```
#include <SD.h> // Load SD library


const int laserTransmitterPin = 3;  // Pin for the laser transmitter
 const int laserReceiverPin = 2;         // Pin for the laser receiver
 const int chipSelect = 4;        // Chip select pin for the SD card module
 const int voltageSourcePin = 5;         // Pin configured as a 5V voltage source


const unsigned long interruptionDuration = 2000; // Minimum time (2 seconds) to count as a bat
visit


unsigned long interruptionStartTime = 0; // Tracks when the interruption started
 bool isLaserInterrupted = false;         // Tracks whether the laser is currently interrupted
 bool batCounted = false;                 // Tracks if the bat visit has been counted


int batCount = 0; // Tracks the number of bats detected
 File file;          // File object for SD card operations


void setup() {
   Serial.begin(9600);


  pinMode(laserTransmitterPin, OUTPUT); // Set laser transmitter as output
   pinMode(laserReceiverPin, INPUT);           // Set laser receiver as input
   pinMode(chipSelect, OUTPUT);      // Chip select pin must be set as output
   pinMode(voltageSourcePin, OUTPUT);   // Configure pin 5 as output for 5V source
   digitalWrite(voltageSourcePin, HIGH); // Set pin 5 HIGH to output 5V


  digitalWrite(laserTransmitterPin, HIGH); // Turn the laser ON


  // Initialize SD card
   if (!SD.begin(chipSelect)) {
```

```
        Serial.println("Could not initialize SD card.");
        while (true); // Halt execution if SD card initialization fails
  }



  // Check if the file already exists and delete it if necessary
   if (SD.exists("bat_log.txt")) {
        Serial.println("File exists. Removing it.");
        if (SD.remove("bat_log.txt")) {
        Serial.println("Successfully removed existing file.");
        } else {
        Serial.println("Could not remove existing file.");
        }
  }



  Serial.println("Bat detection system initialized.");
}



void loop() {
   unsigned long currentTime = millis();
   int sensorState = digitalRead(laserReceiverPin); // Read the state of the laser receiver



  if (sensorState == HIGH) { // Laser is interrupted
        if (!isLaserInterrupted) { // First detection of interruption
        isLaserInterrupted = true;
        batCounted = false; // Reset the bat counted flag for a new interruption
        interruptionStartTime = currentTime; // Record the start time of the interruption
        }
   } else { // Laser is not interrupted
        if (isLaserInterrupted) { // Laser was previously interrupted
        unsigned long interruptionTime = currentTime - interruptionStartTime; // Calculate
interruption time
        if (interruptionTime >= interruptionDuration) {
        // If the interruption lasted long enough to count as a visit
        if (!batCounted) {
        batCount++;
        Serial.print("Bat visit detected! Total visits: ");
        Serial.print(batCount);
        Serial.print(" | Time interrupted: ");
```

```
        Serial.print(interruptionTime);
        Serial.println(" ms");



        logToSD(batCount, interruptionTime); // Log the bat visit count and interruption time to
SD card
        batCounted = true; // Mark this visit as counted
        }
        } else {
        // If the interruption is too short, log as a "tripped" event
        Serial.print("Tripped! Time interrupted: ");
        Serial.print(interruptionTime);
        Serial.println(" ms");
        }
        // Reset the interruption state
        isLaserInterrupted = false;
        }
  }
  delay(100); // Small delay to reduce unnecessary cycles
}



void logToSD(int count, unsigned long interruptionTime) {
  file = SD.open("bat_log.txt", FILE_WRITE); // Open "bat_log.txt" to write data
  if (file) {
        file.print("Bat visit detected. Total visits: ");
        file.print(count);
        file.print(" | Time interrupted: ");
        file.print(interruptionTime);
        file.println(" ms");
        file.close();      // Close the file
        Serial.println("Logged to SD card.");
  } else {
        Serial.println("Could not open file to log data.");
  }
}
```