Assignment:- 3

Aim:- Design and implement real time monitoring system using android phone (Blynk App.) such as 'Infrared radiations'.

Theory:-

IR sensor:

An **IR sensor** (Infrared sensor) is an electronic device that detects and measures infrared radiation in its surroundings. Infrared radiation is a form of electromagnetic radiation with wavelengths longer than visible light, typically between 0.7 micrometers (μm) and 1000 μm.

Components of an IR Sensor

IR sensors generally consist of the following main components:

1. **Infrared Emitter**: This is usually an LED that emits infrared light. The most common wavelengths for IR LEDs are in the range of 850 nm to 950 nm.
2. **Infrared Detector/Receiver**: This is a photodiode or phototransistor that detects the reflected or emitted infrared light. When infrared light hits the detector, it generates an electrical signal.
3. **Signal Processor**: This converts the signal from the detector into a usable form, often amplified or filtered. In digital IR sensors, a microcontroller may be used to process the signal.
4. **Optional Filters**: Some IR sensors include filters to eliminate light from other sources or to enhance specific wavelengths of infrared light.

Key Parameters of IR Sensors

- **Range**: The distance at which the sensor can detect an object. Active IR sensors typically have ranges from a few centimeters to several meters, while PIR sensors can detect motion from several meters away.
- **Field of View (FoV)**: The angle over which the sensor can detect infrared radiation. PIR sensors, for example, have a wide field of view, which is important for motion detection.
- **Response Time**: The time it takes for the sensor to detect an object and provide an output signal. PIR sensors often have a slight delay as they detect slow changes in IR radiation.
- **Sensitivity**: Defines how small a change in IR radiation the sensor can detect. In PIR sensors, sensitivity may be adjusted to detect larger or smaller movements.

Applications of IR Sensors

1. **Proximity Detection**:
    - Mobile phones use IR sensors to detect if a user is holding the phone near their face during a call, so the screen can turn off.
2. **Motion Sensing**:
    - PIR sensors are used in automatic lighting systems, alarm systems, and security cameras to detect motion in a specific area.
3. **Optical Encoders**:

    IR sensors are used in encoders to measure the position, speed, and direction of rotating devices like motors.

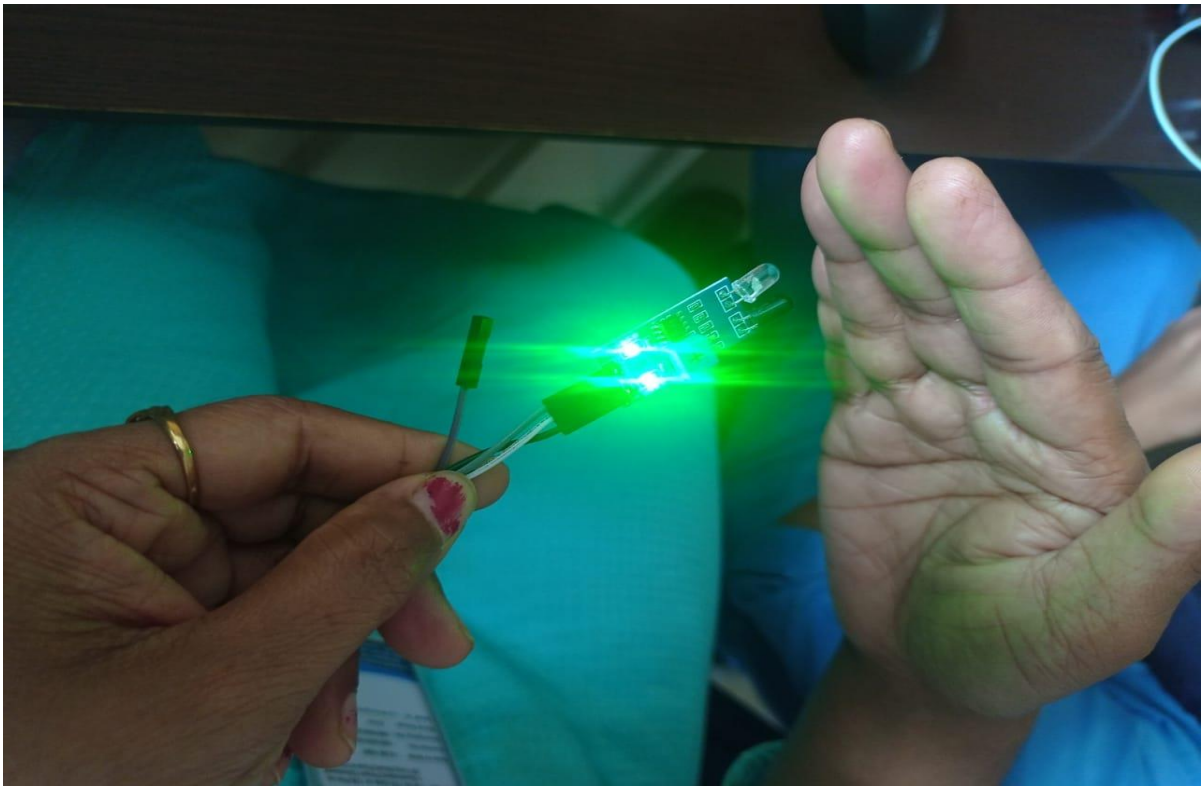
4. **Obstacle Detection**:

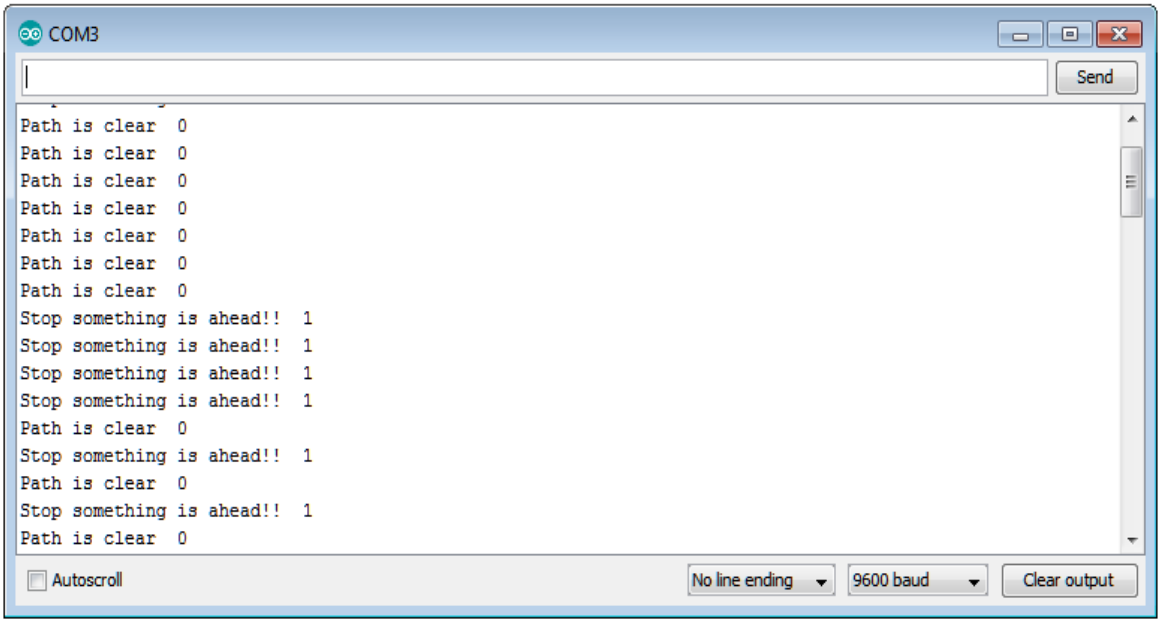
In robotics and autonomous vehicles, IR sensors are employed to detect obstacles and avoid collisions.

Code:-

```
int IRPin=26;
void setup() {
  // put your setup code here, to run once:
pinMode(IRPin,INPUT);
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
digitalRead(IRPin);
delay(500);
}
```




COM3
Send
Path is clear 0
Path is clear 0
Path is clear 0
Path is clear 0
Path is clear 0
Path is clear 0
Path is clear 0
Stop something is ahead!! 1
Stop something is ahead!! 1
Stop something is ahead!! 1
Stop something is ahead!! 1
Path is clear 0
Stop something is ahead!! 1
Path is clear 0
Stop something is ahead!! 1
Path is clear 0
Autoscroll
No line ending
9600 baud
Clear output

Assignment:- 4

Aim:- Design and implement IoT system for one of the applications like: Traffic Application, Medical/Health application, Social Application etc.

Theory:-

IoT-Based Fire Detection and Automated Response System with Motor Activation:-

Fire detection systems are crucial for preventing fire-related accidents and damage. Traditionally, fire detection systems are passive, relying on alarms to alert nearby personnel. With advancements in IoT, fire detection can be integrated into a smart system that not only detects fire early but also responds automatically by activating critical devices like water pumps, sprinkler systems, or exhaust fans (i.e., motors) without human intervention.

This project describes the integration of fire sensors (temperature and smoke), IoT-based communication, and motor control in a real-time, automated fire response system. The system leverages cloud-based platforms for data logging and remote monitoring, and microcontrollers like Arduino or ESP8266/ESP32 for local processing and control.

The system consists of the following main components:

1. **Sensors:**
    - **Temperature Sensor (e.g., DHT11/22 or LM35):** To detect rapid increases in temperature that indicate the presence of fire.
2. **Microcontroller:**
    - A **microcontroller** such as **ESP8266, ESP32, or Arduino** serves as the brain of the system, processing data from the sensors and sending the information to the cloud. It also triggers the motor upon detecting fire.
3. **Motor Driver and Motor:**
    - A **motor driver module** (e.g., L298N or relay module) is used to control the activation of a **motor**, such as a water pump or exhaust fan.
4. **IoT Platform/Cloud Services:**
    - A cloud-based platform (e.g., ThingSpeak, Blynk, or Firebase) is used for remote monitoring and data logging. It allows users to access fire status data via a mobile app or web dashboard.
5. **Power Supply:**

- A regulated power supply ensures that the sensors, microcontroller, and motor are properly powered.

Step-by-Step Process:

1. **System Initialization:**
    - The microcontroller powers up, initializes sensors, and establishes a Wi-Fi connection.
2. **Sensor Data Collection:**
    - The temperature and smoke sensors continuously gather data.
3. **Fire Detection:**
    - If temperature or smoke data crosses the pre-set threshold, the system identifies a fire.
4. **Motor Activation:**
    - Upon detecting a fire, the microcontroller sends a signal to the motor driver to activate the motor (water pump or fan).
5. **Fire Mitigation:**
    - The motor continues running until the fire is extinguished or a manual reset is triggered.

Code:-

```
int relaypin=22;
int IRPin=18;
int IRVal;
void setup() {
  // put your setup code here, to run once:r
  pinMode(IRPin,INPUT);
  pinMode(relaypin,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:

IRVal=digitalRead(IRPin);
if (IRVal==0) {
Serial.println("Fire");
digitalWrite(relaypin,LOW);
delay(500);
}
else{
Serial.println("NoFire");
digitalWrite(relaypin,HIGH);
delay(500);
}
delay(500);
}
```