



Universidad de Magallanes
Facultad de Ingeniería
Departamento de Ingeniería en Computación e
Informática

KubiBot: Informe Técnico

Ivan Mansilla
Ayrton Morrison
Ingeniería Civil en Computación e Informática

Docente guía: Mgt. Eduardo Peña
Curso: Taller de Integración

Resumen

Resumen bien epico carajo

ÍNDICE GENERAL

Índice de figuras	II
1. Introducción	1
2. Objetivos	2
2.1. Objetivo Principal	2
2.2. Objetivos Secundarios	2
3. Marco teórico	3
4. Desarrollo	4
4.1. Análisis previo	4
4.1.1. Módulos del prototipo	4
4.1.2. Diseño	5
4.2. Movimiento	6
4.2.1. Movimiento hacia adelante y hacia atrás	6
4.2.2. Detección de obstáculos	6
4.2.3. Movimiento autónomo	6
4.3. Comunicacion	7
4.3.1. Arquitectura del sistema	7
4.4. Comunicación Raspberry-Arduino	8
4.5. Diseño	8
5. Conclusión	9

ÍNDICE DE FIGURAS

4.1. Diagrama de comunicacion Servidor-Cliente	8
--	---

INTRODUCCIÓN

OBJETIVOS

2.1. Objetivo Principal

Diseñar y desarrollar un prototipo de robot de compañía móvil, integrado con un modelo de inteligencia artificial conversacional, capaz de interactuar de forma autónoma y asistir al usuario dentro de un entorno doméstico

2.2. Objetivos Secundarios

Se plantean los siguientes objetivos secundarios:

1. Desarrollar el sistema de movimiento autónomo del robot, incluyendo diseño del chasis, integración de motores y programación de sistema de evitación de obstáculos, utilizando un microcontrolador Arduino Uno
2. Implementar inteligencia artificial en la plataforma Raspberry Pi, abarcando la configuración del sistema, creación de API como intermediario entre host de la LLM y Raspberry PI, e integración de modelos de reconocimiento de voz (Speech-to-text)
3. Integrar periféricos de comunicación (entrada y salida) para la interacción usuario-robot, esto incluyendo instalación de micrófono y altavoz
4. Ensamblar los distintos módulos de hardware del prototipo, procurando coordinación entre Arduino y Raspberry
5. Validar la funcionalidad completa del robot mediante pruebas de software y hardware

MARCO TEÓRICO

DESARROLLO

4.1. Análisis previo

El punto de partida de este proyecto fue el interés en explorar las aplicaciones prácticas de la inteligencia artificial generativa en un formato físico interactuable, con el fin de crear un dispositivo que genere cercanía y acompañamiento al usuario.

El primer concepto consistía en utilizar un hardware disponible en la institución: una cabeza robótica estática. La idea era dotar a esta cabeza de capacidades conversacionales, aprovechando su estructura existente para simular interacción humana (movimiento de ojos o boca).

Sin embargo, tras un análisis preliminar, se descartó la idea, y se prefirió crear un hardware de cero, ya que entregaría mayor libertad de diseño e integración. Esto llevó entonces hacía un concepto nuevo: un prototipo de robot móvil y compacto.

Esta nueva dirección se eligió por dos ventajas estratégicas:

1. **Simplicidad de integración:** Aunque la movilidad añade el desafío de crear un sistema de navegación, diseñar un chasis propio desde cero simplifica enormemente la integración y cohesión de los componentes que se quisieran utilizar, ya que no tienen que obligatoriamente adaptarse a un sistema pre existente.
2. **Mayor Atractivo e Interacción:** Se determinó que un robot capaz de moverse por la habitación y reaccionar físicamente a su entorno sería percibido por el público como un dispositivo más dinámico y, en definitiva, más atractivo y cercano como "compañero", cumpliendo así el objetivo inicial del proyecto de una forma más efectiva.

4.1.1. Módulos del prototipo

Una vez la idea general definida, fue fundamental realizar un análisis de los componentes necesarios para el funcionamiento del prototipo. Para un desarrollo en paralelo y modular, se dividió el robot en dos módulos principales e inicialmente independientes entre sí, con el objetivo de ser conectados una vez cada uno estuviese lo suficientemente avanzado:

1. **Movimiento:** Este módulo englobaría toda la locomoción física del robot. Se compondría de un chasis estructural, un sistema de ruedas impulsadas por motores y un sensor ultrasónico para la detección de obstáculos. Para controlar el movimiento se

escogió el microcontrolador Arduino Uno, debido a su simplicidad de programación y su disponibilidad en el departamento.

2. **Comunicacion:** Modulo en donde residiría todo lo relacionado con la comunicacion con el usuario. Se determinó inicialmente el levantar localmente una LLM en un Raspberry PI 5, debido a su diseño compacto y potencia. Sin embargo, esta idea fue descartada rapidamente debido a que los modelos utilizados representaban una carga muy grande para el dispositivo. En ella además se conectarían los dispositivos de entrada y salida. Se determinó que la forma de comunicación más cercana y amigable sería a través de voz. Como solución al problema de la potencia, se decidió utilizar una API local para la comunicacion con la IA, utilizando el Raspberry PI como un intermediario entre el usuario y el modelo de lenguaje alojado en otra maquina utilizada como servidor.

4.1.2. Diseño

Una vez la funcionalidad determinada, se analizó la presentación visual del prototipo. Por las características de los componentes y los recursos limitados, se necesitaba una carcasa ligera y a su misma vez accesible. Se optó entonces por crear o buscar un modelo para imprimir en 3D, lo que entonces implicó que el diseño además debiese ser simple para evitar problemas con el filamento o de ensamblaje.

Finalmente se optó por una carcasa completamente cúbica, ya que no solo resulta sumamente fácil de ensamblar, sino que además le da al robot una apariencia agradable. Este diseño entonces se bosquejó, para luego ser trabajado y adaptado en un modelo 3D por Nicolás Poblete, quien durante todo el transcurso del proyecto prestó apoyo en lo que es diseño e impresión del prototipo.

4.2. Movimiento

Para el movimiento del robot se tomó como base un proyecto previo, el cual consistía en un auto a control remoto por bluetooth. Este contenía dos motores, 4 ruedas, un servo para las ruedas delanteras, un Arduino Uno para controlar los motores, baterías y un puente H L293D, componentes que fueron reutilizados en el presente prototipo.

4.2.1. Movimiento hacia adelante y hacia atrás

Para lograr realizar un sistema de ruedas y motores que se muevan hacia una dirección fija, se utilizaron los siguientes componentes

- **Arduino Uno:** Microcontrolador encargado de la lógica, dando la señal de que si se debe avanzar hacia adelante o hacia atrás.
- **Puente H L293D:** Chip que funciona como driver para motores. Su función es controlar la velocidad de los motores.
- **2 Motores DC:** Estos motores son los encargados de hacer que las ruedas traseras giren, funcionando a comando del puente H.
- **4 Ruedas**
- **Protoboard y jumpers:**

4.2.2. Detección de obstáculos

4.2.3. Movimiento autónomo

4.3. Comunicacion

Como se establecio previamente en el analisis del problema, se decidio utilizar un raspberry PI como el nucleo del modulo de comunicacion. Este dispositivo se encargaría de recibir la entrada de audio del usuario, enviarla a un modelo de lenguaje alojado en un servidor externo, y luego reproducir la respuesta generada a través de un altavoz. Este flujo de trabajo suena simple, pero en la practica implico una serie de desafíos técnicos y decisiones de diseño que seran detallados en esta sección.

4.3.1. Arquitectura del sistema

El sistema de comunicación se diseñó adaptando una arquitectura cliente-servidor¹. El Raspberry Pi actua como cliente, mientras que una maquina externa con mayor capacidad de procesamiento aloja el modelo de lenguaje y actua como servidor. Esta separación permite que el Raspberry Pi maneje las tareas de entrada y salida de audio, mientras que el servidor se encarga del procesamiento intensivo requerido por el modelo de lenguaje.

En la figura 4.1 se muestra un diagrama de funcionamiento entre los componentes principales del sistema de comunicación.

`client.py` es el script principal que corre en el Raspberry PI. Este se encarga de detectar constantemente si el usuario ha emitido una entrada de voz especifica o Wake Word (en este caso "Hey Bot"). Al detectar esta entrada, el script graba la voz del usuario durante un periodo de tiempo y la envía al servidor a través de una solicitud HTTP POST. Esto se realiza utilizando la libreria `Por`

¹Modelo de aplicación que distribuye las tareas entre los proveedores de recursos o servicios y solicitantes de servicios. [1]

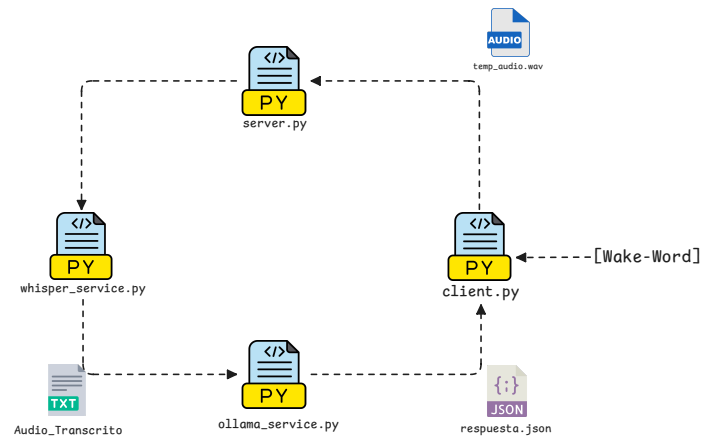


Figura 4.1: Diagrama de comunicacion Servidor-Cliente

4.4. Comunicación Raspberry-Arduino

4.5. Diseño

CONCLUSIÓN

BIBLIOGRAFÍA

- [1] IBM, *Client/server model*, <https://www.ibm.com/docs/en/zos/2.5.0?topic=applications-clientserver-model>, Accessed: 2023-10-27, n.d.