

UNIVERSIDAD NACIONAL DE INGENIERIA
Facultad de Ingeniería Industrial y de Sistemas
Sección de Post-Grado



*"RECEPCIÓN DE UNA SEÑAL A DISTANCIA EN
TIEMPO REAL"*

TESIS

**Para optar el Grado de Maestro en Ciencias:
MENCION EN INGENIERIA DE SISTEMAS**

Ing. Jimmy Aurelio Rosales Huamani

Lima - Perú

2001

INDICE

DESCRIPTORES TEMÁTICOS.....	9
------------------------------------	----------

RESUMEN.....	10
---------------------	-----------

INTRODUCCION	13
---------------------------	-----------

CAPITULO I

LOS SISTEMAS DE TIEMPO REAL (STR)

<i>1.1. Definición de un STR.....</i>	<i>15</i>
<i>1.2. Clasificación de un STR.....</i>	<i>16</i>
<i>1.3. Características de los STR.....</i>	<i>17</i>
<i>1.4. Consideraciones sobre los sistemas</i>	<i>18</i>
1.4.1 Aspectos de integración y rendimiento	18
1.4.2 Manejo de interrupciones.....	22
1.4.3 Sistemas operativos de tiempo real.....	25
1.4.4 Software de tiempo real	26
1.4.5 Sincronización y comunicación de tareas.....	28
<i>1.5. Componentes de un STR.....</i>	<i>29</i>
<i>1.6. Areas de aplicación</i>	<i>31</i>
<i>1.7. Estructura genérica de un STR</i>	<i>32</i>

CAPITULO II

ASPECTOS GENERALES DE LOS EQUIPOS USADOS EN LA EXPERIENCIA

<i>2.1. Los sensores.....</i>	<i>33</i>
2.1.1 Clasificación.....	33
<i>2.2. El puerto paralelo.....</i>	<i>34</i>
<i>2.3. El hardware del puerto paralelo</i>	<i>38</i>

CAPITULO III

REQUERIMIENTOS E IMPLEMENTACION DEL HARDWARE DE ADQUISICION DE DATOS

3.1	<i>Adquisición de datos</i>	44
3.2	<i>Hardware de un sistema de adquisición de datos</i>	46
3.2.1	Arquitectura general.....	46
3.2.1.1	Multiplexores analógicos.....	47
3.2.1.2	Circuitos de muestra y retención (Sample/Hold. S/H).....	48
3.2.1.3	Conversores analógico/digital (A/D).....	48
3.2.1.4	Conversores digital/analógico (D/A).....	49
3.2.1.5	Acondicionamiento de la señal.....	49
3.3	<i>Ventajas de los sistemas de adquisición de datos</i>	49
3.4	<i>Componentes y periféricos para la automatización con la computadora</i>	50
3.5	<i>Descripción de las tarjetas de adquisición de datos</i>	51
3.5.1	Tarjetas A/D.....	51
3.5.2	Tarjetas D/A.....	52
3.5.3	Tarjetas I/O.....	53
3.6	<i>Costo de un Sistema de Adquisición de Datos</i>	54
3.7	<i>Implementación del hardware</i>	55

CAPITULO IV

IMPLEMENTACION DE PROGRAMAS

4.1	<i>Programa 1: Creación del modulo de tiempo real</i>	61
4.2	<i>Programa 2: Creación del programa de usuario</i>	64
4.3	<i>Programa 3: Creación del makefile</i>	66
4.4	<i>Programa 4: Búsqueda de direcciones puerto serial y paralelo</i>	68
4.5	<i>Breve explicación de los programas</i>	69
4.6	<i>Diagrama de Procesos</i>	74

CAPITULO V

PARTE EXPERIMENTAL

5.1	<i>Metodología a usar</i>	79
5.1.1	El sensor de luz.....	79
5.1.2	El conector del puerto paralelo.....	80
5.1.3	El hardware de adquisición de datos.....	80
5.1.4	Detalles de los registros de estado.....	80
5.1.5	Conexión con la computadora.....	83
5.1.6	Resultados obtenidos.....	84

CAPITULO VI

APLICACIÓN EN EL ANALISIS INSTRUMENTAL

6.1	<i>Clasificación de los métodos analíticos</i>	88
6.1.1	Métodos clásicos	89
6.1.2	Métodos instrumentales	89
6.1.2.1	Tipos de métodos instrumentales.....	90
6.2	<i>Instrumentos para el análisis</i>	93
6.3	<i>La selección del método analítico</i>	96
6.4	<i>Términos usados en la espectroscopia de absorción</i>	97
6.4.1	Transmitancia (T)	97
6.4.2	Absorbancia (A)	98
6.5	<i>El fotómetro</i>	99
6.6	<i>Parte experimental</i>	100
6.6.1	Resultados obtenidos	101
6.6.1.1	1ra experiencia	101
6.6.1.2	2da experiencia.....	102
6.6.1.3	3ra experiencia	103
6.6.1.4	4ta experiencia.....	104
6.6.1.5	5ta experiencia.....	104
6.6.1.6	6ta experiencia.....	105
6.7	<i>Tratamiento estadístico de los resultados experimentales</i>	107

ANEXO I

EL SISTEMA OPERATIVO LINUX	116
---	------------

ANEXO II

EL REAL TIME LINUX (RT-LINUX)	138
--	------------

CONCLUSIONES	158
---------------------------	------------

RECOMENDACIONES	160
------------------------------	------------

BIBLIOGRAFIA Y ENLACES	162
-------------------------------------	------------

INDICE DE FIGURAS

N ° de Figura

1	Manejo de Interrupciones	23
2	Estructura genérica de un STR.	32
3	Diagrama de bloques de un sistema de adquisición de datos.	46
4	Arquitectura de un sistema de adquisición de datos.	47
5	Esquema de la tarjeta de adquisición de datos A/D.	52
6	Esquema de la tarjeta de adquisición de datos D/A.	53
7	Esquema de la tarjeta de adquisición de datos I/O.	54
8	Esquema del Hardware de adquisición de datos.	59
9	Visión global del sistema de conexión eléctrica.	84
10	Instrumentos para el análisis.	93
11	Esquema de la medida de la transmitancia.	97
12	Perdidas por reflexión y dispersión.	99
13	El fotómetro.	100
14	Esquema de la parte experimental a realizar.	100
15	Gráfica de la transmitancia vs. espesor.	106
16	Gráfica de la absorbancia vs. espesor.	107
17	Constitución de un sistema de computación moderno.	117
18	Estructura del sistema operativo Linux.	128
19	Arquitectura del sistema operativo.	131
20	Las llamadas al sistema de las extensiones Posix.	137
21	Las tareas de tiempo real	142

INDICE DE TABLAS

N ° de Tabla

1	Las funciones de cada pin del conector 1284 tipo A.	39
2	El registro de datos.	41
3	El registro de Estado.	41
4	El registro de control.	42
5	Los métodos instrumentales.	92
6	Componentes de los instrumentos para el análisis.	96

DESCRIPTORES TEMÁTICOS

1. Sistemas de tiempo real
2. Adquisición de datos
3. Sistemas Operativos
4. El Puerto paralelo
5. Linux
6. RT-Linux

RESUMEN

El presente trabajo trata del estudio de los sistemas de tiempo real y una aplicación en el campo de la Ingeniería.

Se puede decir que un Sistema de Tiempo Real es aquel sistema informático cuyo correcto funcionamiento no depende solo de que proporcione la respuesta adecuada a la entrada, sino que también lo haga en un determinado momento, este es el aspecto fundamental que diferencia a los Sistemas de Tiempo Real del resto de los sistemas convencionales.

El diseño de un sistema de tiempo real requiere de técnicas de análisis, diseño y pruebas que son algunas veces desconocidas.

Como cualquier sistema basado en computadora, un sistema de tiempo real debe integrar hardware, software, personas y elementos de una base de datos, para conseguir adecuadamente un conjunto de requisitos funcionales y de rendimiento.

Las áreas de aplicación de los Sistemas de Tiempo Real en la actualidad son bastantes numerosas y muchas de ellas se usan por ejemplo en :

Control de procesos, Robótica, Estructuras, Sistemas de Potencia, Sistemas de gran escala, Inteligencia Artificial, Sismología, Electrónica de consumo, Edificios inteligentes, Control de tráfico aéreo, Telecomunicaciones, Análisis químico, etc.

Mi planteamiento de elaboración de la tesis, será aplicar esta técnica de tiempo real en la medida de la señal de luz mediante un sensor adaptado a nuestro sistema y luego obteniendo los datos en el computador, usando para ello el puerto paralelo de la impresora.

Este puerto se usará debido a que una computadora normal no dispone de una tarjeta de adquisición de datos para leer dicha tensión (voltaje).

El trabajo que va a ser realizado nos dará una idea de cómo este sistema puede implementarse en forma sencilla y que parámetros deberán usarse. Este trabajo lo resumiremos en cuatro partes:

La primera parte consta del desarrollo de la parte teórica de lo que es un Sistema de Tiempo Real y su clasificación.

La segunda parte consta del desarrollo del hardware relacionado con la conexión del sensor de luz al computador.

La tercera parte consta del desarrollo del Sistema de Tiempo Real (implementación de software y programas de aplicación).

La ultima parte consta de una aplicación de los Sistemas de Tiempo Real en una de las áreas ya mencionadas.

Finalmente en cuanto al software a usar se menciona que se trabajara con el Sistema Operativo Linux y su extensión de tiempo real el RT-Linux.

INTRODUCCION

En el pasado se han instalado computadores que podían tomar decisiones en fracciones de segundos pero no producían la información, donde se necesitaba, con la rapidez requerida.

Por lo tanto, a pesar de la alta velocidad de los computadores, se les tenía rodeados de un sistema y organización que no se podían mantener a su paso.

Ahora, en cambio, todos los datos pueden estar a disposición inmediata de los computadores por medio de archivos de acceso directo. Además con las líneas de telecomunicaciones se puede comunicar al computador con cualquier punto geográfico y acelerar el flujo de datos desde y hacia el computador.

Actualmente muchas instalaciones tienen en uso sistemas de **critos como en línea y muy pocos de tiempo real.**

La motivación fundamental del presente trabajo es el estudio y aplicación de los sistemas de tiempo real como una solución a problemas que se presentan en los diversos campos de la ingeniería.

Por ejemplo cuando se quiere implementar un sistema convencional cualquiera se presentan dos problemas:

Primero que el sistema tenga una respuesta adecuada a la entrada y segundo que lo haga en un determinado momento. Estos problemas podrán ser resueltos implementando un sistema de tiempo real.

En la década pasada, el desarrollo de los sistemas de tiempo real se consideraba un arte oscuro. Hoy en día la Ingeniería de Software de Tiempo Real requiere de practicas especiales.

En el presente trabajo se dará las pautas para implementar un Sistema de Tiempo Real sin necesidad de tener que adquirir equipos automatizados y caros, además se necesitara una metodología adecuada para dicha implementación.

Finalmente se menciona que uno de los propósitos de esta tesis ha sido el de entregar este trabajo con un alto contenido experimental.

CAPITULO I

LOS SISTEMAS DE TIEMPO REAL (STR)

1.1. Definición de un STR

Un sistema de tiempo real (STR) es aquel sistema informático en el que la corrección del sistema no solo depende de los resultados lógicos de los algoritmos, sino que también depende del momento en el que estos se producen.

Es decir no es suficiente que los resultados obtenidos sean correctos, sino que tienen que obtenerse dentro de un intervalo de tiempo especificado.

Si las restricciones de tiempo no se cumplen, se dice que ha ocurrido una falla en el sistema, por lo tanto es esencial que se garantice el cumplimiento de las restricciones de tiempo del sistema. Garantizar el comportamiento temporal requiere que el sistema sea predecible, también es deseable que el sistema logre un alto grado de utilización cumpliendo las restricciones de tiempo del sistema.

De esta definición no se deduce que un sistema de tiempo real tenga que ser necesariamente rápido, como quizás se pudiera pensar. Por ejemplo: el sistema de

guiado de un barco puede parecer en principio que **no** es de tiempo real, pues la velocidad del barco es relativamente baja y el sistema dispone de bastante tiempo (del orden de algunos minutos) para tomar las decisiones de control. Pero se sabe que **si** es un sistema de tiempo real según la definición anterior

En otras palabras, un resultado lógicamente correcto pierde todo significado si el tiempo de espera por el mismo sobrepasa ciertos límites preestablecidos.

1.2. Clasificación de un STR

Los sistemas de tiempo real se pueden clasificar de dos formas:

En base al tiempo de respuesta pueden ser:

a) Sistemas de Tiempo Real Duros(Hard real time)

Los STR duros son aquellos sistemas que tienen un tiempo de ejecución definido, una sola respuesta fuera de plazo compromete la seguridad del sistema.

Por ejemplo: El control de temperatura de una central nuclear.

b) Sistemas de Tiempo Real Blandos(Soft real time)

Los STR blandos son los que tienen restricciones reducidas en la temporalidad, pero que deben operar rápidamente y de forma repetida. Se puede tolerar que se sobrepase algún plazo de forma ocasional, pero no compromete su seguridad.

Por ejemplo: Reservas de avión, cajeros automáticos.

Sobre la base del punto de vista físico pueden ser:

a) Sistemas embebidos(Embedded system)

Cuando los sistemas de tiempo real suelen ser componentes de otros sistemas, en los que realizan funciones de control. Se puede decir que está completamente encapsulado por el hardware que lo controla.

b) Sistemas orgánicos(Organic system)

El software del sistema no es altamente independiente del hardware en el cual funciona y generalmente usan interfases.

1.3. Características de los STR

Deben responder a diversos eventos externos, asegurando un tiempo de respuesta máximo determinado(tiempo de respuesta que transcurre entre la presentación de un conjunto de entradas al sistema y la obtención de sus resultados asociados)

La secuencia de ejecución de las tareas del sistema no solo esta determinada por decisiones del sistema, sino por eventos que ocurren en el mundo real. Normalmente esta secuencia es asincrónica, lo que implica demoras, sincronizaciones, sobrecargas del subsistema de entrada/salida y posibilidades de conflicto temporal por los recursos.

Deben ofrecer facilidades de interfaz con una gran cantidad de dispositivos y periféricos, fundamentalmente sensores y actuadores.

Deben presentar alto nivel de seguridad. Es crítico asegurar la confiabilidad del sistema, así como su respuesta controlada ante situaciones de sobrecarga.

Sé autorrecuperan de situaciones de excepción.

No pueden volver atrás y reiniciar su ejecución desde un contexto preexistente.

Las demandas del ambiente externo a un sistema de tiempo real suelen ser en paralelo, provocando problemas de planificación y prioridades.

1.4. Consideraciones sobre los sistemas

Como cualquier sistema basado en un computador , un sistema de tiempo real debe integrar hardware, software y personas para conseguir adecuadamente su implementación, además de su correcto funcionamiento.

1.4.1 Aspectos de integración y rendimiento

Globalmente, un sistema de tiempo real enfrenta al ingeniero de sistemas con difíciles decisiones sobre el hardware y el software. Una vez que el elemento

software ha sido designado, se establecen los requisitos detallados del software y debe desarrollarse un diseño fundamental del software. Entre los muchos aspectos relativos al diseño del tiempo real están: la coordinación entre las tareas de tiempo real, el procesamiento de interrupciones del sistema, el manejo de entradas y salidas para asegurar que no se pierdan datos, la especificación de las ligaduras de tiempo externas del sistema y el asegurar la precisión de su base de datos.

Cada diseño de tiempo real relativo al software debe ser aplicado en el contexto del rendimiento del sistema. Un factor importante a considerar es que los patrones que se utilizan para medir el mérito de estos sistemas deben ser distintos a los usados para evaluar sistemas tradicionales. Criterios como el rendimiento(cantidad de trabajos por unidad de tiempo), tiempo de retorno(tiempo entre el comienzo y fin de un trabajo), grado de utilización de recursos, o un tiempo medio de respuesta no son indicativos.

En cambio se utilizan los siguientes conceptos.

a) Predictibilidad

Se desea obtener respuesta predecible ante eventos urgentes. No importa la velocidad de dichas respuestas, sino asegurar que las mismas siempre se obtengan antes de una meta dada. Para cumplir con ese objetivo, al activar una tarea debe ser posible determinar con certeza su tiempo de terminación.

Para ello hay que tomar en cuenta el estado del sistema y las necesidades de recursos de las tareas.

b) Planificabilidad

El análisis de planificabilidad consiste en aplicar unas pruebas al conjunto de tareas de tal forma que si estas pasan el test entonces se puede garantizar que ninguna tarea perderá su plazo de ejecución. Si no pasan el test entonces se tiene que volver al principio y empezar de nuevo, utilizando otro procesador más potente o utilizando otros algoritmos para implementar las tareas.

c) Estabilidad

Ante sobrecargas momentáneas. Cuando el sistema está sobrecargado y es imposible cumplir las metas de todas las tareas, aún debemos garantizar el cumplimiento de algunas tareas críticas elegidas.

d) Confiabilidad

Es un pre-requisito para los sistemas de tiempo real. Las restricciones de tiempo real no pueden cumplirse si los componentes del sistema no son confiables. La mayoría de los sistemas de tiempo real debe alcanzar alto estándares de confianza, eficacia y seguridad, ya que el costo de una falla del

sistema puede exceder la inversión en la computadora y el objeto controlado. El diseño del sistema debe garantizar el desempeño en todas las situaciones (preferentemente de forma anticipada), y el sistema debe diseñarse para tolerar fallas en el hardware o las causadas por disturbios del ambiente.

e) Adaptabilidad

Los cambios en la configuración del sistema, en las especificaciones del sistema y en el estado (Por ejemplo, ante sobrecargas o fallas). La adaptabilidad es particularmente importante, por que si un sistema es adaptable, uno no tiene que redefinir el sistema o recalcular asignación de recursos y tareas para cada cambio pequeño. Dos requisitos que se derivan de la adaptabilidad, son la mantenibilidad y la expansibilidad, por ende, un sistema adaptable permite reducir costos de desarrollo y mantenimiento.

Lograr que un sistema de Tiempo Real tenga las características mencionadas previamente suele resultar muy complejo, ya que muchos de estos requerimientos son conflictivos entre sí y muchas veces la organización del hardware aumenta aún más la complejidad, ya que las computadoras, sensores y actuadores pueden estar separados y funcionando en ambientes aislados. El control mismo del sistema puede estar compartido entre varias computadoras, no necesariamente todas en el mismo sitio y, por lo tanto, hay que transmitir información entre las mismas, siempre cumpliendo la meta .

Si el sistema esta compuesto por una sola tarea, entonces no hay problema de tiempo real, el procesador puede hacer el trabajo en el tiempo requerido o no. En caso de no ser lo suficientemente rápido el procesador, entonces simplemente se cambia de procesador.

Los problemas de tiempo real (TR) aparecen cuando el sistema esta compuesto por varias tareas, y hay que repartir el procesador (o procesadores) entre todas ellas. Para ello no podemos utilizar un sistema clásico de tiempo compartido como puede ser el utilizado por Linux con los procesos normales.

El objetivo del sistema es garantizar que todas las tareas (en todas sus actividades) cumplan sus plazos de ejecución. La forma de conseguirlo es teniendo un sistema predecible.

1.4.2 Manejo de interrupciones

Una característica que sirve para distinguir a los sistemas de tiempo real de cualquier otro tipo es el manejo de interrupciones. Un sistema de tiempo real debe responder a un estímulo externo(interrupción) en un tiempo dictado por el mundo externo.

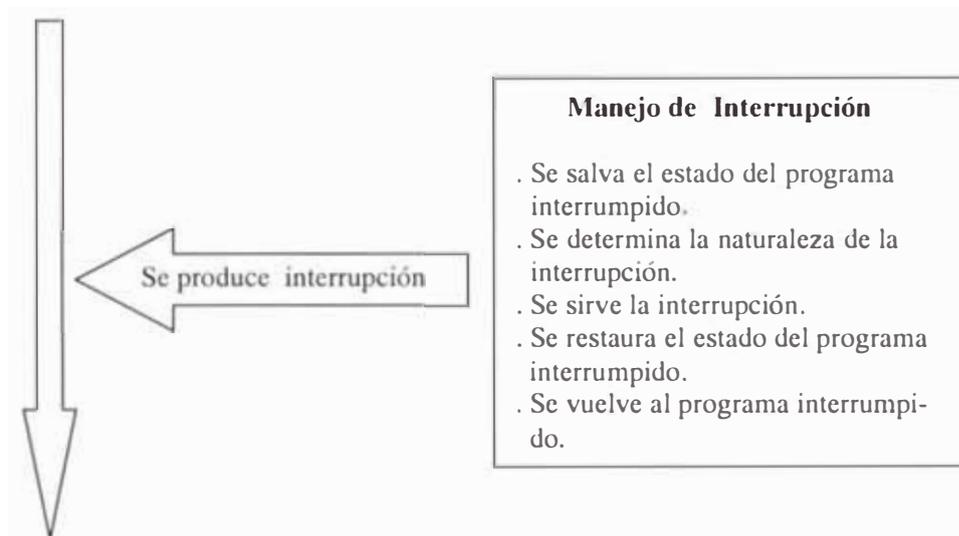
Debido a que frecuentemente, se presentan múltiples interrupciones, deben establecerse prioridades e interrupciones prioritarias. En otra palabras,

la tarea más importante debe ser siempre servida dentro de las restricciones de tiempo predefinidas, independientemente de otros sucesos.

El manejo de interrupciones supone, no solo almacenar información, de forma que la computadora pueda establecer correctamente la tarea interrumpida, sino también evitar ínter bloqueos y bucles sin fin. El enfoque global para el manejo de interrupciones se ilustra a continuación.

Figura N ° 1
MANEJO DE INTERRUPCIONES

Flujo normal de procesamiento



Se observa como el flujo normal de procesamiento es interrumpido por un suceso que es detectado por el hardware del procesador. Un suceso es cualquier ocurrencia que necesita un servicio inmediato y que puede ser generado por el hardware o software. Se salva el estado del programa interrumpido (guardando todos los contenidos de los registros, los bloques de

control, etc.) y se pasa el control a una rutina de servicio de interrupción, que bifurca al software apropiado para el manejo de la interrupción. Al terminar el servicio de interrupción, se restaura el estado de la maquina y continua el flujo normal de procesamiento.

En muchas situaciones, el servicio de interrupción de un suceso puede a su vez ser interrumpido por otro suceso de mayor prioridad. Pueden establecerse niveles de prioridad de interrupciones.

Para manejar las interrupciones y cumplir también las restricciones de tiempo del sistema, muchos sistemas operativos de tiempo real hacen cálculos dinámicamente para determinar si pueden cumplirse los objetivos del sistema. Estos cálculos dinámicos se basan en la frecuencia media de ocurrencia de sucesos, la cantidad de tiempo que le lleva el servirlos y las rutinas que pueden interrumpirlos y temporalmente evitar su servicio.

Si los cálculos dinámicos muestran que es imposible manejar los sucesos que pueden ocurrir en el sistema y también cumplir las restricciones de tiempo, el sistema debe decidir sobre un plan de acción.

Un posible esquema consiste en almacenar en un buffer los datos, de forma que puedan ser procesados rápidamente cuando el sistema este preparado.

1.4.3 Sistemas operativos de tiempo real

Algunos sistemas operativos de tiempo real son aplicables a una gran gama de configuraciones de sistemas, mientras que otros están orientados a un enfoque en particular, independientemente del entorno electrónico circundante.

Hay dos amplias clases de sistemas operativos que se utilizan para los trabajos de tiempo real:

- Un sistema operativo diseñado exclusivamente para aplicaciones de tiempo real.
- Un sistema operativo de propósito general que se ha reforzado para suministrar capacidades de tiempo real con un gestor de tiempo real. Comportándose como software de aplicación, el gestor ejecuta varias funciones del sistema operativo particularmente las que afectan el rendimiento de tiempo real de una forma más rápida y eficiente que el sistema operativo.

Todos los sistemas operativos deben tener un mecanismo de planificación de prioridades, pero los sistemas operativos de tiempo real deben dar un mecanismo de prioridades que permita que las interrupciones de prioridad alta tengan precedencia sobre la menos importante. Además, debido a que las interrupciones ocurren en respuesta a sucesos asíncronos no

recurrentes, deben ser servidas sin consumir primero un tiempo de carga de un programa de disco.

Para garantizar el tiempo de respuesta requerido, un sistema operativo de tiempo real debe tener un mecanismo de bloqueo de memoria, es decir, mantener unos mínimos programas en memoria principal, de forma que se evite la sobrecarga de almacenamiento en ella.

Para determinar que tipo de sistema operativo real es mas adecuado a una aplicación, puede definirse y evaluarse medidas de la calidad de un sistema operativo de tiempo real.

1.4.4 Software de tiempo real

El software que mide, analiza, controla sucesos del mundo real conforme ocurren, se denomina de tiempo real. Entre los elementos del software de tiempo real se incluye:

- Un componente de adquisición de datos que recolecta y da formato a la información recibida del entorno externo.
- Un componente de análisis que transforma la información según lo requiera la aplicación.
- Un componente de control/salida que responda al entorno externo.

- Un componente de monitorización que coordina con todos los demás componentes, de forma que pueda mantenerse la respuesta en tiempo real.

Se conoce que debido a los requisitos especiales de rendimiento y de fiabilidad demandados por los sistemas de tiempo real, es importante la elección del lenguaje de programación, aunque dichos lenguajes existentes en la actualidad no son muy apropiado para el desarrollo de un sistema de tiempo real, pueden usarse con efectividad muchos lenguajes de programación de propósito general como C, Fortran, Modula-2. Sin embargo, se tiene una clase de lenguajes de tiempo real como Ada, Jovial, y otros que se utilizan frecuentemente en aplicaciones especializadas de comunicaciones militares.

Varias características hacen un lenguaje de tiempo real diferente de un lenguaje de propósito general. Estas incluyen la capacidad de multitarea, las construcciones para implementación directa de funciones y las características modernas de programación que ayuden a asegurar la corrección del programa.

Es importante además que el lenguaje de programación soporte directamente la multitarea debido a que los sistemas de tiempo real deben responder a sucesos asíncronos que ocurren simultáneamente. Aunque muchos sistemas operativos de tiempo real dan capacidades de multitarea, frecuentemente existe software de tiempo real empotrado sin un sistema operativo.

1.4.5 Sincronización y comunicación de tareas

Un sistema de multitarea debe suministrar un mecanismo por el que las tareas se pa en información unas a otras, así como asegurar su sincronización. Para estas funciones, los sistemas operativos y los lenguajes con soporte de tiempo real, utilizan frecuentemente semáforos de colas, buzones o sistemas de mensajes.

Los semáforos suministran sincronización y señalización pero no contienen información. Los mensajes son similares a los semáforos excepto en que ellos llevan una información asociada. Por otra parte, los buzones no señalizan la información sino que la contienen.

En un sistema de tiempo real, los semáforos se utilizan frecuentemente para implementar y gestionar buzones. Los buzones se almacenan temporalmente en buffers para enviar mensajes de un proceso a otro. Un proceso produce una información, la sitúa en el buzón y luego señala a un proceso consumidor que hay una información en el buzón para que la utilice.

Algunos métodos para los sistemas operativos de tiempo real ven los buzones como la forma más eficiente de implementar comunicaciones entre procesos.

Otro método para la comunicación y sincronización entre procesos es un sistema de mensajes. Con un sistema de mensajes, un proceso envía un mensaje a otro. El último se activa entonces automáticamente por el sistema de soporte de tiempo de ejecución o sistema operativo para que procese el mensaje.

Tal sistema incurre en sobrecarga debido a la transferencia real de la información, pero suministra una mayor flexibilidad y facilidad de uso

1.5 Componentes de un STR

Los componentes de un sistema de tiempo real suelen ser:

a) Computadora

El decrecimiento en los costos de las computadoras y en el desarrollo de hardware y software especializado, ha hecho posible que las computadoras sean empleadas en tareas de control, monitoreo, procesamiento de información, supervisión, etc. En la actualidad se ha llegado a que las computadoras puedan realizar el control de tiempo real, cumpliendo las siguientes funciones:

- Adquisición de datos. Recolección y registro de datos
- Control Digital Directo. Regulación y seguimiento automático de consignas.
- Control Secuencial. Control lógico discreto de tareas.
- Interfase de Operador. Monitoreo, control y actuación.

- Control Supervisorio. Comprobación del correcto funcionamiento de la planta.
- Control Distribuido. Distribución de las funciones de control entre diferentes PC.

b) Interfases de entrada y salida

c) Planta

Objeto a controlar con ayuda de un sistema, en el cual son necesarios y permitibles acciones organizadas con el fin de lograr los resultados de funcionamiento deseado.

d) Elementos de medición

Componentes que permiten conocer el valor de las variables de salida por ejemplo: los sensores, transmisores.

e) Elementos finales de control

Componente que permite realizar la actuación directa sobre la planta, frecuentemente se trata de una válvula de control aunque no siempre. Otros elementos finales de control comúnmente utilizados son las bombas de velocidad variable, motore eléctricos.

f) Driver

Es un grupo de datos en el núcleo que constituye la interface con algún dispositivo hardware(puerto de entrada o salida).

Existen ciertas consideraciones para elaborar un driver como:

- Definir las funciones a implementar
- Averiguar como trabaja el hardware
- Averiguar los puertos de E/S a usar
- Definir el tipo de transferencia de datos de acuerdo al hardware
- Documentar el driver.

g) Interfase de usuario y operador

Hay que mencionar que dependiendo del área de aplicación se utilizaran algunos de estos componentes.

1.6. Areas de aplicación

Los sistemas de tiempo real son usados en un amplio rango de aplicaciones, estos pueden ser por ejemplo en el área de:

- Control de Procesos
- Control de Temperatura, nivel, presión, flujo
- Robótica
- Sistemas de Potencia
- Despacho de carga
- Producción y distribución de energía
- Sistemas de gran escala
- SCADA (Sistemas de adquisición control y supervisión)

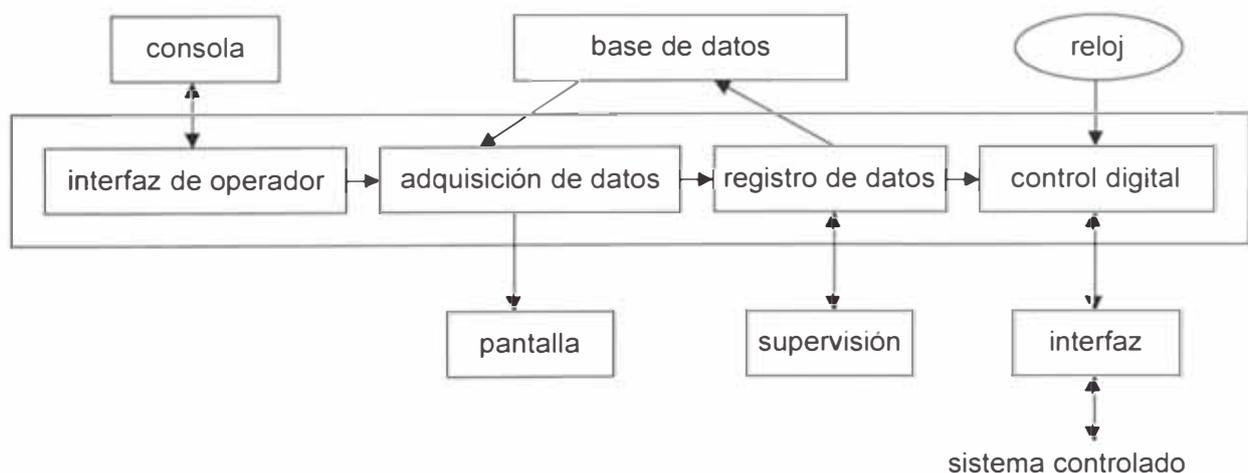
- Inteligencia Artificial
- Sismología
- Análisis Químico
- Telecomunicaciones
- Control de tráfico aéreo
- Electrónica en consumo
- Edificios inteligentes, etc.

1.7. Estructura genérica de un STR

En la siguiente figura se muestra la estructura genérica de un sistema de tiempo real, en el cual se muestra la interacción entre el entorno físico, el reloj para poder planificar el tiempo, las posibilidades de almacenamiento y visualización que provee la computadora.

Figura N ° 2

ESTRUCTURA GENERICA



CAPITULO II

ASPECTOS GENERALES DE LOS EQUIPOS USADOS EN LA EXPERIENCIA

Mencionaremos a continuación ciertos componentes fundamentales de nuestro trabajo

2.1. Los sensores

El sensor es un elemento que esta en contacto directo con una variable fisica que se desea medir, y que provee una salida de acuerdo al valor de dicha variable.

Como se conoce los sensores permiten recopilar vital información del entorno como temperatura, presión, luz, color, aceleración, posición, tacto, presencia, etc En general los sensores son dispositivos que convierten magnitudes fisicas(como temperatura, fuerza, velocidad) en magnitudes inteligibles por un sistema de lectura(por lo general eléctricas).

2.1.1 Clasificación

Los sensores pueden ser divididos en tres categorías según la complejidad de la información que estos proporcionen:

- a) **Umbral.** Se utilizan para detectar límites de movimiento y para detección por proximidad, se pueden encontrar desde interruptores mecánicos lentos hasta interruptores electrónicos que son muchos más rápidos.
- b) **Medida.** Normalmente dan información sobre magnitudes físicas como pueden ser la distancia, tamaño (con precisión de hasta 0,1mm), fuerza, temperatura, intensidad de luz.
- c) **Imagen.** Se utilizan para la adquisición de información relacionada con estructuras, formas, topología y aunque las necesidades de precisión son menos severas que para los sensores de medida, la complejidad de información es mucho mayor(pueden emplearse imágenes de una, dos o tres dimensiones).

También se pueden clasificar como: pasivos y activos.

En los *sensores pasivos* hay que proveer una tensión para obtener una respuesta, en cambio los *sensores activos* auto generan tensión. Aplicando sensores a un manipulador se incrementa la precisión del proceso a estudiar.

2.2. El puerto paralelo

Existen dos métodos básicos para transmisión de datos en las computadoras modernas. En un esquema de transmisión de datos en serie un dispositivo envía datos

tomado la información (los datos). Observe que ambas partes sincronizan su respectivo acceso a las líneas de datos, la parte receptora no leerá las líneas de datos hasta que la parte transmisora se le indique en tanto que la parte transmisora no colocará nueva información en las líneas de datos hasta que la parte receptora remueva información y le indique a la parte transmisora que ya ha tomado los datos, a esta coordinación de operaciones se le llama acuerdo o entendimiento. A esta coordinación de operaciones entre la parte transmisora y la parte receptora se le llama handshaking, que en español es el acto con el cual dos partes manifiestan estar de acuerdo, es decir se dan un apretón de manos.

El handshaking

Para implementar el handshaking se requieren dos líneas adicionales. La línea de estroboscopio (en inglés strobe) es la que utiliza la parte transmisora para indicarle a la parte receptora la disponibilidad de información. La línea de admisión (acknowledge) es la que utiliza la parte receptora para indicarle a la parte transmisora que ha tomado la información (los datos) y que está lista para recibir más datos. El puerto paralelo provee de una tercera línea de handshaking llamada en inglés busy (ocupado), esta la puede utilizar la parte receptora para indicarle a la parte transmisora que esta ocupada y por lo tanto la parte transmisora no debe intentar colocar nueva información en las líneas de datos. Una típica sesión de transmisión de datos se parece a lo siguiente:

Parte transmisora:

La parte transmisora chequea la línea busy para ver si la parte receptora está ocupada. Si la línea busy e tá activa, la parte transmisora espera en un bucle hasta que la línea busy esté inactiva.

La parte transmisora coloca la información en las líneas de datos.

La parte transmisora activa la línea de strobe.

La parte transmisora espera en un bucle hasta que la línea acknowledge está activa.

La parte transmisora inactiva la línea de strobe.

La parte transmisora espera en un bucle hasta que la línea acknowledge esté inactiva.

La parte transmisora repite los pasos anteriores por cada byte a ser transmitido.

Parte receptora:

La parte receptora inactiva la línea busy(asumiendo que está lista para recibir información).

La parte receptora espera en un bucle hasta que la línea strobe esté activa.

La parte receptora lee la información de las líneas de datos(y sí es necesario, procesa los datos).

La parte receptora activa la línea acknowledge.

La parte receptora espera en un bucle hasta que esté inactiva la línea de strobe.

La parte receptora inactiva la línea acknowledge.

La parte receptora repite los pasos anteriores por cada byte que debe recibir.

Se debe ser muy cuidadoso al seguir estos pasos, tanto la parte transmisora como la receptora coordinan sus acciones de tal manera que la parte transmisora no intentará colocar varios bytes en las líneas de datos, en tanto que la parte receptora no debe leer más datos que los que le envíe la parte transmisora, un byte a la vez.

2.3. El hardware del puerto paralelo

El puerto paralelo de una típica PC utiliza un conector hembra de tipo D de 25 pines, este es el caso más común, sin embargo es conveniente mencionar que existen tres tipos de conectores definidos por el estándar IEEE 1284, el primero llamado 1284 tipo A es un conector hembra de 25 pines de tipo D.

El segundo es un conector 1284 tipo B que es un conector de 36 pines y el tercero se denomina 1284 tipo C.

Se menciona que el usado en la experiencia es el primer conector y el orden de los pines es el siguiente:

1 2 3 4 5 6 7 8 9 10 11 12 13
25 24 23 22 21 20 19 18 17 16 15 14

La siguiente tabla describe la función de cada pin del **conector 1284 tipo A:**

Tabla N ° 1

FUNCION DE CADA PIN DEL CONECTOR 1284

Pin	E/S	Polaridad activa	Descripción
1	Salida	0	Strobe
2-9	Salida	-	Líneas de datos
10	Entrada	0	Línea de acknowledge
11	Entrada	0	Línea busy
12	Entrada	1	Línea falta papel
13	Entrada	1	Línea select
14	Salida	0	Línea autofeed
15	Entrada	0	Línea error
16	Salida	0	Línea init
17	Salida	0	Línea select input
18-25	-	-	Tierra eléctrica

Se observa que el puerto paralelo tiene 12 líneas de salida y 5 de entrada, hay también tres direcciones de E/S asociadas con un puerto paralelo de la PC, estas son: **el registro de datos, el registro de estado, el registro de control.**

Estos tres puertos tienen ocho bits de longitud y cada uno de sus bits, representa un pin determinado del puerto. Con los tres registros de ocho bits cada uno se puede pensar controlar los 25 pines del puerto paralelo, pero la explicación es que no controlamos 25, sino 17. Los pines que van desde el pin 18 al pin 25 son para tierra.

El primero de los tres puertos es registro de datos, este es un puerto solo escritura de ocho bits. Solo se puede variar mediante la programación, pero ningún dispositivo lo podrá hacer electrónicamente.

El segundo puerto corresponde al registro de estado, este puerto solo es de entrada. Solo un dispositivo electrónico lo puede variar, pero mediante software es imposible. Este puerto lo utiliza la impresora para darnos información sobre su estado.

El tercer y último registro es el de control, este puerto es de escritura o lectura. Este se puede modificar tanto por el dispositivo electrónico como por el software.

Existen tres direcciones base para el puerto paralelo asociadas con tres posibles puertos paralelos como:

0x3BCh, 0x378h, 0x278h, nos referimos a estas como las direcciones base para el puerto LPT1, LPT2, LPT3, respectivamente. El registro de datos se localiza siempre en la dirección base de un puerto paralelo, el registro de estado aparece en la dirección base + 1, y el registro de control aparece en la dirección base + 2

Para trabajar con el puerto paralelo necesitamos en primer lugar conocer la dirección base asignada por el BIOS, en nuestro caso dicha dirección es **0x378h** (puede cambiar dependiendo de la PC), además para hallar dicha dirección se implementa un pequeño programa como se muestra más adelante. La distribución de las diferentes señales para cada uno de los tres registros del puerto paralelo están dadas por las siguientes tablas:

Tabla N° 2

EL REGISTRO DE DATOS

Dirección	Nombre	Lectura o Escritura	Bit #	Propiedades
Base + 0	Puerto de Datos	Escritura	Bit 7	Dato 7
			Bit 6	Dato 6
			Bit 5	Dato 5
			Bit 4	Dato 4
			Bit 3	Dato 3
			Bit 2	Dato 2
			Bit 1	Dato 1
			Bit 0	Dato 0

Tabla N° 3

EL REGISTRO DE ESTADO

Dirección	Nombre	Lectura o Escritura	Bit #	Propiedades
Base + 1	Puerto de Estado	Lectura	Bit 7	Busy
			Bit 6	Acknowledge
			Bit 5	Falta papel
			Bit 4	Select In
			Bit 3	Error
			Bit 2	IRQ (Not)
			Bit 1	Reservado
			Bit 0	Reservado

Tabla N° 4

EL REGISTRO DE CONTROL

Dirección	Nombre	Lectura o Escritura	Bit #	Propiedades
Base + 2	Puerto de Control	Lectura o Escritura	Bit 7	No usado
			Bit 6	No usado
			Bit 5	Permite puerto bidireccional
			Bit 4	Permite IRQ a través de acknowledge
			Bit 3	Selecciona impresora
			Bit 2	Inicializa impresora
			Bit 1	Nueva línea automática
			Bit 0	Strobe

Se menciona que la combinación de las 4 tablas anteriores nos sirven de gran ayuda para enlazar el conector con los diversos dispositivos electrónicos y utilizar el puerto para la adquisición de datos.

CAPITULO III

REQUERIMIENTOS E IMPLEMENTACION DEL HARDWARE DE ADQUISICION DE DATOS

Introducción

Aunque la mayoría de las computadoras digitales pueden usarse para sistemas de tiempo real, no todas ellas están igualmente adaptadas para tal trabajo. Tales computadoras deben poder comunicarse de forma efectiva tanto en el entorno como con el personal, a diferencia de una computadora tradicional donde prima solamente la comunicación con el usuario.

Una característica esencial de las computadoras es que sean modulares y proveen facilidades para agregar unidades, esto es fundamental para poder insertar tarjetas tales como: tarjetas digitales, tarjetas tipo relé, etc. Que se pueden insertar en un slot, igual que una red. Se menciona que una parte de nuestro trabajo se centrara en la adquisición de datos.

Los requerimientos de hardware de los sistemas de tiempo real implican el empleo de:

Computadora(PC, microprocesador, etc.)
Interfases de Entrada/Salida
Buses de Entrada/Salida
Periféricos de la computadora
Interfases del operador
Instrumentos de Medición/Actuación
Convertidores Digital/Analógico(D/A)
Convertidores Analógico/Digital(A/D)
Multiplexadores
Multiprocesadores
Circuitos de acondicionamiento de la señal

3.1 Adquisición de datos

Muchas de las variables físicas que se manejan son de naturaleza analógica y pueden tomar cualquier valor dentro de un rango continuo de estos. Ejemplos de variables de este tipo incluyen: temperatura, presión, intensidad luminosa, señales de audio, posición, velocidad de flujo entre otras. Los sistemas digitales llevan a cabo todas sus operaciones internas mediante el uso de circuitería y operaciones digitales.

Cuando un sistema digital tal como un computador va a utilizarse para vigilar y/o controlar un proceso físico, el diseñador se encuentra con fenómenos físicos que deben ser digitalizados para un tratamiento determinado.

Con el desarrollo y la disminución de los costos de la tecnología digital, se han desarrollado sistemas cada vez más complejos y eficaces para enfrentar el problema planteado. Uno de estos sistemas que se han desarrollado es el sistema de Adquisición y Conversión de Datos.

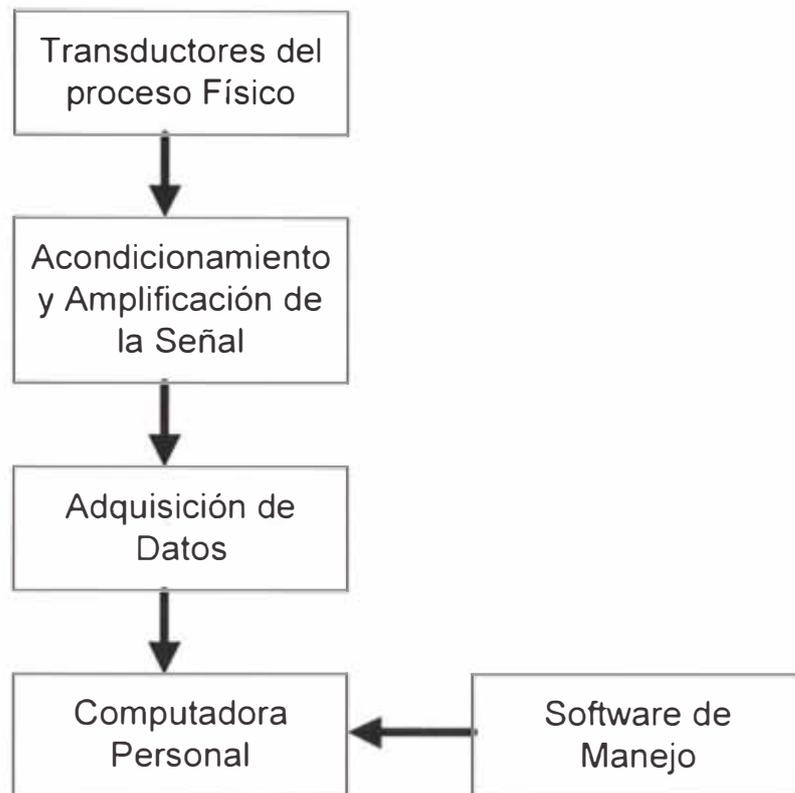
Un sistema de este tipo se emplea para adquirir señales analógicas y convertirlos a la forma digital para su procesamiento y análisis subsiguientes mediante computadoras o para la transmisión de datos, se emplean en un gran número de aplicaciones como: áreas industriales y científicas. También se emplean en control de procesos industriales tales como: químicos, petroquímicos, monitoreo ambiental, etc.

Luego un sistema de adquisición de datos se puede definir como un sistema que engloba tanto a los productos como al proceso usado para coleccionar información, documentar o analizar algún otro fenómeno físico.

El siguiente diagrama de bloques nos ilustra un Sistema de Adquisición de Datos:

Figura N ° 3

SISTEMA DE ADQUISICION DE DATOS



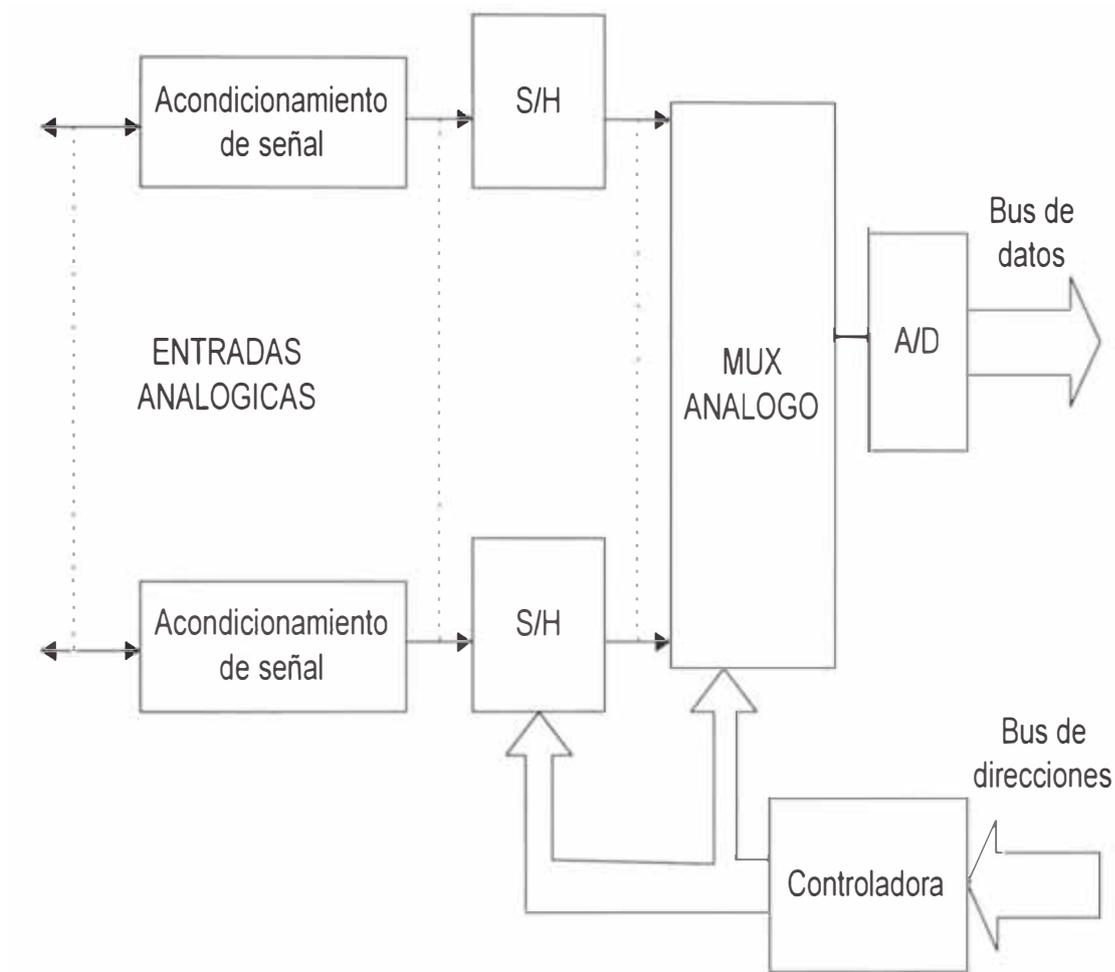
3.2 Hardware de un sistema de adquisición de datos

3.2.1 Arquitectura general

Un sistema de adquisición de datos provee los componentes de interfase entre las señales continuas en el tiempo, como los parámetros de fenómenos físicos y su equivalente digital discreto en el tiempo. En la figura 9 se ve brevemente la arquitectura de estos sistemas

Figura N° 4

ARQUITECTURA DE UN SISTEMA DE ADQUISICIÓN DE DATOS



Citemos a continuación los elementos más usuales que tiene nuestra interfase:

3.2.1.1 Multiplexores analógicos

Si los datos de muchas fuentes deben ser procesados por la misma computadora o canal de comunicación vía un simple conversor, un multiplexor es usualmente empleado para acoplar las señales de entrada

dentro del conversor A/D. Además se usa para poder aprovechar un elemento del computador o de la propia interfase (convertidor A/D) para varios periféricos.

3.2.1.2 Circuitos de muestra y retención (Sample/Hold, S/H)

En muchos sistemas de adquisición la señal analógica varía rápidamente; ya que las conversiones toman lugar en tiempo discreto, y como un conversor A/D no puede digitalizar la señal de entrada instantáneamente, los cambios sustanciales del nivel de la señal durante el proceso de conversión pueden resultar en un gran error.

Los dispositivos Sample/Hold hacen una captura de la señal que está variando y mantienen el valor de la señal capturada en ese instante para realizar el proceso de conversión.

3.2.1.3 Conversores analógico/digital (A/D)

Son conversores de datos analógicos (de voltaje o de corriente) en una forma digital equivalente. Las características principales de estos conversores son: precisión, linealidad, resolución, velocidad de conversión, estabilidad y además el precio.

3.2.1.4 Conversores digital/analógico (D/A)

Los conversores digital/analógico realizan la operación inversa de un conversor A/D, es decir convierte la señal digital en una señal analógica

3.2.1.5 Acondicionamiento de la señal

La información medida por el sensor, debe ser cambiada a una forma apropiada antes de ser enviada al sistema de adquisición de datos, en la mayoría de casos esto significa cambiar la salida de los sensores a un voltaje modificando el rango dinámico de los sensores para maximizar la exactitud del sistema de adquisición de datos, quitando las señales no necesarias y limitando el espectro del sensor. El acondicionamiento puede incluir amplificación, atenuación o filtrado.

3.3 Ventajas de los sistemas de adquisición de datos

Los sistemas de adquisición de datos presentan las siguientes ventajas:

- Recolección de datos del proceso con precisión, periodicidad y confiabilidad adecuadas.
- Supervisión de variables generando alarmas cuando se detectan condiciones anormales

- Almacenamiento de datos históricos para verificación de tendencias y realización de cuadros estadísticos.
- Datos de producción y calculo de rendimiento de productos.

Estas ventajas proporcionan al operador una supervisión mas adecuada del proceso, tomando las acciones de control más conveniente, por otro lado la ingeniería del proceso y producción dispone de un mayor volumen de información, permitiendo conocer mejor el proceso facilitando con ello su mejor control.

Finalmente el operador obtiene información sobre el proceso, dialogando con la computadora a través de sus terminales de vídeo alfanumérico o semigrafico y analizando resultados de cálculos, balances, registro, alarmas o eventos que son obtenidos de los periféricos de impresión.

3.4 Componentes y periféricos para la automatización con la computadora

Para comunicar la PC con el mundo externo es necesario contar con los componentes electrónicos, tarjetas las cuales son insertadas dentro de las PC en localizaciones llamadas slots. Según el tipo de utilidad adquieren una determinada denominación, para el campo de multimedia se tiene la tarjeta de vídeo, tarjeta de sonido, etc.

Para nuestro campo de adquisición de datos y control se utilizan:

- tarjetas A/D
- tarjetas D/A

- tarjetas I/O
- tarjetas de comunicaciones

Además se cuentan con otras tarjetas que son prácticamente una derivación de las tarjetas mencionadas y que sirven para una determinada área.

3.5 Descripción de las tarjetas de adquisición de datos

3.5.1 Tarjetas A/D

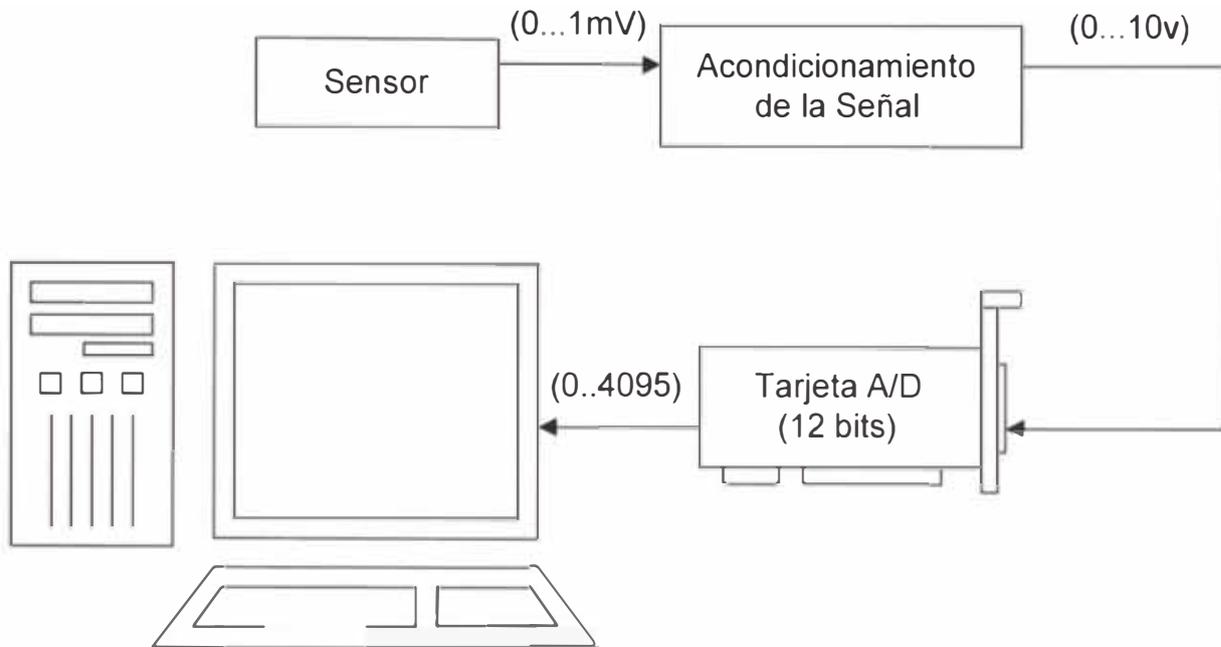
Son tarjetas comerciales desarrolladas para convertir las señales analógicas en señales digitales. Las señales analógicas son proporcionadas por los sensores que miden los diferentes fenómenos físicos del proceso (temperatura, presión, etc.). Estas tarjetas son utilizadas para la adquisición de datos y poseen un convertidor A/D, circuitos de muestreo y almacenamiento, multiplexor, circuitos de sincronización y temporización.

Los rangos de tensión de entrada comúnmente utilizados son 10V, -5V a 5V, 0 a 5V, 0 a 10V. También hay tarjetas A/D que miden corrientes entre 4 a 20 mA.

La figura 5 muestra el circuito completo que se emplea en todo sistema de adquisición de datos empezando por el sensor, acondicionamiento de la señal, tarjeta A/D y la PC.

Figura N° 5

TARJETA A/D



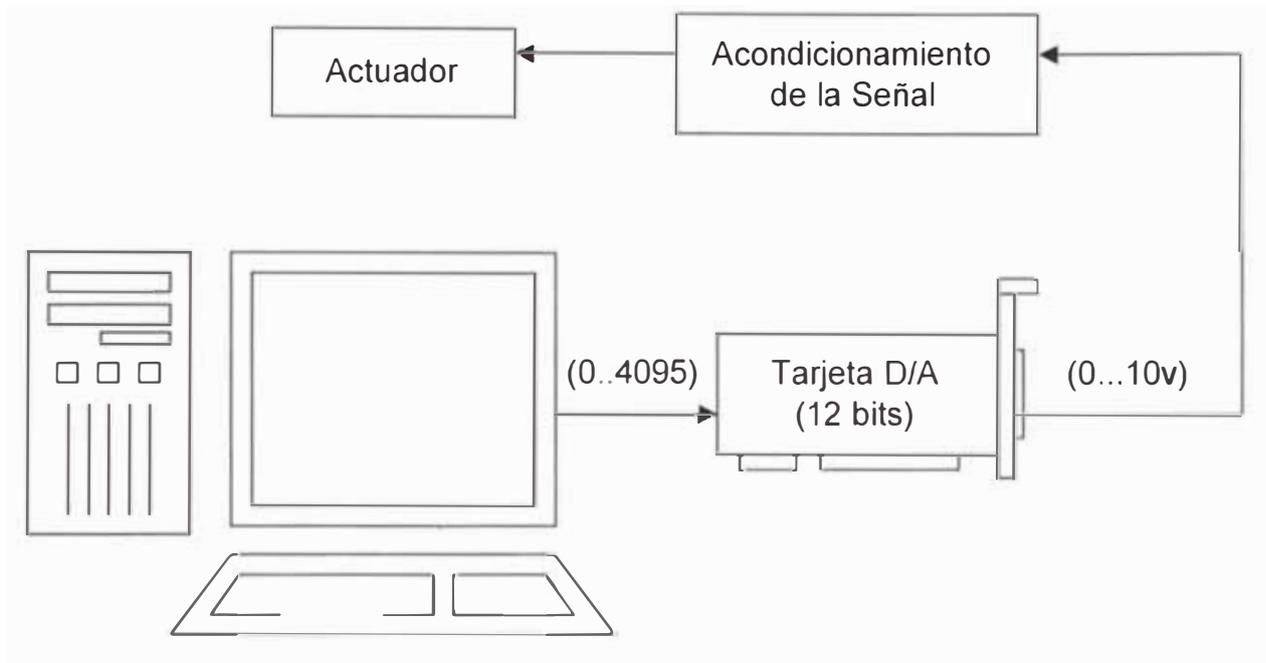
3.5.2 Tarjetas D/A

Llamadas también tarjetas digital-analógicas, están convirtiendo una señal digital dada por la computadora en una señal analógica. Los rangos normales de salida de tensión que otorgan estas tarjetas son de +5V,+10 V, 0 a 5V y de 0 a 10 V. También hay tarjetas que generan corriente de 4 a 20 mA.

La figura 6 muestra el circuito completo que se emplea en todo sistema de adquisición de datos.

Figura N ° 6

TARJETA D/A



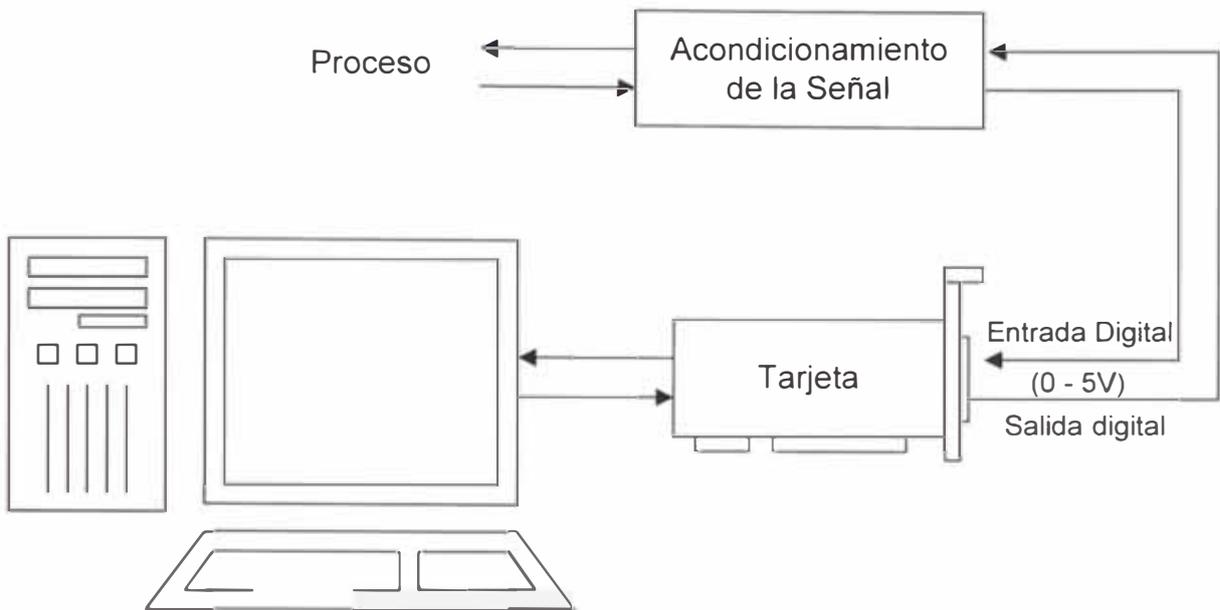
3.5.3 Tarjetas I/O

Llamadas también tarjetas de entradas y salidas digitales(Input/Output). Estas tarjetas miden y generan señales TTL, que poseen solo dos estados de 0 y 5V, a estos estados también se les denomina estados 0 o 1, apagado o prendido, bajo o alto.

Mediante estas tarjetas se pueden accionar todo fenómeno que implique cambio entre estos dos estados, esto se muestra en la figura siguiente.

Figura N° 7

TARJETA I/O



3.6 Costo de un Sistema de Adquisición de Datos

El precio de una tarjeta de adquisición de datos es variable y oscila entre los 300 dólares las más sencillas, hasta 5000 dólares para las de altas prestaciones. Además, casi todas las tarjetas de adquisición de datos ofrecen valores de voltaje y corrientes muy bajos en sus entradas y salidas, tanto analógicas como digitales, lo que hace necesario utilizar algún tipo de adaptador externo para poder conectarle sensores y dispositivos.

Aunque esto puede ser una complicación para la utilización de este tipo de tarjetas, todos los fabricantes ofrecen, junto con la tarjeta, circuitos ya montados que

solucionan estos problemas, y que permiten conectar directamente nuestros dispositivos.

Además estas tarjetas de conexión ofrecen una seguridad añadida, ya que si alguna conexión se realiza de forma incorrecta, impiden que la tarjeta y la computadora se dañen, bien por un cortocircuito o por voltaje y corrientes inadecuados.

Una tarjeta bastante conocida es la LAB PC+ que se utiliza en diversos laboratorios de automatización y control. .

3.7 Implementación del hardware

A continuación se implementara el hardware de adquisición de datos, con componentes electrónicos bastantes sencillos sin necesidad de adquirir una costosa tarjeta de adquisición de las antes ya mencionadas. Se menciona que nuestro hardware simulara a una **tarjeta de adquisición A/D**.

Para la construcción de dicha tarjeta no se necesitan muchos conocimientos de electrónica, sino los conocimientos básicos. Para la implementación del hardware se necesita los siguientes recursos:

Recursos
Humanos

un alumno del Post-grado

Físicos	ambiente de laboratorio
Equipos e Instrumentos	Voltímetro, una computadora.
Materiales	herramientas necesarias, alicates, destornilladores cuchillas, etc. software de tiempo real, linux y rt-linux componentes electrónicos: cable de 25 pines para la comunicación en paralelo convertor Digital/Analógico(D/A) 0800, 8 bits transistor de potencia sensor de intensidad de luz(fotocelda) convertor Analógico/Digital(A/D) 0804, 8 bits multiplexor digital 74157 2 a 1, cuádruple lampara 24 V y 40 W amplificadores operacionales acondicionadores de señal transformador de 24 V y 3A transformador de 15V con toma central de 1A puente rectificador de onda completa condensadores resistencias eléctricas leds potenciómetro(resistencia variable)

Con todos estos recursos se construye nuestra pequeña tarjeta de adquisición de datos, habiendo especificando al técnico que nos ayudo en construir el hardware, lo que necesitamos obtener para el trabajo y su aplicación correcta, mencionando que nuestra tarjeta de adquisición tomara una señal eléctrica, hacia el computador para procesarla.

Para tal efecto se hace necesaria una conversión de variables analógicas en variables digitale para ser manipuladas a través del software, esta tarjeta de adquisición se mostrara en la experiencia. Se menciona que no es muy difícil de manipularla por lo que se puede conectar directamente al puerto paralelo de la impresora y obtener la recolección de datos.

Se muestra a continuación el diagrama de dicha tarjeta.

FOTO DEL HARDWARE CONECTADO A LA COMPUTADORA

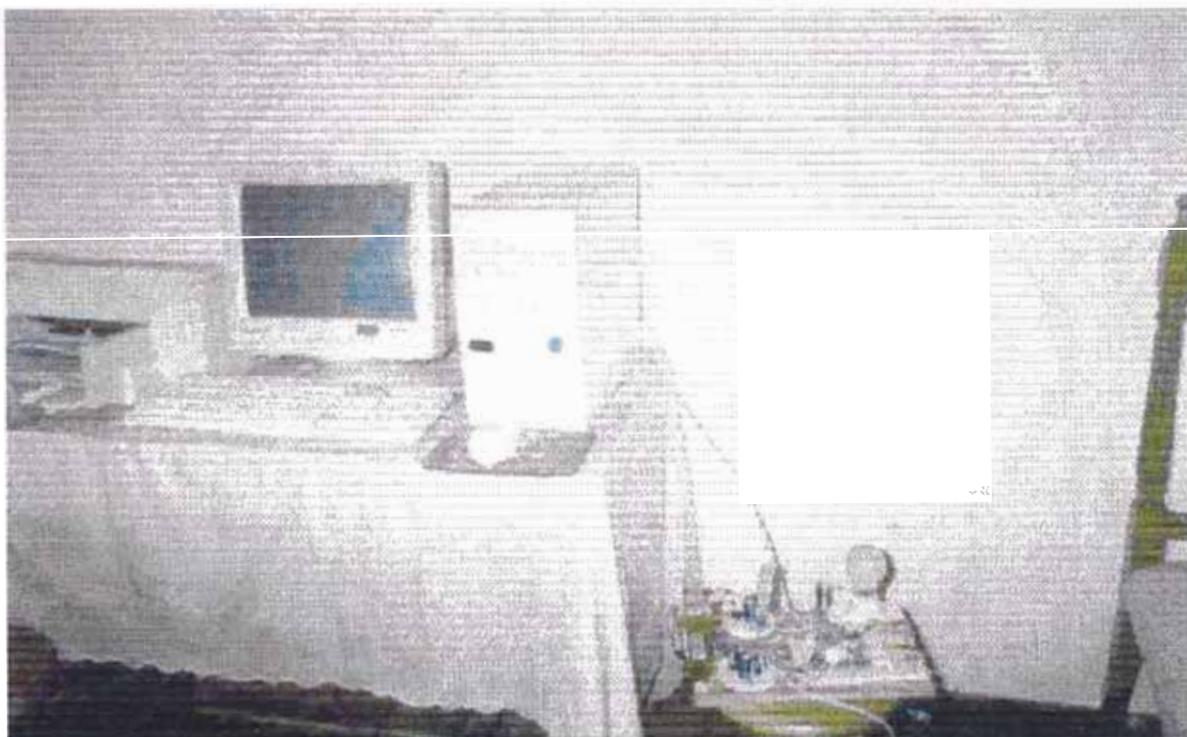
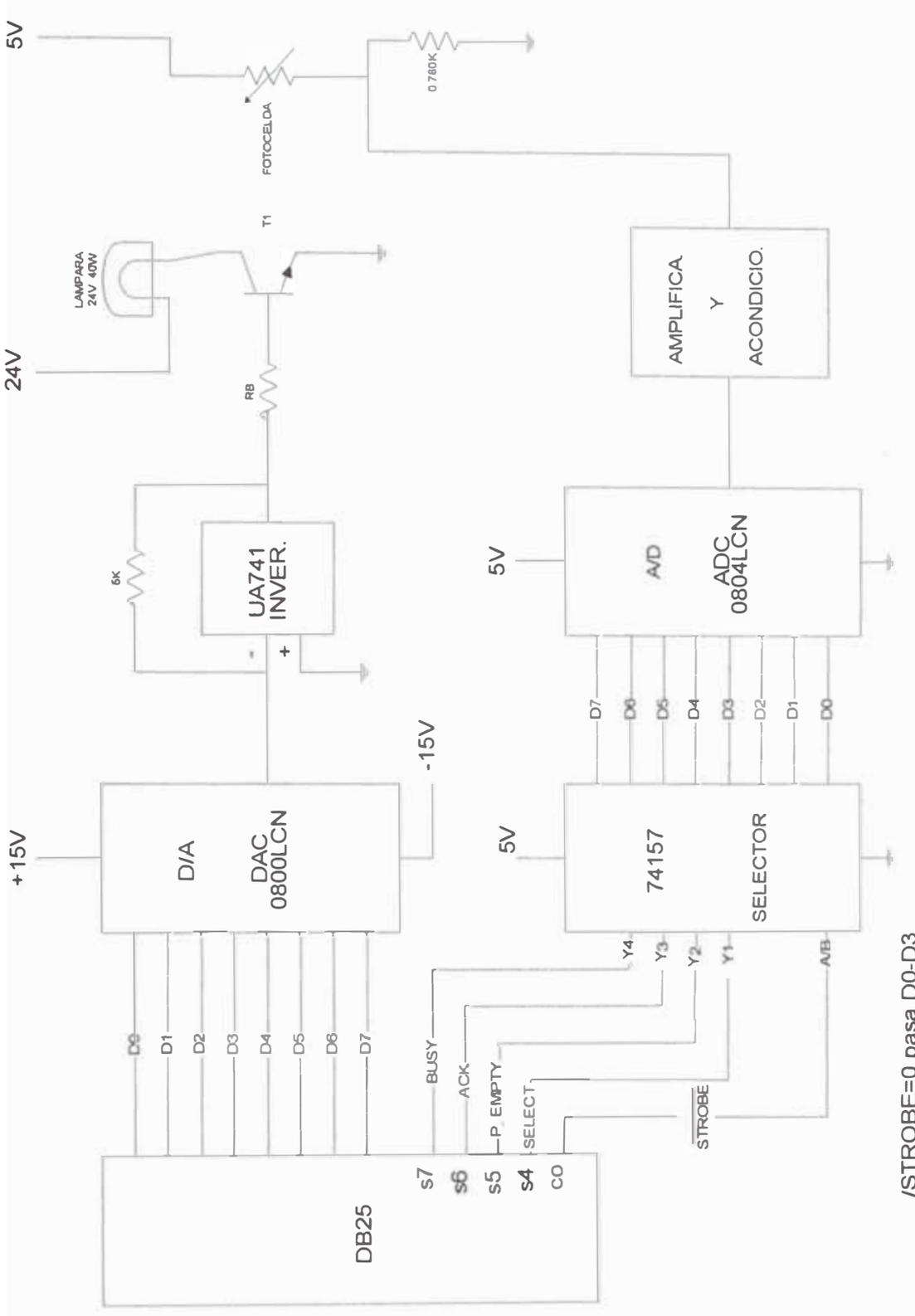


FIG. 8 HADWARE DE ADQUISICION DE DATOS



/STROBE=0 pasa D0-D3
/STROBE=1 pasa D4-D7

CAPITULO IV

IMPLEMENTACION DE PROGRAMAS

Para la implementación de los programas se ha escogido para trabajar con el sistema operativo Linux, que se ha reforzado con el RT-Linux para suministrar capacidades de tiempo real. Inicialmente se tuvieron algunos problemas en la creación de los programas, mas que nada esto era debido a las diferentes versiones del kernel de RT-Linux, también se tuvo ciertos problemas en hallar las diferentes direcciones de los puertos, ya que no todas las computadoras tienen asignadas las mismas direcciones. En forma complementaria se implemento un pequeño programa para hallar dichas direcciones en Linux.

Tal como se explica en los anexos, la implementación de los programas presenta las siguientes etapas: creación del modulo en RT-Linux, creación del programa de usuario, creación del makefile correspondiente.

A continuación se plantean los siguientes programas:

programa 1: creación del modulo en RT-LINUX

programa 2: creación del programa de usuario para obtener los datos sensados

programa 3: creación del makefile correspondiente

programa 4: creación de un programa para hallar las direcciones de los puertos.

4.1 Programa 1: Creación del modulo de tiempo real

```
/* Este modulo esta en tiempo real para tomar los datos y enviarlo por el puerto paralelo de la impresora */
```

```
#include <linux/config.h>
```

```
#include <linux/module.h>
```

```
#include <linux/kernel.h>
```

```
#include <linux/version.h>
```

```
#include <linux/errno.h>
```

```
#include <linux/ioport.h>
```

```
#include <asm/io.h>
```

```
#include <asm/segment.h>
```

```
#include <asm/system.h>
```

```
#include <rtl sched.h>
```

```
#include <rtl fifo.h>
```

```
#define SIZE 8192
```

```

#define SALIDA 0x378 // Definición de puertos

#define ENTRADA 0x379

#define SELECT 0x37A

unsigned int entrada_data(void);

pthread_t rt_task;

//-----Tarea de tiempo real-----

void *Real_Time_Task(void *tag){

char data;

while(1) {

pthread_wait_np();

data = entrada_data(); // tomar datos de entrada

rtf_put(1, &data, sizeof(data)); // coloca los datos en el fifo l

}

}

// ----- Programa Principal-----

int init_module(void) { // inicializa el modulo

int fifo_status;

int fifo_size = 4000;

struct sched_param p;

hrtime_t now = gethrtime(); // obtiene el tiempo del sistema

```

```

rtf_destroy(1);

fifo_status = rtf_create(1, fifo_size);           // crea fifo1

if (fifo_status) {

rtl_printf("No se pudo crear FIFO.fifo_status=%d\n", fifo_status);

return -1;

}

pthread_create(&rt_task, NULL, Real_Time_Task, void *)1);    // crea la tarea

pthread_make_periodic_np(rt_task, now + 2* NSECS_PER_SEC,31230000);

pthread_setfp_np(rt_task, 1);                       // hace
                                                    periódica

p.sched_priority = 1;

pthread_setschedparam (rt_task, SCHED_FIFO, &p);

return 0;

}

//-----Para descargar modulo-----

void cleanup_module(void) {                          // desinstala el modulo

pthread_delete_np(rt_task);                          // destruye la tarea

rtf_destroy(1);                                       // destruye el fifo1

}

// -----Adquisición de datos-----

unsigned int entrada_data(void)

{

```

```

char entrada, auxiliar;

outb_p(0, SELECT);           // envío datos por el puerto select

auxiliar = inb_p(ENTRADA);   // para pasar los 4 bits menos significativos
auxiliar ^= 0x80;           // se invierte el bit más significativo

entrada = (auxiliar>>4) & 0x0F; // se traslada a la posición menos
                                // significativa

outb_p(0xFF, SELECT);

auxiliar = inb_p(ENTRADA);   // para pasar los 4 bits más significativos
auxiliar ^= 0x80;           // se invierte el bit más significativo

auxiliar = auxiliar & 0xF0;

entrada = entrada | auxiliar; // se agrupan los 8 bits correspondientes a la
                                // entrada

return entrada;

}

```

4.2 Programa 2: Creación del programa de usuario

```

#include <unistd.h>

#include <fcntl.h>

#include <sys/io.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <stdio.h>

#include <linux/ioport.h>

#include <asm/io.h>

```

```

#include <asm/segment.h>

#include <asm/system.h>

#include <math.h>

int main() {

    int fd;

    unsigned char data;

    int cont=0;

    if ((fd = open("/dev/rtf1", O_RDONLY)) < 0) {        // abro el file descriptor

        fprintf(stderr, "Error abriendo /dev/rtf1\n");

        exit(1);

    }

    printf("Iniciando lectura de FIFO...\n");

    ioperm(0x2F8, 1, 0x2F9);                            // permiso al kernel

    while(1) {

        read(fd, &data, sizeof(data));                    // leo el valor enviado por

                                                            data

        printf("Valor de voltaje leído es %f\n", (float)data*0.0196);

        cont++;

    }

    close(fd);                                            //cierro él file descriptor

    return 0;

}

```

4.3 Programa 3: Creación del makefile

```
all: treall.o usuario
```

```
include rtl.mk
```

```
MYCFLAGS = -O2 -Wall
```

```
usuario: usuarl.c
```

```
$(CC) ${INCLUDE} -O2 -Wall usuarl.c -o usuario
```

```
treall.o: treall.c
```

```
$(CC) ${INCLUDE} -O2 ${CFLAGS} -c treall.c
```

```
#test, prueba el modulo real-time y ejecuta la aplicación usuario
```

```
test: all
```

```
@echo "Presione <return> para continuar"
```

```
@read junk
```

```
(cd /usr/src/rtlinux-2.0/rtl; ./insrtl)
```

```
@echo "Instalando modulo real-time"
```

```
@echo "Presione <return> para continuar"
```

```
@read junk
```

```
@insmod treall.o
```

```
@echo "Iniciando la aplicación"
```

```
@echo "Presione <return> para continuar"
```

```
@read junk
```

```
@./usuario
```

```
clean:
```

```
rm -f usuario *.o
```

```
#Automatically generated by rtl Makefile
```

```
RTL_DIR = /usr/src/rtlinux-2.0/rtl
```

```
RTLINUX_DIR = /usr/src/rtlinux-2.0/linux
```

```
INCLUDE= -I/usr/src/rtlinux-2.0/linux/include -I/usr/src/rtlinux-2.0/rtl/include -  
I/usr/src/rtlinux-2.0/rtl
```

```
CFLAGS = -I/usr/src/rtlinux-2.0/linux/include -I/usr/src/rtlinux-2.0/rtl/include -  
I/usr/src/rtlinux-2.0/rtl -I/usr/src/rtlinux-2.0/rtl/include/posix -Wall -Wstrict-
```

```
prototypes -O2 -fomit-frame-pointer -D__SMP__ -D__RTL__ -D__KERNEL__ -
```

```
DMODULE -pipe -fno-strength-reduce -m486 -malign-loops=2 -malign-jumps=2 -  
malign-functions=2 -DCPU=686
```

```
ARCH = i386
```

```
CC = gcc
```

```
RTL_MODULES=/lib/modules/2.2.13-rtl2.0/misc
```

4.4 Programa 4: Búsqueda de direcciones puerto serial y paralelo

```
#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

#include<unistd.h>

int main(void)

{

    int fd,fd1,i,j;

    short int lp[3],lp1[4];

    if((fd=open("/dev/mem",O_RDONLY))==-1){

        printf("No puede abrir /dev...\n");

        exit(1);

    }

    if((fd1=open("/dev/mem", O_RDONLY))==-1){

        printf("No puede abrir /dev...\n");

        exit (2);

    }

    lseek(fd,0x408,SEEK_SET);

    lseek(fd1,0x400,SEEK_SET);

    for(i=0;i<3;i++)

        read(fd,&lp[i],2);

    printf("\nDireccion del Puerto Paralelo:

    lp1=%X,lp2=%X,lp3=%X\n",lp[0],lp[1],lp[2]);
```

```

close(fd);

for (j=0;j<4;j++)

    read(fd1,&lp1[j],2);

    printf("\n\nDireccion del Puerto serial:

    lp1=%X,lp2=%X,lp3=%X,lp4=%X\n\n\n",lp1[0],lp1[1],lp1[2],lp1[3]),

close(fd1);

exit(0);

}

```

Una vez que están instalados los programas se siguen los siguientes pasos para ejecutarlos en RT-LINUX desde los respectivos directorios:

- 1) root] # cd /usr/src/rtlinux-2.0/rtl/
- 2) /rtl] # ./insrtl
- 3) /rtl] # lsmod
- 4) /jimmy] # insmod treall.o para instalarlo
- 5) /jimmy] # ./usuario compilo y ejecuto el programa de usuario
- 6) /jimmy] # rmmod treall para desinstalarlo

4.5 Breve explicación de los programas

La forma más sencilla de medir la intensidad de la luz a través del puerto paralelo es crear una tarea periódica que se encargue de leer dichos valores. Estos valores se almacenan en el FIFO, un proceso Linux tiene que estar esperando a leer de esa FIFO. Los valores que lee son alternativamente la intensidad o voltaje del foco de luz

dependiendo del usuario(en nuestro caso se escogió voltaje y se hizo la conversión de escala).

Realizar una buena medida de la señal entregada por el sensor de luz es fundamental, de hecho, este es el único motivo que justifica el uso de tiempo real. Interpretar la secuencia de valores que entrega la tarea de tiempo real y para posteriormente analizarla.

El valor del periodo dado se ha obtenido por la experiencia para tener un mejor resultado.

En cuanto al programa de usuario este imprime por la salida estándar normal lo que lee del dispositivo dev/rtf1. Con este simple programa podemos ver los valores de voltaje o intensidad emitidos. Además si en algún caso se quisiera hacer una comunicación serial con otra PC introducimos en dicho programa el permiso del Kernel y luego lo enviamos por dicho puerto.

En la otra PC se obtendría dicho valor por medio de un programa de usuario, previamente instalado el modulo del driver. Se recalca que los programas de usuario de recepción de datos por el puerto paralelo y el de lectura para el puerto serial podrían ser creados en ambiente Linux, no siendo necesario implementarlo en RT-Linux .

A continuación hacemos una presentación esquemática de lo que hace cada programa:

a) El programa 1 esta constituido por:

.Tarea de tiempo real

-Implementa la funcion Real_Time_Task

-Acá se crea un lazo infinito donde toma los datos de entrada invocando la función data.

-Coloca los datos en el fifo1.

-Suspende la ejecución de la llamada de la tarea hasta el comienzo del próximo periodo.

.Programa principal

-Acá sé inicializa él modulo.

-Obtiene el tiempo del sistema.

-Crea el fifo por donde se comunicara la tarea.

-Crea la tarea.

-Hace periódica la tarea.

.Descarga de Modulo

-Acá descargamos el modulo

-Destruimos los fifos creados.

.Adquisición de datos

Se adquiere los datos de usando el puerto select y el de entrada, se menciona que los bits van enviando en dos formas, primero los primeros 4 bits menos significativos y después los 4 bits más significativos. Finalmente se agrupan todos los bits.

b) En el programa 2 que es el programa de usuario se hace lo siguiente:

-Se crea la función main

-Se abre un file descriptor para lectura

-Se pide permiso al kernel para lectura

-Se crea un lazo infinito donde se va leyendo los valores ya sensados y son mostrados en pantalla.

-Finalmente se cierra el file descriptor.

c) En el programa 3 se hace lo siguiente :

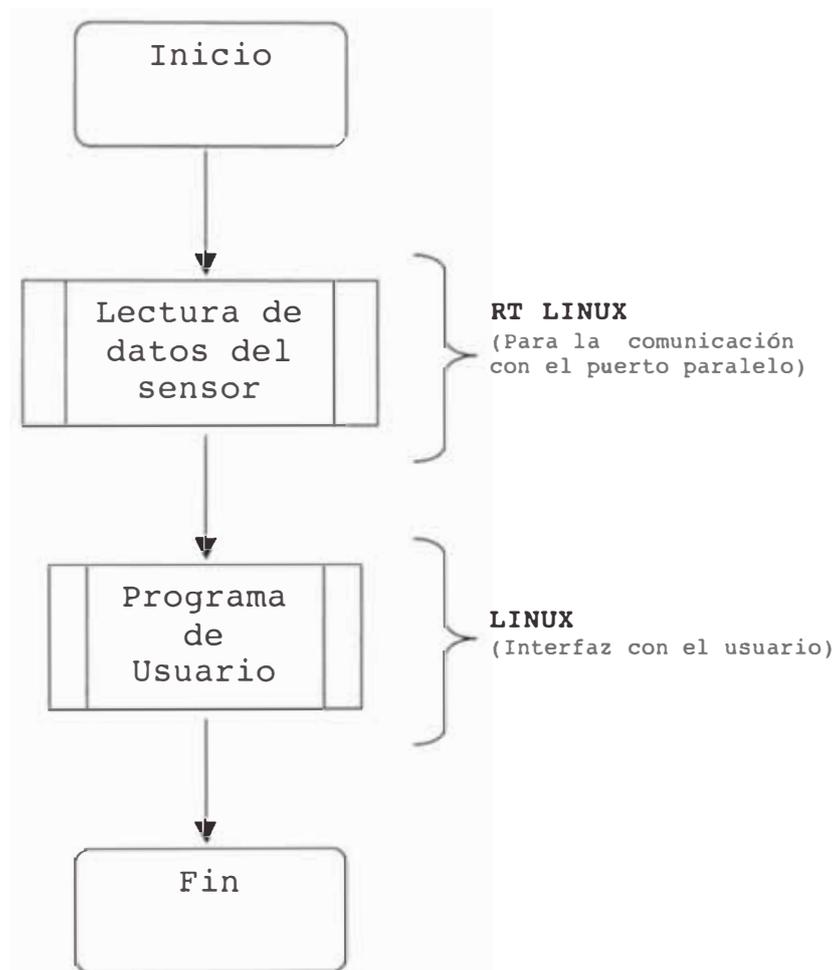
Se crean los denominados Makefile que vienen a ser pequeños programas de ayuda para la compilación y ejecución de los programas, si no se crean estos la compilación sería dificultosa. Generalmente en la compilación de programas en RT-Linux y Linux se hace uso de ellos.

d) En el programa 4 se hace lo siguiente:

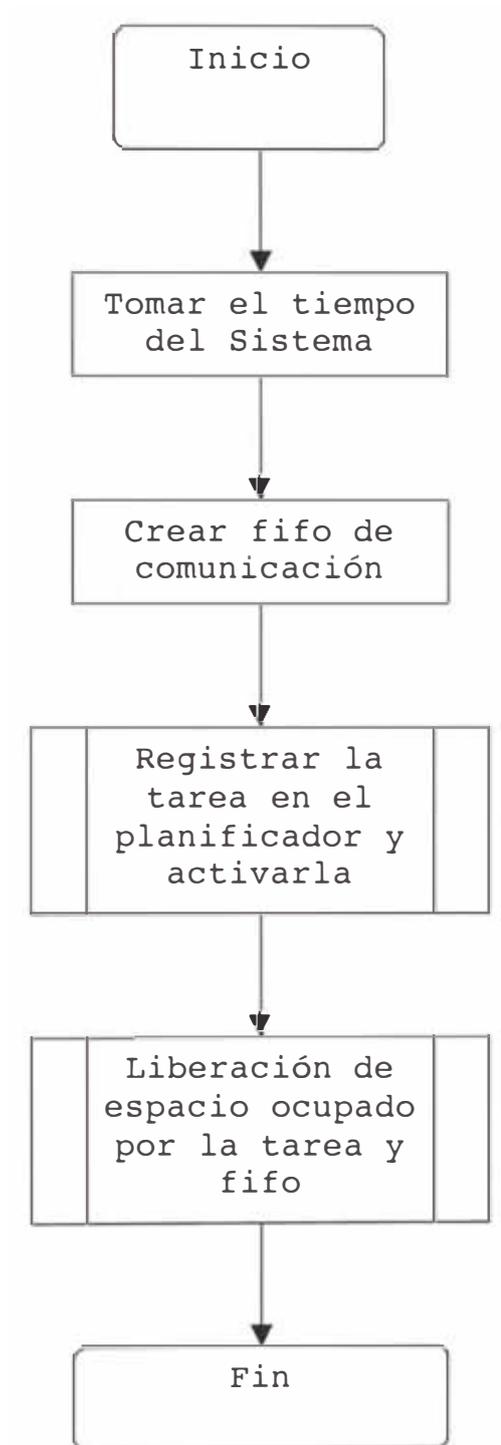
Este es un programa opcional, que se crea con el fin de hallar la dirección de los puertos paralelos y seriales de la PC ya que como es sabido no todas las computadoras tienen las mismas direcciones. Además como se menciona en el programa 1, se tiene que definir dentro del modulo de RT dicha dirección, porque sino origina problemas en la conexión con el hardware y en el desarrollo de las diversas tareas a hacer.

4.6 Diagrama de Procesos

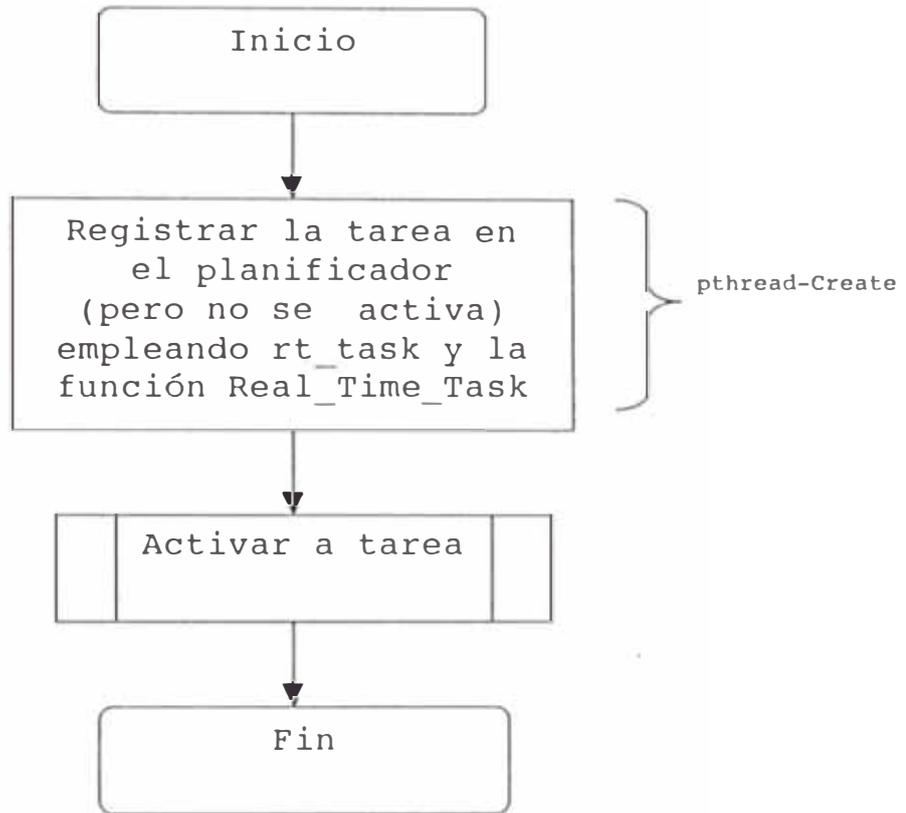
A continuación mostraremos los diagramas de los diferentes procesos, para un mejor entendimiento de los programas anteriores.



LECTURA DE DATOS DEL SENSOR



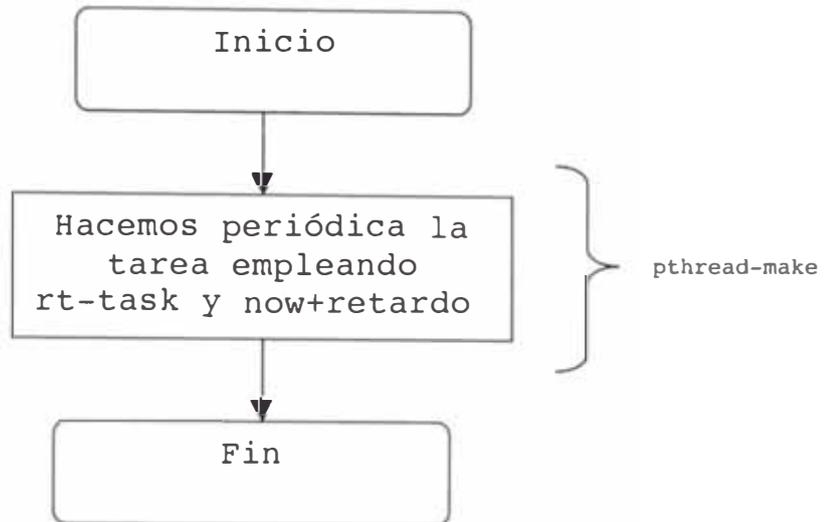
REGISTRAR LA TAREA Y ACTIVARLA



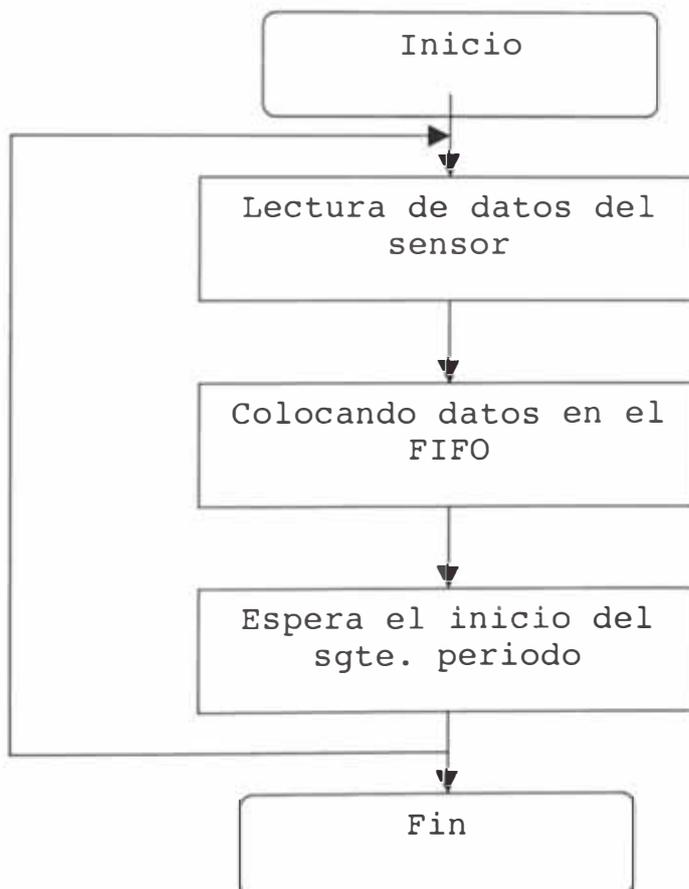
LIBERACIÓN DEL ESPACIO OCUPADO POR LA TAREA



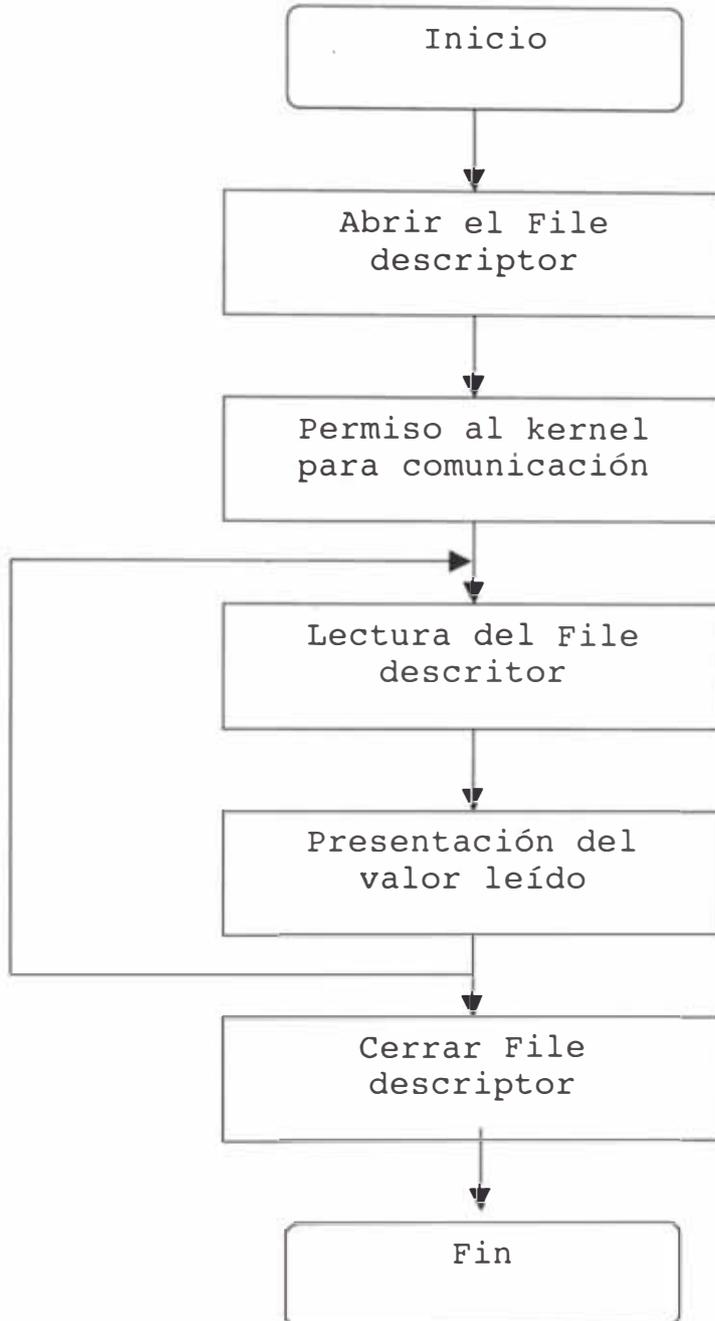
ACTIVAR LA TAREA



FUNCIÓN REAL-TIME-TASK



PROGRAMA DE USUARIO



CAPITULO V

PARTE EXPERIMENTAL

Luego de la descripción de los diversos equipos a utilizar en el trabajo en los anteriores capítulos se menciona la metodología que se usara en el presente trabajo.

5.1 Metodología a usar

Como ya se menciona anteriormente para la implementación del sistema de tiempo real, debe integrarse hardware, software y personal para conseguir el adecuado funcionamiento de este sistema.

La metodología a usar es bastante sencilla, una vez instalado el hardware y conectado con la computadora, por el puerto paralelo de la impresora recibiremos los valores sensados de la luz en tiempo real, para lograr dicha integración necesitaremos los siguientes equipos:

5.1.1 El sensor de luz

En nuestro caso se utiliza un sensor de medida que se encuentra con facilidad en el mercado. Se trabajara con un sensor de luz que esta construido para

detectar la presencia de cualquier luz, independientemente de la fuente que la genere, la luz emitida es detectada por un dispositivo semiconductor que se colocara en el hardware. Dicha luz será graduada en su intensidad con un control manual.

5.1.2 El conector del puerto paralelo

En nuestro caso el puerto paralelo de la PC utilizara un conector hembra de tipo D de 25 pines, habiéndose hecho las conexiones de todos los pines con la tarjeta de adquisición de datos.

5.1.3 El hardware de adquisición de datos

Como ya se menciona anteriormente esta se diseño con algunas características especiales y además se mostró su respectivo diagrama en la Figura N° 8.

5.1.4 Detalles de los registros de estado

Como ya se menciona anteriormente el puerto paralelo tiene 12 líneas de salida(8 líneas de datos, strobe, autofeed, init y select input) y 5 de entrada(acknowledge, busy, falta de papel, select y error).

Hay tres direcciones de E/S asociadas con el puerto paralelo de la PC, estas direcciones pertenecen al registro de datos (0x378 o 0x3BC), al de estado

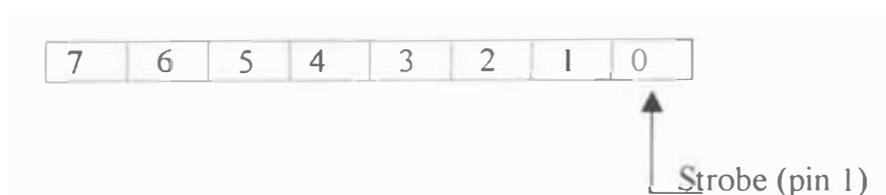
(0x379) y al de control (0x37A). Los valores entre paréntesis representan las direcciones usuales de los registros del primer puerto paralelo.

a) Registro de control

Escribiendo en el registro de control se puede programar el funcionamiento del puerto, este puerto será utilizado como de salida. De los 8 bits del registro, sólo nos interesara en este caso un bit

Este bit 0(Strobe) pin 1 permitirá el paso de los bits menos significativos y luego los más significativos, mediante la condición de que sea uno o cero (como se muestra en el diagrama de la tarjeta de adquisición).

Bits usados



b) Registro de datos

El registro de datos es una puerta directa a los pines del conector. Cualquier dato que escribamos en ese puerto, aparecerá en los pines de 2 a 9 del conector paralelo. Este puerto es sólo de salida, por lo que no podemos saber el valor ("0" o "1") de los pines del conector leyendo el puerto. Puesto que nosotros

tenemos que leer los valores que el receptor de luz entregue, este puerto no será utilizado

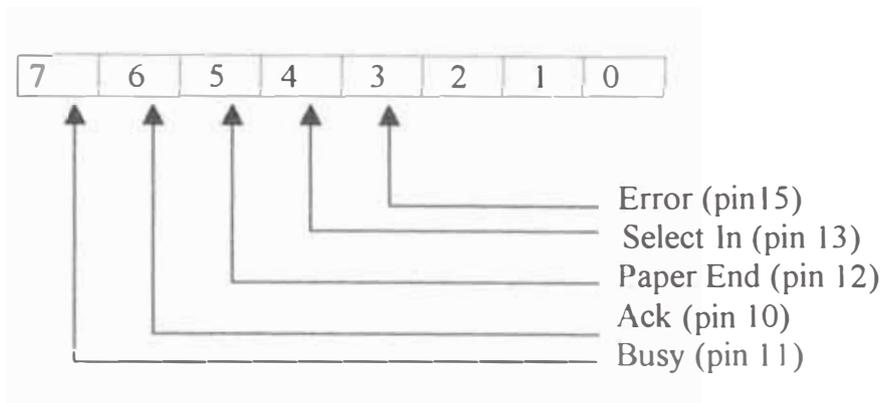
c) Registro de estado

El puerto de estado refleja el estado de las líneas de control con la impresora. Mediante estas líneas la impresora es capaz de informar al ordenador sobre el estado en el que encuentra. Son líneas que gestiona la impresora (o dispositivo externo) y que lee el ordenador, por tanto son líneas de entrada aunque el registro es de 8 bits, solo los 5 bits de mayor peso están conectados a las líneas, los otros tres bits están siempre a cero. Los nombres de las líneas tienen su origen en la función que desempeñan cuando se utilizan para comunicarse con una impresora, pero realmente estas líneas son unas entradas digitales genéricas que se pueden utilizar para casi cualquier cosa (esto se muestra en el diagrama de la tarjeta).

De estas 5 líneas las más usadas son: el bit 6 (ACK) pin 10 del registro, pues además de servir como línea de entrada digital, también se pueden producir interrupciones cuando la línea cambia de estado (de 5 a 0), tal como se mencionaba.

Los otros bits usados son: bit 7(Busy) pin 11, bit 5(Falta de papel) pin12 y finalmente el bit 4 (Select In) pin 4.

Bits usados



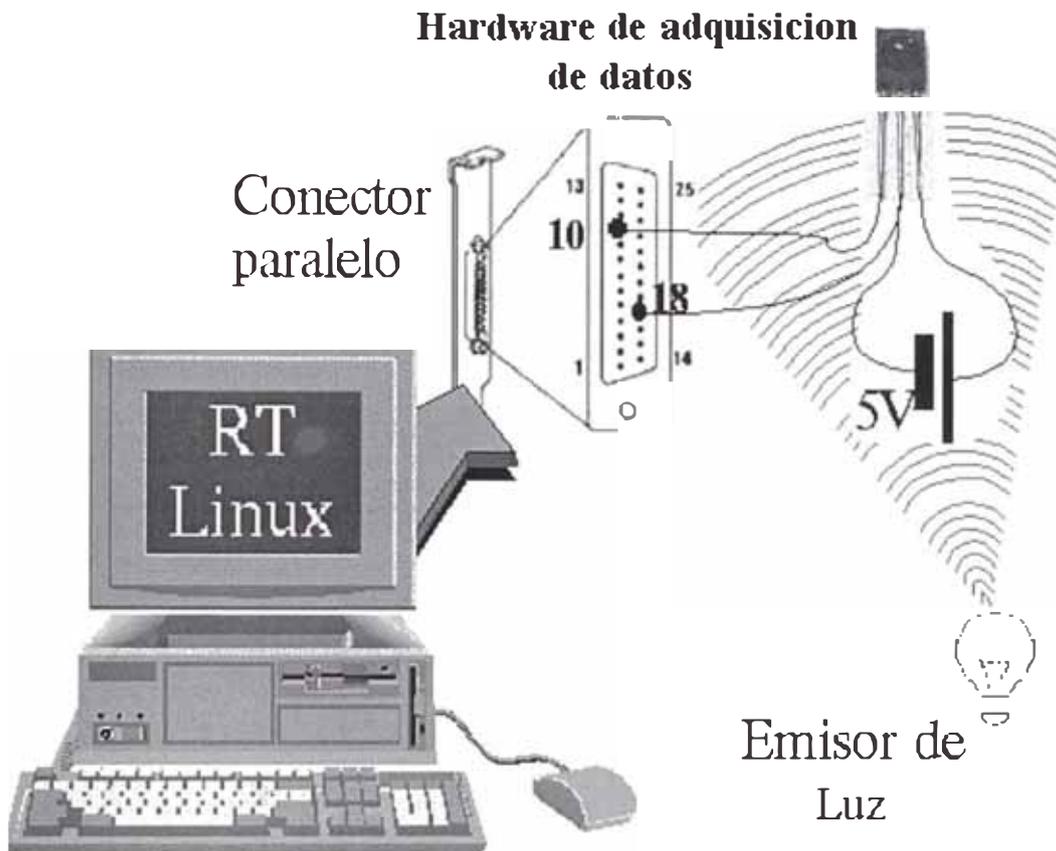
5.1.5 Conexión con la computadora

La conexión del sensor con la PC no es tan complicada, esto se ha realizado con los equipos ya descritos tal como muestra en la siguiente figura. El receptor de luz necesita una tensión de alimentación entre 0 y 5 voltios, si la tensión baja a cero voltios, el sensor deja de funcionar. Esta tensión se puede conseguir de varias formas. Una de ellas es utilizando una fuente de alimentación que es un aparato que convierte tensión continua de bajo voltaje.

En cuanto al hardware de adquisición de datos, se menciona que cuando el sensor envía datos, primero pasan los bits menos significativos y luego los más significativos esto se hace en una función que se invoca en el programa, también se recalca que esto se logra trabajando con el bit 0 (Strobe) del puerto de control quien solo toma dos valores para permitir el paso de dichos bits. Se comenta que se podría usar distintas formas de enviar los bits eso dependería de cómo se habría diseñado la tarjeta y de cómo se haría el programa de adquisición de datos.

Figura N ° 9

VISION GLOBAL DEL SISTEMA Y CONEXIÓN ELECTRICA



5.1.6 Resultados obtenidos

Luego de tener conectada nuestra pequeña tarjeta de adquisición de datos con la computadora se procede a sensar la luz y hacer nuestras primeras pruebas experimentales con el puerto paralelo obteniéndose los valores deseados en el ordenador, estas pruebas pueden ser fácilmente mostradas e implementadas en cualquier computador.

Además todas las pruebas se realizaron con una computadora Pentium II de 400 Mhz con 30 Mbytes de memoria RAM, equipada con un disco fijo. Respecto al sistema operativo, se utilizo la versión 2.2.13.0 del kernel de linux parchado con la versión 2,0 de RT-Linux.

Uno de los problemas que se tuvo fue que en el computador se obtenian los bits que el sensor enviaba, esto no era lo que queríamos obtener. Nosotros queremos obtener el voltaje verdadero, luego se hicieron tomas de voltaje en diversos tiempos como:

VALOR DIGITAL	VOLTAJE (voltios)
0	0
128	2.5
255	5.0

Con estos valores se observa que hay una relación lineal entre el voltaje medido y el valor digital con lo cual se encontró el factor de conversión para nuestro trabajo.

Luego con esto realizamos el escalamiento, obteniéndose la siguiente expresión que nos relaciona voltaje real con el valor digital mostrado en el computador:

$$V \text{ (voltaje real)} = 0.0196 B \text{ (valor digital leído en el computador)}$$

Se debe tener en cuenta que esta expresión se coloca dentro del programa de usuario para tener el valor del voltaje, por ejemplo se tiene algunos valores sensados y se leerían en la computadora como:

```
/jimmy]# Valor del voltaje leído es 0.000  
Valor del voltaje leído es 0.102  
Valor del voltaje leído es 0.234  
Valor del voltaje leído es 0.399  
Valor del voltaje leído es 0.500  
Valor del voltaje leído es 0.686  
Valor del voltaje leído es 0.723  
Valor del voltaje leído es 1.232  
Valor del voltaje leído es 1.960  
Valor del voltaje leído es 1.990  
Valor del voltaje leído es 2.940  
Valor del voltaje leído es 2.960  
Valor del voltaje leído es 2.990  
Valor del voltaje leído es 3.240
```

Valor del voltaje leído es 3.340

Valor del voltaje leído es 3.450

Valor del voltaje leído es 3.550

Valor del voltaje leído es 3.660

Valor del voltaje leído es 3.740

Valor del voltaje leído es 3.820

Valor del voltaje leído es 3.920

Valor del voltaje leído es 4.410

Valor del voltaje leído es 4.999

Luego de obtener los siguientes valores sensados algunos de estos valores pueden ser recopilados en la siguiente tabla:

Valor Digital	Voltaje
	0.000
25	0.500
35	0.686
45	0.882
100	1.960
150	2.940
200	3.920
225	4.410
250	5.000

CAPITULO VI

APLICACIÓN EN EL ANALISIS INSTRUMENTAL

Una de las aplicaciones del presente trabajo de los sistemas de tiempo real es en el campo del análisis químico instrumental, para ello mencionaremos las siguientes definiciones que son importantes de conocer.

Introducción

La Química analítica trata acerca de los métodos de determinación de la composición química de la materia. Un método cualitativo proporciona información respecto a las especies atómicas o moleculares que existen en una muestra; un método cuantitativo, suministra información numérica de la cantidad relativa de uno o varios componentes.

6.1 Clasificación de los métodos analíticos

Los métodos analíticos para un mejor estudio se suelen clasificar en clásicos e instrumentales:

6.1.1 Métodos clásicos

En los primeros años de la química, la mayor parte de los análisis se realizaban separando los componentes de interés de la muestra mediante técnicas de precipitación, extracción, o destilación. En los análisis cualitativos, los componentes separados se trataban seguidamente con reactivos originando así productos que podían identificarse por sus colores, puntos de ebullición, fusión, sus olores, sus solubilidades en una serie de disolventes, sus actividades ópticas o sus índices de refracción.

En los análisis cuantitativos se determinaban por medidas gravimétricas o volumétricas. En la primera se determinaba la masa de la muestra o la de algún compuesto producido a partir del mismo. En los procedimientos volumétricos se determinaba el volumen o el peso de un reactivo estándar que reaccionase completamente con la muestra.

Estos métodos clásicos para separación y determinación de muestras todavía se usan en muchos laboratorios. Sin embargo, su grado de aplicación esta disminuyendo con el paso del tiempo.

6.1.2 Métodos instrumentales

A mediados de los años 30, los químicos comenzaron a explotar otros fenómenos distintos de los descritos anteriormente para la resolución de

problemas analíticos. Así para el análisis cuantitativo de una gran variedad de muestras se empezaron a utilizar mediciones de las propiedades físicas de las muestras, tales como la conductividad, potencial de electrodo, absorción o emisión de luz, fluorescencia, razón masa o carga. Además, algunas técnicas de separación cromatográficas muy eficaces empezaron a reemplazar a la destilación, extracción y precipitación en la separación de mezclas complejas como etapa previa a su determinación cualitativa o cuantitativa.

A estos métodos más modernos para separar y determinar especies químicas se les conoce en conjunto como métodos instrumentales de análisis.

Muchos de los fenómenos en los que se basan los métodos instrumentales se conocen desde hace más de un siglo, sin embargo su aplicación por la mayor parte de los químicos se retrasó por falta de instrumentación sencilla y fiable. De hecho el crecimiento de los métodos instrumentales modernos ha ido paralelo al desarrollo de las industrias electrónica e informática.

6.1.2.1 Tipos de métodos instrumentales

Para este estudio, es conveniente describir propiedades físicas que pueden utilizarse como señales analíticas en el análisis cualitativo o cuantitativo. La tabla 5 enumera la mayoría de señales analíticas que suelen usarse en el análisis instrumental.

Obsérvese que las seis primeras están relacionadas con la radiación electromagnética, las cuatro siguientes son eléctricas. Por último cuatro señales diversas se agrupan conjuntamente y vienen a ser: la razón masa a carga, la velocidad de reacción, las señales térmicas y la radioactividad.

También se observa que en la segunda columna de dicha tabla se indica los nombres de los métodos instrumentales basados en las distintas señales analíticas.

Debería entenderse que existen pocas diferencias que distinguen a los métodos instrumentales de sus equivalentes clásicos. Algunas técnicas instrumentales son más sensibles que las técnicas clásicas, pero otras no.

Tabla N ° 5

LOS METODOS INSTRUMENTALES

Señal	Métodos Instrumentales
Emisión de radiación electrones,	Espectroscopia de emisión (rayos X, UV, visible, de Auger, flouescencia, fosforescencia y luminiscencia (rayos X, UV y visible)
Absorción de radiación	Espectrofotometría y fotometría (rayos X, UV, visible, IR); espectroscopia fotoacústica; resonancia magnética nuclear y espectroscopia de resonancia de espín electrónico.
Dispersión de la radiación	Turbidimetría; nefelometría; espectroscopia Raman.
Refracción de la radiación	Refractometría; interferometría
Difracción de la radiación	Métodos de difracción de rayos X y de electrones
Rotación de la radiación	Polarimetría; dispersión rotatoria óptica; dicroismo circular
Potencial eléctrico	Potenciometría; cro nopotenciometría
Carga Eléctrica	Coulombimetría
Corriente eléctrica	Polarografía; amperometría
Resistencia eléctrica	Conductimetría
Razón masa a carga	Espectrometría de masas
Velocidad de reacción	Métodos cinéticos
Propiedades térmicas	Conductividad térmica y métodos de entalpía
Radiactividad	Métodos de activación y de dilución isotópica

También es bueno mencionar que no necesariamente los procedimientos instrumentales utilizan aparatos mas sofisticados o más costosos que los otros métodos ya mencionados.

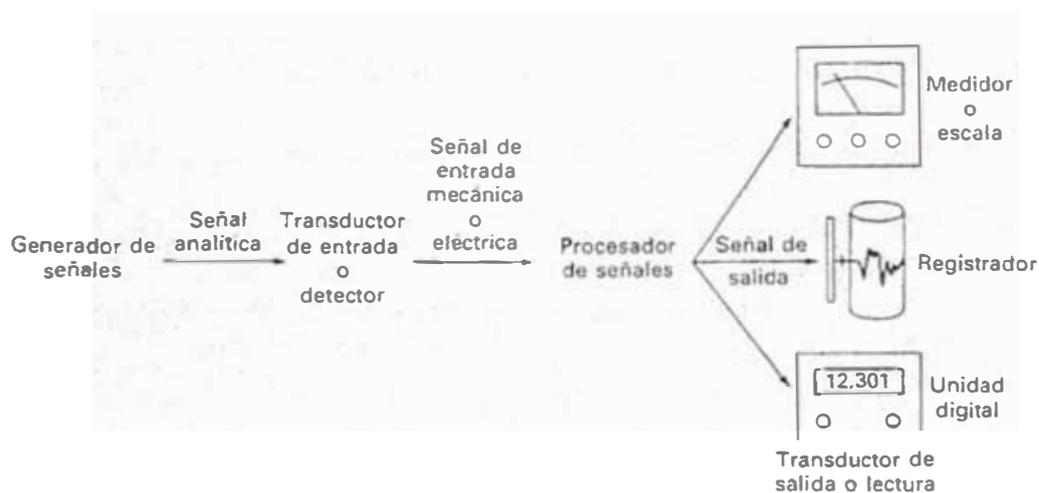
6.2 Instrumentos para el análisis

En un sentido amplio, un instrumento para el análisis químico convierte una señal analítica que no suele ser detectable ni comprensible directamente por un ser humano en una forma si lo es. Así un instrumento analítico puede considerarse como un dispositivo de comunicación entre sistema en estudio y científico.

Un instrumento para el análisis químico suele estar constituido como máximo por cuatro componentes fundamentales como se muestra en la figura siguiente.

Figura N ° 10

INSTRUMENTOS PARA EL ANALISIS



- **Generador de señales**

Un generador de señales produce una señal que detecta la presencia y con frecuencia también la concentración de la muestra. Dicho generador de señales cambia dependiendo del tipo de análisis, esto se muestra mejor en la tabla N° 6.

- **Detector (transductores de entrada)**

Un transductor es un dispositivo que convierte un tipo de energía (señal) en otro, los transductores que actúan sobre una señal química se denominan detectores. La mayor parte de los detectores convierten las señales analíticas en un voltaje o corriente eléctrica que se amplifican fácilmente para accionar un dispositivo de lectura.

- **Procesador de señal**

El procesador de señales modifica la señal transducida procedente del detector de tal forma que se adecue al funcionamiento del dispositivo de lectura. La modificación más común es la amplificación, un proceso en el que la señal se multiplica por una constante mayor que la unidad.

También se realizan de forma habitual la filtración de la señal para reducir el ruido, se multiplican por una constante menor que uno.

- **Dispositivos de lectura**

Un dispositivo de lectura es un transductor que convierten una señal procesada en otra señal que puede ser entendida por un observador humano. Por lo general la señal transducida toma la forma de la posición de una aguja en un medidor de escala.

En algunas ocasiones el dispositivo de lectura da directamente la concentración de la muestra.

Además de los equipos mencionados se tiene otros instrumentos complementarios que también se usan actualmente, como son los circuitos y dispositivos eléctricos en los instrumentos y los microprocesadores. Se muestra algunos componentes de los instrumentos a continuación.

Tabla N ° 6

COMPONENTES DE LOS INSTRUMENTOS

Instrumento	Generador de señal	Señal analítica	Transductor de entrada	Señal transducida	Procesador de Señal	Lectura
Fotómetro	Lampara de tungsteno, filtro de vidrio muestra	Haz de luz atenuado	Fotocélula	Corriente eléctrica	Ninguno	Medidor de corriente
Espectrómetro de emisión atómica	Llama, monocromador, cortador, muestra	Radiación UV o visible	Tubo fotomultiplicador	Potencial eléctrico	Amplificador, desmodulador	Registrador sobre papel
Coulombimetro	Fuente de corriente continua, muestra	Corriente de la celda	Electrodos	Corriente eléctrica	Amplificador	Registrador sobre papel
Medidor de pH	Muestra	Actividad del ion hidrógeno	Electrodos de vidrio y de calomelanos	Potencial eléctrico	Amplificador, digitalizador	Unidad digital
Difractómetro de rayos X para polvo	Tubo de rayos X, muestra	Radiación difractada	Película fotográfica	Imagen latente	Revelador químico	Imágenes ennegrecidas en una película
Comparador de color	Luz solar, muestra	Color	Ojo humano	Señal del nervio óptico	Cerebro humano	Respuesta visual al color

6.3 La selección del método analítico

Se observa de acuerdo a la descripción que hemos mencionado que nuestra aplicación será en el campo de los métodos instrumentales.

Usando nuestro equipo diseñado que por sus características se comportaría como un fotómetro, se usaría como aplicación en la espectroscopia en el campo de la luz visible.

6.4 Términos usados en la espectroscopia de absorción

Se presentan conceptos introductorios que se aplican básicamente a la espectroscopia de absorción de las radiaciones ultravioleta, visible y del infrarrojo cercano como:

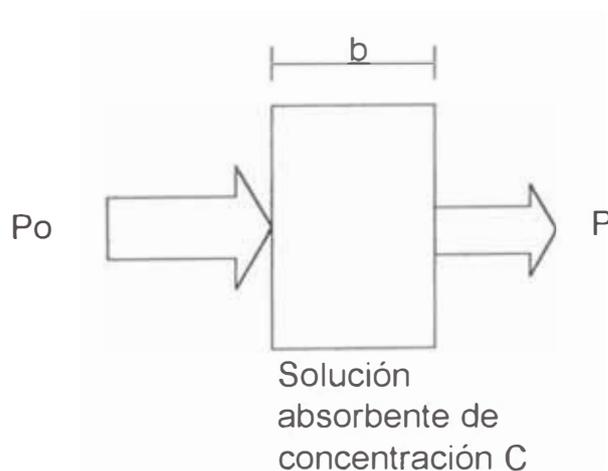
6.4.1 Transmitancia (T)

En la figura siguiente se presenta un haz de radiación paralela, antes y después de haber atravesado una capa de solución de una especie absorbente de concentración c , de b cm de grosor. A causa de la interacción entre los fotones y las partículas absorbentes, la potencia del haz se atenúa de P_0 hasta P .

Luego la transmitancia T de la muestra es por tanto, la fracción de radiación incidente transmitida por la muestra.

Figura N° 11

ESQUEMA DE LA MEDIDA DE LA TRASMITANCIA



Luego

$$T = P / P_0$$

A menudo se expresa como porcentaje:

$$\% T = (P/P_0) \cdot 100$$

6.4.2 Absorbancia (A)

La absorbancia A de una muestra viene definida por:

$$A = -\log T = \log P_0/P$$

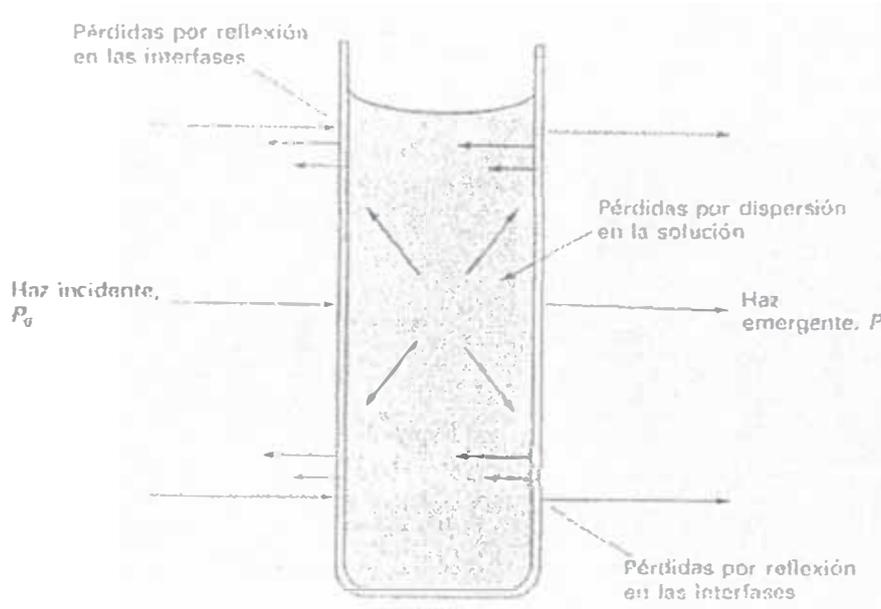
obsérvese que a diferencia de la transmitancia, la absorbancia de una solución aumenta cuanto mayor es la atenuación del haz.

Siempre existen ciertas pérdidas por reflexión y dispersión en la medida de la transmitancia y absorbancia, por ejemplo cuando se atraviesa una cubeta conteniendo agua, aproximadamente se pierde por reflexión 8.5% de un haz de luz. Además, la atenuación del haz puede producirse también por dispersión debido a las moléculas grandes de aire y a veces por absorción en las paredes del recipiente.

Aun así estas pérdidas no son tan grandes para afectar dicha medida, mas aun si estamos calculando la transmitancia y absorbancia relativa.

Figura N° 12

PERDIDAS POR REFLEXION Y DISPERSION



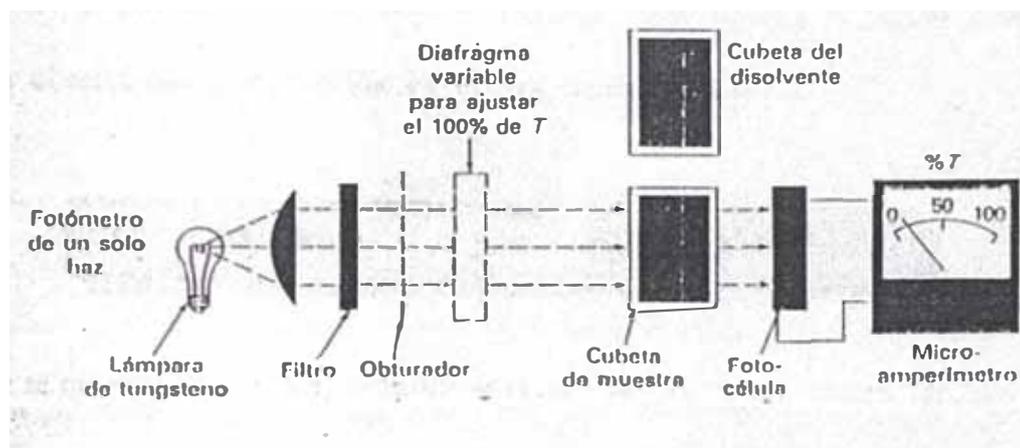
6.5 El fotómetro

Los fotómetros constituyen unas herramientas sencillas y bastante económicas para realizar análisis por absorción. Además estos presentan características de elevados rendimientos energéticos y por tanto una buena relación señal / ruido, incluso con detectores y componentes de los circuitos bastantes sencillos y baratos.

La figura siguiente muestra el esquema de un fotómetro, este consta de una lámpara de filamento de tungsteno, una lente para proporcionar un haz paralelo de luz, un filtro y una célula fotovoltaica. La corriente producida se indica con un amperímetro, cuya característica es que suele tener impresa una escala lineal.

Figura N ° 13

EL FOTOMETRO

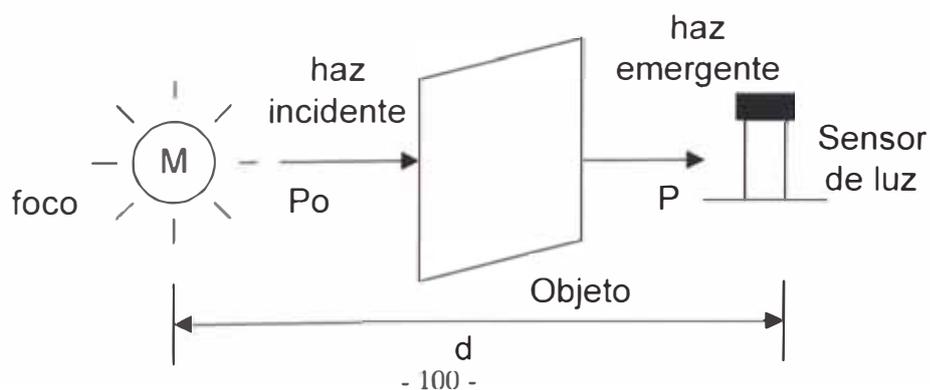


6.6 Parte experimental

De acuerdo a la teoría mencionada se tiene que para la medida de la transmitancia y la absorbancia se hace incidir la luz emitida de la fuente al sensor, colocando entre ellos diversos cuerpos u objetos para calcular la cantidad de haz emergente que llega, esto se muestra en la siguiente figura.

Figura N ° 14

ESQUEMA DE LA PARTE EXPERIMENTAL



Para hacer las diversas experiencias se colocó diversos objetos tales como: una hoja de papel, plástico transparente y plástico de colores con diferentes espesores.

Además para el cálculo de la transmitancia esta se hace de manera aproximada, dividiendo la razón entre los voltajes medidos con el objeto y sin el objeto, se observa que dicha medida es relativa, no es absoluta.

$$T = \text{Voltaje leído con objeto} / \text{Voltaje leído sin objeto}$$

Luego si se quiere calcular la absorbancia esta se hará por complemento, también se suele trabajar en el análisis con porcentajes de transmitancia y absorbancia relativa.

6.6.1 Resultados obtenidos

A continuación se muestran los diversos resultados colocando diversos objetos:

6.6.1.1 1ra experiencia con una hoja bond de espesor 0,09 mm.

SIN PAPEL	CON PAPEL	TRANSMITANCIA
0,000	0,000	0,000
0,470	0,000	0,000
2,391	0,156	0,065
2,959	0,627	0,212
3,567	0,921	0,258
4,037	1,352	0,335
4,194	1,626	0,387
4,841	2,058	0,425
4,998	2,685	0,537
4,999	2,724	0,545

SIN PAPEL	CON PAPEL	TRANSMITANCIA
0,000	0,000	0,000
1,431	0,000	0,000
2,665	0,372	0,139
3,430	1,084	0,316
3,880	1,352	0,384
4,390	1,764	0,402
4,704	2,058	0,437
4,998	2,680	0,563
4,999	2,744	0,549

SIN PAPEL	CON PAPEL	TRANSMITANCIA
0,000	0,000	0,000
1,822	0,000	0,000
3,332	0,921	0,276
4,076	1,685	0,413
4,547	2,110	0,464
4,900	2,370	0,484
4,998	2,744	0,550
4,999	2,760	0,552

6.6.1.2 2da experiencia con un plástico transparente de espesor 0,575mm

SIN PLASTICO	CON PLASTICO	TRANSMITANCIA
0,000	0,000	0,000
2,156	1,940	0,899
2,979	2,744	0,921
4,116	3,920	0,952
4,370	4,155	0,950
4,743	4,547	0,958
4,998	4,998	1,000

SIN PLASTICO	CON PLASTICO	TRANSMITANCIA
0,000	0,000	0,000
0,137	0,0196	0,143
9,548	0,411	0,750
0,999	0,8036	0,804
2,450	2,214	0,903
3,038	2,802	0,922
3,724	3,312	0,889
4,018	3,822	0,951
4,410	4,214	0,955
4,860	4,508	0,927
4,998	4,998	1,000

SIN PLASTICO	CON PLASTICO	TRANSMITANCIA
0,000	0,000	0,000
0,098	0,000	0,000
0,686	0,509	0,742
1,783	1,528	0,857
2,802	2,548	0,909
3,469	3,234	0,932
3,841	3,586	0,933
4,233	4,037	0,954
4,547	4,350	0,956
4,821	4,586	0,951
4,998	4,998	1,000

A continuación usaremos en la experiencia una mica de color ámbar de diversos espesores

6.6.1.3 3ra experiencia con una mica de espesor 0,196mm

SIN MICA	CON MICA	TRANSMITANCIA
0,000	0,000	0,000
0,117	0,000	0,000
1,254	0,274	0,218
2,646	1,254	0,474
3,332	1,920	0,576
3,920	2,450	0,625
4,606	3,175	0,689
4,998	3,684	0,737
4,999	3,704	0,740

SIN MICA	CON MICA	TRANSMITANCIA
0,000	0,000	0,000
1,352	0,411	0,304
1,999	0,842	0,421
3,175	1,803	0,568
3,253	1,783	0,548
4,390	2,940	0,670
4,900	3,547	0,724
4,998	3,684	0,737
4,999	3,704	0,740

SIN MICA	CON MICA	TRANSMITANCIA
0,000	0,000	0,000
0,176	0,000	0,000
0,235	0,000	0,000
1,842	0,705	0,383
1,920	0,725	0,510
3,528	2,156	0,611
4,018	2,567	0,640
4,508	3,096	0,687
4,998	3,743	0,749

6.6.1.4 4ta experiencia con 2 micas de espesor 0,406mm

SIN MICA	CON 2 MICAS	TRANSMITANCIA
0,000	0,000	0,000
0,392	0,000	0,000
1,372	0,000	0,000
2,763	0,686	0,252
3,665	1,372	0,374
3,704	1,293	0,350
4,390	1,881	0,428
4,606	2,156	0,470
4,998	2,763	0,552

SIN MICA	CON 2 MICAS	TRANSMITANCIA
0,000	0,000	0,000
0,000	0,000	0,000
1,215	0,000	0,000
1,313	0,000	0,000
2,352	0,431	0,183
2,822	0,646	0,229
3,528	1,156	0,327
3,724	1,332	0,357
4,488	2,097	0,467
4,998	2,665	0,533

SIN MICA	CON 2 MICAS	TRANSMITANCIA
0,000	0,000	0,000
0,686	0,000	0,000
2,254	0,352	0,156
2,293	0,294	0,128
3,194	0,960	0,300
3,292	1,078	0,327
3,880	1,430	0,368
4,429	2,077	0,469
4,478	2,646	0,531
4,998	2,763	0,553

6.6.1.5 5ta experiencia con 3 micas de espesor 0,6075mm

SIN MICA	CON 3 MICAS	TRANSMITANCIA
0,000	0,000	0,000
0,137	0,000	0,000
0,235	0,000	0,000
1,881	0,000	0,000
3,273	0,254	0,077
3,567	0,450	0,126
4,194	0,823	0,196
4,841	1,313	0,271
4,998	1,646	0,329

SIN MICA	CON 3 MICAS	TRANSMITANCIA
0,000	0,000	0,000
1,117	0,000	0,000
2,077	0,000	0,000
2,704	0,000	0,000
3,528	0,646	0,183
4,566	1,176	0,257
4,606	1,136	0,246
4,998	1,587	0,317

SIN MICA	CON 3 MICAS	TRANSMITANCIA
0,000	0,000	0,000
0,607	0,000	0,000
1,842	0,000	0,000
2,195	0,000	0,000
3,528	0,372	0,105
3,763	0,548	0,145
3,861	0,725	0,187
4,214	0,764	0,181
4,566	1,097	0,240
4,998	1,607	0,321

6.6.1.6 6ta experiencia con 4 micas de espesor 0.8225 mm.

SIN MICA	CON 4 MICAS	TRANSMITANCIA
0,000	0,000	0,000
0,646	0,000	0,000
1,685	0,000	0,000
2,528	0,000	0,000
3,684	0,098	0,026
4,037	0,215	0,053
4,292	0,294	0,068
4,645	0,450	0,097
4,998	0,803	0,160
4,999	0,823	0,164

SIN MICA	CON 4 MICAS	TRANSMITANCIA
0,000	0,000	0,000
2,097	0,000	0,000
2,979	0,000	0,000
3,214	0,000	0,000
3,724	0,098	0,026
4,155	0,137	0,033
4,900	0,627	0,128
4,998	0,705	0,141
4,999	0,833	0,166

SIN MICA	CON 4 MICAS	TRANSMITANCIA
0,000	0,000	0,000
2,018	0,000	0,000
2,861	0,000	0,000
3,077	0,000	0,000
3,861	0,058	0,015
3,978	0,078	0,019
4,214	0,254	0,060
4,880	0,568	0,116
4,900	0,686	0,140
4,998	0,803	0,160

Se menciona que las mediciones de los espesores de los objetos se hicieron con un vernier digital con una precisión de $\pm 0,01$ mm.

A continuación haremos la gráfica de transmitancia y espesor para el plástico de color ámbar, con nuestros datos construiremos nuestra tabla considerando el voltaje máximo para todos los casos, además demostraremos mas adelante que la transmitancia depende del espesor.

Espesor	Transmitancia
0.1966	0.743
0.4066	0.546
0.6075	0.322
0.8225	0.163

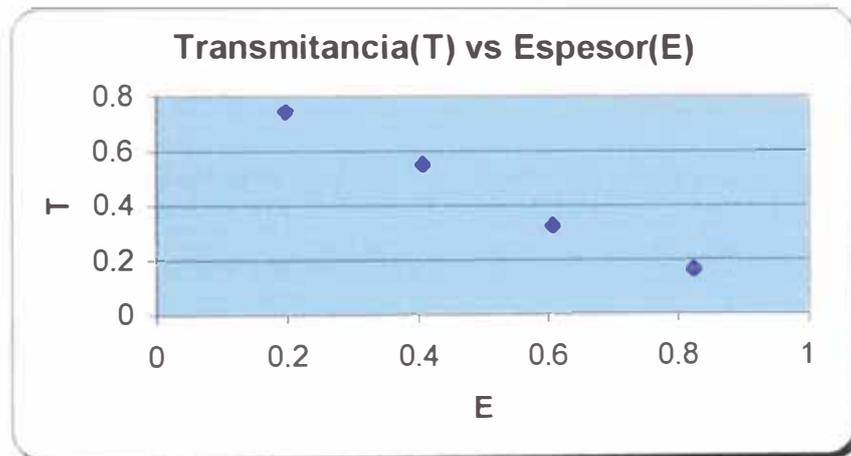


Figura N° 15

De la misma manera se hace la gráfica de absorbancia vs espesor, comprobando la validez de nuestra experiencia con las leyes experimentales conocidas.

Absorbancia	Espesor
0.129	0.1966
0.263	0.4066
0.491	0.6075
0.788	0.8225

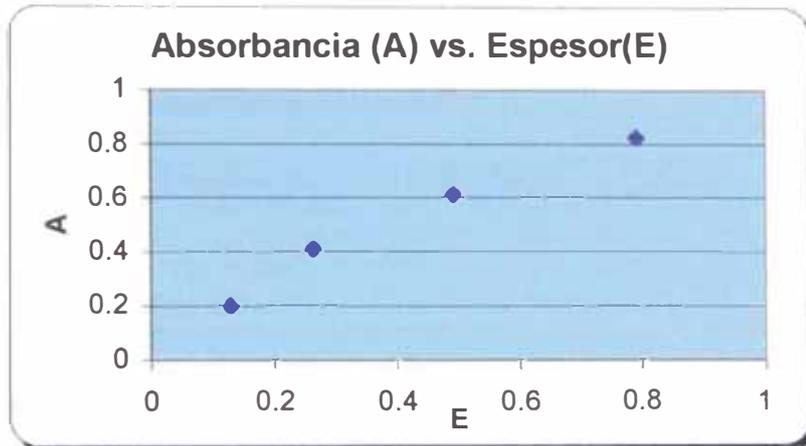


Figura N° 16

En esta figura se observa que hay una parte lineal, esta se da en el siguiente intervalo $[0,129 , 0,263]$ como menciona la ley de Beer y una parte no lineal que es en el otro intervalo $[0,491 , 0,788]$.

6.7 Tratamiento estadístico de los resultados experimentales

A continuación haremos el tratamiento estadístico de nuestros datos experimentales, usando la técnica del análisis de varianza.

Se tendrá en cuenta que en la medida de la transmitancia y absorbancia, solo interviene un factor que es el espesor de la muestra. Además se harán cuatro tratamientos, con un tamaño de muestra adecuado. Se tratara de comprobar que la transmitancia y absorbancia dependerán del espesor de la muestra.

Se tomaron otros datos en la medida de la transmitancia y absorbancia y una mica de plástico de color ámbar con un espesor adecuado.

Análisis de varianza

La siguiente tabla muestra seis observaciones para cuatro muestras de diferentes espesores.

TRANSMITANCIA^a

	ESP0196	ESP0406	ESP0607	ESP08225
1	.737	.470	.271	.160
2	.740	.552	.329	.164
3	.737	.467	.246	.141
4	.740	.533	.317	.166
5	.687	.531	.240	.140
6	.749	.553	.321	.160
Total N	6	6	6	6

a. Limited to first 100 cases.

El análisis de varianza es una técnica que se puede utilizar para decidir si las medias de dos o más poblaciones son iguales.

Con esta técnica nos permite probar la hipótesis : “La Transmitancia promedio para los cuatro tipos de espesor es la misma “; es decir que no existe

diferencia significativa en la igualdad de los promedios $H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4$
contra la hipótesis alternativa que al menos uno de los promedios es distinto.

Los supuestos que requiere esta técnica son:

- Varianzas Homogéneas
- Los Tratamientos deben tener distribución Normal

Comprobación de la distribución Normal

Nuestra Hipótesis nula es H_0 . la distribución es normal

Hipótesis alternativa H_a : no se ajusta a la normal

Si la significancia es $< \alpha$ se rechaza H_0

Si la significancia es $\geq \alpha$ se acepta H_0

Donde $\alpha = 5\% = 0,05$

De la tabla siguiente:

Se observa que para el espesor ESP0196: $0,222 > 0,05$ entonces se acepta H_0 , y así sucesivamente para los demás espesores. Luego se concluye que la población tiene distribución normal.

One-Sample Kolmogorov-Smirnov Test

		ESP0196	ESP0406	ESP0607	ESP08225
N		6	6	6	6
Normal Parameters ^{a,b}	Mean	.73167	.51767	.28733	.15517
	Std. Deviation	2.23E-02	3.92E-02	3.99E-02	1.160E-02
Most Extreme Differences	Absolute	.428	.300	.271	.328
	Positive	.219	.221	.183	.222
	Negative	-.428	-.300	-.271	-.328
Kolmogorov-Smirnov Z		1.048	.734	.665	.804
Asymp. Sig. (2-tailed)		.222	.653	.769	.538

a. Test distribution is Normal.

b. Calculated from data.

Comprobación de la Homogeneidad

Test of Homogeneity of Variances

TRANSMIT

Levene Statistic	df1	df2	Sig.
6.178	3	20	.004

Se plantea la Hipótesis nula Ho: Las varianzas son iguales

Hipótesis alternativa Ha: al menos una es diferente

Si la significancia es $< \alpha$ se rechaza Ho

Si la significancia es $> \alpha$ se acepta Ho

Luego consultando la tabla se tiene: $0,004 < 0,05$ entonces se rechaza Ho y se

concluye que los datos no son homogéneos.

Haciendo la corrida ANOVA

Hipótesis nula Ho: Las medias de los tratamientos son iguales.

Hipótesis alternativa Ha: Al menos una es diferente.

Descriptives

TRANSMIT								
	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Mica espesor 0,196 mm	6	.73167	2.232E-02	9.11E-03	.70824	.75509	.687	.749
2 Micas de espesor 0,406 mm	6	.51767	3.919E-02	1.60E-02	.47654	.55879	.467	.553
3 Micas de espesor 0,607 mm	6	.28733	3.991E-02	1.63E-02	.24545	.32922	.240	.329
4 Micas de espesor 0,8225 mm	6	.15517	1.160E-02	4.74E-03	.14299	.16734	.140	.166
Total	24	.42296	.22699	4.63E-02	.32711	.51881	.140	.749

ANOVA

TRANSMIT					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1.166	3	.389	413.374	.000
Within Groups	1.881E-02	20	9.404E-04		
Total	1.185	23			

Estadístico de la prueba: F

Si $F < F_{\alpha}$ (tabla) se acepta

Si $F > F_{\alpha}$ se rechaza

De la tabla $F_{\alpha} = 3.10$, calculada con 3 y 20 grados de libertad.

Luego : $413.374 > 3.10$ entonces se rechaza H_0 .

Esta tabla muestra la alta diferencia significativa que existe entre la transmitancia promedio para los cuatro tipos de espesor.

Para corroborar el análisis, y ver que a pesar del no cumplimiento de este supuesto, tenemos otra prueba alternativa, que es una dódima no paramétrica (no exige supuestos) y tenemos que:

Hipótesis nula $H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4$

Hipótesis alternativa H_a : Al menos una es diferente.

Con un nivel de significación del 5% concluimos que existe una diferencia significativa entre los promedios de los cuatro tratamientos de acuerdo a la tabla.

Ranks

ESPEJOR	N	Mean Rank
TRANSMIT Mica espesor 0,196 mm	6	21.50
2 Micas de espesor 0,406 mm	6	15.50
3 Micas de espesor 0,607 mm	6	9.50
4 Micas de espesor 0,8225 mm	6	3.50
Total	24	

Kruskal-Wallis Test

Test Statistics^{a,b}

	TRANSMIT
Chi-Square	21.628
df	3
Asymp. Sig.	.000

a. Kruskal Wallis Test

b. Grouping Variable: ESPEJOR

Finalmente comprobaremos la linealidad del modelo

$$Y = \beta_0 + \beta_1 X$$

Donde:

$$H_0: \beta_0 = \beta_1$$

$$H_a: \beta_1 \neq 0$$

Si la significancia $< \alpha$ se rechaza H_0 .

Si la significancia $> \alpha$ se acepta H_a .

De las tablas:

Model Summary^a

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.985 ^a	.970	.968	4.03E-02	.970	706.973	1	22	.000

a. Predictors: (Constant), Mica de espesor

b. Dependent Variable: TRANSMITANCIA

a. Dependent Variable: TRANSMITANCIA

ANOVA^b

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1.149	1	1.149	706.973	.000 ^a
	Residual	3.576E-02	22	1.626E-03		
	Total	1.185	23			

a. Predictors: (Constant), Mica de espesor

b. Dependent Variable: TRANSMITANCIA

Se concluye que $0.000 < 0.05$ entonces se rechaza H_0 luego: $Y = 0.901 - 0.941X$

Con un $R^2 = 0.97$

CONCLUSIONES

1. En cuanto al hardware y software desarrollado podría ser utilizado para sensar otras variables y utilizarlo en otras aplicaciones, como por ejemplo en la medida de la turbidez de sólidos suspendidos en una solución, colocando esta muestra entre el sensor y la fuente de luz(trabajo en proyecto).
2. Con el desarrollo del RT-Linux no es necesario recurrir al MS-DOS para construir un sistema de tiempo real, cosa que resultaría mas complicada porque se tendría que construir drivers y conseguir una costosa tarjeta de adquisición de datos.
3. El motivo del trabajo para el uso de tiempo real es debido a que se necesita realizar una buena medida de la señal entregada por el sensor a la computadora.
4. En nuestro desarrollo los datos obtenidos son "predecibles" porque estamos trabajando con un Sistema de Tiempo Real duro.

5. Los Sistemas de Tiempo Real no son más rápidos que los Sistemas Operativos normales, como podría pensarse ya que el objetivo de estos sistemas es minimizar la predecibilidad.

6. Una ventaja del Sistema Operativo Linux y su extensión de tiempo real es que es gratuito, porque los podemos tomar fácilmente de Internet sin costo alguno y seguir actualizándolo con las nuevas versiones.

7. El Sistema Operativo Linux no puede ser directamente utilizado como Sistema de Tiempo Real en primer lugar porque el sistema es de tiempo compartido, es decir una tarea en ejecución puede ser expulsada por consumir menor cantidad de tiempo. Además los sistemas de memoria virtual utilizados pueden hacer que una tarea crítica tenga paginas en el disco en el momento de entrar en ejecutarse, lo que puede hacer que ocupe un tiempo del orden de miles de veces mayor del esperado.

RECOMENDACIONES

1. Se podría hacer la misma experiencia utilizando los servicios normales de Linux (sin usar RT_Linux) para luego comparar los tiempos de respuesta y su predecibilidad. Luego observar la ventaja de los Sistemas de Tiempo Real.
2. Se podría continuar el trabajo para obtener los datos en otra computadora usando el puerto serial, para ello no se necesitaría usar el RT-Linux, sino hacer un programa de recepción de datos para el usuario en el sistema operativo Linux normal.
3. Se debe tener cuidado al trabajar con este Sistema Operativo y su extensión, porque corremos el riesgo de bloquear nuestra computadora.
4. Este trabajo podría ser ampliado en cuanto a la obtención de datos obtenidos por el ordenador ya que se podría aplicar los conceptos de realimentación y tomar una decisión en cuanto a los valores obtenidos en el computador para un trabajo futuro.

5. Se podría mejorar el trabajo en cuanto al hardware haciendo el análisis para el tratamiento de la señal, por ejemplo eliminando ciertas interferencias como ruido, etc. Aunque en el trabajo se menciona que existe una alta relación señal/ruido.

BIBLIOGRAFIA Y ENLACES

1. Auslander David M., Cheng Real Time Software for Control Program Examples in C.1990,Prentice Hall, Inc.
2. Barabanov Michael. A Linux based Real Time Operating Sistem.Catalina Michael y Catalina Alfredo. Unix/Linux 1999, McGraw-Hill.
3. Bennet Stuart. Real Time Computer Control 1988,Prentice Hall, Inc.
4. Deitel H.M. Deitel P.J. Como programar en C/C++ .1995,Prentice Hall.
5. Espinosa Juan Carlos. Linux 6.1 2000, Alfaomega.
6. Laplante Phillip. Real Time System Design and Analisis an Engineers Handbook.1998 IEE PRESS, New York.
7. Márquez Francisco Manuel. UNIX, Programación avanzada 1994,Editorial Tama

8. Martín James. Diseño de Sistemas de Computadores en Tiempo Real 1980,Prentice Hall.
9. Pressman Roger S. Ingeniería de Software 1998, McGraw-Hill.
10. Robbins Kay A. y Robbins Steven. Unix Programación Practica.1997, Prentice Hall.
11. Skoog Douglas A y Leary James Análisis Instrumental 1994, McGraw-Hill.
12. Smith Carlos y Corripio Armando. Control Automático de Procesos 1991,Editorial Limusa.
13. Tanenbaum Andrew S. Sistemas Operativos Modernos 1993, Prentice Hall.
14. Yodaiken Victor and Barabanov Michael. A Real Time Linux. Journal, February 1997.
15. Amor Juan José. Sistemas de Tiempo Real, realización en Linux. Julio 1998.
<http://aaron.ls.fi.upm.es/~jjamor/linux/rt/charla/charla.html>

16. Cuesta Aitor. Real Time Linux.

<http://members.es.tripod.de/Arioch/>

17. Epplin Jerry. Linux as an Embedded Operating System.

www.espmag.com/97/fe39710.htm

20. Pagina de RT-Linux: www.rtlinux.org.

21. Pagina de Yodaiken Victor: www.fsmlabs.com.

22. Ripoll Ismael. Real Time Linux

[http://www.linuxfocus.org/Castellano/March 2000/article 137.html](http://www.linuxfocus.org/Castellano/March%202000/article%20137.html).

23 RT-Linux Un Sistema operativo en tiempo real.

<http://nuvol.uji.es/~vcholvi/teaching/rt-linux/sld001.htm>.