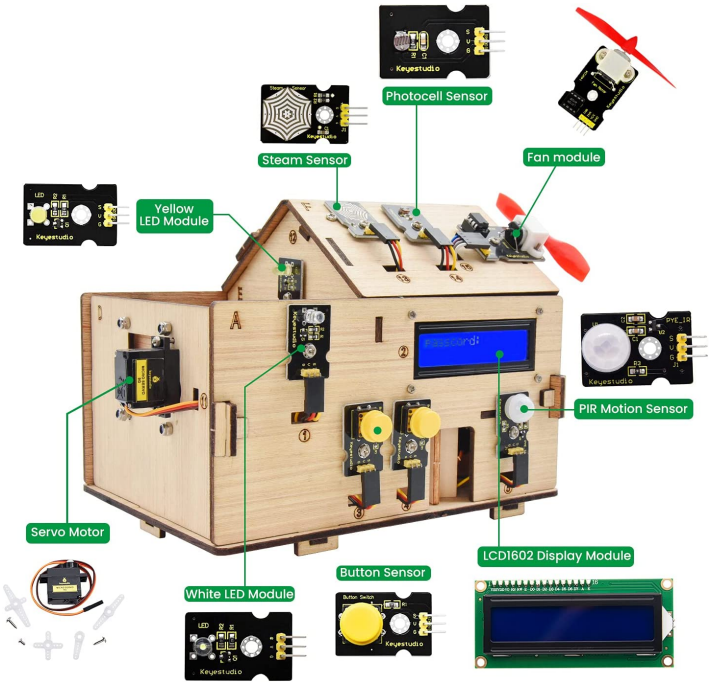
Maleta de la Innovación 4.0

SmartHome Kit

+Componentes extra



UNIVERSITAT POLITÈCNICA DE VALÈNCIA
CAMPUS D'ALCOI

ArduinoBlocks

Versión del documento: 2

Juanjo López
juanjose.lopez@salesianos.edu
arduinoblocks@gmail.com

Índice

El kit de la “Maleta de la Innovación 4.0” es un kit ofrecido por la Escuela Politécnica de Valencia, Campus de Alcoi dentro de la cátedra SmartCity a los centros educativos de Alcoy.

El kit se basa en el Smart Home Kit de keyestudio, un kit basado en Arduino con múltiples sensores, actuadores y periféricos además de las partes de la maqueta cortadas y listas para ensamblar.

https://shop.innovadidactic.com/es/standard-placas-shields-y-kits/1455-keyestudio-smart-home-para-arduino-con-placa-keyestudio-plus.html

El kit se ha complementado con sensores y módulos extra:

- Sensor de CO2
  https://shop.innovadidactic.com/es/standard-sensores/983-keyestudio-ccs811-sensor-de-eco2-dioxido-de-carbono-equivalente-y-tvoc.html

- Sensor de sonido
  https://shop.innovadidactic.com/es/standard-sensores/630-keyestudio-sensor-de-sonido-analogico-con-potenciometro.html

- Sensor y mando IR
  https://shop.innovadidactic.com/es/standard-sensores/668-keyestudio-kit-de-control-remoto-y-receptor-infrarrojo.html

- Módulo de relé doble
  https://shop.innovadidactic.com/es/standard-actuadores/649-keyestudio-modulo-de-rele-dual-o-dos-canales.html

- Sensor magnético (Hall)
  https://shop.innovadidactic.com/es/standard-sensores/615-keyestudio-sensor-de-campo-magnetico-hall.html

- Sensor DHT22 (temperatura y humedad)
  https://shop.innovadidactic.com/es/standard-sensores/1468-keyestudio-sensor-de-temperatura-y-humedad-dht22.html

- Placa solar USB
  https://shop.innovadidactic.com/es/cables/887-placa-solar-con-cable-usb-6v-3-5w-580ma.html

- Batería USB (tipo power bank)
  https://shop.innovadidactic.com/es/otros-steam-y-makers/1558-bateria-auxiliar-portatil-de-2200-mah.html

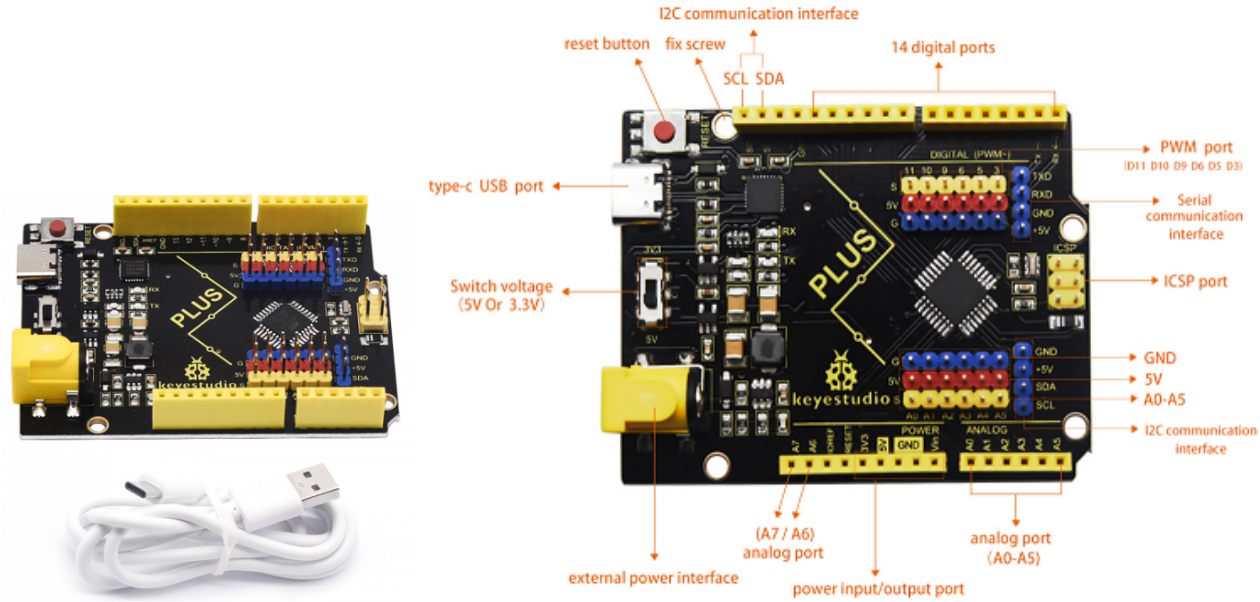
Material compatible con Arduino y ArduinoBlocks (Distribuidor oficial Keyestudio España)
https://shop.innovadidactic.com/es/

La programación del kit se realiza de forma gráfica mediante la plataforma ArduinoBlocks
http://www.arduinoblocks.com/

En esta guía manual se ha recopilado la información y prácticas realizadas durante la formación.

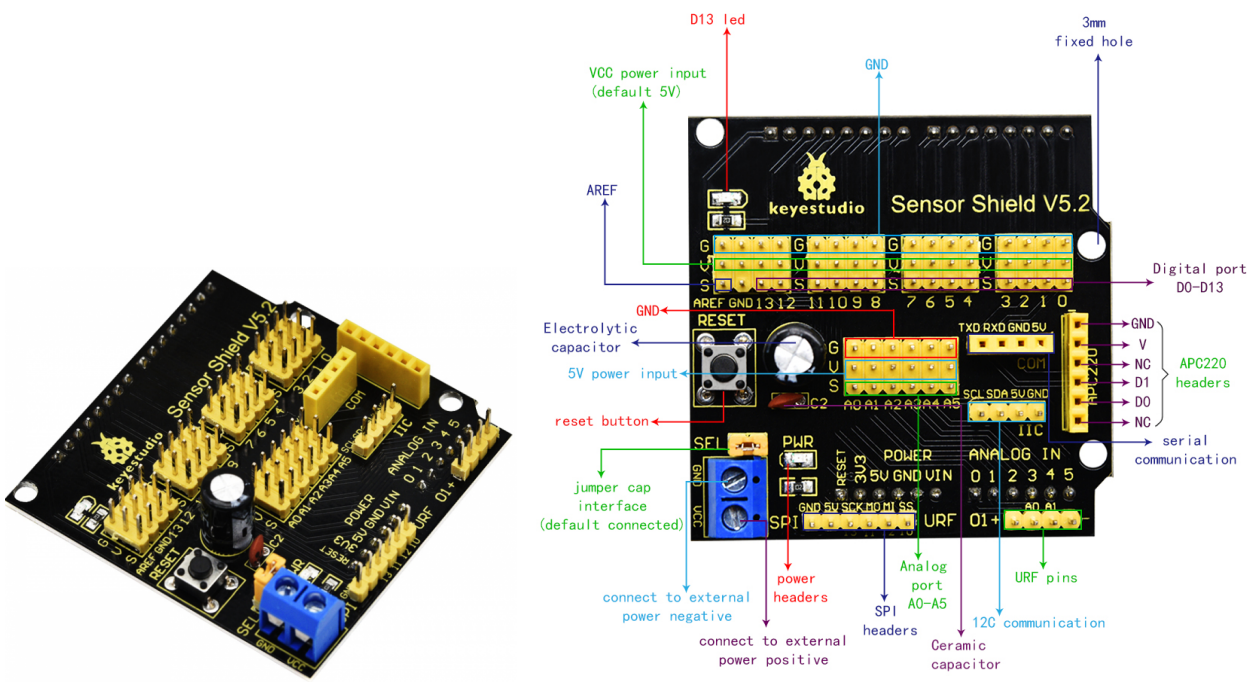
Arduino UNO PLUS + Sensor Shield 5

La placa Arduino PLUS incluida en el kit es un clon Arduino 100% compatible con algunas mejoras y además incorpora una “sensor shield” para facilitar la conectividad de forma modular.



https://wiki.keyestudio.com/KS0486_Keyestudio_PLUS_Development_Board_(Black_And_Eco-friendly)

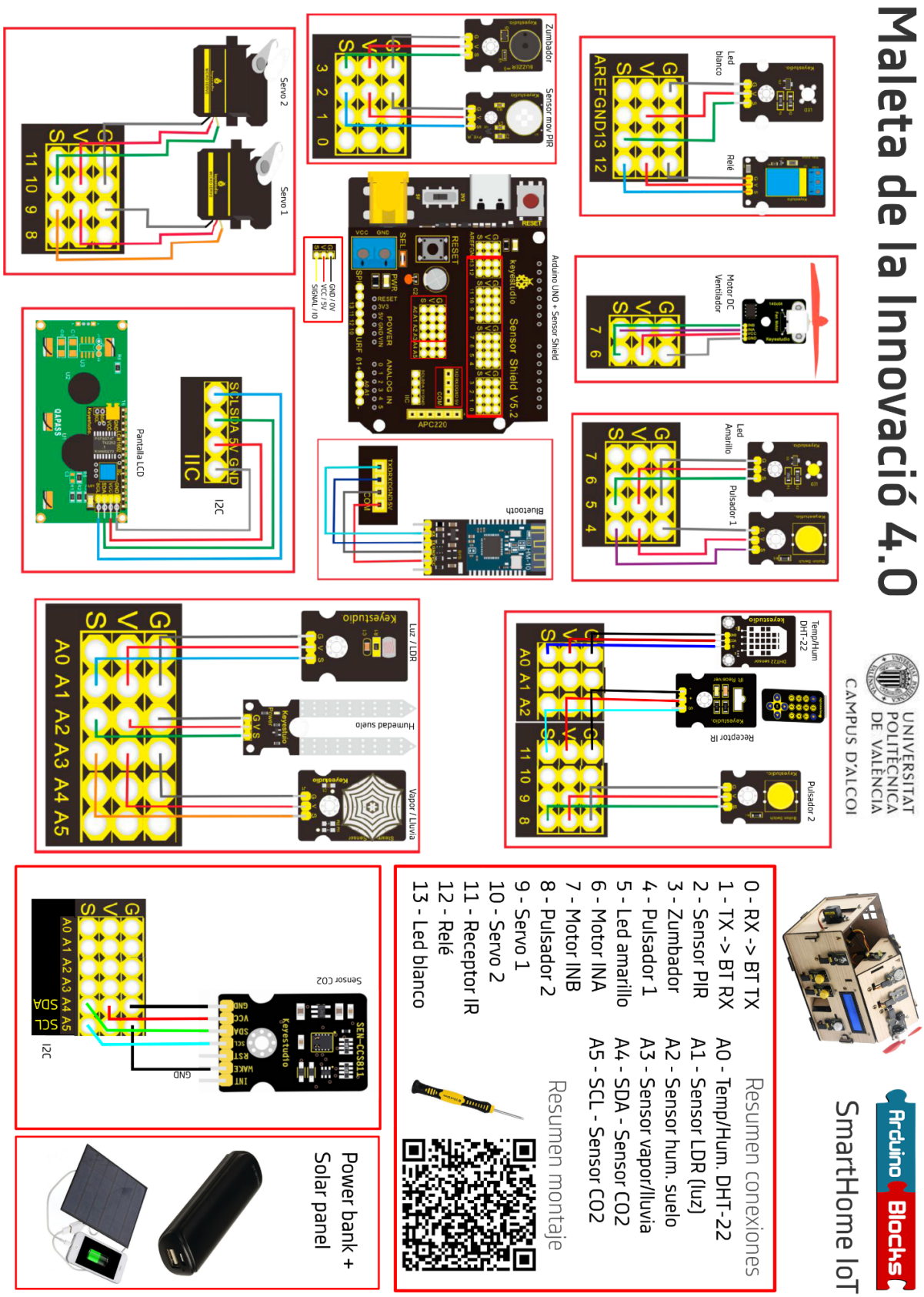
Sensor Shield 5



https://wiki.keyestudio.com/Ks0004_keyestudio_Sensor_Shield_V5

Conexiones

Esquema general de conexiones del kit completo *(utilizado en los ejemplos por separado también)*



Programación con ArduinoBlocks

ArduinoBlocks es un entorno visual de programación basado en bloques, desarrollado por Juanjo López. Es un entorno web, que además de permitir la creación de programas permite gestionar proyectos de forma integral añadiendo documentación, comentarios, adjuntos, etc.

ArduiniBlocks incorpora funcionalidades específicas para gestión de alumnos y proyectos por parte de los profesores. Además soporta las placas más utilizadas en los entornos educativos.

ArduinoBlocks cuenta con el apoyo de la empresa InnovaDidactic y garantiza la compatibilidad con sus productos de Keyestudio para Arduino (InnovaDidactic es el distribuidor oficial de keyestudio en España)

http://www.arduinoblocks.com/

Todos los colaboradores o entusiastas de ArduinoBlocks colaboran activamente realizando documentación y guías prácticas para docentes que están totalmente disponibles de forma gratuita:

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

El libro escrito por Juanjo López sobre programación con ArduinoBlocks, actualmente está disponible de forma totalmente libre para descarga en su versión “free book” online y se va actualizando continuamente.

https://www.amazon.es/ArduinoBlocks-edici%C3%B3n-Programaci%C3%B3n-Bloques-Arduino/dp/1977676588

https://docs.google.com/document/u/1/d/e/2PACX-1vQSrOKHpbLQHVbGFdAvp7DcndoftoHDI20nvwGMaxu_7bGc1bUCmi4U6DZrJWRSudc2iXBg43QMuzCT/pub

ArduinoBlocks-Connector v5

Para poder subir y compilar el programa desde la web de ArduinoBlocks debemos instalar la versión de ArduinoBlocks Connector v5 correspondiente para cada sistema operativo, además de asegurarnos de tener correctamente instalado el driver de Arduino en el sistema.

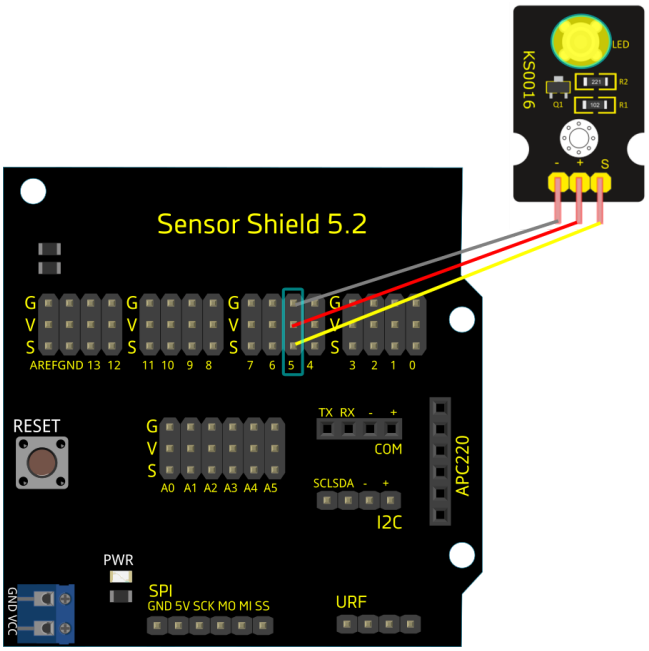
http://www.arduinoblocks.com/web/site/abconnector5

Parpadeo de un led

Módulo Led:

https://wiki.keyestudio.com/KS0016_Keyestudio_White_LED_Module

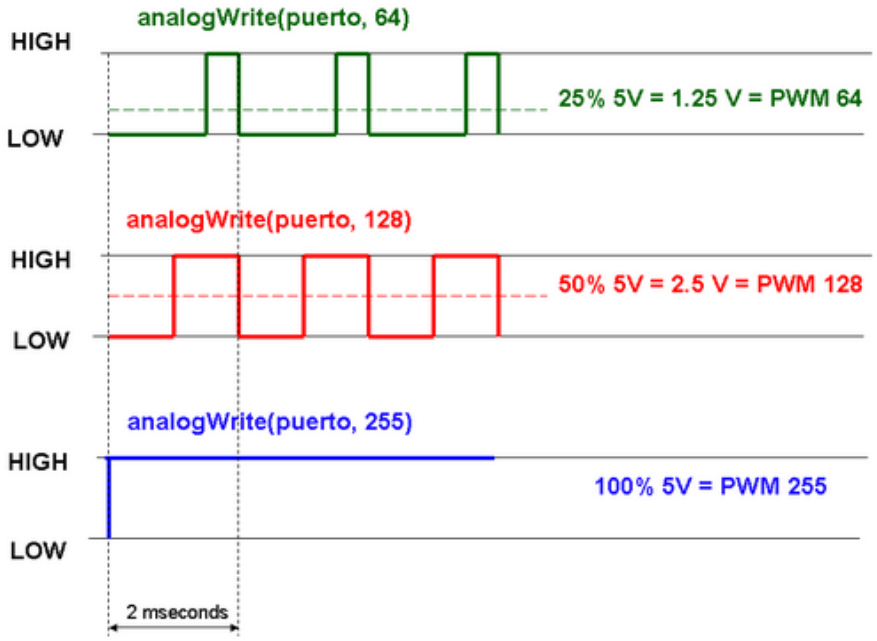
Esquema de conexiones:



Salidas PWM

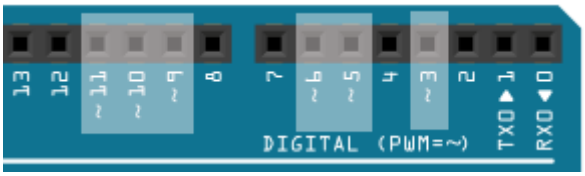
Arduino no tiene salidas puramente analógicas, pero podemos imitar a una salida analógica mediante la técnica PWM (Pulse Width Modulation = Modulación en Anchura de Pulso).

Ejemplo: Gráficas del funcionamiento del PWM:



La técnica PWM permite tener salidas “pseudo-analógicas” que podrán ser utilizadas para regular la intensidad que se aplica a un actuador, como por ejemplo un led (intensidad), motor (velocidad de giro), servo (posición), etc.

Los pines compatibles con PWM en Arduino UNO son:



3, 5, 6, 9, 10, 11

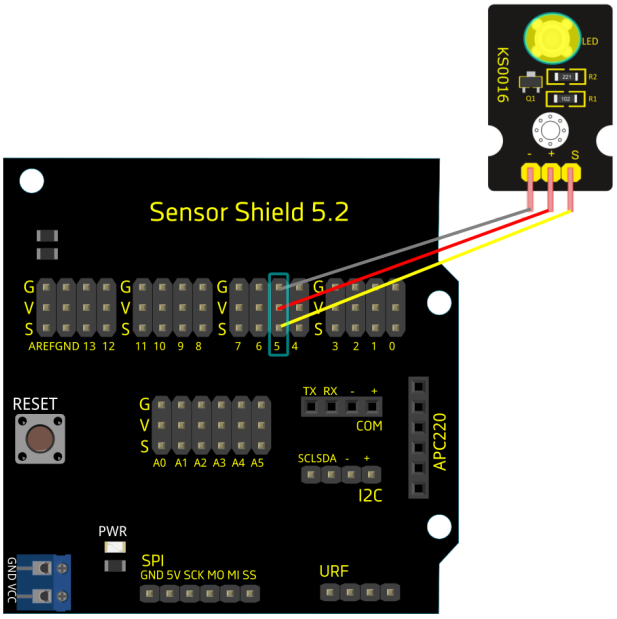
Y están indicados con el símbolo **~**

En ArduinoBlocks podemos escribir en salidas PWM el valor de 0 a 255 con el bloque:

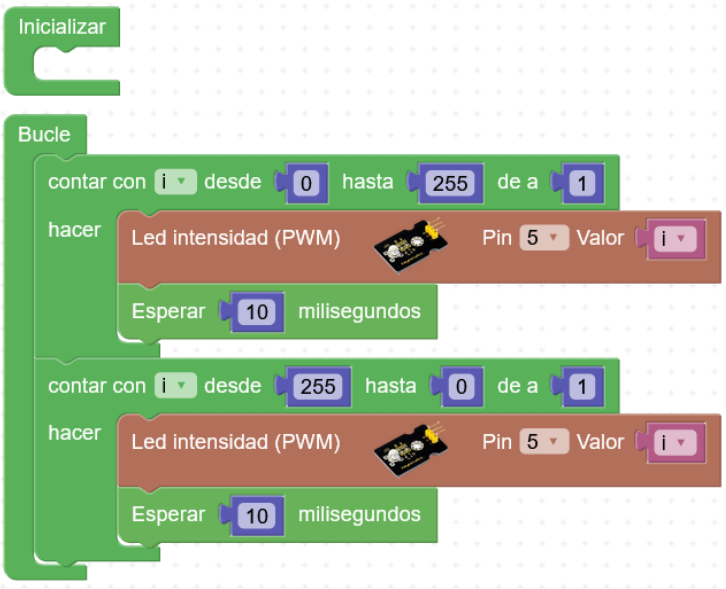
Regulación de intensidad de led

Módulo Led:

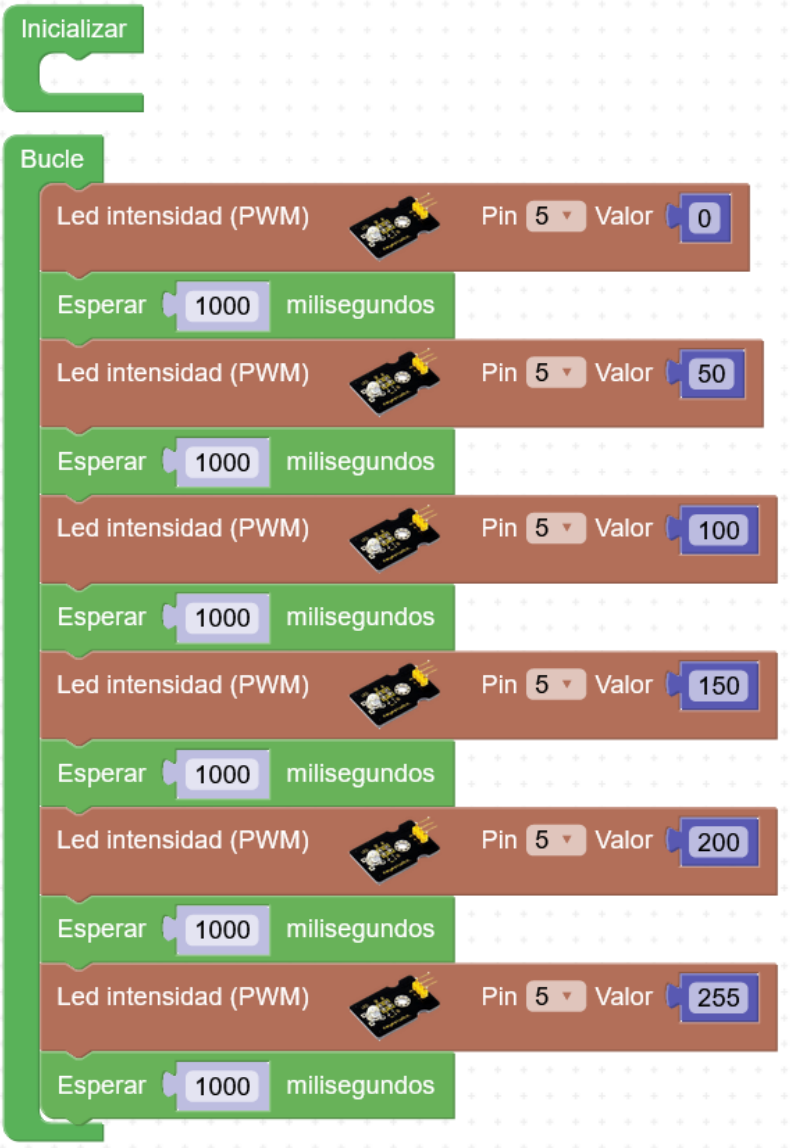
https://wiki.keyestudio.com/Ks0234_keyestudio_Yellow_LED_Module



Programa 1: aumento y disminución de intensidad progresiva del led



Programa 2: Aumento de intensidad en saltos



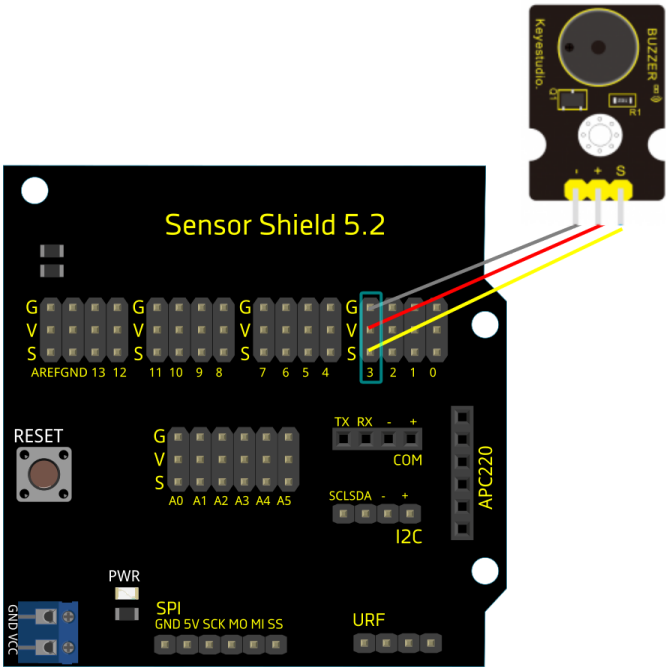
Escala musical

El zumbador pasivo permite emitir tonos en distintas frecuencias, de forma que podemos utilizarlo para emitir señales acústicas, o generar melodías.

Módulo zumbador:

https://wiki.keyestudio.com/Ks0019_keyestudio_Passive_Buzzer_module

Esquema de conexiones:

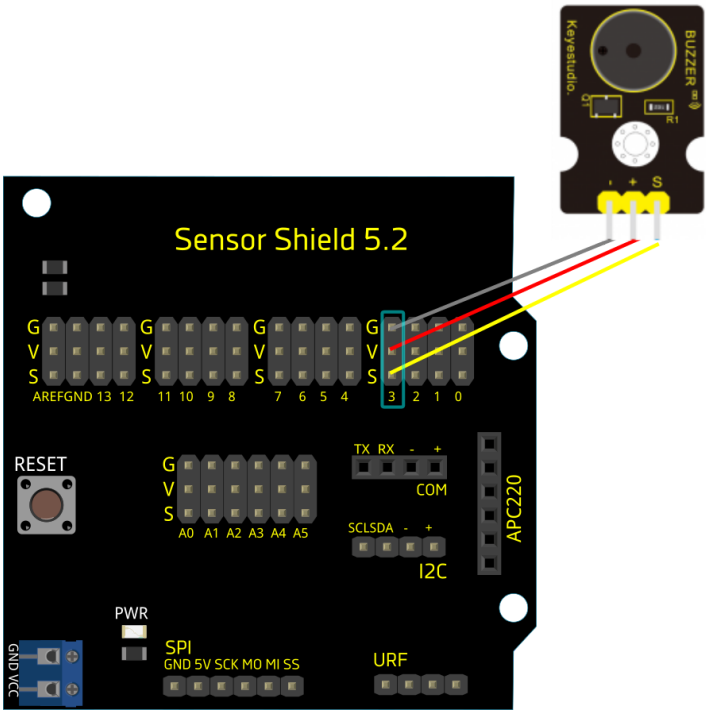


Melodía

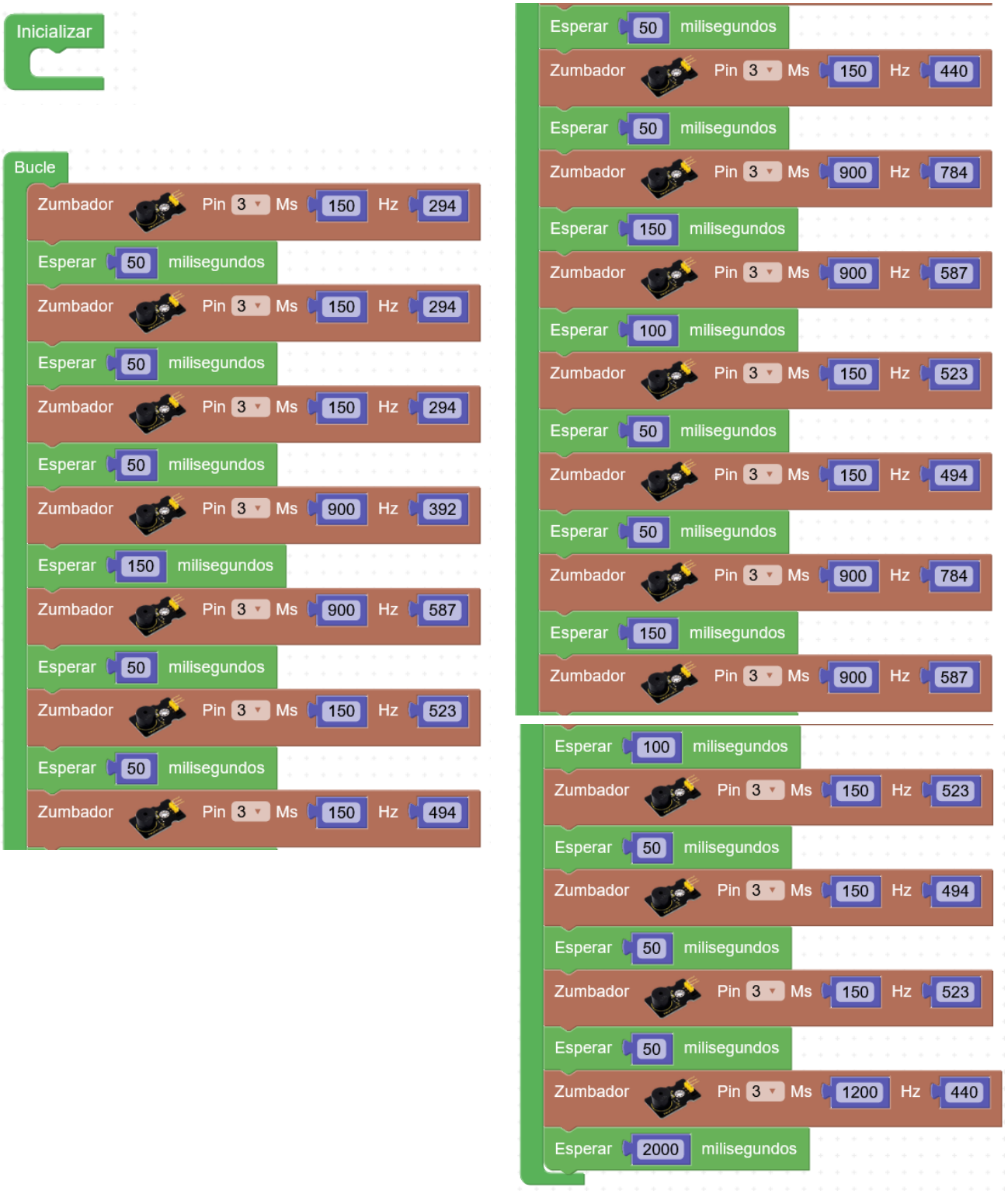
Módulo zumbador:

https://wiki.keyestudio.com/Ks0019_keyestudio_Passive_Buzzer_module

Esquema de conexiones:



Programa con melodía de ejemplo:

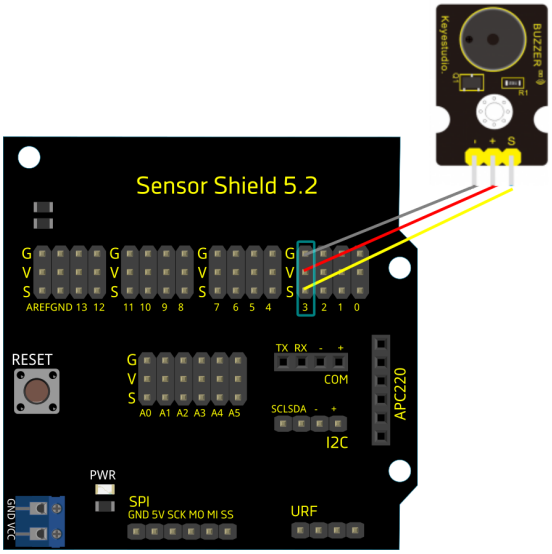


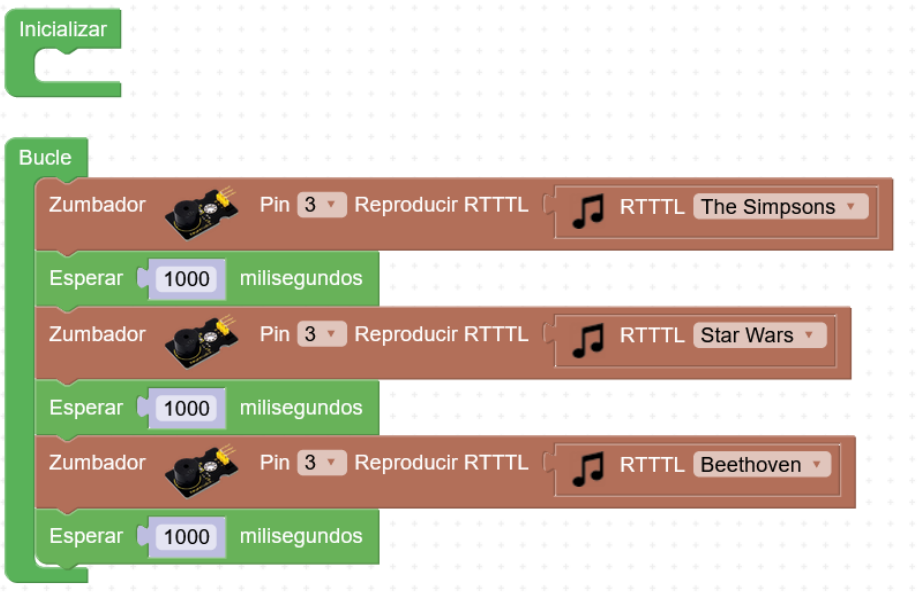
Melodías RTTTL

Módulo zumbador:

https://wiki.keyestudio.com/Ks0019_keyestudio_Passive_Buzzer_module

Esquema de conexiones:





Más información y melodías RTTTL:
http://www.arduinoblocks.com/web/help/rtttl

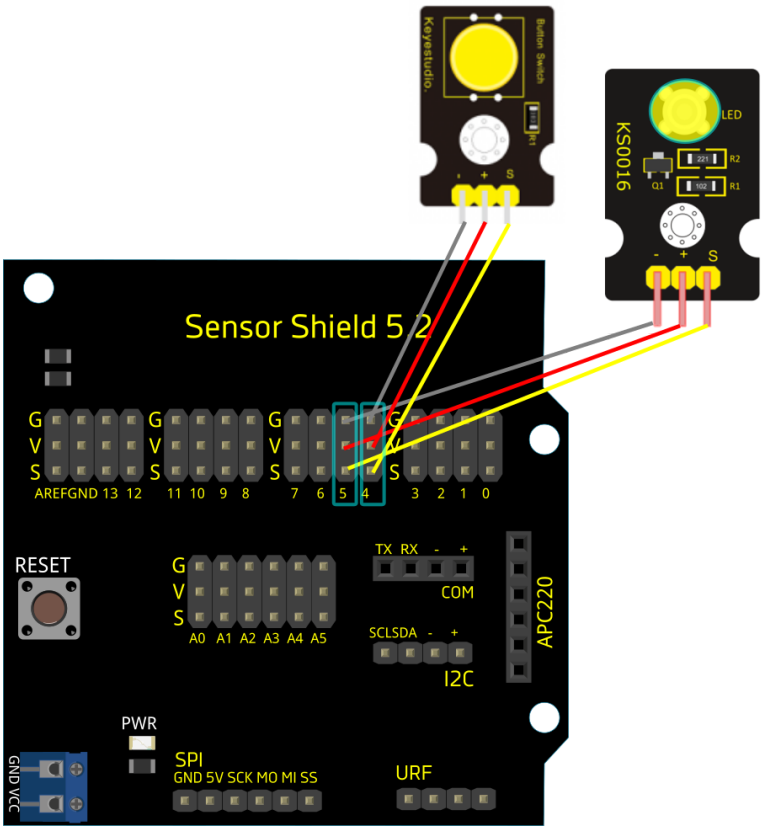
Led controlado por pulsador

Módulo pulsador:

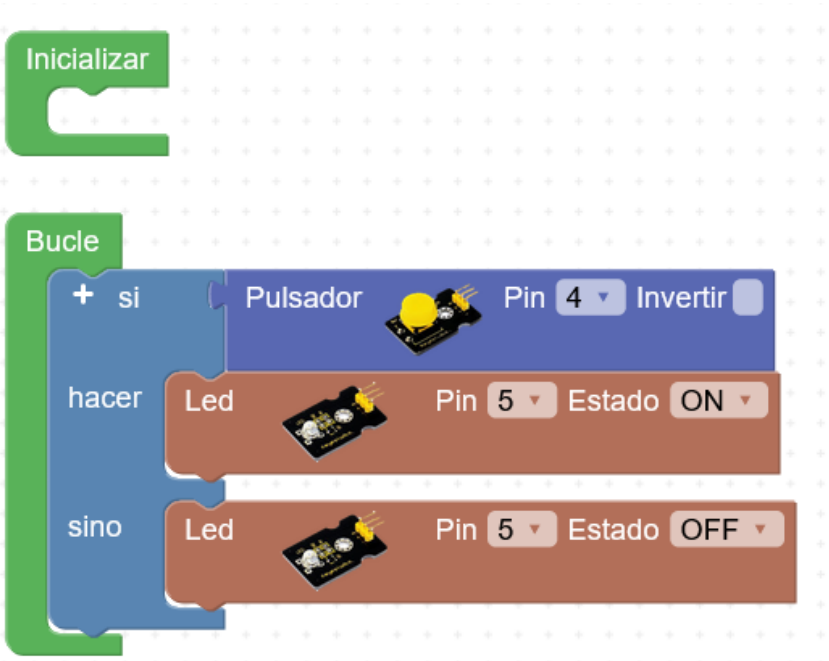
https://wiki.keyestudio.com/Ks0029_keyestudio_Digital_Push_Button

Módulo Led:

Esquema de conexiones:

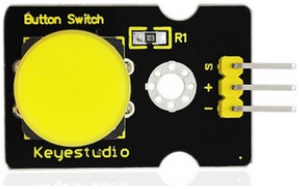


Programa de ejemplo:



Led controlado con pulsador (cambio de estado)

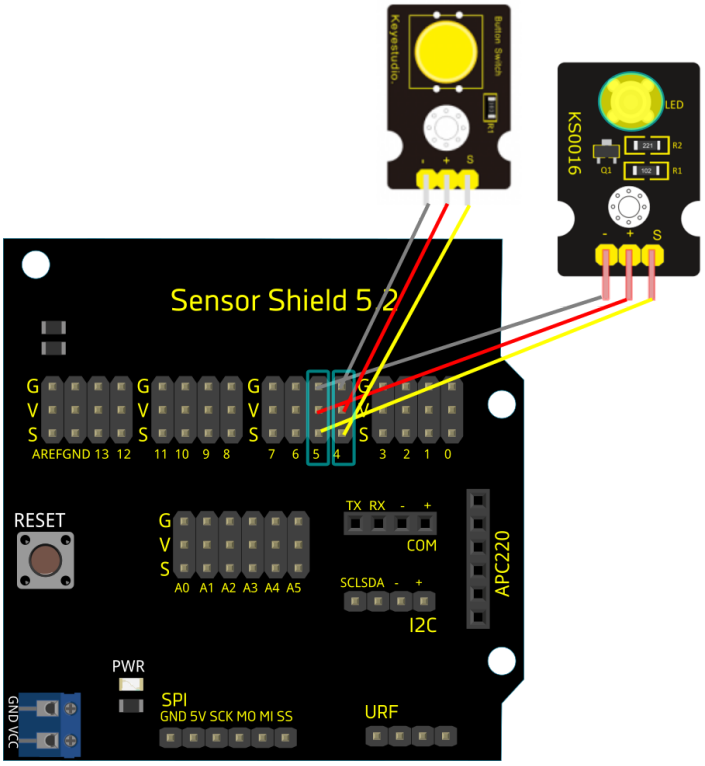
Módulo pulsador:



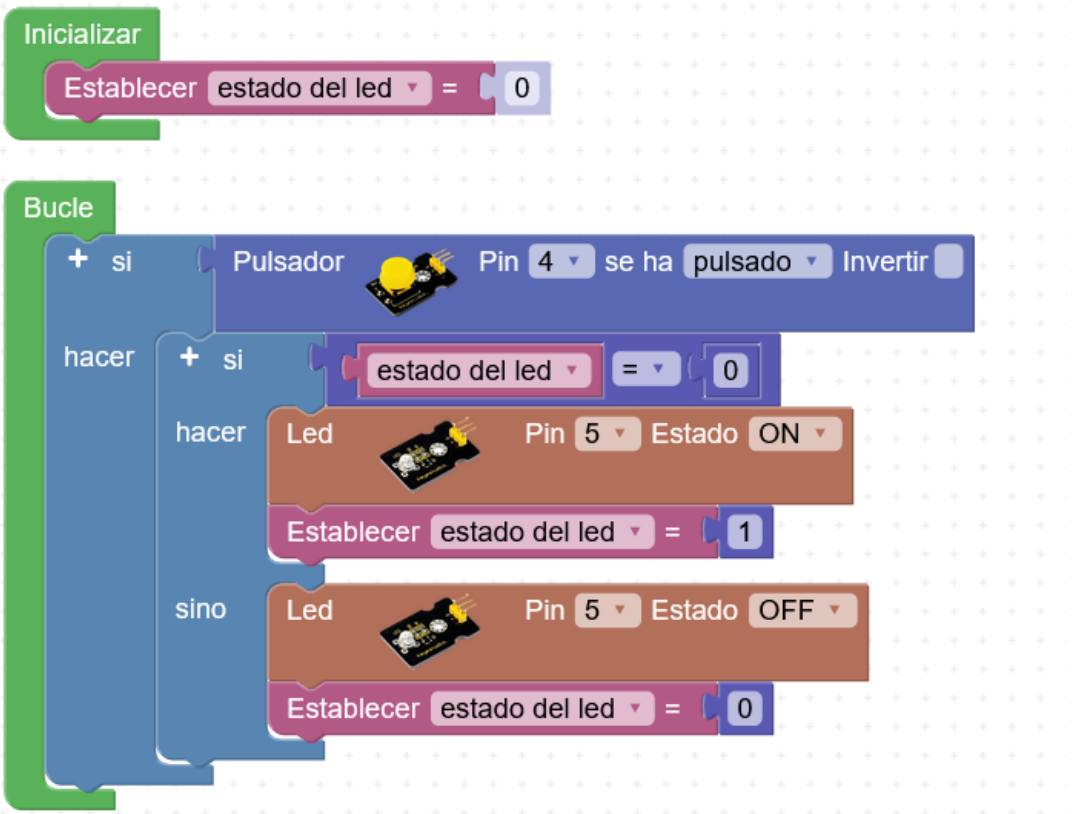
https://wiki.keyestudio.com/Ks0029_keyestudio_Digital_Push_Button

Módulo Led:

Esquema de conexiones:



Programa de ejemplo:



Detector de movimiento + Led

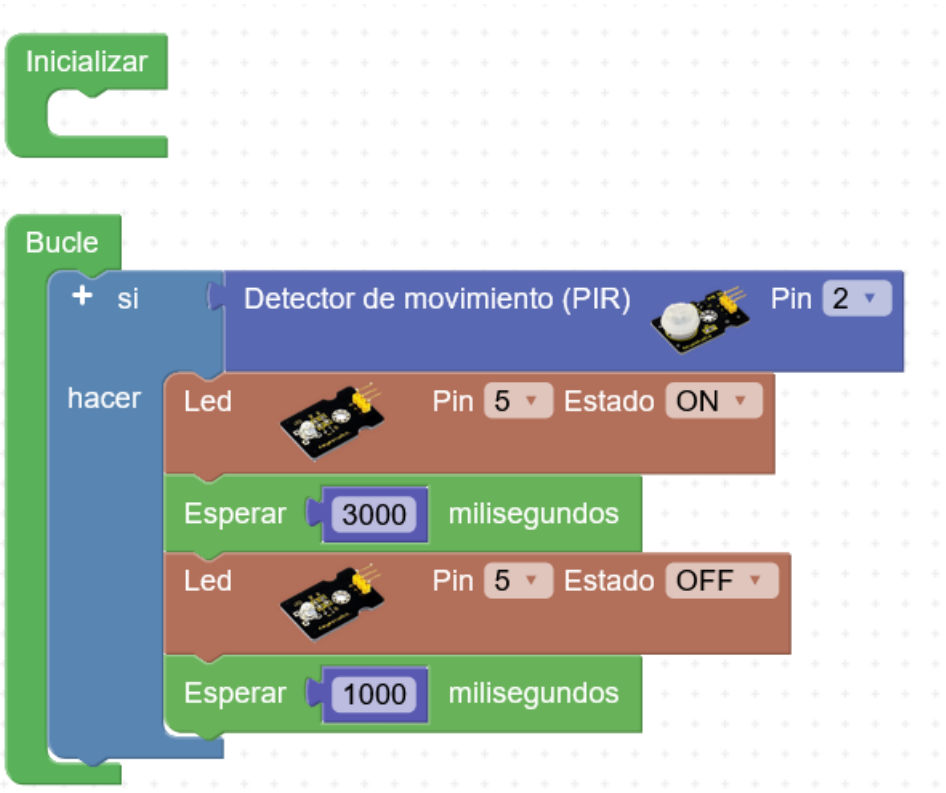
El sensor de movimiento/presencia PIR permite detectar movimiento de cuerpos que emitan radiación infrarrojo (seres vivos)

Módulo detector de movimiento PIR:

https://wiki.keyestudio.com/Ks0052_keyestudio_PIR_Motion_Sensor

Módulo Led:

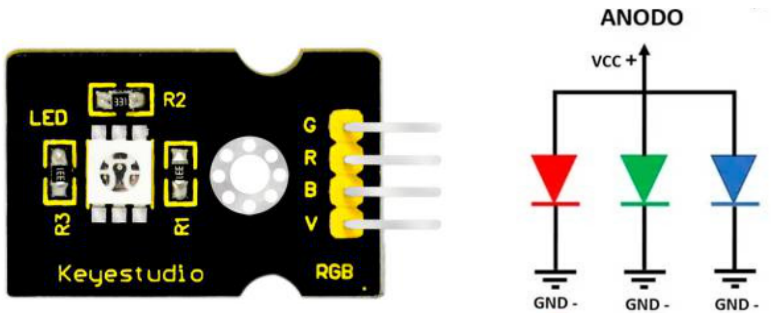
Programa de ejemplo:



Led RGB (colores aleatorios)

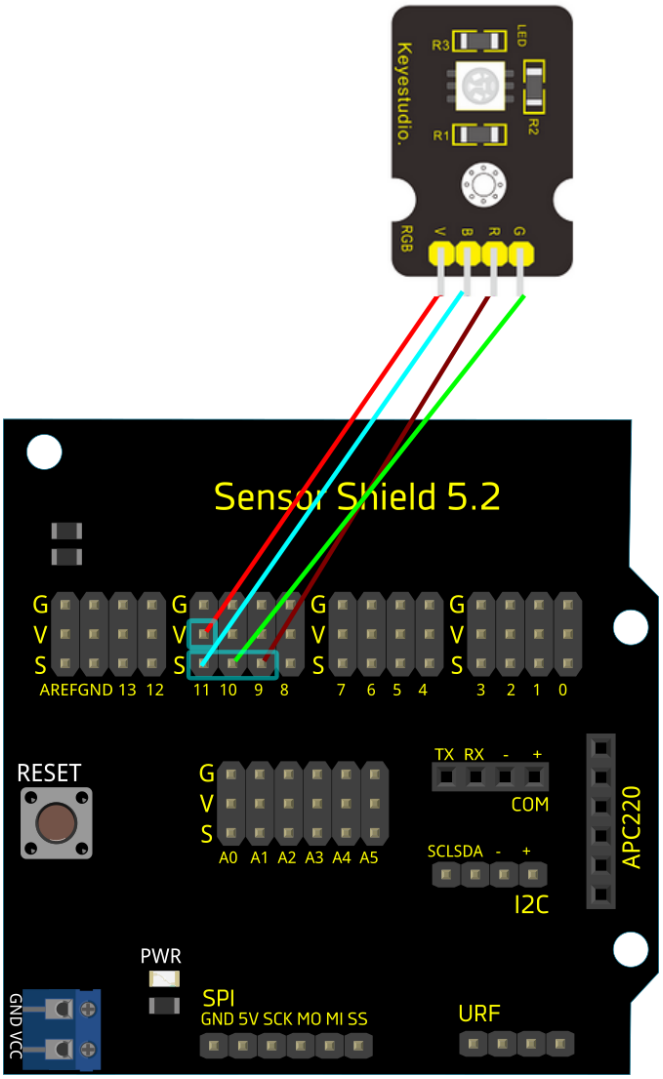
El módulo RGB (de ánodo común) incorpora 3 leds: Rojo, Verde, Azul (con el ánodo todos conectados a VCC, y controlamos la intensidad de cada color mediante PWM de forma inversa, siendo 0v la máxima intensidad)

Módulo led RGB:

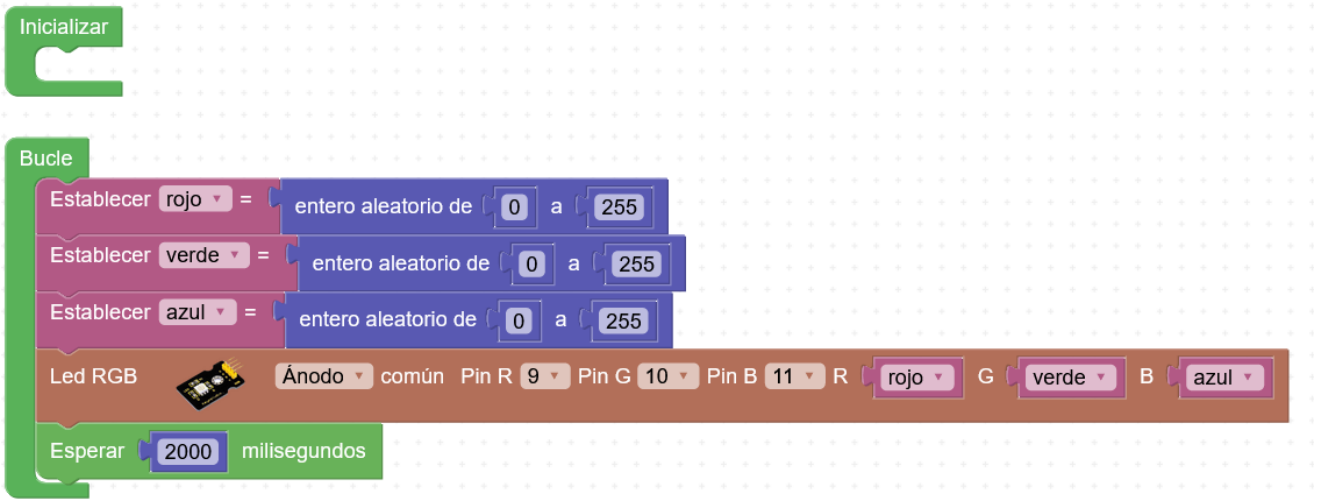


https://wiki.keyestudio.com/Ks0032_keyestudio_RGB_LED_Module

Esquema de conexiones:

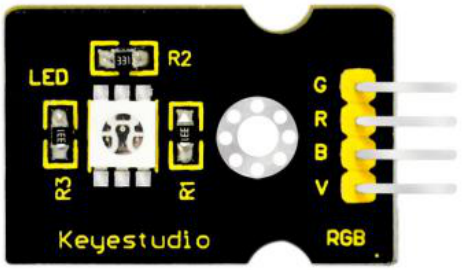


Programa de ejemplo:



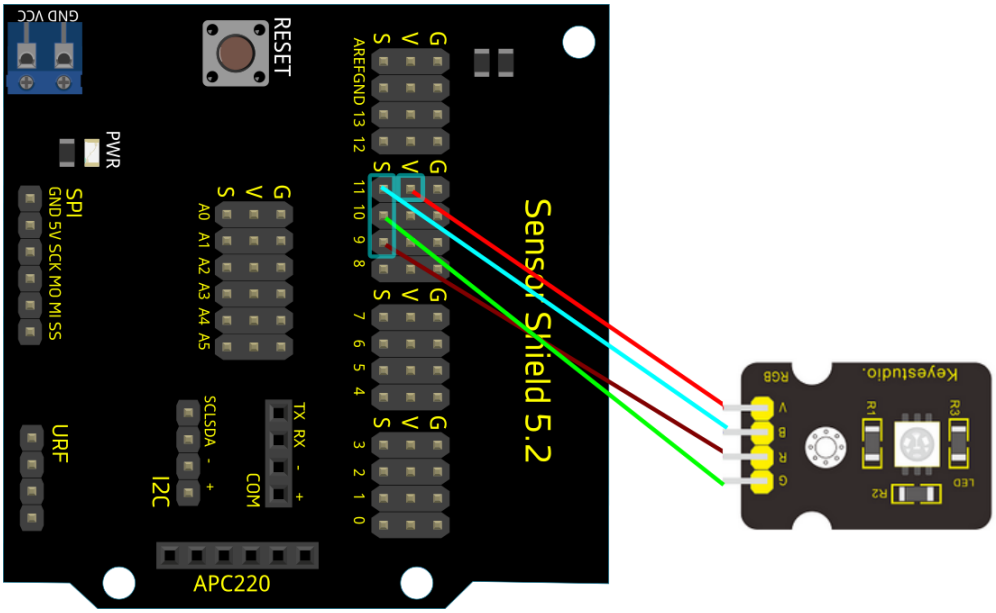
Led RGB (colores fijos)

Módulo led RGB:



https://wiki.keyestudio.com/Ks0032_keyestudio_RGB_LED_Module

Esquema de conexiones:



Programa de ejemplo:

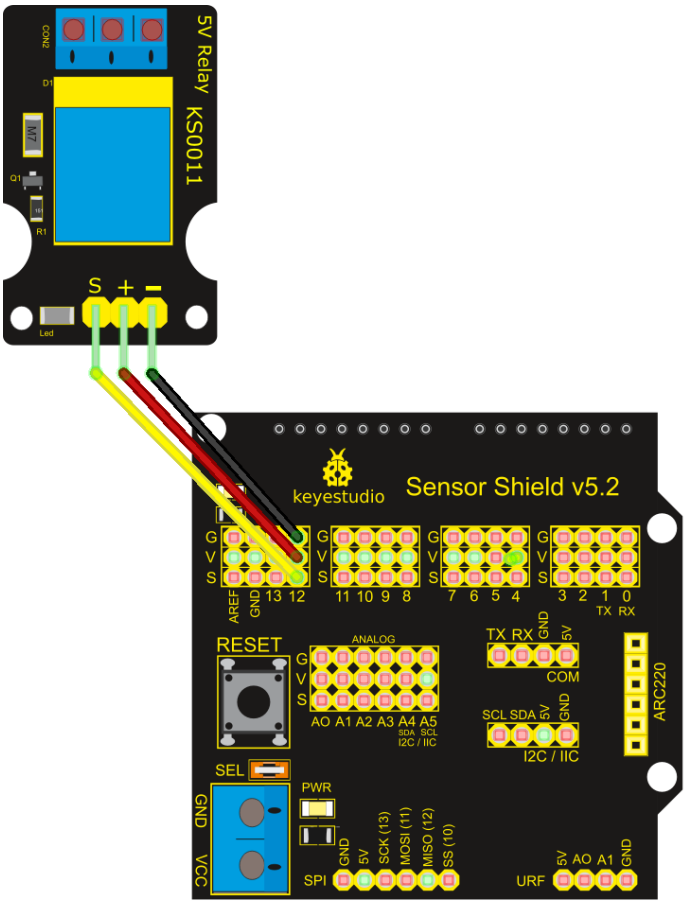
Relé

El relé es un interruptor controlado electrónicamente. Permite controlar cargas de potencia.

Módulo relé

https://wiki.keyestudio.com/Ks0011_keyestudio_5V_Relay_Module

Esquema de conexiones:

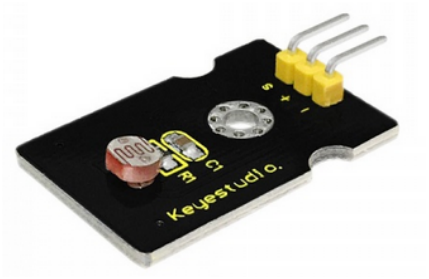


Programa de ejemplo:

LDR (consola serie)

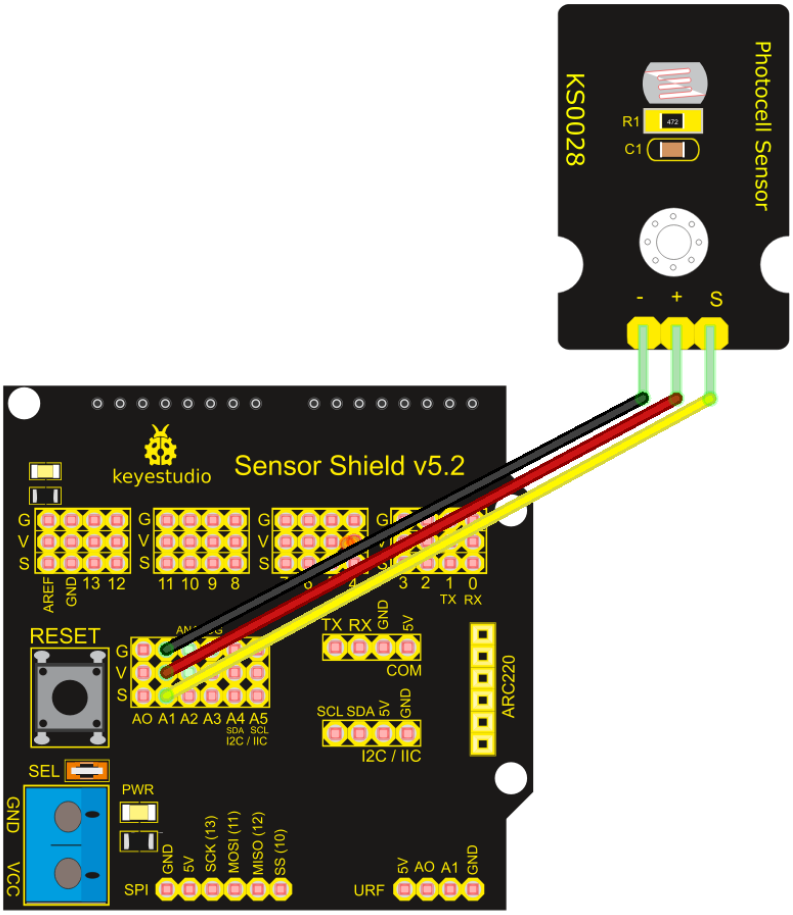
La fotocélula o LDR (Light Dependent Resistor) es una resistencia que varía según la intensidad de la luz, por lo que nos permite medir el nivel de luz ambiente fácilmente.

Módulo LDR:

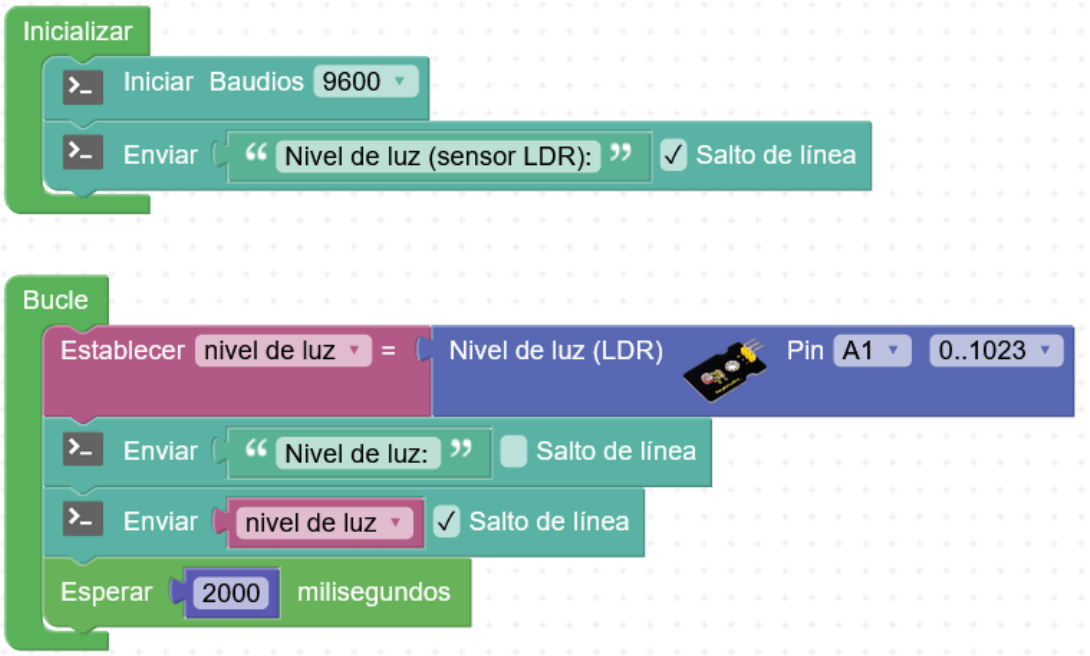


https://wiki.keyestudio.com/Ks0028_keyestudio_Photocell_Sensor

Esquema de conexiones:



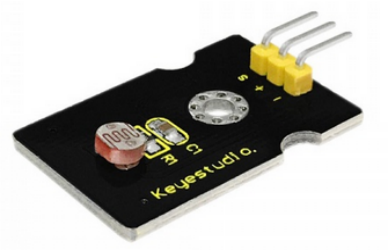
Programa de ejemplo:



Consola serie:

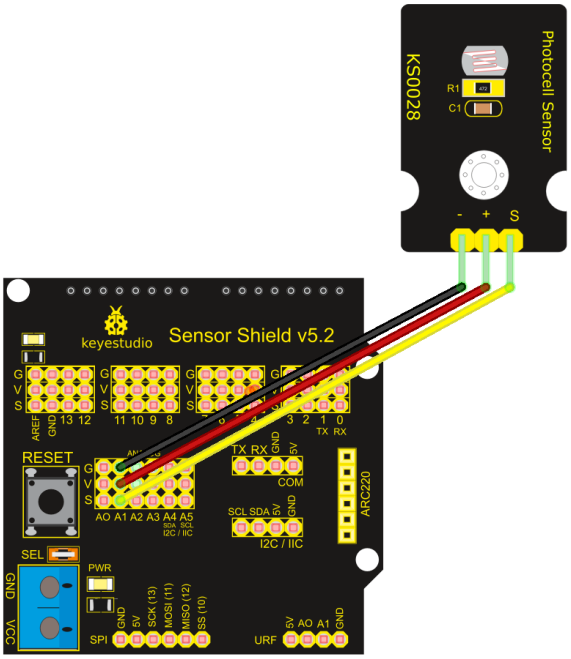
LDR (serial plotter)

Módulo LDR:

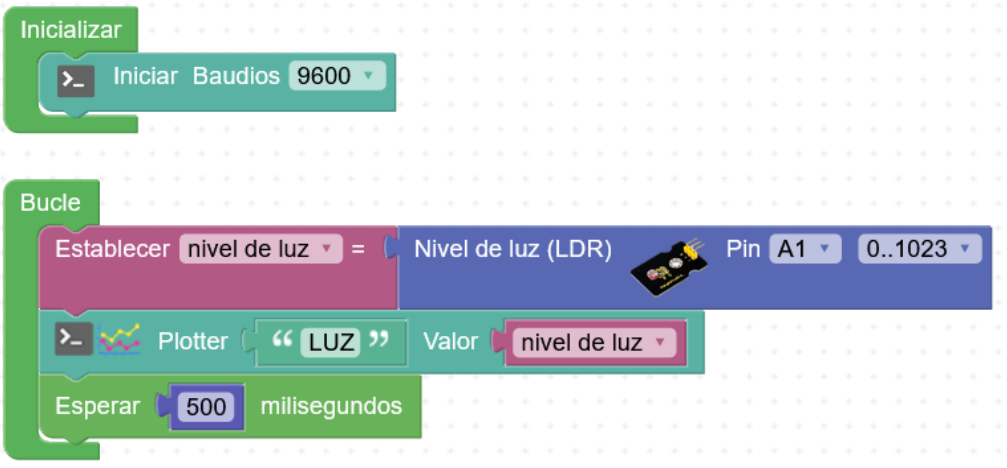


https://wiki.keyestudio.com/Ks0028_keyestudio_Photocell_Sensor

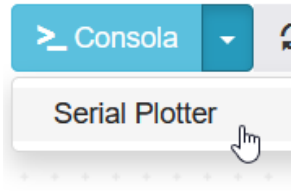
Esquema de conexiones:

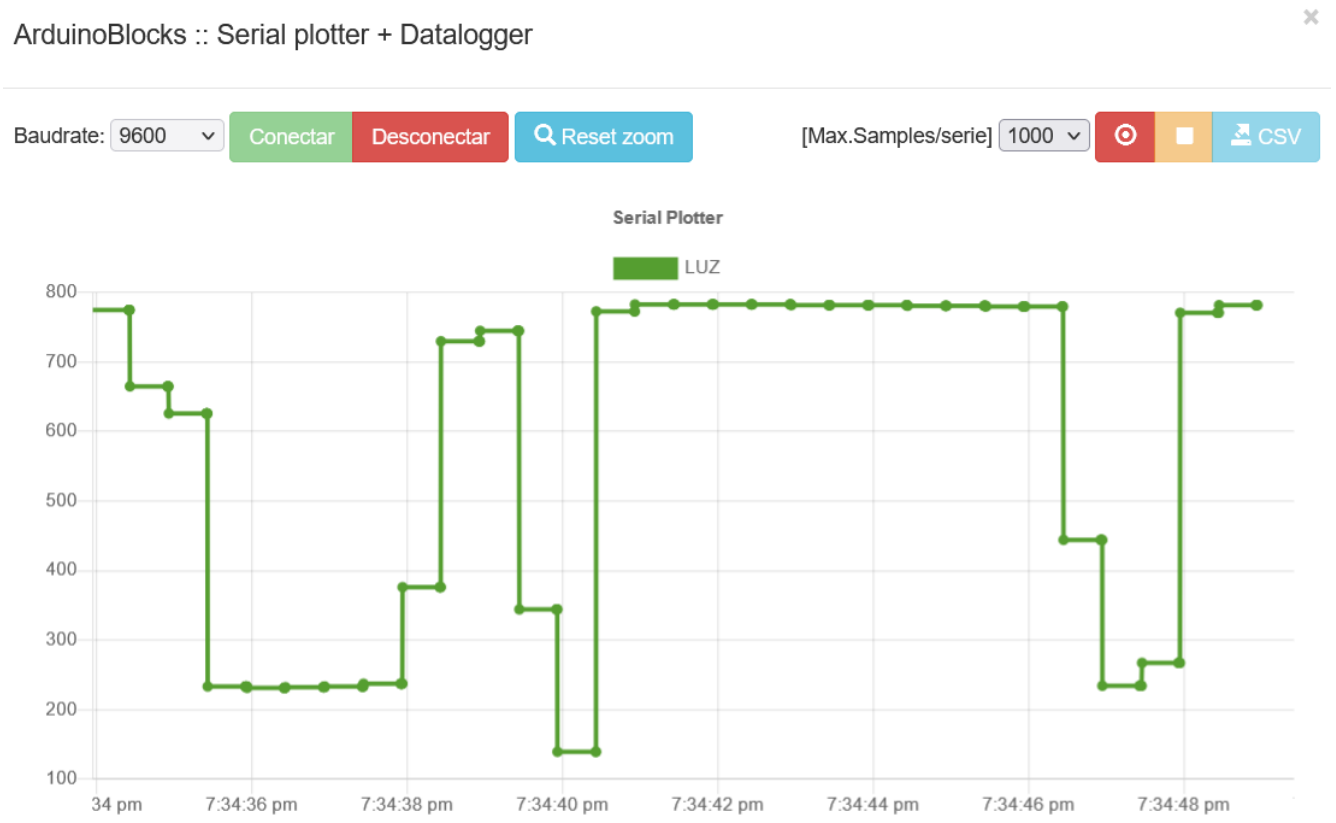


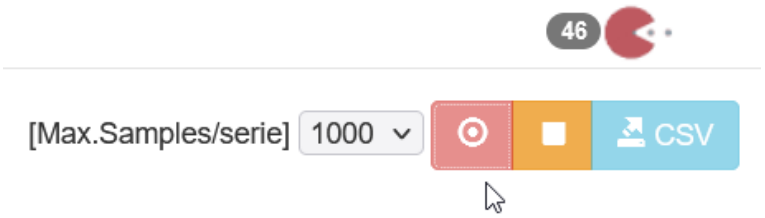
Programa de ejemplo:

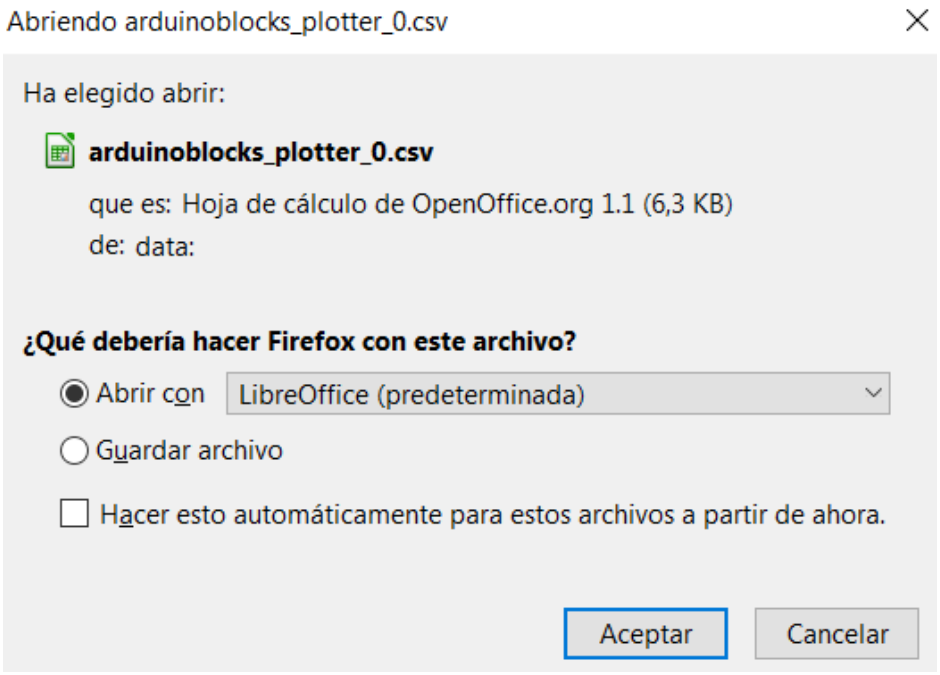


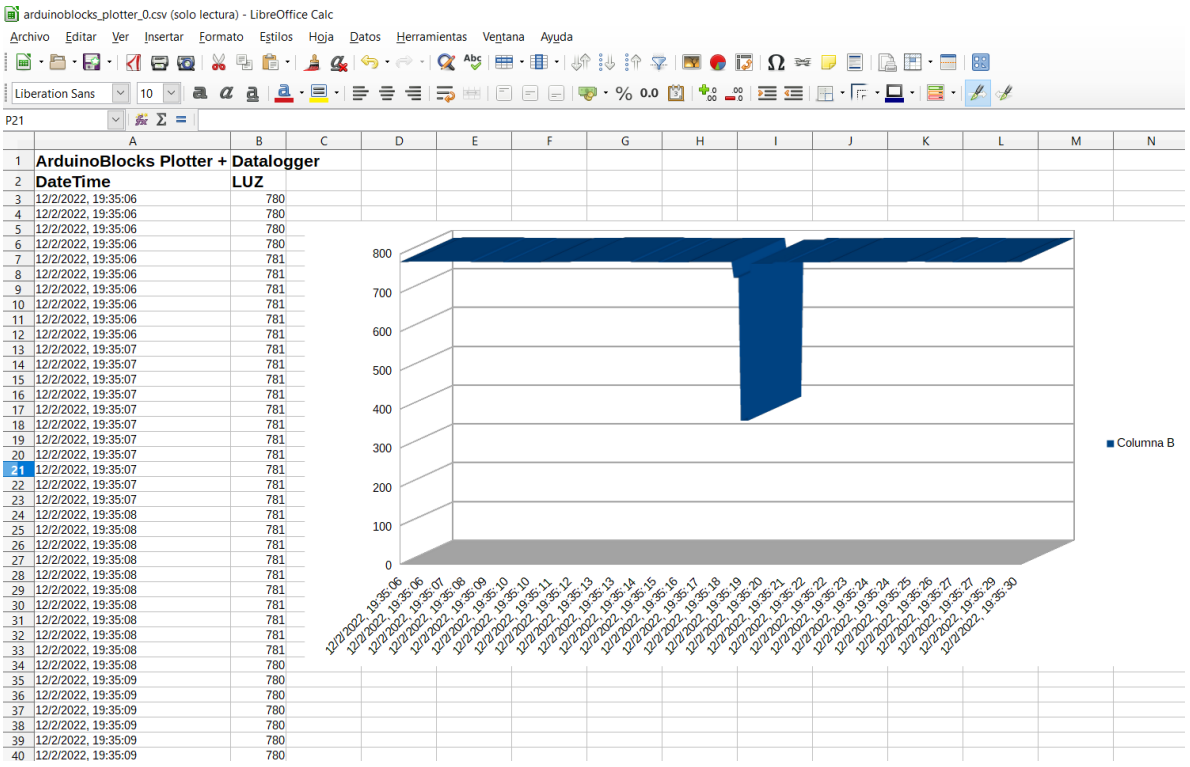
Serial plotter:







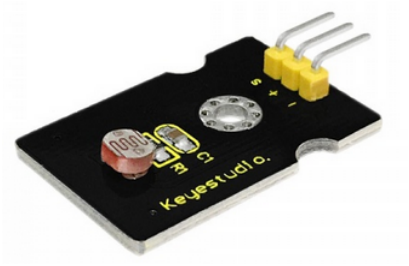




(si usamos varias series de datos, se descargará un .csv por cada una)

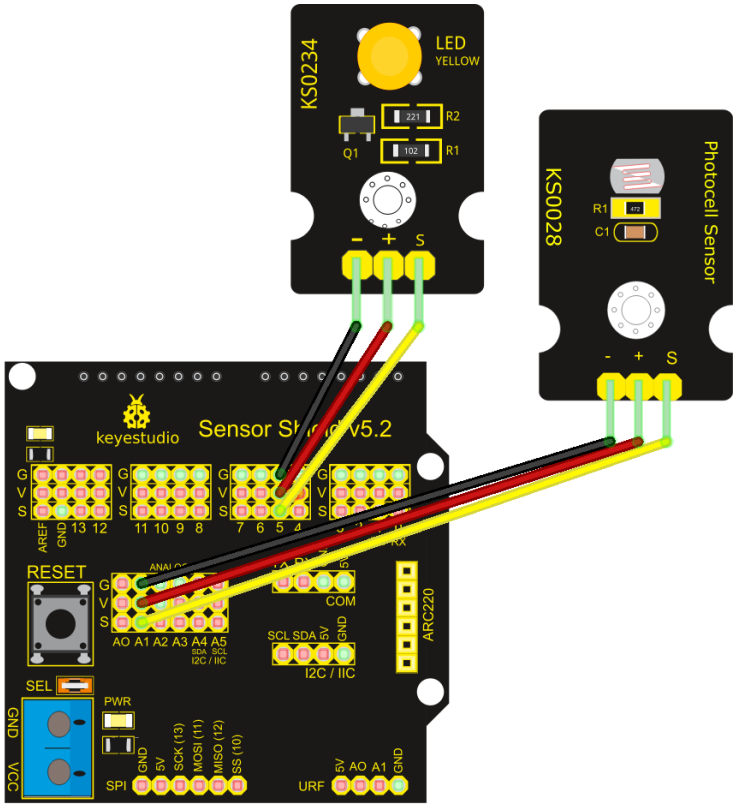
LDR + Relé (encendido automático por nivel de luz)

Módulo LDR:

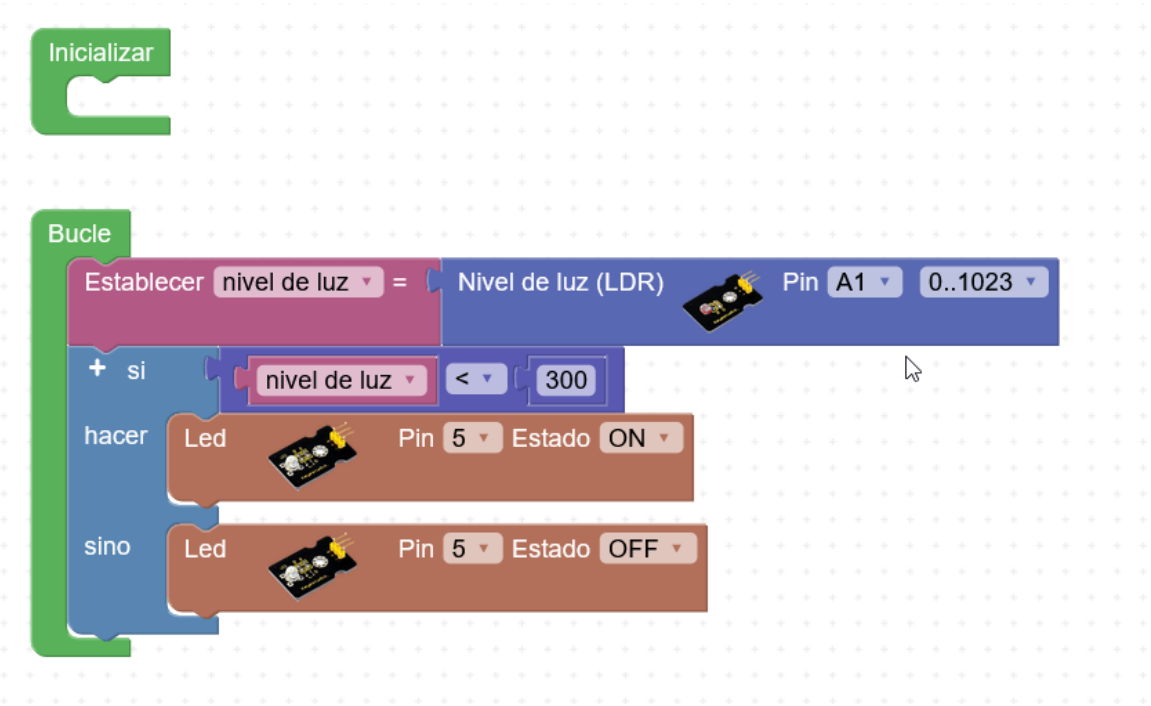


Módulo Led:

Esquema de conexiones:



Programa de ejemplo:



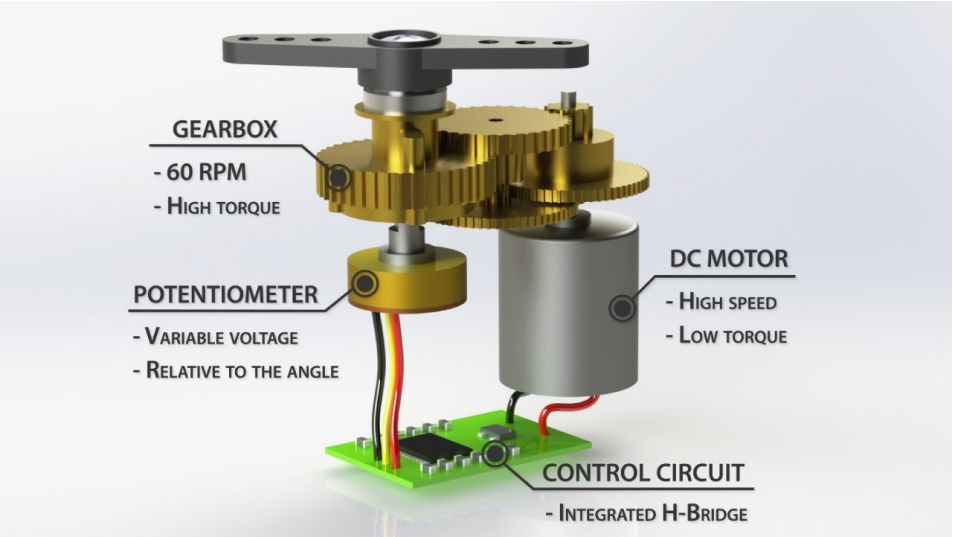
Control de servo (posicionamiento básico)

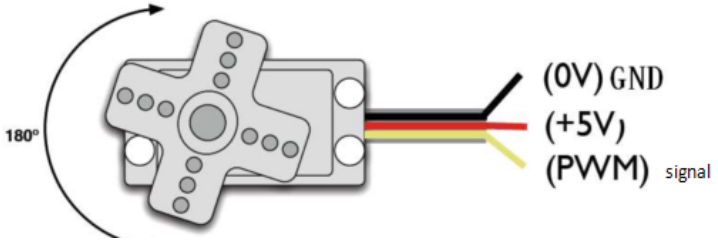
El servomotor es un motor de corriente continua con un controlador interno y un sistema de posicionamiento que nos permite situarlo en la posición deseada con una señal PWM externa. Los servomotores normales no tienen rotación continua.

Módulo servo:

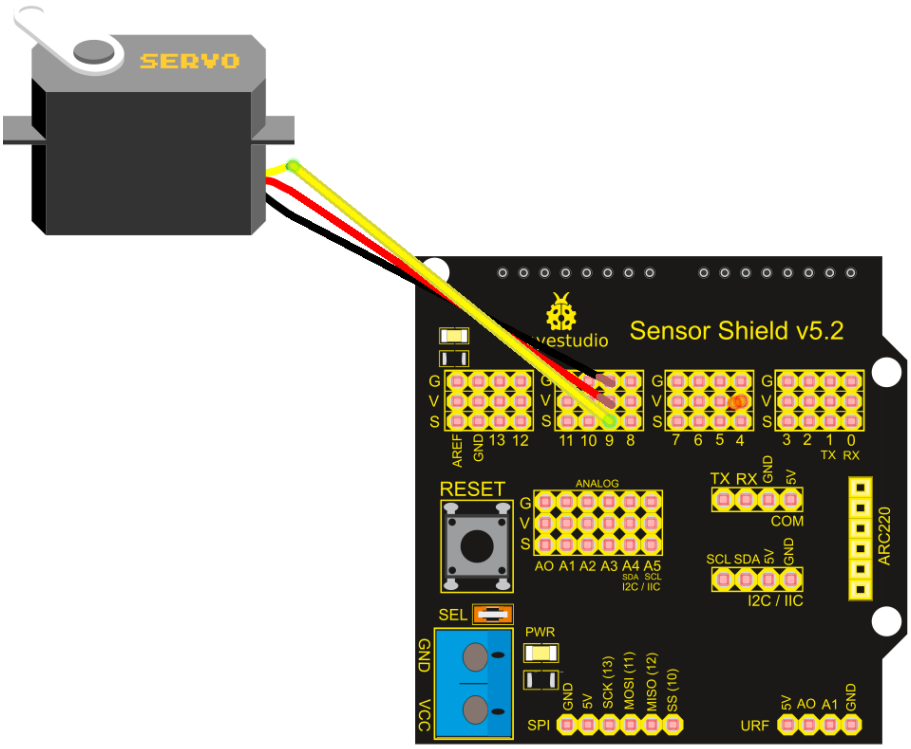
https://wiki.keyestudio.com/Ks0194_keyestudio_Micro_Servo

El microservo utilizado permite situarlo entre 0º y 180º





Esquema de conexiones:

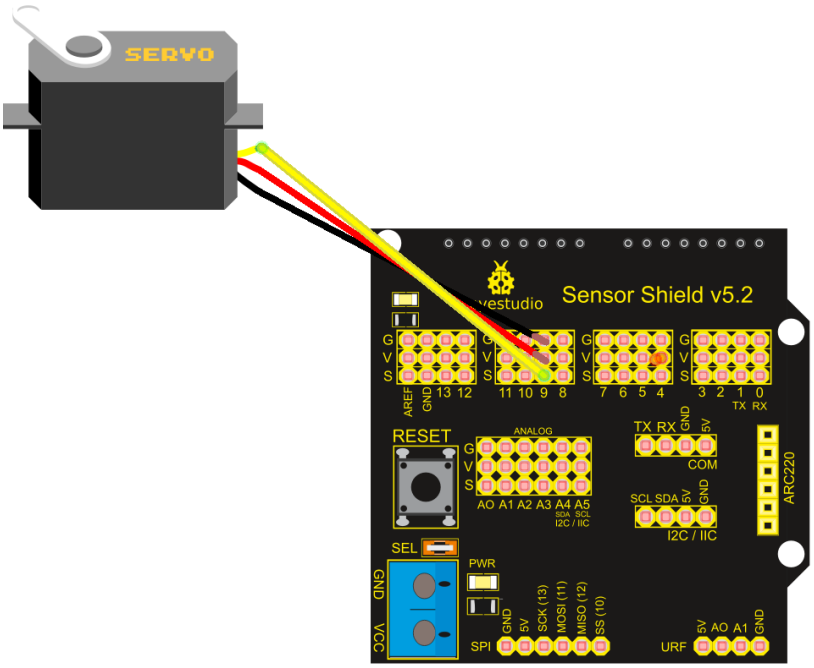


Programa de ejemplo:

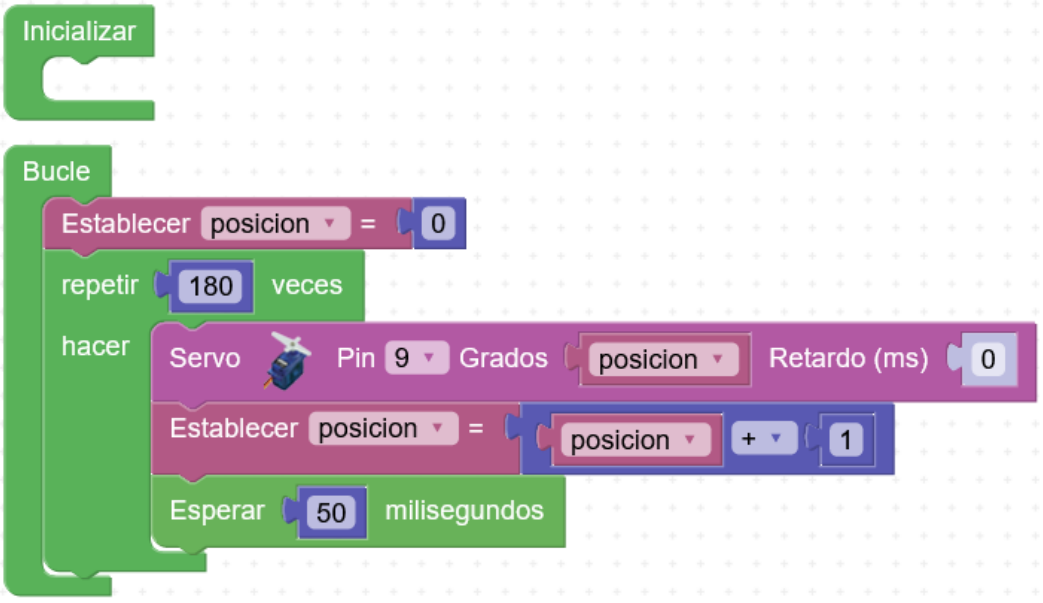
Control de servo (movimiento suave)

Módulo servo:

Esquema de conexiones:



Programa de ejemplo 1:



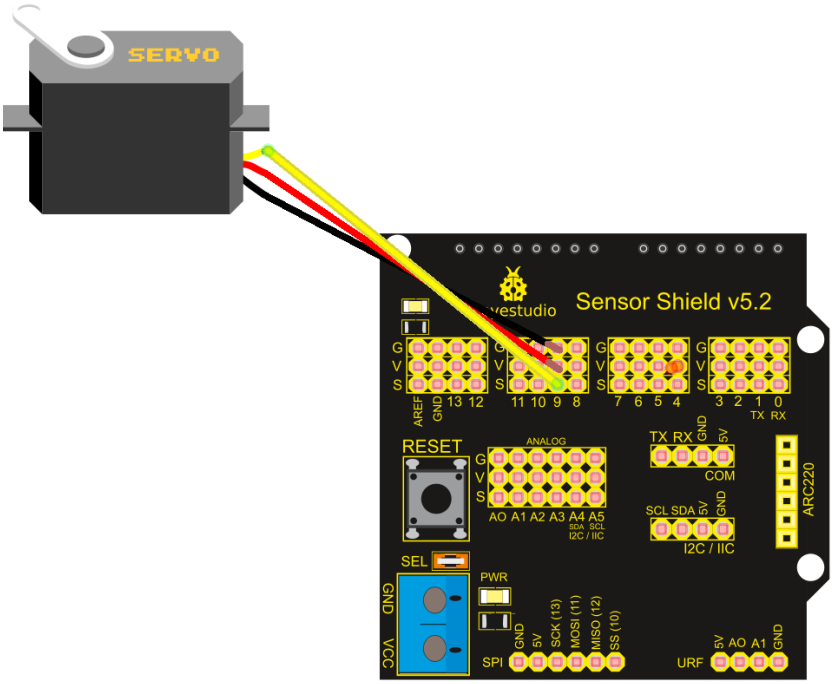
Programa de ejemplo 2:

Programa de ejemplo 3:

Control de servo (osciladores)

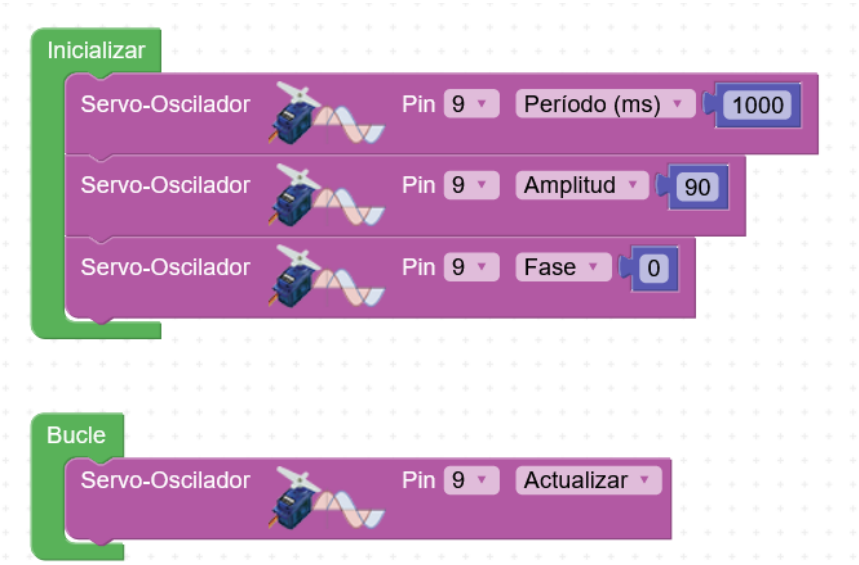
Módulo servo:

Esquema de conexiones:



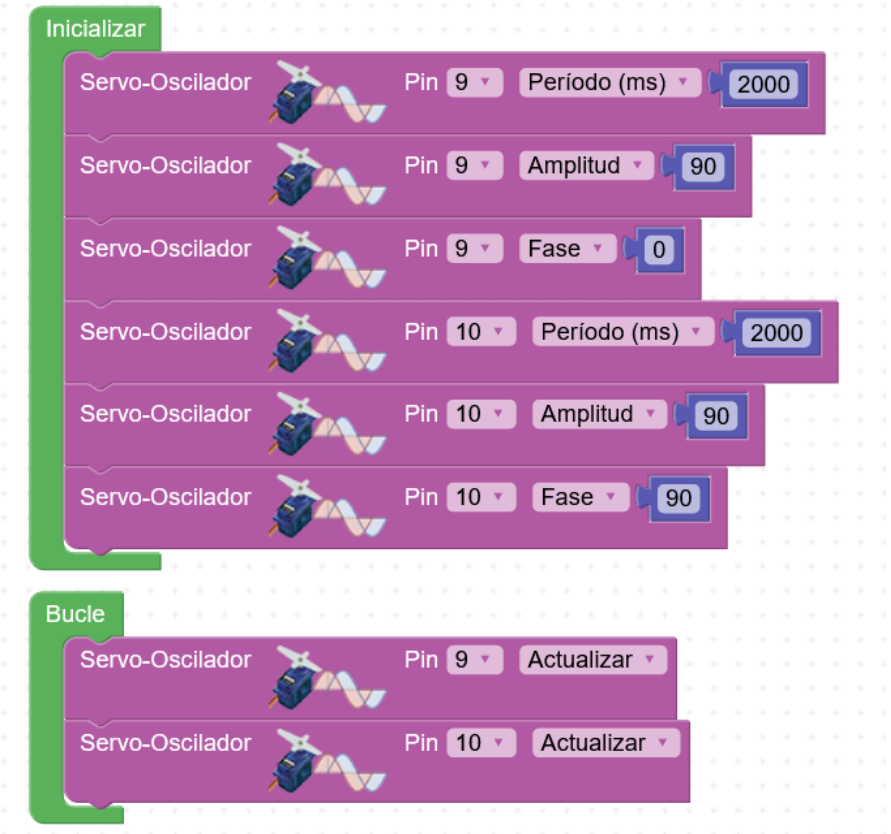
Programa de ejemplo 1:

1 servo, oscilando con una amplitud de +/- 90 ° (180 en total), en un período de 1000 ms (en 1s hace la oscilación completa):



Programa de ejemplo 2 (añadir servo en el pin 10):

2 servos, oscilando +/- 90º en 2000ms y uno desfasado 90º respecto al otro:



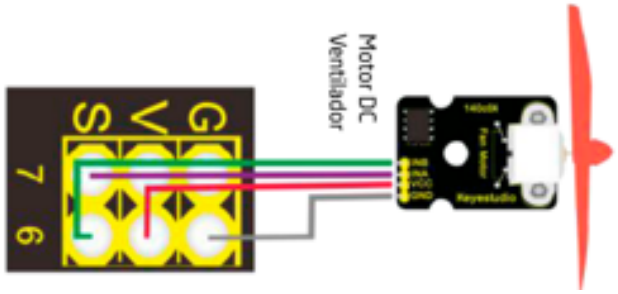
Motor DC (ventilador)

El módulo ventilador incorpora un pequeño motor DC y un driver para controlarlo permitiendo la inversión de giro y el control de velocidad mediante PWM.

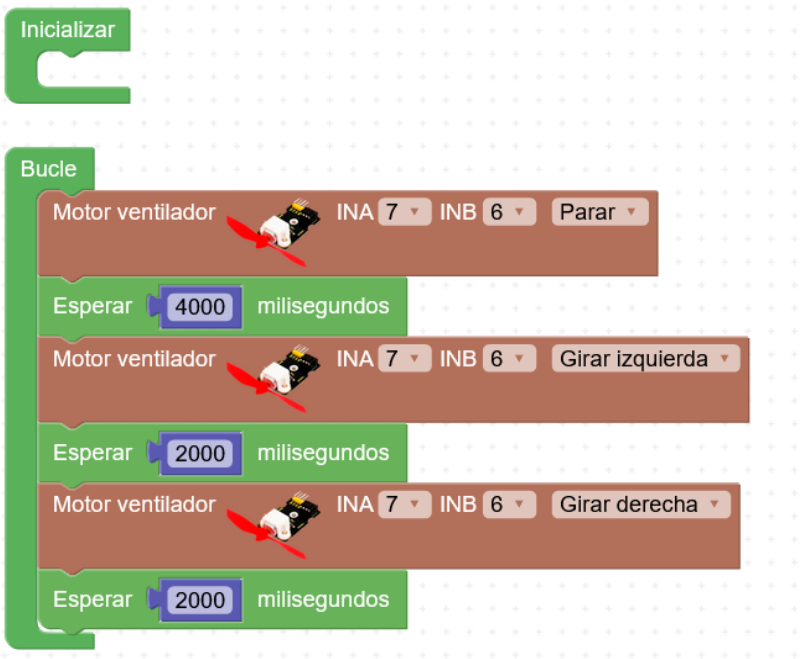
Módulo motor-ventilador:

https://wiki.keyestudio.com/Ks0168_keyestudio_L9110_fan_control_module

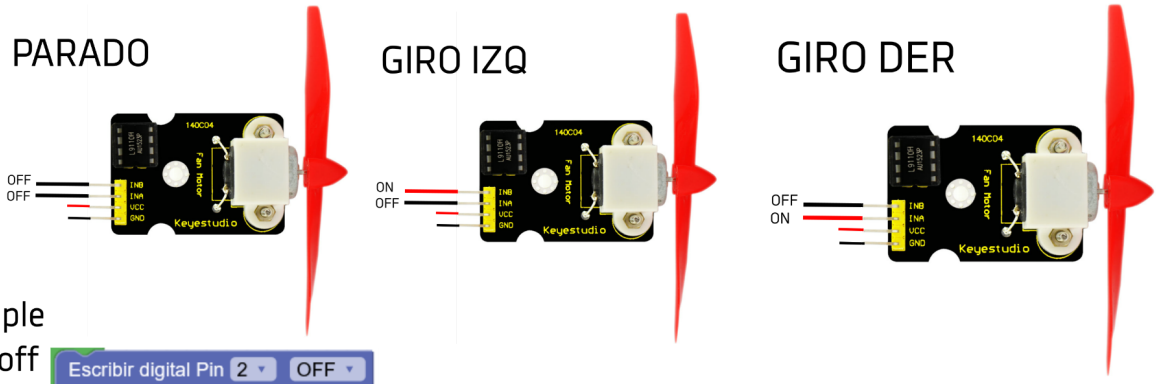
Esquema de conexiones:



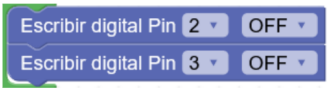
Programa de ejemplo:

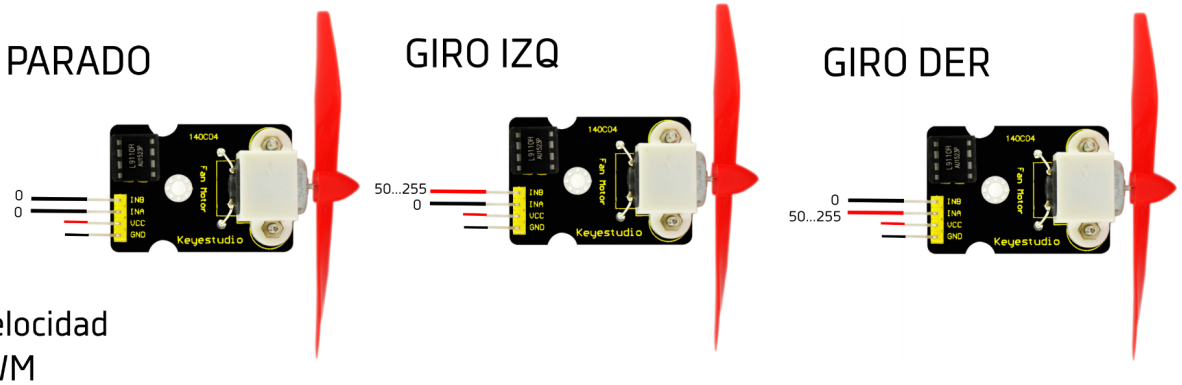


Ejemplo de control del módulo de ventilador:

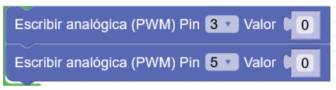


modo simple
pines on/off



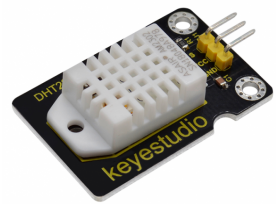


contro velocidad
pines PWM



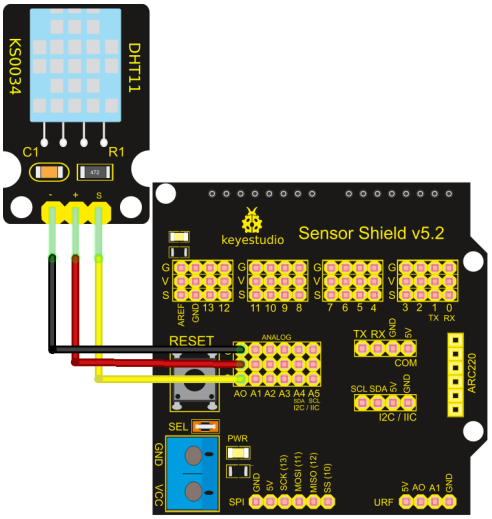
DHT22 (consola / serial plotter)

El sensor DHT22, es un sensor digiral de temperatura y humedad (mejora del DHT11) que permite leer valores de temperatura entre -40 y 125º (saltos de 0.5º) y valores de humedad relativa del aire entre 0 y 100%

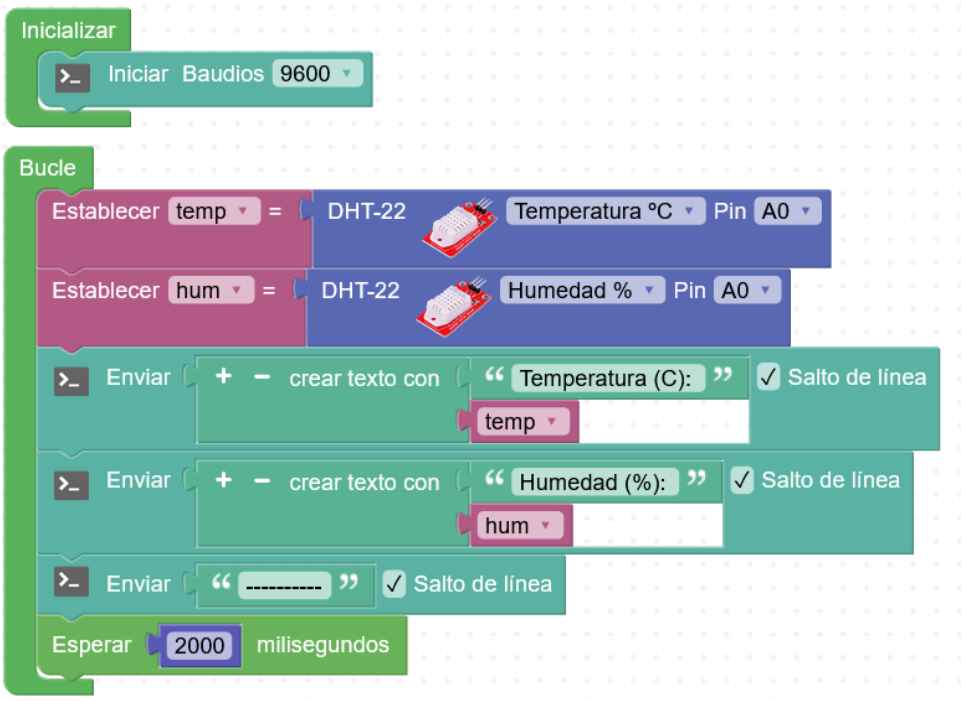


https://wiki.keyestudio.com/KS0430_Keyestudio_DHT22_Temperature_and_Humidity_Sensor

Esquema de conexiones:



Programa de ejemplo:

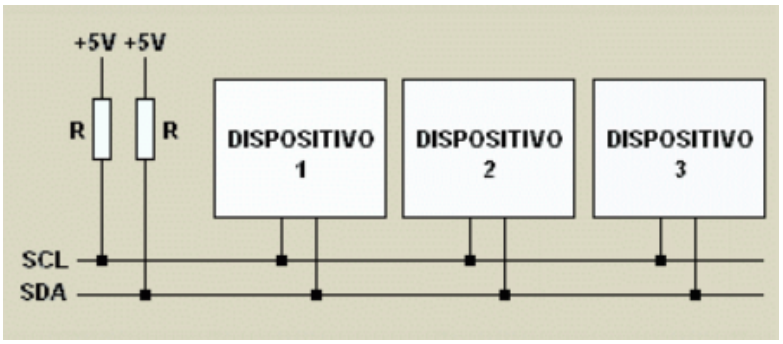


BUS I2C

El bus I2C permite conectar dispositivos o periféricos en forma de BUS (múltiples dispositivos comparten los 2 mismos cables de señales de comunicación)

El bus I2C se compone de dos seañes: SCL y SDA (y VCC y GND para la alimentación de los dispositivos)

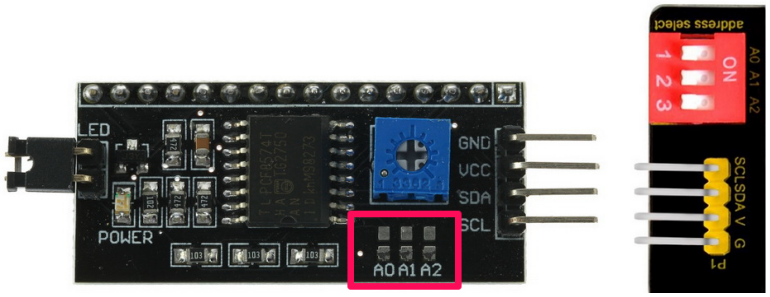
http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/



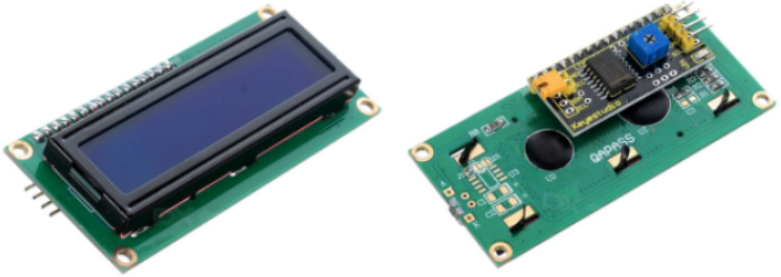
Dentro del bus cada dispositivo debe configurarse en una dirección diferente para evitar colisiones.

Los dispositivos de distinto tipo, por lo general, llevan direcciones diferentes. Si usamos dos dispositivos iguales en el bus (por ejemplo dos pantallas LCD) debemos cambiar la dirección de los dispositivos para que no estén en la misma dirección I2C.

La configuración de la dirección I2C se suele modificar mediante jumpers, microswitchs, o en algún caso hay que soldar y modificar unos contactos en la placa del sensor.



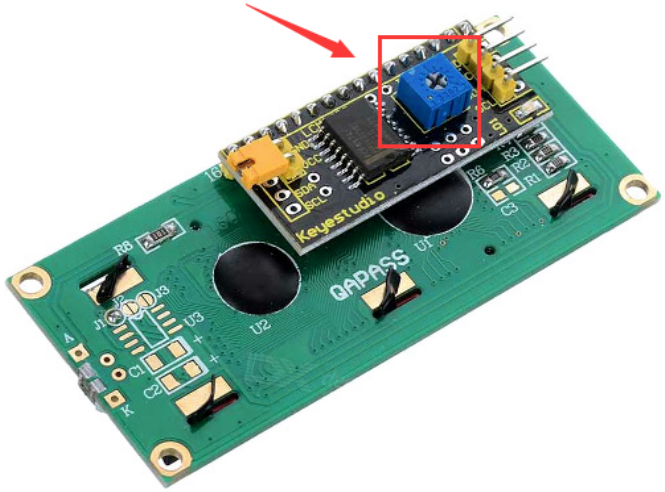
Pantalla LCD 1602 *(16 caracteres ancho x 2 líneas)*:



https://wiki.keyestudio.com/Ks0061_keyestudio_1602_I2C_Module

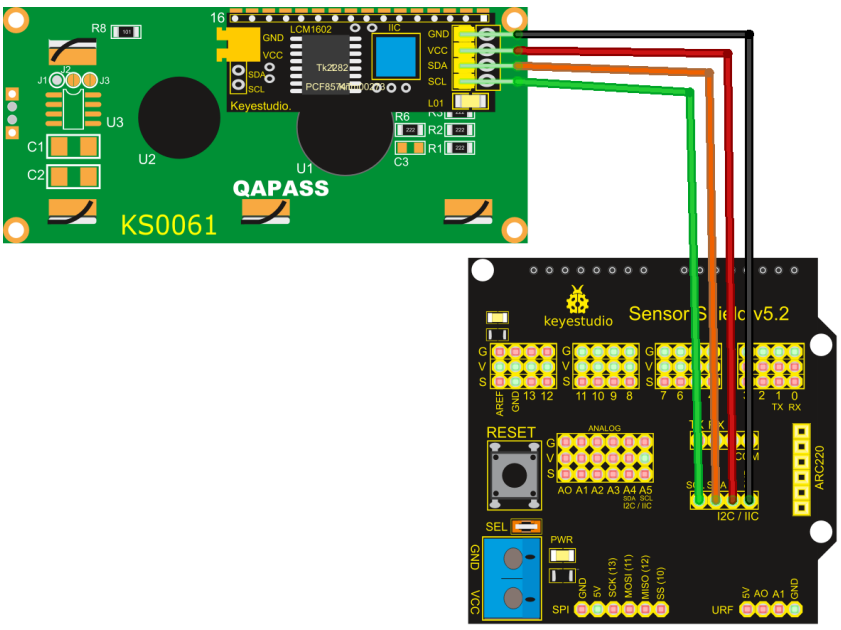
- Default I2C Address:**0x27**

Regulación/Ajuste del contraste:



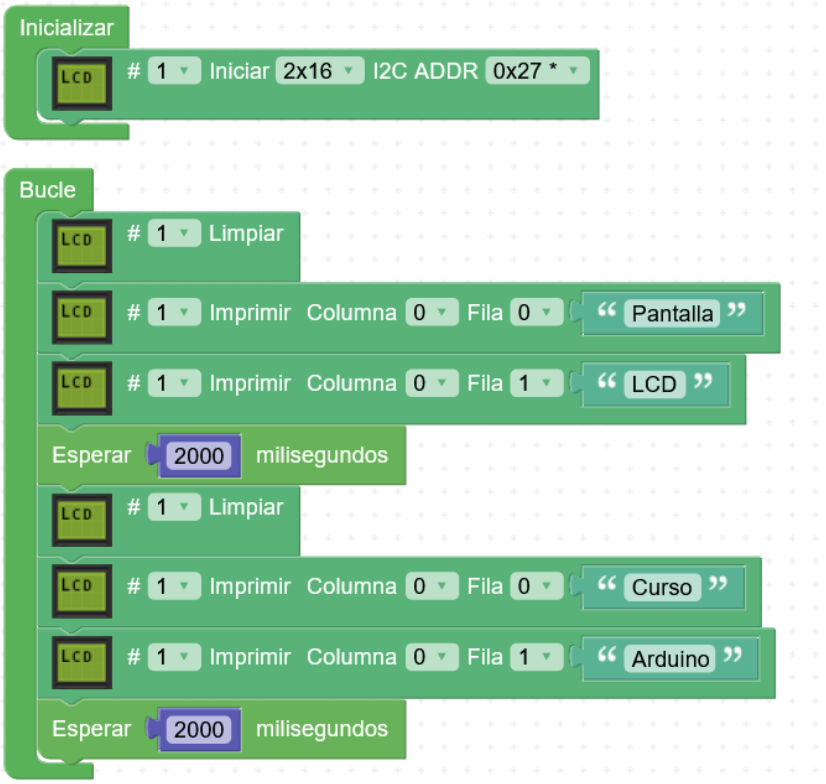
LCD (textos básicos)

La pantalla LCD permite mostrar caracteres alfanuméricos de forma sencilla en una pantalla de 2 filas y 16 columnas. La conexión se realiza mediante bus I2C.

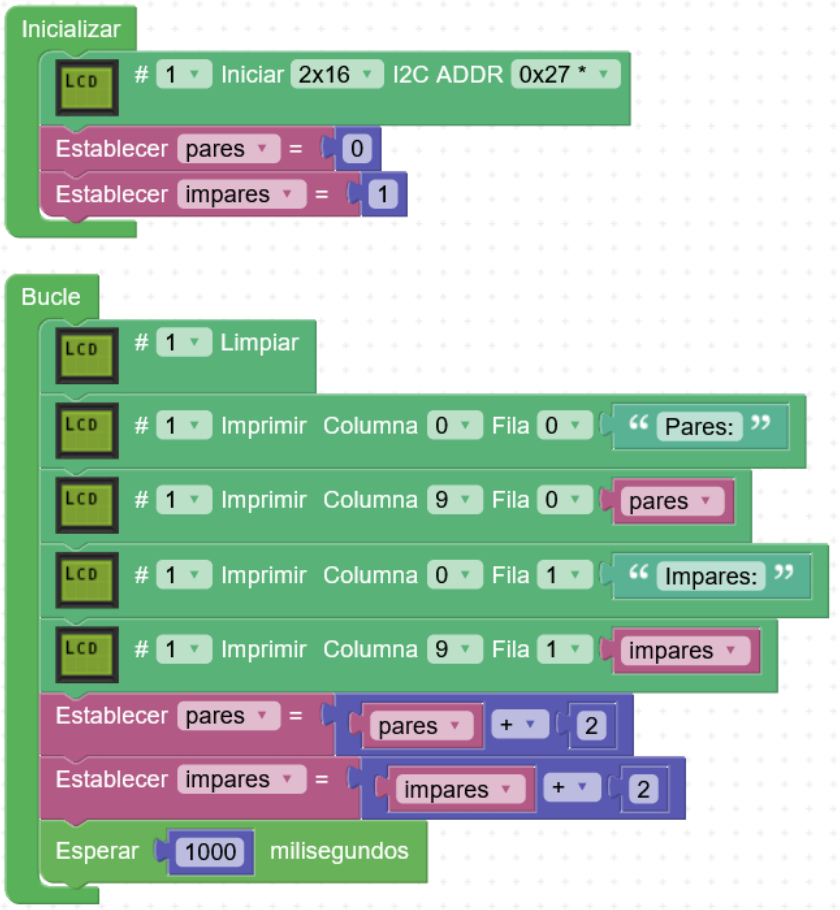


https://wiki.keyestudio.com/Ks0061_keyestudio_1602_I2C_Module

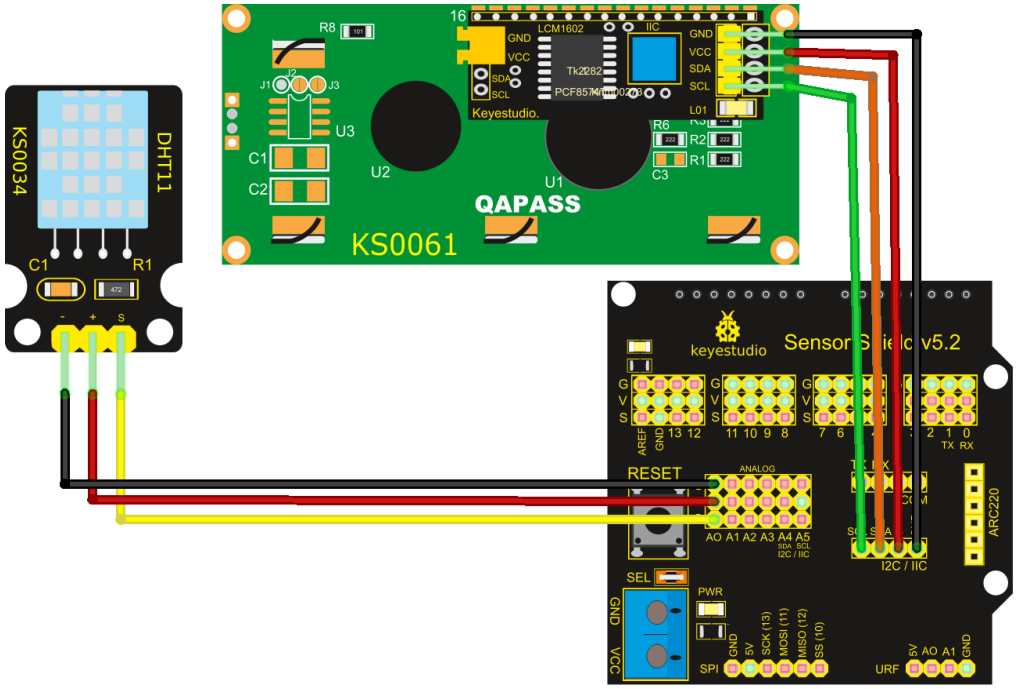
Programa de ejemplo 1:



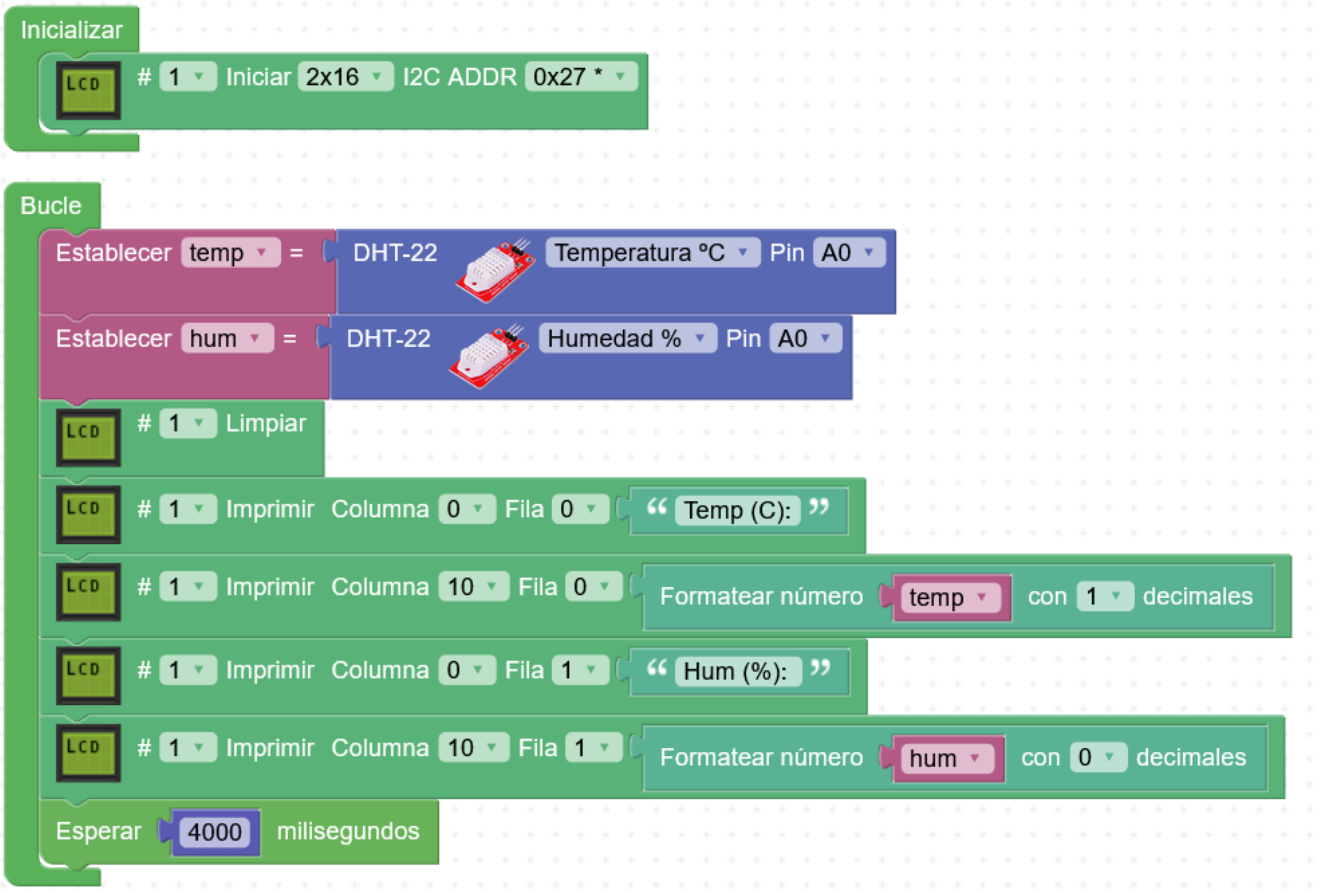
Programa de ejemplo 2:



Termómetro con LCD y DHT22



Programa:

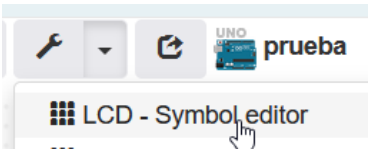


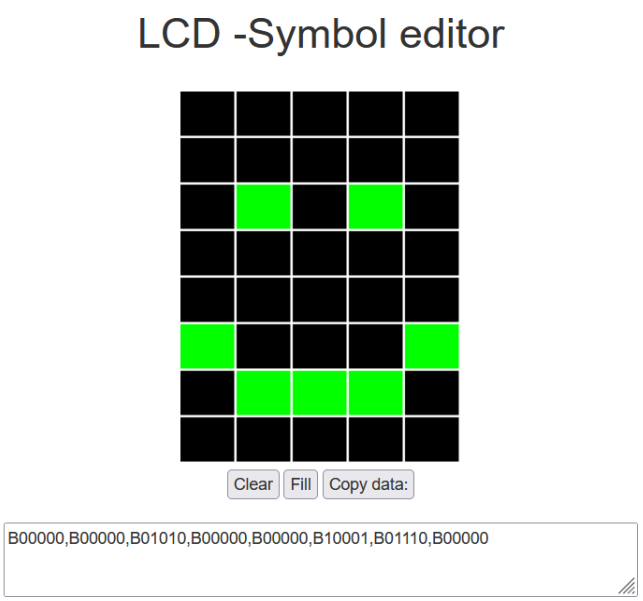
Símbolos personalizados

La pantalla LCD permite definir hasta 8 símbolos personalizados.

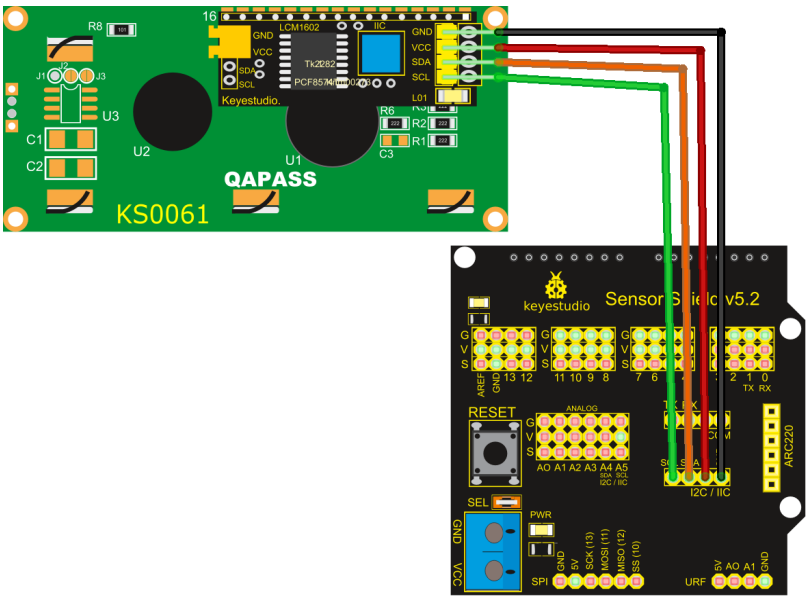
Para definir un símbolo podemos usar el editor que nos generará el "mapa de bits" que representa la imagen del símbolo.

http://www.arduinoblocks.com/web/help/chareditor

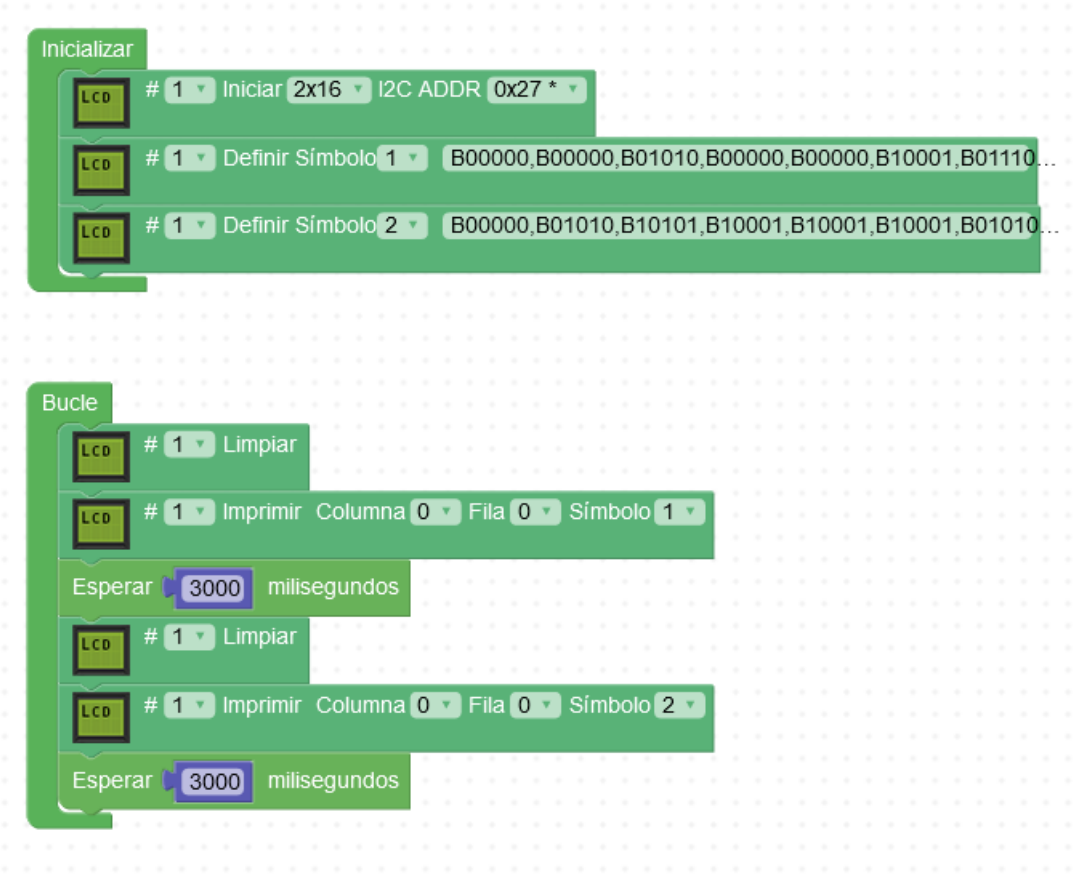




Conexiones:



Programa:

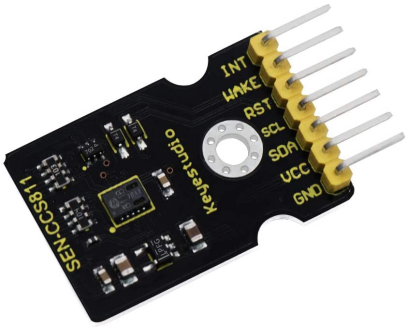


Algunas ideas:

Sensor CO2: CSS811

El sensor CSS811 es un sensor que puede detectar una amplia gama de compuestos orgánicos volátiles (TVOC) incluyendo el eCO2 (CO2 equivalente) .

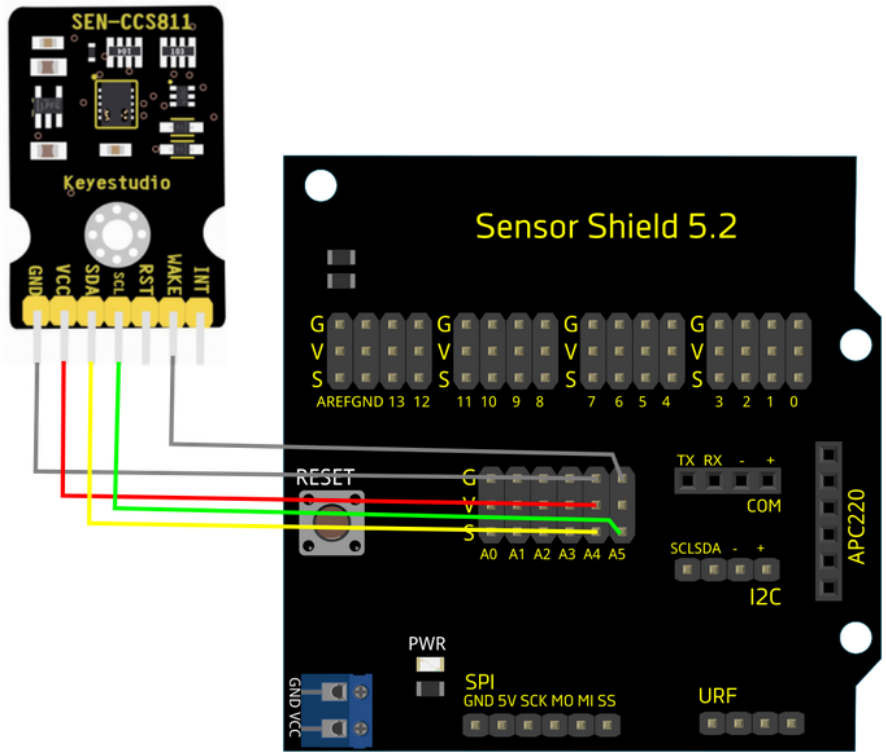
Rango medición eCO2: 400 ... 29206 ppm (partes por millón)



https://wiki.keyestudio.com/KS0457_keyestudio_CCS811_Carbon_Dioxide_Air_Quality_Sensor

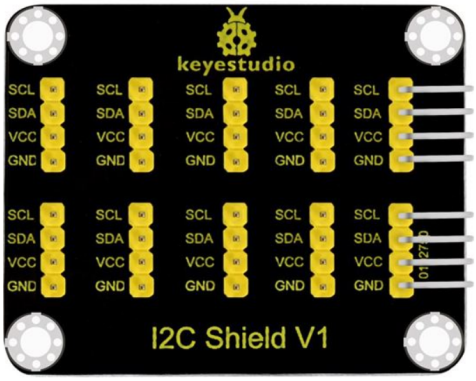
El sensor se conecta mediante bus I2C, podemos conectarlo directamente al bus I2C de la shield.

Pero como posteriormente utilizaremos la pantalla también, podemos conectarlo a los pines que internamente también están conectados al bus I2C: A4=SDA, A5=SCL



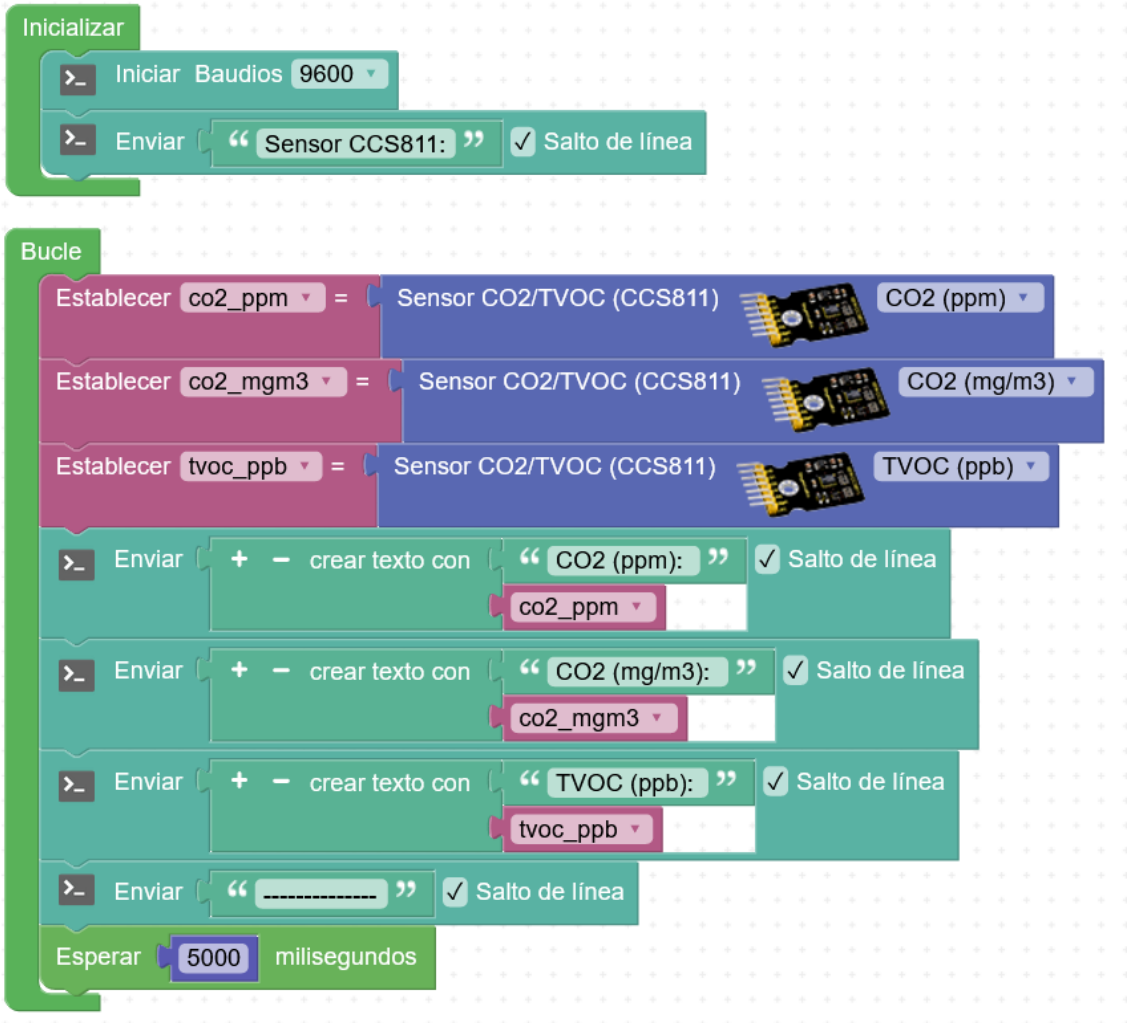
Otra opción posible cuando necesitamos conectar varios dispositivos al bus I2C es usar un HUB I2C para "ramificar el bus":

https://shop.innovadidactic.com/es/standard-perifericos/872-keyestudio-hub-i2c.html



Programa para medir los niveles de eCO2 y TVOC:

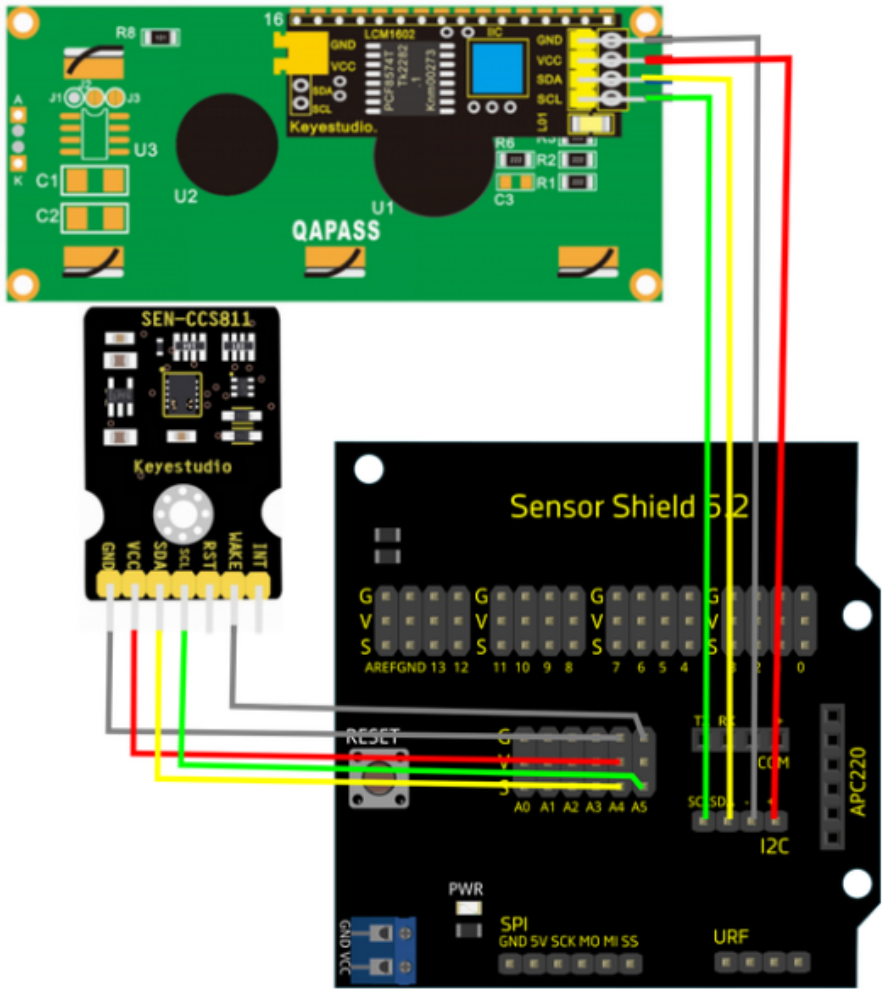
(el sensor necesita un tiempo de calentamiento/ajuste , por lo que puede tardar unos minutos en dar niveles correctos)

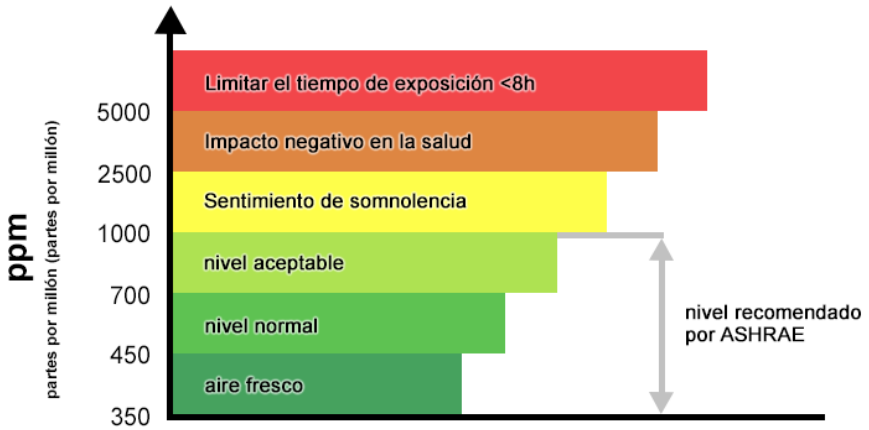


Medidor CO2 con LCD y sensor CCS811

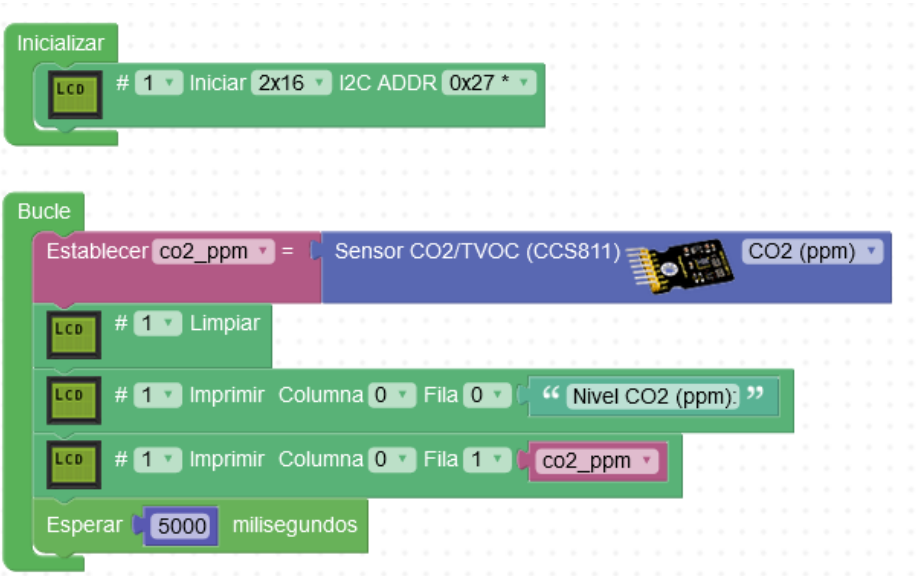
Conexiones: (ya que tanto la pantalla LCD como el sensor CCS811 usan I2C, para no tener que ramificarlo o usar un HUB, usamos el "truco" de que los pines I2C (SDA/SCL) corresponden internamente a los pines A4=SDA y A5=SCL, por lo que ya lo tenemos ramificado en dos conexiones)

Conexiones:

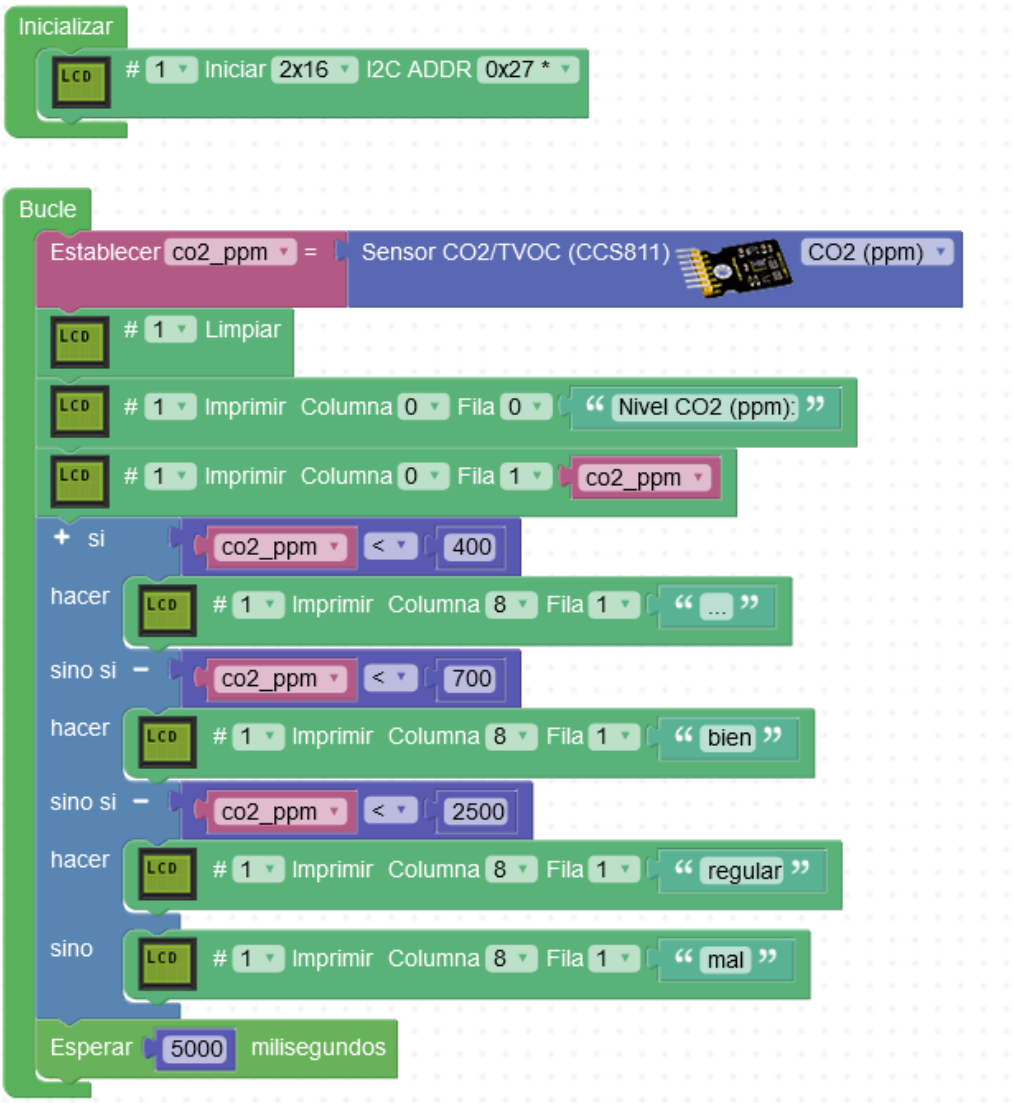




Programa 1:



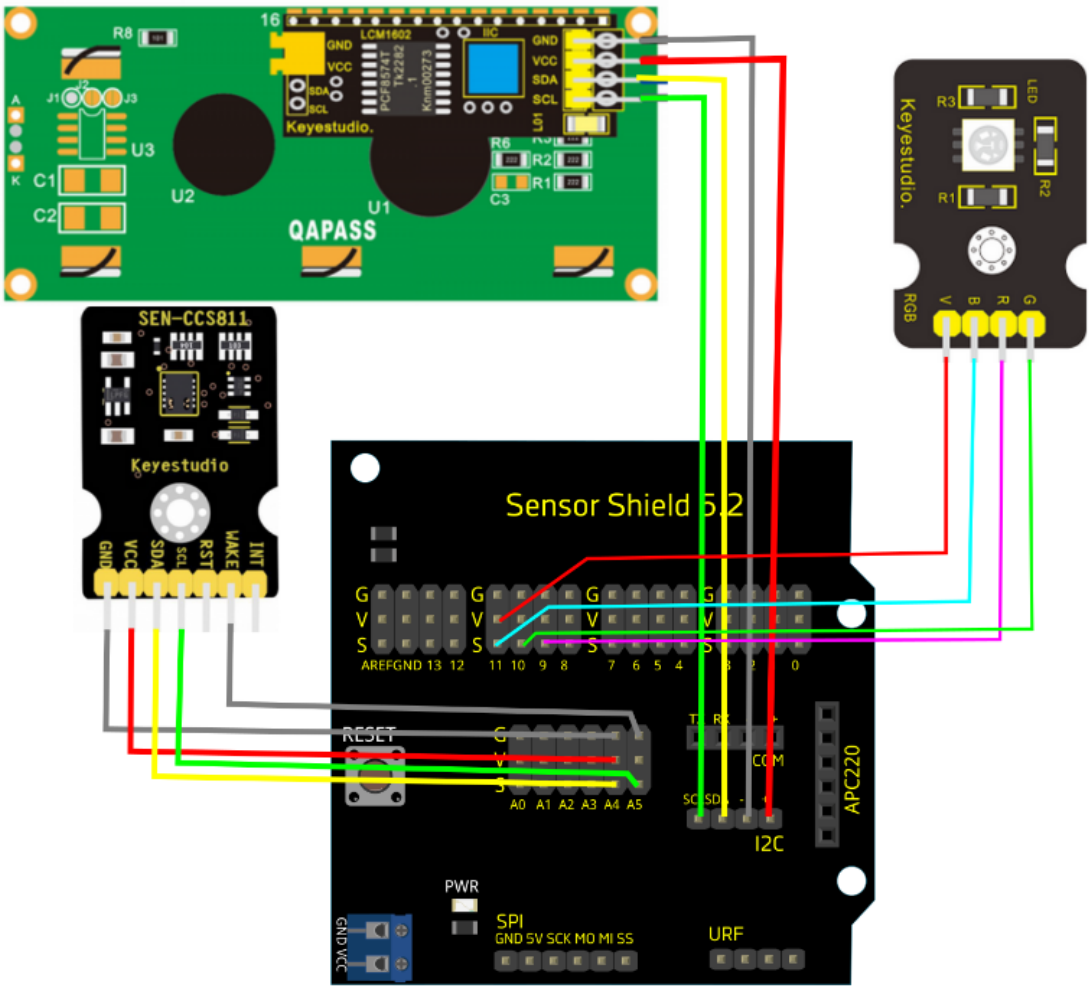
Programa 2:



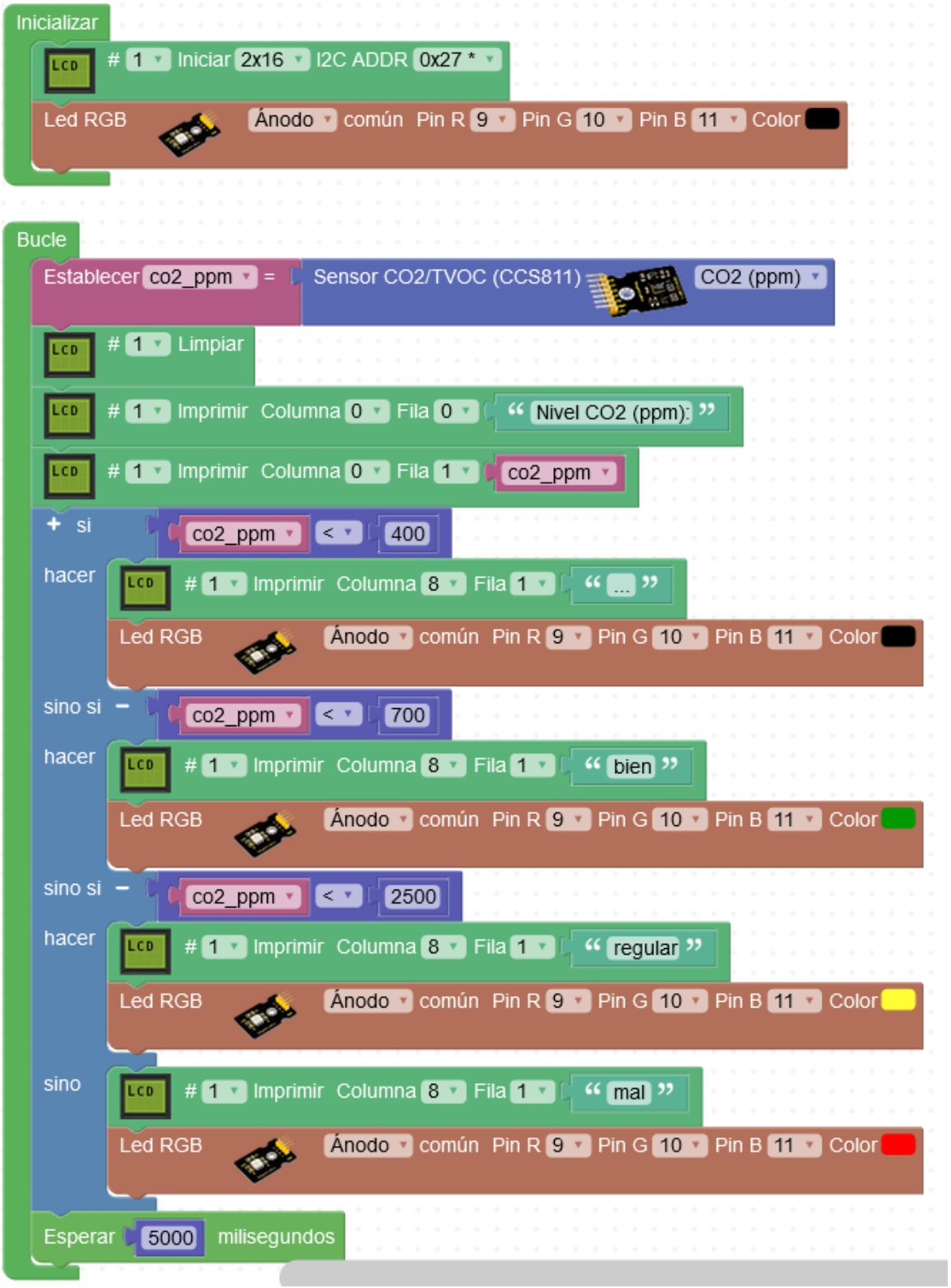
Semáforo CO2 con Led RGB y sensor CSS811

A la práctica anterior le añadimos un led RGB , para visualmente mostrar el estado de la calidad del aire en función del nivel de CO2

Conexiones:

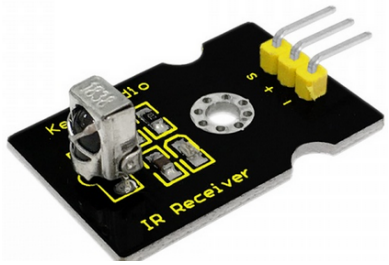


Progama:


Inicializar
LCD # 1 Iniciar 2x16 I2C ADDR 0x27 *
Led RGB Ánodo común Pin R 9 Pin G 10 Pin B 11 Color
Bucle
Establecer co2_ppm = Sensor CO2/TVOC (CCS811) CO2 (ppm)
LCD # 1 Limpiar
LCD # 1 Imprimir Columna 0 Fila 0 " Nivel CO2 (ppm): "
LCD # 1 Imprimir Columna 0 Fila 1 co2_ppm
si co2_ppm < 400
hacer
LCD # 1 Imprimir Columna 8 Fila 1 " ... "
Led RGB Ánodo común Pin R 9 Pin G 10 Pin B 11 Color
sino si co2_ppm < 700
hacer
LCD # 1 Imprimir Columna 8 Fila 1 " bien "
Led RGB Ánodo común Pin R 9 Pin G 10 Pin B 11 Color
sino si co2_ppm < 2500
hacer
LCD # 1 Imprimir Columna 8 Fila 1 " regular "
Led RGB Ánodo común Pin R 9 Pin G 10 Pin B 11 Color
sino
LCD # 1 Imprimir Columna 8 Fila 1 " mal "
Led RGB Ánodo común Pin R 9 Pin G 10 Pin B 11 Color
Esperar 5000 milisegundos

Mando a distancia IR

Receptor IR



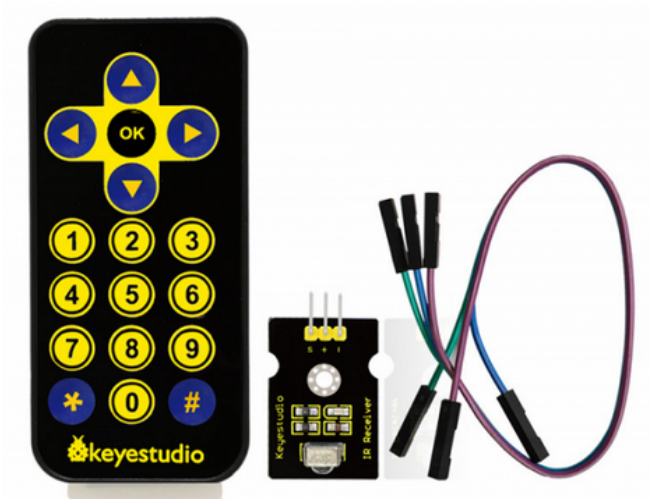
https://wiki.keyestudio.com/Ks0026_keyestudio_Digital_IR_Receiver_Module

https://wiki.keyestudio.com/Ks0088_New_Infrared_IR_Wireless_Remote_Control_Module_Kits_for_Arduino

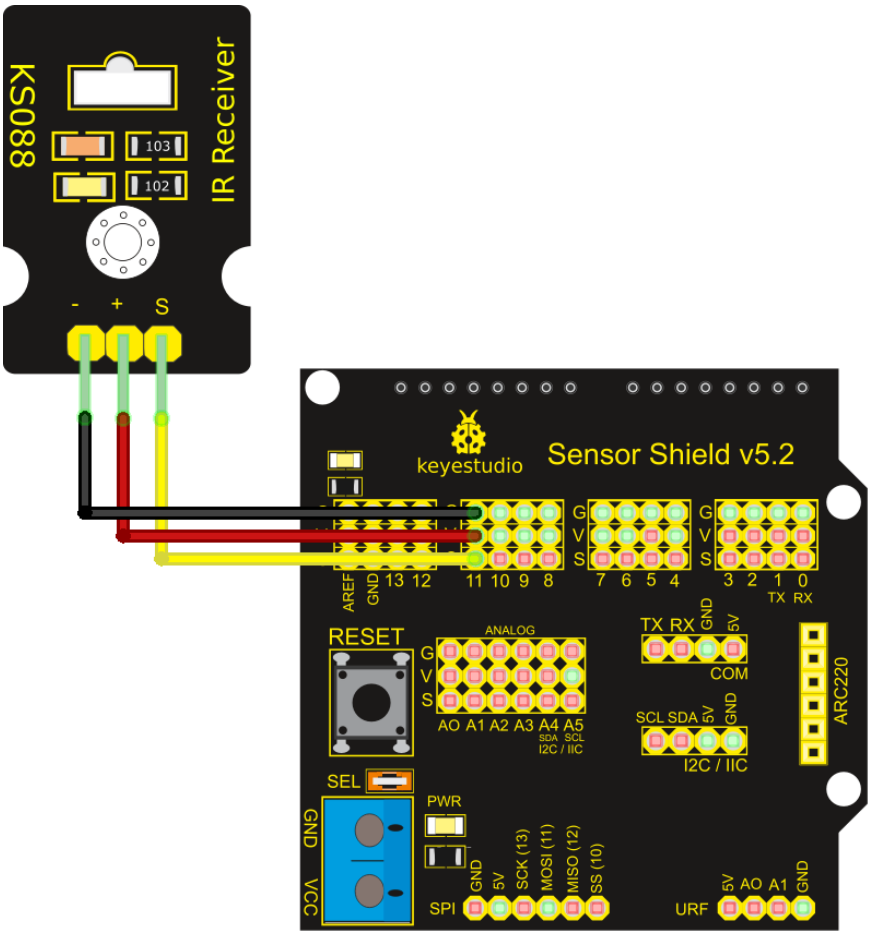
El receptor es capaz de recibir y decodificar señales de luz infrarroja, procedente de mandos a distancia.

Podemos obtener los códigos de los botones de los mandos de casi cualquier marca (protocolos: NEC, Sony, RC5, RC6, Panasonic, JVC, ...)

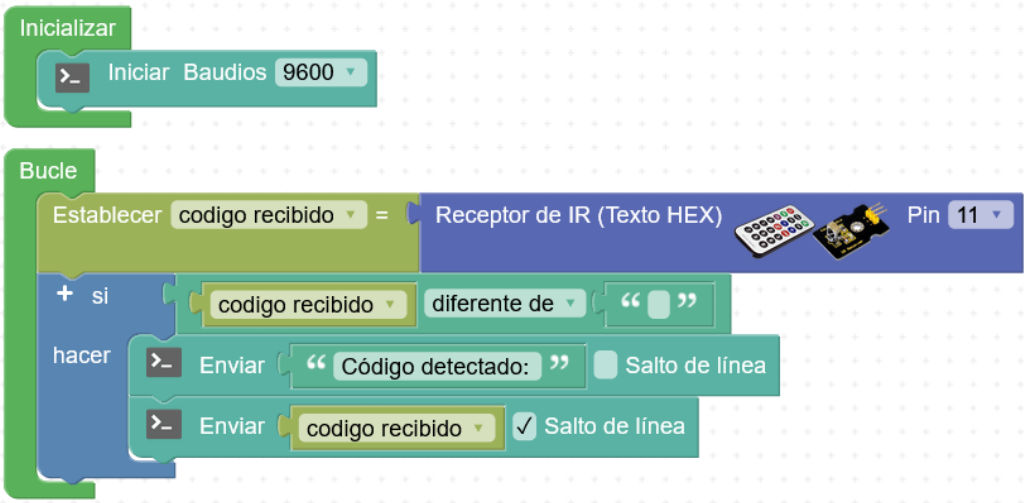
El mando que viene incluido con el sensor es un mando genérico con protocolo RC5:



Conexiones:



Programa para "capturar" códigos de mandos (prueba con el mando de la TV, aire acondicionado, ...) y mostrarlos por la consola:



Para cada tecla que queramos usar de un mando a distancia debemos apuntar los códigos recibidos:

(Asegúrate de pulsar varias veces y ver el código que normalmente se repite y descarta errores. Los códigos tiop FFFFFFF son para indicar que la tecla no se ha soltado)

Tecla 1: 00FF6897

Tecla 2: 00FF9867

Tecla 3: 00FFB04F

...

Tecla flecha arriba: 00FF629D

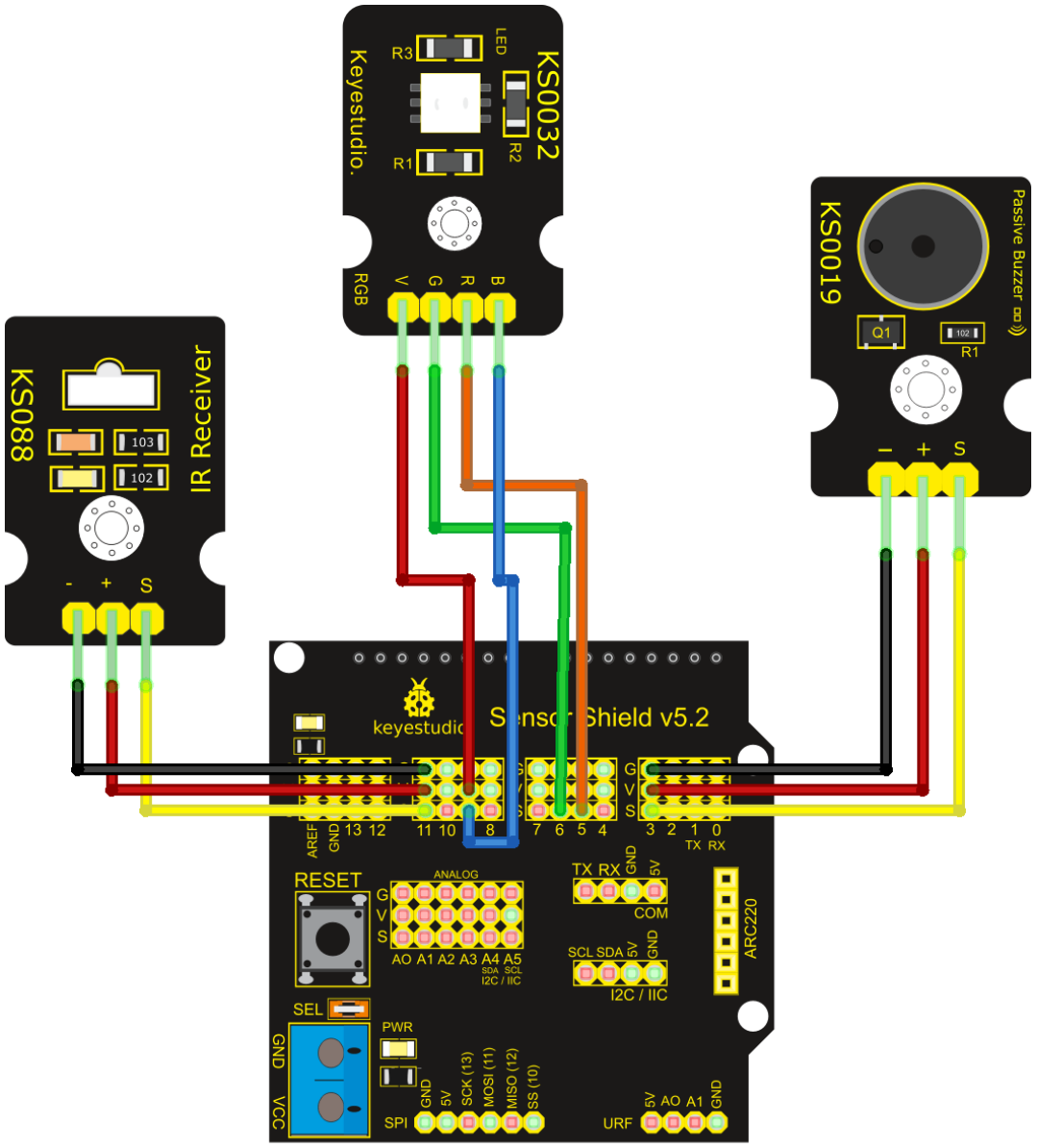
Tecla OK: 00FF02FD

...

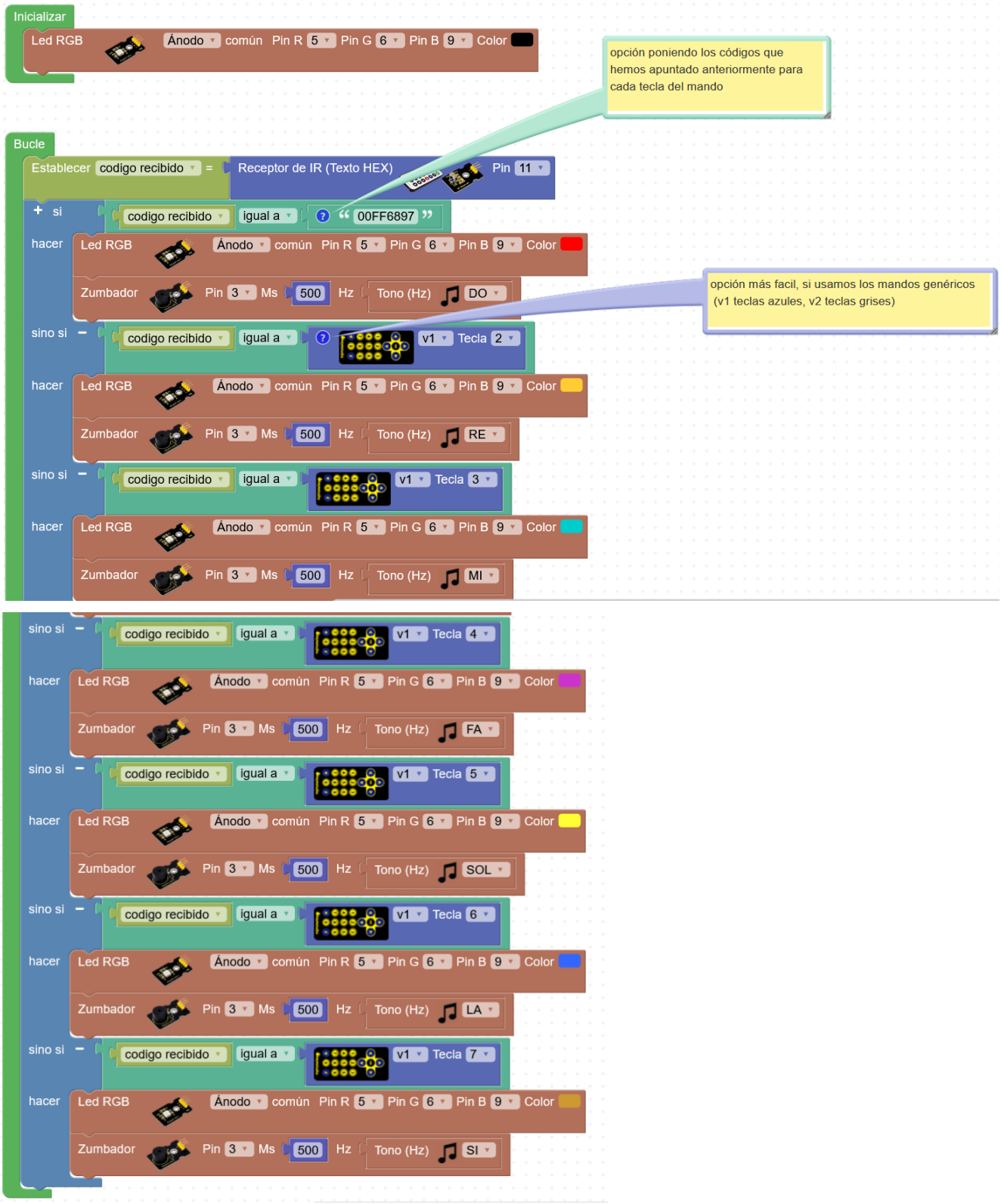
Piano IR

Cada tecla del mando a distancia hará sonar una nota musical diferente en el zumbador y pondrá el led RGB de un color distinto.

Conexiones:



Progama:


Inicializar
Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
opción poniendo los códigos que hemos apuntado anteriormente para cada tecla del mando
Bucle
Establecer codigo recibido = Receptor de IR (Texto HEX) Pin 11
si codigo recibido igual a "00FF6897"
hacer Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
Zumbador Pin 3 Ms 500 Hz Tono (Hz) DO
opción más facil, si usamos los mandos genéricos (v1 teclas azules, v2 teclas grises)
sino si codigo recibido igual a v1 Tecla 2
hacer Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
Zumbador Pin 3 Ms 500 Hz Tono (Hz) RE
sino si codigo recibido igual a v1 Tecla 3
hacer Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
Zumbador Pin 3 Ms 500 Hz Tono (Hz) MI
sino si codigo recibido igual a v1 Tecla 4
hacer Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
Zumbador Pin 3 Ms 500 Hz Tono (Hz) FA
sino si codigo recibido igual a v1 Tecla 5
hacer Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
Zumbador Pin 3 Ms 500 Hz Tono (Hz) SOL
sino si codigo recibido igual a v1 Tecla 6
hacer Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
Zumbador Pin 3 Ms 500 Hz Tono (Hz) LA
sino si codigo recibido igual a v1 Tecla 7
hacer Led RGB Ánodo común Pin R 5 Pin G 6 Pin B 9 Color
Zumbador Pin 3 Ms 500 Hz Tono (Hz) SI

Servo IR

Control de la posición del servo con el mando IR.

Tecla 1: posición 0º

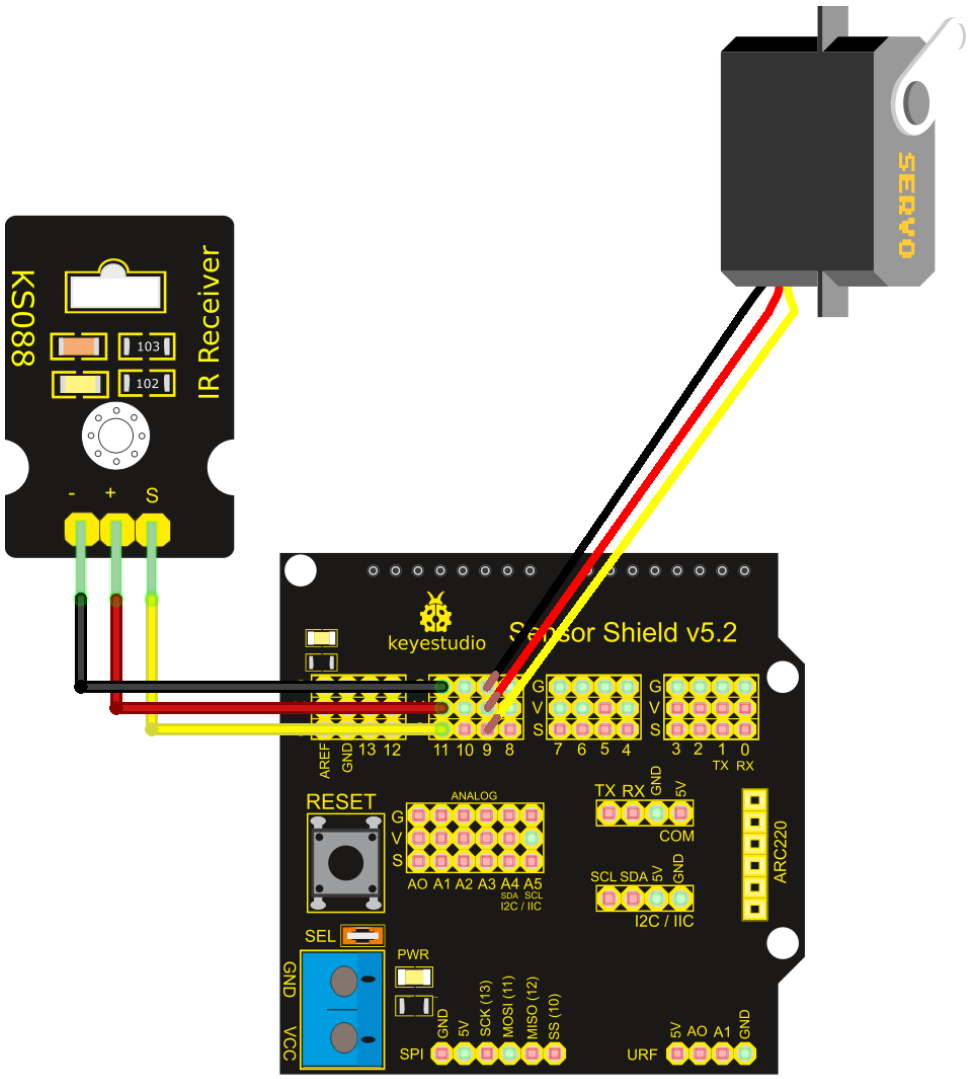
Tecla 2: posición 90º

Tecla 3: posición 180º

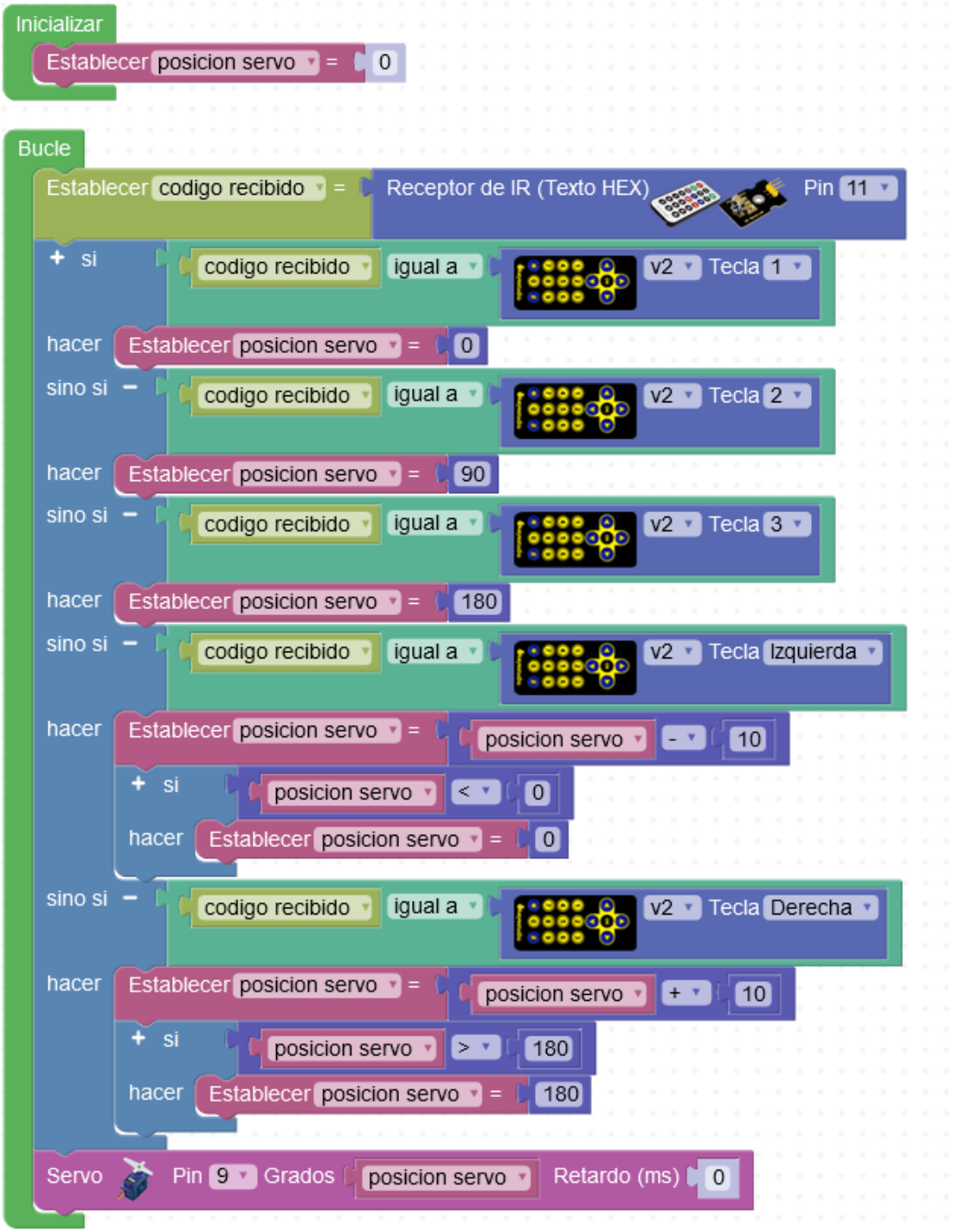
Tecla flecha izquierda: disminuye 10º

Tecla flecha derecha: aumenta 10º

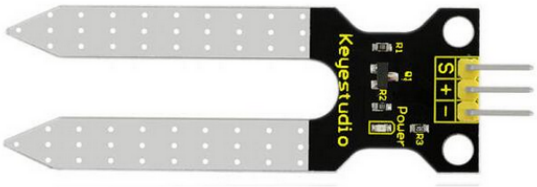
Conexiones:



Programa:


Inicializar
Establecer posicion servo = 0
Bucle
Establecer codigo recibido = Receptor de IR (Texto HEX) Pin 11
si codigo recibido igual a v2 Tecla 1
hacer Establecer posicion servo = 0
sino si codigo recibido igual a v2 Tecla 2
hacer Establecer posicion servo = 90
sino si codigo recibido igual a v2 Tecla 3
hacer Establecer posicion servo = 180
sino si codigo recibido igual a v2 Tecla Izquierda
hacer Establecer posicion servo = posicion servo - 10
si posicion servo < 0
hacer Establecer posicion servo = 0
sino si codigo recibido igual a v2 Tecla Derecha
hacer Establecer posicion servo = posicion servo + 10
si posicion servo > 180
hacer Establecer posicion servo = 180
Servo Pin 9 Grados posicion servo Retardo (ms) 0

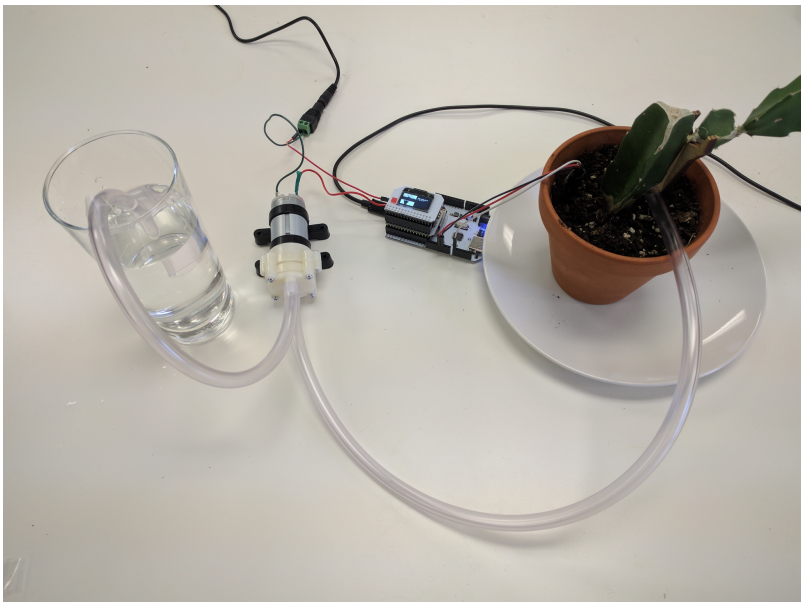
Sensor humedad suelo - Riego automático



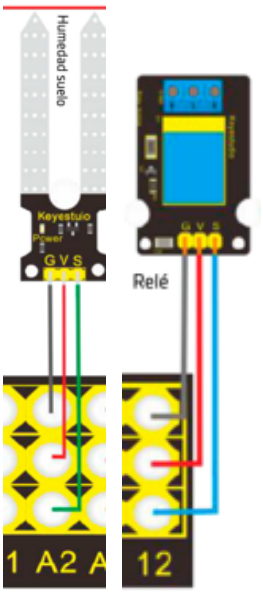
https://wiki.keyestudio.com/Ks0049_keyestudio_Soil_Humidity_Sensor

Utilizaremos el sensor para detectar un nivel bajo de humedad y activar una bomba de riego automáticamente (activación de la bomba mediante un relé).

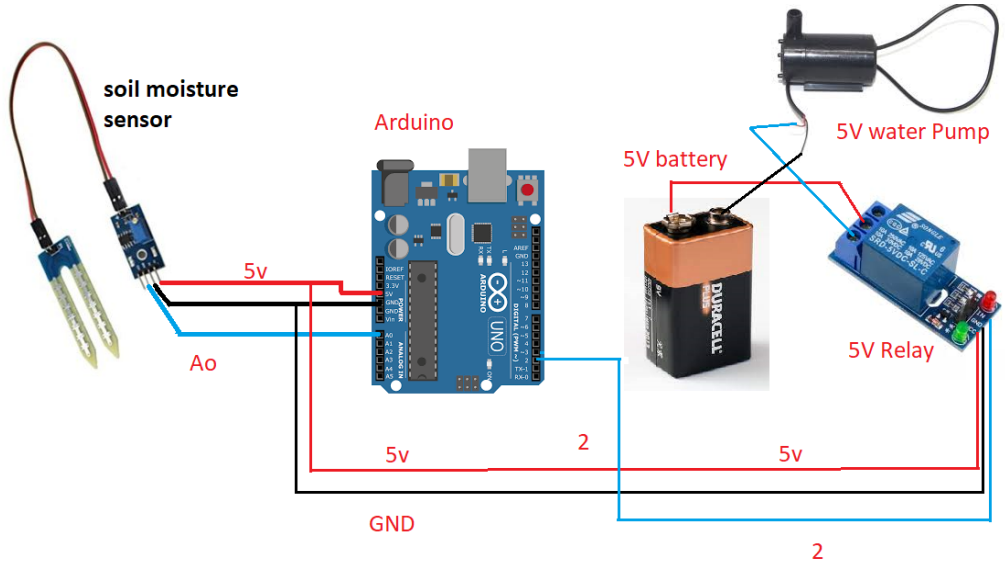
Ejemplo de bomba de agua:



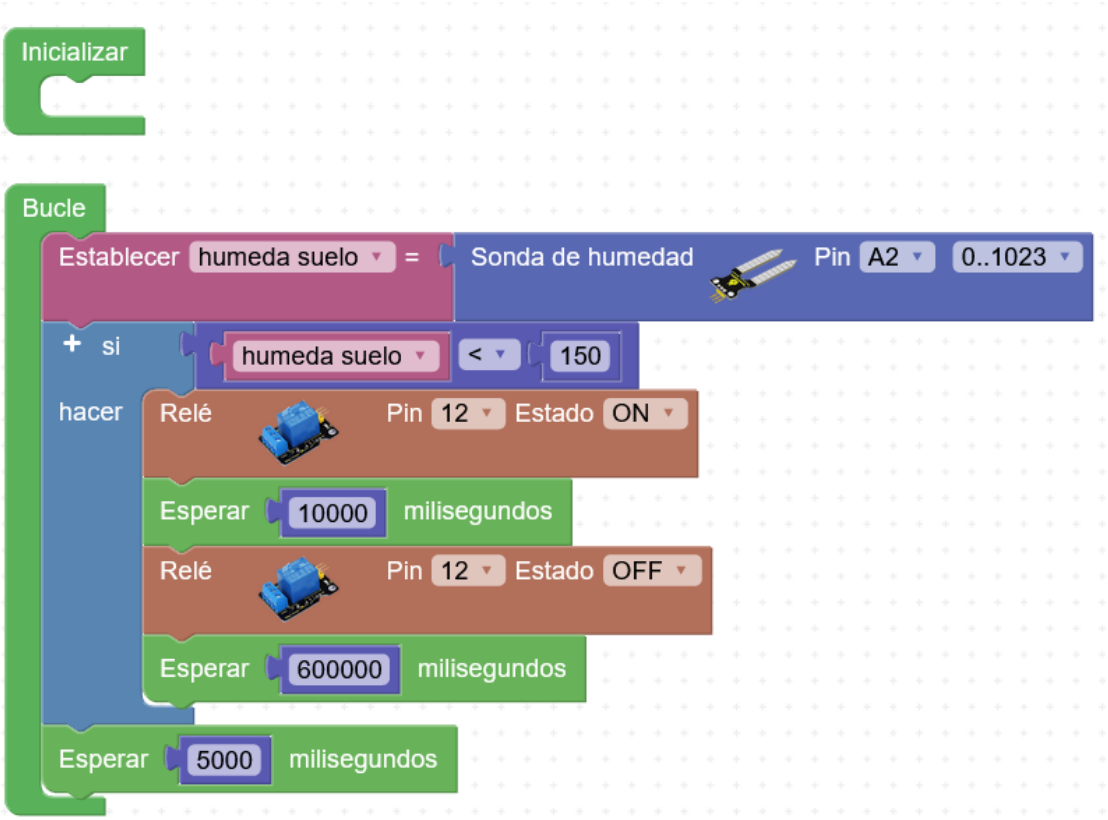
Conexiones:



Ejemplo de esquema completo:



Programa:



Sensor magnético (Velocímetro bicicleta)

Sensor de campo magnético (efecto Hall):

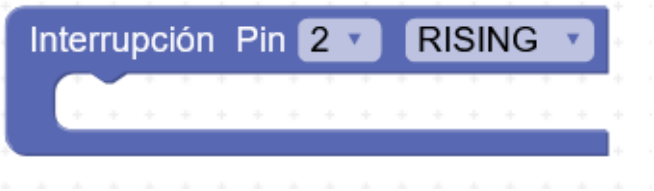
https://wiki.keyestudio.com/Ks0020_keyestudio_Hall_Magnetic_Sensor

El sensor se actica o desactiva al acercar un campo magnético (imán).

Necesitamos saber cuando se activa o desactiva (flanco ascendente o descendente de la activación), para eso usaremos un bloque que genera un evento externo (interrupción) cuando se detecta un cambio en el sensor (además es capaz de detectar cambios a velocidades muy altas)

Flanco de subida o de bajada al cambiar el estado de un sensor:

Para detectar el evento del flanco de subida (de 0 a 5v ,al acercar un imán al senor):



Siguiendo este sistema, vamos a realizar un velocímetro de bicicleta. Cada vez que el sensor genera un flanco de subida quiere decir que la rueda ha dado una vuelta completa, sabiendo el tiempo que ha tardado desde la última detección y la longitud de la circunferencia de la rueda podemos calcular la velocidad:

Cálculo de la velocidad (en Km/h):

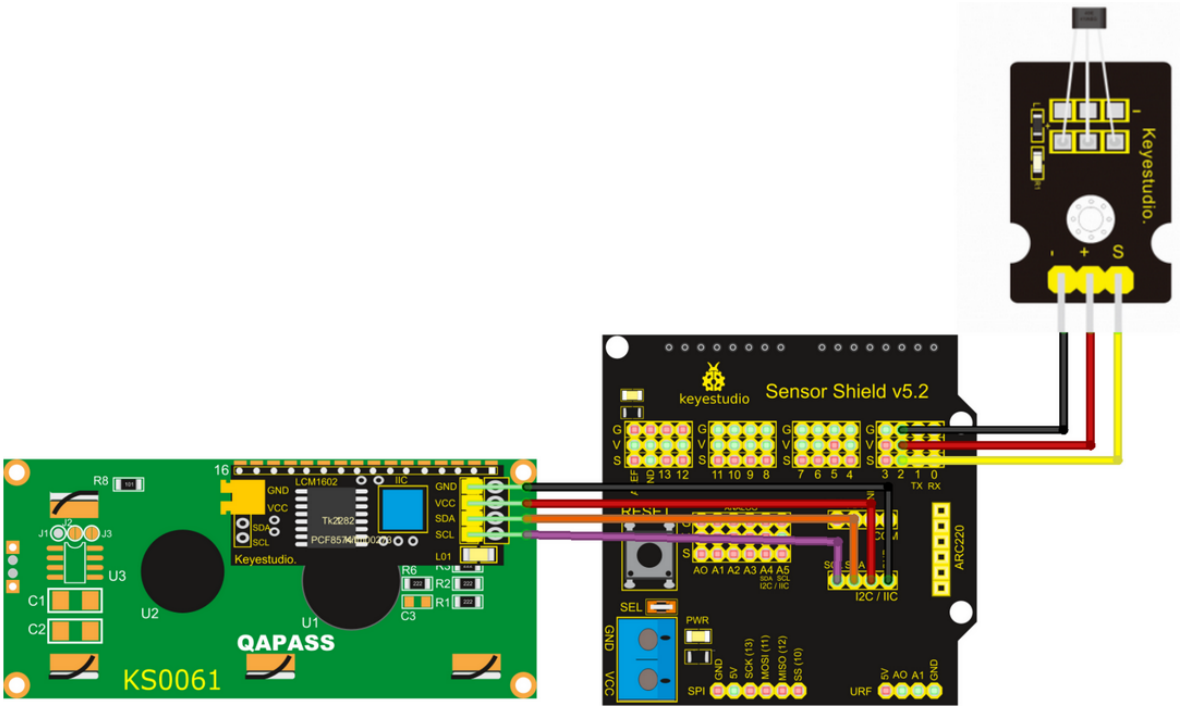
velocidad = (longitud_rueda_metros / 1000) / (tiempo_transcurrido_ms / 1000 / 3600)

Simplificando:

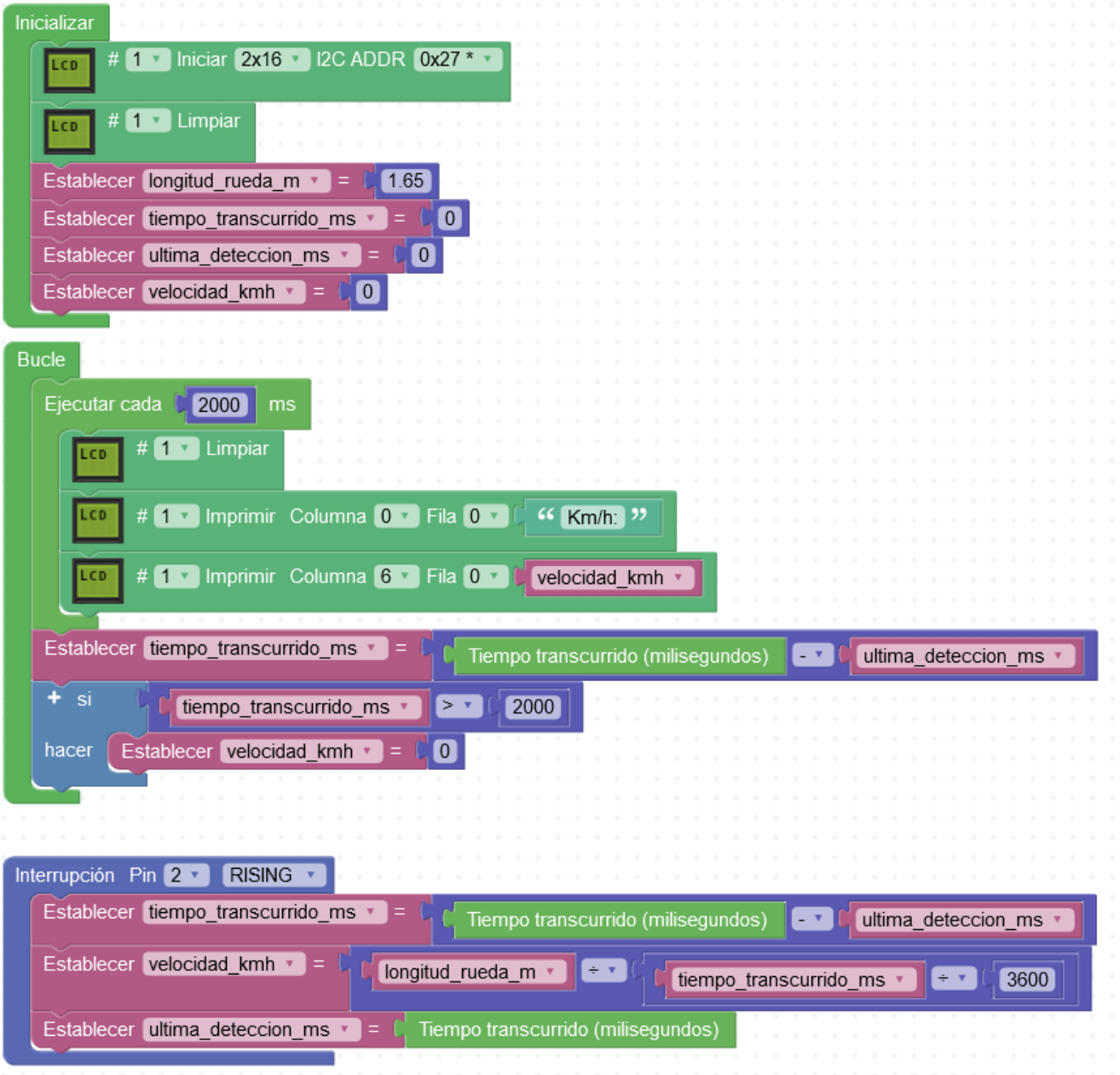
velocidad = longitud_rueda_metros / (tiempo_transcurrido_ms / 3600)

Para visualizar la información se usará la pantalla LCD.

Conexiones:

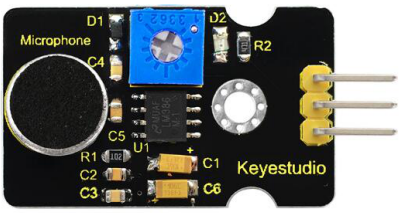


Programa:



Sensor de sonido

Permite, mediante un micrófono, detectar el nivel de sonido ambiente.

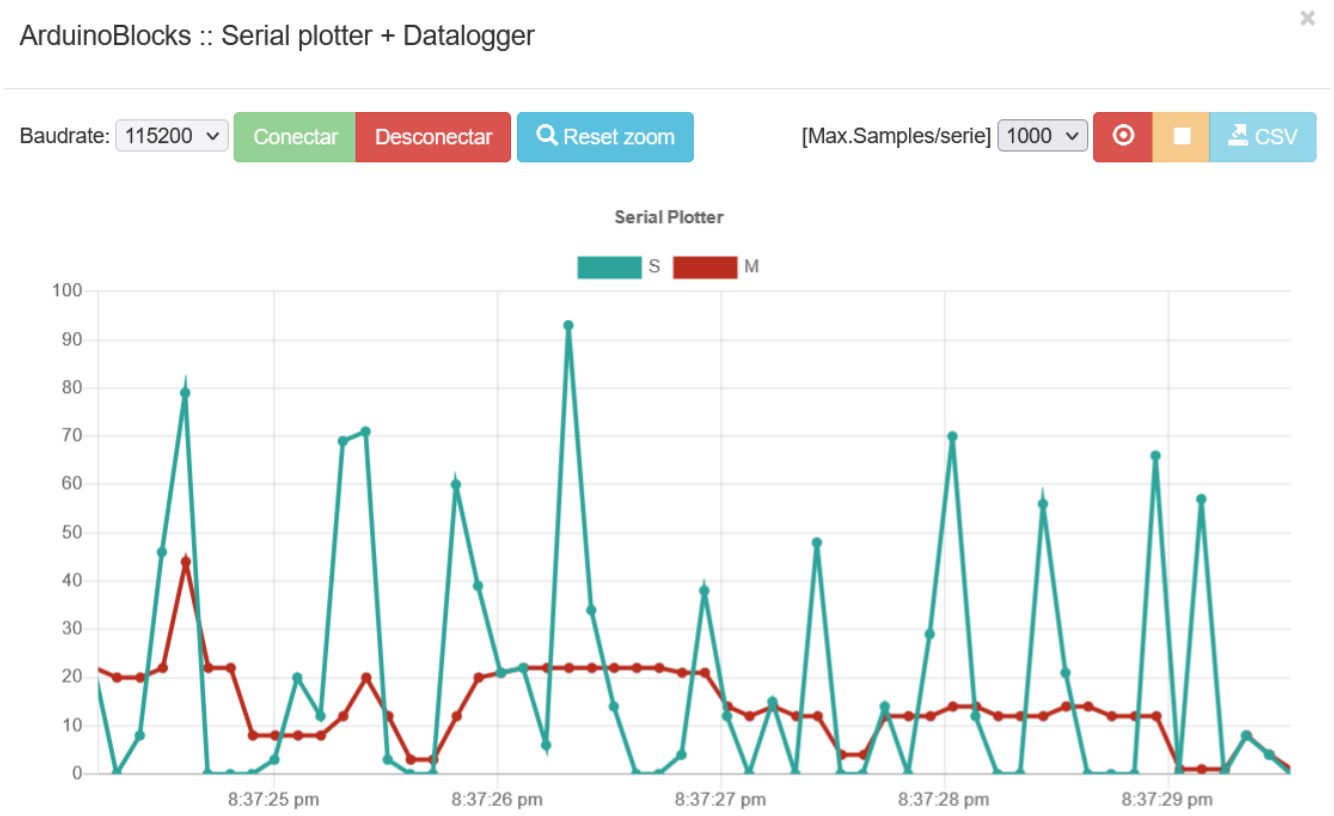


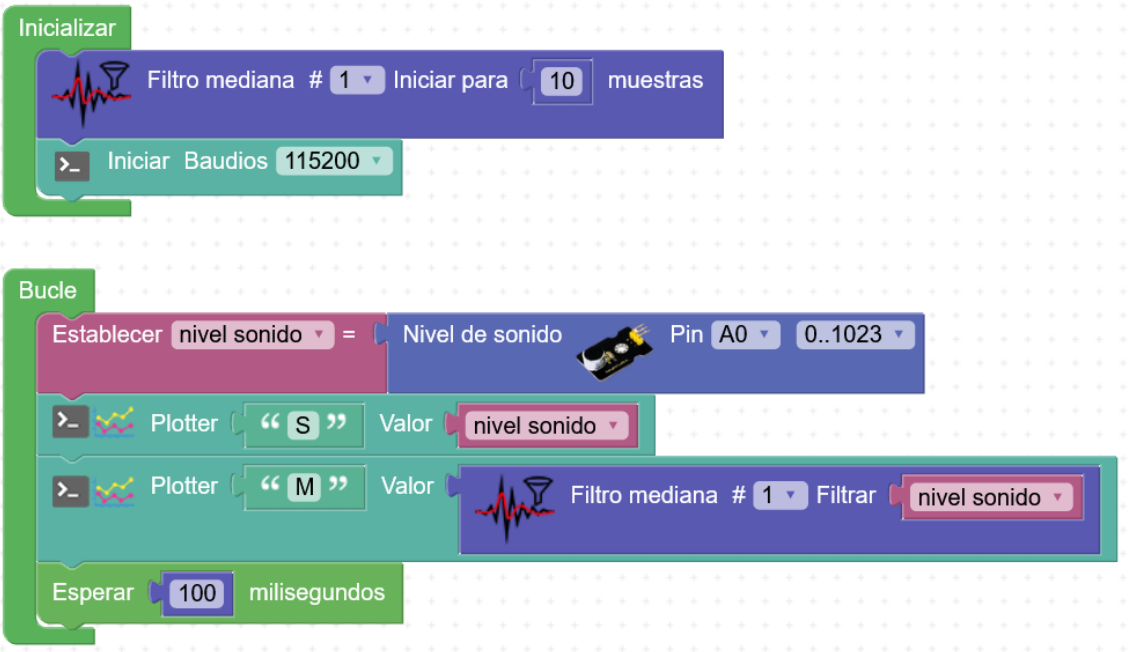
https://wiki.keyestudio.com/KS0035_Microphone_Sound_Sensor_with_Potentiometer

Filtro de la señal de sonido (filtro mediana):

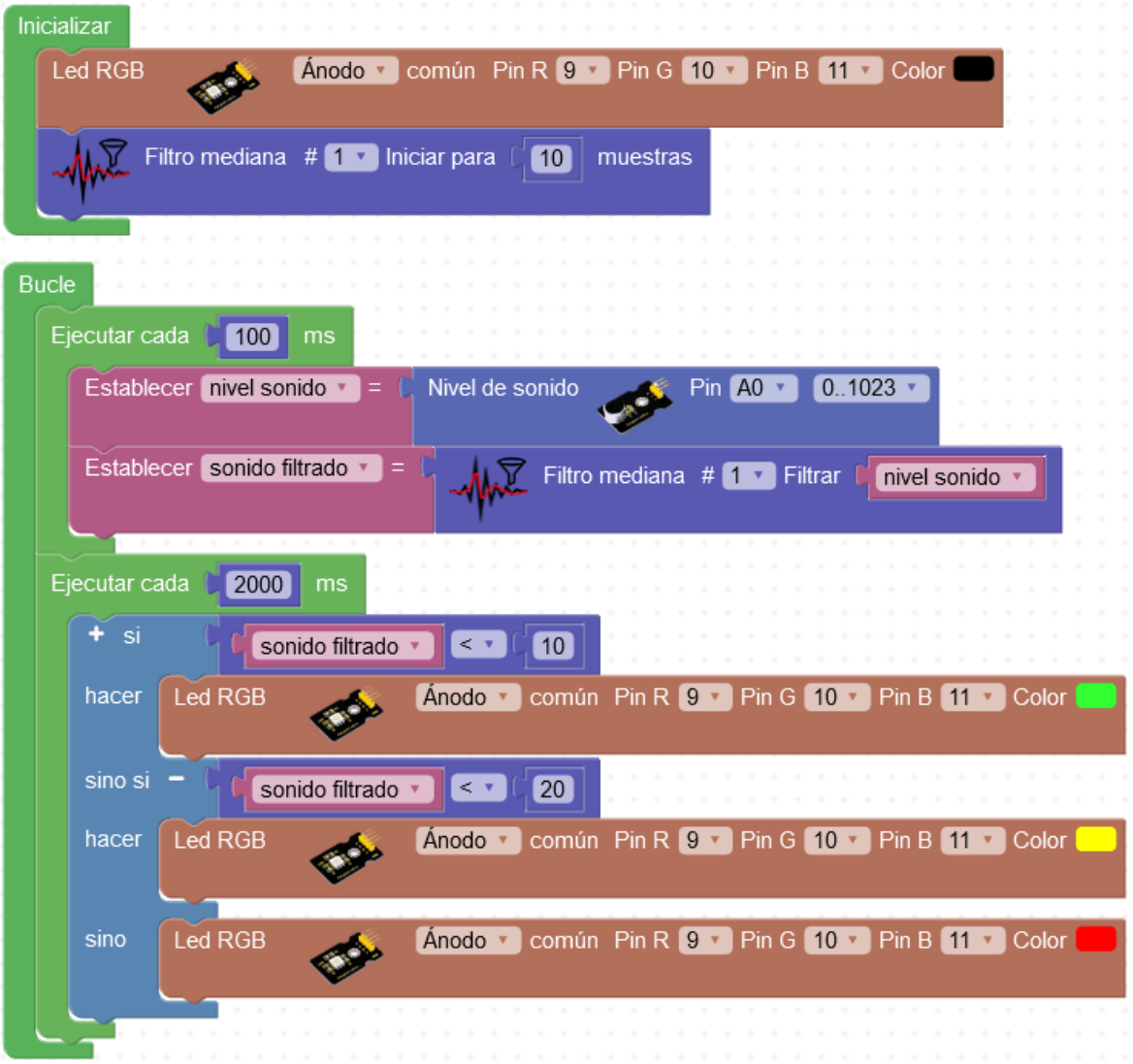
S -> sensor de sonido puro

M -> señal filtrada



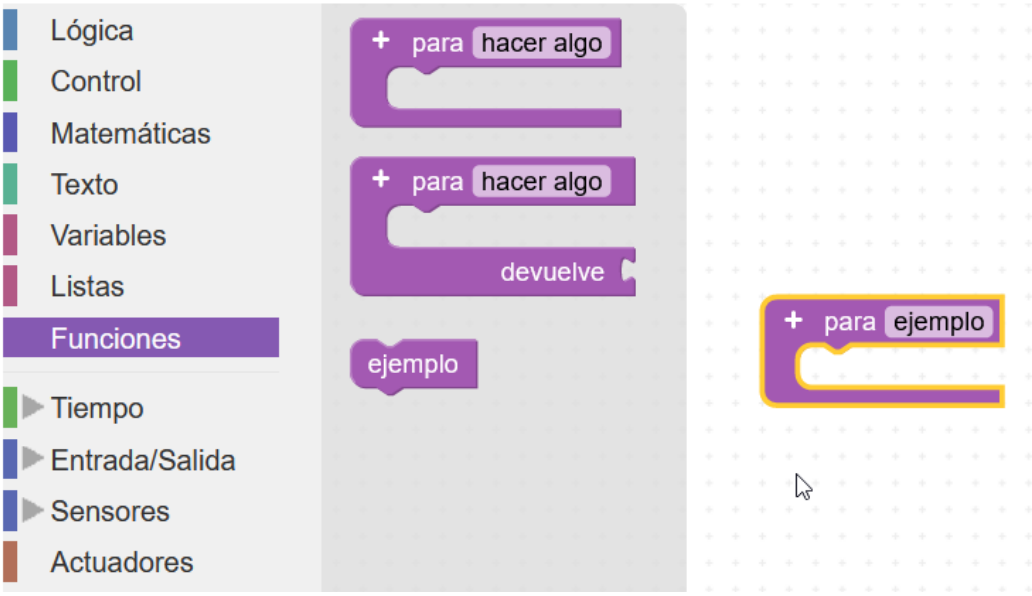


Semáforo de sonido:

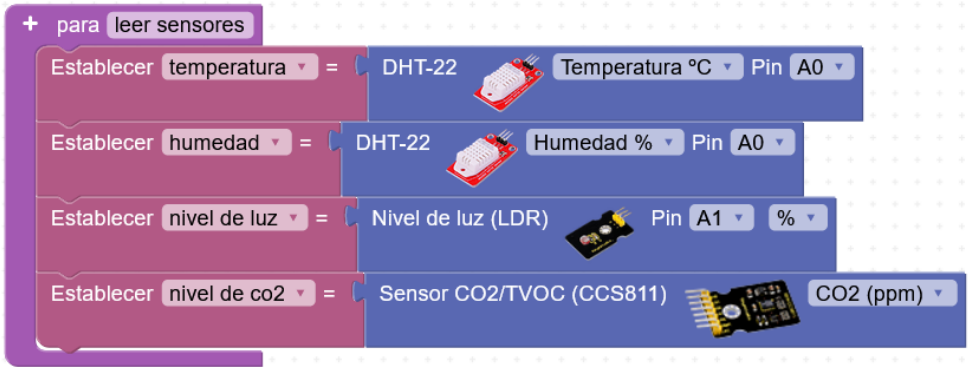


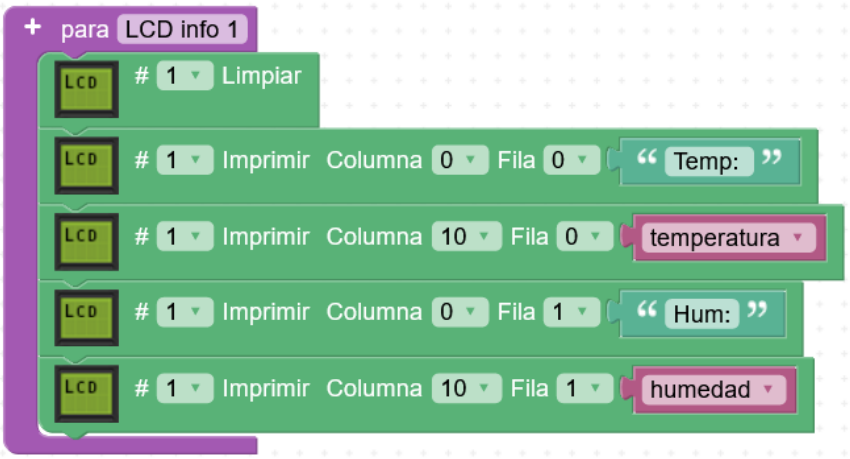
Funciones

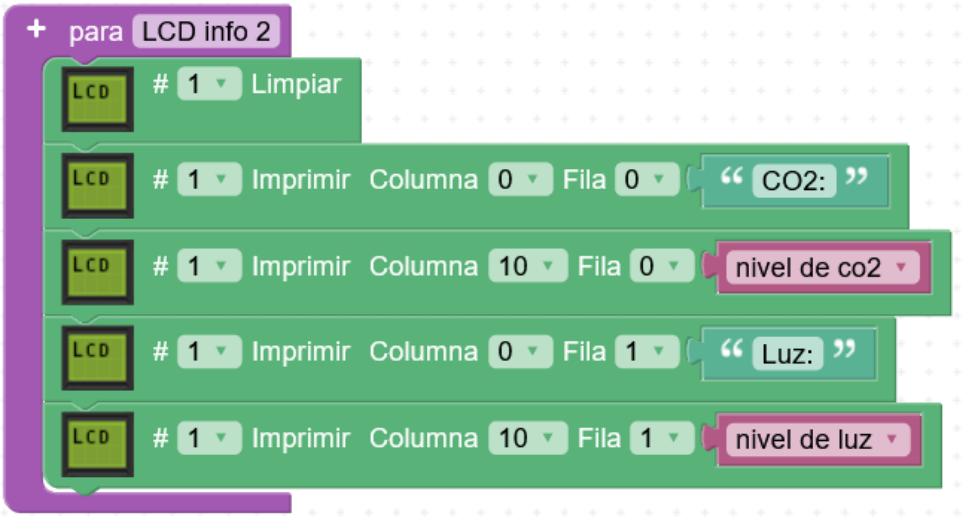
A la hora de realizar programas grandes es aconsejable dividir en tareas el programa. Podemos usar funciones para crear bloques de código que realice alguna función en concreto, y así "llamarlo" o "ejecutarlo" de una forma más clara y sencilla.

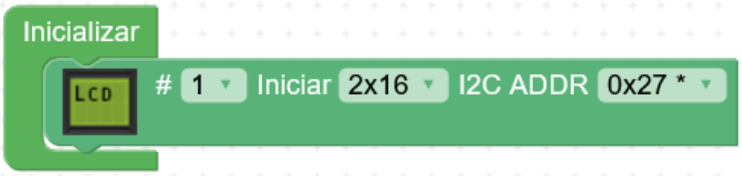


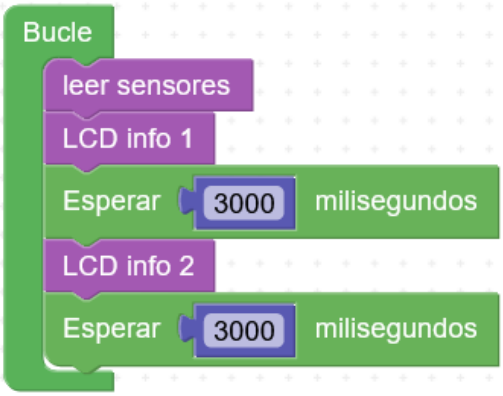
Ejemplo 1:



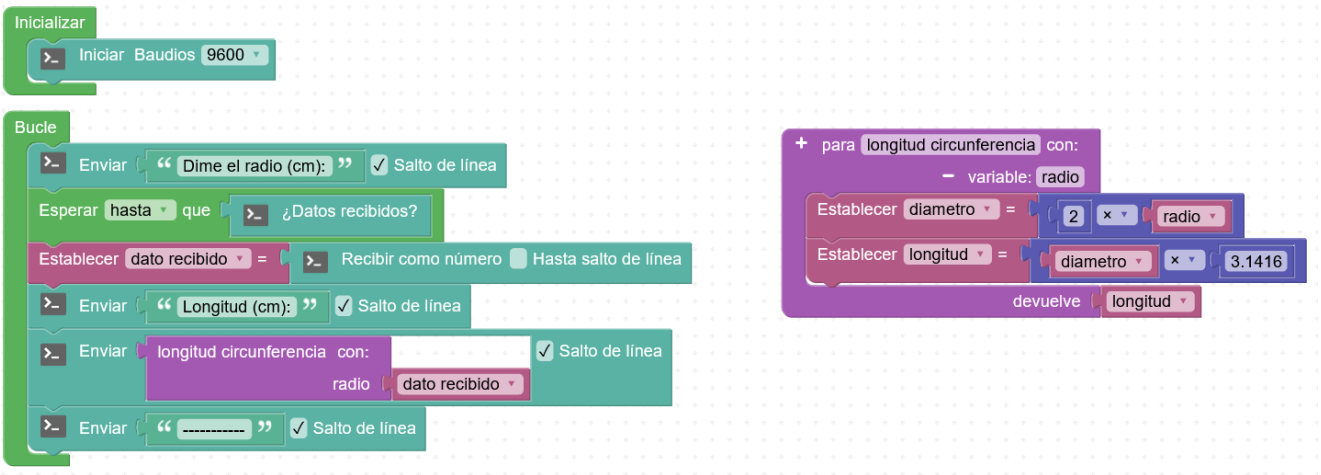



para LCD info 2
LCD # 1 Limpiar
LCD # 1 Imprimir Columna 0 Fila 0 " CO2: "
LCD # 1 Imprimir Columna 10 Fila 0 nivel de co2
LCD # 1 Imprimir Columna 0 Fila 1 " Luz: "
LCD # 1 Imprimir Columna 10 Fila 1 nivel de luz


Inicializar
LCD # 1 Iniciar 2x16 I2C ADDR 0x27 *


Bucle
leer sensores
LCD info 1
Esperar 3000 milisegundos
LCD info 2
Esperar 3000 milisegundos

Ejemplo 2:

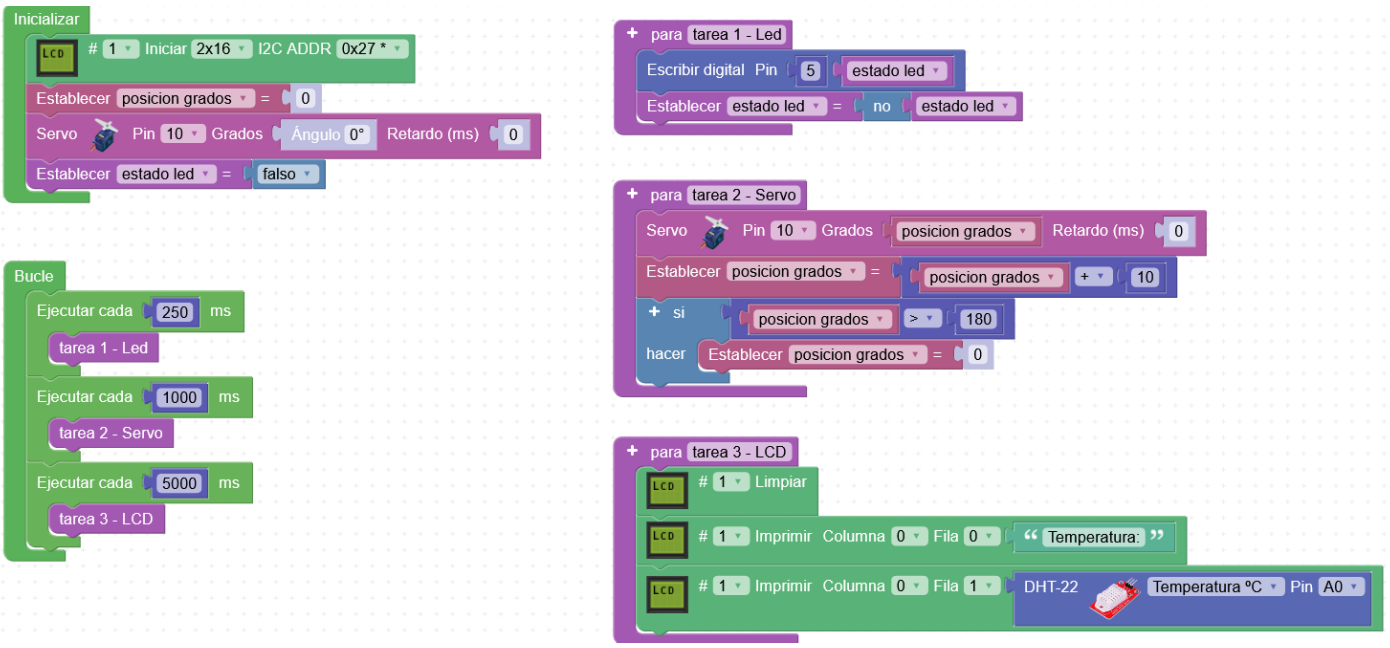

Inicializar
Iniciar Baudios 9600
Bucle
Enviar " Dime el radio (cm): " Salto de línea
Esperar hasta que ¿Datos recibidos?
Establecer dato recibido = Recibir como número Hasta salto de línea
Enviar " Longitud (cm): " Salto de línea
Enviar longitud circunferencia con: radio dato recibido Salto de línea
Enviar " ---------- " Salto de línea
para longitud circunferencia con:
variable: radio
Establecer diametro = 2 × radio
Establecer longitud = diametro × 3.1416
devuelve longitud

Tareas / Multitarea

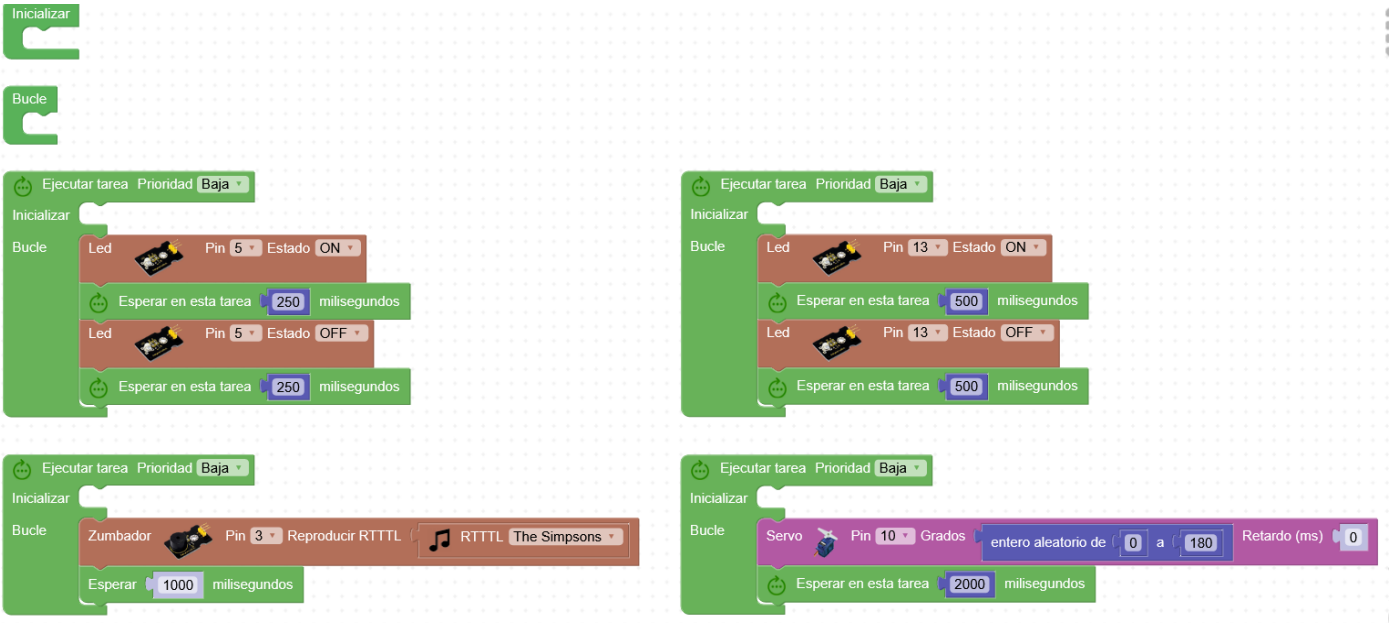
Arduino no implementa un sistema multitarea, sólo ejecuta una cosa a la vez, sin embargo mediante técnicas software podemos simular el ejecutar varias cosas a la vez, o por lo menos que compartan el tiempo y parezca que lo hacen a la vez.

1) Método "casero".

La idea es desglosar todo en tareas, cada tarea debe ser rápida y NO BLOQUEAR la ejecución (PROHIBIDO LOS BLOQUES "esperar" y los bucles extra largos o que se queden esperando una condición)



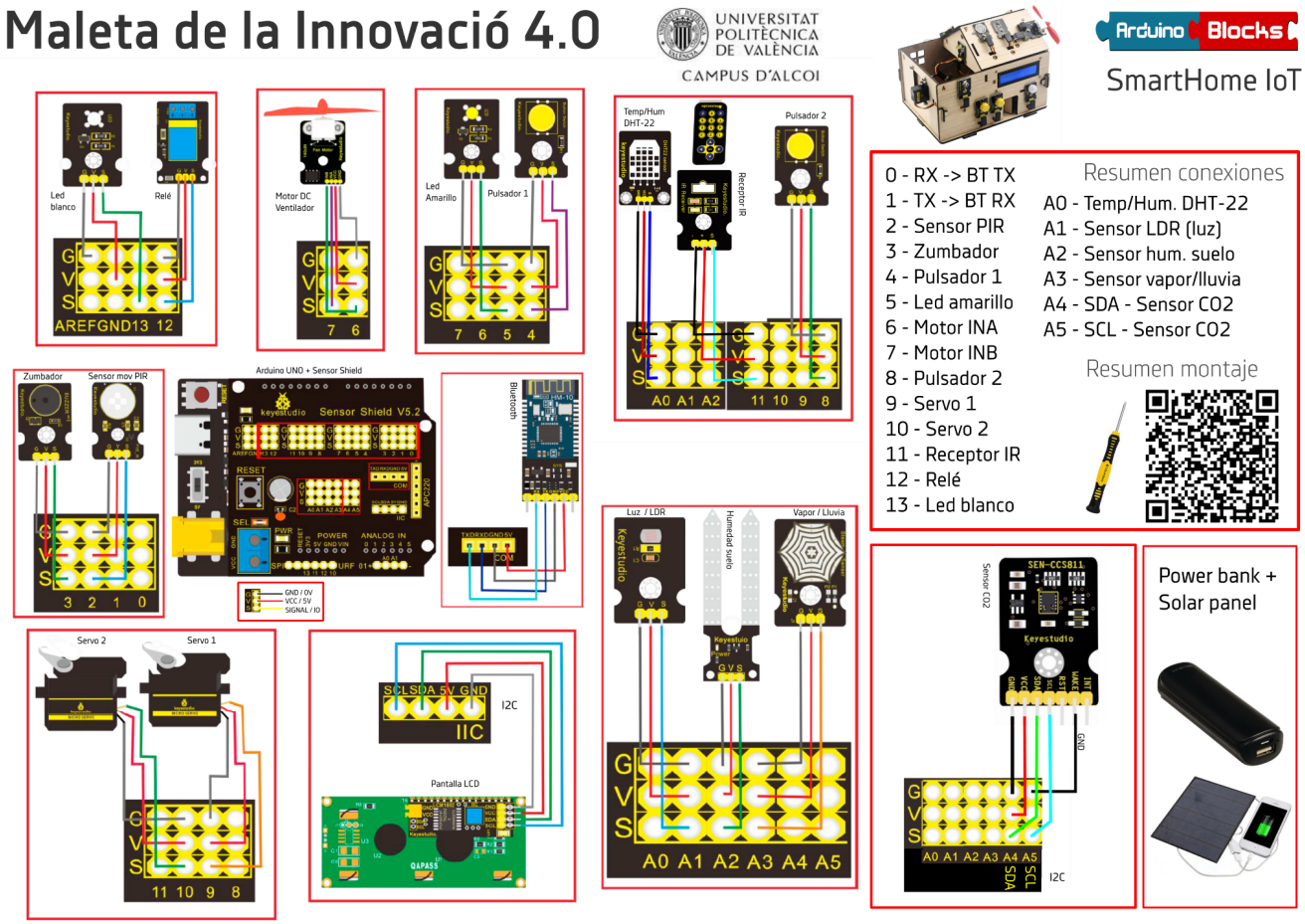
2) Método usando la librería "multitask freeRTOS" un poco limitada en Arduino pero que podemos hacer funcionar. https://drive.google.com/file/d/1r-oo8KUpNBySFMyEHYkjaltZV9eg0NPV/view



SmartHome: conexionado + programación básica

https://wiki.keyestudio.com/KS0085_Keyestudio_Smart_Home_Kit_for_Arduino#Assembled_Guide

Conexiones completas:



Funcionalidades a implementar en el programa para hacer la casa autónoma:

-Mensaje inicial de bienvenida en LCD

-Control con mando IR:

Tecla arriba -> ventilador ON

Tecla abajo -> ventilador OFF

Teclas 1,2,3 -> melodías RTTTL

-Pulsador 1 -> abrir/cerrar ventana

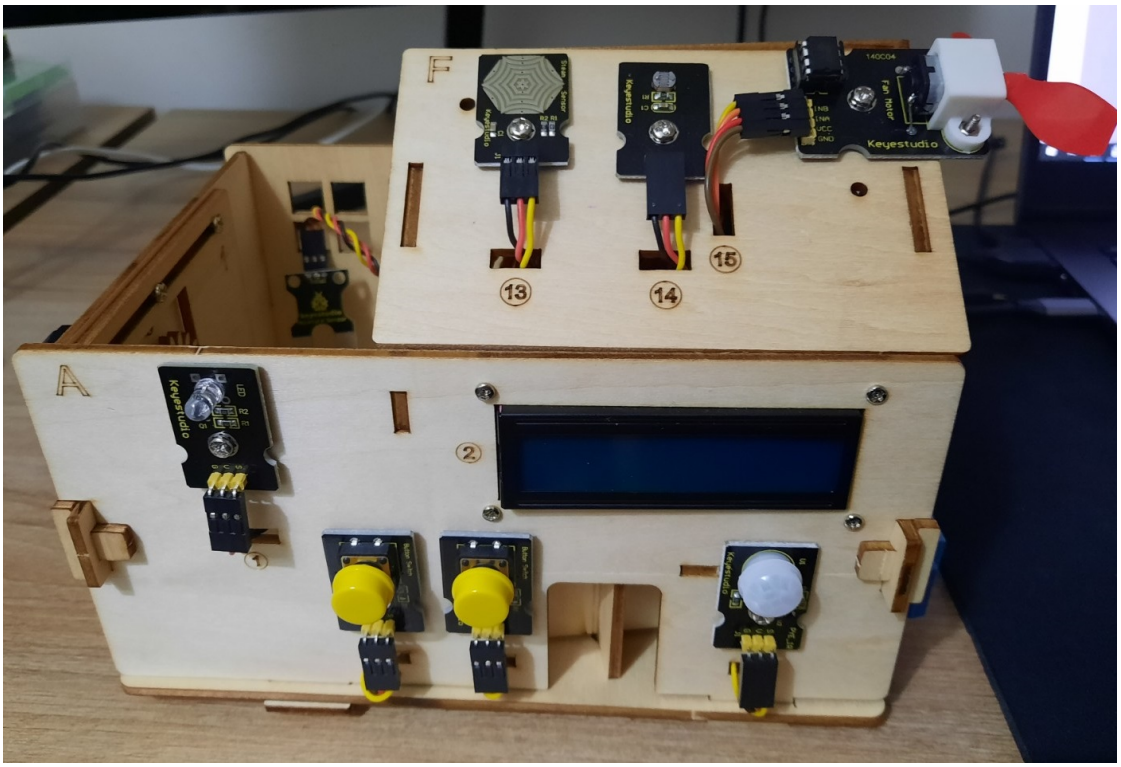
-Pulsador 2 -> abrir/cerrar puerta

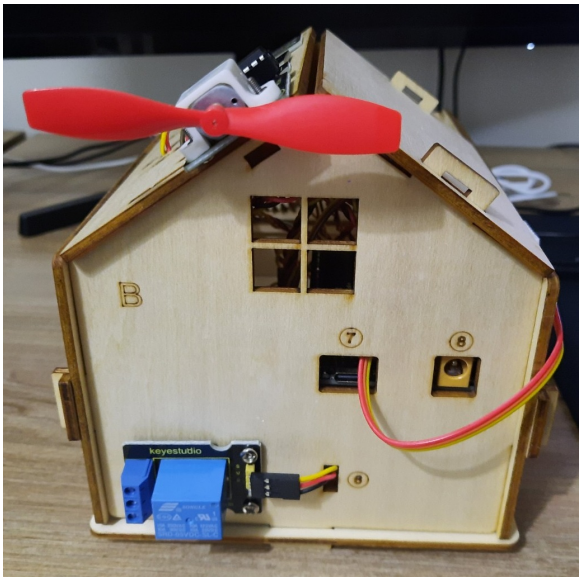
-El sensor de movimiento encenderá automáticamente el led blanco

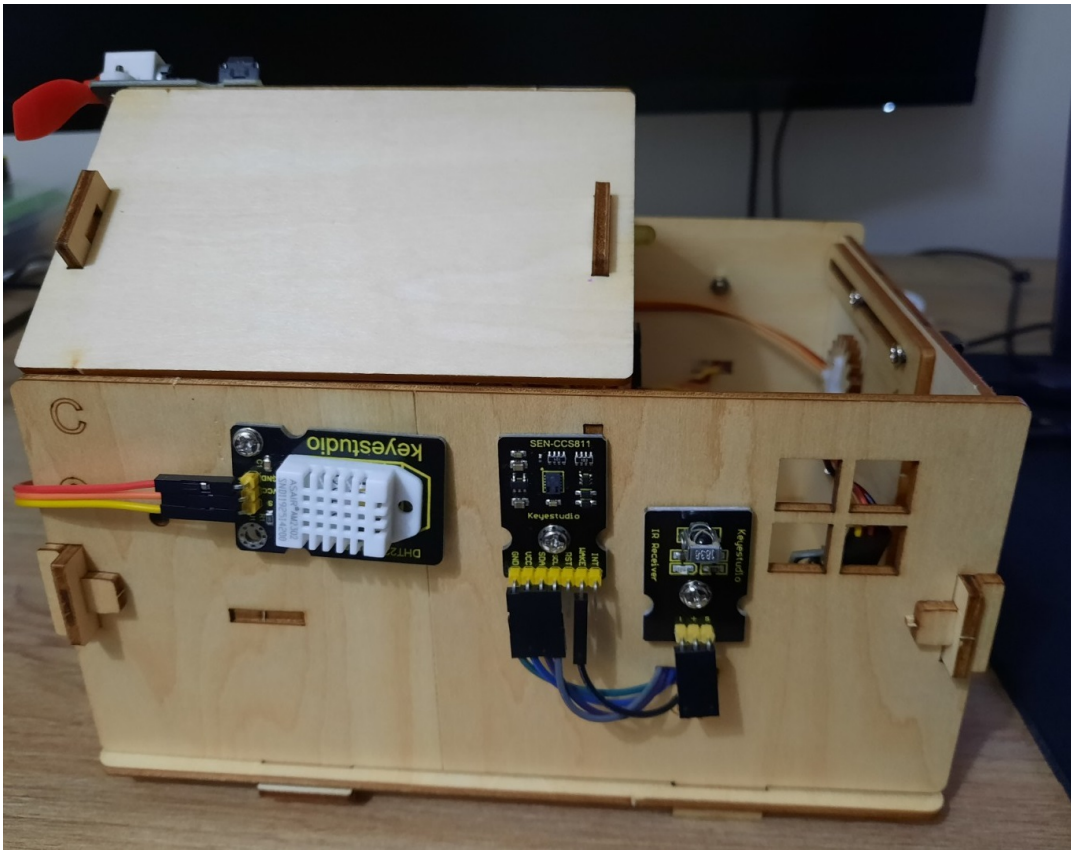
-Si el sensor de humedad de suelo detecta poca humedad y no llueve , se activa el relé para activar el riego

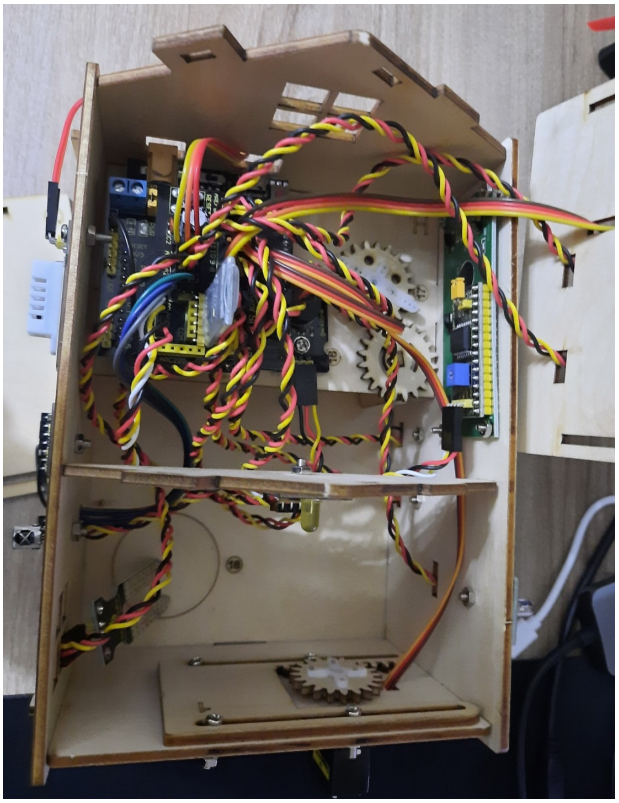
-Si el nivel de luz es bajo, se enciende automáticamente la luz amarilla

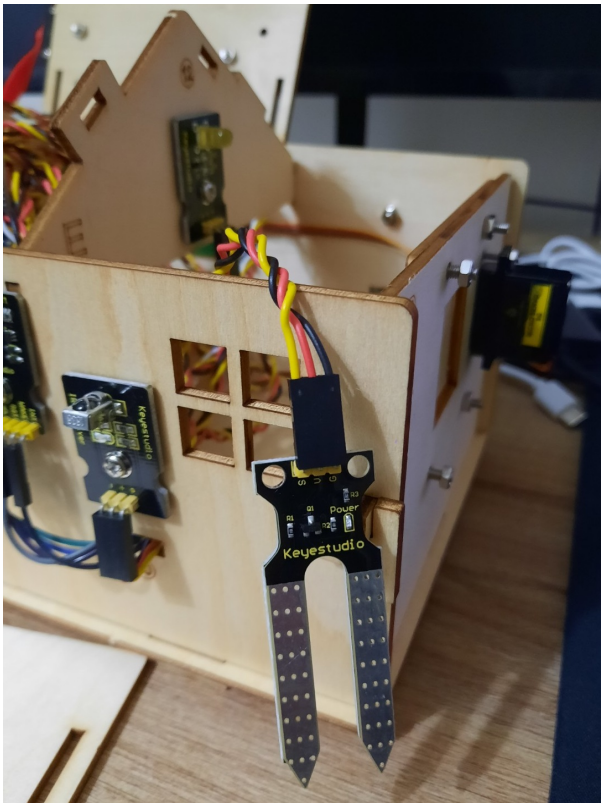
-Cada 5s se muestra por la pantalla LCD el nivel de CO2 , la temperatura y la humedad





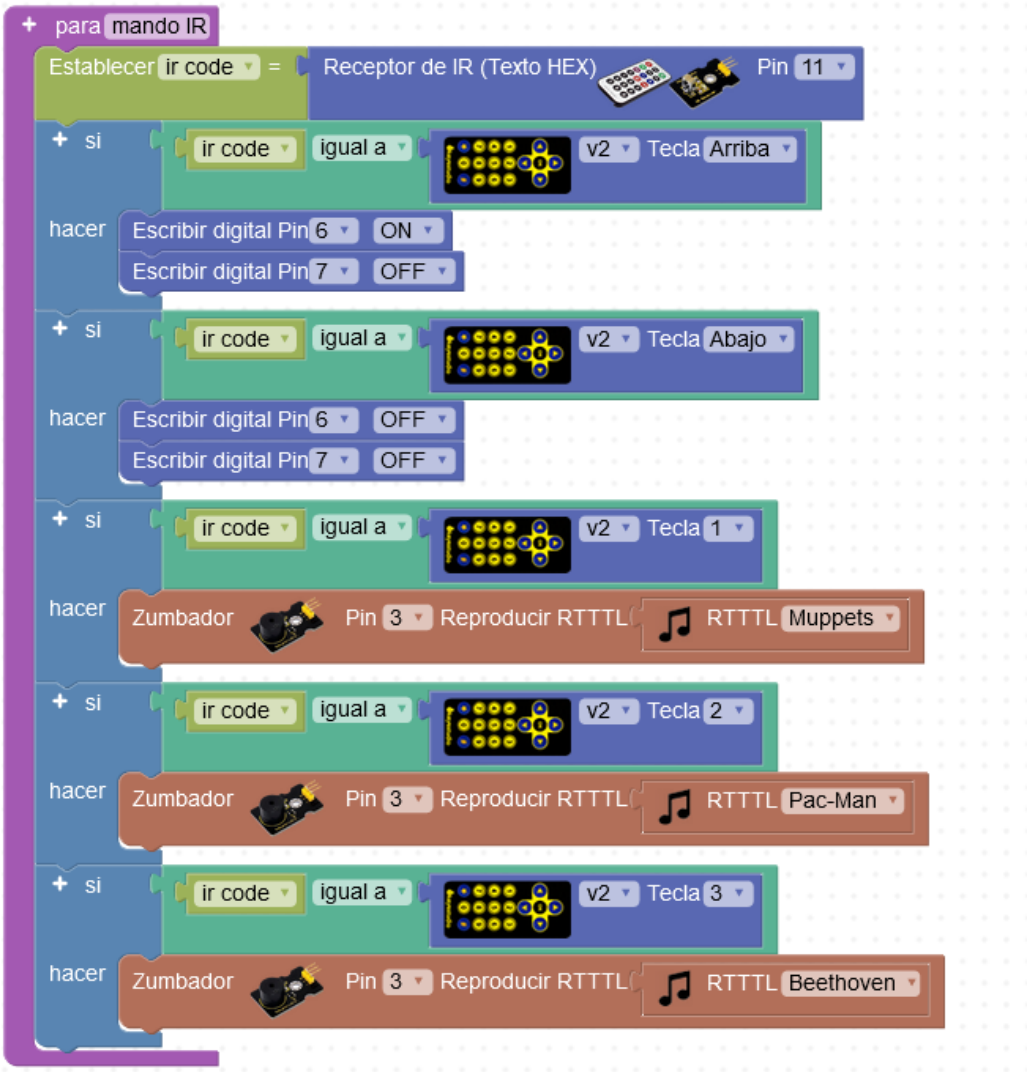

keyestudio
SEN-CCS811
Keyestudio
IR Receiver




Power
Keyestudio

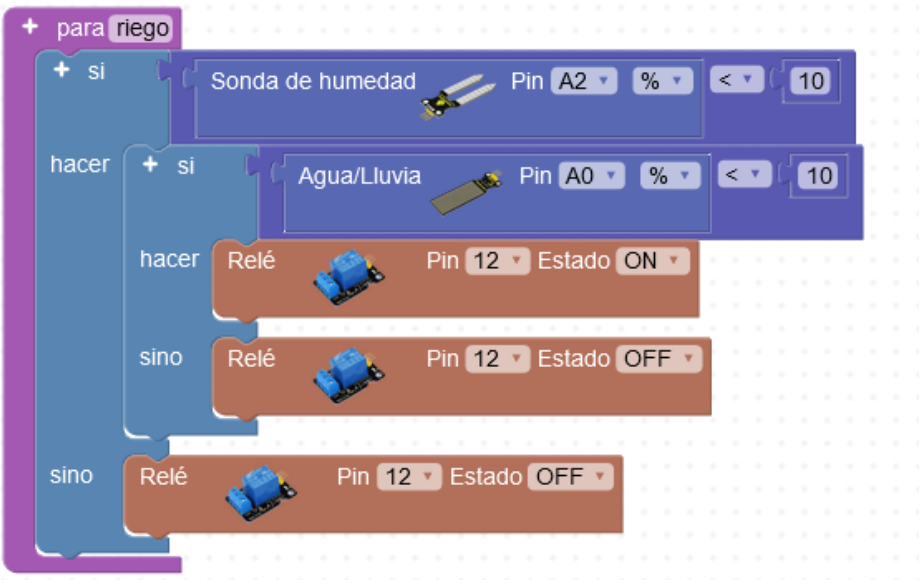
IMPORTANTE: Poner los servos en posición 90º antes de montarlos:

Programa:

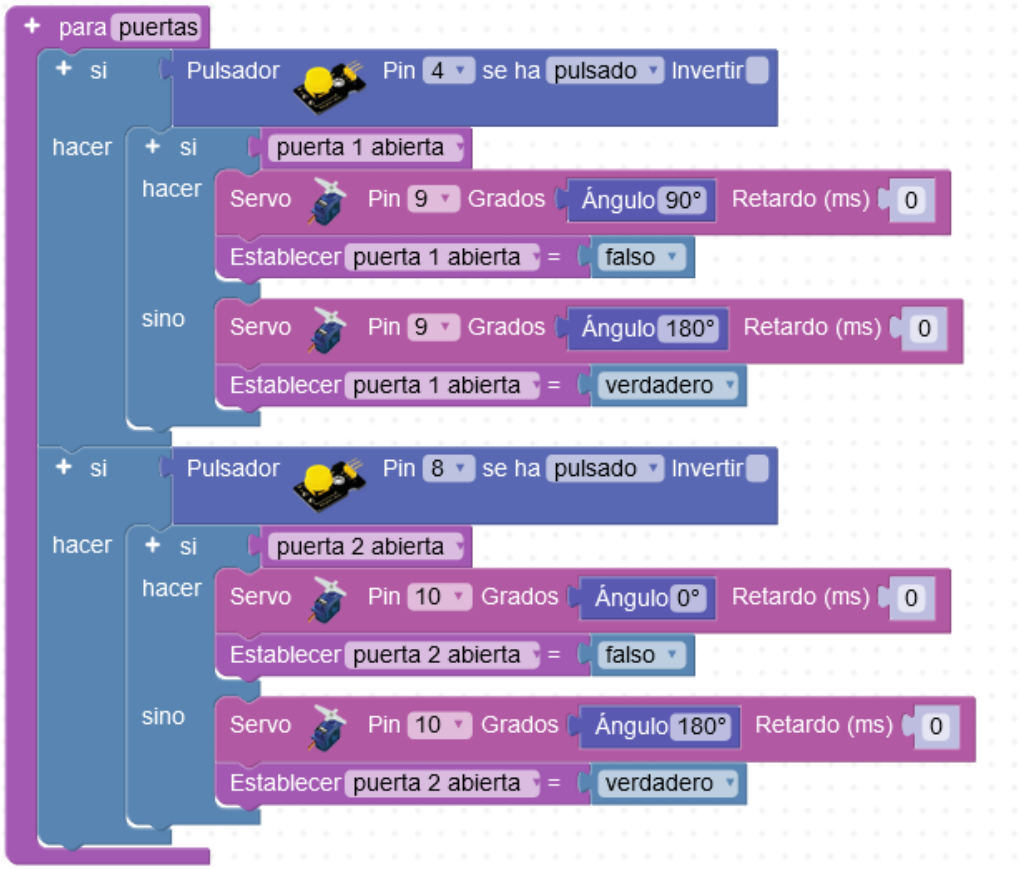


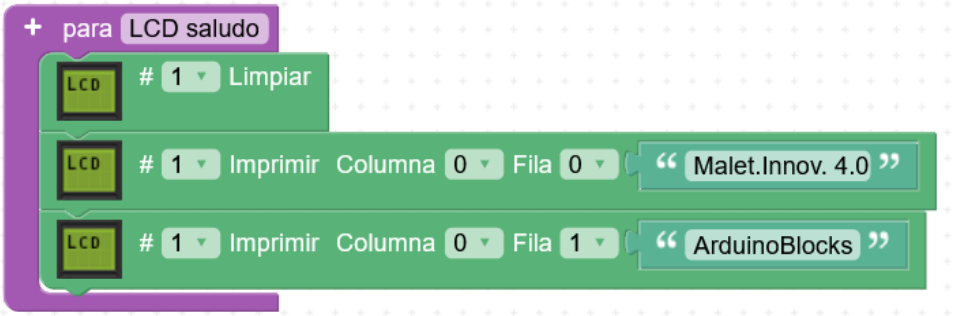

para movimiento
si
Detector de movimiento (PIR)
Pin 2
hacer
Led
Pin 13 Estado ON
sino
Led
Pin 13 Estado OFF

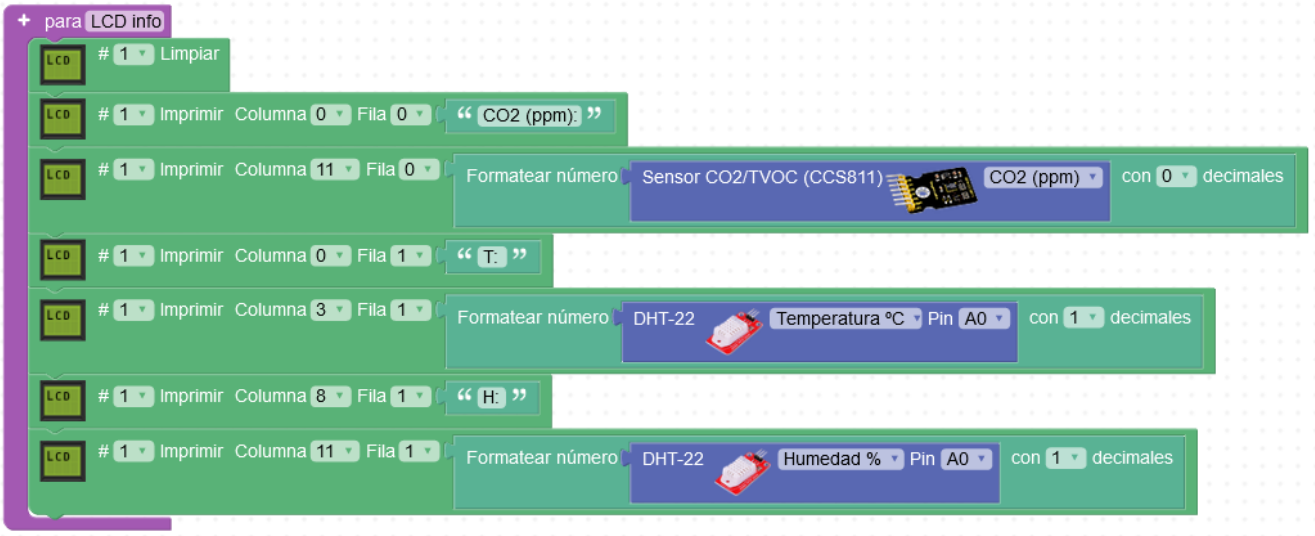

para luz ldr
si
Nivel de luz (LDR)
Pin A1 % < 20
hacer
Led
Pin 5 Estado ON
sino
Led
Pin 5 Estado OFF


para riego
si
Sonda de humedad
Pin A2 % < 10
hacer
si
Agua/Lluvia
Pin A0 % < 10
hacer
Relé
Pin 12 Estado ON
sino
Relé
Pin 12 Estado OFF
sino
Relé
Pin 12 Estado OFF


para iniciar
Establecer puerta 1 abierta = falso
Servo
Pin 9 Grados Ángulo 90° Retardo (ms) 0
Establecer puerta 2 abierta = falso
Servo
Pin 10 Grados Ángulo 0° Retardo (ms) 0
Establecer mov detectado = falso


para puertas
si
Pulsador Pin 4 se ha pulsado Invertir
hacer
si
puerta 1 abierta
hacer
Servo Pin 9 Grados Ángulo 90° Retardo (ms) 0
Establecer puerta 1 abierta = falso
sino
Servo Pin 9 Grados Ángulo 180° Retardo (ms) 0
Establecer puerta 1 abierta = verdadero
si
Pulsador Pin 8 se ha pulsado Invertir
hacer
si
puerta 2 abierta
hacer
Servo Pin 10 Grados Ángulo 0° Retardo (ms) 0
Establecer puerta 2 abierta = falso
sino
Servo Pin 10 Grados Ángulo 180° Retardo (ms) 0
Establecer puerta 2 abierta = verdadero


para LCD saludo
LCD # 1 Limpiar
LCD # 1 Imprimir Columna 0 Fila 0 " Malet.Innov. 4.0 "
LCD # 1 Imprimir Columna 0 Fila 1 " ArduinoBlocks "

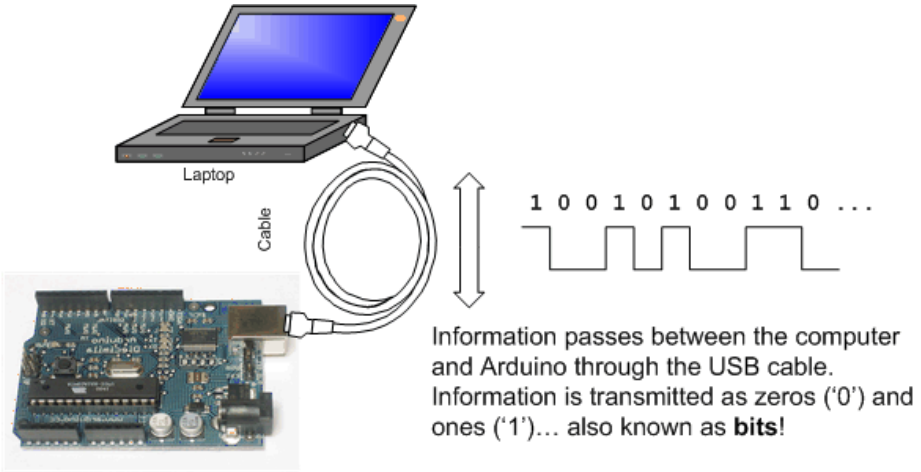

para LCD info
LCD # 1 Limpiar
LCD # 1 Imprimir Columna 0 Fila 0 " CO2 (ppm): "
LCD # 1 Imprimir Columna 11 Fila 0 Formatear número Sensor CO2/TVOC (CCS811) CO2 (ppm) con 0 decimales
LCD # 1 Imprimir Columna 0 Fila 1 " T: "
LCD # 1 Imprimir Columna 3 Fila 1 Formatear número DHT-22 Temperatura °C Pin A0 con 1 decimales
LCD # 1 Imprimir Columna 8 Fila 1 " H: "
LCD # 1 Imprimir Columna 11 Fila 1 Formatear número DHT-22 Humedad % Pin A0 con 1 decimales

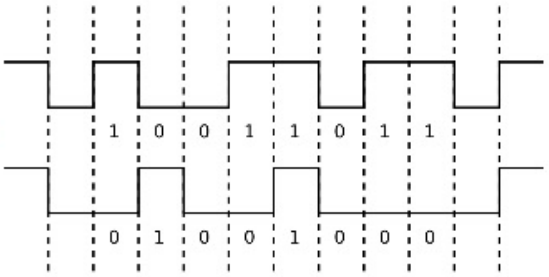
Comunicaciones - Serie / Bluetooth

Arduino permite enviar o recibir información a través de su conexión serie (pines 0,1)

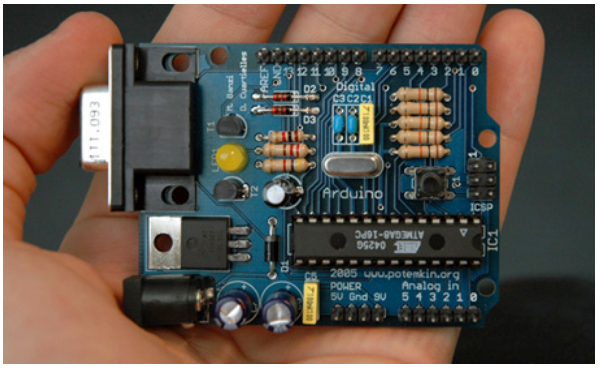
Se utiliza un protocolo UART:

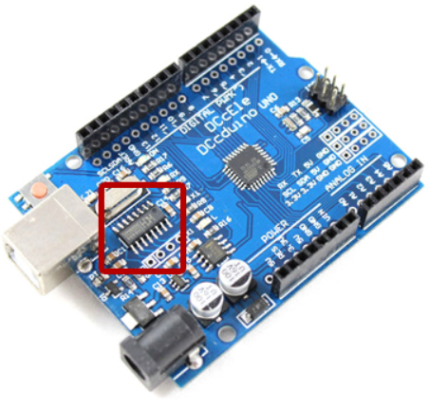
https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html





Originalmente Arduino tenía un puerto serie (tipo RS232) para conectar al PC, hoy en día ese puerto está prácticamente extinguido en los ordenadores y en su lugar se utiliza el USB que sería la evolución de este tipo de puerto (aunque mucho más avanzado, pero al fin y al cabo una comunicación de datos en serie de la misma forma). Arduino incorpora un chip que convierte del standard UART TTL a USB (FTDI, CH340G, CP2102,...)





1) Comunicación serie con cable

La comunicación serie tiene dos propósitos:

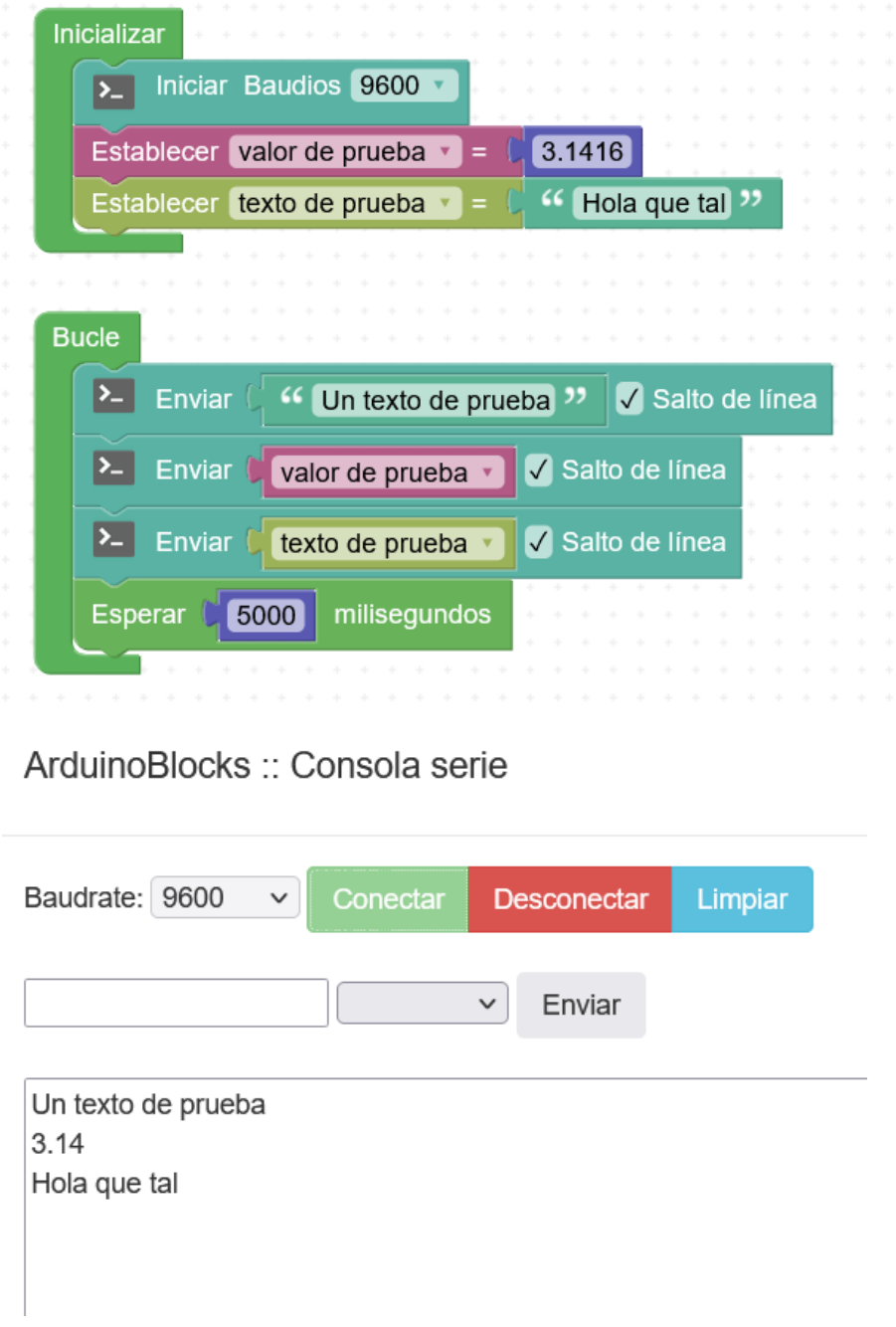
-Permitir el envío del programa para que el propio Arduino lo reciba y se autoprograme la memoria interna de programa (se hace con el "Uploader" dentro del ArduinoBlocks-Connector)

-Intercambiar información entre Arduino y el PC (consola o aplicación específica, en el caso de la consola de ArduinoBlocks el intercambio de datos se realiza con la ayuda del Connector)

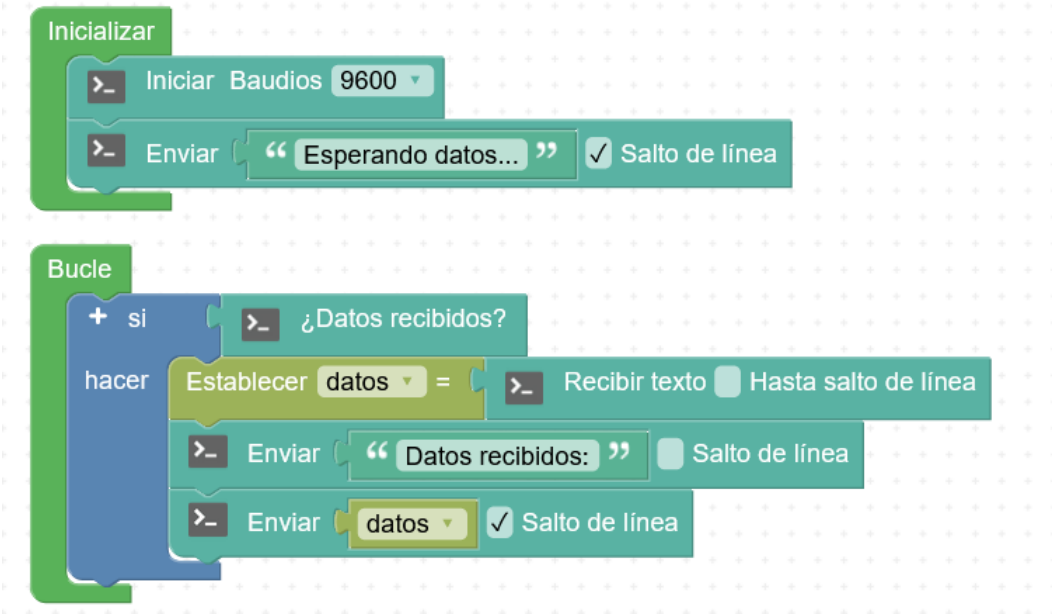
A la hora de intercambiar información entre Arduino y el PC,Móvil,... (cualquier dispositivo capaz de conectarse por comunicación serie con la placa), podemos trabajar a nivel de bytes o de texto (el texto es una secuencia de bytes de caracteres imprimibles)

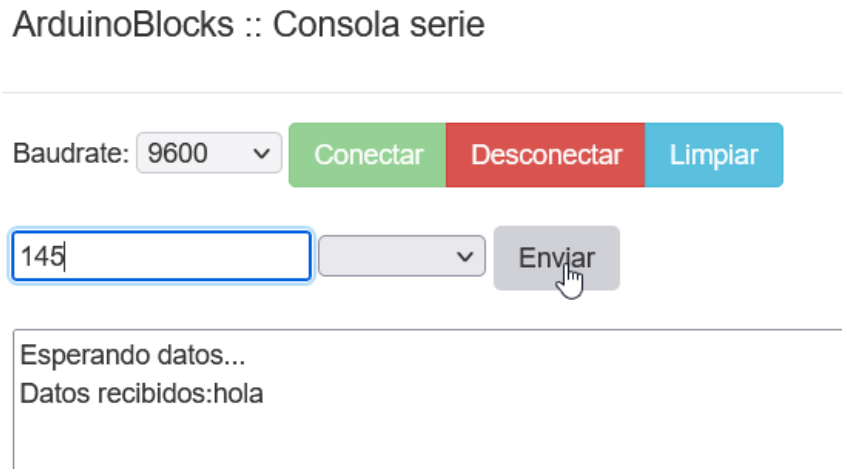
Por simplicidad y por ser más "*entendible*" vamos a ver varios ejemplos de intercambio de datos Arduino <-> PC/Móvil a nivel de texto y cómo interpretar esos datos.

1.1) Ejemplo: Enviar datos Arduino -> PC



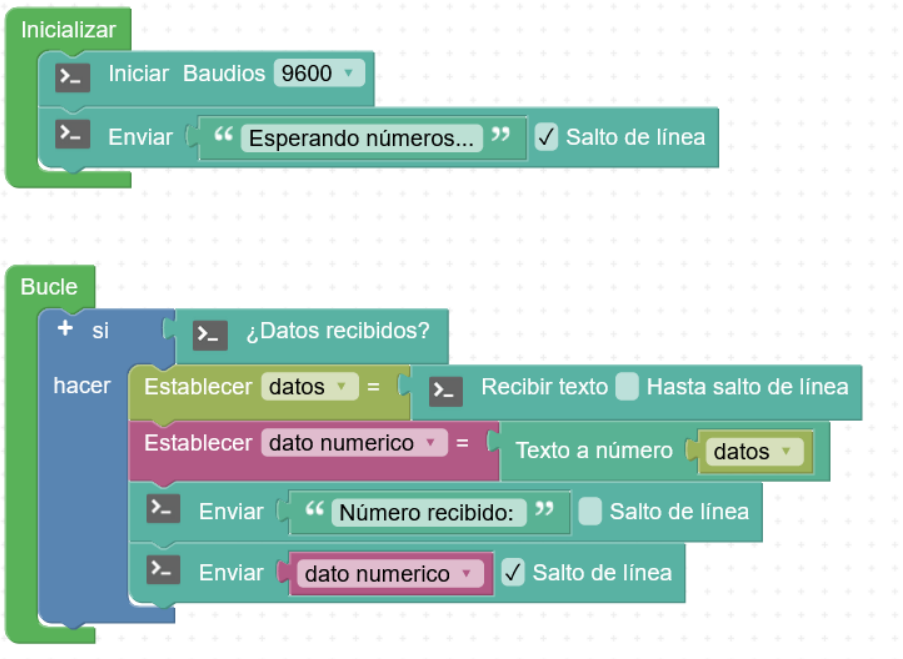
1.2) Recibir datos Arduino <- PC (texto)





1.2) Recibir datos Arduino <- PC (números)

El programa anterior se podría de forma similiar:



Aplicación de ejemplo: "Adivina el número"

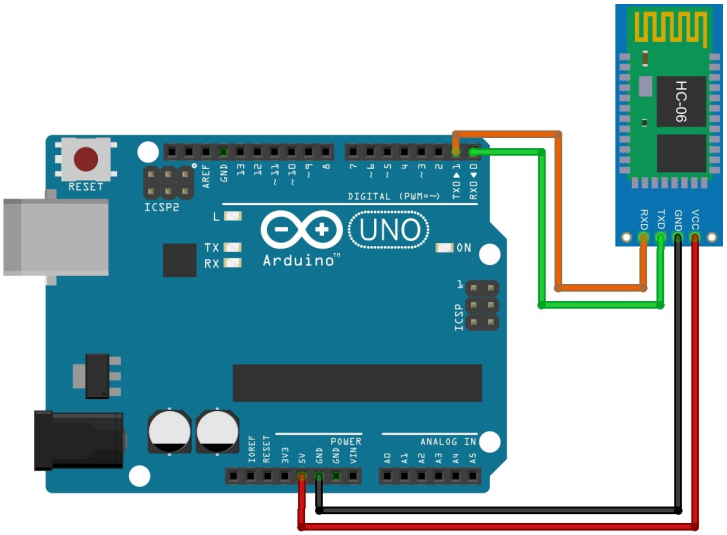
http://www.arduinoblocks.com/web/project/2004

2) Comunicación Bluetooth

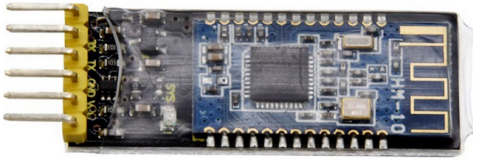
Los módulos Bluetooth para Arduino permiten simular una conexión serie punto a punto de forma inalámbrica a través de la conexión Bluetooth, de forma que la manera de trabajar es prácticamente igual que con la comunicación serie por cable.

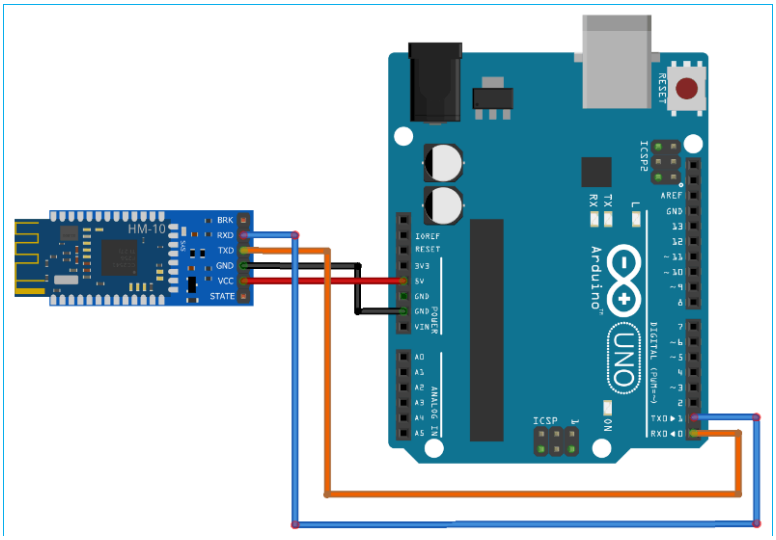
Los módulos Bluetooth más comunes son:

HC-05 / HC-06 (Bluetooth "normal")



HM-10 (Bluetooth BLE 4.0)





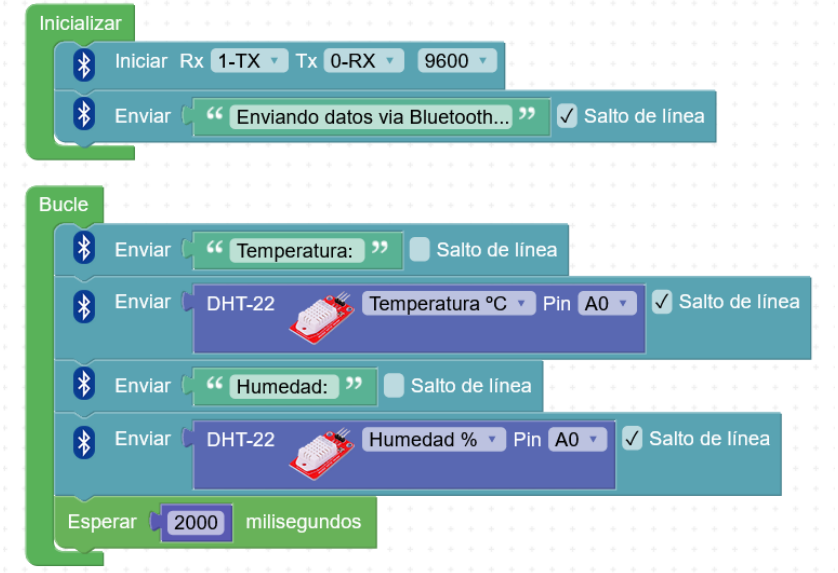
El módulo bluetooth se puede usar en pines diferentes al 0,1 para implementar un puerto serie por software y así dejar el puerto serie por cable utilizable también.

Si usamos el módulo Bluetooth conectado a los pines 0,1 para usar el puerto serie integrado de Arduino, debemos configurar de la siguiente forma:

(el pin RX de Arduino se conecta con el TX del módulo Bluetooth, y viceversa)

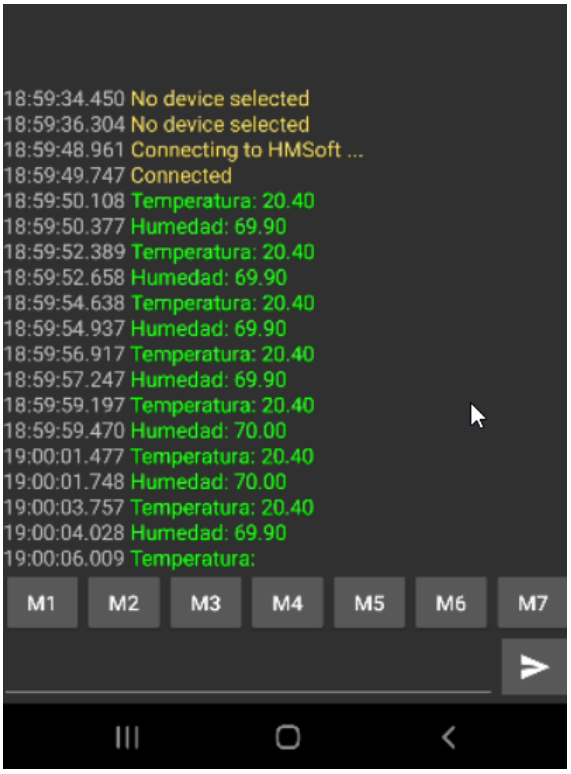
2.1) Enviar datos Bluetooth -> PC/Móvil

Programa para enviar Temperatura y humedad via Bluetooth:

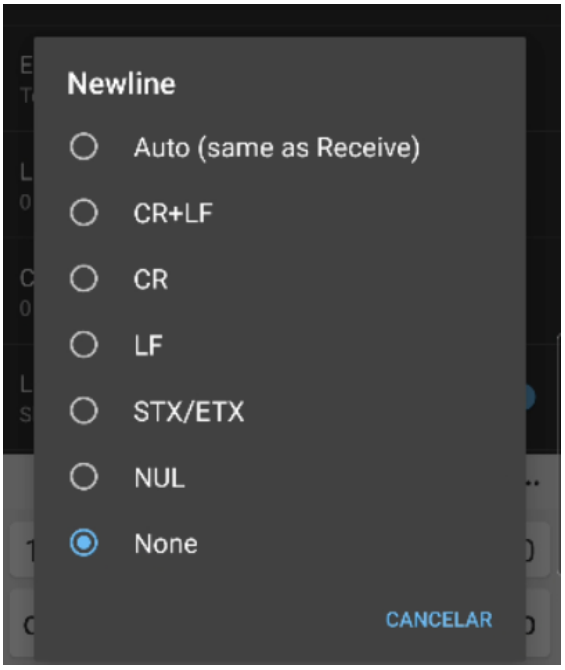


Visualizamos los datos desde una aplicación de "Consola Bluetooth o Terminal Bluetooth" en un móvil Android:

https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=es&gl=US



Configuración para enviar y recibir sin saltos de líneas en la aplicación de Terminal Bluetooth:



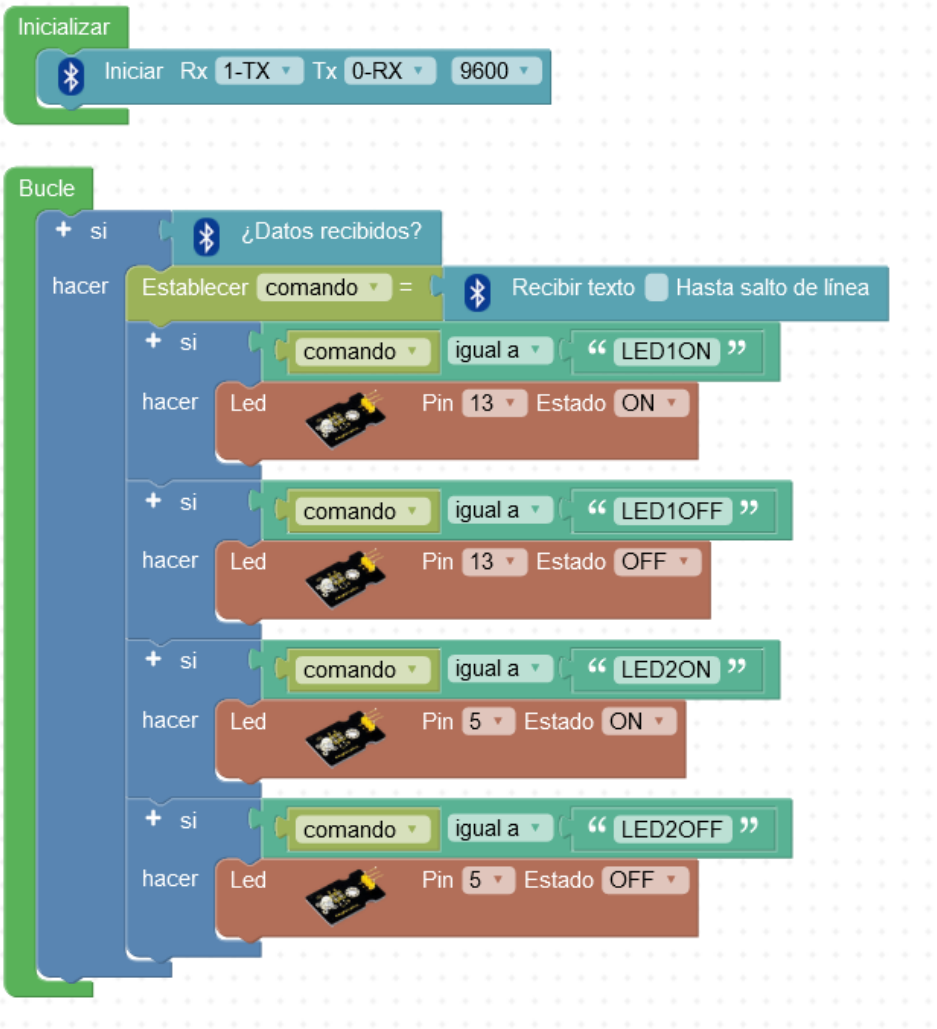
2.2) Recibir datos Bluetooth <- PC/Móvil

Vamos a implementar un sencillo sistema para enviar comandos simples desde la terminal Bluetooth del móvil y según el comando encendemos o apagamos un led conectado a Arduino.

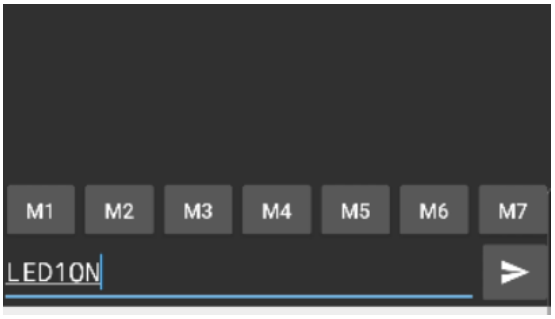
Comandos:

LED1ON / LED1OFF -> Enciende o apaga el led coenctado en el pin 13

LED2ON / LED2OFF -> Enciende o apaga el led conectado en el pin 5



Desde la terminal Bluerooth:



2.3) Protocolo simple de comunicación

En muchas ocasiones necesitamos realizar comandos con algún parámetro, por ejemplo siguiendo el ejemplo anterior podríamos indicar el led y su nivel de intensidad de 0 a 255 (en vez de ON/OFF)

O por ejemplo para situar un servomotor en una posición en concreta entre 0 y 180 grados.

Podemos crear un sistema simple de comando+parámetro con el siguiente formato:

COMANDO/VALOR#

Para que funcione correctamente, siempre los comandos enviados deben seguir ese formato:

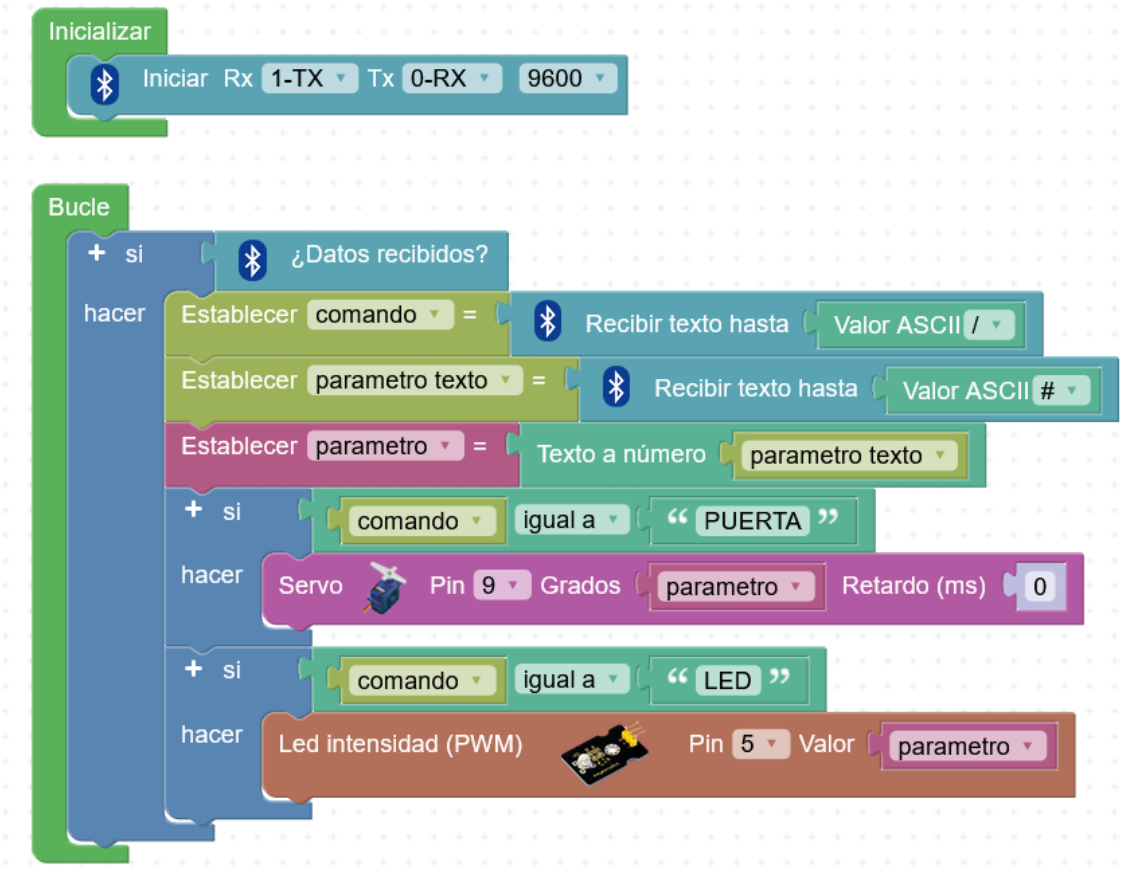
Comando "PUERTA" con parámetro entre 0 y 180

Ejemplo: PUERTA/90#

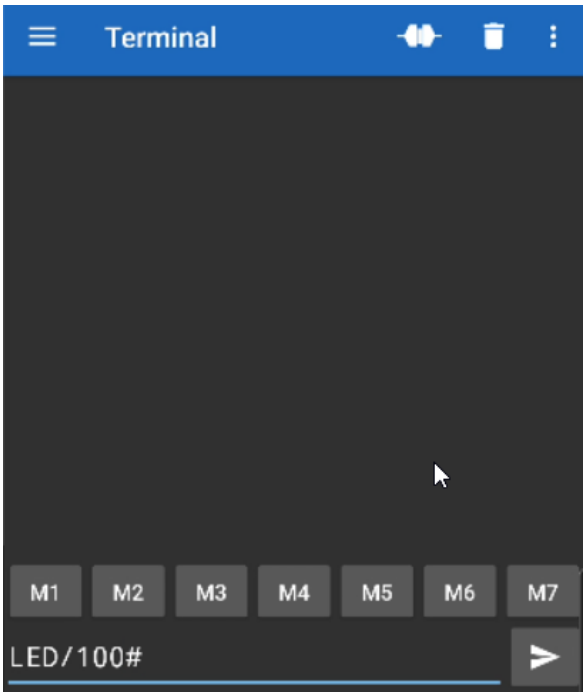
Comando "LED" con parámetro entre 0 a 255

Ejemplo: LED/200#

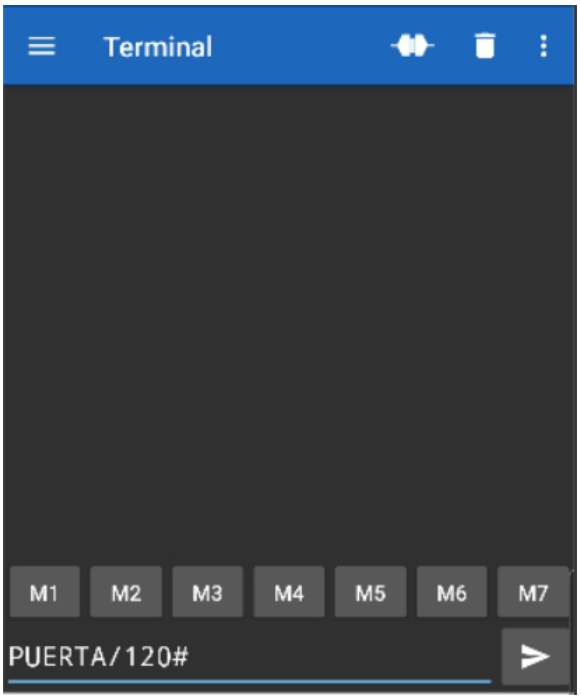
Programa para recibir y procesar el comando + valor según el formato indicado:



Ejemplo para poner el led a intensidad 100:



Ejemplo para mover la puerta a 120 grados:



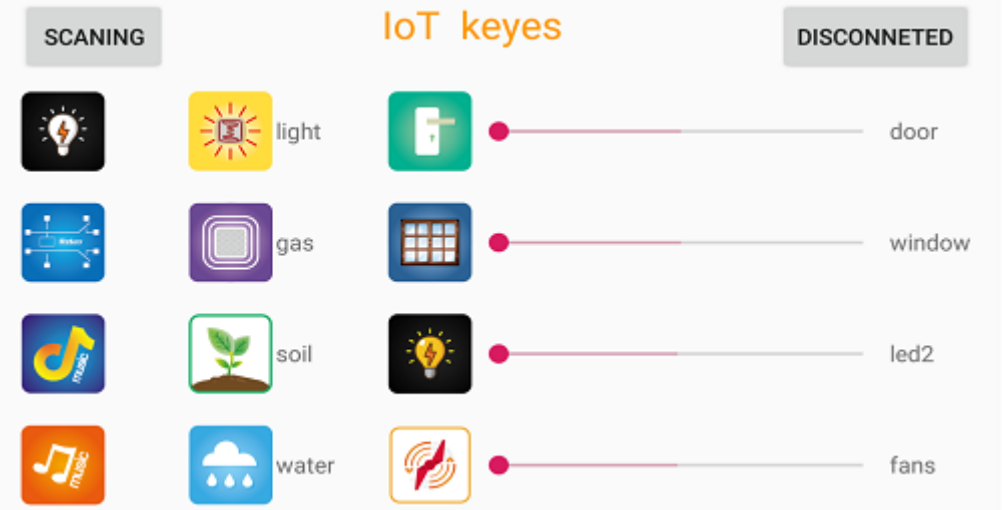
Bluetooth + SmartHome Kit App

Módulo Bluetooth HM-10 (Bluetooth 4.0 BLE)

Programa de ejemplo para funcionar con la APP (versión código Arduino IDE)

https://wiki.keyestudio.com/KS0085_Keyestudio_Smart_Home_Kit_for_Arduino#Project_15.EF.BC.9AMulti-purpose_Smart_Home_Kit

App de ejemplo:



GooglePlay (Android): https://play.google.com/store/apps/details?id=com.keyestudio.iot_keyes

AppStore (iOS): https://apps.apple.com/es/app/iot-keyes/id1487578236

Comandos enviados por la aplicación:

'a' -> luz blanca ON

'b' -> luz blanca OFF

'c' -> relé ON

'd' -> relé OFF

'e' -> música 1

'f' -> música 2

'h' -> nivel de luz (LDR)

'i' -> nivel de CO2 (devuelve el valor)

'j' -> nivel humedad suelo (devuelve valor)

'k' -> nivel de lluvia (devuelve valor)

'l' -> ventana abierta (180º)

'm' -> ventana cerrada (0º)

'n' -> puerta abierta (180º)

'o' -> puerta cerrada (0º)

'p' -> led amarillo ON

'q' -> led amarillo OFF

'r' -> ventilador ON

's' -> ventilador OFF

't' + [valor] -> ventana a posición [valor]

'u' + [valor] -> puerta a posición [valor]

'v' + [valor] -> led amarillo intensidad PWM = [valor]

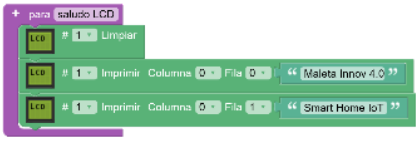
'w' + [valor] -> motor ventilador velocidad PWM = [valor]

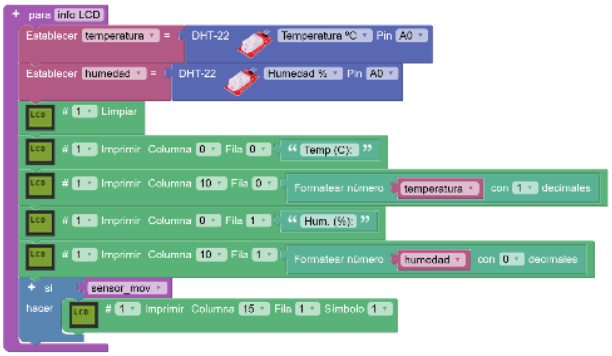
Programa ArduinoBlocks:

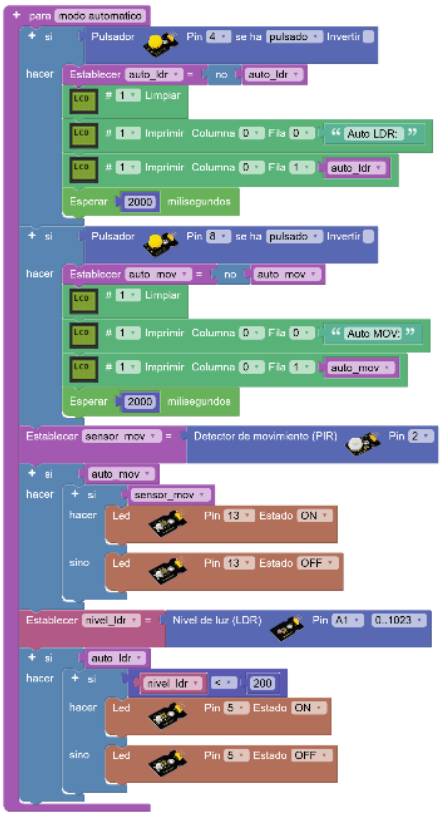
http://www.arduinoblocks.com/web/project/710937

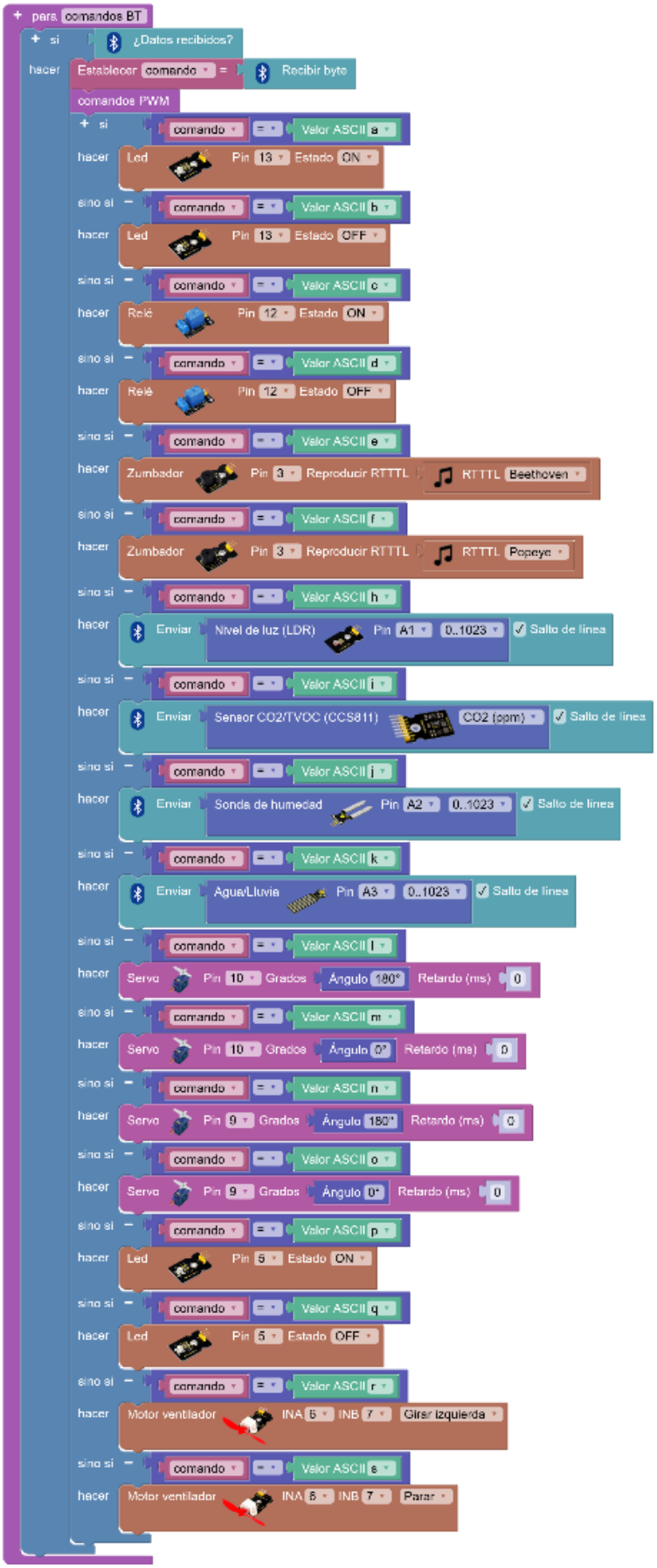

Inicializar
Iniciar Rx 1-TX Tx 0-RX 9600
1 Iniciar 2x16 I2C ADDR 0x27
1 Definir Símbolo 1 B00000,B00100,B01010,B10001,B00100,B01010,B00000...
saludo LCD

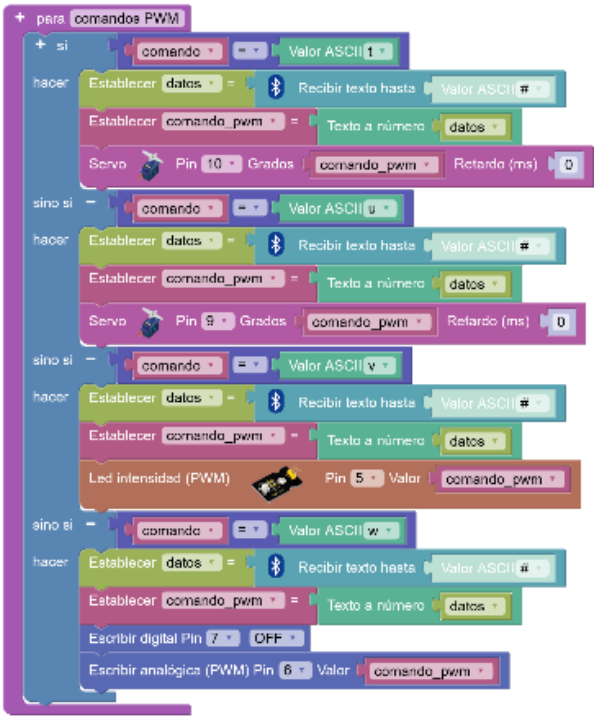

Bucle
comandos BT
modo automatico
Ejecutar cada 5000 ms
info LCD


para saludo LCD
1 Limpiar
1 Imprimir Columna 0 Fila 0 " Maleta Innov 4.0 "
1 Imprimir Columna 0 Fila 1 " Smart Home IoT "


para info LCD
Establecer temperatura = DHT-22 Temperatura °C Pin A0
Establecer humedad = DHT-22 Humedad % Pin A0
1 Limpiar
1 Imprimir Columna 0 Fila 0 " Temp (C): "
1 Imprimir Columna 10 Fila 0 Formatear número temperatura con 1 decimales
1 Imprimir Columna 0 Fila 1 " Hum. (%): "
1 Imprimir Columna 10 Fila 1 Formatear número humedad con 0 decimales
si sensor_mov
hacer # 1 Imprimir Columna 15 Fila 1 Símbolo 1


para modo automatico
si Pulsador Pin 4 se ha pulsado Invertir
hacer Establecer auto_ldr = no auto_ldr
1 Limpiar
1 Imprimir Columna 0 Fila 0 " Auto LDR: "
1 Imprimir Columna 0 Fila 1 auto_ldr
Esperar 2000 milisegundos
si Pulsador Pin 8 se ha pulsado Invertir
hacer Establecer auto_mov = no auto_mov
1 Limpiar
1 Imprimir Columna 0 Fila 0 " Auto MOV: "
1 Imprimir Columna 0 Fila 1 auto_mov
Esperar 2000 milisegundos
Establecer sensor_mov = Detector de movimiento (PIR) Pin 2
si auto_mov
hacer si sensor_mov
hacer Led Pin 13 Estado ON
sino Led Pin 13 Estado OFF
Establecer nivel_ldr = Nivel de luz (LDR) Pin A1 0..1023
si auto_ldr
hacer si nivel_ldr < 200
hacer Led Pin 5 Estado ON
sino Led Pin 5 Estado OFF


para comandos BT
si ¿Datos recibidos?
hacer Establecer comando = Recibir byte
comandos PWM
si comando = Valor ASCII a
hacer Led Pin 13 Estado ON
sino si comando = Valor ASCII b
hacer Led Pin 13 Estado OFF
sino si comando = Valor ASCII c
hacer Relé Pin 12 Estado ON
sino si comando = Valor ASCII d
hacer Relé Pin 12 Estado OFF
sino si comando = Valor ASCII e
hacer Zumbador Pin 3 Reproducir RTTTL RTTTL Beethoven
sino si comando = Valor ASCII f
hacer Zumbador Pin 3 Reproducir RTTTL RTTTL Popeye
sino si comando = Valor ASCII h
hacer Enviar Nivel de luz (LDR) Pin A1 0..1023 Salto de línea
sino si comando = Valor ASCII i
hacer Enviar Sensor CO2/TVOC (CCS811) CO2 (ppm) Salto de línea
sino si comando = Valor ASCII j
hacer Enviar Sonda de humedad Pin A2 0..1023 Salto de línea
sino si comando = Valor ASCII k
hacer Enviar Agua/Lluvia Pin A3 0..1023 Salto de línea
sino si comando = Valor ASCII l
hacer Servo Pin 10 Grados Ángulo 180° Retardo (ms) 0
sino si comando = Valor ASCII m
hacer Servo Pin 10 Grados Ángulo 0° Retardo (ms) 0
sino si comando = Valor ASCII n
hacer Servo Pin 9 Grados Ángulo 180° Retardo (ms) 0
sino si comando = Valor ASCII o
hacer Servo Pin 9 Grados Ángulo 0° Retardo (ms) 0
sino si comando = Valor ASCII p
hacer Led Pin 5 Estado ON
sino si comando = Valor ASCII q
hacer Led Pin 5 Estado OFF
sino si comando = Valor ASCII r
hacer Motor ventilador INA 6 INB 7 Girar izquierda
sino si comando = Valor ASCII s
hacer Motor ventilador INA 6 INB 7 Parar


para comandos PWM
si comando = Valor ASCII 1
hacer Establecer datos = Recibir texto hasta Valor ASCII #
Establecer comando_pwm = Texto a número datos
Servo Pin 10 Grados comando_pwm Retardo (ms) 0
sino si comando = Valor ASCII u
hacer Establecer datos = Recibir texto hasta Valor ASCII #
Establecer comando_pwm = Texto a número datos
Servo Pin 9 Grados comando_pwm Retardo (ms) 0
sino si comando = Valor ASCII v
hacer Establecer datos = Recibir texto hasta Valor ASCII #
Establecer comando_pwm = Texto a número datos
Led intensidad (PWM) Pin 5 Valor comando_pwm
sino si comando = Valor ASCII w
hacer Establecer datos = Recibir texto hasta Valor ASCII #
Establecer comando_pwm = Texto a número datos
Escribir digital Pin 7 OFF
Escribir analógica (PWM) Pin 6 Valor comando_pwm

Bluetooth + AppInventor

AppInventor es un entorno online de programación visual por bloques que permite crear de forma sencilla aplicaciones móviles compatibles con Android (y ahora también iOS, aunque con algunas limitaciones)

https://appinventor.mit.edu/

AppInventor es compatible con Bleutooth Standarda, para trabajar con BLE debemos instalar un plugin:

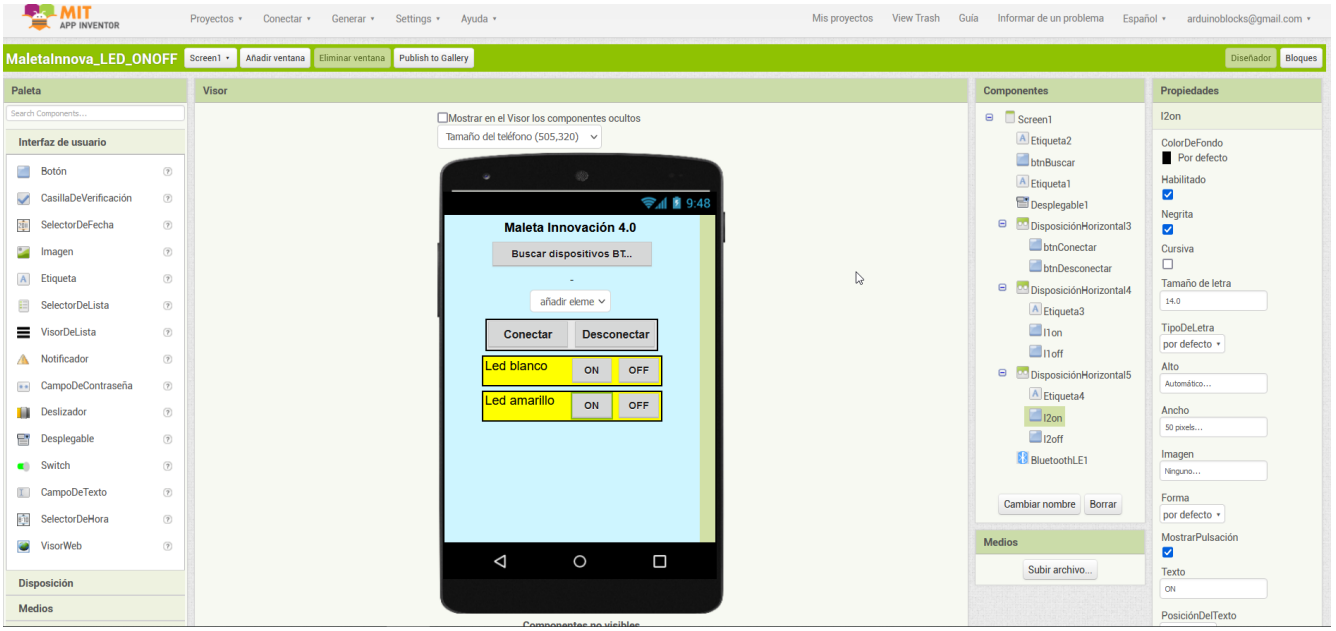
https://mit-cml.github.io/extensions/

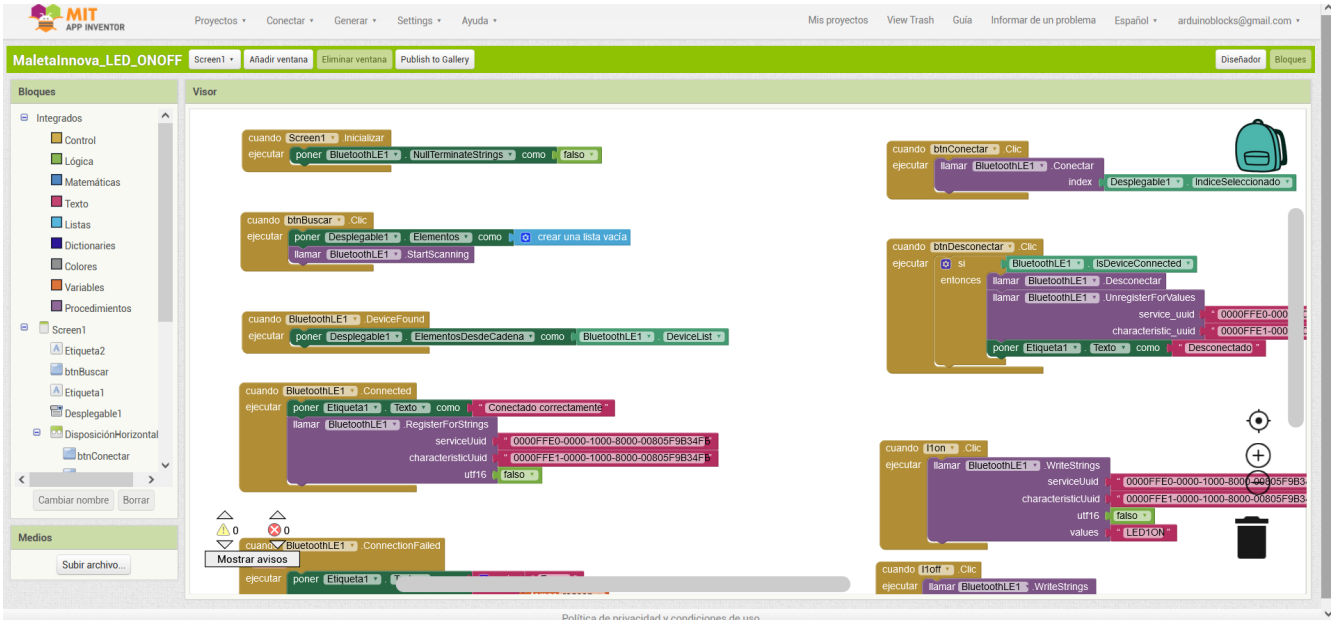
http://iot.appinventor.mit.edu/assets/resources/edu.mit.appinventor.ble-20200828.aix

1)Ejemplo de una aplicación sencilla para controlar el led encendido o apagado con comandos simples:

AppInventor:

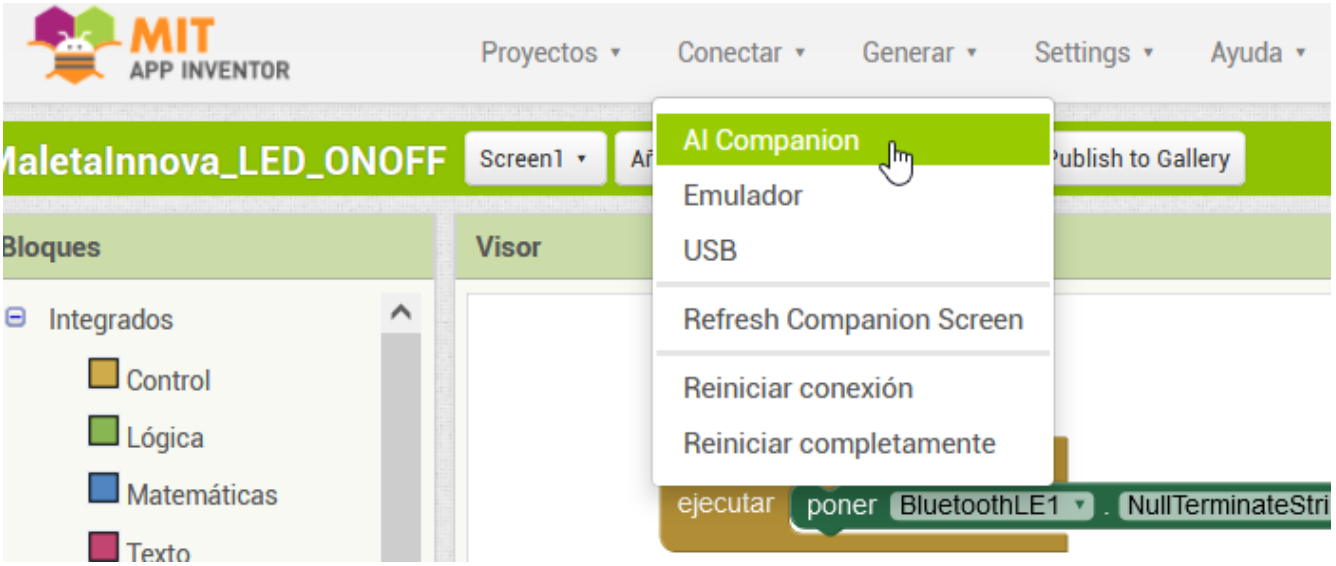
https://drive.google.com/file/d/1ZViHS1HGzNNt_nvboZjLfkMG6g7x5WxH/view?usp=sharing





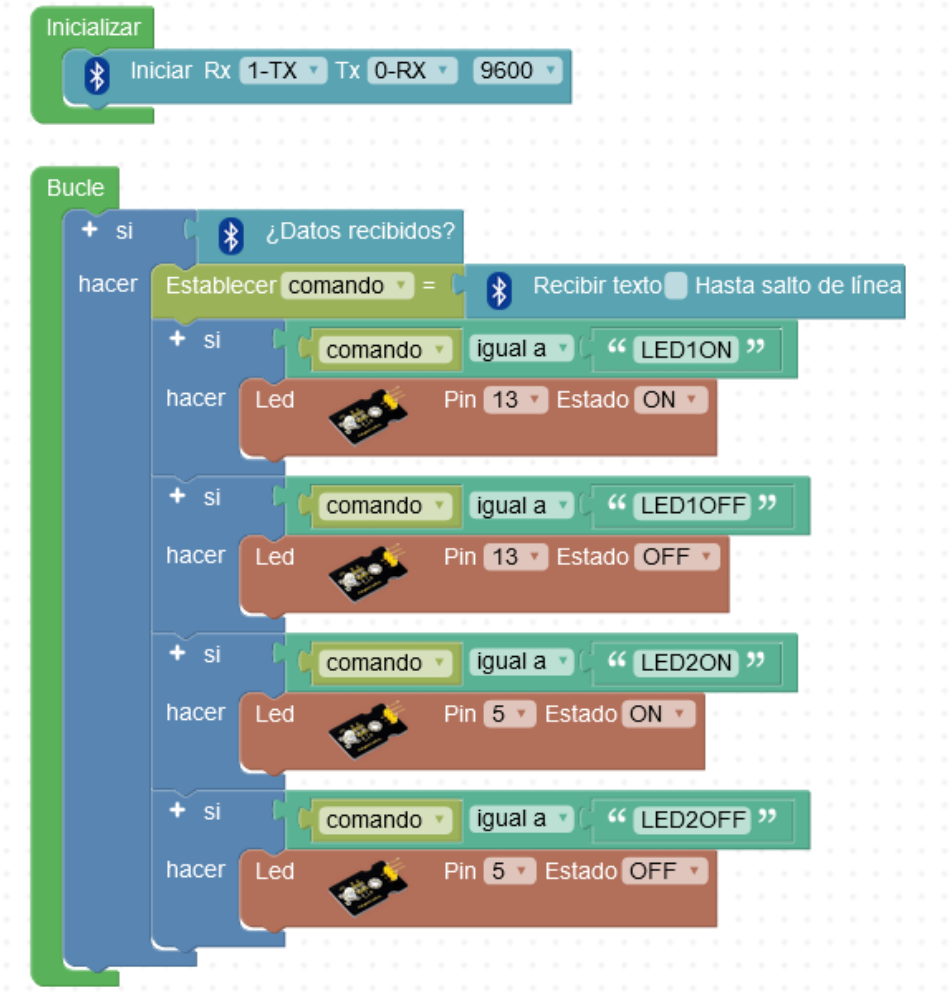
Podemos probar la aplicación en el dispositivo mediante la aplicación "AI Companion" (una vez completada y probada podemos generar la aplicación e instalarla)

https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3&hl=es_419&gl=US



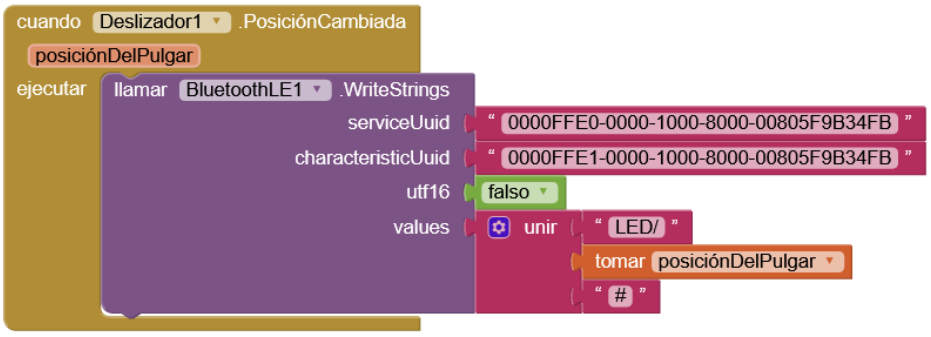
Aplicación ejecutándose en Android:

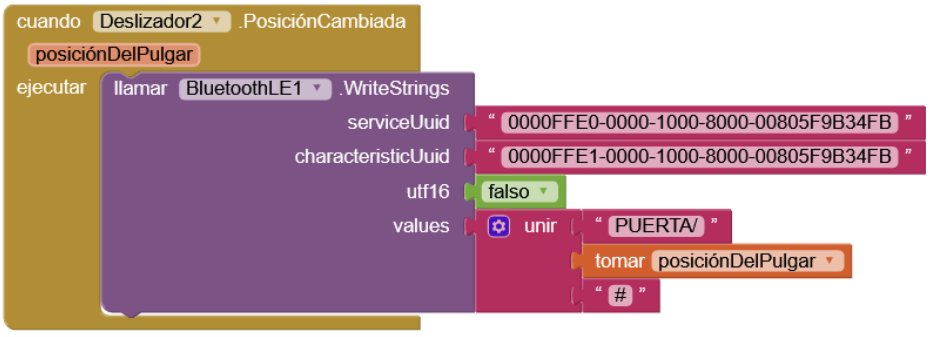
y programa en Arduino:

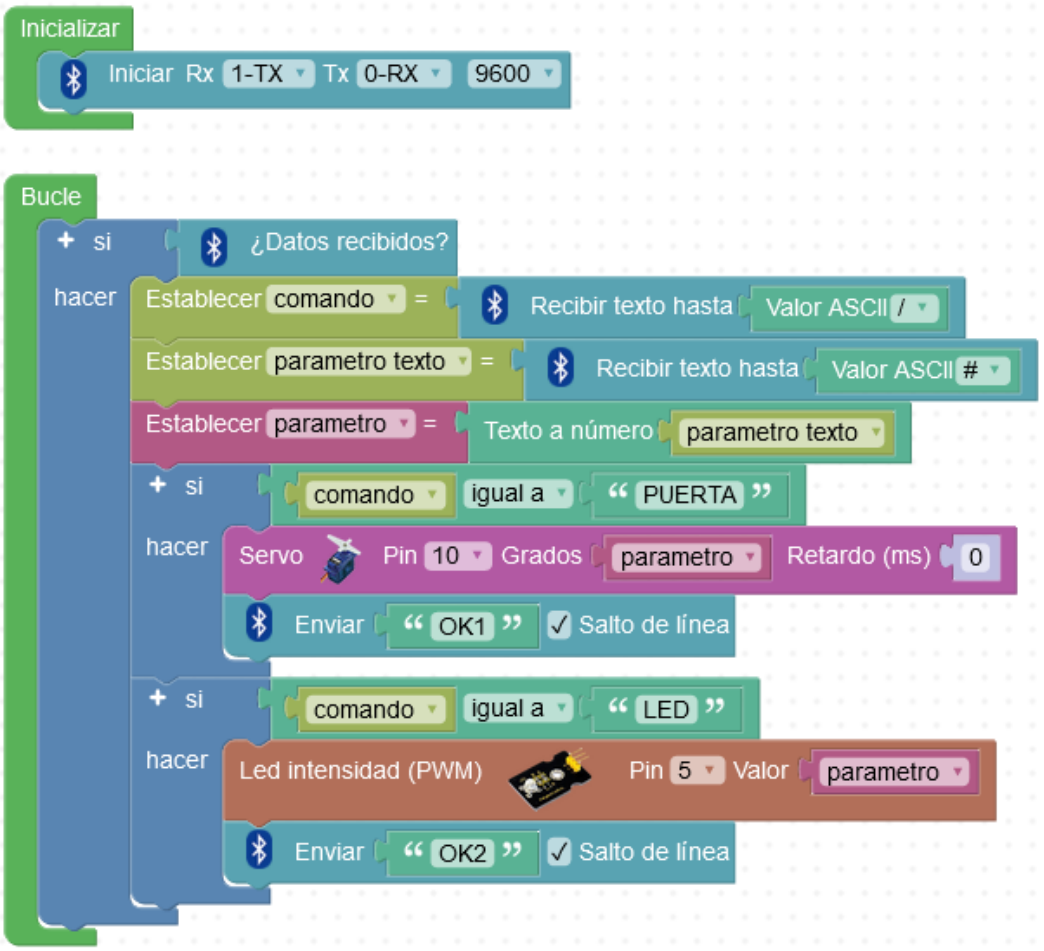


2) Ejemplo de una aplicación sencilla para controlar la intensidad de un led com comando+valor:

https://drive.google.com/file/d/1dRRscU6GJ20I29VQ295Q29qYbEWy7ntc/view?usp=sharing





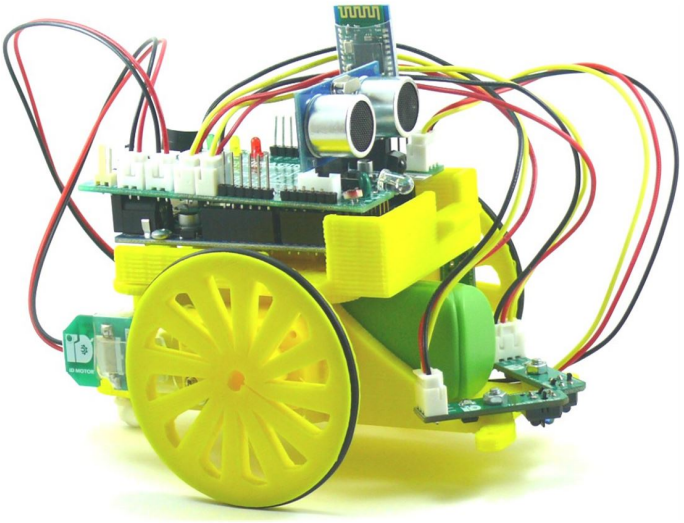

Inicializar
Iniciar Rx 1-TX Tx 0-RX 9600
Bucle
si
¿Datos recibidos?
hacer
Establecer comando = Recibir texto hasta Valor ASCII /
Establecer parametro texto = Recibir texto hasta Valor ASCII #
Establecer parametro = Texto a número parametro texto
si
comando igual a " PUERTA "
hacer
Servo Pin 10 Grados parametro Retardo (ms) 0
Enviar " OK1 " Salto de línea
si
comando igual a " LED "
hacer
Led intensidad (PWM) Pin 5 Valor parametro
Enviar " OK2 " Salto de línea

Otras placas y kits

Arduino UNO es perfecto para iniciarse en la robótica educativa y junto al kit smarthome tenemos un punto de partida con posibilidad de multitud de proyectos y funcionalidades gracias a la variedad de sensores, actuadores y periféricos que incorpora así como la conectividad bluetooth.

Disponemos de kits al mismo nivel con distintas aplicaciones, y también totalmente compatibles con ArduinoBlocks:

- **3dBot**: imprimible en 3d, siguelíneas, evita-obstáculos, control bluetooth, etc.



https://shop.innovadidactic.com/es/placas-kits-y-robots/64-kit-imagina-arduino-3dbot-para-arduinoblocks-y-snap4arduino.html

-Manual de prácticas 3dBot con ArduinoBlocks:

https://drive.google.com/file/d/1ARwRJfFNYqoybMLQeNTnBPpu5zYU0sto/view?usp=sharing

- **KeyBot**: similar al 3dBot pero con soporte metálico (no es imprimible en 3d, y no lleva Arduino reutilizable, es una placa personalizada que incluye el microcontrolador Arduino y conexiones tipo RJ11 para ampliar)

https://shop.innovadidactic.com/es/easy-plug-placas-shields-y-kits/851-keyestudio-kit-robot-keybot-pista.html

-Manual KeyBot con ArduinoBlocks:

https://drive.google.com/file/d/1u78lleYch46GRNQqxNfEb-kh3ZPeiQYa/view?usp=sharing

- **Kit Arduino + TDR Steam:** incuye un Arduino UNO y una shield educativa con múltiples sensores, una pantalla LCD, un mando IR, sensor de sonido externo, y posibilidad de ampliación

https://shop.innovadidactic.com/es/placas-kits-y-robots/1445-kit-imagina-tdr-steam-basado-en-arduino.html

-Manual TDR-STEAM con ArduinoBlocks:

https://drive.google.com/file/d/1d0E5d3q5bRT2IFGZSeYi0npw4KE6XMtY/view?usp=sharing

- **ESP32 STEAMakers:** Una placa muy potencia y con grandes posibilidades para el IoT (internet de las cosas). Incorpora un microcontrolador ESP32 de doble núcleo, WiFi, Bluetooth clásico y BLE. Formato compatible Arduino y con conexiones modulares.

https://shop.innovadidactic.com/es/standard-placas-shields-y-kits/1567-placa-esp32-steamakers.html

-Prácticas ESP32 STEAMakers con ArduinoBlocks:

https://drive.google.com/drive/u/1/folders/1RRSVxQpGqOphfk_Koh1Gvfbjt1JMZo1b

-Videos demostrativos:

https://www.youtube.com/watch?v=MQjIEI7I4ik&list=PL1pKD-Bz2QBAgfy580m8OaQ2Z60v6DOhC