

4.2. Dispositivo Físico

En este apartado se analizan cada uno de los componentes utilizados en el desarrollo del proyecto además del montaje y programación del dispositivo.

4.2.1. Chip RC522

El chip RC522RFID/NFC, es un dispositivo que se puede utilizar para leer o escribir en tags, comunicarse con teléfonos que te permitan la tecnología NFC e incluso puede llegar a actuar como un tag RFID/NFC.

En el proyecto se utiliza el chip RC522 conectado a un NodeMCU, pero este se puede utilizar en otras placas como pueden ser Arduino o Raspberry Pi.

La principal diferencia entre RFID y NFC, es que el NFC añade funciones de seguridad, especialmente las limitaciones del rango de lectura-escritura.

Este módulo trabaja a una frecuencia de 13,56MHz, frecuencia que utiliza la tecnología RFID.

El módulo o chip se alimenta mediante un voltaje de 3,3V.

El chip RC522 solo permite realizar la comunicación mediante SPI:

- **Conexión mediante SPI**

Es un protocolo de comunicación síncrono de 4 hilos que permite alcanzar velocidades muy altas de comunicación.

Es un protocolo que es síncrono porque utiliza líneas diferentes para los datos y el reloj(CLOCK).

El hardware de recepción de este protocolo puede ser un simple Shift register, lo cual lo hace una solución más barata que otros protocolos de comunicación.

Este tipo de comunicación o bus se utiliza principalmente para la transferencia de información o datos entre circuitos integrados en equipos electrónicos.

El bus SPI se puede definir por 4 pines:

- **SCLK o SCK:** Es la señal de reloj que posee el bus. Impone la velocidad a la que va a ser transmitido cada bit.
- **MISO :** Señal de entrada a nuestro dispositivo por donde somos capaces de recibir los datos desde el otro integrado.
- **MOSI:** Se encarga de la transmisión de datos hacia el otro integrado.

- **SS o CS:** Habilita el integrado hacia el que se envían los datos , esta señal puede o no usarse.

En la siguiente imagen se puede ver una representación del Bus SPI con los pines descritos:

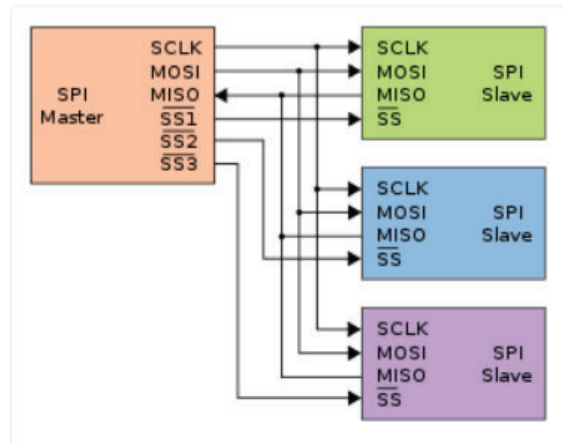


Ilustración 29 Bus SPI

4.2.2. NodeMCU ESP8266

Se trata de un SoC que tiene todo integrado para que funcione de forma independiente. Posee un regulador de tensión integrado además de un puerto USB de programación que permite cargar desde el IDE de Arduino los programas directamente al NodeMCU, sin necesidad de utilizar una placa Arduino[33].

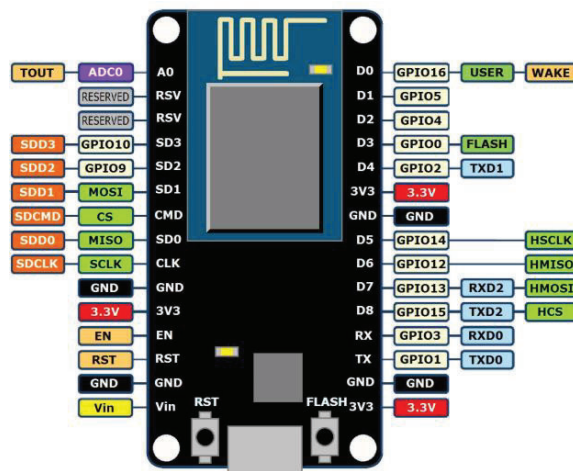


Ilustración 30 NodeMCU ESP8266 datasheet

Con respecto a las características del ESP8266 destacan las siguientes:

- Tiene integrada una MCU de 32bits y bajo consumo llamada Tensilica L106[33].
- Una memoria RAM que ronda una capacidad de unos 50kB[33].
- Una entrada analógica o ADC de 10 bits[33].
- 17 pines para entrada-salida denominados GPIO[33].
- Posee un módulo Wifi para establecer conexión a internet de 2.4 GHz[33].

4.2.3. Diseño módulos del proyecto y montaje del circuito

El proyecto se divide en tres pilares fundamentales. El primero de ellos es el programa realizado con la IDE de Arduino, encargado de recibir el nombre del usuario emisor mediante el chip NFC y obtener la información de dicho usuario de la base de datos mediante el NodeMCU con la ayuda de dos archivos php. El segundo pilar fundamental es la base de datos, donde se almacena la información de todos los usuarios, tanto el nombre de usuario y contraseña para acceder a su perfil web como la información que desea compartir con el resto de los usuarios. Por último, nos encontramos con la página web realizada con NetBeans utilizando HTML5 como lenguaje, donde el usuario podrá acceder con sus credenciales y una vez dentro de su perfil podrá modificar la información que desea compartir y consultar la información que ha recibido de otros usuarios. A continuación, se detallan cada uno de estos elementos que componen el proyecto desarrollado:

En la siguiente imagen se puede observar el circuito usado, donde el chip NFC se encuentra conectado al NodeMCU que es el encargado de proporcionarle alimentación y subir a la base de datos la información recogida por el chip NFC.

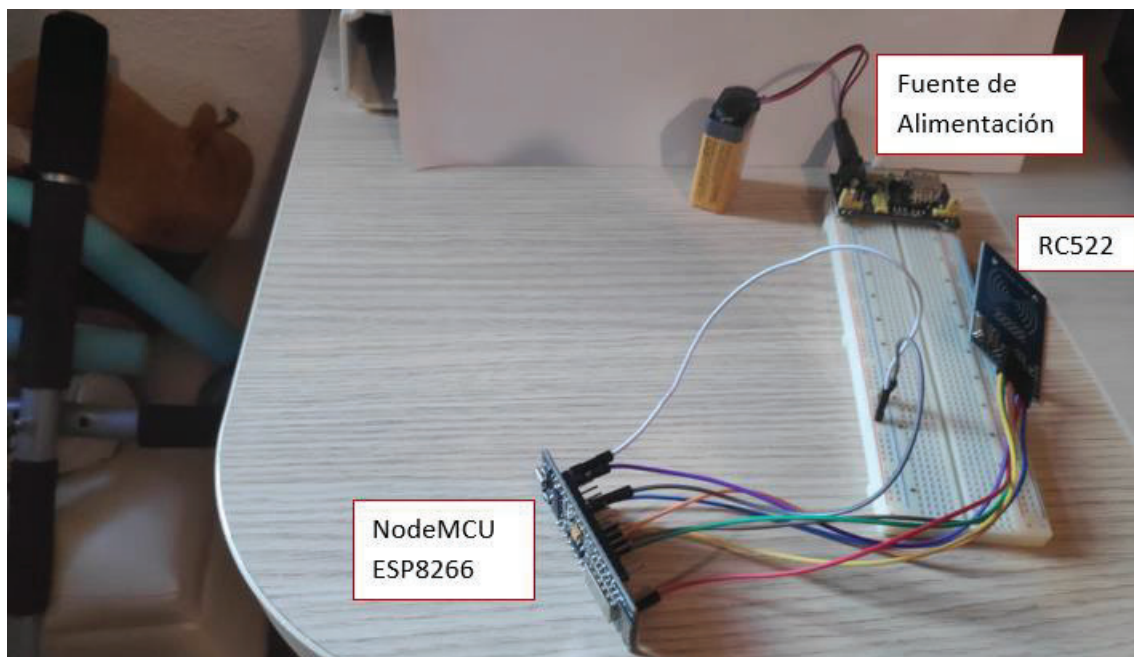


Ilustración 31 Circuito Arduino

Entre los elementos que se encuentran en el prototipo se encuentran:

- **Protoboard:** Placa con orificios que se encuentran conectados entre sí eléctricamente, siguiendo patrones de líneas, en la cual se han conectado cables para el correcto funcionamiento y alimentación de los componentes.
- **Alimentación:** La alimentación se proporciona por medio de una fuente que es capaz de proporcionar un voltaje de 3.3 o 5V conectado a una pila de 9V que es lo que le suministra dicho voltaje.



Ilustración 32 Fuente de alimentación

- **Chip NFC RC522:** El chip NFC utilizado en este proyecto, es el chipRC522, pero existen otra serie de chips NFC compatibles. Este chip establece la conexión con los dispositivos por medio de SPI.

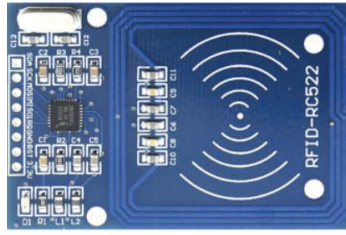


Ilustración 33 chip NFC RC522

- **NodeMCU ESP8266:** Componente utilizado para mantener una conexión con internet y la base de datos, encargado de actualizar y obtener la información de la base de datos.

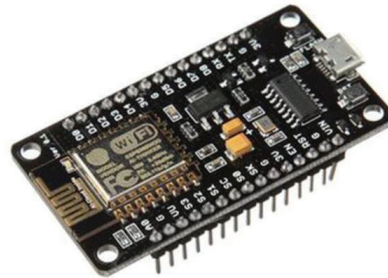


Ilustración 34 NodeMCU ESP8266

4.2.4. Diseño módulos

Todo usuario tiene una tarjeta o etiqueta RFID que irá acoplada al dispositivo con el que desea compartir su información. Esta tarjeta tendrá grabado el nombre de usuario del propietario, para que posteriormente cuando el chip NFC encuentre dicha tarjeta reciba el nombre de usuario y el NodeMCU con el nombre de usuario busque la información de este en la base de datos.

El proyecto tiene un módulo o programa principal que es el encargado de realizar todo el proceso desde que el chip NFC recibe el nombre de usuario hasta que el NodeMCU busca la información en la base de datos de dicho usuario y actualiza la base de datos creando una nueva fila con el nombre del usuario emisor, del usuario receptor y la información que se ha transmitido.

4.2.5. Módulo de recibir información

El módulo encargado de recibir información pone en funcionamiento a los dos componentes que forman el circuito: el chip NFC y el NodeMCU.

El programa comienza mandando la orden al NodeMCU de establecer conexión a internet proporcionándole el nombre del router o SSID y la contraseña, se realiza en el `setup()` del programa Arduino, ya que solo interesa que se conecte a la red Wifi al comienzo del programa y

no cada vez que se envíe información. El fragmento de código para establecer conexión a internet es tal y como se muestra a continuación:

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) { //Cuenta hasta 50 y
//si no se conecta cancela la operación
    delay(500);
    Serial.print(".");
}
```

Como se puede ver en la imagen anterior entra en un bucle while hasta que se conecte, pero si la conexión se demora más tiempo del que tarda el programa en hacer una cuenta hasta 50, se cancela la conexión.

Llegados a este punto, si la conexión ha prosperado, el programa pone en funcionamiento al chip NFC en modo de búsqueda de dispositivos compatibles. Para que el chip NFC sea capaz de buscar dispositivos y ver cuál es compatible y cual no y recibir la información en caso de que lo sea, se ha creado el siguiente método llamado leer, que se ejecutará dentro del método loop() del programa, ya que se quiere que el chip NFC esté constantemente en búsqueda de dispositivos:

```
bool leer(char mensaje[LONGITUD_BYTES])
{

    if (!rfid.PICC_IsNewCardPresent())
    {
        return false;
    }
    if (!rfid.PICC_ReadCardSerial())
    {
        Serial.println("Error leyendo serial");
        return false;
    }
    byte bloque = 1; // El bloque que leemos
    byte longitud = LONGITUD_BYTES;
    byte buferLectura[LONGITUD_BYTES];

    MFRC522::StatusCode estado;
    estado =
    rfid.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
    bloque, &key, &(rfid.uid));
    if (estado != MFRC522::STATUS_OK)
    {
        Serial.println("Error autenticando");
        Serial.println(rfid.GetStatusCodeName(estado));
        return false;
    }
    estado = rfid.MIFARE_Read(bloque, buferLectura,
    &longitud);

    if (estado != MFRC522::STATUS_OK)
    {
```

```

        Serial.println("Error leyendo bloque");
        Serial.println(rfid.GetStatusCodeName(estado));
        return false;
    }

    for (uint8_t i = 0; i < LONGITUD_BYTES - 2; i++)
    {
        mensaje[i] = buferLectura[i];
    }

    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
    return true;
}

```

El método retorna un booleano, devuelve true en caso de que se haya podido leer el contenido del usuario emisor y false en caso contrario. Los dos primeros bucles if se utilizan para saber si no hay tarjeta presente o si no se ha podido leer el contenido en caso de que si encuentre un dispositivo, en el caso de que entre en cualquiera de los dos bucles, el método devuelve false, en caso contrario seguirá para leer el contenido del dispositivo encontrado. En los siguientes dos bucles if, se mirará el estado del dispositivo encontrado y en el caso de que no de OK en ninguno de los dos estados disponibles, querrá decir que ha sucedido un error al intentar leer el contenido del mismo, posiblemente por ser un dispositivo no soportado por el chip NFC utilizado. En caso de que no entre en ninguno de los dos últimos bucles if comentados, pasará a un bucle for donde se recorrerá el contenido del mensaje del dispositivo captado por el chip NFC obteniendo el contenido del mismo, llegados a este punto el método devolverá true ya que ha sido capaz de leer el contenido del dispositivo encontrado.

El método loop() del programa es la parte principal, ya que el chip NFC estará constantemente en búsqueda de dispositivos y el NodeMCU estará igualmente activo, para que cuando el chip NFC reciba el nombre de usuario de un dispositivo, pueda buscar en la base de datos la información de este. Como se puede ver en el fragmento de código anterior, el bucle comienza ejecutando el método de leer, en caso de que devuelva true entrará en el primer bucle if, ya que esto supondría que el chip NFC ha encontrado un dispositivo compatible, en caso contrario el chip NFC seguirá buscando dispositivos hasta que encuentre uno compatible. Una vez encuentre un dispositivo compatible y entre en el primer bucle if, el NodeMCU establece conexión con internet y con la base de datos y ejecuta los archivos php para obtener la información del dispositivo encontrado, utilizando el nombre de usuario recibido por el chip NFC y buscando coincidencias dentro de la base de datos.

Todos los datos obtenidos son enviados a un servidor donde se almacenarán para poder ser consultados posteriormente. Para realizar ese envío de información el NodeMCU utilizará el protocolo HTTP para conectar con un archivo PHP, el cual será el encargado de realizar la inserción de la información en la base de datos. A continuación se muestra el código desarrollado en el IDE de Arduino donde el NodeMCU ejecutará un archivo php cargado en el servidor para crear nuevas etiquetas en la base de datos.

```

String url = "http://127.0.0.1/prueba/envia_datos2.php";
String datoss = (String) contenidoRfid;
String data =
"emisor="+datoss+"&receptor="+receptorr+"&correo=Rodolfo@gmail.
com";

Serial.println("Requesting URL: ");
Serial.println(url);

client.print(String("POST ") + url + " HTTP/1.0\r\n" +
    "Host: " + host + "\r\n" +
    "Accept: *" + "/" + "*\r\n" +
    "Content-Length: " + data.length() + "\r\n" +
    "Content-Type: application/x-www-form-urlencoded" +
    "\r\n" + "\r\n" + data);

delay(10);

Serial.println("Respond: ");

while(client.available()){

    String line = client.readStringUntil('\r');
    Serial.println(line);
}

```

4.3.Servidor

En este apartado se encuentra la información sobre el Servidor Apache utilizado. Como se refleja en la arquitectura del proyecto, dentro del servidor se puede encontrar la base de datos, donde se almacena la información de todos los usuarios y la página web creada, donde cada usuario puede acceder a su perfil mediante unas credenciales únicas para consultar la información que ha recibido y modificar su propia información.

4.4.Base de datos

Este apartado recoge toda la información acerca de la base de datos creada con MySQL, incluyendo un diagrama de entidad relación, que muestra la relación entre las distintas tablas de la base de datos.