BABY TILT MONITOR

The Baby Tilt Monitor is a simple and effective system designed to detect the tilt of a baby's crib or body movement and provide an alert through an LED . The tilt sensor detects changes in position, and when it is activated, the LED blinks to indicate a tilt event.

- COMPONENTS USED:
- **Arduino Board**
- **Tilt Sensor**
- **LED**
- **Resistors (** required to limit current through the LED.)
- **Connecting Wires**
- **Power Source**

MECHANISM:

- The **tilt sensor** is connected to **digital pin 2** of the Arduino. It acts as an input device.
- The **LED** is connected to **digital pin 3** and serves as an output device.
- When the tilt sensor detects movement, it sends a HIGH signal to the Arduino, which activates the LED.
- The LED blinks for a short duration to indicate a tilt event.
- The **tilt sensor** continuously monitors for changes in orientation.
- When a tilt is detected (sensor output is HIGH), the **Arduino reads the input** and triggers the LED.
- The LED turns ON for **1 second** and then turns OFF for **0.5 seconds**, creating a blinking effect.
- If no tilt is detected, the LED remains OFF.
- APPLICATIONS:
- **Baby Safety:** Helps monitor baby movement and prevents falls by alerting parents if the crib or baby tilts unexpectedly.
- **Elderly Care:** Can be used to detect sudden movements of elderly people and provide alerts.
- **Industrial Applications:** Can be adapted for monitoring equipment tilt in various industries.

CODE:

```
const int tiltPin = 2;   // Tilt sensor on Pin 2
const int ledPin = 3;    // LED on Pin 3
void setup() {
  pinMode(tiltPin, INPUT);   // Setting tilt sensor as input
  pinMode(ledPin, OUTPUT);   // Setting LED as output
}
void loop()
{
  if (digitalRead(tiltPin) )
  {   // If tilted
    digitalWrite(ledPin, HIGH);  // Turn led on
    delay(1000);
    digitalWrite(ledPin, LOW);//led off,for blink effect
    delay(500);
  }
  else
  {
    digitalWrite(ledPin, LOW);   // Otherwise,led off
  }
}
```

STEPS COUNTER

The Step Counter is a system designed to track steps using a combination of a force sensor and a tilt sensor. This system detects foot pressure through the force sensor and movement using the tilt sensor for step detection.

Components Required:

1. **Arduino Board**
2. **Force Sensor (FSR)**
3. **Tilt Sensor**
4. **Connecting Wires**
5. **Power Source**

Circuit Design

- The **force sensor** is connected to **analog pin A0** to measure applied pressure.
- The **tilt sensor** is connected to **digital pin 2**, acting as an input to detect motion.
- When both conditions (force and tilt) are met, a step is counted and displayed on the Serial Monitor

Code

```
const int forcePin = A0;     // Force sensor on A0

const int tiltPin = 2;       // Tilt sensor on Pin 2


int stepCount = 0;           // Stores total steps

bool stepDetected = false;   // Prevents double-counting


void setup() {

  pinMode(tiltPin, INPUT);

  Serial.begin(9600);

  Serial.println("Step Counter Ready!");

}
```

```
void loop() {
  int forceValue = analogRead(forcePin);
  int tiltState = digitalRead(tiltPin);


  Serial.print("Force: "); Serial.print(forceValue);
  Serial.print(" | Tilt: "); Serial.println(tiltState);


  if (forceValue > 300 && tiltState == HIGH) { // Lower force threshold
    stepCount++;
    Serial.print("STEP! Total: "); Serial.println(stepCount);
    delay(2000);
  }
}
```

Mechanism

1. The **force sensor** continuously measures foot pressure.
2. The **tilt sensor** detects movement to differentiate actual steps from stationary pressure.
3. If both conditions (sufficient force and tilt activation) are met, the **Arduino registers a step**.
4. The **step count** is displayed on the Serial Monitor.
5. A **2-second delay** prevents multiple counts from a single step.

Applications and Benefits

- **Fitness Tracking:** Can be used as a basic pedometer for monitoring daily step counts.
- **Rehabilitation Monitoring:** Helps in tracking movement patterns for physiotherapy patients.
- **Smart Shoes:** Can be embedded into footwear to monitor walking behavior.

WEATHER INDICATOR

The Weather Indicator is a system designed to classify weather conditions using a **temperature sensor (TMP36)** and a **photoresistor (LDR)**. Based on the measured temperature and light levels, the system determines whether the weather is **sunny, cloudy, or normal** and provides an indication using **LEDs**.

Components Required:

1. **Arduino Board**
2. **Temperature Sensor (TMP36)**
3. **Photoresistor (LDR)** – Measures ambient light intensity.
4. **red LED** – Indicates sunny weather.
5. **Blue LED** – Indicates cloudy weather.
6. **Connecting Wires**
7. **Power Source**

- The **temperature sensor (TMP36)** is connected to **analog pin A0**, providing an analog voltage corresponding to temperature.
- The **photoresistor (LDR)** is connected to **analog pin A1**, giving a resistance-dependent voltage based on light intensity.
- The **red LED** (pin 3) lights up for **sunny** weather, and the **blue LED** (pin 4) lights up for **cloudy** weather.
- The Arduino continuously reads temperature and light levels, then classifies the weather condition accordingly.

Code

```
const int tempPin = A0;      // Temperature sensor on A0

const int lightPin = A1;    // Photoresistor on A1

const int greenLED = 3;      // Green LED on Pin 3

const int blueLED = 4;      // Blue LED on Pin 4
```

```
void setup() {
  pinMode(greenLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  Serial.begin(9600);       // Start Serial Monitor
}
void loop() {
  // Read temperature (convert to Celsius)
  float tempVoltage = analogRead(tempPin) * (5.0 / 1023.0);
  float tempC = (tempVoltage - 0.5) * 100; // TMP36 formula
  // Read light level (0-1023)
  int lightLevel = analogRead(lightPin);
  // Classify weather
  if (tempC > 25 && lightLevel > 500) {      // Hot & Sunny
    digitalWrite(greenLED, HIGH);
    digitalWrite(blueLED, LOW);
    Serial.println("Weather: Sunny");
  }
  else if (tempC < 15 && lightLevel < 300) {  // Cold & Cloudy
    digitalWrite(greenLED, LOW);
    digitalWrite(blueLED, HIGH);
    Serial.println("Weather: Cloudy");
  }
  else {                                   // Neutral
    digitalWrite(greenLED, LOW);
    digitalWrite(blueLED, LOW);
    Serial.println("Weather: Normal");
  }
delay(2000); // Update every 2 seconds}
```

Mechanism

1. The **TMP36 sensor** reads ambient temperature and converts it into Celsius.
2. The **photoresistor (LDR)** detects light intensity, producing a value between **0 and 1023**.
3. The system determines the weather condition:
    - If **temperature > 25°C** and **light level > 500**, it classifies as **Sunny**, turning on the green LED.
    - If **temperature < 15°C** and **light level < 300**, it classifies as **Cloudy**, turning on the blue LED.
    - Otherwise, it classifies as **Normal**, turning off both LEDs.
4. The result is displayed and updated every **2 seconds**.

Applications

- **Basic Weather Monitoring:** Provides a simple way to classify weather conditions.
- **Smart Home Systems:** Can be integrated with home automation for automatic lighting and temperature control.
- **Agricultural Use:** Helps in monitoring weather conditions for farming applications.