Project Title:

Arduino Based Ultrasonic Radar

Group Member:

Abdul Basit (2020-ag-6314)

Nouman Masieh (2020-ag-6323)

Hamna Shakeel (2020-ag-6332)

Submitted To:

Sir Ahsan Raza Saqib

University Of Agriculture Faisalabad

Introduction:

This project is an implementation of an Arduino based radar system that uses an ultrasonic sensor and a servo motor to scan its environment. The ultrasonic sensor is used to measure the distance to objects and the servo motor is used to rotate the sensor in order to scan a full range of angles. The Arduino board receives the distance measurements and controls the servo motor, and also sends the data to a computer via the serial port. The data received is then visualized on the computer using a Processing script to create a radar-like display.

Components and Setup:

- Arduino board
- Ultrasonic sensor
- Servo motor
- Processing IDE
- USB cable
- Jumper wires

The ultrasonic sensor is connected to the Arduino board with its trig and echo pins connected to specific digital pins on the board. The servo motor is connected to another digital pin on the board. The Arduino board is then connected to a computer via a USB cable. The Processing script is run on the computer to visualize the data received from the Arduino board.

Functionality:

The Arduino code controls the servo motor to rotate from 15 to 165 degrees in increments of 1 degree. At each angle, the ultrasonic sensor takes a distance measurement and sends the angle and distance data to the computer via the serial port.

The Processing script then visualizes this data in a radar-like display, with a moving line indicating the angle and distance of the object being detected by the sensor. The radar-like display simulates motion blur and slow fade of the moving line.

Applications:

This project can be used in a variety of applications such as object detection, navigation, and obstacle avoidance for robots and drones. The system can also be used for security, monitoring, and surveillance applications. The system can also be used in any projects where real-time distance information is needed.

Conclusion:

This project demonstrates the capabilities of an Arduino based ultrasonic radar system, which can be used in a variety of applications. The system can be further developed and customized to suit specific requirements and needs.

Coding

This code is written in C++ and is meant to be run on an Arduino board. The code uses the Servo library to control a servo motor, and an ultrasonic sensor to measure distance.

At the top of the code, the Servo library is included, and the pins that the ultrasonic sensor's trigger and echo signals are connected to are defined as constants (trigPin = 10 and echoPin = 11).

There are two variables declared for the duration and the distance measurements, and a Servo object is created to control the servo motor.

In the setup() function, the mode of the trigPin and echoPin are set as OUTPUT and INPUT respectively. The communication with computer is set on baud rate of 9600 and the servo motor is attached to the pin 12.

The loop() function is where the majority of the work is done. It uses a for-loop to rotate the servo motor through a range of angles, from 15 to 165 degrees in increments of 1 degree. For each angle, it calls a function calculateDistance() that measures the distance to an object using the ultrasonic sensor. The angle and distance measurements are then sent to the computer via the serial port. Then it repeats the same process from 165 to 15 degrees.

The calculateDistance() function uses the trigPin and echoPin of the ultrasonic sensor to measure the distance to an object. The trigPin is first set to a low state for 2 microseconds, then it sets to high for 10 microseconds. It waits for the echoPin to return a high state and records the duration of the echo signal. It then multiplies the duration with a constant to convert it to a distance value in centimeters and returns that distance value.

Overall this code controls a servo motor to rotate from 15 to 165 degrees and in each angle it takes distance measurement from ultrasonic sensor and send it over serial communication to the computer. The code allows for a continuous rotation of the servo motor, which can be used for creating a scan of the surrounding environment with the ultrasonic sensor.

This code is written in Processing, a programming language and development environment that is used for creating interactive visual designs. The code uses the Processing Serial Library to communicate with an Arduino board over a serial connection, which allows the Processing program to receive data from the Arduino.

The code starts by importing the necessary libraries for serial communication, reading data from the serial port, and handling exceptions. It then declares several variables, including "angle," "distance," "data," "noObject," "pixsDistance," "iAngle," "iDistance," "index1," "index2," and "orcFont."

In the setup() function, the program sets the size of the window to 1200x700 pixels and starts the serial communication with the Arduino board on the specified serial port (COM5) at a baud rate of 9600. it also set the buffer of the serial port to read characters until it reaches a fullstop.

The draw() function is called repeatedly by Processing and is used to update the display on the screen. In this code, it is responsible for drawing the radar display on the screen, which includes the radar arcs and lines, a moving line that indicates the angle of the object being detected, and the distance of the object from the radar.

It calls several other functions drawRadar(), drawLine(), drawObject(), and drawText() for different part of the display.

The serialEvent(Serial myPort) function is called whenever there is data available from the Arduino board. Inside this function, the program reads the data from the serial port until it reaches a fullstop, which is then split into two pieces: the angle and distance, by using substring method on data. The angle and distance are then converted from string to int for further use.

The drawRadar() function draws the radar display on the screen, which includes arcs and lines that represent the radar's range and angle.

The drawObject() function draws a line on the radar display that indicates the angle and distance of the object being detected by the Arduino board. The angle and distance are received from the Arduino board over the serial connection and are used to position the line on the radar display.

The remaining functions drawLine(), drawText() not discussed here. Overall the code creates a radar-like visualization that is updated in real-time based on the data received from an Arduino board over a serial connection, which is likely connected to a sensor.