// Define the pin connected to the PIR motion sensor

const int pirPin = 2; // Update this if connected to a different pin

const int ledPin = LED\_BUILTIN; // Built-in LED on MKR board

void setup() {

 // Initialize serial communication for debugging

 Serial.begin(9600);

 while (!Serial); // Wait for Serial to be ready (important on MKR boards)

 // Configure pin modes

 pinMode(pirPin, INPUT);

 pinMode(ledPin, OUTPUT);

 Serial.println("PIR motion sensor initialized. Waiting for motion...");

}

void loop() {

 // Read the state of the PIR sensor

 int motionState = digitalRead(pirPin);

 if (motionState == HIGH) {

 // Motion detected

 Serial.println("⚠️ Motion detected!");

 digitalWrite(ledPin, HIGH); // Turn on LED

 } else {

 // No motion

 digitalWrite(ledPin, LOW); // Turn off LED

 }

 // Small delay to debounce the sensor and limit serial output

 delay(200);

}

Servo myServo; // Create servo object

const int servoPin = 4; // Connect servo signal wire to D4

void setup() {

 Serial.begin(9600);

 while (!Serial); // Wait for Serial on MKR boards

 myServo.attach(servoPin);

 Serial.println("SG90 Servo Ready.");

 Serial.println("Enter an angle between 0 and 180:");

}

void loop() {

 if (Serial.available() > 0) {

 String input = Serial.readStringUntil('\n');

 input.trim(); // Remove any whitespace or newline

 if (input.length() > 0) {

 int angle = input.toInt(); // Convert string to integer

 if (angle >= 0 && angle <= 180) {

 myServo.write(angle); // Move servo to the angle

 Serial.print("Moving servo to ");

 Serial.print(angle);

 Serial.println(" degrees.");

 } else {

 Serial.println("Invalid input. Please enter a number between 0 and 180.");

 }

 }

 }

}

const int trigPin = 2; // Trigger pin

const int echoPin = 3; // Echo pin

void setup() {

 Serial.begin(9600);

 pinMode(trigPin, OUTPUT);

 pinMode(echoPin, INPUT);

 Serial.println("Ultrasonic Sensor Initialized");

}

void loop() {

 // Clear the trigger pin

 digitalWrite(trigPin, LOW);

 delayMicroseconds(2);

 // Send a 10 µs pulse to trigger the sensor

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);

 // Read the duration of the echo pulse

 long duration = pulseIn(echoPin, HIGH);

 // Calculate the distance in centimeters

 float distance = duration \* 0.034 / 2;

 // Display the distance

 Serial.print("Distance: ");

 Serial.print(distance);

 Serial.println(" cm");

 delay(500); // Wait for half a second before the next reading