### **1º Objetivo: Tabela da Verdade para 1 Sensor**

#### **Escolha do Sensor**

**Sensor Selecionado:** SW-420 (Sensor de Vibração)

#### **Definição das Entradas e Saídas**

* **Entradas:**
  + **Estado do Sensor de Vibração (DO):**
    - 0: Inativo (sem vibração)
    - 1: Ativo (vibração detectada)
* **Saídas:**
  + **Alerta de Vibração:**
    - 0: Sem alerta
    - 1: Alerta de vibração detectada

#### **Criação da Tabela da Verdade**

|  |  |
| --- | --- |
| **Estado do Sensor (DO)** | **Alerta de Vibração** |
| 0 (Inativo) | 0 (Sem Alerta) |
| 1 (Ativo) | 1 (Alerta) |

#### **Justificativa dos Resultados Esperados**

* **Quando o sensor está inativo (DO = 0):** Não há vibração detectada, portanto, o sistema não deve gerar nenhum alerta (Alerta = 0).
* **Quando o sensor está ativo (DO = 1):** O sensor detecta vibrações, e o sistema deve gerar um alerta para indicar que uma vibração foi detectada (Alerta = 1).

### **2º Objetivo: Tabela da Verdade para Todos os Sensores**

#### **Sensores Considerados**

* **Sensor 1:** SW-420 (Sensor de Vibração)
* **Sensor 2:** MPU-6050 (Acelerômetro)

#### **Definição das Entradas e Saídas Combinadas**

* **Entradas:**
  + **Estado do Sensor de Vibração (SW-420):**
    - 0: Inativo
    - 1: Ativo
  + **Estado do Acelerômetro (MPU-6050):**
    - 0: Sem impacto (acelerações normais)
    - 1: Impacto detectado (acelerações bruscas)
* **Saídas:**
  + **Alerta de Vibração:**
    - 0: Sem alerta
    - 1: Alerta de vibração detectada
  + **Alerta de Impacto:**
    - 0: Sem alerta
    - 1: Alerta de impacto detectado
  + **Ação do Sistema (Envio de Dados):**
    - 0: Não enviar dados
    - 1: Enviar dados

#### **Criação da Tabela da Verdade**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estado do SW-420** | **Estado do MPU-6050** | **Alerta de Vibração** | **Alerta de Impacto** | **Ação do Sistema** |
| 0 (Inativo) | 0 (Sem impacto) | 0 (Sem Alerta) | 0 (Sem Alerta) | 0 (Não enviar) |
| 0 (Inativo) | 1 (Impacto) | 0 (Sem Alerta) | 1 (Alerta) | 1 (Enviar) |
| 1 (Ativo) | 0 (Sem impacto) | 1 (Alerta) | 0 (Sem Alerta) | 1 (Enviar) |
| 1 (Ativo) | 1 (Impacto) | 1 (Alerta) | 1 (Alerta) | 1 (Enviar) |

#### **Justificativa dos Resultados Esperados**

* **Caso 1:** (SW-420 = 0, MPU-6050 = 0) - Nenhum alerta, pois não há vibração nem impacto. O sistema não deve enviar dados.
* **Caso 2:** (SW-420 = 0, MPU-6050 = 1) - Apenas o alerta de impacto é ativado. O sistema deve enviar dados.
* **Caso 3:** (SW-420 = 1, MPU-6050 = 0) - O alerta de vibração é ativado. O sistema deve enviar dados.
* **Caso 4:** (SW-420 = 1, MPU-6050 = 1) - Ambos os alertas são ativados. O sistema deve enviar dados.

### **3º Objetivo: Programação da Acurácia (Usando Probabilidade) para um Sensor**

#### **Definindo o Conceito de Acurácia**

A acurácia é a proporção de leituras corretas de um sensor em relação ao valor verdadeiro esperado. Isso significa que a acurácia mede a precisão das leituras do sensor em comparação com os valores reais ou esperados.

#### **Testando o Sensor Individualmente**

**Sensor Escolhido:** SW-420 (Sensor de Vibração)

**Passos para Testar o Sensor:**

1. **Configuração do Ambiente de Teste:**
   1. Crie um ambiente onde você pode gerar vibrações conhecidas (por exemplo, batendo levemente em uma mesa).
   2. Registre as condições reais de vibração, como a intensidade e a duração.
2. **Programação do Sistema:**
   1. Programe o sistema para capturar dados do sensor SW-420 e compare as leituras com os valores de referência.

**Código Exemplo:**

cpp

VerifyOpen In EditorEditCopy code

1const int vibrationPin = 2; // pino do sensor de vibração  
2int vibrationState = 0;  
3int correctReadings = 0;  
4int totalReadings = 0;  
5  
6void setup() {  
7 Serial.begin(115200);  
8 pinMode(vibrationPin, INPUT);  
9}  
10  
11void loop() {  
12 // Lê o estado do sensor de vibração  
13 vibrationState = digitalRead(vibrationPin);  
14   
15 // Condição de referência (exemplo: vibração esperada)  
16 int expectedState = /\* valor esperado (0 ou 1) \*/;  
17   
18 // Compara a leitura com o valor esperado  
19 if (vibrationState == expectedState) {  
20 correctReadings++;  
21 }  
22 totalReadings++;  
23  
24 // Cálculo da acurácia  
25 float accuracy = (float)correctReadings / totalReadings;  
26  
27 // Envia os dados para o monitor serial  
28 Serial.print("Vibration State: ");  
29 Serial.println(vibrationState);  
30 Serial.print("Correct Readings: ");  
31 Serial.println(correctReadings);  
32 Serial.print("Total Readings: ");  
33 Serial.println(totalReadings);  
34 Serial.print("Accuracy: ");  
35 Serial.println(accuracy);  
36   
37 delay(500); // Ajuste conforme necessário  
38}

#### **Cálculo da Acurácia**

A acurácia pode ser calculada usando a fórmula:

[ \text{Acurácia} = \frac{\text{número de leituras corretas}}{\text{número total de leituras}} ]

**Documentação dos Resultados:**

* Registre o número total de leituras e quantas dessas leituras foram corretas durante o teste.
* Documente como o sensor performa em condições reais, incluindo qualquer discrepância observada entre as leituras do sensor e os valores esperados.

### **4º Objetivo: Programação da Acurácia (Usando Probabilidade) para Todos os Sensores**

#### **Testando os Sensores Simultaneamente**

**Sensores Considerados:** SW-420 (Sensor de Vibração) e MPU-6050 (Acelerômetro)

**Passos para Testar os Sensores:**

1. **Configuração do Ambiente de Teste:**
   1. Crie um ambiente onde você pode gerar vibrações conhecidas e impactos conhecidos.
   2. Registre as condições reais, como a intensidade da vibração e a aceleração.
2. **Programação do Sistema:**
   1. Programe o sistema para coletar leituras de ambos os sensores e compare as leituras com os valores de referência.

**Código Exemplo:**

cpp

VerifyOpen In EditorEditCopy code

1#include <Wire.h>  
2#include <MPU6050.h>  
3  
4MPU6050 mpu;  
5const int vibrationPin = 2; // pino do sensor de vibração  
6int vibrationState = 0;  
7int correctVibrationReadings = 0;  
8int totalVibrationReadings = 0;  
9int correctImpactReadings = 0;  
10int totalImpactReadings = 0;  
11  
12void setup() {  
13 Serial.begin(115200);  
14 Wire.begin();  
15 mpu.initialize();  
16 pinMode(vibrationPin, INPUT);  
17}  
18  
19void loop() {  
20 // Lê o estado do sensor de vibração  
21 vibrationState = digitalRead(vibrationPin);  
22   
23 // Condição de referência para vibração  
24 int expectedVibrationState = /\* valor esperado (0 ou 1) \*/;  
25   
26 // Compara a leitura de vibração  
27 if (vibrationState == expectedVibrationState) {  
28 correctVibrationReadings++;  
29 }  
30 totalVibrationReadings++;  
31  
32 // Lê os dados do MPU-6050  
33 int16\_t ax, ay, az;  
34 mpu.getAcceleration(&ax, &ay, &az);  
35   
36 // Condição de referência para impacto  
37 int expectedImpactState = /\* valor esperado (0 ou 1) \*/;  
38   
39 // Compara a leitura de impacto  
40 if (/\* condição para detectar impacto com base em ax, ay, az \*/) {  
41 correctImpactReadings++;  
42 }  
43 totalImpactReadings++;  
44  
45 // Cálculo da acurácia para vibração  
46 float vibrationAccuracy = (float)correctVibrationReadings / totalVibrationReadings;  
47  
48 // Cálculo da acurácia para impacto  
49 float impactAccuracy = (float)correctImpactReadings / totalImpactReadings;  
50  
51 // Envia os dados para o monitor serial  
52 Serial.print("Vibration State: ");  
53 Serial.println(vibrationState);  
54 Serial.print("Correct Vibration Readings: ");  
55 Serial.println(correctVibrationReadings);  
56 Serial.print("Total Vibration Readings: ");  
57 Serial.println(totalVibrationReadings);  
58 Serial.print("Vibration Accuracy: ");  
59 Serial.println(vibrationAccuracy);  
60  
61 Serial.print("Impact Accuracy: ");  
62 Serial.println(impactAccuracy);  
63   
64 delay(500); // Ajuste conforme necessário  
65}

#### **Cálculo da Acurácia Combinada**

Para cada sensor, determine a frequência de leituras corretas e registre a acurácia de cada sensor separadamente. Se necessário, calcule a acurácia combinada com base em como os dados afetam a resposta do sistema.

**Documentação dos Resultados:**

* Apresente os cálculos e conclusões, destacando a confiabilidade do sistema em cenários reais.
* Justifique as probabilidades e situações testadas, documentando como cada sensor se comportou em diferentes condições operacionais. ### 1º Objetivo: Tabela da Verdade para 1 Sensor

#### **Escolha do Sensor**

**Sensor Selecionado:** MPU-6050 (Acelerômetro)

#### **Definição das Entradas e Saídas**

* **Entradas:**
  + **Estado do Acelerômetro (DO):**
    - 0: Sem movimento
    - 1: Movimento detectado
* **Saídas:**
  + **Alerta de Movimento:**
    - 0: Sem alerta
    - 1: Alerta de movimento detectado

#### **Criação da Tabela da Verdade**

|  |  |
| --- | --- |
| **Estado do Acelerômetro (DO)** | **Alerta de Movimento** |
| 0 (Sem movimento) | 0 (Sem Alerta) |
| 1 (Movimento) | 1 (Alerta) |

#### **Justificativa dos Resultados Esperados**

* **Quando o acelerômetro está inativo (DO = 0):** Não há movimento detectado, portanto, o sistema não deve gerar nenhum alerta (Alerta = 0).
* **Quando o acelerômetro está ativo (DO = 1):** O sensor detecta movimento, e o sistema deve gerar um alerta para indicar que um movimento foi detectado (Alerta = 1).