✨TFB2093 Internet-of-Things

Practical 08 – IoT Programming (TinkerCAD Version)

Topic: ***Simulating Raspberry Pi + Sense HAT using Arduino in TinkerCAD***

🐥Practical Info

Course: TFB2093 Internet-of-Things
Practical: 08
Platform: TinkerCAD Circuits (Virtual Only)
Concept: Sense HAT → simulated using **Arduino + LEDs + sensor + push buttons**

🌀Learning Objectives

By the end of this practical, you will be able to:

1. Simulate **environmental sensing** (e.g., temperature).
2. Build a **3×3 LED matrix** and display icons.
3. Use **push buttons as a joystick** to move a pixel.
4. Combine everything into a **mini interactive game**.

🍂TinkerCAD Components

You will need these **virtual components**:

- Arduino Uno R3
- Breadboard
- Temperature sensor (TMP36)
- 9 LEDs (3×3 grid)
- 9 × 220Ω resistors
- 5 push buttons (Up, Down, Left, Right, Center)
- Jumper wires

> 💡 *Build step-by-step. Test after each small change.*

🍁Activity 1 – Base Circuit & Test Sketch

🌀Goal: Ensure Arduino simulation is running properly.

Steps: 1. Create a **New Circuit** in TinkerCAD.
2. Add Arduino Uno + Breadboard.
3. Connect **5V → + rail** and **GND → – rail**.
4. Switch to **Code → Text**.
5. Upload this test sketch:

```
void setup() {
  Serial.begin(9600);
  Serial.println("Practical 08: TinkerCAD IoT Simulation Started");
}

void loop() {
  Serial.println("Arduino is running...");
  delay(1000);
}
```

1. Start Simulation → Open Serial Monitor.

🔧Should print: *Arduino is running…*

🐜Activity 2 – Temperature Sensor (TMP36)

🌀Goal: Simulate Sense HAT environmental sensing.

Wiring:

- +Vs → 5V
- Vout → A0
- GND → GND

Code:

```
const int TEMP_PIN = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int raw = analogRead(TEMP_PIN);
  float voltage = raw * (5.0 / 1023.0);
  float tempC = (voltage - 0.5) * 100.0;

  Serial.print("Temp (C): ");
  Serial.println(tempC);

  delay(1000);
}
```

🔧Temperature values should update every second.

💐Activity 3 – 3×3 LED Matrix (Pixel Patterns)

🌀Goal: Simulate the Sense HAT LED matrix.

Wiring:

- LEDs arranged in **3×3 grid**
- Short leg → resistor → GND
- Long leg → Arduino pins 2–10

Suggested Pin Mapping:

Row 1 → 2, 3, 4
Row 2 → 5, 6, 7
Row 3 → 8, 9, 10

Code:

```
const int ledPins[3][3] = {
  {2,3,4},
  {5,6,7},
  {8,9,10}
};

void setup() {
  for(int r=0;r<3;r++){
    for(int c=0;c<3;c++){
      pinMode(ledPins[r][c], OUTPUT);
    }
  }
}

void showPattern(byte pattern[3][3]){
  for(int r=0;r<3;r++){
    for(int c=0;c<3;c++){
      digitalWrite(ledPins[r][c], pattern[r][c] ? HIGH : LOW);
    }
  }
}

void loop(){
  byte cross[3][3] = {
    {1,0,1},
    {0,1,0},
    {1,0,1}
  };

  showPattern(cross);
}
```

📝**Try:** Create heart/smiley/arrow patterns.

🌝Activity 4 – Joystick with Push Buttons

🌀Goal: Move a single LED (player pixel) on the grid.

Button Pins:

- UP → 11
- DOWN → 12
- LEFT → A1
- RIGHT → A2
- CENTER → A3

Wiring:

- One side → GND
- Other side → pin
- Use internal pull-ups

Code:

```
const int BTN_UP=11, BTN_DOWN=12, BTN_LEFT=A1, BTN_RIGHT=A2, BTN_OK=A3;
int playerRow=1, playerCol=1;

void clearAll(){
  for(int r=0;r<3;r++) for(int c=0;c<3;c++) digitalWrite(ledPins[r][c],LOW);
}

void drawPlayer(){
  clearAll();
  digitalWrite(ledPins[playerRow][playerCol], HIGH);
}

void loop(){
  if(digitalRead(BTN_UP)==LOW && playerRow>0) playerRow--;
  if(digitalRead(BTN_DOWN)==LOW && playerRow<2) playerRow++;
  if(digitalRead(BTN_LEFT)==LOW && playerCol>0) playerCol--;
  if(digitalRead(BTN_RIGHT)==LOW && playerCol<2) playerCol++;

  drawPlayer();
  delay(150);
}
```

🔧LED moves like a joystick-controlled cursor.

🌲Activity 5 – Simple Reaction Game

🌀Goal: Combine everything into a mini-game.

Game Logic:

1. Light up a **random LED**.
2. Start timer.
3. Player presses **CENTER** quickly.
4. If fast → WIN pattern + reaction time.
5. If slow (>3000 ms) → LOSE pattern.
6. New random target.

Starter Code:

```
unsigned long targetTime;
bool targetOn=false;
int targetRow,targetCol;

void newTarget(){
  targetRow=random(0,3);
  targetCol=random(0,3);
  clearAll();
  digitalWrite(ledPins[targetRow][targetCol], HIGH);
  targetTime=millis();
  targetOn=true;
}

void setup(){
  Serial.begin(9600);
  randomSeed(analogRead(A5));
  newTarget();
}

void loop(){
  if(targetOn){
    if(digitalRead(BTN_OK)==LOW){
      unsigned long reaction=millis()-targetTime;
      Serial.print("Reaction time: ");
      Serial.println(reaction);
      delay(1000);
      newTarget();
    }
    if(millis()-targetTime>3000){
      Serial.println("Too slow!");
      delay(1000);
      newTarget();
    }
  }
}
```

📝**Student Tasks:** - Add **WIN** and **LOSE** patterns.
- Add **score counter**.
- Reduce allowed time to increase difficulty.

🙊Wrap-up

You have: - Simulated **temperature sensing**
- Built a **LED matrix**
- Used **push buttons as a joystick**
- Designed a **simple reaction game**

These concepts directly map to **Raspberry Pi + Sense HAT** in real hardware.
