# **PRUEBAS DE INTEGRACIÓN**

Tabla de contenidos

|  |  |  |
| --- | --- | --- |
| 1 | **Prueba de Integración de Locomoción y Navegación** | …1 |
| 2 | **Prueba de Integración de Interfaz de Usuario y Sistemas de Control Autónomo.** | …2 |
| 3 | **Prueba de Integración de Adaptabilidad y Eficiencia Energética.** | …3 |

1. **Prueba de Comunicación entre Arduino Nano y Raspberry Pi 5**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Comunicación CAN establecida entre Arduino Nano y Raspberry Pi 5. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Arduino Nano. * Raspberry Pi 5. * Módulos CAN msp 2515. |
| **Software requerido** | * Librerías CAN para Arduino. (definir versión) * Librerías CAN para Raspberry Pi. |
| **Objetivo** | Verificar la correcta transmisión y recepción de datos entre el Arduino Nano y el Raspberry Pi 5. |
| Descripción | |
| **Procedimiento** | 1. Conectar el Arduino Nano y el Raspberry Pi 5 a través de los módulos CAN. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS2 HUMBLE en el Raspberry Pi 5y monitorear la comunicación. 4. Enviar y recibir datos de prueba para verificar la comunicación (definir mensaje de prueba con CAN). |
| **Resultado esperado** | Datos enviados desde el Arduino Nano son correctamente recibidos por el Raspberry Pi 5 y viceversa, sin pérdida de información. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Comunicación SSH para Tele operación**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Conexión SSH establecida y funcional. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Computadora con capacidad de conexión SSH. * Raspberry Pi 5 con conexión a red. |
| **Software requerido** | * Cliente SSH en la computadora. * Servidor SSH en el Raspberry Pi 5. |
| **Objetivo** | Asegurar que se pueda controlar el robot de forma remota mediante SSH. |
| Descripción | |
| **Procedimiento** | 1. Configurar el servidor SSH en el Raspberry Pi. 2. Establecer una conexión SSH desde la computadora a través de la red. 3. Ejecutar comandos remotos en el Raspberry Pi para controlar el robot. 4. Verificar la respuesta del robot a los comandos. |
| **Resultado esperado** | Establecimiento exitoso de la conexión SSH y ejecución de comandos remotos en el Raspberry Pi 5 sin retrasos significativos (revisar como medir tiempos de respuesta). |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Integración de Sensores de Posición**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Sensores de posición instalados y calibrados correctamente. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Sensor inercial mpu 9250 * Dynamixel AX-12+ * 74LS241 * Arduino Nano. * Raspberry Pi 5. |
| **Software requerido** | * Librerías de sensores para Arduino y Raspberry Pi (definir versión). * Algoritmo de procesamiento de datos de sensores. * Nodos ROS para integración de sensores. |
| **Objetivo** | Verificar que los sensores de posición proporcionen datos precisos y coherentes para el control del robot. |
| Descripción | |
| **Procedimiento** | 1. Instalar y conectar los sensores de posición al Arduino y Raspberry Pi. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y monitorear los datos de los sensores. 4. Calibrar los sensores y verificar la precisión de los datos. |
| **Resultado esperado** | Los datos de los sensores de posición se leen correctamente y se utilizan para ajustar la postura y movimientos del robot. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Integración de Motores y Controladores**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Motores Dynamixel AX-12A y controladores instalados y configurados correctamente. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Motores Dynamixel AX-12A+. * 74LS241. * Arduino Nano. * Raspberry Pi 5. |
| **Software requerido** | * Librerías de control de motores Dynamixel(Definir version). * Algoritmo de control de movimiento. * Nodos ROS para control de motores Dynamixel. |
| **Objetivo** | Verificar que los motores Dynamixel y sus controladores funcionen correctamente y respondan a los comandos de movimiento. |
| Descripción | |
| **Procedimiento** | 1. Instalar y conectar los motores Dynamixel y controladores al Arduino y Raspberry Pi. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y enviar comandos de control a los motores. 4. Monitorear y verificar los movimientos de los motores. |
| **Resultado esperado** | Los motores responden de manera precisa a los comandos de control, permitiendo movimientos suaves y controlados del robot. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Integración de una Pata**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Motores Dynamixel AX-12A y sensores en una pata instalados y configurados correctamente. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Una pata del robot equipada con 3 motores Dynamixel AX-12A * Arduino Nano. * 74LS241. |
| **Software requerido** | * Librerías de control de motores Dynamixel.(definir librerias) * Algoritmo de control para una pata. * Nodos ROS para control de Dynamixel. |
| **Objetivo** | Verificar que una pata del robot pueda moverse de manera precisa y responder a los comandos de control. |
| Descripción | |
| **Procedimiento** | 1. Instalar y conectar los motores Dynamixel y sensores en una pata del robot. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y enviar comandos de control a la pata. 4. Monitorear y verificar los movimientos de la pata. |
| **Resultado esperado** | La pata se mueve de acuerdo con los comandos de control, respondiendo con precisión a las instrucciones de levantamiento, movimiento y posicionamiento. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Integración con las Cuatro Patas**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Todas las patas del robot equipadas con motores Dynamixel AX-12A y sensores, instalados y configurados correctamente. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Robot completo con las cuatro patas equipadas 12 motores Dynamixel AX-12A+. * Arduino Nano. * Raspberry Pi 5. * Controladores de motores Dynamixel. |
| **Software requerido** | * Librerías de control de motores Dynamixel. * Algoritmo de control para cuatro patas. * Nodos ROS para control de cuatro patas. |
| **Objetivo** | Verificar que las cuatro patas del robot puedan moverse de manera sincronizada y responder a los comandos de control para realizar movimientos complejos. |
| Descripción | |
| **Procedimiento** | 1. Instalar y conectar las cuatro patas equipadas con motores Dynamixel y sensores al Arduino y Raspberry Pi. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y enviar comandos de control a las cuatro patas. 4. Monitorear y verificar los movimientos coordinados de las patas. |
| **Resultado esperado** | Las cuatro patas se mueven de manera coordinada, permitiendo al robot realizar movimientos complejos como caminar, girar y subir obstáculos. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Configuración de Locomoción para HOME**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Algoritmos de locomoción para retornar a HOME implementados. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Robot cuadrúpedo completo. * Sensor inercial mpu 9250. |
| **Software requerido** | * Algoritmo de control de locomoción. * Software de visualización para monitorear la posición. * Nodos ROS para control de locomoción. |
| **Objetivo** | Verificar que el robot pueda retornar a la posición HOME de forma precisa. |
| Descripción | |
| **Procedimiento** | 1. Implementar los algoritmos de locomoción para retornar a HOME en el sistema de control del robot. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y enviar comandos de retorno a HOME. 4. Monitorear y verificar que el robot retorne a la posición HOME de manera precisa. |
| **Resultado esperado** | El robot vuelve a la posición HOME de manera consistente y precisa. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Locomoción para Levantarse**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Algoritmos de levantamiento implementados. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Robot cuadrúpedo completo. * Sensor inercial mpu 9250. |
| **Software requerido** | * Algoritmo de control de levantamiento. * Software de visualización para monitorear el estado del robot. * Nodos ROS para control de levantamiento. |
| **Objetivo** | Verificar que el robot pueda levantarse desde una posición HOME |
| Descripción | |
| **Procedimiento** | 1. Implementar los algoritmos de levantamiento en el sistema de control del robot. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y enviar comandos de levantamiento. 4. Monitorear y verificar que el robot se levante de manera estable. |
| **Resultado esperado** | El robot se levanta de manera estable y sin caerse nuevamente. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Marcha Estática**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Algoritmos de marcha estática implementados. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Robot cuadrúpedo completo. * Sensor inercial mpu 9250. |
| **Software requerido** | * Algoritmo de control de marcha estática. * Software de visualización para monitorear la postura. * Nodos ROS para control de marcha estática. |
| **Objetivo** | Verificar que el robot pueda mantener una postura estática sin moverse. |
| Descripción | |
| **Procedimiento** | 1. Implementar los algoritmos de marcha estática en el sistema de control del robot. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y enviar comandos de marcha estática. 4. Monitorear y verificar que el robot mantenga la postura estática sin moverse. |
| **Resultado esperado** | El robot mantiene una postura estática durante el tiempo especificado sin desviarse. |
| **Resultado obtenido** |  |
| **Comentarios** |  |

1. **Prueba de Locomoción para Movimiento**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**  Algoritmos de locomoción para movimiento implementados. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Robot cuadrúpedo completo. * Sensor inercial mpu 9250. |
| **Software requerido** | * Algoritmo de control de locomoción para movimiento. * Software de visualización para monitorear el movimiento. * Nodos ROS para control de locomoción. |
| **Objetivo** | Verificar que el robot pueda moverse en todas las direcciones de manera precisa y controlada. |
| Descripción | |
| **Procedimiento** | 1. Implementar los algoritmos de locomoción para movimiento en el sistema de control del robot. 2. Configurar y cargar los códigos de prueba en Arduino y Raspberry Pi. 3. Ejecutar los nodos ROS correspondientes y enviar comandos de movimiento. 4. Monitorear y verificar que el robot se mueva en las direcciones deseadas de manera precisa. |
| **Resultado esperado** | El robot se mueve en las direcciones deseadas (adelante, atrás, izquierda, derecha) sin desviaciones significativas. |
| **Resultado obtenido** |  |
| **Comentarios** |  |