

Rover Educativo



SCT
SECRETARÍA DE
COMUNICACIONES
Y TRANSPORTES



AEM

AGENCIA
ESPACIAL
MEXICANA

Hacia el Espacio 

Agencia Espacial Mexicana

Dr. Javier Mendieta Jimenez
DIRECTOR GENERAL

M. en C. Carlos Duarte Muñoz
COORDINADOR GENERAL DE FORMACION
DE CAPITAL HUMANO EN EL CAMPO ESPACIAL

Ing. Mario Manuel Arreola Santander
DIRECTOR DE DIVULGACIÓN

Lic. Rodrigo Pérez González
GERENTE DE E-APRENDIZAJE Y DE E-DIVULGACIÓN

Ing. Aldo Daniel Acosta Durán
TÉCNICO EN FORMACIÓN DE CAPITAL HUMANO

Brenda Urbano Reyes
Cynthia Michelle Diez Martínez Paz
Omar Eduardo Rodríguez Romero
SERVICIO SOCIAL

Ciudad de México, 2018

Contenido

Introducción	5
Conceptos y Componentes Básicos.....	7
Antes de comenzar.	15
Instalación de la placa Arduino	15
Ejecutar ArduBlock.	16
Mapa de Arduino UNO R3.....	17
0 – Circuito Básico.	18
Introducción.....	18
Materiales.....	18
Arma un circuito básico	19
Arma un circuito en serie.	21
Arma un circuito en paralelo.	23
1 – Hola Mundo	25
Introducción.....	25
Materiales.....	25
Circuito	26
Programa	26
Resultados.	30
2 Sensor de luz LDR.	31
Introducción.....	31
Materiales.....	31
Circuito.	32
Programa	32
Resultados	34
3 Sensor de Temperatura	35
Introducción.....	35
Materiales.....	35
Circuito	36
Programa	37
Resultados	39
4 Sensor Infrarrojo (IR).....	41

Materiales	41
Introducción	41
Circuito	42
Programa	43
Resultados	45
5 Sensor Ultrasónico.....	46
Introducción	46
Materiales	46
Circuito	47
Programa	49
Resultados	51
6 Sensor Hall	52
Introducción	52
Materiales	52
Circuito	53
Programa	54
Resultado.....	56
7 Motores.....	57
Introducción	57
Materiales	58
Circuito	59
Programa	60
Resultados	63
Proyecto: Rover educativo	64
Materiales	64
Estructura Rover	0
Ubicación de los dispositivos electrónicos	2
Sensor IR	3
Motorreductores y llantas	4
Pilas o Batería Externa.....	8
Leds Ultrabrillantes (LDR) y Soportes.....	9
Módulo Puente H (L298N)	11

Sensor Ultrasónico.....	12
Arduino y protoboard.....	13
Sensor de Efecto Hall.....	13
Sensor de temperatura LM35.....	14
Sensor LDR.....	15
Led Rojo (temperatura).....	15
Fuente de Energía del Rover.....	17
Conexiones.....	21
Puente H.....	23
Sensor IR.....	26
Conexión LDR y LED ultrabrillantes	28
Conexión Sensor de temperatura LM35	31
Conexión de sensor efecto Hall.....	34
Conexión del sensor ultrasónico.....	36
Diagrama General de Conexiones.....	39
Programa	41
Rover Educativo	46
Sugerencias y soluciones a posibles problemas	51
Pon a prueba tu conocimiento.....	52
Bibliografía recomendada.....	57

Introducción

Este proyecto es elaborado con el propósito de introducir a los niños, jóvenes y público en general a la electrónica básica proporcionando herramientas de bajo costo, que utilizan un proceso de soldadura mínimo y que además se pueden reutilizar en otras aplicaciones adicionales a las mostradas en este texto.

El manual está dividido en prácticas que le mostrarán al lector el funcionamiento de diversos sensores y actuadores cada uno por separado y luego los junta en un pequeño proyecto final.

El proyecto final está inspirado en los Rover exploradores, estos cargan consigo una gran variedad de instrumentos, algunos de ellos similares a los presentados en este manual.

¿Qué es un Rover?

La palabra *Rover* está asociada a las siglas en inglés *ROV (Remotely Operated Vehicle)*, en español: Vehículo Operado Remotamente.

Las sondas exploradoras en marte se pueden catalogar como vehículos semiautónomos operados desde Tierra.

Hitos en la historia de sondas exploradoras

- El primer Rover en aterrizar en un cuerpo celeste, en este caso la Luna, fue el Lunojod I, lanzado por la Unión Soviética en el año de 1970.
- El primero Rover que tocó suelo marciano fue el Sojourner, el cual fue parte de la misión Pathfinder de la NASA. Amartizó en el año de 1997.
- Los exploradores gemelos Spirit y Opportunity, de la misión Mars Exploration Rover, llegaron a Marte en el año 2003, donde el Rover Opportunity posee el récord de mayor distancia recorrida por un vehículo sobre ruedas fuera de la superficie del planeta tierra.
- La misión Mars Science Laboratory es famosa por el Rover Curiosity, el cual amartizó en el 2012.

- La Agencia Nacional Espacial China envió a la Luna a Yutu, un Rover que forma parte de la misión Change 3, el cual alunizó en el año 2013.
- La próxima misión a Marte más esperada es la Mars 2020, la cual cuenta con un Rover de similar tamaño a Curiosity pero aún más sofisticado.

Conceptos y Componentes Básicos

Electricidad

Es una forma de energía relacionada con el *flujo de electrones*. Esta energía es muy versátil, lo que le permite tener una gran variedad de aplicaciones, como el transporte, iluminación, comunicación, computación, entre otras.

Corriente eléctrica

Es el *flujo de electrones* a través de un circuito conductor cerrado, este fluye siempre del polo negativo al polo positivo de la fuente.

Voltaje

Es la medida de cuánta energía se encuentra involucrada en el *movimiento de un electrón* entre dos puntos en un circuito eléctrico.

Resistencia eléctrica

Es la *oposición al flujo de electrones* al moverse a través de un conductor o circuito eléctrico; esta oposición puede ser similar a la fricción. Los factores que influyen para que la resistencia sea mayor o menor, son la temperatura, longitud, diámetro y material.

Ley de Ohm

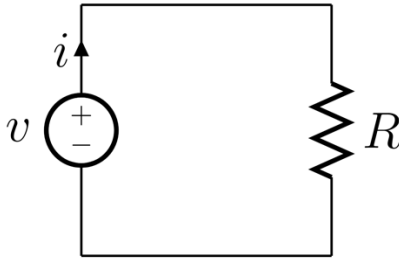
Esta ley establece una *relación* entre la *corriente*, el *voltaje* y la *resistencia* de un sistema eléctrico.

$$I = \frac{V}{R}$$

La ecuación indica que, a mayor voltaje en una resistencia, mayor es la corriente, y si es mayor el valor de la resistencia aplicada sobre el mismo voltaje, menor es la corriente.

Circuito eléctrico

Es una **red cerrada** de componentes eléctricos conectados entre sí, como baterías, resistencias, capacitores, etc. por la cual fluyen los electrones.

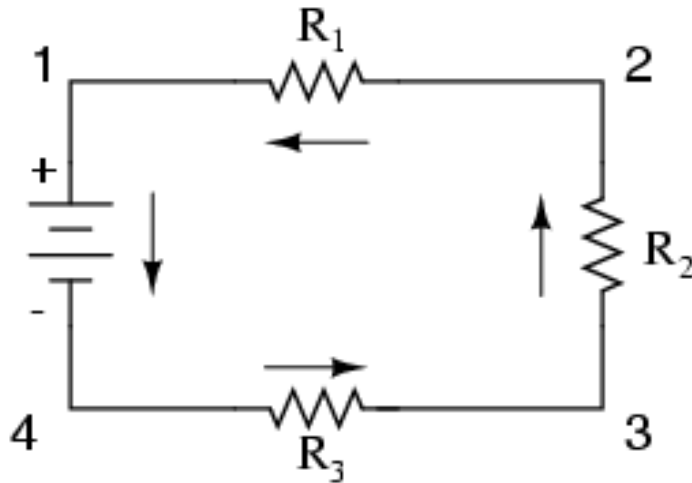


Ejemplo de un circuito eléctrico básico.

A la derecha la fuente de energía que puede ser una batería, y a la izquierda una resistencia que puede ser un foco.

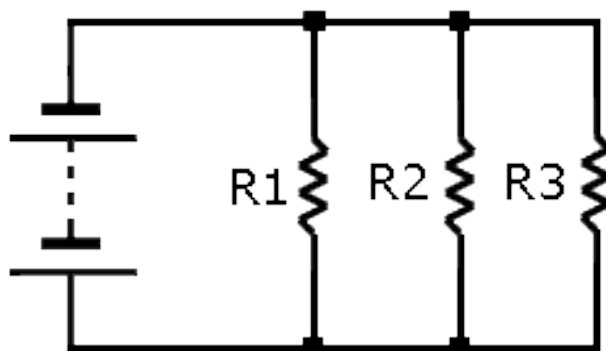
Circuito en serie

Es una configuración de circuito eléctrico en la que las terminales de los elementos eléctricos se conectan **sucesivamente**, es decir, la terminal de salida de un elemento se conecta a la terminal de entrada del siguiente elemento eléctrico.



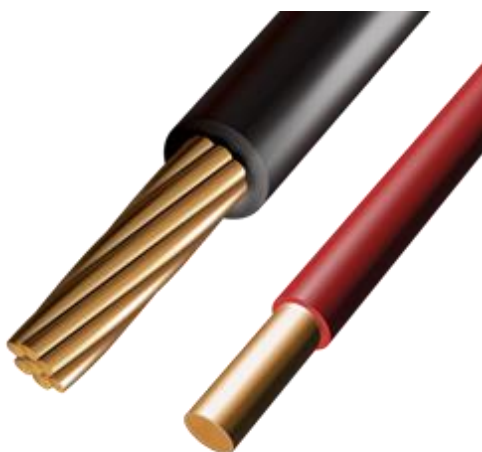
Circuito en paralelo

Es una configuración de circuito eléctrico en la que las terminales de los elementos eléctricos se conectan a un **punto en común**, es decir, las terminales de entrada de todos los elementos coinciden en el mismo punto y las terminales de salida coinciden en otro punto en común.

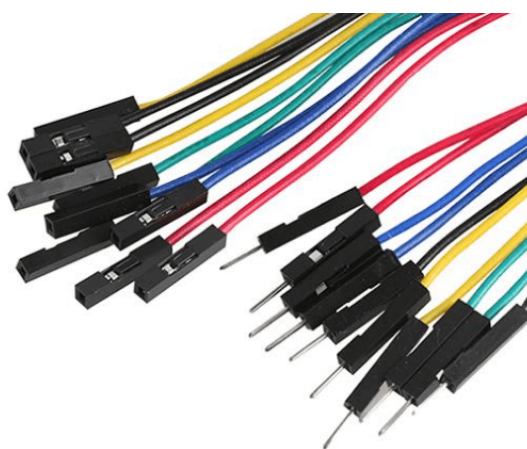


Alambre y cable

Son utilizados para **conducir** la corriente eléctrica. El cable está hecho de varios filamentos trenzados y el alambre es un solo filamento, ambos están protegidos por una cubierta aislante de plástico o goma. En este manual se emplearán **Alambres y Cables Dupont**, estos son cables con conectores en sus extremos, se ocuparán tipo Macho/Macho y Hembra/Macho



Negro: Cable ; Rojo: Alambre



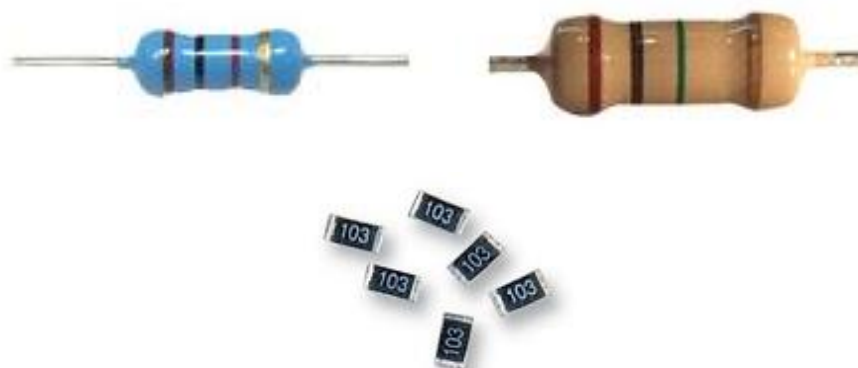
Cables Jumpers o Cables Dupont

Resistencias

Son elementos que añaden **resistencia** en un circuito eléctrico limitando el flujo de electrones. Se pueden encontrar de muchos tamaños y formas. En electrónica se ocupan códigos de colores para indicar el valor de la resistencia en Ohm [Ω].

<p> Código de Colores 0 1 2 3 4 5 6 7 8 9 0 Negro 1 Marrón 2 Rojo 3 Naranja 4 Amarillo 5 Verde 6 Azul 7 Púrpura 8 Gris 9 Blanco ±1% Marrón ±2% Rojo ±5% Dorado ±10% Plateado </p>	<p> Resistencias de 4 Bandas ±1% ±2% ±5% ±10% 1.5K 0 X1 1 1 X10 2 2 X100 3 3 X1000 4 4 X10000 5 5 X100000 6 6 X1000000 7 7 ÷10 8 8 ÷100 9 9 </p>	<p> Resistencias de 5 Bandas ±1% ±2% ±5% ±10% 15K 0 0 X1 1 1 1 X10 2 2 2 X100 3 3 3 X1000 4 4 4 X10000 5 5 5 ÷10 6 6 6 ÷100 7 7 7 8 8 8 9 9 9 </p>	<p> Resistencias de 6 Bandas ±1% 100 50 ±2% 25 15 ±5% 10 5 ±10% 1 PPM 620K 0 0 X1 1 1 1 X10 2 2 2 X100 3 3 3 X1000 4 4 4 X10000 5 5 5 ÷10 6 6 6 ÷100 7 7 7 8 8 8 9 9 9 </p>
---	--	--	---

Las resistencias de uso común en electrónica son hechas de recubrimiento de **carbón** (Café), recubrimiento de **metal** (Azul/verde) y en circuitos impresos unos "Chips" pequeños de color negro con un número llamadas **SMD**

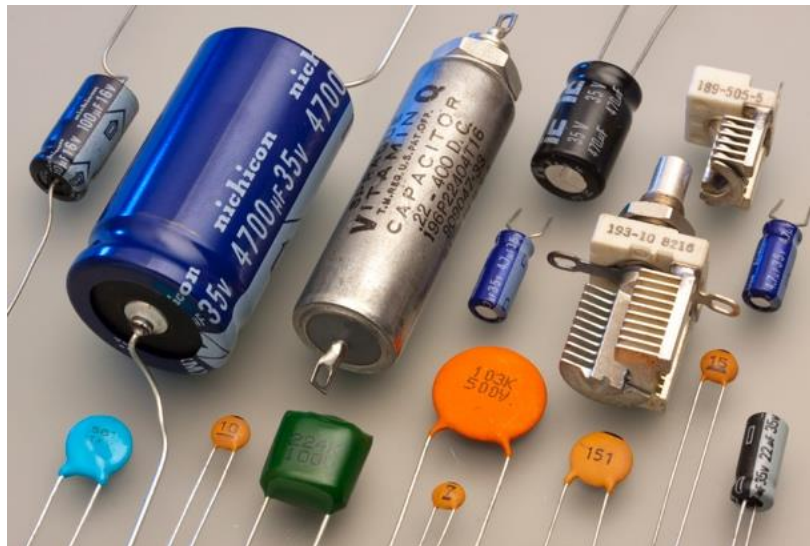


Capacitores

Almacena electrones para después liberarlos. A esta habilidad se le llama capacitancia.

Cuando se conecta un capacitor se debe verificar que cumpla o exceda la capacidad del voltaje que se empleará en el circuito eléctrico, de otra manera puede ser dañado por el exceso de carga almacenada.

Los capacitores se encuentran en diversas presentaciones, pero, los de uso más común son los capacitores *cerámicos* y *electrolíticos*.



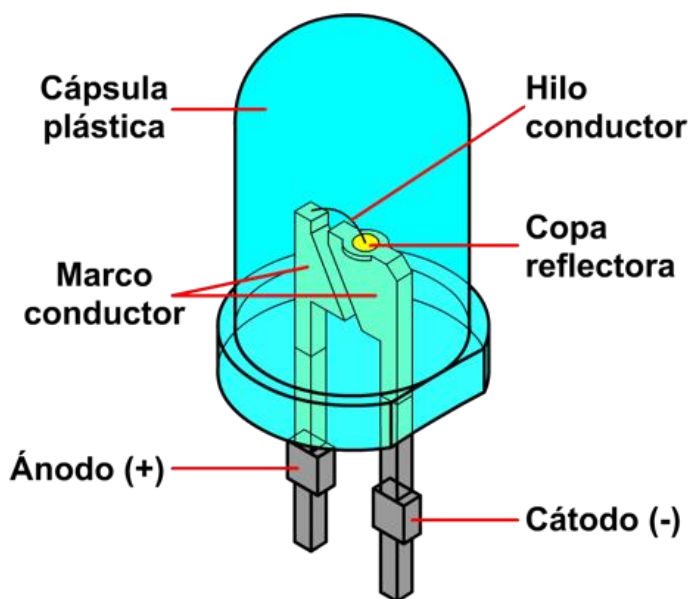
Diodos

Son elementos que permiten que la corriente circule en *una* sola *dirección*. En una terminal posee una resistencia idealmente igual a cero, y en la otra terminal tiene una resistencia idealmente igual a infinito. La franja plateada que tiene en uno de sus extremos indica el lado negativo.



LEDs

La palabra **LED** proviene del inglés *Light Emitting Diode* (Diodo de emisión de luz), son utilizados como indicadores luminosos. Para identificar cuál es la terminal positiva y la negativa hay que observar dentro del LED e identificar la terminal con un “Triángulo” pequeño, esta es la terminal **positiva** (+) y la otra terminal tiene un “Triángulo” más grande e invertido, esta es la terminal **negativa** (-). Si el LED es de color sólido y no se puede ver su interior se puede identificar sus terminales por su longitud, la terminal más larga es la positiva (+) y la terminal más corta es la negativa (-).



Ánodo y Cátodo

Se puede encontrar esta terminología en esquemas, diagramas o bibliografía. Para simplificar la terminología definiremos el **Ánodo** como la terminal **positiva** (+) de un elemento, y **Cátodo** como la terminal **negativa** (-) de un elemento (-).



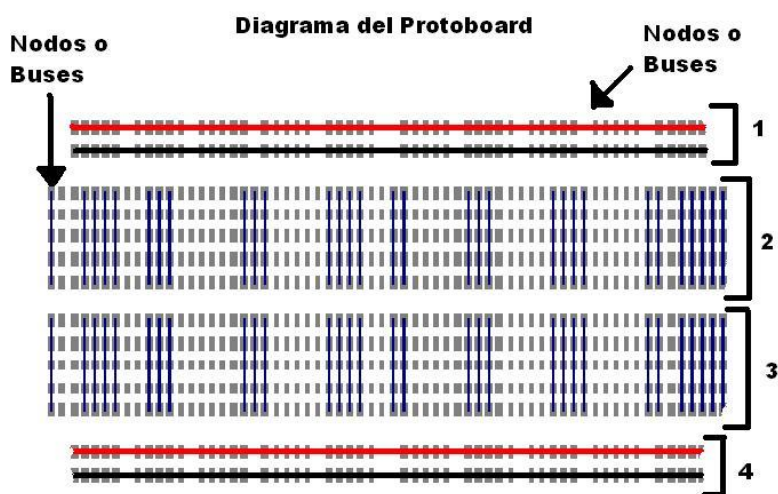
Ánodo

Cátodo

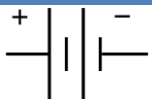


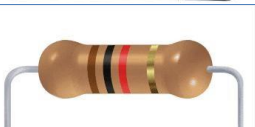


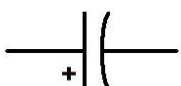





Protoboard – Tableta de prototipos – Placa de pruebas

También llamada **Tableta de prototipos**, o **Placa de pruebas**, es una placa perforada utilizada para ensamblar prototipos electrónicos, sus orificios se encuentran conectados entre sí por patrones señalados en las siguientes imágenes.

La disposición de los orificios es de tal forma que podemos colocar casi cualquier componente básico electrónico, como capacitores, resistencias, interruptores y circuitos integrados.



Simbología Básica:

Componente	Símbolo	Descripción	Imagen
Fuente		Es la encargada de alimentar de corriente al circuito.	
Resistencia		Limita la corriente.	
Capacitor cerámico		Almacena electrones. Se ocupa tanto en corriente directa como en corriente alterna.	
Capacitor electrolítico		Almacena electrones. Se ocupa únicamente en corriente directa.	
Diodo		Permite que la corriente circule en una sola dirección.	
LEDs		Son utilizados como indicadores luminosos.	

Electrónica Analógica.

Es una rama de la electrónica que estudia sistemas donde la corriente eléctrica, voltaje y otras variables, cambian sus valores de forma continua en el tiempo.

Electrónica Digital.

Es una rama de la electrónica que estudia sistemas que poseen solo dos estados llamados niveles lógicos. Estos se representan por valores de voltaje, siendo el valor 0 (cero) correspondiente a 0 [V] y el valor 1 (uno) a un valor otorgado, casi siempre, por la fuente de energía del circuito, comúnmente de 3 a 5 [V]. En la bibliografía, estos valores los podemos encontrar con otros nombres.

1: on, encendido, true, verdadero, high, alto, etc..

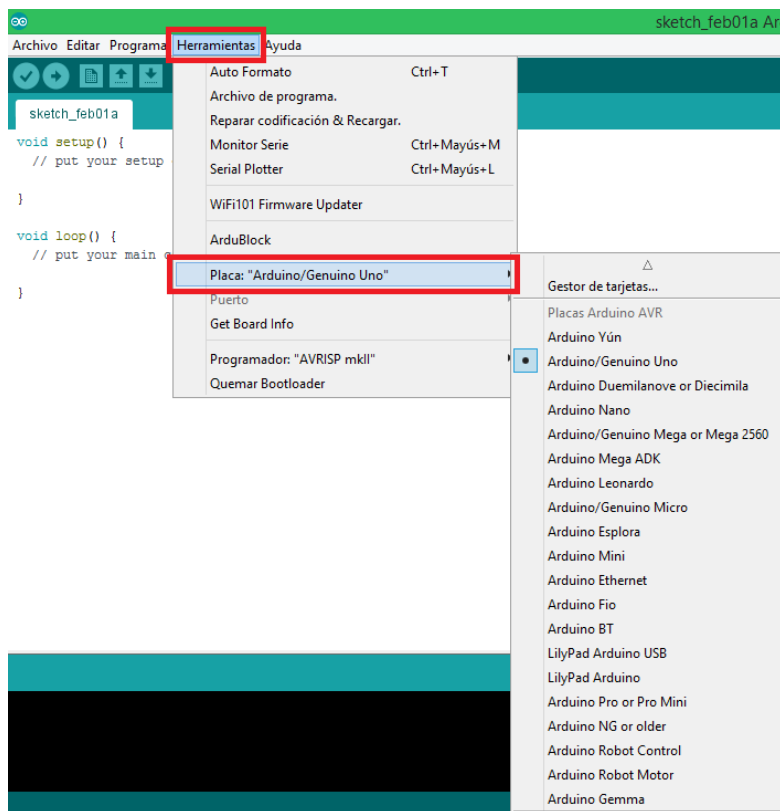
0: off, apagado, false, falso, low, bajo, etc...

Antes de comenzar.

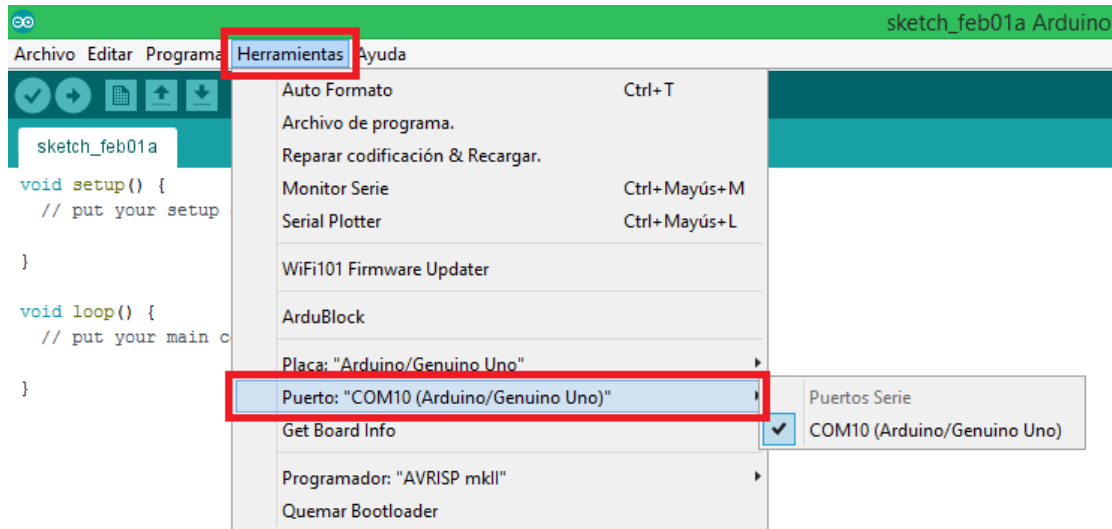
Cerciórate de haber instalado la IDE de *Arduino* y la herramienta *ArduBlocks*. Consulta el manual *Instalación de Arduino IDE* para conocer los requerimientos técnicos y de software con los que se realizaron las prácticas presentadas en este manual.

Instalación de la placa Arduino

Inicia la IDE de *Arduino* y verifica el modelo de placa *Arduino*. Clic en la opción *Herramientas* y selecciona la opción *Placa "Arduino/Genuino Uno"*. Las prácticas contenidas en este manual emplean un modelo *Arduino Uno R3* por lo que seleccionamos la opción "*Arduino/Genuino Uno*", si utilizas otra placa selecciona correctamente el modelo.

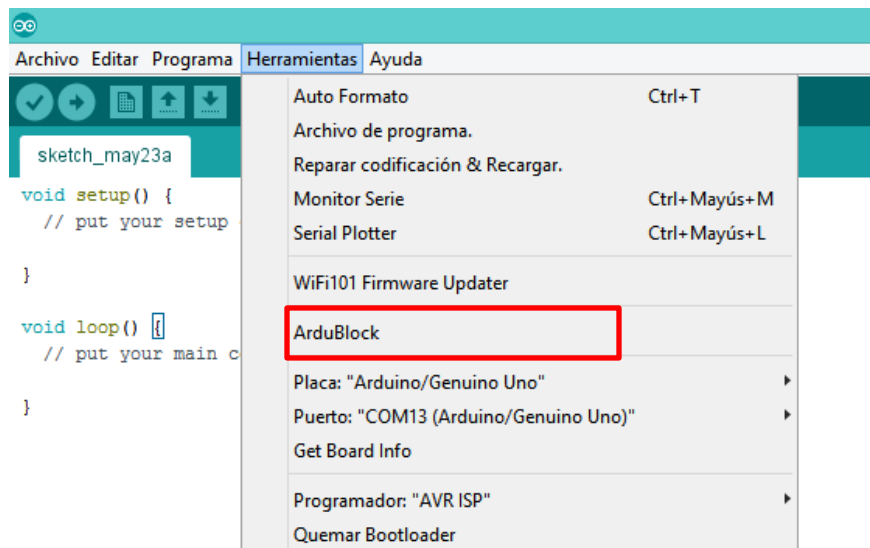


Una vez que seleccionaste el modelo de placa correcto, conecta la placa *Arduino* y observa qué puerto tiene asignado.

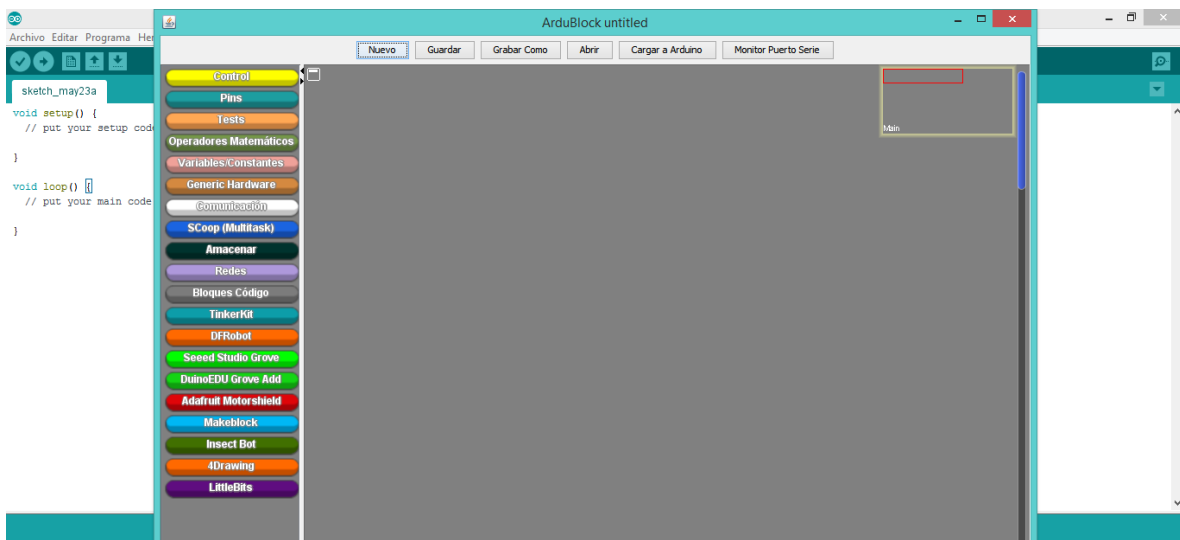


Ejecutar ArduBlock.

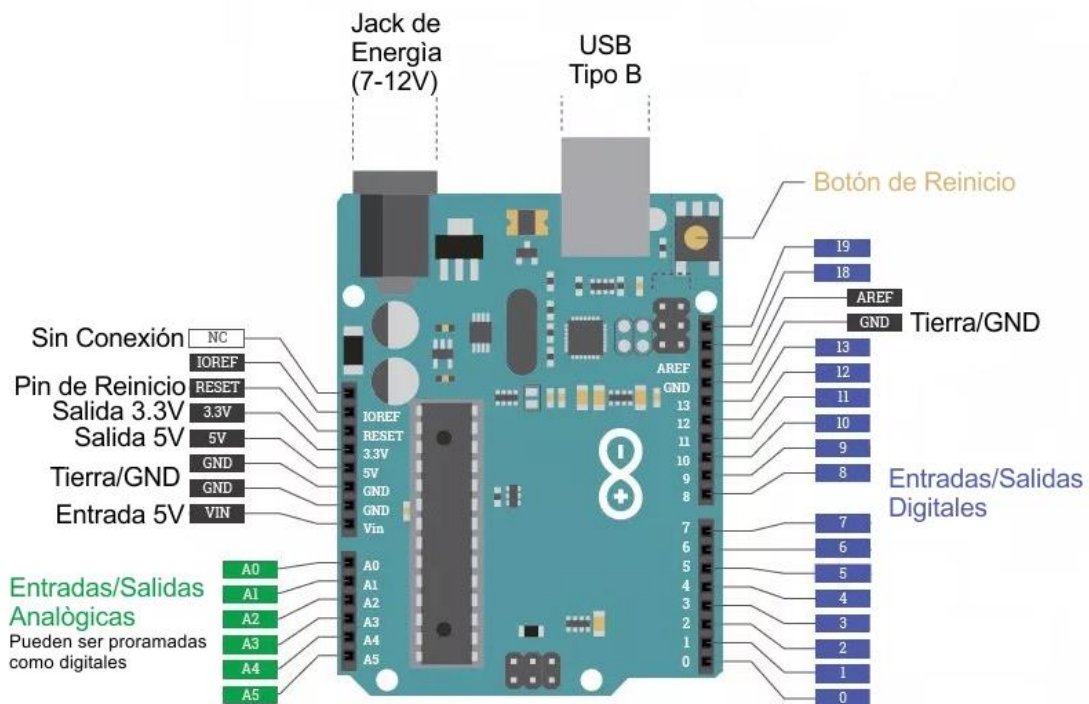
En la IDE de *Arduino*, clic en el menú *Herramientas* y posteriormente Clic en la opción *ArduBlock*



Se abrirá una nueva ventana, como la que se muestra a continuación:



Mapa de Arduino UNO R3



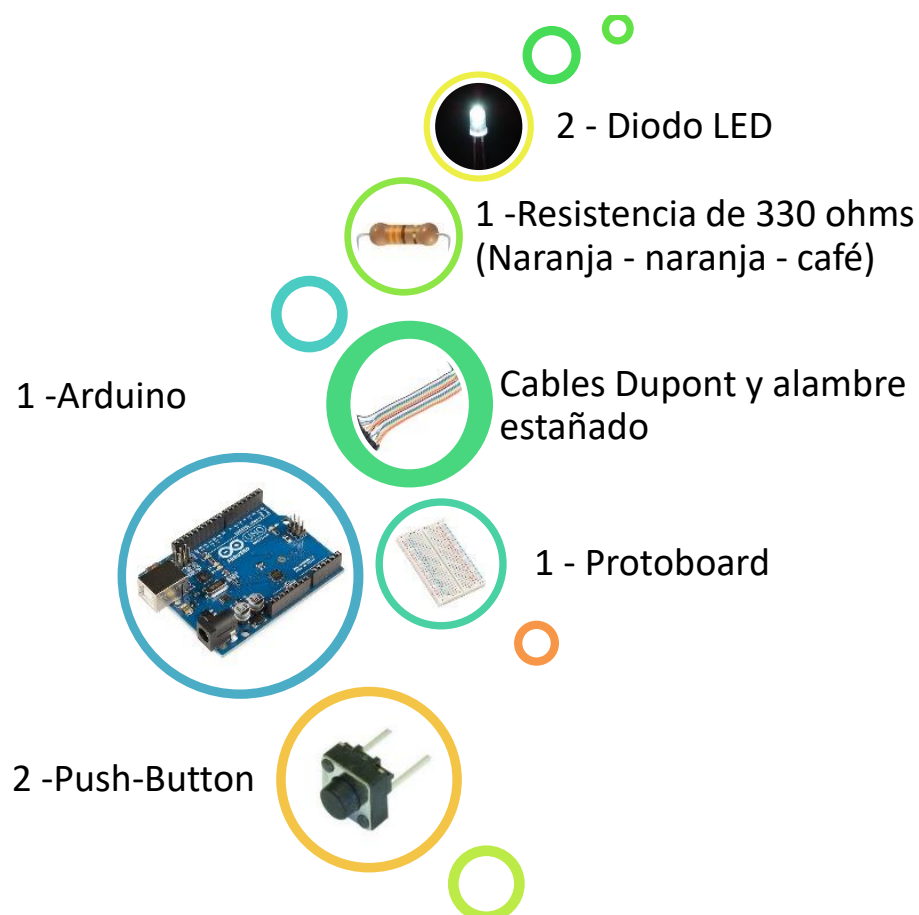
0 – Circuito Básico.

Introducción.

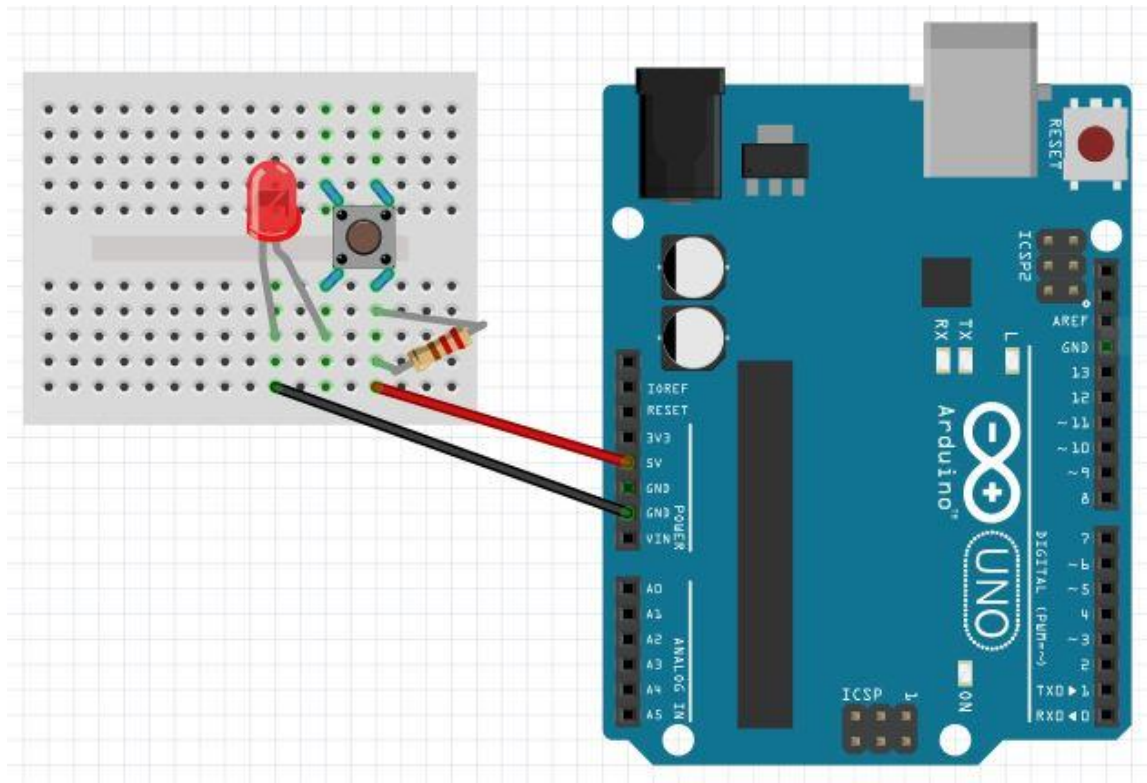
En esta práctica crearás un circuito usando una *protoboard*, *push button*, una *resistencia* de 330 Ohm, un *LED* y *cables jumper*, con la finalidad de obtener los primeros conocimientos del funcionamiento de los dispositivos electrónicos.

Asegúrate que la placa *Arduino* no esté conectada a ninguna fuente de corriente eléctrica antes de montar el circuito.

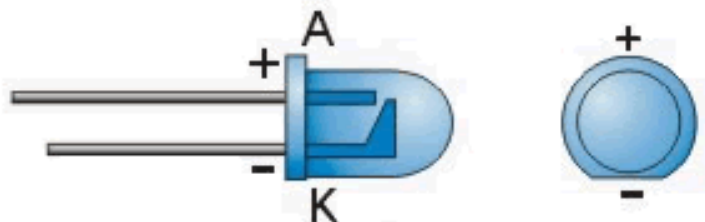
Materiales



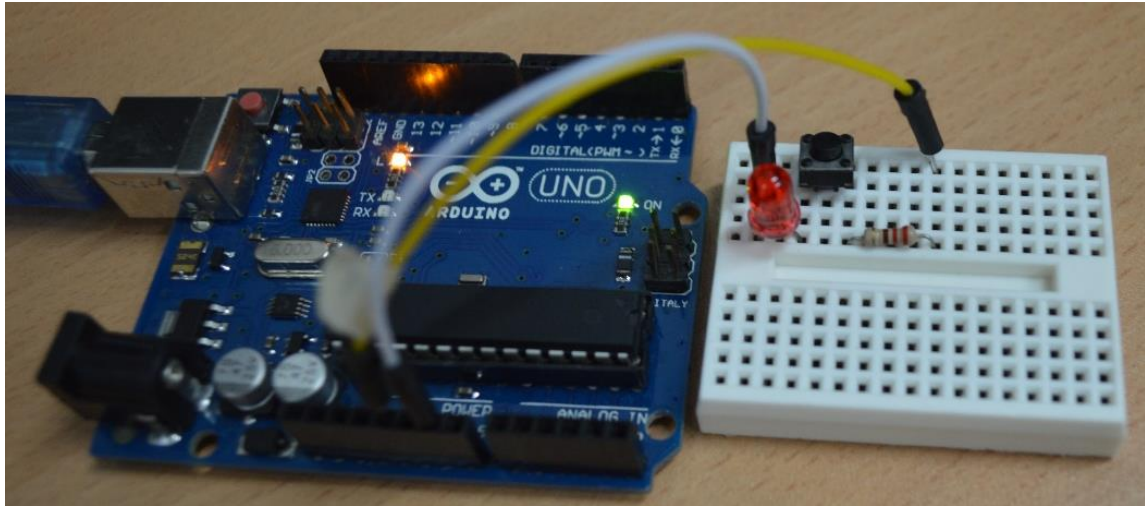
Arma un circuito básico



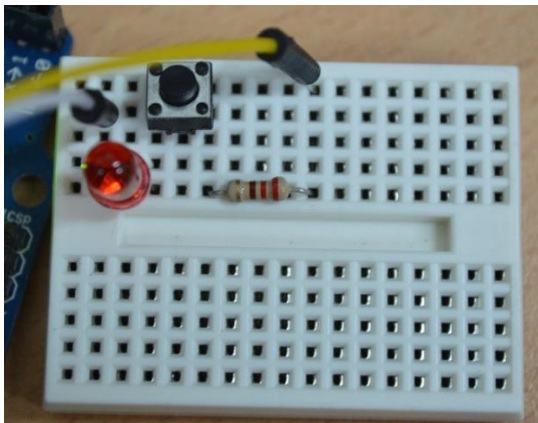
Toma un *cable jumper*, conecta un extremo al pin 5V de Arduino y el otro extremo a un orificio en la placa *protoboard*, en serie, coloca una *resistencia* de 330 ohm y coloca una terminal de la resistencia a una terminal del *push button*. Después en la terminal opuesta del push button, conecta la terminal positiva del *LED* y finalmente la terminal negativa del led se conecta a al pin GND de Arduino con otro *cable jumper*. Recuerda como identificar el positivo y negativo de un *LED*.



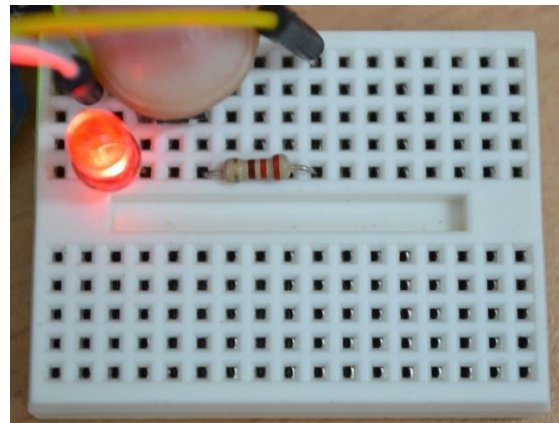
Para alimentar de energía la placa *Arduino*, utiliza el JACK 7-12 V. Si no tienes un conector compatible, puedes utilizar el cable USB para programar tu *Arduino* conectado a tu computadora como fuente de energía. Para este ejemplo usaremos el cable USB.



Una vez conectado, presiona el *botón* y el *LED* deberá encender.



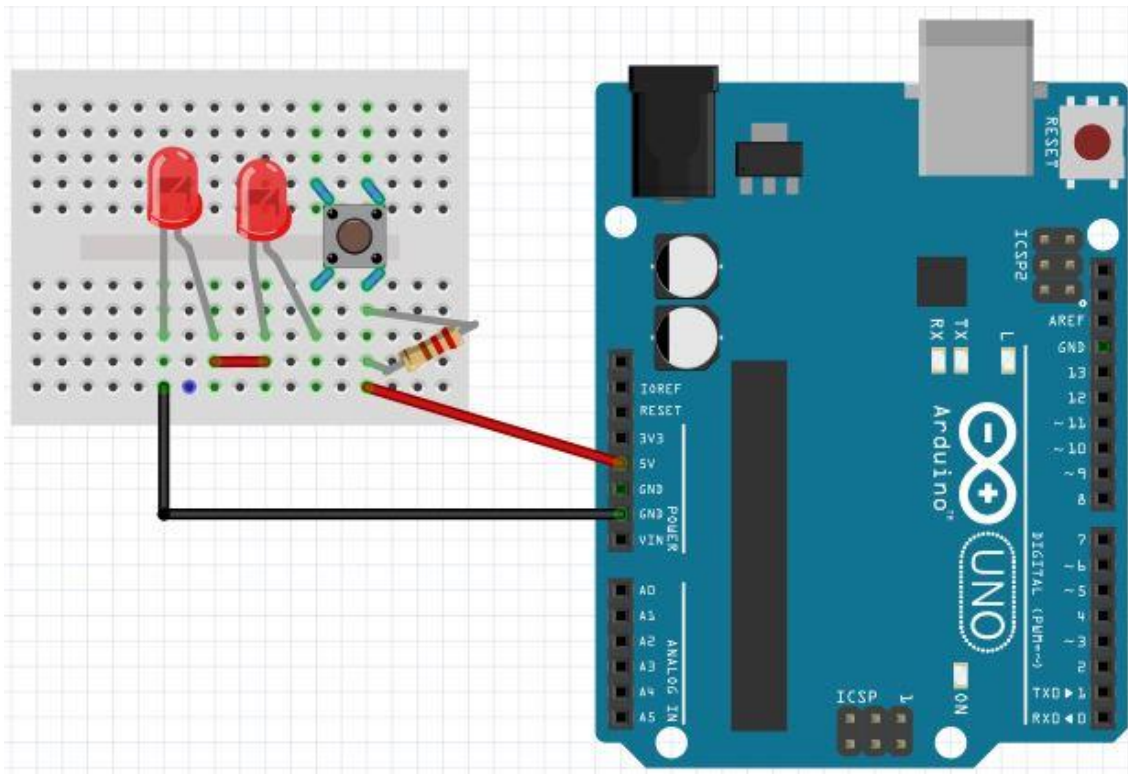
El LED se mantendrá apagado si no presionas el botón.



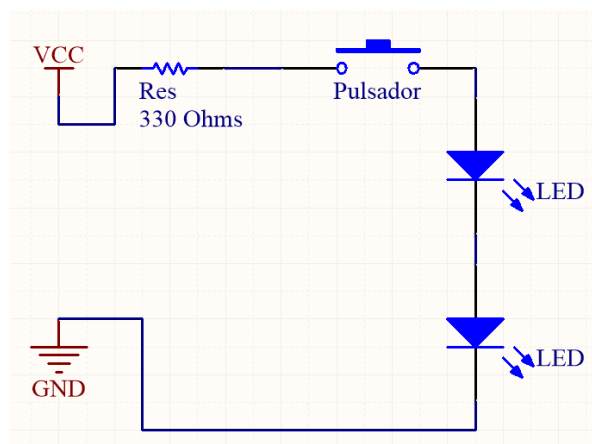
El LED se encenderá cuando presionas el botón.

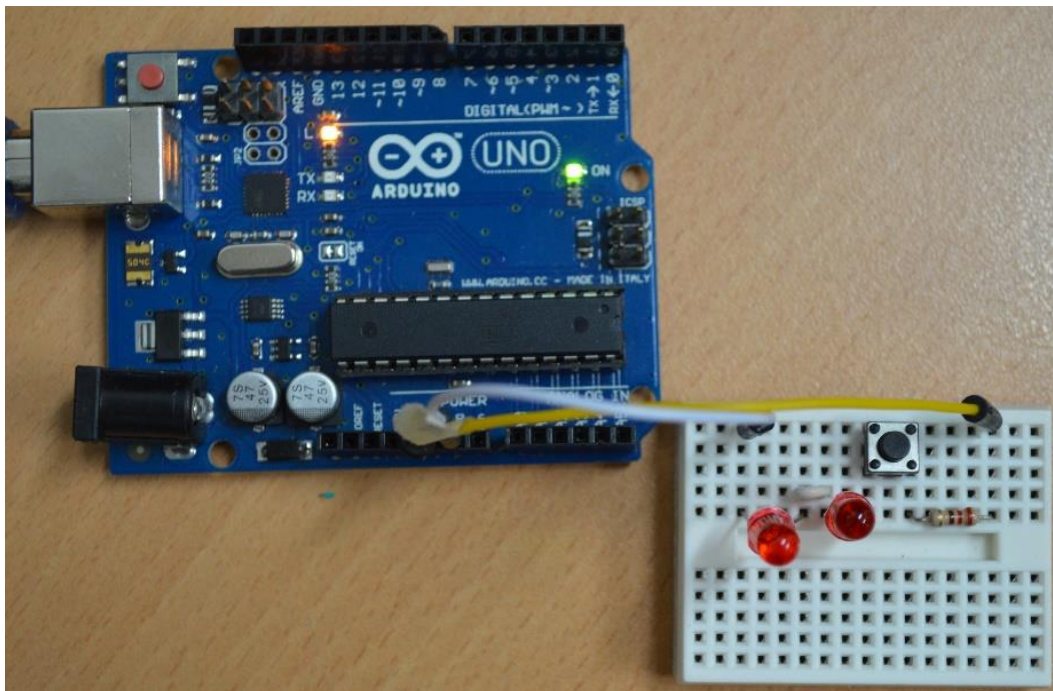
Arma un circuito en serie.

Agrega más *LED* al circuito anterior, corta un trozo de alambre pequeño para conectar las terminales positiva y negativa respectivamente. Utiliza como guía la siguiente figura:

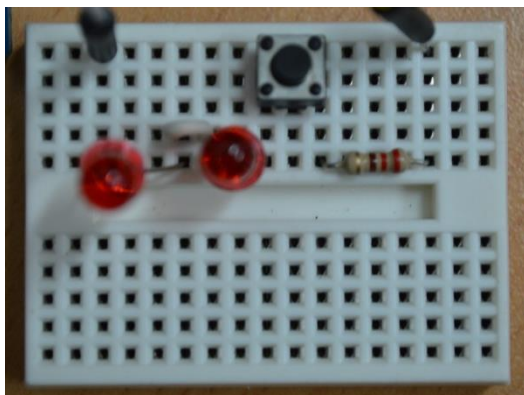


Esta forma de conectar los elementos es conocido como un circuito en serie, observa el siguiente diagrama:

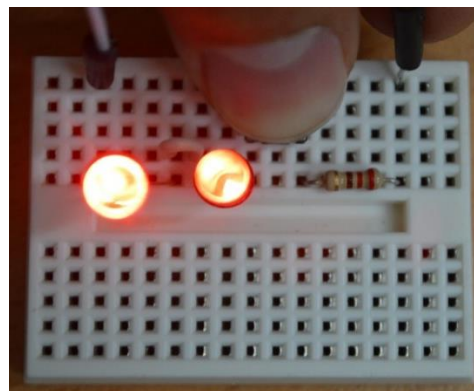




Al presionar el botón ambos *LED* encenderán al mismo tiempo.



Los *LED* permanecerán apagados cada vez que **NO** se presiones el botón

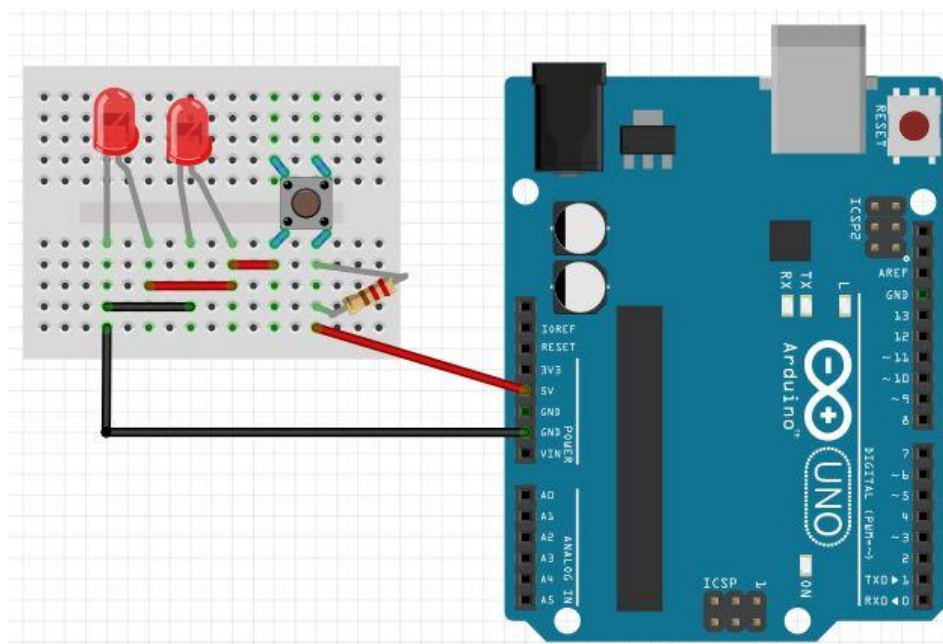


Los *LED* se encenderán cuando se presione el botón

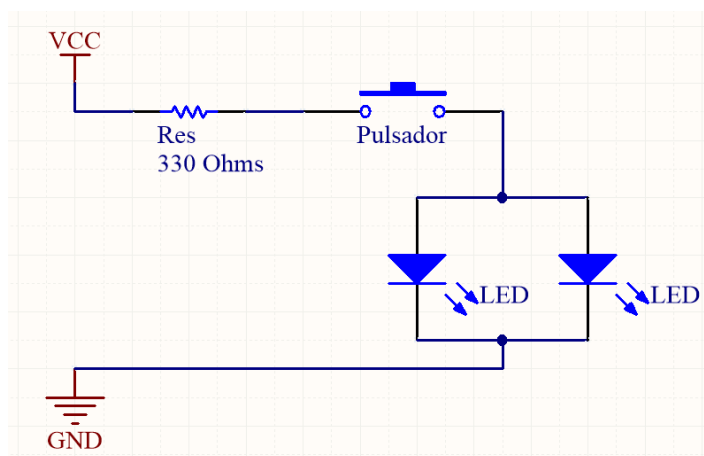
Retira un *LED* sin desconectar ningún alambre o cable. Presiona el botón y observa que, aunque presiones el botón el *LED* restante no enciende.

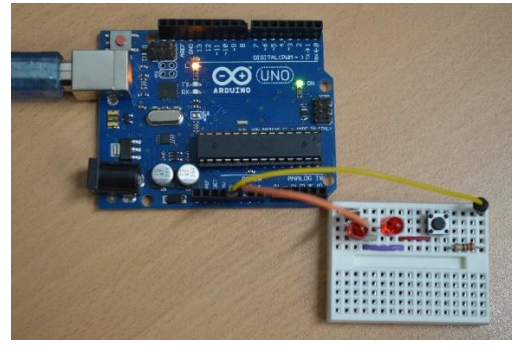
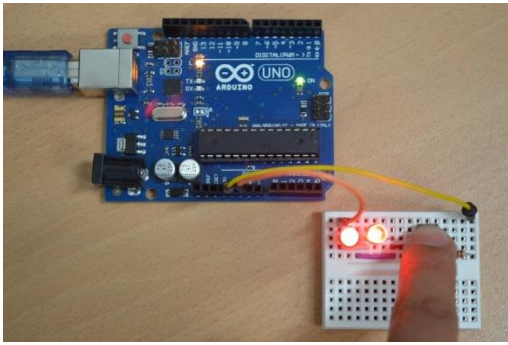
Arma un circuito en paralelo.

Los elementos pueden permanecer en su sitio, pero la conexión entre los *LED* será diferente. Observa la siguiente ilustración:



Las terminales positivas de ambos *LED* se encuentran conectadas en un punto en común, y las terminales negativas a otro punto en común como en el siguiente esquema:





Los *LED* permanecerán apagados cuando no se oprima el botón.

Retira un *LED* sin quitar ningún alambre o cable. Presiona el botón y observa que el *LED* restante enciende aun cuando se retiró el segundo *LED*.

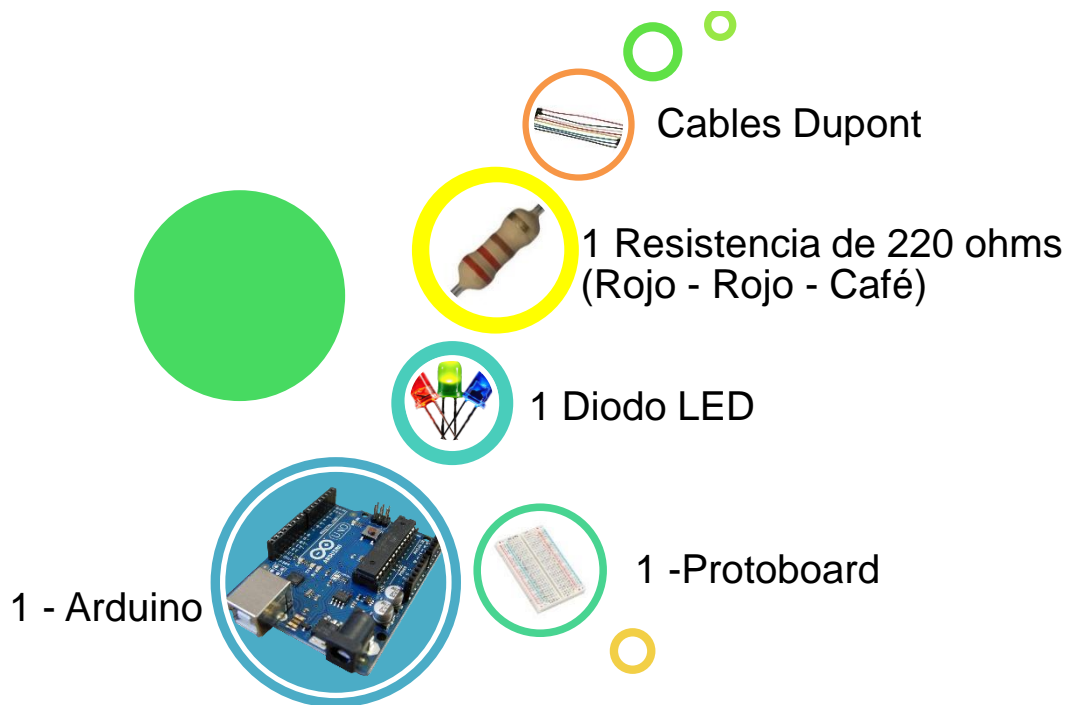
1 - Hola Mundo

Introducción

En el universo de la programación, se le conoce “Hola Mundo” al primer programa que se realiza en cualquier lenguaje de programación como una prueba de que el programa y los sistemas asociados a este, están funcionando adecuadamente.

En el caso de *Arduino*, un “Hola Mundo” es un programa muy sencillo que te permite visualizar el correcto funcionamiento de la placa, sus conexiones y tu programa al mismo tiempo.

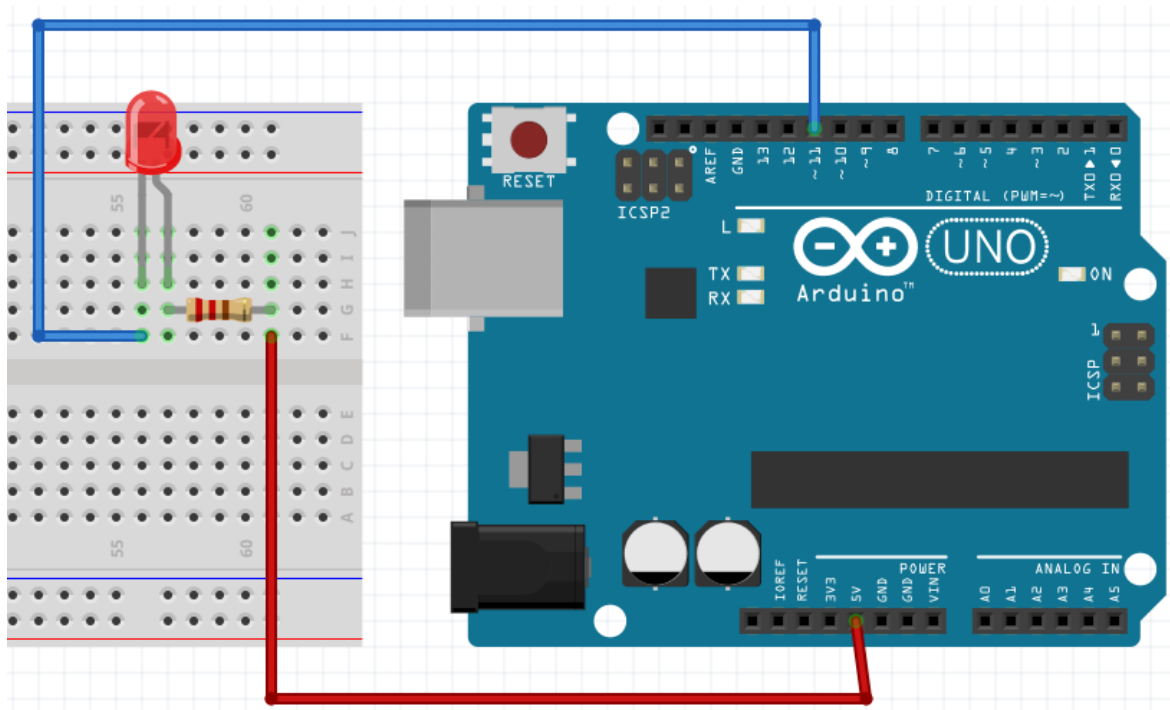
Materiales



Circuito

Para tu primer programa, debes armar el siguiente circuito con el material antes mencionado.

Es un circuito muy sencillo. Las resistencias no tienen terminales positivas negativas, por lo que no importa en que orden la coloques. Observa cómo se coloca el LED, recuerda que este si tiene terminal positiva y negativa.

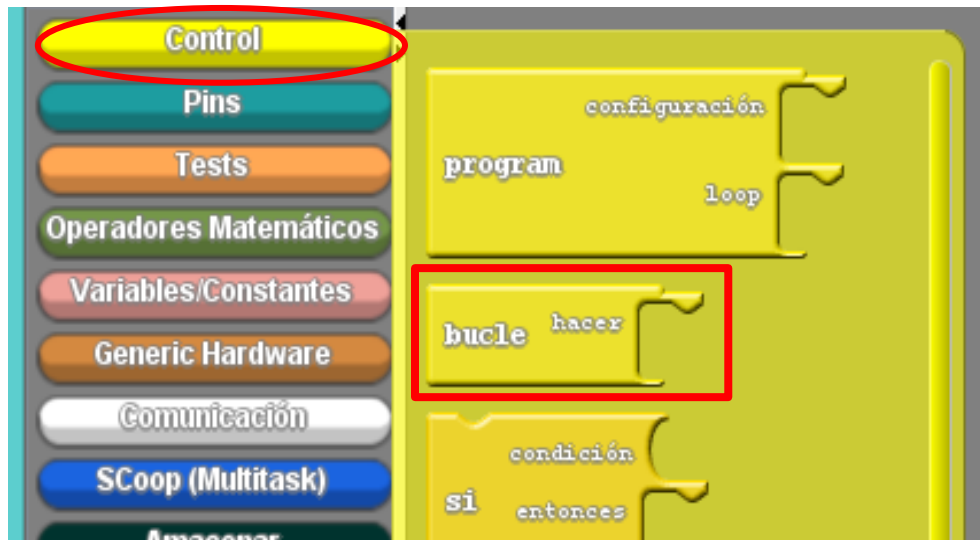


Programa

Ejecuta el ambiente de desarrollo **ArduBlock**.

De lado izquierdo se encuentran todos los bloques que se utilizarán y están ordenados por categorías, selecciona el bloque deseado y arrastrarlo hacia la zona oscura.

Todos los programas comenzarán con el bloque “**bucle Hacer**” o con el bloque “**program – configuración – loop**”



Este bloque tiene la función de crear un ciclo durante tiempo indefinido, deteniéndose sólo cuando se desconecta la alimentación o se reprograma la tarjeta.

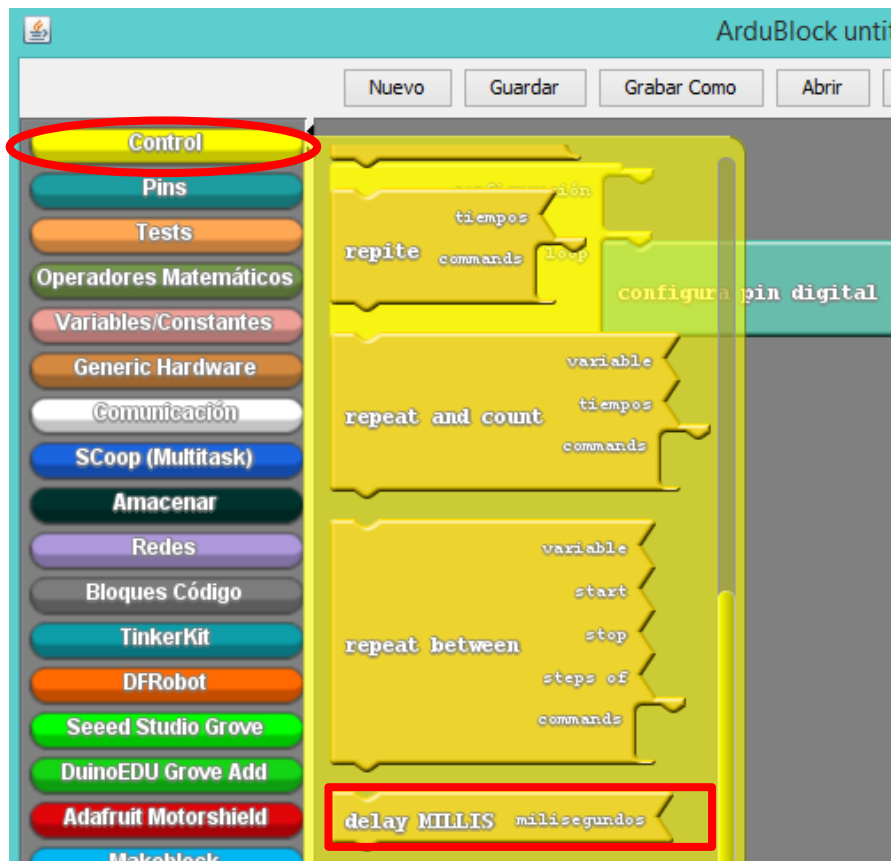
Lo que se colocará dentro de este bloque serán instrucciones para enviar una señal a través del pin 11 de la tarjeta *Arduino* para que el *LED* se encienda.

Arrastra el bloque “**Configura pin digital**” dentro del bloque “**Bucle hacer**”. Ingresa el número del **pin** (11) y la siguiente opción deja seleccionado “**HIGH**”.

La palabra “High” o “Alto” indica que ese pin tendrá como salida digital el número 1. El lenguaje en electrónica es de 1’s y 0’s, por ello se refiere al 1 como en alto, encendido o incluso a manera de indicar que la salida tiene de 3-5 [V] y se refiere al 0 como en bajo (LOW), apagado, o como 0 [V].



Enseguida coloca una pausa de 1.5 segundos (1500 milisegundos), tiempo durante el cual el *LED* se mantendrá encendido.

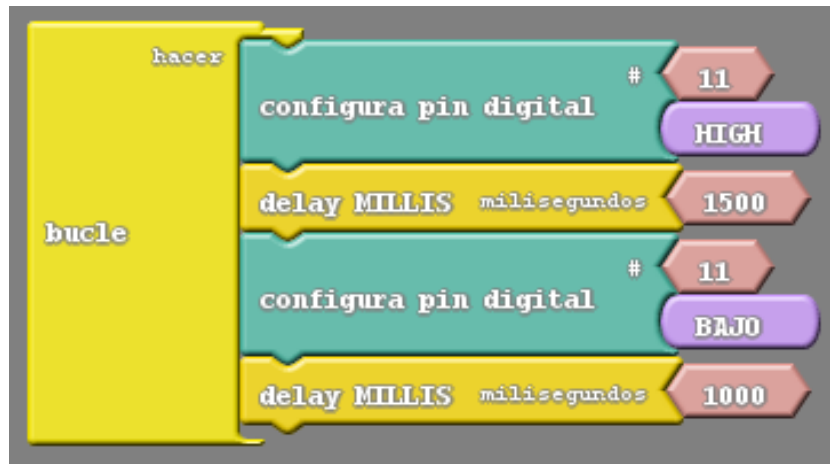


En electrónica, el tiempo, por lo general, se mide en milisegundos, entonces es común que encuentres las instrucciones referidas en esta unidad. Cuando agregues módulos que manejen tiempo deberás considerar si miden el tiempo en milisegundos o en segundos.



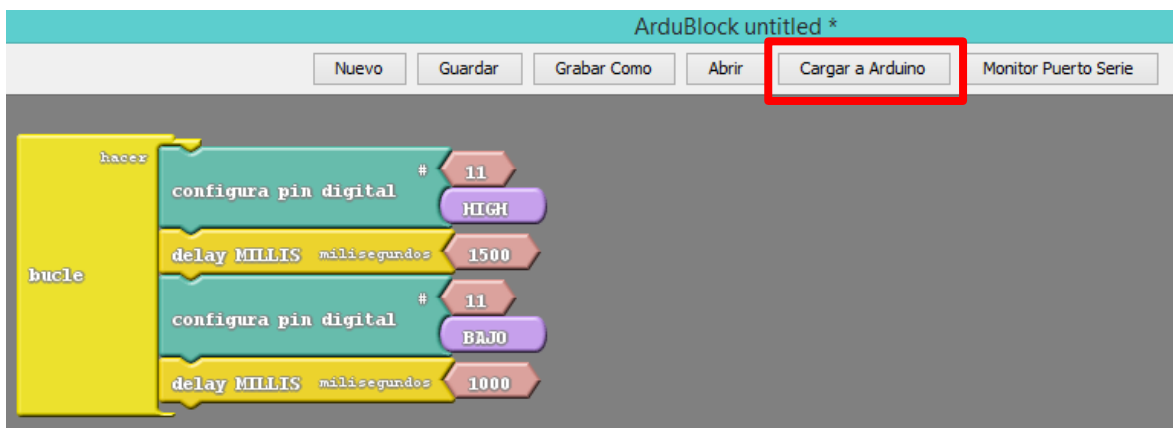
Después agrega otro bloque “**configura pin digital**” y otra pausa de 1 segundo (1000 milisegundos) durante la cual el *LED* permanecerá apagado. Vuelve a ingresar el número de pin (11) y ahora clic en “**HIGH**” y selecciona “**BAJO**”

El programa debe quedar de la siguiente manera:



Puedes encontrar este programa en la carpeta:
KitRoverEducativo\Practicas\HM\HM.abp

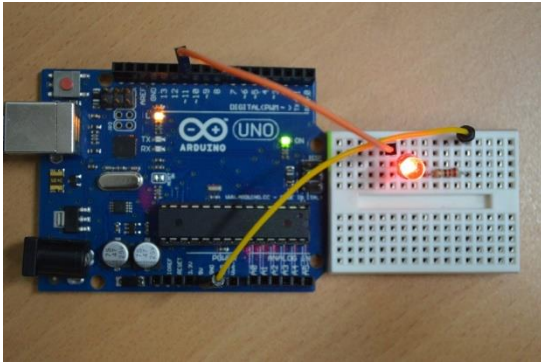
Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.



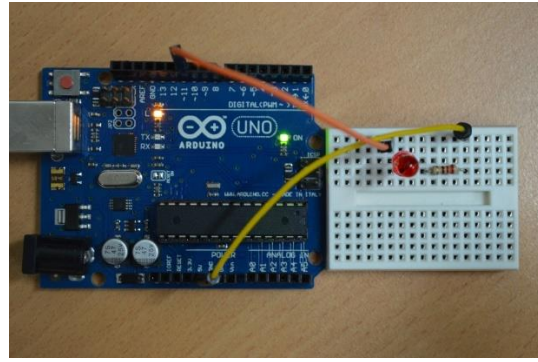
Espera a que se complete el proceso y observa los resultados.

Resultados.

Sin que se presione botón alguno, el LED comenzará a “parpadear”



El LED enciende durante 1500 milisegundos.



El LED se apaga durante 1000 milisegundos.

Modifica tu programa para que encienda y apague más rápido.

¿Qué pasará si cambias el tiempo de encendido y apagado a 100 [ms]? ¿y si colocas 1 [ms] en el tiempo de encendido y apagado?

2 Sensor de luz LDR.

Introducción

En esta práctica, construirás un sistema que mida la cantidad de luz del ambiente, y dependiendo de esta cantidad, el sistema deberá encender o apagar un LED.

Para ello emplearás una **fotorresistencia** (Light Dependent Resistor - LDR) que es un componente electrónico cuya resistencia varía dependiendo de la cantidad de luz que reciba. A mayor cantidad de luz menor resistencia.

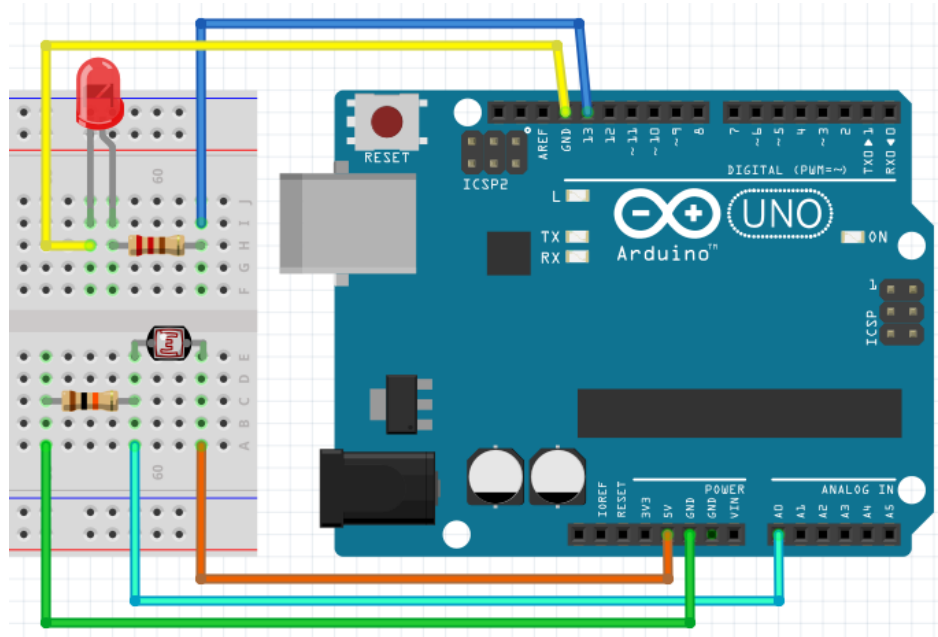
Este sensor, provee un número teórico infinito de valores, pues es un sensor analógico. Por esta razón deberás emplear los pines **analógicos** del *Arduino* para poder obtener la información deseada.

Materiales



Circuito.

Construye el siguiente circuito:



Nota que el sensor LDR, no tiene una terminal positiva o negativa, así que no importa en qué sentido la coloques, siempre y cuando una terminal esté conectada a 5 [V], la otra terminal deberá estar conectada tanto a tierra o GND, como al pin de entrada analógica número 0 (cero)

Programa

Ejecuta el ambiente de desarrollo **ArduBlock**.

Ingresa un bloque de control “**Bucle hacer**” y define una variable que pueda almacenar el valor que arroja el sensor LDR. En el menú izquierdo, clic en la opción variables/constantes y arrastra el bloque “**set integer variable**”.



Nombra la variable como “**ldrStatus**” y del menú izquierdo, clic en “**Pins**”, arrastra la opción “**pin analógico**” y selecciona el pin de donde obtendrás la información, para esta práctica, el pin A0.

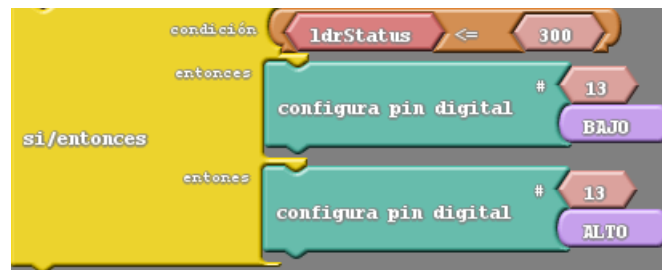
A continuación, debajo del bloque “set integer variable”, clic en el menú izquierdo, opción “Control” y arrastra el bloque “si/entonces”. Este bloque permite tener control sobre las variables y condiciones del programa. Si se cumple la condición especificada, ejecuta una sección del código, y si no se cumple esa condición ejecuta otra parte del código.

En el menú izquierdo, clic en la opción “Tests” y arrastra un bloque con el símbolo “<=” en la parte “condición” del bloque “si/entonces”. Este bloque nos permite comparar valores fijos o almacenados en una variable. En este caso compara el valor obtenido del LDR almacenado en la variable *ldrStatus* con el valor 300.

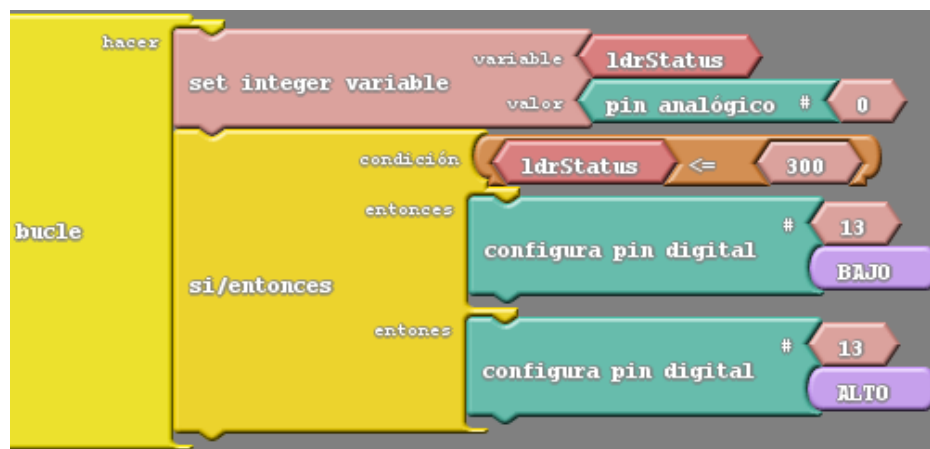


Si *ldrStatus* es “menor o igual a” 300 ejecutará el código en la primera sentencia “entonces” y *si no es menor o igual a 300*, el valor almacenado en *ldrStatus*, es mayor a 300 y por lo tanto ejecuta el código en la segunda sentencia “entonces”.

En la primera sentencia “entonces” ingresa un bloque “configura pin digital” y selecciona el pin 13 en “bajo”. En la segunda sentencia “entonces” ingresa un bloque “configura pin digital” y selecciona el pin 13 en “Alto”.

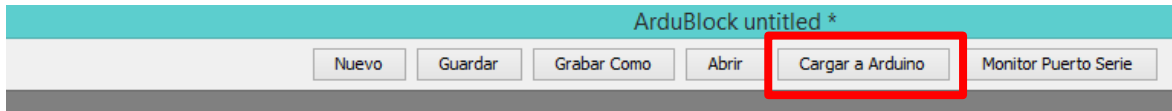


El programa debe quedar de la siguiente manera:



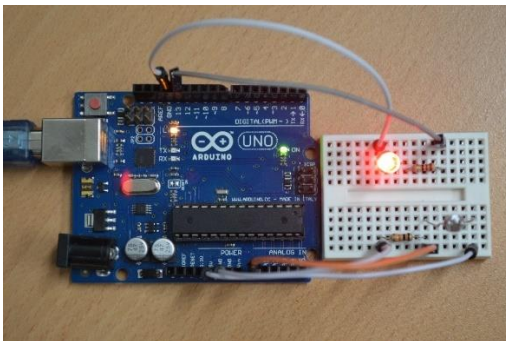
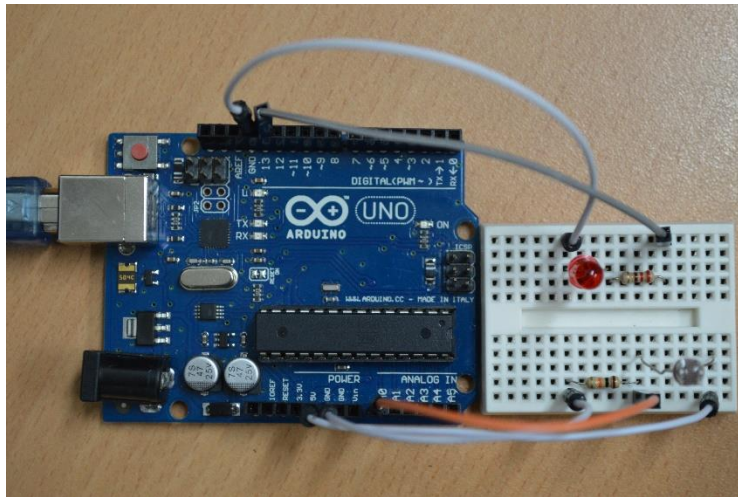
Puedes encontrar este programa en la carpeta:
KitRoverEducativo\Practicas\LDR\LDR.abp

Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.

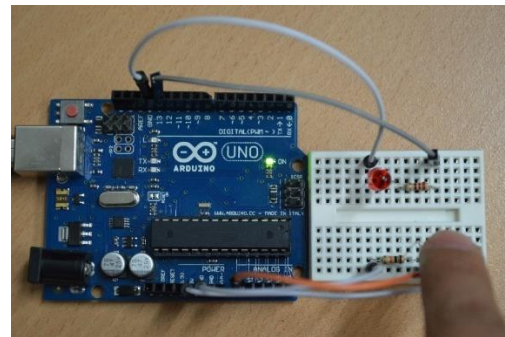


Espera a que se complete el proceso y observa los resultados.

Resultados



El LED se encenderá en cada vez que el LDR este libre



Cuando cubres el LDR se apaga el LED

Intenta modificar el programa para que el sensor LDR, cuando se encuentre libre, el LED esté apagado, y cuando cubras el sensor LDR el LED se encienda. Eso simulará un dispositivo que enciende o apaga un foco en la vía pública o en tu casa.

3 Sensor de Temperatura

Introducción

Este sistema medirá la temperatura ambiente utilizando un sensor **LM35** y mostrará su valor en tiempo real. Para ello se empleará el “**monitor serial**” y adicionalmente, cuando detecte una temperatura igual o superior a 25 [°C] encenderá un LED.

El sensor analógico LM35 puede medir temperaturas de -55 [°C] a 150 [°C], si se expone a temperaturas menores o mayores a este rango, en primera instancia te dará lecturas erróneas o incongruentes, y en última instancia dejará de funcionar. Sin embargo, existen variantes con rangos más cortos.

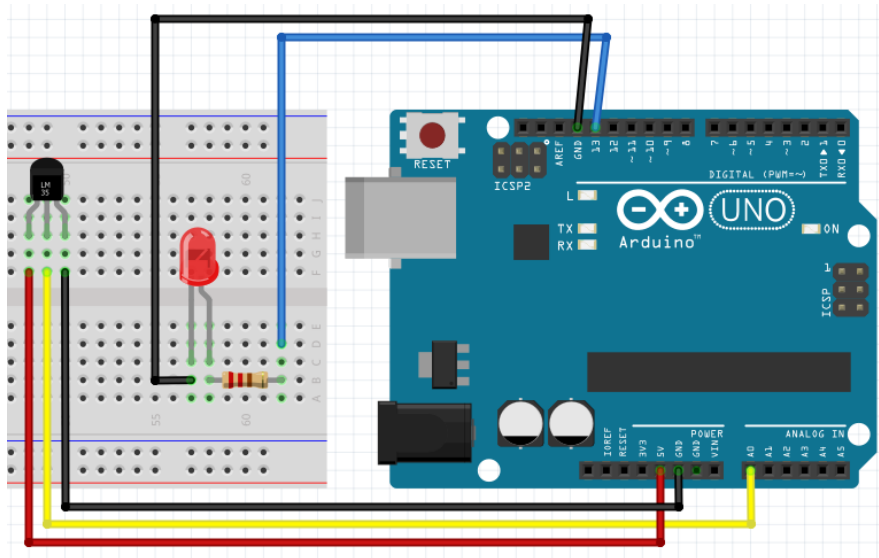
	MIN	MAX	UNIT
LM35, LM35A	-55	150	°C
LM35C, LM35CA	-40	110	
LM35D	0	100	

Materiales

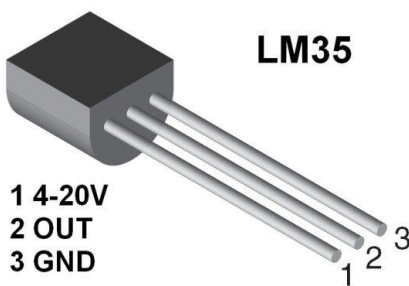


Circuito

Construye el siguiente circuito:



Observa que el sensor es un encapsulado negro con una parte “lisa” y una parte “abultada”. Observa el siguiente esquema para ubicar correctamente las terminales del sensor:



La *terminal* 3 deberá ser conectada a GND o “Tierra”, la *terminal* 1 deberá ser conectada a 5 [V] y la *terminal* 2 al pin A0 por ser un sensor analógico.

Programa

Ejecuta el ambiente de desarrollo **ArduBlock**.

Ingresa un bloque de control “**Bucle hacer**” y define una variable de nombre “**temperatura**” donde se almacenará el valor que del sensor *LM35*. Los valores que brinda el *LM35* deben ser corregidos, para ello deberás agregar la siguiente operación matemática:

$$Temperatura = \frac{((PinA0) * 5)100}{1023}$$

En **ArduBlock** debes arrastrar los bloques del menú izquierdo, opción “**operadores matemáticos**”, emplea bloques anidados para tener la operación completa:



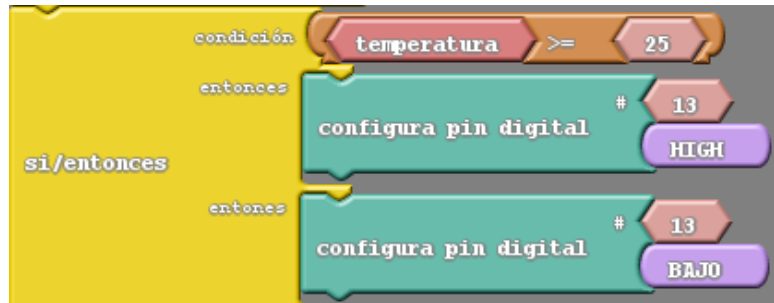
En el menú izquierdo, clic en “**Comunicación**” y selecciona el bloque “**Impresión serial**”. Arrástralo y suéltalo debajo de la definición de la variable *temperatura*. En el mismo menú, Agrega un bloque de nombre “**glue**”. Clic en el menú Variables/Constantes, selecciona el bloque de nombre “**integer variable name**”, arrástralo y suéltalo frente al bloque “**glue**”. Escribe el nombre de la variable *temperatura*.



Agrega un retraso de medio segundo debajo del bloque “**Impresión serial**”:



Agrega un bloque “si/entonces” debajo del retraso de medio segundo. Agrega la condición para un valor de *temperatura* mayor o igual a 25[°C]. Si el valor es mayor a 25 [°C] el LED en el pin 13, deberá *encender*, si no, deberá apagarse.

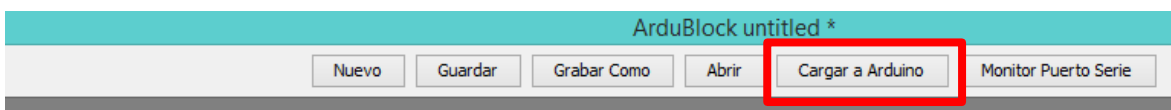


El programa debe quedar de la siguiente manera:



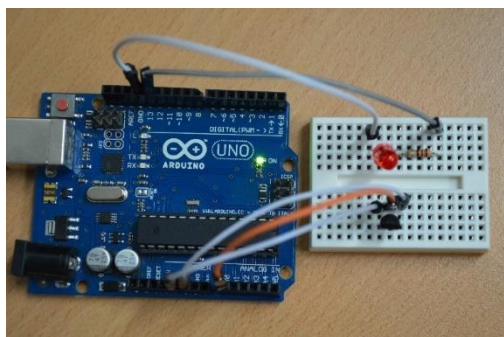
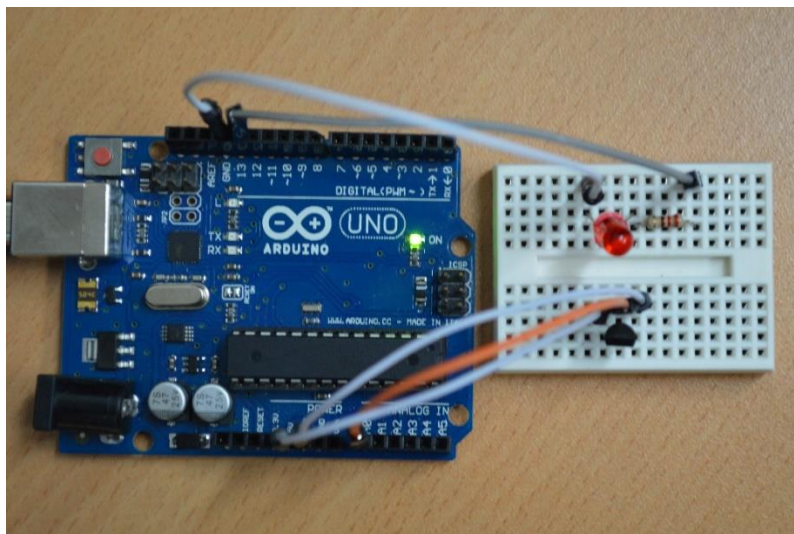
Puedes encontrar este programa en la carpeta:
KitRoverEducativo\Practicas\TEMPERATURA\temperatura.abp

Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.



Espera a que se complete el proceso y observa los resultados.

Resultados



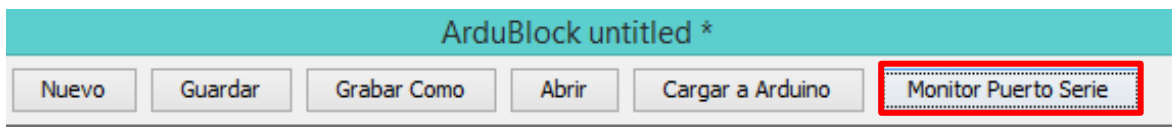
El LED se encuentra apagado cuando el sensor LM35 detecta temperaturas menores a 25 [°C]



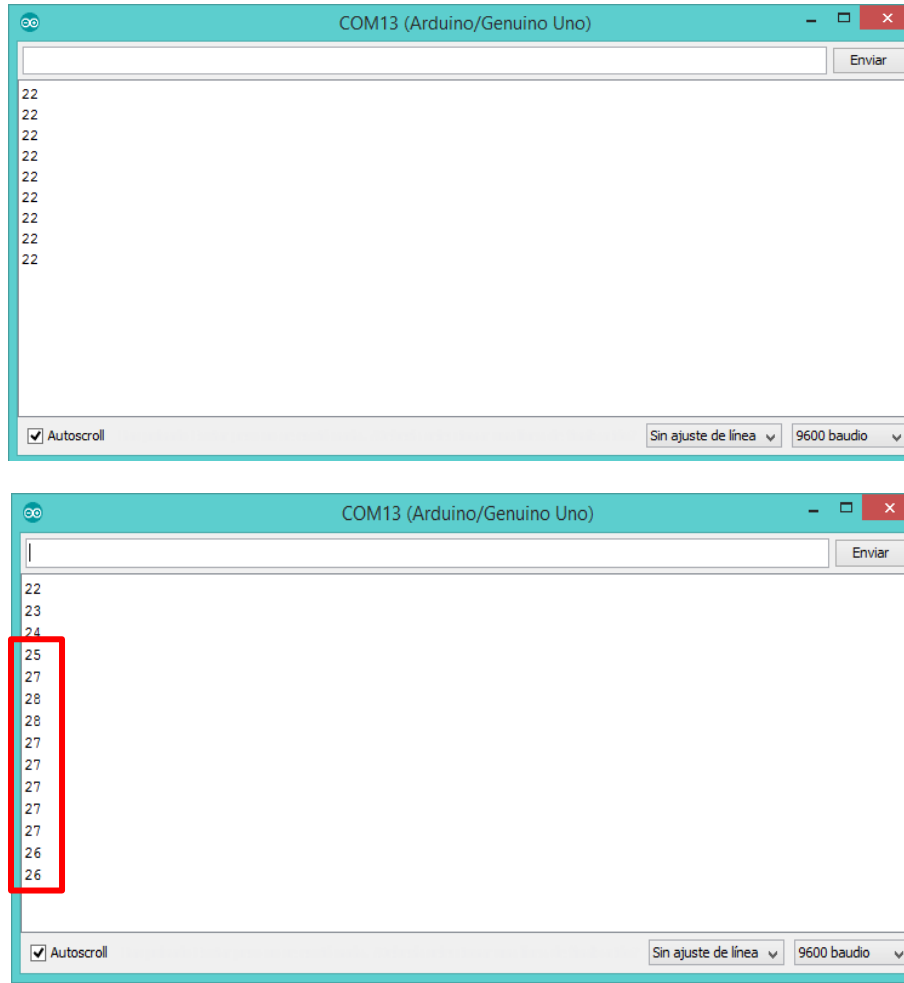
El LED se enciende cuando el sensor LM35 detecta temperaturas mayores a 25 [°C]

Para probar tu sensor, muévelo a un cuarto con una mayor temperatura, ponlo al sol, utiliza una pistola de calor u oprímelo con tus dedos para que el sensor mida tu temperatura corporal.

Para observar los valores que arroja el sensor LM35, abre el Monitor Serial. Clic en el botón “Monitor Puerto Serie”.



Observa que cada renglón, es un ciclo de lectura.



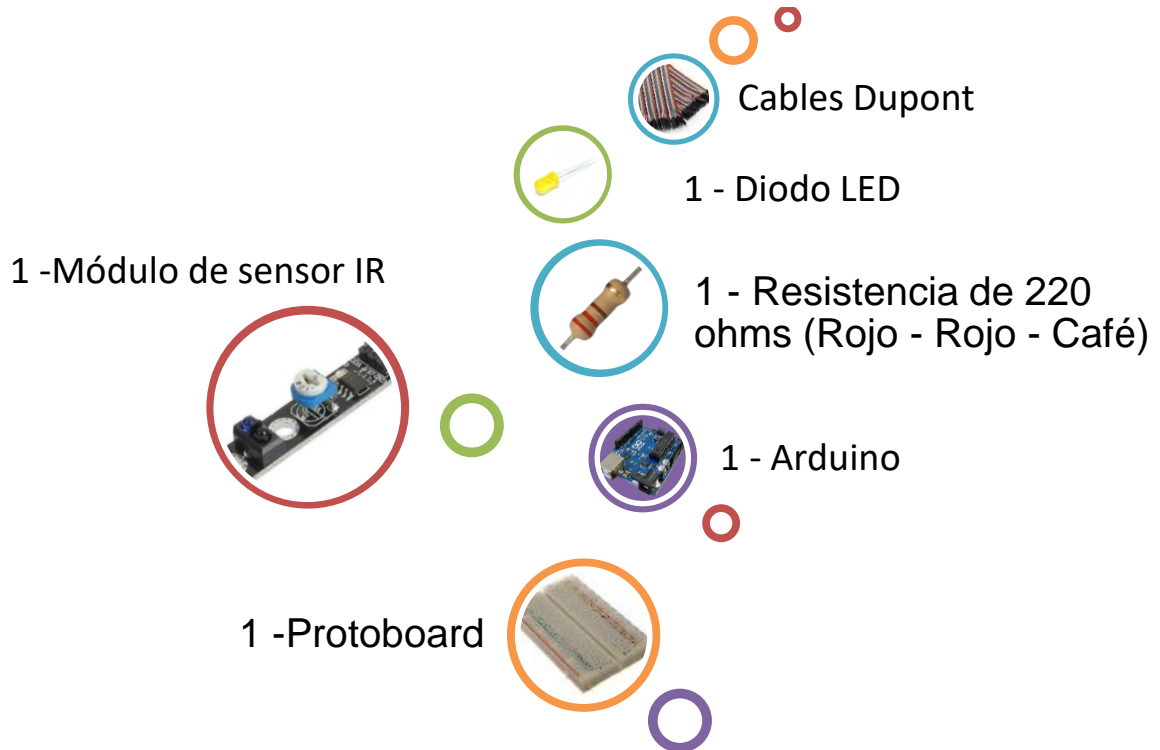
Modifica el programa para que el *LED* encienda a 30 [°C].

Construye un semáforo de temperatura. Agrega 3 **LED** más, utiliza los colores verde, amarillo y rojo. En el bloque “**si/entonces**”, en la segunda sentencia “**entonces**” agrega otro bloque “**si/entonces**”.

Agrega las condiciones correspondientes para que el LED verde encienda cuando la temperatura es menor a 20 [°C], el LED amarillo encienda cuando la temperatura es entre 21 y 30 [°C] y el LED rojo cuando la temperatura es mayor a 30 [°C].

4 Sensor Infrarrojo (IR)

Materiales



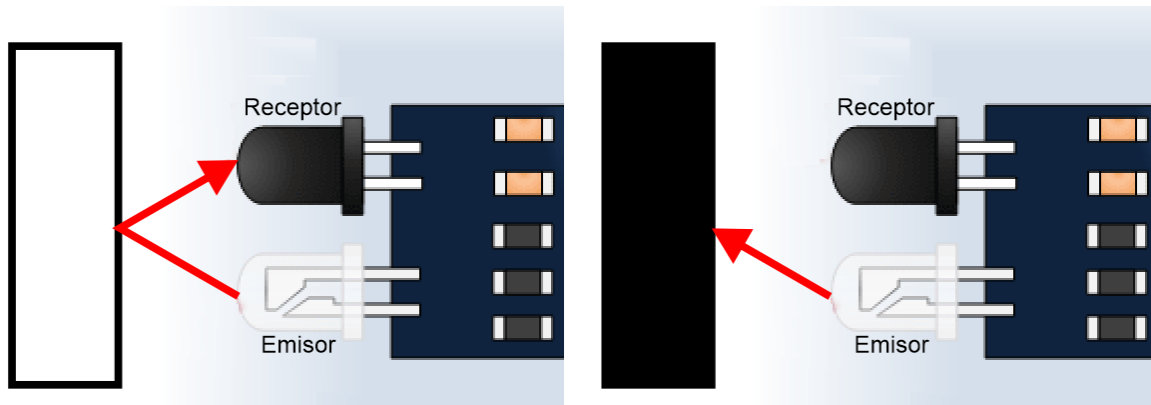
Introducción

Los sensores infrarrojos tienen una diversa aplicación en el mundo de la robótica. Pueden detectar cambios en texturas, medir distancias, funcionar como interruptores o herramientas de control.

Los módulos IR (Infra Rojo) se componen, básicamente, de un emisor, un receptor y una sección que calibra la sensibilidad de estos para mejorar su precisión. El módulo que se ocupará en esta práctica cuenta con el emisor (LED traslucido o azul) y receptor (LED oscuro o negro) “apuntado” a la misma dirección. Con ello se logra que cuando el emisor radie la luz infrarroja, esta rebota en una superficie de regreso al receptor.

En esta práctica se ocupará el módulo IR como un detector de “texturas” o de “tonos”, es decir, calibraremos el módulo para que reaccione ante una superficie de color blanco o negro. Recordemos que el color blanco permite la reflexión de todos los colores,

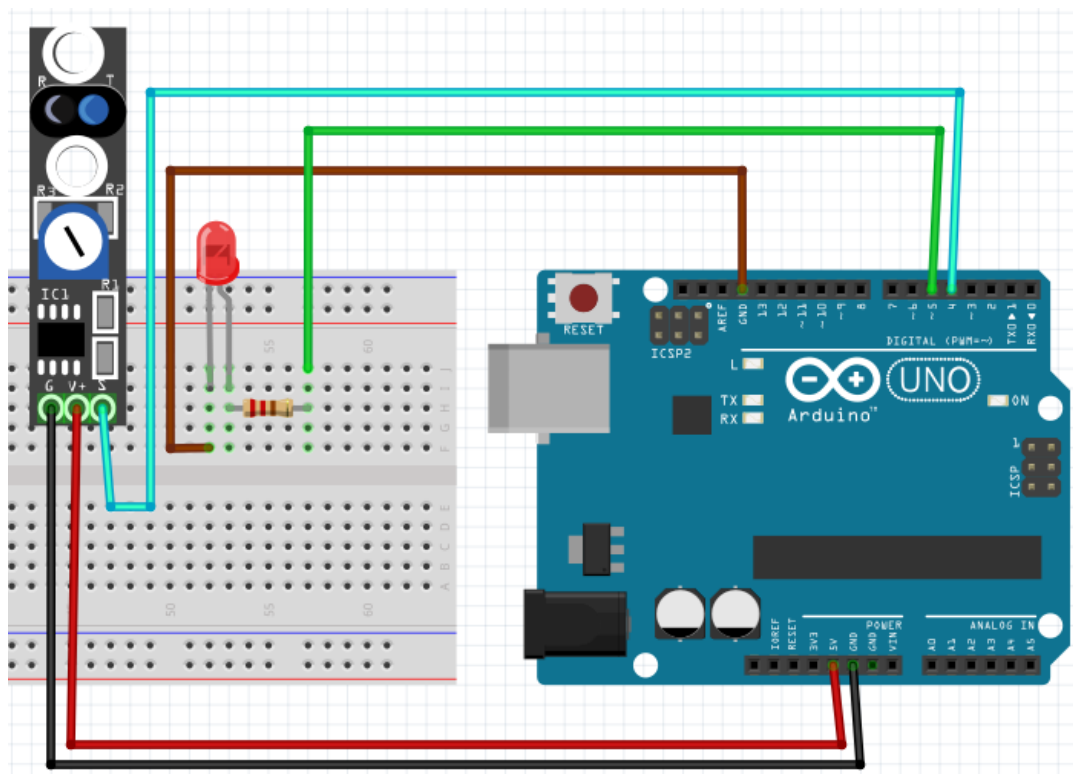
incluyendo la luz infrarroja y el color negro, por el contrario, absorbe la luz impidiendo que el receptor detecte la luz radiada por el emisor.



Ocuparemos este sensor, como un sensor digital, es decir, solo obtendremos dos valores: 1 o 0 (prendido o apagado).

Circuito

Construye el siguiente circuito:



El sensor recomendado en este manual es el siguiente:



Nota que el emisor y receptor están encapsulados “hacia abajo” o “hacia arriba”, esto permite una fácil instalación en sistemas que se mueven y detectan cosas por debajo de ellos, por ejemplo, los vehículos seguidores de líneas.

Conecta la **terminal 1** identificada con la letra G, a GND o tierra; la **terminal 2** identificada con la letra y símbolo V+ a 5[V] y la **terminal 3** identificada con la letra S al **pin digital 4**.

Programa

Ejecuta el ambiente de desarrollo ArduBlock.

Ingresa un bloque de control “Bucle hacer” y define una variable digital de nombre “IRPin” donde se almacenará el valor que del sensor IR. En el menú izquierdo, clic en “DFRobot”, arrastra y suelta el bloque “Sensor Rastreador de Línea” en la sección “valor” de la variable digital “IRPin”. Asigna el pin 4 al bloque “Sensor Rastreador de Línea”.



Debido a un detalle con la librería del sensor, es necesario agregar otra variable digital de nombre “tooClose” y en valor, agrega el pin digital número 4 para obtener correctamente el valor deseado.



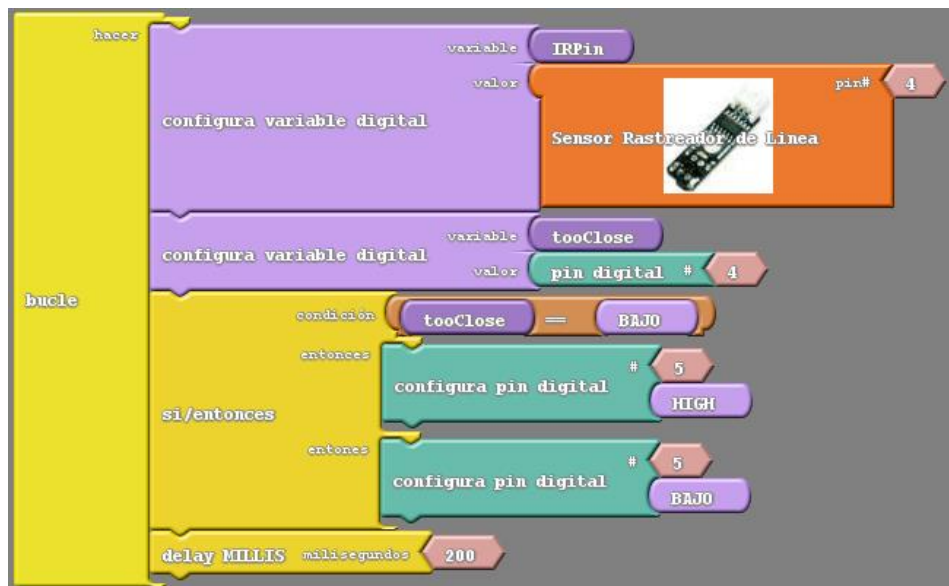
Agrega un bloque “si/entonces”, en la condición deberás comprar la variable “tooClose” con un nivel 0, voltaje 0, o simplemente el número 0. En este ejemplo lo comparamos con “Bajo”.

La primera sentencia “entonces” ejecutará el bloque para encender el *LED* ubicado en el pin 5 y la segunda sentencia “entonces” ejecutará el bloque para apagar el *LED*.

Para finalizar, agrega un retraso de 200 milisegundos

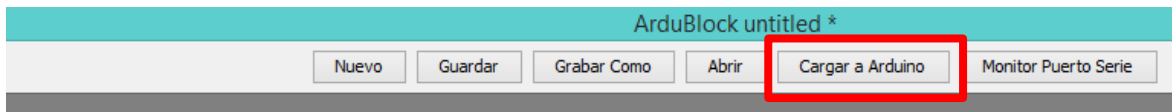


El programa debe quedar de la siguiente manera:



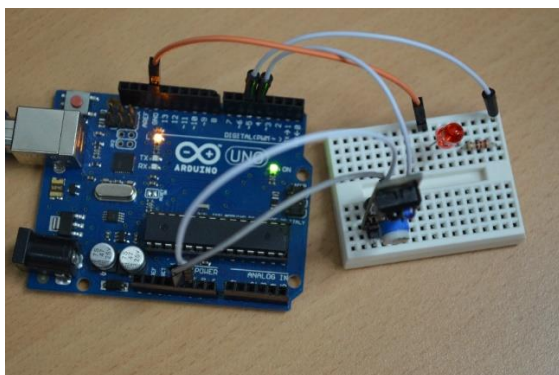
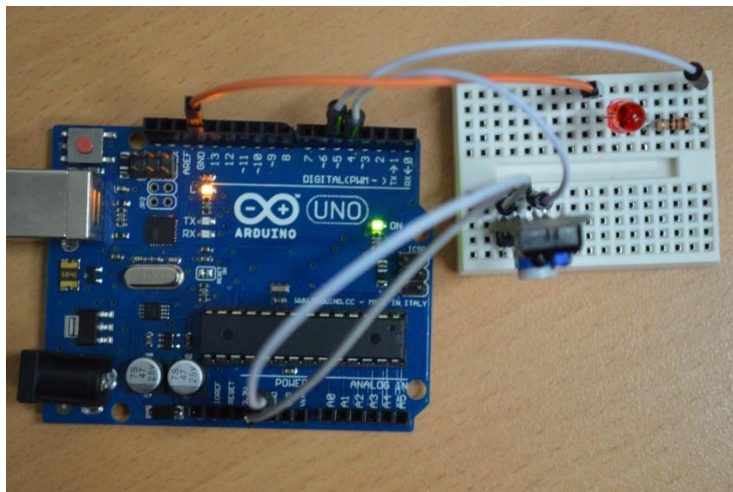
Puedes encontrar este programa en la carpeta:
KitRoverEducativo\Practicas\IR\IR.abp

Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.

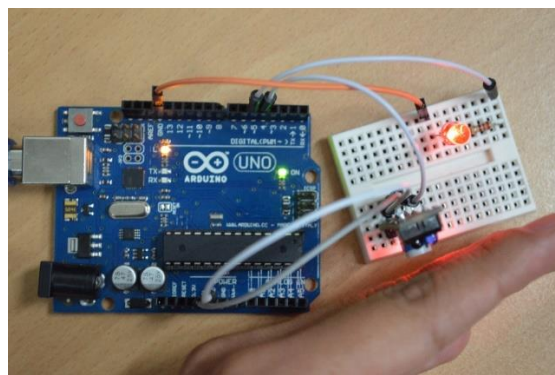


Espera a que se complete el proceso y observa los resultados.

Resultados



El LED se encuentra apagado debido a que no ha recibido una señal por parte del sensor IR



El LED se enciende una vez que recibe una señal por parte del sensor IR, esta señal se obtiene cuando se encuentra algún obstáculo enfrente de sensor.

Para probarlo, utiliza una hoja de papel blanca o algún objeto de tono claro. Posteriormente utiliza una hoja de papel negro o algún objeto del mismo tono.

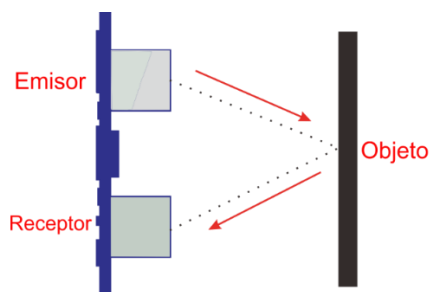
Prueba colocando el objeto o la hoja de papel más cerca, más lejos, inclinada ¿Qué sucede? ¿Funciona?

5 Sensor Ultrasónico

Introducción

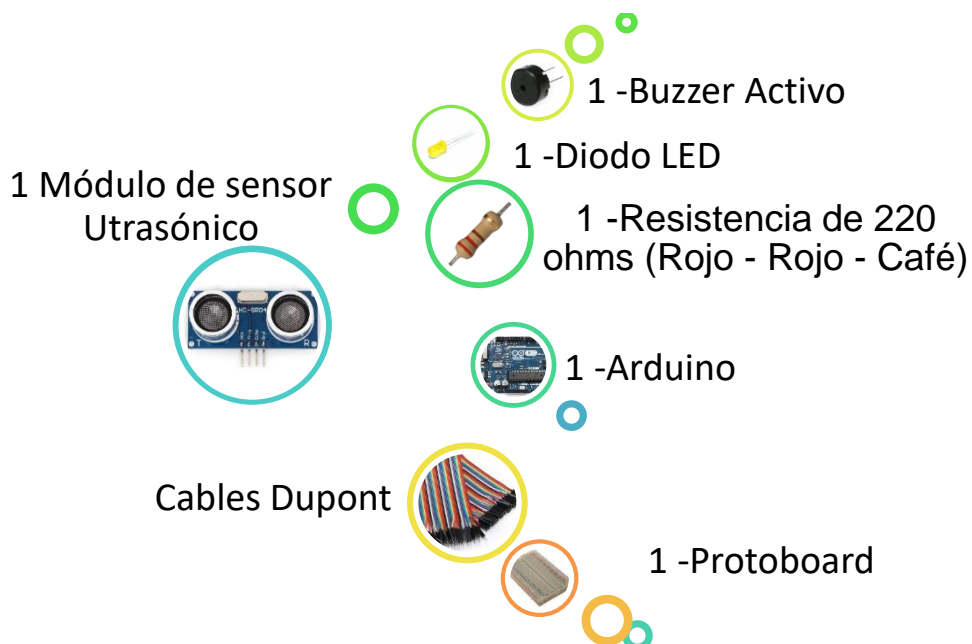
En esta práctica aprenderás a usar el sensor ultrasónico. Este sensor mide distancias a objetos cercanos al sensor. Esta medición es útil para evitar obstáculos, detener procesos, o realizar alguna acción pertinente.

El sensor ultrasónico tiene similitudes con el sensor IR, pues ambos emplean un emisor y un receptor. En el caso del sensor empleado en esta práctica, no utiliza ondas infrarrojas, sino ondas acústicas que no son audibles por el oído humano. Estas ondas acústicas salen del emisor y cuando se topan con un obstáculo, rebotan de regreso al sensor y son captadas por el receptor.



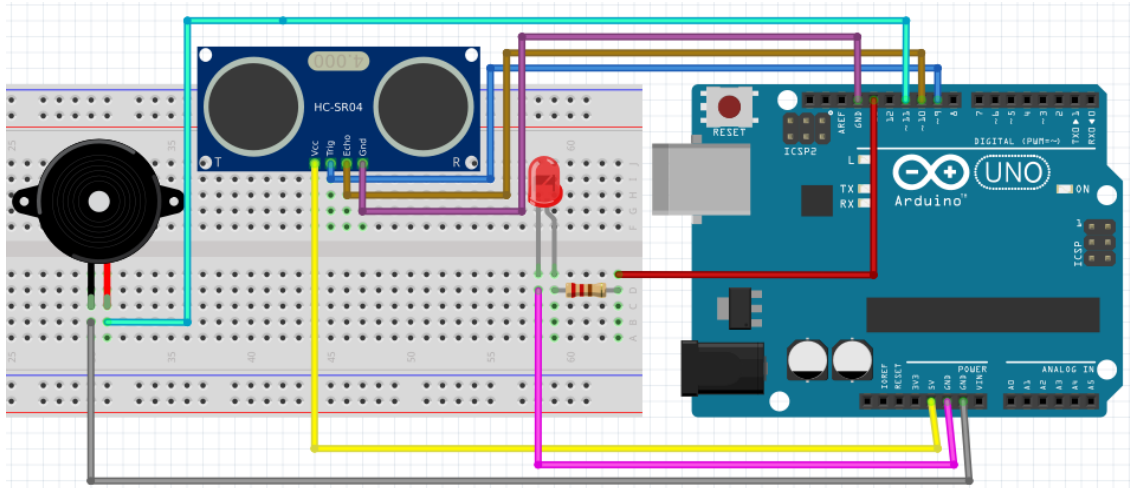
Mediante un cálculo matemático, el cual no será abordado en este manual, el sensor calcula la distancia al objeto y la transmite a la placa *Arduino*.

Materiales



Circuito

Construye el siguiente circuito:



Este circuito incluye un sensor y dos indicadores, un LED y un Buzzer (Zumbador).

El **buzzer** es un dispositivo que emite un sonido cuando se le suministra una corriente eléctrica. Existen dos tipos de **buzzer**, **activo** y **pasivo**: el **buzzer activo** es **digital**, y el **buzzer pasivo** es **analógico**. En esta práctica utilizaremos un **buzzer activo**, el cual al recibir un “1” o un voltaje de 5 [V] emitirá un sonido.

Dependiendo del modelo de **buzzer** las terminales pueden o no tener cables y estos pueden o no estar de color rojo y negro. Algunos en la parte superior indican cual es la terminal positiva, y otros indican con el color rojo la terminal positiva. Conecta la terminal *positiva* o *roja* a el **pin digital 11**, y la terminal *negativa* o *negra* a **GND** o *tierra*.



Los módulos que contienen el sensor ultrasónico tienen 4 terminales. Conecta la terminal con etiqueta “VCC”, a 5[V], la terminal con etiqueta “GND”, a GND o tierra, La terminal con etiqueta “Trig” al pin digital 9, y la terminal con etiqueta “Echo” al pin digital 10.



Dependiendo del modelo, el **módulo de sensor ultrasónico** puede lucir diferente, algunos presentan un dispositivo llamado cristal, de color plateado, al frente y otros por detrás. Incluso algunos tienen las etiquetas de las terminales colocadas en orden diferente, por lo que es importante que cuando conectes las terminales pongas atención al nombre de la etiqueta.

Programa

Ejecuta el ambiente de desarrollo **ArduBlock**.

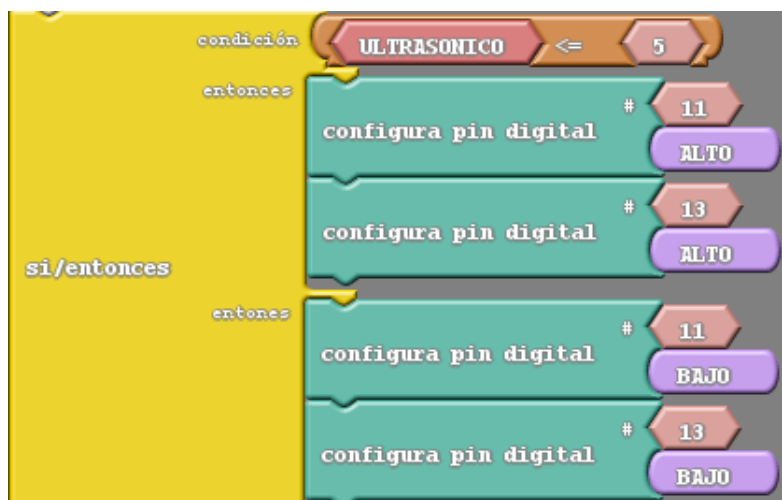
Ingresa un bloque de control “**Bucle hacer**”, el menú izquierdo clic en la opción **Variables/Constantes**, arrastra y suelta el bloque “**set integer variable**”, en la opción “**Variable**” escribe el nombre de esta “**ULTRASONICO**” y en la opción “**valor**” agrega el sensor ultrasónico. En el menú izquierdo, clic en la opción “**Generic Hardware**”, arrastra y suelta el bloque “**ultrasonico**”. En la opción “**trigger**” escribe el número del **pin 9**, y en la opción “**echo**” escribe el número del **pin 10**.



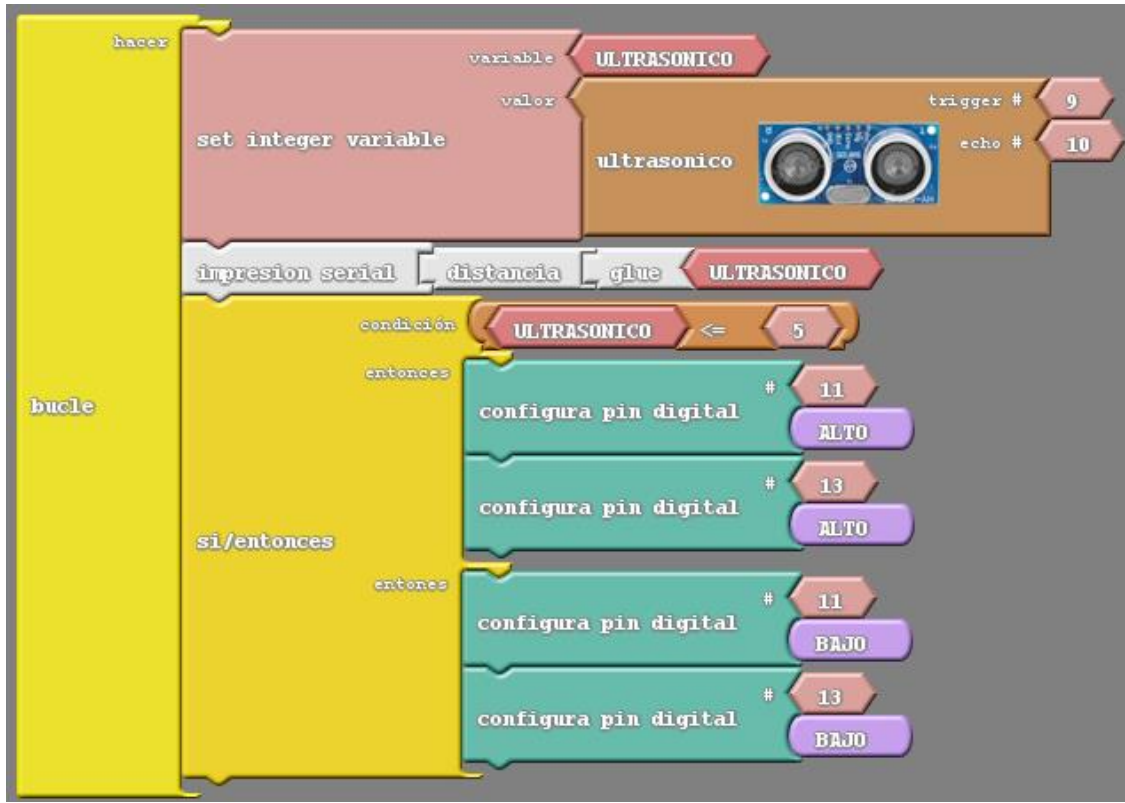
A continuación, agrega un módulo de comunicación serial. Agrega el nombre de la variable como “**distancia**”.



Después, agrega un bloque “**si/entonces**”, en la opción “**Condición**” compara el valor de la variable “**ULTRASONICO**” para cuando esta sea menor o igual al valor 5. En la primera sentencia “**entonces**” agrega los bloques para encender el LED y el buzzer con pines digitales en “**ALTO**” y en la segunda sentencia “**entonces**”, agrega los bloques para apagar el LED y el buzzer con pines digitales en “**BAJO**”.

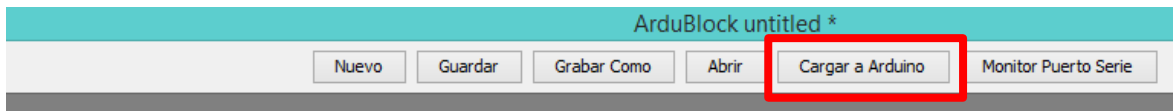


El programa debe quedar de la siguiente manera:



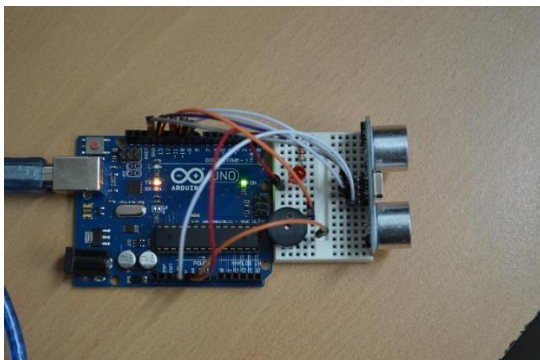
Puedes encontrar este programa en la carpeta:
KitRoverEducativo\Practicas\ULTRASONICO\ultrasonico.abp

Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.

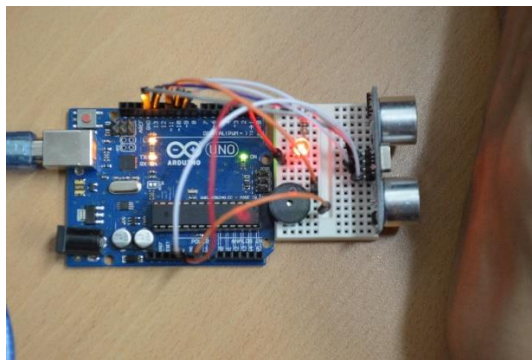


Espera a que se complete el proceso y observa los resultados.

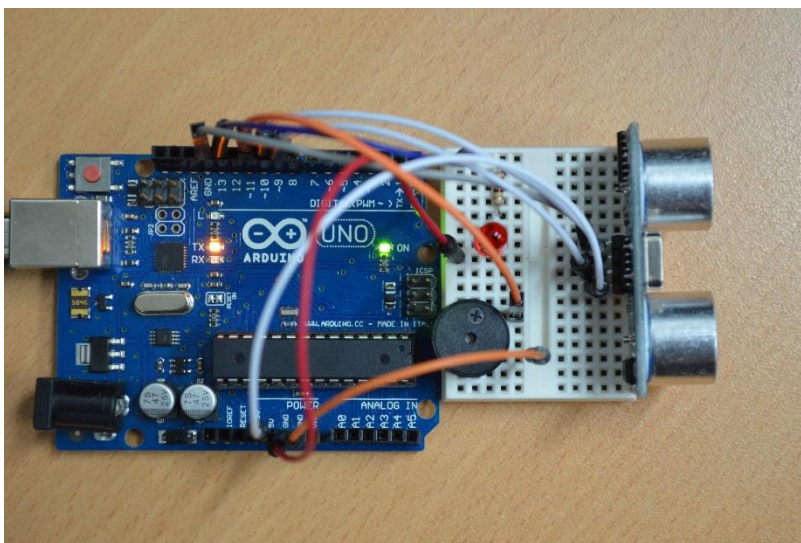
Resultados



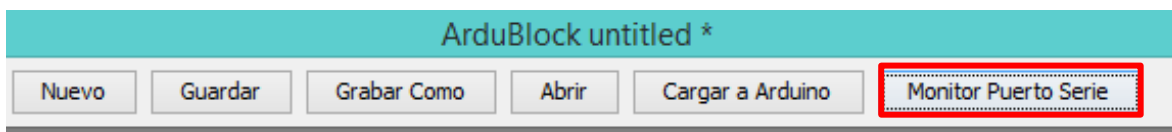
El sensor ultrasónico funciona continuamente midiendo la distancia a los objetos más cercanos.



Cuando un objeto se interpone a 5 cm o menos, el **buzzer** emitirá un sonido y el LED se encenderá.



Para observar los valores que arroja el módulo de sensor ultrasónico, abre el Monitor Serial. Clic en el botón “Monitor Puerto Serie”.



¿Qué aplicaciones puede tener este desarrollo? Intercambia el comportamiento del LED y del Buzzer.

6 Sensor Hall

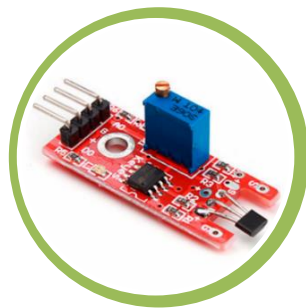
Introducción

En esta práctica utilizarás el **sensor de efecto hall**, el cual tiene como principio de funcionamiento el efecto hall. El sensor, por sí mismo es un interruptor cerrado, es decir, permite el paso de la corriente, cuando el sensor detecta un campo magnético cercano el interruptor se abre impidiendo el paso de la corriente eléctrica. El **módulo de sensor de efecto hall** aprovecha este funcionamiento para detectar campos magnéticos cercanos y mediante un LED indica la presencia de éste.

Cabe mencionar que solo detecta el polo negativo de un imán, por lo que cuando realices la prueba del sensor, debes colocar el polo negativo del imán cerca del sensor.

Materiales

1 -Módulo de sensor Hall



1 - Imán



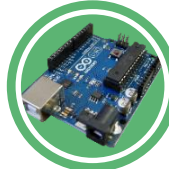
1 - Protoboard



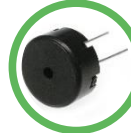
Cables Dupont



Arduino

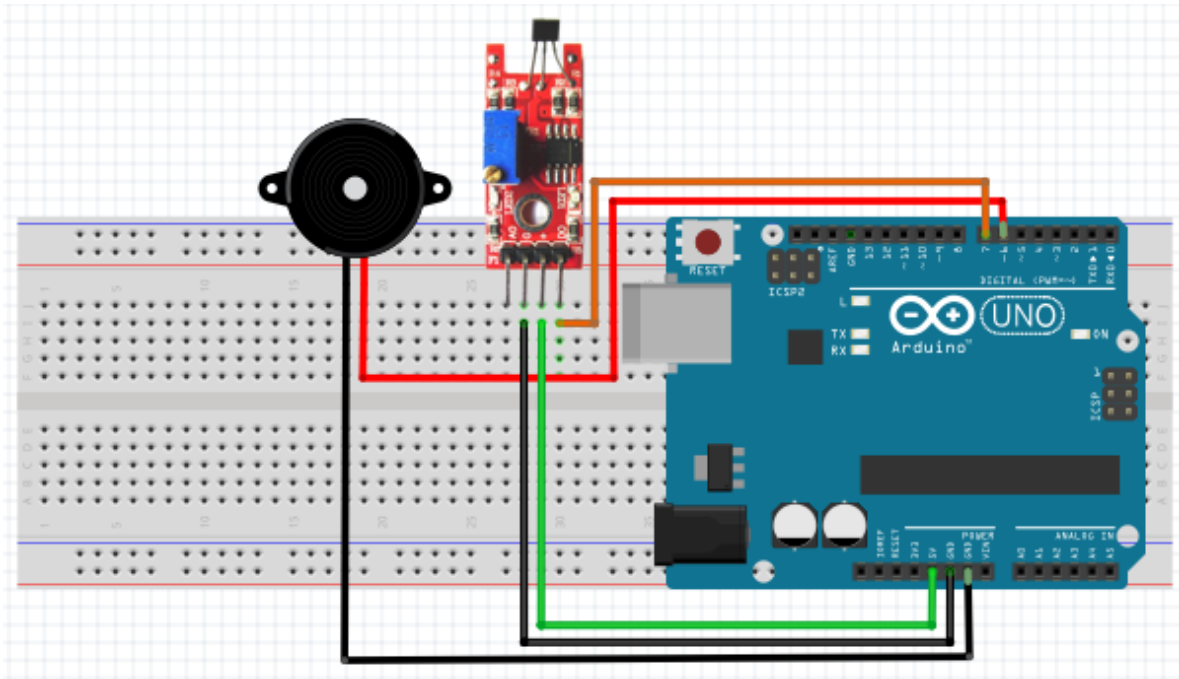


1 - Buzzer Activo



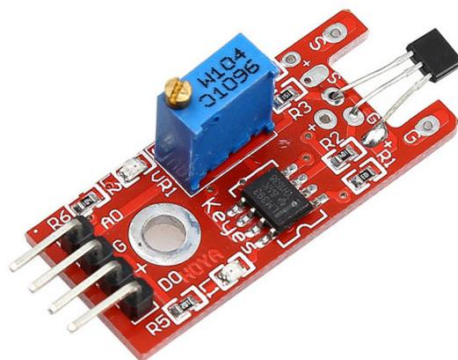
Circuito

Construye el siguiente circuito:



En el circuito se incluye nuevamente un **buzzer activo**, consulta la práctica anterior para cualquier duda sobre los tipos de **buzzer**. Conecta la terminal positiva al **pin digital 6** y la terminal negativa a **GND**.

El módulo de sensor hall viene en diferentes presentaciones. En esta práctica se ocupará el siguiente modelo:

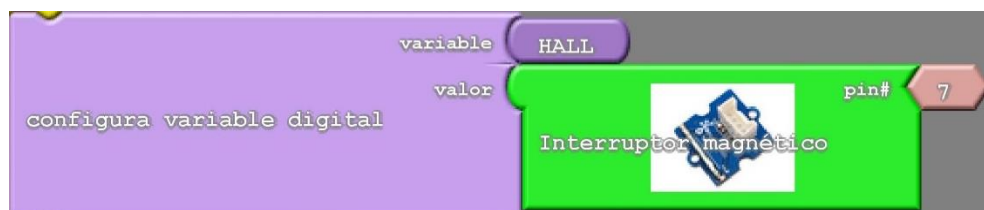


Programa

Ejecuta el ambiente de desarrollo **ArduBlock**.

Ingresa un bloque de control “**Bucle hacer**”, configura una variable digital con nombre “**HALL**”, en el valor de la variable ingresa el bloque “**Interruptor magnético**”, En el menú izquierdo, clic en la opción “**Speed Studio Grove**” y selecciona el bloque “**Interruptor magnético**”, en la opción pin, ingresa un 7.

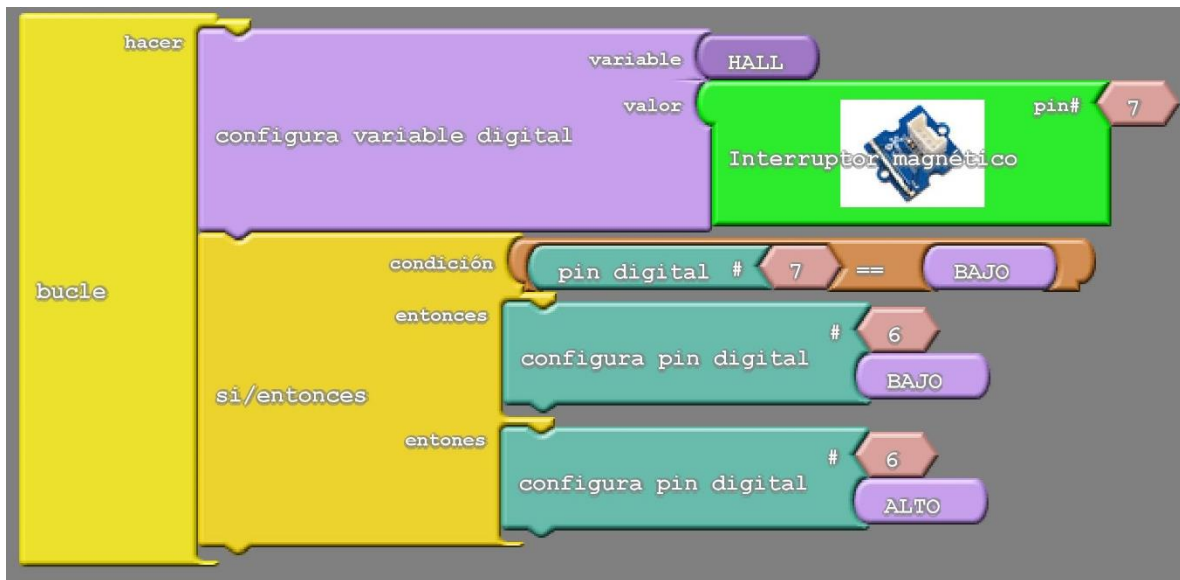
ArduBlock no cuenta como tal con el módulo del sensor hall, sin embargo el bloque “**Interruptor magnético**”, es una buena opción para controlarlo.



A continuación, ingresa un bloque “**si/entonces**”, en la opción de condición verifica que el **pin digital 7** sea igual al valor en “**BAJO**”, en la primera sentencia “**entonces**” configura el **pin digital 6** en “**BAJO**” y en la segunda sentencia “**entonces**” configura el mismo **pin digital 6** en “**ALTO**”.

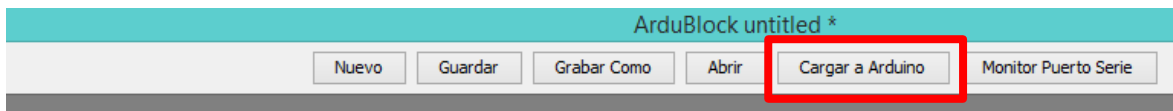


El programa debe quedar de la siguiente manera:



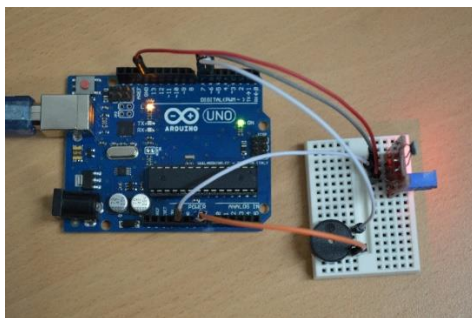
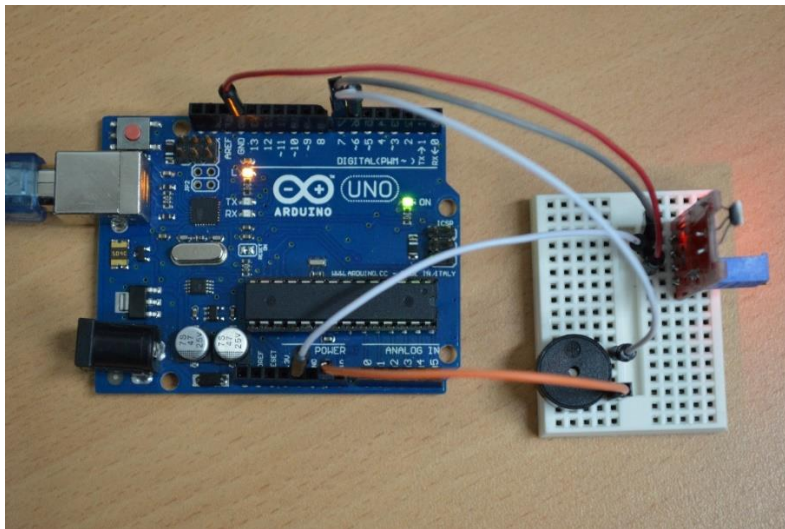
Puedes encontrar este programa en la carpeta:
KitRoverEducativo\Practicas\HALL\hall.abp

Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.

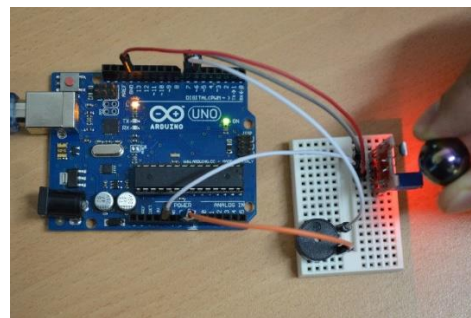


Espera a que se complete el proceso y observa los resultados.

Resultado



Cuando el sensor no detecta un imán u objeto magnético, este no genera ninguna señal.



Cuando el sensor detecta algún imán o algún objeto magnético se genera una señal y el buzzer se activa emitiendo un sonido.

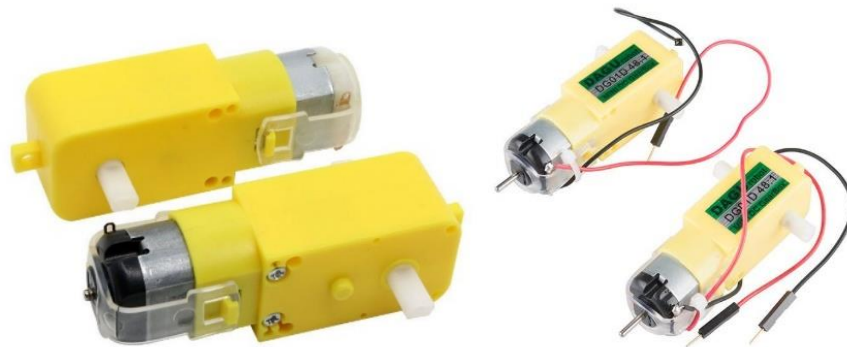
Con un pequeño desarmador plano, juega un poco con el potenciómetro (dispositivo azul con un tornillo) y gira el tornillo en alguna dirección. Esta acción cambiará la sensibilidad del sensor. Prueba con diferentes imanes y observa a que distancia los detecta.

7 Motores.

Introducción

El principal actuador para que un desarrollo tenga movilidad son los motores, estos se encuentran en diferentes presentaciones y tipos. Existen **Motores de Corriente Directa (DC)**, **Motores a pasos**, **Servomotores**, **Motores trifásicos**, etc.

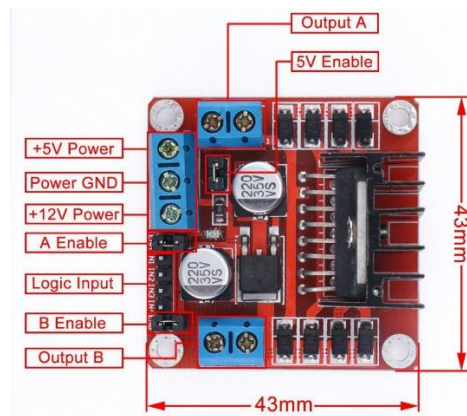
En esta práctica se ocuparán **Motores de DC** con reductor también conocidos como (**Motorreductores**), que son el tipo de motores más sencillos a los cuales solo se necesita agregar llantas. Para pruebas puedes usar cualquier tipo de llanta o amarrar un alambre al eje de los motorreductores para que percibas la correcta dirección de los motores.



La mayoría de estos motores los venden sin cables, por lo que deberás soldar un par de cables utilizando un cautín, soldadura y dos cables/alambres por cada motor. Te recomendamos utilizar cables dupont, cortar un extremo de estos y dicho extremo soldarlo a los motores, siempre es práctico utilizar cables de color negro y rojo. Estos motores no tienen como tal una polaridad, por lo que es indistinto cual terminal es la positiva y cuál es la negativa solo recuerda respetar el código de colores o las terminales durante tus desarrollos.

Finalmente, para poder controlar estos motores de manera adecuada, se necesita un controlador llamado “**Módulo de puente H**” para motores de DC, este permite al motor girar en ambos sentidos, avanzar y retroceder, así como para frenarlo de manera brusca.

Hay diversos módulos de puente H, el que se empleará en esta práctica tiene la siguiente configuración:

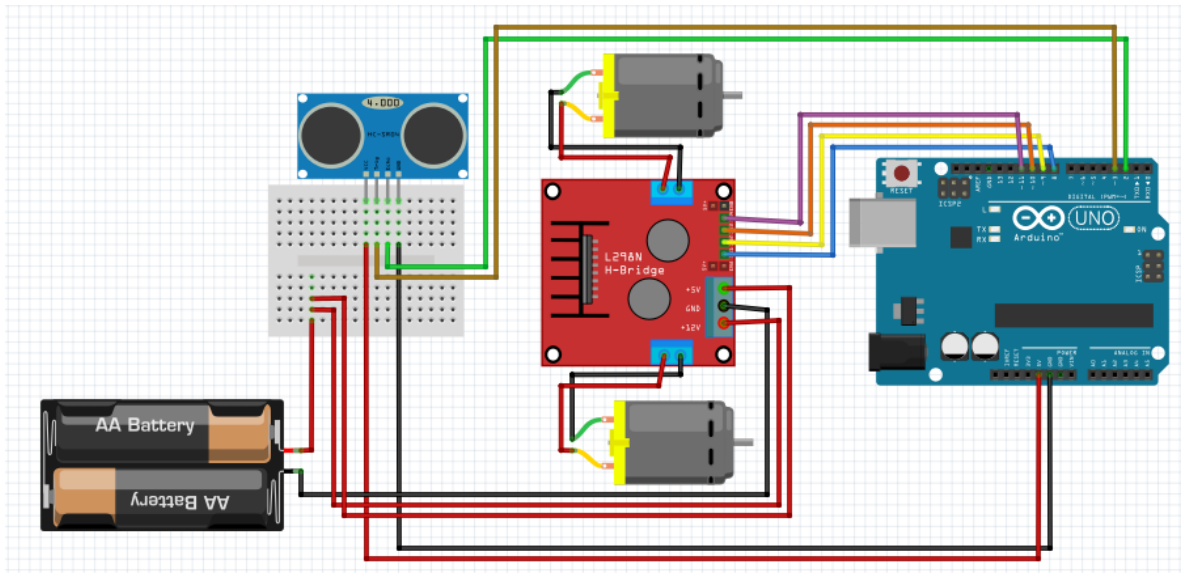


Materiales



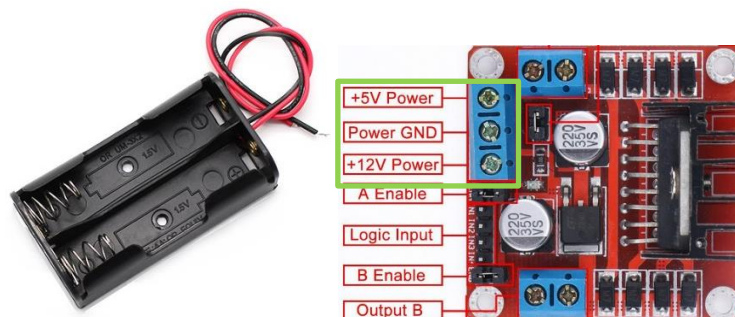
Circuito

Construye el siguiente circuito:



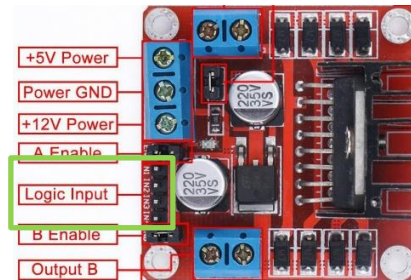
El sensor ultrasónico empleado en esta práctica es el mismo que se emplea en la práctica 5. Conecta las terminales con etiqueta VCC y GND respectivamente a 5[V] y Tierra o GND. La terminal con etiqueta “Trigger” al pin digital 3 y la terminal con etiqueta “Echo” al pin digital 2.

Los porta pilas vienen en diferentes presentaciones, busca uno con alambres/cables previamente soldados, de lo contrario deberás soldarlos también. Cerciórate de que el porta pilas sea para pilas AA.

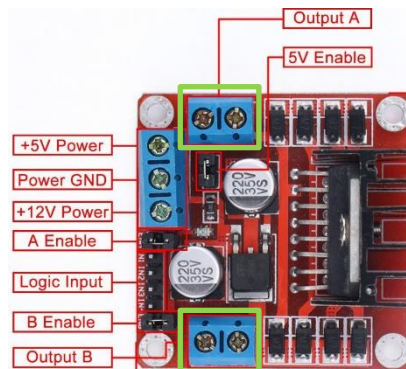


Conecta la terminal **positiva** (roja) al pin de +12[V] y la terminal **negativa** (negra) al pin GND del módulo de puente H.

Del puente H, conecta la terminal con etiqueta “IN1” al pin digital 8, la terminal con etiqueta “IN2” al pin digital 9, la terminal con etiqueta “IN3” al pin digital 10, la terminal con etiqueta “IN4” al pin digital 11:



Finalmente, conecta los motores a las terminales “Output A” y “Output B”, en la terminal con etiqueta “OUT 1” conecta la terminal roja de un motorreductor, en la terminal con etiqueta “OUT 2” la terminal negra del mismo motorreductor. Conecta la terminal con etiqueta “OUT 3” a la terminal negra del segundo motorreductor y la terminal con etiqueta “OUT 4” a la terminal roja del mismo motorreductor.



Programa

Ejecuta el ambiente de desarrollo ArduBlock.

Ingresa un bloque de control “Bucle hacer”, agrega un bloque “set integer variable”, en la opción “Variable” escribe el nombre “Ultrasonic” y en la opción “valor” agrega el sensor ultrasónico. En la opción “trigger” escribe el número del pin 3, y en la opción “echo” escribe el número del pin 2.



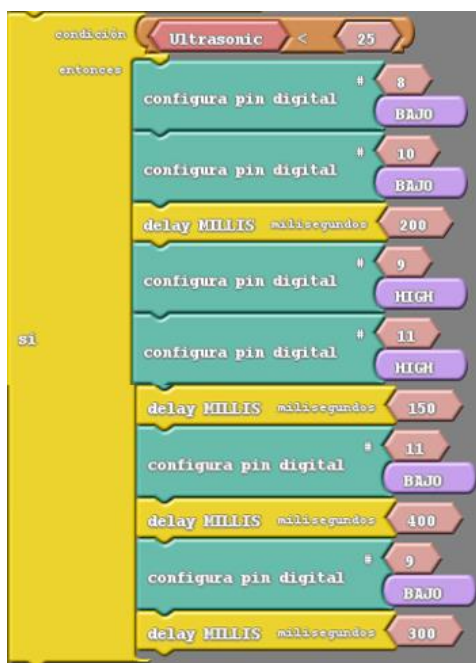
Agrega un bloque de impresión serial asociado a la distancia del sensor ultrasónico:



Agrega dos bloques de configuración para los **pin digital 8 y 10** con un estado en “alto” o “HIGH”. Este bloque hará que los motores avancen en una dirección (frente) desde que se enciende el desarrollo. Al final agrega un retraso de **200 milisegundos**.

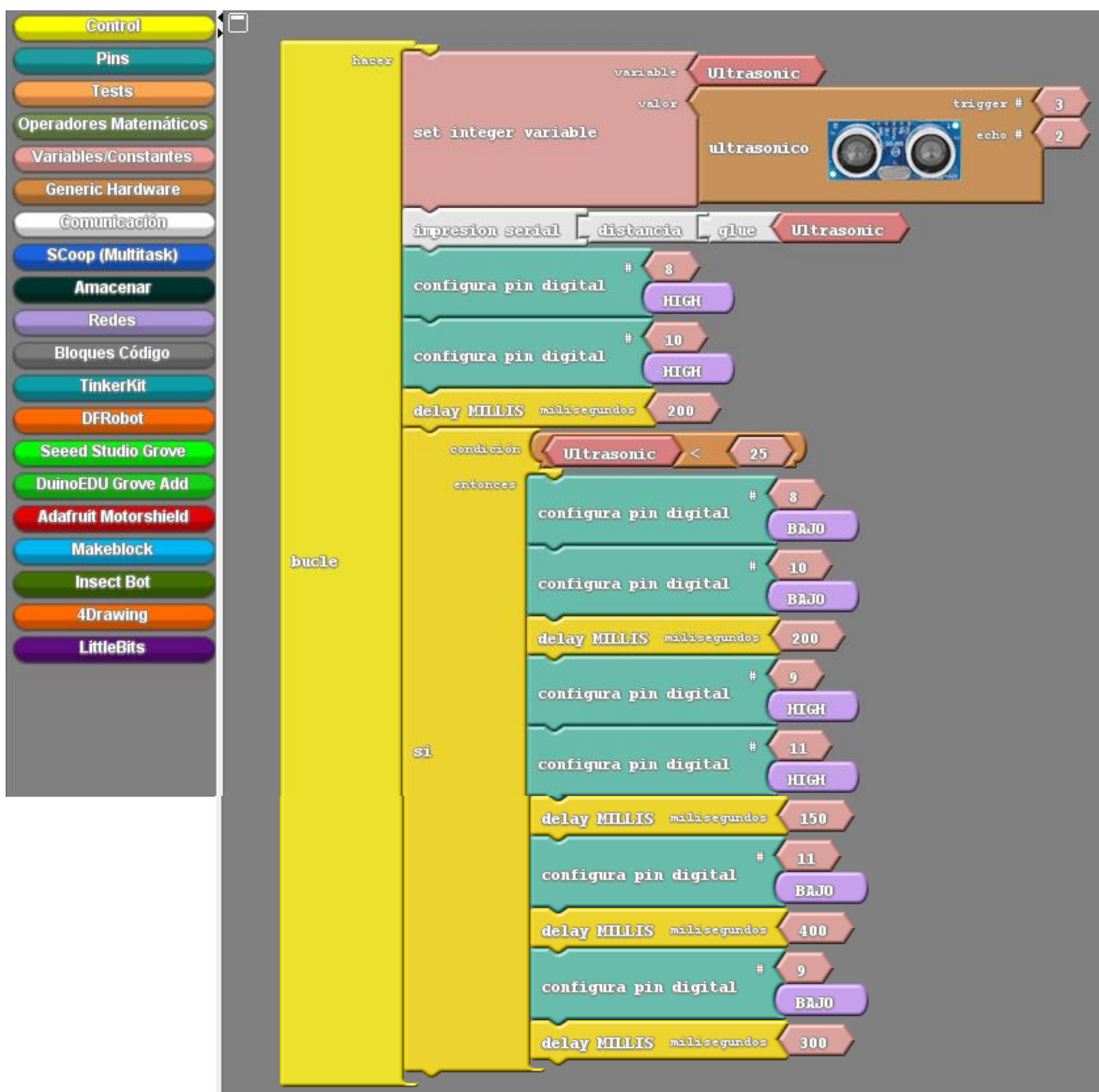


A continuación, ingresa un bloque “si”. En el menú izquierdo, clic en la opción “Control” y arrastra y suelta el bloque “Si”. Nota que este bloque únicamente cuenta con una sola sentencia “Entonces”, en la opción de condición verifica que la variable “Ultrasonic” sea menor que 25, en la sentencia “entonces” configura el **pin digital 8 y 10** en “BAJO”,



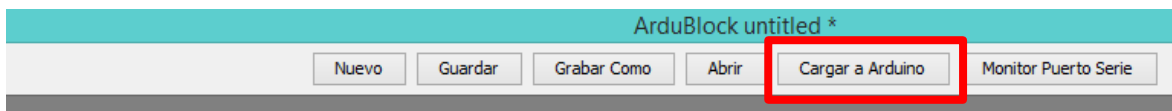
agrega un retraso de **200 milisegundos** y configura el **pin digital 9 y 11** en “Alto” o “HIGH”, a continuación, agrega otro retraso de **150 milisegundos**. Hasta aquí lo que hace el programa es detener primero los motores y luego los activa en “reversa”. Siguiendo con el programa, configura el **pin digital 11** en “BAJO”, agrega un retraso adicional de **400 milisegundos**, finalmente, configura el **pin digital 9** en “BAJO” y agrega un retraso adicional de **300 milisegundos**, estos bloques harán que un motor se detenga mientras el otro continúa moviéndose en “reversa” y luego también detiene ese motor. Al salir de la condición, volverá a configurar los motores para que estos se muevan en la primera dirección configurada o “hacia el frente”

El programa debe quedar de la siguiente manera:



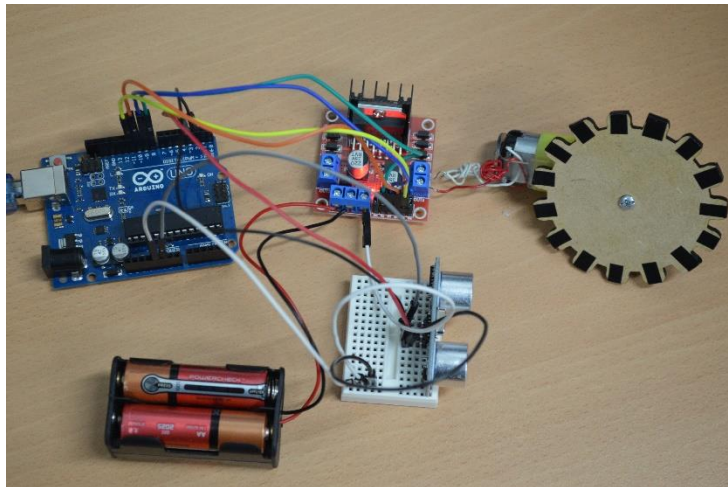
Puedes encontrar este programa en la carpeta:
KitRoverEducativo\Practicas\MOTORES\MOTORES.abp

Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.

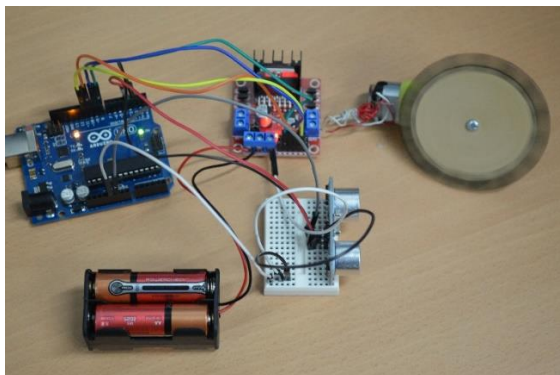


Espera a que se complete el proceso y observa los resultados.

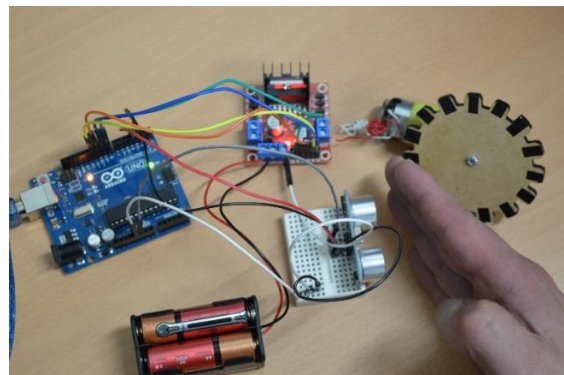
Resultados



Debido a que los motores no cuentan con ruedas, para apreciar el movimiento de los motores es necesario que agregues ya sea, unas ruedas o bien, puedes amarrar unos pequeños tramos de alambre o cable.



Cuando el sensor ultrasónico no detecta algún objeto u obstáculo a menos de 25 cm, los motores giran con “sentido hacia el frente”



Cuando el sensor ultrasónico detecta algún objeto u obstáculo a menos de 25 cm, Los motores ejecutan un breve movimiento en reversa y gira levente para “esquivar” objeto

Prueba agregar esta práctica a alguna base, pega los motores por debajo de la base y por encima coloca el Arduino, las pilas y el módulo de puente H, agrega algunos obstáculos para observar que sucede.

Modifica los tiempos de retraso en el bloque condicional “si”, Con menor o mayor tiempo, ¿Qué ocurre?, ¿Quedará en la misma posición?, ¿Logrará esquivar los obstáculos?

Proyecto: Rover educativo

En este apartado, integrarás todos los conocimientos y materiales empleados en las prácticas anteriores.

Los sensores, y sus funciones, empleados en este proyecto son los siguientes:

- **Sensor de temperatura:** Este sensor medirá la temperatura ambiente, cuando el sensor detecte temperaturas mayores a los 51[C] un indicador LED se encenderá.
- **Sensor LDR:** Este sensor medirá la cantidad de luz en el ambiente, y en caso de ausencia de luz, se encenderán dos LED que se ubicarán en la parte frontal del Rover.
- **Sensor ultrasónico:** Este sensor medirá la distancia a objetos cercanos e indicará cuando se encuentre próximos a ellos para que el Rover pueda esquivarlos.
- **Sensor de efecto hall:** Ese sensor detectará el campo magnético en algún objeto cercano, si lo encuentra emitirá un sonido y esquivará el objeto
- **Sensor IR:** Este sensor identificará un terreno irregular, esta detección servirá para dar instrucción al Rover de detenerse y comenzar a detectar la temperatura ambiente. Si detecta que la temperatura ambiente es mayor a 51 [C] el Rover reanudará la marcha esquivando la zona caliente.

Materiales

En este proyecto se ocuparán varios dispositivos electrónicos que servirán de apoyo para los sensores:

- 1 Arduino Uno
- 1 Mini protoboard
- 1 Sensor de temperatura LM35
- 1 Sensor LDR
- 1 Sensor Ultrasónico HC-SR04
- 1 Sensor de Efecto Hall
- 1 Sensor IR KY-033
- 1 Módulo de puente H L298N
- 2 Motorreductores
- 2 LED ultrabrillantes blancos
- 1 LED rojo
- 1 Buzzer activo
- 25 Cables Dupont macho-hembra
- 11 Cables Dupont macho-macho
- 1 Rueda loca
- 1 Porta pilas para 4 pilas AA o Cable USB Extra
- 4 Pilas AA o batería de respaldo para celular.
- 1 Desarmador plano o cruz según el tornillo
- 8 Tornillos Allen de 1/8
- Tuercas para tornillos Allen de 1/8
- Cable cargador de celular
- Silicón o pegamento blanco.
- Pilas AA o Batería portátil externa
- 1 Bornera de 2 terminales
- 1 Resistencia de 220 Ohm
- 1 Resistencia de 10K Ohm
- 1.5 [m] de Thermofit de ¼"

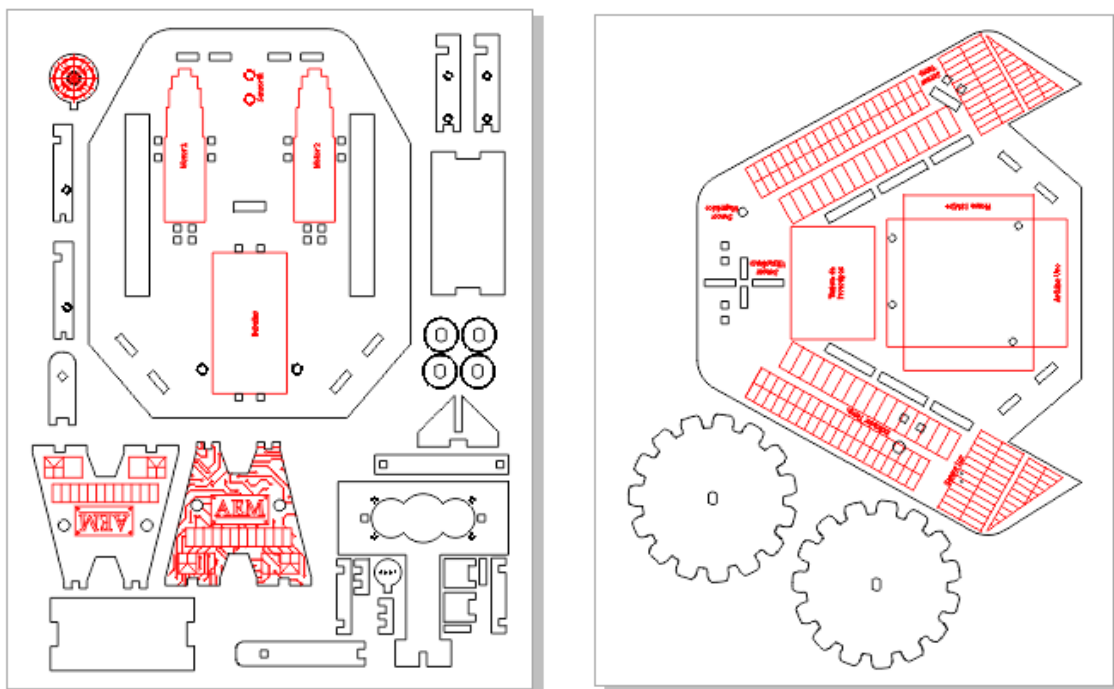
Estructura Rover

En la carpeta **KitRoverEducativo\Rover\Estructura** encontrarás los archivos que te ayudarán a generar la estructura del Rover Educativo. Dependiendo de la herramienta que desees utilizar, hay 2 carpetas. La carpeta de nombre “**CorteLaser**” contiene archivos compatibles con el software AutoCad, son editables y por lo general son los archivos que solicitan en los locales dedicados al corte laser. Te recomendamos utilizar **MDF de 3mm**. Si decides trabajar con esta opción te recomendamos fijar la madera con pegamento blanco o silicón.

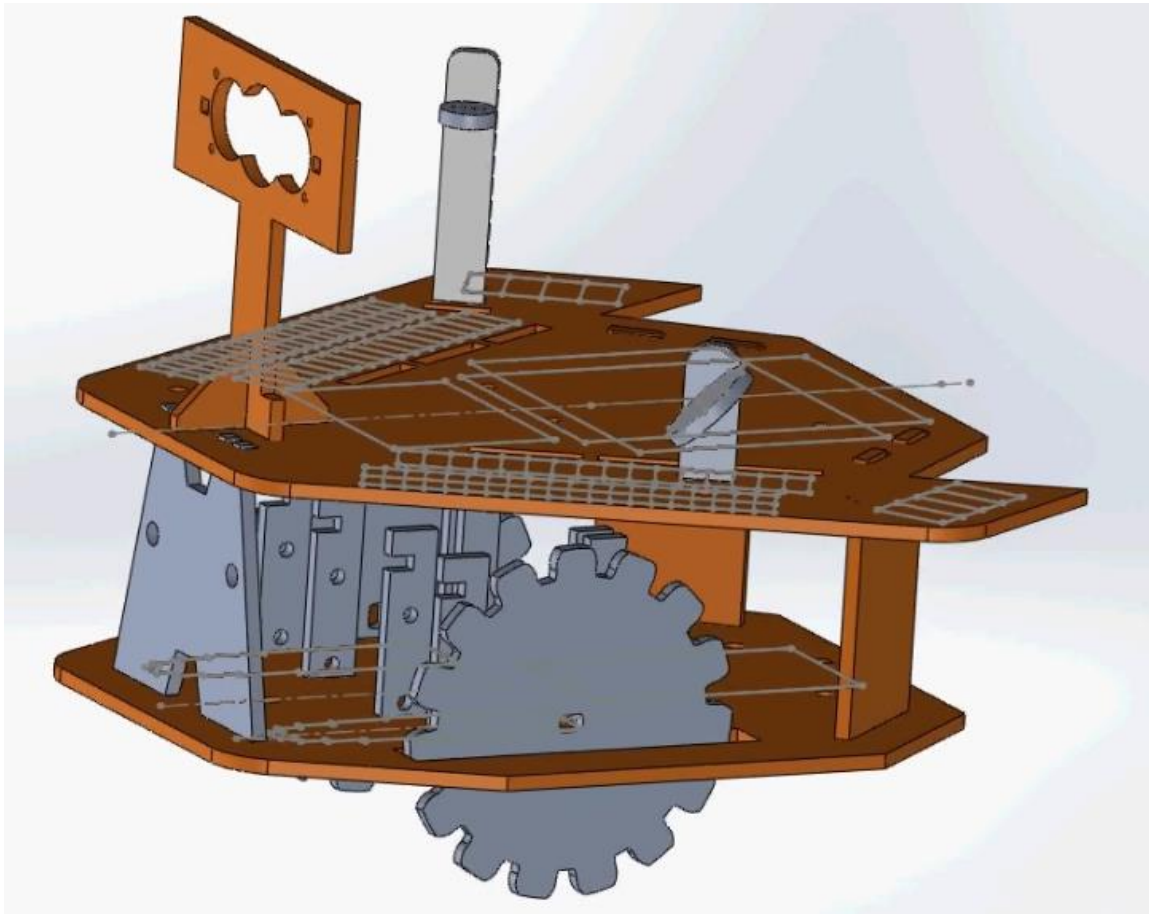
La carpeta con nombre “**3D**” contiene los archivos de modelado en 3D compatibles con el software de diversas impresoras 3D, se recomienda utilizar **PLA**. Si decides trabajar con esta opción te recomendamos fijar la estructura con silicón.

Finalmente, la carpeta de nombre “**SolidWorks**” contiene los archivos editables en modelado 3D compatibles con el software “**Solid Works**”, ocupa estos archivos en caso de que necesites realizar algún ajuste o que quieras mejorar o cambiar el diseño.

El software Auto CAD y Solid Works no vienen incluidos en el Kit de desarrollo del Rover Educativo.



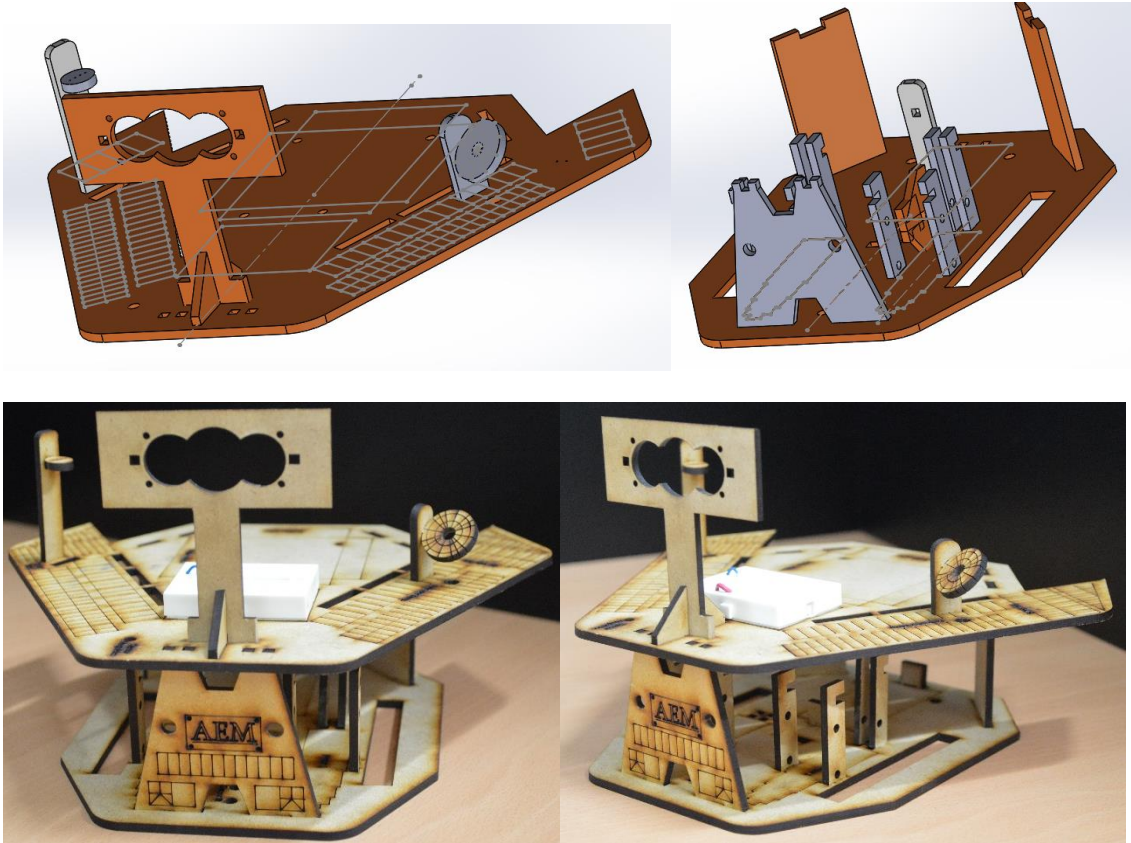
Planos de Corte Laser



Estructura Armada

Ubicación de los dispositivos electrónicos

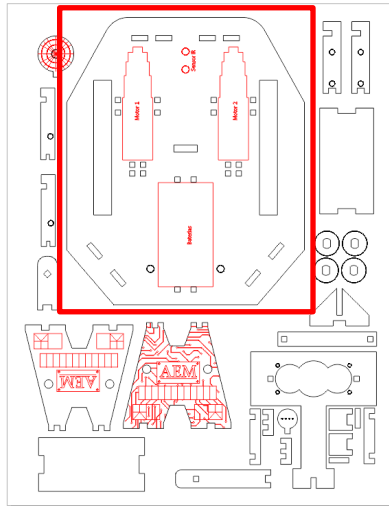
El diseño propuesto del proyecto Rover Educativo, tendrá la siguiente estructura:



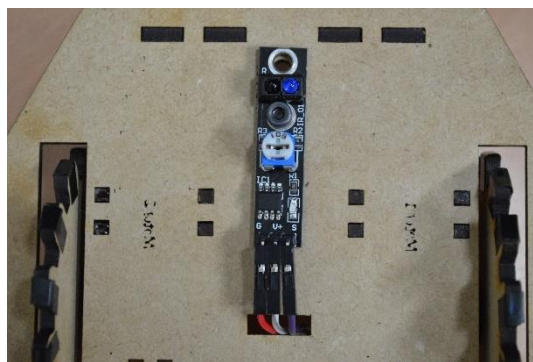
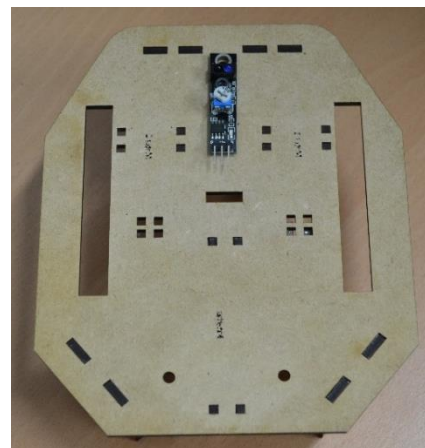
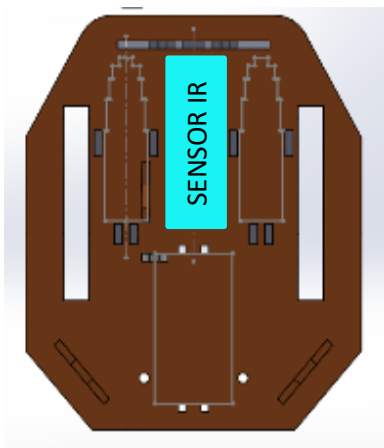
Cada parte del Rover ésta diseñada para colocar cada uno de los sensores y los demás correspondientes dispositivos electrónicos. Para facilitar las conexiones te recomendamos colocar todos los dispositivos que se ubicaran en la parte de abajo.

Sensor IR

El **sensor IR** se ubica en la parte de abajo del Rover. Para fijar el sensor utiliza un tornillo Allen de 1/8. Orienta los LED emisor y receptor hacia la parte frontal del Rover. Utiliza las imágenes para guiarte:



Parte baja del Rover.

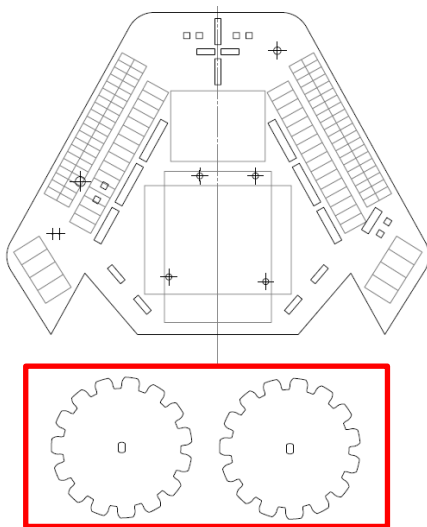


Vista Inferior de la parte baja del Rover

Vista inferior de la parte baja del Rover

Motorreductores y llantas

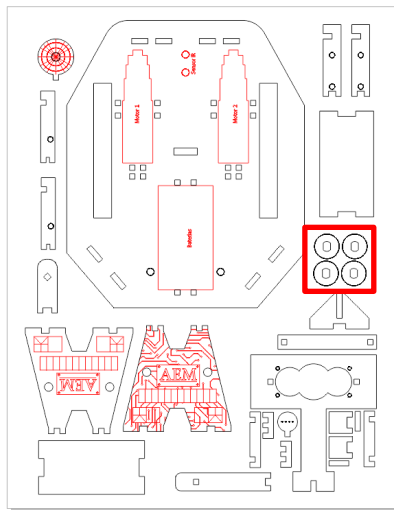
Coloca una “rueda loca” y atornilla las llantas.



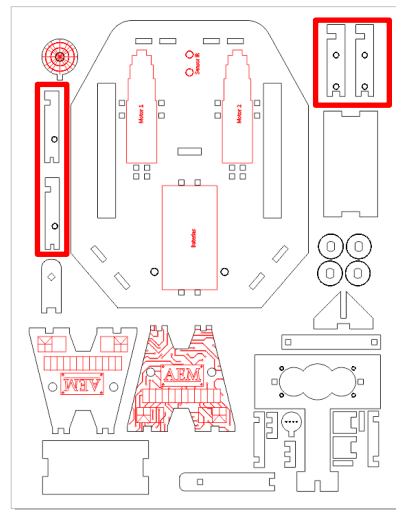
Ruedas



Rueda loca usada en este proyecto

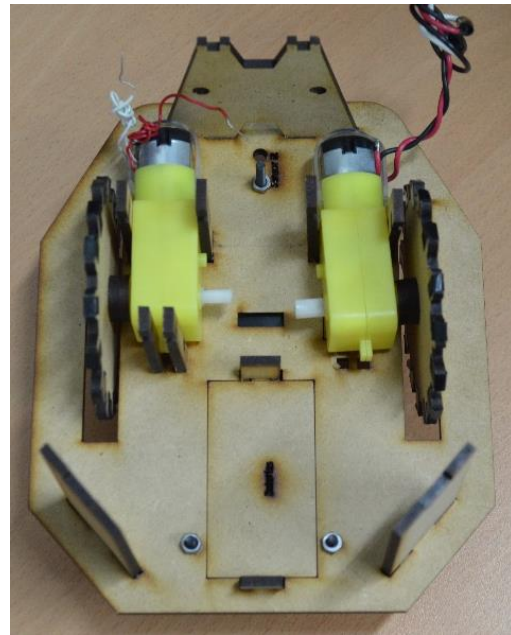
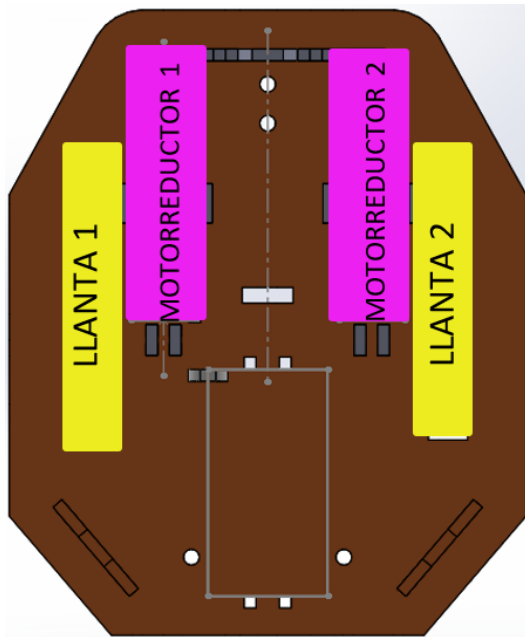


Soportes de eje de las ruedas



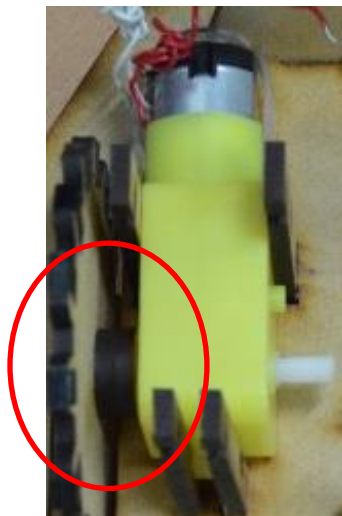
Soportes de los motorreductores

Observa la ubicación y orientación de los componentes:

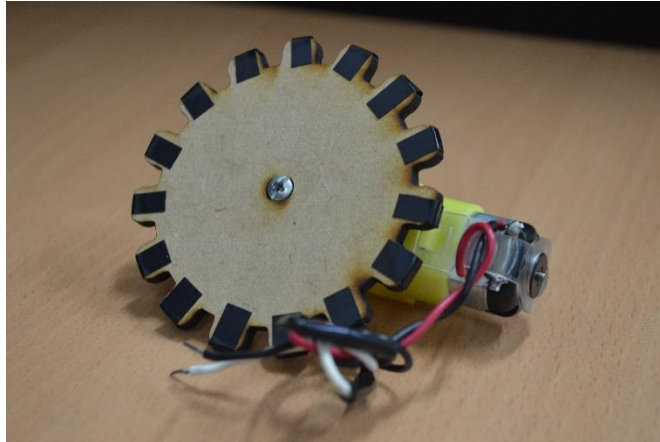


Vista superior de la parte inferior de la estructura

La parte “plateada” de los motorreductores, deberá estar orientada con la parte frontal del Rover. Toma dos soportes de eje de las ruedas, insértalos en el eje del motorreductor, inserta la rueda y atorníllala usando un 1 tornillo de 1/8 de pulgada, repite el proceso con el segundo motorreductor. Los tornillos para la rueda loca dependerán del modelo que hayas conseguido

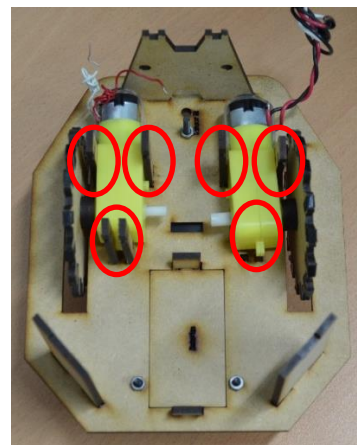
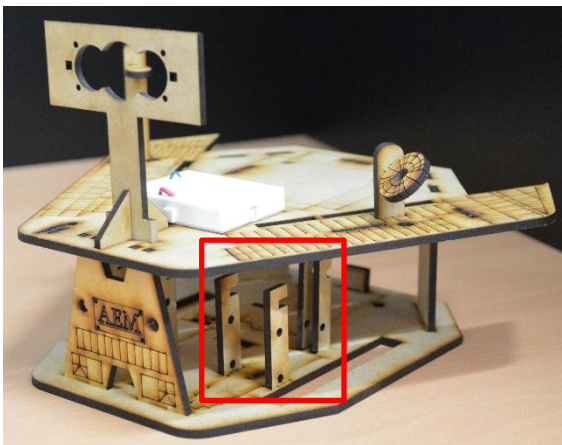


Motorreductores acoplados a las ruedas

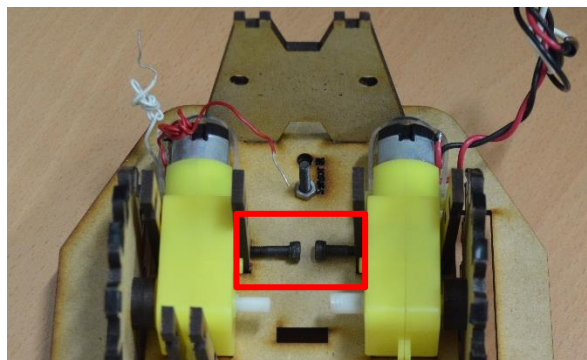


Rueda acoplada al motorreductor

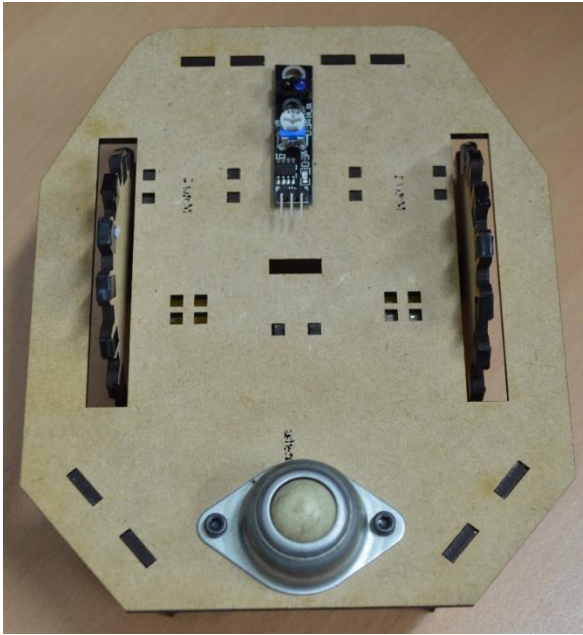
Para acoplar los motorreductores con las llantas a la parte baja del Rover, deberás colocar los soportes de los motorreductores,



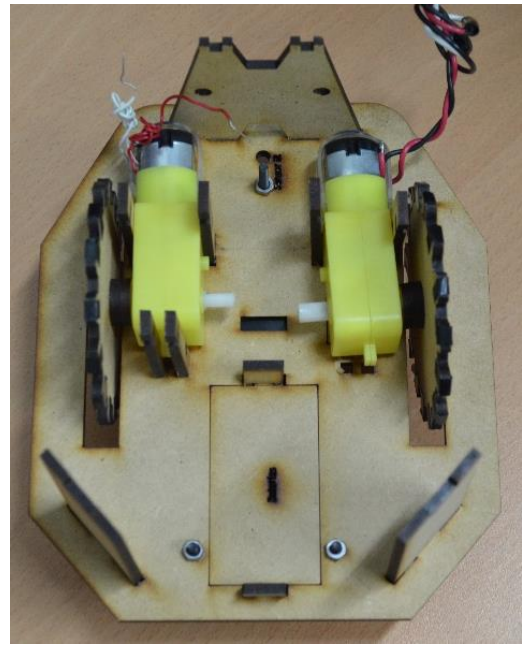
Fija los motorreductores con tornillos Allen de 1/8



Los elementos deberán quedar de la siguiente manera:



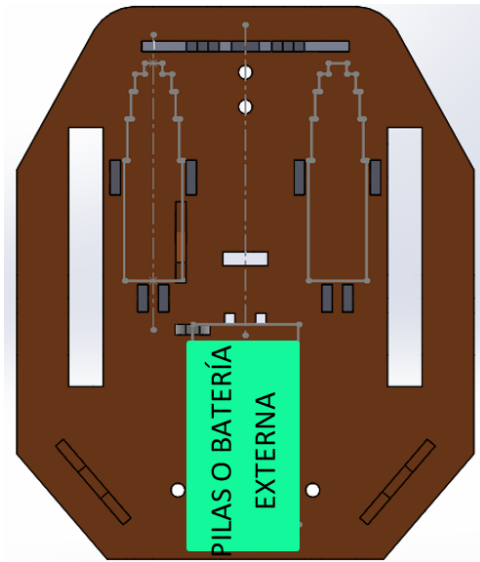
Vista inferior de la parte baja del Rover



Vista Superior de la parte baja del Rover

Pilas o Batería Externa

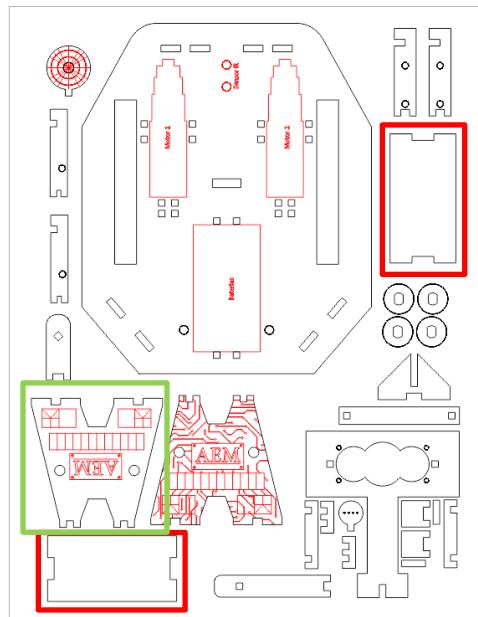
Durante la elaboración de este manual, se probó el Rover Educativo con dos fuentes de energía externas, 4 pilas AA y una batería de respaldo para celular (5[V] 2[A]). Con ambas funciona correctamente, pero dependiendo de que tengas a la mano puedes sustituir una fuente por otra, aunque recomendamos utilizar la batería de respaldo por la ventaja de ser recargable. Observa las siguientes imágenes para conocer la ubicación de la fuente de energía:



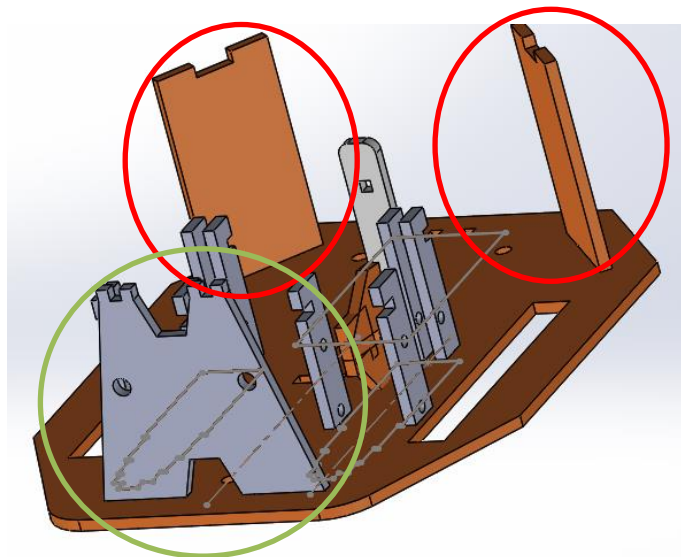
Vista superior de la parte baja del Rover

Leds Ultrabrillantes (LDR) y Soportes

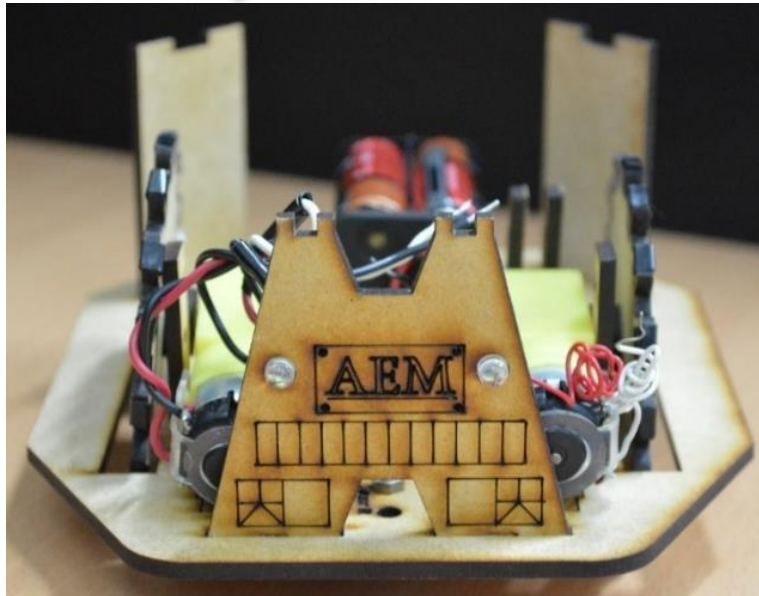
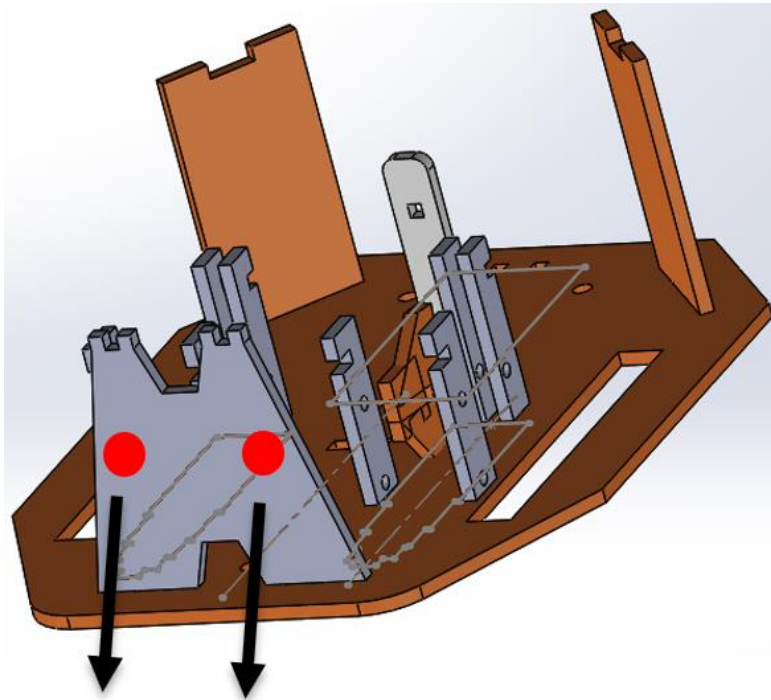
Coloca los soportes traseros y el soporte frontal del Rover, a este último, añade 2 LED ultrabrillantes.



Soportes traseros y frontal



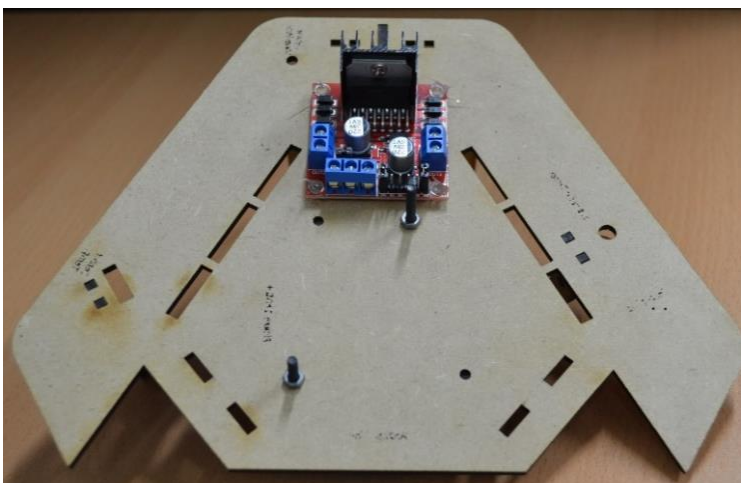
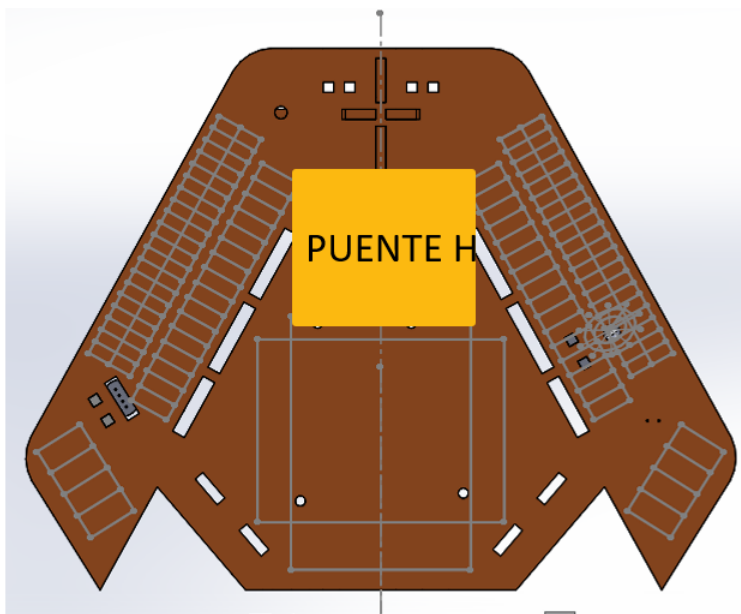
Soportes Traseros (rojo) Soporte frontal (verde)



En el soporte frontal, por la parte de atrás, coloca los LED de modo que la emisión de luz sea hacia el frente. Añade dos cables dupont en cada terminal del LED, te sugerimos usar un cable de color rojo para la terminal positiva y un cable de color negro para la terminal negativa.

Módulo Puente H (L298N)

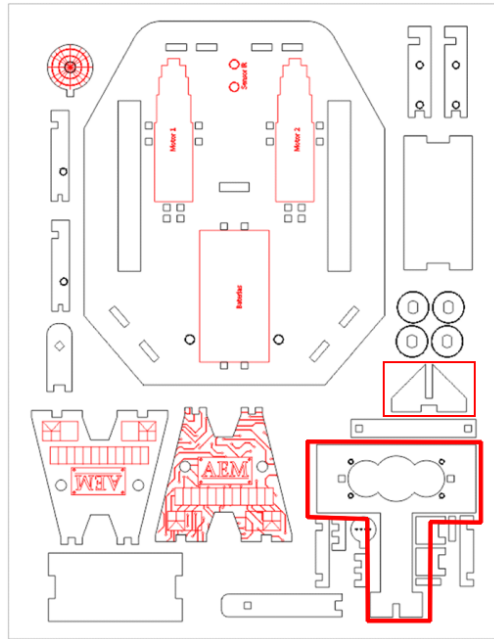
Debido a la distribución de los elementos en el Rover, se sugiere colocar el módulo puente H por debajo de la parte superior del Rover. Te sugerimos fijarlo con cinta doble cara o silicón. Utiliza las siguientes imágenes para guiarte:



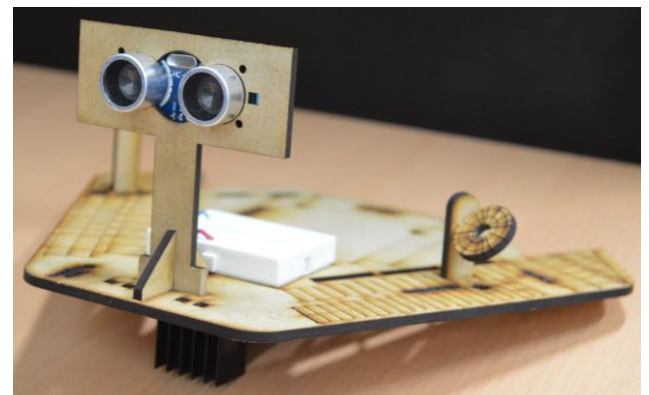
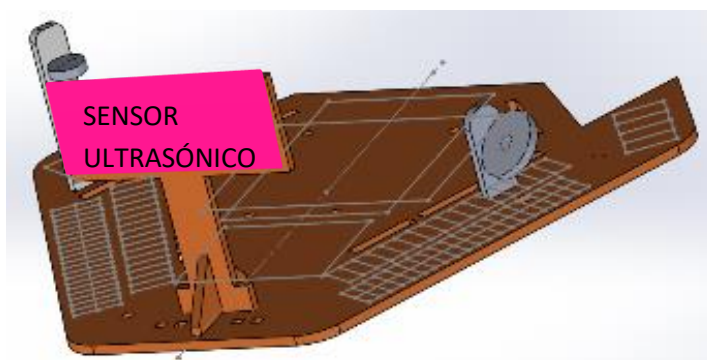
Vista inferior de la parte superior del Rover

Sensor Ultrasónico

Ubica el soporte del sensor Ultrasónico, ármalo con las pizas indicadas y colócalo en la parte frontal de la parte superior del Rover. Inserta el sensor ultrasónico con las terminales hacia abajo. Observa las siguientes imágenes:



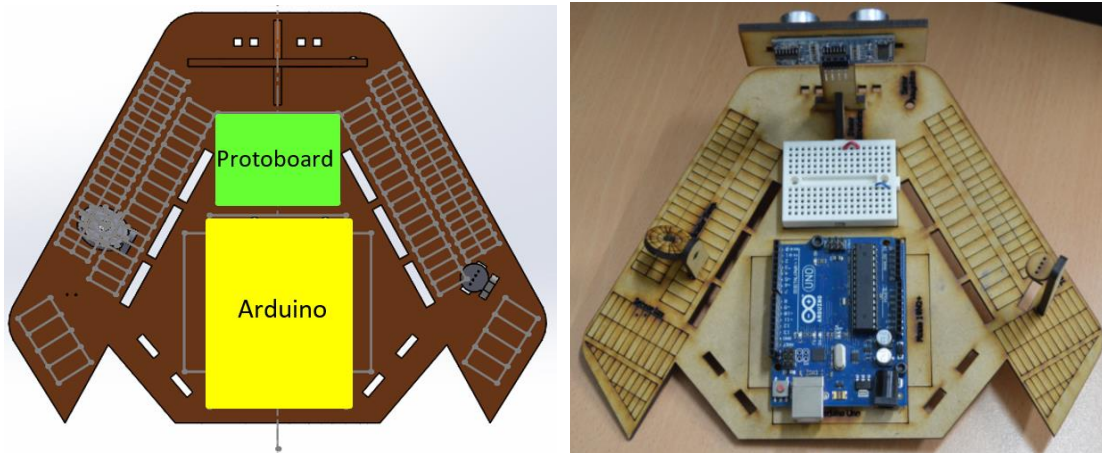
Piezas del soporte del sensor ultrasónico.



Ubicación del sensor ultrasónico.

Arduino y protoboard

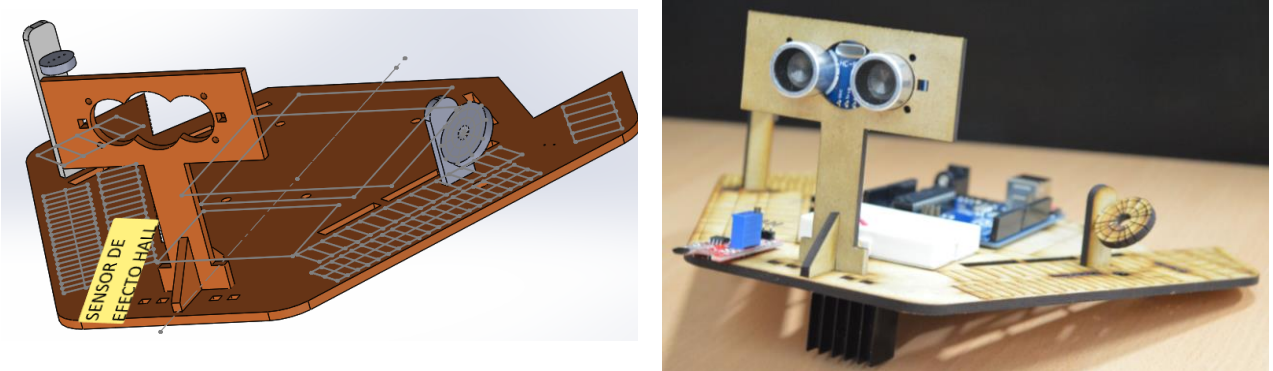
Coloca la placa Arduino y el protoboard en la parte superior del Rover. Fija la placa Arduino con ayuda de tornillos o cinta doble cara. El protoboard usualmente tiene en la parte de abajo cinta autoadherible, si no, agrega cinta doble cara



Vista superior de la parte superior del Rover.

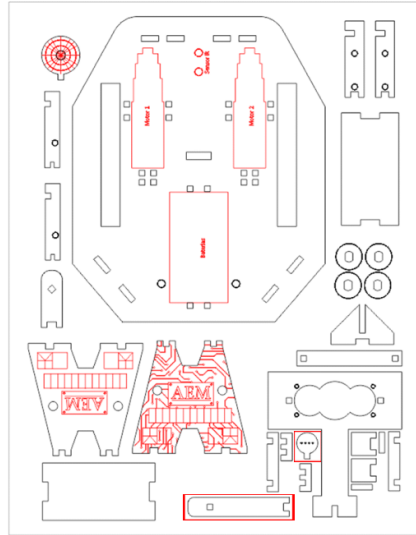
Sensor de Efecto Hall

Coloca el sensor de Efecto Hall al costado derecho del sensor ultrasónico. Puedes fijarlo con silicón o cinta doble cara.

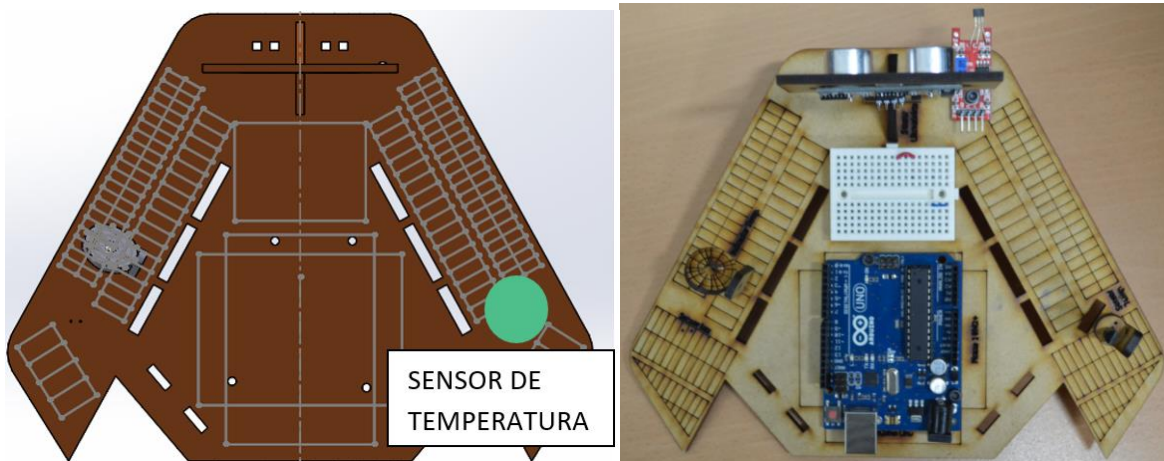


Sensor de temperatura LM35

Arma el soporte para el sensor de temperatura LM35. Ubica el sensor en la parte posterior de la parte superior del Rover. Emplea pegamento o silicón para fijar el poste y soporte. Utiliza las siguientes imágenes para guiarte:



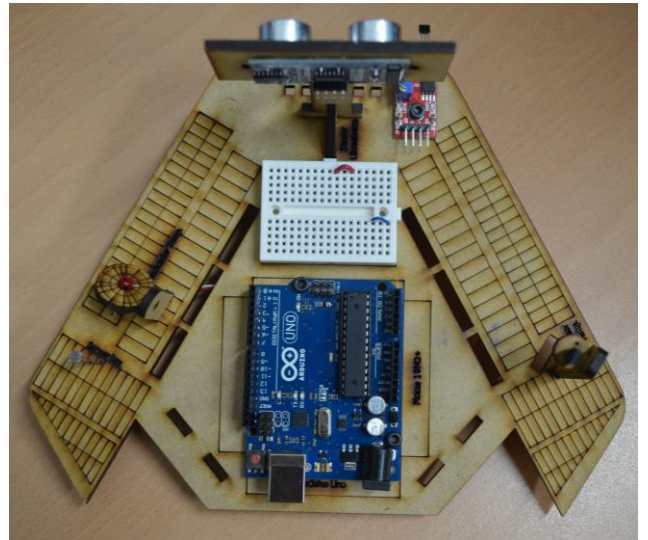
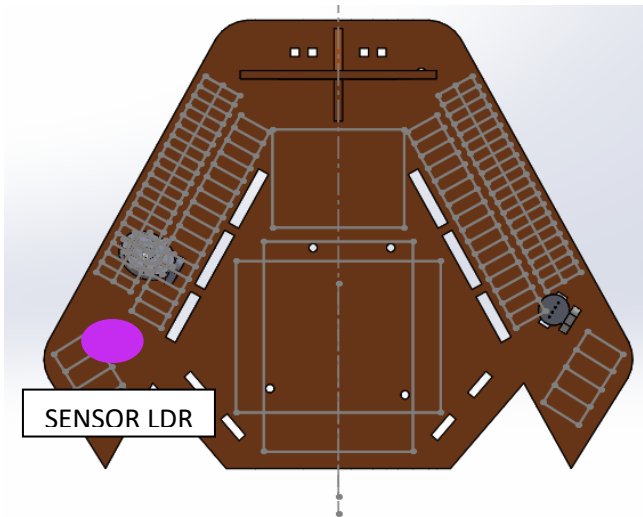
Ubicación de las piezas del soporte del sensor LM35



Vista superior de la parte superior del Rover

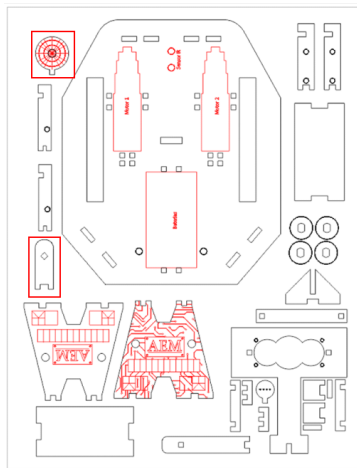
Sensor LDR

Coloca el sensor LDR en el extremo opuesto al sensor de temperatura. Este sensor no requiere ningún soporte en especial, así que solo inserta las terminales en los orificios. Emplea las siguientes imágenes para guiarte:

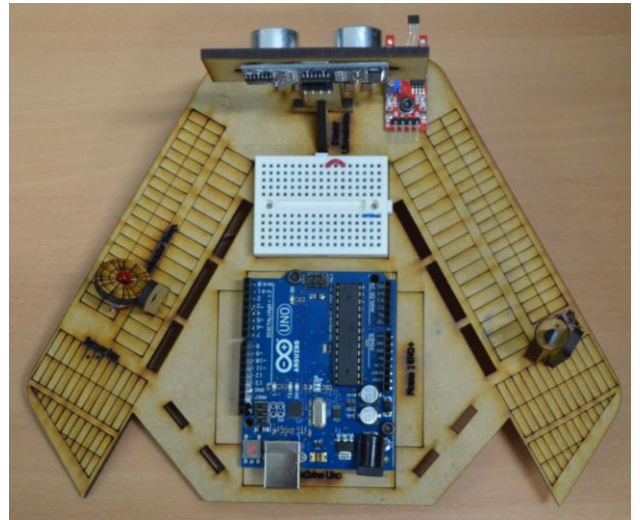
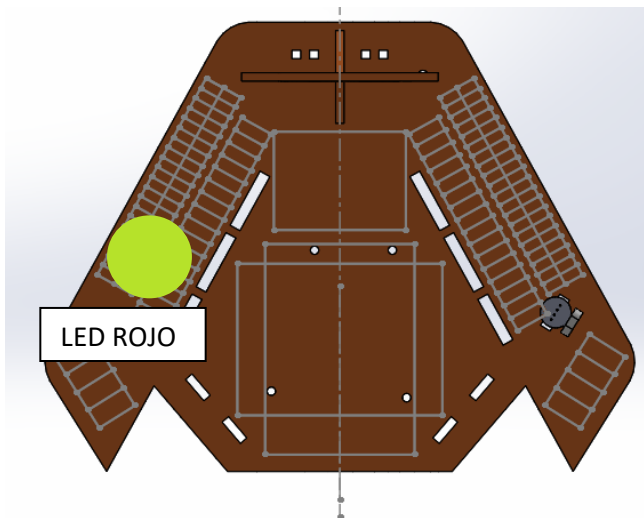


Led Rojo (temperatura)

Finalmente, Arma el soporte para el LED rojo, que indicará el aumento de temperatura. Ubica el soporte al frente del sensor LDR. Utiliza las siguientes imágenes para guiarte:



Ubicación de las piezas del soporte para el LED rojo.



Vista superior de la parte superior del Rover.



Vista frontal de la parte superior del Rover.

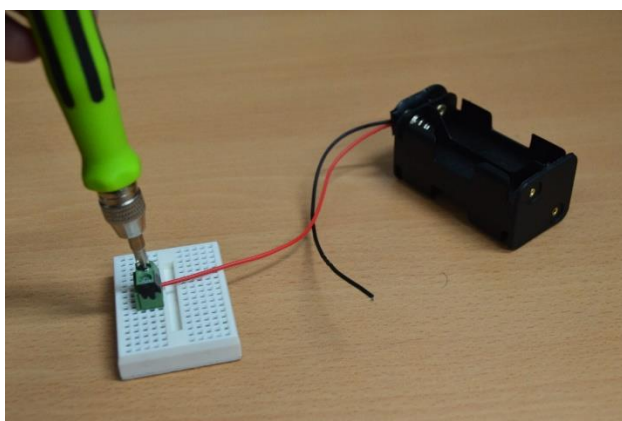
Fuente de Energía del Rover

Los exploradores espaciales llevan consigo dos fuentes de energía, una batería y un panel solar que recarga dicha batería. En este manual solo se empleará como fuente de energía 4 pilas AA o una batería de respaldo para celular.

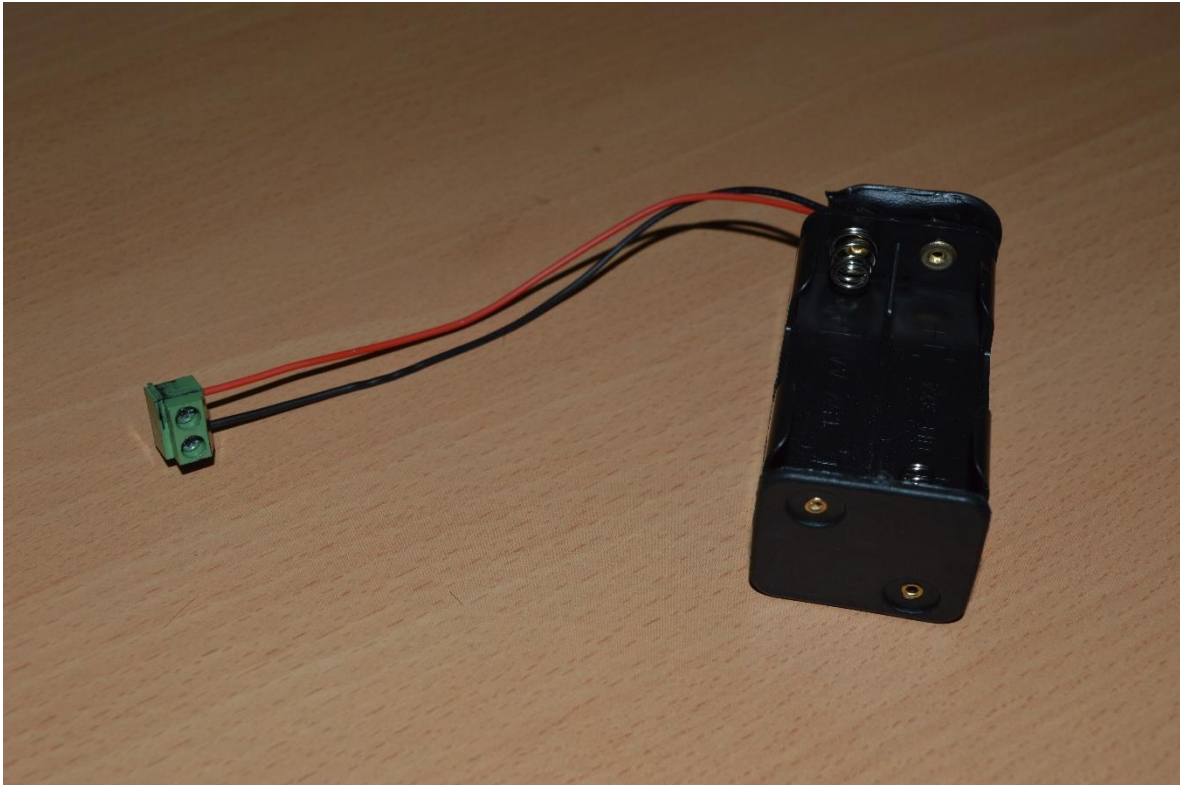
a) Utilizando 4 pilas AA y un porta pilas



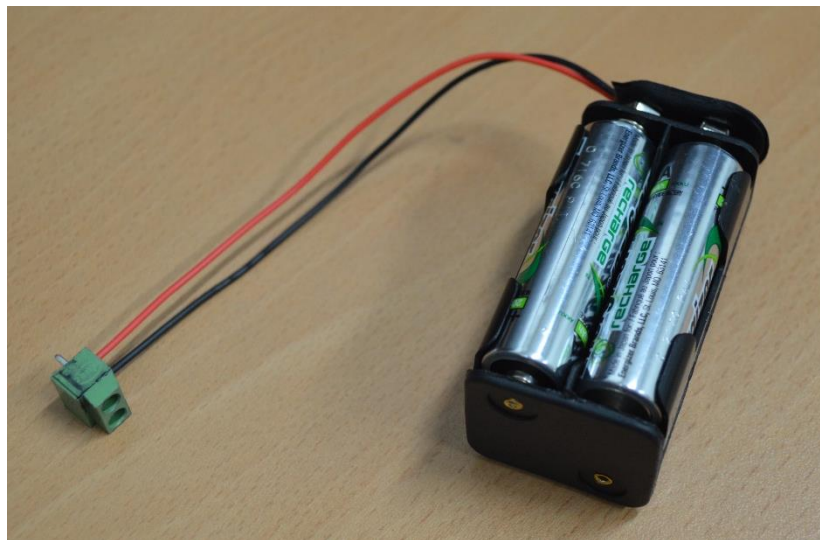
Conecta las terminales del porta pilas en una bornera de dos terminales, Fija bien los cables del porta pilas ajustando los tornillos de la bornera:



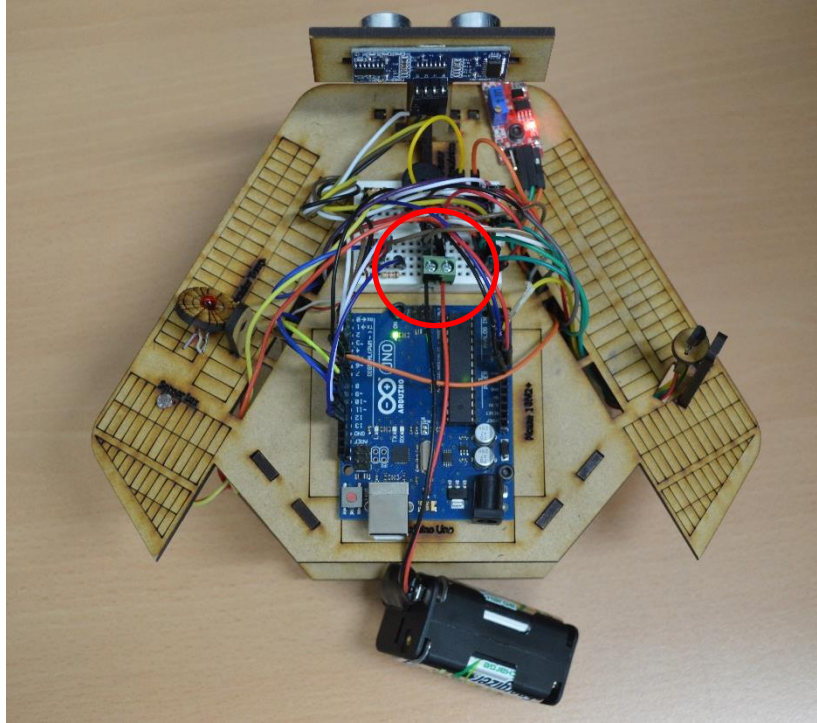
Finalmente nos quedará el porta pilas de siguiente manera:



Como último paso, coloca las pilas en el porta pilas



Conecta las terminales al protoboard



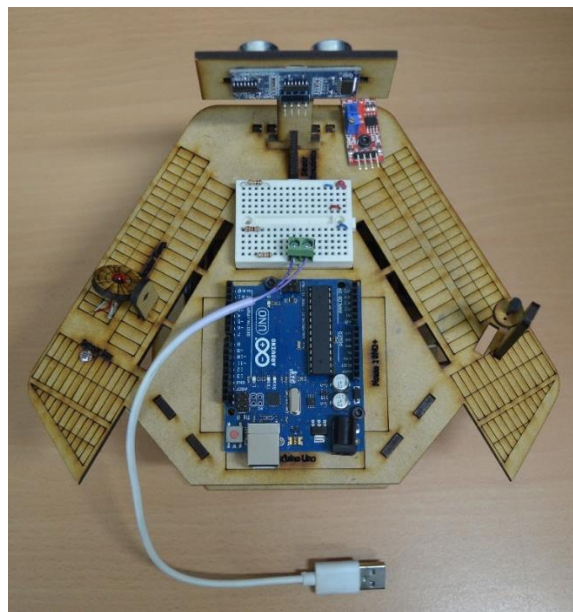
b) Utilizando un cable con puerto USB y una batería recargable externa

Para esta opción necesitarás un cable que tenga una terminal USB Macho tipo A (conexión “universal” para computadoras), corta el extremo contrario a la terminal USB macho.



Dependiendo del fabricante, el cable puede tener 2 a 4 cables. Solo ocuparás los cables marcados con color rojo y negro, ignora los demás cables, intenta cortarlo a ras. Por convención la terminal positiva es marcada de color rojo y la terminal negativa es marcada de color negro, en caso de que no estén marcados puedes utilizar un multímetro para identificar las terminales y marcarlas para identificarlas.

Una vez que tenemos cortado el cable e identificados los cables del extremo los conectaremos a una bornera de dos terminales. Si el cable USB no cuenta con las terminales marcadas con color rojo y negro, cuando lo conectes a la bornera, marca la bornera para identificar cual terminal es la positiva o negativa. El extremo con la terminal USB se conectará a la batería de respaldo.



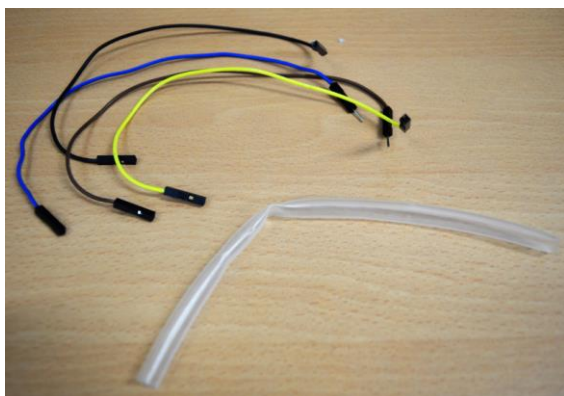
Conexiones

Este paso es muy importante. Observa cuidadosamente las imágenes y diagramas para que las conexiones sean correctas. También puedes guiarte de las prácticas anteriores donde se señalan las conexiones individuales por cada sensor, actuador e indicador.

Con el fin de tener una mejor organización en los cables, te recomendamos usar Thermofit transparente, este es un recubrimiento de plástico que se contrae con el calor. En este ejemplo usaremos un encendedor, pero también puedes emplear una pistola de calor para el ajuste.

Dependiendo del sensor, toma los cables necesarios y pásalos por dentro del thermofit.

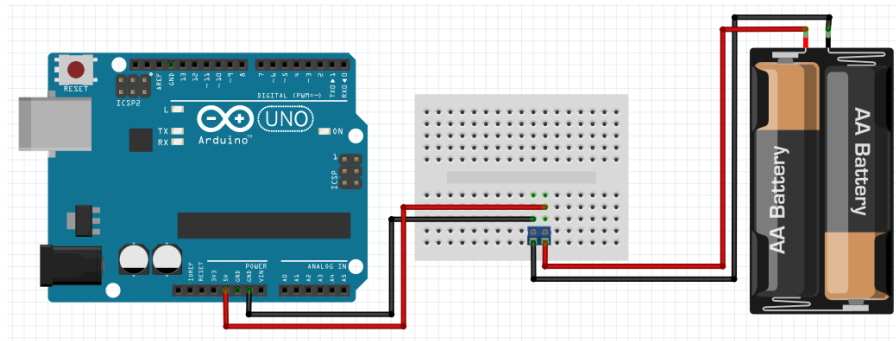
Ocupa un encendedor y acércalo al plástico. Comenzarás a ver como el plástico se encoje, déjalo solo unos segundos, de lo contrario derretirás todos los recubrimientos y dañarás los cables dupont.



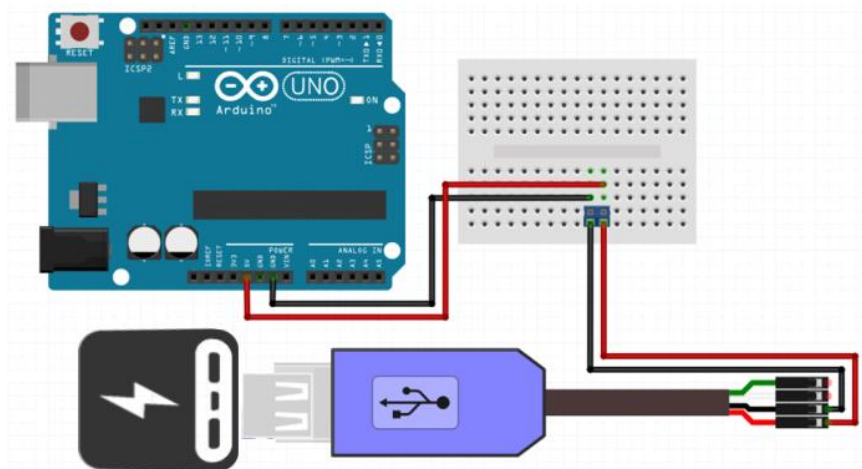
Si no cuentas con Thermofit, puedes usar cinchos para agrupar los cables.

Emplea un protoboard para conectar GND y Vcc de cada sensor ya que la placa Arduino ofrece pocas salidas. Dependiendo del tipo de alimentación que desees utilizar será la conexión:

a) Pilas AA

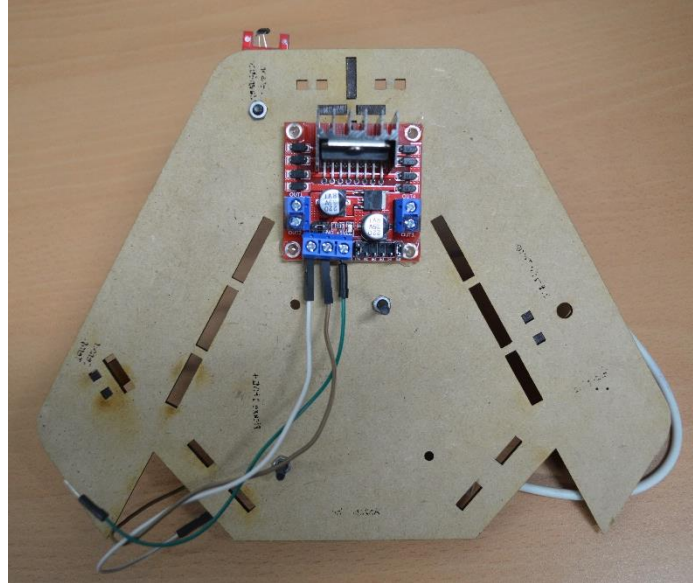


b) Batería Externa recargable

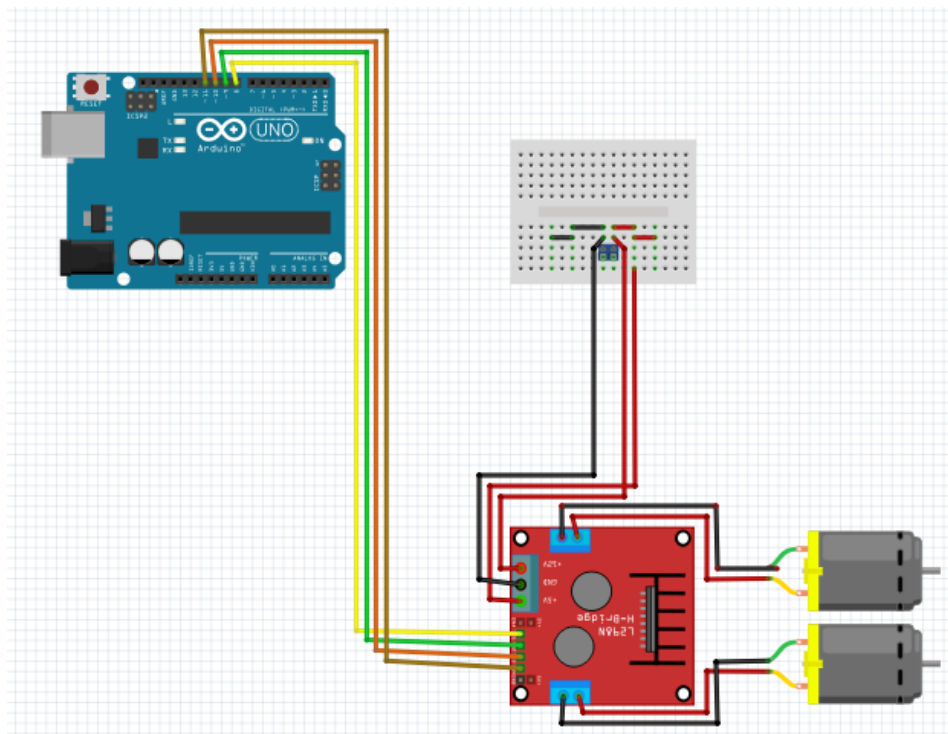


Puente H

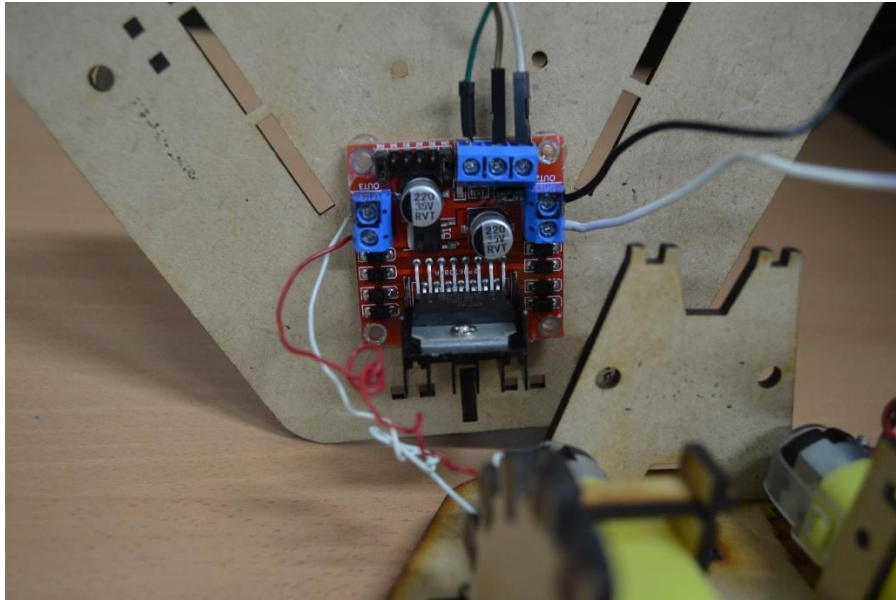
Coloca tres cables dupont en las terminales del puente H como se muestra a continuación:



Conecta los motores al puente H tomando en cuenta la siguiente información:

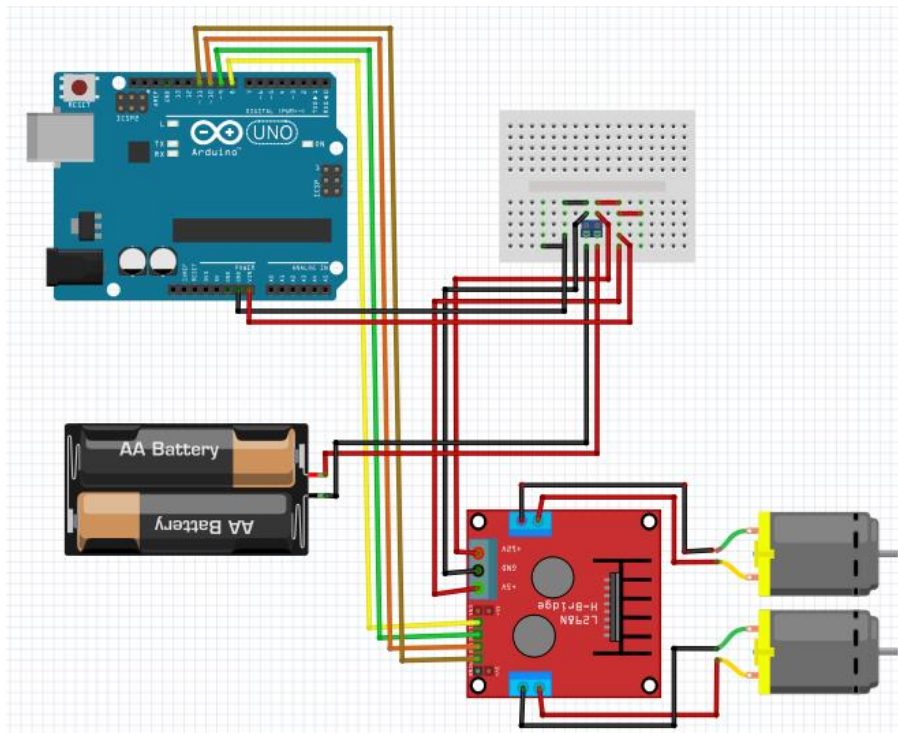


El circuito quedará de la siguiente manera:

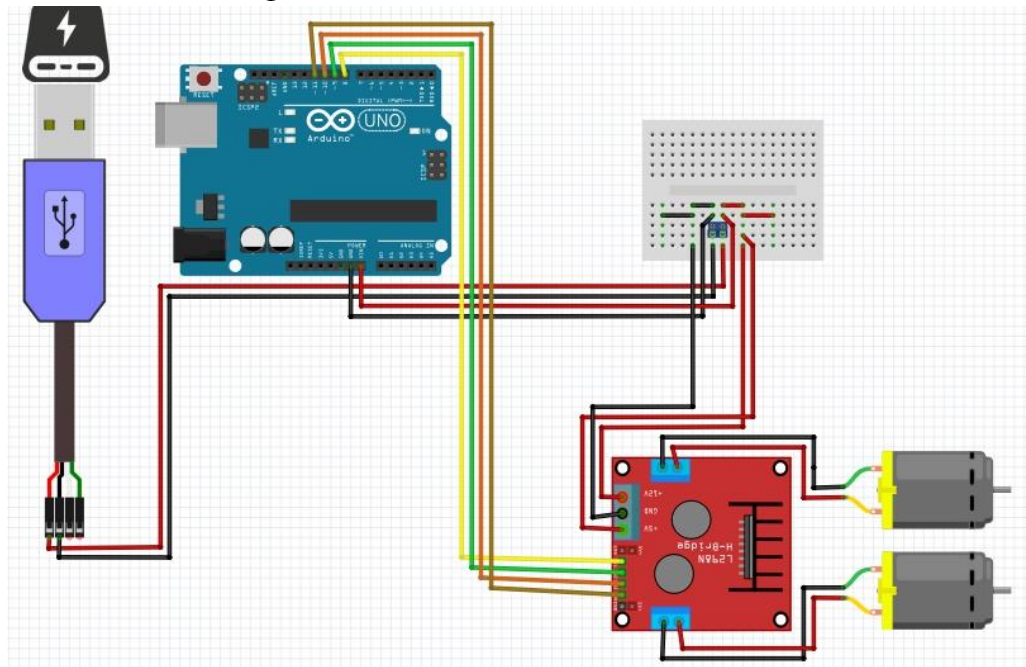


Conecta las cuatro terminales de control del puente H a la placa Arduino, emplea 4 cables Dupont macho-hembra uno para cada terminal. Conecta la terminal con etiqueta “IN1” al pin digital 8, la terminal con etiqueta “IN2” al pin digital 9, la terminal con etiqueta “IN3” al pin digital 10, la terminal con etiqueta “IN4” al pin digital 11:

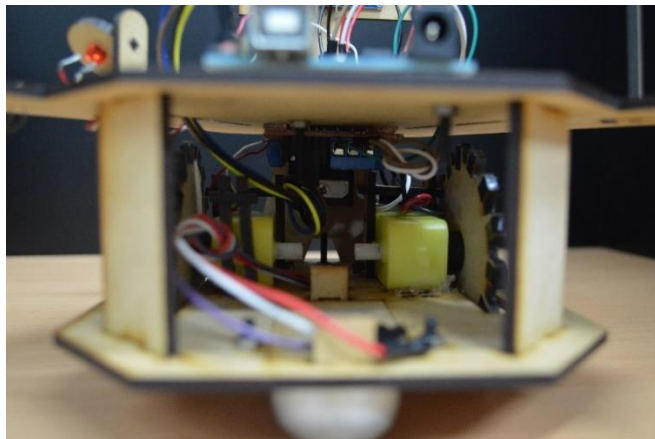
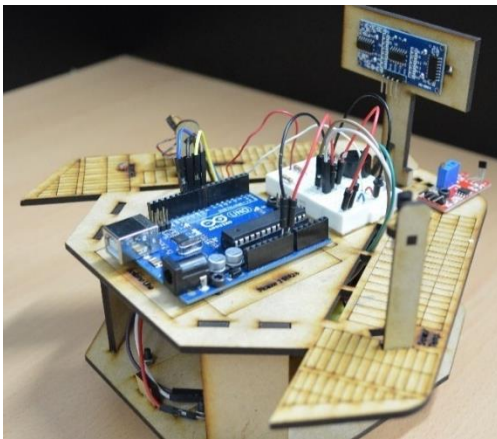
a) Pilas AA:



b) Batería Externa Recargable:

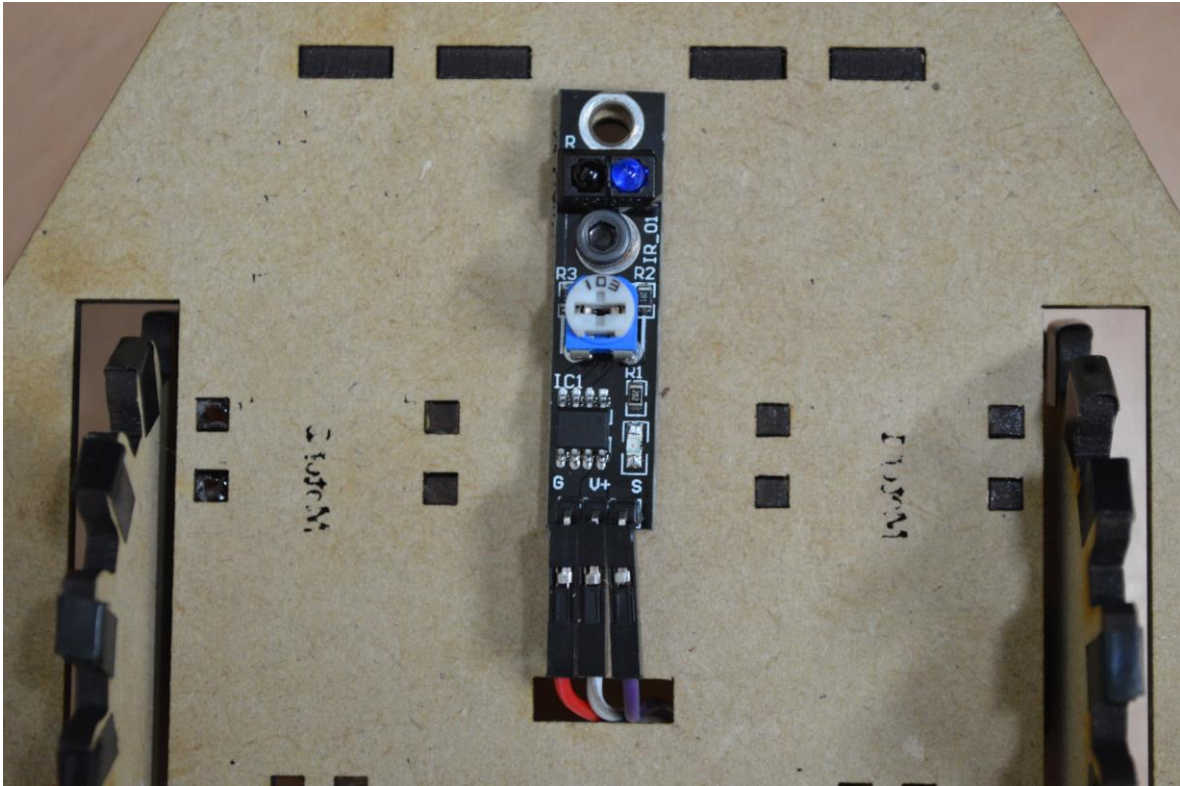


El Rover hasta el momento lucirá de la siguiente manera:



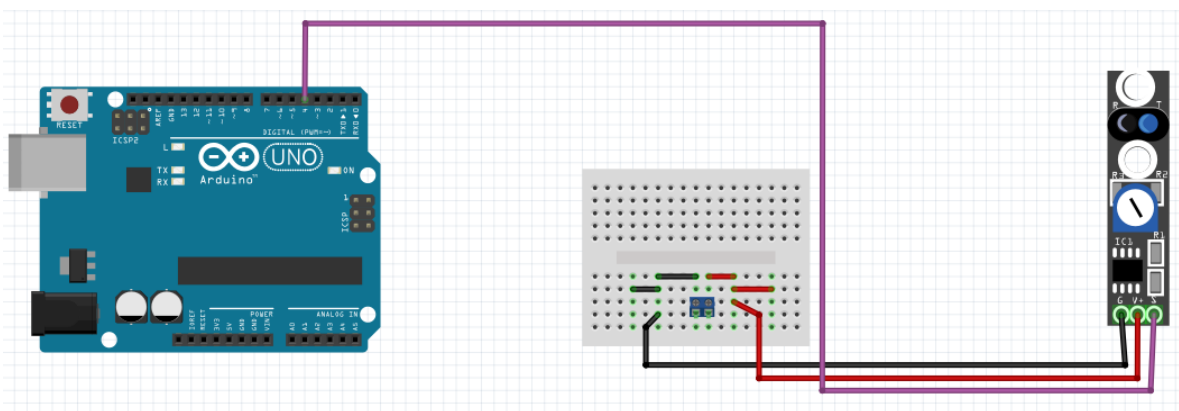
Sensor IR

Utiliza 3 cables macho-hembra, uno para cada terminal, como lo muestra la siguiente imagen:



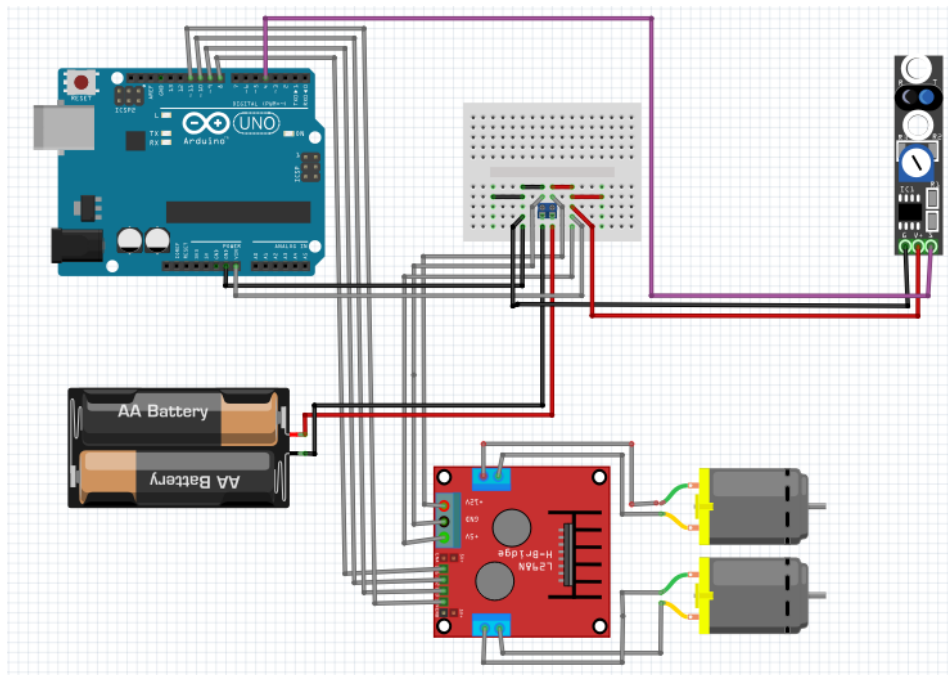
Pasa los cables por la ranura que se encuentra orientada en las terminales. Luego haz pasar los cables por las ranuras laterales de la parte superior del Rover hacia la protoboard.

Conecta el sensor IR ubicado por debajo de la parte baja del Rover, conecta la **terminal 1** identificada con la letra G, a **GND** o **tierra**; la **terminal 2** identificada con la letra y símbolo V+ a 5[V] y la **terminal 3** identificada con la letra S al **pin digital 4**..

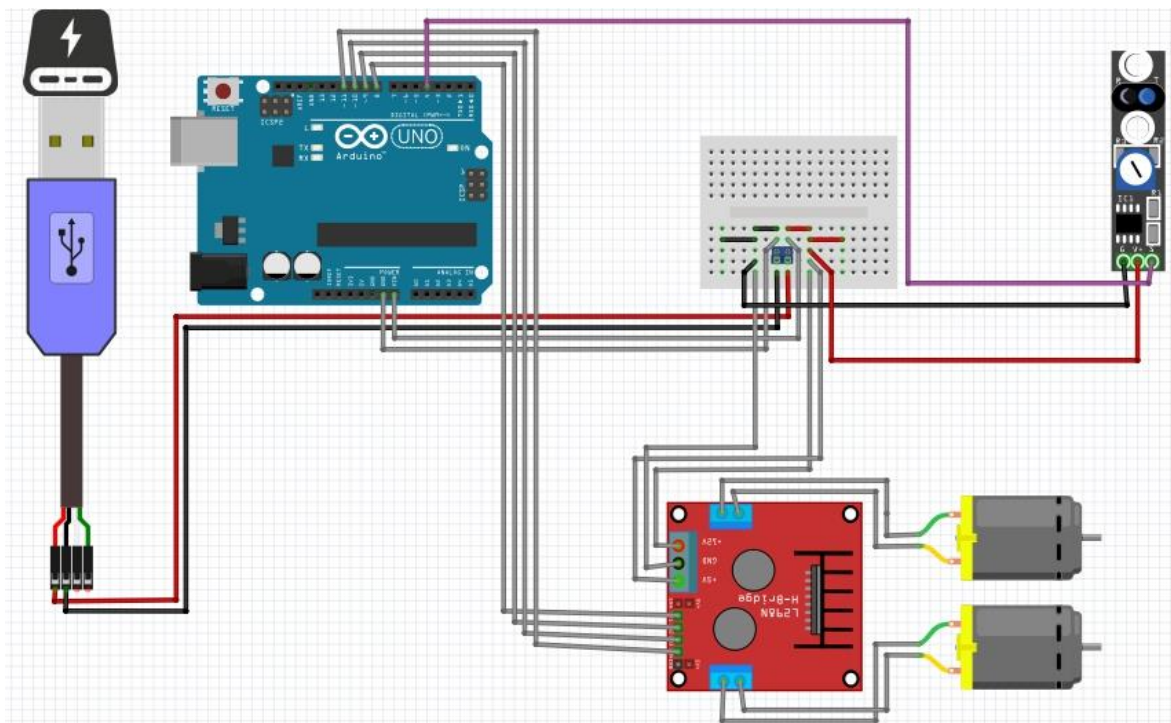


Observa las siguientes imágenes, el circuito deberá quedar de la siguiente forma:

a) Pilas AA



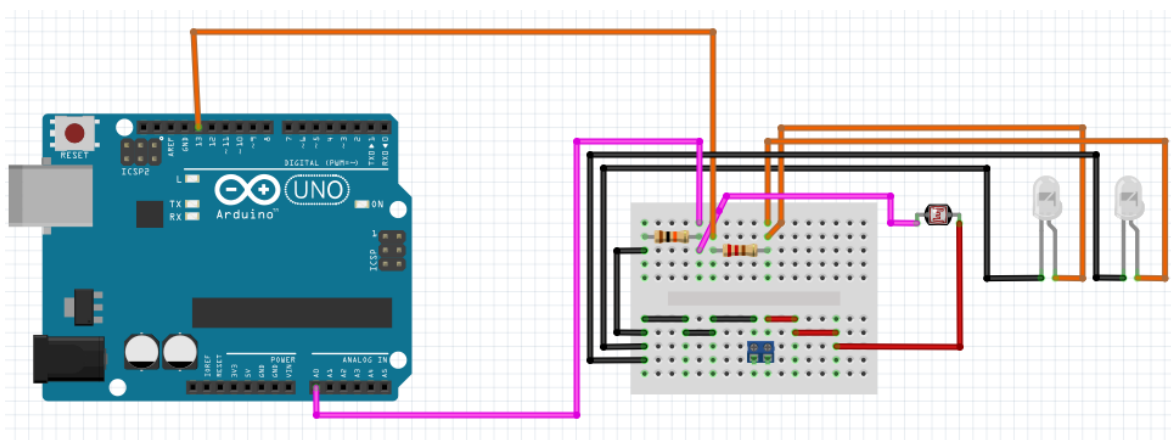
b) Batería Externa Recargable



Conexión LDR y LED ultrabrillantes

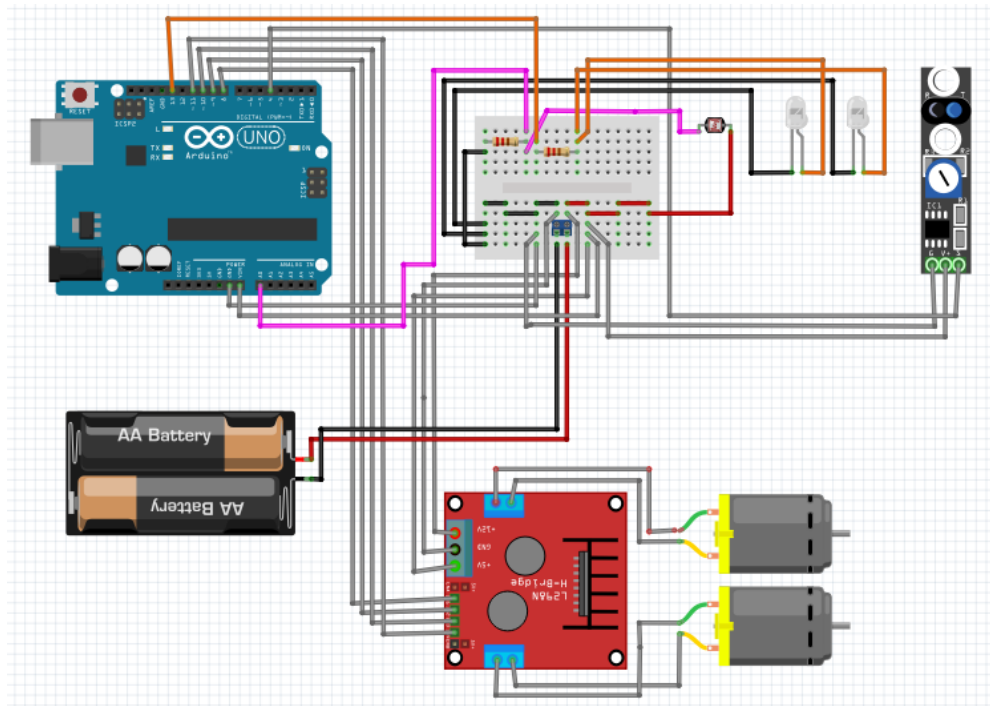
Para conectar los LED ultrabrillantes se empleará la misma salida de la placa Arduino, por lo que deberás conectar la placa Arduino y los dos Led al protoboard Utiliza 6 cables dupont macho-hembra: 2 para el sensor LDR, 2 para el LED 1 y 2 para el LED 2.

Conecta las terminales negativas de los LED ultrabrillantes a **GND** y las dos terminales positivas a una terminal de la resistencia de **220 ohm**, la otra terminal de la resistencia conéctala al **pin digital 13**. Toma un cable dupon macho-macho y extiende la conexión de GND hacia una terminal de la resistencia de **10K ohm**, la otra terminal de la resistencia conéctala al **pin analógico A0** y la misma terminal a una terminal del sensor LDR. Finalmente, la terminal restante del sensor LDR conéctala a **5 [V]**. Observa la siguiente imagen:

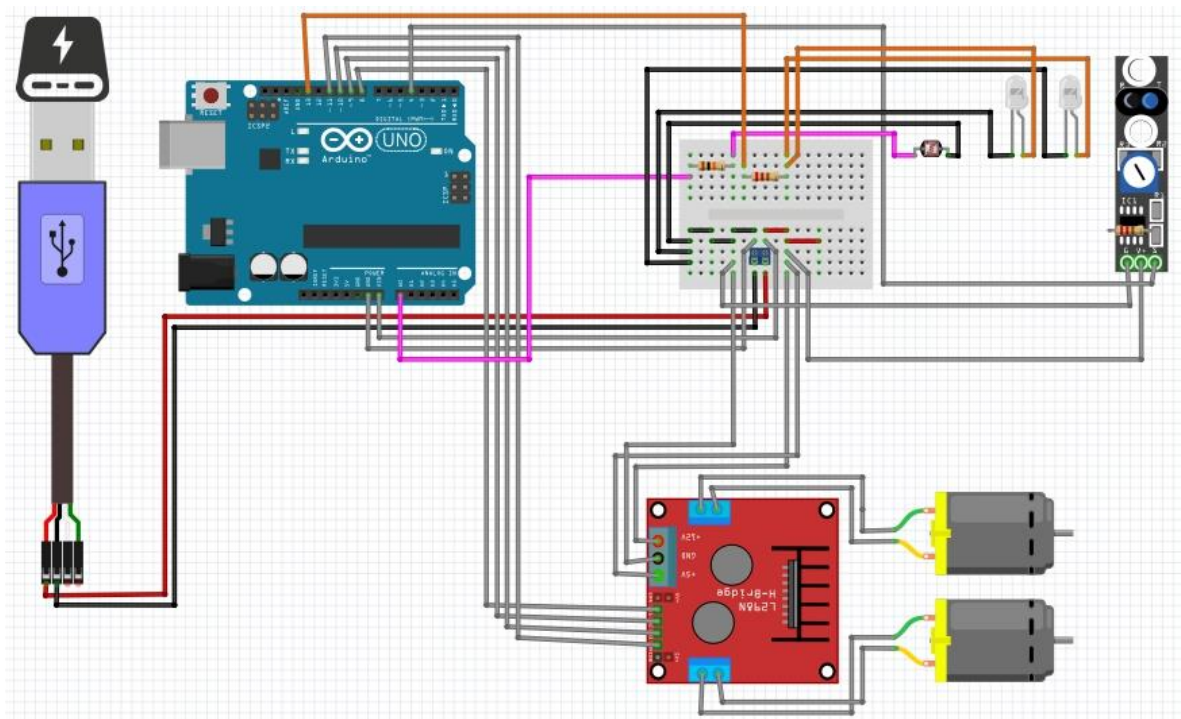


El circuito quedará de la siguiente forma:

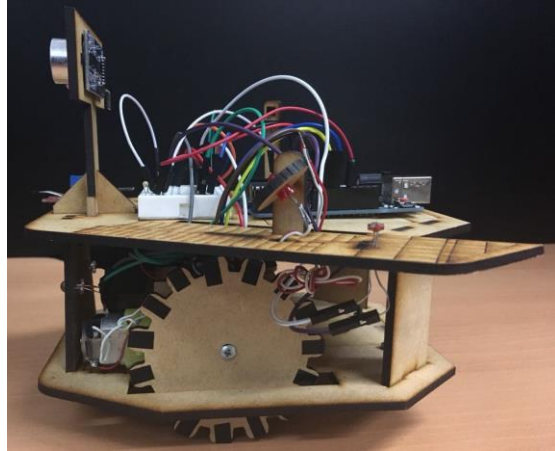
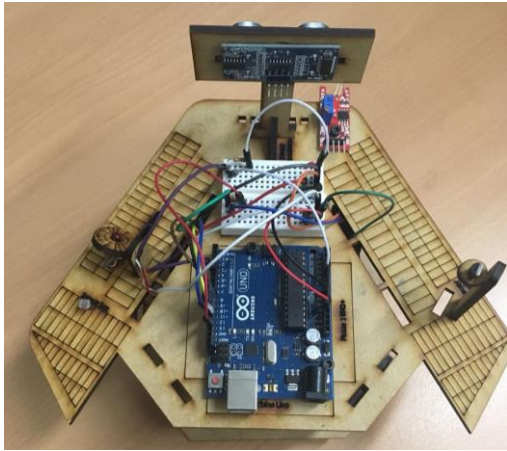
a) Pilas AA



b) Batería Externa Recargable

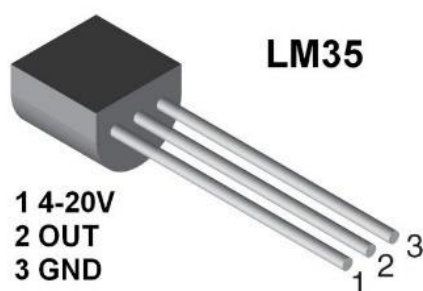


Las conexiones hasta este punto lucirán de la siguiente manera:

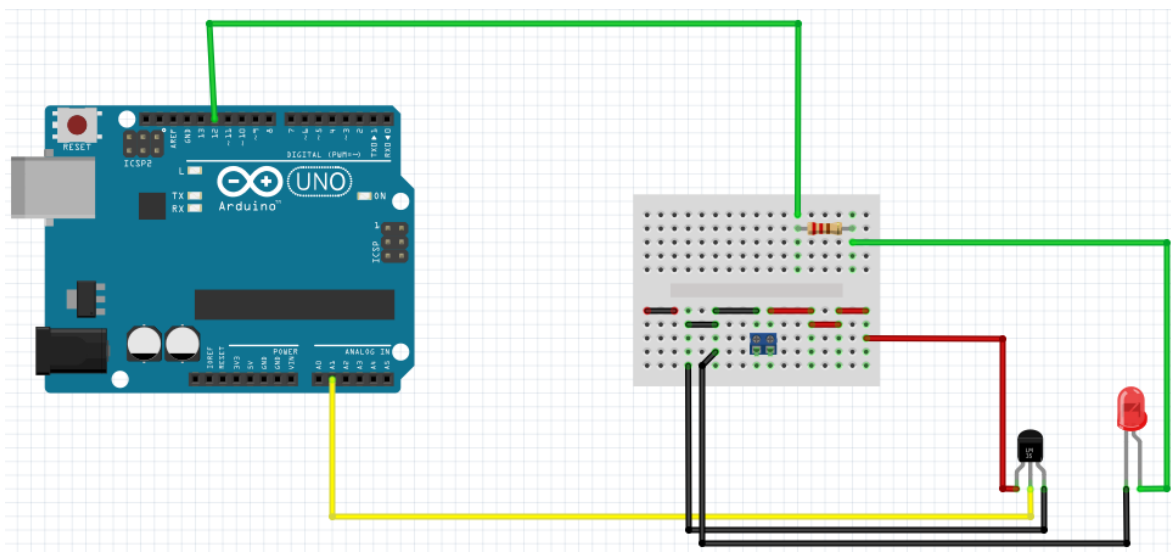


Conexión Sensor de temperatura LM35

En esta conexión conectarás el sensor de temperatura LM35 y el LED Rojo que previamente colocaste en la estructura del Rover.

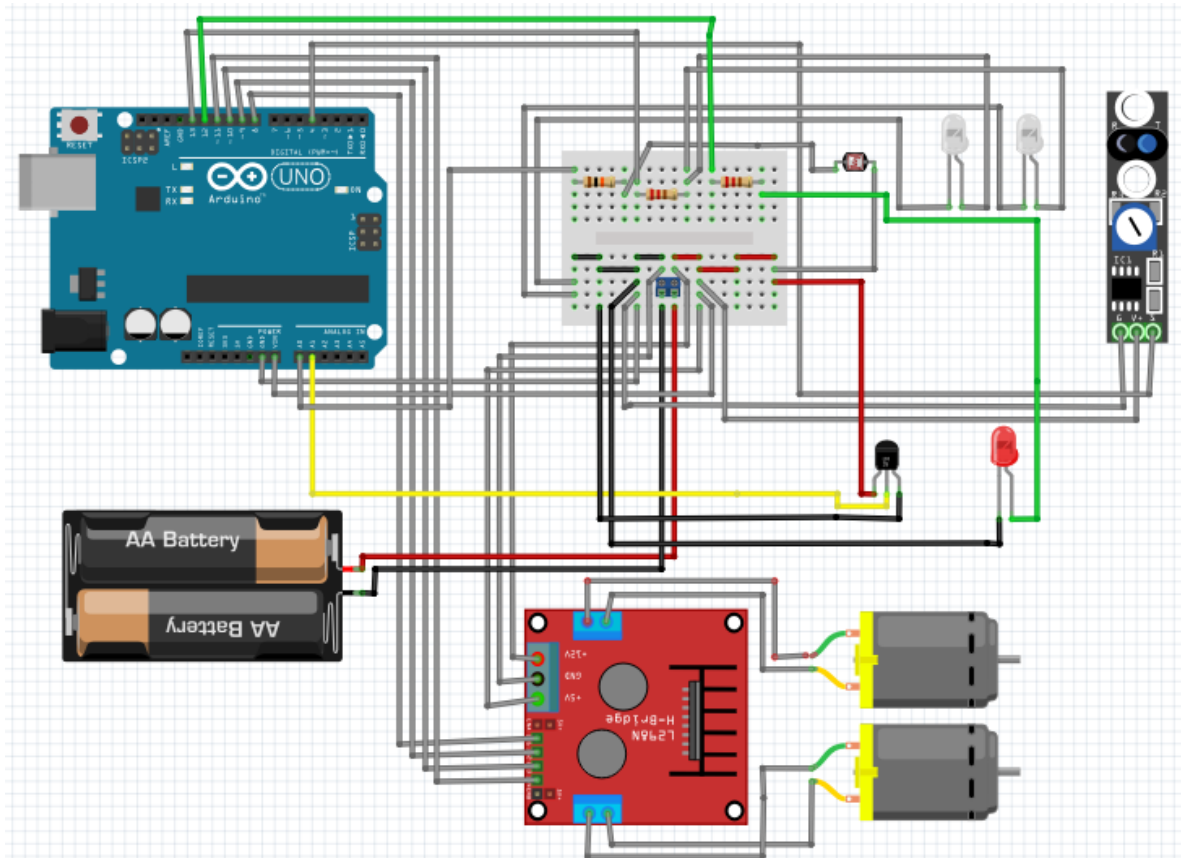


La *terminal 3* deberá ser conectada a **GND** o “Tierra”, la *terminal 1* deberá ser conectada a **5 [V]** y la *terminal 2* al pin **A1**. Para el LED rojo, conecta la terminal negativa a **GND**, la terminal positiva a una resistencia de **220 Ohm** y la otra terminal de la resistencia al **pin digital 12**:

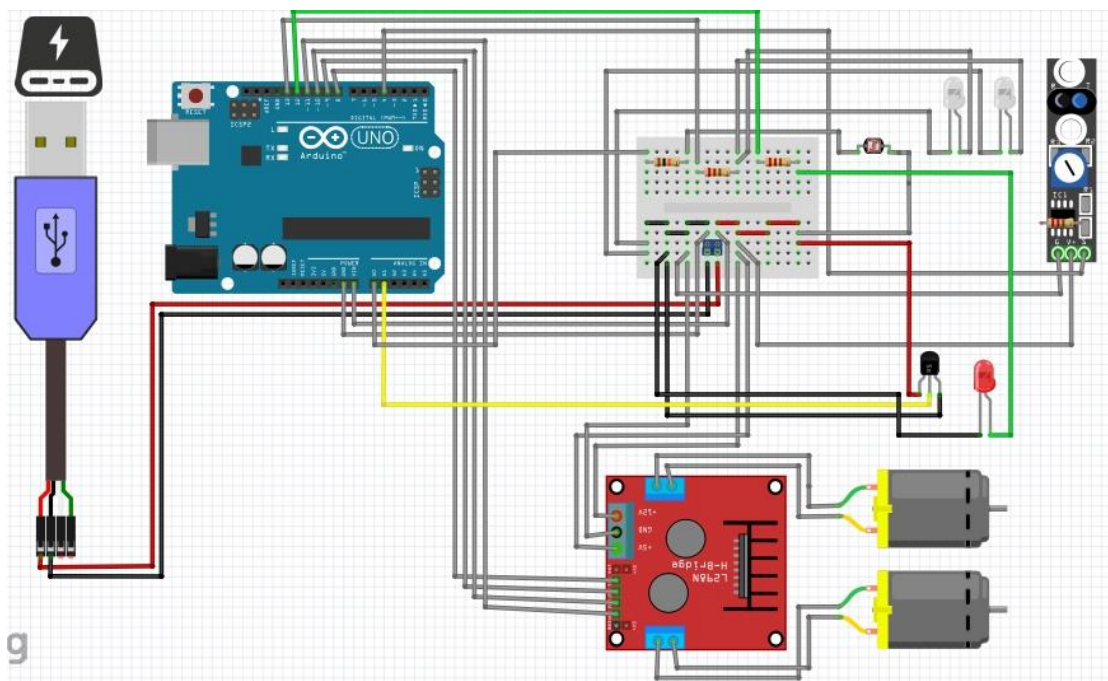


Conectándolo con los siguientes dispositivos, nos queda manera el Rover:

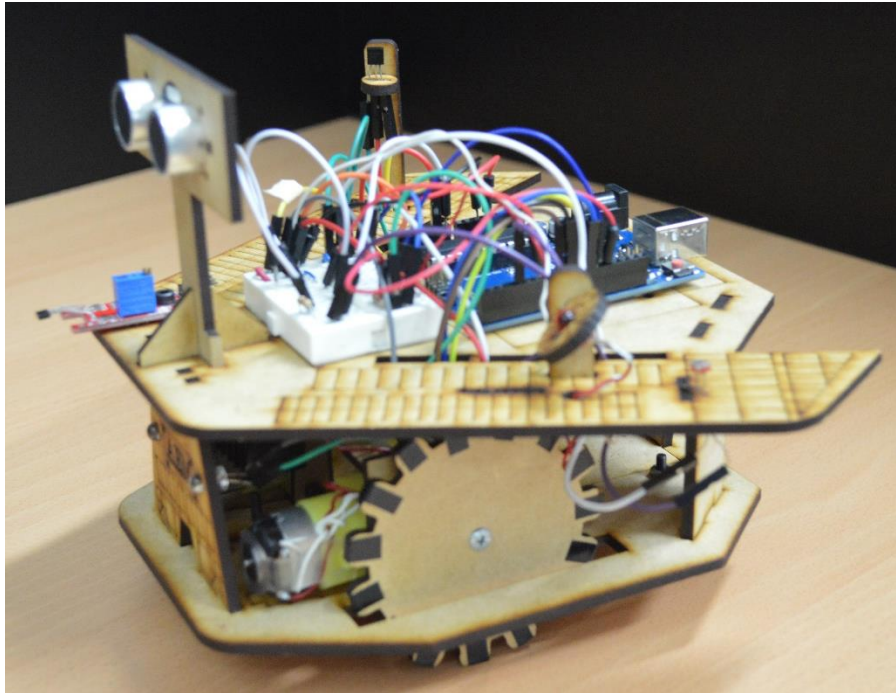
a) Pilas AA



b) Batería Externa Recargable

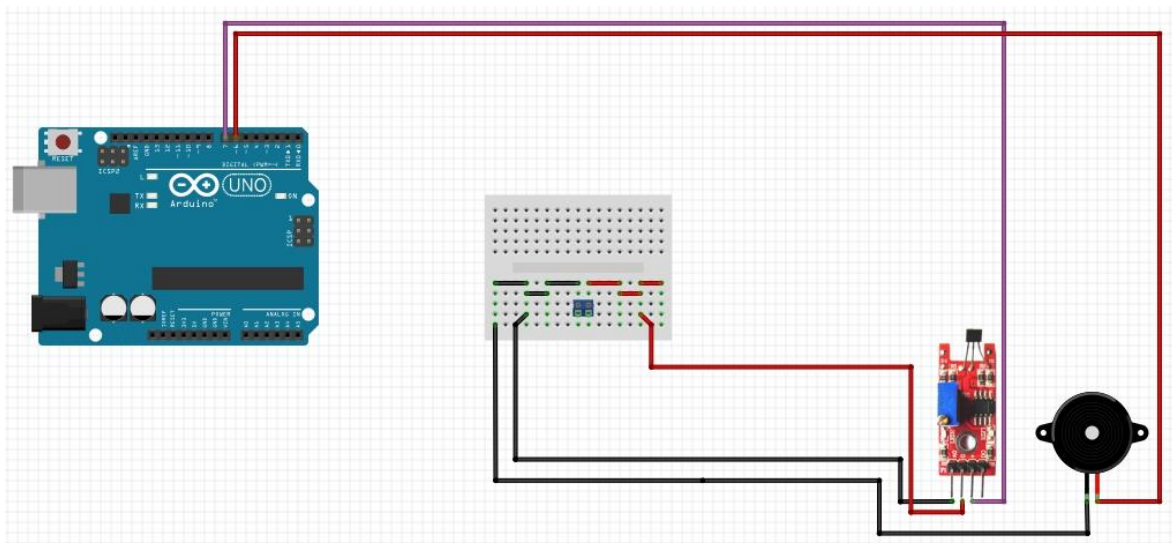


Hasta este punto, las conexiones lucirán de la siguiente forma:



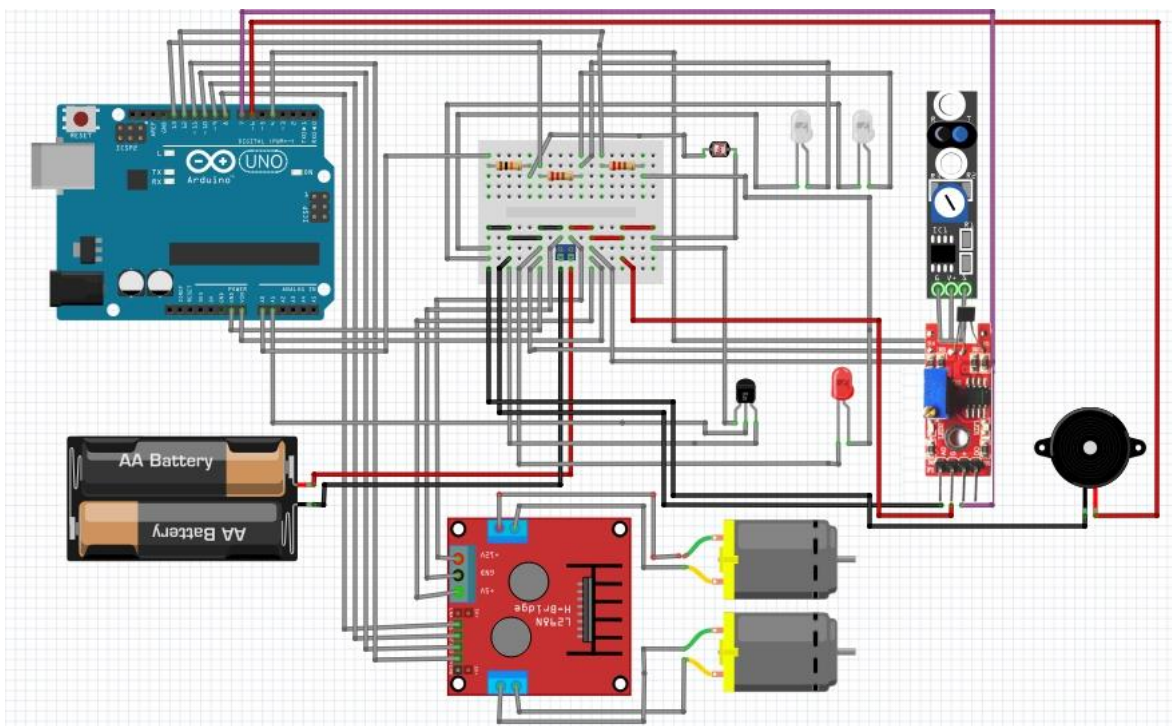
Conexión de sensor efecto Hall

El sensor mostrará su efecto por medio de un buzzer, el cual conectaremos en el protoboard y las conexiones correspondientes en el Arduino. El circuito nos quedará así:

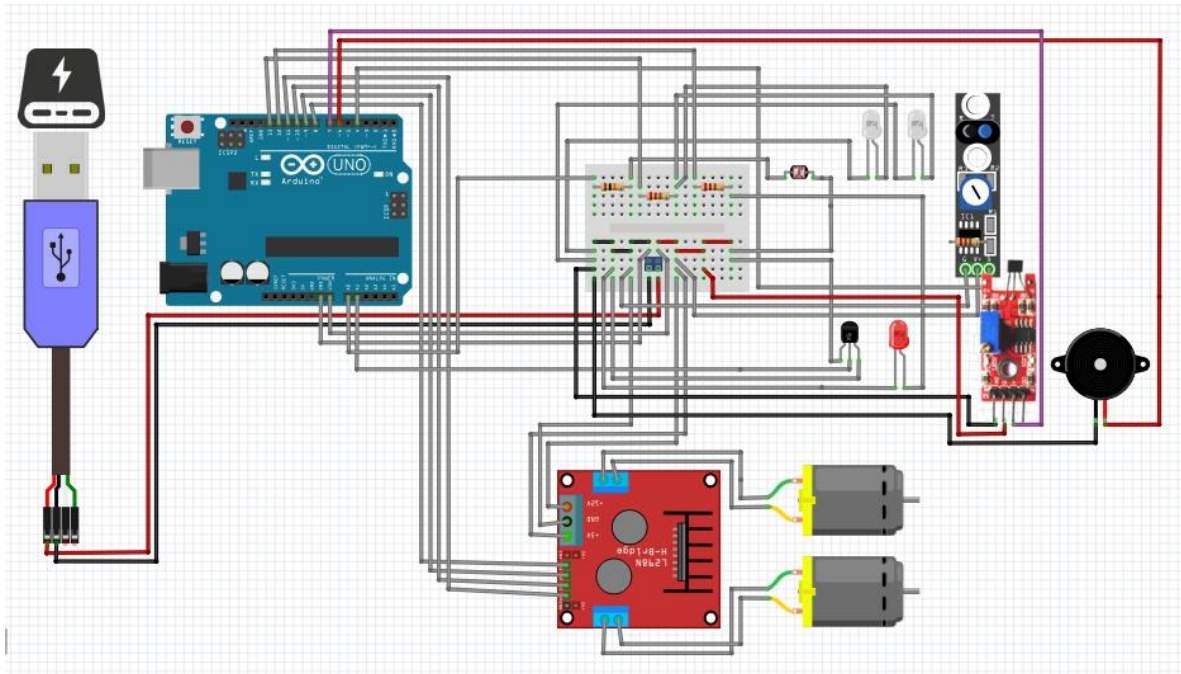


Ahora conectaremos este sensor con sus correspondientes conexiones en el Rover, y tendremos el siguiente circuito:

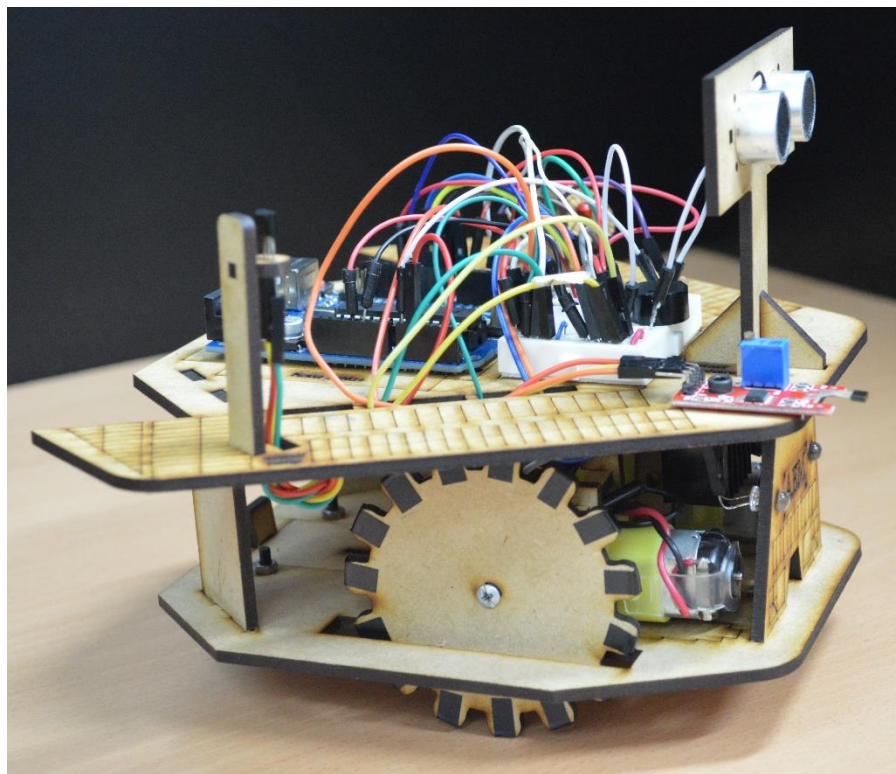
a) Pilas AA



b) Batería Externa Recargable

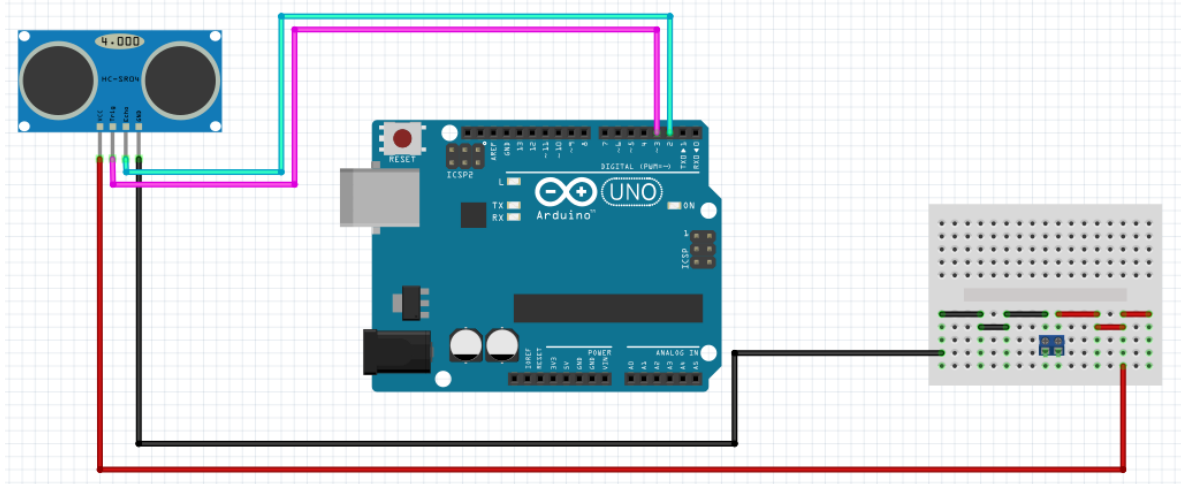


Hasta este punto, dependiendo de tu organización, las conexiones lucirán de la siguiente forma:



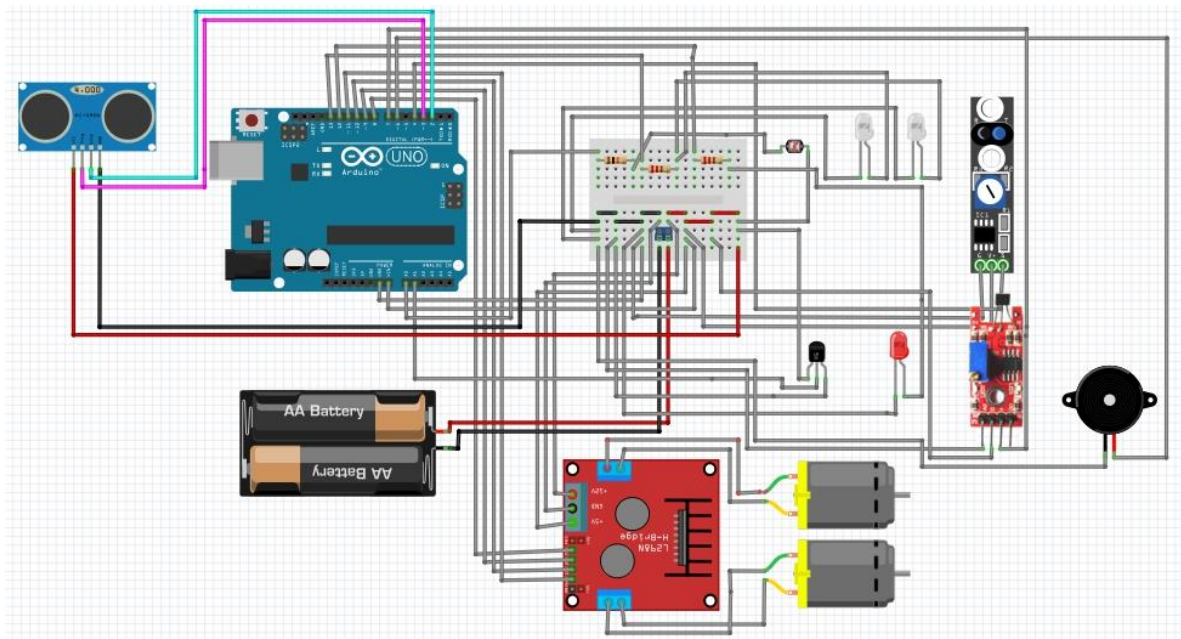
Conexión del sensor ultrasónico

Finalmente, conecta el sensor ultrasónico a la placa Arduino de la siguiente forma:

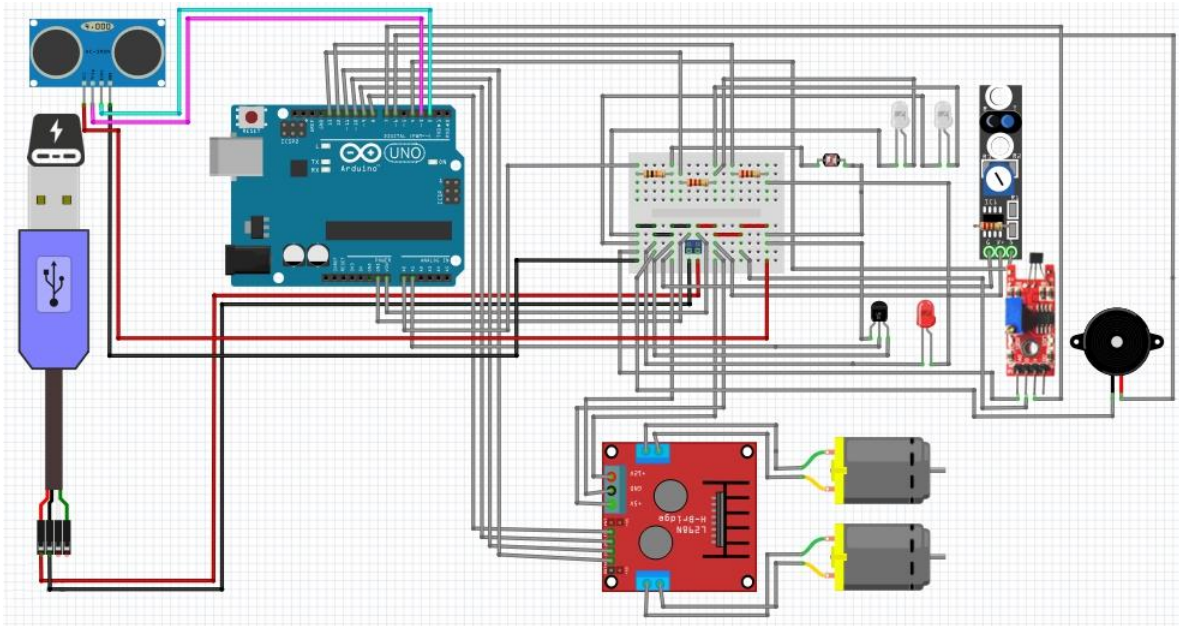


La conexión del sensor ultrasónico representa el último sensor por conectar para tener el circuito final del Rover:

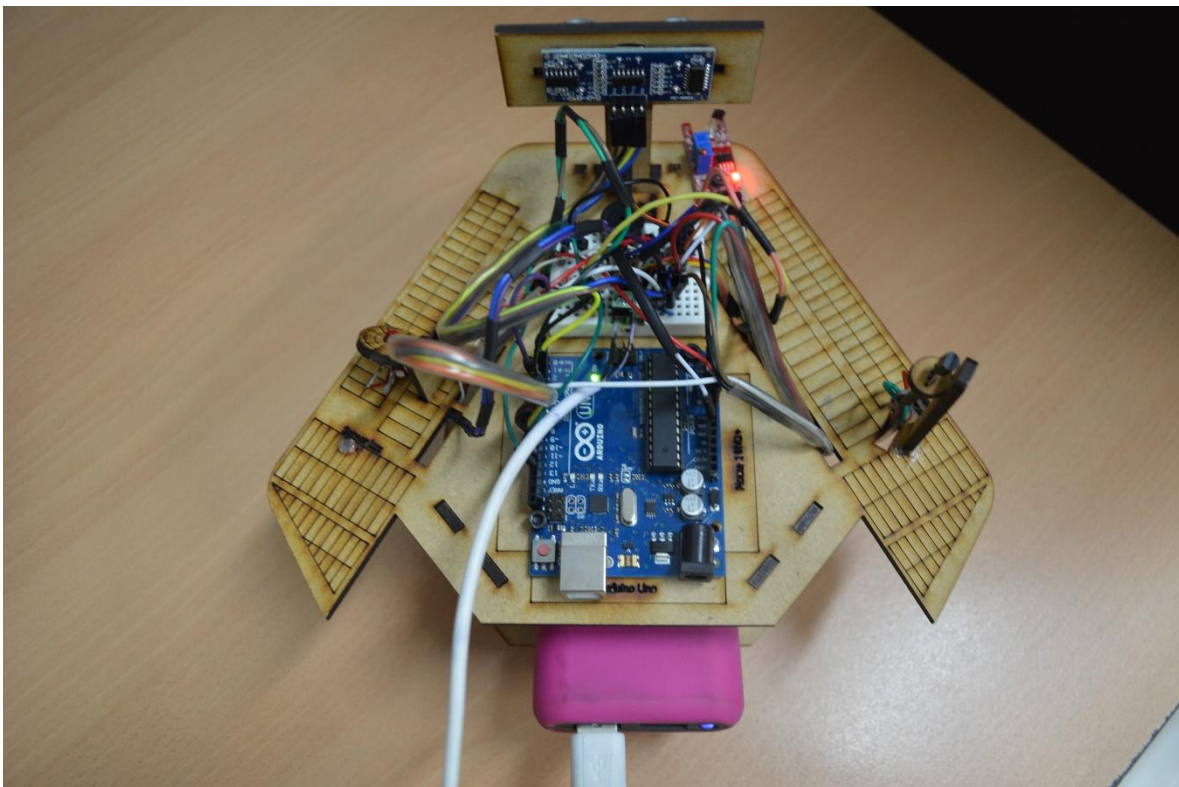
a) Pilas AA



b) Batería Externa Recargable



Hasta este punto, las conexiones lucirán de la siguiente forma:



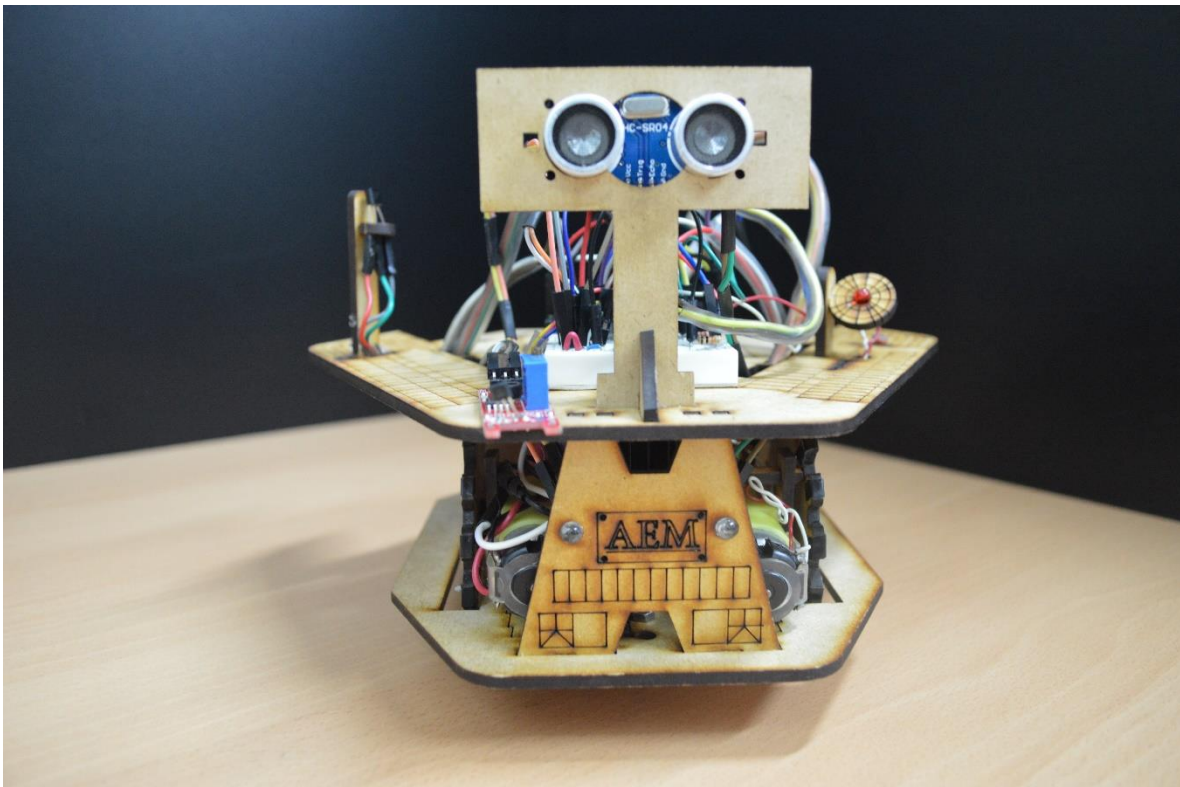
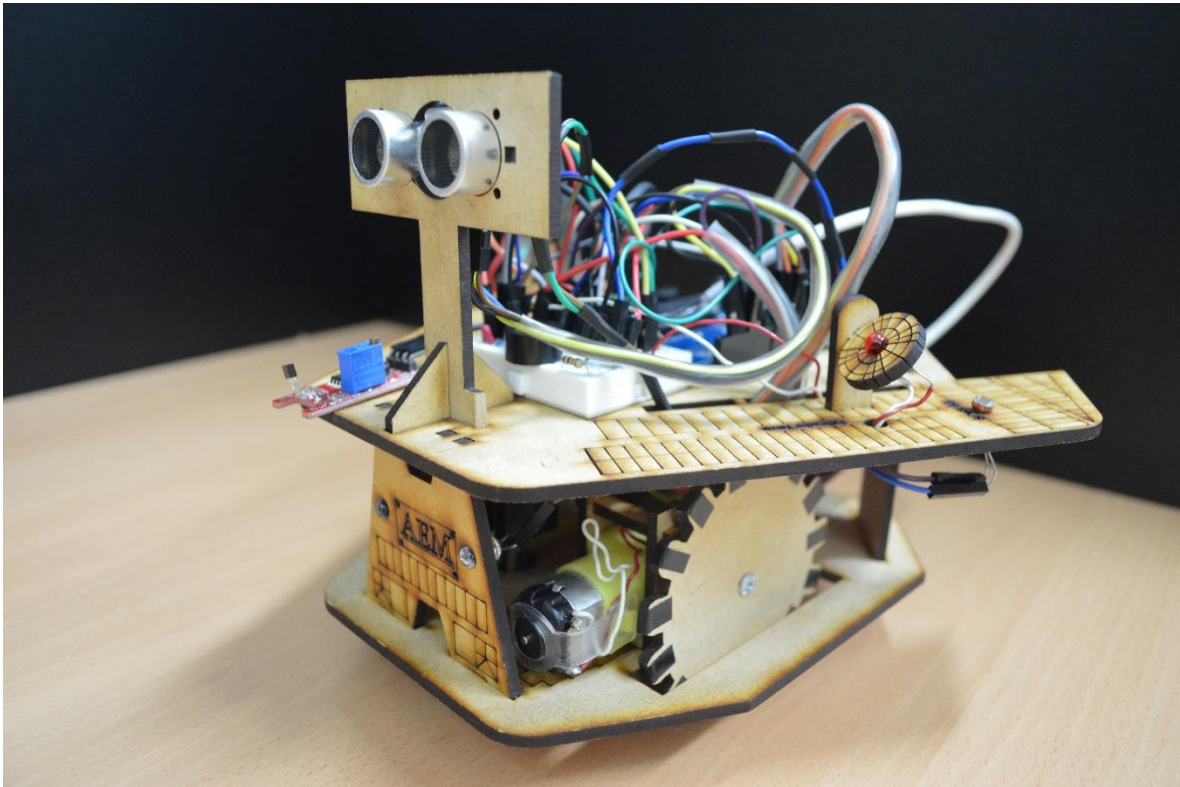
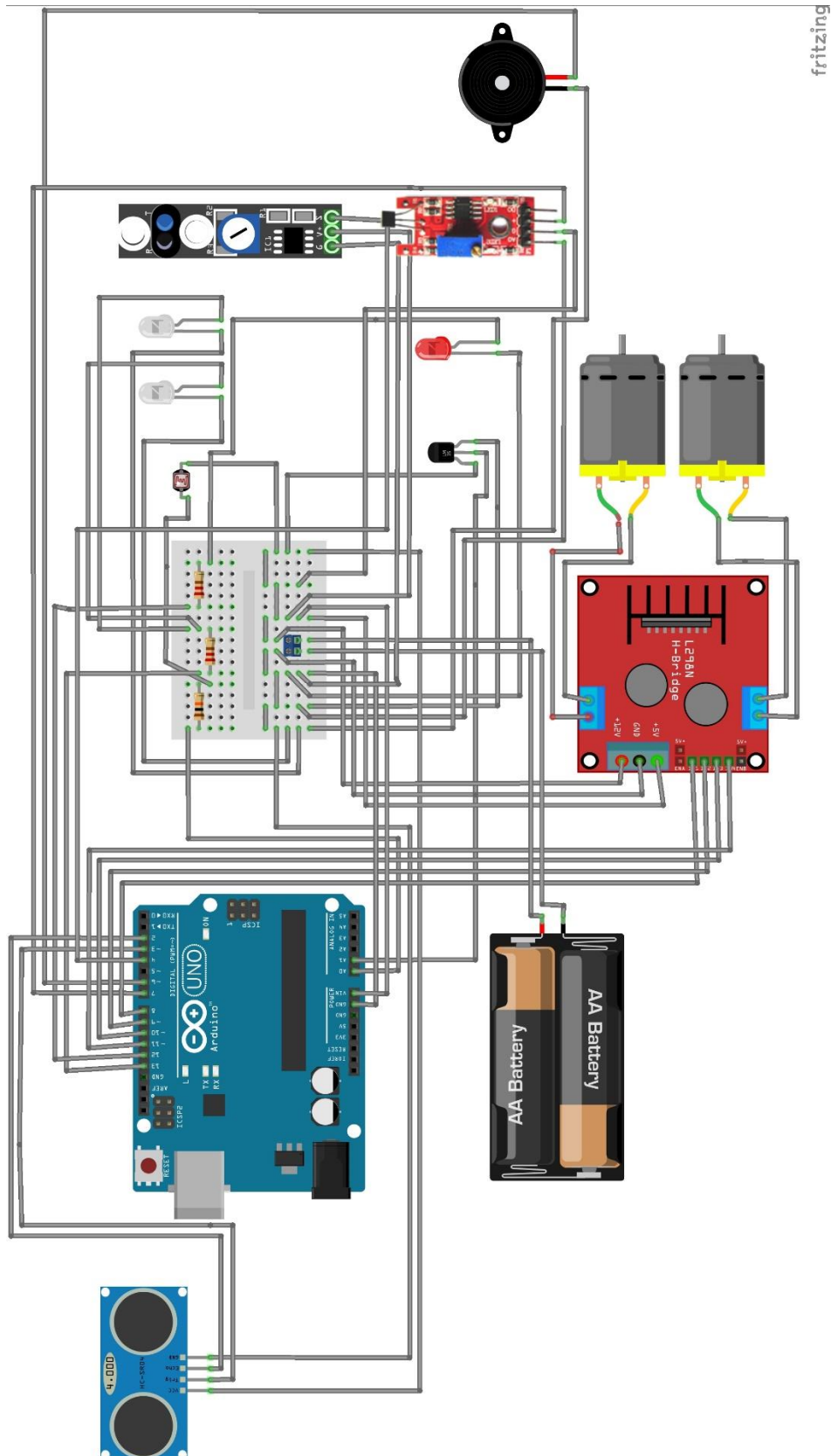
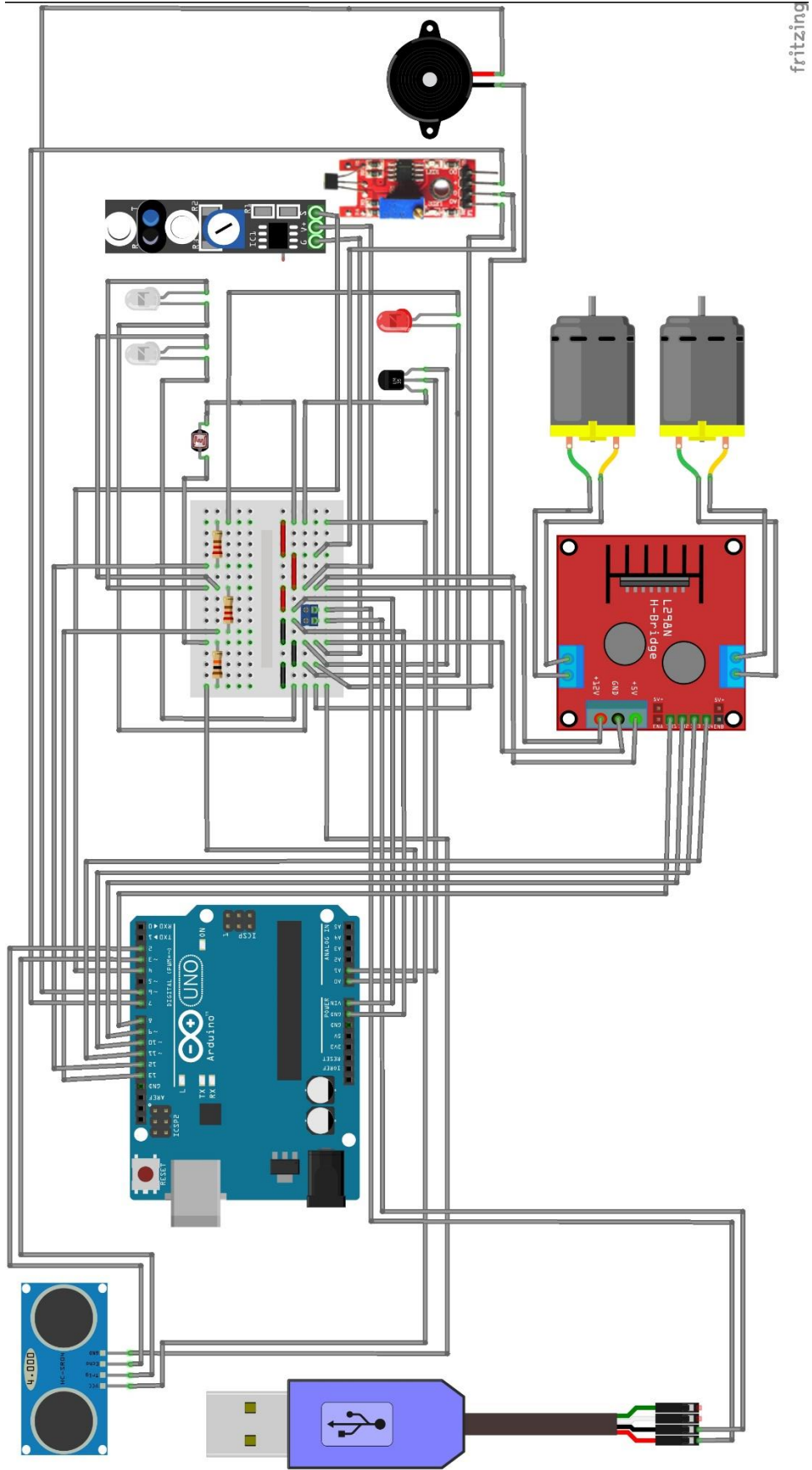


Diagrama General de Conexiones

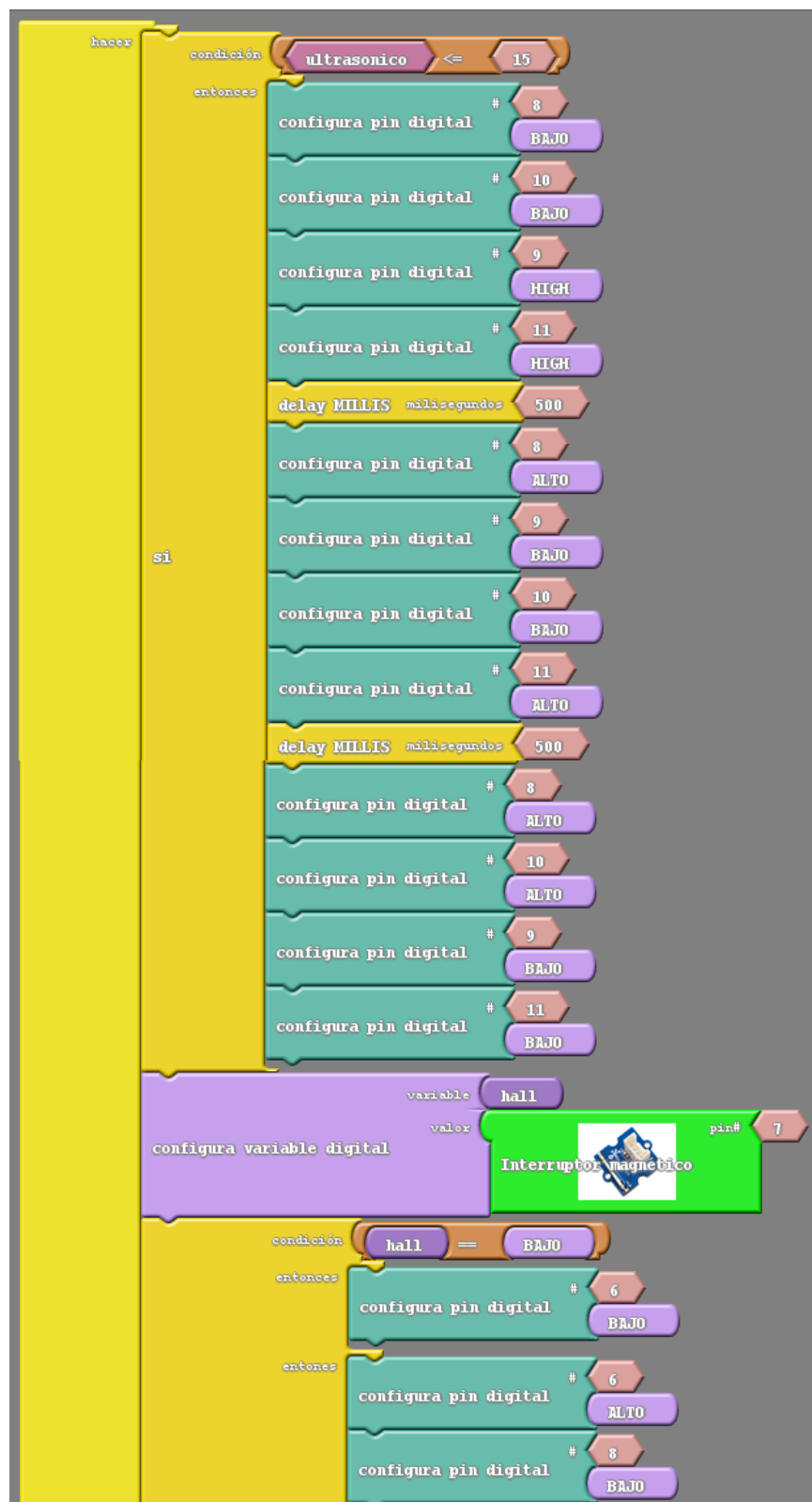


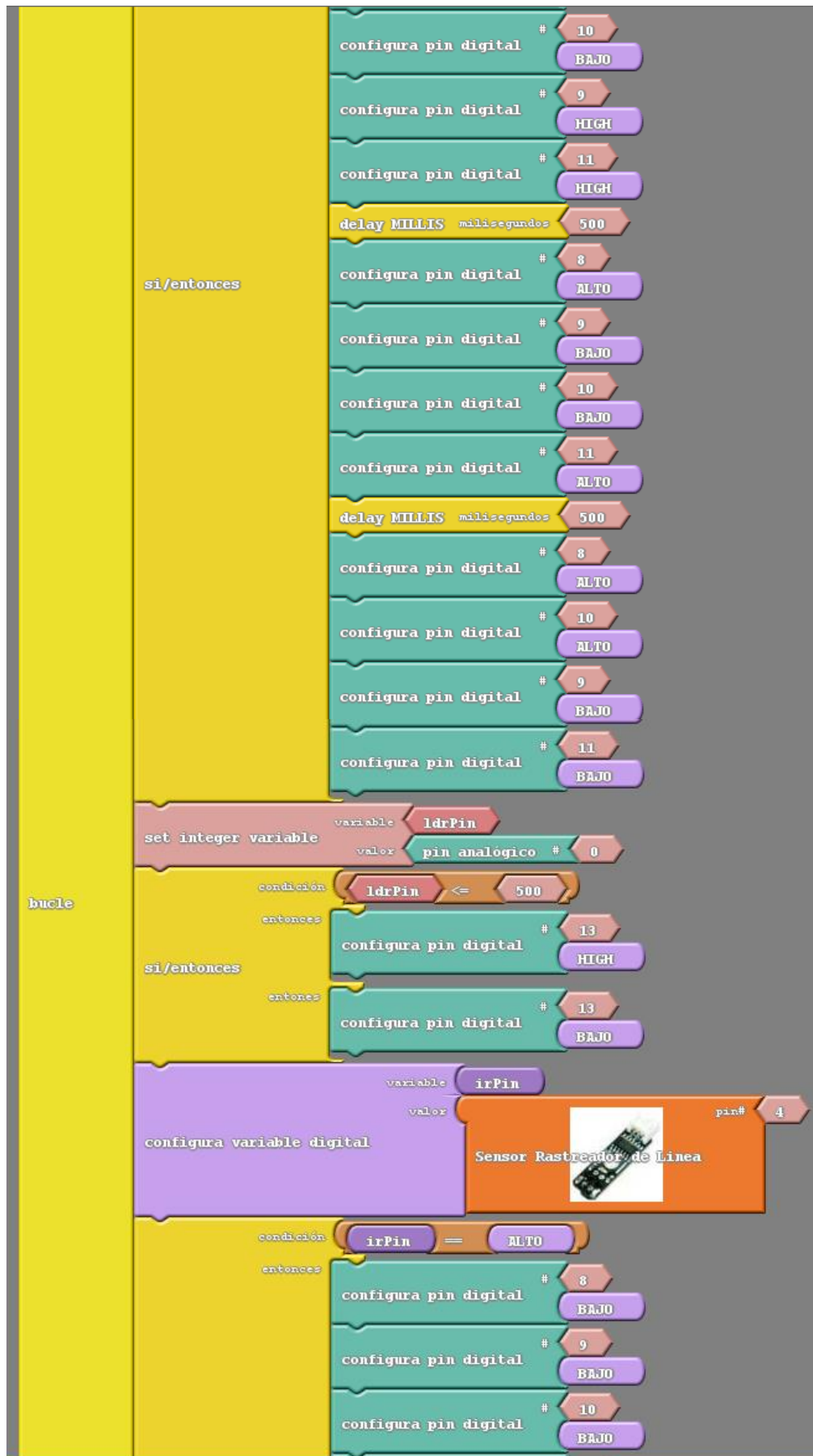


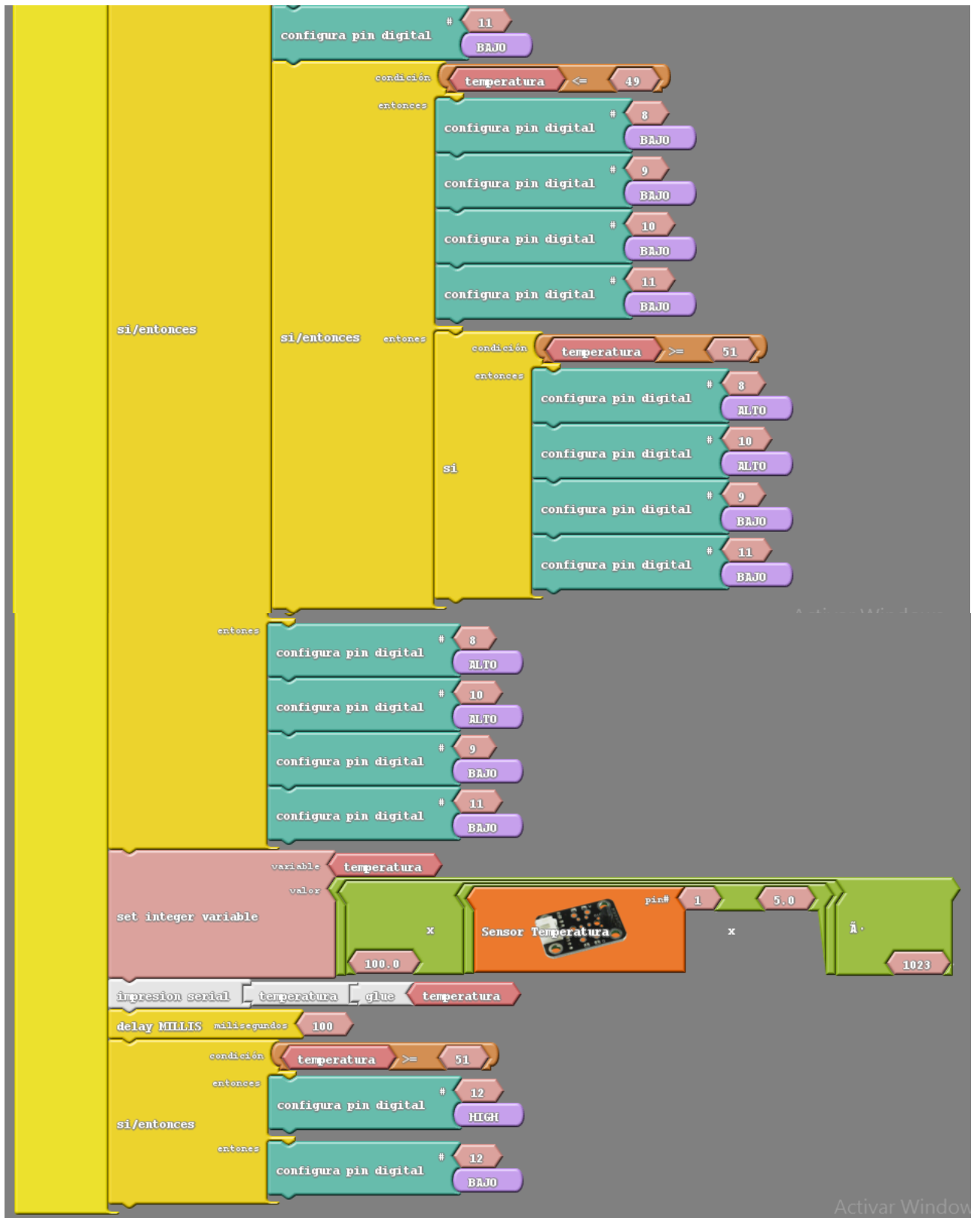
fritzing

Programa

Este programa integra los sensores y actuadores del Rover educativo

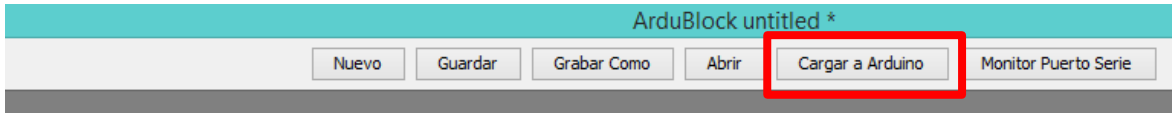






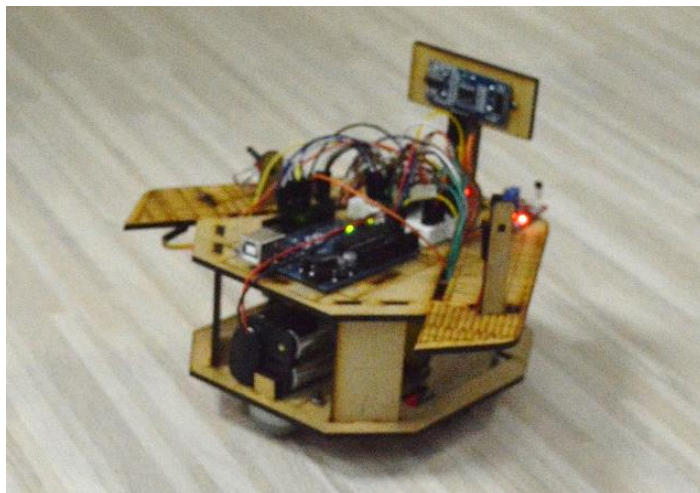
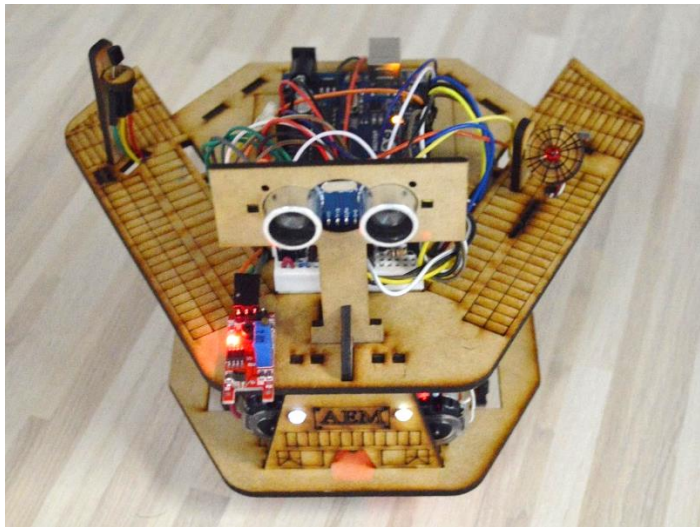
Puedes encontrar este programa en bloques, en la carpeta:
KitRoverEducativo\Rover\Ardublock\Rover.abp

Conecta la tarjeta *Arduino* mediante el cable hacia un puerto *USB* de la computadora y clic en el botón **Cargar a Arduino**.

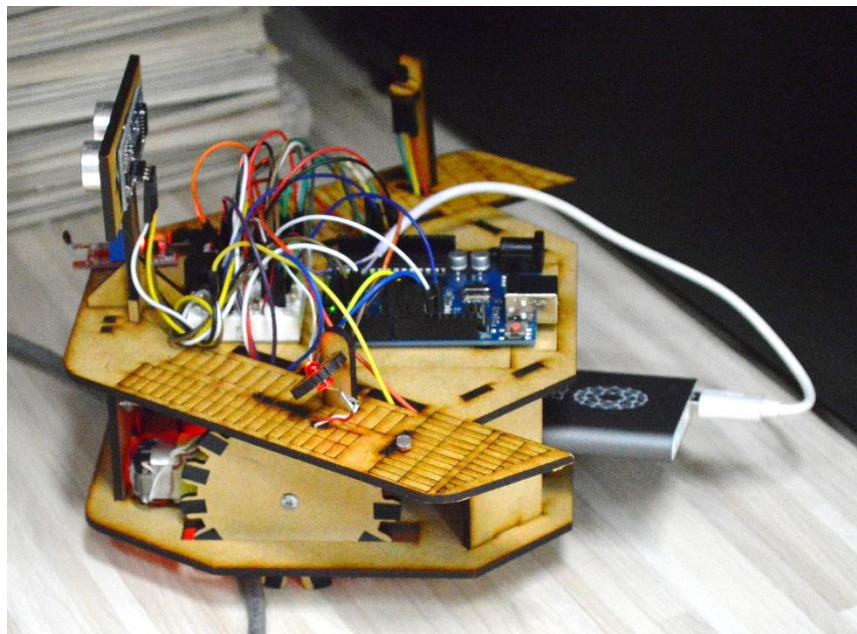
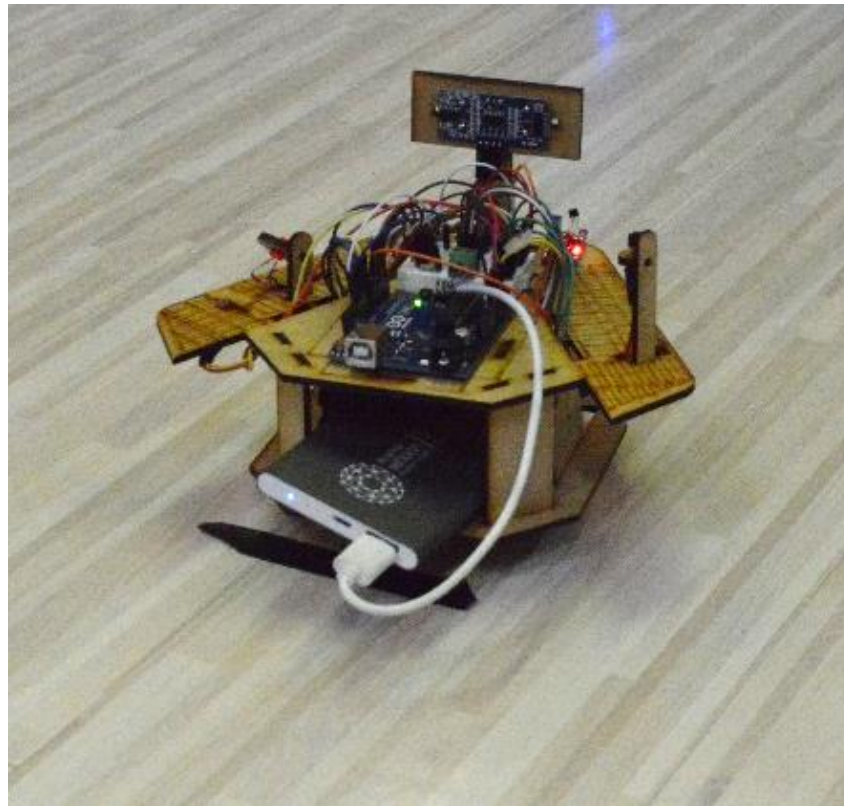


Espera a que se complete el proceso. Al finalizar, desconecta la placa *Arduino* y conecta las baterías o batería de respaldo.

a) Pilas AA

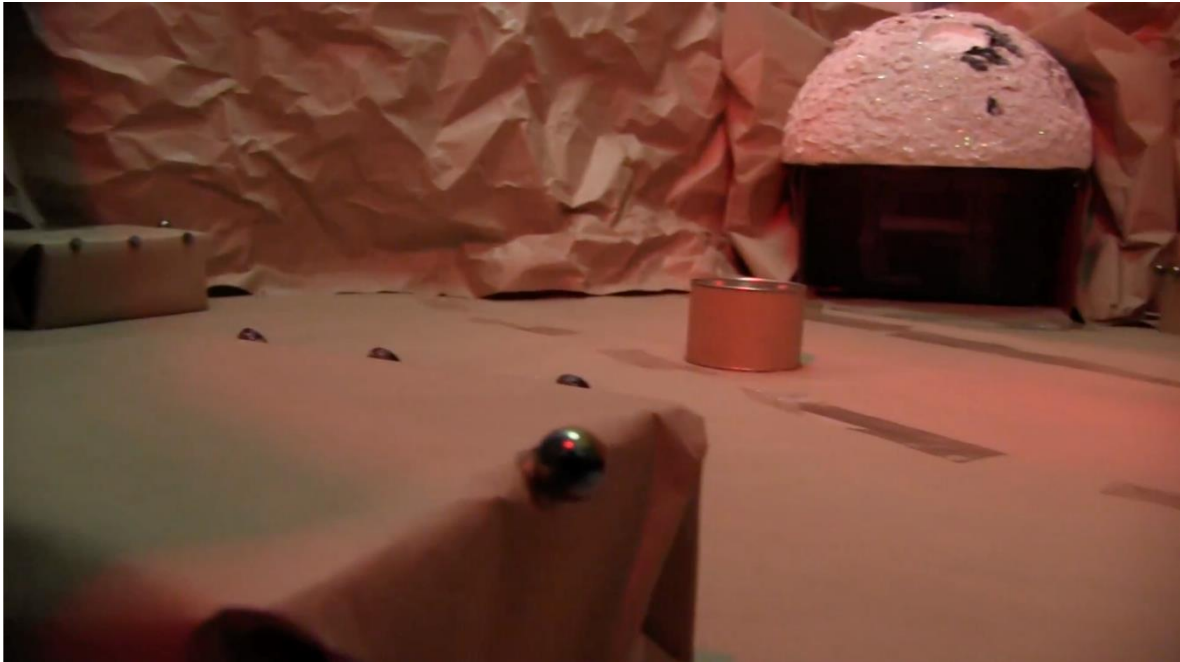


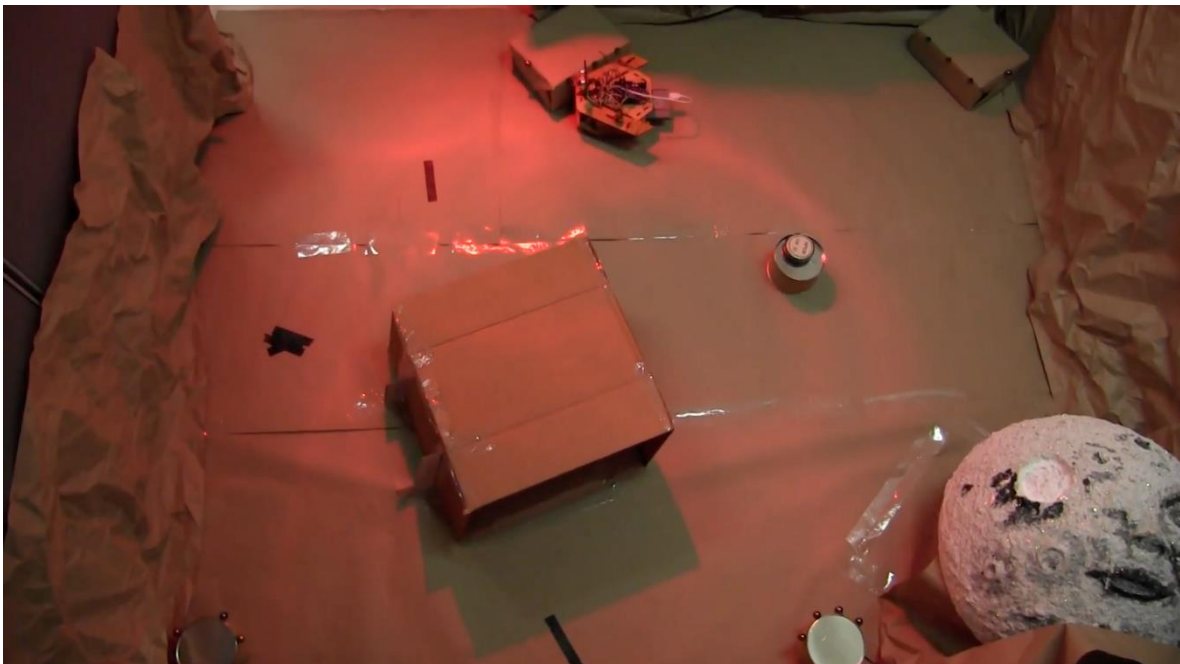
b) Batería recargable externa



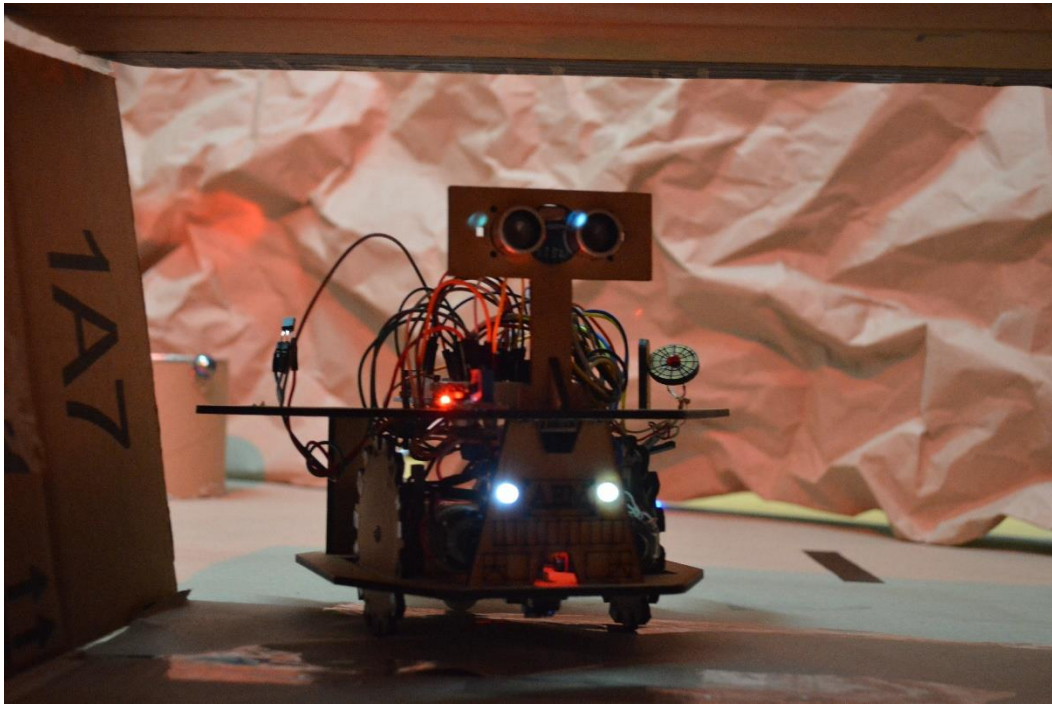
Rover Educativo

Una vez armado el Rover Educativo deberás ponerlo a prueba. Una propuesta es armar un ambiente de exploración:

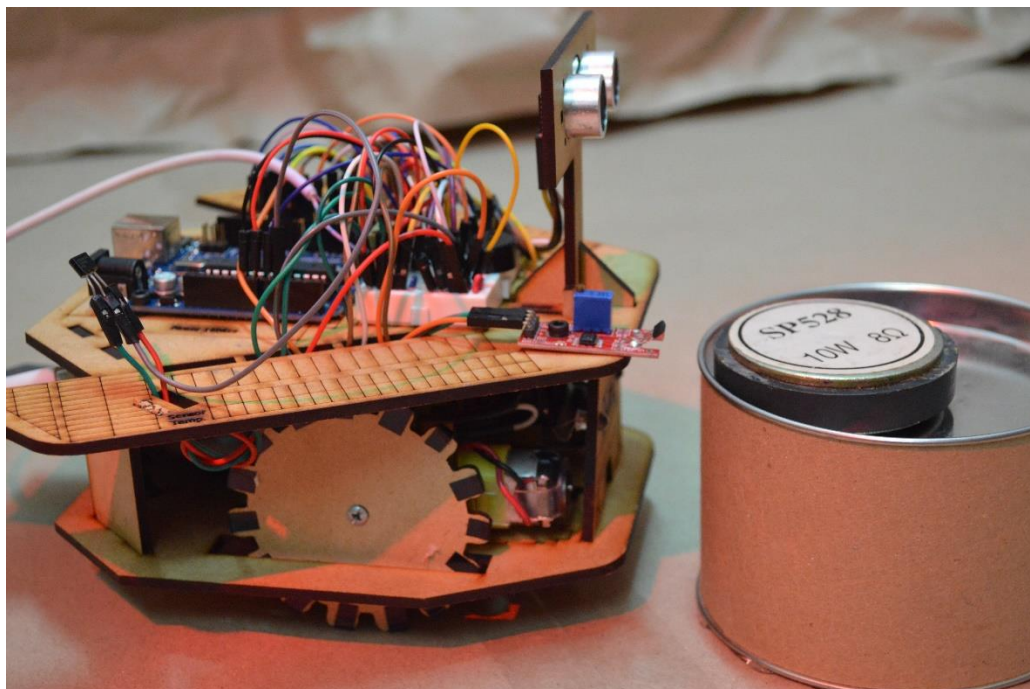


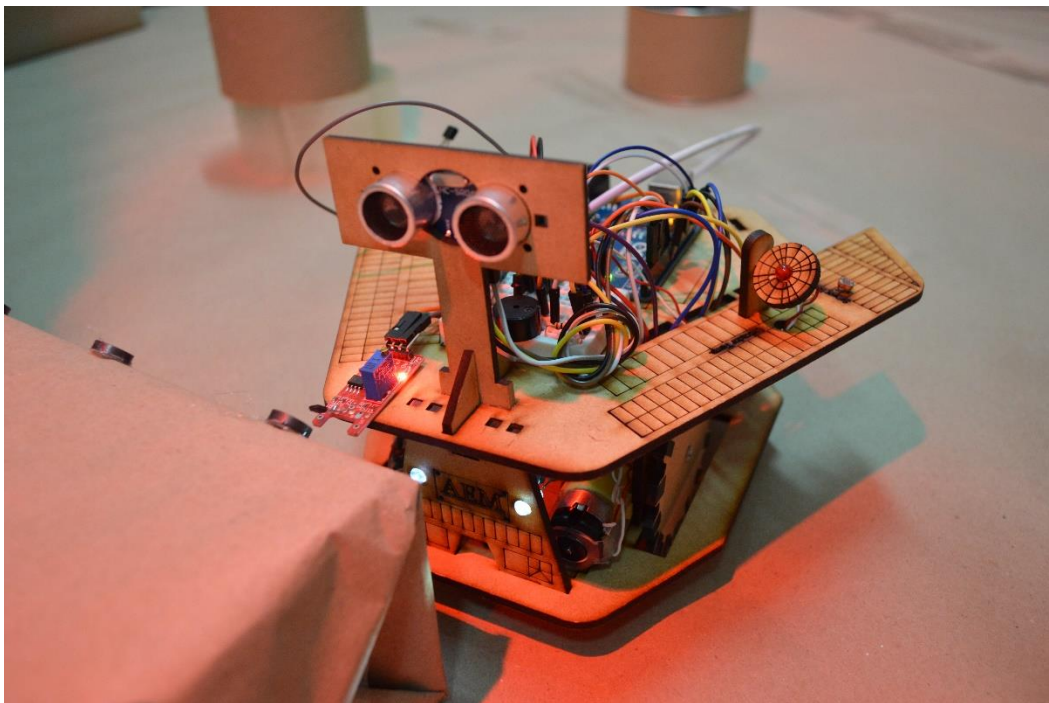
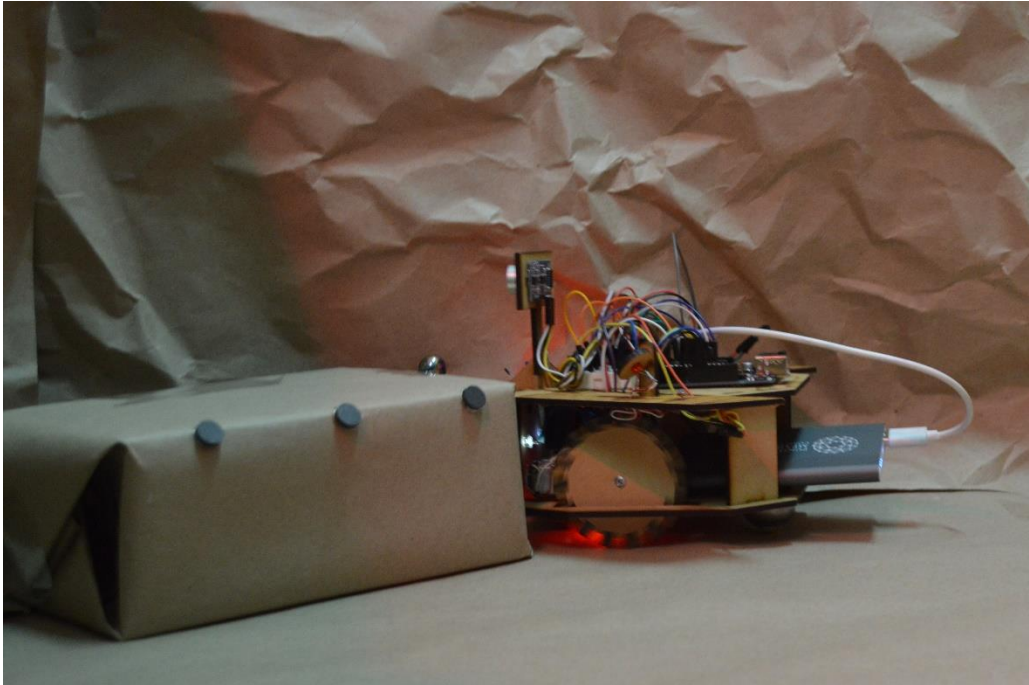


Puedes crear una cueva con baja iluminación para probar el sensor LDR y los LED ultrabrillantes:

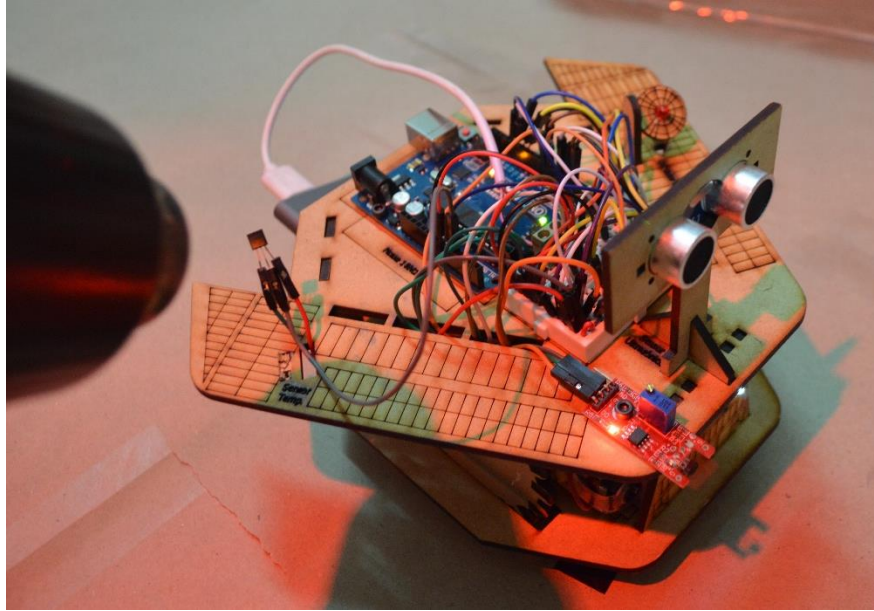


Adhiere algunos imanes a rocas o bases para que el Rover pueda identificar piedras magnéticas.

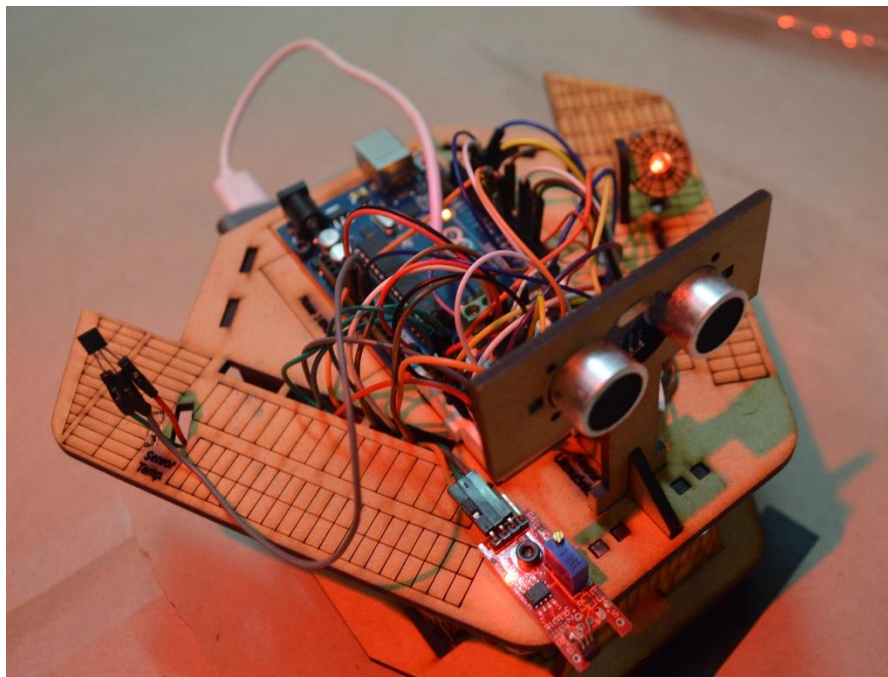




Utiliza una pistola de calor, una lámpara de terrario, o una cueva con calentadores te podrán servir para que el Rover pueda medir la temperatura del lugar. Recuerda usar una marca en el suelo de color negro para que el sensor IR le indique al Rover detenerse y comience a medir.



Observa que cuando la temperatura rebasa los 51 grados que el programa en Ardublock indica, el LED rojo se enciende:



Sugerencias y soluciones a posibles problemas

- Si ya tienes todo el Rover conectado y ya cargaste el programa y no avanza, verifica que los cables de los Motorreductores estén bien conectados al puente H.
- Revisa el buen funcionamiento de cada sensor por separado.
- Si notas que algún sensor no está funcionando correctamente, verifica que conectaste correctamente Vcc y GND en el protoboard además de verificar que la salida del Arduino es la correcta.
- Si notas que algún sensor dejó de funcionar verifica que los cables estén bien conectados.
- Si tu Rover solo mueve una llanta verifica que las salidas del puente H están correctamente conectadas al Arduino: N1=8, N2=9, N3=10 y N4=11.
- Si los LEDs conectados al sensor LDR no encienden, verifica que los cables estén conectados correctamente o no estén “fundidos”.
- Verifica que NO estén conectados directamente Vcc y GND del Arduino al protoboard para suplir de energía a todos los sensores, ya que puede calentar tu Arduino.
- Si detectas algún olor a “quemado” o visualizas humo desconecta inmediatamente la alimentación eléctrica de tu Rover (Pilas, PowerBank, Cable USB).
- Si tu Rover no avanza debido al tipo de suelo en el que se encuentre puedes colocar un poco de cinta de aislar en las puntas de las llantas.
- Si detectas que alguno de los leds no enciende, revisa que se encuentre conectado en el protoboard y al Arduino.
- Cuando programes, verifica que el cable USB se encuentre en buenas condiciones.
- Valida las versiones de software.

Pon a prueba tu conocimiento

El kit incluye el código en lenguaje C/C++ utilizado por la IDE Arduino para el Rover educativo. Este código no es tan visual como los bloques de Ardublock, sin embargo permite una mayor flexibilidad y capacidad de personalización. Adicionalmente, el código incluido, tiene una sección de verificación para el sensor de temperatura que permite tener mejor precisión.

```
1.  int trigPin = 3; //ULTRASONICO
2.  int echoPin= 2;
3.  int N1 = 8;    //PUENTE H
4.  int N2 = 9;
5.  int N3 = 10;
6.  int N4 = 11;
7.  float temperatura = 0; //TEMPERATURA
8.  int led = 12;
9.  boolean verificacion;
10. int hall = 7;   //HALL
11. int buzzer = 6;
12. int heState = 0;
13. int ledPin = 13; //LDR
14. int ldrPin = A0;
15. int ledPin2 = 13; //IR
16. int irPin = 4;
17. int tooClose = 0 ;
18.
19.
20. void setup()
21. {
22.   Serial.begin(9600);
23.   pinMode(trigPin, OUTPUT);
24.   pinMode(echoPin, INPUT);
25.   pinMode(N1, OUTPUT);
26.   pinMode(N2, OUTPUT);
27.   pinMode(N3, OUTPUT);
28.   pinMode(N4, OUTPUT);
29.   pinMode(led, OUTPUT);
30.   pinMode(hall, INPUT);
31.   pinMode(buzzer, OUTPUT);
32.   pinMode(ldrPin, INPUT);
33.   pinMode(ledPin, OUTPUT);
34.   pinMode(ledPin2, OUTPUT);
35.   pinMode(irPin, INPUT);
36.
37.   adelante();
38.
39. }
40.
```

```

41. int distancia()
42. {
43.     int duration, distance;
44.     digitalWrite(trigPin, HIGH);
45.     delayMicroseconds(1000);
46.     digitalWrite(trigPin, LOW);
47.     duration = pulseIn(echoPin, HIGH);
48.     distance = duration*0.034/2;
49.     //Serial.println(distance);
50.     return distance;
51. }
52.
53.
54. void adelante()
55. {
56.     digitalWrite(N1, HIGH);
57.     digitalWrite(N3, HIGH);
58.     digitalWrite(N2, LOW);
59.     digitalWrite(N4, LOW);
60.     //Serial.println("adelante");
61. }
62. void atras()
63. {
64.     digitalWrite(N1, LOW);
65.     digitalWrite(N3, LOW);
66.     digitalWrite(N2, HIGH);
67.     digitalWrite(N4, HIGH);
68.     //Serial.println("atras");
69. }
70. void girar()
71. {
72.     digitalWrite(N1, HIGH);
73.     digitalWrite(N2, LOW);
74.
75.     digitalWrite(N3, LOW);
76.     digitalWrite(N4, HIGH);
77.
78.
79.     //Serial.println("girar");
80. }
81. void detener()
82. {
83.     digitalWrite(N1, LOW);
84.     digitalWrite(N3, LOW);
85.     digitalWrite(N2, LOW);
86.     digitalWrite(N4, LOW);
87.     //Serial.println("stop");
88. }
89.
90. int fnTemperatura() //Se declara la funcion
91. {
92.     int coleccionTemp [] = {0,0,0,0,0,0,0,0,0,0}; //array para guardar las temperaturas

```

```

93.   int score [] = {0,0,0,0,0,0,0,0,0}; //array que guarda las veces que se repite cierta
    temperatura
94.   for(byte i=0; i<10; i++) //For para realizar 10 lecturas de temperatura
95.   {
96.       int temperatura = analogRead(1); //lee la temperatura, remmplazar tusensor por el pin
    del sensor
97.       delay(10); //delay para garantizar una buena medicion
98.       for(byte j=0; j<10; j++) //for para comparar la temperatura medida con las temperaturas
    almacenadas
99.       {
100.        if(temperatura == coleccionTemp[j]) // si la temperatura medida es igual a una que ya
    existe entonces suma un punto en el score
101.        {
102.            score[j] += 1;
103.            j = 10; // j se iguala a 10 para salir del for, ya que el score ya fue actualizado
104.        }
105.        else if(coleccionTemp[j] == 0) // si la el registro contiene un 0 quiere decir que es un
    espacio vacio por lo que ahi se guardara la temperatura nueva
106.        {
107.            coleccionTemp[j] = temperatura;
108.            j = 10; // j se iguala a 10 ya que la nueva temperatura ya fue almacenada
109.        }
110.    }
111. }
112. byte temp = coleccionTemp[0]; // la variable temp se iniicializa con el primer valor de la
    coleccion de temperaturas
113. byte valorAlto = score[0]; // la variable valorAlto se inicializa con el primer valor de la
    coleccion de scores
114. for(byte k=1; k<10; k++) // for para revisar la coleccion de scores y determinar cual es el
    mas alto ya que el score mas alto representa el valor que se repitio mas
115. {
116.     if(score[k] > valorAlto) // si el score k es mas alto que el valorAlto registrado entonces
    se actualiza la informacion
117.     {
118.         valorAlto = score[k]; //valorAlto se actualiza ya que score[k] fue mas alto por lo tanto
    ahora hay un nuevo valor mas alto
119.         temp = coleccionTemp[k]; // temp toma el valor de la temperatura con el score mas
    alto
120.     }
121. }
122. return temp;
123. }
124.
125. void loop()
126. {
127.     if (distancia() <= 15)
128.     {
129.         atras();
130.         delay(500);
131.         girar();
132.         delay(500);
133.         adelante();

```

```

134.
135. }
136.
137. heState = digitalRead(hall); //HALL
138. if(heState==LOW)
139. {
140.     digitalWrite(buzzer, LOW);
141. }
142. else
143. {
144.     digitalWrite(buzzer, HIGH);
145.     atras();
146.     delay(500);
147.     girar();
148.     delay(500);
149.     adelante();
150. }
151.
152.
153. int ldrStatus = analogRead (ldrPin); //LDR
154. if(ldrStatus <= 500)
155. {
156.     digitalWrite (ledPin, HIGH);
157. }
158. else
159. {
160.     digitalWrite(ledPin, LOW);
161. }
162.
163.
164.
165. tooClose = digitalRead(irPin);
166. if(tooClose == HIGH)
167. {
168.     detener();
169.     if(fnTemperatura()<=49)
170.     {
171.         detener();
172.     }
173.     else if(fnTemperatura()>=51)
174.     {
175.         adelante();
176.     }
177. }
178.
179. else
180. {
181.     adelante();
182. }
183.
184.
185. temperatura = ((100.0*(analogRead(1)*5.0))/1023);

```



```
186. Serial.println(temperatura);
187. delay (100);
188. if(fnTemperatura()>=51)
189. {
190. digitalWrite(led, HIGH);
191. }
192. else
193. {
194. digitalWrite(led, LOW);
195. }
196.
197.
198. }
```

Puedes encontrar este programa, en la carpeta:

KitRoverEducativo\Rover\PROGRAMA_ARDUINO\PROGRAMA_ARDUINO.ino

Conecta el cable USB a la placa Arduino y presiona el botón “Subir”:



Espera a que se complete el proceso. Al finalizar, desconecta la placa Arduino y conecta las baterías o batería de respaldo.

Bibliografía recomendada

- Foro oficial en español: <https://forum.arduino.cc/index.php?board=32.0>
- Introducción a Arduino, Massimo Banzi, Editorial Anaya Multimedia, 2016
- Taller de Arduino: Un enfoque práctico para principiantes, Germán Tojeiro Calza, Editorial Marcombo, 2014
- Arduino práctico, Daniel Lozano Equisoain, Editorial Anaya Multimedia, 2017
- 30 proyectos con Arduino para el genio malvado, Simon Monk, Editoria Estribor 2012.

