



INSTITUTO TECNOLÓGICO DE SONORA

MEDIDOR DE CLOROFILA POR MEDIOS ÓPTICOS

TEMA DE TESIS

QUE PARA OBTENER EL TÍTULO DE

**INGENIERO EN ELECTRÓNICA EN
INSTRUMENTACIÓN Y CONTROL**

PRESENTA

FRANCISCO JAVIER RODRÍGUEZ PARTIDA

CD. OBREGÓN, SONORA

FEBRERO DE 2006



— —
AGRADECIMIENTOS
— —



A Dios

Por permitirme lograr este objetivo tan importante en mi vida.

A mis padres

A quienes le debo lo que soy como persona, que me han formado con su amor y paciencia. Por su apoyo en todas las decisiones que he tomado.

A mi novia


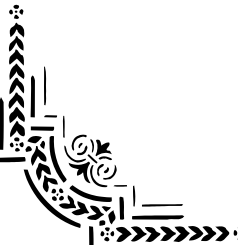
Quien con su gran amor y paciencia me ha sabido brindar su apoyo incondicional.

A mi asesor

Gracias por la confianza que depositó en mí y su apoyo que fueron muy importantes para alcanzar los objetivos deseados.

A mis revisores y maestros

Por compartir sus conocimientos, experiencias y consejos que siempre tendré en mente.



ÍNDICE

	Pág.
Lista de figuras	iv
Lista de tablas	vi
Resumen	vii

I. INTRODUCCIÓN

1.1 Antecedentes	2
1.2 Definición del problema	4
1.3 Justificación	5
1.4 Objetivos	5
1.5 Limitaciones	6
1.6 Delimitaciones	6

II. MARCO TEÓRICO

2.1 Naturaleza de luz	7
2.2 La fotosíntesis	8
2.3 Análisis de absorción	10
2.3.1 Definición	10
2.3.2 Principios de la absorción atómica	10
2.3.3 Extinción, absorción y coeficiente de dispersión	13
2.4 Elementos ópticos	16
2.4.1 El led	16
2.4.2 El fotodiodo	18
2.5 El microcontrolador (μC)	19
2.5.1 Generalidad	19
2.5.2 Modelos de los PIC's	20
2.5.3 Trayectoria de datos en la arquitectura del μC	21
2.5.4 Memoria de programa y pila (stack)	22
2.5.5 Contador del programa (PC)	22
2.5.6 Memoria de datos y registros especiales	22
2.6 Memoria EEPROM serial	22
2.6.1 Características	23
2.6.2 Descripción de los pines	23
2.6.3 Descripción funcional	24
2.6.4 Características del bus	25
2.6.5 Condiciones del protocolo	25
2.6.6 Direccionamiento de un dispositivo	26
2.6.7 Operación de escritura	26

III. MÉTODO

3.1 Bosquejo general del diseño	29
3.2 Diseño óptico y mecánico	30
3.3 Diseño del circuito	32
3.3.1 Polarización de los leds	32
3.3.1.1 Led rojo	32
3.3.1.2 Led infrarrojo	33
3.3.2 Polarización del fotodiodo	34
3.3.4 Control y procesamiento de la señal	35
3.3.4.1 Características del PIC	35
3.3.4.2 Distribución de los pines del PIC	36
3.3.5 EEPROM serial	37
3.3.6 Reloj en tiempo real (RTC)	37
3.3.6.1 Diagrama de conexión	37
3.3.6.2 Mapa de direcciones	38
3.3.7 Comunicación serial RS232	39
3.3.8 Medidor de carga de la pila	40
3.4 Interfaz con el usuario	41
3.4.1 Teclado matricial	41
3.4.2 Pantalla de cristal líquido (LCD)	42
3.5 Diseño del programa para el PIC	43
3.5.1 Herramientas computacionales	43
3.5.1.1 Matemática	43
3.5.1.2 Compilador en C	44
3.5.1.3 Simulación	45
3.5.1.4 Grabación del programa en el PIC	45
3.5.2 Diseño de la estructura de datos	46
3.5.3 Rutinas principales	47
3.5.3.1 Menú principal del sistema	47
3.5.3.2 Rutinas para el proceso de medición	48
3.5.3.3 Rutina de despliegue de datos almacenados	52
3.5.3.4 Rutina para calcular la concentración de clorofila	53
3.6 Interfaz con la computadora	54

IV. ANÁLISIS Y RESULTADOS

4.1 Situación actual	57
4.2 Análisis de resultados	58
4.2.1 Sujetos de pruebas	58
4.2.2 Alimentación	59
4.2.3 Determinación de la concentración de clorofila	59

CONCLUSIONES Y RECOMENDACIONES	63
BIBLIOGRAFÍA	64

APÉNDICES

Apéndice 1 Programador de PIC's y EEPROM serial	65
Apéndice 2 Fotos	71
Apéndice 3 Listado del programa del PIC	73
Apéndice 4 Diagrama completo del circuito	81
Apéndice 5 Programa en MatLab para generar los polinomios de ajuste	83
Apéndice 6 Glosario	85

ANEXOS

Anexo 1 Hoja de datos del PIC16F877A	89
Anexo 2 Hoja de datos del reloj en tiempo real DS12887A	96
Anexo 3 Hoja de datos de la EEPROM serial 24LC512	98
Anexo 4 Hoja de datos del interruptor analógico ADG201A	102
Anexo 5 Hoja de datos del amplificador operacional OPA1013	105
Anexo 6 Hoja de datos del fotodiodo de silicio FDS100	108
Anexo 7 Hoja de datos del Diodo Rojo BVU-5G1RD4	111
Anexo 8 Hoja de datos del led infrarrojo SLI – 0308CP	113

LISTA DE FIGURAS

No.	Descripción de la figura	Página
2.1	Características de una onda	8
2.2	Absorción de determinadas longitudes de onda por un pigmento	9
2.3	Espectros de absorción de los pigmentos fotosintéticos	10
2.4	Absorción atómica	11
2.5	Diagrama básico de un fotoespectrómetro de absorción atómica	12
2.6	Curvas de calibración ideal (negra) y real (roja)	13
2.7	Transmitancia interna	15
2.8	Partes constitutivas de un LED	17
2.9	Partes de un fotodiodo	18
2.10	Modelo Harvard	20
2.11	Distribución de pines de la EEPROM serial 24LC512	24
2.12	Configuración típica de la EEPROM serial	25
2.13	Condición de STAR y STOP	26
2.14	Direccionamiento de un dispositivo	26
2.15	Trama de escritura	27
2.16	Trama de lectura	27
2.17	Trama para leer el dato de la dirección actual	28
2.18	Trama para acceder a una localidad específica de la memoria	28
2.19	Trama para una lectura secuencial de la memoria	28
3.1	Diagrama a bloques del sistema diseñado	30
3.2	Acomodo del led rojo e infrarrojo	31
3.3	Diseño de la pinza de medición	32
3.4	Polarización del diodo emisor rojo	33
3.5	Polarización del diodo emisor infrarrojo	33
3.6	Polarización del fotodiodo	34
3.7	Resistencia equivalente de polarización del fotodiodo	35
3.8	Distribución de pines del PIC	36
3.9	Conexión de la EEPROM serial	37
3.10	Conexión del reloj en tiempo real	37
3.11	Mapa de memoria del reloj en tiempo real	38

3.12	Conexión del MAX232 y el conector a la PC	40
3.13	Medidor de carga de la pila	40
3.14	Funciones del teclado	41
3.15	Teclado matricial	42
3.16	Conexión del LCD	43
3.17	Captura de pantalla del PCW C Compiler	44
3.18	Captura de programa Proteus ISIS Professional	45
3.19	Captura de pantalla del programa IC Prog	46
3.20	Diagrama a bloques de la función principal	48
3.21	Diagrama a bloques del proceso de medición	49
3.22	Diagrama a bloques del proceso de calibración	50
3.23	Diagrama a bloques del proceso para obtener la relación del rojo con respecto al infrarrojo	51
3.24	Diagrama a bloques para obtener la medición promedio de uno de los leds	52
3.25	Diagrama a bloques para ver las mediciones	53
3.26	Diagrama a bloques de la aplicación del polinomio de ajuste	54
3.27	Captura de pantalla de la interfaz	55
3.28	Captura de pantalla de la opción configuración	56
3.29	Captura de pantalla de la opción de ajustar polinomio	56
4.1	Tira de muestra de diferentes colores y tonalidades	59
4.2	Gráfica de los valores obtenidos del ADC del led rojo e infrarrojo	60
4.3	Gráfica de la relación del infrarrojo con respecto al rojo.	60
4.4	Gráfica del medidor de concentración de clorofila comercial contra el desarrollado.	61

LISTA DE TABLAS

No.	Descripción de la tabla	Página
2.1	Materiales y frecuencias de emisión típicas de un LED	17
3.1	Hora, calendario, fecha y alarma en diferentes modalidades	39
3.2	Estructura de los datos	47

RESUMEN

El presente trabajo se desarrolló con el propósito de encontrar una forma de agilizar las mediciones de clorofila total "in situ" en un cultivo y para el posterior procesamiento de los datos. En el invernadero de ITSON se cuenta con un instrumento de medición de clorofila llamado SPAD-501, es eficiente en la toma de la muestra, pero para el análisis de los datos obtenidos, tiene muy poca memoria y no cuenta con comunicación con la computadora para transferir dichos datos.

Para los agrónomos es imprescindible tener un aparato que puede determinar en menos de dos segundos la cantidad relativa de concentración de clorofila total, ya que con esta información ellos puede conocer el estado nutrimental del cultivo y decidir aplicar más nutrientes o no. Para esto se buscó crear un prototipo que cumpla con las características básicas de: tomar una lectura en menos de dos segundos, portátil, capaz de almacenar un gran número de mediciones y con comunicación hacia la computadora para transferencia y almacenamiento de datos.

Los componentes empleados en la creación de este prototipo fueron escogidos con base en las funciones específicas que se deseaban realizar, logrando una circuitería relativamente reducida. Se empleó el microcontrolador PIC16F877A como el cerebro del instrumento, se utilizó su convertidor de analógico a digital (ADC) interno de 10bits, ahorrando usar un ADC externo y la memoria para almacenar las mediciones fue una EEPROM serial de 512Kbytes, de tan solo ocho patas.

Los resultados obtenidos fueron muy favorables, se logró caracterizar el instrumento con las pruebas realizadas con diferentes concentraciones de clorofila y grosores de hoja. Se obtuvo un rango de 0 a 50 unidades SPAD con un error de ± 0.5 .

El prototipo aun requiere de realización de pruebas de campo, todas las mediciones realizadas fueron bajo condiciones de laboratorio.

I. INTRODUCCIÓN

A pesar de que hoy en día se procura ajustar la dosis de fertilizante, siguen existiendo muchos problemas a la hora de dar una recomendación, fundamentalmente cuando se trata de abonado nitrogenado. Esto se debe a que los métodos tradicionales de estimación de clorofila en las plantas no son los más adecuados, hablando de tiempo y dinero, que son factores de gran relevancia.

Por ello estos métodos están siendo reemplazados por otros más prácticos, rápidos y más económicos. Buscando así obtener los mejores beneficios de la aplicación de nitrógeno (N), lo que se requiere y nada más.

Un indicador alternativo al N total es el obtener la concentración total de clorofila, esta determinación se puede realizar en forma rápida y económica *in situ*.

1.1 Antecedentes

El siguiente material es un resumen del artículo “Evaluación de un instrumento medidor de clorofila en la determinación de niveles de nitrógeno foliar en maíz”, la bibliografía es la número uno.

Es un hecho conocido desde hace años, que el maíz (*Zea mays* L.) es un cultivo muy exigente en N y que la fracción de recuperación del N que se le aplica es baja, normalmente inferior al 50 % (Novoa y Loomis, 1981). En Chile las estimaciones más recientes de la eficiencia de uso del N para maíz van de 27 a 38% (Fernández, 1995). Por otra parte la creciente necesidad de reducir costos, de mejorar la eficiencia en el uso de los fertilizantes nitrogenados, y evitar la contaminación con nitratos, hace necesario el uso de nuevas herramientas de medición de la condición nutricional de los cultivos y, mejor aún, si ello se logra con mediciones no destructivas de las plantas.

La técnica por método químico más ampliamente usada para la determinación cuantitativa de nitrógeno total es la de Kjeldahl (Bremmer y Mulvaney, 1982), la cual fue desarrollada en el siglo pasado y, debido al firme principio químico en el que se fundamenta, prácticamente no ha sido modificada. No obstante, tiene el inconveniente de requerir equipo especializado y, cuando se tiene un gran número de muestras para analizar, el gasto de reactivos es excesivo, lo que ocasiona que el costo de la determinación sea alto (Wood et al., 1994). Otro de los inconvenientes es el tiempo invertido desde la toma de la muestra hasta su preparación y análisis (el método completo se encuentra en el apéndice 6).

Otro método químico para determinar el estado nutrimental del cultivo es el método de Harbone, el cual sirve para la determinación de clorofila en tejidos vegetales por extracción con acetona a 80% o con N, Ndimetilformamida (Moran, 1982), es un método utilizado para estimar el estado nutrimental del cultivo en lo que se refiere al nitrógeno, magnesio y fierro principalmente (Sachdchina y Dimitrieva, 1995) y conlleva los mismos inconvenientes de tiempo y costo que la

técnica de Kjeldahl. El método completo de Harbone lo puede consultar en el apéndice 6.

Por ello se ha buscado técnicas alternativas a los métodos tradicionales de medir el estado de una planta. En investigaciones recientes, se determinó que existe una alta correlación entre la concentración de clorofila y el porcentaje de N total en hoja (Schepers et al., 1992; Peterson et al., 1993; Murdock, 1996; Echeverría y Studdert, 1998). La figura 1.1 muestra el resultado de la investigación en donde se puede observar una alta relación entre el valor obtenido del SPAD y porcentaje de N.

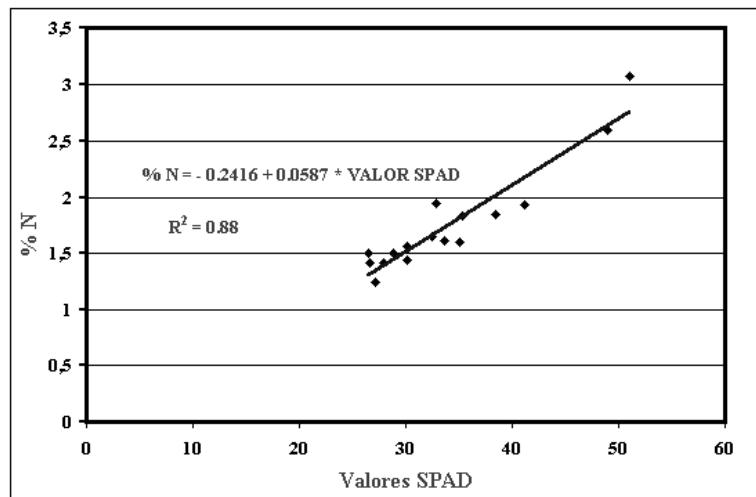


Figura No. 1.1 Relación entre porcentaje de N y valores SPAD en hojas de maíz al estado de seis hojas.

Las unidades SPAD nos indican el índice de verdor de la planta, estas unidades son asignadas por la empresa que fabrica el instrumento, pero igual tienen una alta relación con las unidades de $\mu\text{g}/\text{cm}^2$ utilizada en la determinación de clorofila total por medios químicos. La figura 1.2 muestra el resultado de la investigación realizada en el trabajo citado en la bibliografía dos. Se tomaron mediciones con el SPAD 502 y por medios químicos. El estudio se realizó con duraznos, a los cuales se les aplicó análisis estadísticos para obtener los resultados mostrados.

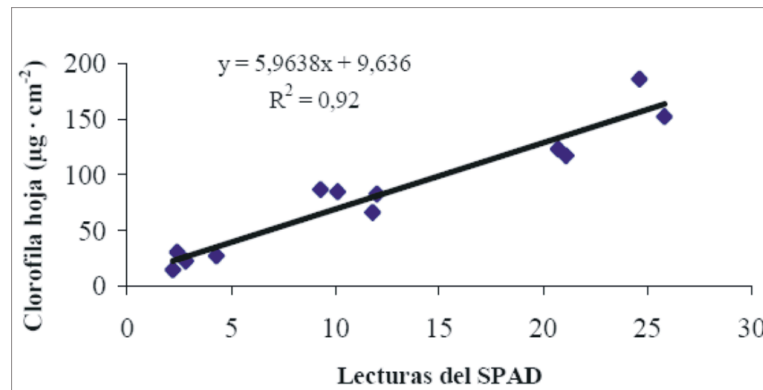


Figura No. 1.2 Gráfica de lectura con el SPAD contra mediciones por medios químicos.

Para cada tipo de cultivo la curva se modifica, pero sigue habiendo una gran relación. Lo que podemos concluir a raíz de estos resultados es que el medidor de clorofila por medio ópticos, puede ser utilizado para realizar una estimación ya sea de clorofila o de N.

Estos son algunos factores que motivaron a iniciar un proyecto apoyado por el ITSON en el 2000 a Carlos Castro Trasviña, para la medición de la clorofila. Por alguna razón no le fue posible culminarlo. Se retomó el proyecto, prácticamente desde cero.

1.2 Definición del problema

El procedimiento para obtener una medida del nitrógeno de cierta planta, por la técnica más utilizada que es la de Kjeldahl. Primeramente se toma una muestra (hoja) de la planta, luego se lleva al laboratorio para partirla en muchas partes, se le aplica reactivos, para evaluar los niveles de nitrógeno, este paso lleva cuando menos un día, antes de saber los resultados.

En base a esto podemos afirmar que las técnicas habituales de medición de nitrógeno son tardadas y costosas, debido a que se utilizan grandes cantidades de reactivos para obtener este parámetro, y requiere de tomar muestras de la planta, esto ocasiona un daño significativo, sí por ejemplo se toma la muestra cuando la planta tiene tan sólo 3 hojas.

1.3 Justificación

El instrumento en desarrollo es una alternativa económica, práctica y sencilla, pero sobre todo rápida, para estimar la cantidad de clorofila que contiene cierta planta *in situ*. La relación que existe entre la clorofila y el Nitrógeno es muy importante, ésta última se utiliza para determinar la cantidad de nutriente que necesita la planta.

Conocer la concentración de nitrógeno en la planta permite tomar una referencia, para decidir si se le aplica más nutriente o esperar hasta que lo requiera. Esto se traduce en un gran ahorro en fertilizante, además tanto la falta como el exceso del mismo son perjudiciales para la planta.

El instrumento traerá beneficios a los agrónomos del ITSON. Ellos cuentan con un invernadero, donde se realizan investigaciones y prácticas de laboratorio. Por ello, es indispensable que cuenten con un medidor de clorofila electrónico. Pero no solamente al ITSON sino a toda la comunidad que se dedique al estudio de las plantas y este proyecto da pie a otras investigaciones, en donde el instrumento puede ser caracterizado para diferentes cultivos y así hacerlo más versátil.

1.4 Objetivos

Agilizar la toma de muestras de la concentración de clorofila del cultivo y el proceso para iniciar el procesamiento de las mediciones realizadas, para ello almacenar 4000 mediciones y contara con comunicación con la computadora.

Las mediciones se realizarán *in situ*, por esa razón el aparato debe de ser práctico y ligero. El trabajo desarrollado permitirá construir un aparato con las características antes citadas, no se llegará a este punto, por limitantes que se verán más adelante.

1.5 Limitaciones

Se cuenta con muy pocas herramientas para trabajar con sistemas ópticos, como para construir lentes difusores y filtros, que serian de gran utilidad.

No se cuenta con material (plástico, aluminio u otros), ni herramientas para poder construir una estructura con acabo de producto final, es decir, con una presentación comercial.

1.6 Delimitaciones

Se realizan pruebas con plantas de hojas planas, cultivadas en el invernadero del ITSON Unidad Nañari o de las cercanas a dicha institución.

Todas las pruebas se realizaran dentro del laboratorio, en otras palabras, no se realizan pruebas al aire libre.

El programa que fungirá como interfaz con el instrumento y la computadora de desarrollara en Visual Basic 6 y la comunicación se hará a través del puerto serie de la computadora.

La presentación final del instrumento en el presente trabajo, será un prototipo para pruebas, el cual en un futuro podría servir como base para un producto comercial.

II. MARCO TEÓRICO

En esta sección se presentarán los conceptos fundamentales para comprender el principio, en el cual se basa el dispositivo que se pretende desarrollar y los componentes necesarios para el procesamiento de la lectura del transductor, para su despliegue y almacenamiento.

A continuación analizaremos algunas propiedades de la luz y de las plantas que nos serán útiles para comprender como se realiza la medición de la clorofila.

2.1 Naturaleza de la luz

El siguiente material es un resumen del artículo *Fotosíntesis* elaborado por el profesor de fisiología vegetal Rubén Hernández Gil. La luz se comporta como una onda y como una partícula. Las propiedades de onda de la luz incluyen la curvatura de la onda cuando pasa de un medio a otro. Las propiedades de partícula se demuestran mediante el efecto fotoeléctrico. En el siglo XIX, con

de la luz, como parte muy pequeña de un espectro continuo de radiación, el espectro de radiación electromagnética. Todas las radiaciones de este espectro se comportan como ondas.

La longitud de onda, es decir, la distancia entre la cresta de una onda y la cresta de la siguiente, va desde décimas de nanómetro ($1 \text{ nm} = 10^{-9} \text{ m}$) en los rayos gamma, hasta kilómetros ($1 \text{ km} = 10^3 \text{ m}$) en las ondas de radio de baja frecuencia (Figura 2.1).

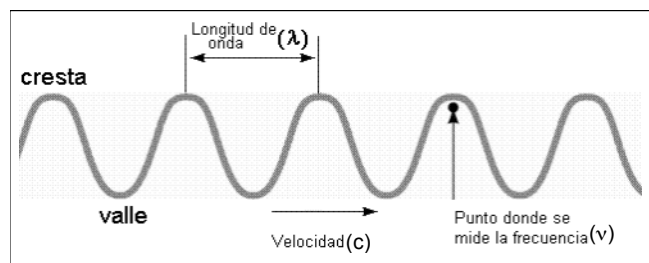


Figura 2.1 Características de una onda.

Dentro del espectro de luz visible, la luz violeta tiene la longitud de onda más corta y la roja, la más larga. Los rayos violetas más cortos contienen casi el doble de energía que los rayos más largos de la luz roja.

2.2 La fotosíntesis

La mayoría de los autótrofos fabrican su propio alimento utilizando la energía luminosa, ésta se convierte en la energía química que se almacena en la glucosa. El proceso mediante el cual los autótrofos fabrican su propio alimento se llama fotosíntesis. La mayoría de los seres vivos dependen directa o indirectamente de la luz para conseguir su alimento

Para que la energía de la luz pueda ser usada por los seres vivos, primero ha de ser absorbida. Una sustancia que absorbe la luz se denomina pigmento.

Algunos pigmentos absorben la luz en todas las longitudes de onda y por lo tanto tienen un color negro. Otros sólo absorben ciertas longitudes de onda y reflejan o transmiten las longitudes de onda que no absorben. En la figura 2.2 por ejemplo

se ve la clorofila, el pigmento que hace que las hojas sean verdes, absorbe la luz en el espectro violeta y azul y también en el rojo. Puesto que transmite y refleja la luz verde, su aspecto es verde.

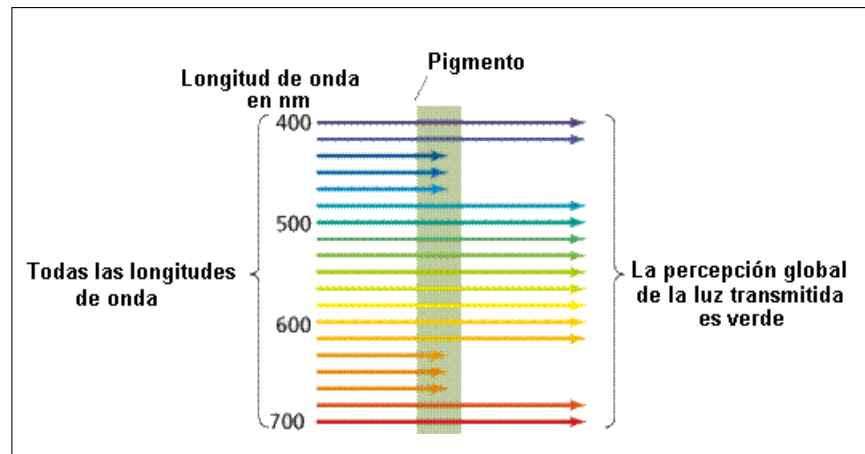


Figura 2.2 Absorción de determinadas longitudes de onda por un pigmento.

Cuando la luz choca con la materia, parte de la energía de la luz se absorbe y se convierte en otras formas de energía. En los eucariotas fotosintéticos (plantas y algas), la clorofila *a* es el pigmento implicado directamente en la transformación de la energía de la luz en energía química. Por ello cuando en una célula la luz del sol choca con las moléculas de clorofila, ésta absorbe alguna de la energía de luz que, eventualmente, se convierte en energía química y se almacena en las moléculas de glucosa que se producen.

La clorofila *b*, los carotenoides y las ficobilinas son capaces de absorber la luz a diferentes longitudes de onda de la clorofila *a*. Al parecer, pueden hacer pasar la energía a la clorofila, con lo que se incrementa la cantidad de luz disponible para la fotosíntesis (Figuras 2.3).

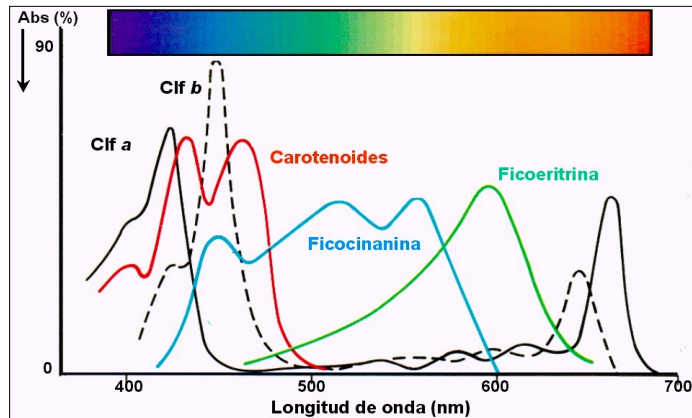


Figura 2.3 Espectros de absorción de los pigmentos fotosintéticos.

Las radiaciones con longitudes de ondas menores de 400 nm (como la luz ultravioleta) y mayores de 700 (como las infrarrojas) pueden tener diversos efectos biológicos, pero no pueden ser aprovechadas para la fotosíntesis.

2.3 Análisis de absorción

2.3.1 Definición

La espectrometría de absorción es una técnica de análisis para la determinación de elementos básicos en la absorción de radiación por átomos libres. Desde el principio y por Walsh en la mitad de los años 50's, el método ha sido mejorado y ahora está establecido en el laboratorio clínico para el análisis cuantitativo de elementos metálicos de interés clínico. La absorción atómica puede ser usada en fluidos corporales y tejidos, para determinar electrolitos, elementos esenciales, uso en terapia y monitoreo de procesos industriales. La relativa simplicidad y facilidad de la instrumentación para la absorción atómica hace posible muchos análisis, que con otras técnicas, serían muy tardadas y costosas.

2.3.2 Principios de la absorción atómica

El átomo que se necesita para la medición de absorción atómica es producido por la aplicación de una energía térmica a la muestra, suficiente para disociar los componentes químicos y producir dicho átomo libre. Los átomos de cada elemento tienen un número de electrones los cuales son asociados con el núcleo y una estructura orbital característica. Para cada electrón en un átomo dado, hay lugares discretos de niveles de energía que el electrón puede ocupar. La

configuración electrónica más estable (de baja energía) de un átomo se conoce como estado bajo. Si se aplica energía de radiación de una longitud de onda apropiada, a los átomos de estado bajo, la energía será absorbida y un electrón orbital externo será lanzado a un nivel de energía alto como se observa en la figura 2.4.

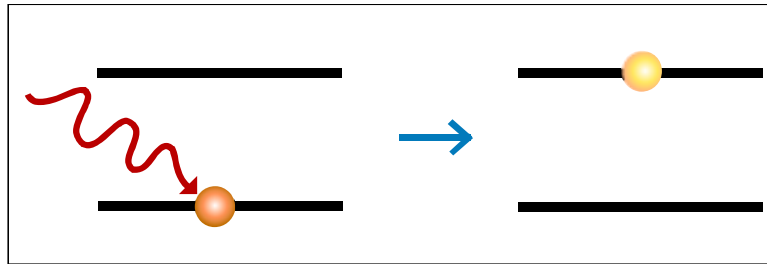


Figura 2.4 Absorción atómica.

Si la energía que se requiere para esta transición es E , entonces un fotón de longitud de onda (λ):

$$\lambda = \frac{h \cdot c}{E}$$

elevará el electrón del estado bajo a el primer nivel de energía excitado (donde h es la constante de Planck y c es la velocidad de la luz).

Aunque el requerimiento de energía para la excitación varía de elemento a elemento, los niveles de energía para un elemento dado son similares, para cada átomo de este elemento. Esto resulta en una banda de absorción muy estrecha, típicamente sólo 0.001 nm de ancho.

Las longitudes de onda a las cuales ocurre la excitación de los electrones de valencia en estado bajo, se refiere a las longitudes de onda de resonancia.

Si la luz a las longitudes de onda resonante de cierto elemento, se dirige a través de una nube de varios átomos, sólo los átomos de dicho elemento van a absorber los fotones. El requerimiento del ancho de banda de la fuente de emisión se satisface por la emisión de una lámpara especial que genera dicha frecuencia.

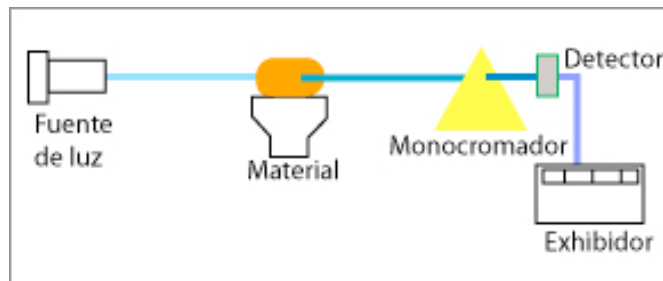


Figura 2.5 Diagrama básico de un fotospectrómetro de absorción atómica.

El proceso de absorción atómico básico se ilustra en la figura 2.5. La luz a una longitud de onda resonante es enfocada a través del material a analizar. El análisis de absorción de este material, se hace con una proporción de la radiación incidente (I_0); la sobrante (I) es transmitida a un detector donde se genera una señal eléctrica. La cantidad de radiación absorbida se determina por la comparación de I e I_0 .

La transmitancia (T) es definida como la razón de la intensidad de radiación que no fue absorbida a la intensidad de la radiación incidente:

$$T = \frac{I}{I_0}$$

Desde un punto de vista práctico, absorbancia es más utilizado. La absorbancia (A) es definida como el logaritmo base 10 del recíproco de la transmitancia:

$$A = \log\left(\frac{1}{T}\right) = \log\left(\frac{I_0}{I}\right)$$

De la ley de Beer, A es directamente proporcional a la concentración de la materia absorbente puesta para la medición de sus condiciones. Por lo tanto, si el detector suministra a un amplificador logarítmico, la señal resultante deberá ser proporcional a la concentración de átomos analizados, asumiendo, por supuesto, que la salida del detector es proporcional a la energía restante.

Cuando la absorbancia de contenidos normales de concentraciones conocidas son trazados contra concentración, el resultado es una gráfica lineal de concentración. En la práctica sin embargo, muchos trazos de calibración no son lineales, particularmente cuando se involucran altas

absorbancias alrededor de 0.5, la curva de calibración se dobla hacia el eje de concentración, como se observa en la Figura 2.6. La curvatura es causada frecuentemente por luz externa incidente. El detector monocromático pasabanda es típicamente de 0.4 nm, éste es apreciablemente más ancho que la luz de emisión y la absorbida por el elemento. Esto permite a la luz que no fue absorbida por el elemento, la oportunidad de alcanzar el detector.

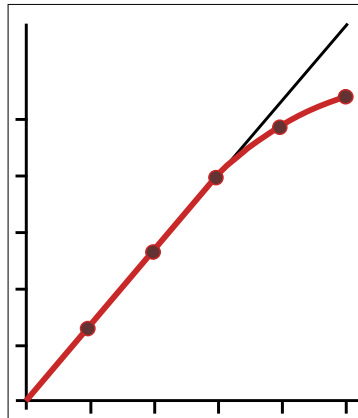


Figura 2.6 Curvas de calibración ideal (negra) y real (roja).

La luz externa que entra al sistema, estará en una señal residual, la cual a alta absorbancia de la longitud de onda resonante, puede influir en la señal de salida, produciendo una calibración no lineal.

2.3.3 Extinción, absorción y coeficiente de dispersión

Hasta aquí, hemos visto sólo el efecto de absorción, ahora veremos que pasa en conjunto con una energía radiante monocromática cuando pasa a través de un medio isotrópico.

La intensidad de radiación será decrecida porque la energía será extraída por 2 procesos:

1. Una verdadera absorción que como ya se vió, la energía radiante será convertida en energía térmica por absorción de fotones en el material y las moléculas y los átomos aceptan una cantidad de energía y convierten la energía radiante en energía molecular o atómica y no térmica.
2. Una dispersión con la heterogeneidad dentro del medio, causa cambios en algunos componentes de la radiación, algunas veces drásticas, de dirección de propagación. Estos componentes aún poseen

consecuentemente en forma de energía radiante.

Una simple fórmula, la ley de Beer-Lambert, considera el cambio en intensidad:

$$I = I_0 10^{-kl}$$

donde: I_0 = Intensidad de radiación inicial.

I = Intensidad restante después de pasar a través de un medio con espesor l .

k = coeficiente de extinción.

Esta extinción puede ser causada por ambos procesos: absorción y dispersión, así que puede ser expresada como la suma del coeficiente de absorción y el coeficiente de dispersión:

$$K = \alpha_c + \beta_c$$

Si se asume un medio homogéneo, con el coeficiente de absorción igual a cero, la relación es:

$$I = I_0 10^{(-\alpha l)}$$

dicha ecuación fue descubierta en el siglo XVIII por Pierre Bouguer, y ahora es la ley de Bouguer.

Cerca de 100 años después A. V. Beer, encontró que el coeficiente de absorción es proporcional a la concentración en un material absorbente, disuelto en un solvente no absorbente. En suma, si el material absorbente es una mezcla, el coeficiente de absorción acordado por la ley de Beer está dado por:

$$\alpha_c = \epsilon_1 c_1 + \epsilon_2 c_2 + \epsilon_3 c_3 + \epsilon_4 c_4 + \dots$$

donde:

c_1, c_2, c_3 = concentraciones individuales de los componentes.

$\epsilon_1, \epsilon_2, \epsilon_3$ = características constantes de cada componente.

La ley de absorción, puede ser derivada del postulado de que cada capa infinitesimal de material absorbente reduce la intensidad por una cantidad di que es proporcional a 2 cantidades, la intensidad de la energía radiante emitida incidente y el espesor de la capa dl :

$$di = -I \alpha_c dl$$

Cuando esto es integrado en los límites:

$$\int_{I_0}^I \left(\frac{di}{I} \right) = \alpha_c \int_0^l dl$$

Obtenemos la ecuación de la ley de absorción.

La constante de proporcionalidad α_c de la ecuación es el coeficiente de absorción. La ley de Beer-Lambert sigue el postulado, el cual no siempre es válido, que cada molécula del material absorbente, absorbe independientemente de otra molécula, las leyes de Beer-Lambert y la de Bouguer son entonces solamente aproximaciones.

Si la longitud de onda de la emisión de energía es variable continuamente en todo un rango, la razón I/I_0 varía.

La trama de esta razón como función de la frecuencia o longitud de onda es llamada espectro de absorción. La transmitancia I/I_0 a través de una muestra de material, como una función de la longitud de onda. Entonces la ecuación cambia como función dependiente de la longitud de onda:

$$I(\lambda) = I_0(\lambda)10^{[-\alpha_c(\lambda)l]}$$

En la figura 2.7 una emisión de intensidad I_1 es mostrada entrando a un material con longitud l . Debido a la pérdida de la interfase, la intensidad emitida es reducida a I_2 , y después de la transmisión a través del material, la intensidad es reducida a I_3 . Finalmente, a la salida del material, la segunda interfase hace que la emisión se reduzca a I_4 , por lo tanto, definimos: Transmitancia externa I_4/I_1 , Transmitancia Interna I_3/I_2

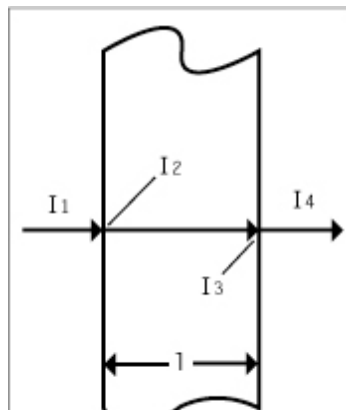


Figura 2.7 Transmitancia interna.

El tema *Análisis de absorción* se obtuvo de la bibliografía 3.

2.4.- Elementos ópticos

Para medir la clorofila se hará uso de componentes ópticos, el diodo emisor de luz (LED) y el fotodiodo, los cuales son el tema de estudio en este apartado.

2.4.1 El led

Casi todos estamos familiarizados con los leds, los conocemos de verlos en el frente de muchos equipos de uso cotidianos, como radios, televisores, teléfonos celulares y display de relojes digitales, sin embargo la falta de una amplia gama de colores y una baja potencia lumínica han limitado su uso considerablemente.

No obstante eso está cambiando gradualmente con la introducción de nuevos materiales que han permitido crear leds de prácticamente todo el espectro visible de colores y ofreciendo al mismo tiempo una eficiencia lumínica que supera a la de las lámparas incandescentes. Estos brillantes, eficientes y coloridos nuevos leds están expandiendo su dominio a un amplio rango de aplicaciones de iluminación desplazando a su anterior campo de dominio que era el de la mera indicación.

Si consideramos su particularidad de bajo consumo energético y su prácticamente imbatible ventaja para su uso en señalamiento exterior (carteles de mensaje variables y señales de tránsito) tendremos que el futuro de estos pequeños dispositivos semiconductores es realmente muy promisorio tal como lo indican los números actuales de crecimiento de mercado a nivel mundial.

Si la energía que se necesita para encenderse el led es pequeña, se tendrá que dicha energía se emitirá en ondas infrarrojas de relativamente baja frecuencia, si el material necesitara más energía para que se produzca el paso de la corriente, las ondas que emitirá el diodo tendrían más energía y se pasaría de emitir luz infrarroja a roja, naranja, amarilla, verde, azul, violeta y ultravioleta.

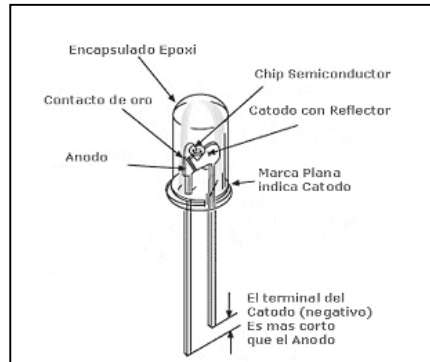


Figura 2.8 Partes constitutivas de un LED

En el esquema de la figura 2.8, podemos observar, que la parte interna de la terminal del cátodo es más grande que el ánodo, la razón es que el cátodo está encargado de sujetar al sustrato de silicio, por lo tanto será éste el encargado de disipar el calor generado hacia el exterior ya que la terminal del ánodo se conecta al chip por un delgado hilo de oro, el cual prácticamente no conduce calor.

Tabla 2.1 Materiales y frecuencias de emisión típicas de un LED.

Frecuencia (nm)	Color	Material	Frecuencia (nm)	Color	Material
940	Infrarrojo	GaAs	565	Verde	GaP
890	Infrarrojo	GaAlAs	555	Verde	GaP
700	Rojo profundo	GaP	525	Verde	InGaN
660	Rojo profundo	GaAlAs	525	Verde	GaN
640	Rojo	AlInGaP	505	Verde turquesa	InGaN/Zafiro
630	Rojo	GaAsP/GaP	498	Verde turquesa	InGaN/Zafiro
626	Rojo	AlInGaP	480	Azul	SiC
615	Rojo – Naranja	AlInGaP	450	Azul	InGaN/Zafiro
610	Naranja	GaAsP/GaP	430	Azul	GaN
590	Amarillo	GaAsP/GaP	425	Azul	InGaN/Zafiro
590	Amarillo	AlInGaP	370	Ultravioleta	GaN

Es muy importante hacer notar que en todos los casos el sustrato del led es el que determina el color de emisión y no el encapsulado.

2.4.2 El fotodiodo de silicio (FDS)

Este transductor es capaz de convertir la luz incidente en una pequeña corriente eléctrica. Los fotodiodos de silicio están contruidos por singulares obleas de silicio, similares a las utilizadas en la construcción de circuitos integrados. La mayor diferencia es que los fotodiodos requieren de un grado mayor de pureza del silicio. La pureza del silicio está directamente relacionada con su resistividad, con una alta resistividad indica una alta pureza del silicio.

La constitución típica de un fotodiodo de silicio se muestra en la figura 2.9. Se inicia con material de silicio de tipo "n". Una pequeña capa de tipo "n" se forma en el área frontal del dispositivo y por difusión térmica o por implantación de ión de un adecuado dopamiento del material (usualmente boro). La unión entre la capa "p" y la capa "n" de silicio se conoce como unión "pn". Un pequeño contacto de metal se coloca en la cara frontal del dispositivo y toda el área de atrás se recubre con un contacto de metal. El contacto trasero es el cátodo y el frontal es el ánodo. El área activa se reviste ya sea con nitrito de silicio, monóxido de silicio o dióxido de silicio para protección como recubrimiento anti-reflexión. El grosor del recubrimiento está optimizado para unas longitudes de onda en particular.

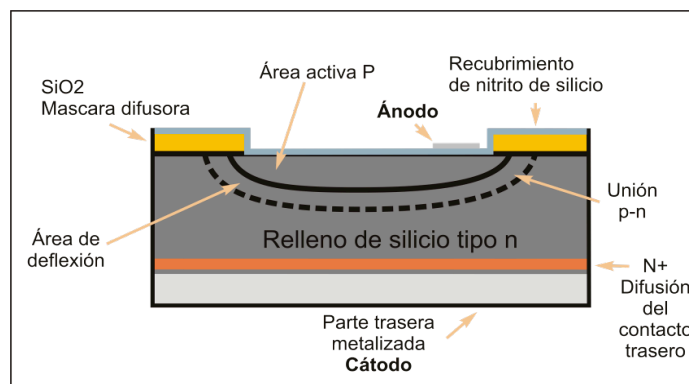


Figura 2.9 Partes de un fotodiodo.

Cuando la luz se absorbe en el área activa, se forma un par electrón-hueco. Los electrones y huecos se separan, los electrones pasan a la región "n" y los huecos a la "p". Esto resulta en la generación de una corriente debido a la luz. A la migración de electrones y huecos a sus regiones respectivamente, se llama efecto fotovoltaico.

2.5 El microcontrolador (μC)

Un microcontrolador es un circuito integrado que incluye casi todos, si no es que todos los componentes necesarios para tener un sistema de control completo.

2.5.1 Generalidad

Fue a principios de los años 70 cuando apareció en el mercado electrónico, el circuito integrado denominado microprocesador, que revolucionó el campo de la electrónica digital y analógica de una manera rapidísima y eficaz. Se implementaron numerosos sistemas de control e instrumentación industrial en torno a los microprocesadores, que sin duda alguna se imponían, no solamente en precio sino además en rendimiento y nuevas posibilidades, a los sistemas hasta entonces existentes. Los microprocesadores funcionan básicamente, como una unidad de procesamiento y control de datos. Para llevar a cabo todas las operaciones que son capaces de realizar, necesitan disponer en su entorno de una serie de elementos, sin los cuales les resultaría imposible llevar a buen término ninguna de ellas. Estos componentes auxiliares del microprocesador son entre otros, las memorias RAM, las memorias PROM, las memorias EPROM, los periféricos de entradas / salidas y más.

Los fabricantes de este tipo de microcircuitos, dándose cuenta de todo esto, desarrollaron componentes que engloban en un solo chip gran parte de estos elementos, es decir, resuelven en un solo componente las funciones propias del microprocesador. Estos nuevos microcircuitos especializados generalmente en aplicaciones industriales, constituyen lo que llamamos los microcontroladores.

Se están lanzando al mercado microcontroladores muy diversos con características particulares, que los diferencian de forma muy apreciable. Incluso un mismo fabricante pone a disposición de los usuarios varios modelos de microcontroladores, que basados en un mismo concepto tecnológico, disponen de muy diferentes posibilidades, pudiendo, de este modo el usuario elegir el modelo que mejor se ajuste a sus necesidades, no sólo en cuanto a servicio o utilidad técnica, sino también económicamente.

Hay un elemento particularmente importante que hace más o menos populares a estos elementos, que es el conjunto de herramientas de desarrollo y programación puestas a disposición del usuario. Estas herramientas nos permiten diseñar, programar, depurar, simular e implementar nuestros propios sistemas de control con relativa facilidad.

Es por esto, entre otras cosas, que la empresa Microchip, está imponiendo en el mercado con gran fuerza sus componentes microcontroladores denominados PIC.

2.5.2 Modelo de los PIC

En general, podemos afirmar que todos los PIC siguen el modelo Harvard de estructura interna, de forma tal que, la Unidad Central de Proceso dispone de los buses necesarios para comunicarse por un lado con la memoria de programa y por otro lado con la memoria de datos, de forma totalmente independiente. En la figura 2.10 podemos ver, cómo internamente, el microcontrolador dispone de buses separados para acceder a las distintas unidades de memoria.

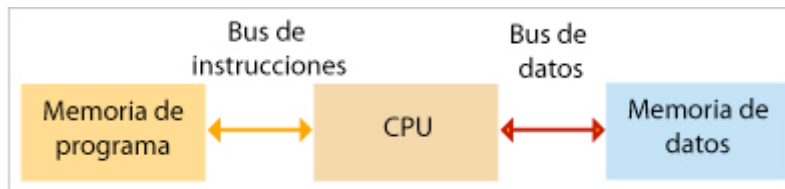


Figura 2.10 Modelo Harvard.

Estos microcontroladores, están diseñados según la tecnología RISC, por la cual disponen de un juego de instrucciones bastante reducido.

En la ejecución de las instrucciones, durante el desarrollo de un programa, los PIC emplean un procedimiento de segmentación que les permite ejecutar la instrucción en curso y al mismo tiempo buscar el código de la instrucción siguiente, ahorrando de este modo un precioso tiempo de ejecución. Cada instrucción se ejecuta en "un ciclo de instrucción" que equivale a "cuatro ciclos de reloj".

Memoria no volátil para contener el programa. No hay posibilidad de utilizar memorias externas de ampliación. Son 5 los tipos de memoria adecuados para soportar esta función, estas son:

1. ROM. Se graba el chip durante su fabricación, implica costos altos y sólo se recomienda cuando se produce en serie.
2. OTP. Se graba una sola vez con un dispositivo que se gobierna mediante un computador personal, que recibe el nombre de grabador, se borra con rayos UV para reprogramarse.
3. EPROM. Se graba por el usuario igual que la memoria OTP, a diferencia de la OTP se puede grabar varias veces.
4. EEPROM. La grabación es similar a la de las memorias EPROM y OTP, la diferencia es que el borrado se efectúa de la misma forma que el grabado, o sea eléctricamente.
5. FLASH. Posee las mismas características que la EEPROM, pero ésta tiene menor consumo de energía y mayor capacidad de almacenamiento, por ello está sustituyendo a la memoria EEPROM.

2.5.3 Trayectoria de datos en la arquitectura del μC

Algunas funciones internas las realiza como sigue:

- La memoria de datos (RAM) está organizada en cuatro bancos de 512 registros de 8 bits,
- La RAM contiene todos los registros del μC , incluyendo los utilizados para puertos y periféricos como para datos.
- Una dirección a la memoria tiene 9 bits y se calcula de dos modos: directo o indirecto. La dirección directa llega del IR y la dirección indirecta llega del registro FSR.
- La ALU trabaja con datos de 8 bits y utiliza los bits de acarreo (C), de acarreo de dígito (DC) y de cero (Z), del STATUS.
- El bus de datos comunica a los registros del μC con la salida de la ALU.
- Los datos de entrada a la ALU son: el registro de trabajo W y otro dato que llega del registro de instrucción o por el bus.
- El resultado que obtiene la ALU puede ir al registro de trabajo W o por el bus a otro destino.

2.5.4 Memoria de programa y Pila (stack)

La memoria de programa de 8kx14 bits está organizada en cuatro bancos de 2kx14bits. Una dirección a la memoria debe tener 13 bits. El registro contador de programa (PC) es de 13 bits y contiene la dirección completa. El stack de 8 niveles guarda el contador de programa en llamadas a subrutinas e interrupciones. El μ C no dispone de instrucciones para meter y retirar datos en el stack.

2.5.5 Contador del programa (PC)

Cuenta con las siguientes características:

- El contador de programa de 13 bits, apunta a la instrucción que el procesador va a ejecutar. La ejecución de la instrucción deja al PC apuntando a la siguiente instrucción.
- La memoria de programa es de 8kx14 (8 x 1024 localidades de 14 bits).
- El stack o pila tiene 8 lugares por lo que se pueden anidar hasta ocho llamadas a subrutinas o servicios a interrupciones.
- La operación de inicialización se ejecuta con el reset manual o al encender la fuente de voltaje, y deja al PC apuntado a la dirección 0x0000, por lo que en esta localidad de la memoria de programa debe estar el programa de inicio o el vector al programa de inicio.

2.5.6 Memoria de datos y registros especiales

La memoria de datos, está organizada en cuatro bancos de 128 bytes, Los primeros 32 bytes se utilizan para manejar periféricos y recursos internos del PIC y el resto queda disponible para uso general, como se detalla enseguida:

- Banco 0: 96 bytes disponibles
- Banco 2: 80 bytes disponibles
- Bancos 2 y 3: 96 bytes cada uno

2.6 Memoria EEPROM serial

Las memorias EEPROM (electrically erasable and reprogrammable ROM) o E²PROMs como son llamadas habitualmente, son memorias más caras y más

rápidas que las EPROM (pueden tener tiempos de acceso alrededor de 35 ns) y una vida media en torno a los 10.000 ciclos de borrado/escritura.

EL componente principal para almacenamiento de las mediciones será éste precisamente, el cual se escogió por las facilidades que brinda como se citan a continuación.

La tecnología de EEPROM de serie es uno de las tecnologías de memoria no-volátiles que han surgido como una solución principal de diseño.

2.6.1 Características

El rasgo principal de este dispositivo de serie es, como su nombre implica, la habilidad de comunicar a través de una interfaz de serie. Esta memoria tiene numerosos beneficios:

1. Primero, la comunicación serial se realiza con un número mínimo de I/O.
2. Otro beneficio de comunicación serie es el tamaño del paquete. Que va de las densidades de 16 a 1 Mbit; esta memoria está disponible en un chip de 8 pines, esto es muy útil para las aplicaciones donde el tamaño del producto y el peso es un factor del plan importante.
3. Consumo actual bajo. Debido a un número limitado de puertos de I/O operando corrientes que están normalmente debajo de 3 mA.
4. El byte de programación, habilidad de borrar y programar al mismo tiempo sin afectar el volumen del paquete.
5. La tasa del reloj está entre 100kHz y 400kHz para dispositivos de dos conductores.

2.6.2 Descripción de los pines

La figura 2.13 nos muestra la distribución de pines, los cuales tienen las funciones que a continuaciones se describen:

- **A0, A1, A2** son los pines de entrada, los cuales son usados por la memoria para múltiples operaciones. Los niveles de entrada son comparados con los bits correspondientes en la dirección del esclavo. El chip habilita si la comparación es verdad

- **SDA (data serial).** Es un pin bidireccional (entrada y salida de datos). Para habilitar el pin de SDA debe tener conectada una resistencia del pin (SDA) a VCC (10 k Ω desde 100 kHz y 2k Ω desde 400 kHz). SDA puede estar habilitado solo cuando el pin SCL está en nivel bajo. La cual puede trabajar en dos condiciones START o STOP.
- **SCL (serial clock).** Sincroniza la transferencia de datos del dispositivo.
- **WP.** Pata de protección, si está conecta a tierra o no está conectada la memoria está desprotegida y se puede leer y escribir, pero si esta a Vcc se encuentra en sólo lectura.

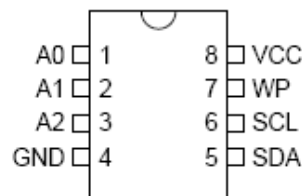


Figura 2.11 Distribución de pines de la EEPROM serial 24LC512.

2.6.3 Descripción funcional

Las principales características de esta memoria son:

- Es una memoria de 512 kbits organizada a 32 k *8 Bits.
- Soporta protocolo de transmisión de datos en bus de dos líneas.
- El bus se controla mediante un master que se encarga de generar el reloj (**SCL**), el control de acceso al bus y además genera las condiciones de **START** y **STOP** mientras la memoria actúa como **slave**.
- El rango de voltaje de alimentación de 1.8 a 5.5 V.
- Máxima frecuencia del reloj de 400kHz.
- Máxima corriente en escritura 3mA a 5.5 V.
- Máxima corriente en lectura 400 μ A a 5.5 V.

La figura 2.12 muestra una configuración típica de esta memoria:

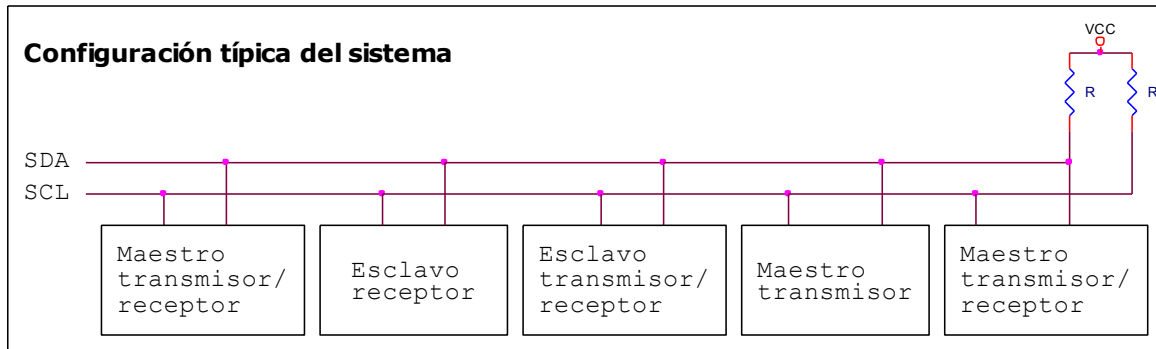


Figura 2.12 Configuración típica de la EEPROM serial.

2.6.4 Características del bus

Sólo puede iniciarse transferencia de datos cuando el bus está libre. Durante la transferencia de datos, estos deben permanecer estables mientras la línea de reloj esté a nivel alto. Si se producen cambios se interpretan como un Start o un Stop.

2.6.5 Condiciones del protocolo

Las condiciones que se deben de cumplir en un tiempo determinado, dependiendo de la operación que se esté realizando.

- Bus libre - Las líneas de datos y de reloj están a valor alto.
- Start en la transferencia - Transición de alto a abajo mientras el reloj está alto.
- Stop en la transferencia - Transición de bajo a alto mientras el reloj está alto.
- Validación de datos - La línea de datos presenta un valor válido cuando después de la condición **start**, la línea se mantiene estable mientras el reloj está alto. Los datos únicamente pueden cambiar mientras el reloj esté en valor bajo (Un pulso por dato). Cada transferencia se inicia con un **start** y finalizada con un **stop**. El número de bytes entre ambos está determinado por el **master** y teóricamente no está limitado, a pesar de todo sólo los últimos 16 bytes se almacenan durante una operación de escritura. Cuando se produce una sobré-escritura se reemplazan los datos en un esquema **FIFO** (First In First Out).
- Reconocimiento - Cada receptor, cuando se le direcciona, está obligado a

generar un pulso de reloj extra que se asocia al **ACK**.

El **ACK** se da colocando un valor bajo sobre **SDA** durante el valor alto del reloj.

La figura 2.13 muestra cada fase:

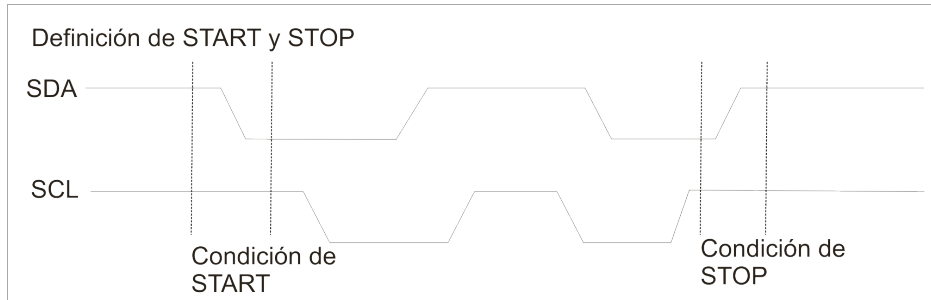


Figura 2.13 Condición de STAR y STOP.

2.6.6 Direccionamiento de un dispositivo y operación

Después del **START** se envía un byte de control, éste consiste en cuatro bits que indican la operación, tres bits indicando la página (**A2**, **A1**, **A0**) que corresponden realmente a los bits de direccionamiento más altos y el último bit que indica la operación a realizar (un **1** - **lectura**, un **0** - **escritura**). Una vez enviado el dispositivo responde con un **ACK**.

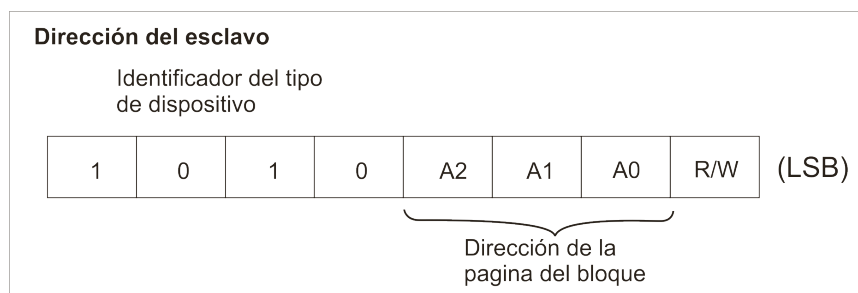


Figura 2.14 Direccionamiento de un dispositivo.

2.6.7 Operaciones de escritura

La escritura comienza por un **START**, direccionamiento del dispositivo, dirección del byte y finalmente el byte a escribir, después de cada elemento se añade un **ACK** por parte de la memoria como se indica en la figura.

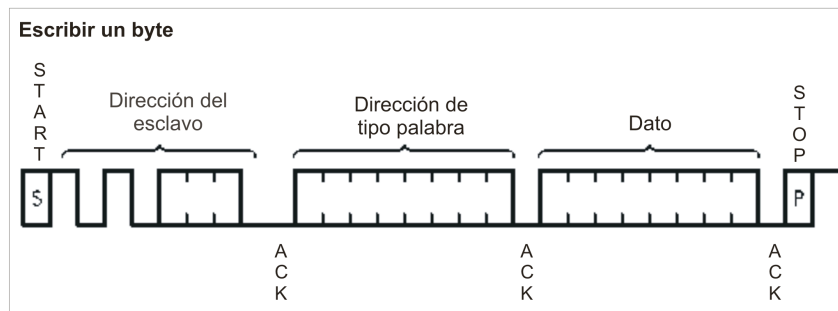


Figura 2.15 Trama de escritura.

Si lo que se desea es escribir un grupo de 16 bytes sólo es necesario enviar la dirección del primero de ellos y a continuación el resto, la memoria posee un contador interno que determina la dirección actual, éste se irá incrementando para cada byte, a esto se le llama escritura secuencial.

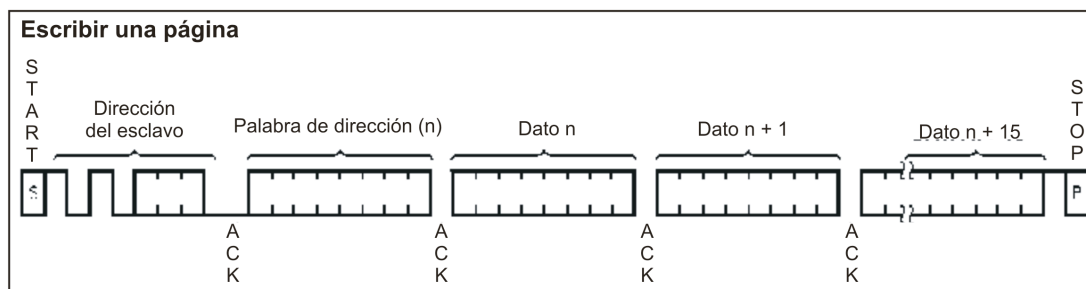


Figura 2.16 Trama de lectura.

La diferencia respecto a la escritura de un único byte estriba en el hecho del señalizador del **STOP**, si éste no aparece significa que se van escribir sucesivos bytes. La figura 2.18 muestra el caso. Una escritura incluye un borrado del contenido de la dirección por lo que es mucho más lento que una lectura.

2.6.8 Operación de lectura

Las operaciones de lectura son 3, la lectura de la dirección actual (almacenada por el contador de dirección), lectura arbitraria en cualquier posición de la memoria y finalmente la lectura secuencial, como se describe a continuación:

- **Lectura actual:** Toma la posición actual de la memoria, como puede verse en la figura, después del dato no hay un **ACK**, ya que éste va dirigido al master.

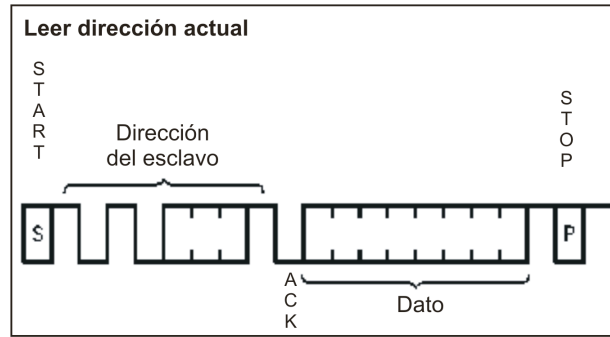


Figura 2.17 Trama para leer el dato de la dirección actual.

- **Lectura arbitraria:** En esta lectura se indica la posición a leer la información, hay que destacar que primero se ha de enviar la dirección y después se recibe el dato, por eso existe un **ACK** después de la dirección.

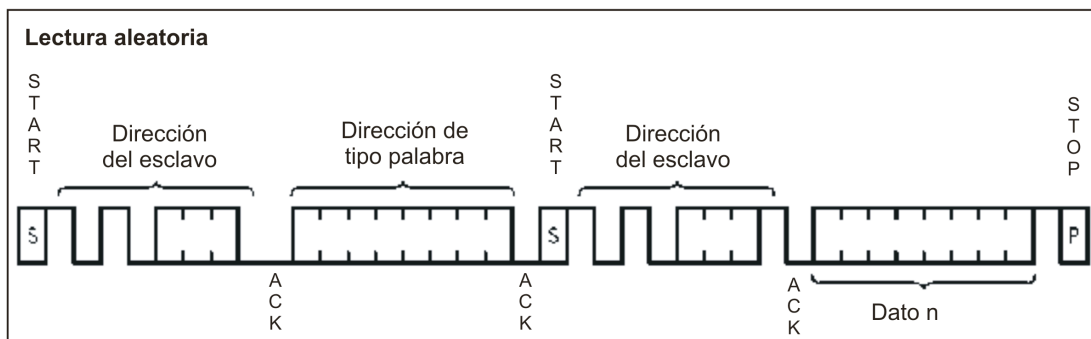


Figura 2.18 Trama para acceder a una localidad específica de la memoria.

- **Lectura secuencial:** La lectura secuencial toma la dirección actual y va extrayendo los datos uno detrás del otro, hay que tener en cuenta que se lee la página actual y ésta no se cambia cuando se llega al final por lo que la lectura como máximo es de 255 bytes.

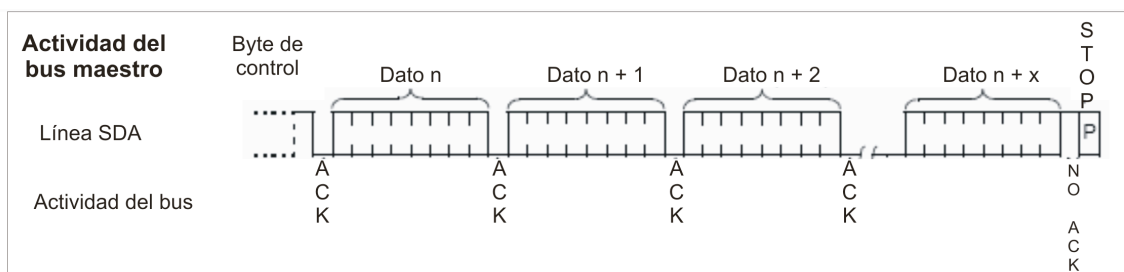


Figura 2.19 Trama para una lectura secuencial de la memoria.

III. MÉTODO

En este capítulo conoceremos el diseño del instrumento medidor de clorofila, tanto de la parte electrónica, óptica y mecánica. Por otro lado veremos las herramientas para: programar el PIC, simular parte del sistema y el programa de interfaz con la computadora, pero antes veremos el proceso que se siguió para el diseño de cada etapa.

3.1 Bosquejo general del diseño

Los pasos que se siguieron en el diseño del instrumento fueron los siguientes:

- Se investigó el espectro de absorción de las hojas de las plantas y de la forma de tomar las muestras y se idearon las formas de realizarlas.
- Elección del modelo del PIC, el cual se escogió en base a las funciones que se necesita que realice el aparato.
- Se armó el circuito propuesto a la vez que se realizaba el programa del PIC.

- Se realizó la pinza de medición y la interfaz para la computadora.
- Por último se realizaron pruebas y ajustes.

Este proceso condujo a realizar el sistema como el que se muestra en el diagrama a bloques a continuación.

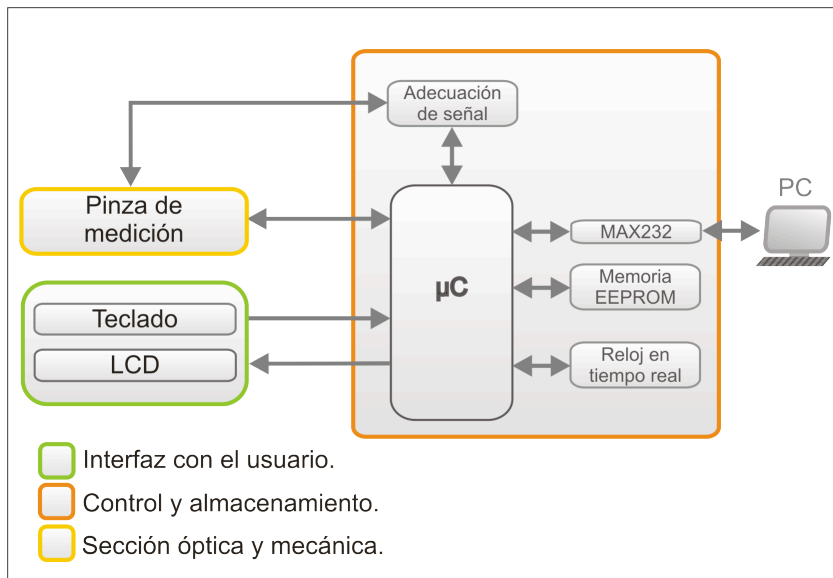


Figura No. 3.1 Diagrama a bloques del sistema diseñado.

3.2 Diseño óptico y mecánico

Como se vio en los antecedentes teóricos en la figura 2.3, donde se muestran los pigmentos que intervienen en la fotosíntesis de la planta, el espectro de color rojo se ve una zona donde la clorofila tipo *a* tiene una mediana absorbancia, muy poco por la clorofila de tipo "*b*" y no hay otros pigmentos que absorben en el espectro del rojo que está entre los 640 y 700nm. La pequeña absorción debido a la clorofila *b* la podemos despreciar.

Siendo más específicos el pico de absorción de la clorofila de tipo *a*, se encuentra en aproximadamente 660nm, por lo cual el led utilizado corresponde con este requerimiento, ya que tiene su máxima intensidad a esta misma longitud de onda, según el fabricante.

Las unidades en las que se realizan la medición son $\mu\text{g}/\text{cm}^2$, como se puede notar, se está considerando solamente un área, aunque obviamente un grosor

muy pequeño casi despreciable, pero no un volumen, por ese motivo se necesita eliminar la absorción debido al grosor de la hoja.

Para ello a pesar de que se cuente con un led con las características deseadas, existe factores como el grosor de hoja ya mencionado, coloraciones oscuras y entre otros, que absorben la luz roja, y que impiden realizar una medición directa, es decir, con una sola longitud de onda.

Es usar una longitud de onda que no sea absorbida por la clorofila *a*, las longitudes de ondas dentro del infrarrojo no son absorbidas por ningún pigmento de hoja. El led infrarrojo se usa para tener dos patrones, primero realizando la medición de absorción con la luz infrarroja y luego con la roja, para determinar que tanto de la absorción del rojo se debió a la clorofila y no a otros factores.

Se colocaron los leds rojo e infrarrojo de manera que los dos tengan dirigido su haz de luz hacia un mismo punto. La manera más sencilla de lograrlo se coloca directamente el led infrarrojo hacia un punto y el led rojo inclinado apuntando al mismo punto, gracias a que la luz roja es más sensible a la difracción, se facilita cambiar la dirección de la luz, con sólo hacerla incidir en un material traslucido de color blanco (de este color para que no se atenué demasiado), como se muestra en la figura 3.2.

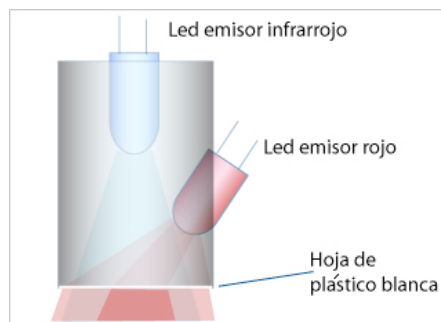


Figura No. 3.2 Acomodo del led rojo e infrarrojo.

El fotodiodo de silicio (FDS100), se escogió con las características adecuadas a nuestras necesidades, dentro de las cuales están: alta sensibilidad a la luz visible (sobre todo para el rojo) y también para la no visible en las longitudes de ondas del infrarrojo, la cual se encuentra arriba de los 700 nm y la capacidad de sensor

La figura 3.3 muestra la estructura para montar los leds, en la cual se incluye un interruptor para saber cuando la pinza se ha cerrado. El tope es con la intención de que la pinza cierre a la misma distancia y exista la misma distancia entre emisor y receptor cuando permanezca cerrada. El limitador de área de sensado tiene una ventan de aproximadamente 1.7x2.5 mm.

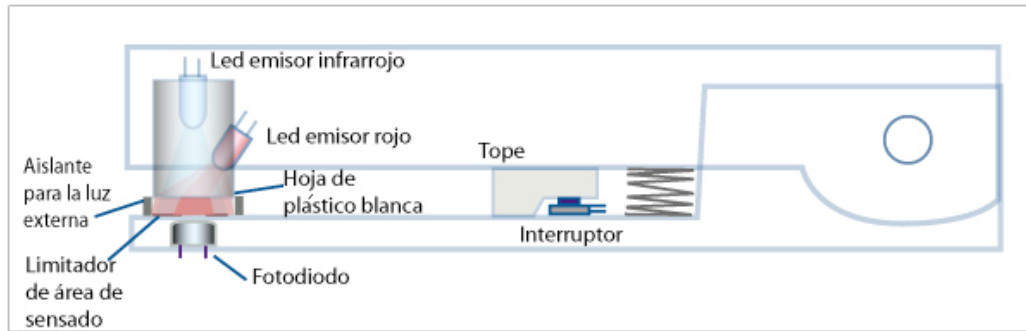


Figura No. 3.3 Diseño de la pinza de medición.

3.3 Diseño del circuito

3.3.1 Polarización de los leds

Se realizará el análisis de varios aspectos que hay que tener en cuenta para el correcto funcionamiento del led rojo e infrarrojo. Es necesario ajustar la intensidad independiente de estos dos ya que se tiene un sólo sensor.

3.3.1.1 Led rojo

La luz roja será absorbida en mayor cantidad, es necesario que ésta sea lo más intensa posible, pero hay dos aspectos a considerar:

1. A mayor corriente que circule a través de el led, tenderá a disminuir la longitud de onda a la que se presenta su pico máximo de luz, por ello lo ajustaremos, para hacer circular una corriente menor de 20mA, que es donde sabemos que tiene un pico máximo arriba de 660nm, según las hojas de datos del fabricante (ver anexo 7).
2. Como se trata de un instrumento portátil, éste será alimentado con una pila de 9V, lo cual es de tomarse en cuenta para disminuir lo más posible el consumo de corriente.

Por estas dos razones expuestas anteriormente fue necesario hacer más

inicialmente cuando se ajustó la polarización del led rojo para que circulara una corriente de 20mA, pero se presentó una variación en el voltaje considerable en la salida del regulador de voltaje (LM7805), en el momento de que el led se encendía, provocando errores en las mediciones.

El PIC tiene suficiente corriente para alimentar al led, como máximo puede suministrar una corriente de 25mA en cualquier pata de todos sus puertos. En la figura 3.4 se muestra la configuración, donde el diodo rojo se alimenta directamente de la pata 2 del puerto A (RA2) a través de un potenciómetro (P1), el cual servirá para ajuste de intensidad a causa de algún cambio que se realice.

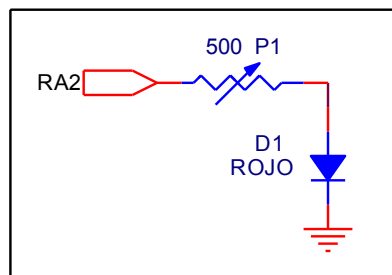


Figura No. 3.4 Polarización del diodo emisor rojo.

3.3.1.2 Led infrarrojo

El emisor infrarrojo requiere aun menor corriente para funcionar, aproximadamente 10 mA menor, por ello al igual que el led rojo se alimenta por una pata del puerto, en este caso RC2. Se ajusta su intensidad de acuerdo a la intensidad que se alcance con el emisor rojo y la sensibilidad del fotodiodo. Por ello se colocó un potenciómetro (P2), como se muestra en la figura 3.5.

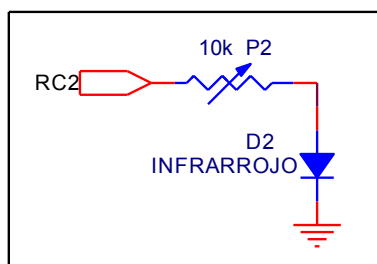


Figura No. 3.5 Polarización del diodo emisor infrarrojo.

3.3.2 Polarización del fotodiodo

Esta etapa del diseño, ha requerido de mayor tiempo de desarrollo, ya que con las pruebas realizadas, fue posible determinar la configuración que se adecua a nuestras necesidades.

El fotodiodo se polariza de forma inversa como se muestra en la figura 3.6. Se polariza con una combinación de cuatro resistencias para conectarlas a tierra se utiliza un circuito integrado que cuenta con cuatro interruptores, es un interruptor bidireccional pero se escogió las patas Dx para ser conectadas entre sí y polarizar al fotodiodo. Combinando las resistencias que se conectan en paralelo a través del interruptor analógico, se logra tener una escala exponencial la cual se muestra en la figura 3.7, la resistencia equivalente va desde $545\text{k}\Omega$ hasta $8.2\text{M}\Omega$. Para obtener la resistencia equivalente hay que ir obteniendo el equivalente en paralelo de cada una de las resistencias que están conectadas al interruptor, cuando hay un cero en RDx el interruptor se cierra, por tanto la resistencia equivalente será todas las combinaciones en paralelo excepto cuando queden todos en uno, es una combinación no válida, porque quedarían todas las resistencias desconectadas. Para ir aumentando paso a paso la resistencia, a los cuatro interruptores se le introduce las combinaciones de un contador ascendente, donde el bit menos significativo es el RD0 y el más el RD3.

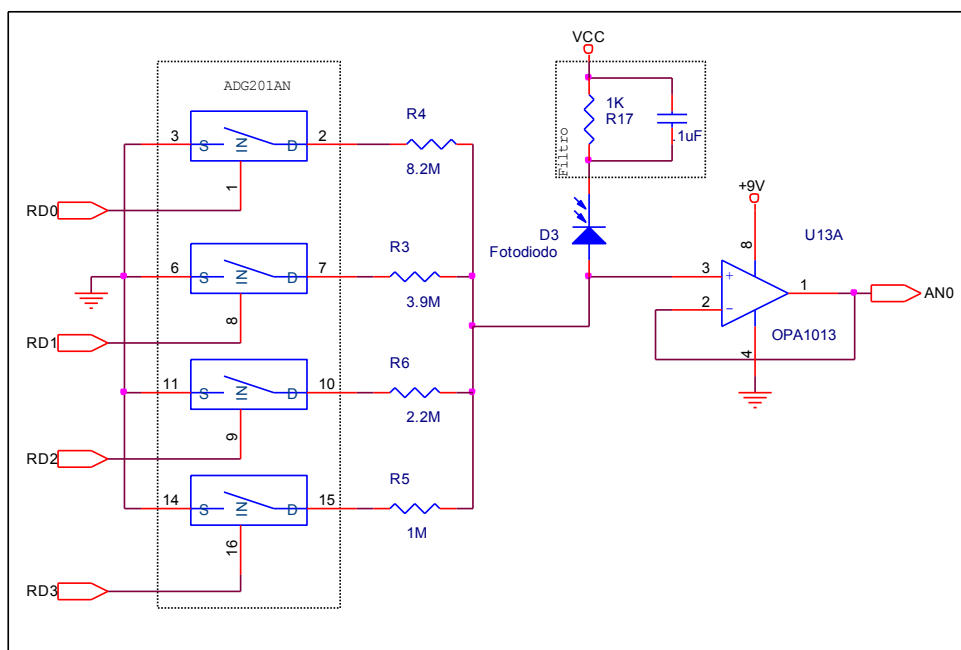


Figura No. 3.6 Polarización del fotodiodo

Como observación decimos que los interruptores presentan una resistencia de 168Ω cuando están cerrados, la cual es muy despreciable, ya que estamos manejando valores de resistencia arriba de los $200k\Omega$.

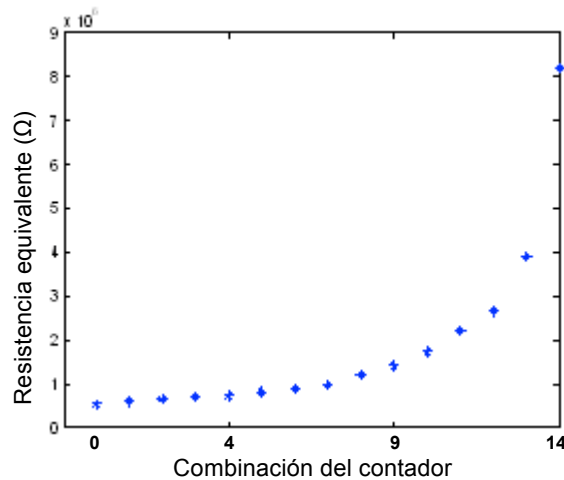


Figura No. 3.7 Resistencia equivalente de polarización del fotodiodo.

El capacitor y la resistencia de $1k\Omega$ es un filtro recomendado por el fabricante del fotodiodo para reducir el ruido de alimentación.

Por último tenemos un seguidor de voltaje, como acoplador de impedancia con la entrada del convertidor de analógico a digital.

3.3.3 Control y procesamiento de la señal

3.3.3.1 Características del PIC

El PIC16F877A se escogió dadas las características que a continuación se describen:

1. Memoria FLASH de programa de 8 K x 14 bits.
2. SRAM de 368 localidades.
3. Tres puertos de 8 bits, uno de 6 bits y otro de tres bits.
4. Comunicación serial RS232.
5. Convertidor de analógico a digital (ADC) de 10 bits.

El factor más importante de los mencionados anteriormente es la memoria de PIC, ya que se preveía que se necesitaría mucha memoria de programa, porque se realizan operaciones de punto flotante, las cuales consumen mucho espacio. Además, para realizar el programa se usó un compilador en C, del cual se

explicará un poco más adelante, éste facilitó mucho la programación, pero la desventaja es la poca eficiencia en el uso de la memoria.

El ADC del PIC en un principio no se pensó usar, ya que se tenía el AD974 de 16 bits, pero conforme se realizaron pruebas se concluyó que para el uso que se le da, resulta igual de útil el ADC interno de 10 bits del PIC, ahorrando espacio, costo del diseño y consumo de corriente, ya que el AD974 es de 24 patas, con comunicación serial y con cuatro entradas analógicas, quedando sobrado para lo que se requiere.

3.3.3.2 Distribución de los pines del PIC

La asignación de los pines del PIC para el manejo de los diferentes dispositivos periféricos se ha hecho de la manera más ordenada posible, esto pensando en facilitar la programación y el diseño de la placa del circuito impreso. En la figura 3.8 podemos ver la conexión básica del PIC para que funcione y la distribución de los pines, para el manejo de los diferentes componentes periféricos. El PIC tiene un cristal de 8MHz, el reset (MCLR) se pone a un nivel alto directamente al no ser necesario reiniciar el PIC constantemente.

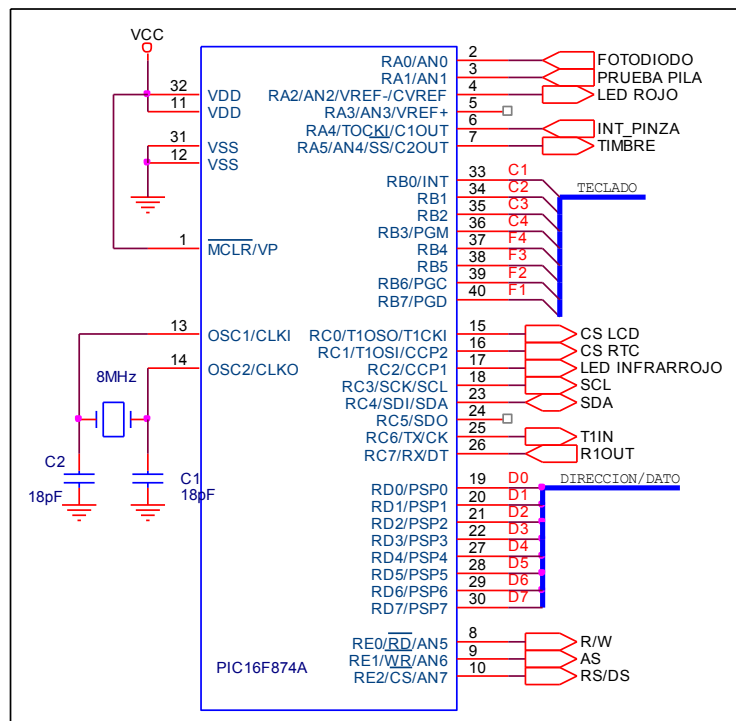


Figura No. 3.8 Distribución de pines del PIC.

3.3.4 EEPROM serial

La conexión de este componente se realiza como se muestra en la figura 3.9, es muy sencilla ya que consta de sólo dos cables.

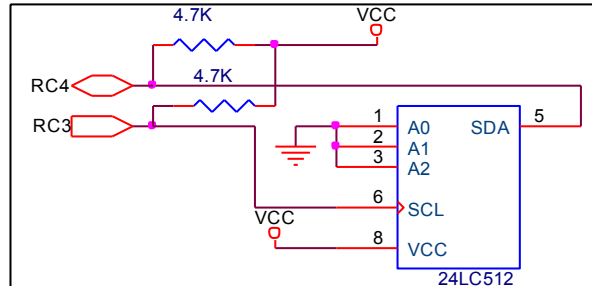


Figura No. 3.9 Conexión de la EEPROM serial.

3.3.5 Reloj en tiempo real (RTC)

Cuenta con calendario, reloj, alarma y además con un espacio para 50 localidades de RAM para uso general. Se utiliza para tener la hora y la fecha actual al momento de que se realiza la medición, ya que esta información se almacenará junto con otros datos. Específicamente lo que se almacena son las horas, minutos, día del mes y el mes.

3.3.5.1 Diagrama de conexión

El bus de direcciones/datos, está implementado igual que el LCD, con AS distingue si es dato o dirección lo que se le envía. DS funciona como habilitación de salida.

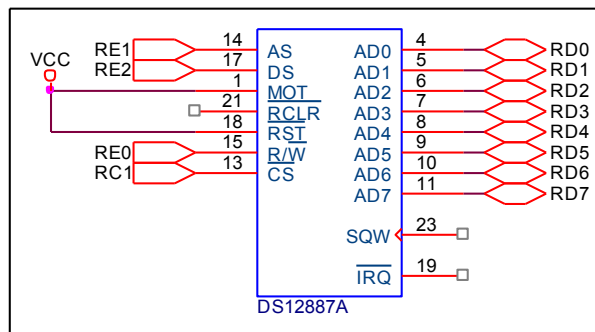


Figura No. 3.10 Conexión del reloj en tiempo real.

3.3.5.2 Mapa de direcciones

La memoria podríamos decir que se encuentra seccionada en dos partes principalmente, como se puede observar en la figura 3.11, de la localidad \$00 hasta la \$0D, para las funciones que realiza y de la \$0E a la \$3F como RAM de uso general.

Para obtener la hora actual y/o la fecha sólo basta con leer la localidad de memoria correspondiente al parámetro que se desea obtener, se tiene que realizar varias lecturas para obtener por ejemplo la hora completa, es decir, primero se leería la localidad \$04 que corresponde a la de las horas, luego los minutos y los segundos.

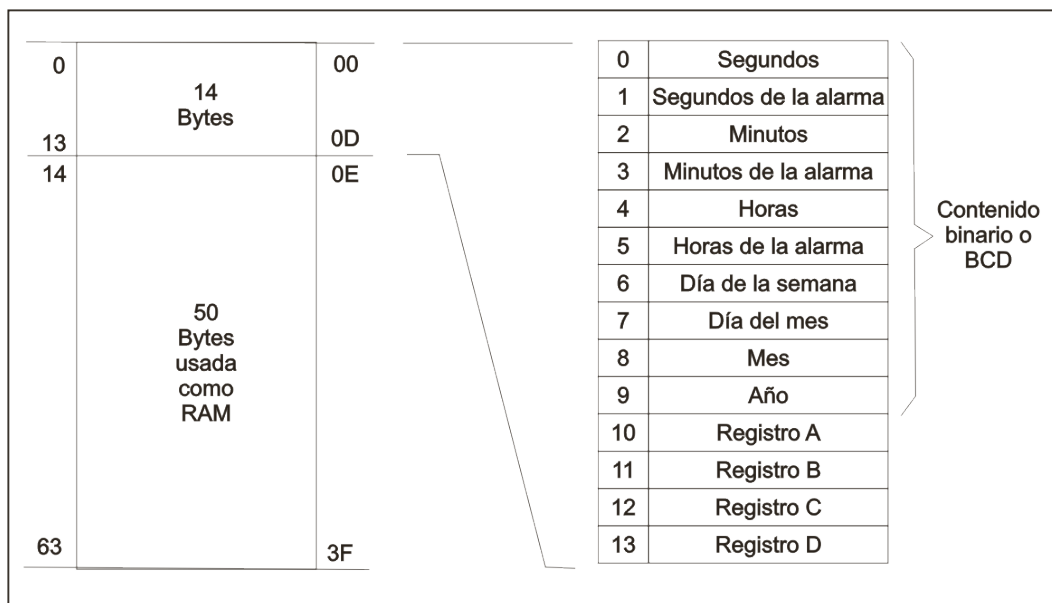


Figura No. 3.11 Mapa de memoria del reloj en tiempo real.

Las localidades destinadas para almacenar la fecha y la hora pueden configurarse para un formato binario o BCD. En la tabla 3.1, se muestra un ejemplo completo, de la forma en que podemos representar una fecha y hora en diferentes formatos.

Tabla No. 3.1 Hora, calendario, fecha y alarma en diferentes modalidades.

Dirección	Función	Rango en decimal	Rango		*Ejemplo	
			Dato en modo binario	Dato en modo BCD	Dato en modo binario	Dato en modo BCD
0	Segundos	0-59	\$00-\$3B	\$00-\$59	15	21
1	Segundos de la alarma	0-59	\$00-\$3B	\$00-\$59	15	21
2	Minutos	0-59	\$00-\$3B	\$00-\$59	3A	58
3	Minutos de la alarma	0-59	\$00-\$3B	\$00-\$59	3A	58
4	Horas (Modo de 12Hrs.) (Modo de 24Hrs.)	1-12	\$01-\$0C(AM) y \$81-\$8C (PM)	\$01-\$12(AM) y \$81-\$92 (PM)	05	05
		0-23	\$00-\$17	\$00-\$23	05	05
5	Horas de la alarma (Modo de 12Hrs.) (Modo de 24Hrs.)	1-12	\$01-\$0C(AM) y \$81-\$8C (PM)	\$01-\$12(AM) y \$81-\$92 (PM)	05	05
		0-23	\$00-\$17	\$00-\$23	05	05
6	Día de la semana Domingo = 1	1-7	\$01-\$07	\$01-\$07	05	05
7	Día del mes	1-31	\$01-\$1F	\$01-\$31	0F	15
8	Mes	1-12	\$01-\$0C	\$01-\$12	02	02
9	Año	0-99	\$00-\$63	\$00-\$99	05	05

*Ejemplo: 5:58:21 AM, jueves 15 de febrero de 2005.

Para actualizar algún dato sólo se necesita escribir en la localidad donde se encuentra el dato a modificar y automáticamente el dispositivo continúa funcionando con el nuevo valor, pero hay que comprobar que el dato se encuentre dentro del rango, ya que no cuenta con ninguna protección contra valores erróneos, por ejemplo si está en formato binario y se escribe un \$50 en los segundos, esto ocasiona que se siga incrementando cada segundo normalmente, más como se pasó de los 60, no mide un segundo, éste se incrementaría hasta 255 y regresa a cero entonces ahora si, cuando llegue a 60 se resetea a cero el mismo e incrementa los minutos, esto mismo sucede con la fecha.

Este dispositivo se debe de configurar al momento de iniciar por primera vez, ya que no se encuentra habilitado ningún divisor del oscilador interno. Esto lo retomaremos en el programa en donde se explica en cual registro y que valor escribir.

3.3.6 Comunicación Serial RS232

Para la comunicación del instrumento con una PC, se utiliza el puerto serie, y

la figura 3.12. Además el conector de la computadora es un DB9 hembra, mientras que el conector que va al instrumento es un conector estéreo, esto con el fin de ahorrar el espacio que requeriría una conector DB9.

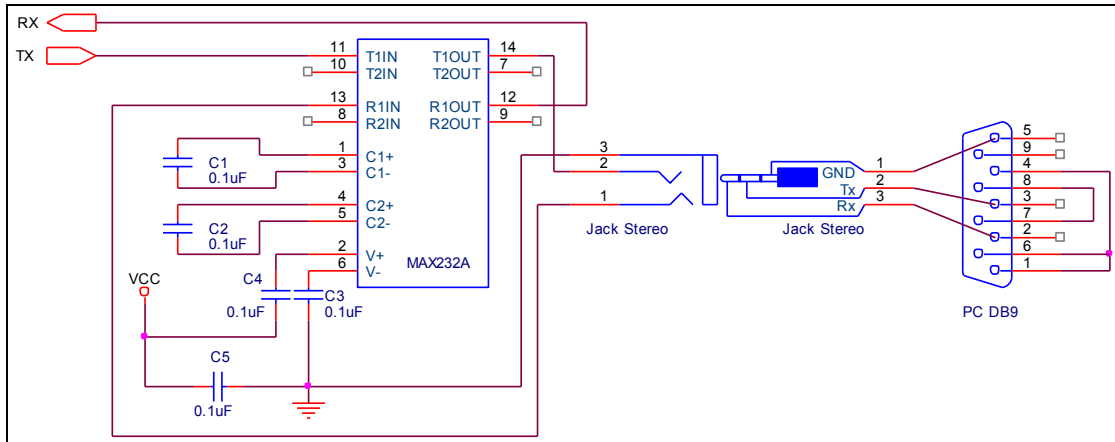


Figura No. 3.12 Conexión del MAX232 y el conector a la PC.

3.3.7 Medidor de carga de la pila

Para medir la carga de la pila se tiene que monitorear el voltaje principal, es decir, el que proviene directamente de la pila, el dispositivo se alimenta con una pila de 9V y el voltaje de referencia del convertidor es de 5 V, por eso se necesita bajar este voltaje para medirlo, por ello usa un divisor de voltaje como se muestra en la figura 3.13, el amplificador operacional es para acoplar impedancias con la entrada AN1 del ADC. El voltaje obtenido a plena escala es de 4.5 V, cuando se tiene una 100% de la carga, pero el voltaje de la pila no tiene que bajar a menos de 7 V, porque la salida del regulador de voltaje (LM7805) no puede mantener el voltaje de 5 V y esto podría afectar las mediciones.

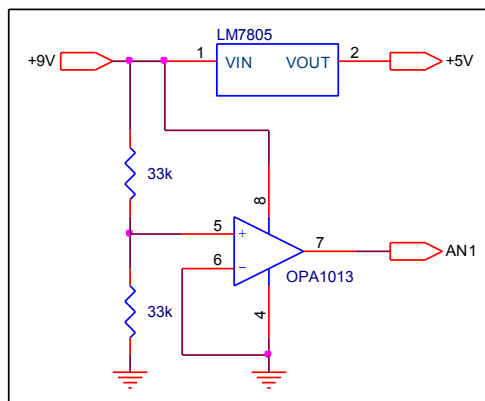


Figura No. 3.13 Medidor de carga de la pila

3.4 Interfaz con el usuario

3.4.1 Teclado matricial

Por medio de un teclado matricial de 4x4, conectado en el puerto B del PIC, se manejará el menú, las opciones y se podrán introducir directamente números o letras, ya que algunas teclas tienen hasta cuatro funciones en cierta parte de los formularios del sistema como se muestra en la siguiente figura.

1 ^{H+} ABC	2 ^{Supr} DEF	3 GHI	
4 JKL	5 MNO	6 PQR	ENTER
7 STU	8 VWX	9 YZ_	
▼	0 [*] -.#	▲	C

Figura No. 3.14 Funciones del teclado.

Cuando se pide introducir el cultivo que se está midiendo, el teclado se convierte en alfanumérico, por ejemplo al presionar la tecla 1, una vez, será una A, pero al volver a oprimirla sin presionar otra tecla será una 'B', al igual la 'C' y luego un '1' y así permanece en un círculo.

Las teclas Derecha/Arriba, Enter e Izquierda/Abajo nos permiten movernos por el menú y seleccionar alguna opción.

La tecla Cancelar funciona cuando se encuentra en alguna opción del menú y desea salir rápidamente y detener la operación.

Como se puede observar en la siguiente figura la conexión de teclado al PIC, ocupando todo el puerto B, no tiene una señal de interrupción cuando se presiona una tecla, como se acostumbra implementar en algunos casos, lo usaremos por medio de sondeo, es decir, se estará constantemente leyendo el nible más significativo (NMS) del puerto para detectar cambios y cuando se dé, significará que se presionó una tecla. En el nible menos significativo (NLS) se hacen corrimientos con cero, esto permite que cuando se presione una tecla se refleje

en uno de los bits del NMS del puerto, por tanto, todos los bits del NMS están en uno lógico, excepto la columna correspondiente a la tecla presionada.

El PIC permite poner en PULL-UP todo el puerto B, no siendo necesario colocar resistencias externas. El PIC no requiere de hacer otra actividad, mientras se espera que elija una opción el usuario por eso se puede hacer por sondeo. La figura 3.15 muestra el acomodo de las teclas y la conexión al PIC.

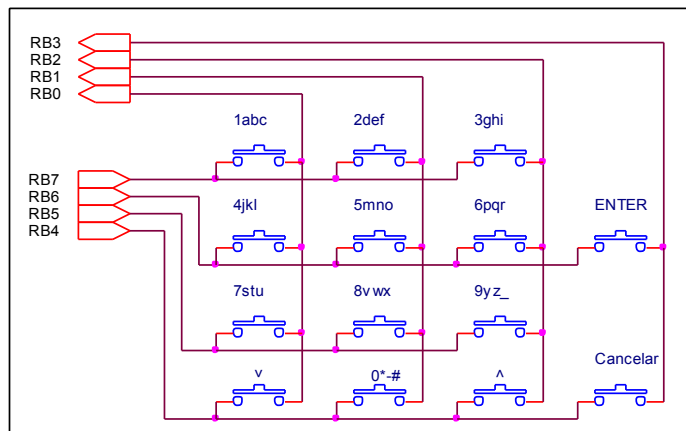


Figura No. 3.15 Teclado matricial.

3.4.2 Pantalla de cristal líquido (LCD)

Se utilizó un LCD para visualizar el menú del aparato y visualizar las mediciones que se realicen.

El LCD dispone de una matriz de 5x8 puntos para representar cada carácter. En total se pueden representar 256 caracteres diferentes. 240 caracteres están grabados dentro del LCD y representan las letras mayúsculas, minúsculas, signos de puntuación, números, y demás.

Dispone de dos líneas con 16 caracteres cada una. El bus de datos/direcciones está conectado al puerto D del PIC, ya que se utilizará el LCD en configuración con bus de datos de 4 bits. Para la entrada Vee, la cual sirve para ajustar el contraste del LCD, se conecta a tierra con una resistencia de 330 Ω , con la cual se obtenga el contraste deseado en la pantalla. La tensión nominal de alimentación es de 5 V, con un consumo menor de 5 mA.

En la figura 3.16 se puede observar como está conectado el LCD con el PIC. El bus de control está formado por 3 señales: *RS*, *R/W* y *E*.

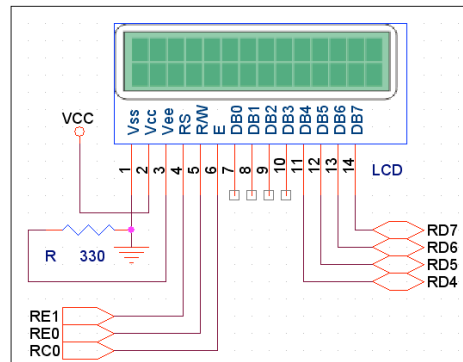


Figura No. 3.16 Conexión del LCD.

La señal *E* es la señal de validación de los datos. Cuando no se utiliza el LCD esta señal debe permanecer a 0. Sólo en las transferencias de información (lecturas o escrituras) es cuando se pone a nivel 1 para validar los datos, pasando después de un tiempo a nivel 0.

La señal *R/W* permite seleccionar si la operación que se va a realizar sobre el LCD es una lectura o una escritura. Cuando $R/W=1$ se realizan lecturas y cuando $R/W=0$ escrituras.

Con *RS* (Register Select) se selecciona el registro interno del LCD sobre el que se va a leer/escribir. El LCD dispone de dos registros internos: Registro de control y registro de datos. Ambos registros son de lectura y escritura. $RS=0$ selecciona el registro de control. $RS=1$ el registro de datos.

3.5 Diseño del programa para el PIC

Primero se mencionarán algunas herramientas utilizadas en el desarrollo del programa y posteriormente veremos algunos diagramas a bloques de las funciones principales. El listado completo del programa se encuentra en el apéndice 3.

3.5.1 Herramientas computacionales

3.5.1.1 Matemática

MatLab es la herramienta matemática empleada para el análisis de los resultados

empleó la función *polyfit()* para encontrar el polinomio que permite ajustar los valores obtenidos con el instrumento a los valores de nuestro patrón de referencia, este proceso se explica más adelante.

3.5.1.2 Compilador en C

Existen diversas formas para programar el PIC, la principal es en ensamblador, así fue como se inició el programa, pero al ir agregando más funciones se comenzó a volver más complicado y previendo que se tendría que realizar operaciones de punto flotante, se optó por utilizar un compilador para PIC, con programación en lenguaje C. Es una gran ventaja, obviamente al ser un lenguaje de más alto nivel, hay que tener cuidado con la memoria, ya que siempre existen varias formas de hacer una misma operación. Al no ser un factor a considerar la rapidez de ejecución del programa, se trata de adecuar el código para consumir lo menos posible de la memoria de programa.

El programa se llama PCW C Compiler versión 3.34, en la figura 3.16 muestra una captura de pantalla del programa. Contiene algunas librerías, para manejar algunos dispositivos como: un teclado matricial, un LCD, un convertidor de analógico a digital, una E²PROM serial, entre otros dispositivos periféricos.

Esto tiene una gran ventaja principalmente en ahorro de tiempo y flexibilidad al programar.

```

49 char CONST str_menu[9][14] =
50 { {'C','a','l','i','b','r','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
51 {'M','e','d','i','c','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
52 {'V','e','r','i','f','i','c','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
53 {'E','s','t','a','d','o',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
54 {'B','o','r','r','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
55 {'C','o','n','f','i','r','m','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
56 {'V','e','r','i','f','i','c','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
57 {'A','j','u','s','t','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
58 {'A','j','u','s','t','a','r',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},
59 };
60
61 byte CONST F2[10] = {'*', 'A', 'D', 'G', 'J', 'M', 'P', 'S', 'V', 'Y'};
62 byte CONST F3[10] = {'-', 'B', 'E', 'H', 'L', 'N', 'Q', 'T', 'W', 'Z'};
63 byte CONST F4[10] = {'#', 'C', 'F', 'I', 'K', 'O', 'R', 'U', 'X', ' '};
64 byte CONST P1[8] = {1,1,0,1,1,0,1,1};
65 byte CONST P2[6] = {1,1,1,1,1,1};
66
67 #define max_menu 8
68 #define MSG1 "Nueva hora:"
69 #define MSG2 "Nueva fecha:"
70 #define HRS " Hrs."
71 #define ERROR1 "Hora incorrecta"
72 #define ERROR2 "Fecha incorrecta"
73 #define LIMPIAR '\f'
74 #define SALTO '\n'
75 #define BLINK 0x0F
76 #define NO_BLINK 0x0C
77 #define DELAY_REP 20000

```

Figura 3.17 Captura de pantalla del PCW C Compiler.

3.5.1.3 Simulación

Mientras se fue desarrollando el programa fue necesario ir realizando pruebas, para ello se apoyó en el programa Proteus ISIS Professional, el cual puede realizar simulación y maneja diferentes modelos de PIC's, en la figura 3.18 se ve una captura del programa.

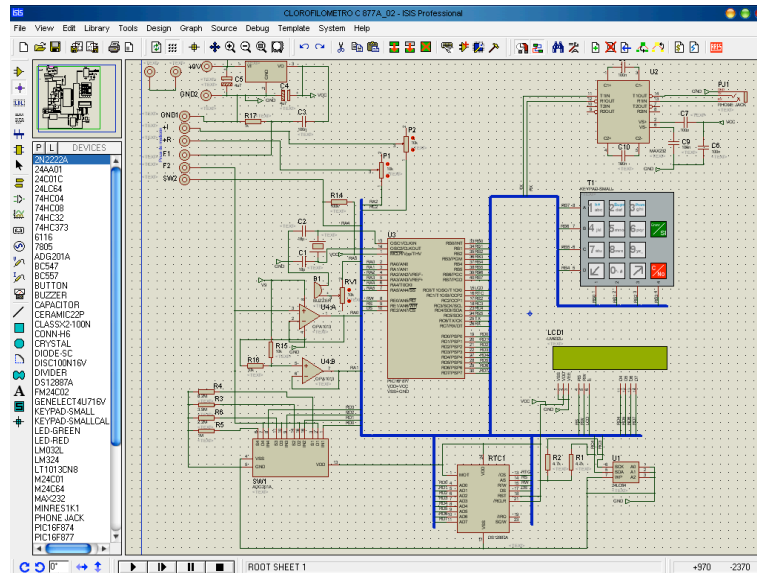


Figura 3.18 Captura de programa Proteus ISIS Professional.

Sólo se logró programar parte del sistema, ya que no cuenta con reloj en tiempo real, pero se hizo un arreglo con compuertas y un SRAM, para asemejar el funcionamiento del mismo, con esto se facilitó la elaboración de las rutinas para su manejo. Y obviamente no se puede simular la etapa óptica.

Básicamente en todas las etapas del desarrollo del programa fue de gran utilidad ya que a pesar de las limitantes descritas anteriormente, se pudo monitorear el comportamiento de los puertos, como por ejemplo para verificar alguna secuencia en específico, pero sobre todo fue muy útil en la etapa de despliegue y de interacción con el usuario.

3.5.1.4 Grabación del programa en el PIC

Para este propósito se empleo el software *IC_Prog v.1.05A*, el cual es un software con capacidad de manejar diferentes hardware de programación serial. El modelo del programador que se armó fue el *jdm programmer*, el diagrama esquemático lo puede consultar en el apéndice 1. El compilador en C genera un archivo .HEX. el cual es un archivo hexadecimal v se carga directamente al

programador y contiene la palabra de configuración para el PIC, el programa y los datos para EEPROM interna, si los hubiera. Una captura del programa se muestra a continuación.

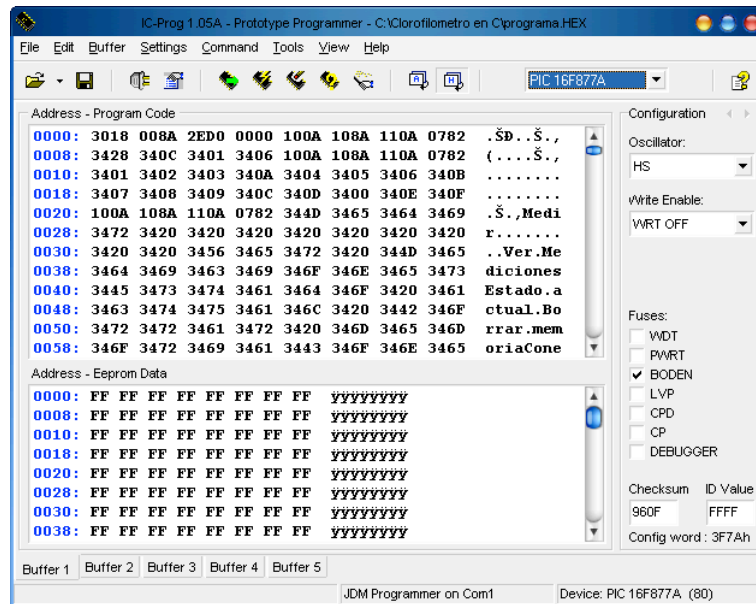


Figura No. 3.19 Captura de pantalla del programa IC Prog.

3.5.2 Diseño de la estructura de datos

Uno de los puntos principales en el desarrollo del programa fue el asignar una estructura a los datos que se almacenarán en la memoria EEPROM, para ello se hizo un análisis de nuestras necesidades de almacenamiento, para ello se consideraron los siguientes aspectos:

- Identificador del cultivo: Este nos sirve para identificar fácilmente a que cultivo se le realizó la medición, se obtuvo información de primera mano de la persona que realiza muestreo de plantas y mencionó que con seis caracteres alfanuméricos es suficiente para identificar el cultivo.
- Hora y fecha: Sólo se almacenará del horario, horas y minutos y de la fecha el día del mes y el mes, con un byte cada uno, tenemos un total de 4 bytes.
- Número de hoja: En la toma de las muestras siempre se hacen más de una medición por hoja, por ello se incluye un campo, para indicar la hoja a la que se le realizó la medición, con un rango de 1 a 255 hojas.
- Número de muestra: Es importante este campo para facilitar el manejo e interpretación de los datos obtenidos de la toma de las muestras, se puede

- Valor del rojo e infrarrojo: Lo que se almacenará será el valor obtenido del convertidor de analógico a digital, por tanto para cada led se necesitara dos bytes. Cuando se desea mostrar la concentración de clorofila se necesita aplicar operaciones matemáticas.

Ahora ya sabiendo lo que se quiere se forma la estructura, y como ya se sabe los rangos que se manejarán es muy sencillo designar el tipo de dato. La siguiente tabla 3.2 nos muestra la estructura.

Tabla No. 3.2 Estructura de los datos.

Posición	Descripción del campo	Tamaño (byte)
0	Clave del cultivo (cultivo)	6
6	Número de Hoja (h)	1
7	Número de muestra (m)	1
8	Día	1
9	Mes	1
10	Horas	1
11	Minutos	1
12	Valor del Infrarrojo (LInfra)	2
15	Valor del rojo (LROjo)	2
TOTAL		16

Ya que se tuvo el tamaño que ocupa cada medición, fue posible determinar que el instrumento es capaz de almacenar 4094 mediciones.

3.5.3 Rutinas principales

Se realizará un diagrama de flujo de las rutinas principales, buscando dejar un bosquejo general del procedimiento que se sigue durante la navegación dentro del menú del sistema y durante la toma de las muestras.

3.5.3.1 Menú del sistema

void main()

Al iniciar, llama a la rutina de inicialización del PIC y los periféricos. La rutina principal permite navegar por el menú principal del sistema, en cual permanece en un ciclo infinito, esperando una orden a través del teclado, ya sea para cambiar de opción del menú o para entrar en una de ellas.

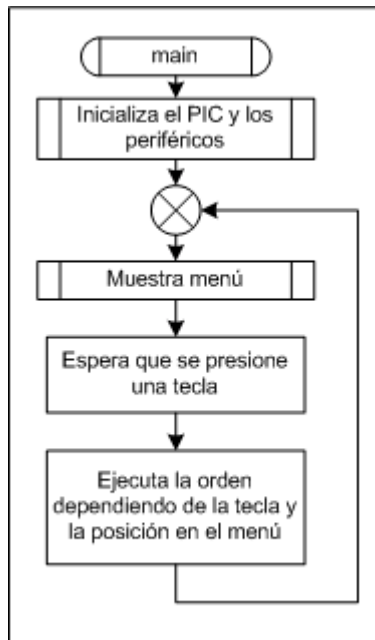


Figura No. 3.20 Diagrama a bloques de la función principal.

3.5.3.2 Rutinas para el proceso de medición

void opt_medir()

Esta rutina lleva a cabo el control del proceso de la toma de muestras y del almacenamiento de las mismas en la memoria EEPROM. Comprueba que el instrumento esté calibrado, de no estarlo se llama a la función para calibrar, la manera en que sabe si está calibrado es por medio de una variable global que se inicializa en menos uno cuando empieza a funcionar el PIC, y una vez que se calibra la variable contiene un número mayor a 0.

Una vez listo para medir el dispositivo dos, da la opción borrar la última medición (si existiera), incrementar el número de hoja (si ya se ha tomado al menos una muestra de la actual) y también nos da la opción de cancelar y no almacenar las mediciones realizadas.

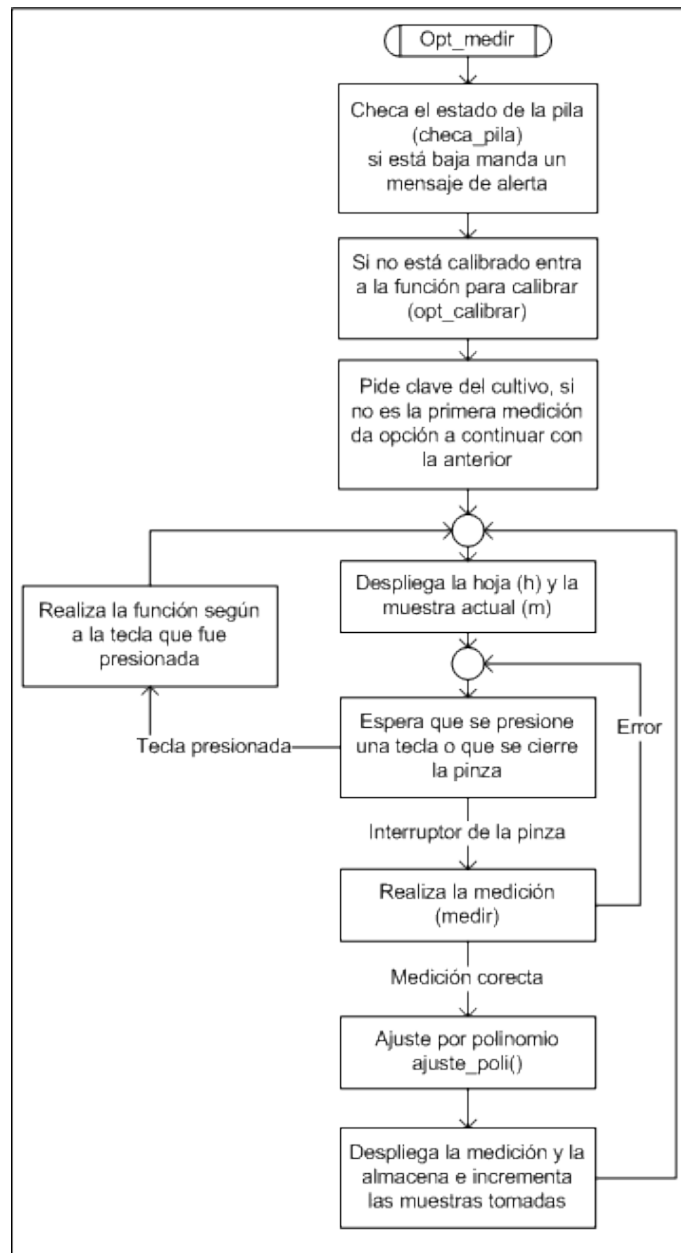


Figura No. 3.21 Diagrama a bloques del proceso de medición.

signed byte opt_calibrar()

En esta función se obtiene los factores de calibración tanto del led infrarrojo como del rojo. ¿Por qué es necesario esto?, hay diversas condiciones que podrían hacer que varíe la intensidad de los leds o la que le llega al FDS y con los factores se hace un ajuste de ganancia para tener los mismos que los de un inicio. Algunas condiciones que se podrán compensar son: ventana de medición sucia, deterioro de los componentes, y otras como la luz muy intensa del sol, humedad, temperatura, pero de estos últimos sólo se puede hacer una aseveración teórica

ya que no se realizaron pruebas para comprobarlos. Realmente es muy sencillo el proceso, como se puede apreciar en el siguiente diagrama a bloques.

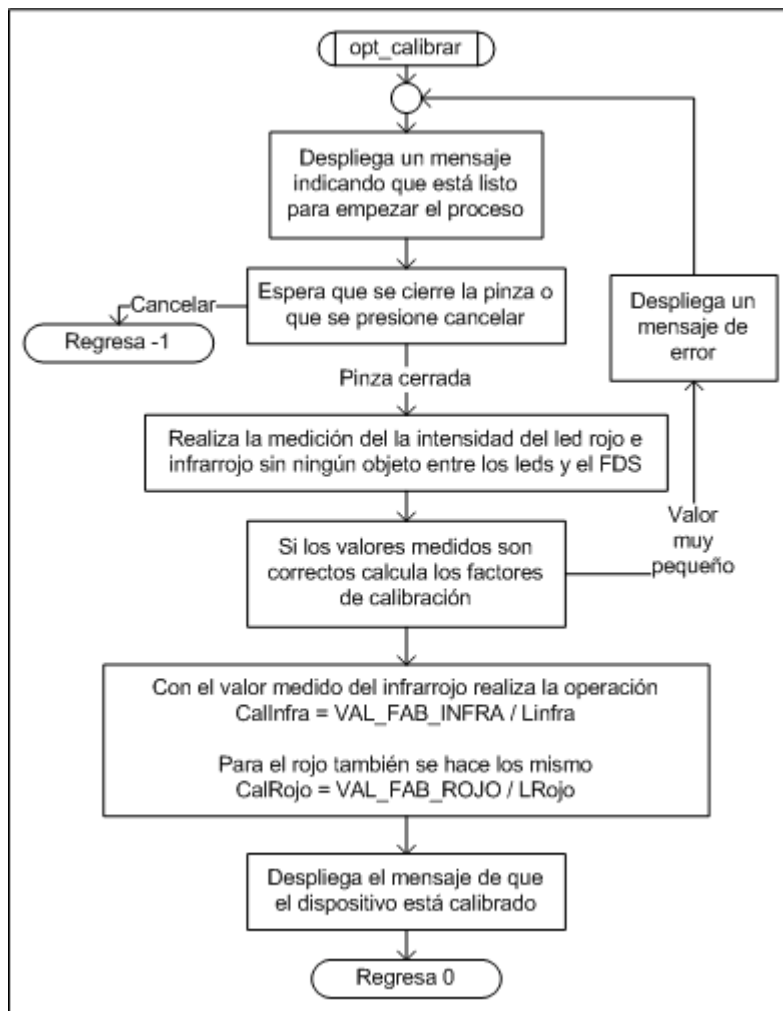


Figura No. 3.22 Diagrama a bloques del proceso de calibración.

Lo que se obtiene son dos factores que se estarán multiplicando con las mediciones que se realicen con el led rojo e infrarrojo, para obtenerlo se mide la intensidad del infrarrojo sin ningún obstáculo al igual con el rojo, ahora para obtener el factor, en el programa se le introduce un valor constante llamado valor de fábrica (VAL_FAB_INFRA), el cual es el valor que presentó el infrarrojo cuando se realizó la caracterización y se obtuvo el polinomio de ajuste, para el rojo también se tiene un valor de fábrica (VAL_FAB_ROJO). Para obtener el factor se divide el valor de fábrica entre el valor obtenido del rojo o infrarrojo según corresponda.

signed byte medir()

En esta rutina se llevan a cabo las operaciones para controlar el proceso de la toma de una muestra. Primero se coloca en la primer escala (cero) a través del puerto D del PIC. La función `signed int16 leer_adc_prom(byte led)` realiza varias mediciones con el infrarrojo y elimina el *offset* que existe en ausencia de la luz de cualquiera de los dos leds, con `led=1` realizará la medición con el infrarrojo y con 2 lo hace con el rojo. De abrirse la pinza en este proceso regresa un “-1”.

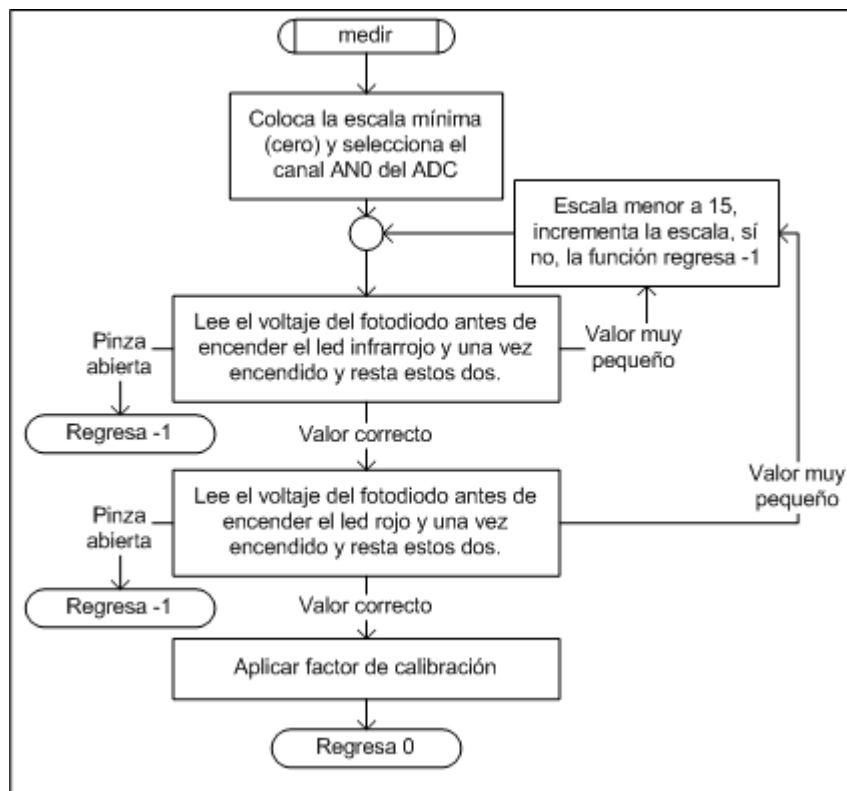


Figura No. 3.23 Diagrama a bloques del proceso para obtener la relación del rojo con respecto al infrarrojo.

Realiza primero la medición con el infrarrojo y si el valor que se obtiene del ADC está por debajo del mínimo incrementa la escala y la resistencia equivalente de polarización del fotodiodo va incrementándose, hasta llegar a un valor validado o al final de la escala y regresa un “-1”, como error. De tenerse un valor válido se realizará la medición ahora con el rojo que sigue el mismo proceso que el infrarrojo, pero ahora los dos estarán encendiéndose hasta tener un valor válido del rojo o hasta sobrepasar la escala máxima en donde también regresará “-1”. Si los valores obtenidos tanto del infrarrojo como del rojo son válidos, el resultado se

función se multiplican tanto LRojo, como LInfra por su factor de calibración y la función regresará un “0” indicando una medición correcta hasta este punto.

signed byte leer_adc_prom(byte led)

Mide la intensidad del infrarrojo si led=1 o del rojo si led=2, pero antes obtiene el *offset* que se encuentra en la salida del amplificador operacional. El *offset* puede ser debido a que se está incidiendo luz al FDS o es propio del circuito.

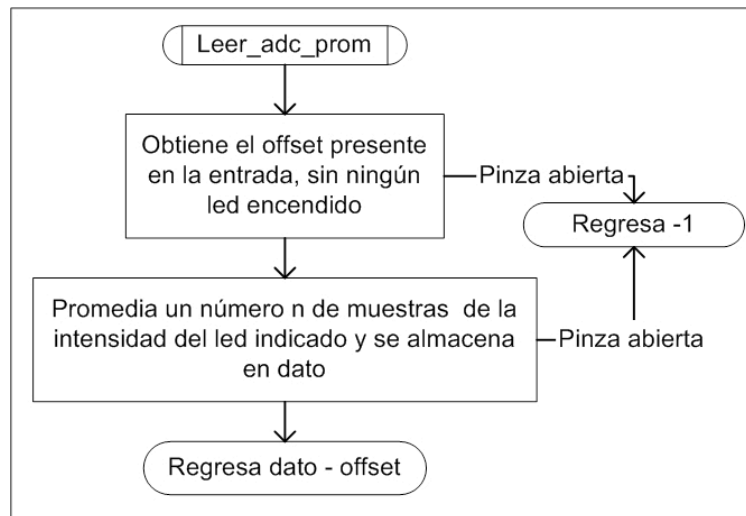


Figura No. 3.24 Diagrama a bloques para obtener la medición promedio de uno de los leds.

Luego toma la medición ya con el led correspondiente y le resta el *offset*. De abrir la pinza durante este proceso inmediatamente regresará un “-1”, en señal de que ocurrió este suceso.

3.5.3.3 Rutina de despliegue de datos almacenados

void opt_ver_datos()

En la rutina se realizan los pasos necesarios para ir desplegando las mediciones que se encuentra almacenada en memoria, como se puede apreciar en el siguiente diagrama a bloques lo primero que se realiza es tomar el número de mediciones almacenadas, es un dato de 16 bits localizado al final de la EEPROM serial, es decir, en la dirección \$FFFE.

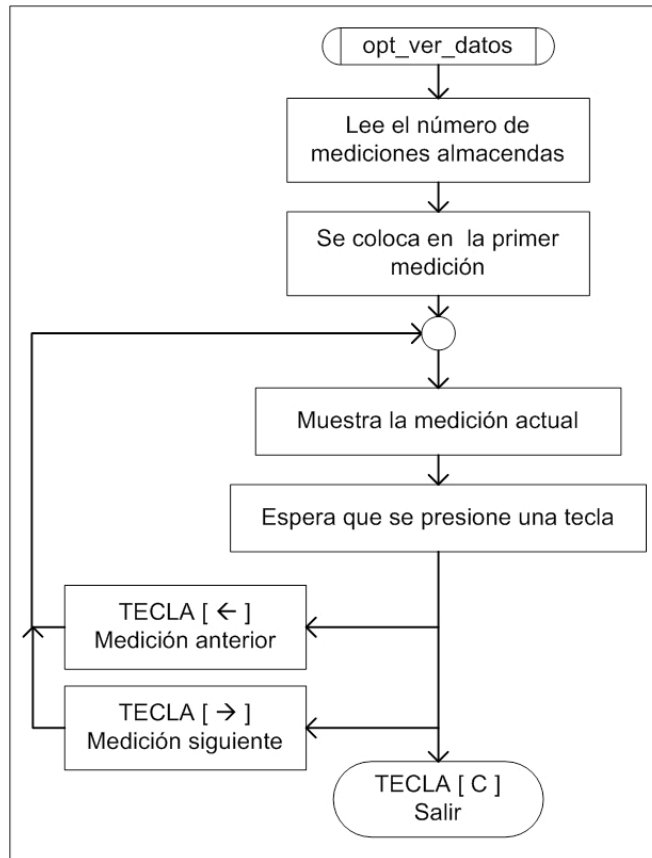


Figura No. 3.25 Diagrama a bloques para ver las mediciones.

3.5.3.4 Rutina para calcular la concentración de clorofila float ajuste_poli()

La manera en que se calcula la concentración de clorofila es obteniendo una relación de transmitancia de la luz infrarroja con la transmitancia tanto de la roja como de la infrarroja, lo cual se hace de la siguiente manera:

$$r_i = \frac{L_{Infra}}{L_{Rojo} + L_{Infra}}$$

Donde: r_i = relación de intensidad del infrarrojo con respecto a la suma de la roja roja e infrarrojo, que corresponde a la variable del polinomio.

L_{Rojo} = transmisión desde el led rojo.

L_{Infra} = transmisión desde el led infrarrojo.

Éste es el primer paso, lo siguiente es aplicar dos polinomios de ajuste. Los valores de relación que se tuvieron son los correspondientes para

concentraciones de clorofila de 0 a 51.2. La aplicación del primer polinomio va desde 1 hasta RI_CAM y el segundo de RI_CAM hasta 0. RI_CAM es una constante que nos sirve para establecer un punto donde se cambiará de polinomio.

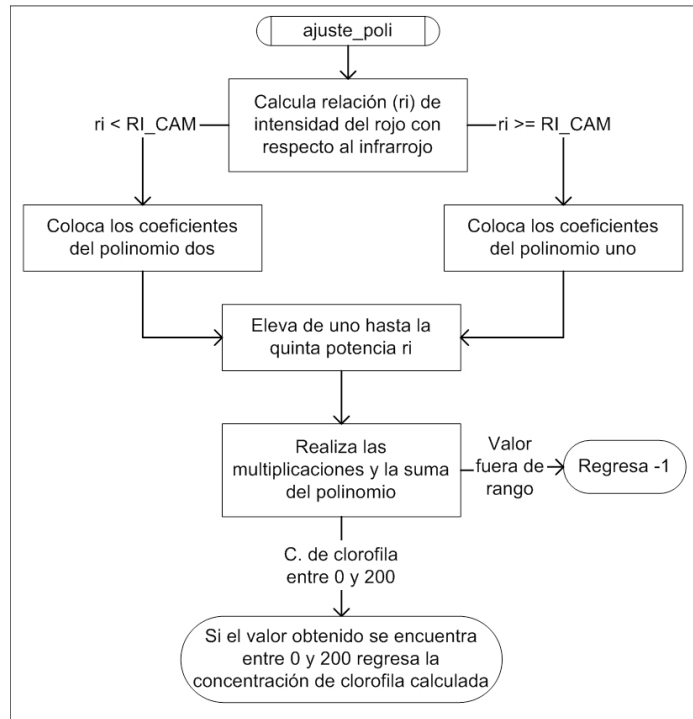


Figura No. 3.26 Diagrama a bloques de la aplicación del polinomio de ajuste.

Se aplican dos polinomios, ya que con uno de quinto orden no es posible lograr un buen ajuste de la curva y en el PIC no fue posible introducir uno de décimo orden. El programa en MatLab para calcular los polinomios se puede consultar en el apéndice III. En el siguiente diagrama se muestran los pasos que se realizan para calcular la concentración de clorofila.

3.6 Interfaz con la computadora

En este apartado veremos el programa de la computadora, para transferir los datos desde el instrumento de medición, la comunicación es a través del puerto serial de manera asíncrona.

El programa fue realizado en Visual Basic 6. El programa no requiere de muchas funciones. El propósito fue hacer es un programa intermedio, entre el instrumento y el programa con el cual se llevará la estadística del estado del cultivo. por ello el

programa cuenta con sólo unas cuantas funciones. La figura 3.27 muestra la captura de pantalla del programa.

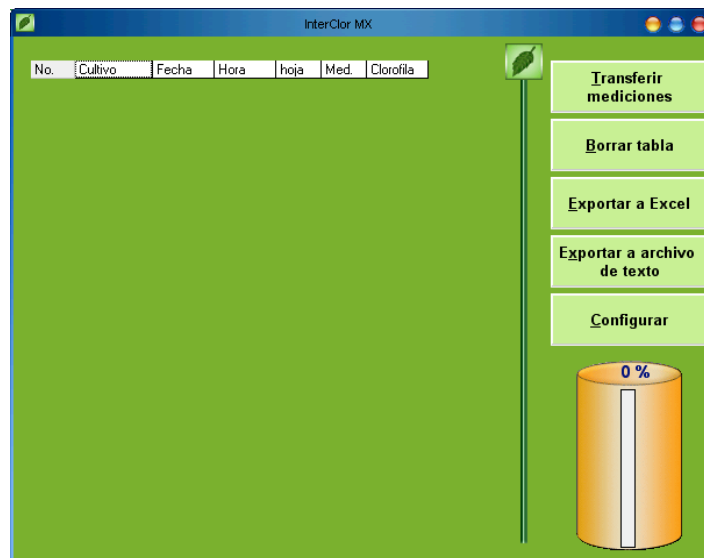


Figura No. 3.27 Captura de pantalla de la interfaz.

Las opciones del programa son las siguientes:

- **Transferir mediciones:** Inicia la transferencia de datos desde el instrumento hacia la computadora. En el instrumento debe de haberse seleccionado la opción del menú “Conectar a PC”, así éste sólo espera la orden desde la computadora para enviar los datos. Al iniciar la transferencia el instrumento le comienza a mandar los datos byte por byte, mientras que la interfaz le regresa los datos recibidos, para que el instrumento los compare y puede saber si ha ocurrido un error.
- **Borrar tabla:** Borra la tabla de mediciones cargadas con anterioridad.
- **Exportar a Excel:** Nos permite transferir todo el contenido de la tabla tal cual a una hoja de Excel, de ser este programa con el cual se llevará el análisis de los datos, de no serlo está la siguiente opción.
- **Exporta a archivo de texto:** Transfiere todo el contenido de la tabla a un archivo de texto sin formato, separada cada columna con tabulación para alinear los datos.
- **Configuración:** Aquí podremos cambiar número del puerto serial por el cual se realizará el enlace con el instrumento. La figura 3.28 nos muestra la ventana que nos aparece con esta opción.

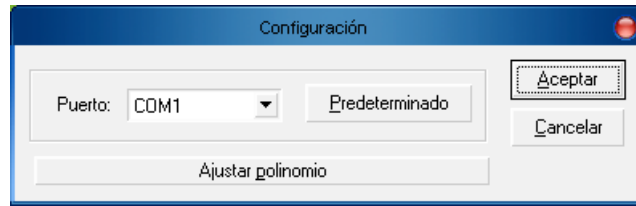


Figura No. 3.28 Captura de pantalla de la opción configuración.

Podemos ver además el botón de Ajustar polinomio, éste nos permite acceder a una ventana como la mostrada en la figura 3.29, en la cual podremos cambiar los valores del polinomio de ajuste, el cual a diferencia del aplicado en el programa del PIC, aquí estamos manejando números de doble precisión y un polinomio de décimo orden. La interfaz recibe los valores binarios de 16 bits de la medición con el led rojo como con el infrarrojo por ello realiza la operación tal como se hace en el PIC, a excepción de que en la computadora es un sólo polinomio.



Figura No. 3.29 Captura de pantalla de la opción de ajustar polinomio.

La barra vertical de color blanco que se encuentra dentro del cilindro, sirve para indicar el progreso de la transferencia, mientras a su vez se indica el porcentaje completado de la transferencia en la parte superior del mismo cilindro.

IV. ANÁLISIS Y RESULTADOS

En este apartado se describen los resultados obtenidos durante el desarrollo del trabajo.

4.1 Situación actual

En términos generales el sistema se encuentra como prototipo. Se ha logrado tener un rango de cero a cincuenta unidades SPAD con una precisión de ± 0.5 unidades SPAD.

El programa del microcontrolador está totalmente funcional, cuenta con las funciones de: medición y despliegue, visualización de datos almacenados, transferencia a la computadora, visualización de la fecha y hora actual, modificación de la fecha y la hora.

El programa de la computadora se utilizó durante las pruebas, para transferir a la computadora las mediciones, y posteriormente analizar, ver el resultado y realizar ajustes.

4.2 Análisis de resultados

En este apartado se describe en análisis de las pruebas realizadas tanto en la programación como en el hardware.

4.2.1 Sujetos de prueba

Las mediciones hechas en la parte final de esta etapa del proyecto arrojaron los resultados que se describen a continuación:

Al principio se utilizaron hojas de plantas con diferentes concentraciones de clorofila, escogidas por sus diferentes grosores y tonalidades de verdes, se tenía que realizar la medición con el SPAD y posteriormente realizarlas con el instrumento en desarrollo. Los problemas para tomar las muestras son: las hojas de las plantas no son uniformes en su superficie, tiene venas, diferentes tonalidades en la misma hoja y debe de coincidir el mismo punto de análisis con los dos aparatos, de lo contrario produce un error y se pierde el patrón de referencia.

Todos estos factores dificultaron mucho las pruebas, por ello se optó por simular diferentes concentraciones de clorofila y grosores de hojas, imprimiendo en hojas blancas bloques con diferentes tonalidades de azul, verde, morado, amarillo entre otros colores, recortando la hoja en tiras como se muestra en la figura 4.1, para facilitar la toma de las muestras. Se logró obtener medición de concentración de clorofila de 0 hasta 52, este rango es suficiente para medir la concentración de clorofila de las hojas a las cuales se les estuvo realizando pruebas y según información proporcionada por el doctor encargado de proyectos de mejoras en el aprovechamiento de los nutrientes para las plantas en el invernadero del ITSON, las hojas de las plantas no están fuera del rango de 0 a 50.

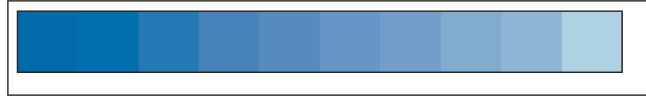


Figura No. 4.1 Tira de muestra de diferentes colores y tonalidades.

4.2.2 Alimentación

El instrumento es portátil, por tanto, será alimentado con pilas, lo cual se hubo que tomar muy en cuenta al momento de realizar el diseño del instrumento, esto por varias razones, la primera es que si no se ahorra lo más posible, en el consumo de corriente, la pila no duraría mucho, es una molestia, ya que lo último que se quiere es estar recargando la pila a cada instante o en el peor de los casos comprando pila nueva.

La pila es de 9V, lo que nos conduce al otro aspecto a considerar, la corriente máxima que puede brindarnos, según se ha observado comercialmente algunas pilas dan como máximo 500mA, lo cual es relativamente poco, si la comparamos con las doble A.

Se escogió una pila cuadrada por el gran ahorro de espacio. Con una sola es posible tener varios niveles de tensión, como se pudo apreciar en el diseño el amplificador operacional está alimentado directamente de la pila, mientras el resto de los componentes lo hace del regulador de 5V.

Aun así, no se alimentó al circuito con pila, una fuente de voltaje del laboratorio hizo su función, por lo cual no es posible hacer una aseveración del comportamiento con la misma. Pero lo que si se puede asegurar es que el circuito no consume más de 35mA, eso asegura de que si pueda ser alimentado con una pila de 9V sin ningún problema.

4.2.3 Determinación de la concentración de clorofila

El primer paso es la toma de la muestra de la hoja por transmitancia, la gráfica de la figura 4.2 muestra la respuesta obtenida a diferentes concentraciones de clorofila, se muestra por separado los valores obtenidos de la medición del led rojo e infrarrojo.

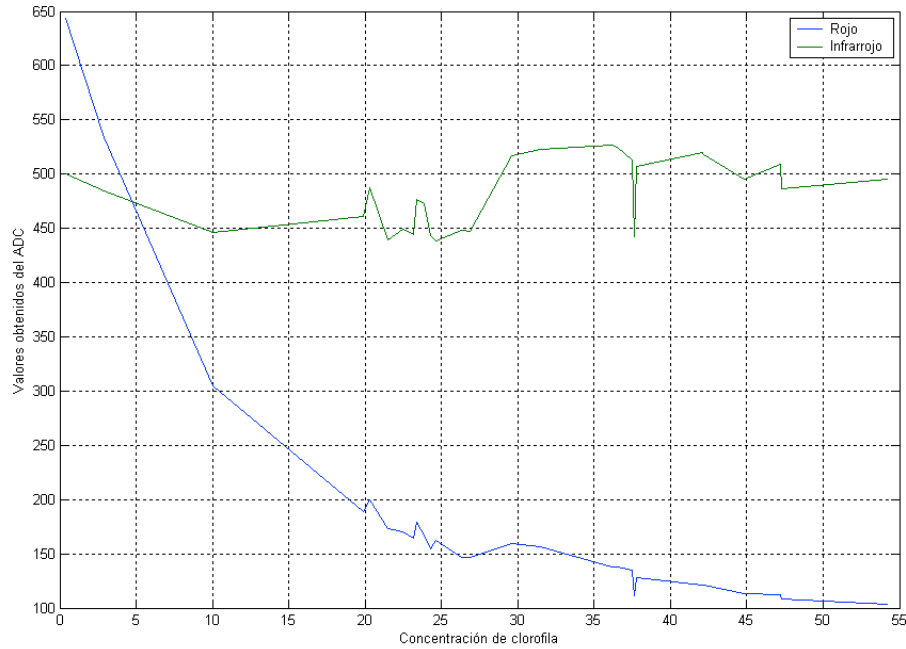


Figura 4.2 Gráfica de los valores obtenidos del ADC del led rojo e infrarrojo.

Como se puede observar a mayor absorción de la luz roja, podemos decir que hay una mayor concentración de clorofila, claro que no necesariamente ya que también está en relación a la absorción de la luz infrarroja, como se menciona en el tema 4.5.3.4. El paso es obtener el porcentaje del infrarrojo con respecto al rojo, como se aprecia en la siguiente gráfica.

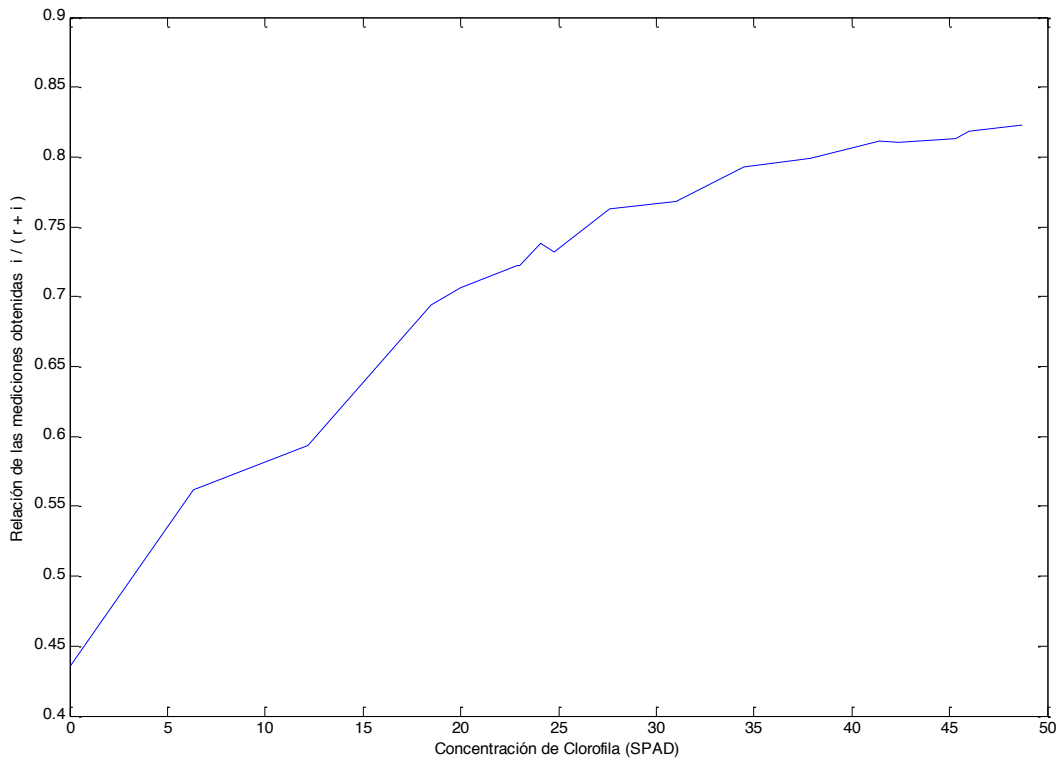


Figura 4.3 Gráfica de la relación del infrarrojo con respecto al rojo

Por último aplicaremos el polinomio de ajuste (ver apéndice 5), para obtener los valores de concentración de clorofila similar al SPAD (patrón de prueba). La gráfica muestra la respuesta del SPAD contra el medidor de concentración de clorofila desarrollado.

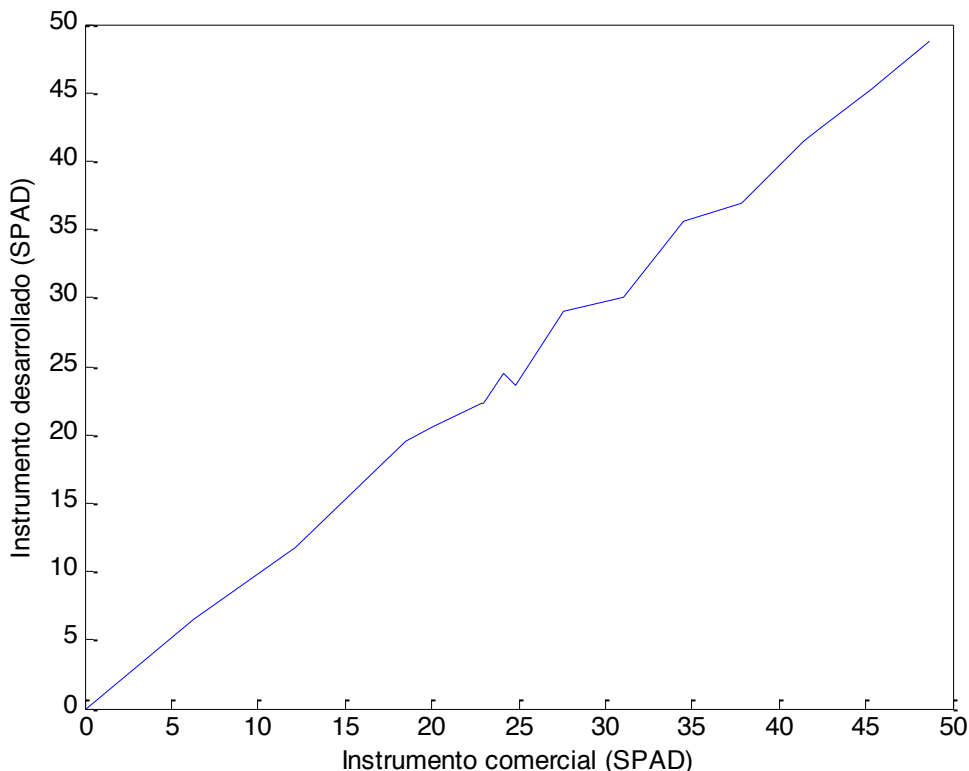


Figura 4.4 Gráfica del medidor de concentración de clorofila comercial contra el desarrollado.

Como se mencionó al principio de este capítulo el error que se tiene en el rango es de ± 0.5 , repetición de ± 0.2 unidades SPAD.

Cuando se comenzó a hacer pruebas se utilizó un fototransistor, se comportó adecuadamente con buena sensibilidad y ganancia, pero el principal problema era una corriente que oscuridad alta. Se eliminaba por software, pero por la sensibilidad adecuada que debería de tener el elemento no fue posible lograr un buen ajuste, y no ayudó el hecho de que sólo se contaba con una sola escala, es decir, una sola resistencia de polarización, lo cual dificultaba mucho el ajuste de la misma ya que si se hacía muy sensible se saturaba el fototransistor.

Después se cambió el fototransistor por un fotodiodo de silicio y de una escala paso a tener 3. Con estos cambios se logró una mejor respuesta principalmente disminuyó muchísimo la corriente de oscuridad, lo cual a primera instancia aumento el rango de medición y con las tres escalas, fue posible tener mayor control para evitar la saturación del FDS. Pero aun así el ajuste de los tres valores de resistencia no era fácil, el cambio de escala era brusco y en algunos casos se saturaba.

Todas estas pruebas condujeron al circuito de polarización que se tiene actualmente, cuenta con 15 escalas como se vio en el capítulo III. Esta configuración es la que mejor ha resultado. Aun así presenta un problema debido al ruido ambiental cuando se encuentra en la escala 13 y 14, que es cuando el FDS está más sensitivo, y por tanto es más susceptible al ruido, la señal empieza a deformarse ligeramente, tendiendo a figurar como una onda senoidal montada sobre el nivel de voltaje que se tenga y la respuesta del FDS es más lenta.

El sistema fue diseñado para ser lo más amigable posible. Fue diseñado sobre las pruebas, es decir, mientras se realizaba las mediciones para caracterizarlo, se le fueron haciendo cambios para volverlo más práctico y con ello más eficiente.

CONCLUSIONES Y RECOMENDACIONES

El objetivo principal se considera cumplido, se sabía que era necesario establecer una relación entre la absorción de la luz roja con respecto a la infrarroja. Aun existen algunas deficiencias debido a una respuesta inesperada, es decir, para algunas hojas funciona adecuadamente con un error de ± 0.5 , mientras que para otras su respuesta es errónea con respecto al patrón de referencia.

No fue posible determinar a que se debe esto, sólo se pueden hacer algunas sugerencias para quién retome el proyecto, y es que, los led's montados en la pinza de medición, su uso no es el más eficiente, al no contar un difusor de luz adecuado, para hacer un bañado uniforme de los haz de luz, provocando tal vez modificaciones a las características.

Se mejoraría la respuesta obtenida incrementando la intensidad y disminuyendo la sensibilidad, para reducir el ruido a una alta sensibilidad, ya que la intensidad de los leds es muy baja y la sensibilidad del FDS es muy alta, el encontrar una buena relación, tomando en consideración el consumo de corriente, si ésta fuera un factor.

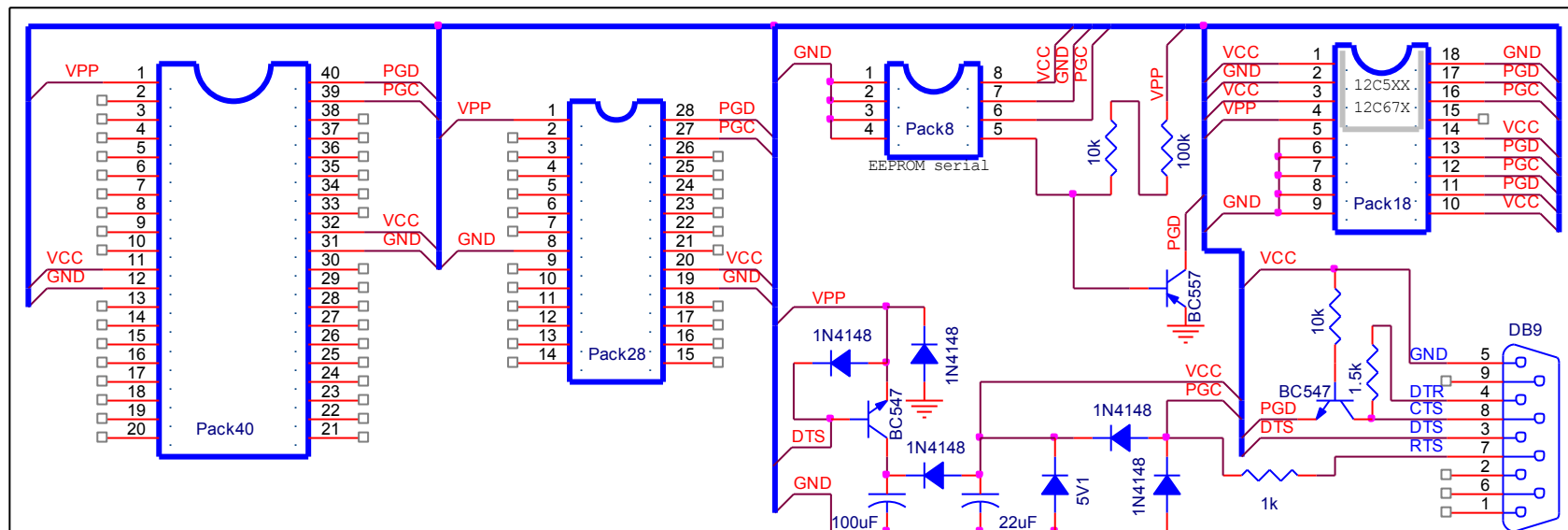
BIBLIOGRAFÍA

1. NOVOA S.-A., Rafael y VILLAGRAN A., Nicolás. **EVALUACIÓN DE UN INSTRUMENTO MEDIDOR DE CLOROFILA EN LA DETERMINACIÓN DE NIVELES DE NITRÓGENO FOLIAR EN MAÍZ.** Agric. Téc.. [online]. ene. 2002, vol.62, no.1 [citado 27 Octubre 2005], p.166-171. Disponible en la World Wide Web: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0365-28072002000100017&lng=es&nrm=iso. ISSN 0365-2807.
2. Valdés Núñez, Graciela Felicia. **DIAGNÓSTICO DE LA CLOROSIS FÉRRICA EN DURAZNERO MEDIANTE EL ANÁLISIS DE HIERRO EN DISTINTOS TEJIDOS,** Memoria de título, Facultad de Ciencias Agronómicas, pagina citada (16) del total de 25, 2004.
3. Padilla Ybarra, Juan José. **MEDICIÓN CONTINUA DE SATURACIÓN DE OXIGENO EN HEMOGLOBINA POR MÉTODO NO INVASIVO.** Tema de sustentación. Cd. Obregón, Sonora, México, 1991.
4. Rubén Hernández, Gil. **FOTOSÍNTESIS.** Material didáctico de la Facultad de Ciencias Forestales y Ambientales. Universidad de Los Andes - Mérida – Venezuela. Página Web: www.forest.ula.ve/~rubenhg, visitada el 22 de febrero de 2005.

APÉNDICE 1
PROGRAMADOR DE PIC'S Y
EEPROM SERIAL

Programador de PIC's y EEPROM serial

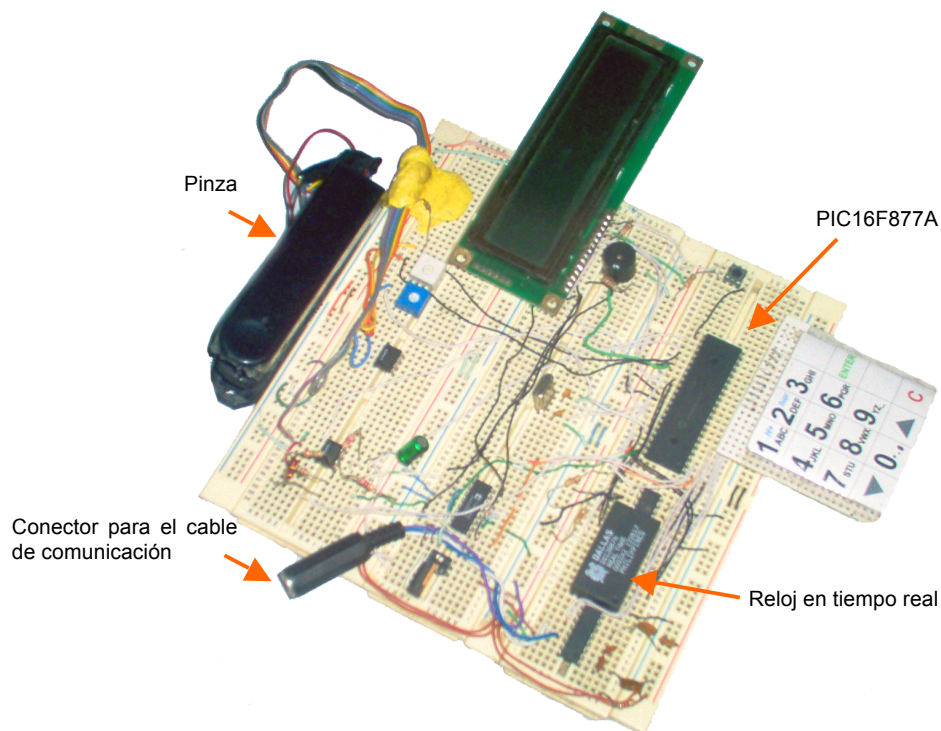
Cuando se escogió el PIC, se buscó un programador adecuado a las necesidades. Se intentó con uno que tiene un software muy básico, no programaba muchos modelos, pero si el PIC16F877A. Se armó y sólo funcionó para el PIC16F84A. Por lo que mejor se buscó uno más completo, que pudiera programar más modelos de PIC's. Se encontró un programador serial en la página http://www.maxoverclocking.com/ele_jdmprog.html, además de PIC's puede programar EEPROM serial. Su software es el IC Prog v.1.05A, el cual está muy completo. Se armó el circuito y hasta el momento no se ha tenido ningún problema una vez que se hizo funcionar. El cable para conectar a la computadora se fabricó, pero puede ser uno comercial. El circuito es el siguiente:



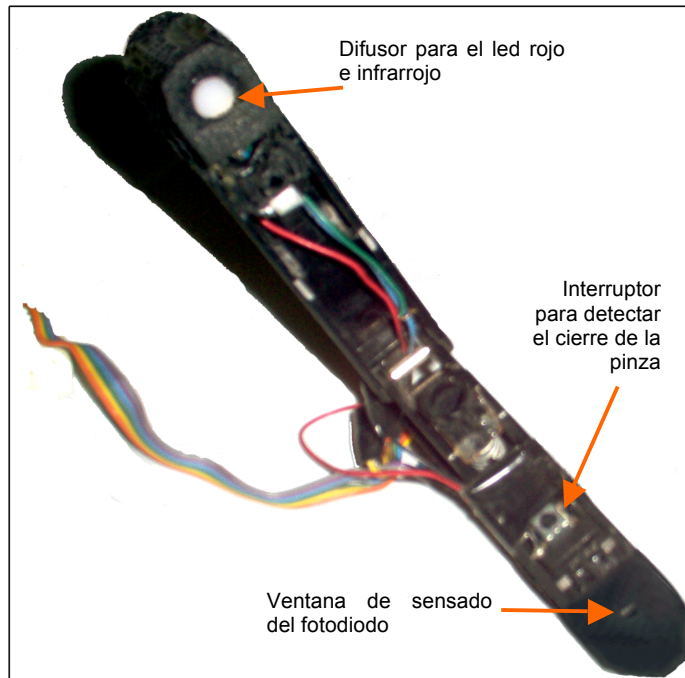
APÉNDICE 2

FOTOS

Vista completa



Pinza



APÉNDICE 3

LISTADO DEL PROGRAMA PARA EL PIC

Programa principal

```

//=====
// La estructura de los datos almacenado en la
// EEPROM es la siguiente:
//
// Posicion  Descripcion bytes
//-----
// 0 Cultivo 6
// 6 hoja 1
// 7 muestra 1
// 8 dia 1
// 9 mes 1
// 10 Horas 1
// 11 minutos 1
// 12 V. d.Infra. 2
// 14 V. d.Rojo 2
//=====
#if defined(__PCM__)
#include <16F877A.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#device *=16 ADC=10
#use delay(clock=8000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#endif
//-----
#include <lcd.c>
#include <Teclado.c>
#include <24LC512.c>
#include <math.h>
#include <rtc.c>
//-----
#byte escalaD = 8 // on to port D (at address 8)
//-----
#define LED_ROJO PIN_A2
#define SW_PINZA PIN_A4
#define BUZZER PIN_A5
#define LED_INFRA PIN_C2
#define ADC_CLK PIN_D0
#define ADC_AD0 PIN_D1
#define ADC_RC PIN_E0
//-----
char CONST str_menu[9][14] =
{{'M','e','d','i','t','a','c','i','o','n','e','s'},
{'V','e','r','M','e','d','i','c','i','o','n','e','s'},
{'E','s','t','a','d','o','a','c','t','u','a','l'},
{'B','o','r','r','a','r','m','e','m','o','r','i','a'},
{'C','o','n','e','c','t','a','r','a','P','C'},
{'V','e','r','h','o','r','a','F','e','c','h','a'},
{'A','j','u','s','t','a','r','H','o','r','a'},
{'A','j','u','s','t','a','r','f','e','c','h','a'},
};
byte CONST F2[10]= {'*','A','D','G','J','M','P','S','V','Y'};
byte CONST F3[10]= {'-','B','E','H','K','N','Q','T','W','Z'};
byte CONST F4[10]= {'#','C','F','I','L','O','R','U','X',''};
byte CONST P1[8] = {1,1,0,1,1,0,1,1};
byte CONST P2[6] = {1,1,1,1,1,1};
//-----
#define max_menu 7
#define HRS " Hrs."
#define LIMPIAR '\f'
#define BLINK 0x0F
#define NO_BLINK 0x0C
#define DELAY_REP 20000
#define DELAY_SW 10000
#define NUM_MED 0xFFFE
#define ESP_RS232 2000
#define PROM 128
#define MaxMed 4094
#define LS 16
#define pos_med 13
#define adc_max 65400
#define val_min_Rojo 50000
#define MIN_ESC_R 100
#define MIN_ESC_I 100

#define MIN_CAL_R 500
#define MIN_CAL_I 200
#define VAL_FAB_INFRA 244
#define VAL_FAB_ROJO 821
//-----
//-----DECLARACIÓN DE VARIABLES GLOBALES-----
//-----
byte cont, menu, horas, minutos, segundos, dia, mes;
byte anno, tranDat;
byte dato, x, h, m, Linea[16], validacion, escala;
int16 nMediciones, n, nMedicionesAnt, Dir, adc_word, i;
int16 datPorTran, contDatTran, LInfra, LRojo;
char k, dato_str[5];
float CalRojo, callInfra;
//=====
// INICIO DEL PROGRAMA //
//=====
====//
// char t_getchar(void)
//
// LEE EL TECLADO HASTA QUE SE PRESIONE
// UNA TECLA
// RECIBE: --
// REGRESA: LA TECLA PRESIONADA
//-----
char t_getchar(void) {
while(TRUE){
k=kbd_getc();
if(k!='X') return(k);
delay_us(500);
}
}
//-----
// float ajuste_poli()
//
// CALCULA LA CONCENTRACIÓN DE CLOROFILA (CC)
//
// RECIBE: LROJO Y LINFRA
// REGRESA: CC
//-----
float ajuste_poli(){
float a,b,c,d,f,fa,fb,fc,fd,ff,ti,med,ri;

ri = (float)LRojo/(LRojo+LInfra);

if(ri > 0.41) {
fa=-16535.4267700572;
fb=49579.7488254241;
fc=-59378.4453662765;
fd=35546.0721624531;
ff=-10736.9014684035;
ti=1344.79830362072;
}
else
{
fa=-607874600.730108;
fb=1103729475.82339;
fc=-800834643.580081;
fd=290242866.784024;
ff=-52543379.6572231;
ti=3801096.63868286;
}

f= ri;
d= f*ri;
c= d*ri;
b= c*ri;
a= b*ri;

med=a*fa + b*fb + c*fc + d*fd + f*ff + + ti;

```

```

if( med < 0 || med > 200) return(-1);
return(med);
}

////////////////////////////////////
// int16 leer_adc(byte canal)
//
// REALIZA UN PROMEDIO DE MEDICIONES DEL ADC,
// UN
// NUMERO DE VECES DADO POR "PROM" CON
// EL LED INDICADO COMO SIGUE:
// Led = 0 promedio con el led infrarrojo
// Led = 1 promedio con el led rojo
//
// RECIBE: LED
// REGRESA: PROMEDIO
////////////////////////////////////
signed int16 leer_adc_prom(byte Led){
int32 suma;
int16 dato, offset;
int1 led_enc;
led_enc=0;
suma=0;
set_adc_channel( 0 );
setup_adc(ADC_CLOCK_INTERNAL );

CON_LED:
delay_ms(15);
for(i=0;i<PROM; ++i){
dato=read_adc();
suma+=dato;
if(input(SW_PINZA))
{ output_low(LED_INFRA);
output_low(LED_ROJO);
setup_adc(ADC_OFF);
return(-1);
}
}
suma=suma/PROM;
dato=suma;

if(led_enc==0)
{ offset=dato;
if( Led == 1 ) output_high(LED_INFRA);
else if( Led == 2 ) output_high(LED_ROJO);
led_enc=1;
goto CON_LED;
}
else
{ output_low(LED_INFRA);
output_low(LED_ROJO);
setup_adc(ADC_OFF);
return(dato-offset);
}
}

////////////////////////////////////
// signed int8 medir()
//
// REALIZA LA MEDICIÓN CON EL LED ROJO E
// INFRARROJO, AJUSTANDO LA ESCALA SEGÚN SEA
// LA ABSORBANCIA PRESENTE EN LA HOJA.
//
// REGRESA: SI LA OPERACIÓN SE REALIZÓ
// CORRECTAMENTE UN 1, DE LO CONTRARIO UN -1.
////////////////////////////////////
signed int8 medir(){
escalaD=0;
escala=0;
aumenta_R;
//-----Medicion Led Infrarrojo
LInfra=leer_adc_prom(1);

if(LInfra== -1) return(-2);
else if(LInfra < MIN_ESC_I)
{ if( escala < 14 ) goto Ajuste;
else return(-1);
}
//-----Medicion Led Rojo
LRojo=leer_adc_prom(2);

if(LRojo== -1) return(-2);
else if(LRojo < MIN_ESC_R)
{ if( escala < 14 ) goto Ajuste;
else return(-1);
}
if(input(SW_PINZA)) return(-2);
LRojo=(float)LRojo*CalRojo;
LInfra= (float)LInfra*calInfra;
return(0);
Ajuste:
escala+=1;
escalaD=escala;
goto aumenta_R;
}

////////////////////////////////////
// void muestra_menu()
////////////////////////////////////
void muestra_menu() {
byte row;

row=menu;
if(menu==max_menu)
{ lcd_putc("\fn>");
--row;
}
else lcd_putc("\f>");
lcd_gotoxy(3,1);
for(i=0;i<28;+i)
{ lcd_putc(str_menu[row][i]);
if(i==13) lcd_gotoxy(3,2);
}
}

////////////////////////////////////
// void lee_hora()
////////////////////////////////////
void lee_hora(){
horas=read_byte_rtc(4);
minutos=read_byte_rtc(2);
segundos=read_byte_rtc(0);
}

////////////////////////////////////
// void lee_fecha()
////////////////////////////////////
void lee_fecha(){
dia=read_byte_rtc(7);
mes=read_byte_rtc(8);
anno=read_byte_rtc(9);
}

////////////////////////////////////
// void despliega_hora()
////////////////////////////////////
void despliega_hora(){
byte K2;
lee_hora();
lcd_gotoxy(1,2);
linea[0]=(horas>>4)+0x30;
linea[1]=(horas&0xf0f)+0x30;
linea[2]=':.';
linea[3]=(minutos>>4)+0x30;
linea[4]=(minutos&0xf0f)+0x30;
linea[5]=':.';
linea[6]=(segundos>>4)+0x30;
linea[7]=(segundos&0xf0f)+0x30;
}

```

```

for(i=0;i<8;++i)
  lcd_putc(Linea[i]);
}

////////////////////////////////////
// void despliega_fecha(int1 Mod)
////////////////////////////////////
void despliega_fecha(int1 Mod){
  lee_fecha();
  if(Mod==0)
    lcd_putc(LIMPIAR);
  else
    lcd_gotoxy(1,2);
  Linea[0]=(dia>>4)+0x30;
  Linea[1]=(dia&0xf0f)+0x30;
  Linea[2]=': ';
  Linea[3]=(mes>>4)+0x30;
  Linea[4]=(mes&0xf0f)+0x30;
  Linea[5]=': ';
  Linea[6]=(anno>>4)+0x30;
  Linea[7]=(anno&0xf0f)+0x30;

  for(i=0;i<8;++i)
    lcd_putc(Linea[i]);
}

////////////////////////////////////
// int opt_despliega_FH()
////////////////////////////////////
int1 opt_despliega_FH(){
  despliega_fecha(0);
  lcd_gotoxy(9,2);
  lcd_putc(HRS);
  while(TRUE){
    despliega_hora();
    for(x=0;x<6;++x){
      for(i=0;i<3000;++i){
        k=kbd_getc();
        if(k!='X') return(0);
      }
    }
  }
}

////////////////////////////////////
// void MoverCursor(byte P, byte n, byte Dir)
//
// Dir
// 1 Derecha
// 0 Izquierda
////////////////////////////////////
void MoverCursor(byte P, byte n, byte Dir){
  boolean estado;

  while(TRUE){
    if(Dir)
      { if(i==(n-1)) goto SALIR;
        else ++i;
      }
    else
      { if(!i) goto SALIR;
        else --i;
      }
    switch(P){
      case 1: estado = P1[i]; break;
      case 2: estado = P2[i]; break;
    }
    if(estado==1) goto SALIR;
  }
SALIR:
  lcd_gotoxy(i+1,2);
}

////////////////////////////////////
// int1 plantilla(byte P, byte n)
//
// PERMITE INTRODUCIR LOS DATOS DE UN
// FORMULARIO,
// POR EJEMPLO CUANDO SE MODIFICA LA HORA
////////////////////////////////////
int1 plantilla(byte P, byte n){
  char k2, k;
  int16 y;

  lcd_send_byte(0,BLINK);
  lcd_gotoxy(1,2);
  i=0;

  while(TRUE){
CICLO:
    k=t_getchar();
    tec_no_rep:
    if(k >= 10) goto comando;
    else {
      if(P==1){
        k+=0x30;
        Linea[i]=k;
        lcd_putc(k);
      }
      else {
        Linea[i]=F2[k];
        lcd_putc(F2[k]);
        k+=0x30;
        cont=1;
      }
    }
  esp_rep:
    lcd_gotoxy(i+1,2);
    for(y=0;y<DELAY_REP;++y)
      { k2=kbd_getc();
        if(k2 == 'X') Continue;
        if(k2>9)
          { k=k2;
            goto COMANDO;
          }
        if((k2+0x30)==k)
          { ++cont;
            if(cont==4) cont=0;
            if(cont==1) k2=F2[k2];
            else if(cont==2) k2=F3[k2];
            else if(cont==3) k2=F4[k2];
            else k2+=0x30;
            lcd_putc(k2);
            Linea[i]=k2;
            goto esp_rep;
          }
        else
          {
            MoverCursor(P,n,1);
            k=k2;
            lcd_putc(F2[k2-0x30]);
            Linea[i]=F2[k2-0x30];
            cont=1;
            goto esp_rep;
          }
      }
    MoverCursor(P,n,1);
    goto CICLO;
  }
  Linea[i]=k;
  MoverCursor(P,n,1);
}

Comando:
switch(k){
  case 11:
    lcd_send_byte(0,NO_BLINK); return(0);
  case 13:
    MoverCursor(P,n,0); break;
  case 14:
    MoverCursor(P,n,1); break;
  case 15:

```

```

        lcd_send_byte(0,NO_BLINK); return(1);
    }
}

////////////////////////////////////
// byte lee_byte_linea(byte posIni)
////////////////////////////////////
byte lee_byte_linea(byte posIni){
byte dato;
    dato=(Linea[posIni]-0x30)<<4;
    return(dato| (Linea[posIni+1]-0x30));
}

////////////////////////////////////
// void opCancelada()
////////////////////////////////////
void opCancelada(){
    lcd_putc("\fOperacion\ncancelada");
    delay_ms(1000);
}

////////////////////////////////////
// int1 CancelarOp()
////////////////////////////////////
int1 CancelarOp(){
    lcd_putc("\fDesea abortar\nla operacion?");
    if(t_getchar()==11) return(1);
    else return(0);
}

////////////////////////////////////
// int1 PreguntaSiNo()
////////////////////////////////////
int1 PreguntaSiNo(){
    if(t_getchar()==11) return(1); //Si
    else return(0); //No
}

////////////////////////////////////
// void despliega_rtc(byte num)
////////////////////////////////////
void despliega_rtc(byte num){
    lcd_putc((num>>4)+0x30);
    lcd_putc((num & 0x0F)+0x30);
}

int8 esta_pila(void){
int16 adc;
signed int16 vol;
byte carga;
    setup_adc(ADC_CLOCK_INTERNAL );
    set_adc_channel(1);
    delay_ms(2);
    adc = read_adc();
    if(adc>935) adc=935;
    vol = adc - 728;
    if(vol < 0) return(0);
    carga = vol / 20;
    if(carga > 9) carga= 9;
    setup_adc(ADC_OFF );
    return(carga);
}

////////////////////////////////////
// void opt_estado(void)
////////////////////////////////////
void opt_estado(void){
byte pila;
    lcd_putc("\fMO:");
    printf(lcd_putc,"%lu",leer_long_eeprom(NUM_MED));
    lcd_putc(" ML:");
    printf(lcd_putc,"%lu",MaxMed-
leer_long_eeprom(NUM_MED));
    lcd_putc("\nPILA: ");

    pila=esta_pila();
    for(i=2; i<=pila; i++)
        lcd_putc(255);
    lcd_gotoxy(15,2); lcd_putc("");
    t_getchar();
}

void preparar_lcd(byte x, byte y, byte n_pos){
    lcd_gotoxy(x,y);
    for(i=0;i<n_pos;++i)
        lcd_putc(" ");
    for(i=0;i<n_pos;++i)
        lcd_putc('\b');
}

////////////////////////////////////
// int1 opt_calibrar()
////////////////////////////////////
// OPTIENE LOS FACTORES DE CALIBRACIÓN.
////////////////////////////////////
int1 byte opt_calibrar(){
int16 offset;
Inicio:
    lcd_putc(LIMPIAR);
    lcd_gotoxy(2,1);
    lcd_putc("> CALIBRAR <\ncierre la pinza ");

espera:
do
    { k=kbd_getc();
    }while(k!='X' && input(SW_PINZA));

if(!input(SW_PINZA))
{
    for(i=0;i<DELAY_SW;++i)
    {
        k=kbd_getc();
        if(input(SW_PINZA)) goto espera;
        else if(k==15) return(0);
    }
    lcd_putc(LIMPIAR);
    BEEP(100, 30);
    escalaD=0;
    LInfra=leer_adc_prom(1);
    LRojo=leer_adc_prom(2);
    if(LRojo < MIN_CAL_R || LInfra < MIN_CAL_I)
    { lcd_putc(LIMPIAR);
      lcd_gotoxy(3,1); lcd_putc("(X) ERROR");
      BEEP(120, 250);
      delay_ms(1000);
      goto inicio;
    }

    callInfra= (float)VAL_FAB_INFRA/LInfra;
    callRojo=(float)VAL_FAB_ROJO/LRojo;
    BEEP(100, 30); delay_ms(40);
    BEEP(100, 40);
    lcd_gotoxy(1,1);
    lcd_putc("*. CALIBRADO .*");
    delay_ms(1000);
    return(1);
}
else if( k != 15 ) goto espera;
return(0);
}

////////////////////////////////////
// int opt_medir()
////////////////////////////////////
int opt_medir(){
signed int8 medOK;
float valorClor;
    if(callRojo == -1)
        if(opt_calibrar() == 0) return(0);
    nMediciones = leer_long_eeprom(NUM_MED);
    nMedicionesAnt=nMediciones;
    lee_fecha();
}

```

```

//-----pregunta inicial-----
Supr:
  Dir=(nMediciones-1)*LS;
  h=leer_eeprom(Dir); ++Dir;
  m=leer_eeprom(Dir);

NuevaMed:
  lcd_putc(LIMPIAR);
  lcd_putc("Cultivo:");
  lcd_gotoxy(1,2);
  for(i=0;i<8;++i) {
    Linea[i]=' ';
  }
  if(plantilla(2,6)) return(0);
  h=1; m=0;
//-----
REINICIAR:
  lcd_putc(LIMPIAR);
  for(i=0;i<6;++i)
    lcd_putc(linea[i]);
  lcd_gotoxy(1,2);
  lcd_putc("H:");
  lcd_gotoxy(7,2);
  lcd_putc("M:");

SIG_M:
  preparar_lcd(3,2,3);
  printf(lcd_putc,"%u",h);
  preparar_lcd(9,2,3);
  printf(lcd_putc,"%u",m);

SWT:
  while(!input(SW_PINZA));
  do
  { k=kbd_getc();
  }while(k=='X' && input(SW_PINZA));

  if(!input(SW_PINZA))
  {
    for(i=0;i<DELAY_SW;++i)
    { k=kbd_getc();
      if(input(SW_PINZA)) goto SWT;
      else if(k!='X') goto T_PRESIONADA;
    }
    BEEP(100, 20);

    medOK= medir();
    preparar_lcd(13,2,4);

    if(medOK < 0) valorClor=ajuste_poli();

    if(medOK < 0 || valorClor == -1)
    { BEEP(125, 100); delay_ms(80);
      BEEP(120, 250);
      lcd_putc("-.");
      lcd_gotoxy(14,1);
      if(medOK==2) lcd_putc("PA");
      else lcd_putc("FR");
      goto SWT;
    }
    lcd_gotoxy(14,1); lcd_putc(" ");
    if(m== 255) {m=1; h++;}
    else ++m;

    printf(lcd_putc,"%01.1f",valorClor);
    BEEP(98, 30);
//-----ALMACENAR MEDICION
    Dir=nMediciones*LS;
    for(i=0;i<6;++i)
    { escribir_eeprom(Dir, Linea[i]);
      ++Dir;
    }

    escribir_eeprom(Dir,h);
    ++Dir;
    escribir_eeprom(Dir,m);

    ++Dir;
    escribir_eeprom(Dir, dia);
    ++Dir;
    escribir_eeprom(Dir, mes);
    ++Dir;
    lee_hora();
    escribir_eeprom(Dir,horas);
    ++Dir;
    escribir_eeprom(Dir,minutos);
    ++Dir;
    escribir_long_eeprom(Dir,LInfra);
    Dir+=2;
    escribir_long_eeprom(Dir,LRojo);
    ++nMediciones;
    escribir_long_eeprom(NUM_MED, nMediciones);
    goto SIG_M;
  }
T_PRESIONADA:
  switch(k){
  case 15:
    if(!CancelarOp())
      goto REINICIAR;
    escribir_long_eeprom(NUM_MED, nMedicionesAnt);
    return(0);
  case 11:
    return(0);
  case 1:
    if(m!=0)
    { ++h ; m=0;
    } goto REINICIAR;
  case 2:
    if(h==1 && m==1)
    { m=0;
      --nMediciones;
      escribir_long_eeprom(NUM_MED, NMediciones);
      goto SIG_M;
    }
    else if(h==1 && m==0) goto SWT;
    else
    { --nMediciones;
      escribir_long_eeprom(NUM_MED, NMediciones);
      goto Supr;
    } goto SWT;
  default: goto SWT;
  }
}

////////////////////////////////////
// int opt_ver_datos()
////////////////////////////////////
int opt_ver_datos(){
  nMediciones=leer_long_eeprom(NUM_MED);
  lcd_putc("\fTotal: ");
  printf(lcd_putc,"%lu",nMediciones);
  lcd_putc("\nMediciones");
  delay_ms(1000);
  n=0;
  Dir=0;
  while(TRUE && nMediciones > 0){
    Dir=n*LS;
    lcd_putc(LIMPIAR);
    for(i=0;i<6;++i){
      lcd_putc(leer_eeprom(Dir));
      ++Dir;
    }
    lcd_putc(' ');
    printf(lcd_putc,"%u",leer_eeprom(Dir));
    ++Dir;
    lcd_putc('/');
    m=leer_eeprom(Dir);
    printf(lcd_putc,"%u",m);
    ++Dir;
    lcd_putc("\n");
    despliega_rtc(leer_eeprom(Dir));
    ++Dir;
    lcd_putc('/');
  }
}

```



```

despliega_rtc(Leer_eeprom(Dir));
++Dir;
lcd_putc('-');
despliega_rtc(Leer_eeprom(Dir));
++Dir;
lcd_putc(':');
despliega_rtc(Leer_eeprom(Dir));
++Dir;
LInfra=Leer_long_eeprom(Dir);
Dir+=2;
LRojo=Leer_long_eeprom(Dir);

printf(lcd_putc, "%1.1f", ajuste_poli());

switch(t_getchar()){
case 14: if(n<(nMediciones-1)) ++n; break;
case 12:
    if(n>0) --n;
    else n= nMediciones-1;
    break;
case 15: return(0);
}
}

////////////////////////////////////
// int opt_mod_Hora()
////////////////////////////////////
int1 opt_mod_Hora(){
START:
    lcd_putc(LIMPIAR);
    lcd_putc("Nueva hora:");
    despliega_hora();
    lcd_putc(HRS);
    if(plantilla(1,8)== 1)
    {
        opCancelada();
        return(0);
    }
    horas=lee_byte_linea(0);
    minutos=lee_byte_linea(3);
    segundos=lee_byte_linea(6);
    if(horas>0x23 | minutos>0x59 | segundos >0x59)
        goto MAL;
    write_clock_rtc(horas,minutos,segundos);
    return(0);
MAL:
    lcd_putc(LIMPIAR);
    lcd_putc("Hora incorrecta");
    delay_ms(1000);
    goto START;
}

////////////////////////////////////
// int1 opt_mod_Fecha()
////////////////////////////////////
int1 opt_mod_Fecha(){
START:
    lcd_putc(LIMPIAR);
    lcd_putc("Nueva fecha:");
    despliega_fecha(1);
    if(plantilla(1,8)== 1)
    {
        opCancelada();
        return(0);
    }
    dia=lee_byte_linea(0);
    mes=lee_byte_linea(3);
    anno=lee_byte_linea(6);
    if(dia>0x31 | mes >0x12 | dia==0 | mes==0 )
        goto MAL;
    write_date_rtc(dia,mes,anno);
    return(0);
MAL:
    lcd_putc(LIMPIAR);
    lcd_putc("Fecha incorrecta");

    delay_ms(1000);
    goto START;
}

////////////////////////////////////
// byte prueba_dato()
////////////////////////////////////
byte prueba_dato(){
    i=ESP_RS232;
    while(!kbhit() && (--i>0)){
        delay_ms(1);
    }
    if(kbhit()){
        if(dato == getch()) return(1);
        else return(0);
    }
    else return(0);
}

////////////////////////////////////
// int opt_transferir()
////////////////////////////////////
int1 opt_transferir(){
    datPorTran=Leer_long_eeprom(NUM_MED)*LS;
    contDatTran=0;
    if(!datPorTran) {
        lcd_putc("\fMemoria vacia");
        t_getchar();
        return(0);
    }
    lcd_putc("\fListo para\ntransferir datos");
    // -----Espera comando de envio o cancelar la
operacion
espera:
    if(kbhit()) getch();
    do { k=kbd_getc();
    }while(!kbhit() && k == 'X');

    if(kbhit()) {
        if(getc()=="E") goto iniciar;
        else goto espera;
    } else if(k == 15) {
        opCancelada();
        return(0);
    } else goto espera;

iniciar:
    lcd_putc("\fEnviando datos...");
    // -----Envia el numero de byte que se
enviaran
    dato=datPorTran>>8;
    putc(dato);
    if(!prueba_dato()) goto Error;
    dato=datPorTran;
    putc(dato);
    if(!prueba_dato()) goto Error;

    while(contDatTran != datPorTran && kbd_getc() != 15){
        dato=Leer_eeprom(contDatTran);
        putc(dato);
        if(!prueba_dato()) goto Error;
        contDatTran++;
    }
    lcd_putc("\fTransferencia\nfinalizada");
    t_getchar();
    return(1);
Error:
    lcd_putc("\f(X) Error de\ncomunicacion");
    t_getchar();
    return(0);
}

////////////////////////////////////
// opt_borrar_mem()
////////////////////////////////////
void opt_borrar_mem(){
    lcd_putc("\fDesea borrar\nla memoria?");
}

```

```

if(!PreguntaSiNo())
{ opCancelada();
  break;
}
}
escribir_long_eeprom(NUM_MED,0);
if(Leer_long_eeprom(NUM_MED)>0){
  lcd_putc("\fError <24512>");
  t_getchar();
}
}

////////////////////////////////////
// void ini_pic(void)
////////////////////////////////////
void ini_pic(void){
  port_b_pullups(TRUE);
  set_tris_e(0);
  output_low(LED_ROJO);
  output_low(LED_INFRA);
  output_low(BUZZER);
  setup_adc_ports( RA0_RA1_RA3_ANALOG );
  setup_adc(ADC_OFF);
  init_ext_eeprom();
  lcd_init();
  ini_rtc();
  calRojo=-1;
}

////////////////////////////////////
// void main(void)
////////////////////////////////////
void main() {
  byte pila;
  ini_pic();
  menu=0;
  while (TRUE) {
  ciclo:
  muestra_menu();
  no_key:

  k=t_getchar();
  switch (k){
  case 13:
    if(menu==max_menu) menu=0; else ++menu; break;
  case 14:
    if(menu==0) menu=MAX_MENU; else --menu; break;
  case 0x0B: goto ENTER; break;
  default: goto no_key;
  }
  goto ciclo;
ENTER:
  switch(menu){
  case 0: // REALIZA LAS MEDICIONES
    if(esta_pila(<2)
    { lcd_putc("\f(!) ADVERTENCIA\n pila muy baja");
      BEEP(80,100);
      t_getchar();
    }
    opt_medir(); break;
  case 1: // DESPLIEGA LAS MEDICIONES
    ALMACENADAS
    opt_ver_datos(); break;
  case 2: //MUESTRA LA MEMORIA OCUPADA Y
    LIBRE, Y EL ESTADO
    opt_estado(); break; //DE LA PILA
  case 3:
    opt_borrar_mem(); break;
  case 4: // TRANSFIERE A LA COMPUTADORA
    opt_transferir(); break;
  case 5: // DESPLIEGA LA FECHA Y LA HORA
    opt_despliega_FH(); break;
  case 6: // AJUSTAR RELOJ
    opt_mod_Hora(); break;
  case 7: // AJUSTAR LA FECHA
    opt_mod_Fecha(); break;
  }
}
}

```

Rutinas para manejar el lcd (LCD.C)

```

////////////////////////////////////
//          LCD.C
//          Driver for LCD
//
// lcd_init()
// Tiene que ser llamada antes de utilizar cualquier
//          rutina del LCD
//
// lcd_putc(c) Coloca un carácter en la posición
//          actual
// y se mueve a la siguiente posición
//          \f limpia el display
//          \n Salta el cursor a la segunda línea
//          \b Regresa una posición
//
// lcd_gotoxy(x,y) Mueve el cursor a la posición
//          correspondiente
////////////////////////////////////

#define RW PIN_E0
#define RS PIN_E1
#define E PIN_C0

#define set_tris_lcd(x) set_tris_d(x)

#define lcd_type 2
#define lcd_line_two 0x40

byte CONST LCD_INIT_STRING[4] = {0x20 |
(lcd_type << 2), 0xc, 1, 6};

#define LCD_WRITE 0x00
#define LCD_READ 0xFF

byte lcd_read_byte() {
  byte low,high;
  set_tris_lcd(LCD_READ);
  output_high(rw);
  delay_cycles(1);
  output_high(E);
  delay_cycles(5);
  high = input_d();
  output_low(E);
  delay_cycles(1);
  output_high(E);
  delay_us(1);
  low = input_d();
  output_low(E);
  set_tris_lcd(LCD_WRITE);
  return( (high<<4) & low);
}

void lcd_send_nibble( byte n ) {
  output_d(n);
  delay_cycles(1);
  output_high(E);
}

```

```

    delay_us(30);
    output_low(E);
}

void lcd_send_byte( byte address, byte n ) {
    output_low(RS);
    while ( bit_test(lcd_read_byte(),7) );
    if(address==0)
        output_low(RS);
    else
        output_high(RS);

    delay_cycles(4);
    output_low(RW);
    output_low(E);
    delay_cycles(4);
    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}

void lcd_init() {
    byte i;
    set_tris_lcd(LCD_WRITE);
    output_low(RS);
    output_low(RW);
    output_low(E);
    delay_ms(15);
    for(i=1;i<=3;++i) {
        lcd_send_nibble(3);
        delay_ms(5);
    }
    lcd_send_nibble(2);
    for(i=0;i<=3;++i)
        lcd_send_byte(0,LCD_INIT_STRING[i]);
}

void lcd_gotoxy( byte x, byte y ) {
    byte address;

    if(y!=1)
        address=lcd_line_two;
    else
        address=0;
    address+=x-1;
    lcd_send_byte(0,0x80|address);
}

void lcd_putc( char c ) {
    switch (c) {
        case '\f' : lcd_send_byte(0,1);
                    delay_ms(2);
                    break;
        case '\n' : lcd_gotoxy(1,2); break;
        case '\b' : lcd_send_byte(0,0x10); break;
        default  : lcd_send_byte(1,c); break;
    }
}

////////////////////////////////////////////////////
// KBDD.C ////////////////////////////////////////////////////

//////////////////////////////////////////////////
// Driver para el teclado ////////////////////////////////////////////////////
// ////////////////////////////////////////////////////
// c = kbd_getc(c) Regresa una 'X' cuando ////////////////////////////////////////////////////
// no se ha presionado la ////////////////////////////////////////////////////
// y cuando se presiona una ////////////////////////////////////////////////////
// tecla regresa el valor de la ////////////////////////////////////////////////////
// correspondiente. ////////////////////////////////////////////////////
// ////////////////////////////////////////////////////

#define kbd = 6
#define set_tris_kbd(x) set_tris_b(x)

#define COL0 (1 << 0)
#define COL1 (1 << 1)
#define COL2 (1 << 2)
#define COL3 (1 << 3)

#define ROW0 (1 << 7)
#define ROW1 (1 << 6)
#define ROW2 (1 << 5)
#define ROW3 (1 << 4)

#define ALL_ROWS (ROW0|ROW1|ROW2|ROW3)
#define ALL_PINS (ALL_ROWS|COL0|COL1|COL2|COL3)

char const KEYS[4][4] = {{1,2,3,10},
                        {4,5,6,11},
                        {7,8,9,12},
                        {13,0,14,15}};

#define KBD_DEBOUNCE_FACTOR 80

void BEEP(byte Periodo, long int time){
    long int i;
    byte p;
    for(i=0;i<time;++i){
        output_high(BUZZER);
        for(p=0;p<periodo;++p);
        output_low(BUZZER);
        for(p=0;p<periodo;++p);
    }
}

char kbd_getc( ) {
    static byte kbd_call_count;
    static short int kbd_down;
    static char last_key;
    static byte col;
    byte kchar;
    byte row;

    kchar='X';
    if(++kbd_call_count>KBD_DEBOUNCE_FACTOR)
    {
        switch (col) {
            case 0 : set_tris_kbd(ALL_PINS&~COL0);
                    kbd=~COL0&ALL_PINS;
                    break;
            case 1 : set_tris_kbd(ALL_PINS&~COL1);
                    kbd=~COL1&ALL_PINS;
                    break;
            case 2 : set_tris_kbd(ALL_PINS&~COL2);
                    kbd=~COL2&ALL_PINS;
                    break;
            case 3 : set_tris_kbd(ALL_PINS&~COL3);

```

Rutinas para el manejo del teclado (KBD.C)

```

        kbd=~COL3&ALL_PINS;
        break;
    }

    if(kbd_down) {
        if((kbd & (ALL_ROWS))==(ALL_ROWS)) {
            kbd_down=false;
            kchar=last_key;
            last_key='X';
        }
    } else {
        if((kbd & (ALL_ROWS))!=(ALL_ROWS)) {
            if((kbd & ROW0)==0)
                row=0;
            else if((kbd & ROW1)==0)
                row=1;
            else if((kbd & ROW2)==0)
                row=2;
            else if((kbd & ROW3)==0)
                row=3;
            last_key =KEYS[row][col];
            kbd_down = true;
            BEEP(80, 40);
        }
        else {
            ++col;
            if(col==4)
                col=0;
        }
    }
    kbd_call_count=0;
}
set_tris_kbd(ALL_PINS);
return(kchar);
}

```

Rutinas para el manejo de la memoria serial (24LC512.C)

```

////////////////////////////////////
//// Librerias para una EEPROM serial
//// 24LC512
////
//// init_ext_eeprom();
//// Inicializa las patas de comunicación
////
//// escribir_eeprom(a, d);
//// Escribe un byte d en la dirección a
////
//// d = leer_eeprom(a);
//// Lee un byte d de la dirección a
////
//// escribir_long_eeprom(a, d);
//// Lee dos byte d de la dirección a
////
//// d = leer_long_eeprom(a);
//// Lee dos byte d en la dirección a
////
////////////////////////////////////

```

```

#ifndef EEPROM_SDA
#define EEPROM_SDA PIN_C4
#define EEPROM_SCL PIN_C3
#endif

```

```

#include I2C(master,sda=EEPROM_SDA,
scl=EEPROM_SCL,
address=0xa0,FAST,FORCE_HW)

#define EEPROM_ADDRESS long int
#define EEPROM_SIZE 65535

void init_ext_eeprom()
{
    output_float(EEPROM_SCL);
    output_float(EEPROM_SDA);
}

void escribir_eeprom(int16 Direccion, byte dato)
{
    init_ext_eeprom();
    i2c_start();
    i2c_write(0xa0);
    i2c_write(Direccion>>8);
    i2c_write(Direccion);
    i2c_write(dato);
    i2c_stop();
    delay_ms(11);
    Direccion++;
}

byte leer_eeprom(int16 Direccion) {
    byte dato;
    init_ext_eeprom();
    i2c_start();
    i2c_write(0xa0);
    i2c_write(Direccion>>8);
    i2c_write(Direccion);
    i2c_start();
    i2c_write(0xa1);
    dato=i2c_read(0);
    i2c_stop();
    return(dato);
}

```

```

void escribir_long_eeprom(int16 Direccion, int16
Dato){
    escribir_eeprom(Direccion, dato >>8);
    ++Direccion;
    escribir_eeprom(Direccion, dato);
}

int16 leer_long_eeprom(int16 Direccion){
int16 dato;
    dato=leer_eeprom(Direccion);
    dato = dato <<8;
    ++Direccion;
    dato=dato | leer_eeprom(Direccion);
    return(dato);
}

```

Rutinas para el manejo del Reloj en tiempo real (RTC.C)

```

////////////////////////////////////
//
// RTC.C
// DRIVER PARA MANEJAR UN RELOJ EN
// TIEMPO REAL
//

```

```

// read_rtc(ADDR)
//
// write_rtc(ADDR, DATO)
//-----//

//
//          DALLAS SEMICONDUCTOR
//          DS12887A
//          +-----+
//          | O       |
//          GND -|MOT  VCC|- VCC
//          -|NC     SQW|-
//          -|NC     NC|-
//          RD0 -|AD0  /RCLR|-
//          RD1 -|AD1  NC|-
//          RD2 -|AD2  /IRQ|-
//          RD3 -|AD3  /RESET|- VCC
//          RD4 -|AD4  DS|- RE2
//          RD5 -|AD5  NC|-
//          RD6 -|AD6  R/W|- RE0
//          RD7 -|AD7  AS|- RE1
//          GND -|GND  /CS|- RC1
//          +-----+
//
//*****
//          DIRECCION | CONTENIDO
//          -----
//          00      SEGUNDOS
//          02      MINUTOS
//          04      HORAS
//          06      DIA DE LA SEM.
//          07      DIA DEL MES
//          08      MES
//          09      AÑO
//          0A      REGISTRO A
//          0B      REGISTRO B
//          0C      REGISTRO C
//          0D      REGISTRO D
//          //-----//

#define AS PIN_E1
#define DS PIN_E2
#define CS PIN_C1

#define WRITE 0x00
#define READ 0xFF
#define tris_rtc(x) set_tris_d(x)

byte read_byte_rtc(byte addr){
byte dato;

    tris_rtc(WRITE);
    output_high(RW);
    output_high(DS);
    output_high(AS);
    output_d(addr);
    output_low(CS);
    delay_cycles(4);
    output_low(AS);
    delay_cycles(4);
    tris_rtc(READ);
    output_low(DS);
    dato=input_d();
    output_high(CS);

    return(dato);
}

void write_byte_rtc(byte addr, byte dato){

    tris_rtc(WRITE);
    output_high(RW);
    output_high(DS);
    output_high(AS);
    output_d(addr);
    output_low(CS);
    delay_cycles(4);
    output_low(AS);
    delay_cycles(4);
    output_low(RW);
    output_d(dato);
    output_high(CS);
}

void ini_rtc(){
    write_byte_rtc(0x0B, 0x80);
    write_byte_rtc(0x0A, 0x2C);
    write_byte_rtc(0x0C, 0x04);
    write_byte_rtc(0x0B, 0x02);
}

void write_clock_rtc(byte horas, byte minutos, byte segundos){
    write_byte_rtc(4, horas);
    write_byte_rtc(2, minutos);
    write_byte_rtc(0, segundos);
}

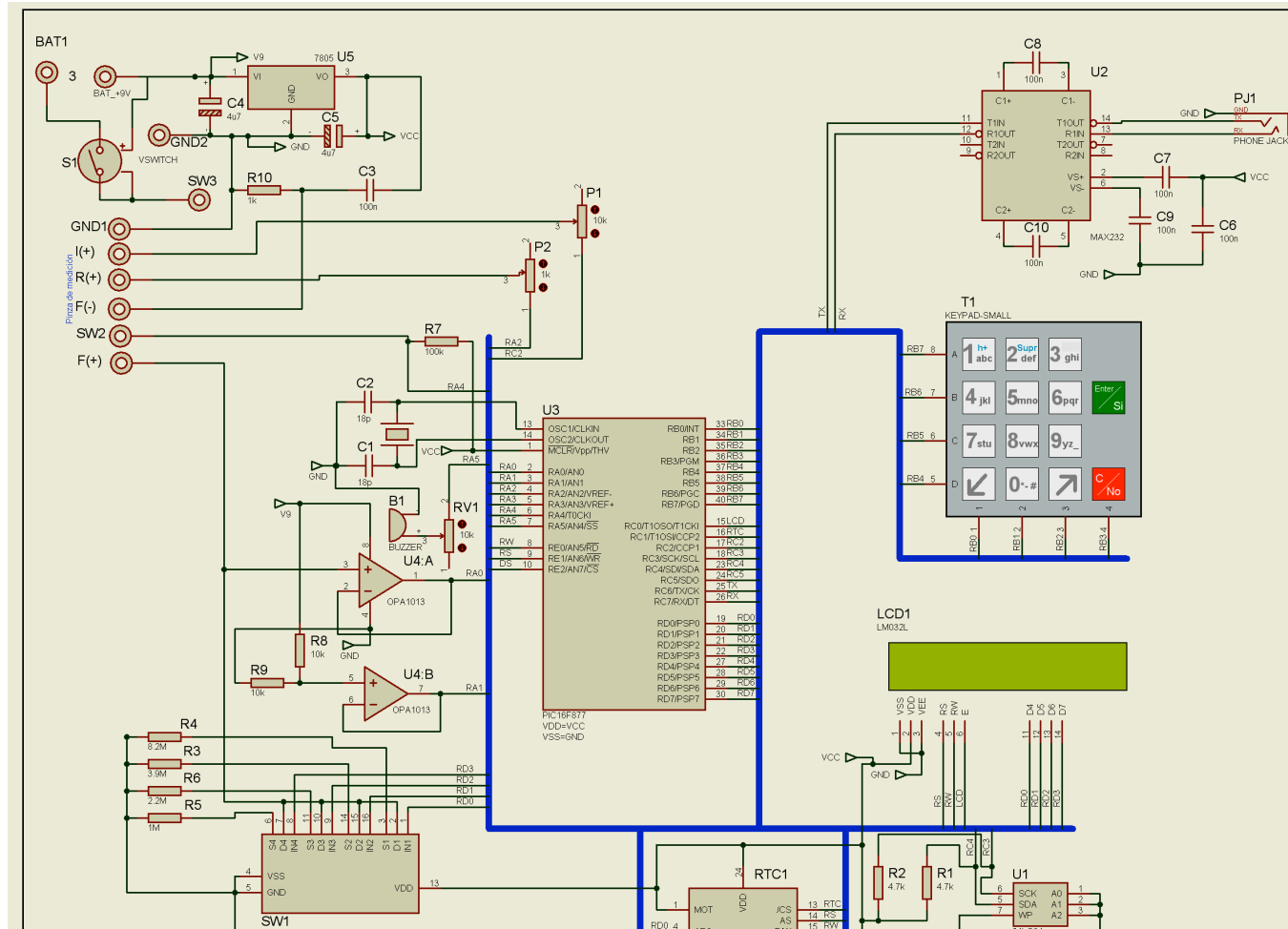
void write_date_rtc(byte dia, byte mes, byte anno){
    write_byte_rtc(7, dia);
    write_byte_rtc(8, mes);
    write_byte_rtc(9, anno);
}

```

APÉNDICE 4

DIAGRAMA COMPLETO DEL CIRCUITO

APÉNDICE 4. Diagrama completo del circuito



APÉNDICE 5

**PROGRAMA EN MATLAB PARA GENERAR LOS POLINOMIOS DE
AJUSTE**


```

%%%%%%%%%%%%%%
% Este código obtiene dos polinomios de quinto orden para ajustar la relación
% ri, a los valores obtenidos con el patrón de referencia.
%
% Los datos los obtiene de una tabla dentro del archivo m.dat.
%%%%%%%%%%%%%%
load m.dat

vs=m(:, 1); i=m(:,2); r=m(:,3);

RI_CAM = 0.63;

f1=1;
f2=1;
ri= r./(i+r);

RI_1=0; vs_1=0;
RI_2=0; vs_2=0;
p1=0; p2=0; y1=0; y2=0; y=0;

for i=1:size(vs)
    if ri(i) > RI_CAM
        RI_1(f1,1)=ri(i);
        vs_1(f1,1)=vs(i);
        f1=f1+1;
    else
        RI_2(f2,1)=ri(i);
        vs_2(f2,1)=vs(i);
        f2=f2+1;
    end
end

p1 = polyfit(RI_1,vs_1,4);
y1 = polyval(p1,RI_1);

p2 = polyfit(RI_2,vs_2,5);
y2 = polyval(p2,RI_2);

f=1;

for i=1:size(y1)
    y(f,1)=y1(i);
    f=f+1;
end
for i=1:size(y2)
    y(f,1)=y2(i);
    f=f+1;
end

figure; plot([vs, y]);

```

APÉNDICE 6

GLOSARIO

In situ: referido a conservación de recursos genéticos, la que se realiza en su medio natural, y que para las especies domesticadas se verifica en el medio donde desarrollaron sus propiedades distintivas

Método del kjeldahl (El Proceso de Determinación de Nitrógeno. Una Aproximación al Sistema): El Análisis de Nitrógeno Kjeldahl se desarrolló a finales del siglo XIX por el científico danés, Johann Kjeldahl. Mientras que el análisis es usado con frecuencia para la determinación de proteína en productos agrícolas, también es aceptado como método opcional para la determinación de nitrógeno en materiales tales como tejidos de plantas, fertilizantes, desechos orgánicos y aguas de desecho.

Los pasos del procedimiento son los siguientes:

1. Los primeros pasos requeridos en los tres sistemas de digestión consisten en pesar las muestras y colocarlas en el matraz o tubo de tamaño apropiado. En el nivel macro, el tamaño del matraz es usualmente de 500 u 800 ml.
2. Cuando se analizan múltiples muestras, los portadores de matraces evitan numerosos viajes de una estación a otra. Labconco proporciona tres estilos diferentes, cada uno diseñado para un matraz o tubo de digestión de tamaño específico. Esto facilita una rápida transportación de la muestras previamente pesadas a la unidad de ácido y cáusticos.
3. El cuarto paso es agregar ácido sulfúrico, sal y catalizador a la muestra pesada.
4. Una vez que se agregan los reactivos a la muestra el proceso de digestión real puede empezar.

5. El proceso de digestión en este nivel toma cerca de dos horas.
6. Antes de que el proceso de destilación pueda empezar en la escala macro, los matraces deben ser devueltos a la unidad de ácido y cáustico. Las muestras son entonces colocadas en la unidad de destilación por aproximadamente 35 minutos.
7. El paso final en el procedimiento es remover los matraces receptores y titular la muestra. Después de la titulación, el contenido total de nitrógeno puede ser calculado de una fórmula estándar. El contenido de proteína entonces puede ser fácilmente determinado de este resultado.

Método de Harbone (Procedimiento para la toma de datos de clorofila **a**, **b** y **total** en hoja bandera de trigo var. Sonora-64).

Materiales

Material biológico

Semillas de trigo (*Triticum aestivum*) variedad Sonora-64

Otros materiales y reactivos

1. Hielera
2. Papel filtro
3. Papel aluminio
4. Sacabocados de 11 mm aproximadamente
5. Balanza
6. Estufa
7. Morteros
8. Acetona
9. Centrífuga clínica
10. Espectrofotómetro
11. Carbonato de calcio

Procedimiento:

1. Las semillas de trigo son sembradas en macetas con 6 kg de suelo bien fertilizado y mantenido a capacidad de campo en un invernadero. Las

semillas de trigo también pueden ser sembradas en el campo. Después de que las plantas exponen la hoja bandera (antes de antesis) puede comenzarse a tomar la hoja bandera, colocándola en papel filtro húmedo, envueltas con papel aluminio y ponerlas en una hielera para su posterior traslado al laboratorio y almacenarlas a -20°C , hasta su análisis.

2. De cada hoja se obtienen 6 discos utilizando el sacabocados, cada disco tiene aproximadamente 11 mm de diámetro. Tres discos se colocan en la estufa hasta su deshidratación para determinar el peso seco. Los otros tres discos se toman para hacer la extracción de clorofila.
3. Esta técnica está basada en el método de Harborne, 1973. Se toman los tres discos de hoja y se maceran con 10 ml de acetona al 80%, agregando un poco de carbonato al calcio. La mezcla se centrifuga a 3000 r.p.m. durante 5 minutos, hasta que el "pellet" se compacta en el fondo de tubo, para tomar libremente el sobrenadante. El sobrenadante se lleva a un volumen conocido como 20 ml, para hacer la lecturas de absorbancia en el espectrofotometro. El volumen para la extraccion puede variar, como también el volumen a que se lleva el sobrenadante, ya que depende de si la clorofila esta muy concentrada, por lo que puede llevarse a un volumen mayor para diluir. Todo este procedimiento se realiza en luz difusa para que no se oxiden las muestras de clorofila.
4. Se toman las lecturas de absorbancia en el espectrofotometro de las muestras a 645 y 663 nm. Es importante mencionar que las muestras deben de mantenerse siempre bajo hielo para evitar la oxidacion. Se utiliza un blanco de acetona al 80% sin clorofila.

Las cantidades de clorofila *a*, *b* y total se obtienen de las siguientes ecuaciones (ver Figura 1 de la practica de clorofila total in situ, en donde se muestra una comparación de clorofila por este método y por el método del Minolta).

$$\text{Clorofila } a = 12.7 \text{ Absorbancia (663 nm)} - 2.69 \text{ Absorbancia (645 nm)} = \text{mg/l}$$

$$\text{Clorofila } b = 22.9 \text{ Absorbancia (645 nm)} - 4.68 \text{ Absorbancia (663 nm)} = \text{mg/l}$$

$$\text{Clorofila Total} = 8.02 \text{ Absorbancia (663 nm)} + 20.2 \text{ Absorbancia (645 nm)} = \text{mg/l}$$

ANEXO 1

HOJA DE DATOS DEL PIC16F877A



PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM),
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin
PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during Sleep via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™
(Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) – 8 bits wide with
external \overline{RD} , \overline{WR} and \overline{CS} controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital
Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference
(VREF) module
 - Programmable input multiplexing from device
inputs and internal voltage reference
 - Comparator outputs are externally accessible

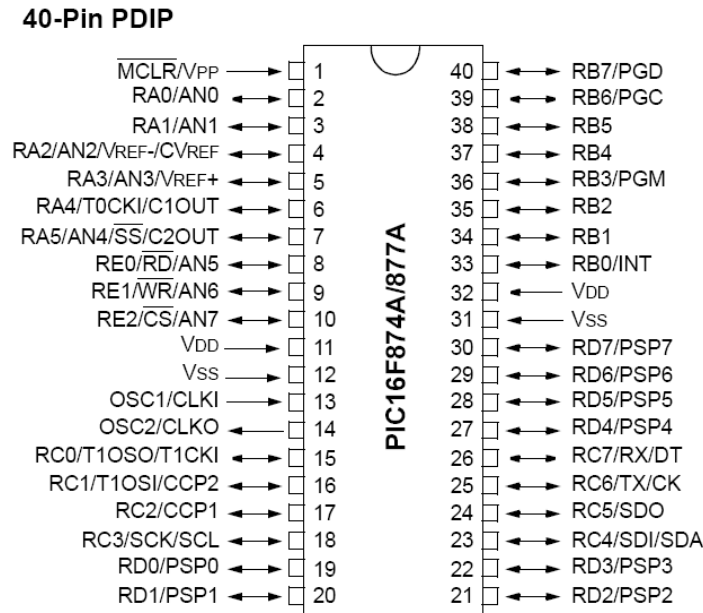
Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash
program memory typical
- 1,000,000 erase/write cycle Data EEPROM
memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™)
via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low-power, high-speed Flash/EEPROM
technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2



17.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias	-55 to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to Vss (except VDD, MCLR, and RA4)	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to Vss	-0.3 to +7.5V
Voltage on MCLR with respect to Vss (Note 2)	0 to +14V
Voltage on RA4 with respect to Vss	0 to +8.5V
Total power dissipation (Note 1)	1.0W
Maximum current out of Vss pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > VDD)	± 20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > VDD)	± 20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by PORTA, PORTB and PORTE (combined) (Note 3)	200 mA
Maximum current sourced by PORTA, PORTB and PORTE (combined) (Note 3)	200 mA
Maximum current sunk by PORTC and PORTD (combined) (Note 3)	200 mA
Maximum current sourced by PORTC and PORTD (combined) (Note 3)	200 mA

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

2: Voltage spikes below Vss at the MCLR pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the MCLR pin rather than pulling this pin directly to Vss.

3: PORTD and PORTE are not implemented on PIC16F873A/876A devices.

† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 15-2: PIC16F87XA INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode			Status Affected	Notes
			MSb	LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS							
ADDWF	f, d Add W and f	1	00	0111	dfff	ffff	C,DC,Z 1,2
ANDWF	f, d AND W with f	1	00	0101	dfff	ffff	Z 1,2
CLRF	f Clear f	1	00	0001	1fff	ffff	Z 2
CLRWF	- Clear W	1	00	0001	0xxx	xxxx	Z
COMF	f, d Complement f	1	00	1001	dfff	ffff	Z 1,2
DECF	f, d Decrement f	1	00	0011	dfff	ffff	Z 1,2
DECFSZ	f, d Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff	1,2,3
INCF	f, d Increment f	1	00	1010	dfff	ffff	Z 1,2
INCFSZ	f, d Increment f, Skip if 0	1(2)	00	1111	dfff	ffff	1,2,3
IORWF	f, d Inclusive OR W with f	1	00	0100	dfff	ffff	Z 1,2
MOVF	f, d Move f	1	00	1000	dfff	ffff	Z 1,2
MOVWF	f Move W to f	1	00	0000	1fff	ffff	
NOP	- No Operation	1	00	0000	0xx0	0000	
RLF	f, d Rotate Left f through Carry	1	00	1101	dfff	ffff	C 1,2
RRF	f, d Rotate Right f through Carry	1	00	1100	dfff	ffff	C 1,2
SUBWF	f, d Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z 1,2
SWAPF	f, d Swap nibbles in f	1	00	1110	dfff	ffff	1,2
XORWF	f, d Exclusive OR W with f	1	00	0110	dfff	ffff	Z 1,2
BIT-ORIENTED FILE REGISTER OPERATIONS							
BCF	f, b Bit Clear f	1	01	00bb	bfff	ffff	1,2
BSF	f, b Bit Set f	1	01	01bb	bfff	ffff	1,2
BTFSC	f, b Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff	3
BTFSS	f, b Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff	3
LITERAL AND CONTROL OPERATIONS							
ADDLW	k Add Literal and W	1	11	111x	kkkk	kkkk	C,DC,Z Z
ANDLW	k AND Literal with W	1	11	1001	kkkk	kkkk	Z
CALL	k Call Subroutine	2	10	0kkk	kkkk	kkkk	
CLRWDT	- Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$
GOTO	k Go to Address	2	10	1kkk	kkkk	kkkk	
IORLW	k Inclusive OR Literal with W	1	11	1000	kkkk	kkkk	Z
MOVLW	k Move Literal to W	1	11	00xx	kkkk	kkkk	
RETFIE	- Return from Interrupt	2	00	0000	0000	1001	
RETLW	k Return with Literal in W	2	11	01xx	kkkk	kkkk	
RETURN	- Return from Subroutine	2	00	0000	0000	1000	
SLEEP	- Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$
SUBLW	k Subtract W from Literal	1	11	110x	kkkk	kkkk	C,DC,Z
XORLW	k Exclusive OR Literal with W	1	11	1010	kkkk	kkkk	Z

- Note 1:** When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3:** If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

**TABLE 17-14: A/D CONVERTER CHARACTERISTICS: PIC16F873A/874A/876A/877A (INDUSTRIAL)
PIC16LF873A/874A/876A/877A (INDUSTRIAL)**

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
A01	NR	Resolution	—	—	10-bits	bit	$V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A03	EIL	Integral Linearity Error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A04	EDL	Differential Linearity Error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A06	E _{OFF}	Offset Error	—	—	$< \pm 2$	LSb	$V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A07	E _{GN}	Gain Error	—	—	$< \pm 1$	LSb	$V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$	
A10	—	Monotonicity	—	guaranteed ⁽³⁾	—	—	$V_{SS} \leq V_{AIN} \leq V_{REF}$	
A20	V _{REF}	Reference Voltage (V _{REF+} – V _{REF-})	2.0	—	$V_{DD} + 0.3$	V		
A21	V _{REF+}	Reference Voltage High	$AV_{DD} - 2.5V$	—	$AV_{DD} + 0.3V$	V		
A22	V _{REF-}	Reference Voltage Low	$AV_{SS} - 0.3V$	—	$V_{REF+} - 2.0V$	V		
A25	V _{AIN}	Analog Input Voltage	$V_{SS} - 0.3V$	—	$V_{REF+} + 0.3V$	V		
A30	Z _{AIN}	Recommended Impedance of Analog Voltage Source	—	—	2.5	kΩ	(Note 4)	
A40	I _{AD}	A/D Conversion Current (V _{DD})	PIC16F87XA	—	220	—	μA	Average current consumption when A/D is on (Note 1)
			PIC16LF87XA	—	90	—	μA	
A50	I _{REF}	V _{REF} Input Current (Note 2)	—	—	5	μA	During V _{AIN} acquisition. Based on differential of V _{HOLD} to V _{AIN} to charge C _{HOLD} , see Section 11.1 "A/D Acquisition Requirements". During A/D conversion cycle	
			—	—	150	μA		

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.
- 2:** V_{REF} current is from RA3 pin or V_{DD} pin, whichever is selected as reference input.
- 3:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.
- 4:** Maximum allowed impedance for analog voltage source is 10 kΩ. This requires higher acquisition time.

FIGURE 17-19: A/D CONVERSION TIMING

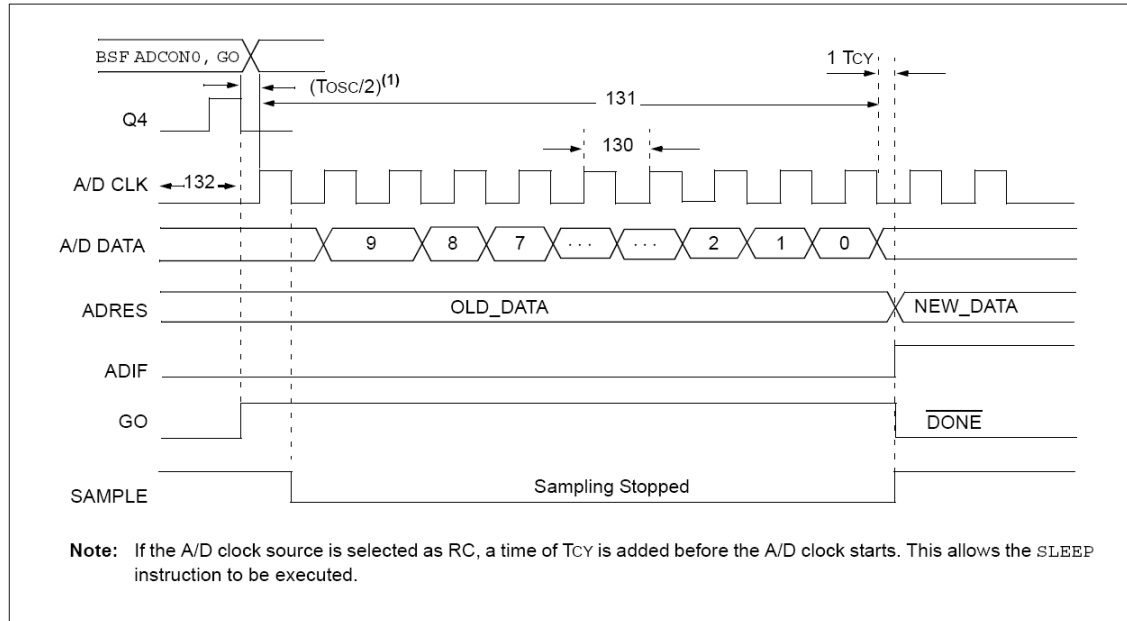


TABLE 17-15: A/D CONVERSION REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions	
130	TAD	A/D Clock Period	PIC16F87XA	1.6	—	—	μs	TOSC based, VREF ≥ 3.0V
			PIC16LF87XA	3.0	—	—	μs	TOSC based, VREF ≥ 2.0V
			PIC16F87XA	2.0	4.0	6.0	μs	A/D RC mode
			PIC16LF87XA	3.0	6.0	9.0	μs	A/D RC mode
131	TCNV	Conversion Time (not including S/H time) (Note 1)		—	12	TAD		
132	TACQ	Acquisition Time	(Note 2)	40	—	μs	The minimum time is the amplifier settling time. This may be used if the “new” input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).	
			10*	—	—	μs		
134	TGO	Q4 to A/D Clock Start	—	$T_{OSC}/2$ §	—	—	If the A/D clock source is selected as RC, a time of T_{CY} is added before the A/D clock starts. This allows the <code>SLEEP</code> instruction to be executed.	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification ensured by design.

Note 1: ADRES register may be read on the following T_{CY} cycle.

Note 2: See Section 11.1 “A/D Acquisition Requirements” for minimum conditions.

TABLE 17-10: I²C BUS START/STOP BITS REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions	
90	TSU:STA	Start condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
		Setup time	400 kHz mode	600	—	—		
91	THD:STA	Start condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
		Hold time	400 kHz mode	600	—	—		
92	TSU:STO	Stop condition	100 kHz mode	4700	—	—	ns	
		Setup time	400 kHz mode	600	—	—		
93	THD:STO	Stop condition	100 kHz mode	4000	—	—	ns	
		Hold time	400 kHz mode	600	—	—		

FIGURE 17-16: I²C BUS DATA TIMING

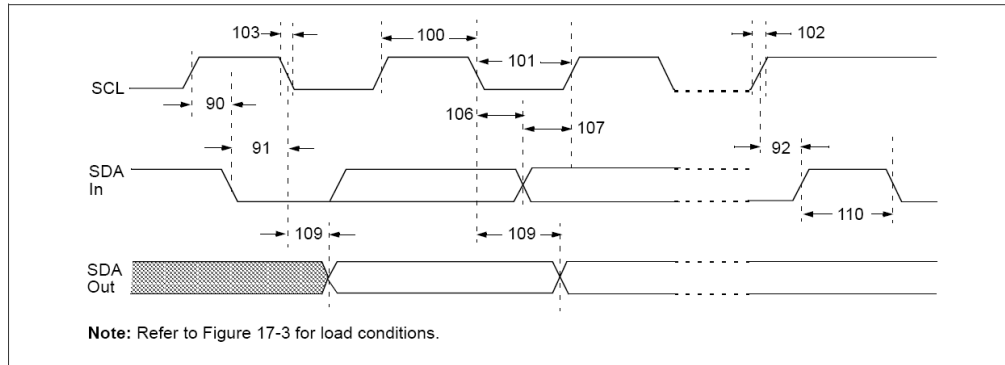


TABLE 17-11: I²C BUS DATA REQUIREMENTS

Param No.	Sym	Characteristic	Min	Max	Units	Conditions	
100	T _{HIGH}	Clock High Time	100 kHz mode	4.0	—	μs	
			400 kHz mode	0.6	—	μs	
			SSP Module	0.5 T _{CY}	—		
101	T _{LOW}	Clock Low Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	1.3	—	μs	
			SSP Module	0.5 T _{CY}	—		
102	T _R	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 C _B	300	ns	C _B is specified to be from 10 to 400 pF
103	T _F	SDA and SCL Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 C _B	300	ns	C _B is specified to be from 10 to 400 pF
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μs	
91	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	T _A A	Output Valid from Clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	—	ns	
110	T _B UF	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
CB		Bus Capacitive Loading	—	400	pF		

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A fast mode (400 kHz) I²C bus device can be used in a standard mode (100 kHz) I²C bus system, but the requirement that, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, T_R MAX. + TSU:DAT = 1000 + 250 = 1250 ns (according to the standard mode I²C bus specification),

ANEXO 2

HOJA DE DATOS DEL RELOJ EN TIEMPO REAL DS12887A



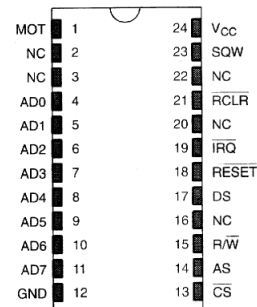
DS12887A Real Time Clock

www.dalsemi.com

FEATURES

- Drop-in replacement for IBM AT computer clock/calendar
- Pin compatible with the MC146818B and DS1287A
- Totally nonvolatile with over 10 years of operation in the absence of power
- Self-contained subsystem includes lithium, quartz, and support circuitry
- Counts seconds, minutes, hours, days, day of the week, date, month, and year with leap year compensation valid up to 2100
- Binary or BCD representation of time, calendar, and alarm
- 12- or 24-hour clock with AM and PM in 12-hour mode
- Daylight Savings Time option
- Selectable between Motorola and Intel bus timing
- Multiplex bus for pin efficiency
- Interfaced with software as 128 RAM locations
 - 14 bytes of clock and control registers
 - 114 bytes of general purpose RAM
- Programmable square wave output signal
- Bus-compatible interrupt signals ($\overline{\text{IRQ}}$)
- Three interrupts are separately software-maskable and testable
 - Time-of-day alarm once/second to once/day
 - Periodic rates from 122 μs to 500 ms
 - End of clock update cycle

PIN ASSIGNMENT



24-PIN ENCAPSULATED PACKAGE

PIN DESCRIPTION

AD0-AD7	- Multiplexed Address/Data Bus
NC	- No Connection
MOT	- Bus Type Selection
$\overline{\text{CS}}$	- Chip Select
AS	- Address Strobe
R/W	- Read/Write Input
DS	- Data Strobe
$\overline{\text{RESET}}$	- Reset Input
$\overline{\text{IRQ}}$	- Interrupt Request Output
SQW	- Square Wave Output
V _{CC}	- +5 Volt Supply
$\overline{\text{RCLR}}$	- RAM Clear
GND	- Ground

DESCRIPTION

The DS12887A Real Time Clock plus RAM is designed to be a direct replacement for the DS1287A. The DS12887A is identical in form, fit, and function to the DS1287A, and has an additional 64 bytes of general purpose RAM. Access to this additional RAM space is determined by the logic level presented on AD6 during the address portion of an access cycle. The $\overline{\text{RCLR}}$ pin is used to clear (set to logic 1) all 114 bytes of general purpose RAM but does not affect the RAM associated with the real time clock. In order to clear the RAM, $\overline{\text{RCLR}}$ must be forced to an input logic "0" (-0.3 to +0.8 volts) during battery back-up mode when V_{CC} is not applied.

The $\overline{\text{RCLR}}$ function is designed to be used via human interface (shorting to ground manually or by switch) and not to be driven with external buffers. All other operation, description and specification is identical to the DS12887.

ANEXO 3

HOJA DE DATOS DE LA EEPROM SERIAL 24LC512



MICROCHIP 24AA512/24LC512/24FC512

512K I²C™ CMOS Serial EEPROM

Device Selection Table

Part Number	Vcc Range	Max. Clock Frequency	Temp. Ranges
24AA512	1.8-5.5V	400 kHz ⁽¹⁾	I
24LC512	2.5-5.5V	400 kHz	I, E
24FC512	2.5-5.5V	1 MHz	I

Note 1: 100 kHz for Vcc < 2.5V

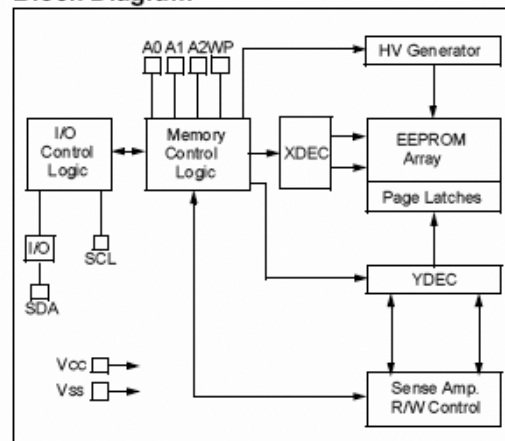
Features:

- Low-power CMOS technology:
 - Maximum write current 5 mA at 5.5V
 - Maximum read current 400 µA at 5.5V
 - Standby current 100 nA typical at 5.5V
- 2-wire serial interface bus, I²C™ compatible
- Cascadable for up to eight devices
- Self-timed erase/write cycle
- 128-byte Page Write mode available
- 5 ms max. write cycle time
- Hardware write-protect for entire array
- Schmitt Trigger inputs for noise suppression
- 1,000,000 erase/write cycles
- Electrostatic discharge protection > 4000V
- Data retention > 200 years
- 8-pin PDIP, SOIC (208 mil), and DFN packages
- 14-lead TSSOP package
- Standard and Pb-free finishes available
- Temperature ranges:
 - Industrial (I): -40°C to +85°C
 - Automotive (E): -40°C to +125°C

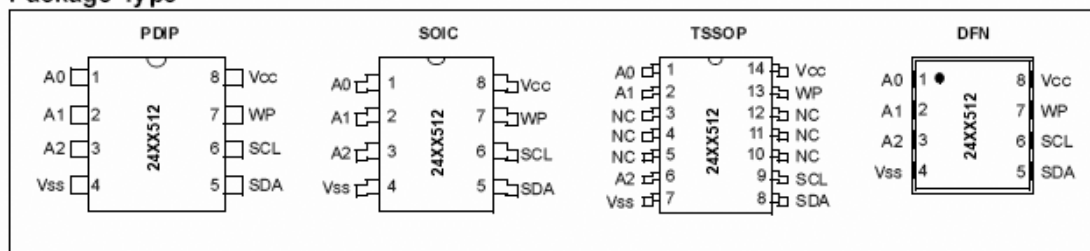
Description:

The Microchip Technology Inc. 24AA512/24LC512/24FC512 (24XX512*) is a 64K x 8 (512 Kbit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low-power applications such as personal communications and data acquisition. This device also has a page write capability of up to 128 bytes of data. This device is capable of both random and sequential reads up to the 512K boundary. Functional address lines allow up to eight devices on the same bus, for up to 4 Mbit address space. This device is available in the standard 8-pin plastic DIP, SOIC, DFN and 14-lead TSSOP packages.

Block Diagram



Package Type



* 24XX512 is used in this document as a generic part number for the 24AA512/24LC512/24FC512 devices.

1.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings (†)

V _{CC}	6.5V
All inputs and outputs w.r.t. V _{SS}	-0.6V to V _{CC} +1.0V
Storage temperature.....	-65°C to +150°C
Ambient temperature with power applied.....	-40°C to +125°C
ESD protection on all pins.....	≥ 4 kV

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 1-1: DC CHARACTERISTICS

DC CHARACTERISTICS			Electrical Characteristics:			
			Industrial (I): V _{CC} = +1.8V to 5.5V		T _A = -40°C to +85°C	
			Automotive (E): V _{CC} = +2.5V to 5.5V		T _A = -40°C to +125°C	
Param. No.	Sym	Characteristic	Min	Max	Units	Conditions
D1	—	A0, A1, A2, SCL, SDA and WP pins:	—	—	—	—
D2	V _{IH}	High-level input voltage	0.7 V _{CC}	—	V	—
D3	V _{IL}	Low-level input voltage	—	0.3 V _{CC} 0.2 V _{CC}	V	V _{CC} ≥ 2.5V V _{CC} < 2.5V
D4	V _{HYS}	Hysteresis of Schmitt Trigger inputs (SDA, SCL pins)	0.05 V _{CC}	—	V	V _{CC} ≥ 2.5V (Note)
D5	V _{OL}	Low-level output voltage	—	0.40	V	I _{OL} = 3.0 ma @ V _{CC} = 4.5V I _{OL} = 2.1 ma @ V _{CC} = 2.5V
D6	I _{LI}	Input leakage current	—	±1	μA	V _{IN} = V _{SS} or V _{CC} , WP = V _{SS} V _{IN} = V _{SS} or V _{CC} , WP = V _{CC}
D7	I _{LO}	Output leakage current	—	±1	μA	V _{OUT} = V _{SS} or V _{CC}
D8	C _{IN} , C _{OUT}	Pin capacitance (all inputs/outputs)	—	10	pF	V _{CC} = 5.0V (Note) T _A = 25°C, f _c = 1 MHz
D9	I _{CC} Read	Operating current	—	400	μA	V _{CC} = 5.5V, SCL = 400 kHz
	I _{CC} Write		—	5	mA	V _{CC} = 5.5V
D10	I _{CCS}	Standby current	—	1	μA	T _A = -40°C to +85°C SCL = SDA = V _{CC} = 5.5V A0, A1, A2, WP = V _{SS}
			—	5	μA	T _A = -40°C to +125°C SCL = SDA = V _{CC} = 5.5V A0, A1, A2, WP = V _{SS}

Note: This parameter is periodically sampled and not 100% tested.

TABLE 1-2: AC CHARACTERISTICS

AC CHARACTERISTICS			Electrical Characteristics:			
			Industrial (I):		V _{CC} = +1.8V to 5.5V	T _A = -40°C to +85°C
			Automotive (E):		V _{CC} = +2.5V to 5.5V	T _A = -40°C to +125°C
Param. No.	Sym	Characteristic	Min	Max	Units	Conditions
1	F _{CLK}	Clock frequency	—	100 400 1000	kHz	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
2	T _{HIGH}	Clock high time	4000 600 500	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
3	T _{LOW}	Clock low time	4700 1300 500	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
4	T _R	SDA and SCL rise time (Note 1)	— — —	1000 300 300	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
5	T _F	SDA and SCL fall time (Note 1)	— —	300 100	ns	All except, 24FC512 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
6	T _{HD:STA}	Start condition hold time	4000 600 250	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
7	T _{SU:STA}	Start condition setup time	4700 600 250	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
8	T _{HD:DAT}	Data input hold time	0	—	ns	(Note 2)
9	T _{SU:DAT}	Data input setup time	250 100 100	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
10	T _{SU:STO}	Stop condition setup time	4000 600 250	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
11	T _{SU:WP}	WP setup time	4000 600 600	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
12	T _{HD:WP}	WP hold time	4700 1300 1300	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
13	T _{AA}	Output valid from clock (Note 2)	— — —	3500 900 400	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
14	T _{BUF}	Bus free time: Time the bus must be free before a new transmission can start	4700 1300 500	— — —	ns	1.8V ≤ V _{CC} < 2.5V 2.5V ≤ V _{CC} ≤ 5.5V 2.5V ≤ V _{CC} ≤ 5.5V 24FC512
16	T _{SP}	Input filter spike suppression (SDA and SCL pins)	—	50	ns	All except, 24FC512 (Notes 1 and 3)
17	T _{WC}	Write cycle time (byte or page)	—	5	ms	—
18	—	Endurance	1,000,000	—	cycles	25°C (Note 4)

Note 1: Not 100% tested. C_B = total capacitance of one bus line in pF.

Note 2: As a transmitter, the device must provide an internal minimum delay time to bridge the undefined region (minimum 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

Note 3: The combined T_{SP} and V_{HYS} specifications are due to new Schmitt Trigger inputs which provide improved noise spike suppression. This eliminates the need for a T_I specification for standard operation.

Note 4: This parameter is not tested but ensured by characterization. For endurance estimates in a specific application, please consult the Total Endurance™ Model which can be obtained from Microchip's web site: www.microchip.com.

ANEXO 4

HOJA DE DATOS DEL INTERRUPTOR ANALÓGICO ADG201A



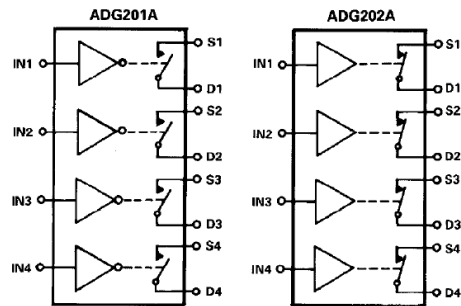
LC²MOS Quad SPST Switches

ADG201A/ADG202A

FEATURES

- 44V Supply Maximum Rating**
- ±15V Analog Signal Range**
- Low R_{ON} (60Ω)**
- Low Leakage (0.5nA)**
- Break Before Make Switching**
- Extended Plastic Temperature Range**
(-40°C to +85°C)
- Low Power Dissipation (33mW)**
- Available in 16-Lead DIP/SOIC and 20-Lead PLCC/LCCC Packages**
- Superior Second Source:**
ADG201A Replaces DG201A, HI-201
ADG202A Replaces DG202

FUNCTIONAL BLOCK DIAGRAMS



SWITCHES SHOWN FOR A LOGIC "1" INPUT

GENERAL DESCRIPTION

The ADG201A and ADG202A are monolithic CMOS devices comprising four independently selectable switches. They are designed on an enhanced LC²MOS process which gives an increased signal handling capability of ±15V. These switches also feature high switching speeds and low R_{ON}.

The ADG201A and ADG202A consist of four SPST switches. They differ only in that the digital control logic is inverted. All devices exhibit break before make switching action. Inherent in the design is low charge injection for minimum transients when switching the digital inputs.

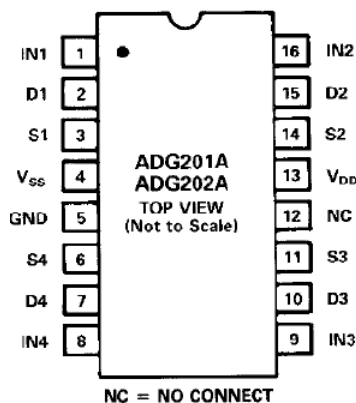
PRODUCT HIGHLIGHTS

- 1. Extended Signal Range:**
These switches are fabricated on an enhanced LC²MOS process, resulting in high breakdown and an increased analog signal range of ±15V.
- 2. Single Supply Operation:**
For applications where the analog signal is unipolar (0V to 15V), the switches can be operated from a single +15V supply.
- 3. Low Leakage:**
Leakage currents in the range of 500pA make these switches suitable for high precision circuits. The added feature of Break before Make allows for multiple outputs to be tied together for multiplexer applications while keeping leakage errors to a minimum.

ADG201A IN	ADG202A IN	SWITCH CONDITION
0	1	ON
1	0	OFF

Table 1. Truth Table

DIP, SOIC



ADG201A/ADG202A — SPECIFICATIONS (V_{DD} = +15V, V_{SS} = -15V, unless otherwise specified)

Parameter	K Version -40°C to +85°C		B Version -40°C to +85°C		T Version -55°C to +125°C		Units	Test Conditions
	25°C		25°C		25°C			
ANALOG SWITCH								
Analog Signal Range	±15	±15	±15	±15	±15	±15	Volts	-10V ≤ V _S ≤ +10V I _{DS} = 1.0mA Test Circuit 1
R _{ON}	60		60		60		Ω typ	
	90	145	90	145	90	145	Ω max	
R _{ON} vs. V _D (V _S)	20		20		20		% typ	V _S = 0V, I _{DS} = 1mA
R _{ON} Drift	0.5		0.5		0.5		%/°C typ	
R _{ON} Match	5		5		5		% typ	
I _S (OFF)	0.5		0.5		0.5		nA typ	V _D = ±14V; V _S = 14V; Test Circuit 2
OFF Input Leakage	2	100	2	100	1	100	nA max	
I _D (OFF)	0.5		0.5		0.5		nA typ	V _D = ±14V; V _S = 14V; Test Circuit 2
OFF Output Leakage	2	100	2	100	1	100	nA max	
I _D (ON)	0.5		0.5		0.5		nA typ	V _D = ±14V; Test Circuit 3
ON Channel Leakage	2	200	2	200	1	200	nA max	
DIGITAL CONTROL								
V _{INH} , Input High Voltage	2.4		2.4		2.4		V min	
V _{INL} , Input Low Voltage	0.8		0.8		0.8		V max	
I _{INL} or I _{INH}	1		1		1		μA max	
DYNAMIC CHARACTERISTICS								
t _{OPEN}	30		30		30		ns typ	Test Circuit 4 Test Circuit 4 V _S = 10V(p-p); f = 100kHz R _L = 75Ω; Test Circuit 6 Test Circuit 7
t _{ON} ¹	300		300		300		ns max	
t _{OFF} ¹	250		250		250		ns max	
OFF Isolation	80		80		80		dB typ	
Channel-to-Channel Crosstalk	80		80		80		dB typ	
C _S (OFF)	5		5		5		pF typ	
C _D (OFF)	5		5		5		pF typ	
C _D , C _S (ON)	16		16		16		pF typ	
C _{IN} , Digital Input Capacitance	5		5		5		pF typ	R _S = 0Ω; C _L = 1000pF; V _S = 0V Test Circuit 5
Q _{INJ} Charge Injection	20		20		20		pC typ	
POWER SUPPLY								
I _{DD}	0.6		0.6		0.6		mA typ	Digital Inputs = V _{INL} or V _{INH}
I _{DD}		2		2		2	mA max	
I _{SS}	0.1		0.1		0.1		mA typ	
I _{SS}		0.2		0.2		0.2	mA max	
Power Dissipation		33		33		33	mW max	

NOTES

¹Sample tested at 25°C to ensure compliance.
Specifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

(T_A = +25°C unless otherwise stated)

V _{DD} to V _{SS}	44V
V _{DD} to GND	25V
V _{SS} to GND	-25V
Analog Inputs¹	
Voltage at S, D	V _{SS} - 0.3V to V _{DD} + 0.3V
Continuous Current, S or D	30mA
Pulsed Current S or D	
I _{ms} Duration, 10% Duty Cycle	70mA
Digital Inputs¹	
Voltage at IN	V _{SS} - 2V to V _{DD} + 2V or 20mA, Whichever Occurs First

Power Dissipation (Any Package)

Up to +75°C	470mW
Derates above +75°C by	6mW/°C

Operating Temperature

Commercial (K Version)	-40°C to +85°C
Industrial (B Version)	-40°C to +85°C
Extended (T Version)	-55°C to +125°C
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering 10sec)	+300°C

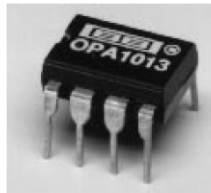
NOTE

¹Overvoltage at IN, S or D will be clamped by diodes. Current should be limited to the Maximum Rating above.

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Only one Absolute Maximum Rating may be applied at

ANEXO 5

HOJA DE DATOS DEL AMPLIFICADOR OPERACIONAL DUAL PA1013



OPA1013

Precision, Single-Supply DUAL OPERATIONAL AMPLIFIER

FEATURES

- SINGLE POWER SUPPLY OPERATION
- INPUT VOLTAGE RANGE TO GROUND
- OUTPUT SWINGS NEAR GROUND
- LOW QUIESCENT CURRENT: 550 μ A max
- LOW V_{OS} : 300 μ V max
- LOW DRIFT: 2.5 μ V/ $^{\circ}$ C max
- LOW I_{OS} : 1.5nA max
- LOW NOISE: 0.55 μ Vp-p, 0.1Hz to 10Hz

APPLICATIONS

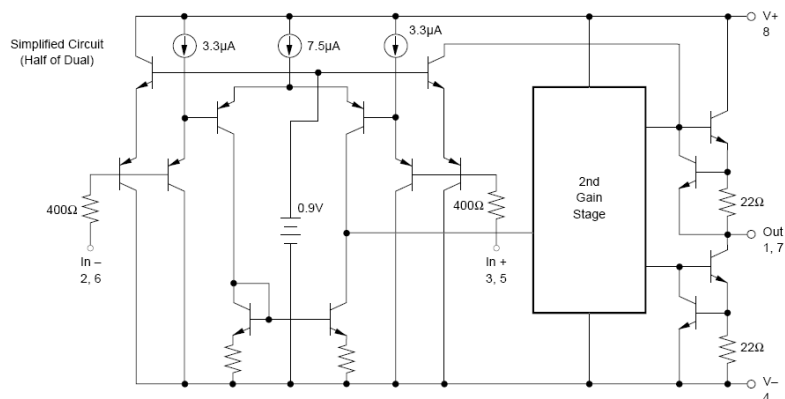
- PRECISION INSTRUMENTATION
- BATTERY-POWERED EQUIPMENT
- BRIDGE AMPLIFIERS
- 4-20mA CURRENT TRANSMITTERS
- VOLTAGE COMPARATOR

DESCRIPTION

The OPA1013 dual operational amplifier provides precision performance in single power supply and low power applications. It is laser trimmed for low offset voltage and drift, greatly reducing the large errors common with LM324-type op amps. Input offset current is also trimmed to reduce errors in high impedance applications.

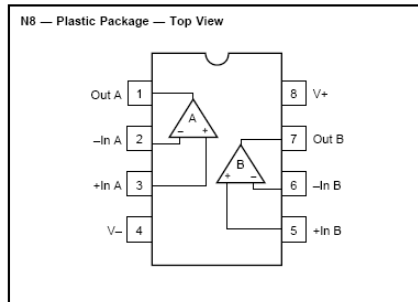
The OPA1013 is characterized for operation at both +5V (single supply) and ± 15 V power supplies. When operated from a single supply, the input common-mode range includes ground and the output can swing to within 15mV of ground. Completely independent biasing networks eliminate interaction between the two amplifiers—even when one is used as a comparator.

The OPA1013 is available in an 8-pin plastic DIP specified for the 0 $^{\circ}$ C to +70 $^{\circ}$ C temperature range.



International Airport Industrial Park • Mailing Address: PO Box 11400 • Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd. • Tucson, AZ 85706
Tel: (520) 746-1111 • Twx: 910-952-1111 • Cable: BBRCORP • Telex: 066-6491 • FAX: (520) 889-1510 • Immediate Product Info: (800) 548-6132

CONNECTION DIAGRAM



SPECIFICATIONS

ELECTRICAL

$V_S = \pm 15V$, $V_{CM} = 0V$, $T_A = +25^\circ C$ unless otherwise noted.

PARAMETER	CONDITION	OPA1013CN8			OPA1013DN8			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Input Offset Voltage			±50	±300		±200	±800	μV
Time Stability			0.5			*		μV/Mo
Input Offset Current			±0.08	±1.5		*	*	nA
Input Bias Current			7	30		*	*	nA
Voltage Noise, BW = 0.1 to 10Hz			0.55			*		μVp-p
Noise Density, f = 10Hz			28			*		nV/√Hz
f = 1kHz			25			*		nV/√Hz
Current Noise Density, f = 10Hz			0.12			*		pA/√Hz
Input Resistance: Differential		70	300		*	*		MΩ
Input Resistance: Common-Mode			4		*	*		GΩ
Open-Loop Voltage Gain	$V_O = \pm 10V$, $R_L = 2k\Omega$	1.2	2.9		*	*		V/μV
	$V_O = \pm 10V$, $R_L = 600\Omega$	0.5	1.9		*	*		V/μV
Common-Mode Input Range		+13.5	+13.8		*	*		V
		-15	-15.3		*	*		V
Common-Mode Rejection	$V_{CM} = +13.5$ to $-15V$	97	114		*	*		dB
Power Supply Rejection	$V_S = \pm 2$ to $\pm 18V$	100	117		*	*		dB
Channel Separation	$V_O = \pm 10V$, $R_L = 2k\Omega$	120	137		*	*		dB
Voltage Output	$R_L = 2k\Omega$	±12.5	±14		*	*		V
Slew Rate		0.2	0.35		*	*		V/μs
Quiescent Current (per amplifier)			±0.35	±0.55		*	*	mA

*Specification same as OPA1013CN8.

$V_S = +5V/0V$, $V_{CM} = 0V$, $V_O = +1.4V$, $T_A = +25^\circ C$ unless otherwise noted.

PARAMETER	CONDITION	OPA1013CN8			OPA1013DN8			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Input Offset Voltage			±90	±450		±250	±950	μV
Input Offset Current			±3.0	±2.0		*	*	nA
Input Bias Current			10	50		*	*	nA
Open-Loop Voltage Gain	$V_O = 5mV$ to $4V$ $R_L = 500\Omega$		0.1			*		V/μV
Common-Mode Input Range		+3.5	+3.8		*	*		V
		0	-0.3		*	*		V
Voltage Output Low	No Load		15	25		*	*	mV
Low	$R_L = 600\Omega$ to Ground		5	10		*	*	mV
Low	$I_{SINK} = 1mA$		200	350		*	*	mV
High	No Load	4	4.4		*	*		V
High	$R_L = 600\Omega$ to Ground	3.4	4		*	*		V
Quiescent Current (per amplifier)			0.33	0.5		*	*	mA

*Specification same as OPA1013CN8.

$T_A = 0^\circ C$ to $+70^\circ C$, $V_S = \pm 15V$, $V_{CM} = 0V$ unless otherwise noted.

PARAMETER	CONDITION	OPA1013CN8			OPA1013DN8			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
Input Offset Voltage			±80	±400		±230	±1000	μV
Input Offset Voltage Drift ⁽¹⁾	$V_S = +5/0V$, $V_O = +1.4V$		±110	±570		±280	±1200	μV/°C
Input Offset Current			0.4	2.5		0.7	5	μV/°C
Input Bias Current	$V_S = +5/0V$, $V_O = +1.4V$		±0.3	±2.8		*	*	nA
			±0.5	±6		*	*	nA
Input Bias Current	$V_S = +5/0V$, $V_O = +1.4V$		9	38		*	*	nA
			13	90		*	*	nA
Open-Loop Voltage Gain	$V_O = \pm 10V$, $R_L = 2k\Omega$	0.7	2.2		*	*		V/μV
Common-Mode Rejection	$V_{CM} = +13$ to $-15V$	94	113		*	*		dB
Power Supply Rejection	$V_S = \pm 2$ to $\pm 18V$	97	116		*	*		dB
Voltage Output	$R_L = 2k\Omega$	±12.0	±13.9		*	*		V
V_O Low	$V_S = +5/0V$, $R_L = 600\Omega$		6	13		*	*	mV
V_O High	$V_S = +5/0V$, $R_L = 600\Omega$	3.2	3.9		*	*		V
Quiescent Current (per amplifier)	$V_S = +5/0V$, $V_O = +1.4V$		±0.37	±0.6		*	*	mA
			0.34	0.55		*	*	mA

*Specification same as OPA1013CN8.

NOTE: (1) Guaranteed by design. This specification is established to a 98% confidence level.

ANEXO 6

HOJA DE DATOS DEL FOTODIODO DE SILICIO FDS100

THORLABS INC435 Route 206 • P.O. Box 366
Newton, NJ 07860-0366Ph. 973-579-7227
FAX 973-300-3600

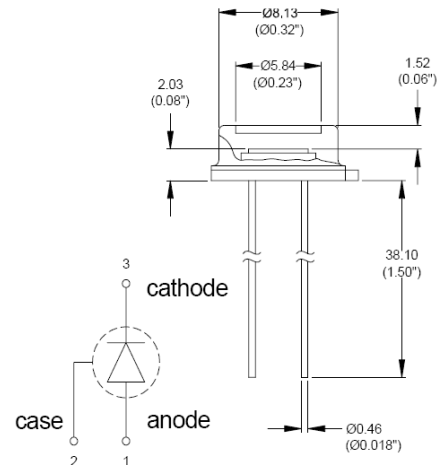
FDS100 Si Photodiode

High Speed
Large Active Area

The FDS100 is a high-speed silicon photodiode with a spectral response from 350nm to over 1100nm. This photodiode has a PIN structure that provides fast rise and fall times with a bias of 20V.

Electrical Characteristics

Spectral Response:	350-1100nm
Active Area:	13.0mm ²
Rise Time (RL=50Ω):	10ns (20V bias)
Fall Time (RL=50Ω):	10ns (20V bias)
NEP@900nm:	1.2 x 10 ⁻¹⁴ W/√Hz (@20V bias)
Dark Current:	20nA max (20V)
Package:	T05, 0.36" can

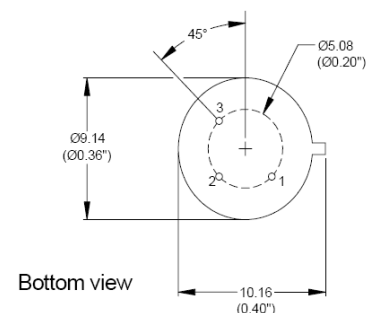


Maximum Ratings

Damage Threshold CW:	100 mW/cm ²
Damage 10ns Pulse:	500mJ/cm ²
Max Bias Voltage:	25V

Pin Description

1. Laser anode
2. Laser case
3. Laser cathode



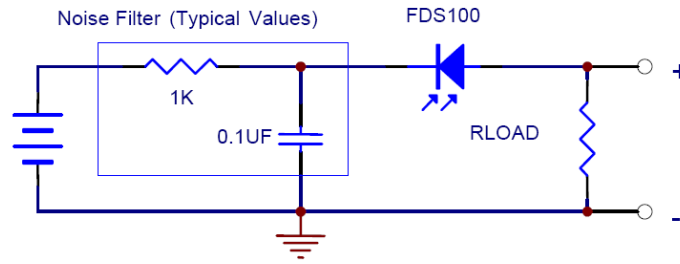
The Thorlabs FDS100 photodiode is ideal for measuring both pulsed and CW light sources, by converting the optical power to an electrical current. The Si detector is housed in a T05 can, with an anode, cathode and case connection. The photodiode anode produces a current, which is a function of the incident light power and the wavelength. The responsivity $\mathfrak{R}(\lambda)$, can be read from **Figure 1** to estimate the amount of photocurrent to expect. This can be converted to a voltage by placing a load resistor (R_{LOAD}) from the photodiode anode to the circuit ground. The output voltage is derived as:

$$V_O = P * \mathfrak{R}(\lambda) * R_{LOAD}$$

The bandwidth, f_{BW} , and the rise time response, t_R , are determined from the diode capacitance, C_J , and the load resistance, R_{LOAD} , as shown below. Placing a bias voltage from the photo diode cathode to the circuit ground can lower the photo diode capacitance.

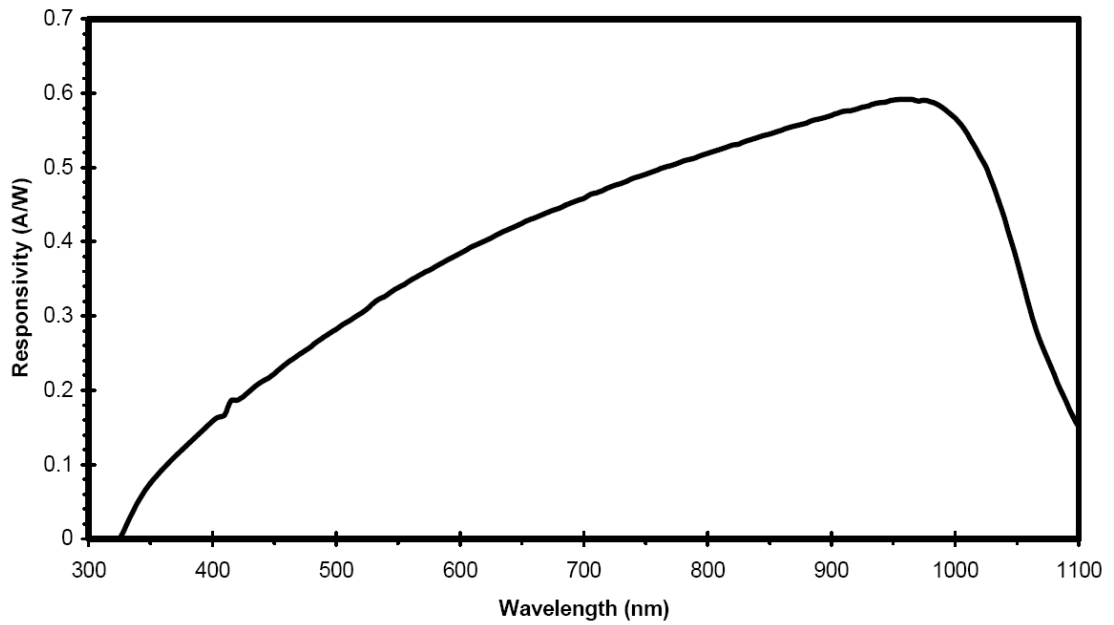
$$f_{BW} = 1/(2\pi * R_{LOAD} * C_J), t_R = 0.35/f_{BW}$$

Typical Circuit Diagram



Typical Plots

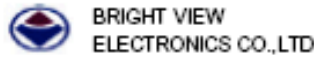
Figure 1 - FDS100 Spectral Responsivity Curve



Typical Responsivity Curve using Thorlabs calibration services.

ANEXO 7

HOJA DE DATOS DEL LED ROJO BVU-5G1RD4



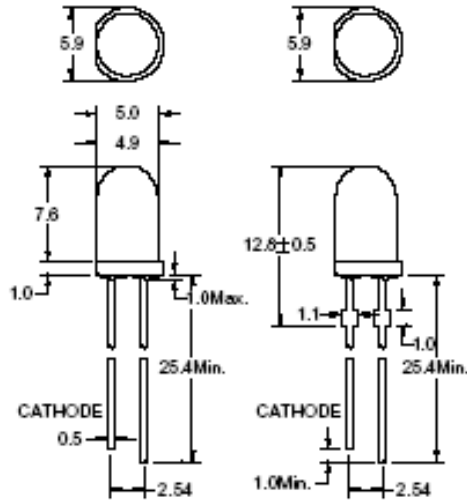
HIGH BRIGHTNESS LED LAMP

BVU-5G1RD4

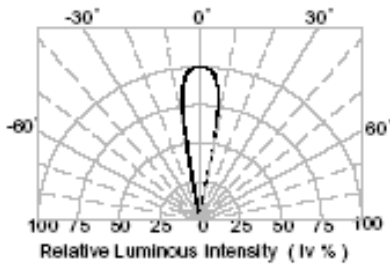
PACKAGE CONFIGURATION

DESCRIPTION

Dice Material : GaAlAs/GaAs Red
 Light Color : Red Color
 Lens Color : Water Transparent
 Stand-Off P/N : BVU-5G1RD4 R



RADIATION PATTERN



Tolerance $\pm 0.25 \text{ mm}$

ABSOLUTE MAXIMUM RATINGS AT Ta = 25 °C

PARAMETER	MAX.	UNIT
Power Dissipation	100	mW
Continuous Forward Current	40	mA
Peak Forward Current (1/10 Duty Cycle , 0.1ms Pulse Width)	200	mA
Reverse Voltage	5	V
Derating Linear From 50 °C	0.4	mA/°C
Operating Temperature Range	-40 °C to + 100 °C	
Storage Temperature Range	-40 °C to + 100 °C	
Lead Solder Temperature 1.6 mm Below Package 260 °C for 5 seconds		

ELECTRICAL / OPTICAL CHARACTERISTICS AT Ta = 25 °C

SYMBOL	PARAMETER	TEST COND.	MIN.	TYP.	MAX.	UNIT
V _F	Forward Voltage	I _F = 20 mA		1.9	2.2	V
I _R	Reverse Current	V _R = 5V			100	μ A
λ_p	Peak Emission Wavelength	I _F = 20 mA		660		nm
λ_d	Dominant Wavelength	I _F = 20 mA		643		nm
2 θ 1/2	Viewing Angle	I _F = 20 mA		20		Deg

BIN GRADE LIMITS (I_F = 20 mA) LUMINOUS INTENSITY / mcd

Bin	F	G	H	I	J	K
Min.	360	465	600	780	1000	1300
Max.	465	600	780	1000	1300	1680

Tolerance $\pm 15\% \text{ mcd}$

*Bright View reserves the rights to alter specifications and remove availability of products at any time without notice.

*Dominant Wavelength, λ_d is according to CIE Chromaticity Diagram base on color of lamps.

* θ 1/2 is the off-axis angle where the luminous intensity is one half the on-axis intensity.

2001/11/13 -A

ANEXO 8

HOJA DE DATOS DEL LED INFRARROJO SLI – 0308CP

INFRARED LAMP SPECS

<i>PRODUCT MODEL</i>	<i>SLI-0308CP</i>	<i>DRWAWING NO.</i>	<i>DSI-95050401</i>
<i>CHECKED BY</i>	<i>Eva Chang</i>	<i>DRAWING BY.</i>	<i>Aite Fu</i>
		<i>DATE</i>	<i>95,05,04</i>

ABSOLUTE NAXIMUM RATINGS (Ta=25°C)

<i>CHARACTERISTIC</i>	<i>SYMBOL</i>	<i>MAX.</i>	<i>UNIT</i>
<i>POWER DISSIPATION</i>	<i>PO</i>	<i>200</i>	<i>mW</i>
<i>CONTINUOUS FORWARD CURRENT</i>	<i>IF</i>	<i>100</i>	<i>mA</i>
<i>REVERSE VOLTAGE</i>	<i>VR</i>	<i>5</i>	<i>V</i>
<i>PEAK FORWARD CURRENT</i> <i>(300 pps, 1us pulise)</i>	<i>Ipf</i>	<i>1</i>	<i>A</i>
<i>STORAGE TEMPERATURE</i>	<i>Tstg</i>	<i>-20 to +80</i>	<i>°C</i>
<i>OPERATING TEMPERATURE</i>	<i>Topr</i>	<i>-30 to +100</i>	<i>°C</i>
<i>SOLDER TEMPERATURE FOR 3 sec.</i> <i>AT 2 mm FROM THE REFLECTOR EDGE</i>		<i>260</i>	<i>°C</i>

ELECTRO-OPTICAL CHARACTERISTICS (Ta=25°C)

<i>CHARACTERISTICS</i>	<i>SYMBOL</i>	<i>CONDITION</i>	<i>TYP.</i>	<i>MAX.</i>	<i>UNIT</i>
<i>FORWARD VOLTAGE</i>	<i>VF</i>	<i>IF=50mA</i>	<i>1.3</i>	<i>1.7</i>	<i>V</i>
<i>REVERSE CURRENT</i>	<i>IR</i>	<i>VR=5V</i>		<i>10</i>	<i>uA</i>
<i>LUMINOUS INTENSITY</i>	<i>IV</i>	<i>IF=50mA</i>	<i>4.0</i>		<i>mW/str</i>
<i>PEAK EMISSION WAVE LENGTH</i>	λP	<i>IF=20mA</i>	<i>940</i>		<i>nm</i>
<i>SPECTRAL LINE HALF WIDTH</i>	$\Delta \lambda$	<i>IF=20mA</i>	<i>100</i>		<i>nm</i>
<i>VIEWING ANGLE</i>	$\Delta \theta$	<i>IF=20mA</i>	<i>±15</i>		<i>Deg</i>
<i>RISE/FALL TIME</i>	<i>Tr/Tf</i>	<i>IF=20mA</i>	<i>30/30</i>		<i>nS</i>