

Contenidos



Para qué necesitamos los robots?	p. 114
Los robots fischertechnik	p. 116
Actuadores	p. 116
Sensores	p. 116
ROBO Interface	p. 117
Software ROBO Pro	p. 117
Alimentación de corriente	p. 117
Pasos de la experimentación	p. 118
Primeros pasos	p. 118
El primer robot sencillo	p. 120
Robots inteligentes sobre ruedas	p. 122
Modelo básico	p. 122
Robot con detección lumínica	p. 124
Robot con seguimiento de rastros	p. 126
Robot con detección de obstáculos	p. 127
Robot con detección lumínica y de obstáculos	p. 130
Robot con detección de bordes	p. 132
El robot andador	p. 135
Posibilidades de extensión	p. 137
Emisor de infrarrojos manual	p. 137
ROBO RF Data Link	p. 137
ROBO I/O-Extension	p. 138
Búsqueda de errores	p. 139

Para qué necesitamos los robots?

■ El concepto de robot se utilizó por primera vez en 1923 en la novela de Karel Capek *El Golem*. Se trataba de una figura de fabricación artificial concebida para reemplazar a los trabajadores humanos gracias a sus funcionalidades.

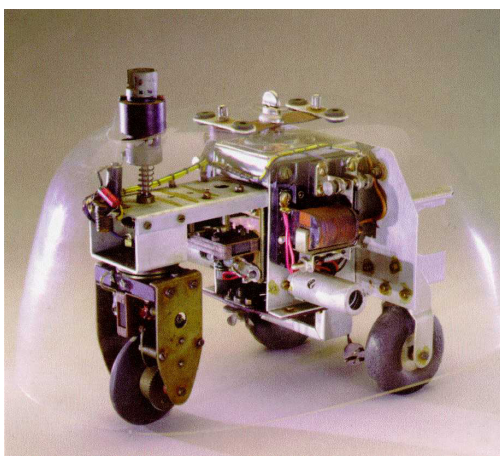
En los años treinta y cuarenta del pasado siglo surgió del robot una especie de autómatas. Los diversos intentos que se hicieron para dotar a los robots de características humanas (por ejemplo, ponerles cabeza con luces intermitentes a modo de ojos) se nos antojan hoy en día una simple curiosidad sin mayor utilidad ni importancia. Poco hay que decir acerca de la movilidad y sobre todo de la inteligencia de estas máquinas. Puesto que el principio del control ha tenido una gran influencia sobre la robótica, el advenimiento de los circuitos electrónicos trajo consigo una mayor dosis de realismo en la construcción de los robots. La cuestión de la "inteligencia" de los robots continúa siendo hoy en día objeto de investigación y análisis en innumerables empresas, institutos y universidades.

■ Las primeras soluciones parecieron estar a punto de llegar de manos de la llamada cibernética. El término "cibernética" procede del griego "kybernetes", una palabra que designaba al piloto de las naves de remos de la Antigua Grecia. El piloto debía determinar la posición de la nave y calcular el rumbo necesario para alcanzar su destino.

De ello se desprende que el objetivo de la cibernética es precisamente dotar al robot de "inteligencia". Pero qué entendemos, para empezar, por un comportamiento inteligente?

Vamos a tratar de aclarar este punto con ayuda de un experimento mental. Todos hemos observado alguna vez el comportamiento de una polilla atraída por el círculo de luz de una bombilla. La polilla reconoce la fuente de luz y vuela decidida hacia ella, para poco antes del choque apartarse de la lámpara. Es evidente por este comportamiento que la polilla detecta la fuente lumínica, localiza un camino que la lleve a ella y emprende el vuelo hacia allí. Estas capacidades se basan en el modelo de conducta inteligente que los insectos poseen como instinto.

Lo que nosotros intentamos es trasladar estas capacidades a un sistema técnico. Debemos no sólo detectar la fuente de luz (sensores ópticos) y llevar a cabo un movimiento (controlar un motor), sino también establecer una conexión práctica y útil entre la detección y el movimiento (programa).



■ El experimento mental que acabamos de realizar fue llevado a la práctica en los años cincuenta por Walter Grey.

Con ayuda de sensores, motores y circuitos electrónicos sencillos consiguió crear una serie de animales cibernéticos que exhibían comportamientos muy específicos, como por ejemplo el de la polilla. La fotografía muestra una reproducción de la tortuga cibernética que se muestra en el Museo Smithsonian de Washington.

También nosotros nos basaremos en la misma reflexión e instilaremos en nuestros robots los modelos de conducta correspondientes, transformándolos en programas para que los robots puedan comprenderlos.

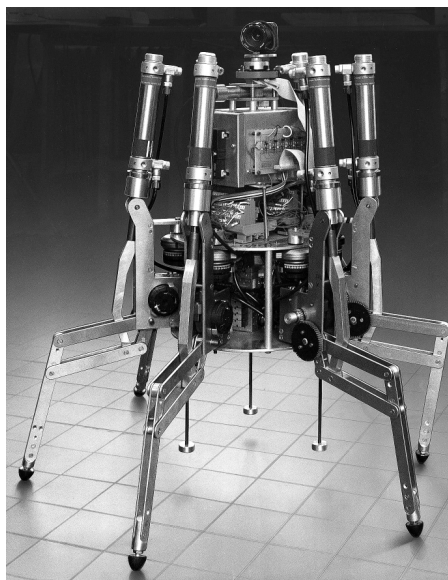
■ Pero para qué necesitamos robots móviles? Vamos a tratar de aplicar a un ingenio técnico el comportamiento de nuestra polilla imaginaria. Para ello comencemos por un ejemplo sencillo: la detección lumínica. Modificamos la fuente de luz trazando una línea luminosa (la pauta o línea directriz) en el suelo y variamos la dirección de los sensores de manera que apunten hacia abajo y no hacia adelante. Con ayuda de la línea directriz, un robot móvil puede orientarse en un almacén, por ejemplo. Otra información adicional, como puede ser en forma de un código de barras que aparezca en determinados puntos de la línea, hace que el robot lleve a cabo otras acciones determinadas al alcanzar esos puntos concretos, como por ejemplo levantar o depositar un palé.



Hoy en día ya existen sistemas robóticos de estas características. Los grandes hospitales se hallan recorridos en parte por grandes vías de transporte de material de desecho, como es la ropa de cama que va a la lavandería. El transporte de estos materiales por parte de personal sanitario resulta caro e implica grandes esfuerzos físicos; además, este tipo de tareas reducen el tiempo que el personal sanitario puede dedicar a los pacientes.

■ Desde hace varios años los científicos trabajan en una forma de movimiento más compleja y muy extendida en la naturaleza: la marcha o la carrera. Se están desarrollando robots dotados de la capacidad de dar pasos sobre patas o piernas. Un ejemplo de robot andador de seis patas es Aquiles, un ingenio electroneumático desarrollado en la Real Academia Militar de Bruselas. Equipado con una cámara en la parte superior y en cada una de las seis patas, este robot está diseñado para reaccionar mecánicamente ante discontinuidades de la superficie sobre la que se desplaza (obstáculos y hoyos).

Estas máquinas andadoras serían de especial utilidad en aquellos lugares que resultan inaccesibles para los vehículos de ruedas o cadenas, como por ejemplo cuando hay que transitar por terrenos extremadamente blandos o accidentados, superar obstáculos, subir escaleras, salvar zanjas o llegar a los puntos de peor acceso y mayor peligrosidad de centrales nucleares, galerías de minas u operaciones de rescate.

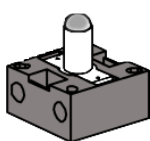


No cabe duda de los robots móviles pueden llegar a desempeñar un papel de importancia en la sociedad moderna.



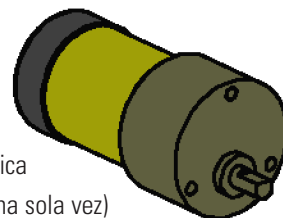
Los robots fischertechnik

Actuadores



Motor eléctrico:

Dos de estos potentes motores de corriente continua (9VDC/2,4W) con transmisión incorporada y una relación de desmultiplicación de 50:1 impulsan los modelos de robot móvil (la relación de desmultiplicación significa que, por cada 50 revoluciones del motor, los árboles que de él salen giran una sola vez)

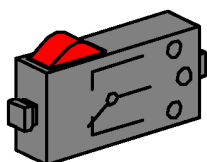


Lámpara lenticular:

Esta lámpara incandescente (9VDC/150mA) sirve para emitir señales luminosas sencillas.

El globo de la bombilla tiene una lente integrada que concentra la luz emitida. Si se dirige el rayo de luz hacia un sensor de luminosidad (un fototransistor, como veremos más adelante) puede construirse una barrera de luz que distinga la luz de la oscuridad. La bombilla también se puede utilizar para indicar determinados estados o como lámpara intermitente para emitir advertencias de peligro. En el kit modular la lámpara se utiliza en conjunción con 2 fototransistores como sensor especial de detección de líneas.

Sensores



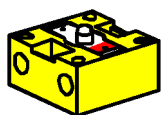
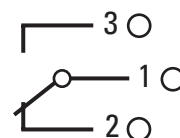
■ Dentro de los sensores existen dos tipos: los analógicos y los digitales.

El pulsador es un ejemplo de sensor digital. Los sistemas digitales pueden adoptar solamente uno de dos estados posibles, que se nombran con los valores 0 y 1. En el caso del pulsador, un valor de 0 indica que el flujo de corriente está interrumpido, mientras que un valor de 1 indica que pasa corriente.

El **pulsador** fischertechnik se ha concebido como conmutador, por lo que está equipado con 3 contactos. Al oprimir el botón rojo, el pulsador conecta físicamente los contactos 1 y 3. Al mismo tiempo se interrumpe el contacto entre el 1 y el 2, que en estado de reposo estaban tocándose. De este modo puede averiguarse ante cuál de las dos posibilidades de salida estamos:

Cerrado en estado de reposo (contactos 1 y 2 juntos)

Abierto en estado de reposo (contactos 1 y 3 juntos).



El **fototransistor** puede utilizarse indistintamente como sensor analógico o digital. En el primer caso su función es detectar transiciones claras de luz a oscuridad, como puede ser, por ejemplo, una línea que se haya trazado. También puede utilizarse, sin embargo, para discriminar grados de luminosidad, puesto que el fototransistor también puede funcionar en modo analógico. Los valores analógicos pueden modificarse a voluntad entre los dos valores extremos. Para que estas magnitudes puedan procesarse informáticamente, tienen que transformarse en sus correspondientes valores numéricos.

El fototransistor constituye, por otra parte, un componente semiconductor cuyas propiedades eléctricas dependen de la intensidad de la luz. De todos son conocidas las células solares, que transforman la luz

del sol en electricidad. Pues bien: el fototransistor puede entenderse como la combinación de una célula solar en miniatura y un transistor. Los impulsos lumínicos que impactan en el fototransistor (fotones) generan una corriente eléctrica muy pequeña que el transistor se encarga de reforzar.

Atención:

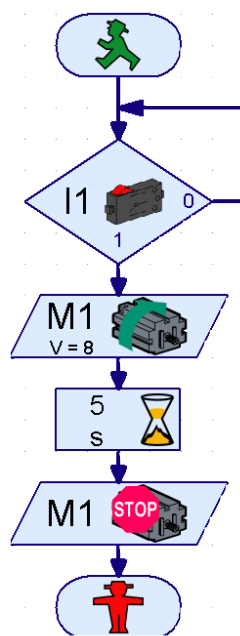
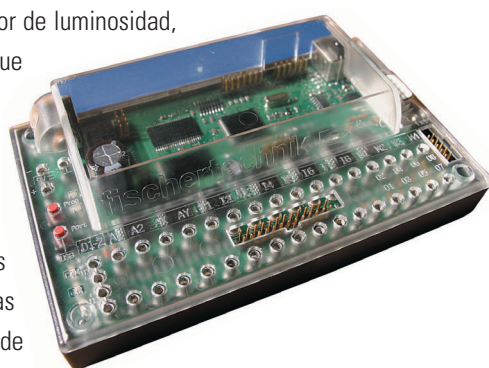
Al conectar el fototransistor es importante localizar perfectamente el polo correcto: rojo = positivo.

Tensión soportada: 30V máx.

■ La ROBO Interface permite la conexión y la utilización de distintos sensores y actuadores. Además de con sus 8 entradas digitales, la ROBO Interface cuenta con varias entradas analógicas. De este modo, una resistencia medida en las entradas AX y AY que oscile entre 0 y $5,5k\Omega$ se transforma en un valor numérico de entre 0 y 1024. Así se obtienen los valores de medición de un sensor de luminosidad, como puede ser un fototransistor, dispuestos para ser procesados de la manera que queramos. En las entradas analógicas A1 y A2 pueden medirse tensiones de entre 0 y 10VDC.

La función más importante de la interfase es el procesamiento lógico de las magnitudes de entrada. Para poder desempeñarla, la interfase necesita una aplicación informática. El programa decide de qué manera se derivarán de los datos de entrada las señales de sensores, los correspondientes datos de salida, las señales de motor etc. Con la ROBO Interface disponemos de suficiente capacidad de computación para diseñar incluso programas complicados.

ROBO Interface



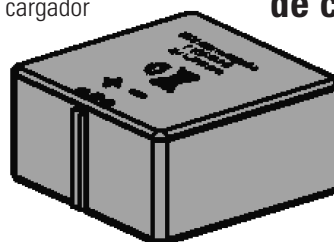
■ Para que podamos elaborar los programas que necesita la interfase de la manera más efectiva posible, tenemos a nuestra disposición una plataforma gráfica de programación. Por plataforma de programación se entiende un software que nos permite crear nuestro programa con gran comodidad. Se trata de una plataforma gráfica en la que podemos programar con ayuda de símbolos visuales. En realidad, el computador interno de la ROBO Interface solamente es capaz de ejecutar las órdenes que forman parte de su juego de instrucciones. Este juego de instrucciones es, en esencia, un conjunto de sencillas estructuras de control cuyas aplicaciones resultan extraordinariamente complicadas para los que se inician en el tema. Por esta razón, la aplicación informática ROBO Pro ofrece un conjunto de elementos gráficos que son traducidos inmediatamente a un lenguaje que la interfase pueda procesar.

Software ROBO Pro

■ Lo único que nos falta para completar el ROBO Mobile Set es el Accu Set, Art. N° 34969. Contiene un acumulador que funciona como alimentación portátil para los modelos de robot que utilizaremos y un cargador especial para utilizar con el acumulador.

Lo mejor es que conectemos el acumulador al cargador ya mismo, para que esté cargado cuando más tarde queramos empezar el experimento.

Alimentación de corriente



Pasos de la experimentación

■ Para entrar en el fascinante mundo de los robots móviles vamos a proceder paso a paso. Empezaremos con un sencillo montaje de prueba para probar las funciones básicas de la interfase y el conjunto de sensores. A continuación construiremos unos modelos sencillos que desempeñan tareas concretas y ordenadas en función de su dificultad, y pasaremos después a sistemas de creciente complicación.

Si en cualquier momento comprobamos que crear nuestros propios programas resulta demasiado complicado y lleva demasiado tiempo, siempre podemos limitarnos a cargar los programas de ejemplo que se adjuntan y manejar los robots con ellos.

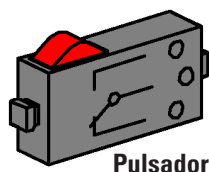
Para evitar desesperarnos con los fallos que se puedan presentar en el proceso, al final de este manual hay un capítulo dedicado a la búsqueda de errores.

Es muy importante proceder con gran cuidado y meticulosidad durante la construcción y puesta en servicio de los robots. Al conectar los componentes eléctricos debemos seguir las instrucciones al pie de la letra y comprobar que todo está correcto no sólo una, sino preferiblemente dos o incluso tres veces. En las construcciones mecánicas, y también con nuestras propias creaciones, siempre pondremos especial cuidado en que los elementos trabajen con suavidad y nos aseguraremos de que se acoplen y funcionen con la holgura justa. Queda en manos de nuestra creatividad el escribir nuestros propios programas y con ello definir nuevos comportamientos para los robots. El único límite serán las capacidades de memoria y procesamiento del hardware. Los siguientes ejemplos ofrecen algunas sugerencias.

Primeros pasos

■ Tras las anteriores consideraciones teóricas, queremos empezar ya a realizar nuestros propios experimentos. Seguramente habrá quien desee empezar a hacer cosas enseguida, quizás incluso con el robot andador grande. Es una opción perfectamente posible: de hecho, si se observan con esmero las instrucciones de montaje, el modelo quedará perfectamente construido incluso tratándose del primer intento.

Pero qué podemos hacer si no funciona? En ese caso habrá que emprender una búsqueda sistemática de la causa del fallo. Eso sí: antes de ponernos a trabajar, debemos probar la interacción entre el ordenador y la interfase.

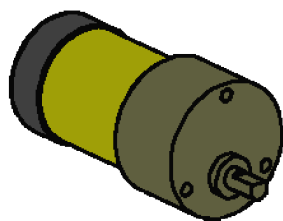


Pulsador

En los capítulos 1 y 2 del manual del software ROBO Pro se explica cómo instalar el software de control en el PC y cómo conectar la interfase. Con ayuda de la prueba de interfase comprobamos los distintos sensores y actuadores.

Pulsador

Podemos, por ejemplo, conectar un pulsador a la entrada digital I1 y observar cómo varía el estado de la entrada al accionar el pulsador.



Motor eléctrico

Motor eléctrico

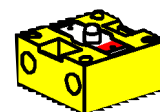
Comprobamos las salidas conectando un motor a una de las de motor, por ejemplo M1. Con el botón izquierdo del ratón podemos establecer el sentido de giro del motor; el regulador deslizante, por su parte, nos permite determinar la velocidad de giro.

Fototransistor

Si también queremos probar la entrada analógica AX, podemos utilizar un fototransistor como sensor analógico.

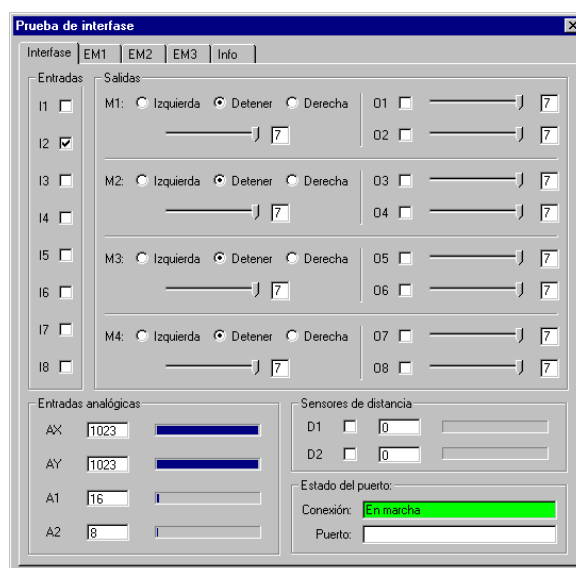
Mientras que en el caso del motor y del pulsador la polaridad de los contactos no desempeña ningún papel (en el peor de los casos el motor giraría en sentido inverso), para que el fototransistor funcione correctamente es imprescindible que esté conectado como es debido.

El contacto del transistor de la marca roja va con un conector también rojo; el otro, con un conector verde. Los dos conectores verdes van a la hembra de la entrada AX, situada cerca del borde de la interfase; el segundo conector rojo va en la hembra de AX que queda más al centro. (Atención: si se conecta el fototransistor a una entrada digital I1-I8, el conector rojo va en la hembra que queda más cerca del borde de la carcasa).



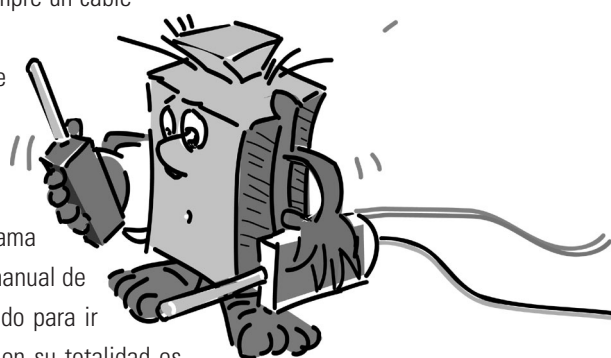
Fototransistor

Ahora podemos modificar la intensidad lumínica detectada por el fototransistor con ayuda de una linterna y con ello hacer que se mueva la rayita azul de AX. Si el indicador no se moviese hasta el máximo, deberemos revisar otra vez los contactos del fototransistor. Si por el contrario el indicador sigue estando a cero a pesar de la linterna, el problema puede ser que la luminosidad de la habitación (la claridad ambiental) sea demasiado alta. El indicador empezará a oscilar si tapamos el fototransistor.

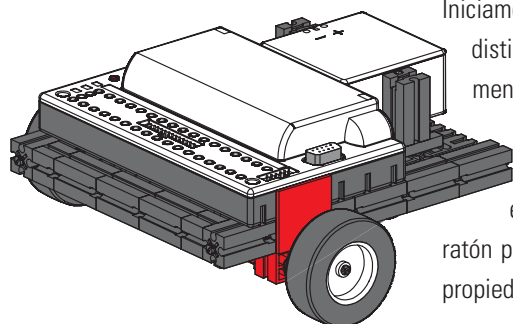


Recordemos brevemente el código cromático de los conectores: ponemos especialísimo cuidado durante el montaje en conectar siempre un conector rojo con el cable rojo y un conector verde con el cable verde. Si al montar un circuito hemos proporcionado la polaridad correcta, tomaremos siempre un cable rojo para el polo positivo y un cable verde para el negativo. Tanta insistencia puede parecer un poco exagerada, pero a la hora de realizar una búsqueda sistemática de errores contar con una clasificación cromática meridianamente clara constituye una ventaja incuestionable.

Queremos dar nuestros primeros pasos en el mundo de la robótica con un programa sencillo. El programa de control de puerta de garaje explicado en el capítulo 3 del manual de Robo PRO no tiene nada que ver con los robots móviles, pero resulta muy apropiado para ir familiarizándonos con el software ROBO Pro. Para poder comprender el programa en su totalidad es suficiente conectar a la interfase el motor y tres pulsadores del kit de construcción ROBO Mobile Set. Todo lo demás viene detalladamente explicado en el manual del software.



El primer robot sencillo



■ Una vez llevada a cabo la prueba de interfase y probado el programa de control de puerta de garaje, procedemos por fin a poner en marcha nuestro primer robot. Montamos el modelo "Robot sencillo" con los dos motores conforme a las instrucciones de construcción. El proceso resulta extremadamente simple y rápido, ya que con toda la intención este modelo se ha concebido únicamente con los elementos imprescindibles para el movimiento de un robot. Conectamos los motores a las salidas M1 y M2.

Iniciamos el software ROBO Pro y creamos un programa nuevo en (ARCHIVO - NUEVO). ROBO Pro ofrece distintos niveles de dificultad en los que podemos operar. El nivel se configura en el elemento de menú NIVEL. Por el momento será suficiente con escoger un nivel 1.

Aparecerá una superficie de trabajo vacía y, en el margen derecho, la ventana de elementos en la que iremos seleccionando los distintos elementos de programa con el botón derecho del ratón para colocarlos sobre la superficie de trabajo. Con el botón derecho del ratón modificamos las propiedades.

Tarea 1 (nivel 1):

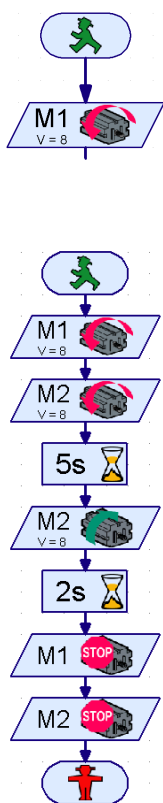
Nuestro "Robot sencillo" debe avanzar en línea recta durante 5 segundos, a continuación girar sobre sí mismo durante 2 segundos y finalmente detenerse.



Consejos:

El primer robot lo programaremos juntos paso a paso:

- Empezamos con el hombrecillo verde que recuerda al de un semáforo. Simboliza el inicio del programa.
- A continuación tomamos el símbolo de motor de la ventana de elementos y lo situamos debajo del elemento de inicio; se dibujará automáticamente una línea de unión entre ambos. En la ventana de propiedades marcamos la salida de motor como "M1" y el sentido de giro como "izquierda", y confirmamos la selección con OK.
- Debajo de este símbolo colocamos de la misma manera otro símbolo de motor, de modo que quede conectado el motor 2.
- Para indicar un margen de espera determinado, utilizamos el elemento tiempo de espera, que colocamos debajo del segundo símbolo de motor y que configuramos con un intervalo de 5 segundos.
- A continuación hacemos que el motor M2 gire en el sentido inverso (hacia la derecha), que espere después 2 segundos y para terminar apagamos ambos motores. Nuestro programa concluye con el símbolo de fin, el hombrecillo rojo. La ilustración muestra el programa completo.



Si no estamos seguros de que todo esté correcto, podemos comparar nuestro programa con el de ejemplo que ya viene incluido. Para ello tenemos que guardar antes de nada el programa que hemos creado y el cargar el archivo Robot sencillo 1.rpp que se halla en la carpeta de ejemplos de ROBO Pro (la ubicación por defecto es C:\Archivos de programa\ROBO Pro\Programmas de ejemplo\ROBO Mobile Set). Si todo está en orden, el programa se traslada a la interfase mediante la funcionalidad de descarga. Tras oprimir el botón Descarga se abre una ventana de diálogo. En ella indicamos que el programa debe almacenarse en la memoria FLASH 1 y que debe iniciarse al terminar la descarga.

Inmediatamente después de la descarga, nuestro modelo empezará a avanzar, girará durante un momento y acabará por detenerse. Si queremos iniciar el programa de nuevo, oprimimos brevemente el botón Prog de la interfase. El LED Prog1 volverá a parpadear durante toda la duración del programa y al final se iluminará permanentemente. El programa continuará almacenado en la memoria FLASH de la interfase incluso después de interrumpida la alimentación de corriente. Vamos a comprobarlo desconectando el acumulador. Tras sacar el conector, volvemos a introducirlo y seleccionamos el programa almacenado apretando el botón Prog hasta que el LED Prog1 se ilumine. Oprimimos de nuevo el botón e iniciamos así el programa.

Hasta ahora el robot no ha hecho demasiado, verdad? Es cierto: por eso queremos ampliar un poco la tarea.



Tarea 2 (nivel 1):

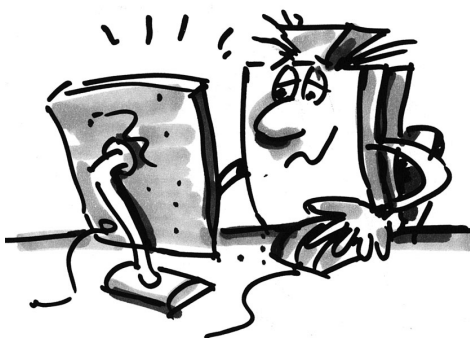
Para que nuestro robot no se detenga al pasar los 7 segundos de la primera tarea, queremos ahora enseñarse un baile.

- **Haz que se mueva hacia delante, a la izquierda, a la derecha y hacia atrás durante intervalos diferentes y a distintas velocidades,**
- **repitiéndose la secuencia hasta que se finalice el programa apretando el botón Prog de la interfase.**

Consejos:

- Para que el robot se mueva en cada momento hacia el lado que desees, simplemente invierte la polaridad cuantas veces sea necesario.
- La velocidad de los motores puedes modificarla en la ventana de propiedades de cada símbolo de motor, ajustándola de 1 a 8. Si los motores M1 y M2 giran a diferentes velocidades pero en el mismo sentido, el robot describirá una curva.
- Para que el programa se repita continuamente, dibuja una línea que conecte la salida del último elemento de programa con la línea que conduce al primer elemento.
- Encontrarás un ejemplo ya terminado de este programa en el archivo [Robot sencillo 2.rpp](#).

Felicidades: acabas de construir y programar tu primer robot! El robot no es demasiado inteligente, porque todavía no detecta los obstáculos y se cae de la mesa si no lo vigilas. Pero eso ya lo cambiaremos en el transcurso de posteriores experimentos.



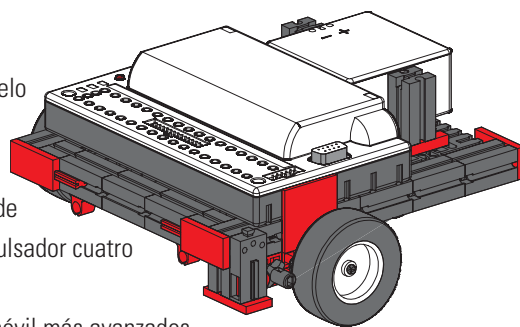
Robots inteligentes sobre ruedas

■ Para que los robots sean capaces de reconocer su entorno, precisamos utilizar sensores. Las siguientes sugerencias de modelos presentan varias variaciones de robots móviles en los que probaremos las funcionalidades de diferentes sensores. Todo depende de cómo combinemos tanto las condiciones internas del robot (por ejemplo, medición del trayecto mediante ruedas de impulsos) como las señales del exterior (por ejemplo, detección de luz o seguimiento de rastros). Para cada modelo se establecerán, por tanto, tareas específicas. Todas ellas deben tomarse como sugerencias que te servirán para ir entrando en materia. Los programas que ejecutan cada tarea específica se hallan en la carpeta de ROBO Pro, en \Programas de ejemplo\ROBO Mobile Set\. Con el tiempo se te ocurrirán nuevas tareas para los modelos; cuando acabes de repasar los siguientes ejemplos, sin duda se te habrán ocurrido ya muchas ideas.

Modelo básico

■ A diferencia de nuestro robot sencillo inicial, el modelo básico posee una construcción más estable y robusta. Contiene además 2 sensores de trayecto, consistentes cada uno en un pulsador y una rueda de impulsos. La rueda de impulsos está conectada con el eje del motor y acciona un pulsador cuatro veces por cada revolución del motor.

Este modelo sirve como base para otros modelos de robot móvil más avanzados.



Construye el modelo básico conforme a las instrucciones, poniendo especial esmero en cada uno de los pasos del montaje. Cuando el mecanismo esté completo, comprueba que los motores funcionan sin trabas conectando cada uno de ellos directamente a los acumuladores (sin interfase) durante unos instantes.

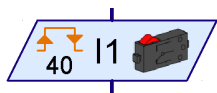


Tarea 1 (nivel 1):

- Programa la interfase para que el modelo avance en línea recta 40 impulsos. Para medir los impulsos, utiliza el pulsador contador de la entrada I1.
- Mide el trayecto que recorre el modelo y calcula cuál es la distancia por impulso.
- Repite el proceso 3 veces y apunta en la tabla cuánto oscilan los valores.

Consejos:

- Empieza por encender ambos motores (sentido de giro a la izquierda).
- Para contar los impulsos de I1, utiliza el elemento de programa **contador de impulsos**.
- Cuenta los dos flancos posibles (0-1 al apretar y 1-0 al soltar el pulsador). Puedes seleccionar uno u otro en la ventana de propiedades, en la sección de tipo de impulso. Al hacerlo aumentas la exactitud de la medición del trayecto.
- Después vuelve a apagar los motores y finaliza el programa.
- Encontrarás el programa completo en el archivo [Modelo básico 1.rpp](#).



**Resultado:**

	Número de impulsos	Distancia recorrida	Distancia/Impulso
Intento 1	40		
Intento 2	40		
Intento 3	40		

Grosso modo puede calcularse que el modelo recorre una distancia de un centímetro por cada impulso.

Al mismo tiempo esto te permite saber qué sentido de giro debes establecer en cada uno de los motores para que el modelo avance hacia uno u otro lado. Apunta estos hallazgos en la tabla siguiente para no tener que deducirlo todo de nuevo cada vez que quieras modificar el sentido de la marcha. Si cableas el modelo exactamente como se ilustra en las instrucciones de montaje, el giro a la derecha de cada uno de los motores hará que la rueda se mueva hacia adelante. Así están programados todos los motores en los programas de ejemplo.

**Completa la tabla:**

Sentido de la marcha del modelo	Sentido de giro M1	Sentido de giro M2
Avance	Izquierda	Izquierda
Retroceso		
Izquierda		
Derecha		
Detención		

Para no tener que colocar dos símbolos de motor en la pantalla cada vez que queremos hacer un cambio en el movimiento, podemos crear un subprograma que se ocupe de esta tarea para cada sentido de la marcha. Esto simplifica enormemente la programación. Las instrucciones para crear un subprograma se hallan en el capítulo 4 del manual del software ROBO Pro. Cuando hayas leído este capítulo completo, estarás en condiciones de atreverte con la siguiente tarea. Pasa la aplicación ROBO Pro al **nivel 2**.

**Tarea 2 (nivel 2):**

- Genera un subprograma para cada sentido de la marcha.
- Programa el modelo para que recorra un cuadrado de un metro de lado.
- Cuál es la exactitud de repetición?

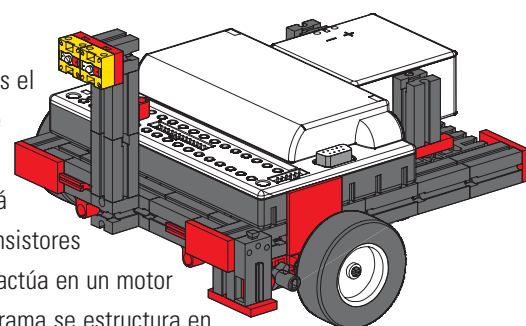




Robot con detección lumínica

Consejos:

- Crea primero un subprograma de **avance**. El resto de los subprogramas puedes fabricarlos a partir de una copia del subprograma de avance, simplemente cambiando el sentido de giro de los motores.
- Tanto para el giro hacia la izquierda como hacia la derecha, utiliza una velocidad reducida. Esto aumenta la precisión.
- Para contar los impulsos utiliza bien el elemento **contador de impulsos** o bien el pulsador de la entrada I1.
- Primero carga el programa en la RAM para ir probándolo hasta que hayas averiguado cuántos impulsos necesitas para que el robot ejecute un giro de 90°. En primer lugar, es más rápido cargar el programa en la memoria RAM que en la FLASH; en segundo lugar, la memoria FLASH tiene una vida limitada de aproximadamente 100.000 descargas.
- El programa ya completo se llama Modelo básico 2.rpp.



■ Una vez investigado en profundidad el modelo básico, es el momento de hacer que el robot aprenda a reaccionar ante señales ambientales. Igual que la polilla del experimento mental propuesto en el capítulo primero, el robot deberá detectar y seguir una fuente de luz. El kit contiene 2 fototransistores que vamos a utilizar como detectores de luz. Cada sensor actúa en un motor para posibilitar el seguimiento de la fuente de luz. El programa se estructura en dos partes. La primera contiene la búsqueda de la fuente lumínica; en la segunda se efectúa el seguimiento o persecución de la fuente. Dentro del programa se emplean varios subprogramas. Inmediatamente tras la puesta en marcha se activa el subprograma de detección lumínica, que no se abandona hasta que se haya localizado la fuente de luz. El programa principal intenta poner al robot rumbo a la fuente de luz. Cada vez que la dirección del robot se desvía perceptiblemente de la línea ideal, uno de los sensores deja de estar bajo la fuente de luz; si eso ocurre, el robot corrige la dirección de modo que ambos sensores puedan detectar de nuevo la fuente lumínica.

Construye ahora el modelo de detección lumínica tal y como se describe en las instrucciones de montaje.

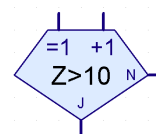
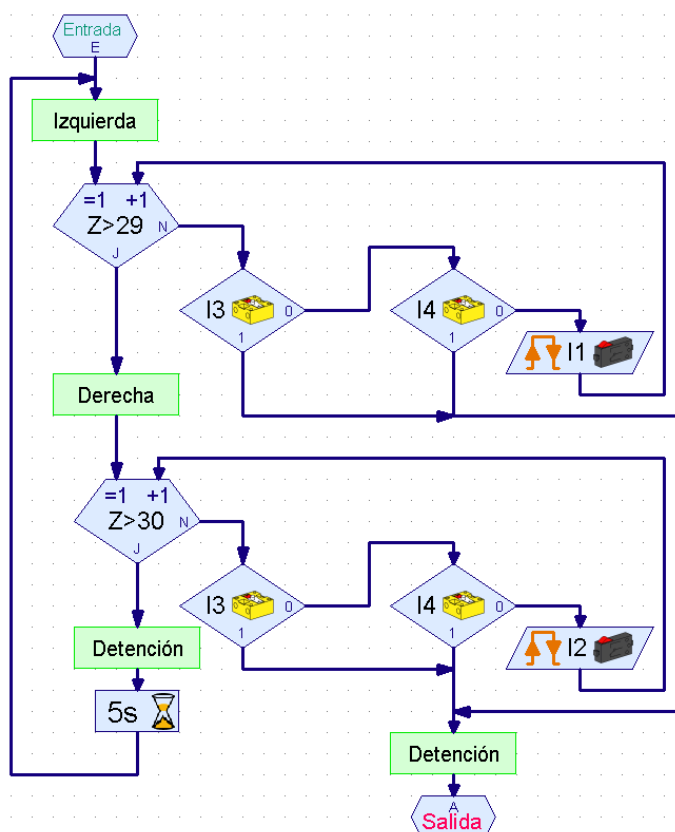


Tarea 1 (nivel 2):

- **Empieza programando la función de detección lumínica. En ella, el robot tiene que definir un giro lento de al menos 360°. Si durante la búsqueda encuentra una luz, el robot se detiene. Si no es así, vuelve a girar otros 360° en sentido inverso. Si sigue sin encontrar ninguna luz, el robot espera 5 segundos y vuelve a emprender la búsqueda.**
- **Si la búsqueda da frutos, el modelo tendrá que poner rumbo hacia la fuente de luz. Si la fuente se mueve a izquierda o derecha, el robot debe seguir el movimiento de la luz. Si pierde el contacto, el programa debe volver al principio y emprender una nueva secuencia de detección lumínica. Comprueba si puedes atraer al robot con una linterna y hacer que atravesase un recorrido de obstáculos.**

Consejos:

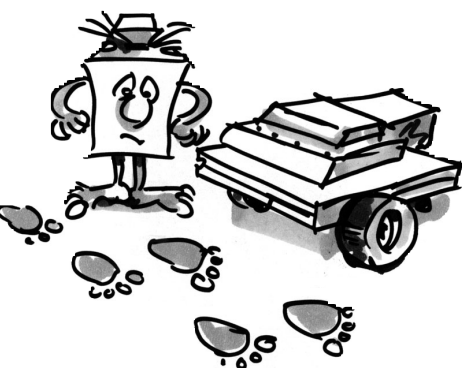
- Para controlar los distintos sentidos de la marcha, utiliza los subprogramas que ya habías programado para el modelo básico. Abriendo el programa Modelo básico 2.rpp encontrarás en la **ventana de grupos de elementos** de ROBO Pro, en **Programas cargados**, el programa Modelo básico 2 y dentro de él los subprogramas que contiene. No tienes más que insertar estos subprogramas en el nuevo programa.
- el subprograma de detección lumínica, utiliza el elemento **bucle contador**. (La descripción de este elemento se encuentra en el manual de ROBO Pro).
- En el bucle entre los contactos N y +1 consulta a los fototransistores y cuenta un impulso en el pulsador I1. El bucle debe repetirse tantas veces como sea necesario hasta que el robot detecte la luz o haya rotado 360°. Para saber cuántas veces se tiene que repetir un bucle para cada rotación completa, simplemente haz la prueba; después ajusta el valor Z del elemento bucle contador en función de lo que hayas averiguado.
- Inmediatamente después, repite el mismo método para programar un segundo bucle, que servirá para realizar la misma búsqueda pero en sentido de giro inverso.
- Si el robot encuentra luz, se detiene y abandona el subprograma.
- Este es el subprograma completo de detección lumínica:

**bucle contador**

- En el programa principal, vuelve a introducir la consulta a los fototransistores y controla los motores en función de cuál de ellos haya reconocido la luz:

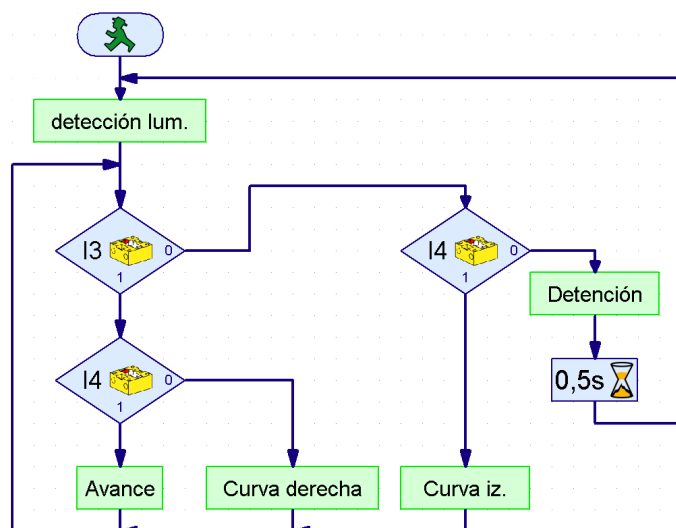
Luz en I3 e I4	Avance
Luz sólo en I3	Curva a la derecha
Luz sólo en I4	Curva a la izquierda
No se detecta luz	Parar, volver al subprograma de detección lumínica

- La curva a la derecha o a la izquierda se consigue aplicando diferentes velocidades a M1 y M2 pero manteniendo el mismo sentido de giro. De este modo se obtiene un estilo de movimiento muy armónico.



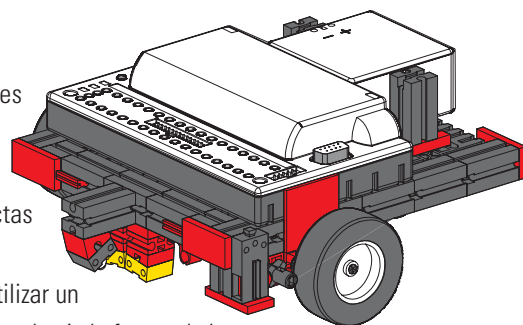
Robot con seguimiento de rastros

- El programa principal tendrá la siguiente apariencia:
- Encontrarás el programa completo en [Detección lumínica.rpp](#).
- Como fuente de luz utiliza una linterna. Procura que el rayo de luz no quede enfocado en un punto demasiado pequeño, para que puedan recibir luminosidad los dos fotosensores. Ten en cuenta que en habitaciones muy iluminadas la luz de la linterna quedará anulada por la de otras fuentes (por ejemplo, la luz del sol que entre por una ventana amplia). En esas circunstancias, el robot no hará caso de la linterna y se encaminará a la luz más intensa.



■ La localización y el seguimiento son propiedades esenciales de las que gozan los seres inteligentes. Con la función de detección lumínica hemos construido y programado un robot que reacciona a señales directas emitidas por su objetivo.

Con la funcionalidad de seguimiento de rastros vamos a utilizar un principio de búsqueda diferente. En lugar de la marcha directa hacia la fuente de luz, vamos a marcar una línea negra que el robot deberá seguir. Esta tarea resulta relativamente fácil de resolver gracias a los fototransistores. Se mide la luz reflejada de la línea trazada y los motores realizan la corrección oportuna. Para que el sistema funcione con precisión, la línea es iluminada por la lámpara. Debemos tener cuidado de que los fotosensores no resulten deslumbrados por luces parásitas de la bombilla a causa de una colocación inconveniente. Un excelente sistema para evitarlo pasa por el enfoque del haz de luz que realiza la lente óptica de la bombilla.



Construye el modelo de seguimiento de rastros conforme a las instrucciones de montaje.



Tarea 1:

- Para empezar, escribe un subprograma que busque el rastro. El robot tendrá que girar una vez sobre sí mismo.
- Si no encuentra ningún rastro, queremos que avance un poco en línea recta y vuelva a realizar la búsqueda. El reconocimiento del rastro se basa en la consulta a los fototransistores.
- Si el robot descubre un rastro, debe seguirlo.
- Si llega al fin del rastro o lo pierde (si se presenta, por ejemplo, un cambio brusco de dirección), él robot deberá emprender de nuevo la búsqueda.

Consejos:

- Una vez encendida la lámpara hay que esperar un momento (aproximadamente un segundo) antes de poder consultar a los fototransistores. De otro modo el fototransistor detectará oscuridad y con ella la presencia de un rastro que no existe, ya que se le realiza la consulta antes de que la bombilla haya tenido tiempo de encenderse por completo.
- Usaremos como rastro un trozo de cinta aislante negra de aproximadamente 20 mm de ancho; también podemos dibujar con un rotulador un rastro negro del mismo ancho sobre una hoja de papel de color blanco. Las curvas no deben ser demasiado cerradas, para evitar que el robot pierda constantemente el rastro.
- Lleva a cabo antes de nada la prueba de interfase para comprobar si los fototransistores reconocen correctamente el rastro. No olvides encender la bombilla.
- Ajusta la lámpara de forma que los dos fototransistores devuelvan el valor 1 al encontrarse sobre una superficie luminosa, incluso estando en marcha los motores M1 y M2. Si el acumulador no se halla totalmente cargado, al encender los motores la luz de la bombilla perderá algo de intensidad. Si no está bien ajustada, puede ocurrir que uno de los fototransistores comunique que detecta oscuridad aunque no haya encontrado ningún rastro.
- El seguimiento de rastros funciona de manera semejante a la detección lumínica. Lo único que debes hacer es adaptar la búsqueda de luz de modo que, si el modelo no detecta nada tras dar una vuelta sobre sí mismo, avance un poco hacia delante antes de volver a rotar.
- Ten en cuenta que en la funcionalidad de seguimiento de rastros el modelo tiene que avanzar en línea recta cuando ambos fototransistores devuelven el valor "oscuridad" (= 0).
- Encontrarás el programa completo en el archivo [Seguimiento de rastros.rpp](#).

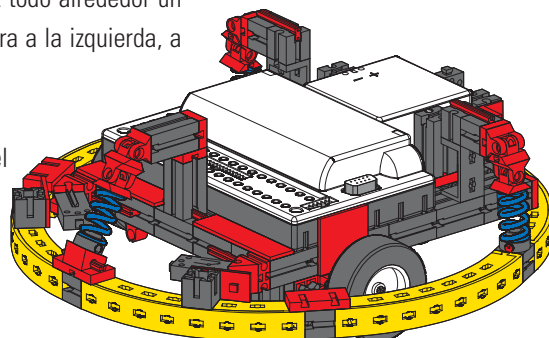
**Tarea 2:**

- **Fabrica un rastro con varias curvas cerradas. Qué radio describe el modelo?**
- **Experimenta la introducción de correcciones con diferentes velocidades de M1 y M2. Qué combinación da mejor resultado?**
- **Crea un rastro en forma de circunferencia. Trata de optimizar las velocidades de modo que el robot complete el círculo lo más rápidamente posible. Estas tareas son perfectas para organizar competiciones con varios robots.**

■ Todos los robots que hemos construido hasta ahora son capaces de recorrer un trayecto determinado o de seguir una fuente de luz o un rastro. Pero qué pasa si se encuentran con un obstáculo en su camino? Una de dos: o bien apartamos el obstáculo a un lado o el robot se queda empujando absurdamente contra él hasta que se le agote el acumulador. Mucho más inteligente sería, sin duda, que el robot detectase el obstáculo y lo evitase en consecuencia. Para ello se le coloca al robot todo alrededor un parachoques flexible con tres pulsadores, que le permiten determinar si se encuentra a la izquierda, a la derecha o detrás del obstáculo.

La reacción que entonces adopte es cuestión ya de la programación que le demos. Empieza montando el modelo "Robot con detección de obstáculos". Para medir el trayecto necesitamos sólo un pulsador (I1), por eso podemos extraer el pulsador I2 del modelo básico y utilizarlo para la función de detección de obstáculos.

Robot con detección de obstáculos





Tarea 1 (nivel 2):

- Primeramente el robot debe avanzar hacia adelante. Si se encuentra con un obstáculo a la izquierda (E4), debe retroceder un poco y evitarlo por la derecha.
- Si se encuentra con un obstáculo a la derecha (E3), debe retroceder un poco y evitarlo por la izquierda.

Consejos:

- La detección de obstáculos durante la marcha atrás no será tratada por el momento.
- En el programa principal se consulta a los pulsadores. En función de cuál de los pulsadores sea activado, el modelo evitará el obstáculo por la izquierda o por la derecha. Cada caso está regulado por su correspondiente subprograma.
- Al contar el número de impulsos registrados en el giro debe distinguirse la izquierda de la derecha (por ejemplo, 3 impulsos a la derecha y 5 a la izquierda), porque si no, puede ocurrir que el modelo vaya a parar a un rincón y no encuentre nunca más la forma de salir de él.
- El programa ya terminado se llama Obstáculos 1.rpp.

Hay dos cosas que el robot detector de obstáculos todavía no puede hacer: detectar obstáculos mientras se mueve hacia atrás y darse cuenta de que tiene un obstáculo justo delante de él. No las hace todavía, pero puede hacerlas. Si durante la marcha atrás se presiona el I5, quiere decir que hay un obstáculo detrás del modelo. Si durante la marcha adelante se activan a la vez el I3 y el I4, quiere decir que hay un obstáculo justo delante del modelo. En este caso el robot podría girar 90°. En conjunto, estas son todas las reacciones con las que puede responder el robot:

Obstáculo	Pulsador	Reacción
derecha	sólo I3	Evitar por la izquierda (giro de aprox. 30°)
izquierda	sólo I4	Evitar por la derecha (aprox. 45°)
delante	I3 e I4	Evitar por la izquierda (aprox. 90°)
detrás	I5	Sólo durante la marcha atrás. Parar, después continuar con maniobra de evasión según lo planeado

Para completar esta tarea con elegancia, te vendrán muy bien una serie de nuevos elementos de programa (como por ejemplo los operadores: Y, O etc.) asociados al nivel 3 de ROBO Pro. En este nivel también se nos da la posibilidad de utilizar flechas naranjas para intercambiar datos entre distintos elementos. Por esta razón, debes empezar cambiando el nivel en el programa. A continuación lo más recomendable es que leas con atención todo el capítulo 5 del manual de ROBO Pro. Una vez lo hayas leído, estarás preparado para emprender la siguiente tarea.



Tarea 2 (nivel 3)

- Transforma el programa de detección de obstáculos para que el modelo reaccione como se muestra en la tabla anterior.
- Utiliza para ello las posibilidades que ofrece el nivel 3 de ROBO Pro.

Consejos:

- Con ayuda de los operadores, se consultan en un subprograma de consulta de obstáculos las diferentes combinaciones de activación de pulsadores posibles. Para cada posibilidad el subprograma posee una salida específica.

Entrada de datos PD = pulsador derecha

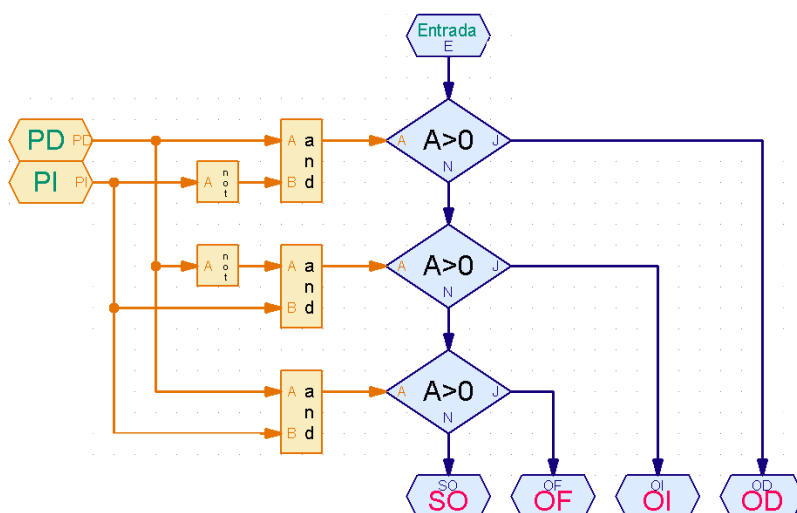
Entrada de datos PI = pulsador izquierda

Salida SO = sin obstáculo

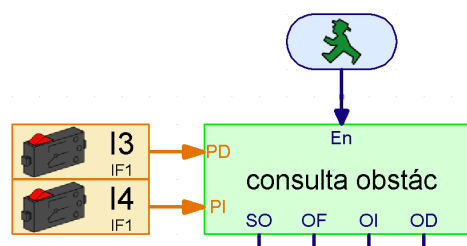
Salida OF = obstáculo al frente

Salida OI = obstáculo a la izquierda

Salida OD = obstáculo a la derecha

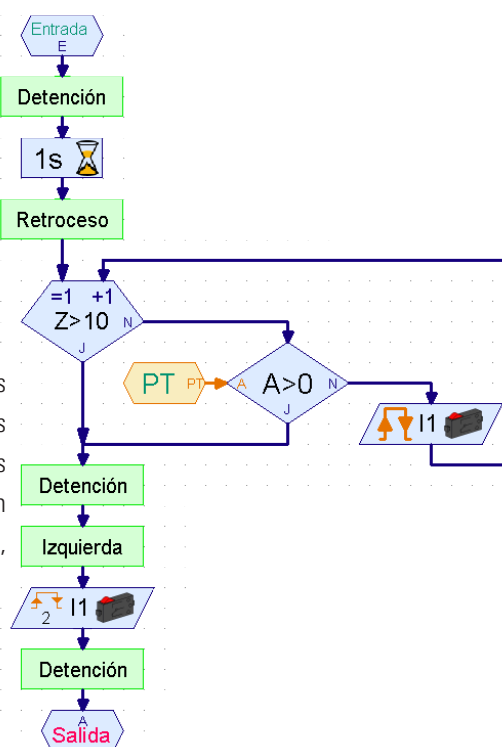


- Para poder saber rápidamente qué pulsadores se consultan, coloca los elementos de pulsador de color naranja en el programa principal y conéctalos con el subprograma mediante entradas de datos.

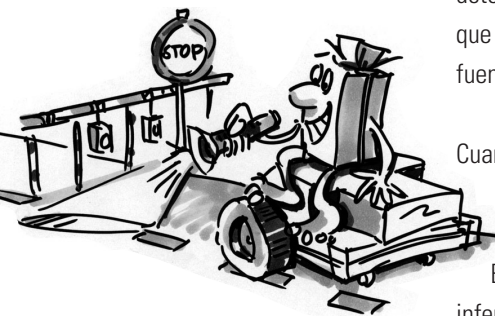


- En los diferentes subprogramas de superación de obstáculos se consulta I5 durante la marcha atrás. El modelo continúa retrocediendo hasta que alcance el número de impulsos fijado o se active I5. I5 se coloca otra vez en el programa principal para que se perciba rápidamente en qué subprogramas se consulta.
- Encontrarás el programa completo en el archivo Obstáculos 2.rpp.

Una ventaja de la técnica de programación utilizada en esta tarea es que te permite ver directamente en el programa principal qué pulsadores se consultan en cada subprograma. Si quieres modificar la entrada, lo haces solamente en un punto y no tienes que dedicarte a buscar en todos los subprogramas dónde puede estar oculto el pulsador que buscas. Además, los operadores permiten crear cadenas lógicas que se comprenden de un vistazo. En principio puede decirse lo mismo de los elementos de bifurcación, pero cuando llegamos a casos de consultas múltiples enseguida se complican las cosas y resulta difícil interpretar lo que se ve.



Robot con detección lumínica y de obstáculos



■ Todavía no hemos explorado ni mucho menos todas las posibilidades que ofrece el ROBO Mobile Set. Vamos a combinar ahora dos funcionalidades que ya hemos visto: la detección lumínica y la detección de obstáculos. Desde el punto de vista científico, el robot está equipado con dos modos de comportamiento. Puesto que los dos modelos de conducta no pueden estar activos simultáneamente, han de recibir diferentes grados de prioridad. El robot se encuentra normalmente en el modo de detección lumínica. Si detecta un obstáculo, es decir, un peligro para él, pasa automáticamente al modo que le permite evitarlo. Si todo está en orden, el robot puede entonces continuar su búsqueda de la fuente de luz.

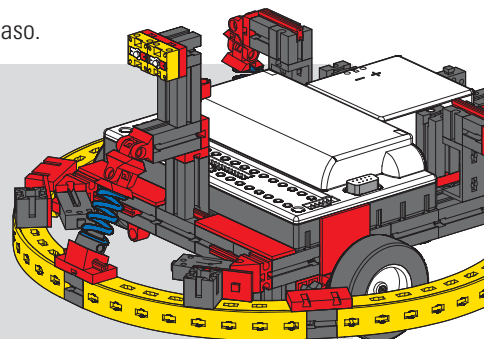
Cuando los programadores profesionales se enfrentan a una tarea tan complicada como esta, no se lanzan a programar como si tal cosa, sino que proceden a hacer uso de una u otra estrategia de programación. Uno de estos métodos es el llamado enfoque "top-down" (o "de arriba a abajo"). En esta manera de proceder se define la totalidad del sistema desde los niveles superiores hacia los inferiores, sin preocuparse al principio de concretar todos y cada uno de los detalles. Este es el método que precisamente vamos a utilizar nosotros en este caso.

Tarea 1 (nivel 3)

Enseña al robot a actuar de la siguiente manera:

- Busca una fuente de luz.
- En cuanto la localices, síguela.
- Si te topas con un obstáculo en el camino, evítalo.
- A continuación vuelve a buscar una fuente de luz.

Completa la tarea utilizando los elementos de programa del nivel 3 de ROBO Pro. Resuélvela empleando un enfoque "de arriba a abajo".



Consejos:

Para empezar, estructura la tarea en tres partes:

- Averiguar si el robot ve alguna fuente de luz (subprograma de detección lumínica)
- Preguntar si está chocando con algún obstáculo (subprograma de detección de obstáculos)
- En función de los resultados de esas dos preguntas, decirle al robot lo que tiene que hacer (subprograma de movimiento).

Para generar los subprogramas de detección lumínica y de obstáculos, debemos empezar por preguntarnos cuáles son las diferentes situaciones en las que se puede encontrar el robot. Asígnale a cada una de ellas un valor numérico y haz que se guarde en una variable con ayuda de un elemento de comando. Cada situación generará entonces una reacción, que se ejecutará en el subprograma de movimiento.

Subprograma de detección lumínica:

Nº	Situación	Estado de los sensores	Reacción
0	No hay fuente de luz disponible	I6=0; I7=0	Buscar luz
1	Fuente de luz justo ante el robot	I6=1; I7=1	Avanzar en línea recta
2	Fuente de luz a la izquierda del robot	I7=1	Describir curva a la izquierda
3	Fuente de luz a la derecha del robot	I6=1	Describir curva a la derecha

Subprograma de detección de obstáculos:

Nº	Situación	Estado de los sensores	Reacción
4	Obstáculo justo ante el robot	I3=1; I4=1	Evitar virando 90°
5	Obstáculo a la derecha del robot	I3=1	Evitar por la izquierda
6	Obstáculo a la izquierda del robot	I4=1	Evitar por la derecha

Ahora debes plasmar esta casuística en ROBO Pro en forma de elementos de programa.

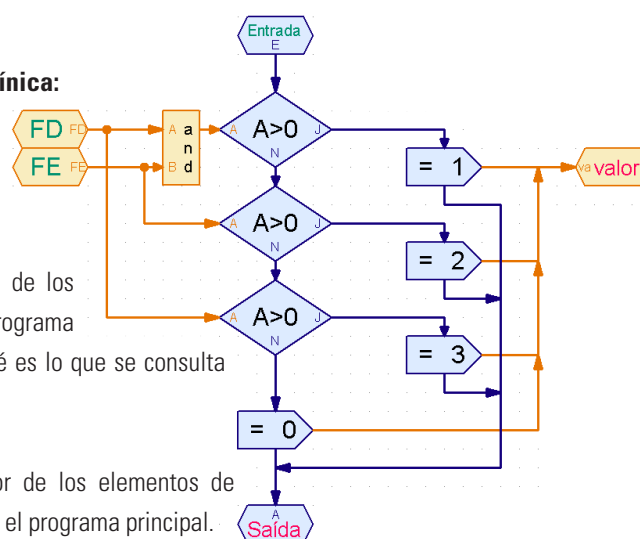
Subprograma de detección lumínica:

FD = fototransistor derecho

FI = fototransistor izquierdo

Coloca los elementos de consulta de los fototransistores también en el programa principal, para poder reconocer qué es lo que se consulta en cada momento.

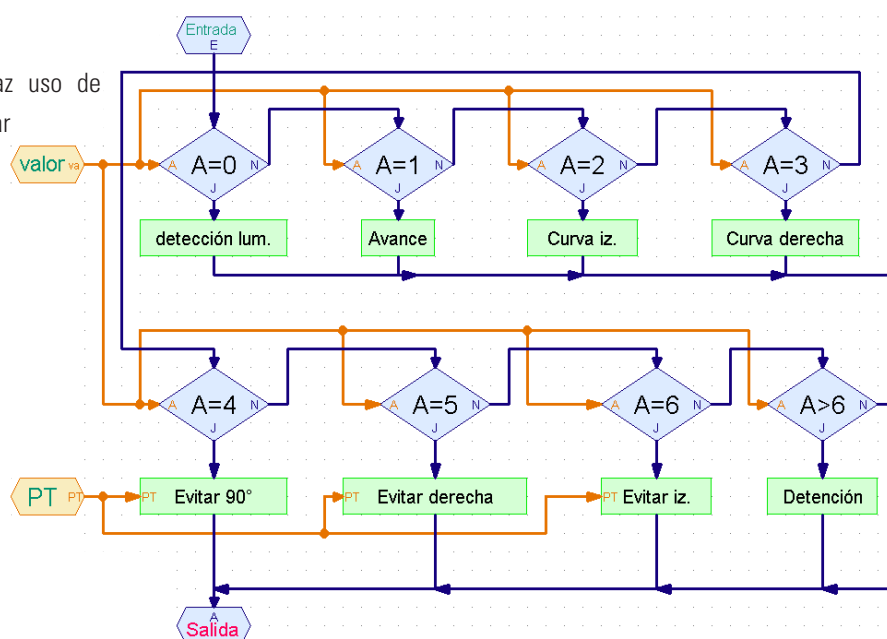
La variable que almacena el valor de los elementos de comando también debe incluirse en el programa principal. Será utilizada en más de un subprograma. Debes conectarla al subprograma mediante una salida de datos.



Elabora el subprograma de detección de obstáculos conforme el mismo principio que el de detección lumínica.

En el subprograma de movimiento haz uso de elementos de bifurcación para consultar cuál es el valor actual de la variable y programa la reacción correspondiente:

PT = pulsador trasero



Como detalle final resta ahora por crear los subprogramas que se usarán en este subprograma. Pero... un momento! Esa tarea está prácticamente ya terminada: El subprograma de detección lumínica, por ejemplo, podemos copiarlo del programa del modelo dotado de la funcionalidad de detección lumínica. Si ya no recuerdas cómo se hacía, léelo en el capítulo 4 del manual de ROBO Pro.

Pero ten cuidado:

En aquel modelo de detección lumínica, los fototransistores estaban conectados en las entradas I3 e I4, mientras que ahora tienen que ir en I6 e I7. Además, en aquel caso la cuenta de impulsos estaba asociada al pulsador I1 para el giro a la derecha y al pulsador I2 para el giro a la izquierda, mientras que ahora sólo tenemos a I1 para esa función, que por otra parte funciona igual de bien. En resumen: aunque puedes copiar el subprograma de detección lumínica, también tendrás que adaptarlo. Como la consulta del pulsador está oculta en el subprograma, es fácil pasarla por alto. Esto se evita colocando las entradas en el programa principal y conectándolas con el subprograma mediante entradas de datos. Pero en el caso de la detección lumínica esto deja de ser una posibilidad.

Los subprogramas que indican al robot cómo evitar los obstáculos ya están hechos, concretamente en el modelo dotado con la funcionalidad de detección de obstáculos. En este caso incluso el pulsador I5, un extra que se consulta en la marcha atrás, se deriva hacia fuera.

Encontrarás el programa completo en el archivo Obstáculos y luz.rpp.

El programa principal parece ya a simple vista extremadamente claro y sencillo. Y sin embargo detrás de los subprogramas se esconde mucho discurrir. Con ayuda del avance paso a paso asociado al enfoque de arriba a abajo, sin embargo, puedes un problema de semejante complejidad.

Una idea: si algún amigo tuyo tiene también un kit de construcción modular ROBO Mobile Set, podéis llevar el experimento con estos robots todavía más lejos. Simplemente tenéis que acoplar una fuente de luz a cada robot para que así los robots se busquen mutuamente.



Robot con detección de bordes

■ Una vez que hemos visto en el anterior ejemplo cómo se procede en la programación de un problema complejo, podemos pasar ya a ocuparnos de otro comportamiento de los robots móviles, un comportamiento de enorme importancia: el robot tiene que aprender a no caerse de la mesa. Si el robot choca contra un obstáculo, lo más probable es que no le ocurra nada malo. Pero si se cae de una mesa, desde una altura de casi un metro desde el suelo, es fácil que quede dañado en algún punto, por muy robustos que sean (como de hecho son) los componentes de fischertechnik. Por esta razón el robot posee unos sensores encargados de detectar la presencia de bordes o discontinuidades en la superficie por la que avanza. Cada detector de bordes consiste en un pulsador que es activado por una rueda giratoria. La rueda no pone resistencia y puede calcarse hacia dentro o salirse hacia fuera. En el momento en que la rueda vaya más allá del borde de la mesa, queda en el aire y sale hacia fuera al no haber ya nada (ningún suelo) que la mantenga calcada hacia dentro; entonces el pulsador deja de estar activado, el programa reconoce que el modelo ha perdido suelo y reacciona en consecuencia. El robot tiene en total 4 detectores de bordes que le permiten reconocer la discontinuidad de la superficie de apoyo tanto en la marcha hacia delante como en el retroceso, en cualquiera de los flancos. Este modelo, en cambio, no tiene pulsador de impulsos para la medición del trayecto. El camino que se recorre es controlado por la duración del encendido de los motores.

Construye, para empezar, el modelo tal y como se describe en las instrucciones de montaje.

Controla especialmente que los detectores de bordes se activan correctamente:

- cuando el modelo llega al borde de una mesa y el pulsador vuelve a presionarse con todo cuidado,
- cuando la rueda vuelve a tomar contacto con la mesa.

Es posible que tengas que ajustar alguno de los pulsadores un poquito hacia arriba o hacia abajo.

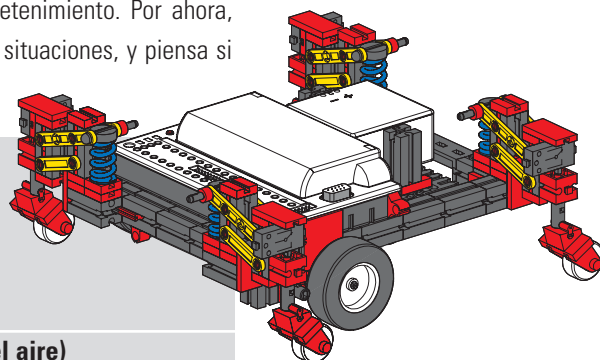


Tarea 1 (nivel 3):

- Piensa durante unos instantes cómo debe reaccionar el robot cuando llegue a una discontinuidad de la superficie de apoyo.
- Si lo piensas bien, llegarás a la conclusión de que existe un buen número de posibilidades de combinación en las que se pueden encontrar los sensores que detectan la pérdida de suelo. Puede que se active uno de los 4 detectores, 2 ó 3 al mismo tiempo o incluso los 4 a la vez.
- Cómo debería reaccionar el robot en cada caso?

Consejos:

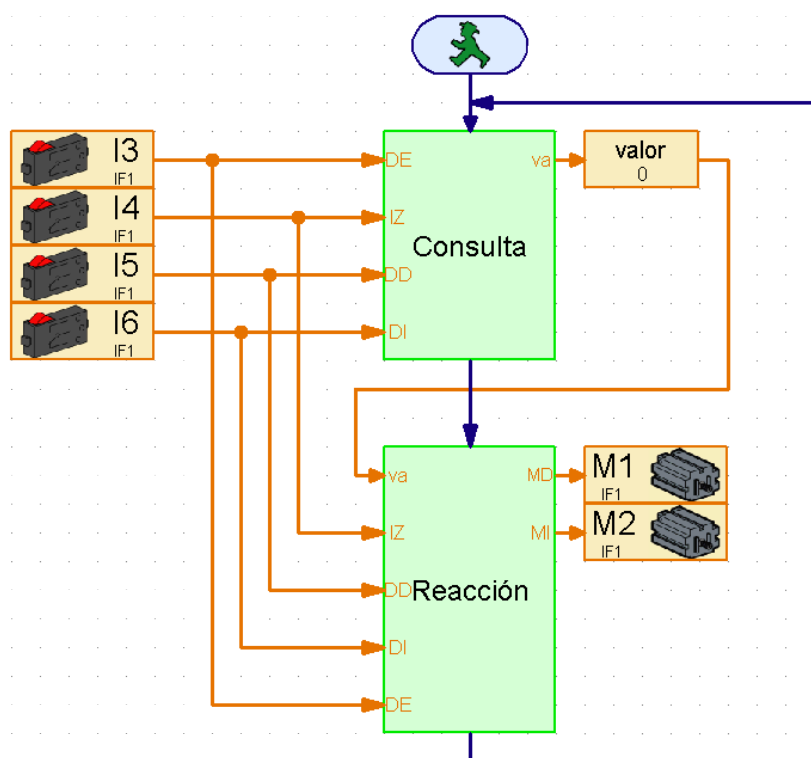
Encontrarás la solución en la tabla que sigue. Los sensores que se encuentran sobre la discontinuidad (pulsador = 0) están marcados con una equis. A cada combinación se le asigna un número. En el programa que crearemos más adelante asignaremos a cada posibilidad el valor numérico correspondiente. Este valor numérico le permite al robot reaccionar a la situación que se registre en cada momento. Más adelante volveremos a tratar esta cuestión con más detenimiento. Por ahora, discurre cómo tiene que encontrarse el robot para que se dé cada una de las situaciones, y piensa si reacciona correctamente.



Nº	delante a la derecha (I3)	delante a la izquierd (I4)	detrás a ala derecha (I5)	detrás a la izquierd (I6)	Reacción
0					Avanzar (no hay sensor en el aire)
1	●	●	●	●	Parar (los 4 sensores están en el aire)
2	●	●	●		Girar un poco a la derecha
3	●	●		●	Girar un poco a la izquierda
4	●		●	●	Girar un poco a la izquierda
5		●	●	●	Girar un poco a la derecha
6	●	●			Primero retroceso, después girar a la derecha
7	●		●		Girar un poco a la izquierda
8	●			●	Girar un poco a la izquierda
9		●	●		Girar un poco a la derecha
10		●		●	Girar un poco a la derecha
11			●	●	Avanzar un poco hacia adelante
12	●				Primero retroceso, después girar a la izquierda
13		●			Primero retroceso, después girar a la derecha
14			●		Avanzar un poco hacia adelante
15				●	Avanzar un poco hacia adelante

Parece bastante complicado, verdad? Pero no hay por qué preocuparse: para este modelo contamos con un programa ya completo que hace uso de todas las ventajas ofrecidas por ROBO Pro. Se llama Bordes.rpp.

Los elementos más importantes se encuentran en el programa principal, para que puedas comprender el conjunto del proceso. Las complejas consultas de los pulsadores están ocultas dentro de los subprogramas. Empecemos con el programa principal:



El proceso comienza con la consulta a los 4 pulsadores. A la izquierda ves qué pulsador se está consultando. Están conectados al subprograma mediante entradas de datos. El subprograma de consultas averigua qué pulsadores están oprimidos y devuelve el valor descrito en la tabla. Este valor se le asigna a la variable del mismo nombre, que puedes observar también en el programa principal. El valor de la variable se transmite al subprograma de reacción, que a continuación controla los dos motores en función de este valor. En el subprograma de reacción se leen igualmente los pulsadores, ya que los sensores de bordes también se consultan mientras el modelo realiza la maniobra evasiva.

Ahora puedes modificar sin más la posición de los pulsadores en la interfase e incluso las salidas de motores, sin tener que repasar punto por punto todos y cada uno de los subprogramas que esconden por alguna parte algún elemento de entrada o un símbolo de motor. Las entradas y las salidas aparecen una sola vez.

Esta técnica de programación es especialmente apropiada cuando queremos usar un subprograma en muchos modelos diferentes y no sabemos exactamente con antelación qué entradas y salidas de la interfase queremos utilizar en cada caso.

Si con la explicación te ha picado la curiosidad, no tienes más que echarle un vistazo al subprograma y tratar de comprenderlo. El principio de programación es semejante al que seguimos en el modelo con detección lumínica y de obstáculos.



Tarea 2 (nivel 3):

Carga el programa en la interfase y haz que el modelo se mueva por encima de una mesa.

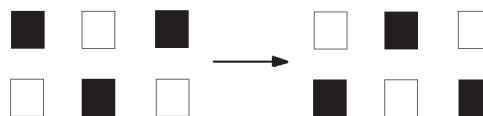
- Reacciona siempre como es debido?
- Debería actuar de un modo diferente en determinadas combinaciones de activación de pulsadores?
- Si es necesario, optimiza el programa.

■ Una vez que hemos hablado con todo detalle los robots que se mueven sobre ruedas, pasamos a otro modo de locomoción que también podemos utilizar en los robots móviles: la carrera o marcha.

El modo de andar de los insectos se antoja extraordinariamente apropiado como modelo para la locomoción de lo que podríamos llamar "robots insectoides". En lo que se denominado "paso a tres", siempre se encuentran separadas del suelo simultáneamente tres de las seis patas: la delantera y la trasera de uno de los lados y la intermedia del lado contrario.

Paso a tres

Las patas que quedan sobre el suelo (marcadas en negro) actúan como un trípode estable, de forma que el modelo siempre está seguro y no vuelca al andar.



Las patas del robot andador de fischertechnik están concebidas como mecanismos de cuatro articulaciones. El sistema de articulación utilizado en estas patas se conoce con el nombre de "mecanismo biela manivela". Impulsadas por una manivela, las piezas móviles del mecanismo efectúan un movimiento alternativo. La distancia entre cada articulación y el punto podal (el punto de apoyo en el suelo) se calcula de modo que el punto podal describa un movimiento elíptico al girar la manivela impulsora. De este modo se produce un movimiento que recuerda al de los pasos dados durante la carrera.

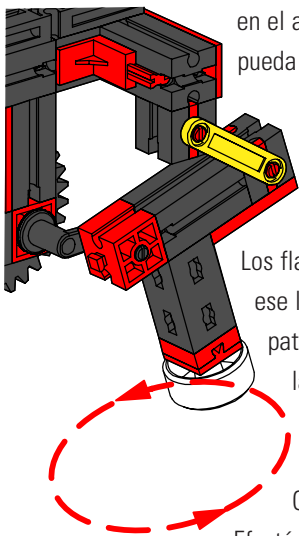
Las 6 manivelas que impulsan las patas deben ajustarse siguiendo al pie de la letra las instrucciones de montaje. La posición de las manivelas de las tres patas que permanecen en el suelo simultáneamente sigue siendo la misma, mientras que las manivelas de las tres patas que en ese momento se levantan en el aire giran 180°. La correcta posición de las manivelas entre sí garantiza que el modelo pueda correr con la secuencia de pasos adecuada, el paso a tres.

Las tuercas con las que se fijan las ruedas dentadas al eje deben estar bien apretadas para que las manivelas no se desajusten durante la carrera.

Los flancos izquierdo y derecho del modelo son impulsados por el motor que corresponde a ese lado (algo necesario para la carrera en curva). Por ello es importante garantizar que la pata intermedia de cualquiera de los lados se encuentra siempre en la misma posición que las patas delantera y trasera del lado contrario. De esta sincronización se ocupa el software gracias a los pulsadores I1 e I2.

Construye ahora el modelo tal y como se describe en las instrucciones de montaje. Efectúa una prueba de interfase para comprobar si todos los pulsadores y motores están bien conectados. Sentido de giro de los motores: izquierda = avance.

El robot andador



Tarea 1 (nivel 1):

Enseña al robot a correr.

- Programa el modelo para que avance en línea recta con paso a tres.
- Utiliza los pulsadores I1 e I2 para sincronizar las patas de la derecha y de la izquierda.
- Recuerda que las dos patas exteriores de un lado y la pata intermedia del otro deben estar siempre en la misma posición.

Consejos:

- Empieza por colocar las patas de ambos flancos en su posición inicial. Enciende después los dos motores (con sentido de giro a la izquierda).
- La carrera no debe continuar hasta que los dos pulsadores I1 e I2 estén oprimidos (necesitamos efectuar estas consultas en el instante en que el modelo haya de dar el segundo paso).
- Los motores deben seguir en marcha hasta que el pulsador correspondiente (I1 para M1, I2 para M2) vuelva a presionarse. Es importante que el modelo no dé el siguiente paso hasta que se opriman los dos pulsadores, porque así es como cada pata se coloca en posición correcta con respecto a las demás. Eso sí: para que funcione es imprescindible que las manivelas impulsoras de las patas estén bien ajustadas, conforme a lo se indica en las instrucciones de construcción.
- En ese momento puede reemprenderse el proceso y el robot dará el segundo paso. El modelo seguirá andando hacia adelante hasta que nosotros detengamos el programa.
- Encontrarás el programa completo en el archivo Robot andador 1.rpp.

Igual que con el modelo básico que se desplazaba sobre ruedas, puedes hacer que el modelo ande hacia la izquierda, hacia la derecha o hacia atrás variando el sentido de giro de los motores. Para contar los pasos puedes usar I1 o I2.

**Tarea 2 (nivel 2):**

- Programa el modelo para que dé 10 pasos hacia adelante, 3 hacia la izquierda, 3 hacia la derecha y otros 10 hacia atrás.
- Aplica un subprograma para cada dirección.
- Para contar los pasos utiliza el elemento bucle contador.

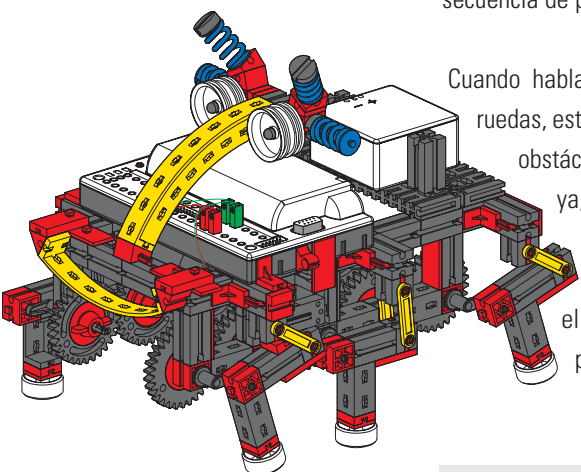
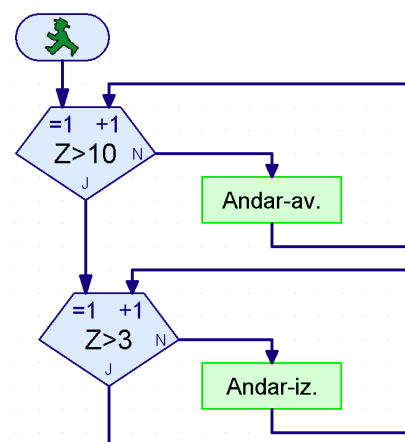
Consejos:

- Copiar sencillamente el programa "Robot andador 1.rpp" en un subprograma.
- Después copia este subprograma tantas veces como sea necesario para cubrir las distintas direcciones de carrera. Ve modificando en cada subprograma el sentido de giro de los motores para que el modelo se mueva en la dirección deseada.
- Utiliza el elemento bucle contador para contar el número de pasos por cada sentido de giro. El modelo da un paso por cada ejecución del subprograma; dicho de otra forma: si el programa hace 10 repeticiones del ciclo con el subprograma, el modelo da 10 pasos.

De esta manera puedes enseñar al robot a efectuar la secuencia de pasos que quieras (Robot andador 2.rpp).

Cuando hablamos de los robots que se desplazan sobre ruedas, estudiamos con detalle el tema de la detección de obstáculos. No vamos a repetir todo que se ha dicho ya, pero sí queremos proponerte que reproduzcas el mismo comportamiento en el robot andador.

Encontrarás todos los sensores necesarios en el kit de construcción modular; para la parte de programación, guíate por lo que hicimos con el robot sobre ruedas. Mucha suerte!



■ Las funcionalidades que ofrece la ROBO Interface no se limitan a las que se han mostrado hasta ahora en referencia a los robots móviles. Para explorarlas, son necesarios determinados componentes extra que no forman parte del kit de construcción modular. Como quiera que entrañan un enorme interés para la actividad con robots, no queremos dejar de presentarte algunos de ellos.

■ La ROBO Interface contiene un diodo que recibe los infrarrojos enviados por el emisor manual IR Control Set Art. N° 30344. Esto te permite consultar los pulsadores del emisor manual desde ROBO Pro como si fuesen entradas digitales para así poder, por ejemplo, encender y apagar los motores.

A modo de ejemplo hemos programado un mando a distancia para el robot andador. Con las 4 flechas ovaladas del mando a distancia podrás controlar el modelo para que avance y retroceda a izquierda y derecha. Antes, eso sí, deberás cargar el programa Robot andador IR.rpp en la interfase.

Otro programa genial también relacionado con el control remoto es el Mobile-Teach-IR.rpp. Este programa Teach-In te permite teledirigir uno de los robots sobre ruedas (por ejemplo el robot sencillo o el modelo básico). El modelo recuerda los trayectos recorridos y puede así repetirlos cuantas veces quieras. Los trayectos memorizados, sin embargo, se pierden al detener el programa.

Crear este tipo de programa es posible gracias al elemento Listado de ROBO Pro. En este elemento pueden almacenarse y recuperarse un gran número de valores (ver el manual de ROBO Pro). El programa en sí es bastante complejo, pero su aplicación no puede ser más sencilla:

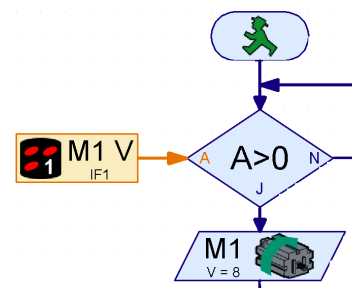
1. Carga el programa Mobile-Teach-In.rpp en la memoria FLASH de la ROBO Interface e inícialo.
2. Oprime el botón **M1** ► / ►► del mando a distancia. Empezará entonces el "proceso de aprendizaje".
3. Dirige el modelo en la dirección que desees con las flechas ovaladas.
4. Oprime el botón **M2** ► / ►►. El trayecto recorrido quedará memorizado.
5. Oprime el botón **M3** ► / ►►. El robot recorrerá el trayecto memorizado.

Con esta aplicación, programar robots se convierte en un juego de niños! Debes tener en cuenta, eso sí, que los trayectos memorizados se pierden al detener el programa con el botón Prog de la interfase.

■ El enlace de datos por radiofrecuencia ROBO RF Data Link Art. N° 93295 sustituye al cable de conexión del PC con la interfase estableciendo una conexión por radiofrecuencia para la transmisión de datos, lo que constituye una gran ventaja. Lo primero que debes hacer cada vez que cargues un programa en la interfase es conectar primero y desconectar después el cable en cuestión. Lo segundo, ejecutar el programa en modo online para detectar los posibles errores con mayor facilidad que si lo hicieses en modo de descarga. Finalmente controlamos desde la pantalla los robots móviles en modo online con un panel de control de ROBO Pro igual que hacemos con el mando a distancia de infrarrojos. A diferencia de lo que ocurre con el mando a distancia, desde la pantalla pueden verse también los datos que devuelve la interfase como por ejemplo valores de variables o entradas analógicas, tensiones de alimentación arrojadas por el acumulador o velocidades de los motores.

Posibilidades de extensión

Emisor de infrarrojos manual



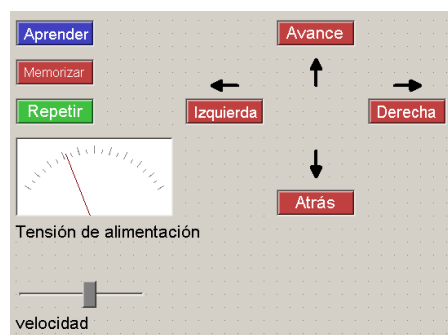
ROBO RF Data Link



Como ejemplo hemos modificado el programa Teach-In para manejar el robot sobre ruedas desde un panel de control. El programa se llama Mobile-Teach-RF.rpp. Huelga decir que también puedes probar con el cable, pero descubrirás que es bastante incómodo. El radio de acción del modelo quedará restringido, el cable acabará hecho un lío y el robot no será capaz de girar bien. Sin duda acabarás deseando hacerte cuanto antes con el RF Data Link.

Carga el programa Mobile-Teach-RF.rpp.

Cambia a panel de control en la barra de función del programa principal. A continuación inicia el programa en modo online. Ya puedes controlar y programar el modelo desde los botones del panel de control.



1. Oprime el botón "Aprender". Empezará entonces el "proceso de aprendizaje".
2. Usa las flechas para dirigir el modelo en la dirección que desees.
3. Oprime el botón "Memorizar". El trayecto recorrido quedará memorizado.
4. Oprime el botón "Repetir". El robot recorrerá el trayecto memorizado.

Aquí también se pierde el camino recorrido al cerrar el programa.

En el manual ROBO Pro encontrarás más información sobre los paneles de control.

ROBO I/O-Extension

■ Si por cualquier razón construyeses un modelo con tantos sensores y motores que las entradas y salidas de la ROBO Interface no fuesen suficientes, puedes conectar a la interfase una extensión ROBO I/O-Extension Art. N° 93294. De este modo obtendrás otras 8 entradas digitales, 4 salidas de motor y una entrada analógica de resistencia. A esta extensión de entradas y salidas puedes conectar todavía un segundo módulo, y a ese segundo módulo todavía un tercero, que podrás después gestionar en su totalidad con una ROBO Interface. En total tendrás a tu disposición 16 salidas de motor, 32 entradas digitales, 5 entradas analógicas de resistencia, 2 entradas analógicas de tensión y 2 entradas para sensores de distancia.

Si todo esto sigue quedándosete corto, todavía tienes la posibilidad de gestionar desde el PC varias interfases en modo online; por ejemplo, una en un puerto COM en serie y otra en un puerto USB, ó 2 interfases en el puerto USB, con un máximo de hasta 3 módulos ROBO I/O-Extension cada una. Da vértigo sólo de pensarlo! El funcionamiento del conjunto se trata también en el capítulo 6 del manual de ROBO Pro.

Búsqueda de errores

■ Experimentar es divertido... mientras todo va saliendo como debe. Casi siempre sale todo bien. Casi, pero no siempre.

Hasta que un modelo no nos da problemas, no comprobamos realmente si hemos comprendido el mecanismo en su totalidad ni detectamos los posibles fallos que podamos haber cometido.

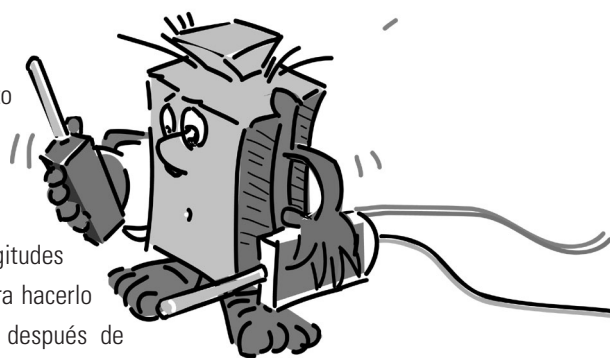
En el caso de los fallos mecánicos, al menos hay algo que se puede ver (componentes mal montados) o percibir (falta de suavidad en el funcionamiento). Pero si hablamos de problemas eléctricos, las cosas se complican.

Para localizar errores, los profesionales utilizan una serie de instrumentos de medición muy variados, desde voltímetros a oscilógrafos. Estos aparatos no están al alcance de cualquiera, por eso vamos a tratar de localizar y corregir errores con medios más sencillos.

Cableado

Antes de empezar con nuestros experimentos, tenemos que poner a punto algunos componentes del kit de construcción modular fischertechnik. Por ejemplo, hay que fijar los conectores que se proporcionan a cada segmento de cable.

Lo primero que debemos hacer es cortar los cables. Para ello medimos las longitudes fijadas y cortamos los segmentos indicados. Después hay que probarlos; para hacerlo necesitaremos el acumulador y la lámpara. Si la bombilla se enciende después de conectarla al acumulador, el cable está en perfecto estado. También tenemos que verificar que los códigos cromáticos están correctos: conectores rojos con cables rojos, conectores verdes con cables verdes.



Prueba de interfase

Si los programas (incluso los que se proporcionan ya hechos) no funcionan con nuestro modelo, procedemos a realizar una prueba de interfase. Esta utilidad nos permite probar por separado las entradas y las salidas. Funcionan los sensores? Giran los motores en el sentido correcto? En todos nuestros robots móviles los motores se conectan para que al girar a la izquierda la rueda o la pata se mueva hacia adelante. Si aquí está todo correcto, buscamos una causa mecánica.

Contactos flojos

Un fallo peor es la presencia de contactos flojos. Por un lado, es fácil que los conectores queden sueltos en las hembrillas. Si este es el caso, podemos abrir un poco las pestañas del conector con un destornillador pequeño. Hay que tener mucho cuidado, porque si nos excedemos podemos llegar a romper los contactos o, cuando menos, hacer que después sea difícil introducirlos en las hembrillas.

Otra posible causa de que los contactos queden flojos es que los elementos de sujeción de los conectores y los cables se hayan aflojado, por eso hay que cerciorarse de que estén bien apretados. Conviene aprovechar la ocasión para comprobar también que no se haya roto ninguno de los hilos de cobre.

Cortocircuitos

Puede darse también el caso de que un cableado mal puesto produzca un cortocircuito que impida desde ese momento que el sistema funcione como debe. El acumulador trae incorporado un sistema de seguridad que interrumpe el flujo eléctrico en caso de que se produzca una subida de tensión o un

recalentamiento. Así mismo, las salidas de la interfase se cierran también en caso de sobrecalentamiento.

También se puede producir un cortocircuito si el tornillo con el que se sujeta el cable no queda bien asegurado y sobresale por encima del borde del conector. Si se introducen dos conectores en dos hembrillas directamente contiguas de la interfase y los tornillos se tocan entre sí, se produce el cortocircuito. Por eso hay que poner especial cuidado en que los tornillos queden siempre perfectamente ajustados y en introducir los conectores de forma que los tornillos no puedan contactar entre sí.

Alimentación de corriente

Si durante el funcionamiento de los aparatos se producen interrupciones inexplicables, puede deberse a que el acumulador esté casi agotado. Justo en el momento de tener que mover una carga (al encender un motor) cae la tensión, lo que dispara la función reset del procesador de la interfase. Si se enciende el LED rojo de la ROBO Interface es señal de que la tensión de alimentación es demasiado baja. Es el momento de recargar el acumulador.

Errores de programación

Si en los programas escritos por ti mismo surgen errores que no consigues explicarte, es conveniente, para estar seguro, que ejecutes uno de los programas que se proporcionan ya hechos, el más parecido posible al que te dé problemas. Esto te permite descartar que los fallos se estén produciendo por defectos mecánicos o eléctricos. En el modo online es posible ir siguiendo los pasos del programa en la pantalla. Si el programa llega a un punto del que no consigue pasar, podemos tratar de buscar la causa en ese punto; por ejemplo, puede ser que haya una entrada o un motor mal seleccionado, que en una bifurcación se consulte el valor que no es o que estén invertidos los contactos S/N.

Si todo esto sigue sin dar resultado, existe la opción de contactar con el servicio técnico de fischertechnik (correo electrónico info@fischertechnik.de).

O de visitar la página de fischertechnik en internet: www.fischertechnik.de, donde encontrarás un foro, un chat, un tablón de anuncios de compra y venta y una galería de imágenes, y donde tendrás la oportunidad de afiliarte sin coste alguno al Fischertechnik Fanclub.

Te deseamos muchas horas de diversión con el ROBO Mobile Set. Estamos seguros de que serán muchas las veces que exclames "eureka!".

