Day 4 of Technomorph

Dipesh Bhattarai

July 2025

Today, I explored two common sensors used in robotics—IR sensor and ultrasonic sensor—and learned how to connect and program them with an Arduino to detect objects and measure distance and check if there is any obstacle or not.

1 IR Sensor:-

An IR (infrared) sensor works by emitting infrared light and detecting its reflection from nearby objects. It usually has three pins: VCC (power, 5V), GND (ground), and OUT (signal output). The sensor gives a digital output—LOW when it detects an object, and HIGH when nothing is present. Its detection range is short, typically from about 2 to 30 centimeters. IR sensors are widely used in obstacle detection and line-following robots.

1.1 Our Project with IR Sensor:-

We connected the IR sensor's OUT pin to Arduino's digital pin 2 and powered it with 5V and ground. We wrote code that used a for loop to check the sensor multiple times. The program printed "Detected" on the serial monitor when the sensor output was LOW (object present), and "Not Detected" when the output was HIGH (no object). This helped me understand digital signals, sensor reading using digitalRead(), and how to use loops in Arduino programming.

2 Ultrasonic Sensor:-

An ultrasonic sensor like the HC-SR04 measures distance by sending high-frequency sound waves (around 40 kHz) and timing how long it takes for the echo to return. It has four pins: VCC (5V), GND, Trig (to trigger the sound wave), and Echo (to receive the echo). It can measure distances roughly from 2 cm up to 400 cm (4 meters) with an accuracy of about 3 mm. Unlike IR sensors, ultrasonic sensors work well in various lighting conditions and with many surface types.

2.1 Our Project with Ultrasonic Sensor:-

We connected the HC-SR04's VCC to 5V, GND to ground, Trig to digital pin 9, and Echo to digital pin 10 of the Arduino. Our code triggered the sensor with digitalWrite() and measured the echo time with pulseIn(). Then, we calculated the distance using the time and speed of sound, and printed it on the serial monitor every half second inside a for loop. When we placed an object close to the sensor, the distance reading went down; moving it farther away increased the reading. This helped me learn how ultrasonic sensors work, how to control sensors with Arduino pins, and how to use timing functions and loops in code.

Overall, today's projects gave me hands-on experience with different sensor types, circuit connections, and Arduino programming fundamentals, especially working with digital inputs and loops.