

```

/*
AZ Mandible · Prototype Firmware v1.0
-----
Hardware:
- Piezo (27mm) · 1M bias · Op-amp · A0
- Bone conduction transducer (TT25-8) via PAM8403 · D9 (PWM)
- Arduino Nano

Function:
- Detects intentional jaw movement peaks
- Simple classification:
1 clear peak · YES
2 quick peaks · NO
- Private bone-conduction feedback
*/

const int PIEZO_PIN = A0;
const int FEEDBACK_PIN = 9; // PWM to PAM8403 / transducer

// ===== TUNABLE VALUES =====
int threshold = 80; // Start here, adjust with Serial Plotter
const int DEBOUNCE_MS = 80;
const int WINDOW_MS = 600; // Time window to count peaks
const int MIN_PEAK_GAP = 120; // Minimum time between peaks

// ===== STATE =====
int peakCount = 0;
unsigned long lastPeakTime = 0;
unsigned long windowStart = 0;
bool inWindow = false;

void setup() {
  Serial.begin(115200);
  pinMode(FEEDBACK_PIN, OUTPUT);
  analogWrite(FEEDBACK_PIN, 0);

  Serial.println("AZ Mandible v1.0 ready");
  Serial.println("1 peak = YES | 2 peaks = NO");
  Serial.println("Open Serial Plotter to tune threshold");
}

void loop() {
  int value = analogRead(PIEZO_PIN);

  // Debug output for Serial Plotter
  Serial.println(value);

  unsigned long now = millis();

```

```

// Peak detection
if (value > threshold) {
  if (now - lastPeakTime > MIN_PEAK_GAP) {
    // Valid new peak
    if (!inWindow) {
      inWindow = true;
      windowStart = now;
      peakCount = 1;
    } else {
      peakCount++;
    }
    lastPeakTime = now;
  }
}

// Window timeout · classify
if (inWindow && (now - windowStart > WINDOW_MS)) {
  classify(peakCount);
  peakCount = 0;
  inWindow = false;
}

```

```

delay(5); // small loop delay
}

```

```

void classify(int peaks) {
  if (peaks == 1) {
    Serial.println("· YES");
    playFeedback(1); // single tone
  }
  else if (peaks >= 2) {
    Serial.println("· NO");
    playFeedback(2); // double tone
  }
}

```

```

void playFeedback(int type) {
  if (type == 1) {
    // YES · single medium beep
    tone(FEEDBACK_PIN, 600, 120);
    delay(140);
  }
  else if (type == 2) {
    // NO · two short beeps
    tone(FEEDBACK_PIN, 400, 80);
    delay(120);
    tone(FEEDBACK_PIN, 400, 80);
    delay(100);
  }
}

```

```
analogWrite(FEEDBACK_PIN, 0);  
}
```