


# PicoBlockly

- Cuatro programas a elegir
- Interface
- Conexión
- Dos formas de ejecutar los programas
- PROYECTO BLINK
- PROYECTO ACTION-REACTION
- PROYECTO Autonomous Lighting
- PROYECTO Thermometer
- PROYECTO Graphic Monitor
- PROYECTO Dominate the Rhythm
- PROYECTO Show Your Reaction
- PROYECTO My Timer
- PROYECTO Alarm Clock
- PROYECTO Know Your Color
- Algo diferente PROYECTO IR

# Cuatro programas a elegir


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



### PICOJR

A simple and visual-based coding platform for kids who are new to robotic coding.


[VIEW MORE](#)



### PICOBLOCKLY

Easily code PicoBricks with block coding and see python code at the same time.


[VIEW MORE](#)



### PICOPY

Do high-level coding with text-based python.

[VIEW MORE](#)



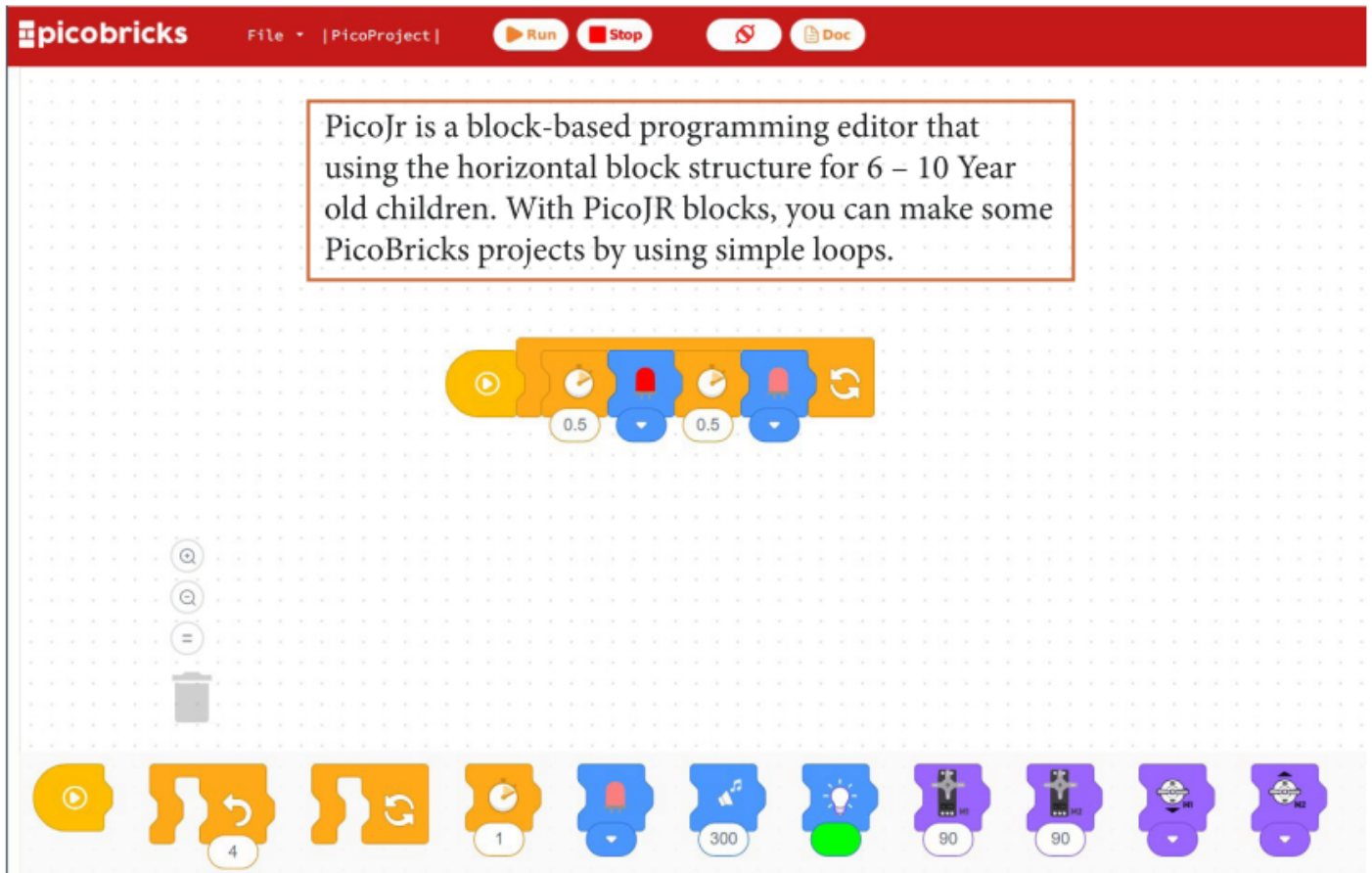
### SIMULATOR

With PicoBricks Simulator you can experience using PicoBricks.

[VIEW MORE](#)

## PICOJR

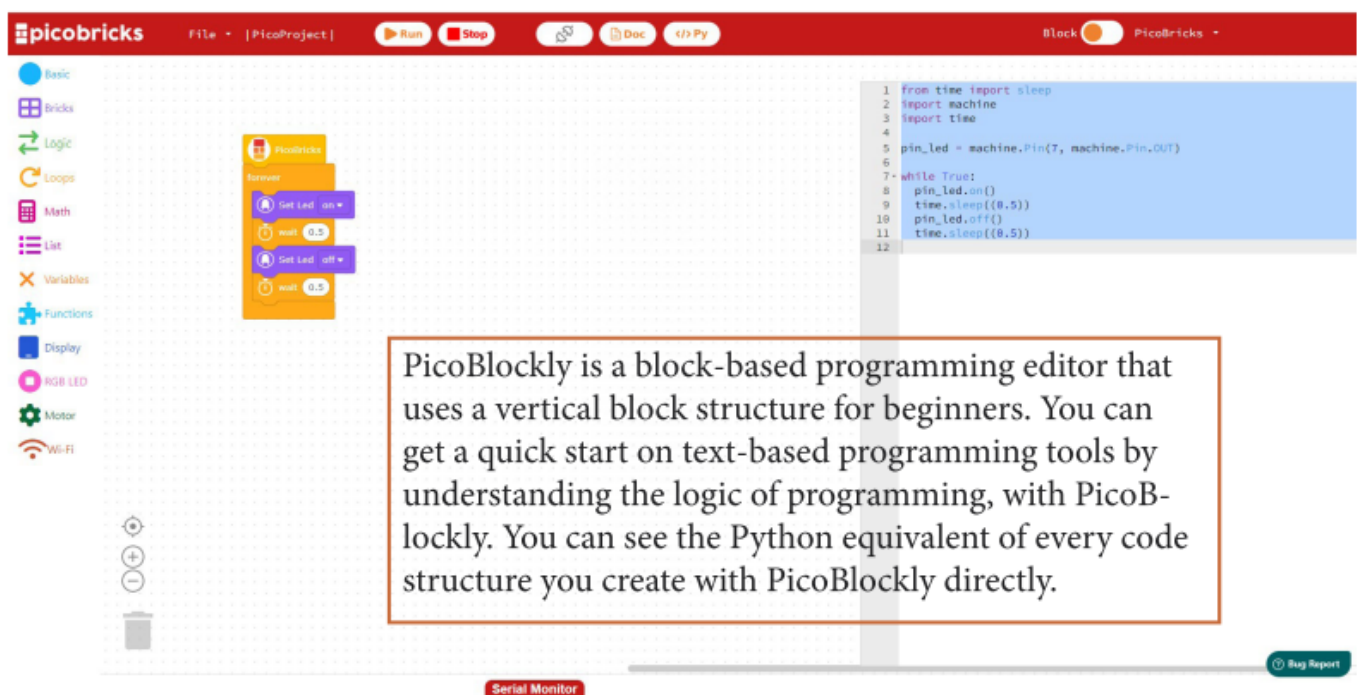
Pensado para programar Picobriks con bloques para etapas de 8 a 10 años con un **mínimo de instrucciones**



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

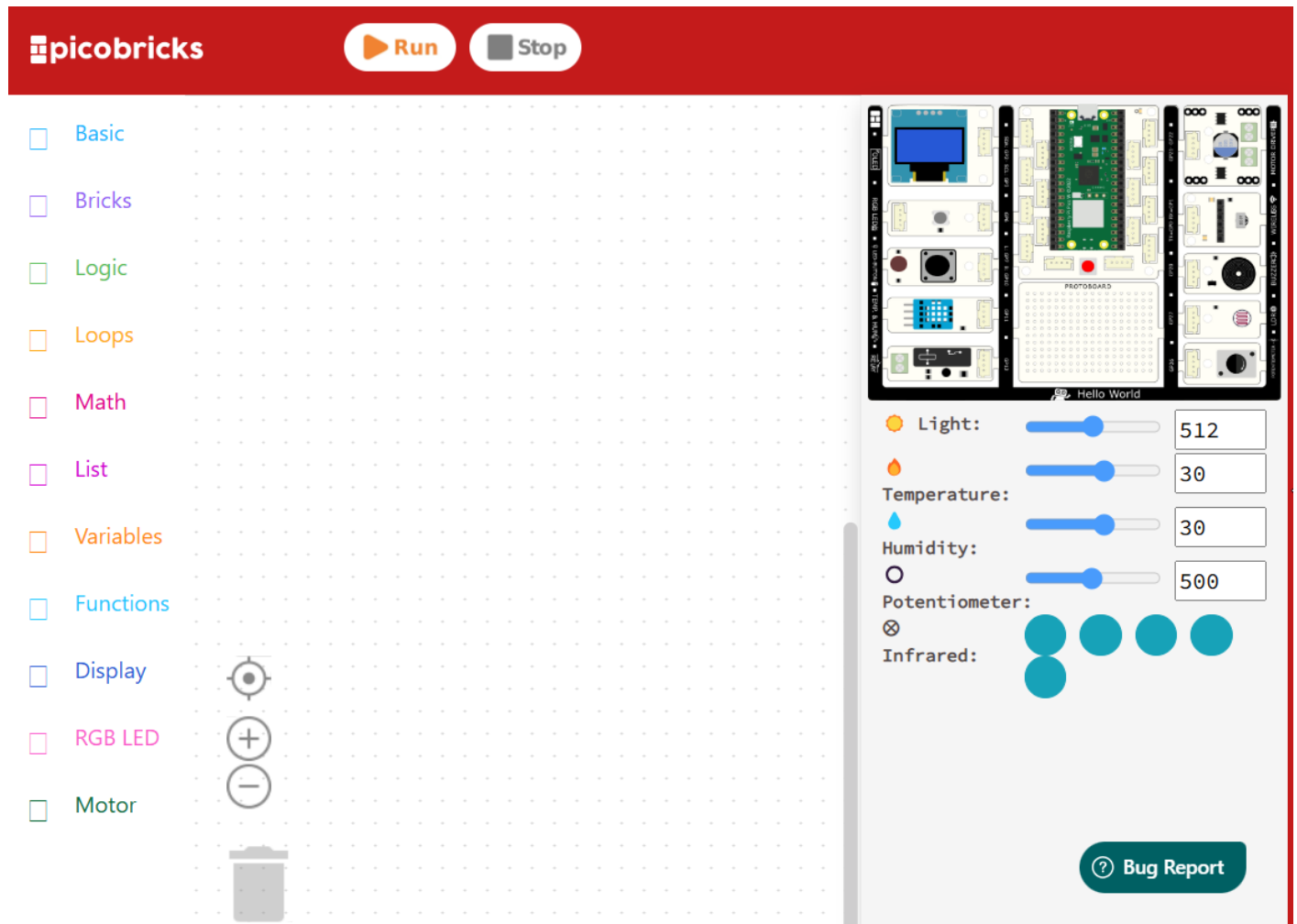


## PicoPy

Para poder editar en Python, lo trataremos en la sección de código

## Pico simulator

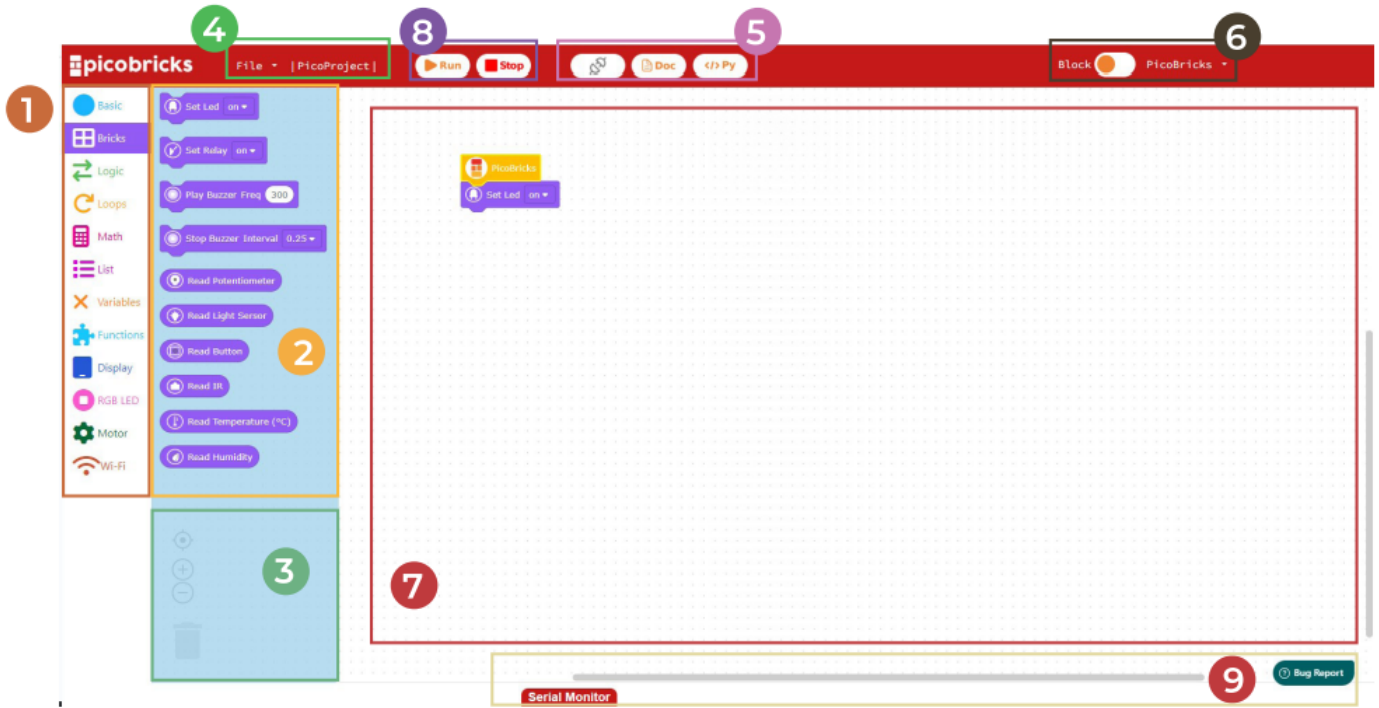
Es un simulador online que permite realizar proyectos sin tener físicamente la Pico bricks



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 bluetooth
  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

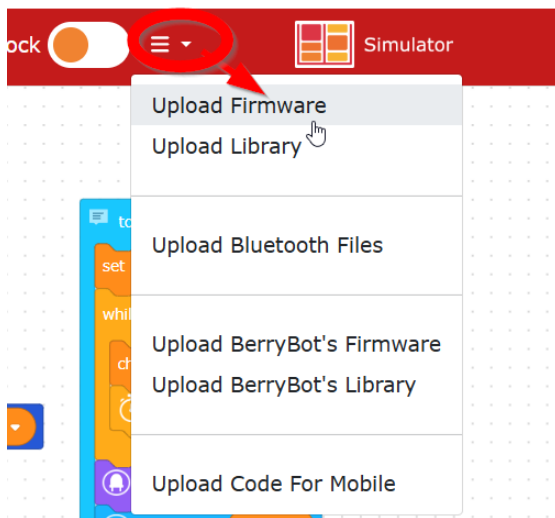
**P: ¿Qué es eso de "firmware"?**

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?**

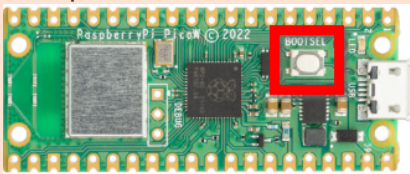
R: Digamos que como se graba en los chips, es un medio camino entre software y hardware, para diferenciarlo del software habitual.

Entramos en el menú y descargamos el firmware

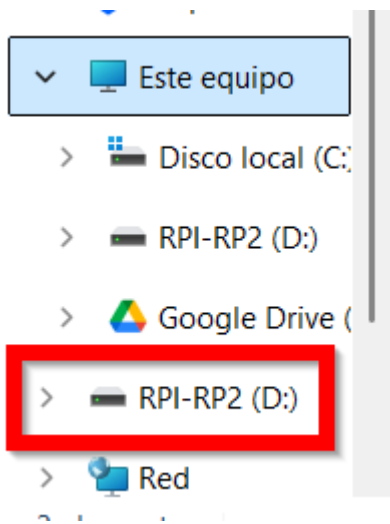


**ATENCIÓN, poner PicoBricks en modo Bootloader**

- 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)



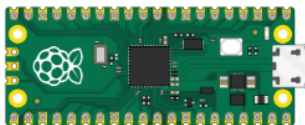
Descargamos el correspondiente al PicoW **Y LO GRABAMOS EN LA UNIDAD NUEVA** en mi caso RPI-RP2 (D:)

### FIRMWARE UPDATE

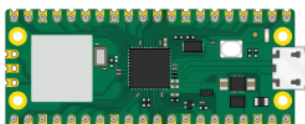


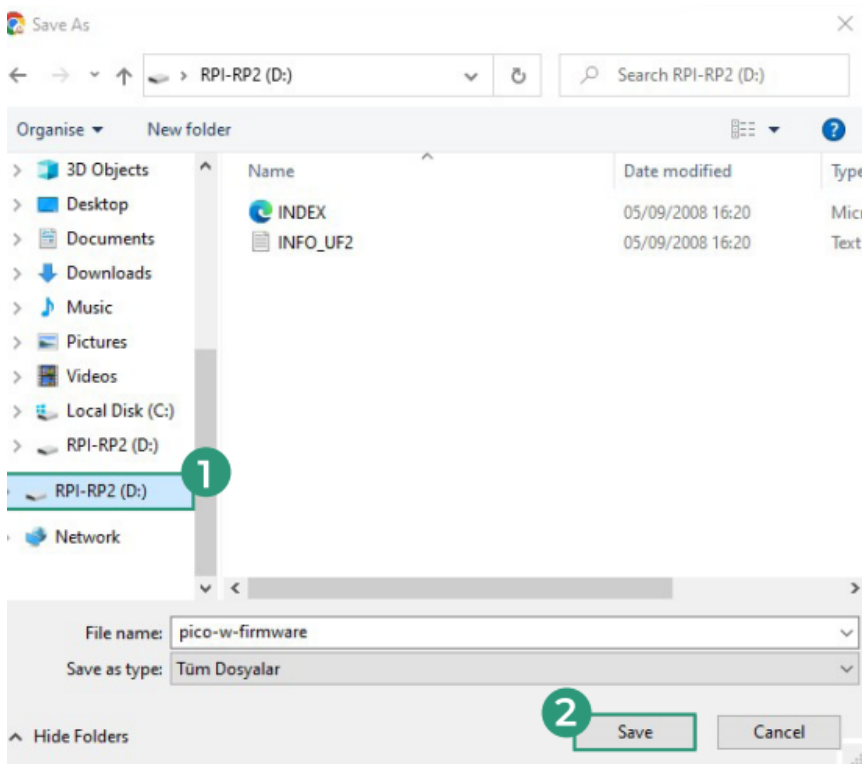
to firmware update hold down the **BOOTSEL** button while plugging the board into USB

Pico



Pico W



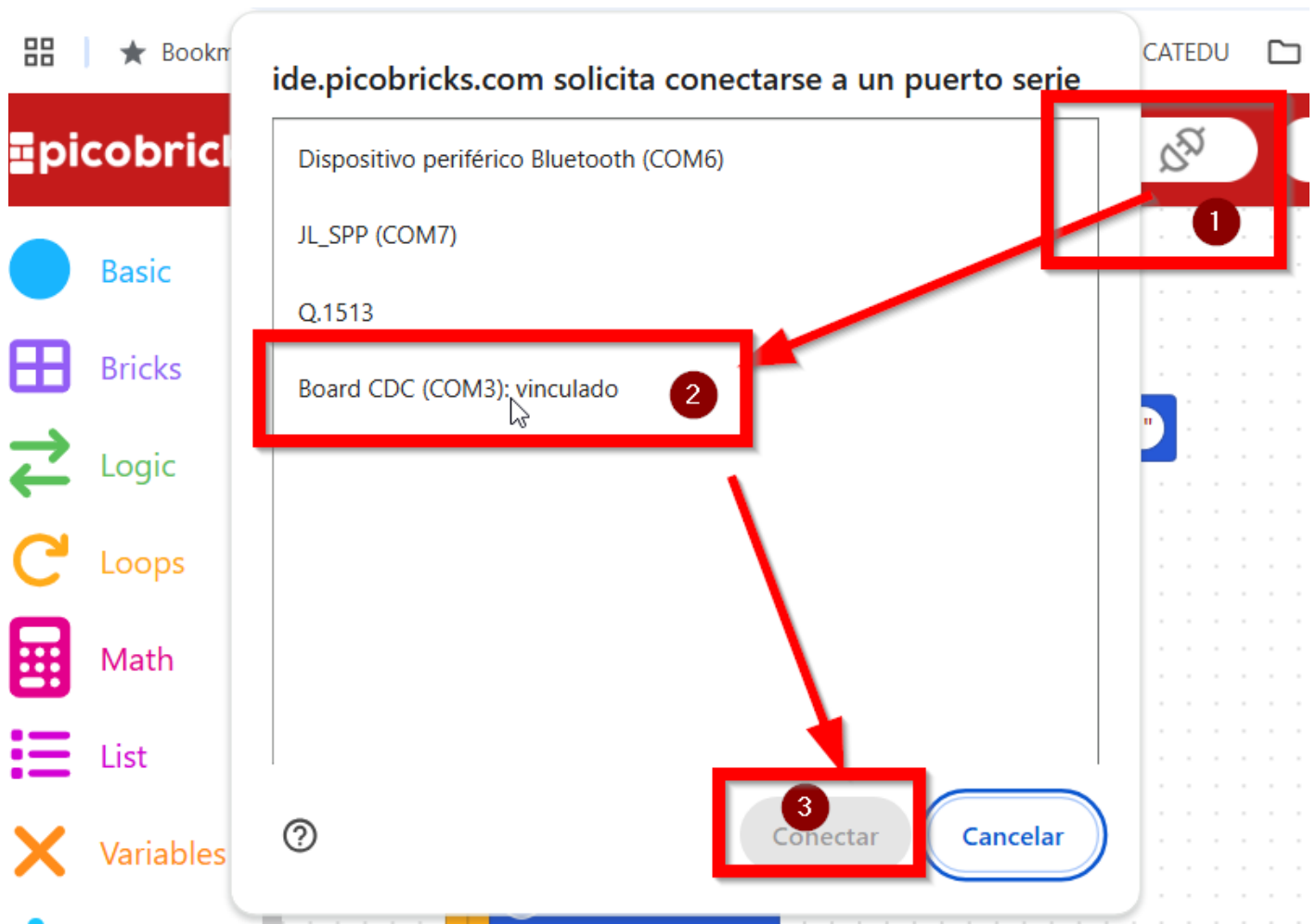


Fuente Pico Bricks IDE Book CC-BY-SA <https://picobricks.com/pages/idebook> ver créditos

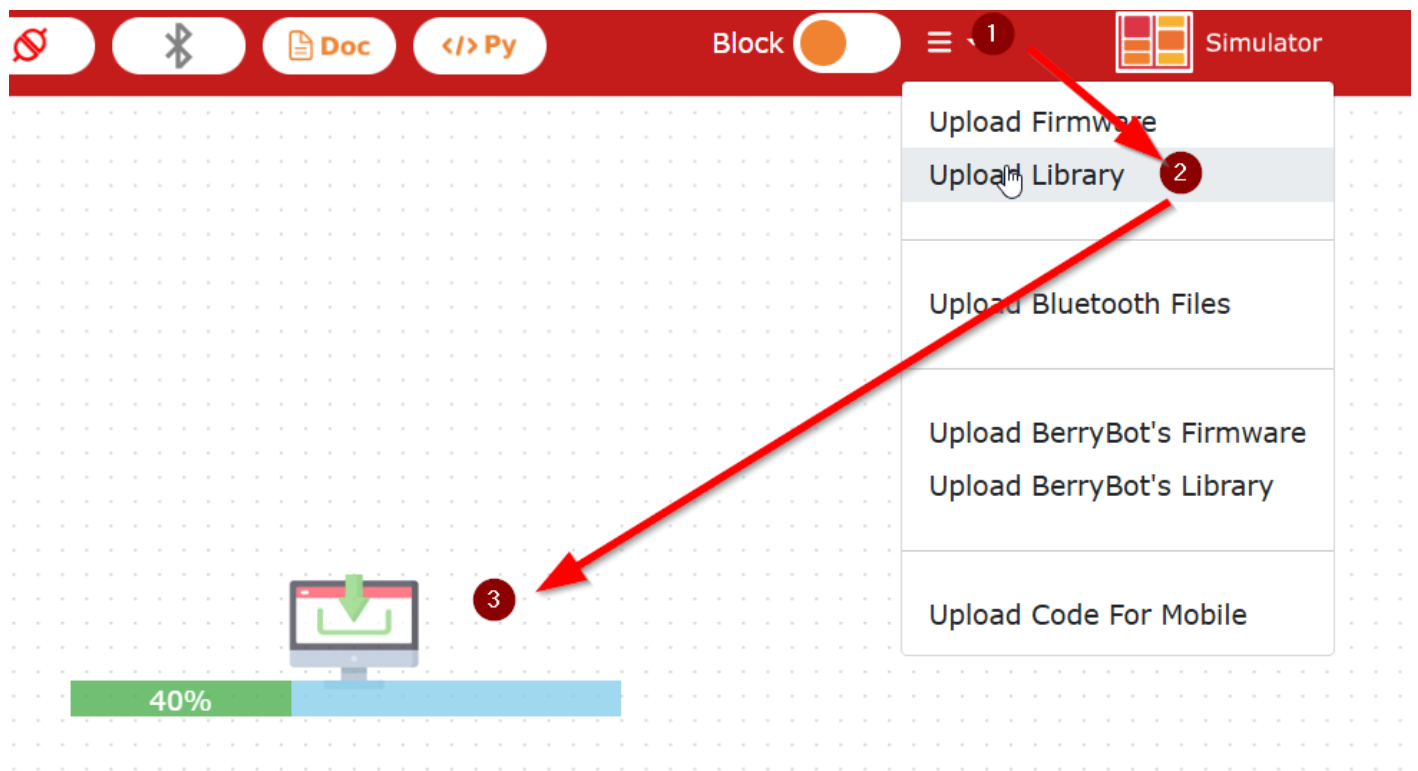
Una vez grabado el firmware, esperamos a que nos salga un mensaje: *Please conect to the board*

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

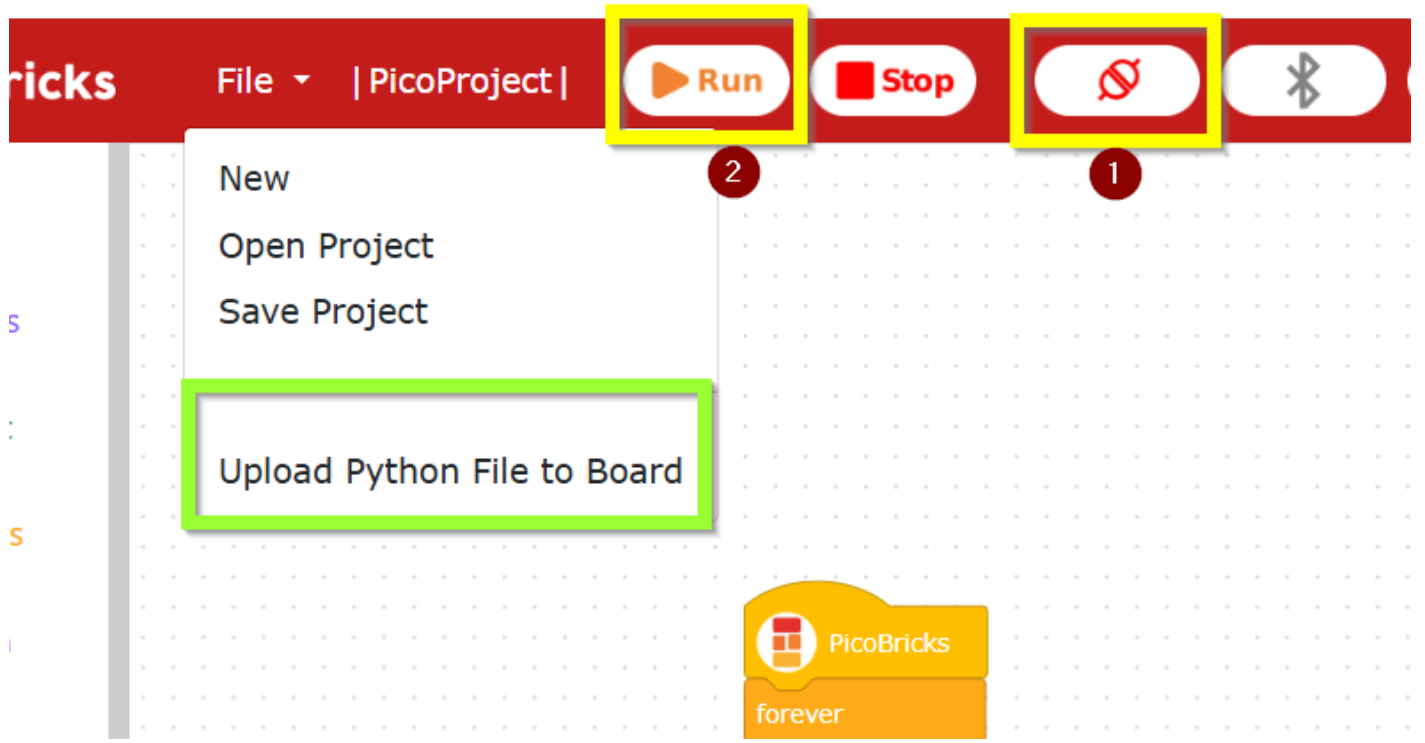
La forma más fácil de trabajar es **EN VIVO** es decir, que los programas se ejecuten **desde nuestro ordenador** es la más rápida y para ello necesita que el PicoBricks tenga el Firmware correspondiente dentro (tal y como hemos visto)

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.

- 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**

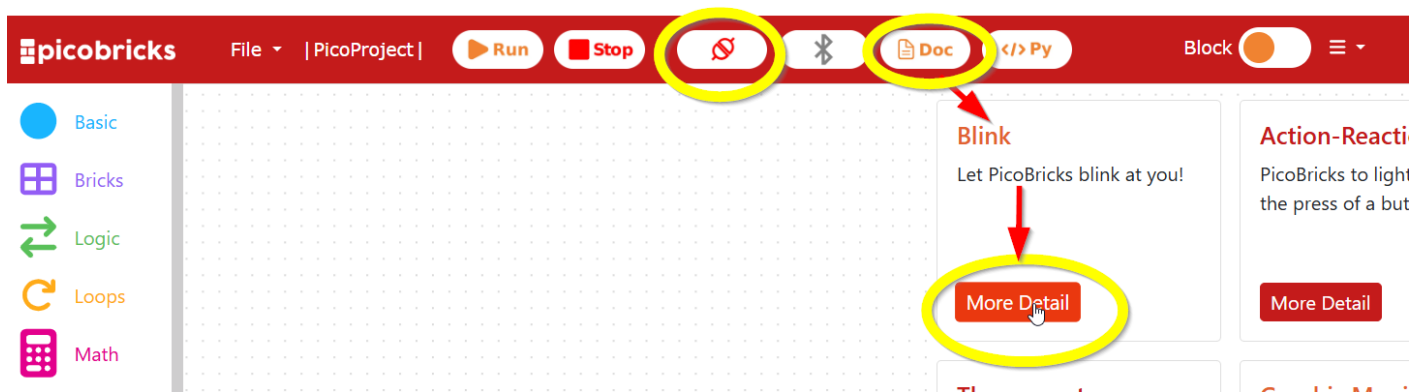


# PROYECTO BLINK

Extraído 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

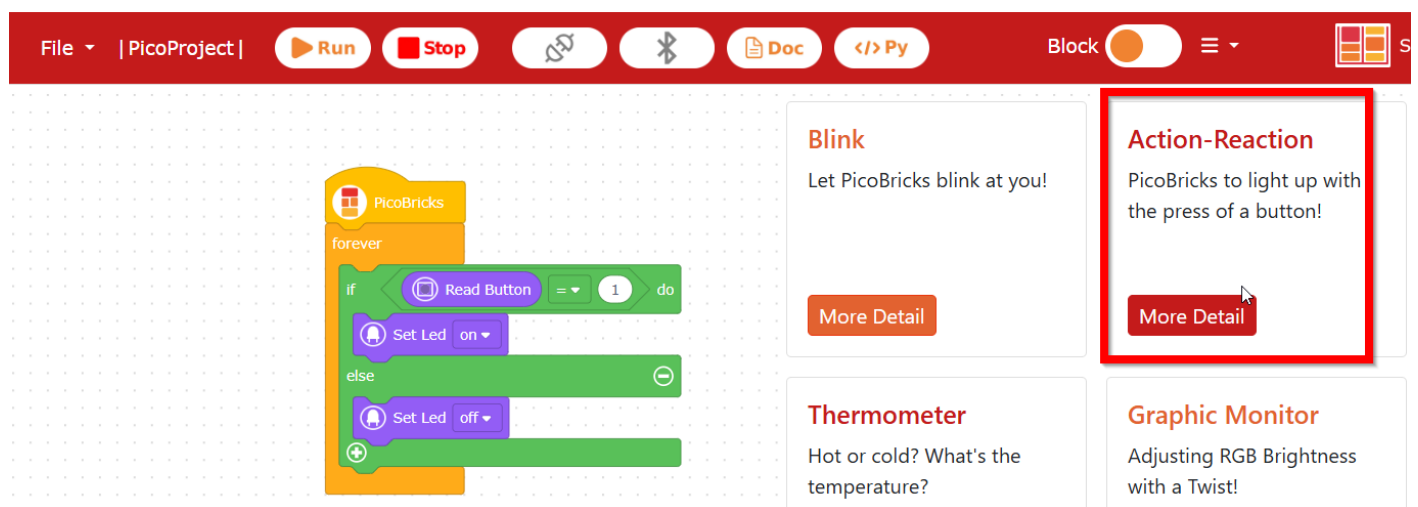
[https://www.youtube.com/embed/nYPWAC\\_SHWA](https://www.youtube.com/embed/nYPWAC_SHWA)



# PROYECTO ACTION-REACTION

Extraído 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:



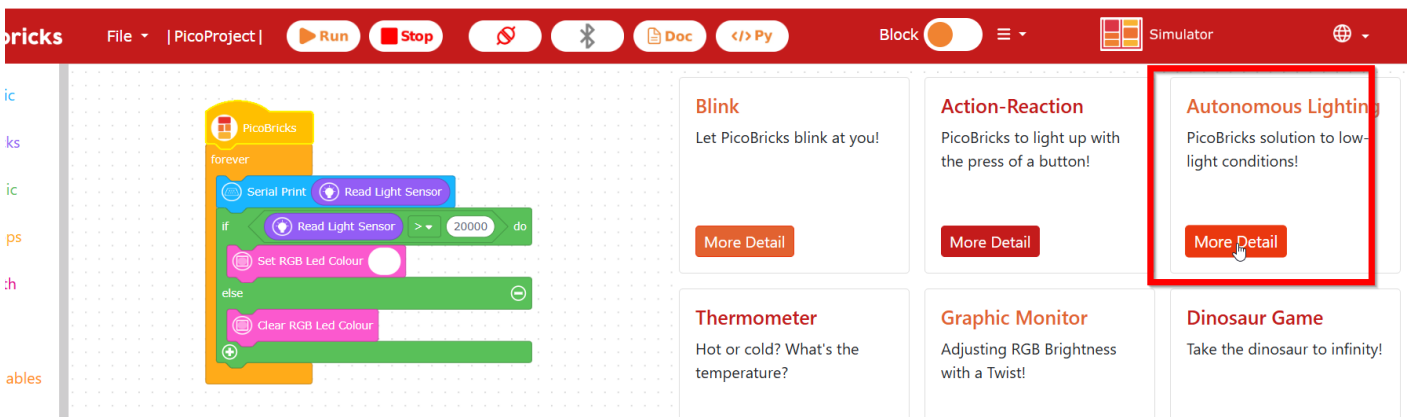
Resultado

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

# PROYECTO Autonomous Lighting

Extraído 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](#)

The screenshot shows the PicoBricks IDE interface. At the top is a red toolbar with buttons for File, PicoProject, Run, Stop, a link icon, a Bluetooth icon, Doc, and a code editor icon. The main workspace is a grid where a script is built. The script starts with a yellow 'PicoBricks' block, followed by an orange 'forever' loop block. Inside the loop, there is a 'wait 1' block, followed by four blue blocks: 'Clear Screen Buffer', 'Write Text to Screen X 15 Y 10 "Temperature: "', 'Write Text to Screen X 55 Y 30 Read Temperature (°C)', and 'Write Text to Screen X 35 Y 50 "degrees"'. The loop ends with a 'Show Screen Buffer' block. On the right side, there are two panels. The top panel is titled 'Blink' and contains the text 'Let PicoBricks blink at you!' and a 'More Detail' button. The bottom panel is titled 'Thermometer' and contains the text 'Hot or cold? What's the temperature?' and a 'More Detail' button. The 'Thermometer' panel is highlighted with a red border.

File | PicoProject | Run Stop [Link] [Bluetooth] Doc </> Py Block

**PicoBricks**

forever

wait 1

Clear Screen Buffer

Write Text to Screen X 15 Y 10 "Temperature: "

Write Text to Screen X 55 Y 30 Read Temperature (°C)

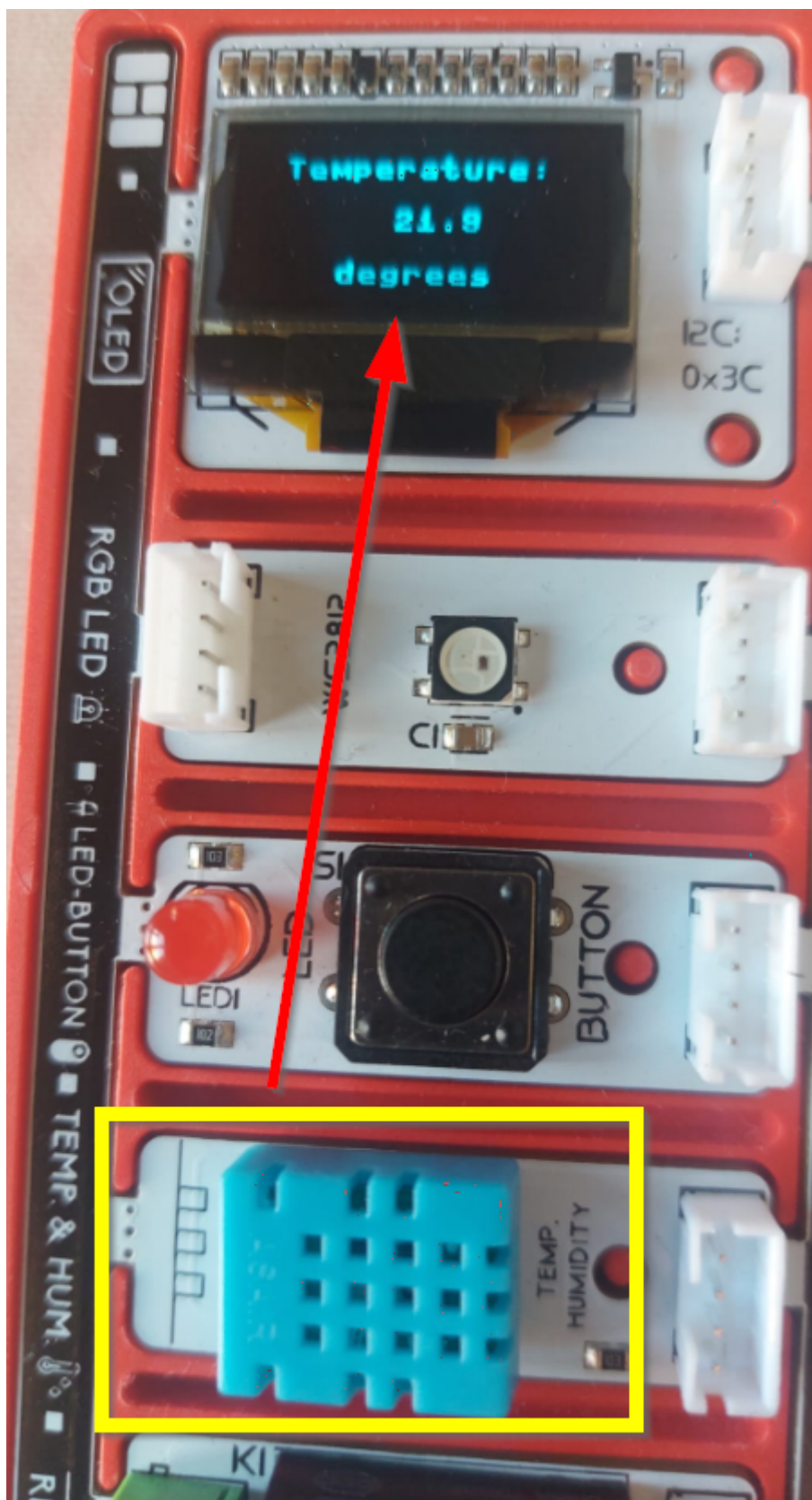
Write Text to Screen X 35 Y 50 "degrees"

Show Screen Buffer

**Blink**  
Let PicoBricks blink at you!  
More Detail

**Thermometer**  
Hot or cold? What's the temperature?  
More Detail

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



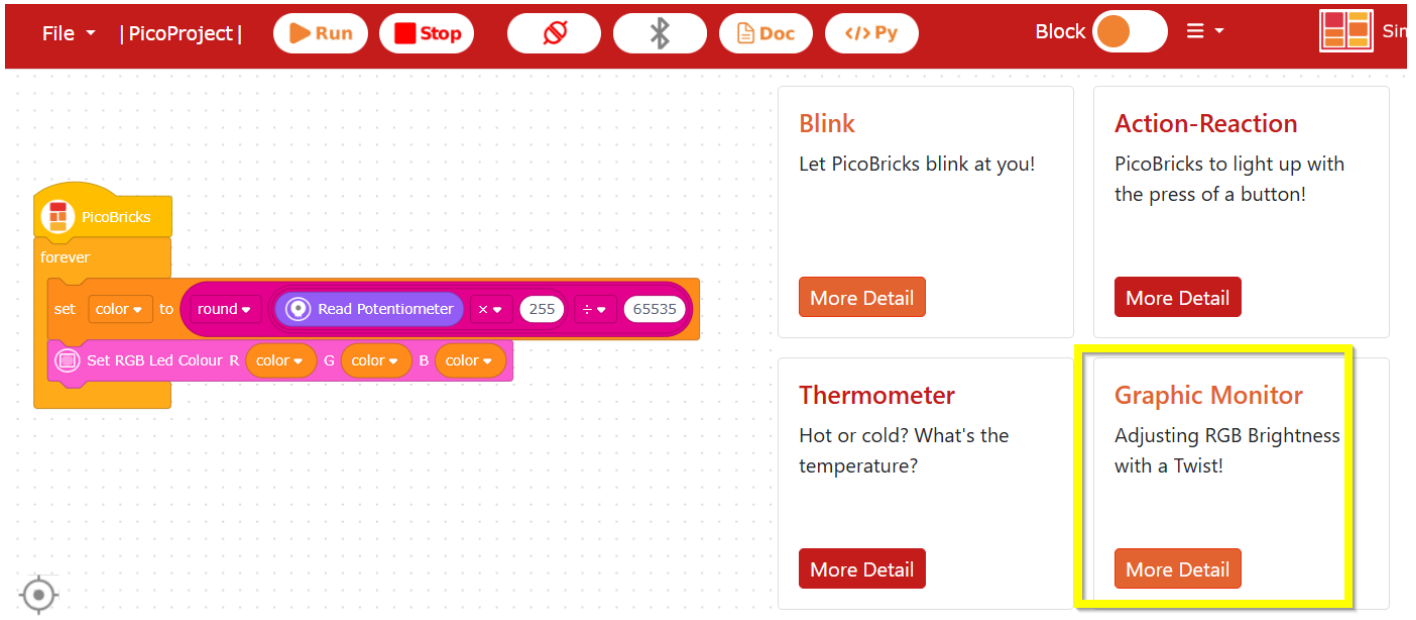
Recomendamos este proyecto cargarlo en el PicoBricks y así funciona autónomo sin necesidad de PC, con lo que se puede colocar en el exterior y ver la temperatura que hace simplemente alimentandolo con un PowerBank en el cable USB

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

Extraído de *Pico Bricks IDE Book* CC-BY-SA <https://picobricks.com/pages/idebook> ver [créditos](#)



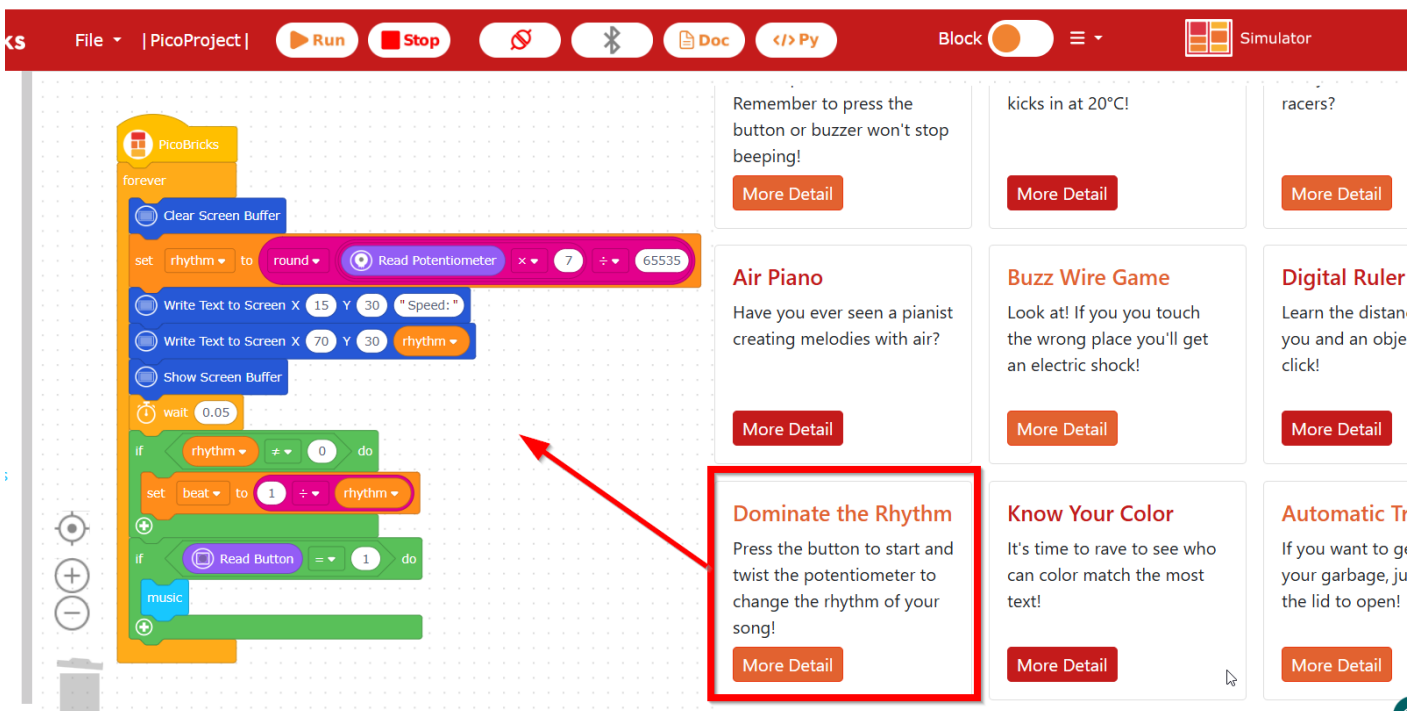
Resultado

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

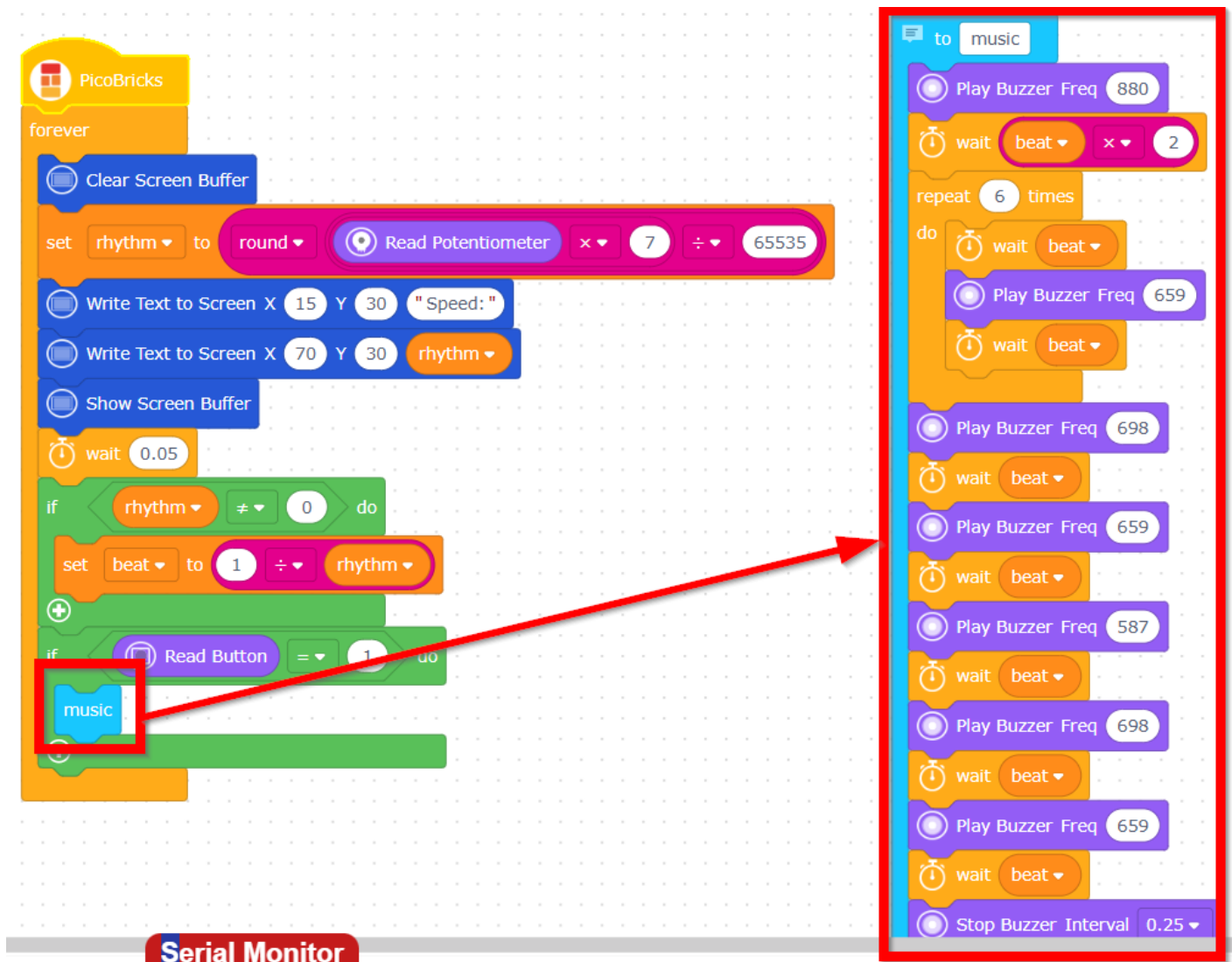
# PROYECTO Dominate the Rhythm

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

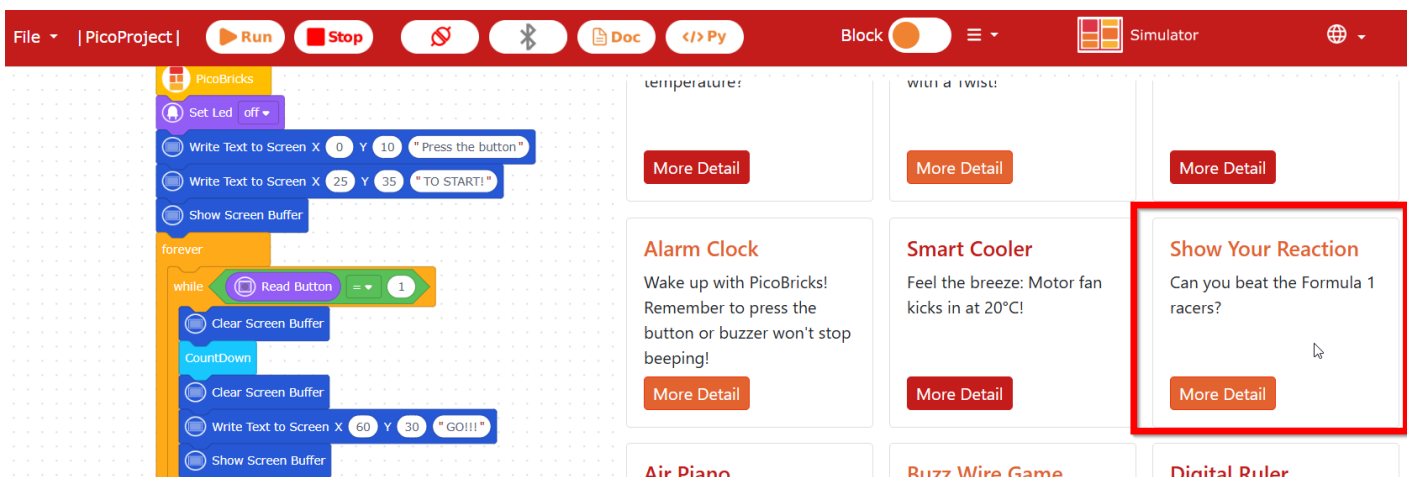
Como se puede ver en el resultado, la primera vez suena la música a un ritmo número 4 pero en la segunda vez subimos con el potenciómetro al ritmo máximo 7 y la música suena más deprisa

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

# PROYECTO Show Your Reaction

Extraído 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 !

The screenshot shows the PicoBricks IDE interface. The top bar includes a 'File' menu, a 'PicoProject' title, and buttons for 'Run', 'Stop', 'Bluetooth', 'Doc', '</> Py', 'Block', and 'Simulator'. The left pane displays a block-based program for a timer. The right pane shows a list of projects, with 'My Timer' highlighted in a red box.

**Program Logic (Left Pane):**

- Set hourv to 0
- Set minutev to 0
- Set secondv to 0
- Set clue to 0
- Forever loop:
  - If clue == 0 do:
    - hour
  - Else if clue == 1 do:
    - minute
  - Else if clue == 2 do:
    - second
  - Else if clue == 3 do:
    - clock
    - timerFunction
  - Else:
    - control
    - If Read Button == 1 do:
      - Set secondv to 0
      - Set clue to 3

**Project List (Right Pane):**

- It's time to rave to see who can color match the most text! (More Detail)
- If you want to get rid of your garbage, just wait for the lid to open! (More Detail)
- My Timer** (Tick-Tock! Countdown begins!) (More Detail)
- Magic Lamb (Make Clap-Clap!) (More Detail)
- Maze Solver Robot (Lost no more: navigate mazes with PicoBricks Maze Robot!) (More Detail)
- Night And Day (Night and day detector: A fun game with light sensor magic!) (More Detail)
- Piggy Bank (Don't forget to feed your Piggy Bank with your money!) (More Detail)
- Smart Greenhouse (Make your plant pal!) (More Detail)

OneDrive - Gobierno de Aragón Report

<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

The screenshot displays the PicoBricks IDE interface. The top bar includes a menu (File), a project name (PicoProject), and buttons for Run, Stop, and other functions. The main workspace shows a script for an alarm clock. The script starts with a 'PicoBricks' block, followed by 'Write Text to Screen X 25 Y 32 "Good night"', 'Show Screen Buffer', and a 'wait 2' block. A 'forever' loop contains a 'wait 1' block, 'Clear Screen Buffer', and an 'if' block. The 'if' block checks 'Read Light Sensor < 20000' and, if true, executes 'Write Text to Screen X 15 Y 32 "Good Morning"', 'Show Screen Buffer', 'Set RGB Led Colour', 'Play Buzzer Freq 300', and another 'if' block. This second 'if' block checks 'Read Button == 1' and, if true, executes 'Clear Screen Buffer', 'Write Text to Screen X 0 Y 32 "Have a nice day"', 'Show Screen Buffer', and 'Clear RGB Led Colour'.

The sidebar on the right lists several project ideas, each with a 'More Detail' button. The 'Alarm Clock' project is highlighted with a red border. The projects listed are:

- Thermometer: Hot or cold? What's the temperature?
- Graphic Monitor: Adjusting RGB Brightness with a Twist!
- Dinos: Take th
- Smart Cooler: Feel the breeze: Motor fan kicks in at 20°C!
- Show: Can yo racers?
- Air Piano
- Buzz Wire Game
- Dinit:

[https://www.youtube.com/embed/8Drcl\\_YEsFs](https://www.youtube.com/embed/8Drcl_YEsFs)

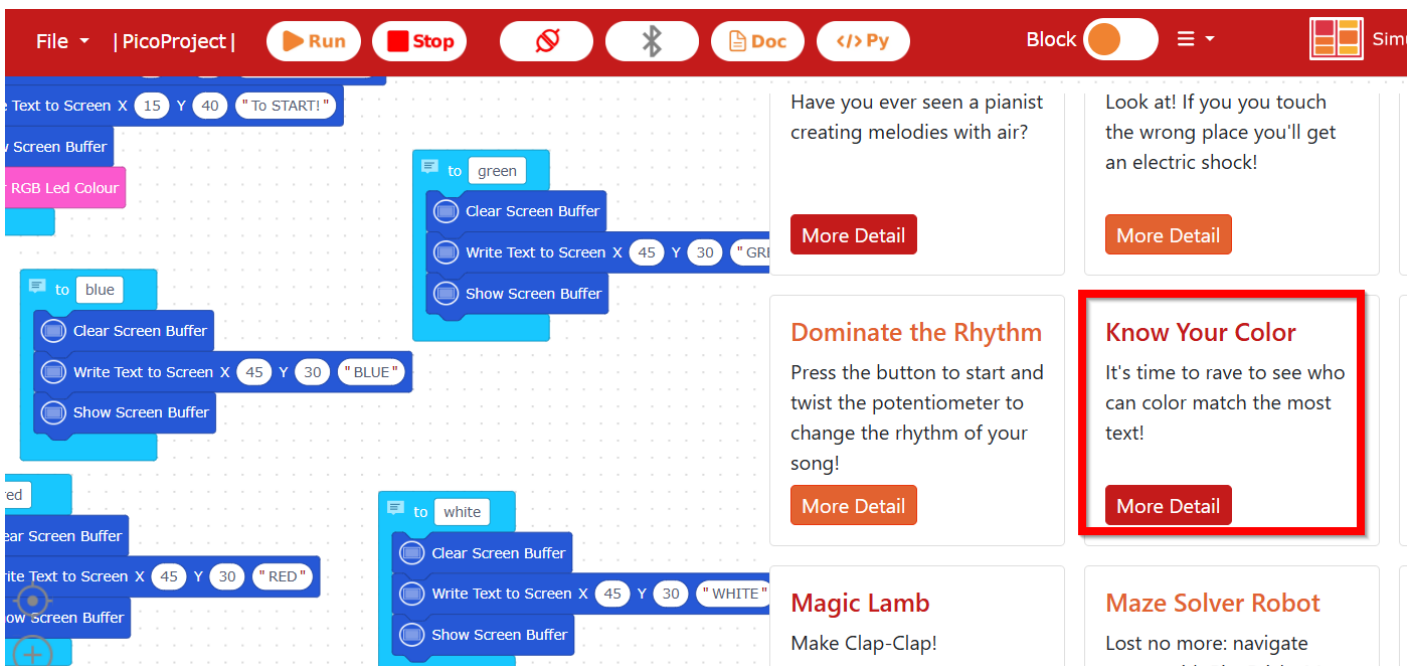
# PROYECTO Know Your Color

Extraído 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>

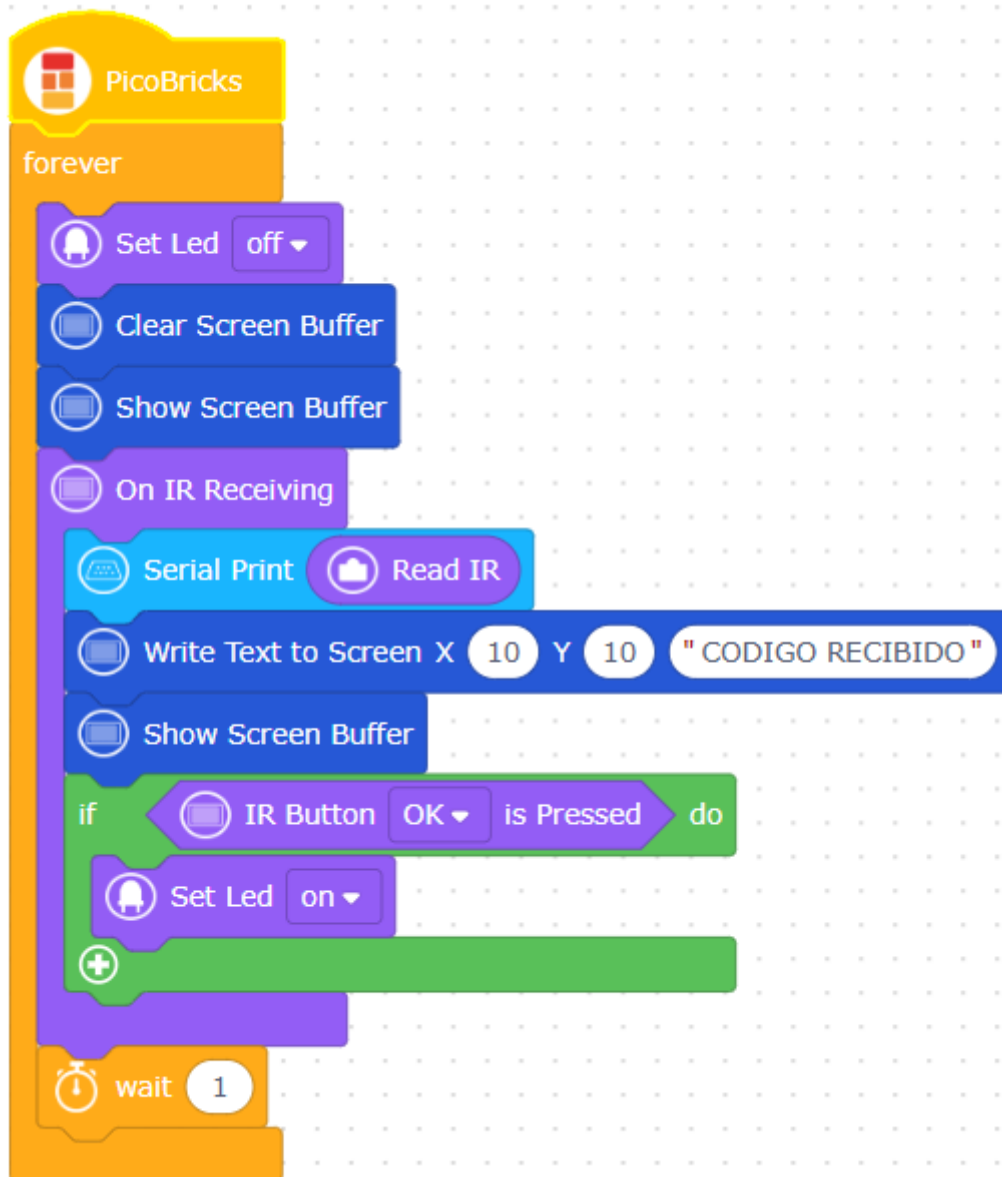
# Algo diferente PROYECTO IR

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:

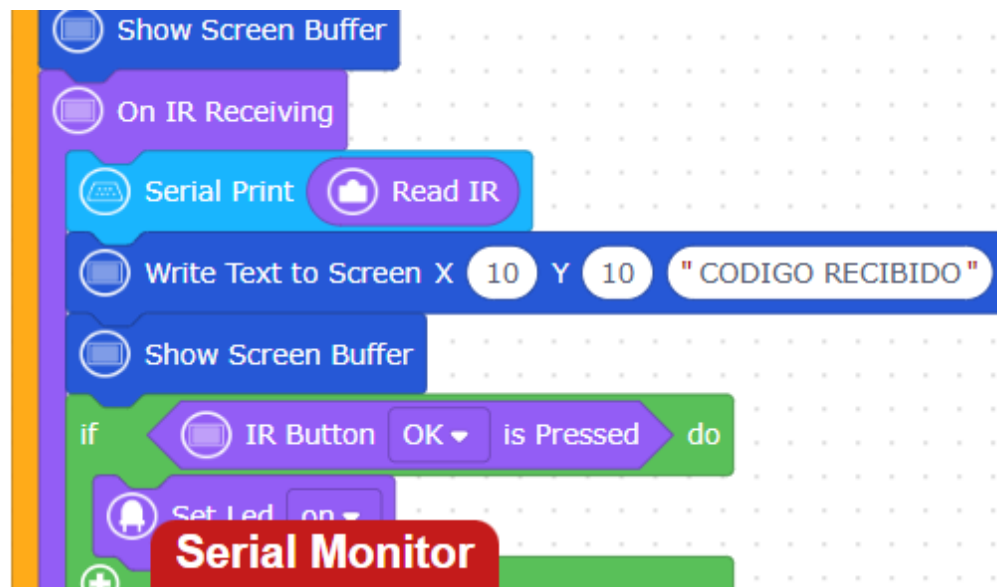
- si se aprieta un botón del mando IR que visualice por la pantalla OLED que ha recibido un código
- visualizará por el puerto serie el código recibido
- si la tecla es OK se encenderá el led rojo

## Solución



## Resultado

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



```
>> None
>> Data 15 Addr 0000
>> None
>> Data 1c Addr 0000
>> None
>> Data 44 Addr 0000
>> None
>> Data 46 Addr 0000
>> None
>> Data 0d Addr 0000
```

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>