Arduino Radar with Simulink

1. Detailed Project Description

The development of this project is focused on the design and implementation of a short-range radar detection system using the Arduino Mega 2560 microcontroller, ultrasonic sensing technology, and a servo motor scanning mechanism. The control logic and system modeling have been developed in the Simulink working environment, which is a MATLAB-based graphical programming platform for modeling, simulation, and analysis of dynamic systems.

Simulink provides direct hardware support for Arduino Mega 2560 through the Simulink Support Package for Arduino Hardware. In this implementation, the Simulink model replaces traditional Arduino IDE programming by generating and deploying embedded code directly to the microcontroller.

Additionally, the Processing software environment is used on the personal computer to receive serial data transmitted from the Arduino board and to generate a real-time radar visualization interface.

The complete system integrates hardware control, signal processing, serial communication, and graphical data visualization into a unified radar detection platform.

The block diagram of the system connection consists of the following main components:

- Arduino Mega 2560
- Ultrasonic Sensor (HC-SR04)
- SG90 Servo Motor
- Personal Computer running Processing software

The structure of the system can be summarized as:

Ultrasonic Sensor + Servo Motor → Arduino Mega 2560 → Serial Communication → Processing Radar Interface

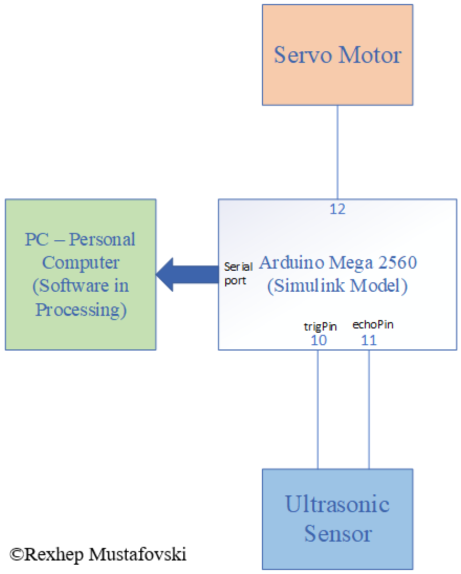

Servo Motor
12
PC – Personal Computer (Software in Processing)
Serial port
Arduino Mega 2560 (Simulink Model)
trigPin
10
echoPin
11
Ultrasonic Sensor
©Rexhep Mustafovski

Figure 1. Block diagram of connection

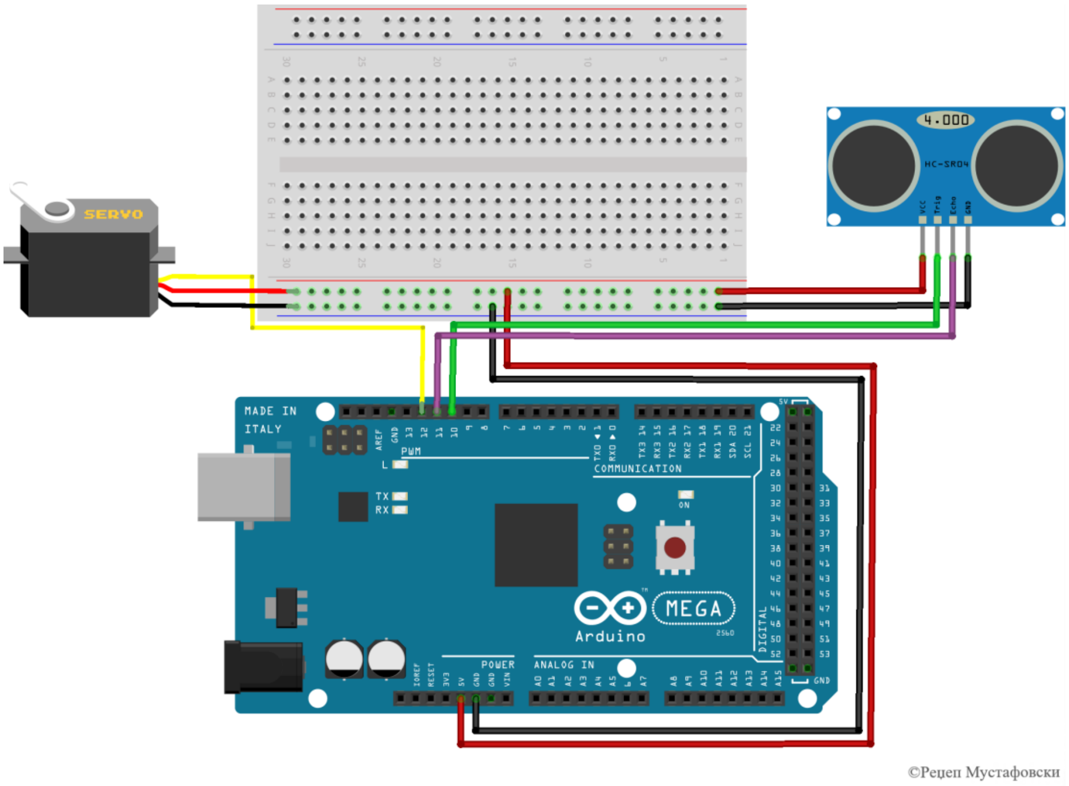


Figure 2. Circuit Connection Diagram of Arduino Radar System with Servo and Ultrasonic Sensor

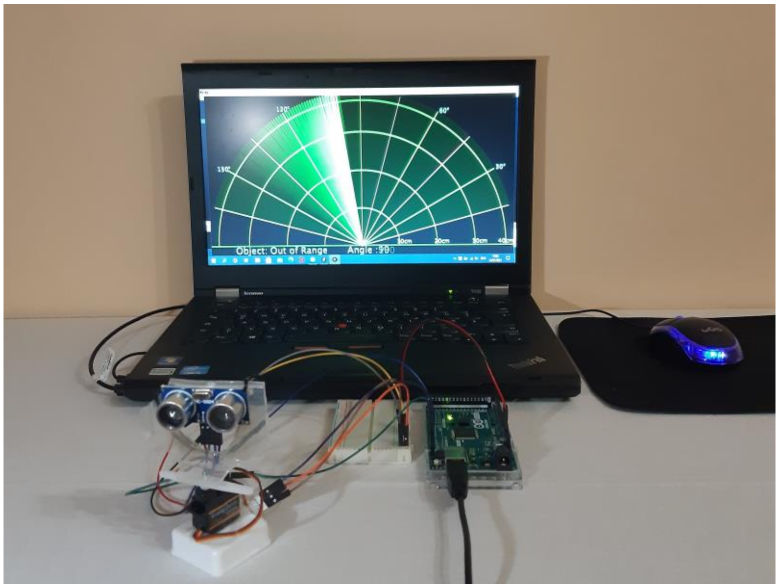


Figure 3. Experimental Setup and Real-Time Radar Visualization Interface

2. Simulink Model Development and Integration with Arduino Mega 2560

To create the radar detection model, the Simulink simulation tool is launched from the MATLAB command window using the "simulink" command. A blank model is created and configured for hardware deployment to the Arduino Mega 2560 board.

The system is developed using block-based programming from the Simulink Library Browser.

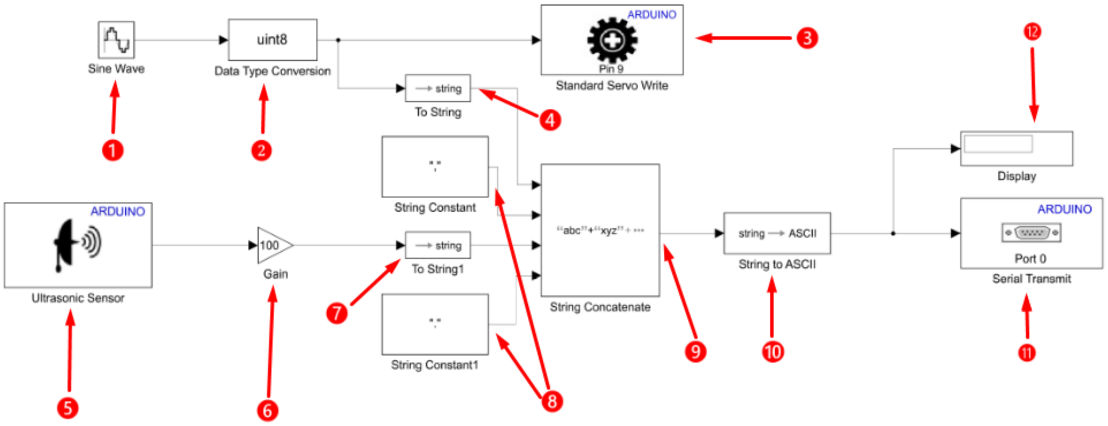


Figure 4. Appearance of the model in the Simulink simulation tool

2.1. Angle Generation and Servo Control

The first functional block used in the model is the "Sine Wave" block located in the "Sources" library. This block generates a periodic sinusoidal signal which is converted into discrete integer values representing angular positions between 0 and 180 degrees.

Since the servo motor operates within a range of 0 to 180 degrees, the generated signal is passed through a "Data Type Conversion" block to ensure integer formatting suitable for servo positioning.

The "Standard Servo Write" block, located in the Simulink Support Package for Arduino Hardware under the "Common" section, is used to control the shaft position of the servo motor. The servo rotates continuously, scanning the environment and changing the orientation of the ultrasonic sensor.

This rotational motion simulates the sweeping behavior of a conventional radar antenna.

2.2. Distance Measurement Using Ultrasonic Sensor

The "Ultrasonic Sensor" block, located in the "Sensors" section of the Simulink Support Package for Arduino Hardware, is used to measure the distance to nearby objects.

The ultrasonic sensor operates by:

1. Sending a high-frequency acoustic pulse.
2. Receiving the reflected echo from an object.
3. Calculating the time difference between transmission and reception.
4. Converting the time interval into distance measurement.

The output of the block provides distance values in meters. To obtain more practical units, a "Gain" block is used to convert the measured value from meters to centimeters.

The sensor continuously measures object distance as the servo motor changes its angular position.

2.2. Serial Data Formatting and Transmission

In order to visualize the radar data on the personal computer, both angle and distance values must be transmitted via the serial port.

The measured angle and corresponding distance are formatted into a single structured string using:

- "String Constant" blocks
- "String Concatenate" block

The resulting data structure has the format:

angle,distance.

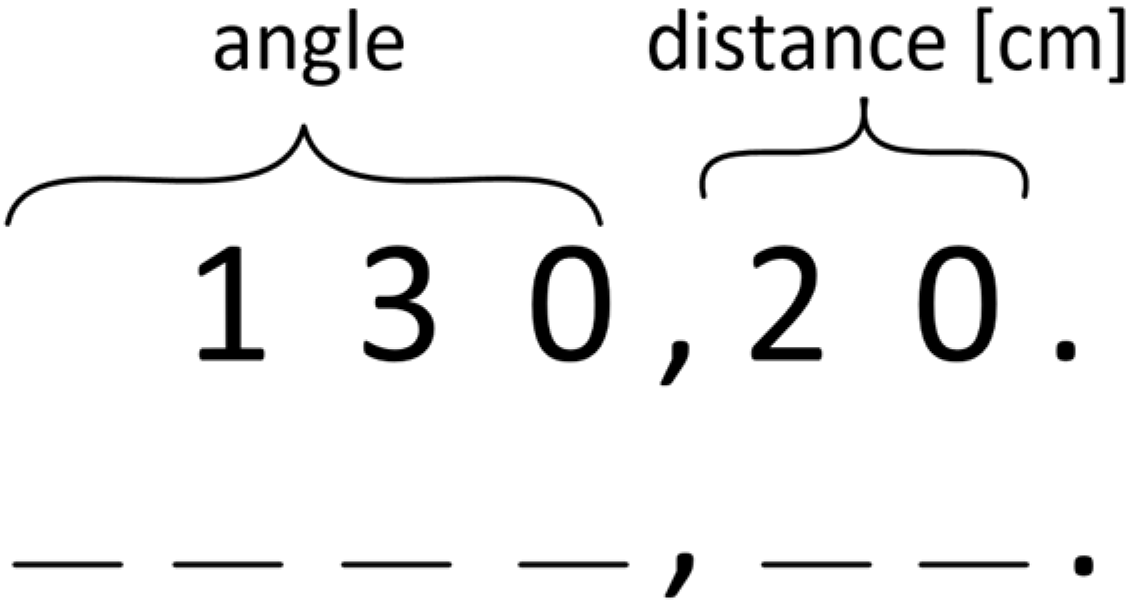


Figure 5. Structure of the data for the serial port

This string is converted into ASCII format using the "String to ASCII" block to ensure compatibility with serial communication protocols.

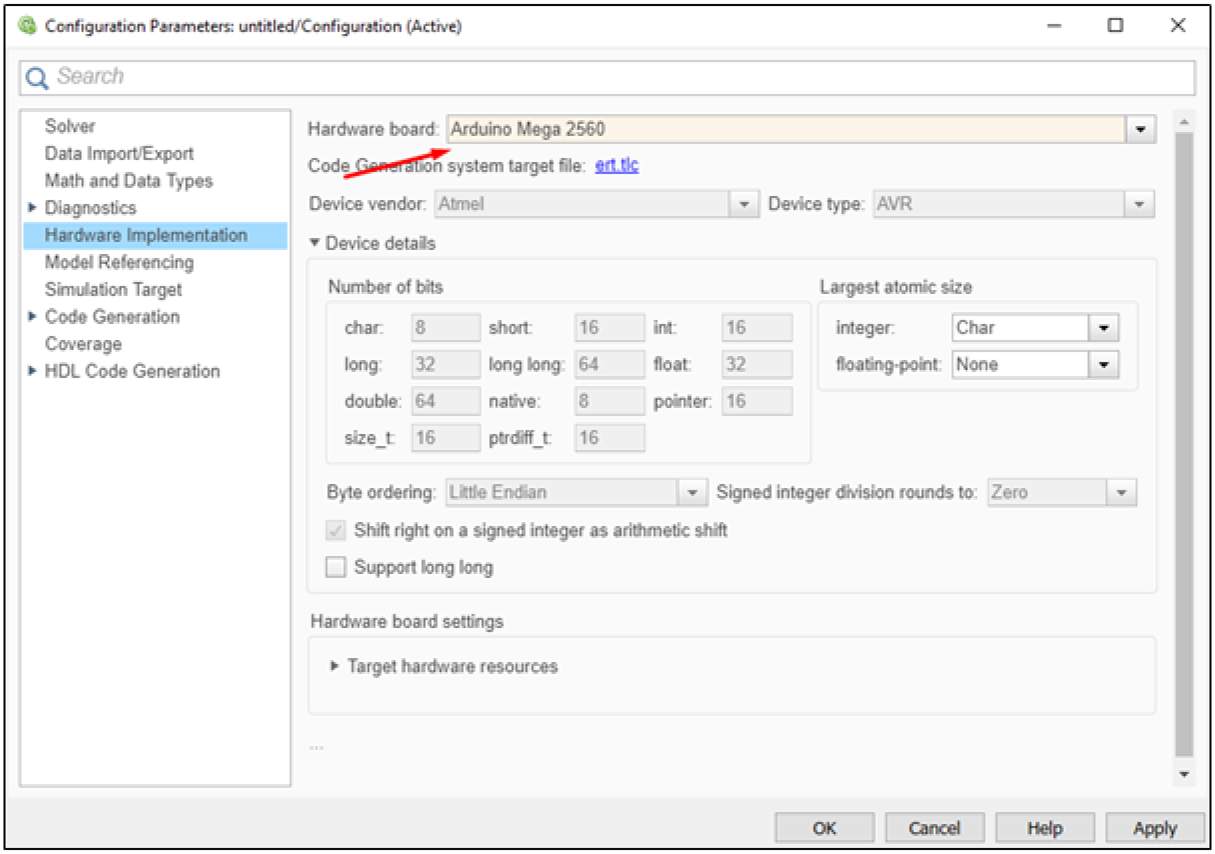
Finally, the "Serial Transmit" block sends the formatted data to the serial port of the Arduino Mega 2560 at a baud rate of 115200 bps.

3. Programming and Deployment of the Simulink Model

Once the Simulink model is completed, the hardware configuration must be selected under the "Hardware Implementation" settings. The Arduino Mega 2560 board is chosen as the target hardware.

Before deployment, the Simulink Support Package for Arduino Hardware must be installed using the MATLAB "Add-Ons" option.

After configuration, the model is compiled and deployed using the "Build, Deploy & Start" function. The generated code is automatically transferred to the Arduino board, which then begins executing the radar scanning algorithm independently.



Configuration Parameters: untitled/Configuration (Active)

Hardware board: Arduino Mega 2560

Code Generation system target file: ert.tlc

Device vendor: Atmel Device type: AVR

Device details

Number of bits

char:	8	short:	16	int:	16
long:	32	long long:	64	float:	32
double:	64	native:	8	pointer:	16
size_t:	16	ptrdiff_t:	16		

Largest atomic size

integer:	Char
floating-point:	None

Byte ordering: Little Endian Signed integer division rounds to: Zero

☑ Shift right on a signed integer as arithmetic shift

□ Support long long

Hardware board settings

Target hardware resources

Figure 6. Selecting the Arduino Mega 2560 hardware in configuration parameters

The Arduino board is physically connected to:

- Ultrasonic sensor (Trigger and Echo pins)
- Servo motor (PWM control pin)
- Personal computer via USB cable

4. Radar Image Drawing on the PC Using Processing

The Processing environment is used to create a real-time radar visualization interface.

The Processing program performs the following operations:

- Initializes serial communication with the Arduino board.
- Continuously reads incoming serial data.
- Parses angle and distance values.
- Converts polar coordinates into Cartesian screen coordinates.
- Draws radar arcs and angular reference lines.
- Displays detected objects in real time.

The graphical interface simulates a traditional radar screen with:

- Green sweeping beam
- Circular distance markers (10 cm, 20 cm, 30 cm, 40 cm)
- Angular markings (0° to 180°)
- Object detection indicator

When an object is detected within 40 cm range, it is displayed in red on the radar screen at the corresponding angle and distance.

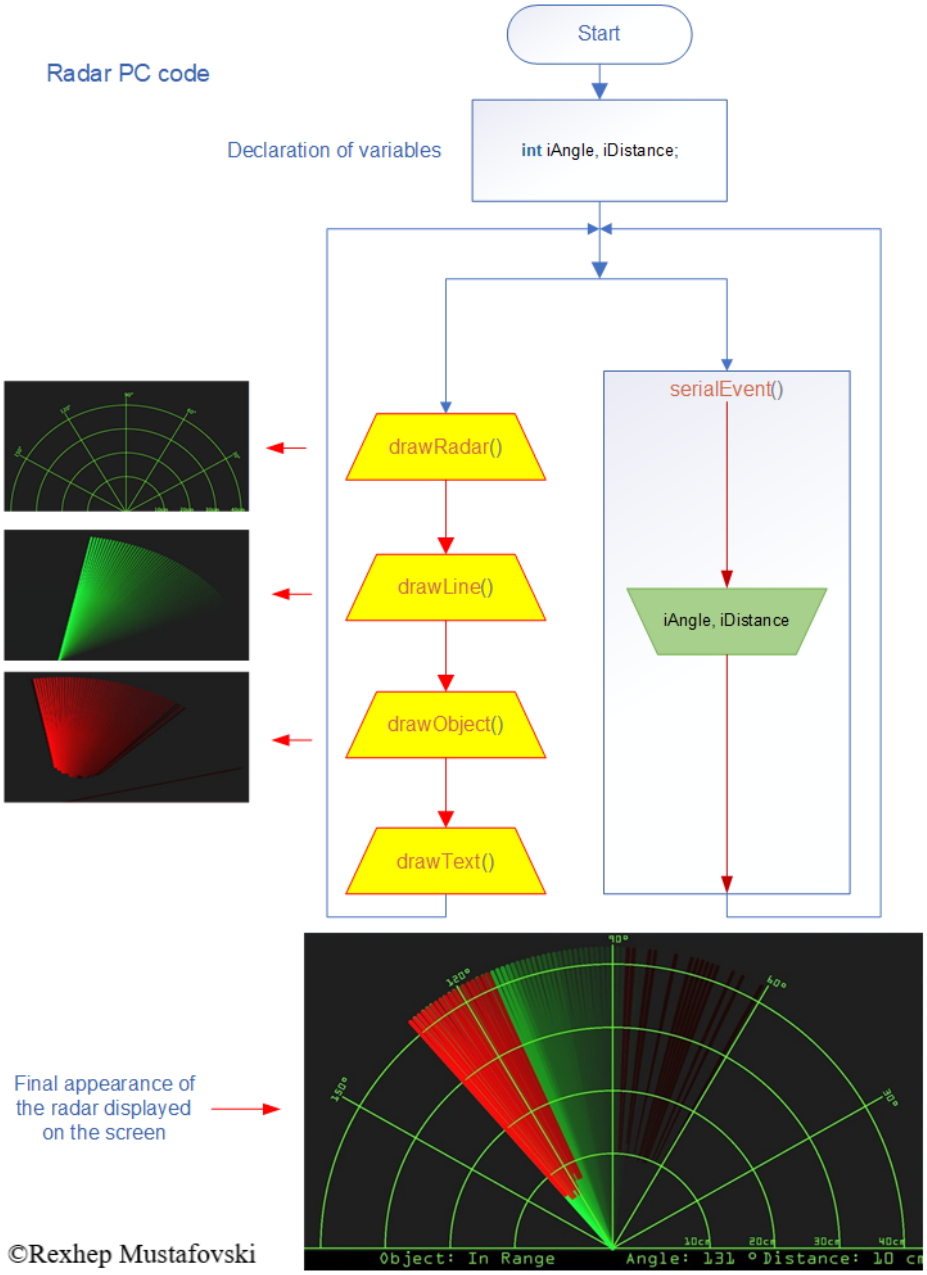
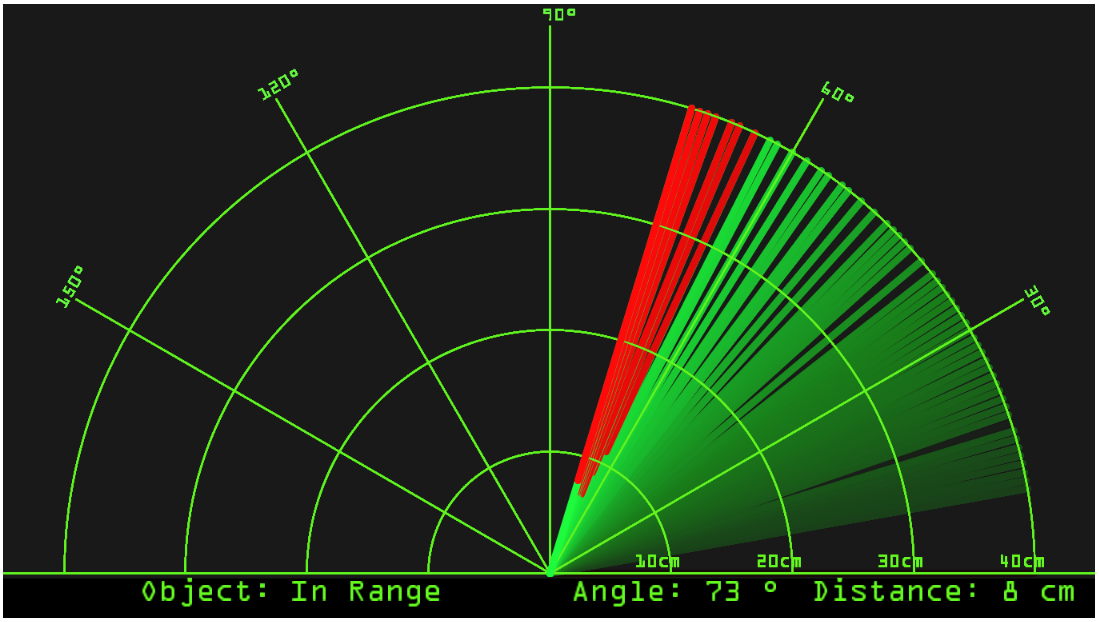


Figure 7. Algorithm of Processing code

For example, if the sensor detects an object at 73° and 8 cm, the Processing interface displays the object location and updates the text fields:

Object: In Range
Angle: 73°
Distance: 8 cm

If no object is detected within the defined range, the system displays "Out of Range".



Object: In Range Angle: 73 ° Distance: 8 cm

Figure 8. The radar appearance from Processing program

5. System Operation Principle

The radar operates by continuously rotating the servo motor between 0° and 180°. At each angular position, the ultrasonic sensor measures the distance to nearby objects.

The system performs the following cycle:

1. Generate angle value
2. Rotate servo motor
3. Measure object distance
4. Format data into structured string
5. Transmit data via serial communication
6. Visualize object position in Processing

This continuous loop simulates real radar scanning behavior in a simplified short-range detection environment.

6. Educational Value

The Arduino Radar with Simulink project provides significant educational value in the fields of embedded systems, sensor integration, model-based design, and real-time data visualization.

From a pedagogical perspective, the project enables students and researchers to understand the practical implementation of a complete sensing and control system, beginning from model development in Simulink to hardware deployment and graphical visualization.

The project supports the development of the following competencies:

- Understanding of model-based design using Simulink
- Integration of microcontroller hardware with simulation tools
- Implementation of ultrasonic sensing principles
- Servo motor control and angular positioning
- Serial communication protocol design
- Real-time data acquisition and visualization

Furthermore, the system demonstrates the transformation of physical measurements into structured digital data and their representation in a graphical radar interface. This process strengthens understanding of signal processing, coordinating transformation, and embedded communication systems.

The project is particularly suitable for laboratory exercises in courses related to:

- Embedded Systems
- Mechatronics
- Control Systems
- Industrial Automation
- Signal Processing
- IoT and Smart Systems

By combining theoretical modeling with practical hardware implementation, the project bridges the gap between simulation-based engineering design and real-world embedded system deployment.

7. Conclusion

The Arduino Radar with Simulink project demonstrates the integration of model-based design, embedded hardware control, real-time sensing, and graphical data visualization.

By combining Simulink modeling, Arduino hardware deployment, and Processing-based visualization, the project provides a complete educational example of:

- Embedded system design

- Sensor integration
- Servo motor control
- Serial communication
- Real-time graphical interface development

The system represents a functional short-range radar prototype suitable for laboratory experimentation, educational demonstrations, and further research development.