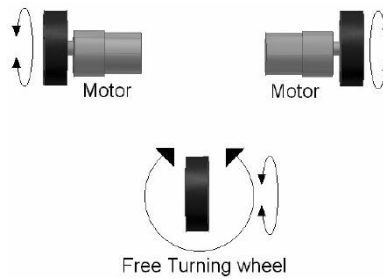


Sistema de equilibrio de un robot móvil

Una unidad de medición inercial (IMU) es un dispositivo electrónico que mide velocidad, orientación y fuerzas gravitacionales usando una combinación de acelerómetros, giroscopios y magnetómetros. Un IMU tiene muchas aplicaciones en robótica; algunos de las aplicaciones están en equilibrio de vehículos aéreos no tripulados (UAV) y navegación de robots.

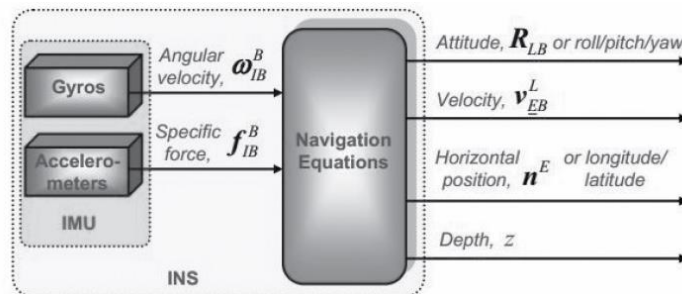
Para este segundo diseño, se requiere implementar un IMU para la navegación de un robot móvil de tipo diferencial. El robot que se requiere diseñar es un robot con ruedas diferenciales, donde el movimiento se basa en dos ruedas accionadas por separado colocadas en ambos lados del cuerpo del robot. Puede cambiar su dirección cambiando la velocidad relativa de rotación de sus ruedas, y por lo tanto, no requiere un movimiento de dirección adicional. Para equilibrar el robot, se puede agregar una rueda giratoria libre o ruedas giratorias.



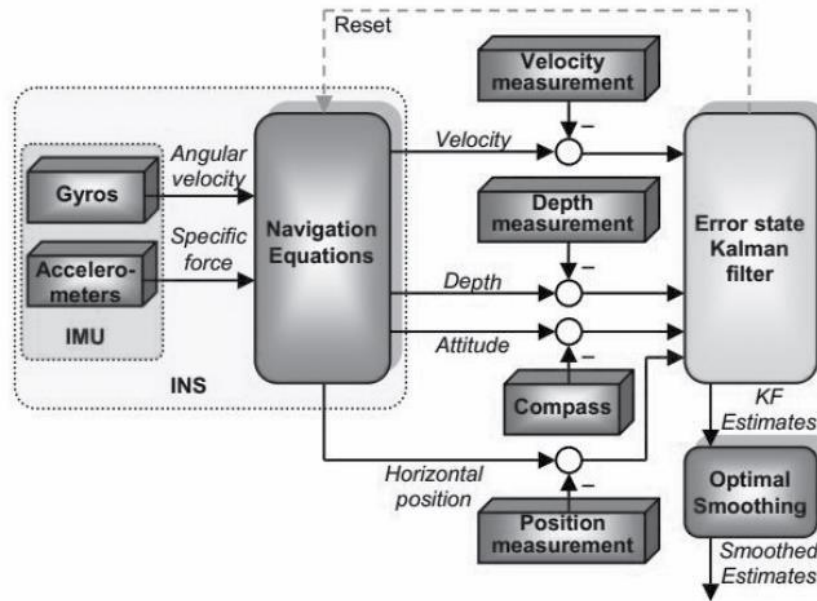
Navegación inercial

Una IMU proporciona aceleración y orientación relativa al espacio inercial, si conoce la posición inicial, la velocidad y la orientación, puede calcular la velocidad integrando la aceleración detectada y la segunda integración da la posición. Para obtener la dirección correcta del robot, se requiere la orientación del robot; esto se obtendrá integrando la velocidad angular detectada del giroscopio.

La siguiente figura ilustra un sistema de navegación inercial, que convertirá los valores IMU a datos odométricos:



Los valores que obtenemos del IMU se convierten en información de navegación usando ecuaciones de navegación y alimentación en filtros de estimación como el filtro de Kalman. Este tipo de filtro usa un algoritmo que estima el estado de un sistema del datos medidos (http://en.wikipedia.org/wiki/Kalman_filter). Los datos del sistema de navegación inercial (INS) tendrá cierta deriva debido al error del acelerómetro y el giroscopio. Para limitar la deriva generalmente es necesario otros sensores que proporcionan mediciones directas de las cantidades integradas. Basado en mediciones y modelos de error del sensor, el filtro de Kalman estima errores en las ecuaciones de navegación. La siguiente figura muestra un diagrama de un sistema de navegación inercial asistida que utiliza el filtro de Kalman:



Junto con los codificadores del motor, el valor de la IMU se puede tomar como el valor del odómetro y se puede utilizar para estimar la posición actual de un objeto en movimiento usando una posición previamente determinada.

Interfaz entre MPU 6050 y el microprocesador

La familia de piezas MPU-6000 / MPU-6050 están diseñados para un bajo consumo de potencia, bajo costo y alto rendimiento. Actualmente se usan en teléfonos inteligentes, tabletas, sensores portátiles y robótica. Los dispositivos MPU-6000/6050 combinan un giroscopio de 3 ejes y un acelerómetro de 3 ejes.



El MPU 6050 se puede conseguir en el siguiente enlace:

<https://www.sparkfun.com/products/11028>

Fase de análisis

Seguridad. Ningún riesgo para el humano, una vez inhabilitado el robot móvil se debe deshacer como desechos electrónicos.

Precisión. El sistema de posicionamiento y la unidad de medición inercial debe ser de precisión media.

Exactitud. Las mediciones deben ser confiables.

Resolución. La resolución debe ser adecuada para un sistema de edometría experimental.

Tiempo de respuesta. En tiempo real.

Ancho de Banda. Entre más rápido mejor, con limite en la transmisión de datos.

Mantenibilidad. Se deben usar dispositivos genéricos y de preferencia en lenguaje de alto nivel.

Testabilidad. Las pruebas deben considerar el despliegue de los datos en terminal.

Compatibilidad. Se debe ajustar a los estándares existentes.

Tiempo medio entre fallas. Debe diseñarse para una vida útil de por lo menos 5 años.

Tamaño y Peso. 20*20 cm² max. Y 1.5 kg max.

Energía. 12 v max, y 3.3 para las I/O digitales.

Costo de ingeniería no recurrente. \$ 1,250.00 max.

Costo unitario. \$ 300.00 aprox.

Tiempo del prototipo. 60 días max.

Tiempo de comercialización. No aplica.

Factores Humanos. 30 aprox.

Documento de Requisitos.

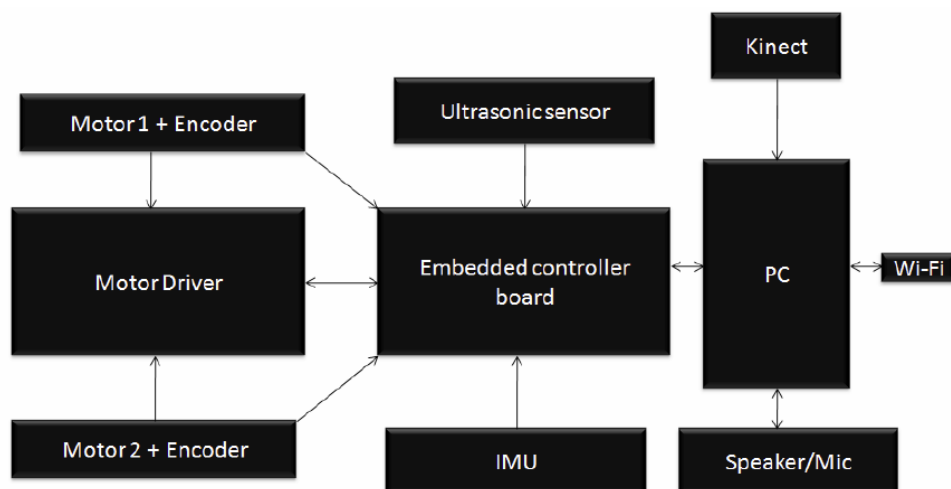
Información general.

Proyecto de robótica de enjambre, FES Aragón, UNAM. 2018

Objetivos.

Disenar y construir un sistema de posicionamiento a través de una unidad de medición inercial (IMU), para un robot móvil inteligente para recolectar información de su entorno.

Proceso.



Refinamiento sucesivo.

¿Qué significa estar en un estado? Enumerar los parámetros del estado

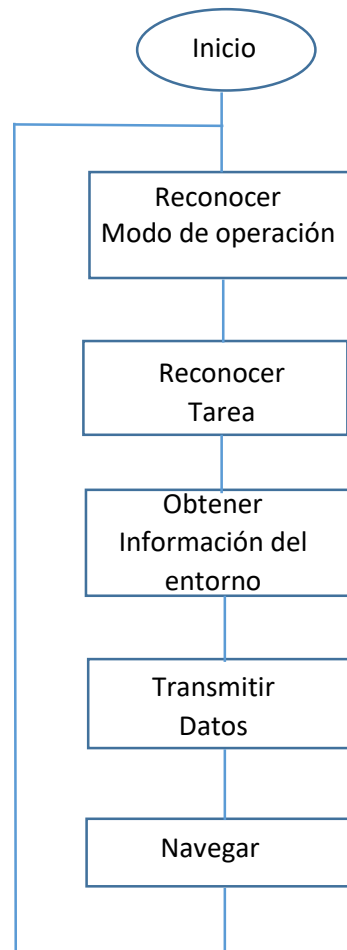
¿Cuál es el estado inicial del sistema? Definir el estado inicial

¿Qué información necesitamos recolectar? Enumerar los datos de entrada

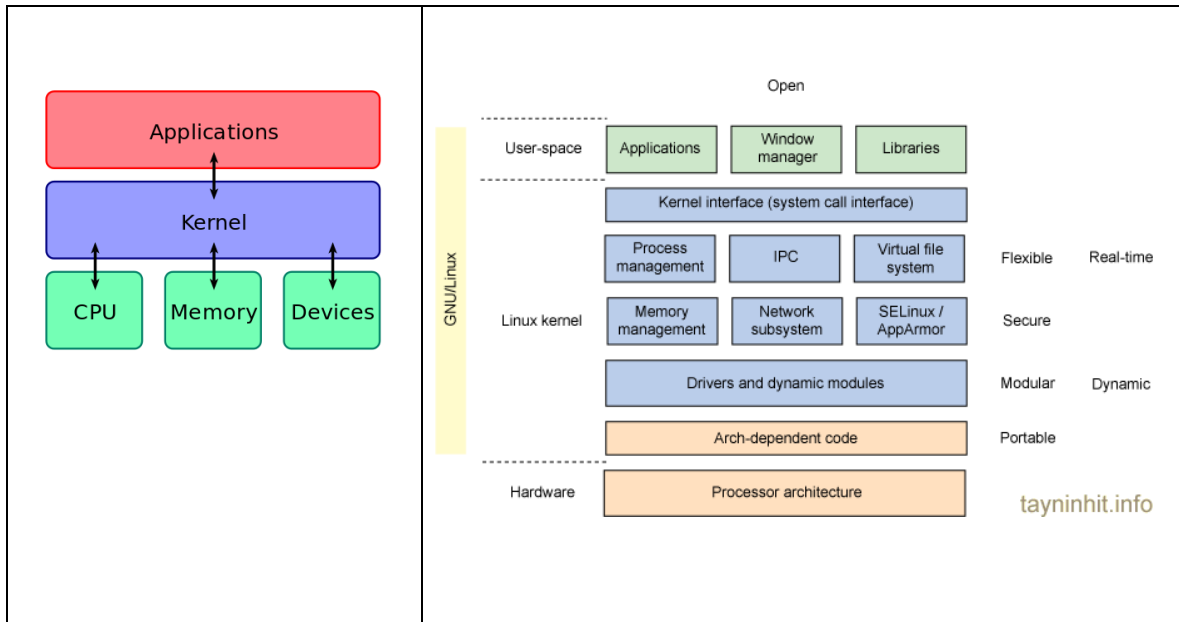
¿Qué información necesitamos generar? Enumerar los datos de salida

¿Cómo pasamos de un estado a otro? Especificar acciones que podríamos realizar

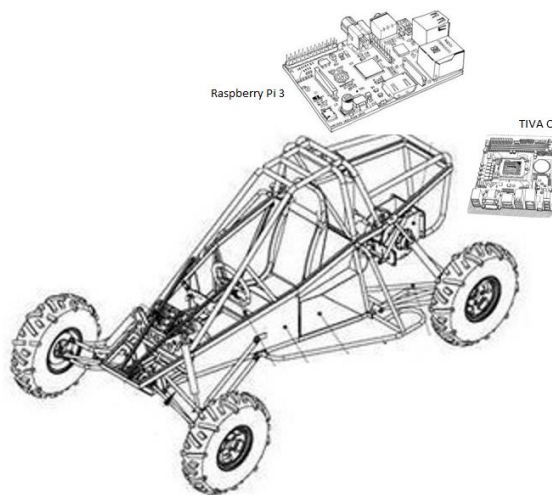
¿Cuál es el estado final deseado? Definir el objetivo final



Tipos de Sistemas Embebidos

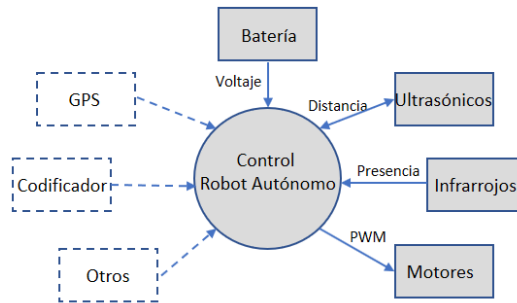


Para este diseño se propone utilizar ambos tipos de sistemas embebidos. La primera arquitectura solo tiene como objetivo hacer el control del sistema de navegación del robot y posteriormente será parte de una aplicación en un sistema de alto rendimiento como Raspberry pi con raspbian o un dispositivo móvil con Android.



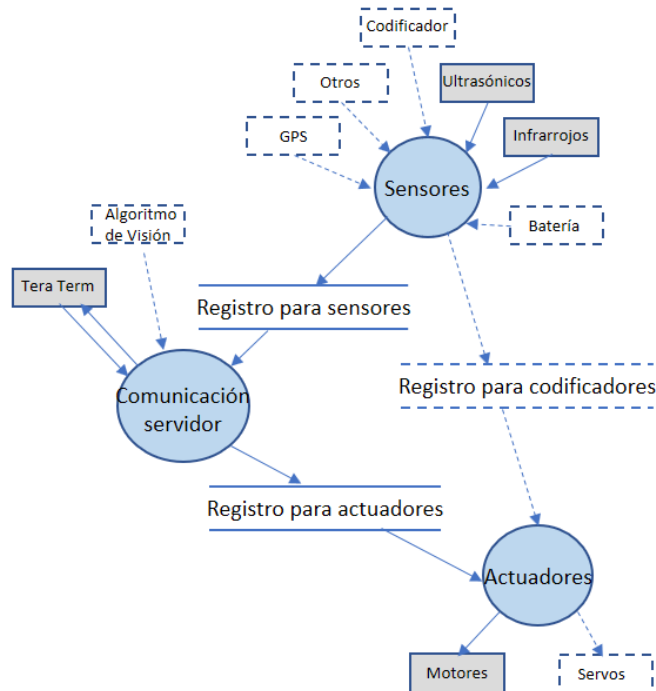
Sistema de detección de obstáculos.

Una navegación ideal significa que un robot puede planificar su trayectoria desde su posición actual hasta el destino y puede moverse sin ningún obstáculo. Estamos usando sensores ultrasónicos para encontrar la distancia a un objeto, y para obtener los datos de odometría del robot; también usamos sensores de proximidad IR (rayos infrarrojos) para detectar los obstáculos y evitar colisiones.



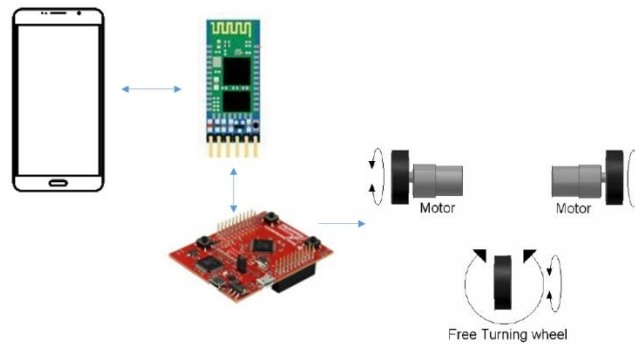
El primer paso del desarrollo de firmware es el diseño de la interacción entre componentes, que se puede observar en el diagrama de contexto de la figura anterior, usando este diagrama se pueden observar las conexiones entre el programa de control y los otros componentes, así como la dirección del flujo de datos. Como se puede ver, la aplicación tiene que procesar el trabajo de muchos sensores, algunos necesitan comunicación bidireccional para hacer la inicialización, medición y lectura de datos; y en otros, sólo se trata de leer los valores. La siguiente tarea del control es activar los actuadores y la comunicación con un sistema superior en caso de ser requerida.

El siguiente diagrama, es considerado como el primer nivel de flujo de datos en la aplicación de control, y proporciona una mirada más cercana al objetivo específico de este trabajo. Describe la división básica en las tres partes (procesos: Sensores, Actuadores y Comunicación con el servidor) y sus conexiones. Estos procesos estarán trabajando independientemente durante la mayor parte del tiempo. Por lo tanto, tiene que haber cuellos de botella entre ellos. Usando la programación jerárquica la información de los sensores se guarda en un registro de memoria (Registro para sensores), que será compartida, y el proceso de comunicación con el servidor podrá leer estos datos. De igual forma se guardó en un registro de memoria para los codificadores y actuadores, logrando de esta forma triangular la información para un control más eficaz.



Primera Etapa – Comunicación.

El sistema de comunicación Bluetooth es un protocolo de comunicación que se ha desarrollado para la comunicación inalámbrica de dispositivos de bajo consumo. EL modulo Bluetooth **HC-05** viene configurado de fábrica como Esclavo, pero se puede cambiar para que trabaje como maestro, además al igual que el hc-06, se puede cambiar el nombre, código de vinculación velocidad y otros parámetros más.



Aunque existen muchas Apps que conectan el módulo **HC-05**, se sugiere utilizar [ApplInventor](#) para hacer un diseño que pueda ser modificado, o en el mejor de los casos programar en Java la aplicación con [Android Studio](#).

Una vez terminada esta la etapa de comunicación el sistema está listo para *Reconocer el Modo de operación*:

- 0 – Manual
- 1 – Automático

Segunda Etapa – Sensores.

El robot móvil deberá contar con varios sensores para determinar la presencia de un obstáculo y la distancia a la que se encuentra. Así como los valores x, y, z de la orientación relativa del IMU para su navegación inercial.

Se puede tomar como referencia el [capítulo 6](#) del libro: Learning Robotics Using Python, Letin Joseph.