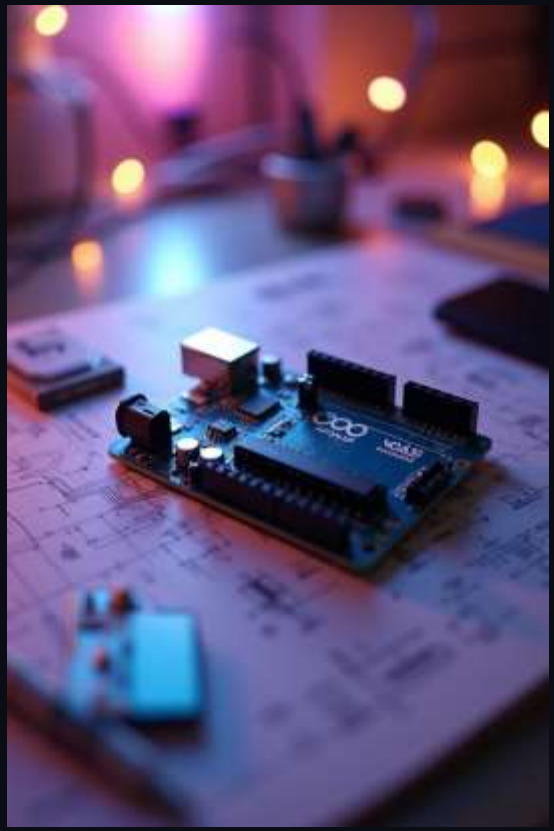
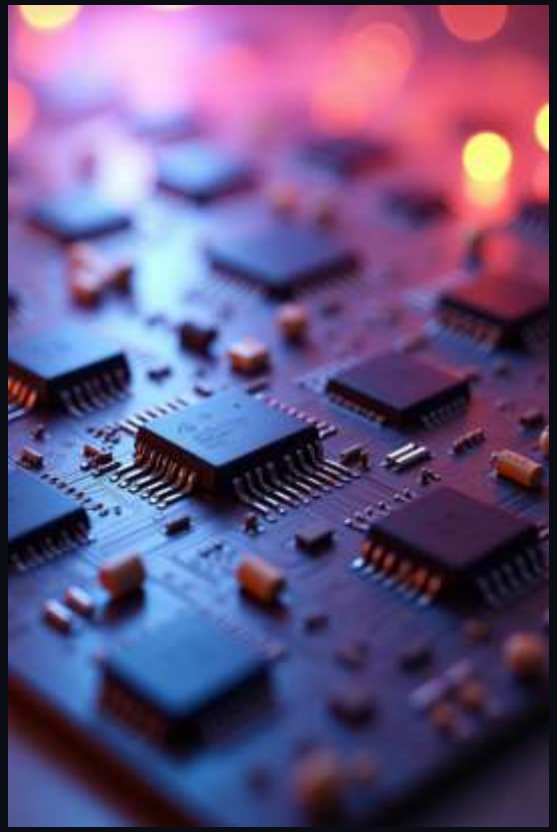
Arduino Journey



Introduction

This presentation summarizes the evolution from basic electronics knowledge to crafting a full robotic system using Arduino without pre-made kits. Emphasis is on improvisation, problem-solving, and hands-on learning that integrates hardware, software, mechanics, and real-time visualization for a comprehensive robotics experience.



01

Hardware Foundations

ARDUINO SKILLS ECOSYSTEM: From Fundamentals to Advanced Integration

Fundamentals

- Strong command of Arduino IDE
- Sketch structure mastery (setup/loop)
- Digital I/O handling
- Pull-up logic understanding
- Using millis() for timing
- Writing clean embedded-style code

Storage & Data Logging

- CSV data structure understanding
- Creating sensor logs manually and by program
- Data formatting for radar-style output
- GPS coordinate logging knowledge

Displays & Output Devices

- LCD 16x2 (regular + I²C)
- OLED usage (basic level)
- Seven-segment display basics
- LED signaling + buzzer integration

Communication Protocols

- UART (GPS, debugging, ESP)
- I²C (display backpacks, sensors)
- SPI (displays/modules)
- Serial data parsing
- Multi-protocol environment handling

Analog & Timing

- ADC reading (pot, LDR, sensors)
- PWM control (motors, servos, fan control)
- Debouncing logic (borrowed from 8051 expertise)
- Basic non-blocking timing concepts

Motor Control

- Servo control: SG90, MG995 (including gear repair & calibration)
- BO motor rotation with PWM
- Radar-style scanning using servo sweep
- Understanding servo potentiometer alignment
- Safe power handling for motors

Sensor Interfacing

- HC-SR04 ultrasonic sensor
- IR obstacle sensor + relay automation
- GPS module (NEO-6M)
- Current sensing basics
- Reading analog sensors for tracking/logging

ARDUINO SKILLS ECOSYSTEM: Advanced Integration & Real-World Application

Wireless & IoT

- ESP32 basics (LED blink, upload troubleshooting)
- GPS-based IoT logic (for your tracker)
- Conceptual understanding of WiFi/Bluetooth modules
- Using Arduino with wireless power demo modules

INTEGRATED SYSTEM HUB

Power & Hardware Knowledge

- Relay interfacing
- Battery charger module integration
- Boost/buck module basics
- Safe wiring + grounding
- Servo/motor power management
- Noise reduction awareness

Interrupts & Real-Time

- External interrupt concepts
- Debouncing via ISR (from 8051 knowledge)
- Real-time event handling
- Interrupt-safe logic understanding

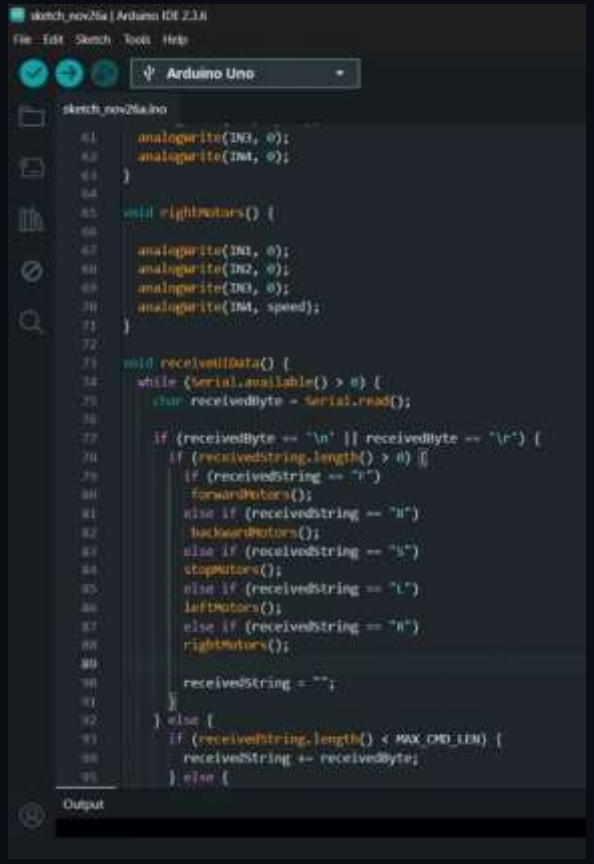
Project-Level Engineering

You have executed real multi-module systems:

- GPS-based solar tracker
- Radar scanner using servo + HC-SR04
- Wireless charging demo project
- Relay-based control system
- Sensor + serial + motor integration
- Hardware troubleshooting (cables, servo internals, alignment)

Embedded-Style Skills (From 8051 → Arduino)

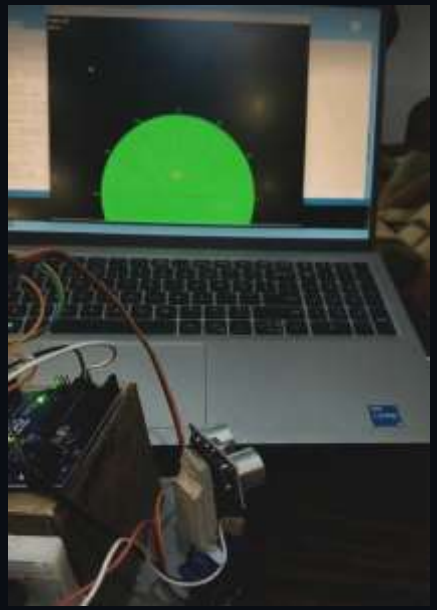
- Thinking in registers, timing, and low-level flows
- Writing structured, modular code
- Using interrupts logically
- Dealing with noise, debounce, and timing conflicts
- System-level thinking (architecture + flowcharts)



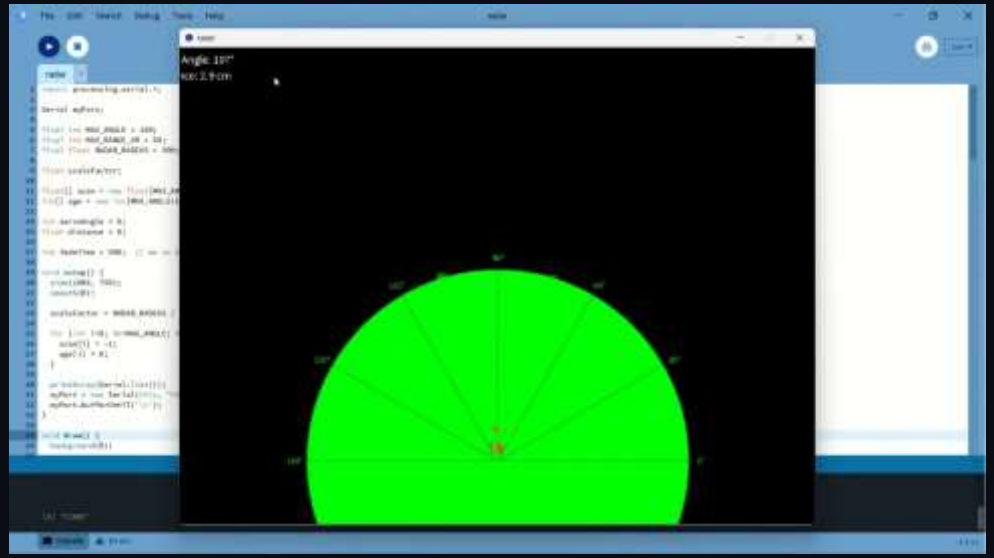
02

Software , Visualization and implimentation

Firmware and GUI Development



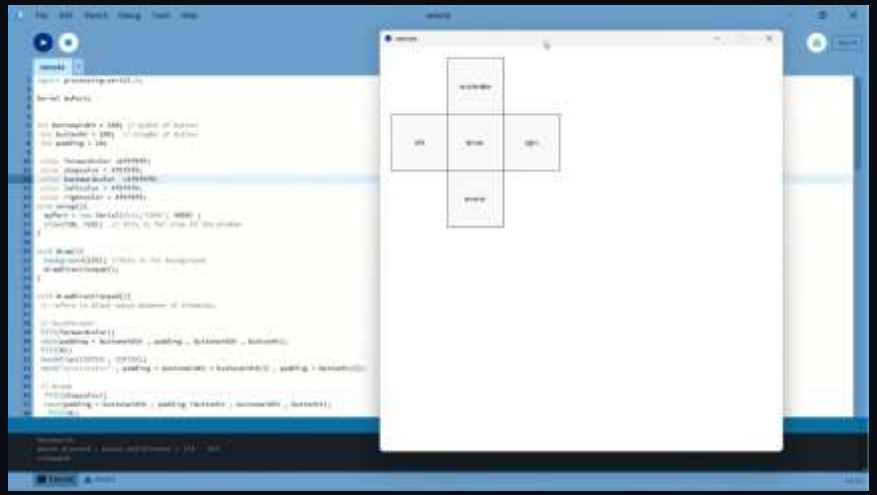
The software architecture consists of Arduino C/C++ firmware for efficient hardware control and a Processing-based GUI for PC interaction. This combination delivers seamless command transmission and telemetry feedback, allowing real-time control and monitoring of the robotic system with an intuitive user interface.



Real-Time Data Processing

Real-time data processing converts raw sensor inputs into actionable insights. Distance measurements from ultrasonic sensors are translated into graphical data with minimal latency. This processing is essential for tasks like obstacle detection and navigation, enabling responsive and accurate robot behavior.

Visualization Techniques and Tools



Visualization employs Processing's graphics engine to render real-time radar scans featuring moving arcs, fading dots, and angular sector displays. These visual elements provide intuitive feedback on the robot's surroundings, enhancing situational awareness and decision-making during operation.

Conclusions

The project successfully integrates **electronics, coding, and mechanical design** into a cohesive robotic system with real-time visualization. Practical experience in planning, debugging, and system optimization lays a strong foundation for future advances in robotics, automation, and IoT applications.

Thank You

Questions?

djnkajdnksnckjsanckj