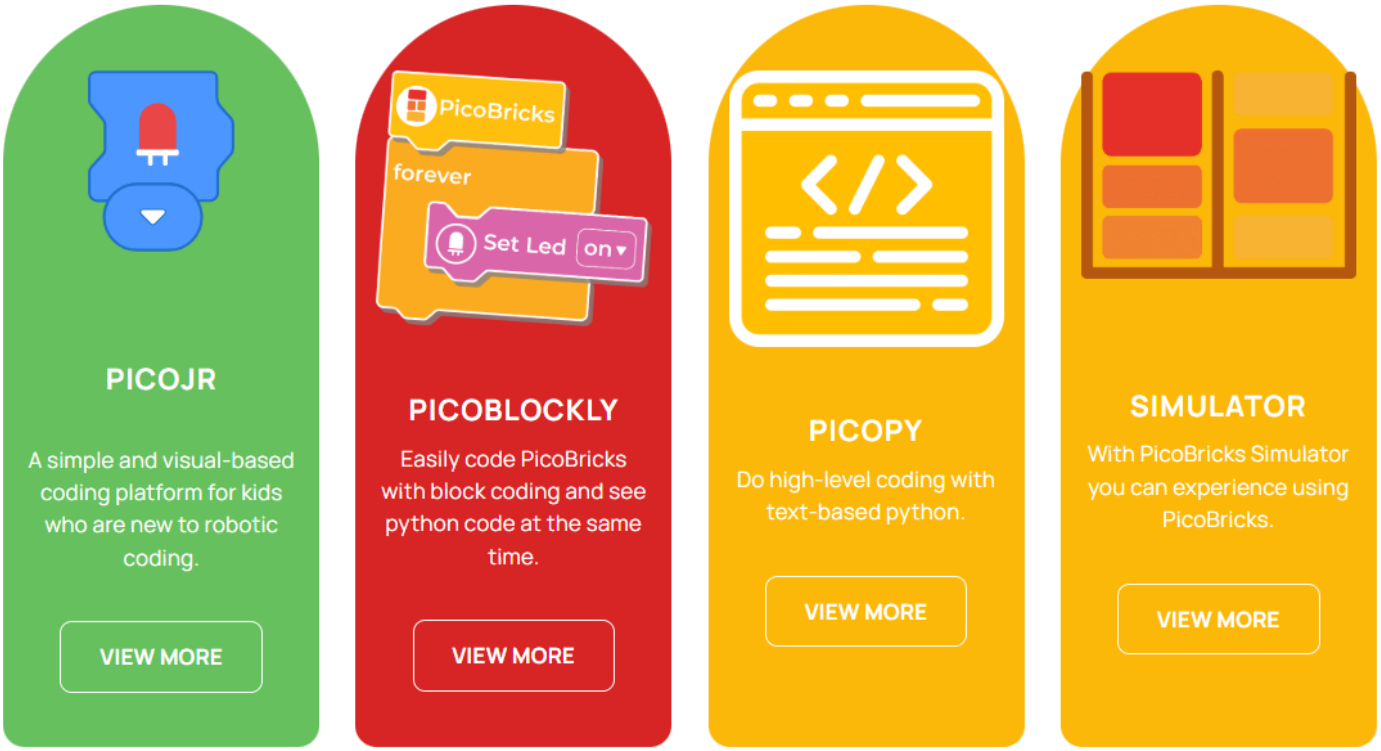
PicoBlockly

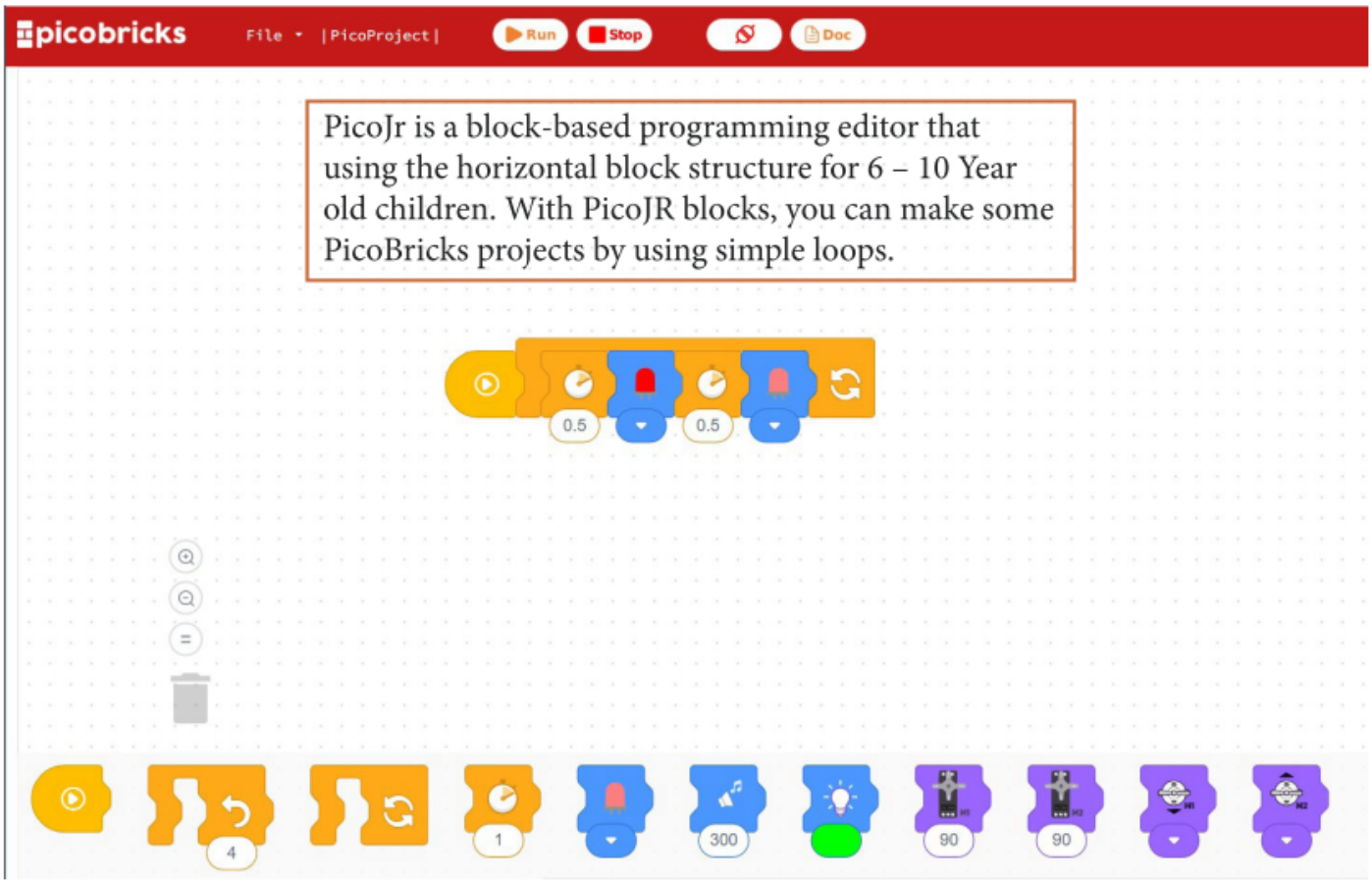
Cuatro programas a elegir

Si entramos en http://rbt.ist/ide podemos ver cuatro opciones



PICOJR

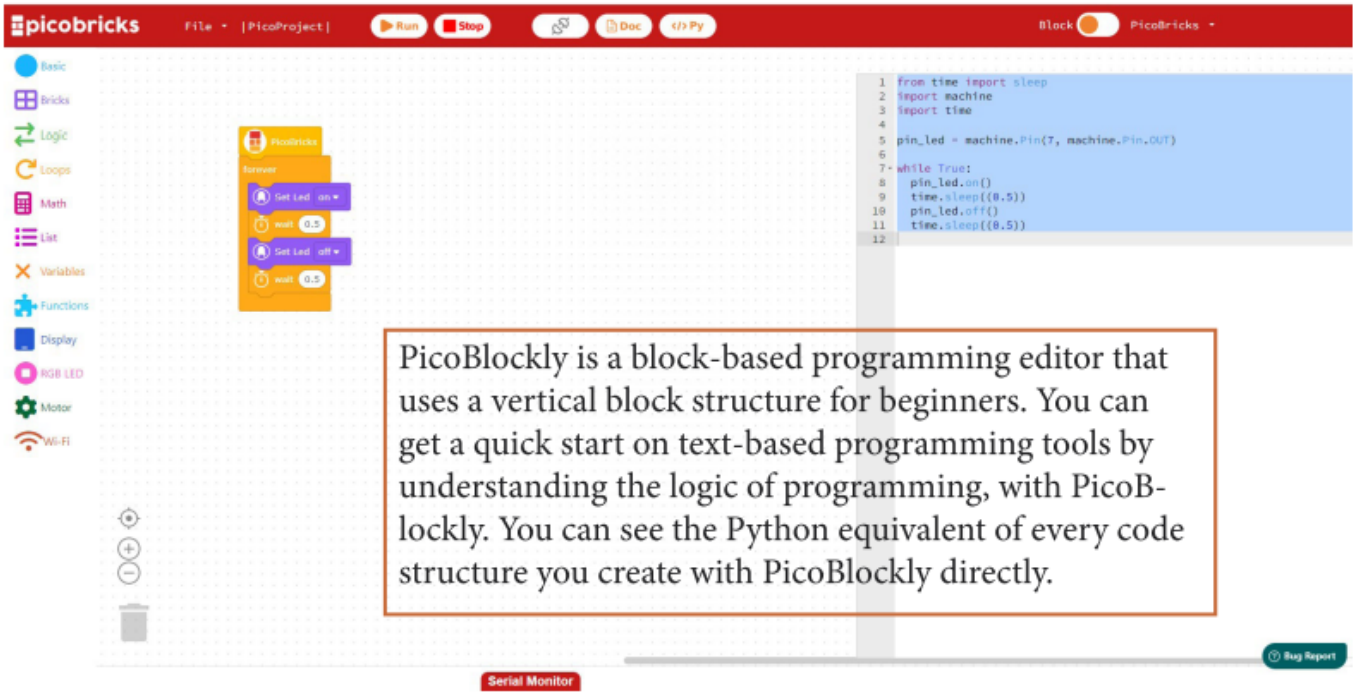
Pensado para programar Picobriks con bloques para etapas de 8 a 10 años con un **mínimo de instrucciones**



Fuente Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook ver créditos

PicoBlockly

Es la opción más recomendada para la mayoría de las etapas



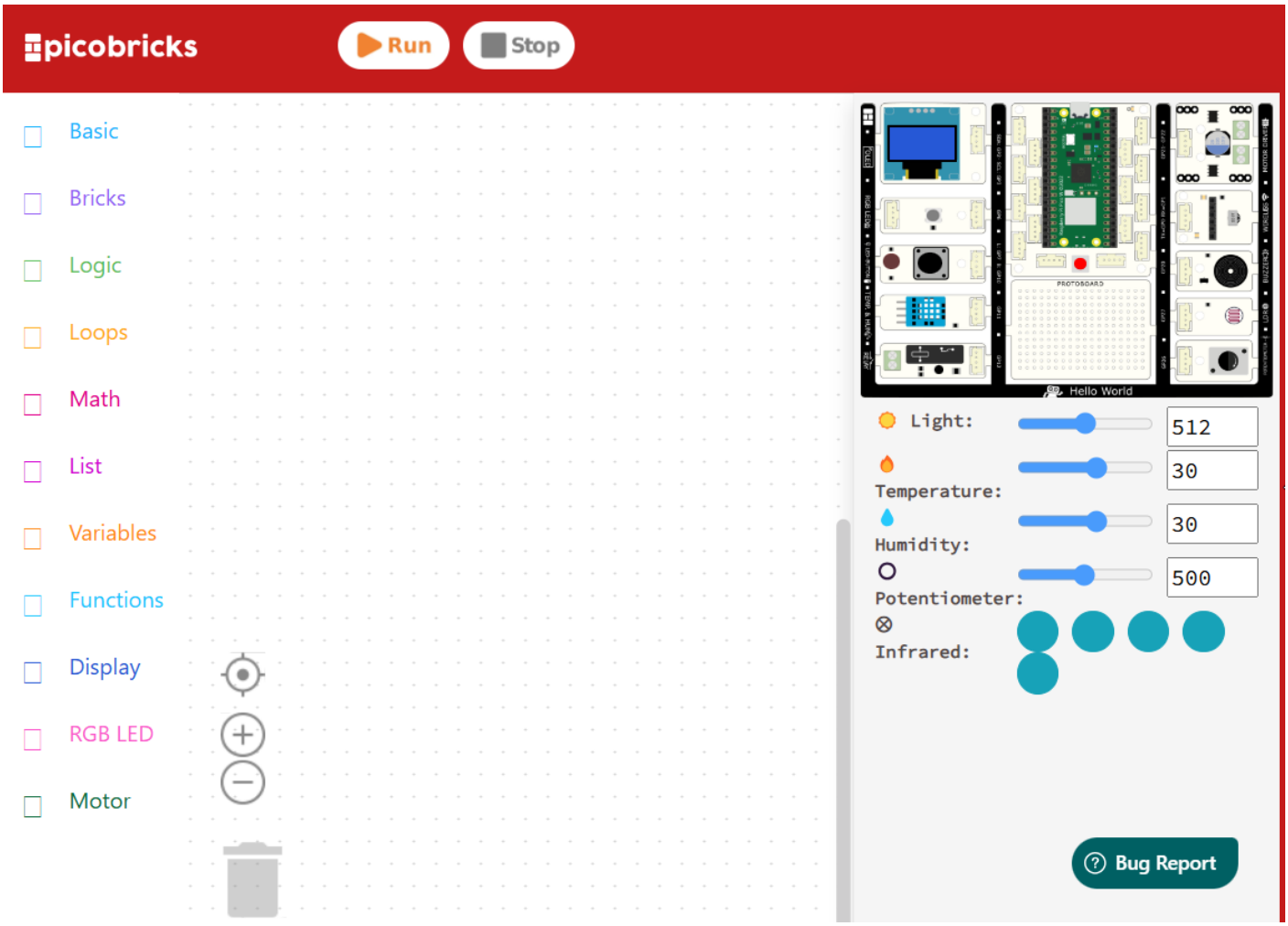
Fuente Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook ver créditos

PicoPy

Para poder editar en Python, no lo intentes por aquí, no va. Para trabajar con Python se trabaja con Thommy que lo trataremos en este curso.

Pico simulator

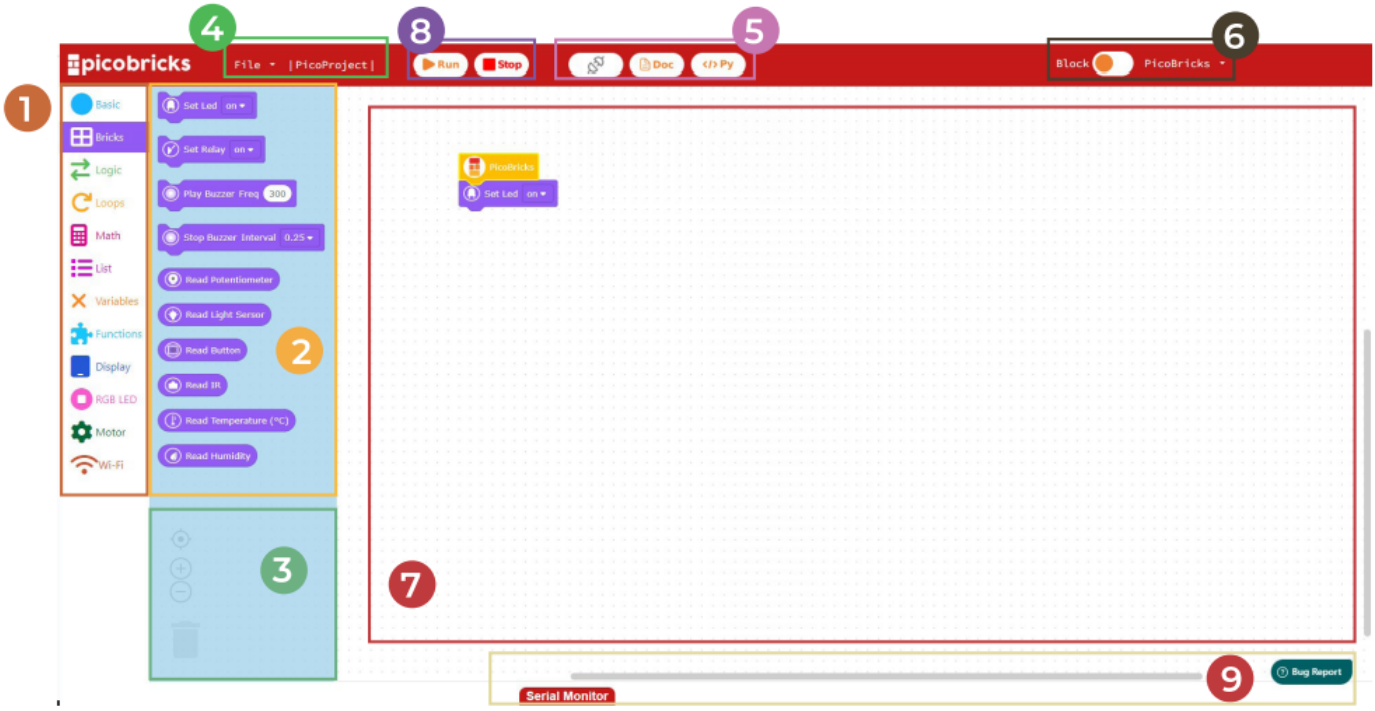
Es un simulador online que permite realizar proyectos sin tener físicamente la Pico bricks



Ojo el simulador no permite gestión de ficheros, es decir, no puedes ni grabar proyectos ni abrirlos, cuando cierras el navegador se pierde todo

Interface

Cuando abrimos Picoblockly tenemos la siguiente ventana:



Fuente Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook ver créditos

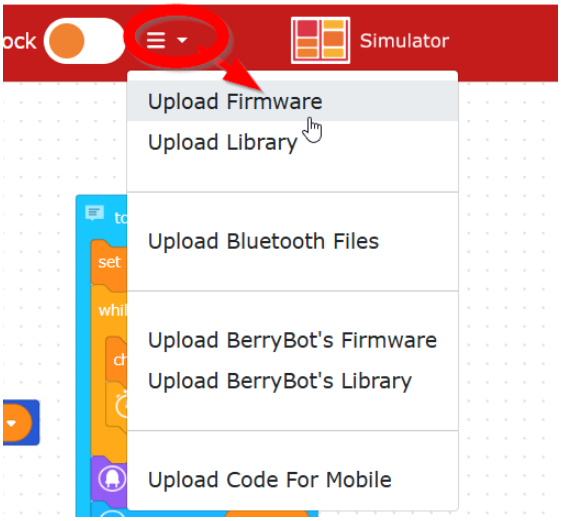
1. Donde encontramos las diferentes instrucciones ordenadas por categorías
2. La paleta de instrucciones preparados para elegir y arrastrar a 7
3. Herramienta de zoom, borrar
4. Menú de fichero para grabar los proyectos o abrirlos (todo localmente)
5. Panel operaciones
   1. Botón de conectar, por cable (recomendado) o bluetooh
   2. Botón de proyectos ya preinstalados
   3. Vista de código Python (también en 6 hay una pestaña para pasar a esta vista)
6. Menú de configuración para descargar los firmwares necesarios para la conexión
7. Área donde programamos
8. Start stop tu programa
9. Área del puerto serie donde podemos ver los valores que desemos

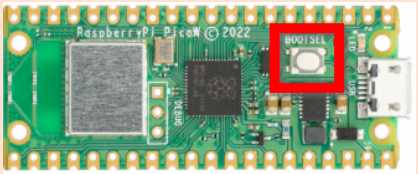
Conexión

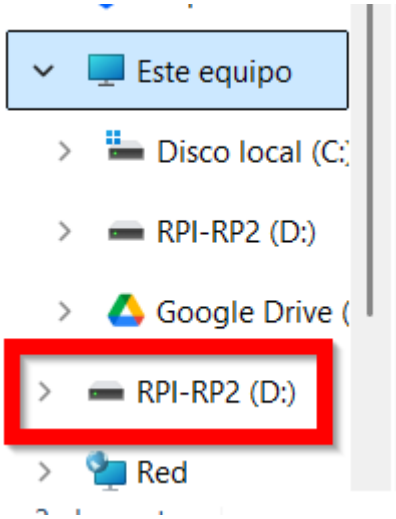
Lo primero que tenemos que hacer es poner el firmware para podernos conectar con Picobriks

> **P: ¿Qué es eso de "firmware"?**
> R: No es más que un software que se graba en los chips de la placa.
> **P ¿Y por qué se llama así, y no se llama software o programa y en paz?**
> R: Digamos que como se graba en los chips, es un medio camino entre software y hardware, para diferenciarlo del software habitual.

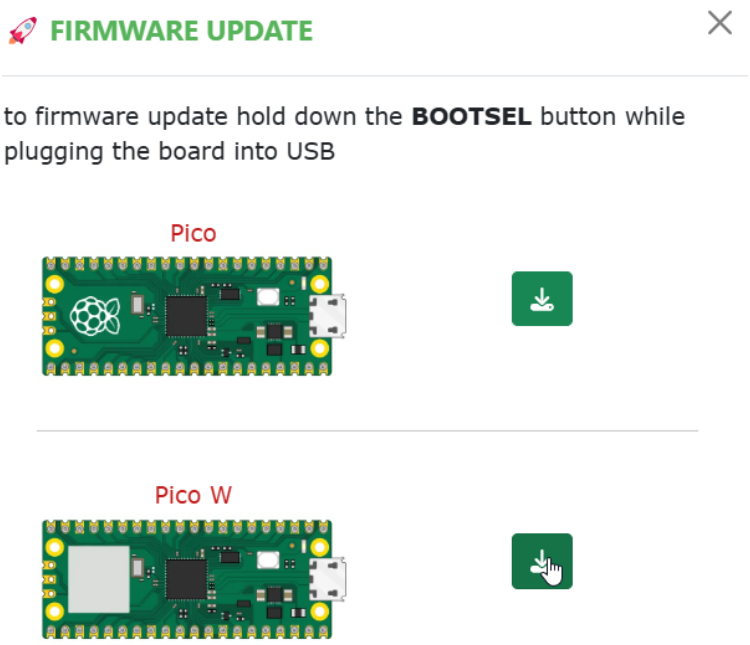
Entramos en el menú y descargamos el firmware

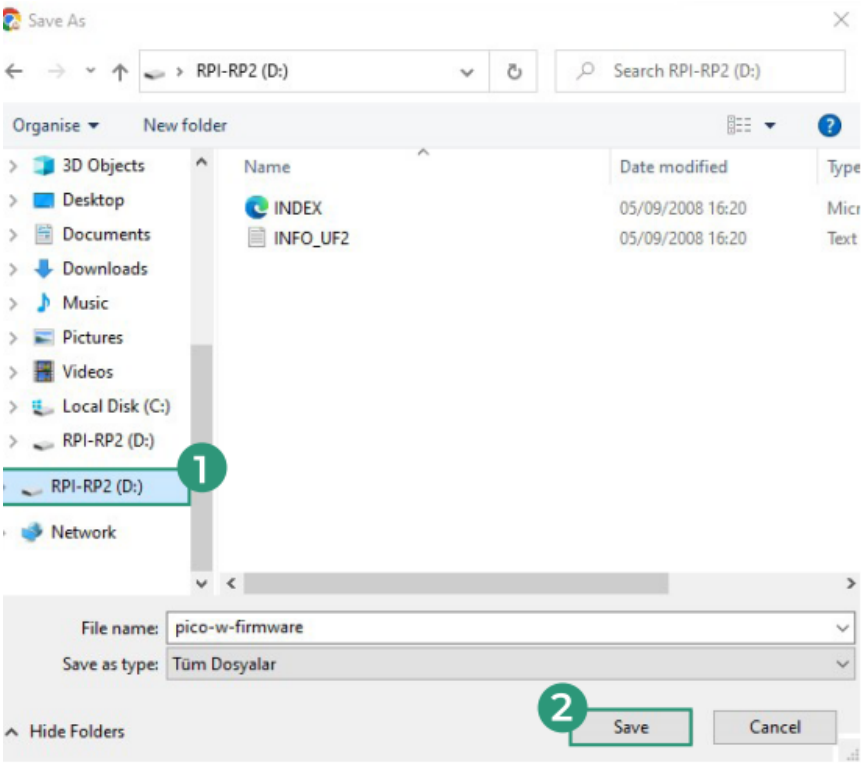


> **ATENCIÓN, poner PicoBricks en modo Bootloader**
> 1.-Desconectamos PicoBricks de nuestro ordenador
> 2.- Apretamos el botón BOOTSEL **mientras** lo volvemos a conectar al puerto USB
> 
> 3.- Automáticamente aparecerá una nueva unidad de disco en nuestro ordenador (ya puedes soltar BOOTSEL)



Descargamos el correspondiente al **PicoW** **Y LO GRABAMOS EN LA UNIDAD NUEVA** en mi caso RPI-RP2 (D:)



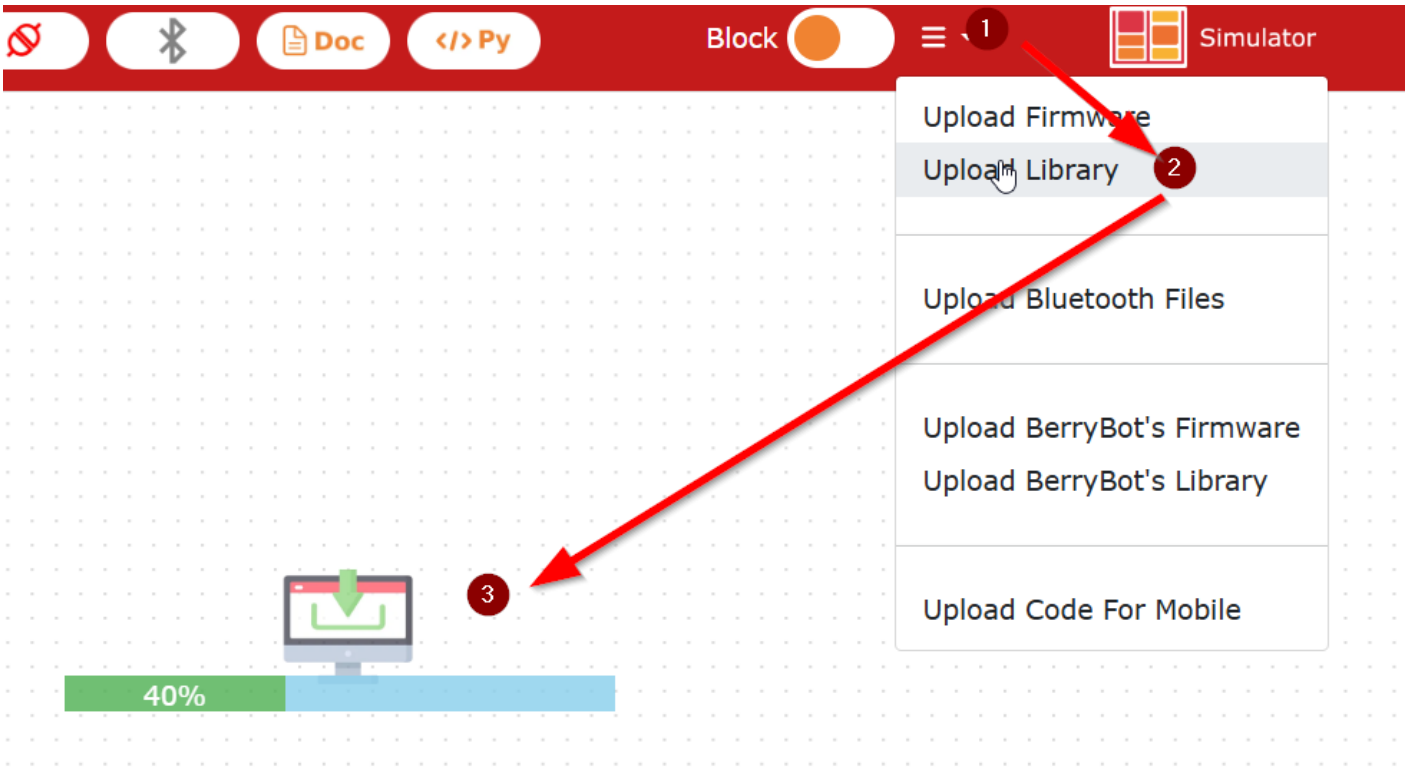


Fuente Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook ver créditos

Una vez grabado el firmware, esperamos a que nos salga un mensaje: *Please conect to the board*

Entonces dar a conectar y seleccionar la placa

Una vez conectado, descargamos las librerías en el PicoBricks para poder usar todas las funciones



YA ESTA, esto lo tienes que hacer **SOLO UNA VEZ** mientras uses PicoBlockly, si te pasas a otro programa y te cargas su firmware, tendrás que volverlo a poner.

Dos formas de ejecutar los programas

Picrobriks permite dos formas de trabajar:

https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWlGvIyVJr83Yk56kAoYD7bXLnYDm_tCQkeAgg/pubembed?start=falseloop=false&delayms=3000

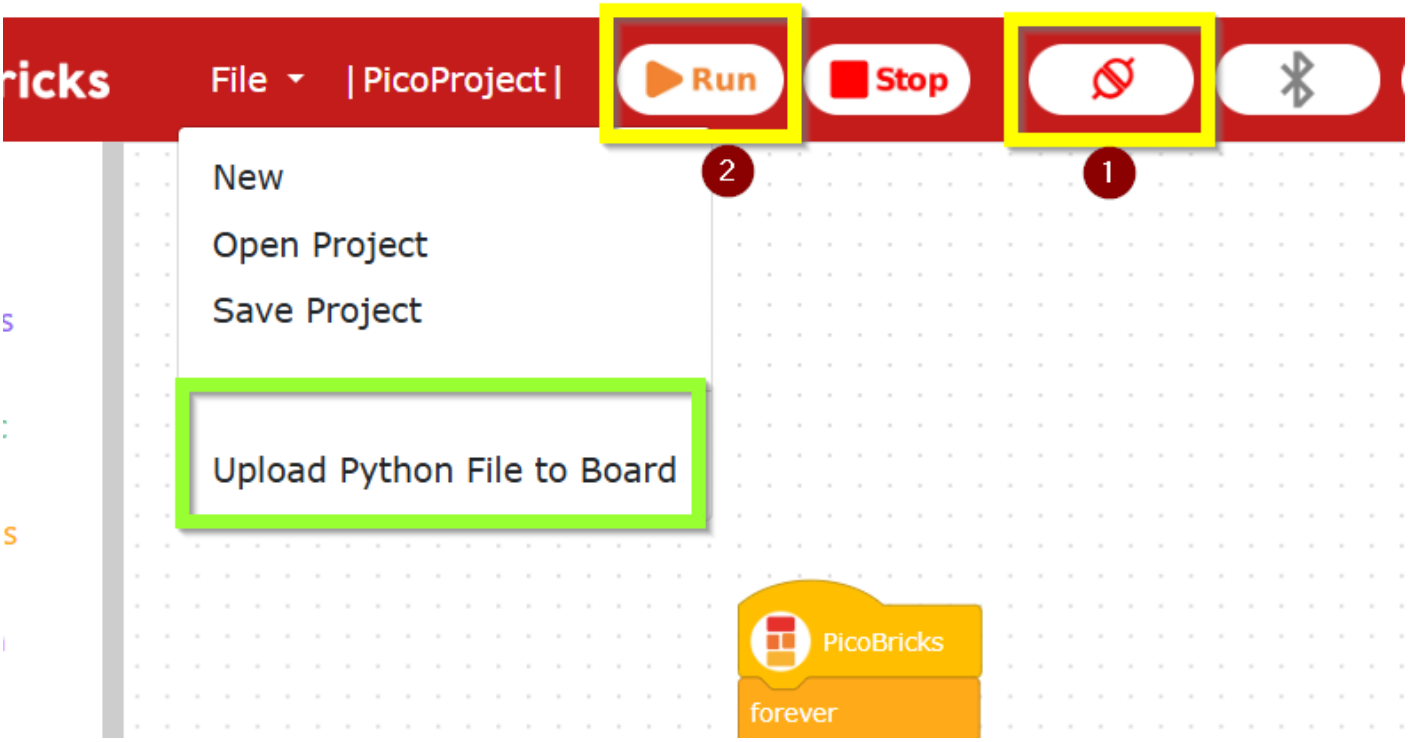
La forma más fácil de trabajar es **EN VIVO** es decir, que los programas se ejecuten **desde nuestro ordenador** es la más rápida y para ello necesita que el PicoBricks tenga el Firmware correspondiente dentro (tal y como hemos visto)

La otra forma de trabajar es **EN CARGA** es decir que los programas se ejecutan **desde dentro de PicoBricks** tiene la ventaja que el programa funciona si necesidad de ordenador. Eso sí, hay que alimentar Picobriks por el cable USB (usando un Powerbank o un cargador de móvil por ejemplo)

> **ATENCIÓN** si trabajamos EN CARGA nos "cargamos" el Firmware, por lo que si queremos volver a trabajar EN VIVO tenemos que volverlo a poner tal y como hemos visto

> Recomendamos EN VIVO por la rapidez y sencillez. Sólo es aconsejable EN CARGA cuando sean proyectos que precisen que el ordenador no esté. Por ejemplo en el PICO COCHE https://libros.catedu.es/books/pico-bricks/page/pico-coche

- Para trabajar **EN VIVO** tenemos que estar **conectados** (1) y darle al **Run** (2) (recuadros amarillos)
- Para trabajar **EN CARGA** entramos en archivo y cargamos el programa dentro de Picobricks (recuadro verde) **Upload Python File to Board**

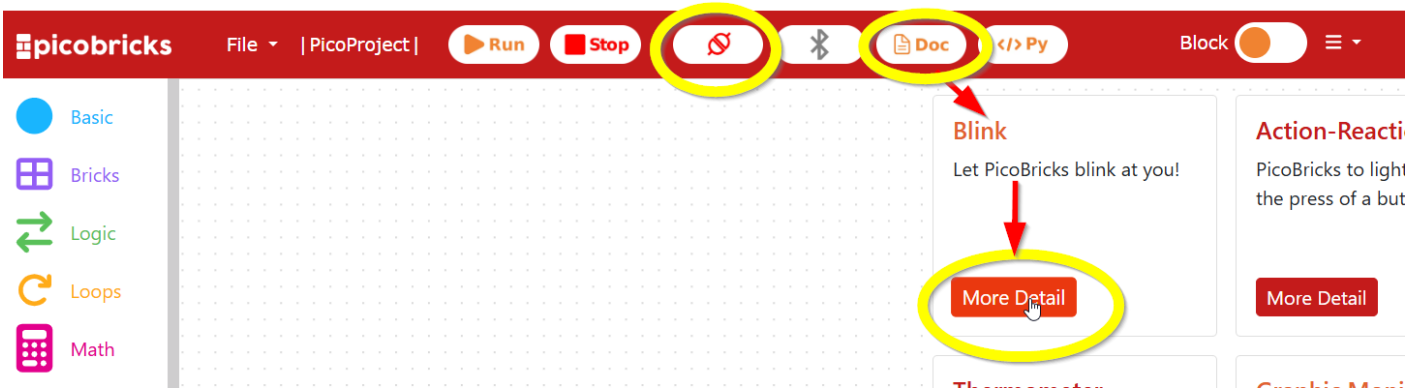


PROYECTO BLINK

> Extraido de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos

Vamos a realizar nuestro primer proyecto, parpadear el led rojo

Como es un programa predeterminado, lo más cómodo es ir los tutoriales que lo explican bien



Vamos al código y si apretamos en este botón, nos aparece en nuestro panel **si necesidad de hacerlo** pero ojo que a veces está escondido tras la ventana, usar el zoom y navegar

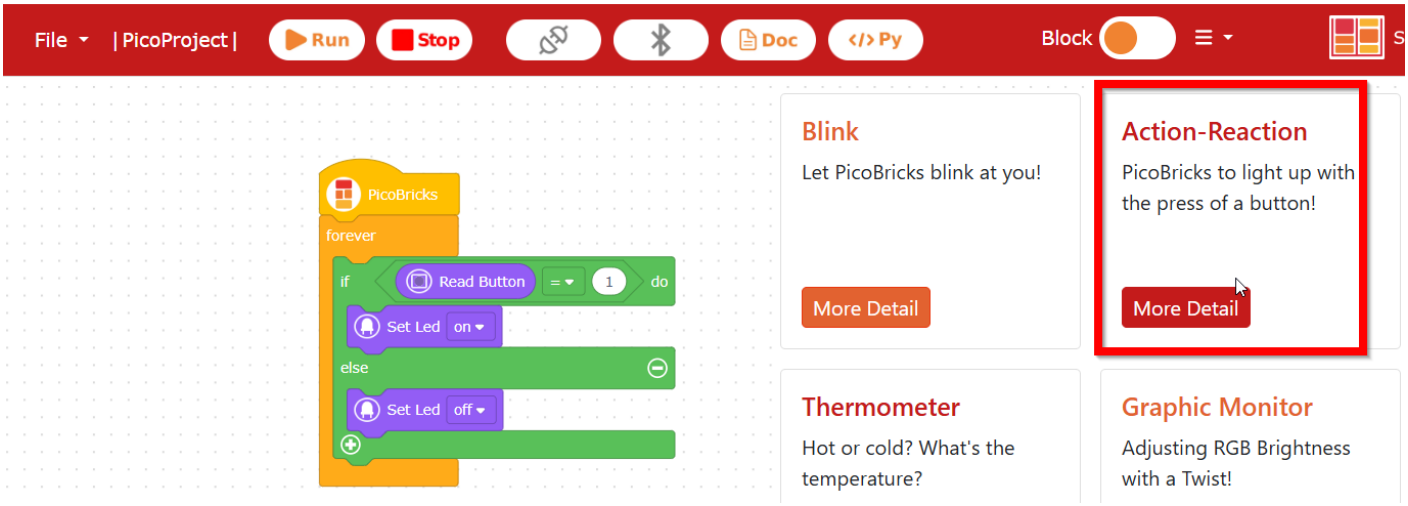
al dar a **RUN** tenemos

https://www.youtube.com/embed/nYPWAC_SHWA

PROYECTO ACTION-REACTION

> Extraido de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos

Ahora será con la interacción del botón. Repetimos los pasos pero con este proyecto:



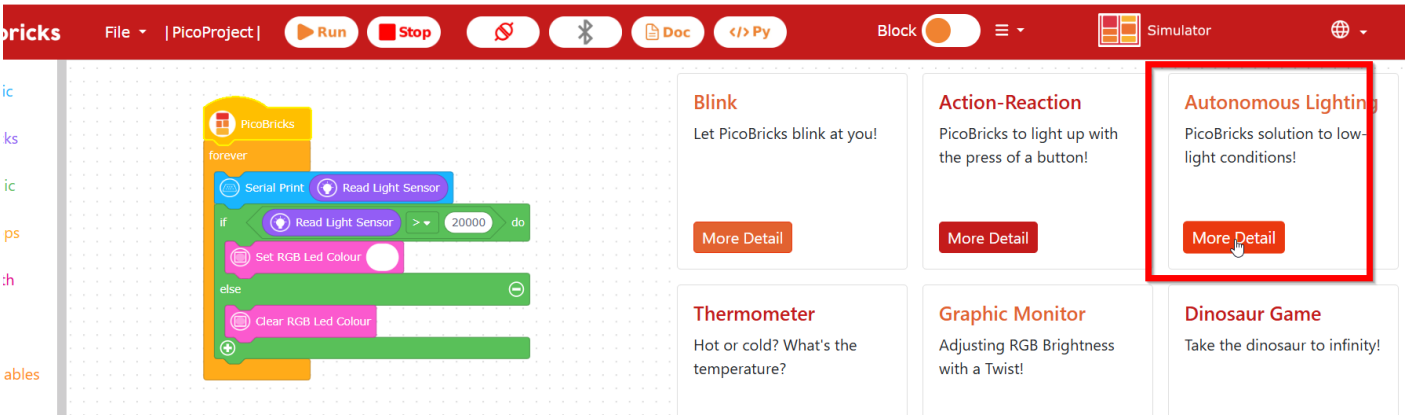
Resultado

https://www.youtube.com/embed/6cZ-Hk3dnJ8

PROYECTO Autonomous Lighting

> Extraido de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos

Lo mismo con el siguiente proyecto

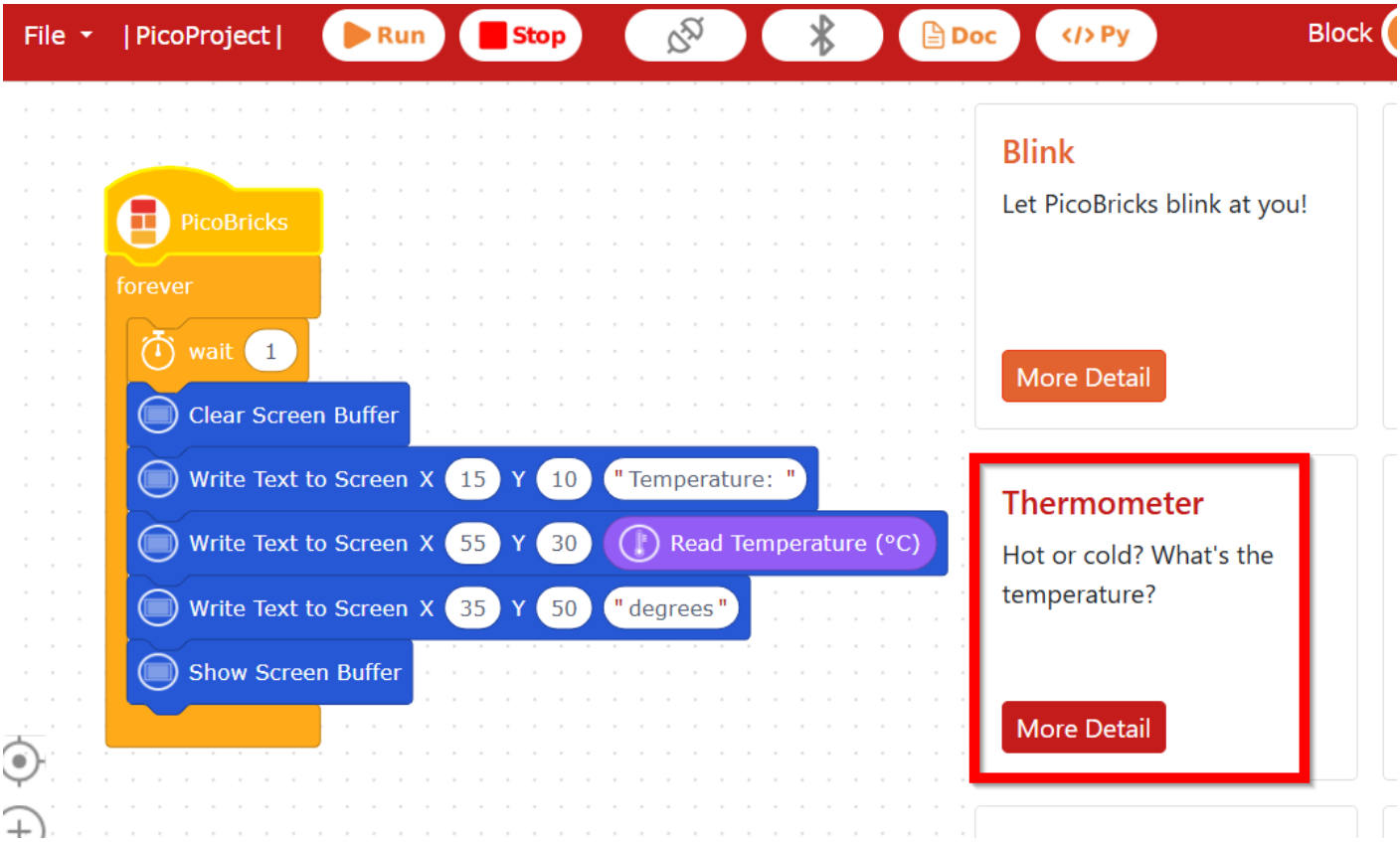


Resultado

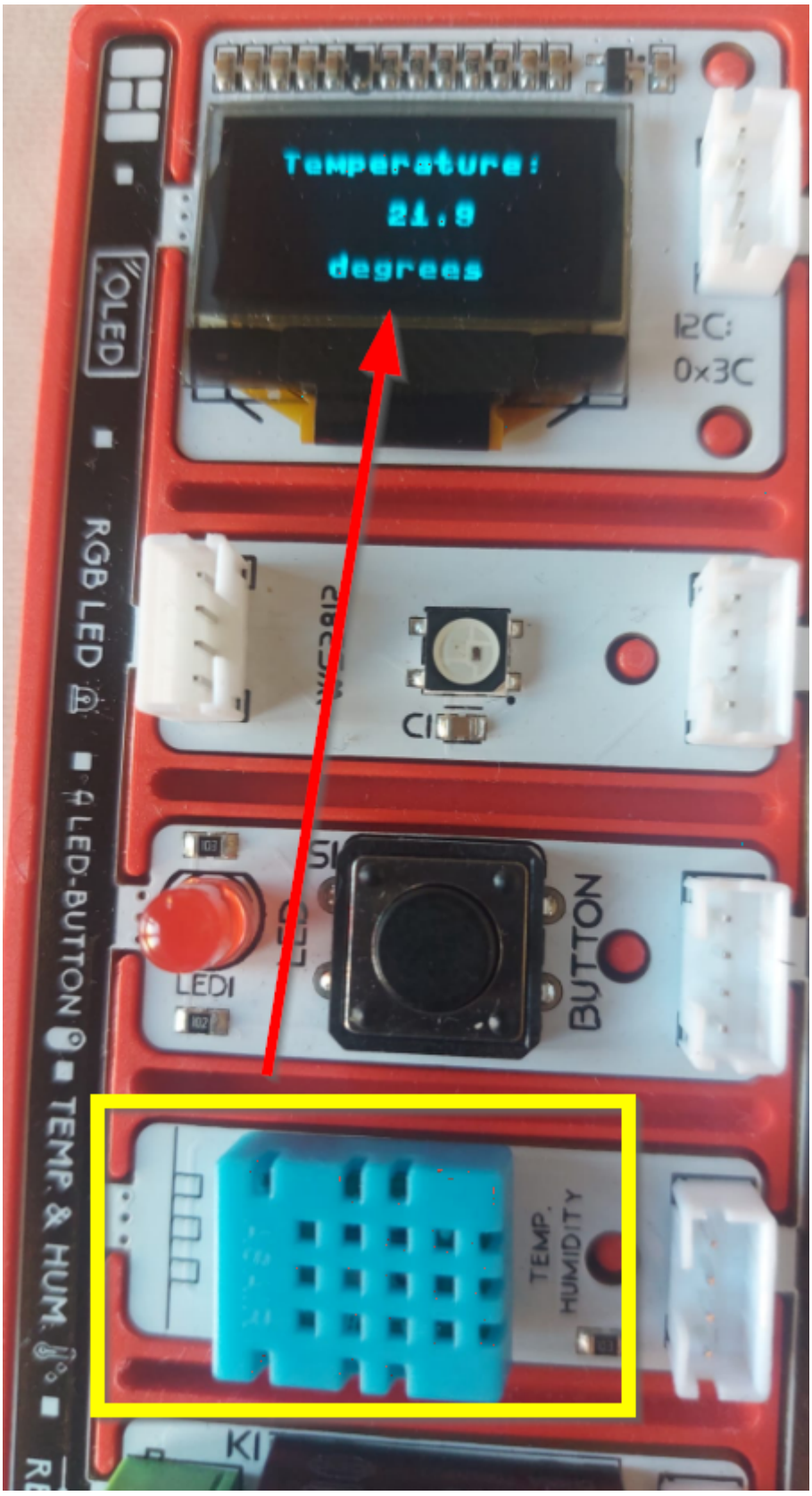
https://www.youtube.com/embed/sN8Y3boBPAg

PROYECTO Thermometer

> Extraído de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos



Si soplamos el aliento sobre el sensor podemos ver como sube la temperatura


Temperature:
21.9
degrees
OLED
I2C:
0x3C
RGB LED
LED-BUTTON
TEMP & HUM
LED1
BUTTON
TEMP.
HUMIDITY

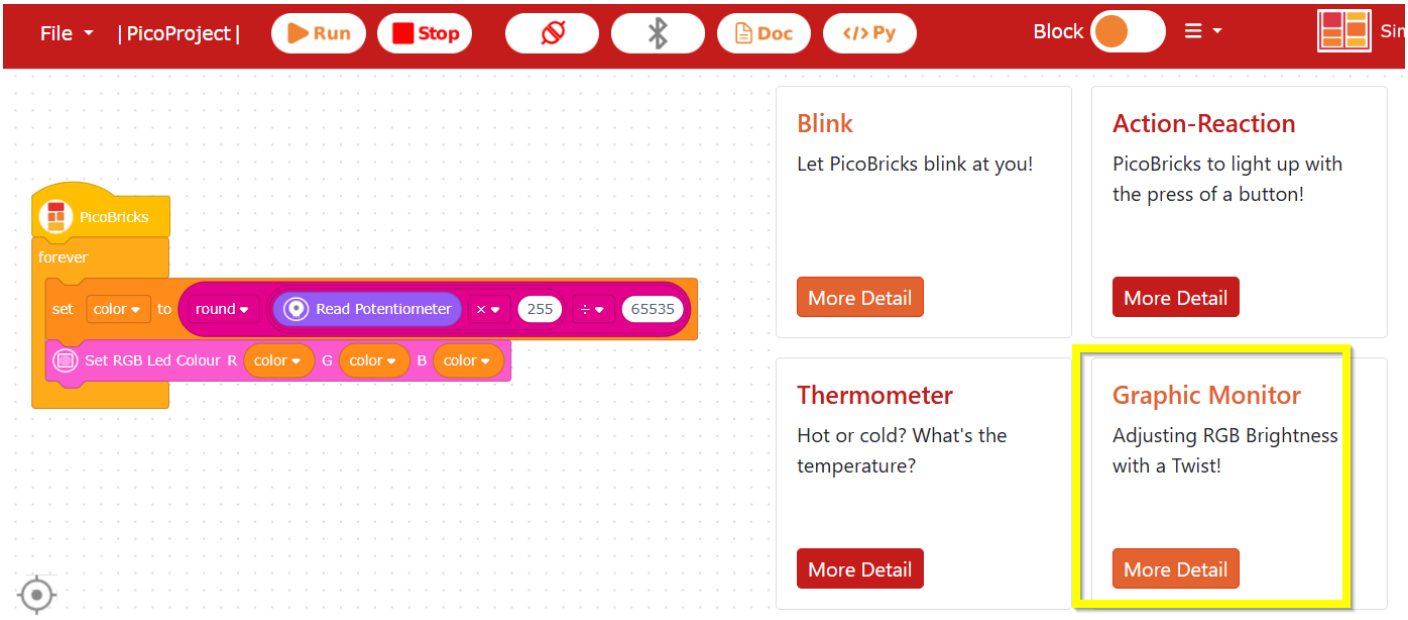
Recomendamos este proyecto cargarlo en el PicoBricks y así funciona autónomo sin necesidad de PC, con lo que se puede colocar en el exterior y ver la temperatura que hace simplemente alimentandolo con un PowerBank en el cable USB

P: ¿No sabes cómo se carga el programa en PicoBricks?

R: Porque no te has leído https://libros.catedu.es/books/pico-bricks/page/dos-formas-de-ejecutar-los-programas

PROYECTO Graphic Monitor

> Extraido de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos



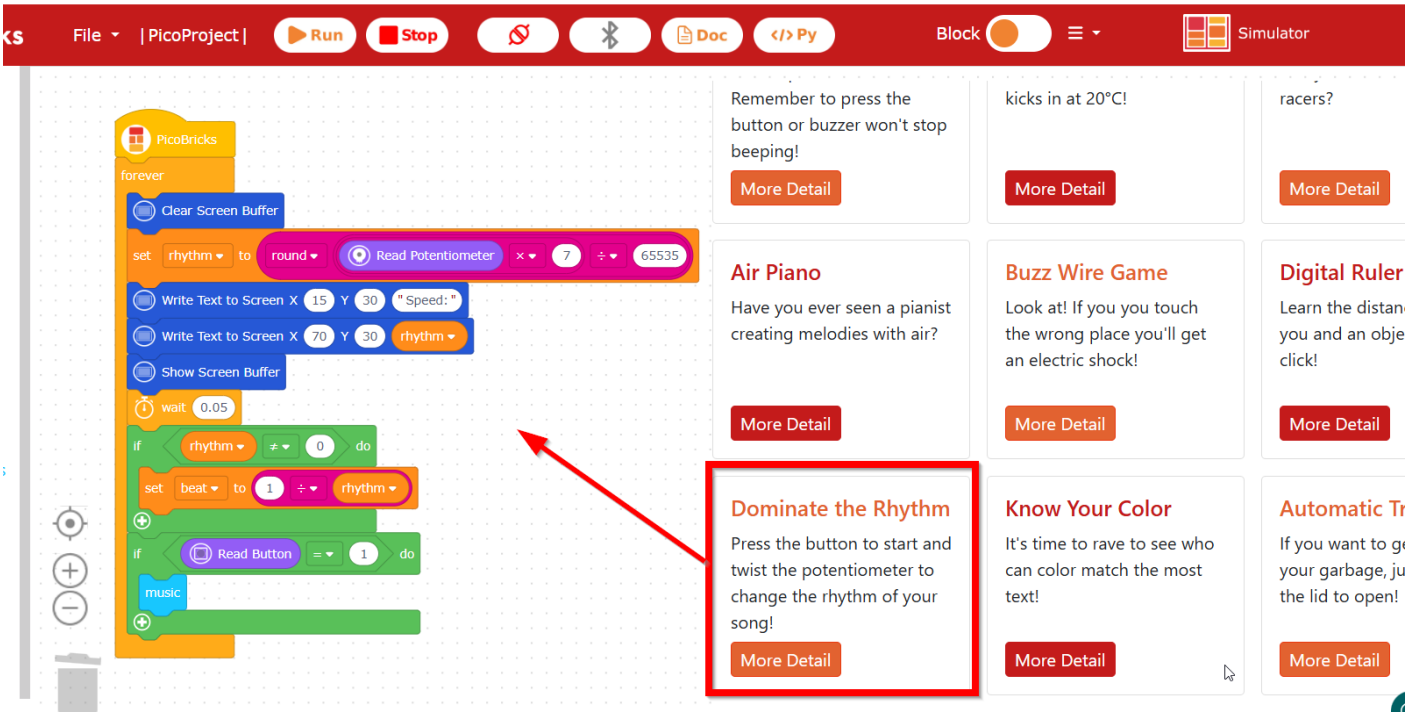
Resultado

https://www.youtube.com/embed/NqovcB6RImA

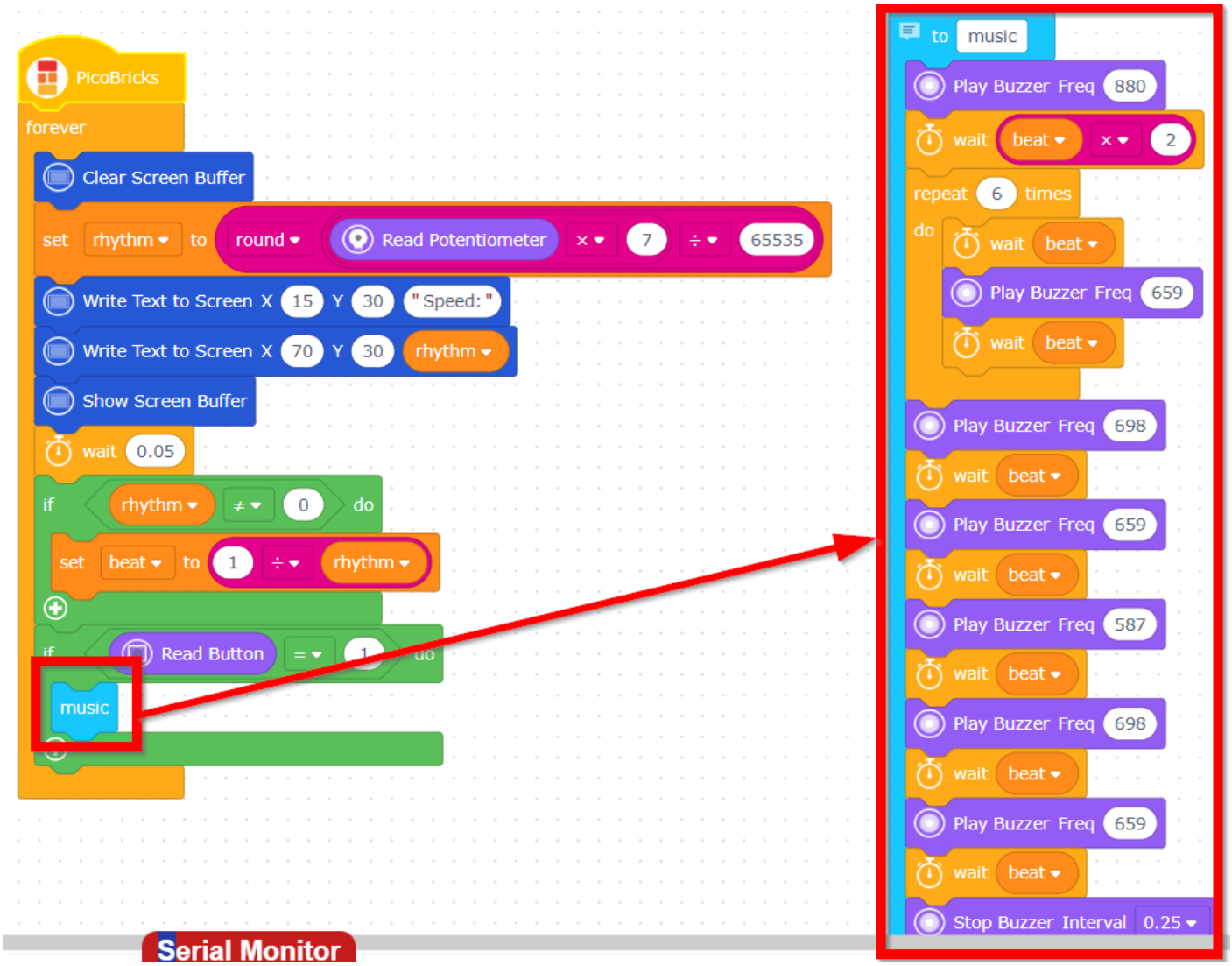
PROYECTO Dominate the Rhythm

> Extraído de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos

Este proyecto ya es más complejo y recomendamos importarlo desde el tutorial como siempre pues es más largo



Implica la utilización de FUNCIONES



Y recomendamos leer el tutorial, esta bien explicado en el libro en la página 34;

https://drive.google.com/file/d/1plad6bjn87FcgHb3cpd1vI-B_A25rnfF/preview

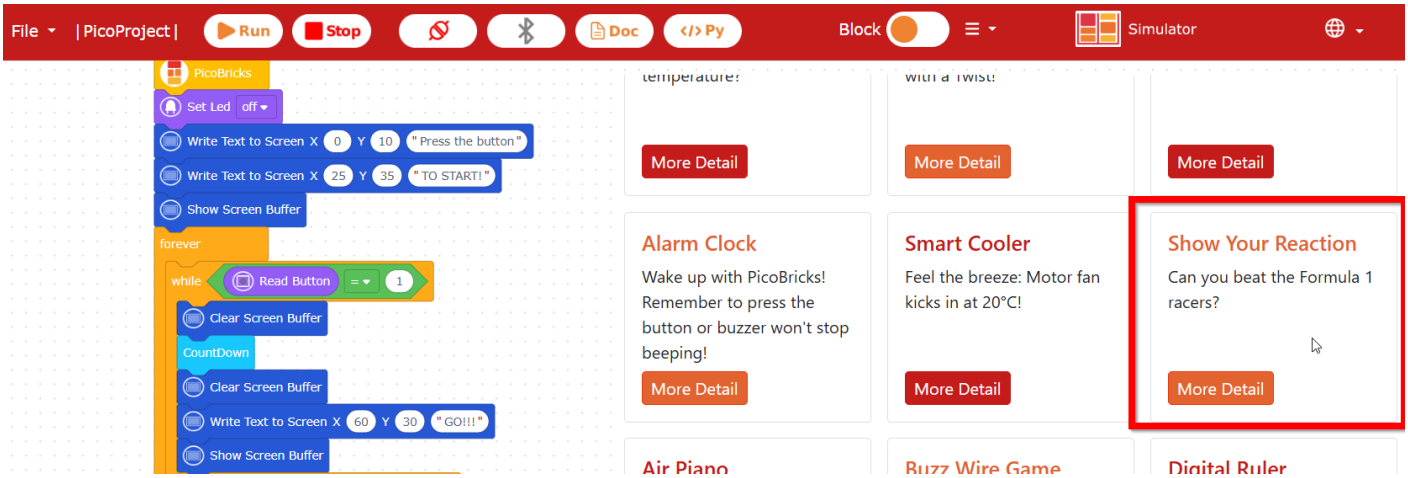
Como se puede ver en el resultado, la primera vez suena la música a un ritmo número 4 pero en la segunda vez subimos con el potenciómetro al ritmo máximo 7 y la música suena más deprisa

https://www.youtube.com/embed/WynkqehvWuw

PROYECTO Show Your Reaction

> Extraido de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos

Cuando nuestro proyecto comience a funcionar, mostraremos un mensaje de bienvenida en el OLED pantalla. A continuación imprimiremos en la pantalla lo que el usuario tiene que hacer para iniciar el juego. Para comenzar el juego, le pediremos al jugador que se prepare contando hacia atrás desde 3 en la pantalla después de presionar el botón. Después del final de la cuenta regresiva, el El LED rojo se encenderá en un tiempo aleatorio entre 2 y 10 segundos. Reiniciaremos el temporizador inmediatamente después se enciende el LED rojo. Mediremos el temporizador tan pronto como el se vuelve a pulsar el botón. Este valor que obtengamos estará en milisegundos. Mostraremos esto en la pantalla como el tiempo de reacción del jugador.



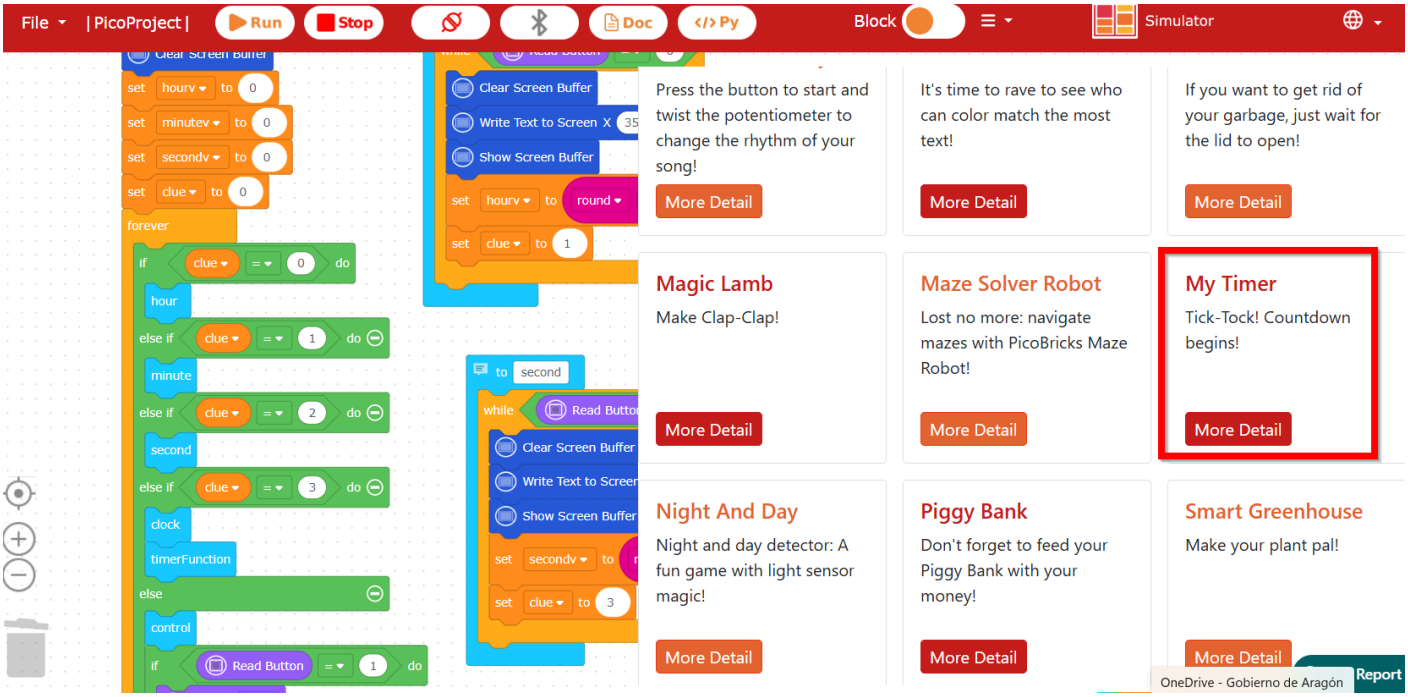
Aquí he ganado pues sólo he tardado 1ms en pulsar el botón

https://www.youtube.com/embed/CLEUZPzl2vI

PROYECTO My Timer

> Extraído de *Pico Bricks IDE Book CC-BY-SA* *https://picobricks.com/pages/idebook* ver créditos

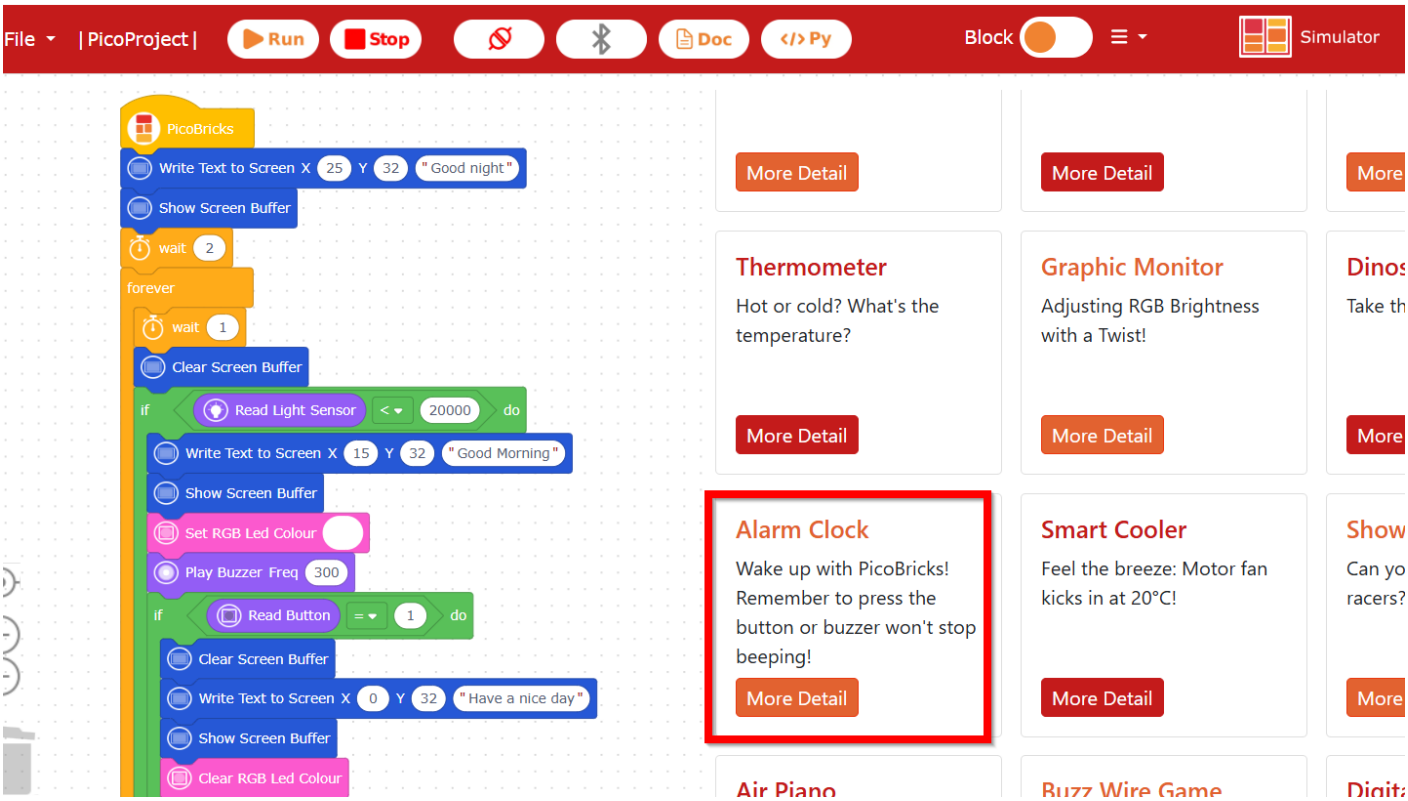
El clásico cuenta atrás pero con la peculiaridad que es fácil de programar con el potenciómetro, hasta las horas !



https://www.youtube.com/embed/QA7Oe8KibCo

PROYECTO Alarm Clock

> Extraído de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos



https://www.youtube.com/embed/8Drcl_YEsFs

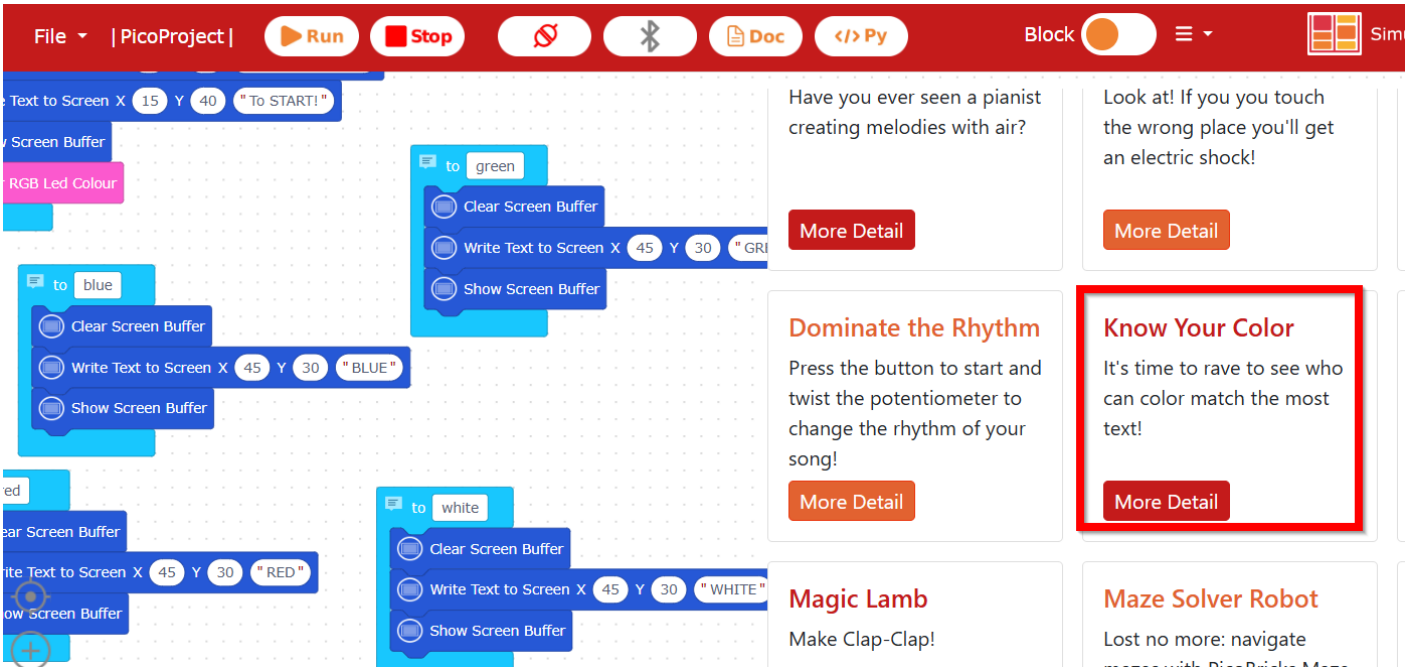
PROYECTO Know Your Color

> Extraido de *Pico Bricks IDE Book CC-BY-SA https://picobricks.com/pages/idebook* ver créditos

El juego que construiremos en el proyecto se construirá en base a que el usuario conozca los colores correcta o incorrectamente. Uno de los colores rojo, verde, azul y blanco se iluminará aleatoriamente en el LED RGB de Picobricks, y el nombre de uno de estos cuatro colores se escribirá aleatoriamente en la pantalla OLED al mismo tiempo. El usuario debe pulsar el botón de Picobricks en 1,5 segundos para utilizar el derecho de réplica.

- El juego se repetirá 10 veces, cada repetición obtendrá
  - 10 puntos si el usuario presiona el botón cuando los colores coinciden
  - -10 puntos si no coinciden

Después de diez repeticiones, la puntuación del usuario se mostrará en el OLED pantalla. Si el usuario lo desea, no podrá hacer uso de su derecho de réplica no pulsando el botón botón.



https://www.youtube.com/embed/Tv9z8krs26g

PROYECTO BUZZ WIRE GAME

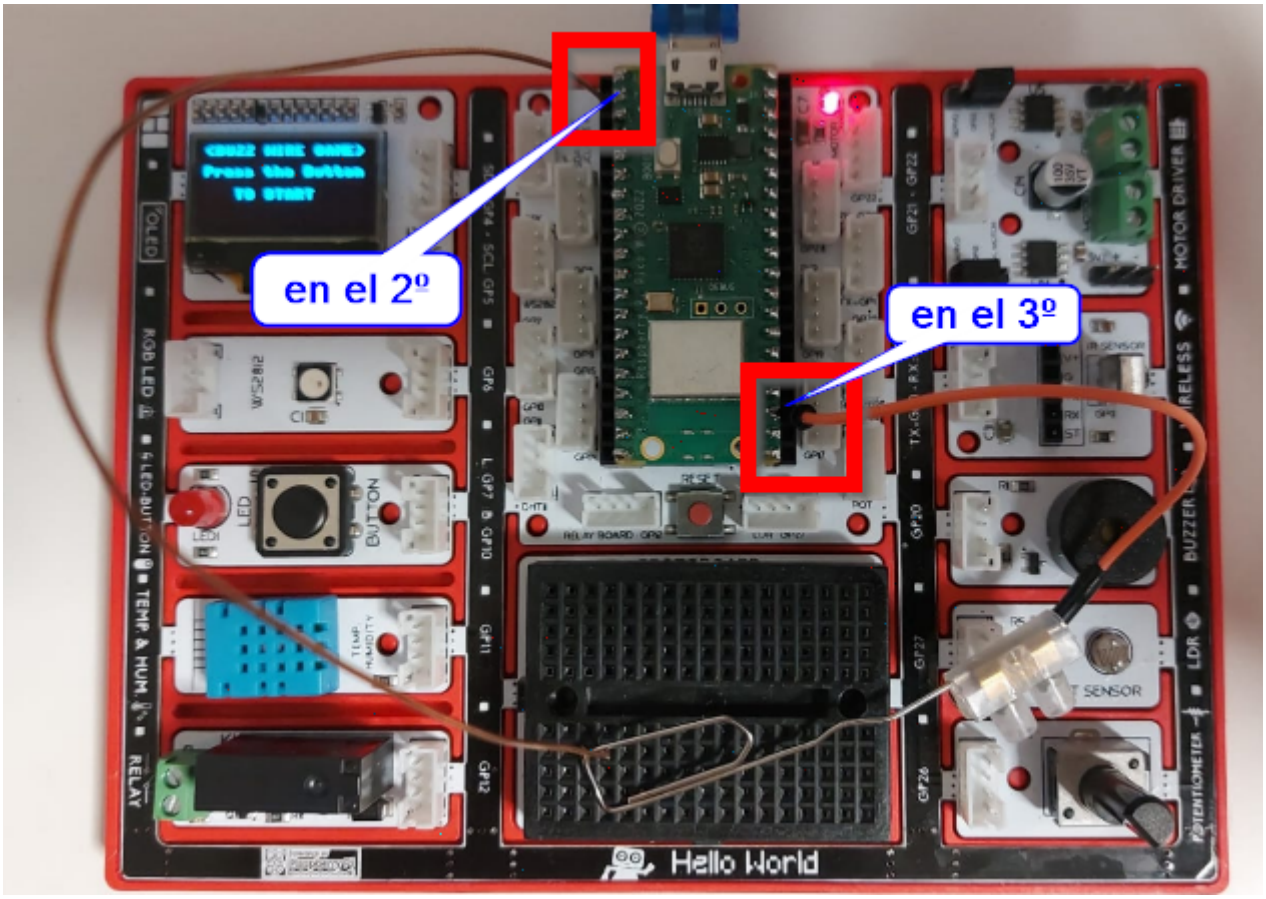
Enunciado

> Este juego es el típico de pasar una arandela por un cable y si toca, suena la alarma

Hardware

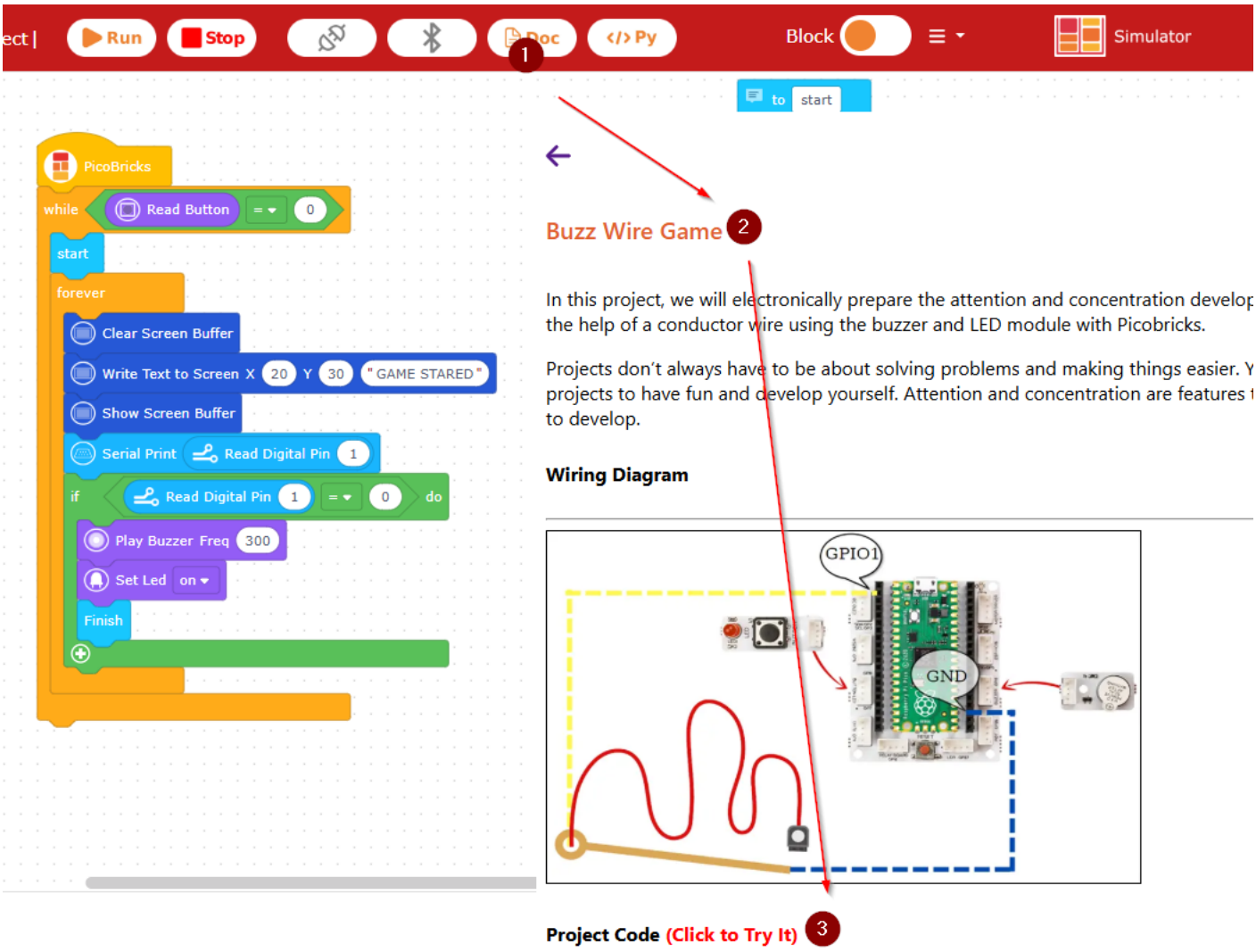
Preparamos :

- un cable pelado, rígido pero fino, conectado en el GP1 (segundo agujero por arriba izquierda) y GND (tercer agujero abajo derecha)
- un cable normal dupond
- un clip unido al cable normal dupond por una regleta pequeña como en la foto.



Software

El programa lo tienes en los tutoriales Buzz Wire Game



Resultado

https://www.youtube.com/embed/CzDTe3UoNcA

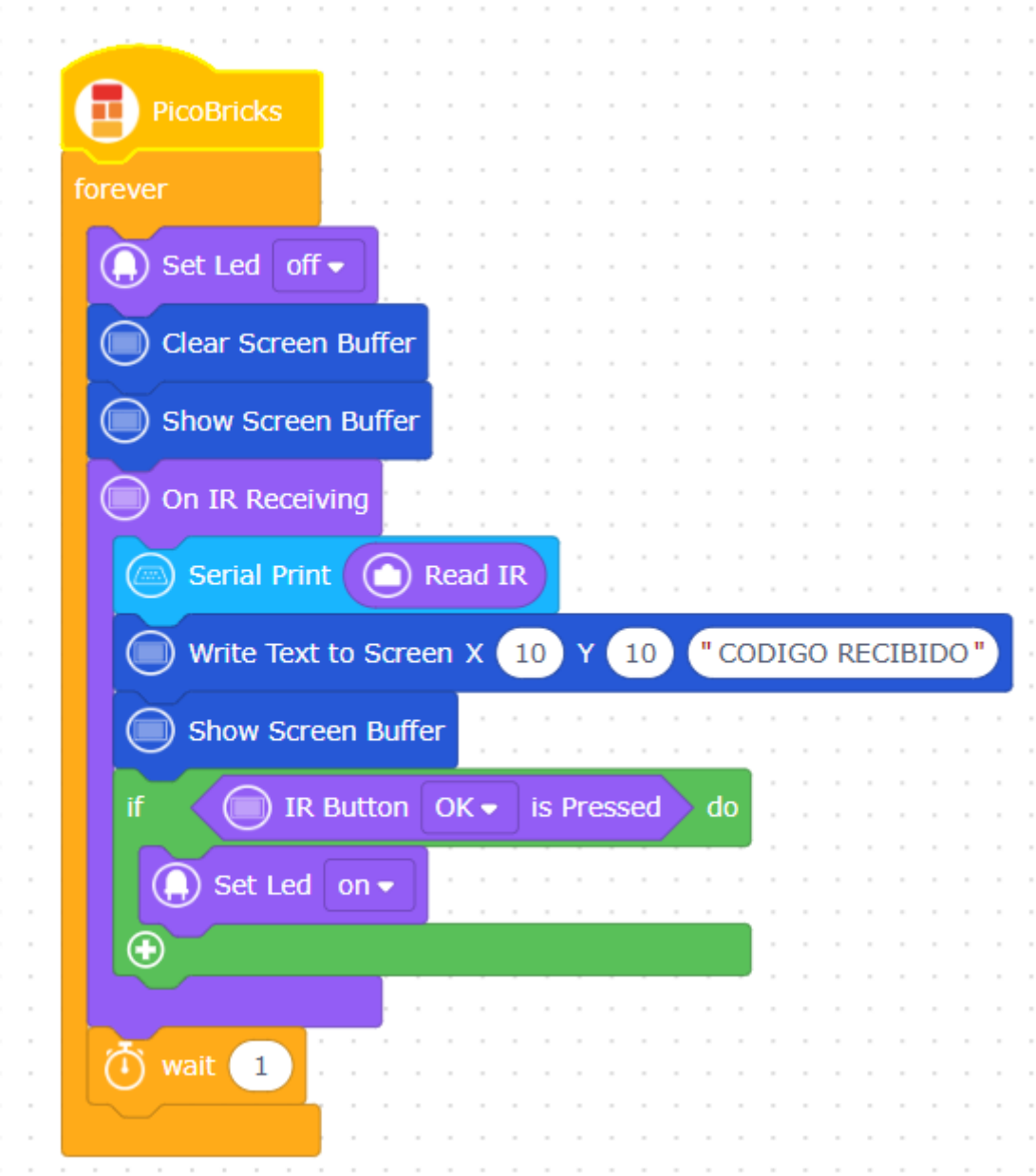
Este proyecto no funciona muy bien con los otros lenguajes de programación, no sé por qué

Algo diferente PROYECTO IR

No hay en los tutoriales ningún proyecto para usar el mando IR, luego este proyecto no pertenece a ninguno de los tutoriales que predetermina PicoBricks. Proponemos el siguiente enunciado

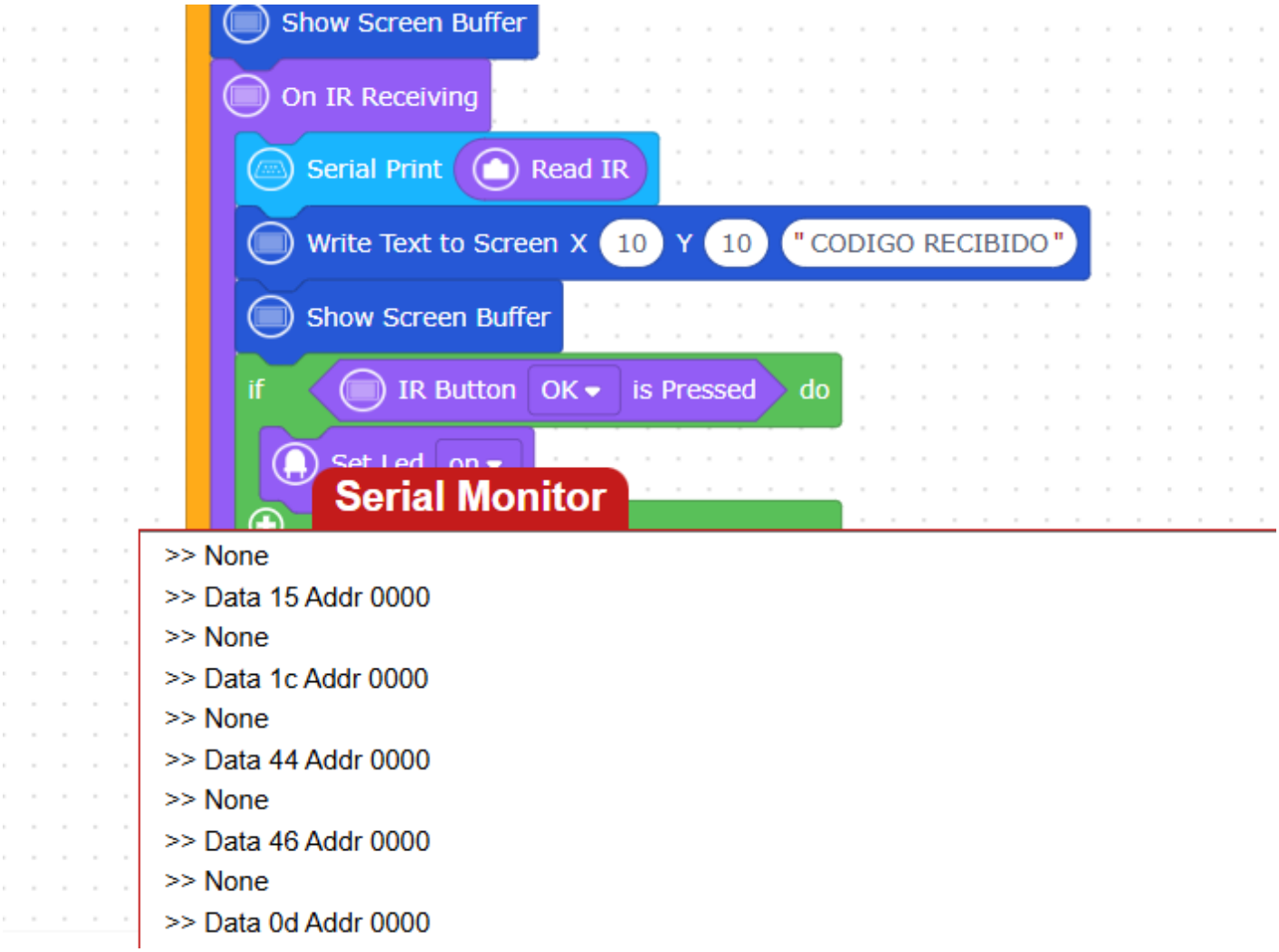
> Realizar un programa que:
> -si se aprieta un botón del mando IR que visualice por la pantalla OLED que ha recibido un código
> - visualizará por el puerto serie el código recibido
> - si la tecla es OK se encenderá el led rojo

Solución



Resultado

Por el puerto serie van apareciendo los códigos de las teclas apretadas en el mando IR



y en la pantalla OLED se visualizaba que se había recibido un código y si era OK se enciende el led rojo:

https://www.youtube.com/embed/6kSRjpbTSDg

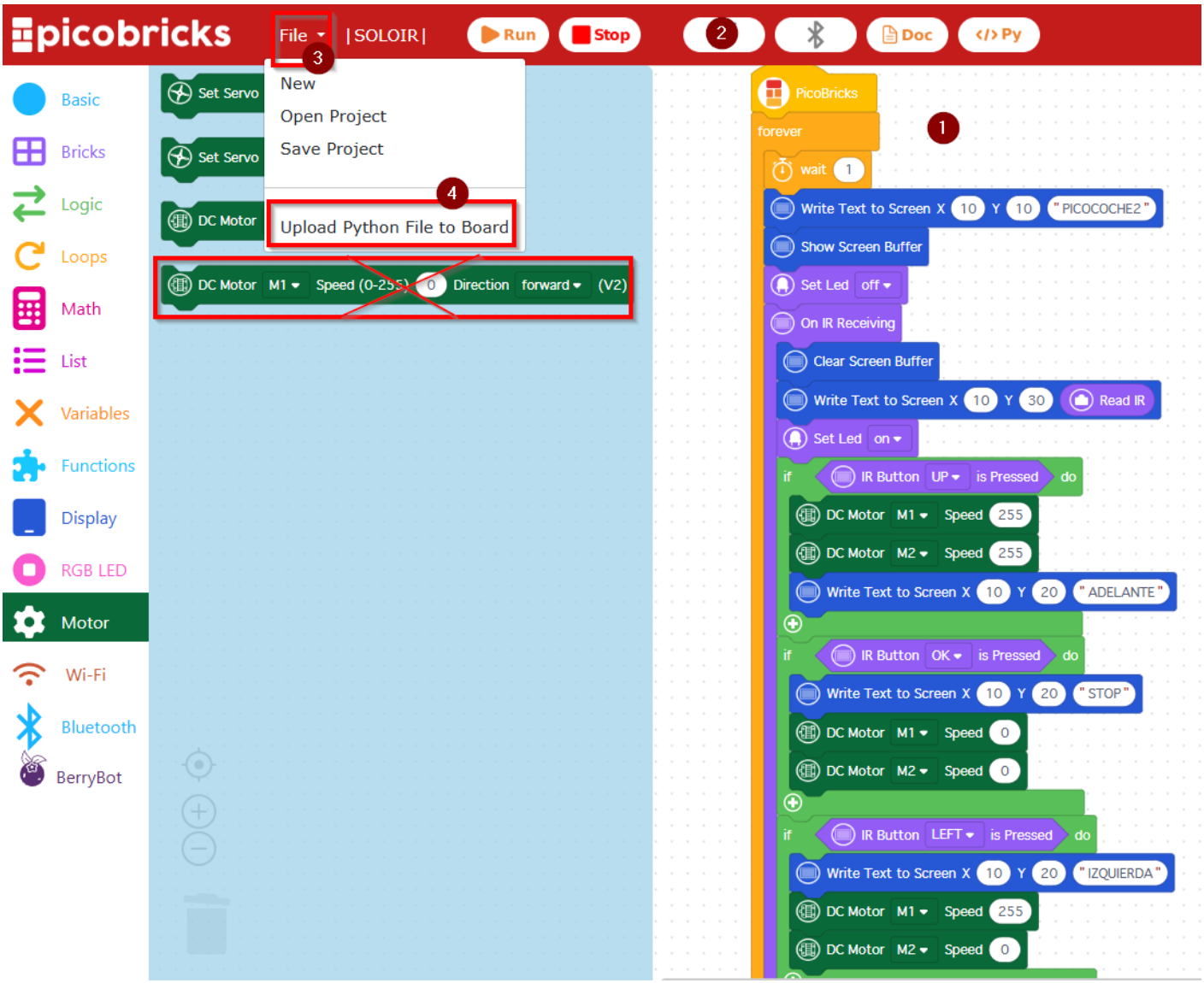
PICO COCHE

Podemos comprar en cualquier tienda de electrónica por unos 10€ un kit de coche, por ejemplo aquí

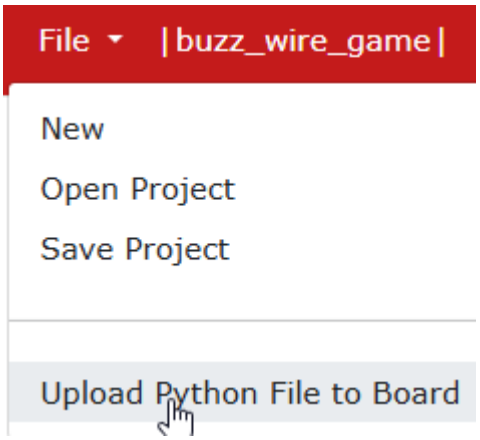
En Picobrkicks creamos un programa similar al anterior de Infrarrojos

> Precaución: No utilizar la instrucción señalada, bloquea nuestra Raspberry Pi Pico W, sospechamos que es para otra versión. Por lo tanto no puede dar marcha atrás

> Consejo: Grabarlo en la placa, y así es independiente

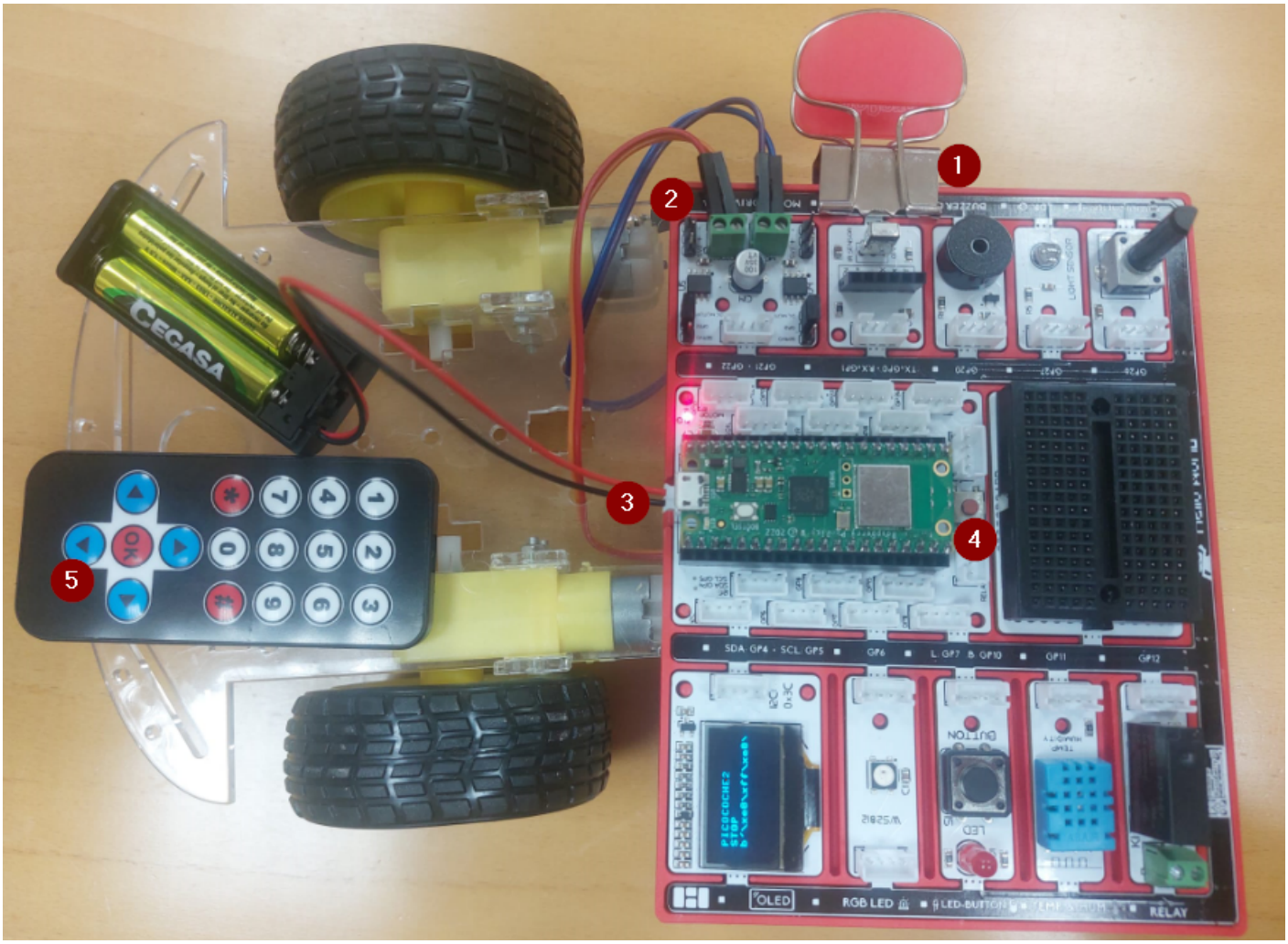


Una vez cargado el programa en la placa Raspberry Pi pico W,

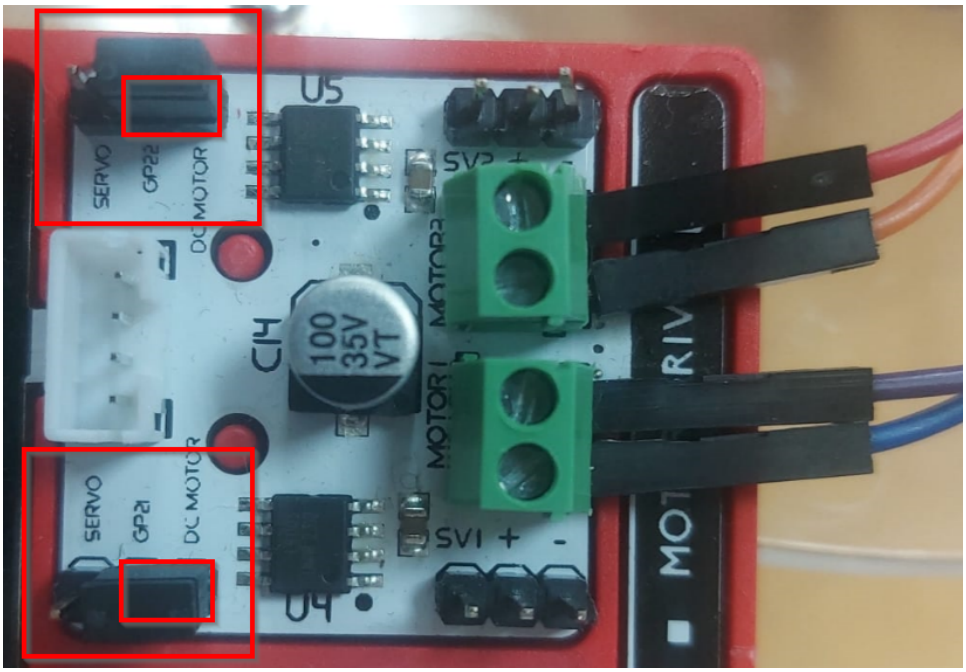


lo montamos en el coche:

1. Ponemos alguna sujeción, en este caso he puesto una pinza, pero puede ser una goma elástica.
2. Conectamos los cables de los motores en los terminales verdes
    1. Si por alguna razón va al revés algún motor, intercambiar los cables
3. Conectamos alimentación
    1. Puede ser también con el cable usb y un powerbank
4. Apretamos el botón reset
5. A jugar



> Detalle de conexión, asegúrate que los jumpers están colocados en modo DCMOTOR, es decir GP21 y GP22 tienen que estar conectados con DCMOTOR1 y DCMOTOR2



las pilas estaban algo gastadas ...

https://www.youtube.com/embed/G1snag4AK9M

Mapeo

¿Qué es eso de "mapeo"?

En la jerga robótica, dicho pronto y mal pero para que se entienda, mapear significa hacer un **cambio de escala**

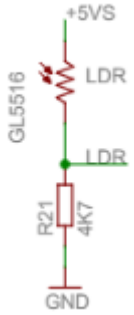
¿Cuándo se dan esas situaciones?

> **SITUACION A :** Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida digital PWM de Arduino que va de 0-255

> **SITUACION B :** Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 e interpretarlo en sus valores de voltios. Si suponemos que la placa se alimenta a 5V la variable de salida irá desde 0 a 5V

> **SITUACION C :** Queremos leer el valor de un LDR, que tapándolo nos da 917 e iluminándolo al máximo es 1023, lo queremos copiar en una salida digital PWM, o sea que la salida va desde 0 a 255

Nota: El mínimo de 917 (puede ser otro número, es un valor experimental) es debido a que los LDR van montados en un divisor de tensión como el de la figura, y la resistencia de abajo, siempre se queda algo de tensión



> **SITUACION D :** Queremos según el valor de un joystick conectado a las entradas analógicas de un Arduino (esto pasa en Echidna) se representen en la pantalla de Scratch

2*220 por 2*180, es decir
- Eje X : el potenciómetro (vamos a llamarlo *potx*) va de 0 a 1023 y la salida (*ejex*) va de -220 a 220
- Eje Y : el potenciómetro (vamos a llamarlo *poty*) va de 0 a 1023 y la salida (*ejey*) va de -180 a 180

SITUACION E: Ídem pero no con el potenciómetro, sino con el acelerómetro (vamos a llamarlo *acel*) que va 250 a 500

SITUACION F : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida de un servo, por lo tanto lo que necesita es un ángulo que va de 0-180

SITUACION G : Idem que F pero una raspberry por lo tanto GPI va de 0-65.535

¿Cómo se consigue mapear?

- Si programas con código ArduinoIDE, tienes la instrucción **map**
- Si no tienes map, por ejemplo, programas con bloques gráficos tipo Scratch, lo tienes que hacer a mano
  - ¿Cómo? Con la ecuación de una recta

Para entendernos :

- **X** será el valor de entrada que tiene unos valores límites $\mathbf{X_1}$ **e** $\mathbf{X_2}$
- **Y** es la variable de salida que queremos y que tiene otros valores límites $\mathbf{Y_1}$ **e** $\mathbf{Y_2}$

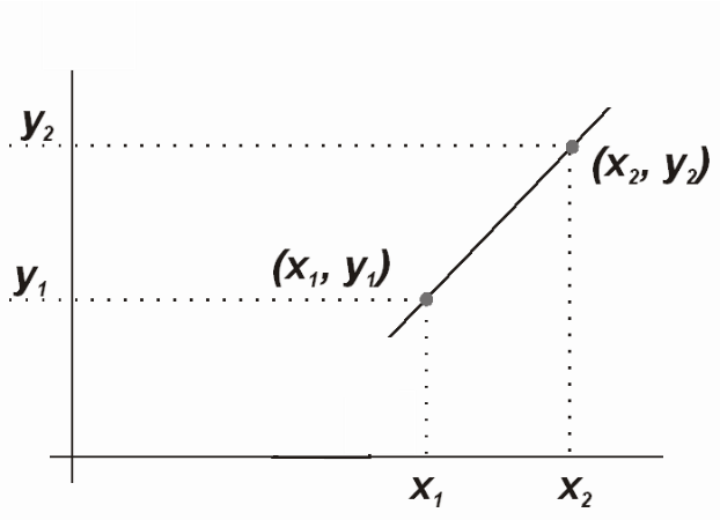
Luego y tiene esta ecuación :

$$y = y_1 + m * (x - x_1)$$

donde m es

$$m = \frac{x_2 - x_1}{y_2 - y_1}$$

Gráficamente



¿Una hoja de cálculo para poder hacer esos cálculos?

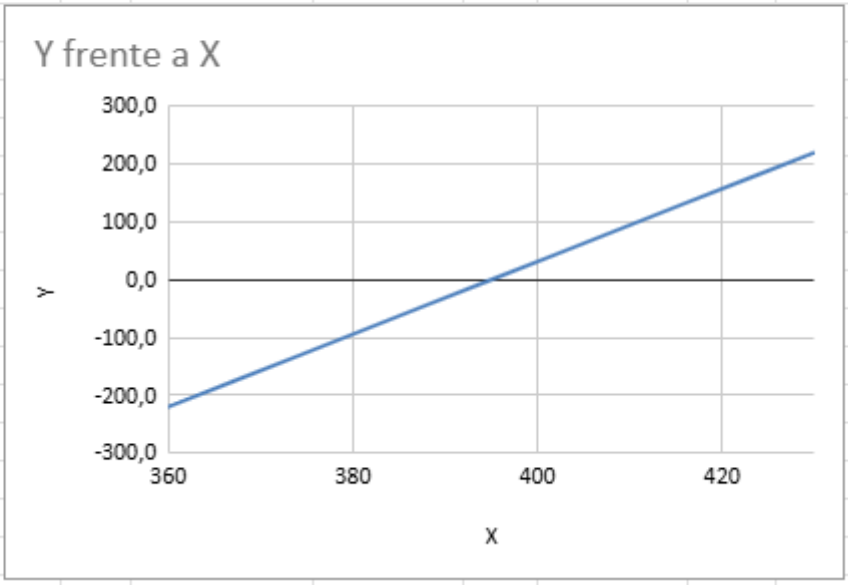
Sí, claro, en este enlace https://docs.google.com/spreadsheets/d/1qNbaZ2c_H1UCNhtvp2LimfWSbaGvZLVl5gJr9Wu0ifU/edit?usp=sharing dale a descargar

Rellena las casillas amarillas, y en las naranjas tienes el resultado de m y n

y = m * x + n

m	6,3	x1	360	y1	-220	
n	-2482,9	x2	430	y2	220	

X	Y
360	-220,0
367	-176,0
374	-132,0
381	-88,0
388	-44,0
395	0,0
402	44,0
409	88,0
416	132,0
423	176,0
430	220,0



¿Me lo puedes hacer para cada situación anterior?

Si claro:

> **SITUACION A :** Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida digital PWM de Arduino que va de 0-255

- Límites de las variables :
  - X de 0-1023
  - Y de 0-255
- Con la instrucción map : Y = map(X, 0, 1023, 0, 255);
- Sin la instrucción map Y = 025 * X pues 255/1023 = 0.25 también podemos escribir Y = X/4

> **SITUACION B :** Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 e interpretarlo en sus valores de voltios. Si suponemos que la placa se alimenta a 5V la variable de salida irá desde 0 a 5V

- Límites de las variables :
  - X de 0-1023
  - Y de 0-5
- Con la instrucción map : Y = map(X, 0, 1023, 0, 5);
- Sin la instrucción map Y = 0.0048 * X pues 5/1023 = 0.0048 o también podemos escribir Y = X/204 que queda mejor pues 1023/5=204 aprox.

> **SITUACION C :** Queremos leer el valor de un LDR, que tapándolo nos da 917 e iluminándolo al máximo es 1023, lo queremos copiar en una salida digital PWM, o sea que la salida va desde 0 a 255

- Límites de las variables :
  - X de 917-1023
  - Y de 0-255
- Con la instrucción map : Y = map(X, 917, 1023, 0, 255);
- Sin la instrucción map Y = 2.4 * X pues 255/(1023-917) = 2.4

> **SITUACION D :** Queremos según el valor de un joystick conectado a las entradas analógicas de un Arduino (esto pasa en Echidna) se representen en la pantalla de Scratch 2*220 por 2*180, es decir
> - Eje X : el potenciómetro (vamos a llamarlo *potx*) va de 0 a 1023 y la salida (*ejex*) va de -220 a 220

- Eje Y : el potenciómetro (vamos a llamarlo *poty*) va de 0 a 1023 y la salida (*ejey*) va de -180 a 180

- EJEX
  - Límites de las variables :
    - potx de 0-1023
    - ejex de -220 a +220
  - Con la instrucción map : ejex = map(potx, 0, 1023, -220, 220);
  - Sin la instrucción map ejex =-220 + 0.43*potx pues (220-(-220))/1023=0.43
- EJEY
  - Límites de las variables :
    - poty de 0-1023
    - ejey de -180 a +180
  - Con la instrucción map : ejey = map(poty, 0, 1023, -180, 180);
  - Sin la instrucción map ejey =-180 + 0.35*poty pues (180-(-180))/1023=0.35

SITUACION E: Ídem pero no con el potenciómetro, sino con el acelerómetro (vamos a llamarlo *acel*) que va 250 a 500

- EJEX
  - Límites de las variables :
    - acelerómetro acel de 250-500
    - ejex de -220 a +220
  - Con la instrucción map : ejex = map(acel, 250, 500, -220, 220);
  - Sin la instrucción map ejex =-220 + 1.76*(acel-250) pues (220-(-220))/(500-250)=1.76
- EJEY
  - Límites de las variables :
    - acelerómetro acel de 250-500
    - ejey de -180 a +180
  - Con la instrucción map : ejey = map(acel, 250 500, -180, 180);
  - Sin la instrucción map ejey =-180 + 1.44*(acel-250) pues (180-(-180))/(500-250)=1.44

SITUACION F : Queremos leer un valor de entrada analógica en un Arduino, por lo tanto va de 0-1023 y queremos que se copie en una salida de un servo, por lo tanto lo que necesita es un ángulo que va de 0-180

- Límites de las variables :

    - X de 0-1023
    - Y de 0-180
- Con la instrucción map : Y = map(X, 0, 1023, 0, 180);
- Sin la instrucción map Y = 0.17* X pues 180/1023 = 0.17 también podemos escribir Y = X/5.7 pues 1023/180=5.7

SITUACION G : Idem que F pero una raspberry por lo tanto GPI va de 0-65.535

- Límites de las variables :
    - X de 0-65535
    - Y de 0-180
- Con la instrucción map : Y = map(X, 0, 65535, 0, 180);
- Sin la instrucción map Y = 0.00274* X pues 180/65535 = 0.00274 pero es más cómodo al revés Y = X/364 pues 65535/180=364

Servo

Una de las aplicaciones más utilizadas de los sistemas de control por ordenador y en la robótica están asociados con los motores, que permiten accionar o mover otros componentes, como puertas, barreras, válvulas, ruedas, etc. Uno de los tipos que vamos a ver en este capítulo son lo servos, hay de dos tipos:

- El **servomotor** o **servos convencionales** que posee la capacidad de posicionar su eje en un ángulo determinado entre 0 y 180 grados en función de una determinada señal.
- **Servo de rotación continua** Son servos por fuera igual que los anteriores, pero pueden girar 360º y se controlan por tiempo

Por defecto cuando se dice **servo**, es un **servomotor o servo convencional**

Servomotores o servos convencionales

Los servos son un tipo especial de motor en el que se añade una circuito lógico electrónico que permite un control mucho más preciso que a un motor normal de corriente continua. Esto les permite posicionar el eje en un ángulo determinado.

El hardware interno se compone de un potenciómetro y un circuito integrado que controlan en todo momento los grados que gira el motor. De este modo, en nuestro caso, desde Arduino, usando las salidas digitales PWM podremos controlar fácilmente un servo. Lo ideal es conectarlo a 6V pero trabajan bien en los 5V del Arduino.

Hay muchos modelos, en robótica educativa cuestan entre 1-5€, el más común es el SG90, muy barato, pero tiene muy poca fuerza, el MG90S tiene algo más, si queremos algo más, ya tiene que ser el MG996R pero ya este modelo **NO se puede conectar directamente al Arduino o Raspberry,** el pico de energía que necesita, provoca el reinicio de la placa. Incluso varios pequeños SG90.

Si quieres saber más, te recomendamos https://www.luisllamas.es/controlar-un-servo-con-arduino/

Ejemplos de uso de servos:

- Curso Arduino con código
- Curso brazo robótico del mClon con nanoArduino
- Apertura de barrera por ultrasonidos en curso Arduino con ArduinoBlocks
- Tractor entrando en el corral Arduino con EchidnaShield
- Apertura de puerta Domótica con Arduino
- Servo con PicoBrick
- Apertura ventana y puerta en Smarth Home para microbit
- Smart Agriculture Kit para micro:bit

Enunciado

Vamos a aprovechar el concepto de **mapeo** con el siguiente enunciado

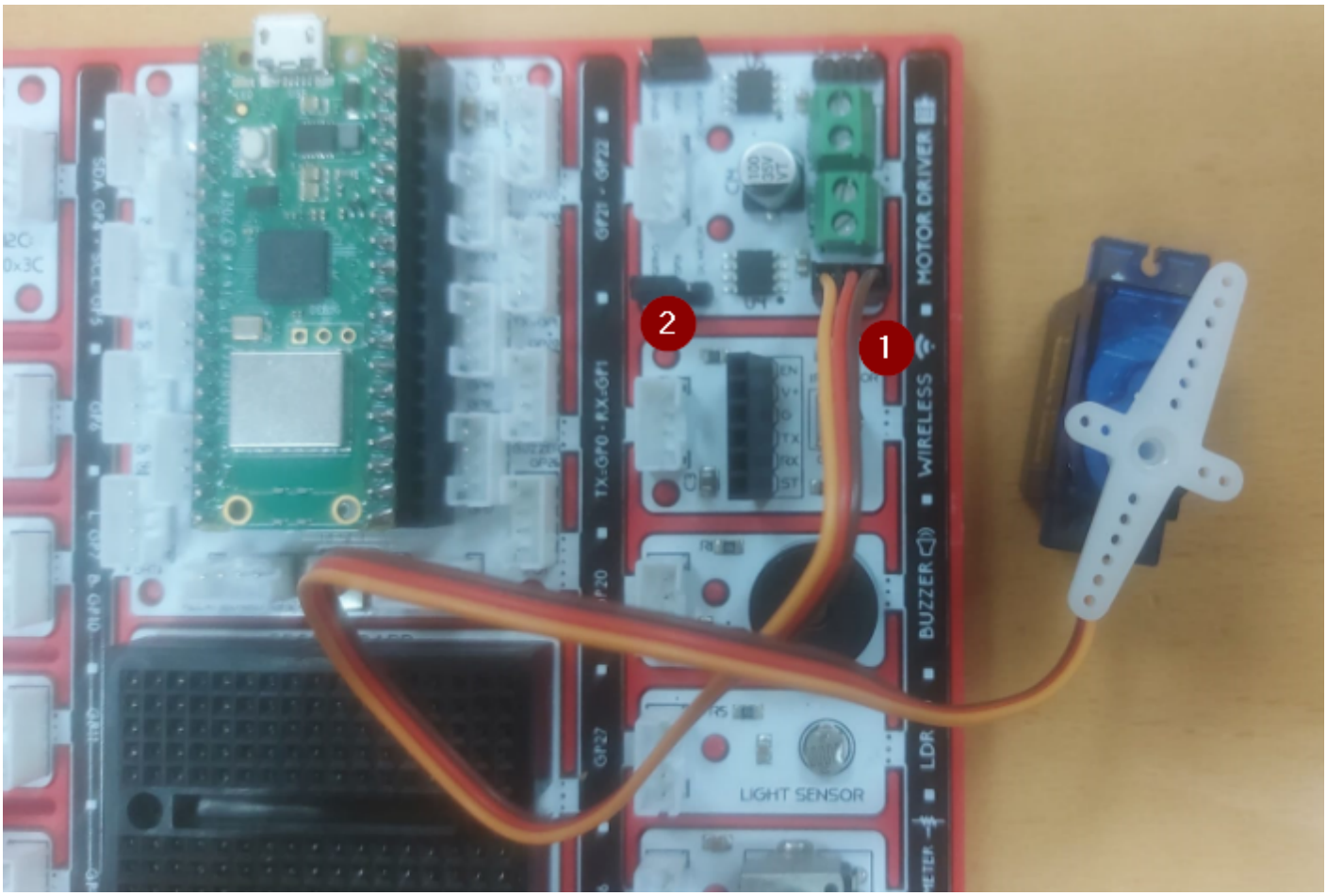
> Queremos que se mueva un servo motor (ángulo de 0º a 180º) en función del potenciómetro, conectado a GP27 por lo tanto sus valores van de 0 a 65535 (que es 2 elevado a 16)

Hardware

1. Conectamos un servo en el slot correspondiente

   > Ojo, fíjate que tienes que conectar el pin - con el marrón. Mira la foto

2. Movemos el Jumper a la posición que conecte GP21 con Servo

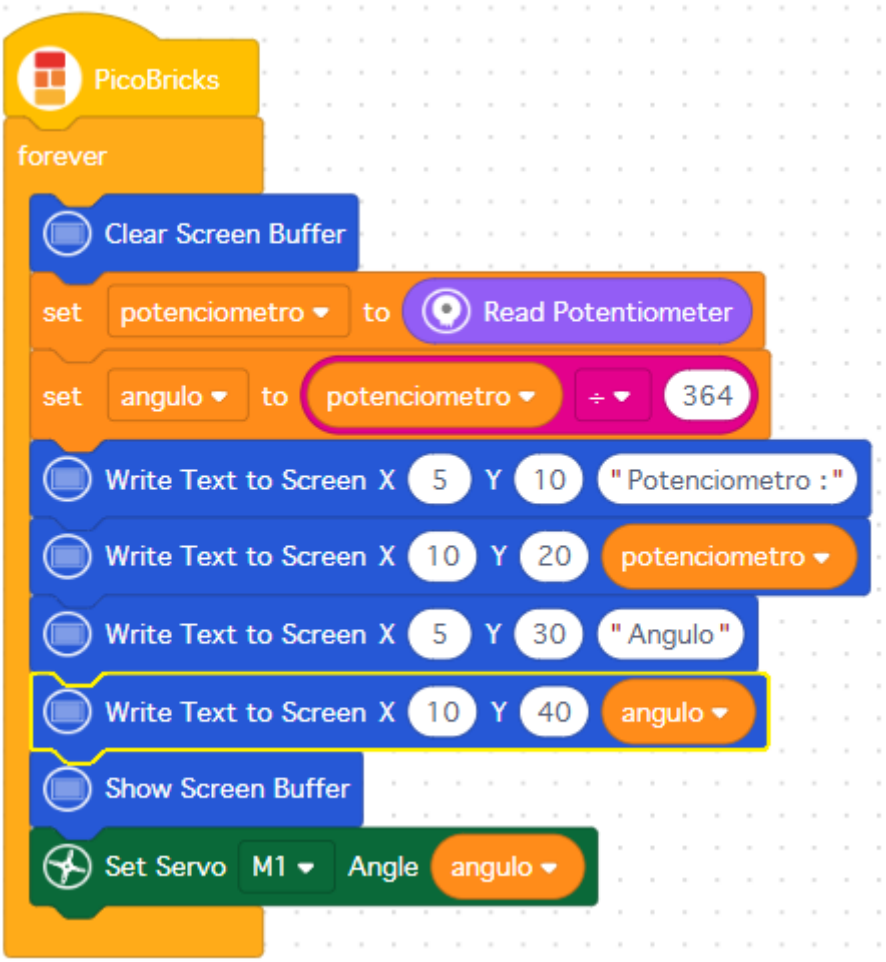


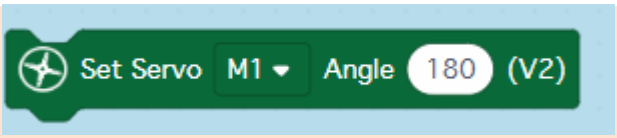
Mapeo

Utilizaremos la situación G de **mapeo** https://libros.catedu.es/books/pico-bricks/page/mapeo por lo tanto dividiremos la posición del potenciómetro por 364

Software

El programa es el siguiente:



No utilizar esta instrucción 

Resultado

https://www.youtube.com/embed/zspyvn8PN4g

Te atreves a... manejar dos servos, por ejemplo con el mando a distancia de IR

Relé

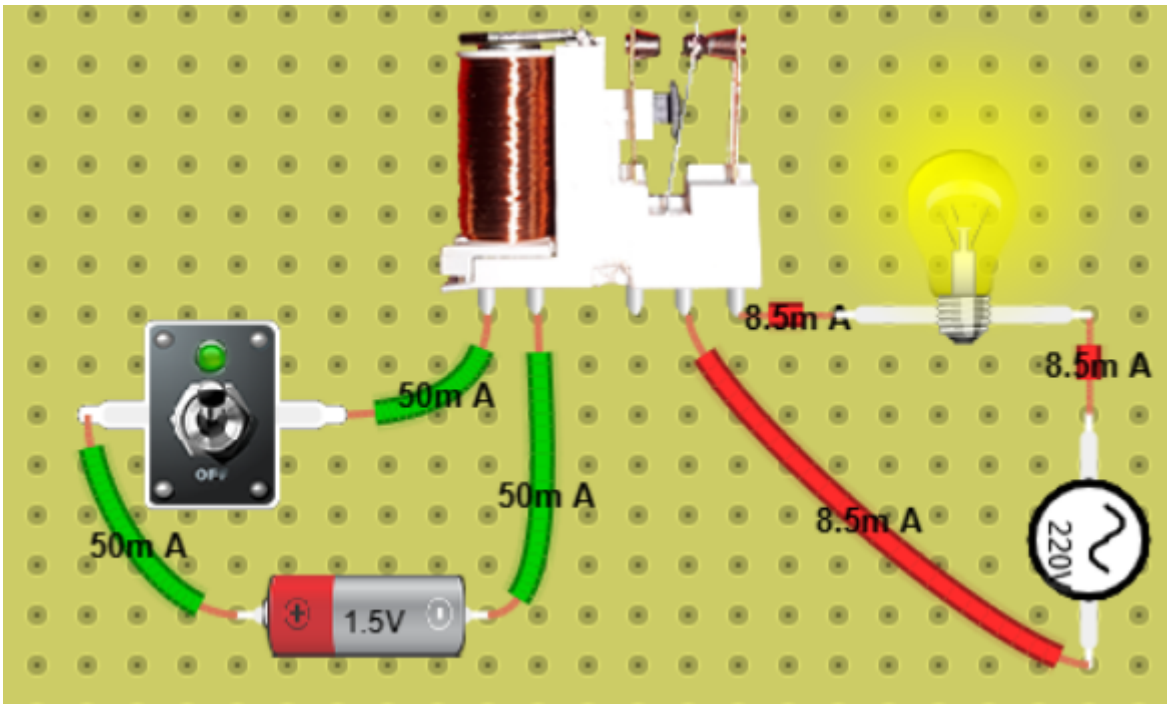
Enunciado

> Buscamos un circuito que cuando haya luz, se encienda y cuando hay oscuridad se apaga. Es al revés de una **luz crepuscular** es decir, que cuando sea de noche se encienda y si es de día que se apague automáticamente

P: **¿Por qué lo hacemos al revés?.**
R: Por que nos evitamos **retroalimentaciones** es decir, si hay oscuridad, se enciende pero como se enciende es de día, por lo que se apaga pero como es de noche se enciende, ... etc... y comienza a hacer clack, clack, clack, clack, clack, clack, clack, clack,...
P **¿Pero entonces cómo se soluciona en las luces públicas?**
R: El sensor LDR no se dirige a las luces a encender, o incluso con una pantalla que haga sombra.

Relé

Un relé es un interruptor activado por un electroimán, lo que permite independizar los circuitos. En el dibujo se ve que el circuito rojo de 220V esta separado del verde, de sólo 1.5V. Pero es el circuito verde que al funcionar, hace que el electroimán mueva el interruptor del relé y encienda la bombilla. El objetivo es que he podido encender una bombilla de 220V sin tocar los 220V peligrosos. En el circuito verde, el interruptor puede ser un Arduino. Experiméntalo en este simulador.

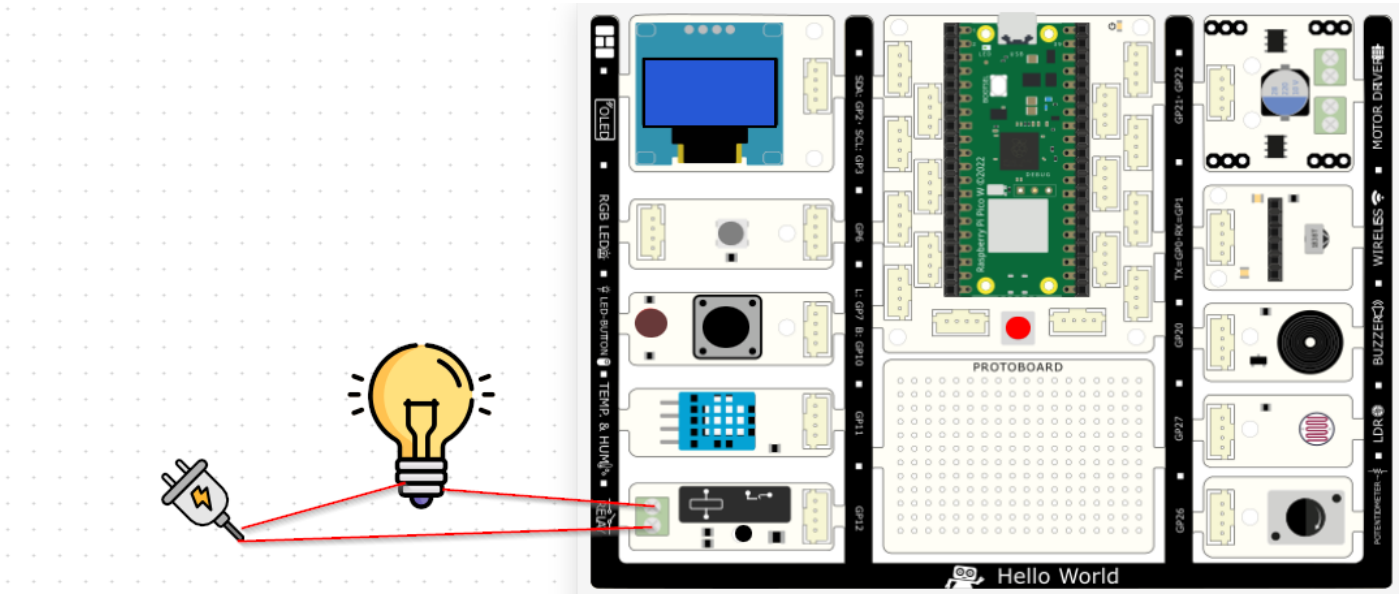


Circuito con Picobriks

ATENCIÓN,
-NO RECOMENDABLE PARA PRIMARIA.
-NO DEJAR A LOS ALUMNOS SOLOS CON ESTE CIRCUITO.
-SE UTILIZAN TENSIONES DE 220V
-EL CONECTOR VERDE DEL RELÉ ES MUY PEQUEÑO:
- los dos bordes están muy próximos, NO UTILIZAR CABLE CON HILOS peligro de que algún hilo cortocircuite
- no utilizar potencias superiores a 20W

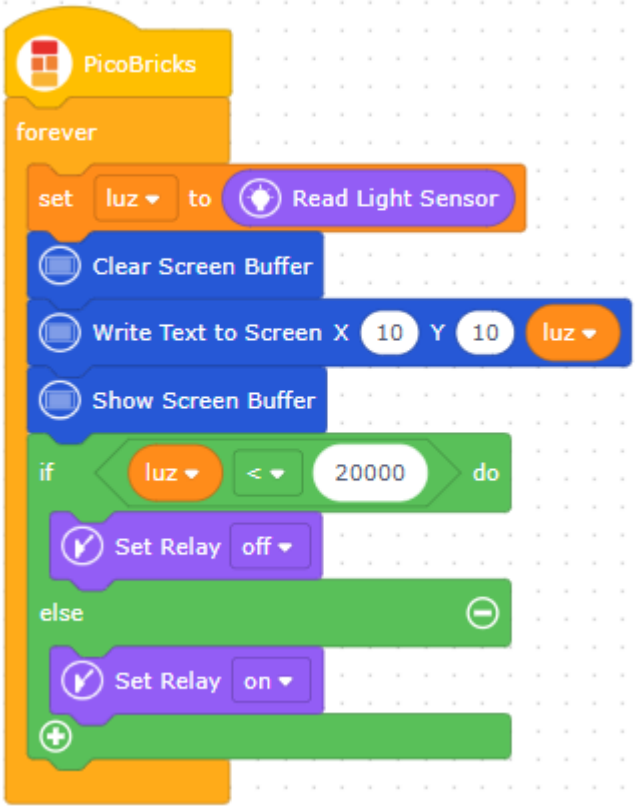
Si no quieres estos peligros,  una pila y la bombilla doméstica por una pequeña de maquetas :

El circuito que se propone es utilizar el relé para que cierre un circuito que encienda una bombilla. El circuito con los cables rojos estarían a 220V, luego **peligro !!!**

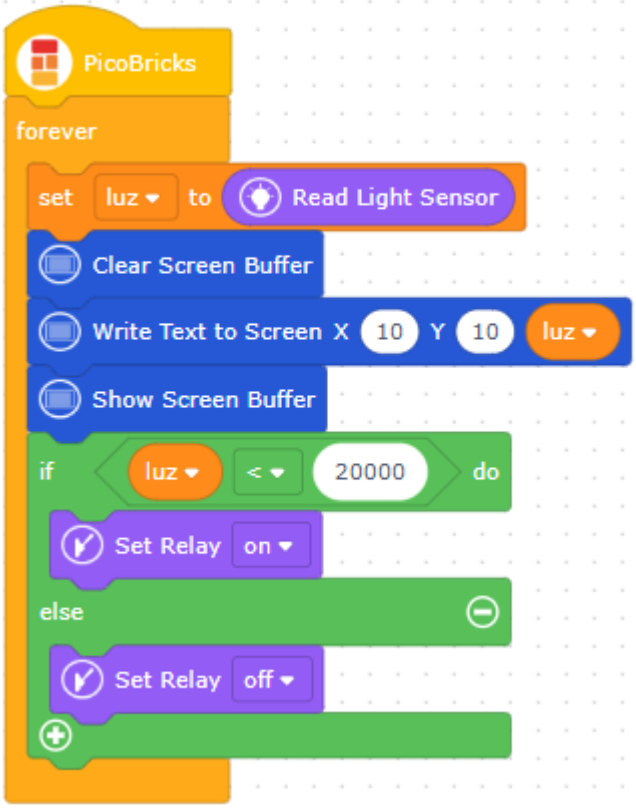


Programa

El programa lee la intensidad luminosa y lo visualiza por la pantalla OLED. Si es inferior a 20.000 eso quiere decir que hay luz, por lo tanto que cierre el circuito.



si quieres hacerlo crepuscular como es en la vida real, cambia el estado del relé pero aleja el sensor de la bombilla o pon una pantalla de sombra



Resultado

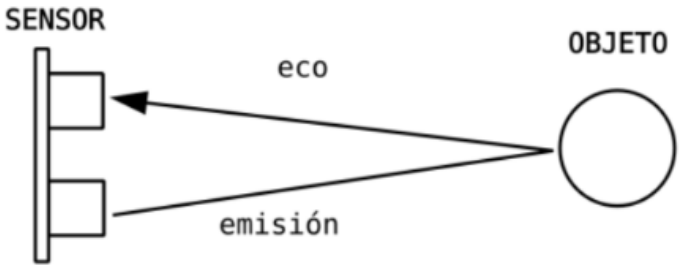
https://www.youtube.com/embed/7CYORIz7Ze8

Sensor de distancia de ultrasonidos

¿Qué es el sensor de distancia HC-SR04?

Es un sensor digital de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 350 cm. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno.
No es un sensor preciso, con una ligera inclinación de la superficie ya da lecturas erróneas pero es muy barato

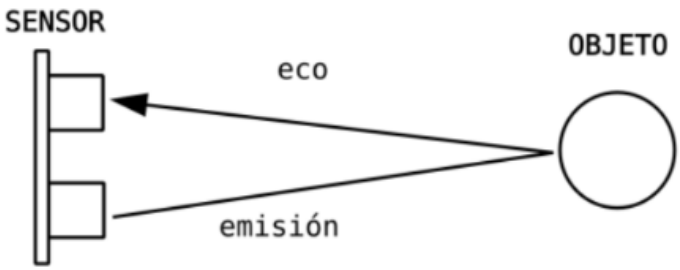
El más común es el **HC-SR04** que tiene 4 pines de conexión: **VCC** **Trig** (Disparo del ultrasonido) **Echo** (Recepción del ultrasonido) y **GND** aunque en algunos modelos como el de Elecfreaks tiene 3 pines. Integra Trig y Echo en uno sólo.
La distancia se calcula con esta fórmula:
Distancia en cm = {(Tiempo en segundos entre Trig y el Echo) * (V.Sonido 34000 en cm/s)} / 2
Si programas en código, tienes que utilizar la fórmula anterior, previamente tienes que programar el cálculo del tiempo entre una emisión de un pulso en Trg y la respuesta en Echo.
Si utilizas bloques, no es necesario, seguro que hay un bloque que lo hace todo



Ejemplos de uso:

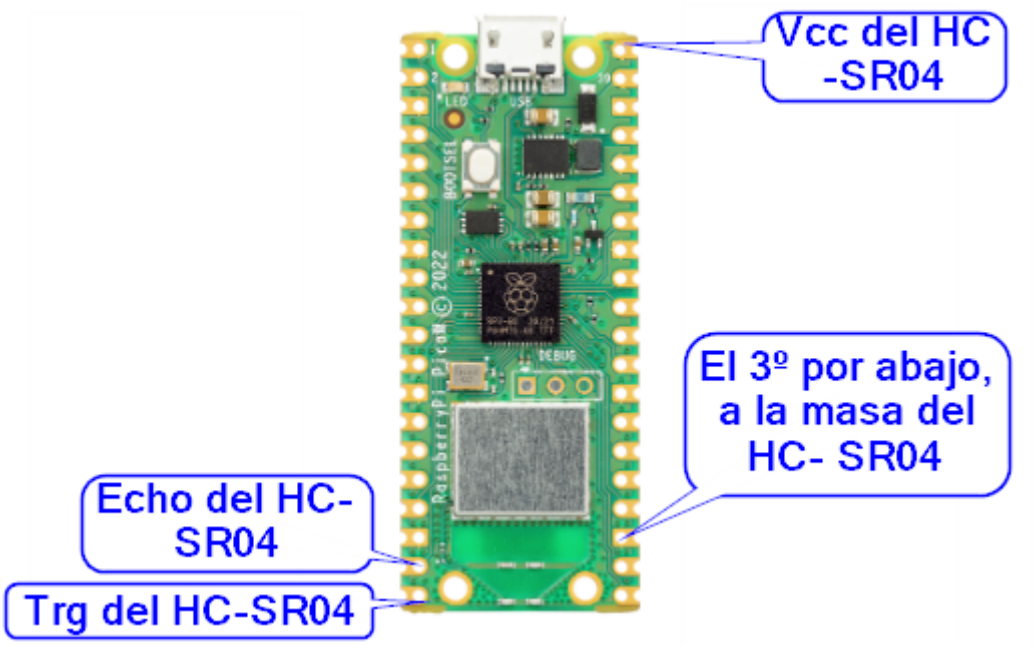
- Alarma en Domótica con Arduino
- Piano invisible en Arduino con ArduinoBlocks,

- Sensor parking en Arduino con ArduinoBlocks
- Piano invisible en Arduino con mBlock
- Sensor parking en Arduino con mBlock
- Sensor de distancia de ultrasonidos con Picobricks

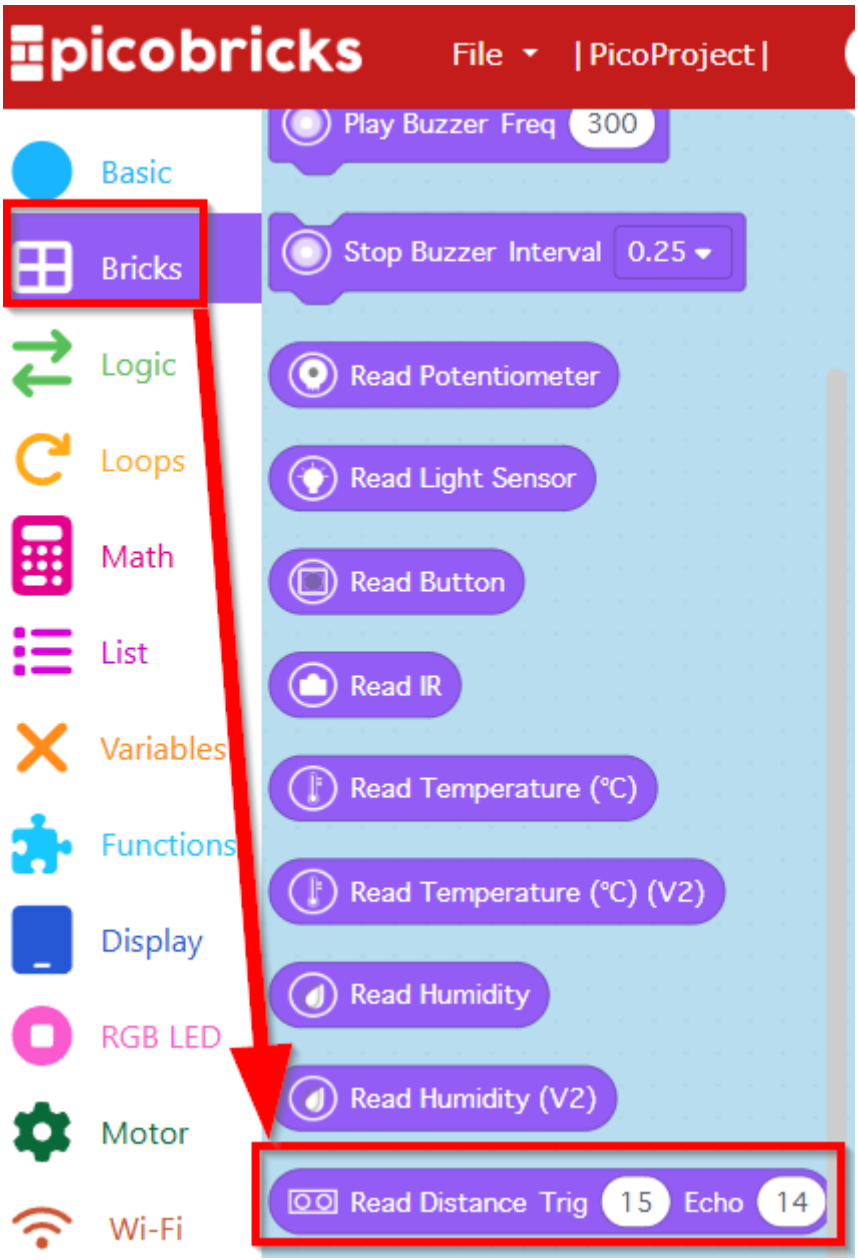


Conexión con la Raspberry Pi Pico E

Si miras https://libros.catedu.es/books/pico-bricks/page/que-es-pico-bricks verás el esquema para poder conectar los pines del HC-SR04 con la Raspberry, te lo mostramos aquí



La conexión de Echo y de Trg es arbritaria, lo hemos puesto en GP14 y GP15 que es la que recomienda el programa Picobricks

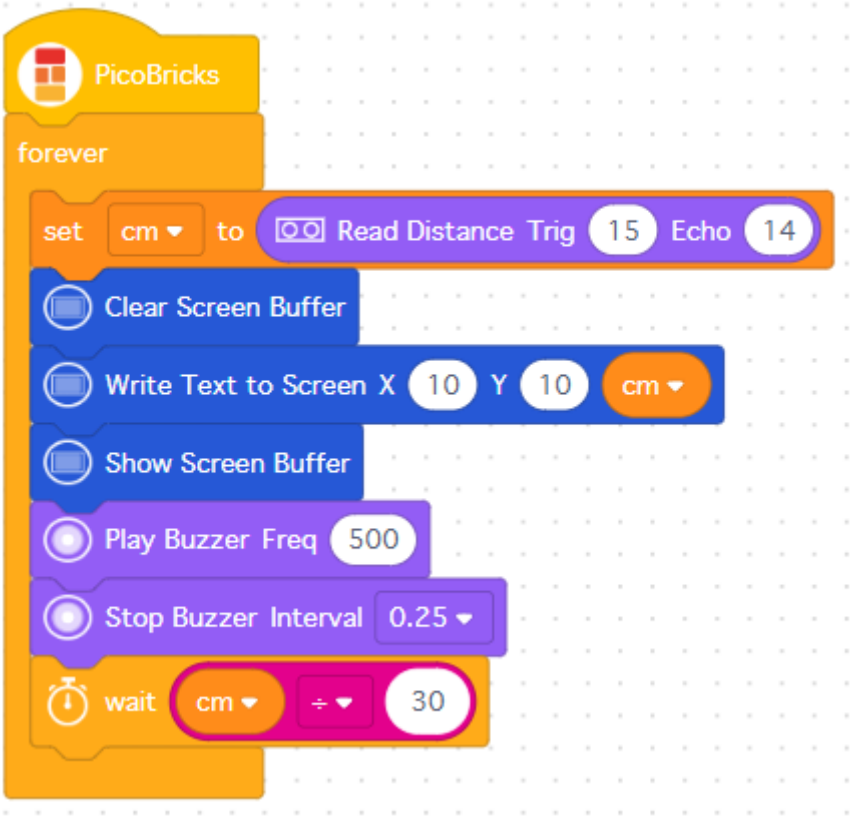


Enunciado

> Mostrar por la pantalla de OLED la distancia y a la vez que suene un radar como en los asistentes de parking de los coches

Programa

Para hacer el pitido intermitente en función de los centímetros del objeto, se utiliza una pausa, como la espera en cm convertirla a segundos es muy grande, se divide por 30, puedes modificar este valor a tu gusto

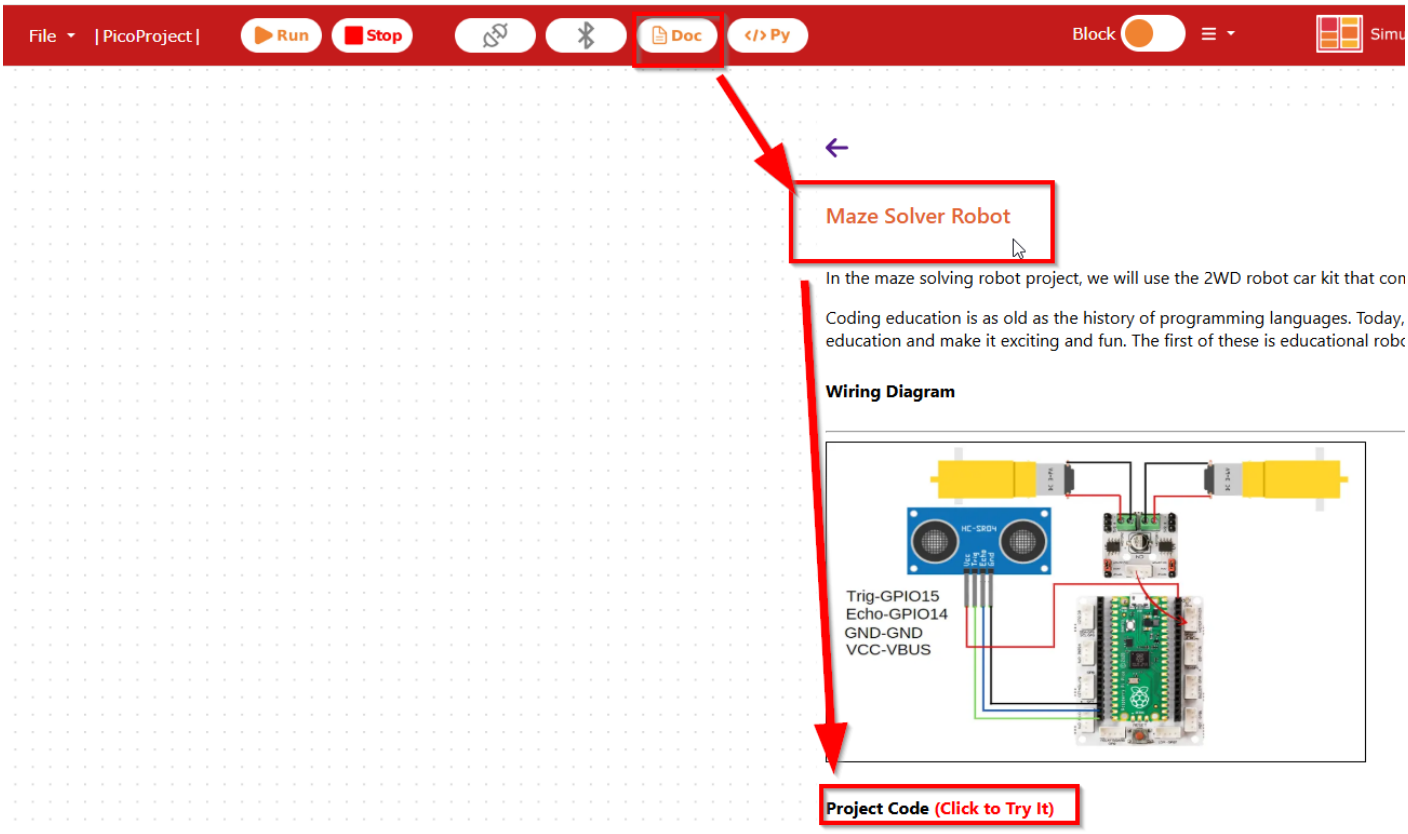


Resultado

https://www.youtube.com/embed/ZV5cjfkenSE

> **Te atreves a...** Juntar el Pico coche con un sensor de distancia de tal manera que vaya autónomo evitando los obstáculos como un romba

¿No te atreves? Pues aquí tienes la solución:


File
| PicoProject |
Run
Stop
Doc
Py
Block
Maze Solver Robot
In the maze solving robot project, we will use the 2WD robot car kit that con
Coding education is as old as the history of programming languages. Today,
education and make it exciting and fun. The first of these is educational robo
Wiring Diagram
Trig-GPIO15
Echo-GPIO14
GND-GND
VCC-VBUS
Project Code (Click to Try It)