



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES

TRABAJO DE GRADUACIÓN:

**”DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE
MICROCONTROLADORES PARA COMUNICACIÓN WI-FI”**

AUTOR(ES):

JUAN CARLOS ANILEMA GUADALUPE

CRISTHIAN MAURICIO YAGOS VILEMA

DIRECTOR:

ING. FABIAN GUNSHA

RIOBAMBA-ECUADOR

AÑO 2015

INFORME DEL TUTOR

Yo Ing. FABIÁN CELSO GUNSHA MAJI, en mi calidad de Tutor, del trabajo investigativo titulado: DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE MICROCONTROLADORES PARA COMUNICACIÓN WI-FI, tengo a bien informar que el mencionado trabajo cumple con los requisitos exigidos para ser expuesto al público, luego de ser evaluado por el tribunal designado.

Riobamba, Diciembre del 2015

Atentamente


Ing. Fabián Gunsha.


TUTOR

Los miembros del Tribunal de Graduación del proyecto de investigación: DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE MICROCONTROLADORES PARA COMUNICACIÓN WI-FI, presentado por Juan Carlos Anilema Guadalupe, Cristhian Mauricio Yagos Vilema y dirigida por Ing. Fabián Celso Gunsha Maji.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente por uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Carlos Peñafiel
Presidente del Tribunal



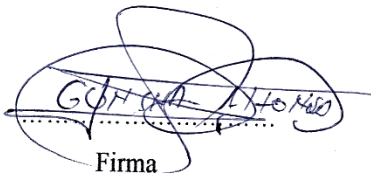
Firma

Ing. Fabián Gunsha
Director del Proyecto



Firma

Ing. Alfonso Gunsha
Miembro del Tribunal



Firma

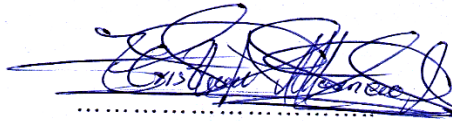
AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido del Proyecto de Graduación, corresponde exclusivamente a: Juan Carlos Anilema Guadalupe, Cristhian Mauricio Yagos Vilema y Director del Proyecto: Ing. Fabián Gunsha; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

Autores:



Juan Carlos Anilema Guadalupe
CI: 060389156-5



Cristhian Mauricio Yagos Vilema
CI: 060398964-1

AGRADECIMIENTO I

Agradezco a mis padres que han sido un pilar de fortaleza para siempre seguir adelante, a mi familia por todo el apoyo durante mi vida universitaria, a los profesores de la Universidad Nacional de Chimborazo por brindarme sus conocimientos, su guía para mi formación académica.

A Lilia Caguana por sus consejos, su apoyo incondicional por sus recomendaciones por su tiempo y guía en la realización del proyecto.

A mi compañero de tesis por sus aportes para la culminación de este trabajo de investigación.

Juan Anilema

DEDICATORIA I

El presente trabajo de investigación lo dedico a Dios por darme bendiciones a mi vida, a mis padres Carmen y José por su apoyo y consejos han sabido guiar mi vida; a mis hermanos: José, María, Blanca y Andrés.

Y dedicatoria especial A Lilia y Andrea son mi inspiración para seguir adelante y nunca rendirse.

Juan Anilema

AGRADECIMIENTO II

Agradezco de manera más sincera a la Universidad Nacional de Chimborazo, a todos los docentes de la escuela de Ingeniería Electrónica y Telecomunicaciones, Tutor de tesis, amigos, compañeros, familiares, compañero de Tesis, a mi madre Concepción que hicieron posible la culminación del tema de investigación, también por su apoyo y confianza que han brindado con cada uno de sus conocimientos y experiencias que hicieron posible la formación de una investigación del tema que se desarrolló.

Cristhian Yagos

DEDICATORIA II

A Dios, por la salud para poder superar todas las adversidades que se presentaron, también por su infinito amor y bondad, a mi madre Concepción por haberme apoyado y por la confianza que me brindo, para María Alexandra, José Ricardo, José David, para toda mi familia en especial para mis Tías Magdalena y Margarita.

Cristhian Yagos

INDICE

| | |
|---------------------------------|------|
| AGRADECIMIENTO I | v |
| DEDICATORIA I | vi |
| AGRADECIMIENTO II | vi |
| DEDICATORIA II | viii |
| INDICE | i |
| INDICE DE GRÁFICOS | iv |
| INDICE DE TABLAS | vi |
| RESUMEN | vii |
| SUMMARY | viii |
| INTRODUCCIÓN | ix |

CAPÍTULO I

FUNDAENTACION TEORICA

| | |
|---|----|
| 1.1. MICROCONTROLADOR | 1 |
| 1.2. MICROCONTROLADOR 16F877A | 2 |
| 1.2.1. RECURSOS PERIFÉRICOS | 3 |
| 1.3. ENTRENADOR DE MICROCONTROLADORES | 3 |
| 1.4. REDES INALÁMBRICAS | 4 |
| 1.5. WLAN..... | 4 |
| 1.5.1. ESTÁNDARES..... | 4 |
| 1.5.2. Wi-Fi..... | 5 |
| 1.5.3. CANALES EN 802.11(WIFI) | 6 |
| 1.5.4. RE-UTILIZACIÓN DE FRECUENCIA EN LOS ACCESS POINT | 8 |
| 1.6. MÓDULO WIFLY | 9 |
| 1.6.1 MÓDULO WIFLY ENTRADA DE COMANDOS | 10 |
| 1.6.2. CONFIGURACIÓN REMOTA UTILIZANDO MODO ADHOC..... | 11 |
| 1.6.3. MÓDULO WIFLY COMO DISPOSITIVO TERMINAL | 12 |
| 1.6.4. MÓDULO WIFLY COMO PUNTO DE ACCESO | 13 |
| 1.7. LM35 (SENSOR DE TEMPERATURA)..... | 14 |
| 1.8. DTH11 (SENSOR DE HUMEDAD Y TEMPERATURA) | 15 |
| 1.9. MQ-6 (SENSOR DE GAS GLP) | 16 |
| 1.10. VISUAL STUDIO | 17 |
| 1.10.1. NET FRAMEWORK 4..... | 18 |
| 1.10.2. VISUAL C# | 18 |

CAPÍTULO II

METODOLOGÍA

| | |
|--|----|
| 2.1. TIPO DE ESTUDIO | 19 |
| 2.2 POBLACIÓN Y MUESTRA..... | 19 |
| 2.3. OPERACIONALIZACIÓN DE VARIABLES | 20 |
| 2.4. PROCEDIMIENTOS..... | 21 |

| | |
|--|----|
| 2.5. DISEÑO DE UN SISTEMA DE MONITERIO DE LABOTORIO MEDIANTE EL ENTRANADOR WIFI | 33 |
| 2.5.1. TOPOLOGIA DE LA RED | 33 |
| 2.5.2. MÓDULO WIFLY SENSOR DE HUMEDAD Y TEMPERATURA..... | 34 |
| 2.5.3. MÓDULO WIFLY SENSOR LM35 (TEMPERATURA)..... | 38 |
| 2.5.4. MÓDULO WIFLY SENSOR DE GAS (MQ6)..... | 41 |
| 2.5.5. SOCKETS CONEXIÓN SOCKET TCP/IP: | 44 |
| 2.6. TRANSMISIÓN DE DATOS DE SENSORES DTH11, LM35, MQ6 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES EN UNA APLICACIÓN EN LA PC ... | 47 |
| 2.6.1 CONFIGURACIÓN MÓDULO WIFLY COMO PUNTO DE ACCESO..... | 47 |
| 2.6.2. TRANSMISION DE DATOS DEL SENSOR DTH11 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES A UNA APLICACIÓN EN LA PC..... | 48 |
| 2.6.3. TRANSMISIÓN DE DATOS DEL SENSOR LM35 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES A UNA APLICACIÓN EN LA PC..... | 50 |
| 2.6.4. TRANSMISION DE DATOS DEL SENSOR MQ6 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES A UNA APLICACIÓN EN LA PC..... | 51 |
| 2.7. PROCESAMIENTO Y ANÁLISIS..... | 55 |
| 2.7.1. ANÁLISIS DEL DISPOSITIVO WIFLY. | 56 |
| 2.7.2. ANÁLISIS PRACTICO DE COMUNICACIÓN WI-FI..... | 57 |
| 2.7.3. TÉCNICAS DE PROCEDIMIENTO PARA EL ANÁLISIS | 59 |
| 2.8. COMPROBACIÓN DE HIPÓTESIS | 59 |
| 2.8.1. PLANTEAMIENTO DE LA HIPOTESIS. | 59 |
| 2.8.2. ESTABLECIMIENTO DEL NIVEL DE SIGNIFICANCIA..... | 59 |
| 2.8.3. ELECCION DE LA PRUEBA DE HIPOTESIS..... | 60 |
| 2.8.4. MUESTRA. | 60 |
| 2.8.5. MEDIA | 60 |
| 2.8.6. PRUEBA DE NORMALIDAD..... | 61 |
| 2.8.7. PRUEBA T PARA MUESTRAS RELACIONADAS | 62 |
| CAPÍTULO III | |
| RESULTADOS | |
| 3.1. PRUEBAS DE CONECCCIÓN | 64 |
| 3.2. PRUEBAS DE ALCANCE DE COBERTURA WIFLY MODULO RNXV-171 | 65 |
| 3.3. ANÁLISIS FINANCIERO | 69 |
| CAPÍTULO IV | |
| CONCLUSIONES Y RECOMENDACIONES | |
| 4.1. CONCLUSIONES..... | 70 |
| 4.2. RECOMENDACIONES..... | 71 |
| CAPÍTULO V | |
| PROPUESTA | |
| 5.1. TÍTULO DE LA PROPUESTA..... | 72 |
| 5.2. INTRODUCCIÓN | 72 |
| 5.3. DISCUSIÓN | 72 |

CAPÍTULO VI

OBJETIVOS

| | |
|---|----|
| 6.1. OBJETIVO GENERAL..... | 74 |
| 6.2. OBJETIVOS ESPECÍFICOS..... | 74 |
| 6.3. FUNDAMENTACIÓN CIENTÍFICO –TÉCNICA | 74 |
| 6.4. DESCRIPCIÓN DE LA PROPUESTA..... | 75 |
| 6.5. DISEÑO ORGANIZACIONAL..... | 75 |

CAPITULO VII

BIBLIOGRAFÍA

ANEXOS

ANEXO A

| | |
|--|----|
| GLOSARIO DE TERMINOS..... | 78 |
| PLACA RELE DE ESTADO SOLIDO Y CRUCE POR CERO | 79 |
| PLACA MODULO WIFLY RN-XV 171..... | 81 |
| PLACA DISPLAYS Y LEDS | 83 |
| ENTRENADOR DE MICROCONTROLADORES APLICADO EN EL AREA DE LAS TELECOMUNICACIONES..... | 85 |

ANEXO B

| | |
|---|-----|
| LABORATORIO 1: CONFIGURACIÓN DEL MÓDULO WIFLY..... | 86 |
| LABORATORIO 2: CONFIGURACION MODULOS WIFLY COMO AP Y COMO TERMINALES DE LA RED..... | 92 |
| LABORATORIO 3: PROGRAMACION DE LOS MICRONTROLADORES PARA SENSORES: DTH11, LM35, MQ6..... | 96 |
| LABORATORIO 4: DESARROLLO DE APLICACIÓN EN VISUAL ESTUDIO PARA RECEPCION Y CONTROL DE CARGAS DE UN LABORATORIO | 111 |

ANEXO C

| | |
|------------------------|-----|
| MANUAL DE USUARIO..... | 122 |
|------------------------|-----|

ANEXO D

| | |
|-----------------|-----|
| DATASHEET | 130 |
|-----------------|-----|

INDICE DE GRÁFICOS

| | |
|---|----|
| Figura N°1: Microcontrolador | 2 |
| Figura N°2: Pines PIC 16F877A | 2 |
| Figura N°3: Entrenador de microcontroladores | 3 |
| Figura N°4: Tasas de transmision estandares mas utilizados | 5 |
| Figura N°5: Canales wifi | 6 |
| Figura N°6: Canales Wi-Fi no interferirles y ancho de banda..... | 7 |
| Figura N°7: Re-utilizacion de frecuencias de AP | 8 |
| Figura N°8: Módulo WIFLY | 9 |
| Figura N°9: Pines de alimentación y TX, RX módulo Wifly..... | 10 |
| Figura N°10: Modo Adhoc | 11 |
| Figura N°11: Escaneo de redes inalámbricas..... | 12 |
| Figura N°12: Módulo Wifly como dispositivo final..... | 13 |
| Figura N°13: Modulo Wifly como AP | 14 |
| Figura N°14: sensor de temperatura LM35 | 15 |
| Figura N°15: Sensor Dth11..... | 15 |
| Figura N°16: Sensor de gas MQ-6..... | 16 |
| Figura N°17: Conexión MQ-6 | 16 |
| Figura N°18: IDE entorno Visual Studio 2012..... | 17 |
| Figura N°19: Diagrama Relé de Estado Solido | 21 |
| Figura N°20: Diagrama Relé de Estado Solido – Red Snubber..... | 22 |
| Figura N°21: Diagrama Análisis Relé de Estado Solido | 23 |
| Figura N°22: Circuito Cruce por Cero..... | 25 |
| Figura N°23: Grafica Salida del osciloscopio del cruce por cero..... | 26 |
| Figura N°24: Diseño en el Ares de la placa Relé de estado Sólido y Cruce por Cero | 28 |
| Figura N°25: Placa Relé de estado Sólido y Cruce por Cero en 3D..... | 28 |
| Figura N°26: Calculo de la resistencia del diodo led..... | 29 |
| Figura N°27: Ruteo en Ares del circuito de Leds y Display..... | 30 |
| Figura N°28: Placa de Leds y Display vista en 3D..... | 30 |
| Figura N°29: Diseño de placa del circuito módulo Wifly | 31 |
| Figura N°30: Ruteo de la palca módulo Wifly | 32 |
| Figura N°31: Placa módulo WIFLY RN-XV 171 vista en 3D | 32 |
| Figura N°32: Diagrama de conexión de la práctica transmisión de datos de sensores a través del módulo Wifly. | 33 |
| Figura N°33: Circuito microcontrolador con sensor DTH11 | 34 |
| Figura N°34: Diagrama de programación de microcontrolador con DTH11 | 35 |
| Figura N°35: Diagrama de programación de microcontrolador con DTH11 parte de interrupción | 36 |
| Figura N°36: Microcontrolador con sensor de temperatura LM35 | 38 |
| Figura N°37: Diagrama de flujo de la programación LM35..... | 39 |
| Figura N°38: Diagrama de flujo de la programación LM35 la parte de interrupción | 40 |
| Figura N°39: Circuito de microcontrolador con sensor MQ6 | 41 |
| Figura N°40: Diagrama de flujo de la programación del sensor MQ6..... | 42 |
| Figura N°41: Diagrama de flujo de la programación MQ6 parte interrupción..... | 43 |
| Figura N°42: Funcionamiento básico de sockets cliente-servidor..... | 44 |
| Figura N°43: Diagrama de flujo programación del Servidor en C# Visual estudio 2012 | 45 |
| Figura N°44: Diagrama de flujo procesos del cliente de la aplicación en Visual Estudio | 46 |
| Figura N°45: Aplicación – 3 sensores y activacion de cargas | 47 |

| | |
|---|----|
| Figura N°46: Circuito del sensor DTH11 | 48 |
| Figura N°47: relé de estado sólido ventilador. | 49 |
| Figura N°48: relé de estado sólido calefactor | 49 |
| Figura N°49: Circuito sensor lm35 | 50 |
| Figura N°50: Circuito sensor lm35 | 51 |
| Figura N°51: Circuito sensor MQ6 – relé de estado sólido sirena | 52 |
| Figura N°52: aplicación a espera de recibir datos de sensores | 53 |
| Figura N°53: aplicación muestra ip de los módulos conectados | 54 |
| Figura N°54: funcionamiento de los botones de la aplicación | 54 |
| Figura N°55: Gráfica de la fuerza de la señal de modulo wifly como AP en el tempo | 57 |
| Figura N°56: Grafica del canal y potencia de la red Tesis | 58 |
| Figura N°57: Grafica de medición de la potencia en dBm red Tesis..... | 58 |
| Figura N°58: Redes analizadas y lectura de datos | 59 |
| Figura N°59: Aceptación de H1 y rechazo de H0..... | 63 |
| Figura N°60: Prueba de error en cmd transmision del modulo wifly | 64 |
| Figura N°61 : Prueba de transmision de paquetes al darle un ping al módulo wifly..... | 65 |
| Figura N°62: prueba correcta de envio de paquetes | 65 |
| Figura N°63: Red TESIS modulo wifly como AP | 66 |
| Figura N°64: conexión de un dispositivo al módulo Wifly exitoso | 66 |
| Figura N°65: Prueba de conexión con el ap(modulo wifly) en interior..... | 67 |
| Figura N°66: Prueba de conexión con el ap(modulo wifly) en exterior | 68 |
| Figura N°67: diseño organizacional del proyecto..... | 75 |

INDICE DE TABLAS

| | |
|--|----|
| Tabla N°1: Componentes especiales de un microcontrolador | 1 |
| Tabla N°2: Recursos Periféricos | 3 |
| Tabla N°3: Lista de canales Wi-Fi..... | 7 |
| Tabla N°4: Canales no interferirles | 8 |
| Tabla N°5: Características del módulo Wifly | 10 |
| Tabla N°6: Comandos para AP (Access Point) | 13 |
| Tabla N°7: operacionalización de variables..... | 20 |
| Tabla N°8: Tiempos tomados de estudiantes para comprobación de hipótesis | 60 |
| Tabla N°9: Pruebas de Normalidad en spss | 61 |
| Tabla N°10: tabla de resultados de normalidad. | 61 |
| Tabla N°11: Estadísticas de muestras relacionadas | 62 |
| Tabla N°12: Correlaciones de muestras relacionadas..... | 62 |
| Tabla N°13: Prueba de muestras relacionadas..... | 62 |
| Tabla N°14: Alcance promedio modulo wifly en interiores y exteriores | 68 |
| Tabla N°15: Análisis Financiero..... | 69 |

RESUMEN

El presente proyecto de investigación detalla el diseño e implementación de un entrenador de microcontroladores con en el módulo de comunicación WiFi (WiFly RN-XV 171) para la transmisión y recepción de datos demostrando diferentes aplicaciones para los microcontroladores en el área de las telecomunicaciones.

El módulo WiFly es un dispositivo electrónico que puede ser configurado para trabajar en modo ad-hoc, AP (Punto de Acceso), dispositivo terminal y para crear redes wlan; los comando de configuración se describen en el presente documento.

Con los estudiantes que cursan la materia de microcontroladores se experimentó la elaboración de prácticas usando el entrenador de microcontroladores, demostrando la practicidad y eficiencia en el tiempo para desarrollo de prácticas.

Se implementó una red wlan usando un dispositivo WiFly como Access Point y tres como dispositivos terminales para transmitir los datos de sensores (dth11, mq6, lm35) usando microcontroladores a una aplicación en la computadora para monitorear y activar cargas de un laboratorio, comprobando conexión y comunicación con el uart de mikroC y Teraterm, para la fiabilidad en la transmisión de datos en la comunicación WiFi.

Para facilitar el desarrollo e implementación de prácticas usando el entrenador se deja un manual para los estudiantes de la escuela de Electrónica y Telecomunicaciones.

SUMMARY



Universidad Nacional de Chimborazo
Facultad de Ingeniería
Centro de Idiomas



Lic. Eduardo Heredia

17 de Diciembre del 2015

SUMMARY

The present research explains in detail the design and implementation of a trainer of microcontrollers with the communication module WiFi (WiFly RN-XV 171) for the transmission and reception of data, by demonstrating different application for microcontrollers in the area of telecommunications. The module WiFly is an electronic device which can be set to operate in mode ad-hoc, AP (Access point), final device and to create wlan webs; the configuration commands are described in the present document. It was experimented with the students who pursue the subject of microcontrollers the elaboration of practices by utilizing the trainer of microcontrollers, and demonstrating practicality as well as efficiency in the time for the development of praxis.

It was implemented a wlan web by using one WiFly device as Access Point and three as terminal devices for transmitting the sensors data (dth11, mq6, lm35) using microcontrollers to an application in the computer to monitor and activate loads of a laboratory, by testing out the connection and communication with the uart of mikroC and Teraterm, for the reliability in the transmission of data in WiFi communication. It is proposed a methodological manual for the students of electronics and telecommunications career in order to facilitate the development and implementation of practices by using the trainer.



INTRODUCCIÓN

El entrenador de microcontroladores para comunicación WiFi se caracteriza por tener un conjunto de circuitos electrónicos de alta funcionalidad, convergente con dispositivos de comunicaciones y de arquitectura abierta para el desarrollo de prácticas y proyectos en el campo de la electrónica.

La comunicación WiFi Wireless Fidelity (Fidelidad Inalámbrica), es el sistema descrito por la norma IEEE 802.11 para realizar comunicaciones inalámbricas de área local. Esta norma es utilizada ampliamente en dispositivos portátiles y permite velocidades de 2 Mbps, 11 Mbps y 54 Mbps.

La comunicación WiFi es muy utilizada en distintas aplicaciones de monitoreo, transmisión, comunicación, control, etc. El módulo Wifly es un dispositivo para transmitir y recibir datos inalámbricos teniendo el aire como medio de transmisión, tiene una tarjeta con pila TCP/IP y aplicaciones, solo con 4 pines de conexión (Power, Tx, Rx y Tierra) para poder comunicarse, el módulo Wifly trabaja con el protocolo 802.11 b/g transmitiendo en una frecuencia de 2.4 GHz.

El presente trabajo de investigación se realiza por la falta de entrenadores de microcontroladores en el área de las telecomunicaciones, tomando en cuenta que la falta de información sobre el módulo Wifly es poca para el aprendizaje de los estudiantes y la enseñanza práctica de los docentes, en nuestra investigación dejamos información muy detallada del funcionamiento y configuración del módulo Wifly que servirá como complemento de estudio y práctica de comunicación WiFi aplicada con los microcontroladores.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. MICROCONTROLADOR

Es un circuito integrado que tiene la capacidad de ser programado y ejecutar todas las instrucciones que fueron grabadas en su memoria con el mínimo error.

Las partes básicas de un microcontrolador son las siguientes:

- Memoria ROM (Memoria de solo lectura).
- Memoria RAM (Memoria de acceso aleatorio).
- Líneas de entrada / salida (I/O).
- Lógica de control. (Coordina la interacción entre los demás bloques).

Componentes especiales de un microcontrolador

| | |
|-----------------------------------|--|
| Convertor Analógico Digital (A/D) | Permiten medir señales no digitales o sensores que emitan señales analógicas. |
| Comunicación Serial | Permite establecer comunicación con otro microcontrolador o computadora a través de la interfaz serial RS-232. |
| Memoria EEPROM | Los datos no se alteren cuando se retira la alimentación, es un tipo de memoria ROM que puede programar o borrar eléctricamente. |
| PWM | Modulación por ancho de pulso |
| Interrupciones | Cuando se requiere atender eventos en tiempo real o se tienen procesos que no esperan |

Tabla N°1: Componentes especiales de un microcontrolador
Fuente: (ANÍBAL, 2007)

En la **figura N°1** se visualiza los componentes especiales que forman parte de un microcontrolador.

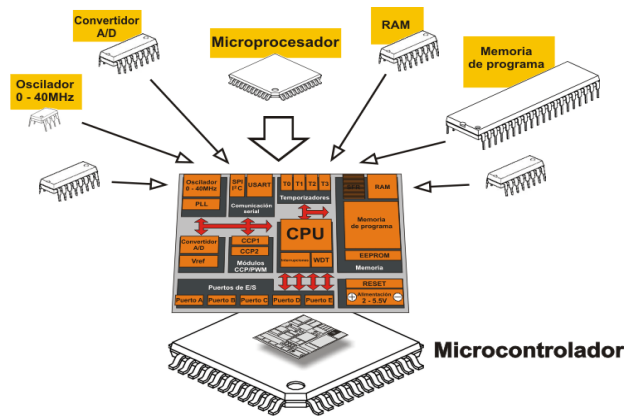


Figura N°1: Microcontrolador

Fuente: <http://www.mikroe.com/chapters/view/79/capitulo-1-el-mundo-de-los-microcontroladores/>

1.2. MICROCONTROLADOR 16F877A

Un microcontrolador de la familia Microchip dispone de todos los componentes más importantes que la mayoría de microcontroladores, en la **figura N°2** se indican los pines de configuración del microcontrolador.

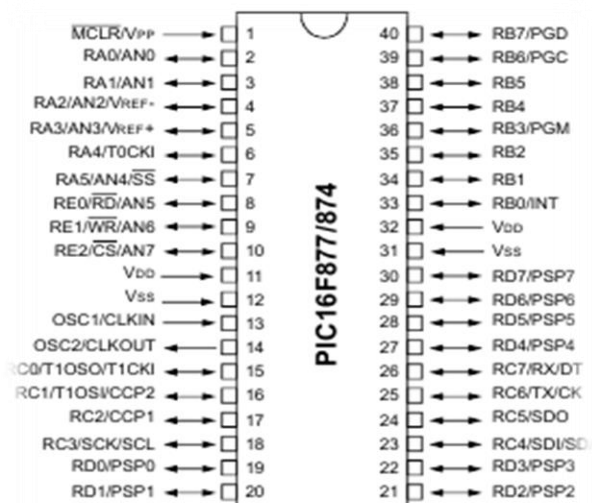


Figura N°2: Pines PIC 16F877A

Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>

A continuación se muestran las características generales más importantes para el microcontrolador 16F877A:

- Procesador de arquitectura RISC avanzada.
- Hay 35 instrucciones con 14 bits de longitud.
- Hasta 8K palabras de 14 bits de memoria de programa tipo FLASH.
- Hasta 368 Bytes de memoria de datos RAM.
- Hasta 256 Bytes de memoria de datos EEPROM.

1.2.1. RECURSOS PERIFÉRICOS

| | |
|---------------|---|
| Timer0 | Temporizador (contador de 8 bits con prescaler de 8 bits). |
| Timer1 | Temporizador (contador de 16 bits con registro de periodo), puede incrementarse de modo sleep de forma externa por un cristal |
| Timer2 | Temporizador (contador de 8 bits) con registro de periodo |
| PWM | Dos módulos de Captura, Comparación |
| USART/SCI | Universal Synchronous Asynchronous Receiver Transmitter) con 9 bits |
| SSP, SPI, I2C | Puerto Serie Síncrono (SSP) con SPI (Modo maestro) e I2C (Master/Slave) |

Tabla N°2: Recursos Periféricos
Fuente: (microchip, 2001-2013).

1.3. ENTRENADOR DE MICROCONTROLADORES

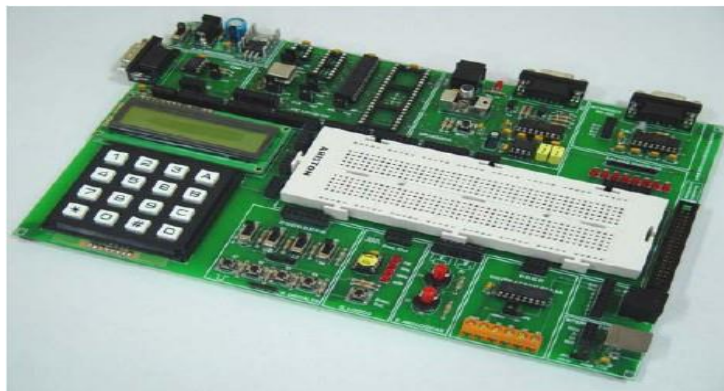


Figura N°3: Entrenador de microcontroladores
Fuente: <http://perso.wanadoo.es/pictob/practica.htm>

El entrenador de microcontroladores para comunicaciones también es conocido como Microtrainers, el "circuito entrenador" es un dispositivo electrónico que sirve como apoyo en el funcionamiento del microcontrolador. El diseñador es el encargado de adaptar el modelo a la configuración del circuito, es decir, establece las conexiones físicas entre los dispositivos de comunicación y el microcontrolador (server-die-alc, 2002).

El entrenador de microcontroladores para comunicaciones está compuesto por múltiples circuitos y dispositivos electrónicos.

1.4. REDES INALÁMBRICAS

Las redes inalámbricas son capaces de transmitir y recibir datos o información usando como medio de transmisión el aire y cada dispositivo terminal de la red se pueden conectar sin necesidad de usar cable, la estandarización IEEE 802.11 se presentó en 1997 con tasas de datos superiores a los 2 Mbps, y en la actualidad se usan IEEE802.11 b /g/ n.

1.5. WLAN

“Una LAN inalámbrica WLAN (Wireless Local Área Network), el usuario con su dispositivo terminal móvil puede conectarse a una LAN a través de enlaces de radiofrecuencia sin la necesidad de cable, entonces han surgido varias versiones del estándar” (CYBERCOM, 2013).

1.5.1. ESTÁNDARES

802.11a Estándar WLAN de alta velocidad en la banda de los 5 GHz soporta hasta 54 Mbps. Utiliza el método de modulación OFMD (Múltiplexación por división de frecuencias ortogonales), en transmisiones exteriores hay un alcance de 30mts a 300mts y en interiores de 12mts a 90mts, entre mayor distancia menor velocidad.

802.11b Estándar WLAN para la banda de 2.4 GHz soporta 11 Mbps. Utiliza el método de modulación DSSS (Modulación de frecuencias directas del espectro extendido).

802.11g Establece una técnica de modulación adicional para la banda de los 2.4 GHz. Dirigido a proporcionar velocidades de hasta 54Mbps. El método de modulación que utiliza es el OFMD (Multiplexacion por división de frecuencias octogonales) y también DSSS (Modulación de frecuencias directas del espectro extendido).

802.11n Proporciona mejoras de mayor capacidad de proceso, se pretende que la proporción de velocidades es de 500Mbps (Caceres, 2013).

En la **Figura 4** se muestra un resumen de los estandares mas utilizados con la velocidad de transmision y banda no licenciada.

| Estándar | Velocidad máxima | Frecuencia | Compatibilidad con versiones anteriores |
|----------|------------------|-----------------|---|
| 802.11a | 54 Mbps | 5 GHz | No |
| 802.11b | 11 Mbps | 2,4 GHz | No |
| 802.11g | 54 Mbps | 2,4 GHz | 802.11b |
| 802.11n | 600 Mbps | 2,4 GHz o 5 GHz | 802.11b/g |

Figura N°4: Tasas de transmision estandares mas utilizados
Fuente: <http://es.slideshare.net/DavidNarvez/acceso-a-la-red>

1.5.2. Wi-Fi

Es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11.

Los estándares IEEE 802.11 b/g disfrutan de una aceptación internacional debido a que la banda de 2.4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbps y 54 Mbps. Existe también el estándar IEEE 802.11n que está en desarrollo y trabaja a 2.4 GHz a una velocidad de 108 Mbps. (Caceres, 2013).

1.5.2.1. CONCEPTOS BÁSICOS DE WI-FI

- **Punto de Acceso (AP/PA):** Es un dispositivo que ejerce básicamente funciones de puente entre una red Ethernet cableada y una red con Wi-Fi.
- **SSID (Service Set Identification):** El identificador suele emplearse en las redes Wireless creadas con infraestructura. Se trata de un conjunto de servicios que agrupan todas las conexiones de los clientes en un solo canal.
- **Roaming:** Propiedad de las redes Wi-Fi por la cual los clientes pueden estar en movimiento e ir cambiando de punto de acceso de acuerdo a la potencia de la señal. (wificlub, 2010)
- **DHCP:** (Dynamic Host Configuration Protocol) Es un protocolo para que cada host en una red reciban ip de forma dinámica.

1.5.3. CANALES EN 802.11(WIFI)

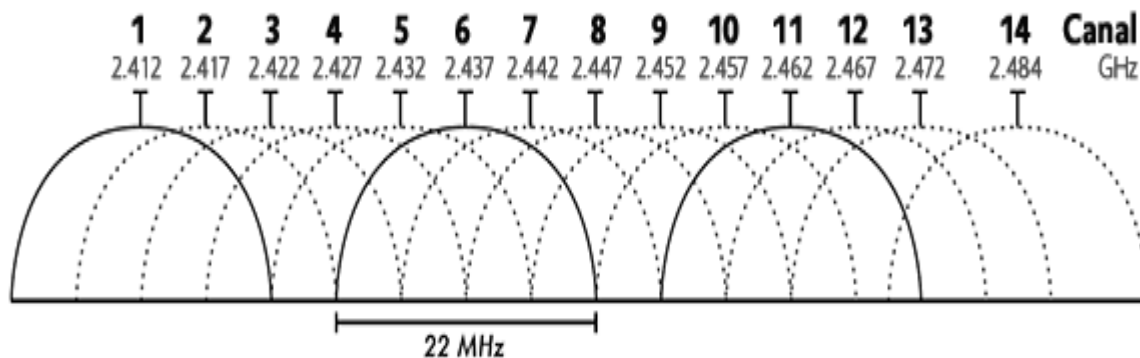


Figura N°5: Canales wifi

Fuente: <http://www.emezeta.com/articulos/internet-mas-rapido>

La comunicación WiFi (Fidelidad Inalambrica) se establece a través de 14 canales y cada uno ocupa 22MHz (Mega Hertz) de ancho de banda como se especifica en la **figura N°5** y la descripción se detalla en la **Tabla 3**.

| Banda | Frecuencia | Canal |
|--------|------------|-------|
| 2.4GHz | 2412.0 MHz | 1 |
| 2.4GHz | 2417.0 MHz | 2 |
| 2.4GHz | 2422.0 MHz | 3 |
| 2.4GHz | 2427.0 MHz | 4 |
| 2.4GHz | 2432.0 MHz | 5 |
| 2.4GHz | 2437.0 MHz | 6 |
| 2.4GHz | 2442.0 MHz | 7 |
| 2.4GHz | 2447.0 MHz | 8 |
| 2.4GHz | 2452.0 MHz | 9 |
| 2.4GHz | 2457.0 MHz | 10 |
| 2.4GHz | 2462.0 MHz | 11 |
| 2.4GHz | 2467.0 MHz | 12 |
| 2.4GHz | 2472.0 MHz | 13 |
| 2.4GHz | 2484.0 MHz | 14 |

Tabla N°3: Lista de canales Wi-Fi

Fuente: <http://www.dd-wrt.espanol.com/canal-inalambrico-recomendado/>

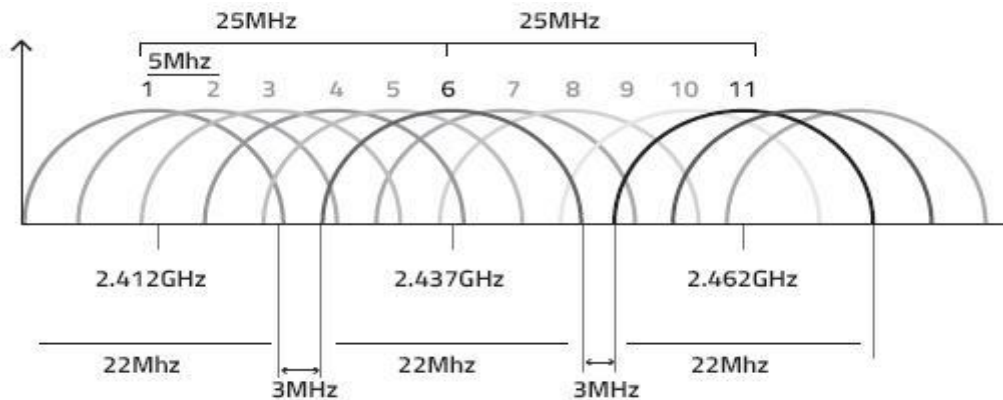


Figura N°6: Canales Wi-Fi no interferirles y ancho de banda

Fuente: http://www.scielo.org.co/scielo.php?pid=S0124-81702013000200001&script=sci_arttext

Por ejemplo los canales 1, 6, y 11 no se solapan. Esto se explica con más detalles en la clase de uso comparativo del espectro no-licenciado como se muestra en la Tabla N°4.

El estándar IEEE 802.11 b/g permite tres canales no interferentes espaciados a 3MHz . En la **Figura N°6** se muestra la disposición de canales y el espaciado entre ellos y se resalta los canales no interferibles los cuales no tienen solapamiento.

Los canales están separados cada 3 MHz, pero las señales 802.11 ocupan 22MHz. Para evitar interferencias se deben escoger canales que no se solapen, es decir que las respectivas señales no se superpongan en ninguna parte del espectro.

| Canal no interferible | Posibles interferencias |
|-----------------------|-------------------------|
| 1 | 2,3,4,5 |
| 6 | 2,3,4,5,7,8,9,10 |
| 11 | 7,8,9,10,12,13,14 |

Tabla N°4: Canales no interferirles

Fuente: http://www.scielo.org.co/scielo.php?pid=S0124-81702013000200001&script=sci_arttext

1.5.4. RE-UTILIZACIÓN DE FRECUENCIA EN LOS ACCESS POINT

En la **Figura N°7** se indica la selección de canales para los AP para que no se solapen . Si los AP se ubican cuidadosamente, se puede cubrir un campus arbitrariamente grande utilizando únicamente tres canales sin interferencia entre canales.

En la práctica el AP nunca es un círculo perfecto de cobertura, hay que tomar en cuenta los problemas topológicos de extender la red en tres dimensiones cuando se trate de un edificio de varios pisos.

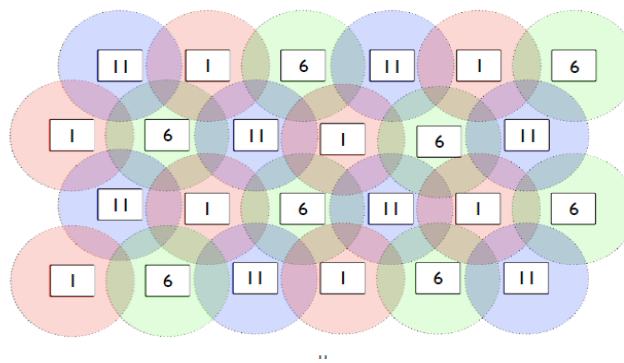


Figura N°7: Re-utilizacion de frecuencias de AP

Fuente: <https://es.scribd.com/doc/212345456/05-Introduccion-a-Las-Redes-WiFi-Es-v2-3-Notes>

1.6. MÓDULO WIFLY



Figura N°8: Módulo WIFLY

Fuente: http://asrob.uc3m.es/index.php/Tutorial_Wifly

El módulo RN-XV de Roving Networks de la **Figura N°8**, es una solución Wi-Fi especialmente diseñada para los usuarios que quieran migrar su actual red 802.15.4 tipo XBee a una red TCP/IP Wi-Fi, no tienen la necesidad de modificar nada de su estructura. Está basado en el robusto módulo RN-171 de Roving Networks e incorpora un emisor 802.11 b/g, un procesador interno de 32 bits, pila TCP/IP, RTC, gestión de energía e interfaz analógica para sensores externos. Viene precargado con el firmware original de Roving Networks para simplificar la integración y minimizar el tiempo de desarrollo, para la configuración más simple, sólo necesita de 4 pines para funcionar (PWR, TX, RX y GND) como se muestra en la **Figura N°9**, y crear una conexión inalámbrica Wifi. (RN-WIFLYCR-UG, 2013)

Características:

- Basado en 802.15.4.
- Muy bajo consumo: 4uA (sleep), 38mA (activo).
- El stack TCP/IP interno soporta: DHCP, UDP, DNS, ARP, ICMP, cliente HTTP, cliente FTP y cliente TCP.
- Potencia configurable por software: 0dBm a 12dBm.
- Interfaz: TTL UART.

- Velocidad de datos: Hasta 464Kbps port UART.
- Soporta redes Adhoc e infrastructure.
- 8 pines I/O.
- Entradas analógicas para sensores.
- RTC para time-stamping, auto-sleep y auto-wakeup.
- Alimentación: 3.3VDC.

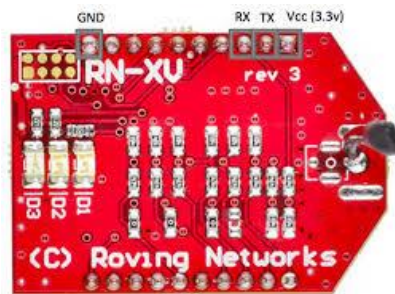


Figura N°9: Pines de alimentación y TX, RX módulo Wifly
Fuente: <https://github.com/lifegraph/hw-tutorial>

Las características del módulo Wifly se detallan en la **Tabla N°5**:

| Radio Characteristics | |
|------------------------------------|--|
| Parameter | Specifications |
| Frequency | 2402 ~ 2480MHz |
| Modulation | DSSS(CCK-11, CCK-5.5, DQPSK-2, DBPSK-1) |
| Channel intervals | 5MHz |
| Channels | 1-14 |
| Transmission rate | (over the air) 1 - 11Mbps for 802.11b / 6 - 54Mbps for 802.11g |
| Receive sensitivity | -85dBm typ. |
| Output level | (Class1) +18dBm |
| Maximum RF input to U.FL connector | 10 dBm |

Tabla N°5: Características del módulo Wifly
Fuente: http://www.mouser.com/new/microchip/roving_surf_board/

1.6.1 MÓDULO WIFLY ENTRADA DE COMANDOS

El módulo Wifly tiene dos formas de configuración, modo de datos y modo comando, en modo datos el modulo tiene la característica de aceptar conexiones entrantes o iniciar con conexiones externas, siempre al encenderse el dispositivo estará en modo de datos, con la siguiente

configuración es para modo comando se presiona \$\$\$ y el dispositivo responderá con CMD indicando que está en modo Comandos.

El SSID (Service Set Identification), Canal, Dirección IP, Opciones de Puerto Serial, y todas las opciones pueden ser vistas y configuradas en el modo Comando, la mayor parte de los comandos validos retornaran con una respuesta AOK, y algunos inválidos retornaran con una descripción EER (error), y para salir de modo comando se escribe “**exit**”.

1.6.2. CONFIGURACIÓN REMOTA UTILIZANDO MODO ADHOC

Usando modo Adhoc el módulo elimina la necesidad de asociarse a un punto de acceso o una red, para esta configuración el módulo crea la solicitud de red para asociarse a una computadora disponible, ver **figura N°10**.

Para activar el modo Adhoc vía hardware se alimenta con (3.3V) el PIO9, el módulo Wifly y crea una red Adhoc con lo siguiente:

- SSID: Wifly-GSX-XX, donde XX son los dos bytes finales de la dirección MAC del dispositivo.
- Canal: 1.
- DHCP: apagado.
- Dirección IP: 169.254.1.1
- Mascara de Red: 255.255.0.0



Figura N°10: Modo Adhoc

Fuente: Autores

1.6.2.1. ESCANEADO DE ACCESS POINT PARA EL MODO ADHOC.

El módulo Wifly soporta el modo Adhoc y el modo infraestructura de red. Sin embargo, este no soporta ambos modos simultáneamente, el escaneo para redes inalámbricas es una función del modo infraestructura, el módulo Wifly tiene que deshabilitar el modo Adhoc y escanear.

Emitir el comando “**scan**” temporalmente deshabilita el modo Adhoc mientras el modulo este escaneando. Si se está conectado al módulo sobre telnet, el resultado del comando de escaneo es enviado sobre telnet y entonces el modo Adhoc es restaurado.

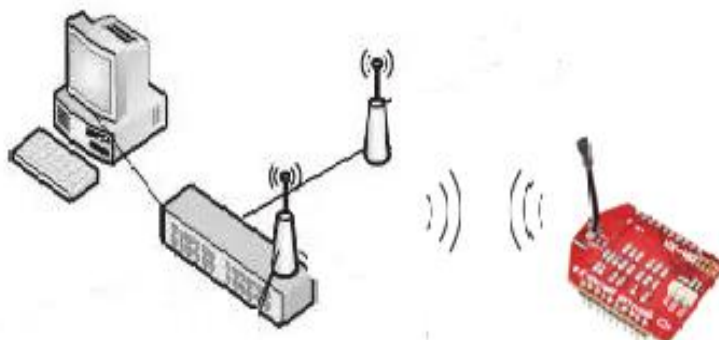


Figura N°11: Escaneo de redes inalámbricas

Fuente: Autores

1.6.3. MÓDULO WIFLY COMO DISPOSITIVO TERMINAL

El módulo WIFLY debe estar en modo comando, seguidamente se escanea las redes como se explica en el punto 1.6.2.1, para asociarle el módulo a una red abierta simplemente se escribe **join roving1**, si la seguridad del Punto de Acceso está habilitada se necesitara configurar la contraseña previa usando el comando **Join**.

Para configurar la contraseña por WPA (Wi-Fi Protected Access) use el comando **set wlan phrase<string>**. Para configuración WEP (Wired Equivalent Privacy) la llave **set wlan key<num>**. (RN-WIFLYCR-UG, 2013)

Una vez asociado a la red la SSID del Punto de Acceso será almacenada, con los anteriores comandos el módulo Wifly se asociara al Punto de Acceso cada vez que lo requiera.



Figura N°12: Módulo Wifly como dispositivo final
Fuente: Autores

1.6.4. MÓDULO WIFLY COMO PUNTO DE ACCESO

El módulo Wifly tiene la característica de poder trabajar como Punto de Acceso inalámbrico, se encarga de determinar en base a su configuración, que dispositivos están autorizados a acceder a la red, ver **figura N°13**.

Comandos para PA (Punto de Acceso)

| | |
|---------------------------------------|---|
| set wlan join <value> | Punto de Acceso. |
| set wlan channel <value> | Se debe configurar el canal de trabajo WLAN, del 1-13 |
| set wlan ssid <name> | (Nombre de la red) nombre de la red |
| set ip dhcp <value> | AP asigne direcciones ip por DHCP |
| set ip address X.X.X.X | ip del AP |
| set ip net X.X.X.X | maskara de la red |

Tabla N°6: Comandos para AP (Access Point)
Fuente: (RN-WIFLYCR-UG, 2013)



Figura N°13: Modulo Wifly como AP
Fuente: Autores

1.7. LM35 (SENSOR DE TEMPERATURA)

El sensor de temperatura tiene un circuito integrado de precisión ver en la **Figura N°14**, la salida de voltaje es linealmente proporcional a Celsius (grados centígrados) de la temperatura. El sensor LM35 tiene una ventaja sobre los sensores de temperatura lineal calibrada en ° Kelvin.

LM35 no requiere ningún circuito externo de calibración o recorte para proporcionar precisiones típicas de $\pm 1/4$ a temperatura ambiente y $\pm 3/4$ durante un total -55 a 150 °C rango de temperatura. (Instruments, 2015)

Características:

- Calibrada directamente in Celsius(Centigrados)
- Factor de escala lineal $+ 10\text{mV}/^{\circ}\text{C}$.
- Rango de -55°C a 150°C .
- Opera de 4V a 30V .
- Menos de $60\text{-}\mu\text{A}$ de consumo de corriente.

La comunicación con este sensor se realiza mediante un protocolo de un único hilo (protocolo 1-wire) que hace que la integración de este sensor en nuestros proyectos sea rápida y sencilla. (UK, 2010)

1.9. MQ-6 (SENSOR DE GAS GLP)

Sensor de uso de gas licuado de petróleo (GLP), apto para detectar concentraciones de GLP (compuesto principalmente propano y butano) en el aire. El MQ-6 puede detectar concentraciones de gas en cualquier lugar de 200 a 10000 ppm (partes por millón).

Este sensor tiene una alta sensibilidad y rápido tiempo de respuesta se puede observar en la **Figura N°16**. La salida del sensor es una resistencia analógica. El circuito de control es simple; todo tiene que hacer el poder de la bobina del calentador con 5V, añadir una resistencia de carga y conectar la salida de un ADC. (SENSORS, 2015)



Figura N°16: Sensor de gas MQ-6
Fuente: <https://www.sparkfun.com/products/9405>

En la **Figura N°17** se muestra la manera de conectar y alimentar el sensor de gas MQ6

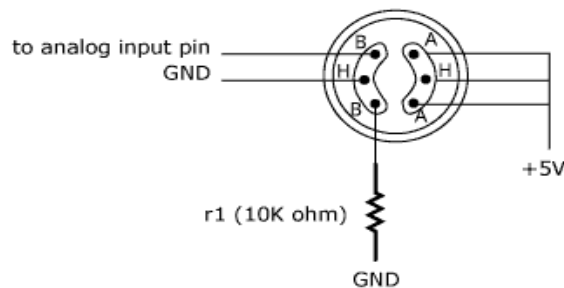


Figura N°17: Conexión MQ-6
Fuente: <http://wiring.org.co/learning/basics/airqualitymq135.html>

1.10. VISUAL STUDIO

Microsoft Visual Studio es un entorno (IDE, por sus siglas en inglés), para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django, etc. el IDE se muestra en la **Figura N°18**.

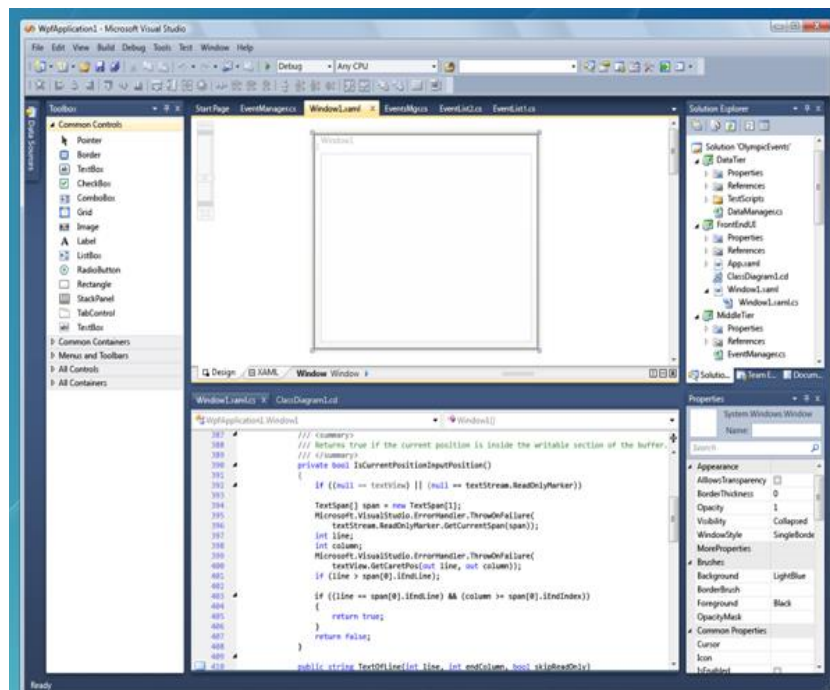


Figura N°18: IDE entorno Visual Studio 2012

Fuente: Autores

Un completo entorno de desarrollo integrado para crear aplicaciones espectaculares para Windows, Android e iOS, además de aplicaciones web y servicios de nube innovadores.

Cree aplicaciones multiplataforma para Windows, iOS, Android, etc.

Herramientas de desarrollo web, para móviles y de nube Compilar aplicaciones para Windows, Android e iOS.

- Diseñadores, editores, depuradores y generadores de perfiles integrados.

- Código de C#, C++, JavaScript, Python, TypeScript, Visual Basic, F#, etc.
- Capacidad de implementar, depurar y administrar servicios de Microsoft Azure.
- Miles de extensiones, desde PHP hasta juegos. (Microsoft, 2015)

1.10.1. NET FRAMEWORK 4

NET Framework es un componente integral de Windows que admite la compilación y la ejecución de la siguiente generación de aplicaciones y servicios Web. Los componentes clave de .NET Framework son Common Language Runtime (CLR) y la biblioteca de clases .NET Framework, que incluye ADO.NET, ASP.NET, formularios Windows Forms y Windows Presentation Foundation (WPF).

.NET Framework proporciona un entorno de ejecución administrado, un desarrollo e implementación simplificados y la integración con una gran variedad de lenguajes de programación. (msdn.microsoft.com, s.f.)

1.10.2. VISUAL C#

C# es un lenguaje de programación diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C.

Visual C# es una implementación del lenguaje C# de Microsoft. Visual Studio ofrece compatibilidad con Visual C# con un completo editor de código, un compilador, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y de fácil uso y otras herramientas. La biblioteca de clases de .NET Framework ofrece acceso a numerosos servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa. (msdn.microsoft.com, s.f.)

CAPÍTULO II

METODOLOGÍA

2.1. TIPO DE ESTUDIO

El presente estudio es orientado a la investigación de entrenadores de microcontroladores para protocolos de comunicación, usados en el ámbito de las telecomunicaciones.

Los tipos de estudio empleados en el presente proyecto de investigación son:

Cuantitativo: En la investigación del proyecto se realizó el armado de prácticas con los estudiantes sin el uso del entrenador y posteriormente usando el entrenador para determinar si el entrenador de microprocesadores cubre todas las expectativas en la parte de las telecomunicaciones.

Método Analítico: El método de estudio se fundamenta en el análisis de los dispositivos que conforman los módulos existentes en el entrenador para los diferentes protocolos de comunicaciones además de las placas que conforman el entrenador de microcontroladores.

Experimental: Se relaciona a los experimentos y aplicaciones desarrolladas con un enfoque científico para comprobar la comunicación entre microcontrolador y dispositivos wifi, usados para prácticas y aplicaciones con el entrenador de microcontroladores.

Observación: Aplicado a la práctica y utilización de los diferentes equipos en el diseño, para caracterizar el modelo del entrenador con sus distintos dispositivos.

2.2 POBLACIÓN Y MUESTRA

El proyecto se implementará para el laboratorio de microcontroladores de la Universidad Nacional de Chimborazo estará disponible para los estudiantes de la Escuela y aquellas personas que trabajan en aplicaciones y programas de microcontroladores.

2.3. OPERACIONALIZACIÓN DE VARIABLES

En la tabla N°7 se muestra la operacionalización de variables:

| VARIABLE | CONCEPTO | CATEGORÍA | INDICADOR | TÉCNICAS E INSTRUMENTO |
|---|--|--|---|--|
| <p>VARIABLE INDEPENDIENTE: El diseño e implementación de un entrenador de microcontrolador es para comunicación</p> | <p>Entrenador de microcontroladores</p> <p>El equipo entrenador desarrollado permite implementar fácilmente prácticas con microcontroladores. Es fácilmente transportable, están presentes todos los módulos y existe un área que está destinada al montaje de los circuitos experimentales a través de una placa protoboard para las conexiones entre los distintos componentes.</p> | <p>Desarrollo de Prácticas con Microcontroladores circuitos experimentales</p> | <p>1.-Aplicación de la práctica. 2.- Investigación y desarrollo con microcontroladores 3.Experimentación practica</p> | <p>Manual de prácticas</p> |
| <p>VARIABLE DEPENDIENTE : el módulo de comunicación Wi-fi permitirá el aprendizaje de aplicaciones con redes inalámbricas.</p> | <p>Módulo WI-FI:</p> <p>Es un circuito integrado que permite las aplicaciones mediante el protocolo de comunicación entre dispositivos</p> <p>Aprendizaje práctico: mediante el cual los estudiantes aplican lo aprendido en clase en el lugar de trabajo, en entornos de investigación.</p> | <p>-Comunicación entre dispositivos Wi-fi</p> <p>-Aprendizaje significativo en la práctica</p> | <p>4.Aplicaciones de comunicación Wi-fi</p> <p>5. Comprensión y entendimiento</p> | <p>-Observación</p> <p>-Análisis de los componentes.</p> |

Tabla N°7: operacionalización de variables

Elaborado: Los autores

2.4. PROCEDIMIENTOS

2.4.1. DISEÑO DE PLACAS ELECTRÓNICAS PARA EL ENTRENADOR DE MICROCONTROLADORES

Las placas electrónicas están diseñadas para realizar la transmisión de datos en una red inalámbrica de sensores para un laboratorio; como primer paso se realizó el diseño individual de cada placa partiendo de cálculos, simulación de circuitos en Proteus, ruteo en Ares y montaje de las placas en el entrenador de microcontrolador.

El diseño y simulación de las placas que componen el entrenador de microcontroladores se puede ver en el **ANEXO A**.

A continuación se describen las placas utilizadas para las prácticas con el módulo Wifly.

2.4.1.1 DISEÑO DEL RELÉ DE ESTADO SÓLIDO - CRUCE POR CERO

Relé de estado sólido

Es un circuito electrónico que se caracteriza por controlar cargas elevadas de potencia a partir de señales de control con poco voltaje e intensidad, es un circuito híbrido separado por un optoacoplador que divide la baja intensidad con la alta intensidad. El circuito del relé de estado sólido simulado en Proteus (Isis) se puede observar en la **Figura N°19**.

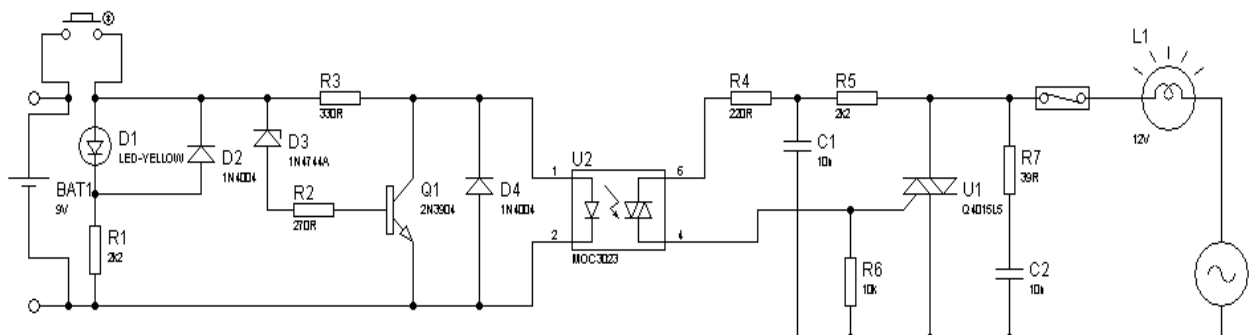


Figura N°19: Diagrama Relé de Estado Solido

Fuente: Autores

El diseño está conformado por una red Snubber ver la **figura N°20** que forman la resistencia (R7) y el condensador (C2), la función principal es suprimir transientes de voltaje inducidos por la carga o la red de 110V, los cuales pueden disparar el triac y causar problemas de operación; el Triac (U1) se utiliza como un interruptor de corriente alterna, a continuación los cálculos para la red de snubber.

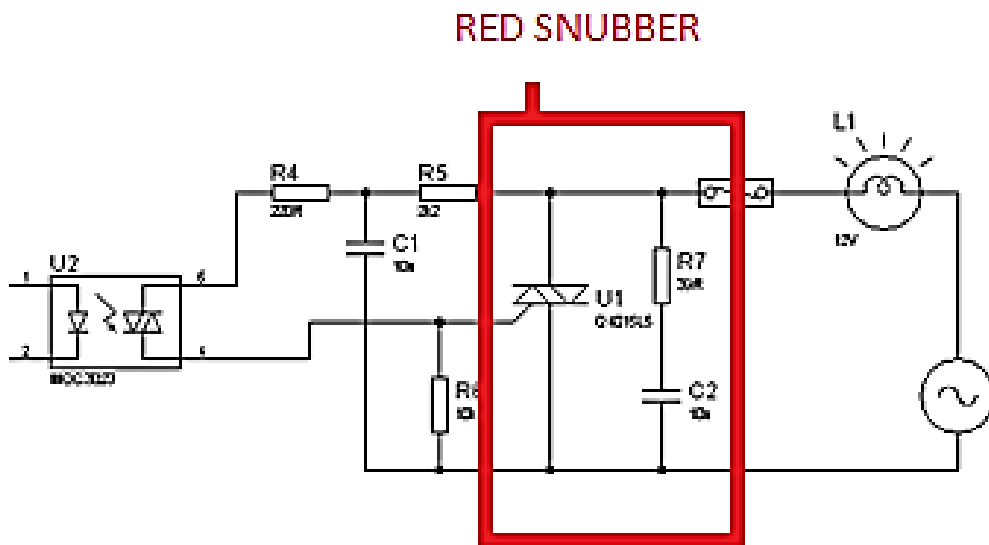


Figura N°20: Diagrama Relé de Estado Solido – Red Snubber
Fuente: Autores

$$V = VR7 + VR5 + VC2$$

$$V = VR7 + VR5 + \frac{1}{C} \int_{t_0}^t i dt$$

$$i(t) = \frac{V}{R5 + R7} e^{-\frac{t}{T}}$$

$$V_{U1} = V - VR5$$

$$V_{U1} = V - \frac{R5 \cdot V}{R5 + R7} e^{-\frac{t}{T}}$$

$$V_{U1} = V - \frac{R5 \cdot V}{R5 + R7}$$

La ecuación de T es la siguiente:

$$\tau = (R5+R7) C2$$

$$\tau = (39\Omega + 2.2K\Omega) * 10nf$$

$$\tau = 22.39 \mu$$

Una vez calculado τ , se realiza el cálculo de voltaje en el diodo zener Q4015L5 sirve para ver con que voltaje se activa.

Voltaje de activación = V

$$V_{U1} = V - \frac{R5 * V}{R5 + R7}$$

$$V_{U1} = 1.5V - \left(\frac{39\Omega * 1.5V}{39\Omega + 2.2k\Omega} \right)$$

$$V_{U1} = 1.473V$$

El diseño tiene un MOC 3023 que separa la parte de control de potencia con la parte de menor intensidad, un dispositivo que acopla señales de un circuito a otro por medio de luz infrarroja separando el circuito de DC con el AC, ver **figura N°21**.

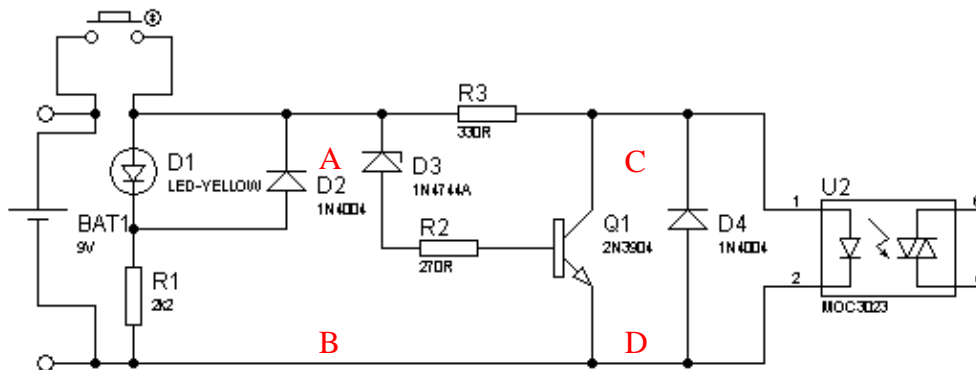


Figura N°21: Diagrama Análisis Relé de Estado Solido

Fuente: Autores

Para los cálculos de circuito primero partimos con la entrada de 5VDC, se considera el diodo dos (D2) como protección, calculamos el valor de la resistencia R1, en el punto ab tenemos el mismo voltaje de la entrada siempre y cuando el diodo 2 se abra.

$$-VDC+VD2+VR1=0$$

$$VDC=VD2+VR1$$

$$VDC-VD2=VR1$$

$$R1 = \frac{5V - 2.2.V}{1.25mA}$$

$$R1=2.24K\Omega$$

El D3 (1N4744) con la alimentación el diodo se abre dejando pasar la corriente por la R3 para tener una caída de tensión en el colector del transistor.

En la realización del lazo con el transistor 2N3904 para obtener el valor de la Resistencia R3, en los cálculos saldrá resistencias no comerciales y habrá que poner para el armado de circuitos físicos las resistencias aproximadas a las más comerciales, a continuación realizamos el análisis de Voltaje del colector.

$$-5V+ICRC+VCE=0$$

$$VCE=VC-VE$$

El voltaje del emisor es igual a cero por que esta puesto a tierra, la corriente de colector a utilizar es de 10.5mA.

$$VCE=VC$$

$$VC=RC*IC$$

$$RC = \frac{3.54V}{10.5mA} = 337.14\Omega$$

El valor VC se remplaza en la primera ecuación.

$$-5V + ICRC + VC = 0$$

$$5V - ICRC = VC$$

$$5V - (10.5mA * 337.14) = VC$$

$$5V - 3.46V = VC$$

$$1.54V = VC$$

El voltaje del colector es de 1.54V, el diodo 4 está en paralelo y la salida del mismo tenemos el mismo voltaje en el punto cd.

Cruce por cero

Este circuito electrónico se caracteriza por detectar la señal de la red eléctrica que cruza por el punto cero, son esenciales en la transmisión de señales digitales a través de circuitos de corriente alterna ejemplo los modem, puede ser usado como accionamiento de semiconductores de potencia como los transistores o para sincronizar señales de corriente alterna.

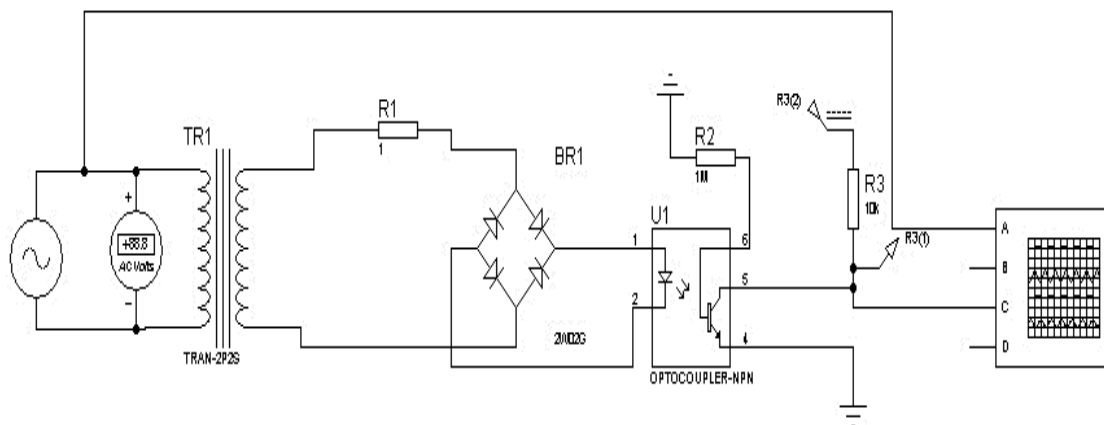


Figura N°22: Circuito Cruce por Cero
Fuente: Autores

La red eléctrica que pasa por el transformador es +12VAC, seguidamente la señal se la rectifica a través de un puente de diodos para obtener una señal de 12VDC manteniendo la misma frecuencia de 60 Hz, la misma señal de 12VDC entra al opto acoplador 4N25. En las

Figura N°22 se puede observar el diagrama del circuito cruce por cero simulado en Proteus (Isis), **en la figura N°23** se puede observar la respuesta en salida del circuito cruce por cero en el osciloscopio

En la **figura N°23** se puede diferenciar que la tensión de red eléctrica cambia de polaridad teniendo en cuenta que un ciclo completo forma de medio ciclo positivo y medio ciclo negativo.

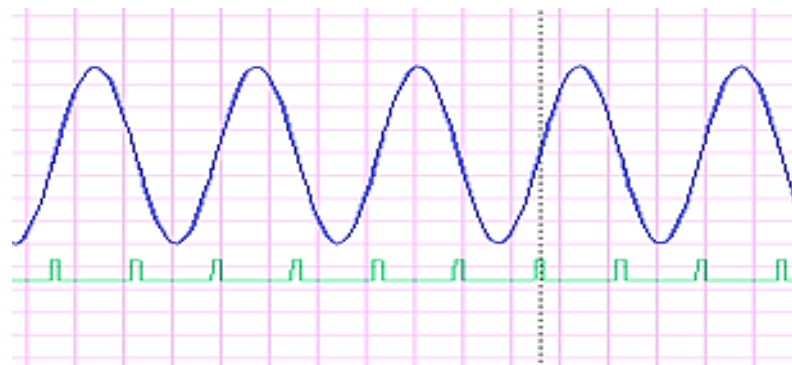


Figura N°23: Grafica Salida del osciloscopio del cruce por cero
Fuente: Autores

Para el diseño electrónico la corriente máxima que soporta el optoacoplador 4N25 es de 60 mA, el dato del datasheet.

El optoacoplador en su parte interior tiene un foto diodo que recibe la señal rectificada si la señal es positiva emite 1 lógico hacia el fototransistor lo cual transmite 0 lógico al microcontrolador y si el fotodiodo transmite 0 lógico el transistor trabaja en corte y transmite 1 lógico.

Los cálculos realizados son para un rectificador de onda completa.

$$V_{dc} = \frac{2V_m}{\pi}$$

$$V_{dc} = \frac{2V_{RMS} - (0.7) * 2}{\pi}$$

$$V_{RMS} = \frac{V_{dc}}{2} * \pi + 1.4$$

$$V_{RMS} = \frac{12}{2} * \pi + 1.4$$

$$V_{RMS} = 20.24 \text{ V}$$

$$I_{dc} = \frac{2I_{Lmax}}{\pi}$$

$$\frac{I_{dc}}{2} * \pi = I_{Lmax}$$

$$I_{Lmax} = \frac{\pi(60 \text{ mA})}{2}$$

$$I_{Lmax} = 94.24 \text{ mA}$$

2.4.1.1.1. DISEÑO DE LAS PLACAS CRUCE POR CERO Y EL RELÉ DE ESTADO SÓLIDO EN ARES

En la Figura N°24 se muestra el diseño realizado en ares cada elemento con su respectivo nombre, el circuito con su respectivo nombre y ruteado, los circuitos están en la misma placa pero tienen diferentes funciones de manera independiente, se realizó un plano de tierra en toda la placa con objetivo de eliminar el ruido, el ruteo de las pistas tiene un ancho de 4mm tomando en cuenta el cálculo realizado a continuación, cada placa mide 10 cm²

$$\text{Ancho} = \left\{ \left[\frac{I}{(k1 * \Delta T^{k2})} \right]^{1/k3} \right\} / (L * 1,378)$$

I= Corriente máxima 1A

K1= Constante estándar que vale 0,0150

K2= Constante estándar que vale 0,5453

K3=Constante que vale 0,7349

L= 2 onzas de cobre

ΔT= Variación de temperatura

$$\text{Ancho} = \frac{\left[\frac{1}{0.0150 * 10^{0.5453}} \right]^{0.7349}}{2 * 1.378}$$

$$\text{Ancho} = \frac{[5.70]^{1.3607}}{2.756}$$

Ancho=3.875mm \approx 4mm

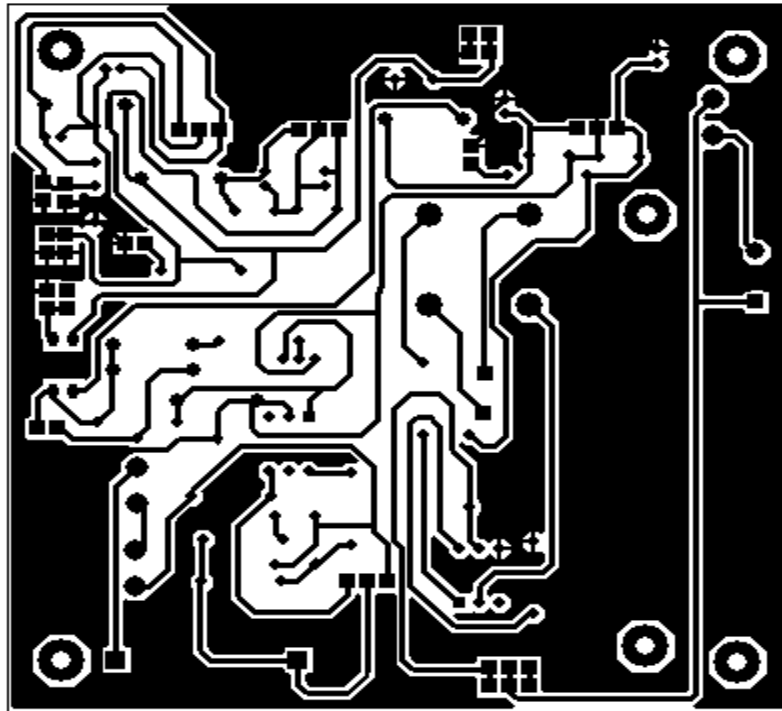


Figura N°24: Diseño en el Ares de la placa Relé de estado Sólido y Cruce por Cero
Fuente: Autores

2.4.1.1.2. DIAGRAMA EN 3 DIMENSIONES PLACA RELÉ DE ESTADO SÓLIDO – CRUCE POR CREO

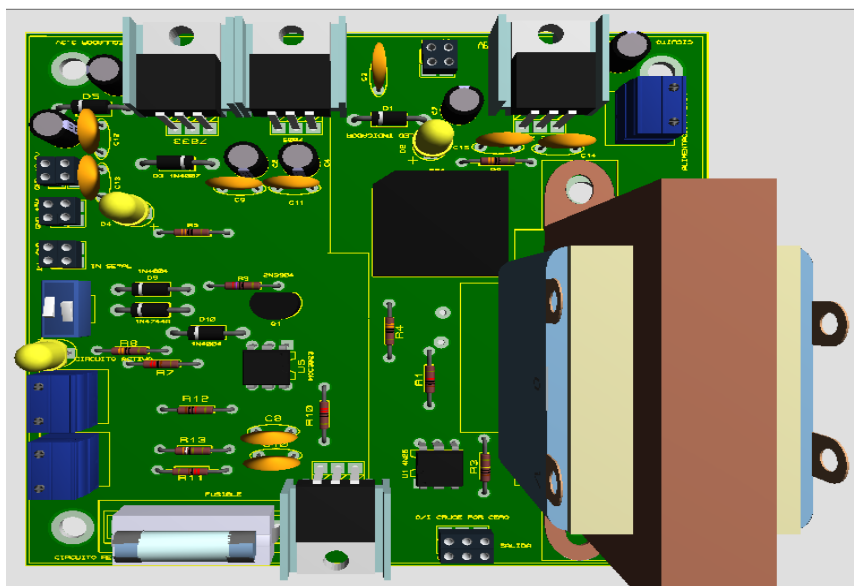


Figura N°25: Placa Relé de estado Sólido y Cruce por Cero en 3D
Fuente: Autores

2.4.1.2. PLACA DISPLAY Y LEDS

La placa electrónica tiene dos circuitos independientes de leds y display en su funcionamiento cada circuito es independiente, en la placa hay 8 leds cada uno con su entrada individual, hay un circuito con cuatro display que puede funcionar directamente conectado a las resistencias correspondientes o con el integrado 74ls47 que es un decodificador de 7 segmentos.

Para el diseño de la placa se realizó el cálculo para el valor de la resistencia que limita el paso de la corriente que circula por el diodo led tomando en cuenta la **figura N°26**, la resistencia se usa en el circuito de leds y Displays. El diseño de esta placa se puede ver en Anexos A.

Datos:

$$V_{in} = 5\text{vdc}$$

$$I_d = 7\text{mA}$$

$$V_d = 3.4\text{ V}$$

$$R = \frac{V_{in} - V_d}{I_d}$$

$$R = \frac{5\text{V} - 3.4\text{V}}{7.2\text{mA}}$$

$$R = 222\Omega \approx 220\Omega$$

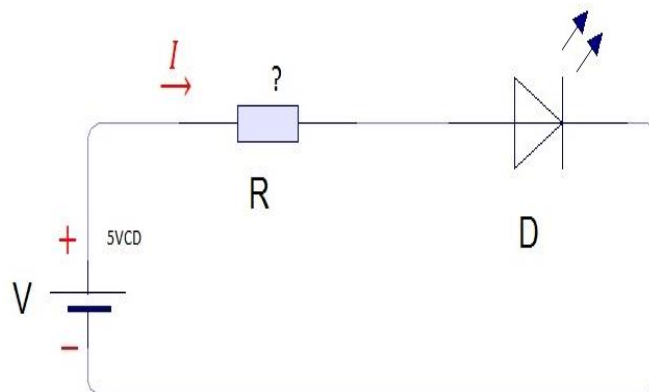


Figura N°26: Cálculo de la resistencia del diodo led
Fuente: Autores

2.4.1.2.1. DISEÑO DE LA PLACA ELECTRÓNICA EN EL ARES DE LOS LEDS Y DISPLAY.

En la **figura 27** se muestra el diseño del circuito de Leds y Display, el ruteado de la placa se realiza en el software Ares con sus correspondientes nombres y parámetros.

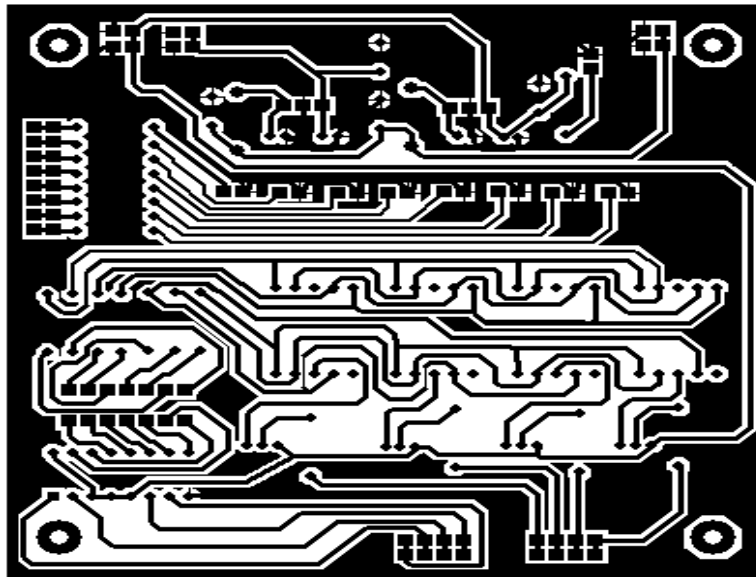


Figura N°27: Ruteo en Ares del circuito de Leds y Display
Fuente: Autores

2.4.1.2.2. DIAGRAMA EN 3 DIMENSIONES PLACA LEDS Y DISPLAY

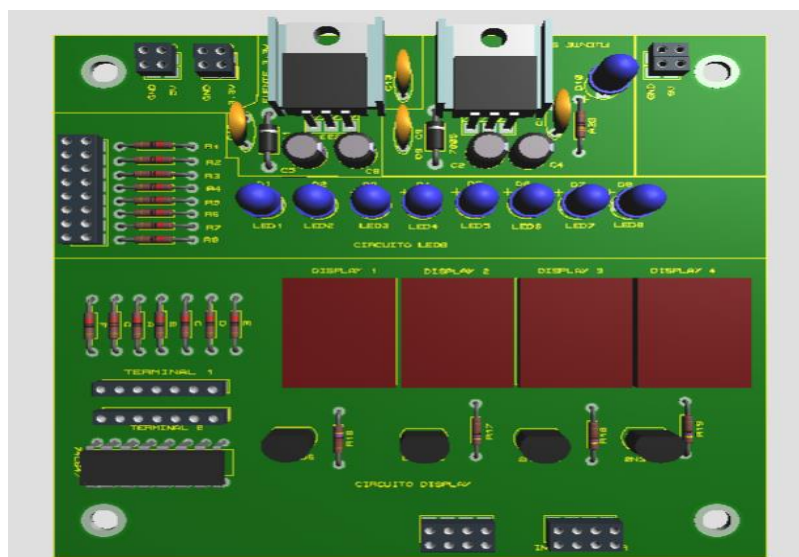


Figura N°28: Placa de Leds y Display vista en 3D
Fuente: Autores

En la **Figura N°28** se muestra la placa electrónica en tres dimensiones con todos los elementos, físicamente se visualizará en el entrenador de microcontroladores para el uso de estudiantes y docentes.

2.4.1.3. DISEÑO DE LA PLACA DEL MÓDULO WIFLY (RN-XV 171)

El diseño del módulo Wifly se realizó de acuerdo a los parámetros del datasheet agregando un divisor de voltaje en los pines de transmisión y un regulador de voltaje de 3.3vdc para la alimentación del módulo Wifly

En la **figura N°29** se muestra el diseño de la placa realizada en Proteus para el funcionamiento del módulo Wifly, con sus respectivas salidas e indicadores.

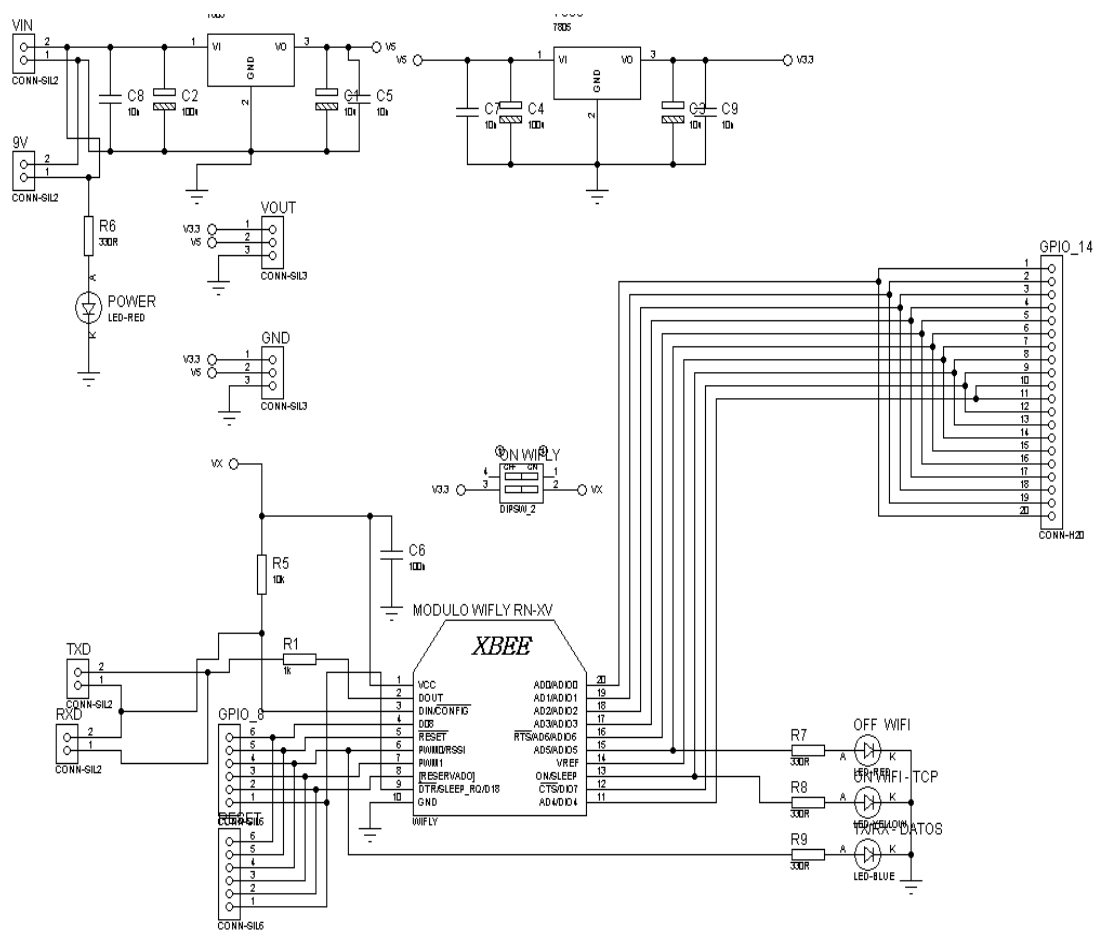


Figura N°29: Diseño de placa del circuito módulo Wifly

Fuente: Autores

2.4.1.3.1. DISEÑO DE LA PLACA ELECTRÓNICA WIFI EN ARES

En la **figura N°30** se muestra el diseño del módulo Wifly en donde se hace el respectivo ruteado con sus correspondientes nombres y parámetros, para el diseño de pistas revisar la sección 2.4.1.1.1 para tener un ancho de pista de 4mm, tomando en cuenta que toda la placa esta puesta a tierra.

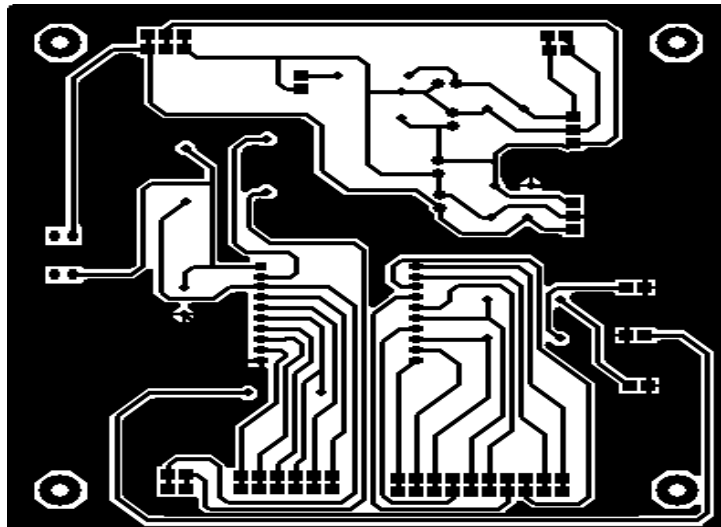


Figura N°30: Ruteo de la palca módulo Wifly

Fuente: Autores

2.4.1.3.2. DIAGRAMA EN 3DIMENSIONES DEL MÓDULO WIFLY

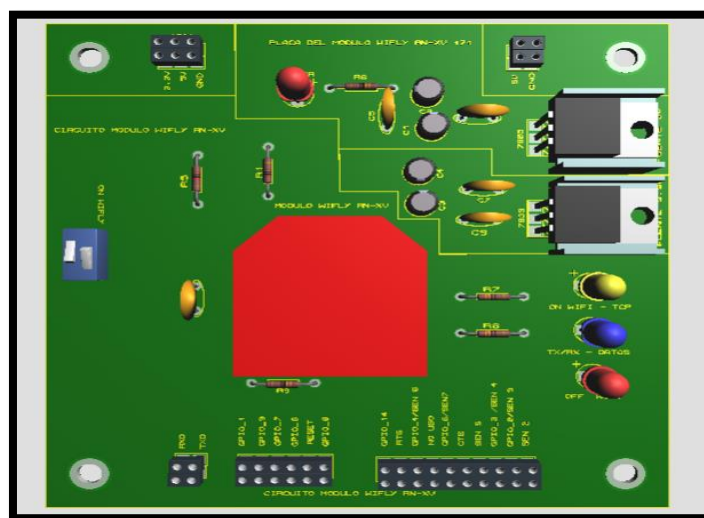


Figura N°31: Placa módulo WIFLY RN-XV 171 vista en 3D

Fuente: Autores

En la Figura N°31 se muestra la placa electrónica en tres dimensiones con todos los elementos, físicamente se visualizará en el entrenador de microcontroladores para el uso de estudiantes y docentes.

2.5. DISEÑO DE UN SISTEMA DE MONITERIO DE LABOTORIO MEDIANTE EL ENTRANADOR WIFI.

2.5.1. TOPOLOGIA DE LA RED

Aplicación para realizar la transmisión de datos mediante el módulo Wifly, el microcontrolador que procesa los datos recibe de tres sensores que monitorean temperatura, humedad y presión, los sensores usados son dth11 (sensor de temperatura y humedad digital), lm35 (sensor de temperatura analógico), mq6 (sensor de gas butano), los datos de las lecturas de los sensores se enviaran al microcontrolador que transmitirá la información inalámbricamente a través de cada módulo Wifly a una aplicación realizada en computadora en visual studio, el diagrama de conexión de la práctica se puede observar en la Figura N°32.

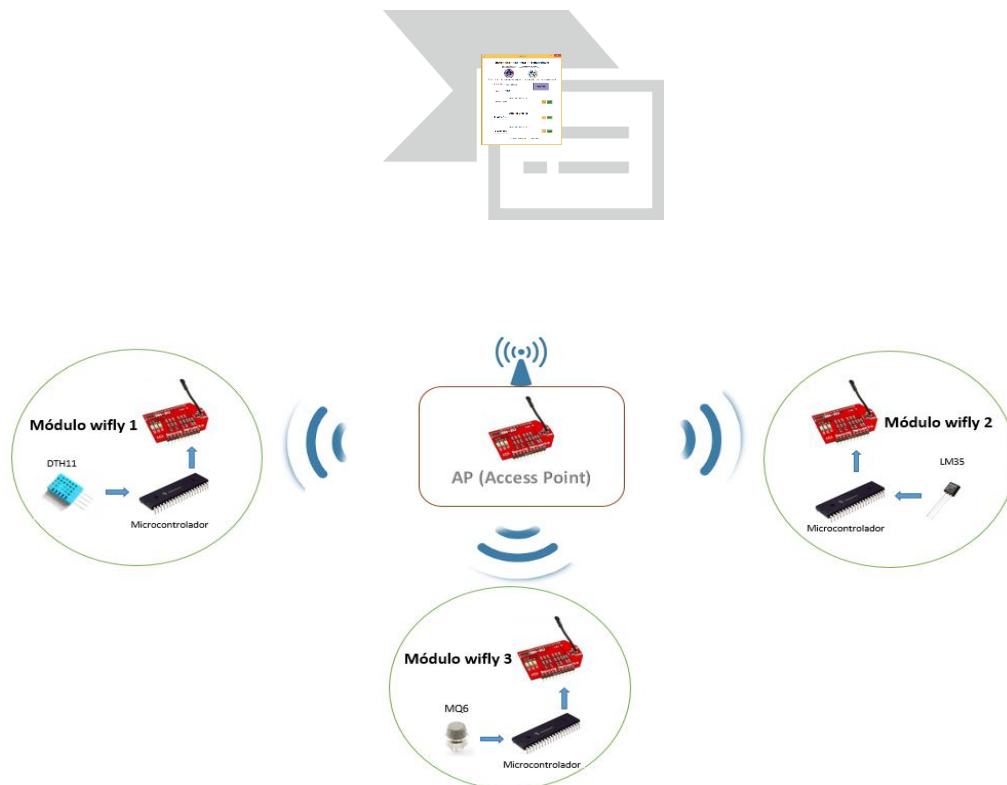


Figura N°32: Diagrama de conexión de la práctica transmisión de datos de sensores a través del módulo Wifly.

Fuente: Autores

Cada microcontrolador es programado para que transmitan o reciban datos por el módulo Wifly a continuación se detalla la conexión y programación de cada circuito por separado de los sensores con el microcontrolador.

La programación que se cargara en cada microcontrolador se encuentra en el anexo B – (PRACTICA 3 PROGRAMACIÓN DE LOS MICRONTROLADORES PARA SENSORES: DTH11, LM35, MQ6)

2.5.2. MÓDULO WIFLY SENSOR DE HUMEDAD Y TEMPERATURA

En la **Figura N°33** muestra el diagrama del circuito montado del módulo Wifly (1) que transmite datos del sensor dth11 al microcontrolador y transmitido por el módulo Wifly a la pc.

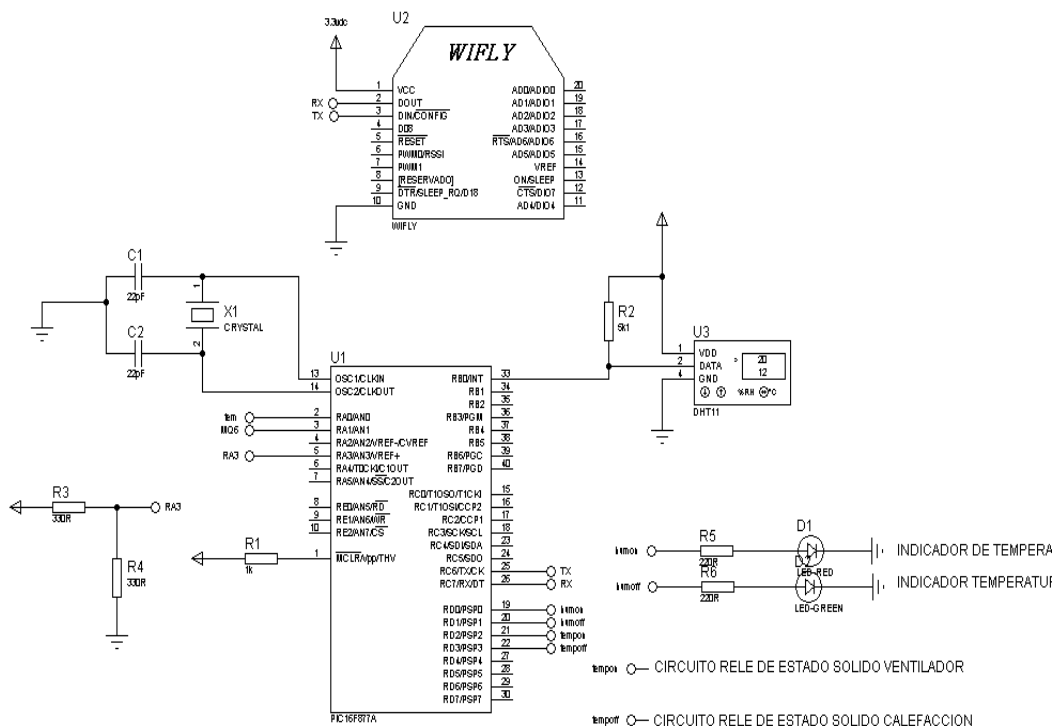


Figura N°33: Circuito microcontrolador con sensor DTH11

Fuente: Autores

En todas las aplicaciones realizadas con el microcontrolador 16f877a y módulo Wifly se realizó un divisor de voltaje con la resistencia R3 y R4 ver la **figura N°33**, la referencia análoga de

voltaje es seleccionada por software permitiendo utilizar la fuente de alimentación del PIC (VDD) o un nivel de voltaje externo aplicado al pin 5 (RA3/AN3/ VREF +).

$$V4 = \frac{R4}{R3+R4} Vin$$

$$V4 = \frac{330\Omega}{330\Omega+330\Omega} (5V)$$

$$V4 = 2.5V$$

Con el cálculo se puede decir que vamos a tener un voltaje de 2,5V para el pin 5 (RA3/AN3/ VREF +).

Diagrama de flujo de la programación del sensor DHT11 en MikroC

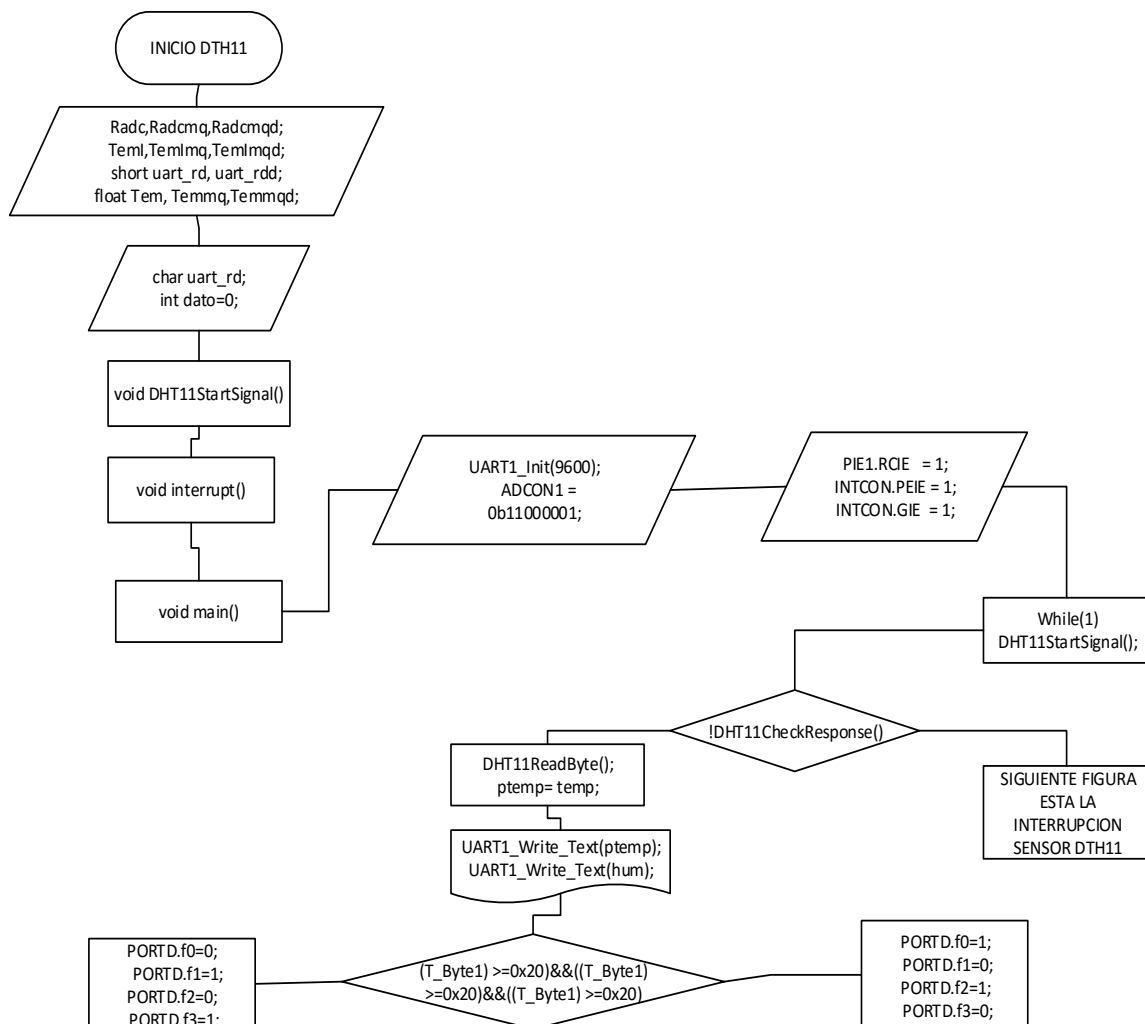


Figura N°34: Diagrama de programación de microcontrolador con DHT11

Fuente: Autores

En la **Figura N°34** se muestra el diagrama de flujo de la programación en MikroC del microcontrolador para el sensor dth11, el programa en mikroC se encuentra en anexo B (LABORATORIO 3 PROGRAMACIÓN DE LOS MICRONTROLADORES PARA SENSORES: DTH11).

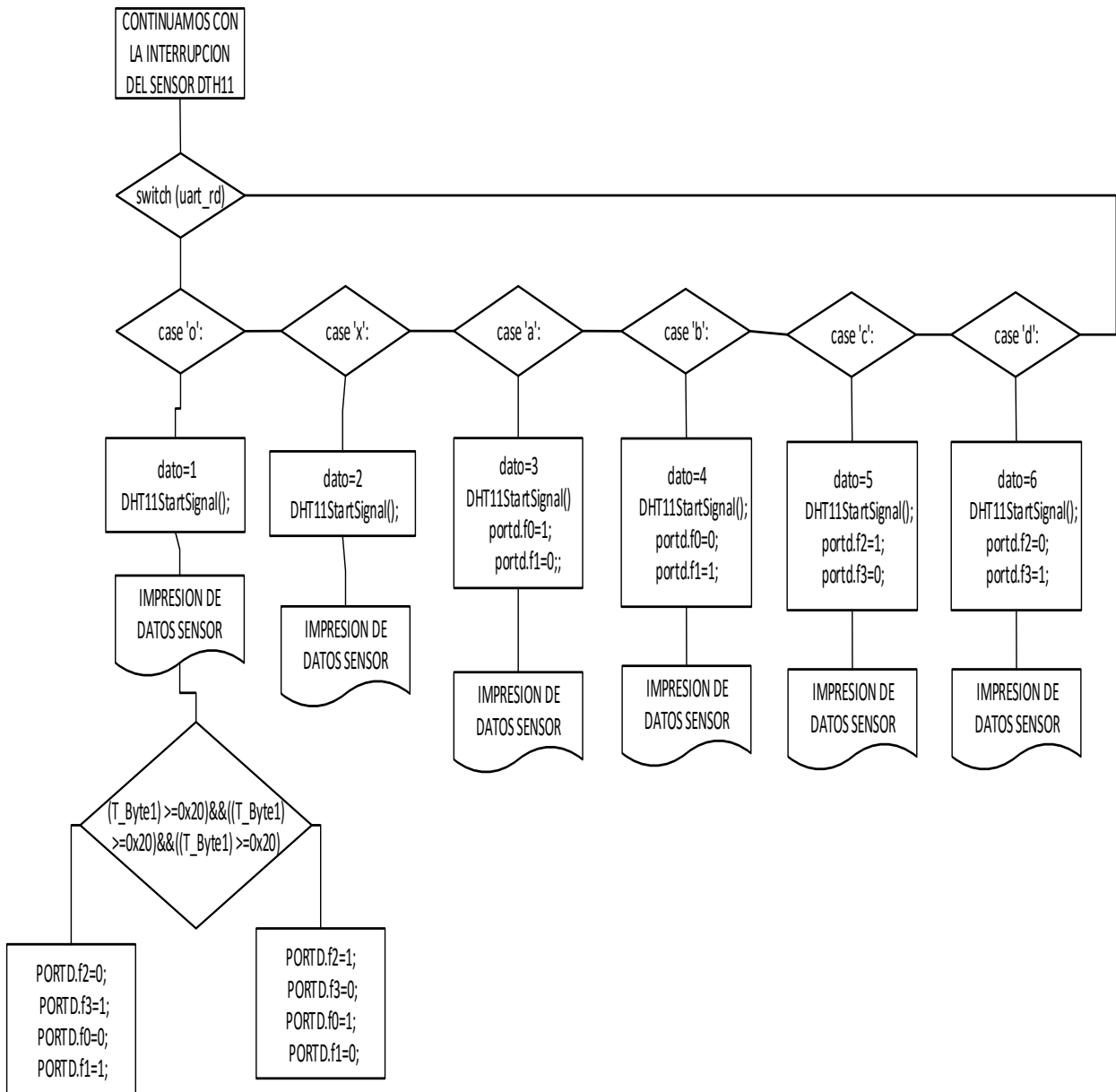


Figura N°35: Diagrama de programación de microcontrolador con DTH11 parte de interrupción

Fuente: Autores

EXPLICACIÓN DE LOS DIAGRAMAS DE FLUJO DEL SENSOR DTH11

Declarar las variables globales que se utiliza en la programación, variables para almacenar datos, declarar (uart_rd) para almacenar datos que llegen a traves de comunicación serial, declarar el puerto b (pin b0) como entrada digital para el sensor, declarar la funcion para que inicie la lectura del sensor cada 25 ms, declarar una función de interrupción, iniciamos el (void main()), dentro de esta funcion se declara las variables locales, la inicializacion de la velocidad de transmision es de 9600 baudios, habilitamos la interrupcion por uart para resivir un dato por el buffer, inicializa la lectura del sensor guardamos en las respectivas variables y se las imprime, una ves vizualizado los datos en la pc se compara los datos y deacuerdo a los parametros establecidos en la programacion se activa y se desactiva los pines 21 y 22 del pic 16f877a, el pin 21 esta conectado un rele de estado solido que tiene de carga un ventilador y el pin 22 tiene conectado un rele de estado solido que tiene de carga un calefactor, todo lo explicado se encuentra en la **Figura N°34**.

En la **figura N°35** es la continuacion de la **Figura N°34**, se explica una interrupcion, si hay la interrupcion el dato que resive por uart se guarda en una variable uart_rd, se compara si el dato que llego es una 'o' inicializa la lectura del sensor e imprime en la pc el valor de la temperatura y humedad y compara el valor de acuerdo a los parametros establecidos en la programacion para activacion y desactivacion de cargas a traves del pin 21 y 22 del pic 16f877a, si el dato es 'x' se imprime el valor de la temperatura y humedad en la pc, si el dato es 'a' se i mprime el valor de humedad y temperatura, se activa la sirena, si es 'b' se imprime el valor de humedad y temperatura, se desactiva la sirena, si el dato es 'c' se imprime el valor de humedad y temperatura, se activa el ventilador y se desactiva calefactor, si el dato es 'd' se imprime el valor de humedad y temperatura, se activa el calefactor y se desactiva el ventilador ver ANEXO B (LABORATORIO 3 PROGRAMACIÓN DE LOS MICRONTROLADORES PARA SENSORES: DTH11)

2.5.3. MÓDULO WIFLY SENSOR LM35 (TEMPERATURA)

En la **Figura N°36** se muestra el diagrama del circuito simulado en Proteus del módulo Wifly (2) que transmite datos mediante uart del sensor LM35, en la simulación se conecta a virtual terminal para recibir los datos enviados desde el microcontrolador mediante comunicación serial, el virtual terminal reemplaza virtualmente al módulo WiFly.

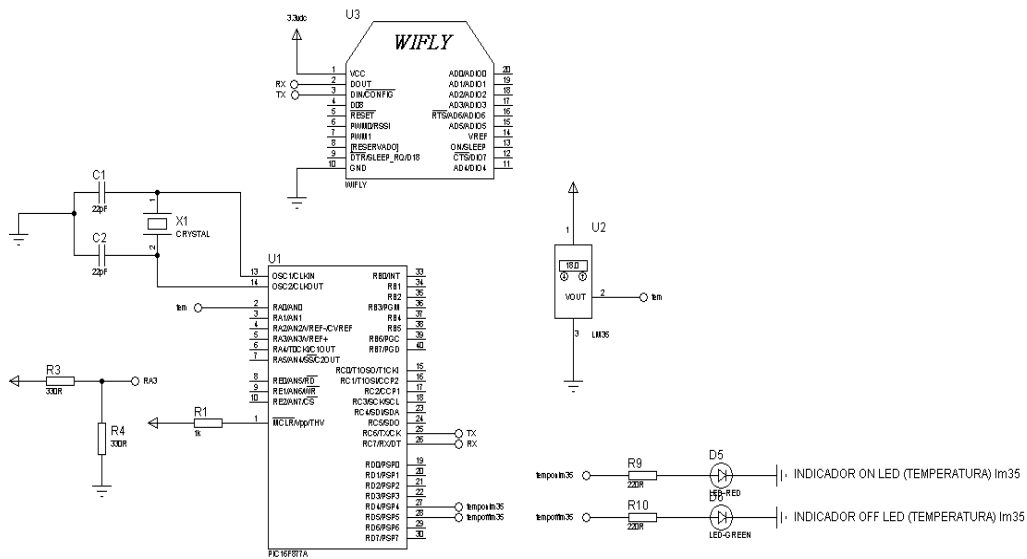


Figura N°36: Microcontrolador con sensor de temperatura LM35

Fuente: Autores

Explicacion de los diagramas de flujo del sensor lm35

Se declara variables globales, las variables para almacenar los datos, creamos la función (void interrupt), declaramos la función (void main), declarar las variables locales, encerramos los puertos, habilitamos para la lectura de valores analógicos del pic 16f877a, habilitamos la interrupción por uart cuando se recibe un dato por el buffer seguidamente leemos el valor analógico del sensor por el analógico 0, se realiza sus respectivas conversiones para la impresión por la pc, comparar los valores para activar o desactivar el pin 27 y pin 28 del microcontrolador, en estos pines van conectados relés de estado sólido que tienen de carga un ventilador para el pin 27 y un calefactor para el pin 28, todo lo explicado se puede ver **figura N°37**.

Diagrama de flujo de la programación mikroC lm35 (figura N°37, N° 38)

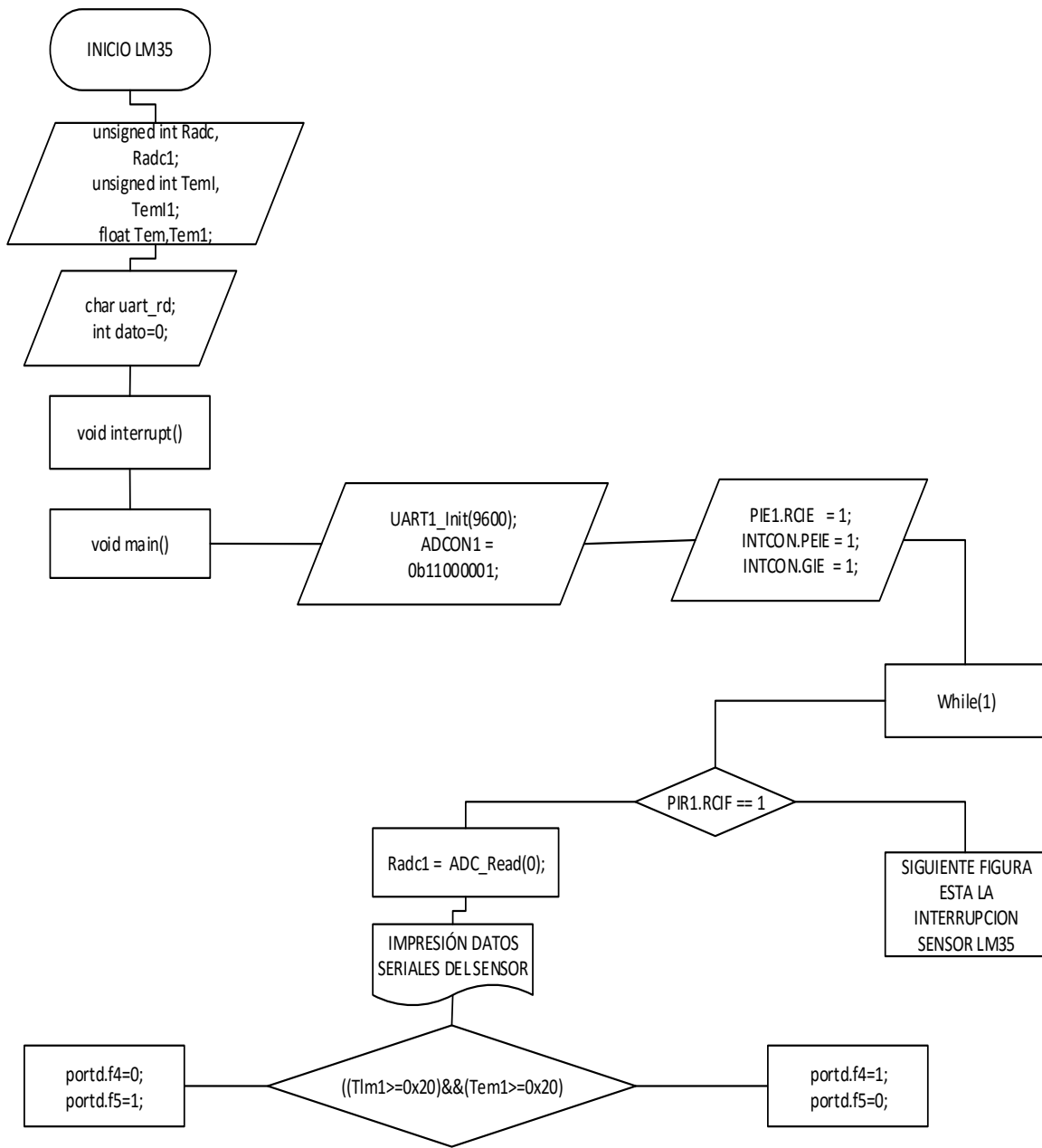


Figura N°37: Diagrama de flujo de la programación LM35
Fuente: Autores

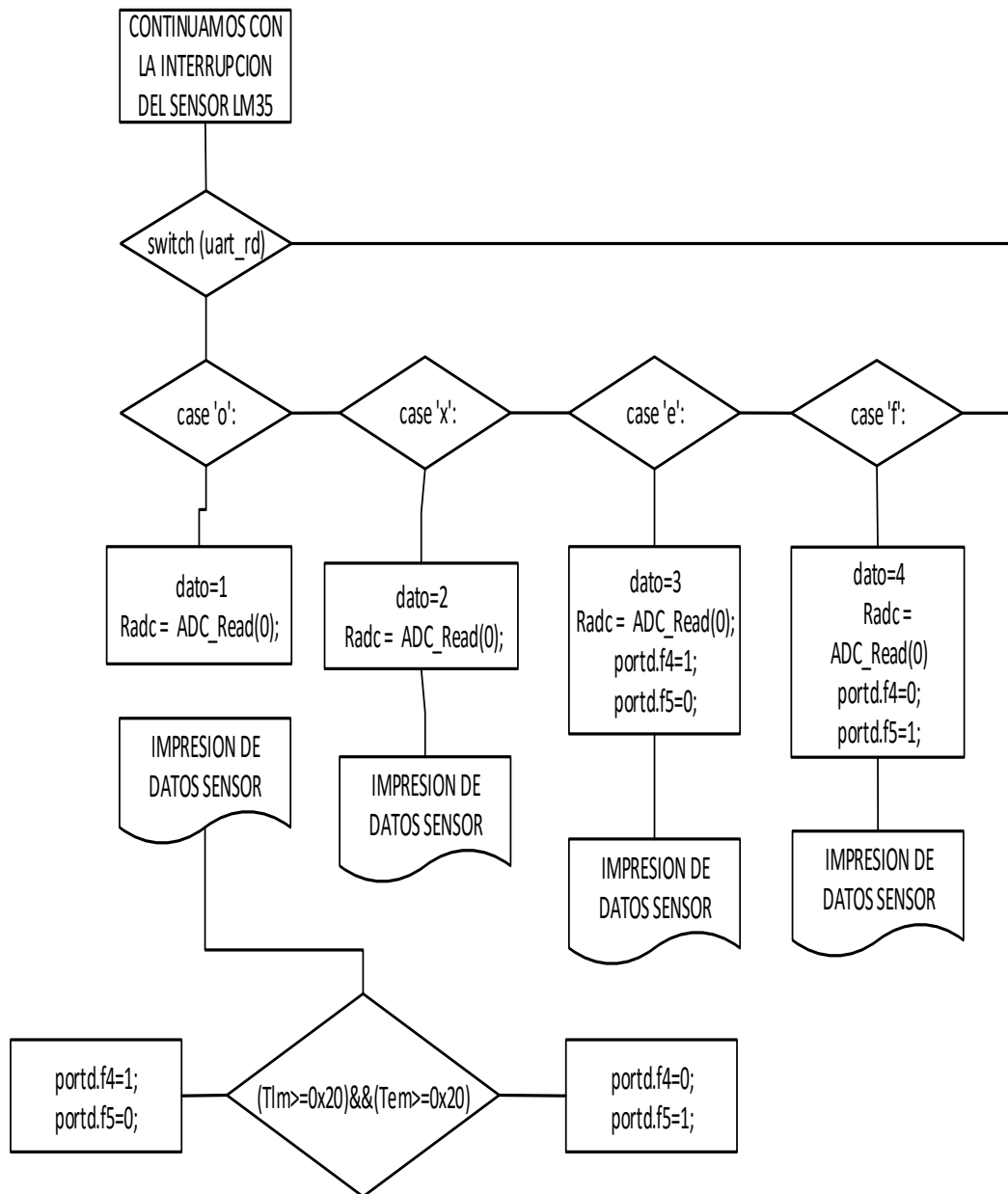


Figura N°38: Diagrama de flujo de la programación LM35 la parte de interrupción
Fuente: Autores

La **figura N°38** es la continuación del diagrama de flujo para una interrupción por uart, se hace una comparación si el dato que llega es 'o' se imprime el dato del sensor que es la temperatura y se activa y se desactiva las cargas de acuerdo a los parámetros del programa, si el dato es 'x' se imprime el valor de la temperatura en la pc, si el dato es 'e' se imprime el valor de la temperatura en la pc, se habilita el pin que está conectado el ventilador y se apaga el pin donde está conectado el calefactor, si el dato que llega es una 'f' se imprime el valor de la

temperatura en la pc, se habilita el pin que está conectado el calefactor y se apaga el pin donde está conectado el ventilador.

2.5.4. MÓDULO WIFLY SENSOR DE GAS (MQ6)

En la **Figura N°39** se muestra el diagrama del circuito en Proteus del sensor mq6 (gas), el sensor MQ6 es remplazado en la simulación con un potenciómetro.

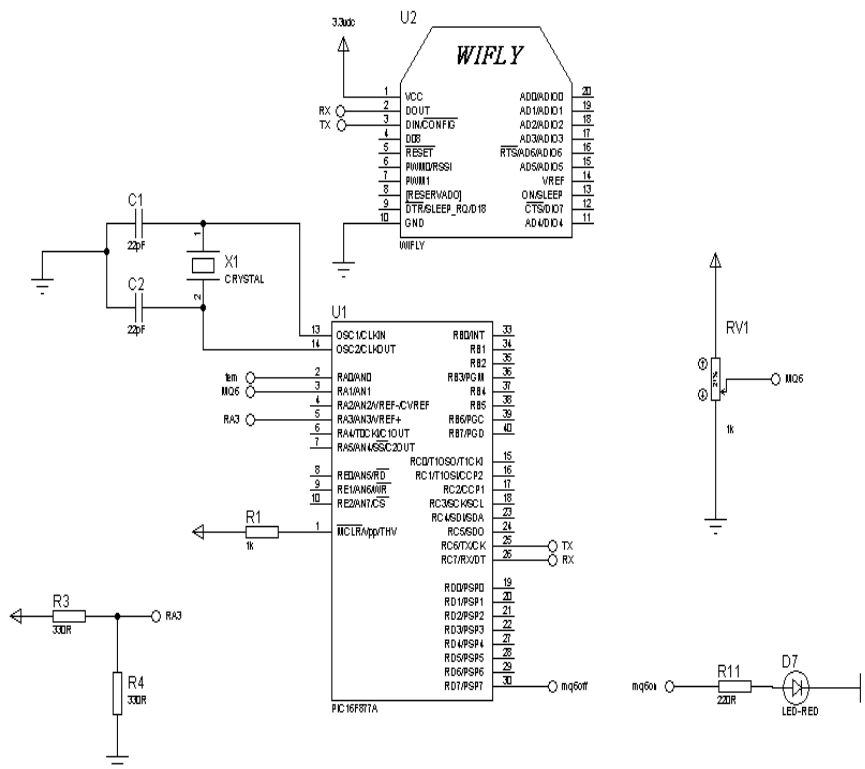


Figura N°39: Circuito de microcontrolador con sensor MQ6

Fuente: Autores

La características de sensibilidad del sensor MQ6, para calibrar que detecte 1000 ppm (partes por millon) de concentracion LGP (Gas Licuado de Petroleo), la resistencia de carga es de 20K Ω , ver conexión de circuito **figura N°17**.

Diagrama de flujo de la programación en mikroC del sensor mq6 (figura N°40, N°41)

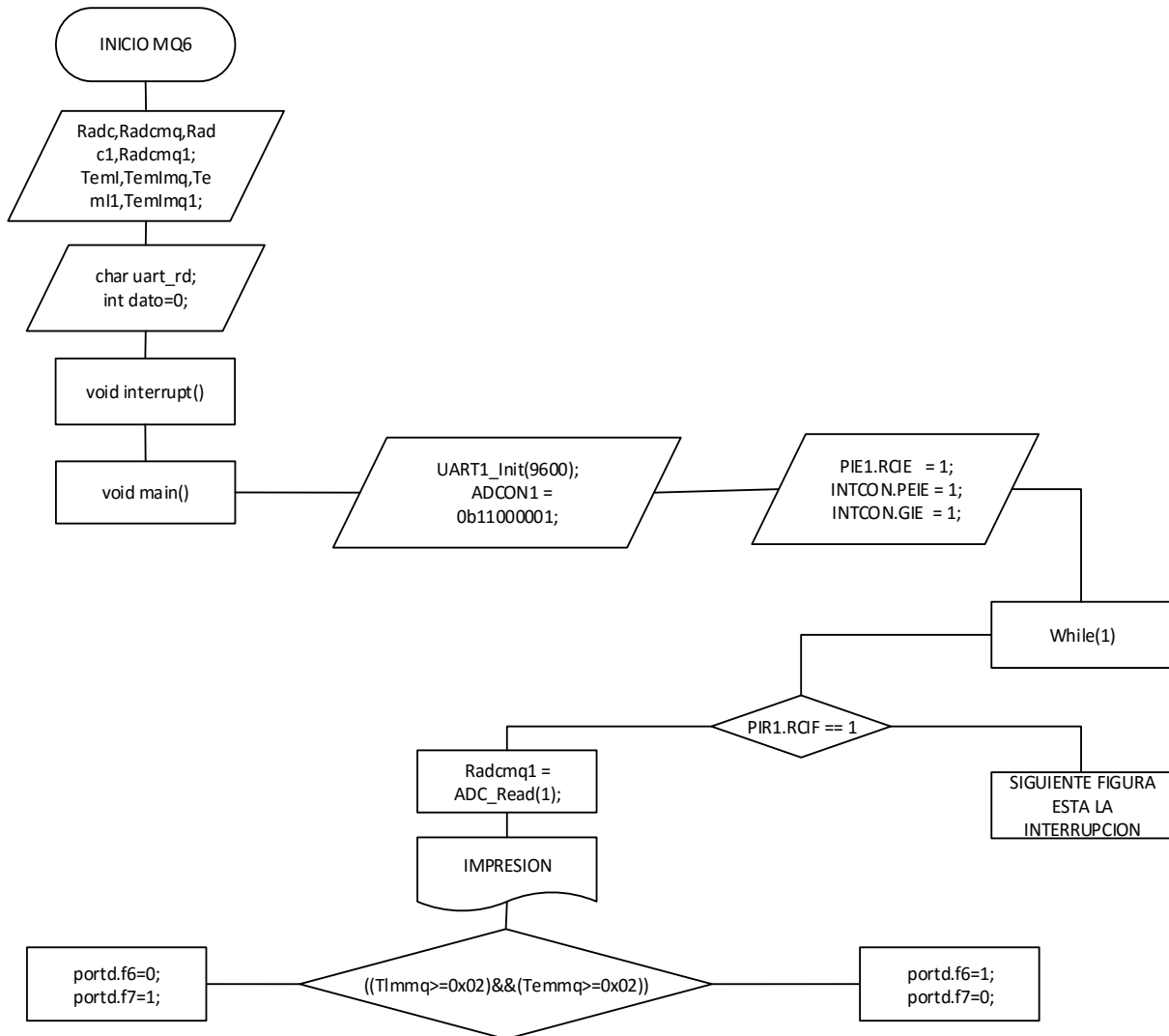


Figura N°40: Diagrama de flujo de la programación del sensor MQ6
Fuente: Autores

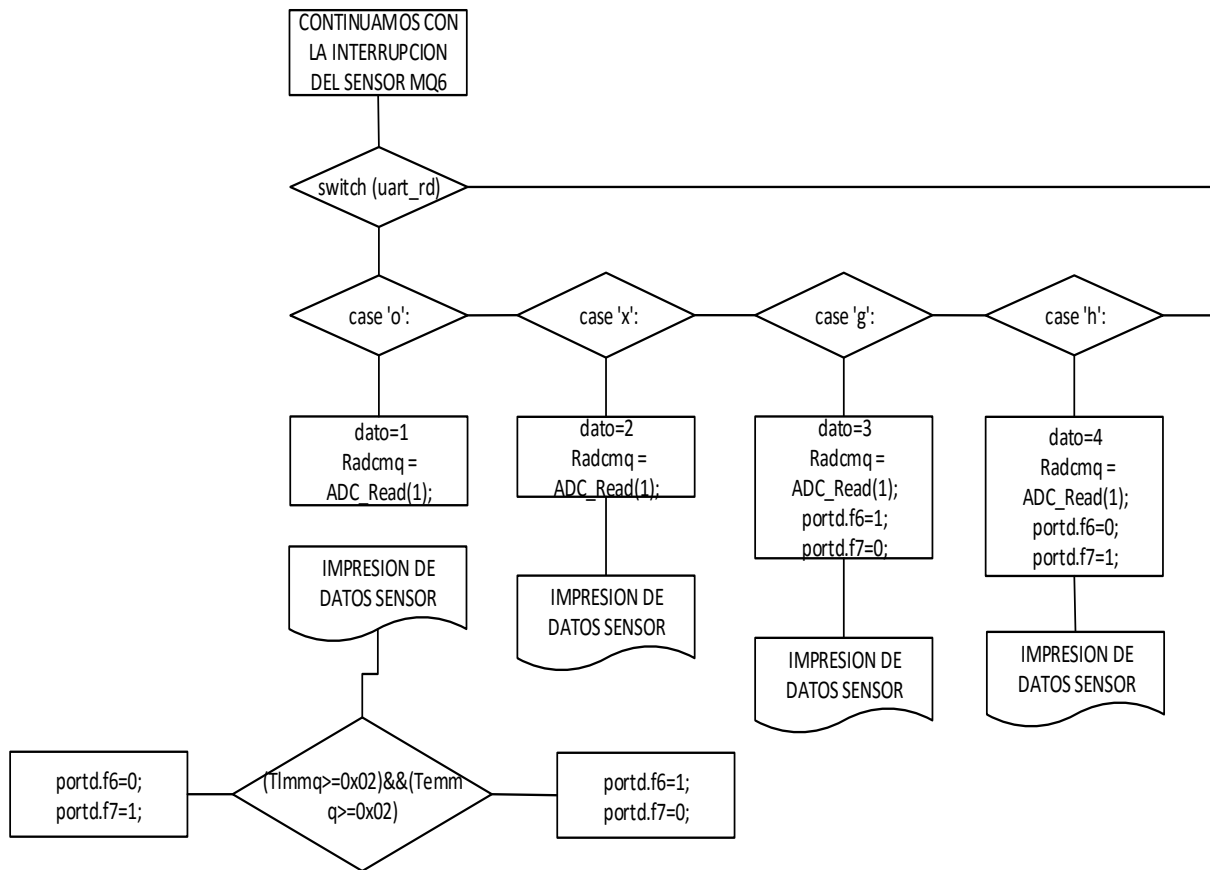


Figura N°41: Diagrama de flujo de la programación MQ6 parte interrupción
Fuente: Autores

Esplicación de los diagramas de flujo del sensor MQ6

Declarar variables globales, creamos la función (void interrupt), la función (void main), las variables locales, encerramos los puertos, habilitamos para la lectura de valores analógicos del pic 16f877a, habilitamos la interrupción por uart cuando se recibe un dato por el buffer seguidamente leemos el valor analógico del sensor por el analógico 0, hacemos sus respectivas conversiones para la impresión por la pc, una vez hecho la impresión procedemos a comparar los valores lo cual se activan o desactivan el pin 29 y pin 30 del microcontrolador, los pines van conectados relés de estado sólido que tienen de carga un ventilador para el pin 29 y un calefactor para el pin 30, todo lo explicado se puede ver **figura N°40**.

La **figura N°41** la continuación del diagrama de flujo para una interrupción por uart, se hace una comparación si el dato que llega es 'o' se imprime el dato del sensor gas y se activa y se

desactiva las cargas de acuerdo a los parámetros del programa, si es una 'x' se imprime el valor del sensor de gas en la pc, si el dato que llega es una 'g' se imprime el valor de gas en la pc, se habilita el pin que está conectado la sirena, si una 'h' se imprime el valor de gas en la pc, se habilita el pin que apaga la sirena.

2.5.5. SOCKETS CONEXIÓN SOCKET TCP/IP:

- Servidor establece un puerto y espera un cierto tiempo (tiempo/s) a que el cliente establezca conexión.
- Cliente solicita conexión, el servidor abrirá la conexión socket.
- El cliente establece una conexión con la máquina host a través del número de puerto designado.
- El cliente y el servidor se comunican con manejadores InputStream y OutputStream.

El funcionamiento del servidor cliente se puede ver en la **Figura N°42**:

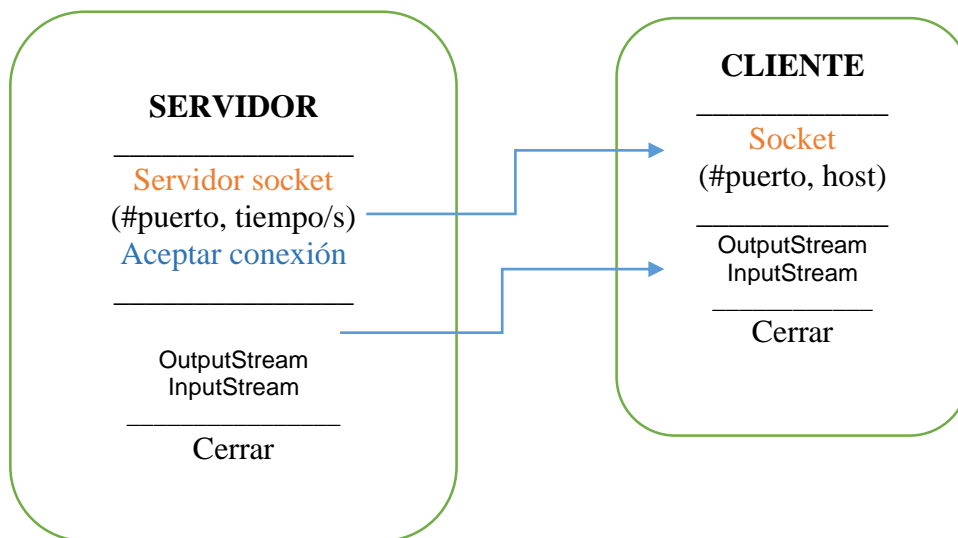


Figura N°42: Funcionamiento básico de sockets cliente-servidor

Fuente: Autores

Para el funcionamiento de la aplicación en visual estudio primero se inicializa el servidor esperando una respuesta del cliente para que pueda realizar conexión como se muestra en el diagrama de flujo de la **Figura N°43**.

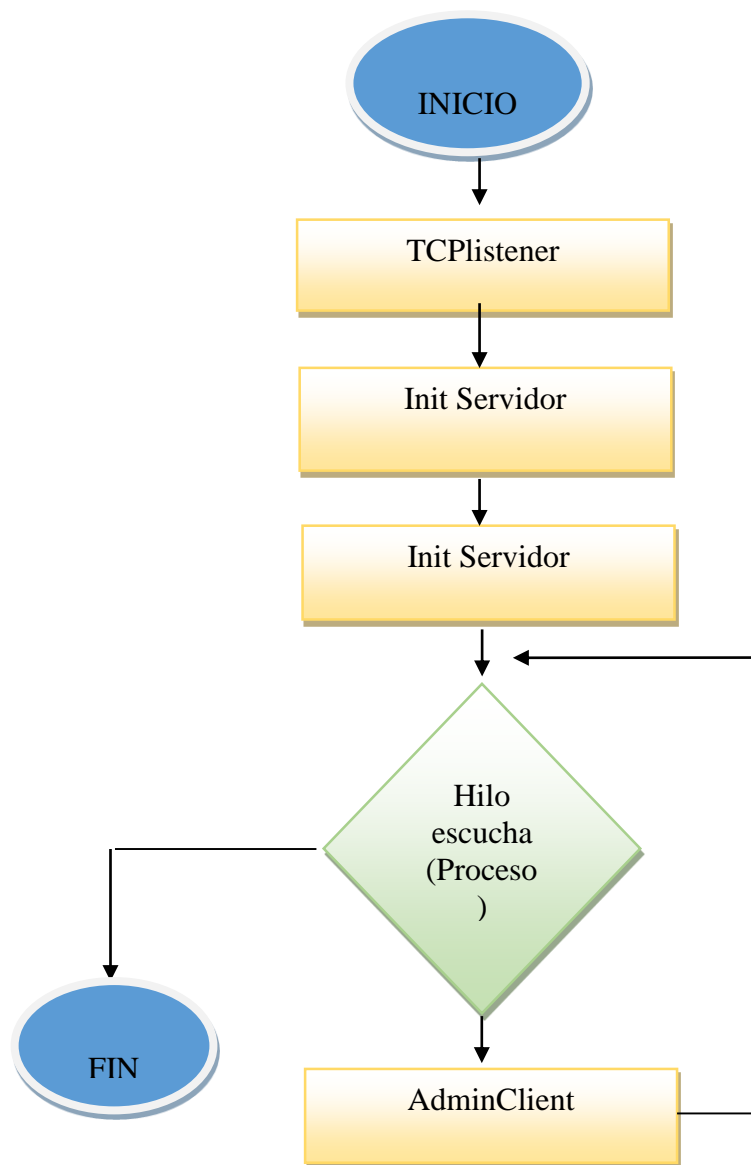


Figura N°43: Diagrama de flujo programación del Servidor en C# Visual estudio 2012
Fuente: Autores

Diagrama de procesos para administrar mensajes del cliente en la aplicación en visual studio para poder conectarse servidor con cliente se necesita que estén en la misma red escucha por el puerto designado de manera que escucha o envía datos mediante los botones para activar o

desactivar cargas de forma manual, el diagrama de flujo del funcionamiento se muestra en la **Figura N°44**.

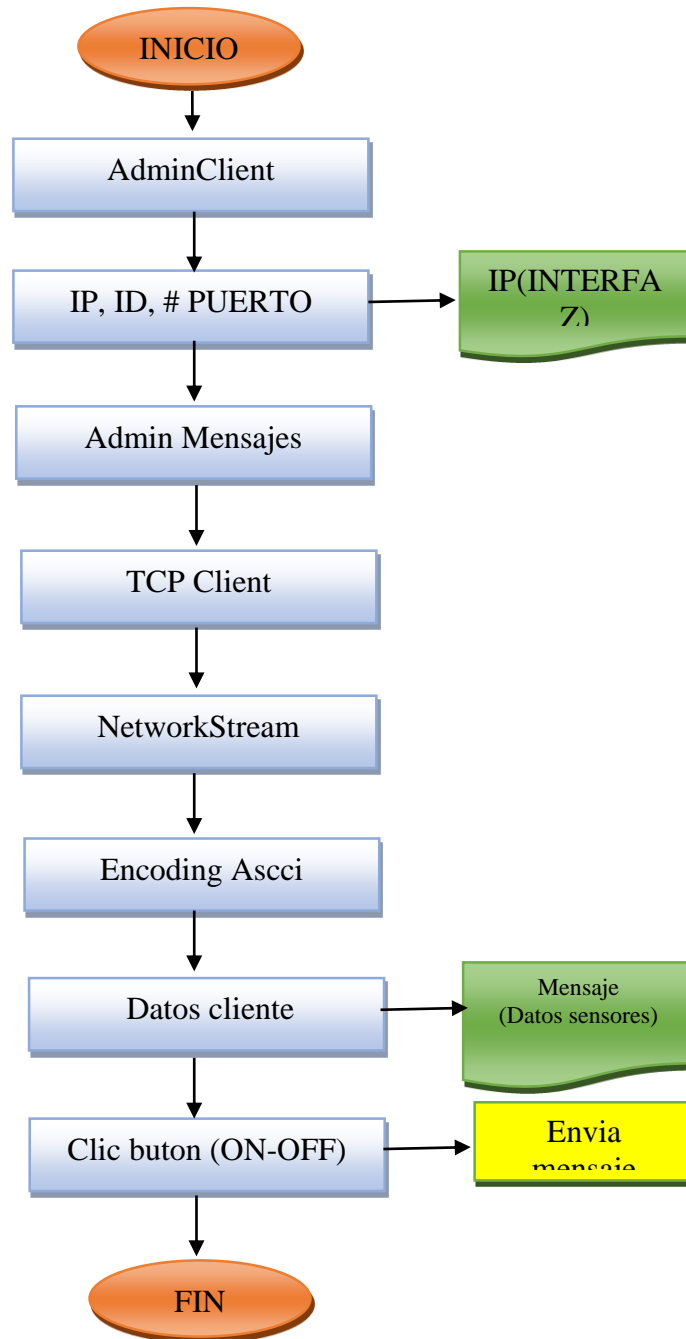


Figura N°44: Diagrama de flujo procesos del cliente de la aplicación en Visual Estudio
Fuente: Autores

Finalmente la interfaz de la aplicación se muestra en la **figura N°45** es necesario unirse a la ip y puerto del servidor para que puedan interactuar los datos de los sensores con la aplicación, la programación y los pasos para la realización de la aplicación en visual estudio se encuentra en el ANEXO B, PRACTICA 4 (DESARROLLO DE APLICACIÓN EN VISUAL ESTUDIO PARA RECEPCION Y CONTROL DE CARGAS DE UN LABORATORIO)



Figura N°45: Aplicación – 3 sensores y activación de cargas
Fuente: Autores

2.6. TRANSMISIÓN DE DATOS DE SENSORES DTH11, LM35, MQ6 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES EN UNA APLICACIÓN EN LA PC

2.6.1 CONFIGURACIÓN MÓDULO WIFLY COMO PUNTO DE ACCESO

Para realizar la configuración del módulo Wifly revisar Anexo B, laboratorio 2 (CONFIGURACION MODULO WIFLY COMO AP)

2.6.2. TRANSMISION DE DATOS DEL SENSOR DTH11 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES A UNA APLICACIÓN EN LA PC

Para la aplicación usted debe armar el circuito de la **figura N°46** tomando en cuenta todos los detalles de configuración y pines de alimentación Ver Anexo B, laboratorio 1 (INFORMACION BASICA), consecutivamente se debe configurar el módulo Wifly como dispositivo terminal ver Anexo B, laboratorio 2 (CONFIGURACIÓN DEL MODULO WIFLY COMO DISPOSITIVO TERMINAL), usted debe realizar el programa en mikroC y quemarle en el microcontrolador ver Anexo B, laboratorio 3 (PROGRAMACION DE LOS MICRONTROLADORES PARA SENSORES: DTH11), para la visualización de los datos de temperatura y humedad del sensor, usted debe realizar la interface en el software Visual studio ver Anexo B, laboratorio 4 (DESARROLLO DE APLICACIÓN EN VISUAL ESTUDIO PARA RECEPCIÓN Y CONTROL DE CARGAS DE UN LABORATORIO).

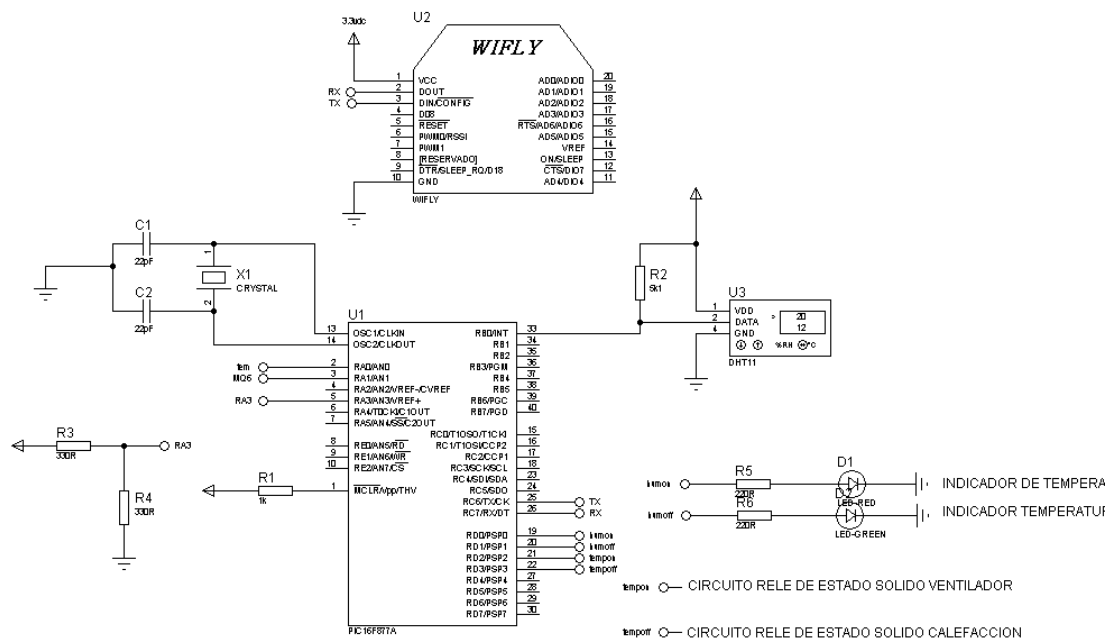


Figura N°46: Circuito del sensor DTH11

Fuente: Autores

Circuito del relé de estado sólido ventilador

El circuito de la **figura N°47** está asociado al circuito de la **figura N°46**, en el pin 21 del microcontrolador 16f877a tiene como salida de acuerdo a los parámetros configurados para el sensor DTH11, este pin hay un circuito acoplado de relé de estado sólido que a su salida tiene un ventilador.

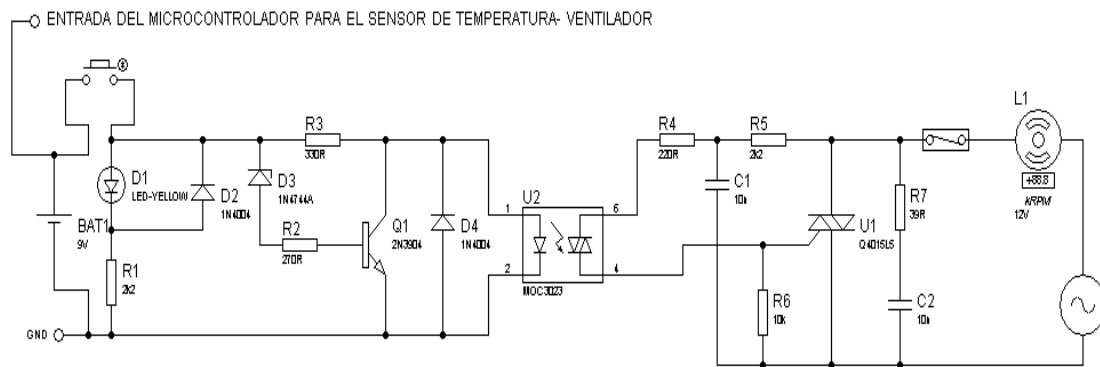


Figura N°47: relé de estado sólido ventilador.

Fuente: Autores

Circuito del relé de estado sólido calefactor

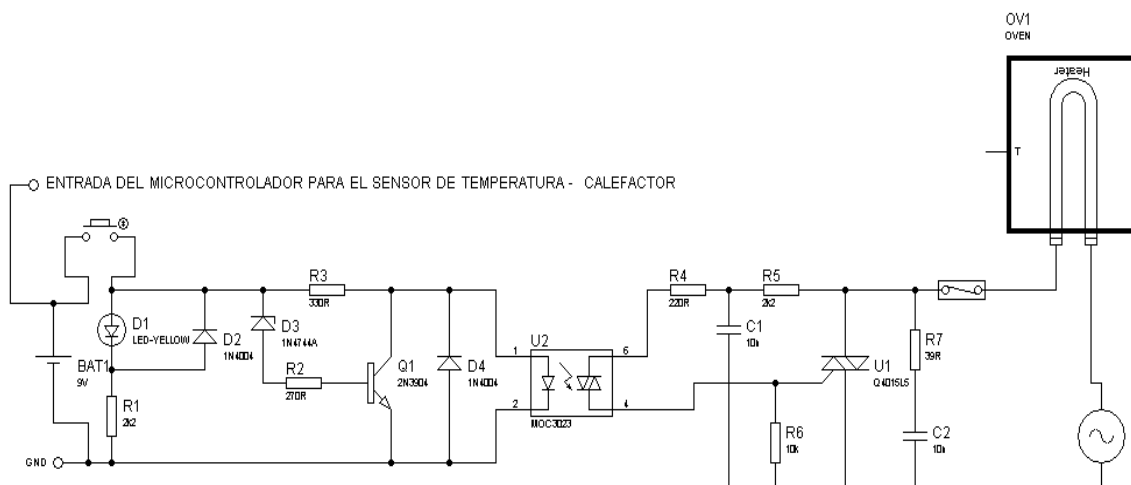


Figura N°48: relé de estado sólido calefactor

Fuente: Autores

El circuito de la **figura N°47** está asociado al circuito de la **figura N°48**, en el pin 22 del microcontrolador 16f877a tiene como salida de acuerdo a los parámetros configurados para el sensor DTH11, en este pin hay un circuito acoplado de relé de estado sólido que a su salida tiene un calefactor.

2.6.3. TRANSMISIÓN DE DATOS DEL SENSOR LM35 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES A UNA APLICACIÓN EN LA PC

Para la aplicación usted debe armar el circuito de la **figura N°49** tomando en cuenta todos los detalles de configuración y pines de alimentación Ver Anexo B, laboratorio 1 (INFORMACION BASICA), consecutivamente usted debe configurar el módulo Wifly como dispositivo terminal ver Anexo B, laboratorio 2 (CONFIGURACION DEL MODULO WIFLY COMO DISPOSITIVO TERMINAL), usted debe realizar el programa en mikroC y quemarle en el microcontrolador ver Anexo B, laboratorio 3 (PROGRAMACION DE LOS MICRONTROLADORES PARA SENSORES LM35).

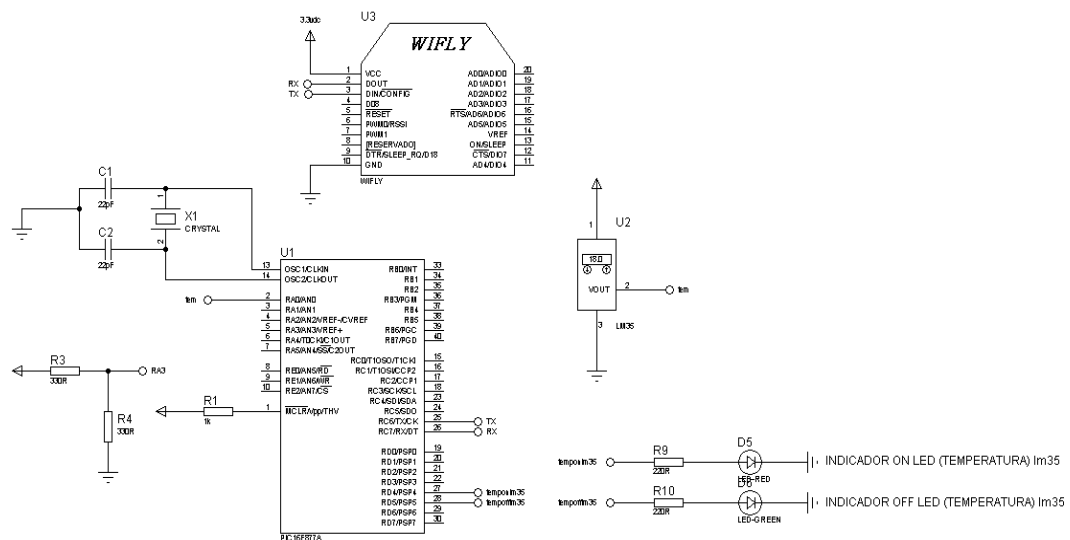


Figura N°49: Circuito sensor Im35

Fuente: Autores

La visualización de los datos de temperatura del sensor, usted debe realizar la interface en el software Visual studio ver Anexo B, laboratorio 4 (DESARROLLO DE APLICACIÓN EN VISUAL ESTUDIO PARA RECEPCION Y CONTROL DE CARGAS DE UN LABORATORIO).

Circuito del relé de estado sólido ventilador (Im35)

En la figura N°50 se tiene el circuito relé de estado sólido que funcionara de acuerdo a los parámetros de programación en el microcontrolador 16f877a para el sensor LM35, a la salida del relé de estado sólido hay un ventilador.

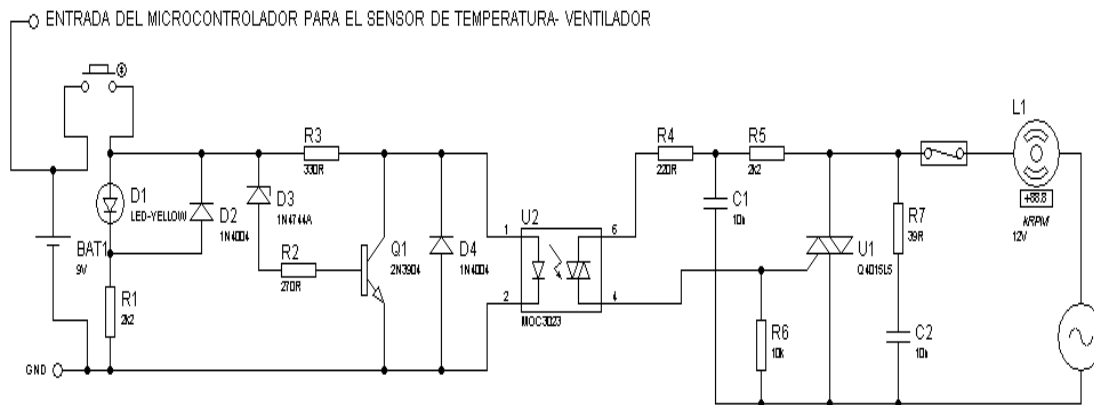


Figura N°50: Circuito sensor Im35

Fuente: Autores

2.6.4. TRANSMISION DE DATOS DEL SENSOR MQ6 A TRAVES DE LA RED WIFI CON MICROCONTROLADORES A UNA APLICACIÓN EN LA PC

Para la aplicación usted debe armar el circuito de la **figura N°51**, todos los detalles de configuración y pines de alimentación Ver Anexo B, laboratorio 1 (INFORMACION BASICA), consecutivamente se debe configurar el módulo Wifly como dispositivo terminal ver Anexo B, laboratorio 2 (CONFIGURACION DEL MODULO WIFLY COMO DISPOSITIVO TERMINAL), usted debe realizar el programa en mikroC y subirle el programa en el microcontrolador ver Anexo B, laboratorio 3 (PROGRAMACION DE LOS

MICROCONTROLADORES PARA SENSORES LM35), para la visualización de los datos de temperatura del sensor, usted debe realizar la interface en el software Visual studio ver Anexo B, laboratorio 4 (DESARROLLO DE APLICACIÓN EN VISUAL ESTUDIO PARA RECEPCION Y CONTROL DE CARGAS DE UN LABORATORIO).

Circuito del relé de estado sólido sirena (mq6)

El circuito de la **figura N°50** está asociado al circuito de la **figura N°51**, el pin 30 del pic 16f877a tiene como salida de acuerdo a los parámetros configurados para el sensor MQ6, en este pin hay un circuito acoplado de relé de estado sólido que a su salida tiene una sirena.

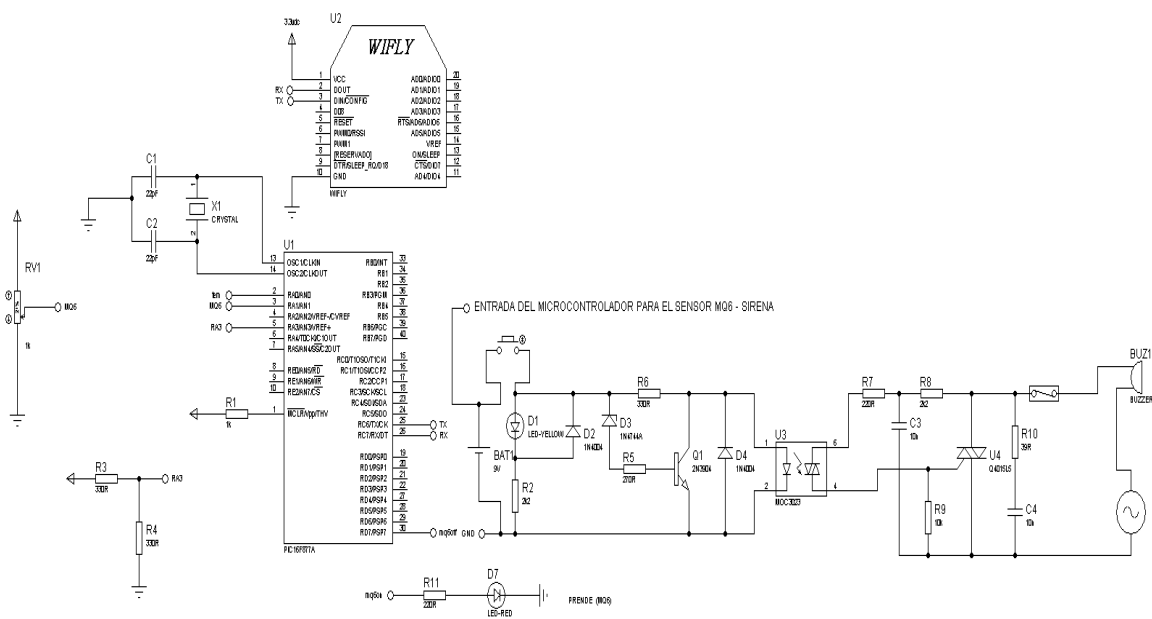


Figura N°51: Circuito sensor MQ6 – relé de estado sólido sirena
Fuente: Autores

Nota: Al momento de alimentar con voltaje los módulos Wifly hay que tener mucho cuidado, el modulo que trabaja como AP debe tener una alimentación única y los otros Wifly que trabajan como terminal deben tener diferente fuente de alimentación, deben tener las mismas tierras entre los Wifly que trabajan como terminal y los sensores, tener en cuenta que los pines

de Transmisión y Recepción del módulo Wifly deben ser conectados después de la alimentación correspondiente.

Se conecta uno por uno cada microcontrolador con su respectivo módulo Wifly y se abre la aplicación en la pc con un clic en el botón iniciar, la interfaz de la aplicación se puede ver en la **figura N°52**.

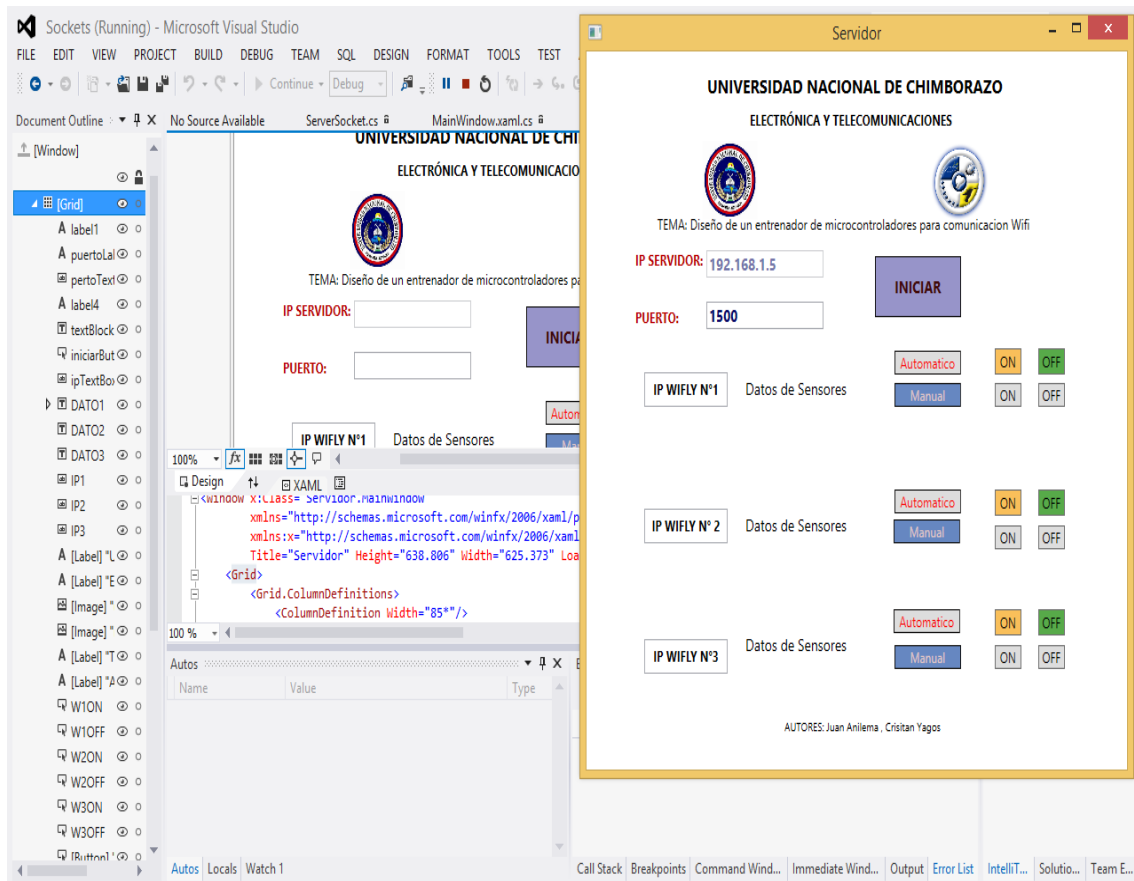


Figura N°52: aplicación a espera de recibir datos de sensores

Fuente: Autores

Al momento de conectarse cada Wifly con su respectivo microcontrolador la aplicación tomara la dirección ip de cada Wifly lo imprimirá en su pantalla como se puede observar en la **figura N°53**, cada que la aplicación recibe datos mediante uart.



Figura N°53: aplicación muestra ip de los módulos conectados
Fuente: Autores

Finalmente se enviara los datos de cada sensor y se mostrara en la aplicación correspondiente así como se podrá usar los botones para control de cargas de cada sensor (DTH11, LM35, MQ6)

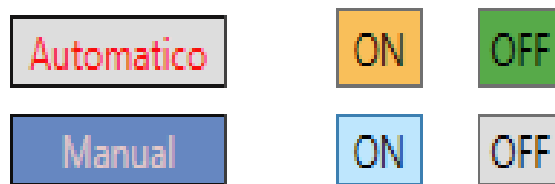


Figura N°54: funcionamiento de los botones de la aplicación
Fuente: Autores

Botón automático: Funciona de manera automática haciendo activar o desactivar cargas de cada sensor de acuerdo a los parámetros preestablecidos y los datos de sensores se imprimirán normalmente en la pantalla de la aplicación.

Sensor DTH11 (humedad y temperatura): temperatura menor a 18°C se activara el calefactor, temperatura mayor a 25°C se activara automáticamente el ventilador y en cuento a humedad superando el 35% se activara un indicador.

MQ6 (gas) superando los 500ppm el circuito de manera automática activara una alarma de alerta.

LM35 (temperatura) se toman en cuenta los mismos parámetros del sensor dth11 pero cada cambio se mostrara con la activación de su respectivo indicador.

Botón Manual: lee con normalidad los datos, para activar las cargas se lo realiza presionando los botones ON Y OFF:

Sensor DTH11:

Botones ON Y OFF primera fila para activar calefactor

Botones ON Y OFF segunda fila para activar ventilador

Sensor MQ6:

Botones ON Y OFF primera fila para activar la sirena

Sensor LM35:

Botones ON Y OFF primera fila para activar la luces del laboratorio

2.7. PROCESAMIENTO Y ANÁLISIS.

El siguiente trabajo de investigacion se obtiene conocimientos adecuados y profundos en el área de las comunicaciones alámbricas e inalámbricas utilizando diferentes tipos de protocolos aplicados a los microcontroladores.

Con el protocolo de comunicación Wi-fi, podremos controlar diferentes sensores, como son de temperatura, humedad, etc.

La lectura de datos serán enviadas al entrenador de microcontroladores que se encontrara en un laboratorio de la escuela de Ingeniería en Electrónica y Telecomunicaciones, desde el cual también podremos controlar dichos sensores y dispositivos, activar o desactivar y leer los datos

de cada elemento, a cierta velocidad de transmisión, el cual se medirá en (segundos o milisegundos) dependiendo de la velocidad de transmisión del módulo wifi, y los datos procesados se medirá dependiendo del sensor utilizado, así se podrá controlar un laboratorio desde una computadora a través del módulo wifly.

2.7.1. ANÁLISIS DEL DISPOSITIVO WIFLY.

El presente trabajo de investigación es el módulo wifly RN-XV, tiene los mismo zócalos de conexión que el módulo xbee que permitiría migrar una red 802.15.4 tipo XBee a una red TCP/IP 802.11 b/g Wi-Fi sin tener que rediseñar el proyecto a implementar, este módulo permite trabajar en modo full duplex que es transmisión y recepción de datos en tiempo real en una frecuencia de 2.4 gigahertz, el tiempo de latencia en una transmisión y recepción de datos es menor al de 1 segundo, este módulo tiene la característica de poder transmitir a una distancia de 35 metros sin obstáculos (exteriores) y con obstáculos (interiores) llega a una distancia de 8.5 metros. Este módulo tiene la capacidad de ser configurado como modo AP (Access Point o Punto de Acceso), también como un servidor, también para trabajar en modo cliente, el módulo wifly tiene la capacidad de poder ser configurado manualmente a través del software Hyper Terminal o por mediante un software que es WIFIBEE SETTING TOOL usando el cable usb serial (TTL) y de manera inalámbrica usando el modo AD HOC con el software WIFIBEE CONFIG TOOL, hay que tener muy en cuenta que estos dos softwares es de la empresa TINYSINE.

El módulo wifly es práctico y de mucha utilidad al realizar aplicaciones y prácticas usando el protocolo de comunicación wi-fi sólo necesita de 4 pines para funcionar (PWR, TX, RX y GND) y crear una conexión inalámbrica Wifi.

2.7.2. ANÁLISIS PRACTICO DE COMUNICACIÓN WI-FI.

La comunicación inalámbrica usando el protocolo wifi se tiene modos de funcionamiento como son access point (AP), managed (cliente), AD-HOC, estos modos pueden ser configurados en el módulo wifly de acuerdo a la aplicación a utilizar.

En la prácticas de comunicación se realizaron usando estas configuraciones para adhoc para probar configuración sin necesidad del cable usb serial, servidor al realizar la práctica de activación de cargas, la aplicación en la pc es el cliente, y como ap en la aplicación final donde tres módulos wifly se enlacen a uno para poder formar una red y transmitir los datos de los sensores enviados por los microcontroladores.

En la práctica los datos tansmitidos a través del módulo wifly es muy confiable, una característica del módulo wifly es tener como margen de error mínima en la pérdida de datos, que puede producir la pérdida de datos en la transmision o recepcion de datos es en la alimentación y en el pin de transmision cuando no se pone una resistencia a Vcc, el módulo wifly tiene la velocidad de datos: 464 Kbps utilizando flujo de hardware que es enviar la información al buffer hasta que esté lleno, el módulo wifly tiene la capacidad de enviar información inalámbrica desde 122 bytes hasta 255 bytes con esta capacidad se puede incluso hacer un servidor web y controlar cargas de modo duplex de acuerdo a las necesidades del usuario.

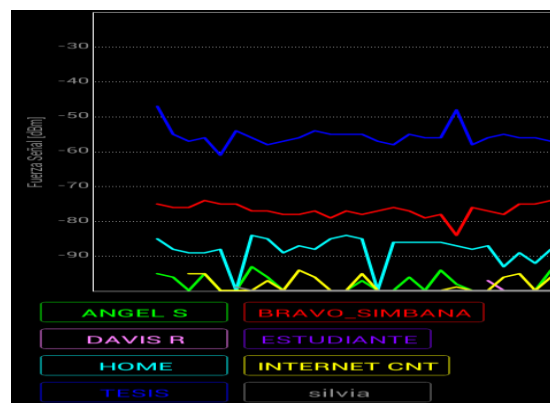


Figura N°55: Gráfica de la fuerza de la señal de modulo wifly como AP en el tiempo

Fuente: Autores

El analisis del gráfico usando el software wifi analyzer de las características de la red AP creada con el modulo wifly en la **Figura N°55** se puede ver el analisis del tiempo.

En la **Figura N°56** se puede ver el canal 3 que esta funcionando el módulo wifly, en la **Figura 57** se puede la grafica de potencia del modulo wifly actuando como AP, en la **Figura N°58** se puede ver la potencia de transmision de la red creada por el módulo wifly llamada TESIS que tiene una potencia de -58dBm.

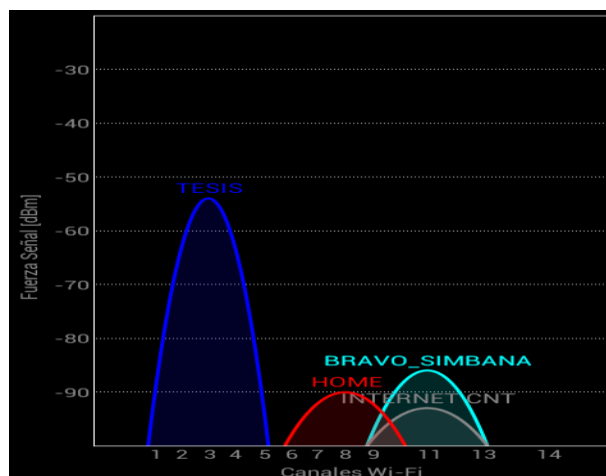


Figura N°56: Grafica del canal y potencia de la red Tesis
Fuente: Autores

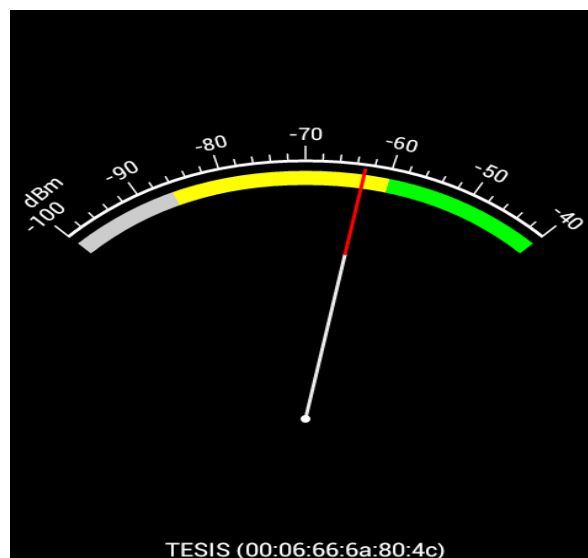


Figura N°57: Grafica de medición de la potencia en dBm red Tesis
Fuente: Autores



Figura N°58: Redes analizadas y lectura de datos
Fuente: Autores

2.7.3. TÉCNICAS DE PROCEDIMIENTO PARA EL ANÁLISIS

El análisis estadístico es cualitativo y cuantitativo, para la parte cualitativo es comprobar la estabilidad y confiabilidad de la comunicación wifi en la transmisión y recepción de datos inalámbricos, para la parte de la cuantitativa se comprobará la eficiencia en el desarrollo de prácticas usando el entrenador de microcontroladores por parte de los estudiantes del área de electrónica y telecomunicaciones.

2.8. COMPROBACIÓN DE HIPÓTESIS

2.8.1. PLANTEAMIENTO DE LA HIPOTESIS.

Se plantea la hipótesis nula y la hipótesis afirmativa:

H₀ Con el entrenador de microcontroladores de comunicación wifi no mejorará la eficiencia en el desarrollo de las prácticas.

H₁ Con el entrenador de microcontroladores de comunicación wifi mejorará la eficiencia en el desarrollo de las prácticas.

2.8.2. ESTABLECIMIENTO DEL NIVEL DE SIGNIFICANCIA.

El nivel de significancia de alfa es, $\alpha = 0,05\%$ que representa el 5% y con un 95% de confianza.

2.8.3. ELECCION DE LA PRUEBA DE HIPOTESIS.

Es una prueba de muestras relacionadas a un mismo grupo se le aplica dos medidas en un tiempo distinto, es un estudio longitudinal, la variable fija que crea los grupos son dos medidas una antes y una medida después, la variable aleatoria de comparación, es la variable tiempo es una variable numérica, por lo tanto se usa la prueba t student con muestras relacionadas para comprobar la hipótesis.

2.8.4. MUESTRA.

La muestra utilizada para comprobación de hipótesis es los 19 estudiantes de sexto semestre de ingeniería electrónica y telecomunicación de la Universidad Nacional de Chimborazo que están cursando la materia de microcontroladores

2.8.5. MEDIA

| Estudiantes | Tiempo sin entrenador | Tiempo con entrenador |
|--------------------|------------------------------|------------------------------|
| 1 | 18.3 | 12.21 |
| 2 | 17.5 | 12.4 |
| 3 | 19.1 | 12.45 |
| 4 | 20.1 | 10.11 |
| 5 | 21.5 | 11.53 |
| 6 | 19.9 | 11.2 |
| 7 | 18.46 | 9.39 |
| 8 | 18.43 | 8.28 |
| 9 | 19.19 | 9.53 |
| 10 | 19 | 9.12 |
| 11 | 21.48 | 10.34 |
| 12 | 19.45 | 11.09 |
| 13 | 20.29 | 8.31 |
| 14 | 17.52 | 9.46 |
| 15 | 19.19 | 11.19 |
| 16 | 21 | 11.56 |
| 17 | 21.4 | 10.14 |
| 18 | 19.47 | 9.36 |
| 19 | 16.5 | 9.13 |

Tabla N°8: Tiempos tomados de estudiantes para comprobación de hipótesis

Fuente: Autores

Media calculada

$M1=(367.78)/19=19.35$ Media (Tiempo sin entrenador)

$M2=(196.8)/19=10.35$ Media (Tiempo con entrenador)

2.8.6. PRUEBA DE NORMALIDAD.

Para calcular la normalidad se usa el metodo de Shapiro Wilk es un metodo usado para muestras menores a 30, se usa el software spss para encontrar el **p valor** (nivel de significación más pequeño posible que puede escogerse), para ver si la variable de tiempo se comporta con normalidad.

Pruebas de normalidad.

| | Kolmogorov-Smirnov | | | Shapiro-Wilk | | |
|-----|--------------------|----|--------|--------------|----|-------|
| | Estadístico | gl | Sig. | Estadístico | gl | Sig. |
| Tse | 0,100 | 19 | 0,200* | 0,963 | 19 | 0,642 |
| Tce | 0,154 | 19 | 0,200* | 0,940 | 19 | 0,262 |

Tabla N°9: Pruebas de Normalidad en spss.

Fuente: Autores

Tse=Tiempo sin entrenador, Tce=Tiempo con entrenador, gl=cantidad de estudiantes.

NORMALIDAD

| | | |
|--------------------------|---|----------------|
| Pvalor(t-antes) 0,642 | > | $\alpha= 0.05$ |
| Pvalor(t-despues)= 0,262 | > | $\alpha= 0.05$ |

Tabla N°10: tabla de resultados de normalidad.

Fuente: Autores.

El valor de datos de p valor (nivel de significancia) son mayor que el nivel de α (error aceptado).

Los datos de tiempo proviene de una distribución normal, la variable tiempo se comporta normalmente.

2.8.7. PRUEBA T PARA MUESTRAS RELACIONADAS

Estadísticos de muestras relacionadas

| | | Media | N | Desviación típica | Error típico de la media |
|-------|-----|----------|----|-------------------|--------------------------|
| Par 1 | Tse | 19,35684 | 19 | 1,408853 | ,323213 |
| | Tce | 10,35789 | 19 | 1,329555 | ,305021 |

Tabla N°11: Estadísticas de muestras relacionadas

Fuente: Autores

Correlaciones de muestras relacionadas

| | | N | Correlación | Sig. |
|-------|-----------|----|-------------|------|
| Par 1 | tse y tce | 19 | ,131 | ,592 |

Tabla N°12: Correlaciones de muestras relacionadas

Fuente: Autores

Prueba de muestras relacionadas

| | Diferencias relacionadas | | | | | t | gl | Sig. (bilatera l) | |
|-------|--------------------------|-------------------|--------------------------|---|----------|----------|--------|-------------------|------|
| | Media | Desviación típica | Error típico de la media | 95% Intervalo de confianza para la diferencia | | | | | |
| | | | | Inferior | Superior | | | | |
| Par 1 | tse – tce | 8,998947 | 1,805737 | ,414265 | 8,128610 | 9,869285 | 21,723 | 18 | ,000 |

Tabla N°13: Prueba de muestras relacionadas

Fuente: Autores

Los grados de libertad (gl) es 18

Criterio a tomar en cuenta en la prueba de muestras relacionadas:

- Si la probabilidad obtenida de P-valor $\leq \alpha$ se rechaza H_0 y se acepta H_1
- Si la probabilidad obtenida de P-valor $> \alpha$ se rechaza H_1 y se acepta H_0 .

La variable tiempo antes y después de la prueba disminuye significativamente, comparando las medias en la tabla.

En la **tabla N°13** se muestra que gl es 18, comprobando con la tabla de distribución t el valor es 1.734 mayor que $\alpha = 0.05\%$ y tomando en cuenta el criterio de decisión de la prueba t student se llega a la conclusión que se rechaza H_0 y se acepta H_1 .

H1 Con el entrenador de microcontroladores de comunicación wifi mejorará la eficiencia en el desarrollo de las practicas.

En la **figura N°59** se puede apreciar la curva de normalidad y las zonas de H0 y H1.

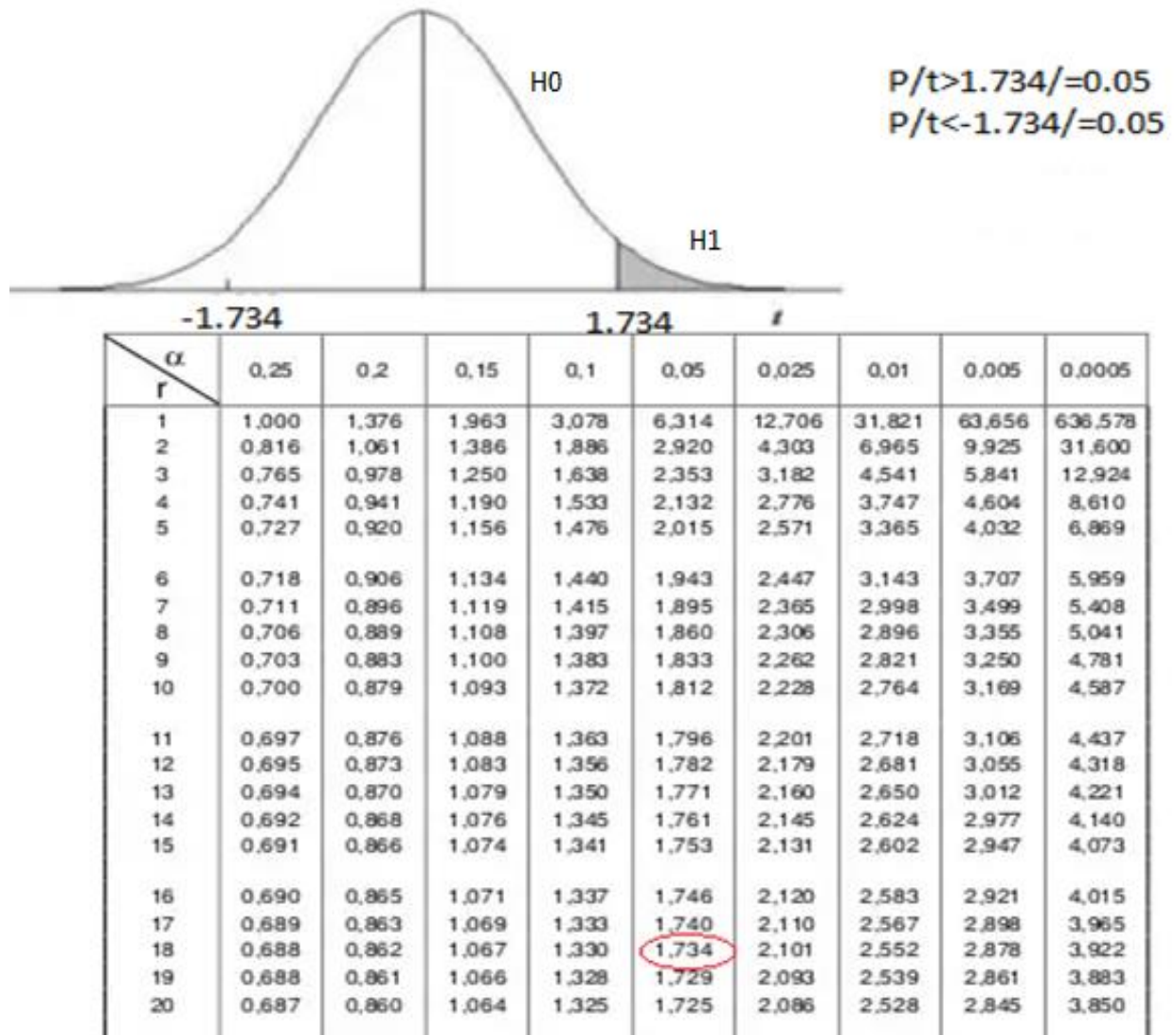


Figura N°59: Aceptación de H1 y rechazo de H0

Fuente: Autores

CAPÍTULO III

RESULTADOS

A continuacion

detalla las pruebas realizadas para configuracion del modulo wifly y pruebas de comunicaci3n

3.1. PRUEBAS DE CONECCCI3N

Se realiza la configuraci3n del modulo wifly con dhcp para unir a la red para con una direccion ip dinamica.

Se procede a comprobar el envio de paquetes de datos realizando un ping en cmd a la direccion ip asignada al modulo wifly que es 192.168.1.24.

El modulo wifly esta alimentado con fuentes difetentes con el microcontrolador, al realizar ping a esta direccion se produjo el siguiente error con perdidas de paquetes como se muestra en la **Figura N°60**.

```
C:\Users\Vicente>ping 192.168.1.24
Haciendo ping a 192.168.1.24 con 32 bytes de datos:
Respuesta desde 192.168.1.24: bytes=32 tiempo=6ms TTL=255
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Respuesta desde 192.168.1.24: bytes=32 tiempo=5ms TTL=255

Estadísticas de ping para 192.168.1.24:
    Paquetes: enviados = 4, recibidos = 2, perdidos = 2
              (50% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Míximo = 6ms, Míximo = 6ms, Media = 5ms
```

Figura N°60: Prueba de error en cmd transmision del modulo wifly
Fuente: Autores

El resultado de esta prueba fue perdida de paquetes en un 50%

Luego se realizo ping extendido 192.168.1.24 -t, los resultados los resultados en cmd de windows se indica en la **Figura N°61**.

```
C:\Users\Uicente>ping 192.168.1.24 -t
Haciendo ping a 192.168.1.24 con 32 bytes de datos:
Respuesta desde 192.168.1.24: bytes=32 tiempo=93ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=106ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=6ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=6ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=3ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=6ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=8ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=3ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=13ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=5ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=7ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=10ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=10ms TTL=255
Tiempo de espera agotado para esta solicitud.
Respuesta desde 192.168.1.24: bytes=32 tiempo=23ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=24ms TTL=255
Tiempo de espera agotado para esta solicitud.
Respuesta desde 192.168.1.24: bytes=32 tiempo=7ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=7ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=7ms TTL=255
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Respuesta desde 192.168.1.24: bytes=32 tiempo=6ms TTL=255
Tiempo de espera agotado para esta solicitud.
Respuesta desde 192.168.1.24: bytes=32 tiempo=4ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=4ms TTL=255
Respuesta desde 192.168.1.24: bytes=32 tiempo=74ms TTL=255
```

Figura N°61 : Prueba de transmision de paquetes al darle un ping al módulo wifly

Fuente: Autores

El resultado es una perdida de paquetes, para solventar este problema al conectar las tierras comunes para ver la ver la transmision correcta de paquetes sin perdidas como se muestra en la **Figura N°62**:



Figura N°62: prueba correcta de envio de paquetes

Fuente: Autores

3.2. PRUEBAS DE ALCANCE DE COBERTURA

Se probó el alcance en interiores del módulo wifly actuando como Access Point realizando la siguiente configuración en el módulo wifly :

Set wlan join 7
Set wlan channel 3
Set wlan ssid tesis
Set ip dhcp 4
Set ip address 192.168.2.1
Set ip net 255.255.0.0
Set ip gateway 192.168.2.1
Save
Rebot

Se verifico el alcance de la conertura realizando ping al modulo wifly, primero hay que conectarse a la red TESIS que indica la **figura N°63**.



Figura N°63: Red TESIS modulo wifly como AP
Fuente: Autores

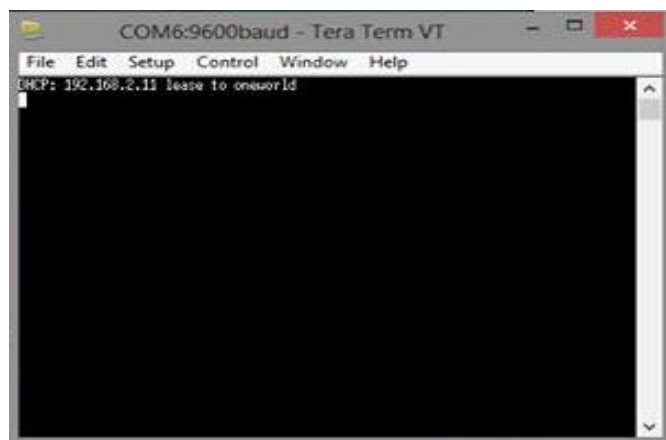
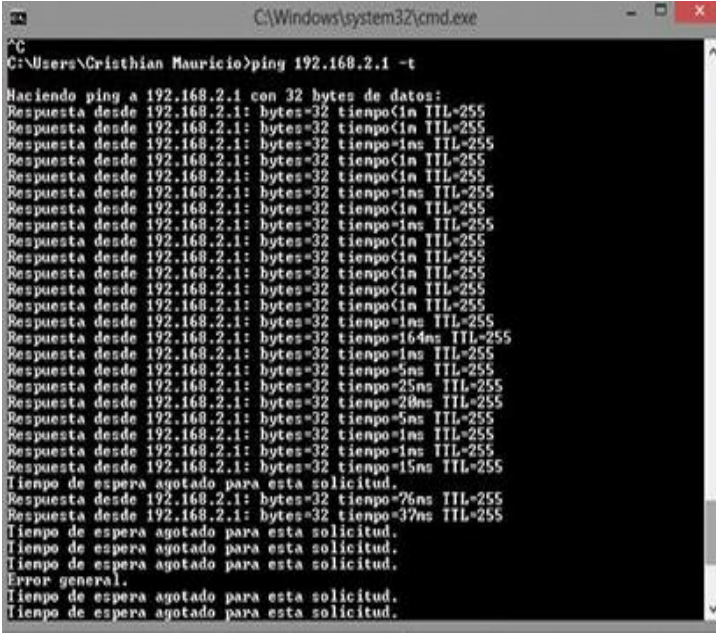


Figura N°64: conexión de un dispositivo al módulo Wifly exitoso
Fuente: Autores

Para ver que la conexión de los dispositivos al modulo wifly conectando el cable usb serial al modulo wifly com se muestra en la **Figura N°64** se observa la conexión de dispositivos wifly terminales se conectan al modulo en modo AP(Access Point)

Se realiza el ping extendido a la direccion del modulo wifly 192.168.2.1 para comprobar con el cmd de Windows los paquetes y a que distancia se pierde la comunicación se puede ver en la Figura N°65.



```
C:\Windows\system32\cmd.exe
C:\Users\Cristhian Mauricio>ping 192.168.2.1 -t
Haciendo ping a 192.168.2.1 con 32 bytes de datos:
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo<1m TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=164ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=5ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=25ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=20ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=5ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=15ms TTL=255
Tiempo de espera agotado para esta solicitud.
Respuesta desde 192.168.2.1: bytes=32 tiempo=76ms TTL=255
Respuesta desde 192.168.2.1: bytes=32 tiempo=37ms TTL=255
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Error general.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
```

Figura N°65: Prueba de conexión con el ap(modulo wifly) en interior
Fuente: Autores

Los resultados fueron que se perdía los paquetes a los 8 metros en interior con obstáculos

La siguiente prueba se realizó en exteriores para comprobar el alcance de conexión del modulo wifly actuando como Access Point, se realizó un ping en cmd de Windows para comprobar el alcance en la transmisión de paquetes entre el modulo wifly y una laptop.

En la **Figura N°66** se puede observar en envío de paquetes y la pérdida de paquetes se da cuando sale del área de cobertura que en esta prueba fue de 35m de promedio al realizar las pruebas en el espacio abierto usando un pin extendido a la dirección ip del modulo wifly que actúa como Punto de Acceso.

3.3. ANÁLISIS FINANCIERO

| Costos | Descripción | Valor Usd. |
|---------------------------------|---|-------------------|
| Hardware | Computador | 0 |
| | GLCD | 320 |
| | Módulos ZigBee | 80 |
| | Módulos IP | 100 |
| | Módulos GPS celular | 150 |
| | Módulos WiFi | 130 |
| | Módulos Bluetooth | 40 |
| | Módulos Transeiver | 100 |
| | Microcontroladores | 40 |
| | Pulsadores | 5 |
| | LCD | 8 |
| | Protoboard | 45 |
| | Resistencias | 5 |
| | Condensadores | 5 |
| | Cristal de cuarzo | 2 |
| | LEDS - RGB | 10 |
| | Potenciómetros | 5 |
| | Cables | 5 |
| | Teclado | 15 |
| | Trabajos en Plaquetas de cobre | 400 |
| Trabajos en acrílico | 200 | |
| Software | MikroC | 0 |
| | Proteus 8 Professional | 0 |
| | Micro C PRO for Pic | 0 |
| | AVR Studio | 0 |
| Proyecto | Transporte | 100 |
| varios | Cables, baquelita, estaño, pasta, Cautín. | 300 |
| Subtotal | | 2065 |
| Imprevistos 10% subtotal | | 265,50 |
| Total | | 2330,50 |

Tabla N°15: Análisis Financiero

Fuente: Los autores

Nota.- El valor estimado se multiplica por dos, quedando como valor final \$4661.00 dólares.

CAPÍTULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES.

- El monitoreo de sensores en un laboratorio usando el módulo wifly permite la transmisión de datos sin pérdidas a una aplicación dentro de la red WLAN.
- El entrenador de microcontroladores para comunicación Wi-Fi permite el desarrollo de sistemas embebidos, para los estudiantes de la escuela de Ingeniería Electrónica y telecomunicaciones.
- El entrenador de microcontroladores es de arquitectura abierta y permite el desarrollo de laboratorios con cualquier tipo de microcontrolador.
- El manual de prácticas tiene las configuraciones, circuitos, y procedimiento para la realización de laboratorios.
- El módulo Wifly tiene cobertura inalámbrica de 35 metros en exteriores y de 8 metros en interiores.
- Para que la aplicación pueda recibir datos inalámbricos de los módulos Wiflys tienen que estar configurados la ip y el puerto.

4.2. RECOMENDACIONES

- El módulo Wifly trabaja con el estándar IEEE 802.11 b/g a una frecuencia de 2.4 GHz.
- En la configuración del módulo Wifly es necesario tener el cable USB serial (TTL) o crear una red ad hoc, software Teraterm y leer el datasheet para realizar cada configuración, tener la guía de prácticas para realizar las conexiones adecuadas.
- Evitar cualquier tipo de ruido en el sistema al momento de configurar el módulo Wifly es recomendable tener una misma fuente, recuerde que si utiliza dos fuentes distintas hay que poner tierras comunes.
- En la configuración del módulo Wifly con el cable USB serial (TTL) se debe desconectar todas las alimentaciones de los circuitos que estén asociados al módulo.
- En la configuración del módulo Wifly como Punto de Acceso se debe trabajar en un canal no saturado, utilizar el software WiflyAnalyzer que proporciona la información detallada del espectro inalámbrico.
- Para entender el manual de usuario el estudiante debe tener conocimientos básicos de electrónica.

CAPÍTULO V

PROPUESTA

5.1. TÍTULO DE LA PROPUESTA

Diseño e implementación de un entrenador de microcontroladores para comunicación Wi-Fi.

5.2. INTRODUCCIÓN

La implementación del entrenador de microprocesadores para comunicación Wi-Fi sirve para la realización de prácticas con otros tipos de microcontroladores teniendo una arquitectura abierta, se puede utilizar el software de Visual Studio para realizar aplicaciones en la transmisión y recepción de datos, utilizando cualquier tipo de microcontrolador de la familia de microchip o de Atmel.

El módulo Wifly puede ser configurado en modo AP (Punto de Acceso), como servidor o cliente para la transmisión y recepción de datos de sensores para la activación y desactivación de las cargas.

El módulo Wifly se puede configurar mediante comandos de programación a través de un microcontrolador o también utilizando el software de Teraterm.

5.3. DISCUSIÓN

La investigación se enfoca en la configuración y prácticas de comunicación wifi usando un entrenador de microcontroladores que incorpora el dispositivo wifly.

Basándose en investigación teórica sobre los modos de configuración canales y frecuencias a las que funciona el protocolo de comunicación wi-fi se realiza pruebas de comunicación usando el microcontrolador para transmitir los datos inalámbricamente a la pc comprobando la estabilidad en la transmision y recepcion de datos.

Se realizaron pruebas de cobertura con el módulo wifly comprobando un alcance mayor en exteriores que en interiores por la atenuación e interferencia producida por los obstáculos.

El módulo wifly es un dispositivo práctico de alta funcionalidad útil para aplicaciones en el área de las telecomunicaciones, el módulo tiene la capacidad de transmitir y recibir datos inalámbricos en una Área de red Local inalámbrica (WLAN), se usará como un dispositivo para facilitar el aprendizaje y realización de sistemas embebidos con microcontroladores.

Se realizan prácticas usando diferentes modos de funcionamiento wifi, configurandolo en modo cliente, servidor, ad-hoc y en modo AP (Access Point o Punto de Acceso) puede asociar módulos wifly para controlar cargas y transmitir datos de sensores analogico y digitales.

Para visualizar y controlar los datos que se envían y reciben se realiza aplicaciones en visual studio.

En las prácticas escogidas se usaron diferentes modos de configuración para comunicación wi-fi asociandole a un router o actuando como AP, e interactuando con la aplicación de visual studio para el manejo e interpretación de los datos transmitidos de manera rápida, eficiente y estable demostrando la practicidad y confiabilidad en el uso del entrenador de microcontroladores en la comunicación wi-fi.

CAPÍTULO VI

6. OBJETIVOS

6.1. OBJETIVO GENERAL

- Diseño e implementación de un entrenador de microcontroladores para comunicación Wi-fi.

6.2. OBJETIVOS ESPECÍFICOS

- Desarrollar una aplicación de comunicación Wi-fi utilizando el entrenador de microcontroladores, para poder monitorear sensores en un laboratorio clínico.
- Diseñar el entrenador para el desarrollo de sistemas embebidos.
- Diseñar un entrenador con arquitectura abierta para el desarrollo de prácticas con otros tipos de microcontroladores.
- Realizar un manual de usuario con las configuraciones respectivas y los distintos tipos de prácticas que se pueden realizar en el entrenador.

6.3. FUNDAMENTACIÓN CIENTÍFICO –TÉCNICA

Contendrá una versión resumida y actualizada del estado del conocimiento que se encuentra el tema específico de la propuesta.

También conocidos como Microtrainers. El "circuito entrenador" es un dispositivo electrónico que sirve como apoyo en el funcionamiento del microcontrolador. El diseñador es el encargado de adaptar el modelo a la configuración del circuito entrenador, es decir, de establecer las conexiones físicas entre éste y el microcontrolador. (server-die-alc, 2002)

6.4. DESCRIPCIÓN DE LA PROPUESTA

El entrenador de microcontroladores para comunicación WiFi utilizando una aplicación en visual studio para recibir la transmisión de datos de los sensores con el módulo WiFly RN-XV 171, el microcontrolador se conecta al módulo wifly para que reciba y envíe los datos mediante UART de acuerdo a los parámetros de programación.

Se configura el módulo WiFly para realizar el trabajo adecuado, tener en cuenta que el módulo WiFly puede trabajar en distintas configuración.

6.5.DISEÑO ORGANIZACIONAL.

Se establecerá la estructura orgánica y funcional de la unidad administrativa que ejecutara la propuesta como se muestra en la **Figura N°67**.

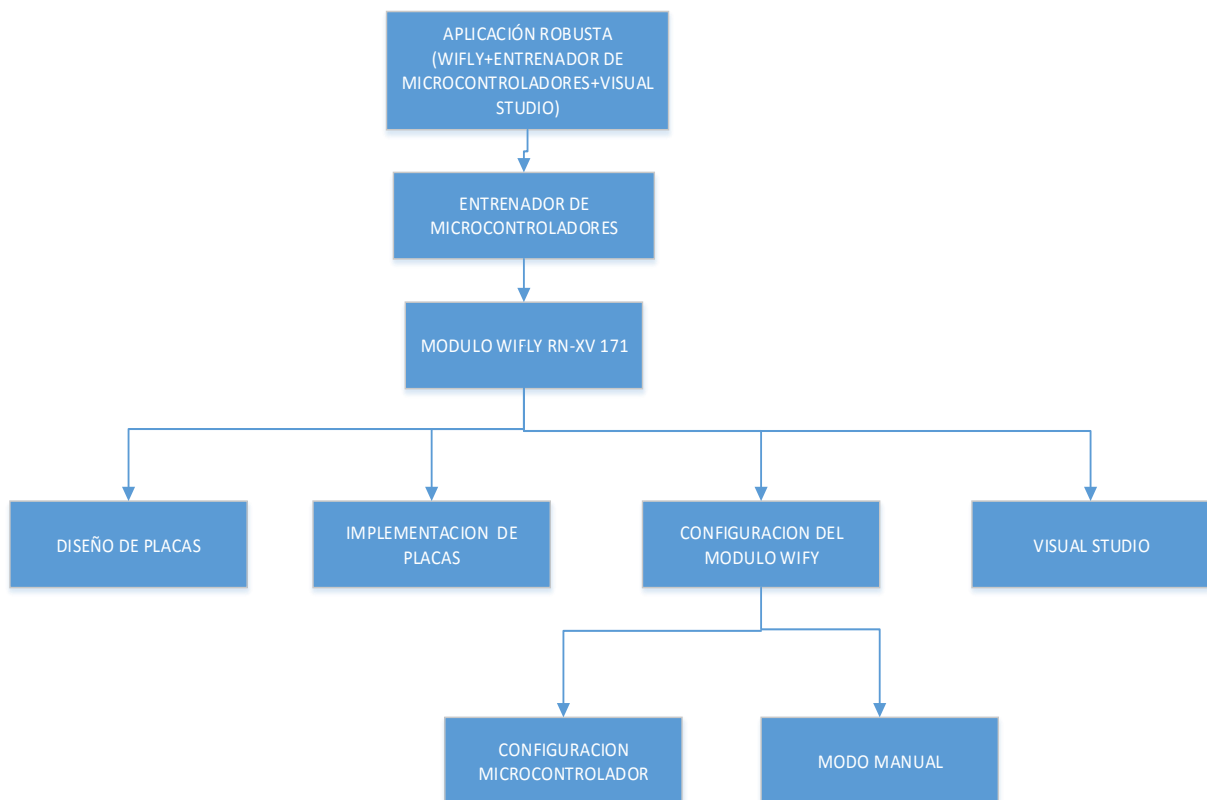


Figura N°67: diseño organizacional del proyecto

Fuente: Autores

CAPITULO VII

BIBLIOGRAFÍA

- ANIBAL, P. B. (2007). HISTORIA DE LOS MICROCONTROLADORES. En P. B. ANIBAL, *MODULO PARA PROBAR EL FUNCIONAMIENTO DE LOS PROGRAMAS GRABADOS EN EL PIC 16F84A* (pág. 107). QUITO.
- Caceres, L. A. (21 de Mayo de 2013). *MANUAL DE REDES DE COMPUTADORAS*. Obtenido de MANUAL DE REDES DE COMPUTADORAS, recuperado: http://yoprofesor.ecuadorsap.org/wp-content/uploads/2013/05/manual_redes.pdf
- Cajamarca, A. (14 de 03 de 2007). *Funcionamiento del pic 18f2550*. Obtenido de Funcionamiento del pic 18f2550, Recuperado: <http://bibdigital.epn.edu.ec/bitstream/15000/1989/1/CD-1095.pdf>
- CYBERCOM. (05 de 11 de 2013). *REDES LAN INALAMBRICAS (Wireless LAN)*. Obtenido de REDES LAN INALAMBRICAS (Wireless LAN), Recuperado: http://www.cybercom-cw.com.ar/pdf/Cybercom_WLAN_Paper.PDF
- drpiltrafa. (2010). WiFiClub. *La red Wifly mas amplia del pais, Recuperado, 22*.
- Flickenger, R. (2010). *eslared.org*. Obtenido de http://www.eslared.org.ve/walc2012/material/track1/05-Introduccion_a_las_redes_WiFi-es-v2.3-notes.pdf
- Instruments, T. (3 de 9 de 2015). *LM35 Precision Centigrade Temperature Sensors*. Obtenido de LM35 Precision Centigrade Temperature Sensors, Recuperado: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- microchip. (29 de 11 de 2001-2013). *microchip*. Obtenido de microchip: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>
- Microsoft. (19 de 6 de 2015). *VisualStudio*. Obtenido de VisualStudio, Recuperado: <https://www.visualstudio.com>
- MikroElectronica. (1998-2015). *El mundo de los microcontroladores - Microcontroladores PIC – Programación en C con ejemplos*. Recuperado el 10 de 05 de 2015, de El mundo de los microcontroladores - Microcontroladores PIC – Programación en C con ejemplos: <http://www.mikroe.com/chapters/view/79/capitulo-1-el-mundo-de-los-microcontroladores/>

msdn.microsoft.com. (s.f.). *Visual Studio*. Obtenido de [https://msdn.microsoft.com/es-es/library/vstudio/w0x726c2\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/vstudio/w0x726c2(v=vs.100).aspx)

Mullisaca, D. (13 de Marzo de 2014). *INTRODUCCION A LAS REDES WIFI*. Obtenido de INTRODUCCION A LAS REDES WIFI, Recuperado: <http://es.scribd.com/doc/212345456/05-Introduccion-a-Las-Redes-WiFi-Es-v2-3-Notes>

Revelo, M. (Diciembre de 2012). *control electronico de potencia*. Obtenido de control electronico de potencia: <http://bibdigital.epn.edu.ec/bitstream/15000/5514/1/CD-4633.pdf>

RN-WIFLYCR-UG. (30 de 03 de 2013). *WiFly Command Reference, Advanced*. Obtenido de WiFly Command Reference, Advanced: <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/WiFi/rn-wiflycr-ug-v1.2r.pdf>

SENSORS, H. (24 de 7 de 2015). *MQ6 SENSOR*. Obtenido de MQ6 SENSOR, Recuperado: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-6.pdf>

server-die-alc. (2002). *server-die*. Obtenido de server-die: [http://server-die.alc.upv.es/asignaturas/lased/2002-03/Soft Hard/hardware/entrenadores/index.html](http://server-die.alc.upv.es/asignaturas/lased/2002-03/Soft%20Hard/hardware/entrenadores/index.html)

teleprocesos. (7 de 12 de 2009). *uazuay*. Obtenido de uazuay, Recuperado: http://www.uazuay.edu.ec/estudios/sistemas/teleproceso/apuntes_1/laninalambricas.htm

UK, D.-R. (30 de 07 de 2010). *DHT11 Humidity & Temperature Sensor*. Obtenido de DHT11 Humidity & Temperature Sensor, Recuperado: <http://www.micropik.com/PDF/dht11.pdf>

www.uazuay.edu.ec. (23 de 08 de 2010). *Uazuay*. Obtenido de Uazuay, Recuperado: http://www.uazuay.edu.ec/estudios/teleprocesos/apuntes_1/laninmalambricas.htm



ANEXOS

ANEXO A

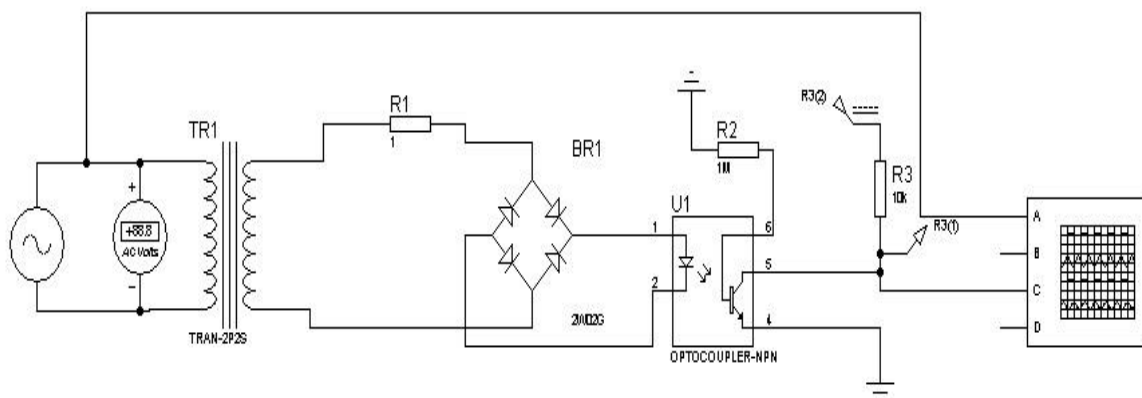
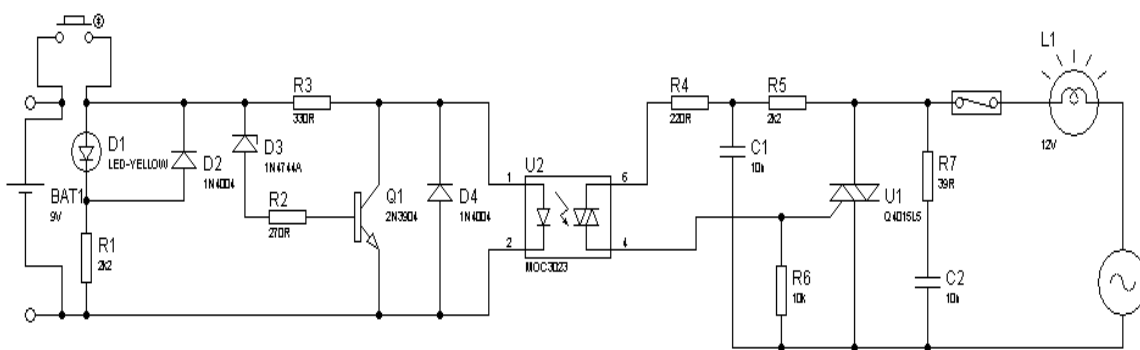
GLOSARIO DE TERMINOS

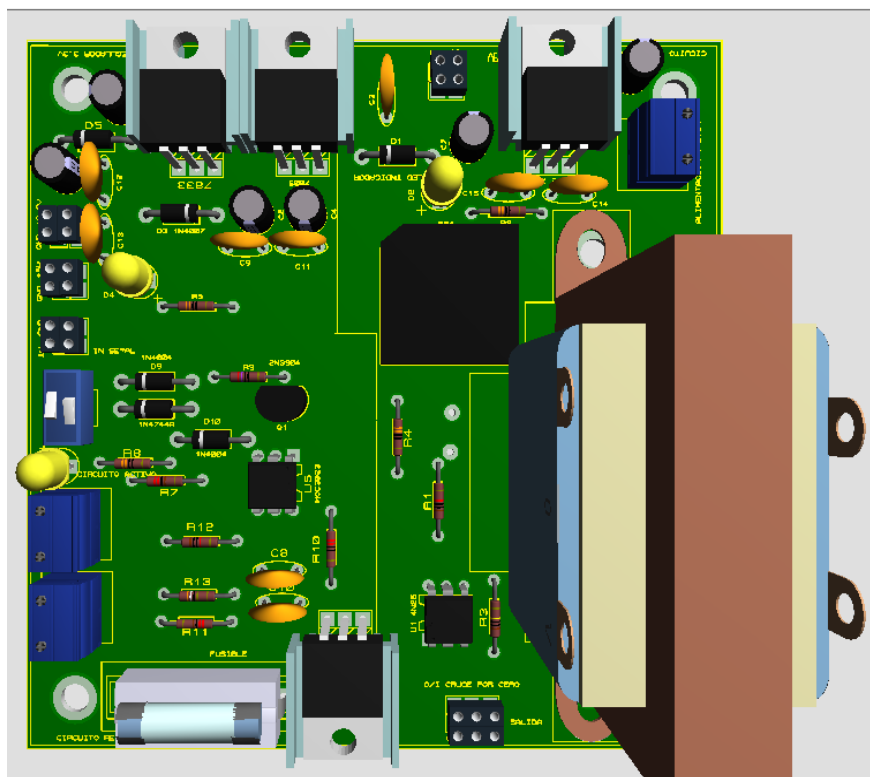
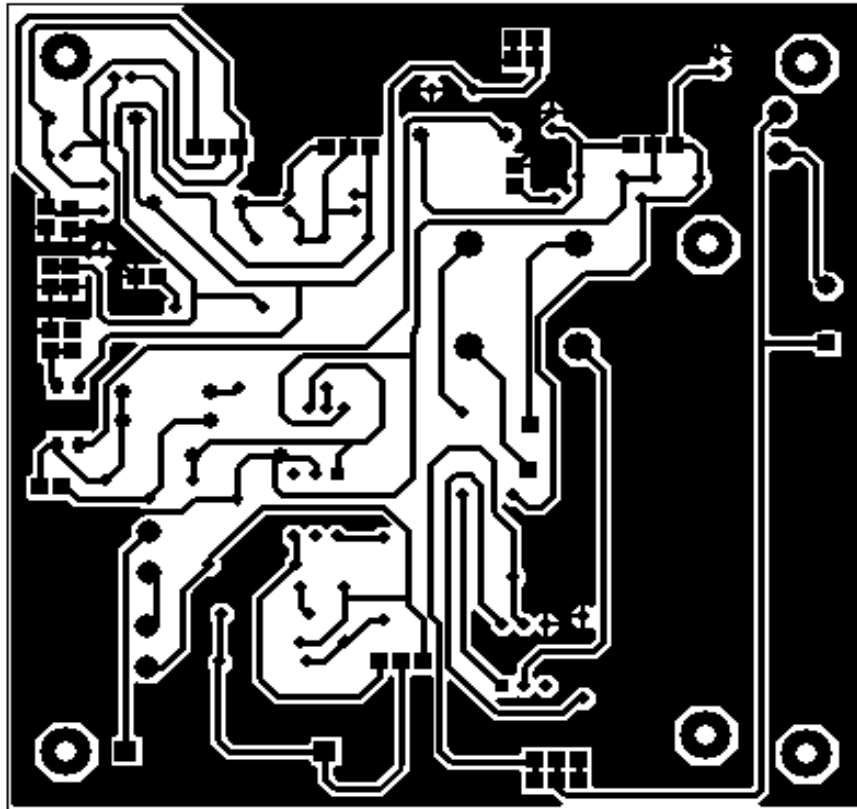
| | |
|-----|-------------------------|
| AP | Access Point |
| °C | Grados Centigrados |
| MQ6 | Sensor de gas |
| GLP | Gas licuado de petróleo |
| ppm | Partes por millón |
| Vin | Voltaje de entrada |
| Id | Corriente del diodo |
| Vd | Voltaje del diodo |
| RX | Recepción |
| Tx | Trasmisión |
| ip | protocolo de internet |
| Tse | Tiempo sin entrenador |
| Tce | Tiempo con entrenador |
| gl | número de estudiantes |

DISEÑO DE PLACAS DEL ENTRENADOR DE MICROCONTROLADORES:

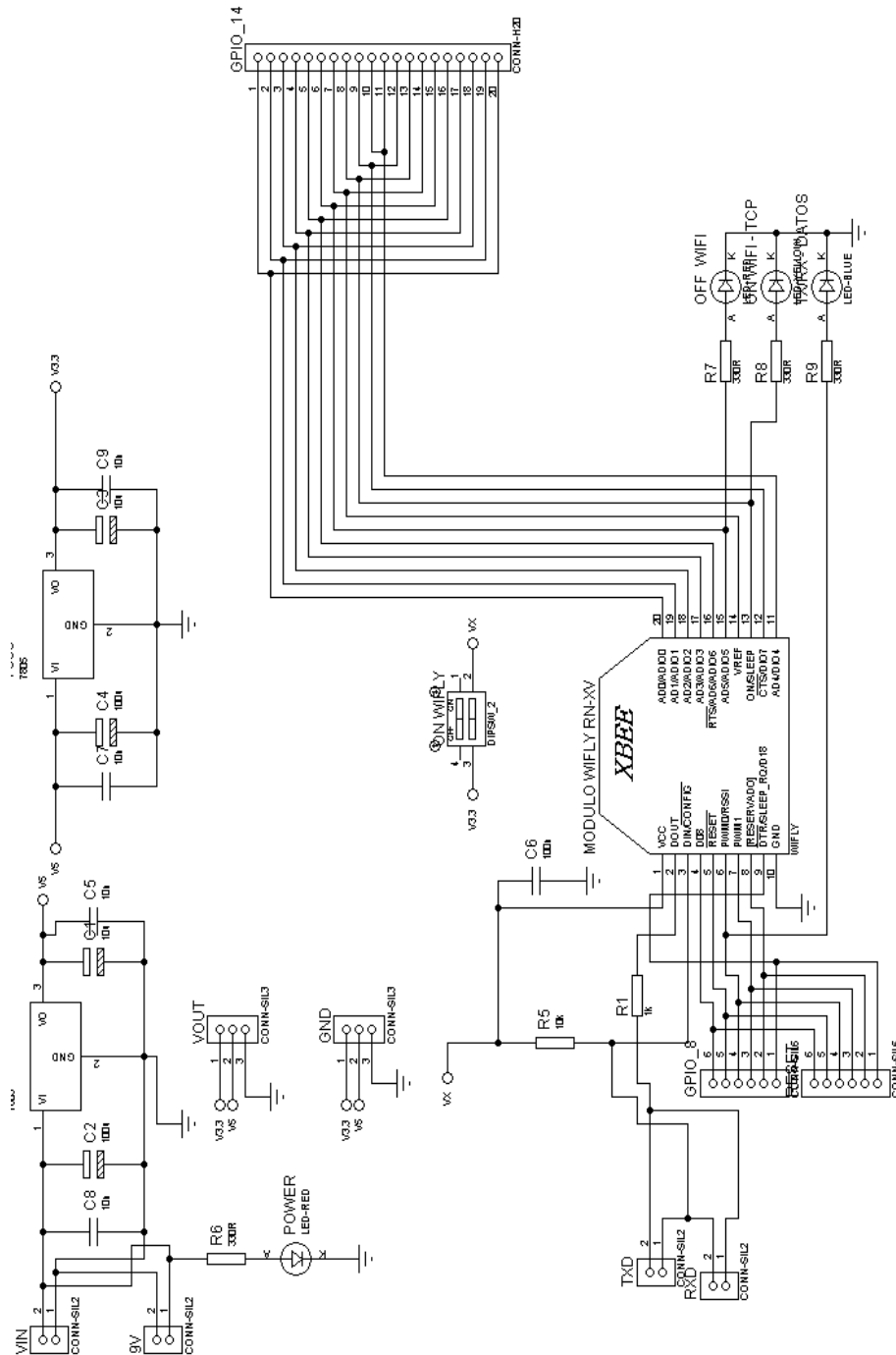
A continuación se muestra los diseños hechos en Isis y ares de las placas que conforman el entrenador de microcontroladores:

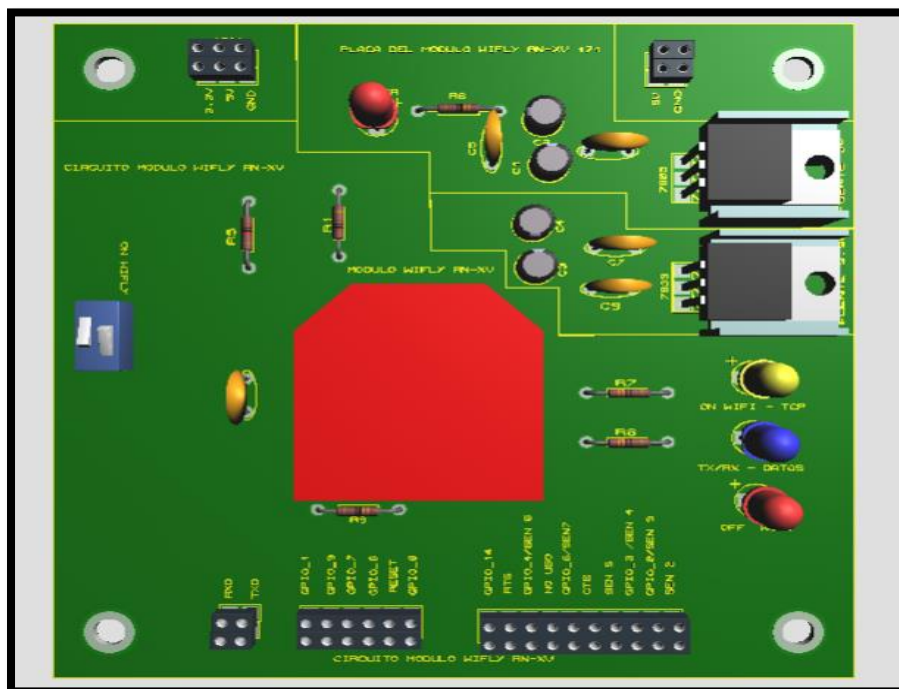
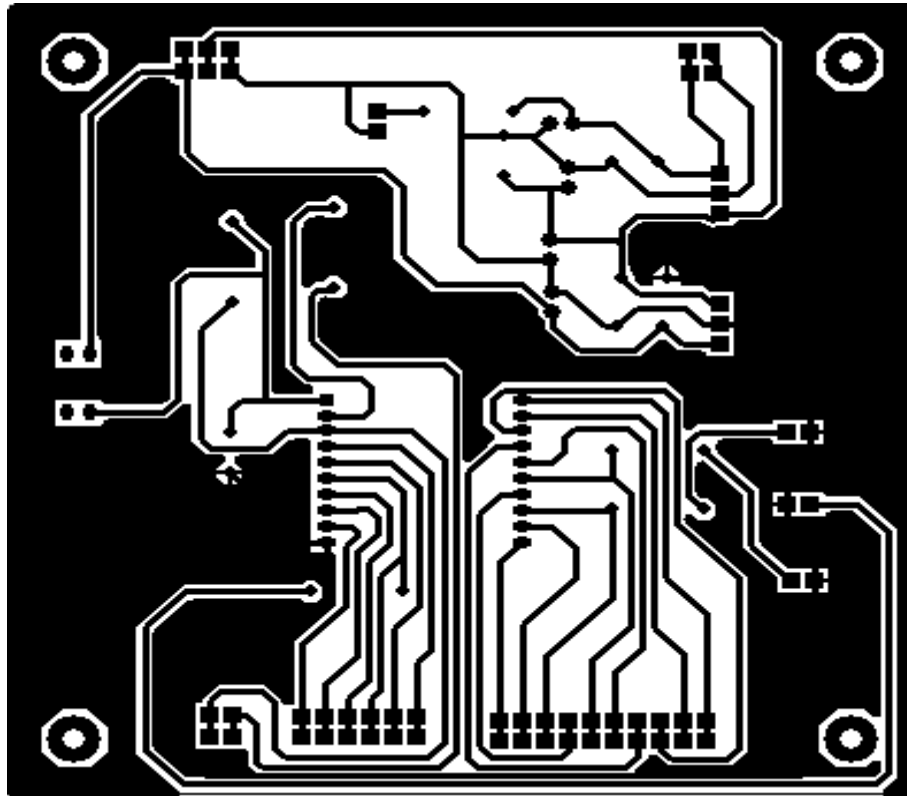
PLACA RELE DE ESTADO SOLIDO Y CRUCE POR CERO



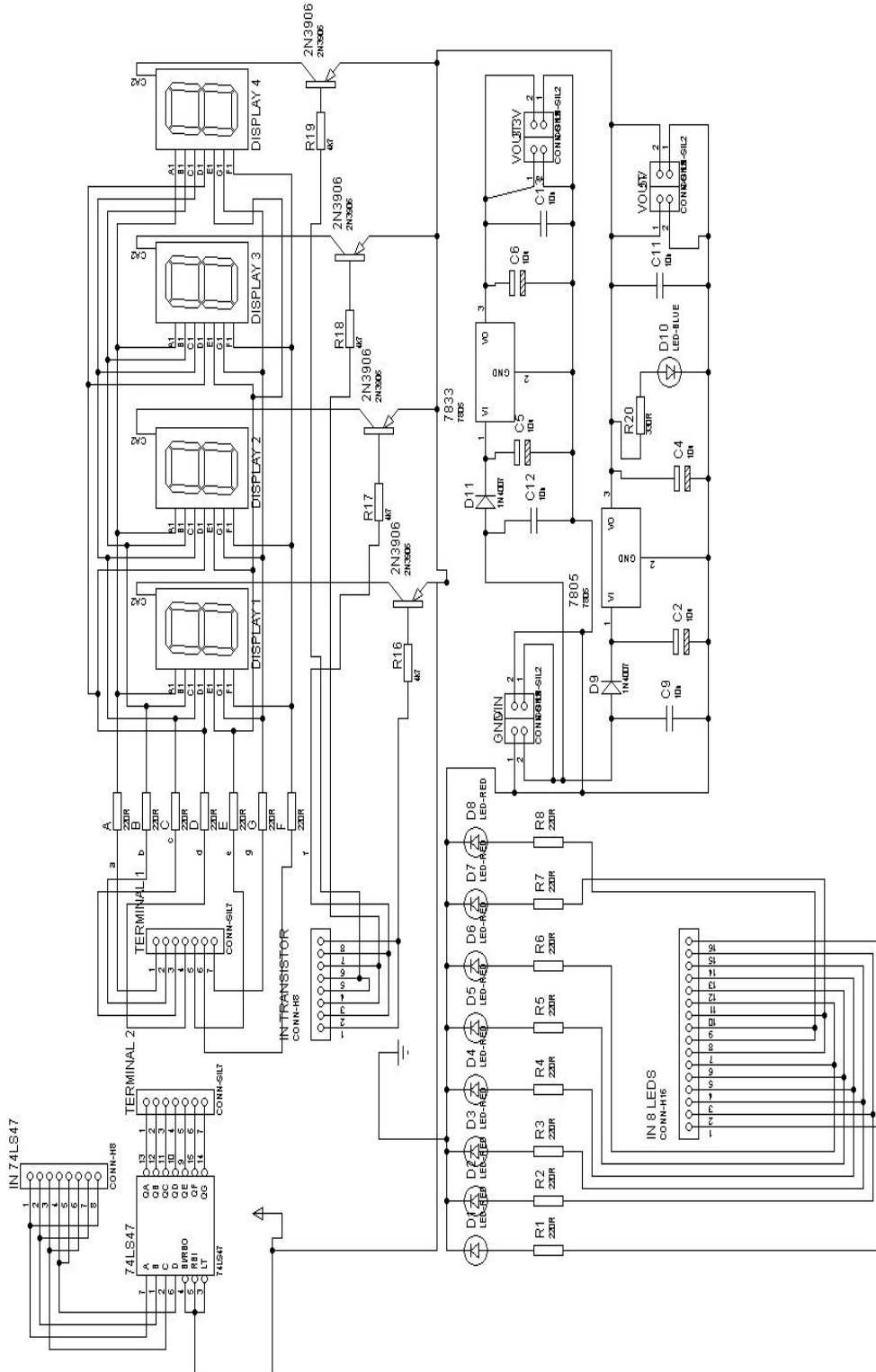


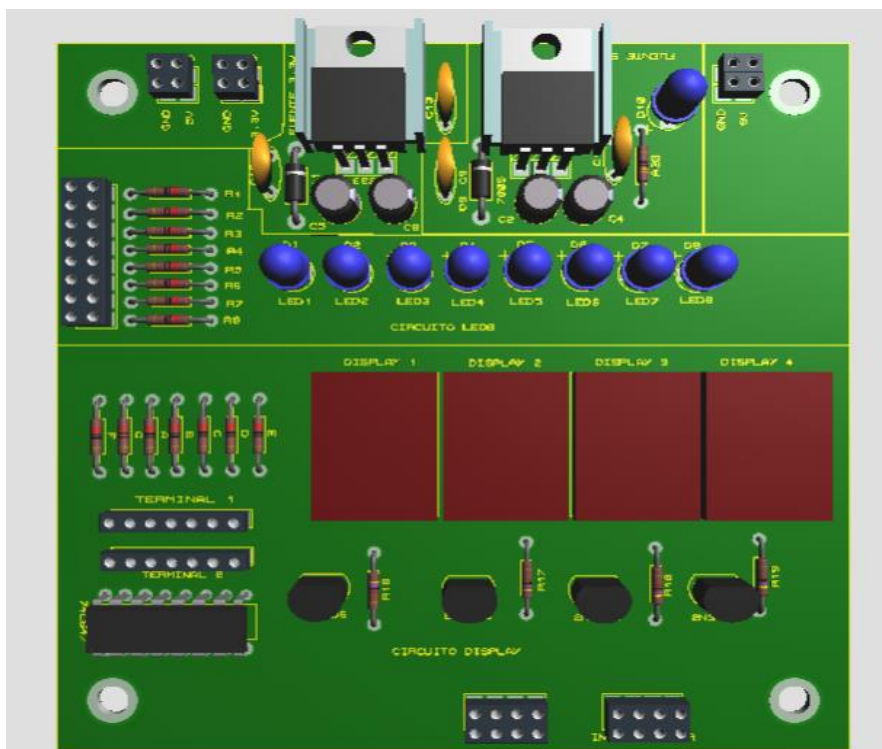
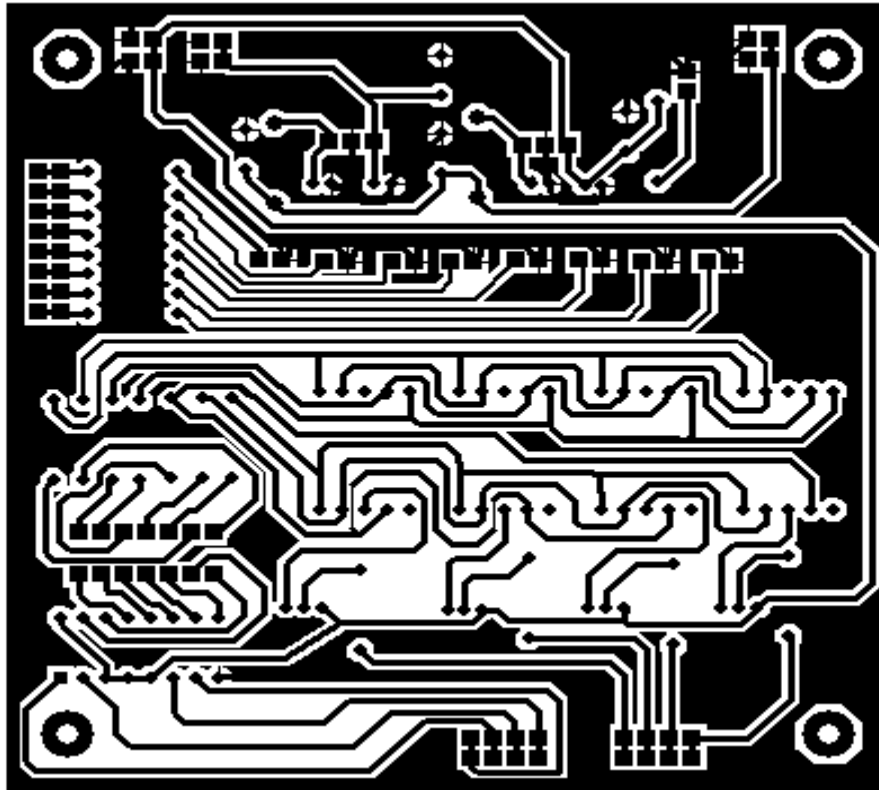
PLACA MODULO WIFLY RN-XV 171





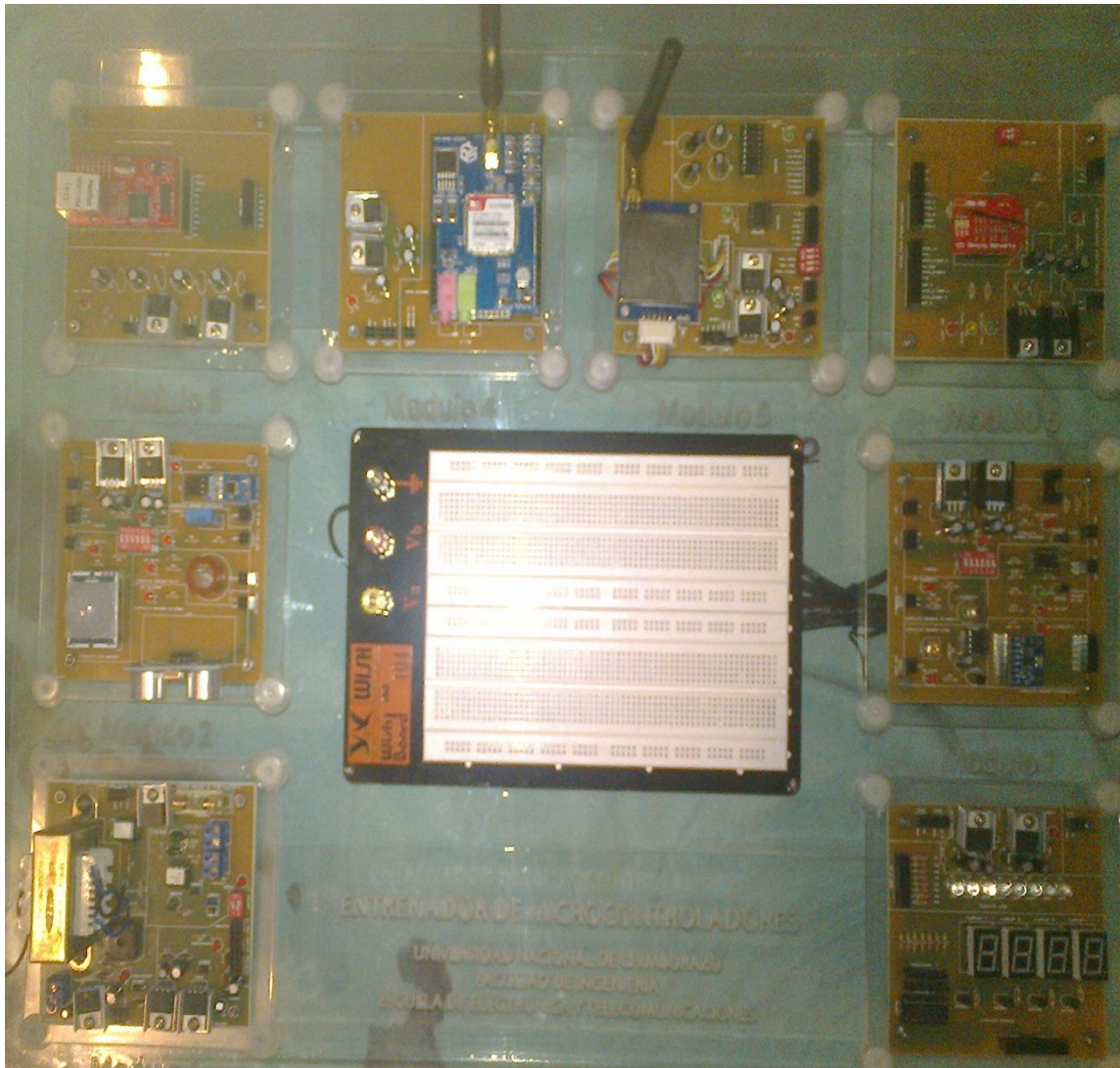
PLACA DISPLAYS Y LEDS





ENTRENADOR DE MICROCONTROLADORES APLICADO EN EL AREA DE LAS TELECOMUNICACIONES.

En la siguiente figura se muestra el entrenador de microcontroladores físicamente, para su utilización en el laboratorio de la escuela de Electrónica y Telecomunicaciones.



ANEXO B

LABORATORIO 1: CONFIGURACIÓN DEL MÓDULO WIFLY

Diagrama de conexión

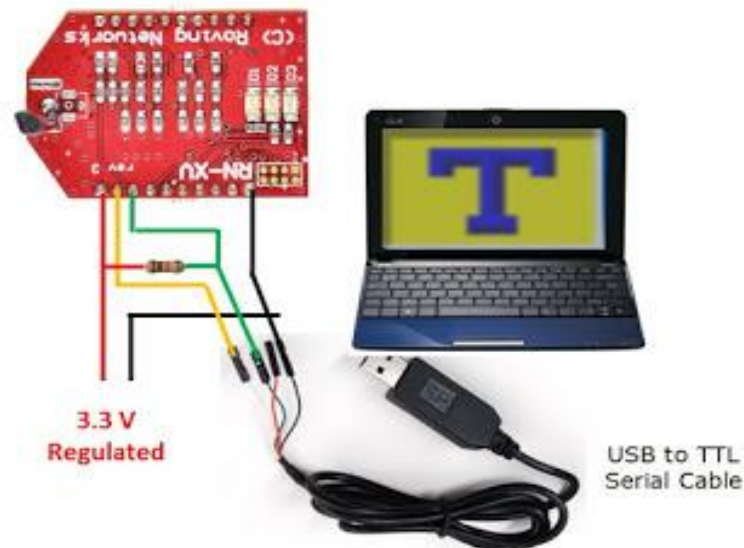


Figura N°1: Configuración

Objetivos de Aprendizaje:

- Conectar de manera adecuada los pines del módulo wifly con el cable USB serial TTL.
- Conectarse mediante el software Teraterm al dispositivo wifly.
- Configurar el módulo wifly con el software Teraterm.

Información Básica:

Para realizar la configuración se requiere los siguientes elementos:

- Cable USB serial como se muestra en la **Figura N°2** del cual el cable Blanco- RX, el cable Verde- TX, el cable Rojo de 5V de salida.
- PC de escritorio o laptop (**Figura N°3**).
- Módulo Wifly (**Figura N°4**).

- Software TERATERM (**Figura N°5**).



Figura N°2: Cable USB Serial



Figura N°3: PC

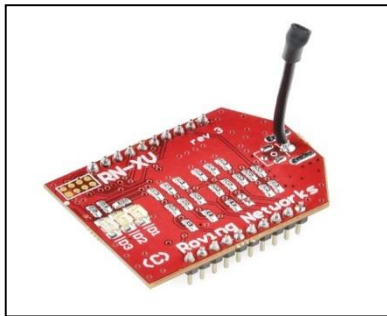


Figura N°4: módulo wifly



Figura N°5: Software Teraterm

Nota: Siempre se debe usar Teraterm para configurar el módulo wifly para que no se pierda la comunicación al configurarlo y se tenga una mejor estabilidad.

PRIMERA PARTE: CONEXIÓN CON TERATERM

1.- Conectar el módulo Wifly Con el cable USB serial TTL (Figura N°6)

- pin 1 (rojo): VDD-3.3V.
- pin2 (amarillo): UART_TX.
- pin3 (verde): UART_RX, con una resistencia externa de 10k pull-up to 3.3 V.
- pin 10 (negro): GND.

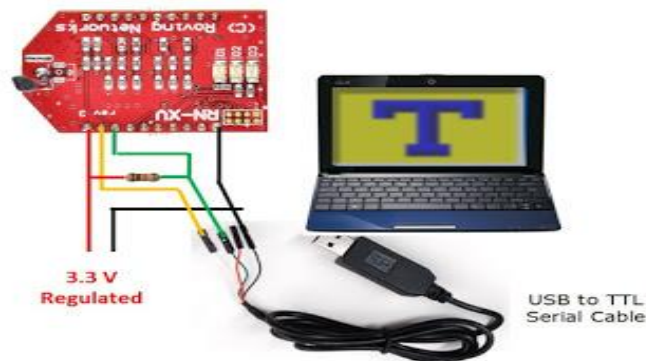


Figura N°6: Conexión del módulo wifly con USB serial

Conectar el cable USB serial TTL a la por el puerto USB, en ese momento los leds indicadores del módulo Wifly verde y rojo parpadean casi simultáneamente.

2.- Buscar el puerto COM que reconoce la PC al conectar el cable USB serial TTL

Ir al **PANEL DE CONTROL > ADMINISTRADOR DE DISPOSITIVOS (Figura N°7)**

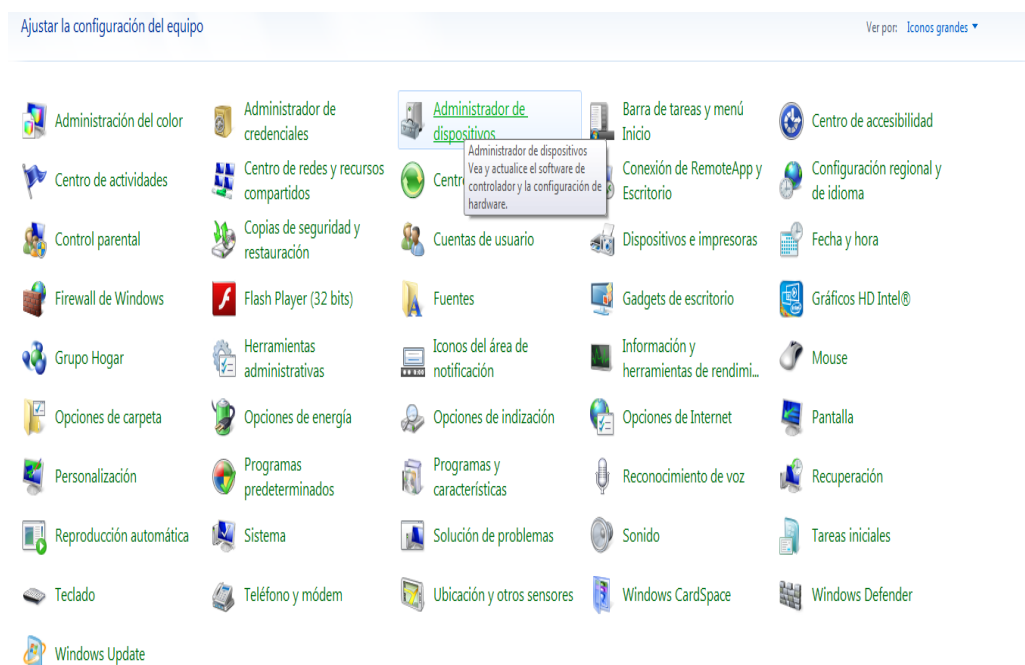


Figura N°7: Panel de control

Buscar el puerto del cable USB serial TTL, ver **Figura N°7** reconoce el puerto COM 6

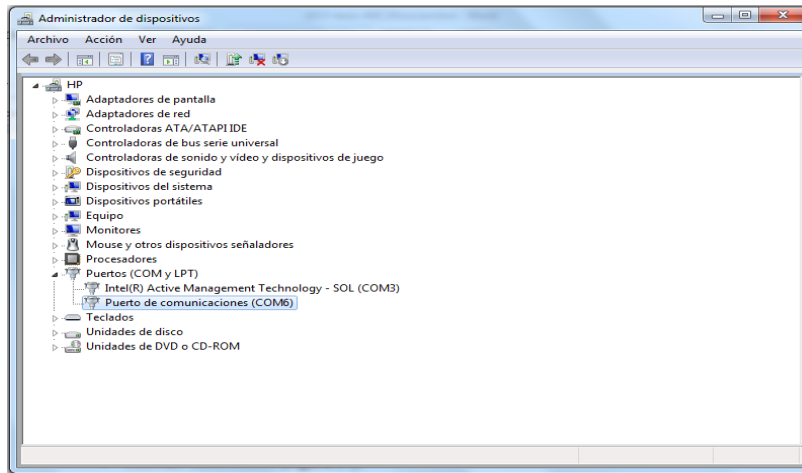


Figura N°8: Puerto COM serial

3.- Abrir el software Teraterm escoger la opción serial y el puerto que reconoce del USB serial, dar clic en ok (**Figura N°7**); se muestra la ventana en color negro en la cual se ingresan los comandos de configuración (**Figura N°8**).



Figura N°9: Pantalla en Teraterm a conexión serial (conexión exitosa)



Figura N°10: Pantalla en Teraterm listo para la configuración

4.- Teclar \$\$\$ y el wifly responde con un CMD para entrar a modo configuración

Nota: Los leds indicadores del módulo wifly al entrar en modo configuración parpadean con mayor frecuencia

5.- Escribir el comando **get everything** para ver las configuraciones por defecto del módulo wifly.

Nota: a partir de este paso se realizara cualquier configuración al módulo wifly

SEGUNDA PARTE: CONFIGURACIÓN

6.- Ingresar los comandos de configuración para que el módulo wifly se una a una red por DHCP.

Set wlan join 1 //Conectar automáticamente al router escogido.

Set wlan ssid HOME //Home es el nombre de la red a conectarse.

Set wlan phrase //Contraseña de la red asociada el módulo wifly

Save //Guardar la configuración realizada.

Reboot // Reiniciar el módulo wifly.



Al reiniciar el módulo wifly quedara conectado a la red HOME con ip dinámica, el led indicador quedara encendido y parpadeando

7.- Escribir en Teraterm **get everything** para saber los parámetros con los que quedo configurado el módulo wifly por DHCP



SISTEMA DE MONITOREO DE UN LABORATORIO CON ENTRENADOR DE MICROCONTROLADOR WIFLY

LABORATORIO 2: CONFIGURACION MODULOS WIFLY COMO AP Y COMO TERMINALES DE LA RED

Objetivos de Aprendizaje:

- Configurar el módulo Wifly con el software Teraterm.
- Configurar el módulo Wifly como Access Point.
- Configurar el módulo Wifly como terminal para conectarse a una aplicación servidor.
- Verificar parámetros de conexión.

Información Básica:

Para realizar la configuración se requiere los siguientes elementos:

- Cable USB serial TTL.
- PC de escritorio o laptop.
- 4 Módulo Wifly.
- Software TERATERM.
- 4 Microcontroladores.
- Sensores DTH11, LM35, MQ6

PRIMERA PARTE: CONFIGURACION MODULO WIFLY COMO AP

- 1.- Seguir los pasos del 1 al 5 de la **Practica 1: Configuración del módulo Wifly**.
- 2.- Ingresar en Teraterm el comando de configuración para establecer los parámetros para que el módulo Wifly actúe como AP (Access Point) **ver Figura N°1**.

Comandos de configuración para el módulo Wifly como AP:



| | |
|-----------------------------------|--------------------------------------|
| set wlan join 7 | comando para que el Wifly sea AP |
| set wlan channel 3 | seleccionamos el canal de trabajo |
| set wlan ssid | (Nombre de la red) nombre de la red |
| set ip dhcp 4 | el AP asigne direcciones ip por DHCP |
| set ip address 192.168.2.1 | ip del AP |
| set ip net 255.255.255.0 | mascara de la red |
| set ip gateway 192.168.2.1 | puerta de salida |
| save | guardar |
| reboot | reiniciar |

```
COM3 - Tera Term VT
File Edit Setup Control Window Help
$$$
CMD
<4.41>set wlan join 7
AOK
<4.41>set wlan channel 3
AOK
<4.41>set wlan ssid
AOK
<4.41>set ip dhcp 4
AOK
<4.41>set ip address 192.168.2.1
AOK
<4.41>set ip net 255.255.255.0
AOK
<4.41>set ip gateway 192.168.2.1
AOK
<4.41>save
Storing in config
<4.41>reboot
*Reboot*wifly-EZX Ver:4.41 Build: r1057, Jan 17 2014 10:23:54 on RN-171
MAC Addr:00:1e:c0:1b:7d:d0
*READY*
AP mode as wifly-EZX-d0 on chan 3
listen on 2000
DHCP Server Init
```

Figura N°1: Comandos de configuración en Teraterm

Una vez configurado el módulo Wifly los leds indicadores parpadean hasta quedar parpadeando el led verde esperando que un host se conecte (**Figura N°2**)



Figura N°2: Modulo Wifly configurado

SEGUNDA PARTE: CONFIGURACION MODULO WIFLY COMO TERMINAL

3.- Seguir los pasos del 1 al 5 de la **Practica 1: Configuración del módulo Wifly.**

4.- Ingresar en Teraterm los comando de configuración para establecer los parámetros para que el módulo Wifly actúe como terminal **para** conectarse a una aplicación servidor **ver Figura N°1.**

COMANDOS PARA EL MODULO WIFLY TRABAJE COMO TERMINAL:

set wlan join 1 se una a la red del AP

set wlan ssid TESIS Nombre de la red

set ip host 192.168.5.11 direccionamos al pc para la visualización de datos

set ip remote 1500 puerto donde escucha el programa

set ip proto 2 modo seguro para recibir paquetes de la dirección ip del host

set sys autoconn 1 se conecta inmediatamente al host remoto

set uart mode 2 para transmitir datos mediante uart



save

guardar

reboot

reiniciar

Nota: Configurar los tres módulos con la configuración como terminal, estos módulos transmitirán datos de sensores a una aplicación en la PC hecha en Visual estudio.

5.- ingresar el comando `get everything` para ver cómo se encuentra los parámetros de configuración del módulo Wifly.



LABORATORIO 3: PROGRAMACION DE LOS MICRONTROLADORES PARA SENSORES: DTH11, LM35, MQ6

Objetivos de Aprendizaje:

- Crear proyecto en MikroC.
- Programar para el microcontrolador 16f877a transmita y controle cargas del sensor dth11.
- Programar para el microcontrolador 16f877a transmita y controle cargas del sensor LM35.
- Programar para el microcontrolador 16f877a transmita y controle cargas del sensor MQ6.
- Subir el programa usando Pickit 2.0 al microcontrolador

Información Básica:

- Para realizar esta práctica es necesario:
- Tener instalado en la PC el software MikroC y Pickit 2.0
- Microcontrolador 16f887a
- Quemador de microcontrolador

Desarrollo:

PRIMERA PARTE: CREACION PROYECTO MIKROC

1.- Para crear un proyecto se abre MikroC (Figura N°1) >Project > New Project (Figura N°2) > next, seleccionar el microcontrolador 16F877A; seleccionar la frecuencia de 20 MHz y escoger el nombre y lugar donde guardar el proyecto

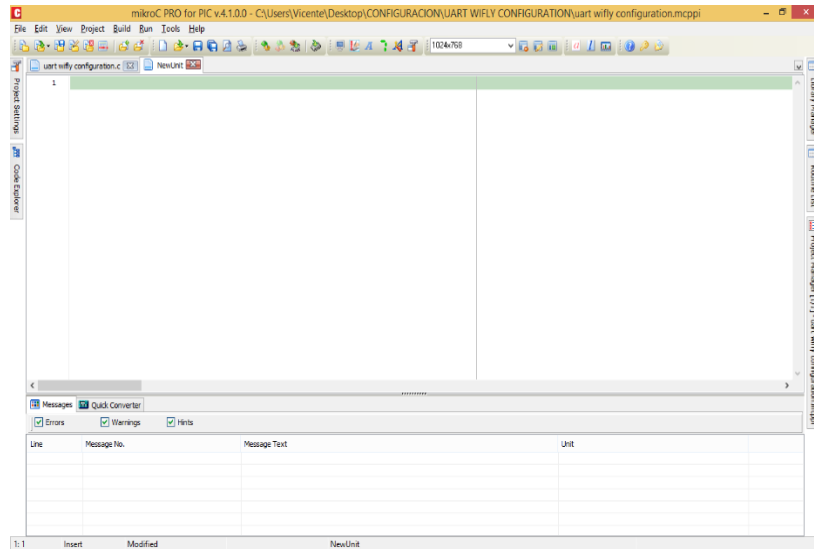


Figura N°1: Pantalla de inicio de MikroC

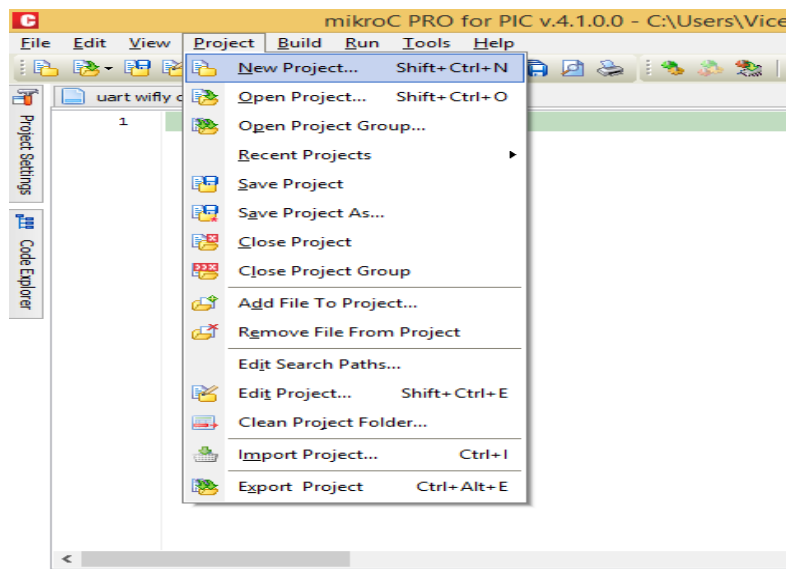


Figura N°2: crear nuevo proyecto

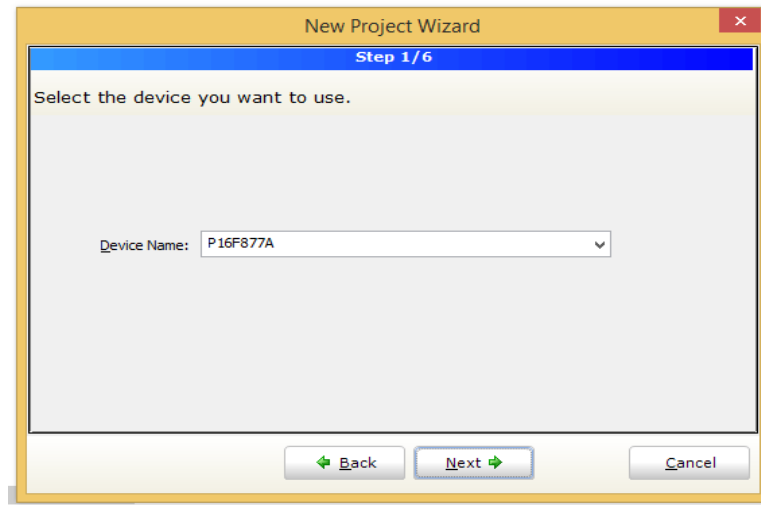


Figura N°3: seleccionar microcontrolador

En la Figura N°4 se indica la pantalla de inicio de MikroC

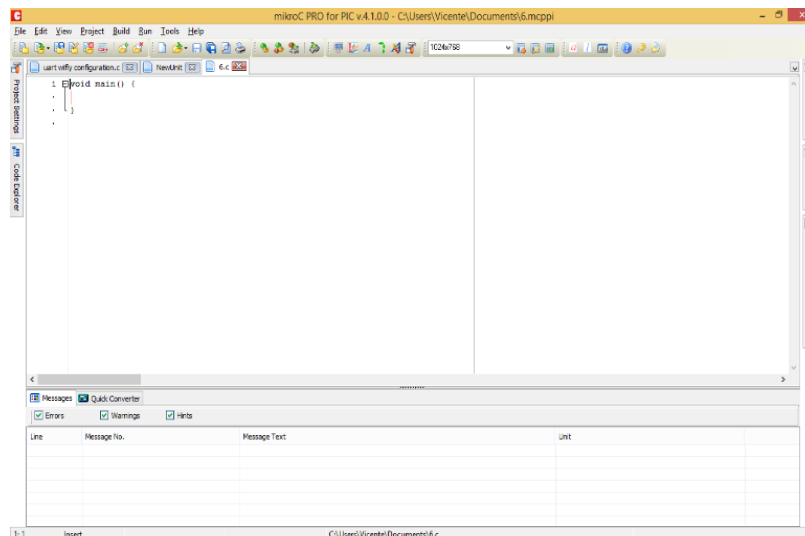


Figura N°4: Pantalla de inicio en MikroC con el nuevo proyecto creado

SEGUNDA PARTE: PROGRAMACION DE LOS MICROCONTRLADORES

2.- Se escribe la programación de acuerdo al sensor para cada microcontrolador

MICROCONTROLADOR CON SENSOR DTH11: la programación se copiará en el proyecto creado DTH11 posterior subido al microcontrolador del circuito con este sensor

```
unsigned int Radc,Radcmq,Radcmqd;  
unsigned int TemI,TemImq,TemImqd;
```



```
short uart_rd, uart_rdd;
float Tem, Temmq, Temmqd;
char Text[3];
char Textmq[3];
char Textmqd[3];
short Tlm, Tlmmq, Tlmmqd;
int dato=0;
sbit Data at RB0_bit;
sbit DataDir at TRISB0_bit;
unsigned short k;
unsigned short T_Byte1, T_Byte2, RH_Byte1, RH_Byte2;
char ptemp [4];
char phum;
char umbral = 0x1E;
char temp[] = "Tem= 00.0 C";
char hum[] = "Hum= 00.0 ";
void DHT11StartSignal(){
    DataDir = 0;
    Data = 0;
    Delay_ms(25);
    Data = 1;
    Delay_us(30);
    DataDir = 1; }
unsigned short DHT11CheckResponse(){
    k = 150;
    while(!Data){
        Delay_us(2);
        k--;
        if(k<1) return 0; // time out
    }
    k = 150;
    while(Data){
        Delay_us(2);
        k--;
        if(k<1) return 0; // time out
    }
    return 1;
}
unsigned short DHT11ReadByte(){
    int i;
    unsigned short num = 0;
    DataDir = 1;
    for (i=0; i<8; i++){
        while(!Data);
        Delay_us(40);
        if(Data) num |= 1<<(7-i);
    }
}
```



```
    while(Data);
  }
  return num;
}
void interrupt() {
  if (PIR1.RCIF == 1){
    uart_rd = UART1_Read();           // save the received caharacter,
    switch (uart_rd) {
      case 'o':
        dato=1;
        break;
      case 'x':
        dato=2;
        break;
      case 'a':
        dato=3;
        break;
      case 'b':
        dato=4;
        break;
      case 'c':
        dato=5;
        break;
      case 'd':
        dato=6;
        break;
    }
  }
  PIR1.RCIF = 0;    // clears RCIF
}
void main() {
  int ptemp;
  UART1_Init(9600);
  TRISD = 0x00;
  PORTD = 0x00;
  ADCON1 = 0b11000001;
  PIE1.RCIE = 1;    //HABILITAMOS LA INTERRUPCION POR USART CUANDO SE
  RECIVE UN DATO POR EL BUFFER
  INTCON.PEIE = 1;  //HABILITAMOS LAS INTERRUPCIONES PERIFERICAS
  INTCON.GIE = 1;  //HABILITAMOS INTERRUPCIONES GLOBALES
  //TRISC.RC1 = 0;
  //TRISC.RC2 = 0;
  while(1){
    while(dato==1) {
      DHT11StartSignal();
      if(!DHT11CheckResponse()) continue;

```



```
RH_Byte1 = DHT11ReadByte();
RH_Byte2 = DHT11ReadByte();
T_Byte1 = DHT11ReadByte();
T_Byte2 = DHT11ReadByte();
DHT11ReadByte(); /* Checksum */
// Set temp
temp[5] = T_Byte1/10 + 48;
temp[6] = T_Byte1%10 + 48;
temp[8] = T_Byte2/10 + 48;
ptemp= temp;
UART1_Write_Text(ptemp);
UART1_Write(10);
UART1_Write(13);
Delay_ms(300);
if(((T_Byte1) >=0x20)&&((T_Byte1) >=0x20)&&((T_Byte1) >=0x20))
{
PORTD.f2=1;
PORTD.f3=0;
}
if(((T_Byte1) <=0x19)&&((T_Byte1) <=0x19)&&((T_Byte1) <=0x19))
{
PORTD.f2=0;
PORTD.f3=1;
}
// Set hum
hum[5] = RH_Byte1/10 + 48;
hum[6] = RH_Byte1%10 + 48;
hum[8] = RH_Byte2/10 + 48;
UART1_Write_Text(hum);
//UART1_Write_Text(RH_Byte2);
UART1_Write(10);
UART1_Write(13);
if(((RH_Byte1) >=0x20)&&((RH_Byte1) >=0x20)&&((RH_Byte1) >=0x20))
{
PORTD.f0=1;
PORTD.f1=0;
}
if(((RH_Byte1) <=0x19)&&((RH_Byte1) <=0x19)&&((RH_Byte1) <=0x19))
{
PORTD.f0=0;
PORTD.f1=1;
}
Delay_ms(300);
}
while(dato==2) {
PORTD=0x00;
```



```
DHT11StartSignal();
if(!DHT11CheckResponse()) continue;
RH_Byte1 = DHT11ReadByte();
RH_Byte2 = DHT11ReadByte();
T_Byte1 = DHT11ReadByte();
T_Byte2 = DHT11ReadByte();
DHT11ReadByte(); /* Checksum */
// Set temp
temp[5] = T_Byte1/10 + 48;
temp[6] = T_Byte1%10 + 48;
temp[8] = T_Byte2/10 + 48;
ptemp= temp;
UART1_Write_Text(ptemp);
UART1_Write(10);
UART1_Write(13);
Delay_ms(300);
hum[5] = RH_Byte1/10 + 48;
hum[6] = RH_Byte1%10 + 48;
hum[8] = RH_Byte2/10 + 48;
UART1_Write_Text(hum);
UART1_Write(10);
UART1_Write(13);
Delay_ms(300);
}
while(dato==3) {
    portd.f0=1;
    portd.f1=0;
    DHT11StartSignal();
    if(!DHT11CheckResponse()) continue;
    RH_Byte1 = DHT11ReadByte();
    RH_Byte2 = DHT11ReadByte();
    T_Byte1 = DHT11ReadByte();
    T_Byte2 = DHT11ReadByte();
    DHT11ReadByte(); /* Checksum */
    // Set temp
    temp[5] = T_Byte1/10 + 48;
    temp[6] = T_Byte1%10 + 48;
    temp[8] = T_Byte2/10 + 48;
    ptemp= temp;
    UART1_Write_Text(ptemp);
    UART1_Write(10);
    UART1_Write(13);
    Delay_ms(300);
    hum[5] = RH_Byte1/10 + 48;
    hum[6] = RH_Byte1%10 + 48;
    hum[8] = RH_Byte2/10 + 48;
```



```
UART1_Write_Text(hum);
UART1_Write(10);
UART1_Write(13);
Delay_ms(300);
}
while(dato==4) {
portd.f0=0;
portd.f1=1;
DHT11StartSignal();
if(!DHT11CheckResponse()) continue;
RH_Byte1 = DHT11ReadByte();
RH_Byte2 = DHT11ReadByte();
T_Byte1 = DHT11ReadByte();
T_Byte2 = DHT11ReadByte();
DHT11ReadByte(); /* Checksum */
// Set temp
temp[5] = T_Byte1/10 + 48;
temp[6] = T_Byte1%10 + 48;
temp[8] = T_Byte2/10 + 48;
ptemp= temp;
UART1_Write_Text(ptemp);
UART1_Write(10);
UART1_Write(13);
Delay_ms(300);
hum[5] = RH_Byte1/10 + 48;
hum[6] = RH_Byte1%10 + 48;
hum[8] = RH_Byte2/10 + 48;
UART1_Write_Text(hum);
UART1_Write(10);
UART1_Write(13);
Delay_ms(300);
}
while(dato==5) {
portd.f2=1;
portd.f3=0;
DHT11StartSignal();
if(!DHT11CheckResponse()) continue;
RH_Byte1 = DHT11ReadByte();
RH_Byte2 = DHT11ReadByte();
T_Byte1 = DHT11ReadByte();
T_Byte2 = DHT11ReadByte();
DHT11ReadByte(); /* Checksum */
// Set temp
temp[5] = T_Byte1/10 + 48;
temp[6] = T_Byte1%10 + 48;
temp[8] = T_Byte2/10 + 48;
```




```
    ptemp= temp;
    UART1_Write_Text(ptemp);
    UART1_Write(10);
    UART1_Write(13);
    Delay_ms(300);
    hum[5] = RH_Byte1/10 + 48;
    hum[6] = RH_Byte1%10 + 48;
    hum[8] = RH_Byte2/10 + 48;
    UART1_Write_Text(hum);
    UART1_Write(10);
    UART1_Write(13);
    Delay_ms(300);
}
while(dato==6) {
portd.f2=0;
portd.f3=1;
    DHT11StartSignal();
    if(!DHT11CheckResponse()) continue;
    RH_Byte1 = DHT11ReadByte();
    RH_Byte2 = DHT11ReadByte();
    T_Byte1 = DHT11ReadByte();
    T_Byte2 = DHT11ReadByte();
    DHT11ReadByte(); /* Checksum */
    // Set temp
    temp[5] = T_Byte1/10 + 48;
    temp[6] = T_Byte1%10 + 48;
    temp[8] = T_Byte2/10 + 48;
    ptemp= temp;
    UART1_Write_Text(ptemp);
    UART1_Write(10);
    UART1_Write(13);
    Delay_ms(300);
    hum[5] = RH_Byte1/10 + 48;
    hum[6] = RH_Byte1%10 + 48;
    hum[8] = RH_Byte2/10 + 48;
    UART1_Write_Text(hum);
    UART1_Write(10);
    UART1_Write(13);
    Delay_ms(300);
}
}
}
```



MICROCONTROLADOR CON SENSOR LM35: la programación se copiará en el proyecto creado DTH11 posterior subido al microcontrolador del circuito con este sensor

```
unsigned int Radc;
unsigned int TemI;
float Tem;
char Text[3];
char TIm;
char uart_rd;
int dato=0;
void interrupt() {
if (PIR1.RCIF == 1){
uart_rd = UART1_Read();           // save the received caharacter,
switch (uart_rd) {
    case 'o':
        dato=1;
        break;
    case 'x':
        dato=2;
        break;
    case 'e':
        dato=3;
        break;
    case 'f':
        dato=4;
        break;

    }
}
PIR1.RCIF = 0;    // clears RCIF
}
void main() {
int ptemp;
UART1_Init(9600);
TRISD = 0x00;
PORTD = 0x00;
ADCON1 = 0b11000001;
PIE1.RCIE = 1;    //HABILITAMOS LA INTERRUPCION POR USART CUANDO SE
RECIVE UN
DATO POR EL BUFFER
INTCON.PEIE = 1; //HABILITAMOS LAS INTERRUPCIONES PERIFERICAS
INTCON.GIE = 1;  //HABILITAMOS INTERRUPCIONES GLOBALES
//TRISC.RC1 = 0;
//TRISC.RC2 = 0;
while(1){
```



```
while(dato==1) {
    Radc = ADC_Read(0);    //Lectura del canal 0 del ADC.
    Tem = 0.244*Radc;      //Uso de la ecuación (13.5).
    TemI = Tem;           //Se convierte el resultado a un número entero.
    IntToStr(TemI, Text ); //Se convierte el número entero a una cadena de caracteres.
    UART1_write_Text("temp=");
    UART1_write_Text(Text);
    UART1_Write(10);
    UART1_Write(13);
    delay_ms(500);
    Tlm=Text;
    if((Tlm>=0x20)&&(Tem>=0x20)){
        portd.f4=1;
        portd.f5=0;
    }
    else
    {
        portd.f4=0;
        portd.f5=1;
    }
}

while(dato==2) {
    PORTD=0x00;
    Radc = ADC_Read(0);    //Lectura del canal 0 del ADC.
    Tem = 0.244*Radc;      //Uso de la ecuación (13.5).
    TemI = Tem;           //Se convierte el resultado a un número entero.
    IntToStr(TemI, Text ); //Se convierte el número entero a una cadena de caracteres.
    UART1_write_Text("temp=");
    UART1_write_Text(Text);
    UART1_Write(10);
    UART1_Write(13);
    delay_ms(500);
}

while(dato==3) {
    portd.f4=1;
    portd.f5=0;
    Radc = ADC_Read(0);    //Lectura del canal 0 del ADC.
    Tem = 0.244*Radc;      //Uso de la ecuación (13.5).
    TemI = Tem;           //Se convierte el resultado a un número entero.
    IntToStr(TemI, Text ); //Se convierte el número entero a una cadena de caracteres.
    UART1_write_Text("temp=");
    UART1_write_Text(Text);
    UART1_Write(10);
    UART1_Write(13);
    delay_ms(500);
}
```



```
}  
while(dato==4) {  
portd.f4=0;  
portd.f5=1;  
Radc = ADC_Read(0); //Lectura del canal 0 del ADC.  
Tem = 0.244*Radc; //Uso de la ecuación (13.5).  
TemI = Tem; //Se convierte el resultado a un número entero.  
IntToStr(TemI, Text ); //Se convierte el número entero a una cadena de caracteres.  
UART1_write_Text("temp=");  
UART1_write_Text(Text);  
UART1_Write(10);  
UART1_Write(13);  
delay_ms(500);  
  
}  
}  
}
```

MICROCONTROLADOR CON SENSOR MQ6: la programación se copiará en el proyecto creado DTH11 posterior subido al microcontrolador del circuito con este sensor

```
unsigned int Radc,Radcmq;  
unsigned int TemI,TemImq;  
//short uart_rd;  
float Tem, Temmq;  
char Text[3];  
char Textmq[3];  
short TIm, TImmq;  
//char Text[3];  
//char TIm;  
char uart_rd;  
int dato=0;  
void interrupt() {  
if (PIR1.RCIF == 1){  
uart_rd = UART1_Read(); // save the received caharacter,  
switch (uart_rd) {  
  
case 'o':  
dato=1;  
break;  
case 'x':  
dato=2;  
break;  
case 'g':  
dato=3;
```



```
        break;
    case 'h':
        dato=4;
        break;
    }
}
PIR1.RCIF = 0;    // clears RCIF
}
void main() {
int ptemp;
UART1_Init(9600);
TRISD = 0x00;
PORTD = 0x00;
ADCON1 = 0b11000001;

PIE1.RCIE = 1;    //HABILITAMOS LA INTERRUPCION POR USART CUANDO SE
RECIVE UN DATO POR EL BUFFER
INTCON.PEIE = 1;  //HABILITAMOS LAS INTERRUPCIONES PERIFERICAS
INTCON.GIE = 1;  //HABILITAMOS INTERRUPCIONES GLOBALES
//TRISC.RC1 = 0;
//TRISC.RC2 = 0;
while(1){
while(dato==1) {
Radcmq = ADC_Read(1);    //Lectura del canal 0 del ADC.
Temmq = 0.00488*Radcmq;    //Uso de la ecuación (13.5).
TemImq = Temmq;    //Se convierte el resultado a un número entero.
IntToStr(TemImq, Textmq ); //Se convierte el número entero a una cadena de caracteres.
UART1_write_Text("MQ6=");
UART1_write_Text(Textmq);
UART1_Write(10);
UART1_Write(13);
delay_ms(500);
Tlmmq=Textmq;
if((Tlmmq>=0x02)&&(Temmq>=0x02)){
portd.f6=1;
portd.f7=0;
}
else
{
portd.f6=0;
portd.f7=1;
}
}
while(dato==2) {
PORTD=0x00;
Radcmq = ADC_Read(1);    //Lectura del canal 0 del ADC.
```



```
Temmq = 0.00488*Radcmq; //Uso de la ecuación (13.5).
TemImq = Temmq; //Se convierte el resultado a un número entero.
IntToStr(TemImq, Textmq); //Se convierte el número entero a una cadena de caracteres.
UART1_write_Text("MQ6=");
UART1_write_Text(Textmq);
UART1_Write(10);
UART1_Write(13);
delay_ms(500);
Tlmmq=Textmq; }
while(dato==3) {
portd.f6=1;
portd.f7=0;
Radcmq = ADC_Read(1); //Lectura del canal 0 del ADC.
Temmq = 0.00488*Radcmq; //Uso de la ecuación (13.5).
TemImq = Temmq; //Se convierte el resultado a un número entero.
IntToStr(TemImq, Textmq); //Se convierte el número entero a una cadena de caracteres.
UART1_write_Text("MQ6=");
UART1_write_Text(Textmq);
UART1_Write(10);
UART1_Write(13);
delay_ms(500); }
while(dato==4) {
portd.f6=0;
portd.f7=1;
Radcmq = ADC_Read(1); //Lectura del canal 0 del ADC.
Temmq = 0.00488*Radcmq; //Uso de la ecuación (13.5).
TemImq = Temmq; //Se convierte el resultado a un número entero.
IntToStr(TemImq, Textmq); //Se convierte el número entero a una cadena de caracteres.
UART1_write_Text("MQ6=");
UART1_write_Text(Textmq);
UART1_Write(10);
UART1_Write(13);
delay_ms(500);
} } }
```

TERCERA PARTE: SUBIR EL PROGRAMA AL MICROCONTROLADOR

Nota: Antes de realizar el paso 3 es necesario poner cada microcontrolador en el quemador de microcontroladores y conectarlo a la pc con el cable USB.

3.- Abrir Pickit 2.0, **File > Import File**, buscar el archivo de programación y subir de acuerdo al sensor, hacerlo para los tres microcontroladores (**Figura N°5**), clic en write y se

carga el programa en el microcontrolador una vez este cargado se visualiza como la **Figura N°6.**

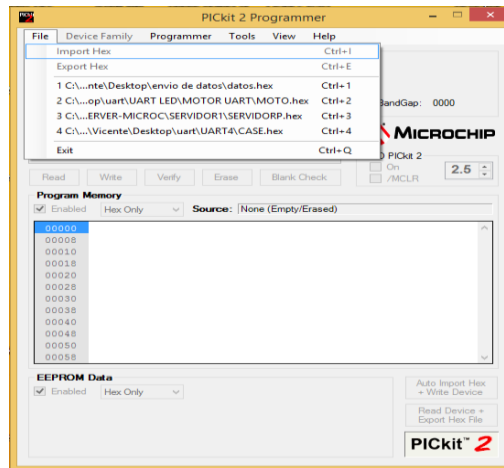


Figura N°5: cargar programa al Pickit 2

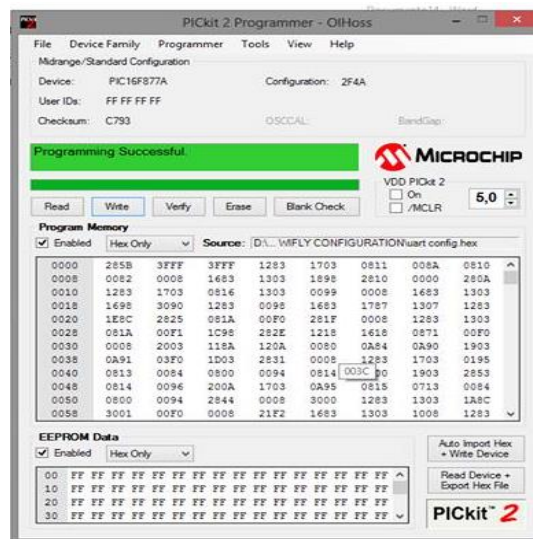


Figura N°6: Quemar microcontrolador en Pickit 2.0

4.- Armar los circuito simulado en Proteus de la Figura N°44 de la conexión del módulo Wifly con el microcontrolador, se conectara el cable USB serial a los pines 2 y 3 (TX, RX) del módulo Wifly para monitorear los comandos que enviara el microcontrolador mediante UART (Universal Transmisión Recepción Asíncronico), primero se alimentara el módulo Wifly con el cable USB serial



LABORATORIO 4: DESARROLLO DE APLICACIÓN EN VISUAL ESTUDIO PARA RECEPCION Y CONTROL DE CARGAS DE UN LABORATORIO

Objetivos de Aprendizaje:

- Crear un proyecto en Visual Studio
- Añadir clases en el proyecto.
- Crear la interfaz gráfica de la aplicación.
- Programar las funciones de la aplicación.
- Probar el funcionamiento de la aplicación.

Información Básica:

Para realizar la configuración se requiere los siguientes elementos:

- Pc o computadora de escritorio
- Tener instalado Visual Studio 2012

Desarrollo:

1.- Abrir Visual Studio 2012(El escritorio se muestra en la Figura N°1)

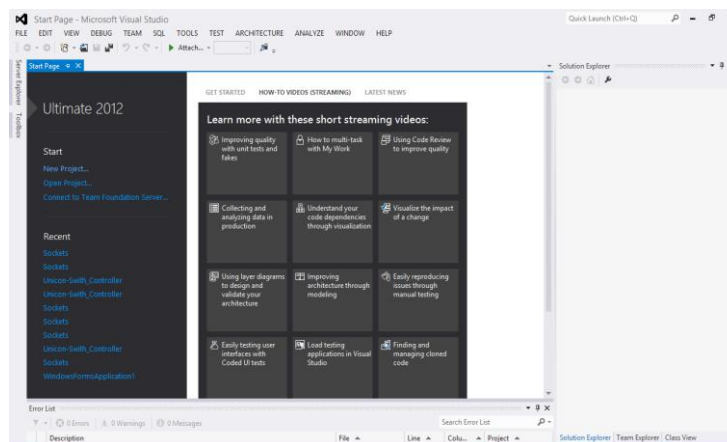


Figura N°1: Escritorio de Visual Studio 2012

2.-Escoger new project Wpf (Figura N°2), en la Figura N°3 esta la pantalla de inicio del proyecto Wpf.

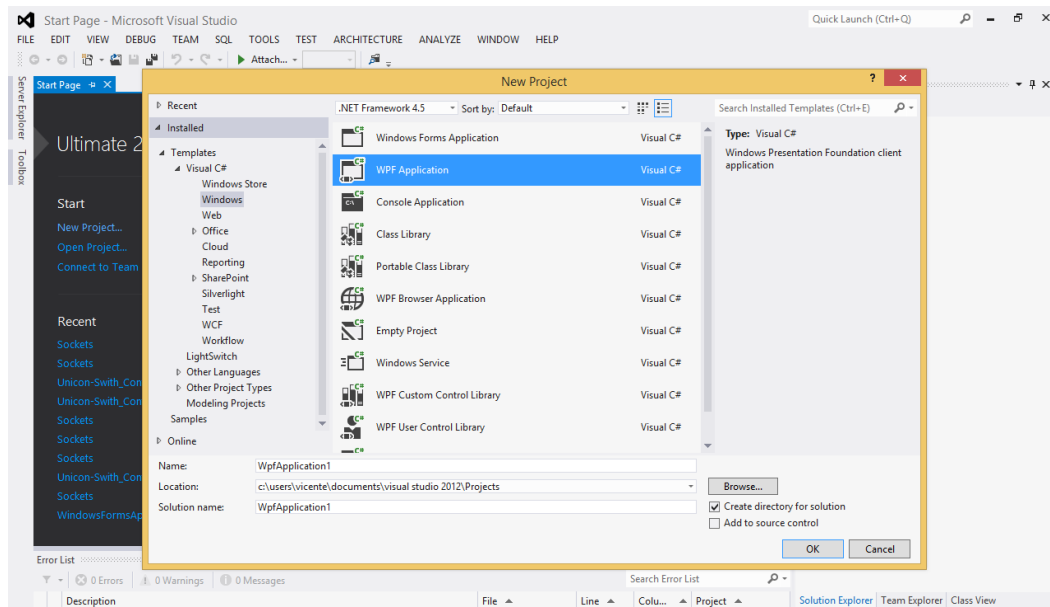


Figura N°2: Escoger Wpf

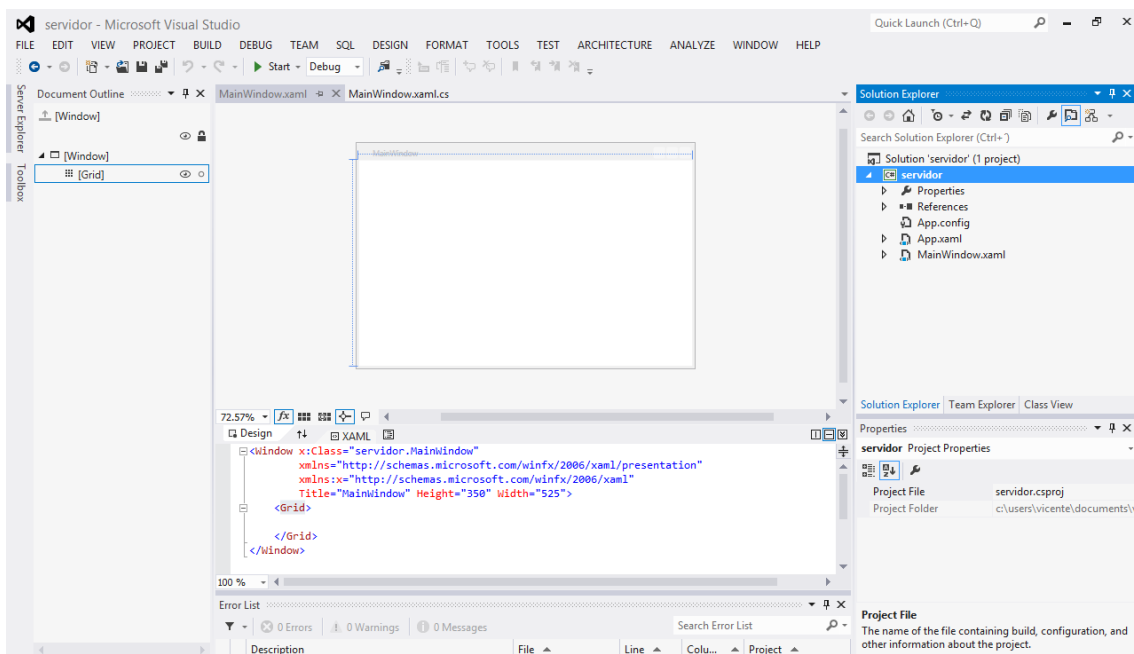


Figura N°3: Pantalla de inicio proyecto Wpf

3.- Dar lic derecho en el proyecto y se añadiran clases al proyecto(Figura N°4), la pantalla por defecto de la clase ServidorSockets creada(Figura N°5)

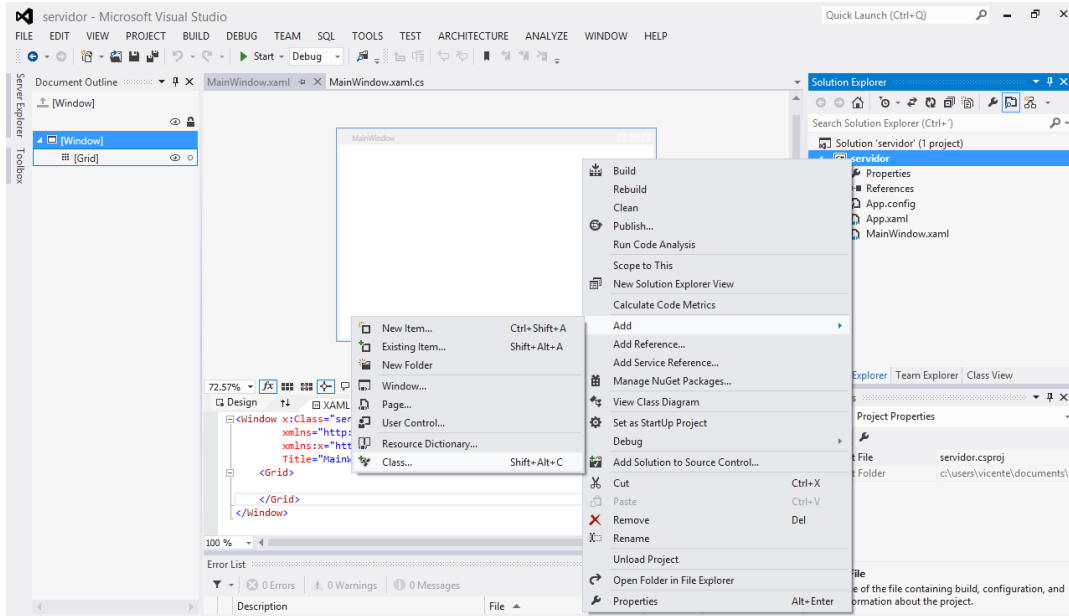


Figura N°4: Añadir clases al proyecto

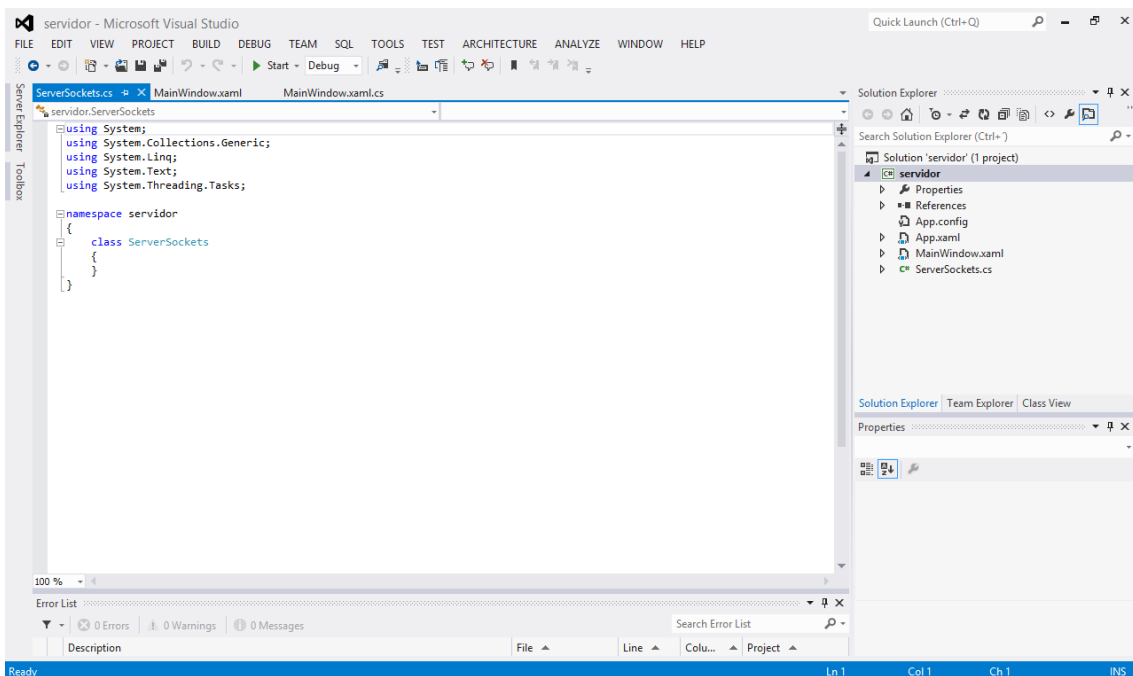


Figura N°5: Creacion del clase ServerSocket

La pantalla por defecto de la clase AdminClient creada se observa en la Figura N°6

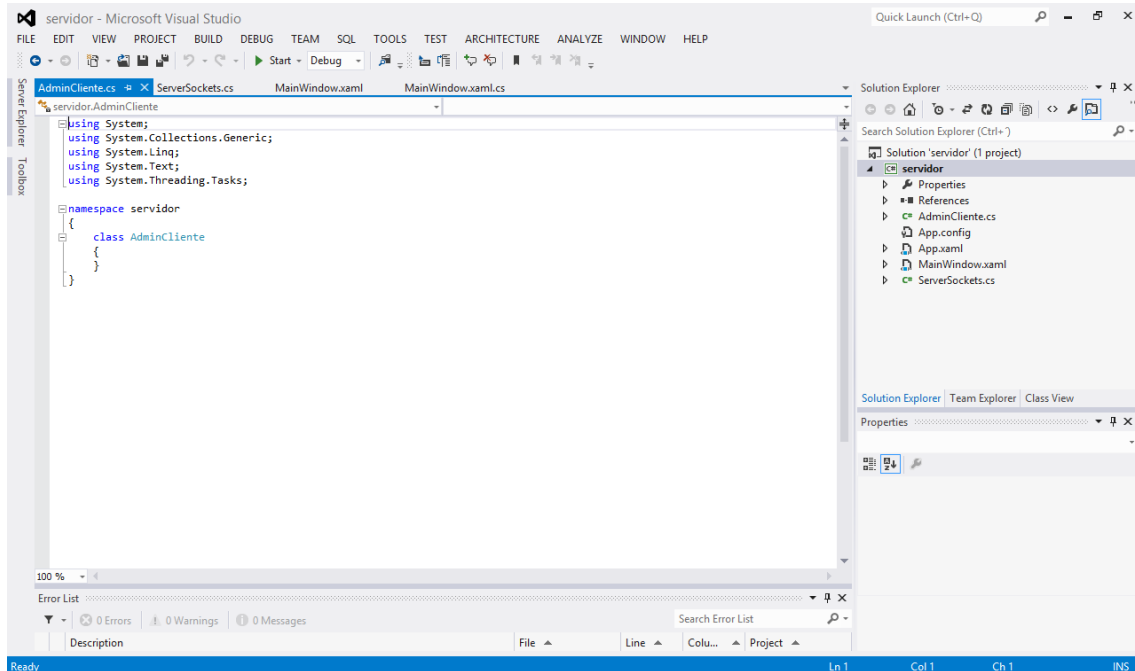


Figura N°6: Creacion del clase AdminCliente

4.- Insetar botones y texbloc en el interfaz grafica (7) arrastrando desde el menu a la interfaz creada en la ventana de trabajo.

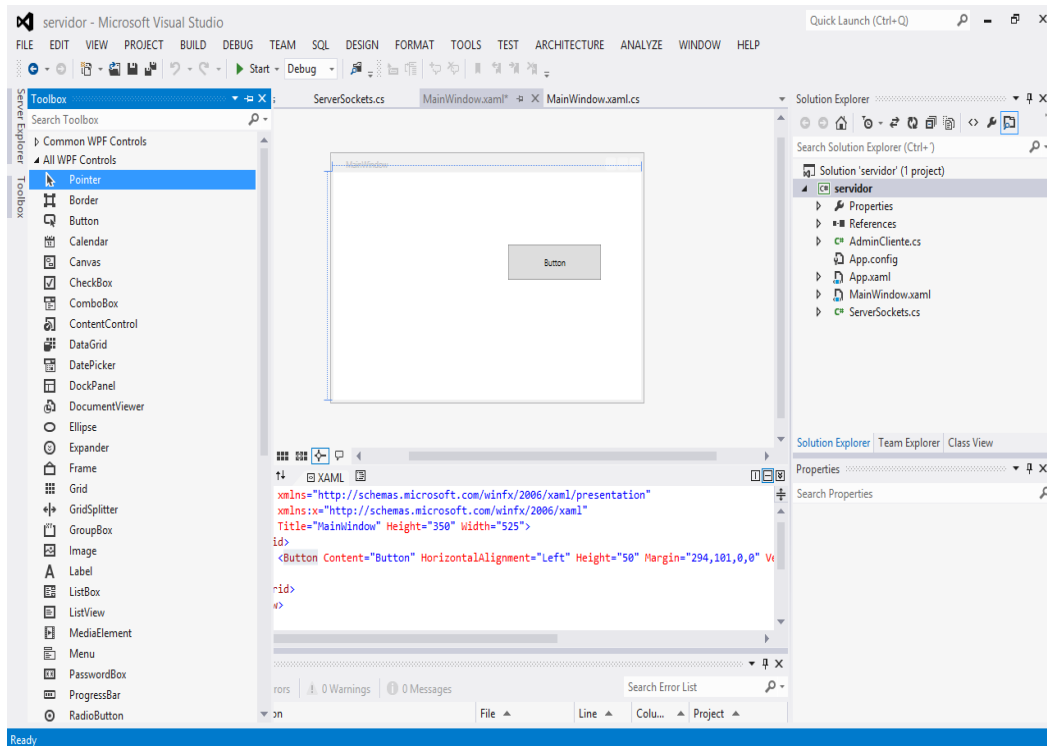


Figura N°7: Insertar botones y para salida de datos

5.- Ingresar la programación de clase ServerSocket (Figura N°8)

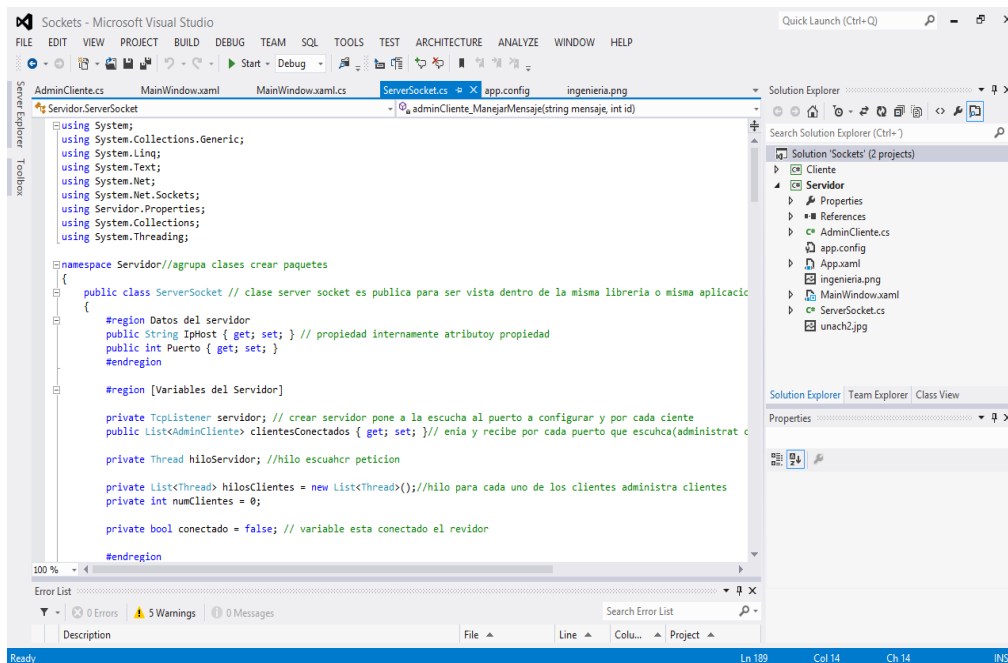




Figura N°8: Programación clase ServerSocket

- Librerías a utilizar en la case serve sockets:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Net;  
using System.Net.Sockets;  
using Servidor.Properties;  
using System.Collections;  
using System.Threading;  
Agrupa clases crear paquetes  
namespace Servidor/
```

- Clase server socket es publica para ser vista dentro de la misma librería o misma aplicación:

```
public class ServerSocket
```

- Se crean variables del Servidor

```
#region Datos del servidor
```

```
public String IpHost { get; set; }    public int Puerto { get; set; }  
#endregion
```

- La parte mas importante donde se crear el servidor que se pone a escuchar el puerto :

```
private TcpListener servidor;
```

- Administracion de clientes:

```
public List<AdminCliente> clientesConectados { get; set; });
```

- Evento notifica en interfaz ip (cada cliente) id nuemro de cliente(0-2):

```
public delegate void ManejadorMensaje(String mensaje, int id    public event  
ManejadorMensaje ManejarMensaje;  
public delegate void ManejadorIps(String ip, int id);  
public event ManejadorIps ManejarIps;  
ReaderWriterLockSlim toBlock = new ReaderWriterLockSlim
```

- Acepta las peticiones de los clientes, lee los mensajes que llegan:

```
TcpClient cliente = servidor.AcceptTcpClient();  
Byte[] bytesCliente = new Byte[256];  
NetworkStream streamCliente = cliente.GetStream();  
streamCliente.Read(bytesCliente, 0, bytesCliente.Length);
```

- Traducimos el stream de bytes a un string en codificación ASCII// por eso se puede ver el texto:



```
String mensajeCliente = Encoding.ASCII.GetString(bytesCliente, 0, bytesCliente.Length);  
mensajeCliente = mensajeCliente.Substring(0, mensajeCliente.IndexOf("\0"));
```

- manejar los mensajes le cliente los siguientes mensajes

```
AdminCliente adminCliente = new AdminCliente(cliente, numClientes);  
adminCliente.Nombre = mensajeCliente;  
adminCliente.ManejarMensaje += new  
AdminCliente.ManejadorMensaje(adminCliente_ManejarMensaje);  
adminCliente.Iniciar();
```

- método de administrar mensaje de cliente:

```
void adminCliente_ManejarMensaje(string mensaje, int id)  
toBlock.EnterReadLock();  
clientesConectados.RemoveAll(r => !r.Cliente.Connected);
```

6.- Programar clase AdminClient(Figura N°9)

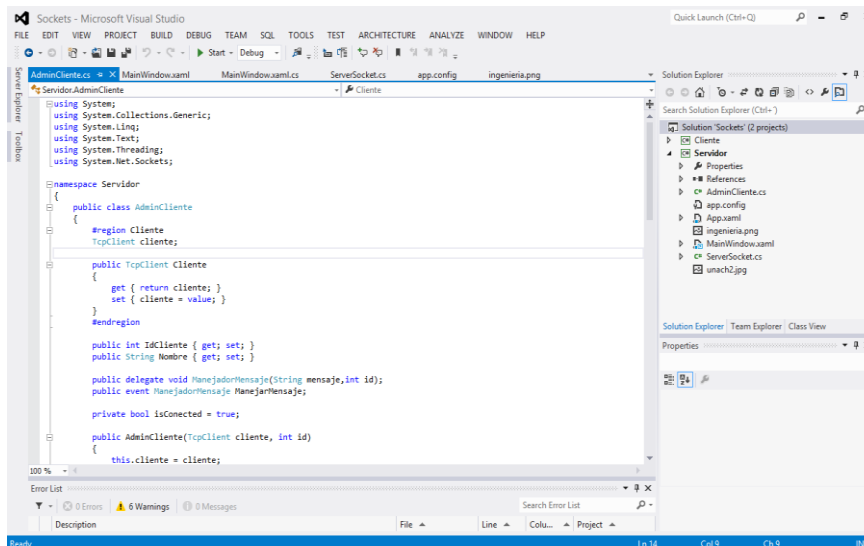


Figura N°9: Programación de clase AdminClient

- Se pone las librerías a utilizar para esta clase:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading;  
using System.Net.Sockets;
```

- agrupar clases para crear paquetes

```
namespace Servidor
```

- manejar mensajes

```
public delegate void ManejadorMensaje(String mensaje,int id);
```

`public event ManejadorMensaje ManejarMensaje;`

- administracion del cliente e id; id es el numero de cliente del 0 al 2
- `public AdminCliente(TcpClient cliente, int id)`

- Manejo de bytes

`Byte[] bytesCliente = new Byte[256]`

`NetworkStream streamCliente = cliente.GetStream();`

Manejo de hilos para los clientes(proceso)

`Thread hiloCliente = new Thread(AdminMensajes);`

`hiloCliente.Start();`

7.- Edición de la interfaz (Figura N° 10)

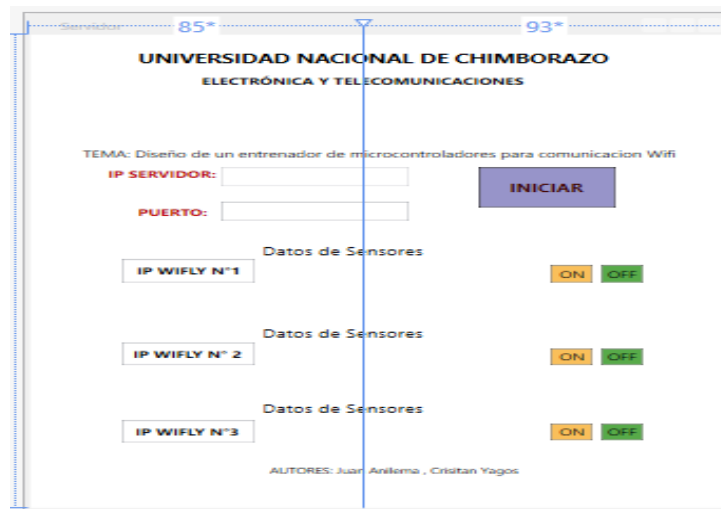


Figura N° 10: Interfaz gráfica del proyecto

- Procedencia del evento, parámetro al asociar el botón

`private void W1ON_Click_1(object sender, RoutedEventArgs e)`

```
{
    server.EnviarMensaje("a", 0);
}
```

`private void W1OFF_Click_1(object sender, RoutedEventArgs e)`

```
{
    server.EnviarMensaje("b", 0);
}
```

- Enviar mensaje de acuerdo a la lista de clientes conectado

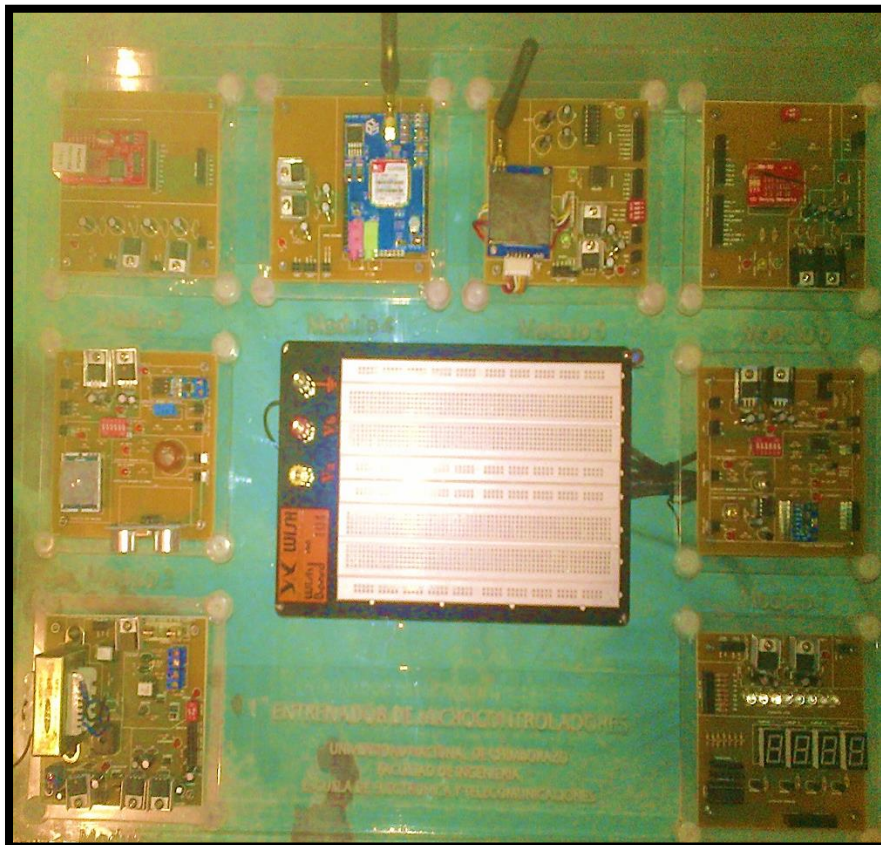
`server.EnviarMensaje("a", 0);`

8.- Para correr la aplicación presionar F12 y quedara esperando una respuesta del envío de datos del modulo wifly



ANEXO C

UNIVERSIDAD NACIONAL DE CHIMBORAZO FACULTAD DE INGENIERÍA ENTRENADOR DE MICROCONTROLADORES MANUAL DE USUARIO



INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES





INDICE

| | |
|--|-------|
| PORTADA..... | 122 |
| 1.MANUAL DE USUARIO..... | 1211. |
| 1.1.INTRODUCCIÓN..... | 121 |
| 1.2 HERRAMIENTAS..... | 121 |
| 2.MÓDULOS QUE CONFORMAN EL ENTRENADOR DE MICROCONTROLADORES | 123 |
| 2.1.MÓDULO RELE DE ESTADO SOLIDO – CRICE POR CERO..... | 123 |
| 2.1.1.PARTES RELE DE ESTADO SOLIDO..... | 124 |
| 2.1.2.PARTES CRUCE POR CERO..... | 125 |
| 2.2.MÓDULO DE LEDS Y DISPLYAS..... | 126 |
| 2.3.MÓDULO WIFLY..... | 128 |

1.- MANUAL DE USUARIO

1.1. - INTRODUCCIÓN

Para tener éxito en el manejo y funcionalidad en el desarrollo de las prácticas con el entrenador, usted debe leer con atención toda la información detallada en el presente documento.

El manual de usuario del entrenador de microcontroladores para comunicación se caracteriza por tener información detallada de las características, funcionamiento y partes importantes de cada una de las placas electrónicas.

El presente documento permitirá a los estudiantes y docentes tener pocos errores y mejor eficaz el desarrollo y aprendizaje de prácticas con diferentes microcontrolador (Microchip, Atmel) aplicado en las telecomunicaciones.

1.2.- HERRAMIENTAS.



Figura N° 1: Herramientas



Todos los estudiantes y docentes que utilicen el entrenador de microcontroladores deben tener las herramientas necesarias para el desarrollo de las prácticas, ver Figura N° 1

- **CABLES.-** Sirven para conectar dispositivos electrónicos.
- **MLTIMETRO.-** Instrumento que sirve para medir magnitudes eléctricas.
- **CAUTIN.-** Es una herramienta que se caracteriza por derretir el estaño y también permite soldar dispositivos electrónicos.
- **PELADORA.-** Sirve para pelar cables en el desarrollo de las prácticas.

2.- MÓDULOS QUE CONFORMAN EL ENTRENADOR DE MICROCONTROLADORES

2.1.- MÓDULO RELE DE ESTADO SOLIDO – CRUCE POR CERO

En la figura N° 2 se visualiza el diseño del módulo, desarrollado en el software Isis, Fuentes independientes de 5 y 3.3 voltios, Relé de Estado Sólido, Cruce por cero.

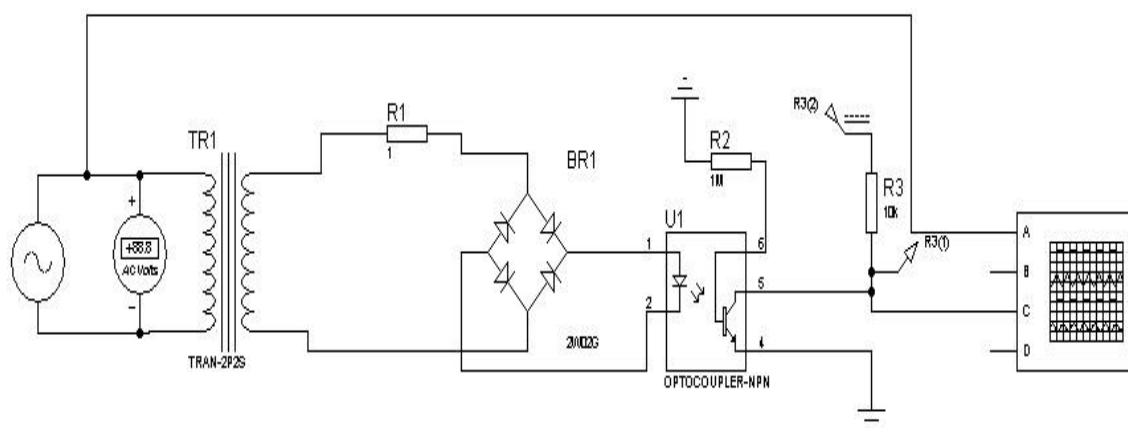
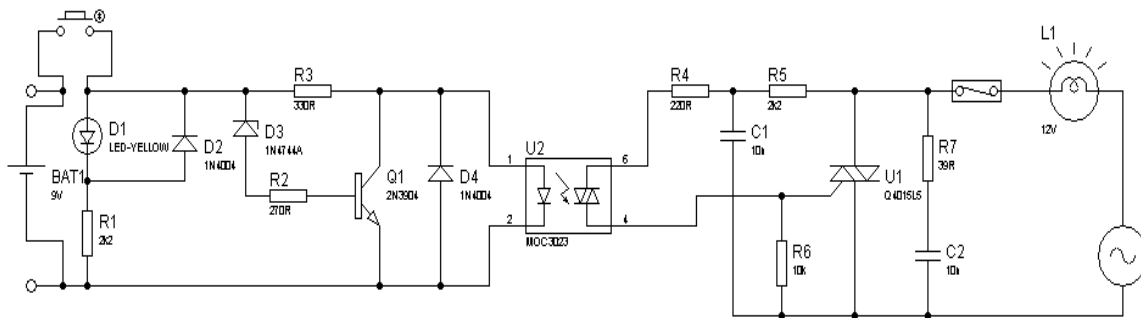


Figura N° 2: Modulo Relé de Estado Solido – Cruce por cero

En la Figura N° 3 se muestra los dos circuitos de Relé de estado Sólido y Cruce por cero con sus fuentes de voltaje independiente

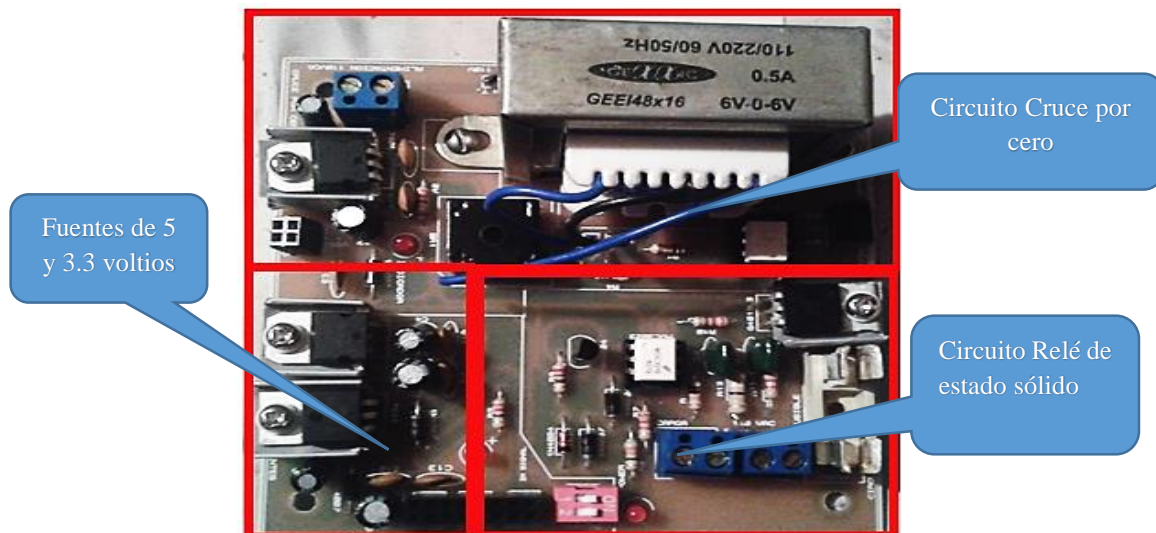


Figura N° 3: Partes importantes

2.1.1.- PARTES RELE DE ESTADO SOLIDO

En la Figura N° 4 se muestra las partes solo del circuito Relé de estado Sólido.

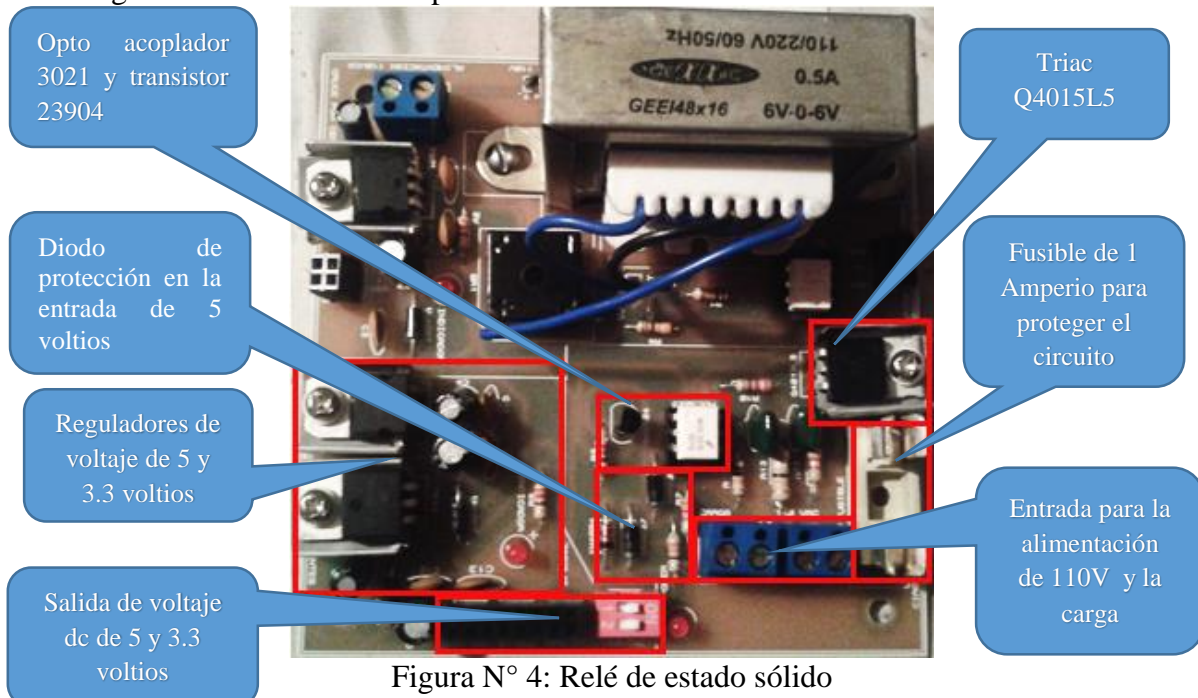


Figura N° 4: Relé de estado sólido

2.1.2.- PARTES CRUCE POR CERO

En la Figura N° 5 se muestra las partes del solo del circuito Relé de estado Sólido.

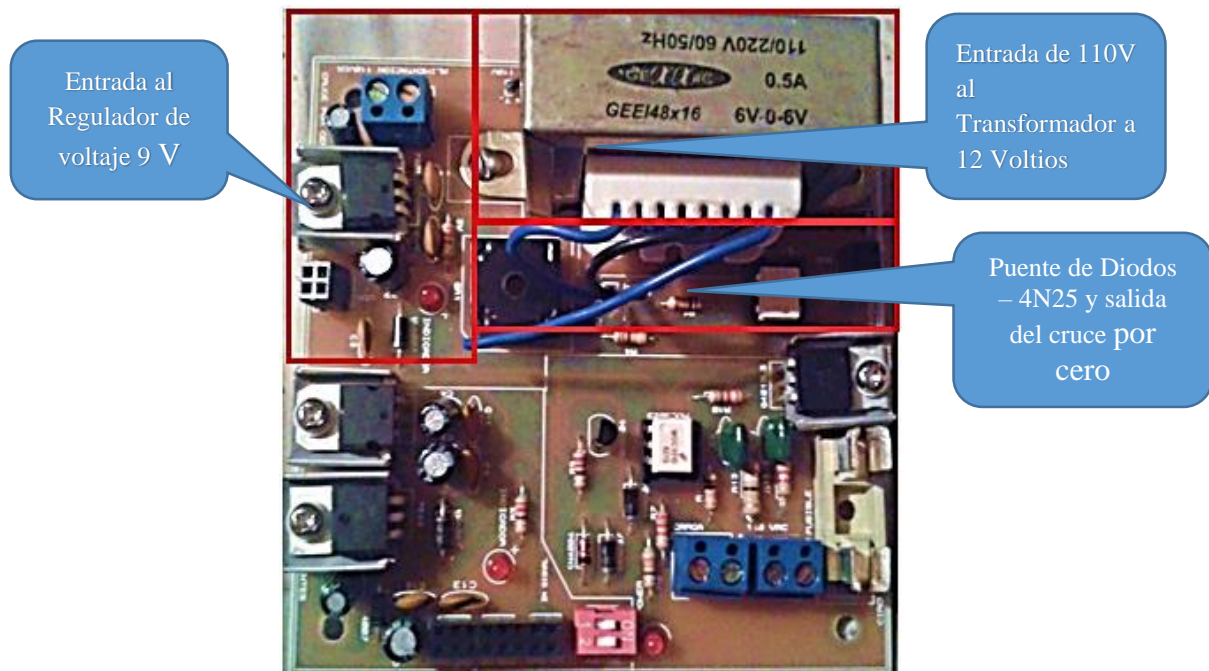


Figura N° 5: Cruce por cero

La placa está compuesta de dos circuitos un relé de estado sólido y un cruce por cero, tiene una entrada de alimentación de 9 voltios y las respectivas salidas de 5 y 3.3 voltios ver figura 3, en la figura 4 se explica la composición del circuito relé de estado sólido, la alimentación de corriente alterna, la carga, la activación con voltaje directo.

En la Figura 5 se muestra el circuito cruce por cero compuesto por la entrada de corriente alterna, transformador y rectificador (puente de diodos) y la parte del optoacoplador.

2.2.- MÓDULO DE LEDS Y DISPLAYS

En la Figura N° 6 se muestra el diseño del módulo de leds y display desarrollado en el software Isis.

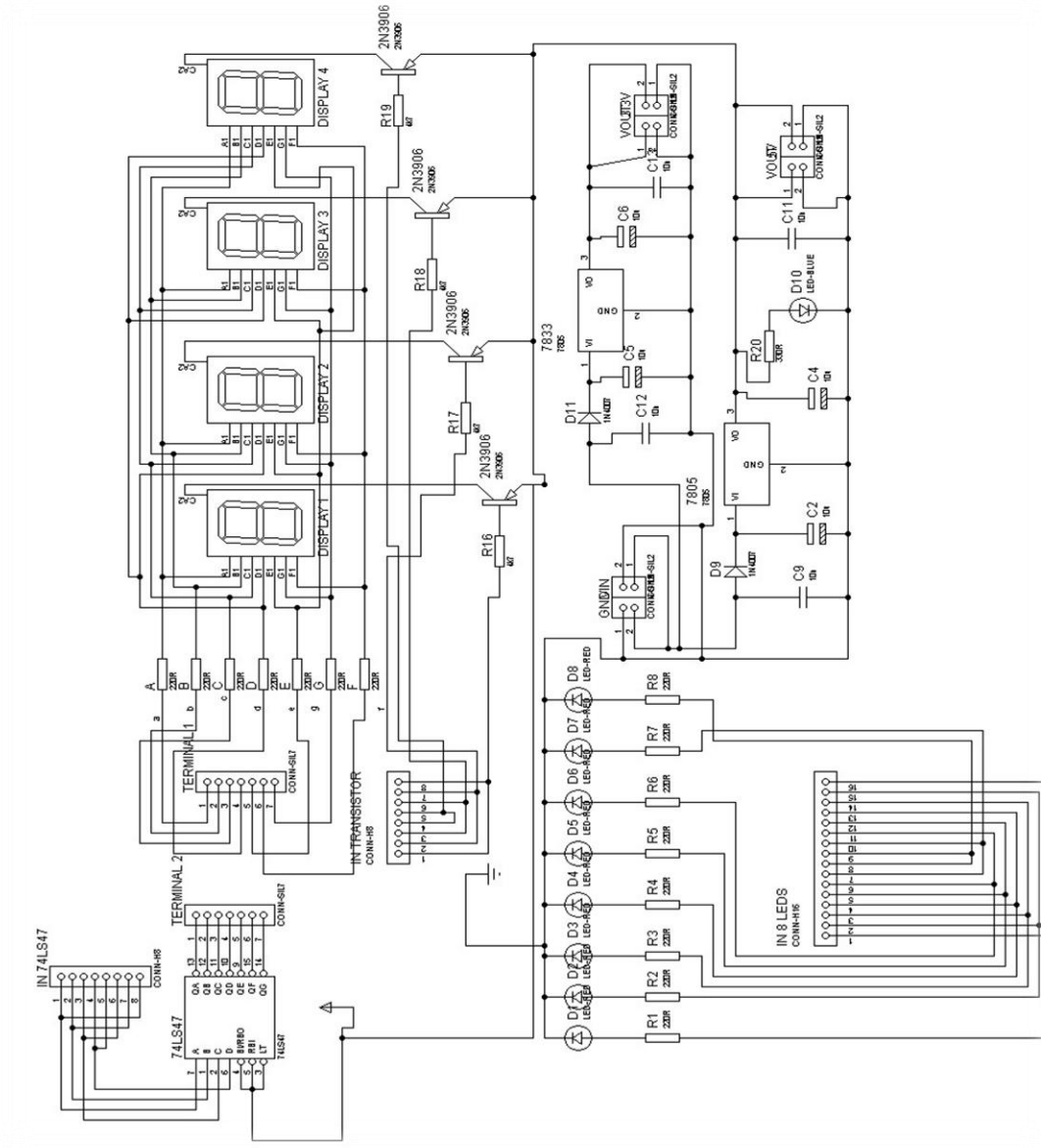


Figura N° 6: Modulo de Leds

En la Figura N° 7 se muestra las partes principales del módulo de leds y display.

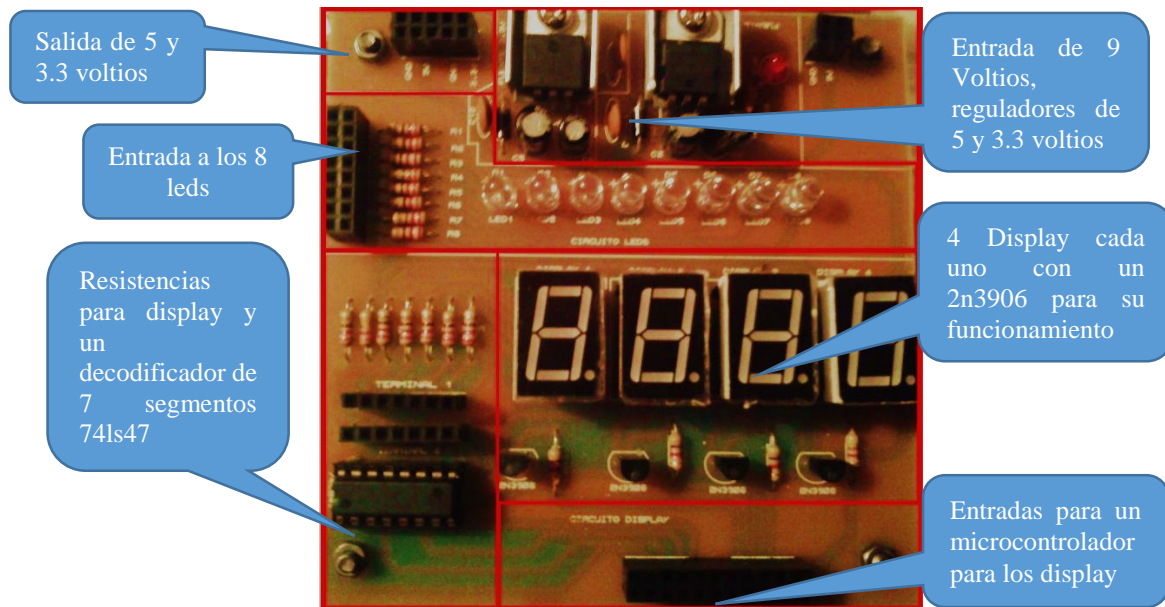


Figura N° 7: Modulo Display

En la figura N° 7 se muestra la composición de la placa leds y display con estrada de 9 voltios y sus respectivas salidas de 5 y 3.3 voltios de corriente directa, entradas de activación de leds, contiene resistencias para display y un decodificador de 7 segmentos, entradas para trabajar directamente con display o microcontrolador.

2.3.- MÓDULO WIFLY

En la Figura N° 8 se muestra el diseño del módulo Wifly desarrollado en el software Isis.

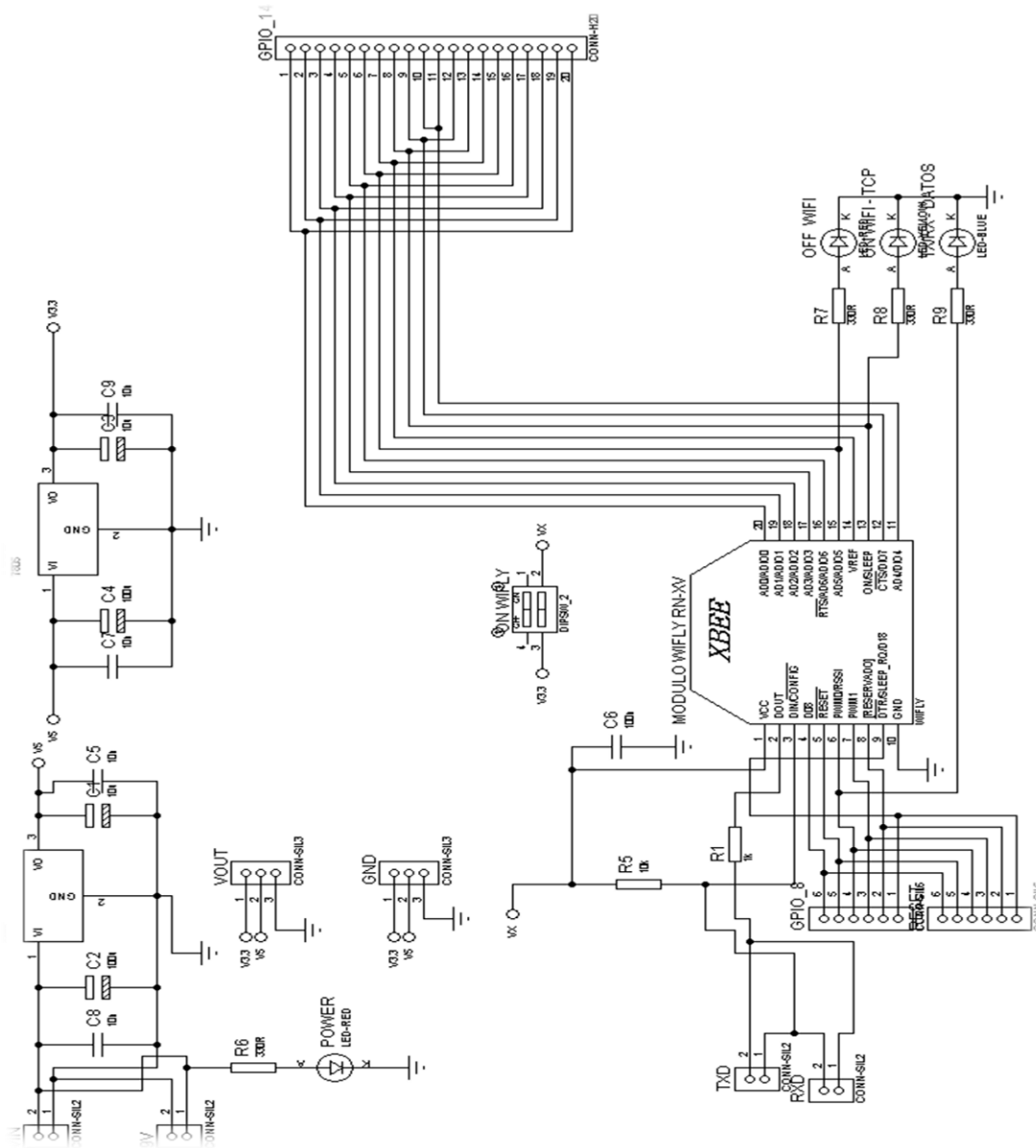


Figura N° 8: Módulo Wifly en Isis

En la Figura N° 9 se muestra las partes principales del módulo Wifly.

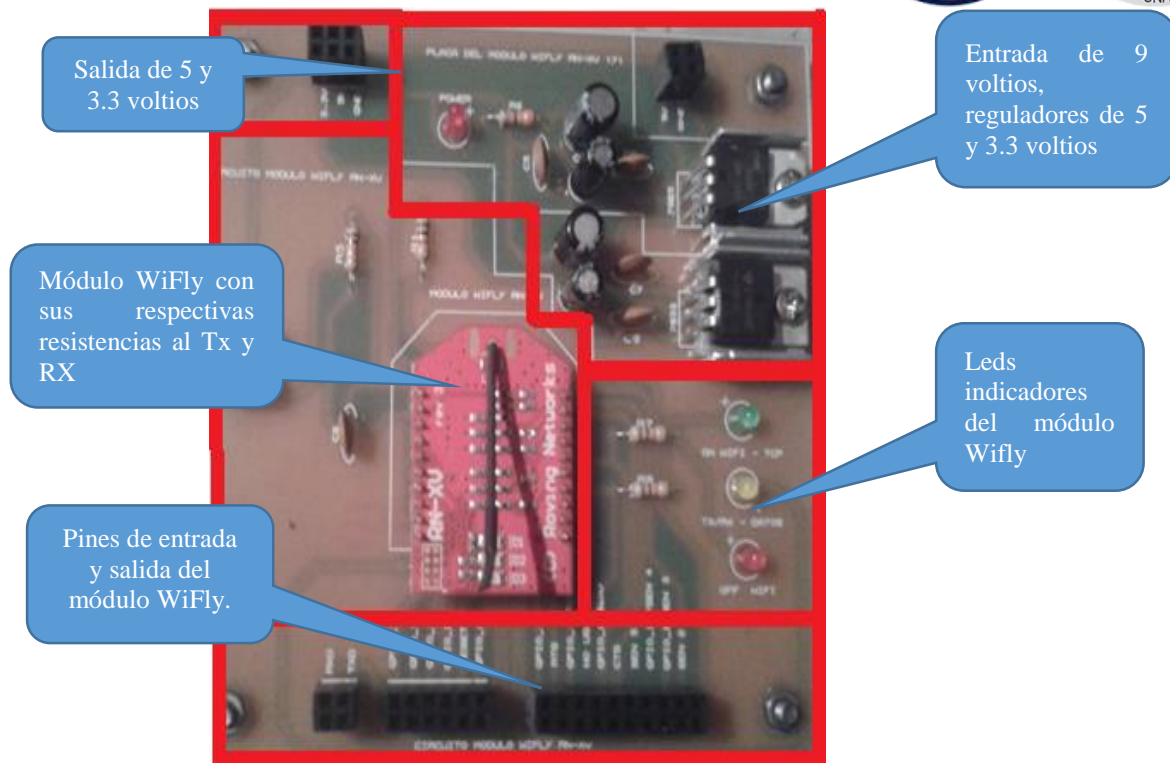


Figura N° 9: Modulo WiFly.

En la figura N° 9 se muestra la composición de la placa wifly con una entrada de 9 voltios y salidas de 5 y 3,3 voltios, contiene leds indicadores, pines de entrada y salida para el funcionamiento del módulo.



ANEXO D

DATASHEET

- ✓ **DATASHEET MODULO WIFLY RN-XV 171**
- ✓ **DATASHEET SENSOR LM35**
- ✓ **DATASHEET SENSOR MQ6**
- ✓ **DATASHEET SENSOR DTH11**