1. DESIGN A SIMPLE EMBEDDED SYSTEM USING A MICROCONTROLLER TO COUNT AND DISPLAY THE NUMBER OF TIMES A PUSH BUTTON IS PRESSED. THIS INVOLVES DEBOUNCING THE BUTTON TO ENSURE ACCURATE COUNTING AND DISPLAYING THE COUNT ON AN LCD OR A SERIAL MONITOR.

ANS: Here’s a design of a **simple embedded system** using a **microcontroller (e.g., Arduino Uno)** to **count button presses**, handle **debouncing**, and **display the count on an LCD**

**✅ SYSTEM OVERVIEW**

**Objective:**  
Count the number of times a push button is pressed, with debouncing, and display the count on:

* **Option 1:** 16x2 LCD
* **Option 2:** Serial Monitor (if LCD not available)

**🔧 COMPONENTS REQUIRED**

1. Arduino Uno (or compatible microcontroller)
2. Push Button
3. 10kΩ Resistor (pull-down or pull-up)
4. 16x2 LCD with I2C module (optional)
5. Breadboard and jumper wires
6. USB Cable (to connect Arduino)
7. **Button Circuit:**

+5V

|

|

[Button]

|

+--------> Digital Pin 2 (Interrupt capable)

|

[10kΩ Resistor]

|

GND

**LCD Connections (I2C):**

LCD SDA --> Arduino A4

LCD SCL --> Arduino A5

LCD VCC --> 5V

LCD GND --> GND

**OFTWARE (Arduino Code)**

**✅ WITH LCD (I2C):**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2); // LCD address might differ (0x27 or 0x3F)

const int buttonPin = 2;

volatile int count = 0;

unsigned long lastDebounceTime = 0;

unsigned long debounceDelay = 50;

void setup() {

lcd.begin();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("Count: 0");

pinMode(buttonPin, INPUT);

attachInterrupt(digitalPinToInterrupt(buttonPin), buttonPressed, RISING);

}

void loop() {

// Nothing here, logic handled in interrupt

}

void buttonPressed() {

unsigned long currentTime = millis();

if (currentTime - lastDebounceTime > debounceDelay) {

count++;

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Count: ");

lcd.print(count);

lastDebounceTime = currentTime;

}

}

**WITHOUT LCD (Using Serial Monitor):**

**const int buttonPin = 2;**

**volatile int count = 0;**

**unsigned long lastDebounceTime = 0;**

**unsigned long debounceDelay = 50;**

**void setup() {**

**Serial.begin(9600);**

**pinMode(buttonPin, INPUT);**

**attachInterrupt(digitalPinToInterrupt(buttonPin), buttonPressed, RISING);**

**}**

**void loop() {**

**// Display updated count**

**}**

**void buttonPressed() {**

**unsigned long currentTime = millis();**

**if (currentTime - lastDebounceTime > debounceDelay) {**

**count++;**

**Serial.print("Button Pressed Count: ");**

**Serial.println(count);**

**lastDebounceTime = currentTime;**

**}**

**}**

**EXPLANATION**

**🔹 Debouncing:**

**Mechanical buttons can bounce, generating multiple triggers.  
We use millis() to ensure 50ms passes between valid button presses.**

**🔹 Interrupt:**

**Using attachInterrupt() for accurate press detection, even if the main loop is idle.**

**🔹 Display:**

**Use either:**

* **LCD (for standalone projects)**
* **Serial Monitor (for debugging or simple display)**

**📊 RESULT**

**When the push button is pressed:**

* **The counter increments once per press (debounced).**
* **The current count is displayed on the LCD or serial monitor.**