



Tecnológico de Monterrey

**INSTITUTO TECNOLÓGICO Y
DE ESTUDIOS SUPERIORES DE MONTERREY**

Herramientas computacionales: el arte de la programación

Documentación PiCar-Pro

Profesor: Mauricio Paletta Nannarone

Miembros del equipo:

Brisa Sofía León Pérez A01771208

Luis Enrique Bartolo Macario A01773034

Mariana Isabel Aguilar Espíndola A01772501

Esteban Leal Menéndez A01369877

21 de marzo del 2025

Introducción

El objetivo de este proyecto fue desarrollar una librería que facilite la interacción con el robot **Adept PiCar-Pro** mediante comandos intuitivos. Esta librería permite controlar el robot sin necesidad de utilizar comandos complejos, simplificando su operación.

Proceso de Desarrollo

1. Investigación Inicial

Para iniciar el proyecto, identificamos el modelo exacto del robot, **Adept PiCar-Pro**, con el fin de acceder a su documentación y tutoriales oficiales. Descargamos y revisamos los archivos proporcionados, que incluían guías sobre su ensamblaje y programación.

2. Configuración del Entorno

Para configurar la Raspberry Pi del PiCar-Pro:

- Conectamos un monitor para facilitar la configuración gráfica.
- Utilizamos un router prestado para establecer una conexión inalámbrica con la Raspberry Pi.
- Configuramos el router para asignar una dirección IP estática a la Raspberry Pi mediante DHCP.
- Verificamos la conexión a través de **SSH**, asegurando su funcionamiento.

3. Definición de Funcionalidades

Posteriormente, elaboramos un documento con las acciones que el robot debía ejecutar:

Comandos	Descripción
moveArmUp	Mueve el brazo del robot hacia arriba 15 grados
moveArmDown	Mueve el brazo del robot hacia abajo 15 grados
rotateArmRight	Gira el brazo del robot hacia la derecha 15 grados
rotateArmLeft	Gira el brazo del robot hacia la izquierda 15 grados
moveWristUp	Mueve la garra del robot hacia arriba 15 grados
moveWristDown	Mueve la garra del robot hacia abajo 15 grados

openClaw	Abre la garra
closeClaw	Cierra la garra
moveForward	Mueve el robot hacia enfrente durante 1 segundo
moveBackward	Mueve el robot hacia atrás durante 1 segundo
rotateRight	Gira el robot hacia la izquierda durante 1 segundo, 15 grados
rotateLeft	Gira el robot hacia la derecha durante 1 segundo, 15 grados
ledOn	Prende las luces de enfrente
ledOff	Apaga las luces de enfrente

4. Diseño y Desarrollo de la Librería

Inicialmente, diseñamos una estructura basada en varias clases, en la que se accedía a los métodos mediante una jerarquía de módulos. Ejemplo: `PiCar.Chassis.moveBackward()`. Tras recibir retroalimentación del profesor Mauricio, simplificamos la arquitectura para permitir el acceso directo a los métodos: `PiCar.moveBackward()`

5. Implementación y Pruebas

Cada integrante del equipo se encargó de programar una funcionalidad específica. Se utilizó un repositorio para gestionar versiones y registrar avances mediante commits. Al finalizar la implementación, realizamos pruebas para verificar el correcto funcionamiento de todas las funcionalidades. Una vez validado el funcionamiento, integramos el código de los distintos módulos.

6. Refinamiento y Documentación

Para concluir el proyecto:

- Realizamos una prueba final para garantizar la estabilidad del sistema.
- Refinamos el código para cumplir con los estándares de programación.
- Documentamos el código para facilitar su comprensión y mantenimiento.

Áreas de Mejora

Durante el desarrollo del proyecto, enfrentamos diversos desafíos técnicos que requirieron soluciones específicas:

- **Error en la clonación del repositorio:** Inicialmente, el robot no respondía a ningún comando. Identificamos que el problema se debía a un error al clonar el repositorio en **GitHub**. La solución fue volver a clonarlo correctamente, lo que permitió recuperar los archivos base sin inconvenientes.
- **Mal funcionamiento del servo 3:** En un punto, el servo 3 dejó de responder. Detectamos que la causa era una instalación incorrecta de la librería **Adafruit_PCA9685**. Para solucionarlo, reinstalamos la librería correctamente, lo que restauró su funcionamiento.
- **Obstrucción en el mecanismo de la garra:** Aunque la garra del robot podía abrirse y cerrarse, no completaba el movimiento debido a un tornillo excesivamente apretado. La solución consistió en desmontar y volver a ensamblar la pieza, ajustando la presión adecuada con la ayuda del profesor Mauricio.
- **Pérdida de conexión con la Raspberry Pi:** En una ocasión, la Raspberry Pi dejó de responder. Descubrimos que el problema se debía a la insuficiente carga de las baterías, lo que afectaba el rendimiento del sistema. La solución fue sustituir la alimentación por una conexión directa a la corriente eléctrica, restaurando así la conectividad.

README.md

PiCar-Pro

Esta es la documentación para usar la librería de PiCar.py Con las siguientes funciones podrás manipular todas las funciones del robot para completar una tarea en específico.

A considerar:

- Al ejecutar el código del robot, todos los servos vuelven a su posición inicial. Por esta razón todas las instrucciones que requieran ser ejecutadas, deberán ser aplicadas en una sola ejecución.

Instrucciones

1. Prender la Raspberry Pi insertando las pilas en la parte inferior del robot o conectando un cable de alimentación tipo C a la Raspberry Pi.
2. Prender un router y conectar tanto la Raspberry Pi como la computadora que se usará para programar al PiCar a la misma red.
3. Conectar la computadora a la Raspberry Pi mediante el método de preferencia (Putty, SSH). Usuario: car, Contraseña: raspberry. Por SSH sería de la siguiente manera: ssh car@ip-de-la-RaspberryPI -p 22
4. Crear un nuevo archivo .py e importar la librería PiCar localizada en Documents/PiCar-Pro
5. Usar las librerías libremente :)
6. Ejecutar el programa con python3 <nombre_del_programa>.py

Movimiento

Las funciones de movimiento son las siguientes:

- `moveForward()`: El robot se mueve hacia enfrente por 1 segundo
- `moveBackward()`: El robot se mueve hacia atrás por 1 segundo
- `rotateRight()`: El robot se mueve hacia la derecha por 1 segundo
- `rotateLeft()`: El robot se mueve hacia la izquierda por 1 segundo

Brazo

Las funciones del brazo son las siguientes:

- `moveArmUp()`: El robot mueve el brazo 15° hacia arriba
- `moveArmDown()`: El robot mueve el brazo 15° hacia abajo
- `rotateArmRight()`: El robot rota el brazo 15° hacia la derecha
- `rotateArmLeft()`: El robot rota el brazo 15° hacia la izquierda

Garra

Las funciones de la garra son las siguientes:

- `openClaw()`: El robot abre la garra
- `closeClaw()`: El robot cierra la garra
- `moveWristUp()`: El robot mueve la garra 15° hacia arriba
- `moveWristDown()`: El robot mueve la garra 15° hacia abajo

Luces

Las funciones de las luces son las siguientes:

- `ledOn()`: El robot enciende las luces
- `ledOff()`: El robot apaga las luces

URL Github

<https://github.com/TC1001S-2025/PiCar-Pro>