

```

// Button pin assignments
const int yellowButtonPin = 3;
const int blueButtonPin   = 4;
const int whiteButtonPin  = 5;
const int blackButtonPin  = 6;
const int greenButtonPin  = 7;

// Rotary encoder pins
const int CLK_PIN = 8;
const int DI_PIN  = 9;
const int SW_PIN  = 10;

// Slide potentiometer pin
const int slidePotPin = A0;

// Colour pairs for button test
const String colorPairs[][2] = {
    {"YELLOW", "BLUE"},
    {"BLACK",  "GREEN"},
    {"WHITE",  "BLUE"},
    {"WHITE",  "BLACK"},
    {"GREEN",  "BLUE"},
    {"GREEN",  "BLACK"},
    {"YELLOW", "WHITE"},
};

// Game state variables
float score = 0.0;
float encoderPosition = 0;
float lastCLKState    = LOW;
float targetNumber     = 0;
bool  gameStarted     = false;
bool  gameDone        = false;

unsigned long startTime;
unsigned long endTime;
unsigned long startTime2;
bool slideDone;

void setup() {
    // Configure button inputs
    pinMode(yellowButtonPin, INPUT);
    pinMode(blueButtonPin,   INPUT);
    pinMode(whiteButtonPin,  INPUT);
    pinMode(blackButtonPin,  INPUT);

```

```

pinMode(greenButtonPin, INPUT);

// Enable encoder
pinMode(CLK_PIN, INPUT_PULLUP);
pinMode(DI_PIN, INPUT_PULLUP);
pinMode(SW_PIN, INPUT_PULLUP);

Serial.begin(9600);
randomSeed(analogRead(0));
}

void loop() {
  // Wait for start command over Serial
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n');
    if (input == "start") {
      score = 0.0;
      Serial.println("Test starting! Get ready...");
      delay(2000);

      //Run tests
      testReaction();      Serial.println("----"); delay(1000);
      buttonAndSliderTest();Serial.println("----"); delay(1000);
      pressAndHoldTest();  Serial.println("----"); delay(1000);
      knobGame2();         Serial.println("----"); delay(1000);
      testReaction();      Serial.println("----"); delay(1000);
      sliderTest2();       Serial.println("----"); delay(1000);
      twoButtonTest();     Serial.println("----"); delay(1000);
      testReaction();      Serial.println("----"); delay(1000);
      sliderTest1();       Serial.println("----"); delay(1000);
      knobGame1();         Serial.println("----"); delay(1000);
      testReaction();      Serial.println("----"); delay(1000);
      buttonAndSliderTest();Serial.println("----"); delay(1000);
      twoButtonTest();     Serial.println("----"); delay(1000);

      // Print final score
      Serial.print("FINAL SCORE: ");
      Serial.print(score);
      Serial.print(" / 19.5  (");
      Serial.print(score / 19.5);
      Serial.println(")");
    }
  }
}

```

```

//one button reaction test
enum { REACTION_POINTS = 1 };
void testReaction() {
    delay(random(1000, 3000));
    int colorIndex = random(0, 5);
    String targetColor =
String[]{"YELLOW", "BLUE", "WHITE", "BLACK", "GREEN"}[colorIndex];

    Serial.print("Press the ");
    Serial.print(targetColor);
    Serial.println(" button NOW!");

    startTime = micros(); // Start timing
    // Wait for correct button
    while (true) {
        if ( (digitalRead(yellowButtonPin) && targetColor=="YELLOW") ||
            (digitalRead(blueButtonPin) && targetColor=="BLUE") ||
            (digitalRead(whiteButtonPin) && targetColor=="WHITE") ||
            (digitalRead(blackButtonPin) && targetColor=="BLACK") ||
            (digitalRead(greenButtonPin) && targetColor=="GREEN") ) break;
    }

    float timeSec = (micros() - startTime) / 1e6;
    score += (timeSec <= 1.0) ? 1.0 : (1.0 / timeSec);
    Serial.print("Reaction Time: "); Serial.print(timeSec); Serial.println(" s");
    Serial.print("Score: "); Serial.println(score);
    delay(300);
}

// Press and hold duration accuracy test
void pressAndHoldTest() {
    int colorIndex = random(0, 5);
    String targetColor =
String[]{"YELLOW", "BLUE", "WHITE", "BLACK", "GREEN"}[colorIndex];
    unsigned long required = random(2000, 7000);

    Serial.print("Hold "); Serial.print(targetColor);
    Serial.print(" for "); Serial.print(required/1000.0);
    Serial.println(" seconds");

    // Wait for press
    while (digitalRead(yellowButtonPin)==LOW && digitalRead(blueButtonPin)==LOW &&
        digitalRead(whiteButtonPin)==LOW && digitalRead(blackButtonPin)==LOW &&
        digitalRead(greenButtonPin)==LOW);
    startTime = millis();

```

```

// Wait for release
while (digitalRead(yellowButtonPin) || digitalRead(blueButtonPin) ||
       digitalRead(whiteButtonPin) || digitalRead(blackButtonPin) ||
       digitalRead(greenButtonPin));
endTime = millis();

float held = (endTime - startTime)/1000.0;
float error = abs(held - (required/1000.0));
float accuracy = (error <= 0.5) ? 1.0 : (error<=1.0 ? 0.7 : 0.1);
score += accuracy;

Serial.print("Held for: "); Serial.print(held); Serial.println(" s");
Serial.print("Score: "); Serial.println(score);
delay(300);
}

// Match encoder number game (knob1)
void knobGame1() {
  Serial.println("Knob1: press switch to start, match number");
  bool started=false;
  float points=1;
  while (true) {
    // Read encoder
    if (digitalRead(CLK_PIN)==HIGH && lastCLKState==LOW) {
      encoderPosition += (digitalRead(DI_PIN)!=digitalRead(CLK_PIN)) ? 1 : -1;
      encoderPosition = constrain(encoderPosition, -200, 200);
      if (started) Serial.println(encoderPosition);
    }
    lastCLKState = digitalRead(CLK_PIN);

    // On switch press
    if (digitalRead(SW_PIN)==LOW) {
      if (!started) {
        targetNumber = random(-200, 201);
        Serial.print("Find the number: "); Serial.println(targetNumber);
        started=true;
      } else {
        if (encoderPosition==targetNumber) {
          Serial.println("Got it!"); score+=points; break;
        } else {
          Serial.println((encoderPosition>targetNumber?"Too high":"Too low");
          points /= 2;
        }
      }
    }
  }
}

```

```

        delay(500);
    }
}
Serial.print("Score: "); Serial.println(score);
delay(300);
}

//Second knob game uses different pins (knob2)
void knobGame2() {
    const int clk2=11, di2=12, sw2=13;
    Serial.println("Knob2: press switch to start, match number");
    bool started=false;
    float points=1;
    while (true) {
        if (digitalRead(clk2)==HIGH && lastCLKState==LOW) {
            encoderPosition += (digitalRead(di2)!=digitalRead(clk2)) ? 1 : -1;
            encoderPosition = constrain(encoderPosition, -200, 200);
            if (started) Serial.println(encoderPosition);
        }
        lastCLKState = digitalRead(clk2);

        if (digitalRead(sw2)==LOW) {
            if (!started) {
                targetNumber = random(-200, 201);
                Serial.print("Find the number: "); Serial.println(targetNumber);
                started=true;
            } else {
                if (encoderPosition==targetNumber) { score+=points; Serial.println("Got
it!"); break; }
                Serial.println((encoderPosition>targetNumber)? "Too high": "Too low");
                points /= 2;
            }
            delay(500);
        }
    }
}
Serial.print("Score: "); Serial.println(score);
delay(300);
}

// Slider matching test (uses A0)
void sliderTest1() {
    int target = random(0, 300);
    Serial.println("Slider1: match the number after 6s");
    delay(6000);
    startTime = millis();

```

```

while (true) {
    int val = 341 - (analogRead(A0)/3);
    if (val == target) {
        float t = (millis()-startTime)/1000.0;
        score += (t<=3?1:(1/(t-2)));
        Serial.print("Time: "); Serial.println(t);
        Serial.print("Score: "); Serial.println(score);
        break;
    }
    delay(150);
}
delay(300);
}

//Second slider test (uses A1)
void sliderTest2() {
    int target = random(0, 300);
    Serial.println("Slider2: match the number after 6s");
    delay(6000);
    startTime = millis();
    while (true) {
        int val = 341 - (analogRead(A1)/3);
        if (val == target) {
            float t = (millis()-startTime)/1000.0;
            score += (t<=3?1:(1/(t-2)));
            Serial.print("Time: "); Serial.println(t);
            Serial.print("Score: "); Serial.println(score);
            break;
        }
        delay(150);
    }
    delay(300);
}

// Button + slider combination test
void buttonAndSliderTest() {
    bool up = random(0,2);
    int idx = random(0,5);
    String c = String[]{"YELLOW","BLUE","WHITE","BLACK","GREEN"}[idx];
    Serial.print("Press "); Serial.print(c);
    Serial.print(" and slide "); Serial.println(up?"UP":"DOWN");

    startTime = millis();
    while (true) {
        int val = 341 - (analogRead(A0)/3);

```

```
bool btn = digitalRead(idx+3)==HIGH; // pins 3-7
if (btn && abs(val - (up?300:0))<=15) {
    float t = (millis()-startTime)/1000.0;
    score += (t<=2?2:(t<=4?1:(1/(t-2))));
    Serial.print("Time: "); Serial.println(t);
    Serial.print("Score: "); Serial.println(score);
    break;
}
delay(50);
}
}
```