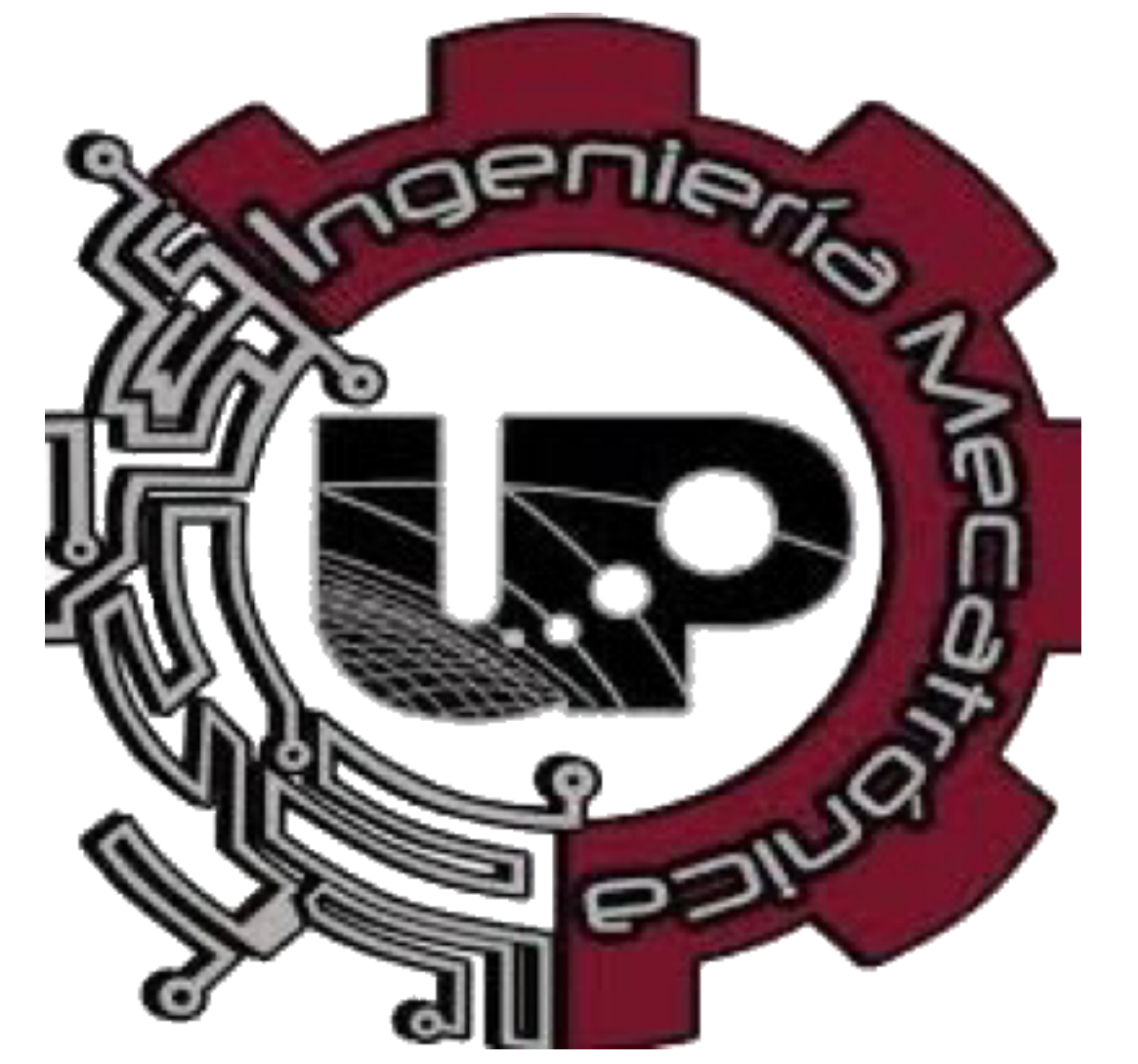




SISTEMA OPERATIVO ROBÓTICO (R O S)



TEVERA-RUIZ A¹, MORALES-COUTIÑO M¹.

1. Departamento de Ingeniería Mecatrónica, Universidad Politécnica de Chiapas, Suchiapa 29150, México

Introducción

Dado el interés incremental en la construcción, diseño y control de robots comerciales y no comerciales, es necesaria la estandarización de su arquitectura para permitir colaboraciones efectivas con el equipo de trabajo, así como desarrollar sistemas complejos a partir de pequeños scripts con base a diversas herramientas y bibliotecas de código abierto para un avance en robótica sustancial en todo el mundo.

Entorno

Aunque su nombre lo indique, ROS es un sistema meta-operativo que permite las principales funcionalidades de un sistema operativo convencional:

- Abstracción de hardware.
- Control de dispositivos de bajo nivel.
- Comunicación entre procesos mediante mensajes.
- Mantenimiento de paquetes.

Su desarrollo está centrado en sistemas GNU/LINUX, principalmente para las distribuciones más reconocidas:

- Ubuntu
- Debian
- Fedora
- Raspbian

Sin embargo, su popularidad ha permitido su adaptación experimental en sistemas operativos más comerciales:

- Mac OS X
- Microsoft Windows

Arquitectura

Está basado en una arquitectura de grafos con los siguientes elementos:

- Nodos - Scripts para el procesamiento de datos para su emisión (nodo publicador) o solicitud (nodo suscriptor).
- Temas - Clasificación de los mensajes.
- Mensajes - Datos para la comunicación unidireccional entre nodos.
- Servicios - Datos para la comunicación bidireccional entre nodos.
- Paquetes - Centralización de nodos para una tarea.
- Stacks - Integración de paquetes para la ejecución de proyectos muy complejos.

REFERENCIA

Documentation - ROS Wiki. (2020). Disponible en <http://wiki.ros.org/>

Aplicaciones

Mediante su versatilidad es sencillo implementar diversas áreas en un mismo robot:

- Visión Artificial
- Inteligencia Artificial
- Simulación y visualización 3D en tiempo real
- Geolocalización
- Control de actuadores
- Planificación de movimiento
- Reconocimiento de voz

Ventajas

En función a la gran variedad de herramientas y librerías, ROS oferta un ambiente de trabajo para proyectos de robótica completo y muy eficiente, permitiendo:

- Desarrollar sistemas de control avanzados mediante la deconstrucción de etapas de funcionamiento.
- Escalabilidad del proyecto.
- Distribución y colaboración adecuada para grandes equipos de trabajo.
- Independencia de lenguaje, permitiendo programar en Python, C++ o Lisp.
- Integración con placas de desarrollo comerciales.

Conclusión

La implementación de ROS en proyectos escolares permite el fortalecimiento de conocimientos adquiridos mediante el desarrollo de proyectos más elaborados considerando una estandarización para cada etapa y un progreso simultáneo en todo el equipo de trabajo.

Además, permite aprovechar las especificaciones de placas educativas, como Arduino, al reducir la carga de procesamiento; enfocándose en la administración de señales.