Geniia .com

2025

Curso de Robótica

Visión, control e inteligencia artificial.

Presented By

Ing. Santiago Sanchez

Descripción

del curso

Este curso integral explora el desarrollo de un sistema de brazo robótico inteligente, desde conceptos fundamentales hasta implementaciones avanzadas. Los estudiantes construirán un sistema de brazo robótico completo capaz de escanear el entorno, detectar objetos y realizar manipulaciones autónomas mediante visión artificial, cinemática, control por voz y modelos de lenguaje de gran tamaño.

Nivel 1.

Fundamentos basicos de robotica, y vision artificial.

Modulo 1: Arquitectura y configuración del sistema

- Introducción y descripción general del sistema.
  - Hoja de ruta y objetivos del curso.
  - Componentes de hardware (Raspberry Pi 5, VEX IQ).
  - Fundamentos de la arquitectura maestro-esclavo.
  - Descripción general de los servicios del sistema.

- Diseño del sistema.
  - Fundamentos y arquitectura modular.
  - Diagrama de flujo del sistema.
  - Comunicación entre modulos.

- Diseño de protocolos de comunicación.
  - Protocolos de comunicación.
  - Fundamentos de la comunicación serial.
  - Implementación de la comunicación serial.
  - Diseño de un protocolo robusto basado en JSON.

Descripción

del curso

Nivel 1.

Fundamentos basicos de robotica, y vision artificial.

Modulo 2: Movimiento básico y control

- Fundamentos basicos de movimiento del motor.
  - Fundamentos del control del motor.
  - Sistemas de control conjunto.
  - Secuencias de movimiento.

- Implementacion de servicios básicos de movimiento.
  - Servicio básico de posición segura.
  - Servicio básico de escaneo y perpcepción del entorno.
  - Servicio básico de pick & place.

Modulo 3: Visión artificial básica

- Introducción a la visión artificial.
  - Fundamentos de la visión artificial.
  - Conexión de camaras a la Raspberry PI.

- Fundamentos de la detección de objetos.
  - Introducción a la detección de objetos.
  - Introducción a la arquitectura YOLO.
  - Configuracion de modelos preentrenados.
  - Inferencia basica YOLO.
  - Conversión de modelos.

- Identificación de objetos.
  - Captura de imágenes.
  - Conceptos basicos de procesamiento y visualización.
  - Dibujar y guardar resultados.

Descripción

del curso

Nivel 1.

Fundamentos basicos de robotica, y vision artificial.

Modulo 4: Integración del sistema

- Proyecto final.
  - Integración de todos los servicios.
  - Construcción del sistema básico completo.
  - Demostración de capacidades basicas.
  - Identificacion de fallos y posibles mejoras.
  - Evaluación del desempeño.

Hardware

Vex IQ

Articulaciones y Juntas Cinemáticas.

Las articulaciones en robótica son los puntos de conexión que permiten el movimiento relativo entre dos partes de un robot. Las juntas cinemáticas describen cómo estas articulaciones facilitan el movimiento en términos de grados de libertad (DOF, por sus siglas en inglés), que indican las direcciones independientes en las que un componente puede moverse. En el caso del Armbot IQ, un robot educativo con un brazo robótico, las articulaciones son esenciales para permitir que el brazo y la garra se desplacen y manipulen objetos.

Tipos de Articulaciones:

Articulaciones Rotativas:

Estas permiten la rotación alrededor de un eje. En el Armbot IQ, es probable que se utilicen articulaciones rotativas para movimientos como:

- Girar la base del brazo en el plano horizontal.
- Girar el hombro para extender el brazo.
- Girar el "codo" del brazo para elevar o bajar la garra.

Este tipo de articulación es común en brazos robóticos debido a su simplicidad y versatilidad.

Articulaciones Prismaticas:

Estas permiten el movimiento lineal a lo largo de un eje (como un pistón). Aunque son menos frecuentes en robots pequeños como los de VEX IQ, podrían usarse para extensiones lineales del brazo en diseños más avanzados.

Hardware

Vex IQ

Relación de Engranajes.

Los engranajes son componentes fundamentales en el Armbot IQ, ya que transmiten el movimiento y la fuerza desde los motores a las articulaciones y la garra. La relación de engranajes define cómo se transforman la velocidad y el torque entre el engranaje conductor (conectado al motor) y el engranaje conducido (conectado a la parte móvil).

Cálculo de Relación de Engranajes:

La relación de engranajes se calcula como:
Relación = Número de dientes del engranaje conducido / Número de dientes del engranaje conductor.

- Ejemplo:
  - Si el engranaje conductor tiene 12 dientes y el conducido tiene 36 dientes:
  - Relación = 36 / 12 = 3:1.
  - Esto significa que el engranaje conductor debe girar tres veces para que el engranaje conducido complete una vuelta. Como resultado, la velocidad se reduce en un factor de 3, pero el torque aumenta en la misma proporción (ignorando pérdidas por fricción).

Aplicación en el Armbot IQ:

- Reducción de Velocidad y Aumento de Torque:
  - En el brazo robótico, los engranajes se usan para disminuir la velocidad de los motores y aumentar el torque, permitiendo levantar cargas más pesadas. Por ejemplo, la articulación del codo podría usar una relación de engranajes como 5:1 para proporcionar suficiente fuerza para elevar el brazo y la garra con un objeto.
- Cadena de Engranajes:
  - Es probable que el Armbot IQ emplee una serie de engranajes conectados (una "cadena de engranajes") para lograr la relación deseada entre el motor y la articulación.

Hardware

Vex IQ

Otros Aspectos Mecánicos.

Además de las articulaciones y los engranajes, hay varios elementos mecánicos clave en el diseño del Armbot IQ:

Estructura y Materiales.

- El chasis y las partes del brazo están fabricados en plástico resistente, lo que los hace ligeros pero duraderos.
- Las piezas se ensamblan con pines y conectores, facilitando el montaje y desmontaje, típico de los sistemas modulares VEX IQ.

Mecanismos de Transmisión.

- Engranajes: Son el método principal para transmitir movimiento en VEX IQ.
- Ejes metálicos: Conectan los engranajes y las articulaciones, asegurando rigidez y precisión en la transmisión del movimiento.
- Aunque las correas o cadenas son posibles en robótica, en VEX IQ los engranajes predominan por su simplicidad.

Comunicación

Protocolos de Comunicación

Un protocolo de comunicación es esencialmente un conjunto de reglas que determinan cómo los diferentes componentes de un sistema intercambian información. En nuestro caso, necesitamos que nuestro brazo robótico y nuestra computadora hablen el mismo idioma.
Profundicemos en los principales protocolos utilizados en robótica:

Comunicación Serial:

Transmite datos bit por bit a través de una única línea. Es como un camino de un solo carril donde los datos deben formar una fila y pasar uno tras otro. Es simple pero efectiva para muchas aplicaciones.

Comunicación I2C (Inter-Integrated Circuit):

Utiliza solo dos líneas: una para datos (SDA) y otra para la señal de reloj (SCL). Imagina una mesa redonda de negociaciones donde un moderador (maestro) da la palabra (reloj) a diversos participantes (esclavos) quienes comparten información cuando les toca su turno.

Comunicación SPI (Serial Peripheral Interface):

Utiliza cuatro líneas: MOSI (Master Out, Slave In), MISO (Master In, Slave Out), SCK (Clock) y SS (Slave Select). Es como un sistema de megafonía donde el administrador (maestro) selecciona a qué departamento (esclavo) hablar mediante una línea dedicada, y luego mantienen una conversación bidireccional exclusiva.

Comunicación CAN Bus (Controller Area Network):

Muy utilizado en automoción y robótica industrial. Piénsalo como un sistema de radio donde todos los dispositivos "escuchan" todos los mensajes, pero solo responden a los que llevan su "nombre" o identificador.

Comunicación

Protocolos de Comunicación

Comunicación Ethernet IP:

Para comunicaciones más robustas y a mayor escala. Es como tener una red de carreteras con múltiples carriles donde grandes cantidades de datos pueden viajar simultáneamente siguiendo reglas de tráfico establecidas.

Comunicación

Comunicación Serial.

Fundamentos:

- Piensa en la comunicación serial como enviar un tren de mercancías por una vía única. Cada vagón representa un bit, y todos deben pasar en secuencia ordenada.
- Al principio de cada tren (mensaje), enviamos un vagón especial llamado "bit de inicio" para alertar a la estación receptora.
- Al final, enviamos otro vagón especial, el "bit de parada", que indica "fin del mensaje".
- Si los trenes van demasiado rápido (baudrate incorrecto), la estación receptora no tendrá tiempo de descargar un vagón antes de que llegue el siguiente, creando confusión y pérdida de carga (datos).

Componentes esenciales:

- Puerto: El "nombre" del punto de conexión donde se conecta nuestro dispositivo..
- Baudrate: La velocidad de transmisión medida en bits por segundo. Usamos 115200 bps.
- Bits de datos: Generalmente 8 bits, forman un byte (un carácter).
- Bits de paridad: Opcional, pero útil para detectar errores.
- Bits de parada: Indica el fin de un byte.
- Control de flujo: Mecanismos como RTS/CTS o XON/XOFF que evitan la sobrecarga de datos.
- Timeout: Cuánto tiempo esperamos antes de considerar que la comunicación ha fallado.

Comunicación

Protocolo JSON.

Fundamentos:

Piensa en JSON como un formulario estandarizado universal. Cuando diferentes departamentos gubernamentales necesitan intercambiar información, utilizan formularios con campos predefinidos donde cada campo tiene un nombre claro (la clave) y un espacio para rellenar (el valor). Cualquier funcionario puede leer estos formularios independientemente de su departamento porque siguen una estructura común y organizada.

Componentes esenciales:

- **Estructura clara y legible:** JSON organiza la información en pares clave-valor, similar a un diccionario en Python. Como nuestra analogía del formulario, cada campo tiene un nombre (clave) y un contenido (valor).
- **Flexibilidad en los tipos de datos:** Podemos enviar diferentes tipos de información sin cambiar el formato:
  - **Números enteros (posiciones de motores):** "position": 90
  - **Números decimales (mediciones precisas):** "distance": 24.7
  - **Texto (mensajes de estado):** "status": "SUCCESS"
  - **Listas (secuencias de posiciones):** "waypoints": [10, 45, 90]
  - **Objetos anidados (configuraciones complejas):** "motor_config": {"speed": 50, "acceleration": 10}
- **Compatibilidad universal:** Prácticamente todos los lenguajes de programación tienen bibliotecas para manejar JSON, lo que facilita la comunicación entre diferentes sistemas (Python en la Raspberry, C++ en el VEX Brain).
- **Validación sencilla:** Es fácil verificar si un mensaje JSON tiene la estructura esperada, lo que nos ayuda a detectar errores rápidamente.
- **Tamaño compacto:** Ocupa menos espacio que XML y otros formatos, lo que es importante en comunicaciones seriales con ancho de banda limitado.

Comunicación

Protocolo JSON.

Estructura personalizada:

- Tipo de mensaje (messageType): Identifica qué servicio o acción se está solicitando.
  - Ejemplos: "MOVE_BASE", "MOVE_ARM", "GRAB_OBJECT", "GET_SENSOR"
- Datos (data): Contiene la información necesaria para ejecutar la acción.
  - Para "MOVE_BASE": ángulos de los motores
  - Para "GET_SENSOR": qué sensor consultar
  - Para "GRAB_OBJECT": fuerza de agarre

```json
{
  "messageType": "MOVE_BASE",
  "data": {
    "motorId": 1,
    "position": 90,
    "speed": 50
  }
}
```

Genilia .com

Explicación completa en nuestro canal de YouTube:

¡Click aqui!

Gracias por visitar nuestro repositorio y esperamos que disfrute trabajando con nuestro codigo.