



Universidad Tecnológica Nacional
Facultad Regional Villa María
Ingeniería Electrónica

Horómetro WiFi Programable

Autor/es:
Devalis Gino Nahuel

Tutor: Ing. José Luis Catalano

Director: Ing. Fabián M. Sensini

Co-Director: Esp. Ing. Héctor D. Ferrari

Fecha de Defensa: 28/04/2023





Dedicatorias

A mis seres queridos presentes y no presentes que me guiaron y aun me guían, quienes me apoyan en cada decisión que he decido tomar y que nunca me permitieron rendirme ante las adversidades.

A mí por el esfuerzo, la dedicación, los sacrificios y la responsabilidad, por seguir adelante ante la duda y nunca abandonar mis objetivos.



Agradecimientos

A mi familia que con su ayuda y amor incondicional me acompañaron durante este hermoso trayecto, quienes con su esfuerzo me permitieron tener esta posibilidad y llegar a donde estoy.

A la Universidad Tecnológica Nacional Facultad Regional de Villa María por brindar los espacios, herramientas y soportes necesarios para realizar la carrera de una manera profesional, cómoda y accesible.

A los profesores que durante el transcurso de la carrera brindaron los conocimientos necesarios con la mejor de las actitudes. Quienes siempre estuvieron a disposición, para aquellos que estuvieron dispuestos a aprender y crecer de manera mutua en la relación profesor estudiante.

A los amigos que estuvieron ahí para sacar una sonrisa en los momentos más adversos, que ayudaron y brindaron sus conocimientos para lograr un mejor resultado.



Resumen

En el marco del presente trabajo se realizó un horómetro wifi programable. Dicho dispositivo realiza la tarea de contar las horas de funcionamiento de un equipo o maquina en la industria, enviar los datos mediante wifi a una aplicación y desde la aplicación programar mantenimientos cada determinado periodo de tiempo.

La idea de este proyecto surge luego de realizar las practicas supervisadas con tareas desempeñadas en el área de mantenimiento de una industria metalmecánica, ya que al encargado de mantenimiento no podía realizar un seguimiento del tiempo de funcionamiento de las máquinas y esto desembocaba en un problema a la hora de planificar tareas a futuro.

Para saber si una maquina está funcionando se utiliza un sensor de corriente no invasivo, es decir, un sensor que se coloca alrededor del cable de alimentación. De esta forma si el equipo comienza a funcionar el sensor detecta la circulación de corriente por el cable y genera una señal que es adquirida por el microcontrolador dando aviso al sistema que debe contar el tiempo de funcionamiento.

En la aplicación del teléfono se puede observar el nombre del equipo, la cantidad de horas de funcionamiento y los mantenimientos programados con anterioridad. Cuando las horas de funcionamiento coinciden con las horas de un mantenimiento programado se da aviso al usuario mediante una alarma en el dispositivo y mediante un aviso en la aplicación.

Las horas de funcionamiento no solo pueden verse en la aplicación, sino también en el dispositivo ya que este cuenta con una pantalla LCD de 20x4. En dicha pantalla además se da aviso de las alarmas y de si el dispositivo está funcionando.

Abstract

Within the framework of the present work, it was intended to design and carry out a programmable Wi-Fi hour meter. This device performs the task of counting the hours of operation of a piece of equipment or machine in the industry, sending the data via Wi-Fi to an application and from the application scheduling maintenance every certain period.

The idea of this project arises after carrying out the supervised practices with tasks carried out in the maintenance area of a metal-mechanic industry, since the maintenance manager could not keep track of the operating time of the machines and this led to a problem at when planning future tasks.

To know if a machine is working, a non-invasive current sensor is used, that is, a sensor that is placed around the power cable. In this way, if the equipment starts to work, the sensor detects the flow of current through the cable and generates a signal that is acquired by the microcontroller, giving notice to the system that it must count the operating time.

In the phone application, you can see the name of the equipment, the number of hours of operation and previously scheduled maintenance. When the hours of operation coincide with the hours of a scheduled maintenance, the user is notified by means of an alarm on the device and by means of a notice in the application.

The hours of operation cannot only be seen in the app, but also on the device as it has a 20x4 LCD screen. This screen also gives notice of alarms and if the device is working.



ÍNDICE

Título	Página
Dedicatorias	3
Agradecimientos	4
Resumen	5
Abstract	5
Introducción.....	8
Análisis del problema.....	8
Análisis de sistemas existentes	8
Descripción de las actividades del proyecto.....	8
Objetivos	10
Objetivos generales	10
Objetivos particulares.....	10
Diseño del Proyecto	10
Revisión de requerimientos y parámetros de operación.....	10
Selección de componentes y dispositivos	10
PIC18F4550.....	10
ESP-12E	11
STC013 5A.....	14
RTC DS3231	16
Memoria EEPROM AT24C08	18
LCD 20x4	19
Level Shifter (Cambiador de nivel)	20
Diagrama en bloques del dispositivo	23
Descripción de cada una de las partes.....	23
Detección de funcionamiento	23
Conteo de tiempo	24
Comunicación serial entre PIC18F4550 y ESP-12E	26
Comunicación WiFi mediante protocolo MQTT (MQ Telemetry Transport).....	28
Protocolos de comunicación en IoT.....	28
Comunicación entre ESP8266 y APK	31
Diagrama de flujo	34
Evaluación Final del Sistema	35
Preparación del prototipo con materiales finales.....	35



Montaje y ensayo real del prototipo.....	39
Presentación final del prototipo	43
Conclusiones.....	47
Anexo I.....	48
Anexo II.....	49
Anexo III	50
Anexo IV	51
Bibliografía.....	52



Introducción

Análisis del problema

Un Horómetro es indispensable a nivel industrial, ya que es la base para realizar los cálculos de vida útil de una maquinaria, y para controlar los mantenimientos y reparaciones. El Horómetro a nivel industrial, cumple el mismo objetivo que el odómetro del coche, controla cantidad de uso, y los tiempos de intervención técnica

Como se mencionó anteriormente el resumen, la idea para realizar este proyecto surge de la necesidad de un seguimiento continuo, preciso y de bajo costo del tiempo de funcionamiento de las maquinas en la industria.

Esta información permite al encargado de mantenimiento planificar tareas a futuro, realizar indicadores de fallas y además permite realizar indicadores de productividad entre otras tantas ventajas que puede generar poseer esta información.

Se trata de un dispositivo útil, pequeño, fácil de manejar, fácil de instalar y no es invasivo, es decir, que no se debe modificar ningún equipo para utilizarlo.

Centrándose más en el campo de mantenimiento, no tener un registro de funcionamiento puede ocasionar una gran cantidad de problemas como averías, reducción de vida útil, encarecimiento de las reparaciones, perdidas en el nivel de producción, detener por completo un proceso productivo o hasta la pérdida total del equipo. Por todo lo mencionado es un dispositivo con buenas proyecciones y utilidades.

Análisis de sistemas existentes

Actualmente no hay registro de un dispositivo similar al que se realizó en este proyecto. Lo sistemas existentes que realizan esta tarea actualmente son horómetros convencionales, software que están embebidos directamente en los equipos más modernos o se realizan a través del sistema SCADA (Supervisory Control And Data Acquisition) que es un concepto que se emplea para realizar un software para ordenadores que permite controlar y supervisar procesos industriales a distancia. Facilita retroalimentación en tiempo real con los dispositivos de campo, y controla el proceso automáticamente. Por desgracia el sistema mencionado con anterioridad no es económicamente accesible para cualquier industria y ni es económicamente factible realizarlo para pocos equipos.

Dentro de los horómetros convencionales podemos encontrar una gran variedad de dispositivos como los que se deben conectar a un circuito de control, los que funcionan a partir de la vibración de los equipos, los de alimentación continua o alterna y los que poseen batería entre otras combinaciones posibles.

Es por ello que el horómetro WiFi es un dispositivo completamente innovador y de fácil acceso para cualquier clase de industria sin importar tamaño, capacidad y cantidad de equipos.

Descripción de las actividades del proyecto

En primera instancia se estudian los requerimientos mínimos que debía tener el proyecto tanto en investigación, desarrollo y fabricación para asegurar la factibilidad del mismo.

Luego de realizar una estimación de los tiempos, se realiza un listado de los insumos necesarios para desarrollar tal proyecto.

Los insumos necesarios son:



Materiales Para Fabricación		
IT	Elemento	Especificación
1	Microcontrolador	PIC18F4550
2	Modulo WiFi	ESP8266 12E
3	Módulo de tiempo	RTC Ds3231
4	Sensor de corriente	Sct013
5	MOSFET Canal N	IRFZ44N
6	Regulador 3,3V	LD1117
7	BUZZER 5V	
8	LED's	5V
9	Conversor USB a TTL	PI2303hx
10	LCD 20x4	020N004L
11	Resistencias	Varias
12	Capacitores electrolíticos	Varios
13	Capacitores cerámicos	Varios
14	Pulsadores sin retención	
15	PCB Simple faz	FR4
16	Cargador de teléfono	5V
17	Memoria EEPROM	AT24c08
18	Quemador de PIC	Pick it 3

Tabla 1: Insumos necesarios para el horómetro.

Una vez definido los controladores y sensores a utilizar, se procede a investigar su funcionamiento y modo de uso. De acuerdo a las hojas de datos y a la información reunida se programa el microcontrolador y se realizan las conexiones necesarias para el correcto funcionamiento del sistema.

Las pruebas se realizan en placa experimentora (Protobaord) y simulación. Cuando el sistema funciona correctamente se continua con el diseño y realización de la PCB. Por último, se hace la carcasa del producto y las pruebas de campo.



Objetivos

Objetivos generales

Los objetivos generales de este dispositivo es brindar información real y actualizada, asistir al personal de mantenimiento en la organización y planificación de tareas, realizar un producto relativamente de bajo costo con gran versatilidad y de simple utilización.

Con los puntos mencionados anteriormente sería simple asistir a una gran cantidad de industrias en su organización y actualización de sistemas de seguimiento. Se pretende lograr este objetivo ya que la actualidad de las empresas y PyMEs de Argentina no es la mejor económicamente hablando, produciendo un declive o estancamiento en la actualización de sus sistemas de mantenimiento.

La intención de dicho proyecto es también reducir el impacto ambiental que genera el gran consumo de hojas de papel para realizar un seguimiento u observar las tareas de mantenimiento de un equipo.

Objetivos particulares

Los objetivos particulares son en un futuro seguir mejorando las falencias que se van encontrando en el proceso. Realizar una página web que permita a la industria crear un usuario y administrar todos los equipos de la planta desde una sola aplicación. Crear una base de datos y administrar toda la información obtenida para luego realizar gráficos y reportes periódicos de consumos, tiempos de operación, tiempos de parada, tiempo entre falla y otros datos más que son muy utilizados para realizar indicadores y seguimiento de equipos.

Realizar un diseño más compacto y con dispositivos más sofisticados para un uso profesional y preciso.

Por ultimo como un objetivo mayor se quiere realizar el producto en masa para alcanzar la mayor cantidad de industrias posible y tal vez un nuevo modelo de administración del área de mantenimiento de plantas.

Diseño del Proyecto

Revisión de requerimientos y parámetros de operación

Como se ha mencionado anteriormente se quiere lograr un dispositivo versátil, no invasivo y de bajo costo, bajo estas condiciones y la disponibilidad del mercado se buscaron los componentes más adecuados.

Para no ser un dispositivo invasivo se utiliza un sensor de corriente de efecto inductivo que se coloca alrededor de los cables. Para mostrar información general al operario de la maquina se utiliza una pantalla LCD, LED's y un buzzer.

Para compartir la información con el usuario de manera remota se optó por la tecnología WiFi, que permite interactuar entre la aplicación del teléfono celular y el dispositivo instalado en la máquina.

Selección de componentes y dispositivos

PIC18F4550

El componente principal del proyecto es el microcontrolador PIC18F45450. Fue elegido debido a su disponibilidad, costo en el mercado, prestaciones y por ultimo a que se ha utilizado a lo largo de la carrera en distintos proyectos, lo cual permite conocer en gran



medida sus prestaciones, versatilidad y limitaciones.

Las principales características de este microcontrolador son:

- CPU: 8-bits, velocidad: 12 MIPS
- Memoria de programa: Tipo Flash, 32 Kbytes (10000 ciclos de escritura/borrado)
- RAM bytes: 2 KBytes
- EEPROM: 256 Bytes
- Número total de pines: 40
- Voltaje de operación: +4.0V a +5.5V
- Pines de E/S programables: 35
- Interfaces de comunicación: USB Serial Interface, Master/Slave SPI Serial Interface, Programable Serial UART, Dos-cables Serial Interface.
- Módulos ADC: 13 Canales de 10 bits de resolución
- Módulos de Timer: un módulo de 8 bits, tres módulos de 16 bits.
- Canales PWM: 4
- Oscilador externo: Hasta 48 MHz
- Oscilador inter: 32 KHz – 8 MHz

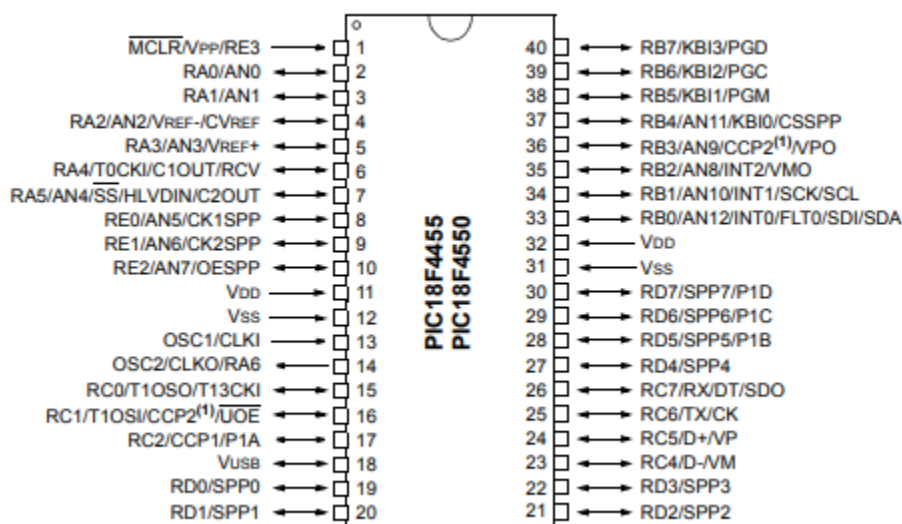


Fig. N° 1 Pinout PIC18F45450.

ESP-12E

Este dispositivo fue elegido por la disponibilidad de sus pines, por sus prestaciones, su previo conocimiento y además por la cantidad de información que se puede encontrar de ella.

Este módulo posee embebido un MCU Tensilica L106 de 32—bits y se caracteriza por su bajo nivel de consumo de energía. Estas prestaciones le permiten a la ESP comunicarse de manera serial con el microcontrolador PIC, por I2C con una memoria EEPROM externa y a través del protocolo MQTT con la App.

Principales características:

- Tensión de alimentación: 3.3V
- Protocolos soportados: 802.11 b/g/n
- Soporte de red: 2,4 GHz

- Wi-Fi Direct (P2p), Soft Access Point
- Stack TCP/IP integrado
- PLL, reguladores y unidades de manejo de energía integrados
- Procesador integrado de 32 bits. Tensilica L106
- RAM de instrucción de 64 KB, RAM de datos de 96 KB.
- Comunicación tipo de interfaz: SPI, Serial, UART (115200 bps)
- Pines de propósito general (GPIO): 16 pines de entradas/salidas general
- Una conversor ADC de 10-bits de resolución

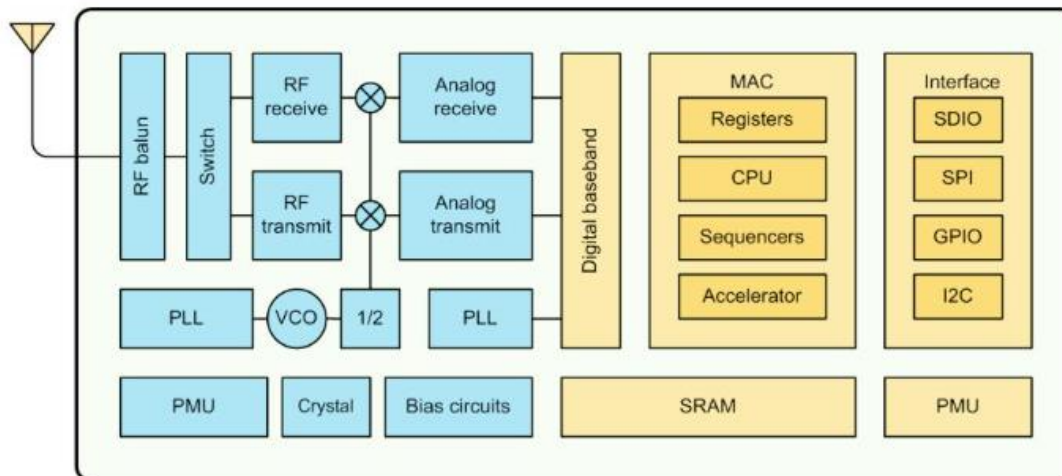


Fig. N° 2 ESP8266EX Diagrama en bloque.



Fig. N° 3 Imagen real de ESP-12E.

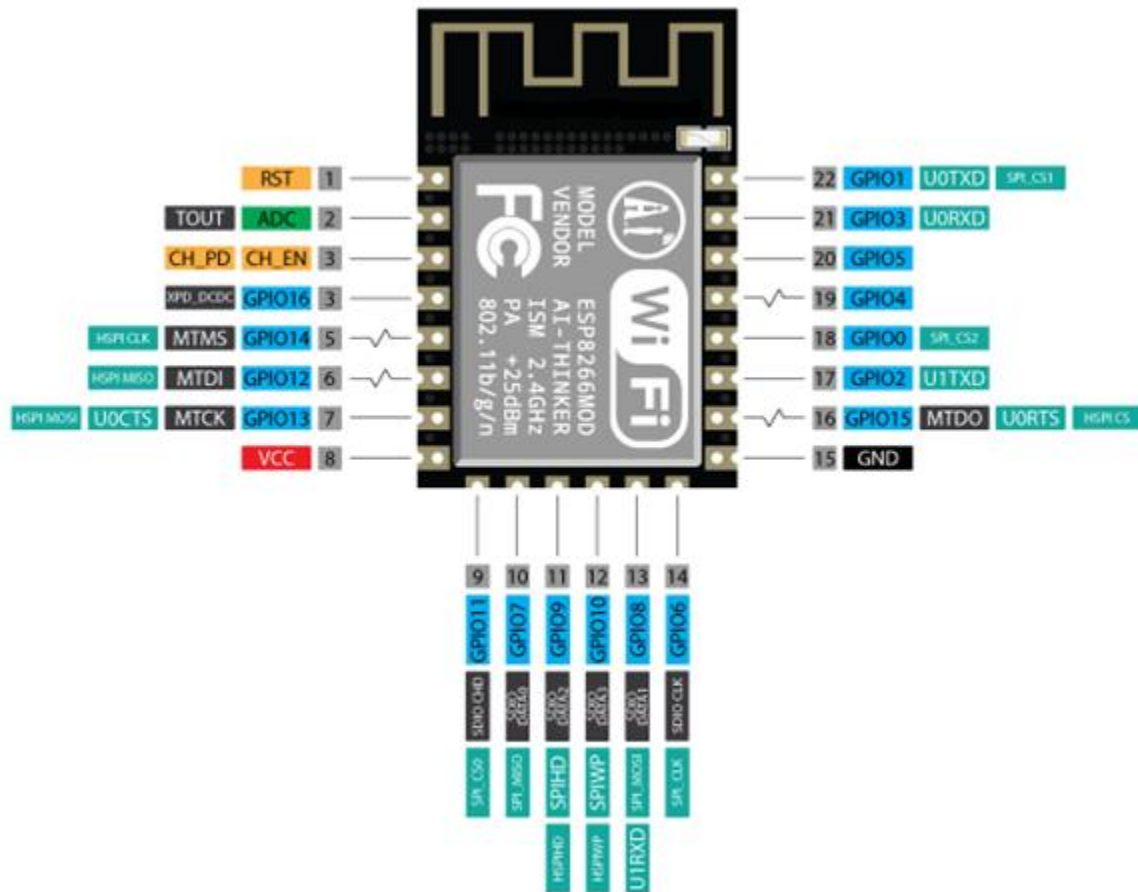


Fig. N° 4 Pinout ESP-12E.

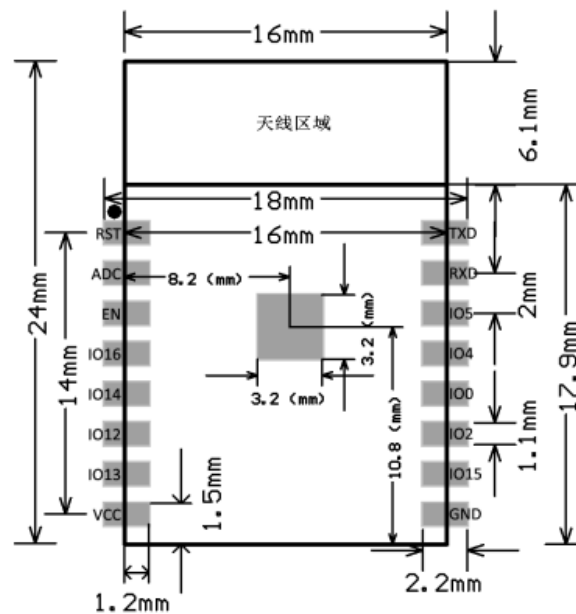


Fig. N° 5 Dimensiones ESP-12E.

STC013 5A

Para entender con más detalle acerca de este dispositivo se explicará de manera resumida que son y cómo funcionan los transformadores.

El transformador eléctrico es una máquina estática de corriente alterna que permite variar alguna función de la corriente como el voltaje o la intensidad, manteniendo la frecuencia y la potencia, en el caso de un transformador ideal. Para lograrlo, transforma la electricidad que le llega al devanado de entrada en magnetismo para volver a transformarla en electricidad, en las condiciones deseadas, en el devanado secundario.

Los transformadores están compuestos por diferentes elementos entre los que destacan como principales el núcleo y los devanadores.

El núcleo del transformador puede ser de una gran variedad de materiales conductores de magnetismo cuya función es evitar la pérdida de flujo magnético. Por su parte el devanado es un hilo de cobre enrollado a través del núcleo en uno de sus extremos y recubierto por una capa aislante, que suele ser barniz. Está compuesto por dos bobinas, la primaria y la secundaria. La relación de vueltas del hilo de cobre entre el devanado primario y el secundario indicará la relación de transformación.

Los transformadores se basan en la inducción electromagnética. Al aplicar una fuerza electromotriz en el devanado primario, es decir una tensión, se origina un flujo magnético en el núcleo de hierro. Este flujo viajará desde el devanado primario hasta el secundario. Con su movimiento originará una fuerza electromotriz en el devanado secundario.

Según la Ley de Lenz, la corriente debe ser alterna para que se produzca esta variación de flujo. El transformador no puede utilizarse con corriente continua.

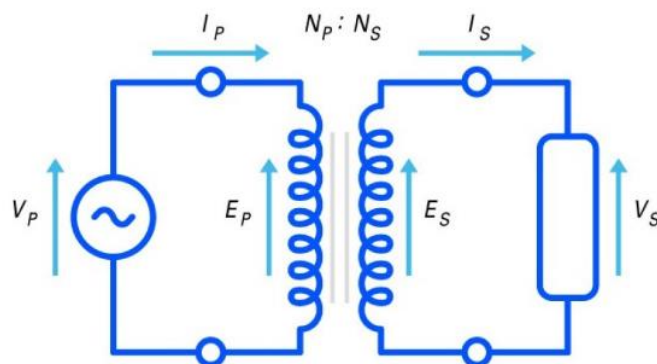


Fig. N° 6 Esquema básico de funcionamiento de un transformador ideal.

La relación de transformación del transformador la definimos con la siguiente ecuación:

$$N_p/N_s = V_p/V_s = I_s/I_p = r_t \quad (\text{Ecu. 1})$$

Donde (N_p) es el número de vueltas del devanado del primario, (N_s) el número de vueltas del secundario, (V_p) la tensión aplicada en el primario, (V_s) la obtenida en el secundario, (I_s) la intensidad que llega al primario, (I_p) la generada por el secundario y (r_t) la relación de transformación.

Los sensores de la serie SCT-013 son sensores que trabajan como transformadores, la corriente que circula por el cable que deseamos medir actúa como el devanado primario (1 espira) e internamente tiene un devanado secundario que dependiendo del modelo pueden tener hasta más de 2000 espiras.

La cantidad de espiras representa la relación entre corriente que circula por el cable y la que el sensor nos entrega, esta relación o proporción es la que diferencia entre los diferentes modelos de sensores SCT-013, adicionalmente pueden tener una resistencia de carga en la salida de esta forma en lugar de corriente se trabaja con una salida voltaje.

En el caso de este proyecto la salida del sensor es de tensión (RMS), y varía en un rango de $[-1V +1V]$, es decir, que cuando circulen 5A por el cable medido, a la salida se tendrá una señal senoidal de $\pm 1V$. Como se sabe, el microcontrolador PIC trabaja solo con tensiones positivas por lo que es necesario acondicionar la señal para no dañar el MCU. Para ello se realiza un circuito externo para añadir un offset central a la medida. Se ha añadido un punto de offset de DC de 2.5V por lo que el rango final es de 1.08V a 3.92V, dentro del rango de las entradas analógicas.

Los cálculos para llegar a los valores anteriores son:

$$V_{pico} = \sqrt{2} * V_{RMS} = \sqrt{2} * 1[V] = 1,414 [V] \quad (Ecu. 2)$$

$$V_{inferior} = 2.5[V] - 1,41[V] = 1,08 [V] \quad (Ecu. 3)$$

$$V_{superior} = 2.5[V] + 1,41[V] = 3.92[V] \quad (Ecu. 4)$$

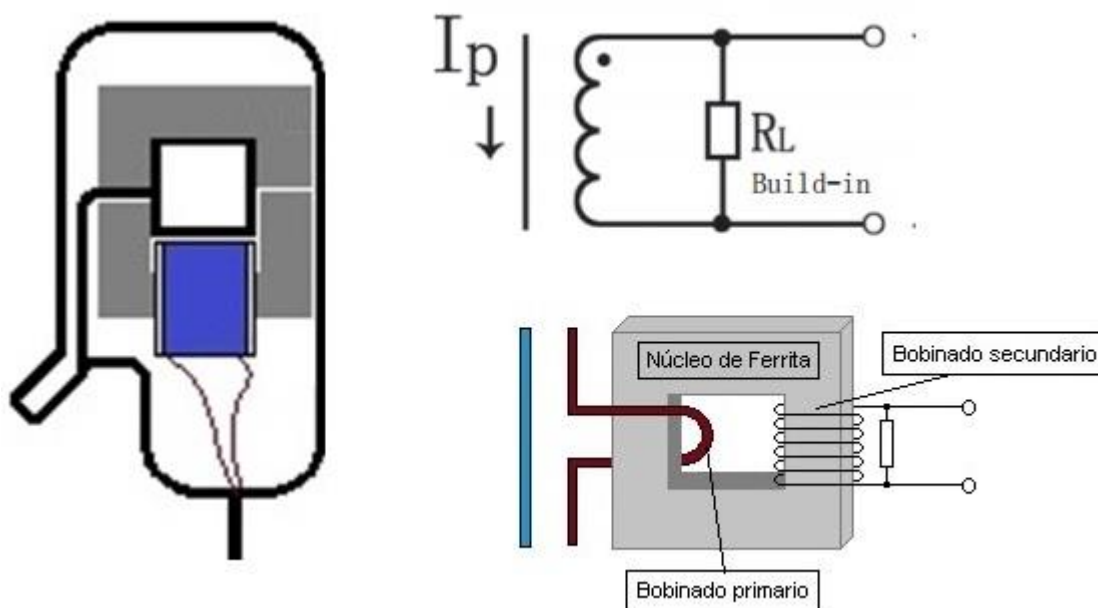


Fig. N° 7 Bloque de funcionamiento STC-013.

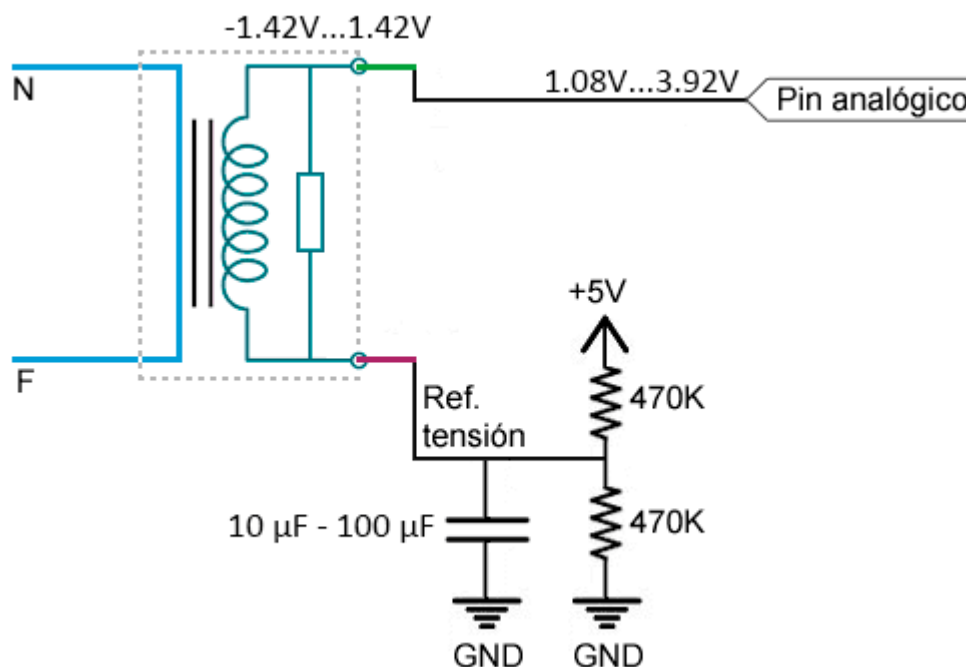


Fig. N° 8 Circuito de offset para acondicionar la señal

RTC DS3231

DS3231 es un módulo de reloj tiempo real de bajo costo y extremadamente preciso I2C (RTC), con un oscilador integrado con compensación de temperatura de cristal (TCXO). El dispositivo incorpora una batería que cuando se desconecta la fuente de alimentación principal mantiene el cronometraje preciso.

La compensación de temperatura de precisión de referencia de voltaje y circuito comparador supervisa el estado de VCC para detectar fallas de energía, proporcionar una salida de reset, y si es necesario, cambia automáticamente a la fuente de alimentación de reserva.

El IC funciona con bajo consumo de energía cuando funciona con una batería, entonces consume menos de 500 nA de energía. Gestiona todas las funciones de cronometraje.

Este reloj/calendario de chip puede proporcionar información de segundos, minutos, horas, día, fecha, mes y año. Este reloj IC funciona en formato de 24 horas o de 12 horas con indicador AM/PM.

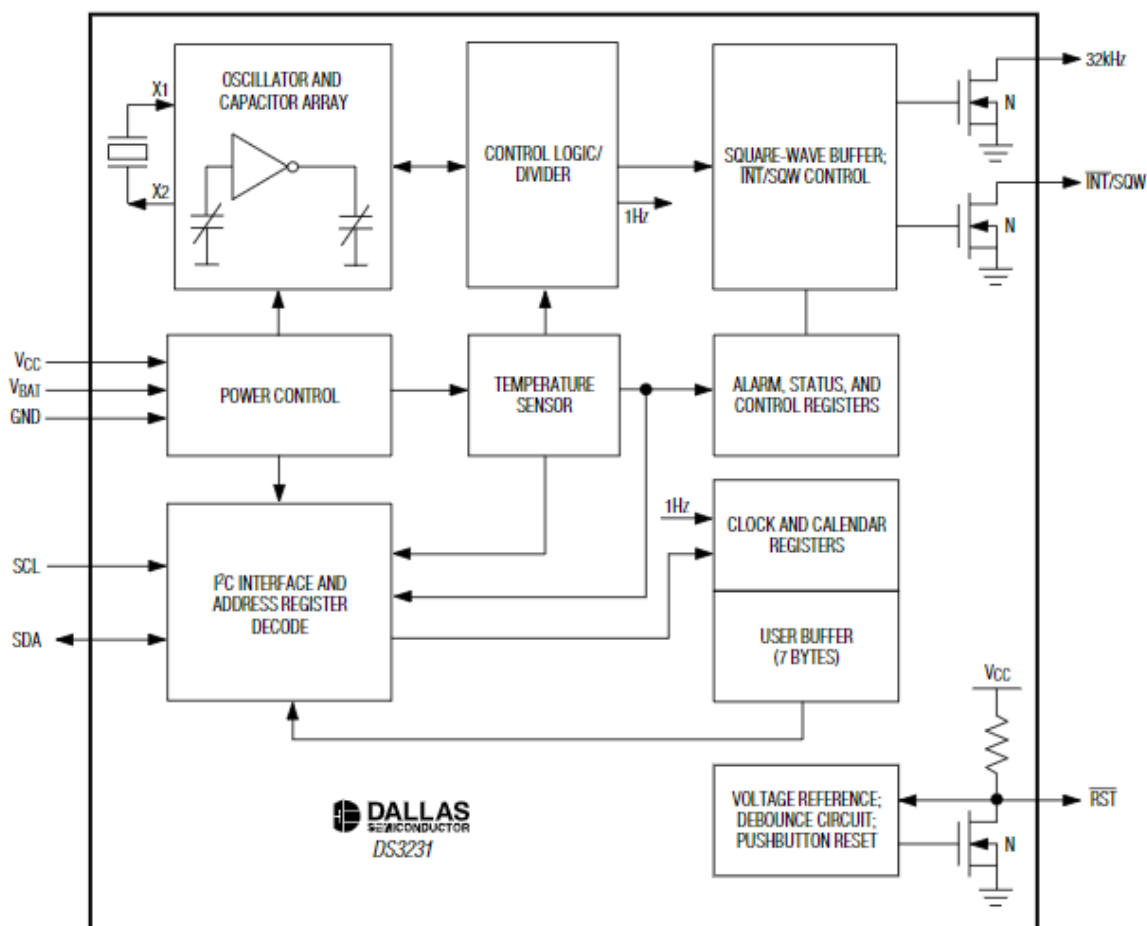


Fig. N° 9 Diagrama en bloques RTC DS3231.

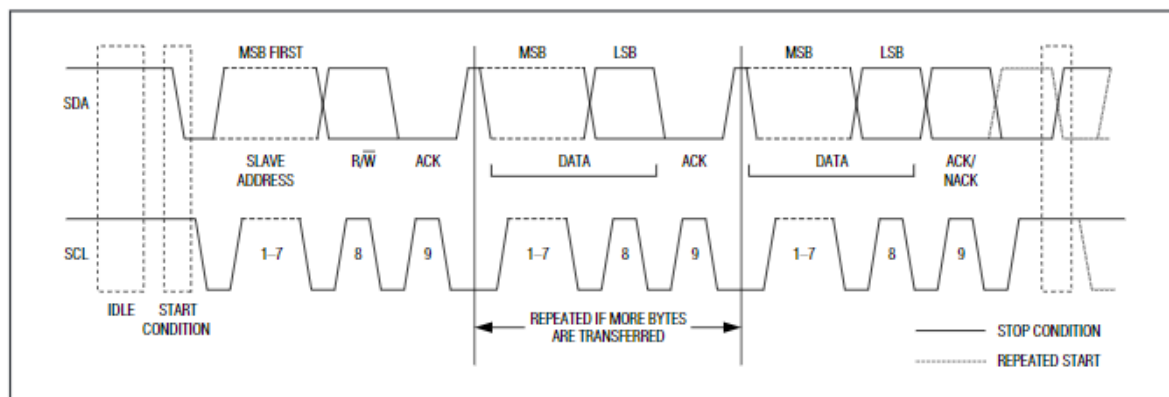


Fig. N° 10 Diagrama de tiempo para transferencia de datos.



Fig. N° 11 Imagen del módulo real.

Memoria EEPROM AT24C08

Las PROM utilizan algún tipo de mecanismo de fundición para almacenar bits, donde un hilo de memoria se funde o queda intacto para representar un 0 o un 1. El proceso de fundición es irreversible; una vez que una PROM ha sido programada no puede cambiarse.

Una EPROM es una PROM borrable. A diferencia de una PROM ordinaria, una EPROM puede ser reprogramada si antes se borra el programa existente en la matriz de memoria.

Una EPROM utiliza una matriz NMOSFET con una estructura de puerta aislada. La puerta del transistor aislada no tiene ninguna conexión eléctrica y puede almacenar una carga eléctrica durante un período de tiempo indefinido. Los bits de datos en este tipo de matriz se representan mediante la presencia o ausencia de una carga almacenada en la puerta. El borrado de un bit de datos es un proceso que elimina la carga de la puerta. Los dos tipos fundamentales de memorias PROM borrables son las PROM borrables por rayos ultravioleta (UV EPROM) y las PROM borrables eléctricamente (EEPROM). (Floyd, 2006, pág. 630)

Las PROM borrables eléctricamente se pueden borrar y programar mediante impulsos eléctricos. Ya que se pueden grabar y borrar eléctricamente, las EEPROM se pueden programar y borrar rápidamente dentro del propio circuito final con fines de reprogramación.

El AT24C08 es un integrado que incorpora una memoria EEPROM que puede ser accedida a través del bus I2C.

Podemos conectar esta EEPROM externa a procesadores como Arduino para almacenar variables que persistan entre reinicios o cuando se pierda la alimentación.

La EEPROM AT24C08 dispone de 8 Kb, y al igual que todas las EEPROM estas poseen una vida útil de acuerdo al número de ciclos de escrituras, en este caso es muy alta con 1.000.000 de ciclos de escrituras.

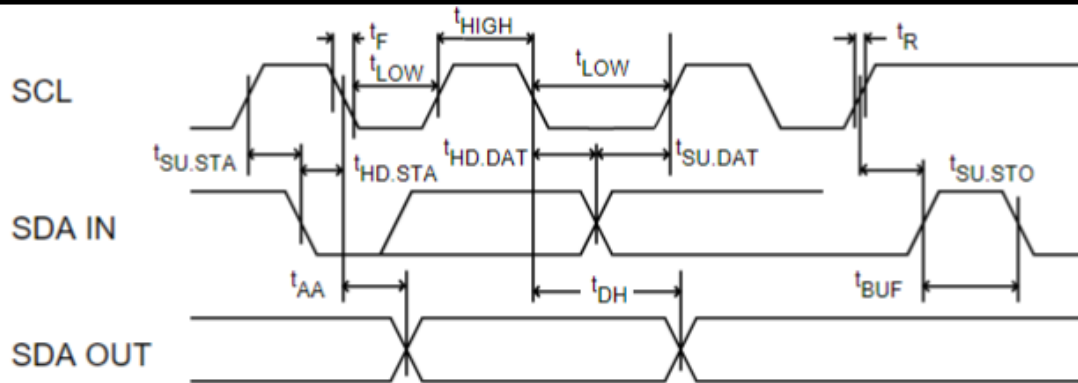


Fig. N° 12 *Diagrama de tiempo para transferencia de datos.*

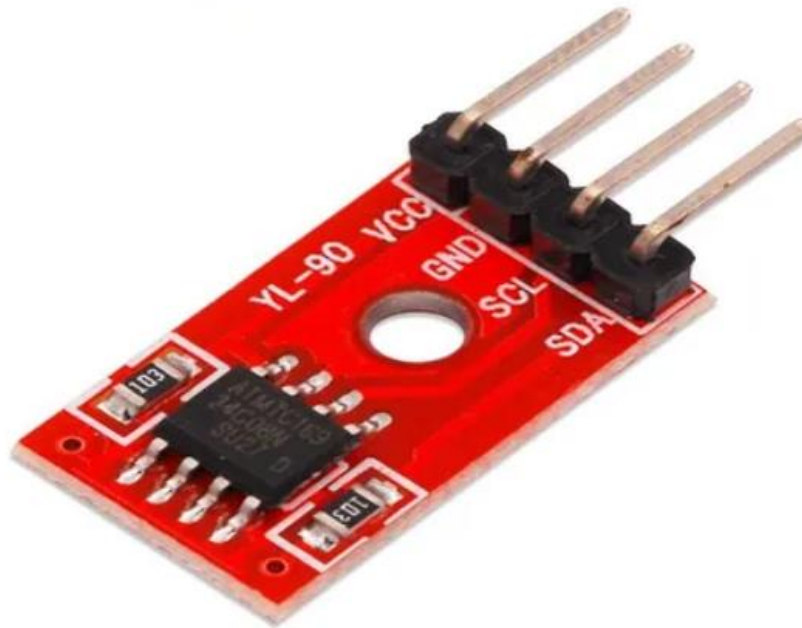


Fig. N° 13 *Imagen del módulo EEPROM AT24C08.*

LCD 20x4

Las pantallas LCD tienen diferentes características porque son fabricadas en diferentes tamaños y su uso depende de la necesidad que se tenga, los tamaños comunes en el mercado son el LCD 16x2 (2 filas y 16 caracteres), de 20x4, 20x2 y 40x2. En este caso la pantalla utilizada es de 20x4.

Para que el LCD consiga mostrar los caracteres se utiliza un controlador en estos dispositivos donde el más popular es el Hitachi HD44780, el cual trabaja con LCDs monocromáticos de hasta 80 caracteres alfanuméricos y símbolos. Y poseen una luz de fondo la cual puede ser de color azul o verde y adicionalmente poseen un pin con el cual se puede variar el contraste de la pantalla LCD utilizando para eso un potenciómetro.

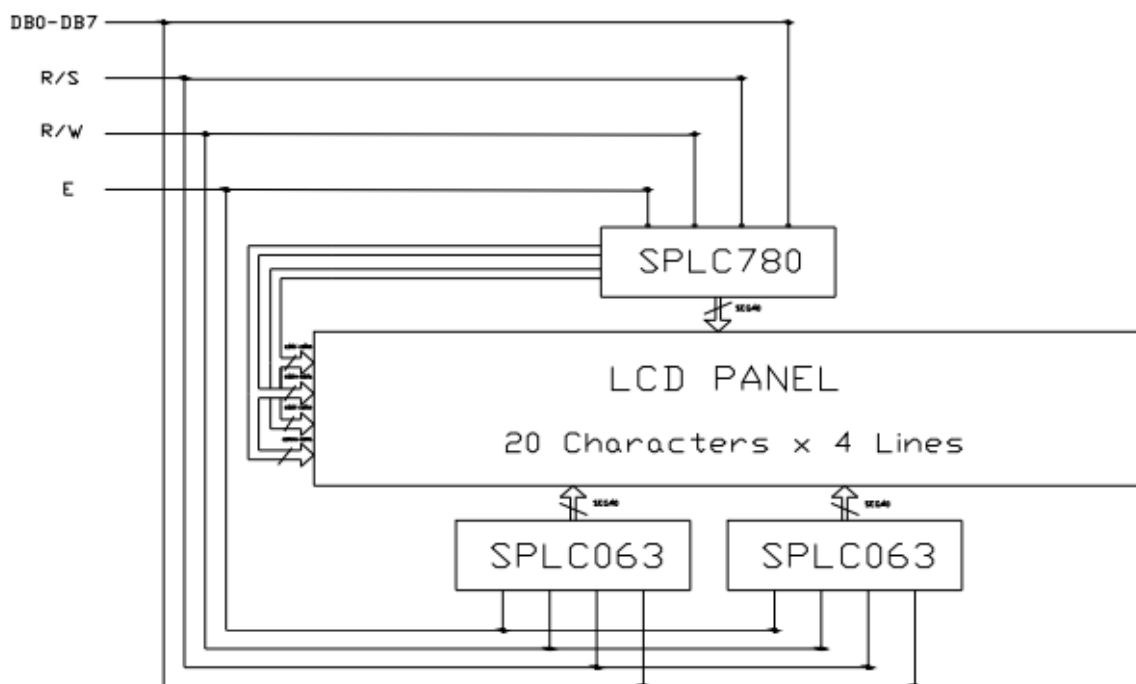


Fig. N° 14 Diagrama en bloques LCD 20x4.



Fig. N° 15 Imagen real del módulo LCD 20x4.

Level Shifter (Cambiador de nivel)

Un cambiador de nivel, también llamado cambiador de nivel lógico o traductor de nivel de voltaje, es un circuito que se utiliza para traducir señales de un nivel lógico o dominio de voltaje a otro, lo que permite la compatibilidad entre circuitos integrados con diferentes requisitos de voltaje. Los sistemas modernos utilizan cambiadores de nivel para unir dominios entre procesadores, lógica, sensores y otros circuitos. En los últimos años, los tres niveles lógicos más comunes han sido 1.8V, 3.3V y 5V.

Existen diferentes tipos de cambiador de nivel:

- Unidireccional: todos los pines de entrada están dedicados a un dominio de voltaje, todos los pines de salida están dedicados al otro.
- Bidireccional con puertos dedicados: cada dominio de voltaje tiene pines de entrada y salida, pero la dirección de datos de un pin no cambia.
- Bidireccional con indicador de dirección externo: cuando se cambia una señal externa, las entradas se convierten en salidas y viceversa.
- Bi-direccional, detección automática: un par de dominios de voltaje de expansión de E / S pueden actuar como entradas o salidas dependiendo del estímulo externo sin la necesidad de un pin de control de dirección dedicado.

En el presente proyecto el cambiador de nivel se utiliza para poder comunicar de manera serial el PIC18F4550 y la ESP-12E, debido a que el PIC trabaja con 5V y la ESP con 3.3v, es decir, una comunicación directa entre ellos generaría la destrucción del módulo ESP-12E.

Para resolver el inconveniente anteriormente menciona es que se utiliza un cambiador de nivel bidireccional. Aunque para transmitir información de la ESP al PIC no sería necesario un circuito como este.

Se utilizaron MOSFET de canal N, más específicamente se utilizaron los 2N7000 que son los transistores recomendados para realizar este tipo de circuito.

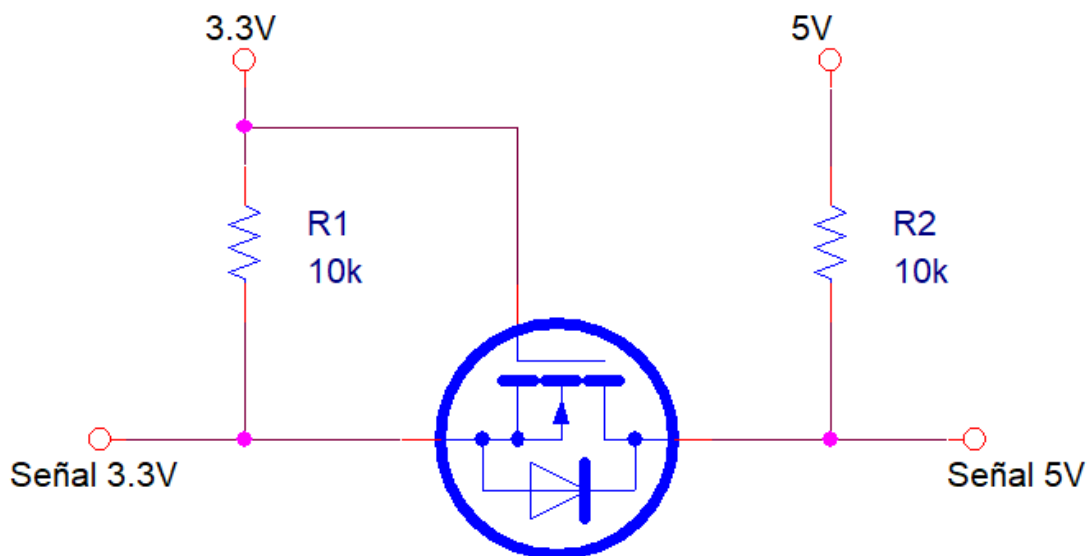


Fig. N° 16 Convertidor de nivel entre Tx(ESP) a Rx(PIC).

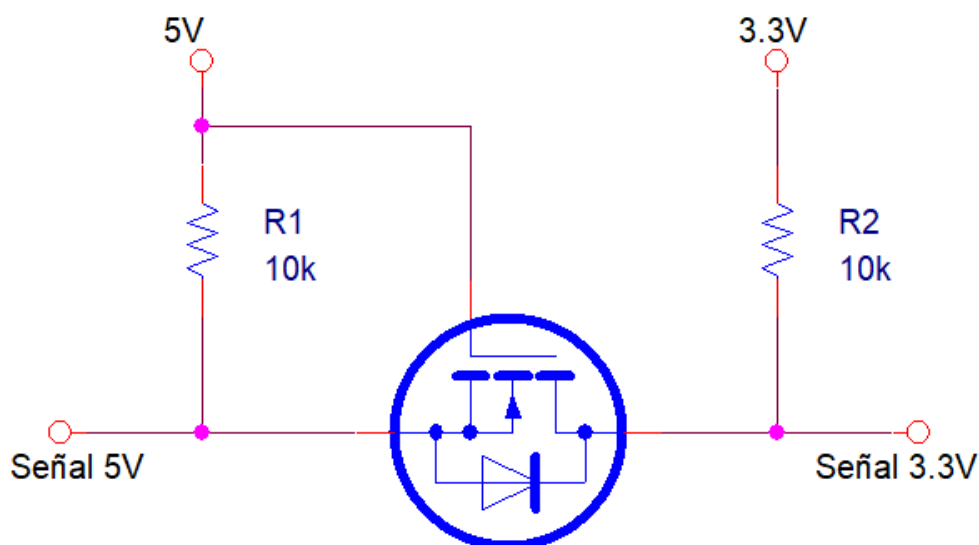


Fig. N° 17 Convertidor de nivel entre Tx(PIC) a Rx(ESP)

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
$V_{(BR)DSS}$	Drain-source breakdown voltage	$I_D = 250\mu A, V_{GS} = 0$	60			V
I_{DSS}	Zero gate voltage drain current ($V_{GS} = 0$)	$V_{DS} = \text{max rating}$ $V_{DS} = \text{max rating}, T_C = 125^\circ C$			1 10	μA μA
I_{GSS}	Gate-body leakage current ($V_{DS} = 0$)	$V_{GS} = \pm 18V$			± 100	nA
$V_{GS(th)}$	Gate threshold voltage	$V_{DS} = V_{GS}, I_D = 250\mu A$	1	2.1	3	V
$R_{DS(on)}$	Static drain-source on resistance	$V_{GS} = 10V, I_D = 0.5A$ $V_{GS} = 4.5V, I_D = 0.5A$		1.8 2	5 5.3	Ω Ω

Fig. N° 18 Características estáticas del IRFZ44N.

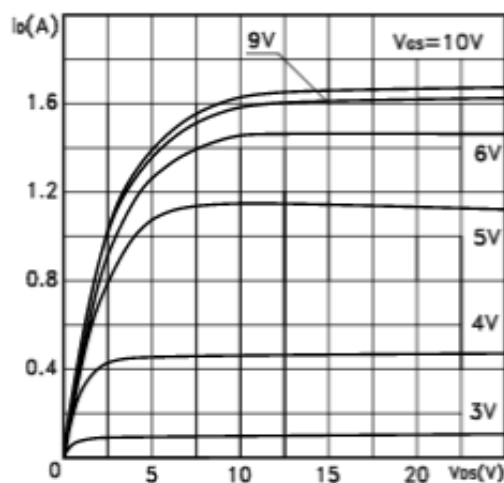


Fig. N° 19 Curvas características de salida del IRFZ44N

Diagrama en bloques del dispositivo

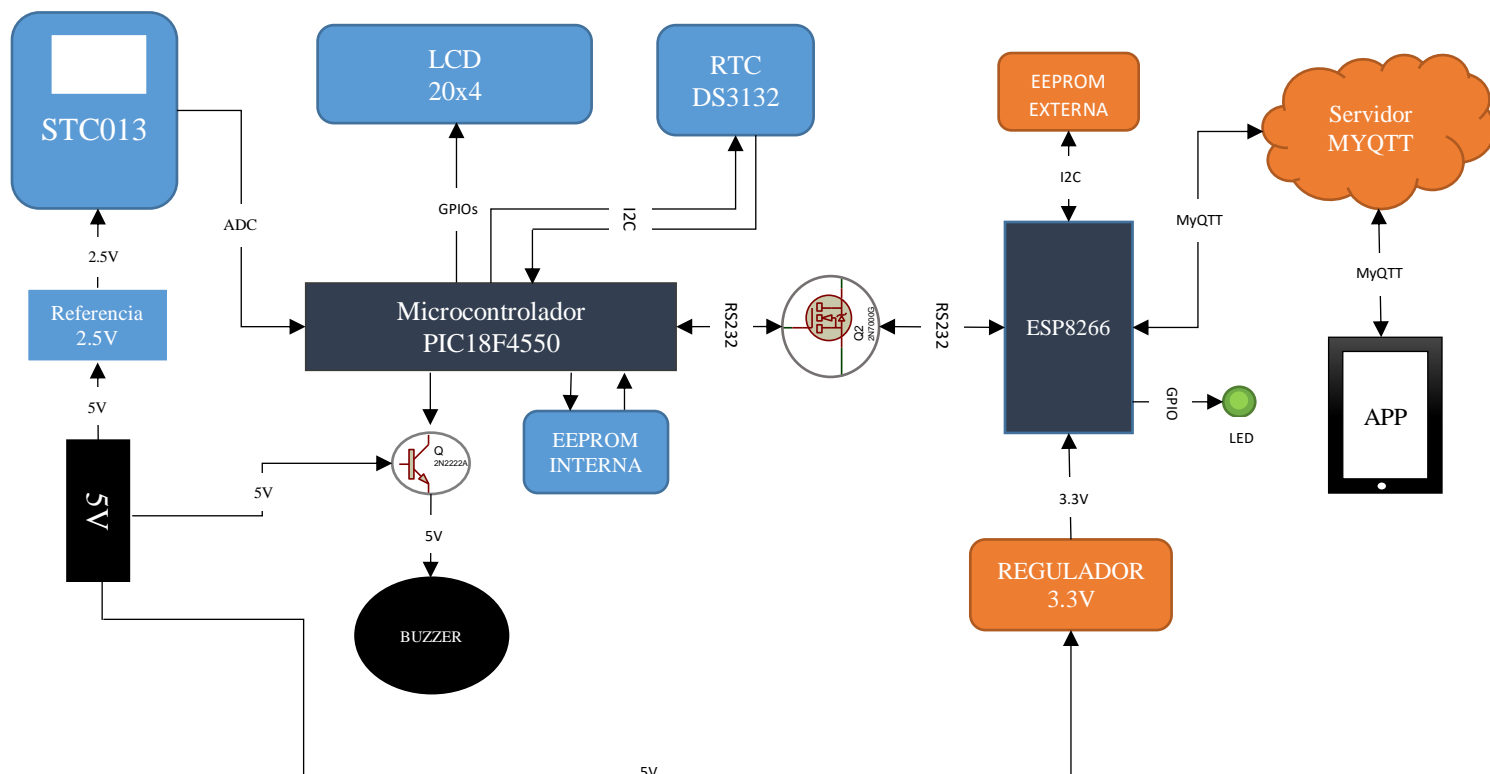


Fig. N° 20 Diagrama en bloque horómetro WiFi programable.

Descripción de cada una de las partes

Detección de funcionamiento

En un principio para que el dispositivo sepa que una maquina está funcionando utiliza el sensor de corriente no invasivo STC013 anteriormente mencionado. Este sensor se coloca a través de una de las fases de alimentación y cuando detecta un consumo genera una señal analógica en su salida, a la cual se le coloca un offset para poder ser interpretada por el MCU. Es importante tener en cuenta que el sensor de corriente va a variar de acuerdo a los consumos a medir, en este caso como se harán pruebas domiciliarias se utilizara un transductor que mide entre 0A y 5A, pero para trabajar de manera industrial vienen transductores de 30A, 50A y 100A.

El offset para poder leer la señal se realiza a través de un cambio en el nivel de tensión de referencia, a su vez este nivel de tensión se logra con un divisor resistivo y la tensión de alimentación, en la figura 21 se observa el circuito utilizado.

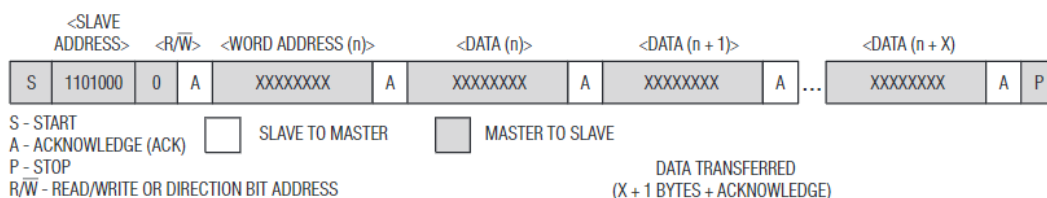


Fig. N° 22 Escritura de datos – Esclavo receptor

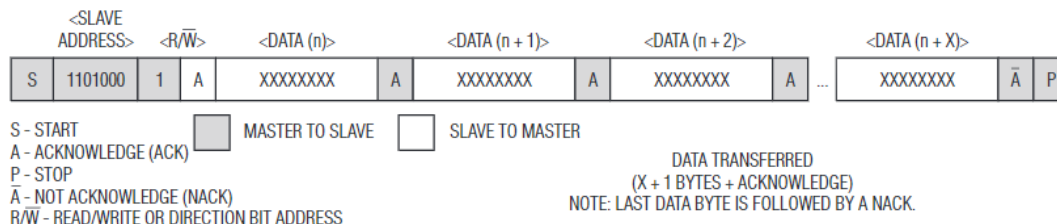


Fig. N° 23 Lectura de datos – Esclavo transmisor

Señales de I2C:

- **START** : Mientras la línea SCL permanece en alto, se genera una transición high-to-low de la línea SDA.
- **STOP**: Mientras la línea SCL permanece en alto, se genera una transición low-to-high de la línea SDA.
- **DATA**: La transmisión de datos ocurre cuando la señal de reloj SCL está en alto (high) mientras que las transiciones entre los niveles de datos se permiten cuando la señal SCL está en bajo (low).
- **ACK**: Mientras el transmisor permite que flote la línea SDA en alto (high) el receptor la jala (pulls) a nivel bajo (low).

Algunas particularidades del I2C son Clock stretching, General Call, y direccionamiento extendido. El Clock Stretching se da cuando el esclavo (receptor) requiere de más tiempo para procesar un bit puede mantener baja (pull-low) la señal de reloj SCL. El maestro espera en ese caso hasta que el esclavo libere la línea del SCL antes de continuar con el siguiente bit. La General call es la transmisión que se recibe por todos los dispositivos conectados al bus. Y por último el direccionamiento extendido de 10 bits s (para diseños nuevos) en caso de que ya no queden disponibles direcciones 7-bits.

Conociendo el funcionamiento del protocolo se pueden definir los pasos para configurar el reloj:

1. En la codificación RTC, necesitamos la primera dirección de dispositivo RTC (dirección esclava) a través de la cual el microcontrolador desea comunicarse con el DS3231.
2. La dirección del dispositivo DS3231 RTC es 0xD0 (Obtenida en la hoja de datos).
3. Inicializar I2C en PIC18F4550
4. Inicie la comunicación I2C con la dirección de escritura del dispositivo, es decir, 0xD0.

5. Luego, envíe la dirección de registro de segundos que es 0x00, luego envíe el valor de segundos para escribir en RTC. La dirección RTC se incrementa automáticamente, así que a continuación, solo tenemos que enviar los valores de minutos, horas.
6. Detener la comunicación I2C.

Una vez configurado el reloj, cuando se detecta un consumo de energía por parte de la maquina se toma la hora de ese preciso momento y se almacena en una variable hasta que se detecta que la maquina se detuvo. Cuando esto sucede se realiza la diferencia entre la hora del comienzo del consumo y la hora de finalización, y se obtiene el tiempo que estuvo funcionando el equipo.

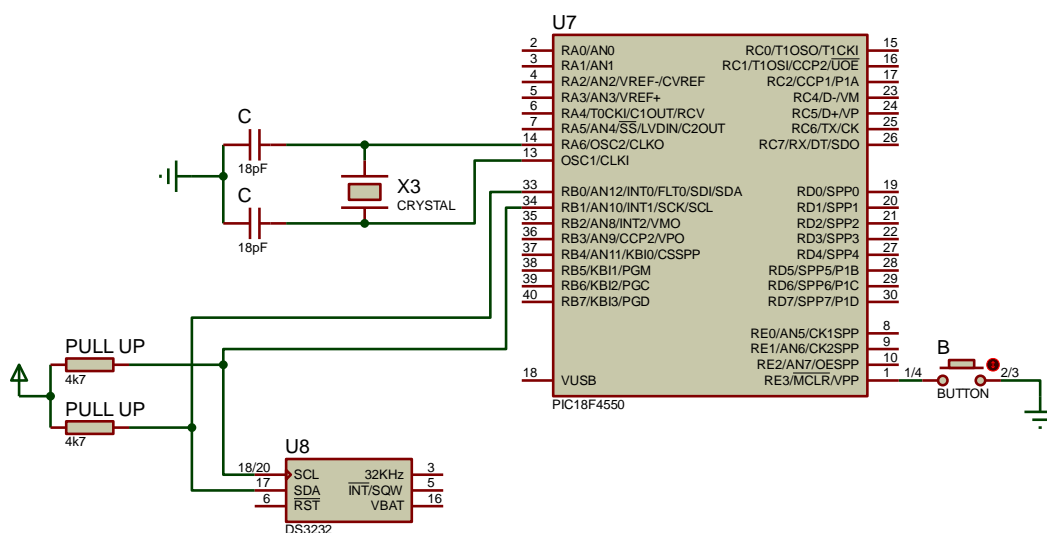


Fig. N° 24 Conexión DS3231 con PIC18F4550

Comunicación serial entre PIC18F4550 y ESP-12E

Cuando se obtiene el tiempo de funcionamiento se procede a transmitirlo desde el PIC a la ESP a través de la comunicación serial RS-232.

La comunicación serial permite la transmisión-recepción bit a bit de un byte completo, típicamente se utiliza para transmitir datos en formato ASCII, es por ello que mediante programación se debe transformar los tipos de variables cuando se envían y se reciben para luego poder trabajar con las variables originales.

Para realizar la comunicación se utilizan tres líneas de transmisión, tierra (GND), transmitir(Tx) y recibir (Rx). Debido a que la transmisión es asíncrona es posible enviar datos por una línea y estar recibiendo por otra.

Este puerto funciona de forma asíncrona, gracias a un protocolo que inicia la transmisión con una señal de “start” que prepara el receptor para recibir la palabra (bits). Tras enviar esta palabra, que será un código ASCII para cada carácter, se envía una señal de “stop” para que



el receptor descansa tras codificar la palabra y se mantenga a la espera para recibir otra.



Fig. N° 25 Conexión DS3231 con PIC18F4550

Las características más importantes de la comunicación serial son:

- Velocidad de transmisión
- Número de bits de datos
- El número de bits de paro
- Y si cuenta con bit de paridad

Indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds), las velocidades de transmisión más comunes son de 115200, 9600, y 4800. En este caso se utiliza 9600 baudios.

Los bits de datos se refieren a la cantidad de bits (palabra) en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits, las cantidades más comunes de bits por paquete son 5, 7 y 8 bits, el número de bits que se envía depende en el tipo de información que se transfiere. Para ASCII extendido es de 0 a 255, lo que utiliza 8 bits pero si el tipo de datos que se está transfiriendo es texto simple (ASCII estándar), entonces es suficiente con utilizar 7 bits por paquete para la comunicación. Un paquete se refiere a una transferencia de un byte, incluyendo los bits de inicio/paro, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usa para referirse a todos los casos.

Los bits de paro se utilizan para indicar el fin de la comunicación de un paquete. Debido a la manera en la que se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj es posible que los dispositivos no estén sincronizados, por lo tanto, los bits de paro no solo indican el final de la transmisión sino además dan un margen de tolerancia para esa diferencia de relojes. Mientras más bits de paro se utilicen mayor será la tolerancia a la sincronía de los relojes a costa de una disminución de velocidad en la transmisión.

La paridad es una forma sencilla de verificar si hay errores en la transmisión serial. En caso de habilitar la paridad par o impar, el puerto serial fijará el bit de paridad (el último bit después de los bits de datos) a un valor para asegurarse que la transmisión tenga un número par o impar de bits en estado lógico alto.

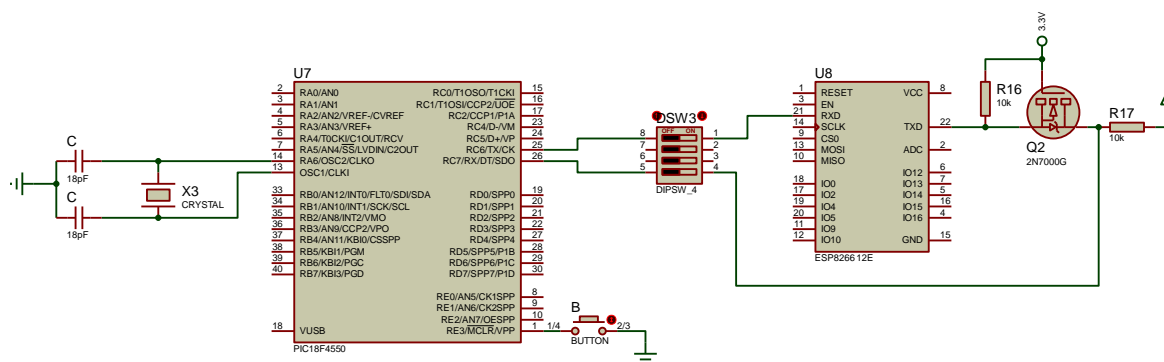


Fig. N° 26 Conexión comunicación serial

Comunicación WiFi mediante protocolo MQTT (MQ Telemetry Transport)

Para comprender de mejor manera el protocolo MQTT primero se va a explicar que es y cómo funciona IoT. La abreviación IoT hace referencia a Internet of thing (Internet de las cosas). Esta describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas.

Muchas tecnologías son las que han posibilitado la utilización de IoT, algunas de ellas son:

- Accede a la tecnología de sensores de bajo costo y potencia
- Conectividad. Una gran cantidad de protocolos de red para Internet ha facilitado la conexión de sensores a la nube y a otras cosas para lograr una transferencia de datos eficiente
- Plataformas de informática en la nube. El aumento en la disponibilidad de plataformas en la nube permite a las empresas y a los consumidores acceder a la infraestructura que necesitan para escalar sin tener que administrarlo todo.
- Además, otras tecnologías permiten el crecimiento y una perspectiva de mejora de IoT, estas pueden ser:
 - Aprendizaje automático y analítica.
 - Inteligencia artificial (IA) conversacional.

Protocolos de comunicación en IoT.

Un protocolo de comunicación es una serie de normas que definimos para que dos o más dispositivos puedan comunicarse y entenderse. Existen muchas formas de realizar la comunicación M2M, Sin embargo, en el campo del IoT tenemos ciertos requisitos especiales que hacen que las habituales formas de comunicación entre dispositivos no sean totalmente adecuadas.

En primer lugar, en el IoT hay (o puede haber) una gran cantidad de dispositivos. Algunos de ellos serán pequeños, como sensores o actuadores y otros serán más grandes, como un servidor que recoge información, almacena datos, y procesa estadísticas.

Otro requisito es que queremos que sea escalable, es decir, que puedan añadirse o retirarse dinámicamente dispositivos sin que el comportamiento global del sistema se modifique.

También es importante mantener débil el acoplamiento entre dispositivos. Es decir, queremos que la dependencia entre los dispositivos sea la menor posible, y deseablemente nula.

Otro requisito es que algunos de los dispositivos serán dispositivos embebidos, con bajo coste y escasa capacidad de cálculo. Por tanto, tiene que ser un protocolo que requiera poca capacidad de procesamiento.

Relacionado con la variedad y número de dispositivos, vamos a querer interoperabilidad. Es decir, que nuestra solución funcione la mayor variedad de dispositivos, sistemas operativos, y lenguajes de programación.

Además, es posible que haya un gran número de comunicaciones simultáneas y, en general, se requiere una respuesta rápida. Esto requiere que los mensajes transmitidos sean pequeños y, nuevamente, no requieran un gran procesamiento.

Por supuesto, tenemos siempre presente el condicionante de la seguridad, ya que estos dispositivos están expuestos a Internet (que no es un lugar nada seguro) y transmiten información privada e incluso controlan sistemas físicos.

Finalmente, tenemos que poder acceder a los dispositivos fácilmente, por lo que tendremos que lidiar con direcciones dinámicas y DHCP, posibles conexiones con mala latencia o ancho de banda, dependencia con la infraestructura de la red, firewalls, etc.

Una posible solución, que está siendo ampliamente empleada, es externalizar la comunicación un servicio de notificaciones centralizado.

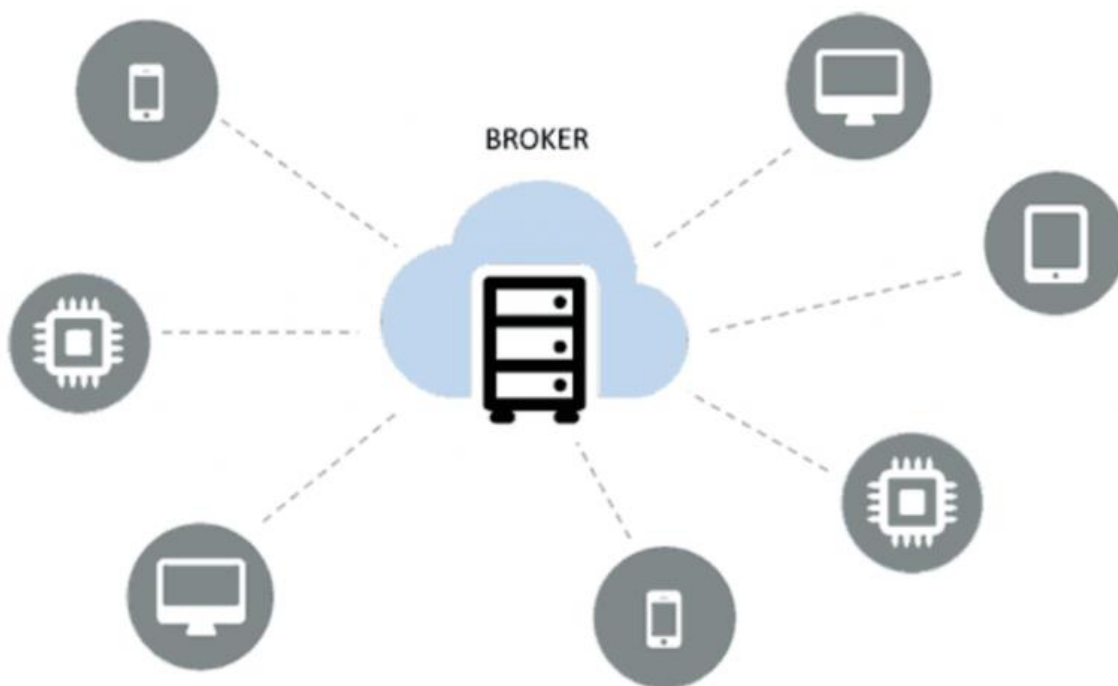


Fig. N° 27 *Conexión comunicación serial*

En definitiva, la solución consiste en disponer un servidor central que se encarga de recibir los mensajes de todos los dispositivos emisores, y distribuirlos a los receptores. De



forma genérica llamaremos a este servidor 'Router' o 'Broker'. El servidor mantiene un registro de los dispositivos conectados, recibe los mensajes, y los distribuye al resto dispositivos, filtrando los destinatarios según algún criterio.

Existen muchos protocolos de comunicación en el área de IoT, algunos de ellos son:

- MQTT (MQ Telemetry Transport)
- AMQP (Advanced Message Queuing Protocol)
- WAMP (Web Application Messaging Protocol)
- CoAP (Constrained Application Protocol)
- STOMP (Streaming Text Oriented Messaging Protocol)
- XMPP (Extensible Messaging and Presence Protocol)
- WMQ (WebSphere MQ)

En este proyecto se optó por utilizar MQTT debido a su popularidad, cantidad de información y sencillez para utilizarlo. MQTT es un protocolo de comunicación Machine to Machine (M2M) de tipo message queue. Está basado en la pila TCP/IP como base para la comunicación. En el caso de MQTT cada conexión se mantiene abierta y se "reutiliza" en cada comunicación.

El funcionamiento del MQTT es un servicio de mensajería push con patrón publicador/suscriptor (pub-sub), en este tipo de infraestructuras los clientes se conectan con un servidor central denominado broker. Para filtrar los mensajes que son enviados a cada cliente los mensajes se disponen en topics organizados jerárquicamente. Un cliente puede publicar un mensaje en un determinado topic. Otros clientes pueden suscribirse a este topic, y el broker le hará llegar los mensajes suscritos. Los clientes inician una conexión TCP/IP con el broker, el cual mantiene un registro de los clientes conectados. Esta conexión se mantiene abierta hasta que el cliente la finaliza.

Uno de los componentes más importantes del protocolo MQTT es la definición y tipología de los mensajes, ya que son una de las bases de la agilidad en la que radica su fortaleza. Cada mensaje consta de 3 partes como se observa en la siguiente imagen.

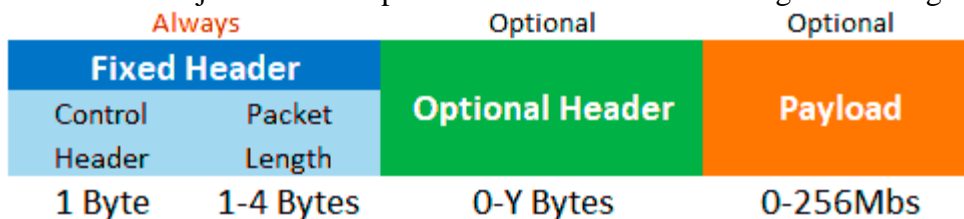


Fig. N° 28 división de bytes en mensajes MQTT

Cabecera fija: Ocupa 2 a 5 bytes, obligatorio. Consta de un código de control, que identifica el tipo de mensaje enviado, y de la longitud del mensaje. La longitud se codifica en 1 a 4 bytes, de los cuales se emplean los 7 primeros bits, y el último es un bit de continuidad.

Cabecera variable: Opcional, contiene información adicional que es necesaria en ciertos mensajes o situaciones.

Contenido(payload): Es el contenido real del mensaje. Puede tener un máximo de 256 Mb aunque en implementaciones reales el máximo es de 2 a 4 kB.

Por las ventajas que ofrece este protocolo de comunicación es por la cual se decidió usarlo, es decir, tenemos todas las ventajas del patrón pub/sub como son escalabilidad,



asincronismo, desacoplamiento entre clientes.

Además, MQTT aporta una serie de características que le han hecho salir sobre otros competidores. La principal, como hemos mencionado, es su sencillez y ligereza. Esto lo hace adecuado para aplicaciones IoT, donde frecuentemente se emplean dispositivos de escasa potencia.

Además, esta menor necesidad de recursos se traduce en un menor consumo de energía, lo cual es interesante en dispositivos que funcionan de manera constante. Otra consecuencia de la ligereza del protocolo MQTT es que requiere un ancho de banda mínimo, lo cual es importante en redes inalámbricas, o conexiones con posibles problemas de calidad.

Por último, MQTT dispone de medidas adicionales importantes, como la seguridad, calidad del servicio (QoS) y es una solución largamente testada y consolidada, que aporta robustez y fiabilidad.

Comunicación entre ESP8266 y APK

La APK se realizó en APP Inventor que es un entorno de desarrollo de software.

Mediante esta aplicación es posible cargar nombre del equipo o maquina a la que se le realizara el seguimiento de mantenimiento y, además, se podrán cargar los tiempos en los que es necesario realizarle algún mantenimiento. A través del protocolo de comunicación antes descrito la APK envía un vector de información que es recibido por la ESP8266 y posteriormente es guardado en una memoria EEPROM. Con cada llegada de un vector se actualiza por completo el almacenado en la memoria para evitar problemas con tiempos eliminados o agregados.

A medida que avanza el conteo de tiempo de funcionamiento de un equipo este valor se ira comparando con cada uno de los valores almacenados en la memoria, y en caso de coincidir generara una alerta para realizar el mantenimiento, tanto en horómetro mediante un buzzer y la pantalla LCD como en la aplicación mediante una notificación en pantalla.

La aplicación no solamente permite cargar la información de mantenimiento sino también permite ver las horas de funcionamiento del equipo al que está conectado el horómetro y da aviso en caso de perder la conexión con el servidor MyMQTT. A continuación, se mostrará una imagen de la vista del usuario sobre la aplicación.

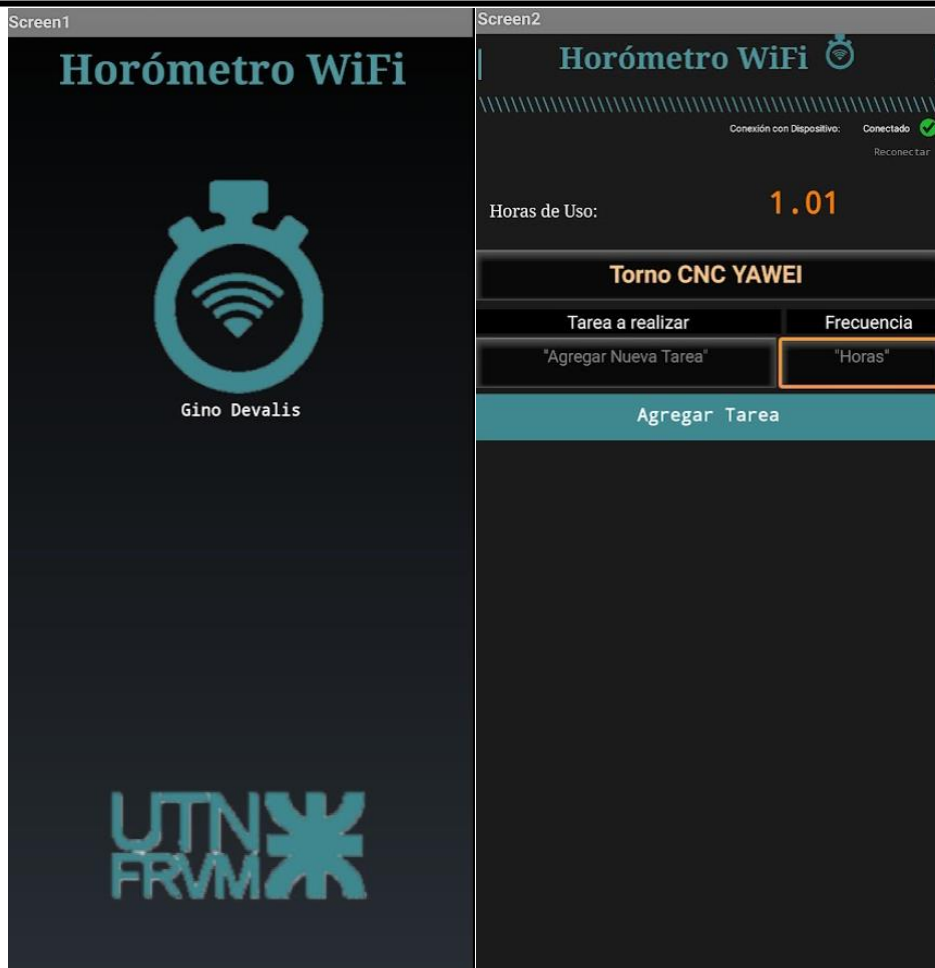


Fig. N° 29 Interfaz con usuario.



Fig. N° 30 Alertas y carga de información.

Diagrama de flujo

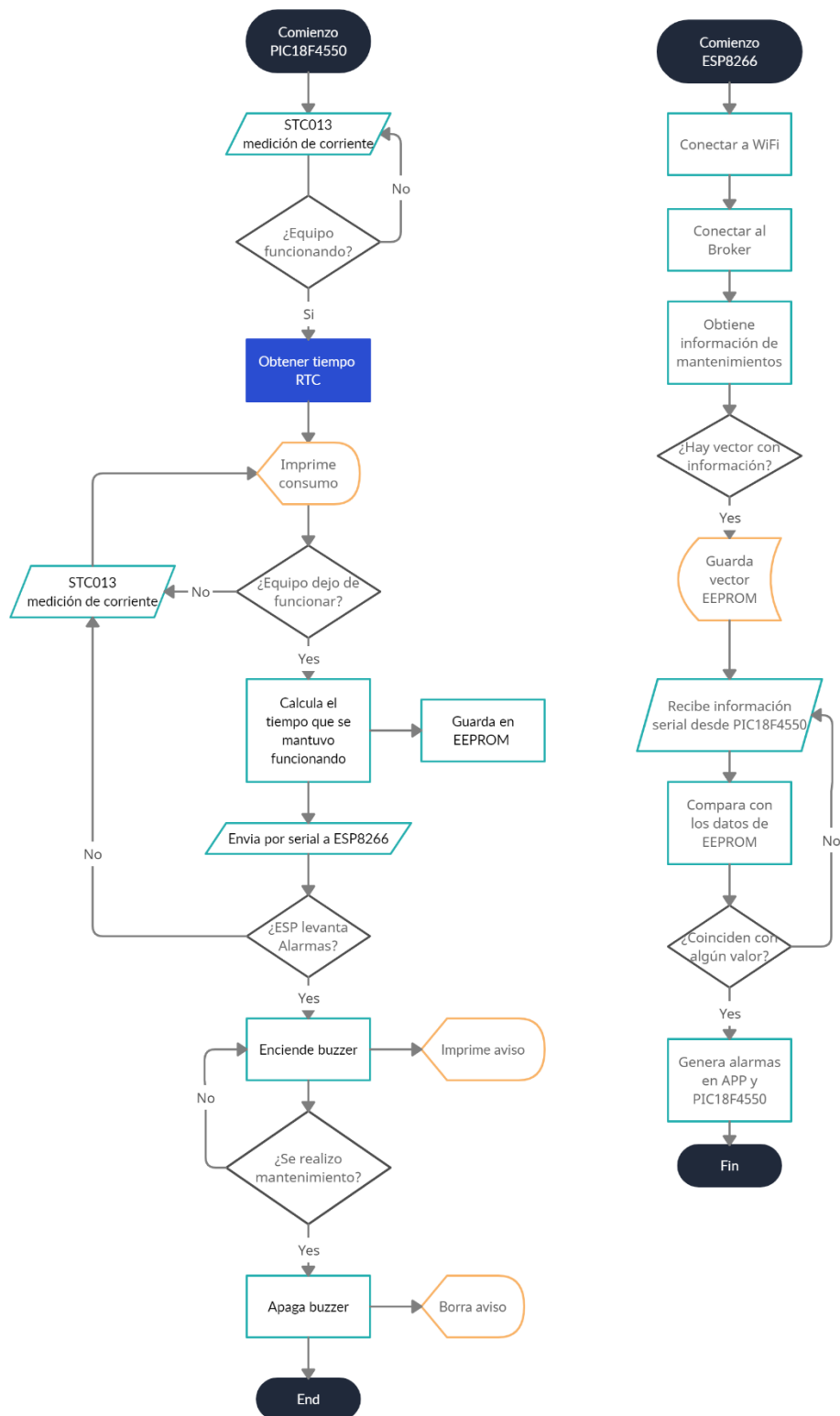


Fig. N° 31 Diagrama de flujo PIC18F4550 y ESP8266.

Evaluación Final del Sistema

Preparación del prototipo con materiales finales

En el diagrama esquemático y el layout del prototipo se ha realizado en el software Proteus 8.9 professional, este se puede observar a continuación en la figura 32.

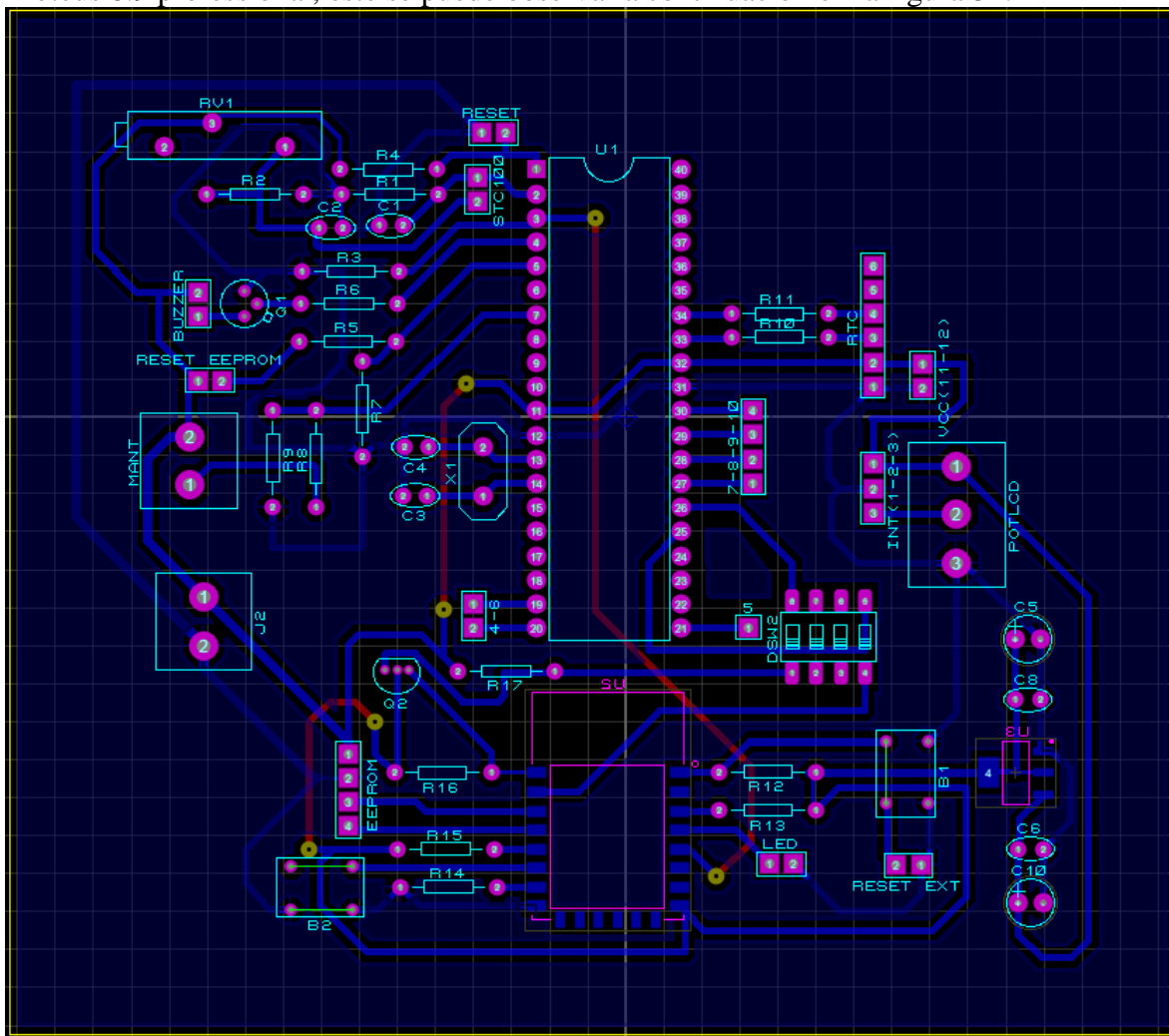


Fig. N° 32 Layout PCB horómetro WiFi.

Una vez definido y asegurado el layout se procede a imprimirlo sobre papel fotográfico para luego transferirlo mediante calor (método de planchado) a la placa de cobre virgen. En la imagen de la figura 33 se observa que la transferencia de tinta no se realizó de manera correcta en los bordes de la placa, por ello es que se realiza una corrección con marcador indeleble, en la figura 34 se ve la corrección de este error y la mejora de algunos detalles que se veían en el proceso.

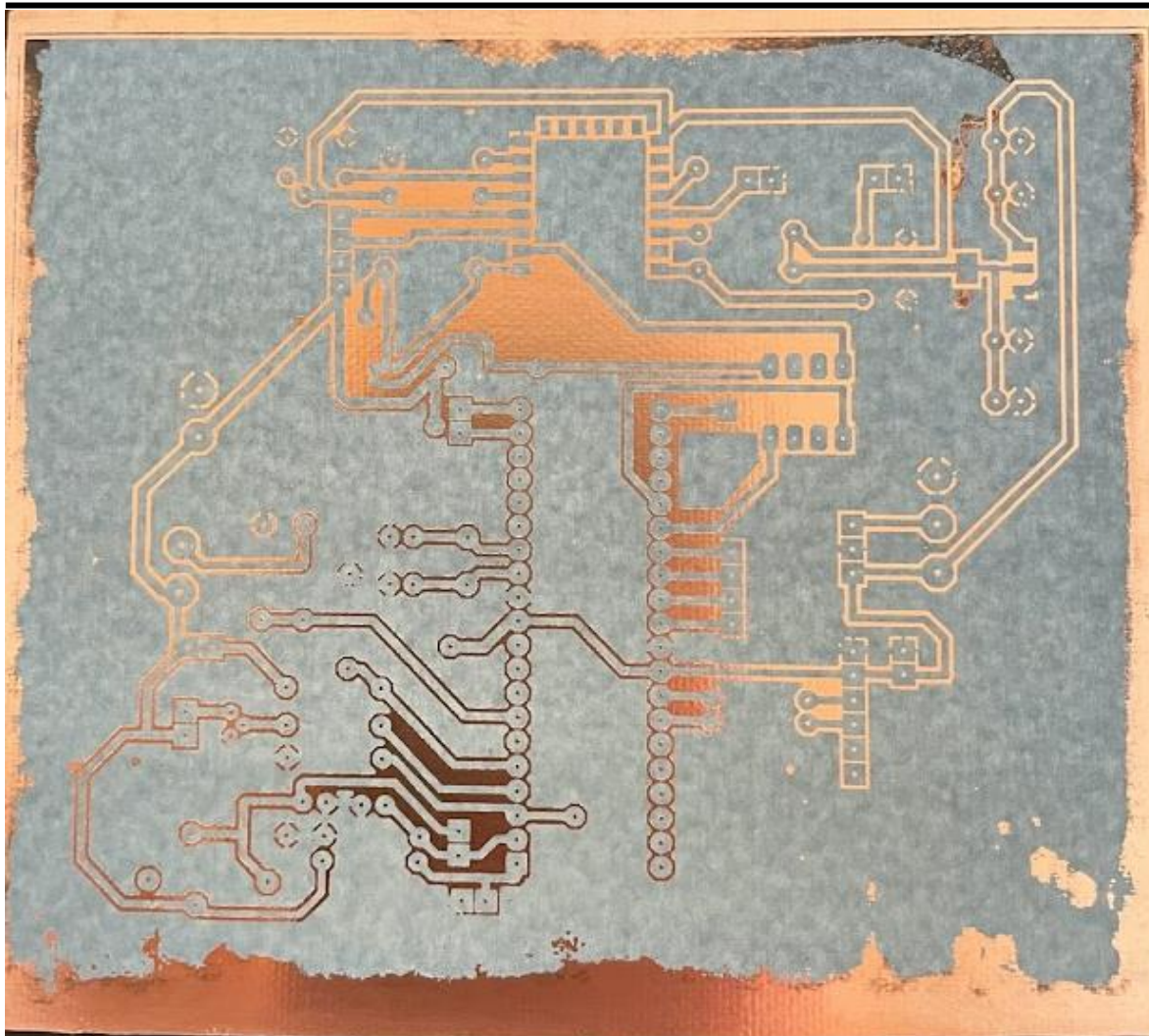


Fig. N° 33 *Layout transferido a la placa de cobre.*

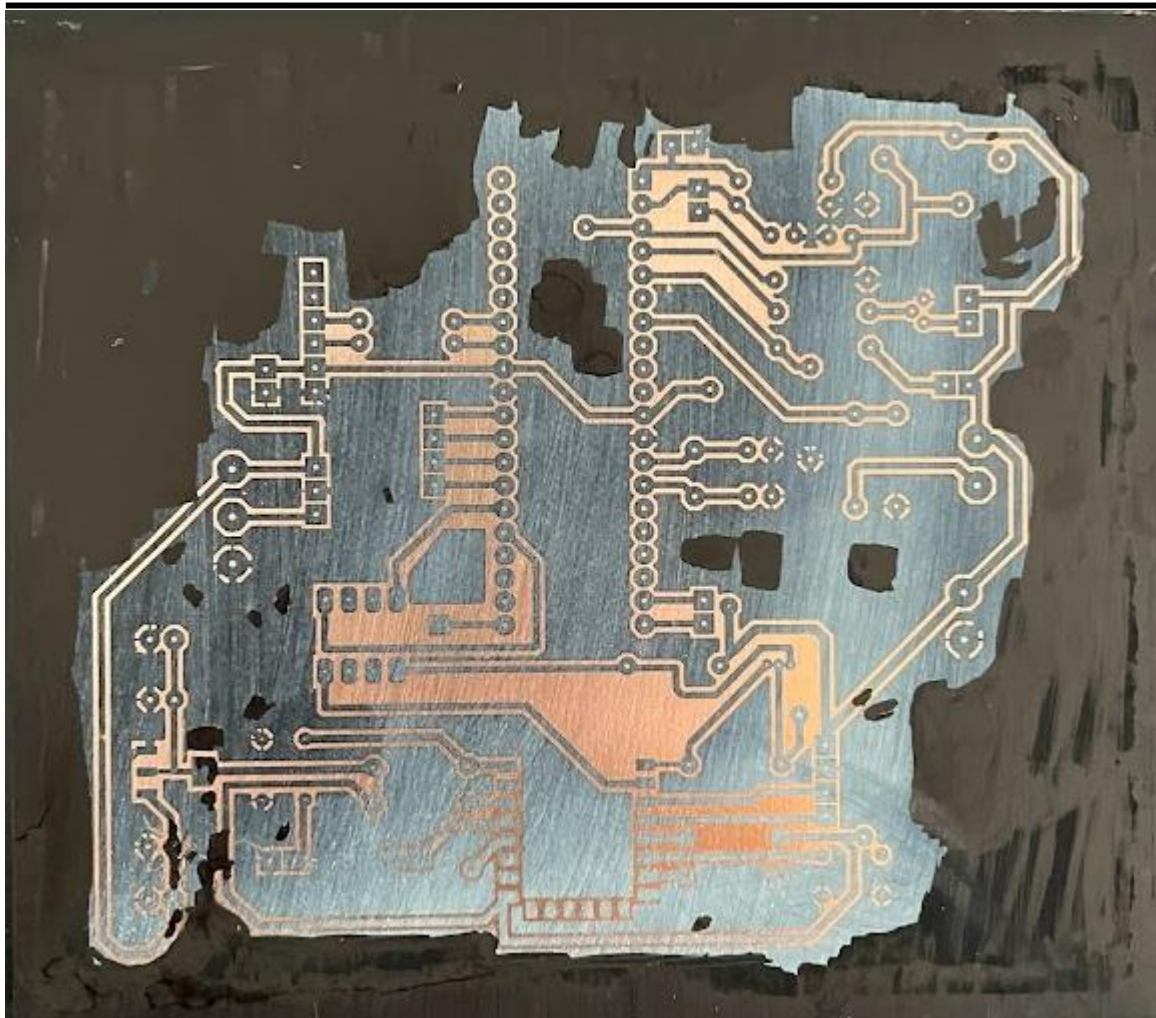


Fig. N° 34 Correcciones por una transferencia incorrecta.

Luego de realizar las correspondientes correcciones se procede a introducir la pcb en cloruro férrico (FeCl_3). El cloruro férrico es corrosivo para metales ferrosos y no ferrosos como el cobre, esto permite que al sumergir la placa en el cloruro este último corroerá todas partes de la pcb que no estén protegidas por la tinta.

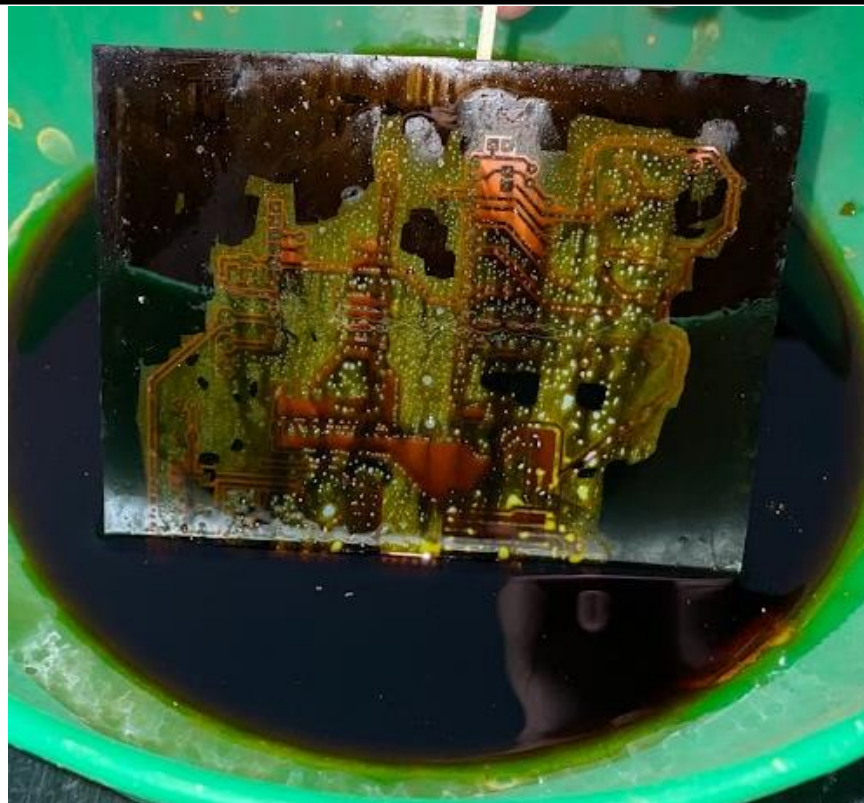


Fig. N° 35 PCB sumergida en cloruro férrico.

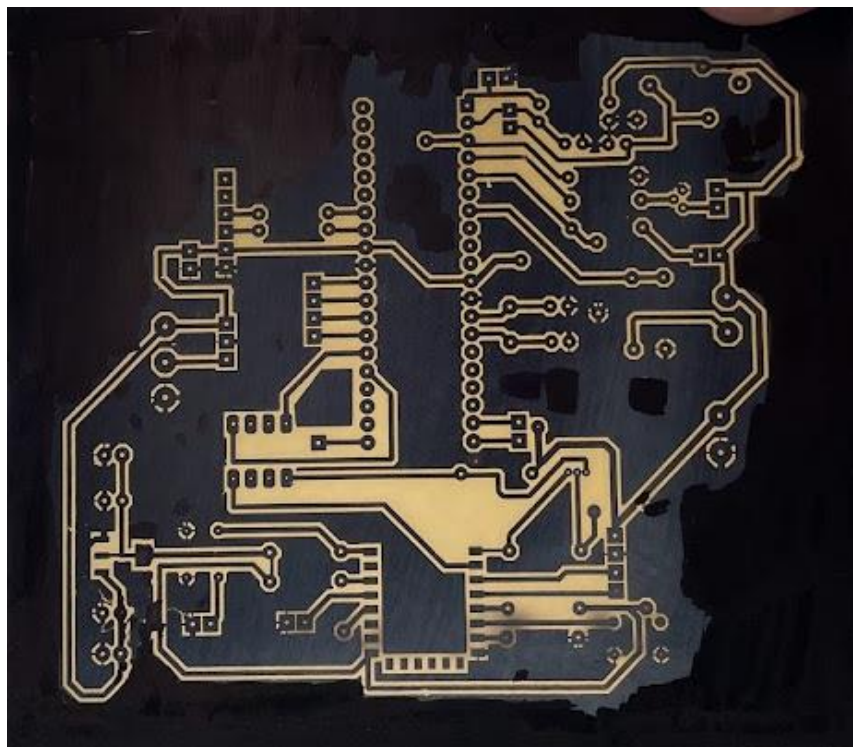


Fig. N° 36 PCB después de que el cloruro férrico actúe sobre el cobre.

Posterior a quitar la tinta de la PCB mediante una esponja de acero se procede a realizar los agujeros. En este caso la totalidad de la placa se agujereó con un minitorno.

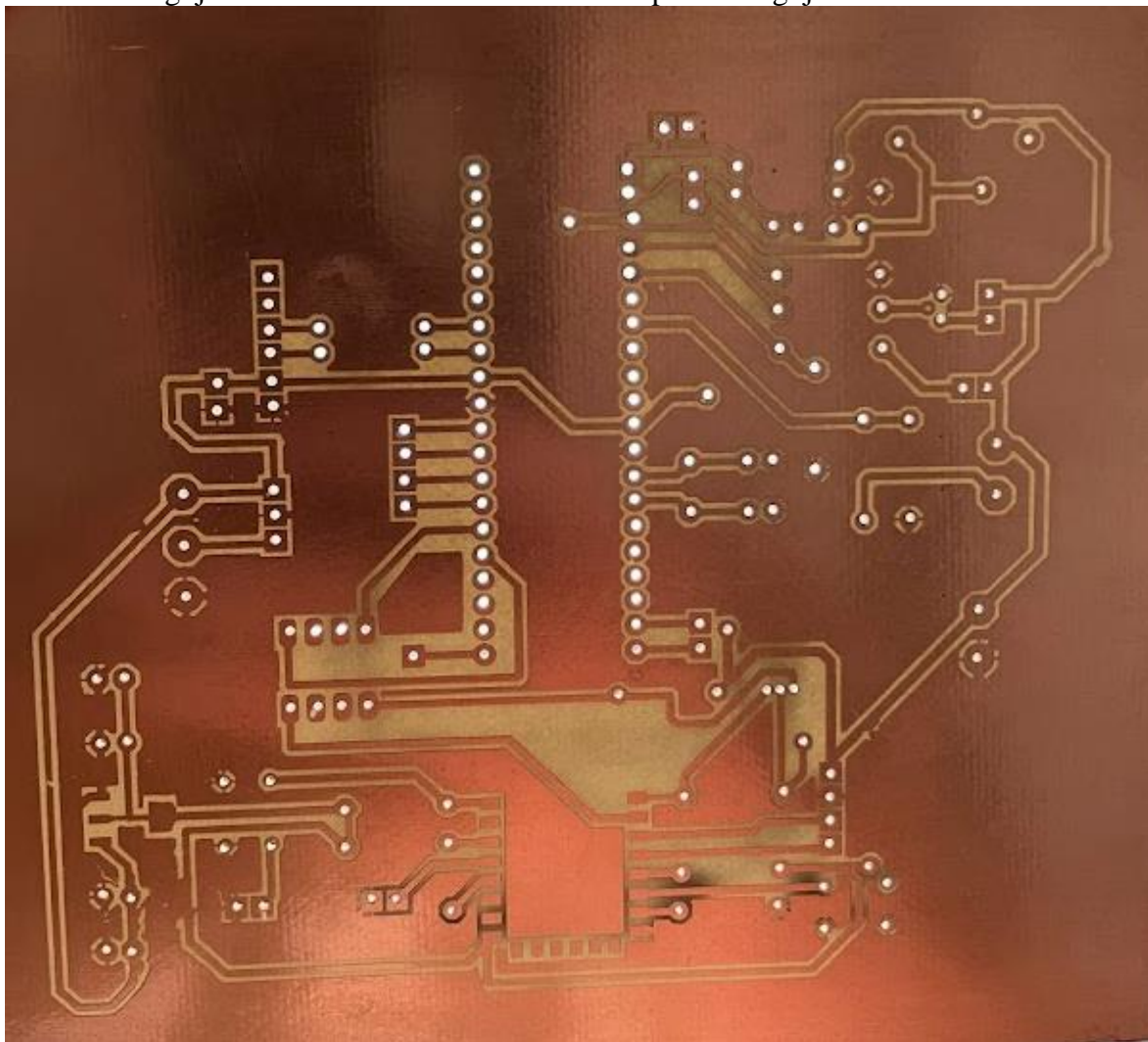


Fig. N° 37 PCB lista para montaje.

Montaje y ensayo real del prototipo

En primera instancia se colocaron y soldaron los componentes referidos a la parte WiFi, es decir, ESP8266, regulador de tensión de 3.3[V] y demás componentes necesarios para programar y comprobar que funciona todo de acuerdo a lo deseado.

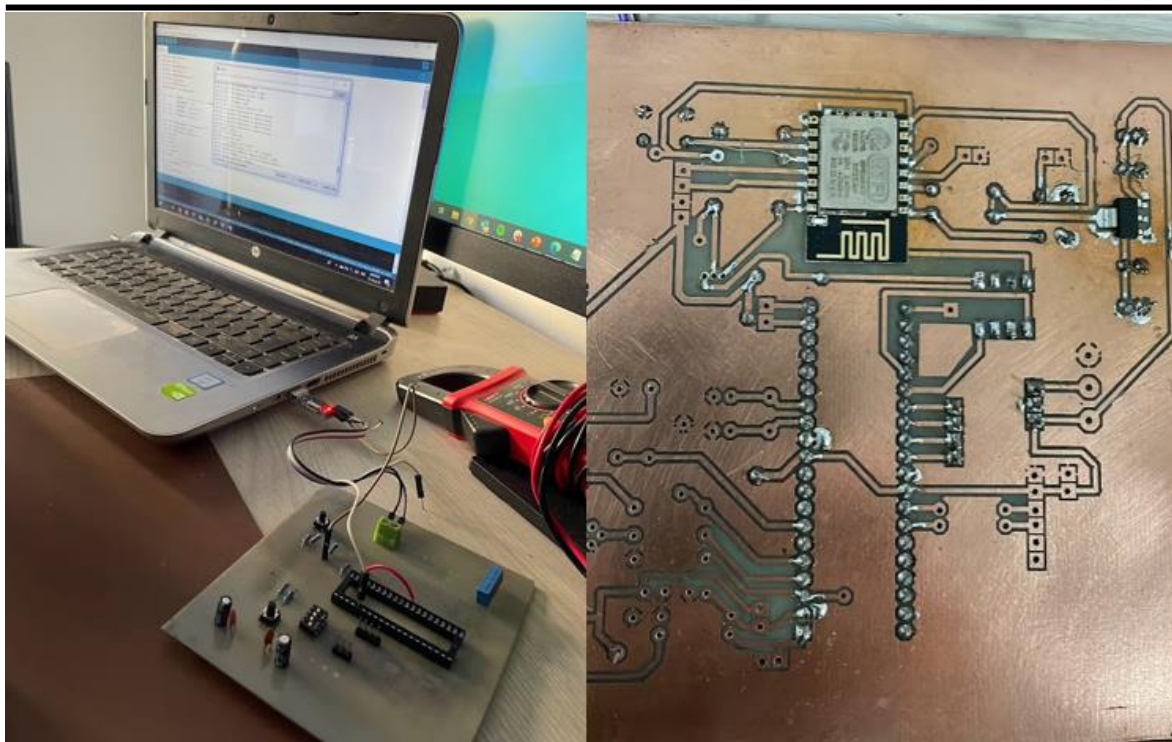


Fig. N° 38 Montaje y prueba de componentes referentes al funcionamiento WiFi.

Para cargar el programa a la ESP8266 se utiliza el circuito de la figura 39. En la placa se dejaron los lugares necesarios para poder programar la esp una vez soldada.

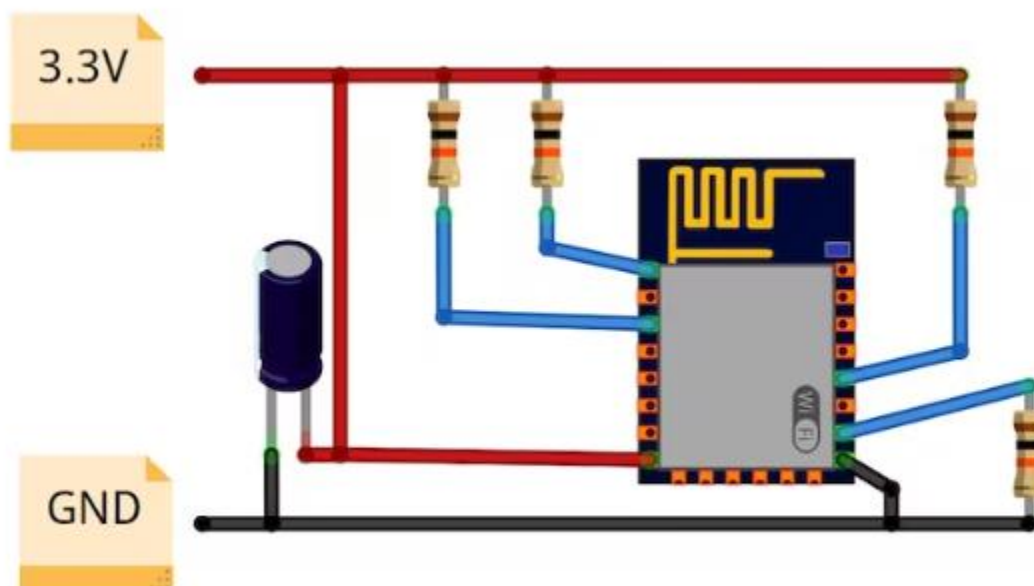


Fig. N° 39 Circuito necesario para cargar programa en la ESP8266.

Una vez comprobado que las conexiones y el programa funcionan de manera correcta se sigue por los demás componentes del proyecto. Cabe aclarar que no se montaron todos los

componentes de una sola vez, sino que se fueron incorporando uno a uno y se fueron realizando las respectivas pruebas para facilitar el debuggeo o corrección de errores. Como no posee sentido alguno colocar las imágenes con cada componente por separado se va a poder observar a continuación en la figura 40 el circuito final ya montado.

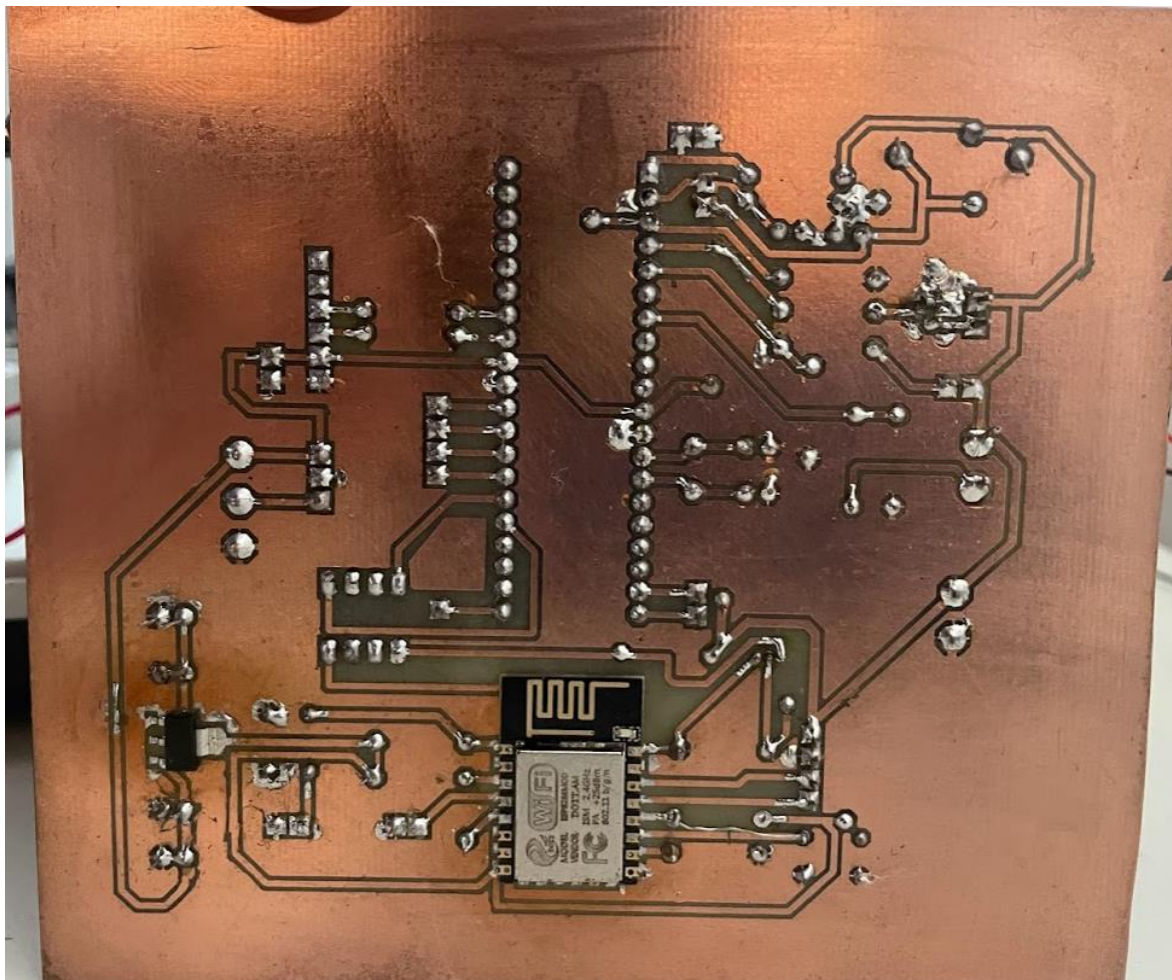


Fig. N° 40 *Montaje final del circuito vista posterior.*

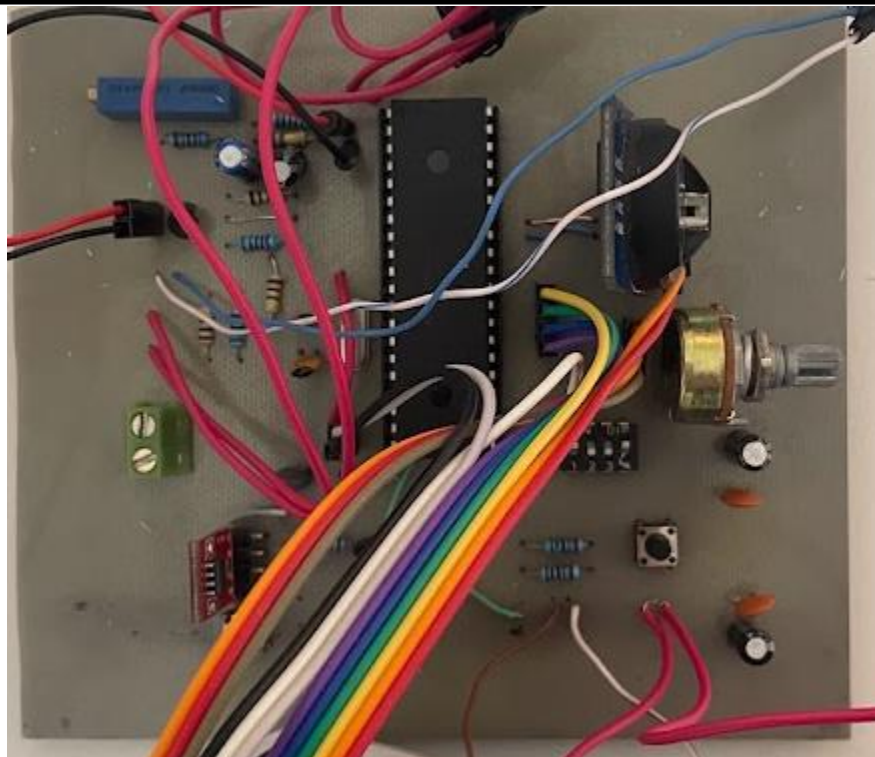


Fig. N° 41 Montaje final del circuito vista superior.

La carga del programa en el PIC18F4550 se realiza a través del circuito que se puede observar en la figura 42. A diferencia de la ESP el PIC se programa en una protoboard, es decir, fuera del circuito final. Cabe aclarar que esto es posible porque en el montaje final el PIC se coloca sobre un zócalo DIP40.

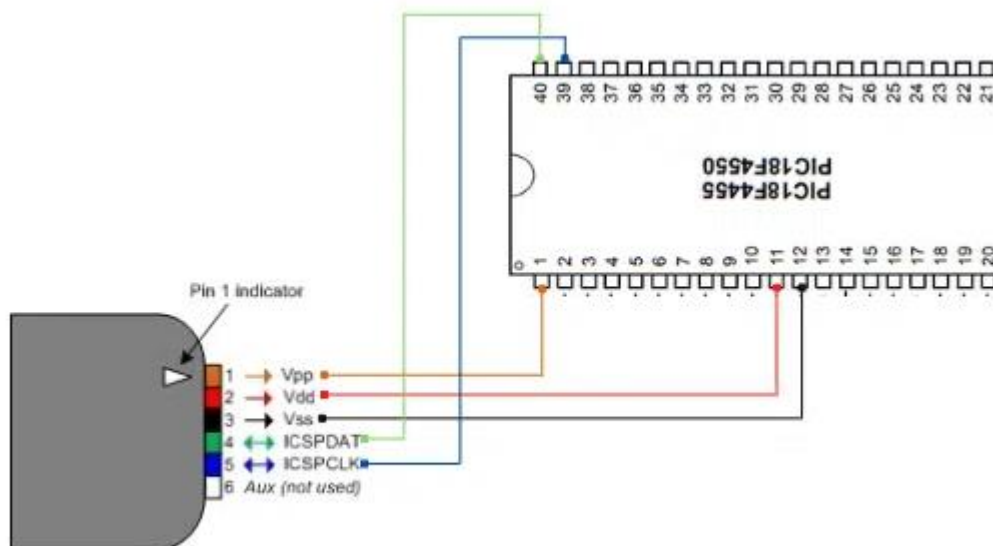


Fig. N° 42 Circuito necesario para cargar programa en la PIC18F4550.

Presentación final del prototipo

El montaje final del prototipo se ha realizado sobre una caja estanca cuyas dimensiones son 15cm de ancho, 15cm de alto y 7,5cm de profundidad. La caja se modificó para poder colocar la pantalla, el led y los pulsadores.



Fig. N° 43 *Prototipo final.*

En la figura 44 se podrá observar el prototipo en funcionamiento y el paso previo a la conexión de la ESP al WiFi, es decir, en el smartphone se pueden ver las redes disponibles al alcance de esta y se decide a cuál se conectará.



Fig. N° 44 conexión a red WiFi.

Una vez elegida y configurada la red Wifi se podrá apreciar la correcta conexión de la ESP a través del led indicador en la parte superior del prototipo. Cuando esta esté encendida se procede a abrir la APK.



Fig. N° 45 *conexión correcta a la red WiFi.*

En la figura 46 se puede observar el funcionamiento del prototipo con la APK en el que el tiempo de realizar un mantenimiento coincide con el tiempo de funcionamiento y dispararon las alarmas tanto visuales como sonoras que no pueden ser apreciadas en la imagen.



Fig. N° 46 Funcionamiento de prototipo.



Conclusiones

Es un proyecto muy interesante debido a que no existe algo igual en el mercado, lo que resulto un desafío en el diseño y la toma de decisiones. Además, por su gran versatilidad se puede introducir en distintos tipos de industrias permitiéndole mayor alcance.

Al no ser un producto disponible en el mercado se tuvo que idear todo el proyecto y su utilidad desde la experiencia adquirida en la PPS (prácticas profesionales supervisadas) y de la información disponible sobre el mantenimiento en industrias. Cabe destacar que sin importar que fuera un producto no comercial, se pudieron conseguir de manera simple y sin grandes costos todos los materiales para llegar al prototipo final.

Se puede decir que realizarlo involucra conocimientos de varias ramas de la carrera, como ser teoría de transformadores, teoría de los circuitos, protocolos de comunicación entre tantos otros. Con respecto a los protocolos de comunicación se amplió el tema al ser necesario sumergirse en el mundo de IoT donde se ha utilizado el protocolo MQTT no visto con anterioridad pero que resulta ser muy beneficioso en la actualidad.

Por ultimo resulta importante reconocer todos los conocimientos que se han adquirido a la hora de realizar este proyecto, ya que no solo es la información plasmada a lo largo de este documento, sino también la experiencia y todos aquellos datos y circuitos que fueron visto para llegar hasta la mejor solución. Todo esto nos permite saber que este proyecto puede escalar aún mucho más en complejidad con componentes profesionales, mejoras mecánicas y de software, pero también nos permite saber que se puede realizar un producto que funcione de manera correcta con componentes que están al alcance de la mano y con los conocimientos adquiridos durante la carrera.

Anexo I

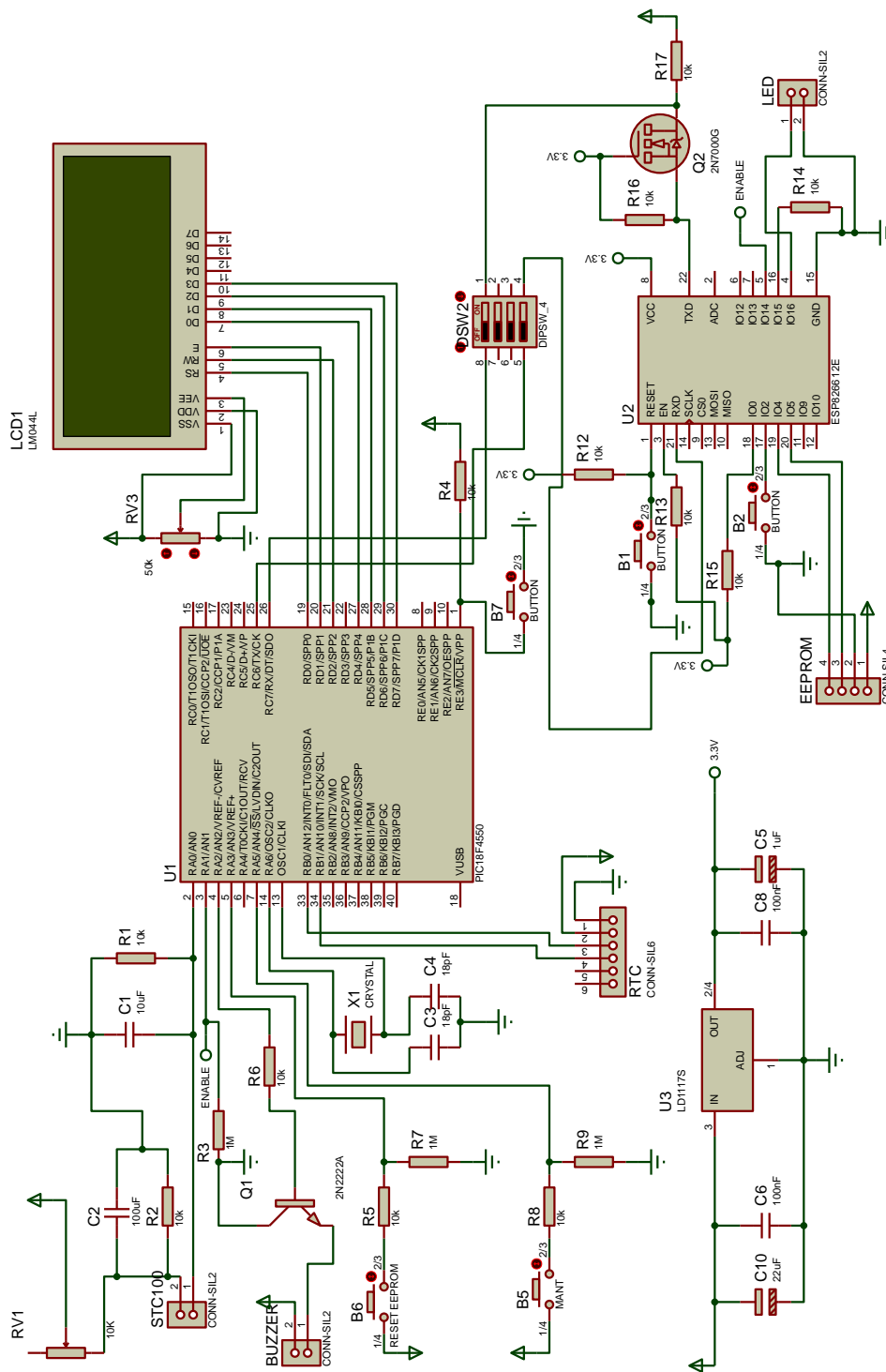


Fig. N° 47 Esquemático horómetro WiFi programable.



Anexo II

Como se mencionó en el proyecto, el STC-013 funciona como un transformador por lo que necesita colocarse alrededor de una sola fase de alimentación, es por ello que se realizó una prolongación para lograr esto. Si el STC-013 se colocara alrededor del neutro y la fase se tendría una medición nula dado que los campos magnéticos de estos se anularían entre si y el transformador no tendría una señal para tomar.



Fig. N° 48 Prolongación necesaria para medir la corriente.



Anexo III

Se han añadido etiquetas para identificación de pulsadores.



Fig. N° 49 *Vista lateral derecha.*



Fig. N° 50 *Vista superior.*

Anexo IV

Primeras pruebas de la carcasa con el hardware para asegurar las correctas dimensiones y corregir errores.



Fig. N° 51 Prueba de prototipo.



Bibliografía

- [1] DALLAS Semiconductor. (Marzo de 2010). *Datasheet DS3231*. Obtenido de ALLDATASHEET.COM: <https://pdf1.alldatasheet.com/datasheet-pdf/view/336170/MAXIM/DS3231.html>, rev:08/10/2021
 - [2] Floyd, T. L. (2006). *Fundamentos de sistemas*. Madrid: PEARSON EDUCACIÓN S.A.
 - [3] Fundación Endesa. (2022). *El transformador eléctrico*. Obtenido de endesa fundación: <https://www.fundacionendesa.org/es/educacion/endesa-educar/recursos/corrientes-alternas-con-un-transformador-electrico#:~:text=Un%20transformador%20el%C3%A9ctrico%20es%20una,caso%20de%20un%20transformador%20ideal>, rev:25/06/2022
 - [4] geändert, Z. (10 de Julio de 2020). *Extensión AI2 MQTT*. Obtenido de ullisroboterseite: <https://ullisroboterseite.de/android-AI2-MQTT-en.html>, rev:13/01/2022
 - [5] henzen Anxinke Technology CO. (2015). *ESP12E ESP8266*. Obtenido de ALLDATASHEET.COM: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1179099/ETC2/ESP12E.html>, rev:16/08/2021
 - [6] LLamas, L. (16 de MAYO de 2016). *EL BUS I2C EN ARDUINO*. Obtenido de LUIS LLAMAS: <https://www.luisllamas.es/arduino-i2c/>, rev:20/12/2021
 - [7] Llamas, L. (24 de Enero de 2017). *SENSOR DE CORRIENTE ELÉCTRICA NO INVASIVO CON ARDUINO Y SCT-013*. Obtenido de LUIS LLAMAS: <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>, rev:12/09/2021
 - [8] Lough, B. (3 de Noviembre de 2018). *3 Simple Ways of Programming an ESP8266 12X Module*. Obtenido de hackster.io: <https://www.hackster.io/brian-lough/3-simple-ways-of-programming-an-esp8266-12x-module-c514ee>, rev:15/05/2022
 - [9] *MICRO controladores*. (s.f.). Obtenido de microcontroladores.com: <https://microcontroladores.com/pic18f4550/>, rev:21/07/2021
 - [10] Patel, D. (28 de Abril de 2020). *Level Shifter*. Obtenido de MAKER PRO: <https://maker.pro/custom/tutorial/how-to-use-a-logic-level-shifter-circuit-for-components-with-different-voltages>, rev:27/12/2021
 - [11] *PIC Microcontroller Projects*. (30 de Agosto de 2017). Obtenido de
-



<http://ccspicc.blogspot.com/>: <http://ccspicc.blogspot.com/2017/08/pic18f4550-ds3231-rtcc.html>, *rev:03/02/2022*

[12] Sec Electronic Inc. (10 de Noviembre de 2015). *Datasheet 2N7000*. Obtenido de ALLDATASHEET.COM: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1115779/SECELECTRONICS/2N7000.html>, *rev:11/01/2022*

[13] Turbo IC, Inc. (20 de Octubre de 2001). *Datasheet 24C08*. Obtenido de ALLDATASHEET.COM: <https://pdf1.alldatasheet.com/datasheet-pdf/view/115409/ETC1/24C08.html>, *rev:09/11/2021*