



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 GRADUACIÓN

**“DISEÑO Y CONSTRUCCIÓN DE UN MICRO WEB SERVER PARA
CONTROL DE SEÑALES ANALÓGICAS Y DIGITALES”**

Autores: Luis Javier Flores Villagómez

Diego Fernando Villacrés Arias

Director: Ing. Daniel Santillán

Riobamba-Ecuador

AÑO

2015

Los miembros del Tribunal de Graduación del proyecto de investigación de título: “DISEÑO Y CONSTRUCCIÓN DE UN MICRO WEB SERVER PARA CONTROL DE SEÑALES ANALÓGICAS Y DIGITALES” presentado por: Luis Javier Flores Villagómez y Diego Fernando Villacrés Arias dirigida por: Daniel Santillán Haro.

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

Nota



Firma

Ing. Geovanny Cuzco

Miembro del Tribunal

Nota



Firma

Ing. Daniel Santillán

Director del Proyecto

Nota



Firma

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Luis Javier Flores Villagómez, Diego Fernando Villacrés Arias y Daniel Santillán Haro; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo



Luis Javier Flores Villagomez

C. I. 0603416512



Diego Fernando Villacres Arias

C.I. 0603415076

AGRADECIMIENTO

Agradezco a mis padres por haberme dado la magnífica oportunidad de realizar estudios superiores.

Un agradecimiento especial a mi esposa María José quien a lo largo de todos estos años de Universidad ha sido mi apoyo y fortaleza.

Mi agradecimiento a los Docentes de La Universidad Nacional de Chimborazo por haber proporcionado los conocimientos que me permiten desenvolverse en la vida profesional.

Diego

DEDICATORIA

A mi esposa María José y mi hija Sofía

las personas que me han impulsado

para conseguir este logro

Diego

ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	iv
ÍNDICE DE GRÁFICOS.....	v
ÍNDICE DE TABLAS.....	v
ÍNDICE DE ANEXOS.....	v
RESUMEN.....	vi
SUMARY.....	vii
INTRODUCCIÓN.....	1-
CAPÍTULO I.....	- 2 -
1 MARCO REFERENCIAL.....	- 2 -
1.1 PLANTEAMIENTO DEL PROBLEMA	- 2 -
1.2 FORMULACIÓN DEL PROBLEMA	- 2 -
1.3 OBJETIVOS.....	- 3 -
1.3.1 OBJETIVO GENERAL.....	- 3 -
1.3.2 OBJETIVOS ESPECÍFICOS.....	- 3 -
1.4 JUSTIFICACIÓN.....	- 4 -
CAPÍTULO II.....	- 5 -
2 FUNDAMENTACIÓN TEÓRICA.....	- 5 -
2.1 ANTECEDENTES.....	- 5 -
2.2 SISTEMAS EMBEBIDOS	- 6 -
2.2.1 CARACTERÍSTICAS DEL SISTEMA EMBEDIDO.....	- 6 -
2.2.2 MEDIOS DE COMUNICACIÓN DE SISTEMAS EMBEBIDOS.....	- 7 -
2.2.3 APLICACIONES DEL SISTEMA EMBEBIDO	- 8 -
2.3 LA FAMILIA DE PROTOCOLOS TCP/IP	- 8 -
2.3.1 LOS PROTOCOLOS DE LA CAPA TRANSPORTE TCP Y UDP	- 9 -
2.3.2 PROTOCOLO UDP	- 12 -
2.3.3 PROTOCOLO DE LA CAPA INTERNET ICMP.....	- 12 -
2.3.4 PROTOCOLO ARP RESOLUCIÓN DE DIRECCIONES IP A DIRECCIONES FÍSICAS.....	- 14 -
2.3.5 PROTOCOLO DE MULTIDIFUSIÓN IGMP.....	- 15 -
2.4 TECNOLOGÍAS LAN	- 15 -
2.4.1 ARQUITECTURA LAN.....	- 15 -
2.5 TOPOLOGÍAS.....	- 17 -
2.5.1 TOPOLOGÍA EN BUS.....	- 17 -
2.5.2 TOPOLOGÍA EN ESTRELLA	- 18 -

2.6 MICRO-CONTROLADORES	- 21 -
2.6.1 APLICACIONES DE LOS MICRO-CONTROLADORES	- 21 -
2.6.2 ARQUITECTURA DE PROCESO	- 21 -
2.6.3 EL PROCESADOR CPU	- 22 -
2.6.4 MEMORIA	- 22 -
2.6.5 LINEAS DE ENTRADA Y SALIDA (E/S)	- 23 -
2.6.6 RELOJ PRINCIPAL	- 23 -
2.6.7 RECURSOS ESPECIALES.....	- 23 -
2.7 TARJETAS CONTROLADORAS DE ETHERNET	- 24 -
2.7.1 ARDUINO ETHERNET SHIELD	- 24 -
2.8 COMUNICACIÓN SPI	- 26 -
2.8.1 ESPECIFICACIONES DEL BUS	- 26 -
2.8.2 SEÑALES ANALÓGICAS Y DIGITALES	- 27 -
2.10 LENGUAJE C ORIENTADO A MICRO-CONTROLADORES	- 28 -
2.10.1 LENGUAJES DE BAJO Y ALTO NIVEL.....	- 28 -
2.8.3 LENGUAJE C.....	- 29 -
2.9 LENGUAJE HTML	- 34 -
2.9.1 ESTRUCTURA BÁSICA HTML	- 35 -
2.9.2 CONFIGURACIÓN DE TEXTO	- 36 -
2.9.3 ENLACES	- 36 -
2.9.4 IMÁGENES.....	- 37 -
2.9.5 PRUEBA T DE STUDENT DE DOS MUESTRAS RELACIONADAS.....	- 38 -
CAPÍTULO III	- 39 -
3 METODOLOGÍA	- 39 -
3.1 TIPO DE ESTUDIO	- 39 -
3.2 POBLACIÓN Y MUESTRA	- 39 -
3.3 PROCEDIMIENTOS	- 40 -
3.3.1 REQUERIMIENTOS DEL SISTEMA	- 40 -
3.3.2 SELECCIÓN DE TARJETA ETHERNET Y MICRO-CONTROLADOR	- 41 -
3.3.3 MICRO-CONTROLADOR.....	- 42 -
3.3.4 CONSTRUCCIÓN DE HARDWARE	- 42 -
3.3.5 DISEÑO E IMPLEMENTACIÓN DE SOFTWARE.....	- 46 -
3.3.6 DISEÑO DE LA PÁGINA WEB	- 51 -
3.4 PROCESAMIENTO Y ANALISIS	- 52 -

3.5 COMPARACION ENTRE SERVIDORES EMBEBIDOS	- 52 -
CAPÍTULO IV	- 55 -
4 RESULTADOS	- 55 -
4.1 RESULTADOS OBTENIDOS.....	- 55 -
CAPÍTULO V.....	- 59 -
5 DISCUSIÓN	- 59 -
CAPÍTULO VI.....	- 60 -
6 CONCLUSIONES Y RECOMENDACIONES	- 60 -
6.1 CONCLUSIONES.....	- 60 -
6.2 RECOMENDACIONES.....	- 61 -
CAPÍTULO VII.....	- 62 -
7 PROPUESTA	- 62 -
7.1 TÍTULO DE LA PROPUESTA	- 62 -
7.2 INTRODUCCIÓN	- 62 -
7.3 OBJETIVOS.....	- 62 -
7.3.1 OBJETIVO GENERAL.....	- 62 -
7.3.2 OBJETIVOS ESPECÍFICOS.....	- 62 -
7.4 FUNDAMENTACIÓN CIENTÍFICO – TÉCNICA	- 63 -
7.5 DESCRIPCIÓN DE LA PROPUESTA	- 63 -
7.6 DISEÑO ORGANIZACIONAL	- 65 -
7.7 MONITOREO Y EVALUACIÓN DE LA PROPUESTA	- 66 -
CAPÍTULO VIII.....	- 67 -
8 BIBLIOGRAFÍA	- 67 -

ÍNDICE DE FIGURAS

FIGURA No. 1. Implementación de Protocolos TCP/IP en Sistemas Finales. Fuente: (Stallings, 2004)	- 9 -
FIGURA No. 2. Transferencia de Datos Usando Protocolos TCP. Fuente: (Comer, 1996).....	- 10 -
FIGURA No. 3. Funcionamiento de Protocolo TCP. Fuente: (Comer, 1996)	- 11 -
FIGURA No. 4. Puertos Asignados	- 12 -
FIGURA No. 5 Comando PING	- 13 -
FIGURA No. 6 Funcionamiento del Protocolo ARP. Fuente: (Comer, 1996).....	- 14 -
FIGURA No. 7 Arquitectura de una LAN. Fuente: (Stallings, 2004).....	- 16 -
FIGURA No. 8 Topología en Bus. Fuente: (Stallings, 2004)	- 18 -
FIGURA No. 9 Topología en Estrella. Fuente: (Stallings, 2004)	- 19 -
FIGURA No. 10. Arduino Ethernet Shield. Fuente: (Arduino, Arduino Ethernet Shield, 2015)-	25 -
FIGURA No. 11 Chip Wiznet W5100. Fuente: (WIZnet Co, 2008).....	- 25 -
FIGURA No. 12 Comunicación SPI. Fuente: (López Pérez, S. F.)	- 26 -
FIGURA No. 13. Señal Analógica en un Intervalo de Tiempo t. Fuente: (Castro, Lechtaler, y Fusario, 1999)	- 27 -
FIGURA No. 14 Señal Analógica en un Intervalo de Tiempo t. Fuente: (Castro, Lechtaler, y Fusario, 1999)	- 27 -
FIGURA No. 15 Niveles de Programación. Fuente: (Abel, 1996).....	- 29 -
FIGURA No. 16. Diagrama De Bloques MICRO WEB SERVER.....	- 41 -
FIGURA No. 17 Pines SPI en el Atmega 328P	- 42 -
FIGURA No. 18 ATMEGA 328P PINES de Alimentación	- 43 -
FIGURA No. 19 Fuente Y Oscilador ATMEGA328P.....	- 43 -
FIGURA No. 20 SPI Micro-Controlador y Arduino Ethernet Shield	- 44 -
FIGURA No. 21. Lectura de Señales Analógicas	- 44 -
FIGURA No. 22 Monitoreo y Control de Señales Digitales	- 45 -
FIGURA NO. 23. Pantalla de Inicio AVR STUDIO 6	- 46 -
FIGURA No. 24. Pantalla de Elección Lenguaje de Programación, Tipo de Proyecto, Nombre y Ubicación	- 47 -
FIGURA NO. 25 Pantalla de Elección del Micro-Controlador.....	- 47 -
FIGURA NO. 26 Plantilla de Programación AVR STUDIO	- 48 -
FIGURA No. 27 Modificar Reloj.....	- 49 -
FIGURA No. 28 Página Web.....	- 51 -
FIGURA No. 29 Arduino Uno	- 53 -
FIGURA No. 30 ENC28J60.....	- 53 -

ÍNDICE DE GRÁFICOS

GRÁFICO No. 1 Tiempos de Carga del Dispositivo Comercial Vs. Micro Web Server.....	- 58 -
GRÁFICO No. 2 Conectividad del Dispositivo Comercial Vs. Micro Web Server.....	- 58 -

ÍNDICE DE TABLAS

TABLA No. 1 Algunos Tipos de Datos	- 30 -
TABLA No. 2 Dispositivos DisponiblePara Selección.....	- 41 -
TABLA No. 3 Pines para SPI	- 44 -
TABLA No. 4 Características de lasTarjetas Empleadas.....	- 53 -
TABLA No. 5. Esquema para Obtención de Datos. Data Center Uniandes Enero 2015	- 54 -
TABLA No 6 Datos Micro Web Server vs. Dispositivo Comercial	- 56 -

ÍNDICE DE ANEXOS

ANEXO No. 1 Atmega328P-Diagrama de Bloques	- 71 -
ANEXO No. 2 SPI (Serial Peripheral Interface) - Diagrama De Bloques	- 72 -
ANEXO No. 3 Oscilador Externo	- 72 -
ANEXO No. 4 Pines Y Condiciones De Trabajo.....	- 73 -
ANEXO No. 5 Configuración Típica.....	- 73 -
ANEXO No. 6 Arduino Ethernet Shield Diagrama Esquemático	- 74 -
ANEXO No. 7 Micro Web Server Diagrama Esquemático	- 75 -
ANEXO No. 8 Programa Micro Web Server.....	- 76 -
ANEXO No. 9 Programa Página Web	- 81 -

RESUMEN

El presente proyecto describe el diseño y la construcción de un MICRO WEB SERVER capaz de controlar señales analógicas y digitales. Las especificaciones consideradas para diseño fueron: interfaz de página web, accesible mediante dirección IP y puerto en una red LAN cableada, funcional para monitoreo y control de señales analógicas y digitales, eficiente y de bajo costo.

El sistema consta de un módulo para adquisición de datos de los sensores, una tarjeta externa para conectividad con la red LAN y la interfaz web que se carga en el micro-controlador. Las señales analógicas se toman de potenciómetros y las digitales de pulsadores y diodos emisores de luz.

Se utilizó un micro-controlador ATMEGA 328P programado en lenguaje C como servidor web, para proporcionar conectividad con la red LAN se añade el módulo ARDUINO ETHERNET SHIELD. La página web se programó en lenguaje HTML utilizando el bloc de notas de Windows, para acceder al servidor se ingresa la dirección IP asignada al dispositivo y el puerto en cualquier navegador de internet.

Para comprobar el funcionamiento del dispositivo fue comparado con un modelo comercial basado en la tarjeta de desarrollo ARDUINO UNO como servidor web y la tarjeta controladora externa ENC28j60 para conexión en la red. Los parámetros considerados fueron el tiempo de carga de la página web y la conectividad en la red utilizando el comando ping, pruebas realizadas en una red cableada.

Aplicando el método de “t de student” los resultados que se procesaron indicaron poca diferencia significativa entre los dos dispositivos puestos a prueba, demostrando la eficiencia del servidor web construido.



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERIA
CENTRO DE IDIOMAS



Lic. Maritza Larrea

19 de Mayo del 2015

SUMMARY

This project describes the design and construction of a MICRO WEB SERVER, this is able to monitor and control analog and digital signals. The specifications designs such as website interface, accessible via IP address and port in a functional monitoring and control for analog and digital, efficient and inexpensive wired LAN signals were considered to this work.

The system consists of a module for data acquisition sensors, an external card for connectivity to the LAN and the web interface is loaded into the microcontroller. Analog signals are taken from the digital potentiometers and switches and light emitting diodes.

A microcontroller ATMEGA 328P programmed in C language as a web server to provide connectivity with the ARDUINO ETHERNET LAN module is added SHIELD was used. The website was programmed in HTML using notepad of Windows, to access the server's IP address assigned to the device and port on any Internet browser is entered.

To check the functioning of the device, it was compared with a business model based on the development board ARDUINO UNO as web server and ENC28j60 external controller card for network connection. The parameters considered were the loading time of the website and network connectivity using ping, tests on a wired network.

Applying the method of "t student" the results were processed showing little significant difference between the two devices tested, demonstrating the efficiency of the built web server.

José María Acosta



INTRODUCCIÓN

En la actualidad las redes de computadoras están al alcance de cualquier persona prácticamente en todo lugar, una de las aplicaciones más comunes es la comunicación. Estas características pueden ser utilizadas para desarrollar sistemas electrónicos capaces de administrar procesos remotamente y mostrar los resultados en navegadores web familiares a los usuarios.

Los sistemas embebidos permiten desarrollar dispositivos con capacidades de conexión a internet. El componente fundamental es el micro-controlador siendo los fabricados por ATMEL de gran utilidad por su versatilidad en cuanto a aplicaciones.

El problema de utilizar un servidor convencional para tareas específicas se puede describir de la siguiente manera: el costo del hardware y la programación son elevados, generalmente un servidor convencional realiza más de una tarea. Los servidores embebidos proponen una alternativa de bajo costo y específica para un proceso.

El proyecto inicia con la elección de los componentes electrónicos más idóneos y la construcción del hardware, el proceso continúa con la programación del prototipo y las primeras pruebas de conectividad en la red.

Los valores de los sensores se muestran a través de una interfaz de página web cargada en el servidor, el acceso a este recurso se hace con un navegador de internet ingresando la dirección IP y el puerto correspondiente, parámetros definidos en la programación del dispositivo.

CAPÍTULO I

1 MARCO REFERENCIAL

1.1 PLANTEAMIENTO DEL PROBLEMA

El alcance del internet actualmente está presente prácticamente en todo lugar, permitiendo acceder a información desde cualquier parte del mundo, el acceso a este servicio no solo se limita a dispositivos fijos, sino también a dispositivos móviles con velocidades de navegación altas.

Los servidores son el soporte de los múltiples servicios de internet, normalmente estos son computadoras especializadas con muy altas capacidades de procesamiento, entre los más utilizados destacan servidores de correo, de base de datos, web, de almacenamiento de información y de telefonía.

Entre los micro-controladores son varios los dispositivos con las características adicionales de incorporar soporte para Ethernet, protocolos como TCP, UDP, comunicación SPI entre otras; permitiendo así desarrollar micro-servidores para monitorear y controlar procesos, utilizando como medio de comunicación el internet.

1.2 FORMULACIÓN DEL PROBLEMA

En los servidores el hardware es más preciso y soporta tareas más complejas, el software está enfocado en proporcionar uno o varios servicios. Gracias a estas prestaciones esta clase de equipos pueden utilizarse para alojar páginas web, servicios de correo electrónico, etc. Los servidores dedicados si bien ejecutan una aplicación o programa concreto, son implementados de igual manera con hardware avanzado pensados en usuarios que manejen grandes cantidades de información y recursos económicos.

Escasamente se implementan mini servidores capaces de prestar servicios para monitoreo de algún proceso específico, es también escaso servicios que adicionalmente permitan controlar el proceso monitoreado.

El desarrollo de un micro servidor que permita monitorear y controlar algún tipo de señal utilizando una página web como medio de visualización para el usuario, necesita la construcción de un hardware y la programación de un software basado en micro-controladores que cumplan tres funciones.

- a) Adquirir los datos que serán mostrados en la página web
- b) Capacidad para almacenar la página web con los datos tomados
- c) Comunicarse con otro equipo a través de internet.

Además se debe incluir un equipamiento adicional que será la tarjeta de red del equipo.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar y construir un micro web server para controlar señales analógicas y digitales.

1.3.2 OBJETIVOS ESPECÍFICOS

- Elegir el micro-controlador más adecuado y el modulo para la interfaz de red del micro-servidor.
- Diseñar y construir el circuito para el micro-controlador, el módulo par la interfaz de red y los sensores.
- Determinar el entorno de programación y programar el micro-controlador.
- Elaborar la página web y cargar en el micro-servidor
- Comprobar conectividad del dispositivo.

1.4 JUSTIFICACIÓN

La implementación de sistemas remotos de monitoreo y control de procesos: industriales, sistemas de vigilancia y seguridad, adquisición de datos, son muy útiles, ya que disminuye costos de movilización y transporte, ahorro de tiempo y utilizan tecnología familiar al usuario. Por lo que es necesario el diseño y construcción de un micro web server para el control de señales analógicas y digitales.

La adquisición de un servidor que cumpla una función específica constituye una inversión costosa y desperdicia recursos. Generalmente estos equipos proporcionan varios servicios en la red y se constituyen de componentes de hardware avanzados, pensados precisamente en ser utilizados en múltiples aplicaciones.

La propuesta planteada de este proyecto de investigación es dar una solución a este problema generalizado de los servidores convencionales. Con un micro-servidor diseñado para realizar tareas específicas, el hardware que soporta el dispositivo se basa en componentes electrónicos fiables y de bajo costo. Construyendo un dispositivo con similar funcionalidad y bajo coste en relación a un servidor convencional.

CAPÍTULO II

2 FUNDAMENTACIÓN TEÓRICA

2.1 ANTECEDENTES

La tecnología Ethernet es la predominante en ámbitos de oficina, comercio, doméstico e industrial, gracias a las ventajas que ofrece:

- Acceso remoto fiable.
- Ethernet se basa en IEEE 802.3 estándar que garantiza compatibilidad entre dispositivos de distintos fabricantes.
- Ethernet facilita la escalabilidad (Comer, 1996).

Los módulos con soporte para IP permiten la utilización de la red Ethernet para realizar tareas de monitorización y control.

Los circuitos integrados que soportan Ethernet los hay de variedad de fabricantes Microchip, Texas Instruments, Atmel por nombrar algunos, básicamente tiene las siguientes características:

- Estándar IEEE 802.3.
- Compatibilidad con 10/100 Base-T.
- Interfaz PHY y dirección MAC.
- Soporte para los modos half y full dúplex.
- Interfaz SPI.
- Soporte TCP/IP. (WIZnet Co, 2008)

2.2 SISTEMAS EMBEBIDOS

Un sistema embebido procesa información para aplicaciones específicas, forma parte de otro sistema más grande, además tiene componentes tales como hardware y software (González Palacio Liliana y Urrego Giraldo Germán, 2008).

2.2.1 CARACTERÍSTICAS DEL SISTEMA EMBEDIDO

La característica más importante de un sistema embebido es la interacción con el mundo exterior basada en el tiempo o en algún estímulo, para que la interacción sea correcta estos sistemas deben tener características de fiabilidad, seguridad y disponibilidad (González Palacio Liliana y Urrego Giraldo Germán, 2008).

Otras características de este sistema son:

- Tamaño.
- Consumo de energía.
- Memoria.
- Portabilidad del sistema.
- Transparentes al usuario, gracias a esta característica el sistema puede funcionar por largos periodos de tiempo.
- El sistema embebido puede cambiar su comportamiento, por la presencia de un determinado evento inesperado.
- Este sistema puede interactuar con otros, formando parte de una estructura más compleja.

Esta clase de sistemas electrónicos permiten añadir módulos para almacenamiento de información, comunicación, visualización entre otros con la finalidad de mejorar el desempeño.

2.2.2 MEDIOS DE COMUNICACIÓN DE SISTEMAS EMBEBIDOS

Los sistemas embebidos presentan una gran posibilidades de comunicación, tal es el caso de:

- USB

Proporciona comunicación fácil y rápida, utilizable para alimentar directamente por cable los dispositivos o para añadir periféricos esclavos.

- Bluetooth

El sistema embebido adquiere capacidades de comunicación inalámbrica, el alcance estándar es de 10 metros.

- Ethernet

La capacidad de un sistema embebido para adquirir conectividad en internet la proporcionan módulos similares a las tarjetas de red convencionales, estos módulos se conectan directamente al microprocesador de la placa.

- GPS

La implementación de módulos GPS (Global Positioning System) transfieren información al sistema referente a posicionamiento, velocidad, tiempo, fecha.

- Puertos digitales

Las entradas/salidas (E/S) se manejan a nivel de bit 0 o 1.

- Puertos analógicos

Estos puertos permiten adquirir datos referentes a temperatura, presión, etc. Se necesita de sensores para leer la información (Sánchez Vitores, 2004).

2.2.3 APLICACIONES DEL SISTEMA EMBEBIDO

Las aplicaciones de los sistemas embebidos son bastante amplias entre las que destacan:

- **Robótica**
Implementa sistemas para la automatización de todo tipo de procesos en especial de alta complejidad
- **Domótica**
El desarrollo de módulos permite aplicaciones para control de edificios, casas, sistemas de vigilancia monitoreados local o remotamente (Sánchez Vitores, 2004).

2.3 LA FAMILIA DE PROTOCOLOS TCP/IP

TCP/IP es la arquitectura más usada para la interconexión de sistemas, consiste en una extensa colección de protocolos entre los más conocidos HTTP, FTP, SMTP, POP, etc. Todos estos se han convertido en los estándares de internet (Stallings, 2004).

No existe un modelo oficial de referencia TCP/IP, pero basándose en los protocolos estándar que han sido implementados esta arquitectura se puede organizar en 5 capas relativamente independientes.

- **Capa Física.-** La capa física especifica las características del medio de transmisión, velocidad de datos y demás temas afines a la interfaz física entre el computador y la red.
- **Capa de Acceso.-** La capa de acceso a la red tiene como finalidad intercambiar datos entre el sistema final y la red a la cual está conectado, es decir esta capa maneja el acceso y encaminamiento de datos dentro de la red local.

- **Capa de Internet.-** La capa de internet permite que los datos atraviesen distintas redes interconectadas, el protocolo internet (IP, Internet Protocol) se implementa en esta capa, este protocolo está presente tanto en dispositivos finales como intermedios.
- **Capa de Transporte.-** La capa de transporte garantiza una transmisión segura, esto es que los datos lleguen a la aplicación destino y en el mismo orden que fueron enviados, el protocolo TCP (Transmission Control Protocol) es el más usado en esta capa.
- **Capa de Aplicación.-** Finalmente la capa de aplicación se encarga de gestionar todas las aplicaciones de usuario (Stallings, 2004).

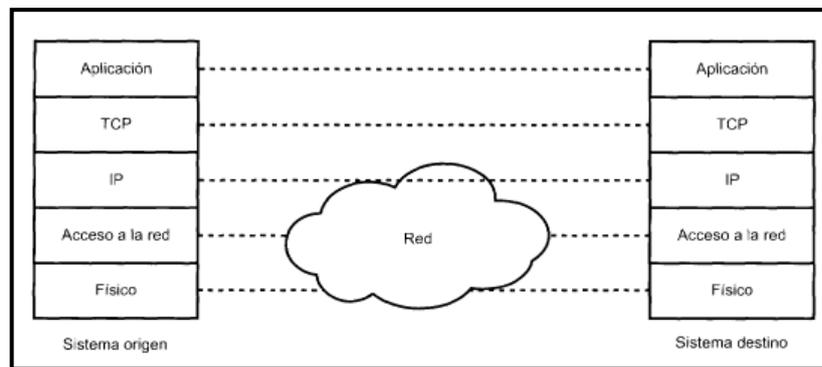


FIGURA No. 1. IMPLEMENTACIÓN DE PROTOCOLOS TCP/IP EN SISTEMAS FINALES.

Fuente: (Stallings, 2004)

La capa física y de acceso a la red indican interacción con la red, mientras tanto la capa aplicación y la capa de transporte implementan los protocolos que facilitan la comunicación entre los sistemas finales.

La capa de internet realiza procesos para el encaminamiento en la red pero también proporciona información para los sistemas finales.

2.3.1 LOS PROTOCOLOS DE LA CAPA TRANSPORTE TCP Y UDP

Dentro de la capa transporte tenemos protocolos orientados a la conexión, como es TCP, o no orientado a la conexión, como es UDP estos protocolos proporcionan un servicio de transferencia de datos extremo a extremo, los

servicios que utilizan estos protocolos no se enteran de los detalles de los sistemas de comunicaciones subyacentes (Comer, 1996).

2.3.1.1 Servicio orientado a la conexión TCP

Para proporcionar confiabilidad estos protocolos implementan una técnica conocida como acuse de recibo, este procedimiento hace que el receptor se comunique con el origen y le envíe un mensaje de acuse de recibo ACK conforme llegan los datos, el receptor espera la llegada del ACK antes de enviar otro paquete (Comer, 1996).

En la figura 2 se muestra como transfiere datos el protocolo más sencillo que utiliza ACK, a la derecha se muestra que pasa en el emisor y a la izquierda el receptor, la distancia vertical indica el tiempo y las líneas cruzadas paquetes en la red.

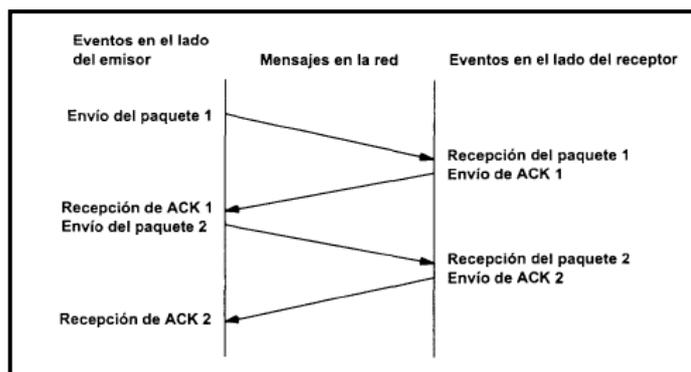


FIGURA No. 2. TRANSFERENCIA DE DATOS USANDO PROTOCOLOS TCP. Fuente: (Comer, 1996)

En casos donde se pierde o corrompe un paquete el sistema transmisor genera un temporizador, al expirar automáticamente reenvía el paquete.

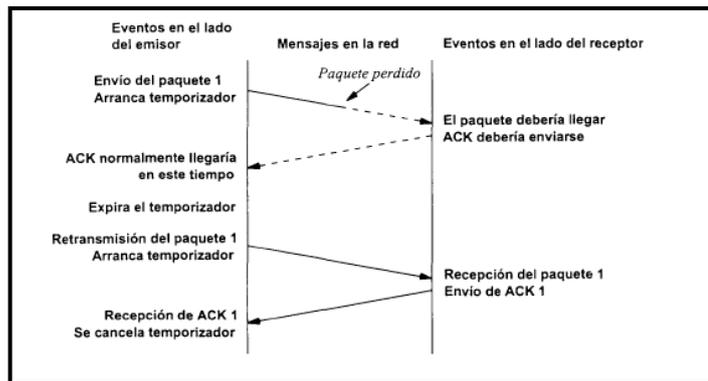


FIGURA No. 3. FUNCIONAMIENTO DE PROTOCOLO TCP. Fuente: (Comer, 1996)

Para casos de duplicación de paquetes los protocolos confiables detectan los duplicados a través de un número de secuencia asignado a cada paquete. En la figura 3 se ejemplifica un diagrama similar, que muestra el funcionamiento del protocolo al perderse o corromperse un paquete.

TCP define que formato de datos y ACK intercambian dos computadoras para tener una transferencia confiable, también define los procedimientos que la computadora utiliza para que los datos lleguen de manera correcta.

2.3.1.2 Puertos de protocolo TCP

TCP utiliza números de puerto de protocolo que identifican el destino final dentro de una máquina, estos puertos se identifican con un número entero.

Cuando el protocolo TCP establece una conexión entre dos máquinas se crea un circuito virtual en que cada terminal se conoce como punto extremo, TCP identifica a cada punto extremo mediante la IP y el puerto por ejemplo 192.168.1.100,25 se refiere al puerto TCP 25 en la maquina con dirección IP 192.168.1.100 (Comer, 1996).

Gracias a que TCP reconoce una conexión como un par de puntos extremos, varias conexiones en la misma maquina pueden compartir un numero de puerto TCP. Los puertos se pueden asignar de manera estática o dinámica dependiendo si se requieren para aplicaciones comunes como correo electrónico o dependiendo

de los programas que se necesiten. En la figura 4 se listan algunos de los puertos asignados.

PUERTO	DESCRIPCIÓN
0	Reservado
7	Echo
15	Estado de red netstat
21	Protocolo de transferencia de archivos FTP
23	Conexión de terminal TELNET

FIGURA No. 4. PUERTOS ASIGNADOS

2.3.2 PROTOCOLO UDP

La pila de protocolos TCP/IP dispone del protocolo de datagrama de usuario o UDP para el envío de paquetes conocidos como datagramas.

El datagrama es enviado al receptor correcto gracias a que en el paquete enviado se incluye tanto el puerto de origen como el puerto de destino. Estos se pueden duplicar, perder o llegar sin orden porque no se implementa ningún mecanismo de control de paquetes, no establece ninguna conexión con el destino, es decir es un servicio sin conexión y no confiable (Comer, 1996).

Para que un datagrama pueda ser enviado, la aplicación que solicita este servicio debe negociar la asignación de un puerto una vez superado este procedimiento los paquetes son enviados. La asignación de puertos se hace de manera que hay algunos pres asignados, dejando una buena cantidad libre.

2.3.3 PROTOCOLO DE LA CAPA INTERNET ICMP

Dentro de la pila TCP/ IP existen protocolos dedicados a la detección de errores en la red, estas implementaciones están presentes como parte de los datagramas.

El protocolo ICMP permite que ruteadores envíen mensajes de error o de control hacia otros ruteadores o anfitriones, el destino definitivo de un mensaje de estas características no es una aplicación o usuario es el software de protocolo de internet en esa máquina (Comer, 1996).

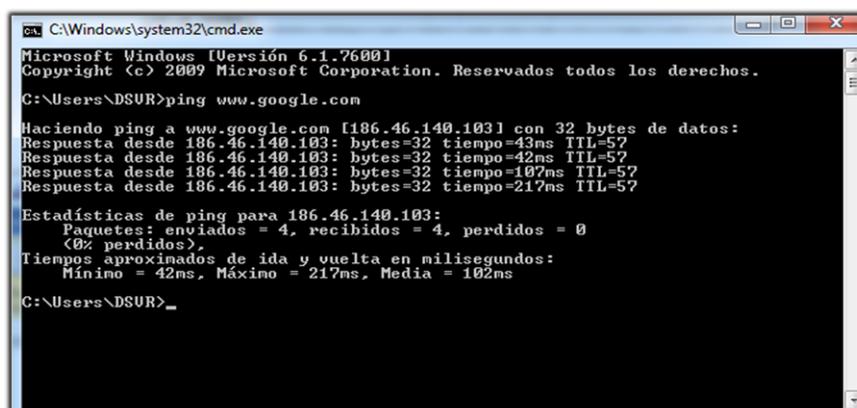
Similarmente el emisor de este tipo de mensajes lo hace desde el software de protocolo de internet, este protocolo también puede usarse para comunicación entre anfitriones. El ICMP solo reporta el error a la fuente de dicho problema, para corregir la dificultad presentada se deben realizar otras acciones.

2.3.3.1 Accesibilidad y estado de un destino

Una de las herramientas más utilizadas para gestión de errores son los mensajes ICMP de echo request (solicitud de eco) y echo reply (respuesta de eco).

El funcionamiento consiste en enviar desde un usuario o ruteador una solicitud de eco hacia un destino específico, este formula una respuesta y la devuelve, la solicitud de eco y respuesta posibilitan determinar si un destino es alcanzable y responde (Comer, 1996).

Gracias a que estas solicitudes viajan en datagramas, se puede comprobar que los componentes principales de transporte funcionan correctamente. En varios sistemas el comando que ejecuta las solicitudes de eco se conoce como ping, en la figura 5 se ejecuta el comando.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\DSUR>ping www.google.com

Haciendo ping a www.google.com [186.46.140.103] con 32 bytes de datos:
Respuesta desde 186.46.140.103: bytes=32 tiempo=43ms TTL=57
Respuesta desde 186.46.140.103: bytes=32 tiempo=42ms TTL=57
Respuesta desde 186.46.140.103: bytes=32 tiempo=107ms TTL=57
Respuesta desde 186.46.140.103: bytes=32 tiempo=217ms TTL=57

Estadísticas de ping para 186.46.140.103:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 42ms, Máximo = 217ms, Media = 102ms

C:\Users\DSUR>
```

FIGURA No. 5 COMANDO PING

2.3.4 PROTOCOLO ARP RESOLUCIÓN DE DIRECCIONES IP A DIRECCIONES FÍSICAS.

Este protocolo implementa procesos para transformar una dirección IP en una dirección física conocida como dirección MAC, dos máquinas en una red se pueden comunicar solamente si se realiza esta transformación, este protocolo se implementa en la capa de acceso a la red (Comer, 1996).

La transformación de direcciones se realiza a lo largo de todo el camino recorrido, un transmisor tiene que traducir direcciones tanto para el destino final como para dispositivos intermedios.

2.3.4.1 Funcionamiento del protocolo ARP

Emplea el servicio de broadcast o difusión de las redes Ethernet, en la figura 6 cuando el anfitrión A requiere la dirección física de B, envía un paquete especial por difusión solicitando la información física de B.

Todos los anfitriones reciben este paquete pero solamente A reconoce su dirección IP y responde a la petición, cuando B recibe la respuesta envía el paquete a su destinatario.

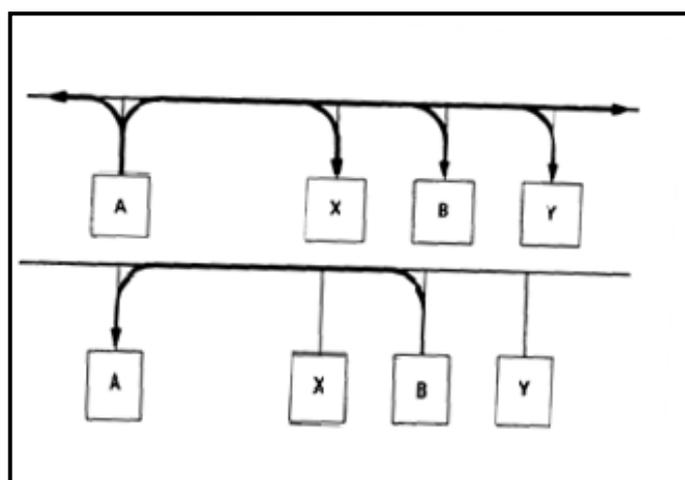


FIGURA No. 6 FUNCIONAMIENTO DEL PROTOCOLO ARP. Fuente: (Comer, 1996)

2.3.5 PROTOCOLO DE MULTIDIFUSIÓN IGMP

Este servicio permite la transmisión de un datagrama a un conjunto de anfitriones que forman un solo grupo de multidifusión. La multidifusión hace que un paquete sea entregado a varios destinos, el concepto es similar al de difusión pero con la variante de que en multidifusión cada máquina decide si quiere participar en ella.

Los datagramas de multidifusión emplean el concepto de entrega con el mejor esfuerzo, significando que pueden perderse, duplicarse, borrarse o llegar sin orden. Cada grupo de multidifusión tiene una dirección de multidifusión específica, algunas de estas son asignadas permanentemente y otras para uso temporal, IGMP es parte de la capa de internet (Comer, 1996).

2.4 TECNOLOGÍAS LAN

Las redes LAN consisten en un medio de transmisión compartido, software y hardware que tienen la finalidad de hacer de interfaz y controlar el acceso de los dispositivos al medio. Las topologías usadas para LAN son anillo, bus, árbol, estrella, los medios de transmisión empleados son par trenzado, fibra óptica y medios inalámbricos.

2.4.1 ARQUITECTURA LAN

La arquitectura de una LAN se especifica definiendo una jerarquía de protocolos que describen su funcionamiento, los protocolos LAN están relacionados con las capas inferiores del modelo OSI (Stallings, 2004). En la figura 7 se hace una comparación entre las capas inferiores del modelo de referencia OSI y el estándar IEEE 802, esta fue creada por el comité que lleva su nombre y ha sido adoptado como base para la creación de nuevas especificaciones para LAN.

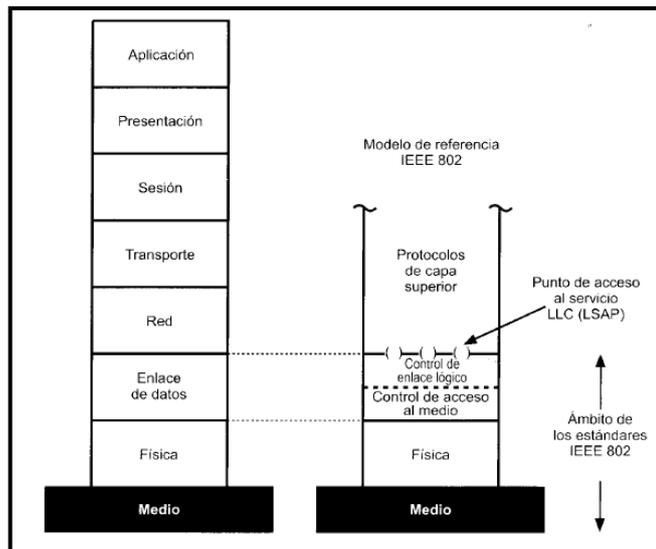


FIGURA No. 7 ARQUITECTURA DE UNA LAN. Fuente: (Stallings, 2004)

Siguiendo las arquitecturas de abajo hacia arriba, la capa inferior IEEE 802 corresponde a la capa física del modelo OSI.

Entre sus funciones se menciona:

- Codificación/decodificación de señales.
- Generación/eliminación de preámbulo
- Transmisión/recepción de bits.

Adicionalmente en el modelo IEEE 802 se especifica el medio de transmisión y la topología.

En la capa inmediatamente superior se alojan las funciones asociadas a los servicios ofrecidos a los usuarios LAN, entre sus funciones:

- En transmisión, ensamblado de datos en tramas con campos de dirección y detección de errores.
- En recepción, desensamblado de tramas, reconocimiento de dirección y detección de errores.
- Control de acceso al medio de transmisión LAN.
- Interfaz con capas superiores y control de errores y de flujo.

Las funciones implementadas en la capa física del modelo IEEE 802 se agrupan con el nombre de control de acceso al medio MAC, la interfaz con capas superiores y control de errores y de flujo se agrupan con el nombre de control de enlace lógico LLC.

2.5 TOPOLOGÍAS

2.5.1 TOPOLOGÍA EN BUS

En este caso de topología todas las estaciones se encuentran directamente conectadas por medio de una interfaz física al medio de transmisión, el funcionamiento es full-dúplex (Stallings, 2004). Hay que considerar que:

- Es necesario el implementar un mecanismo que permita decidir a qué máquina va dirigido un paquete.
- Regular la transmisión en caso de que dos o más máquinas transmitan paquetes a la vez o transmisión prolongada de paquetes.

La solución desarrollada consiste en enviar los paquetes en grupos pequeños llamados tramas, estos mensajes adicional a la información que transportan tienen segmentos de control por ejemplo cada máquina conectada a la red tiene asignada una dirección única, esta información está presente en las tramas (Stallings, 2004).

En la figura 8 se muestra la transmisión de tramas en una red LAN topología bus. En este ejemplo se envía paquetes para A, la trama correspondiente debe incluir la dirección de A, en este proceso B verifica la trama como no coincide con su dirección la deja pasar.

En esta topología no hace falta mecanismos para eliminar tramas del medio, al llegar al destino son absorbidas por el terminal.

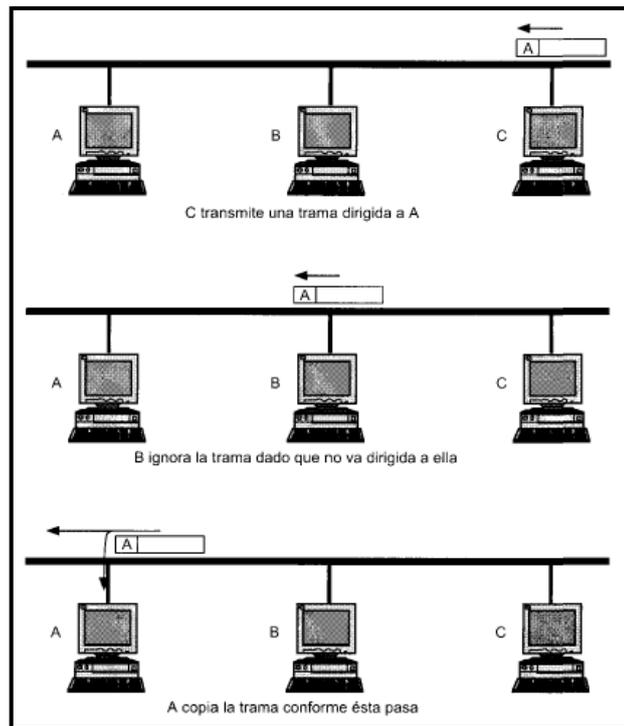


FIGURA No. 8 TOPOLOGÍA EN BUS. Fuente: (Stallings, 2004)

2.5.2 TOPOLOGÍA EN ESTRELLA

Para las redes LAN en topología en estrella cada terminal está directamente conectada a un nodo central, utilizando dos enlaces punto a punto uno dedicado a transmisión y otro dedicado a recepción (Stallings, 2004).

En la figura 9 claramente se puede visualizar el envío de paquetes dentro de esta topología. El nodo central presenta dos tipos de funcionamiento:

- En modo de difusión una trama es enviada a todos los terminales
- En modo conmutación de tramas estas se almacenan en el nodo central para luego ser retransmitido al destino

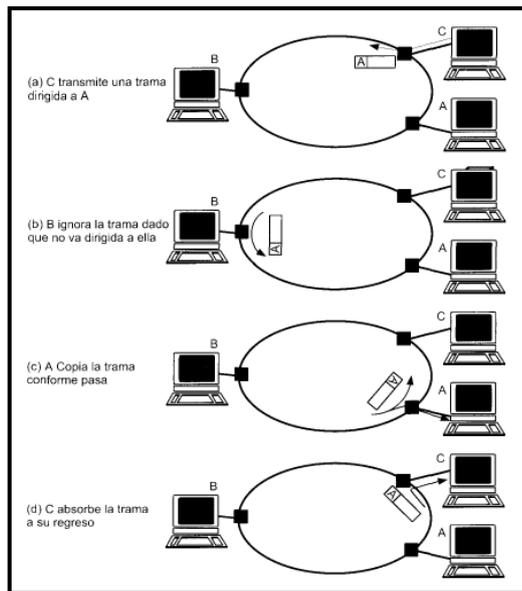


FIGURA No. 9 TOPOLOGÍA EN ESTRELLA. Fuente: (Stallings, 2004)

2.5.2.1 Control de acceso al medio

El protocolo de control de acceso al medio MAC permite el uso eficiente de la capacidad que tiene una red LAN para compartir el medio de transmisión con distintos dispositivos conectados (Stallings, 2004).

Se puede identificar dos técnicas de control de acceso al medio:

- Técnicas síncronas: utilizada en esquemas de multiplicación por división en frecuencias, conmutación de circuitos, dedica una capacidad dada a la conexión, esta implementación no es eficiente en redes LAN debido a que las necesidades de comunicación de las terminales son impredecibles.
- Técnicas asíncronas: permite asignar de forma dinámica la capacidad de la red dependiendo de las solicitudes inmediatas de los terminales.

2.5.2.2 Control de enlace lógico

El protocolo LLC presente en la capa de enlace de datos se emplea para comunicar capas superiores con inferiores, esta comunicación generalmente se refiere al software de la red y el hardware del dispositivo (Stallings, 2004).

Este protocolo generalmente maneja paquetes IPV4, agrega información de control para entregar el paquete al terminal de destino. LLC se implementa en software de tal manera que no depende del hardware del dispositivo, LLC soporta servicios orientados a conexión y servicios no orientados a conexión.

- Especificaciones IEEE 802

El IEEE Instituto de Ingeniería y Electrónica, desarrollo el estándar IEEE 802 para redes LAN, la versión 802.3 especifica la técnica de control de acceso al medio MAC y las opciones de medios de transmisión (Stallings, 2004).

- Especificaciones IEEE 802.3

El estándar IEEE 802.3 se aplica en una LAN CSMA/CD persistente-1, en este tipo de redes si un terminal quiere transmitir escucha el cable, si está ocupado el terminal espera de otra manera transmite inmediatamente (Stallings, 2004).

Cuando dos o más estaciones quieren transmitir a la vez pueden ocasionar una colisión, todas las estaciones cancelan la transmisión y esperan un tiempo aleatorio para repetir el proceso.

- Cableado del estándar IEEE 802.3

10BASE-T es el medio de transmisión que actualmente se utiliza, diez indica la velocidad de transmisión en megabits por segundo Mbps, base abrevia banda base y la T indica el uso de cables de par trenzado concretamente cable utp de categoría cinco (Stallings, 2004).

- Especificaciones IEEE 802.3u

Ethernet a alta velocidad o fast Ethernet son especificaciones desarrolladas por el comité IEEE 802.3, para tener velocidades de transmisión de 100 Mbps pero compatibles con Ethernet. Estos estándares también se conocen como 100BASE-T (Stallings, 2004).

La diferencia entre el estándar IEEE 802.3 y el IEEE 802.3u es a nivel físico, se mantiene CSMA/CD y el protocolo de control de acceso al medio MAC.

El medio más utilizado en fast Ethernet es el cable utp categoría cinco, este estándar es completamente compatible con Ethernet permitiendo utilizar en la red los mismos dispositivos.

2.6 MICRO-CONTROLADORES

Los micro-controladores son circuitos integrados con las siguientes características:

- Procesador (CPU).
- Memorias RAM.
- EEPROM.
- Líneas de entrada y salida.
- Oscilador de reloj.
- Módulos controladores de periféricos.

Un Micro-controlador debe ser programado para que realice desde la tarea más simple hasta sofisticados procesos (Reyes, 2006).

2.6.1 APLICACIONES DE LOS MICRO-CONTROLADORES

Estos circuitos integrados están siendo empleados en gran cantidad de sistemas que se utilizan en la vida cotidiana por nombrar algunos: televisores, impresoras, módems, alarmas. Y en otras aplicaciones un poco más especializadas por ejemplo: instrumentación electrónica, control de sistemas en una nave espacial, etc.

2.6.2 ARQUITECTURA DE PROCESO

En lo referente a datos y direcciones los micro-controladores trabajan de dos formas:

- La arquitectura de Von Neumann tiene una memoria principal donde se almacenan conjuntamente datos e instrucciones, se acceden a través de un bus único.
- La arquitectura Harvard dispone de dos memorias completamente independientes para datos e instrucciones, cada memoria tiene su respectivo sistema de buses de acceso y es posible trabajar a la vez en ambas memorias.

2.6.3 EL PROCESADOR CPU

Es el componente más importante del micro-controlador entre sus funciones se menciona: direccionamiento de la memoria de instrucciones, ejecución de instrucciones, almacenamiento de resultados.

De acuerdo a como procesa las instrucciones existen tres tendencias:

- Computadores de juego de instrucciones complejo (CISC): Con más de 80 instrucciones de máquina disponibles, dependiendo de su complejidad algunas se ejecutaran en varios ciclos.
- Computadores de juego de instrucciones reducido (RISC): Con un número de instrucciones de máquina muy reducido capaces de ejecutarse en un solo ciclo, el hardware y el software del procesador puede ser optimizado.
- Computadores de juego de instrucciones específico (SISC): Adicionalmente de un número de instrucciones reducido las instrucciones se adaptan a la aplicación prevista.

2.6.4 MEMORIA

En los micro-controladores la memoria puede ser volátil tipo RAM y no volátil tipo ROM. Ya que estos se ejecutan en un solo programa, se dice que en la memoria ROM se guarda y se ejecuta el programa, mientras que en la memoria RAM que es de poca capacidad se guardan las variables y los cambios de información que se den en el transcurso del programa.

Los cinco tipos de memoria ROM se describen a continuación:

- ROM con máscara: de solo lectura el contenido se graba con la fabricación del circuito integrado.
- Programable solo una vez OTP: de solo lectura, es el usuario quien graba el programa en el circuito integrado.
- EPROM: de solo lectura puede borrarse y grabarse muchas veces, para borrar se utiliza rayos ultravioleta.
- EEPROM: de solo lectura, programable y borrable eléctricamente.

2.6.5 LINEAS DE ENTRADA Y SALIDA (E/S)

Las líneas de E/S son imprescindibles en un micro-controlador, a un grupo de líneas se les denomina puertos, permiten ingresar y salir información al procesador. Estos puertos en función de la programación implementada soportan diferentes funciones, por ejemplo un grupo de puertos pueden funcionar como E/S digitales o como E/S analógicas.

2.6.6 RELOJ PRINCIPAL

Todas las operaciones que realiza un micro-controlador se sincronizan con una señal de reloj, internamente el circuito integrado dispone de un oscilador de alta frecuencia que genera una onda cuadrada para el reloj.

Adicionalmente se implementa un circuito externo con la finalidad de estabilizar y seleccionar la frecuencia del reloj.

2.6.7 RECURSOS ESPECIALES

En los principales recursos especiales que se añaden a un micro-controlador podemos mencionar:

- Conversor analógico digital A/D
- Conversor digital analógico D/A

- Modulación de ancho de pulso PWM
- Temporizadores o timers.
- Puertos de comunicación.

2.7 TARJETAS CONTROLADORAS DE ETHERNET

Existen micro-controladores con módulos para Ethernet, estos dispositivos son de montaje superficial y de gran cantidad de pines (80, 100).

Otra opción para dotar de conectividad a un dispositivo que originalmente no la tiene es el empleo de tarjetas externas. Estos módulos se encargan de gestionar todo lo referente a la conexión a internet y se comunican con el micro-controlador por SPI. Su ventaja es bajo costo de implementación y el reducido tamaño del equipo. Además estos pueden mejorar el desempeño gracias a que se les puede ir añadiendo características como: conexión inalámbrica, tarjetas de memoria.

Los principales fabricantes de este tipo de tecnología se mencionan: Rabbit, Lynx, Arduino, Microchip y de estos algunos son plataformas libres. Las características que comparten estos módulos son: IEEE802.3, Pila TCP/IP, MAC, soporte para TCP, UDP, HTTP entre otras.

2.7.1 ARDUINO ETHERNET SHIELD

La Arduino Ethernet Shield es una tarjeta controladora externa basada en el chip Wiznet W5100, capaz de conectar a internet un micro-controlador (Arduino, Arduino Ethernet Shield, 2015). Este dispositivo es una plataforma de código abierto, razón por la cual se utilizara en la construcción de la micro web server.

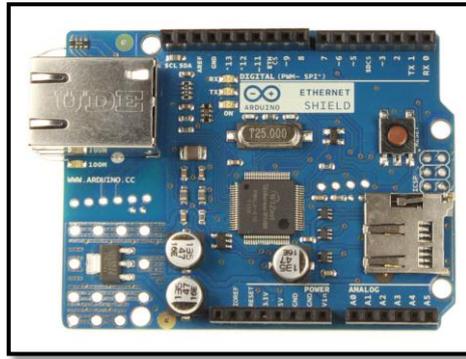


FIGURA No. 10. ARDUINO ETHERNET SHIELD. Fuente: (Arduino, Arduino Ethernet Shield, 2015)

Las características más importantes de la Shield son:

- Voltaje de alimentación 5v
- Controladora Ethernet W5100
- Velocidad de conexión 10/100 Mb
- Comunicación SPI
- Ranura para tarjeta micro sd.

2.7.1.1 Chip Wiznet W5100

Es un chip controlador de Ethernet, el que incluye la pila TCP/IP implementada por hardware, su principal campo de aplicación son sistemas embebidos con conectividad hacia internet.



FIGURA No. 11 CHIP WIZNET W5100. Fuente: (WIZnet Co, 2008)

Características:

- Implementa protocolos TCP, UDP, ICMP, IPV4, ARP, IGMPv2, PPPoE, Ethernet

- Soporta 8 sockets independientes simultáneos
- 10/BaseT/100BaseTX Ethernet PHY
- Auto negociación(Full-duplex/Half-duplex)
- Ancho de bus 16 / 8 bit
- Memoria interna de 128 kbytes para comunicación de datos Tx/Rx
- Modo SPI(Serial Peripheral Interface)
- Opera a 3.3V

2.8 COMUNICACIÓN SPI

SPI (Serial Peripheral Interface) es una conexión de tres hilos síncrona, se usa para comunicar micro-controladores con periféricos u otros micro-controladores (López Pérez, S. F.). En la figura 12 se representa las líneas de comunicación, SPI es un estándar ampliamente utilizado, soporta los modos master y slave.

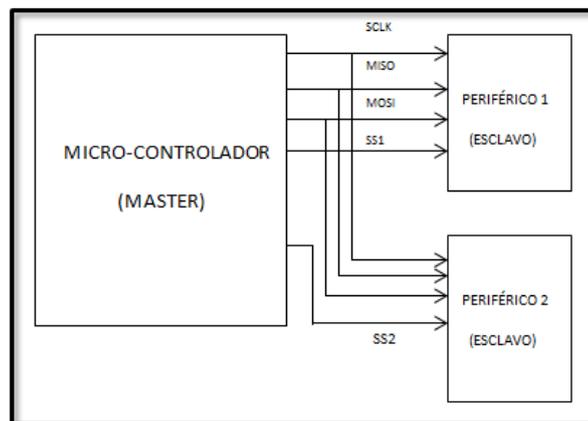


FIGURA No. 12 COMUNICACIÓN SPI. Fuente: (López Pérez, S. F.)

Soporta paquetes de 8 bits, y la comunicación es full dúplex, dos hilos se utilizan para datos y la tercera línea es para el reloj.

2.8.1 ESPECIFICACIONES DEL BUS

- SCLK es el reloj generada por el maestro, sincroniza la transferencia de datos.

- MOSI (Master Out Slave In) comunica datos del maestro al esclavo.
- MISO (Master In Slave Out) comunica datos del esclavo al maestro.
- CS o SS (Chip Select o Slave Select) selecciona esclavo.

2.8.2 SEÑALES ANALÓGICAS Y DIGITALES

- Señales analógicas: Son señales que se representan por funciones que considerando cualquier intervalo toman un número infinito de valores (Castro, Lechtaler, y Fusario, 1999). En la Figura 13 se representa una señal analógica

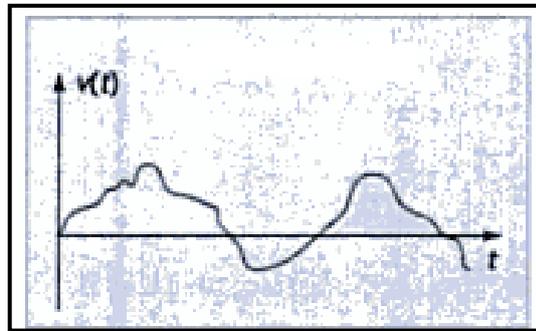


FIGURA No. 13. SEÑAL ANALÓGICA EN UN INTERVALO DE TIEMPO t . Fuente: (Castro, Lechtaler, y Fusario, 1999)

- Señales digitales: Son señales que se representan por funciones que considerando cualquier intervalo toman un número finito de valores (Castro, Lechtaler, y Fusario, 1999). En la Figura 14 se representa una señal digital

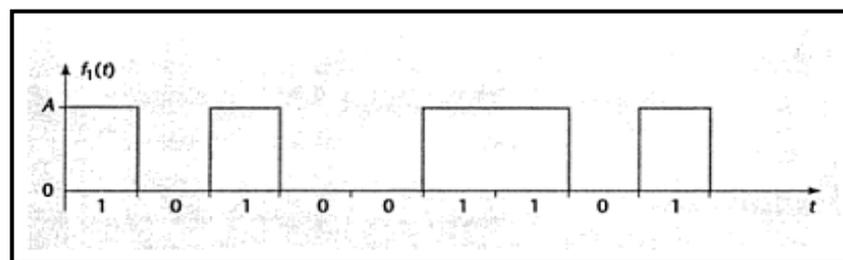


FIGURA No. 14 SEÑAL ANALÓGICA EN UN INTERVALO DE TIEMPO t . Fuente: (Castro, Lechtaler, y Fusario, 1999)

2.10 LENGUAJE C ORIENTADO A MICRO-CONTROLADORES

Todos los micro-controladores guardan los programas en la memoria flash este se conoce como código ejecutable y está en binario. La agrupación de cierta cantidad de bits 12, 14 o 16 se le llama palabra, el micro-controlador interpreta cada palabra como una instrucción a ser ejecutada durante su ciclo de funcionamiento.

Todas las instrucciones que el micro-controlador reconoce y ejecuta se les llaman conjunto de instrucciones, para facilitar la manipulación del código ejecutable este se representa en numeración hexadecimal denominado código Hex (Knudsen, 2011).

2.10.1 LENGUAJES DE BAJO Y ALTO NIVEL

- Lenguaje ensamblador de bajo nivel

La programación en ensamblador lo que hace es codificar instrucciones simbólicas, generando su respectiva instrucción en lenguaje de máquina.

Programar en ensamblador proporciona mayor control sobre el hardware, genera módulos ejecutables más compactos y la ejecución de procesos es más rápida.

- Lenguaje de alto nivel

La programación de alto nivel codifica comandos fuertes que generan muchas instrucciones en lenguaje de máquina ejemplos de este tipo de lenguaje son C, Pascal, C++. Una de las principales ventajas de este tipo de programación es que el código que manipula el programador es considerablemente más pequeño que su equivalente en ensamblador (Abel, 1996).

En la figura 15 se muestran tanto el lenguaje ensamblador de bajo nivel, como el lenguaje de alto nivel, cada uno con sus características más fundamentales.

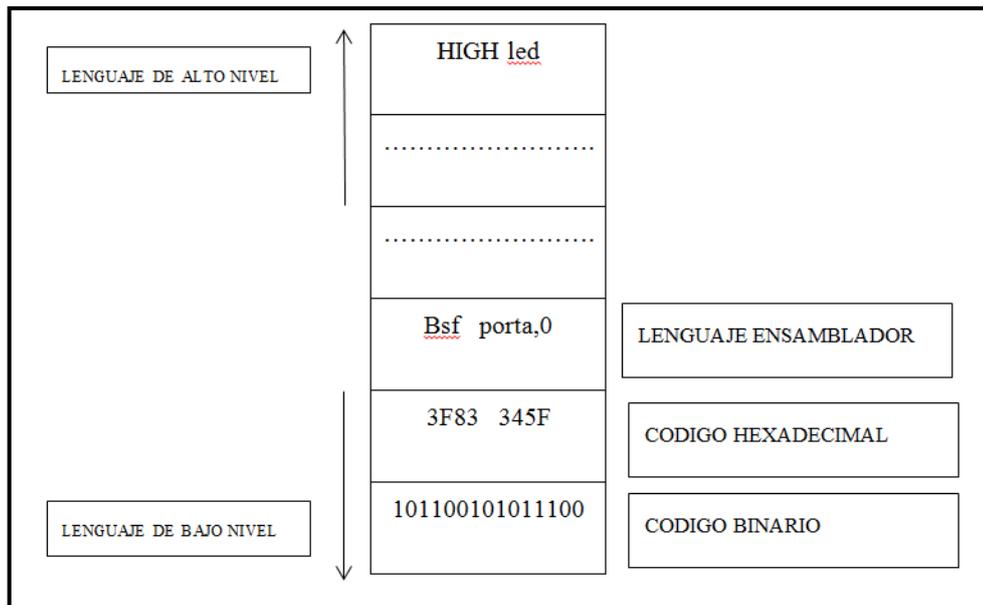


FIGURA No. 15 NIVELES DE PROGRAMACIÓN. Fuente: (Abel, 1996)

2.8.3 LENGUAJE C

2.8.3.1 Elementos básicos

- Comentario

Se utiliza para la documentación del código.

Sintaxis: //comentario

- Inicio y fin de bloque

Agrupar instrucciones para darles prioridad

Sintaxis: { //inicio de bloque

//instrucciones

} // final de bloque

- Identificador y tipo

El identificador es el nombre que se da a una variable o función, el tipo es una palabra reservada que indica que tipo de dato se ha definido. En la tabla 1 se indican los tipos más utilizados.

TABLA No. 1 ALGUNOS TIPOS DE DATOS. Fuente: (Knudsen, 2011)

TIPO	CÓDIGO
char	Entero de 8 bit
int	Entero de 16 bit
long	Entero de 32 bit
float	Real de 32 bit

Para la definición de variables se utiliza el identificador y el tipo de la siguiente manera:

```
char i; //variable tipo char de 8 bits
```

2.8.3.2 Estructura básica de un programa

Todo programa básicamente maneja la siguiente estructura:

```
void setup () {  
  
// código de configuración  
  
}  
  
Void loop () {  
  
// código principal  
  
}
```

- Void setup

La función setup se invoca siempre al iniciar un proyecto, se utiliza para inicializar variables, funciones. Esta parte del código se ejecuta solamente una vez al iniciar el programa.

- Void loop

Indica el inicio del cuerpo principal del programa, en esta sección se ejecutan recursivamente las instrucciones.

2.8.3.3 Instrucciones básicas

Las instrucciones básicas para hacer un programa se consideran:

- Asignaciones

Consiste en mover un valor a una determinada variable

p =1;

q=2;

- Operaciones Matemáticas

Se dispone de las operaciones básicas, división entera y las operaciones de incremento y decremento.

k=a%b; //división entera

k++; //incremento k=k+1

k--; //decremento k=k-1

- Operaciones lógicas

Son aplicables a variables enteras constan de los operadores lógicos y desplazamientos a la izquierda y derecha.

k=i&j; AND

k=ij; OR

k=~i; NEGACIÓN

k<<<1; DESPLAZAMIENTO UN BIT IZQUIERDA

k>>>1; DESPLAZAMIENTO UN BIT DERECHA

- Sentencias de control

Permiten ejecutar un grupo de instrucciones dependiendo de una condición

== Igual

!= Distinto

> < Mayor, menor

>= <= Mayor o igual, menor o igual

&& Operador Y

|| Operador O

Sentencia if-else

Si la condición se cumple se ejecuta un grupo de instrucciones caso contrario se ejecuta las que siguen.

if (condición)

{.instrucciones}

Else

{.....instrucciones} Sentencia While

Evalúa la condición al inicio si se cumple ejecuta el bloque de instrucciones en caso contrario salta.

while (condición){.....

instrucciones}

Sentencia do while

Evalúa la condición al final

```
do {.....
```

```
instrucciones}
```

```
while(condición)
```

Sentencia switch-case

Selecciona de entre varias opciones numéricas

```
switch (expresión){
```

```
case valor1:
```

```
    Sentencias si la expresión es igual a valor 1
```

```
[brake]
```

```
Case valor2:
```

```
    Sentencias si la expresión es igual a valor 2
```

```
[brake]
```

```
.
```

```
default
```

```
    Sentencias que se ejecutan si ningún valor coincide
```

```
}
```

Sentencia for

Permite la iteración de un grupo de instrucciones

```
for (expr_1;expr_2;expr_3)
{
    .....instrucciones
}
```

Expr_1 es la condición de comienzo, expr_2 se mantiene el bucle mientras se cumpla la condición se evalúa al entrar en el bucle, expr_3 indica el paso a seguir al terminar el proceso iterativo.

2.8.3.4 Manejo de puertos de Entrada/Salida

Los micro-controladores disponen de por lo menos tres puertos que pueden ser configurados como entradas-salidas digitales o entradas analógicas. Los nombres que se asignan a los puertos dependen del fabricante por ejemplo PORTA, POTRB, PORTC.

Con el registro TRIS se define que puerto será de entrada o de salida por ejemplo TRISA, TRISB, etc. Para programar los puertos se consideran como variables enteras de 8 bits y pueden ser accedidas por bit o por byte (Reyes, 2006).

```
TRISA.0=0; //El bit cero de puerto A se configura como salida
```

```
TRISB.4=1; //El bit cuatro del puerto A se configura como entrada.
```

2.9 LENGUAJE HTML

HTML es el lenguaje utilizado para la creación de páginas web, las instrucciones y comandos son interpretadas por el programa navegador (Google chrome, Mozilla), HTML tiene las siguientes características:

- Sintaxis sencilla
- No hay variables
- Las instrucciones no se compilan
- Permite escribir hipertexto
- Se denominan etiquetas a las instrucciones
- La página puede ser escrita por completo utilizando cualquier editor de texto, guardando el archivo con extensión .html.

2.9.1 ESTRUCTURA BÁSICA HTML

El lenguaje HTML hace uso de etiquetas (tags) de la manera siguiente:

<XXX> Inicio etiqueta

</XXX> Fin etiqueta

Las letras de la etiqueta no distinguen entre mayúsculas o minúsculas. Todo el documento HTML debe estar entre las etiquetas <HTML> [DOCUMENTO] </HTML>.

Todos los documentos HTML se componen de la siguiente forma:

- Cabecera <HEAD> </HEAD>. Contiene información que generalmente no se muestra en el navegador web, en esta sección se encuentra el nombre del documento HTML. dentro de las etiquetas <TITLE> </TITLE>.
- Cuerpo <BODY> </BODY>. Contiene todo lo que se muestra en el navegador web texto, imágenes, enlaces.

<HTML>

<HEAD>

<TITLE> TÍTULO PÁGINA </TITLE>

</HEAD>

<BODY>

[ETIQUETAS QUE SE VISUALIZAN EN EL NAVEGADOR]

</BODY>

</HTML>

2.9.2 CONFIGURACIÓN DE TEXTO

HTML dispone de varias opciones para manipular el texto que aparece en el navegador, a continuación se describen las principales.

- <P> Separa el texto en párrafos, generalmente no necesita </P> para cerrar la etiqueta.
- <H1> </H1> Permite crear encabezados hasta el número 6 siendo el uno el de mayor tamaño.
- <CENTER> </CENTER> Centra tanto texto como imágenes
- <HR> Etiqueta que permite separar horizontalmente textos, párrafos entre sí, el espacio de separación es definido por el programador en píxeles o en porcentaje.
- <!--> <--> Comentario

2.9.3 ENLACES

La sintaxis que permite la realización de enlaces con recursos dentro o fuera de la página web es YYY . XXX indica el destino del enlace, YYY es texto que aparece en el navegador indicando un enlace (Bernál, S. F.). En el caso de que el enlace sea un recurso fuera del servidor de la página web, se necesita conocer la dirección del enlace externo. La sintaxis para implementar esta función es la siguiente

2.9.4 IMÁGENES

La etiqueta para insertar imágenes en la página web es similar en su sintaxis a la correspondiente para realizar enlaces externos (Bernál, S. F.). La diferencia entre las dos instrucciones es que en lugar de indicar un enlace a otra página se indica el lugar donde se encuentra una imagen. La estructura de la etiqueta es la siguiente ``, se puede agregar características adicionales al proceso de inserción de imágenes, algunos de los comandos son:

- `` Descripción de la imagen

Títulos con respecto a la imagen:

- ``
- ``
- ``

Alineación de imágenes

- ``
- ``
- ``

Dimensionado de la imagen

- ``

2.9.5 PRUEBA T DE STUDENT DE DOS MUESTRAS RELACIONADAS

Este método permite reconocer la existencia de diferencias significativas entre los promedios de dos variables, relacionadas por alguna característica. La aplicación de este método requiere el mismo número de elementos en ambas situaciones (DeVore, 2005).

Este tratamiento de muestras requiere los siguientes parámetros:

- Promedio de los datos obtenidos.

- Varianza.

$$s^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

Donde \bar{x} son las medias aritméticas de las muestras, n el número de muestras.

- Intervalos de confianza.

Es un rango dentro del cual se encuentra el verdadero valor del parámetro promediado.

- Desviación estándar

$$s = \sqrt{s^2}$$

- Diferencia entre intervalos de confianza.

Comparando la diferencia entre intervalos de confianza se puede tener una primera aproximación para la significancia de los promedios de los datos. En el caso de que este parámetro se aproxime en todas las muestras hay indicio de poca significancia, este resultado se utiliza para asumir varianzas iguales en las muestras.

CAPÍTULO III

3 METODOLOGÍA

Las primeras fases del proyecto de investigación comprendían un amplio estudio bibliográfico de temas concernientes a fundamentos de redes, servidores y sistemas embebidos. Se utilizó la metodología de investigación experimental para seleccionar los micro-controladores, componentes electrónicos y entorno de programación más idóneos para el desarrollo del prototipo.

Con esta información el siguiente paso fue construir el circuito electrónico y la página web que alojaría, la experimentación y ensayos proporcionaron los resultados de la investigación.

3.1 TIPO DE ESTUDIO

El tipo de estudio empleado es teórico práctico referente a la electrónica y el software necesario para dotar de conectividad en internet a un micro-controlador. Mediante la investigación con fuentes bibliográficas especializadas en desarrollo de sistemas embebidos y en los manuales de los fabricantes de micro-controladores y aplicaciones con características similares ya implementados en industrias.

3.2 POBLACIÓN Y MUESTRA

La población del proyecto de investigación comprende los servidores embebidos, específicamente en los que manejan el protocolo HTTP para aplicaciones web, SPI como interface de comunicación con periféricos y soporte para el estándar IEEE 802.3.

El proceso de muestreo es no aleatorio debido a que no se generalizaran los resultados obtenidos para toda la población. El servidor web construido será comparado con un equivalente existente en el mercado para verificar funcionalidad.

3.3 PROCEDIMIENTOS

3.3.1 REQUERIMIENTOS DEL SISTEMA

Para el diseño y construcción de un dispositivo capaz de monitorear y controlar señales analógicas y digitales. Donde los resultados se muestren en una página web accesible en una red LAN, se establecen las siguientes especificaciones técnicas:

ESPECIFICACIONES

- Interfaz de página web
- Accesible mediante dirección IP y puerto, en una red LAN cableada
- Monitorea y controla señales analógicas y digitales
- Eficiente y de bajo costo

Para cumplir con las especificaciones propuestas el sistema se desarrolla en base a micro-controladores capaces de procesar señales analógicas y digitales. El prototipo debe tener hardware adicional que gestione la conectividad con la red.

DIAGRAMA DE BLOQUES

El problema se divide en tres partes. En primer lugar se considera el procesamiento de los datos obtenidos por los sensores en el micro-controlador, como esta información debe ser transmitida en una red local se necesita una sección para dotar de conectividad al micro-controlador, se utiliza una tarjeta controladora de Ethernet. Como consideración final se tiene la interfaz con el usuario. En la figura 16 se muestra el diagrama de bloques.

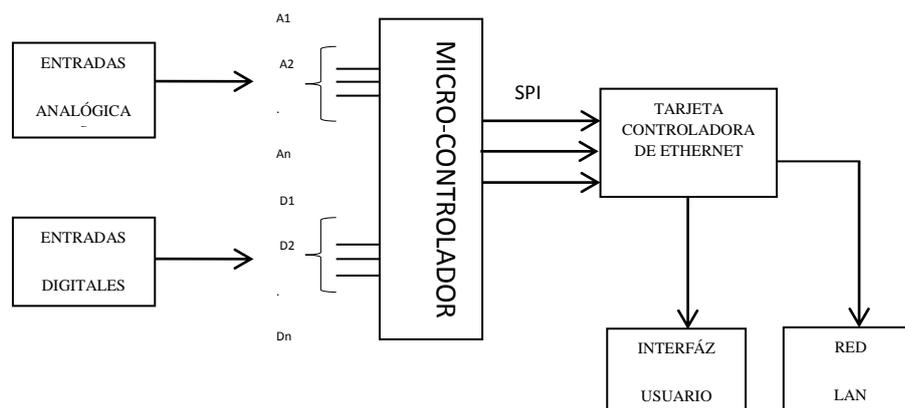


FIGURA No.16. DIAGRAMA DE BLOQUES MICRO WEB SERVER

3.3.2 SELECCIÓN DE TARJETA ETHERNET Y MICRO-CONTROLADOR

Para la selección de la tarjeta controladora externa se analizó los siguientes parámetros:

- Compatibilidad (IEEE 802.3)
- Soporte para la pila TCP/IP
- Dirección MAC incorporada
- Comunicación SPI
- Protocolos TCP, UDP, ICMP, IPV4, ARP, IGMP

TABLA No. 2 DISPOSITIVOS DISPONIBLES PARA SELECCIÓN

TARJETA	ESTANDAR	PROTOCOLO	SPI	TCP/IP	PLATAFORMA ABIERTA	SELECCIÓN
EZ WEB LYNX	IEE 802.3/IEEE802.3u	TCP/UDP/IPV4/ICMP/ARP	NO	SI	NO	X
RABBIT RCM3900	IEE 802.3/IEEE802.3u	TCP/UDP/IPV4/ICMP/ARP	SI	SI	NO	X
ENC624J600	IEE 802.3/IEEE802.3u	TCP/UDP/IPV4/ICMP/ARP	SI	SI	SI	✓
ENC28J60	IEE 802.3	TCP/UDP/IPV4/ICMP/ARP	SI	SI	SI	✓
SP1-SITEPALYER	IEE 802.3/IEEE802.3u	TCP/UDP/IPV4/ICMP/ARP	NO	SI	NO	X
ARDUINO ETHERNET SHIELD	IEE 802.3/IEEE802.3u	TCP/UDP/IPV4/ICMP/ARP	SI	SI	SI	✓
XPORT	IEE 802.3/IEEE802.3u	TCP/UDP/IPV4/ICMP/ARP	NO	SI	NO	NO
TIBBO EM203	IEE 802.3/IEEE802.3u	TCP/UDP/IPV4/ICMP/ARP	SI	SI	SI	✓

Como se puede visualizar en la Tabla No. 2, se muestran una comparativa de las tarjetas controladoras y sus funcionalidades. A parte de las características técnicas de la tarjeta controladora, se debe tomar en cuenta la facilidad de adquisición en el

mercado. Evaluando todos estos criterios la Arduino Ethernet Shield es el componente más idóneo para la construcción de la micro web server.

3.3.3 MICRO-CONTROLADOR

El Atmega 328P es un micro-controlador de bajo consumo de energía, de 8 bits, basado en la arquitectura RISC capaz de ejecutar instrucciones de alta complejidad en un solo ciclo de reloj.

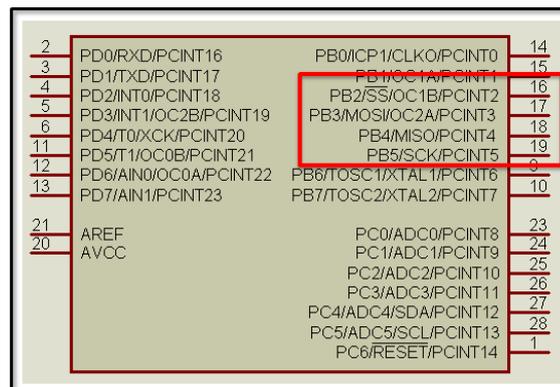


FIGURA No. 17 Pines SPI en el Atmega 328P

La elección del Atmega 328P para la construcción del micro web server se fundamenta en que este micro-controlador soporta comunicación SPI al igual que la Ethernet Shield. En la figura 17 se muestra los pines correspondientes a SPI.

3.3.4 CONSTRUCCIÓN DE HARDWARE

La utilización de software para la simulación del funcionamiento del circuito a construirse es muy importante porque anticipadamente permite detectar errores en el diseño.

3.3.4.1 Circuito micro-controlador

La figura 18 muestra los pines del micro-controlador Atmega 328P en donde se puede ver cuáles son para alimentar el dispositivo.

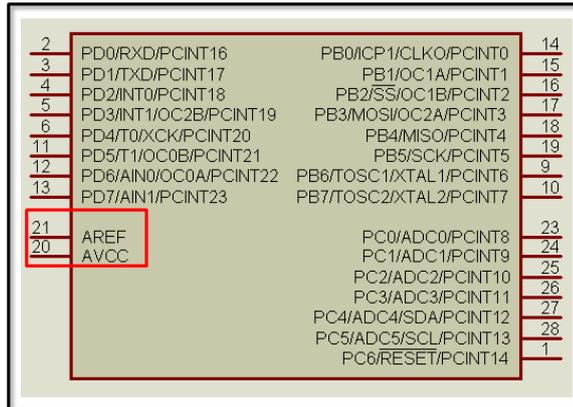


FIGURA No. 18 ATMEGA 328P PINES DE ALIMENTACIÓN

El pin siete va a 5 voltios, el pin ocho y 22 puenteados a tierra, el pin 21 se lo puede dejar sin conexión o ponerlo a 5 voltios y el pin 20 a 5 voltios. Para estabilizar el reloj del micro-controlador entre los pines nueve y 10 se colocan un oscilador de 1Mhz y dos capacitores cerámicos de 1 nF puenteados a tierra. En la figura 19. Se muestra el diagrama de circuito para alimentar el micro-controlador con su respectivo oscilador externo.

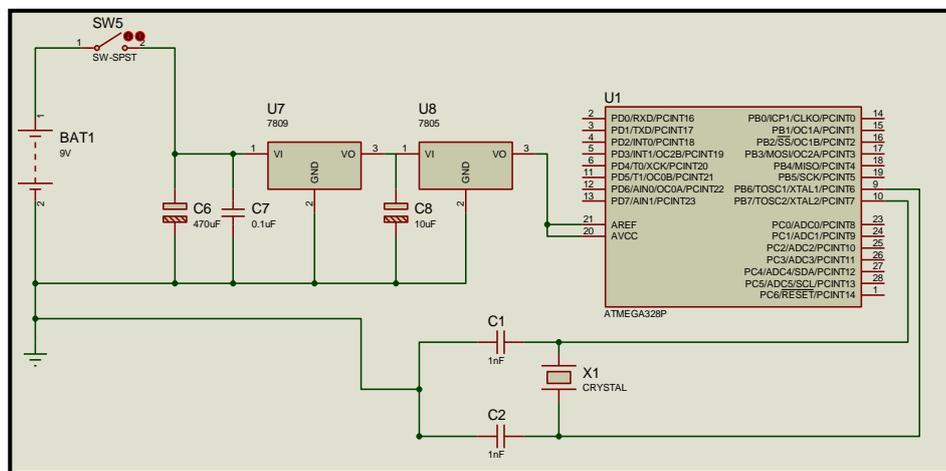


FIGURA No. 19 FUENTE Y OSCILADOR ATMEGA328P

3.3.4.2 Comunicación SPI micro-controlador Ethernet Shield

Para la conexión entre el micro-controlador y el Ethernet Shield se utiliza SPI, en la tabla No. 3 se indican los pines necesarios.

TABLA No. 3. PINES PARA SPI

PIN	ATMEGA328P	ETHERNET SHIELD
MOSI	17	7
MISO	18	5
SCK	19	6

Adicionalmente a las líneas estándar que especifica este protocolo para el correcto funcionamiento se deben puentear los pines 21 y 20 del Atmega 328P con los pines dos y 11 del Ethernet Shield.

Además se especifica la utilización de la tarjeta sd que puede ser puesta en el Ethernet Shield, en donde se conectan los pines seis del Atmega 328P y el pin ocho del Shield. Todo esto se puede visualizar en la figura 20, conjuntamente con lo relacionado a la adquisición y envío de datos

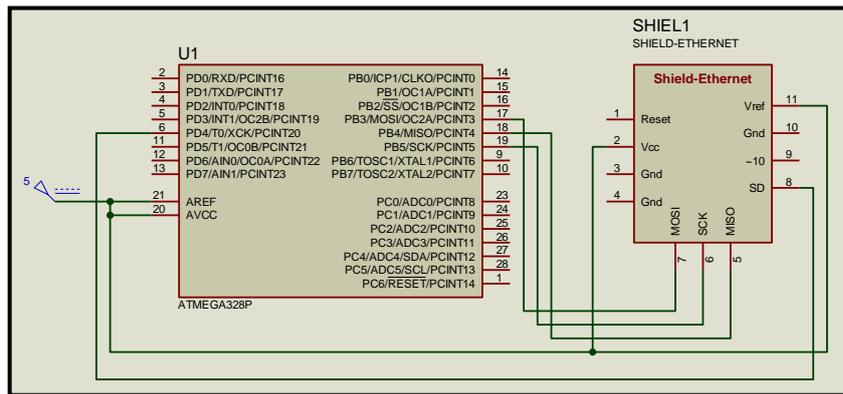


FIGURA No. 20 SPI MICRO-CONTROLADOR Y ARDUINO ETHERNET SHIELD

3.3.4.3 Señales analógicas

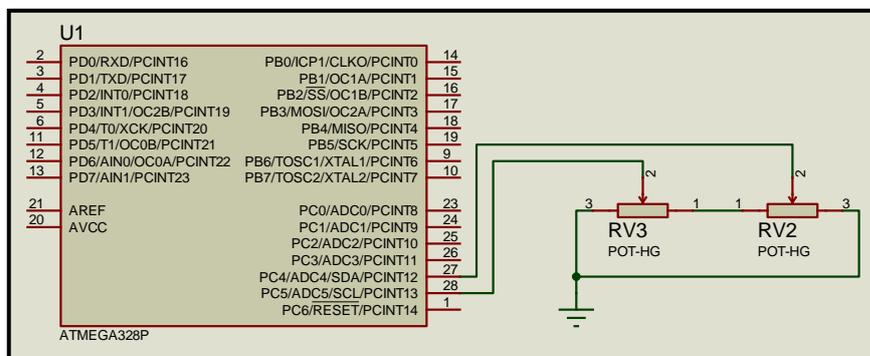


FIGURA No.21 LECTURA DE SEÑALES ANALÓGICAS

3.3.4.4 Señales digitales

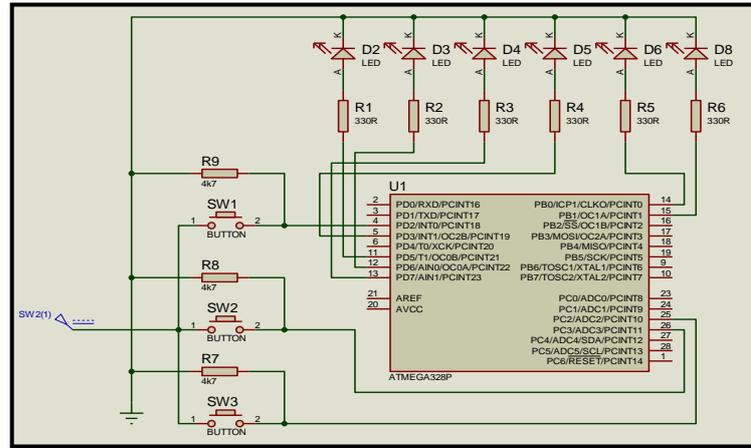


FIGURA No. 22 MONITOREO Y CONTROL DE SEÑALES DIGITALES

El sistema permite el monitoreo de dos potenciómetros cuyos estados pueden ser visualizados en una interfaz de página web. Los valores de estos datos son ingresados al micro-controlador a través de los pines 27 y 28 como se muestra en la figura 21 y en la figura 22 se muestra la circuitería para el envío y recepción de datos

Se modifica el estado de un grupo de diodos led ON/OFF, el circuito permite reconocer el estado de un grupo de pulsadores ON/OFF.

TABLA No. 4. SEÑALES DIGITALES

PINES	DIODO LED	PULSADOR
5	x	
15	x	
14	x	
13	x	
12	x	
11	x	
26		X
25		X
4		X

3.3.5 DISEÑO E IMPLEMENTACIÓN DE SOFTWARE

La programación del micro-controlador la realiza en lenguaje de alto nivel, siendo este el más común, en este caso se necesita también un compilador para traducir el programa a lenguaje ensamblador. Para la programación del micro-controlador se empleó la herramienta AVR studio 6 de Atmel, en la fig. 23 se muestra la pantalla principal al iniciar el programa,

3.3.5.1 Entorno de programación

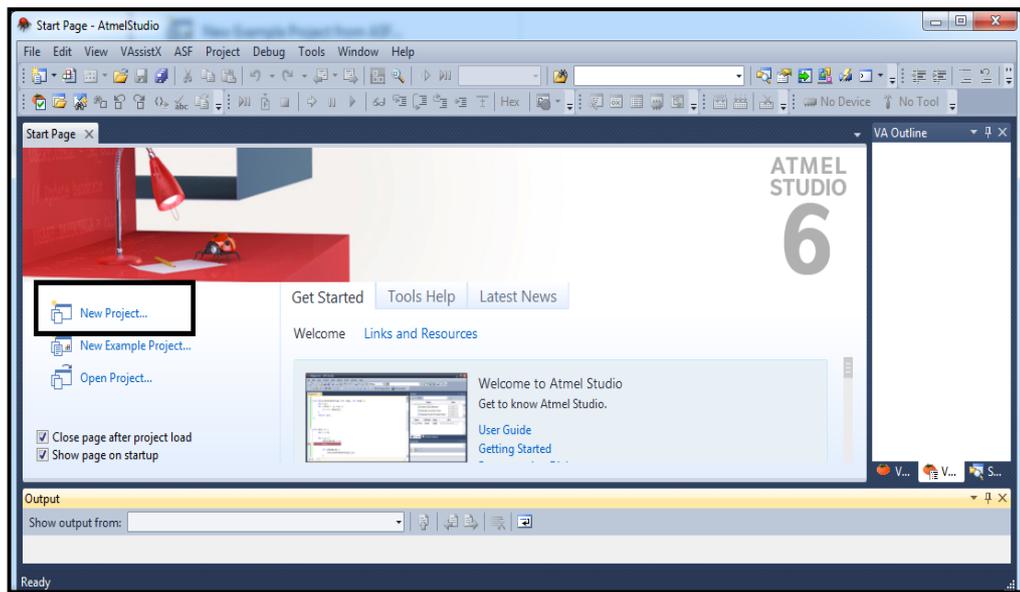


FIGURA NO. 23. PANTALLA DE INICIO AVR STUDIO 6

Este software tiene su propio compilador AVR GNU C Compiler y es capaz de programar micro-controladores de ocho y 32 bits. Luego se elige la opción new project en donde se considera:

- El lenguaje de programación a usar. C/C++ para hacer programas en C (.c).
- El tipo de proyecto a realizarse se escoge GCC C Executable Project.
- Nombre del proyecto y la ubicación.

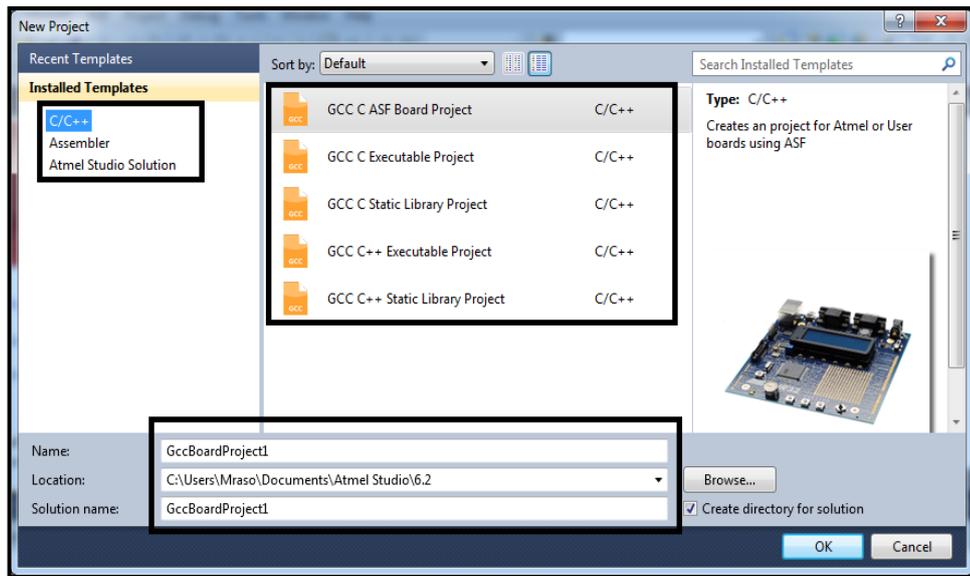


FIGURA No. 24. PANTALLA DE ELECCIÓN LENGUAJE DE PROGRAMACIÓN, TIPO DE PROYECTO, NOMBRE Y UBICACIÓN

Configuradas las opciones del proyecto se escoge el micro-controlador con el que se va a trabajar como se muestra en a figura 25.

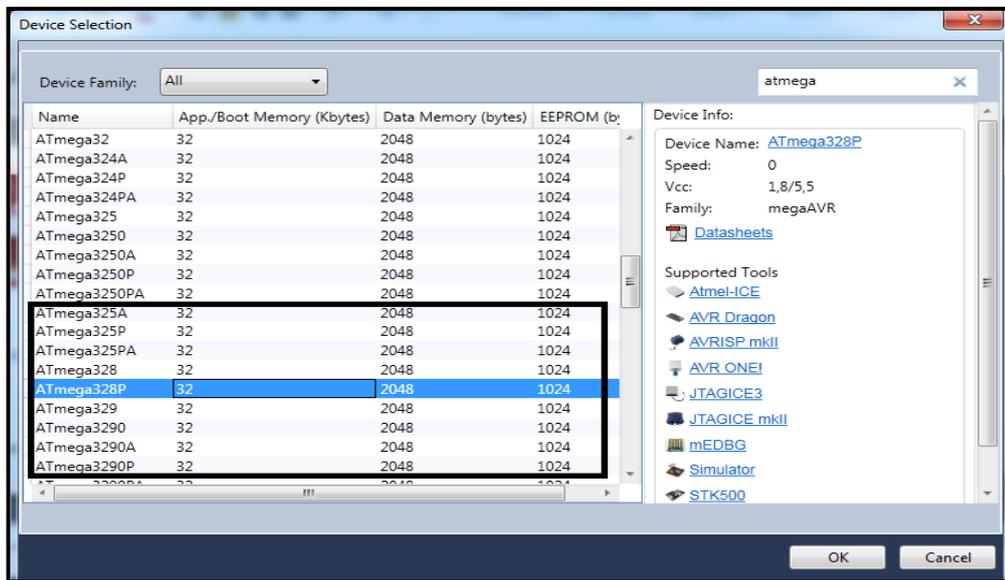


FIGURA NO. 25 PANTALLA DE ELECCIÓN DEL MICRO-CONTROLADOR

Al finalizar el proceso de elección del dispositivo aparece la plantilla donde se escribe el programa. La plantilla incluye algunas líneas de programación que especifican funciones y librerías necesarias para iniciar cualquier proyecto como

se indica en la figura 26. Estas configuraciones iniciales aseguran el reconocimiento de instrucciones básicas y librerías. Siendo indispensables al momento de iniciar un proyecto de programación.

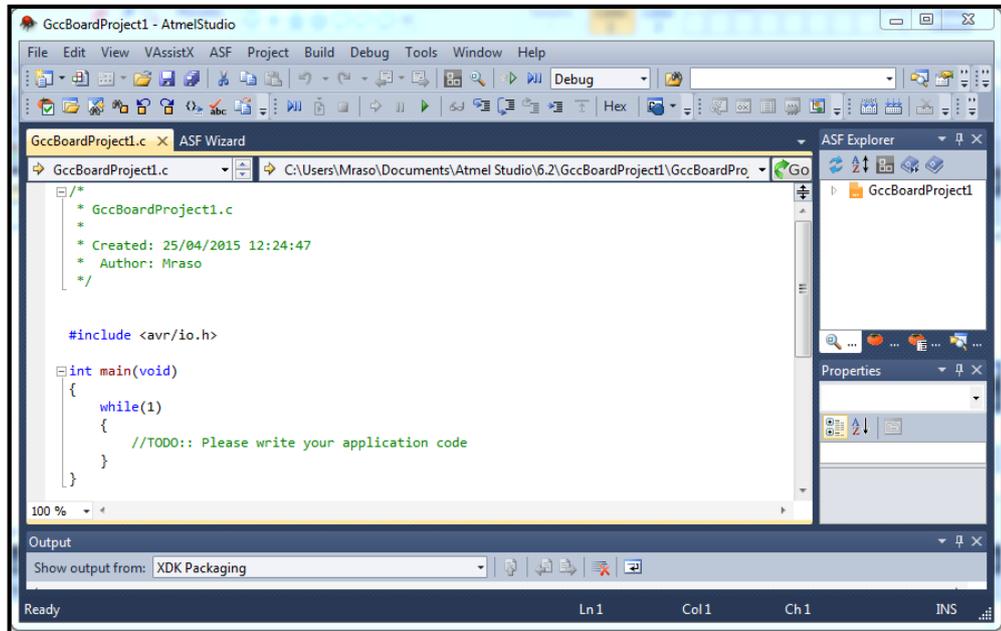


FIGURA NO. 26 PLANTILLA DE PROGRAMACIÓN AVR STUDIO

3.3.5.2 Configuraciones adicionales

Antes de empezar con la programación es recomendable configurar los siguientes parámetros frecuencia de reloj, librerías externas y compilación, como se muestra en la figura 27, para asegurar el correcto funcionamiento del micro-controlador elegido y que el proceso de compilación se estandarice para la mayoría de dispositivos elegidos para ser programados en este entorno de desarrollo.

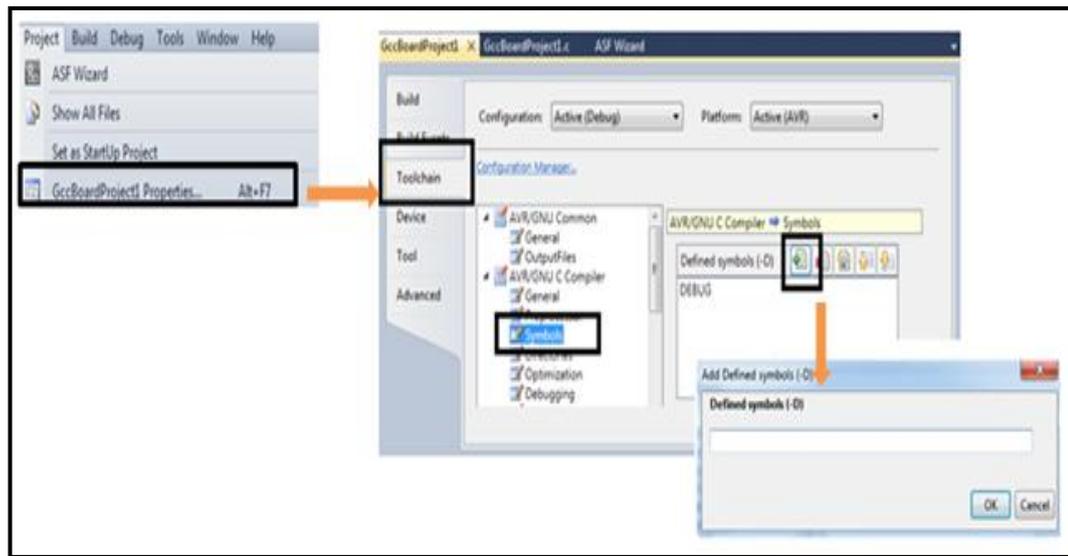


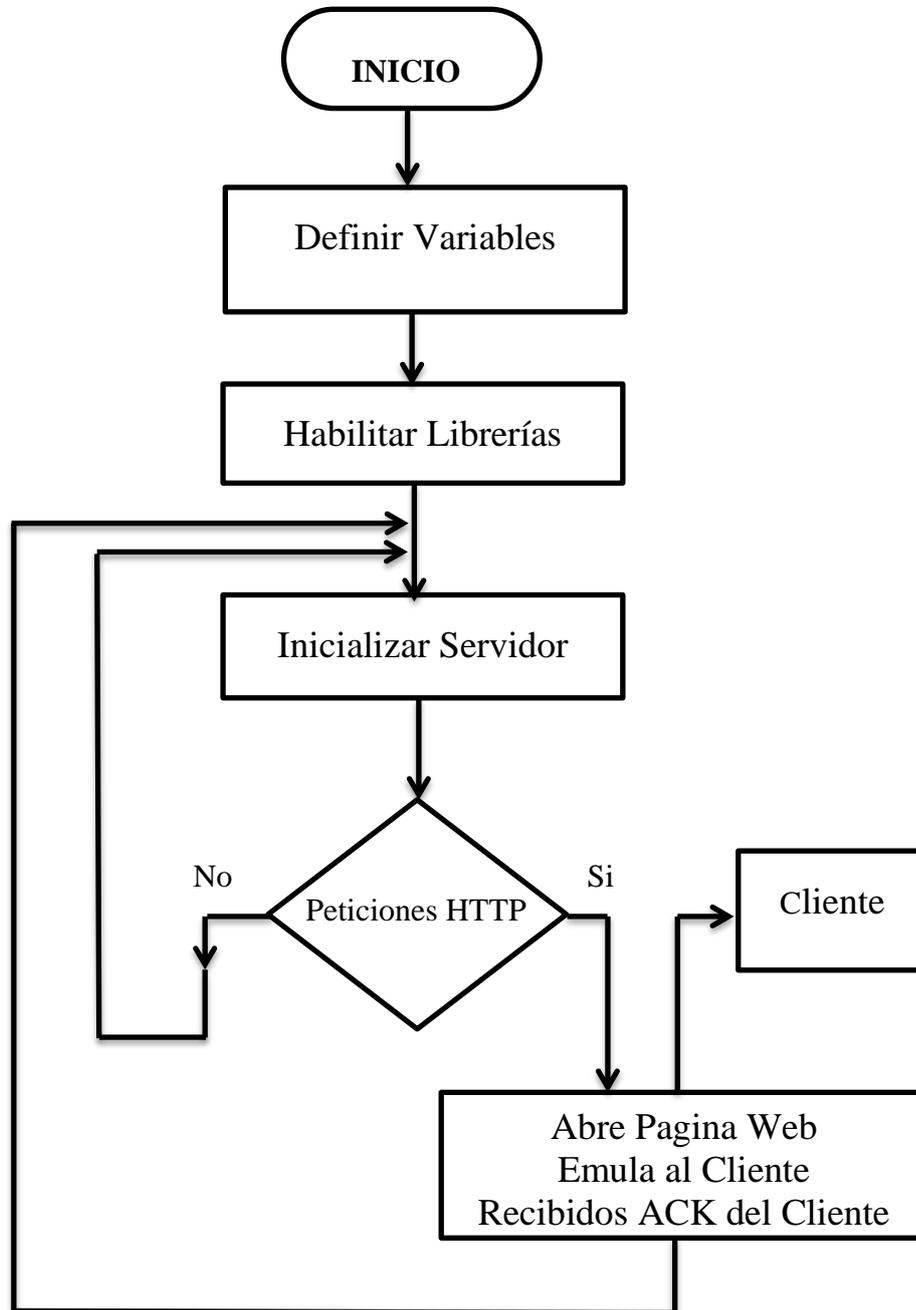
FIGURA No. 27 MODIFICAR RELOJ

3.3.5.3 Programa principal del servidor

El funcionamiento del servidor se basa en el diagrama N° 1 en donde se pueden reconocer los siguientes procedimientos:

- Definición de variables locales y globales, puertos de entrada y salida así como también el puerto de ingreso al servidor, asignación tanto de dirección IP y dirección mac.
- Habilitación de librerías para comunicación SPI, para reconocimiento de la tarjeta Ethernet Shield de Arduino y para utilizar una tarjeta sd externa para aumentar la capacidad de almacenamiento del servidor.
- Se inicializa el servidor a partir de esta configuración el dispositivo es capaz de escuchar por el puerto 8080.
- Se hace una solicitud al servidor este abre la página web y la envía hacia el cliente, el cliente controla o monitorea los procesos y envía un ACK al servidor para terminar la conexión, el servidor recibe el ACK y termina el enlace.
- El servidor regresa al estado de espera de solicitudes.

DIAGRAMA No. 1 PROGRAMACIÓN DEL SERVIDOR



3.3.6 DISEÑO DE LA PÁGINA WEB

3.3.6.1 Consideraciones

Para la realización de la interfaz de la página web se pretende tener un entorno gráfico en el cual se enviarán y recibirán información de control y monitoreo con el micro-controlador Atmega 328P, siempre considerando la simplicidad de manejo de las diferentes opciones. Según el tamaño de la página se puede utilizar una tarjeta sd externa como periférico de apoyo para el almacenamiento de la misma.

3.3.6.2 Estructura de la página

La programación de la página web fue realizada en lenguaje html usando como editor de texto el bloc de notas de Windows, a continuación se describen los principales componentes de la página.



FIGURA No. 28 PÁGINA WEB

- <html> Etiqueta identificativa de página web
- <head> </head> Etiqueta donde se define el título de la página, la utilización del servicio JAVASCRIPT para manejar contenido dinámico y

el enlace hacia la micro web server para mostrar el estado de los potenciómetros, de los pulsadores y manipular el estado de los leds.

- `<body> ...</body>` Etiquetas dentro de las cuales se pusieron fotos, textos y demás características del entorno gráfico de la interfaz.

La página web donde se incorporan las características mencionadas se muestra en la figura 28

3.4 PROCESAMIENTO Y ANALISIS

El dispositivo utiliza una fuente de alimentación de 5 voltios para el micro-controlador, el Ethernet Shield y los demás componentes electrónicos. Para conectar el circuito a la red se utiliza cable utp cat 5e con conectores RJ45.

Para ingresar a la página web contenida en el servidor se debe digitar la dirección IP correspondiente al dispositivo y el puerto 8080 definido para este procedimiento en un navegador web. A continuación se explica el procedimiento efectuado para la comprobación del dispositivo construido.

3.5 COMPARACION ENTRE SERVIDORES EMBEBIDOS

Para comprobar el funcionamiento del micro web server se va a comparar este dispositivo con otro servidor web embebido basado en el módulo ENC28J60 y la tarjeta de desarrollo ARDUINO UNO. Los parámetros propuestos a ser comparados son:

- El tiempo que tarda el servidor web en cargar la página.
- Conectividad en la red mediante ping.

El escenario dispuesto para las pruebas es la red local de la Universidad Autónoma de los Andes UNIDEC con las siguientes características:

- Velocidad de descarga 0.99 Mbps
- Velocidad de subida 0.98 Mbps
- Promedio de usuarios 25

Las pruebas serán realizadas utilizando el servicio de internet por cable y el navegador web google chome.

TABLA No. 4 CARACTERÍSTICAS DE LAS TARJETAS EMPLEADAS

TARJETA	CARACTERÍSTICAS
<p data-bbox="331 831 801 864">DESARROLLO ARDUINO UNO</p>  <p data-bbox="384 1240 735 1267">FIGURA No. 29 ARDUINO UNO</p>	<ul style="list-style-type: none"> • Micro-controlador Atmega 328P • Voltaje de operación 5 V • Interfaz SPI • 14 pines entrada-salida digital • 6 pines de entrada analógica
<p data-bbox="485 1288 635 1321">ENC28J60</p>  <p data-bbox="416 1727 703 1753">FIGURA No. 30 ENC28J60</p>	<ul style="list-style-type: none"> • Chip Ethernet ENC28J60 • Voltaje de operación 3.3. V • Interfaz SPI • Conector RJ45 • Compatibilidad 10/100 Base-T • Soporta Full/Half dúplex

Una vez obtenidos los datos para cada dispositivo el análisis estadístico que se lo realizó para aceptar o rechazar la hipótesis planteada es el método de “t” de Student.

Debido a que se comparó dos dispositivos independientes con características que permiten relacionarlos, para determinar la existencia o no de diferencias significativas entre ellos.

Para recolectar dichos datos se tomó en cuenta los tiempos de carga y conectividad tanto en el comercial como en el desarrollado. Se consideró estos parámetros por ser los únicos que pueden ser aplicados en la tecnología que se está analizando. Toda esta información se la puede visualizar en la tabla No 5.

TABLA No. 5. ESQUEMA PARA OBTENCIÓN DE DATOS. DATA CENTER UNIANDES ENERO 2015

REPETICIONES	DISPOSITIVO COMERCIAL				MICRO WEB SERVER			
	INTENTO 1		INTENTO 2		INTENTO 1		INTENTO 2	
	TIEMPO CARGA (s)	PING (s)	TIEMPO CARGA (s)	PING (s)	TIEMPO CARGA (s)	PING (s)	TIEMPO CARGA (s)	PING (s)
1								
2								
3								
4								
5								

CAPÍTULO IV

4 RESULTADOS

Los datos obtenidos en las pruebas representan el tiempo que demora la página web en ser mostrada en un navegador de internet y la conectividad del dispositivo en la red utilizando el comando ping para enviar paquetes.

Existen otros parámetros que indican el rendimiento de un servidor web, pero requieren la instalación de software adicional en el servidor. El prototipo desarrollado no soporta la instalación de ningún software para tales fines.

El dispositivo desarrollado fue comparado con uno comercial basado en la tarjeta de desarrollo Arduino Uno y la tarjeta controladora de Ethernet ENC28J60 fundamentándose en la coincidencia de algunos parámetros técnicos.

Las pruebas fueron realizadas en el data center de la Universidad Autónoma de los Andes UNIDEC, utilizando únicamente medios de transmisión cableados para efectuar los procedimientos.

El análisis estadístico de los datos obtenidos se lo realizó con el método de “t” de Student, porque permite matemáticamente determinar diferencias significativas entre los promedios de dos grupos.

4.1 RESULTADOS OBTENIDOS

En la tablas 5 se muestran los resultados obtenidos utilizando cable utp cat 5e para conectar los servidores embebidos a la red. Los datos de cada tabla fueron tomados en dos periodos de tiempo distintos del día para asegurar la fiabilidad de la información recolectada.

TABLA NO. 5 DATOS MICRO WEB SERVER VS. DISPOSITIVO COMERCIAL

REPETICIONES	DISPOSITIVO COMERCIAL				MICRO WEB SERVER			
	INTENTO 1		INTENTO 2		INTENTO 1		INTENTO 2	
	TIEMPO DE CARGA	PING	TIEMPO DE CARGA	PING	TIEMPO DE CARGA	PING	TIEMPO DE CARGA	PING
R1	12,98	164	12,9	140	13,05	164	13,01	140
R2	12,77	164	12,41	130	12,90	164	12,45	138
R3	12,50	161	11,9	135	12,66	166	11,5	138
R4	12,35	164	11,35	130	12,69	166	11,46	135
R5	12,05	164	11,15	138	12,15	166	11,17	140

Con los datos obtenidos se hace una primera aproximación para determinar si las muestras son significativamente diferentes. Para este propósito se calcula:

- Promedio de cada grupo de datos.
- Desviación estándar de cada grupo de datos.
- Se construyeron intervalos de confianza alrededor de cada grupo de datos esto es $PROMEDIO \pm DESVIACIÓN ESTÁNDAR$.
- Se calculó la diferencia entre los intervalos de confianza para cada grupo de datos.

En la tabla 6 se muestran los resultados obtenidos.

TABLA No. 1 PRIMERA APROXIMACIÓN PARA DETERMINAR SIGNIFICANCIA

	DISPOSITIVO COMERCIAL				MICRO WEB SERVER			
	INTENTO 1		INTENTO 2		INTENTO 1		INTENTO 2	
	TIEMPO DE CARGA	PING	TIEMPO DE CARGA	PING	TIEMPO DE CARGA	PING	TIEMPO DE CARGA	PING
PROMEDIO	12,53	163,4	11,942	138	12,69	165,2	11,918	140
DESVIACIÓN ESTÁNDAR	0,362	1,342	0,727	4,561	0,341	1,095	0,778	2,049
INTERVALO DE CONFIANZA(PROMEDIO \pm DESVIACIÓN ESTÁNDAR)	12,168	162,058	11,215	133,439	12,349	164,105	11,140	137,951
	12,892	164,742	12,669	142,561	13,031	166,295	12,696	142,049
DIFERENCIA ENTRE INTERVALOS DE CONFIANZA	-0,724	-2,683	-1,455	-9,121	-0,683	-2,191	-1,555	-4,099

El parámetro a tomar en cuenta fue DIFERENCIA ENTRE INTERVALOS DE CONFIANZA. Para el intento dos hay una posible significancia en las pruebas de ping, resultado que debió ser procesado estadísticamente. Para demostrar matemáticamente los resultados obtenidos se aplicó “t “de Student asumiendo varianzas iguales, el proceso se muestra en la Tabla 7.

TABLA No. 2 “t”de STUDENT

	INTENTO 1				INTENTO 2			
	TIEMPO DE CARGA		PING		TIEMPO DE CARGA		PING	
	V 1	V 2	V 1	V 2	V 1	V 2	V 1	V 2
MED	12,53	12,69	165,2	12,69	11,942	11,918	134,600	138,200
VAR	0,131	0,117	1,2	0,117	0,529	0,605	20,800	4,200
OBS	5	5	5	5	5	5	5	5
V A	0,124		0,658		0,567		12,500	
DIF H₀ MED	0		0		0		0	
G L	8		8		8		8	
EST. t	-0,719		297,211		0,050		-1,610	
P(T<=t) UNA COLA	0,246		0		0,481		0,073	
VC t (UNA COLA)	1,860		1,860		1,860		1,860	
P(T<=t) DOS COLAS	0,493		0		0,961		0,146	
V C t (DOS COLAS)	2,306		2,306		2,306		2,306	

Para interpretar los resultados de la tabla No. 3 se analizaron los valores P(T<=t) UNA COLA, este parámetro es una probabilidad. Indica significancia o no de los promedios de dos grupos de datos con el siguiente criterio:

$$P(T \leq t) \text{ UNA COLA} < 0,05$$

Cuando esta condición se cumple hay una diferencia significativa entre el grupo de datos comparados. En la tabla 8 se muestran los resultados obtenidos para este parámetro con su respectiva comparación.

TABLA No. 4 ANÁLISIS DE SIGNIFICANCIA

	INTENTO 1		INTENTO 2	
	TIEMPO DE CARGA	PING	TIEMPO DE CARGA	PING
P(T<=t) UNA COLA	0,246	0	0,481	0,073
SIGNIFICANCIA	ns	ns	ns	ns

En el Gráfico uno y dos se muestran los resultados obtenidos bajo los criterios de comparación propuestos. Tanto la Conectividad y Tiempo de carga.

GRÁFICO No. 1 TIEMPOS DE CARGA DEL DISPOSITIVO COMERCIAL VS. MICRO WEB SERVER

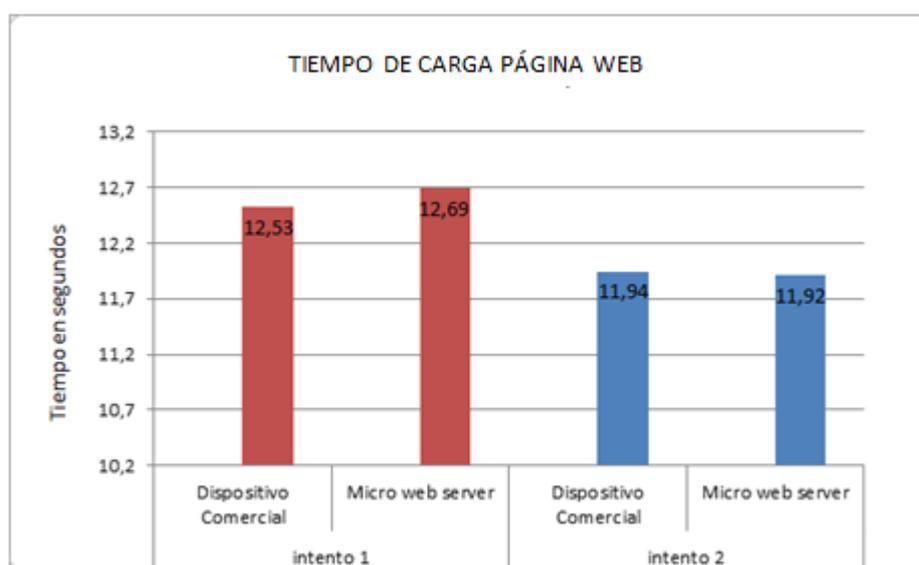
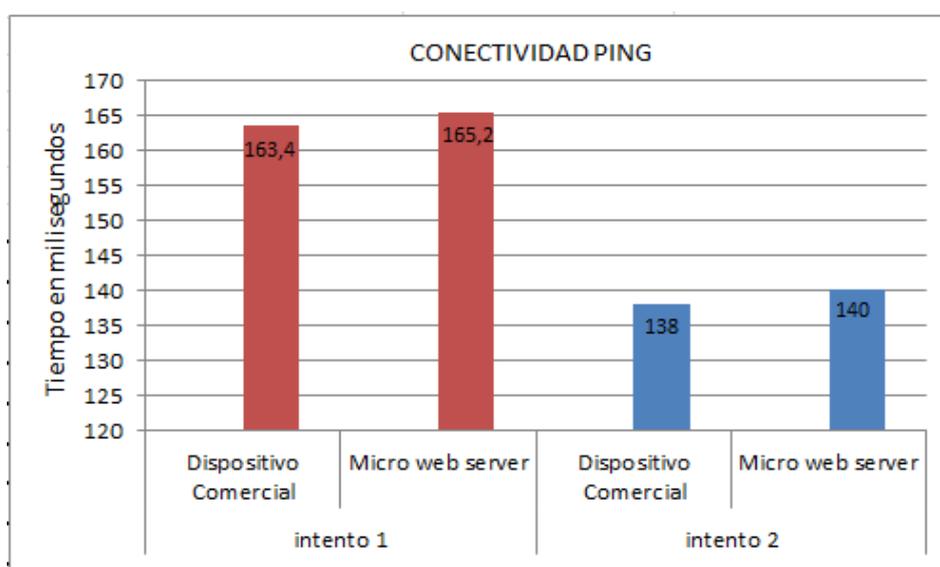


GRÁFICO No. 2 CONECTIVIDAD DEL DISPOSITIVO COMERCIAL VS. MICRO WEB SERVER



CAPÍTULO V

5 DISCUSIÓN

Al iniciar el análisis estadístico se encontró un parámetro llamado diferencia entre intervalos de confianza tabla No. 6, comparando los valores hallados para el tiempo de carga en el intento uno de cada dispositivo prácticamente los valores se superponen. Interpretándose como poca diferencia significativa entre los promedios de este grupo de datos. Realizando la misma comparación para el intento dos el resultado es similar poca diferencia significativa.

Para el parámetro ping en el intento uno el resultado sigue siendo poco significativo, pero en el intento dos los valores no se superponen, habiendo una posibilidad de significancia entre los datos procesados. El método estadístico aplicado fue “t” de student, se asumió varianzas iguales porque la diferencia entre intervalos de confianza sugiere poca diferencia significativa entre los promedios del grupo de datos.

En la tabla No. 5 se muestran los resultados del “t” de student, el principal parámetro considerado fue $P(T \leq t)$ UNA COLA. Cuando este valor es menor que 0,05 los promedios de los grupos de datos analizados son significativamente diferentes. Al revisar los valores obtenidos ninguno llegó a ser menor que 0,05, interpretándose como poca diferencia significativa entre los dos servidores embebidos.

Para aclarar el término poca diferencia significativa en el caso de los servidores analizados, se puede decir que la micro web server si bien no supera a un dispositivo comercial existente fue capaz de desempeñarse de manera similar. Las pruebas y resultados obtenidos se efectuaron en redes locales usando como medio de transmisión cable utp cat 5e.

CAPÍTULO VI

6 CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Basándose en las especificaciones de diseño del dispositivo los componentes elegidos para la micro web server fueron el ATMEGA328P y el Arduino Ethernet Shield, dando como resultado un dispositivo eficiente y de bajo costo.

El micro-controlador ATMEGA328P se conectó a la red utilizando una tarjeta controladora externa, la comunicación con este periférico se implementó mediante el protocolo SPI.

Se construyó una fuente de alimentación de cinco voltios utilizada conjuntamente por el ATMEGA328P y el módulo Arduino Ethernet Shield, enfocada en utilizar el menor espacio posible.

La programación del micro-controlador se realizó en lenguaje C, utilizando la herramienta AVR studio 6 distribuida libremente por ATMEL, el código de programación generado ocupó poco espacio en la memoria ROM del micro-controlador.

La interfaz web se desarrolló en lenguaje html usando como editor de texto el bloc de notas de Windows. Debido al tamaño de la página web se utilizó una tarjeta sd como dispositivo de almacenamiento externo.

Las señales analógicas que fueron aplicadas al dispositivo fueron tomadas de potenciómetros. Las señales digitales de pulsadores y diodos emisores de luz, para estas condiciones el equipo funcionó adecuadamente.

Se utilizó “t” de student aplicado a los promedios de los datos proporcionados por el comando ping y el tiempo de carga de la página web. Comparados con los valores obtenidos en un dispositivo comercial, se determinó matemáticamente que la micro web server funcionó de manera similar a un equipo existente en el mercado.

6.2 RECOMENDACIONES

Para mostrar páginas web de mayor complejidad se recomienda la implementación del servidor embebido con un micro-controlador con mayor capacidad de almacenamiento interno.

Con la finalidad de modificar rápidamente el software que hace trabajar al dispositivo es útil construir un módulo programador capaz de conectarse directamente a la placa del servidor embebido.

El equipo fue probado con medios cableados de conexión a la red. El desempeño y la versatilidad del servidor pueden ser mejorados al agregar capacidades inalámbricas de conexión a la red, utilizando módulos compatibles con el hardware existente.

Se puede lograr construir un equipo más compacto implementado la parte concerniente a conectividad con la red dentro del mismo micro-controlador que se desempeña como servidor. Esta modificación necesariamente requiere el cambio del micro-controlador utilizado.

El acceso al contenido de la página web no está restringido, considerando la importancia de la información que se muestra en la interfaz web se pueden implementar mecanismos de verificación y autenticación de usuarios, característica que se añade en la programación HTML de la página web.

Las señales procesadas fueron tomadas de pulsadores, diodos y potenciómetros, se recomienda poner a prueba el prototipo con señales de audio, temperatura, humedad. Esto para determinar un punto máximo de rendimiento del servidor, a partir del cual se puede considerar modificar componentes electrónicos en base a las nuevas exigencias propuestas.

CAPÍTULO VII

7 PROPUESTA

7.1 TÍTULO DE LA PROPUESTA

Diseño y Construcción de un servidor web embebido para monitoreo y control de procesos industriales con acceso remoto.

7.2 INTRODUCCIÓN

El servidor web embebido consta de 2 componentes, hardware y software, en donde el hardware se basa en un micro-controladores con capacidades de conexión a internet, mientras que el software se desarrolla en lenguaje C.

El dispositivo propone una solución de bajo costo, además permite el monitoreo y control de procesos desde cualquier lugar con acceso a internet. Lo que disminuye el tiempo de respuesta ante un suceso inesperado. La elección de micro-controladores y demás componentes fundamentales para el servidor web se sustenta en la gran variedad de aplicaciones que pueden ser desarrolladas con los mencionados elementos.

7.3 OBJETIVOS

7.3.1 OBJETIVO GENERAL

- Diseñar y construir un servidor web embebido para monitorear y controlar procesos industriales con acceso remoto.

7.3.2 OBJETIVOS ESPECÍFICOS

- Establecer los requerimientos del sistema.
- Elección de componentes electrónicos y entorno de programación.
- Construcción del hardware.
- Programación y pruebas.

7.4 FUNDAMENTACIÓN CIENTÍFICO – TÉCNICA

El networking con micro-controladores amplía las posibilidades para gestionar procesos remotamente. Existen variedad de fabricantes de circuitos integrados que desarrollan dispositivos con soporte para TCP/IP, IEEE 802.3, HTTP, UDP, etc.

Entre los micro-controladores con estas características se puede mencionar:

FABRICANTE	CHIP
ATMEL	AVR32817
MICROCHIP	PIC18F87J60
WIZNET	w5100

Para la construcción de este tipo de sistemas embebidos se puede implementar todo lo concerniente a interfaz con la red dentro del mismo dispositivo que hace de servidor.

Otra opción es utilizar tarjetas externas que gestionan la conectividad con la red, para comunicarse con el dispositivo que se desempeña como servidor generalmente se utiliza SPI. El acceso remoto puede realizarse con una IP pública configurada en el servidor embebido. Otra alternativa es enlazar el dispositivo con un servicio DNS dinámico para poder acceder al servidor sin disponer de una IP pública.

7.5 DESCRIPCIÓN DE LA PROPUESTA

Se propone diseñar y construir un sistema embebido capaz de monitorear y controlar procesos industriales de manera remota, empleando el internet como plataforma de comunicación y la interfaz con el usuario una página web. Para cumplir con los objetivos propuestos el sistema a construirse debe considerar las siguientes especificaciones de diseño.

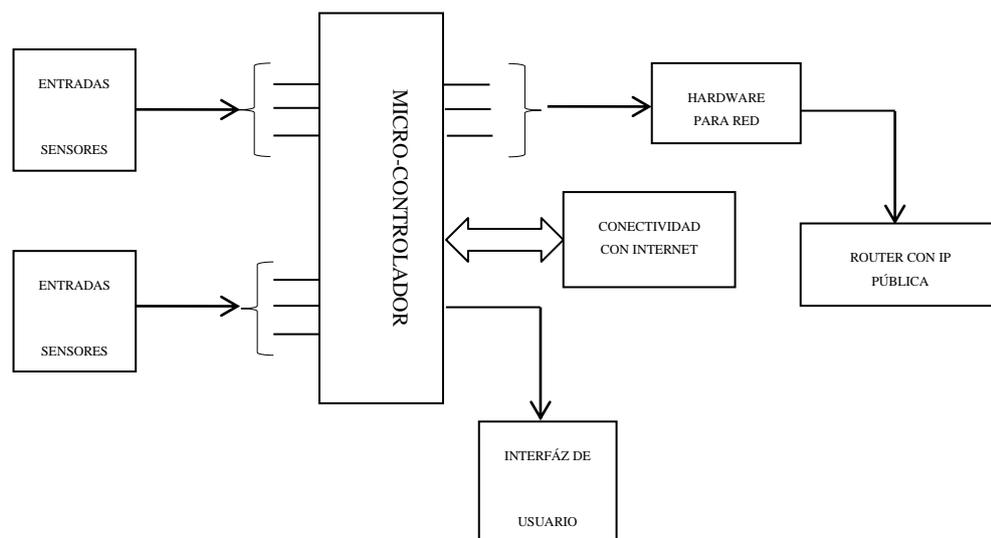
ESPECIFICACIONES

- Interfaz de página web.
- Accesible mediante dirección IP y puerto, de manera remota.
- Monitorea y controla procesos industriales.
- Eficiente y fiable.

Las especificaciones propuestas requieren la utilización de micro-controladores de por lo menos 16 bits, capaces de procesar señales de alta complejidad y con soporte para Ethernet. Para acceder al servidor se propone utilizar una IP pública.

DIAGRAMA DE BLOQUES

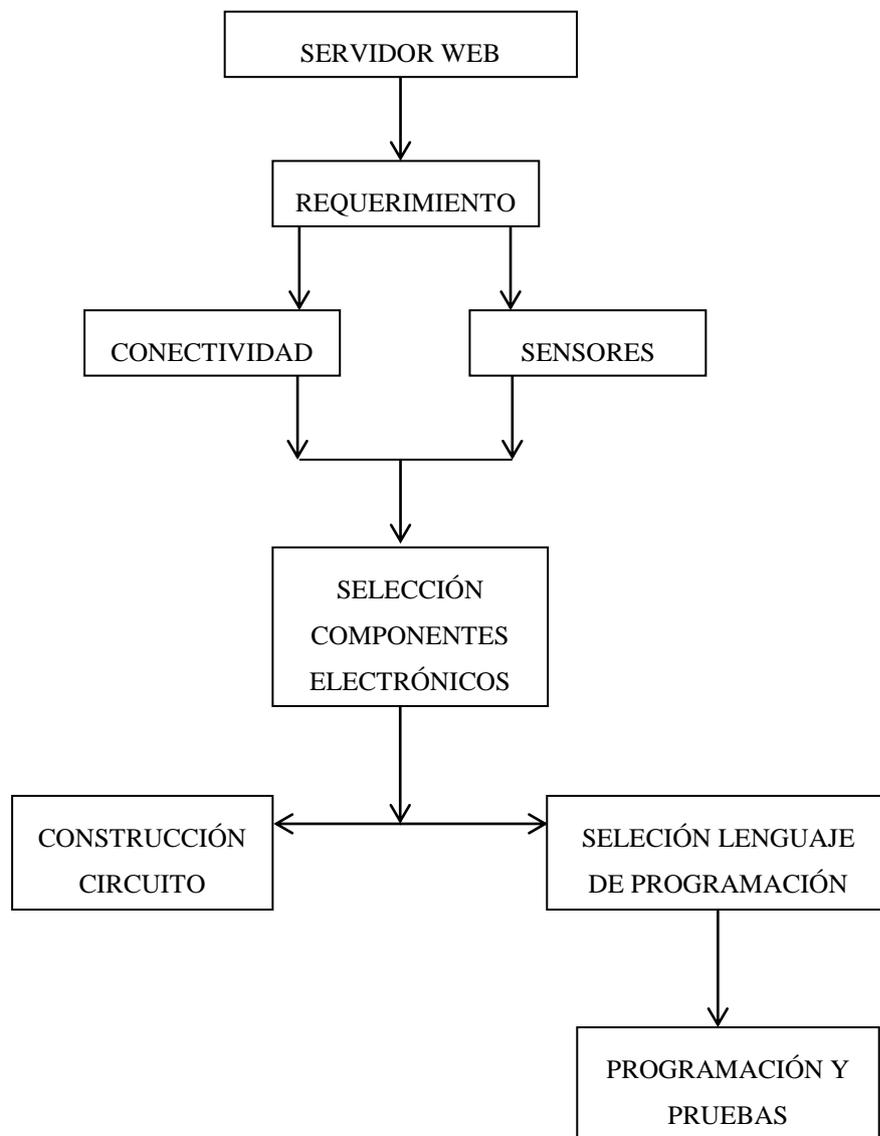
El problema se divide en cuatro partes. En primer lugar se considera el procesamiento de los datos obtenidos por los sensores en el micro-controlador, mediante programación se implementa los protocolos y procedimientos para enlazar el servidor a internet. Se necesita también construir un hardware con el propósito de funcionar como tarjeta de red, la interfaz para el usuario es una página web almacenada en el servidor.



Con estos requerimientos se procede a elegir los componentes electrónicos adecuados para construir el sistema. El entorno de programación propuesto es lenguaje C aplicado a micro-controladores.

El tiempo considerado para llegar a tener un servidor web con las características especificadas se estima en 8 meses.

7.6 DISEÑO ORGANIZACIONAL



7.7 MONITOREO Y EVALUACIÓN DE LA PROPUESTA

El acceso al servidor web se lo puede hacer desde cualquier lugar con servicio de internet. Para lograr este propósito se deben considerar los siguientes parámetros adicionales:

- Contratar un servicio de IP pública, es el método más utilizado para acceder a los recursos de un servidor.
- Servicio de DNS dinámico, implica contratar un recurso externo que actualiza la IP de acceso al servidor cuando la dirección IP del router correspondiente cambia.

La sugerencia es contratar una IP pública, es más eficiente y seguro que contratar un recurso externo para resolver la dirección IP que permita acceder al servidor. Con el objetivo de dar conectividad al dispositivo se disponen de las siguientes posibilidades:

- Tarjeta controladora externa de Ethernet, implementa por hardware todos los requerimientos necesarios para conectar a internet el servidor. Los protocolos principales que debe soportar son TCP, HTTP, ICMP, como consideraciones adicionales se sugiere que sea de plataforma abierta para que pueda ser modificada de acuerdo a las necesidades del servidor web y que soporte el protocolo HTTPS para dar mayor seguridad al acceso de usuarios.
- Implementar por software los mecanismos necesarios para conectar a internet el dispositivo, esta programación se desarrolla en el micro-controlador que se desempeña como servidor. Como resultado de utilizar este procedimiento el prototipo es más compacto y el programador tiene control total de los protocolos y servicios relacionados con la conectividad hacia internet.

CAPÍTULO VIII

8 BIBLIOGRAFÍA

Abel, P. (1996). *Lenguaje ensamblador y programación para pc ibm y compatibles*. México, D.F.: Prentice Hall Inc.

Ambadar, A. (2002). *Procesamiento de señales analógicas y digitales*. México, D. F.: Thomson Editores, S. A.

Arduino. (2015). *Arduino Ethernet Shield*. Obtenido de Arduino: <http://store.arduino.cc/product/A000066>

Arduino. (s.f.). *ARDUINO ETH SHIELD 06 - Rev 3*. Obtenido de <http://www.arduino.cc/en/uploads/Main/arduino-ethernet-shield-06-schematic.pdf>

Atelin, P., & Dordoigne, J. (2007). *TCP/IP y protocolos de internet*. Barcelona: Eni Ediciones.

Atmel, C. (10 de 2014). *ATmega 48A/PA/88A/PA*. Obtenido de ATMEL: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

Bernál, J. L. (S. F.). *Curso de html*. Obtenido de J. L. HTML: : <http://asodifimo.org.do/images/nomina.pdf>

Black, U. (2002). *Redes de transmisión de datos y proceso distribuído*. Madrid: Prentice-Hall.

Boylestad, R. L. (1997). *Electrónica: teoría de circuitos*. México, D. F.: Prentice Hall Hispanoamérica S.A.

- Castro, Lechtaler, y Fusario, A. (1999). *Teleinformática para ingenieros en sistemas de información*. Barcelona: Reverté. S.A.
- Colobram, M., Arqués, J., & Galindo, M. (2008). *Administración de sistemas operativos en red*. Barcelona: UOC.
- Comer, D. E. (1996). *Redes globales de información con internet y tcp/ip*. México, D.F.: Prentice Hall Inc.
- de León, I. (10 de 11 de 2013). *Mejorar la velocidad de carga de un sitio*. Obtenido de UnosTips: <http://www.unostips.com/mejorar-velocidad-wordpress/>
- DeVore, J. (2005). *Probabilidad y estadística para ingeniería y ciencias*. México, D. F.: International Thompson Editores.
- España, M. (2003). *Servicios Avanzados de Telecomunicación*. Madrid: Díaz de Santos.
- González Palacio Liliana y Urrego Giraldo Germán. (4 de 10 de 2008). *Modelo de requisitos para sistemas embebidos*. Obtenido de Revista Ingenierías Universidad Medellín: <http://www.redalyc.org/articulo.oa?id=75071308>
- Herrera, J. (2003). *Tecnologías y redes de transmisión de datos*. México: Limusa.
- Ibeas, Á., Díaz, J., & De la Hoz, D. (2000). *Nuevas tecnologías de la información*. Editan: Barcelona.
- Knudsen, B. (2011). *C compiler for the picmicro devices version 3.2 user's manual*. Obtenido de B Knudsen Data: <http://www.bknd.com/>
- Kuo, B. C. (1996). *Sistemas de control automático*. México, D.F.: Prentice Hall Inc.

López Pérez, E. (S. F.). *Ingeniería en microcontroladores protocolo spi(serial peripheral interface)*. Obtenido de Ingeniería en microcontroladores:
<http://www.i-micro.com/pdf/articulos/spi.pdf>

Reyes, C. A. (2006). *Microcontroladores pic programacion en basic*. Quito.

Romero, B., Martín, M., Quirós, S., & Robredo, R. (1993). *Microelectrónica, Tecnologías, diseño, aplicaciones*. Cantabria: Universidad de Cantabria.

Sánchez Vitores, R. (10 de 4 de 2004). *Aplicaciones de los sistemas embebidos*. Obtenido de Innovación:
<http://www.tecnicaindustrial.es/tiadmin/numeros/15/07/a07.pdf>

Stallings, W. (2004). *Comunicaciones y redes de computadores*. México, D.F.: Pearson Prentice Hall.

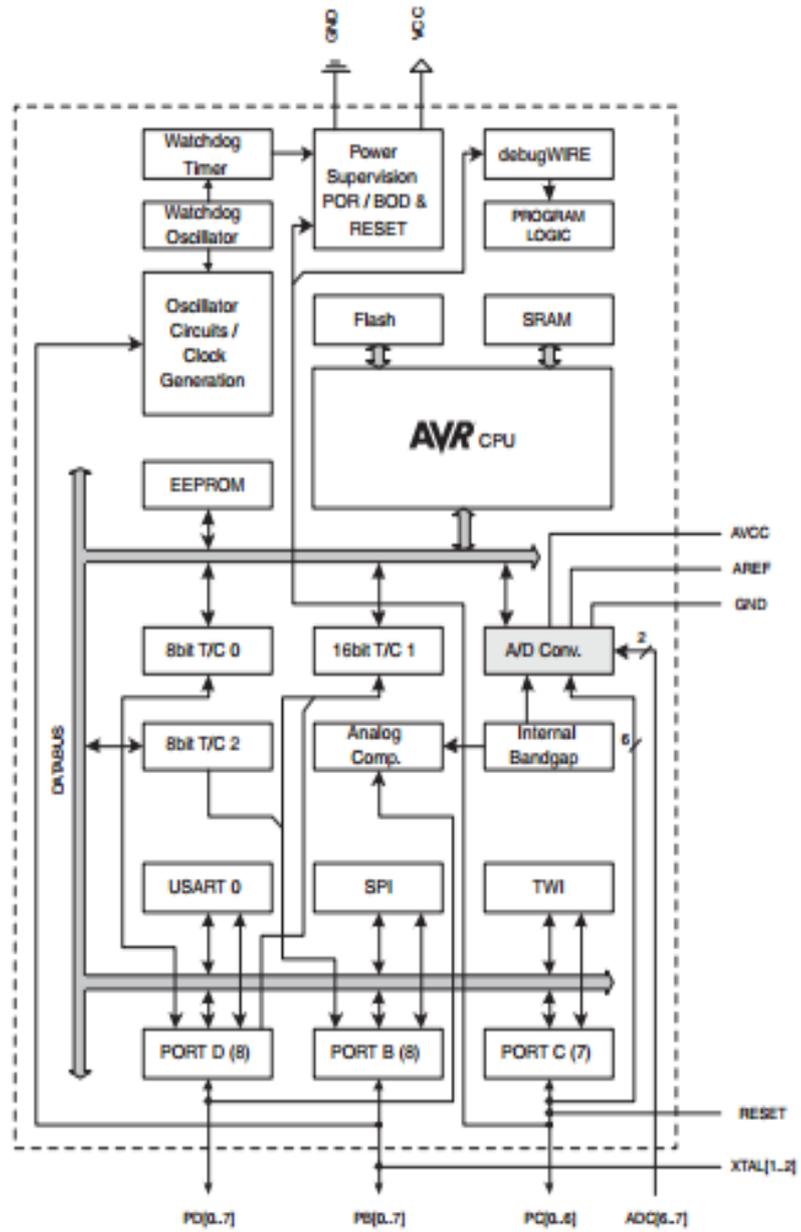
Texas, I. (5 de 2003). *uA7800 SERIES POSITIVE-VOLTAGE REGULATORS*. Obtenido de Texas Instruments:
<https://www.sparkfun.com/datasheets/Components/LM7805.pdf>

Vaqué, Urbaneja, J. (2011). *Componentes mecánicos*. Obtenido de old8bits:
http://old8bits.blogspot.com/2011_04_01_archive.html

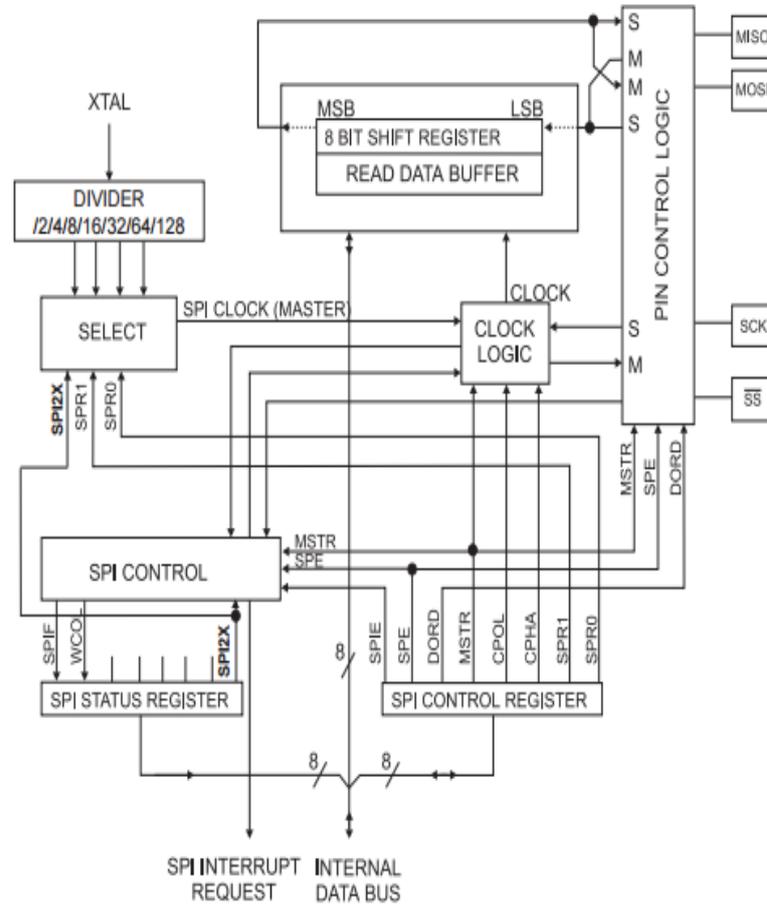
WIZnet Co. (2008). *W5100 Datasheet*. Obtenido de W5100 Datasheet Version 1.1.6:
https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf

ANEXOS

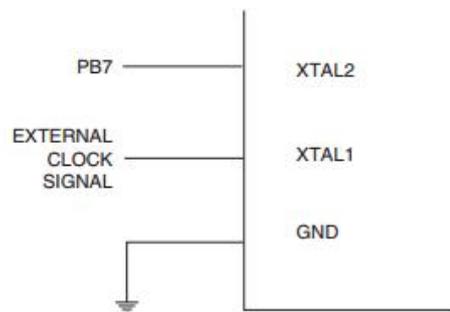
ANEXO No. 1 ATMEGA328P-DIAGRAMA DE BLOQUES Fuente: (Atmel, 2014).



ANEXO No. 2 SPI (SERIAL PERIPHERAL INTERFACE) - DIAGRAMA DE BLOQUES Fuente: (Atmel, 2014).

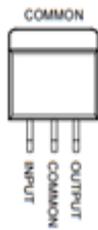


ANEXO No. 3 OSCILADOR EXTERNO Fuente: (Atmel, 2014).



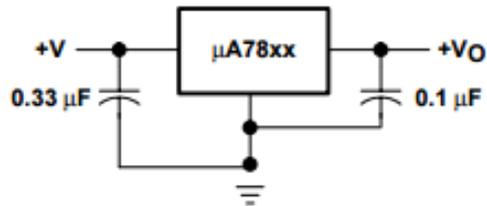
Frequency	CKSEL3...0
0 - 20MHz	0000

ANEXO No. 4 L7805CV PINES Y CONDICIONES DE TRABAJO Fuente: (Texas, 2003)

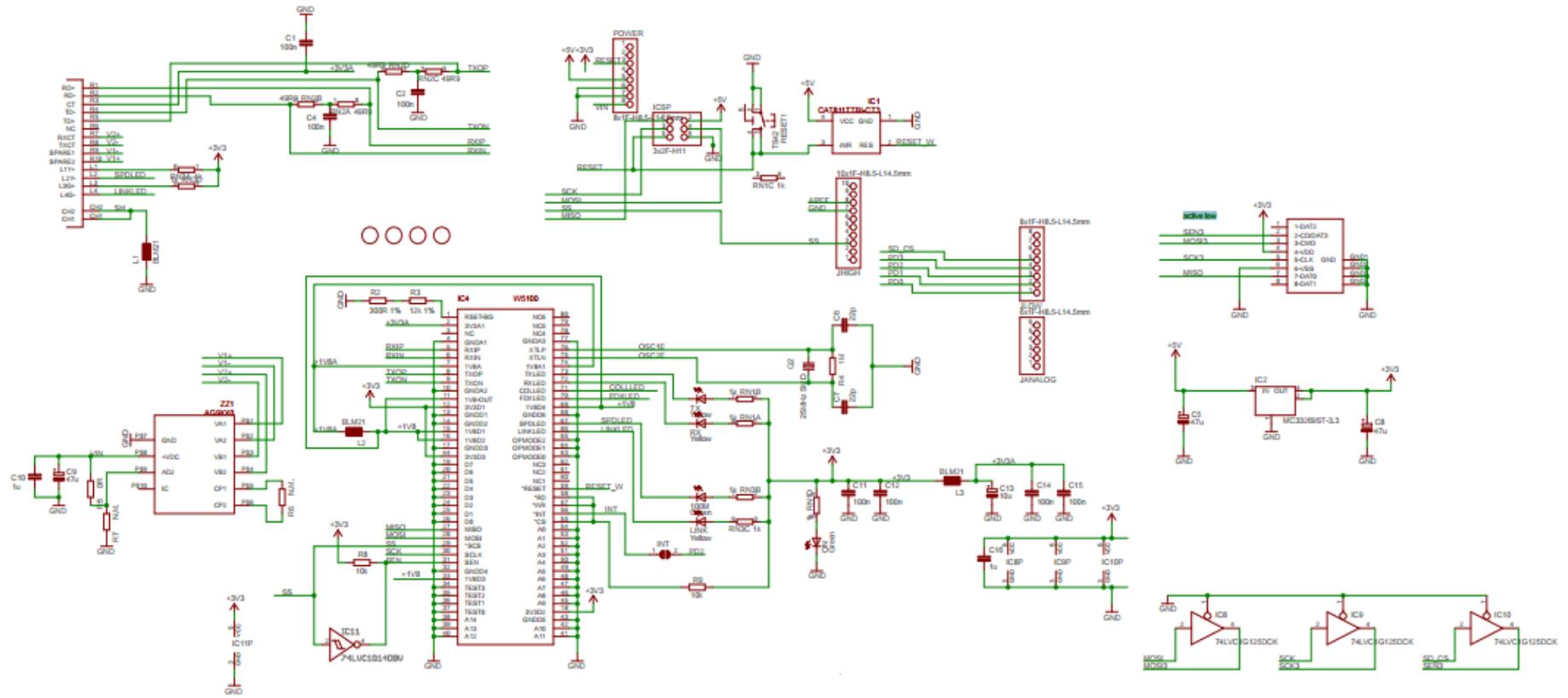


		MIN	MAX	UNIT
V_I Input voltage	$\mu A7805C$	7	25	V
	$\mu A7808C$	10.5	25	
	$\mu A7810C$	12.5	28	
	$\mu A7812C$	14.5	30	
	$\mu A7815C$	17.5	30	
	$\mu A7824C$	27	38	
I_O Output current			1.5	A
T_J Operating virtual junction temperature	$\mu A7800C$ series	0	125	$^{\circ}C$

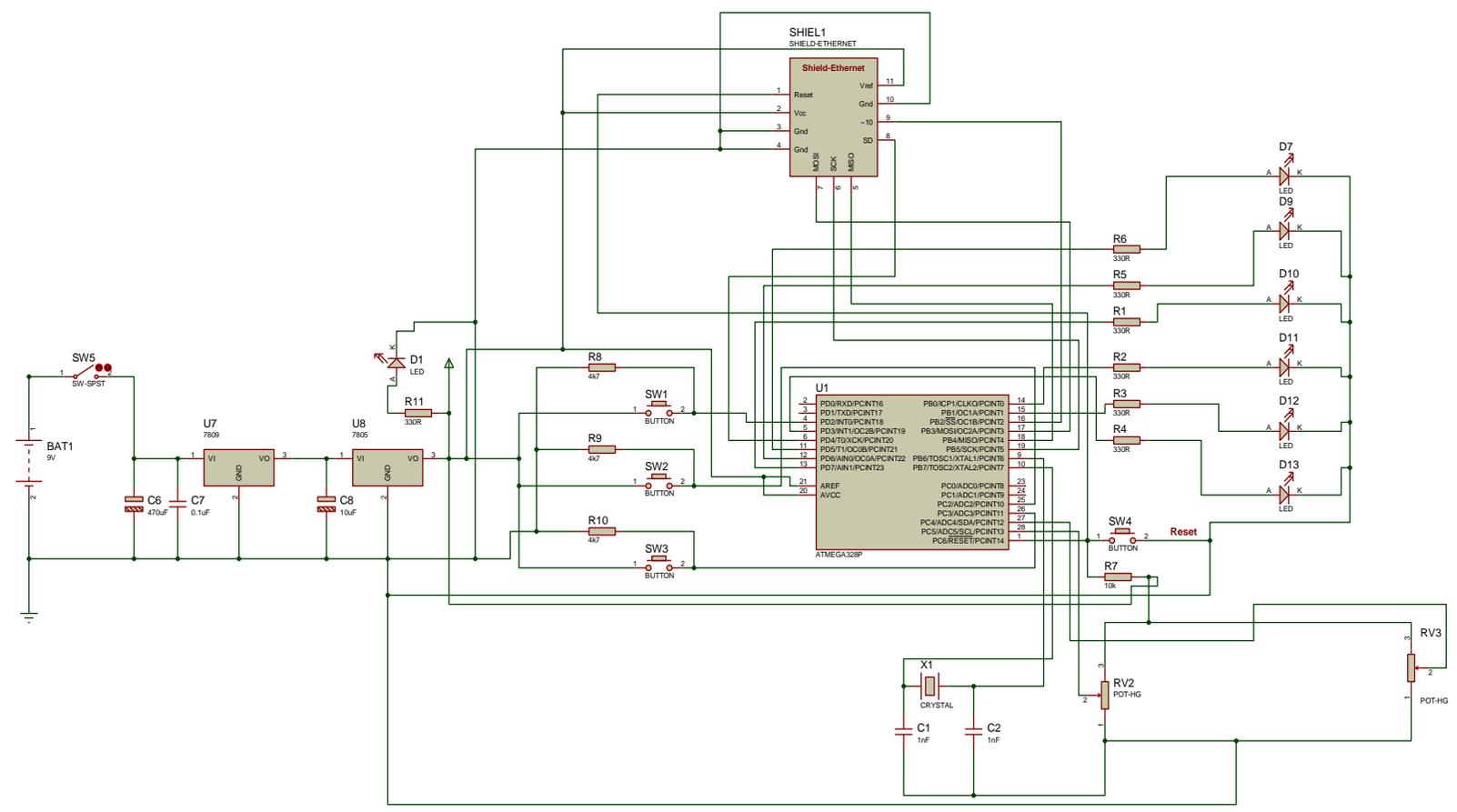
ANEXO No. 5 CONFIGURACIÓN TÍPICA Fuente: (Texas, 2003).



ANEXO No. 6 ARDUINO ETHERNET SHIELD DIAGRAMA ESQUEMÁTICO Fuente: (Arduino, ARDUINO ETH SHIELD 06 - Rev 3).



ANEXO No. 7 MICRO WEB SERVER DIAGRAMA ESQUEMÁTICO



ANEXO No. 8 PROGRAMA MICRO WEB SERVER

```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
#define REQ_BUF_SZ 60
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x42, 0x3E };
IPAddress ip(192,168,1,20);
EthernetServer server(8080);
char HTTP_req[REQ_BUF_SZ] = {0};
boolean LED_state[6] = {0};
void setup()
{
    pinMode(10, OUTPUT);
    digitalWrite(10, HIGH);
    Serial.begin(9600);
    if (!SD.begin(4)) {
        return;
    }
    return;
}
pinMode(2, INPUT);
pinMode(A2, INPUT);
pinMode(A3, INPUT);
pinMode(3, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
Ethernet.begin(mac, ip);
server.begin();
}
void loop()
{
    EthernetClient client = server.available();
    if (client) {
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {
```

```

char c = client.read();
if (req_index < (REQ_BUF_SZ - 1)) {
    HTTP_req[req_index] = c;
    req_index++;
}
if (c == '\n' && currentLineIsBlank) {
    if (StrContains(HTTP_req, "ajax_inputs")) {
        client.println("Content-Type: text/xml");
        client.println();
        SetLEDs();
        XML_response(client);
    } else

    {
        client.println();
        webFile = SD.open("index.htm");
        if (webFile) {
            while(webFile.available()) {
                client.write(webFile.read());
            }
            webFile.close();
        }
    }
    Serial.print(HTTP_req);
    req_index = 0;
    StrClear(HTTP_req, REQ_BUF_SZ);
    break;
}
if (c == '\n') {
    currentLineIsBlank = true;
}
else if (c != '\r') {
    currentLineIsBlank = false;
}
}
}
client.stop();

```

```

    }
}
void SetLEDs(void)
{
    if (StrContains(HTTP_req, "LED1=1")) {
        LED_state[0] = 1;
        digitalWrite(6, HIGH);
    }
    else if (StrContains(HTTP_req, "LED1=0")) {
        LED_state[0] = 0;
        digitalWrite(6, LOW);
    }
    if (StrContains(HTTP_req, "LED2=1")) {
        LED_state[1] = 1;
        digitalWrite(7, HIGH);
    }
    else if (StrContains(HTTP_req, "LED2=0")) {
        LED_state[1] = 0;
        digitalWrite(7, LOW);
    }
    if (StrContains(HTTP_req, "LED3=1")) {
        LED_state[2] = 1;
        digitalWrite(8, HIGH);
    }
    else if (StrContains(HTTP_req, "LED3=0")) {
        LED_state[2] = 0;
        digitalWrite(8, LOW);
    }
    if (StrContains(HTTP_req, "LED4=1")) {
        LED_state[3] = 1;
        digitalWrite(9, HIGH);
    }
    else if (StrContains(HTTP_req, "LED4=0")) {
        LED_state[3] = 0;
        digitalWrite(9, LOW);
    }
    if (StrContains(HTTP_req, "LED5=1")) {

```

```

    LED_state[4] = 1;
    digitalWrite(5, HIGH);
}
else if (StrContains(HTTP_req, "LED5=0")) {
    LED_state[4] = 0;
    digitalWrite(5, LOW);
}
if (StrContains(HTTP_req, "LED6=1")) {
    LED_state[5] = 1;
    digitalWrite(3, HIGH);
}
else if (StrContains(HTTP_req, "LED6=0")) {
    LED_state[5] = 0;
    digitalWrite(3, LOW);
}
}
void XML_response(EthernetClient cl)
{
    cl.println("<?xml version = \"1.0\" ?>");
    cl.println("<inputs>");
    cl.print("<analog>");
    cl.println("</analog>");
    cl.print(analogRead(5));
    cl.println("</analog>");
    cl.print("<switch>");
    if (digitalRead(2)) {cl.print("ON");} else {cl.print("OFF");}
    cl.println("</switch>");
    cl.print("<switch>");
    if (digitalRead(A2)) {cl.print("ON");} else {cl.print("OFF");}
    cl.print("<switch>");
    cl.println("</switch>");
    if (LED_state[0]) {cl.print("checked");} else {cl.print("unchecked");}
    cl.println("</LED>");
    cl.print("<LED>");
    cl.println("</LED>");
    cl.print("<LED>");
    if (LED_state[2]) {cl.print("on");} else {cl.print("off");}

```

```

        cl.println("</LED>");
        cl.print("<LED>");
        if (LED_state[3]) {cl.print("on");} else {cl.print("off");}
        cl.print("<LED>");
        if (LED_state[4]) {cl.print("on");} else {cl.print("off");}
        cl.println("</LED>");
        if (LED_state[5]) {cl.print("on");} else {cl.print("off");}
        cl.println("</LED>");
        cl.print("</inputs>");
    }
void StrClear(char *str, char length)
{
    for (int i = 0; i < length; i++) {
        str[i] = 0;
    }
}
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char len;
    len = strlen(str);
    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }
    return 0;
}

```

ANEXO No. 9 PROGRAMA PAGINA WEB

```
<!DOCTYPE html>
<html>
  <head>
    <title>MICRO WEB SERVER</title>

    <script>

        strLED1 = "";
        strLED2 = "";
        strLED3 = "";
        strLED6 = "";
        var LED3_state = 0;
        var LED4_state = 0;
        var LED5_state = 0;
        var data_val2 = 0;
        var pulse_width = 1500;
        function GetArduinoIO()
        {
            nocache = "&nocache=" + Math.random() * 1000000;
            var request = new XMLHttpRequest();
            {
                if (this.readyState == 4) {
                    if (this.status == 200) {
                        if (this.responseXML != null) {

                            this.responseXML.getElementsByTagName('analog')[0].childNodes[0].nodeValue;
                                                                    data_val1 =

                            document.getElementsByClassName("analog")[1].innerHTML =

                            this.responseXML.getElementsByTagName('analog')[1].childNodes[0].nodeValue;
                                                                    data_val2 =

                                                                    var num_an =
                                                                    for (count = 0; count <
num_an; count++) {
```

```

document.getElementsByClassName("switches")[count].innerHTML =

this.responseXML.getElementsByTagName('switch')[count].childNodes[0].nodeValue;
        }
        if
(this.responseXML.getElementsByTagName('LED')[0].childNodes[0].nodeValue === "checked") {

        }
        else {

        }
        if
(this.responseXML.getElementsByTagName('LED')[1].childNodes[0].nodeValue === "checked") {

document.LED_form.LED2.checked = true;

        }
        else {

document.LED_form.LED2.checked = false;

        }
        if

document.getElementById("LED3").innerHTML = "Send";

LED3_state = 1;

        }
        else {

document.getElementById("LED3").innerHTML = "Gate";

LED3_state = 0;

        }
        if
(this.responseXML.getElementsByTagName('LED')[3].childNodes[0].nodeValue === "on") {

document.getElementById("LED4").innerHTML = "Send";

LED4_state = 1;

        }
        else {

document.getElementById("LED4").innerHTML = "Garage";

LED4_state = 0;

```

```

    }

    if
(this.responseXML.getElementsByTagName('LED')[4].childNodes[0].nodeValue === "on") {

    document.getElementById("LED5").innerHTML = "Send";

    }
    else {

    document.getElementById("LED5").innerHTML = "Panic ADT";

    }

    if
(this.responseXML.getElementsByTagName('LED')[5].childNodes[0].nodeValue === "on") {

    document.getElementById("LED6").innerHTML = "Send";

    LED6_state = 1;

    }
    else {

    document.getElementById("LED6").innerHTML = "Alarm";

    LED6_state = 0;

    }

    }

}

request.open("GET", "ajax_inputs" + strLED1 + strLED2 + strLED3
request.send(null);
setTimeout('GetArduinoIO()', 500);
strLED4 = "";
strLED5 = "";
strLED6 = "";
}
function GetCheck()
{
    if (LED_form.LED1.checked) {
        strLED1 = "&LED1=1";
    }
}

```

```

else {
    strLED1 = "&LED1=0";
}
if (LED_form.LED2.checked) {
}
else {
    strLED2 = "&LED2=0";
}
}
function GetButton1()
{
    LED3_state = 1;
    strLED3 = "&LED3=1";
    strLED3 = "&LED3=0";
    }, pulse_width);
}
function GetButton2()
{
    LED4_state = 1;
    strLED4 = "&LED4=1";
    window.setTimeout(function resetLED4(){
        LED4_state = 0;
    }, pulse_width);
}
function GetButton3()
{
    LED5_state = 1;
    window.setTimeout(function resetLED5(){
        LED5_state = 0;
    }, pulse_width);
}
function GetButton4()
{
    LED6_state = 1;
    strLED6 = "&LED6=1";
    window.setTimeout(function resetLED6(){
        LED6_state = 0;

```

```

        strLED6 = "&LED6=0";
    }, pulse_width);
}
</script>
<style>
* {
}
html, body {
    height: 100%;
.wrapper {
    height: auto !important;
    margin: 0 auto -4em;
}
.footer, .push {
    padding: 13px 0px 13px 0px;
    bottom:0%;
    font-size: 100%;
    color: white;
}
body {
    background:
    radial-gradient(black 15%, transparent 16%) 0 0,
    background-size:16px 16px;
    margin: 0;
}
.IO_box {
    width: 250px;
    background-color:white;
    float: left;
    padding: 0;
}
.skrif {
    vertical-align: -25%;
    color:#383838;
}
.BoxContainer {
    top: 50%;

```

```

        left:60%;
        margin-left: -366px;
    }
    .redbuttonbox {
        margin: 55px 22px 0 22px;
    }
    h1 {
        text-align:center;
        font-size: 120%;
        font:"Courier New", Courier, monospace;
        padding: 0px 0 7px 0;
    }
    .InTouchH1 {
        font-size:220%;
        margin-right:10px;
    }
    .InTouchH2 {
        font-size:180%;
        font-weight: normal;
        color:white;
    }
    h2 {
        text-align: center;
        font-size: 180%;
        border-radius:10px 10px 0px 0px;
        color: White;
        margin: 0 0 1px 0;
    }
    h3 {
        text-align: center;
        background-color:#F94E33;
        padding: 23px 0 23px 0;
        margin: 0 0 10px 0;
    }
    p, form, button {
        color: #252525;
    }

```

```

.small_text {
    font-size: 70%;
    color: #737373;
}

.StatusField {
    padding: 6px;
}

.SwitchesField {
    width:204px;
}

.SignalField {
    width:204px;
}

.FieldLabelText {
    color: grey;
    font-size: 60%;
}

.container > .switch {
    margin: 12px auto;
}

.switch {
    width: 56px;
    height: 18px;
    background-color: white;
    border-radius: 18px;
    background-image: linear-gradient(to bottom, #eeeeee, white 25px);
    margin: 1px 0px 1px 0px;
}

position: absolute;
top: 0;
left: 0;
opacity: 0;
}

display: block;
height: inherit;
font-size: 10px;

```

0.15);

```
border-radius: inherit;
box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.12), inset 0 0 2px rgba(0, 0, 0,
-webkit-transition: 0.15s ease-out;
-webkit-transition-property: opacity background;
-moz-transition-property: opacity background;
transition-property: opacity background;
}
.switch-label:before, .switch-label:after {
position: absolute;
-moz-transition: inherit;
-o-transition: inherit;
transition: inherit;
}
.switch-label:before {
content: attr(data-off);
text-shadow: 0 1px rgba(255, 255, 255, 0.5);
}
.switch-label:after {
content: attr(data-on);
color: white;
text-shadow: 0 1px rgba(0, 0, 0, 0.2);
}
background: #47a8d8;
box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.15), inset 0 0 3px rgba(0, 0, 0,
}
.switch-input:checked ~ .switch-label:before {
opacity: 0;
}
opacity: 1;
}
.switch-handle {
height: 16px;
background-image: -moz-linear-gradient(top, white 40%, #f0f0f0);
background-image: -o-linear-gradient(top, white 40%, #f0f0f0);
background-image: linear-gradient(to bottom, white 40%, #f0f0f0);
-webkit-transition: left 0.15s ease-out;
-moz-transition: left 0.15s ease-out;
```

```

        transition: left 0.15s ease-out;
    }
    .switch-handle:before {
        content: "";
        background-image: -webkit-linear-gradient(top, #eeeeee, white);
    }
    .switch-input:checked ~ .switch-handle {
        left: 40px;
        box-shadow: -1px 1px 5px rgba(0, 0, 0, 0.2);
    }

    .switch-green > .switch-input:checked ~ .switch-label {
        background: #4fb845;
    }

    .button {
        margin: 5px 2px 5px 0px;
        background: black;
        background: -webkit-gradient(linear, left top, left bottom, from(#47a8d8),
to(#24819B));
        background: -webkit-linear-gradient(top, #47a8d8, #24819B);
        padding: 2px 8px;
        -webkit-border-radius: 8px;
        -moz-border-radius: 8px;
        border-radius: 8px;
        color: white;
        font-size: 10px;
        text-decoration: none;
        vertical-align: middle;
    }
    .button:hover {
        border-top-color: #4299d4;
        color: white;
    }
    .button:active {
        background: #24819B;
    }

    .buttonred {

```

```

margin: 5px 2px 5px 0px;
width: 90px; height:90px;
background: black;
background: -webkit-gradient(linear, left top, left bottom, from(#FB674F),
to(#F94E33));
background: -moz-linear-gradient(top, #FB674F, #F94E33);
background: -ms-linear-gradient(top, #FB674F, #F94E33);
-webkit-box-shadow: rgba(1,0,0,0) 0 1px 0;
}
.buttonred:hover {
background: #F94E33;
color: white;
}
.buttonred:active {
border-top-color: #F94E33;
background: #F94E33;
}

.gaugestyle1 {
padding: 5px 0 0 5px;
float:left;
}
.gaugestyle2 {
padding: 0 5px 5px 0;
float:right;
}
.analog{
float: right;
font-size: 1%;
color:white;
}
</style>
</head>
<body onload="GetArduinoIO()">
<div class="wrapper">
<div class="IO_box">
<h2>Controls</h2>

```

```

<fieldset class="SwitchesField">
<legend class="FieldLabelText">Switches</legend>
  <form id="check_LEDs" name="LED_form">
    <label class="switch">
      <input type="checkbox"
name="LED1" value="0" onclick="GetCheck()" class="switch-input"/>
      <span class="switch-label"
data-on="On" data-off="Off"></span>
      <span class="switch-
handle"></span>
      </label> <span class="skrif">Señal
Digital1</span> <br/>
      <label class="switch">
      <input type="checkbox"
name="LED2" value="0" onclick="GetCheck()" class="switch-input"/>
      <span class="switch-label"
data-on="On" data-off="Off"></span>
      <span class="switch-
handle"></span>
      </label> <span class="skrif">Señal
Digital2</span>
    </form>
<fieldset class="SignalField">
<legend class="FieldLabelText" style="margin-left:12px;
padding-left:1px;">Enviar Señal</legend>
  <table border="0" cellspacing="0"
cellpadding="0" width = "201" align="left" style="padding:0 0 0 3px; margin:0px;">
    <tr><td align = "center">
      <button class="button" id="LED3"
type="button" onclick="GetButton1()">Prender Led</button>
    </td></tr>
    <tr>
      <td align = "center">
      <button class="button" id="LED4"
type="button" onclick="GetButton2()">Prender Led1</button>
    </td></tr>
    <tr><td align = "center">
      <button class="button" id="LED6"
type="button" onclick="GetButton4()">Prender Led2</button>
    </td></tr>
  </table>
</fieldset>
<fieldset class="StatusField">

```

```

class="switches">...</span></p>
<p>Electric Fence: <span
class="switches">...</span></p>
</fieldset>
</div>
<div class="IO_box">
<h2>Sensores</h2>
<span class="analog"></span>
data-type="canv-gauge"
data-animation-duration="900"
data-colors-needle="#47a8d8 #24819B"
data-highlights="0 5 #B3EC60, 5 8 #FF9640, 8 10
#FB7A66, 10 12 #F94E33"
data-onready="setInterval( function() {
Gauge.Collection.get('gauge1').setValue(data_val1);}, 1000);"
<</canvas>
<canvas class ="gaugestyle2" id="gauge2" width="155" height="155"
data-major-ticks="1 2 3 4 5 6 7 8 9 10 11 12"
data-glow="false"
data-animation-delay="80"
data-colors-needle="#47a8d8 #24819B"
data-highlights="0 5 #B3EC60, 5 8 #FF9640, 8 10
#FB7A66, 10 12 #F94E33"
data-onready="setInterval( function() {
Gauge.Collection.get('gauge2').setValue(data_val2);}, 1000);"
<</canvas>
</div>
</div>
</div>
</div class="push"></div>
<div class="footer">
<p>Copyright (c) 2015 Javier Flores Diego Villacrez</p>
</div>
</body>
</html>

```