

APRENDIENDO CONEXIONES WIFI Y SERVIDOR

- a) **Conectar Acelerómetro al NodeMCU:** De acuerdo con la ilustración 6, conectar potencia (GND, VCC) y las dos señales del protocolo I2C (SDA, SCL) entre el NodeMCU y el MPU6050 (acelerómetro). Además, conectar un par de LEDs, por ejemplo, a los pines D2 y D3.

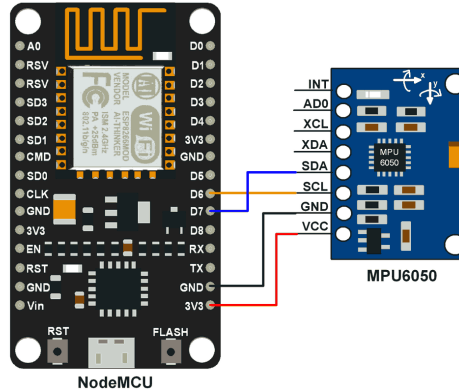


Ilustración 1. Conexiones NodeMCU y MPU6050

- b) **Boards Manager:** Ingresar al menú de Tools-> Board:-> Boards Manager. Buscar Node y verificar la librería para poder utilizar el esp8266.

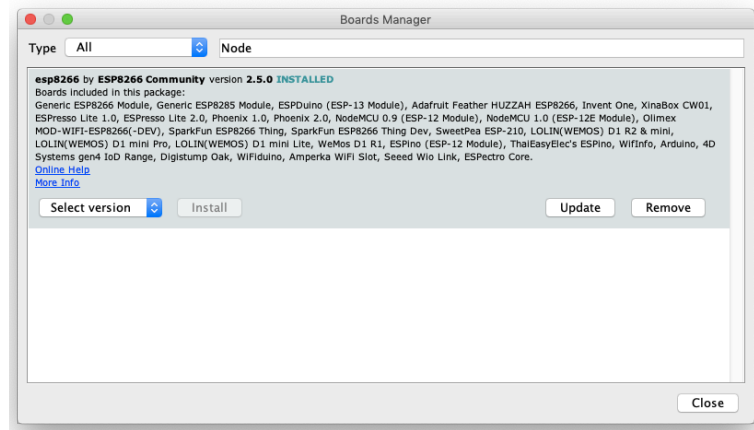


Ilustración 2. Librerías ESP8266

A continuación, ir al menú Tools-> Board: y seleccionar NodeMCU v1. Además, escoger el puerto USB activo donde se tiene conectado el microcontrolador.

- c) **Sketch:** Abrir el Arduino IDE. Abrir sketch **funcional** del laboratorio pasado. Cambiarle el nombre al archivo según su gusto, guardando así una copia.
- d) **Incluir Librerías:** Es necesario agregar las librerías para utilizar el Wifi de la placa, y el protocolo UDP.

```
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
```

- e) **Definir Variables:** Se guardan como constantes las variables del nombre y la contraseña de Wifi. Además, se manejará un identificador para el cliente y un número para cada paquete que se envíe.

```
const char* ssid = "Nombre de Wifi";
const char* password = "Contraseña Wifi";
int Id_client = 1; //Identificación del cliente
int p = 0; // Número indicador de cada paquete enviado
int contconexion = 0; // Contador intentos conexión
WiFiUDP Udp;
```

- f) **Programar setup:** Después de haber declarado todas las variables y verificar que la compilación de estas no fallé, se pasará a realizar el setup del programa. En este es necesario agregar un ciclo donde se espere a que la conexión se establezca.

```
-----TO DO: pinMode de los LEDS-----
WiFi.mode(WIFI_STA); //para que no inicie el Wifi de la placa en el modo normal
WiFi.begin(ssid, password); //Inicia la comunicación WiFi.

while (WiFi.status() != WL_CONNECTED and contconexion < 50) //Espera para intentar conectarse a la red.
{
  ++contconexion; //Aumenta uno el contador de los intentos
  -----TO DO: Encender un LED 1 por 250 milisegundos, Parapadea mientras intenta conectarse a la red. además imprimir
  en la consola "." -----
}

if (contconexion < 50) //Significa que se conecto a la red.
{
  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println(WiFi.localIP());
  Serial.println(WiFi.gatewayIP());
  -----TO DO: Dejar encendido el LED 1-----
}
else //Significa que NO se conecto a la red.
{
  Serial.println("");
  Serial.println("Error de conexión");
  -----TO DO: Dejar apagado el LED 1-----
}
```

- g) **Programar loop:** Se quiere que periódicamente se lean los datos del sensor, se armé un mensaje con todos los datos, separados por # y en el siguiente orden:

```
mensaje = 'Ax' + "#"
          + 'Ay' + "#"
          + 'Az' + "#"
          + 'Gx' + "#"
          + 'Gy' + "#"
          + 'Gz' + "#"
          + '1' + "#"
          + 'paquete'
```

Para ello, es necesario:

```
Udp.beginPacket("192.168.0.100", 9001); // Inicia conexión, Esta ip es la ip del computador servidor y el puerto debe
coincidir
-----TO DO: Encender el LED 2, para mostrar que se inició transmisión-----
```

```
msg = String(Ax) + "#" + String(Ay) + "#" + String(Az) + "#" + String(Gx) + "#" + String(Gy) + "#" + String(Gz) + "#" +  
String(Id_client) + "#" + String(p); //El mensaje completo contiene el id del cliente y el numero de paquete enviado
```

```
for (int i = 0; i < msg.length(); i++)//Escribir el mensaje byte a byte.  
{  
  int old = micros();  
  Udp.write(msg[i]);  
  while (micros() - old < 87);  
}  
Udp.endPacket();// Cierra conexión  
-----TO DO: Apagar el LED 2, para mostrar que se terminó transmisión-----  
-----TO DO: Sumarle uno al identificador de paquetes -----  
-----TO DO: Esperar tiempo-----
```

- h) **Cargar programa:** Una vez sea finalizada la programación, se carga el programa al NodeMCU. **Verificar si se esta conectando a la red, si esta funcionando la lógica de los LEDs y el envio de datos.**
- i) **Servidor:** Después de cargado el programa, y que se este imprimiendo bien la información en la consola Serial. **CORRER EL SERVIDOR EN EL PC (Tutorial *en Sicua*).** Confirmar si en el servidor están llegando los datos de su grupo. Tenga en cuenta que es importante enviar los mensajes completos que tengan la estructura mencionada.

```
mensaje =  
    'Ax' + "#"  
    + 'Ay' + "#"  
    + 'Az' + "#"  
    + 'Gx' + "#"  
    + 'Gy' + "#"  
    + 'Gz' + "#"  
    + 'grupo' + "#"  
    +'paquete'
```