

Universidad de la República
Facultad de Ingeniería

iTDOAS

Intelligent Target Differential Optical Absorption Spectroscopy

Sergio Mario Blanco Reale
Santiago Conde Maine
Juan Alberto Gentilini Clavijo

Tutores
Dra Erna Frins
Ing Roque Gagliano



Agradecimientos.

Este proyecto no habría sido posible sin el apoyo de muchas personas.

En primer lugar queremos agradecer a nuestras familias. Por brindarnos toda su paciencia y estar allí siempre.

A la Dra. Erna Frins, al Dr. José Ferrari y al Ing. Roque Gagliano, por prestar siempre su atención a la hora de orientarnos, corregir borradores y contestar todas nuestras preguntas.

Al encargado del taller de mecánica fina, Antonio Saez. Quien con su buena onda, su ingenio y buenas ideas, y las horas dedicadas a la mecánica del proyecto, llegó a ser, para nosotros, como un integrante más.

Al grupo de proyecto USB4ALL quienes dedicaron su conocimiento y su tiempo para orientarnos en la programación del PIC.

A Amelia Ferrari por toda su paciencia y su buena predisposición para con sus “pedigüeños”.

A Héctor Gentilini, quien supo escuchar y luego compartir su amplio conocimiento y experiencia “en estas cosas”.

A Alberto Paz, quien nos hizo el gabinete metálico.

Y a todos aquellos que aportaron sus ideas y su buena onda: Nicolas Casaballe (“Nico”), Javiera Salas, Daniela Cuentas, Tania Díaz, Elizabeth Scherschener, Luis Longeau, Marcelo David, Inés Scapinelo, Andrés Aguirre, Rafael Fernández...

...y al PIC, que se bancó mil y una.

Contenido

	Pag.
Resumen	1
1. Espectroscopía de absorción óptica diferencial: DOAS	3
1.1. Introducción a la espectroscopía de absorción	
1.2. La Ley de Lambert-Beer	4
1.3. Principio de Medición	
1.4. Procedimiento de Análisis	5
1.5. Correcciones del Espectro	7
1.5.1. Corrección de Corriente de Oscuridad	
1.5.2. Corrección del Offset Electrónico	8
1.5.3. Efecto de Ring	9
1.5.4. El Efecto Solar I_0	
1.6. Configuraciones Experimentales	10
1.6.1. Configuraciones de DOAS Activo	
1.6.2. Configuración de DOAS Pasivo	12
1.6.2.1. ToTaL DOAS	14
2. Objetivos	15
2.1. Objetivos Generales	
2.2. Objetivos Específicos	
3. Nuestra Solución	17
3.1. Actores	18
3.2. Casos de Uso	19
3.2.1. Proceso de Instalación	
3.2.2. Proceso de Administración	
3.2.3. Proceso de Monitoreo	
3.3. Forma de Trabajo	20
4. Sistema Mecánico	23
4.1. Requerimientos	
4.2. Diseño	
4.3. Implementación del Sistema	
4.3.1. Cabecera	
4.3.2. Interior de la Cabecera	25
4.3.3. Diseño de los Engranajes	
4.3.4. Trípode	28
4.3.5. Servidor	
5. Hardware	29
5.1. Diseño de la Placa de Control	
5.1.1. Entradas y Salidas de la Placa	30
5.1.2. Microcontrolador	
5.1.2.1. Conexión del Microcontrolador	31
5.1.3. Buffer	33
5.1.3.1. Motores de Pasos	
5.1.3.2. Ventilador e indicadores lumínicos	34

5.2. Diseño y Herramientas de Software Utilizado	36
5.3. Construcción de la Placa	37
5.3.1. Descripción de los conectores	38
5.4. Conexión de Hardware entre el Servidor y la Cabecera	41
5.5. Protocolo de Comunicación entre el Servidor y la Placa	42
5.5.1. Descripción de los Comandos	43
5.6. Firmware de Microcontrolador	48
5.6.1. Herramientas de Software	
5.6.2. Enfoque del Problema y Primeros Pasos	49
5.6.3. Implementación del Firmware	
5.6.4. Actualización del Firmware	53
6. Software	55
6.1. Software de Control (Drivers)	
6.1.1. Comunicación USB	
6.1.2. Endpoints	
6.1.3. Driver del espectrómetro	57
6.1.4. Driver de la Placa de Control	59
6.1.5. Permisos	60
6.2. Aplicación Web	61
6.2.1. CGI utilizados	62
6.2.2. CGI para la interfaz de dispositivos móviles	63
6.3. Software Desarrollado por Terceros	64
6.4. Modularización del Software	65
6.4.1. Sistema Operativo	
6.4.2. Webmin	67
6.4.2.1. Configuración	
6.4.3. Cambio de Espectrómetro	68
6.4.4. Agregado de Comandos	70
6.5. Configuración	71
6.5.1. Panorama	
6.5.2. Configuración al Inicio y al Final	72
6.5.3. Configuración de Apache+Webmin	73
7. Plan económico del proyecto	75
8. Gestión de Tiempos	79
8.1. Plan del proyecto	
8.2. Esquema de tiempos reales.	80
9. Consumo eléctrico	83
10. Conclusiones	85
11. Trabajo a futuro	87
Apéndice I : Manual del Usuario	89
Apéndice II.: Instalación del sistema Linux-ITDoas	101

Apéndice III: Motores de Paso y Encoders.	105
Apéndice IV Especificaciones Técnicas	111
Apéndice V Términos y definiciones	113
Apéndice VI Contenido del CD	115
Bibliografía	116

Resumen

Hoy en día, más que nunca, el Uruguay se ve enfrentado a la necesidad de controlar la composición de su atmósfera, debiendo tener la capacidad de detectar la presencia de gases contaminantes.

El grupo de *Óptica Aplicada del Instituto de Física de la Facultad de Ingeniería de la Universidad de la República*, trabaja desde hace algunos años, en el desarrollo y la implementación de sistemas que tienen la capacidad de monitorear las emanaciones gaseosas hacia la atmósfera, utilizando principalmente espectroscopía de absorción óptica diferencial (DOAS, Differential Optical Absorption Spectroscopy).

Para continuar con estas investigaciones, era necesario desarrollar un dispositivo capaz de automatizar el proceso de medición, haciéndolo a su vez mas preciso, confiable y económico .

En este marco, surge el proyecto **ITDoas** “*Intelligent tomographic Target light scattering - Differential Optical Absorption Spectroscopy*”.

Este consiste en la automatización y control en forma remota, de un dispositivo que implementa **Total-DOAS**, método que utiliza la radiación solar difundida, para obtener información de las capas bajas de la atmósfera, permitiendo medir simultáneamente emisiones diversas tales como NO₂, SO₂, formaldehído, radicales, etc.

Aunque existen varios dispositivos que utilizan esta técnica, ITDoas presenta varias ventajas con respecto a éstos, entre ellas, logra una mayor precisión debido a un juego de reducciones de 1:1600, en sus dos grados de libertad de movimiento. Es operable a distancia sin necesidad de software adicional en el cliente, ya que cuenta con una interfaz web, que lo hace controlable desde cualquier dispositivo que contenga un navegador web. Permite la programación de rutinas de trabajo haciendo que opere en forma automática. No necesita conexión física por contar con una interfaz inalámbrica. Es fácilmente desmontable y portátil permitiendo efectuar medidas de campo casi en cualquier lugar. ITDoas es modularizable tanto en software como en hardware contando con interfaces estándar como USB, Ethernet y Wi-Fi.

1. Espectroscopía de Absorción Óptica Diferencial: DOAS

En esta sección se hará una introducción al principio general de la Espectroscopía de Absorción Óptica Diferencial (DOAS, por sus siglas en inglés).

DOAS permite la determinación de concentraciones de diferentes trazas gaseosas sin distorsionar el comportamiento químico de las sustancias a ser investigadas.

La técnica mide concentraciones químicas mediante el macheo de patrones de absorción de moléculas conocidas con los patrones del espectro recabado de la luz de la atmósfera.

Estas trazas pueden ser medidas utilizando una luz artificial, lo que se conoce con el nombre de “Active DOAS” o DOAS activo, o haciendo el análisis espectral de la luz solar difundida en la atmósfera (scattered sunlight), reflejada o directa que recibe el nombre de “Pasive DOAS” o DOAS pasivo.

1.1. Introducción a la espectroscopia de absorción

La espectroscopía es un buen método para analizar la composición química de los gases. Ha jugado un papel muy importante en el descubrimiento de las propiedades físicas y químicas de la atmósfera terrestre en los últimos 100 años.

Las bases del método de medición mediante espectroscopía esta basado en la ley de Lambert-Beer, la cual describe la absorción de la radiación electromagnética cuando ésta atraviesa una sustancia

$$I(\lambda) = I_o(\lambda)e^{-L\sigma(\lambda)c} \quad (1.1)$$

$I_o(\lambda)$ denota la intensidad inicial de un haz de luz de una fuente conocida, $I(\lambda)$ es la intensidad de radiación del haz, luego de haber pasado por un recipiente de largo L que contiene el gas de concentración uniforme c

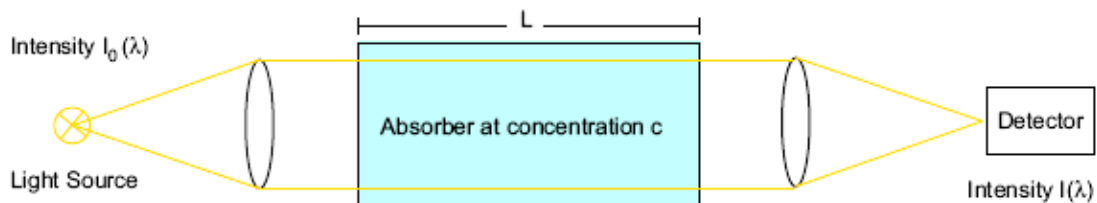


Fig.1-1 Figura extraída de la referencia bibliográfica [1]

La sección eficaz de absorción $\sigma(\lambda)$ a la longitud de onda λ es una propiedad característica de cualquier sustancia. Si la longitud del camino atravesado por la luz es L conocido, el promedio de concentraciones de traza de gas c puede ser calculado mediante la razón

$$c = \frac{\ln\left(\frac{I_o(\lambda)}{I(\lambda)}\right)}{L\sigma(\lambda)} \quad (1.2)$$

Definiéndose como densidad óptica la expresión

$$D = \ln\left(\frac{I_o(\lambda)}{I(\lambda)}\right)$$

La ecuación (1.2) es la base de la mayoría de las aplicaciones de laboratorio sobre espectroscopia de absorción.

Las medidas de las intensidades $I(\lambda)$ e $I_o(\lambda)$ son determinadas mediante medidas con y sin el gas a estudiar en el haz de luz. Sin embargo la verdadera intensidad $I_o(\lambda)$, como sería recibida de la fuente de luz en la ausencia de cualquier gas, es difícil determinar al aire libre. El método de DOAS soluciona este problema midiendo la llamada “absorción diferencial”, esto es, la diferencia de absorción a dos longitudes de onda diferentes.

1.2. La Ley de Lambert-Beer

Cuando medimos trazas de gases atmosféricos al aire libre, la versión simple de la ley de Lambert-Beer tiene que ser modificada. Para poder describir correctamente los procesos en la atmósfera terrestre, tenemos que tener en cuenta múltiples gases, el scattering de Rayleigh y de Mie. Así, la ley de Lambert-Beer sufre las siguientes modificaciones:

$$I(\lambda) = I_o(\lambda) e^{-\int_0^L (\sum_i \sigma_i(\lambda, T, P) c_i(l) + \varepsilon_m(\lambda, l) + \varepsilon_R(\lambda, l)) dl} \quad (1.3)$$

Donde para cada gas traza i , $\sigma_i(\lambda)$ denota la sección eficaz de absorción dependiente de la temperatura T , la presión P y la longitud de onda λ . $C_i(l)$ es la densidad en la posición l a lo largo del camino de longitud total L . Los coeficientes de extinción de Mie y de Rayleigh son $\varepsilon_m(\lambda, l)$ y $\varepsilon_R(\lambda, l)$ respectivamente.

1.3. Principio de Medición

La figura 1-2 muestra un esquema simple del sistema DOAS. La luz emitida por una fuente de luz apropiada es recogida por un telescopio y almacenada en un espectrómetro luego de haber pasado por la masa de aire observada. En el caso del Mini MAX-DOAS, el sol es usado como fuente de luz, pero también se puede utilizar con fuentes de luz artificiales como las usadas en el DOAS activo (por ejemplo lámpara halógenas o de xenón).

En la figura 1-2 (a) se observa el espectro de la luz de intensidad $I(\lambda, L)$ ingresando al espectrómetro bajo la presunción de que el único gas presente en la masa de aire a

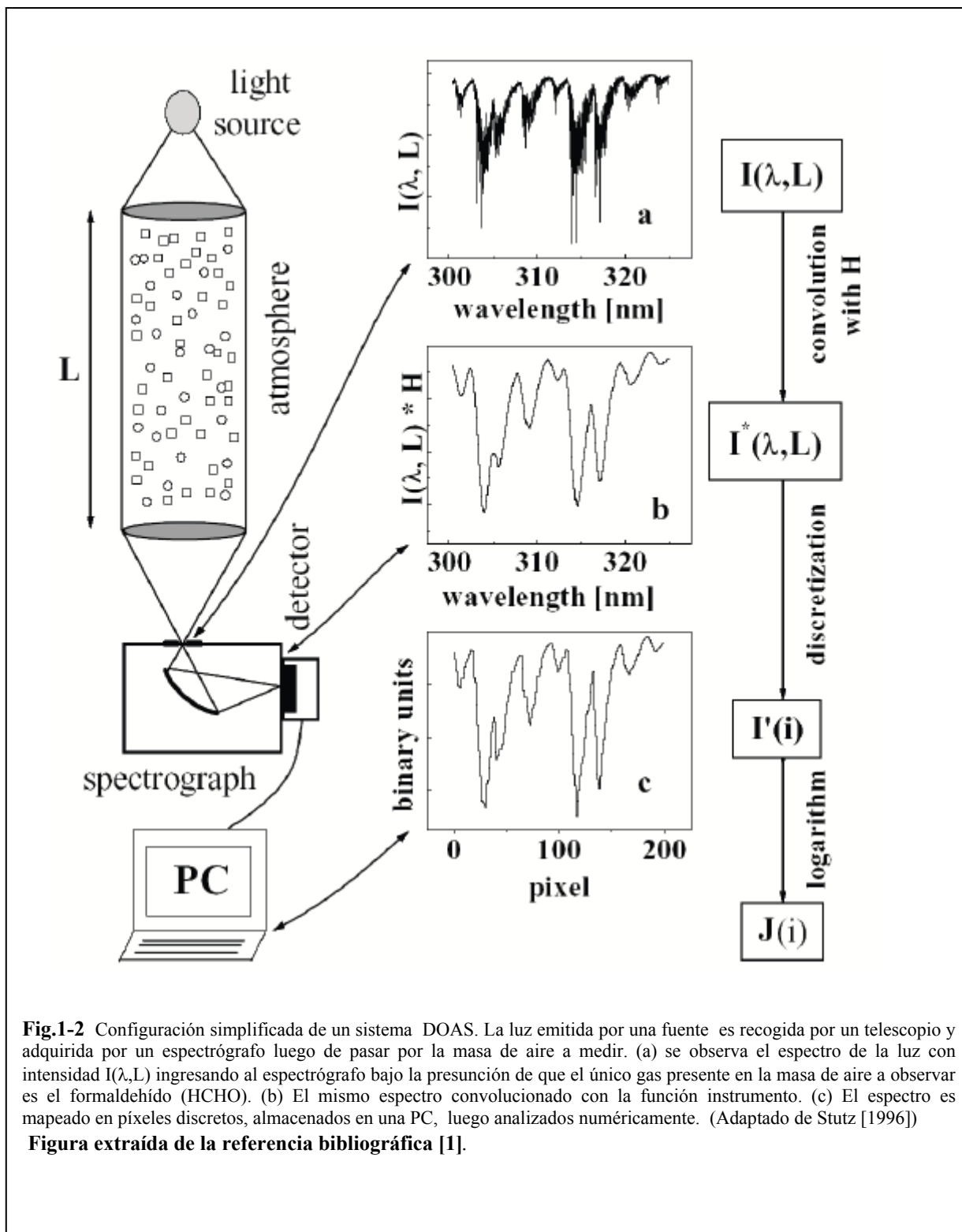
observar es el formaldehído (HCHO). Sin embargo la resolución del espectrógrafo es limitada y como consecuencia de esto la resolución de la luz dispersada también se reduce. Matemáticamente esto se puede describir como la convolución del espectro de absorción de alta resolución con la función de transferencia del espectrógrafo H figura 1-2 (b). Finalmente el espectro es adquirido por un detector, mapeado sobre n píxeles discretos y almacenado en un PC para posterior análisis figura 1-2 (c). Cada píxel i integra la luz en el intervalo de longitud de onda $\lambda(i)$ a $\lambda(i+1)$. Donde el intervalo de longitud de onda es dado por el mapeo longitud de onda - píxel Γ , del instrumento. El ancho espectral de un píxel $\Delta\lambda(i) = \lambda(i+1) - \lambda(i)$ es constante, si la dispersión es lineal: $\Gamma_1: \lambda(i) = \lambda(0) + \gamma_0 \cdot i$. En general Γ_1 puede ser aproximado por un polinomio:

$$\Gamma_I : \lambda(i) = \sum_{K=0}^q \gamma_k \cdot i^k \quad (1.4)$$

El mapeo longitud de onda - píxel se determina por un parámetro (vector) γ_k . Mientras que un cambio en el parámetro γ_0 ocasiona un movimiento espectral del espectro, un cambio en γ_1 resulta en estiramiento, contracción o cambio de parámetros, para altos valores de k causa distorsión de alto orden. Los espectrómetros utilizados para DOAS usualmente muestran una deriva cuando no son térmicamente estabilizados. Por lo tanto, pueden ocurrir cambios en γ_k y tienen que ser corregidos en el procedimiento de análisis.

1.4. Procedimiento de Análisis

El primer paso en el proceso de análisis es la elección del rango de longitudes de onda. Éste tiene que contener las estructuras de absorción de los gases traza de interés y debería incluir varias bandas para evitar interferencias de dos o más especies absorbiendo luz a la misma longitud de onda. En el proceso de análisis, el logaritmo del espectro medido $J(i)$ es modelado tan precisamente como es posible usando secciones eficaces de absorción convolucionados $\sigma_j^*(i)$ de las m especies de gas relevantes para el intervalo de longitudes de onda. El espectro modelado $F(i)$ está dado por una combinación lineal de las respectivas estructuras de absorción.



1.5. Correcciones del Espectro

En esta sección se describirán las correcciones que se tienen que aplicar a los espectros en la evaluación DOAS. Primero nos referiremos a corrección de la tensión de offset y la corriente de oscuridad. Cuando se utilizan luces de gran estructura (Como por ejemplo el sol con fuertes líneas de absorción de Fraunhofer) hay un sinfín de otros efectos a tener en cuenta. El efecto de Ring, que explica el llenado de las líneas de absorción de Fraunhofer por efecto de scattering Raman inelástico en la atmósfera.

Finalmente, se estudiará el efecto I_0 causado por interferencias de líneas de Fraunhofer con estructuras de absorción de las especies gaseosas a ser medidas.

1.5.1. Corrección de Corriente de Oscuridad

Para la detección de los fotones que alcanzan al espectrómetro, se utiliza una CCD (Charged Couple Device). Cada píxel puede ser considerado como un capacitor que produce una descarga proporcional al número de fotones que inciden. Sin embargo, aún en condiciones de oscuridad, la excitación térmica ocasiona corrientes denominadas “corrientes de oscuridad” (dark currents) proporcionales al factor de Boltzmann $I_{DC} \propto e^{-\Delta E/kT}$. La corriente de oscuridad decrece exponencialmente con el descenso de la temperatura del detector. Por ello, enfriar el detector reduce la corriente de oscuridad. Para corregir el remanente en forma digital, se toma un espectro con un barrido o scan, con tiempo de integración muy largo (típicamente 30 seg.) en condiciones de oscuridad. Luego la correspondiente señal de offset es restada. Para cada espectro medido, la señal de corriente de oscuridad tiene que ser restada durante el proceso de evaluación:

$$I_{corr}[n] = I[n] - \frac{t_{int,measure}}{t_{int,dc}} \cdot D_{offcorr}[n],$$

donde $I_{corr}[n]$ es la intensidad en el canal n luego de la corrección de la corriente de oscuridad, $t_{int,measure}$ es el tiempo de integración del espectro medido y $t_{int,dc}$ el tiempo de integración usado para la corriente de oscuridad. $I[n]$ denota la intensidad en el canal antes de la corrección de corriente de oscuridad. $D_{offcorr}[n]$ es la intensidad de la corriente de offset corregida con la de oscuridad en el canal n . Un espectro típico de la corriente de oscuridad, con la temperatura en el espectrómetro de 29 °C, se muestra en la siguiente figura:

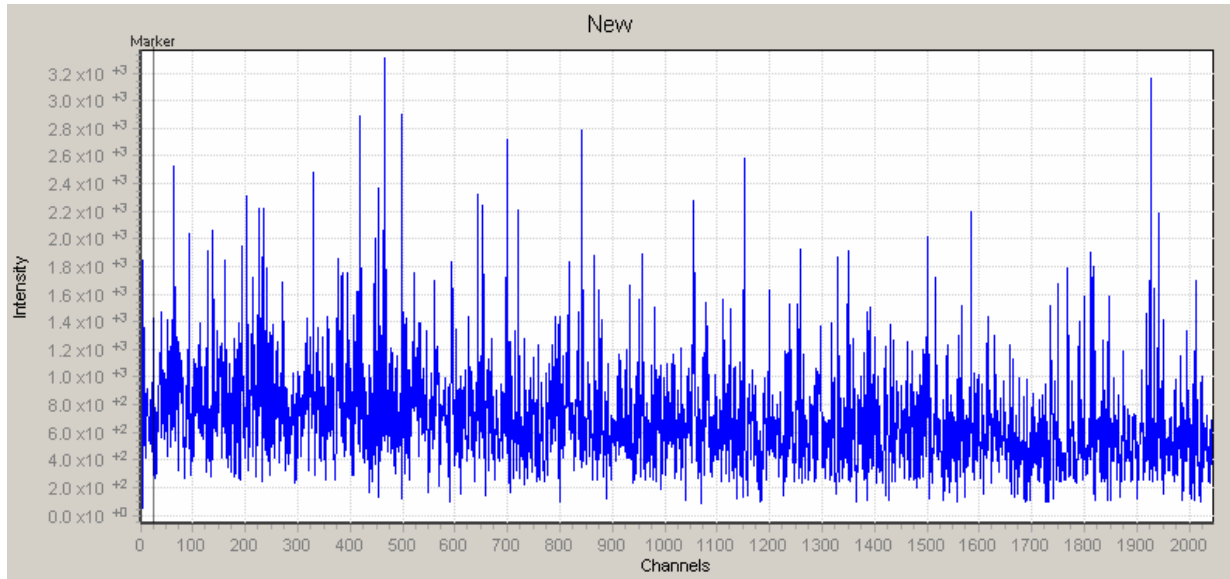


Fig. 1-3 Corriente de oscuridad en ITDoas a 29 °C

1.5.2. Corrección del Offset Electrónico

Para el proceso de análisis en la PC, un conversor A/D convierte la señal de luz analógica en una señal digital. El ruido en el detector puede producir señales de luz negativas que no pueden ser interpretadas por el conversor; por esto se agrega un offset electrónico a cada medida para evitar este problema. Antes del proceso de análisis, el offset electrónico proporcional al número de barridos se tiene que restar al espectro medido:

$$I_{offcorr}[n] = I[n] - \frac{numscans_{measure}}{numscans_{offset}} \cdot O[n]$$

$I_{offcorr}[n]$ es la intensidad en el canal n luego de la corrección de offset, $numscans_{measure}$ es el número de espectros medidos y $numscan_{offset}$ el número de barridos del espectro de offset. $I[n]$ denota la intensidad en el canal n antes de la corrección de offset y $O[n]$ la intensidad de la señal de offset en el canal n . Para determinar el offset electrónico se deben adquirir una gran cantidad de espectros (típicamente 10000) en un tiempo de integración mínimo (3ms). La señal de offset es por lo general dependiente de la temperatura. En la figura 1-4 se muestra un espectro de offset tomado a una temperatura de 29 °C.

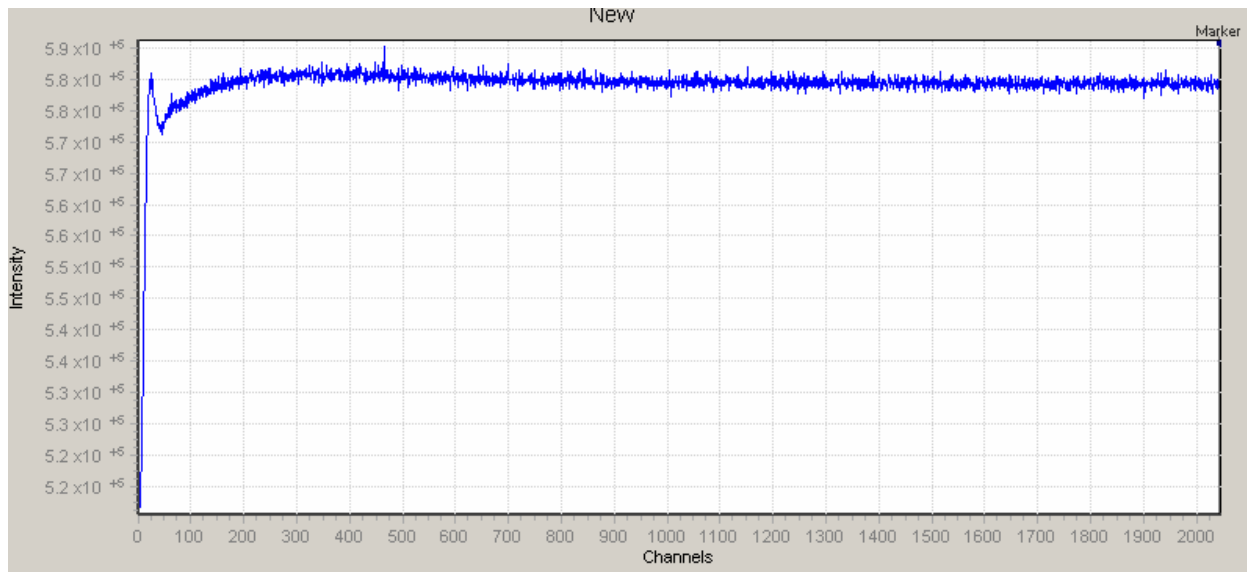


Fig. 1-4 Offset en ITDoas a 29 °C

1.5.3. Efecto de Ring

En 1962, Grainger y Ring descubrieron que la profundidad de las líneas de Fraunhofer en el espectro solar disminuye cuando la luz solar atraviesa la atmósfera terrestre. Las líneas observadas a grandes ángulos de azimut se observan “más completas” en comparación con las observadas para ángulos pequeños. Este efecto se conoce como el efecto de Ring. Típicamente los cambios en la densidad óptica debido al efecto de Ring, son de un porcentaje bajo. Una corrección precisa del efecto de Ring es importante, porque la absorción atmosférica en las evaluaciones de DOAS es a veces de un orden de magnitud más pequeño que los cambios debidos al efecto de Ring. Por ello se debe incluir un espectro de Ring, el cual puede ser calculado o medido, en el proceso de ajuste cuando se utiliza DOAS pasivo.

1.5.4. El Efecto Solar I_0

La intensidad de la radiación solar varía fuertemente con la longitud de onda debida a la alta densidad óptica y número de líneas de Fraunhofer. Estas estructuras de absorción angostas usualmente no se pueden resolver con un espectrómetro típico de DOAS con el rango de resolución usual de décimas de nanómetros hasta varios nanómetros. Como consecuencia, las estructuras de absorción cuando se mide directamente o por reflejo, la luz solar difieren de las medidas con una fuente de luz no estructurada. Entonces la convolución estándar del espectro de alta resolución no describe correctamente las medidas cuando se utiliza un instrumento de baja resolución. Este efecto, es conocido como efecto I_0 porque es causado por el espectro de las estructuras angostas del espectro solar. Para tomar en cuenta este efecto, se puede usar un espectro de referencia “ I_0 -corregido”, el cual se calcula de la siguiente manera:

Primero, el espectro de alta resolución de Fraunhofer I_0 se convoluciona con la función del instrumento H .

$$I_0^*(\lambda) = I_0(\lambda) * H$$

Luego, un espectro de absorción sintético de alta resolución, se calcula, usando una sección eficaz de absorción y una concentración c típica de la especie gaseosa a ser investigada. Esta absorción sintética se convoluciona con la función del instrumento también.

$$I^*(\lambda) = (I_0(\lambda) \cdot e^{-\sigma c}) * H$$

Finalmente, la sección eficaz σ^* puede ser calculada

$$\sigma^* = \frac{1}{c} \cdot \ln \left(\frac{I_0^*(\lambda)}{I^*(\lambda)} \right)$$

Para juntar el espectro atmosférico medido y el espectro de referencia I_0 -corregido, la concentración asumida para el cálculo y la concentración atmosférica deben ser la misma. Para fines prácticos sin embargo, una concentración atmosférica típica puede ser utilizada para todos los espectros.

1.6. Configuraciones Experimentales

El principio de DOAS puede ser aplicado a un amplio rango de variedad de caminos de luz y modos de observación. Se pueden clasificar las técnicas usadas de acuerdo a las fuentes de luz, distinguiendo entre “Active DOAS” y “Passive DOAS”, la primera utiliza fuentes de luz artificiales, mientras que la segunda utiliza fuentes de luz naturales, como el sol, la luna o las estrellas. Una descripción de los experimentos más comunes se observará en las siguientes secciones.

1.6.1. Configuraciones de DOAS Activo

Las Aplicaciones DOAS que utilizan fuentes de luz artificial, se conocen como aplicaciones “active DOAS”, DOAS activo. Usualmente se utilizan fuentes de luz con bandas espectrales anchas (por ejemplo lámparas de xenón). Una configuración óptica envía y recibe la luz emitida a través de un camino bien definido en la atmósfera antes de ser detectada por un espectrómetro. Como la concentración de trazas de gas en la atmósfera terrestre es usualmente pequeña, se necesitan caminos muy largos, de 10 a 20km. Estos caminos largos de luz representan un gran reto para la realización práctica de “active DOAS”. La mayoría del tiempo, los caminos de luz se toman paralelos al suelo y se optimizan para la investigación de gases traza de la troposfera. En resumen DOAS activo es un método elegante de obtener información de gases traza, en la atmósfera, con alta precisión y con caminos largos. La figura siguiente muestra cuatro configuraciones diferentes de DOAS activo. (1) Esquema de la configuración “Long-Path” DOAS, (2) Folded-Path DOAS haciendo reflejar la luz mediante una configuración óptica apropiada se mejora la longitud de camino y por tanto la sensibilidad, (3) Tomographic-DOAS utiliza múltiples haces de luz para determinar la distribución de gases traza en 2 dimensiones. Si los retro reflectores se disponen en

forma vertical, también se puede obtener información sobre la altura por medio de medidas de Long-Path-DOAS.

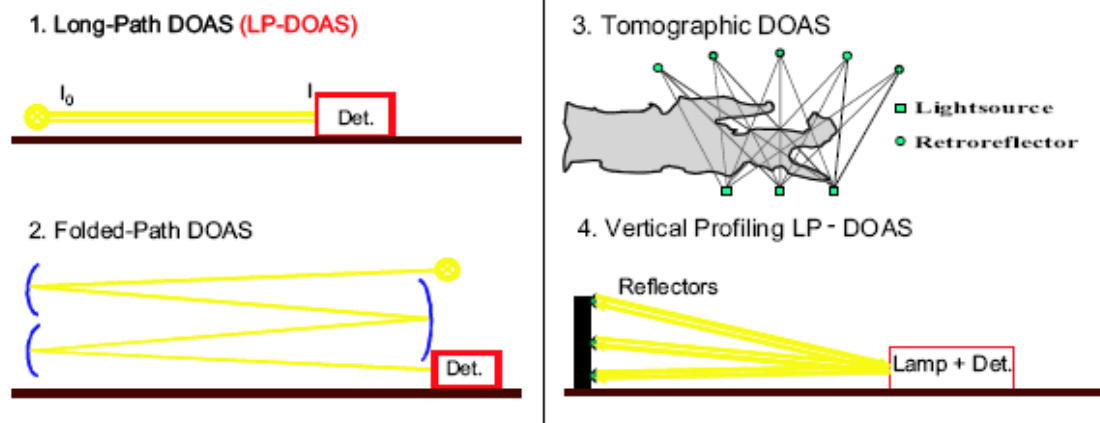


Fig. 1-5 Esquema de varias configuraciones de “Active” DOAS
 Figura extraída de la referencia bibliográfica [1]

1.6.2. Configuración de DOAS Pasivo

Las aplicaciones DOAS que utilizan fuentes de luz natural, se conocen como aplicaciones “passive DOAS” (DOAS pasivo). Generalmente se utilizan como fuentes de luz natural el sol o la luna, pero también se pueden utilizar las estrellas. En caso de utilizarse la luna o las estrellas, sólo se puede utilizar luz directa ya que es muy pequeña la cantidad de luz llega a la tierra desde esas fuentes. Sin embargo en el caso de la luz solar, la luz reflejada por las moléculas y partículas también se puede aprovechar. Por lo que el DOAS pasivo se puede dividir en mediciones directas o indirectas. Al igual que en las medidas de DOAS activo, la ley de Lambert-Beer se puede aplicar directamente en DOAS pasivo con mediciones directas. La figura siguiente muestra algunos ejemplos de DOAS pasivo con medición directa. Medidas directas del sol (5), luna o estrellas desde el suelo. La determinación de perfiles verticales de los gases traza es posible en experimentos desde un globo aerostático (6) cuando la luz es detectada a diferentes alturas durante el ascenso. (7) SCIAMACHY, un instrumento espacial basado en DOAS.

La medición indirecta ofrece una variedad aun mayor de aplicaciones. Uno de los primeros experimentos usando luz indirecta, fue la medida de la luz indirecta de la dirección del cenit (8). Este tipo de medidas se utilizan para medir la estratósfera y juegan un papel importante en el aprendizaje de los químicos estratosféricos. Otra aplicación de la medida indirecta del cenit es la investigación de transporte radiactivo en nubes (14). La técnica de medida de multi-axis DOAS (9) (MAX-DOAS). Max DOAS utiliza múltiples elevaciones de telescopio para relevar diferentes perfiles de absorción en la tropósfera baja. Con esto se logra una gran sensibilidad para los gases cercanos a la tierra, porque la luz detectada con una elevación pequeña del telescopio ha atravesado un camino significativamente mayor a través de la atmósfera baja comparada con la luz adquirida con altas elevaciones del telescopio. AMAX-DOAS es una variante aérea de MAX-DOAS, la que permite medidas por encima y por debajo de la línea de vuelo (10). Una técnica alternativa de configuración de DOAS pasivo que aún está en desarrollo es Imaging-DOAS (11). La medición simultánea de varias elevaciones permite una visualización en 2 dimensiones de las áreas de polución.

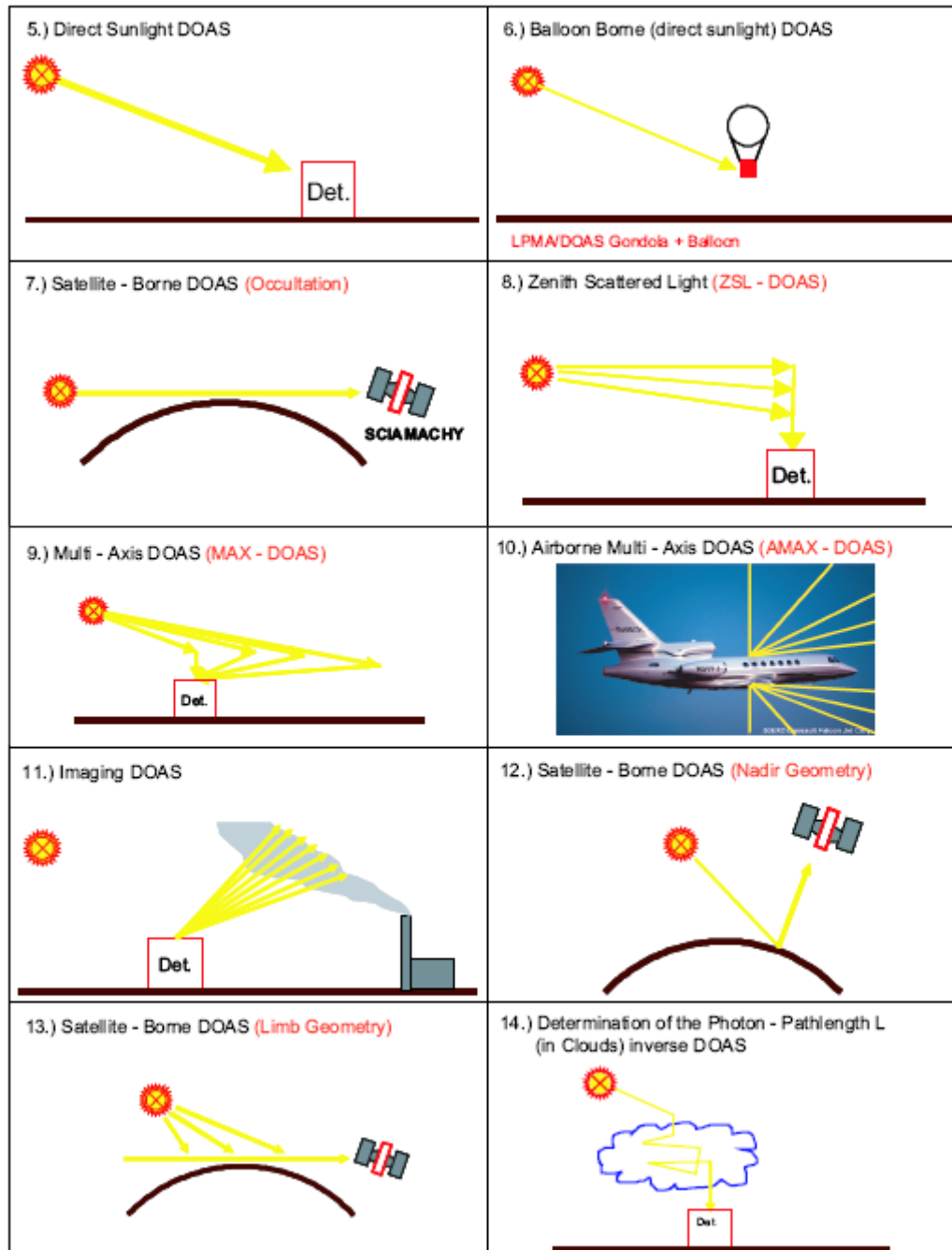


Fig 1-6 varias configuraciones de “passive” DOAS
Figura extraída de la referencia bibliográfica [1]

Dentro de lo que es DOAS pasivo, nuestro dispositivo esta optimizado para utilizar el método de ToTaL DOAS que describimos a continuación.

1.6.2.1. ToTaL DOAS

El método consiste en apuntar a diferentes blancos (“targets”) con un telescopio como se muestra en la figura. Los blancos localizados a diferentes distancias pero en direcciones similares, definen el largo geométrico para obtener la concentración de trazas de gas entre ellos. Al obtener la información, el análisis del espectro puede variar dependiendo del numero y tipo de blancos seleccionado (brillante, medio u oscuro).

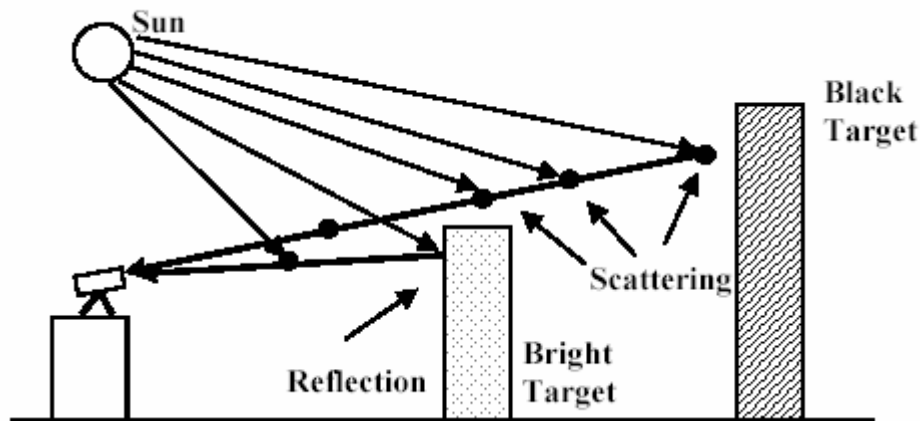


Fig.1-7- ToTaL Doas
(Tomado de Differential Optical Absorption Spectroscopy (DOAS) using
Targets: SO₂ and NO₂ Measurements in Montevideo City)

Del espectro adquirido por el instrumento se pueden obtener varias trazas adecuadas aplicando el método de DOAS. La luz solar $I(\lambda, L)$ recibida por el telescopio contiene dos términos,

$$I(\lambda, L) = I_{\text{atm}}(\lambda, L) + I_{\text{tar}}(\lambda, L)$$

donde $I_{\text{tar}}(\lambda, L)$ es la radiación medida causada por la luz solar reflejada contra el blanco, y $I_{\text{atm}}(\lambda, L)$ la radiación a la longitud de onda λ causada por la luz solar difundida en la línea de vista del telescopio entre los blancos y el instrumento (ubicado a una distancia L de los blancos). La luz difundida o reflejada en la línea de vista del instrumento es afectada por la presencia de trazas de gas de la estratosfera, y atenuada por difusión y absorción en moléculas y aerosoles presentes en la línea de vista. Como se realiza una medición diferencial de la presencia de trazas de gas en la estratosfera no afecta la medida (como una alternativa se puede corregir la misma con una medida del cenit).

2. Objetivos

2.1. Objetivo General

Automatizar y controlar en forma remota un dispositivo que implementa Total-DOAS.

2.2. Objetivos específicos

- Automatizar y mejorar el sistema Total-DOAS utilizado en el grupo de Óptica Aplicada del Instituto de Física de la Facultad de Ingeniería
- Construir un sistema mecánico capaz de lograr que, un paso en el motor, se corresponda con un ángulo de rotación en el sistema óptico adecuado a la precisión deseada, con un máximo de un metro, a una distancia de tres kilómetros.
- Diseñar un sistema de control capaz de tomar en su entrada una posición y a partir de ella enviar las señales necesarias a los motores para que el sistema óptico se posicione adecuadamente.
- Proveer una interfaz con los dispositivos externos.
- Proveer una interfaz amigable para los usuarios pasivos y otra más segura para el administrador.
- Desplegar la información del espectro medido de dos formas, un archivo con los datos y una gráfica de estos.
- Fácil de actualizar para diferente hardware (espectrómetro, motores, servidor, etc.).

3. Nuestra Solución

En el dispositivo original, que se muestra en la figura 3-1, se contaba con un telescopio Tasco y un espectrómetro Ocean Optics conectados por una fibra óptica.

Nuestra solución consintió en, reutilizando todos los componentes ya existentes pertenecientes al dispositivo original, construir un dispositivo automático de medida, que se pudiera manejar en forma remota, que fuera mas robusto y cumpliera con todos los objetivos.

Inicialmente, se construyó un contenedor que albergara el telescopio, el espectrómetro y la conexión de fibra óptica en forma segura.

Luego de tener el contenedor, que de ahora en más llamaremos “cabecera”, tuvimos el reto de construir un sistema que fuera capaz de moverla, con la precisión necesaria.

La cabecera va montada sobre un trípode, y se mueve por la acción de dos juegos de engranajes. Esto es ejecutado por dos motores de pasos y sensado por dos “encoders”.

Los motores de pasos son manejados por un microcontrolador, el cual decide el movimiento de los motores y es monitoreado por los “encoders”.



Fig 3-1 Dispositivo MAX-Doas



Fig 3-2 Dispositivo ITDoas terminado

El microcontrolador es también el encargado de sensar la temperatura ambiente, controlar un ventilador y encender los indicadores lumínicos en la parte posterior de la cabecera.

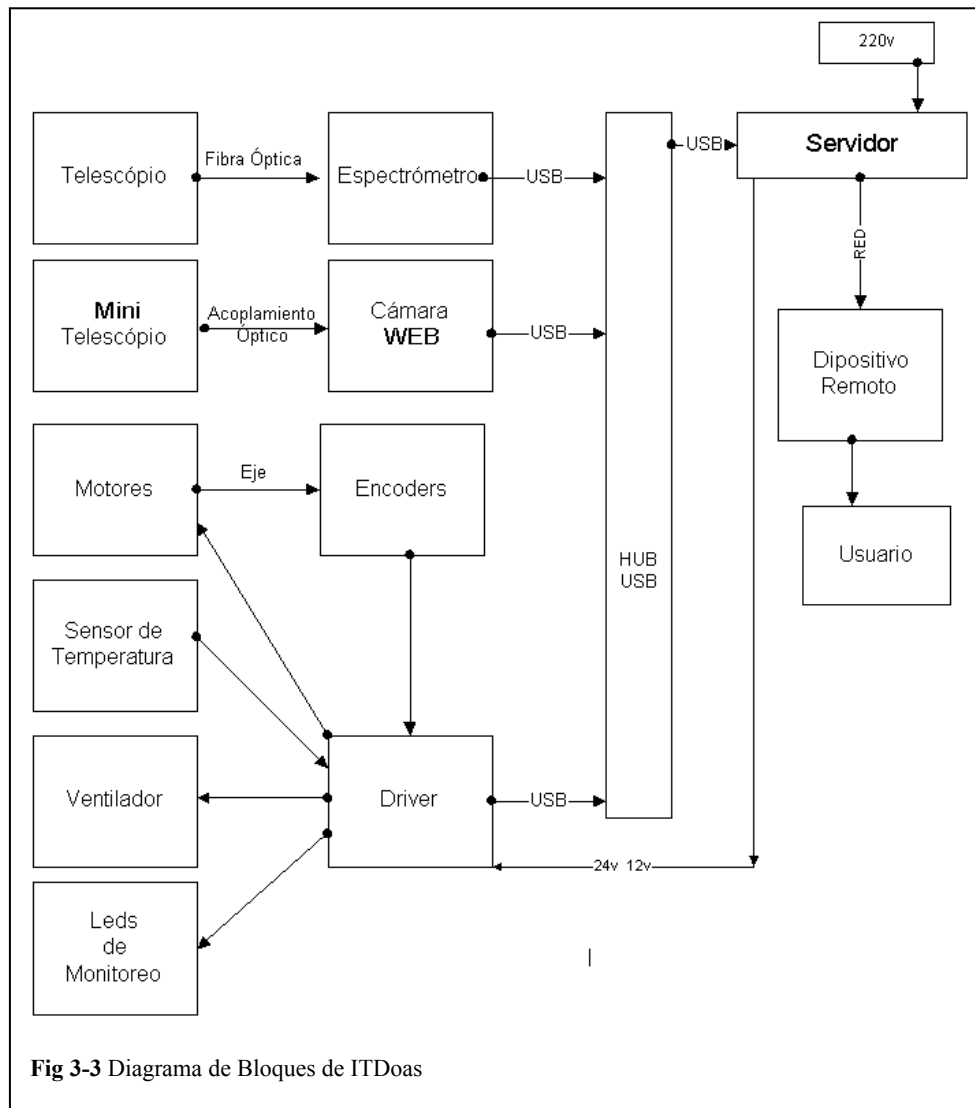
Para que el sistema pudiera ser manejado a distancia, se le incorporó una cámara web, la cual, por un sistema óptico, independiente al correspondiente a la toma de espectros, muestra la posición exacta a la cual está apuntando el telescopio.

El instrumento fue conectado a un servidor web, el cual se comunica con el resto del mundo mediante una interfaz de red .

Todo el dispositivo que incluye la cabecera, los motores, los engranajes, el microcontrolador, el servidor y el tripode fue lo diseñado por

nosotros y llamado ITDoas. (“Intelligent tomographic Target light scattering-Differential Optical Absorption Spectroscopy”)

En la figura 3-2 podemos ver una foto del ITDoas Terminado y en la figura 3-3 un diagrama de bloques.



3.1. Actores

Nuestro sistema se maneja en forma remota, lo que le da la libertad al investigador de controlar sus mediciones desde un sitio diferente al que está instalado; por tal motivo no siempre la persona que lo instala es la que dirige las mediciones. Por esto reconocemos tres actores en nuestro sistema:

- 1) El primero el encargado de instalar el sistema en el punto de medida. A este lo llamaremos “instalador”

- 2) El “administrador de sistema” que es el encargado de decidir que mediciones hacer, configurar rutina de trabajo y de recibir todos los datos del experimento.
- 3) El tercer actor “usuario no administrador” es uno o más de los investigadores que tienen la posibilidad de monitorear el experimento y ver los resultados sin tener permisos de modificar la rutina de trabajo. Son simplemente observadores.

3.2. Casos de Uso

3.2.1. Proceso de Instalación

El instalador ubica un punto donde efectuar sus medidas. Instala el trípode y lo nivela. Monta el telescopio y lo dirige hacia una posición cercana a los objetivos. Coloca y conecta el servidor a una fuente de energía de 230V. Conecta al servidor la cabecera. Conecta el servidor a la red y lo enciende. Espera la notificación de que todo está encendido.

Luego de esto el Administrador puede empezar a trabajar desde su oficina o en el lugar de medida con una PDA o laptop.

3.2.2. Proceso de Administración

Luego de recibir la notificación del sistema, el Administrador se conecta mediante un navegador web con el sistema. Se autentica. Mediante la interfaz web el Administrador ubica los distintos targets. Para cada target tendrá que ingresar una etiqueta. Luego crea una rutina de trabajo y se la envía automáticamente al servidor. A medida que se va corriendo la rutina, se va mostrando en la página web los resultados que se van obteniendo. El usuario Administrador podrá interrumpir en cualquier momento la ejecución de la rutina y reprogramarla. Al final de la rutina el Administrador podrá bajar los datos obtenidos en un formato estándar, pronto para el posterior análisis en software especializado.

3.2.3. Proceso de monitoreo

El usuario No Administrador se conectará a una interfaz web sin autenticación en la que encuentra la rutina que se está llevando a cabo, puede ver la webcam montada en el telescopio y bajar todos los resultados.

3.3. Forma de Trabajo

Al tener que construir un sistema complejo, el cual incluía partes de distintas disciplinas, optamos por dividir el trabajo en tres otorgándole a cada uno de los participantes la responsabilidad de llevarlo a feliz término.

La parte mecánica fue responsabilidad de Juan A Gentilini, la parte de Hardware de Santiago Conde y el Software de Sergio Blanco.

Durante todo el desarrollo se fueron generando distintos dispositivos de prueba, los cuales se sabía que no iban a ser parte de la solución final, para investigar y poder perfeccionar el diseño a medida que transcurría el tiempo. Por ejemplo para saber si los motores serían capaces de mover el dispositivo se creó:

- Un software que fuera capaz de mover los motores de pasos desde el puerto paralelo de una PC.
- Una tarjeta que hiciera de buffer del puerto paralelo.
- Una infraestructura mecánica que simulaba lo que supuestamente podría llegar a ser el sistema final.

Luego de muchas iteraciones se terminó teniendo el siguiente WBS el cual representa cada bloque y sus componentes. Ver Figura 3-4

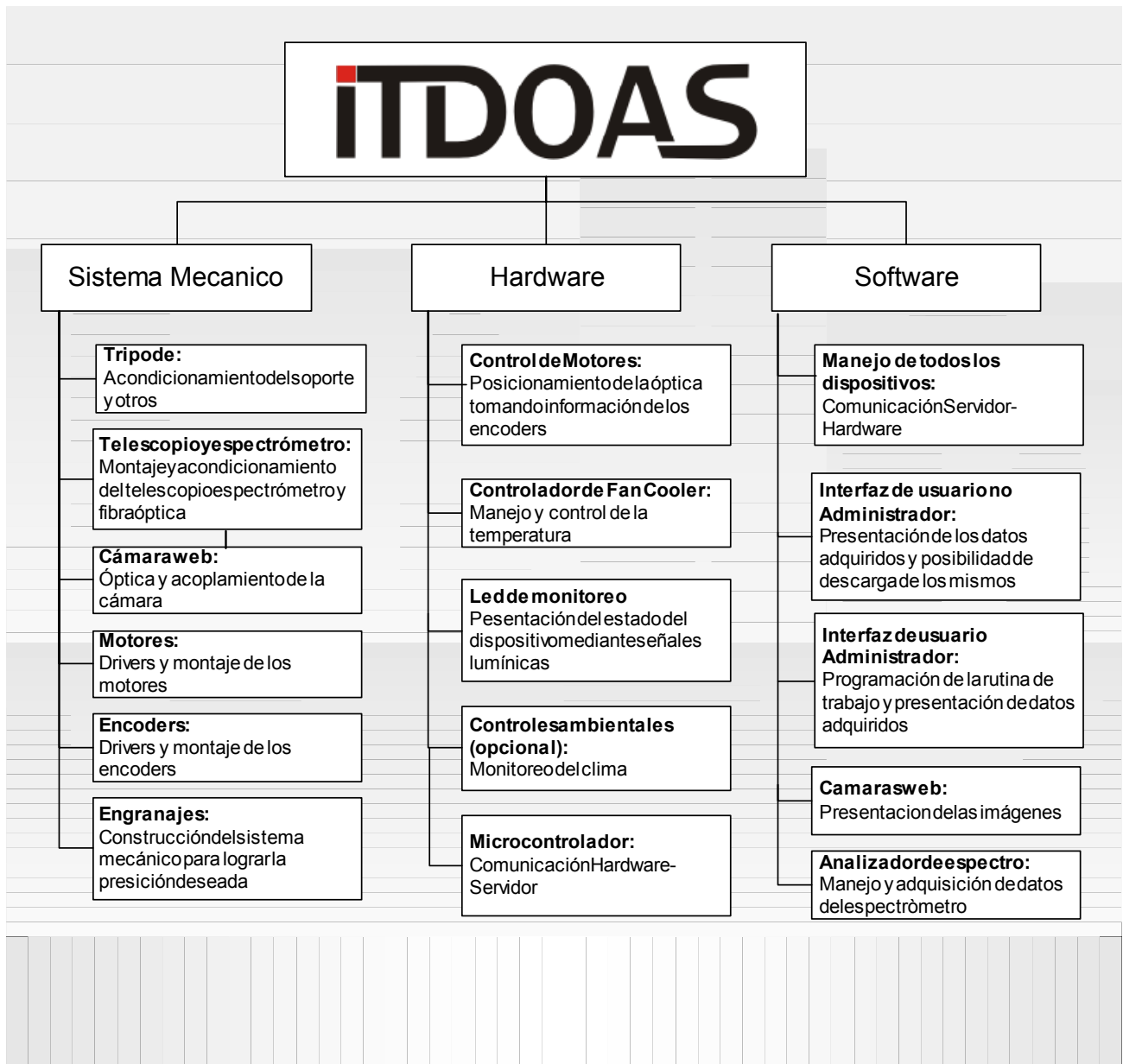


Fig. 3-4 WBS

A continuación se describirán los tres bloques y la forma que tomó el diseño final. En el apéndice se encontrarán algunas reseñas de la construcción de algunos de los dispositivos que no se utilizaron.

4. Sistema Mecánico

4.1. Requerimientos

El dispositivo debería ser capaz de direccionar el telescopio con una precisión de un metro a una distancia de tres kilómetros.

Debería ser portátil y soportar diferentes cambios climáticos por su permanencia durante varias horas a la intemperie.

4.2. Diseño

Uno de los retos más importantes del proyecto era el diseño y la construcción del sistema mecánico. Si bien hemos contado con la invaluable ayuda del Sr. Antonio Saez, encargado del laboratorio de mecánica fina del instituto de física, la fase de diseño y la construcción llevo aproximadamente diez meses. Es decir, la implementación mecánica del proyecto, llevó más tiempo que ninguna otra actividad.

4.3. Implementación del sistema

El Sistema esta compuesto por

- Una cabecera
- Sistema de engranaje y motores
- Trípode
- Servidor

4.3.1. Cabecera

Es la parte que contiene todo el sistema óptico y la lógica de control.

La caja metálica contiene los siguientes componentes. Como parte principal tenemos un telescopio Tasco Galaxee tipo Newton conectado a un espectrómetro Ocean Optics s2000 mediante fibra óptica modelo Ft -600-umt. La fibra está acoplada al ocular del telescopio con un adaptador construido especialmente con ese fin y



Fig. 4-1 Cabecera ITDoas

conectada directamente al espectrómetro.

El espectrómetro a su vez está conectado a un modulo ADC1000 el cual convierte las señales RS232 del instrumento en protocolo USB.

Para informar al operario de dónde provienen los espectros que está recabando, se adosó al sistema una cámara web Philips de 640*480 píxels acoplada a una mirilla de telescopio. Todo el conjunto cámara y mirilla fue puesto en un soporte y unido a la caja. Este soporte permite ajustar la imagen de la cámara para que coincida con la señal del telescopio. La cámara a su vez tiene una salida USB.

Dentro de la cabecera se encuentra la tarjeta que maneja los motores, monitorea los encoders, mide la temperatura, enciende los leds de control y acciona un ventilador. La misma cuenta con una interfaz USB que conjuntamente con la de la cámara web y la del espectrómetro está conectada a un hub quien a su vez se conecta al servidor.

De la cabecera tenemos dos cables que se conectan al servidor, uno de ellos es un cable AB-USB estándar y el otro es un diseño propio para alimentar los motores y la tarjeta con 24 y 12 volts. De esta tarjeta salen otros dos cables, uno a cada uno de los juegos motor-encoder; por estos conductores se energizan las distintas bobinas de los motores, se alimentan los encoders y se recibe la información de ellos.

También cuenta con dos leds de dos colores cada uno que dan información al usuario sobre lo que el dispositivo está haciendo.

En la parte delantera de la cabecera nos encontramos con el telescopio y su mirilla; el telescopio esta conectado a través de una fibra al espectrómetro mientras que su mirilla esta acoplada a la cámara web.

La cámara cuenta con la posibilidad de orientarse en la misma dirección que el telescopio.

4.3.2. Interior de la Cabecera

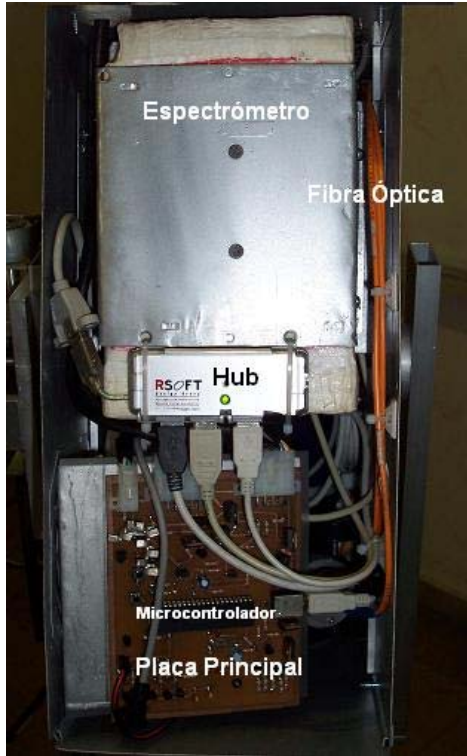


Fig. 4-2 Interior de la Cabecera

Al remover la tapa superior de la cabecera nos encontramos en la parte frontal con la tarjeta que maneja el hardware allí instalado, más hacia atrás el hub-usb y debajo de éste un recubrimiento de espuma donde se encuentra el espectrómetro y el ADC1000.

Si retiramos la tarjeta encontraremos un ventilador, el conjunto cámara web-mirilla y el telescopio principal.

La cabecera del sistema está montada sobre una U de aluminio que contiene los engranajes y el motor del movimiento vertical, mediante dos ejes, uno a cada lado de los laterales de la cabecera. En la base de la U de aluminio, se encuentra un eje vertical que soporta toda la cabecera y es el encargado de mover todo el conjunto en forma horizontal.

4.3.3. Diseño de los engranajes

Para la implementación de los movimientos se consiguieron dos motores de pasos. Estos fueron seleccionados de un lote perteneciente al IF; por esta razón ambos no poseen las mismas especificaciones.

Los modelos de los mismos son

Modelo	Angulo	Voltage	Resistencia	Corriente
42SPM-24D8A	7.5°	24V	24.4Ω	984mA
Sanyo CP023	7.5°	24V	50Ω	480mA

La característica principal de estos motores es que cada paso corresponde a 7.5°, es decir, necesitamos 48 pasos para cubrir los 360°.

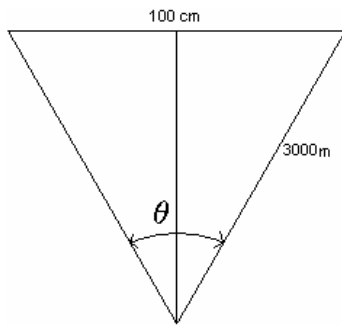


Fig.4-3

Una de las restricciones del dispositivo era que fuera capaz de moverse 1m en 3km, lo que significa que debe ser capaz de moverse 0.019 de grado por cada paso.

En las pruebas realizadas con los motores elegidos, se llegó a la conclusión de que los mismos trabajan hasta una velocidad de 300rpm sin sufrir deterioros.

$$3000m \sin\left(\frac{\theta}{2}\right) = 0.5m \Rightarrow \theta = 2A \sin\left(\frac{0.5}{3000}\right)$$

$$\theta = 0.019^\circ$$

Sabiendo que debemos movernos de al menos 0.019° , por paso, nuestro sistema demorará 1.3 minutos en girar 360° .

$$\frac{360^\circ}{0.019^\circ} = 18947$$

$$\frac{18947}{48 \text{ pasos}} = 395 \text{ vueltas}$$

$$\frac{395 \text{ vueltas}}{300 \text{ Rpm}} = 1.3'$$

Buscando una solución de compromiso entre el tiempo que demora en girar los 360° , la precisión que podemos lograr y las limitaciones mecánicas llegamos a la siguiente implementación:

El dispositivo está compuesto por un motor, acoplado a su eje un sinfín, que trabaja sobre una corona (1), que a su vez tiene acoplado en su eje otro sinfín que trabaja sobre una segunda corona (2), la cual va fija sobre el eje principal, como muestra la figura 4-4.

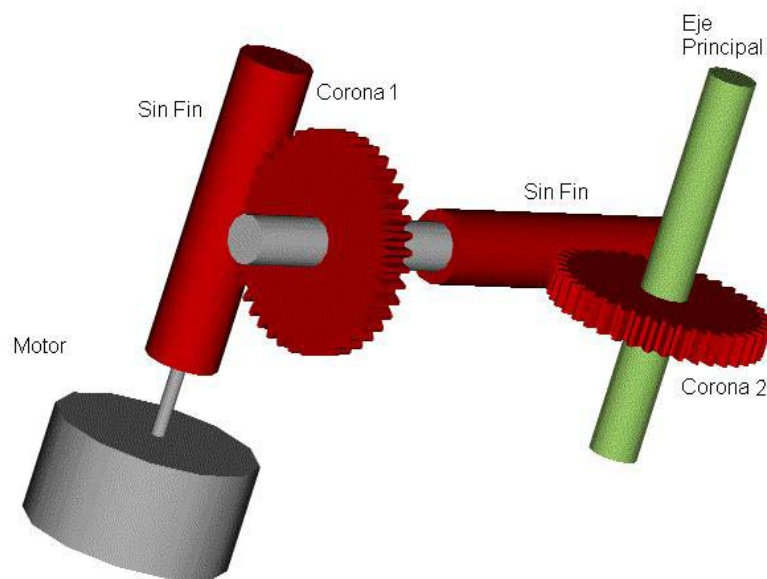


Fig 4-4 Juego de Engranajes

Cada juego tiene la función de hacer variar tentativamente la altura y el azimut con una precisión que cumple los requerimientos.

Para esto construimos las coronas de 40 dientes, lo que significa una relación de vueltas de 1 a 1600, que multiplicado por la cantidad de pasos significa que un paso de motor representa una 76800 parte de circunferencia.

Un paso es igual a $\frac{360^\circ}{76800} = 4.69 \cdot 10^{-3}^\circ$ que es alrededor de la cuarta parte del ángulo pedido en los requerimientos. La distancia d por pasos es igual a:

$$d = 2 * 3000 * \sin(4.69 \cdot 10^{-3}) = 0.2456m$$

Calculando ahora el tiempo estimado en girar 360°

$$\frac{76800 \text{ pasos} * \text{rev}}{48 \text{ pasos}} = 1600 \text{ rev}$$

$$\frac{1600 \text{ rev}}{300 \text{ RPM}} = 5.3'$$

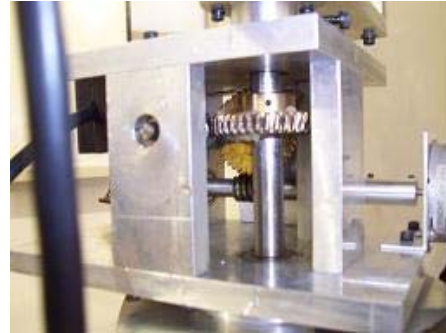


Fig.4-5 Foto del juego de Engranajes

Esta cota superior de tiempo nos parece más que razonable para posicionar la cabecera, ya que en la realidad los targets se encuentran a poca distancia angular entre sí¹ y a su vez tiene muy buena resolución.

Diseñada la solución mecánica, se pensó en la necesidad de tener información de los motores para comprobar los movimientos.

Para esto se adquirieron dos “encoders” ópticos ENA1JB28L00064-ND de 64 pasos cada uno, los cuales fueron acoplados a los ejes de los motores. Un encoder consiste en un dispositivo capaz de medir, en pasos, el giro de un eje, indicando además el sentido².

Por la forma de utilización de los “encoders”, llegamos a tener una resolución de $4 \times 64 = 256$ lo que produce $256 \times 1600 = 409600$ pasos por revolución de la cabecera, lo que significa que podemos medir alrededor de $\frac{360^\circ}{409600} = 0.87 \cdot 10^{-3}^\circ$, que es equivalente a una distancia $d = 2 * 3000 * \sin(0.87 \cdot 10^{-3}) = 0.0920m$, consumiendo una memoria de 19 bits.

¹ Generalmente no se trabajará con ángulos mayores a 180° lo que reduce a la mitad el tiempo.

² Por referencias mirar apéndice iii

4.3.4. Trípode

Dado el peso del dispositivo, fue necesario la adquisición de un trípode del tipo de agrimensura.

Esto conlleva unos inconvenientes.

- El trípode estaba diseñado para ser usado sobre suelo de tierra
- No tiene anclaje en sus patas, es decir estas se pueden abrir tanto como se quiera.

Para eliminar el primer problema se diseñó un tipo de regatón que al fijarlo sobre las puntas de las patas del trípode, permite el uso del mismo en superficies lisas sin dificultad.

Para mantener el trípode en una posición estable se construyó una estrella de aluminio, de tres puntas, con cabeceras de nylon, las cuales se adaptan a las patas del trípode no permitiéndoles a las mismas que se abran o cierren luego de ajustado.



Fig.4-6 Trípode

4.3.5. El servidor

El servidor es una PC DELL desktop sin teclado y monitor modelo OptiPlex GX110 con un procesador P3 de 700mhz y 256 Mb de memoria ram. La misma tiene además una tarjeta Wi-fi genérica.

A la PC se le removió la unidad de disquete y se le colocó en su lugar una fuente de 24V.

Para realizar la conexión con la cabecera se instaló en la parte posterior un conector tipo Canon de cuatro contactos en los cuales se obtienen 24V de la fuente mencionada anteriormente, 12V de la fuente del PC y un borne de tierra.

5. Hardware

Luego de desarrollados e instalados todos los dispositivos del hardware necesarios para alcanzar los requerimientos del proyecto, necesitamos diseñar un elemento capaz de recibir las órdenes que llegan desde el servidor, procesarlas y actuar sobre los anteriores. El mismo debe incluir un módulo que establezca la lógica de control y otro módulo que proporcione las tensiones y corrientes que especifican los periféricos. Este mecanismo es lo que denominamos Placa de Control.

5.1. Diseño de la placa de control

Al diseñar la placa de control optamos por dividirla en dos bloques, uno en el que interviene el microcontrolador (quién establece la lógica de control) y otro compuesto por una serie de buffers que proporcionan la salida de potencia. En la figura 5-1 se muestra un diagrama de bloques con el conexionado entre estas dos partes y los periféricos que tiene que manejar la placa: motores de paso, encoders, ventilador e indicadores lumínicos.

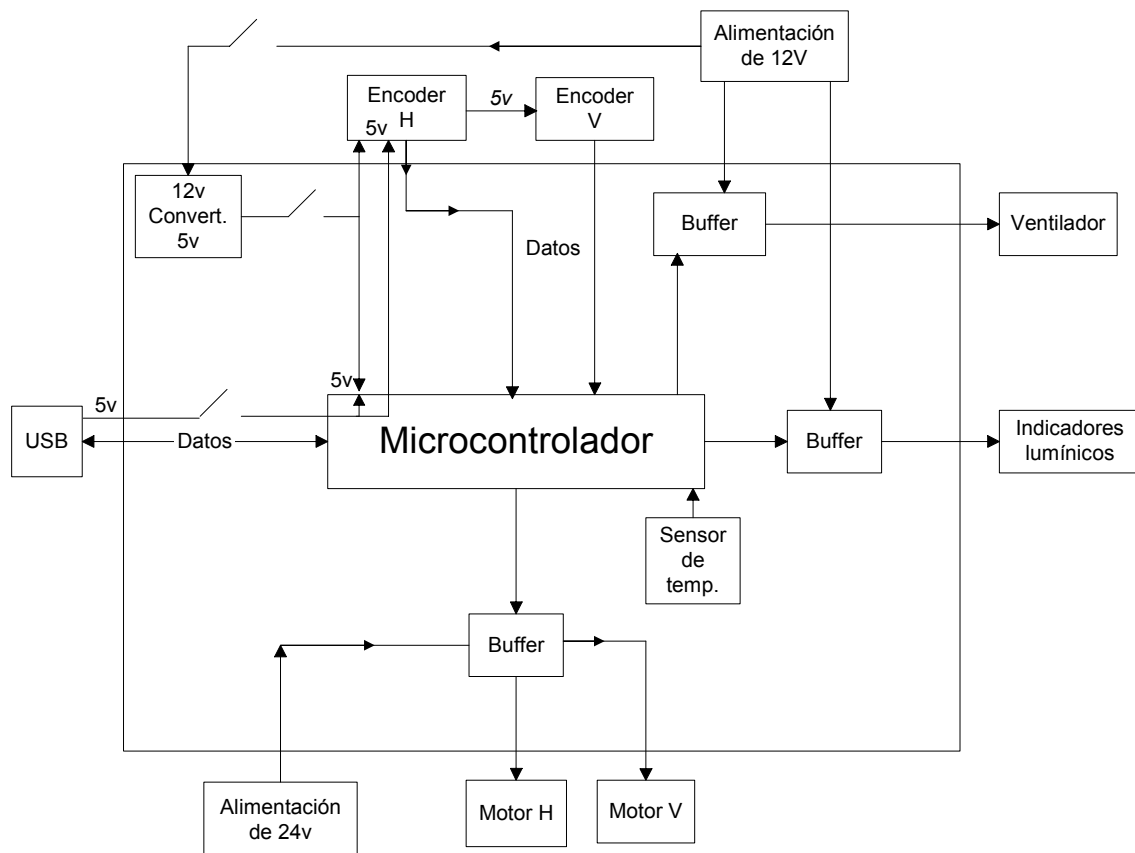


Fig.5-1 Diagrama De Bloques Placa de Control

5.1.1. Entradas y Salidas de la Placa

En el diagrama anterior se puede observar que a la placa entran las tensiones de alimentación y las señales de datos, tanto del cable USB como de los encoders.

En cuanto a las tensiones de la placa, se observa que ésta tiene dos tensiones distintas, una para los motores, la de 24V, y otra, la de 12V, para manejar los indicadores y el ventilador. La entrada de 12V a su vez es convertida a 5V para manejar los “encoders” y el microcontrolador. Estos últimos pueden ser alimentados en forma opcional por la tensión de 5v del cable USB. La opción de decidir de dónde tomará dicha tensión tiene que ver con la posibilidad de usar otro dispositivo como servidor, al cual no se le pueda tomar tanta corriente del puerto USB.

Luego tenemos la señal de entrada proveniente del USB entrando directamente al microcontrolador e interviniendo en el protocolo de comunicación entre los programas del servidor y los del microcontrolador.

Por otro lado las señales de entradas desde los encoders, quienes se encargan de monitorear el movimiento, intervienen en la estrategia de control “cerrando el lazo”, es decir llevando la cuenta de los movimiento de los ejes de los motor.

Por último la placa cuenta además con un sensor de temperatura, el integrado TMP124AID, cuyas señales las podríamos ubicar como externas a la placa dadas sus características de comunicación con el bloque del microcontrolador.

Con respecto a las salidas de la placa contamos con las señales para manejar el ventilador, los indicadores lumínicos y el movimiento de los motores de paso; todas estas provenientes del microcontrolador.

5.1.2. Microcontrolador

Para la construcción del bloque de control utilizamos el microcontrolador PIC 18F4455, un microcontrolador de Microchip de 40 pines con conexión USB 2.0, provisto de 5 puertos programables en su mayoría de 8 bits y cuyas demás características se pueden observar en forma detallada en su hoja de datos mencionada en la bibliografía.

Para iniciar, nos basamos en una aplicación de Microchip, “PICDEM™ FS USB”, un sistema de prueba de dicho microcontrolador, el cual fue una guía para la implementación de nuestro controlador. A partir del anterior sintetizamos el siguiente esquemático formado por el microcontrolador y las partes asociadas para su funcionamiento.

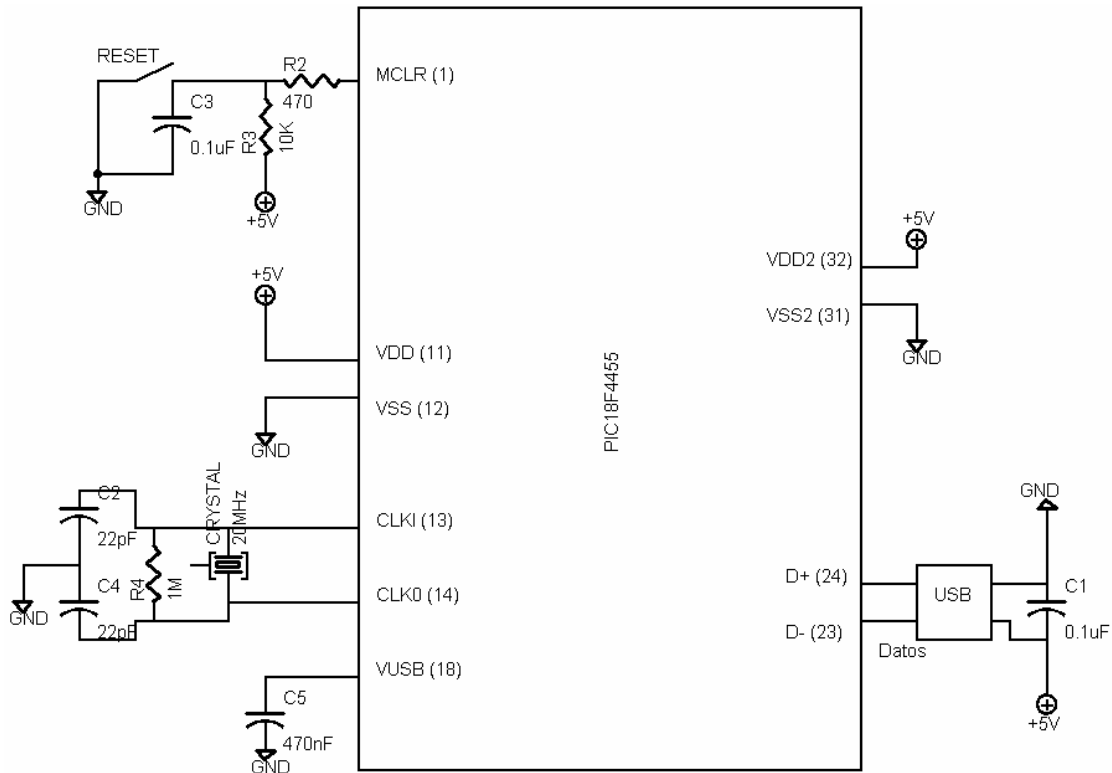


Fig.5-2 PICDEM™ FS USB
Figura extraída de Microchip ref [20]

Para la programación del microcontrolador, utilizamos la utilidad “Bootload Mode” de la aplicación “PicDem” la cual provee un medio simple para re-programar el microcontrolador sin la necesidad de utilizar un dispositivo de programación externo, pudiéndolo programar sobre el mismo circuito. Para ello fue necesario cargar previamente en el microcontrolador el firmware: “Microchip Bootload”, y utilizar el programa “Demo Tool” para cargar los códigos específicos de nuestra aplicación.

5.1.2.1. Conexionado del Microcontrolador

Como mencionábamos al comienzo el microcontrolador deberá interactuar con los siguientes periféricos:

- Servidor.
- Motores de paso horizontal y vertical.
- Encoders, uno para el movimiento horizontal y otro para el vertical.
- Ventilador.
- Indicadores lumínicos.
- Sensor de temperatura.

El PIC 18F4455 tiene un puerto USB en sus pines 23 y 24 los cuales fueron conectados a una cabecera USB hembra tipo B la cual es la conexión de datos con el servidor.

Para la conexión con los motores de paso, como su funcionamiento necesita de la activación de cuatro bobinas para cada uno, (ocho en total) se reservó el puerto “D” de ocho bits.

Para los encoders, cada uno de ellos tiene una salida compuesta por dos bits de datos y por ello se designaron cuatro bits del puerto B del PIC.

Para el sensor de temperatura son necesarios tres bits de datos y por ello se utilizaron otros tres bits del puerto B

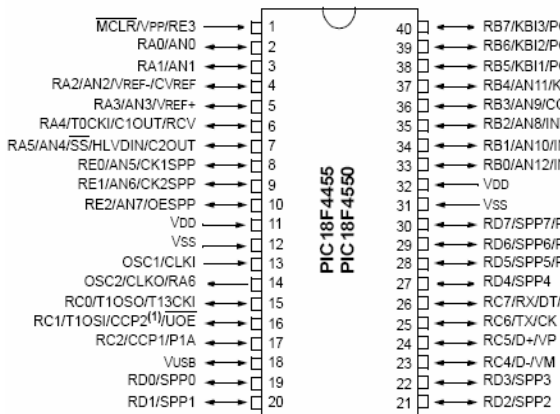
Para el ventilador, por necesitar simplemente de una señal de comando para su encendido se decidió por el bit cero del puerto A.

Para los indicadores lumínicos (5 leds en total), se optó por comandarlos con otros 5 bits del puerto A.

Además del mapeo de puertos que se menciona anteriormente se encuentra el botón de RESET conectado al pin número uno (MCLR), el botón de BOOTLOAD conectado al bit cuatro del puerto B, la señal de reloj y finalmente las tensiones de alimentación.

A continuación se presentan, en la figura 5-3, los esquemáticos tanto del PIC 18F4455 como de su conexionado.

Esquemático del PIC



Esquemático del conexionado del PIC

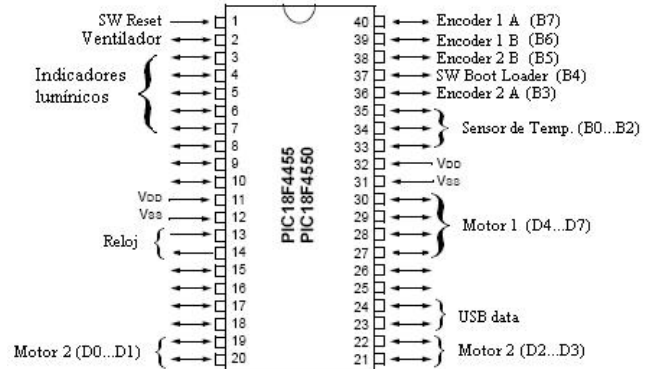


Fig 5-3

5.1.3. Buffer

Un buffer es un dispositivo que evita el efecto de carga en un circuito. En su forma más sencilla es un amplificador operacional funcionando como seguidor. Por consiguiente el voltaje no disminuye en el circuito, ya que éste toma el voltaje de la fuente de alimentación del operacional y no de la señal que se está introduciendo.

En nuestra placa estos buffers constituyen la salida de potencia, dado que las salidas del microcontrolador no pueden entregar la tensión ni la corriente que los periféricos necesitan.

5.1.3.1. Motores de Paso

Los motores de paso, para su funcionamiento requieren de un voltaje de 24v y por sus resistencias de bobinado (24 y 50ohms) esto significa sobrellevar corrientes máximas de 1A.

Con este fin nos concentramos en la activación de una sola bobina. Identificamos dentro del problema a nuestra entrada como una lógica de activación (puerto D del microcontrolador), la cual permitirá, dependiendo de su valor lógico, entregar o no hacia la salida una corriente de 1A proveniente de una fuente externa.

Ya que por el alto consumo, el circuito eléctrico no debe tomar corriente de la lógica de activación, ésta ultima solo decidirá dirigirla o no hacia el motor.

La solución es por lo tanto una etapa de salida compuesta por un transistor de señal seguido de un transistor de potencia funcionando el conjunto como un interruptor (corte y saturación). Esto permitirá tener una alta impedancia a la entrada, evitando así cargar al microcontrolador. Para el transistor de señal elegimos el BC548 y para el de potencia el TIP41C.

El resultado de esto es el circuito que se presenta en la figura 5-4.

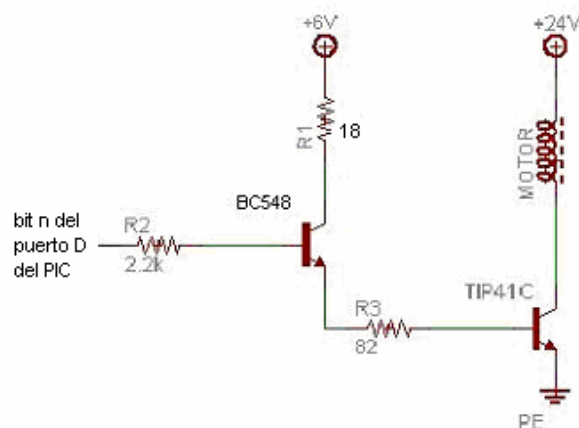


Fig.5-4 Buffer Motor

En el circuito de la figura si se coloca a “1” el bit correspondiente del puerto D (aproximadamente 4.7V) el transistor BC548 trabaja en zona activa con

$$i_c = 100 \left(\frac{4.7v - 0.7v - 0.7v}{2.2k\Omega + 110(82\Omega)} \right) \approx 0.029A, \text{ con lo cual el TIP41C tendrá una } i_c \approx 0.97A,$$

que es aproximadamente la corriente de saturación por ser $I_{Csat} = \frac{24v - 0.3v}{24\Omega} = 0.98A$.

Entregando de este modo la corriente de demanda de 1A.

Ahora, si por el contrario se coloca a 0 la entrada (aproximadamente 0v) no habrá corriente por la base de la primera etapa ni de la segunda quedando ambos transistores en zona de corte.

Dado que cada motor consta de cuatro bobinas, lo que denominamos el buffer del motor estará compuesto por 4 circuitos como el anterior.

Para el otro motor, en el que la resistencia por bobinado es de 50ohms, la corriente de saturación será $I_{Csat} = \frac{24v - 0.3v}{50\Omega} = 0.47A$. Observamos que la corriente de saturación es menor a la calculada para el circuito anterior por lo tanto con un circuito idéntico se logrará el mismo efecto.

5.1.3.2. Ventilador e indicadores lumínicos

Al igual que en los motores de paso, el encendido tanto del ventilador como de los indicadores lumínicos, representaría una carga demasiado grande para las salidas del microcontrolador. Por lo tanto también fue necesario bufferear estas salidas en una forma similar, pero sin utilizar transistores de potencia ya que el consumo de los anteriores es significativamente menor al de los motores de paso.

Para el buffer del ventilador se utilizó el circuito de la figura 5-5.

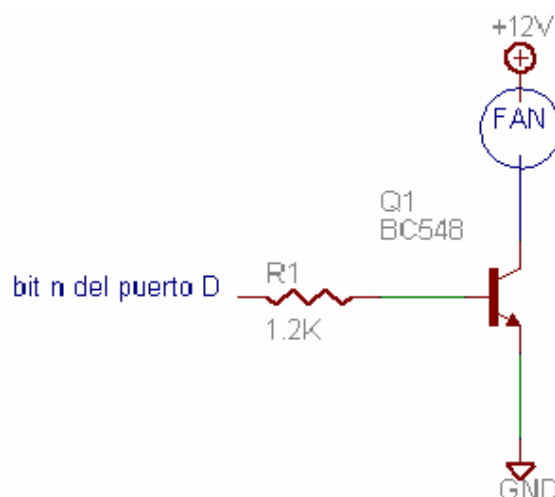


Fig.5-5 Buffer Ventilador

En el circuito de la figura si se coloca a “1” el bit correspondiente del puerto D (aproximadamente 4.7v) el transistor BC548 trabaja en zona activa con

$i_b = \frac{4.7 - 0.7}{1.2k} = 3.33mA$. Dadas las especificaciones del motor del ventilador, teniendo un valor de corriente 0.14A, basta con un β de 50 para lograr la saturación del transistor, el cual es menor al parámetro β de estos transistores. Por lo tanto el transistor está en saturación. Por otro lado si se pone a 0 dicho bit el transistor entra en zona de corte.

De esta forma nuevamente evitamos cargar la salida del microcontrolador.

Finalmente en el caso del buffer de los indicadores lumínicos se utilizó un circuito como el siguiente para cada led.

Del mismo modo que en los buffers anteriores, R1 es calculada para lograr la

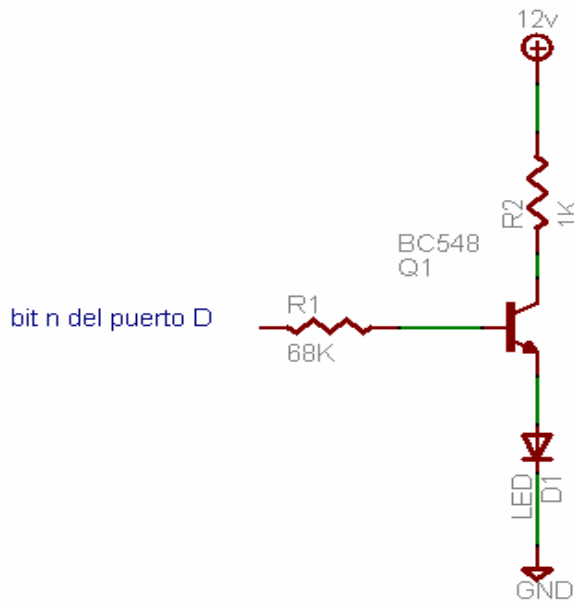


Fig.5-6 Buffer Leds

saturación del transistor. El papel que juega R2 es evitar sobrecorrientes en el led.

En conclusión tenemos las siguientes corrientes tomadas de los pines de salida del microcontrolador:

Lógica Motor: 0.29mA
 Lógica Ventilador: 3.33mA
 Lógica Led: 0.048mA

Totalizando un consumo máximo de $2x(0.29mA) + 3.33mA + 2x(0.048mA) = 4mA$, que es mucho menor al especificado en su hoja de datos de 200mA (máximo consumo para todos los puertos a la vez).

De los valores anteriores es necesario resaltar que el bajo consumo de la lógica del motor no debe interpretarse como un sobre dimensionado en la doble etapa del buffer, ya que de no haber sido así (utilizando solamente el transistor de potencia) el consumo sería de 29mA, mayor a la máxima corriente admitida por cada pin (25mA).

5.2. Diseño y herramienta de software utilizado.

En la etapa final del diseño, con el fin de ahorrar espacio dentro de la cabecera, decidimos montar ambos bloques en una misma placa impresa. Luego de algunas pruebas y observando la gran cantidad de pistas que esto requería, recurrimos a la técnica de doble capa para su implementación. Con esta finalidad, y utilizando como herramienta de diseño el programa Eagle, comenzamos por el esquemático y el board que se muestran a continuación.

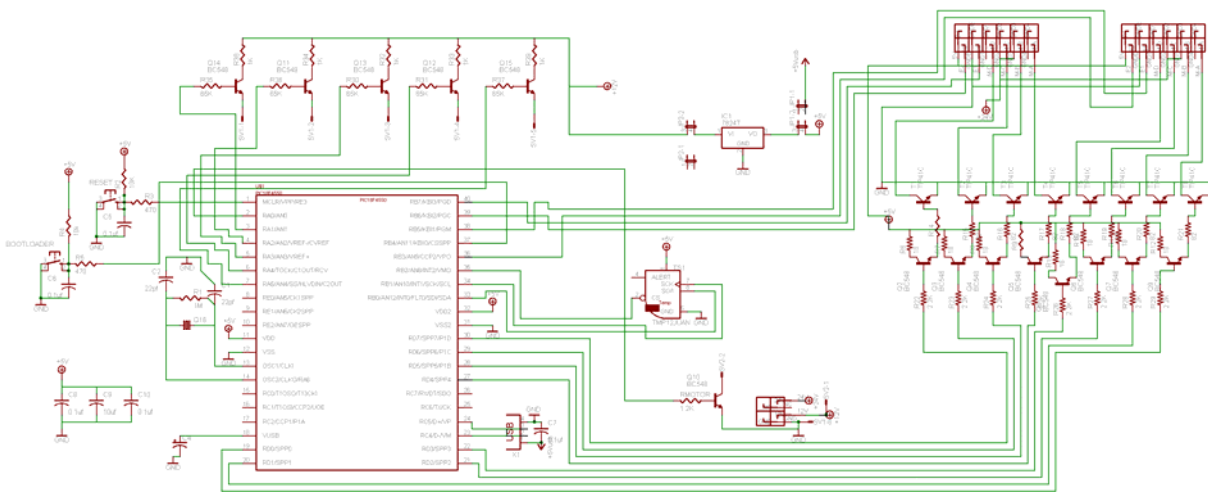


Fig.5-7 Esquemático de la Placa de Control.
(En el apéndice se puede observar en tamaño A3)

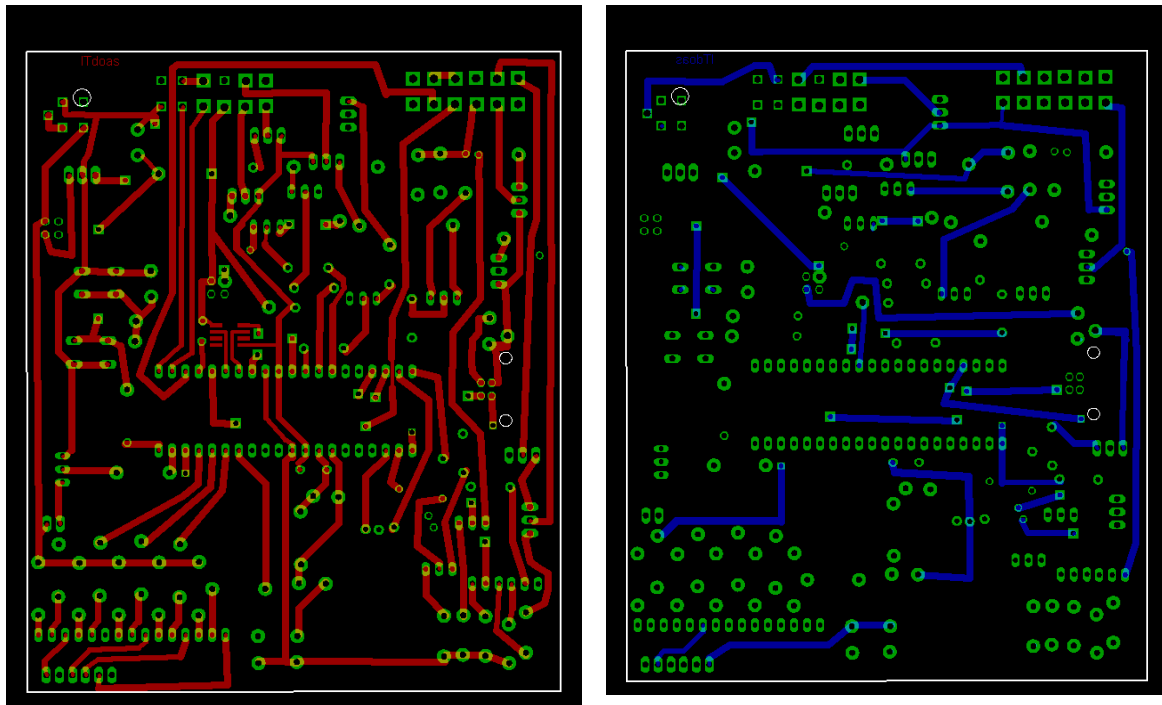


Fig.5-8 Posterior y anterior del Board doble capa.

5.3. Construcción de la Placa

El diseño de la placa de control consta de los siguientes componentes:

- Plaqueta de cobre doble capa.
- Conectores para la alimentación y las señales de entrada-salida.
- Resistencias de la serie E12 de distintos valores.
- Capacitores
- Cristal
- Transistores BC548 y TIP41C
- Regulador de voltaje L7805 (12v – 5v)
- Switches
- Microcontrolador PIC 18F4455

A continuación, en la figura 5-9, se muestra un esquema con la distribución de las conexiones de entrada-salida de la placa y luego se describirán los conectores que intervienen en cada conexión.

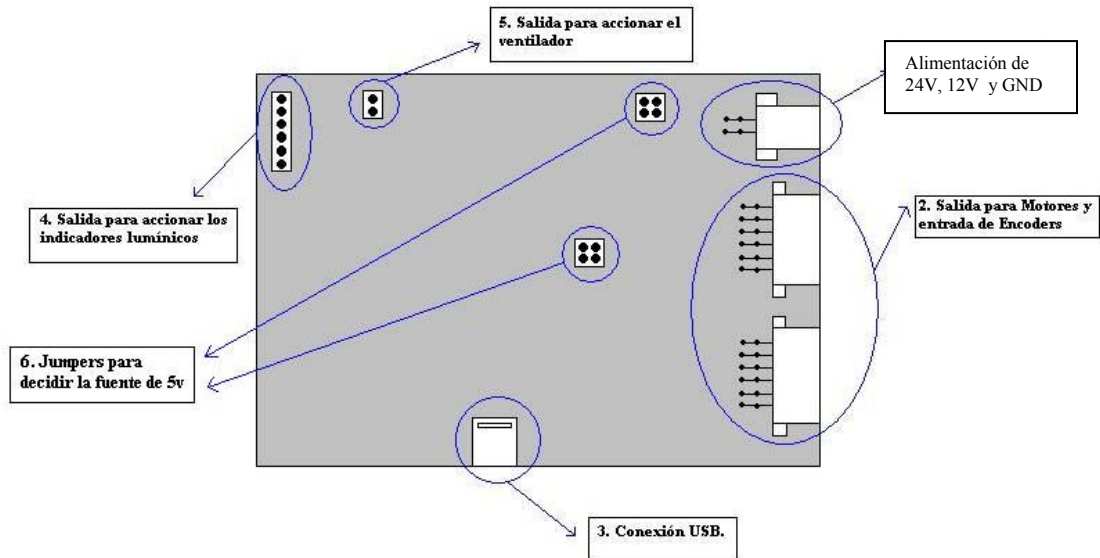


Fig.5-9 Esquema Placa de Control

5.3.1. Descripción de los conectores

1. El conector que provee a la placa las tensiones de 24v, 12v y la referencia del circuito.

En la siguiente vista frontal, figura 5-10, se observa la distribución de terminales.

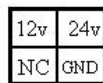


Fig.5-10

2. Conector para los motores y los encoders con la placa.

En los motores, se liga a los terminales de cada bobina (4 en total) con los 24v y el colector del TIP a la que esta asociada. Y en cuanto a los encoders, los provee de la alimentación de 5v, tierra y recibe las señales de datos que estos emiten hacia el microcontrolador (2 en total).

La siguiente es una vista frontal, figura 5-11, de dicho conector.

24v	24v	24v	24v	B4	B3
GND	EA	EB	5v	B1	B2

Fig 5-11

Donde B y E simbolizan bobina del motor y señal del encoder respectivamente.

3. Es un conector USB hembra tipo B que conecta al servidor con la placa. Se constituye de 2 señales de datos, alimentación de 5v y tierra.
 4. Es el conector de los leds. Son 6 pines de los cuales uno es tierra y los restantes 5 ligam al emisor de los transistores BC548 con el cátodo de cada led.
- Nota: si bien no se utiliza un quinto led, se da la conexión para una futura implementación.

La figura 5-12 es una vista superior del conector.

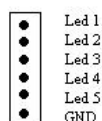
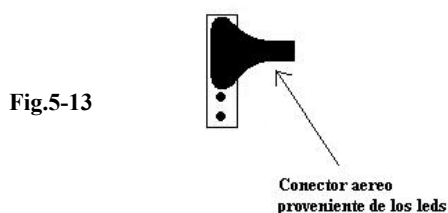


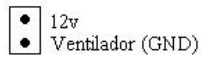
Fig.5-12

Nota: por un tema de practicidad los leds obtienen la referencia directamente de la entrada de tierra en la cabecera, por lo tanto al conectar en la placa el cable que proviene de los mismos tenga presente la siguiente vista superior.



5. Es el conector del ventilador. Son dos pines que ligan a los terminales del motor con 12v y el colector del transistor BC548 correspondiente.
- La siguiente es una vista superior, figura 5-14, del esquema del conector.

Fig.5-14



Nota: A la hora de conectar el ventilador tenga presente que el cable rojo del mismo debe ir a 12v.

6. Estos conectores se utilizan para decidir si los 5v de la placa se tomarán del regulador de voltaje 12v – 5v o del cable USB.

A continuación se muestra la forma de conectarlos para lograr el objetivo deseado.

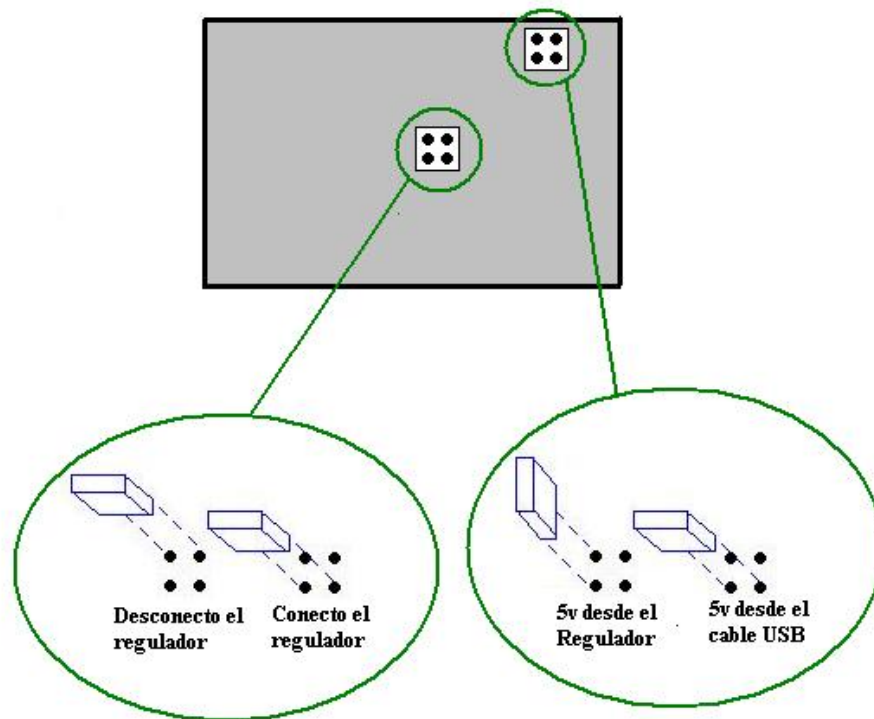


Fig.5-15

La desconexión del regulador de voltaje tiene que ver con un ahorro de energía y se aplica cuando los 5v son tomados desde el cable USB .

A partir del board de la placa elaborado en la etapa de diseño y de la compra de los conectores y componentes eléctricos mencionados anteriormente procedimos a la construcción de placa de Control.

A continuación, en la figura 5-16, se muestran fotografías de la parte anterior y posterior de dicha placa.

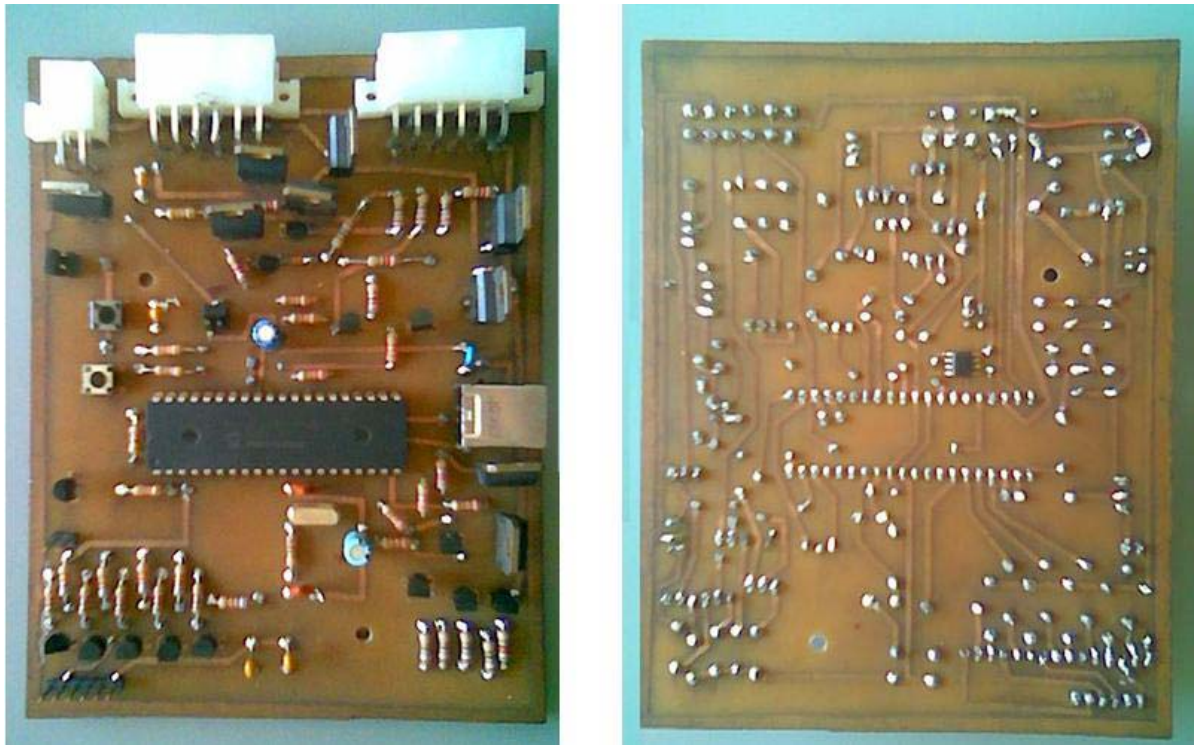


Fig.5-16 Fotografía de la Placa

5.4. Conexión Hardware entre el Servidor y la Cabecera

La conexión entre el servidor y la cabecera se compone de un cable USB (datos y opcional alimentación de 5V) ligado a un cable de tres conductores bajo goma (alimentación). La distribución de pines en el chasis de la conexión de alimentación tanto en el servidor como en la cabecera se muestra en la figura 5-17.

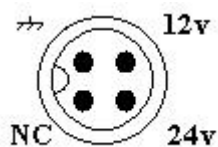


Fig.5-17

5.5. Protocolo de Comunicación entre el Servidor y la Placa.

Antes de comenzar a escribir los códigos de programa que se ejecutarán tanto en el servidor como en la placa de control, es necesario definir un protocolo de comunicación, es decir, el lenguaje que utilizaran dichos programas para intercambiar los datos. Gracias a que el microcontrolador cuenta con interfaz USB solo necesitamos definir los comandos que se utilizarán desde el servidor para setear las variables del microcontrolador que rigen el funcionamiento de los distintos periféricos (motores, ventilador, indicadores). A continuación se muestra una tabla con los comandos y una breve descripción de los mismos.

Valor del Byte de Comando	Descripción
0xFF	Reset
0x31	Ventilador
0x32	Absolute
0x33	Lee_pos
0x34	Lee_Temp.
0x35	Estado
0x36	LEDs_ON
0x37	Relative
0x38	Lee_LEDs
0x39	Stop
0x00	Versión

5.5.1. Descripción de los comandos

Cada uno de las instrucciones se compone de un Byte de comando seguido de una serie de Bytes de datos. La cantidad de estos últimos depende de cada comando.

A continuación se muestran todos los comandos con una descripción de los mismos, bytes del comando y bytes de respuesta.



Fig.5-16

Reset

Descripción: resetea al microcontrolador.

Bytes del Comando

Byte 0
0xFF

Cooler

Descripción: Se encarga del setear el modo del ventilador, así como de la temperatura de encendido para el modo automático.

Modo = 0x01 → encender ventilador
Modo = 0x02 → apagar ventilador
Modo = 0x03 → modo automático.

Temperatura: valor de la temperatura de encendido

Bytes del Comando

Byte 0	Byte 1	Byte 2
0x31	Modo	Temperatura

Bytes de Respuesta

Byte 0
0x31

Absolute

Descripción: Se encarga de setear las variables correspondientes a las coordenadas de la posición final de cada motor. Son necesarios tres bytes para la variable de posición de cada uno de ellos. Los tres primeros bytes de datos corresponden al motor horizontal y los restantes tres al vertical.

Bytes del Comando

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x32	H_LSB	H_MedioSB	H_MSB	V_LSB	V_MedioSB	V_MSB

Bytes de Respuesta

Byte 0
0x32

Lee_pos

Descripción: Obtiene desde el microcontrolador los valores de las variables de posición de ambos motores.

Bytes del Comando

Byte 0
0x33

Bytes de Respuesta

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
H_LSB	H_MedioSB	H_MSB	V_LSB	V_MedioSB	V_MSB

Lee_Temp.

Descripción: Obtiene desde el microcontrolador los valores de las variables que este a su vez obtiene del sensor de temperatura.

Bytes del Comando

Byte 0
0x34

Bytes de Respuesta

Byte 0	Byte 1
T_MSB	T_LSB

Estado

Descripción: Lee el estado de los motores

Estado_H = 0 → motor en movimiento

Estado_H = 1 → motor en detenido

Análogo para **Estado_V**

Bytes del Comando

Byte 0
0x35

Bytes de Respuesta

Byte 0	Byte 1
T_MSB	T_LSB

LEDs_ON

Descripción: Se utiliza para encender los leds o ponerlos intermitentes.

LED_ON → los 5 primeros bits indican con 1's los led a prender

Ej.: 00010100, se encienden los leds 3 y 5, apagándose el 1,2 y 4

TOOGLE → los 5 primeros bits indican con 1's los led a poner intermitentes

Bytes del Comando

Byte 0	Byte 1	Byte 2
0x36	LED_ON	TOOGLE

Bytes de Respuesta

Byte 0
0x36

Relative

Descripción: Se encarga de setear las variables correspondientes a las coordenadas de la posición final de cada motor, de forma tal que el valor que aparece en los bits de datos, corresponde a un desplazamiento relativo a la posición que se encuentran los motores.

Bytes del Comando

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
0x37	H_LSB	H_MedioSB	H_MSB	V_LSB	V_MedioSB	V_MSB

Bytes de Respuesta

Byte 0
0x37

Lee_LEDs

Descripción: Se utiliza para leer el estado de los leds (encendidos, apagados o intermitentes).

LED_ON → los 5 primeros bits indican con 1's los led que están encendidos

Ej.: 00010100, encendidos los leds 3 y 5, apagados el 1,2 y 4

TOOGLE → los 5 primeros bits indican con 1's los led que están intermitentes

Bytes del Comando

Byte 0
0x38

Bytes de Respuesta

Byte 0	Byte 1
LED_ON	TOOGLE

Stop

Descripción: Este comando se encarga de detener el movimiento de ambos motores.

Bytes del Comando

Byte 0
0x39

Bytes de Respuesta

Byte 0
0x39

Version

Descripción: Obtiene los valores correspondientes a la versión del Microcontrolador.

Bytes del Comando

Byte 0
0x00

Bytes de Respuesta

Byte 0	Byte 1	Byte 2	Byte 3
DATO 1	DATO 2	DATO 3	DATO 4

5.6. Firmware del Microcontrolador

El Firmware es un conjunto de instrucciones de programa para propósitos específicos, grabado en una memoria tipo ROM, que establece la lógica que controla los circuitos electrónicos de un dispositivo. Al estar integrado en la electrónica es en parte hardware pero también es software, ya que proporciona lógica y se expresa en algún tipo de lenguaje de programación.

5.6.1. Herramienta de Software

El Firmware de nuestra aplicación fue escrito enteramente en C utilizando el MPLAB C18, un compilador de C que soporta el estándar de C ANSI '89 y pensado para trabajar de forma conjunta con el MPLAB IDE de Microchip, que es el entorno de desarrollo que utilizamos para realizar la programación, simulación, y depuración paso a paso de nuestro microcontrolador.

5.6.2. Enfoque del problema y primeros pasos

El diagrama de bloques, de la figura 5-17 presenta el escenario en el que nos encontramos en nuestro caso particular.



Fig.5-17

Necesitamos entonces desarrollar un Firmware para el Microcontrolador que actúe como intermediario entre los periféricos y las órdenes que éstos reciben desde el servidor. Para ello nos basamos nuevamente en la aplicación de Microchip, “PICDEM™ FS USB”, en donde el Firmware del PIC se compone de los siguientes bloques.

En primer lugar un bloque de inicialización en el cual se definen variables globales y se programan las interfaces del microcontrolador. Luego un bloque compuesto por varias funciones específicas de la aplicación entre las cuales destacamos aquellas significativas para nuestra implementación:

- Rutina para adquisición de temperatura
- Rutina de atención del USB

La primera es muy útil dado que contamos con el mismo sensor de temperatura que la aplicación de Microchip. La segunda es vital dado que se encarga de mantener el contacto con el servidor atendiendo el protocolo USB y evitando así la desconexión entre el servidor y el microcontrolador. Esta rutina debe ser llamada en forma frecuente para lograr el objetivo anterior. Un tercer bloque en el que ubicamos a una función denominada “Service Request”, la que, como su nombre lo indica, es la encargada de recibir las peticiones del servidor vía USB y actuar sobre las variables llamando además a las funciones del segundo bloque e intercambiando datos con la PC a través del USB. Finalmente una función **main()** que no es otra cosa que un loop infinito de Service Request seguida de la rutina de atención del USB.

Una vez en este punto y con base en los bloques anteriores comenzamos a escribir los códigos de nuestra aplicación respetando el protocolo comentado en el numeral anterior.

5.6.3. Implementación del Firmware

El Firmware para el microcontrolador responde al diagrama de bloques de la figura 5-18.

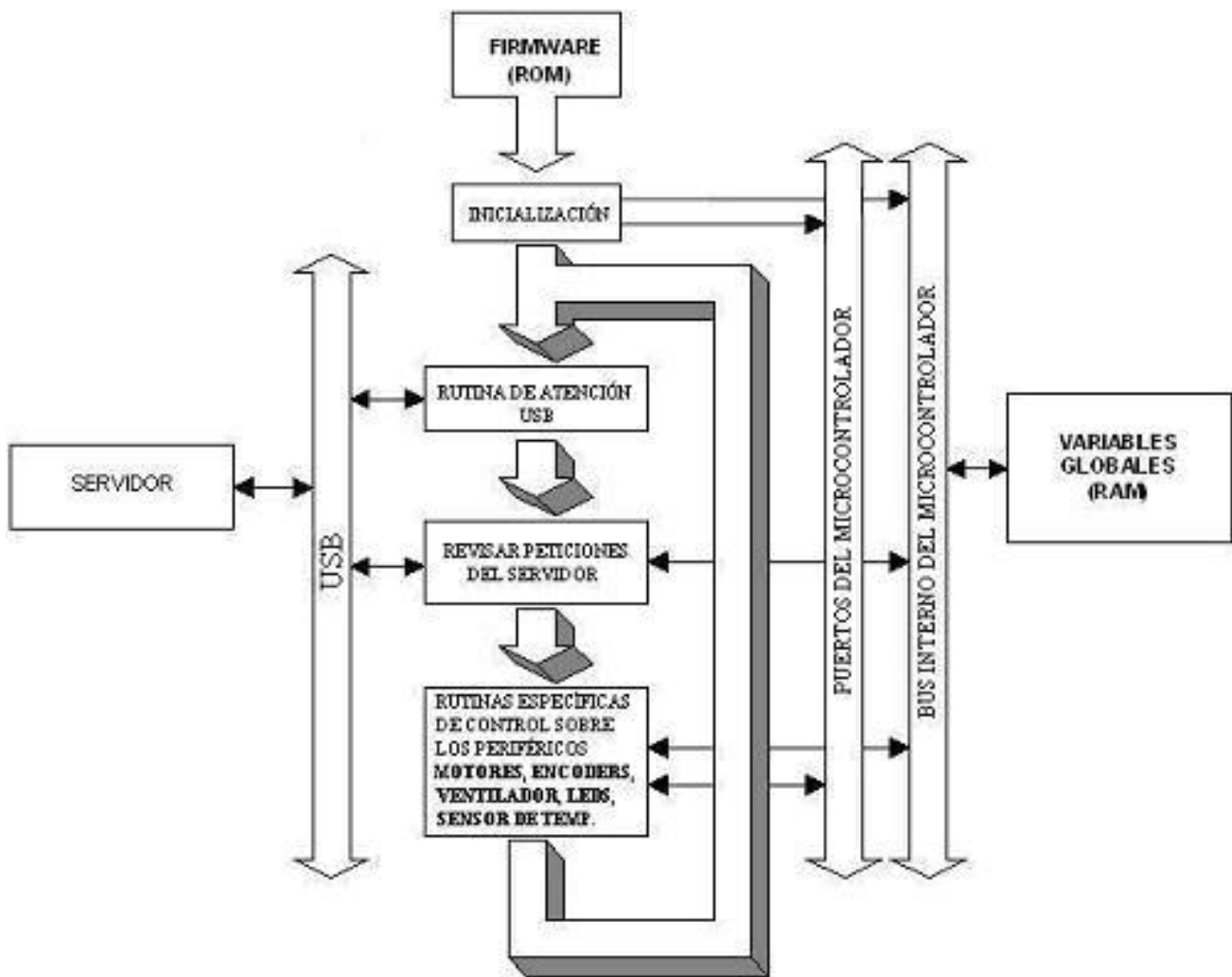


Fig.5-18

El mismo representa el cuerpo de la función **main()** compuesto de las rutinas específicas y la interacción de las mismas tanto con el servidor como con las variables globales y las interfaces del PIC. Es importante destacar que la interacción con el servidor se realiza respetando el protocolo mencionado anteriormente entre el servidor y la placa. A continuación se da una breve descripción de cada una de las rutinas.

Inicialización

Nombre: InitializeSystem(void)

Descripción: Esta asociada con la configuración base para poder utilizar las características de la placa. Entre ellas:

- Programa los puertos del microcontrolador ya sea para salida o entrada y a su vez cada uno de sus pines como digitales o analógicos.
- Configura los registros del bloque USB para la comunicación con el servidor.
- Setea las variables relacionadas con los indicadores lumínicos para indicaciones de arranque

Rutina de Atención al USB

Nombre: USBTasks(void)

Descripción: Mantiene el contacto con el servidor atendiendo el protocolo USB

Precondición: debe llamarse luego de haberse ejecutado InitializeSystem(void).

Nota: debe ejecutarse en forma frecuente de lo contrario el dispositivo es desconectado automáticamente.

Revisar Peticiones del Servidor

Nombre: PollingUsb(void)

Descripción: Atiende las peticiones del servidor que le llegan desde el USB. La secuencia de acciones que se ejecutan en esta rutina son las siguientes:

- Lee del USB el valor del Byte de comando y sus sucesivos bytes de datos y los almacena en una variable tipo vector
- De acuerdo con el protocolo establecido utiliza el byte de comando para decidir sobre qué periférico va a actuar
- Setea las variables relacionadas con dicho periférico con los valores de datos almacenados en el vector y/o lee alguna variable que deba retornar.
- Crea un vector para devolver los valores que retornarán y escribe dicho vector en el USB.

Nota: La rutina escribe en el USB tantos datos como espera el servidor de acuerdo con el protocolo.

Rutinas Específicas de Control Sobre los Periféricos

Rutina: Control sobre los motores.

Nombre: motores(void)

Descripción: Compara el valor de la posición actual, con el valor de la posición final para cada motor y si son distintos actúa sobre el puerto de los motores de forma de activar la bobina que corresponda. Esto último tiene que ver con la última bobina que se activó y el signo de la comparación para saber en que dirección debe moverlo.

Nota: observe que los motores se seguirán moviendo hasta que ambas cantidades sean iguales. El valor de la posición final es seteado en forma externa por el servidor si este lo desea. A continuación veremos en la rutina que controla los encoders, como es actualizada la posición actual.

Rutina: Control sobre los encoders.

Nombre: machin_e_h(void), machin_e_v(void)

Descripción: Es importante mencionar que en la rutina de inicialización se lee el valor de los encoder por el puerto B para establecer el estado inicial. Cada vez que se entra a esta rutina se lee nuevamente la señal de los encoders. Basándose en el estado y la entrada reconoce si hubo movimiento y en que dirección positiva o negativa y de acuerdo a esto setea la posición actual incrementando o decrementando respectivamente. Además se recuerda el nuevo estado en la variable que almacena el estado del encoder.

Rutina: Control sobre el ventilador.

Nombre: ventilador(void)

Descripción: Actúa sobre el bit del puerto A correspondiente al ventilador para prender, apagar, o poner en modo automático según lo indiquen las variables de dicho periférico.

Nota: estas variables son inicializadas en un valor por defecto cuando son creadas, pero pueden ser modificadas por el servidor en la rutina de servicio de peticiones cuando éste lo desee.

Rutina: Control sobre los leds.

Nombre: leds(void)

Descripción: Actúa sobre los bits del puerto A correspondientes a los indicadores para prender, apagar, o poner en modo intermitente según lo indiquen las variables de dicho periférico.

Nota: estas variables son inicializadas en la rutina de inicialización pero pueden ser modificadas por el servidor en la rutina de servicio de peticiones cuando éste lo desee.

Rutina: Control sobre el sensor de temperatura.

Nombre: AcquireTemperature(void)

Descripción: Lee de los bits del puerto B correspondientes a dicho periférico el valor de la temperatura.

5.6.4. Actualización del Firmware

En el caso de que se desee actualizar este Firmware, por ejemplo para la incorporación de un nuevo periférico, solo es necesario agregar sus códigos a este Firmware y reprogramar la EPROM del microcontrolador siguiendo los siguientes pasos:

- Compile el códigos con el MPLAB IDE de Microchip para generar los archivos de salida que se grabaran en el PIC. Tenga cuidado y utilice un código que soporte la aplicación Bootload de Microchip.
- Mantenga presionado el botón Bootload existente en la placa y pulse el botón reset, luego suelte el primero y el microcontrolador entrará en el mencionado “Bootload Mode”.
- Utilizando la aplicación de microchip denominada “Demo Tool” elija modo Bootload y cargue el archivo de salida que generó anteriormente. Finalmente elija la opción programar.

6. Software

El software del proyecto, se puede dividir en tres categorías que se desarrollarán en este capítulo; Software de control (o drivers), software de aplicación web y software desarrollado por terceros.

Por otro lado se explicarán algunas partes consideradas claves en la posterior actualización y mejoramiento del sistema en futuras versiones que es la modularización del software.

6.1. Software de Control (Drivers)

6.1.1. Comunicación USB

Debido a que en este proyecto la comunicación entre el servidor y el dispositivo es mediante el protocolo USB, es que fue necesario contar con una librería que nos permitiera programar la comunicación mediante este protocolo, pero en un nivel alto. Para ello utilizamos una librería de código abierto llamada *libusb* que viene por defecto en la mayoría de las distribuciones de Linux.

6.1.2. Endpoints

Lo mas básico de la comunicación USB son los endpoints. Un endpoint USB puede transmitir información solo en una dirección, o bien desde la computadora (host) al dispositivo (OUT endpoint) o del dispositivo hacia la computadora (IN endpoint). Los endpoint pueden ser de los siguientes tipos dependiendo de cómo transmite los datos:

Control

Los endpoints de control son usados para permitir acceso a diferentes partes del dispositivo USB. Son utilizadas comúnmente para configurar dispositivos, recibir información, enviar comandos al dispositivo, recibir reportes de los mismos. Estos endpoints son usualmente de tamaño pequeño. Cada USB tiene un endpoint de control llamado “endopint 0” que es usado por el USB para configurar el dispositivo cuando se conecta. Esta transferencia esta garantizada por el protocolo USB para que siempre tenga reservada suficiente ancho de banda.

Interrupt

Los endpoint de interrupción transfieren pequeñas cantidades de datos en periodos fijos cada vez que el host USB le pida datos al dispositivo. Estos endpoints son usados basicamente para métodos de transporte para teclados y ratones USB. También son

utilizados para enviar datos al USB para controlar al dispositivo, pero no para transferir grandes cantidades de datos. Esta transferencia esta garantizada por el protocolo USB para que siempre tenga reservada suficiente ancho de banda

Bulk

Estos endpoints transfieren grandes cantidades de datos y son por tanto mucho mas grandes que los endpoints de interrupción. Son pensados para dispositivos que necesitan transferir datos sin pérdida. Esta transferencia no esta garantizada por el protocolo USB para ser transferido en al tiempo especificado. Si no hay suficiente espacio en el bus para mandar el paquete bulk entero, éste se parte en varias transferencias. Estos endopints están presentes por lo general en impresoras o dispositivos de almacenamiento como pendrives.

Isochronous

Estos endpoints también transfieren grandes cantidades de datos, pero los datos no siempre son garantiza su pasaje. Estos endpoints se utilizan en dispositivos en los cuales se permiten pérdidas de datos, y se confía más en mantener un flujo constante de datos. Son utilizados principalmente para aplicaciones de audio y video.

Los endpoint de control y bulk se usan para transferencias asincrónicas, cuando el driver decida usarlas. Los endopint de interrupción e isochronous son periódicos. Esto significa que transmiten datos a intervalos de tiempo fijos continuamente, lo que causa que sus anchos de banda sean reservados por el nuclueo USB.³

En nuestro caso el formato de transferencia elegido fue Bulk para las trasferencias de datos.

³ LDD – Linux Device Drivers – O 'Reily – tercera edición 2005 ISBN: 0-596-00590-3 [M]

6.1.3. Driver del Espectrómetro



Fig.6-1 Espectrómetro S2000 + adc1000
Imagen tomada de Ocean Optics

Durante el desarrollo del dispositivo surgió la necesidad de tener un control del espectrómetro existente (OceanOptics S2000 con un conversor ADC-1000) por medio de una interfaz de consola en Linux, lamentablemente la empresa OceanOptics no proveía drivers para esta plataforma sino que ofrecía toda una suite de desarrollo escrita en Java (multiplataforma) que no cumplía con todos nuestros requerimientos y que tenía un costo muy superior a nuestro presupuesto. Por lo que surgió la idea de construir un driver propio para este dispositivo.

Afortunadamente para nosotros pudimos conseguir una hoja de datos de OceanOptics para la comunicación utilizado por el ADC-1000 por el puerto USB de donde obtuvimos el siguiente protocolo de comunicación además de contar con software como USBSnoop que nos permitió correr el dispositivo en otra plataforma y de esa forma investigar los comandos y parámetros intercambiados entre la PC y el dispositivo.

EP2 Command Byte Value	Description	Version
0x01	Reset ADC1000-USB	1.00.0
0x02	Set Integration Time	1.00.0
0x03	Set Strobe Enable Status	1.00.0
0x04	Reserved	1.00.0
0x05	Query Information	1.00.0
0x06	Write Information	1.00.0
0x07	Write Serial Number	1.00.0
0x08	Get Serial Number	1.00.0
0x09	Request Spectra	1.00.0
0x0A	Set Trigger Mode	1.00.0
0x0B	Set Spectrometer Channel	1.00.0
0x0C	Set Continuous Strobe Rate	1.00.0
0x0D	Set Master Clock Rate	1.00.0

Tabla 6-1 Comandos del dispositivo ADC-1000

Además de información de cómo comunicarnos con el dispositivo se tuvo que aprender cómo ordenar el espectro una vez adquirido, ya que el dispositivo lo devuelve 4096 bytes ordenados de la siguiente manera.

Packet 0 – LSBs for first 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 0 LSB	Pixel 1 LSB	...	Pixel 63 LSB

Packet 1 – MSBs for first 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 0 MSB	Pixel 1 MSB	...	Pixel 63 MSB

Packet 2 – LSBs for 2nd group of 64 pixels

Byte 0	Byte 1	...	Byte 63
Pixel 64 LSB	Pixel 65 LSB	...	Pixel 127 LSB

Tabla 6-2 Bytes devueltos por el ADC-1000

6.1.4. Driver de la placa de control

La placa de control es un dispositivo elaborado por nosotros, por lo que teníamos que realizar el software para la comunicación con este. Para esto se diseñó un protocolo de comunicación, analizado en el capítulo anterior, y se diseñó un único programa en C (llamado target) que nos permitiera acceder a todas las funcionalidades de este según los parámetros que reciba desde la línea de comandos como se observa en el diagrama de flujo de la figura 6-2.

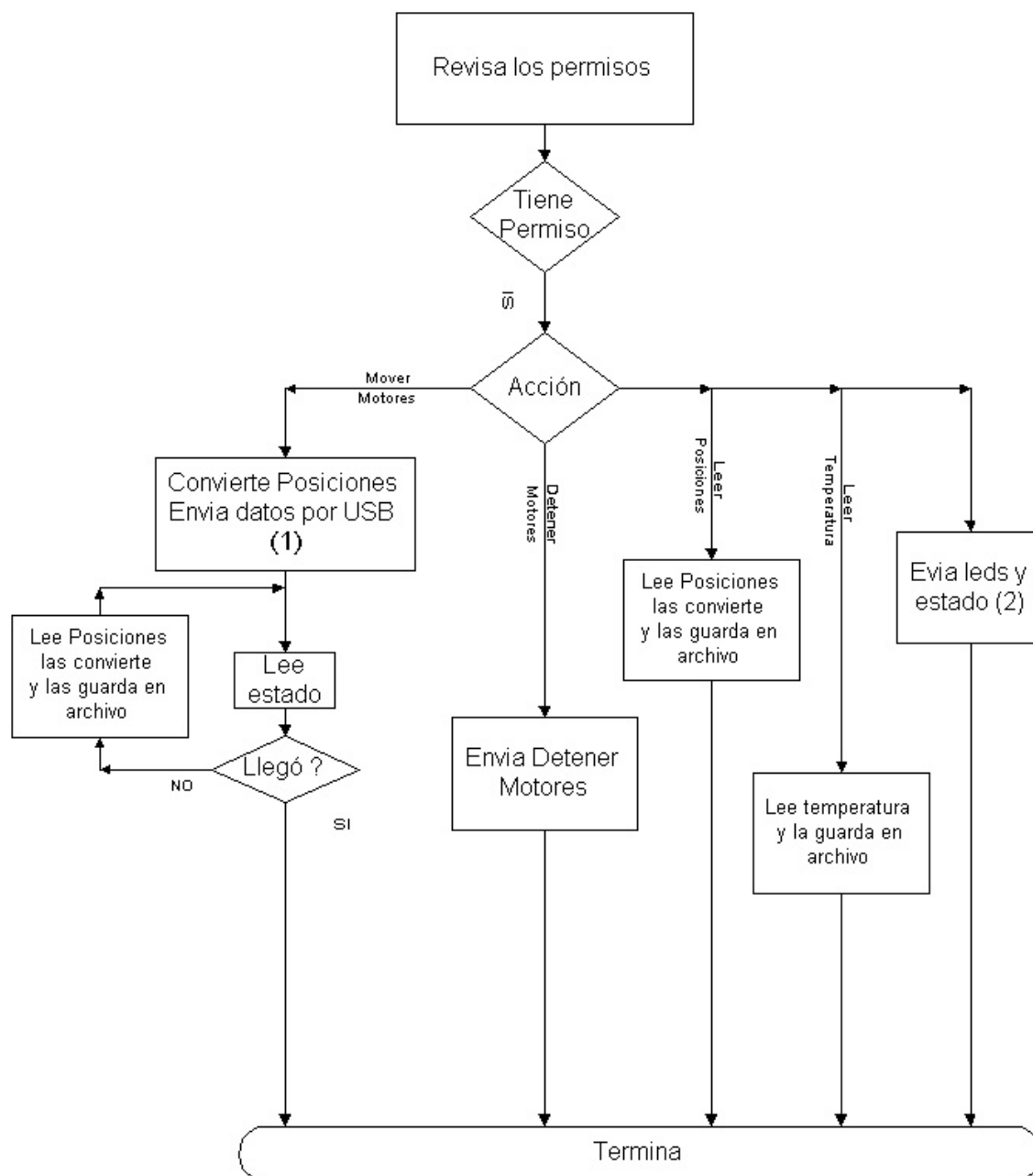


Fig.6-2 Diagrama de flujo del programa target

- (1) Los datos se convierten para no enviar valores negativos hacia el USB
 (2) Los estados de los leds pueden ser rojo, verde, rojo intermitente, etc. .

Este programa solo provee comunicación hacia y desde la placa de control, luego es la misma placa la que se encarga de mover los motores y controlar todos los dispositivos conectados a ella.

Por ultimo se decidió que el programa no retornará (terminará) hasta haber alcanzado la posición deseada, lo que facilita la gestión del mismo.

La sintaxis del programa es la siguiente:

target [opcion] ([valor1]) ([valor2])

Donde opción puede ser:

- a X Y :Mueve al target a la posición (X,Y)
- r x y :Mueve al target x pasos en dirección horizontal e y pasos en dirección vertical
- t :Devuelve la temperatura
- p : Devuelve la posición actual y el estado de los motores
- ax x :Mueve el target a la posición X dejando quieta la dirección Y
- ay y :Mueve el target a la posición Y dejando quieta la dirección X
- rx x :Mueve el target x pasos en dirección horizontal
- ry y :Mueve el target y pasos en dirección vertical
- s :Detiene el movimiento
- e :Devuelve el estado de los motores 1=andando 0= detenidos
- v a t :Control del ventilador a=1 On a=2 Off a=3 auto t temperatura de encendido
- l n a :Controla el led n según a. a=1 On a=2 Off a=3 Toogle (intermitente)
- h :Ayuda

y valor1 y valor2 dependen de la opción.

6.1.5. Permisos

Para evitar congestionamientos y pérdidas de datos u omisiones de comandos se creó un sistema de permisos que evita que dos programas diferentes accedan al bus USB al mismo tiempo⁴, para esto existe una variable que indica si se puede o no utilizar el bus.

⁴ Esto fue una mejora empírica ya que cuando no existía este sistema se detectaban fallas de diversos tipos en la comunicación (Error 11- device temporary unavaliabe)

6.2. Aplicación Web

Como el dispositivo se maneja completamente a través de una interfaz web, la mayor parte de la programación esta orientada a servicios de este tipo. Para esto se decidió utilizar CGI (Common Gateway Interfaz) escritos en Perl, ya que no requieren software extra por parte del cliente y proveían la conectividad necesaria cliente-servidor.

Por otra parte los CGI en Perl son utilizados por la mayoría de los módulos de webmin, y ya que era necesario desarrollar un módulo propio para este servidor se optó por ellos⁵. A continuación se observa una captura autoexplicativa de la interfaz de usuario Administrador, que es donde se utilizan los CGI y luego se listan los mismos.

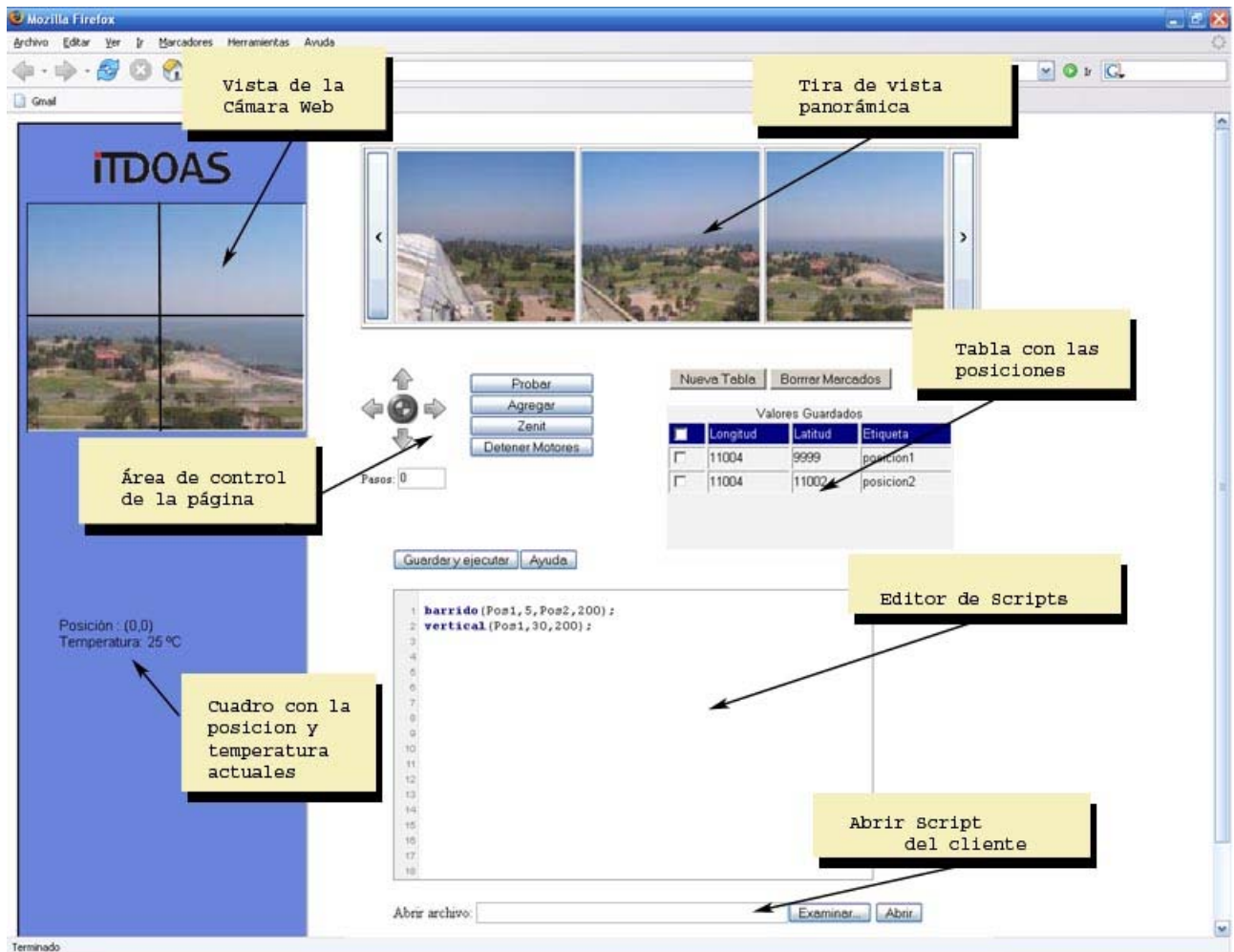


Fig.6-3 Captura de la pantalla de usuario Administrador

⁵ Ver mas adelante software desarrollado por terceros

6.2.1. CGI utilizados

- **index.cgi:**

Script que se encarga de cargar la página principal de configuración.

- **tabla.cgi:**

Si se la llama sin argumentos (por ejemplo cuando se carga al iniciarse la configuración) simplemente despliega la tabla con los valores ya programados.

Si se la llama con parámetros (cuando es invocada por el botón [Agregar]) toma los datos de la posición absoluta, tiempo de exposición y etiqueta de la página y los agrega a la tabla y a un archivo de configuración.

- **borrar.cgi:**

Recibe como parámetros los valores de las filas a borrar de la tabla y los elimina tanto de esta como del archivo de configuración. Esta función es invocada por el botón [eliminar marcados] en el marco de la tabla.

- **nueva.cgi.:**

Crea una tabla y archivo de configuración nuevos.

- **get_pos.cgi:**

Obtiene la posición absoluta total generada en el servidor y la despliega en la página.

- **mueve_motor_h.cgi (y mueve_motor_v.cgi):**

Toma la cantidad de pasos y sentido a moverse de la página, y le envía la instrucción apropiada al servidor. “mueve_motor_h” maneja el movimiento horizontal y “mueve_motor_v” el vertical.

- **stop.cgi:**

Detiene el movimiento de los motores. Este script esta pensado para casos en los que se le indicó un número de pasos al motor mayor al deseado y se esta sbrepasando al objetivo.

- **mover_a.cgi :**

Toma un valor de posición absoluta y le indica al servidor que dirija al objetivo a esa dirección. Pensado para las posiciones fijas de las fotografías superiores.

- **probar.cgi :**

Toma un valor de tiempo de exposición y adquiere un espectro mostrando gráficamente los resultados.

- **ejecutar.cgi :**

Guarda las rutinas, interpreta los datos y arma un script con estos, que ejecuta inmediatamente.

- **webcam.cgi :**

Enciende la cámara web hasta que el dispositivo se mueva nuevamente y la vuelva a apagar.

- **importar.cgi :**

Abre un archivo de la computadora cliente al editor de scripts.

- **devices.cgi :**

Lista los espectrómetros USB instalados en un selector de la página web.

- **sel_device.cgi :**

Cambia el espectrómetro por el elegido en la página web.

- **dark.cgi :**

Obtiene un espectro de oscuridad con tiempo de integración fijo (60000 ms)

- **offset.cgi :**

Obtiene espectro de offset (10000 espectros) con tiempo de integración mínimo 3ms

6.2.2. CGI para la interfaz de dispositivos móviles

- **importar4palm.cgi :**

Abre un archivo de la computadora cliente e inmediatamente interpreta los datos, arma un script y lo ejecuta.

- **Tabla4palm.cgi :**

Hace lo mismo que tabla pero cuando arma los datos los presenta en un formato más amigable para una interfaz como la palm.

6.3. Software desarrollado por terceros

Durante la elaboración del dispositivo se desarrolló buena parte del software pero también se utilizaron programas ya existentes que realizaban algunas de la tareas necesarias, y con un grado de desarrollo bastante avanzado. Cuando se eligieron estos programas se le dio importancia a que cumplieran con nuestras necesidades, fueran manejables desde consola y en lo posible fueran software libre (para evitar problemas de licencias). A continuación se explican los programas utilizados y qué papel cumplen dentro del software del dispositivo.

Apache

Software para servidor web, de amplio uso en el mundo entero. Sobre este corre la interfaz de usuario no administrador.

Webmin

Servidor especialmente construido para manejar conexiones seguras y para administrar computadoras en forma remota. Este producto también es software libre y es totalmente modularizable, permitiendo la creación de módulos propios. Sobre éste corre la interfaz de usuario Administrador, un módulo propio cuya construcción se explicará mas adelante en el apartado modularización.

GnuPlot

Software que permite fácilmente hacer cálculos matemáticos y graficar datos, incluso exportar las gráficas como imágenes. Su trabajo fue el de graficar los datos tomados del espectrómetro y exportarlos en formato PNG (imagen) para poder visualizarlos en la página web o descargarlos.

Webcam

Software que captura imágenes (parte del paquete xawtv) cada determinada cantidad de tiempo y las envía a un directorio (local o remoto) entre otras cosas. Su aplicación es enviar la fotos desde la cámara web hacia un directorio de donde son levantadas hacia las interfaces de usuario.

imageMagic

Completo editor de imágenes desde consola que permite automatizar procesos de edición. Es utilizado para recortar las fotos (al estar la cámara con un pequeño telescopio la imagen se capta circular).

6.4. Modularización del software

Una de las metas principales del proyecto es que fuera modularizable, es decir, que se pudieran incluir modificaciones o utilidades extras por medios de módulos desarrollados externamente. Por eso se diseñó y eligió todo el software que lo permitiera

6.4.1. Sistema Operativo

Como sistema operativo base para el servidor se utilizó Slax que es una versión reducida de Slackware, pensada para ejecutarse desde un CD, y que usa el sistema unionfs, que permite que un sistema de solo lectura se comporte como sistema que permite escritura, el cual admite entre otras cosas instalarlo en medios con espacios reducidos (como pendrives o CD).

Cuando la computadora arranca desde el Live CD, es cargada la imagen del kernel de Linux (vmlinuz). Luego, se crean 4 MB de disco virtual (ramdisk) en la RAM de la PC. La imagen con Rootdisk (initrd.gz) es desempaquetada y montada como raíz del sistema de archivos. Los directorios de gran tamaño (como /usr, /opt,...) son montados directamente desde el CDROM (en modo de solo lectura).

Slax además permite ir acrecentando sus funcionalidades por medios de módulos, es por esto que se decidió que todo el software utilizado en el proyecto fuera un único módulo para este dispositivo. Además de algunos otros módulos con los programas que fueron desarrollados por terceros, los cuales se mencionan en el apartado anterior.

Los módulos de Slax son archivos con extensión **.mo** que se crean a partir de un directorio con los archivos (dispuestos según su ubicación final en el directorio raíz) mediante el comando `dir2mo` provisto por el propio Slax. Dichos módulos se guardan en el directorio `modules` del LiveCD (o del disco duro en una instalación live, ver apéndice III) y se cargan en forma alfabética, según su nombre, es por eso que el módulo `itdoas` se denomina `z_itdoas`.

En la página siguiente se puede ver la composición del módulo `Itdoas` junto con sus principales archivos explicados.

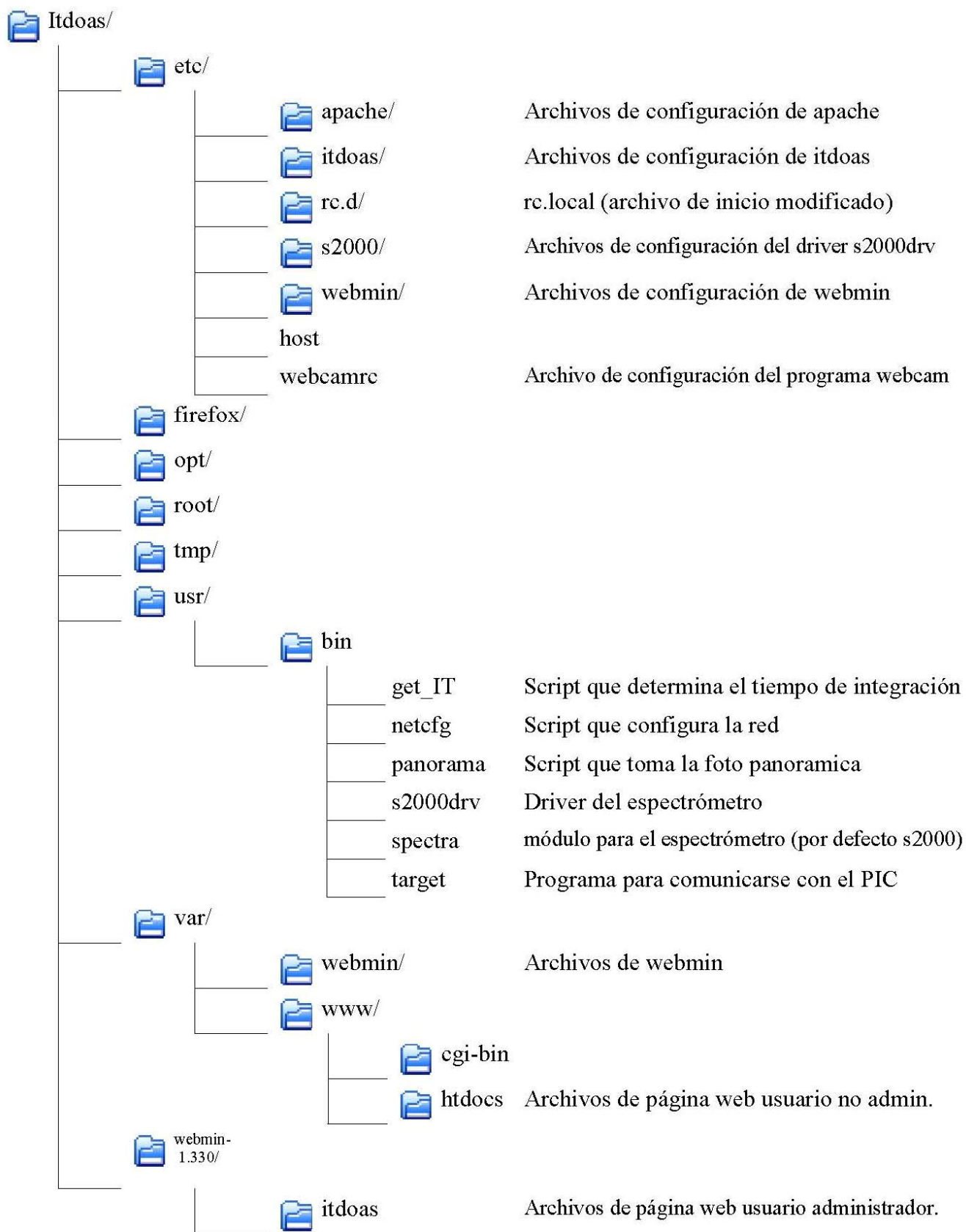


Fig. 6-4 Composición del módulo del ITDoas con sus principales archivos

6.4.2. Webmin

Un modulo de Webmin es un conjunto de scripts CGI agrupados en una carpeta dentro del directorio de webmin, que siguen ciertas reglas.

Para que un módulo pueda ser exhibido en el menú principal de webmin, debe contener por lo menos los siguientes archivos:

- ***Images/icon.gif:***

Icono a exhibir en el menú principal para este módulo. Debe ser de 48x48 píxeles y con el esquema de colores del resto de los iconos del menú principal.

- ***Module.info:***

Este archivo contiene la información sobre el módulo (solo module.info es obligatorio) y los sistemas operativos en los cuales será este funcional.

- ***Index.cgi:***

Primer script que se ejecuta cuando se selecciona este modulo en la página de webmin. Un módulo típico contará con muchos script cgi , que son ligados en index.cgi, cada uno realizando alguna función.

6.4.2.1. Configuración

- ***Lista de control de acceso (ACL)***

Al crear un nuevo módulo, este no se encontrará dentro de la ACL (lista de control de acceso) de ningún usuario Webmin, así que no podrá visualizarlo en el menú principal.

- ***Borrado de caché***

Para que el nuevo modulo pueda ser visto y utilizado primero hay que borrar el caché eliminando el archivo `/etc/webmin/modules.infos.cache` haciendo que webmin vuelva a fabricar una lista de módulos disponible en el directorio webmin

- ***Modificado del ACL***

Entonces para lograr que un nuevo módulo sea visible para un determinado usuario hay que editar el archivo `/etc/webmin/webmin.acl`, o utilizar el administrador de módulos para usuarios de Webmin para conceder el acceso.

La sintaxis para agregar un modulo a un usuario es la siguiente:

En el archivo webmin.acl

usuario: modulo1 modulo2 ... etc

6.4.3. Cambio de espectrómetro

Otro de los factores a considerar en cuanto a modularización es el eventual reemplazo del espectrómetro utilizado actualmente de forma tal que la interfaz web no tuviera que ser cambiada. Para ello se creó un archivo especial llamado spectra que actúa de intermediario entre la interfaz web y el driver del dispositivo, de tal modo que la comunicación web-spectra sea siempre igual pudiendo cambiar la comunicación spectra-driver (ver figura).

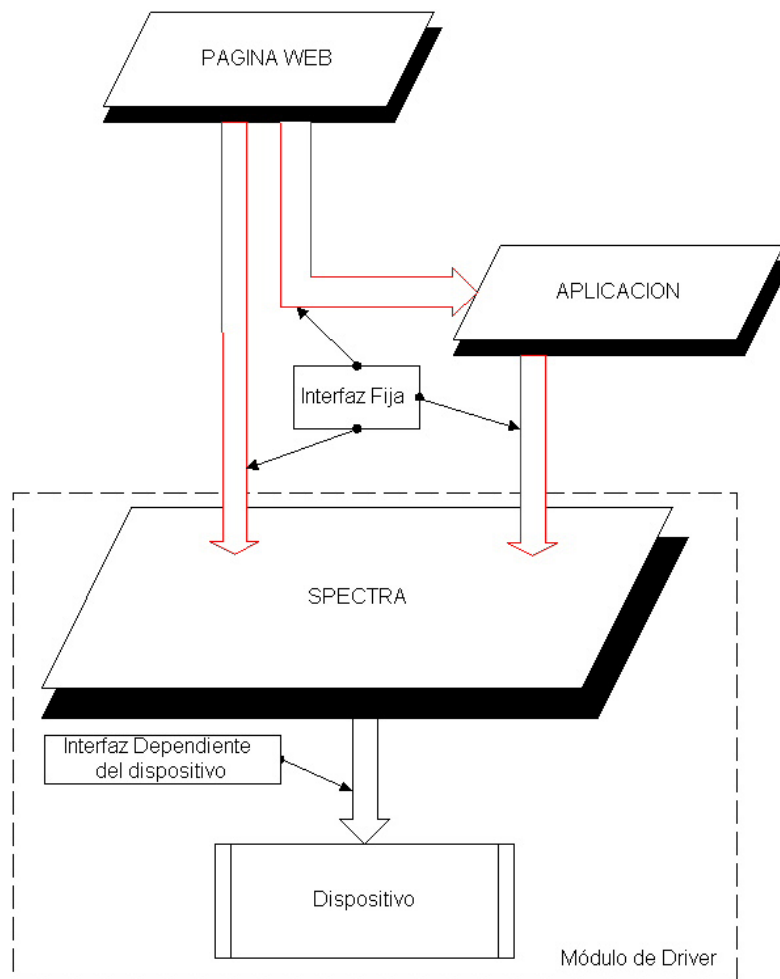


Fig.6-5 Esquema de comunicación con spectra

Para ello cada vez que se instale un nuevo dispositivo (espectrómetro) se deberá instalar los drivers para el Sistema Operativo y construir un script que reciba dos parámetros tiempo de integración y cantidad de espectros.

Ejemplo:

```
ADC2000 espectros tiempo
```

Y se encargue de traducir y ajustar estos valores para comunicarse con el driver y devolver un archivo con el espectro en formato estándar (.std) en el directorio

/tmp/espectro.std y otro archivo con dos columnas, donde la primera representa las longitudes de onda y la segunda los respectivos valores de cantidad de cuentas, en el directorio /etc/s2000/datos.dat.

Este script se debe copiar al directorio */etc/itdoas/devices* y deberá tener un nombre descriptivo del dispositivo (por ejemplo S2000-ADC1000) ya que este será el nombre que verá el usuario cuando quiera seleccionar este otro dispositivo.

Cada vez que se seleccione un dispositivo diferente al ya instalado, el software reemplazará el contenido del archivo spectra por el contenido del script.

Cabe destacar también que el script puede escribirse en cualquier lenguaje interpretado compatible con Linux (perl, bash, etc). Para evitar problemas de compatibilidad entre los archivos se recomienda utilizar un editor que no inserte caracteres DOS al final de línea, para esto se recomienda utilizar el editor incluido en el CD llamado Geany⁶.

⁶ En caso de utilizarse otro editor se puede convertir con el comando dos2unix en Linux.

6.4.4. Agregado de comandos

Si bien a la hora de diseñar los posibles comandos a utilizarse por el editor de script se tuvo especial recaudo en crear los suficientes comandos requeridos para las actividades de medición, se pensó en que en un futuro estos se puedan ampliar mediante el agregado de nuevos comandos. Por lo que cada comando implementado es un script que cumple con ciertos requisitos segundo el diagrama de flujo siguiente.

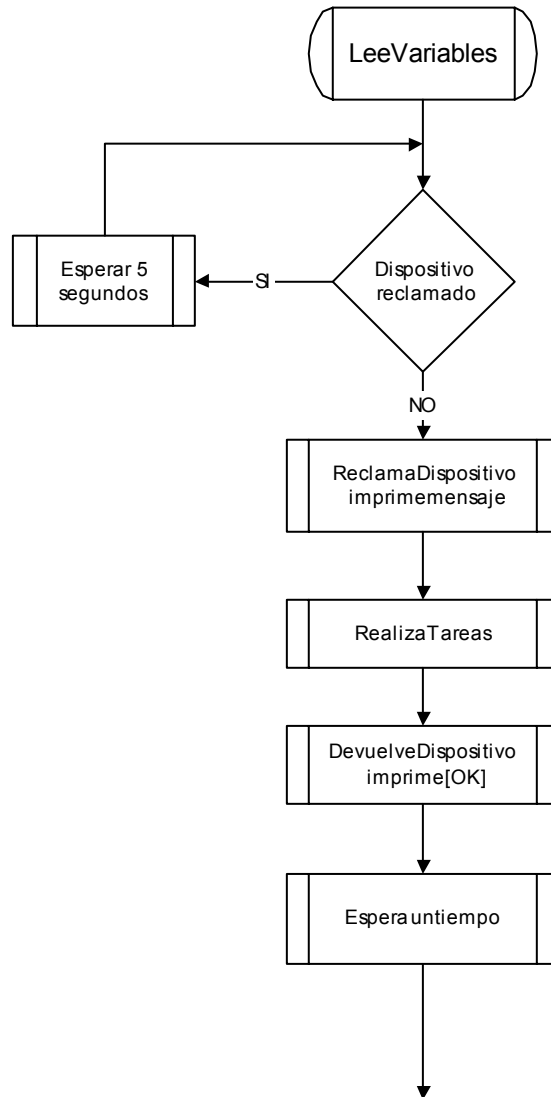


Fig.6-6 Diagrama de flujo de un script de comando

Estos script residen en el directorio */etc/itdoas/cmd* y deben contar además con permisos de ejecución.

6.5. Configuración

6.5.1. Panorama

Debido a que el dispositivo esta pensado para operar en diferentes lugares, cada vez que se lo inicia este ejecuta una rutina en la que almacena una serie de fotografías de su alrededor generándose luego en la interfaz de administrador la tira de imágenes que facilita la gestión de la misma. Esta rutina es realizada por el script panorama cuyo diagrama de flujo se aprecia en la figura 6-7.

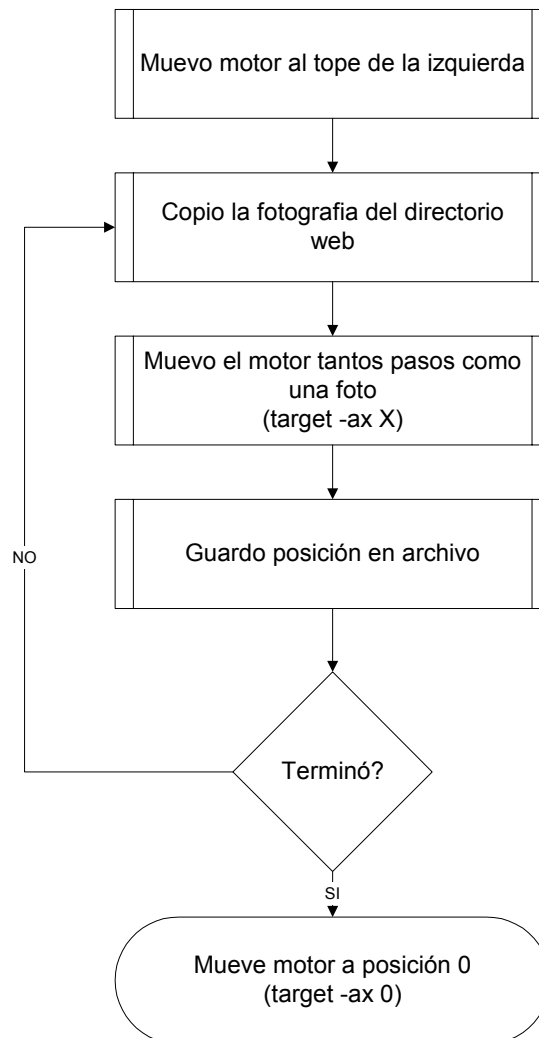


Fig.6-7 Diagrama de Flujo del script panorama

El script panorama (*/usr/bin/panorama*) es el encargado de llamar al programa target y de copiar las imágenes y así conformar la vista panorámica de la interfaz de administrador. Además panorama guarda las coordenadas de cada foto en un archivo de configuración para que el modulo ITDoas de Webmin pueda dirigir al telescopio al lugar de cada foto.

6.5.2. Configuración al inicio y al final

Para el correcto funcionamiento de los programas de control de motores algunos archivos de configuración deben ser modificados al inicio de cada sesión, por lo que la siguiente modificación debe agregarse al final del archivo de inicio de Linux (/etc/rc.local).

Primero encendemos el led superior en verde e intermitente para avisar que el sistema ha entrado en fase de configuración.

```
target -p
target -l 2 2
target -p
target -l 1 3
```

Luego Configuramos hardware de tarjeta de red

```
ndiswrapper -i /opt/wp1200/wpndis51.inf
ndiswrapper -m
modprobe ndiswrapper
depmod -a
```

Configuramos la red a valores por defecto con un pequeño script llamado netcfg.

```
/usr/bin/netcfg &
```

Una vez configurada la red iniciamos Webmin y configuramos el Firewall

```
echo /etc/webmin>/tmp/conf
iptables -A OUTPUT -o eth0 -p tcp -m state --state
ESTABLISHED,RELATED -j ACCEPT
modprobe -ae
/etc/webmin/start
```

Creamos links hacia un pendrive si existe con el script pendrive

```
/usr/bin/pendrive
```

Iniciamos ATD (programador de tareas)

```
atd
```

Configuramos la red desde el pendrive si hay script de configuración y Ejecutamos el script panorama

```
/media/sda1/itdoas/conf/netcfg &
/usr/bin/panorama
```

Encendemos el led de arriba en verde y fijo para indicar que el dispositivo esta listo

```
target -p
target -l 1 1
```

Finalmente también se agregó una linea la archivo rc.6 (que se ejecuta al apagarse el servidor) con el siguiente comando

```
target -a 0 0
```

Para que cuando se apague el sistema quede alineado con la caja.

6.5.3. Configuración de apache + webmin

Se debió configurar apache para que cada vez que se accediera al directorio configurar este hiciera un Proxy inverso hacia el servidor de webmin que escucha en el puerto 10000, esto porque en algunas redes el puerto 10000 y otros son bloqueados. Para esto se realizaron los siguientes cambios en el archivo de configuración de apache:

```
ProxyPass /webmin/ http://localhost:10000/
ProxyPassReverse /webmin/ http://localhost:10000/
<Proxy *>
allow from all
</Proxy>
```

Previo instalación del módulo mod_proxy de apache.

Además se agregaron las siguientes líneas en */etc/webmin/config* :

```
webprefix=/configurar
webprefixnoredir=1
```


7. Plan económico del proyecto

Estimativo del costo se encuentra en la tabla 7-1

Artículo	Costo Total (USD)	Costo Real (USD)
Fibra Óptica	USD 500,00	USD -
Espectrómetro	USD 2.000,00	USD -
Cámara Web	USD 50,00	USD 50,00
Encoders	USD 110,00	USD 110,00
Mecánica	USD 200,00	USD 200,00
Microprocesador	USD 25,00	USD -
Electrónica	USD 100,00	USD 100,00
Trípode	USD 900,00	USD -
Servidor web	USD 400,00	USD -
PDA	USD 300,00	USD -
Acces Point	USD 120,00	USD 120,00
Telescopio	USD 160,00	USD -
Total:	USD 4.865,00	USD 580,00

Tabla 7-1

En la tabla precedente, se observa que el estimativo de costos del proyecto ascendía a la suma de USD 4.865,00 . Dado que se pudo contar con varios componentes el mismo disminuyó a un costo de USD 580,00.

A continuación se presentan, en la tabla 7-2, los gastos reales, desglosados por ítem, en moneda nacional y en dólares americanos. Se separaron los sub totales, los impuestos y el total de cada ítem, para una mejor comprensión de los gastos.

GASTOS EN \$ Y EN U\$S- PROYECTO ITDOAS

FECHA	DESCRIPCIÓN	PROVEEDOR	Precio sin IVA	IVA 23%o 22%	Precio con IVA	Cotización	Precio sin IVA	IVA 23%o 22%	Precio con IVA
09-11-06	Varilla de Acero	Tasco	\$U 50,52	\$U 11,62	\$U 62,14	\$U 23,91	USD 2,11	USD 0,49	USD 2,60
09-11-06	Barra de bronce	BEA Metales	\$U 446,00	\$U 102,58	\$U 548,58	\$U 23,91	USD 18,65	USD 4,29	USD 22,94
11-12-06	Rulmanes	Larique	\$U 612,00	\$U 140,76	\$U 752,76	\$U 24,42	USD 25,06	USD 5,76	USD 30,83
11-12-06	Acero universal Rectificado	Tasco	\$U 88,00	\$U 20,24	\$U 108,24	\$U 24,42	USD 3,62	USD 0,83	USD 4,43
11-12-06	Prisioneros	Bulytor	\$U 115,28	\$U 26,51	\$U 141,79	\$U 24,42	USD 4,72	USD 1,09	USD 5,81
11-12-06	Chapa de aluminio	Abasur	\$U 960,35	\$U 220,88	\$U 1.181,23	\$U 24,42	USD 39,33	USD 9,05	USD 48,37
11-12-06	Tubo rect de aluminio	Abasur	\$U 129,22	\$U 29,72	\$U 158,94	\$U 24,42	USD 5,29	USD 1,22	USD 6,51
12-12-06	Sifnines y coronas	Enbecar	\$U 2.400,00	\$U 552,00	\$U 2.952,00	\$U 24,42	USD 98,28	USD 22,60	USD 120,88
19-04-07	Tornillos y tuercas	Miland	\$U 123,00	\$U 28,29	\$U 151,29	\$U 24,05	USD 5,11	USD 1,18	USD 6,29
19-04-07	Perfil de aluminio	Aluminios del Uruguay	\$U 372,89	\$U 85,76	\$U 458,65	\$U 24,05	USD 15,50	USD 3,57	USD 19,07
19-04-07	Barra redonda	Aluminios del Uruguay	\$U 190,00	\$U 43,70	\$U 233,70	\$U 24,05	USD 7,90	USD 1,82	USD 9,72
22-06-07	Barra de Aluminio	Abasur	\$U 239,00	\$U 54,97	\$U 293,97	\$U 23,92	USD 9,99	USD 2,30	USD 12,29
10-07-07	Oring	Cardelino	\$U 36,30	\$U 7,99	\$U 44,29	\$U 23,92	USD 1,52	USD 0,33	USD 1,85
09-10-07	Collarines	Fivisa	\$U 117,13	\$U 25,77	\$U 142,90	\$U 22,25	USD 5,26	USD 1,16	USD 6,42
MECÁNICA			\$U 5.879,68	\$U 1.350,78	\$U 7.230,48		USD 242,34	USD 55,67	USD 298,01

FECHA	DESCRIPCIÓN	PROVEEDOR	Precio sin IVA	IVA 23%o 22%	Precio con IVA	Cotización	Precio sin IVA	IVA 23%o 22%	Precio con IVA
10-08-06	Tip 41C	Eneka	\$U 66,00	\$U 15,18	\$U 81,18	\$U 23,97	USD 2,75	USD 0,63	USD 3,39
22-06-07	Chapa laminada	Eneka	\$U 64,00	\$U 14,72	\$U 78,72	\$U 23,92	USD 2,68	USD 0,62	USD 3,29
22-06-07	Tip 41C	Fabiet y Bertoni	\$U 94,00	\$U 21,62	\$U 115,62	\$U 23,92	USD 3,93	USD 0,90	USD 4,83
22-06-07	Capacitores	Fabiet y Bertoni	\$U 17,00	\$U 3,91	\$U 20,91	\$U 23,92	USD 0,71	USD 0,16	USD 0,87
22-06-07	Conectores DBq	Elimar	\$U 55,00	\$U 12,65	\$U 67,65	\$U 23,92	USD 2,30	USD 0,53	USD 2,83
22-06-07	Papel para circuitos	Eneka	\$U 131,00	\$U 30,13	\$U 161,13	\$U 23,92	USD 5,48	USD 1,26	USD 6,74
22-06-07	Transistores	Eneka	\$U 12,00	\$U 2,76	\$U 14,76	\$U 23,92	USD 0,50	USD 0,12	USD 0,62
25-06-07	crystal de 20mhz	Mundo Eléctrico	\$U 38,00	\$U 8,74	\$U 46,74	\$U 24,02	USD 1,58	USD 0,36	USD 1,95
25-06-07	Resistencias	Eneka	\$U 39,20	\$U 9,02	\$U 48,22	\$U 24,02	USD 1,63	USD 0,38	USD 2,01
25-06-07	Conector modular	Eneka	\$U 149,32	\$U 34,34	\$U 183,66	\$U 24,02	USD 6,22	USD 1,43	USD 7,65
25-06-07	Capacitores	Eneka	\$U 34,64	\$U 7,97	\$U 42,61	\$U 24,02	USD 1,44	USD 0,33	USD 1,77
10-08-07	Conectores	Eneka	\$U 289,40	\$U 63,67	\$U 353,07	\$U 23,51	USD 12,31	USD 2,71	USD 15,02
15-08-07	Pulsadores	Eneka	\$U 6,82	\$U 1,50	\$U 8,32	\$U 23,65	USD 0,29	USD 0,06	USD 0,35
15-08-07	Reg 7805	Eneka	\$U 11,00	\$U 2,42	\$U 13,42	\$U 23,65	USD 0,47	USD 0,10	USD 0,57
15-08-07	Conectores	Eneka	\$U 35,02	\$U 7,70	\$U 42,72	\$U 23,65	USD 1,48	USD 0,33	USD 1,81
16-08-07	Conectores	Eneka	\$U 33,99	\$U 7,48	\$U 41,47	\$U 23,71	USD 1,43	USD 0,32	USD 1,75
16-08-07	Conectores mini	Eneka	\$U 31,86	\$U 7,01	\$U 38,87	\$U 23,71	USD 1,34	USD 0,30	USD 1,64
16-08-07	Leds	Eneka	\$U 25,70	\$U 5,65	\$U 31,35	\$U 23,71	USD 1,08	USD 0,24	USD 1,32
16-08-07	Cable	Korman	\$U 34,32	\$U 7,55	\$U 41,87	\$U 23,71	USD 1,45	USD 0,32	USD 1,77
30-08-07	Cable USB	La Iguana Hard	\$U 46,20	\$U 10,16	\$U 56,36	\$U 23,65	USD 1,95	USD 0,43	USD 2,38
10-09-07	Placa doble capa	Escar	\$U 25,64	\$U 5,64	\$U 31,28	\$U 23,11	USD 1,11	USD 0,24	USD 1,35
ELECTRÓNICA			\$U 1.240,11	\$U 279,83	\$U 1.519,94		USD 52,14	USD 11,76	USD 63,90

FECHA	DESCRIPCIÓN	PROVEEDOR	Precio sin IVA	IVA 23%o 22%	Precio con IVA	Cotización	Precio sin IVA	IVA 23%o 22%	Precio con IVA
25-06-07	Prensacable	Fivisa	\$U 23,84	\$U 5,48	\$U 29,32	\$U 24,02	USD 0,99	USD 0,23	USD 1,22
25-06-07	conectores dim 8 pines	Mundo eléctrico	\$U 160,00	\$U 36,80	\$U 196,80	\$U 24,02	USD 6,66	USD 1,53	USD 8,19
			\$U 183,84	\$U 42,28	\$U 226,12		USD 7,65	USD 1,78	USD 9,41

FECHA	DESCRIPCIÓN	PROVEEDOR	Precio sin IVA	IVA 23%o 22%	Precio con IVA	Cotización	Precio sin IVA	IVA 23%o 22%	Precio con IVA
11-10-06	Encoders	Digi Key	\$U 2.433,72	\$U 0,00	\$U 2.433,72	\$U 23,86	USD 102,00	USD 0,00	USD 102,00
22-05-07	Cámara Web	Ziber	\$U 409,00	\$U 94,07	\$U 503,07	\$U 24,92	USD 16,41	USD 3,77	USD 20,19
16-08-07	Servidor	MicroCorp	\$U 2.270,00	\$U 499,40	\$U 2.769,40	\$U 23,71	USD 95,74	USD 21,06	USD 116,80
16-08-07	Tarjeta 802.11b	CDR	\$U 331,94	\$U 73,03	\$U 404,97	\$U 23,71	USD 14,00	USD 3,08	USD 17,08
16-08-07	Pendrive 1G	CDR	\$U 343,80	\$U 75,63	\$U 419,43	\$U 23,71	USD 14,50	USD 3,19	USD 17,69
01-11-07	Tripode	Fablet	\$U 720,00	\$U 158,40	\$U 878,40	\$U 21,96	USD 32,79	USD 7,21	USD 40,00
			\$U 6.508,46	\$U 900,53	\$U 7.408,99		USD 275,44	USD 38,32	USD 313,76

SUB-TOTAL	\$U 13.812,10	\$U 577,57
IVA	\$U 2.573,44	\$U 107,51
TOTAL	\$U 16.385,53	\$U 685,09

Como se puede ver en la tabla anterior el gasto total fue de unos USD 685. Es decir se gastaron USD105 de más.

La tabla 7-3 muestra las diferencias entre los distintos ítems

	Presupuestado	Costo Real	Diferencia
Mecánica	USD 200,00	USD 298,01	USD -98,01
Electrónica	USD 100,00	USD 63,90	USD 36,10
Gabinete	USD 0,00	USD 9,41	USD -9,41
Encoders	USD 110,00	USD 102,00	USD 8,00
Cámara Web	USD 50,00	USD 20,19	USD 29,81
Servidor	USD 0,00	USD 116,80	USD -116,80
Access Point	USD 120,00	USD 17,08	USD 102,92
Pendrive 1G	USD 0,00	USD 17,69	USD -17,69
Triode	USD 0,00	USD 40,00	USD -40,00
	USD 580,00	USD 685,09	USD -105,09

Tabla 7-3

En el se puede observar que la parte mecánica costó casi una tercera parte por encima de lo presupuestado y que la electrónica costo solo un 63% de lo pautado.

El proyecto en su totalidad tuvo un costo 15% superior. Pero en realidad esta diferencia del 15% viene dada por la inclusión de dos ítems, que no estaban presupuestados y se adquirieron por decisión del cliente, el servidor y el trípode

Sin este adicional el costo hubiese sido de USD 51,72 menor o sea 9% menor a lo pautado.

8. Gestión de Tiempos

8.1. Plan del proyecto

Cuando comenzamos con el proyecto se propuso el diagrama de Gant de la figura 8-1

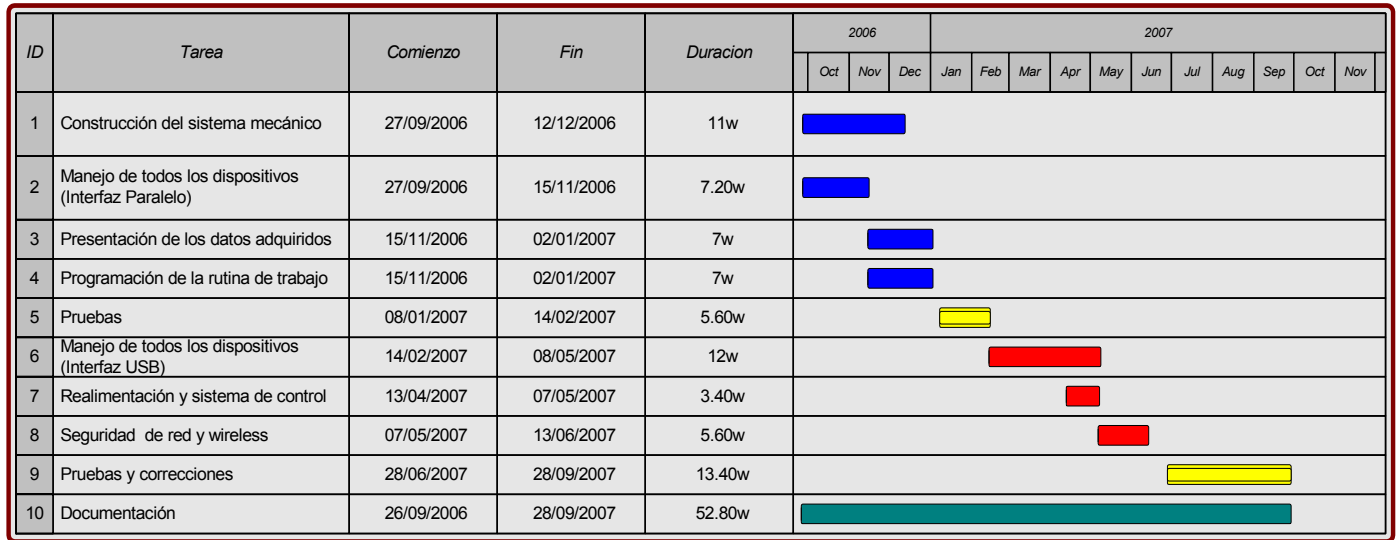


Fig.8-1 Diagrama de Gant proyectado

El plan original del proyecto marcaba dos hitos. El primero, en febrero del 2007, se proponía tener todo el sistema funcionando en loop abierto y con interfaz, de puerto paralelo hacia los motores de paso. Además, haberse completado todo el sistema mecánico y finalizado el diseño. Por ultimo el software para manejar los dispositivos, o incluyendo el driver del espectrómetro y el software para los motores de paso (interfaz LPT) junto con el diseño de la interfaz web, tanto para usuario administrador como para no administrador.

Para el segundo hito, en agosto del 2007, se propuso la operación del sistema en "closed loop" con interfaz USB hacia los dispositivos. Esto incluía la realimentación por parte de los encoders y el programa que envía la posición al dispositivo de control vía USB.

En este esquema se preveían algunos posibles riesgos y un plan de contingencia para estos. Entre ellos el retraso en la fabricación de los engranajes, y el hecho de que no se pudiera contar con un driver para el espectrómetro como el requerido.

8.2. Esquema de tiempos reales.

Al final del proyecto el diagrama seguido fue el siguiente:

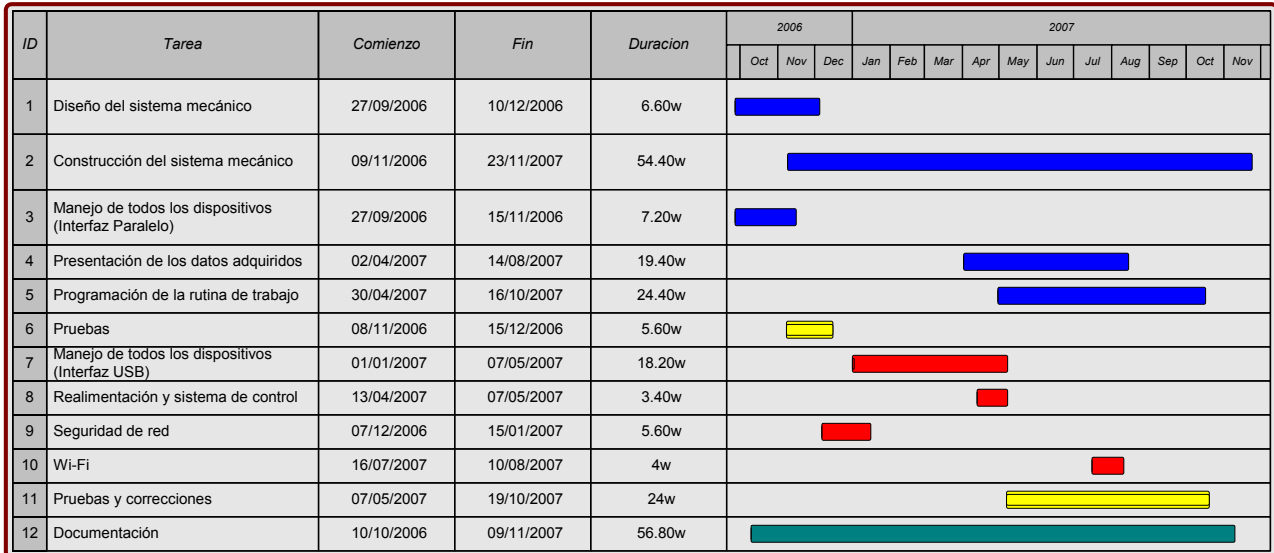


Fig.8-2 Diagrama de Gant real

Como se observa, el mayor retraso se produjo en la construcción de la parte mecánica como se había previsto en los riesgos. Si bien para los primeros días de noviembre del 2006, se contaba con los diseños preliminares, hubo algunos retrasos en su fabricación como se detalla a continuación.

En septiembre de 2006, se pidió presupuesto para la construcción de las cuatro coronas y los cuatro sinfines utilizados en el sistema mecánico, se esperaba disponer en un plazo de aproximadamente dos semanas de una lista de materiales, la misma se retrasó dos meses. Luego de la entrega de materiales se retrasó aún más la fabricación de los engranajes. A todo esto se sumó otro retraso al no tenerse en cuenta la licencia del mecánico del Instituto de Física, para marzo del 2007 sólo se contaba con las dos cajas de reducciones sin el armazón del dispositivo.

Para hacer las primeras pruebas y ensayos sobre los motores hubo que implementarse dos sistemas mecánicos sustitutos; el primero modelaba la masa y el torque del dispositivo final, con una resolución bastante menor a la que realmente se iba a usar, usado para probar el torque de los motores; el segundo, modelaba el movimiento del sistema con dos grados de libertad y se utilizó para estudiar la presentación de la interfaz web, adquiriendo imágenes del movimiento desde una cámara web de prueba adosada al mismo.

Finalmente la construcción de la U que soporta la estructura principal, se terminó de construir a fines de abril de 2007, quedando por delante la unión de todos los componentes mecánicos y de hardware junto con el acabado final.

Como forma de compensar estos atrasos se decidió adelantar, en otros aspectos como por ejemplo la seguridad de red.

A la fecha del primer hito, se contaba, como lo habíamos prometido, con el manejo de los motores en loop abierto, desde la interfaz web, cumpliendo con la entrega de la presentación de datos adquiridos. No se pudo cumplir con el manejo del espectrómetro, dado que el fabricante no nos pudo proveer de un driver acorde a nuestros requerimientos, y en la Universidad de Heidelberg no contaban con este driver como habíamos previsto en los riesgos. Se realizó una investigación sobre el funcionamiento del dispositivo, así como también todo lo referido a la programación USB, para poder implementar un driver para éste. Teniendo como gran contra que el fabricante no nos proporcionó ayuda alguna (debido a que el driver propietario de OceanOptics se vende por USD 500). Esto implicó un aplazamiento en la parte de programación de las rutinas de trabajo ya que todo el potencial en software se invirtió en esto que era la columna vertebral del proyecto.

Para el segundo hito, ya se contaba con el manejo total de los dispositivos con interfaz USB, lo que incluía la placa de control con desarrollo propio en base a un PIC USB y del sistema de realimentación proporcionado por los encoders, el desarrollo de esta tarea se llevó a término de acuerdo a lo previsto, quedando solo la implementación de la placa final.

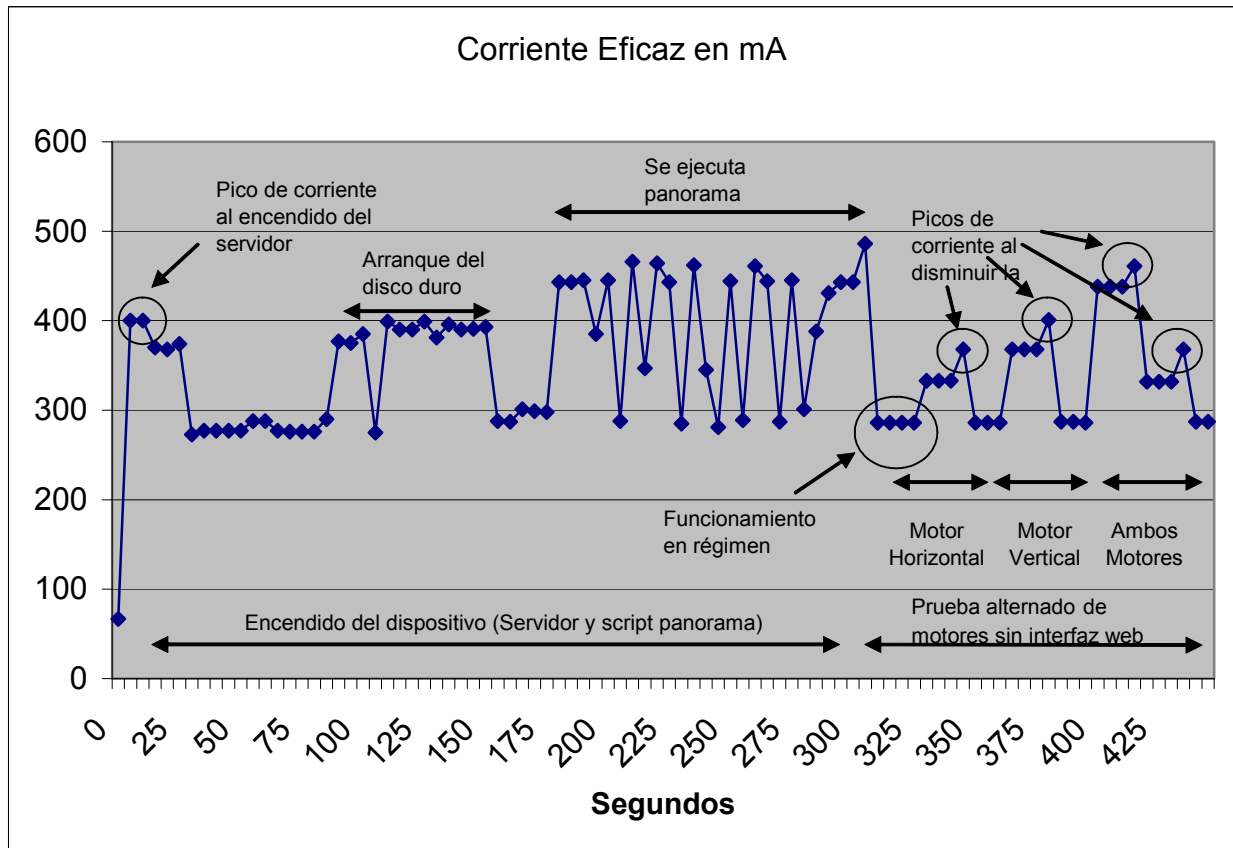
Debido a que la parte mecánica resultó ser más costosa de lo esperado, se decidió reducir los costos en la compra de la tarjeta Wi-Fi y de la cámara web lo que se terminó pagando con más horas de configuración. Esto implicó rediseñar todo el sistema de adquisición y presentación de imágenes.

Desde el segundo hito en adelante se procuró terminar con la parte mecánica, y la programación de tareas, dándole al proyecto un acabo mas profesional. Durante este tiempo se realizaron pruebas internas dentro del laboratorio, y se fueron mejorando aspectos de la interfaz de Usuario Administrador.

Dado todo lo expresado anteriormente, durante las fases de pruebas, se fueron agregando requerimientos extra, para complementar y profesionalizar el desarrollo, por lo cual se decidió solicitar un período de prórroga de la fecha de entrega de dos meses. Esto dio como resultado final que el producto que se entregó, supere las expectativas del proyecto.

9. Consumo eléctrico

Conectando una pinza amperimétrica al cable de alimentación del servidor obtuvimos los datos que se grafican a continuación. Se distinguen tres zonas, arranque del dispositivo, régimen de funcionamiento y movimiento de los motores.



El Servidor apagado se tiene un consumo de energía de 15w por trabajar en “stand by”.

Al encender el dispositivo se aprecia un pico de corriente de unos 400mA, continuando en un régimen de 278mA hasta que se da el arranque del disco duro del servidor. Luego de ejecutarse el script Panorama, en el cual se empieza a ver el consumo del motor horizontal, se observa el consumo en régimen de funcionamiento, es decir, tomando espectros en algún target. Para este valor de corriente se deduce un consumo de potencia de unos 110w.

Luego se prueba la incidencia de los motores, observando distintos consumos según el motor, lo cual era de esperar, dada las diferencias en sus resistencias de bobinado.

Cuando los motores disminuyen su velocidad, como consecuencia de que el dispositivo está próximo al target, se observan aumentos en la corriente siendo esta de un valor máximo de 506mA, con ambos motores en movimiento. Esto representa un consumo máximo de unos 116w. Con lo que el consumo máximo impuesto por los motores es de 51.5w

En conclusión el área de trabajo donde se utilizará deberá contar con una fuente de alimentación alterna de 230v - 135w

Si se utilizara una PC portátil como servidor y baterías para alimentar los motores, solo seria necesario tener un banco de 24v y que pueda soportar un consumo máximo 52W no permanentemente, durante todo el tiempo de trabajo.

10. Conclusiones

Mediante este proyecto pretendimos construir un prototipo, funcional y modularizable el cual ha sido diseñado de forma tal de que a medida que sea necesario, los posibles sucesores tengan la facilidad de ampliarlo, remplazando hardware o implementando nuevo software.

El mismo cuenta con interfaces claras y límites precisos entre los distintos componentes integrando estándares conocidos para su mejor desarrollo.

Se intentó por sobre todas las cosas reutilizar software libre, intentando siempre “no reinventar la rueda”

Creemos que ITDoas va a ser muy útil para el cliente satisfaciendo todas las necesidades planteadas desde un principio.

11. Trabajo a futuro

Una de las posibles mejoras a realizarle al proyecto, y que por falta de dinero no se le pudo hacer, es proporcionarle un servidor de menor tamaño y consumo, o una interfaz en software para controlarlo directamente desde un dispositivo PDA, esto pensando en mediciones de campo, donde factores como el tamaño y el consumo importan mas. Es por esto que una de las cosas en las que más énfasis se puso a la hora del diseño de la comunicación con la cabecera fue que el protocolo utilizado por la placa de control fuera simple y sencillo.

Otra implementación que sería posible de efectuar es la programación de un módulo que funcionara con el software propietario Doasis como utilizan dispositivos similares.

También estuvo contemplado a futuro el cambio del espectrómetro, para lo cual ya existe una interfaz a tales efectos, como se puede ver en el capítulo de modularización

Apéndice I – Manual del usuario

INDICE

1. Pre-configuración.
2. Montaje del ITDoas.
3. Interfaz de Usuario no Administrador.
4. Interfaz de Usuario Administrador.
5. Acceso desde dispositivos móviles.
6. Desarmado del ITDoas.

Pre-configuración

Usted puede configurar los dispositivos de red de ITDoas mediante el uso de un dispositivo USB como un pendrive. El método consiste en editar o crear un archivo y ponerlo en el directorio itdoas/conf del mismo.

Ejemplo: E:/itdoas/conf

Este archivo de configuración puede ser modificado en programas como el block de notas (aunque se recomienda utilizar el editor Geany incluido en el CD)

A continuación se observa un ejemplo de archivo de configuración que es el que trae ITDoas por defecto.

```
#!/bin/bash
ifconfig eth0 164.73.43.89 netmask 255.255.255.128
route add default gw 164.73.43.1
echo "nameserver 164.73.32.2" >/etc/resolv.conf
echo "nameserver 164.73.32.4" >>/etc/resolv.conf
iwconfig wlan0 mode Ad-Hoc channel 11 essid ITDoas
ifconfig wlan0 192.168.0.2 netmask 255.255.255.0
```

Con este archivo ITDoas queda configurado de la siguiente forma

Red Ethernet	
IP	164.73.43.89
Mascara de red	255.255.255.128
Puerta predeterminada	164.73.43.1
Dns1	164.73.32.2
Dns2	164.73.32.4
Red Inalámbrica	
IP	192.168.0.2
Mascara de red	255.255.255.0
Modo	Ad-Hoc
Channel	11
Essid	ITDoas
Web	Libre

Otra forma mas sencilla de lograr el mismo resultado es crear un archivo de configuración net.cfg en la raíz del pendrive que para el caso anterior contendría lo siguiente:

```
IP=164.73.43.89
netmask=255.255.255.128
gateway=164.73.43.1
DNS1=164.73.32.2
DNS2=164.73.32.4

W_IP=192.168.0.2
W_mask=255.255.255.0
channel=11
mode=Ad-Hoc
ssid=ITDoas
web=Libre
```

IMPORTANTE

Para conectarse con ITDoas mediante esta configuración, usted debe estar en una red ethernet que pertenezca al rango especificado o configurar su interfaces inalámbrica para trabajar con una IP del rango 192.168.0.3 mascara 255.255.255.0.

IMPORTANTE

Recuerde que tanto el archivo de configuración como el script son interpretados por Linux, por lo que son sensibles a minúsculas y mayúsculas, por lo que se debe tener cuidado con estas.

Montaje de ITDoas

Para instalar el dispositivo siga estos pasos

1. Arme el trípode y póngalo en el lugar elegido.
2. Monte la estrella sujetadora ente las patas del mismo.
3. Nivele la platina del trípode con una burbuja, corrigiendo la altura de las patas.
4. Saque de la cabecera la arandela y la mariposa.
5. Apoye y ponga en su lugar la cabecera apuntando hacia el principal lugar de medida.
6. Ponga la arandela y ajuste la mariposa.
7. Ubique el servidor bajo el trípode.
8. Conecte la cabecera al servidor mediante el cable USB y el de poder.
9. Conecte si se encuentra disponible el cable de red.
10. Coloque el pendrive (opcional).
11. Conecte a una fuente de 230v 50 Hz el servidor.
12. Encienda el servidor.
13. Espere mientras ITDoas toma la fotos.
14. Cuando la luz verde deja de parpadear y queda prendida el sistema ésta pronto para trabajar.

Mientras la luz verde parpadea el dispositivo ya tiene configurada la red y se puede acceder a el mismo.

Interfaz de Usuario no Administrador

Para ingresar a la interfaz de usuario no administrador, ingrese la dirección IP del dispositivo, desde cualquier navegador web.

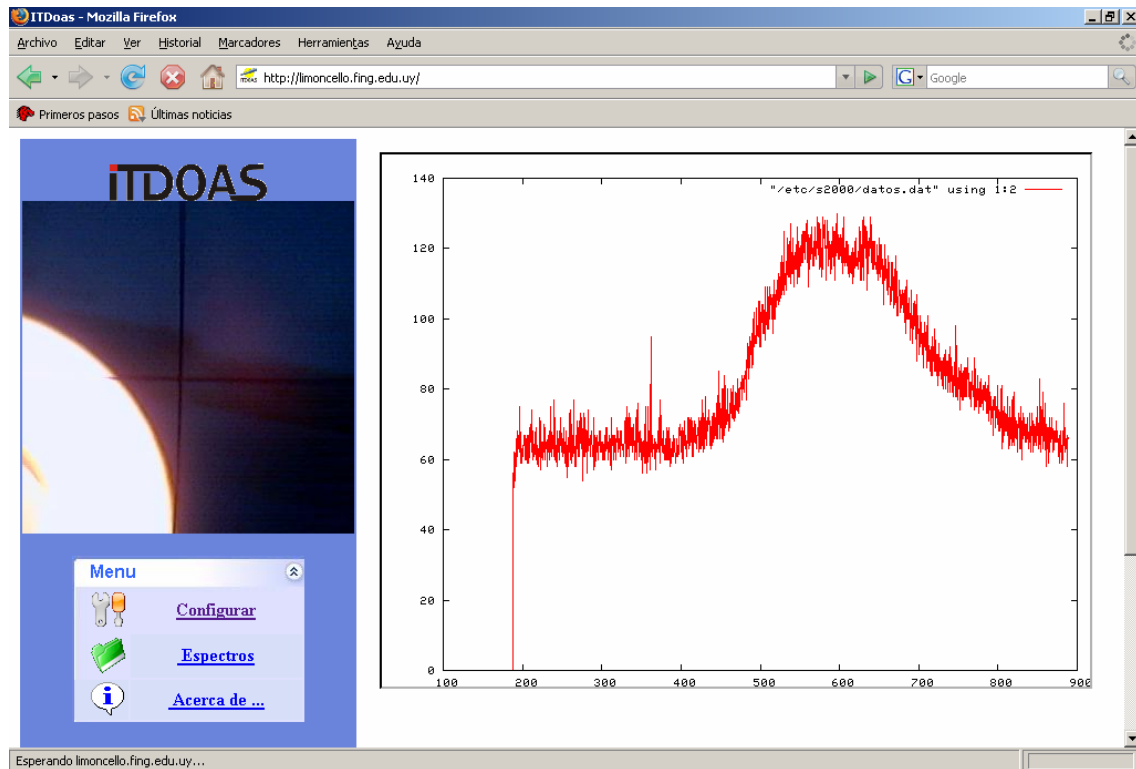


Fig. I-1 Interfaz de Usuario no Administrador

Esta interfaz posee un pequeño menú con las siguientes opciones:

- **Configurar** –Redirige al navegador hacia la interfaz de usuario administrador (ver punto 2)
- **Espectros** –Carga la página desde la cual se puede acceder a los espectros guardados y descargarlos para su posterior análisis. Para descargar un espectro haga clic con el botón derecho del Mouse y pulse *guardar enlace como...* Este directorio cuenta también con un archivo .zip con todos los archivos guardados durante una rutina de script.
- **Acerca de...**–Pantalla que explica de que se trata el ITDoas

Interfaz de Usuario Administrador

Para ingresar a la interfaz de usuario administrador pulse sobre el link configurar el al interfaz de usuario no administrador e inmediatamente se le pedirá un nombre de usuario y contraseña. Luego de identificarse diríjase a configurar Itdoas.

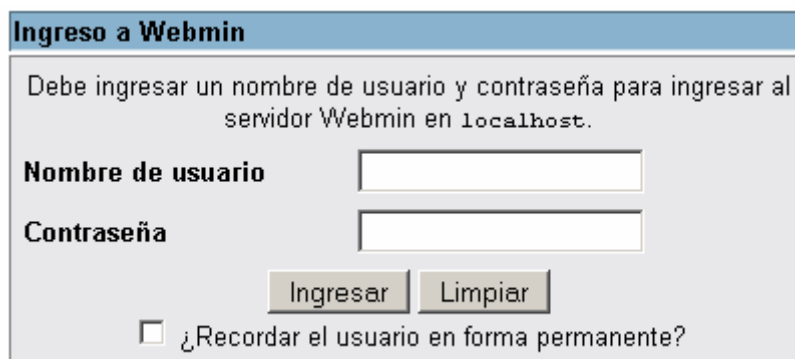
The image shows a web form titled "Ingreso a Webmin" in a blue header bar. Below the title, a message states: "Debe ingresar un nombre de usuario y contraseña para ingresar al servidor Webmin en localhost." There are two input fields: "Nombre de usuario" and "Contraseña". Below these fields are two buttons: "Ingresar" and "Limpiar". At the bottom, there is a checkbox labeled "¿Recordar el usuario en forma permanente?".

Fig.I-2 Login para ingresar a la interfaz de Usuario Administrador

Para una mayor comprensión de la interfaz se dividirá según las siguientes áreas

- a) Tira vista panorámica
- b) Área de control
- c) Tabla
- d) Editor de Scripts
- e) Panel lateral

a) Tira de vista Panorámica

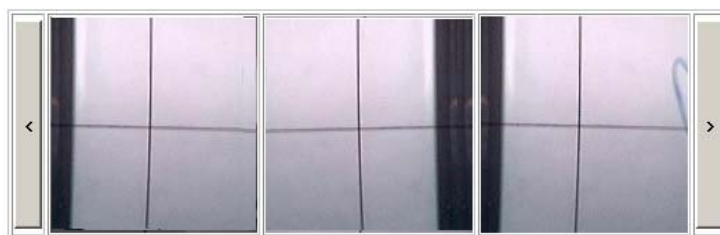


Fig. I-3 Tira de vista panorámica

Permiten dirigir al dispositivo a la posición indicada en la fotografía seleccionada. Para ello se hace clic sobre la imagen deseada y el objetivo automáticamente se redirigirá a esa posición.

Los botones en los extremos permiten ir avanzando la vista panorámica para ver mas posiciones.

b) Área de control

Es el área donde se pueden controlar los movimientos de los motores hasta el más mínimo detalle.

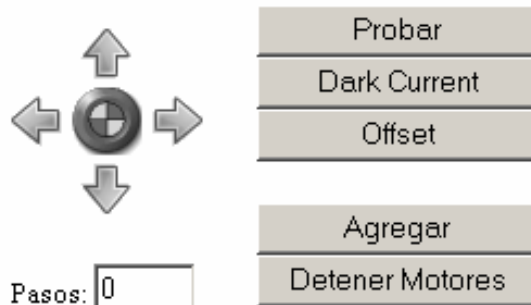


Fig.I-4 Área de Control

Mover

Para mover el dispositivo en una determinada dirección se debe introducir la cantidad de pasos en el cuadro inferior y seleccionar la dirección haciendo clic con el ratón sobre una de las flechas de dirección.

Detener Motores

Detiene el movimiento de los motores anticipadamente (antes de llegar a la posición requerida).

Probar

Pulse probar si lo que desea es adquirir un espectro del lugar donde esta observando, en este caso el programa le solicitará un tiempo de integración y adquirirá un espectro mostrando la grafica en pantalla así como información sobre este (ver figura I-3.4) como tiempo de integración usado, promedio, máximo y mínimo, además de donde ocurrieron.

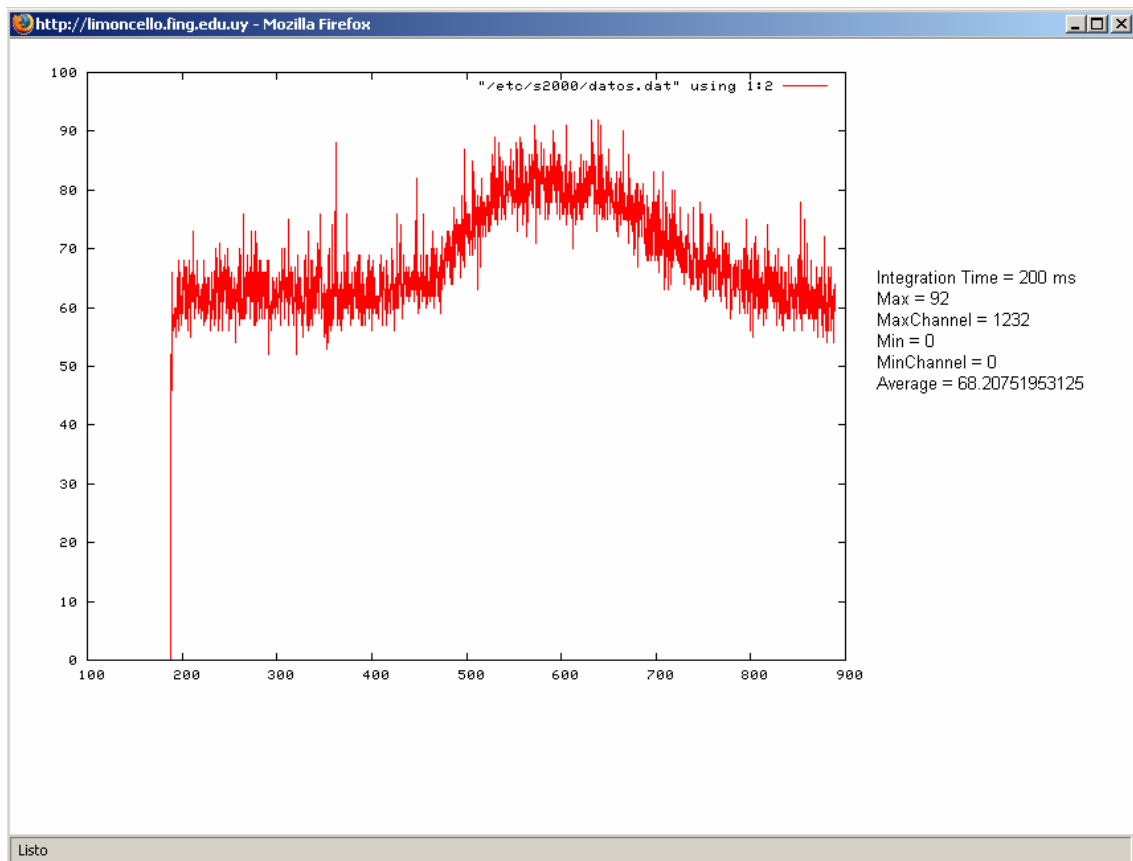


Fig. I-5- Ventana de espectro junto con información del mismo

Dark Current

Pulse sobre el botón dark current para obtener un espectro de oscuridad, un único espectro con tiempo de integración de sesenta mil milisegundos (se guarda en el directorio /espectros/fecha/hora/dc.std)mostrando la grafica en pantalla así como información sobre éste de forma similar a probar.

Offset

Pulse sobre el botón offset para obtener un espectro de offset, diez mil espectros con tiempo de integración de tres milisegundos (este espectro se almacena en el directorio /espectros/fecha/hora/dc.std)mostrando la grafica en pantalla así como información sobre éste de forma similar a probar.

Agregar

Para agregar el target seleccionado a la tabla para su posterior análisis hay dos maneras, haciendo clic sobre el círculo entre los cursores o simplemente mediante el botón del mismo nombre, en ambos casos el programa le solicitará un nombre (etiqueta para la posición).

c) Tabla

Aquí se guardan los *targets* seleccionados junto con sus coordenadas. La tabla no puede tener valores con la misma etiqueta, ni con las mismas coordenadas, y por defecto el comando que agrega los mismos alertará en caso de que esto se intente hacer.

Eliminar Targets

Para eliminar un *target* se hace clic sobre el cuadro a la izquierda de cada uno y se pulsa el botón Borrar Marcados. Para borrar la tabla y comenzar con una nueva se pulsa el botón Nueva Tabla

Nueva Tabla		Borrar Marcados	
Valores Guardados			
☐	Longitud	Latitud	Etiqueta
☐	11004	9999	posicion1
☐	11004	11002	posicion2

Fig.I-6 Tabla con las posiciones

ⓘ ATENCIÓN

Debido a que el programa de edición de script es sensible a mayúsculas y minúsculas se recomienda usar nombres de etiquetas en minúsculas, sin espacios o símbolos extraños.

d) Editor de script

Área de texto donde se pueden escribir rutinas (scripts) para que el dispositivo efectúe automáticamente. Este editor además reconoce la sintaxis de los comandos coloreándolos, y si se pulsa la tecla tabulador cuando se escribió un comando, este agrega los paréntesis con los lugares para las variables, para facilitar la tarea del usuario.



Fig.I-7 Editor de Script

El editor cuenta también con un área para abrir archivos guardados en la computadora cliente. Para ello simplemente se pulsa sobre el botón examinar, se elige el archivo y luego se pulsa abrir.

Una vez copiado o cargado el archivo del script se pulsa sobre el botón Guardar y ejecutar y se cambia el área del editor por la de monitoreo, donde se detalla la evolución del script. También se puede acceder al área de monitoreo mediante el botón Monitorear.

Listado de comandos y explicación

adquirir(Pos1, N):	Adquiere N espectros promediados en la Pos1, donde N es un número entero y Pos una etiqueta de una posición en la tabla.*
adquirir_s(Pos1, N, T):	Igual que el anterior pero recibe además el tiempo de integración (T)
barrido(Pos1, M, Pos2, N):	El dispositivo se mueve entre Pos1 y Pos2 deteniéndose M veces en el medio para adquirir N espectros promediados en cada parada. (M= Cantidad de paradas) donde N y Pos1 (y Pos2) son de mismo tipo que en adquirir*
barrido_s(Pos1, M, Pos2, N, T):	Igual que el anterior pero recibe además el tiempo de integración (T)
vertical(Pos1, G, N):	El dispositivo toma la coordenada X de la posición Pos1y ahí realiza una rutina vertical entre 0 y 90° parando cada G grados para adquirir N espectros promediados *
vertical_s(Pos1, G, N, T):	Igual que el anterior pero recibe además el tiempo de integración (T) *
noon(N):	Realiza N espectros promediados del zenit
noon_s(N,T):	Igual que el anterior pero recibe además el tiempo de integración (T)
save_picture (arch, 'coment'):	Guarda una fotografía de lo que ve el espectrómetro en la posición donde está con el nombre recibido en <i>arch</i> y le adjunta una etiqueta con el comentario recibido en <i>coment</i> .
agendar(hh:mm,"cmd"):	Agrega un comando recibido en cmd a una lista para ser ejecutado a la hora hh:mm (formato hora:minutos).
scant1tn(Pos1,...,PosN,N)	Toma N scans en Pos1, Pos2,..., PosN. Trabaja en loop. Se detiene con detener comando
elevation(Pos1,angle,cant,N)	Toma N espectros en Pos1 y en cada ascenso de <i>angle</i> grados. Dicho ascenso se realiza <i>cant veces</i> y luego se dirige al zenit. Trabaja en loop. Se detiene con detener comando.

(*) El tiempo de integración se ajusta automáticamente según criterio de valor de pico al 80% del valor de saturación del espectrómetro.

Cada script ejecutado (cada vez que se pulsa guardar y ejecutar) genera una carpeta con su nombre (scriptN°), dentro de la carpeta con la fecha del día accesible desde la página web, con todos los datos mas la rutina ejecutada y un archivo zip con el contenido de la carpeta para facilitar su descarga.

Cada comando, a su vez, crea una carpeta descriptiva de la acción ejecutada, dentro de la carpeta con el nombre del script donde guarda sus datos, un archivo con los datos

en formato Standard (compatible con Doasis), una fotografía de la posición y una imagen de la gráfica del espectro.

i ATENCIÓN

El lenguaje de scripting es sensible a mayúsculas y minúsculas por lo que los comando deben ser todos en minúsculas (se deben colorear una vez escritos) y los nombres de las posiciones deben escribirse exactamente igual a los de la tabla (se pueden copiar y pegar mediante clic derecho del Mouse).

e) Panel Lateral

En este panel se ubica la cámara web y un pequeño menú con las siguientes opciones:

- **Espectros** –Carga la página desde la cual se puede acceder a los espectros guardados (ver interfaz de usuario no administrador).
- **Webcam** –Refresca la cámara web del servidor (la enciende hasta que el dispositivo se mueva nuevamente y se vuelva a apagar)
- **Espectrómetro** –Permite seleccionar un espectrómetro de la lista para trabajar con el (debe estar instalado en el dispositivo)

Acceso desde dispositivos móviles

El dispositivo cuenta con conexión Wi-Fi (inalámbrica) que permite que dispositivos como laptops o PDA se puedan conectar con este sin necesidad de cables de red.

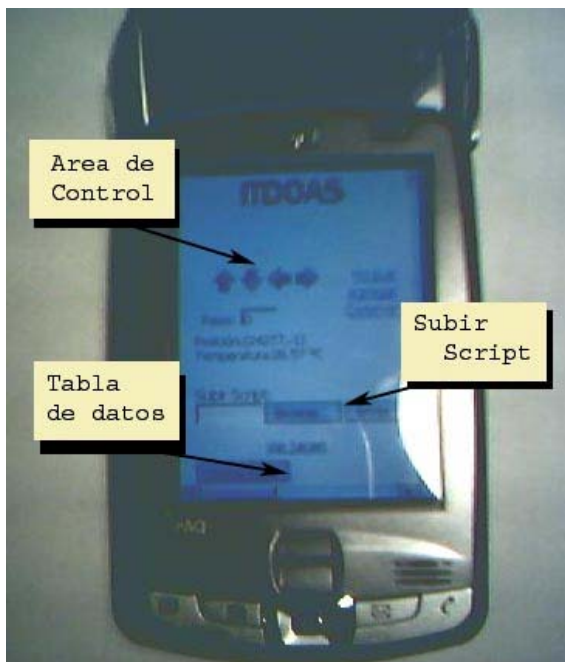


Fig.I-8 Vista de la interfaz en una PDA

Por otra parte el servidor de ITDoas posee una interfaz de baja resolución para dispositivos como PDA que poseen un área de visualización más pequeña (menor a 640x480). Dicha interfaz permite realizar casi todas las mismas operaciones que la interfaz original pero con una interfaz un poco menos amigable.

En la interfaz de usuario Administrador, esta cuenta con una versión adaptada de el área de control y tabla. Mientras que si bien no posee un editor de script como la interfaz normal, ofrece la posibilidad de subir archivos ya editados (al igual que la interfaz normal).

Desarmado del ITDoas

Como desmontar la cabecera del ITDoas.

Paso 1: Posicione la cabecera de forma tal de poder acceder cómodamente a los tornillos que están en la platina del motor vertical. Ese posicionamiento lo debe efectuar moviendo los motores desde la interfaz web

Paso 2: Desconecte el cable de poder y el cable USB de la parte posterior de la cabecera.

Paso 3: Afloje los dos tornillos alen de la platina vertical. La cabecera quedará libre. Tenga la precaución al volver a armarlo que el eje tiene dos muescas, una para cada tornillo.

Paso 4: Retire los dos tornillos tipo alen que se encuentran en la base de la U que soporta el parante que no contiene el motor.

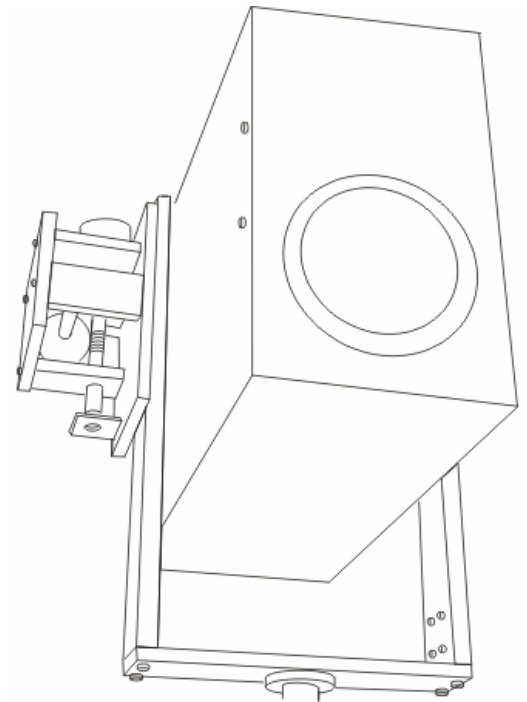


Fig.I-9- Cabecera del ITDoas

i IMPORTANTE

Solo retire los torillos, no realice ningún esfuerzo sobre la cabecera

Paso 5: Tome la cabecera y jale en la dirección del eje. Luego de esto habrá logrado desmontar la cabecera junto con uno de los parantes de la U.

Desmontado del interior de la cabecera.

Paso 1: Retire los seis tornillos de la tapa de la cabecera, tome la manija y remueva la tapa.

Paso 2: Retire las conexiones del HUB USB. A continuación desconecte las conexiones de los indicadores lumínicos y del ventilador

Paso 3: Para remover la tarjeta destornille los tornillos superiores del ventilador. Luego desconecte de la misma el conector USB, los conectores de los motores y el conector de potencia. Retire la placa.

Paso 4: Afloje el prensaestopa que evita jalar de los cables de los motores y retire los mismos.

Paso 5: Retire el conector USB y la fibra óptica del espectrómetro y seguidamente destornille los 4 tornillos que sujetan la base del espectrómetro. Tenga precaución de no romper la fibra óptica. Remueva el espectrómetro de la caja

•

Apéndice II.

Instalación del sistema Linux-ITDoas

El sistema sobre el cual corre el software de servidor de Itdoas esta basado en una distribución especial de Linux llamada Slax. Este nos permite generar en un CD todo el Sistema Operativo y las utilidades necesarias para ejecutar el software de una manera mas sencilla. Al estar basado en Slax, el software del servidor tiene dos formas de ejecutarse, “*Live*” y normal, la diferencia entre ellas se explicará junto con su forma de instalación a lo largo de este apéndice.

Ejecución “*Live*”

En esta forma de ejecución el Sistema Operativo se ejecuta en “ram”, o sea que todos los cambios que se le hagan este se borrarán al apagar el equipo. Esta forma de ejecución no requiere disco duro y se puede ejecutar directamente desde un CD. Para ejecutar el Sistema Operativo Linux-ITDoas de esta forma solo se necesita configurar la computadora donde se lo ejecutará para que arranque (boot) desde el CD.

a) Instalación Live (al disco duro)

Una alternativa a este método (cuando no se desea que el sistema se ejecute desde un CD) es instalar el sistema al disco duro (el sistema igual se ejecutará en formato Live, es decir el sistema no guardará los cambios al apagarse). Esta es la forma en la que viene instalado el sistema por defecto ya que ofrece un a mayor robustez debido a que los archivos del sistema no se corrompen ante un mal reinicio del sistema.

Para instalar el sistema al disco duro se necesita, una partición del disco duro de unos 650 Mbytes de cualquier formato (FAT, EXT2, XT3, etc) y unos 64-128 Mbytes de memoria ram (mínimo, recomendado 256 Mbytes).

1. Para instalar en modo live inserte el CD ITDoas LIVE en la lectora y reinicie la computadora donde va a instalar el software (asegúrese de que este configurado el BIOS de la misma para arrancar desde el CD).



Figura II-1 Icono del instalador

2. Una vez iniciado el Sistema Operativo haga doble clic sobre el icono “Instalar Itdoas Live” sobre el escritorio, como muestra la imagen.

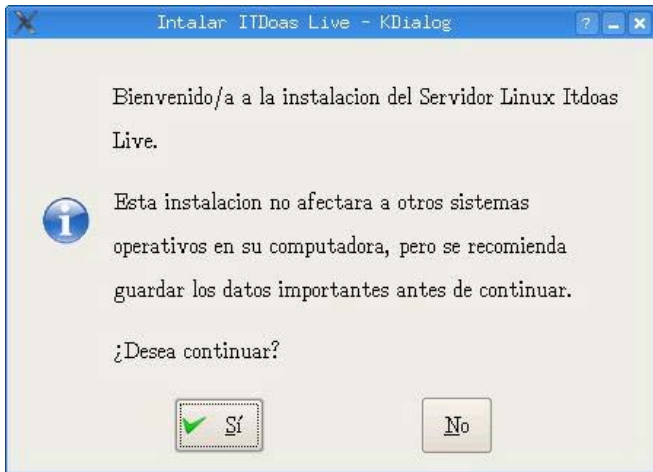


Fig. II-2 Pantalla de bienvenida del instalador

4. En la siguiente ventana se nos preguntará por un disco o partición donde instalar el sistema. En Linux por lo general los discos duros (y otros dispositivos) se “montan” en el directorio /mnt por lo que probablemente el disco donde desea instalar se encuentre ahí. Si su disco es un disco IDE probablemente se encuentre en un directorio llamado hdXN donde X puede ser a para el master y b para el slave del primer IDE (análogamente c y d para el segundo ide) y N un número que indica el nro de partición (1-4 primarias, 5-8 secundarias). Si su disco es SCSI en lugar de hdXN tendra algo de la forma SDXN (también para dispositivos de almacenamiento USB).

3. Luego le aparecerá la siguiente ventana que da la bienvenida al instalador. Debemos presionar Si para continuar con la instalación, si pulsamos No la instalación será cancelada y ningún cambio será realizado al disco duro.

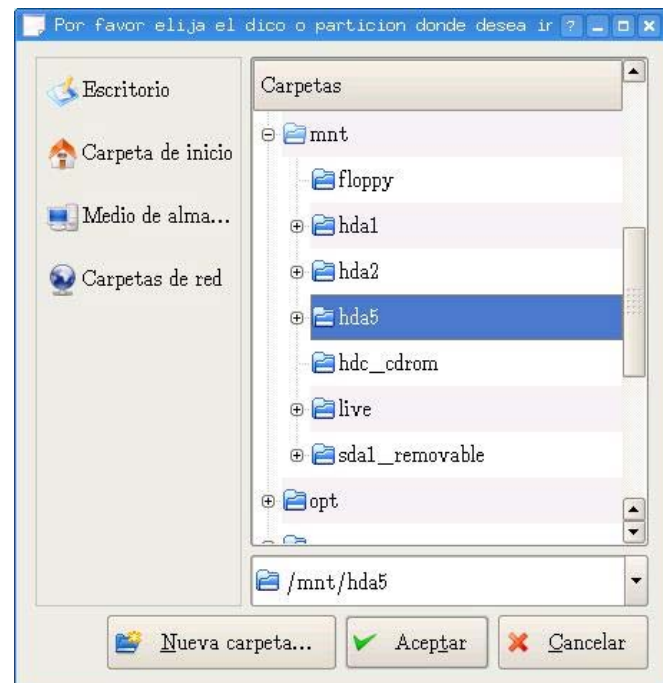


Fig.II-3- Paso 4

5. Una vez aceptada la ventana anterior comenzará la copia de archivos, esta operación puede tardar un tiempo dependiendo de la velocidad de la computadora.

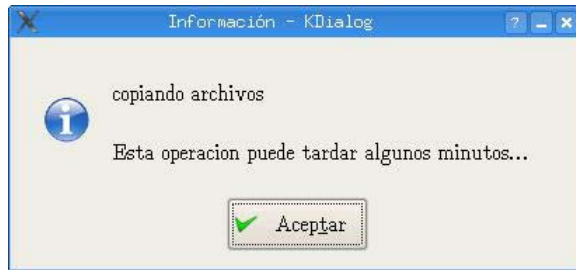


Fig.II-4 Paso 5

6. Cuando finalice la copia de archivos el programa instalará un gestor de arranque en su computadora (LILO Linux LOader)⁷ para que pueda iniciarse la próxima vez y cuando finalice estará listo para utilizar el sistema Live Itdoas directamente desde el disco duro sin necesidad del CD. Solo restará que reinicie la computadora.

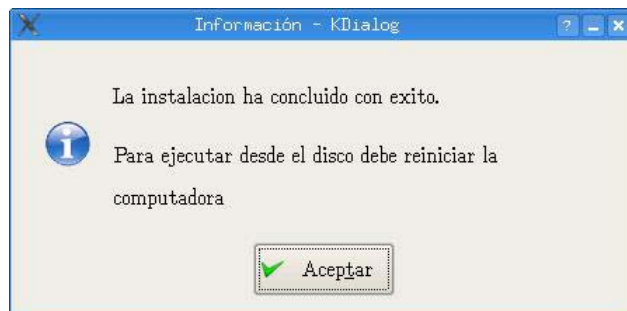


Fig.II-5 Paso 6

⁷ Si tiene otro sistema operativo en el disco tendrá que configurar el arranque a mano para que pueda iniciar la computadora con este.

Ejecución normal

Entendemos, en esta sección, por ejecución normal a cuando el sistema se ejecuta como cualquier otro sistema operativo, es decir desde el disco duro y en su propio sistema rearchivos etc.

Para instalar el sistema de esta forma se necesita, una partición de unos 2 Gbytes con formato Linux (si no la posee el instalador la creará) y la misma cantidad de memoria ram que para el caso anterior. La instalación en este caso es un poco más complicada y requiere un mayor espacio en disco, pero ofrece también mayor eficiencia.

Para esto se incluye una aplicación llamada Slax2hd que viene por defecto en todas las versiones de Slax desde la versión 6.0⁸. Para ejecutarla simplemente haga clic sobre el icono en el escritorio y prosiga las indicaciones en pantalla.

⁸ ITDoas esta basado en la versión 5.8 de Slax, pero igual contiene el instalador ya que fue agregado por nosotros.

Apéndice III

Motores de Paso y Encoders.

Motores de Paso.

Ante el problema de elegir un dispositivo ideal para controlar los movimientos se deja de lado el motor de continua y se opta por el motor de pasos dada la precisión que estos poseen.

Básicamente estos motores constan de un rotor sobre el que se aplican distintos imanes permanentes y por un cierto número de bobinas excitadoras en su estator. Los motores utilizados son el 42SPM-24D8A y el SANYO CP023 , ambos del tipo p-p unipolares y compuestos de 4 bobinados y 5 cables dispuestos como muestra la figura III-1

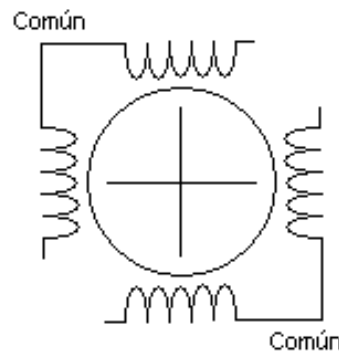


Fig.III-1

La elección de los últimos tiene que ver con el torque que éstos son capaces de desarrollar, característica más que importante dadas las dimensiones del soporte, y son modelos diferentes por ser los únicos motores de 24v existentes en el laboratorio.

Existen tres tipos de secuencias posibles para manejar estos motores: normal, wave drive y medio paso. La primera consiste en activar dos bobinas a la vez, la segunda en activar una sola bobina y la tercera en activar primero dos bobinas, luego solo una y así sucesivamente de modo de avanzar de un paso a la vez.

La secuencia elegida para manejar los motores fue la de tipo wave drive y esta elección se basó en lo siguiente. Por las características del sistema mecánico, una vez posicionado el objetivo desactivamos las bobinas y sabemos con certeza que no se moverá (consecuencias del uso de sinfín y corona en su construcción) y además cual es la posición del último paso (consecuencia de haber desactivado un sola bobina). Esto último no lo podemos garantizar en las otras secuencias ya que al desactivar dos bobinas, el riesgo de hacerlo en distinto tiempo, produce incertidumbre en la posición final.

De acuerdo a la secuencia elegida, para desarrollar la rotación debemos energizar de a una por vez las 4 bobinas en forma cíclica. Lo primero es identificar el cable común y la correspondencia entre los restantes y las 4 bobinas. Para ello, midiendo con un tester las impedancias vistas de a pares de cable, se identifica el cable común como aquel que

da la misma impedancia contra todos los demás. Luego se energizan de a una las bobinas y por ensayo y error se identifica la secuencia correcta para avanzar de a pasos.

Para el 42SPM-24D8A el cable común es el BLANCO y el orden de la secuencia correcta ROJO, AZUL, MARRON, Y AMARILLO. Y para el SANYO CP023 el común es el ROJO y la secuencia correcta NEGRO, NARANJA, MARRON Y AMARILLO

En este punto, lejos de conocer el protocolo USB y la programación y utilización de un microcontrolador, necesitábamos de un driver que nos permitiera realizar las primeras pruebas sobre los motores de paso. Esto nos proporcionaría información sobre el consumo, el torque y la velocidad de los mismos así como un nexo para probar las partes mecánicas que iban saliendo del taller.

a) Primer aproximación al Driver de los motores: Interfaz Puerto Paralelo

Ahora que conocemos en qué orden energizar las bobinas, necesitamos de una placa capaz de manejar las corrientes que deben aplicarse a cada motor. Estos motores requieren de un voltaje de 24v y por sus resistencias de bobinado (24 y 50ohms) esto significa sobrellevar corrientes máximas de 1A. Por ello es necesario bufferear la salida del puerto paralelo y alimentar los motores a través de una fuente externa.

Un esquema de lo que estamos diciendo se muestra a continuación.

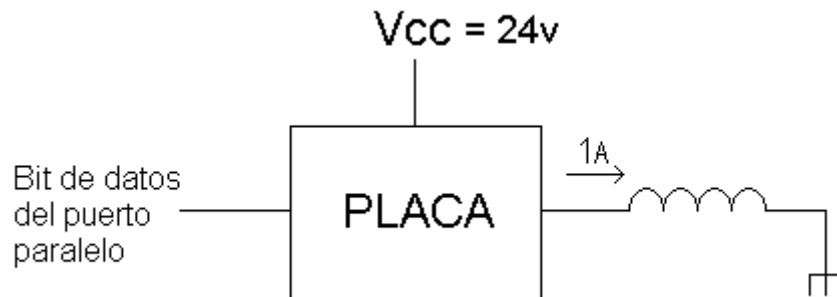


Fig.III-2

Por el alto consumo el motor no debe tomar la corriente de la lógica de activación, sino que esta se encargará de dirigirla o no hacia el motor. La solución es por lo tanto una etapa de potencia trabajando como interruptor (corte y saturación de un transistor), permitiendo así una alta impedancia a la entrada.

El resultado de esto es el circuito que se presenta a continuación en la figura III-3.

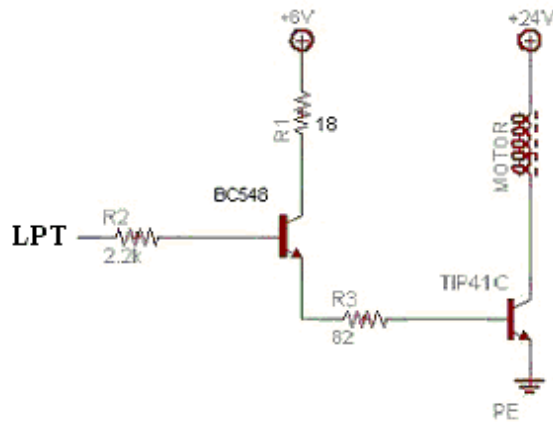


Fig.III-3

En este circuito colocando a “1” un bit del puerto paralelo (aproximadamente 4.0v) el transistor BC548 trabaja en zona activa con

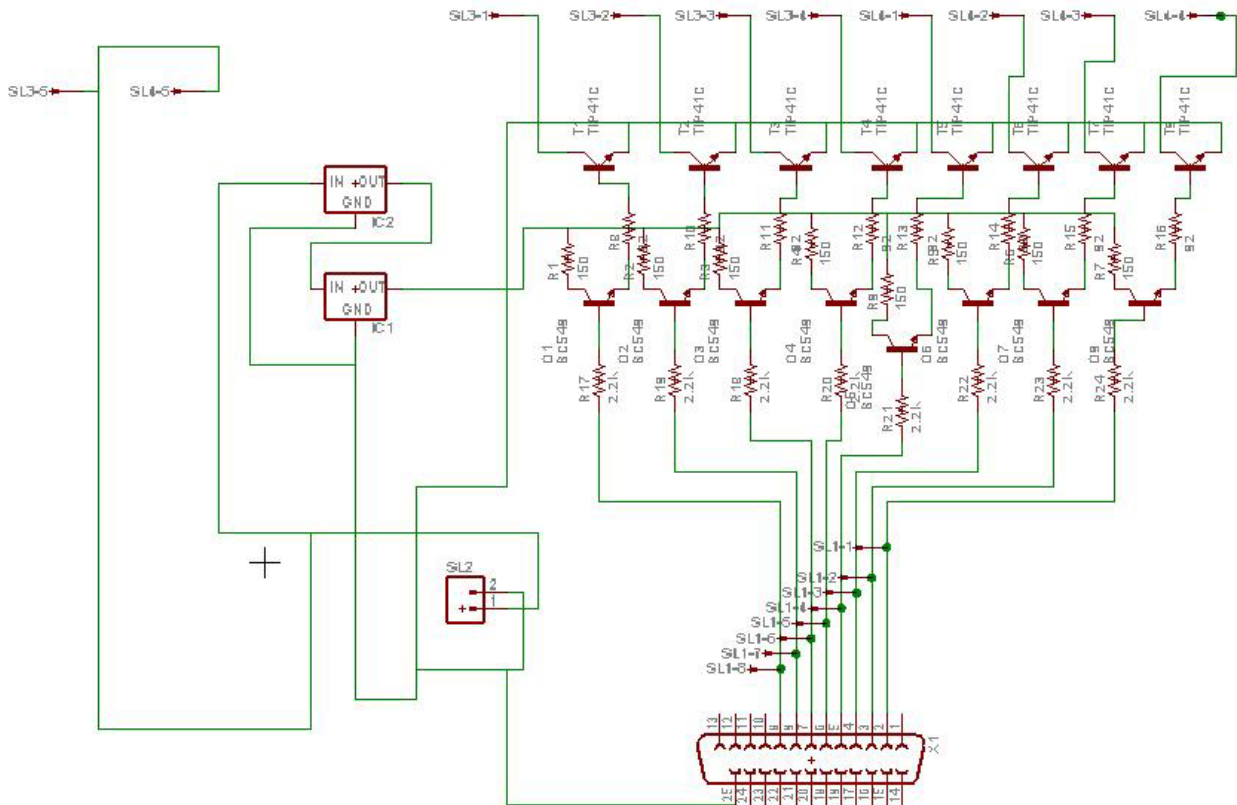
$$i_C = 110 \left(\frac{4.0v - 0.7v - 0.7v}{2.2k\Omega + 110(82\Omega)} \right) \approx 0.025A, \text{ con lo cual el TIP41C tendrá una } i_C \approx 0.97A,$$

que es aproximadamente la corriente de saturación por ser $I_{Csat} = \frac{24v - 0.3v}{24} = 0.98A$.

Entregando de este modo la corriente demandada de 1A.

Ahora, si por el contrario se pone a “0” la entrada (aproximadamente 0v) no habrá corriente por la base de la primera etapa ni de la segunda quedando ambos transistores en zona de corte.

Tanto el esquemático como el diseño final de placa que se muestran a continuación fueron desarrollados sobre Eagle.



V Fig.III-4 Esquemático del driver del puerto paralelo

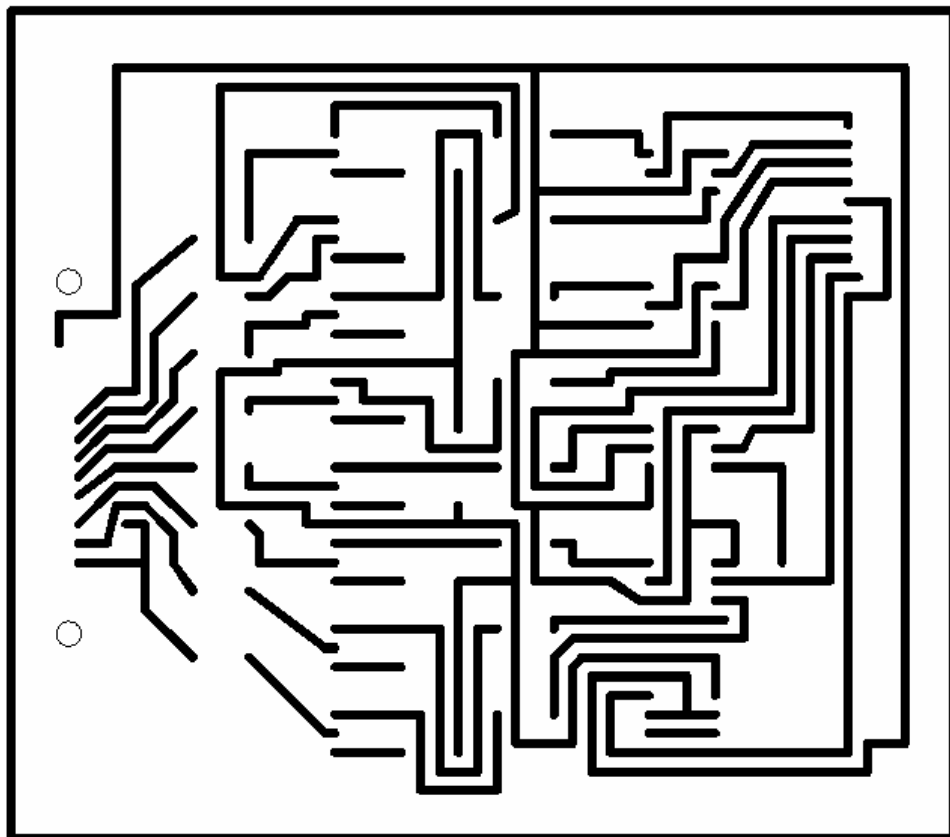


Fig.III-5 Diseño de la placa.

Encoder

Los encoders utilizados son de la marca BOURNS modelo ENA1J-B28-L00064. Estos se componen de un eje y 5 terminales:

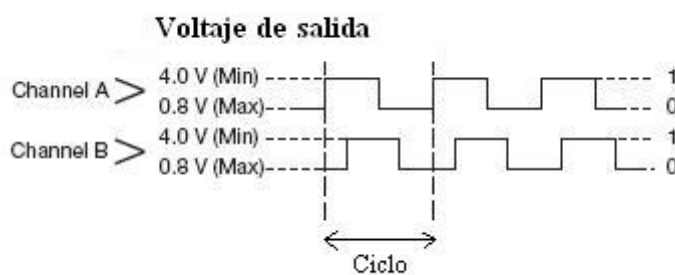
- Positivo.
- Negativo.
- Channel A
- Channel B
- NC



Alimentados con una diferencia de potencial de 5v entre sus terminales positivo y negativo, es posible detectar la dirección del movimiento del eje a partir de las señales provenientes de sus canales A y B.

Estas señales se encuentran en cuadratura, es decir, desfasadas 90 grados entre si y se componen de un nivel alto, con valor mínimo de 4v, y un nivel bajo con valor máximo de 0.8, ambas TTL compatibles. La resolución del dispositivo es de 256 ciclos por revolución.

A continuación se muestra un diagrama con dichas señales.



En nuestro proyecto, el encoder se encuentra acoplado al eje del motor. Sus señales de salida de datos se conectan directamente al microcontrolador, donde son procesadas para detectar la dirección del movimiento. En la figura se observa una foto del mismo como parte del sistema mecánico. En cuanto a su conexión debe tomarse la precaución de enchufar el conector con el cable naranja enfrentado a Channel B.



Fig.III-6 Encoder

Apéndice IV Especificaciones Técnicas

Dimensiones cabecera

Ancho:	18 cm.
Alto:	33.5 cm. ; 59-72 cm. desde la base del trípode
Largo:	39 cm.
Peso:	15 kg (aprox).
Velocidad:	0.167 RPM

Espectrómetro OceanOptics S2000

Resolución óptico:	0.3-10 nm. FWHM
Rango Dinámico:	2×10^8 (system); 2000:1 simple scan
Longitud Focal:	42 mm. (input) 68 mm. (output)
Rango del Detector:	200-1100 nm.

Telescopio Tasco Galaxee

Distancia Focal:	500 mm.
Diámetro Espejo Principal:	114 mm.
Conector de Fibra:	SMA

Otros

Interfaz con el servidor:	USB
Interfaz de red:	RJ-45 ; Wi-Fi (inalámbrica)
Alimentación:	12 VDC y 24 VDC (ambos)
Consumo:	120W
Rango de operación:	0 °C - 55 °C
Precisión de movimiento:	4.69×10^{-3} de grado
Precisión de lectura:	0.87×10^{-3} de grado

Apéndice V - Términos y definiciones

En esta documentación se utilizan los siguientes términos:

ITDoas

Siglas de “Intelligent Target Differential Optical Absorption Spectroscopy” y es el nombre del dispositivo descrito en esta documentación. Dicho dispositivo se descompone en dos módulos descriptos a continuación en esta sección

ITDoas-server

Módulo del sistema ITDoas que se encarga de comunicarse con el resto del los dispositivos y actuar como servidor para el control del ITDoas.

Cabecera-ITDoas

Módulo principal que incluye el espectrómetro, cámara Web, telescopio y placa de control. Es el encargado de recabar los datos.

Doas (Differential Optical Absorption Spectroscopy)

Técnica de observación por medio de espectros.

Espectrómetro

Dispositivo capaz de medir la intensidad de la luz a diferentes longitudes de ondas.

Placa de control

Circuito impreso capaz de comunicarse con el modulo ITDoas-server. Capaz de controlar los movimientos del Modulo-ITDoas , medir la temperatura y activar el ventilador interno entre otras cosas.

Objetivo

Se le llama objetivo (target) al objeto de interés a observar por el dispositivo ITDOAS.

USB (Universal Serial Bus)

Puerto de comunicación serie utilizado en la mayoría de las computadoras modernas que será el reemplazo futuro de los puertos de comunicación actuales (RS-232)

Encoder

Dispositivo capaz de medir variaciones de ángulos y sentidos de giro.

Usuario no Administrador

Usuario con acceso restringido a la página de control de ITDoas, el cual solo puede observar el proceso de medición.

Usuario Administrador

Persona autorizada (mediante sistema de contraseñas) para acceder a los controles de ITDoas, y con permisos para manejarlo completamente.

Apéndice VI Contenido del CD

A continuación se observa una figura ilustrativa de los directorios que componen el CD, y una breve explicación del contenido de los mismos.

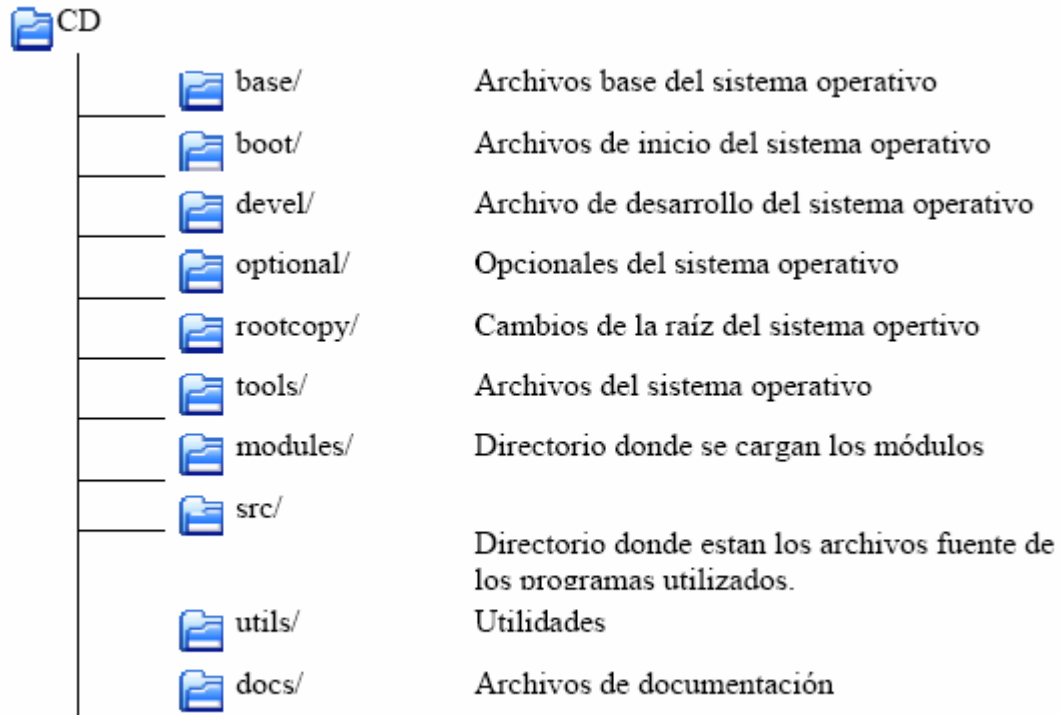


Fig.VI-1 Contenido del CD

Los directorios base, boot, devel, optional, rootcopy y tools son parte del sistema operativo y no se discutirá sobre su contenido, el directorio modules, aunque es parte también del sistema operativo, atrae nuestra atención ya que es donde se incluyen los módulos de este, es decir donde se deben aplicar los archivo de expansión del mismo. El directorio src incluye los códigos fuentes de todos los programas creados, mientras que el directorio utils contiene software libre considerado útil para utilizar con el proyecto (Navegador Mozilla Firefox, editor de textos Geany, etc). Por ultimo en el directorio docs se encuentran el archivo de documentación y manuales de usuario. Los demas archivos en la raíz del CD son del sistema operativo.

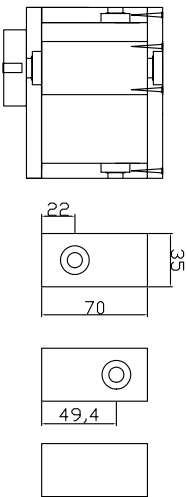
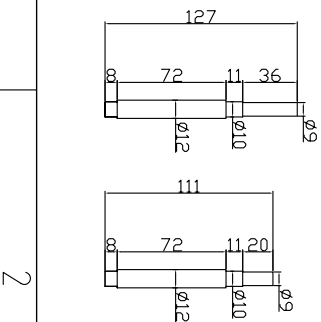
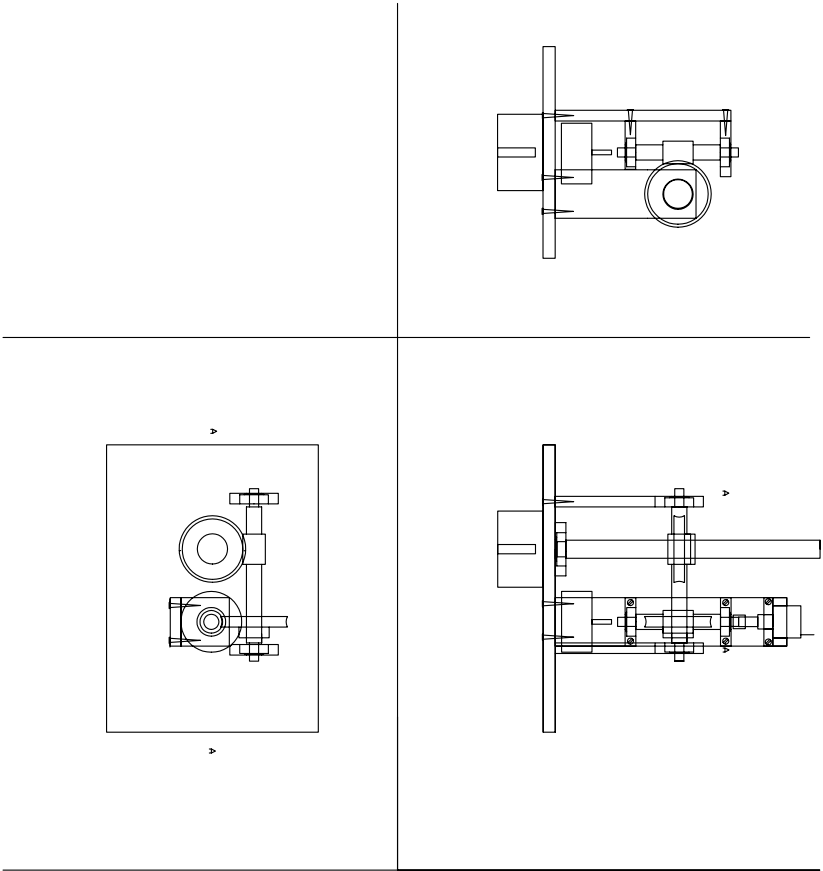
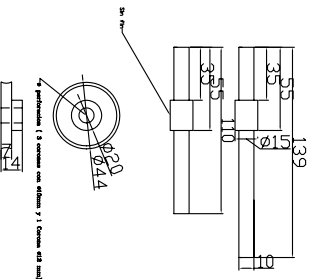
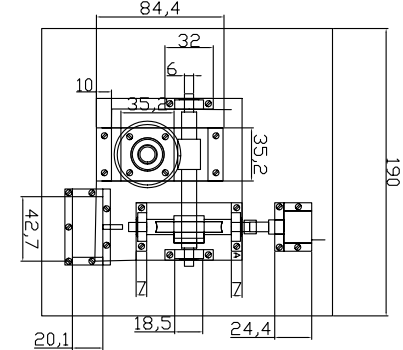
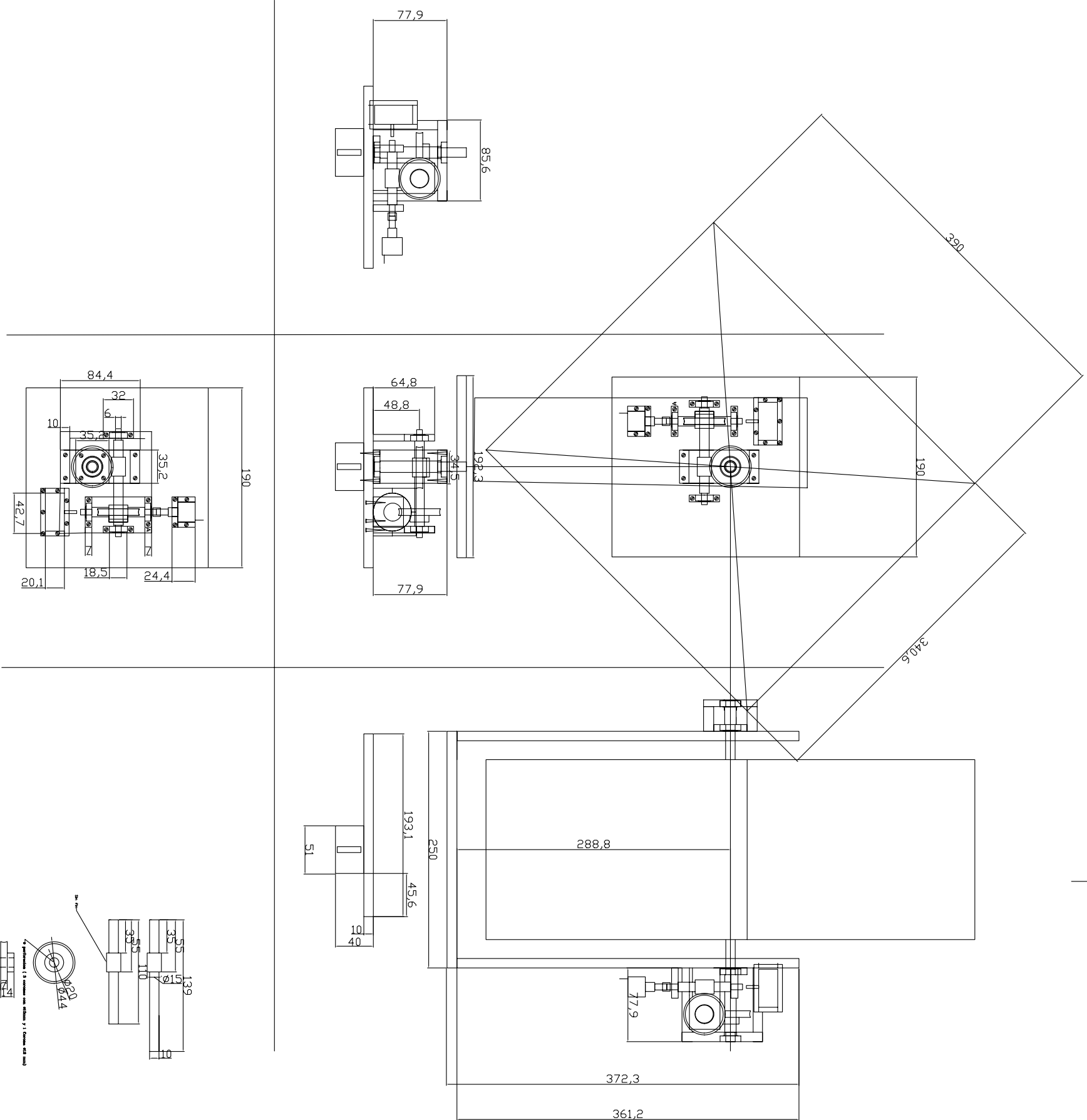
Bibliografía

- [1]. Mini Max-Doas An Introduction
Nicole Bobrowski – Frank Filsinger
University of Heidelberg
- [2]. Tomographic multiaxis-differential optical absorption spectroscopy observations of Sun-illuminated targets: a technique providing well-defined absorption paths in the boundary layer
Erna Frins- Nicole Bobrowski -Ulrich Platt - Thomas Wagner
- [3]. Differential Optical Absorption Spectroscopy (DOAS) using Targets: SO₂ and NO₂ Measurements in Montevideo City
Ilia Loubana - Gustavo Pirizb - Ulrich Platta, - Erna Frinsb
- [4]. Diseño y Construcción de un Sistema para la detección remota de contaminantes atmosféricos por métodos espectroscópicos
Marcello Yanuzzi - Roque Gagliano
Universisdad de la República
- [5]. Seguridad en redes inalámbricas
Tania Díaz - Gabriela Corrales - Patricia Ibarburu
Universisdad de la República
- [6]. Todo Robot (29 de agosto 2006). Tutorial sobre motores de paso
Visitado el 7 de febrero de 2007.
<http://www.todorobot.com.ar/informacion/tutorial%20stepper/stepper-tutorial.htm>
- [7]. Wikipedia, la enciclopedia libre (31 de enero 2007) Acimut
Visitado el 8 de febrero de 2007.
<http://es.wikipedia.org/wiki/Acimut>
- [8]. Wikipedia, la enciclopedia libre (31 de enero 2007) Altura (Astronomía)
Visitado el 8 de febrero de 2007.
http://es.wikipedia.org/wiki/Altura_%28astronom%C3%ADa%29
- [9]. Tasco. Com Galaxsee 461145000
Visitado el 20 de noviembre de 2006 Actualmente offline
http://www.tasco.com/telescopes/print_46114500.shtml
- [10]. Grupo SFK (1989) SFK Catálogo general
Handbook de rodamientos
Reg 47 · 63 000 · 1989-10

- [11]. Apache. http Server Project (2005) Servidor http
Visitado el 12 de febrero de 2007
<http://www.apache.org>
- [12]. Webmin (27 enero 2007) Administración remota
Visitado 12 de febrero de 2007
<http://www.webmin.com>
- [13]. Ocean Optics. Catálogo del ADC2000
Visitado Noviembre de 2006
<http://www.oceanoptics.com/technical/engineering/ADC1000.pdf>
- [14]. Linux Journal (1 de junio 2004)“Writing a Real Driver – In User Space”
Visitado Noviembre de 2006
<http://www.linuxjournal.com/article/7466>
- [15]. Libusb developers guide
Visitado Noviembre de 2006
<http://libusb.sourceforge.net> .
- [16]. Universal Serial Bus Specification (febrero 2007)
Visitado Diciembre 2006
<http://www.usb.org>
- [17]. Sourceforge.net (2007) Usbsnoop
Visitado Enero de 2007
<http://www.sourceforge.net/projects/usbsnoop/>
- [18]. Linux. Slax.
Visitado mayo de 2007
<http://www.slax.org>
- [19]. Linux. Pendrive linux.
Visitado junio 2007
<http://www.pendrivelinux.com/2006/09/20/all-in-one-usb-slaxzip>
- [20]. Gnuplot
Visitado marzo 2007
<http://www.gnuplot.info>
- [21]. xawtv
Visitado mayo 2007
<http://linux.bytesex.org/xawtv>
- [22]. Microchip. PICDEM FS USB Demonstration Board.
Visitado abril 2007
ww1.microchip.com/downloads/en/devicedoc/51526a.pdf

- [23]. LDD – Linux Device Drivers
O 'Reily
Tercera Edición 2005 ISBN: 0-596-00590-3

Notas



Proyecto de Fin de Carrera - Plan 97

Plano del Sistema Mecánico

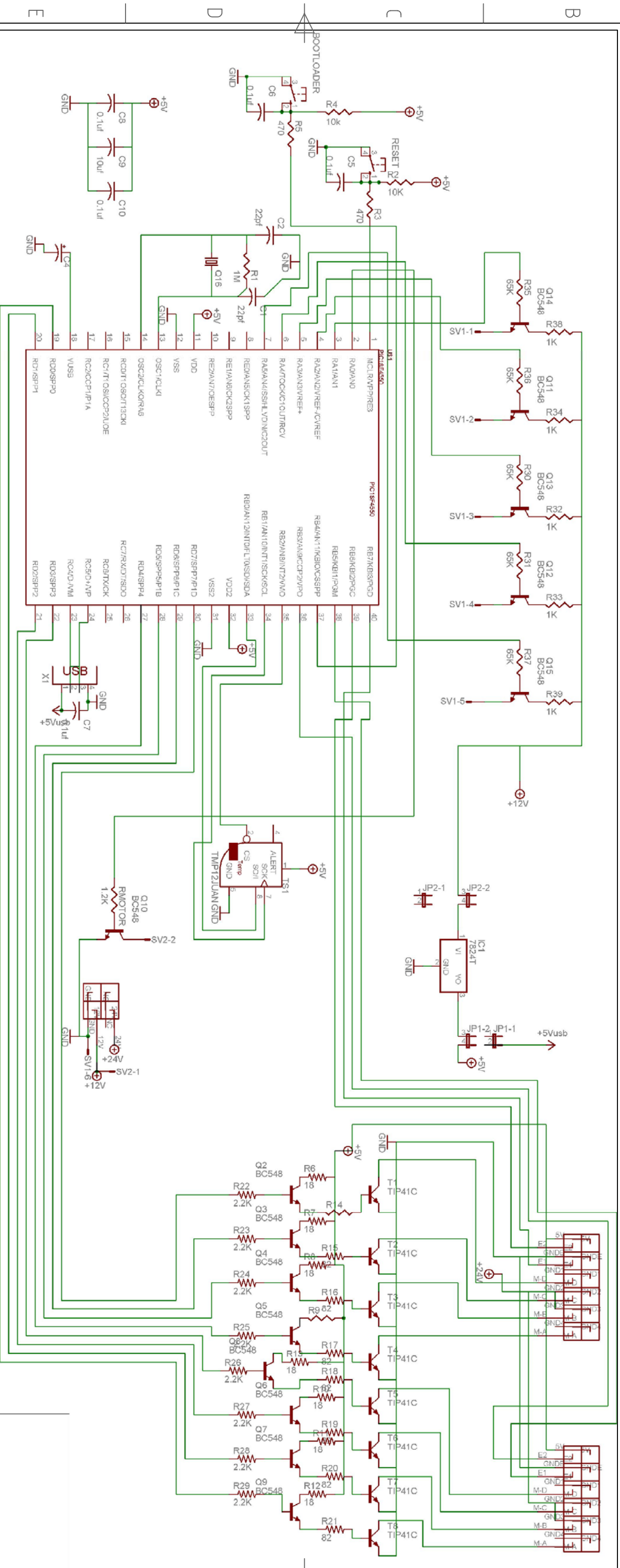
Scale 1:5

Date 10/11/2006



DESIGNED_BY Sergio Blanco
Santiago Conde
Juan Gentilini

Instituto de Ingeniería
Eléctrica
Instituto de Física
Facultad de Ingeniería
Universidad de la República



Proyecto de Fin de Carrera - Plan 97

Placa de Control-Esquemático

Plane 1/3

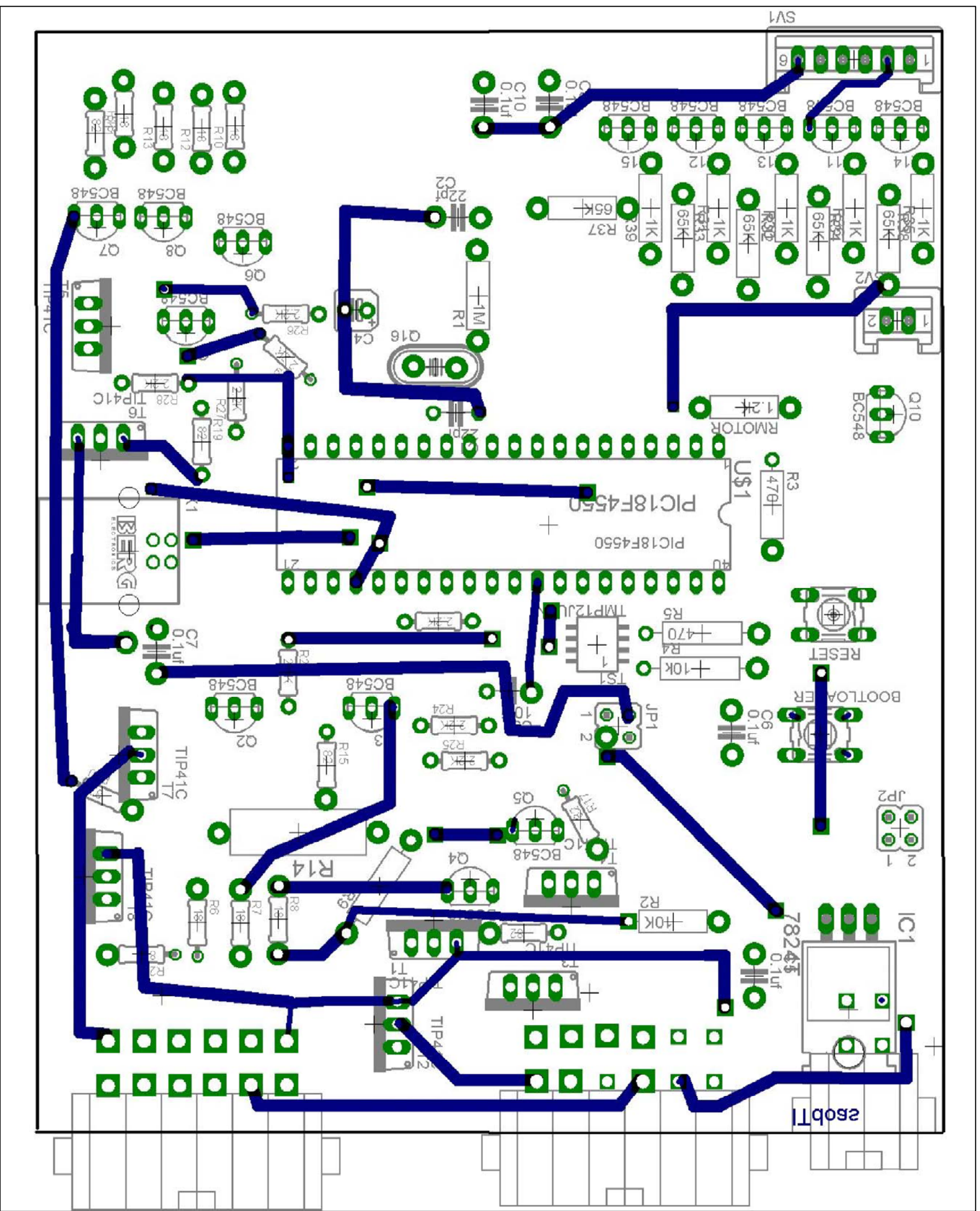
Date 7/8/2007

DESIGNED_BY

Sergio Blanco
Santiago Conde
Juan Gentilini



Instituto de Ingeniería
Eléctrica
Instituto de Física
Facultad de Ingeniería
Universidad de la República



Proyecto de Fin de Carrera - Plan 97

Placa de Control-Board Top

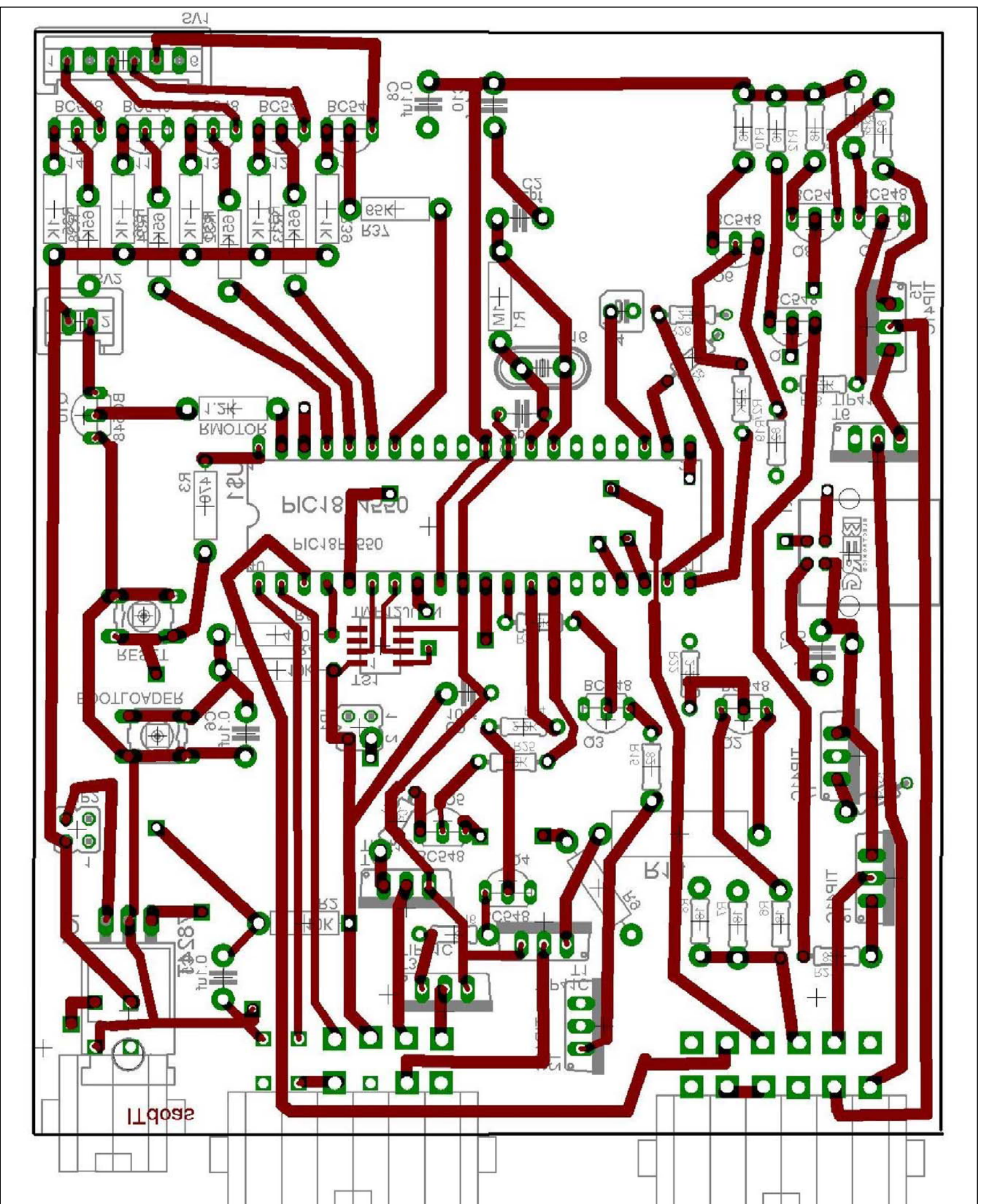
Plane 2/3

Date 7/8/2007



DESIGNED_BY
Sergio Blanco
Santiago Conde
Juan Gentilini

Instituto de Ingeniería
Eléctrica
Instituto de Física
Facultad de Ingeniería
Universidad de la República



Proyecto de Fin de Carrera - Plan 97

Placa de Control-Board Down

Plane 3/3

Date 7/8/2007

DESIGNED_BY

Sergio Blanco
Santiago Conde
Juan Gentilini

Instituto de Ingeniería
Eléctrica
Instituto de Física
Facultad de Ingeniería
Universidad de la República

