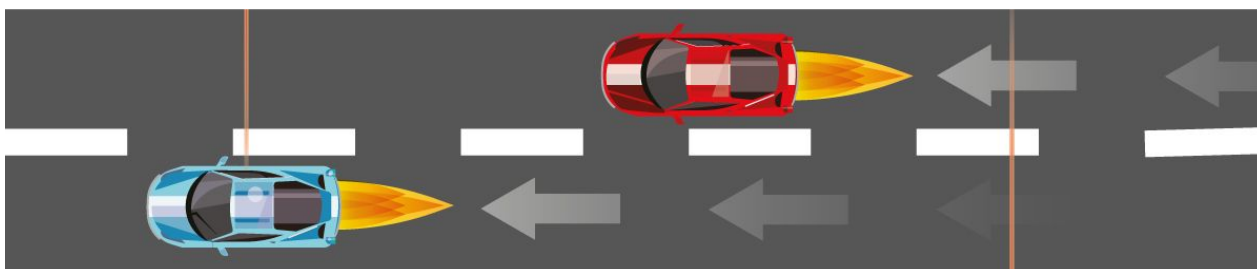


# Construye y programa tu radar de tramo



## Breve descripción del proyecto

En este proyecto vas a descubrir cómo construir y programar tu propio radar de tramo para poder medir la velocidad a la que se desplaza un objeto.

Este sistema cuenta con dos cámaras o radares separados entre sí a una distancia determinada, con el fin de detectar el tiempo que tarda un vehículo en recorrerla y calcular así la velocidad.

**Nivel de dificultad:** Medio

**Tiempo estimado:** 2 horas

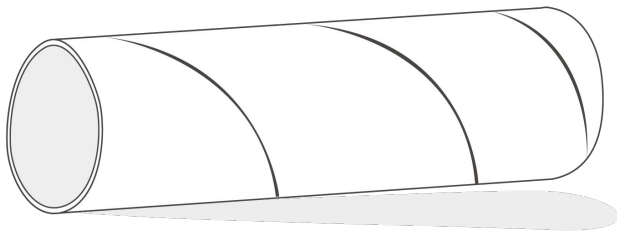
## Materiales:

Cartón o cartulina para construir la vía o túnel  
Tijeras  
Cinta adhesiva  
Punzón  
Ordenador  
Placa controladora  
2x LED  
2x Sensor de luz (LDR)  
Vehículo

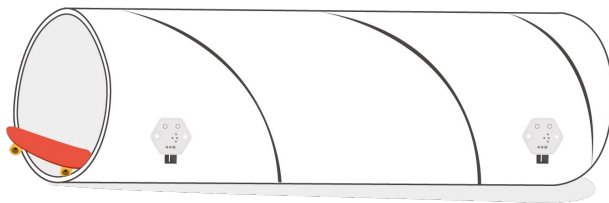
# Construye la estructura principal

## Pasos a seguir

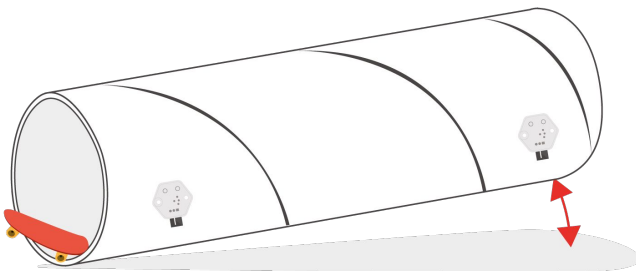
1. Lo primero que debes hacer es conseguir un trozo de cartón plano o cilíndrico de unos 30-50 cm de largo x 10-20 cm de ancho.



2. Deberás recortar el cartón y doblarlo para darle la forma deseada. La idea es conseguir recrear un carril o túnel por el que deslizar nuestro vehículo.



3. Debes agujerear los laterales del carril y colocar un LED enfrente a un sensor de luz. A una distancia de unos 16 cm o más, coloca otro LED enfrente a otro sensor de luz. Sujeta estos componentes en esa posición con celo.



4. Comprueba que están bien sujetos y que pueden aumentar la pendiente de la estructura sin que se caigan.

De esta forma tan sencilla, habrás terminado la construcción de la estructura principal. A continuación, vamos a aprender cómo programarlo.

# ¡A programar!

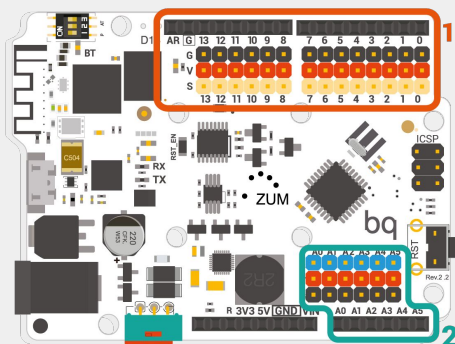
Antes de empezar, vamos a crear un programa en el que el sensor de luz detecte si hay luz u oscuridad, y en consecuencia, apague o encienda un LED.

## Recuerda...

Un componente digital es aquel que solo nos devuelve dos valores (0 o 1), mientras que uno analógico es capaz de devolvernos más de dos valores y un sensor detecta algo mientras que un actuador indica algo.

Por lo tanto, un LED será un actuador digital ya que solo tiene dos estados, o encendido (1) o apagado (0), y emite una señal luminosa que indica algo.

Por otro lado, el sensor de luz será un sensor analógico, ya que es capaz de detectar un valor de iluminación nada menos que de entre 0-800.

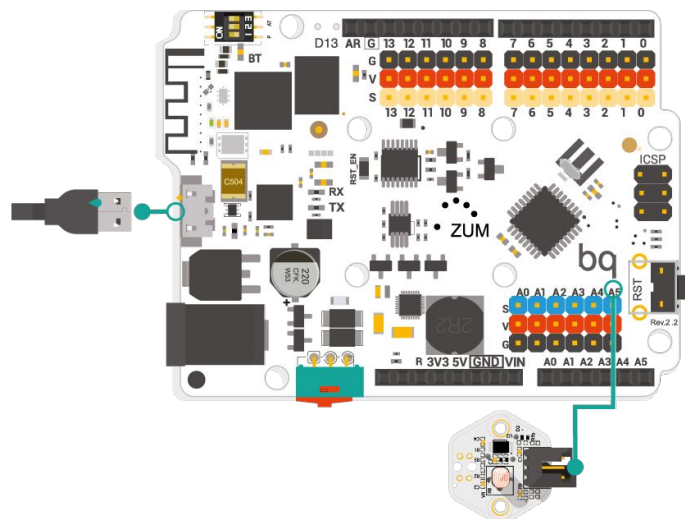


Conectamos el LED a uno de los pines digitales (los marcados con un 1 en la imagen) y el sensor de luz a uno de los pines analógicos (los marcados con un 7 en la imagen) .

## ¿Cuánta luz hay en mi aula?

Vamos a probar como funciona el sensor de luz con un ejercicio.

En primer lugar, conectamos a nuestra placa un sensor de luz y el puerto serie.



Una vez que has conectado ambos componentes, hay que programarlos. Para ello, utilizaremos el apartado *Bucle principal* (*loop*) al que añadimos el bloque mostrar por *puerto serie*, al que incluiremos el bloque *Leer sensor de luz* y finalmente una *Espera* de unos pocos segundos:

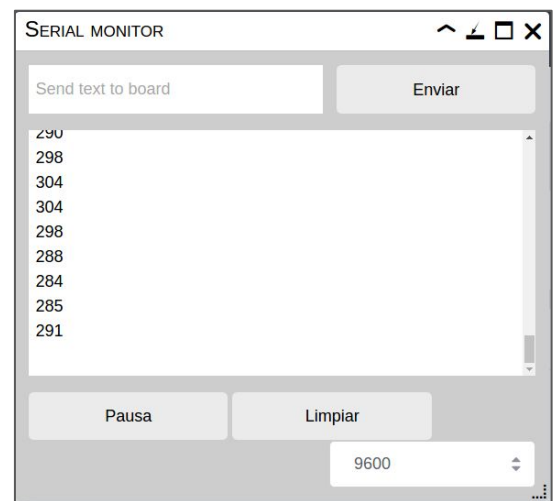
#### — Bucle principal (Loop)



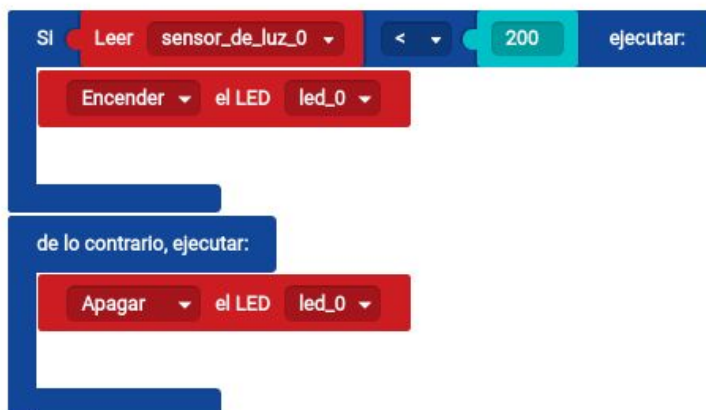
Conecta tu placa al ordenador, clicas sobre el botón cargar y selecciona “Mostar Serial Monitor” en la opción “Ver”. De esta forma, aparecerán los valores de luz detectadas por el sensor de luz en el aula.

Para más información, consulta los siguientes post:

- <http://diwo.bq.com/ve-hacia-la-luz-robotin-programando-el-sensor-de-luz/> - Sensor de luz
- <http://diwo.bq.com/comunicandose-con-puerto-serie-bitbloq2/> - Puerto serie



#### — Bucle principal (Loop)



## Detectando si hay o no luz

Para este ejercicio, conecta a tu placa el sensor de luz y un LED.

Utilizaremos el bloque condicional *Si... ejecutar...*, de manera que, si el sensor de luz detecta un valor menor de 200, encenderá el LED, de lo contrario, lo apagará.

Carga el programa en tu placa y comprueba cómo el LED permanece apagado cuando hay luz y se enciende cuando lo tapamos con la mano.

# Programando el radar de tramo

Para comenzar con la programación del radar, necesitaremos los siguientes componentes: 2 LED y 2 sensores de luz (LDR). Para poder realizar la programación, deberemos conocer cuál va a ser la función que van a tener estos componentes.



## ¿Qué función van a tener estos componentes?

Los LED y sensores de luz nos ayudarán a crear dos barreras de luz, con el fin de que actúen como radar y detecten que el paso del vehículo.

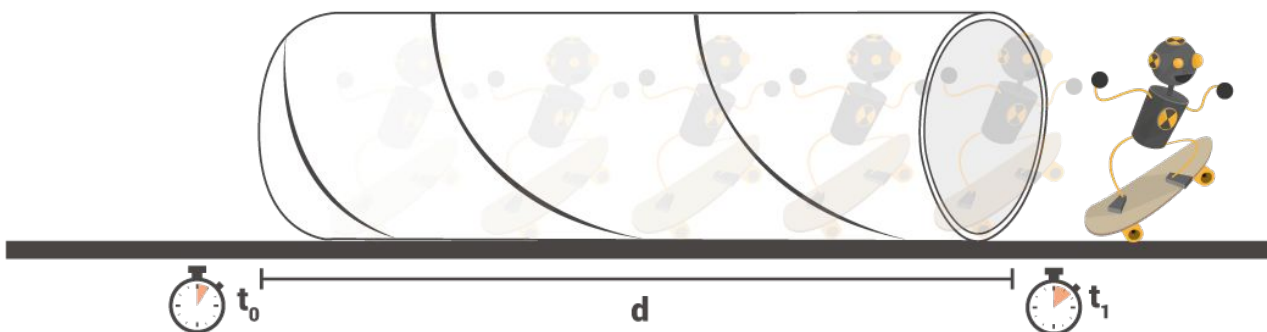
## ¿Cómo se detectará el paso del vehículo?

Nuestros LED iluminan los sensores de luz, y estos recibirán una cantidad de luz determinada. Al pasar el vehículo, tapará el haz de luz que envía el LED por lo que la cantidad de luz que reciba el LDR cambiará, detectando así su paso.

## ¿De qué forma calcularemos la velocidad a la que va el vehículo?

Para calcular la velocidad necesitamos conocer el espacio que va a recorrer el objeto, que en este caso se corresponderá con la distancia que hay entre las dos barreras, y el tiempo que tarda el vehículo en recorrer dicho espacio. Utilizaremos la siguiente fórmula:

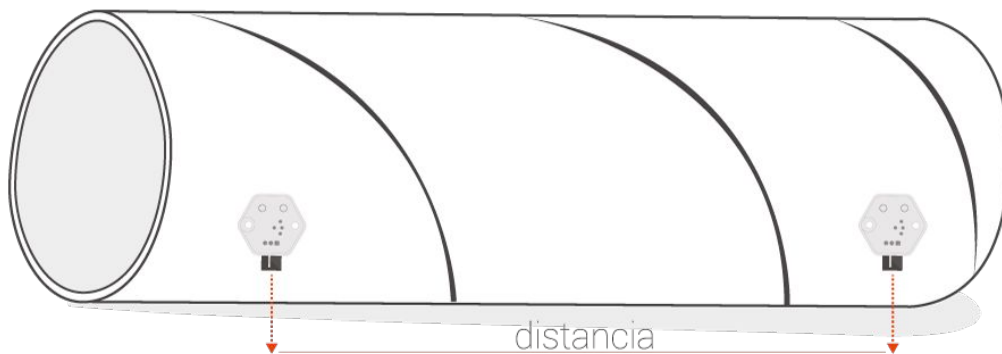
*Velocidad= distancia/tiempo*



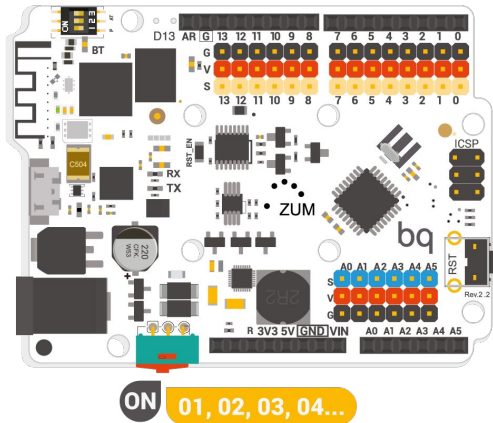
Una vez que conocemos lo que tenemos que programar, vamos a abrir Bitbloq y a conectar estos componentes a la placa controladora. Para facilitar la programación, es mejor que definamos el nombre de los componentes, de manera que distingamos el LED y LDR que van a formar el primer radar de los que van a formar el segundo.

Tras esto, iremos a la parte de *Software* y crearemos la programación siguiendo los siguientes pasos:

1. Para almacenar el tiempo que va a tardar el vehículo, la velocidad a la que va a ir y la distancia entre los radar, deberemos definir unas variables. ¿Cómo funcionan las variables? Consultamos el siguiente enlace para obtener más información sobre ellas <http://diwo.bq.com/programando-con-variables-en-bitbloq-2/>  
En la variable que guardemos la distancia, deberemos escribir el espacio que hay entre el primer radar y el segundo. Podemos medirlo con una regla. Es importante que lo pongamos en metros.



2. Deberemos calcular además, la cantidad de luz que reciben tanto el LDR inicial como el final. Para ello, no sólo deberemos crear una variable para almacenar cada dato, sino programar, que se compruebe al iniciarse la programación, qué cantidad de luz reciben los LDR en ese momento.  
Esto se hará con el fin de que cuando pase el vehículo por las barreras, el programa detecte un cambio de luz y realice la acción pertinente, en este caso calcular la velocidad a la que va y el tiempo que ha tardado.  
¿Cómo lo programamos? Encendiendo los LED al inicio y guardando en una variable la cantidad de luz que reciben los LDR.
3. Por último, deberemos crear la programación que calcule la fórmula de la velocidad. Para ello, lo primero que tenemos que hacer es indicarle al programa que cuando el objeto pase por el primer radar, obtenga el tiempo en el que pasa y lo guarde en la variable tiempo que hemos creado.

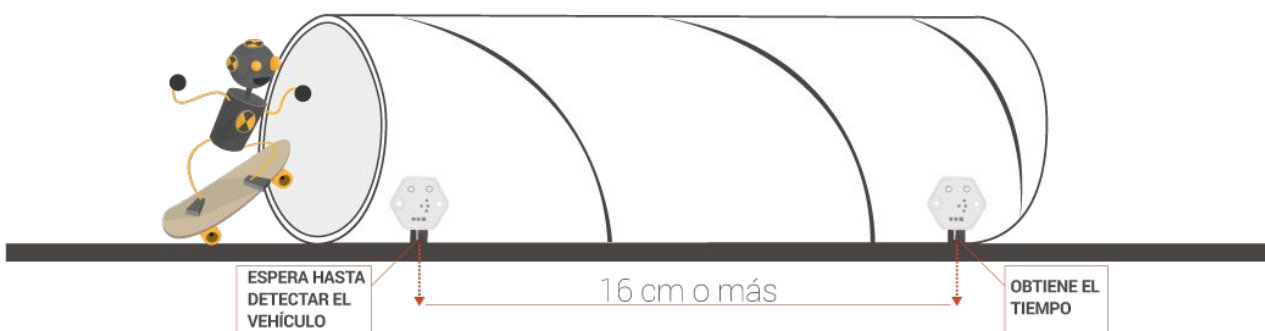


Para medir el tiempo, deberemos conocer el bloque *Obtener tiempo de ejecución*. Cuando se inicia el programa, la placa controladora empieza a contar el tiempo en segundos que lleva encendida.

Para calcular el tiempo que tarda el vehículo en pasar por los dos radar, utilizaremos este bloque.

**Obtener tiempo de ejecución**

Tras esto, deberemos indicar qué tiene que hacer el segundo radar mientras no detecte la presencia del objeto. En este caso, deberemos utilizar el bloque Mientras con el que haremos esperar al segundo radar para que no realice ninguna acción hasta que no detecte la presencia del vehículo.





Una vez que el radar final detecta la presencia del objeto, deberemos programar que se calcule la fórmula de la velocidad. Para ello, tendremos que seguir los siguientes pasos:

- **Obtener el tiempo:** Obtenemos el tiempo en el que el vehículo pasa por el segundo radar y le restamos al tiempo tomado cuando se detectó el vehículo por el primer radar. Además, deberemos cambiar la unidad de tiempo de milisegundos, que es el tiempo con el que trabaja nuestro programa a segundos.
- **Obtener la velocidad:** utilizando la distancia y el tiempo obtenido.

Para terminar nuestro programa, mostramos por el puerto serie el tiempo que ha tardado y la velocidad a la que ha ido el vehículo cuando pasaba por nuestro radar de tramo. Deberemos dejar un tiempo para indicarle al programa cuándo debe ir actualizando y mostrando los datos obtenidos.

