Detector de ocupação máxima

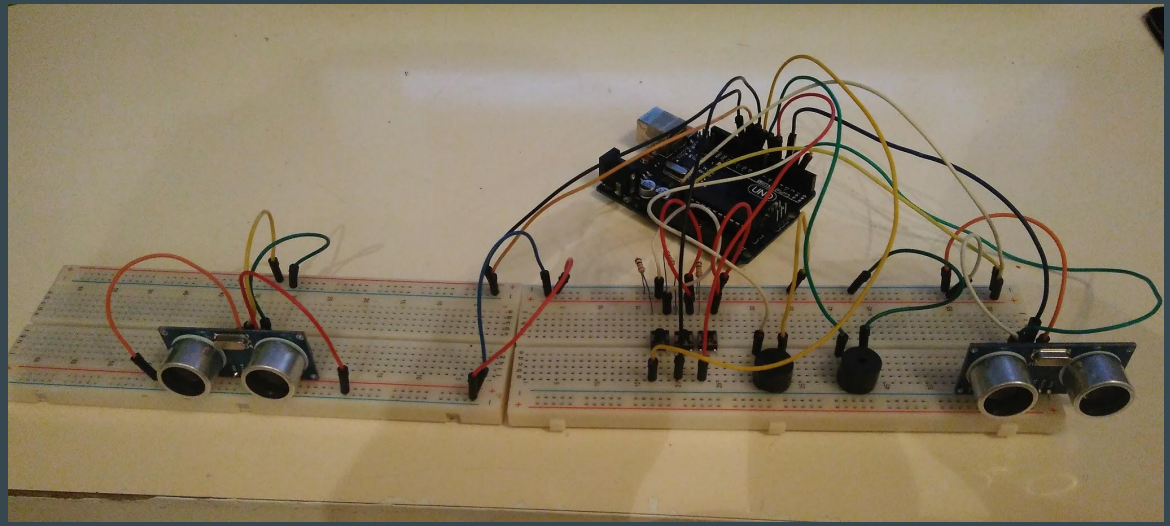
...

Propósito

- Controle de pessoas numa sala
- Permite configuração do número máximo
- Acrescenta contador quando alguém entra
- Decrementa quando sai

Hardware

- 3 botões
- 2 sensores de distância
- 2 beepers
- Não pude usar a LCD



Código

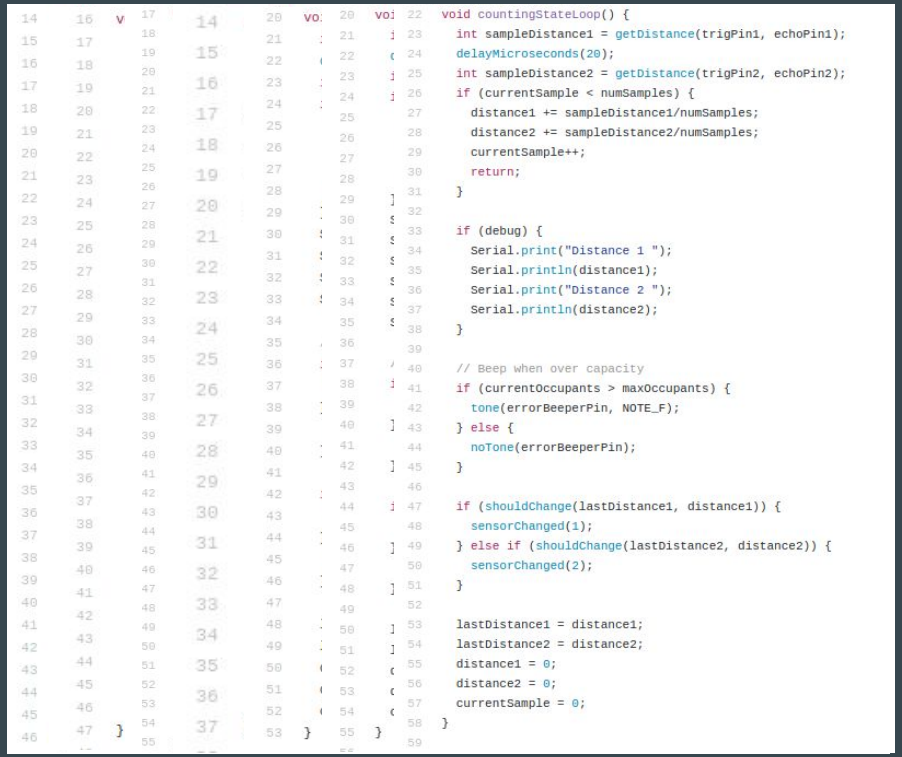
- Reage a eventos
- Máquina de estados

```
void initialStateButtonChanged(int pin, int value) {
  if (value == HIGH) {
    switch (pin) {
      case addButton:
        tone(beeperPin, NOTE_A);
        maxOccupants++;
        printMaxOccupants();
        break;
      case subButton:
        if (maxOccupants > 0) {
```

```
40  void loop_() {
41    switch (currentState) {
42      case initial:
43        initialStateLoop();
44        break;
45      case counting:
46        countingStateLoop();
47        break;
48    }
49  }
50
51  void buttonChanged(int pin, int value) {
52    switch (currentState) {
53      case initial:
54        initialStateButtonChanged(pin, value);
55        break;
56      case counting:
57        countingStateButtonChanged(pin, value);
58        break;
59    }
60  }
```

Dificuldades

- Distâncias diferentes
- Sensores entram em conflito
- Edge cases
- Trial and error



```
void countingStateLoop() {
  int sampleDistance1 = getDistance(trigPin1, echoPin1);
  delayMicroseconds(20);
  int sampleDistance2 = getDistance(trigPin2, echoPin2);
  if (currentSample < numSamples) {
    distance1 += sampleDistance1/numSamples;
    distance2 += sampleDistance2/numSamples;
    currentSample++;
    return;
  }

  if (debug) {
    Serial.print("Distance 1 ");
    Serial.println(distance1);
    Serial.print("Distance 2 ");
    Serial.println(distance2);
  }

  // Beep when over capacity
  if (currentOccupants > maxOccupants) {
    tone(errorBeeperPin, NOTE_F);
  } else {
    noTone(errorBeeperPin);
  }

  if (shouldChange(lastDistance1, distance1)) {
    sensorChanged(1);
  } else if (shouldChange(lastDistance2, distance2)) {
    sensorChanged(2);
  }

  lastDistance1 = distance1;
  lastDistance2 = distance2;
  distance1 = 0;
  distance2 = 0;
  currentSample = 0;
}
```

Soluções

- Solução mais simples foi a melhor
- Usa a média de samples consecutivos
- Reseta os sensors após um intervalo
- Permite configuração de distância mínima