the code according to the problem statement integrating the hardware with the ESP module.   
6. Since there is no dc servo motor available we will light the led in place of it.  
7. Take the string input from keyboard at serial monitor.   
8. Develop the logic and upload it to arduino.

**CODE:**

#include <Wire.h>

#include <BluetoothSerial.h>

#include "driver/touch\_pad.h"

BluetoothSerial SerialBT;

const int TP\_9 = 32;

const int TP\_7 = 23;

const int TP\_5 = 19;

const int TP\_3 = 35;

const int TP\_0 = 23;

const int TP\_1 = 19;

const int LED = 13;

const int ALARM = 12;

const int STOP\_BTN = 14;

const String accessCode = "975301";

String inputSequence = "";

bool alarmTriggered = false;

unsigned long timeStarted;

unsigned long maxTime = 30000;

bool checkTouch(int touchPad) {

return touchRead(touchPad) < 50;

}

void setup() {

Serial.begin(9600);

SerialBT.begin("SecureLocker");

pinMode(LED, OUTPUT);

pinMode(ALARM, OUTPUT);

pinMode(STOP\_BTN, INPUT\_PULLUP);

digitalWrite(LED, LOW);

digitalWrite(ALARM, LOW);

Serial.println("System is active. Awaiting input...");

}

void loop() {

if (checkTouch(TP\_9)<50) { inputSequence += '9'; Serial.println(9);delay(1000);}

if (checkTouc (TP\_7)<50) { inputSequence += '7';Serial.println(7);delay(1000); }

if (checkTouch(TP\_5)<5) { inputSequence += '5';Serial.println(5);delay(1000); }

if (checkTouch(TP\_3)) { inputSequence += '3'; Serial.println(3);delay(1000);}

if (checkTouch(TP\_0)) { inputSequence += '0';Serial.println(0); delay(1000);}

if (checkTouch(TP\_1)) { inputSequence += '1';Serial.println(1);delay(1000); }

if (inputSequence.length() == 1) {

timeStarted = millis();

}

if (inputSequence.length() == 3) {

if (inputSequence == accessCode) {

Serial.println("Access granted! Locker is open.");

SerialBT.println("Access granted! Locker is open.");

digitalWrite(LED, HIGH);

inputSequence = "";

} else {

activateAlarm();

}

}

if (millis() - timeStarted > maxTime && inputSequence.length() > 0) {

activateAlarm();

}

if (alarmTriggered) {

if (digitalRead(STOP\_BTN) == LOW) {

deactivateAlarm();

}

if (SerialBT.available()) {

String command = SerialBT.readString();

if (command == "DISABLE\_ALARM") {

deactivateAlarm();

}

}

}

}

void activateAlarm() {

Serial.println("Wrong input or timeout! Alarm activated.");

delay(5000);

alarmTriggered = true;

digitalWrite(ALARM, HIGH);

SerialBT.println("ALARM\_ON");

inputSequence = "";

}

void deactivateAlarm() {

Serial.println("Alarm deactivated by user.");

SerialBT.println("Alarm deactivated by user.");

alarmTriggered = false;

digitalWrite(ALARM, LOW);

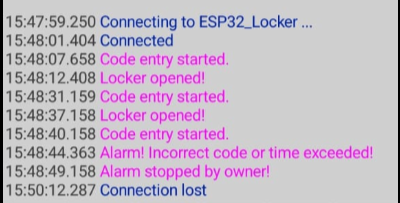
digitalWrite(LED, LOW);

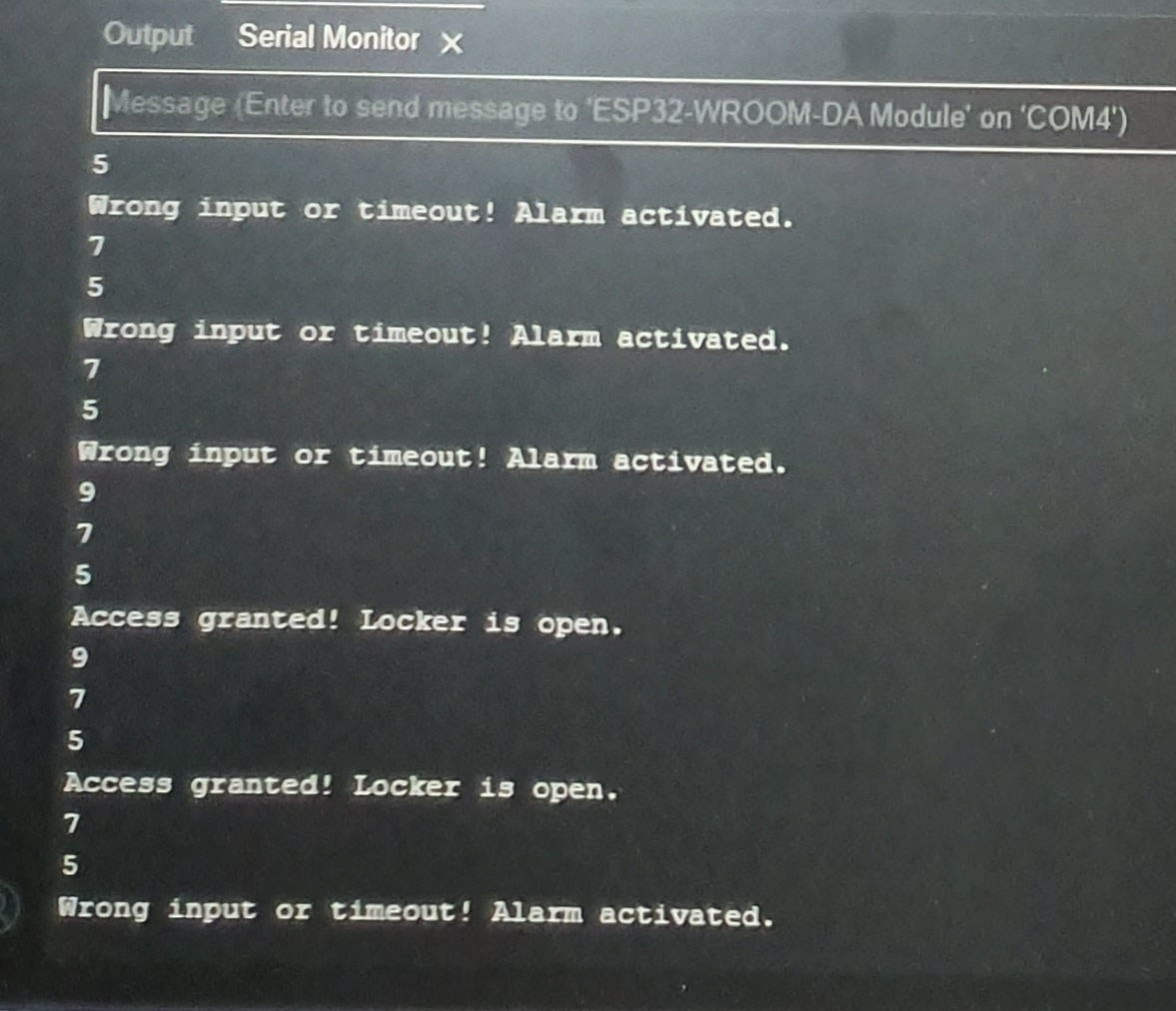
inputSequence = "";

}

**OBSERVATION:**

1. Observe Bluetooth connectivity
2. See on pressing pushbutton alarm is stopping or not



****

**CONCLUSION:**

The conclusion of this experiment is that a reliable and efficient anti-theft security system can be successfully designed using capacitive touch sensors and the ESP32 microcontroller. By integrating Bluetooth Classic for wireless communication, the system is capable of immediately alerting the owner of any unauthorized access to the locker. The experiment demonstrates how low-cost components such as touch sensors and buzzers, combined with the versatile ESP32, can be effectively used to enhance security measures in real-time applications. This system provides a scalable and adaptable solution for various security needs.