



# University of Asia Pacific

Department of Computer Science and Engineering

## CSE 316: Microprocessors and Microcontrollers Lab

### LAB REPORT

---

**Experiment Number: 5**

**Experiment Title:** Wireless Theft Detection using Door Sensor and Buzzer

**Submitted by:**

**Name : Md. Sakib Hossaine**

**Student ID : 22201185**

**Section : D1**

**Submitted to:**

**Zaima Sartaj Taheri**

**Lecturer,  
Department of Computer Science and Engineering**

**Date of Submission: 21,September, 2025**

## 1. Experiment Name

- Mini Project 5: Wireless Theft Detection using Door Sensor and Buzzer

## 2. Objective

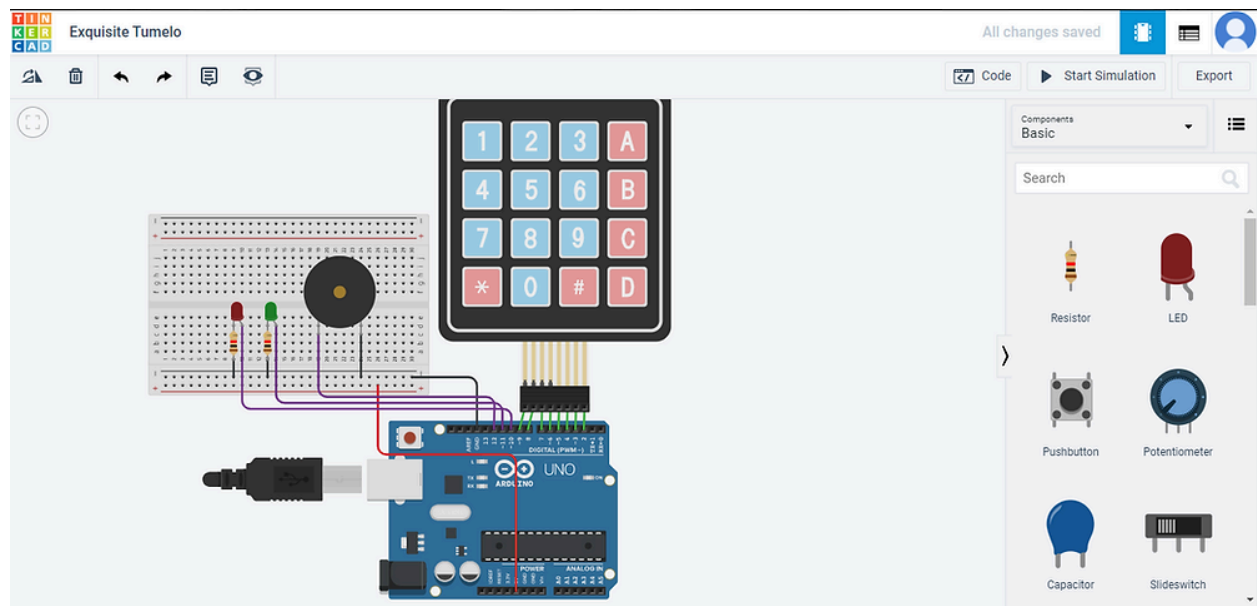
-To create a wireless security system that detects unauthorized door openings and triggers an immediate alarm using a door sensor and buzzer.

## 3. Apparatus / Hardware & Software Requirements

- List all required tools and components:

- Microcontroller (Arduino uno R3)
- Sensors / Actuators ( Light, 4x4 Keypad , Buzzer)
- Software (Arduino IDE,Tinkercad)
- Breadboard, Jumper Wires, Power Source, resistors etc.

## 4. Circuit Diagram / Schematic



## 5. Code / Assembly Program

```
#include <Keypad.h> // Library for handling matrix keypad input
```

```
// ----- Keypad Setup -----
```

```
const byte Rows = 4, Cols = 4; // 4 rows and 4 columns in keypad
```

```
char keys[Rows][Cols] = {           // Key mapping (layout of the keypad)
```

```

    {'1','2','3','A'},

    {'4','5','6','B'},

    {'7','8','9','C'},

    {'*','0','#','D'}

};

byte RowPINS[Rows] = {10, 9, 8, 7}; // Connect keypad ROWs to Arduino pins

byte ColPINS[Cols] = {6, 5, 4, 3}; // Connect keypad COLs to Arduino pins


// Create keypad object

Keypad keypad = Keypad(makeKeymap(keys), RowPINS, ColPINS, Rows, Cols);


// ----- Pins -----

const int DoorPin = 13;

const int Buzzer = 12;

const int LedPin = 11;


// ----- Variables -----

String PinCode = "4321"; // Default PIN

String InputPin = "";      // Stores entered PIN

bool Armed = false;        // System armed/disarmed status

bool AlarmActive = false; // Alarm ON/OFF status


// Timing for wrong PIN beep

unsigned long WrongBeepStart = 0;          // Start time of wrong PIN beep

bool WrongBeepActive = false;              // Tracks if wrong beep is active

const unsigned long WrongBeepDuration = 400; // Duration of wrong beep (0.4 sec)

```

```

void setup() {

  pinMode(DoorPin, INPUT_PULLUP); // Door sensor uses pull-up (LOW = Open)

  pinMode(Buzzer, OUTPUT);      // Buzzer as output

  pinMode(LedPin, OUTPUT);      // LED as output

}

```

```

void loop() {

  char key = keypad.getKey(); // Read keypad input


  // ----- PIN Handling -----

  if (key) { // If a key is pressed

    switch (key) {

      case '#': // Check entered PIN

        if (InputPin == PinCode) {

          Armed = !Armed; // Toggle system (ARM/DISARM)

          if (!Armed) {

            AlarmActive = false; // Turn off alarm when disarming

            noTone(Buzzer);      // Stop buzzer

            digitalWrite(LedPin, LOW); // Turn off LED

          }

        } else {

          tone(Buzzer, 1000);      // Play short beep

          WrongBeepStart = millis(); // Record start time

          WrongBeepActive = true;   // Mark wrong beep as active

        }

      }

    }

  }

}

```

```

    InputPin = ""; // Reset PIN entry

    break;

    case '*': // Clear PIN input

    InputPin = "";

    break;

    default: // If number or letter entered, add to PIN

    InputPin += key;

    }
}

// ----- Door + Alarm -----

if (Armed && digitalRead(DoorPin) == LOW) { // Door opened while ARMED

    if (!AlarmActive) {

        AlarmActive = true; // Trigger alarm

    }

}

// ----- Handle wrong PIN beep duration -----

if (WrongBeepActive && millis() - WrongBeepStart >= WrongBeepDuration) {

    WrongBeepActive = false; // End wrong beep

    if (!AlarmActive) noTone(Buzzer); // Stop buzzer if alarm not active

}

// ----- Alarm Buzzer & LED -----

```

```

if (AlarmActive) {

    tone(Buzzer, 2000);      // Continuous alarm beep

    digitalWrite(LedPin, HIGH); // LED ON

} else if (!WrongBeepActive) {

    noTone(Buzzer);          // Turn off buzzer (if no wrong beep or alarm)

    digitalWrite(LedPin, LOW); // LED OFF

}

}

```

## 6. Output / Observations

1. The system detects door movement via a wireless door sensor.
2. The buzzer emits a loud, continuous alarm when unauthorized access is detected.
3. A red LED lights up to indicate active alarm status during triggering.
4. The system resets automatically or manually after alarm activation.

## 7. Result

The prototype successfully detected door openings wirelessly and triggered the buzzer alarm consistently, providing a reliable theft deterrent mechanism. The visual LED indicator effectively complemented the audio alert for immediate user awareness.

## 8. Conclusion

This project demonstrated the practical implementation of a low-cost, wireless theft detection system using basic components. Though limited by range and power constraints, it effectively showcased core security principles and real-time alert mechanisms for home or small-scale use.