# Sistemas de Comunicación

## Bluetooth.

**¿Qué es?**

Bluetooth es una tecnología de comunicación entre dispositivos de corto alcance.

Bluetooth posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz.

**Historia.**

En 1994, Ericsson inició el desarrollo de esa tecnología, investigando una forma barata de comunicación inalámbrica entre el móvil y sus accesorios. Después de esas investigaciones iniciales, quedó clara la potencialidad de ese tipo de conexión. En 1998, seis grandes empresas: Sony, Nokia, Intel, Toshiba, IBM y Ericsson, realizaron un consorcio para conducir y profundizar el estudio de esa forma de conexión, formando el llamado Bluetooth Special Interest Group.

El nombre «Bluetooth» es un homenaje al rey de Dinamarca y Noruega, Harald Bltand, que en la lengua inglesa es llamado de Harold Bluetooth.

**La conexión Bluetooth**

La frecuencia dedicada a Bluetooth es una banda ISM sin licencia entre los 2,402 GHz y los 2,480 GHz. Los dispositivos compatibles que cumplen los estándares del Bluetooth SIG pueden, como dispositivos de corto alcance o Short Range Devices (SRD), enviar por este rango de frecuencias en todo el mundo y sin licencia. Para poder identificarlo sin ningún género de dudas, cada aparato está provisto de una dirección MAC de 48 bits individual.

La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s (1 Mbit/s de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores). Opera en la frecuencia de radio de 2,4 a 2,48 GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos por segundo. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz; esto permite dar seguridad y robustez.

**Comunicación Bluetooth en domótica.**

La conectividad Bluetooth permite controlar elementos del hogar como la iluminación y las persianas desde un smartphone o tablet, sin necesidad de realizar instalaciones adicionales u obras específicas, haciendo del hogar un espacio más confortable y seguro, sin inversiones adicionales.

**Pros y Contras de Bluetooth como herramienta de comunicación en la domótica.**

PROS: Un bajo consumo de energía, su simplicidad de uso y su implantación por defecto en la mayoría de dispositivos móviles y wereables lo hace sumamente accesible (físicamente ocupa muy poco por lo que se puede meter en cualquier lado prácticamente) No «gasta» datos.

CONS: Alcance corto (unos 10m por norma general aunque puede variar según versiones y escenarios de uso) Pierde mucha eficiencia con los obstáculos y la distancia. No se puede usar de forma remota. La transmisión de datos no es demasiado rápida.

**Bluetooth comunicación serial de arduino**

**Imagen**

**INFO:**

El módulo Bluetooth HC-05 nos permite conectar nuestros proyectos con Arduino a un smartphone, celular o PC de forma inalámbrica (Bluetooth), con la facilidad de operación de un puerto serial. La transmisión se realiza totalmente en forma transparente al programador, por lo que se conecta en forma directa a los pines seriales de nuestro microcontrolador preferido (respetando los niveles de voltaje, ya que el módulo se alimenta con 3.3V). Todos los parámetros del módulo se pueden configurar mediante comandos AT. La placa también incluye un regulador de 3.3V, que permite alimentar el módulo con un voltaje entre 3.6V - 6V. Este módulo es el complemento ideal para nuestros proyectos de robótica, domótica y control remoto con Arduino, PIC, Raspberry PI, ESP8266, ESP32, STM32, etc.

La comunicación Bluetooth se da entre dos tipos de dispositivos: un maestro y un esclavo

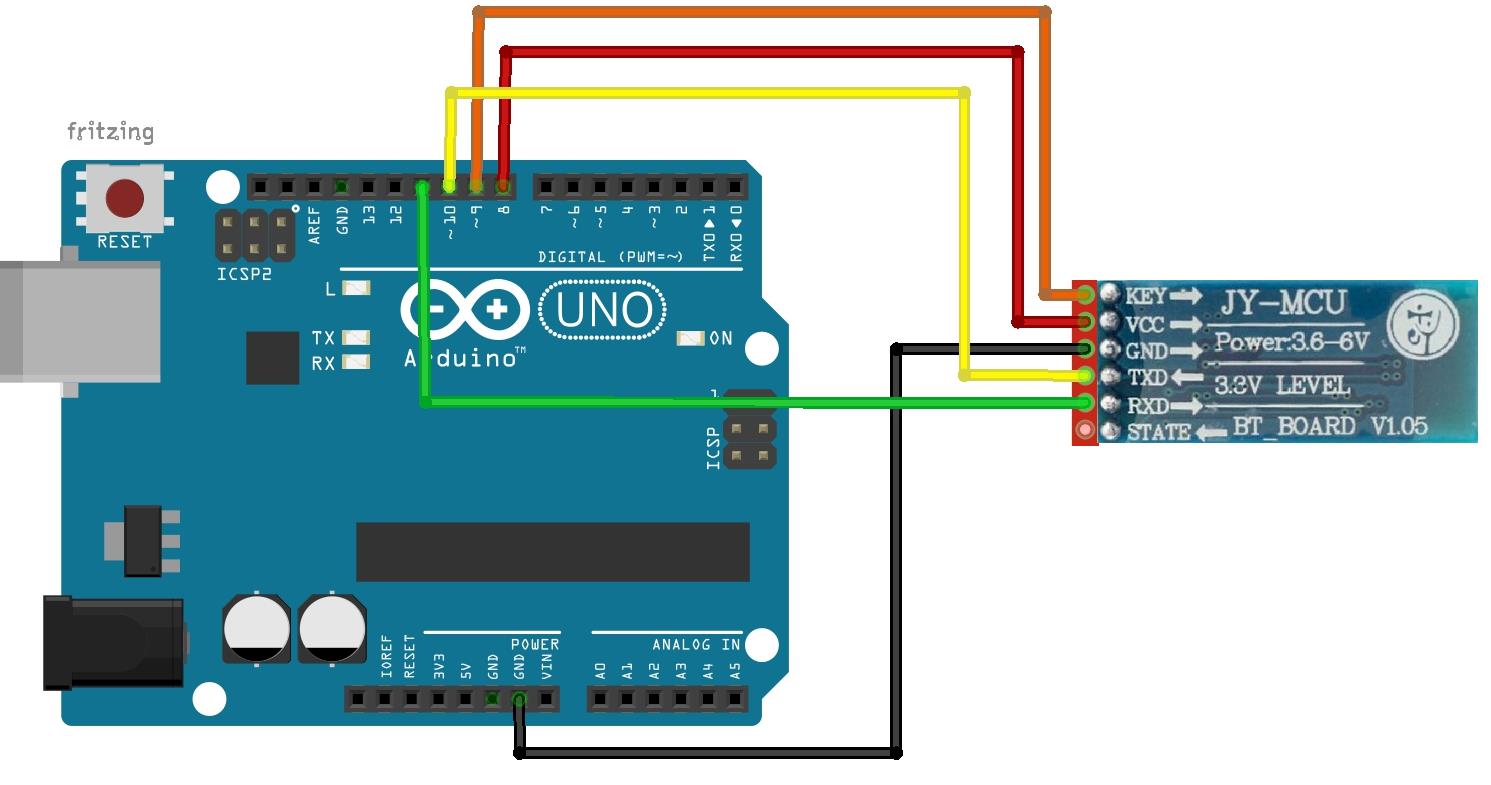
**Especificaciones:**

* Voltaje de operación: 3.6V - 6V DC
* Consumo corriente: 50mA
* Bluetooth: V2.0+EDR
* Frecuencia: Banda ISM 2.4GHz
* Modulación: GFSK(Gaussian Frequency Shift Keying)
* Potencia de transmisión: 4dBm, Class 2
* Sensibilidad: -84dBm a 0.1% BER
* Alcance 10 metros
* Interface comunicación: Serial TTL
* Velocidad de transmisión: 1200bps hasta 1.3Mbps
* Baudrate por defecto: 38400,8,1,n.
* Seguridad: Autenticación y encriptación
* Temperatura de trabajo: -20C a +75C
* Compatible con Android
* Dimensiones: 37\*16 mm
* Peso: 3.6 gramos

**Funciones:**

* Reemplaza un enlace serial cableado por un enlace inalámbrico en forma transparente
* Posee el perfil Bluetooth "Wireless Serial Port"
* Se puede configurar como **Master** o **Slave**
* Ideal para controlar nuestro proyecto Arduino, PIC, Atmel, etc. en forma inalámbrica con nuestra PC

**Conexión con arduino UNO**



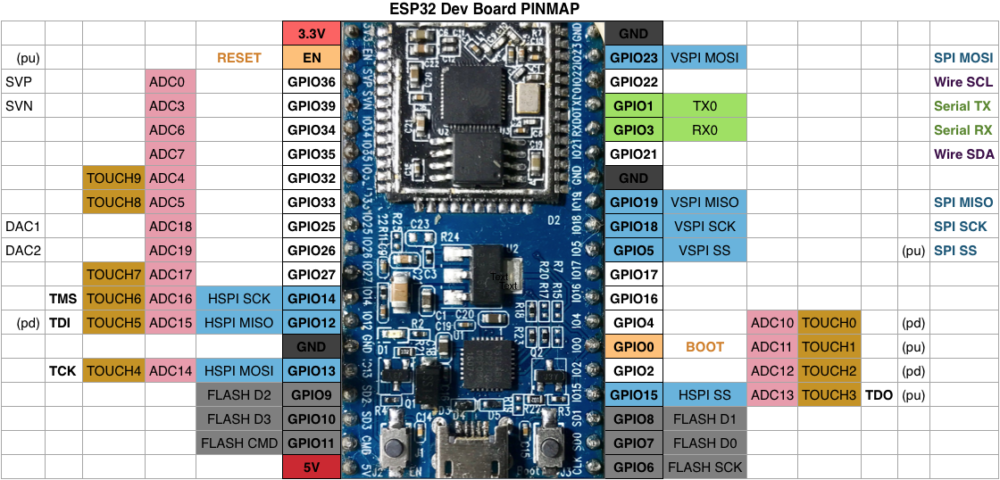
En primer lugar, para que el HC-05 entre en modo comandos AT, requiere que cuando se enciende el modulo, el pin KEY este HIGH. Por eso hemos conectado la tensión Vcc del módulo BlueTooth al pin 8 de nuestro Arduino.

El consumo del módulo es mínimo y nuestro Arduino es capaz de alimentarlo sin problemas, por eso el modulo se encenderá cuando pongamos HIGH en el pin 9. Esto nos permitirá poner en HIGH el pin digital 8, al iniciar nuestro programa y después levantar el pin 8, de este modo cuando arranque entrara sin más en el modo de comandos AT.

El resto de los pines se conectan de forma similar a lo que hicimos en la sesión anterior. Txd y Rxd se deben conectar cruzados con los pines de comunicación de Arduino, que usaremos mediante la librería software Serial.

El pin State refleja, supuestamente, la situación en la que se encuentra el modulo y por ahora no vamos a utilizarlo.

## ¿QUÉ ES EL ESP32?



El ESP32 es un SoC (System on Chip) diseñado por la compañía china Espressif y fabricado por TSMC. Integra en un único chip un procesador Tensilica Xtensa de doble núcleo de 32bits a 160Mhz (con posibilidad de hasta 240Mhz), conectividad WiFi y Bluetooth.

**Características.**

* Procesador Xtensa LX6 de 32 bits de doble núcleo
* Velocidad de 160Mhz (máximo 240 Mhz)
* Co-procesador de ultra baja energía
* Memoria 520 KiB SRAM
* Memoria flash externa hasta 16MiB
* Encriptación de la Flash
* Arranque seguro
* Pila de TCP/IP integrada
* Wifi 802.11 b/g/n 2.4GHz (soporta WFA/WPA/WPA2/WAPI)
* Bluetooth v4.2 BR/EDR y BLE
* Criptografía acelerada por hardware
* 32 pins GPIO
* Conversor analógico digital (ADC) de 12bits y 18 canales
* 2 conversores digital analógico (DAC) de 8bits
* 16 salidas PWM (LED PWM)
* 1 salida PWM para motores
* 11 conversor analógico a digital de 10 pin
* 10x sensores capacitivos (en GPIO)
* 3x UART, 4x SPI, 2x I2S, 2x I2C, CAN bus 2.0
* Controladora host SD/SDIO/CE-ATA/MMC/eMMC
* Controladora slave SDIO/SPI
* Sensor de temperatura
* Sensor de effecto Hall
* Generador de números aleatorios
* Reloj tiempo real (RTC)
* Controlador mando a distancia infrarrojos (8 canales)

**Ventajas**

Las ventajas del ESP32 frente a su predecesor no se quedan allí. Al poseer un segundo núcleo, este se dedica únicamente para manejar los eventos de WiFi (por defecto), aunque se le pueden asignar tareas específicas. Esto permite una ventaja contra el ESP8266, el cual tiene que detener ciertos eventos para procesar las actividades del WiFi. Otra de las ventajas es la posibilidad de utilizar más sensores de lecturas análogas sin la necesidad de utilizar multiplexores.

## Modulo RF 433mhz -Emisor y ReceptorMódulo De RF 433MHz

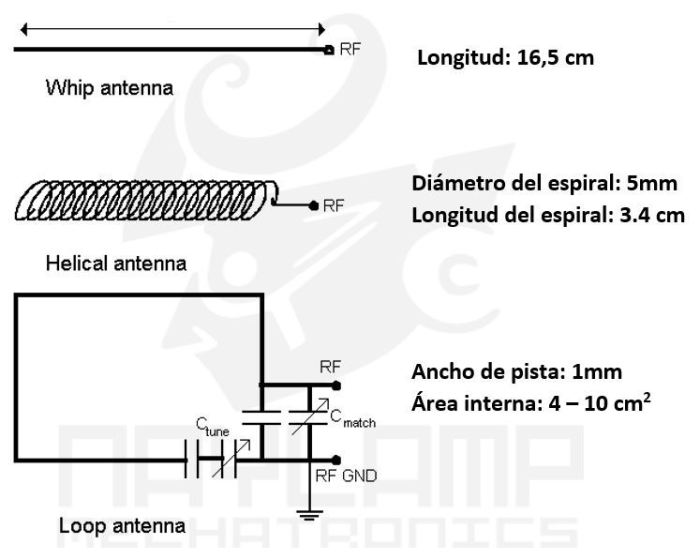
¿Qué es?

Los módulos de radio frecuencia RF 433MHz son transmisores/receptores inalámbricos que podemos usar como forma de comunicación entre procesadores como Arduino. Se compone de dos placas: el emisor (FS1000A) y el receptor (XY-MK-5V). La frecuencia de operación es de 433MHz, aunque también existen módulos similares a 315MHz. Ambas frecuencias pertenecen a bandas libres, por lo que su uso es gratuito.

La comunicación es simplex (canal único y unidireccional) y tienen baja velocidad de transmisión (típicamente 2400bps).

**Tipos De Antena Para Modulo RF 433**

La antena tiene una gran influencia sobre todo en el módulo receptor, por lo que se recomienda conectar un cable de cobre de unos 17cm de largo para alcanzar la frecuencia de 433MHz. También son muy comunes y eficientes las antenas helicoidales de 5mm de diámetro y 3.4cm de longitud. La posición de la antena también afecta la recepción de datos. Todo lo que tenes que hacer es probar cada una de las variantes hasta lograr el alcance requerido dentro de las especificaciones.



**Información General.**

La frecuencia de trabajo es de 433MHz, debido a que es una banda de libre uso. Para utilizar estos módulos basta con alimentarlos y conectar el pin de Datos por un lado del Transmisor (TX) y en otro el Receptor (RX). Para la programación no es necesario agregar ningún tipo de librería, ya que es un proceso "transparente", por lo que el programa sería igual a usar una comunicación serial (UART) entre 2 MCUs, sin embargo existen algunas librerías que nos ofrecen ciertas ventajas como: verificación de errores, redundancia, etc.

La antena tiene una gran influencia en el módulo receptor, por lo que se recomienda conectar un cable de cobre de unos 17cm de largo para alcanzar la frecuencia de 433MHz. La posición de la antena también afecta la recepción de datos. El cable de la antena debe estar debidamente aislado.

**Especificaciones Técnicas.**

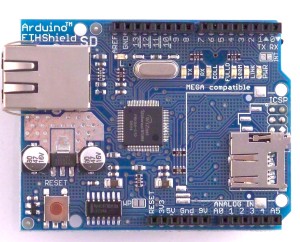
**Módulo TX:**

* Voltaje de Operación: 3.5V~12V
* Entrada de datos: TTL
* Corriente de trabajo: Máximo <= 40mA (12V), mínimo <= 9 mA (3.5V);
* Alcance: 20 metros~200 metros (a mayor voltaje, mayor potencia de transmisión)
* Potencia de transmisión: 10 mW (5V);
* Frecuencia de transmisión: 433MHz
* Tasa de transferencia de datos: 4KB/seg
* Baud Rate recomendado: 2400
* Modulación: ASK/OOK
* Antena (no incluida): Alambre de cobre de 25cm (recomendado)
* Pines de salida: DATA(TX)/VCC/GND

**Módulo RX:**

* Voltaje de Operación: 5V DC
* Consumo de corriente: 4mA
* Sensibilidad del receptor: -105dB
* Frecuencia de recepción: 433MHz
* Modulación: ASK/OOK
* Principio de funcionamiento: Receptor Superregenerativo
* Antena (no incluida): Alambre de cobre de 25cm~32cm (recomendado en forma de espiral)
* Pines: GND/DATA(RX)/VCC

## ethernet shield



**¿Qué es?**

El módulo Arduino Shield Ethernet permite a una placa Arduino conectarse a internet. Se basa en el chip de ethernet Wiznet W5100 (Hoja de datosW5100\_Datasheet). El chip Wiznet W5100 ofrece una red (IP) capaz de usar TCP y UDP. Soporta hasta cuatro conexiones de socket simultáneas. Utilice la biblioteca de Ethernet para escribir bocetos que se conectan a Internet a través del escudo. El escudo de Ethernet se conecta a una placa Arduino usando largas cabeceras wire-wrap que se extienden a través del escudo. Esto mantiene la disposición de las clavijas intacto y permite que otro escudo pueda ser apilado en la parte superior.

**Características**

La shield contiene varios LEDs para información:

* ON: indica que la placa y la shield están alimentadas
* LINK: indica la presencia de un enlace de red y parpadea cuando la shield envía o recibe datos
* 100M: indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
* RX: parpadea cuando el shield recibe datos
* TX: parpadea cuando el shield envía datos

El jumper soldado marcado como “INT” puede ser conectado para permitir a la placa Arduino recibir notificaciones de eventos por interrupción desde el W5100, pero esto no está soportado por la librería Ethernet.

El jumper conecta el pin INT del W5100 al pin digital 2 de Arduino.

Opera a 5V suministrados desde la placa de Arduino

El controlador ethernet es el W5100 con 16K de buffer interno. No consume memoria.

## 

## Xbee



**¿Qué es?**

**Los XBee´s son pequeñas radios que pueden comunicarse de forma inalámbrica unas con otras. Pueden hacer cosas simples, como reemplazar un par de cables en una comunicación serial, lo cual es genial cuando deseas crear, por ejemplo, un vehículo radiocontrolado. Estas radios además cuentan con entradas y salidas digitales y analógicas que puedes controlar e incluso puedes conectar sensores y leer sus mediciones en forma remota. Una gran ventaja de estas radios es su bajo consumo además de su posibilidad de realizar conexiones punto a punto, punto a multipunto y mesh.**

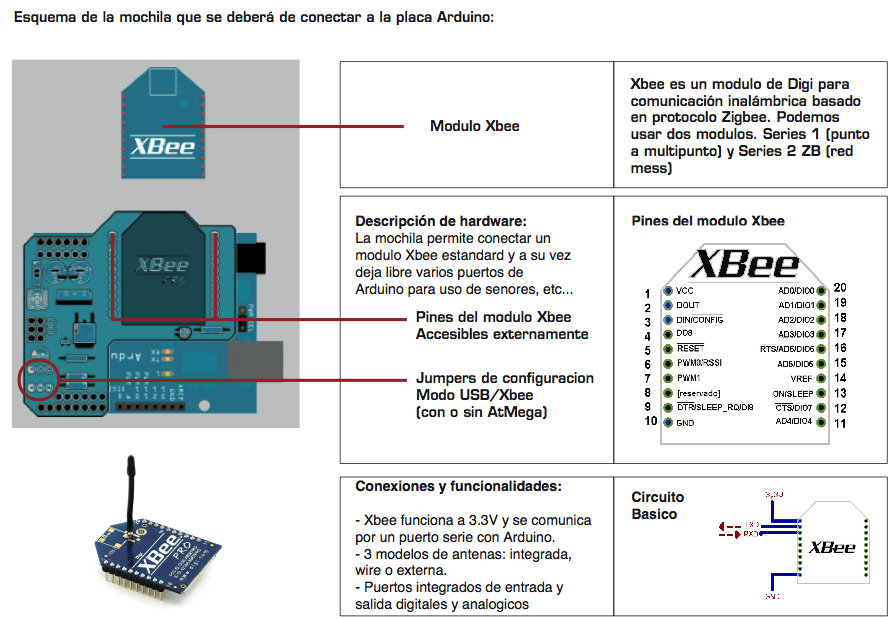
**Lós módulos XBee utilizan el protocolo IEEE 802.15.4 mejor conocido como ZigBee.**

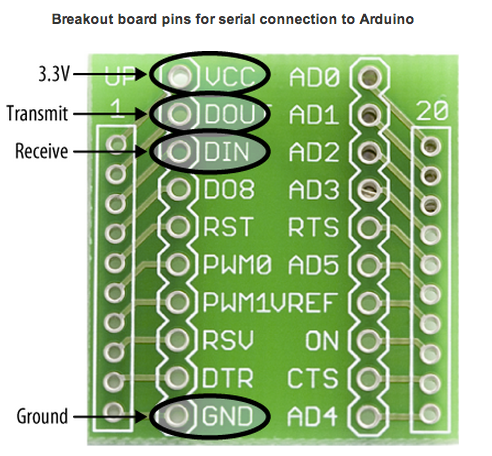
**Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE\_802.15.4. Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philips, ODEM do, Invensys, entre otras), muchas de ellas fabricantes de semiconductores.**

**Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticos.**

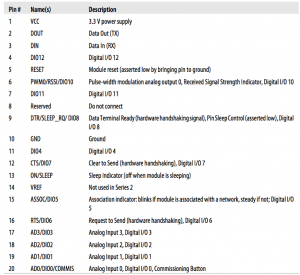
**APLICACIONES**

Cada módulo Zigbee, al igual que ocurre con las direcciones MAC de los dispositivos ethernet, tiene una dirección única. En el caso de los módulos Zigbee cada uno de ellos tiene una dirección única de 64bits que viene grabada de fábrica. Por otro lado, la red Zigbee, utiliza parasus algoritmos de ruteo direcciones de 16 bits. Cada vez que un dispositivo se asocia a una redZigbee, el Coordinador al cual se asocia le asigna una dirección única en toda la red de 16bits. Por eso el número máximo teórico de elementos que puede haber en una red Zigbee es de 2^16 =65535, que es el nº máximo de direcciones de red que se pueden asignar. Estos módulos Xbee, pueden ser ajustados para usarse en redes de configuración punto-a-punto, punto-a-multipunto o peer-to-peer.





**Descripciones de PINES**



## Los módulos Xbee

**Modo AT**

Esta el modo de transmisión serial transparente (Modo AT), en el cual la comunicación se asemeja a lo que seria una transmisión a través de un puerto serial, ya que el dispositivo se encarga de crear la trama y el dato que llegue al pin Tx sera enviado de forma inalámbrica, por lo cual se considera como el modo mas sencillo para utilizar estos nodos, su principal desventaja es que para enviar información a distintos nodos es necesario entrar constantemente al modo configuración para cambiar la dirección de destino.

**Modo API**

El otro modo de comunicación se conoce como Modo API, en este caso un microcontrolador externo se debe encargar de crear una trama especifica al tipo de información que se va a enviar, este modo es recomendado para redes muy grandes donde no se puede perder tiempo entrando y saliendo del modo configuración de los dispositivos. Para redes con topología en Malla este es el modo a utilizar.

## **Características:**

-Buen Alcance: hasta 300ft (100 mts) en línea vista para los módulos Xbee y hasta 1 milla (1.6 Km) para los módulos Xbee Pro.

-9 entradas/salidas con entradas analógicas y digitales.

-Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo sleep.

-Interfaz serial.

-65,000 direcciones para cada uno de los 16 canales disponibles. Se pueden tener muchos de estos dispositivos en una misma red.

-Fáciles de integrar.

## Domótica

**¿Qué es?**

Un sistema domótico es capaz de recoger información proveniente de unos sensores o entradas, procesarla y emitir órdenes a unos actuadores o salidas. El sistema puede acceder a redes exteriores de comunicación o información.

La domótica permite dar respuesta a los requerimientos que plantean estos cambios sociales y las nuevas tendencias de nuestra forma de vida, facilitando el diseño de casas y hogares más humanos, más personales, polifuncionales y flexibles.

**Funciones de la domótica**

La domótica proporciona una gran variedad de ventajas al implementarla como lo son:

* Ahorró de energía.
* Accesibilidad
* Seguridad
* Comunicación

En resumen la domótica consiste en crear un ecosistema mediante tecnológicas de comunicación, wifi, etherne, bluetooth e inflarojo, lo cual nos permiteria poder controlar ciertos aparatos con el celular, además de poder tener un control sobre la casa o el lugar donde se implemente.

## Comunicación serial

## **¿QUÉ ES EL PUERTO SERIE?**

## Un puerto es el nombre genérico con que denominamos a los interfaces, físicos o virtuales, que permiten la comunicación entre dos ordenadores o dispositivos.

## Un puerto serie envía la información mediante una secuencia de bits. Para ello se necesitan al menos dos conectores para realizar la comunicación de datos, RX (recepción) y TX (transmisión). No obstante, pueden existir otros conductores para referencia de tensión, sincronismo de reloj, etc.

## Por el contrario, un puerto paralelo envía la información mediante múltiples canales de forma simultánea. Para ello necesita un número superior de conductores de comunicación, que varían en función del tipo de puerto. Igualmente existe la posibilidad de conductores adicionales además de los de comunicación.

**ARDUINO Y EL PUERTO SERIE**

## Las placas Arduino UNO y Mini Pro disponen de una unidad UART que operan a nivel TTL 0V / 5V, por lo que son directamente compatibles con la conexión USB.

## **Los puertos serie están físicamente unidos a distintos pines** de la placa Arduino. Lógicamente, mientras usamos los puertos de serie no podemos usar como entradas o salidas digitales los pines asociados con el puerto serie en uso.

¿Cómo funciona?

Arduino IDE nos proporciona una herramienta que nos permite enviar y visualizar los datos que se manejan a través del puerto Serie. Dicha herramienta se conoce como Monitor Serial y se puede encontrar en el menú de herramientas, en la opción «Monitor Serial». Es la forma más simple que existe para establecer la comunicación serial con Arduino.

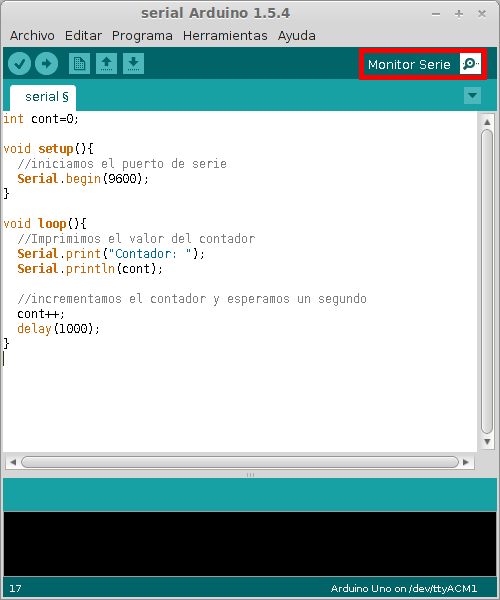
Para iniciar la comunicación serial con Arduino utilizando el Monitor Serial debemos establecer algunos comandos en el Arduino IDE y luego subirlos al microcontrolador.

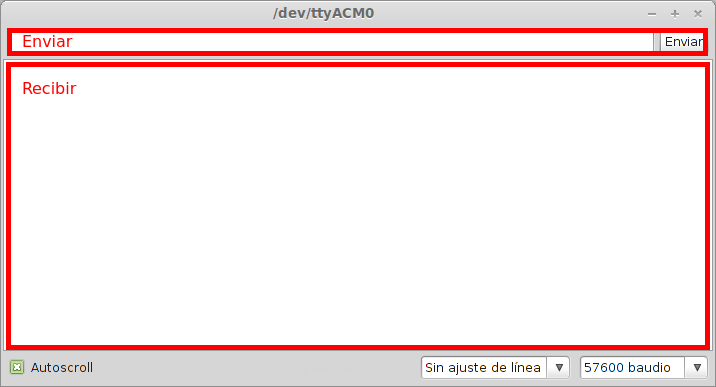
Ventajas:

* Sólo requiere dos cables
* No requiere una señal de reloj
* Posee bits de paridad que evitan errores en la comunicación
* Método bien documentado y extensamente utilizado

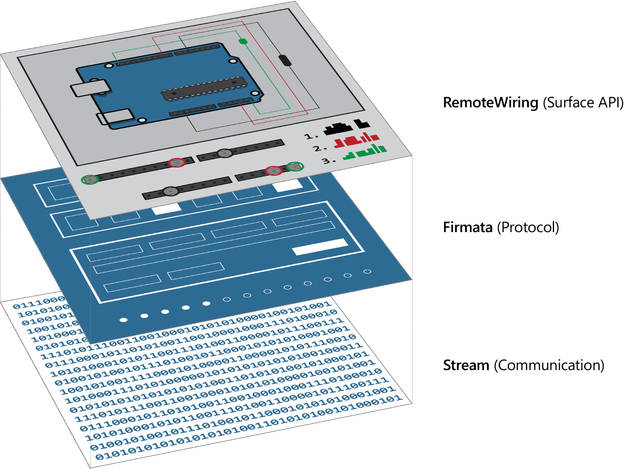
Desventajas

* No permite tener varios dispositivos esclavos
* No permite la comunicación simultánea entre maestro esclavo, ya que comparten el mismo buffer de datos
* Los dispositivos comúnmente se encuentran a cortas distancias.





## Firmata



**¿Qué es?**

Firmata es un protocolo genérico para la comunicación con microcontroladores desde software instalado en un ordenador. Este protocolo se puede implementar en cualquier arquitectura de microcontroladores, así como en cualquier paquete de software.

**Función.**

El objetivo de firmata es permitir controlar completamente Arduino desde software instalado en un ordenador, sin escribir una sola línea de código de Arduino.

**Protocolo Firmata:**

El protocolo Firmata es un protocolo para comunicar con microcontroladores desde cualquier software o cualquier ordenador conectado. El objetivo primordial de este protocolo es hacer que el microcontrolador sea una extensión de nuestro entorno de desarrollo. Fue diseñado para ser abierto y flexible para que pueda ser soportado por cualquier lenguaje de programación, también debe ser simple de implementar en el microcontrolador y en el ordenador.

La versión original es la Standard\_Firmata que viene se incluye dentro de las versiones del entorno oficial de Arduino y Wiring. Hay que recordar que el entorno de desarrollo de Arduino se basa en la plataforma Wiring. Esta versión incluye soporte para las siguientes características:

* Entradas y salidas analógicas
* Entradas analógicas
* Salidas PWM
* Conmutación entre entradas y salidas analógicas
* Servomotores
* Matrices de LEDs
* I2C