

Universidad de Oriente
Facultad de Ingeniería Eléctrica
Departamento de Telecomunicaciones



TRABAJO DE DIPLOMA

**Sistema de supervisión y control para el Nodo
Central de la Universidad de Oriente.**

Autor: Humberto Hervella Mateos

Tutor: MSc. Ing. Daniel Iván Garrido Rodríguez

Santiago de Cuba

Junio, 2015

Universidad de Oriente
Facultad de Ingeniería Eléctrica
Departamento de Telecomunicaciones



TRABAJO DE DIPLOMA

**Sistema de supervisión y control para el Nodo
Central de la Universidad de Oriente.**

Autor: Humberto Hervella Mateos

Tutor: MSc. Ing. Daniel Iván Garrido Rodríguez

Santiago de Cuba

Junio, 2015



COMPROMISO DEL AUTOR

Hago constar que el presente trabajo de diploma es de mi autoría exclusivamente, no constituyendo copia de ningún trabajo realizado anteriormente y las fuentes usadas para la realización del trabajo se encuentran referidas en la bibliografía. Doy mi consentimiento a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización del Tutor o Institución.

Firma del Autor

PENSAMIENTO

"El hombre encuentra a Dios detrás de cada puerta que la ciencia logra abrir."

Albert Einstein

(1879-1955) Físico alemán.

DEDICATORIA

A mi madre, que con tanto amor, sacrificio y entrega incondicional, ha hecho de mí un hombre de bien y ha dado toda su vida por mí.

AGRADECIMIENTOS

A Dios, por estar siempre a mi lado, por ayudarme en los momentos más difíciles y por haberme permitido estar aquí este día.

A mi madre, por su apoyo incondicional, por sus consejos, por su ayuda, por su presencia, por existir, por la educación que me dio, por sus regaños siempre sabios y por todos los momentos que he pasado a su lado, muchas gracias mamita.

A mi familia amada: mis primos y tíos; por su amor tan sincero, en especial a mi tía Ana María por su apoyo y su ayuda para estar aquí y a mi padre por siempre haberme ayudado cuando lo he necesitado.

A mi novia Beatriz porque sin su presencia no hubiera llegado hasta el final, por apoyarme, quererme y ayudarme en lo que ha podido. A Agustín, Idalmis, Fátima e Isa por hacerme sentir parte de su familia.

A mi tutor Daniel Iván Garrido, por su ayuda, paciencia, orientación y amistad; su legado es mi ejemplo a seguir como profesional.

A mis amigos, más que amigos, mis hermanos, Eloy, Leito y Lisset que con su sincera amistad y apoyo me recuerdan cada día lo imprescindibles que son en mi vida.

A mi amigo Sergio, mi "FE", que sin su ayuda no habría podido obtener los resultados que hoy tengo, que a pesar de las distancias su amistad se ha mantenido fuerte y su apoyo incondicional me ha sido tan útil como gratificante.

A mis amigos y compañeros de batalla en estos largos años: David, Bernardino, Manuel, Elier, Albertico, Medina, Hassan, Brito, German, Francisco, Antonio, en fin todos mis compañeros de grupo por lo que sin su ayuda, sentido del humor, sin sus bromas, sin los momentos pasamos juntos, no hubiera podido llegar a este punto.

A todas esas personas que de una forma u otra hicieron posible este día, los que me ayudaron a crecer como persona y a superarme cada día más; muchas gracias por su apoyo.

RESUMEN

En el presente trabajo se diseñó e implementó el *software* y *hardware* de un sistema automatizado de supervisión y control encargado de registrar el voltaje de la línea de alimentación de los equipos de climatización y la temperatura del local de servidores del Nodo Central de la Universidad de Oriente para mantener el correcto funcionamiento de este Centro de Procesamiento de Datos. Los datos almacenados en una memoria SD son transmitidos a través de la red utilizando el protocolo Ethernet. Para la realización de dicho proyecto se utilizaron las tarjetas de tecnología Arduino, las cuales se basan en la filosofía de *software* y *hardware* libre. Se utilizó la tarjeta Arduino NANO para registrar estas variables y almacenarlas en la memoria, además de proporcionar a través de la configuración de un módulo Ethernet, encargado de transmitir los datos procesados, conexión a la red de la Universidad. El sistema muestra los datos en un visualizador gráfico de 320x240 pixel de resolución y permite ingresar datos a través de la pantalla táctil incorporada al mismo, usando como controlador de dichas funciones la tarjeta Arduino Mega que se comunica con el Arduino Nano a través del protocolo de comunicación I2C. Dicho trabajo tiene el objetivo de proteger los equipos de climatización de las variaciones de voltajes de la línea eléctrica, que a su vez protegen a los equipos de cómputo del local manteniendo un rango de temperatura adecuado para el perfecto funcionamiento de este Centro de Procesamiento de Datos.

Palabras clave: Arduino, Ethernet, I2C, LCD gráfico.

ABSTRACT

In this investigation, there were designed and implemented the software and hardware of an automated system of control and supervision. This system is in charge of register the voltage of the line of air conditioning equipment and the temperature of the servers of the Central Node of Universidad de Oriente, to keep the correct functioning of the Data Processing Center. All the data stored in an SD memory are transmitted through the net using Ethernet protocol. To carry out this research, Arduino technology cards were used. They were based on the free software and hardware philosophy. The Arduino NANO card was used to register the variables and store them in the memory. It also provides connection to the net of the University, through the settings of an Ethernet module, which is in charge of transmitting all the processed data. The system shows the data in a graphic 320x240 pixel browser, and allows to enter the information through its tactile screen, using the Arduino Mega card to control the functions, this card communicates with the Arduino Nano one through the communication protocol I2C. This research aims at protecting the air conditioning equipment from voltage variations in the electric lines. Same way the count equipment of the room will be protected by keeping a proper temperature rank for the perfect functioning of this Data Processing Center.

Keywords: Arduino, Ethernet, I2C, *graphic* LCD.

ÍNDICE

INTRODUCCIÓN	1
CAPITULO 1 . MARCO TEÓRICO REFERENCIAL.....	4
1.1. Variables a medir.....	4
1.1.1. Diferencia de potencial (V)	4
1.1.2. Temperatura.....	5
1.2. Hardware Arduino.....	6
1.2.1. ¿Qué es un sistema electrónico?.....	6
1.2.2. ¿Qué es Arduino?	7
1.2.3. ¿Cuál es el origen de Arduino?.....	9
1.2.4. Algunas placas oficiales de Arduino.....	9
1.2.5. ¿Por qué elegir Arduino?	18
1.3. Arduino Mega 2560.....	19
1.3.1. Información General.....	19
1.3.2. Especificaciones Técnicas.....	20
1.3.3. Alimentación.....	21
1.3.4. Memoria.....	22
1.3.5. Entrada y Salida.....	22
1.4. Arduino Nano	24
1.4.1. Información General.....	24
1.4.2. Especificaciones Técnicas.....	25
1.4.3. Alimentación.....	25
1.4.4. Memoria.....	26
1.4.5. Entrada y Salida.....	26
1.5. Pantalla de Cristal Líquido. (LCD).....	28
1.5.1. Tipos de LCD.....	28
1.5.2. Reflectivos y <i>Backlight</i>	29
1.5.3. Reflectivos:.....	29
1.5.4. <i>Backlight</i>	29
1.5.5. Matriz pasiva y matriz activa.....	30
1.5.6. Matriz pasiva.....	30

1.5.7.	Matriz activa.....	31
1.5.8.	Pantalla táctil. (<i>Touch screen</i>).....	32
1.5.9.	Pantalla táctil de 4 cables.....	33
1.5.10.	Pantalla táctil de 5 cables.....	34
1.6.	Controlador Ethernet, ENC28J60.....	35
1.6.1.	Principales Características.....	37
CAPITULO 2 . CENTRO DE PROCESAMIENTO DE DATOS DE LA UNIVERSIDAD DE ORIENTE, CORPUS.....		38
2.1.	Caracterización del local de servidores de CORPUS.....	39
2.1.1.	Descripción física y ubicación.....	40
2.2.	Riesgo de los Centros de Procesamiento de datos.....	41
2.3.	Protección de servicio técnico.....	43
2.3.1.	Suministro de Energía Eléctrica.....	43
2.3.2.	Aire Acondicionado.....	45
2.4.	Parámetros que se van a medir.....	46
CAPITULO 3 . DISEÑO Y PUESTA A PUNTO DEL SISTEMA DE SUPERVISIÓN Y CONTROL.....		47
3.1.	Diseño del hardware.....	47
3.1.1.	<i>Hardware</i> del sistema.....	47
3.1.2.	Sistema de adquisición de datos.....	48
3.1.3.	Sensores.....	49
3.1.4.	Acondicionamiento de la señal de Voltaje.....	51
3.1.5.	Conversión Analógica Digital.....	53
3.1.6.	Pantalla Táctil de 320x240 píxeles.....	54
3.1.7.	Codificador Rotatorio mecánico.....	63
3.1.8.	Comunicación I2C de Arduino.....	66
3.1.8.2.	Comunicación I2c entre Arduino Nano y Arduino Mega.....	68
3.1.9.	Circuito de Fuerza.....	69
3.1.10.	Ethernet.....	71
3.2.	Diseño del <i>software</i>	78
3.2.1.	Entorno de desarrollo para Arduino.....	78
3.2.2.	Librerías.....	79
		X

3.2.3. Registrador de variables eléctricas con monitoreo a través de un navegador web.	80
3.2.4. Programación de la tarjeta Arduino Nano.	82
3.2.5. Diagrama de flujo del software del Arduino Nano.....	82
3.3. Manual de Usuario del dispositivo.	85
CONCLUSIONES	89
RECOMENDACIONES.....	90
REFERENCIAS BIBLIOGRÁFICAS	91
ANEXOS	92

INTRODUCCIÓN

Un CPD (Centro de Procesamiento de Datos) es un edificio o sala de gran tamaño usada para mantener en él una gran cantidad de equipamientos electrónicos. Suelen ser creados y mantenidos por grandes organizaciones con objeto de tener acceso a la información necesaria para sus operaciones.

Entre los factores más importantes que motivan la creación de un CPD se destaca la garantía de la continuidad del servicio a clientes, empleados, ciudadanos, proveedores y empresas colaboradoras, pues en estos ámbitos es muy importante la protección física de los equipos informáticos o de comunicaciones implicadas, así como servidores de base de datos que puedan contener información crítica.

El diseño de un centro de procesamiento de datos comienza por la elección de su ubicación geográfica y requiere un balance entre diversos factores. Aun cuando se disponga del local adecuado siempre es necesario algún despliegue de infraestructura en su interior, como son los falsos techos y falsos suelos, cableado de red y teléfono, doble cableado eléctrico, instalaciones de alarmas, control de temperatura, humedad con avisos SNMP o SMTP, etc.

Generalmente todos los grandes servidores se suelen concentrar en una sala denominada “sala fría”, “nevera”, o “pecera”. Esta sala requiere de un sistema específico de refrigeración para mantener la temperatura baja (entre 21 y 23 grados centígrados), necesaria para evitar averías en las computadoras a causa del sobrecalentamiento, según las normas internacionales la temperatura exacta debe ser 22,3 grados centígrados.

CORPUS es el centro donde se desarrollan e implementan soluciones relacionadas con las Tecnologías de la Información y las Comunicaciones, donde se ofrecen servicios de Internet, correo, etc., a diferentes denominaciones de clientes, ya sean profesores, estudiantes o trabajadores. Para esto cuenta con un Centro de Procesamiento de Datos en el cuál se encuentran los equipos de cómputo que proveen dichos servicios, como son los *Routers*, *Switch*, servidores, etc. Estos equipos son particularmente susceptibles al daño causado producto del calor. Ciertos componentes electrónicos pueden sufrir roturas a temperaturas como 39 °C (100 °F) con un daño permanente como resultado de la

exposición a grados superiores a aproximadamente 49 °C (120 ° F). Las cintas magnéticas, discos flexibles y medios similares pueden comenzar a dañarse a una temperatura ambiente sostenida por encima de 37.8 °C (100 °F) y los discos duros a 66 °C (150 ° F). Por lo que se necesitan equipos de climatización para mantener la temperatura dentro de un rango establecido por las normas internacionales para los CPD. A su vez, estos equipos de enfriamiento son sensibles a los cambios bruscos de voltaje de la línea eléctrica por lo que en el presente trabajo se le da solución a dichas problemáticas.

Cada día se encuentran disponibles en el mercado diferentes dispositivos para realizar el control y monitoreo de los parámetros imprescindibles en un CPD, así como los parámetros eléctricos que permiten el perfecto funcionamiento del mismo, en algunos casos suelen ser diseñados de acuerdo a las necesidades del cliente. En la actualidad muchos de estos dispositivos de monitoreo están basados en la tecnología Arduino¹.

No obstante, el Instituto se vio obligado a cerrar sus puertas precisamente en 2005. Ante la perspectiva de perder en el olvido todo el desarrollo del proyecto Arduino que se había ido llevando a cabo durante aquel tiempo, se decidió liberarlo y abrirlo a “la comunidad” para que todo el mundo tuviera la posibilidad de participar en la evolución del proyecto, proponer mejoras y sugerencias y mantenerlo “vivo”. Y en la actualidad, la colaboración de muchas personas ha hecho de Arduino un proyecto de *hardware* y *software* libre de ámbito mundial.

Importancia del proyecto.

El costo de los registradores de variables tanto eléctricas como ambientales es alto, este proyecto presenta un sistema capaz de registrar los valores de corriente, voltaje y potencia así como la temperatura de bajo costo para el Centro de Procesamiento de Datos de la Universidad de Oriente (UO), CORPUS, ya que el software y el hardware de Arduino son arquitectura abierta.

¹ **Arduino:** Nació en el año 2005 en el Instituto de Diseño Interactivo de Ivrea (Italia), centro académico donde los estudiantes se dedicaban a experimentar con la interacción entre humanos y diferentes dispositivos (muchos de ellos basados en microcontroladores) para conseguir generar espacios únicos, especialmente artísticos. En el capítulo 1 abordamos más sobre este tema.

Problema

La inexistencia de un sistema para la supervisión y el control de la temperatura y el voltaje de la línea del CPD del Nodo Central de la Universidad de Oriente.

Objeto de estudio

Sistema de supervisión y control del Nodo Central de la Universidad de Oriente

Campo de acción

Sistema de supervisión y control para el CPD del Nodo Central de la Universidad de Oriente utilizando la plataforma Arduino.

Objetivo general

Diseñar e implementar un sistema de supervisión y control capaz de proteger los equipos de cómputo y de climatización que se encuentran en el Centro de Procesamiento de Datos del nodo central de la Universidad de Oriente.

Tareas:

1. Caracterizar el Centro de Procesamiento de Datos del Nodo Central de la Universidad de Oriente.
2. Realizar un estudio detallado de las particularidades y especificaciones de las tarjetas Arduino Nano y Arduino Mega 25 60 y el módulo Ethernet ENC28J60.
3. Realizar un estudio detallado del funcionamiento de las pantallas gráficas y táctiles.
4. Diseñar y construir un sistema para supervisar y controlar las variaciones de voltaje de la línea eléctrica y la temperatura del CPD del Nodo Central de la UO utilizando las tarjetas Arduino.
5. Añadirle al dispositivo comunicación a través del puerto Ethernet, utilizando el protocolo TCP para almacenar los datos registrados en una computadora.
6. Almacenar los datos registrados en una memoria SD o microSD.
7. Realizar pruebas utilizando el prototipo construido que permita evaluar el funcionamiento del dispositivo de acuerdo con las exigencias existentes para el Centro de Procesamiento de Datos.

CAPITULO 1 . MARCO TEÓRICO REFERENCIAL.

1.1. Variables a medir.

1.1.1. Diferencia de potencial (V)

La diferencia de potencial o tensión v entre dos puntos de un campo eléctrico es, por definición, el trabajo necesario para desplazar la unidad de carga eléctrica positiva de un punto al otro en contra o a favor de las fuerzas del campo. En el sistema MKS, la unidad de diferencia de potencial es el voltio (V) y corresponde al trabajo de 1 julio (J) al desplazar 1 culombio (C) de carga de uno al otro punto, es decir, $1 \text{ V} = 1 \text{ J/C}$. Si entre dos puntos existe una diferencia de potencial v (d.d.p.)², el trabajo necesario para desplazar una carga q será $q \cdot v$ y la carga se moverá del punto de mayor al de menor potencial.^[1]

Un agente o dispositivo, tal como una batería o un generador, posee una fuerza electromotriz (f.e.m.)³ si es capaz de suministrar a una carga eléctrica la energía suficiente para hacerla circular por él, del terminal de menor al de mayor potencial. La f.e.m. se mide por la d.d.p. en bornes del generador cuando no suministra corriente eléctrica, es decir, en circuito abierto.

Cuando separamos cargas positivas y negativas, tenemos que gastar energía en el proceso. La tensión es la energía por unidad de carga creada por la separación. Expresamos este cociente en forma diferencial como^[2]:

$$v = \frac{dw}{dq} \quad \text{Ec. 1.1}$$

Dónde:

v = tensión en voltios,

w = energía en julios,

² **d.d.p:** Diferencia de potencial.

³ **F.e.m:** Fuerza electromotriz.

q = carga en culombios.

1.1.2. Temperatura.

La temperatura es una magnitud referida a las nociones comunes de caliente o frío. Por lo general, un objeto más “caliente” tendrá una temperatura mayor, y si fuere frío una temperatura menor. Físicamente es una magnitud escalar relacionada con la energía interna de un sistema termodinámico, definida por el principio cero de la termodinámica. Más específicamente, está relacionada directamente con la parte de la energía interna conocida como “energía sensible”, que es la energía asociada a los movimiento de las partículas del sistema, sea en sentido traslacional, rotacional, o en forma de vibraciones. A medida que es mayor la energía sensible de un sistema, se observa que es más caliente, es decir su temperatura es mayor.

El desarrollo de la técnica para la medición de la temperatura ha pasado por un largo proceso histórico, ya que es necesario darle un valor numérico a una idea intuitiva como es lo frío o lo caliente.

Multitud de propiedades fisicoquímicas de los materiales o las sustancias varían en función de la temperatura a la que se encuentren, como por ejemplo su estado, su volumen, la solubilidad, la presión de vapor, su calor o la conductividad eléctrica.

La temperatura se mide con termómetros, los cuales pueden ser calificados de acuerdo a una multitud de escalas que dan lugar a unidades de medición de la temperatura.

Otra vía de medir la temperatura es a través de los sensores. Los sensores son dispositivos capaces de medir magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, etc. Una magnitud eléctrica puede ser una resistencia (como en una RTD⁴), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en los fototransistores), etc.

La resolución de un sensor es el menor cambio en la magnitud de entrada que se aprecia en la magnitud de la salida. Sin embargo, la precisión es el máximo error esperado en la medida.

⁴ RTD: Detector de temperatura Resistivo, son sensores de temperatura basados en la variación de la resistencia de un conductor con la temperatura.

1.2. Hardware Arduino.

1.2.1. ¿Qué es un sistema electrónico?

Un sistema electrónico es un conjunto de: sensores, circuitería de procesamiento y control, actuadores y fuente de alimentación. ^[3]

Los sensores obtienen información del mundo físico externo y la transforman en una señal eléctrica que puede ser manipulada por la circuitería interna de control. Existen sensores de todo tipo: de temperatura, de humedad, de movimiento, de sonido (micrófonos), etc. ^[3]

Los circuitos internos de un sistema electrónico procesan la señal eléctrica convenientemente. La manipulación de dicha señal dependerá tanto del diseño de los diferentes componentes hardware del sistema, como del conjunto lógico de instrucciones (es decir, del “programa”) que dicho hardware tenga pregrabado y que sea capaz de ejecutar de forma autónoma.

Los actuadores transforman la señal eléctrica acabada de procesar por la circuitería interna en energía que actúa directamente sobre el mundo físico externo. Ejemplos de actuadores son: un motor (energía mecánica), una bombilla (energía lumínica), un altavoz (energía acústica), etc.

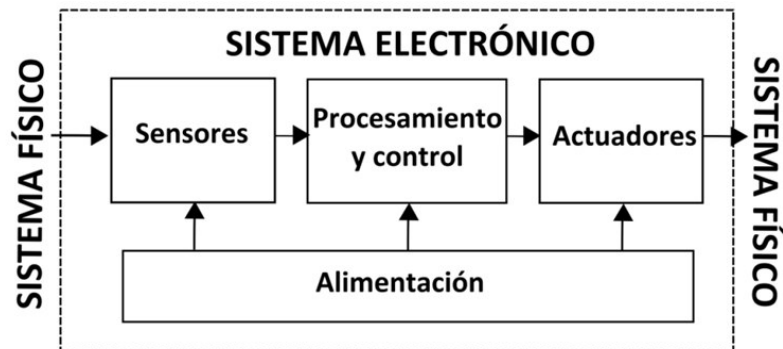


Figura 1.3 Esquema en bloques de un sistema electrónico.

1.2.2. ¿Qué es Arduino?

Una placa hardware libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra (los cuales están unidos internamente a las patillas de E/S del microcontrolador) que permiten conectar de forma muy sencilla y cómoda diferentes sensores y actuadores.^[3]

Cuando hablamos de “placa hardware” nos estamos refiriendo en concreto a una PCB (del inglés “*printed circuit board*”, o sea, placa de circuito impreso). Las PCBs son superficies fabricadas de un material no conductor (normalmente resinas de fibra de vidrio reforzada, cerámica o de plástico) sobre las cuales aparecen laminadas (“pegadas”) pistas de material conductor (normalmente cobre). Las PCBs se utilizan para conectar eléctricamente, a través de los caminos conductores, diferentes componentes electrónicos soldados a ella. Una PCB es la forma más compacta y estable de construir un circuito electrónico (en contraposición a una *breadboard*, *perfboard* o similar) pero, al contrario que estas, una vez fabricada, su diseño es bastante difícil de modificar. Así pues, la placa Arduino no es más que una PCB que implementa un determinado diseño de circuitería interna.

No obstante, cuando hablamos de “placa Arduino”, deberíamos especificar el modelo concreto, ya que existen varias placas Arduino oficiales, cada una con diferentes características (como el tamaño físico, el número de pines-hembra ofrecidos, el modelo de microcontrolador incorporado –y como consecuencia, entre otras cosas, la cantidad de memoria utilizable–, etc.). Conviene conocer estas características para identificar qué placa Arduino es la que nos convendrá más en cada proyecto.

De todas formas, aunque puedan ser modelos específicos diferentes (tal como acabamos de comentar), los microcontroladores incorporados en las diferentes placas Arduino pertenecen todos a la misma “familia tecnológica”, por lo que su funcionamiento es bastante parecido entre sí. En concreto, todos los microcontroladores son de tipo AVR⁵, una arquitectura de microcontroladores desarrollada y fabricada por la marca Atmel.

Un software (más en concreto, un “entorno de desarrollo”) **gratis, libre y multiplataforma** (ya que funciona en *Linux*, *MacOS* y *Windows*) que debemos instalar en

⁵ **AVR**: son una familia de microcontroladores RISC de Atmel. La arquitectura de los AVR fue concebida por dos estudiantes en el *Norwegian Institute of Technology*, y posteriormente refinada y desarrollada en ATMEL Norway, la empresa subsidiaria de Atmel, fundada por los dos arquitectos del chip.

nuestro ordenador y que nos permite escribir, verificar y guardar (“cargar”) en la memoria del microcontrolador de la placa Arduino el conjunto de instrucciones que deseamos que este empiece a ejecutar. Es decir: nos permite programarlo. La manera estándar de conectar nuestro computador con la placa Arduino para poder enviarle y grabarle dichas instrucciones es mediante un simple cable USB, gracias a que la mayoría de placas Arduino incorporan un conector de este tipo. ^[3]

Los proyectos Arduino pueden ser autónomos o no. En el primer caso, una vez programado su microcontrolador, la placa no necesita estar conectada a ningún computador y puede funcionar autónomamente si dispone de alguna fuente de alimentación. En el segundo caso, la placa debe estar conectada de alguna forma permanente (por cable USB, por cable de red Ethernet, etc.) a un computador ejecutando algún software específico que permita la comunicación entre este y la placa y el intercambio de datos entre ambos dispositivos. Este software específico lo deberemos programar generalmente nosotros mismos mediante algún lenguaje de programación estándar como Python, C, Java, Php, etc., y será independiente completamente del entorno de desarrollo Arduino, el cual no se necesitará más, una vez que la placa ya haya sido programada y esté en funcionamiento.

Un lenguaje de programación libre. Por “lenguaje de programación” se entiende cualquier idioma artificial diseñado para expresar instrucciones (siguiendo unas determinadas reglas sintácticas) que pueden ser llevadas a cabo por máquinas. Concretamente dentro del lenguaje Arduino, encontramos elementos parecidos a muchos otros lenguajes de programación existentes (como los bloques condicionales, los bloques repetitivos, las variables, etc.), así como también diferentes comandos así mismo llamados “órdenes” o “funciones” que nos permiten especificar de una forma coherente y sin errores las instrucciones exactas que queremos programar en el microcontrolador de la placa. Estos comandos los escribimos mediante el entorno de desarrollo Arduino.

Con Arduino se pueden realizar multitud de proyectos de rango muy variado: desde robótica hasta domótica, pasando por monitorización de sensores ambientales, sistemas de navegación, telemática, etc. Realmente, las posibilidades de esta plataforma para el desarrollo de productos electrónicos son prácticamente infinitas y tan solo están limitadas por nuestra imaginación.

1.2.3. ¿Cuál es el origen de Arduino?

Arduino nació en el año 2005 en el Instituto de Diseño Interactivo de Ivrea (Italia), centro académico donde los estudiantes se dedicaban a experimentar con la interacción entre humanos y diferentes dispositivos (muchos de ellos basados en microcontroladores) para conseguir generar espacios únicos, especialmente artísticos. Arduino apareció por la necesidad de contar con un dispositivo para utilizar en las aulas que fuera de bajo coste, que funcionase bajo cualquier sistema operativo y que contase con documentación adaptada a gente que quisiera empezar de cero. La idea original fue, pues, fabricar la placa para uso interno de la escuela.

No obstante, el Instituto se vio obligado a cerrar sus puertas precisamente en 2005. Ante la perspectiva de perder en el olvido todo el desarrollo del proyecto Arduino que se había ido llevando a cabo durante aquel tiempo, se decidió liberarlo y abrirlo a “la comunidad” para que todo el mundo tuviera la posibilidad de participar en la evolución del proyecto, proponer mejoras y sugerencias y mantenerlo “vivo”. Y así ha sido: la colaboración de muchísima gente ha hecho que Arduino poco a poco haya llegado a ser lo que es actualmente: un proyecto de hardware y software libre de ámbito mundial.

El principal responsable de la idea y diseño de Arduino, y la cabeza visible del proyecto es el llamado “*Arduino Team*”, formado por Massimo Banzi (profesor en aquella época del Instituto Ivrea), David Cuartielles (profesor de la Escuela de Artes y Comunicación de la Universidad de Malmö, Suecia), David Mellis (por aquel entonces estudiante en Ivrea y actualmente miembro del grupo de investigación *High-Low Tech* del MIT Media Lab), Tom Igoe (profesor de la Escuela de Arte Tisch de Nueva York), y Gianluca Martino (responsable de empresa fabricante de los prototipos de las placas).

1.2.4. Algunas placas oficiales de Arduino. ^[3]

Existen varios tipos de placas Arduino, cada una con características específicas que hay que conocer para poder elegir el modelo que más nos convenga según el caso.

Arduino Uno.

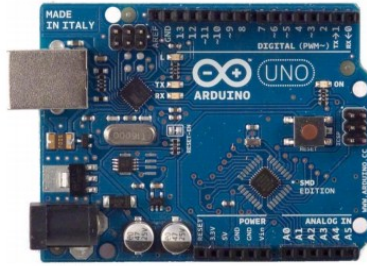


Figura 1.4. Arduino UNO.

Arduino UNO es una placa electrónica basada en el microcontrolador ATmega 328. Cuenta con 14 pines digitales que pueden ser entradas o salidas de los cuales 6 pueden ser usados como salidas PWM, 6 entradas analógicas, un cristal oscilador de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. La placa electrónica tiene todo lo necesario para proporcionar soporte al microcontrolador, basta con conectar la placa a una computadora con un cable USB o a un adaptador AC-DC o a una batería para empezar.

La tarjeta Arduino UNO SMD difiere de todas las placas anteriores ya que no utiliza el chip FTDI USB serie. Por el contrario cuenta con el ATmega16U2 programado como un convertor de USB a serie. “Uno” significa uno en italiano y se llama así por motivos de la próxima versión de Arduino 1.0. La tarjeta Uno y la versión 1.0 serán la versión de referencia de Arduino para seguir adelante. La tarjeta Uno es la última en una serie de placas USB de Arduino, y el modelo de referencia para la plataforma Arduino.

Arduino Mega 2560.



Figura 1.5. Arduino Mega 2560.

Placa basada en el microcontrolador ATmega2560. Como características más destacables diremos que tiene 54 pines de entrada/salida digitales (de los cuales 14

pueden ser usados como salidas analógicas PWM), 16 entradas analógicas y 4 receptores/transmisores serie TTL-UART⁶. Consta de una memoria Flash de 256 Kilobytes (de los cuales 8 están reservados para el *bootloader*), una memoria SRAM de 8 KB y una EEPROM de 4 KB. Su voltaje de trabajo es igual al del modelo UNO: 5 V.

Arduino Mega ADK.

Placa muy similar a la Mega 2560. La diferencia principal está en que la Mega ADK es capaz de funcionar como un dispositivo de tipo “*host USB*” (y la Mega 2560 no).



Figura 1.6. Arduino Mega ADK.

En una comunicación USB entre diferentes dispositivos siempre existe uno que actúa como “maestro” (el llamado “*host*”) y el otro –u otros– que actúan como “esclavos” (los llamados “periféricos”). El “*host*” es el único que puede iniciar y controlar la transferencia de datos entre el resto de dispositivos conectados, mientras que los “periféricos” tan solo pueden responder a las peticiones hechas por el “*host*” y poca cosa más. Los dispositivos “*host*” disponen de un conector USB de tipo A, y los dispositivos “periféricos” disponen de un conector USB de tipo B, mini-B o micro-B. Un “*host*” típico es un computador, al cual se le pueden conectar varios “periféricos”, como lápices de memoria, cámaras de fotos o video, teléfonos móviles de última generación, etc.

La placa Arduino ADK puede funcionar como periférico USB igual que el resto de placas Arduino (por eso incorpora el conector USB de tipo B como las demás) pero también como *host USB* (y por eso, como novedad, también incorpora un conector USB de tipo A). Pero no es suficiente con tener el conector adecuado para que la placa pueda ser un *host USB*: en realidad este modo de funcionamiento es posible gracias a que la placa

⁶ **UART:** Transmisor-Receptor Asíncrono Universal, Universal Asyncthonous Receiver-Transmitter. Este controla los puertos y dispositivos series. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

incorpora un chip específico (un microcontrolador, de hecho) que implementa la lógica necesaria: el MAX3421E del fabricante Maxim, el cual se comunica con el ATmega2560 mediante SPI.

Así pues, a la placa ADK se puede conectar cualquier dispositivo que tenga un puerto USB periférico (teléfonos móviles, cámaras de fotos o vídeo, teclados, ratones, joysticks y mandos de diferentes videoconsolas, etc.) para controlarlo e interactuar directamente con él.

Arduino Ethernet.

Al igual que el modelo UNO, la placa Ethernet está basada en el microcontrolador ATmega328P (y por lo tanto, tiene la misma cantidad de memoria Flash, SRAM⁷ y EEPROM⁸), y también tiene el mismo número de pines de entrada/salida digitales y de entradas analógicas. El resto de características también es muy similar al modelo UNO. La mayor diferencia que existe con la placa UNO es que la placa Ethernet incorpora un zócalo de tipo RJ-45 para poder conectarse mediante el cable adecuado (cable de par trenzado de categoría 5 o 6) a una red de tipo Ethernet.

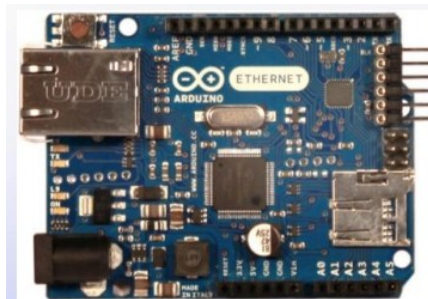


Figura 1.7. Arduino Ethernet.

Las redes Ethernet, basadas en la especificación IEEE802.3, son la tecnología más extendida con diferencia en la construcción de redes de área local (las llamadas “LAN” – Local Area Network–). Las redes LAN permiten el tráfico de datos entre los distintos dispositivos conectados a ella (computadores, impresoras, “routers”, etc.) dentro de una zona de distancias relativamente reducidas, tal como un edificio o un conjunto de edificios adyacentes.

⁷ **SRAM:** Memoria estática de acceso aleatorio es un tipo de memoria basada en semiconductores es capaz de mantener datos sin necesidad de circuitos de refresco. Son memorias volátiles.

⁸ Es un tipo de memoria ROM programables y borrable eléctricamente. Son memorias no volátiles.

Aunque teóricamente se explica de un modo, en la práctica las redes Ethernet utilizan el mismo conjunto de protocolos de comunicación que Internet (la llamada pila “TCP/IP”).

La placa Arduino Ethernet permite, transferir datos entre ella misma (los cuales pueden ser obtenidos de algún sensor, por ejemplo) y cualquier otro dispositivo conectado a su misma red LAN (normalmente, un computador que los recopila y guarda), o viceversa: transferir datos entre un dispositivo conectado a la LAN (normalmente, un computador ejecutando algún tipo de *software* de control) y ella misma (la cual puede estar conectada a algún actuador controlado remotamente por ese computador). También se puede lograr, gracias al establecimiento del enrutamiento de paquetes adecuado, comunicar nuestra placa Ethernet con cualquier dispositivo conectado a cualquier red del mundo fuera de nuestra LAN privada (incluyendo “*Internet*”), con lo que sus posibilidades de uso se disparan.

Evidentemente, solo con disponer de un zócalo RJ-45 la placa Arduino Ethernet no es capaz “por arte de magia” de comunicarse dentro de una red. Lo que permite que esto ocurra es la inclusión dentro de la placa de un chip controlador de Ethernet que, de hecho, implementa por hardware toda la pila de protocolos TCP/IP (en concreto los protocolos TCP, UDP, ICMP, Ipv4, ARP, IGMPv2, PPPoE y por supuesto Ethernet). Se trata del W5100 del fabricante Wiznet. Este chip es el verdadero responsable de que la placa Arduino Ethernet pueda manejar redes de este tipo.

Que este chip “implemente por *hardware* la pila TCP/IP” significa en pocas palabras que permite al desarrollador de programas Arduino utilizar de forma muy sencilla la red (mediante la librería de programación “*Ethernet*”, incorporada por defecto dentro del lenguaje Arduino) para poder transmitir o recibir datos sin tener que preocuparnos por detalles más técnicos (como el control de errores en la transmisión de datos, la sincronización de las señales y datagramas, etc.) de los cuales ya se encarga directamente este chip.

Entre otras características, conviene saber que el W5100 puede funcionar a velocidades de transmisión de 10 y 100 megabits por segundo, que permite hasta cuatro conexiones simultáneas independientes y que dispone de una memoria interna de 16 kilobytes para almacenar temporalmente datos enviados o recibidos de la red. El W5100 se comunica mediante el protocolo SPI con el ATmega328P, por lo que hay que tener en cuenta que los pines digitales números 10, 11, 12 y 13 de la placa Arduino Ethernet están reservados

(es decir, no se pueden utilizar para otra cosa). Por esta razón, el pin que está asociado al LED que viene incorporado en la placa no es el número 13 como ocurría con la placa UNO, sino que es el número 9, y además se reduce el número de pines disponibles respecto el modelo UNO a 9, con solo cuatro pines disponibles como salidas PWM.

Arduino Fio



Figura 1.8. Arduino Fio.

Esta placa contiene un ATmega328P funcionando a 3,3 V y a 8 MHz. Tiene 14 agujeros que pueden utilizarse (mediante soldadura directa o bien mediante la colocación de pines-hembra de plástico) como pines de entrada/salida digitales (6 de los cuales pueden ser usados como salida PWM); también tiene 8 agujeros preparados para utilizarse como entradas analógicas y un botón de reinicio, todo ello dentro de un tamaño muy reducido.

Una novedad de esta placa respecto a las anteriores es que se puede alimentar eléctricamente mediante una batería LiPo gracias a que la placa dispone de un zócalo de tipo JST de 2 pines para poder conectarla directamente allí. La placa Arduino Fio permite ser alimentada también mediante conexión USB, ya que dispone de un conector USB mini-B para ello. De hecho, a través de la alimentación recibida vía USB incluso se puede recargar la batería LiPo que esté conectada en ese momento, ya que la placa incorpora el chip cargador MAX1555 del fabricante Maxim. No obstante, la conexión USB no está pensada para programar el microcontrolador, por lo que se requiere para ello acoplar un adaptador USB-Serie (como los ya comentados cuando vimos la placa Arduino Ethernet) a los agujeros de la placa marcados como GND, AREF, 3V3, RXI, TXO y DTR, mediante la ristra de pines adecuada.

Arduino Pro.



Figura 1.9. Arduino Pro.

Esta placa viene en dos “versiones”: ambas contienen un microcontrolador Atmega328P SMD, pero una funciona con 3,3 V y a 8 MHz y la otra funciona con 5 V y a 16 MHz. Dispone de 14 agujeros pensados para funcionar como pines de entrada/salida digital (6 de los cuales pueden ser usados como salida PWM), 6 agujeros para entradas analógicas, agujeros para montar un conector de alimentación de 2,1 mm, un zócalo JST para una batería LiPo externa, un interruptor de corriente, un botón de reinicio, un conector ICSP y los pines necesarios para conectar un adaptador o cable USB-Serial y así poder programarla (y también alimentarla) directamente vía USB.

Arduino Lilypad.

La placa Arduino Lilypad está diseñada para ser cosida a material textil. Permite además conectarle (mediante hilos conductores) fuentes de alimentación, sensores y actuadores de forma que se puedan “llevar encima”, haciendo posible la creación de vestidos y ropa “inteligente”.

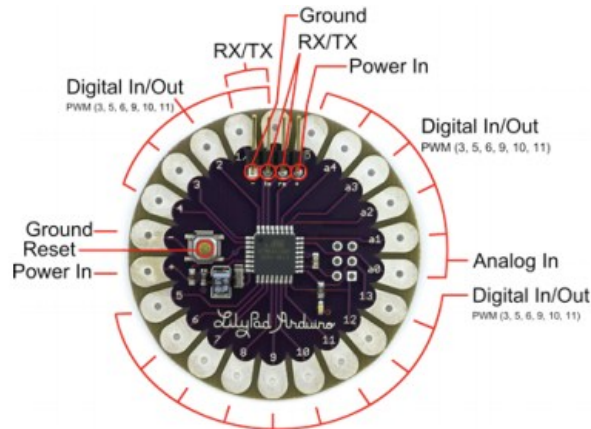


Figura 1.10. Arduino Lilypad.

Además, se puede lavar. Esta placa incorpora el microcontrolador ATmega328V (una versión de bajo consumo del Atmega328P), el cual se programa acoplado a la placa un adaptador o cable USB-Serie.

Arduino Nano.



Figura 1.11. Arduino Nano.

La característica más destacable de esta placa es que a pesar de su tamaño (0,73 pulgadas de anchura por 1,70 de longitud), sigue ofreciendo el mismo número de salidas y entradas digitales y analógicas que la Arduino UNO y la misma funcionalidad que esta. La consecuencia más evidente de su reducido tamaño es que carece del conector de alimentación de 2,1 mm (aunque puede seguir siendo alimentada por una fuente externa mediante el pin “Vin” o “5 V”) e incorpora un conector USB mini-B en vez del conector USB tipo B.

Otra diferencia es que, aunque la placa Arduino Nano se sigue basando en el microcontrolador ATmega328P (en formato SMD), el convertor USB-Serie que lleva incorporado es el chip FTDI FT232RL y no el ATmega16U2.

Esta placa está especialmente pensada para conectarla a una *breadboard* mediante las patillas que sobresalen de su parte posterior, pudiendo formar parte así de un circuito complejo de una manera relativamente fija.

Arduino Mini.

Esta placa es muy parecida a la placa Arduino Nano: está basada igualmente en el microcontrolador ATmega328P SMD funcionando a 16MHz, tiene 14 pines de entrada/salida digitales (6 de los cuales pueden funcionar como salidas PWM) y 8 entradas analógicas. Y al igual que la placa Arduino Nano, la Arduino Mini está especialmente pensada para conectarla a una “*breadboard*” mediante las patillas que sobresalen de su parte posterior, pudiendo formar parte así de un circuito complejo de una manera relativamente fija.

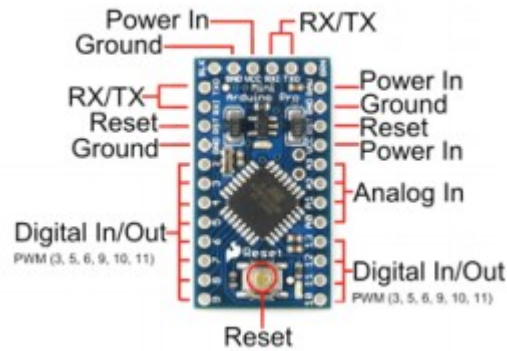


Figura 1.12. Arduino Mini.

La diferencia más importante con la placa Arduino Nano está en que la Arduino Mini (para ahorrar aún más espacio físico y así conseguir un tamaño realmente mínimo de 0,7 pulgadas de ancho por 1,3 de largo) no incorpora ningún chip conversor USB-Serie. Debido a ello, para su programación se necesita utilizar un adaptador USB-Serial externo.

Arduino Leonardo.



Figura 1.13. Arduino Leonardo.

La gran novedad de esta placa es que el microcontrolador que incorpora es el ATmega32U4 (en formato SMD), el cual tiene todas las funcionalidades que ofrece el ATmega328P pero incorpora además 0,5 kilobytes más de memoria SRAM y sobre todo, soporta comunicaciones USB directamente (y por tanto, no necesita de ningún chip suplementario como el ATmega16U2 o el FTDI).

Otras diferencias con la placa UNO es que la placa Leonardo incorpora un pin-hembra digital más que la UNO para ser usado como salida PWM (el nº 13) y 6 entradas analógicas extra más, las cuales están situadas físicamente en los pines-hembras digitales marcados con un puntito en el exterior de la placa.

En cuanto a los pines SDA y SCL para la comunicación I2C/TWI la diferencia radica en que cambian de ubicación respecto al UNO y ahora pasan a estar en los pines-hembra

digitales nº 2 y nº 3. Por otro lado, en la placa Leonardo desaparecen los pines GPIO SPI, por lo que la única manera de comunicar esta placa con el exterior mediante este protocolo es utilizando directamente los pines ICSP.

1.2.5. ¿Por qué elegir Arduino?

Existen muchas otras placas de diferentes fabricantes que, aunque incorporan diferentes modelos de microcontroladores, son comparables y ofrecen una funcionalidad más o menos similar a la de las placas Arduino. Todas ellas también vienen acompañadas de un entorno de desarrollo agradable y cómodo y de un lenguaje de programación sencillo y completo. No obstante, la plataforma Arduino (hardware + software) ofrece una serie de ventajas:

- **Arduino es libre y extensible:** esto quiere decir que cualquiera que desee ampliar y mejorar tanto el diseño hardware de las placas como el entorno de desarrollo software y el propio lenguaje de programación, puede hacerlo sin problemas. Esto permite que exista un rico “ecosistema” de extensiones, tanto de variantes de placas no oficiales como de librerías software de terceros, que pueden adaptarse mejor a nuestras necesidades concretas.
- **Arduino tiene una gran comunidad:** muchas personas lo utilizan, enriquecen la documentación y comparten continuamente sus ideas.
- **Su entorno de programación es multiplataforma:** se puede instalar y ejecutar en sistemas Windows, Mac OS X y Linux. Esto no ocurre con el software de muchas otras placas.
- **Su entorno y el lenguaje de programación son simples y claros:** son muy fáciles de aprender y de utilizar, a la vez que flexibles y completos para que los usuarios avanzados puedan aprovechar y expresar todas las posibilidades del hardware. Además, están bien documentados, con ejemplos detallados y gran cantidad de proyectos publicados en diferentes formatos.
- **Las placas Arduino son baratas:** la placa Arduino estándar (llamada Arduino UNO) ya pre-ensamblada y lista para funcionar, tienen muy bajo costo en el mercado internacional. Incluso, cualquier persona las podría construir (Arduino es

hardware libre) adquiriendo los componentes por separado, con lo que el precio total de la placa resultante sería incluso menor.

- **Las placas Arduino son reutilizables y versátiles:** reutilizables porque se puede aprovechar la misma placa para varios proyectos (ya que es muy fácil de desconectarla, reconectarla y reprogramarla), y versátiles porque las placas Arduino proveen varios tipos diferentes de entradas y salidas de datos, los cuales permiten capturar información de sensores y enviar señales a actuadores de múltiples formas.

1.3. Arduino Mega 2560. ^[4]

El Arduino Mega 2560 fue la tarjeta que se usó para controlar el display gráfico de 320x240 pixeles, y la pantalla táctil, configurando sus funciones y atendiendo a los eventos generados por esta.

1.3.1. Información General.

Arduino Mega 2560 es una plataforma de hardware y software basado en el microcontrolador ATmega2560. Esta tarjeta tiene 54 pines de entrada/salida digitales, de los cuales 14 pueden ser usados como salidas PWM⁹, tiene también 16 entradas analógicas, 4 UART (hardware de puerto serie), un oscilador de cristal a 16 MHz, una conexión USB, un conector para la fuente de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para la alimentación del microcontrolador; simplemente conectándolo a una computadora a través de un cable USB o una fuente de alimentación con adaptador de AC a DC o una batería para dar el encendido.

⁹ **PWM:** La modulación por ancho de pulso de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, ya sea para transmitir información a través de un canal o controlar la cantidad de energía que se envía a una carga.

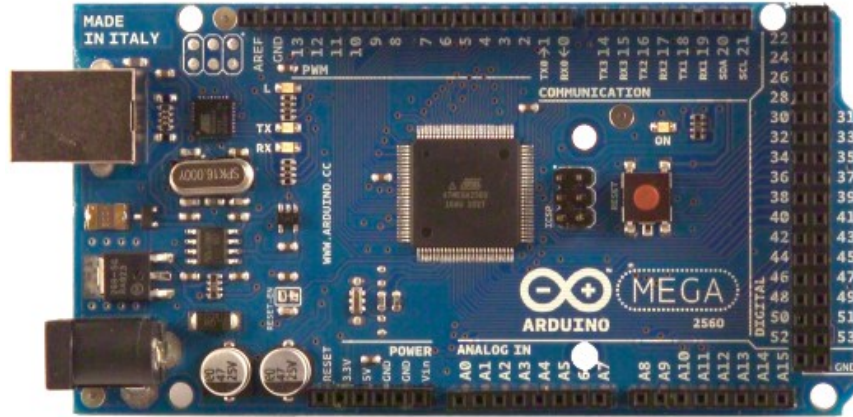


Figura 1.14. Placa Arduino Mega 2560.

1.3.2. Especificaciones Técnicas.

Tabla 1.1 Especificaciones técnicas Arduino Mega2560.

Microcontrolador	ATMEGA2560
Voltaje de Operación	5V
Entrada de Voltaje (Recomendada)	7 – 12V
Entrada de Voltaje (Límites)	6 – 20V
Pines de I/O digitales	54 pines (14 están provistos de salida PWM)
Pines de Entrada Analógica	16 pines
Corriente de DC por pines I/O	40 mA
Corriente de DC para pines de 3.3V	50 mA
Memoria Flash	256 KB
SRAM	8 KB
EEPROM	4 KB
Frecuencia del reloj	16 MHz

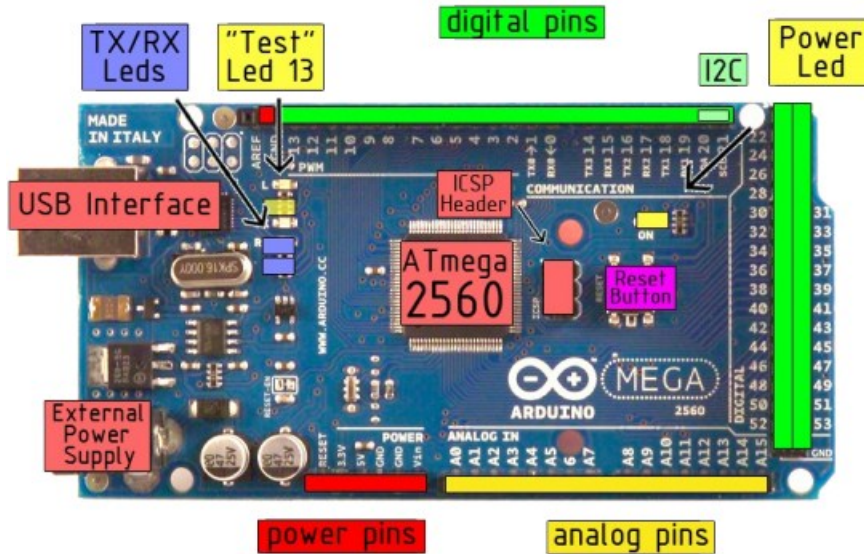


Figura 1.15. Esquema del hardware de la placa Arduino Mega 2560.

1.3.3. Alimentación.

Arduino Mega 2560 puede ser alimentado a través de la conexión USB o con fuente de alimentación externa. La fuente de alimentación es seleccionada automáticamente.

La placa puede operar con una fuente externa de 6 a 20 voltios. Si es alimentada con un voltaje menor a 7V el pin de 5V podría proporcionar menos de cinco voltios y además la placa puede ser inestable. Si se utiliza un voltaje de alimentación mayor a 12V, el regulador de voltaje se puede calentar y dañar la placa. Por lo tanto el rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- **VIN:** Es el pin del voltaje de entrada a la tarjeta Arduino cuando se utiliza una fuente de alimentación externa (opuesta a los 5 voltios de la conexión USB u otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o mediante el enchufe de alimentación de la placa.
- **5V:** Este pin genera cinco voltios regulados por el regulador de la placa. La placa puede ser alimentada con energía eléctrica ya sea a partir de la entrada de alimentación (7–12V), el conector USB (5V) o mediante el pin VIN de la placa (7-12V). El suministro de tensión a través de los pines de 5V o 3.3V no pasan por el regulador y pueden dañar la placa.

- **3.3V:** Este pin proporciona 3.3 voltios generados por el regulador de la placa. El consumo máximo de corriente es de 50 mA.
- **GND:** Pines de tierra.

1.3.4. Memoria.

El ATmega2560 tiene 256 KB de memoria (8 KB son utilizados para el gestor de arranque). También dispone de 8 KB de SRAM y 4 KB de memoria EPROM (que puede ser leído y escrito con la librería EEPROM).

1.3.5. Entrada y Salida.

Cada uno de los 54 pines del Mega puede ser utilizado como entrada o salida, usando las funciones *pinMode()*, *digitalWrite()* y *digitalRead()*. Estos pines operan con 5V. Cada pin puede entregar un máximo de 40 mA y tienen una resistencia de *pull-up*, desconectadas por defecto, de un valor de 20-50 kOhms. En función a esto algunos pines tienen funciones específicas:

Serie: Usado para recibir y transmitir datos TTL modo serie. Los pines 0 y 1 están conectados a los puntos correspondientes del chip serial ATmega8U2 USB-a-TTL, esto permite la comunicación vía serie a través del puerto USB con la computadora y el *driver* correspondiente al Arduino Mega2560.

Tabla 1.2 Relación de pines para la transmisión y recepción serie.

Relación de pines de Trasmisión y Recepción.	
RX	TX
0	1
19	18
17	16
15	14

Interrupciones externas: Estos pines se pueden configurar para disparar una interrupción en un valor bajo, un flanco ascendente o descendente o un cambio en el valor. Para realizar esto se utiliza la función *attachInterrupt()*.

Tabla 1.3 Relación de pines para las interrupciones externas.

Relación de pines con sus respectivas interrupciones.	
Pines	Interrupciones
2	0
3	1
18	5
19	4
20	3
21	2

PWM: Pines de 0 al 13. Proveen 8 bit de salida PWM con la función *analogWrite()*.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Estos pines soportan la comunicación SPI a través de la librería SPI como su nombre lo indica.

LED 13: Hay un LED incorporado conectado al pin digital 13. Cuando el pin esta en ALTO, el LED está encendido, cuando el pin esta en BAJO, se apaga el LED.

I²C: 20 (SDA) y 21 (SCL). Soporta la comunicación I2C usando la librería *Wire*.

El Arduino Mega 2560 tiene 16 entradas analógicas, cada una de las cuales proveen 10 bits de resolución. Por defecto miden de 0 a 5V, aunque es posible cambiar el límite superior de su rango con el pin de AREF y la función *analogReference ()*. Además hay otros pines que tiene funciones específicas como:

AREF. Referencia de voltaje para las entradas analógicas usando la función *analogReference ()*.

RESET: Provoca un valor BAJO en la línea que reinicia al microcontrolador. Normalmente es utilizado para añadir un botón de reinicio en la placa.

1.4. Arduino Nano.

El Arduino Nano fue la tarjeta que se usó como procesador central del sistema y a continuación se hace una descripción de sus características más importantes.

1.4.1. Información General.

Arduino Nano es una pequeña, completa y agradable plataforma de hardware y software basado en el microcontrolador ATmega328. Esta tarjeta tiene más o menos el mismo funcionamiento que la tarjeta Arduino Mega, pero con diferente estructura. Solo cuenta con un conector de alimentación de DC, y trabaja usando el conector mini-B USB. El Nano tiene 14 pines de entrada/salida digitales, de los cuales 6 pueden ser usados como salidas PWM, tiene también 8 entradas analógicas, 1 UART (hardware de puerto serie), un oscilador de cristal a 16 MHz, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para la alimentación del microcontrolador; simplemente conectándolo a una computadora a través de un cable mini-B USB o una fuente de alimentación con adaptador de AC a DC o una batería para dar el encendido. ^[5]

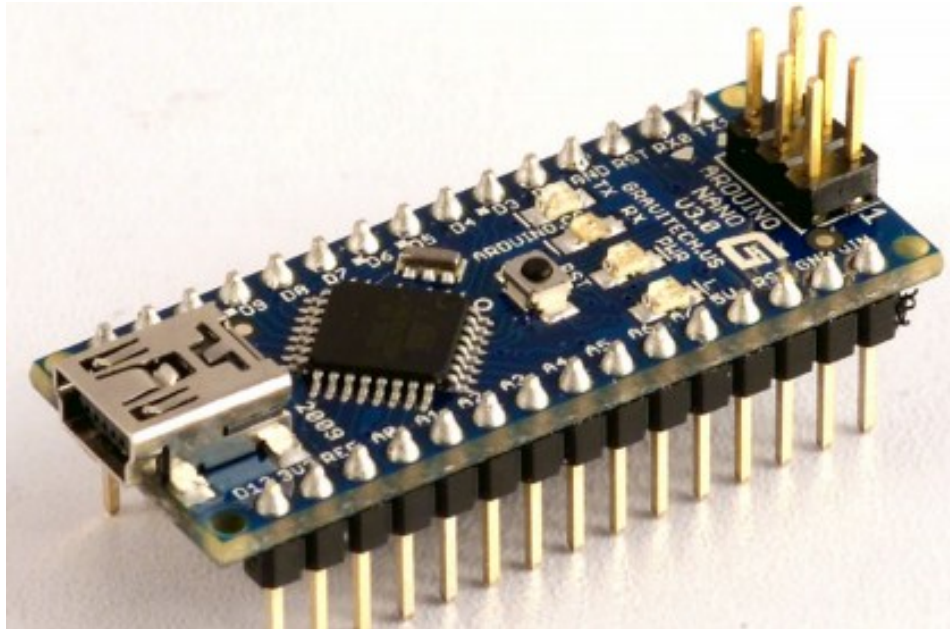


Figura 1.14. Placa Arduino Nano.

1.4.2. Especificaciones Técnicas.

Tabla 1.4 Especificaciones técnicas Arduino Nano.

Microcontrolador	ATMEGA328
Voltaje de Operación	5V
Entrada de Voltaje (Recomendada)	7 – 12V
Entrada de Voltaje (Límites)	6 – 20V
Pines de I/O digitales	14 pines (6 están provistos de salida PWM)
Pines de Entrada Analógica	8 pines
Corriente de DC por pines I/O	40 mA
Corriente de DC para pines de 3.3V	50 mA
Memoria Flash	32 Kb (2 Kb usados para el <i>bootloader</i>)
SRAM	2 Kb
EEPROM	1 Kb
Frecuencia del reloj	16 MHz

1.4.3. Alimentación.

Arduino Nano puede ser alimentado a través de la conexión mini-B USB o con fuente de alimentación externa. La fuente de alimentación es seleccionada automáticamente. También se puede alimentar a través del pin 30 mediante un voltaje no regulado entre 6-20 V o 5V regulado.

El chip FTDI FT232RL del Arduino es solo alimentado si la tarjeta está alimentada a través del puerto USB. Como resultado cuando se alimenta externamente (no USB), la salida de 3.3V no está disponible y los LEDs que indican Tx y Rx fluctuaran si los pines digitales 0 y 1 están en alto.

1.4.4. Memoria.

El ATmega328 tiene 32 KB de memoria (2 KB son utilizados para el gestor de arranque, *bootloader*). También dispone de 2 KB de SRAM y 1 KB de memoria EPROM (que puede ser leído y escrito con la librería EEPROM).

1.4.5. Entrada y Salida.

Cada uno de los 14 pines del Mega puede ser utilizado como entrada o salida, usando las funciones *pinMode()*, *digitalWrite()* y *digitalRead()*. Estos pines operan con 5V. Cada pin puede entregar un máximo de 40 mA y tienen una resistencia de pull-up, desconectadas por defecto, de un valor de 20-50 kOhms. En función a esto algunos pines tienen funciones específicas:

Serie: Usado para recibir y transmitir datos TTL¹⁰ modo serie. Los pines 0 y 1 están conectados a los puntos correspondientes del chip serial FTDI USB a TTL, esto permite la comunicación vía serie a través del puerto USB con la computadora..

Tabla 1.5 Relación de pines para la transmisión y recepción serie.

Relación de pines de Trasmisión y Recepción.	
RX	TX
0	1

Interrupciones externas: Estos pines se pueden configurar para disparar una interrupción en un valor bajo, un flanco ascendente o descendente o un cambio en el valor. Para realizar esto se utiliza la función *attachInterrupt()*.

¹⁰ TTL: Lógica transistor-transistor es una tecnología de construcción de circuitos electrónicos digitales.

Tabla 1.6 Relación de pines para las interrupciones externas.

Relación de pines con sus respectivas interrupciones.	
Pines	Interrupciones
2	0
3	1

PWM: Pines 3, 5, 6, 9, 10 y 11. Proveen 8 bit de salida PWM con la función *analogWrite()*.

SPI: 12 (MISO), 11 (MOSI), 13 (SCK), 10 (SS). Estos pines soportan la comunicación SPI a través de la librería SPI como su nombre lo indica.

LED 13: Hay un LED incorporado conectado al pin digital 13. Cuando el pin esta en ALTO, el LED está encendido, cuando el pin esta en BAJO, se apaga el LED.

I²C: A4 (SDA) y A5 (SCL). Soporta la comunicación I2C usando la librería *Wire*.

El Arduino Nano tiene 8 entradas analógicas, cada una de las cuales proveen 10 bits de resolución. Por defecto miden de 0 a 5V, aunque es posible cambiar el límite superior de su rango con el pin de AREF y la función *analogReference()*. Además hay otros pines que tiene funciones específicas como:

AREF. Referencia de voltaje para las entradas analógicas usando la función *analogReference()*.

RESET: Provoca un valor BAJO en la línea que reinicia al microcontrolador. Normalmente es utilizado para añadir un botón de reinicio en la placa.

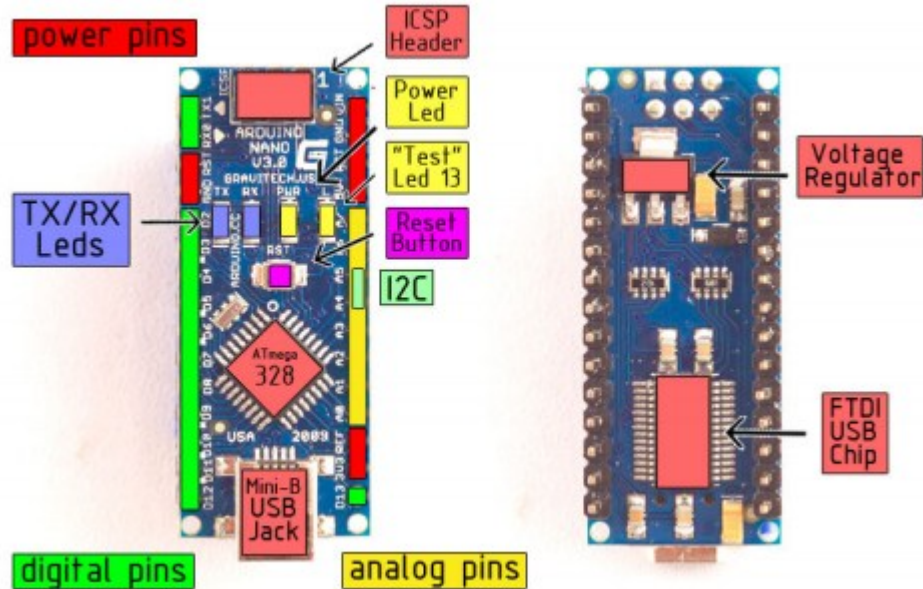


Figura 1.15 Distribución de pines del Arduino Nano.

1.5. Pantalla de Cristal Líquido. (LCD).

Las Pantallas de Cristal Líquido (LCD del inglés, *Liquid Cristal Display*), están altamente difundidas en la actualidad. Son muy útiles porque permiten mostrar información o datos de manera muy clara. La mayoría de los electrodomésticos y diversos equipos electrónicos traen uno o varios de ellos porque presentan la gran ventaja del bajo consumo de potencia.

La magia de los LCD se debe a los “cristales líquidos¹¹”. En sí estas dos palabras suenan contradictorias, pero este material es la razón por la cual este dispositivo funciona.

1.5.1. Tipos de LCD.

Hay varias clases de cristales líquidos, dependiendo de su naturaleza y de la temperatura pueden estar en diferentes fases. Hay dos clases, los termo-trópicos y los lio-trópicos.

Los termo-trópicos reaccionan ante cambios en la temperatura o a la presión. La reacción de los lio-trópicos, los cuales son usados en elaboración de detergentes y jabones, depende del tipo de solvente que se utilice.

¹¹ **Cristal líquido:** Es un tipo especial de estado de agregación de la materia que tiene propiedades de las fases líquidas y la sólida. Algunas de las moléculas nemáticas que presentan los cristales líquidos tienen propiedades ópticas según su orientación permitiendo o impidiendo el paso de la luz o actuando sobre su polarización.

1.5.2. Reflectivos y *Backlight*.

Las moléculas de cristal líquido no son emisoras de luz. Esta división hace referencia al método que se usa para recoger la luz que va a pasar a través de las moléculas de cristal líquido helicoidal. Y depende del costo.

1.5.3. Reflectivos:

Los LCD Reflectivos son los que usan una fuente externa de luz para trabajar. Son muy baratos, se usan en relojes más que todo. El reloj es un buen ejemplo: los números aparecen en donde los electrodos generan una corriente, la cual hace que las moléculas de cristal líquido se alineen y no dejen pasar la luz. La figura x hace referencia a este método:

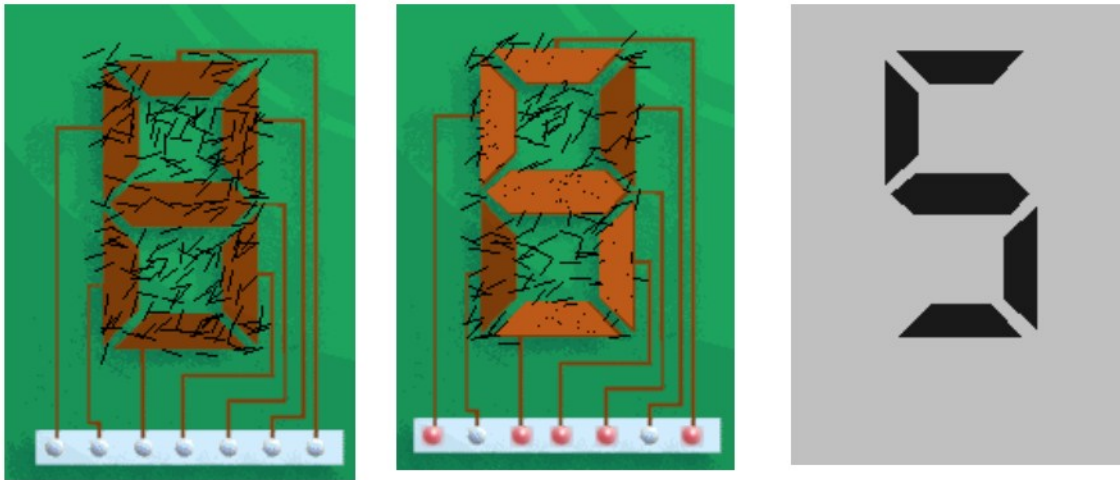


Figura 2.16 Comportamiento de un LCD reflectivo.

En la figura (a) ninguno de los electrodos está generando un campo eléctrico que altere la estructura helicoidal (representados por unas rayas) del cristal líquido. Los sentidos de polarización coinciden, la luz que entra es la misma que sale reflejada por el vidrio trasero y no se forma número alguno. Ya en la figura (b) se han prendido cinco electrodos y sus correspondientes estructuras helicoidales del cristal líquido se destruyen y no dejan pasar luz. Al no haber reflejo de luz, queda más oscuro que el resto de la pantalla. En la figura 2.16 (c), se aprecia el efecto final.

1.5.4. *Backlight*.

Estos LCD no dependen de la luz del ambiente. Los LCD *Backlight* tienen una luz fluorescente en la parte trasera. Esta luz se distribuye uniformemente para asegurar un

patrón de iluminación igual en cualquier punto. Es esta luz la que viaja a través de las moléculas de cristal líquido. Este tipo de LCD es más caro. La figura 50 muestra un corte transversal de un LCD *backlight*:

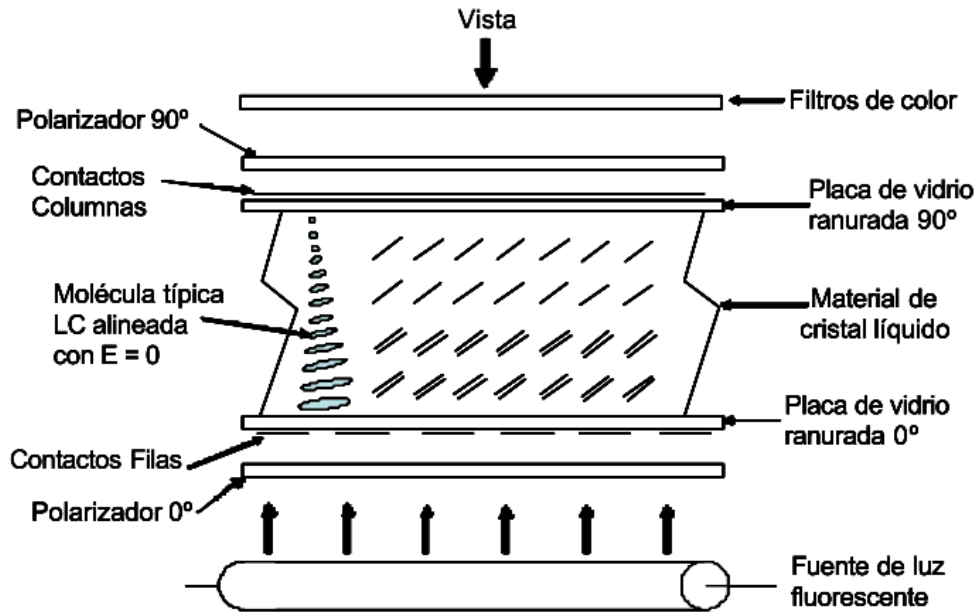


Figura 1.17 Sección transversal de un LCD *backlight*.

1.5.5. Matriz pasiva y matriz activa.

Existe otra clasificación para los LCD. Consiste en el método que usan para excitar las moléculas de cristal líquido.

1.5.6. Matriz pasiva.

Los LCD de matriz pasiva usan una grilla para dar voltaje a un píxel particular en la pantalla. Esta grilla está hecha de dos sustratos de vidrio conductores transparentes. A uno de los sustratos se le asignan las columnas y al otro las filas. Estas filas y columnas están conectadas a un circuito integrado a través de un electrodo, que controla a qué columna o fila se le debe aplicar voltaje y aterrizar a tierra. Entre estas dos placas de sustrato va el cristal líquido. Este se presenta en la figura 1.18:

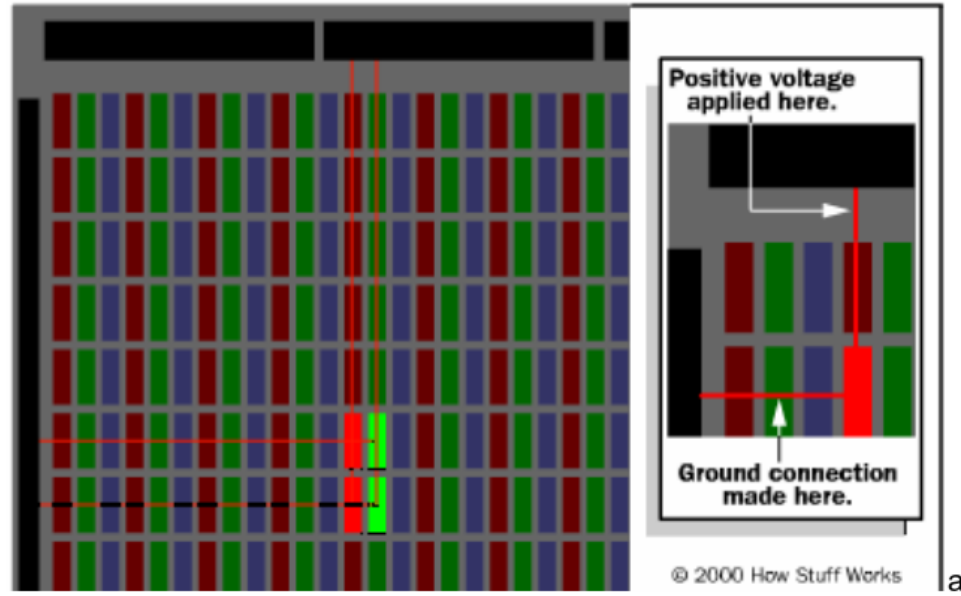


Figura 1.18 Modo de encendido de píxeles.

Este método es muy práctico pero tiene dos inconvenientes:

- Bajo tiempo de respuesta: se refiere a que el LCD se demora mucho tiempo en refrescar la imagen que se muestra. Se manifiesta en “fantasmas” o remanencia de imagen al generar una animación.
- Control impreciso de voltaje aplicado: hace referencia al efecto de aplicar un voltaje a un píxel determinado y que los cristales líquidos circundantes se vean afectados también y alcanzan a “desenroscarse” un poco. Se manifiesta en una imagen borrosa y de mal contraste.

Estos problemas se solucionen con un LCD de matriz activa.

1.5.7. Matriz activa

Este tipo de LCD depende de unos transistores especiales llamados en inglés *Thin Film Transistors* (TFT). Son pequeños transistores y condensadores que trabajan como un conmutador y se sueldan en forma de matriz en un sustrato de vidrio. La estructura interna de un LCD de matriz activa se muestra en la figura 1.19.

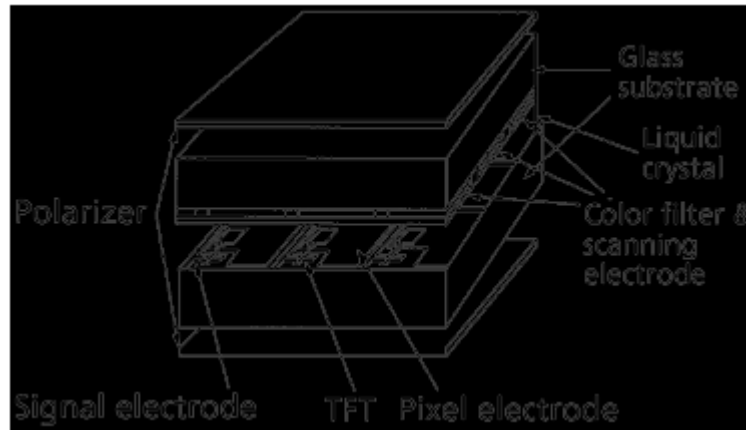


Figura 1.19 Estructura interna de un píxel en una matriz activa.

Cuando se ordena a un determinado píxel que se prenda, el sistema de direccionamiento identifica el transistor correspondiente a la columna del píxel que se necesita encender. Por medio del transistor se envía la carga a la columna. Como todas las filas que interceptan las columnas están a potencial cero, sólo el condensador del píxel designado recibe la carga. El condensador retiene la carga hasta el próximo ciclo de refrescamiento. El transistor permite controlar el voltaje que se envía al píxel permitiendo variar la cantidad de luz que puede pasar. De esta forma se crea una escala de grises.

Este tipo de LCD presenta la ventaja de tener un mejor contraste y una respuesta más rápida. Además, es más exacto en la asignación de voltaje a un píxel individual.

1.5.8. Pantalla táctil. (*Touch screen*).

Una pantalla táctil resistiva consiste al menos en tres capas: una membrana flexible, la cual, se coloca sobre un sustrato de vidrio o acrílico. Ambas superficies están recubiertas de una capa conductora transparente. Estas capas están separadas por un espaciador, de esta manera no hay conexión eléctrica a menos que se aplique una presión sobre la capa superior, cerrando el circuito.

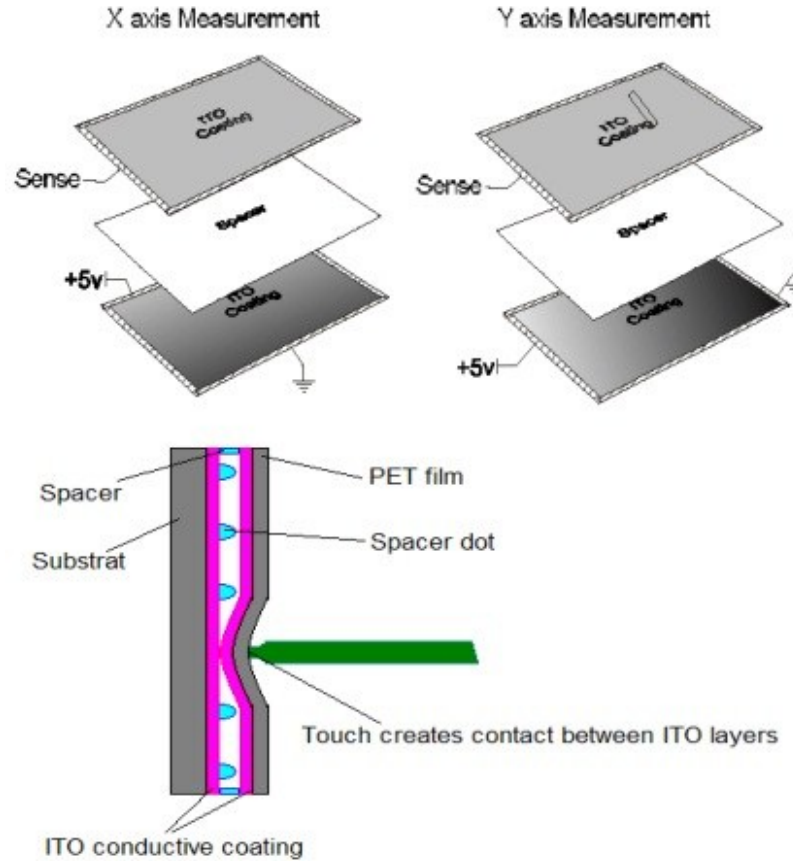


Figura 1.20 Touch panel.

1.5.9. Pantalla táctil de 4 cables.

Como las pantallas táctiles son una red resistiva al presionar la superficie se cierra el circuito. Los hilos se conocen como X+ (izquierda), X-(Derecha), Y+ (superior) e Y- (abajo). Una ventaja de la pantalla táctil de 4 hilos es la posibilidad de determinar el tacto con presión mediante la medición del contacto resistencia entre las dos capas.

Coordenada X

X+ = GND , X- = Vcc, Y-= ADC.

Coordenada Y

Y+ = GND , Y- = Vcc, X-= ADC.

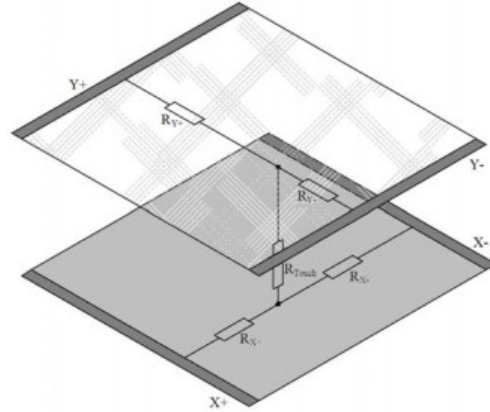


Figura 1.21 Pantalla táctil de 4 cables.

1.5.10. Pantalla táctil de 5 cables.

Las pantallas táctiles de 5 cables también consiste en un arreglo de resistencias pero el método de medición se realiza por el mismo terminal.

Coordenada X:

UL,LL = GND,

UR,LR = Vcc, Sense= ADC.

Coordenada Y:

UL,UR = GND ,

LL,LR = Vcc, Sense= ADC.

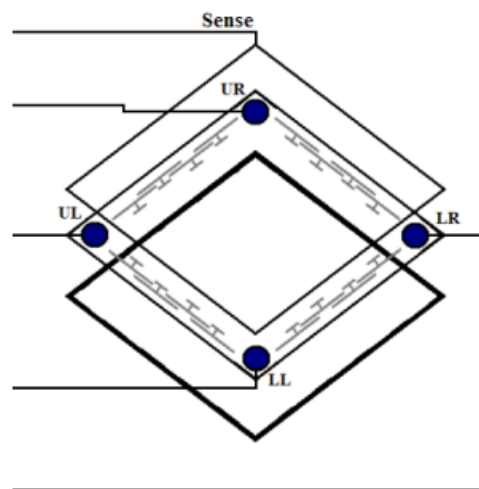


Figura 1.22 Pantalla táctil de 5 cables.

1.6. Controlador Ethernet, ENC28J60.

El módulo Ethernet ENC28J60 que utiliza el controlador que lleva su mismo nombre de Microchip ofrece una serie de características para manejar una red y dotar de conectividad al Arduino o acoplado directamente a la mayoría de los microcontroladores existentes en el mercado mediante una interfaz estándar SPI. La velocidad de transferencia de datos puede alcanzar hasta 20 MHz.

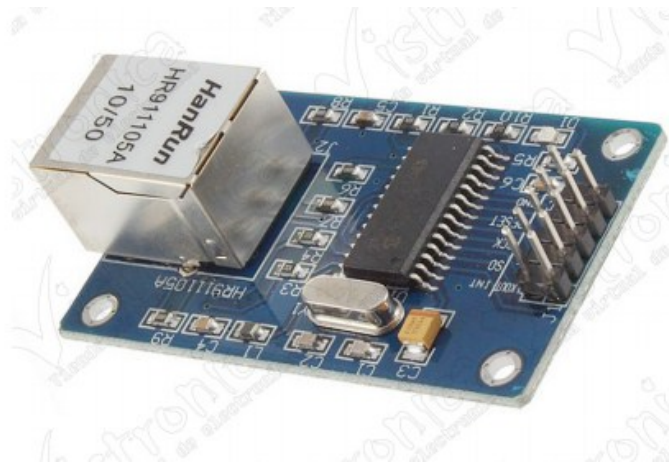


Figura 1.23 ENC28J60.

Esta pequeña tarjeta es básicamente un controlador Ethernet dotado de un puerto SPI (*Serial Peripheral Interface*), diseñado especialmente para actuar como un puente entre una red Ethernet y un microcontrolador equipado con SPI. El ENC28J60 cumple con las especificaciones IEEE 802.3, e incorpora una serie de esquemas de filtrado de paquetes para limitar el número de paquetes entrantes. También provee un módulo DMA interno para facilitar el flujo de datos y hardware específico para el cálculo de las sumas de control (*IP checksums*). La comunicación con el microcontrolador que hace las funciones de *host* se realizan mediante el bus SPI a una velocidad de hasta 10Mb/s y dos pines especiales capaces de provocar interrupciones en el ENC28J60. Otros dos pines pueden utilizarse para manejar LEDs que indiquen la conexión y la actividad de la red. Este módulo se encarga de la conexión de sensores, relés, entre otros dispositivos al Arduino; es muy interesante pero para poder consultar todo esto desde una interfaz sencilla y a distancia es fundamental poder servir todos estos datos a través de una web. Para esto se necesita permanecer conectado a una red y se necesita de una interfaz Ethernet. Los

módulos Ethernet para Arduino son uno de los componentes más interesantes con los que se pueden trabajar. Gracias a sus características, son más económicos y pequeños, permitiendo de una forma económica montar un servidor web. Como contrapartida decir que no están soportado por Arduino de forma estándar aunque cada vez hay más librerías y mejor terminadas.

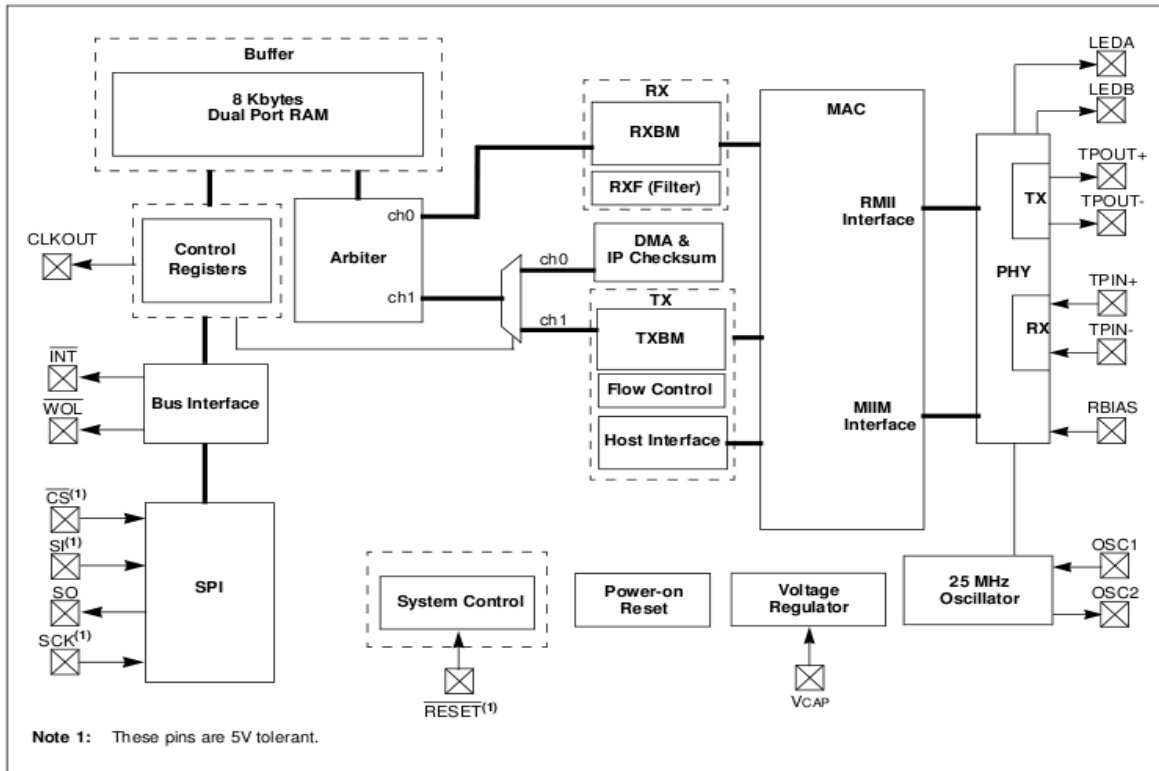


Figura 1.24 Diagrama en bloques del ENC28J60.

El esquema de la figura 1.25 corresponde a una aplicación típica del ENC28J60:

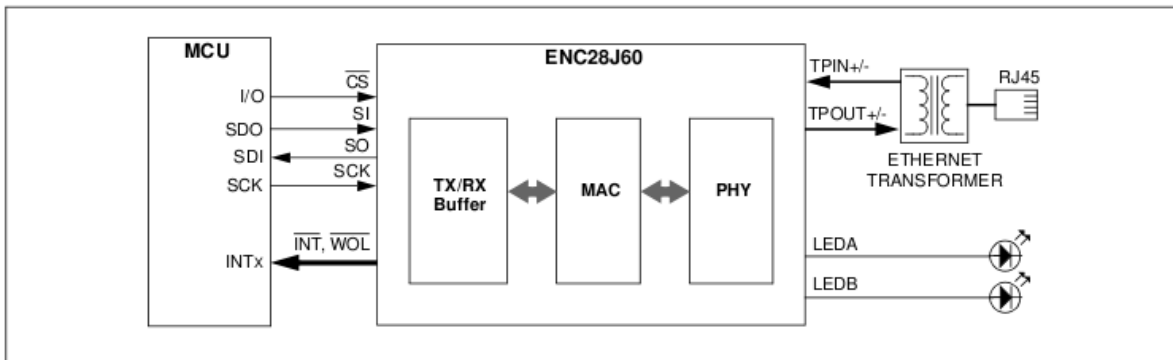


Figura 1.25 Circuito de aplicación típico.

Además del integrado, puede verse que se utilizan dos transformadores de pulso y varios componentes pasivos indispensables para conectar el ENC28J60 a una red Ethernet de 10Mb/s. El ENC28J60 tiene siete bloques funcionales importantes. Además de ello, el dispositivo cuenta con otros bloques de soporte, como el oscilador, regulador de voltaje interno, adaptadores de nivel para proveer E/S de 5V y lógica de control.

1.6.1. Principales Características.

Tabla 1.7 Principales características del controlador Ethernet ENC28J60.

Microcontrolador	ENC28J60
Voltaje de Operación	3.3 V
Interfaz	SPI (compatible con SmartM51 y AVR)
Reloj SPI	20 MHz
Máxima Taza de transferencia.	10 Mbps
Frecuencia del reloj	25 MHz
Conector	Estándar 2x5 RJ45
Compatibilidad	10/100/1000Base-T
Soporta	Modo <i>Full-Duplex</i> y <i>Half-Duplex</i>
Tamaño	5.9 cm x 3.5 cm

CAPITULO 2 . CENTRO DE PROCESAMIENTO DE DATOS DE LA UNIVERSIDAD DE ORIENTE, CORPUS.

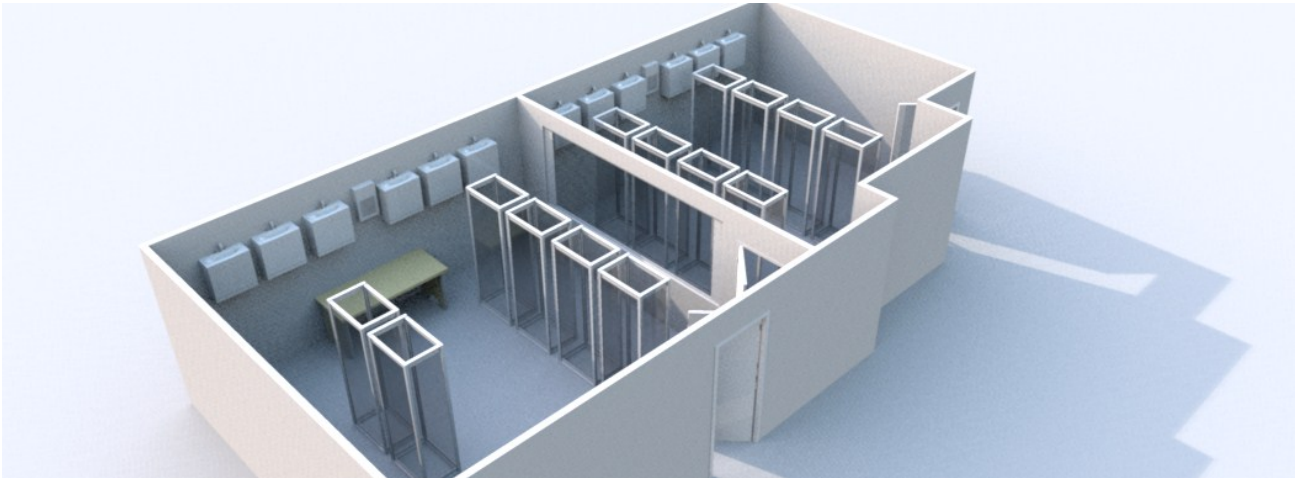


Figura 2.1 CPD de la Universidad de Oriente, vista 3D.

Se denomina centro de procesamiento de datos a aquella ubicación donde se concentran todos los recursos necesarios para el procesamiento de la información de una organización.

Un CPD (Centro de Procesamiento de Datos) es un edificio o sala de gran tamaño usada para mantener en él una gran cantidad de equipamientos electrónicos. Suelen ser creados y mantenidos por grandes organizaciones con objeto de tener acceso a la información necesaria para sus operaciones.

Entre los factores más importantes que motivan la creación de un CPD se pueden destacar el garantizar la continuidad del servicio a clientes, empleados, ciudadanos, proveedores y empresas colaboradoras, pues en estos ámbitos es muy importante la protección física de los equipos informáticos o de comunicaciones implicadas, así como servidores de base de datos que puedan contener información crítica.

El diseño de un centro de procesamiento de datos comienza por la elección de su ubicación geográfica y requiere un balance entre diversos factores. Aun cuando se disponga del local adecuado siempre es necesario algún despliegue de infraestructura en su interior, como son los falsos techos y falsos suelos, cableado de red y teléfono, doble cableado eléctrico, instalaciones de alarmas, control de temperatura, humedad con avisos SNMP o SMTP, etc.

Generalmente todos los grandes servidores se suelen concentrar en una sala denominada “sala fría”, “nevera”, o “pecera”. Esta sala requiere de un sistema específico de refrigeración para mantener la temperatura baja (entre 21 y 23 grados centígrados), necesaria para evitar averías en las computadoras a causa del sobrecalentamiento, según las normas internacionales la temperatura exacta debe ser 22,3 grados centígrados.

La pecera suele contar con medidas estrictas de seguridad en el acceso físico, así como medidas de extinción de incendios adecuadas al material eléctrico.

Cada día se encuentran disponibles en el mercado diferentes dispositivos para realizar el control y monitoreo de los parámetros imprescindibles en un CPD, así como los parámetros eléctricos que permiten el perfecto funcionamiento del mismo, en algunos casos suelen ser diseñados de acuerdo a las necesidades del cliente, como es el caso de este trabajo, siendo una de las vías, para su ejecución el uso de microcontroladores en nuestro caso la plataforma Arduino.

2.1. Caracterización del local de servidores de CORPUS.

En el CPD de la Universidad de Oriente se desarrollan e implementan soluciones relacionadas con las Tecnologías de la Información y las Comunicaciones, en función de elevar la calidad, eficiencia y eficacia de la gestión de los procesos universitarios, en favor de mejorar la formación integral de los estudiantes, la superación académica y la producción científica de la universidad.

En este centro se ofrecen servicios como:

- Desarrollo de proyectos en Redes y Telemática
- Diseño, desarrollo y explotación de intranets empresariales
- Desarrollo de Sistemas de Información para Intranet/Internet

- Desarrollo de Sitios Web en Intranet/Internet
- Desarrollo de Sistemas de Tratamiento de la Información y/o Toma de Decisiones de amplio espectro
- Servicios de proyectos, auditorías y supervisión de instalaciones de redes
- Asistencia Técnica
- Formación

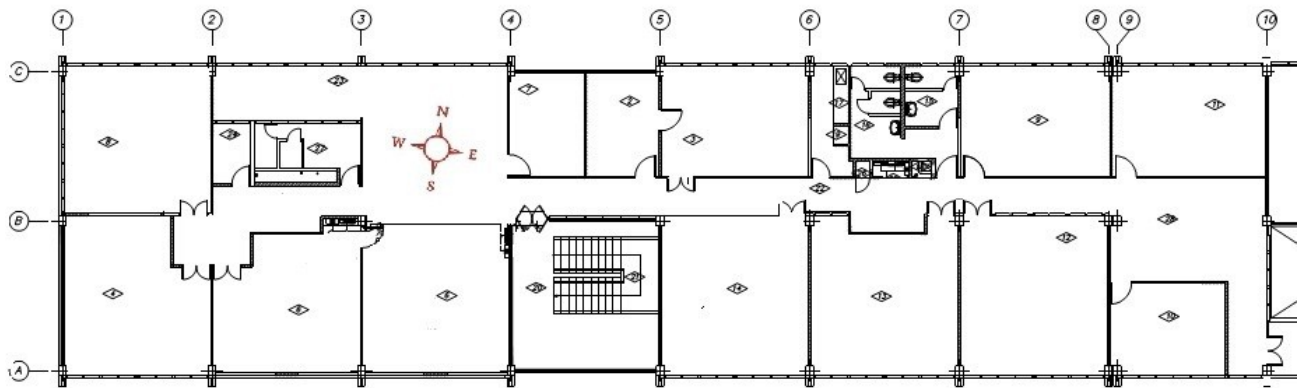
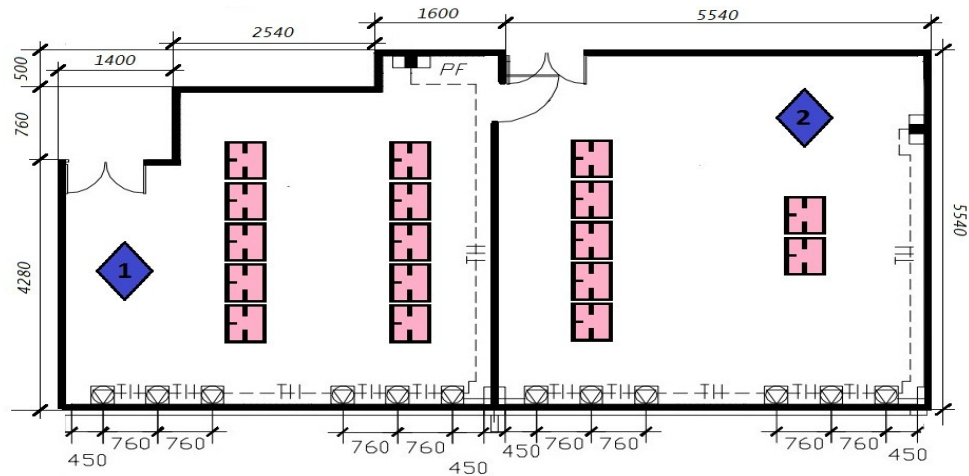


Figura 2.2 Vista superior de CORPUS.

2.1.1. Descripción física y ubicación.

El local de servidores ocupa un área total de 58.38 m², las habitaciones uno y dos tienen 27.7 m² y 30.69 m² respectivamente. Dentro de los equipos tecnológicos que dispone se encuentran seis equipos de climatización o aires acondicionados de 1 ton de refrigeración cada uno, armarios de telecomunicaciones, servidores, *routers*, *switch*, entre otros medios de comunicaciones. Sin embargo, no existe un sistema de control y registro automatizado de los parámetros más importante para mantener el correcto funcionamiento del CPD, tales como, la temperatura, el voltaje de la línea, la corriente y el consumo de los equipos de climatización. Estas variables deben ser supervisadas y controladas de acuerdo a los parámetros de calidad permisibles para cada uno. Las dimensiones físicas, ubicación de cada local y los equipos del mismo se observan en la figura.



LEYENDA

- ☒ AIRES ACONDICIONADOS
- MINICANAL DE PVC CON SEPARADOR 3 VIAS.
- TH CONDUCTORES DE TIERRA, FASE Y NEUTRO.
- PANEL ELÉCTRICO DE FUERZA (PF)
- ARMARIO DE TELECOMUNICACIONES

NOTA

LAS COTAS ESTÁN EXPRESADAS EN mm.

Figura 2.3 Descripción arquitectónica de los locales de servidores.

2.2. Riesgo de los Centros de Procesamiento de datos.

Debido a la evolución de las tecnologías, de los servicios y de los entornos empresariales en general, la información se ha convertido quizás en el primer patrimonio de las empresas. De ahí que se pueda asegurar que la Seguridad en los Centros de Procesamientos de Datos (C.P.D.) es una necesidad impuesta a toda entidad o empresa de cualquier rango.

La seguridad constituye, por consiguiente, uno de los principales problemas en todo sistema de procesamiento de datos; la expansión de los sistemas informáticos hace que sea imprescindible la implantación de nuevos elementos de seguridad que protejan de una forma adecuada estos entornos.

La seguridad del Centro de Procesamiento de Datos hace referencia a los riesgos que afectan a las instalaciones donde se encuentra el mismo y a las soluciones que han de adoptarse para su protección.

La clasificación general de estos riesgos contemplados desde la posibilidad de que ocurra un evento/siniestro en función de los recursos y el entorno, es la siguiente:

Siniestros en:

- Recintos y edificio
- Instalaciones auxiliares
- Equipos o hardware
- Software

Mapa de riesgos:

- Incendio
- Gases
- Explosión/implosión
- Daños por agua
- Rayo. Tensión inducida
- Sobretensiones/cortocircuitos
- Robo. Hurto
- Actos vandálicos
- Avería de componentes
- Cambio de condiciones ambientales
- Virus informático
- Modificaciones o espionaje de datos
- Pérdidas de datos/copias de seguridad

Algunos de estos riesgos representan una problemática diferente, que debe ser tratada individualmente y de acuerdo con las peculiaridades de cada instalación.

2.3. Protección de servicio técnico.

Las computadoras y equipos electrónicos son particularmente susceptibles al daño causado por los productos de calor, vapor, humedad, fluctuaciones incontroladas de red de energía eléctrica.

Daños a ciertos componentes electrónicos pueden ocurrir a temperaturas como 39 °C (100 °F) con un daño permanente como resultado de la exposición a temperaturas superiores a aproximadamente 49 °C (120 ° F). Daño a las cintas magnéticas, discos flexibles y medios similares puede comenzar a temperatura ambiente sostenidos por encima de 37.8 °C (100 °F), y temperaturas ambiente sostenidos de 66 °C (150 ° F) pueden dañar los discos duros.

Los componentes electrónicos son también susceptibles a los daños de humo, el hollín y las partículas corrosivos producidos por un incendio. Por ejemplo, las unidades de disco son susceptibles a daños causados por partículas tan pequeñas como 0,5 micras de diámetro.

2.3.1. Suministro de Energía Eléctrica.

Es importante tener contratadas dos compañías suministradoras de electricidad independientes para evitar así la falta de alimentación. En caso de no disponer de una segunda fuente de energía externa, se debe salvaguardar la continuidad de la corriente mediante baterías que aseguren automáticamente y sin interrupción el funcionamiento adecuado de la instalación al menos durante 60 horas o bien grupos electrógenos. En caso de fallo de energía, se deben iluminar las salas con luces de emergencia.

En los casos en los que una interrupción del ordenador suponga grandes pérdidas económicas, se utilizarán cables F3 capaces de suministrar corriente incluso cuando ocurra un incendio. Estos cables serán utilizados para la alimentación del ordenador, la instalación de aire acondicionado y la instalación de protección contra incendios.

Si este suministro falla, el sistema queda totalmente fuera de juego inmediatamente y durante el tiempo que el fallo dure, pudiendo también verse afectados los sistemas de aire acondicionado y de protección de incendios. Los paros en el acondicionamiento del aire pueden originar pérdidas de información, que pueden llegar a ser parciales o totales,

temporales o definitivas, en discos y cintas. Por supuesto que la pérdida total de suministro no es la única fuente de problemas: variaciones de voltaje o frecuencia, por encima de los valores especificados.

Las instalaciones reciben su alimentación de los suministros públicos de electricidad, y debe considerarse la posibilidad de fallos de ese suministro debido a daños accidentales en las subestaciones, cables subterráneos, daños por tormentas en líneas aéreas, excesos de carga en casos de fuerte demanda o, incluso, acciones terroristas contra el sistema de alimentación. Algunas perturbaciones pueden ser de tan corta duración que son muy difíciles de detectar y de relacionar con fallos en el funcionamiento de los equipos. Para detectar variaciones transitorias se requiere el uso de equipos especiales para controlar la alimentación y registrar las perturbaciones. Algunas de las causas posibles de perturbación en el suministro incluyen:

- Reducciones en el voltaje en la frecuencia en los momentos de alta demanda, por periodos de pocas horas.
- Reducciones en el voltaje debidos a fuertes corrientes producidas por plantas eléctricas cercanas, como sucede durante el arranque de ciertos tipos de motores eléctricos, para cortos periodos, de algunos ciclos de la alimentación hasta algunos segundos.
- Transitorias durante algunos pocos ciclos en la alimentación, tales como relámpagos repentinos sobre las líneas de alimentación, o la operación de dispositivos eléctricos, tales como motores de ascensores o equipos de aire acondicionado.
- Introducción de voltajes armónicos en la alimentación de operación de equipos eléctricos cercanos.

Si hay posibilidades de que el suministro sea de objeto de perturbaciones, puede ser necesario disponer de una fuente de alimentación no sujeta a la influencia de las perturbaciones, considerando incluso la posibilidad de instalar filtros o un motor-alternador que actúen como un amortiguador entre el suministro y la computadora que pueden vencer variaciones en la alimentación que tengan duraciones por debajo de aproximadamente 100 milisegundos. Para variaciones de más larga duración, se deben tomar medidas especiales, tales como la incorporación de un volante en el juego motor - alternador. Para perturbaciones más largas o incluso interrupciones, deben considerarse

algunas formas de fuentes alternativas de energía. El sistema más completo y más complejo es el que se denomina habitualmente SAI¹² (Sistema de Alimentación Ininterrumpida), que es una unidad de conversión de energía eléctrica que proporciona corriente alterna de alta calidad. Acepta diversos suministros de energía de entrada, dentro de unos parámetros especificados, y los convierte en la energía de salida necesaria para el equipo de proceso de datos, dentro de los parámetros que éste precisa. Las entradas de energía aceptables por SAI incluyen los suministros de las compañías, generadores locales o baterías. Un SAI de tipo estático se compone básicamente de un rectificador y un inversor. La corriente alterna de entrada se rectifica y alimenta al inversor y a la batería.

2.3.2. Aire Acondicionado.

Debido a la permanente facilidad a dañarse de los equipos electrónicos cuando se exponen al calor, es conveniente su localización en salas debidamente acondicionadas en las cuales se realice un control riguroso de la temperatura y humedad ambiente con instalaciones de acondicionamiento. Las unidades del C.P.D. suelen disponer de indicadores o sensores de las variaciones anormales de temperatura y humedad, que desconectan el equipo electrónico, evitando así posibles daños.



Figura 2.4 Imagen de un CPD.

¹² **SAI:** Sistema de alimentación ininterrumpida también conocido por sus siglas en inglés UPS, es un dispositivo que gracias a sus baterías pueden proporcionar energía eléctrica tras un apagón. Filtra las subidas y bajadas de la tensión mejorando la calidad de la energía eléctrica y elimina los armónicos de la red en caso de ser AC.

Los conductos de ventilación representan un riesgo importante de extensión del incendio y de humo a sectores distintos de los de origen. Por ello deben estar formados por elementos resistentes al fuego. Para evitar la propagación del incendio dentro del propio conducto cada vez que atraviesen un sector de compartimentación se deberá disponer de trampillas cortafuego. En caso de incendio, la instalación de aire acondicionado será rápidamente desactivada, bien manualmente o bien mediante la instalación de detección automática, las compuertas cerrarán automáticamente cuando la instalación automática de extinción entre en funcionamiento. Los filtros de aire serán incombustibles y estará equipado con un sistema que detecte posibles embasamientos. En cada local de servidores, se instalará un panel eléctrico de fuerza para la puesta en marcha y parada del sistema de aire acondicionado.

2.4. Parámetros que se van a medir.

Se debe diseñar un sistema de adquisición de datos para la supervisión de la temperatura y el voltaje de la línea en el nodo central de la Universidad de Oriente, con el objetivo de proteger los equipos de enfriamiento que están en el local, frente a variaciones de tensión y registrar tales fluctuaciones, además mantener la temperatura del local aproximadamente constante con la utilización de estos aires acondicionados por cada una de las dos habitaciones de servidores. En cada habitación estará trabajando por lo general dos aires acondicionados y los otros entrarán en funcionamiento en caso de elevarse la temperatura del local por encima de los 23°C y vuelven a apagarse una vez que esté por debajo de los 21°C. Para evitar que los equipos de enfriamiento trabajen constantemente durante las 24 horas, el sistema debe prever la conmutación de los mismos de modo que trabajen solo un tiempo mientras las otras estarán de reserva. El sistema debe además proteger estos aires acondicionados de los cambios bruscos de la tensión de la línea, una vez que esta se salga de los parámetros permisibles, los cuales son 190 V para el límite inferior o bajo voltaje y 250 V para alto voltaje. Además se deben registrar tales valores junto a la fecha, la hora y el día en que se efectuaron para poder utilizar estos datos a la hora de que sea necesario el reclamo a la empresa de suministros eléctricos. Una vez registrados todos los parámetros se deben visualizar en un *display* LCD gráfico, y crear un software para el constante monitoreo de tales valores y comprobar el perfecto funcionamiento del local de procesamiento de datos de la universidad. Para la comunicación entre el sistema y el administrador de la red se utilizará el protocolo TCP-IP a través del módulo de Ethernet ENC28J60, usando esta tarjeta como servidor Web.

CAPITULO 3 . DISEÑO Y PUESTA A PUNTO DEL SISTEMA DE SUPERVISIÓN Y CONTROL.

3.1. Diseño del *hardware*.

3.1.1. *Hardware* del sistema.

Para tener una visión general del hardware diseñado, para la supervisión y registro de la temperatura y el voltaje de la línea del local de servidores del Nodo Central de la Universidad de Oriente, se ha dividido en varias partes para lograr un mejor entendimiento del sistema. En la figura 3.1 se muestra un diagrama en bloques del hardware implementado.

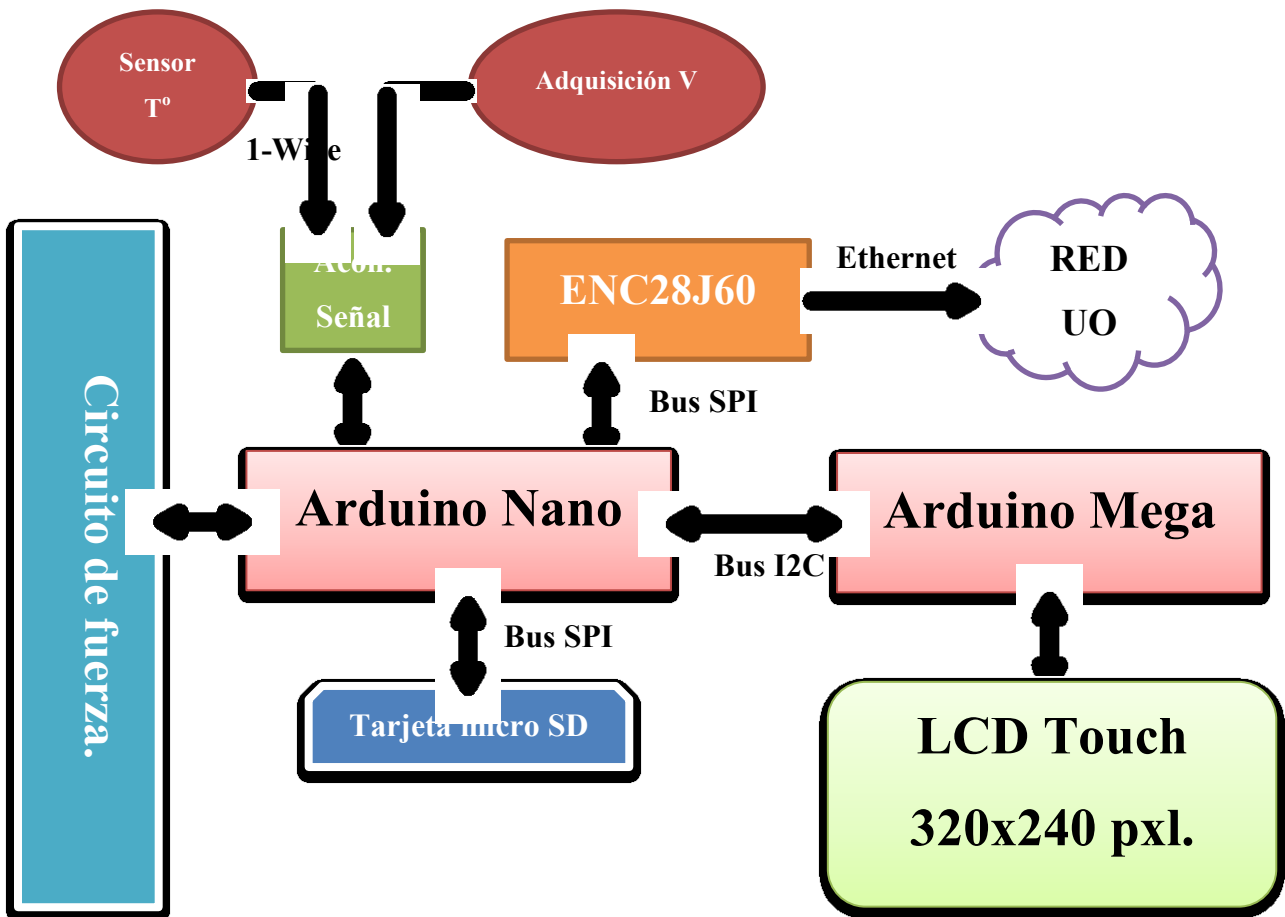


Figura 3.1 Diagrama en bloque del hardware del sistema implementado.

De acuerdo a la figura anterior el hardware consta de las siguientes partes:

- 1. Sensor de Temperatura (DS1820):** Integrado encargado de convertir la magnitud física de la temperatura a un valor digital, se comunica con el micro a través del bus 1-Wire.
- 2. Tarjeta Arduino Nano:** Realiza la lectura de las mediciones de los sensores, procesa esa información y realiza las funciones pertinentes de acuerdo a la interpretación de las mismas. Configura el controlador Ethernet ENC28J60, y realiza la escritura y la lectura de la memoria micro SD, en la cual se guardan los datos capturados junto a la fecha y la hora a la que se tomaron las muestras.
- 3. Tarjeta Arduino Mega:** Realiza la función de controlador del display LCD, y configura la visualización en el mismo, atendiendo a la pantalla táctil que le permite al usuario interactuar con el sistema.
- 4. Controlador Ethernet (ENC28J60):** Permite enviar a través de una conexión Ethernet los datos adquiridos por la tarjeta Arduino Nano, para la supervisión de los parámetros medidos por parte del administrador de la red o el encargado de dicha función.
- 5. Circuito de Fuerza:** Se encarga de poner a funcionar los aires acondicionados que mantendrán la temperatura en un rango predefinido por el usuario, la tarjeta Arduino Nano es la encargada de darle la orden del encendido o apagado de los aires acondicionados.

3.1.2. Sistema de adquisición de datos.

Un sistema de adquisición de datos es un equipo que permite tomar señales físicas del entorno y convertirlas en datos para llevarlas a un computador donde se puede procesar y presentar dichas señales. ^[6]

La mayor parte de sistemas de adquisición de datos hoy en día son basados en computadores o en algún tipo de microprocesador que tenga un conversor Analógico-Digital. En la figura 3.2 se muestran los bloques que componen un sistema de adquisición de datos:



Figura 3.2 Esquema en bloques de un sistema de adquisición de datos.

3.1.3. Sensores.

Sensor es el elemento que se encuentra en contacto directo con la magnitud física que se va a evaluar. Estos dispositivos pueden transformar las magnitudes físicas o químicas en una señal de salida normalizada (normalmente tensión o corriente) las cuales pueden ser procesadas por el sistema de adquisición de datos.

Para la implementación y desarrollo del presente trabajo se empleó el sensor de temperatura DS1820.

3.1.3.1. Sensor de Temperatura. DS1820.

El sensor de temperatura DS1820 provee de 9 bit de resolución los cuales dan información de la temperatura en grados centígrados. Este sensor utiliza el protocolo de comunicación 1-Wire, que como su nombre lo indica solo requiere de una línea de datos para la comunicación con un microprocesador central. El DS1820 tiene un rango de operación de -55°C a 125°C con una precisión de $\pm 0.5^{\circ}\text{C}$ entre los -10°C y 85°C .

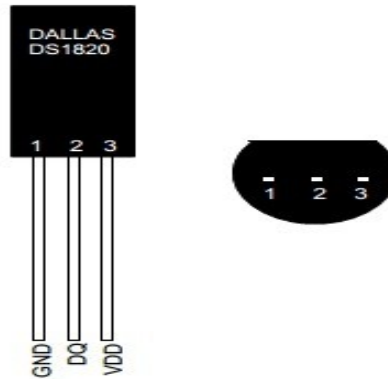


Figura 3.3 Encapsulado del sensor DS18B20

Sobre el sensor DS18B20...

- Es un termómetro digital de alta precisión, entre 9 y 12 bits de temperatura en grados Celsius (el usuario puede escoger la precisión deseada).
- Su temperatura operativa se encuentra entre -50 y 125 grados Celsius. La precisión, en el rango comprendido entre -10 y 85 grados es de ± 0.5 grados.
- Su precio es económico, su interfaz de funcionamiento es sencilla y su uso es muy provechoso para proyectos que requieran mediciones precisas y confiables.

3.1.3.2. Operación del sensor DS1820.

El núcleo funcional del DS1820 es directamente el sensor digital de temperatura. El sensor de temperatura provee una salida de 9 bit de resolución los cuales corresponden a saltos de $0,5^{\circ}\text{C}$.

Los datos de salida del DS1820 están calibrados en grados Celsius, para usarlos en Fahrenheit se debe hacer una rutina de conversión.

Con Arduino podemos “leer” la temperatura que registra este sensor que posee una característica muy peculiar. Utiliza la comunicación OneWire, la cual es un poco complicada para los que no tienen mucha experiencia en el área de electrónica digital.

Básicamente se trata de un protocolo especial que permite enviar y recibir datos utilizando un solo cable, a diferencia de la mayoría de los protocolos que requiere dos vías. De hecho, Arduino posee los pines RX y TX que son los encargados de enviar y recibir información. Los pines VCC y GND deben ir conectados entre sí. La alimentación se introduce en el pin DATA, por medio de una resistencia *pull-up* que debe ser igual o mayor a 4.7 KOhm.

La mayoría de las veces que no se consigue leer este sensor es debido a una mala conexión. El código requiere de la utilización de dos librerías, que deben ser instaladas antes de cargar el código a la placa. Estas son DallasTemperature.h y OneWire.h.

3.1.4. Acondicionamiento de la señal de Voltaje.

Para adquirir la señal de voltaje se utiliza el esquema presentado en la figura 3.4, es un circuito acondicionador de nivel que convierte el voltaje alterno de la línea en voltaje de directa pulsante con un nivel adecuado para la entrada analógica del microcontrolador, en este caso el Arduino Nano, el CAD de este micro convierte el voltaje de la entrada referenciado a 5 V, pero esta referencia puede ser cambiada en el pin de entrada Vref para mejorar la resolución del mismo, en otras palabras disminuir el LSB del CAD.

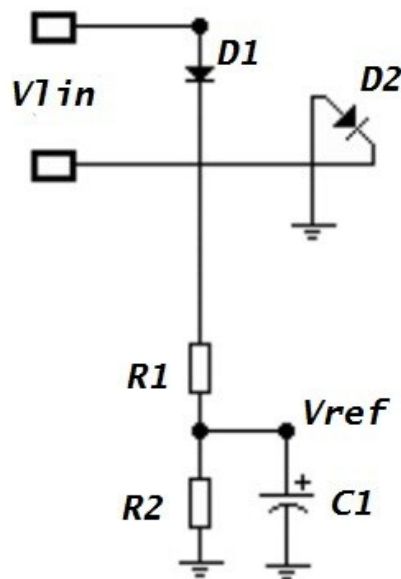


Figura 3.4 Circuito acondicionador de la señal de voltaje.

Vref es el voltaje equivalente al voltaje de la línea en este caso se adecúa la señal de 220 V hasta un nivel aceptable para el CAD del Micro. Con el circuito mostrado en la figura anterior se obtiene la siguiente tabla de mediciones, en la cual solo se muestran los valores más relevantes puesto que si está por encima o por debajo de estos el sistema sigue funcionando de la misma manera hasta que no se restablezca nuevamente el voltaje de la línea:

Tabla 3.1 Equivalencia del voltaje de la línea con el voltaje de referencia.

V (Línea)	Vref (V)
265	3,58
255	3,45
245	3,31
235	3,18
225	3,04
220	2,97
215	2,9
205	2,77
195	2,64
185	2,5

Para lograr estos niveles de voltaje se usó una resistencia de 650 kOhm y otra de 20 kOhm para R1 y R2 respectivamente. El capacitor debe escogerse de tal manera que la carga y descarga se realicen en un tiempo muy pequeño de manera tal que pueda seguir

las variaciones de voltaje de la línea, en este caso se escogió un capacitor de 10 uF que cumple con dichas condiciones.

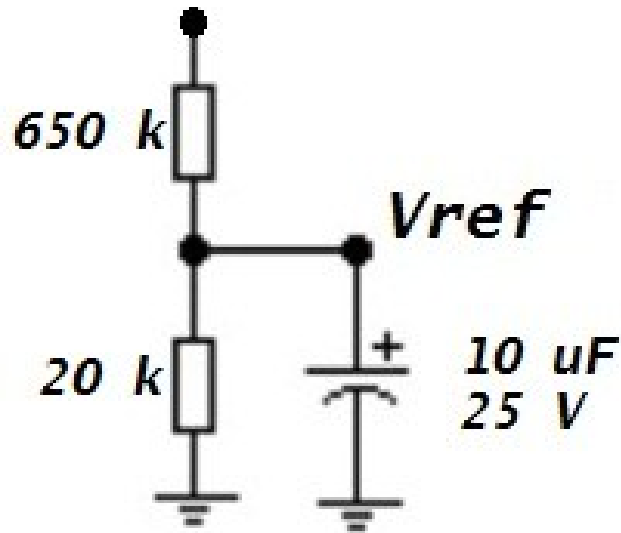


Figura 3.5 Valores de los componentes del circuito acondicionador.

3.1.5. Conversión Analógica Digital.

La tarjeta Arduino Nano lee el valor de tensión en el pin analógico declarado. Esta posee 8 canales, como ya se había visto en capítulos anteriores, conectados a un conversor analógico digital de 10 bits. Esto significa que convertirá tensiones entre 0 y 5 voltios a un número entero entre 0 y 1023. Esto proporciona una resolución en la lectura de: 5 voltios / 1024 unidades, es decir, 0 .0049 voltios (4.9 mV) por unidad. (**LSB** = 4.9 mV).

El conversor tarda aproximadamente 100 microsegundos en leer una entrada analógica por lo que se puede llevar una tasa de lectura máxima aproximada de 10000 lecturas por segundo.

La función que se utiliza para la lectura de las entradas analógicas es: **analogRead (pin)**.

Donde, pin indica el número del pin de la entrada analógica que deseamos leer (0 a 5 en la mayoría de las tarjetas, de 0 a 7 en la tarjeta Arduino Nano y de 0 a 15 en la tarjeta Arduino Mega).

La función **analogReference (tipo)** (donde, tipo indica el tipo de referencia que se desea usar: *DEFAULT*, *INTERNAL* y *EXTERNAL*) configura el voltaje de referencia

usado por la entrada analógica. La función **analogRead ()** devolverá un valor de 1023 para aquella tensión de entrada que sea igual a la tensión de referencia. Las opciones son:

- a) *DEFAULT*: Es el valor de referencia analógico que viene por defecto que es de 5 voltios en tarjetas Arduino y 3.3 voltios en otros modelos de tarjetas Arduino que funcionan con dicho voltaje.
- b) *INTERNAL*: Es una referencia de tensión interna de 1.1 voltios en el ATmega168 o ATmega328 y de 2.56 voltios en el ATmega8.
- c) *EXTERNAL*: Se usará una tensión de referencia externa que tendrá que ser conectada al pin AREF.

3.1.6. Pantalla Táctil de 320x240 píxeles.

Es bastante conocida la implementación de pantallas gráficas TFT con la posibilidad de poseer *Touch Screen*, existen muchas librerías para implementar su uso y difieren desde su propia complejidad hasta la plataforma para la cual son escritas. En esta ocasión vamos a trabajar con la plataforma Arduino Mega y un TFT Mega Shield el cual lleva conectado una pantalla *TFT 320x240 + Touch Screen + SD Card* y la librería a utilizar es la UTFT cuya autoría pertenece a Henning Karlsen.



Figura 3.7 Pantalla táctil de 320x240 píxeles.

El fabricante de este módulo es *Elec Freaks* y los productos que constituyen la solución son:

- 3.2" TFT LCD Screen Module
- LCD TFT01 Arduino Mega Shield
- Arduino Mega 2560

Así mismo se requiere de las siguientes librerías, estas librerías se pueden encontrar en la web de Henning Karlsen, después de descargarlas simplemente se descomprimen dentro de la carpeta *libraries* que está en la carpeta de instalación del IDE.

- UTFT
- tinyFAT
- UTFT_tinyFAT
- UTouch
- UTFT_Buttons

3.1.6.1. Materiales para su implementación.

Requerimos de la TFT, una tarjeta SD que soporte FAT16, *TFT Mega Shield* y de un Arduino Mega 2560. Más adelante se ofrece información acerca del conexionado y esquemático de cada módulo.

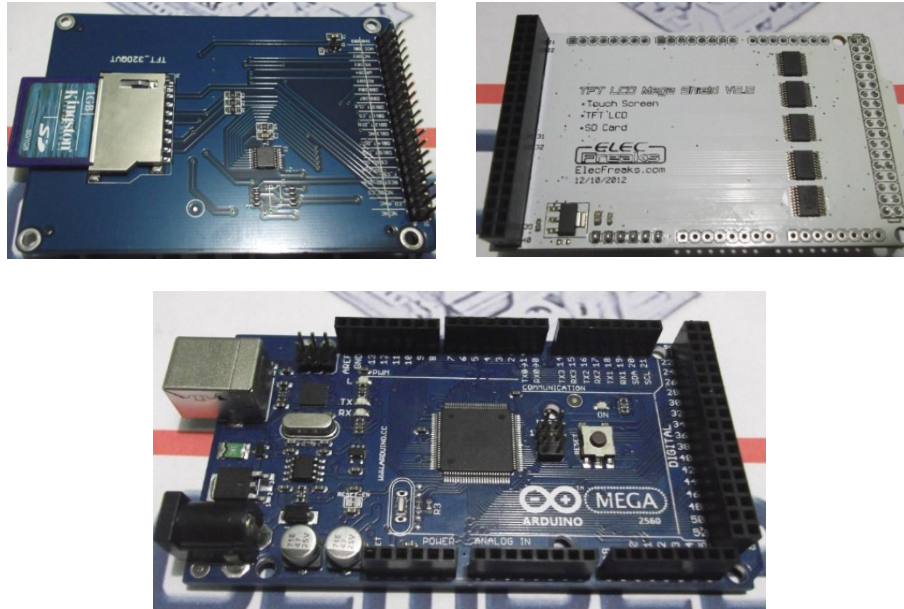


Figura 3.8 Módulos requeridos para la su implementación.

Así mismo requerimos del software **ImageConverter565.exe** también creado por Henning y que viene dentro de la carpeta **TOOLS** de la librería **UTFT**, este software se encarga de convertir las imágenes a formato **raw**¹³, que es el formato con que se puede trabajar este tipo de *displays*.



Figura 3.9 Software ImageConverter565.

¹³ **Formato RAW:** Es un formato de archivo digital de imágenes que contiene en su totalidad de los datos de la imagen tal y como ha sido captada por el sensor digital de la cámara fotográfica.

3.1.6.2. Pruebas y funcionamiento

Parte de los ejemplos demostrativos que vienen con las librerías requieren de leer imágenes desde la tarjeta SD. Como se menciona brevemente líneas arriba, la librería UTFT requiere que las memorias SD estén formateadas bajo FAT16¹⁴. Pero el formateo que proporciona el sistema operativo *Windows* no sirve ya que inserta mucha basura en las regiones de la tarjeta SD por lo cual la librería *tinyFAT* no puede levantar dicha tarjeta con tal formateo. Para tener un formateo limpio y sin vicios se debe utilizar el programa para formatear en FAT16, **SD Formatter 4.0 for SD/SDHC/SDXC**. Una vez que se tiene formateada la memoria se debe guardar en estas las imágenes con que se vaya a trabajar en el formato permitido (.raw). Luego de estos pasos se puede probar los ejemplos creados por el autor de la librería.

3.1.6.3. Descripción eléctrica del Módulo utilizado.

El módulo *TFT Mega Shield* tiene un regulador LDO de 3.3V, CE6209. El *jumper* J1 está para alimentar el *display* en caso de que el voltaje VCC de alimentación ya sea de 3.3V.

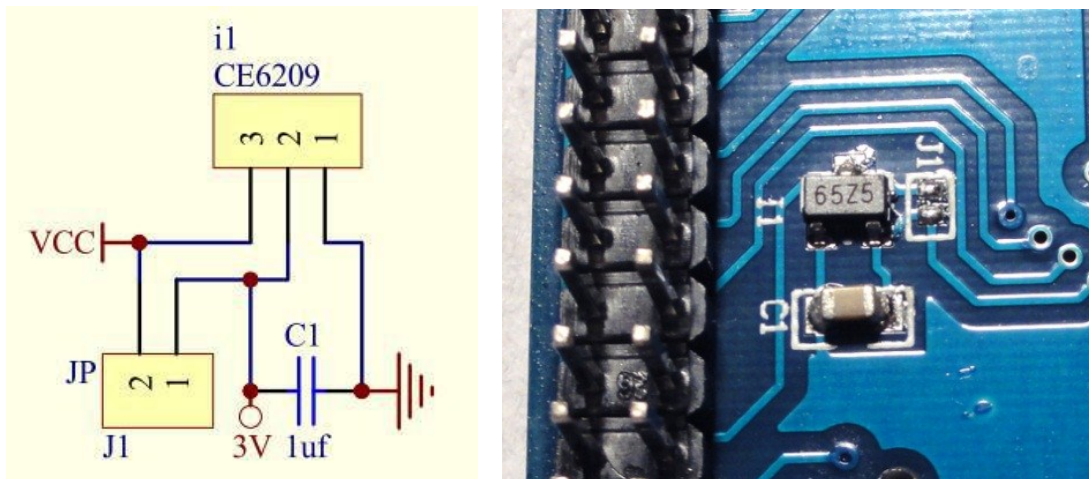


Figura 3.10 Regulador LDO de 3.3V del Shield TFT Mega.

El controlador de la pantalla táctil es el XPT2046 compatible con el TI ADS7843. La distribución de sus pines es como se muestra en la imagen 3.11 a continuación.

¹⁴ **FAT:** Tabla de Asignación de Archivos, es un sistema de archivos desarrollados para MS-DOS, así como el sistema de archivo principal de las ediciones no empresariales de Microsoft Windows.

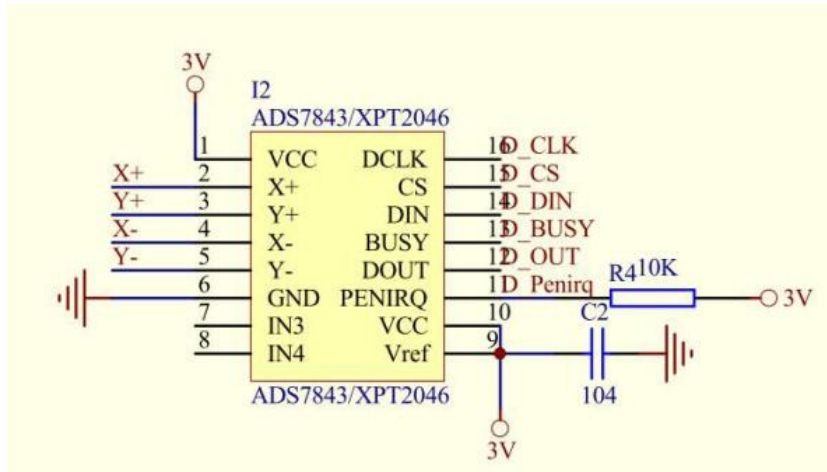


Figura 3.11 Controlados táctil XPT2046.

El modulo además cuenta con un conector para tarjetas SD o adaptadores SD-microSD. Las memorias SD trabajan con voltajes de 3.3V por lo que se necesita adecuar la señal de salida del microcontrolador en este caso el Arduino, que el voltaje de la salida en alto es de 5V, para esto se utiliza los CI buffer 74HC541PW que adecuan la señal a 3.3V y en los pines de entrada se ponen resistencias de 1KOhm para limitar la corriente que se le extrae al CI como se muestra en la figura 3.12.

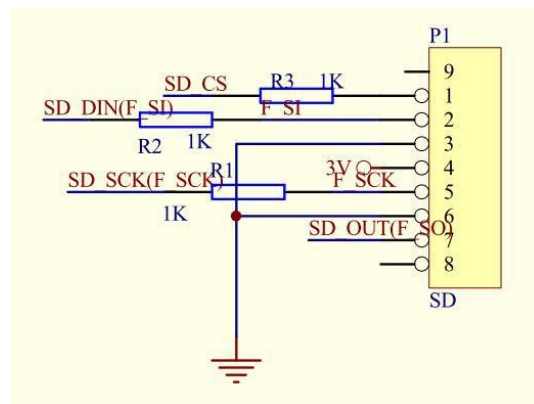


Figura 3.12 Configuración eléctrica para la lectura y escritura de la tarjeta SD/microSD.

Para controlar este módulo, Pantalla LCD, controlador Táctil y TFT¹⁵ Mega Shield, se hace necesario usar diferentes librerías para su configuración como se mencionaba anteriormente, algunas de las librerías utilizadas en este trabajo son las que menciono a continuación, las cuales me permiten el trabajo con este *display*.

Pantalla LCD.

La librería usada para trabajar con la pantalla LCD es la “*UTFT*” desarrollada por Henning Karlsen. Esta librería es la continuación de librerías *ITDB02_Graph*, *ITDB02_Graph16* y *RGB_GLCD* para Arduino y *chipkit*. Esta biblioteca soporta *displays* gráficos de 8 bit, 16 bit y LCD series.

Control del Touch Screen.

La librería usada es la “*UTFTTouch*” del mismo desarrollador de la anterior, Henning Karlsen. Es compatible con la configuración eléctrica del Shield del Arduino Mega mostrado en la figura 3.13 y otras configuraciones como la usada en este trabajo como es la que se muestra en la figura 3.14.

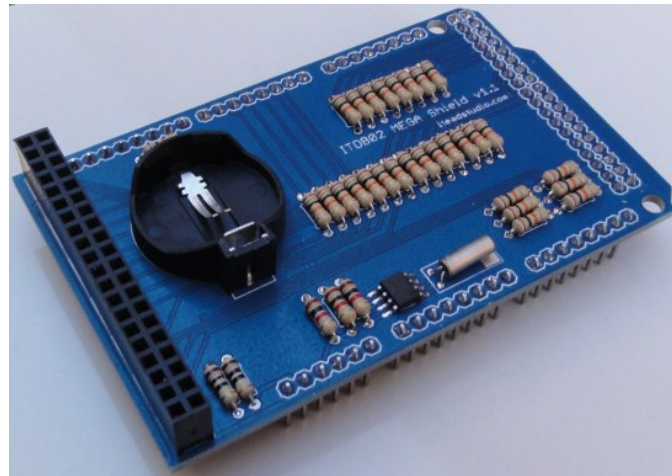


Figura 3.13 Shield TFT Mega v1.1.

¹⁵ **TFT:** Es un tipo especial de transistor de efecto de campo que se fabrica depositando finas películas de un semiconductor activo así como una capa de material dieléctrico y contactos metálicos sobre un sustrato de soporte.

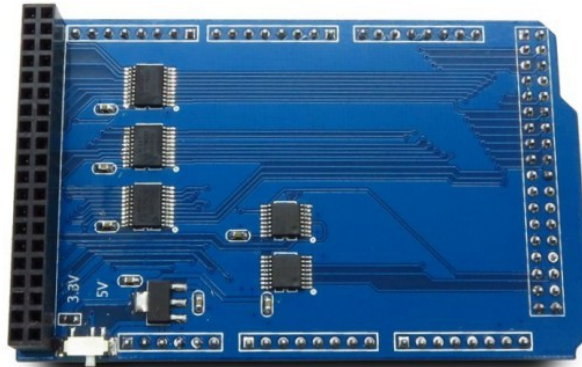


Figura 3.14 Shield TFT Mega v2.0.

SD Card

Henning Karlen también desarrollo una librería para la lectura y escritura de memorias SD o micro SD. Sin embargo esta librería está limitada a formatos FAT16 para memorias hasta 2GB de capacidad y nombres de archivos de hasta 8 caracteres. También está la librería SD de Arduino que soporta sistemas de archivos FAT16 y FAT32 para tarjetas estándar SD y tarjetas SDHC, esta también está limitada a 8 caracteres para los nombres de los archivos.

3.1.6.4. Pines de entrada y salida asignados al módulo LCD Touch.

Los pines usados por la pantalla TFT se muestran en la figura 3.15. Se incluyen la pantalla LCD, el controlador táctil, el lector de tarjetas SD y flash. La designación de las diferentes interfaces de los dispositivos son las siguientes.

- Dxx: Pantalla TFT
- SDxx: Lector tarjeta SD.
- D_xx: Controlador táctil.
- F_xx: Flash (no implementado en este módulo en particular, ni en la v1.1 ni en la v2.0, esta última es la usada en nuestro trabajo.)

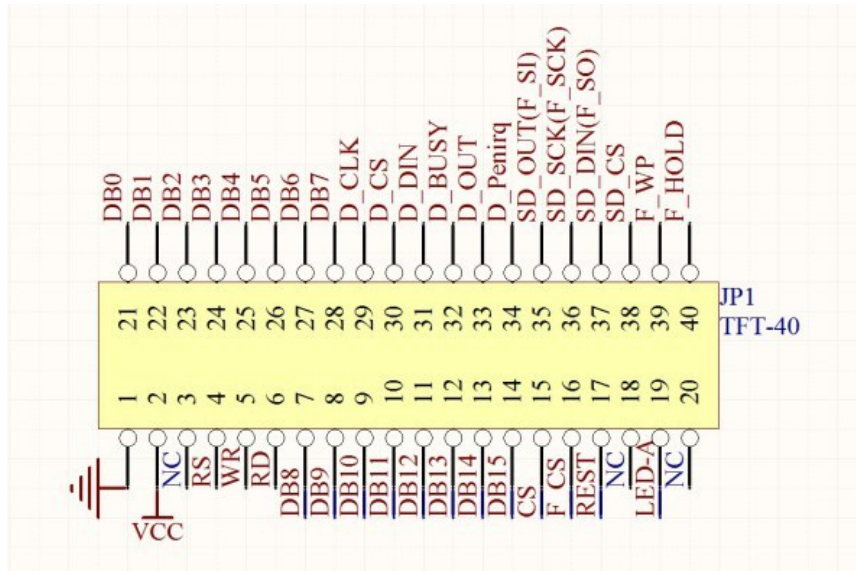


Figura 3.15 Distribución de pines IO de la pantalla TFT táctil.

Mapeo de los pines entrada salida del Shield TFT y el Arduino Mega.

El Shield TFT Mega v2.0 se utiliza para adecuar los valores de voltaje y corriente de salida del microcontrolador Arduino Mega 2560 a los parámetros adecuados para el trabajo con el lector de tarjeta SD y la pantalla LCD, en este caso trabajan con voltajes de alimentación son de 3.3 V, en el caso del controlador táctil trabaja con los mismos valores TTL (0 – 5V). Este módulo utiliza el CI buffer 74HC541PW para adecuar estos niveles de voltaje. La versión v1.1 utiliza los pines del Arduino Mega 2560 mostrados en la figura 3.16.

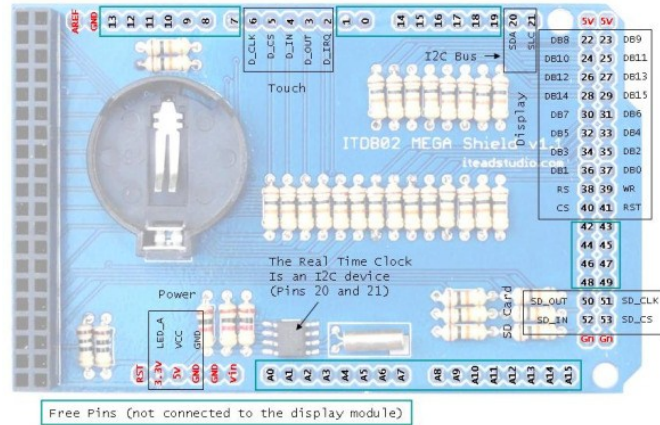


Figura 3.16 Distribución de pines de la v1.1 del Shield TFT Mega.

Para la versión 2.0 que se diferencia en que esta no incluye el reloj en tiempo real por tanto no utiliza el bus I2C por tanto la distribución de los pines es como se muestra en la figura 3.17. Por supuesto los pines que no se usan en el control de la pantalla están libres para otras funciones.

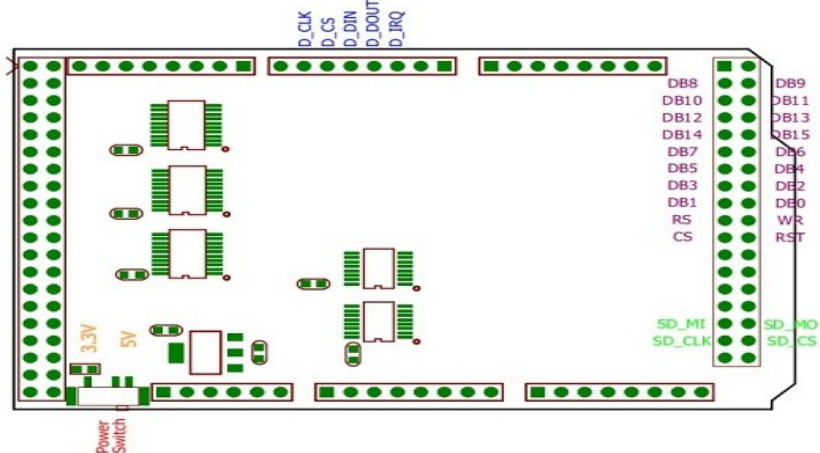


Figura 3.17 Distribución de pines de la v2.0 del Shield TFT Mega.

La v1.1, figura 3.16, se ha remplazado por este nuevo modelo, figura 3.17. No incluye el CI de reloj en tiempo real y remplace los resistores por un circuito lógico 74xx541 utilizado como buffer y convertor del nivel de voltaje, el cual tiene una mejor implementación. Además tiene un interruptor de selección de entrada de voltaje para hacerlo compatible tanto con Arduino (opera con 5V) como con Chipkit (opera con 3.3 V).

La conexión del Shield con el Arduino mega queda como se muestra en la figura 3.18.

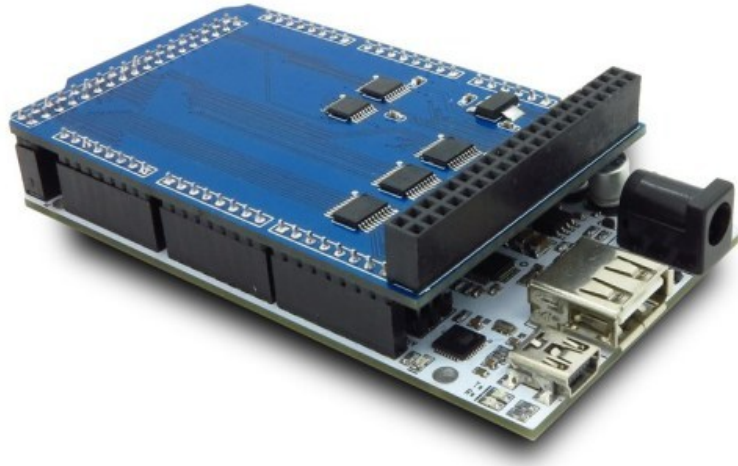


Figura 3.18 Conexión del Arduino mega con el Shield TFT Mega v2.0.

3.1.7. Codificador Rotatorio mecánico.

Un codificador rotatorio incremental también conocido como codificador en cuadratura, tiene dos salidas. Estas pueden ser mecánicas u ópticas, en este caso son mecánicas. Los codificadores con salidas mecánicas son típicamente usados como potenciómetros digitales en equipos de consumo digital. Los más recientes equipos de música de casa y de carros usan codificadores rotatorios para el volumen. Estos codificadores son los más ampliamente usados por el costo tan barato que tienen ya que solo requieren dos sensores. Se pueden encontrar también sensores con una tercera salida, los cuales se usan cuando hace falta tener una referencia inicial. En la figura 3.19 se muestra la foto de un codificador rotatorio.



Figura 3.19 Codificador Rotatorio Mecánico.

Las salidas A y B son llamadas salidas en cuadratura porque entre ellas hay 90° de fase como se observa en la figura 3.20

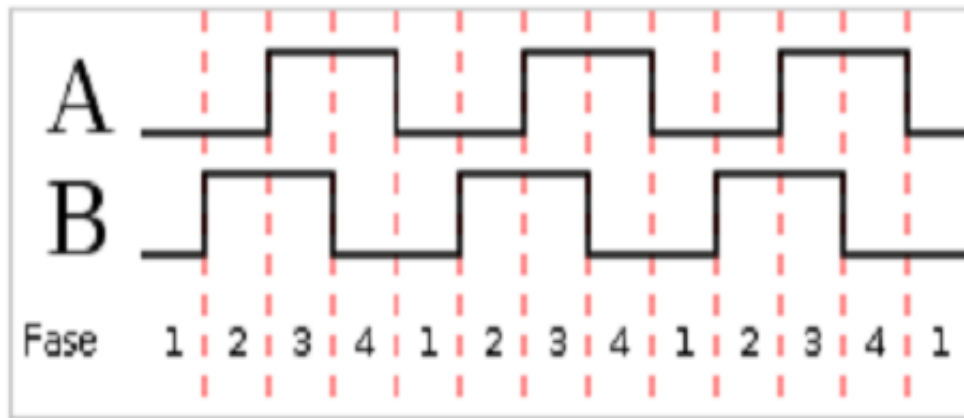


Figura 3.20 Salidas A y B en cuadratura.

La idea de usar los codificadores para múltiples aplicaciones como volumen digital y en el detector de movimiento de los ratones de computadoras es que poseen movimiento circular interminable y según el sentido de giro entrega una secuencia que cumple con el código de Gray¹⁶. La figura anterior muestra que la secuencia entregada por el codificador de izquierda a derecha es: 00, 01, 11, 10 y se repite. En el sentido inverso es la misma secuencia pero al revés.

Las señales deben ser decodificadas por software para determinar en qué sentido giró el codificador y así tomar una decisión. Para decodificar las señales por software se pueden usar interrupciones por flanco de caída o flanco de subida en ambos pines.

Los codificadores mecánicos por lo general tienen un tope que divide los 360° de giro en 24 pasos aproximadamente, a veces un poco más de pasos o incluso menos. El estado de los pines de salida para cada paso se alterna entre 00 y 11. Si el codificador tiene sus señales A y B en 00, se conoce que el próximo paso es 11, pero lo que determina el sentido del giro es el estado intermedio por el que pasa antes de llegar a 11, si pasa por 01 habrá girado en sentido horario, si lo hace por 10 habrá girado en el sentido contrario.

¹⁶ **Código Gray:** El código binario reflejado o código Gray, nombrado así en honor al investigador Frank Gray, es un sistema de numeración binario en el que dos valores sucesivos difieren solamente en uno de sus dígitos.

3.1.7.1. Conexión eléctrica usada para el codificador rotatorio.

La próxima figura muestra un esquema funcional del codificador mecánico y los componentes utilizados para su correcto funcionamiento.

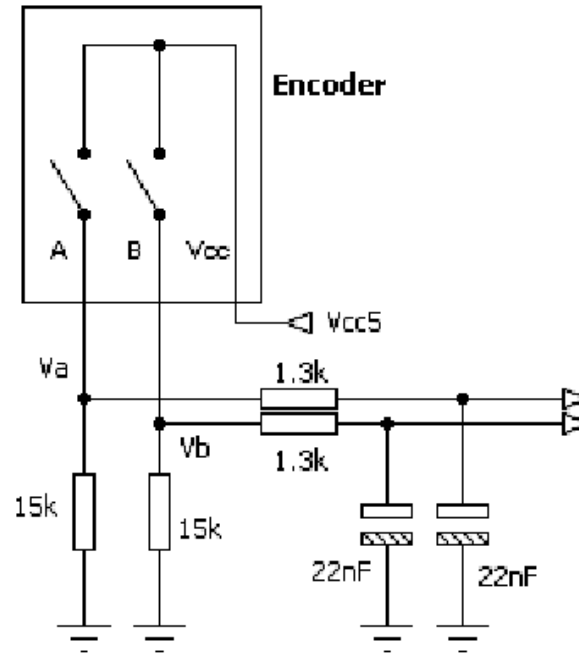


Figura 3.21 Conexión del codificador rotatorio.

Si el interruptor A está abierto, el nivel de tensión en V_a es igual a 0V. Si el interruptor está cerrado, el nivel de tensión en $V_a = 5V$. El objetivo de las resistencias de 15k es impedir que se cortocircuite la fuente. Los resistores de 1.3k y los capacitores de 22nF se utilizan para evitar el rebote, fenómeno que trae consecuencias desastrosas y mal funcionamiento del codificador.

Los capacitores absorben el rebote ya que se cargan durante un tiempo determinado por RC hasta el valor de la fuente. En el camino contrario, se descargan a través de las resistencias de 1.3 y 15 kOhm. La carga y descarga de los capacitores evita los rebotes porque la señal no puede variar bruscamente. Los valores de los componentes son valores prácticos utilizados en múltiples esquemas. La condición necesaria para calcularlos es que el tiempo RC debe ser mucho menor que el tiempo que demora un cambio de estado en el codificador.

El usuario puede hacer girar el codificador 300° como máximo en un intento, lo cual equivale a 20 pasos. Entre cada paso hay dos cambios de estado, por tanto el resultado de estados máximo es 40. El giro del codificador puede tardar como mínimo aproximadamente 1 segundo, entonces:

$$\text{Demora de 1 estado} = 1\text{seg} / 40 = 25 \text{ ms}$$

La demora RC debe ser mucho menor que el tiempo mínimo que demora un estado para que el circuito responda a los cambios del codificador.

$$\text{Constante de tiempo de carga} \quad RC = 1.3\text{k} * 22\text{nF} = 28.6\text{us}$$

$$\text{Constante de tiempo de descarga} \quad RC = (1.3\text{k} + 15\text{k}) * 22\text{nF} = 358\text{us}$$

3.1.8. Comunicación I2C de Arduino.

El acrónimo I2C o I^2C es un bus de comunicaciones serie, fue diseñado por *Philips* al inicio de la década de 1980. Su nombre viene de *Inter Integrated Circuit*. Este bus tiene una velocidad de transmisión de 100Kbits por segundo en el modo estándar, también permite velocidades de 3.4 Mbit/s. Es un bus usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas integrados y generalizando más para poder comunicar circuitos integrados entre sí, que normalmente residen en un mismo circuito impreso. ^[7]

La principal característica de I2C es que utiliza dos líneas para transmitir la información: una para los datos y por otra la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia. Como suelen comunicarse circuitos en una misma placa que comparten una misma tierra esta tercera línea no suele ser necesaria. En la figura 3.22 se muestra un esquema de conexión del bus I2C.

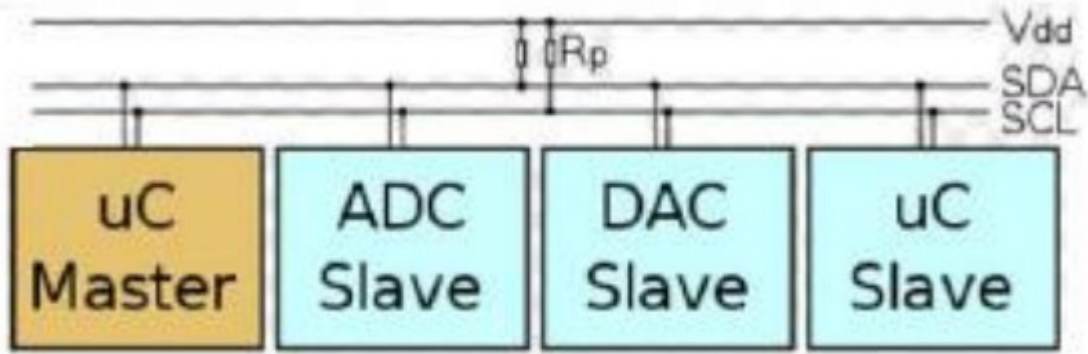


Figura 3.22 Bus I2C.

Las líneas SDA y SCL son de drenado abierto, por lo que necesitan resistencias de *pull-up*. Los dispositivos conectados al bus I2C tienen una dirección única para cada uno. También pueden ser maestros o esclavos. El dispositivo maestro inicia la transferencia de datos y además genera la señal de reloj. ^[7]

3.1.8.1. Protocolo del Bus I2C.

El bus I2C sólo define dos señales, además del común:

- **SDA:** Es la línea de datos serie (*Serial Data*, en inglés), semi-bidireccional. Eléctricamente se trata de una señal a colector o drenado abierto. Es gobernada por el emisor, sea este un maestro o un esclavo.
- **SCL:** Es la señal de sincronía (reloj serie, o *Serial Clock* en inglés). Eléctricamente se trata de una señal a colector o drenado abierto. En un esclavo se trata de una entrada, mientras que en un maestro es una salida. Un maestro, además de generar la señal de sincronía suele tener la capacidad de evaluar su estado; el motivo es poder implementar un mecanismo de adaptación de velocidades. Esta señal es gobernada única y exclusivamente por el maestro; un esclavo sólo puede retenerla para forzar al maestro a ralentizar su funcionamiento.

Esta particularidad física de que las salidas de los excitadores I2C hayan de ser a colector o drenador abierto no es casual sino que resulta de vital importancia para que a este bus con tan sólo una señal de datos y otra de sincronía se le pueda dotar de todas sus características funcionales. Un dispositivo lógico con este tipo de salida permite realizar la función producto lógico con una simple conexión eléctrica. ^[8] Evidentemente, un enlace I2C necesitará sendas resistencias de elevación en las respectivas líneas SDA

y SCL. De esta manera un excitador I2C realmente sólo gobierna el estado 0 lógico en las líneas I2C, mientras que el estado 1 lógico no es suministrado por el excitador directamente sino por medio de la oportuna resistencia de elevación. Así, el concepto de aislamiento del bus se consigue con tal sólo hacer que el excitador del nodo que se desea aislar ofrezca a su salida el estado alto (1 lógico), al que le corresponde una elevadísima impedancia de salida, equivalente a un aislamiento eléctrico con respecto al bus. ^[9]

3.1.8.2. Comunicación I2c entre Arduino Nano y Arduino Mega.

La tarjeta Arduino Nano le envía a través del bus I2C la temperatura y el voltaje de la línea registrados. En caso de que se configurara algunos parámetros a través de la red entonces también se los envía para ser visualizados en el *display* LCD. Por su parte el Arduino Mega le envía al Nano los datos de configuración que se modificaron.

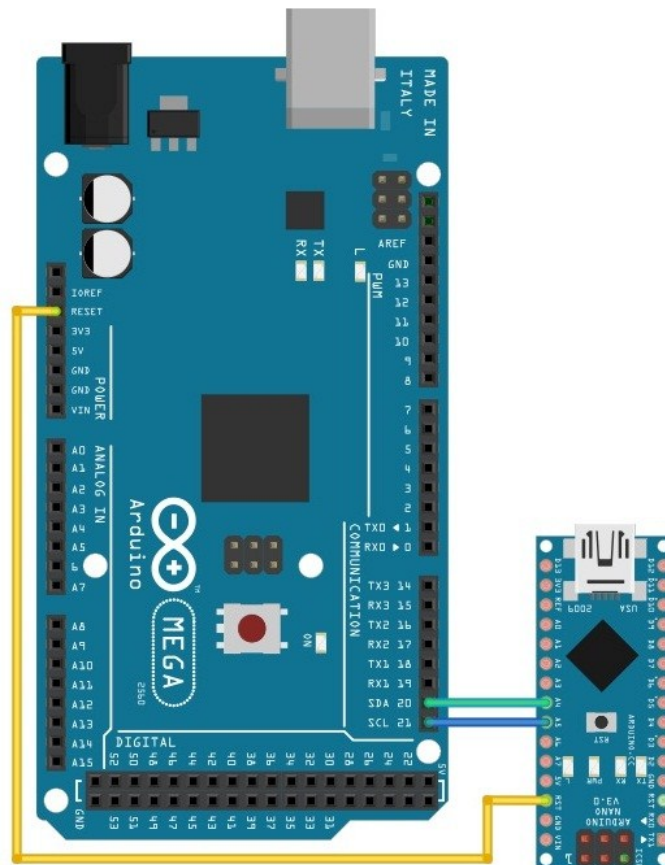


Figura 3.23 Comunicación I2C entre los Arduinos utilizados.

Los pines correspondientes al bus I2C del Arduino Mega son los pines 20 y 21, SDA y SCL respectivamente y estos se conectan directamente con los pines homólogos en la tarjeta Arduino Nano, en esta tarjeta corresponden a las entradas A4 y A5, SDA y SCL respectivamente. En la figura 3.23 se muestra la conexión entre el Arduino Mega y el Nano.

3.1.9. Circuito de Fuerza.

Cada pin del microcontrolador ATMEGA 328, en el cual se basa la plataforma Arduino Nano, solo permite una corriente de salida máxima de 40 mA, pero lo ideal es evitar esta situación y extraer la menor corriente posible, así el microcontrolador estará protegido contra corrientes excesivas y por tanto contra sobrecalentamiento. Por eso se calculan las resistencias de polarización de los transistores de manera tal que cumpla con lo antes mencionado.

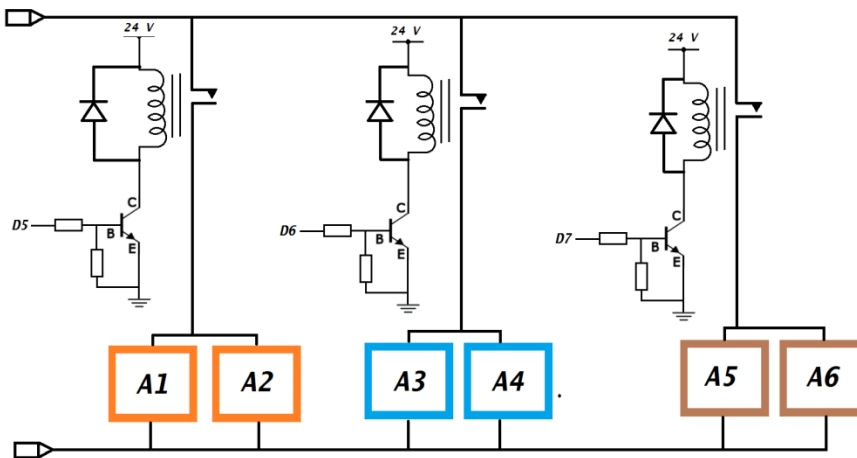


Figura 3.24 Circuito de fuerza.

El control de este hardware se implementó con lógica directa: un 0 a la salida del microcontrolador desactiva el control mientras que un 1 lo activa. Si el control está activados pone la salida en alto este satura el transistor NPN se activa el relé y se entrega corriente a la carga. Si el control está desactivado se realiza el proceso contrario, se pone la salida en nivel bajo, se corta el transistor NPN y se desactiva el relé. En la figura 3.24 se muestra el circuito de fuerza que permite conectar y desconectar los aires acondicionados.

3.1.9.1. Control de los Aires Acondicionados.

El circuito diseñado realiza el control automáticamente, se programó un mecanismo de rotación del funcionamiento de estos aires acondicionados con el objetivo de disminuir el tiempo de explotación diaria. El administrador del sistema establece un tiempo de rotación durante el cual estará una salida activada poniendo en funcionamiento dos aires acondicionado para mantener cierta temperatura en el local, otra salida de reserva lo que quiere decir que solo se entregará corriente a la carga, se encienden los aires de reserva, si la temperatura excede el límite establecido y por último la salida que queda estará desactivada.

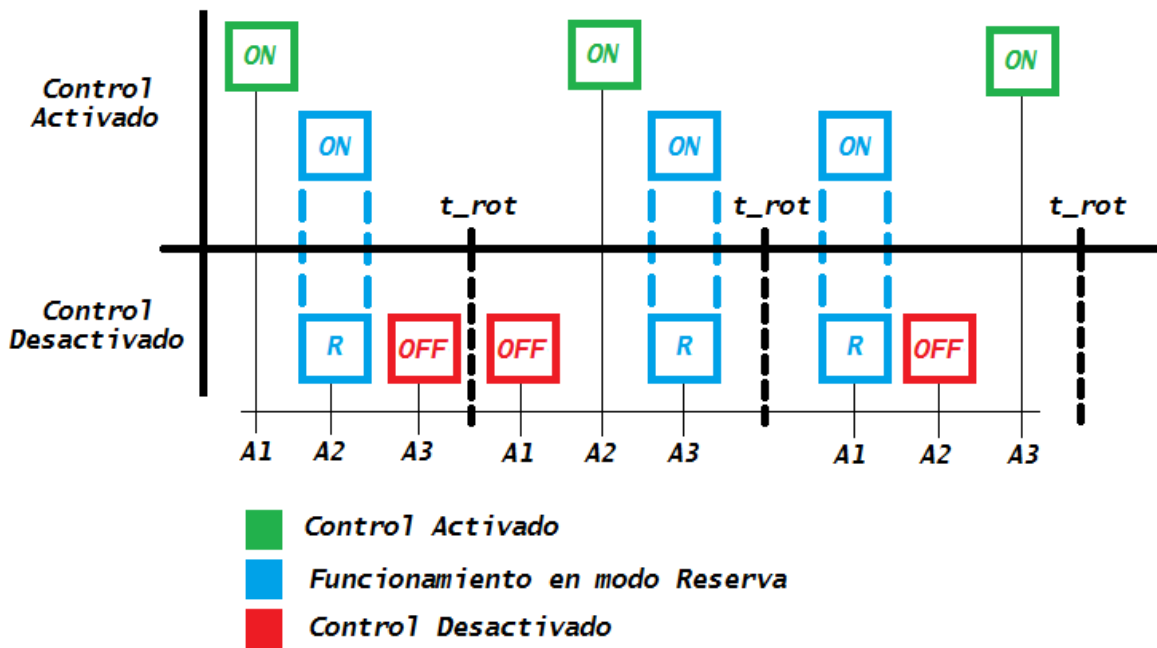


Figura 3.25 Esquema de control de los aires acondicionados.

En la figura 3.25 se puede observar el funcionamiento anteriormente explicado, por ejemplo si el tiempo de rotación es 8 horas, cada salida estará activada solo 8 horas diaria siempre y cuando la temperatura del local no sobrepase su límite superior que entonces la salida que esté de reserva trabajará un tiempo de más mientras la temperatura no vuelva a estar por debajo del límite inferior. Esto es un ciclo, cuando termina se repite de manera circular como se muestra en la figura 3.26.

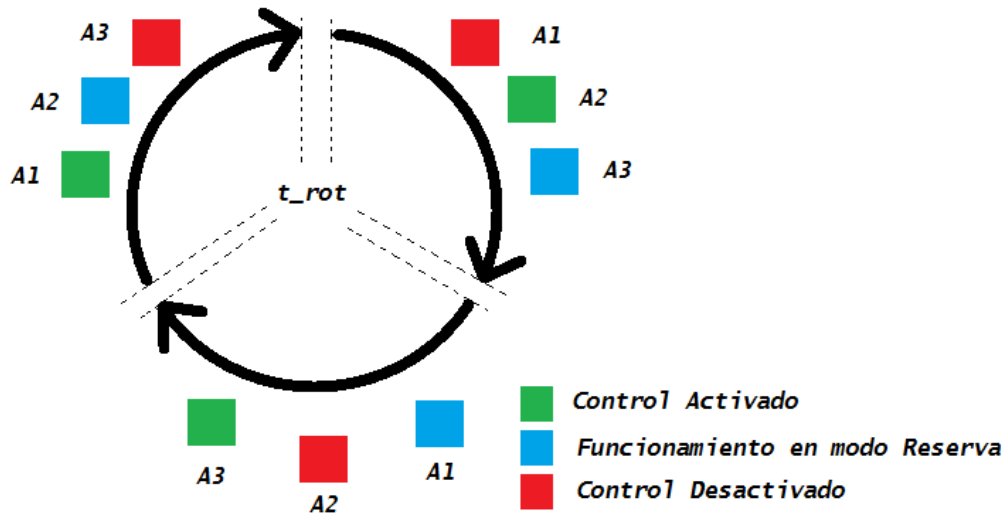


Figura 3.26 Ciclo de funcionamiento para el ejemplo de 8h de tiempo de rotación.

3.1.10. Ethernet.

De todos los esquemas utilizados para organizar redes de área local, Ethernet es la que tiene mayor aceptación. Aunque es uno de los esquemas más antiguos, ha mejorado con el tiempo y su tasa de transmisión aumentó de su valor original de 10 Mb/s a 100 Mb/s y en la actualidad, hasta 1 Gb/s. ^[10]

Xerox dio origen a Ethernet, con la participación posterior de Intel y *Digital Equipment Corporation*. Alrededor de 1980, cuando el IEEE empezó a redactar normas para redes de área local, Ethernet ya era de facto una norma. Por consiguiente, aunque se dice algunas veces que Ethernet sigue la norma 802.3 del IEEE, ¡es más exacto decir que la norma IEEE sigue a Ethernet! En la actualidad, la norma Ethernet está apoyada por muchos vendedores. (Figura 3.27)

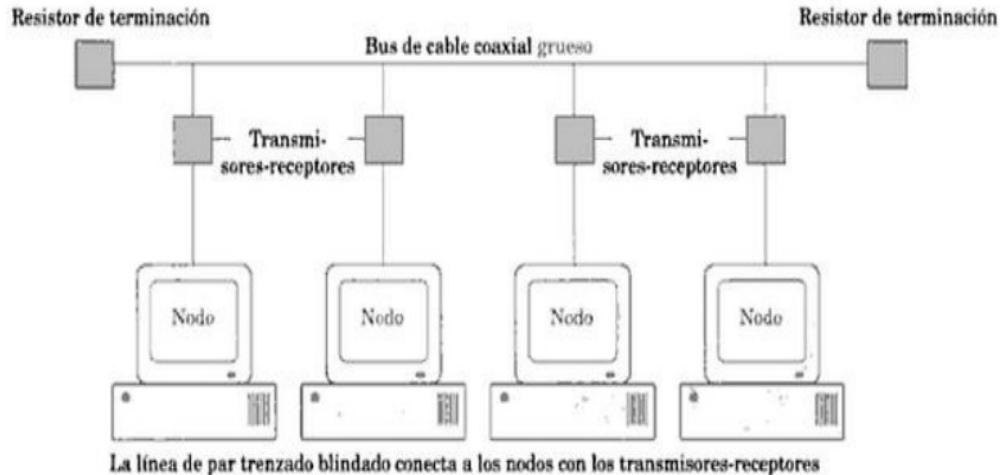


Figura 3.27 Red Ethernet Gruesa.

El cable Ethernet grueso se reserva en la actualidad para los tendidos de cable de estructura principal (*backbone*) que conecta a grupos (conglomerados o *clusters*) de computadora. Las redes pequeñas y grupos de nodos en las redes más grandes utilizan "Ethernet delgada" (*Thin Ethernet*, en términos coloquiales conocida como "*cheapernet*"), en la que se usa el cable coaxial común, barato, RG -58/U. en las normas IEEE, a esta instalación se la llama 10Base2 (10 Mb/s, banda base, longitud de 200 metros). Aunque la norma especifica una longitud máxima de 200 metros, en la práctica la longitud puede ampliarse hasta casi 300 sin que surjan problemas. El cable se conecta a las computadoras mediante conectores en T BNC ordinarios como se ilustra en la figura 3.28.

La Ethernet también se emplea con la línea de par trenzado, como la que se utiliza para la telefonía de voz. Este sistema de 10BaseT (T para trenzado) se restringe a una longitud de 100 m, con dos pares de alambres. Aunque lógicamente se trata de un sistema de bus, la conexión física es en estrella con un concentrador de cableado central. El concentrador desconecta un nodo defectuoso o uno con cableado en corto circuito.

También hay versiones del estándar para tasas de transmisión de datos más altas (100MB/s) en cable de par trenzado y cable de fibra óptica. Debido a la dificultad de construir conectores en T de fibra óptica, en las redes de fibra siempre se utiliza una configuración física en estrella. La red Ethernet puede operar en distancias más grandes con amplificación adicional (hasta una distancia de 2.8 Km en ciertas condiciones).

La razón por la limitación en la longitud total de una LAN Ethernet se encuentra en el protocolo CSMA/CD utilizado con la Ethernet.

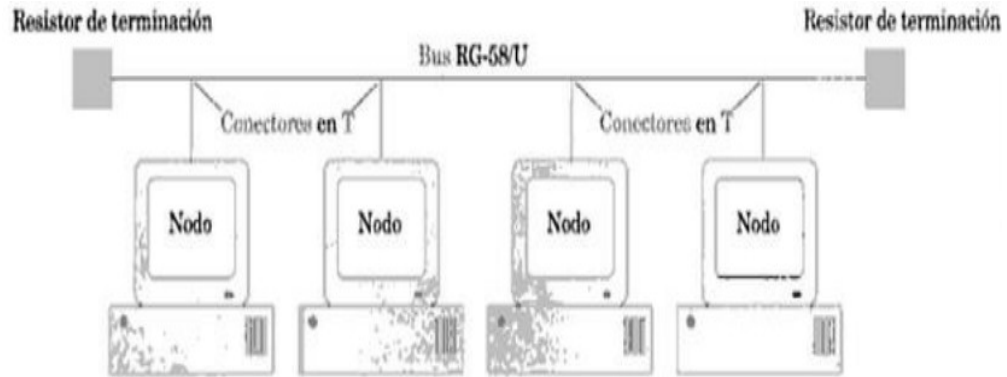


Figura 3.28 Red Ethernet delgada.

3.1.10.1. Servidor Web con el ENC28J60.

El módulo de Ethernet ENC28J60 permite conectar el Arduino a Internet. Está basado en el chip que lleva su mismo nombre ENC28J60 de la compañía Microchip. Tiene un conector Ethernet RJ-45 con una velocidad de hasta 10 Mb pero tiene compatibilidad con 10/100/1000Base-T. Se usa la librería *EtherCard* para escribir el programa.

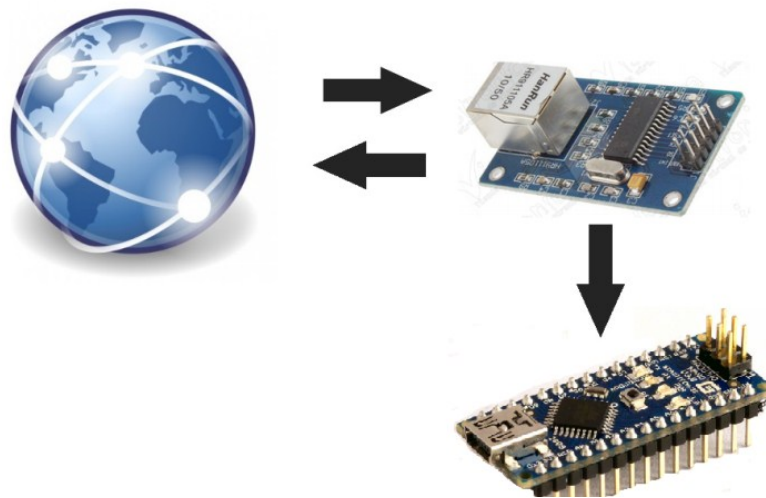


Figura 3.29 Servidor Web con Arduino.

Para comunicarse con la placa Arduino, utiliza el puerto SPI (pins 10, 11 ,12 y 13) con la librería *SPI*. Estos pines no pueden ser usados como entradas o salidas digitales. Gracias a la este módulo de Ethernet podemos usar el Arduino como servidor web o como cliente web.

El módulo de Ethernet ENC28J60 permite conectar el Arduino a Internet. Está basado en el chip que lleva su mismo nombre ENC28J60 de la compañía Microchip. Tiene un conector Ethernet RJ-45 con una velocidad de hasta 10 Mb pero tiene compatibilidad con 10/100/1000Base-T. Se usa la librería *EtherCard* para escribir el programa. Para comunicarse con la placa Arduino, utiliza el puerto SPI (pines 10, 11 ,12 y 13) con la librería *SPI*. Estos pines no pueden ser usados como entradas o salidas digitales. Gracias a la este módulo de Ethernet podemos usar el Arduino como servidor web o como cliente web.

Arduino como servidor Web

Antes que nada hay que saber que es un servidor web. Un servidor web es una computadora que guarda y transmite datos vía Internet. La información que almacenan es en forma de página Web. Cuando un cliente (navegador Web) accede a un servidor Web, envía una petición HTTP que recibe el servidor Web. Luego este envía la información a través de protocolo HTTP al cliente en formato HTML.

La petición HTTP que envía el cliente al servidor sería algo parecido a lo siguiente:

GET / HTTP/1.1

Host: 192.168.1.100 (dirección IP del servidor)

Accept: text/html

User-Agent: Mozilla/5.0 (Windows) (Navegador y sistema operativo)

[Línea en blanco]

Y la respuesta del servidor al cliente sería similar a lo siguiente:

```
HTTP/1.1 200 OK
Date: Sun, 25 Des 2014 17:42:10 GMT
Content-Type: text/html
<html>
  <!-- código HTML -->
</html>
```

La tarjeta Arduino Nano está conectada a la red con una IP que le asignamos. El código HTML de la web que enviará lo escribiremos en el sketch. Cuando ponemos esta dirección IP en un buscador Web de un ordenador conectado a la misma red (o desde fuera a través de Internet si abrimos un puerto que nos dirija a la IP del Arduino), estamos enviando una petición HTTP. Cuando el Arduino lea esta petición, nos enviará el código HTML para que podamos visualizar la página Web en el navegador.

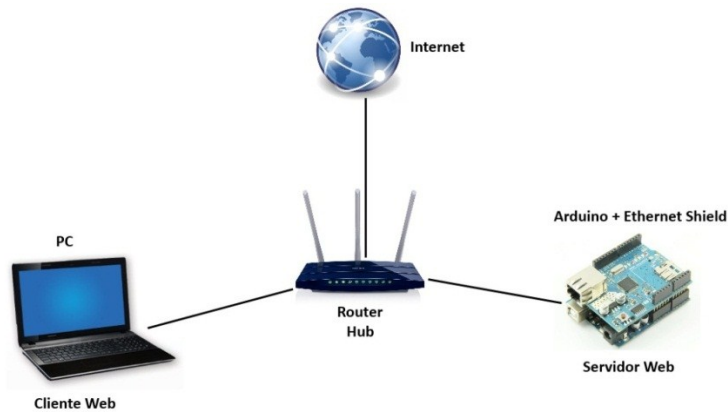


Figura 3.30 Conexión de la tarjeta Arduino a la red.

Tenemos que asignarle una IP al Arduino, esta tiene que estar dentro del mismo rango que la IP de la puerta de enlace (*Gateway*). Para saber esto, en Windows nos vamos a la consola (poniendo *cmd* en el buscador de *Inicio*), escribimos "*ipconfig*". Aquí nos pondrá la IP asignada del ordenador, la máscara subred y la puerta de enlace. (Figura 3.31).

```
Dirección IPv4. . . . . : 192.168.1.96
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

Figura 3.31 Obtener la IP de la puerta de enlace.

En este ejemplo la puerta de enlace es 192.168.1.1, por lo tanto asignaré al Arduino la IP 192.168.1.100. Para comprobar si esta IP no está asignada a ningún otro equipo, haremos un PING a esta dirección como se muestra en la figura 3.32. Si al hacer PING no obtenemos respuesta, quiere decir que la dirección IP está libre.

```
C:\Users\Adrià>ping 192.168.1.100

Haciendo ping a 192.168.1.100 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.1.100:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
      (100% perdidos),
```

Figura 3.32 Comprobación de la existencia de la IP escogida.

3.1.10.2. Configuración del ENC28J60.

Este módulo de Ethernet es una simple solución para dotar a la plataforma Arduino de conexión a una red. Esta tarjeta puede trabajar con distintos modelos de tarjetas Arduino, UNO, MEGA, NANO, etc. Funciona con un oscilador de cristal de 25 MHz.

En la tabla 3.1 se muestra como conectar una tarjeta Arduino al controlador Ethernet ENC28J60. En esta tabla se describe cuáles son los pines de la tarjeta Arduino a los que pueden ser conectados los pines del módulo ENC28J60.

Tabla 3.1 Distribución de pines del ENC28J60 y conexión con Arduino.

Symbol	Function		UNO, Nano Duemilanove	MEGA
INT	Interrupt	↔	2 or NC	2 or NC
SO	SPI MISO Serial data out	↔	12	50
SCK	SPI SCK Serial clock in	↔	13	52
RESET	Reset	↔	Reset	Reset
GND	Ground	↔	GND	GND
CLKOUT	Clock output	↔	NC	NC
WOL		↔	NC	NC
SI	SPI MOSI Serial data in	↔	11	51
CS	SPI SS Chip select	↔	10	53
VCC	Power supply 3.3V	↔	3V3	3V3

En la figura 3.33 se muestra el esquema eléctrico del módulo controlador de Ethernet ENC28J60 comúnmente usado.

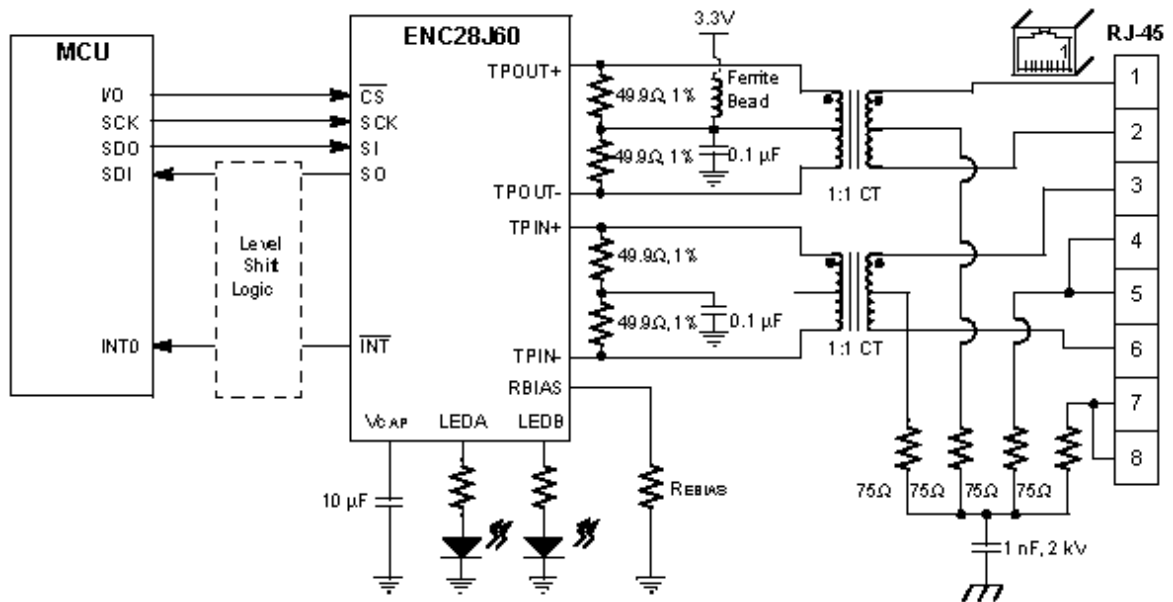


Figura 3.33 Esquema eléctrico más común para usar el controlador ENC28J60.

3.2. Diseño del software.

3.2.1. Entorno de desarrollo para Arduino.

El entorno de desarrollo para Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús como se indica en la figura 3.34. Este software permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.



Figura 3.34 Entorno de desarrollo para Arduino.

Arduino utiliza para escribir la programación lo que denomina "sketch" (programa), los programas son escritos en el editor de texto. En el entorno de desarrollo de Arduino existe la posibilidad de cortar/pegar y buscar/reemplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra los errores.^[11]

La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie:

3.2.2. Librerías.

Las librerías proveen funcionalidad extra a nuestro sketch, por ejemplo: para trabajar con hardware o para manipular datos. Para usar una librería dentro de un sketch, se puede seleccionar desde *Sketch > Import Library*. Si se desea usar una librería que no vienen junto con Arduino, es necesario instalarla. Para hacerlo, hay que descargar la librería y descomprimirla. ^[12]Debe ubicarse a la librería en una carpeta propia, y normalmente, contiene dos archivos, uno con sufijo “.h” y otro con sufijo “.cpp”. Al abrir la carpeta del software de Arduino, si ya existe una carpeta llamada “*libraries*”, se debe colocar la carpeta de la librería ahí dentro. A continuación se tiene que reiniciar el IDE de Arduino para encontrar la nueva librería en el menú *Sketch > Import Library*.

Las librerías usadas en este trabajo son:

1. **DallasTemperature:** Esta librería se encarga de obtener los valores digitales de temperaturas enviados por el sensor DS1820. Establece la comunicación 1-Wire entre la tarjeta Arduino y el sensor de temperatura.
2. **EtherCard:** Junto con el controlador Ethernet ENC28J60 esta librería permite a la tarjeta Arduino Nano conectarse a internet. Puede funcionar tanto como servidor capaz de aceptar conexiones entrantes o como cliente permitiendo realizar conexiones de salida. La librería permite hasta cuatro conexiones simultáneas (conexiones entrantes, salientes, o la combinación de ambas).
3. **SdFat:** Es una librería que soporta los sistemas de archivos FAT16 y FAT32 en tarjetas de memoria SD estándar y tarjetas de memoria de alta capacidad SDHC. SdFat sólo admite nombres de archivos cortos. Las clases principales utilizadas en la librería SdFat son: Sd2Card, SdVolume y SdFile.
4. **Wire:** Las comunicaciones a través del protocolo I2C o 1-Wire se realizan mediante la utilización de esta librería la cual es compatible con ambos protocolos.
5. **SPI:** Esta librería permite la comunicación con dispositivos SPI con la tarjeta Arduino, siendo esta el dispositivo maestro.
6. **UTFT:** Esta librería fue originalmente la continuación de las librerías ITDB02_Graph, ITDB02_Graph16 y RGB_GLCD para Arduino y *Chipkit*. Permite el trabajo con los *displays* LCD gráficos, soporta LCD gráficos de 8 bit, de 16 bit y

con comunicación serie y puede trabajar tanto con tarjetas Arduino como con *Chipkit*.

7. **UTouch:** Esta librería es un complemento a la librería UTFT que permite trabajar con pantallas táctiles.
8. **UTFT_Buttond:** Es una adición a la librería UTFT pero no tienen que trabajar simultáneamente. Esta librería requiere del uso de otra librería, *UTouch*. Es una manera más fácil de agregar botones en el display gráfico y extender el uso de las librerías *UTFT* y *UTouch*.

3.2.3. Registrador de variables eléctricas con monitoreo a través de un navegador web.

Como su nombre lo dice esta aplicación permite monitorear en tiempo real los datos registrados por el sistema a través de cualquier navegador web, ya que la tarjeta Arduino NANO y la módulo de Ethernet ENC28J60 están programadas para funcionar como un servidor web que puede soportar hasta cuatro conexiones simultáneas. Para que las tarjetas permitan el acceso a su información mediante un navegador web es necesario habilitar el puerto 80 (HTTP); y, además hay que asignar una dirección IP y una dirección MAC a las tarjetas Arduino UNO y Arduino Ethernet Shield, las mismas que son:

Dirección MAC: 0xDE.0xAD.0xBE.0xEF.0xFE.0xED

Dirección IP: 166.254.128.100 (esta dirección es la usada en las pruebas realizadas).

La página Web se encuentra en el servidor creado por el Arduino Nano y el controlador Ethernet ENC28J60 en la dirección IP mencionada anteriormente. Cuando ponemos esta dirección IP en un buscador Web de un ordenador conectado a la misma red (o desde fuera a través de Internet si abrimos un puerto que nos dirija a la IP del Arduino), estamos enviando una petición HTTP. Cuando el Arduino lea esta petición, nos enviará el código HTML para que podamos visualizar la página Web en el navegador. La página principal muestra los valores de temperatura y voltajes registrados, figura 3.35, y permite a través de un link configurar el servidor web, esta página se muestra en la figura 3.36.

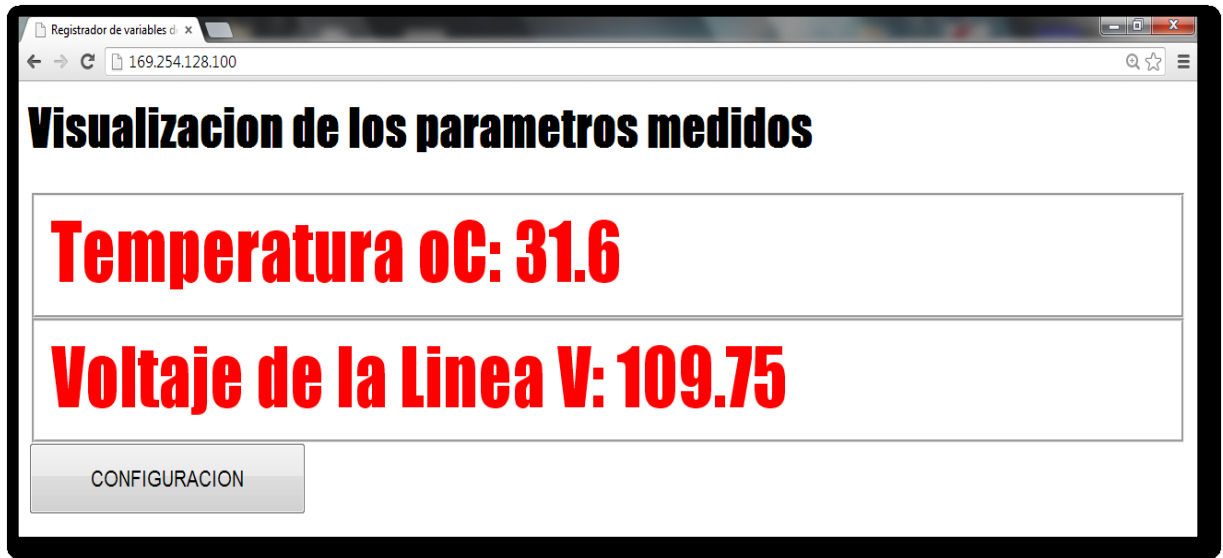


Figura 3.35 Visualización de los datos en la web.

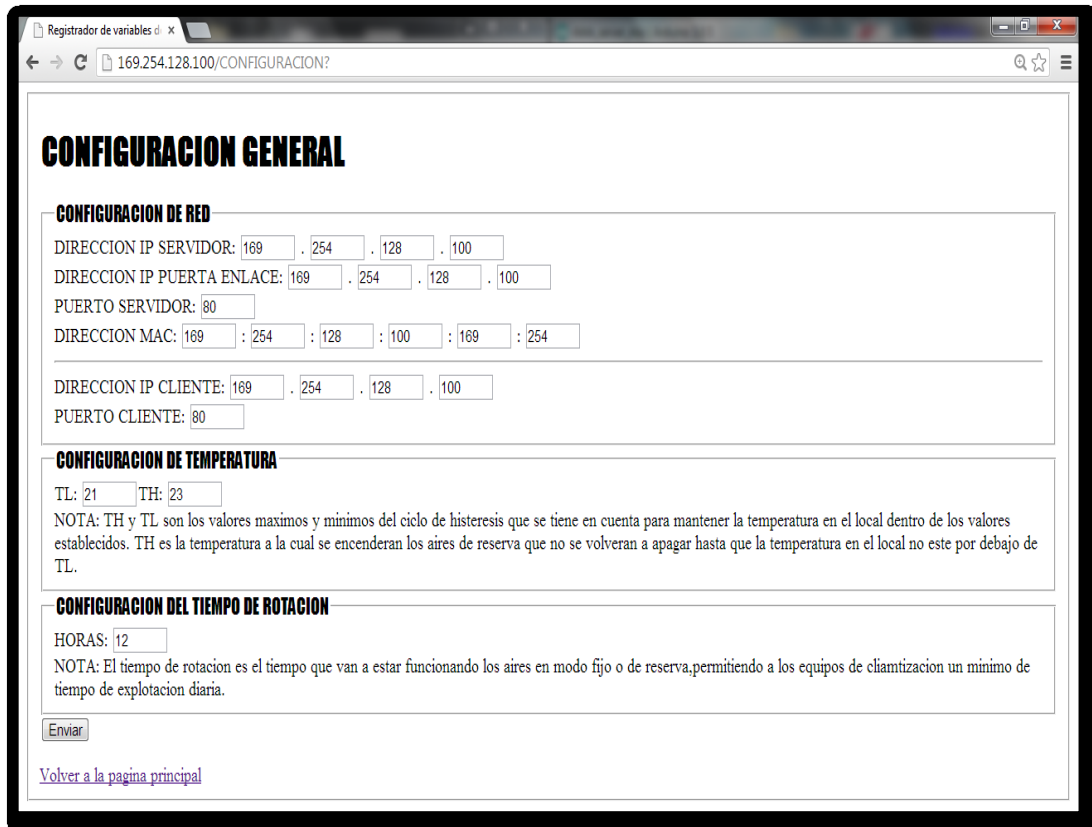


Figura 3.36 Página de configuración.

3.2.4. Programación de la tarjeta Arduino Nano.

El programa que se diseñó en la tarjeta Arduino Nano realiza las siguientes funciones:

- Toma los valores de la temperatura enviados por el sensor DS18B20 a través del bus I2C.
- Acondiciona digitalmente la señal del sensor de voltaje.
- Envía los datos registrados a través del bus I2C a la tarjeta Arduino Mega para que sean presentados en el *display* LCD.
- Crea el archivo de almacenamiento “SSC-C00.csv” cada vez que se reinicia el sistema, el mismo se enumera de 0 a 99 automáticamente.
- Almacena la información procesada del sensor de temperatura y el sensor de voltaje junto a la fecha y hora en una tarjeta de memoria micro SD cada 2 segundos en el archivo de almacenamiento “SSC-Cxx.csv”.

3.2.5. Diagrama de flujo del software del Arduino Nano.

En la figura 3.37 se muestra el diagrama de flujo del algoritmo de programación que se utilizó en el software del Arduino nano. El diagrama de flujo de la interrupción interna del *timer1* del Arduino Nano se evidencia en la figura 3.38.

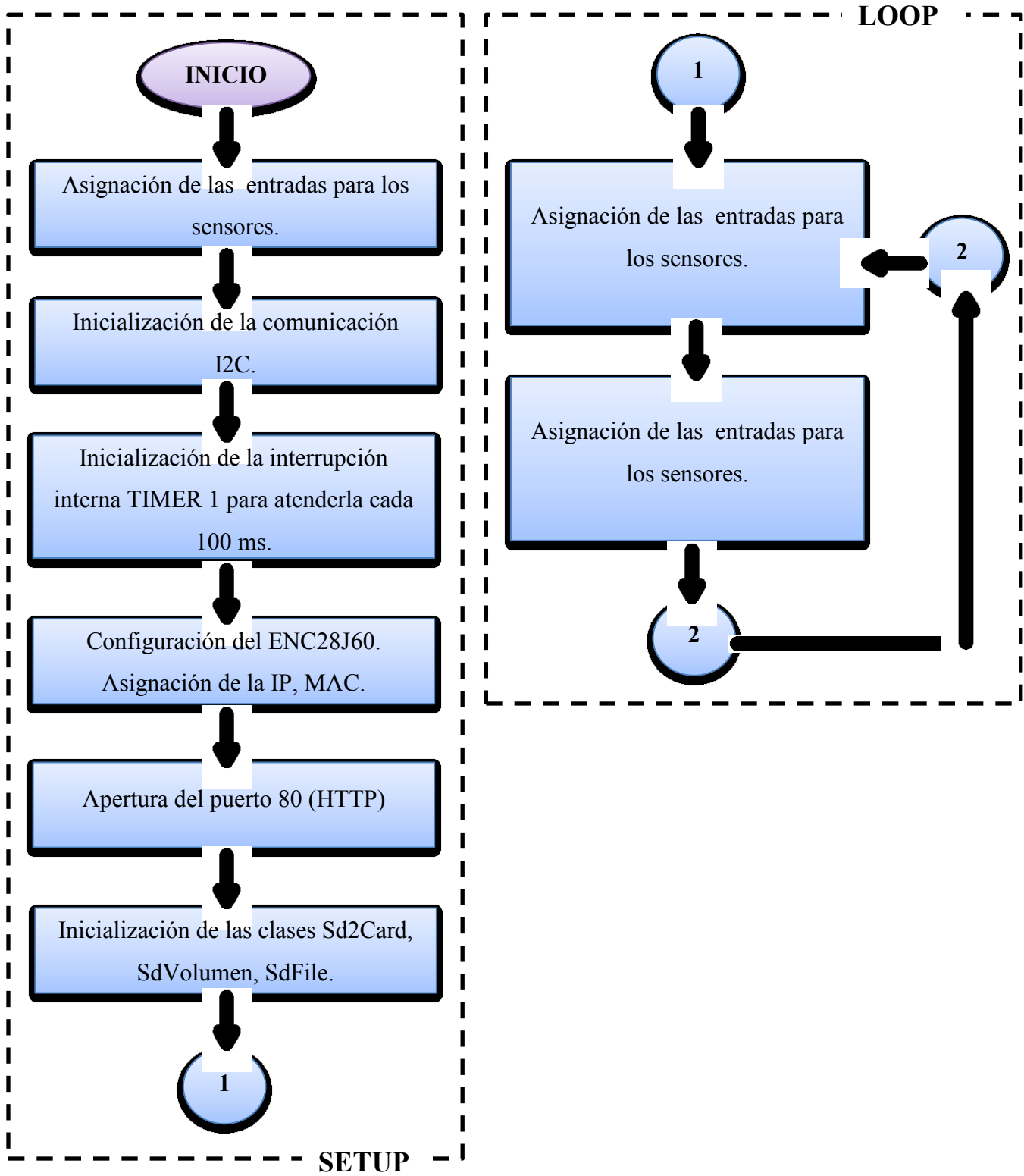


Figura 3.37 Diagrama de flujo del sistema de supervisión y control.

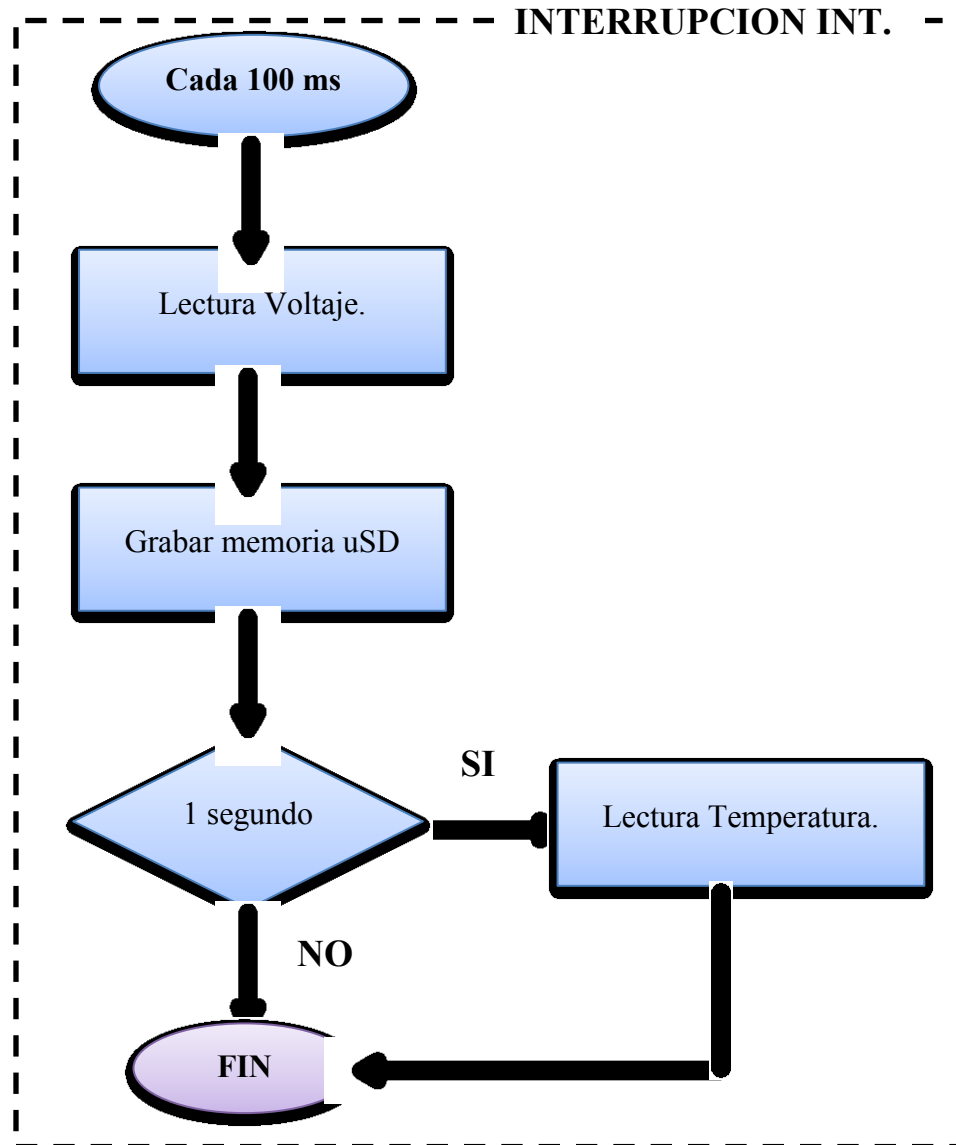


Figura 3.38 Diagrama de flujo de la interrupción interna del Timer 1.

3.3. Manual de Usuario del dispositivo.

A continuación se describe cómo funciona y como se configura el dispositivo supervisor y controlador de las temperatura y el voltaje de alimentación de los aires que mantienen el clima del CPD del Nodo Central de la UO.

Inicialmente el dispositivo actualiza los valores que va a mostrar y los visualiza en la pantalla principal del sistema como se muestra en la figura 3.39. Esta pantalla incluye además de los valores de temperatura y voltaje, la capacidad máxima de la tarjeta micro SD que se está utilizando y la espacio ocupada en la misma, he indica la fecha y hora actual y el estado de conexión del sistema con la red. Presenta 6 botones que nos ofrecen diferentes funcionalidades como son:

- RED: Este botón visualiza los datos concernientes a la conexión Ethernet.
- AIR: Permite activar o desactivar las salidas que apagan o encienden respectivamente los aires acondicionados del local.
- ESTAD: Muestra las estadísticas más relevantes como son la temperaturas mínimas, máximas y los voltajes limites registrados junto a la fecha y hora en que se tomaron dichos valores.
- AYUDA: Como su nombre lo indica ofrece una breve ayuda para trabajar con el dispositivo.
- CONFIG: Permite la configuración del dispositivo de control y supervisión.
- IV: **No se usa.**



Figura 3.39 Menú principal.

El botón de RED inicialmente estará desactivado mientras no exista conectividad con la red, una vez que se logre la conexión se indicará en el *display* y se activará su funcionalidad, como se mencionaba anteriormente visualiza los datos de red como la dirección IP del dispositivo, la dirección MAC, la dirección IP de un servidor al que quiera conectarse y el puerto de comunicación por el cual lo realizará. (Figura 3.40)



Figura 3.40 Menú de RED.

Si se selecciona la tecla AIR se visualizará un menú como se muestra en la figura 3.41 donde se crean y visualizan 5 botones adicionales que permiten activar o desactivar las salidas que encienden o apagan los aires acondicionados.



Figura 3.41 Menú para encender o apagar los aires acondicionados.

En la interface gráfica del menú de configuración, figura 3.42, se muestran todas las opciones que pueden ser modificadas con sus valores actuales, tiene un botón por cada uno de estos parámetros. Se permiten modificar la dirección IP del servidor, dirección IP del cliente, la dirección MAC del dispositivo (figura 3.43) y el puerto de comunicación (figura 3.44). Además se pueden variar los valores de temperatura tanto para el límite superior como el inferior, este último nunca superará el valor máximo. Y por último se puede establecer un valor de tiempo de rotación como se observa en la figura 3.45.



Figura 3.42 Menú de Configuración.

Cada menú de configuración tiene 4 botones que permiten la variación de los parámetros, estos son los botones *UP* para incrementar el valor, *DOWN* para decrementarlo, *LEFT* y *RIGHT* para moverse por los otros valores, esto se puede observar en las figuras de a continuación.



Figura 3.43 Configuración de la dirección IP y la dirección MAC.



Figura 3.44 Configuración del puerto de comunicación.

En el caso de los valores de temperatura y el tiempo de rotación como se mencionaba anteriormente tienen limitaciones, en el caso de la temperatura la diferencia entre ambos

extremos nunca será mayor de 5 °C ni menor de 1 °C. El tiempo máximo permitido para la rotación del funcionamiento de los aires acondicionados es de 12 horas.

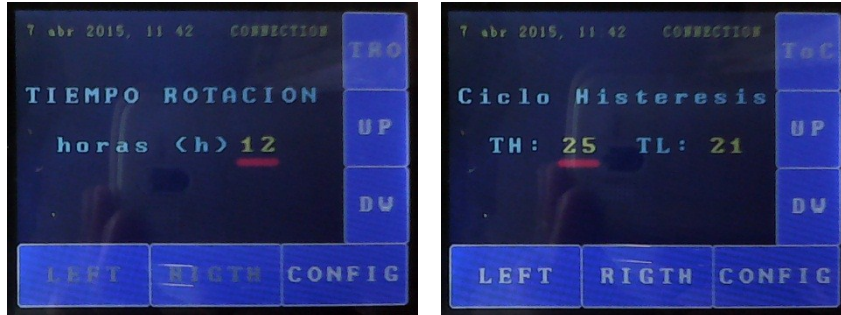


Figura 3.45 Menú de configuración de la temperatura y el tiempo de rotación.

CONCLUSIONES

A partir del estudio detallado de las particularidades y especificaciones de las tarjetas Arduino Nano, Mega 2560 y el controlador de Ethernet ENC28J60 se realizó el diseño e implementación de un sistema de supervisión y control capaz de proteger los equipos de cómputo y climatización del CPD del Nodo Central de la Universidad de Oriente.

El sistema permite comunicarse con la red de la Universidad proporcionando la visualización de los datos registrados y la configuración del sistema desde cualquier punto de la red.

El almacenamiento de valores registrados por el sistema permite realizar comparaciones numéricas entre datos antiguos y datos actuales.

El costo de los sistemas supervisores y controladores de variables tanto eléctricas como ambientales es alto y posee software y hardware propietarios como se mencionó en capítulos anteriores, por lo que el dispositivo implementado es una propuesta de bajo costo y su interfaz de funcionamiento es sencilla debido al uso de la plataforma Arduino ya que presenta software y hardware de arquitectura abierta.

RECOMENDACIONES

Como recomendaciones se pueden señalar las siguientes:

1. Implementar sistemas de alarmas al dispositivo de control y supervisión utilizando el protocolo de gestión de red SNMP o SMTP para una mayor eficiencia.
2. Seguir desarrollando y mejorando la propuesta realizada midiendo otros parámetros de interés como es la humedad relativa del CPD y el consumo eléctrico de los equipos de climatización.

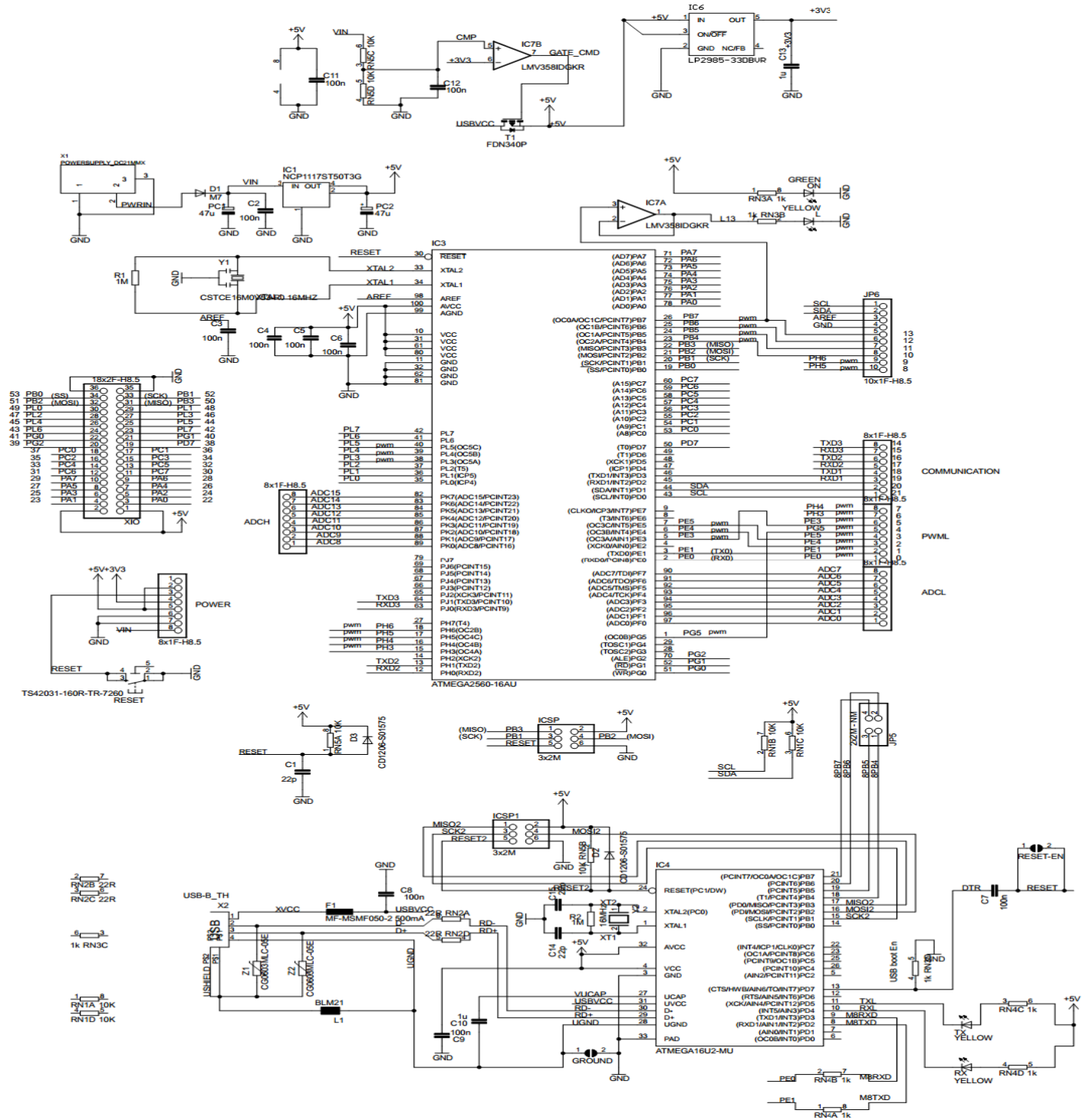
REFERENCIAS BIBLIOGRÁFICAS

- [1] EDMINISTER, JOSEPH E. "Circuitos Eléctricos" Pág. 2
- [2] NILSSON, JAMES W.; RIEDEL, SUSAN A. "Circuitos Eléctricos" Pág. 14
- [3] FRIAS GUSTAVO, JESUS L., Diseño e implementación de un software y hardware de un registrador de variables eléctricas con comunicación Ethernet basado en la tecnología Arduino y sistema de supervisión HMI., Latacunga, Ecuador, enero del 2013.
- [4] Oscar Torrente Artero. Arduino. Curso práctico de formación. Editorial Alfaomega. Primera Edición. Páginas 588. ISBN 978-607-707-648-3
- [5] ARDUINO. "Arduino Nano". [Documento HTML]. 6 de agosto del 2012 <http://arduino.cc/en/Main/ArduinoBoardNano>
- [6] Pallas Aren, Ramón: "Adquisición y Distribución de Señales". Editorial Marcombo. Lovera.
- [7] Depto. Técnico de Roso-Control, Ibor Sistema de desarrollo para Microcontroladores PIC ATOM, El bus I2C y la iBoard. Parte I: Memorias 24X de Microchip. Caso específico 24LC256, 2006, September.
- [8] Irazabal, Jean-Marc, Blozis, Steve, I2C Manual, Application note I2C Bus, Philips Semiconductors, TecForum at DesignCon, San Jose, CA, USA, 2003, January.
- [9] Prophet, Graham, PULSE, I2C bus runs at 1Mhz, EDN Europe, 2006, July, p. 30
- [10] ARDUINO. "Arduino Ethernet Shield". [Documento HTML]. 7 de agosto del 2012. <http://arduino.cc/en/Main/ArduinoEthernetShield>
- [11] C. C. Juan José, M.H. Rigoberto Alfonso, V. N. Gustavo Enrique, Monitor visual del consumo de energía eléctrica en viviendas, Escuela de Ingeniería Eléctrica y Electrónica, ISBN: 978-99923-982-8-9, Santa Tecla, marzo de 2010
- [12] Hugh B. Jonathan Oxer, Practical Arduino: Cool Projects for Open Source Hardware, ISBN-13 (electronic): 978-1-4302-2478-5, Editorial Board.

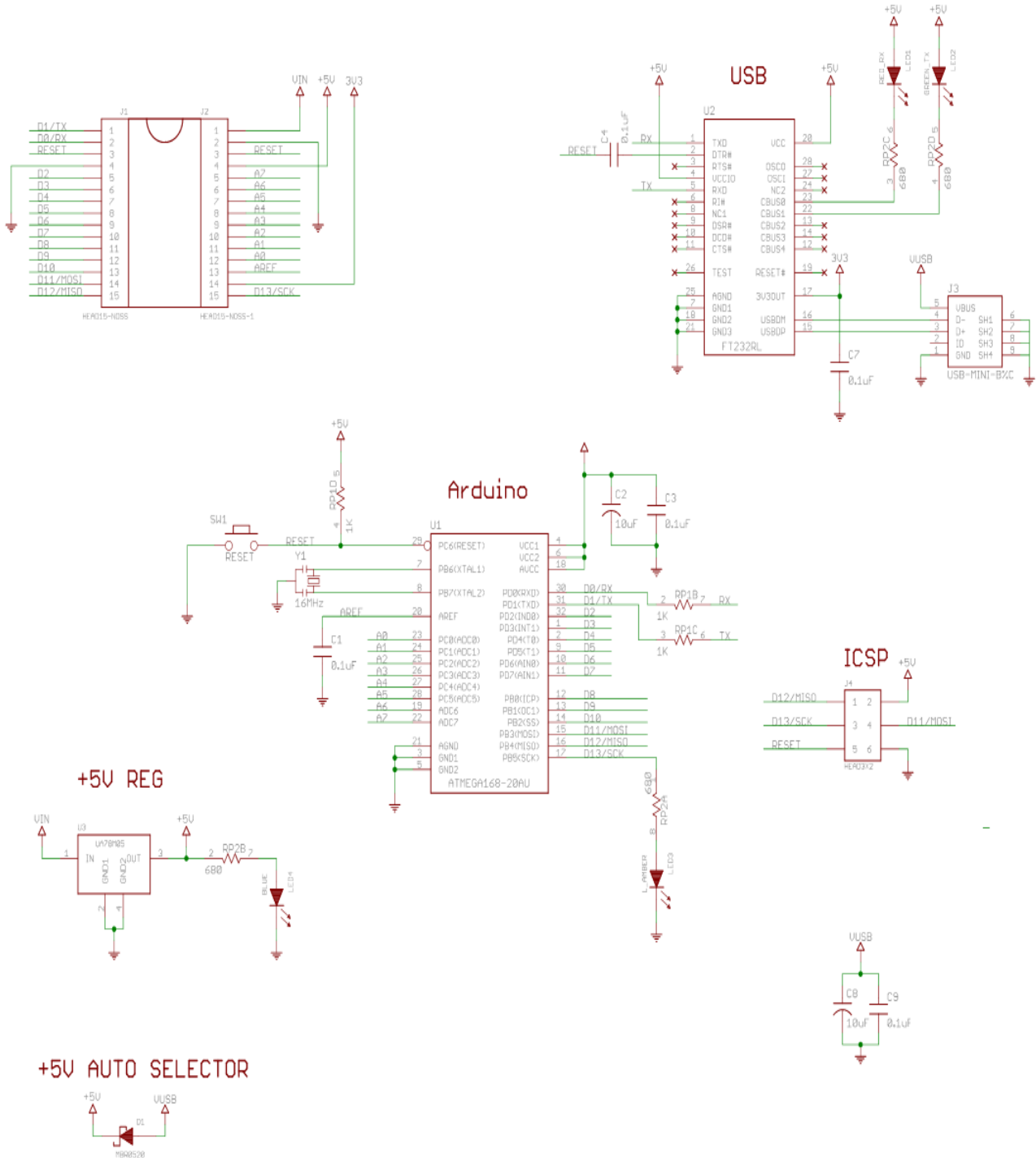
GLOSARIO DE TÉRMINOS

- CPD – Centro de Procesamiento de Datos.
- I2C – Circuitos Inter-Integrados
- LAN – Red de área local.
- LCD – *Liquid Crystal Display*.
- LED – Diodo emisor de luz.
- PCB – Placa de circuito impreso.
- PWM – Modulación de Ancho de Pulso
- RTD – Detector de Temperatura Resistivo.
- SAI – Sistema de Alimentación Ininterrumpida.
- SCL – *Serial Clock*.
- SDA – *Serial Data* .
- SNMP – Protocolo Simple de Administración de Red.
- SPI – *Serial Peripheral Interface*
- SRAM – Memoria estática de acceso aleatorio
- UO – Universidad de Oriente.
- USB – Conductor Universal en Serie.

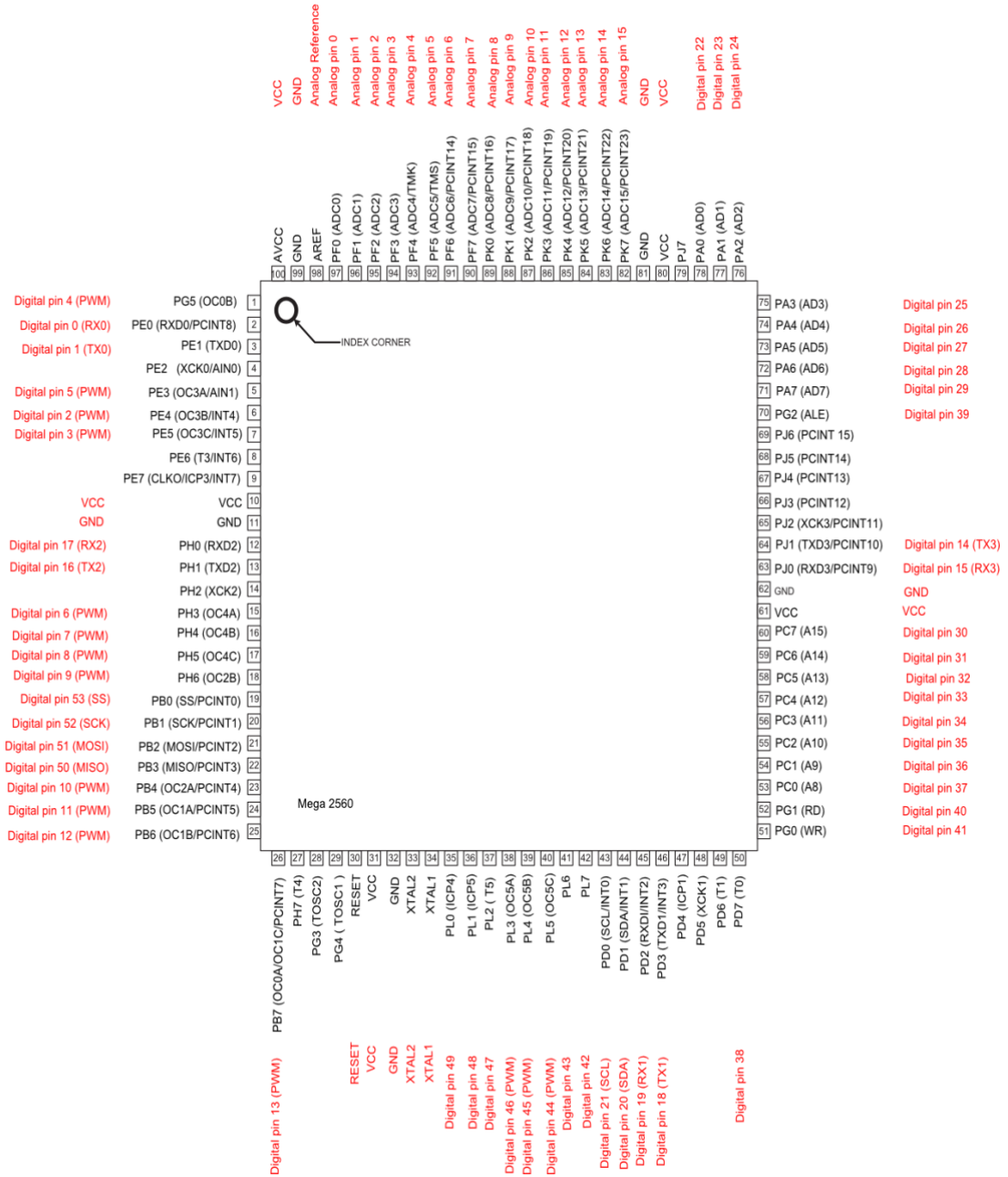
ANEXOS I. ESQUEMA ELÉCTRICO DEL ARDUINO MEGA 2560.



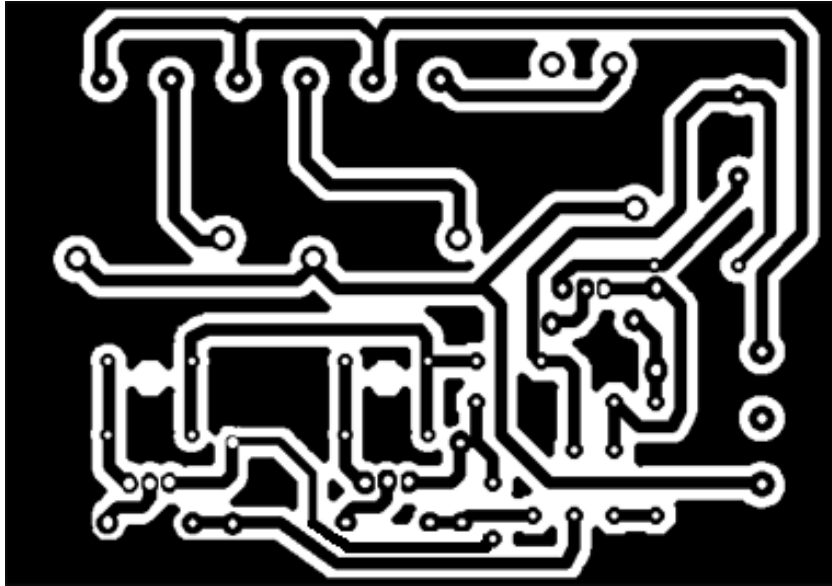
ANEXOS II. ESQUEMA ELÉCTRICO DEL ARDUINO NANO.



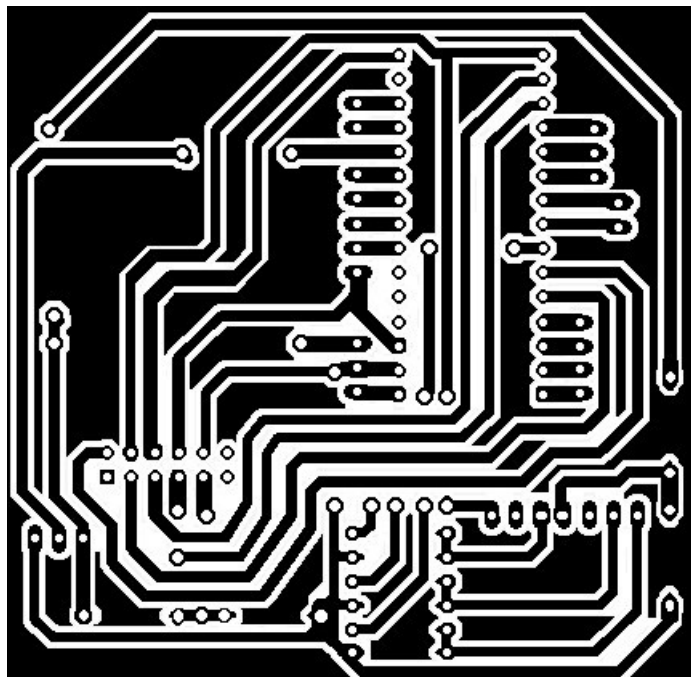
ANEXOS III. MAPEO DE PINES DEL ARDUINO MEGA 2560.



ANEXOS IV. CIRCUITO IMPRESO.



PCB del Circuito de Fuerza,



PCB del Módulo Central.

ANEXOS V. CÓDIGO HTML DE LA PÁGINA PRINCIPAL.

```
<html>
<head>
  <title>Registrador de variables de la Universidad de Oriente</title>
</head>
<body>
  <fieldset>
    <legend><FONT FACE="impact"><h1>Visualización de los parámetros medidos</h1></FONT></legend>
    <fieldset>
      <FONT FACE="impact" SIZE=10 COLOR="red">Temperatura oC: </FONT>
    </fieldset>
    <div><h1> </h1></div>
    <fieldset>
      <FONT FACE="impact" SIZE=10 COLOR="red">Voltaje de la Línea V: </FONT>
    </fieldset>
    <div><h1> </h1></div>
    <input type="submit" value = "CONFIGURACION" onClick=location.href=./?PAGE_AIRES\/>
  </fieldset>
</body>
</html>
```

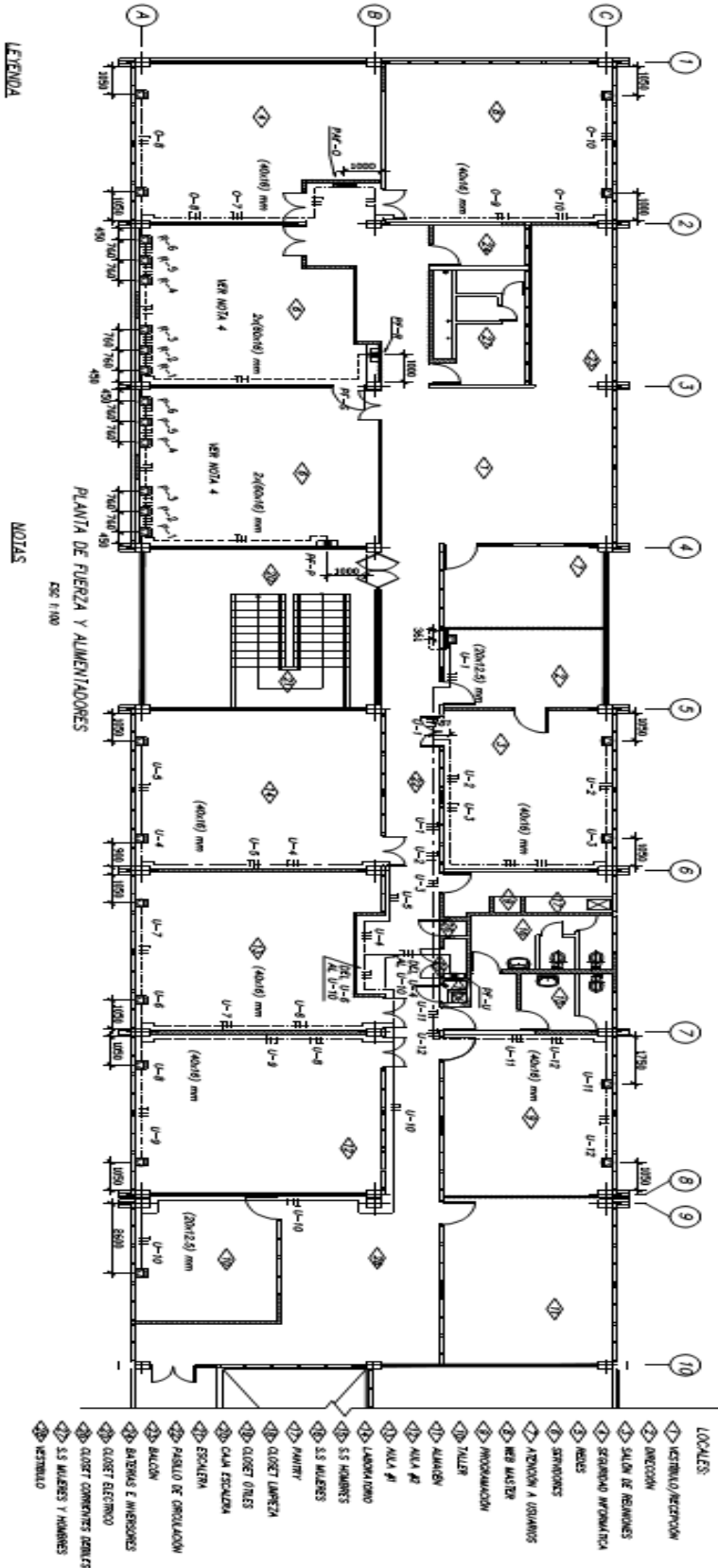
ANEXOS VI. CÓDIGO HTML DE LA PÁGINA DE CONFIGURACIÓN.

```

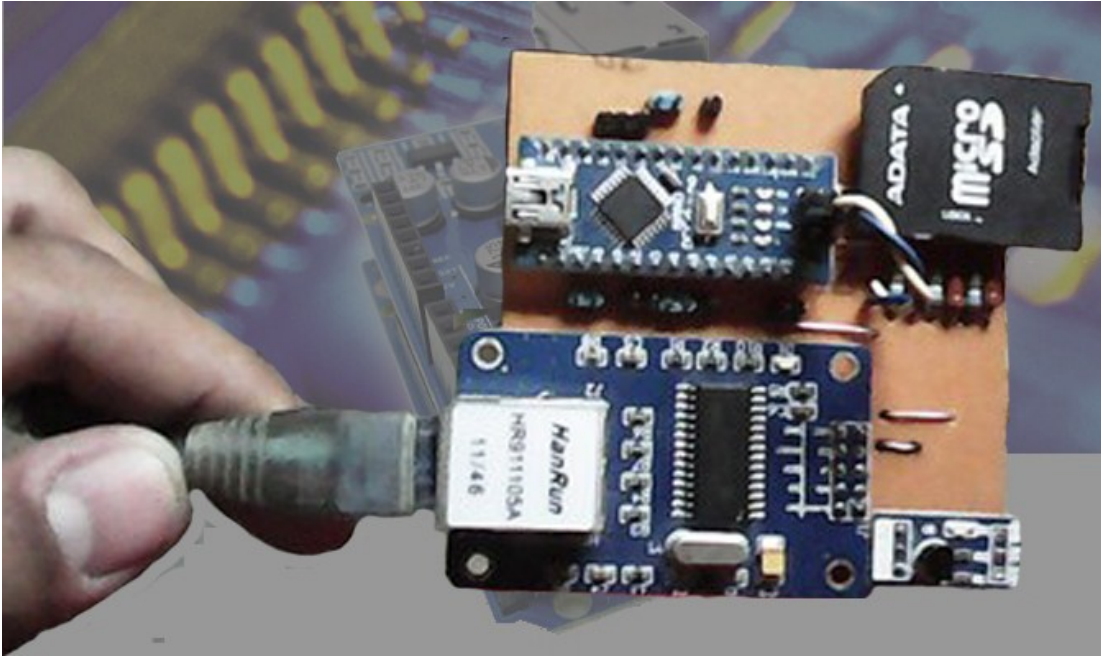
<html>
<head>
<title>Registrador de variables de la Universidad de Oriente</title>
</head>
<body>
<fieldset>
<FORM METHOD = POST ACTION = http://169.254.128.100>
<legend><FONT FACE="impact"><h1>CONFIGURACION GENERAL</h1></FONT></legend>
<fieldset>
<legend><FONT FACE="impact" SIZE=4>CONFIGURACION DE RED</FONT></legend>
<div>DIRECCION IP SERVIDOR:
<input type="text" name="IPserver0" value="169" maxlength="3" size="4"/> .
<input type="text" name="IPserver1" value="254" maxlength="3" size="4"/> .
<input type="text" name="IPserver2" value="128" maxlength="3" size="4"/> .
<input type="text" name="IPserver3" value="100" maxlength="3" size="4"/></div>
<div>DIRECCION IP PUERTA ENLACE:
<input type="text" name="IPgateway0" value="169" maxlength="3" size="4"/> .
<input type="text" name="IPgateway1" value="254" maxlength="3" size="4"/> .
<input type="text" name="IPgateway2" value="128" maxlength="3" size="4"/> .
<input type="text" name="IPgateway3" value="100" maxlength="3" size="4"/></div>
<div>PUERTO SERVIDOR:
<input type="text" name="portserver" value="80" maxlength="4" size="4"/></div>
<div>DIRECCION MAC:
<input type="text" name="MAC0" value="169" maxlength="3" size="4"/> :
<input type="text" name="MAC1" value="254" maxlength="3" size="4"/> :
<input type="text" name="MAC2" value="128" maxlength="3" size="4"/> :
<input type="text" name="MAC3" value="100" maxlength="3" size="4"/> :
<input type="text" name="MAC4" value="169" maxlength="3" size="4"/> :
<input type="text" name="MAC5" value="254" maxlength="3" size="4"/></div><hr />
<div>DIRECCION IP CLIENTE:
<input type="text" name="IPclient0" value="169" maxlength="3" size="4"/> .
<input type="text" name="IPclient1" value="254" maxlength="3" size="4"/> .
<input type="text" name="IPclient2" value="128" maxlength="3" size="4"/> .
<input type="text" name="IPclient3" value="100" maxlength="3" size="4"/></div>
<div>PUERTO CLIENTE:
<input type="text" name="portclient" value="80" maxlength="4" size="4"/></div>
</fieldset>
<fieldset>
<legend><FONT FACE="impact" SIZE=4>CONFIGURACION DE TEMPERATURA</FONT></legend>
<div>TL:
<input type="text" name="tl" value="21" maxlength="2" size="4"/>
TH:
<input type="text" name="th" value="23" maxlength="2" size="4"/></div>
<div>
NOTA: TH y TL son los valores maximos y minimos del ciclo de histeresis que se tiene en cuenta para
mantener la temperatura en el local dentro de los valores establecidos. TH es la temperatura a la
cual se encenderan los aires de reserva que no se volvergan a apagar hasta que la temperatura en el
local no este por debajo de TL.
</div>
</fieldset>
<fieldset>
<legend><FONT FACE="impact" SIZE=4>CONFIGURACION DEL TIEMPO DE ROTACION</FONT></legend>
<div>HORAS:
<input type="text" name="trot" value="12" maxlength="2" size="4"/></div>
<div>
NOTA: El tiempo de rotacion es el tiempo que van a estar funcionando los aires en modo fijo o de reserva,
permitiendo con esto un minimo de tiempo de explotacion diaria a estos equivos.
</div>
</fieldset>
<input type= submit value="Enviar Datos" />
</FORM>

```

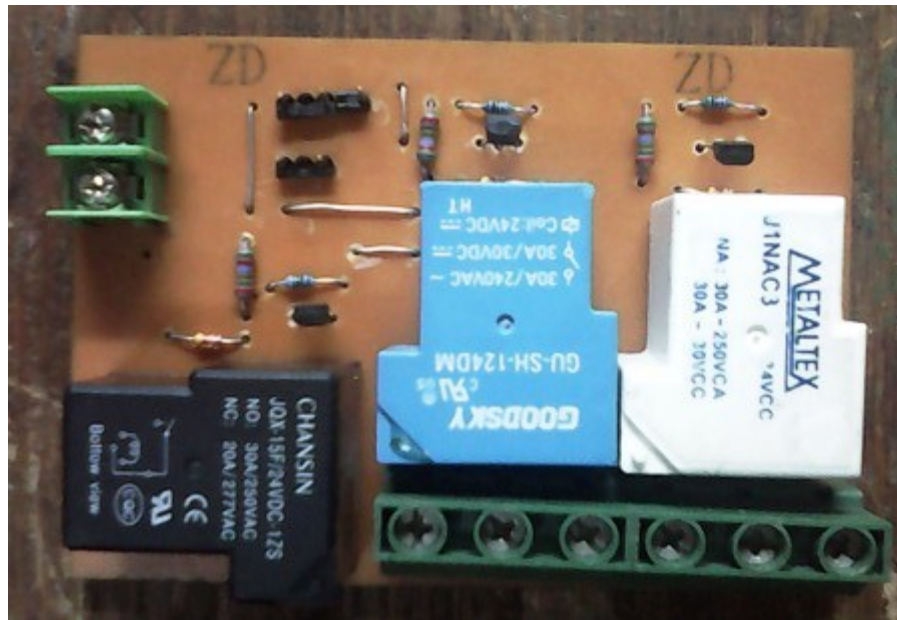
ANEXOS VII. PLANTA DE FUERZA Y ALIMENTACIÓN.



ANEXOS VIII. DISPOSITIVO DE CONTROL Y SUPERVISIÓN.



Módulo Central del dispositivo de supervisión y control.



Circuito de Fuerza.