

::: ESP8266 Audio :::

Autor

Freddy Alcarazo | @surflaweb | @alcarazolabs

Objetivo:

Reproducir audio streaming de radios con el ESP8266.

Que vamos hacer?

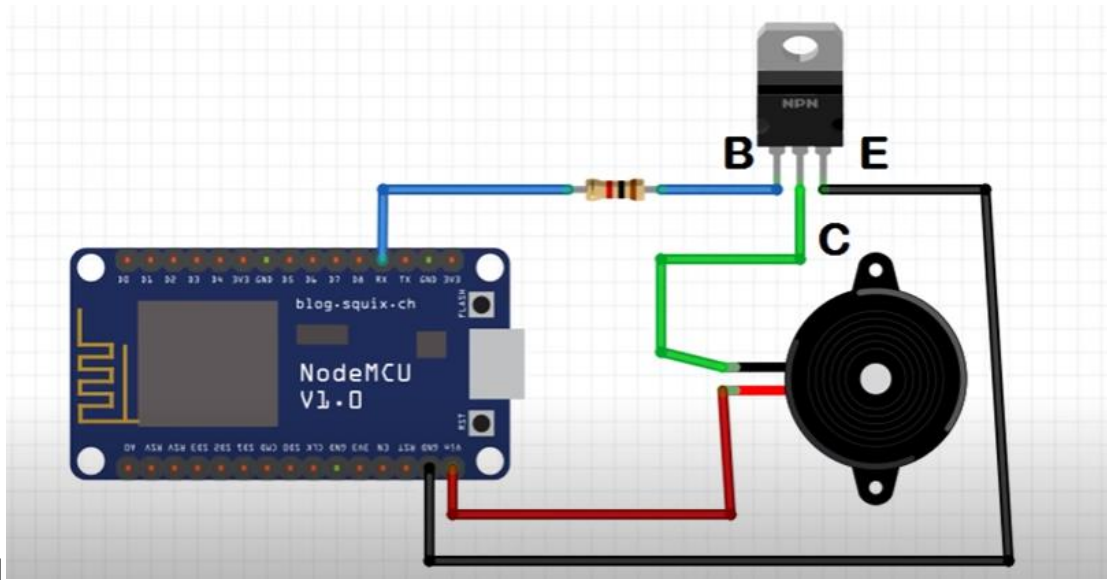
Crear una radio Wifi.

Materiales:

- X1 ESP8266
- X1 transistor TIP 120 ó TIP 122
- X1 Un parlante de 2W ó 3W (O el que tengan).
- X1 Fuente de alimentación 5v/6v (Usaré un arduino nano para aprovechar sus 5v).
- X1 Resistencia de 1k Ohms. Yo use una de 42 Ohms usando el TIP 120. Esta es opcional.
- Cables
- Protoboard

Esquemas:

El siguiente esquema usa el TIP 120 con una resistencia de 42 ohms. El problema de este esquema para mi fue que mi parlante no encendia con los 3.3v del ESP8266 usando el PIN vin, asi lo que hice fue usar una fuente de 5v.



Buscando en google encuentre el siguiente esquema:

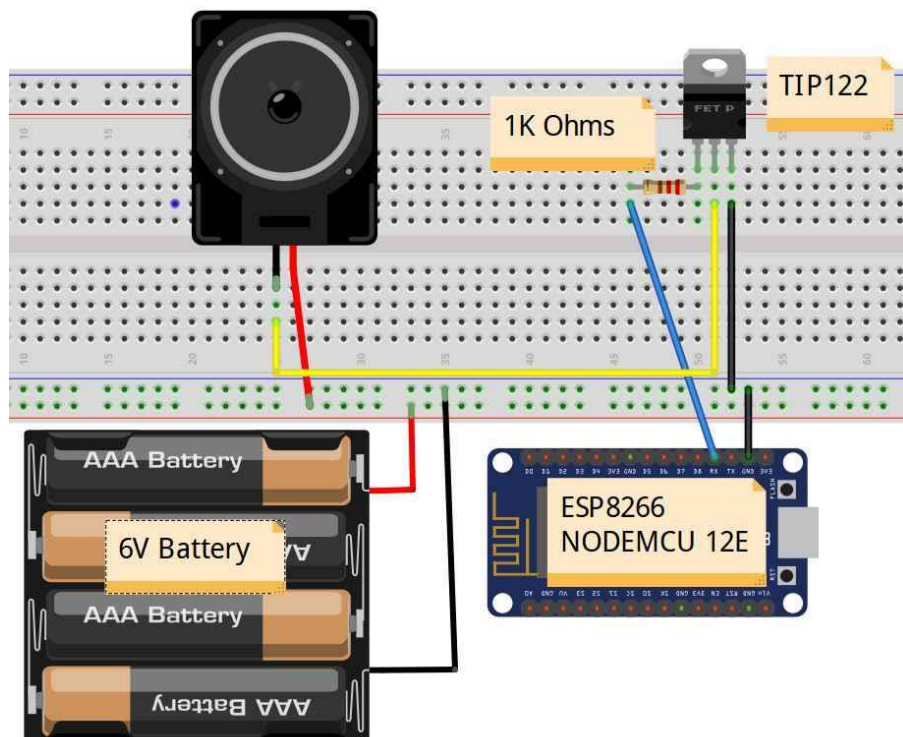
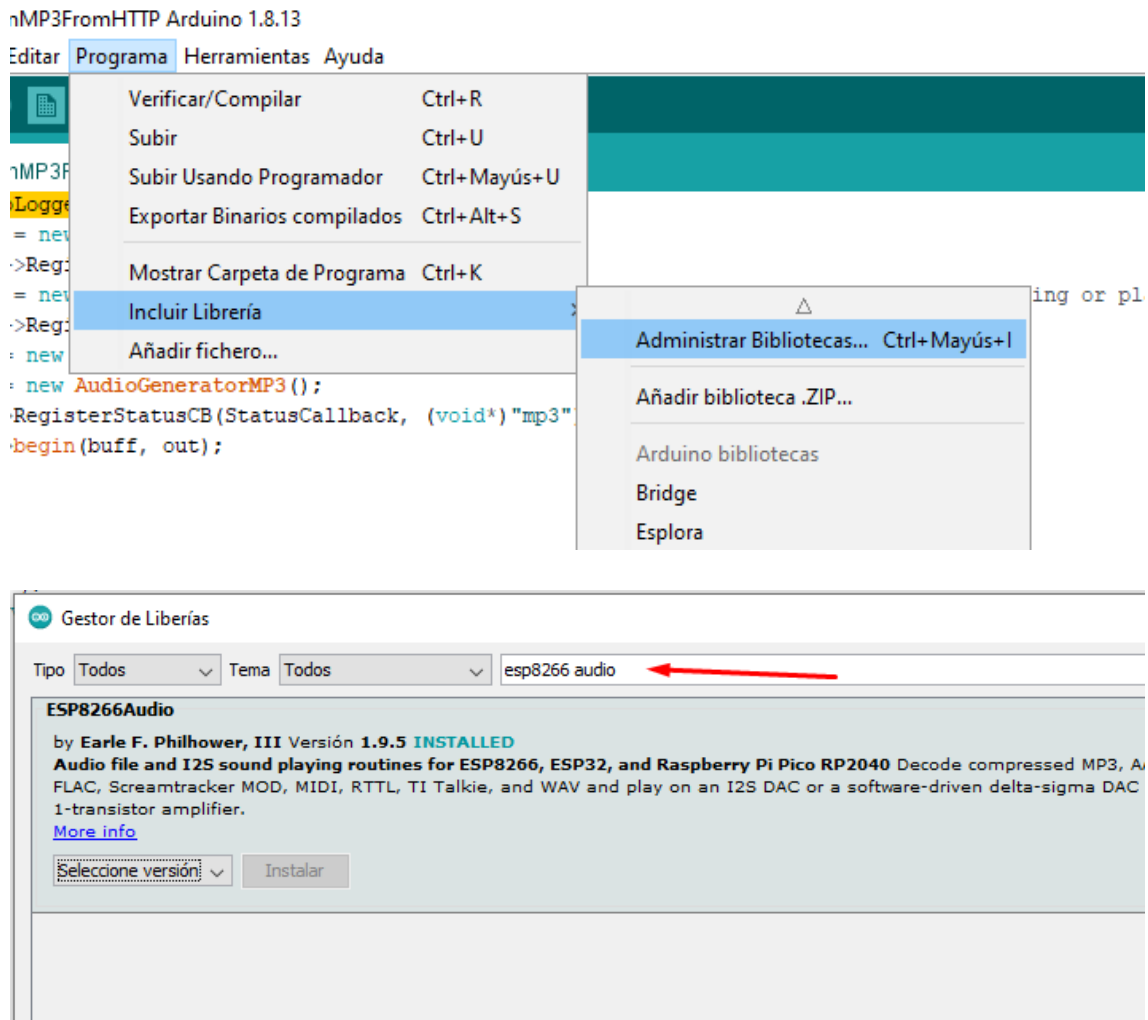


Figura 1. Fuente: <https://roboticadiy.com/esp8266-how-to-make-wi-fi-radio/>

Este usa el TIP 122 el cual es reemplazo del TIP 120 así que no hay problema, lo único bueno es que este presenta una fuente de 6v, en mi caso utilicé una de 5v como dije antes, en las pruebas pensé que el parlante no servía, pero al final fue que tuve que agregar una fuente de alimentación externa por que la que nos da, no es suficiente para mi parlante de 2W.

Programación

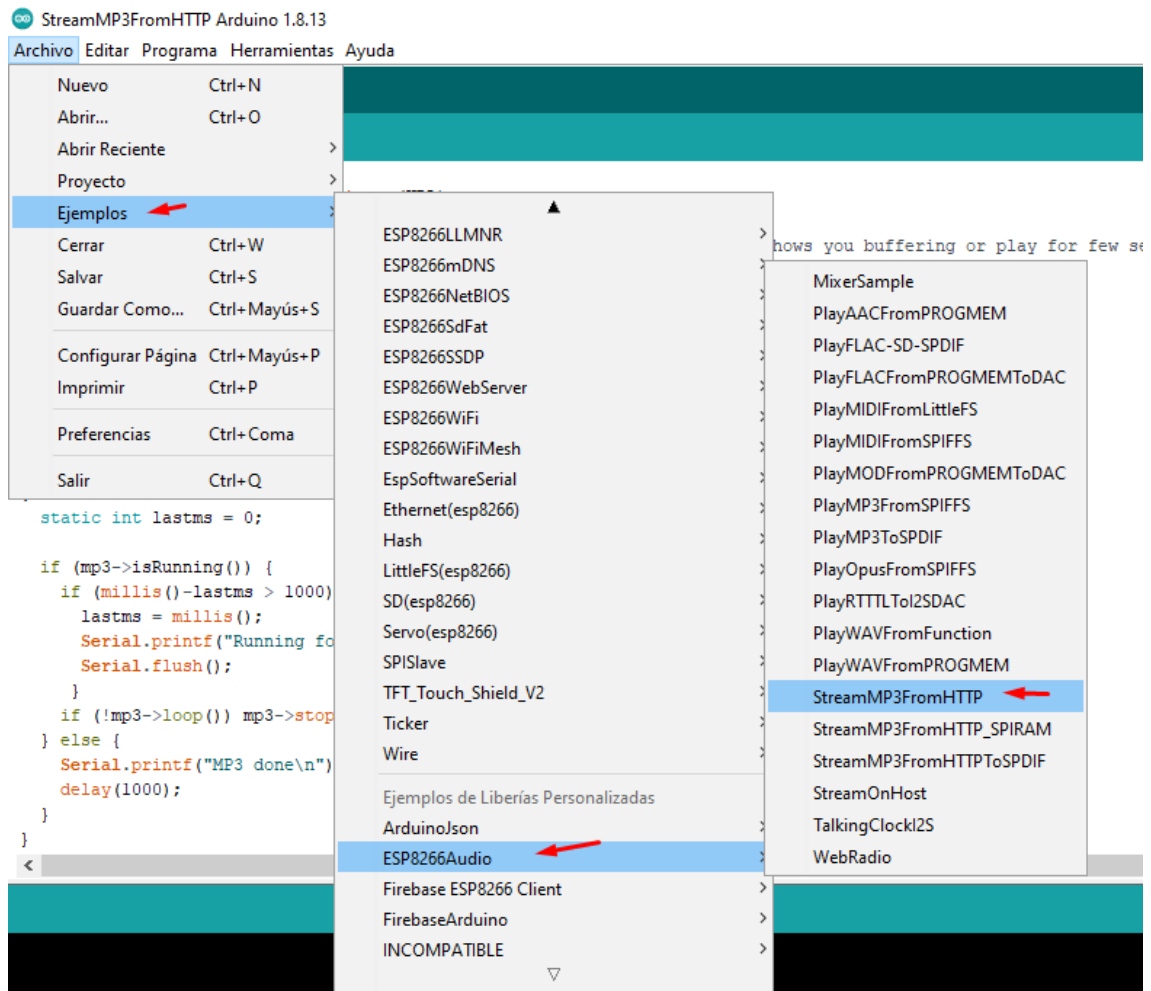
Para hacer esto vamos a usar la librería **ESP8266 audio**.



Me instale la última versión.

Ejecutar ejemplo del radio Wifi

Para probar nuestro circuito usaremos el ejemplo que trae la librería que hemos añadido a Arduino. Buscamos el ejemplo **StreamMP3FromHTTP**



Configuramos el SSID de nuestra red y la clave.

```
// To run, set your ESP8266 build to 160MHz, update the SSID info, a

// Enter your WiFi setup here:
#ifndef STASSID
#define STASSID "WLAN_WAL -1"
#define STAPSK "942918777"
#endif

// const char* ssid = STASSID;
```

Nota: El STASSID es el nombre de nuestra red wifi, si no tienen wifi crear red wifi con su teléfono si tienen datos y STAPSK es la clave de la red wifi.

Más abajo vemos lo siguiente:

```
// Randomly picked URL
const char *URL="http://kvbstreams.dyndns.org:8000/wkvi-am";
```

Esa url es donde se encuentra el streaming. Buscar la url de su emisora preferida o bien si tienes tu propia emisora debes de poner el url de su servidor.

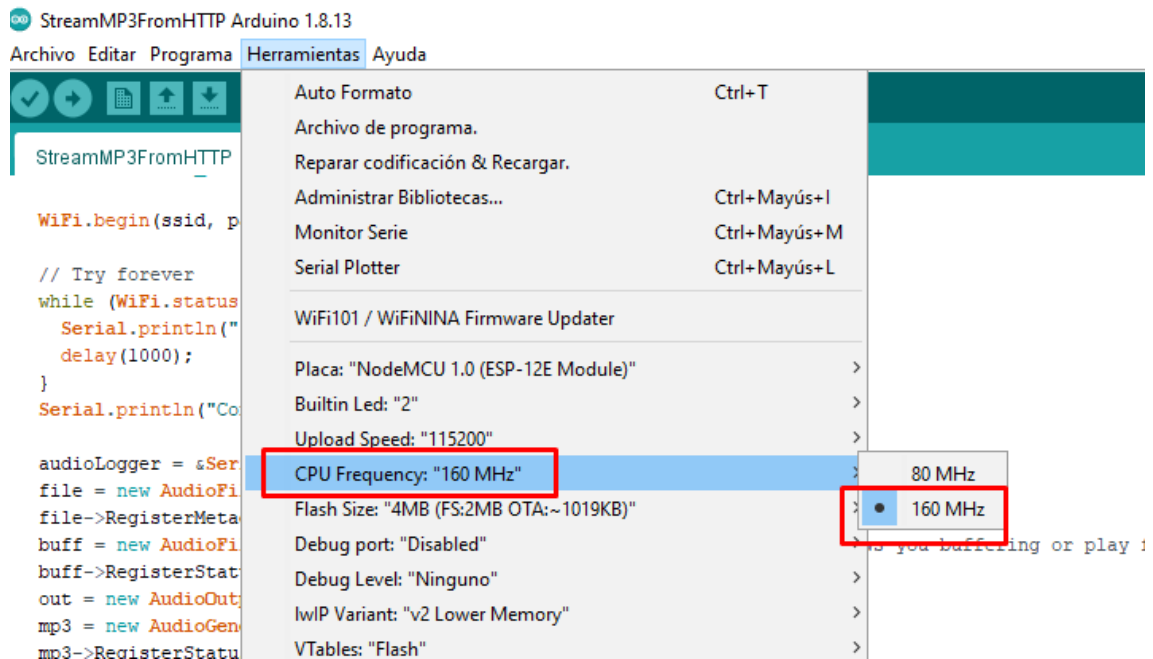
Más abajo:

Cambiamos el tamaño del buffer de 2048 a 8192:

```
-----, -----, ,  
  
audioLogger = <Serial;  
file = new AudioFileSourceICYStream(URL);  
file->RegisterMetadataCB(MDCallback, (void*)"ICY");  
buff = new AudioFileSourceBuffer(file, 8192) //if radio contentio  
buff->RegisterStatusCB(StatusCallback, (void*)"buffer");  
out = new AudioOutputI2SNoDAC();  
mp3 = new AudioGeneratorMP3();  
mp3->RegisterStatusCB(StatusCallback, (void*)"mp3");  
mp3->begin(buff, out);  
}
```

A 8192, por defecto vino en 2048, podemos usar además 4096 y 8192.

Y antes de subir el código cambiamos la frecuencia de la CPU de la placa a 160 MHZ.



Eso es todo, subir el código.

Si no puede subir el código, desconectar los cables de los pines RX y GND del ESP8266 y volver a conectar cuando el código se haya subido.



- **Nota: Si el streaming se pausa modificar el código:**
Solo volvemos a llamar a la función `setup()` debajo de la línea de código donde se imprime "Mp3 Done":

```

void loop()
{
    static int lastms = 0;

    if (mp3->isRunning()) {
        if (millis()-lastms > 1000) {
            lastms = millis();
            Serial.printf("Running for %d ms...\n", lastms);
            Serial.flush();
        }
        if (!mp3->loop()) mp3->stop();
    } else {
        Serial.printf("MP3 done\n");
        delay(1000);
        setup();
    }
}

```

Esto volvería a reconectar al streaming.

Demo: <https://youtu.be/UnkhsaCuU1k>

¿Es posible reproducir solo un mp3?

Sigue mi issue: <https://github.com/earlephilhower/ESP8266Audio/issues/514>

Fuentes:

- <https://roboticadiy.com/esp8266-how-to-make-wi-fi-radio/>
- <https://www.youtube.com/watch?v=p6BzdsJR4mE>