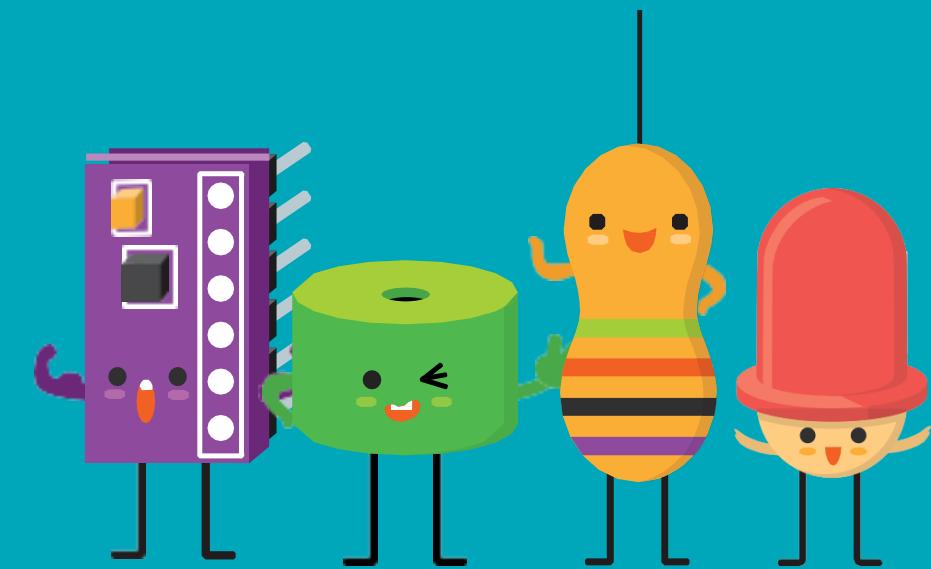


Samsung Innovation Campus

Curso de codificación y programación

Samsung
Innovation
Campus

Capítulo 6
Arduino
Codificación y programación





Three colorful cartoon characters standing together. From left to right: a yellow character with a striped pattern, a green character with a circular shape, and a purple character with a rectangular shape.

CLASE DE INTRODUCCIÓN

01

01. Una navegación sencilla
 02. El circuito electrónico
 03. Led Parpadeante
 04. Como controlar un led
 05. Diseño creativo 1 (Hablemos con luces)

SEMANA

01

Una navegación sencilla

Objetivos de aprendizaje

- Observar cómo las placas de creación se usan en varias aplicaciones y pensar en futuros proyectos.

Materiales para preparar

Hoja A4, marcadores de colores, cámara, led (r, g, b), batería de mercurio, imán, pistola de pegamento y pizarrón grande.



LECCIÓN

¡Hola a todos!

Durante las próximas quince semanas, tomarás esta clase. Espero que después de ver el título de este libro, pienses en los contenidos que aprenderás en esta clase. ¿Cómo crees que será esta clase? Podría ser una pregunta intrigante.

La mayoría de los estudiantes no ha tomado una clase como esta antes.

Es por eso que intentaré explicarte de qué se trata a partir de ahora.

¿Alguna vez creaste algo en tu vida? Algunos estudiantes podrían tener experiencia cocinando, creando robots o tejiendo bufandas para el invierno. Los estudiantes a los que les gusta la tecnología electrónica o el ensamblado podrían haber creado elementos divertidos con un conjunto de ensamblaje de radio, un kit de ciencias o un Lego Mindstorm.

En esta clase, queremos crear algo con una computadora de mano. Aún no tienes que saber detalles sobre esta pequeña computadora. Aprenderás paso a paso.

¿Una computadora de mano? Usa la imaginación.

Existen muchos tipos de computadoras, desde computadoras de escritorio hasta súper computadoras. Los teléfonos inteligentes y las computadoras pequeñas también son computadoras. Existen muchos tipos de computadoras que no conoces.



Las computadoras existen en cientos de distintas formas con distintos propósitos. ¿Con qué propósitos usas la computadora principalmente? Es probable que la uses para documentarte, editar fotos, navegar en Internet, jugar, etc.

Entre las varias formas de computadoras, usaremos una computadora con rendimiento limitado.

Entre ellas, aprenderemos la forma más sencilla de usar una computadora.

Cuando termines este curso, podrás darle órdenes a una computadora.

Es como entrenar a un perro.

Entonces, ¿qué podrás crear con una computadora tan pequeña?

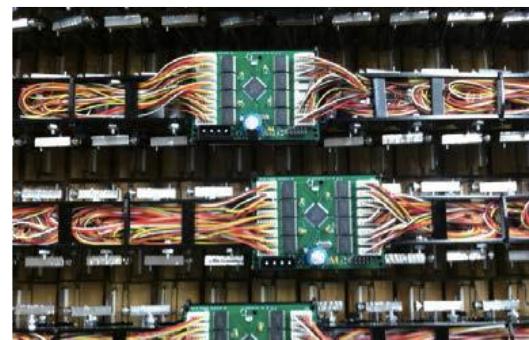
Por ejemplo, ¿cómo apagas las luces cuando llegas a tu casa? En la mayoría de los casos, enciendes las luces en la habitación oscura al presionar el interruptor de la pared. Sin embargo, ¿qué harías si te resultara inconveniente encender la luz con el interruptor y prefirieras que la luz se encendiera automáticamente al abrir la puerta?

Piénsalo un momento. Cuando termines este curso, verás que existen muchas formas posibles. No te preocupes si el material parece muy difícil. Esta clase te permitirá aprender de muchos ejercicios para crear, arreglar y confirmar. Sé que disfrutarás este proceso de aprendizaje.

Antes de comenzar, te presentaré algunos proyectos creados con una computadora de mano que te ayudarán a comprender.

El espejo de madera de Daniel Rozin

El espejo de madera de Daniel Rozin es una obra de arte conceptual hecha con madera. Cada pieza de madera está unida a un submotor que refleja instantáneamente la superficie. Los sub motores unidos a las piezas de madera reflejan el rostro de una persona y se mueven según la sombra del reflejo del rostro.



Sitio web: <http://www.smoothware.com/danny/>
Video: <https://www.vimeo.com/7820888>

La caja musical de Jin-Yo Monk

La caja musical de Jin-Yo Monk es una obra de arte que emite sonidos cuando los espectadores ingresan un patrón led al girar una manivela. La forma en que interactúa por Internet da la impresión de una caja musical.



Sitio web http://jonpasang.com/wp/?page_id=183
 Video http://www.youtube.com/watch?v=P2_jxY9hRVs#t=250
<https://vimeo.com/46857169>

La mayoría de las obras de arte presentadas son piezas interactivas que usan tecnología digital. Este campo se denomina arte interactivo. Los artistas usan varios tipos de tecnologías para crear sus obras.

Arte interactivo

Gracias al desarrollo de la tecnología de la información, se han dado a conocer muchos medios digitales en la actualidad. La televisión, los videos y los teléfonos inteligentes, que puedes encontrar fácilmente a tu alrededor, son distintos tipos de medios digitales. El género artístico que los artistas modernos crean mediante medios digitales se denomina “arte de los medios”. Las famosas obras de arte en video de Baek Nam-jun son un ejemplo del arte de los medios. El arte interactivo se refiere a las obras que responden ante la audiencia o que interactúan con ella.



Te presentaré algunos proyectos más interesantes además de las obras de arte.

El arpa láser de Stephen Hobley

El arpa láser es un instrumento hecho de luz. La luz se dispara hacia el cielo y cada vez que la tocas, el sonido que emite cambia.

Es un instrumento virtual que usa un láser y opera como un instrumento con un teclado.

Sitio web <http://www.stephenhobley.com/blog/laser-harp-2009/>
 Video <http://www.youtube.com/watch?v=Yta0aJrbOxU>



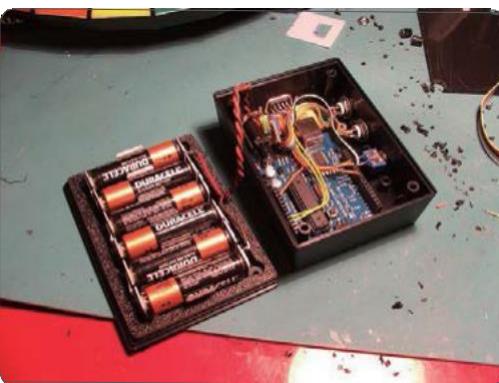
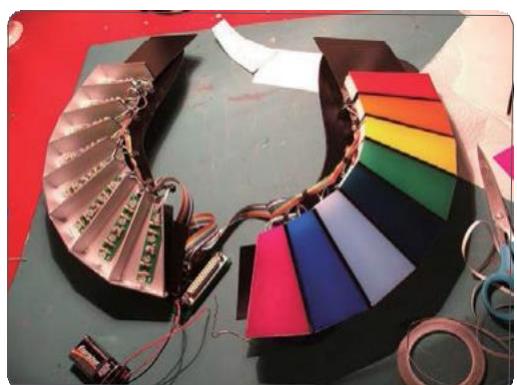
“Pinocchio” de Adam Ben-Dor, Shan Shan Zhou y Joss Dojet

La lámpara Pinocchio es una versión viviente de la famosa lámpara Luxor, conocida por su logotipo. Puede reconocer los rostros de las personas con la ayuda de una cámara web.



Sitio web: [Vi](http://www.ben-dror.com/pinokio/)
deo: <https://vimeo.com/53476316>

El casco led de Daft Punk, creado por Volpin Props



Sitio web: <http://www.volpinprops.com/daft-punk-final-2/>
Video: [http://www.youtube.com/watch?v=H0TBZeCgLOE](https://www.youtube.com/watch?v=H0TBZeCgLOE)

Botanicalls, la alerta del estado de las plantas



Sitio web: <http://www.botanicalls.com/>

Video: <http://www.youtube.com/watch?v=af6cayzWW1Y>

Por último pero no menos importante, quisiera presentarles a algunos amigos de su edad. Sylvia Todd y Quin Etnyre ya tienen muchos fanáticos.

Q Techknow de Quin Etnyre

Quin es un niño de trece años obsesionado con la ingeniería electrónica que se desempeña como director ejecutivo de la empresa de electrónica Q Techknow, que vende kits electrónicos y distintos tipos de sensores. Quin da clases sobre dispositivos electrónicos regularmente y sus charlas se destacan por tener una mejor calidad que las clases dictadas por adultos. Su representante es Fuzzbot, un robot completamente autónomo.



Sitio web de Q Techknow de Quin: <http://qtechknow.com/>

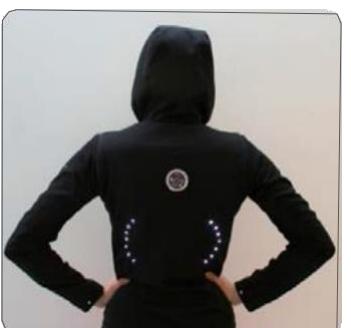
Vestido de galaxia de CuteCircuit

Sitio web: <http://cutecircuit.com/collections/the-galaxy-dress/>

Video: <http://www.youtube.com/watch?v=rX9FOGFxN9A>



La chaqueta de ciclismo intermitente



Es una chaqueta para ciclistas que tiene luces led instaladas en la parte de atrás, similares a la luz de giro de los autos.

Sitio web: http://web.media.mit.edu/~leah/LilyPad/build/turn_signal_jacket.html

Video: <http://www.youtube.com/watch?v=ZtNEPkwCfxA>

El programa de creación súper increíble de Sylvia Todd

Sylvia filmó y compartió sus proyectos para que todos puedan verlos. Este proyecto es fácil de seguir, incluso para niños, y ya tiene más de 1,5 millones de visitas. Ganó la medalla de plata en la Competencia Nacional de Robots y recientemente le presentó el WaterColorBot que inventó para el presidente Barack Obama.



Sitio web de Sylvia: <http://sylviashow.com>

Puedes encontrar más información en <http://popsci.hankooki.com/Article/ArticleView.php?UID=10138122>

Hemos presentado a nuestros amigos y los proyectos que usaron computadoras de mano. ¿Cuál es tu primera impresión? La tecnología se ha desarrollado tanto, que cualquiera puede crear nuevos productos con facilidad. Al participar en los proyectos de esta clase, podrás crear lo que deseas al igual que Quin y Sylvia.

¿Comenzamos?



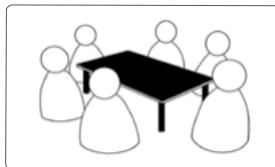
LECCIÓN

Ahora, empecemos con los rompehielos.

Como la palabra lo dice, se trata de romper el hielo. Pero no literalmente. Al igual que al romper un cubo sólido de hielo, se crea un ambiente cálido mediante la actividad corta llamada "romper el hielo".

1 Sentarse en grupos

Siéntense entre 4 o 5 compañeros enfrentados entre sí.



2 Buscar tres palabras clave

Escribe tu nombre en grande en una hoja A4. Escribe tres palabras que te describan (puede ser tu película favorita, tu profesión soñada, un pasatiempo, un talento, etc.).

3 Hacer preguntas

Cuando termines, pásale la hoja a la persona que está a tu derecha. Luego, una persona selecciona una de las palabras clave y hace una pregunta sobre esa palabra.

4 Responder

La persona que hace la pregunta dibuja algo o escribe un comentario debajo de la palabra clave correspondiente. Después de varias preguntas y respuestas, otro miembro del equipo hace una pregunta. La respuesta a cada pregunta se debe completar en menos de tres minutos.

5 Turnarse y presentarse

Después de una ronda de preguntas y respuestas, pasa la hoja A4 a la persona que está a tu derecha. Si las hojas se van pasando, volverás a recibir tu hoja y todos habrán analizado las tres palabras, ¿verdad? Si no se analizó alguna palabra clave, debes dar una explicación.

6 Terminar

Cuando termines de aprender sobre tus compañeros, la actividad rompehielos habrá terminado. Avanzaremos a la próxima actividad con el mismo grupo.

Podrás crear proyectos diversos y divertidos solo con ledes. Un led es como una pequeña bombilla que emite luz. Observa lo fascinantes que son las luces led en distintos juegos.

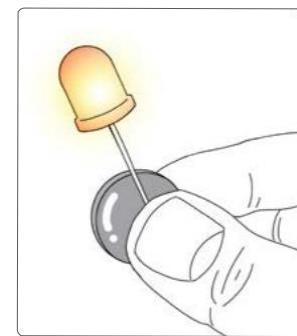
Práctica

Cubo con luz led

1 Prepara el led, la batería de mercurio y el papel sobre tu escritorio.

2 Para que el led emita luz, debe estar conectado

a la batería. El led y la batería de mercurio tienen electrodos, por lo que tendrás que distinguir entre el ánodo (+) y el cátodo (-). La pata más larga es la positiva. Verás el “+” en la batería de mercurio, por lo que será sencillo distinguir cuál es el lado positivo. Une los ánodos del led y del mercurio, y coloca la batería entre las dos patas del led y sostenlo entre el pulgar y el índice. El led comenzará a brillar.

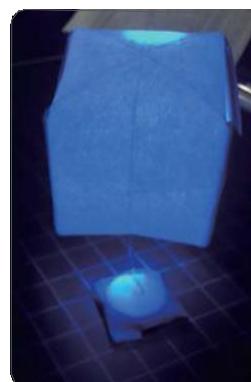


3 Crearemos un cubo con una hoja de papel.

Prepara un papel cuadrado y consulta en el material sobre origami para aprender a formar un cubo con el papel.

Inserta el led en el cubo terminado a través de un pequeño agujero en el fondo. Podrás crear una luz led como la de la imagen que se encuentra a continuación.

¿Juntamos las luces que crearon tus amigos? Apaguemos la luz y disfrutemos de las luces led.

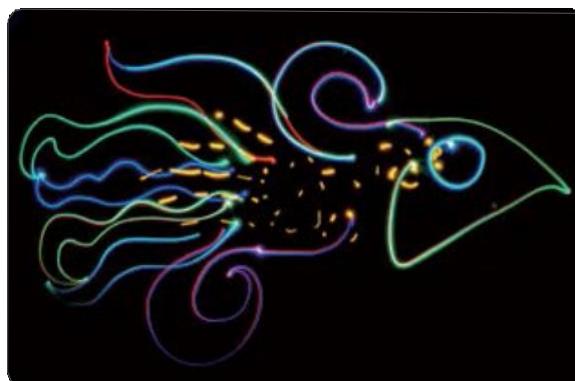


Práctica

Dibujar con el led

Puedes dibujar con ledes si usas la técnica de larga exposición de la cámara.

En la oscuridad, con el obturador de la cámara a velocidad lenta, puedes dibujar en el aire con el led frente al lente de la cámara. No uses un único color. Puedes dibujar con muchos colores de ledes. ¿No es más divertido si trabajas en este ejercicio con tus amigos?



¿Te puedes electrocutar?

El led es un buen conductor de electricidad y la mano humana también, pero tiene alta resistencia, por lo que no permite que la electricidad fluya hasta tu mano. Además, la batería de mercurio que usamos produce un voltaje bajo de 3 V, por lo que no recibirás una descarga eléctrica.

Práctica

Led Throwies

Podrás realizar actividades divertidas al colocar un imán en el led. Fija los imanes a la batería de mercurio que se conecta al led.

Ahora, arroja el led sobre la placa de metal o en el pizarrón en el que se pegará el imán. Los ledes se ven más glamorosos de noche que de día.

El próximo proyecto es una aplicación de *Led Throwies* desarrollados por el *Graffiti Research Lab (GRL)*.



¿Cómo funciona?

El led es el tipo de luz que encontrarás con mayor facilidad entre los materiales que emiten luz. Puedes crear un proyecto fantástico con solo ledes, que encontrarás fácilmente.

Acabamos de crear un pequeño proyecto con solo un led para entrar en calor. Ahora, crearás proyectos más divertidos por tu cuenta. Más adelante, tendrás grandes habilidades para crear el proyecto que quieras tú solo. Si te esfuerzas un poco más, podrás crear muchos proyectos divertidos, ¿verdad?

Ahora, empecemos de verdad.



Las baterías usadas contienen materiales peligrosos, por lo que debes desecharlas de manera segura y ecológica. Puedes reciclar las baterías usadas en contenedores para baterías.

REVISIÓN

Repasso

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.

SEMANA
02

Entender El circuito electrónico

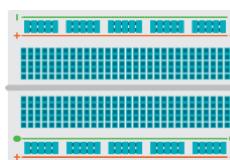


LECCIÓN

Objetivos de aprendizaje

- Entender el circuito electrónico con componentes simples.
- Crear varios circuitos en la placa de pruebas.

Materiales para preparar



Placa de pruebas



Led



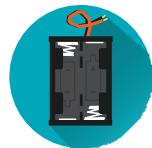
Resistencia de $330\ \Omega$



Cable eléctrico



1 interruptor



Soporte para baterías AA

¿Cuál es el componente más fácil de encontrar y más fundamental para la clase de CREACIÓN?

Para construir el circuito más sencillo, necesitas tres partes además de la placa de creación. La placa de pruebas es una base de construcción que une los ledes que emiten luces, las resistencias que regulan el flujo de corriente eléctrica y otras piezas con el exterior de la placa de creación.

Led

El componente que se ilumina cuando fluye la corriente se denomina **diodo emisor de luz (Light-Emitting Diode, LED)**. Los ledes se presentan en varios tamaños y colores según cada uso y son algo que utilizamos en nuestra vida diaria, como los televisores led, los monitores led y las lámparas led.



Primero, analizaremos la forma de led que queremos usar. El led está formado por lentes cilíndricas que emiten luz y tienen dos patas. Cuando conectas la electricidad a las dos patas, se enciende. **El conductor más largo es positivo (+) y el más corto es negativo (-).** Tienes que tener cuidado a la hora de conectar los materiales con terminales positivos y negativos porque es importante tener en cuenta la dirección en la que los conectas. Incluso podrías dañarlo. Los materiales con terminales positivos y negativos funcionan solo cuando la corriente fluye en la dirección correcta.

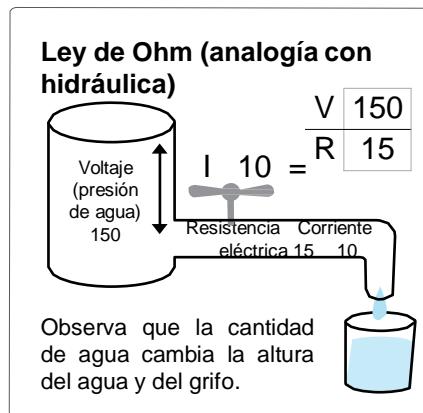
Cable

El **cable eléctrico** es el material que ayuda a que la corriente eléctrica fluya. Es como las cañerías que ayudan a que el agua fluya. Sirve para conectar los materiales como la placa de creación, el led y el interruptor. El cable eléctrico en sí no tiene electrodos positivos o negativos, por lo que no tienes que preocuparte por la dirección en la que lo conectas.

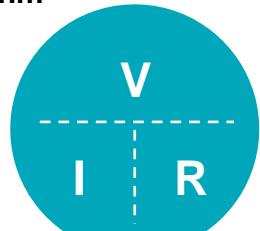
Resistencia

La **resistencia** es un componente que interfiere en la corriente. En otras palabras, la cantidad de corriente eléctrica disminuirá cuando exista una resistencia en comparación con una falta de resistencia.

Sería más fácil de comprender si piensas en la cantidad de agua como el voltaje, el grifo como la resistencia y la cantidad de agua que corre como la corriente eléctrica. Cuando la cantidad de agua (voltaje) es constante y aumentas la presión del grifo (resistencia), la cantidad del flujo de agua (corriente eléctrica) disminuye.



Ley de Ohm



$$V = IR \quad | \quad I = V/R \quad | \quad R = V/I$$

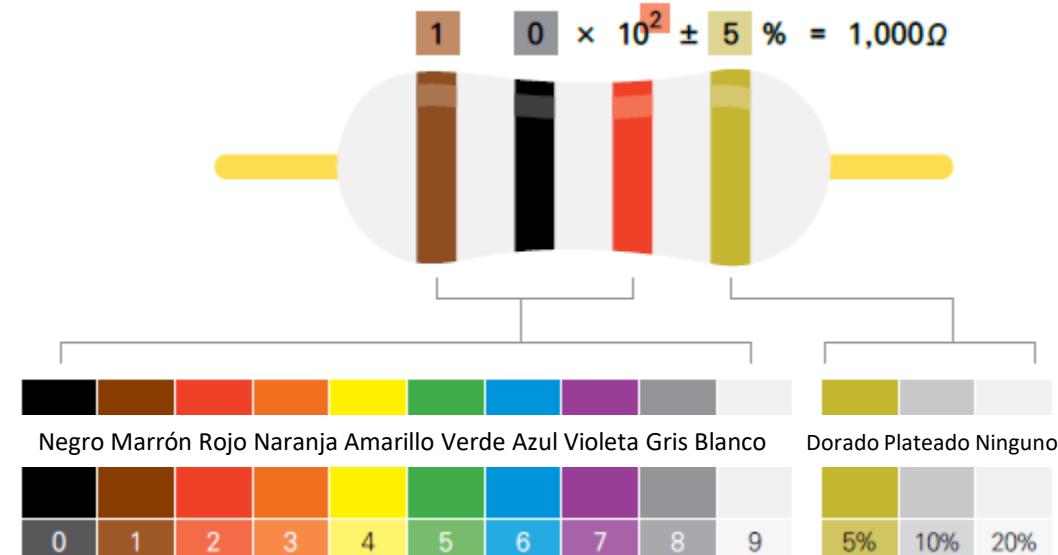
- Voltaje (V): (unidad: voltio o V)
→diferencia en el potencial eléctrico entre dos puntos.
- Corriente (I): (unidad: amperio)
→cantidad de carga por unidad de tiempo.
- Resistencia (R): (unidad: ohmio o Ω)
→grado en el que un componente impide el paso de la corriente.

El motivo por el cual la resistencia es importante es porque la entrada de corriente eléctrica puede dañar el componente. Es por eso que necesitamos usar la resistencia para modificar la cantidad de corriente y para mantener un nivel estable de voltaje.

Existe un código de colores en la resistencia que representa la intensidad según los colores. Las resistencias son pequeñas y es por eso que se representa con bandas de colores en vez en texto.

Consulta la tabla del código de colores para determinar el valor de la resistencia. Escribe los números de la primera y la segunda banda de colores en orden y multiplica los dos dígitos por diez elevado a la potencia del número de la tercera banda. La cuarta banda representa la tolerancia y la precisión.

La dirección en la que debes leer el código de colores podría confundirte, pero solo debes recordar que los colores dorado y plateado deben estar del lado derecho y empezar a leer



¿Puedes calcular el valor de esta resistencia?

La banda de colores tiene naranja, naranja y marrón en secuencia, lo cual representa 3, 3 y 1, respectivamente. Si se lo calcula, es $33 * 10^1 = 330 \Omega$. Cuando la tensión varía de 5 a 9 V, es adecuado usar entre 200 y 300 Ω .

Placa de pruebas

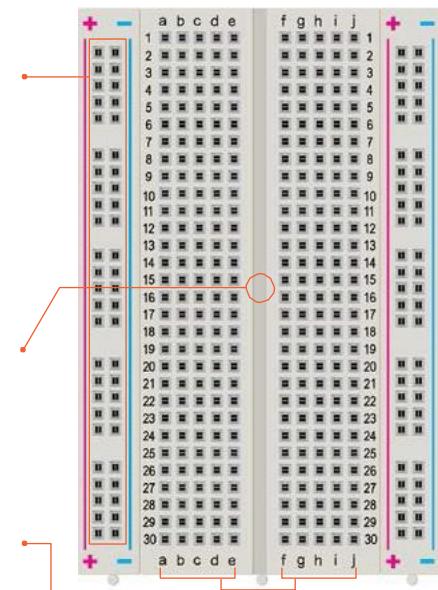
La **placa de creación** es pequeña. Esto se debe a que, en proyectos reales, es mejor usar placas de pruebas pequeñas, pero es difícil conectar los componentes del hardware directamente. Por lo tanto, usamos la placa de pruebas para expandir las placas de creación. La ventaja de la placa de pruebas es que puedes conectar los componentes fácilmente sin soldar. Puedes conectar y desconectar los componentes eléctricos fácilmente, como el led, la resistencia y el interruptor. En un circuito electrónico, existe una regla que establece que la línea roja representa un terminal positivo, mientras que la línea negra o azul representa el terminal negativo. Por lo tanto, puedes distinguir los terminales positivos de los negativos gracias a los colores. Necesitamos conectar el terminal positivo de la batería con el rojo y el negativo con el azul.

Composición de la placa de pruebas y método de conexión

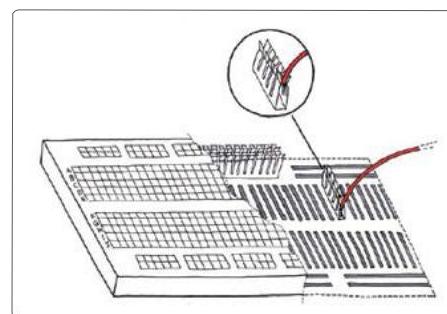
Los orificios de cada fila vertical están interconectados dentro de la placa de pruebas y se usan para establecer una conexión a tierra o una conexión eléctrica del circuito. Son los suministros de energía como los terminales positivos y negativos de las baterías. Tienes que encontrar el terminal correcto cuando conectes los componentes con terminales positivos y negativos.

El lado izquierdo y derecho se ven exactamente iguales y cumplen la misma función, pero no están conectados entre sí. Para expandir la conexión, debes conectar ambos lados con componentes como el cable o la resistencia.

Los cinco orificios de cada fila horizontal están interconectados por dentro. En cada fila horizontal, el componente que deseas conectar con el componente conectado en el lado izquierdo debe estar conectado en "b", "c", "d" o "e" en la izquierda. No se deben conectar en "f", "g", "h" ni "i" a la derecha.



Sería más fácil de entender si observas la estructura interna de la derecha. Como puedes ver en la imagen, los lados izquierdo y derecho tienen una división en el medio. Cada grupo de cinco orificios en el medio se conecta horizontalmente, pero no verticalmente. Por lo tanto, cuando quieras conectar los componentes o los cables, debes conectarlos entre la misma fila horizontal.



Primero, verifica en qué fila está conectado el componente y conecta el otro componente en el otro orificio de la misma fila horizontal que el primero. Como no están conectados verticalmente, debes conectarlos con un cable. Ahora, construiremos varios circuitos para entender mejor la placa de pruebas.

Práctica

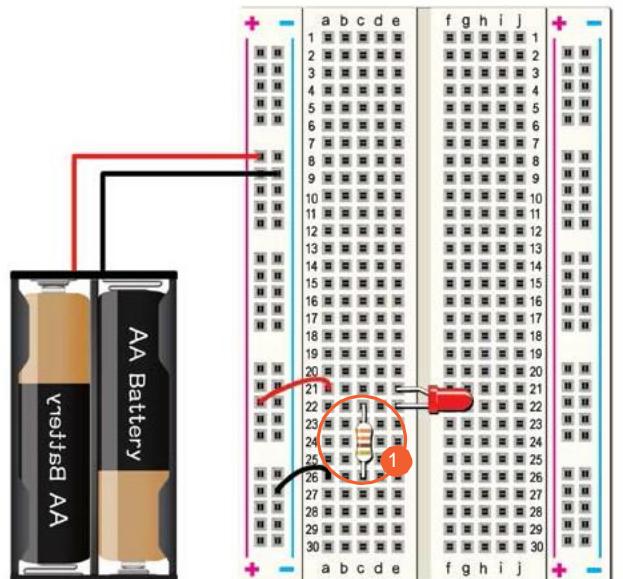
Encender el led

Objetivos del proyecto

Encender el led al conectar el circuito.

Expresión del hardware

El led empieza a brillar cuando conectas la batería o el suministro eléctrico en la placa de pruebas. En este caso, la corriente eléctrica fluye en la siguiente dirección: batería - placa de pruebas →led.



Debes observar más detenidamente la longitud de las patas del led al conectar los terminales positivos y negativos. El lado más doblado del diagrama del circuito es el terminal positivo.

Práctica

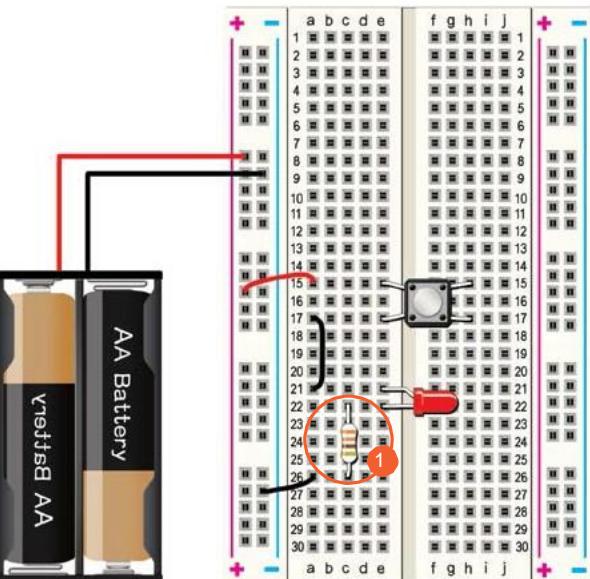
Usar el interruptor para encender el led

Objetivos del proyecto

Encender el led al conectar el circuito y accionar el interruptor

Expresión del hardware

El led no se enciende incluso cuando se construye el circuito como en el diagrama que se encuentra a continuación. Esto ocurre debido al interruptor. Debes accionar el interruptor para encender el led.



Práctica

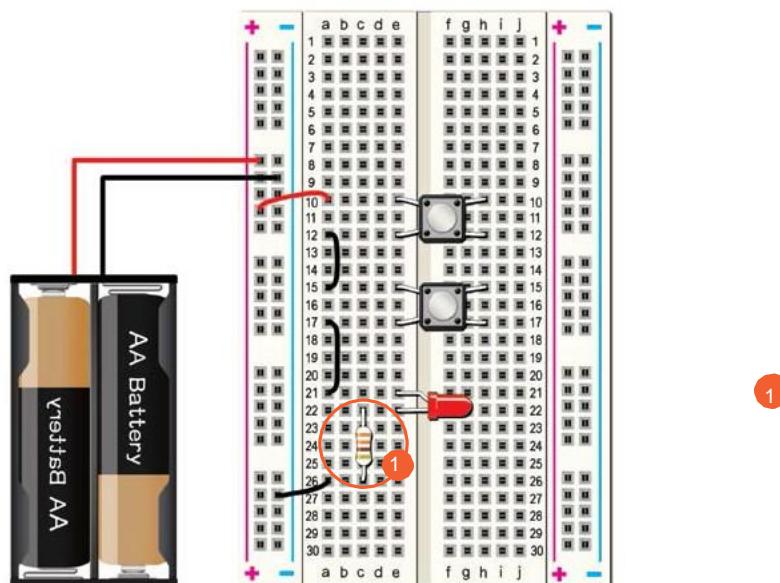
Usar dos interruptores para encender el led (circuito en serie)

Objetivos del proyecto

Encender el led al conectar el circuito y accionar dos interruptores.

Expresión del hardware

Es similar al proyecto anterior, pero esta vez usaremos dos interruptores. Los dos interruptores están conectados en serie. Por lo tanto, debes presionar ambos interruptores para encender el led.



Práctica

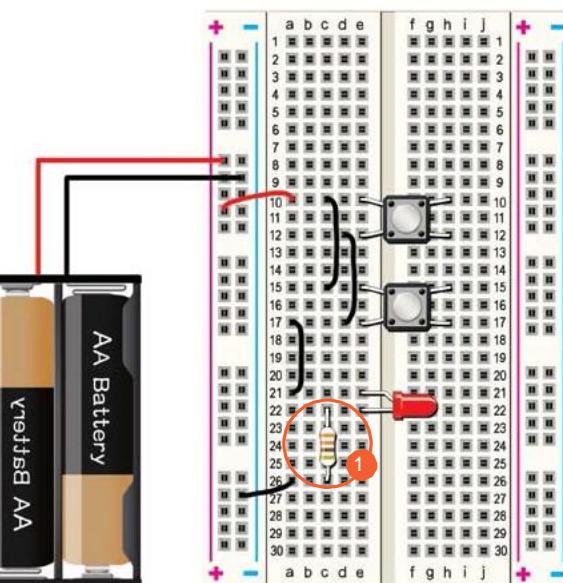
Usar dos interruptores para encender el led (circuito paralelo)

Objetivos del proyecto

Encender el led al conectar el circuito y accionar dos interruptores.

Expresión del hardware

Usaremos dos interruptores otra vez, pero esta vez estarán conectados en paralelo. Por lo tanto, el led comenzará a brillar al encender uno de los interruptores.



REVISIÓN → **Repaso**

Si no entiendes qué son los conductos en serie y en paralelo, piensa en calles simples y bifurcaciones. El conducto en serie es como una calle que termina si el comienzo o el final de la calle se bloquea. Sin embargo, en las bifurcaciones puedes pasar incluso si un lado está bloqueado siempre que haya otra parte habilitada.

¿Terminaste los cuatro proyectos exitosamente? ¿Qué piensas de los proyectos con ledes? Los proyectos en los que trabajamos no requirieron programación. Aprendiste que el led empieza a brillar cuando la corriente eléctrica fluye mediante el suministro de energía externa a la placa de creación, al cable, al interruptor y luego hasta el led incluso sin programación. Aquí se aplica un principio similar al que aprendimos en la primera semana en la que encendimos el led al conectar la batería.

A partir de la tercera semana, trabajaremos en un proyecto que requiere programación. También examinaremos por qué es necesario programar.

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.

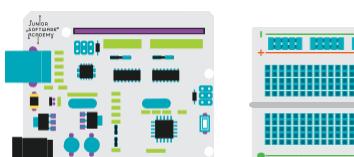
SEMANA
03

A brillar **Led parpadeante**

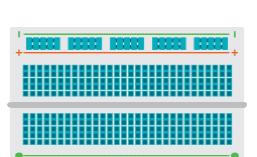
Objetivos de aprendizaje

- Entender qué es la placa de creación para crear un programa que haga que un led parpadee y un programa que haga que otros dos ledes emitan un parpadeo alternándose.

Materiales para preparar



Placa de creación



Placa de pruebas



Cable USB



3 ledes



2 resistencias de $330\ \Omega$



Cable



LECCIÓN

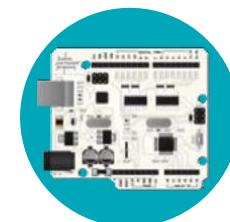
Recuerda lo que aprendimos durante la primera semana al encender el led con la batería de mercurio y trabaja en un verdadero proyecto de led. ¿Qué tendrías que hacer para crear un pequeño semáforo con ledes rojo, amarillo y verde? Estamos intentando crear un semáforo en el que el led rojo se encienda durante 5 segundos y se apague; en el que led amarillo brille por dos segundos y se apague; y, por último, en el que el led verde brille durante cinco segundos y se apague. Después de esta ronda, en orden inverso, el led amarillo brilla durante dos segundos y se apaga, seguido del led rojo, que brilla durante cinco segundos y se apaga. Pensemos en formas de crear un semáforo que repita constantemente el orden secuencial del rojo (5 segundos), el amarillo (2 segundos) y del verde (5 segundos), y luego el amarillo (2 segundos) y el rojo (5 segundos).

Para crear un semáforo que se active en este orden y en esta secuencia temporal, necesitamos una configuración que lo active automáticamente. Entonces, ¿qué debes hacer para que el hardware se active automáticamente? Así es. Necesitas un programa que le dé órdenes a una computadora. El programa puede controlar las computadoras (PC). Entonces, ¿cómo unes los ledes rojo, amarillo y verde en la computadora?

De esto se encargará la placa de creación, sobre la cual aprenderemos brevemente. Puedes considerar la placa de creación como una computadora de mano. Actúa como computadora de etapa media o puente intermedio para conectar los componentes como el led a la computadora para ponerlos en uso.

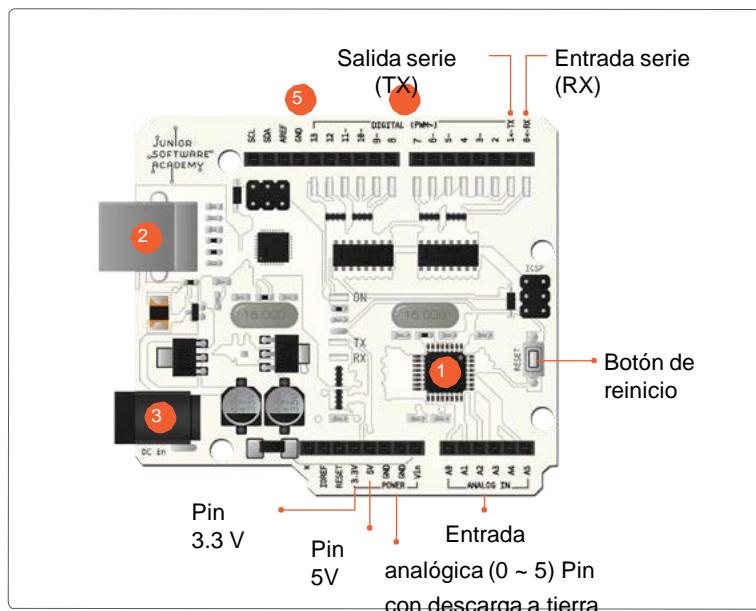
La placa de creación contiene la unidad de control (CPU) que puede procesar el control y los comandos de cómputo, y tiene muchas partes (también llamadas conectores) en las que puedes conectar los pines de los componentes. Puedes conectar los componentes como el led, el submotor y el sensor de medición de temperatura a la computadora mediante la placa de creación.

Estudiaremos la plataforma Arduino que está formada por la placa de creación (hardware) y el programa (software). La placa de creación es una configuración física para conectar la computadora y los componentes del hardware, y los programas activan los componentes unidos a la placa de creación. Puedes transferir los contenidos que programaste desde la ventana de la PC a la placa.



Arduino, el prototipo de placa de creación, se inventó para artistas, diseñadores e ingenieros aficionados que estaban interesados en los objetos animados. Por lo tanto, la ventaja de usar Arduino es que las personas pueden crear muchas cosas sin tener un conocimiento profundo sobre el hardware. ¿Analizamos la plataforma Arduino en más detalle?

Placa de creación



Observaremos las partes que cumplen una función principal en la placa de creación con mayor detalle.

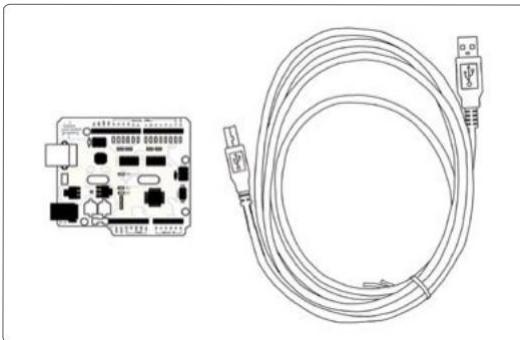
① El cerebro de la placa de creación

El microcontrolador es como el CPU de una computadora. Cumple la función de nuestro cerebro. Guarda y ejecuta programas, ya que tiene una pequeña área de almacenamiento.

② Pin USB

El pin USB conecta la placa de creación con la computadora. También suministra energía además de conectarlos.

Entre los enchufes que están en ambos lados del cable USB, conecta el enchufe plano (USB A, del lado derecho) en el conector USB de la computadora, y el enchufe redondo (USB B, del lado izquierdo) en el conector USB de la placa de creación.



③ Conector externo de energía

Como un conector que se conecta a una fuente de energía externa (batería), la energía se suministra cuando conectas la batería aquí. Si conectas el conector USB, subes el programa a la placa de creación y suministras la energía al conectar el adaptador de la batería con el conector de energía externo, la placa de creación se activaría incluso si desenchufas el conector USB.

④ Pin de entrada/salida digital

El pin de entrada/salida digital permite la entrada/salida de la señal digital. Cada pin tiene un número entre 1 y 13, y los que tienen el símbolo “~” también permiten el paso de la señal de entrada/salida analógica. En nuestras placas de creación, cada pin tiene un led incorporado (este será nuestro próximo ejercicio). La entrada/salida digital es positiva (+).

⑤ Pin digital con descarga a tierra

GND es negativo y significa descarga a tierra. Para la corriente, necesitas terminales positivos (+) y negativos (-). Este pin es el terminal negativo de la placa de creación.

Los temas que no se han cubierto se explicarán gradualmente a medida que trabajemos en nuestro proyecto. Ahora, mientras examinamos la placa detenidamente, ¿por qué no repasamos los programas?



¿El cable USB utilizado tiene algo en particular?

El cable USB en la imagen se ve distinto del cable USB que usamos normalmente, ¿verdad? Los cables USB se presentan en una variedad de tipos y los nombres cambian según los enchufes del cable unidos a cada extremo. Para ser más exactos, el cable USB utilizado aquí se denomina USB A-B porque es una convergencia del enchufe USB A que conectas a la computadora y del USB B que conectas a Arduino.

Programa Arduino

Práctica

Instalar el software Arduino

El programa Arduino se usa para los comandos de programación que la placa de creación ejecutará, y esta programación se llevará a cabo en tu computadora. Cuando transfieres tus programas a la placa de creación, puedes ejecutar el hardware que está conectado a la placa.

El proceso en el que transfieres el programa a la placa de creación se realiza de la siguiente manera:



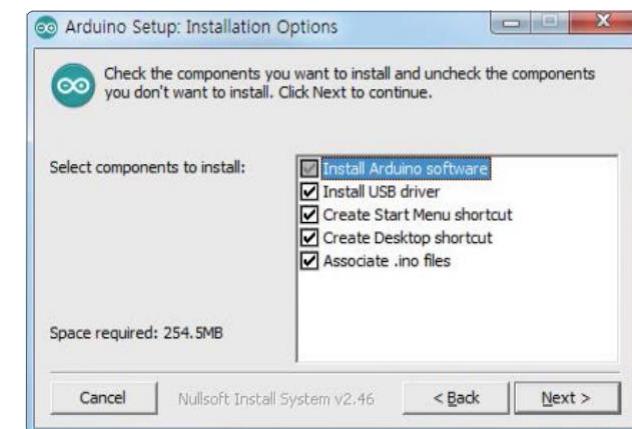
En la plataforma Arduino, el proceso de programar se denomina **“sketching”** y significa que puedes escribir el programa con facilidad como si hicieras dibujos. Para escribir los programas, nuestra placa de creación usa el mismo programa que Arduino para ejecutar los programas. Además, el acto de codificar el programa para crear la placa de creación (o el microcontrolador de la placa de creación, para ser más exactos) se denomina **“compilar”**. El acto de transferir los códigos compilados se denomina **“subir”**. El software de Arduino se encarga de todo, desde iniciar el entorno del programa y escribir el programa, hasta transferir el código compilado a la placa de creación. Es muy fácil usar el software de Arduino, por lo que no te llevará mucho aprender a usar el programa.

Primero, instalaremos el software Arduino. Un punto a tener en cuenta es que cuando instalamos el software, la placa de creación y la computadora se deben desconectar del cable USB.

1 Descarga el programa correspondiente a tu SO en <http://arduino.cc/en/main/software>.

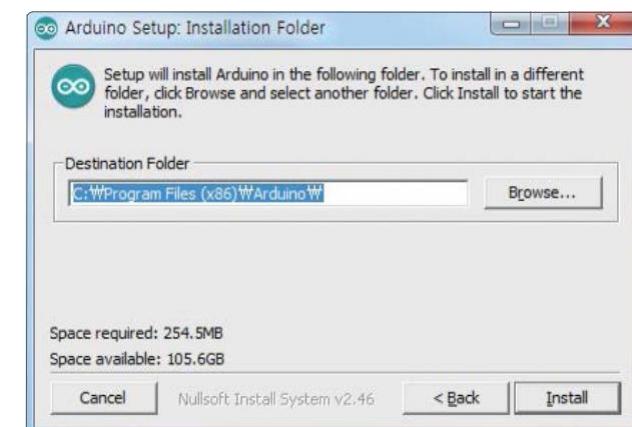
2 Cuando implementas el archivo, aparece una ventana. Presiona “Aceptar” para continuar.

3 No cambies la configuración predeterminada inicial y presiona “Siguiente”.



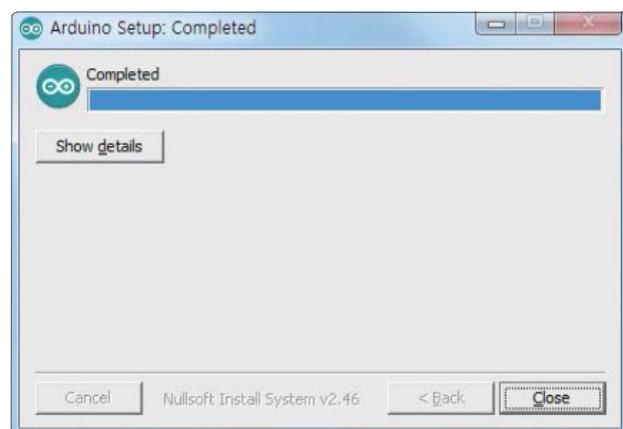
4

Presiona “Navegar” y selecciona una carpeta para guardar el archivo. El archivo más comúnmente designado es C:/ Archivos del programa/Carpeta Arduino.

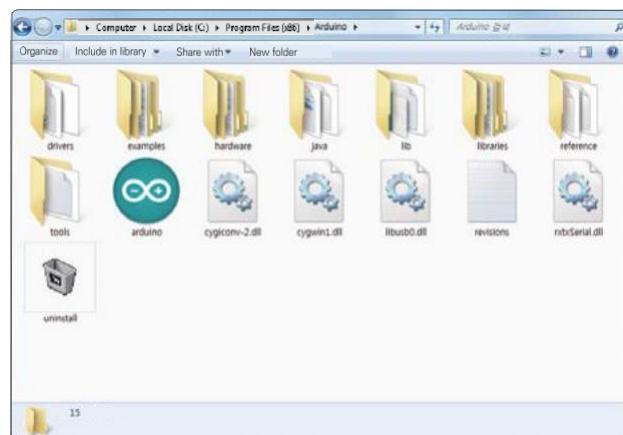


5 Presiona “Instalar” en la pantalla anterior. Aparecerá una ventana de instalación.

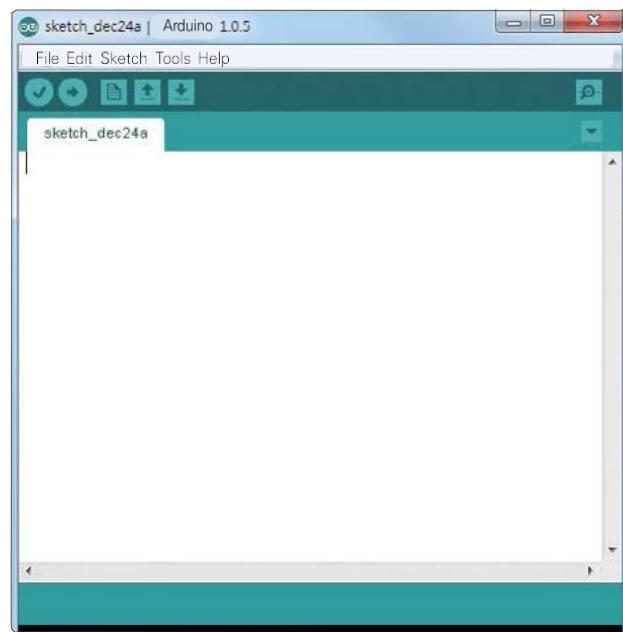
- 6**
Presiona “Cerrar” cuando finalice el proceso de instalación.



- 7**
Muchos archivos se formarán bajo el archivo designado.

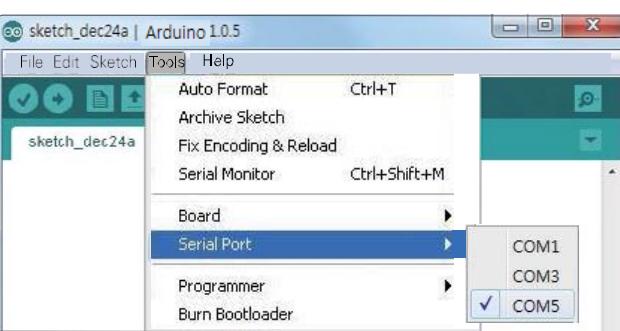
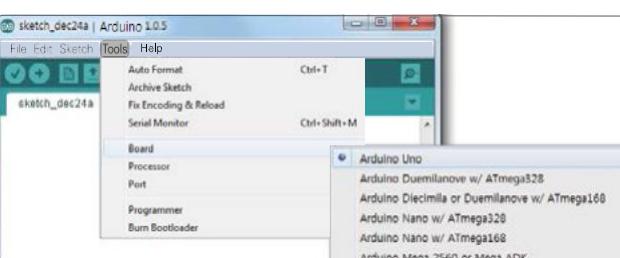
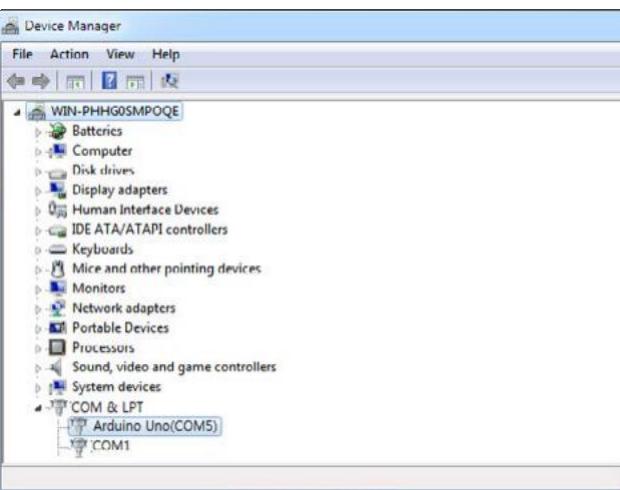


- 8**
Se habrá completado la instalación del software y el controlador. Ahora, ejecuta *arduino.exe*.



- 9**
Conecta la computadora y la placa de creación con el cable USB. Antes de escribir el programa, debes designar un puerto. Ingresa a “Inicio”, luego a “Panel de control”, “Sistemas y seguridad” y “Gestión del equipo”. Abre el menú “Puerto” y verifica qué le sigue al paréntesis de Arduino Uno (COM1 o COM3). Fíjate que esté marcado como “Arduino Uno” y no como “Placa de creación”.

- 10** Luego, designa el nombre del modelo de la placa. Selecciona el modelo utilizado en “Barra de herramientas” y luego **11** Regresa al software Arduino Uno y designa un puerto con el mismo nombre que el que encontraste en “Gestión del equipo” de “Barra de herramientas” y “Puerto serie”. Si no designas un puerto, no podrás transferir el *sketch* a la placa.



Después de la instalación, debes designar un modelo. No podrás subir el programa a menos que los dos estén correctamente designados. Solo necesitas instalar el programa una vez, pero debes designar el puerto y el modelo cada vez que ejecutes el programa. Ahora estás listo para leer el programa.



¿Puerto? ¿Modelo? ¿Por qué son importantes?

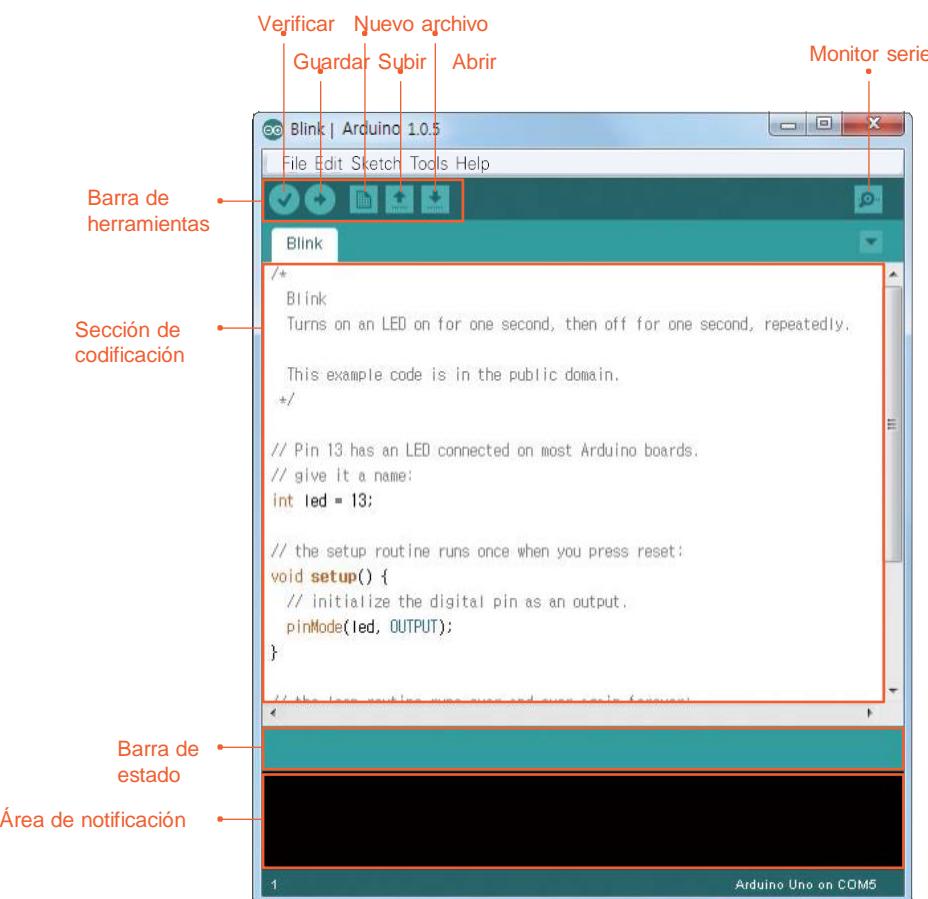
En resumen, el puerto y modelo son lugares y métodos de comunicación. Indican en dónde encontrarse y cómo comunicarse entre sí.

Práctica

Examinar la pantalla del software

Entorno de desarrollo integrado (IDE) del software Arduino

Ahora, analicemos el menú y las barras de herramientas. Son las mismas que ves en distintos software, como Hangle, Word y Excel. En el área grande del medio, puedes escribir tu código (se denomina *sketch* en el lenguaje Arduino, ¿recuerdas?). La barra de estado muestra el proceso o el resultado de la compilación, y el área de notificación notifica los datos o el mensaje enviado desde el software hasta la placa de creación.



Repasemos las herramientas que se usan más frecuentemente. Los cinco botones de la izquierda son los que más se usan. Encontrarás otro botón con otra función a la derecha.



1 Verificar

Verifica tu *sketch* para detectar problemas, y si no hay errores, convierte el *sketch* en un lenguaje que la máquina pueda entender mediante el proceso denominado compilación.

2 Subir

Sube el código convertido (sin errores) después de presionar el botón “Verificar” a la placa Arduino.

3 Nuevo archivo

Es un botón que presionas cuando creas un nuevo archivo de programación. Se abre un nuevo archivo.

4 Abrir

Se usa para abrir el *sketch* que escribiste anteriormente.

5 Guardar

Se usa para guardar el *sketch* que estás escribiendo.

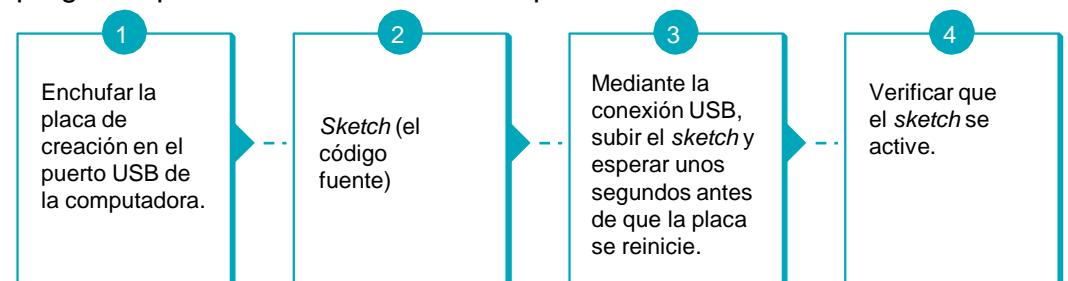
6 Monitor serie

Se usa para verificar los valores que envías en las series de un lado al otro.

Práctica

Ejercicio con ejemplos de Arduino

El software de Arduino tiene programas básicos con los que los nuevos usuarios pueden jugar. Como dice el dicho, ver las cosas por uno mismo vale tanto como todos los libros y viajes. Aprendamos cómo escribir un programa para Arduino al observar la placa de creación.



3 De la forma en que se abre el archivo de ejemplo *Blink*, aparece el *sketch* de ejemplo en la sección de codificación. Este *sketch* de ejemplo tiene el código para que el led conectado al pin 13 en la placa de creación emita una luz intermitente.

```

Blink | Arduino 1.0.5
File Edit Sketch Tools Help
Sketchbook
Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.

Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output:
  pinMode(led, OUTPUT);
}

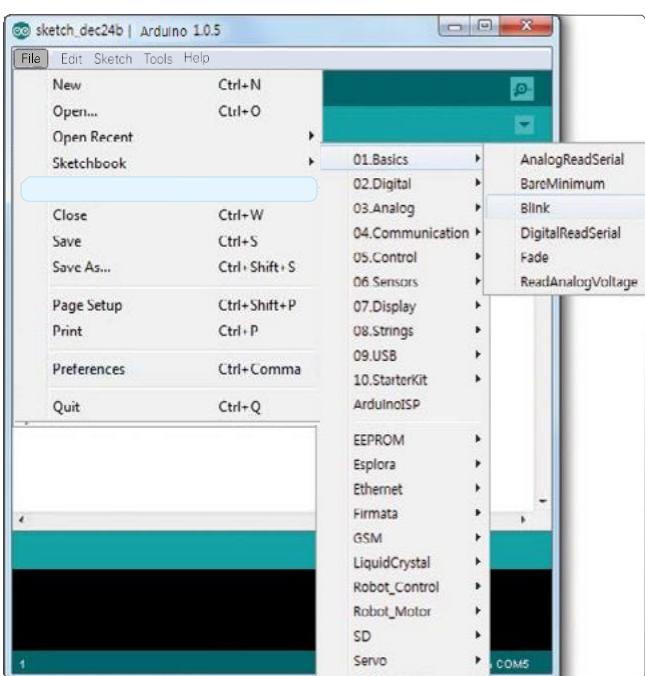
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // Turn the LED on (HIGH is the voltage level)
  delay(1000);               // Wait for a second
  digitalWrite(led, LOW);     // Turn the LED off by making the voltage LOW
  delay(1000);               // Wait for a second
}

```

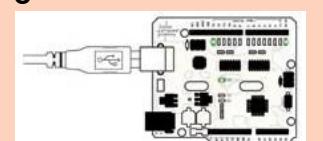
Ejemplos de ejecución

1 Conecta la placa de creación y la computadora con un cable USB.

2 Prepara tu *sketch*. Usa el *sketch* de ejemplo en el software Arduino. “Archivo”, “Ejemplo”, “01 Basics” y “Blink”



¿Ya necesitas un led?



Todos los pines digitales en la placa de creación que usamos tienen un led incorporado, por lo que no necesitas conectar un led aparte.

4 Empieza a compilar cuando presionas el botón “Verificar”.

compilation complete

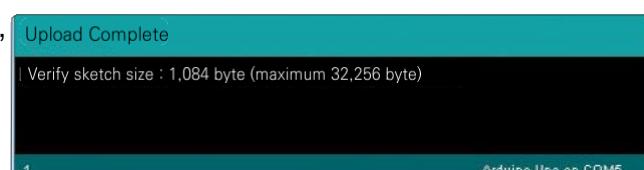
Verify sketch size : 1,084 byte (maximum 32,256 byte)

5 Cuando termina de verificar, muestra “Compilación completa” en la barra de estado. Encontrarás el resultado en el área de notificación y podrás determinar que el tamaño del código de ejemplo es de 1084 bytes.

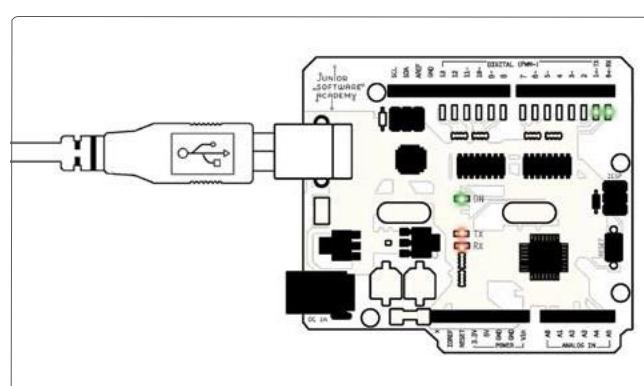
6 Ahora, envía el código compilado a la placa de creación. Presiona el botón “Subir”.



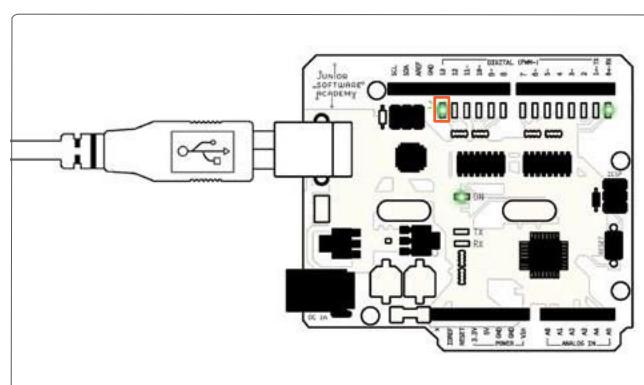
7 Si se subió correctamente, aparecerá un mensaje que muestra “Subida completa” en la barra de estado.



8 Cuando termines de subir, verás que las luces de transmisión (TX) y recepción (RX) parpadean de forma intermitente. Esto significa que los datos se están enviando desde la computadora hasta la placa de creación y viceversa constantemente.



9 Esto muestra el *sketch* que se ejecuta después de la subida. El led insertado en la superficie de la placa de creación parpadea una vez por segundo.



Cuando la placa de creación empieza a ejecutarse. ¿cómo la detienes?

Para detener la activación de la placa de creación, debes desconectar la energía (el USB que conecta la computadora y la placa de creación). No existe un botón de encendido/apagado, por lo que debes desconectar la energía para que la placa de creación deje de ejecutarse.

Examinar el código de ejemplo

Analicemos el código del ejemplo. Estudia el código y recuerda los principios que se mencionan a continuación.

o código se ejecuta de
a hacia abajo.

2

El código en la función `setup()` se ejecuta una sola vez cuando la placa de creación comienza a activarse. Aquí realizarás todas las preparaciones para activar el programa completo.

3

El código en la función `loop()` se ejecuta de forma repetida siempre que la placa de creación tenga energía. Aquí escribirás el código del programa.

	0	/*	
1	1		
2	0		
3	3	*	
4	0	*/	
5	6	//	
6	6	int led = 13;	Designa un nombre.
7	5	//	
8	3	void setup() {	
9	9	pinMode(led, OUTPUT);	Configura el pin digital correspondiente como salida.
10	1	}	
11	0	//	
12	1	void loop() {	
13	1	digitalWrite(led, HIGH);	Enciende el led (HIGH se refiere al nivel de voltaje). Espera un segundo.
14	4	delay(1000);	
15	1	digitalWrite(led, LOW);	Apaga el led (asegúrate de que el voltaje sea bajo). Espera un segundo.
16	5	delay(1000);	
17	1	}	
18	6	*/	

Es probable que no estés seguro de qué buscar, ya que no conoces el formato del código. Examinemos cada parte detenidamente.

1 Nota al pie

La primera parte que observaremos es la **nota al pie**. La primera parte no tiene nada que ver con la ejecución del programa. En el código, la primera parte contiene un espacio para las frases que los codificadores (personas) dejan para que sea más fácil de leer. Cuando la nota al pie tiene más de dos líneas, escribes la nota entre “`/*`” y “`*/`”. Además, cuando la nota

al pie está en una sola fila, solo debes escribir “//” al principio. La computadora lee el programa salteándose las filas con estos símbolos. Debemos interpretar la intención del programador cuando aparecen estos símbolos. Al leer la nota al pie, sabrás que este programa es para encender y apagar el led una vez por segundo.

② int led = 13;

Como explicamos antes, este *sketch* de ejemplo se puede aplicar en una placa Arduino Uno estándar. Es por eso que hay una nota al pie que expresa “la mayoría de las placas Arduino tienen el pin 13 conectado al led incorporado”.

③ Variables y configuración predeterminada

El código *sketch* está formado por la parte que designa las variables que se usarán en un programa, *void setup()* y *void loop()*. Los explicaremos en orden: primero, nombra el número trece como “led”. Luego, ejecuta *void setup()*. Esta sección es la configuración del programa o configuración predeterminada. Esta parte solo se ejecuta una vez. Usa una función llamada *pinMode* aquí y configura el número del pin digital 13 como salida. *PinMode* es la función que se usa para designar el pin correspondiente como entrada o salida. Usa *pinMode(led, INPUT)*; cuando uses el pin como entrada.

El punto y coma (;) unido a cada código establece que un comando se ha completado. Esto es parte de la sintaxis de la computadora, por lo que debes escribirlo así. El programa no se podrá compilar si no se coloca el punto y coma en el lugar correspondiente.

④ Ejecución repetitiva

Void loop() se ejecuta de forma repetitiva. *DigitalWrite* es una función que se usa para encender o apagar un pin. *HIGH* y *LOW* representan el nivel de voltaje; *HIGH* significa encender y *LOW*, apagar. *Delay* es una función que detiene el programa temporalmente. Puedes configurar la hora al escribir un número entre paréntesis después de la función. La unidad de tiempo entre paréntesis está expresada en milisegundos (1/1000 segundos). Cuando escribes “1000”, el programa se detiene por un segundo.



¡Aplausos! ¡Lo estás haciendo muy bien! Te has capacitado para usar la placa de creación. Los contenidos que siguen son proyectos con los que puedes trabajar fácilmente al aplicar lo que has aprendido hasta ahora. Para prepararnos para la próxima clase, hemos creado un ejercicio que te permite usar una placa de pruebas real y un led. Desafíate a ti mismo si deseas probarlo antes de la próxima clase.

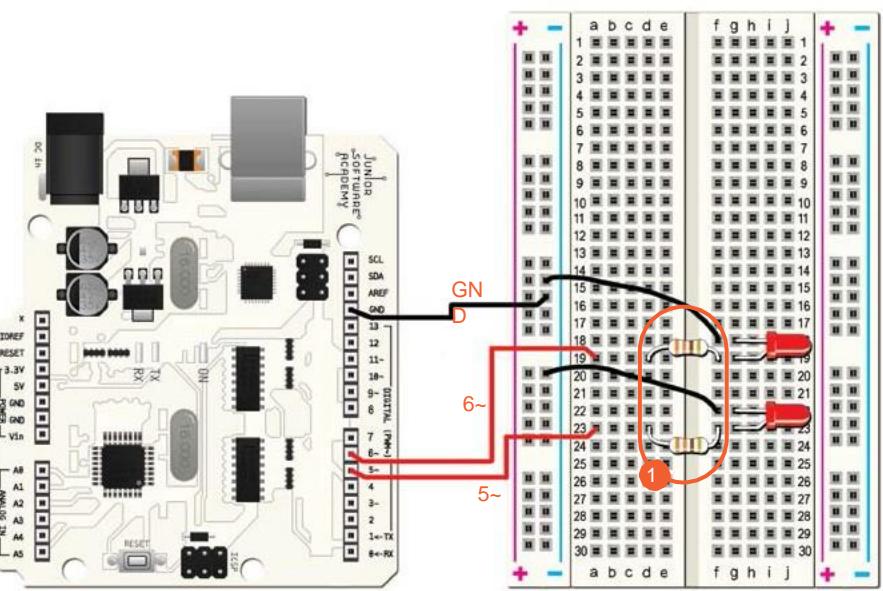
Práctica

Hacer que dos ledes emitan un parpadeo intermitente

Objetivos del proyecto

Escribir un programa en el que dos ledes emitan un parpadeo intermitente durante un segundo de forma alternada.

Expresión del hardware



① Resistencia de 330 Ω



Pensar en código

- 1 Presiona “Archivo nuevo” en la barra de herramientas y crea un *sketch* nuevo. Cuando presionas “Archivo nuevo”, aparece un espacio en blanco para el código.
- 2 Ingresá el mismo *sketch* que ves a continuación. Analicemos el código y la nota al pie.

```

01 int led1 = 5;
02 int led2 = 6;
03 void setup() {
04     pinMode(led1, OUTPUT
05 ); pinMode(led2, OUT
06 PUT);
07 }
08 void loop() {
09     digitalWrite(led1, HIGH);
10     delay(1000); digitalWrite
11 (led1, LOW); delay(1000);
12     digitalWrite(led2, HIGH);
13     delay(1000); digitalWrite
14 (led2, LOW); delay(1000);
15 }
16 }
```

Conecta el led1 con el pin digital 5.

Conecta el led2 con el pin digital 6.

Enciende el led1.
Espera un segundo.
Apaga el led1.
Espera un segundo.
Enciende el led2.
Espera un segundo.
Apaga el led2.
Espera un segundo.

REVISIÓN

Ejecutamos y examinamos el ejemplo del led que parpadea intermitentemente. Si alguna vez has visto un lenguaje de programación que las computadoras usan para comunicarse, probablemente viste varias palabras o contenidos que no conoces. Sin embargo, te sentiste muy bien al ejecutar un programa, ¿verdad? Tendrás la posibilidad de aprender sobre los términos y las funciones del programa de forma gradual. Será mejor para ti si creas el programa por tu cuenta. A partir de la próxima clase, crearás proyectos divertidos con la placa de creación uno por uno.

Reaso

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

- 3 Compila el programa y presiona el botón “Verificar”.
- 4 Envía el código compilado a la placa Arduino al presionar el botón “Subir”.

Verás que ambos ledes emiten un parpadeo alternado.

El led 5 continúa parpadeando. ¿Qué debo hacer?

Cuando presionas el botón “Subir”, se borra el *sketch* anterior en la placa Arduino y aparece uno nuevo. Si el programa nuevo no se sube correctamente, presiona el botón “Reiniciar” e intenta volver a subirlo.

Si desenchufaste y volviste a enchufar el cable mientras escribías los dos programas, debes volver a configurar el puerto. Aprendiste cómo configurar el puerto, ¿verdad?

Piensa en la parte que más te costó y compártela con tus compañeros.



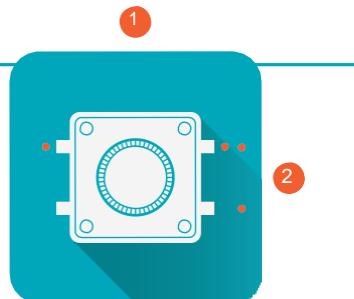
SEMANA
04

Cómo controlar el led

Práctica

Encender el led con un interruptor

¿Recuerdas el proyecto en el que trabajamos durante la semana 3? Era un programa en el que el led de la placa de creación emitía un parpadeo intermitente por segundo mediante un programa simple. Esta vez, quisimos crear un programa que conecte el led, la resistencia y el interruptor a la placa de pruebas para que el led se encienda solo cuando presionamos el interruptor.



- ③ Dos pines están interconectados.
- ④ Dos pines no están interconectados.

Cuando presionas el botón de la parte inferior, los dos puntos de los circuitos se conectan y la corriente empieza a fluir. El componente que permite que la corriente fluya cuando el circuito está conectado y que impide que fluya cuando está desconectado se denomina “interruptor”. El tipo exacto del interruptor que utilizaremos es interruptor pulsador. La corriente fluye solo cuando presionas el interruptor. Cuando observas la estructura interna, se conectan pares de pines, como puedes ver en la imagen. Las cuatro patas se conectan cuando presionas el botón. Por lo tanto, el circuito desconectado se conecta cuando presionas el botón

y así es como empieza a fluir la corriente.



Circuitos cerrados y abiertos

Circuito cerrado se refiere a un circuito en el que la corriente puede fluir porque todo está conectado. Por otro lado, en un circuito abierto la corriente no puede fluir porque algunas partes están desconectadas. Los circuitos cerrados también se denominan “circuitos ininterrumpidos”, y los abiertos, “circuitos interrumpidos”. Aunque “cerrado” tiene una connotación más negativa que “abierto”, recuerda que en un circuito cerrado la corriente fluye.

Objetivos del proyecto

Crear un circuito que encienda el led cuando presionas el interruptor.

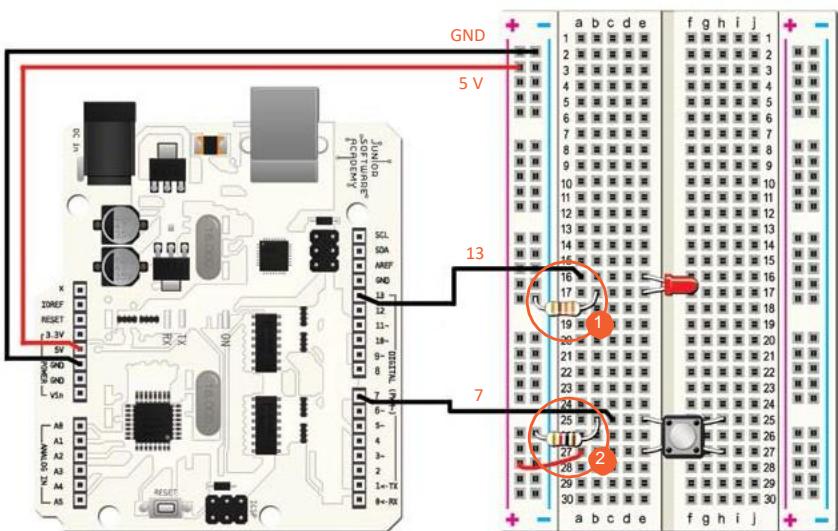
Expresión del hardware

Observa el diagrama del hardware e intenta comprender cómo funciona.

- 1 Un lado del interruptor está conectado al pin digital 7 y a la resistencia (con descarga a tierra). La corriente no fluye hasta el pin 7 porque el circuito interior no está conectado.

2 Cuando presionas el interruptor, la potencia de 5 V se suministra al pin 7, por lo que se vuelve *HIGH* o adquiere un estado 1.

3 Cuando el pin 7 adquiere un estado 1, cambia el voltaje de salida del pin 13 conectado al led a *HIGH* para encender el led.



1 Resistencia de $330\ \Omega$



2 Resistencia de $1\ K\Omega$

Pensar en código

```

01 int led = 13; -----o
02 ; int swPin -----o
03 = 7; int swV -----o
04 al = 0; -----o
05 void setup() { pinMode(-----o
06   led, OUTPUT); pinMod -----o
07   e(swPin, INPUT); -----o
08 }
09
10 void loop() {
11   swVal = digitalRead(swPi -----o
12   n); if(swVal == HIGH) { -----o
13     digitalWrite(led, HIGH); -----o
14   } else { -----o
15     digitalWrite(led, LOW); -----o
16   }
17 }
```

Define el pin digital utilizado por el led 13.

Define el pin digital conectado al interruptor.

Aplica el valor inicial de 0 a la variable para guardar el valor del interruptor.

Configura el pin del led como salida.

Configura el interruptor como entrada.

Lee el valor de entrada del interruptor y lo guarda en la variable *swVal*.

Cuando el valor de entrada es *HIGH* (cuando se presiona el interruptor), enciende el led.

Cuando el valor de entrada es *LOW* (cuando el interruptor no se presiona), apaga el led.

1 int swVal = 0;

Es una variable de tipo *int* que verifica la variable para guardar el valor, sin importar si el interruptor está encendido o apagado. A diferencia de las invariables, las variables se refieren a los números que pueden cambiar después de que se inicia el programa. Aquí, declaramos el valor inicial de la variable *swVal* como 0. Cuando presionas el interruptor, la variable se convierte en 1. Por el contrario, cuando alejas tu mano del interruptor, el valor de la variable regresa a 0 otra vez.

La palabra *int* delante de la variable declara que es una variable que guarda un número entero. No tienes que saber esto en detalle por ahora, pero intenta comprender que debes declarar la forma después de considerar el tamaño del valor, ya sea que el valor es un número decimal o solo un número entero. La forma *int* puede guardar cualquier valor entre -32.768 y 32.767. Podría ser molesto, pero el motivo por el que categorizamos los valores es que el espacio de almacenamiento de la placa de creación no es suficiente, por lo que debemos usarlo de manera eficiente.

2 Enunciado *if*

Este enunciado verifica si el valor de *swVal* que muestra si el interruptor está presionado o no es *HIGH* o *LOW*. Cuando cumple la condición, enciende o apaga el led.

¿Qué es un enunciado *if*?

El “enunciado *if*” es una sintaxis que brinda distintos resultados según la condición. Se suele denominar “enunciado condicional”. Está formado por enunciados condicionales y oraciones, y cuando la condición del enunciado condicional se cumple, se ejecuta la oración.

Ejemplo	Explicación
<pre>if(expresión condicional) {oración 1; } else { oración 2; }</pre>	Si la expresión condicional es verdadera (si la condición se cumple), la oración 1 se ejecuta. Sin embargo, si la expresión condicional es falsa, la oración se saltea y continúa con el siguiente código. El siguiente código podría ser otra expresión condicional o podría ser un enunciado <i>else</i> . Cuando la condición no se cumple, se ejecuta la oración 2 debajo de <i>else</i> .
<pre>if(swVal == HIGH) { digitalWrite(LED, HIGH); } else { digitalWrite(LED, LOW); }</pre>	Si el valor de entrada es <i>HIGH</i> (cuando se presiona el interruptor), enciende el led. Si el valor de entrada es <i>LOW</i> (cuando el interruptor no se presiona), apaga el led.

a == b	Ejecuta si <i>a</i> es igual que <i>b</i>	a != b	Ejecuta si <i>a</i> no es igual que <i>b</i>
a < b	Ejecuta si <i>a</i> es menor que <i>b</i>	a > b	Ejecuta si <i>a</i> es mayor que <i>b</i>
a <= b	Ejecuta si <i>a</i> es menor o igual que <i>b</i>	a >= b	Ejecuta si <i>a</i> es mayor o igual que <i>b</i>



Práctica

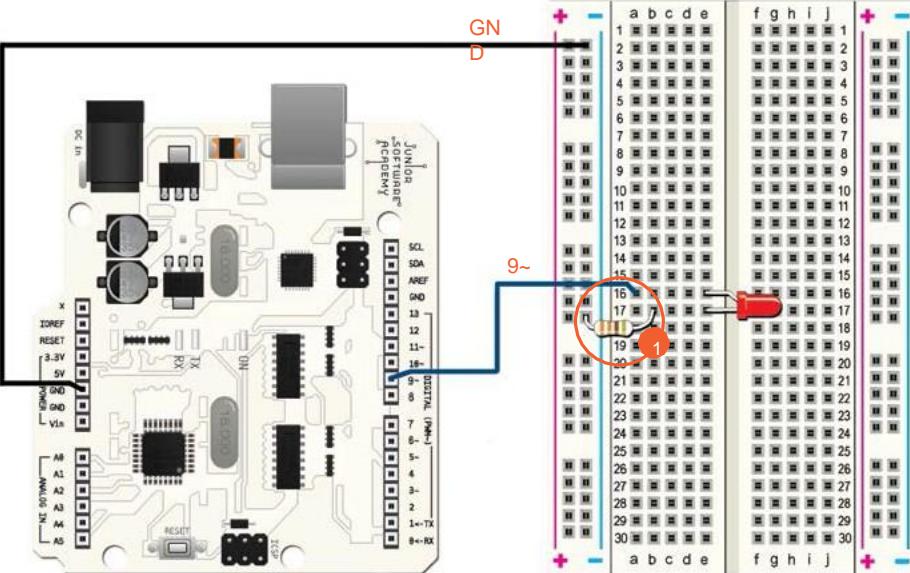
Encender el led progresivamente

Objetivos del proyecto

Lograr que el led se encienda con una luz suave.

Expresión del hardware

La composición es muy sencilla, pero debes asegurarte de conectar ambos terminales con el pin 9.



1 Resistencia de 330 Ω



Pensar en código

```

01 int ledPin = 9;           Define el pin digital utilizado por el led 9
02
03 void setup() {           Aquí no debes hacer nada.
04
05 }
06
07 void loop() {
08     for(int fadeValue = 0; fadeValue <= 255; fa
09         deValue += 5) { analogWrite(ledPin, fad
10             eValue);
11             delay(30);
12     }
13
14     for(int fadeValue = 255; fadeValue >= 0; fa
15         deValue -= 5) { analogWrite(ledPin, fad
16             eValue);
17             delay(30);
18 }

```

¿Qué es el enunciado **for**?

El enunciado **for** es un tipo de bucle formado por una expresión inicial, una expresión condicional y expresiones de incremento o decremento.

Forma

for (expresión inicial 1; expresión condicional 2; expresión de incremento o decremento 3)
4 oraciones
}

El orden de ejecución del enunciado **for** es el siguiente. Primero, importa el resultado después de ejecutar la expresión inicial 1, y si es verdadera cuando se la compara con la expresión condicional 2, ejecuta la oración 4 y aumenta o disminuye según las expresiones de incremento/decremento 3. Luego, verifica la expresión condicional 2 otra vez y ejecuta las 4 oraciones otra vez si es verdadera. Este proceso se repite hasta que la expresión condicional 2 sea falsa (la condición no se cumple).

En otras palabras, el proceso se repite en este orden (1243,243,243,243) hasta que la oración 2 sea falsa.



Usaré el programa que tiene tres ledes y haré que parpadeen en un intervalo de un segundo como ejemplo.

Código que no usa el enunciado **for**

```

int ledPin1 = 5; int
ledPin2 = 6; int
ledPin3 = 7;

void setup() {
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(ledPin3, OUTPUT);
}

void loop() {
    digitalWrite(ledPin1, HIGH);
    delay(1000); digitalWrite(ledPin1,
    LOW); delay(1000);
    digitalWrite(ledPin2, HIGH);
    delay(1000);
    digitalWrite(ledPin3, HIGH);
    delay(1000);
    digitalWrite(ledPin3, LOW);
    delay(1000);
}

```

Código que usa el enunciado **for**

```

int del=100;

void setup() {
    for(int i = 5; i <= 7; i++) {
        pinMode(i, OUTPUT);
    }
}

void loop() {
    for(int i = 5; i <= 7; i++) {
        digitalWrite(i, HIGH);
        delay(del);
        digitalWrite(i, LOW);
        delay(del);
    }
}

```



Salida digital y salida analógica

Hasta ahora, solo hemos utilizado ejecuciones para encender o apagar. Son ejecuciones digitales, en las que solo hay dos condiciones, *HIGH(1)* o *LOW(0)*. Probablemente hayas escuchado hablar de que en el mundo de las computadoras solo existen “0” y “1”. La función que usamos aquí es *digitalWrite*. Por el contrario, la función analógica permite un cambio gradual hacia el nivel más alto, como el control de volumen. El proyecto usó la función *analogWrite*, que permite una salida analógica, para modificar el brillo en etapas en vez de encender y apagar el led. El número “0”, como valor mínimo, representa el estado apagado, y el “255”, como valor máximo, representa el estado con más brillo. Por ahora, solo debes recordar que el brillo del led cambia levemente en proporción a los valores que están en el medio. Volveremos a analizar este material en la semana 11. Usarás la función *digitalWrite* para la salida de los valores de la señal digital. También debes tener en cuenta que se utiliza la función *analogWrite* en el número del pin de salida digital (~3,~5,~6, ~9, ~10, ~11) para la salida de valores de señales analógicas.

La próxima vez, crearás tu propio proyecto. Piensa en un objeto (como una muñeca, alcancía, almohada, etc.) al que le quieras agregar luces led y llévalo a la próxima clase.

REVISIÓN

Reaso

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.



SEMANA
05



Proyecto

DISEÑO CREATIVO 01

Hablemos con luces



Objetivos de aprendizaje

Crear mi propio proyecto según los conceptos aprendidos hasta la semana 4.



Materiales para preparar

Kit de creación, varios objetos (objetos viejos como muñecas, alcancías, almohadas, vasos de papel o pelotas de ping-pong), herramientas para hacer manualidades y cinta adhesiva.

Lámpara con envases de yogur

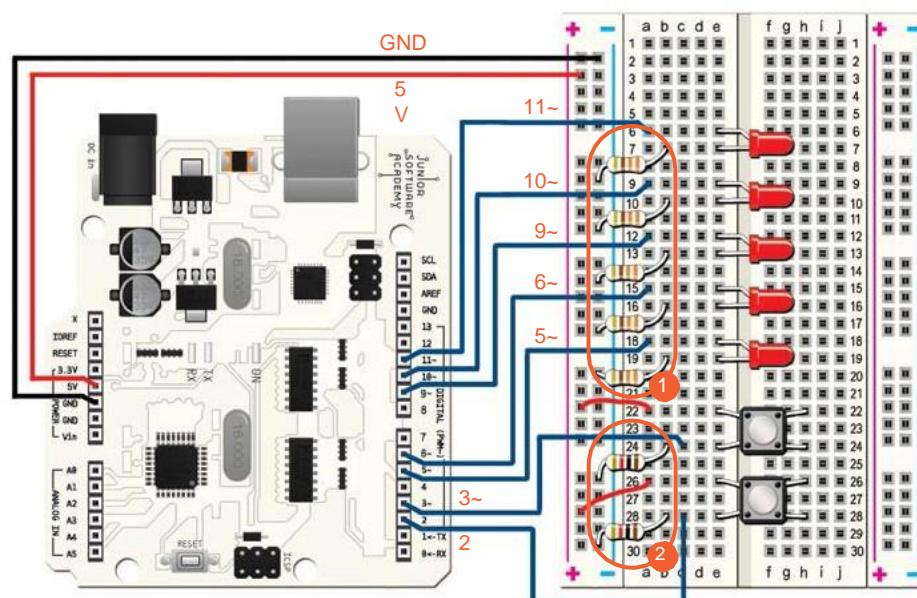
Objetivos del proyecto

Crearemos una lámpara al colocar el led dentro de botellas de yogur.

- 1 Prepara varias botellas de yogur vacías.
- 2 Puedes pintarlas, decorarlas o colocar papeles de origami dentro de las botellas vacías.
- 3 Conecta el cable a 3 o 5 ledes (puedes conectar más ledes si así lo deseas).
- 4 Agujerea el fondo de las botellas y coloca el cable conectado al led dentro de las botellas y fíjalo.
- 5 Coloca los vasos de papel hacia abajo y fija un interruptor en el fondo. Crea un agujero en un lado del vaso de papel, pasa el cable por ahí y conéctalo con los pines del interruptor. El interruptor se utiliza para encender y apagar los ledes.
- 6 Conecta los cables con los ledes dentro de las botellas de yogur. Mientras los conectas con los dos cables de los soportes de baterías, conecta un lado con el fondo del vaso de papel. Cuando conectes el cable, fíjalo con la cinta adhesiva. Debes tener en cuenta que los dos terminales del led tienen que estar conectados al terminal polar correcto de las baterías.
- 7 Cuando termines de conectar, enciende el interruptor para verificar si el led se enciende.

Expresión del hardware

Esta es una composición de hardware y una fuente que puedes usar como referencia cuando creas algo que no solo se enciende y se apaga, sino que puede realizar varios movimientos. Cuando uses la creatividad para tus proyectos, podrás crear algo increíble



1 Resistencia de 330 Ω



2 Resistencia de 1 KΩ



Arduino Uno

Este circuito es un ejemplo de cómo puedes controlar más de un LED con un solo microcontrolador. Los tres primeros LEDs están controlados por interruptores que se activan individualmente. Los otros dos están controlados por interruptores que se activan juntos.

Los tres primeros LEDs están conectados en serie entre la salida digital 5 y el suelo. Los demás están conectados en serie entre la salida digital 10 y el suelo. Los interruptores están conectados entre la salida digital 11 y el suelo, y entre la salida digital 10 y el suelo respectivamente.

Los tres primeros LEDs están controlados por interruptores que se activan individualmente. Los demás están controlados por interruptores que se activan juntos.

Los tres primeros LEDs están controlados por interruptores que se activan individualmente. Los demás están controlados por interruptores que se activan juntos.

Los tres primeros LEDs están controlados por interruptores que se activan individualmente. Los demás están controlados por interruptores que se activan juntos.

Los tres primeros LEDs están controlados por interruptores que se activan individualmente. Los demás están controlados por interruptores que se activan juntos.

Pensar en código

```

01 const int led1 = 5;
02 const int led2 = 6;
03 const int led3 = 9;
04 const int led4 = 10;
05 const int led5 = 11;
06
07 const int swPin1 = 2;
08 const int swPin2 = 3;
09
10 int swValue1 = 0;
11 int swValue2 = 0;
12
13 int count = 0;
14
15 void setup() {
16   pinMode(swPin1, INPUT);
17   pinMode(swPin2, INPUT);
18
19   pinMode(led1, OUTPUT);
20   pinMode(led2, OUTPUT);
21   pinMode(led3, OUTPUT);
22   pinMode(led4, OUTPUT);
23   pinMode(led5, OUTPUT);
24 }
25
26 void loop() {
27   swValue1 = digitalRead(swPin1);
28   swValue2 = digitalRead(swPin2);
29
30   if(swValue1 == HIGH) {
31     count = 1;
32   }
33   if(swValue2 == HIGH) {
34     count = 2;
35   }
36   if(swValue1 == HIGH && swValue2 == HIGH) {
37     count = 3;
38   }
39 }
```

Define el pin digital utilizado por el primer led en 5.
Define el pin digital utilizado por el segundo led en 6.
Define el pin digital utilizado por el tercer led en 9.
Define el pin digital utilizado por el cuarto led en 10.
Define el pin digital utilizado por el quinto led en 11.
Define el pin digital utilizado por el primer interruptor en 2. Define el pin digital utilizado por el segundo interruptor en 3.
Variable que indica el estado de encendido/apagado de *swPin1*. Variable que indica el estado de encendido/apagado de *btPin2*.
Variable que indica qué número de botón le corresponde a cada botón.
Configura *swPin1* en modo de entrada.
Configura *swPin2* en modo de entrada.
Configura led1 en modo de salida.
Configura led2 en modo de salida.
Configura led3 en modo de salida.
Configura led4 en modo de salida.
Configura led5 en modo de salida.
Lee el estado de encendido/apagado del valor de *swPin1* y lo guarda en *swValue1*.
Lee el estado de encendido/apagado del valor de *swPin2* y lo guarda en *swValue2*.
Si *swValue1* está encendido (si *swPin1* está presionado), guarda 1 como variable *count*.
Si *swValue2* está encendido (si *swPin2* está presionado), guarda 2 como variable *count*.
Si *swValue1* y *swValue2* están encendidas (si se presionan los dos interruptores), guarda 3 como variable *count*.

```

40  if(count == 1) { ..... Cuando la variable count es 1.
41      digitalWrite(led1, HIGH);
42      digitalWrite(led2, HIGH);
43      digitalWrite(led3, HIGH);
44      digitalWrite(led4, HIGH);
45      digitalWrite(led5, HIGH);
46      delay(300);
47      digitalWrite(led1, LOW);
48      digitalWrite(led2, LOW);
49      digitalWrite(led3, LOW);
50      digitalWrite(led4, LOW);
51      digitalWrite(led5, LOW);
52      delay(300);
53  }
54  else if(count == 2) { ..... Cuando la variable count es 2.
55      digitalWrite(led1,HIGH);
56      delay(300);
57      digitalWrite(led1, LOW);
58      delay(300);
59      digitalWrite(led2,HIGH);
60      delay(300);
61      digitalWrite(led2, LOW);
62      delay(300);
63      digitalWrite(led3,HIGH);
64      delay(300);
65      digitalWrite(led3, LOW);
66      delay(300);
67      digitalWrite(led4,HIGH);
68      delay(300);
69      digitalWrite(led4, LOW);
70      delay(300);
71      digitalWrite(led5,HIGH);
72      delay(300);
73      digitalWrite(led5, LOW);
74      delay(300);
75  }
76  else if(count == 3) { ..... Cuando la variable count es 3.
77      for(int i = 0; i < 255; i += 5) {
78          analogWrite(led1, i);
79          delay(20);
80      }

```

```

81      for(int i = 0; i < 255; i += 5) {
82          analogWrite(led2, i);
83          delay(20);
84      }
85      for(int i = 0; i < 255; i += 5) {
86          analogWrite(led3, i);
87          delay(20);
88      }
89      for(int i = 0; i < 255; i += 5) {
90          analogWrite(led4, i);
91          delay(20);
92      }
93      for(int i = 0; i < 255; i += 5) {
94          analogWrite(led5, i);
95          delay(20);
96      }
97      for(int i = 255; i >= 0; i -= 5) {
98          analogWrite(led5, i);
99          delay(20);
100     }
101    for(int i = 255; i >= 0; i -= 5) {
102        analogWrite(led4, i);
103        delay(20);
104    }
105    for(int i = 255; i >= 0; i -= 5) {
106        analogWrite(led3, i);
107        delay(20);
108    }
109    for(int i = 255; i >= 0; i -= 5) {
110        analogWrite(led2, i);
111        delay(20);
112    }
113    for(int i = 255; i >= 0; i -= 5) {
114        analogWrite(led1, i);
115        delay(20);
116    }
117  }
118}
119

```

De esta manera, podrás controlar 5 ledes con dos interruptores en tres modos diferentes. Funciona con los siguientes tres modos: cuando presionas el interruptor 1; cuando presionas el interruptor 2; cuando presionas ambos interruptores, 1 y 2. ¿Crea este proyecto emocionante?

- . Cuando el valor de *count* es 1, todos los ledes se apagan juntos por un intervalo de 0,3 segundos. Aquí utilizamos la función *digitalWrite*.
- . Cuando el valor de *count* es 2, los ledes se encienden por 0,3 segundos y se apagan por 0,3 segundos en orden. Este proceso se repite desde el led 1 al led 5, uno después de otro.
- . Cuando el valor de *count* es 3, los ledes de 1 a 5 se encienden uno después del otro en un intervalo de 5,12(20*256 ms) segundos y se apagan en orden inverso en un intervalo de 5,12 segundos. Utilizamos la función *analogWrite* para encender y apagar los ledes. Esta función también puede expresar el brillo de la luz en 256 etapas. El nivel de brillo comienza en 0 cuando encendemos la luz y aumenta hasta llegar a 256. Funciona de manera inversa cuando los ledes se apagan.



El enunciado *if* en la línea 36 utiliza dos condiciones. De la misma manera, los enunciados *if* pueden distinguir diversas condiciones. Si colocamos **&&** (*and*) entre las condiciones, la máquina funciona cuando se cumplen ambas condiciones y si colocamos **||** (*or*), la máquina funciona cuando cualquiera de las dos condiciones se cumplen. Utilizamos **&&** (*and*) para hacer que los ledes funcionen al presionar ambos interruptores.

Este código funciona en orden consecutivo, por lo que cuando la tercera función (*count==3*) de ledes está ejecutándose, no cambia a otra función incluso cuando presionas el botón. Por lo tanto, para cambiar a otra función, deberás presionar el botón cuando el código llegue al final o deberás mantener el interruptor presionado hasta que se ejecute otra función. Debido a que la placa utiliza la función *delay*, este problema no se puede resolver con este código.

Proyecto CREATIVO

Hablemos con luces

¿Qué tal si creamos nuestro propio proyecto?

Podrás crear fabulosos trabajos al aplicar lo que aprendiste hasta la semana 4 y lo que aprendiste durante el proyecto de la lámpara de botellas de yogur.

1 Formar grupos

Siéntense en grupos de 4 a 6. Piensen en una actividad grupal en cinco minutos. Escriban algunos *sketches* mientras piensan en distintas ideas. Cada uno deberá exponer sus ideas y qué roles tendrán sus compañeros por uno o dos minutos. Los compañeros del grupo no deberán juzgar si el proyecto es bueno o malo. En cambio, deberán brindar críticas constructivas que puedan ayudar a desarrollar las ideas de cada uno.

2 Prepararse

Ahora, comienza a prepararte para tu proyecto. Prepara el kit de creación y otros materiales para manualidades que puedan servir para utilizar ledes. Trajiste los materiales a la clase, ¿verdad?

3 Resumir ideas

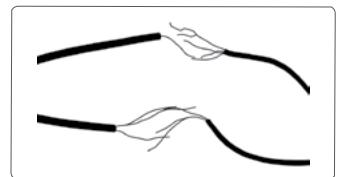
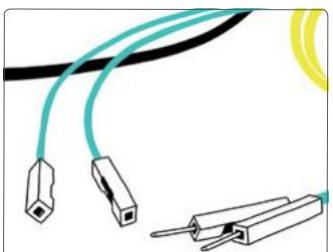
Crea un breve resumen de tus ideas con imágenes y textos. Luego, piensa en un *sketch*, la composición del hardware, un circuito y describe en detalle lo que harás y qué acciones realizarás.



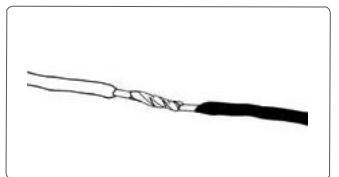
4 Conectar el circuito

Utilizando lo que aprendiste en las semanas 2 y 4, crea un proyecto que utilice un botón y un led. Podrías tener que unir dos cables para crear un cable más largo. El circuito será más resistente si sueldas los cables, pero esta vez trabajaremos sin soldaduras. Retuerce y fija dos cables entre los componentes para unirlos. También puedes fijarlos con una pistola de pegamento luego de unir los cables con los componentes.

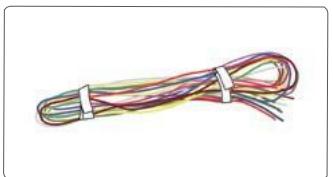
Algunos cables se parecen a los de las imágenes del kit. Si los cables son muy cortos, puedes conectarlos para crear un puente. Si el pin está flojo, asegúralo con cinta.



Retorcer los cables



Soldar



Manojo de cables

REVISIÓN**Reaso**

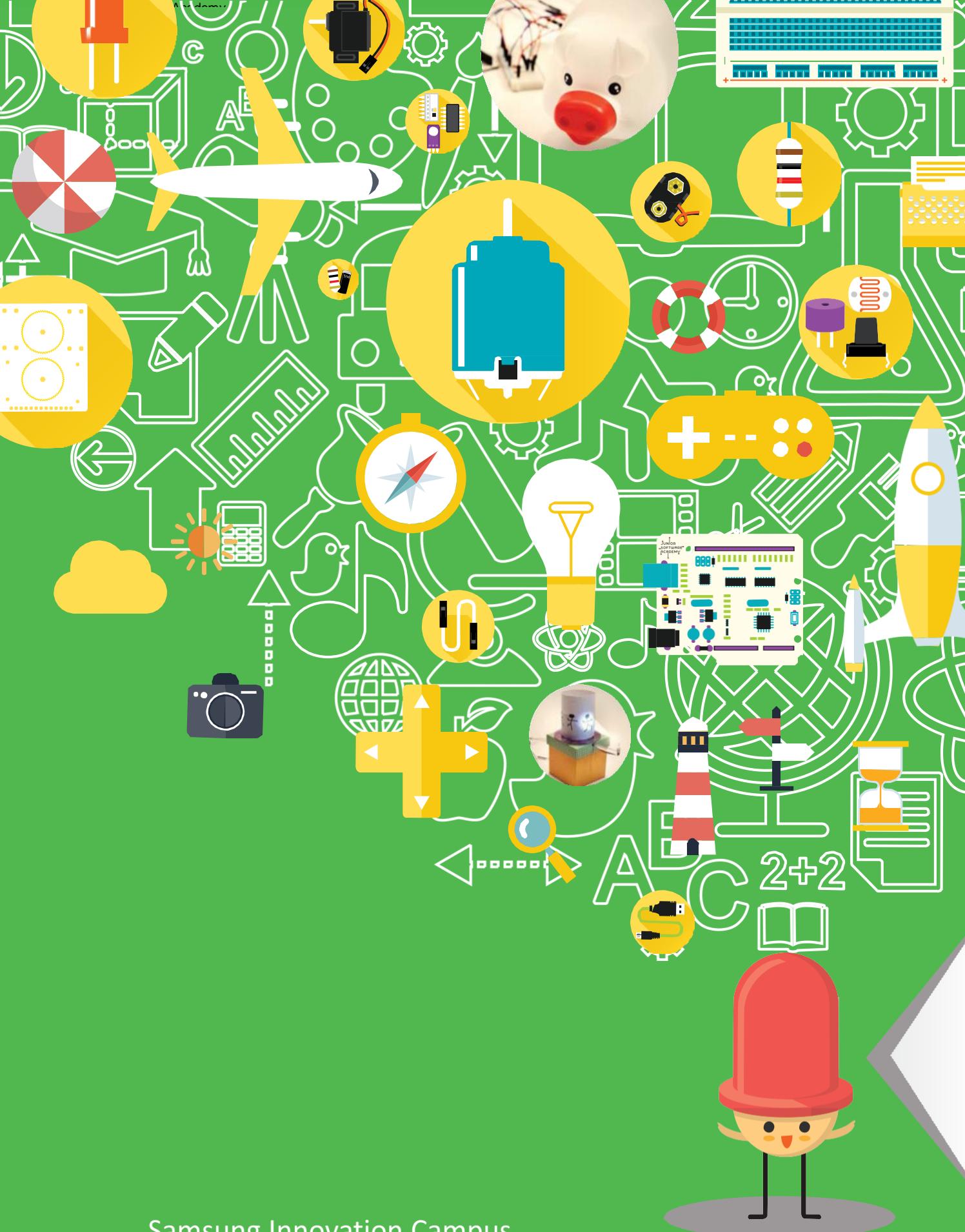
Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.

5 Presentar

Como parte final de esta clase, nombra tu proyecto y haz una presentación acerca de tu creación.

Durante la presentación, debes explicar el nombre del proyecto y describirlo (por qué se creó; cómo se creó y qué movimientos aplica), y luego escuchar las opiniones de tus compañeros sobre el proyecto.



CLASE DE CREACIÓN

02

- 06. Sensor 1
- 07. Sensor 2
- 08. Reproducir sonidos
- 09. Diseño creativo 02 (Hablemos con sonidos)

SEMANA
06

Sensor 1



LECCIÓN

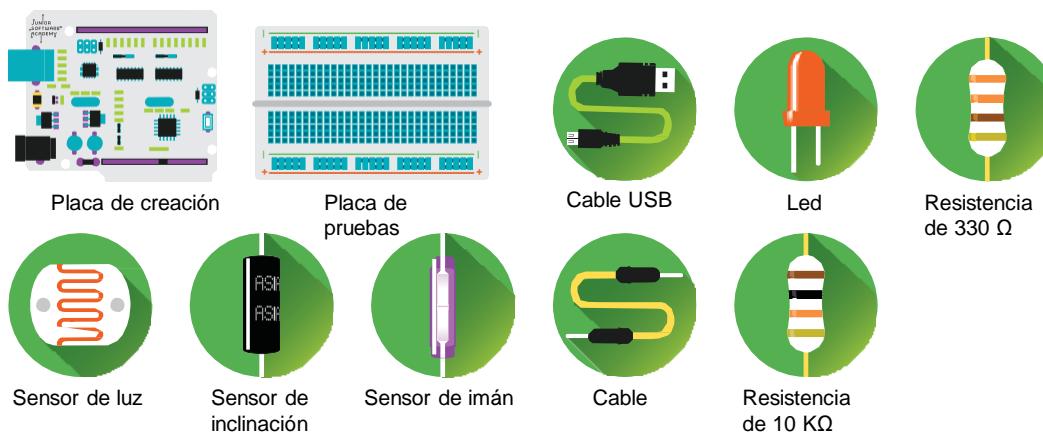
Aprendimos sobre varios proyectos que conectan la placa de creación y el led hasta la semana

5. Intentaremos utilizar algo llamado sensor a partir de esta clase. Adquirirás la inteligencia necesaria para distinguir qué son los objetos que usas en la vida diaria.

Objetivos de aprendizaje

- Conectar algunos tipos de sensores que usamos en nuestra vida diaria con la placa de creación y controlarlos.

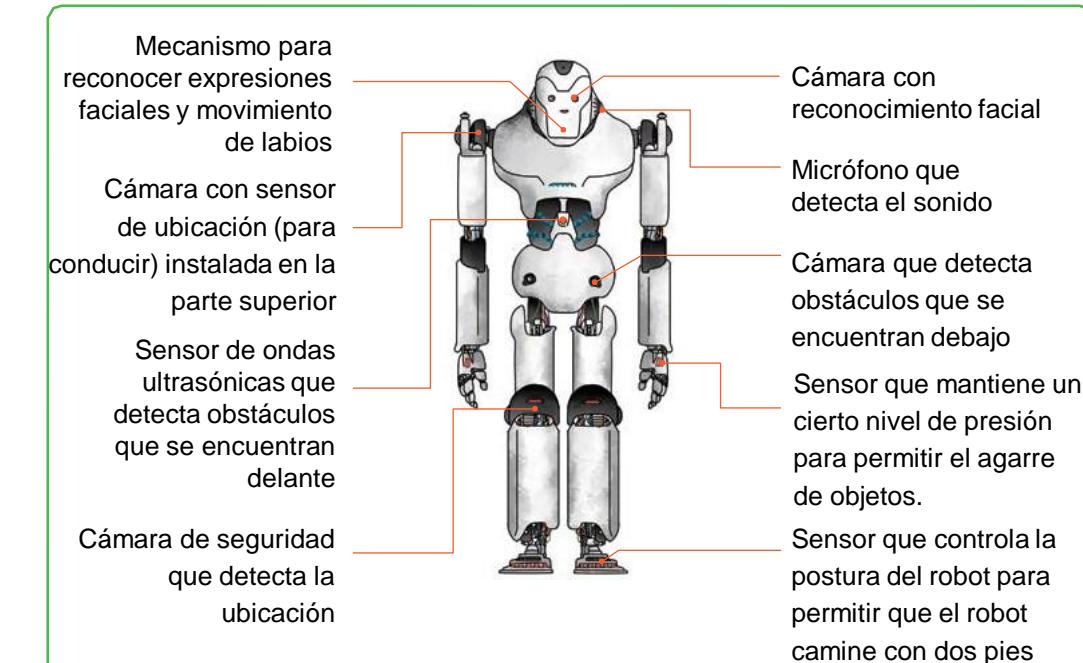
Materiales para preparar



¿Qué es un sensor?

¿Qué es un sensor? El sensor es un equipo que detecta lo que ocurre alrededor y lo transfiere en señales que las máquinas pueden leer. Puede medir la luz, el sonido, la temperatura, el contacto y la distancia. En otras palabras, al igual que nuestros cinco sentidos (vista, audición, olfato, gusto, tacto), la placa de creación percibe el ambiente que la rodea gracias al sensor.

Los sensores suelen utilizarse para crear robots. Ya existen robots que ven y distinguen las apariencias de las personas, que tienen capacidad auditiva para percibir sonidos y que tienen capacidad de discernir obstáculos.



Descubramos qué tipos de sensores están instalados a nuestro alrededor. Existen muchos sensores ocultos en teléfonos inteligentes que probablemente usas con frecuencia. También existen varios sensores dentro de una clase. ¿Lo has notado? Presentemos varios sensores que descubriste y escribe o haz un *sketch* en el espacio en blanco a continuación.



Los sensores que nos rodean:

1. Sensor de detección corporal: la luz se enciende automáticamente cuando las personas pasan por la puerta.
2. Sensor infrarrojo: la puerta se abre automáticamente o gira.
3. Control remoto infrarrojo: se activa si presionas el botón.
4. Sensor magnético: suena una alarma cuando se abren puertas cerradas.
5. Sensor de luz: se enciende el farol cuando es de noche y se apaga cuando es de día.

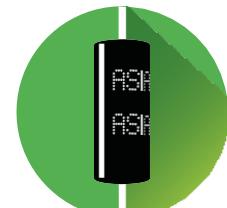
¿Descubriste cuántos sensores se usan? Por más que lo hubieras notado antes o no, verás que recibimos mucha ayuda de los sensores.

¿Solo existen dos tipos de sensores?

Los sensores se puede categorizar como sensores digitales y sensores analógicos. Aprendimos la diferencia entre digital y analógico cuando explicamos los métodos de salida. De la misma forma, los sensores también se pueden categorizar en sensores digitales que distinguen "O" o "X", y en analógicos que cambian a múltiples valores. Por lo tanto, puedes usar sensores digitales o analógicos según el objeto o el entorno que deseas percibir.

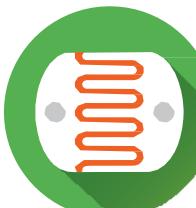
Observemos los sensores magnéticos y de inclinación, que son sensores digitales y el sensor de luz, que es analógico, dentro del kit.

- Sensor digital



Sensor de inclinación

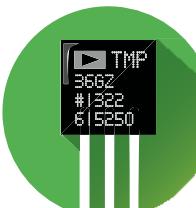
- Sensor analógico



Sensor magnético



Sensor de luz



Sensor de temperatura

Encontrar los sensores que nos rodean

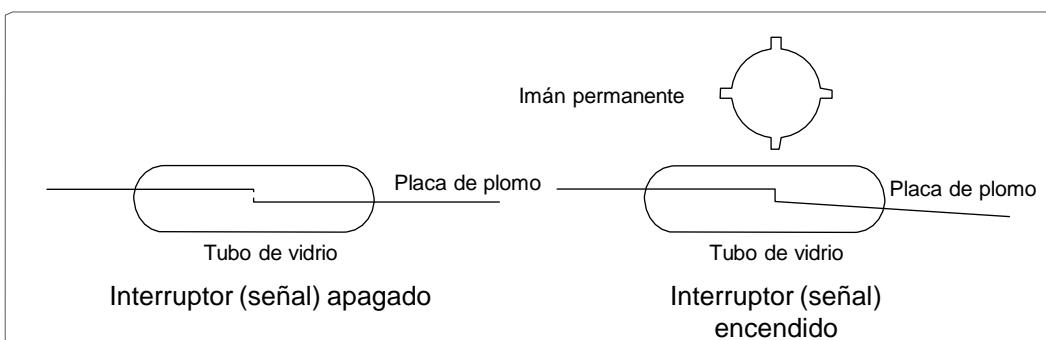
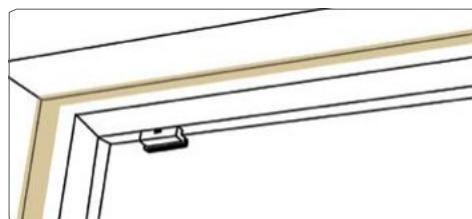
Descubramos los sensores que nos rodean y aprendamos sus funciones.

Sensor de inclinación

Este tipo de sensor se enciende y se apaga según la pendiente. Contiene una pequeña bolita dentro del sensor que gira según la pendiente y actúa como interruptor. También existe un sensor analógico que te permite verificar el ángulo de inclinación.

Sensor magnético

El sensor magnético, que se suele encontrar en las puertas que nos rodean, se usa con propósitos de seguridad.



Al igual que el diagrama que se encuentra en la parte superior, el sensor magnético tiene propósitos de seguridad, ya que si fijas el sensor magnético en la parte superior de una puerta, empieza a sonar una alarma cuando el imán se cae. Como puedes ver en el siguiente diagrama, cuando el imán está cerca de la placa de plomo, se fija, mientras que cuando se aleja, la alarma empieza a sonar, ya que la placa de plomo se cae. Para que sepas, al igual que el sensor de inclinación, el sensor magnético se puede controlar con un interruptor.

Sensor infrarrojo

El sensor infrarrojo se usa comúnmente en los controles remotos. Existe un led que envía el rayo infrarrojo invisible y otro que lo detecta. La parte que envía luz en los controles remotos se denomina “parte emisora” y la que recibe la luz en la televisión se denomina “parte receptora”. El led que se utiliza en la parte receptora funciona como sensor que detecta rayos infrarrojos.

Sensor de luz

La luz nos permite ver los objetos que nos rodean. Si puedes medir esa cantidad específica de luz, podremos enumerar el brillo y la oscuridad que nos rodean. Si aplicamos este principio, podremos crear un proyecto interesante que reaccione ante el brillo. También se puede aplicar a los faroles que se encienden si oscurece más que un nivel determinado.

Práctica

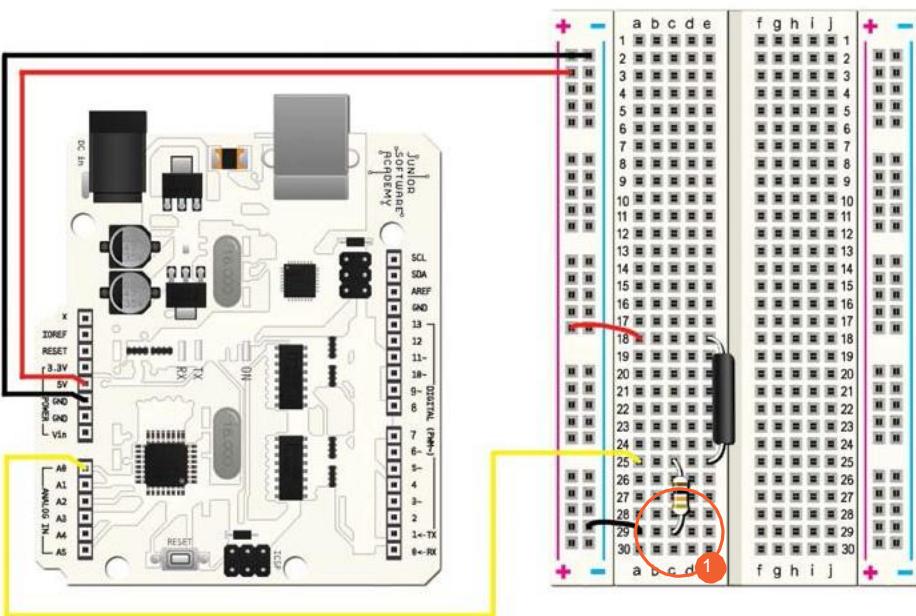
Hacer que la placa de creación baile con el sensor de inclinación

Empecemos con un proyecto que une un sensor a la placa de creación y lo activa. ¿Recuerdas cómo encendimos el led con un interruptor? Ahora, verifiquemos el funcionamiento del sensor al construir circuitos uno por uno. Puedes usar la placa de creación de muchas formas distintas si usas varios tipos de sensores.

Objetivos del proyecto

Al conectar el sensor de inclinación en la placa de pruebas y en la placa de creación, crearemos un circuito y conectaremos un sensor en él.

Expresión del hardware



1 Resistencia de 10 KΩ

Pensar en código

01	int sensorPin = 0;	Conecta el sensor al pin analógico 0.
02		
03	void setup() {	Fija la velocidad básica para las comunicaciones en serie.
04	Serial.begin(9600);	
05	}	
06		
07	void loop() {	Lee el valor analógico que proviene del sensor
08	int reading = analogRead(sensorPin);	Transmite el valor que lee la serie a la computadora
09	Serial.println(reading);	
10	delay(100);	Retrasa por 0,1 segundos y ejecuta la función <i>loop()</i>
11	}	

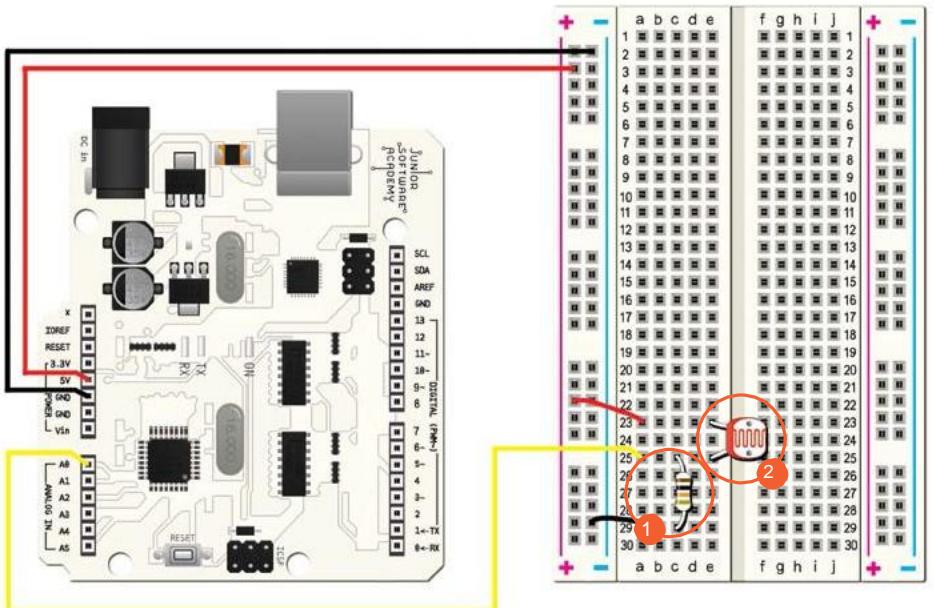
Resultados del proyecto

Después de cargar el *sketch*, podrás obtener información de inclinación que envía la serie a la placa de creación cuando presionas el botón de monitor serie en el lado derecho de la barra de herramientas.



Existen similitudes entre todos los sensores que utilizaremos. Todos tienen dos patas. Puedes activar muchos tipos de sensores al cambiar la posición de los sensores en el circuito, ya que todos tienen dos patas. Además, como los sensores con dos patas no tienen terminales positivos y negativos, podrás conectarlos y se activarán de todas formas. El sensor de inclinación y el sensor digital pueden percibir valores entre 0 y 1023, y el sensor de luz, que es un sensor analógico, cambiará los valores según el brillo. Los sensores digitales solo perciben el estado de encendido o apagado, al igual que los botones. Por el otro lado, los sensores analógicos pueden percibir varios valores según el cambio de luz.

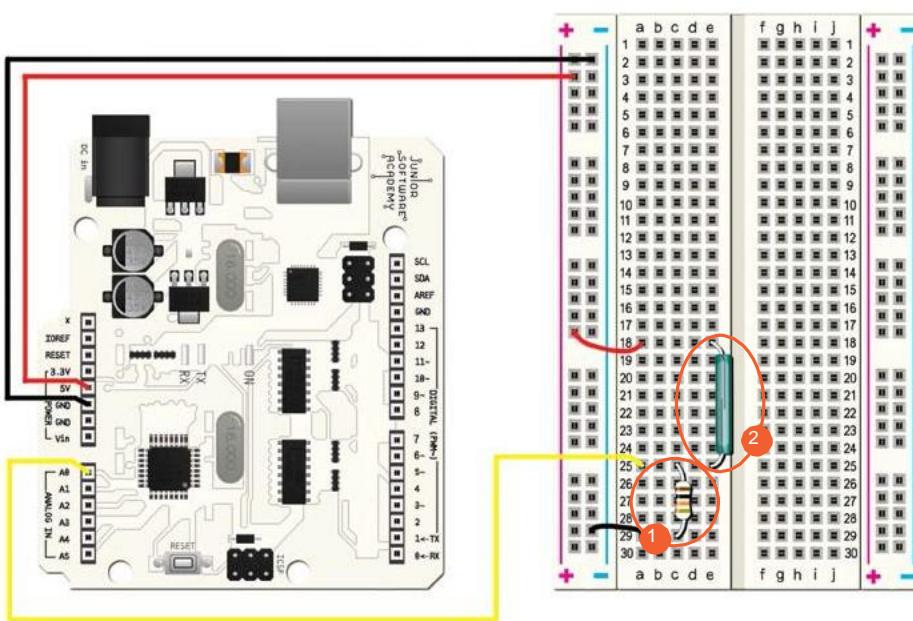
Ahora, conectemos el sensor magnético o el sensor de luz, en vez del sensor de inclinación y veamos



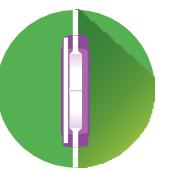
1 Resistencia de 10 KΩ



2 Sensor de luz (iluminación)



1 Resistencia de 10 KΩ



2 Sensor magnético (interruptor de plomo)

¿Funciona? Puedes hacer que funcione fácilmente si las conexiones son correctas. Ya probamos tres sensores que comparten una estructura similar. El resto de los sensores se activan cuando cambias el sensor y usas el mismo circuito y código. ¿No es divertido?

Como puedes ver, podrás crear varios tipos de equipos de entrada con el circuito y el código más básico. Si entiendes el principio de funcionamiento del sensor, podrás usar otros sensores además de los que vimos ahora.

Práctica

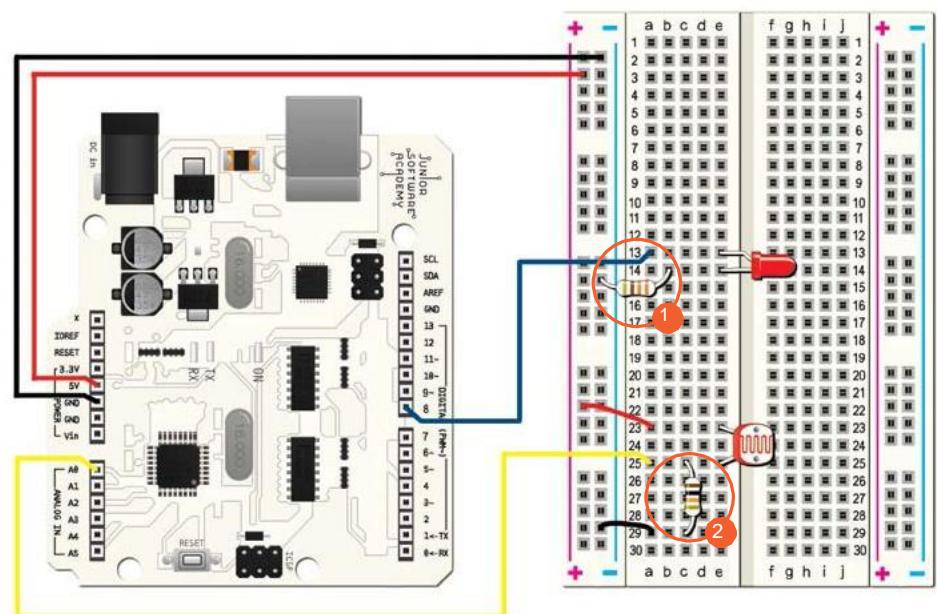
Crear un led que se ilumina cuando oscurece

El sensor de luz simplemente mide el brillo del entorno. Cuando se utiliza, puedes encender o apagar el led al hacer que perciba el brillo y la oscuridad en comparación con un valor fijo. ¿Por qué no creamos algo parecido a un farol que se encienda y apague automáticamente? Al usar la función de salida *PWM* de la placa de creación, hagamos que el led se encienda suavemente y gradualmente.

Objetivos del proyecto

Crearemos un led que se encienda automáticamente cuando el entorno oscurezca.

Expresión del hardware



1 Resistencia de 330 Ω



2 Resistencia de 10 KΩ

Pensar en código 1

```

01 int sensorPin = 0; -----o
02 int sensorVal = 0; -----o
03
04 void setup() { Serial.b-----o
05   egin(9600); -----o
06 }
07
08 void loop() {
09   sensorVal = analogRead(sensorPin)
10   ); Serial.println(sensorVal);
11
12   sensorVal = map(sensorVal, 0, 400, 0, 255)-----o
13   ; sensorVal = constrain(sensorVal, 0, 25
14   5);
15
16   analogWrite(8, 255 - sensorVal);
}

```

Conecta el sensor al pin analógico 0.
Inserta el valor inicial a 0 en la variable que guarda el valor del sensor.

Configura la velocidad básica para comunicaciones en serie.

Consulta las siguientes explicaciones.
Consulta las siguientes explicaciones.

Salidas en método *PWM*

La función Row 12: *map()* modifica automáticamente el valor en un rango particular en un valor en un rango determinado. En el código, 0 se fija en 0, y 400 se fija en 255. En otras palabras, los valores comprendidos entre el rango 0 a 400 se fijan en valores comprendidos en un rango de 0 a 255. El texto y el número entre paréntesis detrás de la función representan el “valor de la variable”, el “valor mínimo del sensor”, el “valor máximo del sensor”, el “valor mínimo a cambiar” y el “valor máximo a cambiar”.

La función Row 13: *constrain()* restringe los valores para que no sobrepasen cierto rango. Incluso si el valor de la variable es negativo (menor que 0) se designa como 0. El valor mayor que 255 queda restringido a 255. En otras palabras, los valores que percibe el sensor se fijan en los valores entre 0 y 255 cuando supera row13.

Pensar en código 2

```

01 int ledPin = 8; -----
02 int sensorPin = 0; -----
03 int sensorVal = 0; -----

04

05 void setup() {
06     pinMode(ledPin, OUTPUT);
07     Serial.begin(9600);
08 }

09

10 void loop() {
11     sensorVal = analogRead(sensorPin);
12     Serial.println(sensorVal);

13

14     sensorVal = map(sensorVal, 0, 400, 0, 255);
15     sensorVal = constrain(sensorVal, 0, 255);

16

17     if(sensorVal > 125) {
18         digitalWrite(ledPin, LOW);
19     }
20     else {
21         digitalWrite(ledPin, HIGH);
22     }
23 }
```

REVISIÓN

Reaso

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.



Puedes aplicar este proyecto a los siguientes proyectos.

- Forma el circuito con sensores de inclinación o magnéticos en vez del sensor de luz e intenta encender y apagar el led.
- Haz que el led ilumine el ambiente si percibe brillo y que se apague si está oscuro. Según lo que aprendiste en las semanas 3 a 5, crearemos variaciones y
- lo intentaremos.

SEMANA
07

Sensor 2

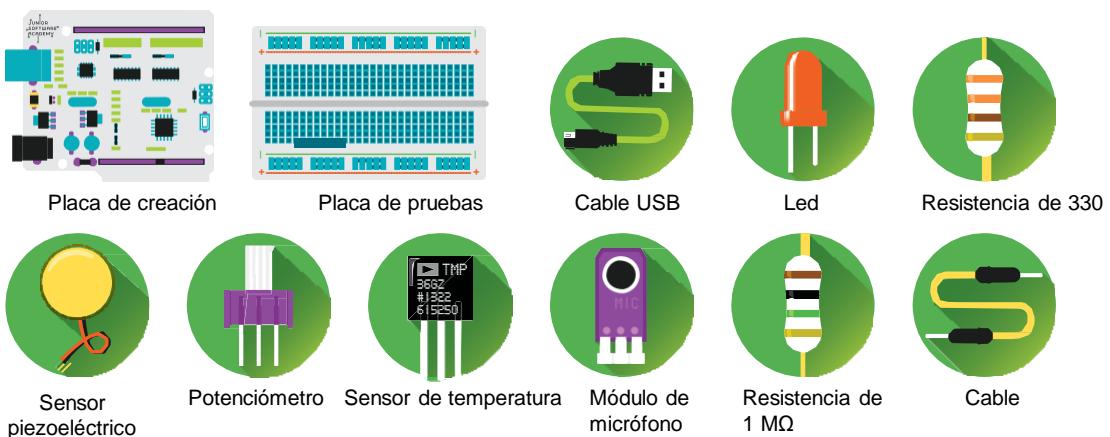

LECCIÓN

En esta clase, aprenderás sobre distintas maneras de utilizar sensores que no utilizamos en la clase anterior. Todos los sensores que utilizaremos son analógicos. Primero, veremos la función de cada sensor.

Objetivos de aprendizaje

- Conectar y controlar los sensores de uso diario con la placa de creación

Materiales para preparar



Sensor piezoelectrónico

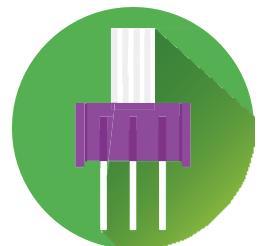
En la vida diaria, entramos en contacto con innumerables objetos. Escribimos en teclados y golpeamos puertas casi todos los días. En tales interacciones, se generan vibraciones. El sensor que detecta las vibraciones se conoce como sensor piezoelectrónico. Los sensores piezoelectrónicos también se conocen como detectores de vibraciones o sensores “piezo”.



Algo muy interesante es que el sensor piezoelectrónico detecta las vibraciones y luego emite sonidos. Es por esto que se trata de un componente central invaluable para los proyectos de electrónica relacionados con la música. El sensor está compuesto por dos superficies y se genera voltaje cuando el sensor detecta presión o torsión. Puedes determinar el grado de vibración al medir el voltaje generado por la entrada analógica de la placa Arduino. Por el contrario, cuando se envía voltaje a ambas partes, se produce una transformación proporcional al voltaje de ambos lados y se genera el sonido.

Potenciómetro

El valor de la resistencia se puede fijar y controlar al girar la perilla. Los potenciómetros generalmente se utilizan para controlar el volumen de máquinas como las guitarras y amplificadores de audio.



Sensor de temperatura

Cuando observas el pronóstico para el clima, recibes información sobre cuál será la temperatura para los días siguientes. La temperatura se puede medir fácilmente con un termómetro Celsius o con un termómetro digital. Sin embargo, mediremos la temperatura exterior utilizando un sensor de temperatura analógico, que podrás conectar a la placa de creación y utilizar para verificar la temperatura a través del monitor serie. Aprenderás cómo Arduino procesa los datos analógicos cambiantes y por qué se requiere de un proceso de conversión para mostrar la temperatura.

Micrófono

El micrófono es el componente que transforma los sonidos del ambiente en señales eléctricas. No solo puede transmitir sonidos, también puede medir el volumen del sonido.

Práctica

El sensible sensor piezoelectrónico

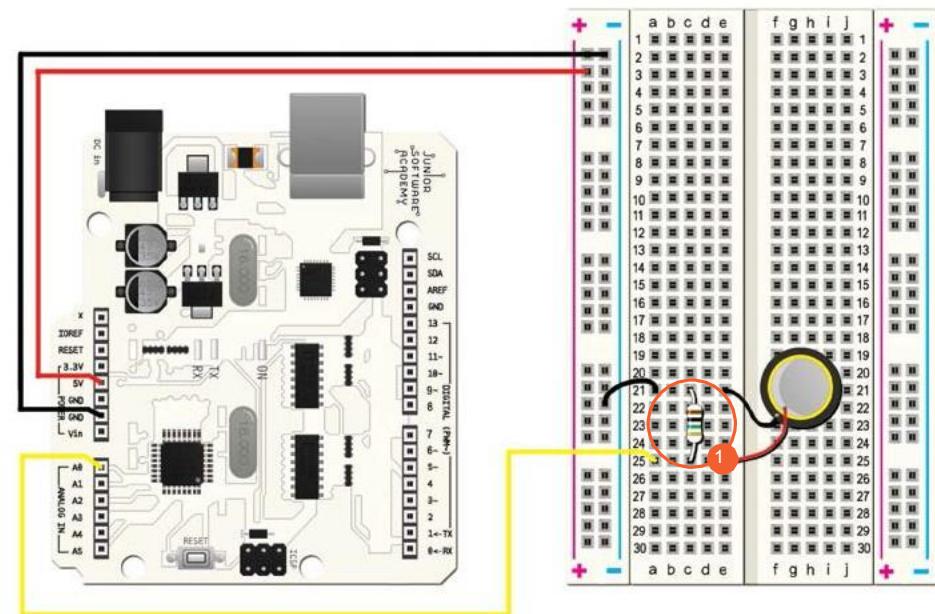
De los cuatro sensores que utilizaremos, el sensor piezoelectrónico es el único que se conecta de manera distinta de los otros tres sensores. Primero, veremos la construcción del circuito para conectar el sensor piezoelectrónico.

Objetivos del proyecto

Construir un circuito para conectar el sensor piezoelectrónico a la placa de pruebas y la placa de creación.

Expresión del hardware

No es necesario conectar la alimentación y el voltaje para el sensor piezoelectrónico. Esto se debe a que la placa Arduino lee el voltaje generado en el sensor como una entrada analógica. Ten en cuenta que el sensor requiere una resistencia de $1\text{ M}\Omega$ para funcionar, una resistencia mayor que



1 Resistencia de $1\text{ M}\Omega$

Pensar en código

```

01 int knockSensor = A0;----- Conecta el sensor al pin analógico 0.
02 int sensorVal = 0;----- Ingresa el valor inicial 0 a la variable para guardar el
                           valor del sensor.

04 void setup() {----- Fija la velocidad básica para las comunicaciones en serie.
05   Serial.begin(9600);-----}

07 void loop() {----- Lee el valor analógico que proviene del
08   sensorVal = analogRead(knockSensor);----- sensor. Transmite el valor registrado a la
09   Serial.println(sensorVal);----- computadora.
10   delay(500);----- Luego de retrasarlo por 0,5 segundos, inicia la función
11 }----- loop().
12 }
13 
```

Resultados del proyecto

Luego de cargar el *sketch*, podrás detectar la información de sonido enviada a través del puerto de serie de la placa de creación al hacer clic en el botón “Monitor Serie” a la derecha de la barra de herramientas.

Los sensores que utilizaremos tienen tres patas, a excepción del sensor piezoeléctrico. Los demás sensores se ensamblan de la misma forma en el circuito; únicamente el sensor piezoeléctrico se conecta de manera diferente. Sin embargo, los sensores que tienen tres patas tienen los terminales positivo y negativo, a diferencia de aquellos que tienen dos patas. El sensor no se activará si no está conectado adecuadamente a los terminales correctos y no podrás verificar los valores detectados por el sensor. A veces, los componentes pueden dañarse. Si no sabes cómo conectarlos, es mejor verificar la hoja de datos de los componentes antes de conectarlos. Realiza las conexiones del circuito teniendo en cuenta la polaridad.

¿Qué es una hoja de datos?

Para ser exactos, es como un instructivo para los componentes electrónicos. Las hojas de datos describen características como la empresa fabricante de los productos, el número del producto, el símbolo en el circuito y el voltaje de activación.



PIN 1, +V_s; PIN 2, V_{OUT}; PIN 3, GND



Esta es la **VISTA INFERIOR**. Indica lo que verás si levantas el componente y observas el extremo inferior. En otras palabras: la energía, la salida analógica y la conexión a tierra (GND) se deben conectar de izquierda a derecha en el orden correcto. Aprenderemos más sobre esto comparando el método de conexión en el diagrama de circuito que aparece debajo.

Práctica

Cómo usar el potenciómetro, el sensor de temperatura y el micrófono

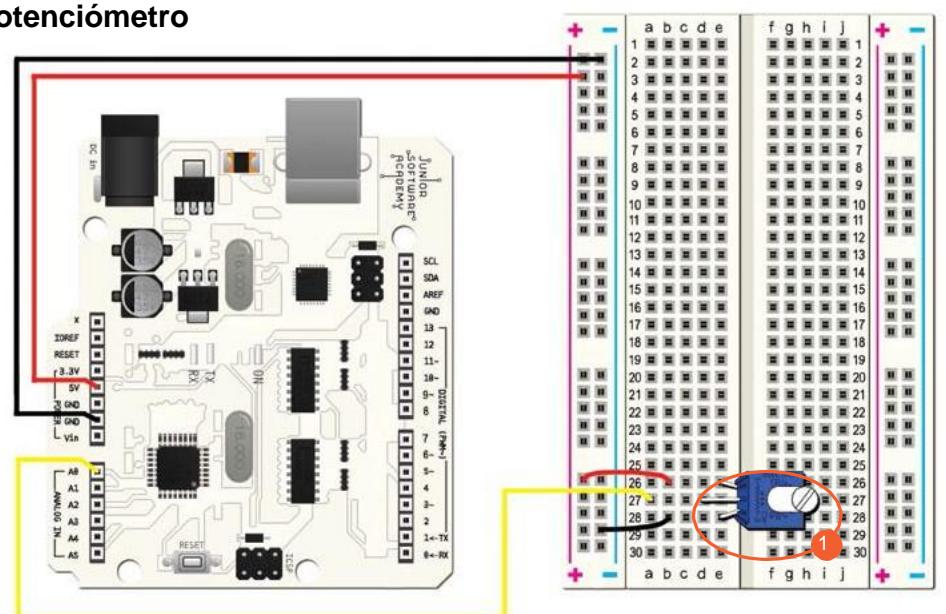
Probaremos utilizar otros sensores analógicos diferentes del sensor piezoelectrónico.

Objetivos del proyecto

Construir un circuito para cada sensor al realizar cambios en la construcción del circuito del sensor piezoelectrónico.

Expresión del hardware

Potenciómetro

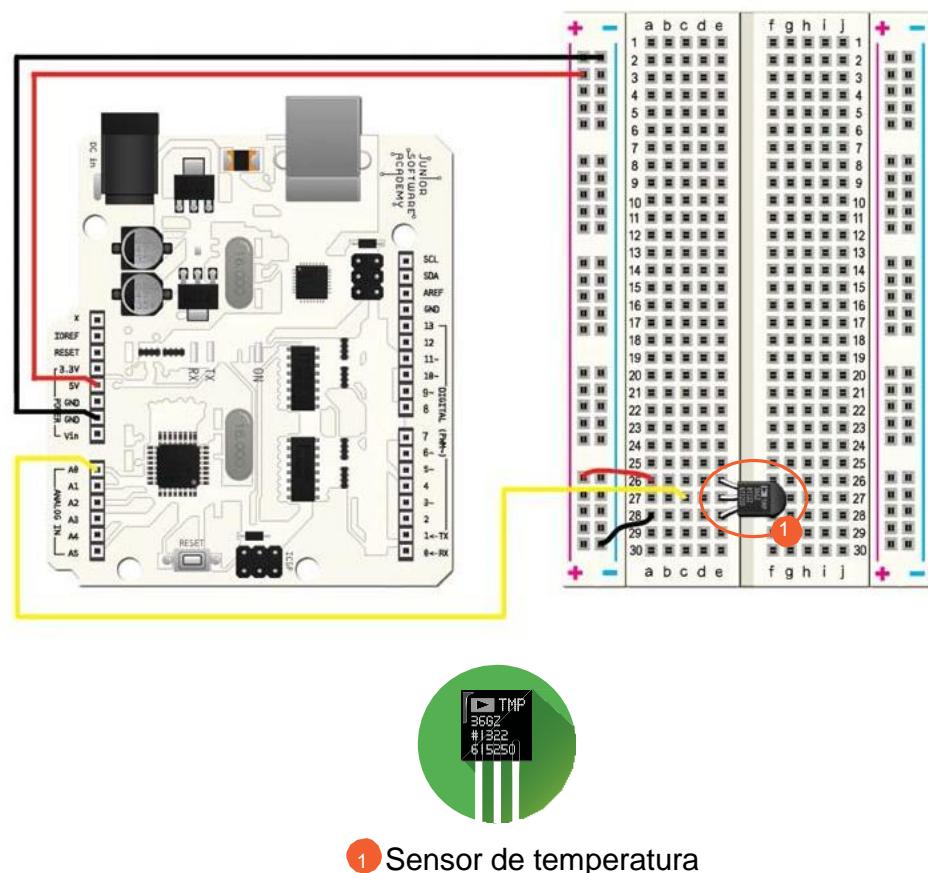


1 Potenciómetro

El potenciómetro tiene tres patas pero no distingue estrictamente las polaridades. Siempre que la señal para el terminal del medio esté conectada, el potenciómetro funcionará bien. Al girar la perilla hacia el terminal positivo, se approxima al valor máximo, mientras que al girar la perilla hacia el terminal negativo, se approxima al valor mínimo.



Sensor de temperatura

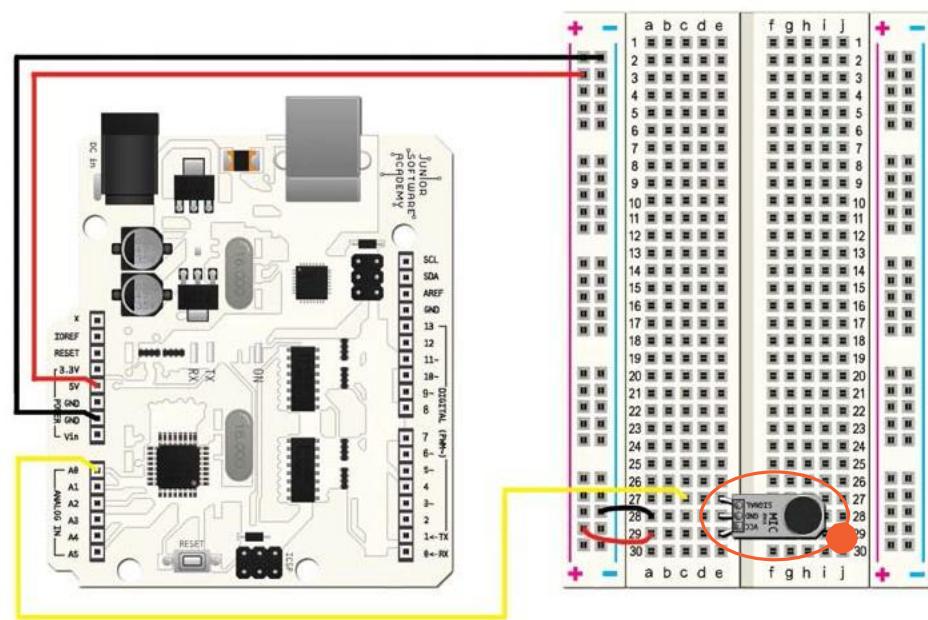


1 Sensor de temperatura



A diferencia del potenciómetro, deberás tener en cuenta la polaridad al conectar el sensor de temperatura. Si lo conectas de manera inversa, la placa de pruebas a la que conectaste el sensor podría calentarse demasiado y derretirse. Es por eso que nunca debes tocar la placa de pruebas cuando está energizada, incluso si el valor de la temperatura no se mide adecuadamente. Para poder tocar la placa, deberás desconectar el cable USB y la alimentación de energía y esperar a que esté fría. Desde el lado del componente que muestra su nombre, el lado izquierdo es el positivo.

Micrófono



Sensor

Pensar en código

01	int sensorPin = 0;	Conecta el sensor al pin analógico 0.
02	int sensorVal = 0;	Ingresa el valor inicial 0 a la variable para guardar el valor del sensor.
03		
04	void setup() {	
05	Serial.begin(9600);	Prepara la comunicación en serie.
06	}	
07		
08	void loop() {	
09	sensorVal = analogRead(sensorPin);	Lee el valor analógico proveniente del sensor.
10	Serial.println(sensorVal);	Transmite el valor leído a la computadora. Luego de
11	delay(500);	retrasarlo por 0,5 segundos, inicia la función <i>loop()</i> .
12	}	
13		

Resultados del proyecto

Luego de cargar el *sketch*, podrás detectar los datos analógicos enviados a través del puerto de serie de la placa de creación al hacer clic en el botón “Monitor Serie” a la derecha de la barra de herramientas.

Verifica el valor de salida de cada sensor y anota el rango de valores.

Sensor	Potenciómetro	Sensor de temperatura	Micrófono
Valor de salida			
Rango del valor			

Práctica

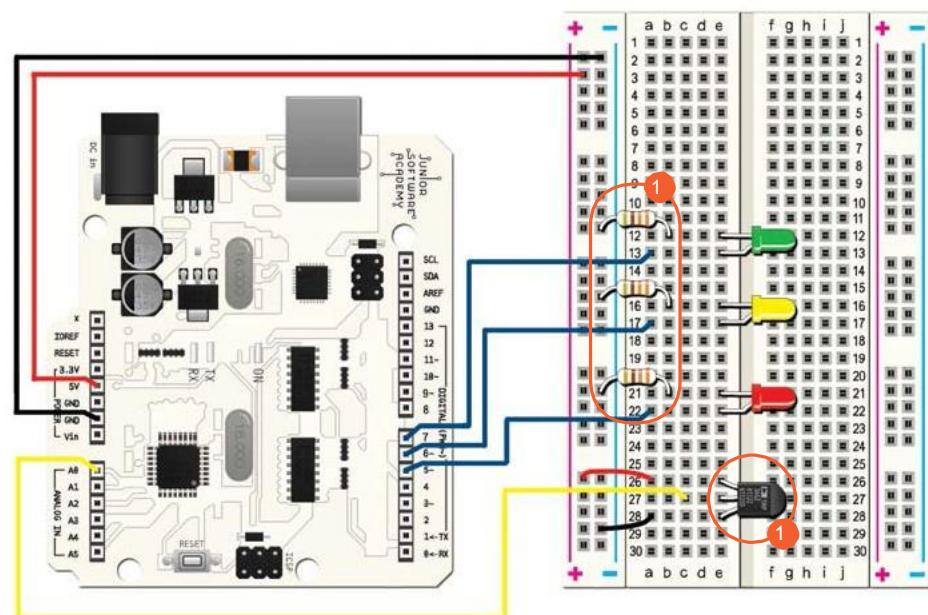
Cómo crear un semáforo que reaccione ante los cambios de temperatura

Ejecutaremos una estructura de circuito que se ilumine cuando se cumpla la condición de temperatura con ledes de tres colores. El sistema se basará en el código para medir la temperatura que escribimos antes. Utilizando expresiones condicionales, ejecutaremos la salida digital según el rango de un valor particular. Será más fácil de comprender si lo imaginas como un sistema que emite una alarma cuando la temperatura alcanza un determinado rango.

Objetivos del proyecto

Crearemos un semáforo que detecte y reaccione ante los cambios de temperatura.

Expresión del



1 Resistencia de 330 Ω



2 Sensor de temperatura

Pensar en código

```

01 int sensorPin = 0;
02 int sensorVal = 0;
03
04 int ledR = 5;
05 int ledY = 6;
06 int ledG = 7
07 ;
08
09 void setup() {
10   Serial.begin(9600
11   pinMode(ledR, OUTPUT
12   ); pinMode(ledY, OUT
13   PUT); pinMode(ledG,
14   } OUTPUT);
15
16 void loop()
17 {
18   sensorVal = analogRead(sensorPin);
19   float voltage = sensorVal * 5.0;
20   voltage /= 1024.0; -----
21   float temperatureC = (voltage - 0.5)
22   * 100; Serial.println(temperatureC);
23   int temp = temperatureC;
24
25   if (temp < 27) {
26     digitalWrite(ledR, HIGH);
27     digitalWrite(ledY, LOW); digit
28     alWrite(ledG, LOW);
29   } else if (temp < 28) {
30     digitalWrite(ledR, LOW); digit
31     alWrite(ledY, HIGH); digitalWr
32     ite(ledG, LOW);
33   } else if (temp < 29) {
34     digitalWrite(ledR, LOW); digit
35     alWrite(ledY, LOW); digitalWri
36     te(ledG, HIGH);
37   }
38   delay(100);
39 }
```

Conecta el sensor al pin analógico 0.
Ingresa el valor inicial 0 a la variable para guardar el valor del sensor.
Define el pin digital 5 para el led rojo. Define el pin digital 6 para el led amarillo. Define el pin digital 7 para el led verde.

Calcula el voltaje como valor de entrada*5/1024.
Transforma la tensión en temperatura en Celsius.

El número se cambia a números enteros para poder utilizarlo en la expresión condicional que aparece debajo.

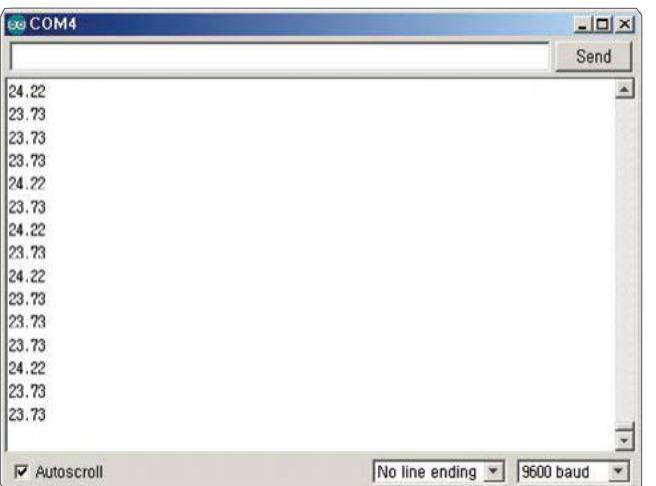
REVISIÓN → Repaso

- 1 Asignar un valor en números enteros para los puertos que se utilizarán en este programa.
- 2 Configurar los pines conectados a ledes en modo de salida (*output*).
- 3 Transformar la tensión medida en temperatura
- 4 Transformar la temperatura a números enteros
- 5 Fijar los rangos de antemano y decidir qué ledes se iluminarán

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Resultados del proyecto

Luego de cargar el *sketch*, podrás detectar la información de temperatura enviada a través del puerto de serie de la placa de creación al hacer clic en el botón "Monitor Serie" a la derecha de la barra de herramientas.



Piensa en la parte que más te costó y compártela con tus compañeros.



Puedes aplicar este proyecto a los siguientes ejemplos:

- Mostrar la temperatura en Fahrenheit (utilizando la fórmula de conversión de Celsius a Fahrenheit)
- Cambiar el intervalo de medición de la temperatura

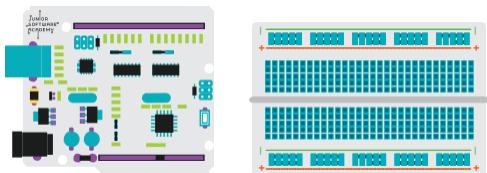
SEMANA
08

Reproducir sonidos

Objetivos de aprendizaje

- Reproducir sonidos que cambien de diversas maneras utilizando sensores y un zumbador, en base a lo que aprendiste en las semanas 6 a 7.
- Expresar escalas de notas según la frecuencia
- Aprender a utilizar la función *tone()*.

Materiales para



Placa de creación

Placa de pruebas



Cable USB



Resistencia de 10 KΩ



Resistencia de 330 Ω



Zumbador



Sensor de luz



Cable



LECCIÓN

Sonidos libres con un zumbador

De la misma manera que detectamos variaciones con el sensor piezoeléctrico, para esta clase utilizaremos un zumbador que produzca sonidos. Adentro del cilindro del cuerpo del zumbador, como se muestra en la imagen de la derecha, hay un sensor piezoeléctrico fijo. Si enviamos señales eléctricas, el sensor piezoeléctrico vibra y comienza a producir sonidos. Además, el volumen del sonido varía según el tamaño del zumbador. Los zumbadores vienen con diversos tamaños, de pequeños a grandes.



Ahora, reproduciremos sonidos al conectar el zumbador a la placa de creación.

Práctica

Cómo construir una placa que emite sonidos con zumbadores

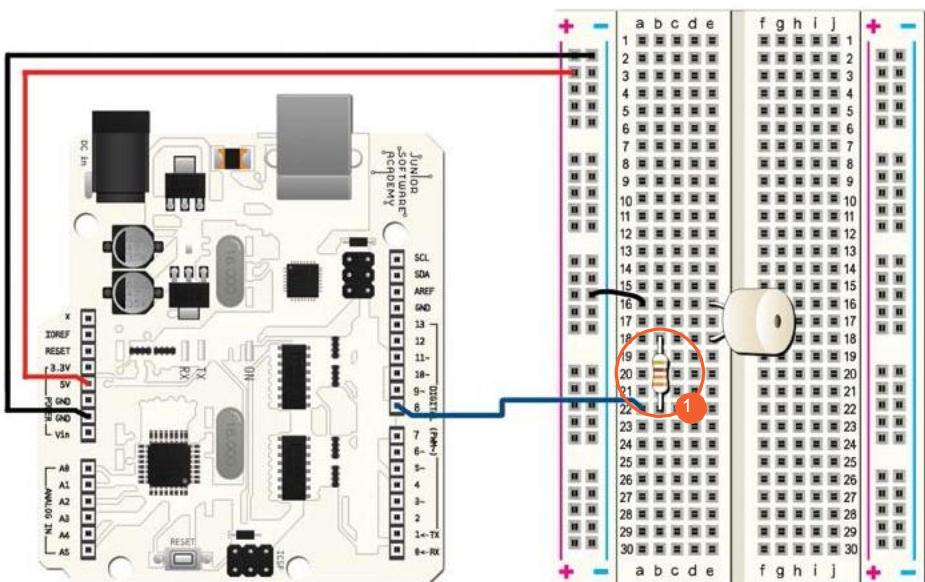
Los zumbadores generalmente se utilizan como alarmas. Por ejemplo, cuando hay un error en tu computadora y no enciende, distintos tipos de alarmas suenan según el tipo de error. Puedes saber de qué error se trata por el tipo de alarma.

Ahora, conectaremos el zumbador a Arduino para reproducir sonidos reales. Conectar el zumbador es muy sencillo. No es necesario preocuparse por terminales positivos y negativos, solo es necesario conectarlo en los pines correctos.

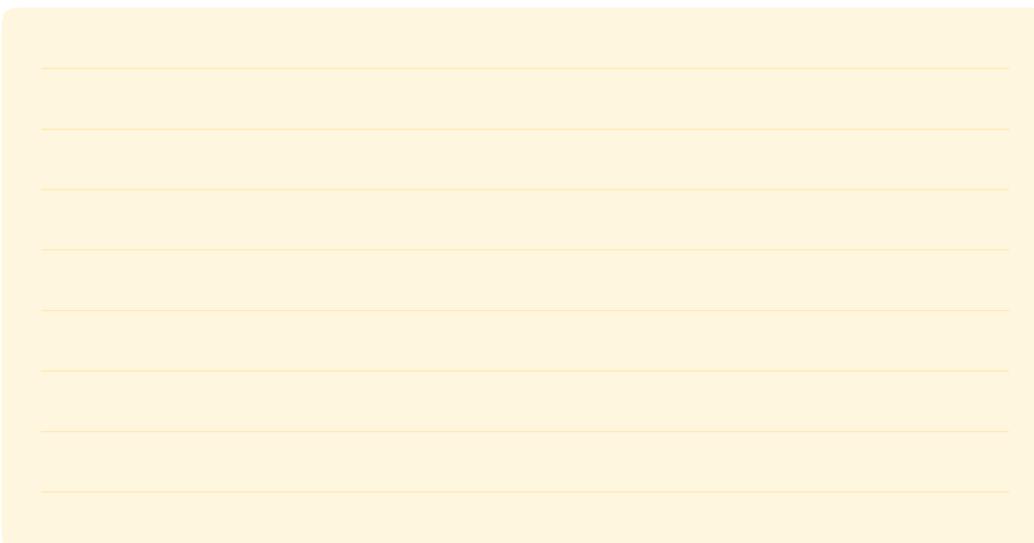
Objetivos del proyecto

Construir un circuito que conecte el zumbador a la placa de creación y la placa de prueba.

Expresión del hardware



① Resistencia de 330 Ω



Pensar en código

```

01 int speakerSensor =
02
03 8; void setup() {
04   pinMode(speakerSensor, OUTPUT);
05 }
06
07 void loop() {
08   digitalWrite(speakerSensor, HIGH); delay(1000);
09   digitalWrite(speakerSensor, LOW); delay(1000);
10 }
11
12 }
```

Resultados del proyecto

El zumbador repite zumbidos en intervalos de un segundo. Este *sketch* expresa el estado de encendido/apagado del sonido, al igual que el estado de encendido/apagado del led.

Práctica

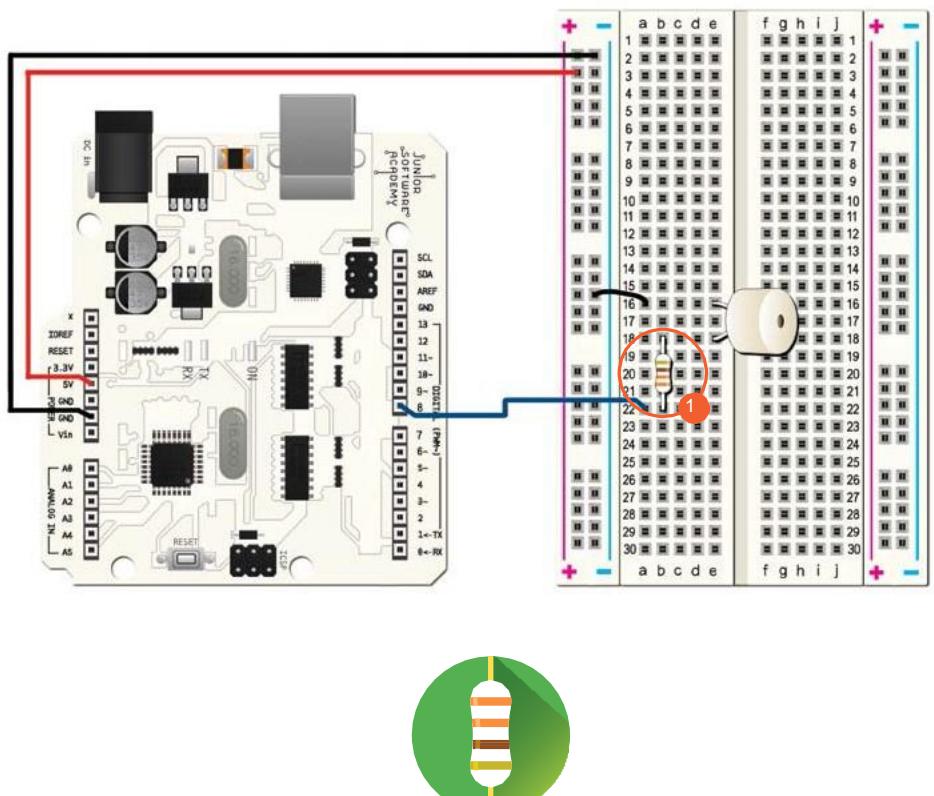
Cómo crear un piano simple con el zumbador

Los zumbadores no solo hacen “¡BZZZ!”. También pueden expresar las notas Do, Re, Mi , Fa, Sol, La, Si, Do. Haremos que la placa de creación toque las notas de un piano utilizando un zumbador. Podrás tocar cualquier canción si lo complementas con funciones adicionales.

Objetivos del proyecto

Expresaremos las notas Do, Re, Mi , Fa, Sol, La, Si, Do con un zumbador.

Expresión del hardware



Pensar en código

```

01 int note_c4 = 262;
02 int note_d4 = 294;
03 int note_e4 = 330;
04 int note_f4 = 349;
05 int note_g4 = 392;
06 int note_a4 = 440;
07 int note_b4 = 494;
08 int note_c5 = 523;
09
10 int speakerSensor = 8;
11 int noteDuration = 500;
12
13 void setup() {
14   pinMode(speakerSensor, OUTPUT)
15 }
16
17
18 void loop() {
19   tone(speakerSensor, note_c4, noteDuration
20 ); delay(500);
21   tone(speakerSensor, note_d4, noteDuration
22 ); delay(500);
23   tone(speakerSensor, note_e4, noteDuration
24 ); delay(500);
25   tone(speakerSensor, note_f4, noteDuration
26 ); delay(500);
27   tone(speakerSensor, note_g4, noteDuration
28 ); delay(500);
29   tone(speakerSensor, note_a4, noteDuration
30 ); delay(500);
31   tone(speakerSensor, note_b4, noteDuration
32 ); delay(500);
33   tone(speakerSensor, note_c5, noteDuration
34 ); delay(500);
35 }
```

Establece una frecuencia para Do.
Establece una frecuencia para Re.
Establece una frecuencia para Mi.
Establece una frecuencia para Fa.
Establece una frecuencia para Sol.
Establece una frecuencia para La.
Establece una frecuencia para Si.
Establece una frecuencia para Do.
Conecta el zumbador al pin digital 8.
Inserta un valor a la variable para guardar la duración del sonido.

Reproduce el sonido correspondiente a Do por 0,5 segundos.
Reproduce el sonido correspondiente a Re por 0,5 segundos.
Reproduce el sonido correspondiente a Mi por 0,5 segundos.
Reproduce el sonido correspondiente a Fa por 0,5 segundos.
Reproduce el sonido correspondiente a Sol por 0,5 segundos.
Reproduce el sonido correspondiente a La por 0,5 segundos.
Reproduce el sonido correspondiente a Si por 0,5 segundos.
Reproduce el sonido correspondiente a Do por 0,5 segundos.

Resultado del proyecto

Escribimos un *sketch* con un código que reproduce las notas Do, Re, Mi, Fa, Sol, La, Si y Do.

Aquí utilizaremos una función nueva. Es la función *tone()*. Si utilizas esta función, puedes reproducir sonidos con facilidad.

¿Puedes crear tu propia música?



tone(número de pin, tono de la nota, duración de la nota)

- Número de pin: el número de los pines a los que conectaste el zumbador.
- Tono: establece el tono de la nota.
- Duración de la nota: configura la duración del sonido.

Práctica

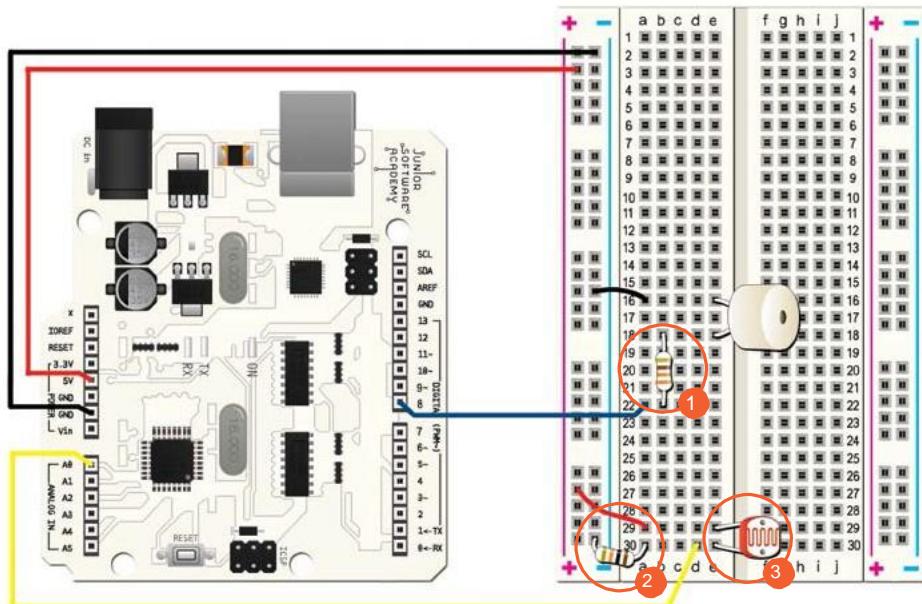
Cómo tocar un instrumento con el sensor de luz

Puedes crear un instrumento simple con el sensor de luz y el zumbador que utilizamos antes. ¿Qué tal si lo intentas?

Objetivos del proyecto

Crear un instrumento simple y sencillo con un sensor de luz.

Expresión del hardware



① Resistencia de 330 Ω



② Sensor de luz



③ Resistencia de 10K Ω

Pensar en código

```

01  • int note_c4 = 262; ..... Fija una frecuencia para Do.
02  int note_d4 = 294; ..... Fija una frecuencia para Re.
03  int note_e4 = 330; ..... Fija una frecuencia para Mi.
04  int note_f4 = 349; ..... Fija una frecuencia para Fa.
05  int note_g4 = 392; ..... Fija una frecuencia para Sol.
06  int note_a4 = 440; ..... Fija una frecuencia para La.
07  int note_b4 = 494; ..... Fija una frecuencia para Si.
08  • int note_c5 = 523; ..... Fija una frecuencia para Do.

09
10 int speakerSensor = 8; ..... Conecta el zumbador al pin digital 8.
11 int noteDuration = 100; ..... Fija la duración de la nota a 0,1
12 ..... segundos.
13 int cdsPin = 0; ..... Conecta el sensor al pin analógico 0.
14 int cdsValue = 0; ..... Inserta el valor inicial 0 en la variable
15 int noteValue = 0; ..... que guarda el valor del sensor.
16
17 void setup() {
18     pinMode(speakerSensor, OUTPUT);
19     Serial.begin(9600);
20 }
21
22 void loop() {
23     • cdsValue = analogRead(cdsPin);
24
25     • if(cdsValue < 200) {
26         noteValue = note_c4;
27     } else if(cdsValue > 200 && cdsValue < 230) {
28         noteValue = note_d4;
29     } else if(cdsValue > 230 && cdsValue < 260) {
30         noteValue = note_e4;
31     } else if(cdsValue > 260 && cdsValue < 290) {
32         noteValue = note_f4;
33     } else if(cdsValue > 290 && cdsValue < 320) {
34         noteValue = note_g4;
35     } else if(cdsValue > 320 && cdsValue < 350) {
36         noteValue = note_a4;
37     } else if(cdsValue > 380 && cdsValue < 410) {

```

Fija una frecuencia para Do.
Fija una frecuencia para Re.
Fija una frecuencia para Mi.
Fija una frecuencia para Fa.
Fija una frecuencia para Sol.
Fija una frecuencia para La.
Fija una frecuencia para Si.
Fija una frecuencia para Do.

Conecta el zumbador al pin digital 8.
Fija la duración de la nota a 0,1 segundos.
Conecta el sensor al pin analógico 0.
Inserta el valor inicial 0 en la variable que guarda el valor del sensor.
Inserta el valor inicial 0 a la variable para guardar la nota.

```

38     noteValue = note_b4;
39 } else if(cdsValue > 410) {
40     noteValue = note_c5;
41
42
43     Serial.println(cdsValue);
44     • tone(speakerSensor, noteValue, noteDuration);
45     delay(10);
46 }

```

- 1 Define la frecuencia para cada nota.
- 2 Guarda el valor del sensor de luz en *cdsValue*.
- 3 Con los enunciados *if/else...if*, los rangos se dividen en la cantidad de notas que existen entre los valores mínimos y máximos del sensor de luz.
- 4 Reproduce el sonido según la frecuencia del valor insertado al pin 8 mediante la función *tone()*.

Resultados del proyecto

Podrás detectar el brillo de la luz con el sensor de luz. Primero divide los rangos en el número de las notas, después de medir los valores mínimos y máximos del sensor de luz con el puerto de comunicación de serie. Luego ajusta el brillo y la intensidad del brillo se transformará en sonidos.



Puedes aplicar este proyecto a los siguientes ejemplos:

- Prueba crear sonidos utilizando los otros sensores.



SEMANA
09

DISEÑO CREATIVO 02

Hablemos con sonidos



Objetivos de aprendizaje

Crear algo al aplicar todo lo que aprendiste hasta la semana 8.

Crear objetos que reproduzcan sonidos.

Crear un nuevo proyecto que detecte los cambios exteriores y emita sonidos con un sensor.



Materiales para preparar

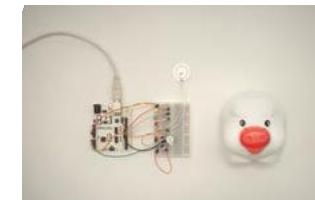
Kit de creación, diversos objetos (instrumentos de ritmo, instrumentos de percusión, tarjetas desplegables, alcancías, objetos que puedan utilizarse como instrumento), herramientas para manualidades.



Proyecto

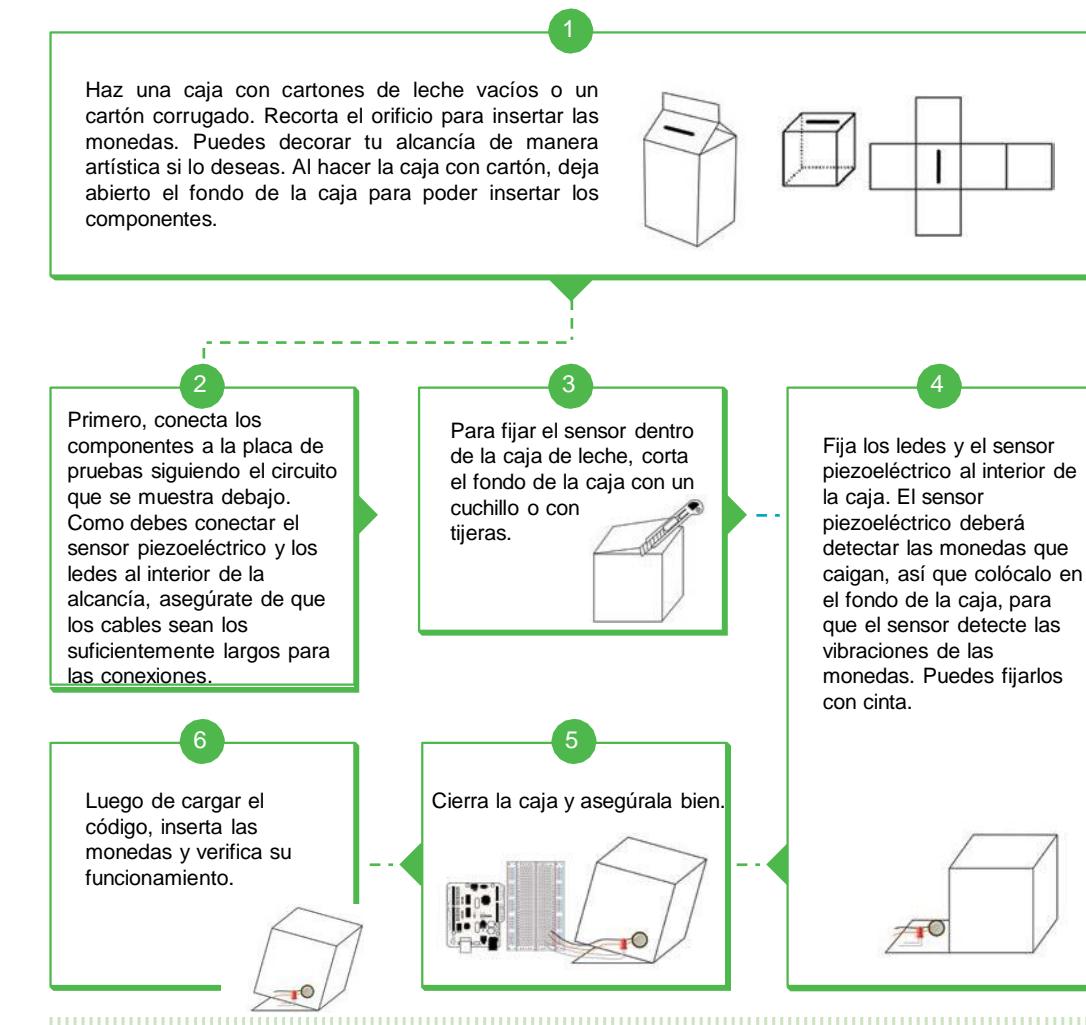
Alcancia musical

Este es un proyecto para una alcancia musical que podrán crear fácilmente siguiendo las instrucciones de cada etapa.



Objetivos del proyecto

Crear una alcancia que emita sonidos y luces al insertar monedas. Lee las instrucciones del objeto terminado y crea tu propia alcancia utilizando cajas de leche vacías o cartón corrugado.

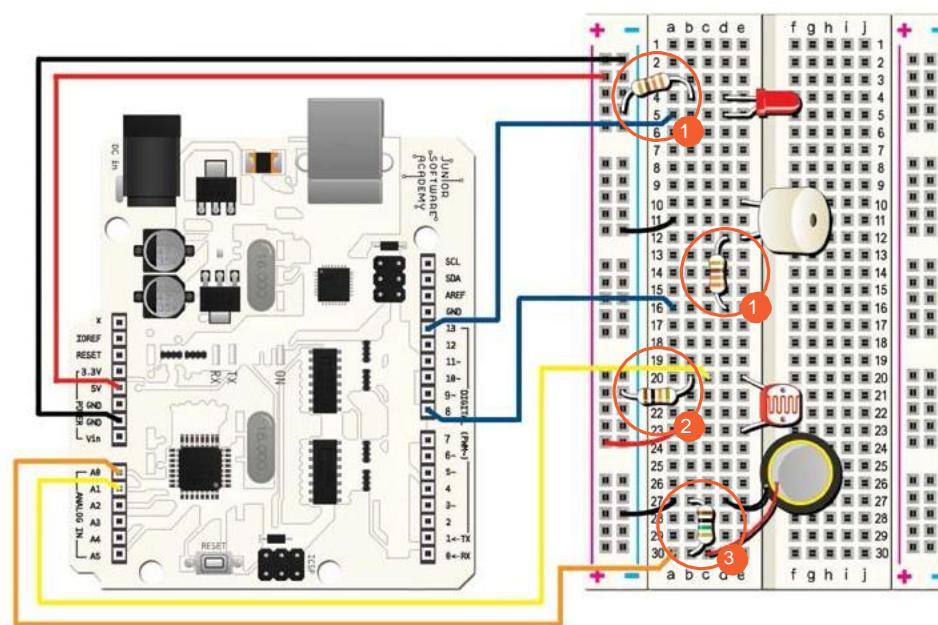




¿Solo podemos utilizar sensores piezoelectrinos para detectar movimientos?

Puedes imaginar muchas ideas posibles para tu propio proyecto. Otra posibilidad puede ser utilizar un rayo de luz infrarroja con ledes infrarrojos, verificar el estado del rayo infrarrojo del otro lado y registrar si se interrumpe. Lo más importante de un proyecto es la idea. Hay muchas maneras posibles de crear lo que deseas.

Expresión del hardware



1 Resistencia de 330 Ω



2 Resistencia de 10 K Ω



3 Resistencia de 1 M Ω

Pensar en código

```

01 int ledPin = 13; ----- Conecta el led al pin digital 13.
02
03
04 int knockSensor = 0; ----- Sensor piezoelectrico
05 int knockSensorThreshold = 100; ----- Conecta el sensor piezoelectrico al pin analógico 0 al recibir voltaje.
06 int knockSensorReading = 0; ----- Fija el valor básico con el que se activará el sensor piezoelectrico en 100.
07
08
09 int cdsSensor = 1; ----- Inserta el valor inicial de la variable para guardar el valor de vibración del sensor piezoelectrico.
10 int cdsSensorThreshold = 500; ----- Sensor de luz
11 int cdsSensorReading = 0; ----- Conecta el sensor de luz al pin analógico 1 al recibir voltaje.
12
13 int buzzerPin = 8; ----- Fija el valor básico con el que se activará el sensor piezoelectrico en 500.
14
15 void setup() { ----- Inserta el valor inicial de la variable para guardar el valor de brillo del sensor de luz.
16   pinMode(ledPin, OUTPUT); ----- Se
17   Serial.begin(9600); ----- rial.begin(9600);
18 }
19
20
21 void loop() { ----- Zumbador
22   knockSensorReading = analogRead(knockSensor); ----- Conecta el zumbador al pin 8.
23   ; ----- Configura el pin digital 13 en modo de salida.
24   cdsSensorReading = analogRead(cdsSensor); ----- Fija la velocidad básica para las comunicaciones en serie.
25
26   if(cdsSensorReading >= cdsSensorThreshold) { ----- Mide la tensión generada en el sensor piezoelectrico. Lee el valor de brillo del sensor de luz.
27     digitalWrite(ledPin, HIGH); ----- Enciende el led si el valor del sensor de luz es mayor o igual que el valor de referencia.
28   }
29   else { ----- Si no, apaga el led.
30     digitalWrite(ledPin, LOW); ----- Emite un sonido.
31   }
32
33   if (knockSensorReading >= knockSensorThreshold) { ----- Detiene el sonido.
34     for(int thisNote = 1; thisNote < 4; thisNote++) {
35       tone(buzzerPin, 440*thisNote, 100);
36       delay(130);
37       noTone(buzzerPin);
38     }
39   }
40
41   delay(10);
42 }
```

Puede parecer confuso, pero este *sketch* contiene las dos funciones para que el zumbador suene y la luz se encienda si el valor de entrada del sensor piezoelectrónico o el sensor de luz superan el valor de referencia. Además, se configuraron de antemano los valores de referencia de estas dos variables, *knockSensorThreshold* y *cdsSensorThreshold*. Al igual que con otros proyectos con los que trabajamos, los valores analógicos se leen con la función *analogRead()*. Utilizamos la función *tone()* para hacer sonar y detener el zumbador y utilizamos *digitalWrite()* para encender y apagar el led. Este proceso se repite en un intervalo de 0,1 segundos.

En función de los materiales que se muestran a continuación, puedes aplicar esto de muchas maneras. ¡Inténtalo!

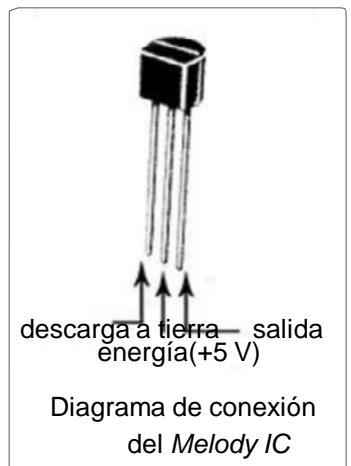


Componente para producir una melodía

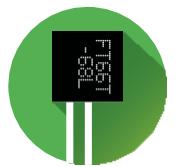
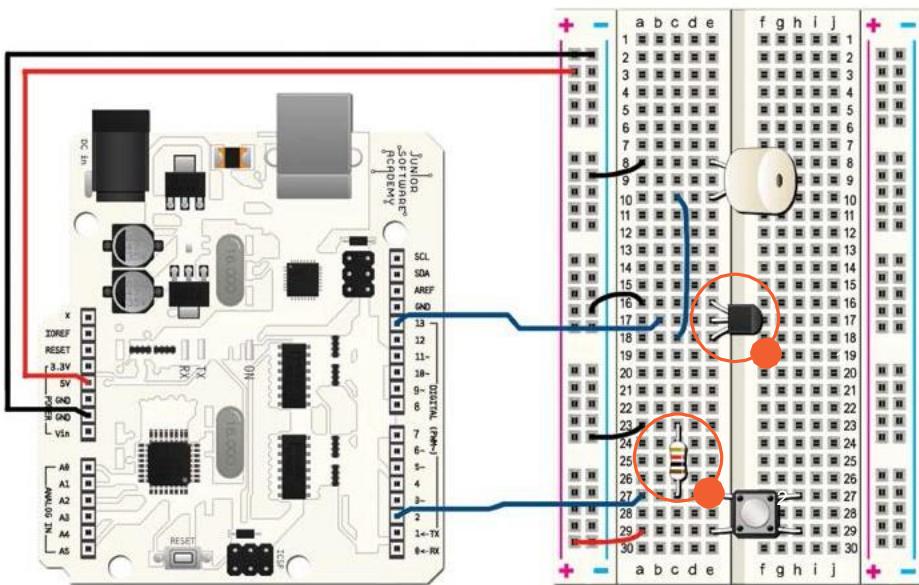
Primero, aprenderemos sobre los componentes necesarios para producir sonidos. Seguro habrás escuchado alguna vez una tarjeta con melodía o una caja musical. El componente para generar estos sonidos se conoce como *Melody IC*.

El *Melody IC* tiene tres patas. Sin embargo, el orden de las patas es distinto del orden del sensor de temperatura. Intenta no confundirlos.

Construiremos un circuito adicional utilizando este *Melody IC*.



Expresión del hardware



1 Melody IC



2 Resistencia de 1 KΩ

Pensar en código

```

01 int buttonPin = 2;-----o Conecta el botón al pin digital 2.
02 int melodyPin = 13;-----o Conecta el Melody IC al pin digital 13.

03
04 int buttonState = 0;-----o Inserta el valor inicial a la variable para
05 guardar el estado del botón presionado.

06 void setup() {
07     pinMode(melodyPin, OUTPUT);
08     pinMode(buttonPin, INPUT);
09 }
10
11 void loop() {
12     buttonState = digitalRead(buttonPin);-----o Lee el estado del botón y guarda la respuesta.

13
14     if(buttonState == HIGH) {-----o Al presionar el botón, la corriente fluye
15         digitalWrite(melodyPin, HIGH);-----o hacia el Melody IC.
16     }
17     else {
18         digitalWrite(melodyPin, LOW);
19     }
20 }
```

Si ya instalaste el componente *Melody*, revisa tu código para que se reproduzcan varias notas al insertar las monedas utilizando los valores de las notas. La semana anterior, declaramos el valor para cada nota en “Cómo tocar un instrumento con el sensor de luz”. Consulta la tabla de la derecha y cambia o agrega las notas que deseas.

Nota	B0	C1	CS1	D1	DS1	E1	F1	FS1	G1	GS1	A1	AS1
Valor	31	33	35	37	39	41	44	46	49	52	55	58
Nota	B1	C2	CS2	D2	DS2	E2	F2	FS2	G2	GS2	A2	AS2
Valor	62	65	69	73	78	82	87	93	98	104	110	117
Nota	B2	C3	CS3	D3	DS3	E3	F3	FS3	G3	GS3	A3	AS3
Valor	123	131	139	147	156	165	175	185	196	208	220	233
Nota	B3	C4	CS4	D4	DS4	E4	F4	FS4	G4	GS4	A4	AS4
Valor	247	262	277	294	311	330	349	370	392	415	440	466
Nota	B4	C5	CS5	D5	DS5	E5	F5	FS5	G5	GS5	A5	AS5
Valor	494	523	554	587	622	659	698	740	784	831	880	932
Nota	B5	C6	CS6	D6	DS6	E6	F6	FS6	G6	GS6	A6	AS6
Valor	988	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865
Nota	B6	C7	CS7	D7	DS7	E7	F7	FS7	G7	GS7	A7	AS7
Valor	1976	2093	2217	2349	2489	2637	2794	2960	3136	3322	3520	3729
Nota	B7	C8	CS8	D8	DS8							
Valor	3951	4186	4435	4699	4978							

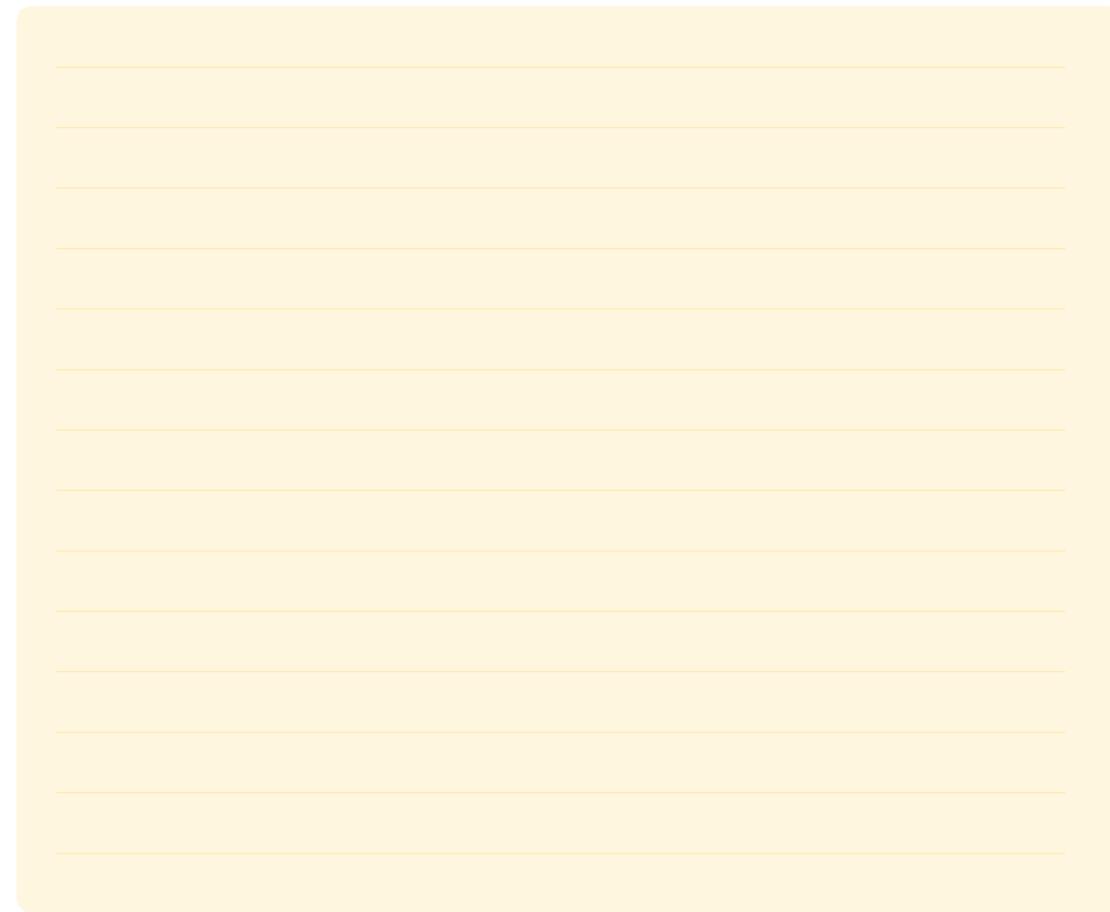
Hablemos con sonidos

¿Qué tal si creamos nuestro propio proyecto?

Ya creaste tu propio proyecto con lo que aprendiste hasta la semana 4. Ahora es momento de hacer algo increíble con todo lo que aprendiste hasta la semana 8.

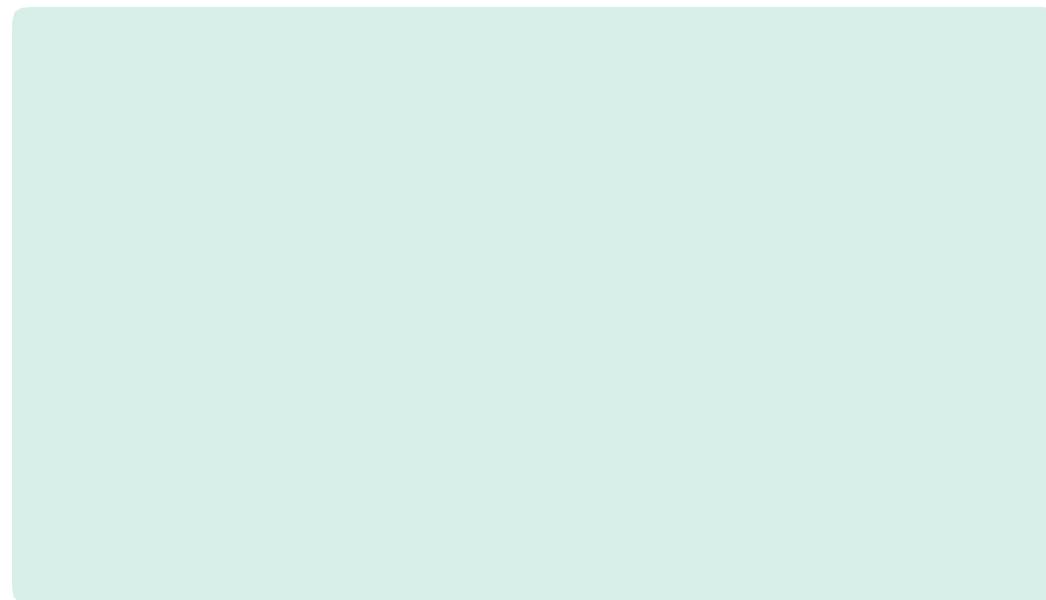
1 Formar grupos

Siéntense en grupos de 4 a 6. Piensen en una actividad grupal en cinco minutos. Escriban algunos *sketches* mientras piensan en distintas ideas. Cada uno deberá exponer sus ideas y qué roles tendrán sus compañeros por uno o dos minutos. Los compañeros del grupo no deberán juzgar si el proyecto es bueno o malo. En cambio, deberán brindar críticas constructivas que puedan ayudar a desarrollar las ideas de cada uno.



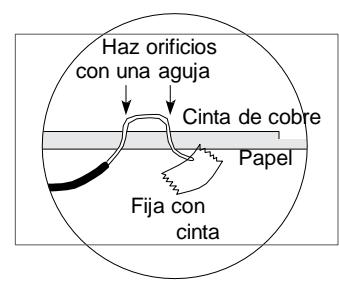
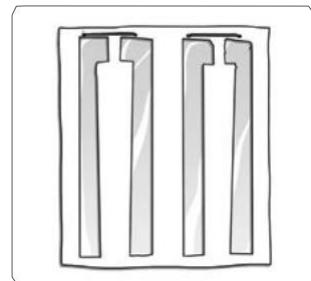
2 Resumir ideas

Crea un breve resumen de tus ideas con imágenes y textos. Luego, piensa en un *sketch*, la composición del hardware, un circuito y describe en detalle lo que harás y que acciones realizarás.



3 Conectar el circuito

Utilizando lo que aprendiste en las semanas 6 a 8, crea un proyecto que utilice los sensores. Es bueno utilizar los cables que vienen incluidos en el kit para realizar las conexiones. En caso de no tener cables, una forma de crear un circuito más resistente es soldar los terminales. Sin embargo, estamos trabajando sin soldar, por lo que deberás unir y retorcer los cables para conectarlos. También puedes fijarlos con una pistola de pegamento o cinta de cobre luego de unir los cables con los componentes,



4 Presentar

Como parte final de esta clase, nombra tu proyecto y haz una presentación acerca de tu creación.

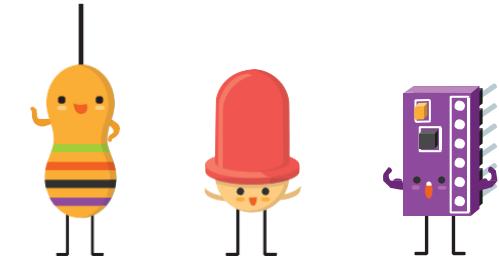
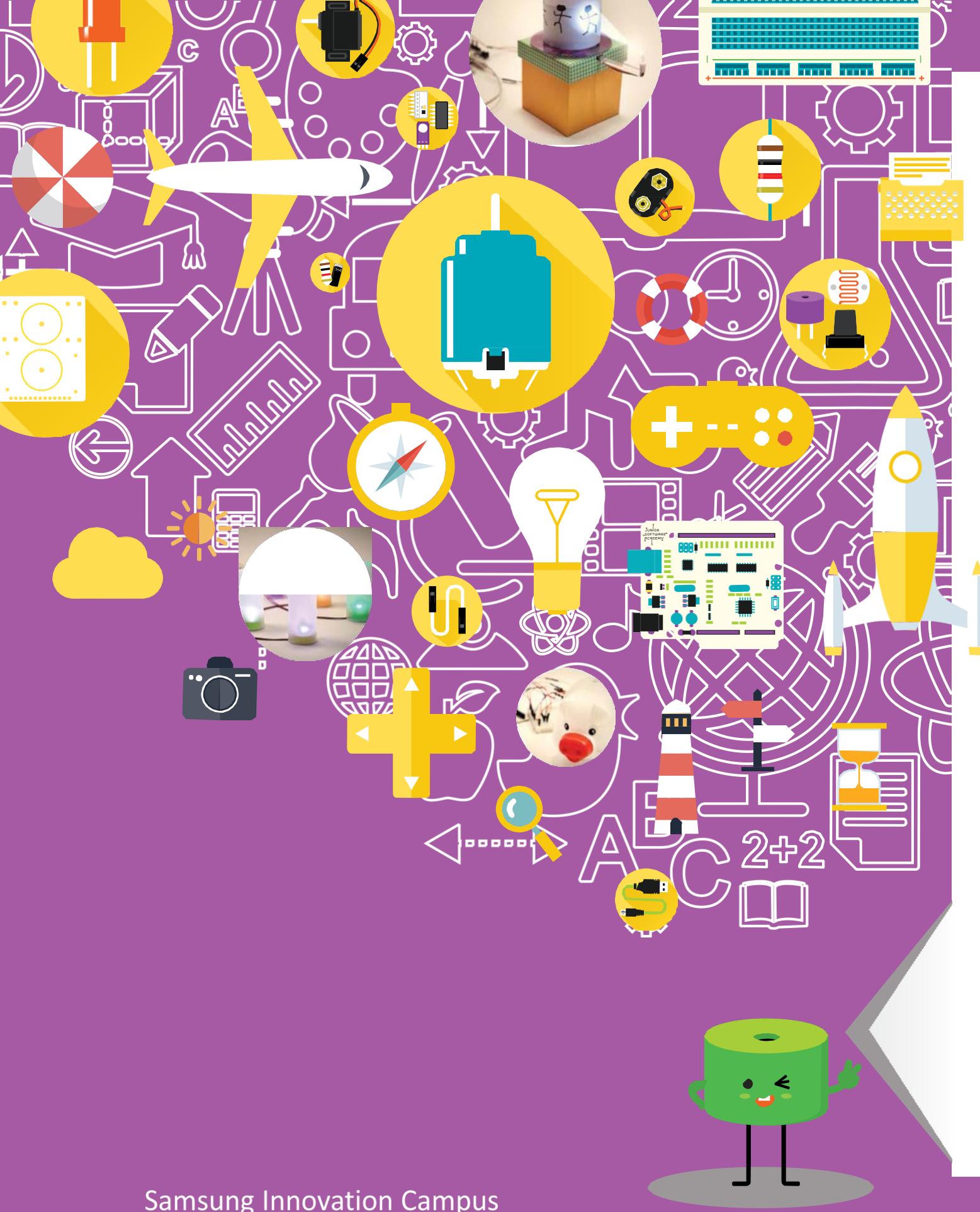
Durante la presentación deberás mencionar el nombre del proyecto y la descripción (por qué lo creaste; cómo; cómo funciona) y escucha las opiniones de tus compañeros sobre tu proyecto.

REVISIÓN

Repaso

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.



CLASE DE CREACIÓN

03

10. Robot dibujante con motor CC
11. Diseño creativo con servomotores 03
12. Prototipos digitales interactivos
13. Diseño creativo 4 (Hablemos con movimiento)
14. Prototipos digitales interactivos (Proyecto Final)

SEMANA

10

Robot dibujante

Objetivos de aprendizaje

- Utilizar un motor CC y construir un robot que dibuje con vibraciones.
- Comprender los principios básicos del motor y crear un motor simple con una bobina y una batería.

Materiales para preparar

Motor CC, baterías AA, soporte para baterías, vaso de papel, cuatro lápices para dibujar, otras decoraciones, un imán redondo, alambre esmaltado, dos sujetapapeles.



LECCIÓN

Ya has realizado varios proyectos que conectan la placa de creación y diversos sensores. A partir de ahora, utilizaremos la placa de creación y moveremos objetos al codificar acciones. Para esto, utilizaremos un motor.

¿Qué es un motor? El conductor, que deja pasar la energía, recibe la potencia del campo electromagnético. Así es como producimos la electricidad y con ella generamos energía rotacional. El generador produce electricidad y el motor produce energía rotacional. Estos dos tipos de energía son casi similares en términos estructurales pero la dirección de la potencia es diferente. La regla de Flemming explica el movimiento de cada potencia. Los motores transforman la energía eléctrica en energía cinética gracias al electromagnetismo.

Entonces, ¿dónde utilizamos motores? Puedes encontrarlos en autos de juguete o en refrigeradores. ¿Lo has notado? Presenta varios tipos de motores que hayas descubierto y describelos o dibújalos en el espacio que aparece debajo.

¿Qué tal si fabricamos un robot simple que utilice las funciones básicas de los motores?

Práctica

Alcancía musical

Objetivos del proyecto

Crear un robot que dibuje con las vibraciones del motor CC.

¿Qué es un robot dibujante?

Aprenderemos las características del motor CC al crear y activar el robot dibujante. Puedes crear un robot dibujante con un motor CC, un vaso de papel, lápices y otros materiales que puedes encontrar a tu alrededor. Al activarlo, el robot dibujante se moverá



por sí solo y comenzará a dibujar. Una vez que hayas terminado, verifica cada paso y diseña y crea tu propio robot dibujante.

El robot dibujante utiliza las funciones básicas del motor, por lo que no utiliza una placa de creación. Veamos los materiales para el robot dibujante y comencemos con el proyecto.

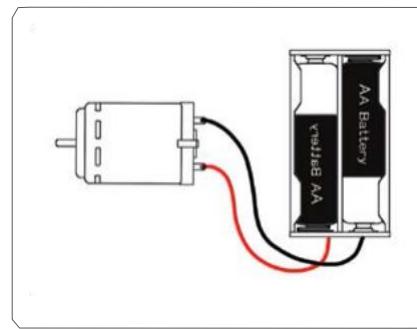
Materiales	Rol
Vaso de papel	Cuerpo del robot dibujante.
Motor	Hace que el robot vibre y dibuje.
Cinta	Se utiliza para fijar el motor CC, el soporte para baterías, los lápices para dibujar y otras decoraciones al cuerpo del robot (vaso).
Lápices	Se utilizan para dibujar, pero también conforman las patas del robot. Utiliza más de cuatro lápices para nivelar el cuerpo del robot y dibuja con el movimiento de las patas.
Decoración	Además de los materiales mencionados anteriormente, puedes preparar todo tipo de decoraciones para fijarlas al cuerpo.

Conectar el motor CC a una batería AA

Conecta el terminal positivo del motor CC a la batería AA con un alambre y verifica si el motor funciona. El voltaje que necesitas es de 1,5 V. Cuando el motor gire, analiza la dirección y la velocidad. Si colocas un trozo de cinta en el eje del motor, podrás verificar con facilidad la dirección y la velocidad del motor. Luego, desconecta el alambre y conéctalo al terminal negativo y verifica si el motor gira en la dirección opuesta.

Conectar el motor CC al soporte de baterías

Luego de colocar dos baterías en el soporte, conéctalo al motor CC. El voltaje que necesitas es de 3 V. Compara la dirección y la velocidad del motor CC cuando conectas una y dos baterías. Luego conéctalo al terminal del polo opuesto y verifica si

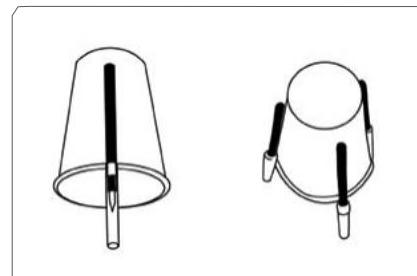


el motor gira en la dirección contraria. Como mencionamos anteriormente, para verificar la dirección y la velocidad del motor de manera más fácil, coloca un trozo de cinta en el eje del motor.

Expresión del hardware

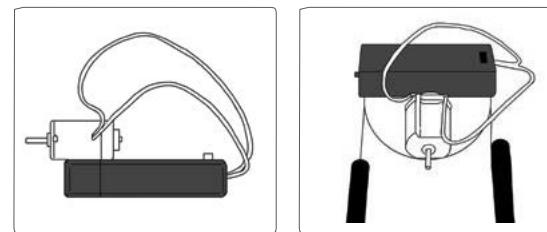
1 Cómo hacer el cuerpo

Fija tres o cuatro lápices para hacer las patas del robot. Presta atención al equilibrio del robot mientras lo haces.



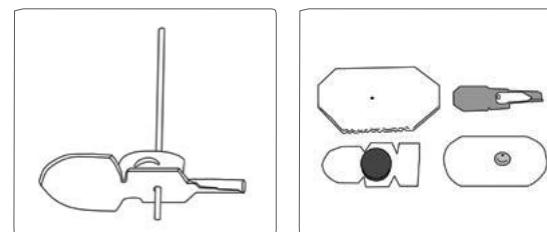
2 Cómo fijar el motor

Luego de conectar cada lado del cable al soporte para baterías en los terminales positivo y negativo, fija el soporte para baterías y el motor a la parte correspondiente del cuerpo del robot (vaso) e inserta las baterías AA para iniciar el funcionamiento.



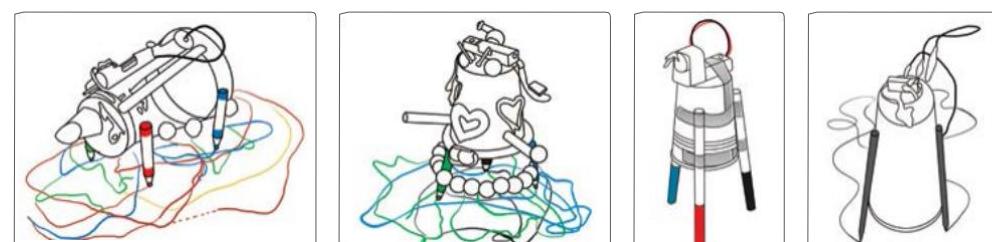
3 Decoración

Luego de verificar el movimiento del robot dibujante, retira la batería y decora el robot con los materiales que preparaste.



Resultados del proyecto

Puedes crear varios tipos de robots dibujantes según tus preferencias.



Crea un robot dibujante con tu propio estilo

Práctica

Crea un motor simple

Objetivos del proyecto

Crear un motor con alambre esmaltado e imanes.

¿Qué es un motor simplificado?

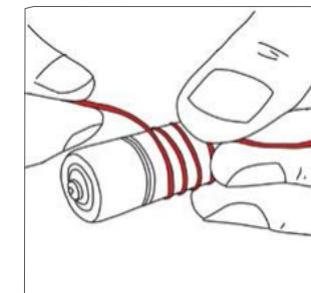
Intentaremos comprender el funcionamiento de un motor al crear un motor simplificado, que tiene una estructura similar a la de un motor CC, utilizando alambre esmaltado bobinado y un imán. Cuando la corriente fluye por el alambre esmaltado, el imán comienza a empujar y tirar del rotor y el eje comienza a girar.

Los voltajes de entrada de los motores CC tienen distintos rangos, que oscilan entre los 1.2 V y los 24 V. No hay separación de terminales positivos y negativos en los extremos de un motor CC. En lugar de eso, la dirección de rotación se invierte si el cable se fija al terminal del polo opuesto. Además, la velocidad de rotación del motor se relaciona con el voltaje. Si se provee un voltaje bajo, el motor gira lentamente. Si se aumenta el voltaje, el motor gira más rápido. Mientras más fuerte sea la corriente, más fuerte será el campo electromagnético que genera la potencia.

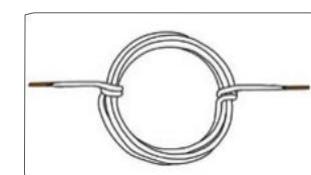
Expresión del hardware

1 Crear un rotor con el alambre esmaltado

Enrolla el alambre esmaltado alrededor de la batería con cuatro a seis vueltas.



Luego, enrulla alambres cortos dos o tres veces alrededor de la bobina de alambre esmaltado a ambos lados y extiende el resto del alambre.

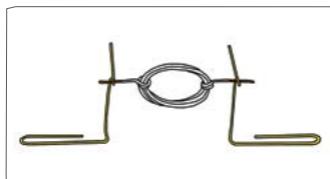
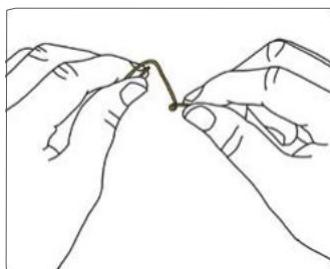


Este módulo es el rotor. Deberás retirar todo el revestimiento de esmalte de un lado y la mitad del otro lado. Para hacerlo, puedes utilizar papel de lija o el filo de unas tijeras.

② Crea un soporte para el rotor con sujetapapeles

Dobla la parte interior del sujetapapeles para crear una base y dobla la parte exterior para crear el soporte para el rotor. Consulta la imagen de la derecha.
Ahora ajusta la altura para que el imán y el rotor tengan 1 cm de separación.

Crea dos soportes y cuelga el rotor de alambre esmaltado que hiciste entre ambos soportes.

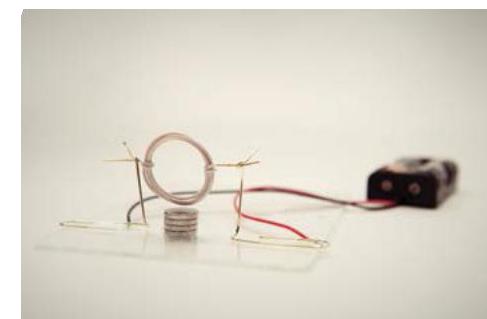
**③ Composición**

Inserta dos baterías AA en el soporte para baterías. Obtendrás un suministro de voltaje de 3 V. Conecta los cables del soporte para batería a cada uno de los sujetapapeles de soporte y coloca un imán entre los sujetapapeles.

Para comprender mejor las instrucciones, consulta la imagen debajo de Resultados del proyecto.

Resultados del proyecto

Luego de colocar las baterías en el soporte de baterías y hacer que la corriente fluya, toca suavemente el rotor de alambre, debería comenzar a moverse. Es muy fácil crear un motor en casa, ¿verdad?



REVISIÓN

Reaso

Piensa en la parte que más te costó y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.

SEMANA

11

El motor CC

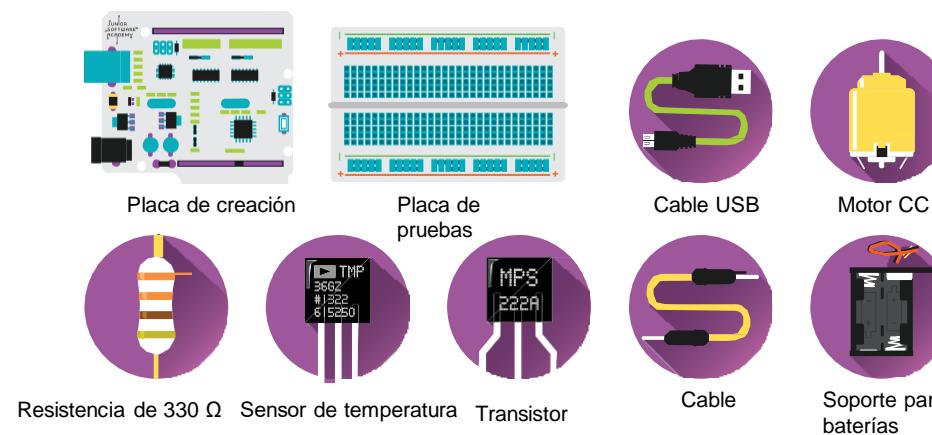
LECCIÓN



Objetivos de aprendizaje

- Controlar un motor CC utilizando la salida PWM de la placa Arduino.
- Controlar el motor CC utilizando los valores de resultado del sensor de temperatura.

Materiales para preparar



Durante la clase de hoy, utilizaremos un motor CC, la placa de creación y otros componentes. Antes de comenzar, analizaremos el motor CC, un componente importante, y también examinaremos las características de los transistores que utilizaremos junto con las resistencias.

El motor CC

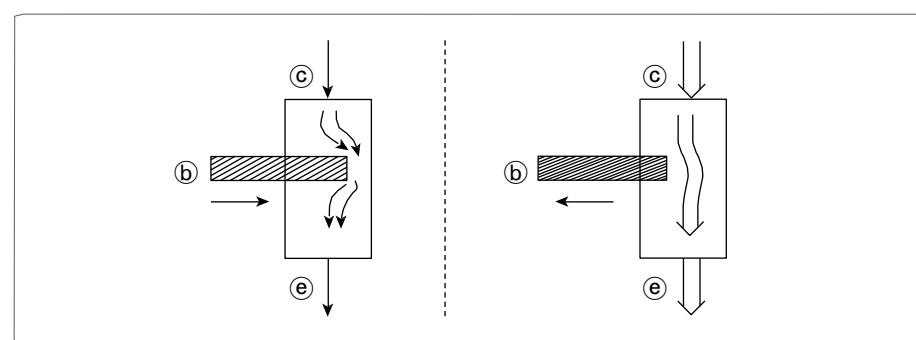
Como explicamos anteriormente, los motores son dispositivos que transforman la energía eléctrica en energía cinética y son fáciles de encontrar a nuestro alrededor. Se han convertido en objetos indispensables para nuestra vida diaria. Los motores eléctricos se dividen en motores CC y motores CA. La diferencia es que unos utilizan fuentes de energía con corriente alterna (CA) que cambia en términos de potencia y dirección con frecuencia, mientras que otros utilizan energía con corriente continua (CC) que fluye uniformemente de baja potencia a alta potencia. Los motores CA tienen la ventaja de tener estructuras pequeñas, sencillas y livianas, pero son difíciles de controlar con precisión. Por lo tanto, ya que pueden controlar con precisión la velocidad y generar una alta potencia, los motores CC se utilizan ampliamente.

Para modificar la velocidad del motor al cambiar el valor del voltaje que se envía al motor CC, deberás utilizar la salida analógica que vimos en la semana 4. En la vida real, utilizamos modulación por ancho de pulsos (*Pulse-Width Modulation, PWM*). Aprenderemos los conceptos en detalle y aplicaremos este método.

Transistor

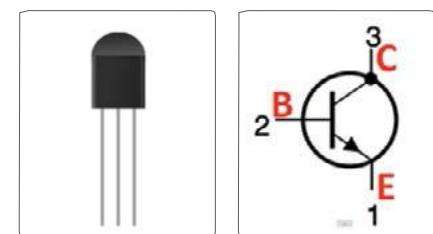
El transistor actúa como un interruptor que cambia la dirección de la corriente o aumenta la potencia de la corriente. El transistor que utilizaremos (2N2222), que tiene el número de serie P2N22AA50 al frente, controla el flujo de corriente enviado mediante su pata central (la base, que se denomina B) y puede permitir o bloquear el paso de corriente que fluye desde la pata derecha (el colector, que se denomina C) hasta la pata izquierda (el emisor, que se denomina E).

Primero, aprenderemos los principios básicos.



Imagina un tubo cilíndrico con agua que fluye de C a E. El tubo también cuenta con una compuerta llamada B. Si empujamos B hacia adentro, la entrada de agua se cierra y el flujo se detiene, y si halamos de B, el agua comienza a fluir. La forma en que el agua fluye de C a E al ejercer potencia sobre B es similar a la forma en que funciona un transistor.

En el proyecto real, el flujo de corriente se controla según el flujo enviado por la placa Arduino a la pata central del transistor.



Práctica

Cómo controlar el motor a voluntad

Ya vimos cómo la velocidad de rotación del motor cambia según la cantidad de voltaje. La salida digital solo permite dos estados, encendido o apagado, por lo que si queremos cambiar el voltaje que enviamos al motor, tendremos que utilizar la salida analógica. La función `analogWrite()` que aprendiste en la semana 4 controla las funciones analógicas. Esta función utiliza una tecnología conocida como modulación por ancho de pulsos (*PWM*) para crear señales analógicas virtuales utilizando pulsos digitales. No pueden crear una salida analógica real pero producen un efecto similar.

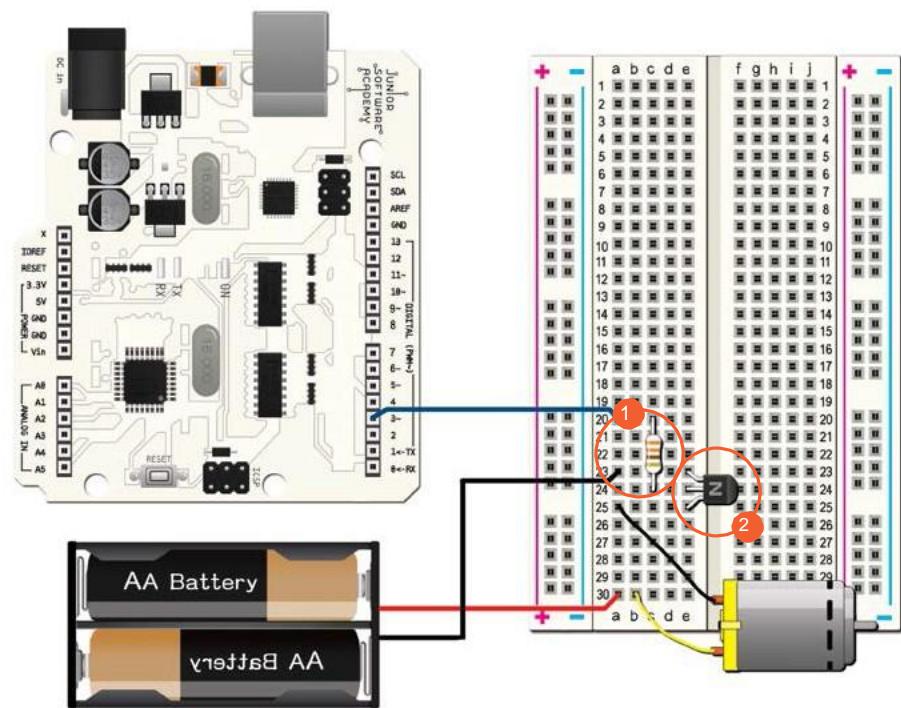
Las señales analógicas tienen un rango de 0 a 255, y puedes controlar la velocidad de rotación del motor con este valor. Al igual que cuando controlamos el brillo del led, utilizamos el número intermedio entre 0 y 255, que es 128. El voltaje desciende al 50 %, de 5 V a 2,5 V y la velocidad de rotación del motor desciende a la mitad de la velocidad original. Verificaremos esto al cambiar los valores. .

Objetivos del proyecto

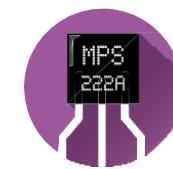
Intentaremos controlar el motor con dos métodos.

Expresión del hardware

Conecta el pin 3 de la placa Arduino a la base del transistor a través de una resistencia. La conexión se debe realizar a través de una resistencia porque cualquier exceso de corriente hacia el transistor podría dañarlo.



1 Resistencia de 330 Ω



2 Transistor

Pensar en código

```

01 int motorPin = 3; ----- Fija el pin 3 para conectar el motor.
02 int speed; ----- Declara la variable para guardar la velocidad.

03

04 void setup() {
05   Serial.begin(9600); ----- Inicialización de serie.
06   pinMode(motorPin, OUTPUT); ----- Fija el pin 3 en modo de salida.
07   while(!Serial); ----- Verifica si el puerto de serie está configurado
08   Serial.println("Speed 0 to 255"); ----- adecuadamente. Envía "Velocidad 0 a 255"
09 }

10

11 void loop() {
12   if (Serial.available()) { ----- Verifica si está listo para utilizar la
13     speed = Serial.parseInt(); ----- serie. Lee la velocidad de la serie en
14     if (speed >= 0 && speed <= 255) { ----- la variable de velocidad (speed).
15       analogWrite(motorPin, speed); ----- Verifica si la velocidad es mayor que 0
16       Serial.print("Speed : "); ----- y menor que 255.
17       Serial.println(speed); ----- Gira el motor con la velocidad
18     } ----- de entrada. Envía "Speed :"
19   }
20 }
```

Fija el pin 3 para conectar el motor.
Declara la variable para guardar la velocidad.

Inicialización de serie.
Fija el pin 3 en modo de salida.
Verifica si el puerto de serie está configurado adecuadamente. Envía "Velocidad 0 a 255" como serie.

Verifica si está listo para utilizar la serie. Lee la velocidad de la serie en la variable de velocidad (speed).
Verifica si la velocidad es mayor que 0 y menor que 255.
Gira el motor con la velocidad de entrada. Envía "Speed :"
como serie. Envía la velocidad como serie.

Si escribes la velocidad de rotación en la sección superior del código del monitor serie y presionas “Enviar” o la tecla Enter, el texto ingresado se transforma en números y cambia la velocidad del motor conectado al pin 3. La función `Serial.available()` verifica si el puerto serie funciona adecuadamente. La función `Serial.parseInt()` transforma el texto ingresado en el monitor serie a números enteros y los regresa. Puedes transmitir la velocidad que deseas con esta función. La función `analogWrite()` cambia la velocidad de rotación del motor.

Pensar en código

Si escribes la velocidad de rotación en la sección superior del código del monitor serie y presionas el botón “Enviar” o la tecla Enter, el texto ingresado se transforma en números y cambia la velocidad de rotación del motor conectado al pin 3. La función

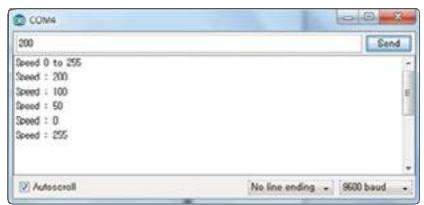
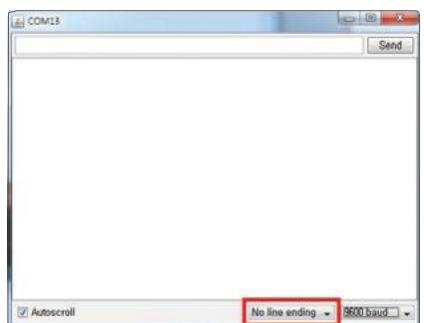
`Serial.available()` verifica si el puerto serie funciona adecuadamente. La función `Serial.parseInt()` transforma el texto ingresado en el monitor serie a números enteros y los regresa. Puedes transmitir la velocidad que deseas con esta función. La función `analogWrite()` cambia la velocidad de rotación del motor.

Para controlar el motor con el valor ingresado al monitor serie, deberás cambiar la segunda categoría desde la izquierda a “*No line ending*”.

Consulta la imagen de la derecha.

Resultados del proyecto

Si tienes el monitor serie en pantalla luego de comenzar a escribir el *sketch* e ingresas la velocidad del motor CC al monitor serie, podrás controlar la velocidad del motor CC según el valor ingresado.



Práctica

Cómo crear un ventilador que se encienda cuando la temperatura aumenta

Sería conveniente tener un ventilador que se encienda automáticamente cuando hace calor, como lo hacen los aires acondicionados. En la actualidad existen muchos tipos de ventiladores en miniatura que puedes conectar a la computadora. ¡Sería fantástico si pudiéramos encender el ventilador automáticamente o cambiar su velocidad según la temperatura! Primero, conectaremos el sensor de temperatura con el motor para poder medir la temperatura.

Ya utilizaste el sensor de temperatura en la semana 6. Vamos a repasar los conceptos del sensor de temperatura (*TMP36*). *TMP36* es el sensor de temperatura más comúnmente utilizado y detecta un rango de temperatura de -40 °C a 125 °C. Posee tres patas. Visto desde el frente, donde aparece el nombre del modelo, las tres patas corresponden a la entrada de voltaje, la salida de voltaje y la descarga a tierra, de izquierda a derecha. Repasa los materiales de la semana 6 para ver más detalles.

Objetivos del proyecto

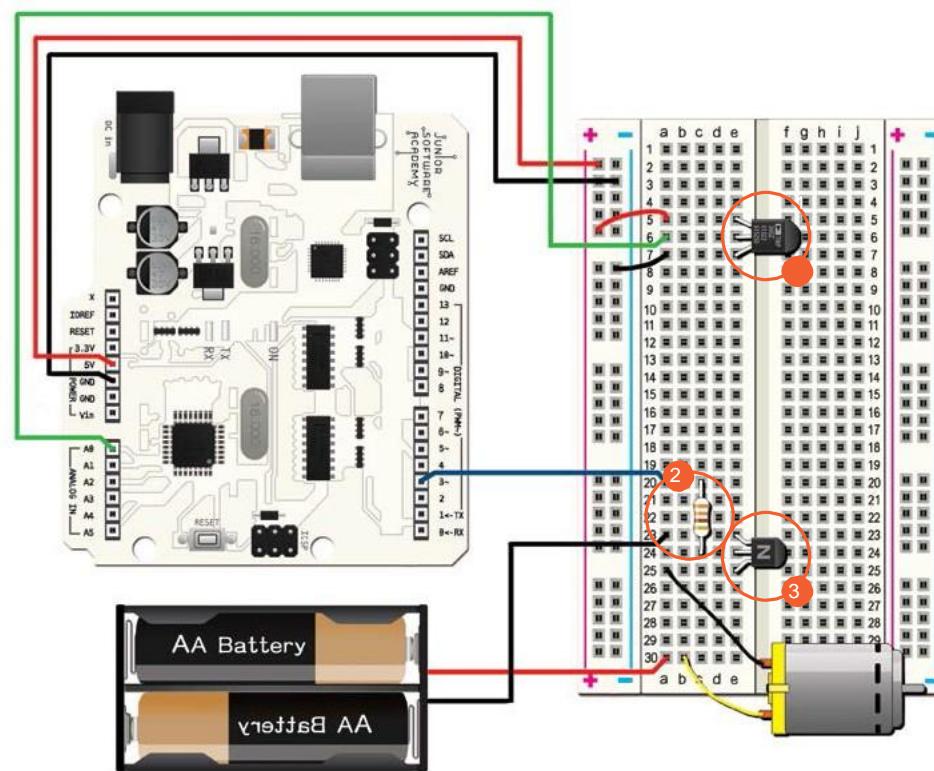
Activar un motor que mida la temperatura y se encienda cuando la temperatura alcance cierto nivel.

Expresión del hardware

Deberás tener precaución al utilizar el sensor de temperatura en proyectos que conecten componentes a la placa de creación. Ya aprendimos esto en la semana 7, pero lo haremos otra vez.

1 Debes tener en cuenta la polaridad al conectar el sensor de temperatura. Si lo conectas de manera inversa, la placa de pruebas a la que conectaste el sensor podría calentarse demasiado y derretirse.

2 Es por eso que nunca debes tocar la placa de pruebas cuando está energizada, incluso si el valor de la temperatura no se mide adecuadamente. Para poder tocar la placa, deberás desconectar el cable USB y la alimentación de energía y esperar a que esté fría.



- 1 Sensor de temperatura
- 2 Resistencia de 330 Ω
- 3 Transistor

Pensar en código

```

01 int motorPin = 3; ..... o
02 int tempPin = 0; ..... o
03 float V, C;
04
05 void setup() {
06   Serial.begin(9600); ..... o
07   pinMode(motorPin, OUTPUT) ..... o
08   ; while(!Serial); ..... o
09 }
10
11 void loop() {
12   V = (analogRead(tempPin) * 5.0 / 1024.0); ..... o
13
14   C = (V - 0.5) * 100; ..... o
15
16   Serial.print(" C : "); ..... o
17   Serial.println(C); ..... o
18
19   if(C >= 20) {
20     digitalWrite(motorPin, 200);
21   }
22   else if(C >= 10) {
23     digitalWrite(motorPin, 10
24     0);
25   }
26   else {
27     digitalWrite(motorPin, 0);
28   }
29   delay(1000);
}

```

Fija el pin 3 para conectar el motor.

Fija el pin 0 para conectar el sensor de temperatura.

Inicialización de serie.

Fija el pin 3 en modo de salida.

Verifica si el serie está configurado adecuadamente.

Cambia el valor de lectura a un voltaje de 0 a 5 V.

Obtiene la temperatura utilizando el valor del voltaje

Envía "C:" como serie de salida.

Envía la temperatura como serie de salida

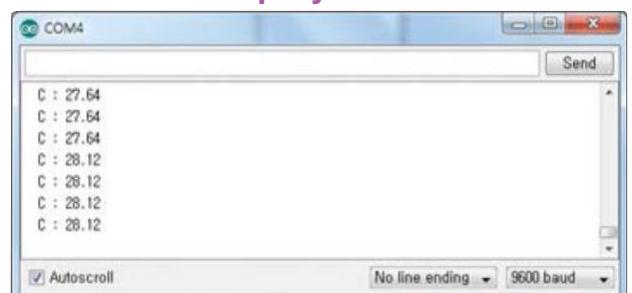
Conecta el motor al pin 3 e inserta el sensor de temperatura al pin 0. Cambia el valor recibido desde el sensor de temperatura a un valor de voltaje de 0 a 5 V y envía la serie de salida y activa el ventilador según la temperatura. Si la temperatura supera los 20 °C, el motor gira con una potencia de 200. Si la temperatura es superior a los 10 °C e inferior a los 20 °C, el motor gira con una potencia de 100. El motor se detiene si la temperatura es inferior a 10 °C. Repite este proceso cada segundo.

Analizaremos cómo el valor de entrada del sensor de temperatura se transforma en un valor de voltaje.

```
.voltaje(V) = 5.0 / 1024.0  
           * analogRead()  
  
.temperatura(C) = 100 * (V - 0.5)
```

El TMP36 utiliza la función *analogRead()* para convertir el valor analógico de voltaje (0 a 5 V) en valores del rango de 0 a 1023 y lee dichos valores. 0 V equivale a 0 y 5 V equivalen a 1023, por lo que si calculas la fórmula de voltaje que aparece en el gráfico, podrás obtener el valor del voltaje.

Resultados del proyecto



Puedes aplicar este proyecto a los siguientes ejemplos:

- Construir un motor que responda a la luz con el sensor de luz.
- Construir un motor que se active únicamente cuando una persona se pare en frente utilizando un sensor infrarrojo.

REVISIÓN

Reaso

Piensa en la parte que más te costó y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.

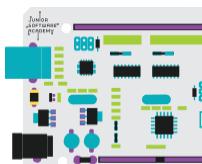
SEMAN
A
12

Servomotor

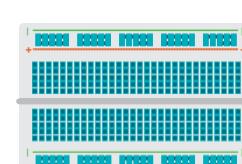
Objetivos de

- Controlar servomotores utilizando la salida PWM de la placa de creación.
- Controlar la dirección de los servomotores utilizando el potenciómetro.

Materiales para



Placa de creación



Placa de pruebas



Cable USB



Servomotor



Cable

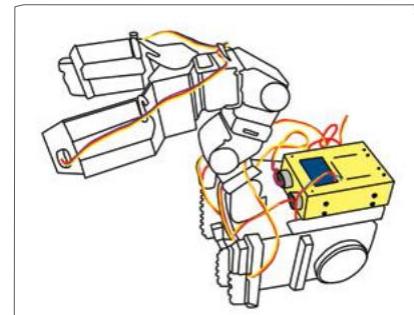


Potenciómetro


LECCIÓN

¿Qué es un servomotor?

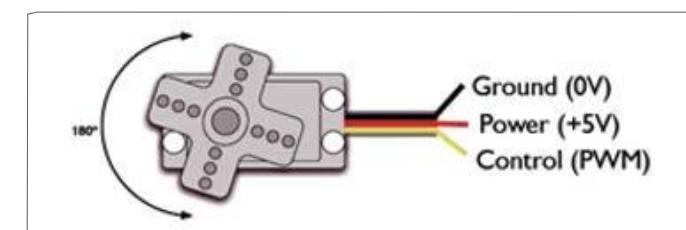
Un servomotor es diferente de un motor CC. El servomotor tiene un rango de rotación de 0 a 179 en lugar de un giro completo. Por lo tanto, es muy útil para controlar movimientos físicos con precisión. Los servomotores cuentan con un controlador incorporado, por lo que son muy fáciles de controlar una vez conectados. Se utilizan con frecuencia para automóviles a control remoto, aviones



a control remoto y en robots humanoides. Durante esta semana activaremos la placa de servos de la placa de creación y aprenderemos a controlar el ángulo de rotación.

Las características de los servomotores son:

- Son útiles para ejercer un control preciso, ya que se mueven en una posición determinada en lugar de girar en círculos.



- El eje gira en un ángulo de 0 a 179.
- El servomotor se puede controlar de manera directa al calcular su señal. La respuesta del servomotor corresponde a la duración del ancho del pulso. Si el ancho de pulso es inferior a 1 milisegundo (ms), el servo gira hacia un extremo y si el ancho de pulso es igual o mayor que 2 ms, gira hacia el otro extremo.
- Se puede controlar fácilmente con la función de biblioteca de control de servos que provee el entorno de Arduino.

Práctica

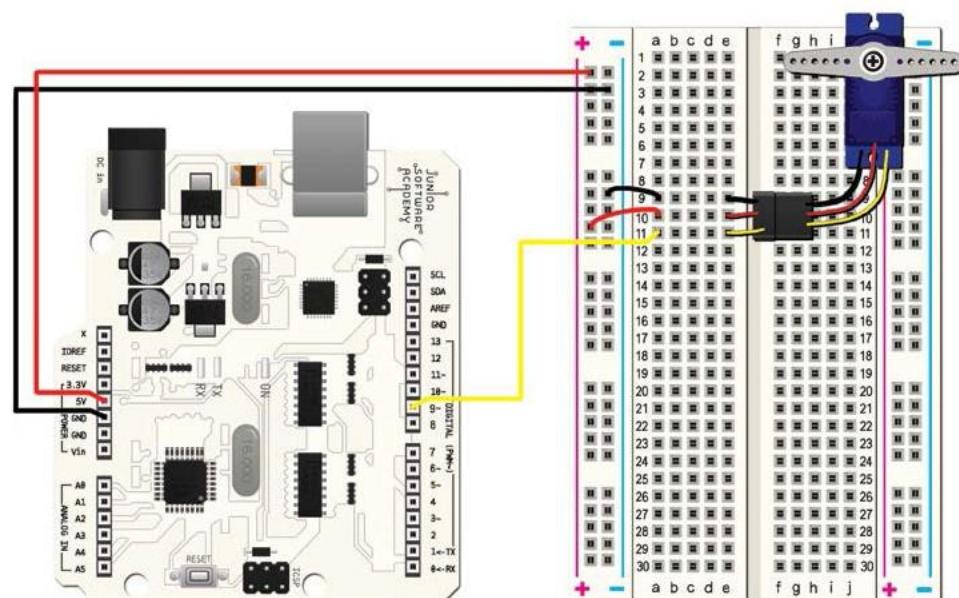
Cómo controlar el servomotor a voluntad

Objetivos del proyecto

Activar y controlar el servomotor.

Expresión del hardware

Conecta el terminal de alimentación de energía (rojo) al 5 V de Arduino, la conexión a tierra (negro o marrón) a GND de Arduino, y el terminal de control (amarillo o naranja) al pin digital 9 de Arduino.



Responde:

¿Qué pin digital del Arduino se conecta al terminal de control del servomotor?

El pin digital **9** se conecta al terminal de control del servomotor.

Pensar en código

```

01 #include <Servo.h>
02
03 Servo servo; ----- Crea un objeto para controlar el servomotor.
04 int motorPin = 9; ----- Fija el pin 9 para conectar el servomotor.
05 int angle; ----- Declara la variable angle para guardar el
06
07 void setup() { ----- ángulo.
08   Serial.begin(9600); ----- Inicialización de puerto serie.
09   servo.attach(servoPi ----- Conecta el objeto servomotor al pin 9.
10   n); ----- Espera a que la inicialización del puerto serie
11   while(!Serial); ----- esté completa. Envía "Servomotor" como seri
12   Serial.println("Servomotor Mortor"); ----- e de salida.
13 }
14
15
16 void loop() { ----- En un incremento de 0 a 179, gira el servomoto
17   for(angle = 0; angle < 180; angle++) { ----- r según el ángulo de salida.
18     servo.write(angle);
19     Serial.print("Servomotor Mortor angle :");
20     Serial.println(angle);
21     delay(15);
22   }
23
24
25   for(angle = 179; angle >= 0; angle--) {
26     servo.write(angle);
27     Serial.print("Servomotor Mortor angle :");
28     Serial.println(angle);
29     delay(15);
30   }
31 }
```

Crea un objeto para controlar el servomotor.
Fija el pin 9 para conectar el servomotor.
Declara la variable *angle* para guardar el
ángulo.

Inicialización de puerto serie.
Conecta el objeto servomotor al pin 9.
Espera a que la inicialización del puerto serie
esté completa. Envía "Servomotor" como seri
e de salida.

En un incremento de 0 a 179, gira el servomoto
r según el ángulo de salida.

En un decremento de 179 a 0, gira el servomot
or según el ángulo de salida.

Conecta el servomotor al pin 9 e inicia la serie. Aumenta de 0 a 179 por 1 cada
0,015 segundos y envía el ángulo modificado en el monitor serie. Una vez terminado,
deja el primer bucle *for* e ingresa el segundo bucle *for*. Para el segundo bucle *for*,
reduce de 170 a 0 cada 0,015 segundos y envía el ángulo modificado en el monitor
serie.

Recuerda que la clase *Servo* controla al servomotor y utilízala cuando sea necesario.

Resultados del proyecto

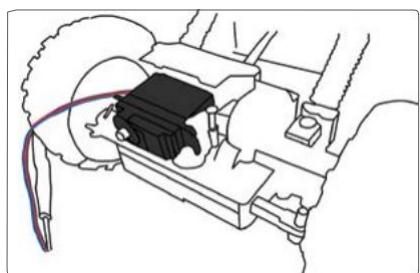
Luego de subir el *sketch*, el servomotor que está conectado a la placa de creación repite la rotación hacia la izquierda y hacia la derecha dentro del rango de 180 grados. Puedes verificar el ángulo de rotación al hacer clic en el botón “monitor serie” del lado derecho de la barra de herramientas.



Práctica

Crear un volante de automóvil a control remoto

Muchos automóviles a control remoto utilizan servomotores en las ruedas delanteras para cambiar de dirección. Luego, el automóvil se controla al girar el volante del vehículo. Teniendo en mente este método de control, intenta controlar el ángulo de rotación del motor al girar la perilla del potenciómetro a modo de volante.

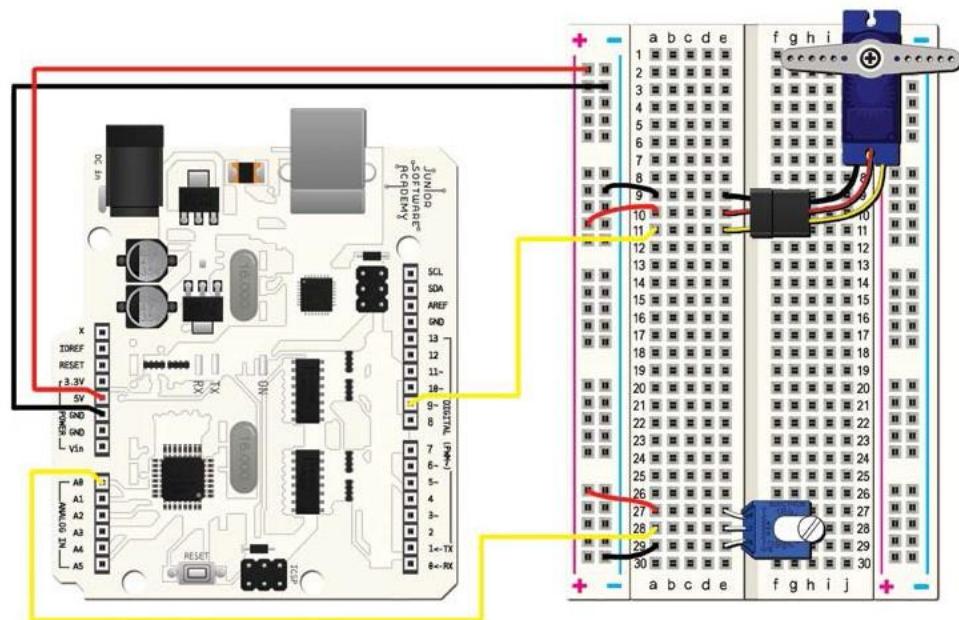


Objetivos del proyecto

Controlar el ángulo de rotación del servomotor utilizando la perilla del potenciómetro.

Expresión del hardware

Primero, construye un circuito igual al que utilizamos para controlar el servomotor. Luego, construye el circuito para el potenciómetro. Al girar la perilla del potenciómetro, el voltaje del pin del potenciómetro cambiará a un valor dentro del rango de 0 a 5 V. Cuando la placa de creación transforme el valor analógico a un valor digital, el valor tendrá un rango de 0 a 1023. Los valores de 0 a 1023 se convierten a un rango de 0 a 179 grados y la posición del servomotor cambiará de manera correspondiente.



Pensar en código

```

01 #include <Servo.h>
02
03 Servo servo;-----o Crea un objeto para controlar el servomotor.
04 int potenPin = 0;-----o Fija el pin 0 para conectar el potenciómetro.
05 int servoPin = 9;-----o Fija el pin 9 para conectar el servomotor. Decla-
06 int angle;-----o ra la variable angle para guardar el ángulo.
07 int sensorVal;-----o Declara la variable sensorVal para guardar los
08 -----o valores recibidos desde el potenciómetro.
09 void setup() {
10     Serial.begin(9600);
11     servo.attach(servoPin);-----o Inicialización de serie.
12     while(!Serial);-----o Conecta el objeto "servomotor" al pin 9.
13     Serial.println("Servor Mortor");-----o Espera a que la inicialización del puerto
14 }-----o serie esté completa. Envía "Servomotor"
15 -----o como serie de salida.
16
17 void loop() {
18     sensorVal = analogRead(potenPin);
19     angle = map(sensorVal, 0, 1023, 0, 179) ;-----o Lee y obtiene el valor del potenciómetro.
20     ; _ Serial.print("Servor Mortor angle : "); _ Serial.println(angle);
21     servo.write(angle);-----o Crea un mapa del servomotor con cada
22     delay(100);-----o rango de ángulo.
23 }

```

Lee el valor de resistencia del potenciómetro en la función *analogRead()* y convierte los ángulos de 0 a 179 grados que admite el servomotor. El valor que se lee con la función *analogRead()* oscila entre 0 y 1023, pero esto no coincide con el rango que admite el servomotor. Esta conversión se conoce como “mapeo”. Envía el mapa de valores y lo aplica a los grados del servomotor. Este proceso se repite cada 0,1 segundos.

Resultados del proyecto



Puedes aplicar este proyecto a los siguientes ejemplos:

- Abrir puertas automáticamente al detectar que un objeto se acerca.
- Tocar música con varios servomotores y un xilófono.

REVISIÓN Repaso

Piensa en la parte que más te costó y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.



SEMANA
13



Proyecto



DISEÑO CREATIVO 03

Hablemos con movimientos



Objetivos de aprendizaje

Crear un objeto que se mueva aplicando los conceptos que has aprendido hasta la semana 12.

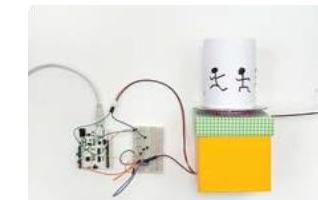


Materiales para preparar

Kit de creación, todo tipo de motores, varios objetos (disco, rueda de acrílico, papel, móvil, aletas y otros objetos que sería divertido mover), herramientas para manualidades.

Estroboscopio

Presentamos un estroboscopio que puedes fabricar fácilmente y explicamos cada paso del proceso.

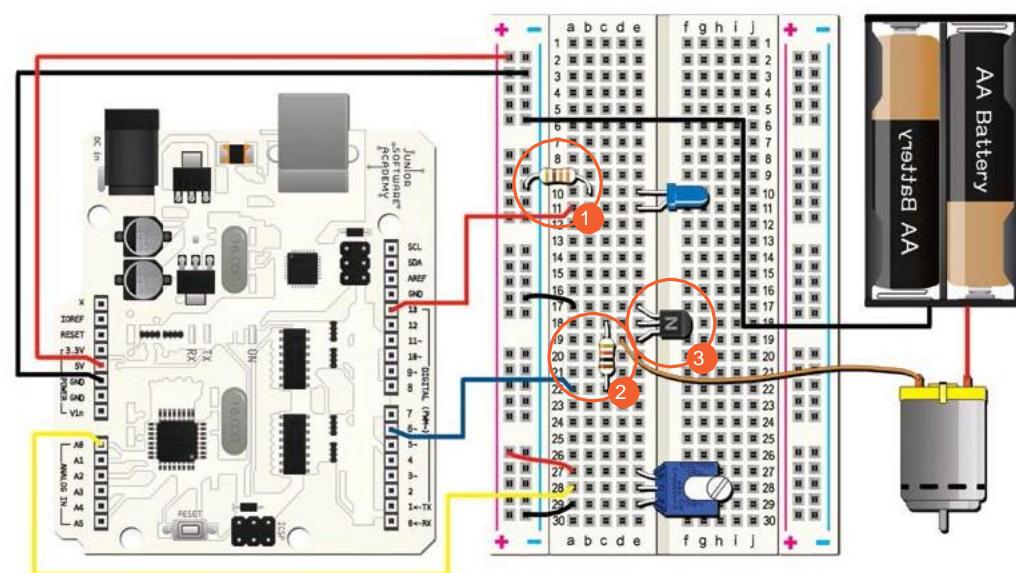


Objetivos del proyecto

Crear un estroboscopio para crear movimientos con imágenes estáticas. Consulta las imágenes del proyecto terminado a la derecha y lee las instrucciones que aparecen debajo.

- 1 Inserta una pequeña rueda de acrílico al eje del motor que incluye el kit.
- 2 Coloca un CD y fíjalo al extremo superior de la rueda de acrílico con cinta.
- 3 Dibuja una secuencia de movimiento en un papel rectangular
- 4 Dobra cada viñeta para crear un tubo cilíndrico y fíjalo sobre el CD.
- 5 Fija ledes apuntados hacia las imágenes, ajusta la altura de cada led.
- 6 Consulta el proyecto terminado y la expresión del hardware para conectarlo a la placa de creación.

Expresión del hardware



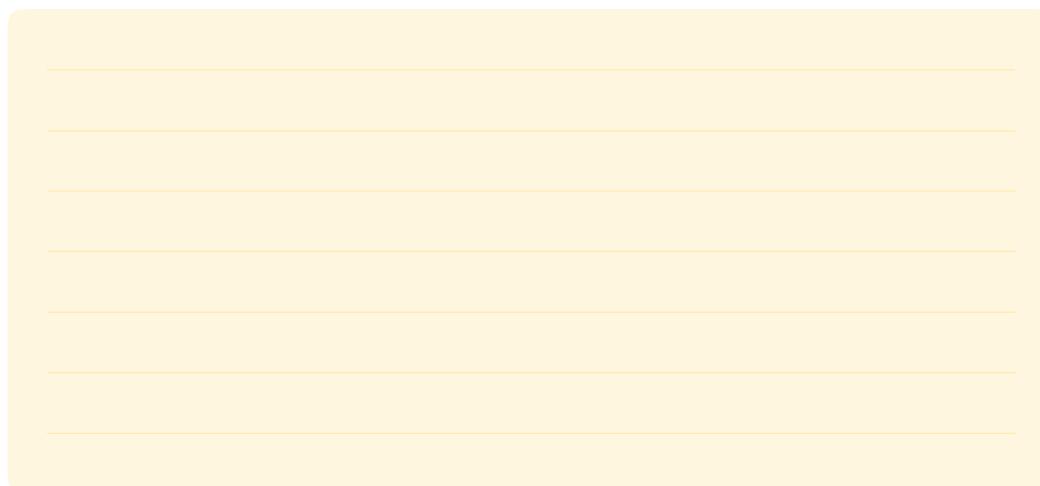
1 Resistencia de
330 Ω



2 Resistencia de
1 K Ω



3 Transistor



Pensar en código

```

01 int ledPin = 13;
02 int motorPin = 6;
03
04 int potenPin = 0;
05 int potenVal = 0;
06
07 void setup() {
08   pinMode(ledPin, OUTPUT);
09   pinMode(motorPin, OUTPUT);
10 }
11
12 void loop() {
13   potenVal = analogRead(potenPin);
14   potenVal = map(potenVal, 0, 1023, 0, 20);
15
16   if(potenVal <= 0) {
17     digitalWrite(ledPin, LOW);
18     digitalWrite(motorPin, LOW);
19   }
20   else {
21     digitalWrite(motorPin, HIGH);
22     digitalWrite(ledPin, HIGH);
23     delay(potenVal * 0.7);
24     digitalWrite(ledPin, LOW);
25     delay(potenVal * 25);
26   }
27 }
28 }
```

Fija el pin 13 para conectar los ledes. Fija el pin 6 para conectar al motor.

Fija el pin analógico 0 para conectar el potenciómetro. Declara la variable *potenVal* para guardar el valor del estado del potenciómetro. Fija *ledPin* en modo de salida. Fija *motorPin* en modo de salida.

Lee el valor del potenciómetro.

Si el valor del potenciómetro es igual o menor que 0 , detiene la activación de los ledes y el motor.

Si el valor del potenciómetro es mayor que 0, inicia la activación de los ledes y el motor.

- ➊ El valor del potenciómetro recibido desde la función *analogRead()* se guarda en la variable *potenVal*. Sin embargo, los valores del rango de 0 a 1023 no son adecuados para los ledes intermitentes, por lo que deberás ajustar el valor al rango de 0 a 20 utilizando la función *map()*.
- ➋ Detiene el parpadeo de los ledes y el funcionamiento del motor si el valor del potenciómetro es igual o menor que 0.
- ➌ Activa los ledes y el motor si el valor del potenciómetro es mayor que 0 y ajusta la velocidad del parpadeo de los ledes según el valor del potenciómetro.



- Al controlar la velocidad del parpadeo de los ledes con el potenciómetro, crea la ilusión de que las imágenes se mueven.
- Cuando el ciclo de rotación del motor coincide con la velocidad de parpadeo de los ledes, se ve como si la imagen se moviera en el lugar. Intenta perfeccionar la animación al ajustar la velocidad de la función `delay()` con precisión.

Proyecto CREATIVO

Hablemos con movimientos

¿Qué tal si creamos nuestro propio proyecto?

Ya creaste tu propio proyecto con lo que aprendiste hasta la semana 4. Ahora es momento de hacer algo increíble con todo lo que aprendiste hasta la semana 8.

1 Formar grupos

Siéntense en grupos de 4 a 6. Piensen en una actividad grupal en cinco minutos. Escriban algunos *sketches* mientras piensan en distintas ideas. Cada uno deberá exponer sus ideas y qué roles tendrán sus compañeros durante uno o dos minutos. Los compañeros del grupo no deberán juzgar si el proyecto es bueno o malo. En cambio, deberán brindar críticas constructivas que puedan ayudar a desarrollar las ideas de cada

A large, light-yellow rectangular area with horizontal ruling lines, intended for students to draw their sketches.

2 Resumir ideas

Crea un breve resumen de tus ideas con imágenes y textos. Luego, piensa en un *sketch*, la composición del hardware, un circuito y describe en detalle lo que harás y que acciones realizarás.

**3 Conectar el circuito**

Utilizando lo que aprendiste, crea un proyecto que utilice los sensores y motores. Podrías tener que unir dos cables para crear un cable más largo. Una forma de crear un circuito más resistente es soldar los terminales. Sin embargo, estamos trabajando sin soldar, por lo que deberás unir y retorcer los cables para conectarlos. También puedes fijarlos con una pistola de pegamento luego de unir los cables con los componentes. También aprendimos a conectar cables con cinta de cobre.

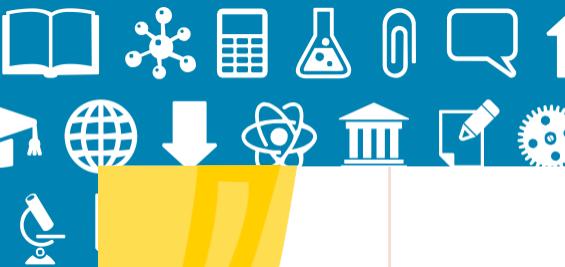
4 Presentar

Como parte final de esta clase, nombra tu proyecto y haz una presentación acerca de tu creación. Durante la presentación deberás mencionar el nombre del proyecto y la descripción (por qué lo creaste; cómo; cómo funciona) y escucha las opiniones de tus compañeros sobre tu proyecto.

REVISIÓN**Reaso**

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.



Clase final

14

PROTOTIPOS
DIGITALES
INTERACTIVOS

Crear mi propio proyecto

Objetivos
de aprendizaje

Crearás algo en función de los conceptos que aprendiste este semestre. Elegirás un tema y, durante las siguientes dos semanas, construirás circuitos, ensamblarás el hardware y escribirás los *sketches*. Luego presentarás tu proyecto y realizaremos una muestra.

Crearás algo basado en los materiales vistos en clase este semestre.

Desarrollarás ideas, construirás circuitos, ensamblarás el hardware y escribirás el *sketch* de código.

Materiales
para preparar

Baterías, lápiz, notas autoadhesivas y otros materiales para tu proyecto

Proyecto
CREATIVO

Resumen de lo que aprendiste este semestre

Durante las siguientes dos semanas, tendrás tiempo para crear tu propio proyecto, basándote en lo que has aprendido hasta ahora. Un proyecto de tu propia creación sería muy importante, ¿no?

Repasa lo que hemos aprendido hasta ahora y analiza los contenidos. Piensa en algo que querrías crear.

Elegir un tema

¿Qué idea se te ocurre? Ya que es el último proyecto, ¿qué tal si creamos un proyecto sobre un mismo tema? Sería bueno encontrar un tema común, analizar diversas maneras de abordarlo y expresar ese tema, así como también analizar las distintas funciones y métodos. Hablaremos sobre el tema que deseas tratar. Comenzaremos con una lluvia de ideas.

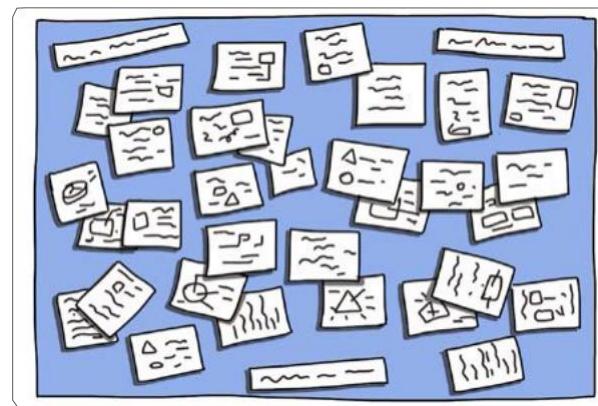
- 1 Peguen una hoja de papel por cada diez estudiantes en la pared.
- 2 Repartan notas autoadhesivas y comparten.
- 3 Luego de un breve análisis, seleccionen temas generales y escríbanlos en las hojas. Pueden utilizar los temas que aparecen debajo o pedir ayuda al maestro. Intenten seleccionar un tema que sea interesante para la mayoría de los estudiantes
 - Obsequio: haz un obsequio para tu familia o amigos
 - Juego: crea varios controladores de juego con Arduino. Puedes utilizar un programa llamado Scratch, es muy fácil de usar.
 - Juegos populares: transforma un juego regional a su versión digital y dale una nueva perspectiva.
 - Ropa de moda: modifica bolsos, zapatos y gorros para transformarlos en prendas divertidas.
 - Automatizar tu habitación: diseña y construye un sistema automatizado para tu propia habitación.

4

En las notas autoadhesivas, escriban ideas interesantes que se les ocurran, incluso si no están relacionadas con el tema, e ideas acerca de lo que desean hacer en relación con el tema y péguenlas en la hoja. Mientras más ideas haya, mejor.

5 Organicen las notas que tengan ideas similares. Si es necesario, pidan ayuda al maestro.

Lluvia de ideas
<ul style="list-style-type: none"> . Mientras más ideas, mejor. . No critiques la opinión de los demás. . Apoya las ideas que sean buenas y divertidas. . Aporta nuevas ideas.



6 Analicen el tema elegido y seleccionen el tema más interesante. Si escribieron ideas que corresponden al tema seleccionado, deberán realizar una presentación sobre dicho tema para utilizarlas en el proyecto.

Producción del proyecto

Organiza tus ideas acerca del tema. Diseña los circuitos y los *sketches*. Crea tu propio proyecto luego de preparar los materiales y componentes necesarios.

La primera clase es para pensar en ideas, por lo que tal vez no cuenten con todos los materiales necesarios. Enfóquense en diseñar un circuito; escribir un *sketch*; y realizar las acciones que desean.

Organiza tus ideas acerca del tema. Diseña los circuitos y los *sketches*. Crea tu propio proyecto luego de preparar los materiales y componentes necesarios.

La primera clase es para pensar en ideas, por lo que tal vez no cuenten con todos los materiales necesarios. Enfóquense en diseñar un circuito; escribir un *sketch*; y realizar las acciones que desean.

Presentación y exhibición

1 Desplieguen su proyecto en un escritorio. Coloquen el panel con su nombre, el título del proyecto, una breve introducción y el proyecto en sí.



2 Tomen turnos para realizar sus presentaciones, incluyendo una demostración de cinco minutos. Durante su presentación, mostrarán el nombre del proyecto, qué es, las partes que fueron fáciles, los aspectos difíciles y las áreas que desearían mejorar. Si su proyecto final resultó distinto de cómo lo planificaron, pidan consejos a sus compañeros. Luego de cada presentación y exhibición, analicen las opiniones de sus compañeros. Si sus compañeros pensaron en una idea fantástica que no habían pensado, felícítenlos. Podrían divertirse durante el proyecto, pero será más divertido compartir la diversión con sus compañeros.

3 Cuando todos hayan terminado sus presentaciones, tomen un momento para apreciar el trabajo de sus compañeros.

Conclusión

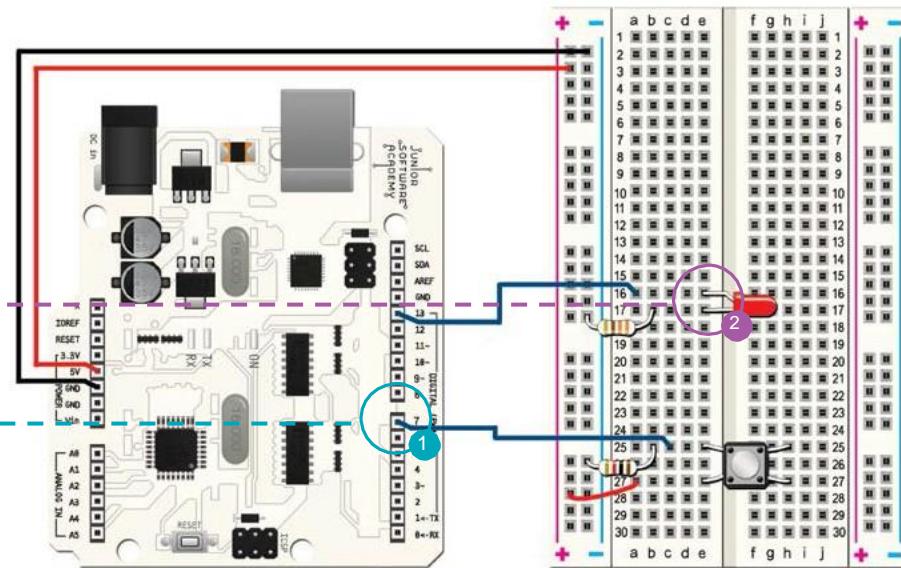
REVISIÓN → Repaso

- 1 Evalúa tu compromiso o esfuerzo para trabajar en los proyectos de esta clase. Tu maestro habrá tomado fotografías de la clase. Observa las fotografías escuchando música por 2 a 3 minutos.
- 2 Habla sobre lo que sentiste durante el último proyecto o durante el semestre.
- 3 Concluyan la clase con una ronda de aplausos para ustedes y todos los que se esforzaron este semestre.

Piensa en tu parte favorita de las actividades de hoy y compártela con tus compañeros.

Piensa en la parte que más te costó y compártela con tus compañeros.

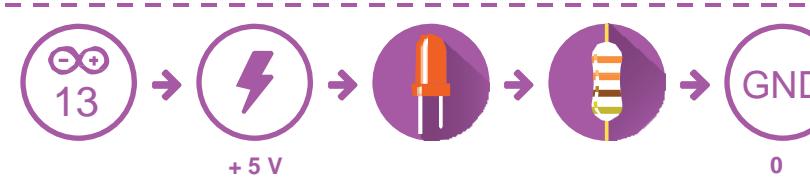
Un vistazo a la placa de creación



1
El suministro de energía del pin 7 conectado es de 0 V cuando el interruptor no está presionado. La corriente fluye al pin 7 y la energía aumenta a 5 V al presionar el interruptor.



Los ledes conectados se encienden cuando la corriente fluye al pin 13 de la placa Arduino. La pata más larga de cada led debe estar conectada al terminal positivo. Para proteger el circuito de ledes, la corriente dirigida hacia 0 V (tierra) debe pasar por una resistencia.



```
int led = 13;
int swPin = 7;
int swVal = 0;
```

- Declara 13 como la salida digital de Arduino.
- Declara la entrada digital 7 para el pin conectado al interruptor.

```
void setup()
{
    pinMode(led, OUTPUT);
    pinMode(swPin, INPUT);
}
```

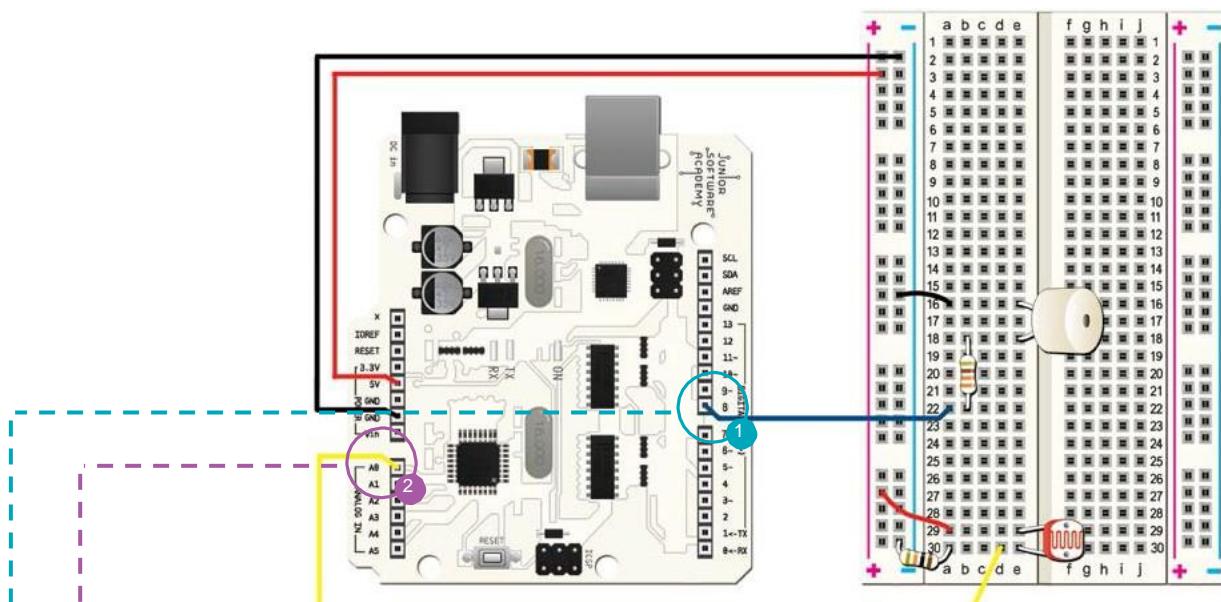
- Declara el pin 13 conectado al led como salida
- Declara el pin 7 conectado al interruptor como entrada.

```
void loop()
{
    swVal = digitalRead(swPin);
    if (swVal == HIGH) {
        digitalWrite(led, HIGH);
    } else {
        digitalWrite(led, LOW);
    }
}
```

- Al pulsar el interruptor, el valor del pin se vuelve mayor que 0 (HIGH).
- Envía corriente al pin declarado en (). En este caso, es el pin 13 y los ledes conectados se iluminan.

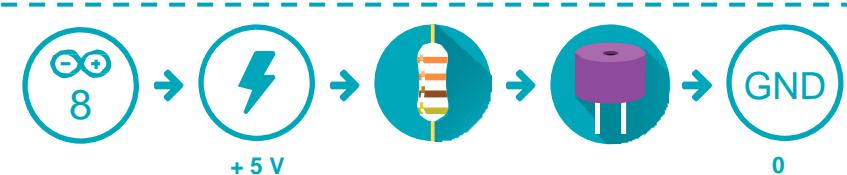


Un vistazo a la placa de creación



1

Conecta el zumbador al pin digital 8.
El sonido varía según la frecuencia que envía la función `tone()`. Designa la frecuencia para cada nota de antemano.



2

Los valores detectados por el sensor de luz se ingresan en el pin analógico 0.
La función `analogRead()` lee estos valores y los utiliza.
Los valores registrados por el sensor analógico deben estar conectados al pin analógico de Arduino.



```
int note_c4 = 262; int note_d4 = 294; int note_e4 = 330;
Int speakerSensor = 8; Int noteDuration = 100; Int cdsPin = 0;
```

```
void setup()
{
    pinMode(speakerSensor, OUTP
    UT); Serial.begin(9600);
}
```

```
void loop()
{
    cdsValue = analogRead(cdsPin);
    if(cdsValue < 200) {
        noteValue = note_c4;
    } else if(cdsValue > 200 && cdsValue < 230) {
        noteValue = note_d4;
    } else if(cdsValue > 230 && cdsValue < 260) {
        Serial.println(cdsValue);
        tone(speakerSensor, noteValue, noteDuration);
        delay(10);
    }
}
```

Declara un nombre fácil de entender para la variable de las notas y declara los valores correspondientes.

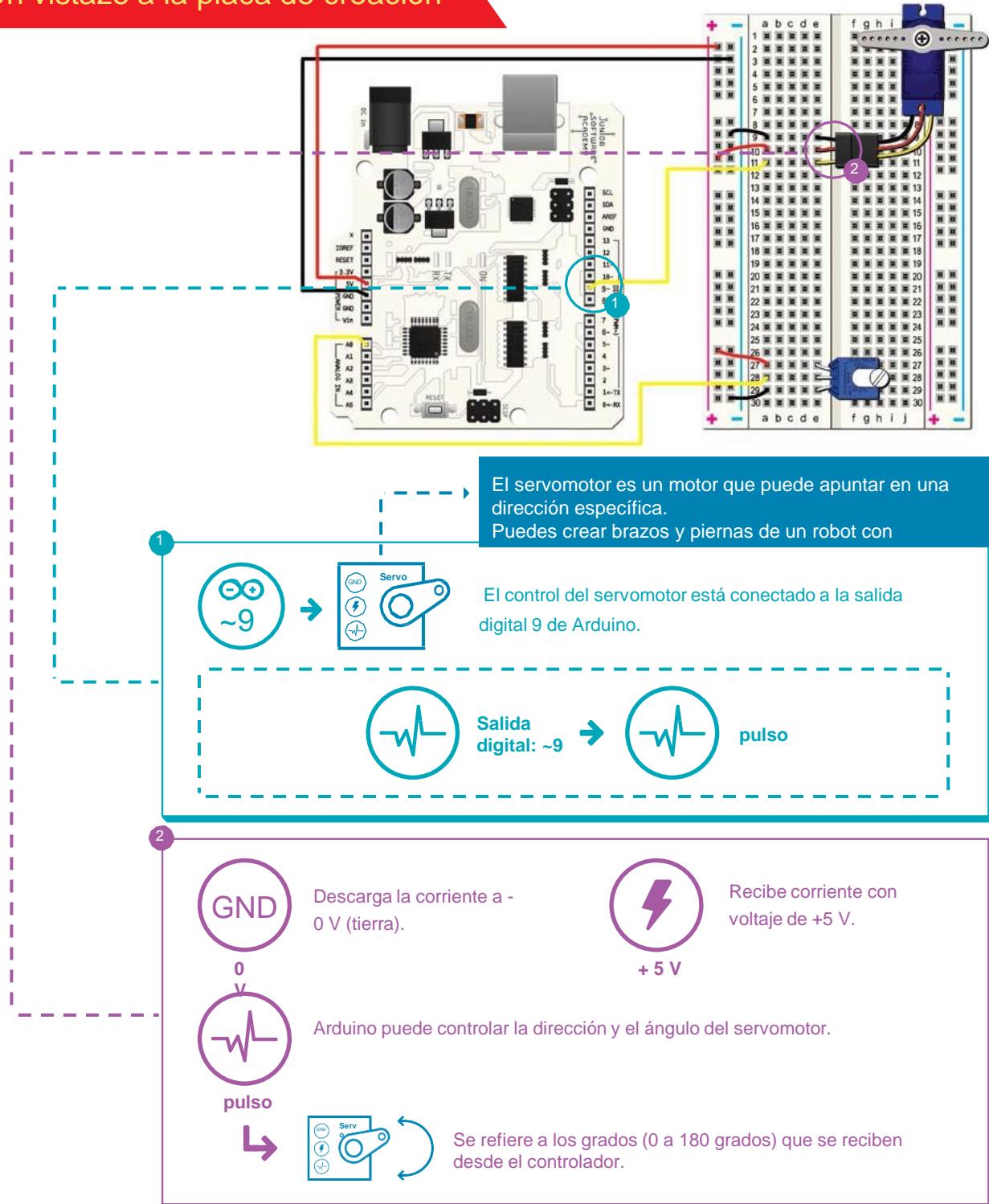
Declara el zumbador como salida digital 8 de Arduino. Fija la duración de cada nota. 100 significa 0,1 segundos.

Lee los valores del sensor de luz.

Decide las notas según los valores del sensor de luz. La frecuencia declarada se designa como variable para las notas.

El valor de la frecuencia designada en el pin del valor declarado dentro de () se emite en forma de sonido. Desde el zumbador conectado al pin 8, el sonido de la frecuencia designada se emite por 0,1 segundos.

Un vistazo a la placa de creación



```
#include <Servo.h>

Servo servo;
int poten_pin = 0;
int servo_pin = 9; // Declara la salida digital 9 de Arduino como el pin para el servomotor.

void setup()
{
    Serial.begin(9600);
    servo.attach(servo_pin); // Conecta el control del servomotor con la entrada digital 9 de Arduino.
    while(!Serial);
    Serial.println("Servo Mortor");
}

void loop()
{
    sense_value = analogRead(poten_pin);
    angle = map(sense_value, 0 , 1023, 0 , 179); Serial.print("Servo Mortor angle : "); Serial.println(angle);
    servo.write(angle); // El servomotor apuntará al ángulo determinado por el valor entre paréntesis.
    delay(100);
}
```



Enabling People

Education for Future Generations

©2019 SAMSUNG. Todos los derechos reservados.

Samsung Electronics Corporate Citizenship Office es el único titular de los derechos de autor de este manual.

Este manual es propiedad literaria protegida por las leyes de derechos de autor, queda prohibida cualquier reimpresión o copia no autorizada. Se debe obtener la autorización por escrito del titular de los derechos de autor para utilizar este manual para cualquier uso distinto del plan de estudios de Samsung Innovation Campus o para utilizar la totalidad o parte de este manual.