



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

Trabajo de grado previo a la obtención del Título de Ingeniero en Electrónica y
Telecomunicaciones

TRABAJO DE GRADUACION

**DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE
MICROCONTROLADORES PARA RED ETHERNET.**

Autores:

MARCO VINICIO PARRA CHÁVEZ

LUIS ALBERTO GAIBOR MELENDRES

Director:

Ing. Fabián Gunsha

RIOBAMBA – ECUADOR

2015

Los miembros del Tribunal de Graduación del proyecto de investigación de título:
DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE
MICROCONTROLADORES PARA RED ETHERNET.

Presentado por:

Marco Vinicio Parra Chávez.

Luis Alberto Gaibor Meléndres

Y dirigida por:

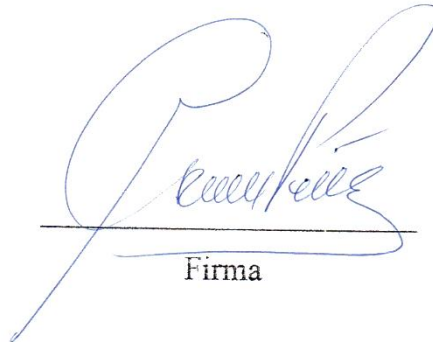
Fabián Celso Gunsha Maji.

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 para 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. Anibal Llanga.

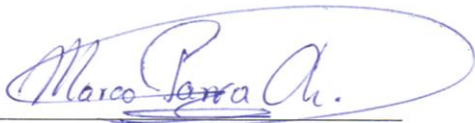
Miembro del Tribunal



Firma

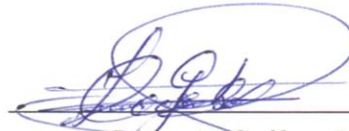
AUTORIA DE LA INVESTIGACION

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Parra Chávez Marco Vinicio, Gaibor Meléndres Luis Alberto e Ing. Fabián Gunsha; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”.



Marco V. Parra Ch.

060402659-1



Luis A. Gaibor M.

210056428-1

AGRADECIMIENTO I

Mi Gratitude a Dios Todopoderoso y a La Virgen Santísima por acompañarme siempre con sus bendiciones; por permitir que mis seres más amados se encuentren junto a mí.

A mi abuelita y a mi madre, por el apoyo incondicional que me supieron dar, gracias por los sabios consejos en los momentos más difíciles.

A mi hermano por alentarme constantemente.

A mi novia por su paciencia, apoyo, entrega y colaboración.

A mis compañeros de la Universidad, que por muchos años me han enseñado valores que me han ayudado a crecer y compartir con ellos momentos inolvidables.

A mi Director de Tesis, Fabián Gunsha, por compartir sus conocimientos.

A toda mi familia y a quienes compartieron conmigo los días de esfuerzo y sacrificio, mil gracias, sus nombres quedan grabados en mi corazón.

Gratitud a todos ello...

DEDICATORIA I

A Dios y a La Virgen Santísima

Por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente, por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante mi vida.

A mi abuelita Beticita y a mi madre Marthita.

Por ser el pilar fundamental en todo lo que soy, en mi educación, tanto académica como de la vida, por su apoyo incondicional; ya que gracias a sus esfuerzos, y su cariño, he podido salir adelante y culminar una etapa más en mi camino.

Este logro va dedicado a las personas que más van a disfrutar este triunfo, mi abuelita y mi madre, porque del mismo modo en que ellas compartieron mis alegrías, preocupaciones y desvelos, lo hago hoy yo, y esta victoria también son de ellas.

A mi hermano Jorge

A pesar de que tenemos nuestras discusiones y malos encuentros, has sido también una de las principales personas involucradas en ayudarme a que esta victoria fuera posible.

A mi tío Fausto

Por todo su apoyo incondicional, en los momentos de apremio.

A mi novia Anita Carola

Por ser el impulso durante mi carrera, que con su apoyo constante y su amor incondicional ha sido amiga y compañera inseparable, fuente de calma y sabiduría en todo momento.

Marco.

AGRADECIMIENTO II

Agradezco principalmente a Dios por amarme, iluminarme, guiarme y bendecirme en todo momento.

A mi madre y a mi esposa por su amor, comprensión apoyo y esfuerzo.

Al Ing. Fabián, por dirigirnos en la realización de este proyecto.

DEDICATORIA II

Dedico este trabajo a Dios, a mi esposa, padres y hermanos.

Luis

ÍNDICE GENERAL

ÍNDICE GENERAL	1
ÍNDICE DE TABLAS	7
ÍNDICE DE FIGURAS	8
RESUMEN	11
INTRODUCCION	13
CAPÍTULO I	15
1 FUNDAMENTACIÓN TEÓRICA	15
1.1 ENTRENADOR EN MICROCONTROLADORES	15
1.1.1 MÓDULOS ENTRENADORES COMERCIALES.....	16
1.1.1.1 ENTRENADOR PIC TRAINER PARA MICROS DE 18 PINES.	16
1.1.1.2 ENTRENADOR KIT EB88.	17
1.1.1.3 ENTRENADOR TARJETA DESARROLLO PIC COMPATIBLE PROGRAMADOR USB + LCD, PIC.....	18
1.1.1.4 ENTRENADOR MiE 32U.	18
1.1.1.5 ENTRENADOR MicroTRACK/ARM-C Complete Microcontroller Training Kit.....	19
1.1.1.6 ENTRENADOR dsPIC30F2010 DEVELOPMENT BOARD.....	20
1.1.1.7 ENTRENADOR ATmega TRAINING & DEVELOPMENT BOARD.	20
1.1.1.8 ENTRENADOR THE EasyPIC5 DEVELOPMENT TOOL.	21
1.2 LENGUAJE DE PROGRAMACIÓN	22
1.2.1 LENGUAJE DE PROGRAMACIÓN BASIC.	22
1.3 ETHERNET.....	22

1.3.1	HARDWARE PARA COMUNICACIÓN ETHERNET	23
1.3.1.1	NIC, O ADAPTADOR DE RED ETHERNET.....	23
1.3.1.2	REPETIDOR O REPEATER.....	23
1.3.1.3	ENRUTADOR O ROUTER.....	23
1.3.1.4	CONMUTADOR O SWITCH.....	24
1.3.2	ARQUITECTURA ETHERNET.....	24
1.3.3	DEFINICIÓN DE RED ETHERNET	24
1.3.4	DEFINICIÓN DE RED INDUSTRIAL ETHERNET	24
1.4	PROGRAMADOR USB.....	26
1.4.1	USBASP.....	26
1.5	MICROCONTROLADOR.....	26
1.5.1	MICROCONTROLADOR ATMEGA644P.....	27
1.6	SOFTWARE DE PROGRAMACIÓN BASCOM-AVR.....	28
1.7	SENSOR DE TEMPERATURA LM35.....	29
1.8	SENSOR DE HUMEDAD DHT11	30
1.9	OSILADOR DE CUARZO	31
1.10	REGULADOR DE VOLTAJE LM7805.....	32
1.11	MÓDULO ETHERNET	33
1.11.1	MODULO DE COMUNICACIÓN ETHERNET ENC28J60.....	34
1.11.2	MODULO ETH484	34
1.11.3	MODULO SHIELD ETHERNET.....	35
1.11.4	MODULO CONTROLADOR ETHERNET FUNDUINO.....	35
1.11.5	WIZNET W5100	39

1.12	PROTOCOLO DE COMUNICACIÓN SPI.....	40
1.13	SERVIDOR WEB	41
1.14	LENGUAJE HTML.....	41
CAPÍTULO II.....		43
2	METODOLOGÍA DE LA INVESTIGACIÓN.....	43
2.1	TIPO DE ESTUDIO.....	43
2.2	POBLACIÓN Y MUESTRA	43
2.2.1	POBLACIÓN	43
2.2.2	MUESTRA	43
2.2.3	HIPOTESIS	44
2.3	OPERACIONALIZACIÓN DE VARIABLES.....	44
2.4	PROCEDIMIENTOS	44
2.4.1	DISEÑO DE CIRCUITO ELECTRONICO Y PCB PARA EL MÓDULO ETHERNET.	45
2.4.1.1	DISEÑO DEL CIRCUITO ELECTRÓNICO.	45
2.4.1.2	DISEÑO DEL CIRCUITO PCB PARA EL MÓDULO ETHERNET.....	47
2.4.2	DISEÑO DE CIRCUITO ELECTRONICO Y PCB PARA EL MÓDULO DISPLAY's.	48
2.4.2.1	DISEÑO DEL CIRCUITO ELECTRÓNICO.	48
2.4.2.2	DISEÑO DEL CIRCUITO PCB PARA EL MÓDULO DISPLAY's.....	50
2.4.3	DISEÑO DE CIRCUITO ELECTRONICO Y PCB PARA EL MÓDULO LCD.....	50
2.4.3.1	DISEÑO DEL CIRCUITO ELECTRÓNICO.	50
2.4.3.2	DISEÑO DEL CIRCUITO PCB PARA EL MÓDULO DISPLAY's.....	52

2.4.4 CONFIGURACIÓN DE DISPOSITIVOS (MICROCONTROLADOR, FUNDUINO, COMPUTADOR).....	52
2.4.4.1 CONFIGURACIÓN DE RED.....	53
2.4.4.2 CONFIGURACIÓN DEL IDE BASCOM PARA LA COMUNICACIÓN SPI/IP.....	54
2.4.5 DISEÑO DEL SISTEMA DE CONTROL Y MONITOREO DE UN VIVERO MEDIANTE RED ETHERNET.....	56
2.4.5.1 DISEÑO CIRCUITO ELECTRONICO CONTROLADOR FUNDUINO.....	56
2.4.5.2 CIRCUITO ELECTRONICO-SENSORES.....	56
2.4.5.3 CIRCUITO DE ACTIVACIÓN DE LOS ACTUADORES.....	57
2.4.5.4 CIRCUITO DEL MICROCONTROLADOR-FUNDUINO.....	58
2.4.5.5 CIRCUITO GENERAL DEL CONTROL DEL SISTEMA DE UN VIVERO POR MEDIO DEL CONTROLADOR ETHERNET.....	59
2.4.5.6 PROGRAMACIÓN DEL MICROCONTROLADOR.....	59
2.4.5.6.1 PROGRAMACIÓN DEL CONTROL MANUAL DEL VENTILADOR.....	60
2.4.5.7 VISUALIZACIÓN DEL SERVIDOR WEB.....	62
2.5 PROCESAMIENTO Y ANÁLISIS.....	63
2.5.1 ESTUDIO DEL ENTRENADOR DE MICROCONTROLADORES PARA RED ETHERNET.....	63
2.5.2 ANÁLISIS DE LA COMUNICACIÓN DEL SISTEMA.....	64
2.5.3 ADMINISTRACIÓN DE DISPOSITIVOS.....	64
2.6 COMPROBACIÓN DE LA HIPÓTESIS.....	64
CAPÍTULO III.....	68
3 RESULTADOS.....	68
3.1 ANÁLISIS FINANCIERO.....	68

CAPÍTULO IV	70
4 DISCUSIÓN.....	70
CAPÍTULO V	71
5 CONCLUSIONES Y RECOMENDACIONES	71
5.1 CONCLUSIONES.....	71
5.2 RECOMENDACIONES.....	71
CAPÍTULO VI.....	72
6 PROPUESTA	72
6.1 TÍTULO DE LA PROPUESTA	72
6.2 INTRODUCCIÓN.....	72
6.3 OBJETIVOS	73
6.3.1 OBJETIVO GENERAL.....	73
6.3.2 OBJETIVO ESPECÍFICO.....	73
6.4 DESCRIPCIÓN DE LA PROPUESTA.....	73
6.5 DISEÑO ORGANIZACIONAL.....	74
6.6 MONITOREO Y EVALUACIÓN DE LA PROPUESTA.....	74
CAPÍTULO VII.....	75
7 BIBLIOGRAFÍA.....	75
CAPÍTULO VIII	77
8 ANEXOS.....	77
8.1 ANEXO 1. TABLA DISTRIBUCIÓN t STUDENT	77
8.2 ANEXO 1. VISUALIZACIÓN DEL SERVIDOR WEB.....	78
8.3 ANEXO 2. CONFIGURACIÓN DE FUSES PARA CARGAR EL PROGRAMA AL MICROCONTROLADOR.....	79

8.4	ANEXO 3. PROGRAMACIÓN.....	80
-----	----------------------------	----

ÍNDICE DE TABLAS

Tabla 1.-Sensores de Temperatura LM35	29
Tabla 2.- Reguladores LM78XX.....	33
Tabla 3.-Designación de los pines del controlador Ethernet.	37
Tabla 4.- Designación de los pines del controlador Ethernet.	38
Tabla 5.- Operacionalización de variables	44
Tabla 6.- Tiempo de desarrollo de la práctica	64
Tabla 7.- Diferencias entre los tiempos de desarrollo con y sin entrenador.....	66
Tabla 8.- Análisis Financiero.....	68

ÍNDICE DE FIGURAS

Figura 1.- Entrenador de microcontroladores.....	15
Figura 2.- Entrenador PIC TRAINER.	17
Figura 3.- Entrenador KIT EB88.....	17
Figura4.- Entrenador TARJETA DESARROLLO PIC COMPATIBLE PROGRAMADOR USB + LCD, PIC.....	18
Figura 5.- Entrenador MiE 32U.....	19
Figura 6.- Entrenador MicroTRACK/ARM-C Complete Microcontroller Training Kit.	19
Figura 7.- Entrenador MicroTRACK/ARM-C Complete Microcontroller Training Kit.	20
Figura 8.- Entrenador ATmega TRAINING & DEVELOPMENT BOARD.....	21
Figura 9.- Entrenador THE EayPIC5 DEVELOPMENT TOOL.	21
Figura 10.- Diagrama de flujo operación de Ethernet.	25
Figura 11.- Programador USB para AVR's	26
Figura 12.- Microcontrolador ATmega644P	27
Figura 13.- Distribución de pines del ATmega	28
Figura 14.- Sensor de temperatura LM35.....	29
Figura 15.- Sensor de humedad DHT11	30
Figura 16.- Osciladores de cuarzo	31
Figura 17.- LM7805 Regulador de voltaje 5VCC.....	32
Figura 18.- Modulo de comunicación Ethernet ENC28J60	34
Figura 19.- Modulo de comunicación Ethernet ETH484	34
Figura 20.- Módulo Shield Ethernet.....	35
Figura 21.- Módulo controlador Ethernet Funduino	36

Figura 22.- Pines del Controlador Ethernet	36
Figura 23.- LED's señalización	38
Figura 24.- Chip WIZnet Ethernet W5100.....	40
Figura 25.- SPI Master-Slave	40
Figura 26.-Polarización de capacitores en el regulador de voltaje	45
Figura 27.-Diseño Ethernet.....	46
Figura 28.- Diseño de pistas, placa controlador Ethernet Funduino	47
Figura 29.- Módulo controlador Ethernet Funduino 3D	47
Figura 30.-Triángulo de la ley de Ohm	48
Figura 31Módulo Display's.....	49
Figura 32.- Diseño de pistas, módulo Display's.....	49
Figura 33.-Módulo LCD.....	50
Figura 34.- Placa LCD's	51
Figura 35.- Diseño de pistas, placa controlador Ethernet Funduino	51
Figura 36.-Módulo LCD.....	52
Figura 37.- Topología de Conexión.....	52
Figura 38.- Red alámbrica de conexión modulo Ethernet - Computador.....	53
Figura 39.- Red alámbrica de conexión modulo Ethernet – Computador	53
Figura 40.- Conf. Microcontrolador en el IDE de BASCOM.	54
Figura 41.- Comandos de activación protocolo SPI.....	54
Figura 42.- Comandos de activación protocolo SPI y enlace de micro con ordenador....	55
Figura 43.- Entrada y salida del control del vivero.	56
Figura 44.- Circuito electrónico de los sensores.....	57

Figura 45.- Circuito activación-actuadores.	57
Figura 46.- Circuito electrónico para el control y monitoreo de un vivero.	59
Figura 47.- Flujograma de programación de encendido del ventilador.....	60
Figura 48.- Programación de encendido-apagado del ventilador.....	61
Figura 49.- Programación de encendido-apagado del calefactor.	61
Figura 50.- Programación de encendido-apagado de luz.	62
Figura 51.- Diagrama de procedimientos.	63
Figura 52.- Diagrama de procedimientos.	63
Figura 53.- Distribución de Probabilidad de la distribución t de Student	67
Figura 54.- Diagrama de procedimientos.	74

RESUMEN

El desarrollo de herramientas para el aprendizaje de cualquier rama del conocimiento ha sido motivo de continua evolución; en Electrónica, nuevos desarrollos y avances tecnológicos aparecen cada día, lo que amerita una inmediata adaptabilidad para estar en un nivel competitivo ante otras sociedades tecnológicas.

Un microcontrolador es un dispositivo programable que contiene las tres unidades eficaces de una computadora: CPU, Memoria y Periféricos de Entrada/Salida. De esta forma, el microcontrolador se convierte en un pequeño ordenador y es el desarrollador el encargado de programarlo según sus necesidades. Durante el desarrollo del proyecto investigativo del diseño e implementación de un entrenador de microcontroladores, se enfoca a la creación de módulos que añaden nuevas funcionalidades permitiendo el desarrollo de aplicaciones con microcontroladores de cualquier marca enfocados en el área de las telecomunicaciones, esencialmente en la comunicación por red Ethernet.

Culminado el ensamble del entrenador y su completa funcionalidad, se realizaron varias prácticas en la comunicación antes ya mencionada; ahora con el enfoque al desarrollo de un sistema de control de un vivero, permitiendo el control de cargas dentro un vivero, utilizando dispositivos electrónicos y sensores que contiene el mismo entrenador. El sistema consiste en un sistema de monitoreo manual o automático mediante el uso de Ethernet y un servidor web, en el cual se mostrará las lecturas de los sensores, permitiendo al usuario accionar de forma manual los actuadores o de forma automática.

Para la comprobación de la hipótesis: ¿El uso de entrenadores en microcontroladores reduce el tiempo de elaboración en las prácticas de laboratorio? se desarrolló un diseño experimental a estudiantes de la asignatura de microcontroladores. Cada estudiante desarrollo una práctica con el entrenador y sin el entrenador, con un 95 % de confianza se pudo probar que el tiempo promedio que tardan los estudiantes en elaborar una práctica con el uso de entrenadores es menor al tiempo promedio que tardan en elaborar la práctica sin éste, determinando así la eficiencia que tienen los entrenadores para el desarrollo de prácticas.



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERIA
CENTRO DE IDIOMAS



Lic. Ruth Molina

16 de Diciembre del 2015

SUMMARY

The development of tools for learning any branch of knowledge has been motif of continuous evolution; in Electronics, new developments and technological developments appear every day, which requires an immediate adaptability to be at a competitive level against other technological societies.

A microcontroller is a programmable device containing three efficient units of a computer: CPU, memory and peripherals Input / Output. Thus, the microcontroller becomes a small computer and the developer is responsible for program it according its necessities. During the development of the research project design and implementation of a coach microcontroller, it focuses on creating modules that add new functionality allowing the development of microcontroller applications of any brand focused on the area of telecommunications, essentially communication Ethernet network.

Completing the assembly of the coach and its functionality, several practices were conducted in the above-mentioned communication; now with the focus on developing of a control system of a nursery, allowing the load control into the nursery, using electronic devices and sensors that contain the same coach. The system consist of a system of manual or automatic monitoring using Ethernet and a web server, in which the sensor readings will be displayed, allowing to user to operate manually the actuators or automatically.

To test the hypothesis: The use of coaches in microcontrollers reduces the processing time in the labs? It developed an experimental design to students of the subject of microcontrollers. Each student developed a practice with the coach and without the coach, with 95% of confidence could prove that the average time that student stake to develop a practice with the use of coaches is less than the average time that they takes to develop practice without this, thus determining the efficiency with coaches for the development of practices.



INTRODUCCION

En el transcurso del tiempo, se ha manifestado un gran interés de los alumnos de ingeniería en electrónica y telecomunicaciones de utilizar elementos de control y automatización en el área de las Telecomunicaciones, las dificultades para el estudio y preparación de los estudiantes en éste campo es la carencia de módulos de ensayo para microcontroladores en los laboratorios, lo cual el estudiante debe desarrollar el mecanismo del dispositivo; debido a costos, construir el mecanismo con los mejores elementos del mercado no resulta tan factible para el estudiante, dando como producto un dispositivo con pocas características para la educación, y como consecuencia a una experiencia casi nula.

El mercado actual proporciona una gran variedad de módulos de entrenamiento para microcontroladores, teniendo estos un detalle muy particular, se ha podido descubrir que estos entrenadores están diseñados para funcionar con elementos de su misma línea de fábrica, es decir estos entrenadores no son de tecnología abierta, si se utiliza de una marca determinada se deberá utilizar solo elementos de esta línea sin dar lugar a la reutilización de los componentes con otras tecnologías en microcontroladores.

En el diseño e implementación propuesta para este trabajo de graduación; la carencia de entrenadores y el desarrollo de sistemas abiertos es sumamente dominada, ya que el diseño es capaz de adecuarse a cualquier tipo y marca de microcontrolador de cualquier industria, pudiendo este entrenador ser reutilizable con sus módulos correspondientes para el continuo aprendizaje; con el sistema abierto y la reutilización de los módulos permite tener como resultado un sistema embebido robusto.

Con el diseño e implementación de un entrenador de microcontroladores, se mejorará la enseñanza y el desarrollo de actividades en relación a las Telecomunicaciones utilizando electrónica en control.

Para la implementación del proyecto consta de varias fases, siendo la primera el diseño y la evaluación de los dispositivos si se adaptan a los requerimientos necesarios, además la

selección de sensores y módulos con más alcance tecnológico como es uno de ellos el módulo Ethernet Funduino.

El módulo Ethernet Funduino es utilizado para desarrollar una práctica real (simulación de un vivero el cual monitorea y controla distintas “cargas” utilizando una red LAN¹) con la cual permita constatar, el uso del entrenador en microcontroladores con el desarrollo de una situación netamente real.

Vale recalcar, que el módulo Ethernet Funduino es de tecnología “abierta”, lo que quiere decir, este módulo puede ser activado o controlado con cualquier tipo de microcontrolador, pero lo que se tiene que poner énfasis con este módulo Ethernet es la velocidad de transmisión y recepción de datos. Como sugerencia se debe escoger un microcontrolador que, para cada instrucción le tome un ciclo de reloj (es decir alrededor de 1.0 MIPS/MHz.).

Finalmente la creación de un servidor web, que tenga una interfaz gráfica amigable con el usuario, para su fácil utilización e interpretación.

Estas fases permitirán la creación del entrenador de microcontroladores el cual soportará el desarrollo de prácticas de sistemas embebidos robustos con la utilización de cualquier microcontrolador.

¹ Red de área local

CAPÍTULO I

1 FUNDAMENTACIÓN TEÓRICA

1.1 ENTRENADOR EN MICROCONTROLADORES

Un entrenador de microcontroladores es el conjunto de módulos, que a su vez están formados por elementos electrónicos, que conectados y con un diseño específico, forman una aplicación que sirve como complemento al microcontrolador para su ejecución. También es utilizado como material didáctico ya que reúne medios y recursos que facilitan a los estudiantes la adquisición de conceptos, habilidades y destrezas en cuanto a microcontroladores. (González Rojas & Méndez Caviativa, 2011, pág. 25)

En la **figura 1** se puede mostrar un ejemplo de entrenador en microcontroladores.

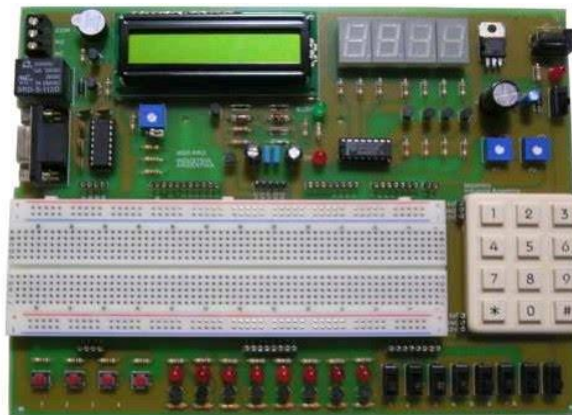


Figura 1.- Entrenador de microcontroladores.

Fuente: https://sites.google.com/site/alehering/2_1283430_3_2006227-8-22-1.jpg

Un entrenador en microcontroladores es robusto siempre y cuando cumpla con las siguientes características:

- **Modular:** con esta característica se asegura el usuario de poder realizar el montaje y desmontaje de los módulos, para ello el modulo debe poseer sus salidas y entradas disponibles para la configuración del microcontrolador.

- **Capacidad de soportar gamas de microcontroladores:** en el mercado no solo existe un solo proveedor de microcontroladores, por ello el entrenador debe soportar cualquier microcontrolador; caso contrario el estudiante se ve obligado a estar limitado a las bondades de un microcontrolador de una sola marca.

1.1.1 MÓDULOS ENTRENADORES COMERCIALES.

Como cualquier dispositivo electrónico, en el mercado existe una gran variedad de entrenadores de diversos fabricantes, que se destacan desde los más sencillos hasta los más complejos, a continuación algunos de ellos.

- Entrenador PIC TRAINER para micro de 18 pines.
- KIT EB88.
- Entrenador tarjeta Desarrollo PIC Compatible Programador USB + LCD, PIC.
- Entrenador MiE32U.
- Entrenador MicroTRAK / ARM-C Complete Microcontroller Training.
- Entrenador dsPIC30F2010 Development Board.
- Entrenador ATmega Training & Development Board.
- Entrenador The EayPIC5 development tool.

1.1.1.1 ENTRENADOR PIC TRAINER PARA MICROS DE 18 PINES.

El entrenador PIC trainer para micros de 18 pines como se muestra en la **figura 2**, es de configuración sencilla y está dirigido para personas que deseen aprender la utilización del microcontrolador 16F628A de Microchip.

El entrenador posee ICSP para la programación del dispositivo más no incluye el programador.



Figura 2.- Entrenador PIC TRAINER.

Fuente: <https://www.ucontrol.com.ar/forosmf/tienda-ucontrol/entrenador-pic-trainer-para-micros-de-18-pines/>

1.1.1.2 ENTRENADOR KIT EB88.

El entrenador EB88 como se muestra en la **figura 3** fue diseñado como KIT de aprendizaje por parte de la empresa MICROCHIP enfocado al microcontrolador 16F88. Sus principales componentes electrónicos de este entrenador son: 8 LED's², 4 switches, puertos serial asíncrono y síncrono, puertos de conversión A/D³, 13 puertos de salida o entrada, interfaz RS-232, sensor digital de temperatura DS18B20, puerto SPI⁴, teclado matricial, LCD⁵, el entrenador Kit EB88 de igual manera no posee programador, por lo que el PIC16F88 posee el Firmware dentro de su memoria. El Firmware o Bootloader es



Figura 3.- Entrenador KIT EB88.

Fuente: <http://www.puntofotante.net/EVOLUPICBootloader.htm>

² Light emisor diode (diodo emisor de luz)

³ Análogo/digital.

⁴ Interfaz Periférico Serial

⁵ Liquid Cristal Display

el programa que realiza la exportación e importación de los códigos hacia el microcontrolador o hacia el ordenador mediante el software de programación TINY.

1.1.1.3 ENTRENADOR TARJETA DESARROLLO PIC COMPATIBLE PROGRAMADOR USB + LCD, PIC.

Este entrenador tarjeta desarrollo pic compatible programador usb + lcd, se muestra en la **figura 4** el cual fue fabricado por MICROCHIP, puede ser utilizado para los PIC's de 28 y 40 pines de la familia 16XXX y 18XXX. La tarjeta tiene las siguientes características: receptor infrarrojo, pulsadores, driver para motor P.A.P.⁶, teclado matricial 4x4, display de siete segmentos ánodo común, buzzer, programador para PIC's de 28 y 40 pines.

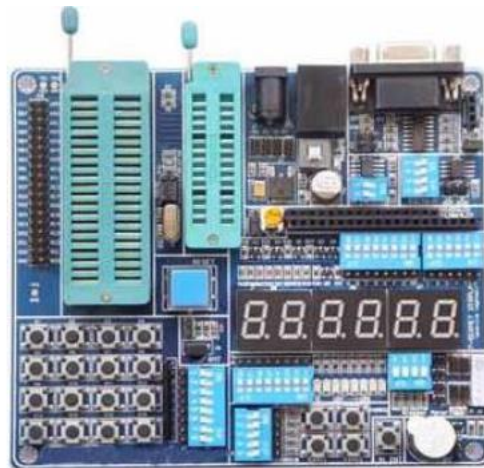


Figura 4.- Entrenador TARJETA DESARROLLO PIC COMPATIBLE PROGRAMADOR USB + LCD, PIC.
Fuente: http://articulo.mercadolibre.com.mx/MLM-64909860-tarjeta-desarrollo-pic-compatible-programador-usb-lcd-pic-_JM

1.1.1.4 ENTRENADOR MiE 32U.

El entrenador MiE 32U **figura 5**, fue hecha por MICROCHIP y enfocada al microcontrolador PIC32MX440F512H para agilizar la implementación de prototipos de control y comunicaciones, sus características son: Entradas y Salidas digitales, 512 KB de

⁶ Paso a paso

Flash tipo ROM⁷, 32 KB de SDRAM, entradas análogas, convertidor A/D de 10 bits y posee el conector ICSP⁸ para conexión con grabador/depurador PICKit3.



Figura 5.- Entrenador MiE 32U.

Fuente: <http://www.migsantiago.com/index.php/tutoriales/21-review-mie32u-de-microingenia>

1.1.1.5 ENTRENADOR MicroTRACK/ARM-C Complete Microcontroller Training Kit.

Entrenador creado por la compañía BiPOM **figura 6**, es el primer entrenador en salir al mercado con ejemplos en lenguaje assembler, basic y C, sus componentes principales son: Teclado matricial de 16 caracteres, bus I²C, driver para motor DC⁹, conversión A/D, comunicación serial, comunicación puerto paralelo, LCD 24x2, dip-switch, etc.



Figura 6.- Entrenador MicroTRACK/ARM-C Complete Microcontroller Training Kit.

Fuente: <http://www.robotshop.com/ca/bipom-microtrack-arm-c-complete-microcontroller-training-kit-1.html>

⁷ Read only memory.

⁸ In Circuit Serial Programming.

⁹ Corriente continua.

1.1.1.6 ENTRENADOR dsPIC30F2010 DEVELOPMENT BOARD.

El entrenador dsPIC30F2010 es una tarjeta desarrolladora basada en el PIC30F2010 **figura 7**, se caracteriza por: primer entrenador que incluye un protoboard de desarrollo, buzzer y parlante, programador externo, pulsadores, LED's columna de potenciómetros para entradas analógicas, LED's, sensores de humedad.

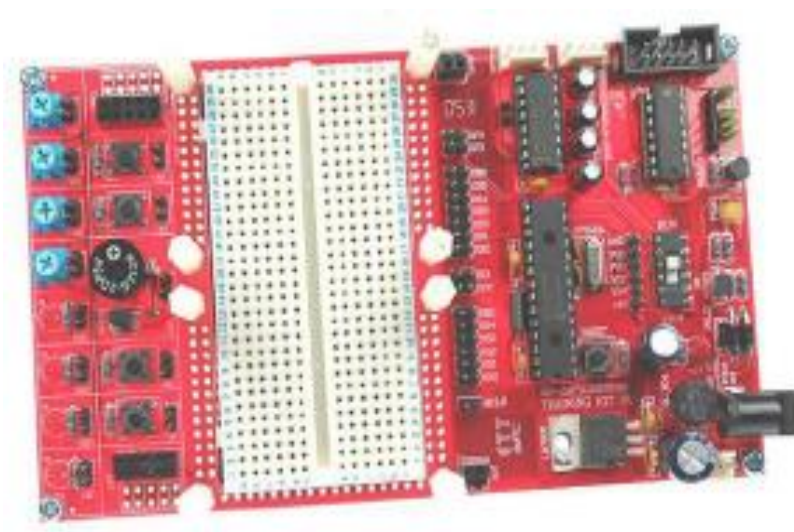


Figura 7.- Entrenador MicroTRACK/ARM-C Complete Microcontroller Training Kit.

Fuente: http://www.futurlec.com/Pictures/dsPICBRD1_600.jpg

1.1.1.7 ENTRENADOR ATmega TRAINING & DEVELOPMENT BOARD.

El entrenador ATmega Training **figura 8**, fue desarrollado por la firma Atmel AVR microcontrollers, su funcionamiento está enfocado al microcontrolador ATmega128 el cual se encuentra en una tarjeta removible, su principal característica. Otras de sus características son: comunicación ISP¹⁰, comunicación serial puerto RS-232, puerto RS-485, oscilador, pulsadores, displays de siete segmentos, conversores análogos / digitales, buzzer, LCD's.

¹⁰ Serial Peripheral Interface.



Figura 8.- Entrenador ATmega TRAINING & DEVELOPMENT BOARD.

Fuente: http://microcontrollershop.com/product_info.php?products_id=3404

1.1.1.8 ENTRENADOR THE EasyPIC5 DEVELOPMENT TOOL.

Este entrenador diseñado por MICROCHIP con el aval de mikroElektronika **figura 9**, diseñaron el entrenador con el propósito de mejorar a los entrenadores anteriores, ya que posee los siguientes componentes: GLCD¹¹ touch screen, matriz de LED's 4x8, comunicación serial por puerto RS-232, comunicación mediante cable USB¹² tipo B, conjunto de dipswitch para entradas analógicas.

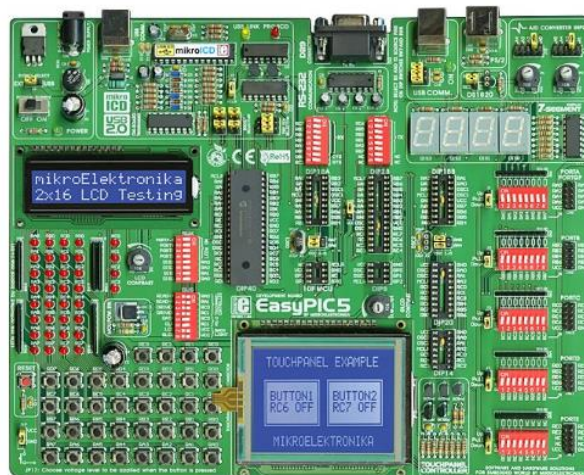


Figura 9.- Entrenador THE EasyPIC5 DEVELOPMENT TOOL.

Fuente: <http://www.mikroe.com/easypic/v5/>

¹¹ Pantalla de Cristal Líquida Gráfica

¹² Universal serial bus.

1.2 LENGUAJE DE PROGRAMACIÓN

Los lenguajes de programación son la herramienta que permite establecer una correcta comunicación entre la computadora y el usuario. Sin lenguaje de programación no sería posible expresar las instrucciones que se desea ejecutar, ni tampoco se podría mostrar los resultados de dichas instrucciones. Los lenguajes de programación más conocidos son: Ada, Asp, BASIC, C, C++, C#, Clipper, COBOL, Delphi, Eiffel, Ensamblador, Fénix, FORTRAN, Java, Lexico, Lingo, Lisp, Logo, Lua, MAGIC, Modula, Natural, Oberon, Object, Pascal, Perl, PHP, Sh, Simula, Smalltalk, TI-BASIC, TCL, VBA, VisualBasic, VisualDialogScript, VisualFoxpro, Yurix, ZPL, etc. (N/A, 2006)

1.2.1 LENGUAJE DE PROGRAMACIÓN BASIC.

El BASIC es un lenguaje de programación muy utilizado a nivel mundial; su nombre proviene de la expresión inglesa Beginner's All-purpose Symbolic Instruction Code (BASIC), que significa en español código de instrucciones simbólicas de propósito general para principiantes.

1.3 ETHERNET

Ethernet es un protocolo de nivel de enlace para redes de área local (LAN) basado en datagramas y definido en el estándar IEEE 802.326. El protocolo original se basa en una topología de bus con medio compartido, es decir, varias estaciones que pueden enviar datos al mismo tiempo. Por tanto se debe arbitrar un mecanismo de control de acceso al medio y resolver los problemas que conllevan los accesos simultáneos (colisiones). (Tanenbaum, 2003)

Por ello las redes Ethernet utilizan la técnica de comunicación llamada CSMA/CD¹³ Acceso múltiple con detección de portadora con detección de colisiones, previniendo las colisiones y la pérdida de información.

¹³Carrier Sense Multiple Acces with Collision Detection

CSMA/CD tiene las siguientes características:

Acceso múltiple: significa que las computadoras y/u ordenadores tienen el mismo derecho, o posibilidad de transmitir sus paquetes de datos.

Detección de portadora: significa que si bien todas las computadoras tienen el mismo derecho de enviar sus datos, previamente deberán “escuchar un momento, hasta que haya silencio” (es decir, hasta que no haya ninguna máquina transmitiendo); a partir de allí, podrá empezar a transmitir la que primero lo intente.

Detección de colisión: luego de que una computadora empezó a transmitir sus paquete de datos (tramas) en la red, deberá controlar que en ese instante no haya otra que también esté realizando una transmisión, de lo contrario, se produce una colisión de paquetes. (Rosales, 2013)

1.3.1 HARDWARE PARA COMUNICACIÓN ETHERNET

1.3.1.1 NIC, O ADAPTADOR DE RED ETHERNET.

La tarjeta adaptadora de red Ethernet o también mal llamada NIC¹⁴, permite al ordenador o computador conectarse a una red, además la NIC tiene consigo una dirección MAC¹⁵ única que le identifica al ordenador dentro de la red.

1.3.1.2 REPETIDOR O REPEATER.

El repetidor es un dispositivo utilizado para aumentar el alcance del enlace para la comunicación, disminuyendo la degradación de la señal eléctrica en el medio físico.

1.3.1.3 ENRUTADOR O ROUTER.

El funcionamiento del Router es gracias a las tablas y algoritmos de enrutamiento que posee, los cuales le permiten escoger al router el mejor “camino” para que el paquete llegue a una determinada dirección de destino.

¹⁴ Network Information Center

¹⁵ Media Acces Control

1.3.1.4 CONMUTADOR O SWITCH.

El switch permite la interconexión de múltiples segmentos de red, es más sofisticado y rápido que el *bridge* (ya no es utilizado en la actualidad).

1.3.2 ARQUITECTURA ETHERNET

La arquitectura de tipo Ethernet es el estándar (una norma técnica) para conectar un computador u ordenador a una red (por lo general internet no Internet), es por eso que una arquitectura es fundamental en la organización de un sistema, las relaciones entre sus componentes y los principios que rigen su diseño y evolución. Una arquitectura ofrece a sus clientes un marco para la optimización de sus recursos técnicos en apoyo de su actividad empresarial; este tipo de referencia se enmarca dentro de los siguientes lineamientos y necesidades:

- Contener un conjunto de principios que permitan orientar el diseño e implementación de la red, estableciendo relaciones entre los componentes de la red y el control de la misma.
 - Determinismo y necesidades en tiempo real.
 - Consideraciones de seguridad.
 - Aumento significativo de los protocolos de automatización basados en Ethernet.
- (Remache Bevides, 2009, pág. 09)

1.3.3 DEFINICIÓN DE RED ETHERNET

Red Ethernet es un sistema para transmitir información entre dos o más dispositivos empleando un medio compartido, especifica el medio utilizado, la señal que se utiliza y la forma que la información debe tener. (Sánchez Lobón, 2007, pág. 4)

1.3.4 DEFINICIÓN DE RED INDUSTRIAL ETHERNET

Una red industrial Ethernet tiene la capacidad de supervisar y monitorear las transmisiones de datos, la existencia de servicios como Telnet, que se utilizan para el

equipo de configuración y/o reconfiguración, y la capacidad de cargar programas de dispositivos de control desde una locación central. (Romero, 2005, pág. 03)

En la **figura 10** se muestra de forma general como un ordenador y/o computador debe seguir las órdenes para el envío de un mensaje (datos, trama) con arquitectura Ethernet.

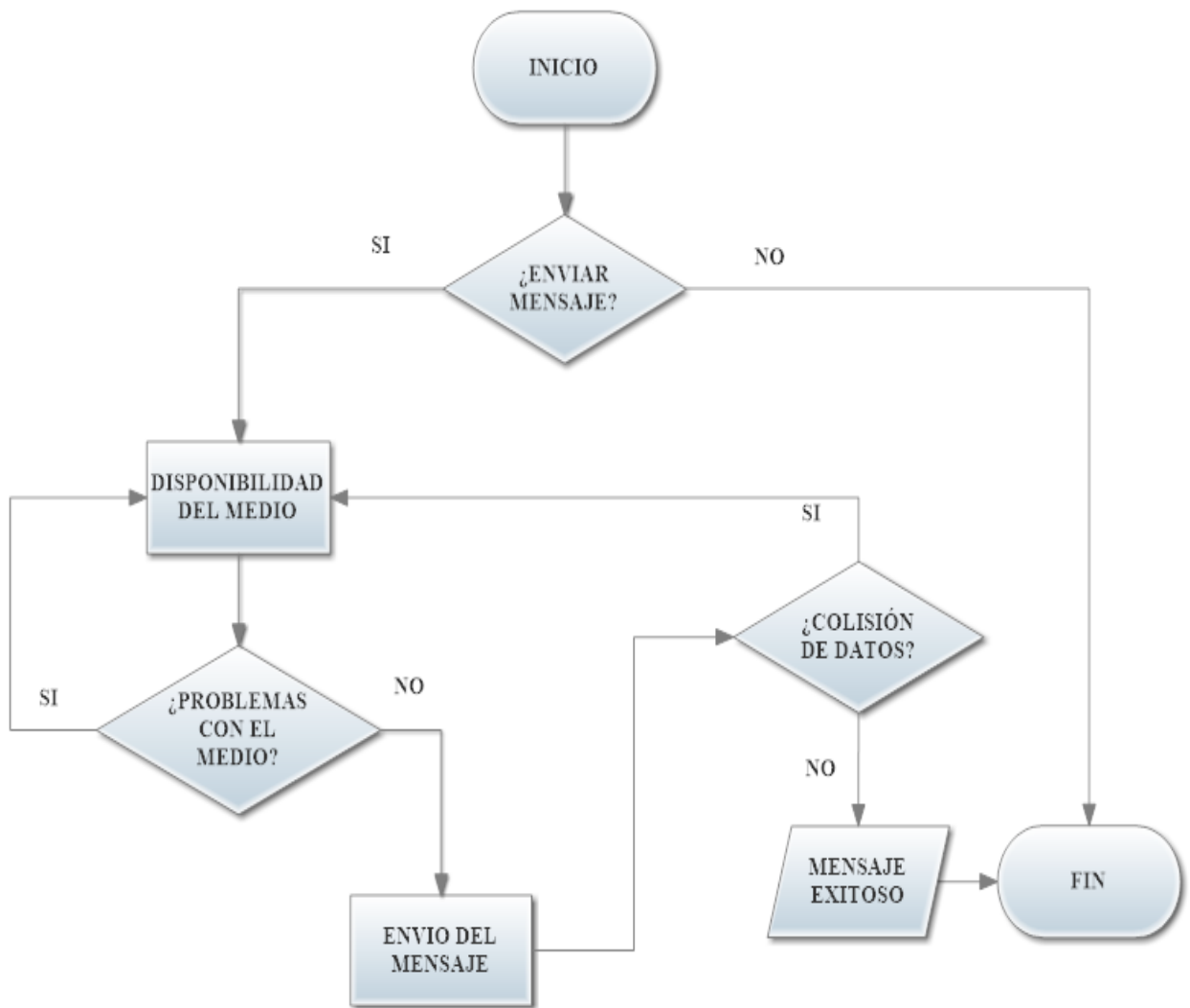


Figura 10.- Diagrama de flujo operación de Ethernet.

Fuente: Los Autores

1.4 PROGRAMADOR USB.

1.4.1 USBASP.

USBasp es un programador específicamente para microcontroladores AVR ATMEL, el programador utiliza un controlador USB-firmware, el cual le permite al usuario cargar el programa al microcontrolador, caso contrario si no lo posee este controlador instalado en su ordenador tendrá dificultades al momento de cargar el programa al microcontrolador.

Figura 11.

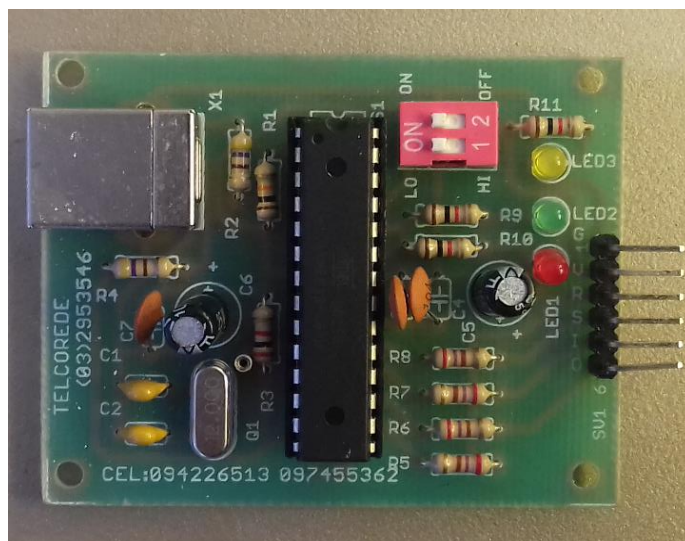


Figura 11.- Programador USB para AVR's

Fuente: Los Autores

1.5 MICROCONTROLADOR

Un microcontrolador, es un circuito integrado de alta escala de integración posee la arquitectura de un computador, esto es CPU¹⁶, memorias RAM¹⁷, EEPROM¹⁸, y circuitos de E/S. (Reyes, 2006)

¹⁶ Unidad de procesamiento central.

¹⁷ memoria de acceso aleatorio

¹⁸ ROM programable borrable

1.5.1 MICROCONTROLADOR ATMEGA644P

El ATmega644P **figura 12** es un microcontrolador de tipo CMOS¹⁹ de baja potencia de 8 bits basado en la arquitectura RISC²⁰. En un solo ciclo de reloj, el ATmega644P logra rendimientos de 1 MIPS²¹ por MHz, permitiendo al sistema la optimización del consumo de energía en comparación con la velocidad de procesamiento. (Atmel Corporation, 2013)



Figura 12.- Microcontrolador ATmega644P

Fuente: Autores

Las características (**figura 13**) que le hacen ser al microcontrolador ATmega644P robusto es por poseer, 64 Kbyte en memoria Flash, 2 Kbyte en memoria EEPROM y 4 Kbyte en memoria RAM, y lo más importante, es capaz de realizar sentencias y órdenes establecidas en su memoria. Por tal motivo el ATmega644P es el encargado de ejecutar las funciones de los elementos controlados como son los sensores de temperatura, humedad y lúmenes, su capacidad de procesamiento y memoria, son los indicados para la administración y ejecución de los datos, con el cumplimiento de las funciones.

¹⁹ Semiconductor complementario de óxido metálico.

²⁰ Reduced Instruction Set Computer.

²¹ Millones de instrucciones por segundo

LEGEND	
GND	
POWER	
CONTROL	
PORT PIN	
ATMEGA PIN FUNC	
DIGITAL PIN	
ANALOG-RELATED PIN	
PWM PIN	
SERIAL PIN	
ARDUINO PIN	

Using Arduino as ICSP Programmer for ATMEGA644P

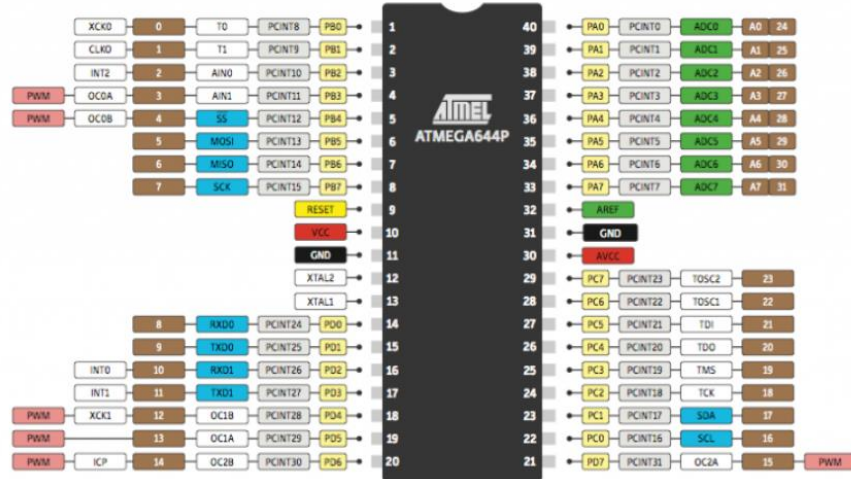
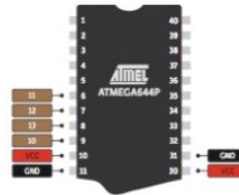


Figura 13.- Distribución de pines del ATmega

Fuente: www.pighixxx.com

1.6 SOFTWARE DE PROGRAMACIÓN BASCOM-AVR.

BASCOM-AVR, es una plataforma de programación exclusiva para microcontroladores ATMEL, BASCOM-AVR utiliza el lenguaje de programación BASIC, permitiendo al usuario desarrollar el prototipo a base de instrucciones, de acorde a la necesidad del diseño electrónico, permitiendo así la comunicación entre el microcontrolador y el usuario.

Los módulos son los dispositivos desarrollados con una función propia para poder ser agrupados de distintas maneras con otros elementos para constituir una unidad mayor comúnmente llamados modulares.

1.7 SENSOR DE TEMPERATURA LM35

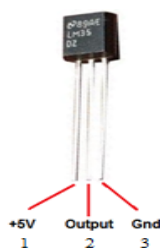


Figura 14.- Sensor de temperatura LM35

Fuente: Autores

Los sensores LM35 son circuitos integrados utilizados para medir temperatura; tiene la salida de voltaje linealmente proporcional a la temperatura en grados Celsius. El dispositivo LM35 no requiere de ninguna calibración externa y puede medir desde los -55° hasta los 150° dependiendo el tipo de encapsulado. (TEXAS INSTRUMENTS, 2015, pág. 01)

El LM35 es un elemento transductor (de temperatura a señal eléctrica) y de precisión, su linealidad se refleja con la igualdad:

$$1 \text{ mV} = 1^{\circ}\text{C}$$

En el mercado existe variedad de sensores que empiezan con la serie LM35 en el encapsulado por lo que se debe tomar en cuenta la siguiente tabla:

Tabla 1.-Sensores de Temperatura LM35

Fuente: Los Autores

	ENCAPSULADO	MIN.	MAX.	UNIDAD
Especificaciones en temperatura: $T_{\text{MIN}} - T_{\text{MAX}}$	LM35, LM35A	-55	150	°C
	LM35C, LM35CA	-40	110	
	LM35D	0	100	
VOLTAJE APLICADO		4	30	V

El sensor LM35 (**figura 14**) posee tres pines que permiten su configuración:

- Pin 1 Pin de entrada de voltaje positivo desde 4 VCD hasta 30 VCD.
- Pin 2 Pin de salida.
- Pin 3 Pin de conexión a un chasis o GND (ground).

1.8 SENSOR DE HUMEDAD DHT11

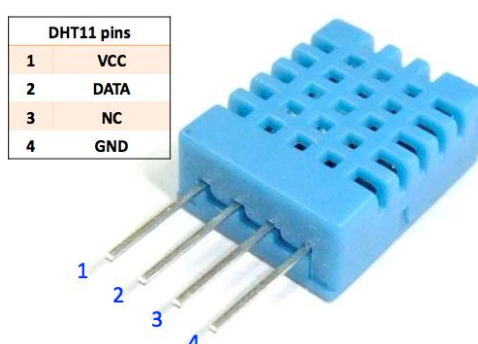


Figura 15.- Sensor de humedad DHT11
Fuente: Autores

El sensor DHT11 **figura 15** permite lecturas de humedad en valores analógicos, su frecuencia de “actualización” es de 1 Hz lo que permite la toma de los valores cada segundo, el sensor tiene una desventaja, su resolución, la cual es limitada (20% - 90%). Algo tener en cuenta es la relación entre distancia/pull-up, en otras palabras cuando el cable de conexión es menor a 20 metros, se recomienda una resistencia pull-up de 5 K Ω , y cuando el cable de conexión supera los 21 metros es recomendable escoger la mejor resistencia; pero se recomienda una de 12 K Ω .

El sensor posee cuatro pines de activación y funcionamiento.

- Pin 1 Pin de entrada de voltaje positivo (5 VCD).
- Pin 2 Pin de salida.
- Pin 3 Pin NC (no connection).
- Pin 4 Pin de conexión a un chasis o GND (ground).

1.9 OSILADOR DE CUARZO

El oscilador es utilizado como una fuente de reloj, en otras palabras como un tipo de “ritmo musical” que el microcontrolador usa para ejecutar el código en cierta manera. Sin un reloj, el microcontrolador no sabe cómo correr el código, y con un reloj irregular (uno que varíe mucho) el código correrá con un paso indeterminado.

Existen varios tipos de osciladores que se utilizan en los microcontroladores:

- RC Externo.- Generalmente usado para aplicaciones de bajo costo. Utiliza un resistor y un capacitor, la razón de carga/descarga puede ser usada como un reloj de entrada.
- RC Interno.- El propio microcontrolador puede generar su propio reloj.
- Oscilador Externo.- Es usado de forma estándar se conecta un cristal de cuarzo (abreviado como cristal o XTAL) a los dos pines OSC y el código se ejecutará a la frecuencia entregada.
- Reloj Externo.- Es utilizado en vez de un oscilador, y se puede ocupar un generador de pulsos de reloj, el cual es ocupado cuando se necesita que múltiples componentes corran con la misma frecuencia.

El oscilador externo es el más utilizado **figura 16**, generalmente los osciladores de cristal de cuarzo vienen para diferentes frecuencias. Algunas de las frecuencias más típicas son de 30MHz, 20MHz, 16MHz, 10MHz y 4MHz.

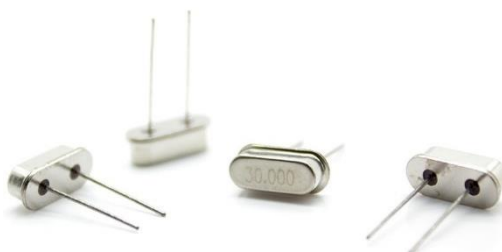


Figura 16.- Osciladores de cuarzo

Fuente: Autores

El oscilador de cuarzo está constituido generalmente por un pequeño cristal de cuarzo cortado con precisión para que la pieza de cristal vibre a una frecuencia específica de allí su nombre. El cristal es un elemento absolutamente pasivo que requiere algún mecanismo externo para volverse un “oscilador”. Lo cual conlleva a la utilización de condensadores de carga. (Seidle , 2008)

1.10 REGULADOR DE VOLTAJE LM7805

El regulador LM7805 es un dispositivo electrónico que permite la limitación del voltaje, permitiendo obtener un voltaje de salida de +5VDC, independiente al voltaje de entrada.

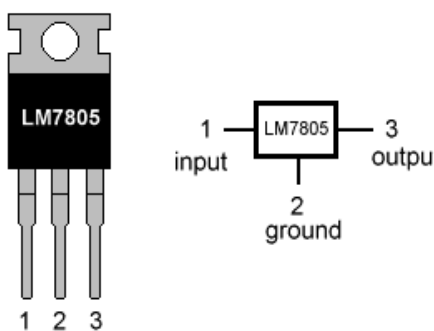


Figura 17.- LM7805 Regulador de voltaje 5VCC
Fuente: Los Autores

El regulador LM7805 pertenece a la familia de la serie de LM78XX los cuales son muy utilizados en el desarrollo de sistemas empotrados por su versatilidad como regulador lineal, además con el dissipador correcto se puede obtener la mayor eficiencia en corriente de estos dispositivos, los LM78XX posee protección por sobrecarga térmica y cortocircuitos.

A continuación se muestra una tabla de los reguladores existentes de la familia LM78XX

Tabla 2.- Reguladores LM78XX

Fuente: Los Autores

Designación	Voltaje de entrada	Voltaje de salida
LM7805	7-25	5
LM7806	8-25	6
LM7808	10,5-25	8
LM7809	11,5-25	9
LM7810	12,5-25	10
LM7812	14,5-30	12
LM7815	17,5-30	15
LM7818	21-33	18
LM7824	27-38	24

La **figura 17** muestra al regulador con sus pines, el cual permite la salida de voltaje según el encapsulado; los pines están enumerados desde el lado izquierdo con el número uno, siempre y cuando la inscripción del encapsulado de hacia el usuario; la forma de polarizar para su correcto funcionamiento es el siguiente:

- Pin número 1.- Pin de entrada de voltaje (input), el cual soporta el ingreso de tensión eléctrica de acuerdo a la tabla de especificaciones antes mencionado.
- Pin número 2.- Pin tierra común.
- Pin número 3.- Pin de voltaje de salida.

1.11 MÓDULO ETHERNET

Tiene la particularidad de emular sistemas de control y monitoreo a nivel industrial sobre una red Ethernet. Existen varios módulos controlares en el mercado, como se citan a continuación:

1.11.1 MODULO DE COMUNICACIÓN ETHERNET ENC28J60

Este entrenador fue desarrollado por la empresa MICROCHIP su limitación es el uso de microcontroladores de la misma corporación. Su electrónica es la necesaria para conectarse directamente a un Router mediante un patch-cord de cobre (RJ-45). Su comunicación es mediante el bus I²C. **Figura 18.**



Figura 18.- Modulo de comunicación Ethernet ENC28J60

Fuente: http://www.unrobotica.com/proyctorevista/Manual_ModEthernet_V1.0.pdf

1.11.2 MODULO ETH484

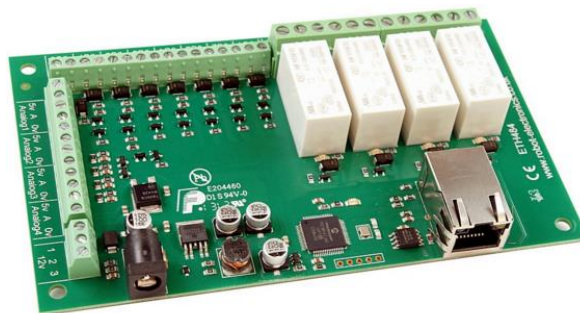


Figura 19.- Modulo de comunicación Ethernet ETH484

Fuente: <http://www.superrobotica.com/Images/S310238big.JPG>

En la **figura 19** se muestra la placa ETH484, siendo esta de uso netamente industrial. Este módulo posee cuatro entradas analógicas las cuales controlaran a sus cuatro relés que posee como salidas.

La desventaja que posee el ETH484 no son sus relés, son sus cuatro entradas analógicas las cuales no pueden ser cambiadas, ya que en el campo de la electrónica a un puerto analógico se lo puede convertir en un puerto digital.

1.11.3 MODULO SHIELD ETHERNET

La **figura 20** muestra a una placa desarrollada por ARDUINO, este shield permite conectar una placa Arduino con una LAN, basándose en el chip de Ethernet Wiznet W5100, su comunicación lo realiza por el protocolo SPI. El problema con este módulo es la utilización de microcontroladores solo de ATMEL, dejando imposibilitado el uso de otros microcontroladores.

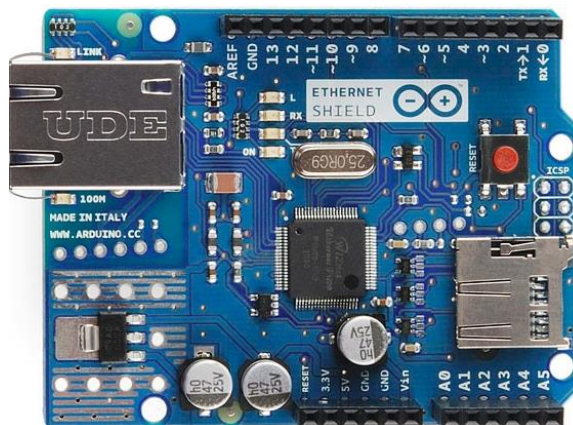


Figura 20.- Módulo Shield Ethernet

Fuente: <http://mikroe.es/wp-content/uploads/2014/01/ArduinoEthernetShieldV3.jpg>

1.11.4 MODULO CONTROLADOR ETHERNET FUNDUINO

El módulo controlador Ethernet Funduino **figura 21**, tiene una gran similitud al módulo Shield Ethernet de Arduino, la diferencia que existe entre estos dos controladores además de su color y tamaño, es la aplicación que se le puede dar al módulo controlador FUNDUINO con cualquier microcontrolador, en otras palabras se puede utilizar cualquier marca de microcontrolador. Por tal motivo, el módulo controlador Ethernet Funduino ha sido escogido para el desarrollo del entrenador. La ventaja de este controlador es poseer el chip de red el Wiznet5100 y su comunicación con el protocolo SPI permitiendo la

comunicación con el microcontrolador, el cual puede ser configurado como un servidor web.



Figura 21.- Módulo controlador Ethernet Funduino

Fuente: Los Autores

El módulo controlador Ethernet Funduino destaca tanto el conector hembra RJ-45, leds de señalización, pines de conexión (10) y el chip de red.

El controlador posee 10 pines de interconexión muy importantes para la comunicación en la **figura 22** se muestran los pines que serán detallados en la **tabla 3**:



Figura 22.- Pines del Controlador Ethernet

Fuente: Los Autores.

Tabla 3.-Designación de los pines del controlador Ethernet.

Fuente: Los Autores

PINES DEL CONTROLADOR ETHERNET FUNDUINO	
Nombre del Pin	Designación
P+ (POE+)	“POE” (power over ethernet) permite alimentar dispositivos con el mismo cable de red, está es una tecnología que incorpora alimentación eléctrica a una infraestructura LAN.
P- (POE-)	
G (GND)	Conector del polo negativo
MI (MISO)	“MI” en el módulo representa a MISO (master input slave output) que en efecto es la salida de datos del dispositivo “esclavo” y entrada al dispositivo que hace de “jefe”.
MO (MOSI)	“MO” al igual que MI representa a MOSI (master output slave input) en cambio este permite la salida de datos del “jefe” y entradas de datos al “esclavo”.
CK (CLOCK)	“SOK” en el módulo representa el SCLK ó CLOCK este pin permite el paso del pulso de sincronización es decir que con cada pulso de este reloj, se lee o envía un bit.
SS (NSS)	“NSS” se utiliza para seleccionar un “esclavo”, o para que el “jefe” le diga al “esclavo” que se active, también se lo conoce como SS/Select.

R (RESET)	RST es un pin utilizado para el reset del módulo controlador.
+5 (VCD)	Conector del polo positivo
G (GND)	Conector del polo negativo

Sus leds de señalización **figura 23** son importantes los cuales también son detallados en la **tabla 4**:



Figura 23.- LED's señalización
Fuente: Los Autores.

Tabla 4.- Designación de los pines del controlador Ethernet.

Fuente: Los Autores

LEDs DEL CONTROLADOR ETHERNET FUNDUINO	
Nombre del LED	Designación
Tx	El led está intermitente cuando el controlador envía datos.
Rx	El led está intermitente cuando el controlador recibe datos.
COL	El led está intermitente cuando la conexión colisionó o se borró.

FDX	Indica que la conexión de red es full dúplex.
SPD	Indica la presencia de una red de conexión de 100 Mb/s.
LINK	Indica la presencia de una conexión de red y el led está en intermitencia cuando el controlador Ethernet transmite o recibe datos.
PWR	Indica que el módulo está energizado.

Cuando se conecta el modulo controlador Ethernet al computador o router se debe considerar utilizar patch cord estándar Ethernet CAT5/CAT6 (categoría 5 o categoría 6) con conector RJ-45.

1.11.5 WIZNET W5100

En la **figura 24** se observa al chip Wiznet W5100, o dispositivo de red programable que se puede utilizar como elemento de conexión que no requiere “esfuerzo” de dispositivos externos para interactuar entre el chip y el mainbord del módulo, además el Wiznet W5100 ofrece una red (IP) capaz de trabajar con TCP²². (WIZ811 MJ Datasheet, 2012, pág. 05)

Además el chip de red wiznet W5100 ofrece en su propio encapsulado con varias ventajas a utilizar como son el protocolo TCP/IP, MAC (control de acceso al medio) y PHY (interface física entre el MAC y el medio inalámbrico) con soporte de comunicación SPI tipo bus.

²² Protocolo de control de transmisión.



Figura 24.- Chip WIZnet Ethernet W5100.

Fuente: <http://www.wiznet.co.kr/product-item/w5100/>

1.12 PROTOCOLO DE COMUNICACIÓN SPI

El protocolo SPI es un estándar de enlace de datos seriales sincronizados por un reloj que operan en modo full dúplex (**figura 25**). Los dispositivos se comunican en modo maestro/esclavo donde el dispositivo maestro inicia el data frame (trama de red o marco de datos). Múltiples dispositivos esclavos están permitidos en líneas Slave Select (SS) individuales. (López Pérez, 2005).

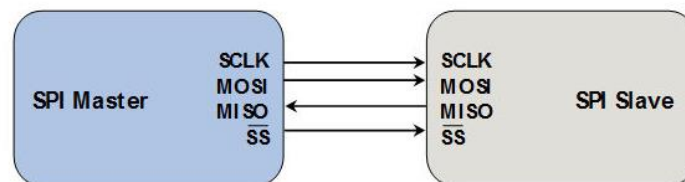


Figura 25.- SPI Master-Slave

Fuente: Los Autores

Las ventajas que se puede mencionar con la utilización del protocolo SPI son varias, algunas de ellas son:

- Mayor velocidad de transmisión que con I²C.
- Consume menos energía, debido que posee menos circuitos como son las resistencias pull-up utilizadas en I²C.
- Los dispositivos clientes usan el reloj que envía el servidor.
- Protocolo flexible en que se puede tener un control absoluto sobre los bits transmitidos.

1.13 SERVIDOR WEB

El servidor web no es más que el almacenamiento de datos o información para su posterior envío vía internet (LAN, MAN, PAN, WAN). El almacenamiento de la información es en forma de página web. Cuando el usuario accede a un servidor web, envía una petición HTTP²³ que recoge el servidor, luego envía la información a través del mismo protocolo, al cliente en formato HTML²⁴.

Es importante saber que un servidor web tiene que tener acceso al exterior (Internet), pero tampoco es estrictamente necesario. Por ejemplo, un mismo equipo puede tener instalado el servidor web, la base de datos y actuar también como cliente de acceso a través del navegador de Internet. (Granados La Paz, 2014)

1.14 LENGUAJE HTML

HTML es un lenguaje de programación usado para la creación de páginas web de una manera fácil en cualquier editor de texto, para ello se necesita del editor de texto que siempre viene en los sistemas operativos, como es en el caso de Windows se utiliza el *bloc de notas*, en MS-DOS el *edit* y en Linux el *vi*.

- **SINTAXIS HTML**

HTML basa su sintaxis en un elemento de base al que llama etiqueta. La etiqueta presenta frecuentemente dos partes:

Apertura y cierre; una **apertura** de forma general “<ETIQUETA>” y un **cierre** de tipo “</ETIQUETA>”; todo lo incluido en el interior de esa etiqueta sufrirá las modificaciones que caracterizan a la misma, por ejemplo:

La etiqueta y definen un texto en negrita por lo que si se escribe

²³ protocolo de transferencia de hipertexto

²⁴ Lenguaje de marcado de hipertexto

 Hola Mundo

El resultado sería: **Hola Mundo**

- **ESTRUCTURA DE UN DOCUMENTO HTML**

HTML por ser un texto plano, es definido por sus etiquetas (< >) y posee tres partes principales:

- Inicio y Fin de la Programación. <HTML> ... </HTML>
- Cabecera del documento. <HEAD> ... </HEAD>
- Contenido del documento. <BODY> ... </BODY>

Quedando la estructura del documento:

```
<HTML>
    <HEAD>
    </HEAD>
    <BODY>
    </BODY>
</HTML>
```

- Cabecera: se inicia mediante el comando <HEAD> y se termina con </HEAD>. Dentro de la cabecera hay información del documento, que no se ve en la pantalla principal, y que precisa las características del documento, principalmente el título del documento. El título del documento se declara entre las etiquetas <TITLE> y </TITLE>, este debe ser breve y descriptivo del contenido.
- Cuerpo: utiliza el comando <BODY> y se termina con el comando </BODY>; este comando acepta varios modificadores. (UNIVERSIDAD DE MURCIA, 2008)

CAPÍTULO II

2 METODOLOGÍA DE LA INVESTIGACIÓN

2.1 TIPO DE ESTUDIO.

El estudio es de tipo experimental, ya que se hace un control de la variable independiente (uso de entrenador), con el fin de analizar o caracterizar el tiempo de desarrollo de la práctica de laboratorio.

Es de tipo inductivo, por cuanto a partir de un experimento controlado se busca generalizar sus resultados en cuanto a la influencia del uso de entrenadores en el desarrollo de las prácticas de laboratorio.

2.2 POBLACIÓN Y MUESTRA

2.2.1 POBLACIÓN

La población está formada por los estudiantes de la asignatura de microcontroladores de la carrera de Ingeniería Electrónica y Telecomunicaciones de la Universidad Nacional de Chimborazo en el periodo Octubre 2015 – Marzo 2016.

2.2.2 MUESTRA

Por tratarse de un estudio experimental no es necesario tomar una muestra sino repeticiones del experimento, para lo cual se necesita como mínimo tres unidades experimentales en cada tratamiento del experimento.

Para garantizar que las variaciones observadas del tiempo de desarrollo en las prácticas se deban a los cambios inducidos a la variable independiente (uso del entrenador), es necesario controlar las covariables para garantizar la homogeneidad de las unidades experimentales (estudiantes), por lo tanto se tomará en cuenta la siguiente característica en el proceso de selección: repitencia de los estudiantes.

2.2.3 HIPOTESIS

¿El uso de entrenadores en microcontroladores reduce el tiempo de desarrollo en las prácticas de laboratorio?

2.3 OPERACIONALIZACIÓN DE VARIABLES

La **tabla 5** indica la operacionalización de variables.

Tabla 5.- Operacionalización de variables

Fuente: Los Autores

VARIABLE	TIPO	CATEGORIA	ESCALA
INDEPENDIENTE Uso del entrenador	Cualitativa	<ul style="list-style-type: none">• Sí• No	Nominal
DEPENDIENTE Tiempo de elaboración de la práctica de laboratorio.	Cuantitativa		Razón

2.4 PROCEDIMIENTOS

A continuación se describe los pasos para el desarrollo e implementación del módulo entrenador en microcontroladores para red Ethernet, el cual será ampliado junto al diseño y programación aplicado al control de sus cargas:

- Luz.
- Humedad.
- Temperatura.

2.4.1 DISEÑO DE CIRCUITO ELECTRONICO Y PCB PARA EL MÓDULO ETHERNET.

2.4.1.1 DISEÑO DEL CIRCUITO ELECTRÓNICO.

Para el correcto funcionamiento de los módulos se utiliza siempre reguladores de voltaje los cuales debe utilizar capacitores. La **figura 26** muestra como el capacitor C_I es de $0.33 \mu\text{F}$, este capacitor es requerido cuando la fuente de tensión es una fuente VCD a mayor rango de los 5VCD y el capacitor C_O tiene el valor de $0,1 \mu\text{F}$ se lo debe colocar para el mejoramiento de la señal, previniendo el rizamiento del voltaje y así obtener la estabilidad y mejor performance del dispositivo al que se le suministre en la salida.

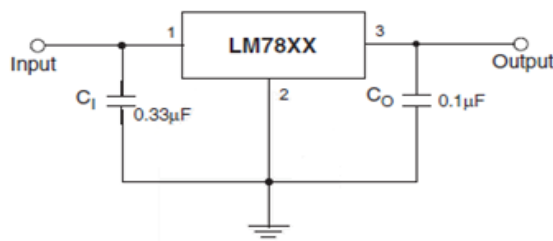


Figura 26.-Polarización de capacitores en el regulador de voltaje

Fuente: Los Autores

- **Calculo del capacitor**

Para el cálculo del capacitor se utiliza la siguiente ecuación:

$$C = \frac{i_L}{V_r f} = \frac{V_m}{R_L V_r f}$$

V_m = Voltaje máximo (Soporta voltaje de entrada del LM7805) .

R_L = Resistencia de carga. = $714[\Omega]$

V_r = Frecuencia de la red eléctrica = $60[\text{Hz}]$.

f = Valor del ripple o rizo = 2

$$R_L = \frac{25}{35[mA](max. consumo del micro)} = 714[\Omega]$$

$$C = \frac{25[V]}{714[\Omega] \cdot 60[Hz] \cdot 2}$$

$$C = 290\mu F$$

El diseño del circuito del controlador Ethernet Funduino se lo realizó con la ayuda del software PROTEUS.

Para realizar el rutado, seleccionamos el trazado de bordes de la placa dimensionado 10*10 cm, se realiza el desplazamiento de los elementos de una forma ordenada y comprensiva para que el diseño del circuito pueda ser comprendido, **figura 27**, el enrutado de las pistas se efectúa a T30, T40 y V60 **figura 28** (estos valores especifican el espesor de las líneas de comunicación).

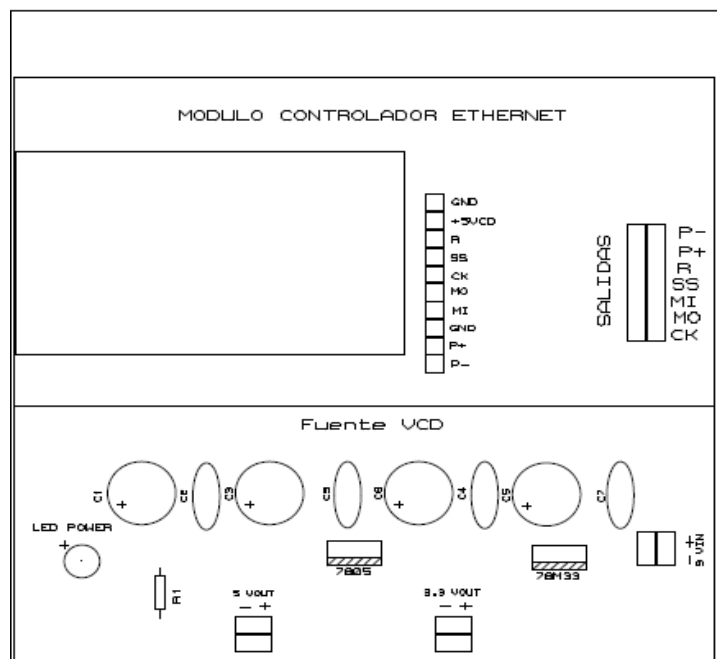


Figura 27.-Diseño Ethernet

Fuente: Los Autores

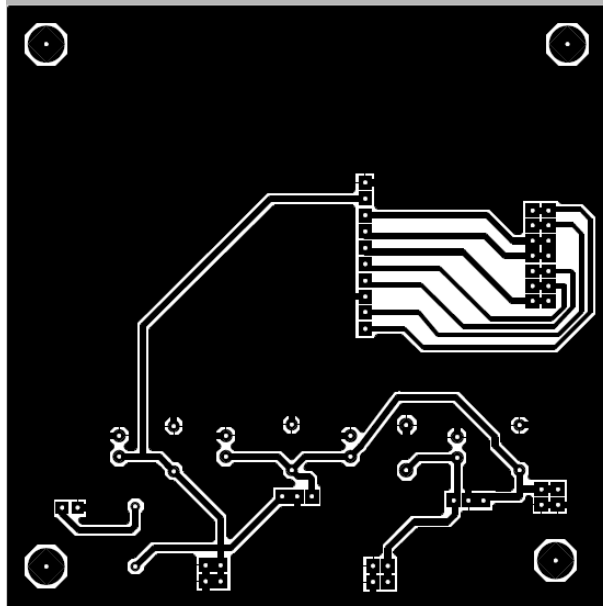


Figura 28.- Diseño de pistas, placa controlador Ethernet Funduino

Fuente: Los Autores

2.4.1.2 DISEÑO DEL CIRCUITO PCB PARA EL MÓDULO ETHERNET

Es una gran ayuda el tener disponible la herramienta 3D de PROTEUS, ya que proporciona información tridimensional, ya que es de suma importancia conocer la separación de cada uno de los componentes del diseño para que los elementos no se sobrepongan uno sobre otro. También es utilizado con fines estéticos para su presentación.

Figura 29.

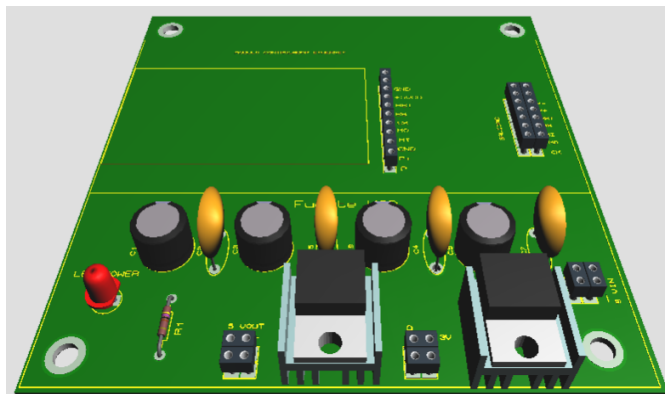


Figura 29.- Módulo controlador Ethernet Funduino 3D

Fuente: Los Autores.

2.4.2 DISEÑO DE CIRCUITO ELECTRONICO Y PCB PARA EL MÓDULO DISPLAY's.

2.4.2.1 DISEÑO DEL CIRCUITO ELECTRÓNICO.

Este módulo está compuesto generalmente por led los cuales son utilizados como señalética, para ello se debe realizar el cálculo de la resistencia de protección del led. Para proceder con el cálculo se debe considerar el triángulo de Ohm. **Figura 30.**

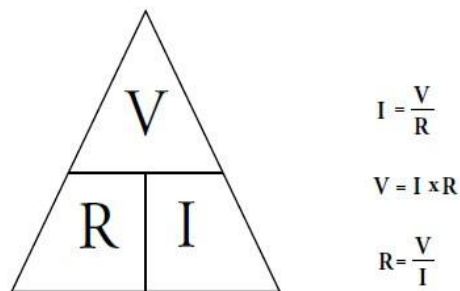


Figura 30.-Triángulo de la ley de Ohm
Fuente: Los Autores

Con el triángulo de Ohm una vez establecido se sigue con el cálculo respectivo de la resistencia de protección del diodo led tomando en cuenta la fórmula correspondiente para la resistencia.

Datos.

$$V = 9 \text{ [Vcd]}$$

$$I = 25 \text{ [mA]}$$

$$R = \frac{V}{I}$$

$$R = \frac{9}{250 \text{ mA.}} = R = 360 \Omega$$

Para el diseño del circuito del módulo Display's se lo realizó en PROTEUS. Para realizar el rutado, seleccionamos el trazado de bordes de la placa dimensionado 10*10 cm, se realiza el desplazamiento de los elementos de una forma ordenada y comprensiva para que el diseño del circuito pueda ser comprendido, **figura 31**, el enrutado de las pistas se efectúa T30, T40 y V60 **figura 32** (estos valores especifican el espesor de las líneas de comunicación).

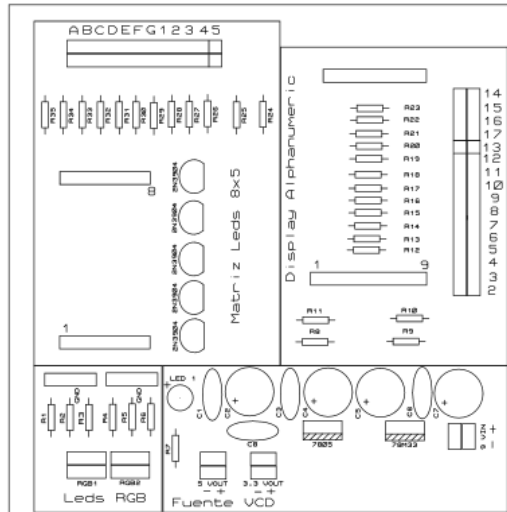


Figura 31 Módulo Display's

Fuente: Los Autores

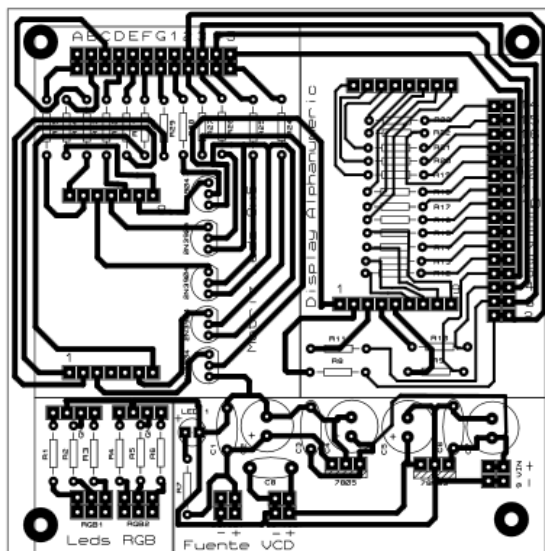


Figura 32.- Diseño de pistas, módulo Display's

Fuente: Los Autores

2.4.2.2 DISEÑO DEL CIRCUITO PCB PARA EL MÓDULO DISPLAY's

La herramienta 3D de PROTEUS, proporciona información tridimensional, lo que permite conocer la separación entre componentes para evitar la sobre posición de los elementos. También es utilizado con fines estéticos para su presentación. **Figura 33**



Figura 33.-Módulo LCD

Fuente: Los Autores

2.4.3 DISEÑO DE CIRCUITO ELECTRONICO Y PCB PARA EL MÓDULO LCD.

2.4.3.1 DISEÑO DEL CIRCUITO ELECTRÓNICO.

El diseño del circuito del LCD se lo realizó en PROTEUS. Para realizar el rutado, seleccionamos el trazado de bordes de la placa dimensionado 10*10 cm, se realiza el desplazamiento de los elementos de una forma ordenada y comprensiva para que el diseño del circuito pueda ser comprendido, **figura 34**, el enrutado de las pistas se efectúa a T30, T40 y V60 **figura 35** (estos valores especifican el espesor de las líneas de comunicación).

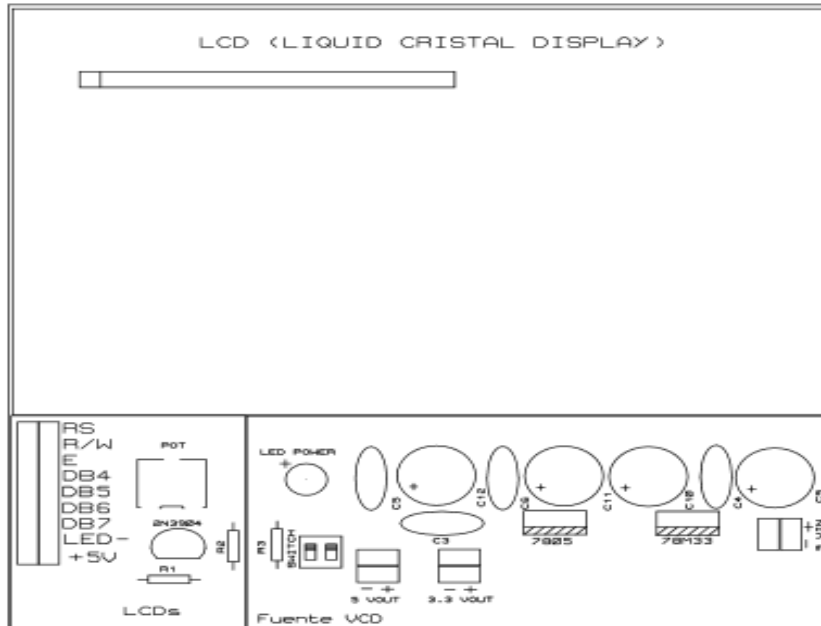


Figura 34.- Placa LCD's
Fuente: Los Autores

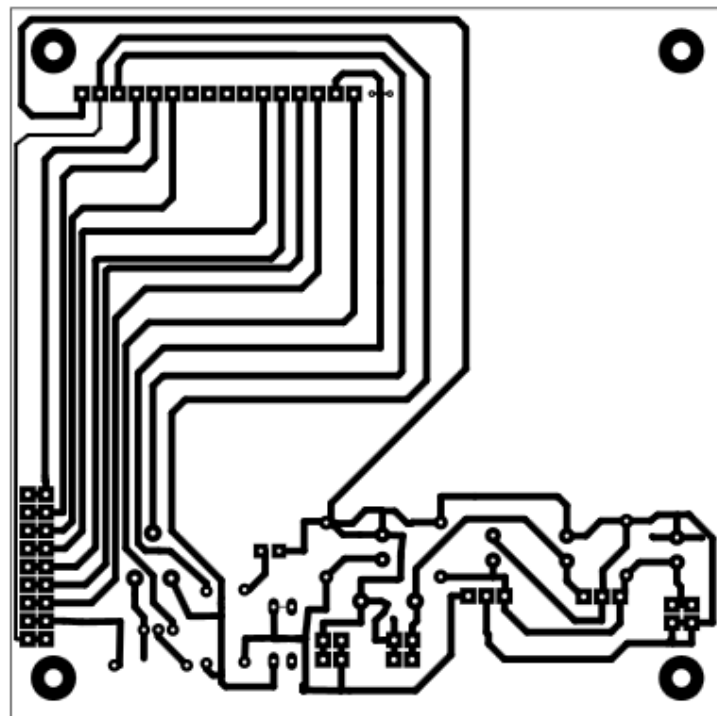


Figura 35.- Diseño de pistas, placa LCD
Fuente: Los Autores

2.4.3.2 DISEÑO DEL CIRCUITO PCB PARA EL MÓDULO DISPLAY's.

La herramienta 3D de PROTEUS, proporciona información tridimensional, lo que permite conocer la separación entre componentes para evitar la sobre posición de los elementos. También es utilizado con fines estéticos para su presentación. **Figura 36.**

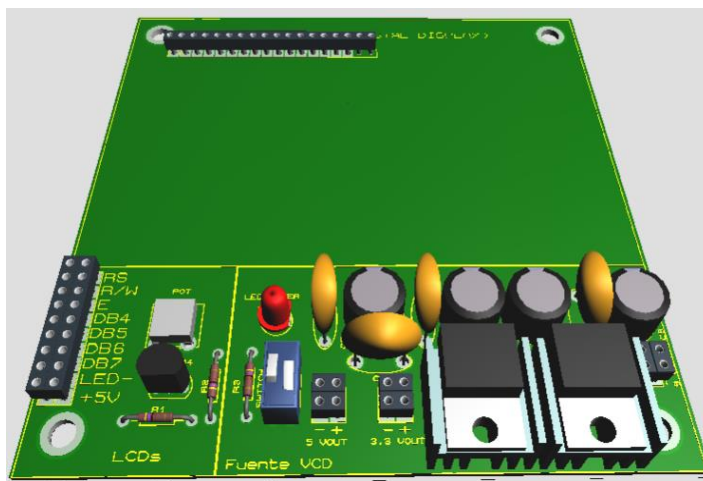


Figura 36.-Módulo LCD

Fuente: Los Autores

2.4.4 CONFIGURACIÓN DE DISPOSITIVOS (MICROCONTROLADOR, FUNDUINO, COMPUTADOR)

La configuración de los dispositivos permitirá el correcto alineamiento para que exista la comunicación entre microcontrolador, computador, módulo Ethernet y los actuadores mediante la red LAN.

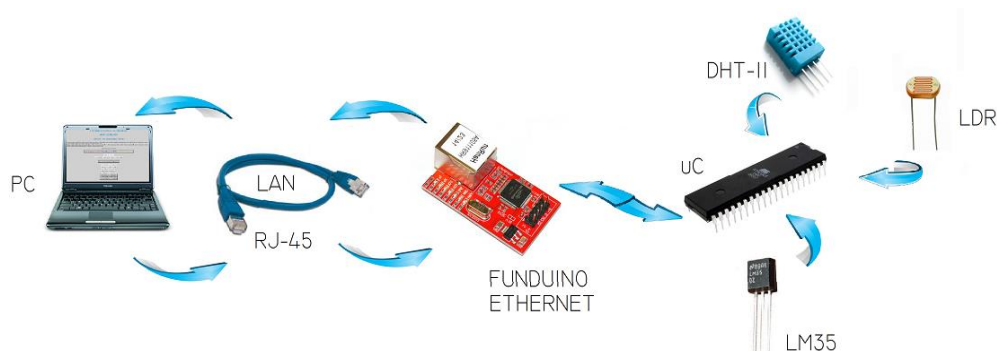


Figura 37.- Topología de Conexión

Fuente: Los Autores.

2.4.4.1 CONFIGURACIÓN DE RED

Para la configuración del módulo controlador Ethernet se procede a la conexión del patch cord Ethernet CAT6 en el slot de red tanto del módulo controlador como del ordenador o computador; al momento de la conexión del cable de los dispositivos, automáticamente se identifica como una red alámbrica, **figura 38**.

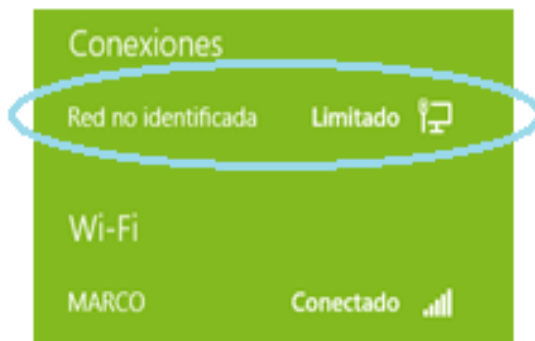


Figura 38.- Red alámbrica de conexión modulo Ethernet - Computador
Fuente: Los Autores.

Para el correcto funcionamiento del ordenador controlador Ethernet se debe configurar al módulo con la dirección IP que posee el computador para que ambos dispositivos se encuentren en red (LAN). Para ello se utilizará la consola de MS-DOS o CMD de Windows y se tomará nota de los parámetros (Dirección IPv4, Máscara de subred, Puerta de enlace predeterminada) que muestra el Adaptador de Ethernet Ethernet, para su posterior utilización. **FIGURA 39**.

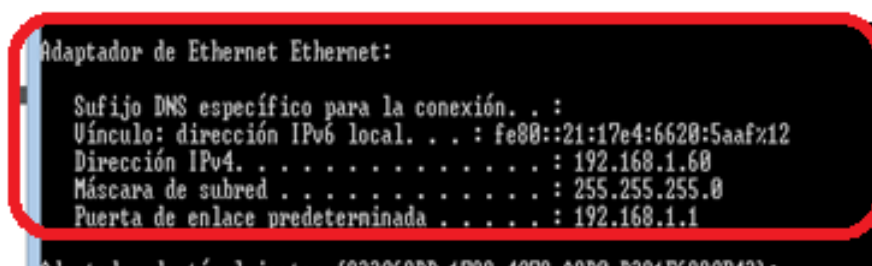


Figura 39.- Red alámbrica de conexión modulo Ethernet – Computador
Fuente: Los Autores.

2.4.4.2 CONFIGURACIÓN DEL IDE BASCOM PARA LA COMUNICACIÓN SPI/IP.

En este apartado se detallará la configuración que se debe realizar en el IDE²⁵ BASCOM para que el microcontrolador tenga conectividad con el modulo controlador Ethernet y el ordenador, para ello necesitamos el microcontrolador antes mencionado el ATmega644P, la plataforma de programación BASCOM-AVR versión [1.11.9.8] o superiores y el modulo controlador Ethernet Funduino.

Lo primero que se hace en BASCOM-AVR es configurar al microcontrolador, con su respectiva numeración, el cristal u oscilador y la velocidad de comunicación en baudios, como se muestra en la **figura 40**.

```
-----  
$regfile = "m644def.dat"      Microcontrolador  
$crystal = 16000000          Oscilador  
$baud = 9600                 Velocidad en Baudios  
-----
```

Figura 40.- Conf. Microcontrolador en el IDE de BASCOM.
Fuente: Los Autores.

Luego se procede con la activación de la comunicación tipo bus mediante el protocolo SPI con los siguientes comandos. **Figura 41**

```
'Configuration of the SPI-bus  
  
Config Spi = Hard , Interrupt = Off ,  
Data Order = Msb , Master = Yes ,  
Polarity = Low , Phase = 0 , Clockrate = 4 ,  
Noss = 0
```

Figura 41.- Comandos de activación protocolo SPI.
Fuente: Los Autores.

²⁵ Ambiente de desarrollo integrado.

Lo importante a tomar en cuenta, es el direccionamiento IP, el cual es colocado mediante programación en el microcontrolador, para ello se debe considerar lo explicado anteriormente en el subcapítulo 2.4.2.1 Configuración de red, de este documento.

En la **figura 36** se muestra los parámetros a tener en cuenta:

- Dirección IPv4 :192.168.1.60
- Máscara de subred..... :255.255.255.0
- Puerta de enlace predeterminada :192.168.1.1

Estos parámetros son los que permiten estar en red tanto al ordenador, al microcontrolador y al controlador Funduino. Los comandos a utilizar en el IDE de BASCOM para la configuración del microcontrolador como Master-Slave y la inicialización del protocolo SPI son sumamente sencillos. Para inicializar la comunicación por el protocolo SPI el comando a utilizar en BASCOM es “**Spinit**”. El comando a utilizar para las componentes de red son:

Const IP [] Comando para configurar la dirección IP.

Const Sn [] Comando para configurar la Máscara de red.

Const Gw [] Comando para configurar el Gateway.

Const Mac [] Comando para configurar la MAC.

En la **figura 42** se muestra como es la utilización de los comandos para la activación del protocolo SPI y para que haya comunicación entre los tres dispositivos (ordenador, microcontrolador, Funduino).

```

Spinit
Const Ip[1] = 192
Const Ip[2] = 168
Const Ip[3] = 1
Const Ip[4] = 60

Const Sn[1] = 255
Const Sn[2] = 255
Const Sn[3] = 255
Const Sn[4] = 0

Const Gw[1] = 192
Const Gw[2] = 168
Const Gw[3] = 1
Const Gw[4] = 1
  
```

Figura 42.- Comandos de activación protocolo SPI y enlace de micro con ordenador.

Fuente: Los Autores.

2.4.5 DISEÑO DEL SISTEMA DE CONTROL Y MONITOREO DE UN VIVERO MEDIANTE RED ETHERNET.

El diseño fue realizado para el control y monitoreo de un vivero, mediante la red Ethernet; consta del módulo Funduino el cual permite la interacción de los elementos con el microcontrolador y el servidor web. **Figura 43**

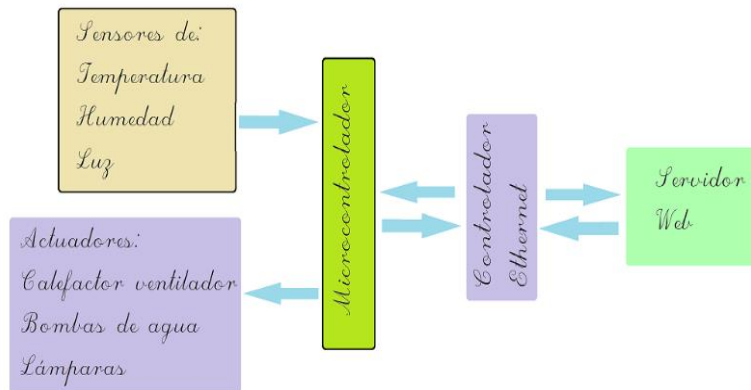


Figura 43.- Entrada y salida del control del vivero.
Fuente: Los Autores.

2.4.5.1 DISEÑO CIRCUITO ELECTRONICO CONTROLADOR FUNDUINO.

2.4.5.2 CIRCUITO ELECTRONICO-SENSORES.

El diseño de los circuitos de los sensores se lo realizó con la ayuda del software PROTEUS **figura 44** y consta de tres sensores:

- LM35 Medirá la temperatura.
- DHT11 Controlará los niveles de humedad.
- LDR Administrará el nivel de lúmenes.

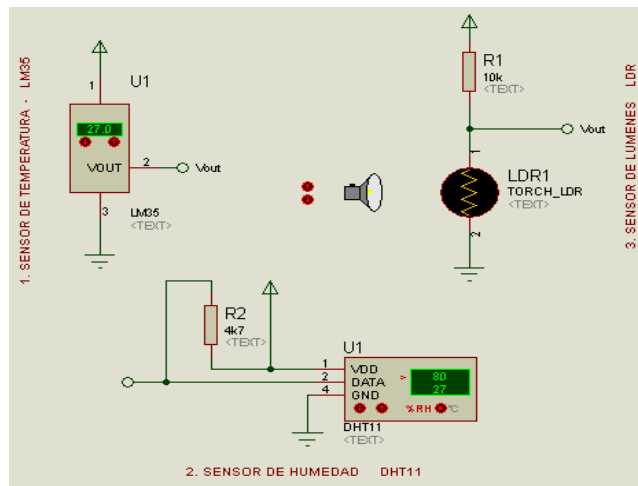


Figura 44.- Circuito electrónico de los sensores

Fuente: Los Autores

2.4.5.3 CIRCUITO DE ACTIVACIÓN DE LOS ACTUADORES.

El circuito de activación, está compuesto de un opto acoplador el cual sirve como protección contra voltajes y corrientes transitorias, permitiéndole aislar el circuito de control del circuito de potencia. Al opto acoplador se conecta un relé que funciona como un switch permitiendo la activación de la carga según lo estipulado en el microcontrolador o en el servidor web.

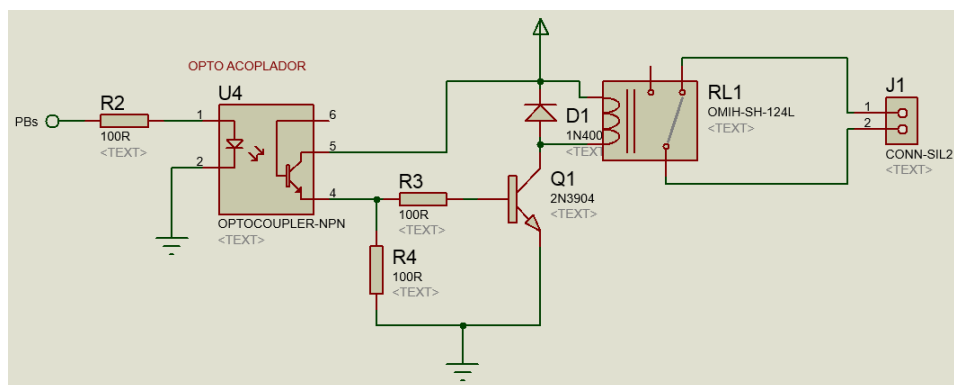


Figura 45.- Circuito activación-actuadores.

Fuente: Los Autores.

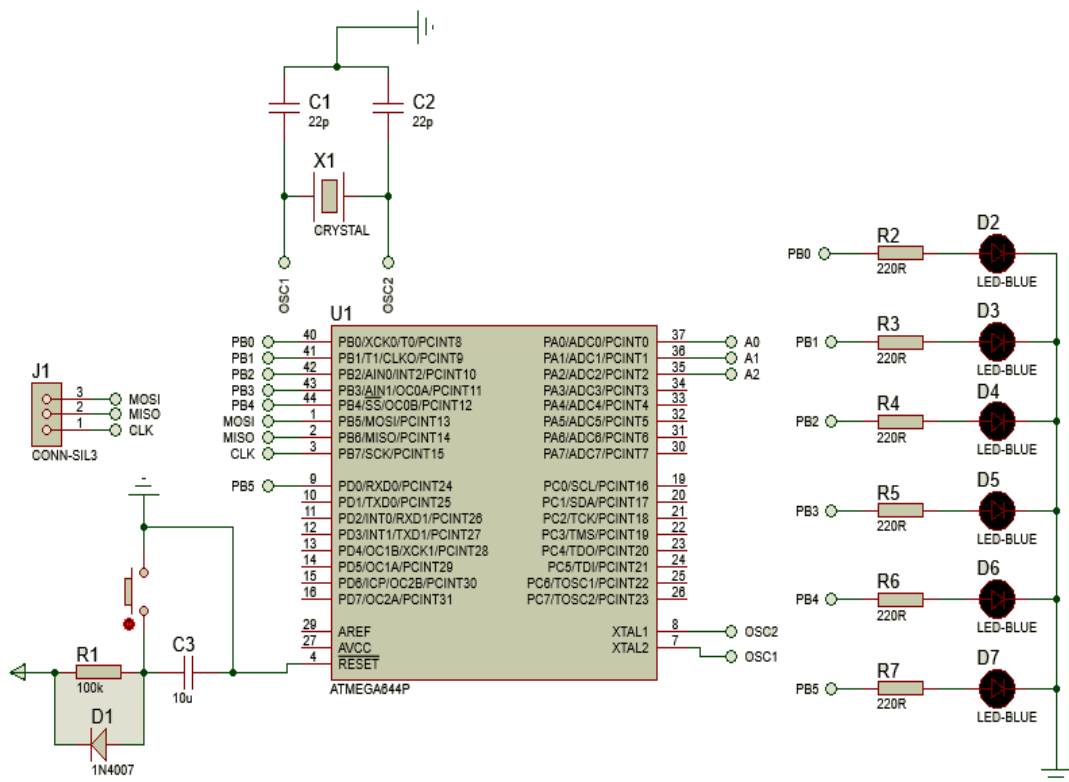


Figura 47.-(B) Circuito electrónico para el control y monitoreo de un vivero.
Fuente: Los Autores.

2.4.5.5 CIRCUITO GENERAL DEL CONTROL DEL SISTEMA DE UN VIVERO POR MEDIO DEL CONTROLADOR ETHERNET.

El circuito fue diseñado con el propósito de controlar un vivero, con la utilización del módulo Ethernet y con la ayuda de un microcontrolador. Para la administración y control de los componentes electrónicos se utilizó el microcontrolador ya antes mencionado el ATmega644P, este microcontrolador se encuentra conectado por medio de sus pines para la ejecución del programa realizado junto a los tres circuitos antes mencionados (sensores, actuadores y microcontrolador-funduino).

2.4.5.6 PROGRAMACIÓN DEL MICROCONTROLADOR

A continuación se detallará la programación del microcontrolador para ejecutar el control de los actuadores.

2.4.5.6.1 PROGRAMACIÓN DEL CONTROL MANUAL DEL VENTILADOR.

La **figura 47** se muestra el diagrama de flujo el cual da el patrón de lógica de programación en el que el microcontrolador opera en la etapa de control de temperatura, cuando el sensor (LM35) determine la magnitud fuera del rango previamente establecido.

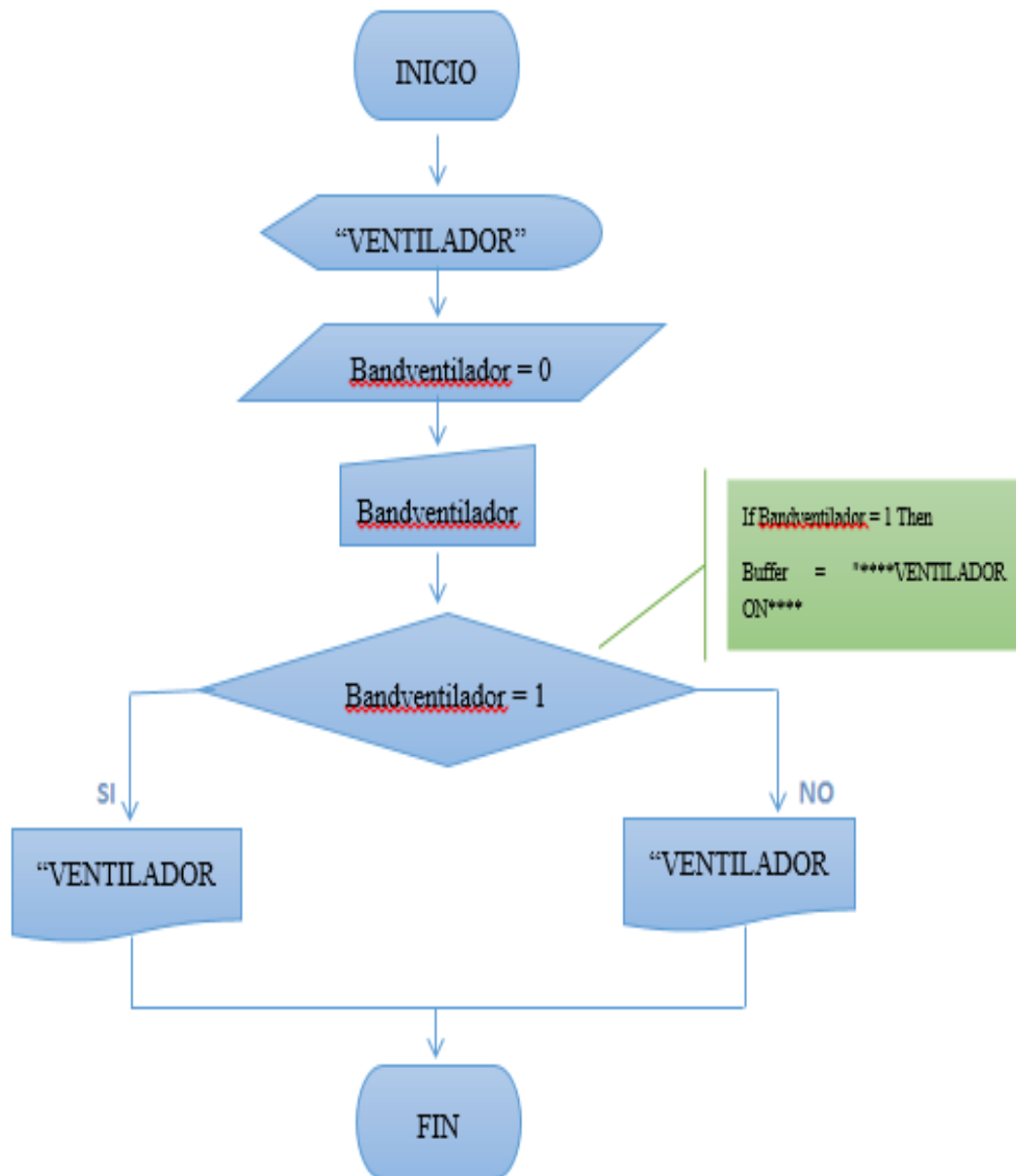


Figura 48.- Flujograma de programación de encendido del ventilador.

Fuente: Los Autores

En la **figura 48**, se ve la programación para el encendido del ventilador cuando el sensor de temperatura envíe el dato de la temperatura; el cual exceda el rango permitido se procederá a encender el ventilador. Siguiendo el mismo patrón para la activación de los demás actuadores. Como se muestra en la **figura 49** y **figura 50**.

```

Elseif Buffer = "%VENTILADOR%" Then
  If Bandventilador = 1 Then
    Buffer = "*VENTILADOR ON*"
  Else
    Buffer = "*VENTILADOR OFF*"
  End If
If Instr(buffer , "?VENTILADORON") > 0 Then
  Ventilador = 1
  Bandventilador = 1
  'Waitms 500
  'Relais1 = 0
End If
If Instr(buffer , "?VENTILADOROFF") > 0 Then
  Ventilador = 0
  Bandventilador = 0
  'Waitms 500
  'Relais2 = 0
End If

```

Figura 49.- Programación de encendido-apagado del ventilador.
Fuente: Los Autores.

```

If Instr(buffer , "?CALEFACTORON") > 0 Then
  Calefactor = 1
  Bandcalefactor = 1
  'Waitms 500
  'Relais1 = 0
End If
If Instr(buffer , "?CALEFACTOROFF") > 0 Then
  Calefactor = 0
  Bandcalefactor = 0
  'Waitms 500
  'Relais2 = 0
End If
If Instr(buffer , "?CALEFACTORON") > 0 Then
  Calefactor = 1
  Bandcalefactor = 1
  'Waitms 500
  'Relais1 = 0
End If
If Instr(buffer , "?CALEFACTOROFF") > 0 Then
  Calefactor = 0
  Bandcalefactor = 0
  'Waitms 500
  'Relais2 = 0
End If

```

Figura 50.- Programación de encendido-apagado del calefactor.
Fuente: Los Autores.

```

If Instr(buffer , "?LUCESON") > 0 Then
  Luces = 1
  Bandluces = 1
  'Waitms 500
  'Relais1 = 0
End If

If Instr(buffer , "?LUCESOFF") > 0 Then
  Luces = 0
  Bandluces = 0
  'Waitms 500
  'Relais2 = 0
End If

If Instr(buffer , "?LUCESON") > 0 Then
  Luces = 1
  Bandluces = 1
  'Waitms 500
  'Relais1 = 0
End If

If Instr(buffer , "?LUCESOFF") > 0 Then
  Luces = 0
  Bandluces = 0
  'Waitms 500
  'Relais2 = 0
End If

```

Figura 51.- Programación de encendido-apagado de luz.
Fuente: Los Autores.

2.4.5.7 VISUALIZACIÓN DEL SERVIDOR WEB.

La visualización del servidor web es gracias a la compatibilidad entre HTML y BASCOM. Ver anexo 1.

Además, la misma muestra varias opciones para el monitoreo del entorno a un vivero como se muestra en anexo B, las cuales son:

- Control Manual/Automático.
- Botones de SETUP para temperatura, humedad y lúmenes.
- Botones de On/Off de los actuadores en modo manual.
- Indicadores del estado de los Actuadores.
- Indicadores del estado de los sensores.

La pantalla que muestra el servidor web, da las opciones de accionar de forma manual los actuadores, configurar los rangos de los sensores, visualización en tiempo real el valor de los sensores.

2.5 PROCESAMIENTO Y ANÁLISIS



Figura 52.- Diagrama de procedimientos.
Fuente: Los Autores.

En la **figura 51**, muestra un resumen de las fases que conforman el proyecto, permitiendo llevar el proceso continuo en el desarrollo de la investigación.

2.5.1 ESTUDIO DEL ENTRENADOR DE MICROCONTROLADORES PARA RED ETHERNET.

El entrenador se comunica mediante un patch cord RJ45, por medio de una LAN, la misma encargada de enviar y recibir las sentencias del dispositivo y el entrenador. Para que haya comunicación entre el entrenador y la pc, deben estar conectados en la misma red, cumpliendo así con las acciones indicadas. **Figura 52.**

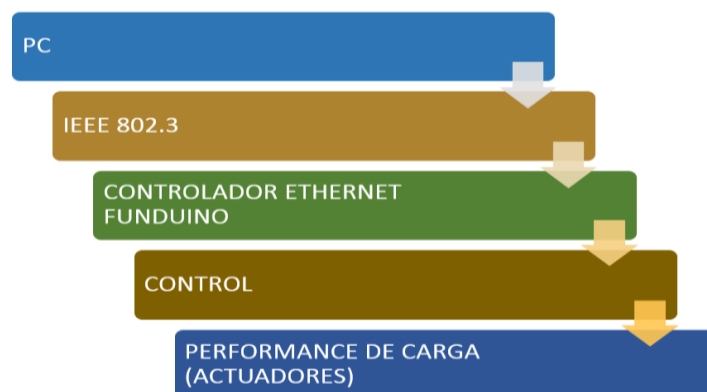


Figura 53.- Diagrama de procedimientos.
Fuente: Los Autores.

2.5.2 ANÁLISIS DE LA COMUNICACIÓN DEL SISTEMA

El módulo controlador Ethernet Funduino cumple con el estándar 802.3, el cual con su configuración correcta y por medio de HTML garantiza la comunicación entre el controlador y la pc.

El uso de un patch cord de cobre RJ45 certificado para la comunicación es un requerimiento indispensable junto a la configuración de red, los cuales permiten el acceso a la red LAN para lograr el control y monitoreo de la aplicación. Además se utiliza el protocolo de comunicación SPI, el cual permitirá el envío y recepción de datos de la manera más eficiente y rápida como lo caracteriza el estándar IEEE 802.3

2.5.3 ADMINISTRACIÓN DE DISPOSITIVOS

La correcta administración, permite la linealidad y el ordenamiento de la puesta de los dispositivos para un óptimo funcionamiento, permitiendo establecer un ensamblaje preciso y eficaz en la construcción del entrenador para cada función establecida.

2.6 COMPROBACIÓN DE LA HIPÓTESIS

El procedimiento mostrado a continuación sigue un conjunto de pasos para comprobar o refutar la hipótesis planteada, de acuerdo a los resultados obtenidos de la experimentación, es decir, los tiempos (en minutos) que tardaron los 22 estudiantes en desarrollar una práctica con y sin el uso del entrenador, los mismos que se muestran a continuación:

Tabla 6.- Tiempo de desarrollo de la práctica

Fuente: Los Autores.

Estudiante	Con entrenador	Sin entrenador
1	6.85	24.73
2	6.53	24.59
3	8.39	23.36
4	6.94	27.03
5	7.64	26.83
6	8.05	24.25
7	7.66	24.2
8	7.62	24.85
9	6.80	25.53

10	8.96	24.13
11	6.10	22.68
12	9.05	23.37
13	7.94	27.07
14	7.68	24.94
15	8.93	25.78
16	5.77	26.71
17	9.94	21.30
18	6.90	18.20
19	5.79	26.78
20	6.76	21.94
21	9.08	25.06
22	10.18	23.00

Se ha realizado una prueba de hipótesis para observaciones pareadas, la misma que es utilizada usualmente en diseño de experimentos cuando en las unidades experimentales se aplican dos tipos de tratamientos. Los pasos para realizar esta prueba de hipótesis son los siguientes:

Planteamiento de la hipótesis nula y alternativa

H_0 : El tiempo promedio que los estudiantes tardan en desarrollar una práctica con y sin el uso del entrenador es el mismo, por lo tanto, su diferencia es igual a cero

H_1 : El tiempo promedio que los estudiantes tardan en desarrollar una práctica con el uso del entrenador es menor al tiempo que tardan en desarrollar la práctica sin el uso del mismo, por lo tanto, su diferencia es menor que cero.

$$H_0: \mu_c - \mu_s = 0$$

$$H_1: \mu_c - \mu_s < 0$$

Establecimiento del nivel de significancia α

Los resultados de la investigación tienen una confiabilidad del 95%, por lo tanto el nivel de significancia es 0.05.

Cálculo del estadístico de prueba t

$$t = \frac{\bar{d} - d_0}{\frac{s_d}{\sqrt{n}}} = \frac{(-16.67) - 0}{\frac{2.71}{\sqrt{22}}} = -28.80$$

Donde:

\bar{d} y s_d representan a la media muestral y la desviación estándar de las diferencias de los tiempos que los estudiantes tardan en desarrollar una práctica con y sin el uso del entrenador.

d_0 es la diferencia hipotética de los tiempos medios que los estudiantes tardan en desarrollar una práctica con y sin el uso del entrenador.

n es el número de estudiantes que realizaron la práctica.

Tabla 7.- Diferencias entre los tiempos de desarrollo con y sin entrenador

Fuente: Los Autores.

Estudiante (i)	Con entrenador (C _i)	Sin entrenador (S _i)	diferencias (d _i =C _i -S _i)
1	6.85	24.73	-17.88
2	6.53	24.59	-18.06
3	8.39	23.36	-14.97
4	6.94	27.03	-20.09
5	7.64	26.83	-19.19
6	8.05	24.25	-16.2
7	7.66	24.2	-16.54
8	7.62	24.85	-17.23
9	6.8	25.53	-18.73
10	8.96	24.13	-15.17
11	6.1	22.68	-16.58
12	9.05	23.37	-14.32
13	7.94	27.07	-19.13
14	7.68	24.94	-17.26
15	8.93	25.78	-16.85
16	5.77	26.71	-20.94
17	9.94	21.3	-11.36
18	6.9	18.2	-11.3

19	5.79	26.78	-20.99
20	6.76	21.94	-15.18
21	9.08	25.06	-15.98
22	10.18	23	-12.82

Media \bar{d} = -16.67
Desviación estándar s_d = 2.71

Cálculo del valor crítico $t_{\alpha, n-1}$ y las regiones de aceptación y rechazo

Para el cálculo del valor crítico se busca en la tabla de la distribución t de Student, (ANEXO 1.) el valor de probabilidad para un nivel de significancia $\alpha = 0.05$ y grados de libertad $(n-1)=21$

$$t_{0.05, 21} = -1.72$$

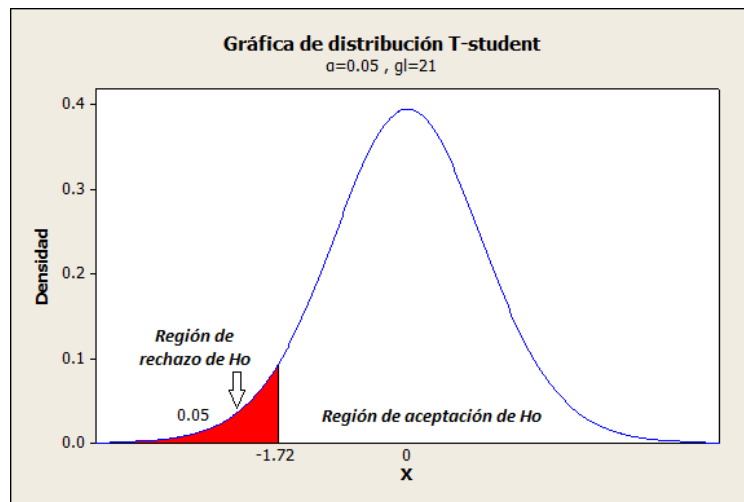


Figura 54.- Distribución de Probabilidad de la distribución t de Student

Fuente: Los Autores.

Regla de decisión

Si $t < t_{\alpha, n-1}$ se rechaza H_0

CAPÍTULO III

3 RESULTADOS

Como resultado tenemos que el estadístico de prueba $t=-28.80$, es menor que el valor crítico $t_{0.05; 21}=-1.72$ por lo que se rechaza la hipótesis nula (H_0), por tal motivo se concluye que existen evidencias suficientes para decir que el tiempo promedio que los estudiantes tardan en desarrollar una práctica con el uso del entrenador es significativamente menor al que tardan en desarrollar la práctica sin el uso del mismo con un 95% de confianza.

3.1 ANÁLISIS FINANCIERO

La **tabla 8** muestra la lista de elementos y materiales usados para la implementación de los módulos entrenadores.

Tabla 8.- Análisis Financiero

Fuente: Los Autores

CATIDAD	DETALLE
10	Módulo controlador Ethernet Funduino.
10	Módulos Sim900.
10	Módulos Xbee.
10	Módulos WiFi.
10	Módulos
50	Condensadore
100	Resistencias
60	Circuitos integrados LM7805
10	LCD's 16x4
10	Displays Alphanumeric
10	Matriz de LED's 7x5
10	Teclados
10	Sensores de humedad DHT-11
10	Fotorresistencias
10	Dipswitchs
10	Giroscopios
10	Módulos Bluetooth

20	LED's RGB
10	LED's rojos de señalización
10	Opto acopladores ECG3100
35	Regletas hembra dobles
4	Transeiver SV651
1	Patch cord de cobre RJ45
1	Microcontrolador 16F877A
1	Microcontrolador ATmega644P
20	Conectores mecánicos
2	Osciladores de 20 MHz
2	Osciladores de 16 MHz
1	Proto board
1	Carrete de suelda de estaño
1	Cautín de precisión
120	Baquelitas
5	Planchas de acrílico

El desarrollo de las piezas que conforman el entrenador se lo realizó de manera individual, de tal forma el presupuesto alcanzó a un total de \$1500 dólares americanos.

CAPÍTULO IV

4 DISCUSIÓN

El desarrollo del entrenador de microcontroladores en red Ethernet, está enfocado y sustentado al avance tecnológico de los dispositivos de telecontrol; a través de la investigación y la recolección íntegra de información acerca de nuevas técnicas entre las telecomunicaciones y control, se logró acertar con el módulo adecuado que permitan el desarrollo de un sistema estable sobre una red local.

La investigación se enfocó en el estudio de las telecomunicaciones, pero con más énfasis en el ámbito del Ethernet, el uso frecuente de esta tecnología se da en el sector productivo, administrativo e industrial por su gran versatilidad de adaptación al medio, es decir el Ethernet utiliza múltiples protocolos de comunicación y puede conectar entornos informáticos heterogéneos, incluyendo Netware, UNIX, Windows y Macintosh. Ethernet es también conocida como la red de optimización de procesos, ya que mejora la calidad y disponibilidad de la información, por tal motivo se lo toma como punto de estudio en el proyecto de investigación para ayudar a aprovechar esta tecnología.

El desarrollo de la aplicación para el monitoreo y control de un vivero cumple con un esquema lógico, donde debe cumplir el envío de sentencias dada desde el ordenador para la activación de los actuadores y la lectura de las mediciones de los sensores. El software utilizado para el desarrollo de la aplicación fue Bascom y HTML por su gran facilidad de manejo.

Con la culminación del prototipo, los equipos y dispositivos son capaces de cumplir eficazmente y con mínimo error todas las expectativas planteadas de forma eficiente.

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El entrenador en microcontroladores para red Ethernet, permite la agilidad en el ensamblaje y ahorro de tiempo.
- El control de los dispositivos se efectúa mediante la conexión a la red LAN, sin la necesidad de tener servicio de Internet.
- Se podrá desarrollar sistemas embebidos para controlar, supervisar y operar equipos semi-industriales.

5.2 RECOMENDACIONES

- Poseer un patch cord certificado categoría 5e o más.
- La selección de los modos manual/automático, setup's y encendido/apagado se lo hace manualmente y uno a la vez.
- Es necesario establecer la conexión entre los dos dispositivos (PC y Funduino) antes de ejecutar cualquier función.
- En la programación del acceso de red se debe tener en cuenta el direccionamiento IP, ya que los dispositivos deben estar sobre la misma red.
- Verificar que el software BASCOM, sea compatible con el código html. Versión de compatibilidad de BASCOM 1.11.9.8 o más avanzados.
- El uso del entrenador, debe ser con cuidado ya que está hecho de un material sensible a rasguños.
- Tener en cuenta las conexiones necesarias para el funcionamiento del controlador Funduino es decir MISO, MOSI, CLK.
- Para el correcto funcionamiento del controlador Funduino debe tener una alimentación de voltaje y corriente adecuada.

CAPÍTULO VI

6 PROPUESTA

6.1 TÍTULO DE LA PROPUESTA

“DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE MICROCONTROLADORES PARA RED ETHERNET”

6.2 INTRODUCCIÓN

La tecnología hoy en día, avanza a pasos colosales, especialmente en el campo de las telecomunicaciones y telecontrol, por tal motivo para estar actualizado e ir a la par del avance tecnológico se necesita determinar e investigar los métodos de comunicación de una red Ethernet que permitan el correcto funcionamiento en la transmisión y recepción de datos. Después de la investigación de los requisitos y elementos necesarios para la conexión, se integra una aplicación que permita interconectar entre un microcontrolador y una red Ethernet.

Para cristalizar los objetivos, se desarrollará un esquema aplicativo de diseño utilizando el entrenador de tal manera que el desempeño del sistema sea eficiente y cumpla con todas las exigencias necesarias: comunicación por medio de la red LAN, monitoreo de las magnitudes de cambio de medidas de los sensores y el control utilizando un servidor web para el mando de los actuadores.

Con la investigación antepuesta que se hará a las tecnologías se podrá determinar que módulo, software y aplicación se debe utilizar para culminar el proyecto consiguiendo la eficacia y precisión de los dispositivos.

6.3 OBJETIVOS

6.3.1 OBJETIVO GENERAL

Diseñar e implementar un entrenador de microcontroladores para realizar comunicaciones a través de Red Ethernet.

6.3.2 OBJETIVO ESPECÍFICO

- Diseñar entrenadores con arquitectura abierta para el desarrollo de aplicaciones con distintos microcontroladores existentes en el mercado.
- Elaborar un manual de prácticas.
- Desarrollar una aplicación de comunicación vía red Ethernet utilizando el entrenador de microcontroladores, para el monitoreo de sensores y funcionamiento de sus cargas.
- Utilizar el entrenador como plataforma para el desarrollo de sistemas embebidos.
- Fortalecer el nivel académico e investigativo de los estudiantes y docentes de la Escuela de Electrónica y Telecomunicaciones.

6.4 DESCRIPCIÓN DE LA PROPUESTA

La propuesta se basa en la exigüidad de equipos para realizar estudios y mejorar así las líneas de investigación en el campo de los microcontroladores con aplicaciones enfocadas al desarrollo en telecomunicaciones. El uso de tecnologías de información y comunicación (TIC) servirán para el desarrollo de aplicaciones útiles para monitorear sensores y controlar sus cargas por medio de la red LAN, permitiendo que un servidor web sirva como control de todo el sistema.

El sistema garantiza dos tipos de control, la primera de forma manual y la segunda de forma automática, en la forma manual el usuario podrán activar los actuadores según vea conveniente. En la segunda parte de forma automática, el servidor web recibirá los datos provenientes de los sensores, indicando que alguna magnitud medida por los sensores se encuentre fuera del rango establecido previamente, realizara la acción encomendada como la activación o desactivación de los actuadores y el monitoreo de las magnitudes.

6.5 DISEÑO ORGANIZACIONAL

En la **figura 54** se observa el esquema organizacional y funcional de la unidad administrativa con la se ejecutó el desarrollo y la implementación del proyecto.

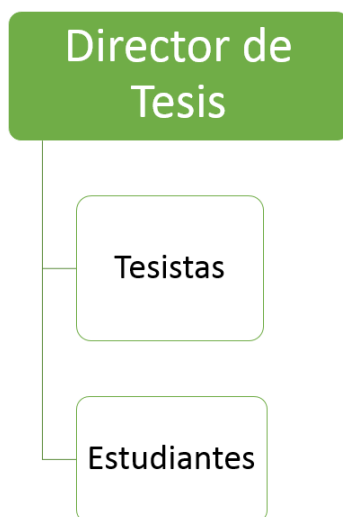


Figura 55.- Diagrama de procedimientos.
Fuente: Los Autores.

6.6 MONITOREO Y EVALUACIÓN DE LA PROPUESTA

Para el monitoreo y evaluación de la propuesta, se realizaran pruebas con los estudiantes en el armado de una práctica, tomando en cuenta los tiempos de elaboración de dicha práctica con el entrenador y sin el entrenador. Además se tendrá en cuenta la ejecución del monitoreo de un vivero a través de una red LAN con el uso del controlador Funduino, determinando la confiabilidad de la transmisión y recepción de datos.

Para evaluar el sistema se realizará un manual de prácticas para que el usuario pueda realizar los trabajos y aplicaciones en el entrenador.

CAPÍTULO VII

7 BIBLIOGRAFÍA

Atmel Corporation. (28 de 02 de 2013). *ATmega164P/324P/644P datasheet*. Recuperado el 08 de 09 de 2015, de atmel: http://www.atmel.com/Images/Atmel-8011-8-bit-AVR-Microcontroller-ATmega164P-324P-644P_datasheet.pdf

Chauhan, D. S., & Kulshreshtha, D. C. (2009). *Electronics Engineering : (As Per The New Syllabus, B.Tech. I Year Of U.P. Technical University)*. New Age International. Recuperado el 21 de Octubre de 2015

González Rojas, K. V., & Méndez Caviativa, J. A. (23 de 09 de 2011). *Diseño y Fabricación de un módulo entrenador implementado la familia de microcontroladores dsPIC33F*. Recuperado el 03 de 04 de 2015, de repository.upb.edu.co: http://repository.upb.edu.co:8080/jspui/bitstream/123456789/1669/1/digital_21147.pdf

Granados La Paz, R. L. (2014). *Desarrollo de aplicaciones web en el entorno servidor*. (1° ed.). ANTEQUERA, Málaga: IC Editorial. Recuperado el 08 de 09 de 2015

López Pérez, E. (07 de 08 de 2005). *INGENIERIA EN MICROCONTROLADORES*. Recuperado el 28 de 04 de 2015, de i-micro: <http://www.i-micro.com/pdf/articulos/spi.pdf>

N/A. (2006). *LA REVISTA*. Recuperado el 20 de 08 de 2015, de LAREVISTAINFORMATICA.COM: <http://www.larevistainformatica.com/LENGUAJES-DE-PROGRAMACION.html>

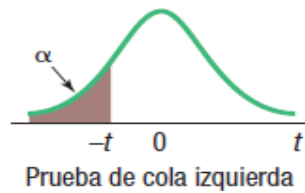
- Remache Bevides, W. A. (20 de 09 de 2009). *Diseño de una red Ethernet/IP para la implementación de un ambiente industrial*. Recuperado el 03 de 04 de 2015, de repositorio.espe: <http://repositorio.espe.edu.ec/bitstream/21000/482/1/T-ESPE-026669.pdf>
- Reyes, C. A. (2006). *Microcontroladores PIC Programación en Basic*. QUITO: RISPGRAF.
- Romero, D. M. (23 de 10 de 2005). *Introducción a Ethernet Industrial*. Recuperado el 03 de 04 de 2015, de IEEE: <http://www.ieee.org.ar/downloads/romero-eth-ind.pdf>
- Rosales, J. (10 de Marzo de 2013). *Manual de redes informáticas*. Recuperado el 04 de 05 de 2015, de mredesinfo: <http://mredesinfo.blogspot.com/>
- Sánchez Lobón, J. (09 de 10 de 2007). *Ethernet Industrial*. Recuperado el 03 de 04 de 2015, de isa-spain: http://www.isa-spain.org/images/biblioteca_virtual/rt%20isa%20ethernet%20industrial.pdf
- Seidle , N. (19 de Junio de 2008). *Beginning Embedded Electronics - 3*. Recuperado el 16 de Octubre de 2015, de sparkfun: <https://www.sparkfun.com/users/7185>
- Tanenbaum, A. S. (2003). *Redes de Computadoras*. Mexico: PEARSON EDUCACIÓN.
- TEXAS INSTRUMENTS. (09 de 04 de 2015). *LM35 Precision Centigrade Temperature Sensors*. Recuperado el 21 de 04 de 2015, de ti: <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- UNIVERSIDAD DE MURCIA. (13 de 08 de 2008). *MANUAL BÁSICO DE CREACIÓN DE PÁGINAS WEB*. Recuperado el 28 de 10 de 2015, de Universidad de Murcia: <https://www.um.es/atika/documentos/html.pdf>
- WIZ811 MJ Datasheet. (12 de 12 de 2012). Recuperado el 03 de 04 de 2015, de sparkfun: https://www.sparkfun.com/datasheets/DevTools/WIZnet/DEV-09473-Datasheet_V_1.1.pdf

CAPÍTULO VIII

8 ANEXOS

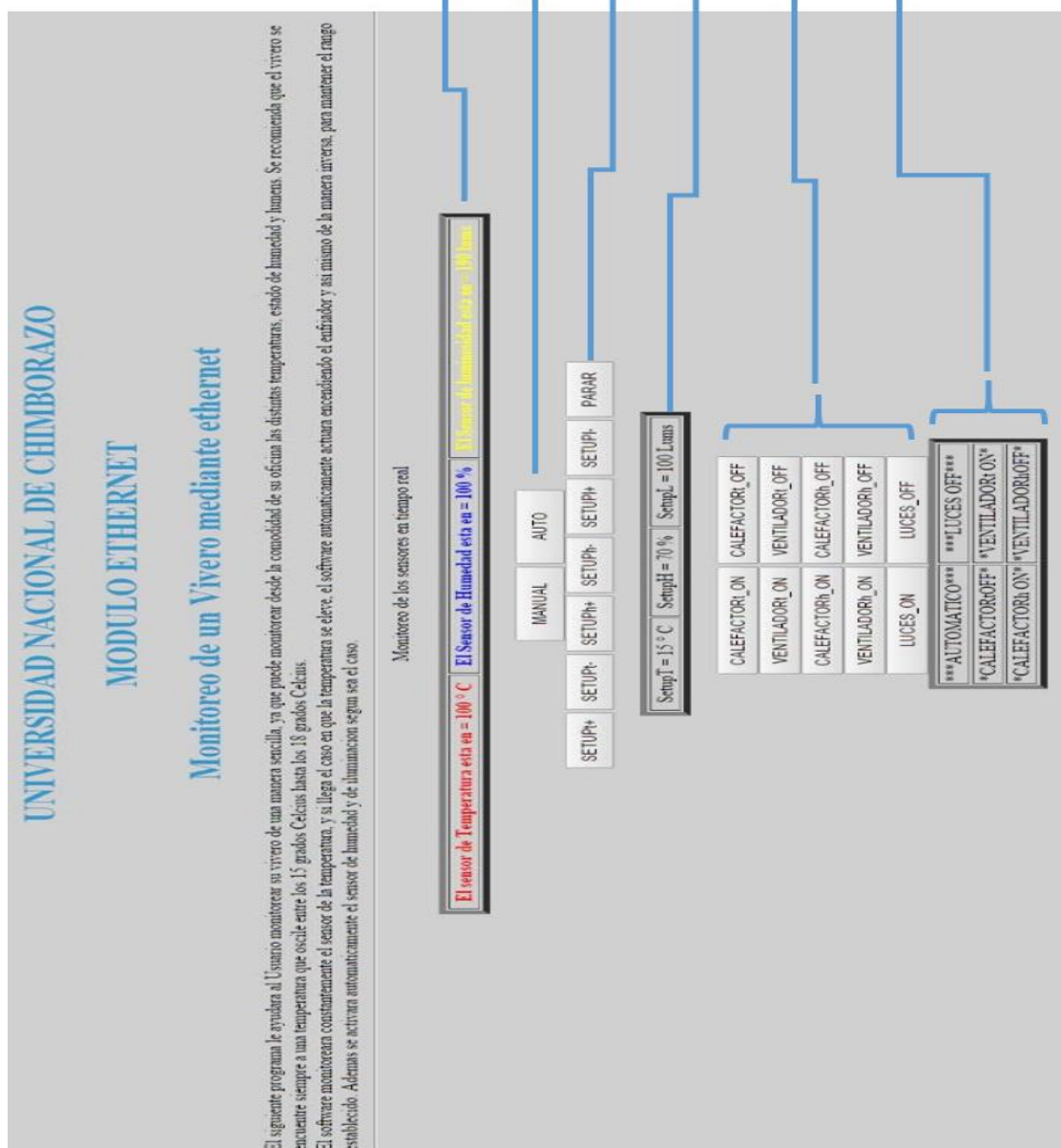
8.1 ANEXO 1. TABLA DISTRIBUCIÓN t STUDENT

Distribución t de Student



gl	Nivel de significancia para una prueba de una cola, α					
	0.10	0.05	0.025	0.01	0.005	0.0005
1	3.078	6.314	12.706	31.821	63.657	636.619
2	1.886	2.920	4.303	6.965	9.925	31.599
3	1.638	2.353	3.182	4.541	5.841	12.924
4	1.533	2.132	2.776	3.747	4.604	8.610
5	1.476	2.015	2.571	3.365	4.032	6.869
6	1.440	1.943	2.447	3.143	3.707	5.959
7	1.415	1.895	2.365	2.998	3.499	5.408
8	1.397	1.860	2.306	2.896	3.355	5.041
9	1.383	1.833	2.262	2.821	3.250	4.781
10	1.372	1.812	2.228	2.764	3.169	4.587
11	1.363	1.796	2.201	2.718	3.106	4.437
12	1.356	1.782	2.179	2.681	3.055	4.318
13	1.350	1.771	2.160	2.650	3.012	4.221
14	1.345	1.761	2.145	2.624	2.977	4.140
15	1.341	1.753	2.131	2.602	2.947	4.073
16	1.337	1.746	2.120	2.583	2.921	4.015
17	1.333	1.740	2.110	2.567	2.898	3.965
18	1.330	1.734	2.101	2.552	2.878	3.922
19	1.328	1.729	2.093	2.539	2.861	3.883
20	1.325	1.725	2.086	2.528	2.845	3.850
21	1.323	1.721	2.080	2.518	2.831	3.819
22	1.321	1.717	2.074	2.508	2.819	3.792
23	1.319	1.714	2.069	2.500	2.807	3.768
24	1.318	1.711	2.064	2.492	2.797	3.745
25	1.316	1.708	2.060	2.485	2.787	3.725
26	1.315	1.706	2.056	2.479	2.779	3.707
27	1.314	1.703	2.052	2.473	2.771	3.690
28	1.313	1.701	2.048	2.467	2.763	3.674
29	1.311	1.699	2.045	2.462	2.756	3.659
30	1.310	1.697	2.042	2.457	2.750	3.646

8.2 ANEXO 1. VISUALIZACIÓN DEL SERVIDOR WEB.



8.3 ANEXO 2. CONFIGURACIÓN DE FUSES PARA CARGAR EL PROGRAMA AL MICROCONTROLADOR

The screenshot shows the AVR Studio Fuse configuration window. The interface includes a menu bar (File, Command, Buffer, Select, About) and a toolbar with buttons for PROGRAM, BUFFER, CONFIG, Readme, and Update. The main area is divided into several sections:

- Fuse Bits:**
 - Low Fuse Bits:** CLKDIV8 (0), CKOUT (1), SUT1 (1), SUT0 (0), CKSEL3 (0), CKSEL2 (0), CKSEL1 (1), CKSEL0 (0).
 - High Fuse Bits:** OCDEN (1), JTAGEN (0), SPIEN (0), WDTON (1), EESAVE (1), BOOTSZ1 (0), BOOTSZ0 (0), BOOTRST (1).
 - Extend Fuse Bits:** BODLEVEL2 (1), BODLEVEL1 (1), BODLEVEL0 (1).
 - Lock Bits:** NA (1), NA (1), BLB12 (1), BLB11 (1), BLB02 (1), BLB01 (1), LB2 (1), LB1 (1).
- Programmer Interface:** USBASP, usb, Select CPU (ATmega644), Chip Signature (1E : 96 : 09 RD).
- Calibration:** 8.0 MHz (00), 2MHz (00), 4MHz (00), 8MHz (00), Read.
- Buttons:** Load Flash, Read Flash, Load Eeprom, Read Eeprom.
- Flash:** 0/65536, Eeprom: 0/2048.
- Programming Options:**
 - Read Signature
 - Chip Erase
 - Blank Check
 - Program FLASH
 - Program EEPROM
 - Verify FLASH
 - Verify EEPROM
 - Program Fuse
 - Lock Chip
 - Data Reload
 - Enabled XTAL
 - Image Data
- ConfigBit Navigation:** LowValue (62), HighValue (99), ExtValue (FF), Lock Value (FF).
- Buttons:** Read, Write, Erase, Auto.
- State Bar:** State: Ready, Use Times: 00:00:00, Copyright(r) Zhifeng Software, Inc 2007.

8.4 ANEXO 3. PROGRAMACIÓN

```
$regfile = "m644Pdef.dat"           'MICROCONTROLADOR A UTILIZAR
'$regfile = "m328pdef.dat"
$crystal = 1600000
$baud = 9600                        'put terminal on 9600 baud
$hwstack = 80
$swstack = 80
$framesize = 90
'Used variables
Dim Bandcalefactor As Bit
Dim Bandventilador As Bit
Dim Bandluces As Bit
Dim Bandestado As Bit
Dim Bandsetup As Byte
Dim A As Byte
Dim Wert As Byte
Dim Status As Byte
Dim Oldstatus As Byte
Dim Pointer As Word
Dim Tmp_value As Byte
Dim Tmp_str As String * 1
Dim Highbyte As Byte
Dim Lowbyte As Byte
Dim X As Integer
Dim Userpass As String * 20
Dim Tmp_auth As String * 20
Dim Authentication As Byte
Dim Website As Byte
Dim Buffer As String * 700
Dim Setup As Byte
***** Converso ADC *****
```

```

Config Adc = Single , Prescaler = Auto , Reference = Avcc

Start Adc

Dim A0 As Word

Dim A00 As Single

Dim A1 As Word

Dim A11 As Single

Dim A2 As Word

Dim A22 As Single

Dim Valor As String * 10

Dim Var As String * 5

Const Setmayor = 40

Const Setmenor = 10

Const Http_auth = 0

Userpass = "admin:demo"

Tmp_auth = Base64enc(userpass)

'Other used ports and pins

Luces Alias Portb.0

Ventilador Alias Portb.1

Calefactor Alias Portb.2

Manauto Alias Portb.3

'Config the port/pin directions

Config Luces = Output

Config Ventilador = Output

Config Calefactor = Output

Config Manauto = Output

Luces = 0

Ventilador = 0

Calefactor = 0

Manauto = 0

Config Watchdog = 2048                                'Watchdog configuration for stable reboot if system hangs

'Configuration of the SPI-bus

Config Spi = Hard , Interrupt = Off , Data Order = Msb , Master = Yes , Polarity = Low , Phase = 0 , Clockrate = 4 , Noss = 0

```



```

'Init the spi pins

Spiinit

Const Ip[1] = 192
Const Ip[2] = 168           'Own IP-Adress
Const Ip[3] = 254
Const Ip[4] = 80
Const Sn[1] = 255          'Own Networkmask
Const Sn[2] = 255
Const Sn[3] = 255
Const Sn[4] = 0
Const Gw[1] = 192          'IP-Adress of the gateway
Const Gw[2] = 168
Const Gw[3] = 254
Const Gw[4] = 1
Const Mac[1] = &HDE        'Own MAC-Adress
Const Mac[2] = &HAD
Const Mac[3] = &HBE
Const Mac[4] = &HEF
Const Mac[5] = &HFE
Const Mac[6] = &HED
Const Wiz5100_opcode_read = 15      'OP-Code for Read from the Wiz5100
Const Wiz5100_opcode_write = 240    'OP-Code for write to the Wiz5100

'Here we declare the used sub routines

Declare Sub Wiz5100_init
Declare Sub Wiz5100_readvalue(byval Highbyte As Byte , Byval Lowbyte As Byte)
Declare Sub Wiz5100_writevalue(byval Highbyte As Byte , Byval Lowbyte As Byte , Byval Value As Byte)
Declare Sub Wiz5100_receive
Declare Sub Wiz5100_send
Declare Sub Wiz5100_disconnect
Declare Sub Wiz5100_reset

Call Wiz5100_init           'We initialize the Wiz5100

Start Watchdog             'Start the watchdog

```

```

Bandcalefactor = 0

Bandventilador = 0

Bandluces = 0

Bandestado = 0

Bandsetup = 0

Setup = 15

Do

Reset Watchdog           'Reset the watchdog

'Get socket status

Call Wiz5100_readvalue(&H4 , &H3)

Status = Wert

If Status <> Oldstatus Then

    Oldstatus = Status

End If

'Connection was closed, we start the socket new

If Status = &H0 Or Status = &H1C Then

    Call Wiz5100_disconnect()

End If

'As long as connection is established (&H17) we will look if client send new data

If Status = &H17 Then

    Buffer = ""

    Call Wiz5100_receive()

End If

Loop

End

'Disconnect connection: close socket, reopen socket and wait for new connection

Sub Wiz5100_disconnect

    Call Wiz5100_writevalue(&H4 , &H1 , &H0)           'CLOSE

    Call Wiz5100_writevalue(&H4 , &H1 , &H1)           'OPEN

    Call Wiz5100_writevalue(&H4 , &H1 , &H2)           'LISTEN

End Sub

'Send headers and websites

```

Sub Wiz5100_send

Local Sendsize As Integer , Pos_tx As Word , Freesize As Word , Offset As Word , Tx_wr As Word

Local Startadress As Word , Offsend As Word , Uppersize As Word , Glaenge As Word , Startpos As Word

Local Adctemp As Word , Tablewidth As Word , Color As String * 7

'We restore the used website data

If Website = 0 Then

 Restore Websitewrong

Elseif Authentication = 0 Then

 Restore Websiteok

Else

 Restore Websiteauth

End If

Do

 Read Buffer

 'Look for the end of a website

 If Buffer = "%END%" Then

 Exit Do

 End If

 'Look for variable and replace them, here we can insert new own variables if we want

 If Buffer = "%ESTADO%" Then

 If Bandestado = 1 Then

 Buffer = "*****MANUAL*****"

 Else

 Buffer = "****AUTOMATICO****"

 End If

 Elseif Buffer = "%CALEFACTOR%" Then

 If Bandcalefactor = 1 Then

 Buffer = "*CALEFACTOR ON*"

 Else

 Buffer = "*CALEFACTOR OFF*"

 End If

 Elseif Buffer = "%VENTILADOR%" Then

```

If Bandventilador = 1 Then
    Buffer = "*VENTILADOR ON*"
Else
    Buffer = "*VENTILADOR OFF*"
End If

Elseif Buffer = "%LUCES%" Then
    If Bandlucos = 1 Then
        Buffer = "****LUCES ON****"
    Else
        Buffer = "****LUCES OFF****"
    End If
End If

Elseif Buffer = "%SETUP%" Then
    Buffer = Str(setup)
End If

Elseif Buffer = "%ADC0%" Then
    A0 = Getadc(0)
    A00 = A0 * 100
    A00 = A00 / 1023
    Valor = Fusing(a00 , "#.#")
    Buffer = Valor
End If

Elseif Buffer = "%ADC1%" Then
    A1 = Getadc(1)
    A11 = A1 * 100
    A11 = A11 / 1023
    Valor = Fusing(a11 , "#.#")
    Buffer = Valor
End If

Elseif Buffer = "%ADC2%" Then
    A2 = Getadc(2)
    A2 = A2 / 4
    Valor = Str(a2)
    Valor = Format(valor , "000")
    Buffer = Valor
End If

```

```

Sendsize = Len(buffer)

Freesize:

Call Wiz5100_readvalue(&H4 , &H20)

Freesize = Wert

Shift , Freesize , Left , 8

Call Wiz5100_readvalue(&H4 , &H21)

Freesize = Freesize + Wert

If Freesize < Sendsize Then

    Goto Freesize

End If

Call Wiz5100_readvalue(&H4 , &H24)

Tx_wr = Wert

Shift , Tx_wr , Left , 8

Call Wiz5100_readvalue(&H4 , &H25)

Tx_wr = Tx_wr + Wert

Startpos = Tx_wr

Offset = Tx_wr And &H7FF

Startadress = &H4000 + Offset

Offsend = Offset + Sendsize

Pointer = Startadress

If Offsend > &H800 Then

    Uppersize = &H800 - Offset

    For X = 1 To Uppersize

        Highbyte = High(pointer)

        Lowbyte = Low(pointer)

        Tmp_str = Mid(buffer , X , 1)

        Tmp_value = Asc(tmp_str)

        Call Wiz5100_writevalue(highbyte , Lowbyte , Tmp_value)

        Incr Pointer

    Next X

    Pointer = &H4000

    Incr Uppersize

```

```

For X = Uppersize To Sendsize
    Highbyte = High(pointer)
    Lowbyte = Low(pointer)
    Tmp_str = Mid(buffer , X , 1)
    Tmp_value = Asc(tmp_str)
    Call Wiz5100_writevalue(highbyte , Lowbyte , Tmp_value)
    Incr Pointer
Next X
Else
For X = 1 To Sendsize
    Highbyte = High(pointer)
    Lowbyte = Low(pointer)
    Tmp_str = Mid(buffer , X , 1)
    Tmp_value = Asc(tmp_str)
    Call Wiz5100_writevalue(highbyte , Lowbyte , Tmp_value)
    Incr Pointer
Next X
End If
Call Wiz5100_readvalue(&H4 , &H24)
Startpos = Wert
Shift , Startpos , Left , 8
Call Wiz5100_readvalue(&H4 , &H25)
Startpos = Startpos + Wert
Glaenge = Startpos + Sendsize
'Send Site
Highbyte = High(glaenge)
Call Wiz5100_writevalue(&H4 , &H24 , Highbyte)
Lowbyte = Low(glaenge)
Call Wiz5100_writevalue(&H4 , &H25 , Lowbyte)
'Set SEND flag
Call Wiz5100_writevalue(&H4 , &H1 , &H20)
Loop

```

```

'Set DISCON flag

Call Wiz5100_writevalue(&H4 , &H1 , &H8)

End Sub

'Receive data

Sub Wiz5100_receive

Local I As Word

Local Top As Word

Local Bytes_rcv As Word

Local Headerend As String * 4

'Check for new incoming data

Call Wiz5100_readvalue(&H4 , &H26)

Bytes_rcv = Wert

Shift , Bytes_rcv , Left , 8

Call Wiz5100_readvalue(&H4 , &H27)

Bytes_rcv = Bytes_rcv + Wert

If Bytes_rcv > 0 Then

    Bytes_rcv = Bytes_rcv - 1

    For I = 0 To Bytes_rcv

        Top = I + &H6000

        Highbyte = High(top)

        Lowbyte = Low(top)

        Call Wiz5100_readvalue(highbyte , Lowbyte)

        'We only use correct header, correct header must be < 700 bytes!

        If Bytes_rcv < 700 Then

            Buffer = Buffer + Chr(wert)

        End If

    Next

    'Is header complete or is it a post reply?

    Headerend = Chr(13) + Chr(10) + Chr(13) + Chr(10)

    If Instr(buffer , Headerend) > 0 Then

        Website = 1

        #if Http_auth

```

```

If Instr(buffer , Tmp_auth) > 0 Then
    'Authentication Found And Correct
    Authentication = 0
'
If Instr(buffer , "POST /") = 1 Then
    If Instr(buffer , "?CALEFACTORON") > 0 Then
        Calefactor = 1
        Bandcalefactor = 1
        'Waitms 500
        'Relais1 = 0
    End If
    If Instr(buffer , "?CALEFACTOROFF") > 0 Then
        Calefactor = 0
        Bandcalefactor = 0
        'Waitms 500
        'Relais2 = 0
    End If
    If Instr(buffer , "?VENTILADORON") > 0 Then
        Ventilador = 1
        Bandventilador = 1
        'Waitms 500
        'Relais1 = 0
    End If
    If Instr(buffer , "?VENTILADOROFF") > 0 Then
        Ventilador = 0
        Bandventilador = 0
        'Waitms 500
        'Relais2 = 0
    End If
    If Instr(buffer , "?LUCESON") > 0 Then
        Luces = 1
        Bandluces = 1
        'Waitms 500

```



```

'Relais1 = 0

End If

If Instr(buffer , "?LUCESOFF") > 0 Then

    Luces = 0

    Bandluces = 0

    'Waitms 500

    'Relais2 = 0

End If

If Instr(buffer , "?ESTADOON") > 0 Then

    Manauto = 1

    Bandestado = 1

    'Waitms 500

    'Relais1 = 0

End If

If Instr(buffer , "?ESTADOOFF") > 0 Then

    Manauto = 0

    Bandestado = 0

    'Waitms 500

    'Relais2 = 0

End If

If Instr(buffer , "?SETUP+") > 0 Then

    Setup = Setup + 1

    If Setup >= Setmayor Then

        Setup = Setmayor

    End If

    Bandsetup = Bandsetup + 1

    If Bandsetup <= 3 Then

        Waitms 1000

    Else

        Bandsetup = 0

    End If

End If

```

```

If Instr(buffer , "?SETUP-") > 0 Then

    Setup = Setup - 1

    If Setup <= Setmenor Then

        Setup = Setmenor

    End If

    Bandsetup = Bandsetup + 1

    If Bandsetup <= 3 Then

        Waitms 1000

    Else

        Bandsetup = 0

    End If

End If

End If

End If

Else

    'Authentication not found or not correct

    ' Authentication = 1

    ' End If

#else

Authentication = 0

' If Instr(buffer , "POST /") = 1 Then

    If Instr(buffer , "?CALEFACTORON") > 0 Then

        Calefactor = 1

        Bandcalefactor = 1

        'Waitms 500

        'Relais1 = 0

    End If

    If Instr(buffer , "?CALEFACTOROFF") > 0 Then

        Calefactor = 0

        Bandcalefactor = 0

        'Waitms 500

        'Relais2 = 0

    End If

```

If Instr(buffer , "?VENTILADORON") > 0 Then

Ventilador = 1

Bandventilador = 1

'Waitms 500

'Relais1 = 0

End If

If Instr(buffer , "?VENTILADOROFF") > 0 Then

Ventilador = 0

Bandventilador = 0

'Waitms 500

'Relais2 = 0

End If

If Instr(buffer , "?LUCESON") > 0 Then

Luces = 1

Bandluces = 1

'Waitms 500

'Relais1 = 0

End If

If Instr(buffer , "?LUCESOFF") > 0 Then

Luces = 0

Bandluces = 0

'Waitms 500

'Relais2 = 0

End If

If Instr(buffer , "?ESTADOON") > 0 Then

Manauto = 1

Bandestado = 1

'Waitms 500

'Relais1 = 0

End If

```
If Instr(buffer , "?ESTADOOFF") > 0 Then
```

```
    Manauto = 0
```

```
    Bandestado = 0
```

```
    'Waitms 500
```

```
    'Relais2 = 0
```

```
End If
```

```
If Instr(buffer , "?SETUP+") > 0 Then
```

```
    Setup = Setup + 1
```

```
    If Setup >= Setmayor Then
```

```
        Setup = Setmayor
```

```
    End If
```

```
    Bandsetup = Bandsetup + 1
```

```
    If Bandsetup <= 3 Then
```

```
        Waitms 1000
```

```
    Else
```

```
        Bandsetup = 0
```

```
    End If
```

```
End If
```

```
If Instr(buffer , "?SETUP-") > 0 Then
```

```
    Setup = Setup - 1
```

```
    If Setup <= Setmenor Then
```

```
        Setup = Setmenor
```

```
    End If
```

```
    Bandsetup = Bandsetup + 1
```

```
    If Bandsetup <= 3 Then
```

```
        Waitms 1000
```

```
    Else
```

```
        Bandsetup = 0
```

```
    End If
```

```
End If
```

```
End If
```

```
#endif
```

```

'Set RECV flag

Call Wiz5100_writevalue(&H4 , &H1 , &H40)

Call Wiz5100_send()

Elseif Bytes_rcv > 700 Then

'Set RECV flag

Call Wiz5100_writevalue(&H4 , &H1 , &H40)

Website = 0

Call Wiz5100_send()

End If

End If

End Sub

Sub Wiz5100_init

Call Wiz5100_reset           'Hardware reset

'Register reset

Call Wiz5100_writevalue(&H0 , &H0 , &H80)

'Set gateway IP adress

Call Wiz5100_writevalue(&H0 , &H1 , Gw[1])

Call Wiz5100_writevalue(&H0 , &H2 , Gw[2])

Call Wiz5100_writevalue(&H0 , &H3 , Gw[3])

Call Wiz5100_writevalue(&H0 , &H4 , Gw[4])

'Set Subnetmask

Call Wiz5100_writevalue(&H0 , &H5 , Sn[1])

Call Wiz5100_writevalue(&H0 , &H6 , Sn[2])

Call Wiz5100_writevalue(&H0 , &H7 , Sn[3])

Call Wiz5100_writevalue(&H0 , &H8 , Sn[4])

'Set MAC

Call Wiz5100_writevalue(&H0 , &H9 , Mac[1])

Call Wiz5100_writevalue(&H0 , &HA , Mac[2])

Call Wiz5100_writevalue(&H0 , &HB , Mac[3])

Call Wiz5100_writevalue(&H0 , &HC , Mac[4])

Call Wiz5100_writevalue(&H0 , &HD , Mac[5])

Call Wiz5100_writevalue(&H0 , &HE , Mac[6])

```

```

'Set own IP adress

Call Wiz5100_writevalue(&H0 , &HF , Ip[1])

Call Wiz5100_writevalue(&H0 , &H10 , Ip[2])

Call Wiz5100_writevalue(&H0 , &H11 , Ip[3])

Call Wiz5100_writevalue(&H0 , &H12 , Ip[4])

'Initialize socket 0 with TCP on port 80

Call Wiz5100_writevalue(&H4 , &H0 , &H1 )      'TCP

Call Wiz5100_writevalue(&H4 , &H4 , &H0 )      'Port 80 HTTP

Call Wiz5100_writevalue(&H4 , &H5 , &H50 )

Call Wiz5100_disconnect()

End Sub

Sub Wiz5100_writevalue(byval Highbyte As Byte , Byval Lowbyte As Byte , Byval Value As Byte)

' Reset Wiz5100_cs

A = Wiz5100_opcode_write

Spiout A , 1

A = Highbyte

Spiout A , 1

A = Lowbyte

Spiout A , 1

Spiout Value , 1

' Set Wiz5100_cs

End Sub

Sub Wiz5100_readvalue(byval Highbyte As Byte , Byval Lowbyte As Byte)

' Reset Wiz5100_cs

A = Wiz5100_opcode_read

Spiout A , 1

A = Highbyte

Spiout A , 1

A = Lowbyte

Spiout A , 1

Spiin Wert , 1

' Set Wiz5100_cs

```


Data " </td><td> El Sensor de Humedad esta en = "

Data "%ADC1%"

Data " % :"

Data " </td><td> El Sensor de luminosidad esta en = "

Data "%ADC2%"

Data " lums :"

Data " </td></tr>"

Data "</tbody></table>"

Data "</center><center>
"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "MANUAL" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?ESTADOON'" , &H22 , ">"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "AUTO" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?ESTADOOFF'" , &H22 , ">"

Data "
"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "SETUP+" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?SETUP+'" , &H22 , ">"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "SETUP-" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?SETUP-'" , &H22 , ">"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "PARAR" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?PARAR'" , &H22 , ">"

Data "
"

Data "<center>"

Data "
<table border=" , &H22 , "5" , &H22 , ">"

Data "<tbody><tr>"

Data "<td><center>"

Data " Setup = :"

Data "%SETUP%"

Data " °C :"

Data "</center> </td></tr>"

Data "</tbody></table>"

Data "</center>
"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "CALEFACTOR_ON" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?CALEFACTORON'" , &H22 , ">"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "CALEFACTOR_OFF" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?CALEFACTOROFF'" , &H22 , ">"

Data "
"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "VENTILADOR_ON" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?VENTILADORON'" , &H22 , ">"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "VENTILADOR_OFF" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?VENTILADOROFF'" , &H22 , ">"

Data "
"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "LUCES_ON" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?LUCESON'" , &H22 , ">"

Data "<input type=" , &H22 , "submit" , &H22 , " value=" , &H22 , "LUCES_OFF" , &H22 , " style=" , &H22 , "width:135px;height:35px" , &H22 , " onclick=" , &H22 , "location.href='./?LUCESOFF'" , &H22 , ">"

Data "
"

Data "</center><center>"

Data "
<table border=" , &H22 , "5" , &H22 , ">"

Data "<tbody><tr>"

Data "<td><center>"

Data "%ESTADO%"

Data "</center></td>"

Data "<td><center>"

Data "%CALEFACTOR%"

Data "</center></td></tr>"

Data "<td><center>"

Data "%VENTILADOR%"

Data "</center></td><td><center>"

Data "%LUCES%"

Data "</center></td></tr>"

Data "</tbody></table>"

Data "</center>"

Data "<iframe style=" , &H22 , "display: none; visibility: hidden;" , &H22 , "></iframe><script src=" , &H22 , "./UNDCHI2_files/pops" , &H22 , "></script><script src=" , &H22 , "./UNDCHI2_files/p" , &H22 , "></script></body></html>"

Data "%END%"