4. Comunicaciones

Introducción

Las tecnologías de comunicaciones se basan en la transmisión de datos entre puntos distantes. Estos datos, se transmiten en forma de señales eléctricas y pueden ser enviadas através de cables o de manera inalámbrica.

En el Arduino trabajamos con dos tipos de comunicaciones: * **Alámbrica** Comunicación puerto serie: * *PC-Arduino*: La comunicación vía puerto serie: * Lo vimos por primera vez en el semáforo. * Hay que incializar el puerto serie **Serial.begin(9600)** * Con la función **Serial.print** Arduino puede enviar al ordenador los datos que queramos. * *Arduino- Arduino* * Veremos en esta unidad una forma muy sencilla de comunicarse dos arduinos también por puerto serie. * **Inalámbrica**: La comunicación vía *Bluetooth* * En esta unidad vamos a utilizar un nuevo dispositivo **JY-MCU** * Emparejaremos con nuestro Smartphone (Android) y podremos enviar órdenes de nuestro móvil al Arduino.

si tienes dudas técnicas en este capítulo pon un ticket a http://soporte.catedu.es/ o al Telegram Whatsapp en www.catedu.es - información y te ayudaremos:


CATEDU
Usuario invitado | Iniciar sesión
Inicio Centro de Soporte
Abrir un nuevo Ticket
Ver Estado de un Ticket
Buscar en nuestra base de conocimiento
Buscar
Abrir un nuevo Ticket
Ver Estado de un Ticket
Preguntas destacadas
Problema técnico en mi centro
Dos o más cursos on-line coinciden
Programa de Dietas, Gestión o Inventario RyC
¿Cómo recibo el certificado de mis cursos de AULARAGON?, debo de solicitarlos de algún modo?

¿Qué vamos a hacer?

Existen módulos adicionales que se pueden conectar a la placa básica Arduino que pueden dotar de una gran funcionalidad a los proyectos que queramos realizar. En esta práctica utilizaremos un **módulo Bluetooth** que nos permite establecer una comunicación inalámbrica con el entorno, el dispositivo elegido más fácil va a ser un **móvil**.

Conocimiento previo

- Programación básica Arduino
- Uso de librerías externas y comunicación serie (para configuración de parámetros).

Objetivos

- Conectar el módulo Bluetooth a Arduino.
- Realizar programas para comunicar Arduino con el exterior vía Bluetooth.
- Configurar los parámetros del módulo de Bluetooth (avanzado).

Lista de materiales:

- Arduino UNO.
- Módulo Bluetooth.
- Móvil con Android

https://giphy.com/embed/zArTrAsfv4onu

via GIPHY

Un poco de teoría Bluetooth

ONDAS

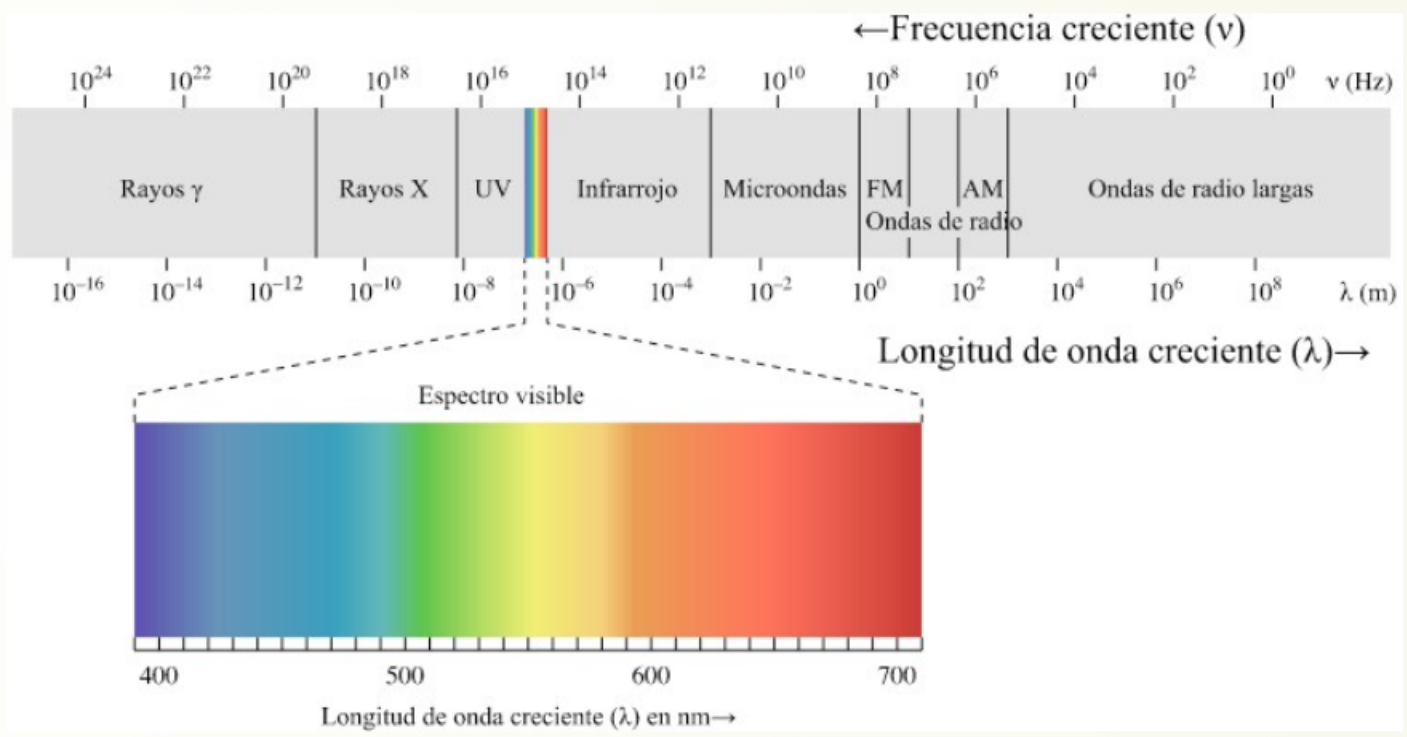
Una onda es una señal que se propaga por un medio. Por ejemplo el sonido, que es una onda mecánica que viaja usando el aire o cualquier otro material. Pero en el caso de las señales eléctricas pueden ser enviadas por el cable o a través del vacío (no necesitan un medio para transmitirse).

Dependen de 3 parámetros principalmente:

- **Amplitud:** altura máxima de la onda. Hablando de sonido representaría el volumen. Si nos referimos a una onda eléctrica estaríamos representando normalmente el voltaje.
- **Longitud de onda λ:** distancia entre el primer y último punto de un ciclo de la onda (que normalmente se repite en el tiempo).
- **Frecuencia f :** Número de veces que la onda repite su ciclo en 1 segundo (se mide en hertzios).
- **Periodo T** es simplemente es la inversa de la frecuencia. T=1/f

La relación entre ellas es muy fácil pues las ondas electromagnéticas viajan a la velocidad de la luz c y si velocidad es espacio/tiempo luego c = λ/T luego **c= λ*f**

Dentro del espectro electromagnético encontramos diferentes tipos de señales dependiendo de las características de su onda.



TRANSMISIÓN INALÁMBRICA: BLUETOOTH.

- Hoy en día, este grupo está formado por miles de empresas y se utiliza no sólo para teléfonos sino para cientos de dispositivos.
- Bluetooth es una red inalámbrica de corto alcance pensada para conectar pares de dispositivos y crear una pequeña red punto a punto, (sólo 2 dispositivos).
- Utiliza una parte del espectro electromagnético llamado "**Banda ISM**", reservado para fines no comerciales de la industria, área científica y medicina. Dentro de esta banda también se encuentran todas las redes WIFI que usamos a diario. En concreto funcionan a 2,4GHz. (Un G son 10^9) luego entre FM y Microondas.

> **¿Sabias que?**
> Su curioso nombre viene de un antiguo rey Noruego y Danés, y su símbolo, de las antiguas ruinas que representan ese mismo nombre.

Hay 3 clases de bluetooth que nos indican la máxima potencia a la que emiten y por tanto la distancia máxima que podrán alcanzar:

CLASE	POTENCIA	DISTANCIA
Clase 1	100 mW	100 m
Clase 2	2,5 mW	10 m
Clase 3	1 mW	1 m

También es muy importante la velocidad a la que pueden enviarse los datos con este protocolo:

Versión	Velocidad
1.2	1 Mbps
2	3 Mbps
3	24 Mbps
4	24 Mbps

> Mbps : Mega Bits por segundo. MBps: Mega Bytes por segundo.
> kb = 1.024 b M = 1.024 k G = 1.024 M

¿Te atreves a calcularlo …. ?

¿Cuántos ciclos por segundo tendrán las ondas que están en la **Banda ISM?** ¿Cuál es el periodo de esas ondas?

Solución
a) f = 2.4G
b) λ=c/f= 12.5cm o sea, las antenas tendrían que ser de esta longitud. Hay muchos trucos para reducirla, una de ellas es la forma de serpiente que puedes ver en el HC-06

¿Te atreves a calcularlo...?

¿A qué distancia y cuanto tiempo tardarían en enviarse los siguientes archivos por Bluetooth?

1. Un vídeo de 7Mb usando versión 2 clase 2
2. Una imagen de 2.5Mb usando versión 3 clase 1
3. Un archivo de texto de 240KB usando versión 1.2 clase 1

Solución
1) 7Mb / 3Mbs = 2.3 seg.
2) 2.5Mb / 24Mbs = 0.1 seg.
3) 240 kB 8b/B = 1.920 kb 1.920 kb / 1.024 = 1.875 Mb 1.875Mb / 1Mbs = 1.875 seg.

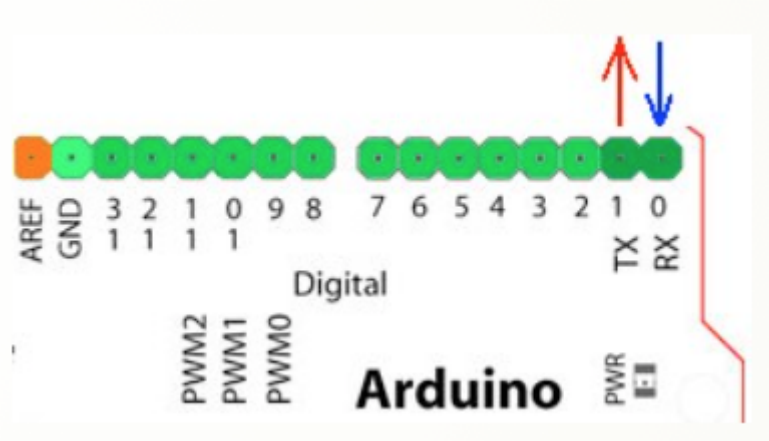
Módulo HC-06

Vamos a utilizar en estos ejemplos un módulo **esclavo** de bluetooth **JY-MCU** o también **HC-06** muy común y económico. Es posible usar otros módulos ya que existe un mercado de desarrollo continuo, en cualquier caso el funcionamiento básico es el mismo. Dicho módulo por tratarse de un módulo **esclavo**, está configurado para conectarse a un maestro y recibir órdenes de él.

Inicialmente no necesitas configurarlo, sino que al cargar el código desde el ordenador, conectarás el módulo y este **empezará a parpadear** indicando que está buscando un master al que conectarse, (por ejemplo tu teléfono o una llave bluetooth usb conectado a un pc...).

Como ya sabrás los dispositivos de este tipo tienen que “emparejarse” y tienen que compartir una contraseña para que los datos puedan intercambiarse. Por defecto, estos módulos tienen la contraseña **1234**, aunque tanto esto como el nombre, pueden ser actualizados mediante unos comandos especiales, llamados AT y que veremos un poco más adelante.

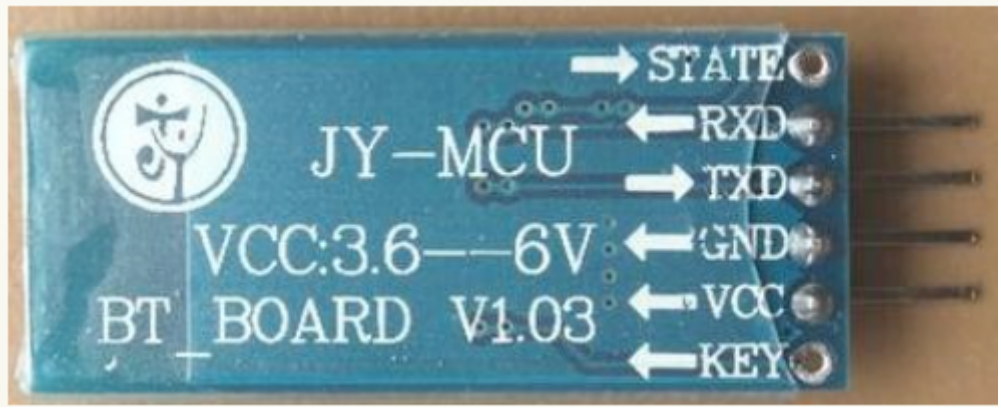
Arduino tiene 2 pines que permiten enviar y transmitir datos serie (uno datos tras otro). Lo usamos continuamente cuando enviamos un programa desde nuestro ordenador a Arduino o cuando hacemos una lectura desde el monitor serie (con un **Serial.print();**).



Arduino tiene definidos estos pines como:

- pin digital 0: RX <- (Arduino recibe a través de este pin).
- pin digital 1: TX -> (Arduino envía a través de este pin).

El módulo bluetooth tiene 4 patillas. 2 para la alimentación y 2 para la comunicación.

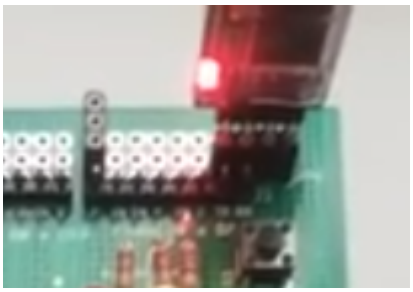


> Es **MUY IMPORTANTE** conectar de manera correcta estos pines con Arduino para la correcta comunicación. **La patilla que emite los datos (TX) en el bluetooth debe estar conectada a la que recibe los datos (RX) en Arduino, y viceversa.**

Aunque el módulo funciona a 3.3v, normalmente las placas comerciales, (como la que estamos usando), llevan un regulador y las podemos conectar directamente a los 5v de Arduino.

Conexión en la shield Edubásica

La conexión es muy fácil, ya tiene JP6 para conectarlo diréctamente, con la luz led mirando hacia dentro de la placa:



Conexión sin la shield Edubásica

Es también simple, utilizando una placa Protoboard pero **intercambiando Rx y Tx** es decir Rx del HC-06 con Tx del Arduino y Tx del HC-06 con Rx del Arduino.

> Image not found or type unknown
>
> **Nota**; No necesariamente hay que conectar Rx y Tx a D0 y D1, pues esto tiene una fuerte desventaja: Ocupas el puerto serie cuando conectas el Bluetooth.
> **Hay que quitar el HC-06 cada vez que te conectes por el puerto serie (es decir cuando te conectas por USB para cargar un programa)**
> la ventaja que tiene es que no ocupas otros pines de datos.
> Yo personalmente sin la shield, prefiero conectar HC_06 en otros pines D0 y D1 y así no tengo que ir quitando y poniendo el módulo HC-06. Mira el curso DOMOTICA CON ARDUINO

> para ver un ejemplo del HC-06 no conectado al D0 D1
>
> https://libros.catedu.es/books/domotica-con-arduino/page/bluetooth
>
> https://libros.catedu.es/books/domotica-con-arduino/page/bluetooth-2

Ordenes

Si la luz está intermitente, el módulo no está vinculado, si está encendido permanente, ya está vinculado.

Una vez vinculado, la orden es sencilla: ***dato = Serial.read();***

donde dato es tipo byte : ***byte dato;***

> **Recuerda**, si estas usando los 2 mismos pines que Arduino usa para la comunicación USB con el ordenador (D0, D1), **no puedes usar el monitor serie** ni para cargar un programa ni para visualizar los datos utilizando el Bluetooth. Igualmente la velocidad tiene que ser igual para entenderse, no pueden ser diferentes.

APP Arduino Bluetooth Control

La APP en el móvil

La comunicación con la placa en este caso es muy sencilla, pues estamos empleando el Bluetooth como esclavo, es decir, será como receptor de datos nuestra placa electrónica (slave) y el móvil (master) como emisor de datos.

En nuestro caso usaremos caracteres (bytes) que enviaremos desde un master, como un teléfono móvil. Hay muchas aplicaciones gratuitas para enviar datos. Podemos usar **cualquier APP que emita un código por Bluetooth**. hay muchos, nosotros hemos elegido este :

Arduino Bluetooth Control

Esta APP es muy completa y configurable, aquí para descargarla de Google Play.

El código de programa que tenemos que cargar en la placa se basa en escuchar de forma continua el puerto serie. **Cuando llegue el dato, se ejecutará la acción que le indiquemos**. ¡¡así de sencillo !!

https://giphy.com/embed/xTiIzoyw4Yh3mRM5DG

Vincular móvil

Hay que vincular nuestro móvil y nuestra APP de Android con el Arduino, para ello sigue este sencillo tutorial:

Reto: comunicación Bluetooth

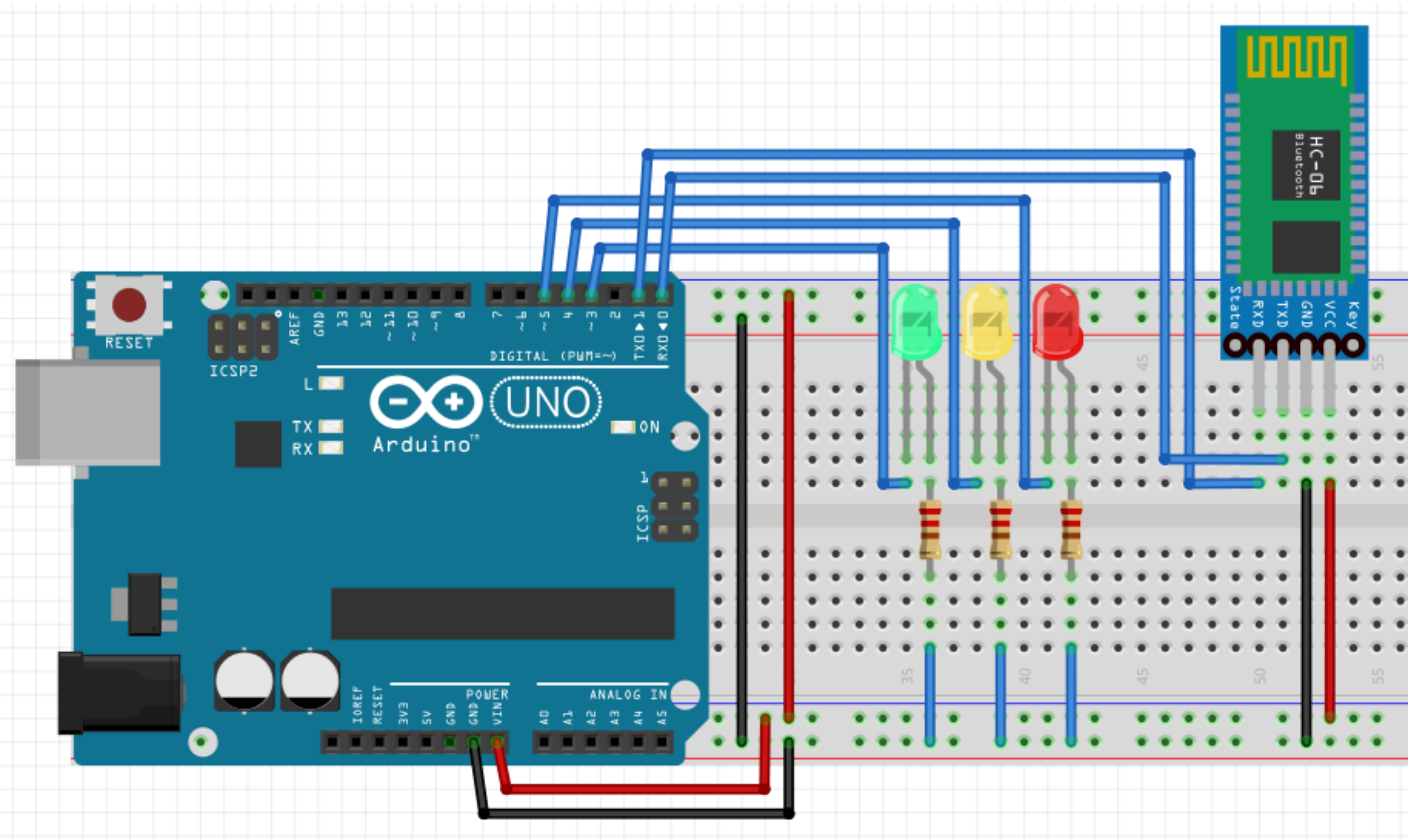
Montaje 21 Encender LEDs

El objetivo de esta práctica es encender los LEDs de EDUBASICA con el móvil:

- Cuando se pulsa la **flecha arriba**, la APP lanza el dato **U** y tiene que encenderse el led **ROJO.**
- Cuando se pulsa el botón **flecha derecha**, la APP lanza el dato **R**y tiene que encenderse el led **AMARILLO.**
- Cuando se pulsa la **flecha abajo**, la APP lanza el dato **D**y tiene que encenderse el led **VERDE.**
- Cuando se pulsa la **flecha izquierda**, la APP lanza el dato **L**y tienen que apagarse todos.

Montaje 21 SIN EDUBASICA

No pasa nada, con una placa Protoboard pon 3 leds en D3,D4 y D5 y el módulo Bluetooth.



Montaje 21 CON EDUBÁSICA

No hay que hacer nada especial, sólo conectar el módulo Bluetooth

Montaje 21 RESULTADO

> El vídeo está realizado con otra APP ya desfasada, pero sirve igual de ejemplo, es increíble, sólo pasó un mes !

https://www.youtube.com/embed/b6YviyYLsTk

Montaje 21 PROGRAMA

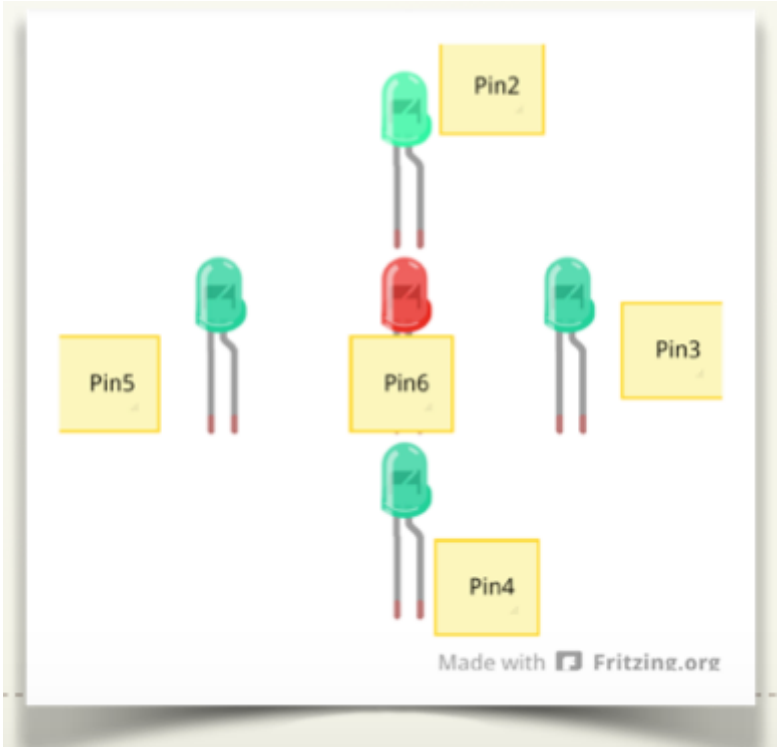
Este es el programa que tienes que cargar en el Arduino. Súbelo, empareja tu móvil tal y como hemos visto anteriormente y conseguirás que **la APP** encienda los leds como en el vídeo.

https://create.arduino.cc/editor/javierquintana/1724fd04-6b04-4c26-8cb1-1da9f8779f21/preview

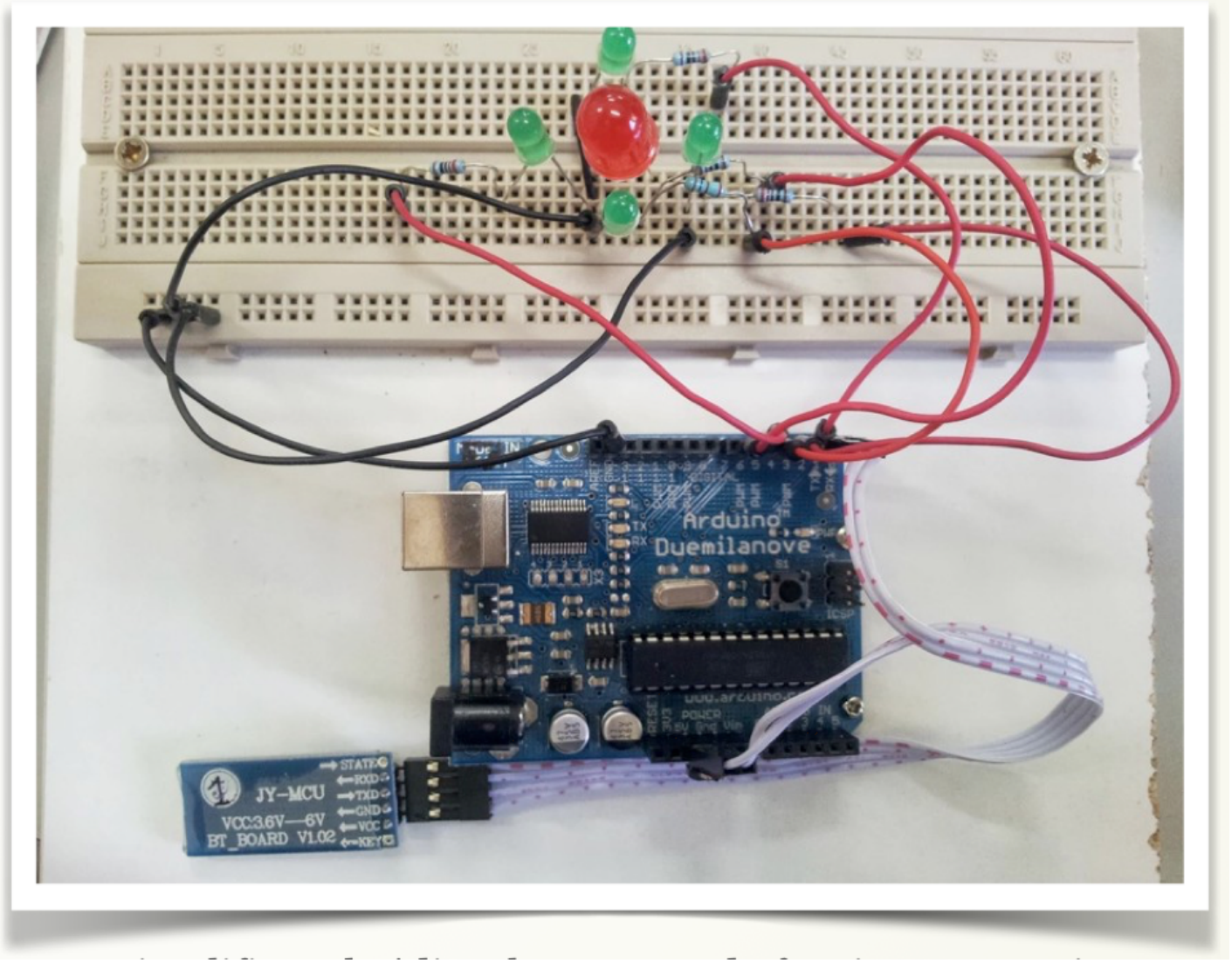
https://create.arduino.cc/editor/javierquintana/1724fd04-6b04-4c26-8cb1-1da9f8779f21/preview?embed

Montaje 22 Encender LEDs sin EDUBASICA

Este ejemplo es más avanzado, pero no se puede hacer con edubásica: Vamos a ver un ejemplo implementando un **mosaico de LEDs**:



Las conexiones serán las siguientes:



El objetivo es que según la tecla que presionemos en la aplicación “Blue Control”, se encenderá el led correspondiente: (arriba, abajo, izquierda, derecha y centro). Además si pulsamos alguno de los botones laterales, los leds deberán realizar una animación de todos los leds:

- Encendido de los leds en sentido horario.
- Encendido de los leds en sentido antihorario.
- Encendido intermitente de los leds exteriores y el interior.
- Encendido intermitente de todos los leds.

INVENTA MÁS ANIMACIONES PARA INCLUIRLAS EN LOS BOTONES QUE SOBRAN EN LA APLICACIÓN

Para simplificar el código, hemos creado funciones para ejecutar cada una de las animaciones, estas funciones están al final del programa.

La lectura se hace mediante 2 funciones:

- La función **Serial.available()** nos indica si hay un dato disponible en el puerto serie (verdadero/falso)
- Con la función **dato = Serial.read();** guardamos el dato en una variable (de typo byte)

Con esto tendremos el código ASCII del caracter enviado por el maestro, por ejemplo si hemos enviado una A tendremos el 65, B = 66, a = 97, b = 98, ... (ascii.cl/es/)

Lo único que nos queda es comparar el dato recibido y elegir la acción que tiene que hacer Arduino.

https://create.arduino.cc/editor/javierquintana/6d9fd0f4-369a-486f-86ab-10098b3815c0/preview

https://create.arduino.cc/editor/javierquintana/6d9fd0f4-369a-486f-86ab-10098b3815c0/preview?embed

Montaje 23 Configuración no típica: HC-06 no conectado al D0 y D1 y otras configuraciones

Si quieres modificar cosas como la velocidad de conexión, el nombre o la contraseña de tu módulo, aquí te dejamos un código para que subas a tu arduino y mediante el monitor serie lo configures.

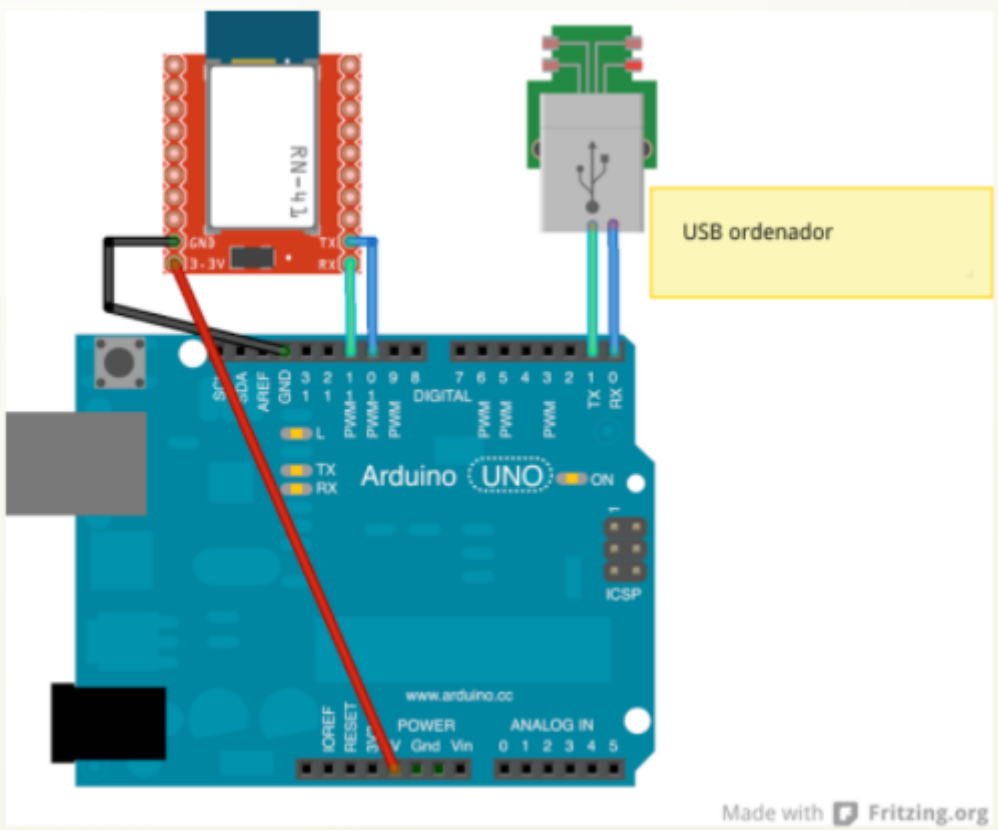
> Nota: No se pretende que realices el montaje, pero creemos que es importante que veas que el módulo Bluetooth no es cerrado en su configuración.

Para ello hemos creado un nuevo puerto serie para que no interfiera con el USB y podamos usarlo simultáneamente, lo haremos **en las patillas 10 y 11. LUEGO NO PUEDES USAR LA SHIELD**

EDUBÁSICA.

Deberás conectar el módulo como ves en la figura, y luego cargar el código. Una vez súbido, abre la consola serie y (EN MAYÚSCULAS) ejecuta los comandos que necesites.

Una vez finalizado, puedes desconectar el módulo BT y usarlo con normalidad.



https://create.arduino.cc/editor/javierquintana/b175c6e8-9e49-493d-9c79-705a663975df/preview

https://create.arduino.cc/editor/javierquintana/b175c6e8-9e49-493d-9c79-705a663975df/preview?embed

Bluetooth maestro HC-05

El **HC-05** puede funcionar como maestro y esclavo. La diferencia con el módulo **HC-06** que sólo funciona como esclavo. El **HC-05** es por lo tanto un módulo técnicamente superior. Físicamente el **HC-05** tiene 6 pines mientras que el **HC-06** sólo 4. Esto le permite al **HC-05** trabajar con distintos modos de funcionamiento más avanzados, (ver https://fgcoca.github.io/BlueTooth-HC-05-y-HC-06/modulos/)

Para acceder a este modo especial en el master lo podemos hacer de 2 formas:

1. Conectando Key a 3.3v y encender el módulo. Así funciona a 38400 baudios.
2. Encendiendo el módulo y después conectando el key a 3.3v. Así funciona a 9600 baudios, (es más sencillo pues es el que usa por defecto).

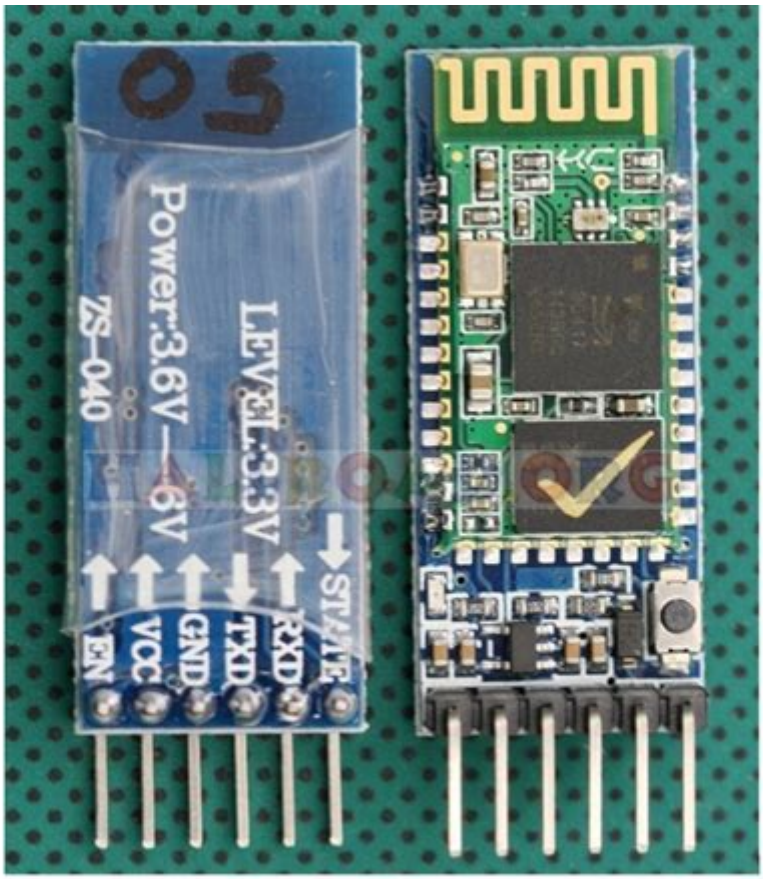
Los comandos AT en HC-05, al contrario que en el HC-06 (esclavo), que es el que tendrá mucha gente , tienen que llevar el símbolo "=", por ejemplo:

En HC-06: AT+NAMEnombre

En HC-05: AT+NAME=nombre

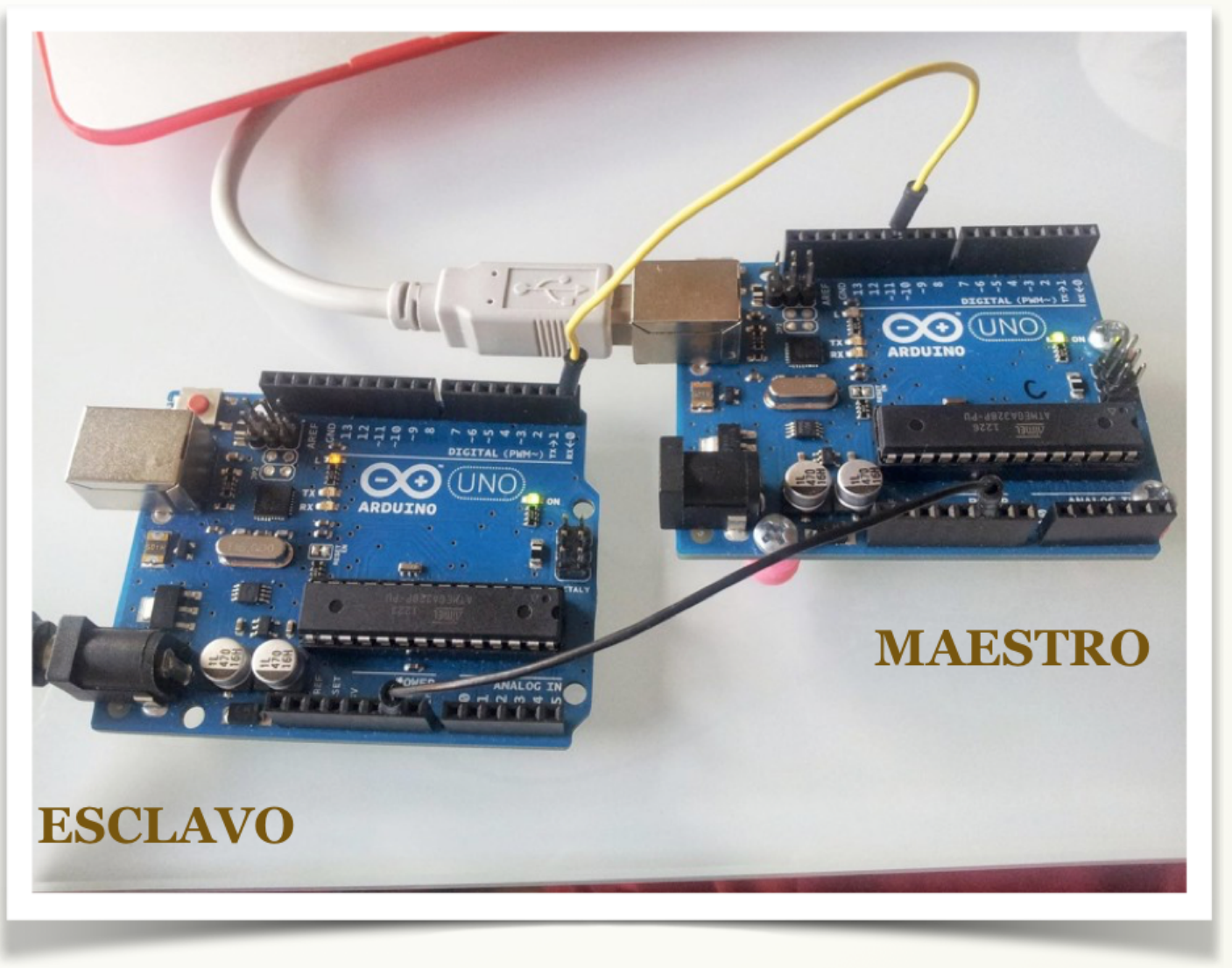
El datasheet indica que por defecto vienen con el modo CMODE=1 (para conectarse a cualquier esclavo disponible), sin embargo hay que comporbarlo (AT+CMODE?) por si tienen el CMODE=0 por lo que se intenta conectar al último emparejado, (en este caso no se emparejaría con ningún esclavo), así que hay que cambiar el CMODE con AT+CMODE=1)

El HC-05 tiene 6 pines:



HC-05

Por último comunicación Arduino-Arduino



Podemos conectar dos placas Arduino de distintas maneras: Bluetooth, Xbee, Ethernet, WIFI... Pero la forma más sencilla es aprovechar la conexión para la comunicación serie que ya posee Arduino.

Conocimiento previo

- Programación básica de Arduino.
- Bucles **for,** sentencias if-else, switch-case.

Objetivos

- Comunicación serie.
- Configuración maestro / esclavo.
- Crear un nuevo puerto serie.

Lista de materiales:

- 2 placas Arduino.

SI NO DISPONES DE DOS PLACAS DE ARDUINO, **TE PROPONEMOS SIMULARLO**

Montaje 24: Conectar dos Arduinos

El proceso es parecido al bluetooth. Aquí queremos que una Arduino envíe (MAESTRO) y que otra reciba (ESCLAVO), así que cada una correrá un programa distinto.

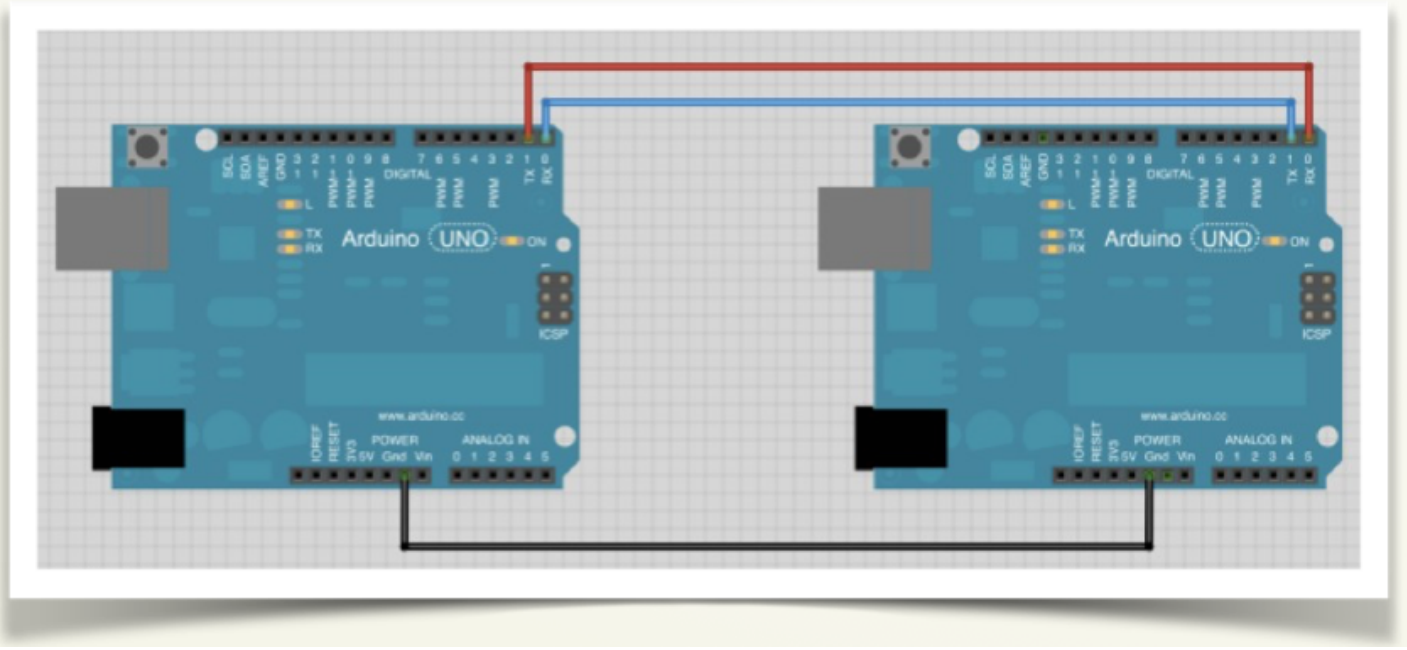
Podemos hacer también que ambas tarjetas envíen y reciban datos, para ello, la modificación sobre lo que expongo aquí serían muy sencillas, (básicamente copiar y pegar los trozos de código intercambiados).

Conexiones:

Usaremos los pines estándar de comunicación serie de Arduino:

- 0 : RX (pin por el que RECIBE los datos serie)
- 1 : TX (pin por el que ENVÍA los datos serie)

Para comunicación en 2 direcciones: los 2 pueden enviar / recibir. Las conexiones TX/RX se intercambian (lo que uno envía -TX- tiene que entrar en el otro -RX-). Cualquiera de las 2 puede ser Maestro o Esclavo. **IMPORTANTE**: Conectar ambas GND de las placas.



En el siguiente ejemplo , el maestro, cada 3 segundos envía un carácter al esclavo.

- Si envía una "r", el esclavo hará parpadear su led (d13) rápido.
- Si envía una "l", el esclavo hará parpadear su led (d13) lento.

El programa para el Arduino MAESTRO es:

https://create.arduino.cc/editor/javierquintana/4e485b39-a075-416c-a951-4c7caa4edcc0/preview

https://create.arduino.cc/editor/javierquintana/4e485b39-a075-416c-a951-4c7caa4edcc0/preview?embed

El programa para el Arduino ESCLAVO es:

https://create.arduino.cc/editor/javierquintana/605a2867-7265-4e29-8253-d311696dbe05/preview

https://create.arduino.cc/editor/javierquintana/605a2867-7265-4e29-8253-d311696dbe05/preview?embed

Si no tienes dos ARDUINOS

Puedes hacerlo con una simulación en https://www.tinkercad.com en nuestro caso este fue el resultado:

Lo tienes aquí https://www.tinkercad.com/things/dMPodSrbw0n-comunicacion-maestro-esclavo/editel?sharecode=FKrzh6UQa4CZ2UL4ma2WfTuJ3aIRTXRyNJtCWJAAvww

Y además lo puedes embeber en tu blog, página web... por ejemplo lo tienes aquí, ¡¡pincha en simulación!!

https://www.tinkercad.com/embed/b44kP2GnRXy?editbtn=1

Si n lo ves claro, pincha en el vídeo

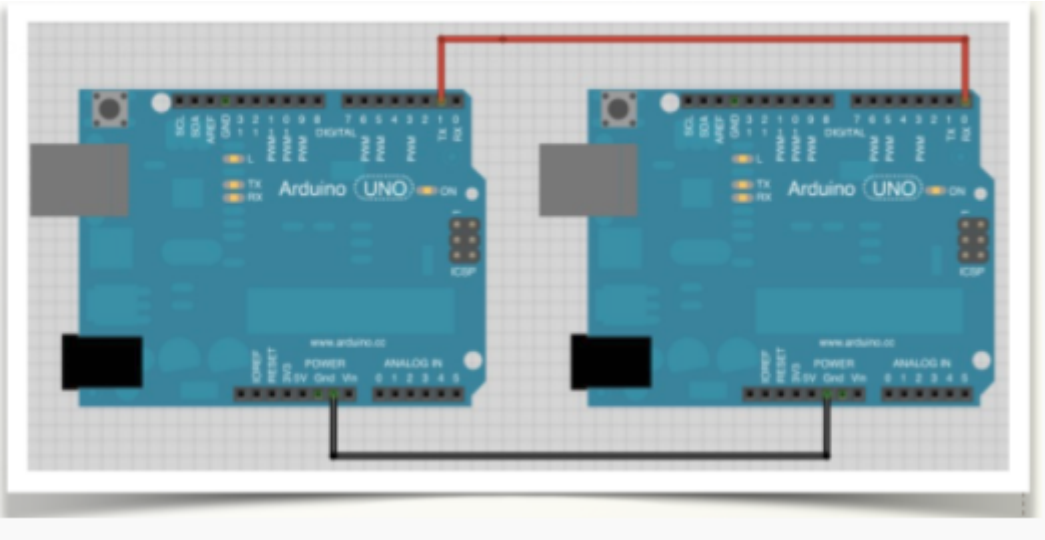
https://www.youtube.com/embed/GNtVo0xP9mA?rel=0

Si tienes dos ARDUINOS

Pues a disfrutar de tu "Red particular" :

https://www.youtube.com/embed/VmFl1mXa65U?rel=0

Otras conexiones



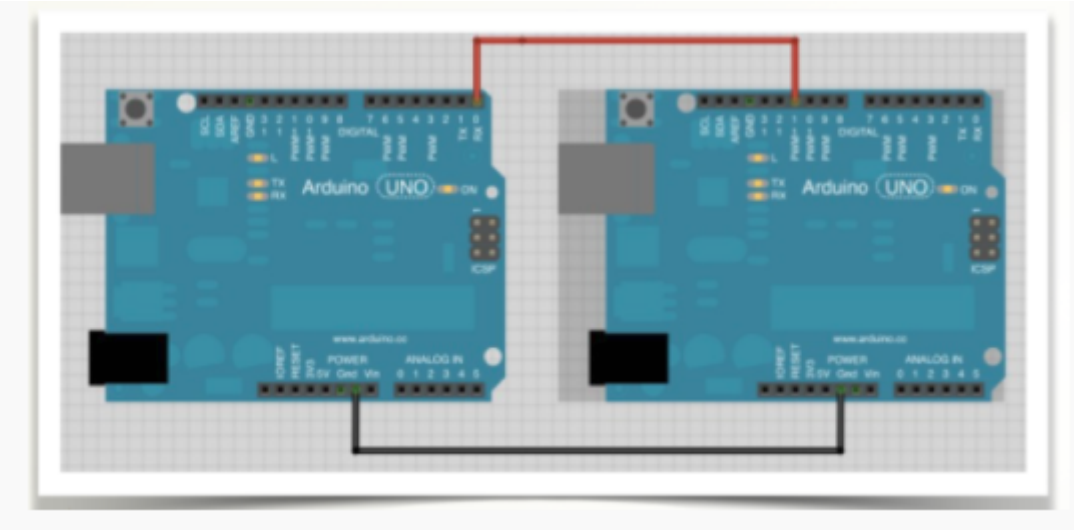
maestro(izquierda) -> esclavo(derecha) sólo necesitamos 1 conexión:

NOTA:

Ocurre que esos pines también los usa para comunicarse por USB cuando está conectado al ordenador, de manera que si queremos tener ambas conexiones (USB/trasmisión serie) deberemos crear una nueva conexión serie (en una conexión software). Sólo podemos conectar 2 Arduinos pues sólo hay un puerto de serie en cada uno de ellos. Aunque la conexión es en un sentido, es necesario conectar los dos cables TX-RX y RX-TX

En este ejemplo, una de las Arduino la vamos a tener conectada al PC, por tanto, en el MAESTRO vamos a crear la conexión software serie sobre los pines 10(RX), 11(TX).

Lo puedes comprobar en la siguiente imagen:



El programa sería el siguiente en el maestro

```
///// MAESTRO
int i=0;
//CREAMOS UN NUEVO PUERTO SERIE (RX, TX)
SoftwareSerial Serie2(10,11);
void setup()
{ pinMode(13,OUTPUT);
  Serial.begin(9600);         //Inicializa puerto estándar
  Serie2.begin(9600);         //Inicializa nuevo puerto
  digitalWrite(13,LOW);
}
void loop()
{ Serie2.write("r");
  delay(3000);
  Serie2.write("l");
  delay(3000);
}
```

Y en el esclavo sería el siguiente código :

```
//////////////////////esclavo
void setup()

{ pinMode(13,OUTPUT);
  Serial.begin(9600);
}
void loop()
```

```
{ while (Serial.available())
  {
    //Guardamos en la variable dato el valor leido
     char dato= Serial.read();
     //Comprobamos el dato
     switch(dato)
     {//Si recibimos una 'r' ...
       case 'r':
       {for(int i=0; i<20 i++)
            digitalWrite(13,HIGH);
            delay(80);
            digitalWrite(13,LOW);
            delay(80);}
        break;
       }
       case 'l':
       {for(int i=0; i<10 i++)
            digitalWrite(13,HIGH);
            delay(200);
            digitalWrite(13,LOW);
            delay(200);}
        break;
}}}}
```