

Percepción (Inicio)



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN



Manual para construir un robot de enjambre.

Percepción (PAPIME PE109416)

Este manual forma parte de una serie de documentos diseñados para guiar a los participantes del Taller de Robótica de Enjambre: Nivel 2 - PERCEPCIÓN. Este segundo taller práctico tiene como meta programar un microcontrolador para controlar la dirección y velocidad de motores.



Manual para construir un robot de enjambre.

Percepción (PAPIME PE109416)

Definición de Robot.

En su muy agradable libro introductorio sobre el tema de la robótica llamado *The Robotics Primer*, Maja J. Mataric [1] utiliza la siguiente definición: "Un robot es un sistema autónomo que existe en el mundo físico, puede detectar su ambiente, y puede actuar para lograr algunos objetivos ". La primera parte dice: "Un robot es un sistema autónomo". Por autónomo, queremos decir que un robot toma decisiones por sí mismo; no está controlado por un humano. y elimina cualquier máquina que esté controlada por alguien (a diferencia por ejemplo de una lavadora).

Los Robots de los que hablaremos a lo largo de este manual a veces pueden tener algún tipo de función remota, que permite a un ser humano controlarlo a distancia, pero esta funcionalidad generalmente está incorporada como una especie de medida de seguridad para diagnosticar un posible problema. Sin embargo, el objetivo principal es construir robots que puedan ser capaces de actuar y funcionar por su cuenta.



Fig. 3.1 Un robot puede tomar decisiones por sí mismo.

Continuando con la definición, necesitamos la otra parte de la oración que dice "que existe en el mundo físico", para diferenciarlos de los llamamos "robots de software" o simplemente "bots", por lo que un robot real necesita saber qué está sucediendo en el entorno en que opera. Además, esta es la razón por la cual la siguiente parte de la definición dice, "puede sentir su entorno".

Introducción al diseño con Microcontroladores

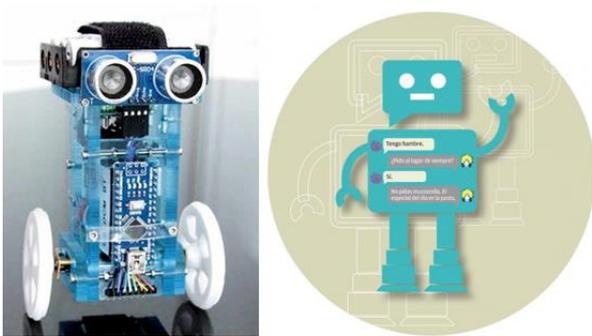


Este proyecto tiene como objetivo hacer un manual para el diseño de robots de enjambre usando hardware informático de código abierto para microcontroladores de 32 bits con arquitectura ARM (**Advanced RISC Machines**)

En la actualidad, los robots de código abierto aún no son lo suficientemente sofisticados para llevar a cabo gran parte de las tareas de una persona, sin embargo, las estrategias de visión y la computación evolutiva es una oportunidad de desarrollo para los nuevos ingenieros que usan las herramientas tecnológicas para generar un perfil profesional que los hace valiosos en cualquier compañía.



Detectar lo que está sucediendo alrededor de un robot real es posiblemente la más importante característica. Para detectar sus entornos circundantes, los robots generalmente tienen sensores. Estos son dispositivos que miden las características físicas del ambiente y proporcionar esta información de vuelta al robot para que pueda, por ejemplo, reaccionar ante un ataque repentino cambios de temperatura, humedad o presión. Si miramos las semejanzas entre los robots y los humanos, podemos pensar en subsistemas sensores como reemplazos artificiales para órganos humanos que proporcionan información para el cerebro.



Una consecuencia importante de esta definición es que cualquier cosa que no tenga sentido en su entorno no se puede llamar un robot. Esto incluye cualquier dispositivo que simplemente conduzca a ciegas o moverse de forma aleatoria, porque no utilizan la información del entorno en el que puede basar su comportamiento. Nuestra definición de robot refleja esto en su última parte cuando dice, "*puede actuar en consecuencia para lograr algunos objetivos*".

Fig. 3.2 Bots vs Robots.

Actuar sobre el medio ambiente puede sonar como una tarea muy compleja para un robot, pero en este caso, solo significa cambiar el mundo de alguna manera (incluso muy leve). Llamando estas partes del robot como efectores, y haciendo una similitud con el ser humano su equivalente artificial podrían ser las manos, piernas y otras partes del cuerpo que le permiten moverse. Los efectores hacen uso de algunos sistemas de nivel inferior como como motores o músculos que realmente llevan a cabo el movimiento.

Aunque los actuadores artificiales parecen funcionar de forma similar a los biológicos, una mirada más cercana revelará que en realidad son bastante diferentes, ya que no solo se trata de que el robot actúa sobre el medio ambiente, sino también sobre el logro de algunos objetivos. La mayoría de los robots se construyen para llevar a cabo (o, preferimos decir, para ayudar con) algunas tareas, como mover piezas pesadas en una fábrica o localizar víctimas en áreas afectadas por desastres naturales.

Sin embargo, el objetivo de un robot no necesariamente tiene que ser algo tan complejo y ambicioso como "realizar el trabajo duro para humanos". Puede ser algo simple, como "no toparse con obstáculos" o "enciende el interruptor de la luz".



Fig. 3.3 Robots que cumplen una tarea.



El uso de los microprocesadores y microcontroladores en la robótica.

En los módulos 1 y 2 de este manual se hace referencia al diseño y construcción de la parte física del robot. Y ahora analizaremos la parte de la percepción del robot, si no fuera por los controladores, un robot nunca podría ser completamente autónomo. De tal manera que se puedan usar datos de sensores para decidir qué hacer a continuación y luego ejecutar algunas acciones usando efectores. Esto puede parecer una descripción simple, pero al final, resulta que los controladores son bastante difíciles de programar, especialmente cuando se utilizan por primera vez, por lo que, se iniciará con los conceptos generales, la selección del controlador y el uso del ambiente de desarrollo para finalmente programarlo y realizar las tareas mencionadas.

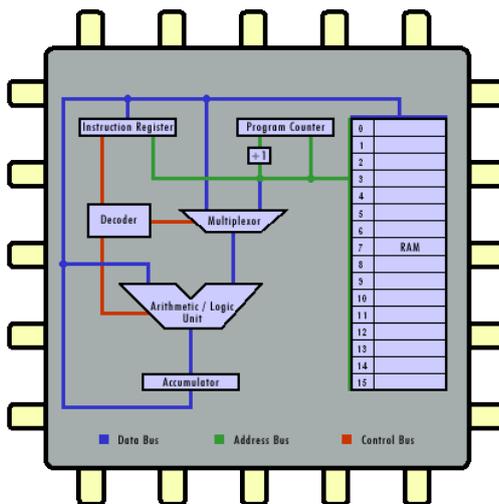


Fig. 3.4 Microprocesador.

Comenzaremos por definir a un microprocesador como un solo circuito integrado de muy alta escala de integración que en general se conoce como hardware. Y se suele llamar por analogía el «cerebro» de un ordenador. Tiene como características principales la programabilidad y la universalidad, por ser un dispositivo electrónico que se puede usar en muchas aplicaciones. De tal manera que al ejecutar una serie de instrucciones secuenciales que en general se conoce como software, puede ser reprogramado para realizar tareas desde encendido y apagado de relevadores en un tiempo determinado hasta cálculos aritméticos de alta precisión.

El microprocesador se conecta a un circuito de reloj, normalmente basado en un cristal de cuarzo capaz de generar pulsos a un ritmo constante, de modo que genera varios ciclos (o pulsos) en un segundo. Este reloj, en la actualidad, genera miles de megahercios.

Desde el punto de vista lógico, singular y funcional, el microprocesador está compuesto básicamente por: varios registros, una unidad de control, una unidad aritmético-lógica, y dependiendo del fabricante, puede contener una unidad de coma flotante.



El microprocesador ejecuta instrucciones almacenadas como números binarios organizados secuencialmente en la memoria principal. Cuando al microprocesador se le conectan mas dispositivos de entrada y salida se dice que es una microcomputadora. Cuando el término se introdujo por primera vez a fines de la década de 1970, se refería a una computadora con un único microprocesador como su CPU, es decir, la computadora personal (PC).

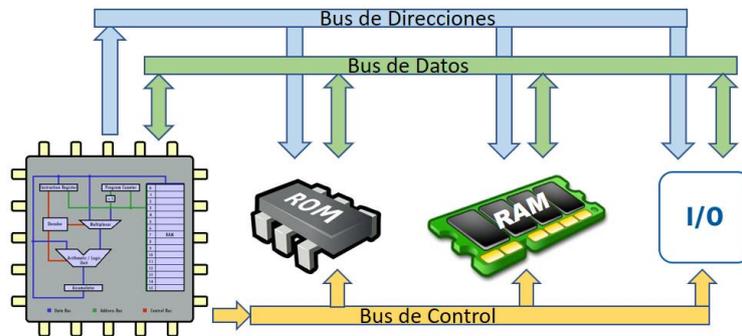


Fig. 3.3. Microcomputadora.

Se puede afirmar que un Microcontrolador es una microcomputadora en un único chip que contiene el procesador (la CPU), memoria no volátil para el programa (ROM o flash), memoria volátil para entrada y salida (RAM), un reloj y una unidad de control de E / S. Disponible en numerosos tamaños y arquitecturas, miles de millones de unidades de microcontroladores (MCU) se incrustan cada año en productos desde juguetes hasta electrodomésticos y automóviles.

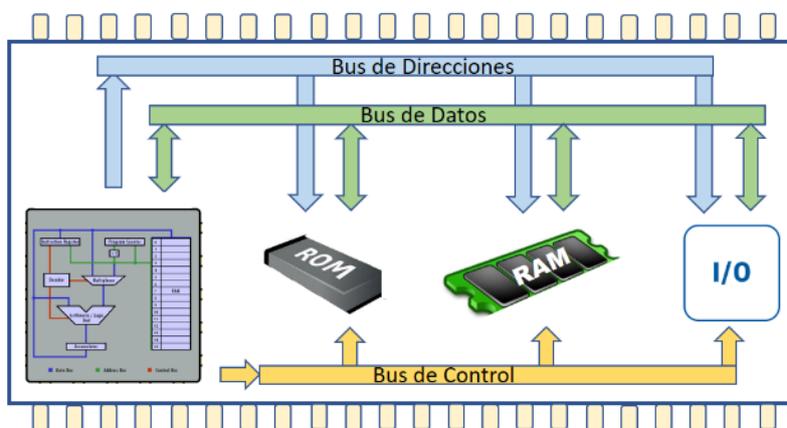


Fig. 3.4. Microcontrolador.

Actividad 3.1.

Se sugiere revisar estos conceptos en referencias especializadas, así como los fundamentos e historia de los microprocesadores y microcontroladores.

Ver [Material de clase. Tema I](#)



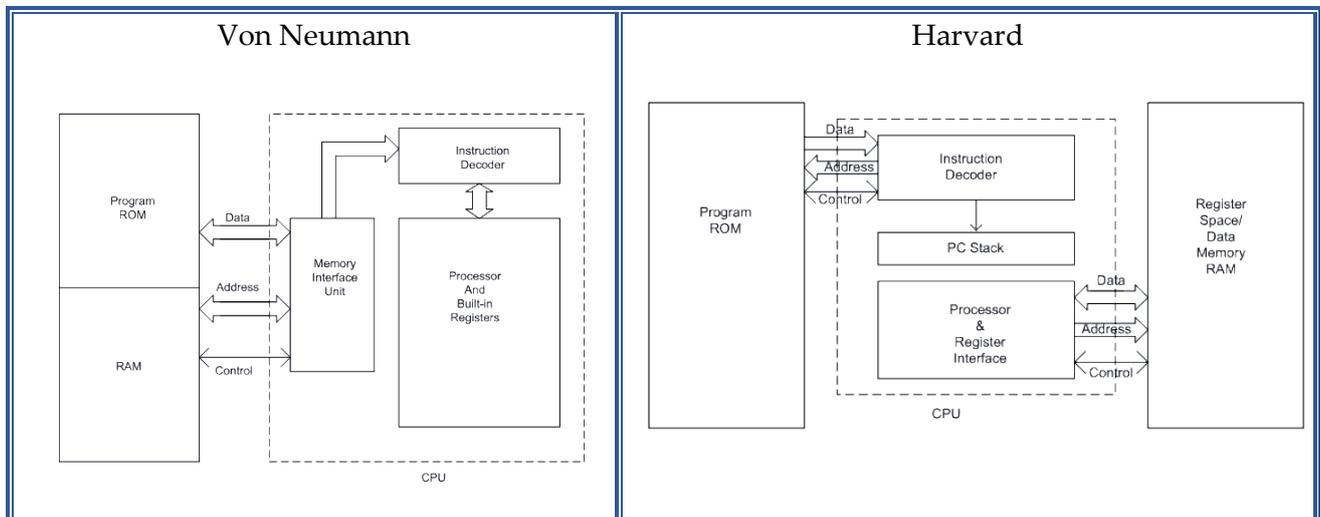
Contraste entre la arquitectura Von Neumann y Harvard.

A fines de la década de 1940, el gobierno de los EE. UU solicitó a las universidades de Harvard y Princeton que crearan una arquitectura informática para utilizar en el cálculo de distancias de la artillería naval para aplicaciones de defensa. Princeton sugirió la arquitectura de la computadora con una única interfaz de memoria. También se conoce como la arquitectura Von Neumann después del nombre del científico jefe del proyecto.

Harvard sugirió una computadora con dos interfaces de memoria diferentes, una para los datos / variables y la otra para el programa / instrucciones. Aunque la arquitectura de Princeton fue aceptada por su simplicidad y facilidad de implementación, la arquitectura de Harvard se hizo popular más tarde, debido al paralelismo de la ejecución de instrucciones.

Aunque la tecnología de proceso y de encapsulado son vitales en la elaboración de procesadores más rápidos, es la arquitectura del procesador lo que hace la diferencia entre el rendimiento de una CPU y otra. Así que, dependiendo de cómo el procesador almacena los operandos en la memoria, existen tres tipos de juegos de instrucciones: basadas en pilas, en el acumulador y basadas en registros. Las arquitecturas RISC (Conjunto Reducido de Instrucciones) y CISC (Conjunto Complejo de Instrucciones) son ejemplos de CPU con un conjunto de instrucciones para arquitecturas basadas en registros. Y debido al paralelismo, la arquitectura de Harvard ejecuta más instrucciones en un tiempo dado en comparación con la arquitectura de Von Neumann.

Tabla 3.1 Arquitectura Harvard vs. Von Neumann.



Los Microcontroladores en los sistemas de control.

Los microprocesadores se utilizan principalmente con fines computacionales, mientras que los microcontroladores encuentran una amplia aplicación en dispositivos que necesitan un procesamiento en tiempo real.

Si tratamos de ver las partes de un robot de manera abstracta, esencialmente hay tres procesos importantes: detección (hecho por sensores), actuación (hecho por efectores), y planeación (hecho por controladores). Dependiendo de cómo juntemos estos tres procesos (ya que son los bloques de construcción también se les llama primitivos), podemos obtener diferentes arquitecturas con diferentes propiedades, también llamadas paradigmas.

Control reactivo

El control reactivo es probablemente la arquitectura (o paradigma) más simple, en este paradigma, como podemos ver en la siguiente figura, no hay un proceso de planificación involucrado. Hay una conexión directa entre la detección y la actuación, lo que significa que tan pronto como entran los datos del sensor, los efectores actúan sobre el medio ambiente de alguna manera predefinida:



Fig. 3.5. Control reactivo.

Así como los reflejos en nuestro cuerpo no envían la información al cerebro sobre algo que está pasando (que sería bastante lento), sino más bien simplemente a la médula espinal que es más cercana para que la respuesta sea rápida. Entonces el control reactivo del robot no tendrá ningún cálculo complejo, sino acciones rápidas y precalculadas que serán almacenadas.

Control jerárquico (deliberativo)

Supongamos que estás programando un robot de juego de ajedrez, sería genial si hiciera alguna planificación sobre el futuro para que pueda anticipar los turnos del oponente y luego ajustar su estrategia, basado en el turno actual del oponente. Una configuración como esta será perfecta para el paradigma de control jerárquico (o deliberativo). Como puede ver en la siguiente figura, el ciclo de planificación, actuación y detección es cerrado. Por lo tanto, el sistema puede moverse activamente hacia su objetivo, sea lo que sea:



Fig. 3.6. Control deliberativo.

Control híbrido

Los paradigmas de control anteriores son rápidos, pero no muy flexibles, o inteligente. Lo que realmente necesitaremos en muchos casos es algo intermedio. Y esto es precisamente lo que un paradigma de control híbrido intenta ofrecer. ¿Cómo podemos usar esto en un robot de la vida real? Supongamos que queremos construir un robot que tenga un cierto comportamiento (por ejemplo, las



hormigas que recolecta y almacena comida). Tal robot definitivamente necesitaría tener su propia representación de su ubicación y donde debe llevar lo que recolecta

Una vez que se le haya dado la tarea de entregar el alimento, tendrá que planificar su camino y luego moverse junto a ese camino. Como podemos esperar, quizás haya otros elementos que también realizan la misma función. No podemos dejar que nuestro robot choque con cualquier objeto, y mucho menos dirigirse a otro destino Para esto, necesitamos un controlador reactivo bien ajustado, en la siguiente figura se muestra el esquema del paradigma de control híbrido y podemos ver que el robot al principio planifica su tarea, pero la divide en una serie de acciones que pueden ser ejecutadas por el paradigma reactivo. Una cosa interesante de notar aquí es el hecho de que los datos sensoriales están disponibles para ayudar a la planificación y las partes que actúan del sistema:

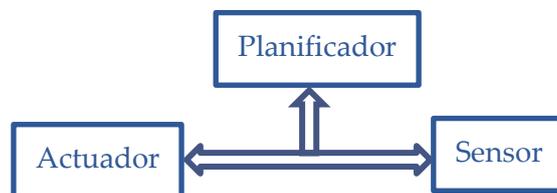


Fig. 3.7. Control híbrido.

La aplicación de microcontroladores es numerosa, desde aplicaciones domésticas, como lavadoras, televisores, aire acondicionado, hasta su utilización en automóviles, también en la industria de control de procesos, teléfonos celulares, unidades eléctricas, robótica y en aplicaciones espaciales.

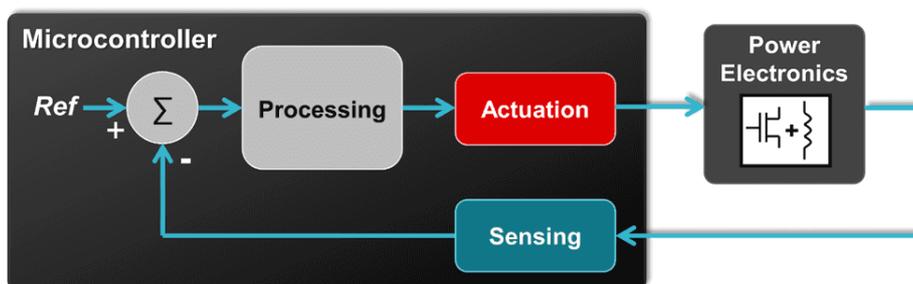


Fig. 3.8. El microcontrolador en los sistemas de control.



Los fabricantes de semiconductores ofrecen una gran variedad de microcontroladores de 8, 16 o 32 bits, según la aplicación que se desee controlar.



Fig. 3.9. Fabricantes y familias de circuitos para las principales aplicaciones en la industria.

Actividad 3.2.

Se sugiere visitar los sitios de los fabricantes más populares y revisar modelos y costos relacionados con los microcontroladores de 32 bits con arquitectura ARM (**Advanced RISC Machines**) Series Cortex-M.

Ver Wikipedia [ARM Cortex-M](#)



Tarjeta de desarrollo Tiva-C LaunchPad.

Las tarjetas de desarrollo son típicamente placas de E / S, que pueden enviar señales de control en forma de pulsos digitales a la placa controladora de motores y puede recibir entradas de varios tipos de sensores También se pueden interconectar los codificadores de motores a la placa de control para la retroalimentación del movimiento.

Los LaunchPads Tiva-C (también conocidos como TM4C) [2] son microcontroladores autónomos de bajo costo, que cuentan con una CPU ARM Cortex-M4F de 32 bits que opera de 80 a 120 MHz, fabricados por Texas Instruments. Las E/S digitales y analógicas pueden configurarse fácilmente, permitiendo una gran variedad de aplicaciones, también los múltiples puertos seriales ofrecen la capacidad de interactuar con más elementos como tarjetas de prueba u otros módulos de comunicaciones.

El reloj de 80 o 120 MHz, hace que sea de 5 a 7 veces más rápida que el microcontrolador ATmega328P de 16 MHz del Arduino Uno. Al igual que con cualquier Cortex M4, la CPU tiene algunas instrucciones DSP (procesador de señal digital). Y la CPU contiene una unidad de coma flotante, así como un puerto USB adicional lo que permite la conexión de múltiples dispositivos, también tienen LED (s) RGB, que le permite generar varios colores combinando los tres colores básicos (rojo, azul y verde).

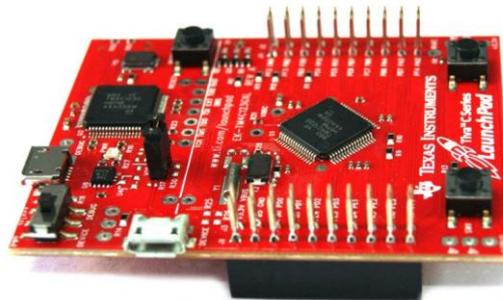


Fig. 3.10. Tiva-C Tm4c123gh6pm.

Actividad 3.3.

Se sugiere leer el instructivo de la caja y correr el demo para revisar su correcto funcionamiento.

Ver [ReadMe First \(Rev. A\)](#)