TFB2093 Internet-of-Things

Practical 04: Interfacing with Digital Sensors & Actuators (PIR + Buzzer + LEDs + Buttons)

Objectives

• Read a **PIR motion sensor** with *digitalRead()*.

• Drive a **buzzer / LED / relay** with *digitalWrite()*.

• Build a **mini alarm system** with **Arm/Disarm** and **Silent Mode** toggles.

• Log status to **Serial Monitor**.

0) Do this in Tinkercad

• Create → Circuits → New Circuit.

• Add parts: Arduino UNO, PIR sensor, Piezo buzzer, 2× pushbuttons, 2× LEDs, 2× 220Ω resistors, wires.

• Set the PIR to “Digital” mode (default in Tinkercad PIR).

Suggested Pin Map (easy wiring)

Device	Pin on UNO	Notes
PIR OUT	D2	5V and GND to PIR VCC/GND
Buzzer +	D9	Buzzer – to GND
ARM LED (status)	D10 (via 220Ω)	LED – to GND
Motion LED	D11 (via 220Ω)	LED – to GND
Arm/Disarm Button	D4	Other leg to GND (INPUT_PULLUP)
Silent Mode Button	D5	Other leg to GND (INPUT_PULLUP)

Why INPUT_PULLUP? It removes external resistors. Button released = HIGH, pressed = LOW.

1) Activity: Read a PIR (Digital In)

Wiring (minimum): PIR VCC → 5V, GND → GND, OUT → D2

```
// Step-1: PIR read demo
const int PIR = 2;

void setup() {
  pinMode(PIR, INPUT);
  Serial.begin(9600);
}

void loop() {
  int motion = digitalRead(PIR);  // HIGH = motion
  if (motion == HIGH) {
    Serial.println("Motion Detected");
  } else {
    Serial.println("No Motion");
  }
  delay(300);
}
```

Expected: Serial Monitor prints “Motion Detected” / “No Motion”.

2) Activity: Digital Out (Buzzer / LED / Relay)

Wiring: Buzzer + → D9, – → GND; Motion LED Anode → D11 via 220Ω, Cathode → GND

```
// Step-2: PIR activates buzzer/LED
const int PIR = 2;
const int BUZ = 9;
const int LEDM = 11;

void setup() {
  pinMode(PIR, INPUT);
  pinMode(BUZ, OUTPUT);
  pinMode(LEDM, OUTPUT);
}

void loop() {
  int motion = digitalRead(PIR);
  if (motion == HIGH) {
    digitalWrite(LEDM, HIGH);
    tone(BUZ, 2000);         // simple alarm tone
  } else {
    digitalWrite(LEDM, LOW);
    noTone(BUZ);
  }
  delay(50);
}
```

3) Activity: Add Arm/Disarm Status LED + Button

Wiring: ARM LED Anode → D10 via 220Ω, Cathode → GND; Arm Button: one leg → D4, other → GND (INPUT_PULLUP).

```
// Step-3: Arm/Disarm with a button (toggle)
const int PIR   = 2;
const int BUZ   = 9;
const int LEDM  = 11;  // motion LED
const int LEDA  = 10;  // armed LED
const int BTN_ARM = 4;

bool armed = false;
int prevArm = HIGH;       // because INPUT_PULLUP -> idle HIGH
unsigned long lastDebounce = 0;

void setup() {
  pinMode(PIR, INPUT);
  pinMode(BUZ, OUTPUT);
  pinMode(LEDM, OUTPUT);
  pinMode(LEDA, OUTPUT);
  pinMode(BTN_ARM, INPUT_PULLUP);
  Serial.begin(9600);
}

void loop() {
  // --- button toggle with simple debounce ---
  int readArm = digitalRead(BTN_ARM);
  if (readArm != prevArm && (millis() - lastDebounce) > 30) {
    lastDebounce = millis();
    if (readArm == LOW) {         // pressed
      armed = !armed;
      Serial.println(armed ? "System ARMED" : "System DISARMED");
    }
    prevArm = readArm;
  }

  digitalWrite(LEDA, armed ? HIGH : LOW);  // show armed status

  // --- alarm logic ---
```

```
  int motion = digitalRead(PIR);
  if (motion == HIGH) {
    digitalWrite(LEDM, HIGH);
    if (armed) tone(BUZ, 2000); else noTone(BUZ);
  } else {
    digitalWrite(LEDM, LOW);
    noTone(BUZ);
  }
}
```

4) Activity: Silent Mode Toggle (log motion, no sound)

Wiring: Silent Button: one leg → D5, other leg → GND (INPUT_PULLUP).

Final Combined Sketch (Arm/Disarm + Silent Mode)

```
// Practical 04 - Mini Alarm System (Tinkercad-ready)
// PIR + Buzzer + LEDs + Arm/Disarm + Silent Mode

// Pins
const int PIR     = 2;   // PIR OUT
const int BUZ     = 9;   // Buzzer +
const int LED_M   = 11;  // Motion LED
const int LED_A   = 10;  // Armed LED
const int BTN_ARM = 4;   // Arm/Disarm (to GND)
const int BTN_SIL = 5;   // Silent Mode (to GND)

// States
bool armed = false;
bool silentMode = false;

// For simple debounce / edge detect
int prevArm = HIGH;
int prevSil = HIGH;
unsigned long lastDebounceArm = 0;
unsigned long lastDebounceSil = 0;

void setup() {
  pinMode(PIR, INPUT);
  pinMode(BUZ, OUTPUT);
  pinMode(LED_M, OUTPUT);
  pinMode(LED_A, OUTPUT);
  pinMode(BTN_ARM, INPUT_PULLUP);
  pinMode(BTN_SIL, INPUT_PULLUP);
  Serial.begin(9600);
  Serial.println("Boot OK. System DISARMED. Silent=OFF");
}

void loop() {
  // --- Toggle Arm ---
  int readArm = digitalRead(BTN_ARM);
  if (readArm != prevArm && (millis() - lastDebounceArm) > 30) {
    lastDebounceArm = millis();
    if (readArm == LOW) {
      armed = !armed;
      Serial.println(armed ? "System ARMED" : "System DISARMED");
    }
    prevArm = readArm;
  }
  digitalWrite(LED_A, armed ? HIGH : LOW);

  // --- Toggle Silent Mode ---
  int readSil = digitalRead(BTN_SIL);
  if (readSil != prevSil && (millis() - lastDebounceSil) > 30) {
    lastDebounceSil = millis();
    if (readSil == LOW) {
      silentMode = !silentMode;
      Serial.println(silentMode ? "Silent Mode: ON" : "Silent Mode: OFF");
    }
    prevSil = readSil;
  }

  // --- Motion handling ---
  int motion = digitalRead(PIR);     // HIGH = motion
  if (motion == HIGH) {
    digitalWrite(LED_M, HIGH);
    Serial.println("Motion Detected");
```

```
      if (armed && !silentMode) {
        tone(BUZ, 2000);
      } else {
        noTone(BUZ);
      }
    } else {
      digitalWrite(LED_M, LOW);
      noTone(BUZ);
    }

    delay(40); // gentle pacing
  }
```

Lab Tasks (Demo Checklist)

• PIR only: Serial logs change with movement.

• Actuator: PIR lights motion LED and drives buzzer.

• Arm/Disarm: Toggle with the Arm button; status LED + Serial message.

• Silent Mode: Toggle; motion still logs, buzzer stays off.

Quick Troubleshooting (Tinkercad)

• If the PIR never changes, run simulation at fast speed and allow 5–10 s warm■up.

• INPUT_PULLUP logic: released=HIGH, pressed=LOW.

• Check GND connections for LEDs and buzzer.

Extension Ideas (still Tinkercad■only)

• Replace buzzer with a relay to simulate a siren.

• Add a blink pattern on the armed LED when Silent Mode is ON.

• Log timestamps (millis) to measure detection latency.