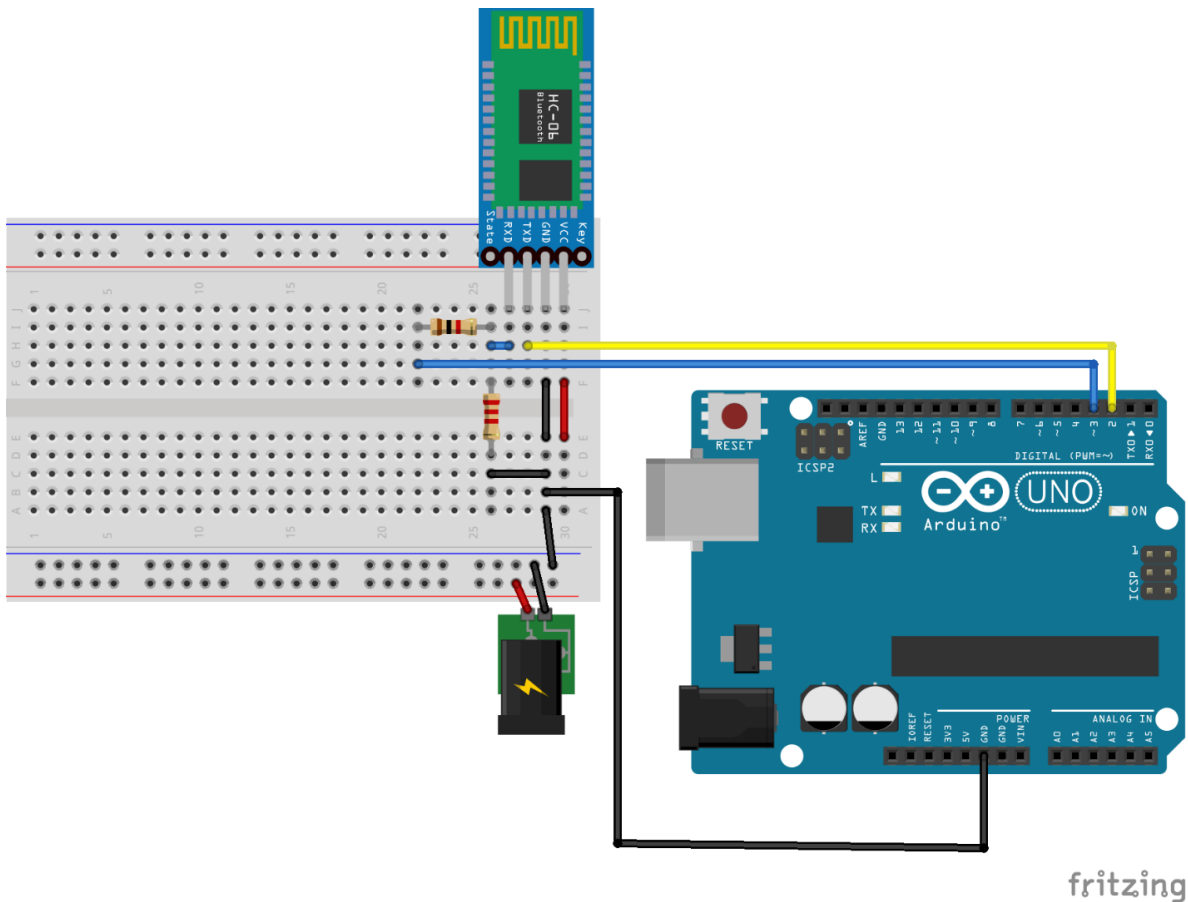


Práctica 1

Para esta práctica se necesitarán los siguientes componentes:

- 1 Arduino con cable USB
- Cables
- 1 Protoboard
- 1 Módulo Bluetooth HC-05 o HC-06
- Resistencias de 1k Ω , 2.2k Ω , 10k Ω
- Baterías o cargador de celular

1.- Conectar el circuito como se muestra en la imagen a continuación:



2.- En el IDE de Arduino, escribir el siguiente programa para probar el funcionamiento del módulo:

```
#include <SoftwareSerial.h>
SoftwareSerial bt(2, 3); // RX | TX

void setup() {
  Serial.begin(9600);
  Serial.println("Ingrese comando AT:");
  bt.begin(9600);
}

void loop(){
  //Lectura del módulo Bluetooth y envío de respuesta a consola
  if (bt.available()){
    Serial.write(BTserial.read());
  }
  //Lectura de consola y envío de respuesta al módulo
  if (Serial.available()){
    BTserial.write(Serial.read());
  }
}
```

3.- Hacer las pruebas correspondientes con comandos AT para verificar la respuesta del módulo Bluetooth.

4.- Utilizando el código anterior, modifique los circuitos y programas del bloque 3 (Ejercicios con LED) y del bloque 5 (Ejercicios con motores) para controlarlos con el módulo Bluetooth. Puede agregar componentes si es necesario.

Práctica 2

Para esta práctica se necesitarán los siguientes componentes:

- 1 Arduino con cable USB
- 1 Shield Ethernet W5100

1.- Conectar el shield al Arduino como se indica en la siguiente imagen, procurando que los puntos de soldadura del shield no hagan contacto con algún componente del Arduino:

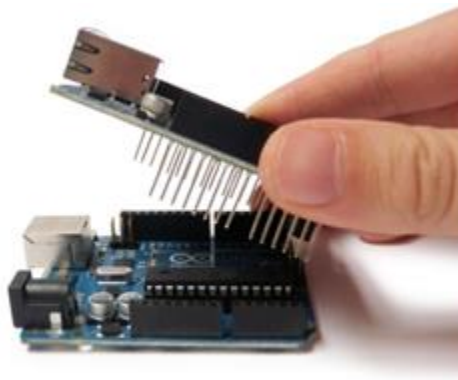


Imagen 1- Ensamble de Shield Ethernet a Arduino

2.- En el IDE de Arduino, escribir el siguiente programa para probar el funcionamiento del shield:

```
#include <SPI.h> //Importamos librería comunicación SPI
#include <Ethernet.h> //Importamos librería Ethernet

//Dirección física del adaptador, puede cambiarse si se desea.
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip( ); //Designar dirección IP de la red local
EthernetServer servidor(80); //Creamos un servidor Web con el puerto 80 que es el puerto HTTP por defecto

void setup(){
  Serial.begin(9600); //Iniciamos conexión serial a 9600 bps

  // Inicializamos la comunicación Ethernet y el servidor
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Iniciando servidor en ");
  Serial.println(Ethernet.localIP());
}

void loop(){
  EthernetClient cliente = servidor.available(); //Creamos un cliente Web

  if (cliente) { // Entra en el "if" si detecta un cliente a través de una petición HTTP
    Serial.println("nuevo cliente");
    boolean lineaEnBlanco = true; //Una petición HTTP acaba con una línea en blanco
```

```

String cadena=""; //Creamos una cadena de caracteres vacía
while (cliente.connected()) {
  if (cliente.available()) {
    char c = cliente.read();//Leemos la petición HTTP carácter por carácter
    Serial.write(c);//Visualizamos la petición HTTP por el Monitor Serial
    cadena.concat(c); /*Unimos el String 'cadena' con la petición HTTP (c). De esta manera convertimos la
petición HTTP a un String*/

    //Cuando reciba una línea en blanco, quiere decir que la petición HTTP ha acabado y el servidor Web está
listo para enviar una respuesta
    if (c == '\n' && lineaEnBlanco) {

      // Enviamos al cliente una respuesta HTTP
      cliente.println("HTTP/1.1 200 OK");
      cliente.println("Content-Type: text/html");
      cliente.println();

      //Página web en formato HTML
      cliente.println("<html>");
      cliente.println("<head>");
      cliente.println("</head>");
      cliente.println("<body>");
      cliente.println("<h1 align='center'>Servidor Web Arduino</h1>");
      cliente.println("</b><br />");
      cliente.println("</b></body>");
      cliente.println("</html>");
      break;
    }
    if (c == '\n') {
      lineaEnBlanco = true;
    }
    else if (c != '\r') {
      lineaEnBlanco = false;
    }
  }
}
//Dar tiempo al navegador para recibir los datos
delay(1);
cliente.stop();// Cierra la conexión
}
}

```

3.- Hacer las pruebas correspondientes para verificar la respuesta del servidor web montado con el shield Ethernet.

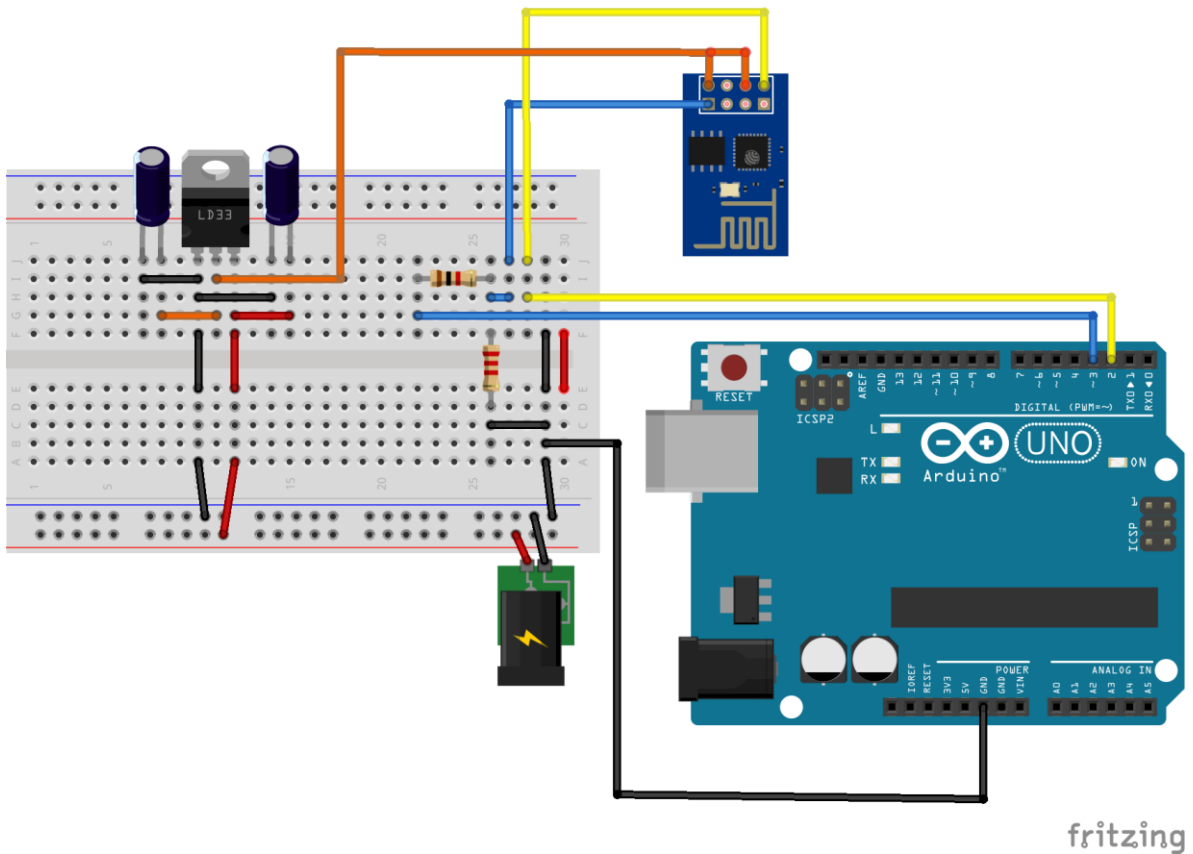
4.- Utilizando el código anterior, modifique los circuitos y programas del bloque 3 (Ejercicios con LED's) y del bloque 5 (Ejercicios con motores) para controlarlos por medio de una página web. Puede agregar componentes si es necesario.

Práctica 3

Para esta práctica se necesitarán los siguientes componentes:

- 1 Arduino con cable USB
- 1 Módulo WiFi ESP-01 (Puede utilizar el de su preferencia)
- Resistencias de $1k\Omega$, $2.2k\Omega$, $10k\Omega$
- Baterías o cargador de celular
- 1 Protoboard
- 1 Regulador de voltaje LM1117 o LD33
- 2 Capacitores de $1000\mu F$

1.- Conectar el circuito al Arduino como se indica en la siguiente imagen:



2.- En el IDE de Arduino, escribir el siguiente programa para probar el funcionamiento del módulo:

```
#include <SoftwareSerial.h>

const byte rxPin = 2; // Conectar a TX del ESP8266
const byte txPin = 3; // Conectar a RX del ESP8266

//Establecemos una conexión serial con el módulo
SoftwareSerial ESP8266 (rxPin, txPin);
```

```
void setup() {
  /*Iniciamos la consola y la conexión serial del módulo a 115200 bps. Cambie este parámetro de
  acuerdo a las especificaciones del módulo*/
  Serial.begin(115200);
  ESP8266.begin(115200);
  delay(1000); //Damos tiempo a que el módulo inicialice
}

void loop() {
  Serial.println("Enviando comando AT...");
  ESP8266.println("AT");
  delay(30);
  while (ESP8266.available()){//Mientras haya una conexión al módulo
    String respuesta = ESP8266.readStringUntil('\n');//Leemos la consola hasta encontrar nueva
    linea
    Serial.println("Respuesta de ESP8266: " + respuesta);
  }
}
```

3.- Hacer las pruebas correspondientes con comandos AT para verificar la respuesta del módulo WiFi.

4.- Utilizando el código anterior, modifique los circuitos y programas del bloque 3 (Ejercicios con LED's) y del bloque 5 (Ejercicios con motores) para controlarlos por medio de una página web. Puede agregar componentes si es necesario.

Referencias:

Arduino Ethernet Library (2018). Recuperado de: <https://www.arduino.cc/en/Reference/Ethernet>

Arduino and HC-06 (2014, Octubre 8). Recuperado de: <http://www.martyncurrey.com/arduino-and-hc-06-zs-040/>

Imagen 1 - Ensamble de Shield Ethernet a Arduino (2018). Recuperado de: <https://jp.mathworks.com/help/supportpkg/arduino/ug/arduinoethernetshield.png>

ESP8266 Arduino Core (2018). Recuperado de: <http://esp8266.github.io/Arduino/versions/2.0.0/doc/filesystem.html>