

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Programando directamente un ESP8266

MEMORIA

Autor: Carles Ubach Granados
Director: Manuel Moreno Eguílaz
Convocatoria: Abril 2017



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resumen

El presente proyecto trata de definir, desarrollar e implementar una mejora en el método de ensayo de ruido de coche según Reglamento UN 51.03 implementado por la directiva europea 2007/46, mediante el uso de un ESP8266.

Dicho ensayo se lleva a cabo en las pistas de ensayos de ruido ISO-1 e ISO-2 en Applus+ IDIADA situado en l'Albornar.

La mejora se lleva a cabo programando directamente un ESP8266 para que opere de forma independiente y se pueda utilizar como punto de acceso. La finalidad es poder acceder a toda la información que nos pueda aportar, sin necesidad de acceso a internet.

Este proyecto está estructurado en varias partes, cada una de ellas destinadas al estudio particular de cada uno de los factores que influyen en la metodología de ensayo y en la alternativa desarrollada.

- El primero de los bloques definirá que es un ESP8266 para poder entender sus limitaciones de diseño, se verán sus diversas variantes y en que se diferencian. Además, se hablará de su procesador, un componente común en todas sus variantes, y cuál es su arquitectura. Seguidamente se definirá como podemos comunicarnos con el MCU, y de los lenguajes que puede interpretar.
- El segundo bloque desarrollara lo que son las homologaciones y lo que es un Reglamento UN, de dónde proviene y cuál es su campo de aplicación. Seguidamente se hablará del reglamento de ruido R51 y en particular de la enmienda 03; se explicará su método de ensayo y los términos técnicos a cumplir. Se presentará la pista y se enseñará el lugar de ensayo dentro de los límites de la confidencialidad.
- El tercer y último bloque contendrá toda la información relativa a la mejora que se quiere implementar, y se desarrollará en su totalidad. Se especificará en que aspectos difiere con el método de ensayo actual y en qué aspectos se considera que es mejor.

Sumario

RESUMEN	1
SUMARIO	2
1. GLOSARIO	5
2. PREFACIO	7
2.1. Origen del proyecto	7
2.2. Motivación del proyecto	7
3. INTRODUCCIÓN	9
3.1. Objetivos del proyecto	9
3.2. Alcance del proyecto	9
4. ESP8266	10
4.1. ¿Qué es?	10
4.2. Especificaciones del ESP8266:.....	11
4.3. Procesador:	11
4.3.1. Arquitectura:	12
4.4. Variantes:.....	13
4.5. ESP-12	20
4.6. Conversor ADC.....	22
4.7. Plataformas.....	24
4.7.1. Arduino:	24
4.7.2. Lua NodeMCU:.....	25
4.7.3. Javascript:.....	26
4.7.4. ESP8266 Basic:.....	27
4.8. Conectando con ESP8266:	28
4.8.1. Convertidores USB → UART.....	29
4.8.2. Comandos AT.....	29
4.9. NodeMCU 1.0:	30
5. REGLAMENTO UN R51.03:	33
5.1. Homologación:.....	33
5.1.1. Naciones Unidas:.....	35
5.1.1.1. Reglamentos UN:.....	35
5.1.1.1.1.....Reglamento de ruido R51.03:	35

5.1.1.1.1	Objetivo y alcance del reglamento:	35
5.1.1.1.2	Definiciones	36
5.1.1.1.3	Procedimiento de ensayo:	37
5.1.1.1.3.1	El cálculo teórico:	37
5.1.1.1.3.2	El ensayo en la pista de pruebas:	38
5.1.1.1.4	Mediciones:	39
5.1.1.1.5	Resultado final (L_{urban}):	39
5.1.1.1.6	Applus+ IDIADA:	39
6.	APLICACIÓN DEL ESP8266 EN EL METODO DE ENSAYO:	41
6.1.	VBOX:	41
6.2.	Situación actual:	42
6.3.	Problema:	43
6.4.	Solución:	43
6.4.1.	Acceso:	44
6.4.2.	Interfaz y uso:	45
6.4.3.	Código:	47
	CONCLUSIONES	49
	AGRADECIMIENTOS	50
	Bibliografía	51
	Referencias bibliográficas	51

1. Glosario

MCU: Micro Controller Unit;

SoC: System on a Chip;

IoT: Internet of things; Internet de las cosas es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet.

USB: Universal Serial Bus

UART: Universal Asynchronous Receiver-Transmitter

UN: United Nations;

ADC: Analogic to Digital Converter

IDIADA: Instituto de Investigación Aplicada al Automóvil

2. Prefacio

2.1. Origen del proyecto

El presente proyecto tiene su origen en un proyecto académico de investigación sobre el MCU ESP8266 como componente de bajo coste poco usado en el ámbito de la ingeniería, y sobre la programación de una aplicación directamente sobre el ESP8266 mediante una comunicación USB a UART.

Posteriormente tras iniciar mi trabajo como ingeniero en homologaciones en Applus+ IDIADA, a raíz de ir consiguiendo cada vez más responsabilidades e ir aprendiendo sobre los diferentes ensayos realizados en las instalaciones; surgió la idea de aplicar el ESP8266 a un procedimiento que se lleva a cabo usualmente para realizar parciales de ruido de homologación de tipo.

El hecho de ser un trabajo que se encuentra en sus primeras fases de diseño, donde se establece la idea y un prototipo a implementar, no tiene un alcance muy avaricioso. Pero se llegan a identificar las ventajas del uso de un ESP8266, sus limitaciones y las posibles mejoras que se podrían introducir que quedan fuera del alcance de un ESP8266.

2.2. Motivación del proyecto

El motivo principal por el cual se ha seleccionado un proyecto de programación y uso de un MCU, es el interés que siempre he tenido por la programación desde que entre en la carrera. Una de mis motivaciones siempre ha sido programar ya que siempre se me ha dado muy bien cualquier lenguaje y los he aprendido todos de forma autodidacta.

Por otro lado, me fascinó el mundo de la electrónica cuando la estudié en profundidad en la ETSEIB, me fascina como llegan a funcionar todos los sistemas y el hecho de controlarlos.

Finalmente, hay que destacar que prácticamente todo lo que nos rodea cada vez se vuelve más automatizado y más electrónico; cada vez más la vida se nos facilita tanto en comodidad en nuestro día a día, como en depurar errores a la hora de realizar procesos industriales. Por eso he creído oportuno aplicar el ESP8266 a un proceso de ingeniería para facilitarlo.

3. Introducción

3.1. Objetivos del proyecto

El objetivo del presente proyecto es el de definir, programar y justificar (siempre en cumplimiento con la normativa vigente) la implementación de una mejora en el proceso de homologación de un sistema de ruido para categorías de vehículos pesados.

Se intenta optimizar el tiempo de ensayo, y además depurar posibles errores de comunicación.

Se trata de programar un ESP8266 para que pueda operar como punto de acceso, y se puedan ver los datos de un VBOX (equipado en el vehículo ensayado en movimiento) a tiempo real para determinar si la pasada del vehículo cumple con las condiciones objetivo del ensayo, mientras se realizan las mediciones de ruido de dicho vehículo.

3.2. Alcance del proyecto

No se pretende inventar un nuevo producto, ni sustituir a nivel profesional ningún equipo utilizado en la metodología del ensayo de ruido, al menos en la fase en la que se encuentra en proyecto.

Trata de encontrar una aplicación del ESP8266 en el ámbito de la ingeniería, desarrollar esa aplicación y conocer el funcionamiento del ESP8266.

En ningún momento se tomará el programa hecho como criterio único en lo que concierne a los límites de cumplimiento del ensayo y mucho menos sin antes de pasar por el sistema de calidad.

Además, dicho proyecto no conllevará ningún tipo de calibración del producto, un requisito necesario para todos los elementos involucrados en las mediciones de los ensayos que se realizan en Applus+ IDIADA:

El alcance será meramente académico a modo de demostración de las capacidades de un ESP8266 y del estudio de dicho MCU.

4. ESP8266

4.1. ¿Qué es?

ESP8266 es el nombre de un micro controlador diseñado por Espressif Systems, una compañía china con sede en Shangai. El volumen de producción de estos micro controladores no empezó hasta principios de 2014.

El ESP8266 se anuncia a sí mismo como una solución autónoma de redes WiFi que se ofrece como un puente entre los microcontroladores que hasta ahora existían hasta los MCU con WiFi, siendo además capaz de ejecutar aplicaciones independientes.

Si se fuera a usar un ESP8266 salido de fábrica probablemente no se sabría qué hacer. Y es gracias a que los fabricantes los construyen encima de circuitos impresos y placas pre-fabricadas, que estos quedan listos para nuestro uso. Por eso se da lugar a varias versiones de ESP8266 (Ej: ESP-01, ESP-02...) pero todas con el mismo procesador, lo que las diferencian son el número de pines GPIO expuestos, la cantidad de memoria flash, las dimensiones, la forma de exponer los pines, y otras consideraciones varias relativas a su construcción. Pero desde una perspectiva de programación todas son iguales.

El circuito integrado ESP8266 viene en un pequeño paquete, tal vez cinco milímetros cuadrado. Obviamente, a menos que se sea un maestro soldador no va a hacer mucho con eso. Las buenas noticias son que una serie de proveedores han creado placas breakout que nos hacen el trabajo mucho más simple, ya que estas tarjetas contienen un único componente que, debido a sus reducidas dimensiones no pueden ser conectados directamente a una placa de desarrollo para sus salidas y por lo que son "rotos" o ampliados con el fin de tener acceso a sus terminales.

Esto nos permite trabajar con este componente único acoplado a un microcontrolador, para desarrollar un proyecto para evaluar su funcionamiento, o para complementar un proyecto que requiere su presencia. Este es el caso de un circuito integrado que tiene entonces sus pines accesibles de modo que puedan ser fácilmente conectados a un microcontrolador, es equivalente a una matriz de contacto, pero en la forma de una placa de circuito impreso con algún componente o componentes ya soldadas

4.2. Especificaciones del ESP8266:

Tabla 1: Características generales

Voltaje	3.3 V
Consumo de corriente	10 μ A – 170 mA
Memoria Flash	16 MB máx. (512 k normal)
Procesador	Tensilica L106 32 bit
Velocidad del procesador	80 – 160 MHz
GPIOs	17
Analógico a digital	1 entrada con 10 bit de resolución (1024 valores)
Soporte de 802.11	b/g/n/d/e/i/k/r
Máximas conexiones simultáneas	5

4.3. Procesador:

El system on a chip (SoC) ESP9266EX usa un microcontrolador Tensilica Xtensa L106, que es un procesador de 32 bit con instrucciones de 16 bit.



Ilustración 1 Procesador ESP8266EX

El SoC describe la tendencia cada vez más frecuente de usar tecnologías de fabricación que integran todos o gran parte de los módulos que componen un computador o cualquier otro sistema informático o electrónico en un único circuito integrado o chip.

Como podemos ver a continuación:

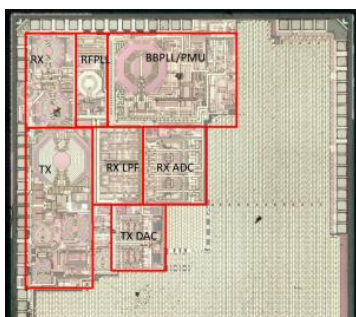


Ilustración 2 Interior de un ESP8266 con periféricos anotados

El procesador funciona por defecto a 80 MHz pero puede ir hasta 160 MHz, tiene ~ 80kB de DRAM (Data RAM), y ~ 35kB IRAM (Instruction RAM). La IRAM se carga en el arranque con lo que el usuario quiere mantener en el procesador, aunque el procesador puede ejecutar el código directamente fuera del flash externo a una velocidad más baja.

4.3.1. Arquitectura:

Tiene una arquitectura de Harvard, con lo cual la CPU puede tanto leer una instrucción como realizar un acceso a la memoria de datos al mismo tiempo, incluso sin una memoria caché. En consecuencia, una arquitectura de computadores Harvard puede ser más rápida para un circuito complejo, debido a que la instrucción obtiene acceso a datos y no compete por una única vía de memoria.

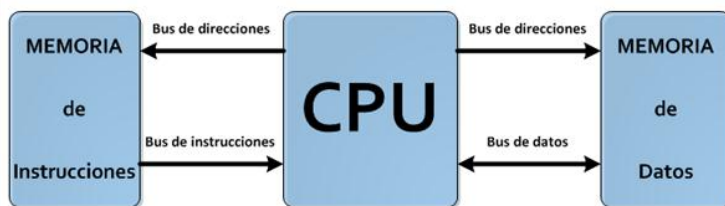


Ilustración 3 Arquitectura de Harvard

4.4. Variantes:

Como bien se ha comentado con anterioridad el ESP8266 tiene siempre el mismo procesador, pero su versión varia a la hora de construirlo sobre una placa impresa ya que sus características de construcción difieren en diferentes aspectos; los cuales se pincelan a continuación.



Ilustración 4 Evolución de las variantes de ESP8266

ESP-01:

El primero de ellos y uno de los más populares,

- GPIOs: 2 Pines GPIO (GPIO 0/2)
- Dimensiones: 14,3 x 24,8mm
- Antena: Impresa

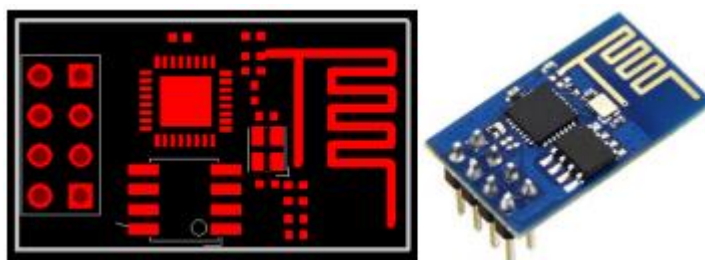


Ilustración 5 ESP-01

ESP-02:

- GPIOs: 3 (GPIO0/2/15)
- Dimensiones: 14,2x14,2mm
- Antena: Externa (U-FL)

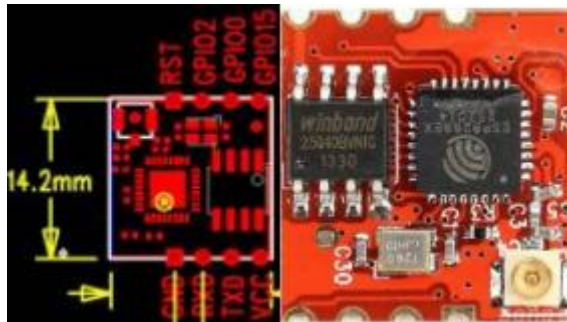


Ilustración 6 ESP-02

ESP-03:

- GPIOs : 7 Pines GPIO (GPIO0/2/12/13/14/15/16)
- Dimensiones: 17,3 × 12,1mm
- Antena: Cerámica

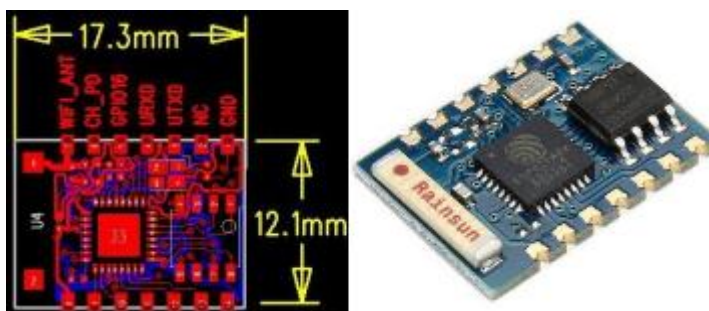


Ilustración 7 ESP-03

ESP-04:

- GPIOs : 7 Pines GPIO (GPIO0/2/12/13/14/15/16)
- Dimensiones: 14,7 x 12,1mm
- Antena: Ninguna

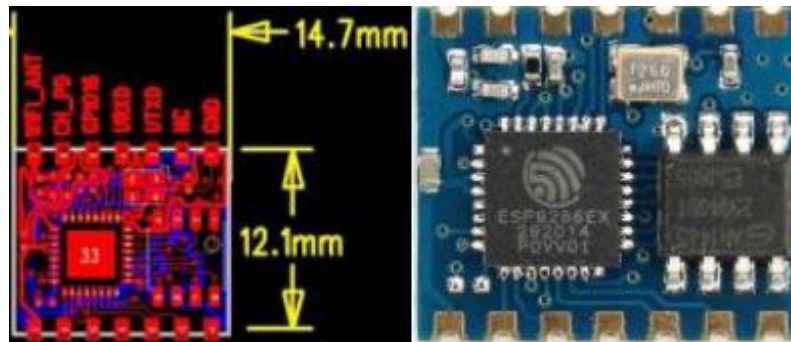


Ilustración 8 ESP-04

ESP-05:

- GPIOs: Ninguno
- Dimensiones: 14,2 x 14,2mm
- Antena: Externa (U-FL)

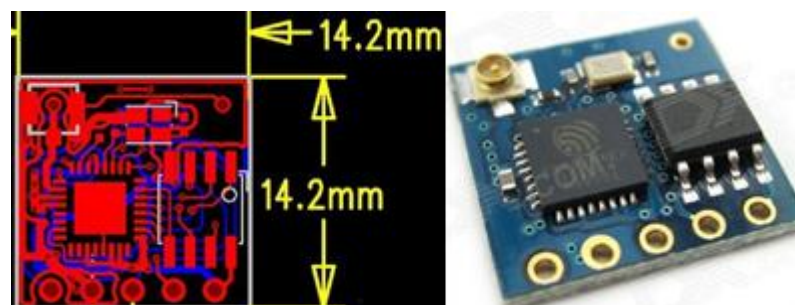


Ilustración 9 ESP-05

ESP-06:

- GPIOs: 3 (GPIO0/2/15)
- Dimensiones: 14,2x14,7 mm
- Antena: Ninguna

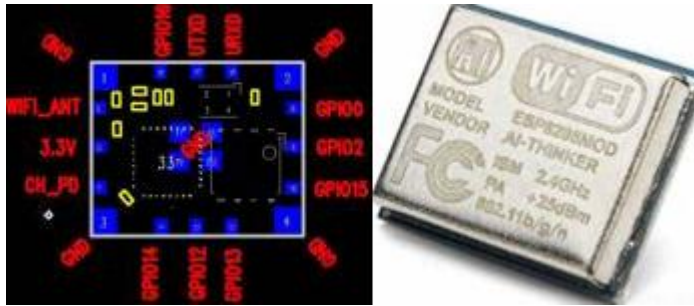


Ilustración 10 ESP-06

ESP-07:

- GPIOs: 9(GPIO0/2/4/5/12/13/14/15/16)
- Dimensiones: 22 x 16mm
- Antena: Cerámica y Externa (U-FL)

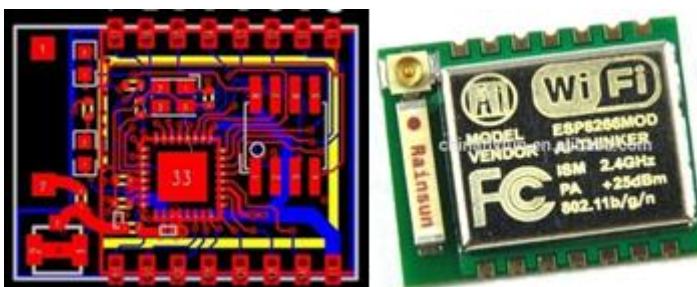


Ilustración 11 ESP-07

ESP-08:

- GPIOs: 7 (GPIO0/2/12/13/14/15/16)
- Dimensiones: 17x16mm
- Antena: Ninguna

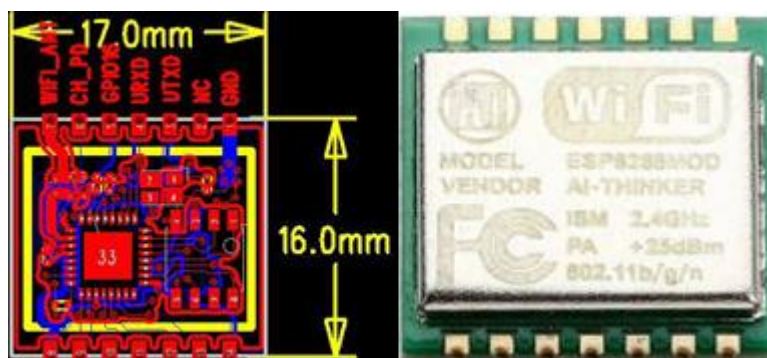


Ilustración 12 ESP-08

ESP-09:

- GPIOs: 6 (GPIO0/2/12/13/14/15)
- Dimensiones: 10x10mm
- Antena: Ninguna

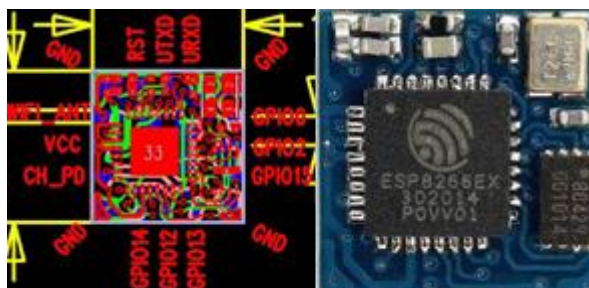


Ilustración 13 ESP-09

ESP-10:

- GPIOs: Ninguno
- Dimensiones: 14,2x10 mm
- Antena: Ninguna

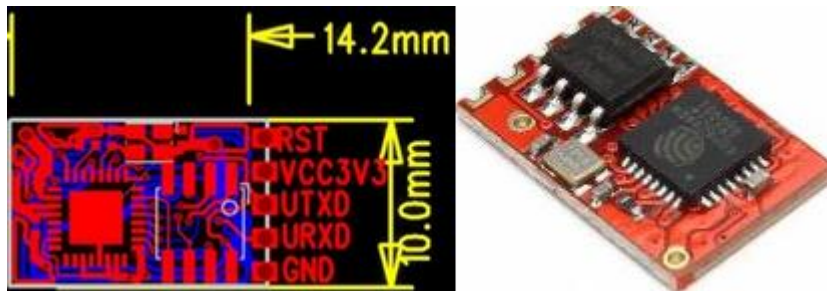


Ilustración 14 ESP-10

ESP-11:

- GPIOs: 2 (GPIO0/2)
- Dimensiones: 19,3x13 mm
- Antena: Cerámica

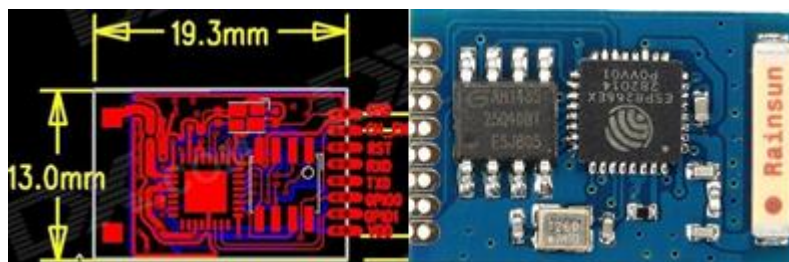


Ilustración 15 ESP-11

ESP-12:

- GPIOs: 9 GPIOs + ADC
- Dimensiones: 24x16 mm
- Antena: Impresa

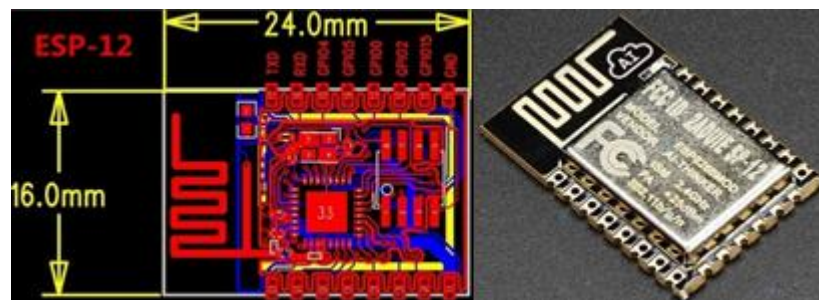


Ilustración 16 ESP-12

ESP-13:

- GPIOs: 9 GPIOs + ADC
- Dimensiones: 18x20 mm
- Antena: Impresa

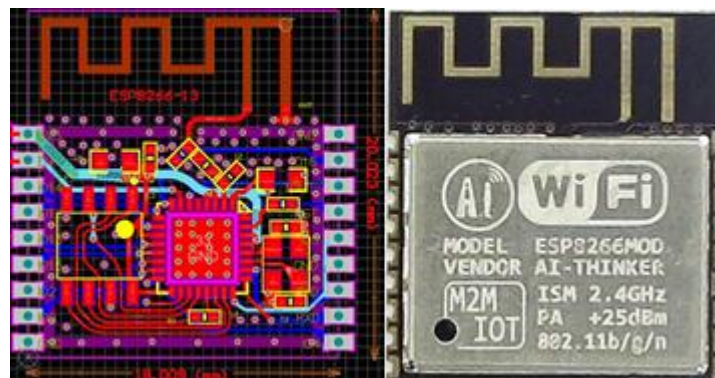


Ilustración 17 ESP-13

4.5. ESP-12

De todos los modelos de ESP8266, para este proyecto se utilizará un ESP-12.

En el proyecto se ha usado un módulo NodeMCU del que se hablará posteriormente, este módulo utiliza un ESP-12 como base, con la ventaja de que viene con un conector USB a UART incorporado en una misma placa.

El módulo NodeMCU se utilizará por comodidad, pero además a modo de testeo y didacta se ha utilizado y cableado un ESP-12 como se explica a continuación. Ya que uno de los objetivos del presente trabajo es aprender sobre el ESP8266, y no sobre un módulo prefabricado.

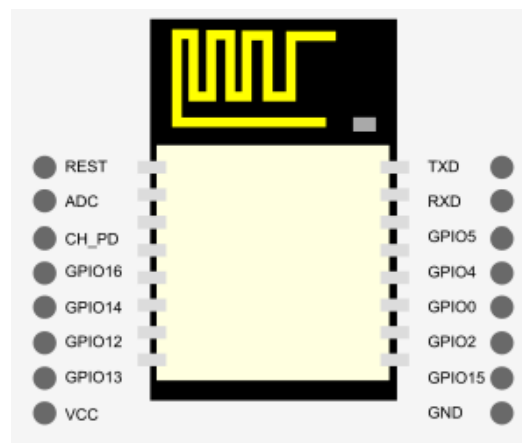


Ilustración 18 Distribución pines ESP-12

El ESP-12 es la versión actual más popular y flexible disponible hoy en día, siendo esta una de las que expone la mayor cantidad de pines GPIO para su uso.

El ESP-12 tiene un LED azul montado en la parte superior derecha que parpadea cuando hay tráfico UART.

A continuación, se muestra una breve descripción de sus pines:

Tabla 2 Características de los pines

Nombre	Descripción
VCC	3.3 V
GPIO 13	También usado como MOSI
GPIO 12	También usado como MISO
GPIO 14	También usado como Clock

GPIO 16	
CH_PD	Habilitador del chip. Debe estar en HIGH para funcionamiento normal
ADC	Entrada analógica a digital
RESET	Reset externo por lógica negativa
TXD	UART 0 transmitir
RXD	UART 0 recibir
GPIO 4	GPIO regular
GPIO 5	GPIO regular
GPIO 0	Debería estar en HIGH en el arranque, LOW para actualizar la memoria flash
GPIO 2	Debería estar en HIGH en el arranque
GPIO 15	Debería estar en LOW en el arranque y en el flasheo
GND	Tierra

A continuación, se muestra el esquema electrónico que se ha seguido:

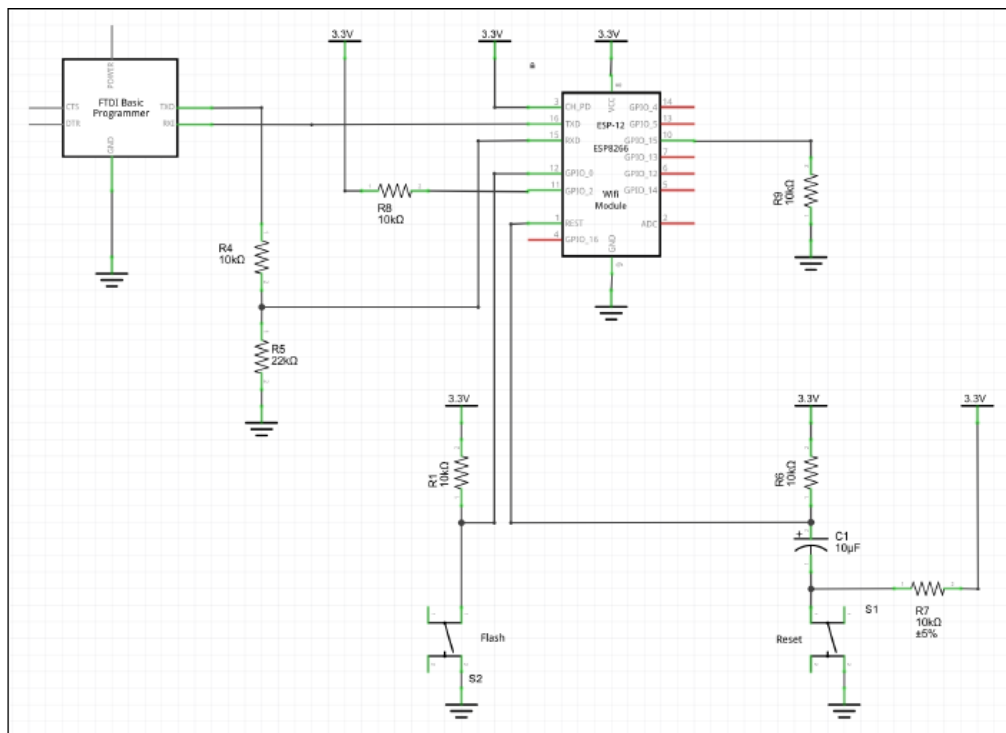


Ilustración 19 Esquema electrónico del montaje

Finalmente se muestra el circuito construido sobre una placa de pruebas:

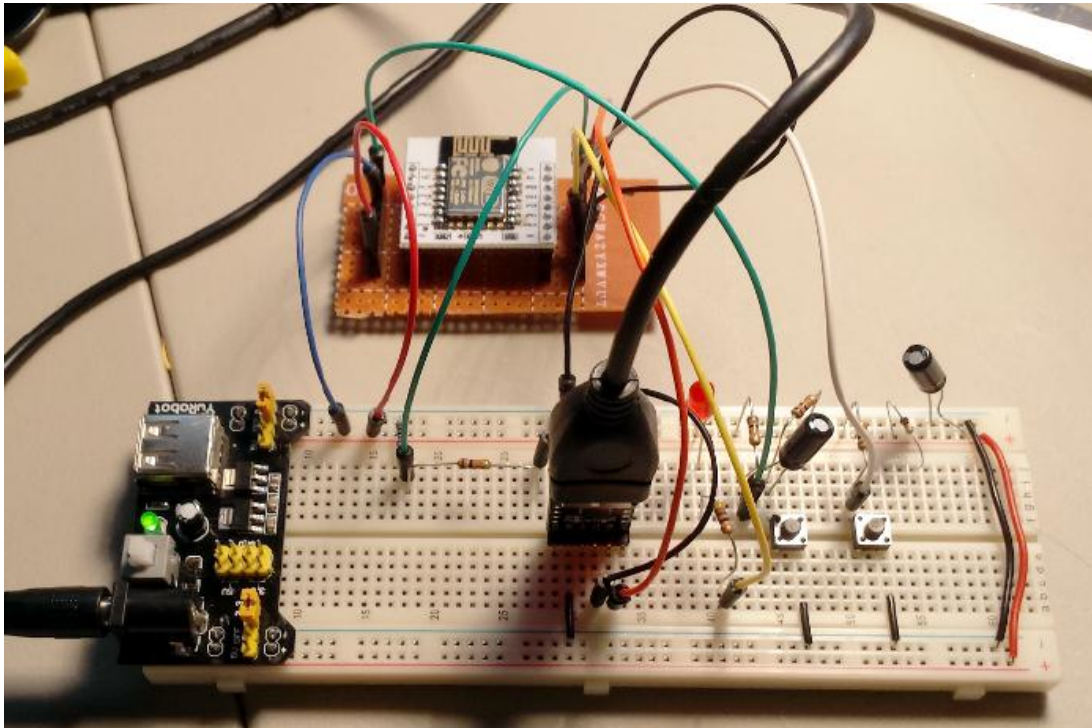


Ilustración 20 Muestra de circuito construido

4.6. Conversor ADC

La Conversión de analógico a digital es la capacidad de leer un nivel de voltaje de un pin entre 0 y algún valor máximo, para convertir ese voltaje analógico en una representación digital.

La variación de la tensión aplicada al pin cambiará el valor leído. El ESP8266 tiene un convertidor analógico a digital incorporado con una resolución de 1024 valores distintos.

Es decir que a 0 voltios producirá un valor digital de 0 mientras que al voltaje máximo produce un valor digital de 1023, los rangos de voltaje entre 0 y el voltaje máximo producirá un valor digital correspondientemente escalado.

Para leer el valor digital del voltaje analógico, la función llamada `system_adc_read ()` debe ser llamada, que en el caso de ESP8266Basic es tan simple como el comando `io(ai)`.

El pin físico en el ESP8266 del cual se lee el voltaje se llama TOUT y no sirve para ningún otro propósito.

El rango de entrada en el pin ADC es de 0V a 1V, esto implica que el voltaje de entrada a la ADC no puede ser la tensión máxima utilizada para alimentar el propio ESP8266 (3.3V). Así que se tendrá que usar un circuito divisor de tensión.

Como se muestra a continuación una resistencia una resistencia R2 de 10k, con una resistencia R1 de 22k es suficiente para el montaje.

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in} \quad R_1 = \frac{10000 \cdot 3.3}{1.0} - 10000 = 23000$$

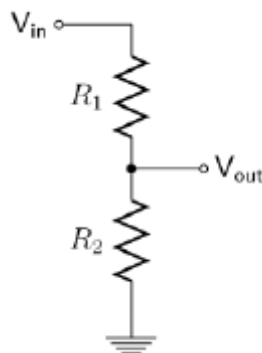


Ilustración 21 Esquema divisor de tensión

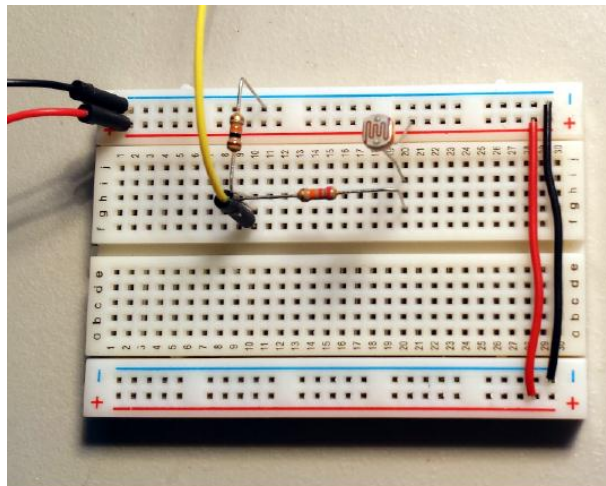


Ilustración 22 Montaje de un divisor de tensión conectado al pin A0

4.7. Plataformas

4.7.1. Arduino:

Mucho antes de que hubiera un ESP8266, había el Arduino. Una aportación de vital importancia a la comunidad de hardware de código abierto y el punto de entrada para la mayoría de los aficionados en el mundo de los circuitos y procesadores construidos en casa.

Una de las principales atracciones del Arduino es su baja complejidad, permitiendo a cada uno la capacidad de construir algo rápidamente y fácilmente. El entorno de desarrollo integrado (IDE) para el Arduino siempre ha sido gratuito para descargar desde Internet.

Además de proporcionar un editor de lenguaje C más herramientas para compilar e implementar, Arduino IDE proporciona bibliotecas pre-suministradas de rutinas C que "ocultan" complejos detalles de la implementación que de otra manera podrían ser necesarios cuando se programan las placas de Arduino.

Por ejemplo, la programación UART sin duda tendría que establecer registros, manejar interrupciones y más. En lugar de hacer que los usuarios tengan que aprender estas APIs técnicas, la gente de Arduino proporcionó bibliotecas de alto nivel que podrían llamarse desde los bocetos con interfaces más limpias. Esta noción es clave ya que estas bibliotecas, tanto como cualquier otra cosa, proporcionan el entorno para los programadores de Arduino.

Arduino entra en conexión con ESP8266 cuando una cantidad de personas con talento han construido un proyecto Open Source Github que proporciona un "plug-in" o "extensión" a la herramienta IDE Arduino. Lo que hace esta extensión es permitir escribir bocetos en el IDE de Arduino que aprovechan las interfaces de la biblioteca Arduino que, en tiempo de compilación y despliegue, generan código que se ejecutará en el ESP8266.

Lo que esto significa efectivamente es que podemos usar el IDE de Arduino y construir aplicaciones ESP8266 como si fuera un Arduino.

El ESP8266 Arduino además incluye librerías para comunicar con WiFi utilizando TCP y UDP, crear servidores, usar un sistema de archivos en la memoria flash, trabajar con tarjetas SD, servos, y demás periféricos.

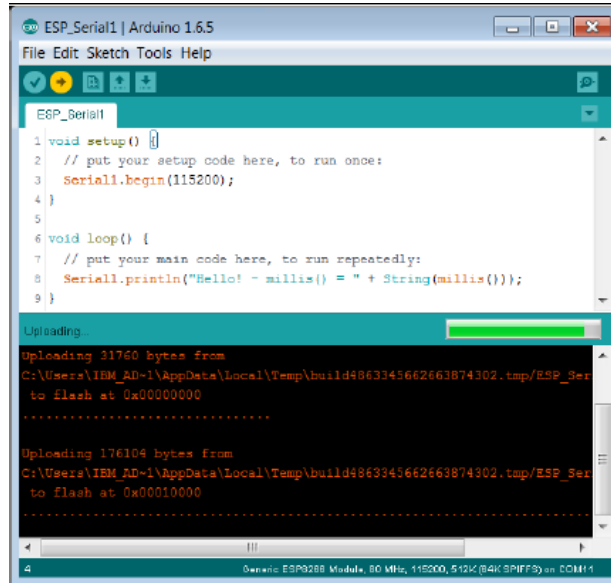


Ilustración 23 Ejemplo Arduino IDE

4.7.2. Lua NodeMCU:

Es un firmware para el ESP8266 basado en el Espressif Non-OS SDK y usa el lenguaje de programación Lua.

Lua es un potente lenguaje de scripting disponible en entornos de ESP8266. La aplicación más popular de Lua para el ESP8266 se conoce como el NodeMCU Lua Firmware y está disponible en su repositorio github.

Una vez que tengas una copia del firmware, puedes flashearla usando una herramienta de flasheo para el ESP8266, como puede ser la propia del NodeMCU.

No se describirá el lenguaje Lua en este trabajo, ya hay excelentes libros ya escritos en Lua y también se pueden encontrar referencias y tutoriales en Internet.

El IDE de ESPlorer es un entorno de desarrollo para la construcción de aplicaciones Lua para el ESP8266.

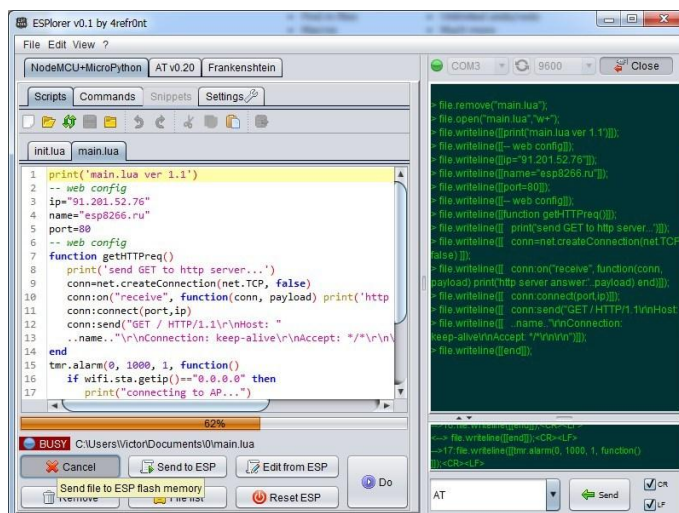


Ilustración 24 Ejemplo ESPlorer IDE

4.7.3. Javascript:

El JavaScript es un lenguaje de alto nivel interpretado. Algunas de sus construcciones centrales son mecanografía suelta, orientación a objetos, soporte de funciones lambda, soporte de cierres y, lo que es más importante, se ha convertido en el lenguaje de la web. Si uno está escribiendo una aplicación alojada en el navegador, entonces es una certeza que se escribirá en JavaScript.

Por un tiempo ahora JavaScript ha estado ganando terreno en código de servidor a través de proyectos como Node.js. Como un lenguaje para ejecutar código de servidor, tiene un conjunto significativo de características para realizar esta capacidad.

Específicamente, apoya un paradigma de arquitectura impulsado por eventos. En JavaScript, podemos registrar funciones que se devolverán cuando se detecten eventos. Estas retrollamadas se pueden definir como simples funciones en línea sobre qué hacer.

Si se logra implementar un buen modelo JavaScript, se correlacionará excelentemente con el modelo ESP8266 el cual es impulsado por eventos a través de retrollamadas.

Espruino es un proyecto de código abierto para proporcionar un tiempo de ejecución de JavaScript para dispositivos integrados. Se ha implementado para los procesadores ARM Cortex M3 / M4 y otros. Actualmente está en desarrollo para implementarlo en el ESP8266.

4.7.4. ESP8266 Basic:

Basic es un idioma amado por millones de personas. Es como comenzó Microsoft y una de las razones para el crecimiento explosivo de las computadoras en los años 80. Basic es un lenguaje simple pero potente que te permite hacer cosas increíbles sin necesidad de un título en ciencias de la computación.

Además de proporcionar un editor de lenguaje Basic en la propia página, ESP8266 Basic proporciona bibliotecas pre-suministradas de rutinas que "ocultan" complejos detalles de la implementación que de otra manera podrían ser necesarios.

El entorno de desarrollo como bien se ha dicho es vía web, donde accederemos al propio ESP8266 y podremos escribir nuestro código para flashearlos. Además, dispone de un sistema de archivos en la memoria flash.

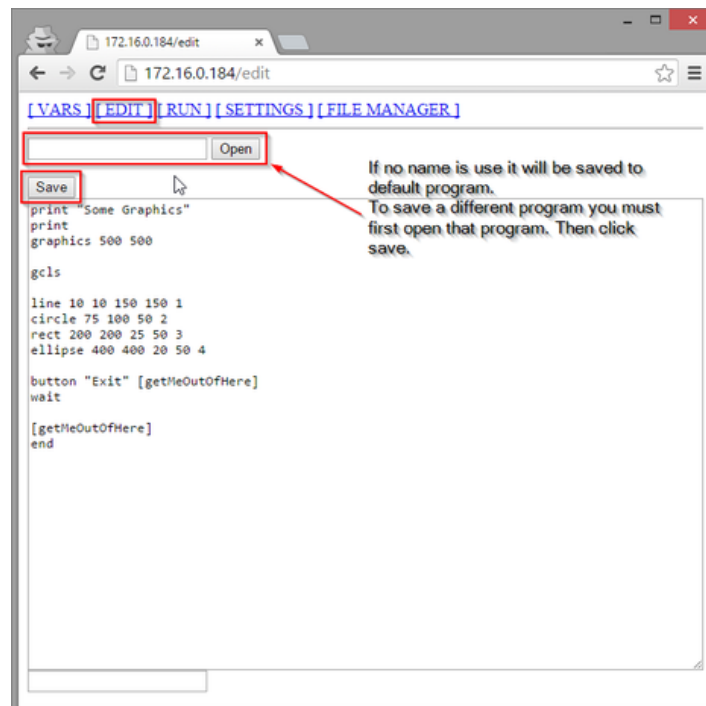


Ilustración 25 Ejemplo ESP8266 Basic IDE

4.8. Conectando con ESP8266:

El ESP8266 es un dispositivo WiFi y, por lo tanto, finalmente se conectará a él mediante protocolos WiFi, pero primero se requiere un arranque.

El dispositivo no sabe a qué red se ha de conectar, la contraseña a utilizar ni otros parámetros necesarios. Además, si queremos que opere como punto de acceso para poder interactuar con el ESP8266 como en nuestro caso, debemos cargar al dispositivo nuestra propia aplicación con la que deberemos interactuar.

Esto quiere decir que existe otra forma de interactuar con el dispositivo además de los protocolos WiFi, esta otra forma es mediante el UART (Puerto Serie). El propio ESP8266 tiene una interfaz UART dedicada con pines llamados TX y RX.

Como se ha comentado con anterioridad el pin TX es usado para transmitir datos desde el ESP8266 y el pin RX lo utiliza el dispositivo para recibir datos. Estos pines se pueden conectar a un interlocutor UART, que en nuestro caso será un convertidor USB → UART.

A través de la UART, podemos adjuntar un emulador de terminal para enviar pulsaciones de teclas y mostrar los datos recibidos por el ESP8266 como caracteres en la pantalla. Esto se utiliza ampliamente si trabajamos con los comandos AT.

Cuando usamos un UART, tenemos que considerar el concepto de velocidad de transmisión. Esta es la velocidad de comunicación de datos entre el ESP8266 y su socio. Durante el arranque, el ESP8266 intenta determinar automáticamente la velocidad en baudios del interlocutor y emparejarla, asume un valor predeterminado de 74880.

El ESP8266 tiene un segundo UART asociado con él que sólo se emite. Uno de los propósitos principales de este segundo UART son los diagnósticos de salida y depuración de información. Esto puede ser extremadamente útil durante el desarrollo y como tal, se recomienda conectar dos convertidores USB → UART al dispositivo. Este segundo UART se multiplexa con el pin GPIO2.

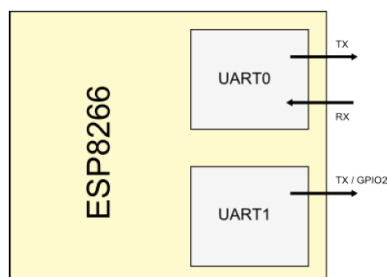


Ilustración 26 Esquema de los UART incorporados en el ESP8266

4.8.1. Convertidores USB → UART

Como bien hemos comentado no se puede programar un ESP8266 sin suministrar datos a través de un UART. La manera más fácil de lograr esto es a través del uso de un convertidor USB a UART.

Para el montaje cableado se ha utilizado uno basado en CP2102.



Ilustración 27 USB a UART basado en CP2102

4.8.2. Comandos AT

La manera más rápida y fácil de empezar con un ESP8266 es acceder a él a través de la interfaz de comandos AT.

Cuando pensamos en un dispositivo ESP8266, encontramos que tiene una conexión UART (Serial). Esto significa que puede enviar y recibir datos utilizando el protocolo UART. También sabemos que el dispositivo puede comunicarse con WiFi.

¿Qué pasaría si tuviéramos una aplicación que funcionara en el ESP8266 que tomó "instrucciones" recibidas sobre el enlace serial, las ejecutó y luego devolvió una respuesta? Esto nos permitiría entonces utilizar el ESP8266 sin tener que saber nunca los lenguajes de programación que son nativos al dispositivo.

Esto es exactamente lo que hace un programa pre-instalado en el ESP8266. El programa se denomina "procesador de comandos AT" nombrado con el formato de los comandos enviados a través del enlace serie. Estos comandos están todos prefijados con "AT" y siguen (aproximadamente) el estilo conocido como "conjunto de comandos Hayes".

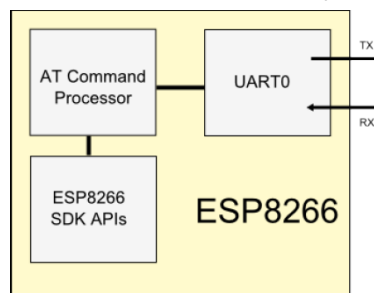


Ilustración 28 Esquema de operación del flujo de datos de comandos AT

4.9. NodeMCU 1.0:

Existe una línea open-source de placas de desarrolladores llamada NodeMCU. En el mercado podemos encontrar diferentes modelos de estas nuevas placas basadas en el ESP8266. Estos varían en su memoria interna y en la cantidad de pines de entrada/salida disponibles. En el caso del NodeMCU, esta placa fue diseñada principalmente para trabajar con Lua, aunque es posible utilizarla con cualquier plataforma con la que se le haga el flasheo.

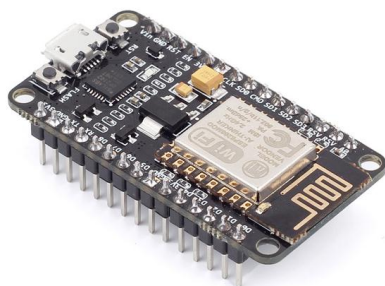


Ilustración 29 NodeMCU 1.0

Están basados en el ESP12E, el cual es una variante del ESP8266, y llevan incorporados un puente USB→UART Silicon Labs CP2102

Algunas de las características de este módulo son:

- Voltaje de entrada (USB): 5V
- Voltaje de salida en los pines: 3.3V
- Voltaje de referencia en el ADC: 3.3V
- Corriente nominal por pin: 12mA
- Frecuencia de procesador: 80MHz (160MHz max.)
- 4MB Flash
- Consumo de corriente en stand-by @80MHz: 80mA
- Consumo de corriente en stand-by @160MHz: 90mA

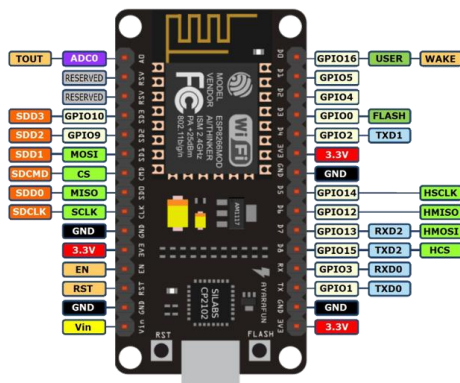


Ilustración 30 Distribución de pines NodeMCU 1.0

GPIO	Pin Name	NodeMCU	Risk
GPIO0	GPIO0_U	D3	Red
GPIO1	U0TXD_U	D10	Orange
GPIO2	GPIO2_U	D4	Orange
GPIO3	U0RXD_U	D9	Orange
GPIO4	GPIO4_U	D2	Green
GPIO5	GPIO5_U	D1	Green
GPIO6	SD_CLK_U		Red
GPIO7	SD_DATA0_U		Red
GPIO8	SD_DATA1_U		Red
GPIO9	SD_DATA2_U	SD2	Red
GPIO10	SD_DATA3_U	SD3	Red
GPIO11	SD_CMD_U		Red
GPIO12	MTDI_U	D6	Green
GPIO13	MTCK_U	D7	Green
GPIO14	MTMS_U	D5	Green
GPIO15	MTDO_U	D8	Yellow
GPIO16	???	D0	Yellow

Ilustración 31 Equivalencias de pines entre ESP-12 y NodeMCU

A términos prácticos el NodeMCU es como utilizar un ESP-12 cableado por nosotros mismos, pero con la particularidad de que en un espacio reducido tenemos todos los pines soldados y adaptados para utilizarlos en una placa de pruebas, un puente USB→UART integrado y dos pulsadores, uno para Reset y otro para Flash.

En este trabajo se introduce en la memoria flash la plataforma ESP8266 Basic, y se prescindirá de toda la interfaz que ofrece NodeMCU.

5. REGLAMENTO UN R51.03:

5.1. Homologación:

Cuando se habla de homologación automovilística, se refiere a la aprobación del vehículo para la circulación por la vía pública.

El proceso de homologación de un vehículo es un proceso largo y complicado, en el que han de encajar todas y cada una de las partes. Esto es debido a que no se homologa un vehículo individualmente (Es una posibilidad de homologación a número de bastidor, pero que no vamos a contemplar en este estudio) se homologa un modelo o varios modelos de vehículos con diferentes características que pertenecen al mismo tipo de vehículo.

La cumbre es poder matricular vehículos con número de bastidor y esto se consigue avalando que el tipo de vehículo que se quiere matricular viene cubierto por una homologación de tipo de vehículo completo. Esta homologación de tipo de vehículo completo actualmente se consigue mediante la directiva europea 2007/46, la llamada directiva marco.

Dentro de esta directiva marco se especifica cada uno de los requisitos que se han de cumplir para poder conseguir dicha homologación, que resultará en la obtención de una contraseña de homologación (A nivel español con formato e9*2007/46*XXXX*YY, siendo XXXX el número de la homologación y YY la extensión de dicha homologación de tipo).

Dichos requisitos a cumplir en la 2007/46 son entre otros obtener la homologación de cada uno de los sistemas que forman parte de un vehículo, especificados en el Anexo IV de la directiva europea 2007/46.

La homologación de los sistemas o más comúnmente llamados parciales, conlleva someter un vehículo a ensayos tanto destructivos como no destructivos. Dicho vehículo ha de ser representativo de la familia de vehículos que se quiere homologar.

El criterio para escoger el vehículo representativo se ha de hacer buscando el peor de los casos (Worst Case) de todos los modelos que se quiere cubrir en la homologación de ese parcial.

La culminación de una homologación de un sistema de un vehículo, da como resultado una contraseña de homologación, que avala que ese sistema queda homologado para todos y cada uno de los modelos contemplados en el informe que el servicio técnico ha dado como válidos contrastándolos detalladamente con la normativa que aplica para ese sistema.

La clave para encajarlo todo es la palabra tipo, las homologaciones de vehículo completo se hacen para tipo de vehículo. Y en cada uno de los Reglamentos que aplican para poder ensayar y homologar un sistema, existe el punto 2. Definiciones; donde se explica que define tipo de vehículo. Necesitando cada tipo una nueva contraseña de homologación, es decir una nueva homologación de sistema.

Así dejando la tarea realmente complicada para el servicio técnico, comprobar que los requisitos y los resultados de cada una de las homologaciones de sistema aportadas abarcan todos los modelos que se quieren introducir en la contraseña de homologación de la directiva europea 2007/46

Finalmente hay que recalcar que todos los componentes que forman parte de los sistemas del vehículo, ha de tener su contraseña de homologación como componente. La cual se consigue haciendo los ensayos pertinentes sobre los componentes.

Así pues, podemos ver el proceso de homologación como la construcción de una casa:

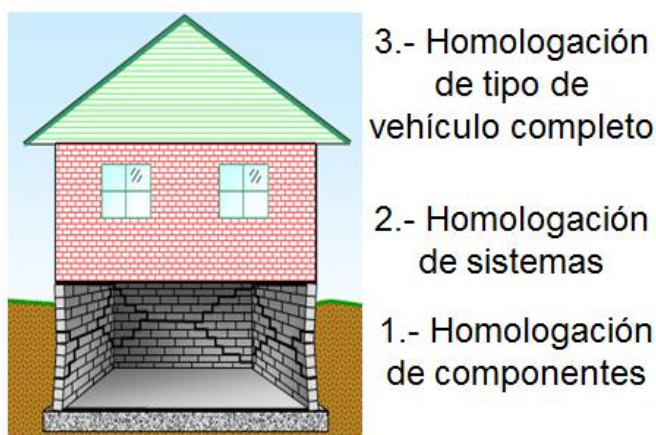


Ilustración 32 Ejemplo de los requisitos para una homologación

Cabe remarcar que todo lo anterior expuesto es el proceso de homologación actual, tras la armonización de los reglamentos a nivel europeo, y la derogación de ciertas

Anterior a la armonización de la normativa a nivel europeo relativa al automóvil, la normativa relativa a una homologación de vehículo venía tipificada en el Real Decreto 750/2010.

5.1.1. Naciones Unidas:

5.1.1.1. Reglamentos UN:

Los reglamentos de las naciones unidas no tienen ningún peso legal, pero a partir del acuerdo de Ginebra en el 1958, se establecieron como obligatorios para poder conseguir una homologación de tipo de vehículo completo. La forma de hacerlo fue haciéndolos aparecer en la Directiva europea 2007/46, que automáticamente impuso la vigencia a nivel español.

5.1.1.1.1 Reglamento de ruido R51.03:

El título oficial del reglamento “Disposiciones uniformes relativas a la homologación de los vehículos de motor de al menos cuatro ruedas con respecto a sus emisiones sonoras” como podemos ver a continuación en la ilustración 33 del fragmento sustraído del documento oficial de las Naciones Unidas.

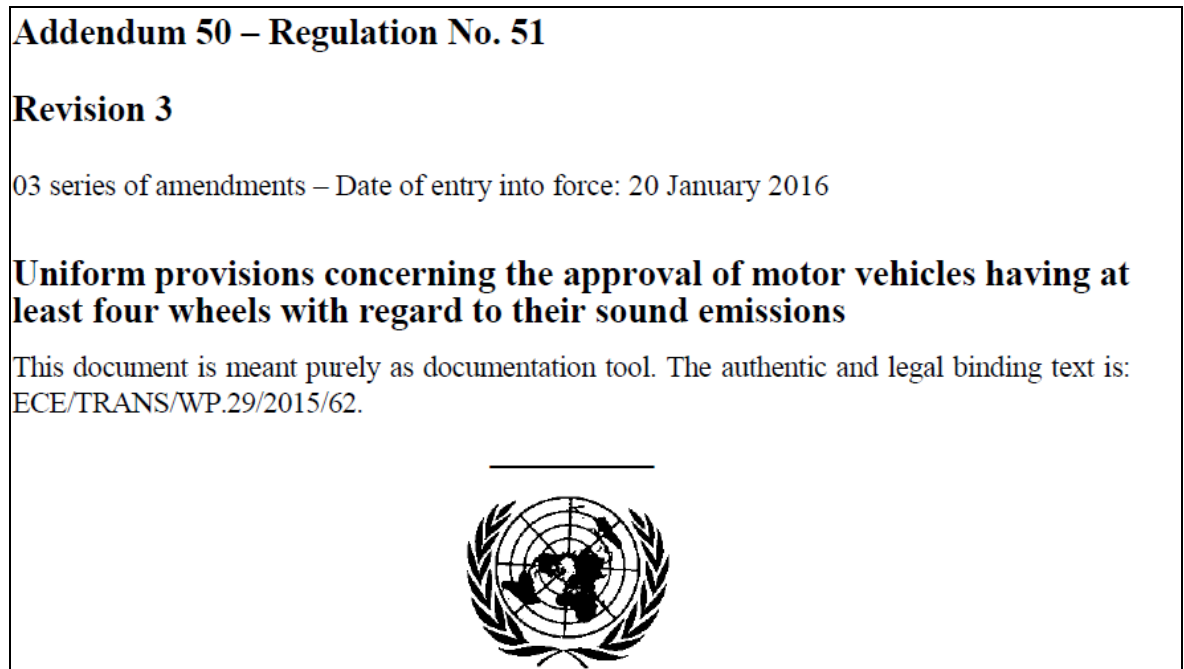


Ilustración 33 Fragmento de la portada del reglamento N° 51 Enmienda 03

5.1.1.1.1.1 Objetivo y alcance del reglamento:

Este reglamento está destinado a regular las emisiones sonoras de los motores de los vehículos a motor, en el que se establecen unas condiciones de ensayo que simulan el ruido que puede emitir un vehículo en ciudad.

El alcance son todos los vehículos de categorías M y N, pero en vehículo industrial solo se hacen los ensayos para los vehículos pesados, es decir $M2 > 3.500 \text{ kg MTMA}$, $M3$, $N2$, $N3$.








Vehículos de pasajeros		DESCRIPCION	RANGO
M_1		No mas de 8 plazas (excluida la del conductor)	NA
M_2		Mas de 8 plazas (excluida la del conductor)	Hasta 5 Tn
M_3		Mas de 8 plazas (excluida la del conductor)	Mas de 5Tn
M_1	Especiales 	Autocaravanas, ambulancias, blindados, vehículos con acceso sillas minusválidos	NA
Transporte Mercancias			
N_1		Furgonetas y camiones ligeros	Hasta 3.5 Tn
N_2		Furgonetas y camiones tamaño medio	Entre 3.5 y 12Tn
N_3		Camiones pesados	Mas de 12Tn

Ilustración 34 Resumen de las categorías de vehículos

5.1.1.1.2 Definiciones

Antes de explicar un poco el procedimiento de ensayo hay que tener claros algunos conceptos previos.

“Punto de referencia; se refiere a uno de los siguientes puntos:

- Para vehículos con el motor delante, la parte delantera del vehículo.
- Para el resto de vehículos, el borde del motor más cercano a la parte delantera del vehículo.

"Potencia del motor neta, S":

Significa la velocidad declarada del motor en min^{-1} (rpm) en la que el motor desarrolla su potencia neta máxima nominal.

“ m_{ro} ” Masa del vehículo en orden de marcha:

La masa del vehículo, con su(s) depósito(s) de combustible llenado al menos el 90 por ciento de su capacidad o sus capacidades, incluida la masa del conductor, del combustible y líquidos, equipado con el equipo estándar conforme las especificaciones del fabricante y, en su caso, la masa de la carrocería, la cabina, el acoplamiento y la rueda(s) de repuesto así como las herramientas.

“n_{target}” Velocidad del motor objetivo:

La velocidad del motor a cumplir.

Resultado intermedio:

El resultado intermedio es el valor más alto de los dos promedios matemáticamente redondeados al primer decimal

5.1.1.1.1.3 Procedimiento de ensayo:

5.1.1.1.1.3.1 El cálculo teórico:

Para determinar la marcha que será significativa para el ensayo, hay que realizar un cálculo teórico previo. Se usarán las condiciones determinadas en el reglamento para cada categoría de vehículo, que implican la velocidad del vehículo y la velocidad del motor para cada marcha.

Las condiciones de ensayo vienen acotadas de la siguiente forma:

- La línea central del vehículo deberá seguir la **línea CC'**.
- Cuando el punto de referencia alcanza la **línea AA'** el acelerador debe estar apretado al máximo, hasta que el punto de referencia alcance la **línea BB'+5m**.
- En el momento que el punto de referencia alcanza la **línea BB'**:
 - La velocidad del vehículo deberá ser **35 km/h ± 5 km/h**.
 - Para **M2 MTMA>3500 kg, N2**:
 - La velocidad del motor deberá ser **[70-74] %** de S.
 - Para **M3, N3**:
 - La velocidad del motor deberá ser **[85-89] %** de S.

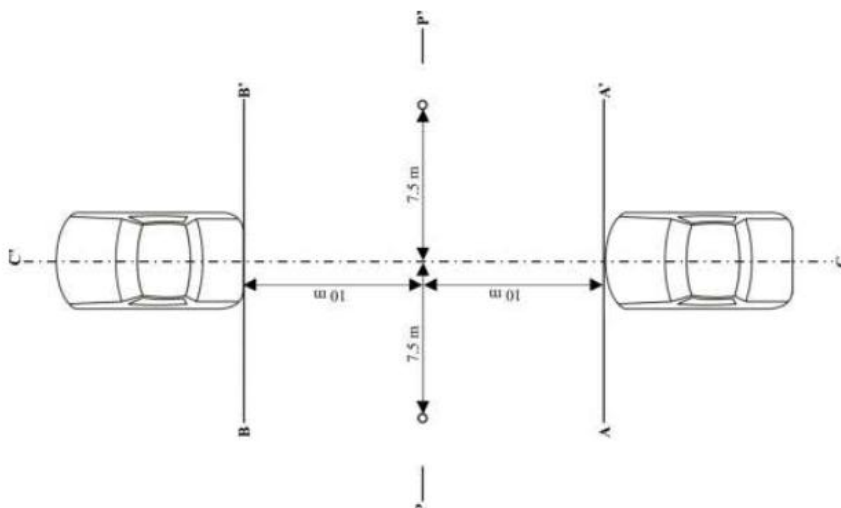


Ilustración 35 Determinación de las líneas de referencia de la pista

El cálculo teórico de la velocidad se realiza primero fijando el valor de la velocidad del motor y aplicando la siguiente fórmula.

$$v = \frac{60 * RPM * \varnothing}{RelGeari * RelDifferential * 1000} \left(\frac{km}{h} \right)$$

El diámetro exterior de las ruedas, la relación de transmisión de cada marcha y del puente trasero son datos necesarios facilitados por el fabricante.

5.1.1.1.1.3.2 El ensayo en la pista de pruebas:

El ensayo debe realizarse con el/los engranaje/s determinados anteriormente.

Se tomarán medidas de la velocidad del vehículo cada vez que pase la línea BB', esta velocidad se ajustará hasta que cumpla la condición 35 km/h \pm 5 km/h.

El control del acelerador debe estar completamente apretado desde AA'. Esto significa que el ingeniero de pruebas necesita comunicarse con el conductor, hasta que se encuentre la velocidad de entrada en AA' para alcanzar la velocidad requerida en BB' + 5 m.

Una vez que se ha determinado la velocidad de entrada en AA', la prueba debe realizarse en cada velocidad determinada antes.

Cuatro medidas de cada lado deben tomarse, las medidas sólo son válidas si la diferencia entre medidas consecutivas es inferior a 2 dB (A).

5.1.1.1.1.4 Mediciones:

El máximo nivel sonoro A-ponderado dB(A) medido durante el paso del punto de referencia del vehículo entre las líneas AA' y BB' + 5 m, deberá redondearse al primer dígito significativo después del punto decimal (por ejemplo, XX, X).

5.1.1.1.1.5 Resultado final (L_{urban}):

L_{urban} = Resultado intermedio

5.1.1.1.1.6 Applus+ IDIADA:

Applus+ IDIADA es una empresa multinacional que proporciona servicios de diseño, ingeniería, pruebas y homologación para la industria del automóvil. Cuenta con más de 2.000 empleados en 23 países.

La empresa fue fundada por primera vez en 1971 como IDIADA o 'Institut d'Investigació Aplicada de l'Automòbil' de la Universidad Politécnica de Cataluña. En 1990, IDIADA fue separado de la universidad y se estableció como empresa independiente propiedad de la Generalidad de Cataluña. Fue privatizada en 1999 como una empresa de propiedad del 80% por Applus+ y el 20% por la Generalidad de Cataluña.

Applus + IDIADA cuenta con una pista de pruebas en España, que es considerado como una de los mejores del mundo. La compañía proporciona servicios de ingeniería para la seguridad activa y pasiva, sistemas de propulsión, material eléctrico y electrónico del vehículo, Ruido, vibraciones y asperezas (NVH), vehículo con aire acondicionado, durabilidad y fiabilidad.

También proporciona servicios de homologación y certificación en todo el mundo para una amplia variedad de vehículos, incluyendo motocicletas, camiones y autobuses y componentes incluidos cascos y viseras.



Ilustración 36 Imagen desde el aire de IDIADA en l'Albornar

Entre todas las pistas que tiene Applus+ IDIADA, se encuentran las pistas de ruido ISO-1 e ISO-2. Estas pistas siguen la normativa ISO 10844 requerida en el Reglamento No. 51 de ruido.



Ilustración 37 Pistas de ruido en Applus+ IDIADA



Ilustración 38 Ejemplo de un ensayo de ruido en la pista ISO-2

6. APLICACIÓN DEL ESP8266 EN EL METODO DE ENSAYO:

6.1. VBOX:

El galardonado VBOX es un potente instrumento utilizado para la medición de la velocidad y posición de un vehículo en movimiento. El mismo se basa en una nueva generación de receptores de satélites de última generación midiendo velocidad, distancia, aceleración, tiempo de vuelta, posición y mucho más.

VBOX está reconocido internacionalmente como un patrón de calidad en la medida de velocidad y distancia para frenado, prestaciones y ADAS testing. Casi todos los fabricantes de vehículos y neumáticos en el mundo ahora tienen un VBOX en su Departamento de Ensayos.

Debido principalmente a su pequeño tamaño y sencillo procedimiento de instalación, el VBOX está idealmente indicado para ser utilizado en coches, bicicletas y vehículos todoterrenos.



Ilustración 39 Repertorio variado de VBOX

En el área de homologaciones de Vehículo Industrial donde tienen lugar los ensayos de ruido de vehículos pesados, se dispone de un VBOX 3i para dichos ensayos.



Ilustración 40 VBOX 3i utilizado en VI

6.2. Situación actual:

El VBOX 3i se monta sobre el propio vehículo a ensayar, al que se le conectan la señal de Trigger, varios Displays para que el conductor pueda ver la velocidad del vehículo, y además se conecta el ordenador para poder ver la velocidad de las pasadas a tiempo real.

El Trigger (cuando el vehículo pasa por el punto de salida) mediante un sensor óptico envía una señal al VBOX para que empiece a medir la velocidad, en ese momento se muestra en el ordenador la distancia recorrida y la velocidad a la que va en vehículo. Un ingeniero desde dentro del vehículo ha de estar monitorizando estos datos, ya que se ha de determinar si la pasada se ha hecho de forma correcta y cumple las condiciones objetivo del ensayo en el punto de salida (determinado por la distancia). Finalmente se comunica desde el interior del vehículo a un segundo ingeniero que está fuera en la pista de ensayos midiendo el nivel sonoro, si la pasada ha sido válida y si se debe anotar el valor de ruido.

Actualmente el montaje del VBOX 3i para en ensayo de ruido que se utiliza es el siguiente:



Ilustración 41 Montaje VBOX ensayo 51R03

6.3. Problema:

El problema actual es claramente la necesidad de la comunicación entre los dos ingenieros que realizan el ensayo, esta se lleva a cabo mediante comunicación verbal a través de un dispositivo móvil.

También hay que tener mucho cuidado a la hora de tomar las mediciones de ruido, de no hablar y de no escribir valores que no son los correctos debido a picos de sonido indeseados.

Además, hay otro factor que puede afectar al error de las medidas, y es que el ingeniero no pueda leer bien los datos del VBOX en el ordenador, sumándole la dificultad de leer entre muchas filas de datos el valor exacto de forma rápida.

6.4. Solución

Cuando pude presencias el primer ensayo de ruido pude identificar claramente el problema, así que me puse a investigar más sobre el VBOX 3i que se estaba utilizando y todo el potencial que no se usaba.

El VBOX 3i tiene además de las conexiones que ya se utilizan como se ha mostrado en la Ilustración 35, otras más que no se utilizan y pueden ser de mucha utilidad:

- 2 x Salidas digitales: Dos salidas digitales están disponibles en VBOX 3i. Una salida digital se asigna a velocidad o distancia (configurable a través de impulsos por metro). Mientras que la segunda es una salida de conmutación de nivel que permite a los usuarios seleccionar cualquiera de los canales registrados y asignarle un valor de umbral.
- 2 x Salidas analógicas: Dos salidas analógicas de 16 bits para la velocidad de salida (u otros parámetros GPS) para su uso por equipos de registro de datos adicionales. El rango de salida de voltaje es de 0 a 5v DC con una resolución de 76 μ V por bit.

Se trata de utilizar una salida analógica configurada en un rango de velocidades (ej. De 0 a 60 km/h) y conectarla a un divisor de corriente con una R2 de 10k y una R1 de 40k. La salida del divisor de corriente irá conectada al pin ADC A0 del ESP8266, donde se monitorizará su valor cada 500 ms.

Como el valor que interesa es el del momento de salida del vehículo por el punto de referencia establecido, se dará la orden al VBOX para que cuando la distancia recorrida por

el vehículo llegue al valor umbral, la salida digital envíe un pulso a uno de los pines disponibles en el ESP8266 (ej. GPIO4). En ese mismo instante el dispositivo comparará el valor de la velocidad del vehículo con el valor de la velocidad objetivo establecida en el ensayo, mostrando en la pantalla un OK o FAIL en función de si la pasada es válida o no.

Todo esto lo estará viendo el ingeniero en pistas midiendo el ruido a través de un dispositivo móvil, mientras está conectado al ESP8266 como se explicará en el punto 6.4.2.

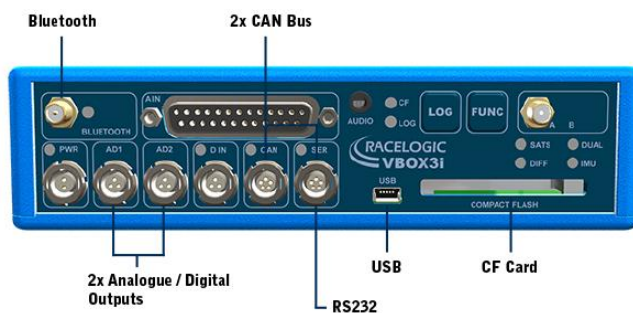


Ilustración 42 Muestra de las conexiones de VBOX 3i

6.4.1. Acceso:

El acceso al programa es muy simple ya que el ESP8266 operará como punto de acceso, únicamente hace falta un dispositivo WiFi y navegador de internet. Hay que buscar el dispositivo como si se fuera a conectar a cualquier red WiFi, será llamado CUGEsp8266. Finalmente hay que introducir la contraseña de autenticación para poder tener acceso.

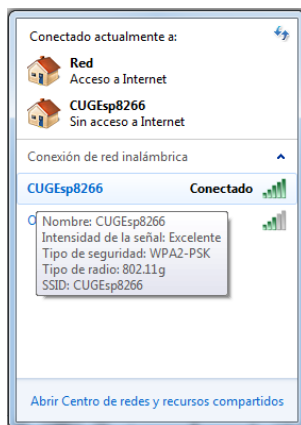


Ilustración 43 Ejemplo de conexión al ESP8266

Posteriormente se debe abrir cualquier navegador e introducir en la barra del navegador 192.168.4.1 como se muestra a continuación en la ilustración.

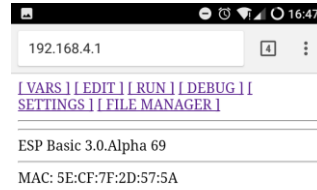


Ilustración 44 Ejemplo de acceso al ESP8266

6.4.2. Interfaz y uso:

El programa tiene dos objetivos, el primero es hacer el cálculo teórico explicado anteriormente para determinar las marchas a ensayar, y el segundo es poder mostrar a tiempo real y desde cualquier dispositivo si la pasada del vehículo es o no válida.

Si introducimos la dirección 192.168.4.1/run en el navegador, accederemos directamente al programa donde nos encontraremos una portada similar a esta.



Ilustración 45 Página principal

Si procedemos a través de la pantalla mediante el botón Start, llegaremos a la siguiente página donde nos pide datos sobre el vehículo a ensayar.

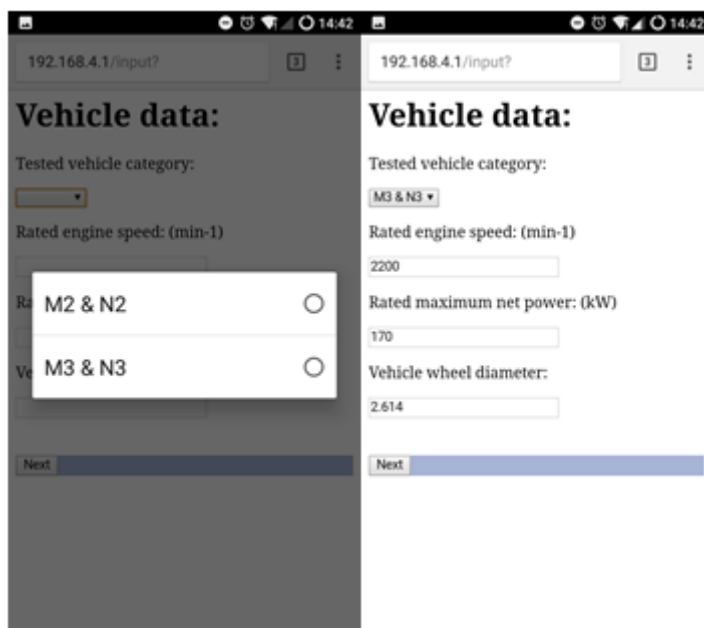


Ilustración 46 Página de datos del vehículo

Si seguimos procediendo a través de la pantalla nos seguirá pidiendo más datos del vehículo, en este caso las relaciones de cada marcha.

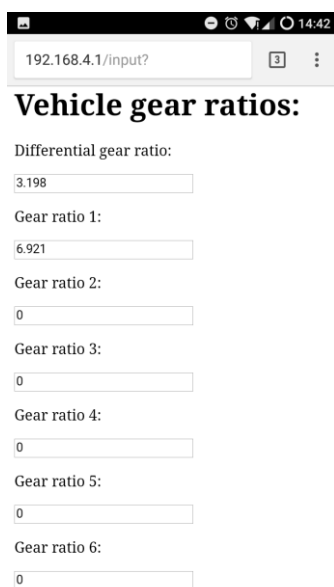


Ilustración 47 Página de datos sobre las marchas del vehículo

Todos estos datos servirán al programa para decidir que marcha tenemos que utilizar a la hora de hacer el ensayo, los lectores podrán utilizar el programa con todas sus características el día de la presentación desde cualquier dispositivo WiFi con navegador.

6.4.3. Código:

Debido a que el código está subido en el propio ESP8266, se enseñarán algunos fragmentos de código significativos.

El fragmento de código de la página principal es muy simple, y lo podemos ver a continuación.

```
memclear

[HOME]
cls
wprint |<html>|
wprint |<head>|
wprint |<title>Regulation 51.03</title>|
wprint |</head>|
wprint |<body>|
wprint |<div align="left">|
wprint ||
wprint |</div>|
wprint |<div align="center">|
wprint |<h1>Regulation 51</h1>|
wprint |<h2>03 Series of Amendments</h2>|
wprint |<h3>Uniform provisions concerning the approval of motor vehicles having at least four wheels with regard to their sound emissions</h3>|
wprint ||
wprint |<h2>UNITED NATIONS</h2>|
wprint |<div style="background-color: rgb(167, 180, 214);">|
button "Start", [DEF1]
wprint |</div>|
wprint |</div>|
wprint |</body>|
wprint |</html>|
wait
```

Ilustración 48 Fragmento de código de página principal

En cuanto a la primera página de datos incrementa un poco la complejidad como vemos a continuación.

```
[DEF1]
memclear
cls
wprint |<html>|
wprint |<head>|
wprint |<title>Regulation 51.03</title>|
wprint |</head>|
wprint |<body>|
wprint |<h1> Vehicle data:</h1>|
wprint |<div align="left">|
wprint |<p> Tested vehicle category:</p>|
dropdown category$, "M2 & N2, M3 & N3"
wprint |<p>Rated engine speed: (min-1)</p>|
textbox engspeed
wprint |<p>Rated maximum net power: (kW)</p>|
textbox netpwr
wprint |<p>Vehicle wheel diameter:</p>|
dropdown dim, "235/65 R 16, 235/75 R 17.5, 245/70 R 19.5, 245/75 R 17.5, 255/70 R 22.5, 265/70 R 19.5, 275/70 R 22.5, 275/80 R 22.5"
if dim="235/65 R 16" then
    diameter = 2.236
elseif dim="235/75 R 17.5" then
    diameter = 2.431
elseif dim="245/70 R 19.5" then
    diameter = 2.559
elseif dim="245/75 R 17.5" then
    diameter = 2.406
elseif dim="255/70 R 22.5" then
    diameter = 2.837
elseif dim="265/70 R 19.5" then
    diameter = 2.644
elseif dim="275/70 R 22.5" then
    diameter = 2.922
else
    diameter = 3.087
end if
wprint |<br><br><br>|
wprint |<div style="background-color: rgb(167, 180, 214);">|
button "Next", [DEF2]
wprint |</div>|
wprint |</div>|
wprint |</body>|
wprint |</html>|
wait
```

Ilustración 49 Fragmento de código de la primera página de datos

Finalmente se mostrará el código de la primera página en la que se determina si se ensaya o no dicha marcha.

```
[DEF3]
cls
targetenginelow = 0
targetenginehigh = 1
if category$ = "M2 & N2" then
targetenginelow = engspeed*0.70
targetenginehigh = engspeed*0.74
else
targetenginelow = engspeed*0.85
targetenginehigh = engspeed*0.89
end if
wprint |<html>|
wprint |<head>|
wprint |<title>Regulation 51.03</title>|
wprint |</head>|
wprint |<body>|
wprint |<h1>Gears that can fulfill the target vehicle speed:</h1>|
wprint |<h1>At target engine speed, the vehicle shall be at: (km/h)</h1>|
wprint |<p>Gear ratio 1:</p>|
textbox 60*targetenginelow*diameter/(gear1*difratio*1000)
textbox 60*targetenginehigh*diameter/(gear1*difratio*1000)
if (60*targetenginelow*diameter/(gear1*difratio*1000))>30 or (60*targetenginehigh*diameter/(gear1*difratio*1000))<40 then
wprint "TEST"
else
wprint "NO TEST"
end if
wprint |<p>Gear ratio 2:</p>|
textbox 60*targetenginelow*diameter/(gear2*difratio*1000)
textbox 60*targetenginehigh*diameter/(gear2*difratio*1000)
if (60*targetenginelow*diameter/(gear2*difratio*1000))>30 or (60*targetenginehigh*diameter/(gear2*difratio*1000))<40 then
wprint "TEST"
else
wprint "NO TEST"
end if
wprint |<p>Gear ratio 3:</p>|
textbox 60*targetenginelow*diameter/(gear3*difratio*1000)
textbox 60*targetenginehigh*diameter/(gear3*difratio*1000)
if (60*targetenginelow*diameter/(gear3*difratio*1000))>30 or (60*targetenginehigh*diameter/(gear3*difratio*1000))<40 then
wprint "TEST"
else
wprint "NO TEST"
end if
wprint |<p>Gear ratio 4:</p>|
textbox 60*targetenginelow*diameter/(gear4*difratio*1000)
textbox 60*targetenginehigh*diameter/(gear4*difratio*1000)
if (60*targetenginelow*diameter/(gear4*difratio*1000))>30 or (60*targetenginehigh*diameter/(gear4*difratio*1000))<40 then
wprint "TEST"
else
wprint "NO TEST"
end if
wprint |<p>Gear ratio 5:</p>|
```

Ilustración 50 Fragmento de página de resultados

Conclusiones

El presente proyecto define, programa y justifica (siempre en cumplimiento con la normativa vigente) la implementación de una mejora en el proceso de homologación de un sistema de ruido para categorías de vehículos pesados, utilizando un ESP8266. No se llega a demostrar si se disminuyen los tiempos de ensayo y sus errores.

Además, se estudia en profundidad lo que es un ESP8266 conociendo sus características, las distintas formas de utilizarlo y programarlo, todas sus versiones y en que difieren todas y cada una de ellas. Que medios de comunicación con el ESP8266 tenemos a nuestro alcance y las diversas plataformas que podemos utilizar, desde las más conocidas a algunas que eran completamente desconocidas.

Se estudian los reglamentos que actualmente aplican al mundo de la automoción y se estudia en profundidad el Reglamento No.51. También se realiza un procedimiento de ensayo de la nueva enmienda de ruido en el que se han utilizado fragmentos para este trabajo. Dicho procedimiento ha acabado en el departamento de calidad pendiente de ser incorporado como procedimiento de IDIADA vehículo industrial, además de una propuesta de nuevas hojas de ensayo.

Se conocen en profundidad diversos dispositivos electrónicos utilizados en los ensayos entre ellos el VBOX 3,i como utilizarlos y su posible potencial no utilizado para la disminución de errores.

Finalmente, concluir que el coste total del proyecto no superaría los 20 € si no contamos todo el equipo del que ya se dispone en Applus+ IDIADA, el coste de un ESP8266 no alcanza los 5 €, sumados a las placas de pruebas y resistencias más cableado no superan la cantidad descrita.

Agradecimientos

En primer lugar, me gustaría dar las gracias al director del presente proyecto, el profesor Manuel Moreno Eguílaz, por prestarme su idea y las primeras herramientas para emprender este bonito viaje que ya culmina. Además, por tener paciencia conmigo debido a mis ausencias por mi trabajo, siempre es un placer enseñar el trabajo a alguien que pueda apreciarlo, espero que te guste.

Seguidamente, agradecer el constante apoyo de mi familia y amigos, sin el cual muchas veces no hubiera podido seguir adelante. Gracias por facilitarme tanto las cosas.

Una mención especial se merece mi pareja, Eva. Ella mejor que nadie sabe lo que ha significado esta carrera y la realización del presente proyecto, llenos ambos de altibajos constantes. Por ello, toda palabra es poca para agradecerle su apoyo, ayuda y comprensión durante todos estos años, de principio a fin. Al fin lo conseguimos.

Finalmente, quiero agradecer a todo el personal de Applus+ IDIADA con el que he tenido contacto y me han ayudado a resolver mis dudas relativas al proceso de homologación, al reglamento que concierne este proyecto y a la instrumentación de los vehículos para los ensayos. En especial mención Manuel Lara, que ha sido para mí un pilar de confianza para resolver dudas, es para mí una motivación como objetivo de rendimiento, eficacia y profesionalidad.

Bibliografía

Referencias bibliográficas

- [1] UNITED NATIONS, Regulation No.51; Uniform provisions concerning the approval of motor vehicles having at least four wheels with regard to their sound emissions, Geneva: 2016.
- [2] Marco Schwartz, Internet of Things with ESP8266, 2016
- [3] Neil Kolban. *Kolban's Book on ESP8266, 2016*
- [4] Claus Kuhnel, Building an IoT Node for less than 15 \$: NodeMCU & ESP8266, 2015