SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY

CYBER PHYSICAL SYSTEMS FOR INDUSTRIAL APLLICATIONS

BATCH NO:61

SHAIK MOHAMMAD SUHAIL – 21781A04H5

SHAIK AYMAN TAHA - 21781A04H4

Y.SREEKANTH - 21781A04L7

POTTERI BEERAIAH -22785A0421

GITHUB LINK : https://github.com/ShaikMohammadSuhail7/SVCET-BATCH-61.git

AIM :

To design a system where the radar sensors interfaced with the controller can detect the objects and track it.

PROBLEM STATEMENT AND SOLUTION :

Problem Identification :

In industrial automation, there is an increasing demand for efficient and reliable detection systems to enhance safety, productivity, and operational efficiency. Traditional detection methods, such as proximity sensors, have limitations in detecting objects at longer distances and in adverse environmental conditions. To address these challenges design a system where the radar sensors interfaced with the controller can detect the objects and track it.

Solution :

DESIGNING AN Arduino Radar Project – Ultrasonic Based Radar

1. Servo Motor Adjustment:
    - Check the servo motor's power supply and ensure it's receiving a stable 5V from the Arduino.
    - Verify the servo motor's calibration and adjust it if necessary. You can use the Servo.write() function in the Arduino code to set the desired position (0-180 degrees).
    - If the issue persists, consider replacing the servo motor with a more reliable one.
2. Ultrasonic Sensor Troubleshooting:
    - Use the Arduino Serial Monitor to check the ultrasonic sensor's output values. This will help identify if the issue is with the sensor or the code.
    - Verify the sensor's wiring and ensure it's connected correctly to the Arduino.
    - If the issue persists, consider replacing the ultrasonic sensor with a new one.
3. Radar Scanning Issues:
    - Review the radar scanning algorithm and ensure it's correctly implemented in the code.
    - Check for any limitations or constraints in the ultrasonic sensor's range or scanning angle.
    - Consider adding error handling or debugging mechanisms to the code to better understand the issue.

PROJECT DESIGN SPECIFICATIONS :

Project Feasibility:

While the original project can be built within a week, keep in mind that it requires some understanding of Arduino programming and electronics. With zero prior knowledge, it might be challenging to complete the project on time. However, with dedication and focus, it's still possible to learn and adapt the project.

Recommended Approach:

Follow the steps provided in the original tutorial, and focus on understanding the basics of Arduino programming and electronics. You can start by learning the fundamentals of Arduino through online resources, such as tutorials and documentation. This will help you grasp the concepts and make the project more manageable.

Alternative Arduino Board:

Yes, you can use an Arduino Uno R3 instead of the Arduino Mega 2560. The Uno R3 is a more beginner-friendly board and still suitable for this project.

Radar Graphical Display:

The original project uses Processing for graphical display. If you're not familiar with Processing, you can consider alternative options:

- Use an LCD touchscreen (2.8 or 3.2 inches) with an Arduino-compatible library, such as the U8g2 library. This will require additional coding and setup.
- Alternatively, use a dedicated graphical display module, like the SparkFun OLED Display Shield, which provides a simple-to-use interface for Arduino.

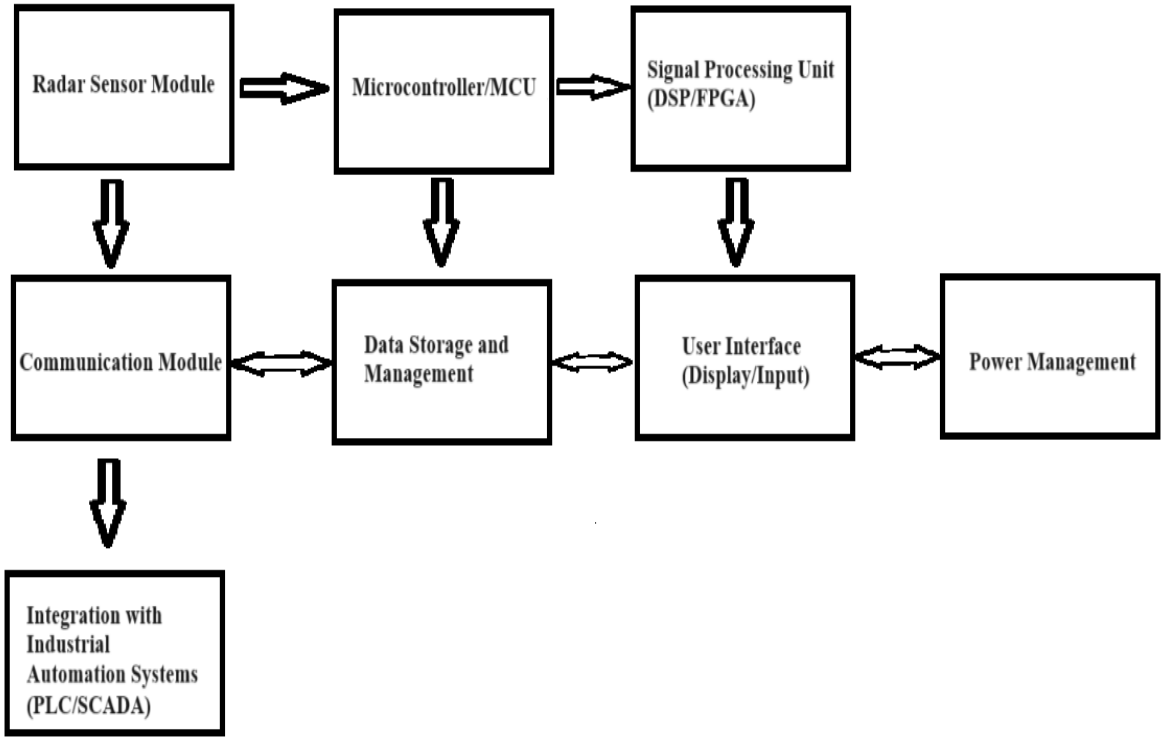
Precision:

To achieve high precision, you may need to refine the ultrasonic sensor calibration and averaging techniques. You can experiment with different calibration methods and averaging techniques to improve the accuracy.

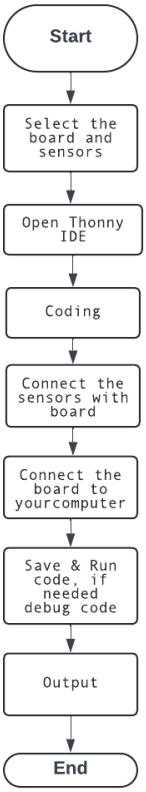
Raspberry Pi Alternative:

If you prefer to use a Raspberry Pi, you can adapt the project by using a compatible ultrasonic sensor and a Python library like RPi.GPIO for controlling the sensor and processing the data. You'll need to develop a custom graphical display using a Python library like Pygame or a dedicated GUI framework.

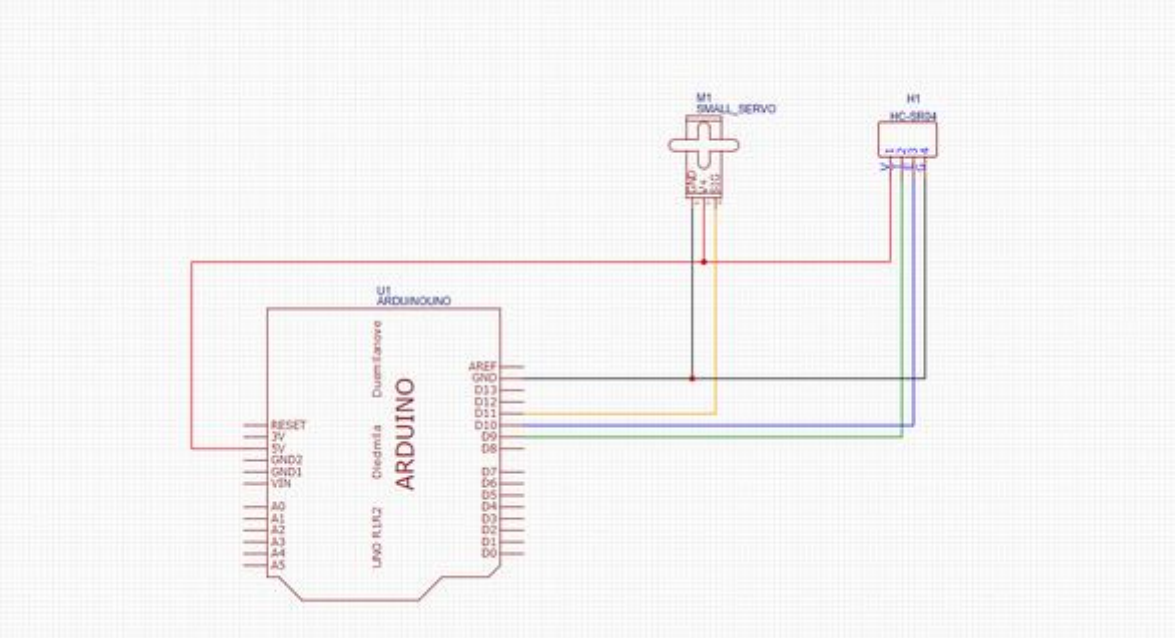
PROJECT ARCHITECTURE :



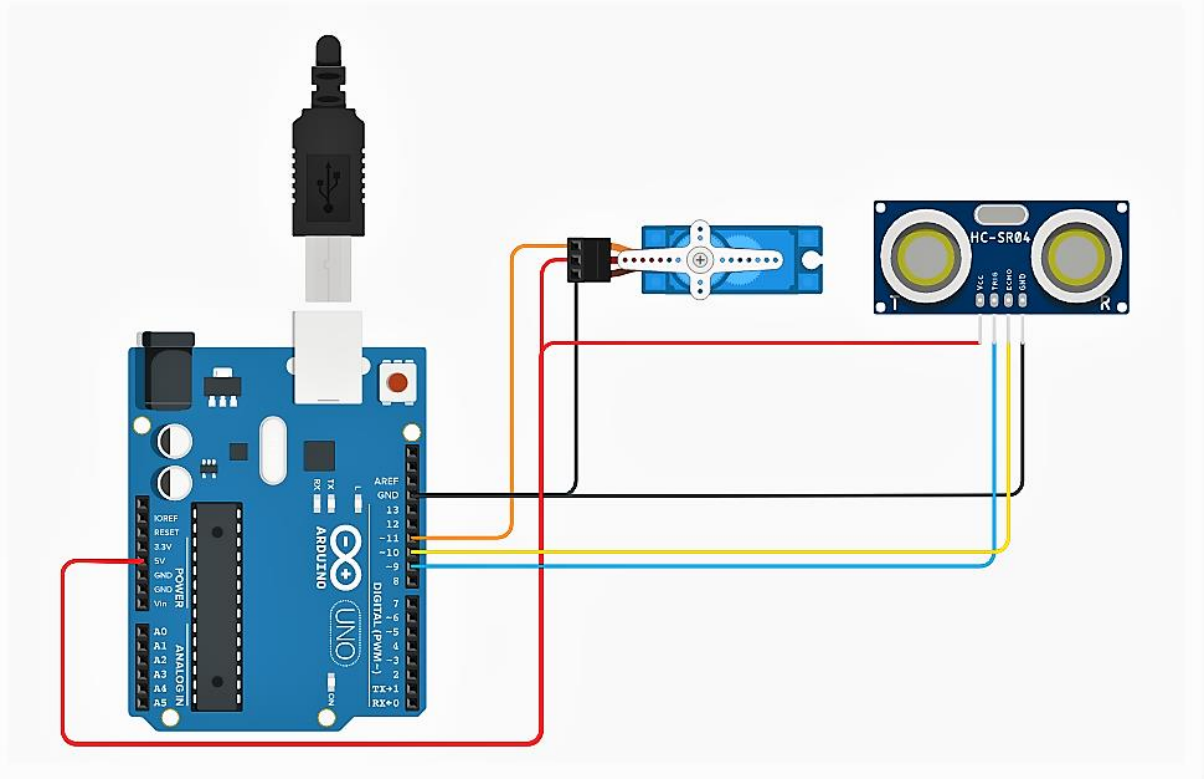
FLOW EXPLINATION :



WIRING DIAGRAM :



KiCad PCB Design :



KiCad Gerber File :

COMPONENTS WOKING PRINCIPLES/FUNCTIONALITY :

ULTRASONIC SENSOR :

An **ultrasonic sensor** is a proximity sensor that is used to measure the distance of a target or object. It detects the object by transmitting ultrasonic waves and converts the reflected waves into an electrical signal. These **sound waves** travel faster than the speed of the sound that humans can hear.

t has two main components: the **transmitter** and the **receiver**. The transmitter emits the sound using a **piezoelectric crystal**, and the receiver encounters the sound after it has traveled to and from the target.

For the calculation of the object distance, the sensor measures the time taken by the signal to travel from the transmission of the sound by the transmitter to the reflection back toward the receiver.

The formula for this calculation is,

D = ½ T x C

Where,

- **D = distance,**
- **T = time**
- **C = speed of sound, which is 343 meters per second.**

SERVO MOTOR :

The **servo motor** is a simple DC motor that can be controlled for specific angular rotation with the help of additional servomechanism. This motor will only rotate as much as we want and then stop. The servo motor is a closed-loop mechanism that uses positional feedback to control the speed and position.

This closed-loop system includes a control circuit, servo motor, shaft, potentiometer, drive gears, amplifier, and either an encoder or resolver.

The servo motor is unlike a standard electric motor, which starts and stops according to the power input. According to the signal, the servo motor will work.

Nowadays, servo motors are widely used in industrial and robotics applications. They are also commonly seen in **remote-controlled toy cars, RC planes, and CD or DVD players.**

ARDUINO UNO :

The **Arduino Uno** is an **open-source microcontroller board based on the ATmega328P 8-bit microcontroller**. It has 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

Key Features

- Compatible with Arduino IDE
- ATmega328P microcontroller with 32KB of Flash memory and 2KB of RAM
- Resettable polyfuse for USB port protection
- Can be powered via USB or external power supply
- UART TTL (5V) serial communication on digital pins 0 (RX) and 1 (TX)
- ATmega16U2 (or ATmega8U2 up to version R2) as a USB-to-serial converter

BILL OF MATERIALS(BOM) PREPARATION :

SN	Device/Instrument	Make	Model	Quantity	Est Unit Price	Total
1	**Arduino uno**			**1**	**800**	**800**
2	**Servo motor**			**1**	**150**	**150**
3	**Ultrasonic sensor**			**1**	**60**	**60**
						1010

ASSEMBLING HARDWARE COMPONENTS AND CODING:

GITHUBLINK: https://github.com/ShaikMohammadSuhail7/SVCET-BATCH-61.git

PROJECT OUTCOMES :

1. **Functional Radar System:**
    - **Your radar system will be able to measure distances to nearby objects.**
    - **It will rotate the ultrasonic sensor (mounted on a servo motor) to cover a 180-degree field of view.**
    - **The system will emit ultrasonic waves, receive their reflections, and calculate the distance based on the time taken for the waves to travel.**
2. **Visual Output:**
    - **You can display the radar output on your computer screen using the Processing IDE.**

- **The output will show the detected objects as blips or dots, with their corresponding distances.**
- **As the servo motor rotates, the radar display will update in real-time.**

3. **Learning Opportunities:**
   - **While building this project, you'll gain insights into:**
     - **Arduino programming (writing code to control the sensor and servo motor).**
     - **Sensor interfacing (connecting the ultrasonic sensor to the Arduino).**
     - **Servo motor control (rotating the sensor).**
     - **Basic radar principles (echo time, distance calculation).**
4. **Customization:**
   - **Feel free to modify the project:**
     - **Adjust the rotation angle of the servo motor.**
     - **Experiment with different ultrasonic sensors (some have wider or narrower detection angles).**
     - **Enhance the visualization by adding more features (e.g., color-coded blips).**