```
#include<SoftwareSerial.h>
const int pins[4][2] = {{2, 3}, {4, 5}, {6, 7}, {8, 9}};
const int light_pin = 4;
//int led = 12;

float distances[4] = {0, 0, 0, 0};
int num_ultra = 1;
float global_differenc = 2;
int initial_time = 0;
int state = 0;
int fixed_time = 12000;

int Incoming_value, time_gap;
float diff;

SoftwareSerial bluetoothSerial(10, 11); // RX, TX

void setup() {
  // put your setup code here, to run once:
 delault_diatance_initialization();
  initial_time = millis();
 Serial.begin(57600);
 bluetoothSerial.begin(9600);
  for(int i = 0; i < num_ultra; i++) {
    pinMode(pins[i][0], OUTPUT);
    pinMode(pins[i][1], INPUT);
  }
 pinMode(light_pin, OUTPUT);
  //pinMode(led, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(500);
  if(bluetoothSerial.available() > 0)
  {
    Incoming_value = bluetoothSerial.read();      //Read the incoming data and
store it into variable Incoming_value
    Serial.print(Incoming_value);        //Print Value of Incoming_value in Serial
monitor
    Serial.print("\n");                  //New line
    if (Incoming_value == 0) {
      digitalWrite(light_pin, HIGH);
      //digitalWrite(led, HIGH);
       state = 0;
    }
```

```
    else if (Incoming_value == 1) {
      digitalWrite(light_pin, LOW);
      //digitalWrite(led, HIGH);
      Serial.println("Light on");
      initial_time = millis();
       state = 1;
     }
   }
   if (state == 1) {
     /*Serial.print("Time: ");
     diff = millis() - initial_time;
    Serial.println(diff);*/
    ultrasonic_activation();
   }
}

int ultrasonic(int pin) {   // Measurement of distances from ultrasonic sensors
  int trigPin, echoPin;
  float duration, distance;
  trigPin = pins[pin][0];
  echoPin = pins[pin][1];
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  echoPin = pins[pin][1];
  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2;
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
  return distance;
}

int detection() {    // Detection of differences and return yes or no according
threshold
   for (int i = 0; i < num_ultra; i++) {
     diff = ultrasonic(i) - distances[i];
     if (diff < 0) diff = diff*(-1);
    Serial.print("Difference: ");
    Serial.println(diff);
     if (diff > global_differenc) {
       return 1;
     }
   }
```

```
  return 0;
}

void ultrasonic_activation(){   // Activing ultrasonic sensors for active is 1
  int local_time;
    if (detection() == 0) {
      local_time = millis();
      time_gap = local_time - initial_time;
      Serial.print("Time: ");
      Serial.println(time_gap);
      Serial.println();
      if (time_gap > fixed_time) {
        Serial.println("Light turns off");
        digitalWrite(light_pin, HIGH);
         state = 0;
       }
    } else {
      initial_time = millis();
      Serial.println("Time initialised again");
     delault_diatance_initialization();
    }

}

void delault_diatance_initialization() {  // //Initialization the default values for
empty  bed
 Serial.println("Initializing distances:");
  for (int i = 0; i < num_ultra; i++) {
    distances[i] = ultrasonic(i);
    Serial.print("Ultrasonic No-");
    Serial.print(i);
    Serial.print("-: ");
    Serial.println(distances[i]);
  }
  delay(1000);
}
```