CLIENTE DNS DINÁMICO

ALONSO, ENRIQUE

INTRODUCCIÓN A LA PROGRAMACIÓN DE MICROCONTROLADORES

PROFESOR: JOSÉ LUIS DI BIASE

DESCRIPCIÓN:

El presente proyecto muestra una implementación de un cliente DNS dinámico. Un cliente DNS dinámico permite la actualización automática de la dirección IP de un host en un servidor DNS y permite acceder a determinados nodos de red que tienen dirección ip dinámica (como las conexiónes hogareñas de internet.) apuntando a un hostname conocido de antemano.

Esta implementación utiliza, aunque modificada, una librería llamada EASYDDNS.

La estructura sobre la que fue montado este proyecto consta de 2 partes esenciales.

El modulo ESP8266 que realiza una conexión a internet mediante Wifi para luego enviar las actualizaciones al servidor DDNS designado. La llamaremos Parte A.

El modulo de control, utilizando Arduino UNO, mediante el cual por medio de una pantalla LCD y una botonera nos permitirá configurar todo lo referido a internet y al servicio DDNS. La llamaremos parte B.

Los materiales utilizados fueron:

1 x Arduino Uno

1 x ESP8266

1 x LCD display 16x2

1x Fuente 3.3v/5v (opcional)

5x Botones

Cables M-M/H-M

1x Protoboard

1x Led

Resistencias 10Kohm

Resistencias 220ohm

Resistencias 2Kohm

Resistencias 1Kohm

PARTE A

En esta parte nos encargaremos de la programación del modulo ESP8266.

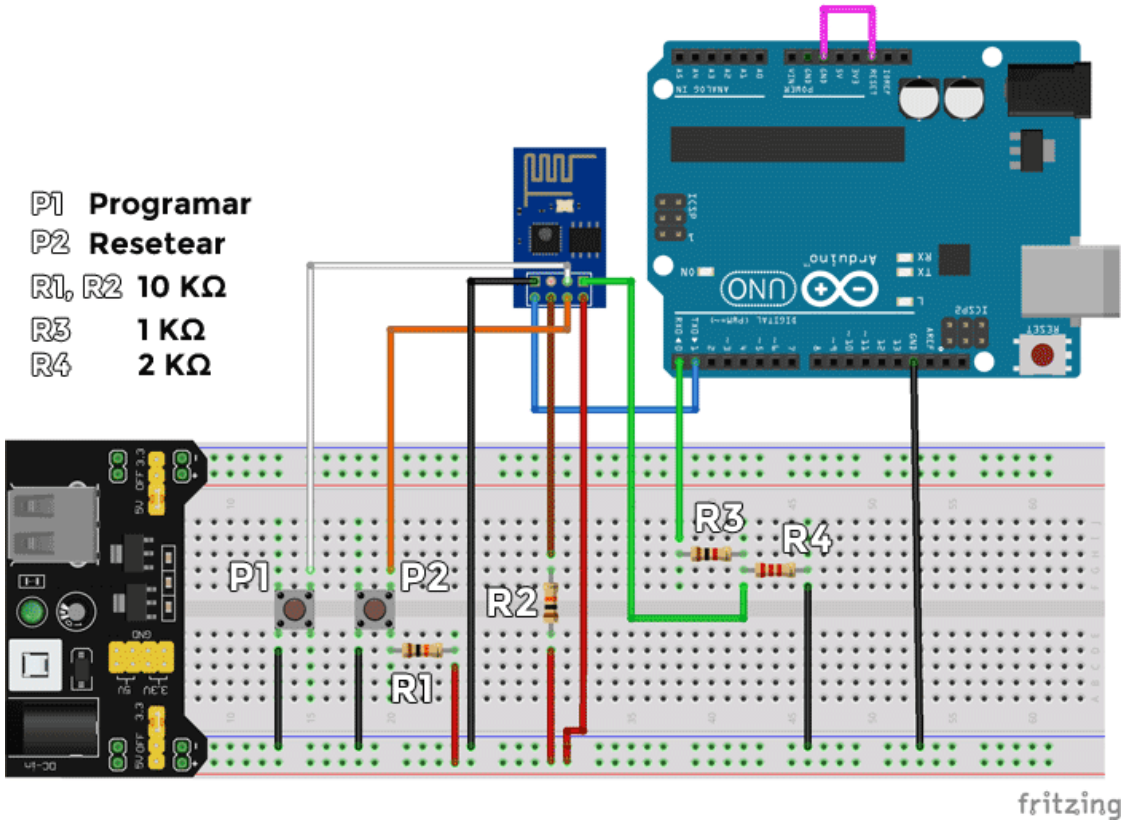
Para poder programar este modulo hay varias metodologías que se pueden revisar en las referencias de este documento. Se eligió la metodología utilizando Arduino UNO como puente.

Para lograr que nuestro código fuente se escriba en la memoria del ESP8266 en lugar del arruino UNO debemos verificar en el IDE de Arduino los siguientes parámetros y que las conexiones respeten el siguiente diagrama.

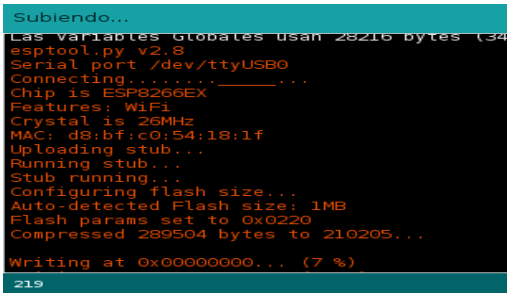
Parámetros IDE Arduino:

- Flash Mode → «DIO»
- Flash Frequency → «40MHz»
- CPU Frequency → «80MHz»
- Flash Size → «512K (64K SPIFFS)»
- Debug port → «Disabled»
- Debug Level → «Ninguno»
- Reset Method → «ck»
- Upload Speed → «115200»

Esquema de conexión



Una vez enviado el upload desde Arduino IDE al ESP8266 se debe pulsar el botón P1 para permitir la programación, de lo contrario el código no se almacenará. La pantalla lucirá así:



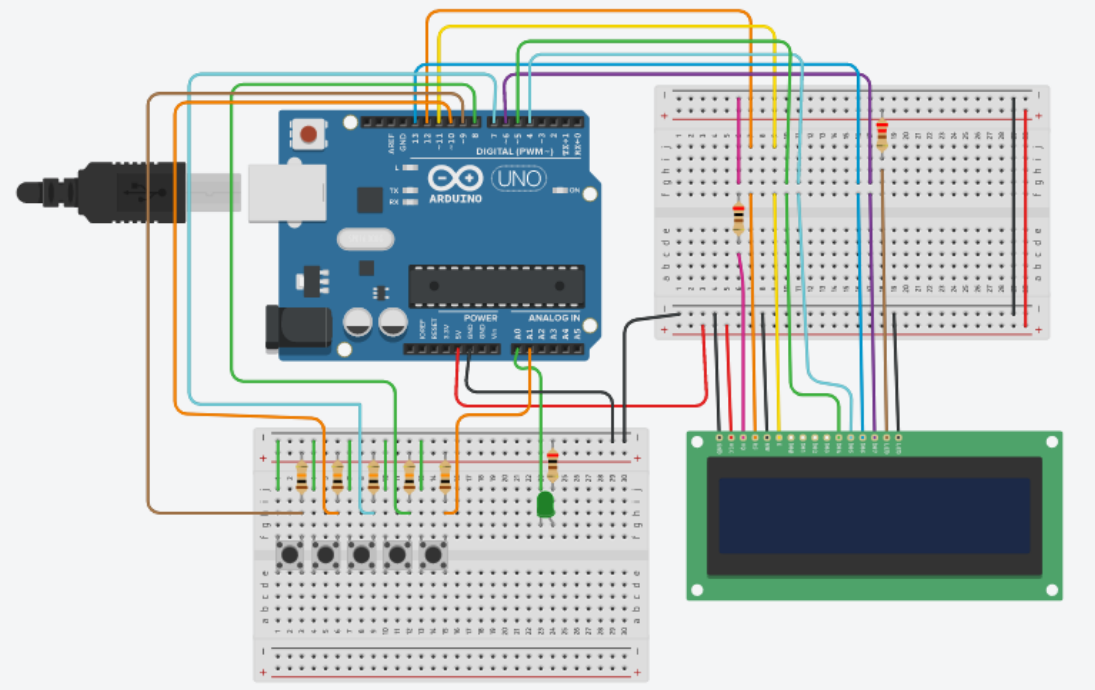
Este es el link del código fuente de la Parte A:

https://drive.google.com/file/d/1qR5JvrXl6bo09LhLJN7QuCQdW03uKUB-/view?usp=sharing

PARTE B

En esta parte nos encargaremos de programar el modulo Arduino UNO.

El esquema de conexión es el siguiente:



Este es el link del código fuente de la Parte B:

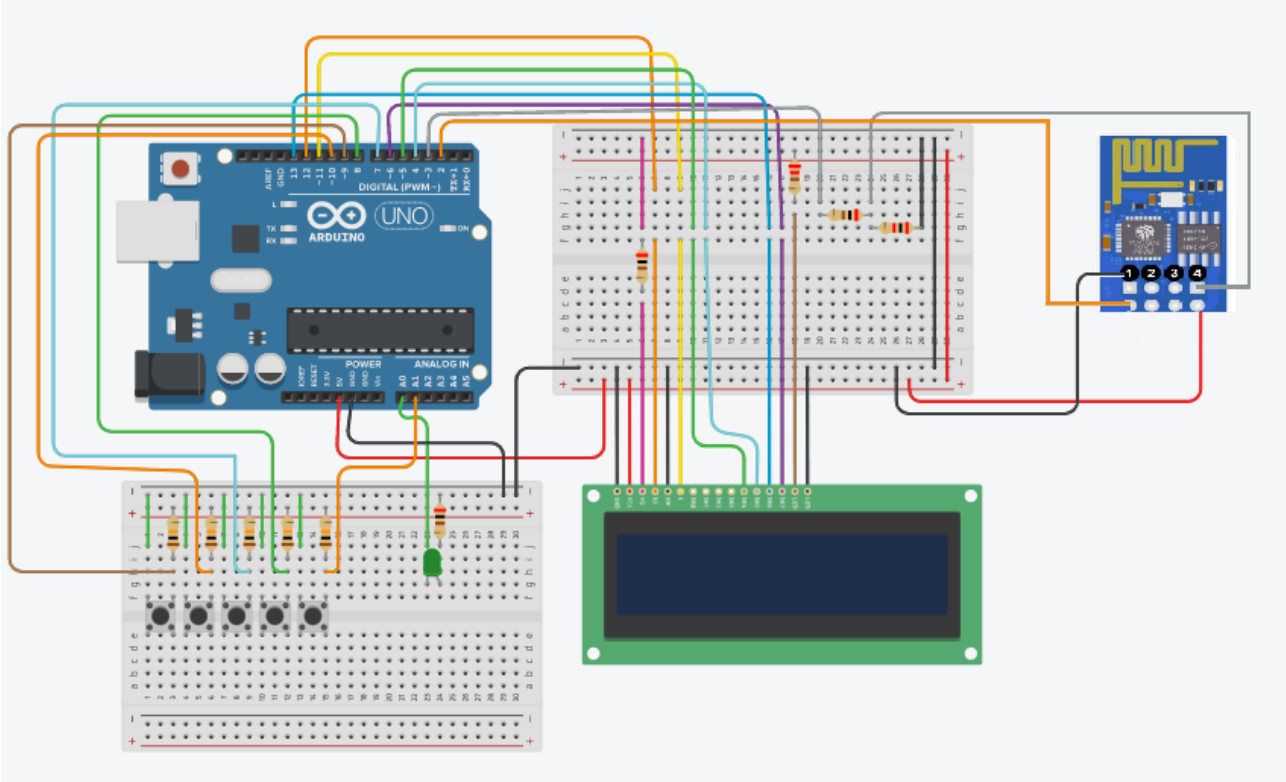
https://drive.google.com/drive/folders/1i8bY_HMmaKaZUThanPD9W_PIsW3RXY8j?usp=sharing

El archivo .INO se deberá compilar una vez que las carpetas EADDNS y CLIENTEDDNSPARTEB_LIB estén copiadas en el directorio de librerías de Arduino.

INTEGRACIÓN

Para finalizar, bastará con integrar ambos módulos como se indica en la siguiente imagen.

Puede energizarse mediante el USB de una computadora o una fuente como la descripta en los materiales utilizados. Recomendamos el uso de la segunda opción dado que aporta mayor corriente eléctrica lo que el modulo ESP8266 puede requerir para funcionar sin sobresaltos.



EXPERIENCIA

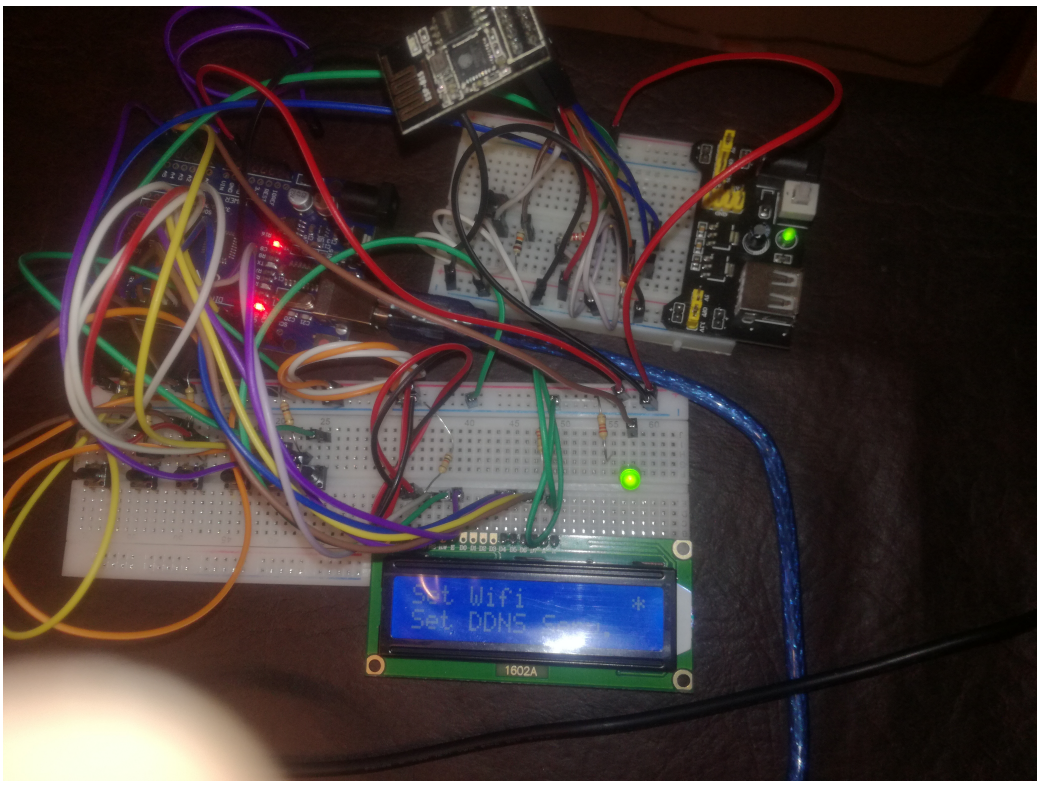
La comunicación de los módulos por medio de SoftwareSerial generó demoras en el proyecto ya que interviene negativamente en la performance del loop de arduino. También la velocidad del puerto tuvo que ser configurada en 57600 para que no genere errores de comunicación.

La lógica del menú y control por LCD requirió muchas horas, las cuales pudieron ser menos realizando un programa que permita la configuración vía web. Así también se evitaría el uso de SoftwareSerial.

La interfase de uso de la memoria EEPROM, para persistir los datos, difiere para ESP8266 de lo que es para una placa Arduino. Varias funciones o procedimientos que simplifican las tareas de manejo de EEPROM y que funcionan en Arduino UNO no funcionan para ESP8266.

El manejo óptimo de la memoria es crucial para lograr que el programa logre el funcionamiento adecuado.

El proyecto en desarrollo se ve de la siguiente manera:


Set Wifi *
1602A

Referencias:

Código fuente del proyecto:

https://github.com/alonsoem/DDNS_Client_LCD_Controlled

Conexión y configuración de un modulo ESP8266

https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/

Ejemplo de conexión de pantalla LCD

https://www.geekfactory.mx/tutoriales/tutoriales-arduino/pantalla-lcd-16x2-con-arduino/

Documentación oficial sobre SoftwareSerial

https://www.arduino.cc/en/Reference/SoftwareSerial

Fuentes de librería EASYDDNS

https://github.com/ayushsharma82/EasyDDNS