# **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 Integración sensor inercial y sensor de gas**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos: Requerimientos**   * La red debe ser capaz de medir los distintos gases que mide el sensor mq4 * La red debe ser capaz de medir la orientación de la plataforma utilizando un sensor inercial. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Módulos CAN * Arduino Nano 1 * Arduino Nano master * Sensor inercial * Sensor de gas MQ4 |
| **Software requerido** | * Sistema operativo para Raspberry Pi (Ubuntu) * IDE de Arduino * Librerías de sensores y comunicación. |
| **Objetivo** | * Verificar la correcta integración y funcionamiento del sensor inercial y el sensor de gas |
| Descripción | |
| **Procedimiento** | * Conectar el sensor inercial y el sensor MQ4 al arduino Nano * Conectar los arduinos entre sí a través de los modulos CAN * Configurar los códigos para el funcionamiento de estos dos sensores y transmitir la información * Ejecutar los códigos y verificar el correcto funcionamiento de este |
| **Resultado esperado** | * Los datos del sensor de gas se transmiten de manera correcta * Los datos del sensor inercial se transmiten correctamente |
| **Resultado obtenido** | * Los datos de el sensor inercial y el sensor de gas se enviaron y se recibieron correctamente   \*inserte foto aquí\* |
| **Comentarios** | * Se debe prestar mucha atención a la velocidad de envío de los datos debido que si esta se encuentra en una en la que no se logran sincronizar los datos de todos los sensores estos no serán capaces de llegar. |

1. **Prueba de Sistema Completa con Dos Esclavos y un Maestro**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**   * La red debe permitir la comunicación entre todos los dispositivos y sensores. * La plataforma debe permitir la comunicación entre múltiples Arduinos. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Arduino Nano Master * Arduino Nano 1 (Esclavo) * Arduino Nano 2 (Esclavo) * Otros sensores conectados a los Arduinos * Módulos CAN |
| **Software requerido** | * IDE de Arduino * Librerías de comunicación y sensores. |
| **Objetivo** | * Verificar el funcionamiento del sistema completo con un Arduino Nano Master, tres módulos CAN y dos Arduino Nano esclavos. |
| Descripción | |
| **Procedimiento** | * Ejecutar el código en los Arduino Nano para manejar las solicitudes y respuestas de los sensores. * Verificar la comunicación entre los Arduino Nano esclavos y el maestro. * Ejecutar el código y verificar que todos los datos se muestren correctamente. |
| **Resultado esperado** | * Todos los sensores envían datos correctos al maestro * La comunicación entre todos los dispositivos es correcta y sin errores. |
| **Resultado obtenido** | * Los datos correspondientes a los sensores se enviaron correctamente al maestro   \*inserte foto aquí\* |
| **Comentarios** | * Los tiempos de respuesta entre los dos esclavos y el maestro se ve condicionada por el tiempo de respuesta de algunos sensores |

1. **Prueba de Integración : Cámara de Visión Nocturna, Lidar 2D y Micrófono**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN** | |
| **Requerimientos**   * La red debe ser capaz de detectar objetos mediante una cámara de visión nocturna. * La red debe ser capaz de detectar objetos mediante un LiDar 2D. * La red debe generar comandos en respuesta a una entrada de voz. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Raspberry Pi * Cámara de Visión Nocturna * LiDar 2D * Micrófono USB |
| **Software requerido** | * Sistema operativo para Raspberry Pi (Ubuntu) * Librerías de visión, procesamiento de imagen, LiDar y reconocimiento de voz. |
| **Objetivo** | * Verificar la correcta integración y funcionamiento de la cámara de visión nocturna, LiDar 2D, y micrófono con la Raspberry Pi. |
| Descripción | |
| **Procedimiento** | * Conectar la cámara de visión nocturna, LiDar 2D y micrófono USB a la Raspberry Pi. * Configurar el código en la Raspberry Pi para manejar cada uno de los dispositivos. * Enviar una solicitud de detección de objetos a la cámara de visión nocturna y al LiDar. * Activar el micrófono y enviar una entrada de voz. * Verificar que los datos de la cámara, LiDar y micrófono se procesen y se muestren correctamente. |
| **Resultado esperado** | * Las imágenes capturadas por la cámara de visión nocturna se muestran correctamente. * Los datos del LiDar se procesan y se muestran correctamente. * Los comandos generados en respuesta a la entrada de voz se muestran correctamente. |
| **Resultado obtenido** | * Se logro una correcta integración de los tres sensores a través del midware ros2 en la raspberry pi 5, logrando lanzar todos los nodos de cámara, lidar y micrófono tanto por separado en nodos como también en un launch que ejecuta todos a la vez correctamente |
| **Comentarios** | Ajustar la forma en que muestra los datos que recibe el suscriptor del launch para mostrar los datos |

1. **Prueba de Integración de comunicación en tiempo real de sensores**

|  |  |
| --- | --- |
| **PRUEBA DE INTEGRACIÓN EN TIEMPO REAL** | |
| **Requerimientos**   * La plataforma debe permitir la comunicación en tiempo real de los sensores. | |
| **Tipo de prueba** | Integración |
| **Hardware requerido** | * Raspberry Pi * Arduino Nano * Modulo CAN * Sensores |
| **Software requerido** | * Sistema operativo para Raspberry Pi (ubuntu) * IDE de Arduino * Librerías de comunicación (Serial, I2C, etc.) |
| **Objetivo** | * Verificar la comunicación en tiempo real de los sensores entre el Arduino Nano y la Raspberry Pi. |
| Descripción | |
| **Procedimiento** | * Conectar la Raspberry Pi a Arduino Nano a través de un enlace de comunicación * Conectar los Arduinos Nano con la comunicación CAN. * Conectar sensores al Arduino Nano. * Configurar el código en la Raspberry Pi para enviar una señal de inicio a Arduino Nano. * Configurar el código en Arduino Nano para solicitar datos en tiempo real de los sensores y enviarlos a Raspberry Pi. * Ejecutar el código y verificar que los datos de los sensores se muestren en la Raspberry Pi en tiempo real. |
| **Resultado esperado** | * La Raspberry Pi recibe y muestra los datos de los sensores en tiempo real. * No hay pérdida de datos ni retrasos significativos en la transmisión. |
| **Resultado obtenido** | * Se realizo la implementación de la lectura de los sensores de la red can y llegan al master pero debido a que toca limitar el cada tiempo que envia las tramas para que lleguen todos los sensores hay algunas tramas que se repiten varias veces al enviar los datos |
| **Comentarios** | * Ajustar la forma en la que esta enviando y recibiendo las tramas de datos y ajustar los tiempos de lectura |