VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

Arduino-Based Radar System

Team Members:

BRANCH-AI B

Srikanth (Roll Number: 21911A3598)

Spoorthi (Roll Number: 21911A3599)

Monish (Roll Number: 21911A35A0)

Snehith (Roll Number: 21911A35A1)

Introduction:

The Arduino-based radar system is a project developed by a team of students to create a radar system capable of detecting objects within a specified range. This radar system utilizes an Arduino microcontroller, an ultrasonic sensor, and a servo motor. The system sweeps the ultrasonic sensor's field of view using the servo motor and calculates distances to objects based on the returned echoes.

Components:

Arduino Board

The central component of the project, the Arduino board, serves as the microcontroller that controls the operation of the radar system.

Jumper Wires

Used to establish connections between different components on the breadboard and Arduino board, jumper wires facilitate electrical connections in the circuit.

Breadboard

Provides a platform for prototyping the circuit. Components are connected on the breadboard using jumper wires, allowing for easy assembly and modification.

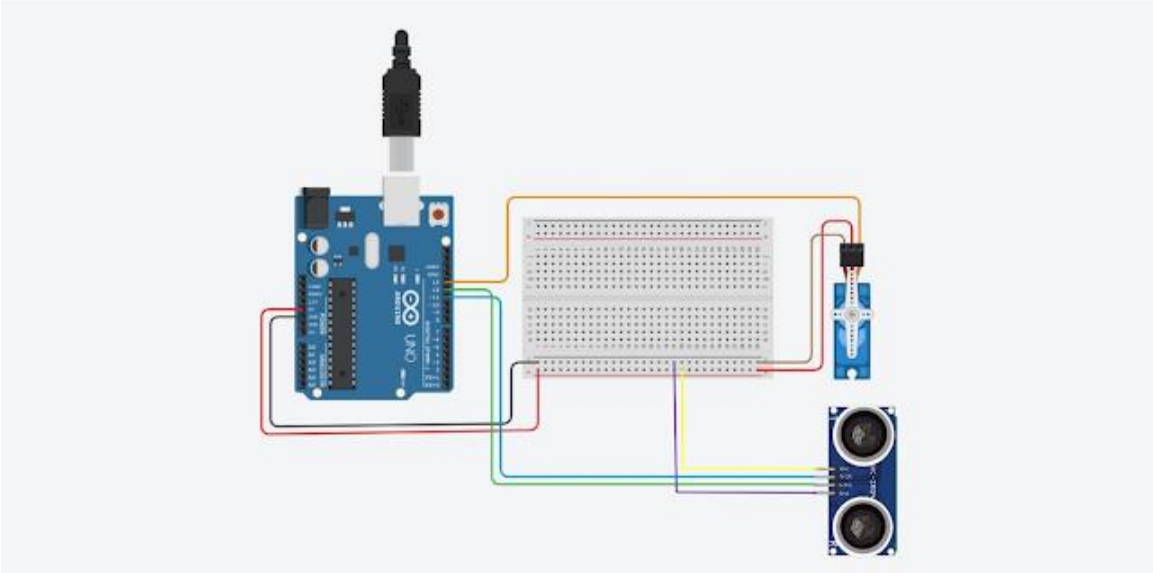
Ultrasonic Sensor

The ultrasonic sensor is the primary sensing component of the radar system. It emits ultrasonic waves and measures the time taken for the waves to bounce back after hitting an object, thus determining the distance to the object.

Servo Motor

The servo motor is responsible for sweeping the ultrasonic sensor's field of view. It rotates the sensor at specific angles to cover a range of directions, enabling the radar system to detect objects in multiple directions.

Circuit Diagram:



Code:

```
// Includes the Servo library
#include <Servo.h>

// Defines Trig and Echo pins of the Ultrasonic Sensor
const int trigPin = 11; // Pin connected to the trigger pin of the ultrasonic sensor
const int echoPin = 12; // Pin connected to the echo pin of the ultrasonic sensor

// Variables for the duration and the distance
long duration; // Variable to store the duration of the ultrasonic pulse
int distance; // Variable to store the calculated distance to objects

Servo myServo; // Creates a servo object for controlling the servo motor

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Initializes serial communication at 9600 baud rate
  myServo.attach(13); // Attaches the servo motor to pin 13
}

void loop() {
  // Rotates the servo motor from 15 to 165 degrees
  for(int i=15; i<=165; i++){
    myServo.write(i); // Rotates the servo to the specified angle
```

```
   delay(25); // Delays for 25 milliseconds for the servo to reach the position
   distance = calculateDistance(); // Calls a function for calculating the distance measured by the Ultrasonic sensor for each degree

   Serial.print(i); // Sends the current degree into the Serial Port
   Serial.print(","); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
   Serial.print(distance); // Sends the distance value into the Serial Port
   Serial.print("."); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
  }

  // Repeats the previous lines from 165 to 15 degrees
  for(int i=165; i>15; i--){
   myServo.write(i); // Rotates the servo to the specified angle
   delay(25); // Delays for 25 milliseconds for the servo to reach the position
   distance = calculateDistance(); // Calls a function for calculating the distance measured by the Ultrasonic sensor for each degree

   Serial.print(i); // Sends the current degree into the Serial Port
   Serial.print(","); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
   Serial.print(distance); // Sends the distance value into the Serial Port
   Serial.print("."); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing
  }
}

// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){
  digitalWrite(trigPin, LOW); // Clears the trigPin
  delayMicroseconds(2); // Waits for 2 microseconds

  digitalWrite(trigPin, HIGH); // Sets the trigPin on HIGH state for 10 microseconds
  delayMicroseconds(10); // Waits for 10 microseconds
  digitalWrite(trigPin, LOW); // Sets the trigPin off

  duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in microseconds
  distance= duration*0.034/2; // Calculates the distance in centimeters

  return distance; // Returns the calculated distance
```

```
}
```

Explanation of the Code:

- The code begins by including the Servo library, which provides functions for controlling servo motors.
- Pins for the trigger and echo of the ultrasonic sensor are defined.
- Global variables are declared to store the duration of the ultrasonic pulse and the calculated distance to objects.
- In the setup() function, the pins are configured as input and output, and serial communication is initialized.
- The main loop() function rotates the servo motor from 15 to 165 degrees and back, in increments of one degree.
- Within each loop iteration, the calculateDistance() function is called to measure the distance to objects.
- The calculateDistance() function triggers the ultrasonic sensor, measures the duration of the echo pulse, and calculates the distance based on the speed of sound.

Conclusion:

In conclusion, the Arduino-based radar system developed by the team demonstrates the effective use of Arduino microcontrollers, sensors, and motors to create a functional radar system capable of detecting objects within a specified range. The project showcases the team's skills in electronics, programming, and problem-solving, with potential applications in robotics, security systems, and automation.