

Voice-Controlled Robot Car

Xiyana V. Figuera — Team Lead · with Seong-Yong Jung, Yong-Un Kim, Da-Young Kim

Team project · School of Convergence Mechanical Engineering, Silla University · Advisor: Prof. Sang-Seok Yoon ·
December 2018

OVERVIEW

A first-year embedded-systems team project: a small robot car that drives on spoken commands. The goal was to use the human voice as the interface between a person and a machine, leaning on open-source hardware/software (Arduino, Raspberry Pi) and a commercial voice assistant (Amazon Alexa) rather than training any model of our own. Two independent control paths were built and compared — a cloud pipeline driven by Amazon Alexa, and a lighter direct-Bluetooth path — to weigh their trade-offs. As team lead, I coordinated the work and the integration across cloud services, two microcomputers and four codebases.

SYSTEM ARCHITECTURE

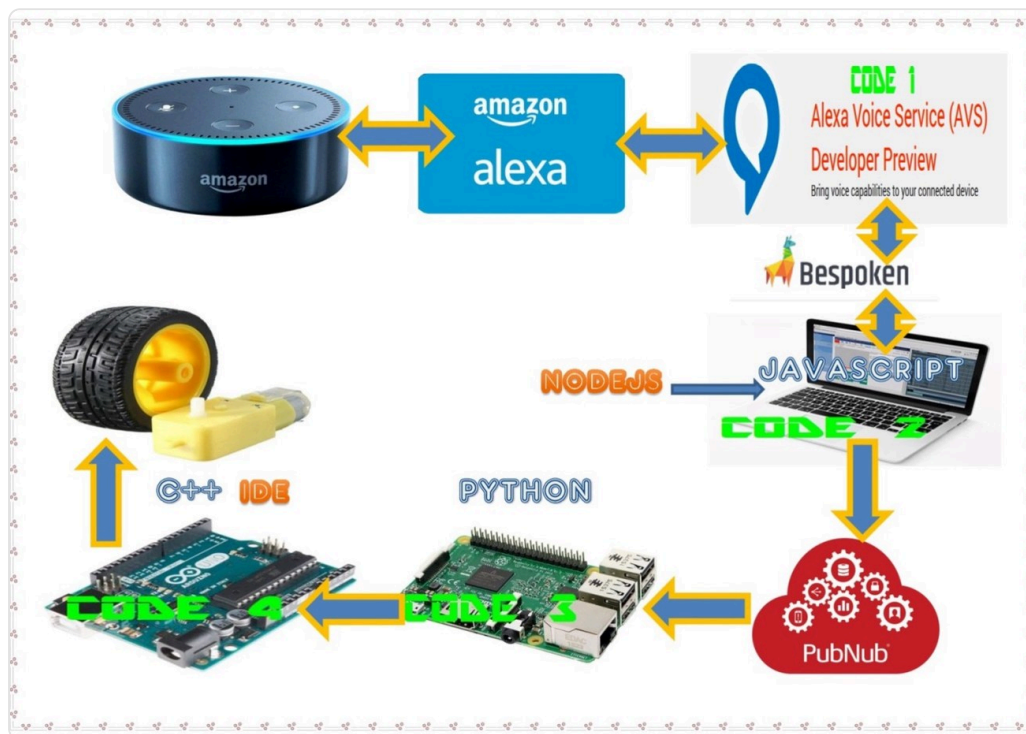


Fig. 1. Alexa control path: a spoken command travels from the Amazon Echo through the Alexa Voice Service and a Node.js skill, over PubNub cloud messaging, to the Raspberry Pi and Arduino that drive the motors.

TWO CONTROL PATHS COMPARED

A · Amazon Alexa (cloud)

richer, programmable, more moving parts

- Echo Dot → Alexa Voice Service (AVS)
- Node.js skill (tested with Bespoken)
- PubNub cloud pub/sub messaging
- Raspberry Pi (Python) → Arduino (C++)
- **A single pre-programmed phrase can trigger a whole sequence of actions.**

B · Bluetooth app (direct)

simpler, self-contained, less flexible

- "Arduino Voice Control" app (Google Play)
- Bluetooth → Arduino (C++) → motors
- No cloud, no extra services
- Fast to set up and fully offline
- Each action must be dictated one command at a time.

TECHNOLOGY STACK

Amazon Echo / Alexa

Alexa Voice Service

Bespoken

Node.js / JavaScript

PubNub

Raspberry Pi · Python

Arduino · C++

Bluetooth

DC motors

OUTCOME & WHAT I LEARNED

Both paths successfully drove the car by voice. The comparison made the trade-off clear: the Bluetooth app is quick and offline but requires dictating each command, whereas the Alexa pipeline is heavier to build but lets one spoken phrase invoke a pre-programmed routine — a more scalable interaction. Beyond the result, the project taught me the versatility of Arduino as a hardware-and-software platform, serial communication, working with open-source tooling, and — hardest of all as a first-year — wiring cloud services to physical hardware and making four languages and platforms cooperate end to end.

Note: this is a condensed, reconstructed brief of an undergraduate team project (2018) for portfolio purposes. It is a systems-integration project; no machine-learning models were trained (Amazon Alexa is used as a service).