



UNIVERSIDAD
DE LIMA

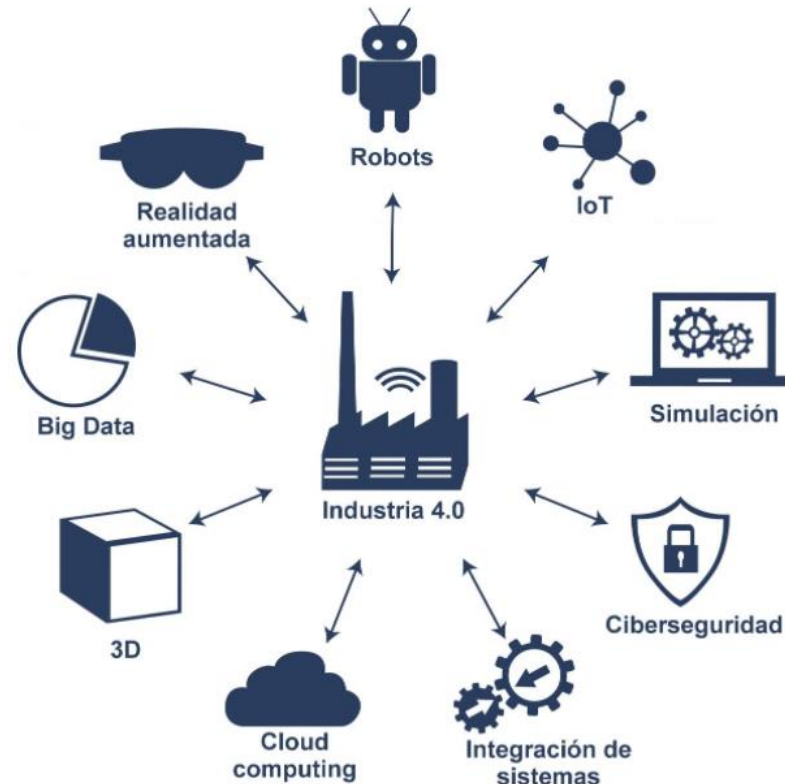
PROTOSCOLOS DE COMUNICACION

Introducción

- Es un conjunto de reglas que gobierna el intercambio ordenado de datos dentro de la red
- Los elementos básicos de un protocolo de comunicaciones son:
 - Un conjunto de símbolos llamados conjunto de caracteres,
 - Un conjunto de reglas para la secuencia y sincronización de los mensajes contruidos a partir del conjunto de caracteres
 - Los procedimientos para determinar cuando ha ocurrido un error en la transmisión y como corregir el error

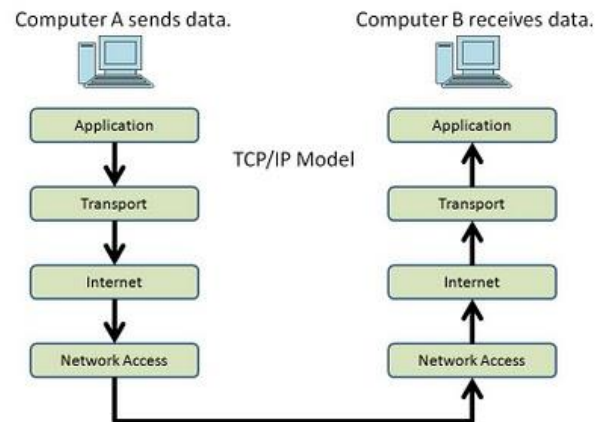
Tipos

- Los protocolos gestionan dos niveles de comunicación distintos:
 - Las reglas de alto nivel definen como se comunican las aplicaciones
 - Las de bajo nivel definen como se transmiten las señales.



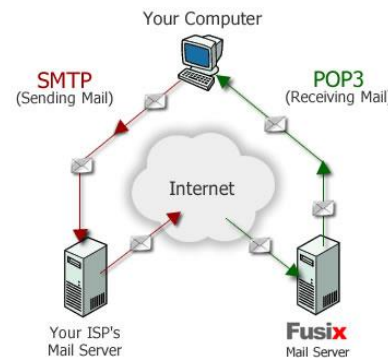
TCP/IP

- Transmission Control Protocol / Internet Protocol
- Se podría definir como el conjunto de protocolos básicos de comunicación que permite la transmisión de información de redes de computadoras.
- Fue desarrollado para interconectar los nodos de las redes ARPANET, PRNET (packet radio) y SATNET (packet satellite).
- Las máquinas en internet son llamados "HOSTS" o nodos



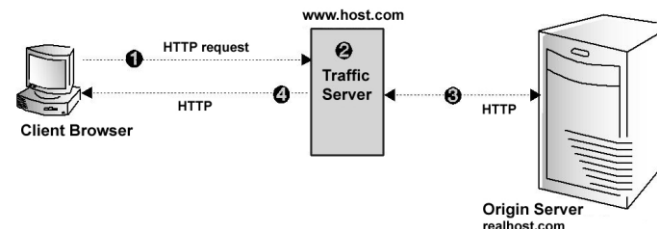
SMTP

- Simple Mail Transfer Protocol o protocolo para transferencia simple de correo
- Es un conjunto de reglas que rigen el formato y la transferencia de datos en un envío de correo electrónico.
- Muy usado para obtener los mensajes de correo electrónico almacenados en un servidor remoto.



HTTP

- Hypertext Transfer Protocol o protocolo de transferencia de hipertexto.
- Diseñado para recuperar información y llevar a cabo búsquedas indexadas.
- No solo permite la transferencia de textos HTML, sino de un amplio y extensible conjunto de formatos.
- Algunas funciones especiales de la web:
 - Resolver los problemas planteados por un sistema hipermedial.



MQTT



https://www.youtube.com/watch?v=-fQzbyLqsMc&ab_channel=INNOVADOMOTICS

MQTT

- MQTT son las siglas MQ Telemetry Transport, aunque en primer lugar fue conocido como Message Queing Telemetry Transport.
- Es un protocolo de comunicación M2M (machine-to-machine) de tipo message queue.

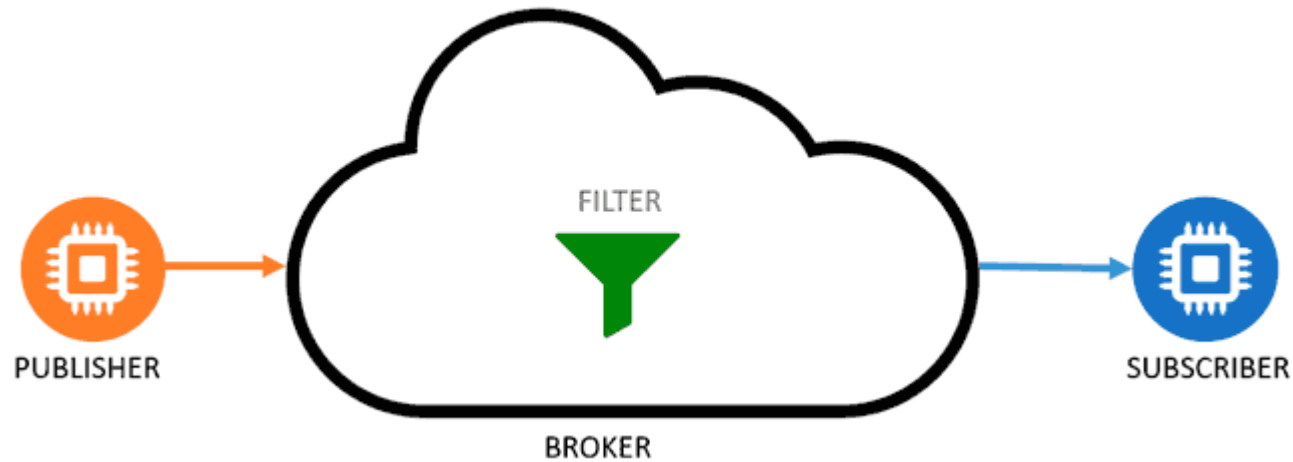


MQTT

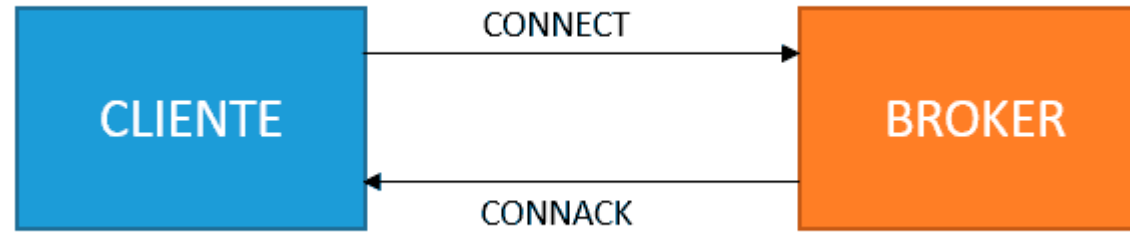
- Está basado en la pila TCP/IP como base para la comunicación.
- En el caso de MQTT cada conexión se mantiene abierta y se "reutiliza" en cada comunicación
- Inicialmente era un formato propietario, en 2010 fue liberado y pasó a ser un estándar en 2014 según la OASIS (Organization for the Advancement of Structured Information Standards).

MQTT FUNCIONAMIENTO

- Un servicio de mensajería push con patrón publicador/suscriptor (pub-sub).



MQTT FUNCIONAMIENTO



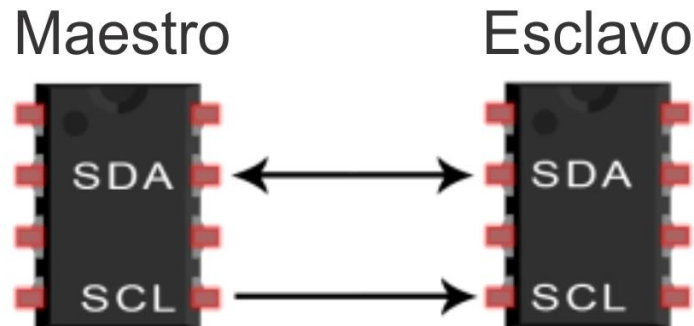
I2C



https://www.youtube.com/watch?v=IIFoJ7AyUIs&ab_channel=IgnacioAguilera

I2C

- I2C significa Circuito Interintegrado (Inter-Integrated Circuit)
- protocolo de comunicación serial desarrollado por Phillips Semiconductors
- El protocolo I2C toma e integra lo mejor de los protocolos SPI y UART.
- Con el protocolo I2C podemos tener a varios maestros controlando uno o múltiples esclavos.



I2C

- SDA – Serial Data. Es la vía de comunicación entre el maestro y el esclavo para enviarse información.
- SCL – Serial Clock. Es la vía por donde viaja la señal de reloj.



I2C

Ventajas.

- Sólo utiliza 2 cables de comunicación.
- Soporta múltiples Maestros y múltiples Esclavos, haciendo las conexiones adecuadas.
- Hay confirmación de información recibida con éxito. Usando los bits ACK/NACK.
- El hardware es menos complicado que el protocolo UART.
- Es un protocolo ampliamente conocido y utilizado.

Desventajas.

- Es un más lento que el protocolo SPI.
- El tamaño de paquetes de información de transferencia está limitado a 8 bits.
- El hardware es más complicado que el protocolo SPI.

Redes inalámbrica

- Las redes inalámbricas se basan en un enlace que utiliza ondas electromagnéticas en lugar de cableado estándar
- Dependiendo del tamaño de la red o de la cobertura que proporciona, se pueden clasificar en los diferentes tipos:
 - WLAN
 - WMAN
- Existen otros protocolos inalámbricos actuales:
 - RFID
 - Bluetooth
 - Zigbee
 - LoRa

WLAN

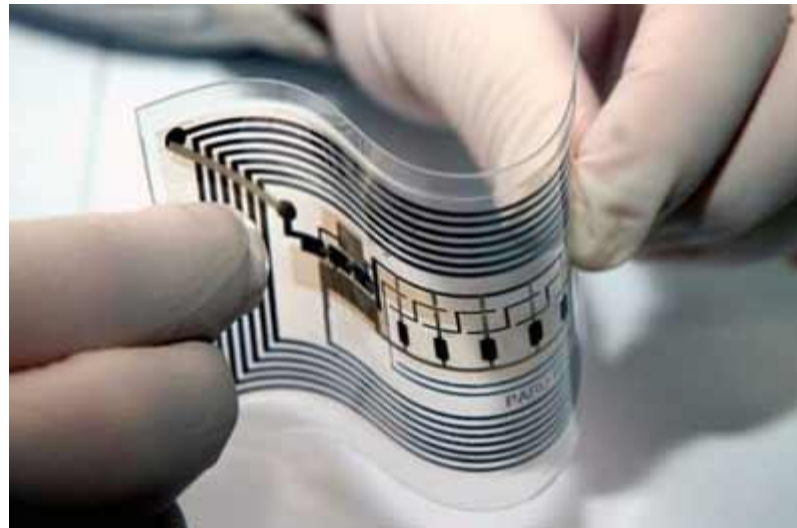
- *Wireless Local Area Network*
- En las redes de área local, se pueden encontrar tecnologías inalámbricas basadas en HiperLAN como HiperLAN2 o tecnologías basadas en WiFi, que siguen el estándar IEEE 802.11x.

WMAN

- *Wireless Metropolitan Area Network*
- Las WMAN se basan en el estándar IEEE 802.16x o WiMax, así como en LMDS (Local Multipoint Distribution Service).

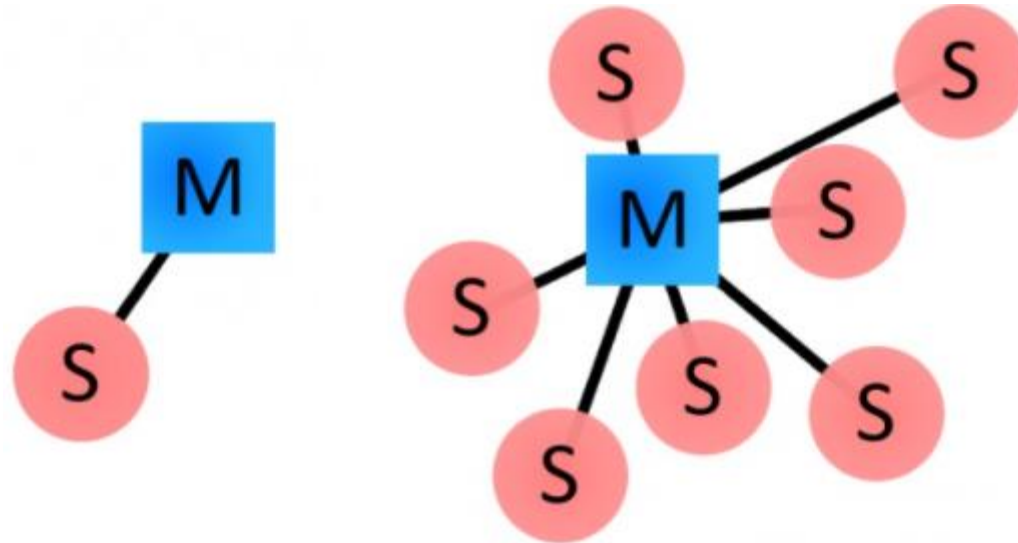
RFID

- La *RFID* (Identificación por radiofrecuencia) es un protocolo de IoT donde el uso inalámbrico de campos electromagnéticos ayuda a identificar objetos.
- Las etiquetas de lectura pueden almacenar información y no requieren energía para funcionar



Bluetooth

- Bluetooth es una tecnología de comunicaciones de corto alcance que funciona a 2,4 GHz.

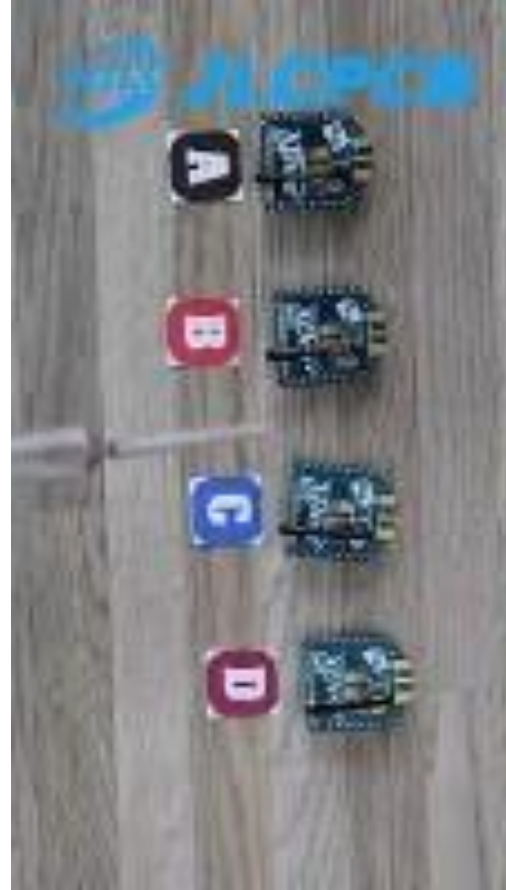


Zigbee

- Zigbee tiene similitudes a Bluetooth pero funciona creando una red de área local (LAN) de malla.
- Se basa en el protocolo IEEE802.15.4, una tecnología de redes inalámbricas a 2,4 GHz
- Funciona bien para intercambios de datos poco frecuentes, a bajas velocidades y a distancias cortas, por ejemplo, en casas o edificios.



XBEE



<https://www.youtube.com/shorts/uLXxPeRfOWw>

LoRa

- LoRa es un Protocolo Inalámbrico de Largo Alcance para dispositivos inalámbricos IoT y M2M en redes regionales, nacionales y globales.
- Las velocidades de datos pueden variar entre 0,3 kbps y 50 kbps.
- Las señales pueden atravesar obstáculos y viajar a través de largas distancias gracias a los chips LoRa.

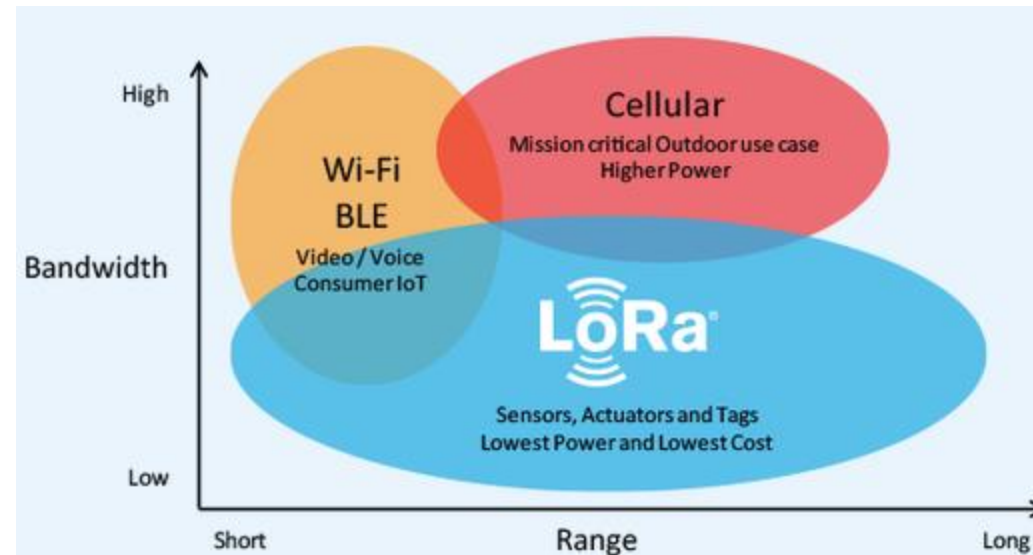


LoRa



https://www.youtube.com/watch?v=vJzBQY4vMG0&ab_channel=Tecun

Comparativas de redes inalámbricas



TRANSFERENCIA DE DATOS Y CONTROL DE ACCESOS

¿Qué es la transferencia de datos?

- La transferencia es el **volumen de datos** que fluye a través de una o varias líneas de conectividad.
- Este volumen de datos lo forman los elementos que, por ejemplo, carga la página cada vez que alguien entra en una URL:
 - Los datos que transferimos vía FTP
 - El uso del correo electrónico
 - Todas las llamadas producidas al servidor y las respuestas del mismo.

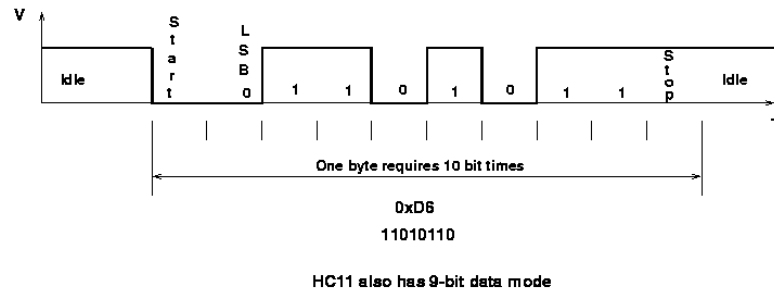


¿Qué es el ancho de banda?

- Es el **caudal máximo de datos** que se puede enviar a través de una línea.
- Supone la cantidad de datos (bits) que es posible transmitir por segundo.
- Tendremos que adecuar nuestro ancho de banda al caudal de datos que nos lleguen, y, solo en el caso de que tuviéramos demasiado caudal, plantearnos agrandar nuestro ancho de banda.



¿Qué es la comunicación serial?



- Es un protocolo muy común (no hay que confundirlo con el USB) para comunicación entre dispositivos que se incluye de manera estándar.
- El puerto serial envía y recibe bytes de información un bit a la vez.
- Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias.

Velocidades de transmisión: BTS

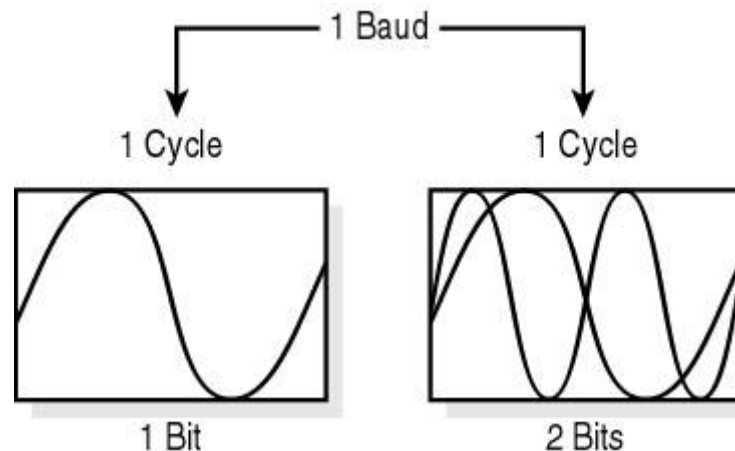
- Los bits por segundo como unidad del SI Sistema Internacional de Unidades son utilizados para expresar la velocidad de transmisión de datos o bit rate.
- Téngase en cuenta que una velocidad de transmisión expresada en bits por segundo dividida entre 8, equivale a la velocidad bytes por segundo.

Unidades de ancho de banda

Unidad de ancho de banda	Abrev.	Equivalencia
Bits por segundo	bps	1 bps = unidad fundamental de ancho de banda
Kilobits por segundo	kbps	1 kbps = 1.000 bps = 10^3 bps
Megabits por segundo	Mbps	1 Mbps = 1.000.000 bps = 10^6 bps
Gigabits por segundo	Gbps	1 Gbps = 1.000.000.000 bps = 10^9 bps

Velocidad de transmisión: Baud rate

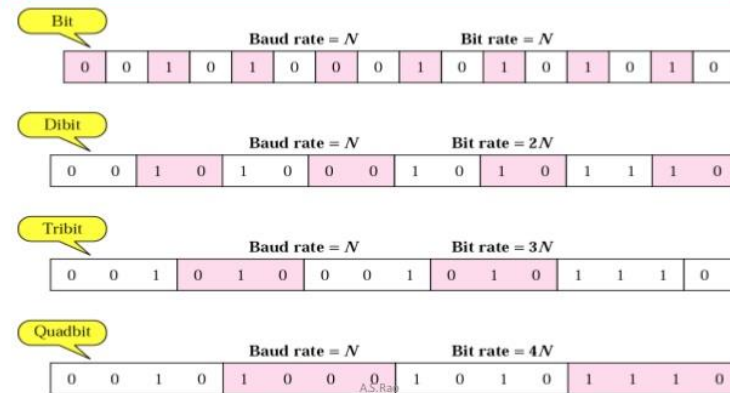
- Indica el número de unidades (símbolos) por segundo que se transfieren por el puerto serial, y se mide en baudios (*bauds*)
- Es posible tener velocidades más altas, pero se reduciría la distancia máxima posible entre los dispositivos.



Diferencias entre BPS y Baud rate

Bit Rate Vs Baud Rate

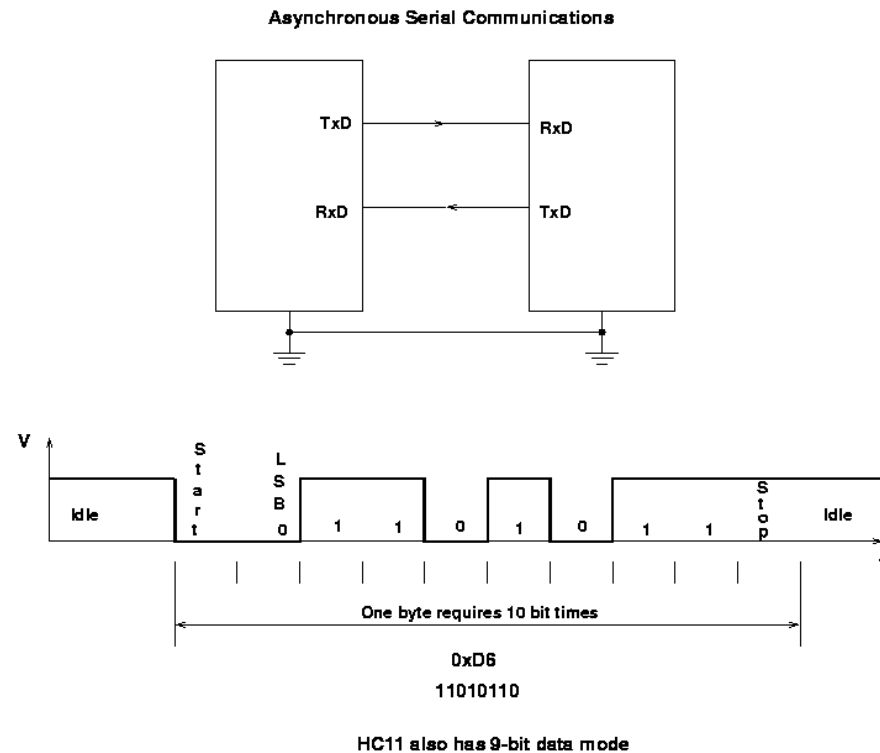
Bit rate is the number of bits per second. Baud rate is the number of signal units (symbols) per second. Baud rate is less than or equal to the bit rate.



- Existe una diferencia entre bits por segundo (bps) y baudios, debida al tipo de modulación empleada.
- Hay que diferenciar entre velocidad de señalización y velocidad de transmisión de información. Esto hace a la diferencia que existe entre baudios y bits por segundo.
- Entonces, un baudio no es sinónimo de bits por segundo ya que los signos pueden representar más de un bit. Los baudios sólo son iguales a bits por segundo cuando la señal representa un único bit.

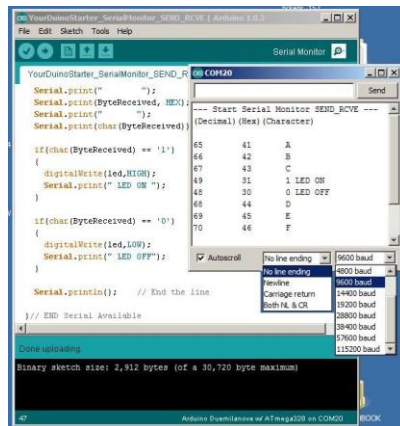
Transferencia de datos en IoT

- Se utiliza siempre la comunicación por el puerto serie Tx y Rx



Ejemplos de transmisión de datos

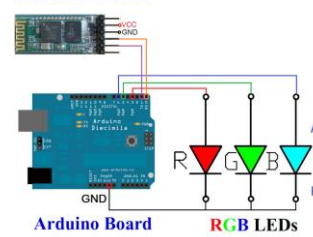
- Monitor Serial
 - Comunicación con una computadora
- Bluetooth
 - Comunicación con dispositivos externos
- Wifi
 - Comunicación e integración con la nube y la niebla



Android Device

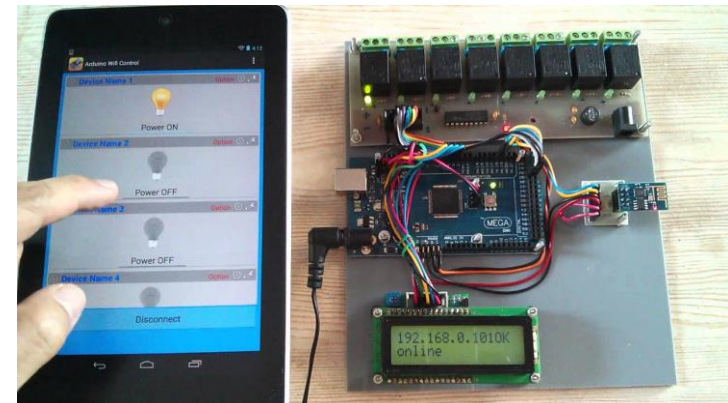


Bluetooth Module



Arduino Board

RGB LEDs

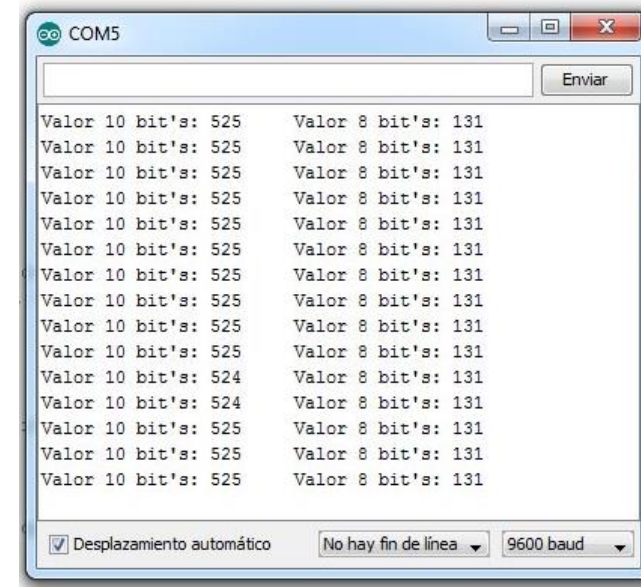


Monitor Serial

```
int LED = 10;    //Asociamos el pin
int NIVEL_TENSION = 0; //Asociamos
int VALOR = 0;   //Creamos una vari

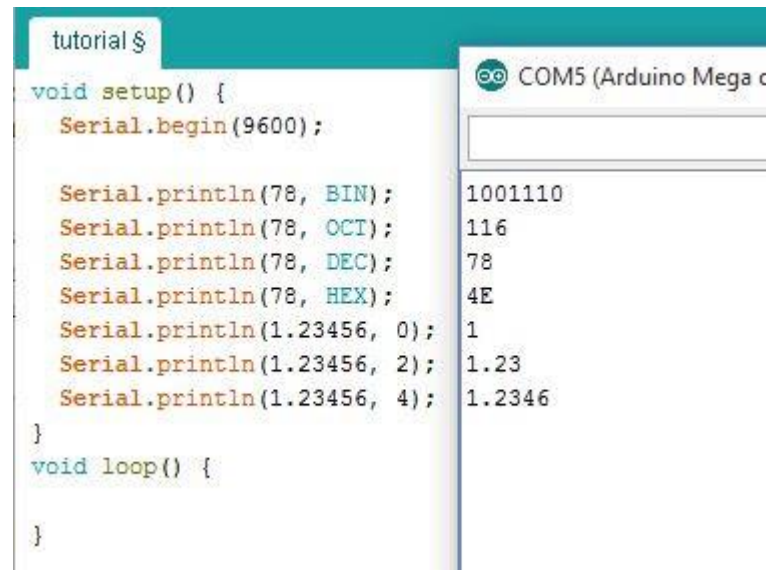
void setup(){
  Serial.begin(9600);
  pinMode(LED, OUTPUT); //Establece
  pinMode(NIVEL_TENSION, INPUT); //
}

void loop(){
  VALOR = analogRead(NIVEL_TENSION);
  Serial.print("Valor 10 bit's: ");
  Serial.print(VALOR);
  Serial.print("\t");
  VALOR = VALOR/4; //Al leer un valo
                //(una manera ser
  analogWrite (LED, VALOR);
  Serial.print("Valor 8 bit's: ");
  Serial.println(VALOR);
  delay(1000);
}
```



Monitor Serial

```
Serial.print(data, DEC);    // decimal en ASCII
Serial.print(data, HEX);    // hexadecimal en ASCII
Serial.print(data, OCT);    // octal en ASCII
Serial.print(data, BIN);    // binario en ASCII
Serial.print(data, BYTE);   // un Byte
```



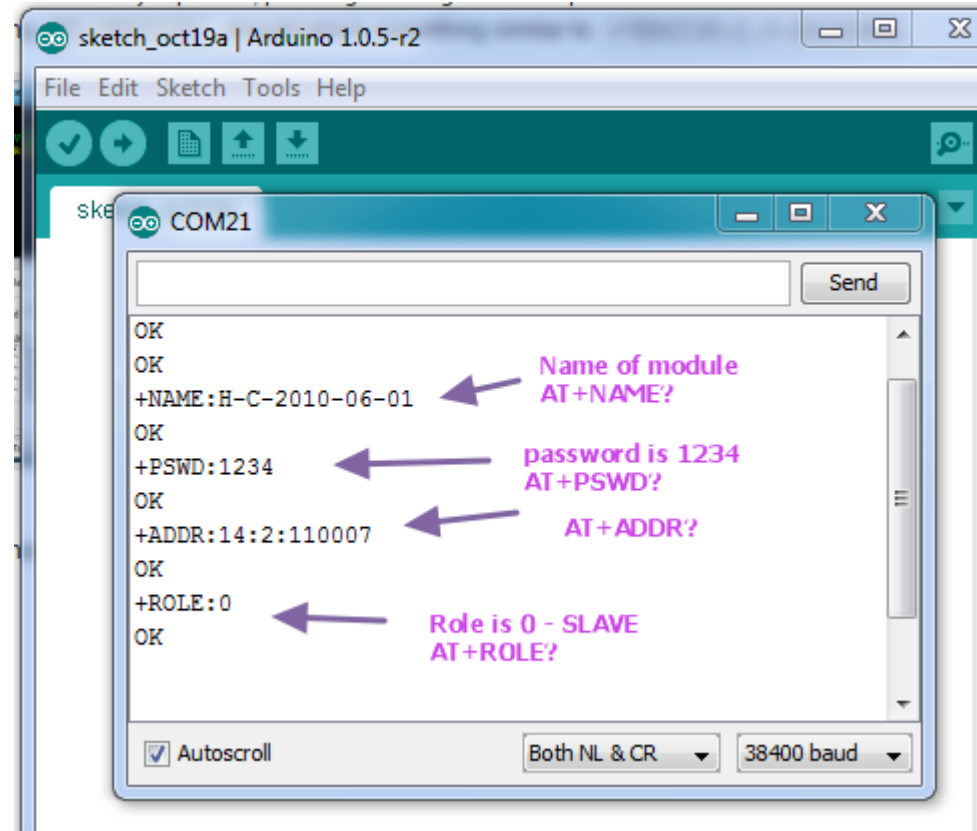
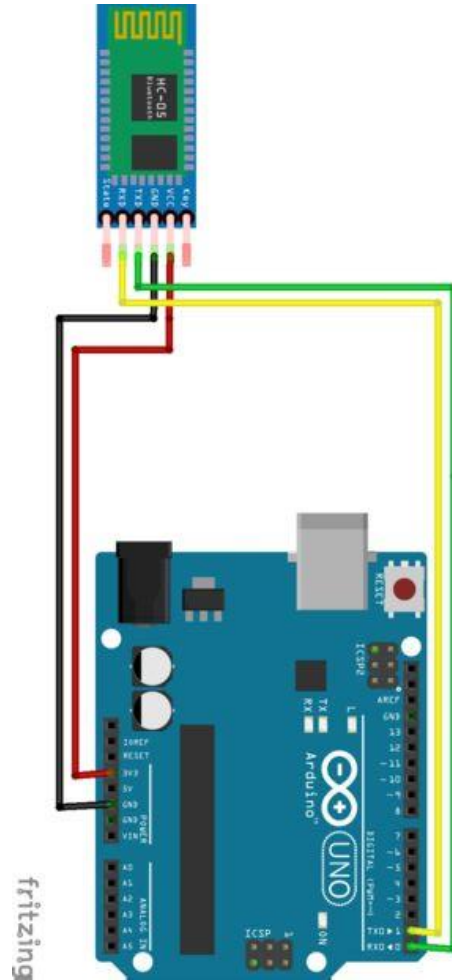
```
tutorial $
void setup() {
  Serial.begin(9600);

  Serial.println(78, BIN);
  Serial.println(78, OCT);
  Serial.println(78, DEC);
  Serial.println(78, HEX);
  Serial.println(1.23456, 0);
  Serial.println(1.23456, 2);
  Serial.println(1.23456, 4);
}
void loop() {
}
```

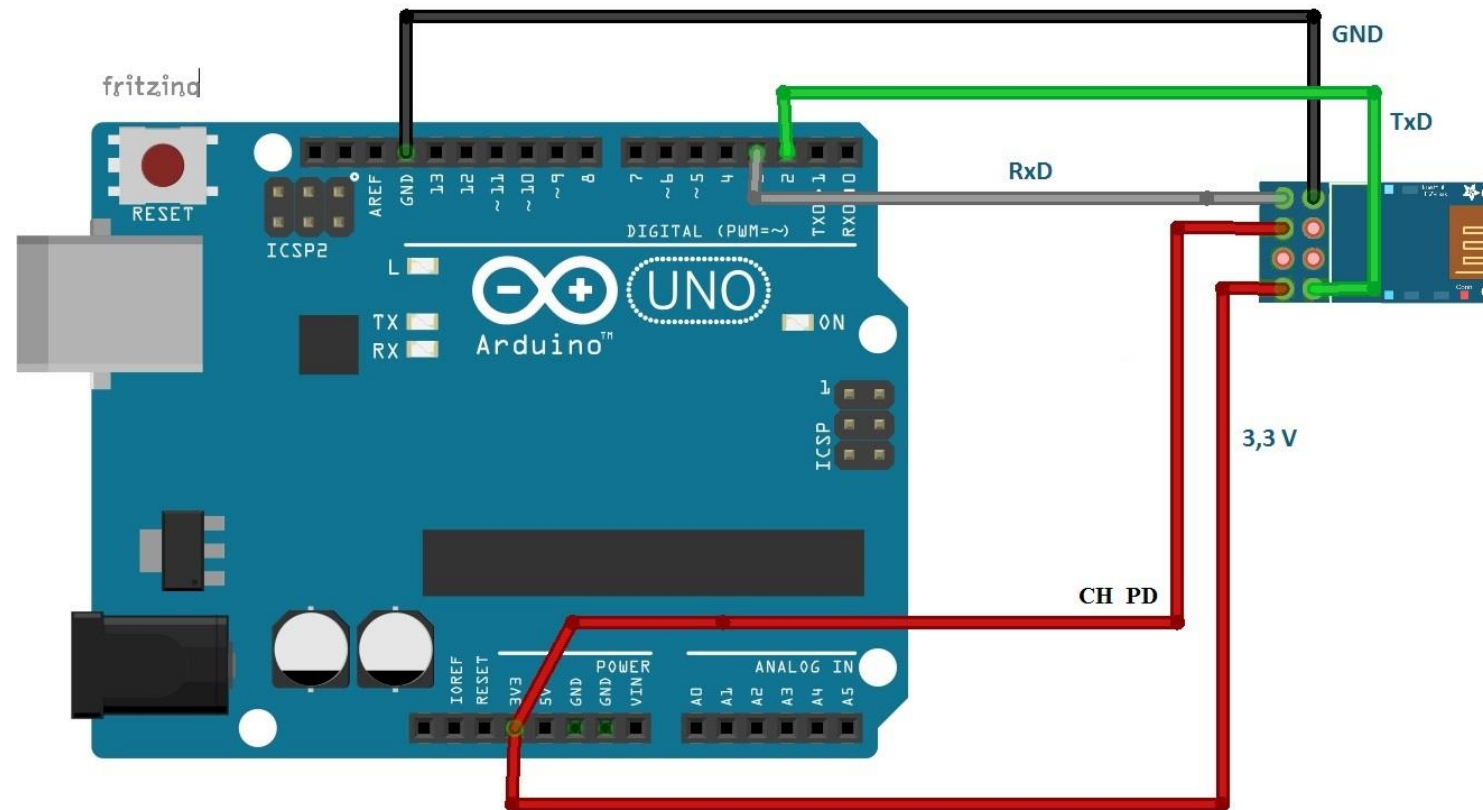
COM5 (Arduino Mega c

1001110
116
78
4E
1
1.23
1.2346

Bluetooth



WiFi



WiFi

```
COM4 (Arduino/Genuino Uno)

+CWLAP: (3, "AQ_2.4G", -88, "60:e3:27:77:d2:61", 1, 13, 0)
+CWLAP: (3, "HP-Print-B9-Officejet Pro 8600", -86, "00:9c:02:cd:56:b9", 6, -12, 0)
+CWLAP: (3, "EdisonNet", -63, "34:97:f6:1f:df:50", 6, 10, 0)
+CWLAP: (4, "ATT3J5K2f6", -76, "40:b7:f3:70:93:f0", 6, 21, 0)
+CWLAP: (3, "Vinson_wifi", -81, "08:95:2a:9d:72:a3", 6, 18, 0)
+CWLAP: (4, "ATT6G4Q583", -74, "44:e1:37:cb:8c:70", 6, 21, 0)
+CWLAP: (3, "AppleheadHQ2.4_Ext", -92, "14:91:82:dd:9a:ae", 7, -6, 0)
+CWLAP: (3, "Packing Rounds", -84, "14:ed:bb:a5:e8:0a", 9, 85, 0)
+CWLAP: (4, "ATT064", -74, "6c:ca:08:61:aa:e0", 11, 21, 0)
+CWLAP: (4, "tomels", -90, "58:6d:8f:12:cb:32", 11, 33, 0)

OK
AT+CWJAP= "YOUR_SSID", "YOUR_PASSWORD"
WIFI CONNECTED
WIFI GOT IP

OK
AT+CIFSR
+CIFSR: STAIP, "192.168.0.131"
+CIFSR: STAMAC, "5c:cf:7f:27:cd:7c"

OK
```

ESP8266 access point

AT+RST	reset
AT+CWMODE=2	configure as access point
AT+CWSAP?	get your AP details
AT+CWSAP="ssid","pwd",1,3	define your wifi AP change to unique ssid
connect to the AP with your phone	
AT+CIFSR	show ip address
AT+CIPMUX=1	allow multiple connects
AT+CIPSERVER=1,80	start server on port 80
go to browser and type IP address check reply in serial monitor	



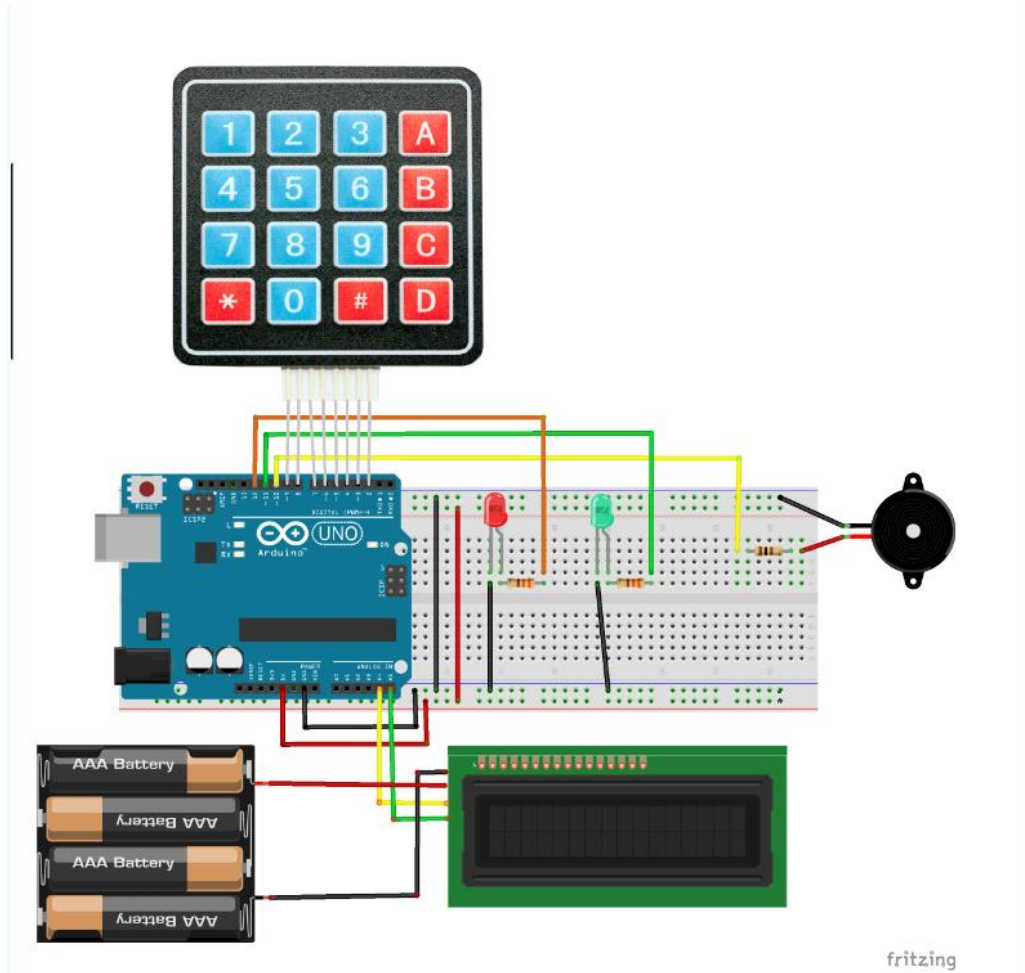
ESP8266 Arduino workshop 2015



¿Qué es un control de accesos?

- Generalmente se refiere al mecanismo que en función de la identificación ya autenticada permite acceder a datos o recursos.
- Ejemplos:
 - sistemas de controles de acceso por software cuando digitamos nuestra contraseña para abrir el correo
 - colocamos nuestra huella en un lector para encender el PC.
 - Apertura de una puerta, un torniquete o una talanquera, etc.

IoT en el control de accesos



Control de accesos

- Otra manera de crear mayor seguridad en la transmisión de datos, es crear un protocolo de comunicación
- Utilizar comandos o códigos que solo tu sistema entienda y pueda realizar acciones

Ejemplos

- Lectura de una cadena

```
//Declaracion de variables
char inChar;
String string="";

//Configuracion de puerto serial y reservacion de variable string
void setup() {
    Serial.begin(9600);
    string.reserve(200);
}

void loop(){
    if (Serial.available()){
        //Lectura de caracteres
        inChar = Serial.read();
        //Suma de caracteres en variable string
        string+=inChar;
        //Imprime la variable con los caracteres acumulados hasta la ","
        if (inChar==','){
            Serial.print("Lectura: ");
            Serial.println(string);
            //Borra la variable string para almacenar nuevos datos
            string="";
        }
    }
}
```

Ejemplo

- Comparación de una cadena para encender y apagar un led

```
int nDato;
char dato;
char Cadena[11];
char Encender[9] = "encender";
char Apagar[7] = "apagar";

void CompararCadenas() {
    if(strcmp(Cadena, Encender) == 0) { //compara la cadena, si son iguales devuelve 0
        digitalWrite(13, HIGH);
    }
    else if(strcmp(Cadena, Apagar) == 0) {
        digitalWrite(13, LOW);
    }
}

void LeerSerial() {
    nDato = 0;
    if (Serial.available())
    {
        while (Serial.available() > 0)
        {
            dato = Serial.read();
            if (dato != '\n')
            {
                Cadena[nDato] = dato;
                nDato++;
            }
            else
                break;
        }
    }
    Cadena[nDato] = '\0';
}

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

void loop() {
    LeerSerial();
    CompararCadenas();
    delay(100);
}
```