JUAN JOSÉ LÓPEZ ALMENDROS

arduinoblocks.com

ÍNDICE

1.2 PLATAFORMA ARDUINOBLOCKS

ArduinoBlocks es una plataforma web online donde podemos programar nuestra placa Arduino de forma visual sin necesidad de conocer el lenguaje C++ que utiliza Arduino IDE.

La programación en ArduinoBlocks se realiza con bloques al estilo *AppInventor* o *Scratch*. No tenemos que escribir líneas de código y no nos permitirá unir bloques incompatibles evitando así posibles errores de sintaxis.

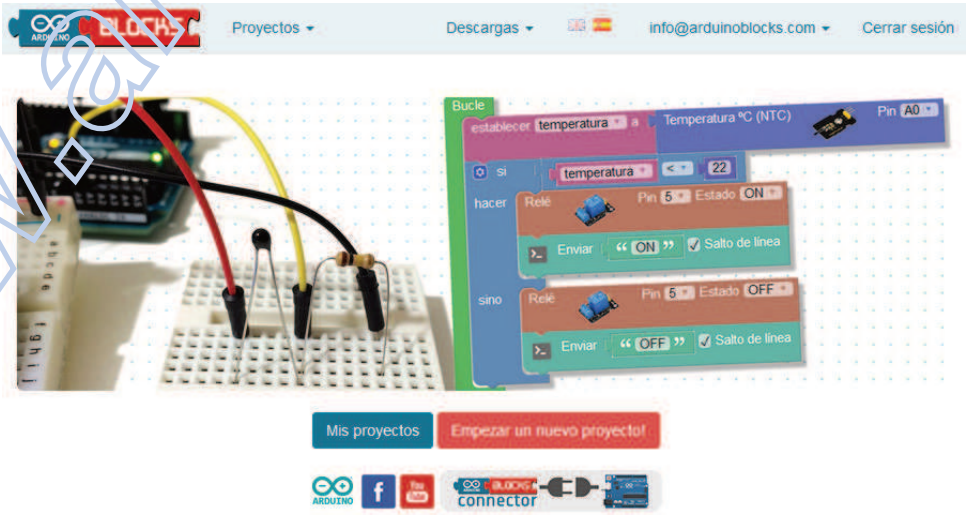
La plataforma ArduinoBlocks genera, compila y sube el programa a la placa Arduino por medio de la conexión USB. Una vez subido el programa, la placa Arduino no necesitará de la conexión al PC para funcionar pudiendo alimentarla con baterías o una fuente de alimentación para que funcione de forma autónoma.

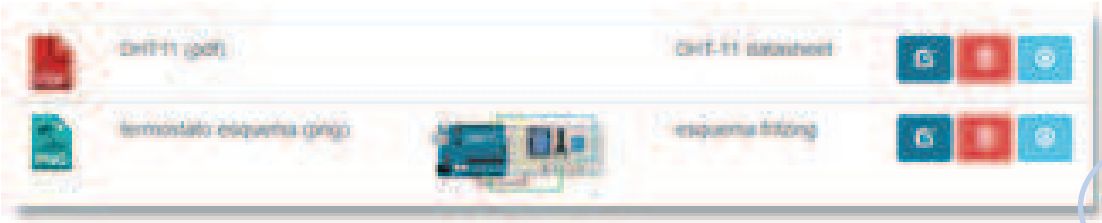
ArduinoBlocks actualmente funciona con todos los navegadores de última generación: Mozilla Firefox, Google Chrome, Opera, Safari, etc.

Registrándonos como usuarios de la plataforma ArduinoBlocks podemos aprovechar todas estas posibilidades:

- Guardar tus proyectos en la nube de ArduinoBlocks.
- Añadir información al proyecto: descripción, componentes utilizados, imágenes, etc.
- Añadir archivos adjuntos relacionados con el proyecto: esquemas, fotos, archivos para impresión 3D, aplicaciones, etc.
- Compartir proyectos con el resto del mundo.
- Importar proyectos compartidos por otros usuarios.
- Valorar y comentar proyectos
- Programar directamente Arduino desde el propio navegador (Con la aplicación: *ArduinoBlocks-Connector, ver apdo. 1.3*)
- Utilizar la consola serie desde el propio navegador

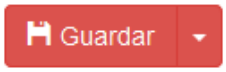
www.arduinoblocks.com

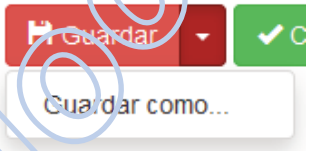




- **Guardar**

  ArduinoBlocks guarda automáticamente el proyecto cada cierto tiempo. En caso de querer asegurarnos el guardado podemos pulsar el botón "*Guardar*".
  También podemos crear un nuevo proyecto a partir del actual pulsando la opción "*Guardar como*". Automáticamente se abrirá el nuevo proyecto creado a partir del primero.





- **Barra de información**

  En la parte inferior derecha podemos obtener la información de guardado y algunos avisos que nos muestra la aplicación.

- **Estructura de un nuevo proyecto:** Un proyecto Arduino tiene siempre dos estructuras importantes en su interior, esto se ve reflejado claramente al crear un nuevo proyecto en ArduinoBlocks.

  1. **Bloque "*inicializar*" o "*setup*":**

El contenido de <u>este bloque sólo se una vez durante el inicio</u> del microcontrolador de Arduino (o si pulsamos el reset y la placa Arduino se reinicia).
Este bloque se utiliza para inicializar variables, configurar sensores, actuadores o periféricos, etc.

  2. **Bloque "*bucle*" o "*loop*":**

El contenido de <u>este bloque se repite indefinidamente</u>. Dentro de este bloque añadiremos los bloques de nuestro programa con la funcionalidad deseada.

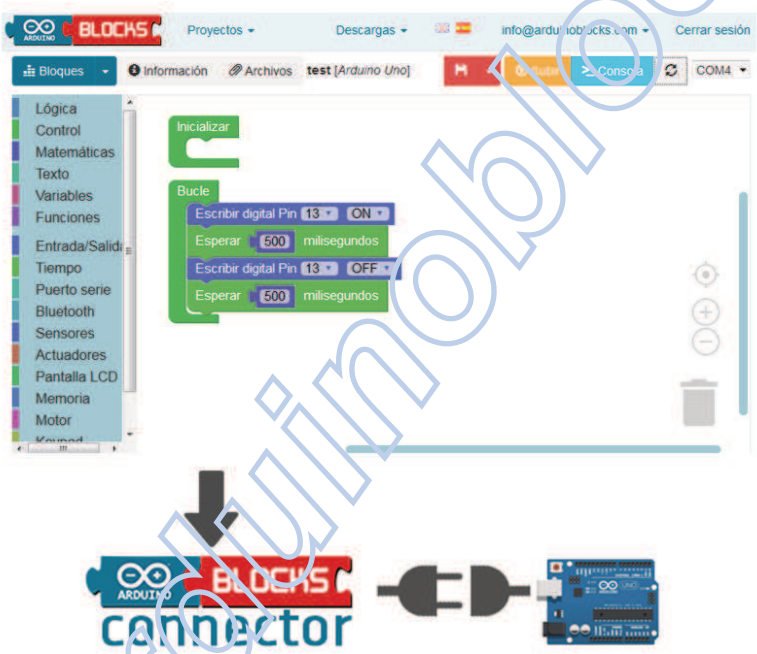
1.3 ARDUINOBLOCKS-CONNECTOR

ArduinoBlocks-Connector es una aplicación nativa que hace de puente entre la plataforma on-line ArduinoBlocks y el hardware Arduino.

La aplicación *ArduinoBlocks-Connector* se encarga de recibir el código generado por ArduinoBlocks, compilarlo y subirlo a la placa Arduino, sin esta aplicación ArduinoBlocks funciona pero no puede subir el programa a la placa Arduino pues el navegador web no dispone de posibilidad de realizar estas funciones por sí sólo.

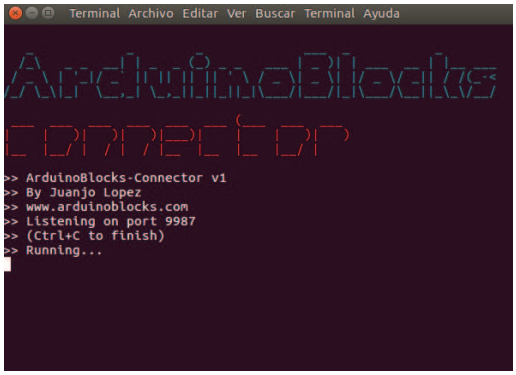
ArduinoBlocks-Connector está disponible para los principales sistemas operativos. Accede al área de descargas de arduinoblocks.com para obtener la última versión y más información sobre el proceso de instalación y configuración.

http://www.arduinoblocks.com/web/site/abconnector



ArduinoBlocks-Connector-1 ejecutándose bajo Windows

ArduinoBlocks-Connector-1 ejecutándose bajo Ubuntu



3.2 BLOQUES DE USO GENERAL

Los bloques de uso general nos permiten implementar funciones comunes en cualquier entorno o sistema programable. Esto incluye funciones lógicas, matemáticas, condiciones, bucles, funciones de texto, etc.

3.2.1 LÓGICA

Con estos bloques tenemos acceso a las funciones lógicas necesarias para implementar en nuestro programa de Arduino.

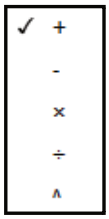
Las funciones lógicas trabajan con valores o expresiones de "verdadero" o "falso"

- **Condición / decisión**: Evalúa una condición lógica, si se cumple realiza el bloque "hacer" si no se cumple realiza el bloque "sino" (opcional)

Ejemplo:

- **Ángulo**: Permite definir un valor de ángulo en grados. Es un valor numérico tal cual, pero con la ventaja que permite definir el valor de una forma visual viendo el ángulo gráficamente.

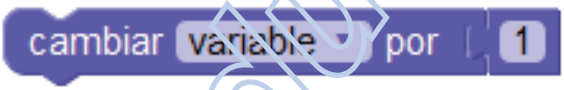
- **Operaciones básicas:**



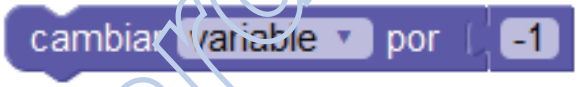
+	Suma
-	Resta
x	Multiplicación
÷	División
^	Potencia

Ejemplo:

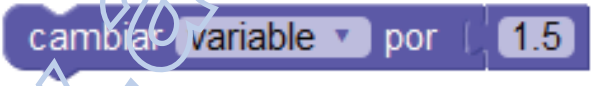
- **Cambiar variable**: Aumenta o disminuye el valor de una variable por el valor indicado (si es un valor positivo aumenta si es negativo disminuye)



Aumenta la variable en +1
variable = variable + 1

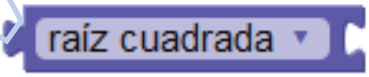


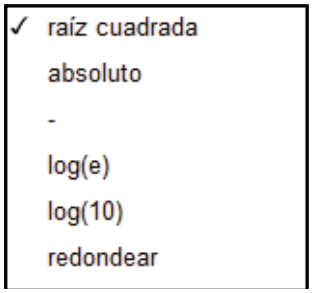
Disminuye la variable en -1
variable = variable – 1



Aumenta la variable en +1.5
variable = variable + 1.5

- **Funciones matemáticas:**





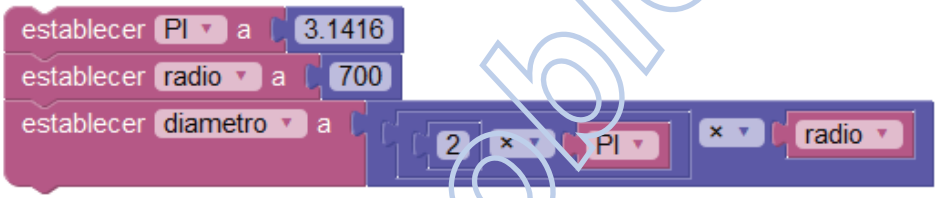
3.2.5 VARIABLES

Una variable es un hueco en la memoria donde el programa puede almacenar valores numéricos. El sistema nos permiten asignarles un nombre simbólico como por ejemplo "temperatura exterior", "velocidad", "posición servo 1",... para facilitar su uso.

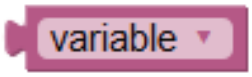
Las variables internamente usan un valor decimal de 32 bits (4 bytes) que permite almacenar valores en el rango: -3.4028235E+38 a 3.4028235E+38

- **Establecer variable**: Permite fijar el valor de una variable, es decir, guardar un valor en el hueco de memoria al que hace referencia la variable.

Ejemplo:



- **Obtener valor:** Obtiene el valor almacenado dentro de la variable.



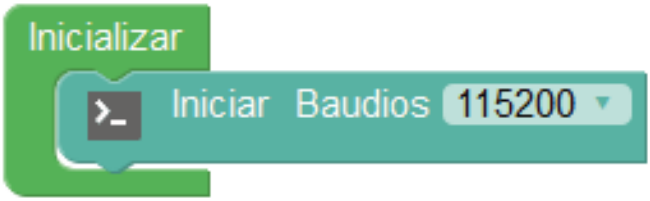
Ejemplo:

3.2.6 FUNCIONES

Las funciones permiten agrupar bloques de código. Esto es útil cuando un bloque de código se repite en varias partes del programa y así evitamos escribirlo varias veces o cuando queremos dividir el código de nuestro programa en bloques funcionales para realizar un programa más entendible.

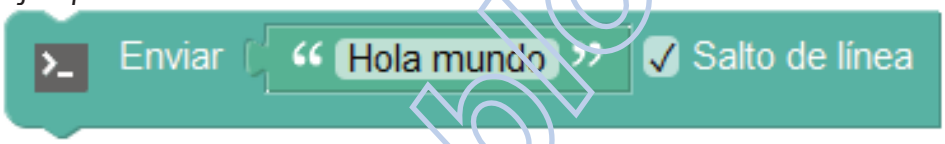
- **Definición de una función**: La definición consiste en crear el grupo donde podremos insertar el código de bloques que forma la función. Debemos darle un nombre representativo que utilizaremos para llamar a esa función y ejecutarla.

Ejemplo:



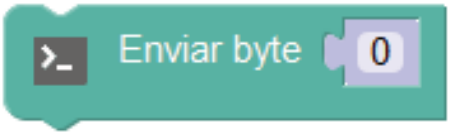
- **Enviar**: Escribe un valor de texto o el valor de una variable en el puerto serie. La opción "Salto de línea" permite añadir o no un retorno de carro al final del envío para bajar de línea.

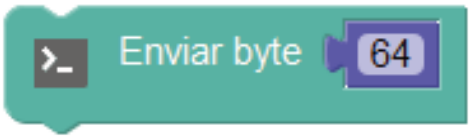
Ejemplo:



- **Enviar byte**: Envía un valor numérico como un byte (8 bits). Por tanto el valor debe estar comprendido entre 0 y 255.

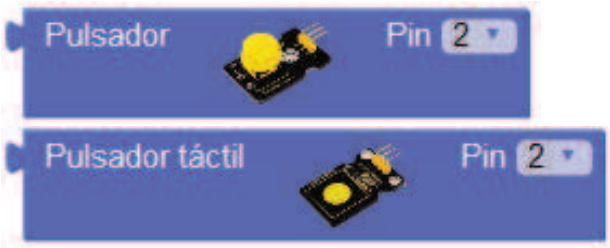
Ejemplo: Enviar byte con valor 64





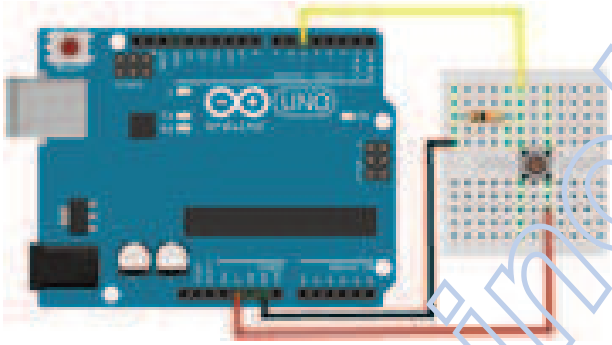
- **Sensor pulsador/pulsador táctil:** Botón para interactuar de forma táctil.

Tipo: Digital *Pin: 2-13* *Valor: 0/1 (F/V, Off/On)*

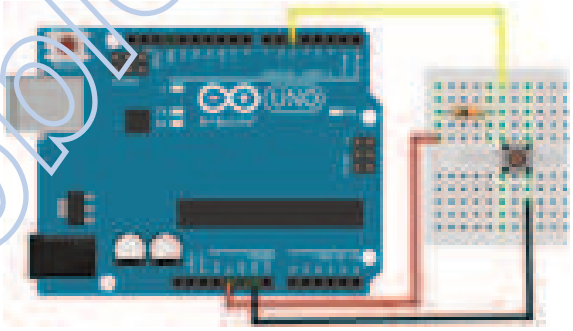


Dependiendo de la conexión que hagamos del pulsador, o en caso de utilizar módulos de pulsador de diferentes fabricantes, la lógica de funcionamiento del pulsador puede ser diferente:

Conexión:
sin presionar "off" / presionado: "on"



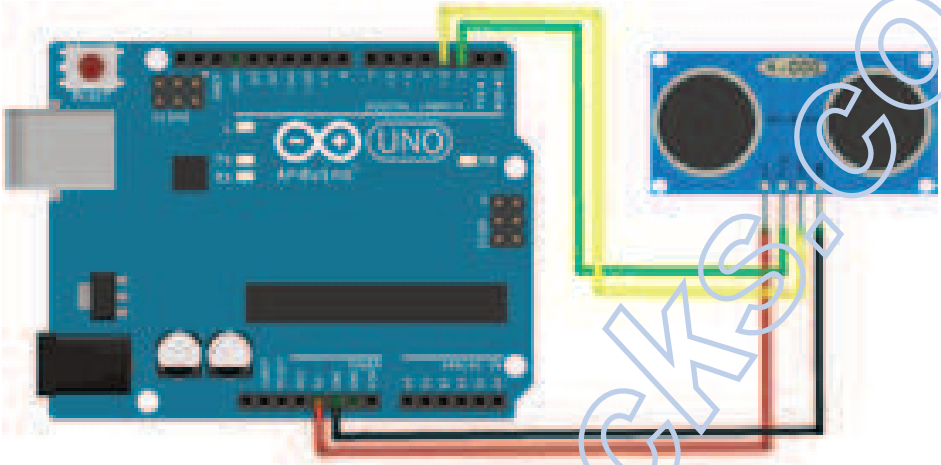
Conexión:
sin presionar "on" / presionado: "off"



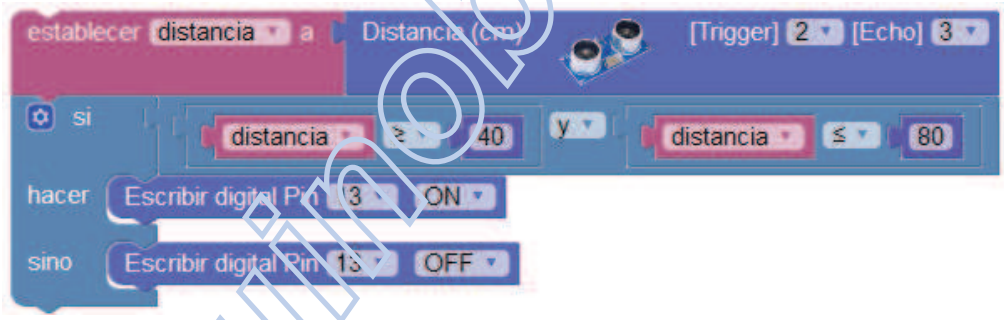
Alguno módulos de pulsador internamente trabajan de forma inversa por su conexión interna. En ese caso el pulsador siempre está dado una señal "On" y cuando lo pulsamos genera la señal "Off". En ese caso podemos invertir la condición para detectar cuando está pulsado:

- **Sensor de movimiento (PIR):** Se activa cuando detecta movimiento a su alrededor, a partir de un tiempo sin detección el sensor vuelve a desactivarse.

Tipo: Digital *Pin: 2-13* *Valor: 0/1 (F/V, Off/On)*



Ejemplo: Activación del led en el pin 13 cuando se detecta un objeto entre 40 y 80 cm de distancia



- **Sensor receptor de infrarrojos:** Permite decodificar los protocolos de señales de pulsos infrarrojos utilizados por los mandos a distancia.

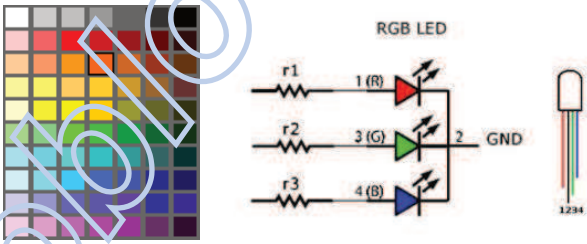
  Protocolos detectados: RC5, RC6, NEC, SONY, PANASONIC,JVC, SAMSUNG, WHYNTER, AIWA, LG, SANYO, MITSUBISHI, DENON.

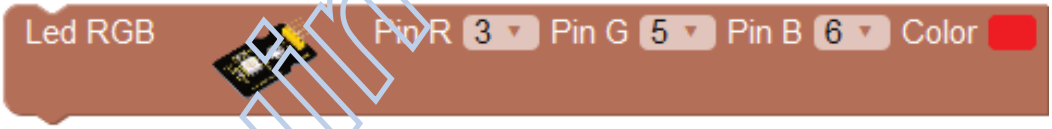
  Tipo: Datos
  Pin: 11
  Valor: código recibido / 0 = ningún código detectado.

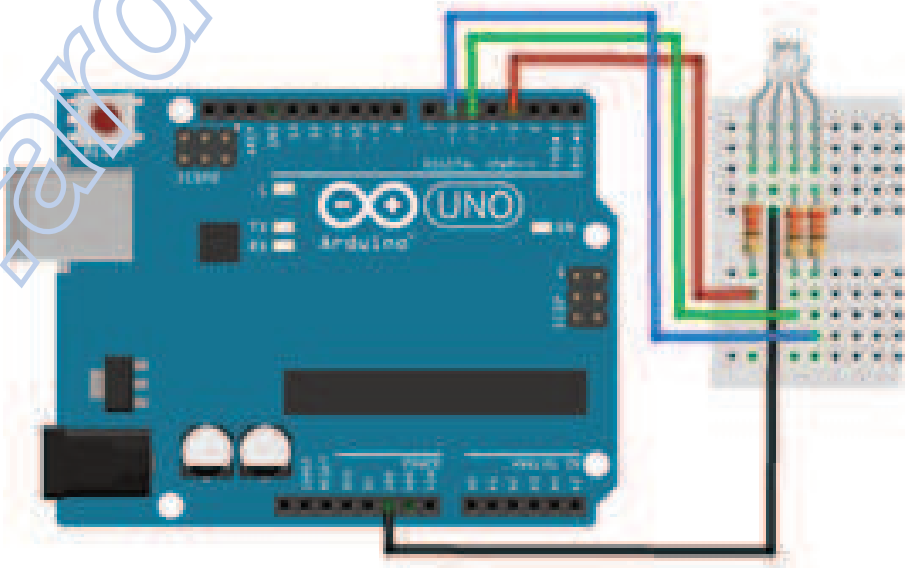
Ejemplo: Aumento progresivo de la intensidad del led:

- **Led RGB:** Controla un led RGB. Define un color calculando automáticamente los valores de cada componente R,G y B para definir el color.

Tipo: PWM
Pin R/G/B: 3,5,6,9,10,11
Valor: Color

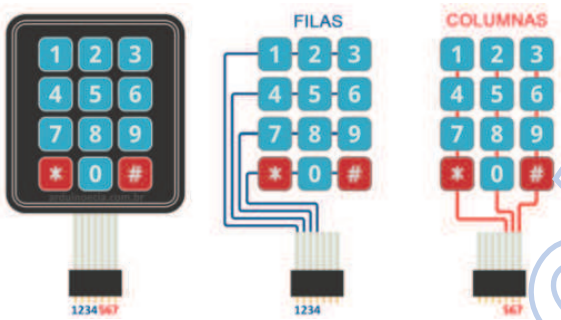






3.3.10 KEYPAD

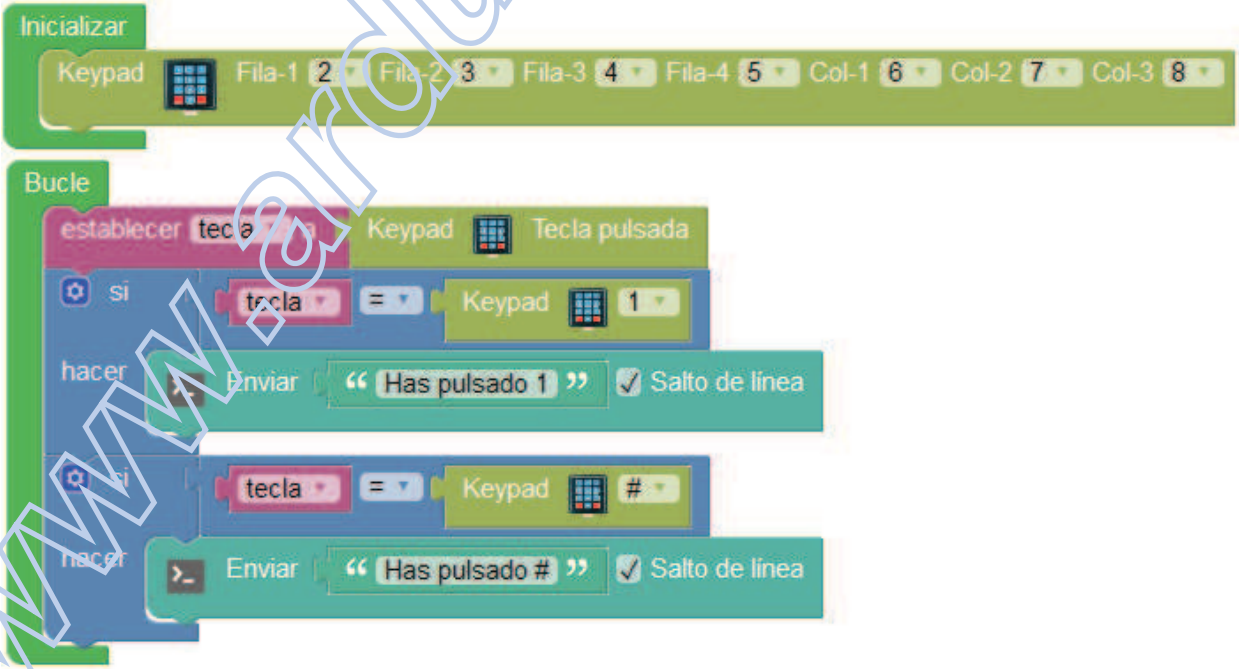
El teclado o "keypad" nos permite de una forma sencilla añadir un pequeño teclado numérico a nuestro proyecto. Se basa en una botonera conectada de forma matricial por filas y columnas. ArduinoBlocks gestiona automáticamente la detección de filas y columnas activadas para detectar la tecla pulsada.



- **Configuración del keypad:** define los pines de conexión para las filas y columnas del keypad.

- **Tecla pulsada:** obtiene la tecla pulsada actualmente en el keypad.

Ejemplo: Detección de las teclas '1' y '#'



4 PROYECTOS

A continuación se detallan 25 proyectos desarrollados en ArduinoBlocks con esquemas y programas de bloques.

La funcionalidad de cada proyecto está bastante simplificada. El objetivo es mostrar las posibilidades de la plataforma con aplicaciones reales sencillas.

En la web de ArduinoBlocks podemos buscar proyectos realizados por otros usuarios (proyectos compartidos) que nos sirvan también como inspiración o punto de partida para nuestros propios proyectos.

Listado de proyectos resueltos:

P01.-Secuenciador de luces
P02.-Simulación amanecer y anochecer
P03.-Lámpara con regulación manual
P04.-Semáforo
P05.-Timbre
P06.-Control inteligente de iluminación
P07.-Encendido automático por movimiento
P08.-Contador manual
P09.-Cronómetro
P10.-Fotómetro
P11.-Iluminación crepuscular
P12.-Encendido y apagado con palmada
P13.-Termómetro
P14.-Termostato
P15.-Medidor de distancia
P16.-Riego automático
P17.-Lámpara multicolor con control IR
P18.-Piano con teclado
P19.-Sensor de aparcamiento
P20.-Control pan/tilt con joystick
P21.-Control de un led desde PC
P22.-Control de relés por Bluetooth
P23.-Estación meteorológica
P24.-Control de led por voz
P25.-Control domótico

P02 - SIMULACIÓN DE AMANECER Y ANOCHECER

Con este proyecto vamos a simular el ciclo solar de anochecer y amanecer donde la luz disminuye o aumenta progresivamente de forma suave.

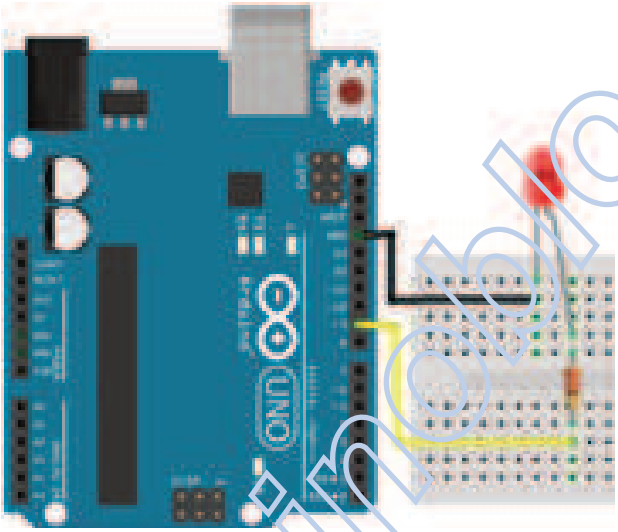
Aplicaciones de ejemplo:

-Belén Navideño con simulación de día/noche

-Aviario para cría de aves

Material necesario:

- 1 x led
- 1 x resistencia de 220 Ω.
- Placa de prototipos, cables de interconexión.



Conexiones:
Led = Pin ~9

P12 - ENCENDIDO / APAGADO CON PALMADA

Este montaje nos permitirá sorprender a nuestros invitados en casa. Con un sonido intenso como el de una palmada podemos encender y apagar la luz de nuestra habitación.

Este sencillo sistema nos permite controlar la luz sin movernos del sofá. Como habrás comprobado no sólo sirve una palmada, cualquier sonido que supere el umbral configurado activará el sistema (la palmada nunca falla).

Material necesario:

- 1 x módulo de sensor de sonido
- 1 x potenciómetro rotativo 10kΩ
- 1 x led
- 1 x resistencia 220Ω
- 1 x módulo de relé
- Placa de prototipos, cables de interconexión

Conexiones:
Sensor de sonido = Pin A0
Potenciómetro = Pin A1
Led = Pin 6
Relé = Pin 7