PROJECT ABSTRACT

Ultrasonic Radar Scanner using Arduino

A 180° ultrasonic scanning radar — HC-SR04 on a servo with Arduino time-of-flight ranging, OLED readout and a live PC polar plot — delivered as a built-to-order student project with firmware, plot script, report, PPT and viva kit.

01. Title

Ultrasonic Radar Scanner using Arduino

02. Introduction / Background

Radar — radio detection and ranging — maps its surroundings by emitting energy and timing the echo. Ultrasound offers the same principle at a student's bench: the HC-SR04 fires a 40 kHz burst and measures the echo delay, converting round-trip time into distance at the speed of sound. Mounted on a servo that steps through a 180° arc, one sensor becomes a scanner, and each reading becomes a point on a polar plot — angle on the dial, distance as the radius. The result is the familiar radar sweep, rendered in real hardware and software, with an OLED panel giving standalone readout alongside the PC visualization.

03. Problem Statement

Radar remains an abstraction for most undergraduates: a block diagram with no moving parts they can touch. Robotics courses meanwhile teach obstacle sensing as single fixed readings, never showing how angle-plus-distance becomes a map. A scanning ultrasonic radar makes the principle tangible — students watch the servo sweep, see obstacles appear on the polar plot, and connect echo time to centimetres. The project must handle the physics honestly (beam width, soft surfaces, sound-speed temperature drift) and deliver both a standalone instrument and a PC visualization pipeline.

04. Objectives

- Sweep an HC-SR04 through a 180° arc on a servo with configurable step size.
- Measure distance per step by ultrasonic time-of-flight (datasheet range 2 cm–400 cm).
- Display live angle, distance and sweep status on a 0.96-inch OLED.
- Stream angle–distance CSV over USB for PC-side logging.
- Render a live polar radar plot in a Python script.
- Trigger a buzzer alert when an obstacle enters a configurable warning radius.

- Expose sweep speed, step size and warning distance as tunable constants.

05. Proposed Methodology

The SG90 servo is driven in discrete steps across the 180° arc, pausing at each for the HC-SR04 to fire and the echo pulse to be timed; distance follows from the round-trip time at 343 m/s nominal sound speed. The OLED UI shows the current step angle and measured distance with sweep direction. A CSV protocol (angle,distance) over USB feeds a Python plotting script that draws the live polar sweep, marks obstacles and records sessions to file. A buzzer threshold constant implements the proximity alarm. Testing characterizes real behaviour: detection of hard vs soft targets, blind zones, multi-echo artefacts and the effect of step size on scan speed, all documented as measured performance.

06. System Architecture / Working

The Arduino steps the servo to each angle in turn and commands the HC-SR04 to ping; the echo pin's high duration converts directly to distance. Each angle–distance pair is written to the OLED and emitted as a CSV line over USB. The PC script parses the stream and paints the polar plot in real time — one dot per step, sweep line advancing, obstacle blobs persisting. When a reading falls inside the warning radius, the buzzer sounds while the sweep continues. The scan direction reverses at the arc limits for continuous scanning.

07. Hardware / Software Requirements

- Arduino Uno (ATmega328P).
- HC-SR04 ultrasonic sensor (40 kHz, datasheet range 2 cm–400 cm).
- SG90 micro servo (180° sweep).
- 0.96-inch OLED display (SSD1306, I2C).
- Active buzzer module for proximity alerts.
- USB cable to PC for CSV streaming.
- Software: Arduino IDE (C/C++ firmware); Python 3 with plotting libraries for the polar-plot script.

08. Expected Outcomes

A working scanner that sweeps the full arc, reports per-step distance on the OLED, and paints a live polar plot of the surroundings on the PC. The buzzer warns on near obstacles per the configured radius. Characterization documents honest detection behaviour — effective range, target-type dependence, artefacts — so the viva discussion is grounded in measured results rather than datasheet claims.

09. Applications

- Robotics obstacle mapping demos.
- Embedded-sensing teaching labs.
- Parking-sensor and proximity demonstration rigs.
- Tank-level sensing proof of concept.
- Exhibition and science-fair interactive displays.

10. Future Scope

A continuous-rotation version with slip ring would give true 360° radar; temperature compensation of sound speed would tighten accuracy. Multiple sensors, Kalman-style smoothing, and occupancy-grid mapping are the natural path toward real robotic mapping (SLAM-lite) on this platform.