



Universidad Autónoma de Baja California  
Ensenada

---

---

Facultad de Ingeniería  
Electrónica

# Sistema para la medición de flujos turbulentos

para la determinación de fuentes  
y sumideros de CO<sub>2</sub>

TESIS

que para obtener el grado académico de

**INGENIERO EN ELECTRÓNICA**

presenta

**Angel Castro Castro**



Ensenada, B. C.

Abril, 2007

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA

Sistema para la medición de  
flujos turbulentos  
para la determinación de  
fuentes y sumideros de de  $CO_2$

Tesis profesional

que presenta

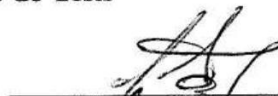
Angel Castro Castro

aprobada por:

Directores de Tesis

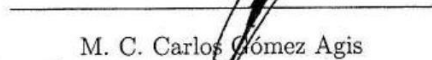


Dr. Raul Michel Murillo



M. C. Martín Escoto Rodríguez

Sinodales



M. C. Carlos Gómez Agis



Dr. Oscar López Bonilla



Dr. Miguel Martínez Rosas

## DEL SALÓN EN EL ÁNGULO OSCURO...

Del salón en el ángulo oscuro,  
de su dueño tal vez olvidada,  
silenciosa y cubierta de polvo  
veíase el arpa.  
¡Cuánta nota dormía en sus cuerdas,  
como el pájaro duerme en la ramas,  
esperando la mano de nieve  
que sabe arrancarlas!  
¡Ay!, pensé. ¡Cuántas veces el genio  
así duerme en el fondo del alma,  
y una voz, como Lázaro, espera  
que le diga: "Levántate y anda!"  
- Gustavo Adolfo Bécquer -

## Agradecimientos

La lista de personas a las cuales debo agradecer y merecen ser agradecidas es larga y concurrida, así que inicio con temor de olvidar a alguna. Aquellos cuyos nombres no sean mencionados deben saber que son apreciados y es la premura de esta redacción la que os ha olvidado, no mi estimación.

Por su incansable esfuerzo y dedicación, sus desvelos, sus privaciones, su entusiasmo y su enorme voluntad agradezco a mis padres Hernolberto y Saida Castro, así como a mi hermana Claudia. Son realmente pocas las palabras capaces de retratar un momento de contentamiento como éste que ahora intento describir.

Este agradecimiento lo hago extensivo y no menos importante a aquellos nuestros ascendientes cuyas vidas han culminado en las nuestras a través de sus éxitos y sus fracasos. A mi abuelos Alicia, Aureliano, Adela y Heriberto, que en paz descanse: gracias.

Por su apoyo y su paciencia en la realización del presente trabajo deposito mi confianza y aprecio en mis asesores Raúl Michel y Martín Escoto. El camino hasta hoy recorrido ha sido largo y a veces sinuoso, son las rugosidades del sendero las que forjan a aquellos que lo caminan, gracias por caminar conmigo.

Son ya varios años de respaldo los que cuentan en su haber, es también su paciencia y su reconocimiento quienes son dignos de referencia y meritorios de mención. Si bien redundo en mis palabras debo aclarar que así es necesario. A las familias Esparza-Núñez y Pánfilo-Osuna, con todo mi aprecio.

A mi estimada familia -que no somos pocos- dedico reflexión y agradecimiento. Es una misma la rama que nos cobija en aquellos días de frío y vendaval. A mis tíos y a toda la primada extendiendo mi afección.

Mis amigos son tantos y a la vez tan pocos, pero no por pocos menos valiosos, cada uno de ustedes ocupa un lugar muy especial y si son pocas las veces que lo menciono no son pocas aquellas que lo reafirmo. Por su amistad, su comprensión y simpatía, mil gracias. Mil... son realmente tan pocas, pero como todo, debo ahorrar, no se si ya se fijaron pero he mencionado la palabra *gracias* no se ya cuantas veces y aunque aún tengo muchas por repartir son también muchos mis deudores.

Aunque siempre he dicho que los maestros son máquinas expresamente diseñadas para calificar exámenes, este es el momento justo para reconocer su apoyo, sus bondades y también sus defectos ya que estos también forman - o deforman - a los incautos estudiantes, sinceramente espero pertenecer al primer grupo. Como en todo, el orden es el camino para limar asperezas y evitar entuertos, en orden cronológico agradezco a mis maestros: a Judith S. por la formación académica y humana recibida durante la escuela primaria. A las maestras Esperanza L., Alicia G. y Georgina M. ya que bajo su firme comando hube de realizar apreciables acciones. A los maestros Carlos G., Israel G., Ricardo C. y Humberto C. así como Sonia R. de la Facultad de Ingeniería de la UABC. Al maestro Juan P. quién me apoyó en mi estancia del Verano de la Investigación del 2005 en Cd. Obregón, Sonora, y muy especialmente a la familia Gómez quienes me recibieron durante dicha estancia. A la maestra Patricia D. de la Facultad de Idiomas por su tolerancia y sus conocimientos.

Al los Dres. Stephen Smith y Steve Bullock, así como al Biol. Mario Salazar, involucrados en el proyecto TERMAR. Al departamento de Instrumentación del IA-UNAM, Ensenada, donde gran parte del trabajo descrito en las siguientes páginas ha tenido lugar. A mi amigo Ing. Juan Martínez Osuna, notable compañero y colaborador.

Al personal de investigadores del IA-UNAM, cuya actividad no deja de sorprenderme. Agradezco puntualmente la ayudada brindada por los Drs. Raul Michel y Roberto Vázquez (máximos benefactores materiales del cubículo de tesis de licenciatura del IA-UNAM-Eda). También aquellos que cuidan por nuestra seguridad y la de las instalaciones Oscar M. y Gaspar M.

A todo el personal del Observatorio Astronómico Nacional en San Pedro Mártir y muy especialmente a las cocineras Gema C., Angela M. y Carmen G. que hacen nuestras estancias allá arriba más sabrosas.

Al CONACyT por financiar el desarrollo descrito en este trabajo (proyecto SEP-CONACyT V43422-F) y por su apoyo durante mi carrera universitaria.

A. Castro

# Índice general

<b>1. Resumen</b>	<b>11</b>
<b>2. Antecedentes</b>	<b>13</b>
2.1. Estación meteorológica . . . . .	13
2.2. Estaciones de meteorológicas del Observatorio Astronómico Nacional . . . . .	14
2.3. Torres de Correlación Turbulenta . . . . .	14
2.3.1. Correlación o covarianza turbulenta . . . . .	16
2.3.2. Mediciones micrometeorológicas . . . . .	17
2.4. Proyecto TERMAR . . . . .	17
2.4.1. Objetivos específicos . . . . .	19
2.4.2. Características de la zona de estudio . . . . .	21
<b>3. Sistema de medición de flujo turbulento</b>	<b>25</b>
3.1. Microcontrolador ADuC848 . . . . .	26
3.2. Protocolos de comunicación y circuito de conversión EIA-485/EIA-232 . . . . .	30
3.2.1. Estándar EIA-232 . . . . .	31
3.2.2. Estándar EIA-485 . . . . .	34
3.2.3. Convertidor EIA-232/EIA-485 . . . . .	36
3.3. Módulos de adquisición de datos y acondicionamiento de señales . . . . .	38
3.3.1. Tarjeta de adquisición de datos . . . . .	38
3.3.2. Selector RS-232/RS-485 . . . . .	40
3.3.3. Tarjeta de Acondicionamiento de señales . . . . .	45
3.4. Microcomputadora Gumstix WaySmall 200ax . . . . .	45

3.4.1. Sistemas integrados . . . . .	45
3.4.2. Microcomputadora Gumstix . . . . .	47
3.4.3. Tarjetas de expansión de puertos . . . . .	47
3.4.4. Memoria externa . . . . .	50
3.5. Sistema de energía solar . . . . .	52
<b>4. Sensores meteorológicos y edáficos</b>	<b>55</b>
4.1. Analizador de concentración de gases de $CO_2/H_2O$ . . . . .	55
4.1.1. Descripción general del LI-7500 . . . . .	55
4.1.2. Realización de mediciones . . . . .	56
4.1.3. Software y señales de salida . . . . .	58
4.1.4. Calibración . . . . .	58
4.2. Anemómetro ultrasónico tridimensional . . . . .	59
4.2.1. Especificaciones . . . . .	60
4.3. Presión barométrica . . . . .	62
4.4. Precipitación pluvial . . . . .	63
4.5. Temperatura ambiental y humedad relativa . . . . .	64
4.5.1. Sensor de temperatura . . . . .	64
4.5.2. Sensor de humedad relativa . . . . .	64
4.6. Temperatura del suelo . . . . .	65
4.6.1. Cálculo de las resistencias del puente de Wheatstone . . . . .	65
4.6.2. Determinación de $R_{TH}$ . . . . .	67
4.7. Sensores de radiación solar . . . . .	67
4.7.1. Radiación solar directa . . . . .	67
4.7.2. Radiación fotosintéticamente activa . . . . .	68
4.7.3. Radiómetro neto . . . . .	69
4.8. Humedad del suelo . . . . .	70
4.8.1. Calibraciones . . . . .	70
<b>5. Torres experimentales de correlación turbulenta</b>	<b>73</b>
5.1. Programación . . . . .	77

<i>ÍNDICE GENERAL</i>	3
<b>6. Sistema de archivos</b>	<b>81</b>
6.1. Formato de los archivos de datos . . . . .	81
6.1.1. Archivos del módulo de adquisición . . . . .	81
6.1.2. Archivos del sistema IRGA-Anemómetro . . . . .	82
6.2. Archivos de calibración . . . . .	83
6.3. Archivos de configuración . . . . .	84
<b>7. Datos experimentales</b>	<b>87</b>
<b>8. Conclusiones</b>	<b>91</b>
<b>A. Esquemáticos y PCBs</b>	<b>95</b>
<b>B. Programación</b>	<b>109</b>
B.1. Programas empleados en las TCT . . . . .	110
B.1.1. Programa del microcontrolador ADuC848 . . . . .	110
B.1.2. Programa de la Gumstix para lectura del Anemómetro Ultrasonico . . . . .	122
B.1.3. Programa para la lectura de la red de módulos de adquisición . . . . .	126
B.2. Programas para despliegue y procesamiento de datos . . . . .	131
B.2.1. Programa para el calculo de medias estadísticas . . . . .	131
B.2.2. Programa para graficación de datos . . . . .	132
<b>C. Configuración de la microcomputadora Gumstix©</b>	<b>137</b>
C.1. Instalación de archivos en la estación de trabajo (PC o Laptop) . . . . .	137
C.1.1. Instalación de la Subversión y Buildroot . . . . .	137
C.1.2. Compilador cruzado . . . . .	139
C.2. Acceso a la Gumstix vía puerto serial . . . . .	139
C.3. Configuración de Interfaz USBnet . . . . .	142
C.3.1. Asignación temporal de IP para comunicación Gumstix- PC vía USBnet . . . . .	142
C.3.2. Asignación definitiva de IP para comunicación Gumstix- PC vía USBnet . . . . .	143
C.4. Configuración de memoria externa Compact Flash . . . . .	145
C.5. Reemplazo de la imagen del sistema (usando KERMIT) . . . . .	146
C.6. Cargando El Kernel . . . . .	150

C.7. Recuperación de Archivos Corruptos del Boot . . . . .	150
C.8. Problemas comunes . . . . .	151
C.9. Cambio de Baudrate . . . . .	151
C.10. Transferencia de archivos . . . . .	153
C.11. Acceso remoto . . . . .	154
C.12. Configuración de arranque . . . . .	154
C.13. Liberación del puerto serie ttyS0 . . . . .	155
C.14. Recuperación del puerto serie ttyS0 . . . . .	156
<b>D. Scripts</b>	<b>157</b>
D.1. Adaptador USB-Serial . . . . .	157
D.2. Memoria USB . . . . .	158
D.2.1. Montaje de memorias USB . . . . .	158
D.2.2. Desmontaje de memorias USB . . . . .	158
D.3. Acceso a la Gumstix . . . . .	158
D.4. Descarga de datos . . . . .	159
<b>E. Galería fotográfica TCT</b>	<b>161</b>

# Índice de figuras

2.1. Despliegue gráfico de la información climatológica correspondiente al Telescopio de 1.5m en SPM . . . . .	15
2.2. Ciclo del carbono. El $CO_2$ es integrado a la atmósfera por concepto de respiración y combustión. El material orgánico es degradado en el sustrato, y reciclado como materia prima o nutriente . . . . .	18
2.3. Arrastre bruto de material producto de la erosión de las cuencas ( <i>arriba</i> ). Erosión del carbono orgánico ( <i>abajo</i> ). . . . .	22
2.4. Características de relieve, se muestra en el gráfico la pendiente ( <i>arriba</i> ). La vegetación de la zona de Ensenada ( <i>abajo</i> ). . . . .	23
3.1. Configuración con fuente de voltaje analógica y digital . . . . .	28
3.2. Esquemas de planos de tierra . . . . .	29
3.3. Comparación de estándares para la transmisión de datos. . . . .	30
3.4. Características eléctricas del RS-232. Codificación NRZ. Intervalos del voltaje de transmisión en el alambre conductor. . . . .	31
3.5. Características eléctricas del RS-232. Codificación NRZ. . . . .	32
3.6. Distribución de terminales en el conector DB9 . . . . .	33
3.7. Campo magnético (vertical), corriente inducida de ruido (horizontal) . . . . .	34
3.8. Cadena de margaritas . . . . .	35
3.9. Conversor RS-232/RS-485 construido en el IA-UNAM . . . . .	36
3.10. Convertidor RS-232/RS-485 . . . . .	37
3.11. Conexión física de la cadena de margaritas . . . . .	37
3.12. Diagrama a bloques del módulo de adquisición de datos . . . . .	39
3.13. Tarjeta de adquisición de datos de diseñada y construida para las torres de correlación turbulenta. Es un sistema cuyas característica le permiten ser empleado en una gran variedad de aplicaciones. . . . .	40

3.14. Sistema básico. . . . .	43
3.15. Puertos de la tarjeta de adquisición de datos. . . . .	43
3.16. (a)MAX232, (b)Selector RS-232/RS-485. . . . .	44
3.17. a)Bloque básico de acondicionamiento de señal, b)Referencia de voltaje. . . . .	45
3.18. La tarjeta de acondicionamiento de señales incluye un banco de amplificadores de instrumentación así como un amplificador en modo de saturación. . . . .	46
3.19. Tarjeta madre de la Gumstix WaySmall 200ax. . . . .	47
3.20. Tarjeta de expansión de puertos de la Gumstix . . . . .	49
3.21. Modificaciones a la WaySmall-Original para implementar el puerto ttyS2. . . . .	50
3.22. Memoria Compact Flash tipo II High Speed. . . . .	51
3.23. CFSStix (izquierda), montura del sistema completo (derecha). . . . .	51
3.24. Diagrama de la montura polar y almacenamiento de reserva. . . . .	52
4.1. Diagrama de la estructura interna del analizador de gases infrarrojo LI-7500. . . . .	57
4.2. LI-610. Generador de punto de condensación ( <i>izquierda</i> ) para calibración del flujo mediante la inserción del tubo de calibración en la trayectoria del haz infrarrojo ( <i>derecha</i> ). . . . .	59
4.3. Anemómetro ultrasónico tridimensional Young 81000V. . . . .	60
4.4. Orientación del anemómetro ultrasónico tridimensional Young 81000V. . . . .	61
4.5. Medidor de presión barométrica con diagrama de conexiones. . . . .	62
4.6. Pluviómetro de cazoletas basculantes. . . . .	63
4.7. Sensor de temperatura ambiente y humedad relativa HMP45C. . . . .	64
4.8. Puente de Wheatstone. . . . .	65
4.9. Medidor de radiación solar directa. . . . .	68
4.10. El sensor de radiación fotosintéticamente activa ( <i>izquierda</i> ) detecta longitudes de onda en el espectro electromagnético de 400 a 700 nm, que constituye el rango de radiación útil para las plantas ( <i>derecha</i> ). . . . .	68
4.11. Conexiones radiómetro neto NR LITE. . . . .	69
4.12. Sensor de contenido volumétrico de agua en el suelo. . . . .	71
5.1. Torre de correlación turbulenta. Este tipo de torres son más complejas que sus análogas meteorológicas. El modelo construido para los fines del proyecto TERMAR incluye algunas mejoras sobre los torres empleadas por otros grupos de investigación. . . . .	75

5.2. Diagrama básico de operación. Este sistema se encuentra en operación con solo un módulo conectado a la red RS-485. Cada módulo cuenta con ocho canales de conversión A/D y 2 contadores. Cuenta también con 4 puertos I2C y los datos obtenidos pueden ser almacenados en memoria externa. . . . .	76
5.3. Diagrama de flujo del programa del microcontrolador. Este programa se encuentra constantemente leyendo el puerto serial en espera de escuchar su dirección, cuando esto ocurre promedia todas las muestras que ha tomado desde la última ocasión en que estos datos le fueron requeridos y una vez que envía una cadena de datos continúa muestreando en espera de que le sean requeridos nuevamente. . . . .	78
5.4. Diagrama de flujo del programa de lectura de la red de módulos de adquisición. Este programa envía las direcciones de los módulos y lee las cadenas de datos enviadas por cada uno de ellos en forma secuencial. Luego espera un intervalo de tiempo configurable por el programador, en este caso de un minuto. . . . .	79
5.5. Diagrama de flujo del programa de lectura de la red de módulos de adquisición. Este programa envía las direcciones de los módulos y lee las cadenas de datos enviadas por cada uno de ellos en forma secuencial. Luego espera un intervalo de tiempo configurable por el programador, en este caso de un minuto. . . . .	80
7.1. Señales adquiridas por el sistema de adquisición de datos y almacenamiento. . . . .	88
7.2. Comportamiento de los parámetros de viento el 10 de febrero de 2007. . . . .	89
7.3. Datos obtenidos con los sensores IRGA-Anemómetro del periodo del 17 a 26 de enero de 2007 en la TCT experimental. . . . .	90
A.1. Esquemático: Tarjeta de acceso a ductos. . . . .	96
A.2. Esquemático: Tarjeta de adquisición, página 1/2. . . . .	97
A.3. Esquemático: Tarjeta de adquisición, página 2/2. . . . .	98
A.4. Esquemático: Convertidor RS-485/RS-232. . . . .	99
A.5. Esquemático: Tarjeta de acondicionamiento de señales, página 1/2. . . . .	100
A.6. Esquemático: Tarjeta de acondicionamiento de señales, página 2/2. . . . .	101
A.7. PCB: Tarjeta de acceso a ductos. . . . .	102
A.8. PCB: Tarjeta de adquisición, máscara de componentes. . . . .	102
A.9. PCB: Tarjeta de adquisición, capa superior. . . . .	103
A.10. PCB: Tarjeta de adquisición, capa inferior. . . . .	103
A.11. PCB: Convertidor RS-485/RS-232 (ambas capas). . . . .	104
A.12. PCB: Tarjeta de acondicionamiento de señales, máscara de componentes. . . . .	105
A.13. PCB: Tarjeta de acondicionamiento de señales, capa superior. . . . .	106

A.14.PCB: Tarjeta de acondicionamiento de señales, capa inferior. . . . . 107

C.1. Ubicación del puerto ttyS0. . . . . 140

# Índice de tablas

3.1. Descripción de terminales en el conector DB9 . . . . .	33
3.2. Asignación de líneas en el puerto digital (conector DB15) . . . . .	41
3.3. Líneas asignadas al puerto analógico (conector DB25) . . . . .	42
3.4. Selección manual del tipo de comunicación en la tarjeta de adquisición . . . . .	45
3.5. Contenido de la microcomputadora Gumstix . . . . .	48
5.1. Descripción de componentes montados sobre la Torre de Correlación Trubulenta . . . . .	74
6.1. Descripción del nombre de archivo asignado a los módulos de adquisición de datos . . . . .	81
6.2. Archivo de datos crudos (voltajes) provenientes de los sensores. En el nombre del archivo se define el módulo al que corresponde. . . . .	82
6.3. Descripción del nombre de archivo asignado a los datos obtenidos del sistema IRGA-Anemómetro. . . . .	82
6.4. Archivo de datos generado por el sistema IRGA-Anemómetro. En el nombre del archivo se indica la torre a la cual pertenece. . . . .	83
6.5. Los archivos de calibración se organizan en columnas conteniendo los coeficientes de un polinomio de quinto orden y comentarios que no son leídos por el programa principal. . . . .	84
6.6. Solo se requiere indicar el puerto serial que será leído. . . . .	84
6.7. En los archivos de configuración se indica la cantidad de modulos conectados a la red de datos, así como la etiqueta correspondiente a cada uno de ellos y el puerto serial a través del cual son leídos. Pueden conectarse hasta 8 módulos en un mismo puerto. El ID puede corresponder a la torre marina (MAR), torre terrestre (TER) o la torre móvil (MOV) y se verá reflejado en los nombres de los archivos de datos. . . . .	85



# Capítulo 1

## Resumen

El Protocolo de Kyoto es un instrumento político que busca disminuir en un 5 %, dentro del periodo de 2008 a 2012, las emisiones de los principales gases causantes del denominado calentamiento global. Dentro del marco de este protocolo ha tenido lugar una enorme actividad científica con el propósito de cuantificar la liberación de los gases de carbono, nombrados fuentes y sumideros respectivamente, en diferentes ecosistemas. La mayor parte de estas investigaciones han empleado torres de correlación turbulenta (TCT) como herramienta fundamental de medición.

En el presente trabajo, desarrollado en el Instituto de Astronomía de la UNAM (IA-UNAM) y Centro de Investigación Científica y Educación Superior de Ensenada (CICESE), se presenta el diseño e implementación de sistemas electrónicos para adquirir información meteorológica y edáfica que permitirán, en un futuro cercano, la cuantificación de los flujos verticales y horizontales de carbono. Este tipo de sistemas de medición no son nuevos y han sido empleados en numerosos estudios alrededor del mundo. Los objetivos principales de este trabajo fueron la disminución del espacio físico ocupado por el sistema así como los costos de producción, comparados con equipos de medición empleados por la San Diego State University (SDSU) y el Centro de Investigaciones Biológicas del Noroeste (CIBNOR).

El sistema de medición que aquí se describe consiste básicamente en un dispositivo maestro encargado de la adquisición y procesamiento de datos provenientes de una red de sensores controlados por medio de módulos de adquisición. Demostrando ser un sistema versátil capaz de ser empleado en los distintos ambientes para los cuales fue diseñado.

El sistema se divide básicamente en cuatro bloques bien diferenciados: acondicionamiento de datos, adquisición de datos, procesamiento de datos y fuente de energía solar. El primero de ellos permite el máximo aprovechamiento de los canales de conversión analógica a digital de las tarjetas de adquisición. La adquisición y procesamiento de datos culminan en archivos de datos que contienen las variables medidas con las TCTs. Una fuente de energía solar permiten al equipo operar en sitios remotos donde el acceso a la energía eléctrica convencional es difícil o imposible. De esta manera las mediciones pueden ser realizadas de forma autónoma y sin necesidad de mantenimiento frecuente.

Además de una serie de módulos de adquisición de datos, se usan sensores especializados para la medición de los flujos de  $CO_2$  y vapor de agua. Se emplea una microcomputadora Gumstix como elemento de control de flujo de información y procesamiento de datos y almacenamiento de datos en una memoria externa.

Un propósito fundamental de este desarrollo ha sido el abaratamiento de este tipo de sistemas de medición, lo que podría permitir la expansión de este tipo de investigaciones a amplias regiones de nuestro país, así como países con un nivel socio-cultural similar al nuestro, y más específicamente permitir la instalación de varias torres constituyendo una matriz distribuida sobre zonas específicas de investigación. Esto permitirá un conocimiento más profundo de los factores que intervienen en la emisión y captura de gases de carbono.

## Capítulo 2

# Antecedentes

### 2.1. Estación meteorológica

Una estación meteorológica es un equipo dedicado a la medición de los parámetros físicos directamente relacionados con el tiempo atmosférico. Estos equipos cuentan con elementos sensibles que permiten la medición de los componentes meteorológicos. Un transductor es un elemento que puede convertir un determinado tipo de energía en otra diferente. En el ramo de las ciencias eléctricas son utilizados para adquirir información sobre entornos reales mediante la conversión de una determinada variable física a señales o impulsos eléctricos medibles por los sistemas electrónicos. Mediante el uso de estos transductores es posible adquirir información sobre parámetros tales como temperatura, precipitación, humedad relativa, radiación solar, velocidad y dirección del viento, entre otros.

Las estaciones meteorológicas se clasifican en forma general como de lectura visual directa o como registrador. Las estaciones registradoras, a diferencia de las de lectura directa, permiten tener acceso a datos de las variables en estudio tiempo después de ocurridas las mediciones. Esto debido a que tienen la capacidad de almacenar información ya sea plasmándola en forma gráfica o mediante archivos de datos que son retenidos en una memoria interna o en dispositivos externos de almacenamiento.

## 2.2. Estaciones de meteorológicas del Observatorio Astronómico Nacional

En el Instituto de Astronomía de la UNAM (IA-UNAM), campus Ensenada, se tiene experiencia en el diseño y construcción de estaciones meteorológicas, las cuales se están empleando para la caracterización del sitio astronómico de San Pedro Mártir (SPM) y de la distribución de temperaturas en sus telescopios y edificios (Michel et al. 2001, Michel et al. 2003).

Debido a la necesidad de tener un registro del clima en SPM, a finales de 1998 se instalaron dos estaciones meteorológicas rudimentarias en los telescopios de 0.84 y 1.5m, las cuales poseían un conjunto de sensores conectados a una PC por medio de una tarjeta de adquisición de datos comercial. Con ellas fue posible recabar información meteorológica por más de cuatro años (Michel, Hiriart y Chapela 2003). El objetivo principal de estas estaciones era la obtención de datos sobre la distribución de temperaturas dentro y fuera de los edificios, así como la temperatura de los espejos primario y secundario de los telescopios, con el propósito de estudiar su efecto en la calidad de las imágenes obtenidas.

En 2003 se instaló una nueva estación meteorológica en el telescopio de 0.84m, la cual estaba basada en el microcontrolador AT89C52 de Atmel (Chapela 2004) y en 2005 fueron instaladas nuevas estaciones en los telescopios de 0.84, 1.5 y 2.1m (Martínez 2005) con un sistema basado en el microcontrolador ADuC832 de Analog Devices. Estas estaciones han estado fundamentadas en la construcción de módulos, con varias entradas analógicas y digitales, lo que ha permitido generar sistemas robustos, eficientes y de bajo nivel de cableado.

Actualmente existe una página de internet (<http://132.248.4.10/~tlaloc/>) donde se presenta en forma gráfica la información recabada en cada telescopio. En la Figura 2.1 se muestran los datos recientes correspondientes al Telescopio de 1.5 en SPM.

## 2.3. Torres de Correlación Turbulenta

El viento, en interacción con otros factores como el relieve, la vegetación y la temperatura, entre muchos otros, provoca la condición de arrastre de sustancias y material orgánico superficial conocida como flujo

SPM 1.5m Telescope. 03/02/2007 22:40 UT.

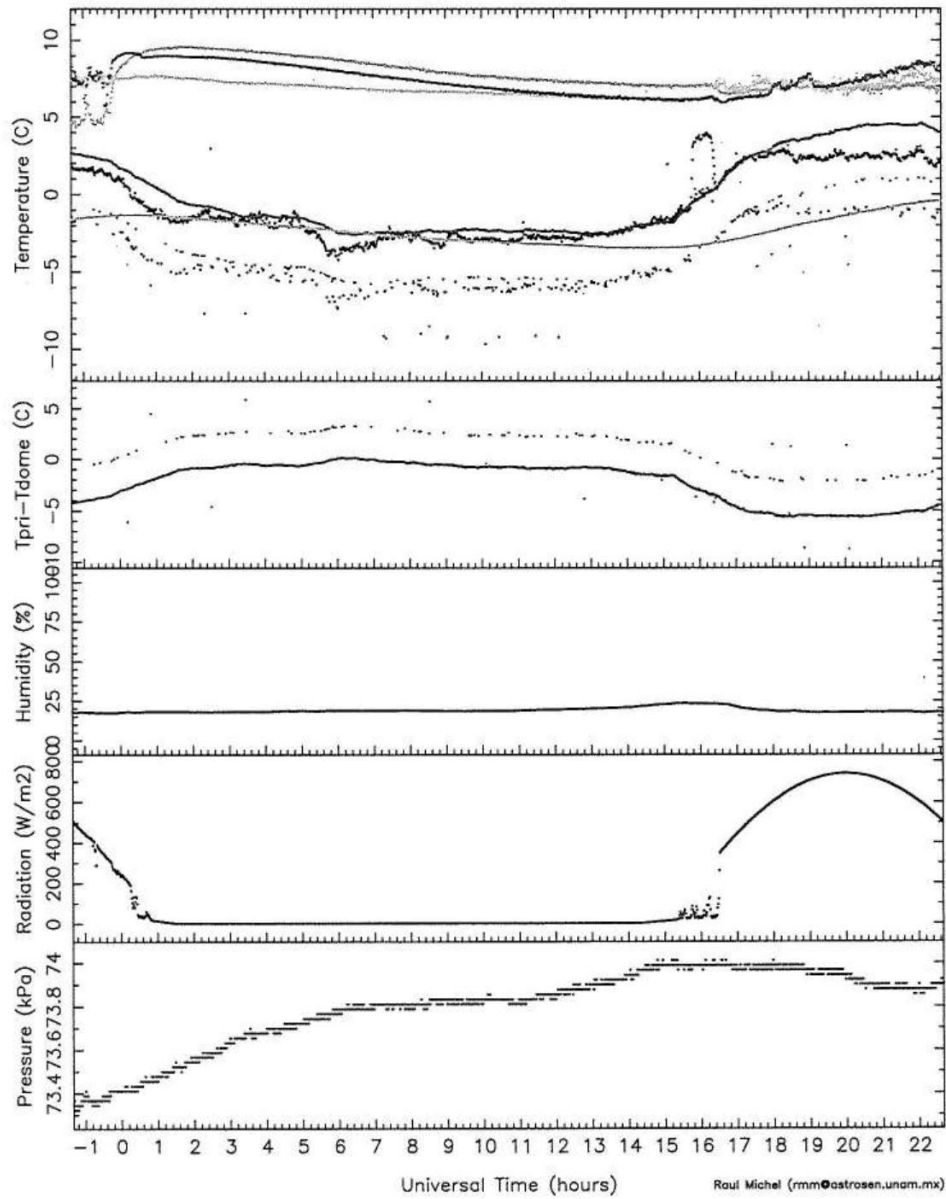


Figura 2.1: Despliegue gráfico de la información climatológica correspondiente al Telescopio de 1.5m en SPM

turbulento, cuyas características son muy diferentes en los vientos presentes a mayores altitudes. El estudio de estos flujos permiten cuantificar el intercambio de masa y energía entre la superficie y la atmósfera. Los flujos se estudian mediante equipo especializado el cual es montado en las denominadas “torres de correlación turbulenta” (*eddy flux correlation towers*), que son, en su conjunto, más complejas que las denominadas torres meteorológicas.

Actualmente existe una red internacional de torres de correlación turbulenta denominada FLUXNET, que cuenta con más de 180 sitios alrededor del mundo (Baldocchi 2003) y que ha permitido el estudio intensivo del *intercambio neto de dióxido de carbono ( $CO_2$ ) en diversos ecosistemas, que van desde los trópicos hasta el Polo Norte*. AMERIFLUX es la denominación que recibe la subred de torres de correlación turbulenta ubicadas en Norteamérica. En México solo hay una torre de este tipo en funcionamiento constante y se encuentra en La Paz, Baja California Sur, mientras que en Sonora existen dos más operando en forma intermitente.

### 2.3.1. Correlación o covarianza turbulenta

Esta técnica micrometeorológica cuantifica directamente el intercambio de masa y energía entre la superficie y la atmósfera mediante la medición del transporte turbulento de vapor de  $H_2O$ ,  $CO_2$  y calor (Baldocchi et al. 1988; Verma 1990). Para medir las variaciones medias de las componentes vertical y horizontal de la velocidad del viento se requiere de un anemómetro-termómetro tridimensional de respuesta rápida que sea capaz de realizar hasta 10 mediciones por segundo. Para cuantificar las concentraciones de  $CO_2$  y vapor de  $H_2O$  se requiere también de un analizador infrarrojo de gases de trayectoria libre que opere en sincronía con el anemómetro. Mediante la aplicación de un proceso de filtrado es posible obtener la dirección predominante del viento, así como las concentraciones de gases de  $CO_2$  y  $H_2O$  en trayectorias verticales tanto ascendentes como descendentes. Los datos crudos generados por estos dos aparatos son procesados por un computador mediante técnicas de filtrado recursivo utilizando datos correspondientes a un determinado intervalo temporal con la finalidad de aplicar procedimientos estadísticos, así como efectuar el proceso de correlación con los datos de otros sensores climatológicos involucrados en las mediciones.

En el trabajo que se describe en esta tesis el método de correlación turbulenta no ha sido una línea de desarrollo y estudio, sino la generación y desarrollo de la electrónica respectiva a la instrumentación y adquisición de datos, así como la implementación de algunos procesos de *software* y funciones estadísticas implicadas en la

obtención y almacenamiento de datos.

### 2.3.2. Mediciones micrometeorológicas

Para completar y validar el método de correlación turbulenta es necesario además cuantificar parámetros meteorológicos y edáficos (del suelo) de importancia tales como radiación neta, la densidad de flujo fotónico fotosintético, la radiación difusa, la temperatura del aire y del suelo, la presión barométrica, la humedad relativa de suelo y del aire, la precipitación pluvial y los gradientes de temperatura en el suelo, entre otros.

## 2.4. Proyecto TERMAR

El efecto del ciclo del carbono en el clima global es cada vez más entendido y apreciado. Las actividades humanas (especialmente la quema de combustibles fósiles) han incrementado el contenido de  $CO_2$  en la atmósfera con lo que se ha promovido el efecto invernadero. Falkowski et al. (2000) han demostrado que los niveles de  $CO_2$  en la Tierra están 35 % por arriba de cualquier otro nivel registrado en los últimos 420,000 años. El 60 % de éste  $CO_2$  se encuentra en la atmósfera, el 35 % en el océano y el resto aparentemente es absorbido por la biosfera terrestre. El estudio del ciclo de carbono (Figura 2.2) es importante en el contexto del “Kyoto Protocol to the United Nations Framework Convention on Climate Change”, firmado por México y 202 países más el 29 de septiembre de 2003, donde, entre otros asuntos, se determina que los países firmantes deben demostrar que están disminuyendo sus emisiones antropogénicas de acuerdo a un tabulador establecido. Debe mencionarse que Estados Unidos de América y Australia, a pesar de no ser firmantes de dicho tratado, muestran mucha disposición a la realización de éste tipo de estudios más no en la reducción de sus emisiones. Como resultado de las reglas acordadas por el Protocolo ha ocurrido un incremento en la investigación para estimar el almacenamiento del carbono en la biomasa en crecimiento. Dicha investigación se realiza principalmente utilizando torres de correlación turbulenta, que proporciona una elegante contabilidad de los flujos verticales de  $CO_2$ , vapor de agua y calor en el ecosistema. Esta técnica es utilizada comúnmente en sistemas terrestres aunque puede ser empleada, con más dificultad, en ambientes marinos.

El CICESE ha iniciado un proyecto de investigación denominado “*Flujos TERMAR: Enumerating  $CO_2$  sources and Sinks in a regional coastal terrestrial and marine habitat matrix*” (SEP-Conacyt 43422), en el que se estudia el ciclo del carbono en los alrededores de Ensenada.

El proyecto difiere conceptualmente de otros estudios de flujo de  $CO_2$  realizados por distintos grupos de investigación. La originalidad de este proyecto radica en que intenta incluir los flujos laterales y verticales del carbono, así como el comparar flujos marinos y terrestres en regiones adyacentes. La hipótesis es que una parte significativa del  $CO_2$  fijado como materia orgánica por comunidades vegetales es desplazada de los sitios iniciales de fijación y oxidada fuera del sistema original. En tierra el transporte se realiza mediante la erosión por aire o agua, mientras que en el océano se realiza por circulación del agua, mediante corrientes horizontales inducidas por vientos y mareas, así como surgencias a lo largo de la costa del Pacífico de Baja California. La importancia de ésta hipótesis es que los flujos verticales de  $CO_2$ , muestreados únicamente de forma local, sobrerrepresentan a las comunidades de absorción o producción de carbono orgánico.

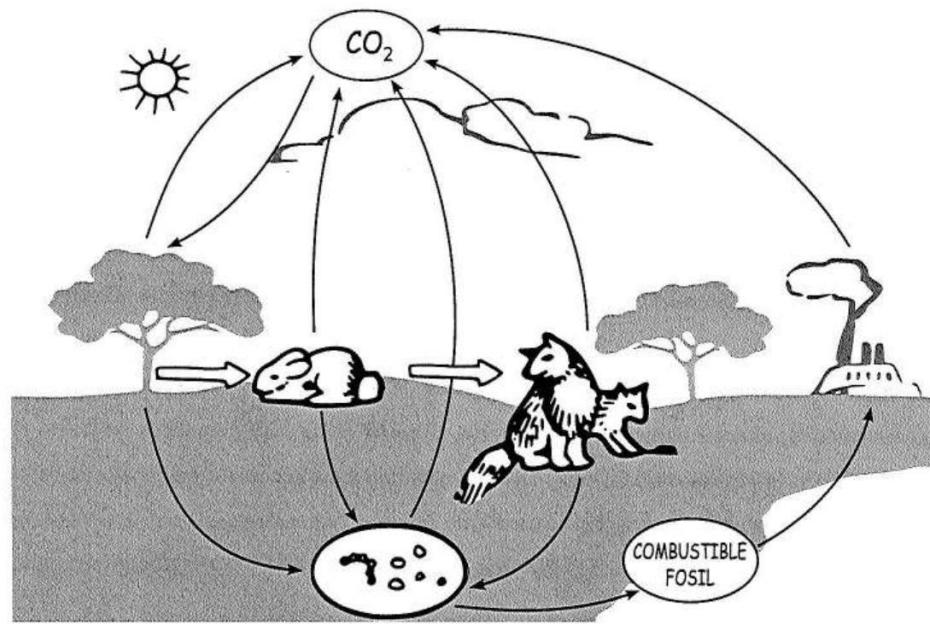


Figura 2.2: Ciclo del carbono. El  $CO_2$  es integrado a la atmósfera por concepto de respiración y combustión. El material orgánico es degradado en el sustrato, y reciclado como materia prima o nutriente

La investigación propuesta por el proyecto *Flujos TERMAR* intenta evaluar, para comunidades terrestres y marinas costeras, los flujos verticales y laterales de materiales (principalmente carbono, vapor de agua y aire). A continuación se describen los aspectos principales del estudio del flujo del carbono orgánico en los ecosistemas.

1. Fijación del  $CO_2$  en las plantas y liberación del  $CO_2$  durante procesos de oxidación de material orgánico (usualmente respiración).
2. Retención del carbón en suelos.
3. Utilización de compuestos carbónicos por humanos.
4. Transporte de carbono, en varias formas (orgánica e inorgánica; disuelta y particulada) desde un entorno a otro. El tiempo de permanencia del material orgánico en el entorno puede ser afectado por este transporte. Por ejemplo, el material orgánico puede ser más reactivo si es desplazado desde un ambiente seco a uno subacuático o hacia algún manto freático.
5. Un tipo interesante de transporte es la afluencia del carbón arrastrado por algún caudal hacia aguas estuarinas y costeras.
6. Transporte de carbono a lo largo de la plataforma continental por circulación costera.
7. Intercambio de  $CO_2$  entre la atmósfera y el océano.
8. Mezcla vertical de los componentes carbónicos por efecto del viento, oleaje, convexiones, surgencias, mareas y bioturbulencias.
9. Modulación de la mezcla vertical por estratificación y flotabilidad, y procesos eventuales como los vientos de Santa Ana.
10. Almacenamiento de carbón en los sedimentos.
11. Ciclo del carbono retenido en sedimentos. Ordinariamente, este ciclo es extremadamente lento, tomándolo alrededor de  $10^6$ - $10^8$  años. Sin embargo las actividades humanas, como la quema de combustibles fósiles, contribuyen a la aceleración de dicho proceso.

#### 2.4.1. Objetivos específicos

Los objetivos científicos que sigue el proyecto, son los siguientes:

1. Establecer estaciones de monitoreo, en sitios selectos de la región de Ensenada, Baja California, con la finalidad de conocer las características de las transferencias verticales de  $CO_2$  entre la atmósfera y los sistemas marinos y terrestres, así como relacionar estas observaciones con modelos propuestos de productividad primaria. La red de monitoreo debe incluir dos sitios de monitoreo permanente y una torre móvil para experimentos comparativos, la cual será instalada durante periodos de corta duración en diferentes ubicaciones de interés, esto con el objetivo de caracterizar la heterogeneidad de la región. En el Valle de Guadalupe se colocará la torre permanente de monitoreo terrestre, mientras que en Isla de Todos Santos se encontrará la torre de monitoreo marino. El sitio en el Valle de Guadalupe se encuentra en una zona representativa del matorral costero y chaparral de Baja California. Otras zonas podrán ser elegidas para su estudio de acuerdo a sus características ambientales, de ser posible, en sitios con suelo aluvial, y otros lugares libres de la intervención de las actividades humanas. Para ambos sitios, terrestres y marinos, se analizará la variabilidad temporal de los parámetros de interés. En estos sitios se plantea obtener los parámetros que aquí se mencionan:
2. Flujo de  $CO_2$  en aire y tierra utilizando directamente la técnica de correlación turbulenta.
3. Utilizando sensado remoto, observaciones de campo e información de la literatura disponible sobre el tema se pretende mapear las características biogeoquímicas de los sitios. Este mapeo deberá incluir el aspecto histórico de la región (al menos los últimos 30 años) con la finalidad de entender la variabilidad de flujos a largo plazo.
4. Modelaje del clima de la región, particularmente respecto a vientos y precipitación, para inferir la productividad y transporte, así como surgencias y corrientes oceánicas. Estas variables son de mucha importancia para la determinación de los flujos tanto verticales como horizontales del carbono.
5. Modelaje y medición de los flujos laterales de carbono y materiales resultado de la interacción tierra-aire-mar. Un aspecto de relevante importancia es la acción erosiva del viento y el agua, particularmente en lo que refiere al arrastre de materiales disueltos y particulados hacia zonas más bajas. Utilizando un modelo de erosión se han obtenido gráficas del área de estudio, donde podemos apreciar comparativamente las zonas donde se encuentran los principales depósitos aluviales, así como el arrastre de carbono orgánico hacia dichas zonas (Figura 2.3).

6. Realización de muestreos periódicos de la producción primaria en ambos entornos (marinos y terrestres) en la costa oceánica. Además de un seguimiento muestral de la presión parcial de  $CO_2$  (la atmósfera está compuesta de diferentes gases, con una presión total igual a la suma de las presiones parciales de cada uno los gases que la componen).
7. Generación de recursos humanos. Uno de los propósitos principales del proyect ha sido la preparación de elementos humanos, de todos los niveles académicos, en colaboración cercana con los investigadores asociados al grupo de investigación.

#### 2.4.2. Características de la zona de estudio

El área alrededor de la ciudad de Ensenada, resulta de particular interés para este tipo de estudios ya que la erosión de sus colinas ha redundado en depósitos aluviales de escurrentía y planos aluviales costeros. Las características de relieve y vegetación de esta área pueden apreciarse en la Figura 2.4. El clima de Ensenada es mediterráneo, con inviernos frescos y húmedos y veranos cálidos y secos. La precipitación promedio se estima en  $315 \text{ mm año}^{-1}$  con cerca del 85 % de la precipitación entre los meses de noviembre y abril. No hay caudales (flujos de arroyos) perennes significativos. Esto quiere decir que la mayoría de los arroyos se secan en alguna época del año y si acaso hay algún flujo, éste es muy pequeño. La temperatura anual media es de  $17 \text{ }^\circ\text{C}$ . Ya que la mayor parte de la ciudad de Ensenada y el área agrícola de Maneadero descansan sobre planos aluviales, se esperaría que dichas zonas sean sitios naturales de oxidación del carbono orgánico. Las actividades humanas han incrementado las tasas de oxidación. En el área urbana, gran parte del plano aluvial se encuentra cubierto por cemento o asfalto y ya no representa un posible ambiente de oxidación. Sin embargo, las actividades humanas importan grandes cantidades de material orgánico, el cual es oxidado a tasas que son más altas que las de oxidación natural. En medios agrícolas las actividades antopogénicas tienden a favorecer la oxidación orgánica al proporcionar oxígeno y aire al material orgánico del suelo.

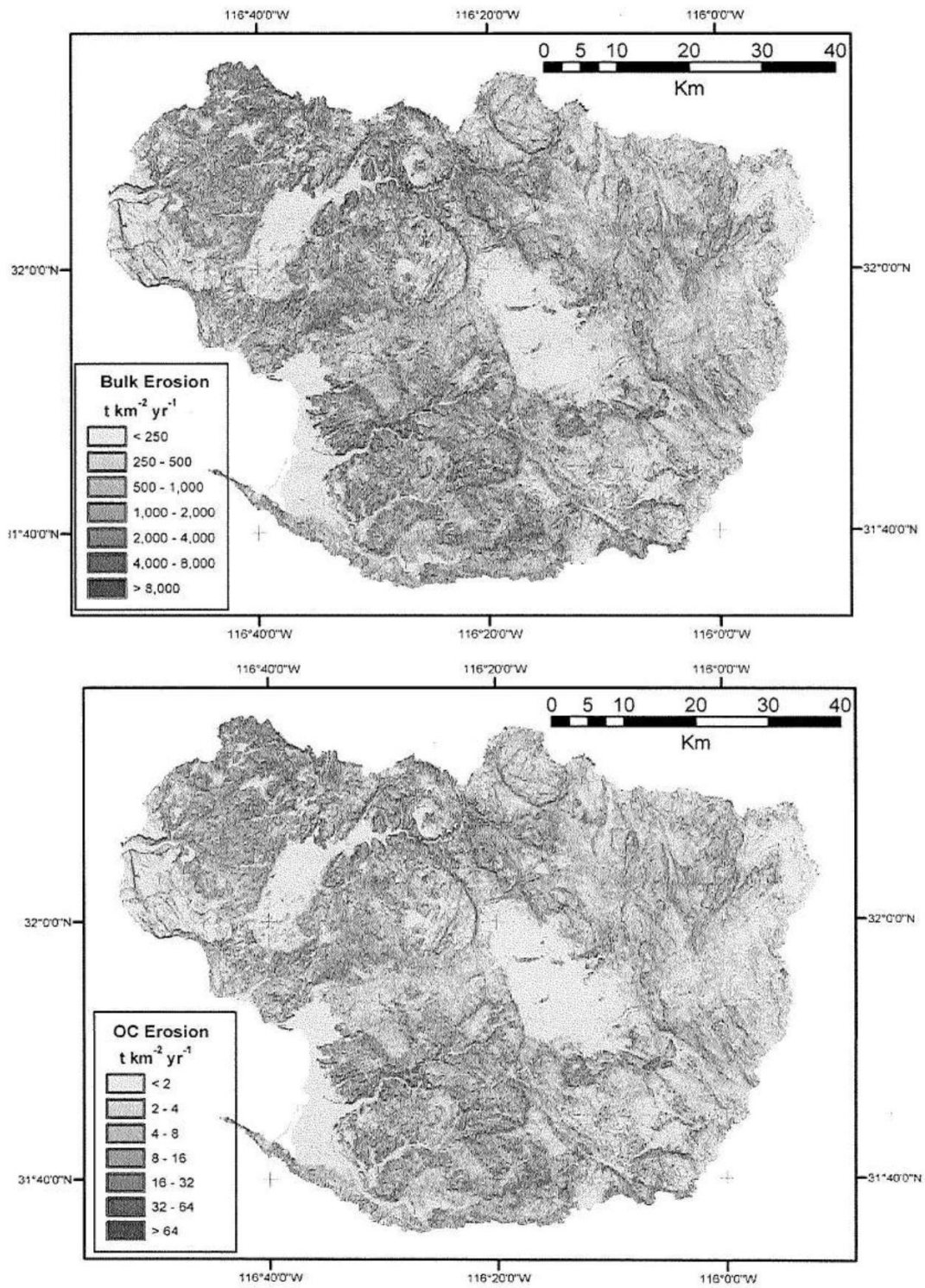


Figura 2.3: Arrastre bruto de material producto de la erosión de las cuencas (*arriba*). Erosión del carbono orgánico (*abajo*).

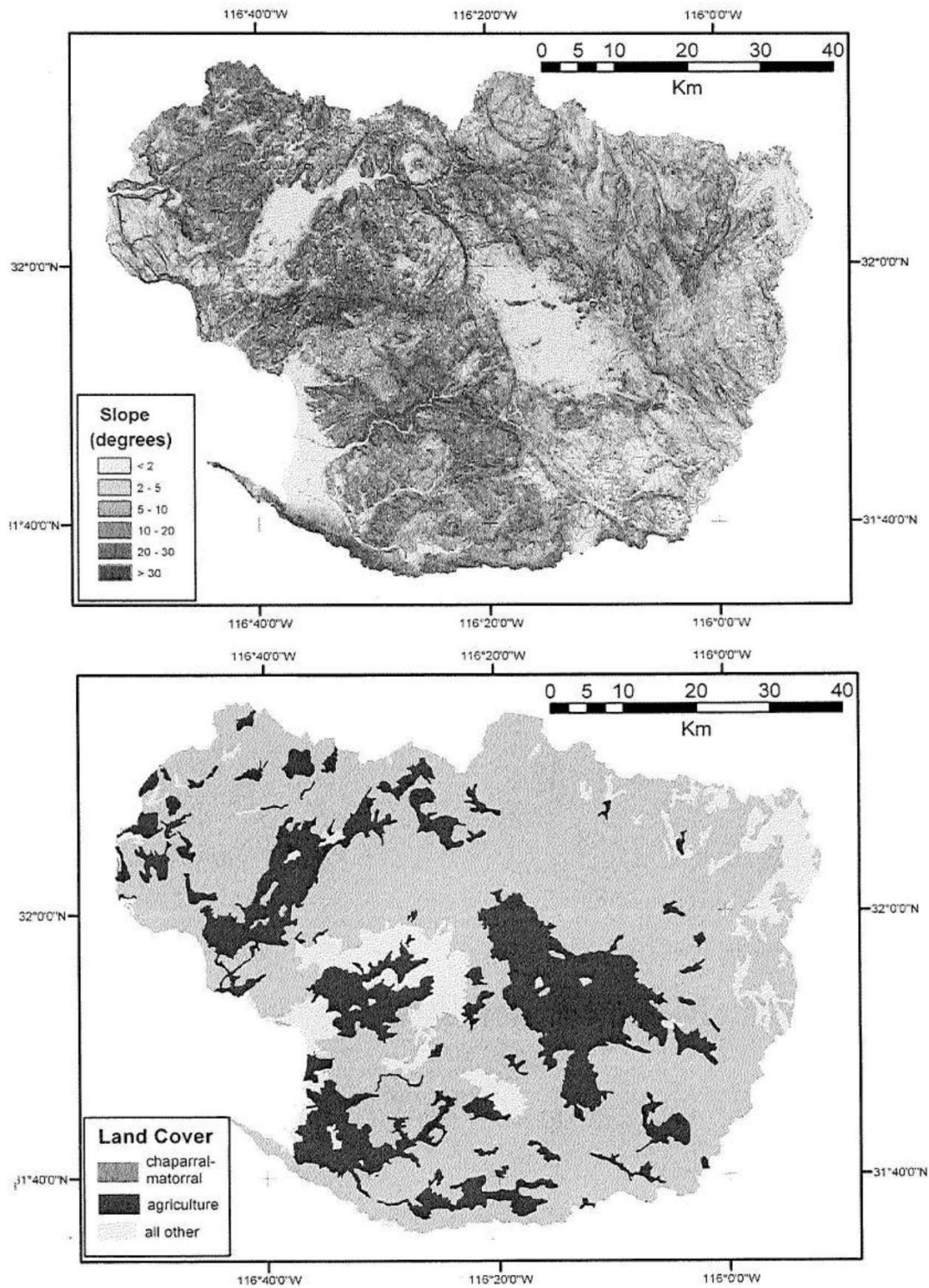
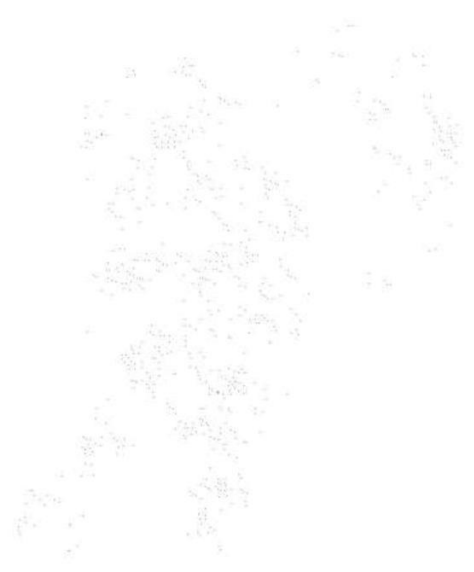


Figura 2.4: Características de relieve, se muestra en el gráfico la pendiente (arriba). La vegetación de la zona de Ensenada (abajo).



## Capítulo 3

# Sistema de medición de flujo turbulento

Con la intención de medir las variables atmosféricas y edáficas que habrán de permitir la determinación de los flujos del carbono en los sitios de estudio, son requeridos sistemas electrónicos de adquisición de datos capaces de operar de forma continua y en completa autonomía, con una tasa de muestreo variable y con amplia capacidad de almacenamiento de datos. Estas estaciones también deberán ser capaces de operar en condiciones meteorológicas desfavorables y contar con suficiente energía de reserva para evitar la pérdida de datos. Así mismo, deberán ser reprogramables y poder realizar las operaciones estadísticas básicas así como funciones de calibración por software y compatibilidad con sensores analógicos y digitales. El alcance del presente trabajo es congruente a la adquisición, almacenamiento y procesamiento básico de variables meteorológicas y edáficas, lo cual permitirá además la determinación de los flujos verticales de  $CO_2$ . Todo ello dentro del marco conceptual del proyecto "Flujos TERMAR: Enumerating  $CO_2$  sources and sinks in a regional coastal terrestrial and marine habitat matrix" del Laboratorio de Ecología Terrestre del CICESE en colaboración con el Departamento de Instrumentación del Instituto de Astronomía de la UNAM, en Ensenada, para el desarrollo de los módulos de adquisición aquí descritos.

El sistema desarrollado está compuesto por una tarjeta de adquisición de datos, una tarjeta de acondicionamiento de señales y un conversor RS-485/RS-232. Además de la programación necesaria para el procesamiento de los datos por una microcomputadora y su almacenamiento en una tarjeta de memoria. A continuación se

describe el sistema completo de adquisición y procesamiento de datos.

### 3.1. Microcontrolador ADuC848

El núcleo del sistema de adquisición de datos es un ADuC848, de la compañía Analog Devices, el cual es programable en el lenguaje de alto nivel C++ mediante un compilador externo otorgado por el fabricante. Este microcontrolador cuenta con 24 líneas programables de entrada/salida, un puerto de 8 entradas analógicas con 16 bits de resolución (con rango dinámico de 2.5 V) permutable a puerto de entrada/salida de 8 bits. Este dispositivo puede operar a una tasa de 12.58 MIPs. Dispone además de 4 kB en memoria de datos, un DAC con salida de voltaje con resolución de 12 bits, dos DACs Sigma-Delta de 16 bits para las salidas de modulación de ancho de pulso y 3 contadores de 16 bits cada uno. Cuenta también con puertos seriales UART, SPI e I2C. Este microcontrolador utiliza un voltaje de alimentación de 5V y requiere un cristal de cuarzo de 32,768 kHz con el que por medio de un lazo de amarre de fase (PLL) integrado genera una frecuencia de 12.582912 MHz. El reloj del núcleo es el del PLL dividido por 8, es decir 1.572864 MHz. El microcontrolador utiliza un núcleo 8052, que le permite alcanzar 12.58 MIPs manteniendo compatibilidad con el juego de instrucciones del 8051. Dispone de 2304 bytes de RAM integrados. La memoria de programa del ADuC848 puede ser configurada como memoria de datos para permitir hasta 60 kB de memoria no-volátil. El *firmware* del dispositivo permite su programación sin necesidad de ser desmontado del circuito (el software es descargable de forma serial vía UART).

Los contadores en un 8052 estándar se incrementan en uno cada ciclo de máquina, el ADuC848 no es la excepción de tal forma que los contadores se pueden incrementar a la misma tasa del reloj. A continuación se muestran algunas de características relevantes del microcontrolador.

- El ADuC848 es capaz de acceder hasta 16 MB de memoria RAM externa.

Su memoria RAM interna se encuentra dividida en dos memorias, de 128 bytes cada una. La parte baja puede ser accesada mediante direccionamiento directo o indirecto mientras que la parte alta solo puede ser accesada mediante direccionamiento indirecto.

Contiene 2 kB de memoria XRAM integrada. Esta memoria es accesada mediante la instrucción MOVX.

Esta memoria interna XRAM son mapeados a la parte baja del espacio de memoria externa si el registro CFG848.0 es puesto en alto. El CFG848 SFR contiene los bits necesarios para configurar la memoria interna XRAM y el *Stack Pointer* (SP). El microcontrolador se encuentra configurado por defecto para el usuario en modo 8051, sin embargo en este modo el SP y la XRAM están deshabilitados.

- En el presente trabajo ha sido utilizado un ADuC848 en empaquetado MQFP, el cual posee 8 canales de conversión analógica a digital (ADC) de 16 bits e incluye un amplificador de ganancia programable. Puede ser configurado para operar como 4 entradas de voltaje diferencial u 8 entradas pseudo-diferenciales. Su ganancia puede ser programada para uno de ocho rangos de entrada de  $\pm 20$  mV a  $\pm 2.56$  V ( $V_{REFx1,024}$ ). Con esto es posible recibir señales directamente de los sensores sin la necesidad de acondicionamiento de señal (sin los reforzadores internos habilitados el reforzamiento externo es requerido). En el modelo ADuC848 el ADC es capaz de realizar conversiones a una tasa de 20Hz. El modulador  $\Sigma$ - $\Delta$  convierte la señal muestreada en un tren de pulsos digital que contiene la información.

El rango de voltaje en modo común para las referencias de voltaje diferencial va de tierra analógica (AGND) a voltaje analógico ( $AV_{DD}$ ). Cuando se utiliza una referencia de 2.5V el ADC primario reconoce a ésta como una referencia de 2.56V ( $V_{REFx1,024}$ ). Cuando se calcule el valor del bit menos significativo (LSB) se debe considerar este efecto. Es posible también utilizar la referencia interna con que cuenta el dispositivo (con un voltaje de 1.25V), sin embargo, una consecuencia de utilizarla es una apreciable degradación en la resolución *pico-pico*. Por esta razón se recomienda la operación del sistema con una referencia de voltaje externa.

Las entradas de referencia proveen una alta impedancia y carga dinámica a las conexiones externas. Debido a que la carga sobre cada entrada de referencia es dinámica, combinaciones resistencia/condensador en esta terminal pueden provocar un error en *dc*, dependiendo de la impedancia de salida de la fuente que controla las entradas de referencia. En caso de bajas impedancias de salida en la referencia de voltaje, es requerido el empleo de condensadores de desacople (típicamente de 0.1  $\mu$ F) en las entradas  $REFIN\pm$  o  $REFIN2\pm$ , dependiendo del modo empleado. Cuando se utiliza una fuente de voltaje de alta impedancia, este condensador no es requerido.

- Es posible utilizar los 62 kB de memoria de programa como un solo bloque de memoria. En este modo

toda la memoria es de solo-lectura para el código del usuario. Sin embargo una gran cantidad de esta memoria puede ser escrita durante la de ejecución mediante el modo ULOAD, que permite que los 56 kB de la parte baja de la memoria puedan ser borrados y reprogramados. Los 6 kB restantes son programables vía descarga serial o paralela. Esto quiere decir que este espacio aparece como de sólo-lectura para el código de usuario, por lo cual no puede ser borrado accidentalmente o reprogramado durante ejecución, manteniéndolo seguro. Este código es descargado el dispositivo utilizando el software Windows©Serial Downloader (WSD) que provee el fabricante como parte de su sistema de desarrollo *Quick Start*.

- El voltaje de alimentación puede estar entre 2.7 y 5.25V. Las fuentes de voltaje analógico y digital ( $AV_{DD}$  y  $DV_{DD}$ , respectivamente) permiten reducir el ruido electrónico inducido en las líneas del sistema y especialmente sobre la línea de  $DV_{DD}$  (ver Figura 3.1). De esta forma el circuito puede ser diseñado para operar con dos voltajes diferentes si es requerido.
- En sistemas que operan a altas velocidades, así como en aquellos que poseen convertidores de alta resolución, debe otorgarse especial atención a la forma en que son aterrizados los componentes sobre el circuito impreso con la finalidad de obtener un desempeño óptimo. Aunque el microcontrolador tiene terminales individuales para las tierras analógica y digital el usuario no deberá conectar planos separados a menos que los una en un solo punto muy cerca del microcontrolador, ya que si se unen en otros lugares se producirán lazos de tierra. Los diagramas de conexión de planos se muestran en la Figura 3.2a.

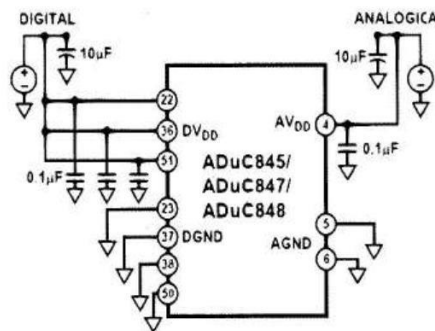


Figura 3.1: Configuración con fuente de voltaje analógica y digital

En sistemas donde la tierra analógica y digital se encuentran unidos en algún otro lugar sobre la placa, estas no pueden ser conectadas de nuevo cerca del microcontrolador debido a que esto produce un lazo de tierra. En estos casos deberán conectarse las terminales AGND y DGND a la sección del plano de tierra analógica, como se muestra en la Figura 3.2b. En sistemas con solo un plano de tierra deberán ser separados físicamente los componentes analógicos de los digitales con el objetivo de evitar que los retornos de corriente fluyan cerca de la circuitería analógica y viceversa. Es recomendado colocar el microcontrolador en la sección de componentes analógicos (ver Figura 3.2c). Debe evitarse alimentar componente del lado analógico mediante el empleo de  $DV_{DD}$  ya que esto forzará corrientes de retorno hacia la parte analógica del sistema a través de AGND. Todas las conexiones deben hacerse directamente sobre los planos de tierra, evitando el uso de pistas delgadas para unir las terminales a tierra.

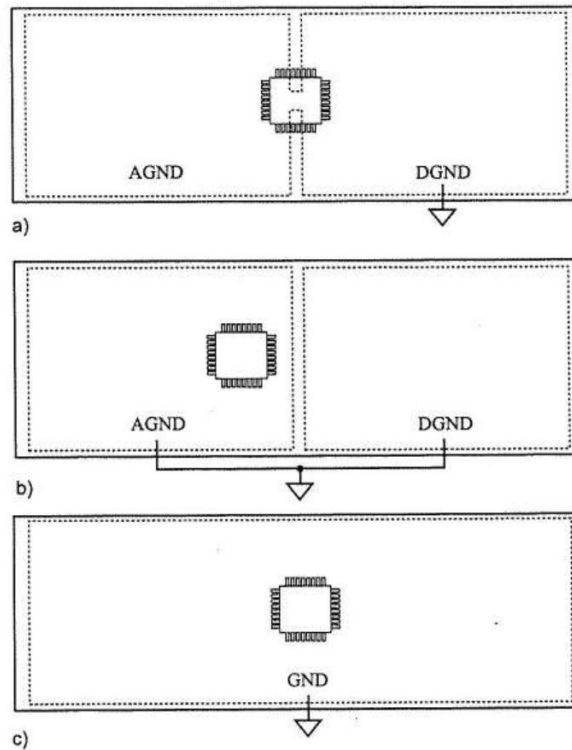


Figura 3.2: Esquemas de planos de tierra

### 3.2. Protocolos de comunicación y circuito de conversión EIA-485/EIA-232

Dependiendo de las necesidades del usuario la información debe transmitirse a través de un medio, en la mayoría de los casos se trata de un cable con características determinadas por la aplicación a la que está destinado. Los fabricantes proveen el cable específico para la aplicación de interés, la cual depende además de la frecuencia de trabajo del sistema. En la Figura 3.3 se presenta un esquema en el cual puede observarse el comportamiento de diferentes interfaces de comunicación entre dispositivos electrónicos de acuerdo a la tasa de transferencia de datos y la longitud máxima del cable que dicha interfaz puede alcanzar. Existen

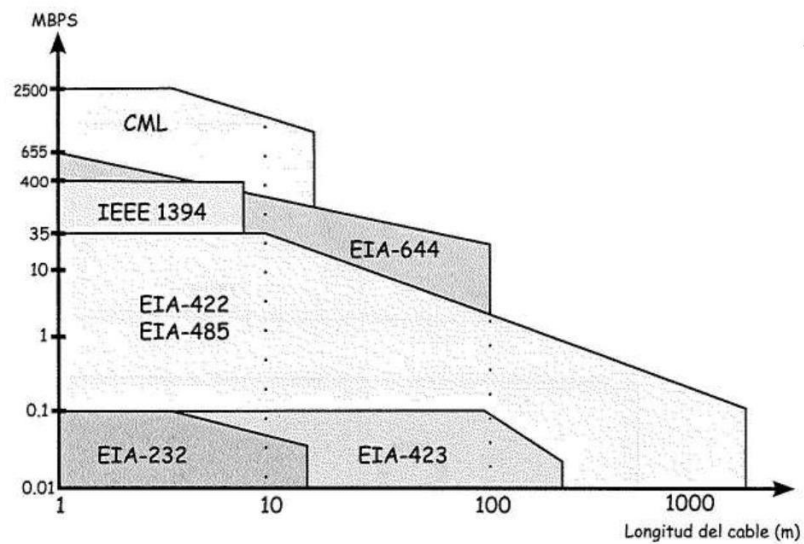


Figura 3.3: Comparación de estándares para la transmisión de datos.

otras características que pueden poner un protocolo en ventaja sobre otro aún en condiciones similares de operación. Para el caso específico del presente trabajo el protocolo EIA-485 es de gran utilidad tanto por su tasa de transmisión, como por la posibilidad de tener varios dispositivos compartiendo un mismo par de alambres, capacidad que el protocolo EIA-232 no posee, ya que este último opera en el modo punto-a-punto, es decir, el bus solo puede ser usado por dos elementos: uno que envía y otro que recibe. La elección del protocolo de comunicación juega un papel relevante respecto a su compatibilidad con otros sistemas de uso comercial.

### 3.2.1. Estándar EIA-232

La Electronics Industry Association (EIA) ha desarrollado estándares para transmisión de datos. Estos estándares fueron originalmente marcados con el prefijo *RS* (Recommended Standard). Actualmente son generalmente indicados a través del prefijo *EIA*. RS-232 fue introducido en 1962, en el caso del protocolo RS-232E la letra adicional indica que se trata de la quinta revisión.

#### Descripción

Las especificaciones EIA-232 definen las características mecánicas, eléctricas y funcionales. RS-232 es una interfaz no balanceada (*single ended*), unidireccional (*point-to-point*). La señal es referenciada a tierra. Los niveles de voltaje sobre el cable RS-232 son entre -5V y -15V, para el nivel inferior y +5V a +15V para el nivel superior, mientras que en el dispositivo receptor estos niveles deben ser limitados al intervalo -3V a +3V, como se muestra en la Figura 3.4. La interfaz EIA-232 es *síncrona* cuando los relojes son utilizados (DA/DB), de otra forma se dice que es *asíncrona*. Utiliza la codificación *NRZ*, donde los niveles de voltaje del canal oscilan de un valor positivo a uno negativo, sin permanecer nunca en cero. Es capaz de operar a tasas de hasta 20kbps.

La longitud máxima del cable no está bien definida, ésta depende de la capacitancia máxima de línea, de 2500pF, con una impedancia de carga de 3k a 7k Ohms. Esta combinación usualmente permite utilizar cables de hasta 20 metros pero se recomienda trabajar con longitudes no superiores a 10 metros para minimizar el ruido inducido sobre el cableado que depende de su longitud y la frecuencia de trabajo.

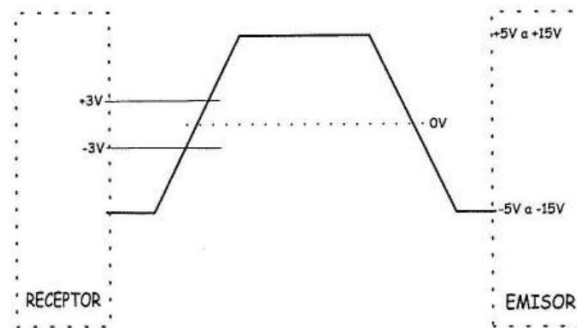


Figura 3.4: Características eléctricas del RS-232. Codificación NRZ. Intervalos del voltaje de transmisión en el alambre conductor.

Normalmente la información es enviada en serie en palabras de 7 u 8 bits, iniciando con el bit menos significativo (LSB). Un bit de inicio determina el inicio de la secuencia de transferencia, véase Figura 3.5. El bit de inicio es activo bajo (los manejadores RS-232 invierten los niveles de voltaje, de tal manera que en el cable RS-232 es observado como un activo alto, con una amplitud de entre +5V y +15V). La palabra de 8 bits sucede al bit de inicio. Un nivel lógico alto aparece en el bus como un nivel de voltaje entre -5V y -15V. Un bit de paridad puede suceder a la palabra dato, dependiendo del protocolo usado (según sea convenido por el usuario). Un bit de término sigue generalmente a la cadena (o al bit de paridad, si éste es utilizado). El bit de término es utilizado para asegurar que la señal permanezca en un nivel lógico alto una vez que la transmisión haya concluido (recuerde que será invertido, así que en el cable el bit de término es un nivel bajo de voltaje y el bit de inicio es una transición de bajo a alto).

### Características funcionales y mecánicas

Los dispositivos con interfaz serial pueden ser conectados a través de cables con conectores tipo D de 9 ó 25 terminales. Existe terminología diversa para nombrar estos conectores, los más comunes son DB9, DB25, CANNON 9 y CANNON 25 respectivamente. Son fabricados en modelos hembra y macho, según las necesidades físicas presentes en el sistema empleado por el usuario. La distribución de terminales es la que se muestra en la Tabla 3.2.1 y la Figura 3.6.

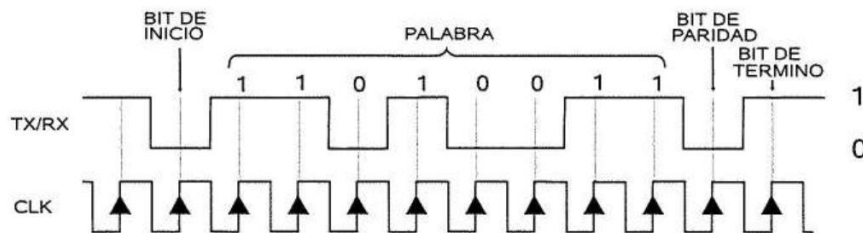


Figura 3.5: Características eléctricas del RS-232. Codificación NRZ.

### Handshaking

El control de flujo por hardware (*HW Handshaking*) es conocido también como control de flujo RTS/CTS. Utiliza dos cables extras del cable de conexión serial, en lugar de caracteres extra en la cadena de datos de transmisión. El control de flujo por hardware no contribuye a la disminución de la velocidad del sistema

PIN	Señal	Descripción
1	CD	Detección de portadora
2	RXD	Recepción
3	TXD	Transmisión
4	DTR	Terminal de datos lista
5	GND	Tierra
6	DSR	Paquete de datos listo
7	RTS	Solicitud de envío
8	CTS	Listo para enviar
9	RI	Indicador de marcado

Tabla 3.1: Descripción de terminales en el conector DB9

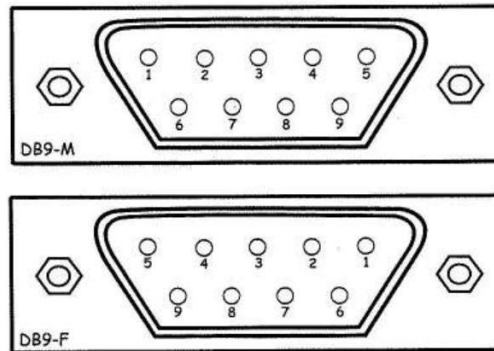


Figura 3.6: Distribución de terminales en el conector DB9

como en el caso de control por Xon-Xoff. Cuando la computadora desea enviar datos debe activar la línea de RTS, si el dispositivo receptor cuenta con capacidad de aceptación responderá activando la línea CTS y la computadora iniciará la transmisión de datos. Si el dispositivo aceptor no tiene capacidad de aceptación, este no activará su línea de CTS.

El control de flujo por software (*SW Handshaking*), algunas veces expresado como Xon-Xoff usa dos caracteres ASCII, 17 y 19 para Xon y Xoff respectivamente, para indicar a la computadora cuando el buffer se encuentra disponible para almacenamiento, de esta forma controla el flujo de información según su capacidad

de recepción. Para los fines de este trabajo ha sido requerida la utilización del sistema de control del flujo por hardware. En ningún caso se ha utilizado el control Xon-Xoff durante este trabajo.

### 3.2.2. Estándar EIA-485

#### Señales diferenciales con RS-485: Mayor distancia y mayor velocidad de transmisión

Uno de los problemas con el RS-232 representa su baja inmunidad al ruido inducido sobre las líneas de transmisión, ver Figura 3.7. Los dispositivos de transmisión y recepción comparan sus niveles de voltaje de información y de *handshake* con una línea de tierra común. Niveles de *offset* sobre el voltaje de tierra pueden tener efectos de pérdida de la información. El nivel de voltaje respecto al cual la señal puede ser considerada como información es relativamente alto, de 3V. El ruido puede ser fácilmente inducido en la línea cuando el cable, de acuerdo al protocolo utilizado, se aproxima a su longitud máxima o a su velocidad de comunicación límite.

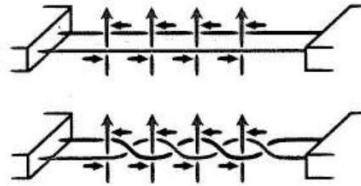


Figura 3.7: Campo magnético (vertical), corriente inducida de ruido (horizontal)

En el caso de la comunicación con RS-485 no existe un cero común o señal de referencia. Las señales se encuentran flotando y cada señal es transmitida a través de las líneas TX+ y TX- (también nombradas A y B). En un sistema diferencial balanceado el voltaje es detectado a partir de un par de líneas de señal, A y B, que transmiten una sola señal. El receptor RS-485 compara los voltajes entre ambas líneas, en lugar del voltaje absoluto respecto a la tierra común de una señal. Esto previene la existencia de lazos de tierra, un problema común en sistemas de comunicación. Si el voltaje diferencial  $V_{AB}$  es mayor a +200mV el receptor fijará un estado lógico específico en su terminal de salida. Si al contrario, el voltaje es menor a -200mV, el receptor fijará en su salida el estado lógico contrario.

El estándar RS-485 permite la transmisión de datos a través de una red multipunto con capacidad para

conectar hasta 32 pares emisor-receptor. El rango de voltaje en modo común que el dispositivo emisor y receptor puede tolerar se encuentra entre +12V y -7V. Se recomienda utilizar par de cable trenzado. La topología recomendada consiste en una *cadena de margaritas* como se esquematiza en la Figura 3.8. En esta conexión todos los sensores se encuentran conectados al mismo par de cables, un dispositivo se conecta a otro mientras que el cable continúa hacia el siguiente sensor y así sucesivamente.

Las redes RS-485 sobre un par de cable operan en modo *half-duplex*: la información viaja solo en una dirección a la vez. Todos los dispositivos en la red pueden ser emisores o receptores, pero solo uno puede transmitir en un tiempo dado. Esto significa que el sistema debe tener un dispositivo maestro. El dispositivo maestro dirige la red mientras que los esclavos responden. Los esclavos son normalmente receptores mientras que el maestro es normalmente emisor. Todos los dispositivos reciben la información transmitida por el canal, sin embargo solo el dispositivo al que va dirigida es quien responde. Los esclavos están normalmente *escuchando* el canal en espera de recibir su dirección, periodo durante el cual el dispositivo se encuentra en modo de alta impedancia. Cuando el dispositivo escucha su dirección conmuta del estado de recepción al de transmisión. El lector se convierte en un transmisor RS-485, espera a que el controlador (el maestro) conmute al estado de recepción, y entonces trasmite la información en la red multipunto.

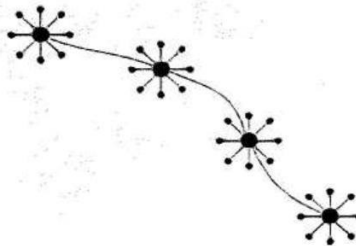


Figura 3.8: Cadena de margaritas

La red RS-485 debe ser terminada mediante una resistencia de 120 Ohms conectada entre las terminales TX+ y TX- del dispositivo más alejado al maestro. La finalidad de este procedimiento es la de evitar la reflexión de señales en el bus. Cuando un dispositivo no se encuentra transmitiendo la línea TX+ debe estar al menos +200mV sobre la línea TX-. Esto quiere decir que la diferencia entre las terminales A y B debe ser de al menos 200mV.

### 3.2.3. Convertidor EIA-232/EIA-485

Cada dispositivo RS485 es a la vez transmisor y receptor, este tipo de elemento recibe el nombre de *transceptor*. Sobre un bus RS485 puede haber varios dispositivos, sin embargo, solo uno puede transmitir a la vez, lo cual recibe el nombre de transmisión *half-duplex*. El circuito que se presenta en la Figura 3.10 ha sido probado e implementado en el Observatorio Astronómico Nacional del Sierra de San Pedro Mártir y adaptado para los propósitos del presente trabajo. Este dispositivo convierte las líneas RS232 en señales RS485 half-duplex balanceadas. La configuración mostrada no toma en cuenta el estado de las líneas RTS y CTS. Permite comunicación half-duplex sobre un par de cables solamente y es capaz de habilitarse automáticamente durante periodos de transmisión. Durante este mismo periodo la recepción es deshabilitada para evitar rebotes de la señal sobre el dispositivo RS232.

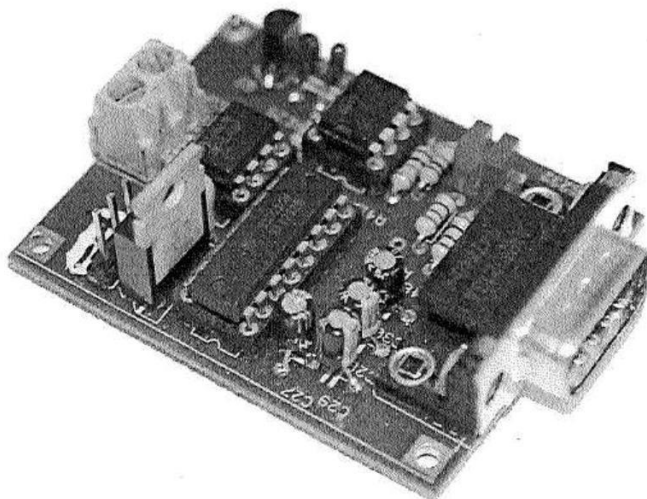


Figura 3.9: Conversor RS-232/RS-485 construido en el IA-UNAM

Debe asegurarse que la línea TX+ sea al menos 200mV superior a TX-, por lo tanto, es recomendado que al final del bus sea colocada una resistencia de terminación de 120 Ohms. En el extremo terminal del bus deberá colocarse además un resistencia, de valor ajustable por el usuario (560 a 4.7k Ohms) según los requerimientos del sistema individual, de *pull-up* a un valor positivo de voltaje (normalmente +5V), mientras que el otro extremo de la resistencia de terminación deberá ser conectado a tierra mediante una resistencia del

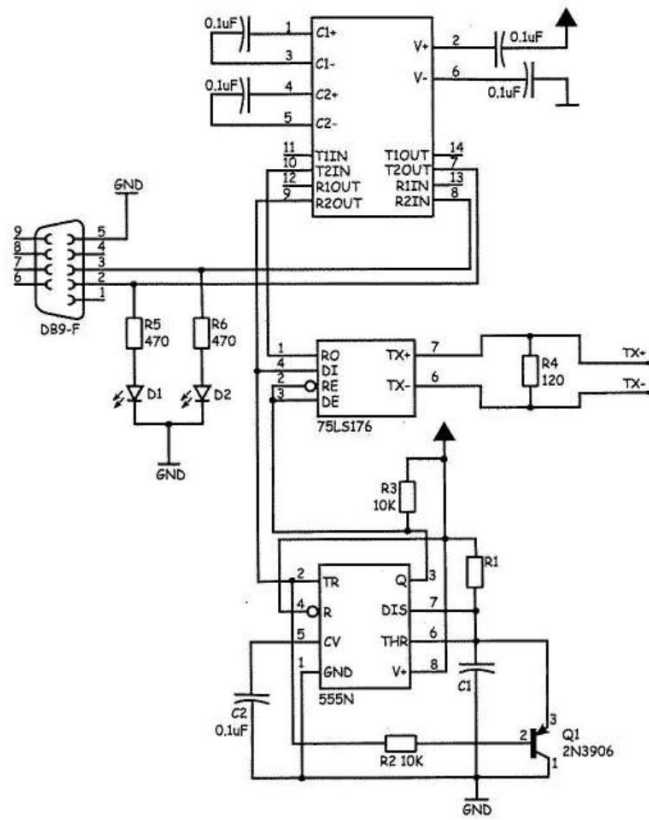


Figura 3.10: Convertidor RS-232/RS-485

mismo valor. Esto permitirá mantener un estado lógico definido en el bus cuando el total de dispositivos se encuentren en estado de reposo, conocido también como de alta impedancia.

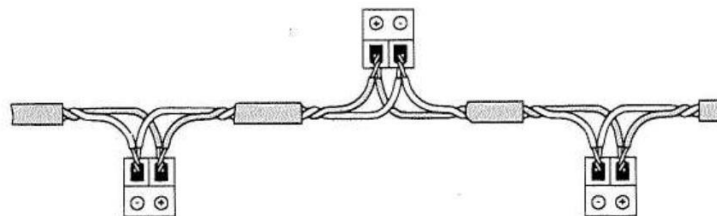


Figura 3.11: Conexión física de la cadena de margaritas

El convertidor RS-232/RS-485 (ver Figura 3.9) conectado a un PC o dispositivo análogo constituye lo que en el bus RS-485 es considerado el maestro; el resto de los dispositivos en la cadena de margaritas son esclavos, mientras que en el extremo contrario de la cadena el resistor de terminación debe ser colocado. No debe ser empleada utilizar la configuración de estrella (conectar varios dispositivos a un mismo par de terminales).

### 3.3. Módulos de adquisición de datos y acondicionamiento de señales

El proceso de adquisición de datos se ha separado en dos etapas: acondicionamiento de señales y adquisición de datos. Por medio de la tarjeta de acondicionamiento las señales provenientes de los sensores son ajustadas al rango dinámico de los conversores ADC del microcontrolador ADuC848 (2.5V), mientras que por medio de la tarjeta de adquisición de datos pueden ser leídos estos voltajes, convertidos al formato digital y filtrados y promediados. Esta última tarjeta puede transferir datos bidireccionalmente con una computadora o un sistema empotrado a través de los estándares EIA-232 y EIA-485. Debe hacerse notar que la tarjeta de adquisición puede llevar a cabo el proceso de muestreo sin necesidad de la tarjeta de acondicionamiento para aquellos casos donde los voltajes de salida de los sensores empleados no deban ser ajustados. En la Figura 3.12 se muestra el diagrama a bloques del sistema de adquisición de datos.

#### 3.3.1. Tarjeta de adquisición de datos

El diagrama mostrado en la Figura 3.14 presenta los elementos básicos que componen el sistema de adquisición de datos, integrado por el microcontrolador ADuC848 y el sistema mínimo requerido para que éste funcione apropiadamente. Mientras que en la Figura 3.15 es posible observar los puertos de los cuales ha sido dotado. Las fuentes de voltaje analógico y digital, conexiones de tierra, filtros de acoplamiento y otros aspectos han sido obviados en esta parte del documento, sin embargo el diagrama electrónico completo se muestra en el apéndice A.

El sistema ha sido integrado de forma modular lo cual permite el reemplazo de alguna tarjeta dañada, sin detener la operación del resto del sistema. Cada módulo debe ser provisto de su propia dirección a través de uno de los puertos del módulo (puerto de asignación de dirección Figura 3.15), la cual se establece a través un DIP switch (S1), evitando que la dirección asignada sea utilizada por los otros módulos compartiendo la red RS-485. A través del puerto digital se permite la comunicación con sensores que manejan niveles típicos

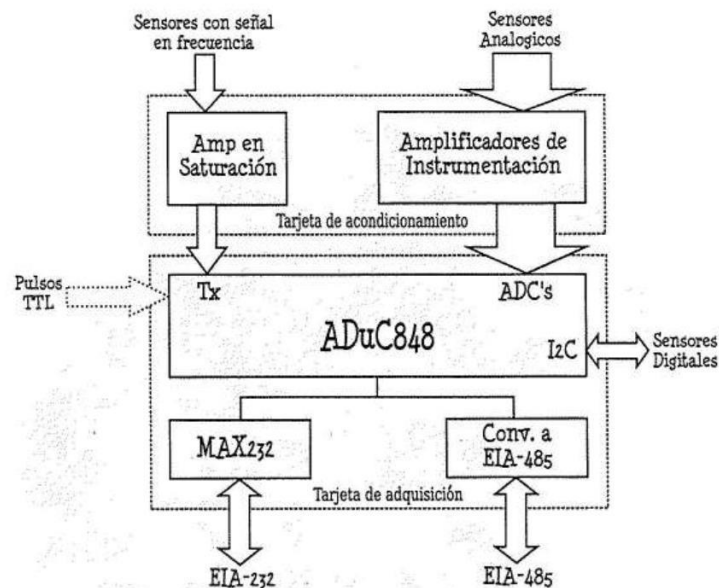


Figura 3.12: Diagrama a bloques del módulo de adquisición de datos

TTL. Aunque para esta aplicación ha sido específicamente acondicionada para leer sensores digitales bajo el protocolo I2C. Las terminales de conexión para RS-232, RS-485 y de entrada principal de alimentación al sistema (9V a 30V, se recomienda el uso de 12V) han sido concentrados en el puerto digital, lo cual permite una baja cantidad de partes móviles, disminuyendo así los ruidos de línea producto de vibraciones mecánicas propias de conectores de terminales de opresión. También se muestra el puerto analógico, el cual permite la conexión a la tarjeta de acondicionamiento de señales para los casos donde es requerida, en este mismo conector han sido habilitadas tres terminales para contadores digitales en aplicaciones específicas. Se ha agregado además un puerto suplementario que contiene terminales que para las presentes aplicaciones no han sido implementadas pero aprovechables para aplicaciones futuras.

Ha sido empleada una referencia de voltaje de precisión de 2.5V (LM336-2.5), cuya función es proveer al sistema de un nivel de voltaje preciso y de bajo nivel de histéresis. La línea RESET puede ser activada presionando el interruptor normalmente abierto SPT1. Para su funcionamiento el dispositivo requiere de un cristal de cuarzo de 32.768 kHz conectado entre las líneas XTAL1 y XTAL2. La línea \*EA ha sido permanentemente referenciada a voltaje, lo que permite al dispositivo acceder los 62 kB disponibles de

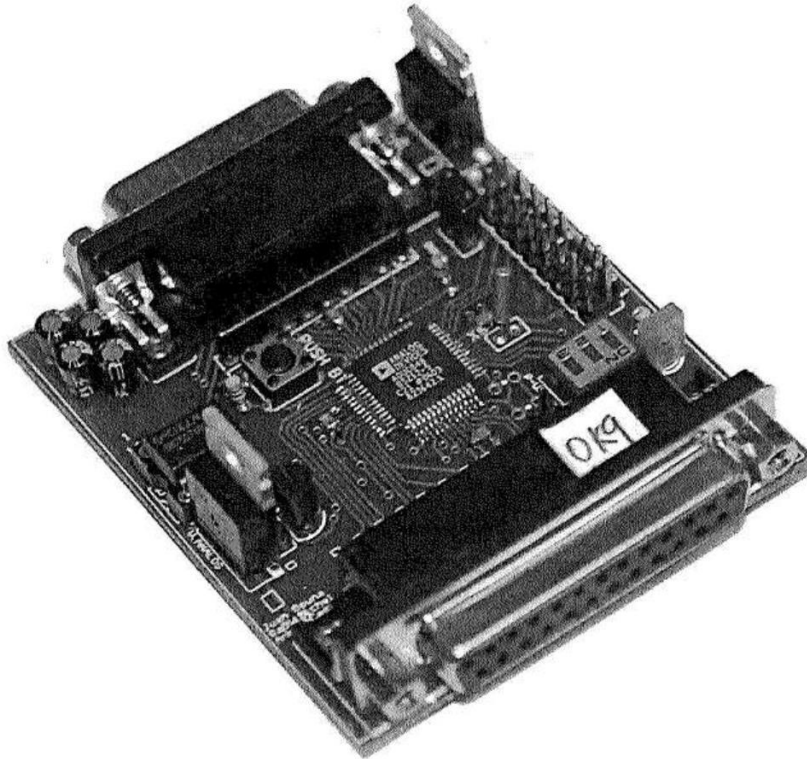


Figura 3.13: Tarjeta de adquisición de datos de diseñada y construída para las torres de correlación turbulenta. Es un sistema cuyas característica le permiten ser empleado en una gran variedad de aplicaciones.

memoria Flash/EE de programa.

### 3.3.2. Selector RS-232/RS-485

Una gran cantidad de sistemas digitales basados en microprocesador utilizan un elemento en hardware para convertir los datos de formato paralelo a serie para realizar comunicación con dispositivos periféricos. También realiza el proceso inverso. A este elemento se le denomina UART (*Universal Asynchronous Receiver-Transmitter*). El UART es comunmente utilizado asociado a otros estándares de comunicación como el EIA-232. Actualmente un UART se encuentra integrado en la mayor parte de los microcontroladores comerciales.

Para evitar el problema que implica enviar señales a través de largos cables en paralelo, los bits de datos son

No. de terminal	Nombre	Descripción	Sentido
1	SCL3	CLK del Bus 3 I2C	→
2	SDA3	SDA del Bus 3 I2C	↔
3	SCL2	CLK del Bus 2 I2C	→
4	SDA2	SDA del Bus 2 I2C	↔
5	SCL1	CLK del Bus 1 I2C	→
6	SDA1	SDA del Bus 1 I2C	↔
7	SCL0	CLK del Bus 0 I2C	→
8	SDA0	SDA del Bus 0 I2C	↔
9	NET_A	TX+ (EIA-485)	↔
10	NET_B	TX- (EIA-485)	↔
11	GND	GND (EIA-232)	
12	PC/TX	TX (EIA-232)	←
13	PC/RX	RX (EIA-232)	→
14	GND	GND General (Alimentación)	
15	12V	Voltaje de Alimentación	

Tabla 3.2: Asignación de líneas en el puerto digital (conector DB15)

No. de terminal	Nombre	Descripción	Sentido
1	T2	Contador T2	←
2	T1	Contador T1	←
3	T0	Contador/Temporizador T0	←/ →
4	AVDD	5V (Analógicos)	→
5	AVDD	5V (Analógicos)	→
6	AI8	Canal 7 del ADC	←
7	AI7	Canal 6 del ADC	←
8	AI6	Canal 5 del ADC	←
9	AI5	Canal 4 del ADC	←
10	AI4	Canal 3 del ADC	←
11	AI3	Canal 2 del ADC	←
12	AI2	Canal 1 del ADC	←
13	AI1	Canal 0 del ADC	←
14	12V	12 V (Alimentación)	
15	GND	Tierra general (Alimentación)	
16	AGND	GND Analógica	
18	AGND	GND Analógica	
19	AGND	GND Analógica	
20	AGND	GND Analógica	
21	AGND	GND Analógica	
22	AGND	GND Analógica	
23	AGND	GND Analógica	
24	AVDD	5V Analógicos	→
25	AVDD	5V Analógicos	→

Tabla 3.3: Líneas asignadas al puerto analógico (conector DB25)

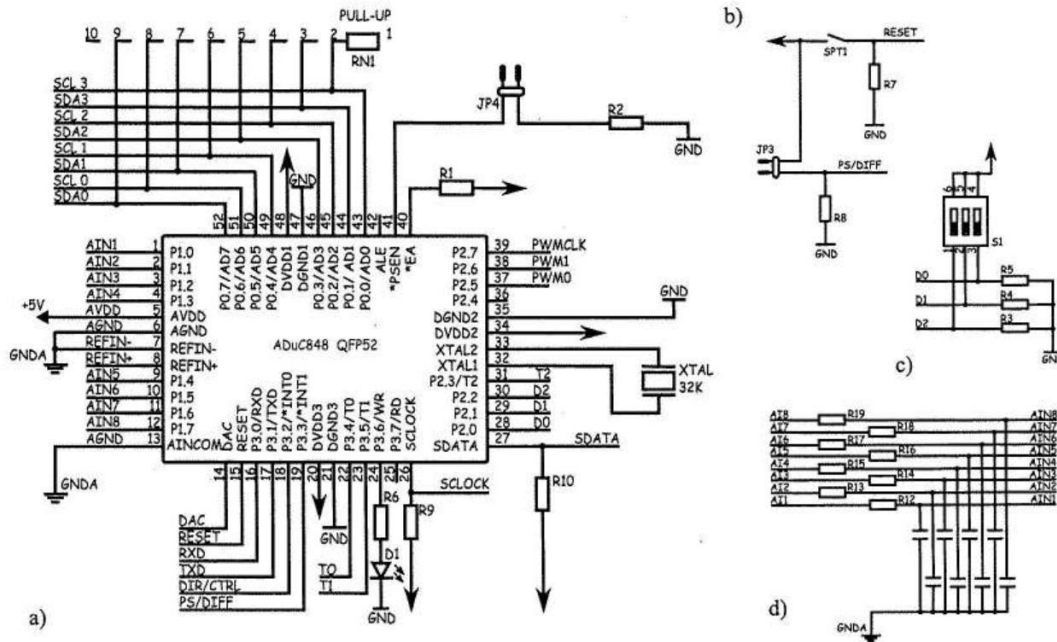


Figura 3.14: Sistema básico.

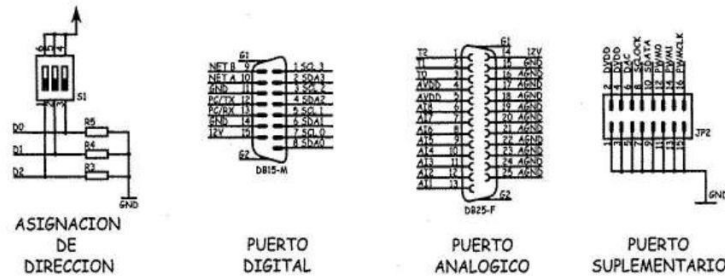


Figura 3.15: Puertos de la tarjeta de adquisición de datos.

enviados secuencialmente. Por convención el UART envía un bit de inicio, de 5 a 8 bits de información donde el LSB es el primero en ser enviado, un bit opcional de paridad y finalmente uno o dos bits de término. El UART no genera o recibe los niveles externos de señalización directamente, para esto se emplea una interfaz para convertir los niveles de señalización del UART a los niveles de señalización eléctrica externos. Ejemplos de estos estándares son EIA-232, EIA-422 y EIA-485.

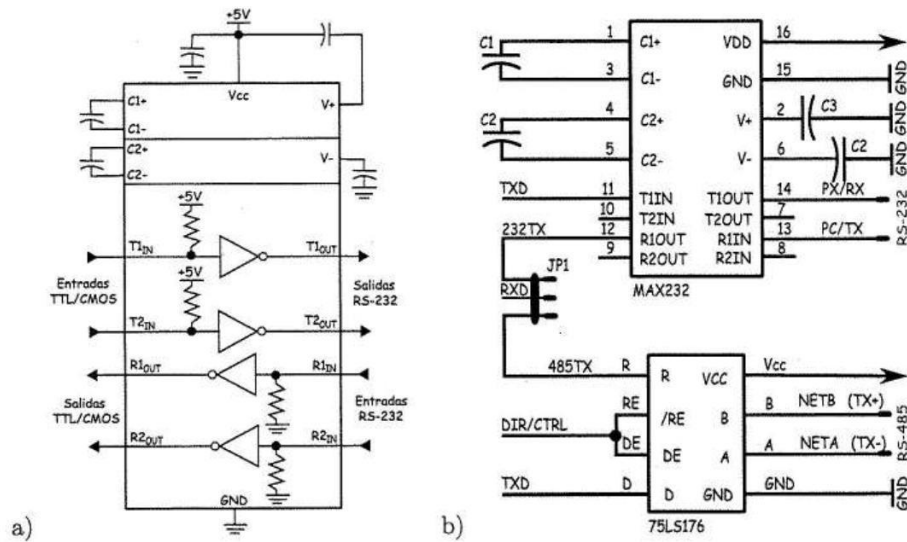


Figura 3.16: (a)MAX232, (b)Selector RS-232/RS-485.

El MAX232 es un circuito integrado de la compañía Maxim Dallas que cambia los niveles TTL a los del estándar EIA-232 para la transmisión, mientras que cambia los niveles RS-232 a TTL para la recepción. El esquema básico es el que se muestra en la Figura 3.16a. La tarjeta de adquisición ha sido provista de la capacidad de transmitir información con los estándares EIA-232 ó EIA-485, el cual es definido mediante la conmutación física del *jumper* JP1(ver Figura 3.16b y Tabla 3.4). Aunque puede transmitir y recibir con cualquiera de los protocolos, la programación del microcontrolador solo es posible a través del EIA-232.

El SN75LS176 permite convertir los niveles TTL de voltaje a los niveles de voltaje apropiados para el estándar EIA-485. De esta forma la terminal de recepción serial del microcontrolador es conmutada respecto a las terminal de salida decodificada de los circuitos integrados MAX232 y SN75176. Es decir que la señal enviada por el dispositivo microcontrolador es enviada simultáneamente a los codificadores de EIA-232 y EIA-485, mientras que la terminal de salida de cada uno de los decodificadores es conmutada seleccionando el tipo de protocolo con el que se establecerá la comunicación half-duplex.

Posición del Jumper	Tipo de Comunicación
1-2	EIA-232
2-3	EIA-485

Tabla 3.4: Selección manual del tipo de comunicación en la tarjeta de adquisición

### 3.3.3. Tarjeta de Acondicionamiento de señales

Para permitir el uso óptimo del rango dinámico disponible en los convertidores A/D del microcontrolador ADuC848 ha sido desarrollada una tarjeta de acondicionamiento de señales adaptable al puerto analógico de la tarjeta de adquisición de datos. La tarjeta está organizada a manera de banco de amplificadores de instrumentación que ofrecen la posibilidad de emplear sensores tanto diferenciales como pseudo-diferenciales, además permiten montar su señal de salida sobre un nivel de voltaje determinado (1.24V) mediante la conexión de un jumper en el conector determinado para esta prestación. Un diagrama electrónico de esta tarjeta se muestra en el apéndice A. Esta tarjeta ha sido construida en el laboratorio de Electrónica del IA-UNAM (ver Figura 3.13).

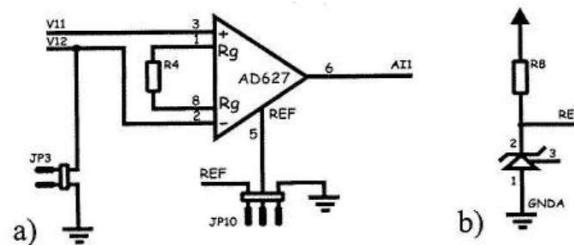


Figura 3.17: a) Bloque básico de acondicionamiento de señal, b) Referencia de voltaje.

## 3.4. Microcomputadora Gumstix WaySmall 200ax

### 3.4.1. Sistemas integrados

Un sistema integrado (conocido también como *empotrado*) es un software de uso específico insertado dentro de uno mucho mayor, usualmente son incrustados en memorias y procesadores pequeños y de bajo costo. En un sistema empotrado los componentes se encuentran generalmente integrados en una tarjeta madre, por lo que un daño en un componente significa el reemplazo de la tarjeta entera, aunque su bajo costo lo compensa.

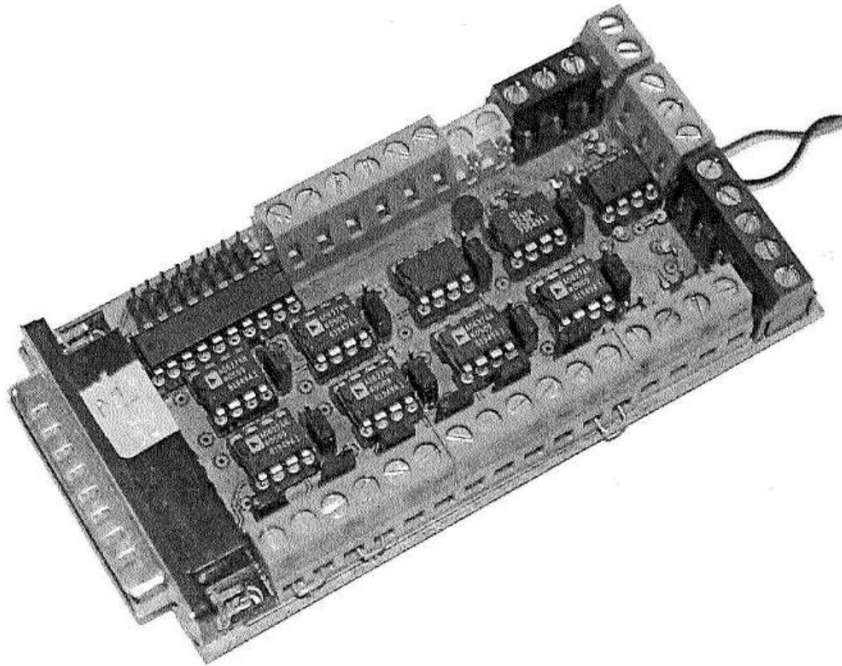


Figura 3.18: La tarjeta de acondicionamiento de señales incluye un banco de amplificadores de instrumentación así como un amplificador en modo de saturación.

Un microprocesador constituye la unidad central de procesamiento y es capaz de acceder tanto memoria interna como externa. La comunicación con otros sistemas electrónicos se realiza a través de diversos puertos incorporados dentro del conjunto de prestaciones. Los puertos comunmente empleados son los del tipo RS-232, RS-485, SPI, I2C y USB, entre otros.

Los sistemas integrados pueden controlar una gran diversidad de procesos, por medio de sus subsistemas de entrada/salida, entre los que podemos encontrar sistemas de redes de sensores, sistemas de control automático, dispositivos de transmisión de datos, actuadores y hasta otros microcontroladores. En suma, un sistema empotrado cuenta básicamente con un microprocesador, memorias, periféricos y *software* dedicado a la administración de estos recursos. Un arreglo más complejo puede emplear un sistema operativo para la ejecución de los programa que contiene. Se emplea el término **Linux empotrado** cuando el sistema operativo

Linux es incrustado en un sistema integrado. El kernel de Linux es muy estable, no requiere el pago de una licencia y solo ocupa alrededor de 2MB de memoria.

### 3.4.2. Microcomputadora Gumstix

La microcomputadora WaySmall 200ax, de la compañía Gumstix, es el sistema empotrado elegido para desempeñar las tareas de manejo y almacenamiento de datos generados por la red de módulos de adquisición de los datos y la lectura de los dispositivos digitales relativos a la medición de los flujos de  $CO_2$  y vapor de  $H_2O$ . Es también la encargada de realizar el procesamiento básico de datos recibidos desde los módulos (a los cuales hasta este punto se le han realizado ya un proceso de filtrado por promediación de las mediciones directas), así como de su almacenamiento en el dispositivo de memoria externa. En la Tabla 3.5 se presentan las características generales del hardware disponibles en la microcomputadora.

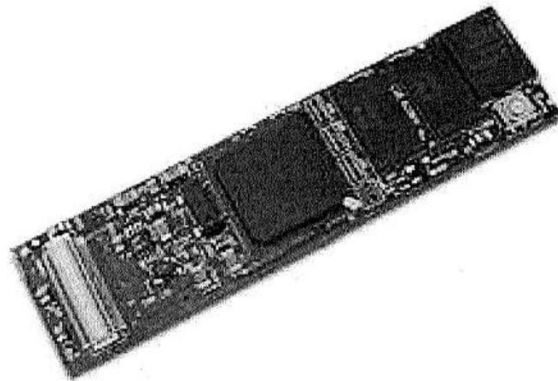


Figura 3.19: Tarjeta madre de la Gumstix WaySmall 200ax.

### 3.4.3. Tarjetas de expansión de puertos

#### Gumstix UARTs

Los dispositivos Gumstix están basados en el procesador Intel PXA255. Cuenta con cuatro UARTs, cada uno de los cuales puede manejar un puerto serial específico. Los niveles lógicos manejados por la interfaz UART son de 3.3V. Los puertos `ttyS0` y `ttyS2` incluyen un convertidor de voltaje por lo que son compatibles con dispositivos TTL. Ambos puertos pueden usar las señales TXD, RXD, CTS y RTS. Los puertos se distribuyen

Modelo	WaySmall 200ax
Dimensiones	83 x 36 x 15mm
Procesador	Intel XScale PXA255
Memoria	64 MB en RAM, 16MB en Flash
Velocidad	200MHz
Software	Linux 2.6.18.gum
Puertos seriales	2 puertos seriales mini-DIN8 para conectar la computadora WaySmall con una PC vía Interfaz serial UART
USB	Puerto USB mini-B
Incluye:	Tarjeta madre <i>basix</i> Tarjeta de expansión HWUART Cubierta plástica Adaptador de pared (4.5V)

Tabla 3.5: Contenido de la microcomputadora Gumstix

de la siguiente manera:

- **FFUART** 230 kbps (normalmente mapeado al puerto `ttyS0`)
- **BTUART** 230 kbps (normalmente mapeado al puerto `ttyS1`)
- **STUART** 230 kbps (normalmente mapeado al puerto `ttyS2`)
- **HWUART** 921 kbps (normalmente mapeado al puerto `ttyS3`)

#### WaySmall-Original

Esta tarjeta añade dos puertos seriales y un USB cliente (con un conector USB-mini-B), lo que permite a la microcomputadora establecer comunicación con otros dispositivos digitales. El uso del puerto `ttyS3`

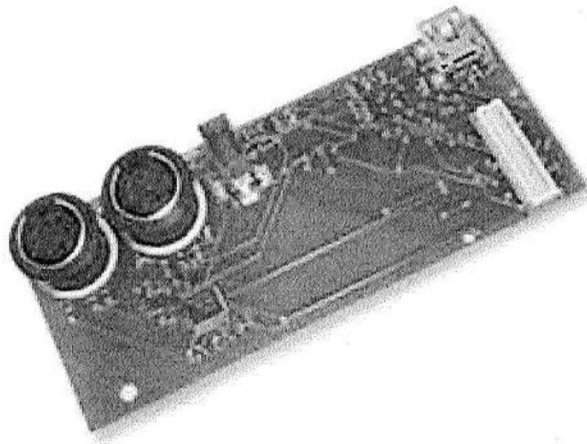


Figura 3.20: Tarjeta de expansión de puertos de la Gumstix

(normalmente ligado al conector miniDIN8 del segundo puerto serial de la microcomputadora) debido a errores en su diseño se encuentra deshabilitado en la tarjeta WaySmall-Original. En la presente versión de la tarjeta dicho puerto ha sido restringido para la implementación de Bluetooth. Para poder utilizar la terminal puerto se requieren algunas modificaciones a la tarjeta WaySmall Original. Deben ser abiertas dos líneas del circuito impreso y conectadas las terminales RXD y TXD del puerto `ttyS2` con las terminales RXD y TXD del anterior puerto `ttyS3` (ver Figura 3.21). De esta manera es posible acceder el puerto `ttyS2` desde la terminal originalmente destinada para el `ttyS3`. Este problema ha sido corregido en las versiones más nuevas de la

tarjeta de expansión WaySmall-STUART. Más información puede ser consultada a través de los hipervínculos [http://docwiki.gumstix.org/Gumstix\\_UARTs](http://docwiki.gumstix.org/Gumstix_UARTs), donde se proporciona información sobre los puertos UART y <http://cs.gmu.edu/~eclab/projects/robots/flockbots/pmwiki.php?n=Main.Gumstixmods>, donde se ilustra el proceso aquí mencionado, así como las modificaciones necesarias para habilitar la comunicación I2C, cuando esta sea requerida.

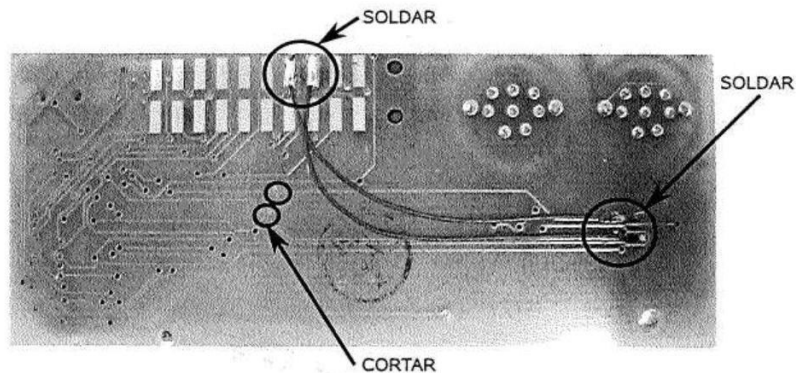


Figura 3.21: Modificaciones a la WaySmall-Original para implementar el puerto ttyS2.

#### 3.4.4. Memoria externa

##### Memoria Compact Flash tipo II

Una memoria Compact Flash (CF) es un elemento de almacenamiento de datos, empleado en algunos dispositivos electrónicos portátiles. Utiliza memoria flash no-volátil, por lo que la información no se pierde cuando el dispositivo es deselectrizado. Hay dos tipos de tarjetas CF: el tipo I y el tipo II que es ligeramente más grueso. Existen modelos que operan en dos velocidades (CF original y CF de alta velocidad). En el presente trabajo han sido empleadas memorias tipo II de alta velocidad de 256MB, 512MB y 4GB. El último modelo ha sido elegido para desempeñar la tarea de almacenamiento de datos de campo obtenidos por la torre de correlación turbulenta. La Figura 3.22 ilustra la apariencia física de este dispositivo de memoria.

##### CFStix

La tarjeta CFStix (Figura 3.23) permite a la Gumstix el acceso a memoria externa para almacenamiento de datos. Siendo empleado durante este trabajo el formato de almacenamiento FAT32, un sistema de ficheros

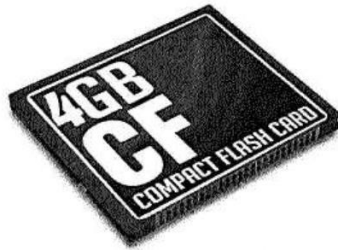


Figura 3.22: Memoria Compact Flash tipo II High Speed.

desarrollado por MS-DOS. El proceso de lectura/escritura tiende a hacerse más lento con el tiempo, por lo cual requiere ser desfragmentada regularmente. Este formato de almacenamiento permite que los archivos de datos puedan ser manejados tanto en Windows como en Linux.

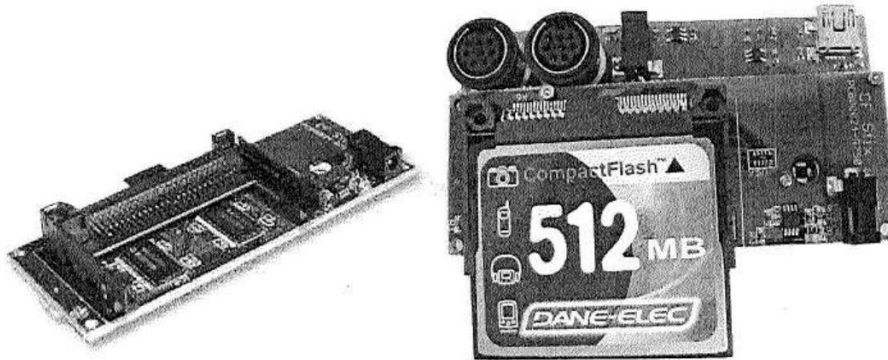


Figura 3.23: CFStix (izquierda), montura del sistema completo (derecha).

La tarjeta CFStix requiere para su funcionamiento 4V (aunque posee un conector de entradas, estos son alimentados desde la tarjeta madre). Se debe tener cuidado en lo siguiente: aunque la tarjeta madre puede alimentarse con voltajes dentro del intervalo de 4 a 5V, en los casos donde sea empleada la tarjeta CFStix todo el sistema deberá emplear un fuente de voltaje de 4V.

### 3.5. Sistema de energía solar

Debido a que las torres de correlación turbulenta serán colocadas en sitios remotos, donde el acceso a la energía eléctrica convencional es difícil o imposible y siendo la energía eléctrica la base del funcionamiento de este conjunto, debe recurrirse a un sistema alternativo de generación y almacenamiento de energía.

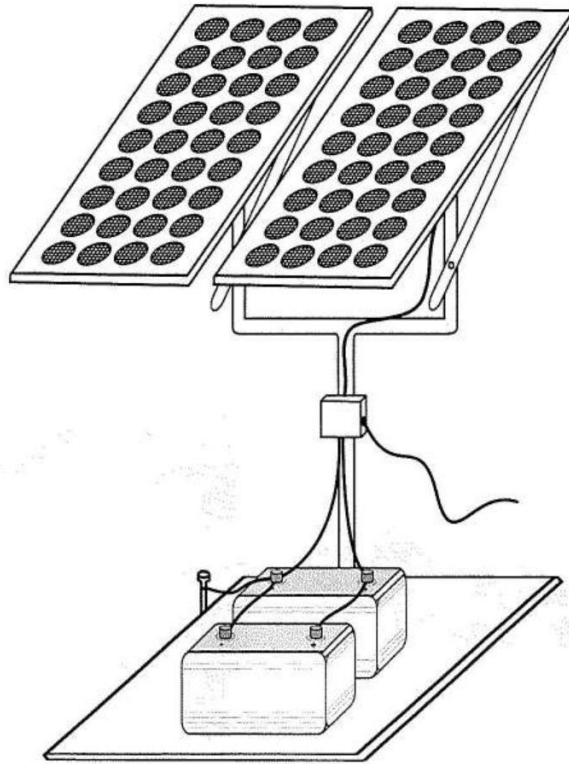


Figura 3.24: Diagrama de la montura polar y almacenamiento de reserva.

La energía solar es la energía que puede ser obtenida directamente del sol mediante el empleo de celdas fotovoltaicas. De este modo la energía radiante es transformada en energía eléctrica mediante el denominado efecto fotoeléctrico. Las celdas fotovoltaicas son fabricadas a base de semiconductor de silicio y se encuentran comercialmente disponibles en sus formas amorfa, policristalina y monocristalina, siendo la última la que ofrece la mayor eficiencia. Ya que la intensidad de la corriente generada aumenta con la radiación (permaneciendo el voltaje prácticamente constante), las celdas deben ser orientadas de forma que durante el día

reciban la mayor cantidad de radiación directa posible. El aumento de la temperatura de las celdas supone un aumento de la corriente, pero al mismo tiempo una disminución notable en el voltaje. Durante los máximos de radiación ( $1000W/m^2$ ), dependiendo de la latitud, la temperatura de los paneles puede aumentar hasta 30 grados sobre la media del ambiente provocando una disminución de potencia generada de hasta 15 %. Por ésta razón es importante que los paneles sean colocados en lugares despejados donde puedan ser ventilados eficientemente.

La nubosidad es un factor impredecible que tiene consecuencias directas sobre la cantidad de energía captada por los módulos fotovoltaicos. Para que el sistema sea capaz de operar incluso en condiciones adversas ha sido provisto de baterías de ciclo profundo, que al contrario de las pilas comunes, son capaces de tolerar descargas completas en repetidas ocasiones. Mediante el trabajo conjunto de paneles solares y baterías es posible estabilizar la energía suministrada al sistema aún en condiciones de baja iluminación. Para las necesidades del proyecto presente, la combinación elegida de dispositivos permitirá mantener el sistema operando de manera autónoma durante al menos cuatro días de ausencia total de luz, una condición muy poco probable, garantizando así la funcionalidad del sistema de adquisición de datos. Debe hacerse notar que el consumo eléctrico del sistema es relativamente alto principalmente por los requerimientos del anemómetro ultrasónico y el analizador de gases infrarrojo empleados para el análisis de los flujos de carbono.

El sistema de energía solar ha sido integrado como sigue: dos módulos solares de 65 watts, una base polar para módulos de 65 watts, un controlador de carga 12A/12VDC, dos baterías AGM 12VDC/230Ah y terminales y accesorios requeridos para el montaje de los elementos citados. El esquema de la montura polar de los paneles solares se muestra en la Figura 3.24.



## Capítulo 4

# Sensores meteorológicos y edáficos

Las torres de correlación turbulenta consisten de equipo especializado dedicado a la obtención de parámetros atmosféricos y del suelo, lo que la diferencia de las comunes torres meteorológicas, principalmente por el uso al cual estos datos están destinados.

### 4.1. Analizador de concentración de gases de $CO_2/H_2O$

El analizador infrarrojo de gases (IRGA) de trayectoria abierta LI-7500 de la compañía LI-COR, permite mediciones rápidas y precisas de densidades *in situ* de  $CO_2$  y  $H_2O$  en flujos turbulentos de aire. En estudios de flujo turbulento estos datos son empleados, en conjunto con la información provista por un anemómetro ultrasónico, para determinar los flujos de  $CO_2$  y el vapor de agua. Las mediciones obtenidas con estos sistemas tiene aplicaciones en la determinación de los presupuestos de dióxido de carbono y vapor de agua en ambientes agrícolas y forestales así como otros estudios ecológicos y de calentamiento global.

#### 4.1.1. Descripción general del LI-7500

El analizador LI-7500 se constituye por dos componentes: la cabeza del sensor y la caja de control que contiene la electrónica del sistema. La cabeza del sensor tiene 12.5 cm de trayectoria abierta por la cual el aire atraviesa pasivamente. Cuenta con dos lentes por los cuales se hace pasar un haz infrarrojo de un centímetro de diámetro. Filtros de referencia centrados a  $3.95 \mu\text{m}$  y  $2.40 \mu\text{m}$  proveen correcciones de atenuación para longitudes de onda no absorbidas mientras que la absorción en las longitudes de onda de  $4.26$  y  $2.49 \mu\text{m}$

proveen información para la medición de  $CO_2$  y vapor de agua, respectivamente. Este dispositivo es capaz de operar en el intervalo de temperaturas de  $-25$  a  $+50$  °C.

En la Figura 4.1 se muestran los principales componentes del analizador de gases. Una fuente infraroja emite radiación cuyo haz se hace pasar a través de cada uno de los filtros de la rueda de filtros (montada sobre el eje de un motor que gira a 9000 revoluciones por minuto) y por los lentes de enfoque y es finalmente medido en el detector ubicado en el otro extremo de la trayectoria. De acuerdo al fabricante tiene una vida útil mínima de 20,000 horas.

Posee ventanas de zafiro ubicadas en los extremos de la trayectoria óptica para proteger las delicadas lentes del instrumento. La electrónica se encuentra contenida en una caja a prueba de agua y de sol.

#### 4.1.2. Realización de mediciones

La caja de control del LI-7500 normalmente se monta de manera vertical o con una leve inclinación para permitir el escurrimiento del agua sobre las ventanas y para causar la menor obstrucción al aire sobre el rango de las direcciones azimutales. La cabeza del sensor puede ser montada en cualquier dirección deseada sin afectar su desempeño.

El relieve, la vegetación y otras fuentes de rugosidad pueden inducir turbulencias por lo que es requerido un sistema de medición de alta velocidad y precisión. Los flujos de  $CO_2$  y  $H_2O$  son obtenidos de la componente vertical de la velocidad del viento y sus respectivas concentraciones.

Para minimizar las perturbaciones al flujo de viento la cabeza del sensor del LI-7500 tiene un perfil aerodinámico. Además las mediciones en trayectoria abierta eliminan la necesidad de una bomba, lo cual permite ahorrar una gran cantidad de energía, un factor específicamente importante para estaciones remotas donde el consumo eléctrico es determinante. El IRGA consume menos de 10 W en operación y hasta 30 W durante el arranque.

El análisis en trayectoria abierta de los flujos de aire reduce considerablemente los retados en tiempo, además de que elimina la posibilidad de fallas de presión de la bomba (para aquellas versiones anteriores que requie-

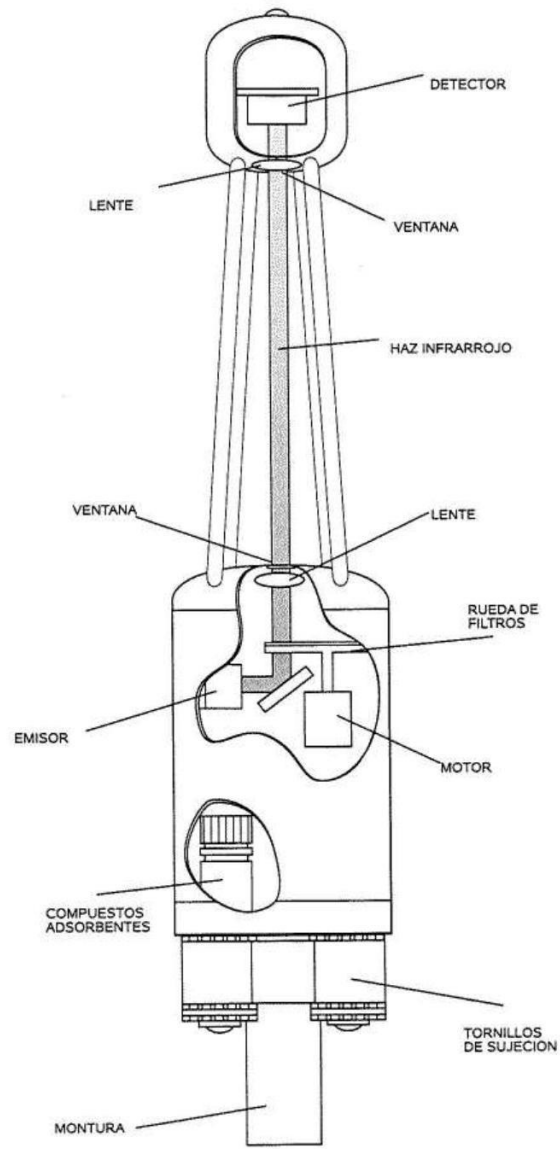


Figura 4.1: Diagrama de la estructura interna del analizador de gases infrarrojo LI-7500.

ren inyectar aire a sus sistemas de medición, a lo que se le denomina como *trayectoria cerrada*) así como la adsorción/desorción de vapor de agua en las tuberías empleadas en dichos sistemas.

Dentro del dispositivo se encuentran colocadas dos botellas plásticas las cuales contienen Ascarita II (un adsorbente tradicional de  $CO_2$ ) y Perclorato de Magnesio ( $Mg(ClO_4)_2$ ) empleado como elemento desecante. Estos compuestos (que permiten mantener el detector libre remanentes de gases de  $CO_2$  y vapor de agua respectivamente) deben ser reemplazados anualmente.

#### 4.1.3. Software y señales de salida

El LI-7500 posee tres formatos de salida de datos:

- Voltajes analógicos linealizados, vía dos convertidores digital-a-analógico (DACs). Configurables por el usuario los cuales son refrescados 300 veces por segundo por un procesador digital de señales (DSP).
- Salida EIA-232 disponible a tasa de transferencia de datos de hasta 38,400 baudios, con tasas de muestreo de hasta 20 muestras por segundo. Pueden ser empleados anchos de banda de 5, 10 ó 20 Hz para tiempos de respuesta de 0.1, 0.05 y 0.025 segundos, respectivamente.
- Salida *Synchronous Device for Measurement* (SDM) para registradores de la compañía Campbell Scientific Incorporated con la que pueden obtenerse hasta 40 muestras por segundo.

#### 4.1.4. Calibración

El analizador de gases infrarrojo es calibrado de dos maneras: una calibración de fábrica que consiste en determinar los valores de los coeficientes y la calibración por el usuario (realizada semanal o mensualmente) que tiene la finalidad de corregir los niveles de cero y de deriva. La precisión final del aparato depende de ambas calibraciones.

El usuario puede calibrar el LI-7500 empleando un generador de punto de condensación LI-610, de la compañía LI-COR, mostrado en la Figura 4.2, certificado por el National Institute of Standards and Technology (NIST). El IRGA es calibrado insertando el tubo de calibración del LI-610 en la trayectoria óptica, entonces el volumen del tubo es puesto en vacío para para fijar el cero y posteriormente relleno con una concentración conocida

para fijar el rango de operación.

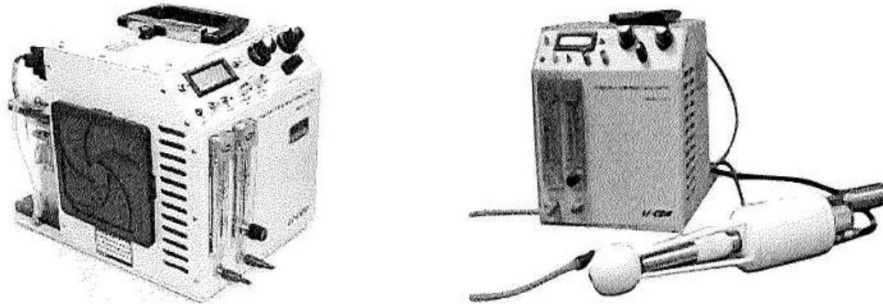


Figura 4.2: LI-610. Generador de punto de condensación (*izquierda*) para calibración del flujo mediante la inserción del tubo de calibración en la trayectoria del haz infrarrojo (*derecha*).

## 4.2. Anemómetro ultrasónico tridimensional

El anemómetro ultrasónico modelo Young 81000V mide la velocidad del viento y el sonido en el espacio tridimensional basado en el tiempo de tránsito de una señal acústica ultrasónica. Posee además un sistema que le permite calcular la temperatura sónica. Los datos pueden ser obtenidos por medio de las interfaces EIA-232 o EIA-485 o mediante canales de voltajes analógicos. El anemómetro puede ser configurado vía menú usando algún programa de comunicación serial, usualmente Windows Hyperterm o Linux C-Kermit.

Este tipo de anemómetro tiene las ventajas de que no posee partes móviles y puede ser empleado en aplicaciones que requieren de respuesta rápida y alta resolución. El sensor está contruido en acero inoxidable y componentes de aluminio anodizado para evitar la corrosión. Los transductores se encuentran organizados en tres pares opuestos de manera que las mediciones son efectuadas en un volumen común. Internamente realiza mediciones a una tasa de muestreo de 160 Hz y ofrece tasas de salida de datos de 4 a 32 Hz seleccionables por el usuario. Para aplicaciones que requieren mediciones analógicas sincronizadas este modelo incluye cuatro canales para voltajes de entrada. Viento, temperatura sónica y dichos voltajes analógicos pueden ser transmitidos serialmente. En el presente desarrollo los canales DAC1 y DAC2 del IRGA son conectados al Anemómetro Young 81000V, de tal manera que estos datos son agregados y leídos por el dispositivo maestro

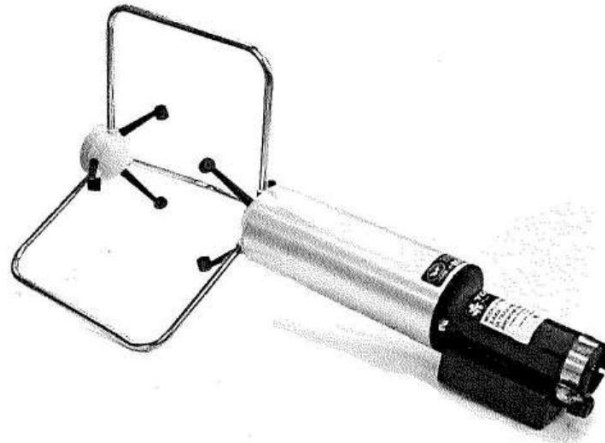


Figura 4.3: Anemómetro ultrasónico tridimensional Young 81000V.

en la cadena de salida de datos del anemómetro. Debe recordarse que estas entradas de voltaje analógico son normalizadas a 4 V.

#### 4.2.1. Especificaciones

##### Rapidez del viento

Operable en el rango de 0 a 40 m/s (0 a 90mph) con una resolución de 0.01 m/s, umbral de 0.01 m/s, precisión de 1% rms  $\pm 0.05$  m/s en el intervalo de 0 a 30 m/s y de  $\pm 3\%$  rms en el intervalo de 30 a 40 m/s.

##### Dirección del viento

Opera en los 360 grados del plano U-V (azimutal) y  $\pm 60$  grados de elevación con una resolución de 0.1 grados y una precisión de  $\pm 2$  grados en el intervalo de 1 a 30 m/s y de  $\pm 5$  grados de 30 a 40 m/s.

##### Velocidad del sonido

Ofrece mediciones comprendidas en el intervalo de 300 a 360 m/s con resolución de 0.01 m/s y una precisión de  $\pm 0.1\%$  rms  $\pm 0.05$  m/s.

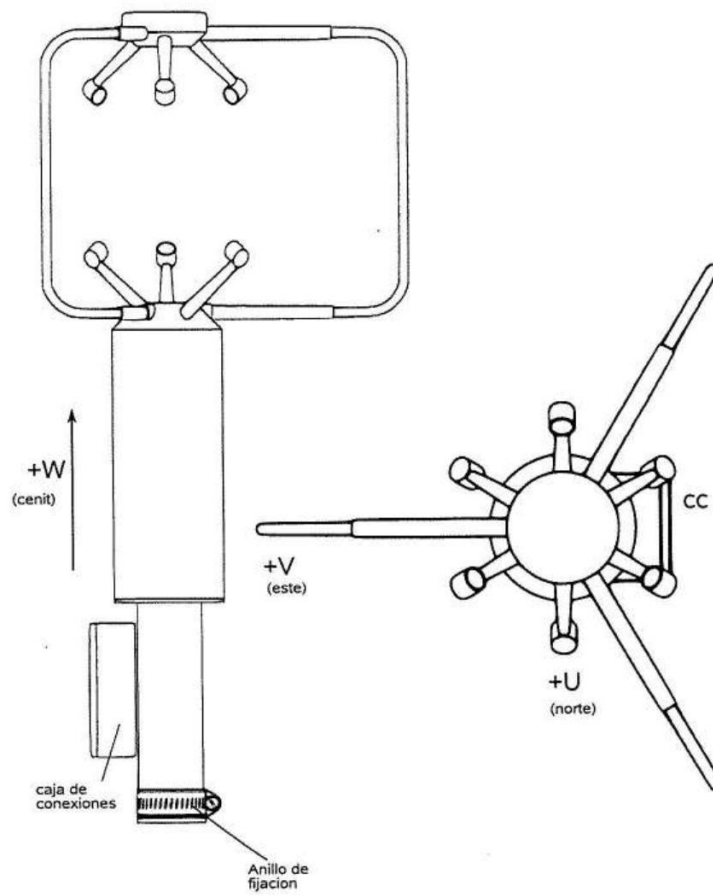


Figura 4.4: Orientación del anemómetro ultrasónico tridimensional Young 81000V.

### Temperatura sónica

Intervalo de mediciones entre -50 y +50 C con resolución de 0.01 C y  $\pm 2$  °C de precisión (0 a 30 m/s).

### Salida serial

RS-232 o RS-485 de 1200 a 38400 baudios. Salida seleccionable por usuario entre 4 y 32 Hz en pasos fijos. Cadena de salida tipo ASCII programable.

### Convertidores A/D

Los canales V1 y V2 aceptan voltajes en el rango de 0 a 5 V, mientras que los canales V3 y V4 de 0 a 1 V, los dos pares de canales con una resolución de 0.25 mV y una precisión de  $\pm 0.1\%$  en escala completa.

## 4.3. Presión barométrica

El barómetro analógico CS105, de la compañía Vaisala, utiliza un sensor capacitivo diseñado para mediciones estables y precisas con una salida lineal de 0 a 2.5 V (600 a 1060 mb). Puede ser operado en modo de activación por pulso o modo continuo. Figura 4.5.

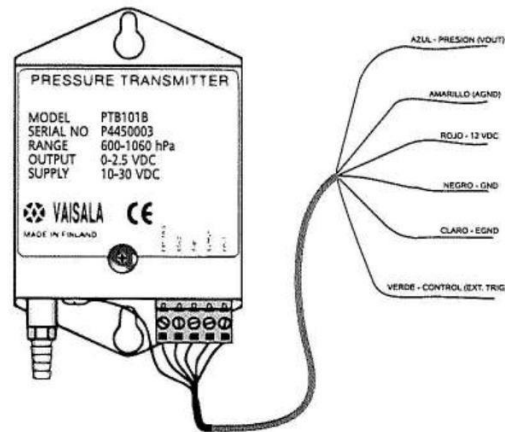


Figura 4.5: Medidor de presión barométrica con diagrama de conexiones.

#### 4.4. Precipitación pluvial

El pluviómetro es un instrumento empleado para medir la precipitación de agua desde la atmósfera, ya sea por concepto de lluvia o nieve. La cantidad de agua precipitada es expresada en milímetros de altura. El diseño básico de un pluviómetro consiste en un recipiente por donde el agua ingresa a través de un embudo hacia un colector de tipo balancín, con dos canastas con una capacidad de 8.2 ml cada una, el cual se encuentra acoplado a un interruptor de mercurio líquido permitiendo el registro de la intensidad de lluvia caída en forma de pulsos eléctricos. Un milímetro de lluvia equivale a un litro de agua distribuido sobre una superficie de un metro cuadrado. El pluviómetro posee un superficie colectora conocida y canastas con capacidad fija, lo que permite estimar su equivalente en milímetros de agua sobre metro cuadrado. Para el pluviómetro empleado esta equivalencia es de 0.26 mm por pulso. Este dispositivo fue recuperado de un pluviógrafo en desuso. Véase Figura 4.6.



Figura 4.6: Pluviómetro de cazoletas basculantes.

## 4.5. Temperatura ambiental y humedad relativa

El HMP45C, de la compañía Campbell Sci, contiene un sensor resistivo de temperatura de platino y un sensor capacitivo de humedad relativa de la marca Vaisala HUMICAP® 180. El HMP45C puede ser alimentado con un voltaje constante o mediante control de encendido por conmutación. Figura 4.7.

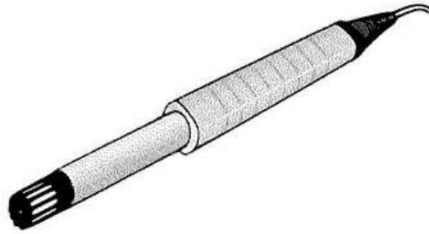


Figura 4.7: Sensor de temperatura ambiente y humedad relativa HMP45C.

### 4.5.1. Sensor de temperatura

El sensor específico con el que se realiza el sensado de temperatura en el HMP45C es un  $1000 \Omega$  PRT, IEC 751 1/3 Clase B, que opera en el rango de temperaturas de  $-40$  a  $+60$  grados centígrados correspondientes a una salida de  $0.008$  a  $1.0$  V. Es un sensor muy estable, con una desviación anual de menos de un punto porcentual, pero se recomienda que sea recalibrado anualmente. El sensor consiste en un Pt1000 de respuesta lineal con una resistencia estándar de  $1000$  Ohms a  $25$  grados centígrados.

### 4.5.2. Sensor de humedad relativa

Este sensor es un HUMICAP® 180. Opera en el rango de mediciones de  $0$  a  $100\%$  de humedad relativa, correspondientes a una salida de voltaje de  $0.008$  a  $1.0$  V, con una desviación anual de menos de uno por ciento. Su tiempo de respuesta es de aproximadamente  $15$  segundos. Tiene una precisión de  $\pm 2\%$  en el intervalo de  $0$  a  $90\%$  y de  $\pm 3\%$  en el intervalo de  $90$  a  $100\%$  así como una desviación típica de  $1\%$  anual.

## 4.6. Temperatura del suelo

Un termistor es un dispositivo semiconductor cuya resistencia eléctrica varía con los cambios de temperatura. Para los termistores NTC (*Negative Temperature Coefficient*) esta resistencia decrece a medida que la temperatura aumenta. Poseen un alto coeficiente de temperatura negativo, de tal forma que su conductividad crece, de manera no-lineal, al aumentar la temperatura. Esta no-linealidad hace que en algunos casos se requiera la utilización de elaborados algoritmos de linealización. Debido a su alta impedancia el empleo de estos dispositivos reduce los errores provocados por la longitud de los hilos conductores. Estos termistores pueden ser conectados a puentes de Wheatstone convencionales (Figura 4.8) o a otros circuitos para la medición de resistencia.

### 4.6.1. Cálculo de las resistencias del puente de Wheatstone

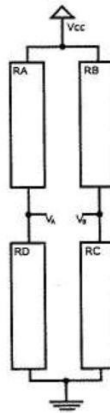


Figura 4.8: Puente de Wheatstone.

Analizando el circuito convencional de un puente de Wheatstone se busca obtener los valores de resistencia apropiados tomando en cuenta tanto el voltaje de alimentación del puente ( $V_{CC}$ ), como del rango dinámico del ADC del microcontrolador ( $V_M$ ). El voltaje de salida diferencial entre los nodos  $V_A$  y  $V_B$  está dado como  $V_O = V_A - V_B$ , donde  $V_O$  corresponde con  $V_M$ . El circuito está inicialmente balanceado. Se establecen condiciones iniciales  $R_A = R_C$  y  $R_C = R_D$ , en el cual  $R_D$  es el valor mínimo de resistencia del termistor.

$$R_A = R_B, R_C = R_D \quad (4.1)$$

Aplicando la Ley de Voltajes de Kirchoff en ambos lazos del puente de Wheatstone se obtienen las siguientes relaciones:

$$V_A = V_{CC} \left( \frac{R_D}{R_D + R_A} \right) \quad (4.2)$$

$$V_B = V_{CC} \left( \frac{R_C}{R_C + R_B} \right) \quad (4.3)$$

A partir de las cuales se obtiene la salida diferencial  $V_O$ .

$$V_O = V_{CC} \left[ \frac{R_D R_B - R_C R_A}{(R_D + R_A)(R_C + R_B)} \right] \quad (4.4)$$

Partiendo del supuesto que el puente está inicialmente balanceado, se tiene que  $V_{OB} = 0V$ , donde  $V_{OB} = 0$  es el valor mínimo de voltaje en la salida diferencial. Al realizar el despeje correspondiente se obtiene la relación  $R_A R_C = R_B R_D$ .

Suponiendo una nula afectación térmica sobre las resistencias  $R_A$ ,  $R_B$  y  $R_C$ , se incluye la variación en la salida diferencial del voltaje ( $\Delta V_O$ ) como función del cambio de impedancia en el termistor, de tal forma que

$$V_{OB} + \Delta V_O = \Delta V_O = V_{CC} \left[ \frac{(R_D + \Delta R_D) R_B - R_C R_A}{(R_D + \Delta R_D + R_A)(R_C + R_B)} \right] \quad (4.5)$$

Haciendo la sustitución  $R_C = R_D$  y  $R_A = R_B$ , la ecuación queda con dos variables

$$V_{OB} + \Delta V_O = V_{CC} \left[ \frac{V_{CC} \Delta R_D R_A}{(R_D + \Delta R_D) R_D + (R_D + \Delta R_D) + R_A R_D + R_A^2} \right] \quad (4.6)$$

Sustituyendo  $\beta = V_{OB} + \Delta V_O$  se obtiene

$$\beta R_A^2 + R_A [2\beta R_D + \Delta R_D (\beta - V_{CC})] + \beta (R_D + \beta R_D) R_D = 0 \quad (4.7)$$

Los valores máximos y mínimos de resistencia para el termistor en estudio son los que corresponden al intervalo si se conocen los valores dentro de los cuales oscila el sistema y son calculadas las resistencias óptimas para dicho comportamiento, en lugar de desperdiciar la capacidad de conversor analógico a digital del microcontrolador al dotarle de capacidad de muestreo en rangos de temperatura que el sistema no alcanza.

$$R_{TH} = [ R_{THmax} \quad R_{THmin} ] \quad (4.8)$$

Donde  $R_{THmax}$  y  $R_{THmin}$  son los valores máximo y mínimo del termistor, respectivamente. De la expresión anterior se tiene que

$$\Delta R_D = R_{THmax} - R_{THmin} \quad (4.9)$$

De la ecuación 4.7 se despeja el valor de  $R_A$ , donde  $R_A = R_B$ .

#### 4.6.2. Determinación de $R_{TH}$

De la ecuación 4.4 se obtiene la relación

$$(R_D + R_A)V_O = V_{CC}R_D - V_B(R_D + R_A) \quad (4.10)$$

De donde se despeja la variable de interés de  $R_D$

$$R_D = \frac{R_A(V_O + V_B)}{V_{CC} - V_B - V_O} \quad (4.11)$$

Para el caso específico de la punta de prueba modelo 108 de la compañía Campbell Scientific, el termistor se encuentra en serie con una resistencia de valor constante ( $40k\Omega$ ), la cual debe ser restada del valor de  $R_D$  calculado.

$$R_{TH} = R_D - 40k\Omega \quad (4.12)$$

## 4.7. Sensores de radiación solar

### 4.7.1. Radiación solar directa

El piranómetro LI-200X mide la suma de la radiación solar y dispersada por el cielo a través de un detector fotovoltaico de silicio. El detector genera salidas de corriente las cuales son convertidas a voltaje por medio de una resistencia de precisión, permitiendo que la señal pueda ser medida por instrumentos de adquisición de datos con entradas de voltaje. Opera la región del espectro visible en la banda de 400 a 1100 nm. Tiene una sensibilidad de 0.2 kW por metro cuadrado por milivoltio. Ver Figura 4.9.



Figura 4.9: Medidor de radiación solar directa.

#### 4.7.2. Radiación fotosintéticamente activa

Durante la fotosíntesis las plantas utilizan energía de la región del espectro electromagnético en el intervalo de 400 a 700 nm. La radiación en este rango es conocida como *radiación fotosintéticamente activa* (PAR) y puede ser medida en unidades de energía ( $Wm^{-2}$ ) o como densidad de flujo fotónico fotosintético (PPFD) cuya unidad son fotones por metro cuadrado por segundo ( $mmols^{-1}m^2$ ). Este sensor puede ser empleado en ambientes tanto con iluminación natural como artificial. Véase la Figura 4.10.

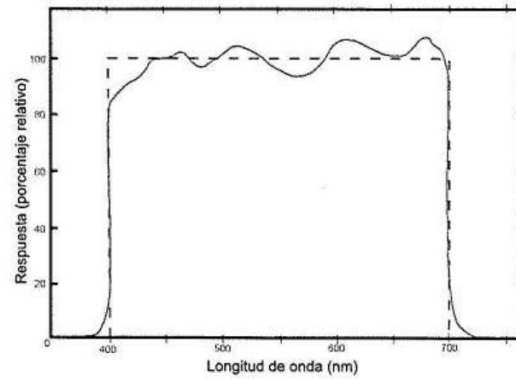


Figura 4.10: El sensor de radiación fotosintéticamente activa (*izquierda*) detecta longitudes de onda en el espectro electromagnético de 400 a 700 nm, que constituye el rango de radiación útil para las plantas (*derecha*).

### 4.7.3. Radiómetro neto

El radiómetro neto NR LITE (ver Figura 4.11) es un instrumento para la medición del balance de radiación solar e infrarroja. Este balance es llamado *radiación neta*. Este sensor posee dos caras sensitivas, desde la cara superior recibe la radiación visible e infrarroja desde su hemisferio superior completo, mientras que desde su sensor inferior recibe radiación reflejada por el suelo. Ambas señales son sumadas y enviadas hacia las terminales de voltaje como una sola salida, el voltaje resultante representa la radiación neta que puede ser interpretada como la energía radiante que es absorbida por la superficie del suelo. La salida debe ser expresada en Watts por metro cuadrado.

Este detector emplea una termopila como sensor de radiación. La termopila consiste en un cierto número de termopares unidos en serie y es esencialmente un sensor diferencial de temperatura. Estos dispositivos generan un voltaje de salida de forma pasiva, esto quiere decir que no requieren una fuente de alimentación para operar.

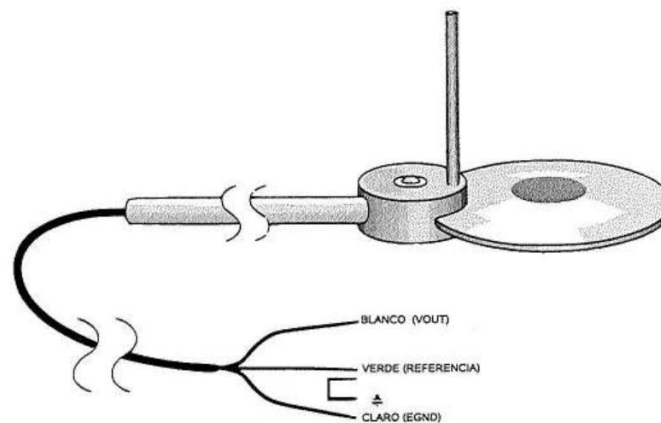


Figura 4.11: Conexiones radiómetro neto NR LITE.

## 4.8. Humedad del suelo

El CS616 mide el contenido volumétrico de agua en un medio poroso usando un método de medición que recurre a la permitividad dieléctrica de un material. El sensor consiste en dos varillas de treinta centímetros conectadas a una circuitería protegida con resina epóxica. Estas varillas pueden ser insertadas sobre la superficie o paralelamente a la superficie con la finalidad de conocer las propiedades de contenido de agua en diferentes estratos de estudio.

Las varillas de la punta de prueba actúan como guías para una onda de propagación cuya velocidad depende de la permitividad dieléctrica del medio a través del cual viaja, en este caso el medio es el suelo. El retorno de las ondas de reflexión es cuidadosamente medido estableciendo una relación de diferencia temporal proporcional a las características del medio. Debido a que el agua tiene una permitividad dieléctrica significativamente mayor a las de los otros componente del sustrato el contenido promedio de agua en el medio circundante a las puntas puede ser estimado fácilmente. La frecuencia de oscilación original en megahertz es reducida para que pueda ser tratada por sistemas de adquisición con capacidades limitadas. Este dispositivo se muestra en la Figura 4.12.

### 4.8.1. Calibraciones

La calibración, es el procedimiento que permite determinar con suficiente exactitud el valor de los errores de los instrumentos de medición con respecto a patrones previamente establecidos. Cuando los datos presentan diferencias deben ser ajustados mediante las denominadas *curvas de calibración*, una forma ampliamente utilizada es la aproximación lineal aunque para sensores más complejos se requiere a veces de polinomios de órdenes superiores. La expresión que expresa un ajuste lineal es la siguiente:

$$a_0 = \bar{y} - a_1\bar{x} \quad (4.13)$$

donde  $\bar{y} - a_1\bar{x}$  son las medias de  $y$  y  $x$  respectivamente.

### Regresión de polinomios

Para algunos casos, una aproximación lineal no es suficientemente adecuada, conviene ajustar una curva. Una alternativa es ajustar polinomios con los datos mediante *regresión de polinomios*. Es posible extender

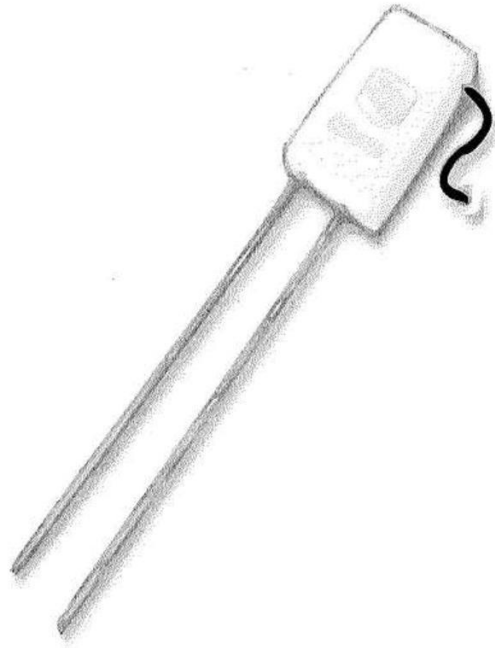


Figura 4.12: Sensor de contenido volumétrico de agua en el suelo.

el procedimiento de mínimos cuadrados a polinomios de orden superior. Para el caso de un polinomio de  $k$ -ésimo orden se obtiene la expresión

$$\begin{bmatrix} N & \sum x_i & \sum x_i^2 & \dots & \sum x_i^k \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{k+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \sum x_i^k & \sum x_i^{k+1} & \sum x_i^{k+2} & \dots & \sum x_i^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ a_k \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \cdot \\ \cdot \\ \sum x_i^k y_i \end{bmatrix} \quad (4.14)$$

donde todas las sumatorias son de  $i = 1$  hasta  $N$ . Conforme el orden del polinomio se eleva, la solución al sistema de ecuaciones lineales se vuelve más complicada. Se requiere de la utilización de métodos numéricos para encontrar la solución a estos sistemas. Sin embargo dichos métodos no se presentan como elemento de estudio de este trabajo. Se ha optado por la utilización del ajuste de curvas proporcionado por el programa *gnuplot* que corre bajo diversas distribuciones de *Linux*.



## Capítulo 5

# Torres experimentales de correlación turbulenta

Los sistemas de adquisición de datos, energía solar, sensores, estructuras metálicas y cajas de protección son integrados en la llamada *torre de correlación turbulenta* (ver Figura 5.1), completando el equipo normalmente empleados para el estudio de flujos turbulentos. En la Figura 5.2 se muestra el esquema básico de conexiones para un sistema mínimo. Cada tarjeta de adquisición de datos posee ocho canales de lectura analógica y dos canales adaptados como contadores de pulsos digitales, así como cuatro puertos de comunicación I2C. Mediante la operación simultánea de varias tarjetas de adquisición a través del estándar EIA-485 esta cantidad puede ser elevada hasta ocho tarjetas de adquisición en operación paralela, lo que ofrece la posibilidad de elevar hasta en un número de 64 sensores analógicos, 16 contadores y 32 canales de comunicación I2C para sensores digitales. La microcomputadora Gumstix tiene dos puertos seriales, uno de ellos es ocupado por el sistema IRGA-Anemómetro y el restante por la red de módulos de adquisición de datos. Los datos son almacenados en sitio en una memoria tipo *compact flash* de alta velocidad y se encuentra actualmente en desarrollo el envío de datos vía radio mediante el empleo de un tercer puerto serial de la microcomputadora.

A los componentes listados se les han agregado los índices alfabéticos mostrados en la Tabla 5 donde se hace referencia al elemento empleado en las torres, con la finalidad de facilitar su identificación en los diagramas electrónicos y estructuras que se indican en las próximas páginas.

Indicador	Descripción/Parámetro	Sensor/Especificación
A	Anemómetro ultrasónico tridimensional	Young 81000V
B	Analizador de gases infrarrojo	LI-7500
C	Antena Yagi	896-970 MHz, 9dB
D	Piranómetro	LI-200X
E	Radiómetro neto	NR LITE
F	Caja DSP - IRGA	Calidad Nema-4
G	Sensor de humedad relativa y temperatura	HMP45C
H	Caja de Instrumentación	Calidad Nema-4
I	Pluviómetro	
J	Sensor de humedad del suelo	CS616
K	Sensor de gradiente de temperatura en suelo	HFT3
L	Sensor de temperatura del suelo	CS-108
M	Sensor digital de temperatura	LM73
N	Sensor de presión barométrica	CS-105
O	Medidor de PPFD	LI-190SA
P	Convertidor RS-485/RS-232	
Q	Tarjeta de adquisición de datos	
R	Tarjeta de acondicionamiento de señal	
S	Microcomputadora Gumstix	
T	Memoria Compact Flash	≤ 4 GB
U	Transceptor serial	CONNEX LINK 900 MHz
V	Estación de trabajo	
W	Paneles solares	65 W
X	Controlador de carga	8 A
Y	Batería de ciclo profundo	12V, 230A/h

Tabla 5.1: Descripción de componentes montados sobre la Torre de Correlación Turbulenta

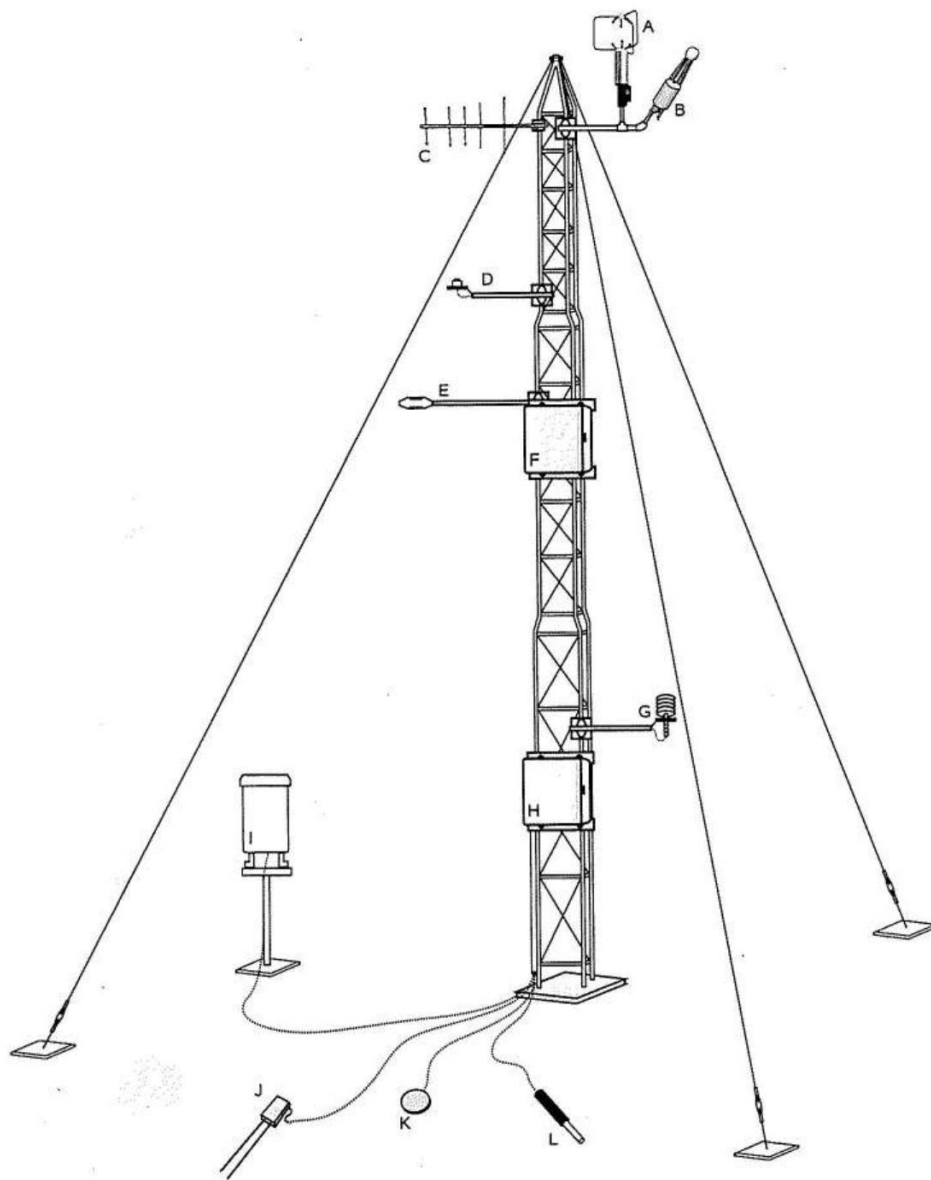


Figura 5.1: Torre de correlación turbulenta. Este tipo de torres son más complejas que sus análogas meteorológicas. El modelo construido para los fines del proyecto TERMAR incluye algunas mejoras sobre los torres empleadas por otros grupos de investigación.

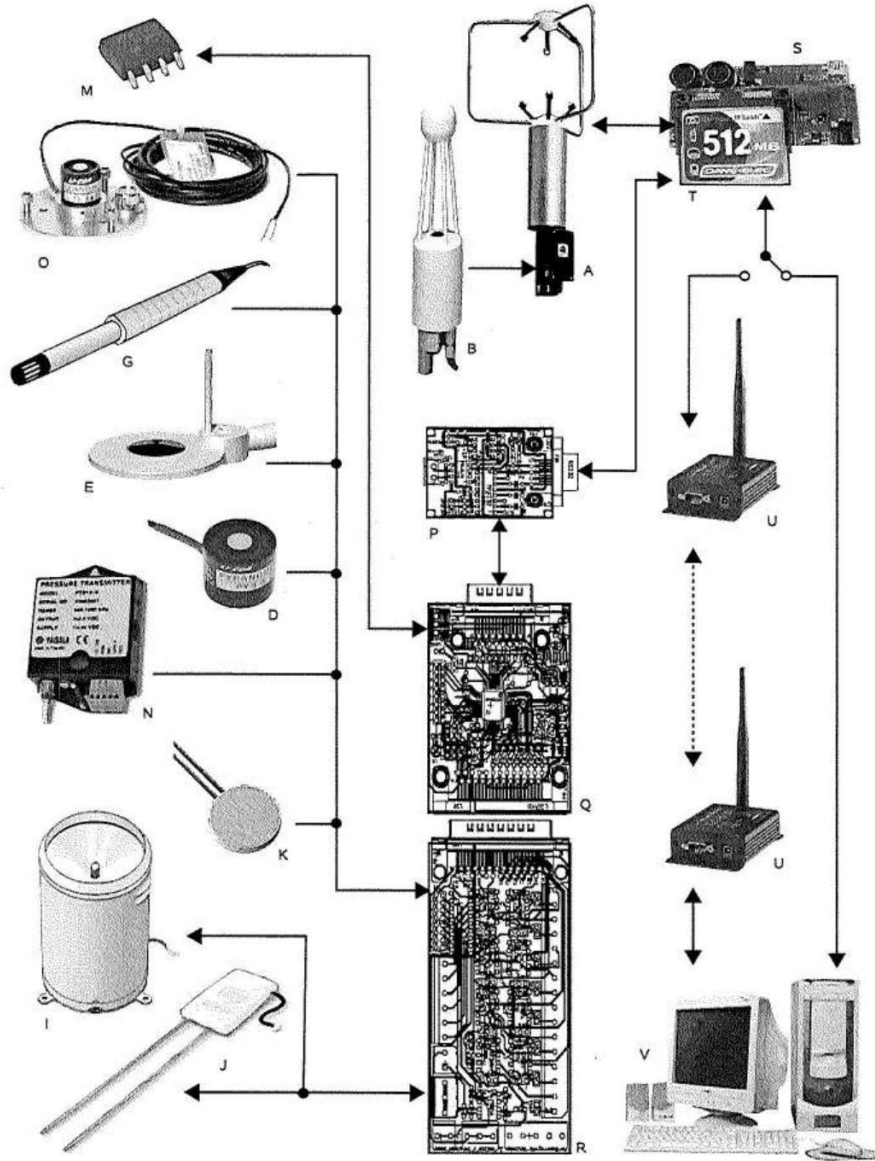


Figura 5.2: Diagrama básico de operación. Este sistema se encuentra en operación con solo un módulo conectado a la red RS-485. Cada módulo cuenta con ocho canales de conversión A/D y 2 contadores. Cuenta también con 4 puertos I2C y los datos obtenidos pueden ser almacenados en memoria externa.

## 5.1. Programación

Tanto el microcontrolador como la microcomputadora requieren de programas que coordinen sus operaciones de adquisición de datos. En el microcontrolador es necesario configurar sus puertos de comunicación así como controlar la lectura de los canales analógicos y con estos datos realizar las sumatoria que permitirán efectuar algunos procesos estadísticos simples, ver Figura 5.3. En la microcomputador se requiere supervisar la lectura de datos desde los puertos seriales asignados tanto a la red de módulos de adquisición (ver Figura 5.4) como al anemómetro digital (ver Figura 5.5). En la microcomputadora se realiza también el proceso de calibración de datos mediante ecuaciones hasta de quinto orden y se realiza el almacenamiento de datos en memoria externa.

La lectura de los distintos módulos la realiza la microcomputadora Gumstix WaySmall 200ax operando bajo el sistema Linux, Kernel 2.6.18.gum, y realizando comunicación con los módulos a una tasa de 9600 baudios. La tasa de comunicación con el anemómetro ultrasónico es configurable por el usuario, usualmente 115200 baudios. Todos los programas han sido desarrollados en el lenguaje de programación de alto nivel C++ y compilados con el compilador cruzado arm-linux-gcc en la estación de trabajo.

Los programas de la Gumstix han sido probados en la estación de trabajo (computadora de escritorio o laptop) antes de ser compilados para el sistema integrado. De esta forma, es posible que las mediciones realizadas por la Gumstix puedan ser también realizadas por la computadora con algunas modificaciones en las rutas donde los archivos de configuración son localizados, así como los destinos para almacenamiento de datos.

Para el caso del programa contenido en el microcontrolador ADuC848, este ha sido editado y compilado con el programa Keil 3.0 en ambiente Windows y su funcionamiento fue probado mediante el uso de la HyperTerminal en el mismo ambiente. El programa Keil 3.0 ha permitido la inclusión de librerías específicas para el uso de microcontroladores, en el presente caso un ADuC848 de la compañía Analog Devices. Cuando el programa se encuentra compilado y en formato hexadecimal es descargado al microcontrolador vía serial usando el programa WSD, que es un software de aplicación específica para la programación de microcontroladores de Analog Devices.

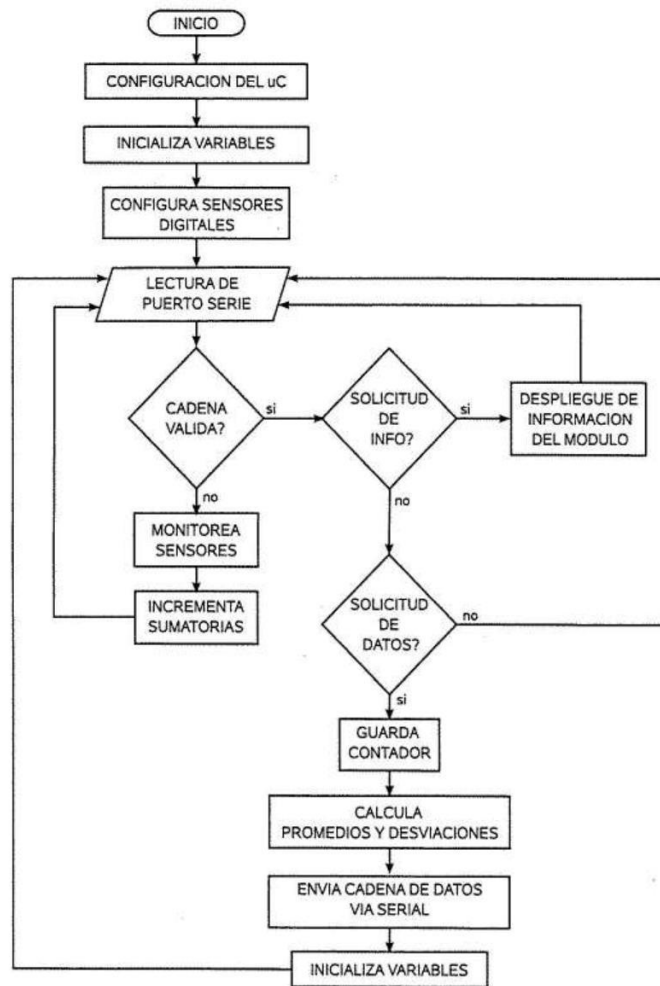


Figura 5.3: Diagrama de flujo del programa del microcontrolador. Este programa se encuentra constantemente leyendo el puerto serial en espera de escuchar su dirección, cuando esto ocurre promedia todas las muestras que ha tomado desde la última ocasión en que estos datos le fueron requeridos y una vez que envía una cadena de datos continúa muestreando en espera de que le sean requeridos nuevamente.

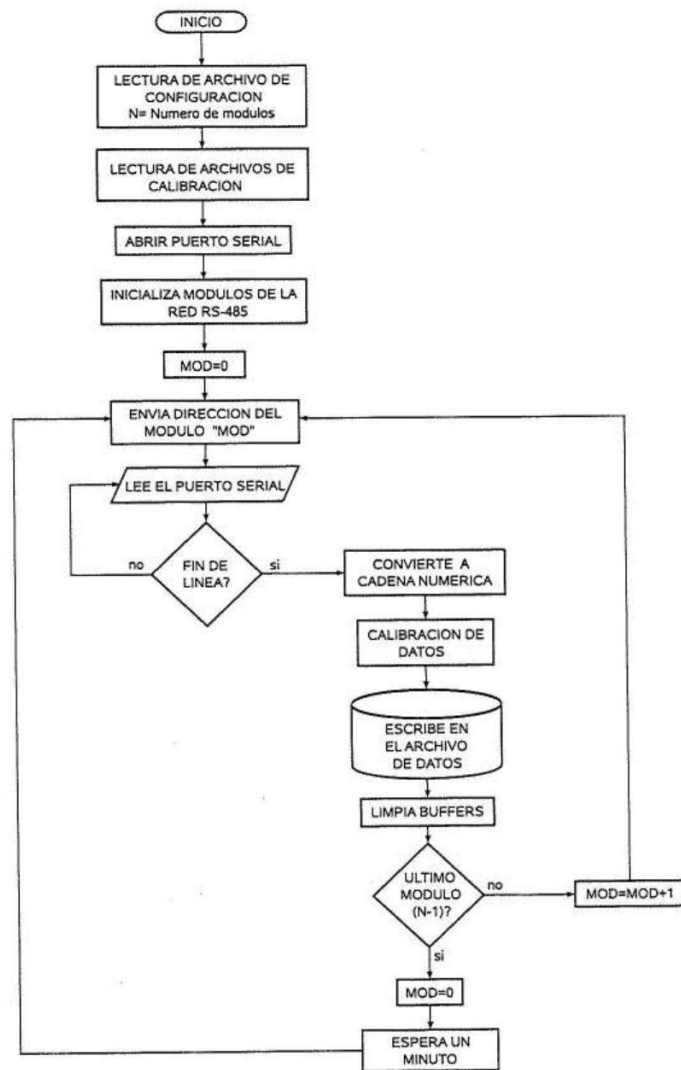


Figura 5.4: Diagrama de flujo del programa de lectura de la red de módulos de adquisición. Este programa envía las direcciones de los módulos y lee las cadenas de datos enviadas por cada uno de ellos en forma secuencial. Luego espera un intervalo de tiempo configurable por el programador, en este caso de un minuto.

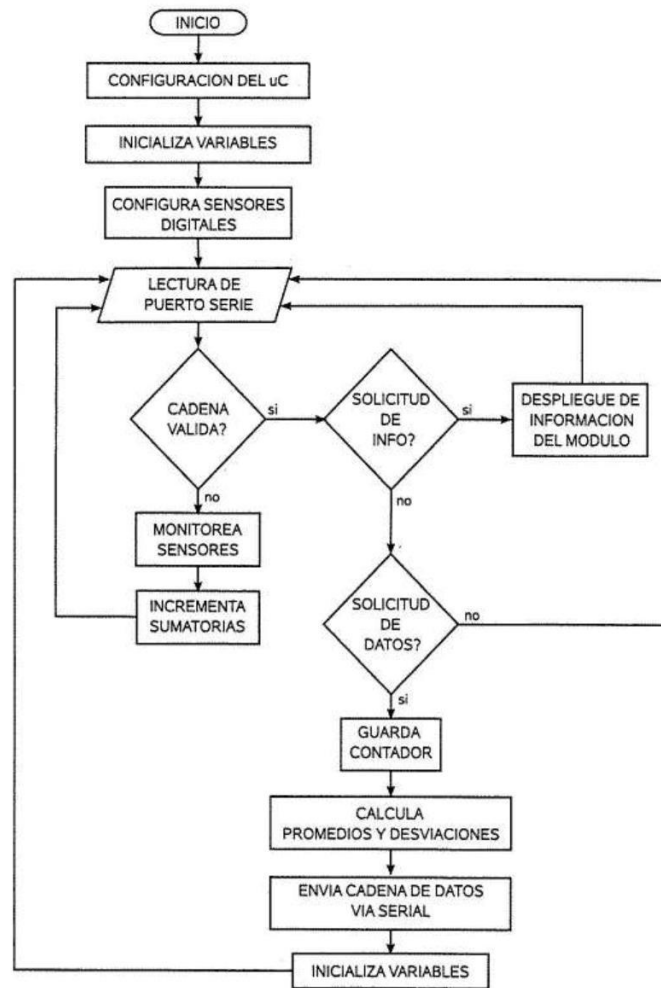


Figura 5.5: Diagrama de flujo del programa de lectura de la red de módulos de adquisición. Este programa envía las direcciones de los módulos y lee las cadenas de datos enviadas por cada uno de ellos en forma secuencial. Luego espera un intervalo de tiempo configurable por el programador, en este caso de un minuto.

## Capítulo 6

# Sistema de archivos

### 6.1. Formato de los archivos de datos

#### 6.1.1. Archivos del módulo de adquisición

La tarjeta de adquisición de datos toma muestras durante un periodo indeterminado de tiempo, no es sino hasta que los datos le son requeridos que los promedia. De esta manera es la estación de trabajo que esté a cargo del control de la red RS-485 la que determina los tiempos de lectura de los datos generados por los módulos conectados a dicha red. Para los requerimientos específicos de las torres de correlación turbulenta los datos son obtenidos cada minuto. En la microcomputadora Gumstix la red de sensores es canalizada a través del puerto `ttyS0`. Esta microcomputadora genera un archivo diario con nombre `aaaammddi_tt.dat`, donde

Parámetro	Descripción
aaaa	4 dígitos de año
mm	2 dígitos del mes
dd	2 dígitos del día
i	Número de módulo
tt	Etiqueta de identificación de Torre

Tabla 6.1: Descripción del nombre de archivo asignado a los módulos de adquisición de datos

Cuando una tarjeta de adquisición de datos recibe el comando de lectura responde con una cadena numérica con el siguiente formato:

```
N T0 T1 T2 ADC1 ADC2 ADC3 ADC4 ADC5 ADC6 ADC7 ADC8<cr>
```

Donde N es el número de muestras tomadas. Los datos almacenados en el archivo contienen el siguiente formato:

HH	MM	N	T0	T1	T2	ADC1	ADC2	ADC2	ADC4	ADC5	ADC6	ADC6	ADC8
12	56	230	0	0	0	0.273	0.332	2.218	0.553	0.661	0.000	0.000	0.908
12	57	223	0	0	0	0.272	0.318	2.218	0.553	0.669	0.000	0.000	0.907
12	58	231	0	0	0	0.271	0.308	2.218	0.554	0.669	0.000	0.000	0.906
12	59	230	0	0	0	0.271	0.303	2.218	0.554	0.663	0.000	0.000	0.905
13	00	231	0	0	0	0.271	0.298	2.218	0.554	0.661	0.000	0.000	0.905
13	01	230	0	0	0	0.270	0.287	2.218	0.555	0.670	0.000	0.000	0.904
13	02	231	0	0	0	0.269	0.284	2.218	0.557	0.657	0.000	0.000	0.903

Tabla 6.2: Archivo de datos crudos (voltajes) provenientes de los sensores. En el nombre del archivo se define el módulo al que corresponde.

### 6.1.2. Archivos del sistema IRGA-Anemómetro

El IRGA (LI-7500) es conectado y sincronizado con el anemómetro ultrasónico (81000V). Este sistema no requiere que los datos le sean solicitados, ya que mientras está activo se encuentra enviando cadenas de datos.

El nombre del archivo asignado a posee el formato `aaaammdd_tt.dat`, donde

Parámetro	Descripción
aaaa	4 dígitos de año
mm	2 dígitos del mes
dd	2 dígitos del día
tt	Etiqueta de identificación de la Torre

Tabla 6.3: Descripción del nombre de archivo asignado a los datos obtenidos del sistema IRGA-Anemómetro.

HH	MM	SS	H2O	CO2	U	V	W	3D-SPEED	AZ	ELEV	SOS	Ts
14	37	0	624	1856	-2.03	0.88	-0.99	2.43	293.50	-29.70	343.04	18.86
14	37	0	624	1857	-1.73	1.21	-0.78	2.25	304.90	-25.40	343.06	18.91
14	37	1	624	1856	-1.70	1.05	-0.70	2.13	301.80	-19.40	343.02	18.83
14	37	1	624	1854	-1.79	0.85	-0.54	2.07	295.40	-15.40	343.02	18.83
14	37	1	623	1855	-1.79	0.73	-0.61	2.04	292.20	-17.60	343.04	18.88
14	37	1	624	1856	-1.49	0.77	-0.60	1.79	297.50	-19.70	343.06	18.91

Tabla 6.4: Archivo de datos generado por el sistema IRGA-Anemómetro. En el nombre del archivo se indica la torre a la cual pertenece.

El anemómetro digital envía una cadena numérica con el siguiente formato:

```
H2O CO2 U V W 3D-SPEED AZ ELEV SOS Ts<cr>
```

Donde H2O es la concentración de vapor  $H_2O$ , CO2 la concentración de  $CO_2$ ; U, V y W son las componentes vectoriales del flujo de viento, 3D-SPEED la rapidez del viento en el plano tridimensional, AZ la componente azimutal, ELEV la elevación angular, SOS la velocidad del sonido en el medio y Ts la temperatura sónica.

La Tabla 6.1.2 muestra un fragmento del archivo 20071702\_IRGA.dat, el cual contiene las mediciones de flujos  $CO_2$  y vapor de  $H_2O$  tomadas durante el día 17 de febrero de 2007. El formato de columnas de este archivo es el que se muestra en la misma Tabla.

## 6.2. Archivos de calibración

El programa encargado de la lectura del puerto ttyS0 (LI-700.exe) se encarga además de realizar un procesamiento inicial con la finalidad de generar datos que serán almacenados en los archivos finales ya no en forma de voltajes analógicos si no en las magnitudes meteorológicas o edáficas correspondientes. Para hacer estas conversiones se emplean los archivos CalibX.dat, donde X es un índice empleado para la identificación del módulo de adquisición, que deben ubicarse en el directorio /termar/programas/, de la Gumstix, junto con los programas ejecutables. Cada línea de datos corresponde a un vector conteniendo los coeficientes de un polinomio de quinto orden de la forma  $a_0 + a_1v + a_2v^2 + a_3v^3 + a_4v^4 + a_5v^5$ , con  $a_0$  en la primera columna y  $a_5$  en la sexta columna. También se incluye un texto donde se da información relativa al sensor al cual

corresponden los coeficientes, esta columna no es leída por el programa durante el proceso de calibración. Un ejemplo de estos archivos es el que se muestra en la Tabla 6.2.

Renglón	Columna 1	Columna 2	Columna 3	Columna 4	Columna 5	Columna 6	Columna 7
1	0.0	6.591471	0.0	0.0	0.0	0.0	LI-190SA
2	0.0	1.0	0.0	0.0	0.0	0.0	LI-200X
3	600.0	184.0	0.0	0.0	0.0	0.0	CS105
4	-40.8065	100.8065	0.0	0.0	0.0	0.0	HMP45(Temp)
5	0.0	100.8065	0.0	0.0	0.0	0.0	HMP45(RH)
6	0.0	1.0	0.0	0.0	0.0	0.0	---
7	0.0	1.0	0.0	0.0	0.0	0.0	---
8	55.881	-88.496	86.187	-57.222	19.657	-2.67	Thermistor 1

Tabla 6.5: Los archivos de calibración se organizan en columnas conteniendo los coeficientes de un polinomio de quinto orden y comentarios que no son leídos por el programa principal.

### 6.3. Archivos de configuración

Los programas ejecutables contenidos en la Gumstix, configurados mediante archivos donde se especifica cual puerto serial será leído y cuantos módulos se encuentran en su red RS-485, para esto al primer módulo en ser leído se le deberá nombrar como *módulo 0*; mientras en para el caso del sistema IRGA-Anemómetro solo se requiere indicar el puerto al que está conectado.

El archivo `irga.cfg` tiene el siguiente formato:

No. de renglón	Columna 1	Columna 2
1	IRGA	Etiqueta del archivo de datos
2	/dev/ttyS2	Puerto serie a utilizar

Tabla 6.6: Solo se requiere indicar el puerto serial que será leído.

Con algunas modificaciones referentes a las trayectorias en el sistema de directorios de Linux, los programas para la lectura de módulos de adquisición y del anemómetro ultrasónico son ejecutables en una computadora de escritorio. Para el caso donde sea leído a través de un convertidor Serial-USB conectado al `ttyUSB0`) esta trayectoria deberá sustituirse por: `/dev/tts/USB0` (revisar apéndice D.1).

El archivo `torres.cfg` tiene el siguiente formato:

No. de renglón	Columna 1	Columna 2
1	ID	Identificador de torre
2	/dev/ttyS0	Puerto serie a utilizar
3	n	Número de módulos en la red RS-485
4	$m_1$	Identificador de módulo 1
5	$m_2$	Identificador de módulo 2
⋮	⋮	⋮
n+3	$m_n$	Identificador de módulo n

Tabla 6.7: En los archivos de configuración se indica la cantidad de módulos conectados a la red de datos, así como la etiqueta correspondiente a cada uno de ellos y el puerto serial a través del cual son leídos. Pueden conectarse hasta 8 módulos en un mismo puerto. El ID puede corresponder a la torre marina (MAR), torre terrestre (TER) o la torre móvil (MOV) y se verá reflejado en los nombres de los archivos de datos.



## Capítulo 7

# Datos experimentales

Desde el mes de octubre de 2006 ha sido instalada una torre de prueba en una de las áreas verdes del CI-CESE, con el propósito de probar los sistemas de adquisición de datos, energía solar y de transmisión por radio. Esta estación ha recogido datos de humedad relativa, temperatura del aire, temperatura del suelo, presión atmosférica y radiación fotosintéticamente activa. También han sido registrados los parámetros más importantes relativos a los flujos de aire como la rapidez del viento, componentes vectoriales tridimensionales, orientación y elevación, velocidad y temperatura sónica, así como las concentraciones de vapor de agua y  $CO_2$  en el ambiente.

Los módulos de adquisición entregan cada minuto (cuando le es requerido) el promedio de aproximadamente 230 muestras. El número de muestras almacenadas depende del tiempo que el maestro tarde en pedirle sus datos, ya que una vez que los envía elimina los datos almacenados para iniciar un nuevo sumatoria de datos. Los DACs del IRGA operan a una tasa constante de refresco de 300 Hz, mientras que los parámetros de comunicación del anemómetro ultrasónico son configurables por el usuario dependiendo de la velocidad necesaria. La tasa de transferencia de datos del anemómetro ha sido configurada a 38,400 baudios.

El proceso de adquisición de datos, tanto de los sensores analógicos como del sistema IRGA-Anemómetro, son presentados en gráficas que muestran los promedios por minuto (Figuras 7.1 y 7.2) ya que para la computadora es complicado procesar la enorme cantidad de información contenida en los archivos de datos crudos. En la Figura 7.3 se presenta el seguimiento dado a los parámetros viento y concentración de gases durante

un periodo de una semana aproximadamente, donde puede observarse un comportamiento cíclico diario de las variables medioambientales.

La gráfica de la Figura 7.1 muestra los datos obtenidos por el sistema de adquisición de datos de los sensores de presión, temperatura y humedad relativa del aire y temperatura del suelo. Los datos se muestran respecto al día del año al cual corresponden, de tal forma que el día primero de febrero corresponde al día número 32 del año. La tarjeta de adquisición obtiene en promedio 127 muestras por minuto de cada uno de los sensores y se presentan solamente las medias estadísticas de cada variable.

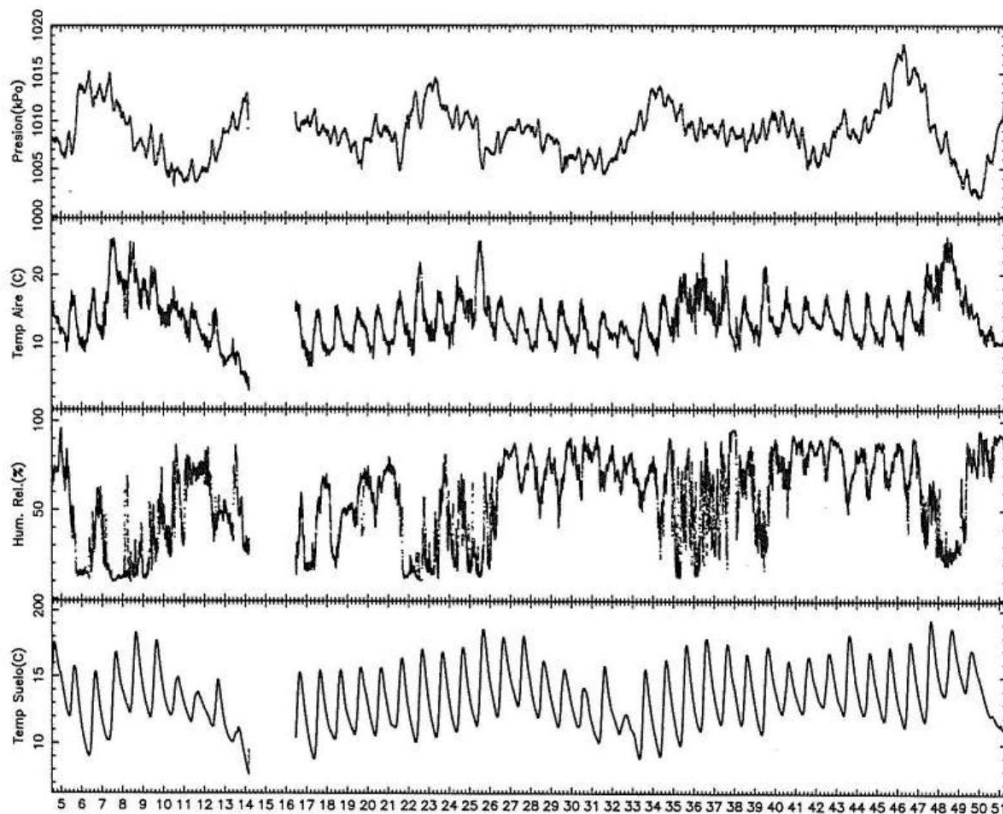


Figura 7.1: Señales adquiridas por el sistema de adquisición de datos y almacenamiento.

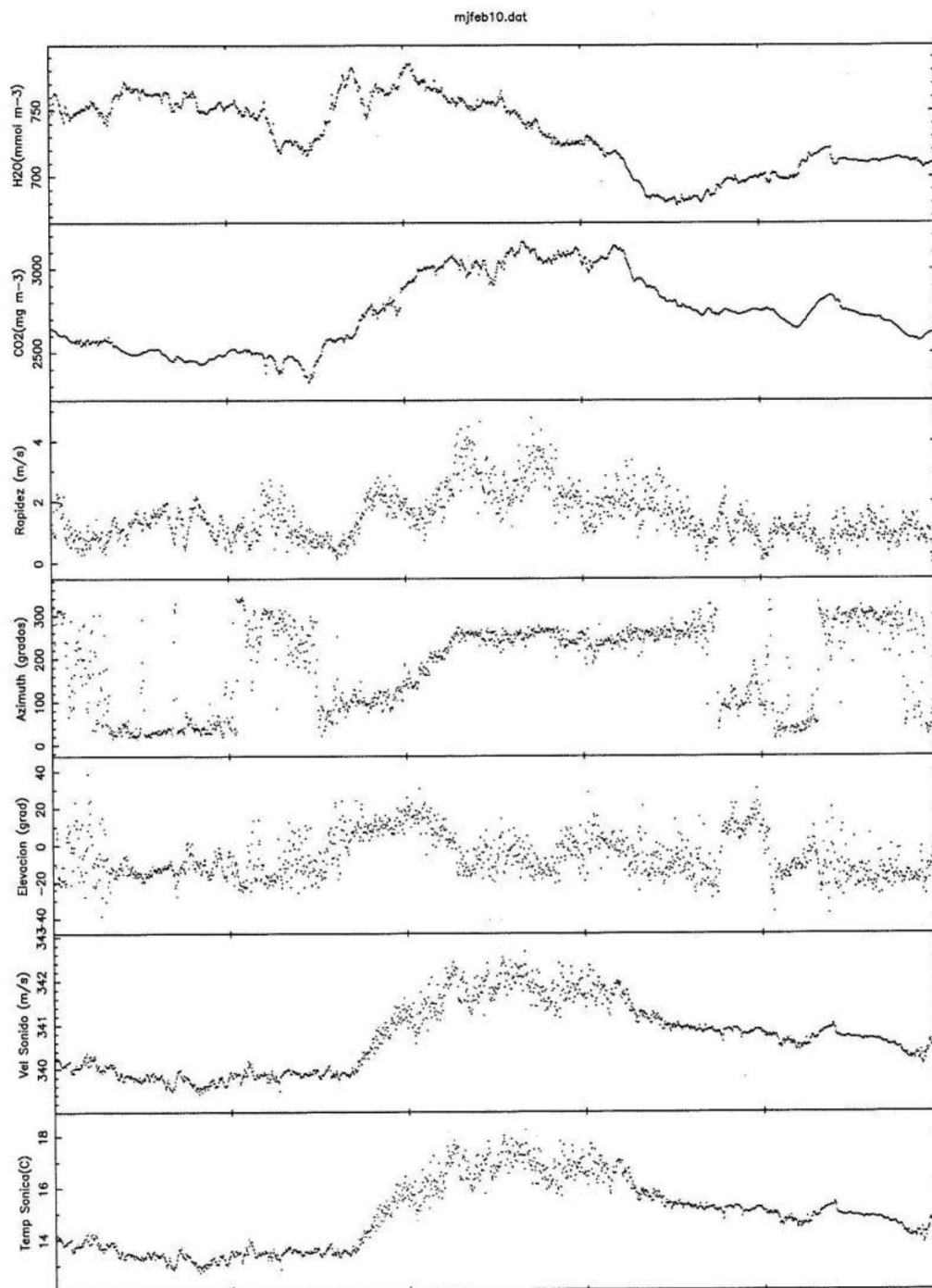


Figura 7.2: Comportamiento de los parámetros de viento el 10 de febrero de 2007.

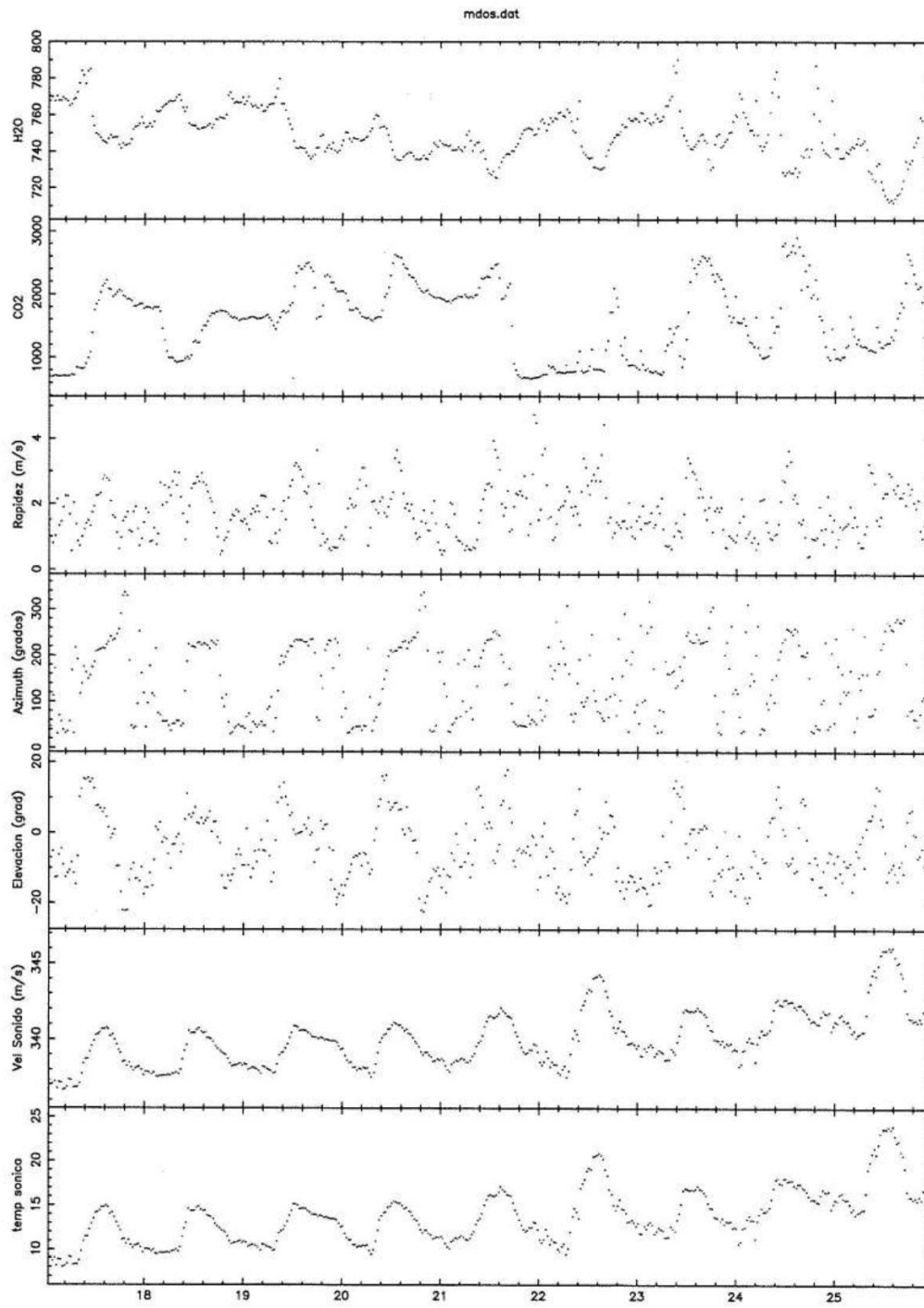


Figura 7.3: Datos obtenidos con los sensores IRGA-Anemómetro del periodo del 17 a 26 de enero de 2007 en la TCT experimental.

## Capítulo 8

# Conclusiones

El análisis de los efectos del cambio global requiere de nuevos sistemas de medición cada vez más eficientes y baratos que permitan la realización masiva de este tipo de estudios. Como resultado del presente trabajo de tesis se han producido sistemas de adquisición y almacenamiento de datos para el estudio de fuentes y sumideros de  $CO_2$  cuyas capacidades son comparables a las de sus antecesores ubicados en Sky Oaks (SDSU) y La Paz (CIBNOR).

Los objetivos principales fueron la disminución del espacio físico ocupado por dichos sistemas así como el costo total de los sistemas de medición. De esta manera fueron construidas tarjetas de adquisición de datos con dimensiones de 7x6 cm, las cuales pueden ser acopladas a las tarjetas de acondicionamiento de señales, cuyas medidas son de 10x6 cm. Una TCT (Torre de Correlación Turbulenta) requiere dos módulos de adquisición de datos, cuya lectura es temporizada mediante la microcomputadora comercial Gumstix WaySmall 200ax (8x3.5x1.5 cm). El sistema completo de medición, almacenamiento y procesamiento de datos es contenido dentro de la denominada caja de instrumentación, con medidas de 42x33x20 cm. Debe hacerse notar que esta caja es de material plástico calidad NEMA-4 resistente a condiciones de interperie. Dentro de este contenedor existe una gran cantidad de espacio no utilizado que permite realizar conexiones y trabajar con comodidad. Esto reduce el tamaño total del sistema en proporción de 1 a 3, comparado con las TCT del CIBNOR y la SDSU (2 cajas de 45x40x22.5 cm cada una).

Este proyecto ha sido pionero en el desarrollo de alternativas económicas para la medición de flujos de carbono

mediante las TCT. Para ello fue reemplazada la instrumentación y dispositivos de procesamiento y almacenamiento por opciones económicamente viables. El sistema descrito en los capítulos anteriores sustituye una Laptop, un Data-Logger (CR10x), un dispositivo de memoria externa y un multiplexor (todo ello con un precio aproximado de 3,500 dólares), por uno cuyo costo total de producción y ensamblaje alcanza 400 dólares, reduciendo el costo del sistema de medición y registro de datos en proporción de 8.5 a 1, así como un ahorro sustancial en cableado y diversos tipos de adaptadores empleados por el resto de la TCTs. La sustitución de las cajas de instrumentación originales reduce este consumo de 610 a 82 dólares, adicional a los costos mencionados.

En la TCT-CICESE se dispone también de un colector de energía solar ajustado a las necesidades del sistema, evitando así que dichos elementos sean subutilizados, como ocurre en las TCTs empleadas por la red AMERIFLUX, donde para abastecer las necesidades de energía de los componentes ahí empleados se requiere de convertidores DC/AC y subsecuentemente de convertidores AC/DC (debido a que los dispositivos empleados disponen de conexión eléctrica a pared). De esta forma la cantidad de paneles solares ha sido reducida también. El tipo de batería eléctrica para almacenamiento de carga fue elegida conforme las necesidades propias, lo cual reduce la cantidad y capacidad de las pilas empleadas usadas por la TCT.

Esta disminución de costos se espera pueda alentar a los diferentes organismos de investigación, en México y otros países con problemáticas sociales y económicas similares, a la realización de este tipo de estudios.

# Bibliografía

- [1] Lenton TM., Huntingford C., 2003. *Global terrestrial carbon storage and uncertainties in its temperature sensitivity examined with a simple model*. Global Change Biology, 9. 1333-1352.
- [2] Baldocchi DD, 2003. *Assesing the eddy covariance technique for evaluating carbon dioxide exchange rates of ecosystems: past, present and future*. Global Change Biology, 9. 479-492.
- [3] Falkowski, 2000. *The global carbon cycle: A test of our knowlegde of Earth as a System*. Science 290, 291-296.
- [4] Michel R., Bohigas J., Arroyo E. y Zazueta S., 2001. *The meteorological stations of the 1.5 an 0.84m telescopes of the OAN: Description and results*. Rev. Mex. Astron. Astrof. 37. 165-171
- [5] Michel R., Hiriart D. y Chapela A., 2003. *Four years of meteorological measurments at San Pedro Mártir Observatory.*. Rev. Mex. Astron. Conf. Ser. 19, 99-102.
- [6] Chapela A., 2004. *Estación meteorológica para el telescopio de 84cm de San Pedro Mártir.*. Tesis de licenciatura, Fac. de Ingeniería UABC.
- [7] Vourlitis G. et al., 2004. *Seasonal variations in the evapotranspiration of a transitional tropical forest of Mato Grosso, Brazil*. Vol. 38, No. 0, 3-4.
- [8] Vourlitis G. L., Priante-Filho N., Hayashi M. S., De Sousa Nogueira J., Casseiro F. T., Campelo Jr. J. H., 2001. *Seasonals variations in the net ecosystem CO<sub>2</sub> exchange of a mature amazonian tropical transitional forest*. Funct. Ecol., 15, 388-395.
- [9] Verma S. B., 1990. *Micrometeorological methods for measuring surface fluxes of mass and energy*. Remote Sensing Reviews, 5. 99-115.
- [10] Martínez-Osuna J., 2005. *Estaciones meteorológicas y de monitores de temperaturas para los telescopios del Observatorio Astronómico de San Perdo Mártir*. Tesis de licenciatura. Fac. de Ingeniería UABC.
- [11] J. Steinhart, S. Hart, 1968. *Calibration curves for thermistors*. Deep Sea Res., 15, 497-503.
- [12] Chapra S., Canale R., 1999. *Métodos Numéricos para ingenieros*. McGraw Hill, 465-485.
- [13] Gutiérrez J. L., 1980. *Estadística*. Cultural, S. A. Ediciones.



## Apéndice A

# Esquemáticos y PCBs

Los esquemáticos presentados en este apéndice fueron realizados con la ayuda del editor de circuitos impresos **Eagle 3.55r3**. Las tarjetas de adquisición de datos y el convertidor RS-232/RS-485 fueron impresas en los Estados Unidos de América a través de la compañía **PCB4u** dedicada a la fabricación de circuitos impresos. El resto de las tarjetas de acondicionamiento y de prueba fueron impresas en el Instituto de Astronomía de la UNAM, en Ensenada. La calidad alcanzada por la entidad comercial como por el laboratorio de Electrónica del IA-UNAM es comparable, excepto por la incapacidad del sistema local para conectar ambas caras del circuito impreso a través una *vía*, esto debido al que el método empleado consiste en devastado de superficie alrededor de la pista sin de desposición de material sobre la tarjeta.

Todos las tarjetas han sido diseñadas con el objetivo específico de medir las variables necesarias para la determinación de fuentes y sumideros de  $CO_2$ , sin embargo pueden ser utilizadas en aplicaciones distintas para las cuales fueron contruidas.

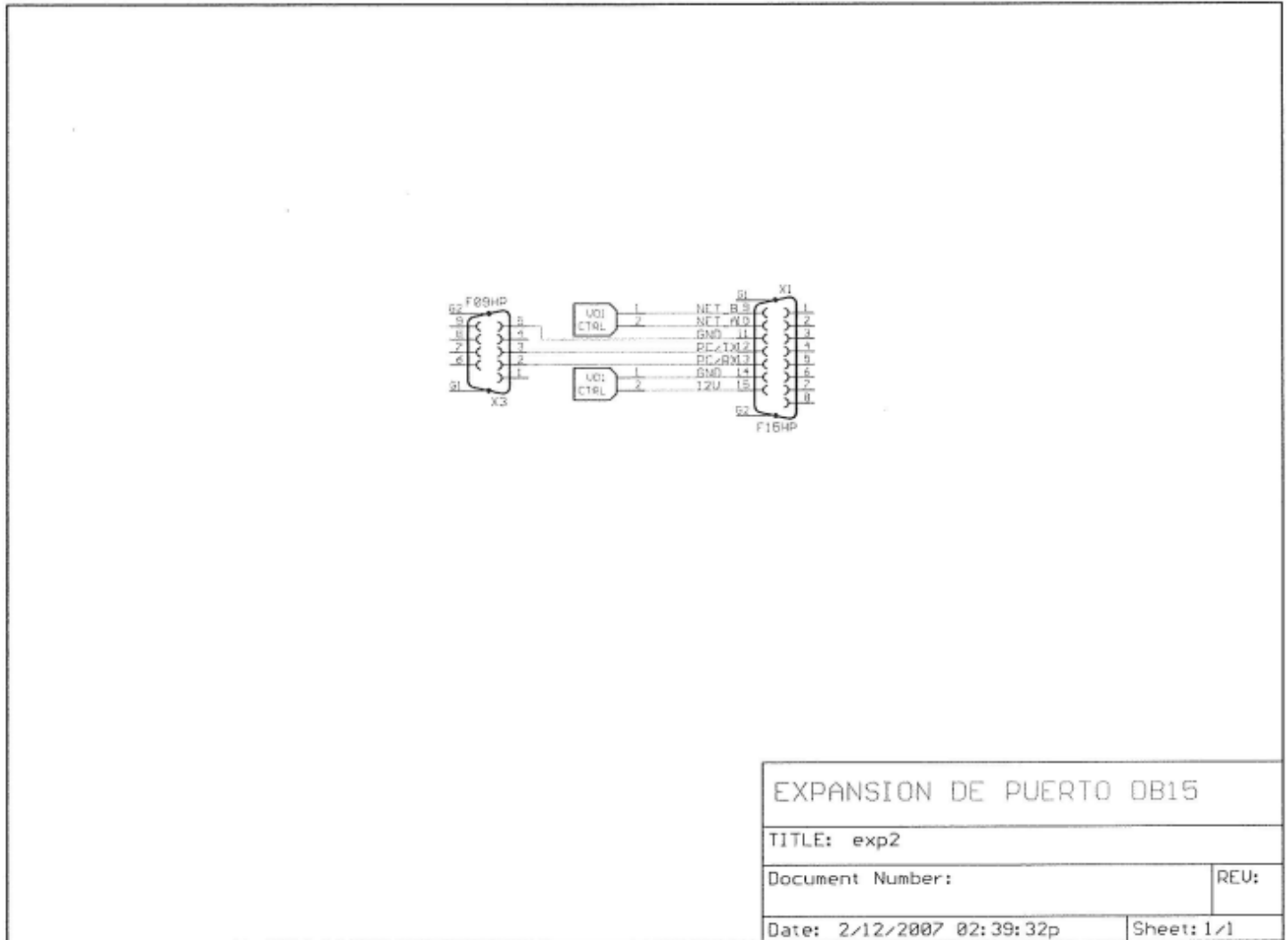


Figura A.1: Esquemático:Tarjeta de acceso a ductos.

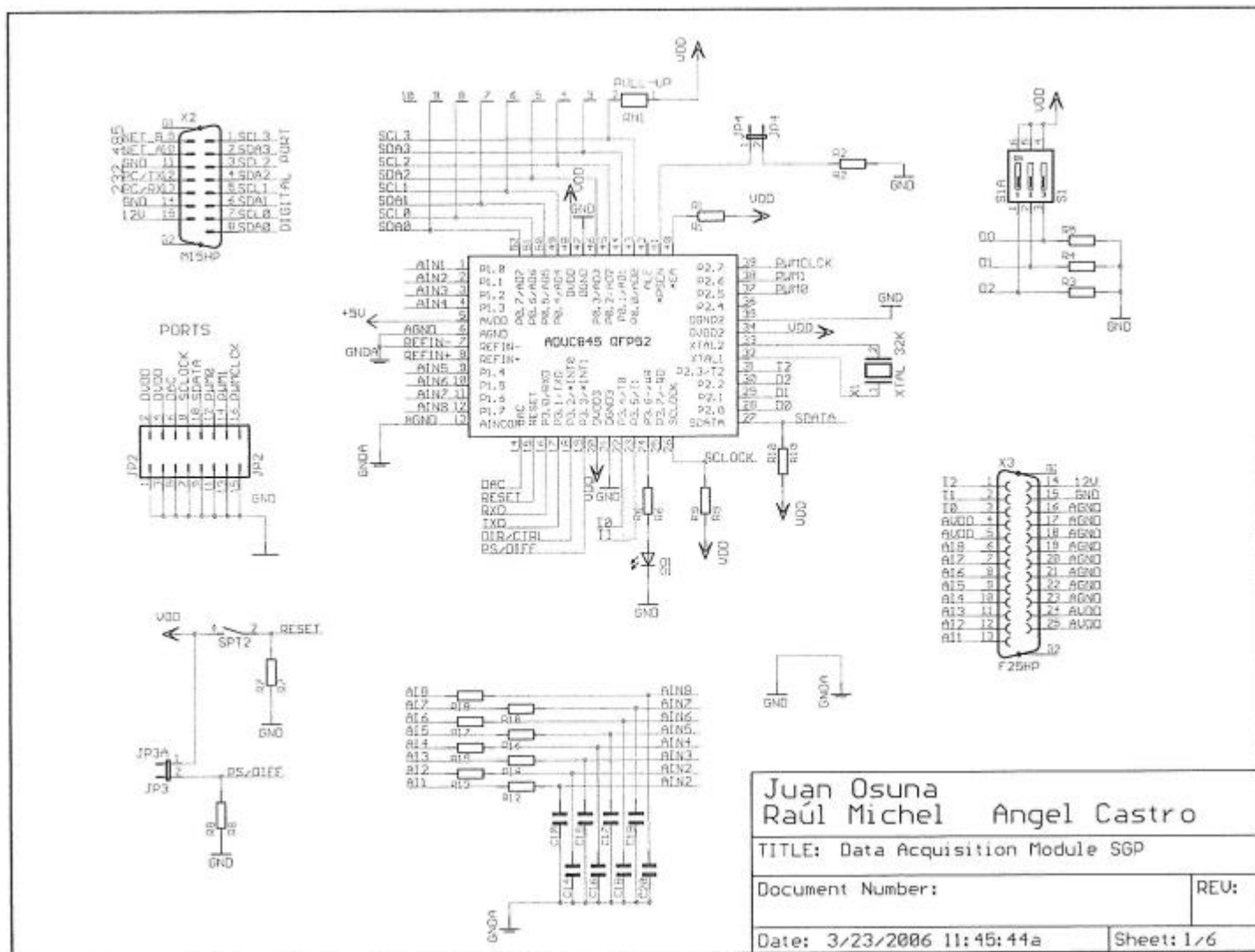


Figura A.2: Esquemático: Tarjeta de adquisición, página 1/2.

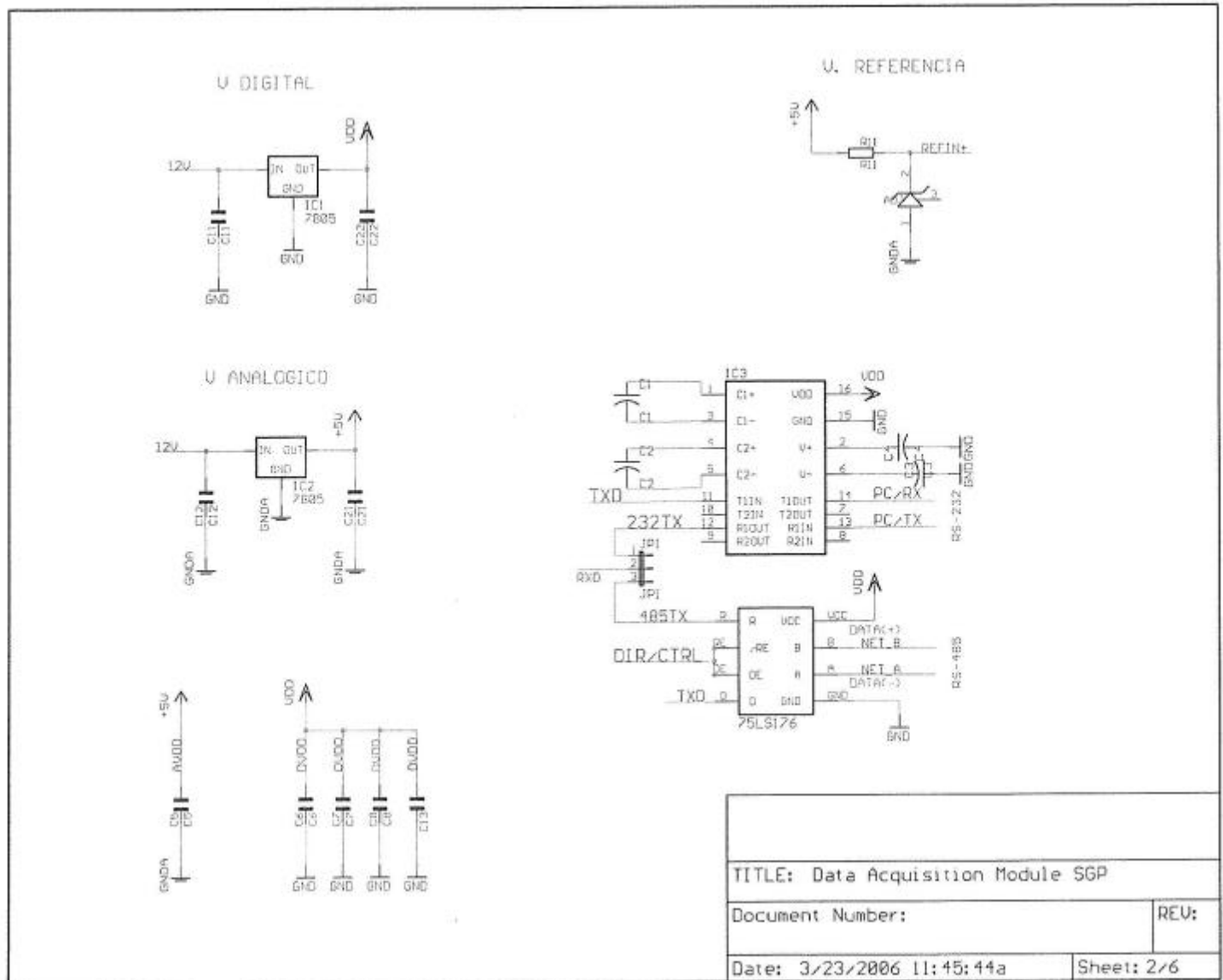
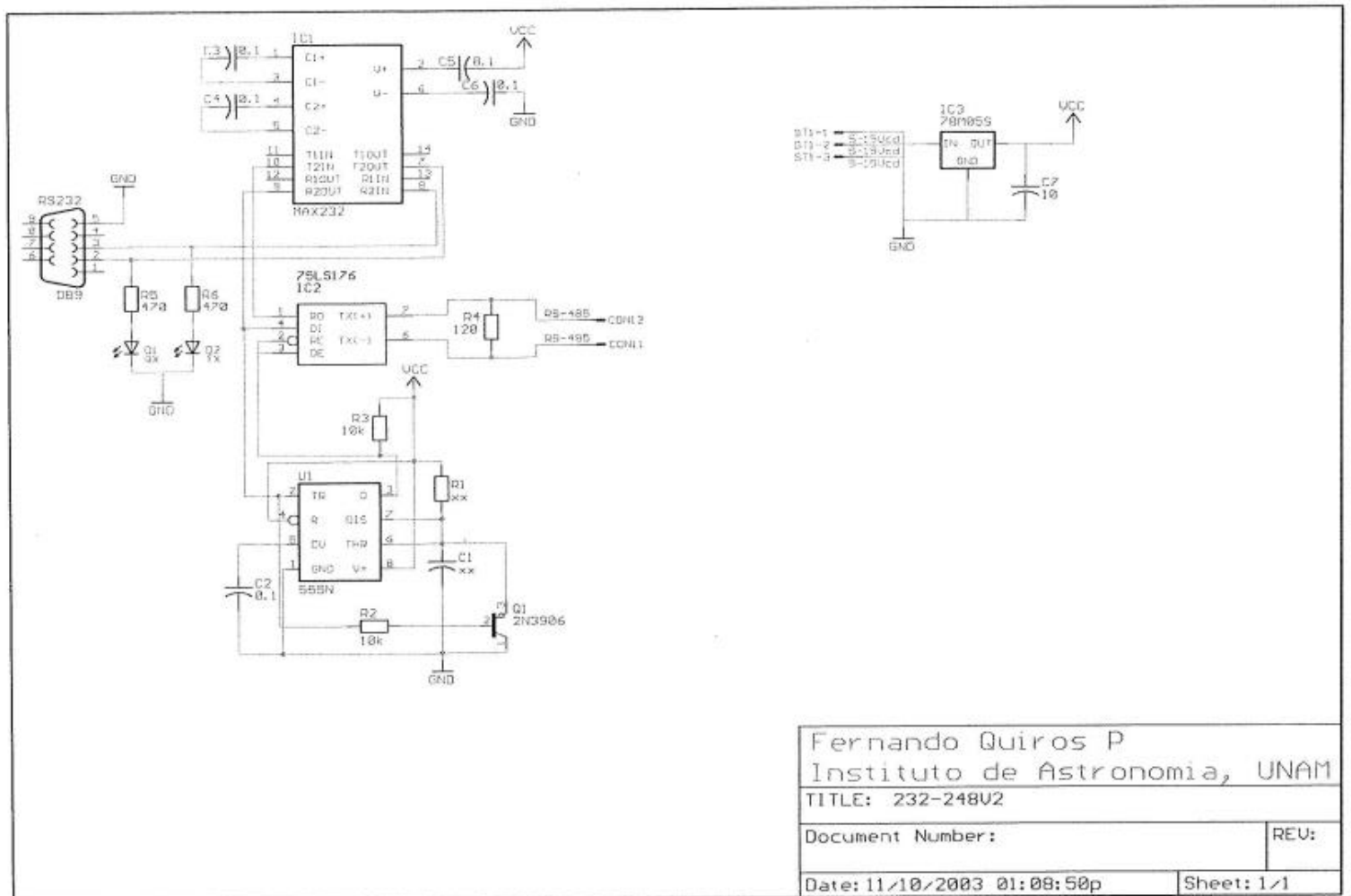


Figura A.3: Esquemático: Tarjeta de adquisición, página 2/2.



Fernando Quiros P	
Instituto de Astronomia, UNAM	
TITLE: 232-248V2	
Document Number:	REV:
Date: 11/10/2003 01:08:50p	Sheet: 1/1

Figura A.4: Esquemático: Convertidor RS-485/RS-232.

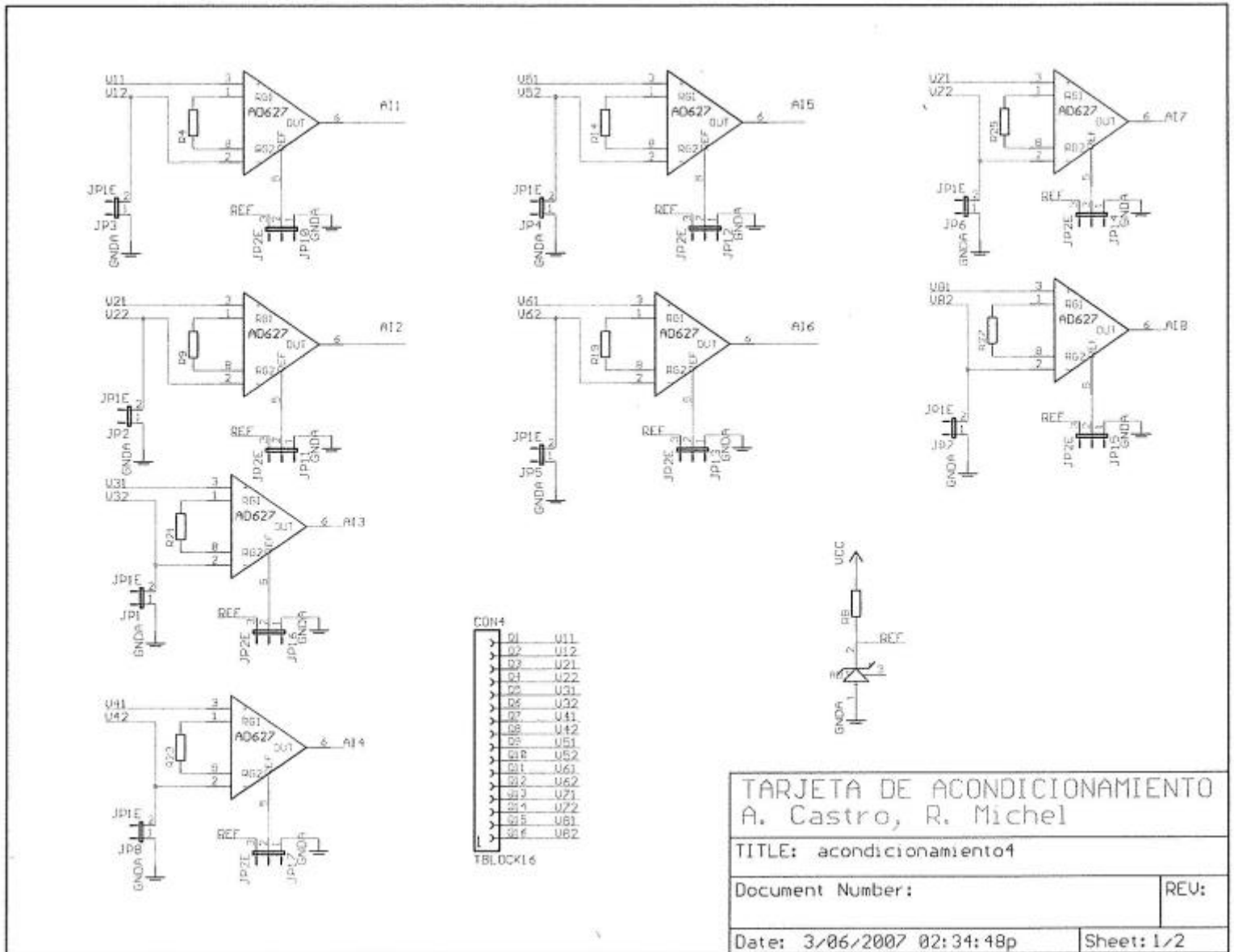


Figura A.5: Esquemático: Tarjeta de acondicionamiento de señales, página 1/2.

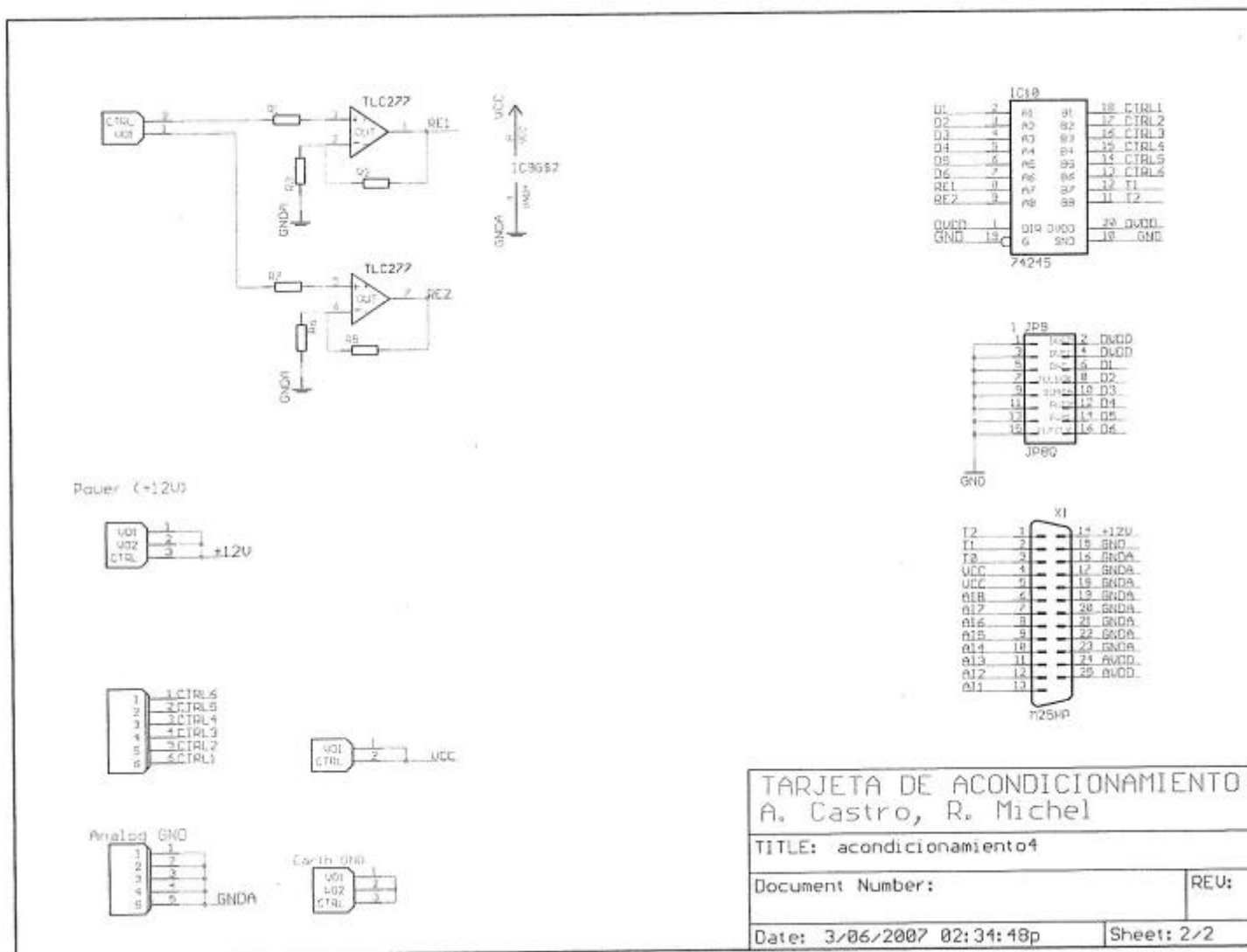


Figura A.6: Esquemático: Tarjeta de acondicionamiento de señales, página 2/2.

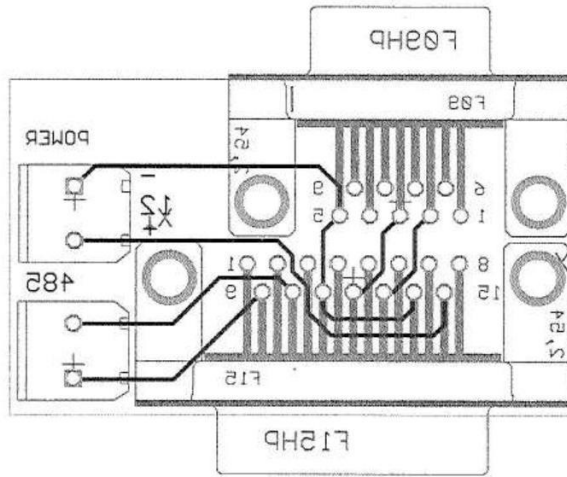


Figura A.7: PCB:Tarjeta de acceso a ductos.

+

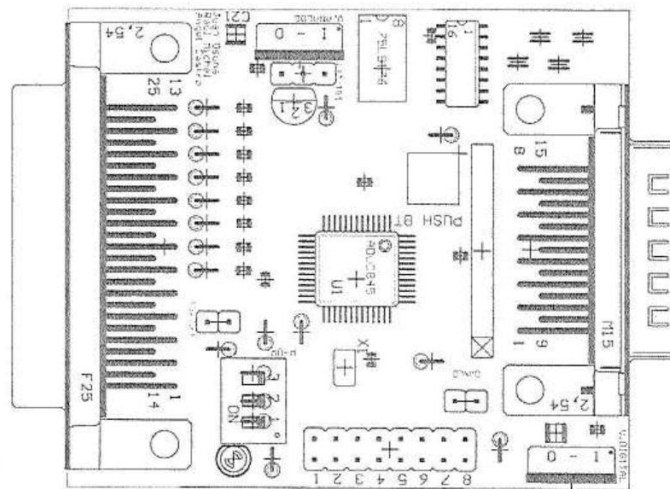


Figura A.8: PCB:Tarjeta de adquisición, máscara de componentes.

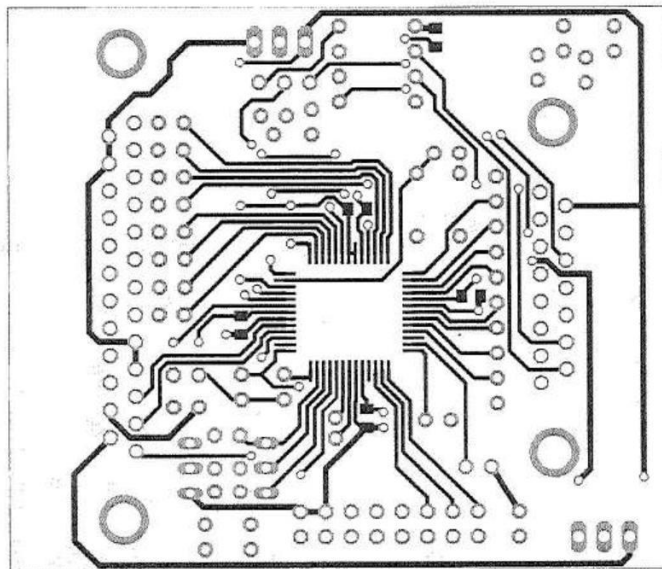


Figura A.9: PCB:Tarjeta de adquisición, capa superior.

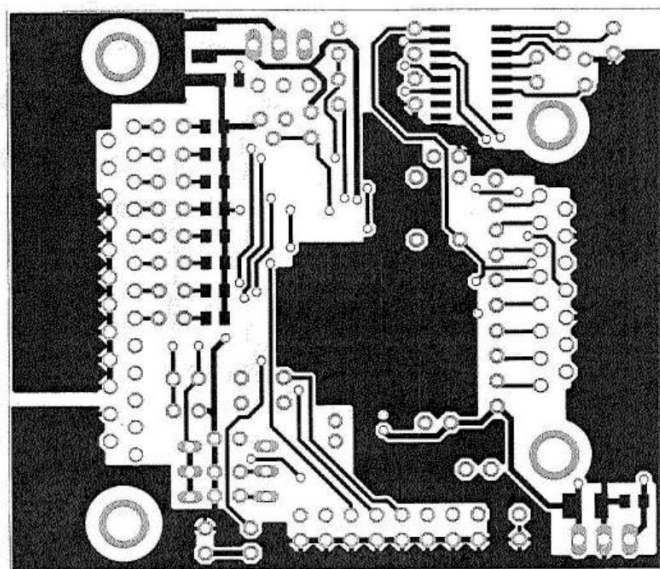


Figura A.10: PCB:Tarjeta de adquisición, capa inferior.

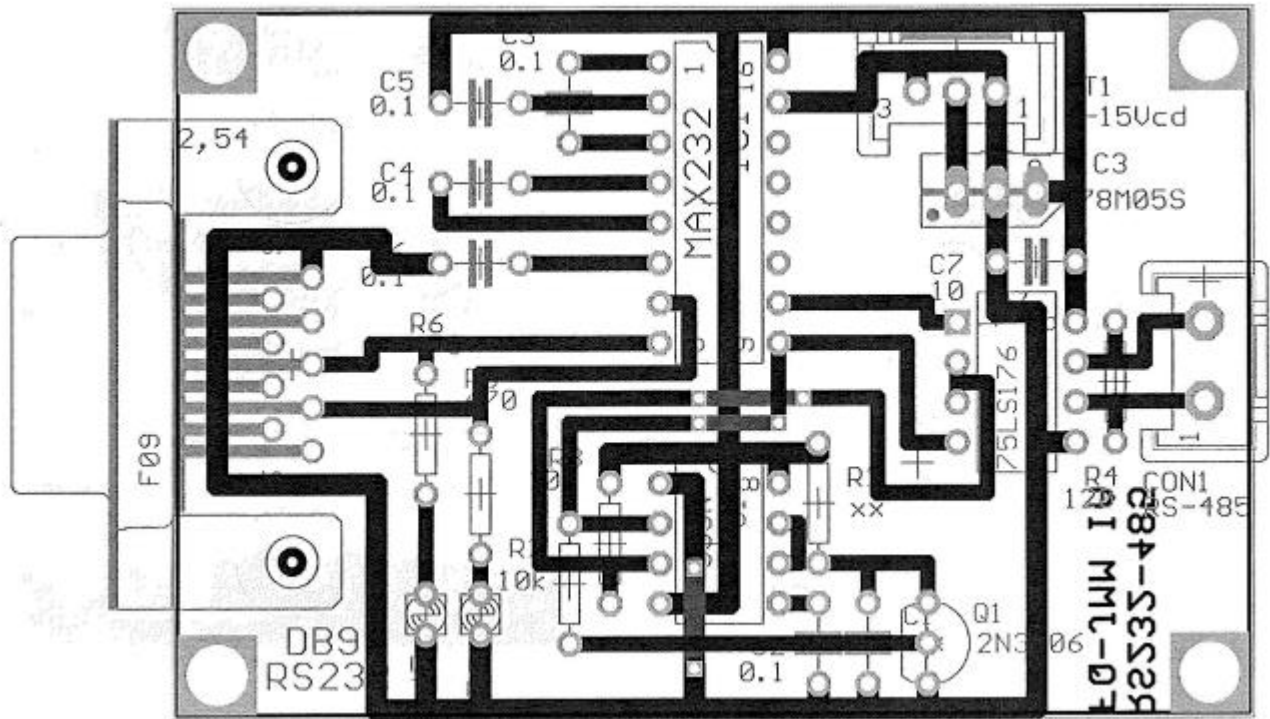


Figura A.11: PCB:Convertidor RS-485/RS-232 (ambas capas).

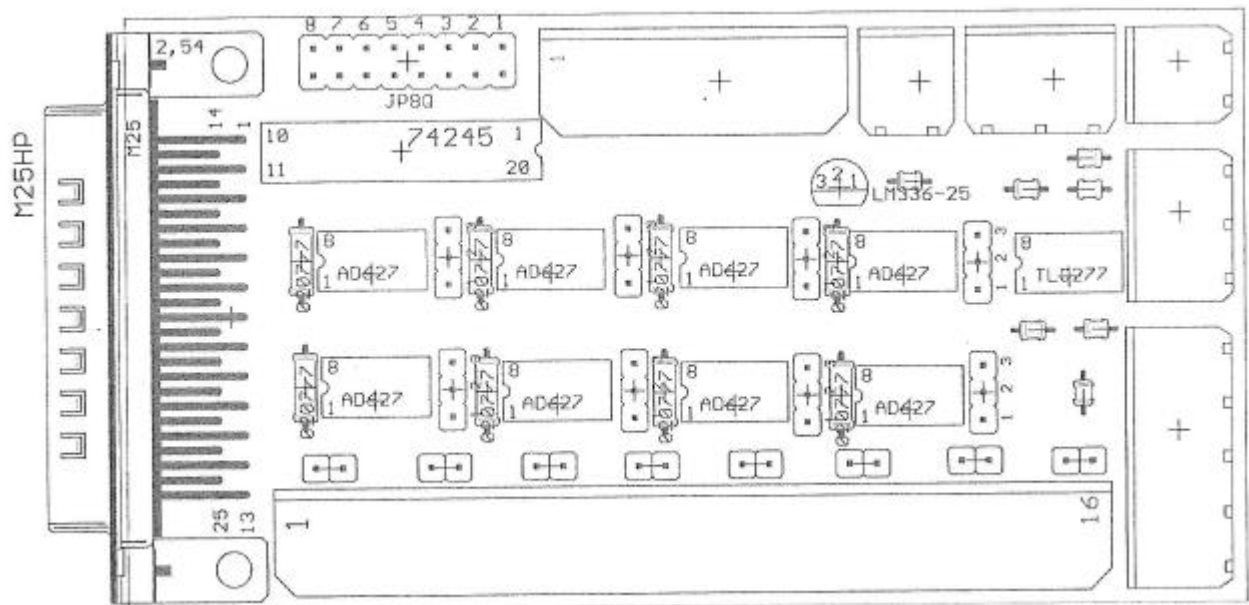


Figura A.12: PCB: Tarjeta de acondicionamiento de señales, máscara de componentes.

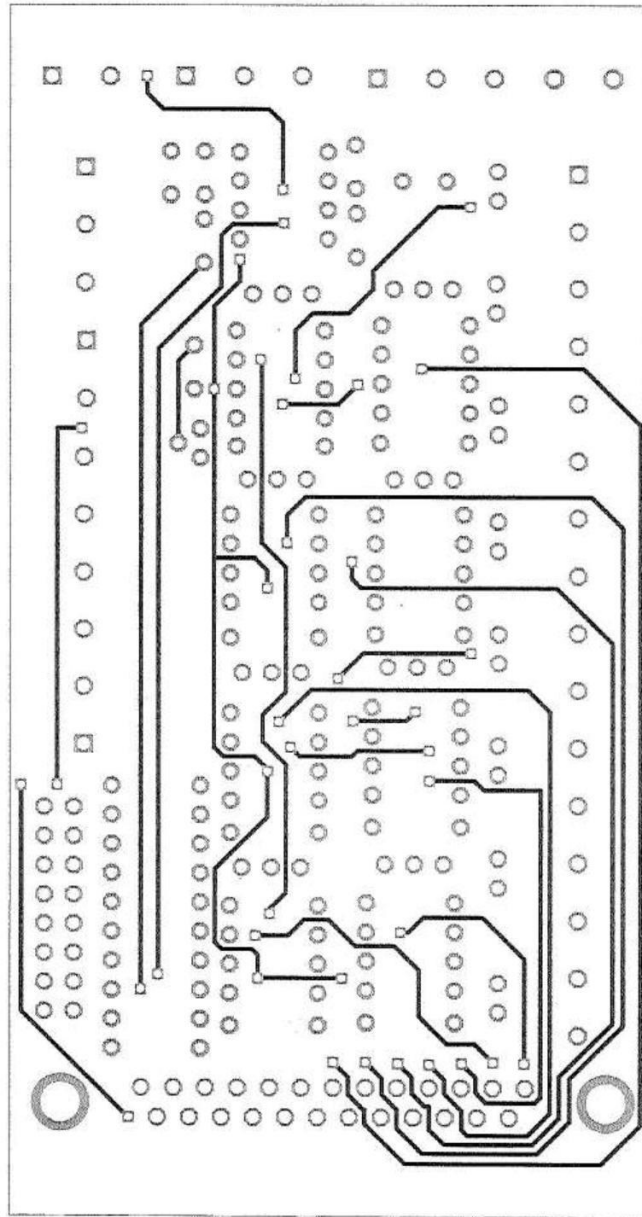


Figura A.13: PCB: Tarjeta de acondicionamiento de señales, capa superior.

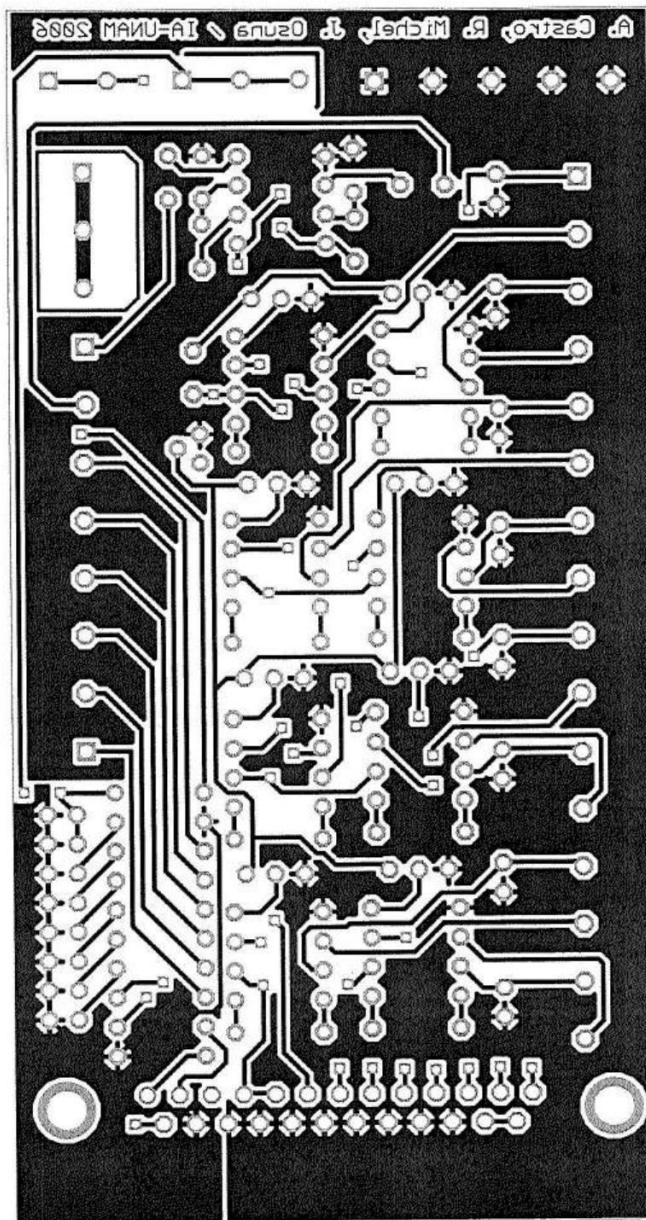


Figura A.14: PCB: Tarjeta de acondicionamiento de señales, capa inferior.



## Apéndice B

# Programación

Los programas empleados en este trabajo de tesis fueron elaborados en el lenguaje de programación de alto nivel C++ utilizando los editores de texto Keil 2.0 y Keil 3.0. El editor Keil posee características que lo hacen un editor muy poderoso cuando se realizan programas empleados en microcontroladores. El sistema requiere para su funcionamiento dos programas computacionales: uno contenido en el microcontrolador, cuya función es la de leer los canales de conversión analógica a digital del ADuC848, y otro, que es fijado en dispositivo maestro, cuya función es la de regular el tráfico de información en el ducto RS-485, así como leer las cadenas de datos enviadas por los diferentes módulos de adquisición de datos colgados éste ducto y del sistema IRGA-Anemómetro.

El programa contenido en la estación de trabajo (maestro) requiere ser compilado mediante un compilador cruzado para su posterior transferencia hacia la microcomputadora. Debido a su limitada capacidad, este tipo de sistemas integrados requieren que los programas que ejecutarán sean compilados externamente. Para ello se emplea el `arm-linux-gcc` contenido en el `buildroot` de la Gumstix. Los archivos de calibración y de configuración mencionados en las secciones 6.2 y 6.3 pueden ser editados externamente y transferidos a la microcomputadora o editados en el sistema de la Gumstix empleando el editor VI.

## B.1. Programas empleados en las TCT

### B.1.1. Programa del microcontrolador ADuC848

```
//=====
// Programa desarrollado en: C++
// R. Michel, A. Castro, J. Martinez. Mayo 2006.
// Inclusion de librerias y cabeceras del programa. sio.h> Serial Input-Output
// funciones.h > prototipos de las funciones utilizadas pro este programa.
//=====
#include <ADuC845.h>
#include <stdio.h>
#include <math.h>
#include "sio.h"
#include "funciones.h"
//=====
// Var. Globales son declaradas al principio del programa
//=====
extern float sqrt (float val);
sbit P0_7 = 0x87;
sbit P0_6 = 0x86;
sbit P0_5 = 0x85;
sbit P0_4 = 0x84; // Conf. del puerto 0 para sensores de temperatura
sbit P0_3 = 0x83; // Nombre "alias" a cada uno los pines del puerto
sbit P0_2= 0x82;
sbit P0_1 = 0x81;
sbit P0_0 = 0x80;
sbit PCTL = 0xB2; // Direccion de Dir/Ctrl de 485 (mapa P3.2)
sbit P3_3 = 0xB3; // Bit de control (mapa P3.3)
sbit P3_4 = 0xB4; // Direccion de T0 (mapa P3.4)
sbit P3_5 = 0xB5; // Direccion de T1 (mapa P3.5)
sbit P1_0 = 0x90; // Direccion de T2 (mapa P1.0)
sbit LED = 0xB6; // Direccion del LED (mapa: P3.6)
int N,pin3;
unsigned int b,c,e;
unsigned int ciclos1,ciclos2;
char h,i,j,ID,p3,bus,index=0,comando[6];
bdata int i2c_data; // Declaracion de variable de signo de I2C
bdata unsigned char i2c_addr=0x48; // Dir. default del sensor I2C=01001000
bdata unsigned char pointer=0x04; // Pointer hacia Control Reg=00000100
bdata unsigned char control=0x60; // Control/Status Register 01100000
sbit signo =i2c_data^15 ;
xdata unsigned int a;
xdata float dig_temp,adc_volts,volts;
//-----
//Var. destinadas para resultados, 8 adc y 12 sensores digitales LM73
//-----
xdata float sum[8],sum_2[8];
xdata float temp0[4],temp0_2[4];
xdata float temp1[4],temp1_2[4];
```

```

xdata float temp2[4],temp2_2[4];
xdata float temp3[4],temp3_2[4];
//=====
// Seccion principal de codigo: Configura uC, sensores, y verifica comunicacion
//=====
void main(void)
{
    configura_micro();
    configura_termometros();
    return_pointer_to_cero();
    printf ("Configuracion lista PADTEF Ver1.0, ID = %c\n\r",ID);
    PCTL=0;
    revisa_serie();          // Inicia Muestreo de sensores
}
//=====
// Conf. basica de uC: BaudRate, XRAM, Timers, ADC, Identificacion del Modulo
//=====
void configura_micro(void)
{
    CFG834=0x81;          // Registro para configurar la XRAM interna
    com_initialize();    // Inicializa I/O serial usando el TIMER 3
    com_baudrate();      // 9600 baudios
    T0 = 1;              // Habilita TIMER 0
    T1 = 1;              // Habilita TIMER 1
    T2 = 1;              // Habilita TIMER 2
    SF = 0x200;
    ADCCON1 = 0x27;      // Full Buffer, Unipolar, 0-2.56V range
    ADCCON2 = 0x40;      // Refin+/-, Ain1->Aincom (PseudoDiferencial)
    ADCMODE = 0x22;      // Conversion sencilla en el canal principal
    EXO = 0;             // Deshabilita la interrupcion INTO
    ITO = 1;             // INTO interrupt edge triggered
    ETO = 0;             // Deshabilita el timer 0 INT
    EA = 1;              // Habilita interrupciones (P3.2=1)
    LED ^= 0x0;
    PCTL = 0;           // Bit de control de dir. de mensaje (RS-485)
    lee_ID();           // Lee direccion del modulo
    inicializa_variables(); // Limpia variables
    elige_conteo();     // Lee bit de ctrl para conf. de los cont/temp
    configura_contadores(); // Configura contadores/temporizadores
}
//=====
// Com. Serial: Lee Identif. y establece comunicacion por el puerto
//=====
void revisa_serie()
{
    ciclos1 = 25;
    ciclos2 = 100;
do          // ciclo infinito
{
    for(i=0; i<=ciclos1; i++)

```

```

for(j=0; j<=ciclos2; j++)
{
  while (com_rbuflen()>=1)
  {
    comando[0] = comando[1];
    comando[1] = comando[2];
    comando[2] = comando[3];
    comando[3] = comando[4];
    comando[4] = comando[5];
    comando[5] = com_getchar();
    if (comando[0]=='.' && comando[1]=='2' &&
        comando[2]=='M' && comando[3]==ID &&
        comando[5]==';')
    {
      if (comando[4] == 'V')
      {
        N--;
        muestrea_contadores();
        TH0=0;TL0=0;TH1=0;TL1=0;TH2=0;TL2=0; //Tx reset
        manda_resultados();
        inicializa_variables();
        i=0; j=0;
        PCTL=0;
      }
      if (comando[4] == 'I')
      {
        printf("\n\rPADTEF Ver1.0 Soy el modulo:%c \r\n",ID);
        PCTL=0;
      }
    }
  }
}
LED ^= 1;
monitorea_sensores();          // Muestrear sensores analogicos y digitales
N++;                          // Incrementa contador de muestras.
LED ^= 0;
} while(1);
}
//=====
void lee_ID(void)
{
  ID=(P2&0x07);                //ID del microcontrolador (comando [3])
  ID = ID + 48;
}
//=====
void inicializa_variables(void)
{
  N=1;
  a=0;
  ADCOH=0;
}

```

```

    ADCOM=0;
    adc_volts=0;
    for(j=0;j<4;j++)
    {
        temp0[j]=0; temp0_2[j]=0;
        temp1[j]=0; temp1_2[j]=0;
        temp2[j]=0; temp2_2[j]=0;
        temp3[j]=0; temp3_2[j]=0;
    }
    for(j=0;j<8;j++) {sum[j]=0; sum_2[j]=0;}
}
//=====
void elige_conteo(void)
{
    p3 = P3_3;           //Elige modo de los contadores
    pin3 = (int) p3;
}
//=====
void configura_contadores(void)
{
    if (pin3 == 0)
    {
        TMOD = 0x51;    // Conf. del T0 y T1:T0 (temporizador) Y T1 (contador)
    }
    if (pin3 == 1)
    {
        TMOD = 0x55;    // Configuracion del timer 0 y timer 1 (contadores)
        TCON = 0x50;
    }
    T2CON = 0x06;       // Configuracion del timer 2.
    TH0=0; TLO=0;
    TH1=0; TL1=0;
    TH2=0; TL2=0;
}
//=====
void monitorea_sensores(void)
{
    lee_termometros(); // Procesa sensores de temperatura (digitales)
    lee_analogicos();  // Procesa sensores analogicos.
    if (p3 == 1){ lee_pulsos(); }
}
//=====
void lee_analogicos(void)
{
    for (i=0; i<8; i++)
    {
        adc_volts=lee_ADC(i);
        sum[i]=adc_volts+sum[i];
        sum_2[i]+=adc_volts*adc_volts;
    }
}

```

```

//=====
float lee_ADC(char sensor)
{
    ADCOH=0;
    ADCOM=0;
    RDY0=0;
    SF=0x200;
    ADCOCON2 = 0x40|sensor;          // Refin+/-, Ain1->Ain2 (Diferenciales)
    ADCMODE = 0x22;                 // Chop Enable (105Hz conversion rate)
    while (!RDY0)
    {
        //Espera a que termine la conversion;
    }
    a=(unsigned int)ADCOH&0xFF;
    a=a<<8;
    a|=(unsigned int)ADCOM;
    volts=a*0.0000390625;           // para Vref=2.5V (2.56V)
    return(volts);
}
//=====
void lee_termometros(void)
{
    bus=0;
    for (j=0; j<3; j++)
    {
        dig_temp = lee_tempdig(j);
        temp0[j] = dig_temp+temp0[j];
        temp0_2[j] += dig_temp*dig_temp;
    }
    bus=1;
    for (j=0; j<3; j++)
    {
        dig_temp=lee_tempdig(j);
        temp1[j]=dig_temp+temp1[j];
        temp1_2[j]+=dig_temp*dig_temp;
    }
    bus=2;
    for (j=0; j<3; j++)
    {
        dig_temp=lee_tempdig(j);
        temp2[j]=dig_temp+temp2[j];
        temp2_2[j]+=dig_temp*dig_temp;
    }
    bus=3;
    for (j=0; j<3; j++)
    {
        dig_temp=lee_tempdig(j);
        temp3[j]=dig_temp+temp3[j];
        temp3_2[j]+=dig_temp*dig_temp;
    }
}

```

```

}
//=====
float lee_tempdig(unsigned char sensor)
{
    unsigned char ack,i,x=0,k,addr;
    int d=0;
    float f;
    unsigned char dd=100000;           //Revisar variando tiempo de espera
    addr=i2c_addr|sensor;
    i2c_start();
    i=7;
    send_clock(0);
    do {
        i--;
        x=(addr>>i)&0x01;
        send_data(x);
        send_clock(1);
        delay(dd);
        send_clock(0);
    } while(i);
    send_data(1);                       // Mandar R/W BYTE
    send_clock(1);
    delay(dd);
    send_clock(0);
    send_clock(1);                       // Read ACK
    ack=get_data();
    if (ack==0)
    {
        send_clock(0);
        i2c_data=0;                       // Limpiar variable
        // LECTURA 8 BITS MSB
        send_data(1);                       // Por lo del Open Drain
        for(i=15;i>=8;i--)
        {
            send_clock(1);
            delay(dd);
            k=get_data();
            send_clock(0);
            i2c_data=((unsigned int)k<<i)|i2c_data;
            d=((unsigned int)k<<i);
            i2c_data=d|i2c_data;
        }
        // MASTER ACK
        send_data(0);
        send_clock(1);
        delay(dd);
        send_clock(0);
        // LECTURA 8 BITS LSB
        send_data(1);
        for(i=0,x=7;i<=7;i++,x--)

```

```

        {
            send_clock(1);
            delay(dd);
            k=get_data();
            send_clock(0);
            if(x==0) i2c_data=((unsigned int)k)|i2c_data;
            else i2c_data=((unsigned int)k<<x)|i2c_data;
        }
        // MASTER NACK
        send_data(1);
        send_clock(1);
        delay(dd);
        send_clock(0);
        // STOP
        i2c_stop();
        d=i2c_data>>2; // Se recorre dos a la derecha, ya que
                    // el D2 del Temperature Register es el LSB
        if(signo==1) {d=~d+1}*=-1;} //Complemento a dos
        f=(d*0.03125); //Checar esta conversion (0.03125 = LSB)
    return(f);
    }
    else
    { return(-55); }
}
//=====
void configura_termometros(void) // Se configuran a 14 bits
{
    pointer=0x04; control = 0x60;
    for (bus=0; bus<4; bus++)
    {
        for (j=0; j<3; j++) {write_control(j); }
    }
}
//=====
void return_pointer_to_cero (void)
//El registro de puntero tiene memoria, se debe regresar a 0's.
{
    pointer=0x00; control =0x00;
    for (bus=0; bus<4; bus++)
    {
        for (j=0; j<3; j++) {write_control(j); }
    }
    //Condiciones de BUS desocupado y listo
    for (bus=0; bus<=3; bus++){send_data(1); send_clock(1);}
}
//=====
int write_control(unsigned char sensor)
{
    unsigned char ack,i,x=0,addr;
    unsigned char dd=1000;

```

```

addr=i2c_addr|sensor;
delay(dd);
i2c_start();

i=7; // En la direccion se envian 7 bits
send_clock(0);

do { // Envia [Address Register]
    i--;
    x=(addr>>i)&0x01;
    send_data(x);
    send_clock(1);
    delay(dd);
    send_clock(0);
} while(i);
// Mandar R/W BYTE = 0, significa que escribira en un registro
send_data(0);
send_clock(1);
delay(dd);
send_clock(0);
    delay (dd);
    i=8; // En el pointer se envian 8 bits
send_clock(1); // Read ACK
delay (dd);
ack=get_data();
if (ack==0)
{
    send_clock(0);
    do { // Envia [Pointer Register]
        i--;
        x=(pointer>>i)&0x01;
        send_data(x);
        send_clock(1);
        delay(dd);
        send_clock(0);
    } while(i);
}
else{ return(-56); } // Fallo en transmision
i=8;
send_clock(1); // Read ACK
ack=get_data();
if (ack==0)
{
    send_clock(0);
    do { // Envia [Control/Status Register]
        i--;
        x=(control>>i)&0x01;
        send_data(x);
        send_clock(1);
    } while(i);
}
}

```

```

        delay(dd);
        send_clock(0);
    } while(i);
}
else{      return(-56); }           // Fallo en transmision
send_clock(1);
ack=get_data();                     // Lee el ultimo Acknowledge
if (ack==0)
{
    send_clock(0);
}
else{      return(-56); }           // Fallo en transmision
i2c_stop();
}
//=====
void i2c_start(void)
{
    send_data(1);
    delay(100);
    send_clock(1);
    delay(100);
    send_data(0);
    delay(100);
    send_clock(0);
}
//=====
void i2c_stop(void)
{
    send_data(0);
    delay(100);
    send_clock(1);
    delay(100);
    send_data(1);
}
//=====
void send_data(char p)
{
    switch(bus)
    {
        case 0:{P0_7=p&0x01; break;}
        case 1:{P0_5=p&0x01; break;}
        case 2:{P0_3=p&0x01; break;}
        case 3:{P0_1=p&0x01; break;}
    }
}
//=====
void send_clock(char p)
{
    switch(bus)
    {

```

```

        case 0: {P0_6=p&0x01; break;}
        case 1: {P0_4=p&0x01; break;}
        case 2: {P0_2=p&0x01; break;}
        case 3: {P0_0=p&0x01; break;}
    }
}
//=====
unsigned char get_data(void)
{
    unsigned char p;
    switch(bus)
    {
        case 0:{p=P0_7; break;}
        case 1:{p=P0_5; break;}
        case 2:{p=P0_3; break;}
        case 3:{p=P0_1; break;}
    }
    return(p&0x01);
}
//=====
void muestraa_contadores(void)
{
    if (pin3 == 0)
    {
        TRO=1; TR1=1; TR2=1;
        while(TH0 != 0x7F)
        {
            lee_pulsos();
        }
        TRO=0; TR1=0; TR2=0;
    }
    if (pin3 == 1)
    {
        lee_pulsos();
    }
}
//=====
void lee_pulsos(void)
{
    b=(long int)TH0&0xFF;
    b=b<<8;
    b|=(long int)TL0;
    c=(long int)TH1&0xFF;
    c=c<<8;
    c|=(long int)TL1;
    e=(long int)TH2&0xFF;
    e=e<<8;
    e|=(long int)TL2;
}
//=====

```

```

void manda_resultados()
{
char h;
//long int f1,f2;
//if (pin3 == 0)      { f1=(c*1572864)/b;f2=(e*1572864)/b; }
//Se moveran al prog. de la PC
for(h=0; h<8; h++)
{ sum[h]=(sum[h]/N); sum_2[h]= (sum_2[h]/N)-((sum[h])*(sum[h])); }
//printf("\n\rCuentas RefTemp T1 T2 8_Analogicos \n\r");
if(pin3 == 0) {printf("%d %d %d %d ",N,b,c,e);}
else {printf("%d %d %d %d ",N,b,c,e);}
for(h=0; h<8; h++) {printf("%1.4f ", sum[h]);}
//printf ("\n\rSensores digitales LM73\n\r");
/*
for(h=0; h<3; h++)
{
temp0[h]=(temp0[h]/N);
temp0_2[h]=sqrt((temp0_2[h]/N)-((temp0[h])*(temp0[h])));
temp1[h]=(temp1[h]/N);
temp1_2[h]=sqrt((temp1_2[h]/N)-((temp1[h])*(temp1[h])));
temp2[h]=(temp2[h]/N);
temp2_2[h]=sqrt((temp2_2[h]/N)-((temp2[h])*(temp2[h])));
temp3[h]=(temp3[h]/N);
temp3_2[h]=sqrt((temp3_2[h]/N)-((temp3[h])*(temp3[h])));
}
for(h=0; h<3; h++) {printf("%1.5f ", temp0[h]);}
for(h=0; h<3; h++) {printf("%1.5f ", temp1[h]);}
for(h=0; h<3; h++) {printf("%1.5f ", temp2[h]);}
for(h=0; h<3; h++) {printf("%1.5f ", temp3[h]);}
//printf ("\n\rCuadraticos Analogicos\n\r");
for(h=0; h<8; h++) {printf("%1.5f ", sum_2[h]);}
//printf ("\n\rCuadraticos Digitales\n\r");
for(h=0; h<3; h++) {printf("%1.5f ", temp0_2[h]);}
for(h=0; h<3; h++) {printf("%1.5f ", temp1_2[h]);}
for(h=0; h<3; h++) {printf("%1.5f ", temp2_2[h]);}
for(h=0; h<3; h++) {printf("%1.5f ", temp3_2[h]);}
*/
printf("\n\r");
}
//=====
char putchar (char c)
{
volatile unsigned int i;
while (com_putchar (c) != 0)
{
for (i=0; i<1000; i++)
{
/**** DO NOTHING ****/
}
}
}
}

```

```
return (c);
}
//=====
void delay(unsigned int pausa)
{
    int x;
    for(x=0;x<=pausa;x++);
}
}
```

## B.1.2. Programa de la Gumstix para lectura del Anemómetro Ultrasónico

```

//=====
// Programa desarrollado en: C++
// R. Michel, J. Martinez, A. Castro. Mayo 2006.
//Este programa es ejecutado en la Gumstix. Se encarga de la lectura del puerto
//ttyS2 (como se indica en el archivos irga.cfg) y almacena datos en un
//archivo especifico. El conjunto IRGA-Anemometro opera con 4Hz 9600bps. Los
//datos son almacenados en archivos que contienen 3 horas de mediciones.
//=====
#include <sys/types.h>
#include <stdio.h>
#include <sys/time.h>
#include <sys/ioctl.h>
#include <unistd.h>
#include <string.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <time.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>

#define BAUDRATE B9600
#define MaxBuf 800

struct termios oldtio, newtio;

time_t rawtime;
tm *ptm;

char Linea[MaxBuf], Nombre[MaxBuf], Nombre_ant[MaxBuf];
char COM[30], Torre[10], error[81];
char car, path[] = "/termar/";
char pathCF[] = "/mnt/CF/";
float x[10];
int u[3],Tiempo;
int i, j, k, l, puerto, readerr;
int horas, minutos,segundos, min_ini, min_fin, seg_ini, seg_fin;
int h_ant;
FILE *Archivo;
//=====
void LeeConfiguracion() // Lee la configuracion del hardware.
{
    sprintf(Nombre,"%sprogramas/irga.cfg",path);
    Archivo=fopen(Nombre,"r");
    if(Archivo == NULL)
    {
        printf("%s\n", Nombre);
        perror(error);
    }
}

```

```

        exit(-1);
    }
    fscanf(Archivo, "%s %s \n", Torre, Linea);
    printf("%s %s\n", Torre, Linea);
    fscanf(Archivo, "%s %s \n", COM, Linea);
    printf("%s %s\n", COM, Linea);
    fclose(Archivo);
    printf("--- %s leído.\n\n", Nombre);
}

//=====
void AbrePuerto()                // Inicializa el puerto serie.
{
// r/w sin esperar a que llegue algo
    puerto = open(COM, O_RDWR | O_NOCTTY);
    if (puerto<0) { perror(COM); exit (-1); }
// Guardar condiciones de puerto actuales.
    tcgetattr (puerto, &oldtio);
    bzero (&newtio, sizeof(newtio));
// Baudrate=9600, CS8=8n1(8 bits, sin paridad y 1 stop bit)
// CLOCAL=conexionn local, sin control de modem
// CREAD=habilitar caract. recepcion.
    newtio.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD;
// IGNPAR= Ignorar bytes con error de paridad.
    newtio.c_iflag = IGNPAR;
// Salida sin formato.
    newtio.c_oflag = 0;
    newtio.c_lflag = 0;
    newtio.c_cc[VTIME] = 0x00;
    newtio.c_cc[VMIN] = 0x00;
    tcflush (puerto, TCIFLUSH);
    tcsetattr (puerto, TCSANOW, &newtio);
    printf("--- Puerto %s abierto. \n\n", COM);
}
//=====
int SoloDigitos(char h)
{
    if(h=='0' || h=='1' || h=='2' || h=='3' || h=='4' || h=='5' || h=='6' ||
        h=='7' || h=='8' || h=='9' || h=='.' || h=='-' || h==' ' || h=='\n' || '
        ') return (1);
    else return (0);
}
//=====
void LeeSensor()
{
    int rflag, resp;
    time(&rawtime);
    ptm = localtime(&rawtime);
    seg_ini = ptm->tm_sec;
    j=0;
    do

```

```

    {
        rflag = read(puerto, &car, 1);
        resp = SoloDigitos(car);
        if(rflag>0 && resp>0) Linea[j++] = car;
        time(&rawtime);
        ptm = localtime(&rawtime);
        seg_fin = ptm->tm_sec;
    }
    while (car != '\r');
    car=0;
    rflag=0;
}
//=====
void Convierte_a_Numeros()          // Extrae valores en el buffer.
{
    int DatosConvertidos;
    DatosConvertidos = sscanf (Linea,
        "%d %d \
        %f %f %f %f %f %f %f %f ",
        &u[ 0],&u[ 1],
        &x[ 0],&x[ 1],&x[ 2],&x[ 3],&x[ 4],&x[ 5],&x[ 6],&x[ 7]);
        %f %f %f %f %f %f %f %f \n\r",
        u[ 0],u[ 1],
        x[ 0],x[ 1],x[ 2],x[ 3],x[ 4],x[ 5],x[ 6],x[ 7]);*/
}
//=====
void ArchivaDatos()
{
    LeeSensor();
    time (&rawtime);
    ptm = localtime(&rawtime);
    h_ant = ptm->tm_hour;          //hora de inicio
    sprintf(Nombre, "%sirga/%d%02d%02d%02d_%.dat",
        pathCF,
        ptm->tm_year+1900,
        ptm->tm_mon+1,
        ptm->tm_mday,
        h_ant,
        Torre);
    Archivo=fopen(Nombre, "a+");
    if (Archivo == NULL)
        { perror(error); usleep(500000); Archivo=fopen(Nombre, "a+");}
    if (Archivo == NULL)
        { perror(error); usleep(500000); Archivo=fopen(Nombre, "a+");}
    if (Archivo != NULL)
    {
        do
        {
            LeeSensor();
            Convierte_a_Numeros();

```

```

horas = ptm->tm_hour;\section{Programa de la Gumstix}[!ht]
minutos = ptm->tm_min;
segundos = ptm->tm_sec;
if (((horas%3==0)&&(segundos==0))|((horas==0)&&(segundos==0)))
{
    fclose (Archivo);
    fflush (Archivo);
    fflush (NULL);
    h_ant=horas;
    sprintf(Nombre, "%sirga/%d%02d%02d%02d_%.s.dat",
        pathCF,
        ptm->tm_year+1900,
        ptm->tm_mon+1,
        ptm->tm_mday,
        h_ant,
        Torre);
    Nombre_ant=Nombre;
    Archivo=fopen(Nombre, "a+");
    if (Archivo == NULL)
        { perror(error); usleep(500000); Archivo=fopen(Nombre, "a+");}
    if (Archivo == NULL)
        { perror(error); usleep(500000); Archivo=fopen(Nombre, "a+");}
}
if (Archivo != NULL)
{
    fprintf(Archivo, "%d %d %d ",horas, minutos, segundos);
    fprintf(Archivo, "%d %d %1.2f %1.2f %1.2f %1.2f %1.2f %1.2f %1.2f %1.2f \n",
        u[ 0],u[ 1],x[ 0],x[ 1],x[ 2],x[ 3],x[ 4],x[ 5],x[ 6],x[ 7]);
}
}while (1);
}
}
//=====
int main (void)
{
    printf("\n\r Lectura de IRGA(LI-7500)-Anemometro(81000V)");
    printf("\n\r No ejecute el programa mas de 1 vez");
    printf("\n\r de lo contrario elimine la tarea utilizando");
    printf("\n\r [ kill -9 <tarea> ]. Ver indice con [ ps -aux ]");
    LeeConfiguracion();
    usleep(100000);
    AbrePuerto();
    usleep(100000);
    printf("\n\r Corriendo ... \n\r");
    LeeSensor(); //Limpiar valores en buffer (si existen).
    ArchivaDatos();
    return 0;
}

```

### B.1.3. Programa para la lectura de la red de módulos de adquisición

```
//=====
// Programa de recoleccion de datos de las Torres Eddy Flux.
// R. Michel, J. Martinez, A. Castro. Dic. 2006
//=====
#include <sys/types.h>
#include <stdio.h>
#include <sys/time.h>
#include <sys/ioctl.h>
#include <unistd.h>
#include <string.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <time.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>
#define BAUDRATE B9600
#define MaxBuf 800
struct termios oldtio, newtio;
time_t rawtime;
tm * ptm;
char Linea[MaxBuf], Nombre[MaxBuf];
char COM[30], Torre[10], error[81];
char path2[] = "/mnt/CF/mod/";
float y[48], x[48], a[8][4][8];
int i, j, k, puerto, readerr, Mediciones;
int horas, minutos, min_ini, min_fin, seg_ini, seg_fin;
int NumModulos, Modulo[8], Cuentas[3];
FILE *Archivo;
//=====
void LeeConfiguracion() // Lee la configuracion del hardware
{
    sprintf(Nombre, "%sprogramas/torres.cfg", path);
    Archivo = fopen(Nombre, "r");
    if (Archivo == NULL)
    {
        printf("%s\n", Nombre);
        perror(error);
        exit(-1);
    }
    fscanf(Archivo, "%s %s\n", Torre, Linea);
    printf("%s %s\n", Torre, Linea);
    fscanf(Archivo, "%s %s\n", COM, Linea);
    printf("%s %s\n", COM, Linea);
    fscanf(Archivo, "%i %s\n", &NumModulos, Linea);
    printf("%d %s\n", NumModulos, Linea);
    for (i = 0; i < NumModulos; i++)
    {
        fscanf(Archivo, "%i %s\n", &Modulo[i], Linea);
        printf("%d %s\n", Modulo[i], Linea);
    }
    fclose(Archivo);
}
```

```

    printf("--- %s leído.\n\n", Nombre);
}
//=====
void LeeCalibraciones()                // Lee archivos calibracion
{
    for(i=0;i<NumModulos;i++)
    {
        sprintf(Nombre, "%sprogramas/Calib%d.cfg", path, Modulo[i]);
        Archivo=fopen(Nombre,"r");
        if(Archivo == NULL)
        {
            printf("%s\n", Nombre);
            perror(error);
            exit(-1);
        }
        for(j=0; j<8; j++)
        {
            fscanf(Archivo, "%f %f %f %f %s \n", &a[j][0][i],
                    &a[j][1][i], &a[j][2][i], &a[j][3][i], Linea);
            printf("%1.3f %1.3f %1.3f %1.3f %s \n", a[j][0][i], a[j][1][i],
                    a[j][2][i], a[j][3][i], Linea);
        }
        fclose (Archivo);
        printf("--- %s leído.\n\n", Nombre);
    }
}
//=====
void AbrePuerto()                    // Inicializa el puerto serie
{
    // r/w sin esperar a que llegue algo
    puerto = open(COM, O_RDWR | O_NOCTTY);
    if (puerto<0) { perror(COM); exit (-1); }
    // Guardar condiciones de puerto actuales
    tcgetattr (puerto, &oldtio);
    bzero (&newtio, sizeof(newtio));
    // Baudrate=9600, CS8=8n1(8 bits, sin paridad y 1 stop bit)
    // CLOCAL=conexionn local, sin control de modem
    // CREAD=habilitar caract. recepcion.
    newtio.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD;
    // IGNPAR= Ignorar bytes con error de paridad.
    newtio.c_iflag = IGNPAR;
    // Salida sin formato.
    newtio.c_oflag = 0;
    newtio.c_lflag = 0;
    newtio.c_cc[VTIME] = 0x00;
    newtio.c_cc[VMIN] = 0x00;
    tcflush (puerto, TCIFLUSH);
    tcsetattr (puerto, TCSANOW, &newtio);
    printf("--- Puerto %s abierto. \n\n", COM);
}
//=====

```

```

int SoloDigitos(char h)
{
    if(h=='0' || h=='1' || h=='2' || h=='3' || h=='4' || h=='5' || h=='6' ||
        h=='7' || h=='8' || h=='9' || h=='.' || h=='-' || h==' ' || h=='\n') return (1);
    else return (0);
}
//=====
void LeeModulo()
{
    int rflag, resp;
    sprintf(Linea, ":2M%dV;", Modulo[k]);
    write(puerto, &Linea, 6);
    time(&rawtime);
    ptm = localtime(&rawtime);
    seg_ini = ptm->tm_sec;
    j=0;
    do
    {
        rflag = read(puerto, &car, 1);
        resp = SoloDigitos(car);
        if(rflag>0 && resp>0) Linea[j++] = car;
        time(&rawtime);
        ptm = localtime(&rawtime);
        seg_fin = ptm->tm_sec;
    } while (car != '\r' && j < (MaxBuf-2) && (seg_fin-seg_ini) < 1);
}
//=====
void EsperaUnMinuto()
{
    time (&rawtime);
    ptm = localtime(&rawtime);
    min_ini = ptm->tm_min;
    sleep(50);
    do
    {
        usleep(100000);
        time (&rawtime);
        ptm = localtime(&rawtime);
        min_fin = ptm->tm_min;
    } while (min_fin == min_ini);
}
//=====
void Convierte_a_Numeros() // Extrae valores en el buffer
{
    int DatosConvertidos;
    DatosConvertidos = sscanf (Linea,
        "%d %d %d %d\
        %f %f %f %f %f %f %f %f ",
        &Mediciones,&Cuentas[ 0],&Cuentas[ 1],&Cuentas[ 2],
        &x[ 0],&x[ 1],&x[ 2],&x[ 3],&x[ 4],&x[ 5],&x[ 6],&x[ 7]);
}

```

```

}
//*****
void Calibra() // Convierte a unidades reales.
{
    for(j=0;j<8;j++) y[j] = a[j][0][k] + a[j][1][k]*x[j] +
        a[j][2][k]*x[j]*x[j] + a[j][3][k]*x[j]*x[j]*x[j];
}
//=====
void ArchivaDatos()
{
    sprintf(Nombre, "%s/%d%02d%02d%d_%.s.dat",
        path2,
        ptm->tm_year+1900,
        ptm->tm_mon+1,
        ptm->tm_mday,
        Modulo[k],
        Torre);
    Archivo=fopen(Nombre, "a");
    // Si NO logra abrirlo, tratar varias veces.
    if (Archivo == NULL) { perror(error); usleep(500000); Archivo=fopen(Nombre, "a");}
    if (Archivo == NULL) { perror(error); usleep(500000); Archivo=fopen(Nombre, "a");}
    // Si logro abrirlo, escribir datos, si NO continuar.
    if (Archivo != NULL)
    {
        //Hora, mediciones y cuentas
        fprintf(Archivo, "%02d %02d %d %d %d %d ", horas, minutos, Mediciones,
            Cuentas[ 0], Cuentas[ 1], Cuentas[ 2]);
        //Sensores Analogicos
        fprintf(Archivo, "%1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f\n",
            y[ 0], y[ 1], y[ 2], y[ 3], y[ 4], y[ 5], y[ 6], y[ 7]);
        fflush(Archivo);
        fclose(Archivo);
    }
}
//=====
int main (void)
{
    printf("\n Proyecto TERMAR Dic. 2006\n");
    LeeConfiguracion(); LeeCalibraciones(); AbrePuerto();
    printf("\n Corriendo ... \n\n");
    for (k=0; k<NumModulos; k++) {LeeModulo(); sleep(1);} // Resetea modulos.
    do // Esperar el primer minuto.
    {
        time (&rawtime);
        ptm = localtime(&rawtime);
        seg_ini = ptm->tm_sec;
    } while (seg_ini != 0);
    do // Ciclo infinito de adquisicion de datos.
    {
        EsperaUnMinuto();

```

```
    horas = ptm->tm_hour;
    minutos = ptm->tm_min;
    for (k=0; k<NumModulos; k++)
    {
        LeeModulo();
        Convierte_a_Numeros();
        Calibra();
        ArchivaDatos();
    }
} while(1);
return 0;
}
```

## B.2. Programas para despliegue y procesamiento de datos

### B.2.1. Programa para el calculo de medias estadísticas

```

=====
! Programa desarrollado en: Fortran 77
! A. Castro, R. Michel. Febrero 2007.
! Este programa calcula las medias de 30 min para archivos de datos generados
! a partir de cadenas de salida del sistema IRGA-Anemometro.
=====
      INTEGER i, j, k, mm
      REAL*8 a(200), suma(200)

      OPEN(7,FILE='original.dat',STATUS= 'OLD')
      OPEN(8,FILE='medias.dat',STATUS= 'UNKNOWN')

      READ(7,*,END=20,ERR=10) (a(i),i=1,14)
      k1 = INT(a(3))/30
!      write(*,*) k1
      DO i=5,14
        suma(i) = a(i)
      ENDDO
      j = 1
10  READ(7,*,END=20,ERR=10) (a(i),i=1,14)
      k2 = INT(a(3))/30
!      write(*,*) k2
      IF(k2 .EQ. k1) THEN
        DO i=5,14
          suma(i) = suma(i) + a(i)
        ENDDO
        j = j +1
      ELSE
        mm = 15
        IF(k1 .EQ. 0) mm = 45
        WRITE(8,'(f9.3,10f9.2)')
&      a(1)+a(2)/24.+mm/1440.,(suma(i)/(j-1),i=5,14)
        k1 = INT(a(3))/30
        DO i=5,14
          suma(i) = a(i)
        ENDDO
        j = 1
      ENDIF
      goto 10

20  close(7)
      close(8)

      END

```

## B.2.2. Programa para graficación de datos

```

=====
! Programa desarrollado en: Fortran 77
! R. Michel, A. Castro. Agosto 2006.
! Este programa grafica archivos de datos del sistema IRGA-Anemometro (incluye
! un vector de horario decimal). Puede ser modificado para graficar los datos
! obtenidos en base las cadenas de datos de los modulos de adq. de datos.
=====
! Program to plot the 1.5m's meteorological station data
! Raul Michel, A.Castro
! Aug. 2006
=====
PARAMETER (m = 500000)

REAL      hora(m), h2o(m), co2(m)
REAL      u(m), v(m), w(m)
REAL      rapidez(m), azimuth(m), elevacion(m)
REAL      vsonido(m), tempsonica(m)
REAL      TMin, TMax, yMin, yMax
INTEGER   i, j, Total

CHARACTER yy*6,mm*6,dd*6,hhh*6,mmm*6,Fecha*20
CHARACTER FileName*20
LOGICAL   ex0, ex1, fin_dia

WRITE(*,*) 'File to plot? '
READ(*,*) FileName

i = 1
OPEN(7, FILE=FileName, STATUS = 'old')
22 READ(7,*,end=33,err=22)
& hora(i), h2o(i), co2(i), u(i), v(i), w(i),
& rapidez(i), azimuth(i), elevacion(i),
& vsonido(i), tempsonica(i)
WRITE(*,'(11f8.2)')
& hora(i), h2o(i), co2(i), u(i), v(i), w(i),
& rapidez(i), azimuth(i), elevacion(i),
& vsonido(i), tempsonica(i)
i = i + 1

GOTO 22
33 CLOSE(7)
Total = i - 1

CALL PGBEGIN(0,'irgaim.ps/vcps',1,1)
CALL PGSCH(0.5)
CALL PGSCI(1)
CALL PGSLW(2)

TMin = 10e6

```

```
TMax = -10e6
DO i=1,Total
  TMin = MIN(TMin,hora(i))
  TMax = MAX(TMax,hora(i))
ENDDO

CALL PGVSIZ(0.5, 7.4, 1.0, 10.0)
CALL PGSWIN(TMin,TMax,0.0, 1.0)

! First part. H2O

yMin = 10e6
yMax = -10e6
DO i=1,Total
  yMin = MIN(yMin,h2o(i))
  yMax = MAX(yMax,h2o(i))
ENDDO

delta = (yMax-yMin)/8.0
CALL PGVSIZ(0.5, 7.4, 8.6, 10.0)
write(*,*) TMin,TMax,yMin-delta,yMax+delta

CALL PGSWIN(TMin,TMax,yMin-delta,yMax+delta)
CALL PGBOX('BCTS',1.0,5,'BCNTS',0.0,0)
CALL PGLABEL(' ', 'H2O(mmol m-3)',Filename)

CALL PGSCI(4)           ! blue, Text
CALL PGPT(Total,hora,h2o,20)

CALL PGSCI(1)

! Second part. CO2

yMin = 10e6
yMax = -10e6
DO i=1,Total
  yMin = MIN(yMin,co2(i))
  yMax = MAX(yMax,co2(i))
ENDDO

delta = (yMax-yMin)/8.0
CALL PGVSIZ(0.5, 7.4, 7.2, 8.6)

CALL PGSWIN(TMin,TMax,yMin-delta,yMax+delta)
CALL PGBOX('BCTS',1.0,5,'BCNTS',0.0,0)
CALL PGSCI(4)           ! blue, Text
CALL PGPT(Total,hora,co2,20)

CALL PGSCI(1)
CALL PGLABEL(' ', 'CO2(mg m-3)', '')
```

```

! Third part. Rapidez

yMin = 10e6
yMax = -10e6
DO i=1,Total
  yMin = MIN(yMin,rapidez(i))
  yMax = MAX(yMax,rapidez(i))
ENDDO

delta = (yMax-yMin)/8.0
CALL PGVSIZ(0.5, 7.4, 5.8, 7.2)

CALL PGSWIN(TMin,TMax,yMin-delta,yMax+delta)
CALL PGBOX('BCTS',1.0,5,'BCNTS',0.0,0)
CALL PGSCI(4) ! blue, Text
CALL PGPT(Total,hora,rapidez,20)

CALL PGSCI(1)
CALL PGLABEL('', 'Rapidez (m/s) ', '')

! 4th. azimuth

yMin = 10e6
yMax = -10e6
DO i=1,Total
  yMin = MIN(yMin,azimuth(i))
  yMax = MAX(yMax,azimuth(i))
ENDDO

delta = (yMax-yMin)/8.0
CALL PGVSIZ(0.5, 7.4, 4.4, 5.8)

CALL PGSWIN(TMin,TMax,yMin-delta,yMax+delta)
CALL PGBOX('BCTS',1.0,5,'BCNTS',0.0,0)
CALL PGSCI(4) ! blue, Text
CALL PGPT(Total,hora,azimuth,20)

CALL PGSCI(1)
CALL PGLABEL('', 'Azimuth (grados)', '')

! 5th part. elevation

yMin = 10e6
yMax = -10e6
DO i=1,Total
  yMin = MIN(yMin,elevacion(i))
  yMax = MAX(yMax,elevacion(i))
ENDDO

```

```
delta = (yMax-yMin)/8.0
CALL PGVSIZ(0.5, 7.4, 3., 4.4)

CALL PGSWIN(TMin,TMax,yMin-delta,yMax+delta)
CALL PGBOX('BCTS',1.0,5,'BCNTS',0.0,0)
CALL PGSCI(4) ! blue, Text
CALL PGPT(Total,hora,elevacion,20)

CALL PGSCI(1)
CALL PGLABEL('', 'Elevacion (grad) ', '')

! 6th. Velocidad del sonido

yMin = 10e6
yMax = -10e6
DO i=1,Total
  yMin = MIN(yMin,vsonido(i))
  yMax = MAX(yMax,vsonido(i))
ENDDO

delta = (yMax-yMin)/8.0
CALL PGVSIZ(0.5, 7.4, 1.6, 3.0)

CALL PGSWIN(TMin,TMax,yMin-delta,yMax+delta)
CALL PGBOX('BCTS',1.0,5,'BCNTS',0.0,0)
CALL PGSCI(4) ! blue, Text
CALL PGPT(Total,hora,vsonido,20)

CALL PGSCI(1)
CALL PGLABEL('', 'Vel Sonido (m/s)', '')

! 7th. temperaura sonica

yMin = 10e6
yMax = -10e6
DO i=1,Total
  yMin = MIN(yMin,tempsonica(i))
  yMax = MAX(yMax,tempsonica(i))
ENDDO

delta = (yMax-yMin)/8.0
CALL PGVSIZ(0.5, 7.4, 0.2, 1.6)

CALL PGSWIN(TMin,TMax,yMin-delta,yMax+delta)
CALL PGBOX('BCTS',1.0,5,'BCNTS',0.0,0)
CALL PGSCI(4) ! blue, Text
CALL PGPT(Total,hora,tempsonica,20)

CALL PGSCI(1)
CALL PGLABEL('', 'Temp Sonica(C)', '')
```

999 END

C -----

SUBROUTINE INUMB2TXT(Entera, a)

INTEGER Entera  
CHARACTER\*6 a

INTEGER i  
i = Entera

IF(i .LE. 999999) THEN  
  a(1:1) = CHAR(i/100000+48)  
  i = i - (i/100000)\*100000  
  a(2:2) = CHAR(i/10000+48)  
  i = i - (i/10000)\*10000  
  a(3:3) = CHAR(i/1000+48)  
  i = i - (i/1000)\*1000  
  a(4:4) = CHAR(i/100+48)  
  i = i - (i/100)\*100  
  a(5:5) = CHAR(i/10+48)  
  i = i - (i/10)\*10  
  a(6:6) = CHAR(i+48)

ELSE  
  a = '#####'

ENDIF  
RETURN  
END

! -----

INTEGER FUNCTION LEN1( String )

! Calcular nmero de caracteres de una cadena

CHARACTER\*(\*) String

DO i = LEN( String ), 1, -1

  IF( String(i:i) .NE. ' ' ) THEN

    LEN1 = i

    RETURN

  ENDIF

ENDDO

LEN1 = 0

RETURN

END

## Apéndice C

# Configuración de la microcomputadora Gumstix©

### C.1. Instalación de archivos en la estación de trabajo (PC o Laptop)

EL *buildroot* es el sistema empleado para crear las imágenes de los archivos de sistema empleados en la Gumstix. Con el *buildroot* es posible construir un sistema de archivos personalizado para la Gumstix. El *buildroot* está diseñado para operar sistemas Linux. Trabaja en sistemas basados en Linux 2.6.x. Para sistemas 2.4.x deberá consultar el procedimiento sugerido por el fabricante.

El *Kernel* de la Gumstix debe ser compilado externamente en una computadora personal y posteriormente descargado vía serial a la microcomputadora. Los programas realizados en el lenguaje C deberán ser compilados, mediante un *compilador cruzado*, también la estación de trabajo y descargados ya sea vía serial o USBnet. Para realizar estas procesos deberá instalarse la *subversion-1.2.3* y el *buildroot* de la Gumstix en la máquina.

#### C.1.1. Instalación de la Subversión y Buildroot

Para compilar los programas que serán ejecutados en la Gumstix, se requiere que la estación (PC o Laptop) presente las siguiente:

- Un cliente *Subversion* para adquirir, y en su caso actualizar, los códigos fuente provistos por la compañía Gumstix. Una *subversion* es un software que permitirá el acceso a las revisiones provistas por Gumstix.
- Un compilador de C y las herramientas relacionadas a este compilador.
- Conexión a Internet libre del **FireWall**.

### Instalación de la Subversion 1.2.3

Deberán seguirse los siguientes pasos:

1. Bajar el archivo `subversion-1.2.3.tar.gz` de la Internet de Gumstix o empleando el vínculo `http://subversion.tigris.org` y guardarlo en el directorio de **usuario** (`/home/user/`). En el prompt esto aparece indicado como `user@linux:~>`, aunque depende de la versión de linux empleada.
2. En el modo **usuario** se debe descomprimir el archivo `.tar.gz`

```
user@linux:~>tar -zxvf subversion-1.2.3.tar.gz
```

3. Entrar al recién creado directorio `.../subversion-1.2.3` y actualizar la configuración del sistema.

```
user@linux:~>cd subversion-1.2.3
user@linux:~>./configure
```

4. Ahora cambie al modo **root** y teclee

```
linux:/home/user/#make
linux:/home/user/#make install
```

### Instalación del Buildroot

Para instalar el buildroot deberán seguirse los siguientes pasos:

1. Accese una consola empleando el modo **usuario** e ingrese la línea que se muestra a continuación

```
>svn co http://svn.gumstix.com/gumstix-buildroot/trunk gumstix-buildroot
```

Esto permitirá bajar la versión más reciente del buildroot. Sin embargo, si desea alguna versión en específico deberá teclear:

```
>svn co -rXXX http://svn.gumstix.com/gumstix-buildroot/trunk gumstix-buildroot
```

Donde XXX corresponde al código de la versión requerida. Para más información consulte la página de Gumstix.

2. Entrar al recién creado directorio `.../gumstix-buildroot`

```
user@linux:~>cd gumstix-buildroot
```

3. Cambie al modo **root** y reconfigure el sistema

```
linux:/home/user/#make
```

Esto creará el sistema de archivos del buildroot. Es un proceso automático que toma un largo tiempo, sea paciente. Creará una enorme cantidad de archivos y puede señalar algunos errores, pero mientras no se detenga el proceso pueden ser ignorados.

Una vez terminado el proceso contará con el nuevo sistema de archivos de la Gumstix: `rootfs.arm_nofpu.jffs2`. Verifique la existencia de éste archivo en su directorio para poder continuar. El siguiente paso es fijar la imagen del sistema en la microcomputadora, como se indica en C.5.

### C.1.2. Compilador cruzado

Un **compilador cruzado** es un compilador capaz de crear códigos ejecutables en una plataforma diferente a la cual donde el compilador cruzado corre. Cuando el acceso a la plataforma es limitado, o en el caso de los sistemas empotrados y algunos microcontroladores, donde debido a su limitación en memoria interna y espacio físico para almacenamiento de datos, es imposible o muy complicado compilar en ambientes como estos.

GCC es un conjunto de compiladores desarrollados por el proyecto GNU (Linux GNU). Es un software de distribución libre considerado como estándar en los sistemas Linux. La distribución actual de GCC contiene *front-ends* (en software, el front-end es la parte del software que interactúa con el usuario, mientras que el *back-end* es la parte que procesa los datos de entrada desde el front-end) para los lenguajes de programación C (GCC), Fortran(GFortran), C++(G++), Ada(GNAT) y Java(GCJ). Otros como Pascal, Modula-2, Modula-3, Mercury, VHDL y PL/I también son soportados, pero no se incluyen en la distribución principal.

Una vez que la subversion-1.2.3 y el buildroot han sido instalados, la máquina cuenta con lo necesario para compilar programas (escritos en el lenguaje de programación C) y ser instalados en la microcomputadora empleando el compilador cruzado GCC. Dicho programa se encuentra en la estación de trabajo en la dirección `/home/user/gumstix-buildroot/build_arm/st`.

Para compilar un archivo en C (`archivo_fuente.cpp`) debe emplearse el siguiente formato:

```
user@linux~>arm-linux-gcc -o <archivo_executable> <archivo_fuente.cpp>
```

El nombre del archivo destino puede o no poseer una extensión. Por fines visuales en este trabajo ha sido empleada la extensión `.exe` para indicar de una manera clara aquellos archivos que sean ejecutables.

## C.2. Acceso a la Gumstix vía puerto serial

Se puede acceder al *u-boot* de la Gumstix utilizando un cable serial estándar y el programa *Kermit/cKermit* (que deberá estar previamente instalado en la Laptop o PC, para los casos donde así se requiera), o cualquier otro programa para transferencia vía serial de archivos que soporte el protocolo estilo Kermit. Los usuarios en sistemas Linux (*users*)

se encuentran organizados bajo un dominio o estación de trabajo (*estación*), manejada por un administrador, conocido también como el *superusuario*. En la estación de trabajo (PC o Laptop) es el superusuario quien puede acceder al puerto a través del programa C-Kermit o MiniCOM. Los procesos que a continuación se muestran corresponden a instrucciones ejecutables por el programa C-Kermit.

NOTA: Usando una máquina con entrada serial (típicamente computadoras de escritorio) la ruta de acceso al puerto serial es `/dev/ttyS0`, en el caso presente debido a la utilización de un conversor USB-Serial es `/dev/ttyUSB0`.

Se deberán seguir los pasos siguientes:

1. La Gumstix debe estar apagada.
2. Conecte el conversor USB-Serial a la Laptop (usando el puerto `ttyUSB0`)
3. Conecte el cable serial (null mode, DB9-MiniDIN8) al conversor y Gumstix respectivamente. En la Gumstix el puerto que se deberá utilizar es el `ttyS0` (segundo puerto de la esquina superior izquierda, como se muestra en la Figura C.1).

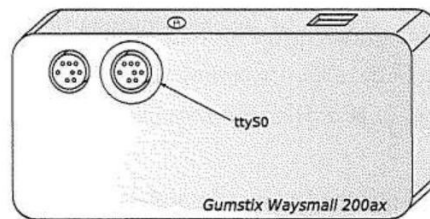


Figura C.1: Ubicación del puerto `ttyS0`.

4. Accese al directorio de usuario

```
$cd /home/user/
```

5. En una consola escriba el comando

```
$dmesg
```

Deberá aparecer un listado de operaciones, entre las cuales se debe ver:

```
usb 2-2: PL-2303 converter now attached to ttyUSB0
```

6. Y aprovechando que la Laptop ya tiene cargados los scripts, entre en modo superusuario y teclee:

```
$/Cargar_Adaptador_USB_Serial
```

7. Una vez detectado y cargado el adaptador USB-Serial ejecute el programa Kermit, teclee

```
$cd /home/user/download/kermit/  
$kermit
```

La línea del prompt aparecerá como (/home/user/download/kermit/) C-Kermit>

8. Ahora se debe configurar la consola de comunicación serial tecleando los siguientes comandos:

```
$set line /dev/tts/USB0  
$set speed 115200  
$set file type bin  
$set reliable  
$fast  
$set flow-control none  
$set carrier-watch off  
$set prefixing all  
$connect
```

9. Conecte o reinicie la Gumstix, según se requiera.

Deberá esperar algunos segundos, hasta un minuto en algunos casos. Espere. Al conectarse al dispositivo Gumstix aparece el siguiente mensaje del Kermit:

```
Escape character: Ctrl- (ASCII 28, FS): enabled  
Type the escape character followed by C to get back,  
or followed by ? to see other options.
```

```
-----  
Welcome to the Gumstix Linux Distribution!  
gumstix login:
```

Las dos últimas líneas no aparecerán si han sido anteriormente deshabilitadas por el usuario. En caso de ser así no tendrá acceso al sistema y deberá reactivar esta prestación. Una vez que la microcomputadora inicia manda estas líneas que podrían ser interpretadas por el dispositivo periférico conectado al puerto de consola como algún tipo de señal y desencadenar una respuesta. Si esto ocurre el programa interno recibiría información inapropiada y puede provocar problemas de lectura. Se recomienda operar el sistema en modo de consola silenciosa (`silent mode true`)

una vez que el dispositivo está programado.

Si el acceso a la Gumstix ha sido satisfactoriamente realizado, accederá a la consola del microcomputador y en la terminal de trabajo podrá observarse:

```

Welcome to Gumstix!
By default, this gumstix is configured for MMC support.
Unfortunately, this means Compact Flash support has to be
disabled out of the box. To turn off MMC and turn on CF,
edit the file /etc/modules and comment out or delete the MMC
lines; then rename /etc/init.d/s30pcmcia to /etc/init.d/S30pcmcia
and reboot.

```

```
#
```

Para salir de la Gumstix solo teclee:

```
#exit
```

Para salir cerrar la comunicación serial dentro del programa Kermit teclee: CTRL+\ y luego la letra c.

Para salir finalmente del programa Kermit, teclee nuevamente

```
$exit
```

### C.3. Configuración de Interfaz USBnet

Para este tipo de conexión se utiliza un cable tipo USB-miniUSB que viene usualmente integrado en el paquete de desarrollo de la Gumstix. Usando el comando `ifconfig` es posible asignar una dirección de IP a la estación de trabajo y a la Gumstix a través de dos formas de configuración: temporal y definitiva. Se analizará primero la temporal. Los números de identificación han sido asignados de acuerdo a necesidades propias de proyecto presente, sin embargo, pueden ser sustituidos conforme convengan al usuario.

#### C.3.1. Asignación temporal de IP para comunicación Gumstix- PC vía USBnet

Deberá acceder a la Gumstix por medio del puerto de consola como se instruye en la sección de *acceso vía serial* en modo superusuario. Una vez que esté dentro el sistema, ejecute la instrucción siguiente:

```
# ifconfig usb0 192.168.0.10x netmask 255.255.255.0 up
```

donde x corresponde al número de identificación de las microcomputadoras Gumstix empleadas.

En la estación (PC o Laptop) ingrese a través de una terminal de trabajo en modo superusuario y teclee la siguiente instrucción:

```
# ifconfig usb0 192.168.0.1 netmask 255.255.255.0 up
```

Con el comando mostrado arriba la interfaz USBnet usb0 de la estación queda configurada.

La instrucción `ifconfig -a` permite mostrar la configuración que guardan todas las interfaces en cada máquina. En la estación por ejemplo:

```
[root@workstation ~]# ifconfig -a
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:2379 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2379 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3808422 (3.6 MiB) TX bytes:3808422 (3.6 MiB)
usb0     Link encap:Ethernet HWaddr 46:59:DE:1D:71:EC
          inet addr:192.168.0.1 Bcast:10.0.0.255 Mask:255.255.255.0
          inet6 addr: fe80::4459:deff:fe1d:71ec/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:380 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:218880 (213.7 KiB) TX bytes:432 (432.0 b)
```

### C.3.2. Asignación definitiva de IP para comunicación Gumstix- PC vía USBnet

La Gumstix puede establecer automáticamente la configuración necesaria para manejar la USBnet, mediante la edición del archivo `/etc/network/interfaces`. El editor VI es sugerible estudiarlo previante a realizar estos cambios, se sugiere también realizar una copia de seguridad del archivo antes mencionado antes de inicar cambios. La copia deberá realizarse de la siguiente forma, en terminal de la Gumstix:

```
#cd /etc/network
#cp interfaces interfaces.bak
```

Deberá accesarse este mediante el editor VI. Permaneciendo en la posición del directorio ubicada por la última instrucción. Teclee:

```
#vi interfaces
```

Comentar la línea `iface usb0 inet dhcp` debajo añadir la siguiente sección de código.

```

iface usb0 inet static
address 192.168.0.10x
netmask 255.255.255.0
pre-up /sbin/modprobe g_ether; /sbin/modprobe af_packet
post-down /sbin/modprobe -r g_ether

```

Donde x es el dígito identificador del dispositivo utilizado. Estos cambios serán ejecutados la próxima ocasión que la microcomputadora sea reiniciada.

En la estación ingrese a través de una terminal de trabajo en modo superusuario y teclee la siguiente instrucción:

```
# ifconfig usb0 192.168.0.1 netmask 255.255.255.0 up
```

NOTA: Linux requiere de identificación de los dispositivos *host* con los cuales se comunica, generando un código de identificación intransferible para cada IP de estos dispositivos. Cada ocasión en que se cambie el IP de uno de los dispositivos deberá editarse el archivo `/root/.ssh/known_hosts` de la estación (PC o Laptop), eliminando la línea que contenga la dirección que estamos actualizando. Esto solo es requerido en los casos donde el dispositivo *host* haya establecido previamente comunicación con la estación.

La estación Linux puede no reconocer a `usb0` como una interfaz sino hasta que la Gumstix haya iniciada (recuerde que le toma cierto tiempo, aproximadamente de un minuto o poco menos). Cuando se intenta configurar el `usb0` mientras la microcomputadora se encuentra aún cargando el kernel parece el mensaje

```

[root@workstation ~]# ifconfig usb0 192.168.0.1 netmask 255.255.255.0 up
SIOCSIFADDR: No such device
usb0: unknown interface: No such device
SIOCSIFNETMASK: No such device
SIOCGIFADDR: No such device
SIOCSIFBROADCAST: No such device
usb0: unknown interface: No such device

```

La configuración puede ser probada mediante la instrucción `ping`. Desde la Gumstix por ejemplo:

```

# ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=64 time=8.2 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=5.0 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=4.6 ms
--- 192.168.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 4.6/5.9/8.2 ms

```

Desde la estación puede probarse también el enlace USBnet (en el ejemplo con el dispositivo Gumstix con IP 192.168.0.101

```
[root@workstation ~]# ping 192.168.0.101
PING 192.168.0.101 (192.168.0.101) 56(84) bytes of data.
64 bytes from 192.168.0.101: icmp_seq=0 ttl=64 time=2.03 ms
64 bytes from 192.168.0.101: icmp_seq=1 ttl=64 time=1.32 ms
64 bytes from 192.168.0.101: icmp_seq=2 ttl=64 time=1.21 ms
--- 192.168.0.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.217/1.524/2.036/0.365 ms, pipe 2
```

Debe notarse que cuando sea reemplazada la imagen del sistema estos cambios no permanecerán en la memoria flash de la microcomputadora, por lo tanto requerirá recargar la configuración de USBnet, memoria Compact Flash, y programas y directorios asociados a las aplicaciones de usuario.

## C.4. Configuración de memoria externa Compact Flash

La compañía Gumstix ofrece una tarjeta adaptable a su producto principal, la Gumstix WaySmall Computer, sobre la cual provee el soporte necesario para su apropiado funcionamiento. Esta tarjeta es la *cfstix*, con entrada para memoria *compact flash* de primera generación. Han sido realizadas pruebas empleando tarjetas de 256 MB, 512MB y 4 GB.

Por defecto la Gumstix es configurada para soporte de memoria MMC. Esto significa que el soporte para memoria Compact Flash (CF) se encuentra inicialmente deshabilitado. Para deshabilitar el soporte a MMC y activar el soporte a CF, debe editarse el archivo `/etc/modules` y comentar o borrar las líneas relativas a MMC.

Luego deberá sustituir la línea

```
/etc/init.d/s30pcmcia
```

por

```
/etc/init.d/S30pcmcia
```

Los cambios serán ejecutables la próxima ocasión que el sistema sea reiniciado. La tarjeta no es montada al sistema automáticamente, aunque es reconocida de esta forma si ha sido insertada previamente al reinicio del dispositivo. Una vez que haya accedido al sistema de datos de la microcomputadora debe ejecutar las instrucciones que a continuación se enlistan.

1. Inserte la memoria CF mientras la microcomputadora está apagada.
2. Encienda la maquina y espere a que accese al sistema.

3. Para verificar que ha sido detectada por el sistema digite en la terminal de la Gumstix la instrucción:

```
#df -k
```

o ya sea mediante la instrucción

```
#df -h
```

La tarjeta CF debe encontrarse montada sobre `/dev/hda1`, si es así puede proceder al siguiente paso, de lo contrario deberá verificar que los pasos anteriores hayan sido realizados correctamente.

4. Para montar el dispositivo de memoria externa CF, debe crear previamente el directorio `/mnt/CF` y otorgarle propiedades de lectura/escritura:

```
#mkdir /mnt/CF  
#chmod a+rw /mnt/CF
```

5. Ahora ingrese la instrucción para montar el dispositivo sobre el directorio `/mnt/CF/`:

```
#mount -t vfat /dev/hda1 /mnt/CF
```

Hasta este punto la tarjeta de memoria ha sido montada al sistema, de tal forma que al enviar información hacia el directorio `/mnt/CF/` se estará escribiendo sobre la memoria CF.

## C.5. Reemplazo de la imagen del sistema (usando KERMIT)

En caso de daño al software o reparaciones de mayor grado donde sea necesario volver a cargar la imagen del sistema a la Gumstix. La combinación con la que se ha realizado la operación de reemplazo es mediante el empleo de una Laptop Dell Latitude 110L (Linux Slackware 10, Kernel 2.6.10), un conversor USB-Serial, y una Gumstix Waysmall 200ax (Kernel 2.6.18.gum), a la cual se le nombrará simplemente como Gumstix. Aunque el proceso es válido para sistemas genéricos.

Se deberán seguir los pasos siguientes:

1. La Gumstix debe estar apagada.
2. Conecte el conversor USB-Serial a la Laptop (usando el puerto `ttyUSB0`)
3. Conecte el cable serial (null modem, DB9-MiniDIN8) al conversor y Gumstix respectivamente. En la Gumstix el puerto que se deberá utilizar es el `ttyS0` (segundo puerto de la esquina superior izquierda).
4. Accese al directorio de usuario

```
$cd /home/user/
```

5. En una consola escriba el comando

```
$dmesg
```

Deberá aparecer un listado de operaciones, entre las cuales se debe ver:

```
usb 2-2: PL-2303 converter now attached to ttyUSB0
```

6. Y aprovechando que la Laptop ya tiene cargados los *scripts*, entre como superusuario y teclee:

```
$/Cargar_Adaptador_USB_Serial
```

7. Una vez detectado y cargado el adaptador USB-Serial ejecute el programa Kermit, teclee

```
$cd /home/user/download/kermit/  
$kermit
```

La línea del prompt aparecerá como (/home/user/download/kermit/) C-Kermit>

8. Ahora se debe configurar la consola de comunicación serial tecleando los siguientes comandos:

```
$set line /dev/tts/USB0  
$set speed 115200  
$set file type bin  
$set reliable  
$fast  
$set flow-control none  
$set carrier-watch off  
$set prefixing all  
$connect
```

9. Conecte el adaptador de energía a la Gumstix o resetéela (si ya estaba encendida) y presione inmediatamente cualquier tecla (aprox. 2 segundos para que presione la tecla) para detener la ejecución del boot y acceder a la consola del u-boot de la Gumstix (GUM>). Escriba ahora la instrucción:

```
GUM>loadb a2000000 (Note que son 6 ceros)
```

El mensaje que aparece entonces es:

```
Ready for binary (kermit) download to 0xA2000000 at 115200 bps...
```

10. Presione la combinación de teclas CTRL+\ y luego la letra c para salir a la línea de comandos del Kermit. Si lo anterior no funciona, siga los siguientes pasos:
  - a) Mate la ventana del Kermit (llamada terminal en los ambientes linux) en la que se han realizado las instrucciones anteriores, ya sea cerrando la ventana o mediante la instrucción `kill -9 <n\ 'umero de proceso>`, donde el número de proceso es el que corresponda a la consola donde se está ejecutando el Kermit.
  - b) Abra de nuevo una ventana y ejecute una vez más el Kermit como anteriormente se mencionó.
  - c) Vuelva a configurar la consola de comunicación serial como se hizo en el paso 7, pero no escriba `connect`. Ahora la Gumstix se encuentra lista para recibir la transmisión de archivos. Si presenta dificultades al escribir el `$set line /dev/tts/USB0`, vuelva a intentar cerrando la terminal y ejecutando kermit en una nueva.
11. Una vez que se complete el paso anterior, en la misma terminal escriba lo siguiente y espere a que la transmisión concluya, esto tomará algunos minutos:

```
$send /home/user/download/gumstix-buildroot/rootfs.arm_nofpu.jffs2
```

La anterior es la ruta completa de donde se encuentra el archivo `rootfs.arm_nofpu.jffs2`.

Una vez que la transferencia de archivos culmine con éxito, la Gumstix retendrá el nuevo sistema de archivos en RAM. Los contenidos de su memoria Flash están aún intactos, si el proceso se abortara en este punto se conservaría el anterior sistema de archivos.

12. Conéctese de nuevo a la Gumstix escribiendo:

```
$connect
```

Aparecerá de nuevo el prompt de la consola de Gumstix (`GUM>`).

13. Para borrar el anterior sistema de archivos se utilizan los comandos

```
GUM>protect on 1:0-1
GUM>erase all
```

La primer línea protege los sectores 0 y 1 de la memoria Flash, que contiene el u-boot con el que se esta trabajando en ese momento. Esta línea es muy importante. Si se omite, su Gumstix no podrá ser usada y deberá ser recargado el sistema de archivos con un proceso mucho más complicado.

14. Ahora puede confiarse el nuevo sistema de archivos al flash del Gumstix.

```
GUM>cp.b a2000000 40000 ${filesize}
```

Esto fija el nuevo sistema de archivos dentro de la memoria flash de la microcomputadora (en la dirección de memoria 40000H). `$(filesize)` es una referencia a la variable `filesize`, que fue enviada durante la transferencia de archivos. Aunque algunas versiones diferentes de u-boot puede utilizar ya sea `$(filesize)`, `$(filesize)` o `$filesize`.

15. Una vez que la copia este completa, se puede ejecutar el boot de manera directa tecleando

```
GUM>boot
```

Si aparece el siguiente mensaje de error:

```
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(31,2)
```

El nuevo `root_fs` necesita ser montado desde un dispositivo `1f01` en lugar de un `1f02`. Linux cambió la numeración de los dispositivos MTD ordenándolos entre 2.6.11 y 2.6.15 y por lo tanto el dispositivo `root` fue movido en la Flash. En la línea de comandos del u-boot haga lo siguiente (debe resetear la Gumstix y presionar dentro del plazo de dos segundos una tecla para interrumpir el boot y acceder a la consola del u-boot):

```
GUM>setenv bootargs console=ttyS0,115200
GUM>setenv bootargs root=1f01 rootfstype=jffs2 reboot =cold,hard
GUM>saveenv
GUM>bootd
```

Si aparecen muchos mensajes sobre el desconocimiento de nodos, es normal. Puede ignorarlos:

```
## JFFS2 loading 'boot/uImage' to 0xa2000000
Scanning JFFS2 FS: . Unknown node type: 2006 len 4288 offset 0x5ef40
Unknown node type: 2006 len 6480 offset 0x7e6b0
/ Unknown node type: 2006 len 4988 offset 0x9ec84
. Unknown node type: 2006 len 7368 offset 0xbe338
- Unknown node type: 2006 len 10128 offset 0xdd870
```

Cuando se conecte a la Gumstix de nuevo, mientras el puerto `ttyS0` no haya sido bloqueado, después de la instrucción `connect` en Kermit y el reset de la Gumstix el proceso de acceso tardará algunos segundos, no se desespere. Recuerde que debe conectarse a través de la terminal y luego resetear el dispositivo.

## C.6. Cargando El Kernel

El propósito del u-boot es cargar el Kernel de Linux. Esto se hace corriendo los siguientes comandos:

```
GUM>fsload a2000000 boot/uImage
GUM>bootm a2000000
```

Con los apropiados argumentos el boot

Para cambiar los argumentos provistos al Kernel al tiempo de ejecución, se debe ejecutar la siguiente instrucción:

El Kernel 2.6.10gum utiliza:

```
GUM>setenv bootargs console=ttyS0,115200n8
GUM>setenv bootargs root=1f02 rootfstype=jffs2 reboot=cold,hard
```

El Kernel 2.6.13gum y mas recientes utilizan un número diferente para identificar sus particiones en Flash:

```
GUM>setenv bootargs console=ttyS0,115200n8
GUM>setenv bootargs root=1f01 rootfstype=jffs2 reboot=cold,hard
```

Se utiliza `printenv` para desplegar los valores del entorno, y `saveenv` para almacenarlos en la Flash una vez que los valores se consideren satisfactorios.

## C.7. Recuperación de Archivos Corruptos del Boot

Si se requiere corregir algunos de los archivos del cargador (como `/etc/inittab` o algún otro) y el Kernel arranca pero el modo usuario no, puede recuperarse cambiando el programa del `init`. Puede indicar al Kernel el nombre del programa de inicio a utilizar usando `init= opción en la línea de comandos del Kernel`. Por defecto, el Kernel utiliza `/sbin/init`.

Si se agrega `init=/bin/sh` a la línea de comandos del Kernel (variable de argumentos del boot del u-boot) entonces el Kernel los ejecutará directamente.

Se requiere conservar los otros argumentos del boot además de agregar el `init`. La forma más simple de agregarlo es que al final de los argumentos de boot escriba los comandos:

```
GUM>setenv bootargs $bootargs init=/bin/sh
GUM>boot
```

Se puede revisar la funcionalidad de los nuevos archivos del boot ejecutando:

```
GUM>exec init
```

## C.8. Problemas comunes

Si aparece el siguiente mensaje de error:

```
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(31,2)
```

El nuevo `root_fs` necesita ser montado desde un dispositivo `1f01` en lugar de un `1f02`. Linux cambió la numeración de los dispositivos MTD ordenándolos entre 2.6.11 y 2.6.15 y por lo tanto el dispositivo `root` fue movido en la Flash. En la línea de comandos del u-boot haga lo siguiente (debe resetear la Gumstix y presionar dentro del plazo de dos segundos una tecla para interrumpir el boot y acceder a la consola del u-boot):

```
GUM>setenv bootargs console=ttyS0,115200n8  
GUM>setenv bootargs root=1f01 rootfstype=jffs2 reboot=cold,hard  
GUM>saveenv  
GUM>bootd
```

Si aparecen muchos mensajes sobre el desconocimiento de nodos, es normal. Puede ignorarlos:

```
## JFFS2 loading 'boot/uImage' to 0xa2000000  
Scanning JFFS2 FS: . Unknown node type: 2006 len 4288 offset 0x5ef40  
Unknown node type: 2006 len 6480 offset 0x7e6b0  
/ Unknown node type: 2006 len 4988 offset 0x9ec84  
. Unknown node type: 2006 len 7368 offset 0xbe338  
- Unknown node type: 2006 len 10128 offset 0xdd870
```

## C.9. Cambio de Baudrate

Para algunas aplicaciones tales como interconectividad con dispositivos periféricos se requiere disminuir la frecuencia de transmisión de datos a través de un puerto serial específico de la micromcomputadora Gumstix. Para lo cual deberán

ser ejecutados los siguientes pasos:

1. Abrir el puerto serial conforme las instrucciones siguientes (usando Linux C-Kermit):

```
$set line /dev/tts/USB0
$set speed 115200
$set file type bin
$set reliable
$fast
$set flow-control none
$set carrier-watch off
$set prefixing all
$connect
```

2. Resetear la Gumstix, antes de dos segundos oprima una tecla para acceder al *gumstix u-boot*.

Aparecerá el promompt como GUM>

3. Para el ejemplo se cambiará a una frecuencia de 57600 baudios el puerto *ttyS0* (puerto de consola). Teclee lo siguiente:

```
GUM>setenv bootargs console=ttyS0,57600n8
GUM>setenv bootargs root=1f01 rootfstype=jffs2 reboot=cold,hard
GUM>saveenv
GUM>bootd
```

4. Después de establecer las condiciones del boot, la Gumstix reacesará automáticamente con el baudrate anteriormente especificado (el que tenía antes de las instrucciones anteriores). Teclee el usuario y password (definidas por defecto como *root* y *gumstix*, respectivamente, a excepción que usuario haya cambiado dichos parámetros con anterioridad).
5. Se deberá modificar el archivo */etc/inittab*, mediante la siguiente forma:

```
#cd /etc
#vi inittab
```

6. Esto permitirá abrir el archivo *inittab*, utilizando el editor de textos básico VI. Posiciónese en la línea `::respawn:/sbin/getty -L ttyS0 115200 vt100`. Oprima *esc + o*, lo cual le permitirá agregar una nueva línea debajo. Reescriba la línea cambiando el parámetro 115200 por 57600, o la velocidad que requiera (éstas pueden ser de 9600, 19200, 48400, 57600 y 115200. Consulte propiedades de los puertos *ttyS0* y *ttyS2* para frecuencias más elevadas).

7. Posicione el puntero sobre la línea anterior y presionando *esc + d + d*, esta será eliminada.
8. Presione *esc*, enseguida teclee *:wq* y presione *enter*, esto le permitirá salir guardando los cambios en el archivo.
9. Reinicie la Gumstix y conforme las instrucciones del primer paso abra la conexión nuevamente sustituyendo la velocidad por la de la nueva configuración, esto en la instrucción *set speed 57600*, donde sustituirá 57600 por la velocidad apropiada. No olvide especificar esta misma velocidad en el dispositivo transceptor de radio (para el caso del proyecto desarrollado en este trabajo son empleados dispositivos transceptores de radio de 900 MHz, la frecuencia especificada para el puerto serial con el cual el dispositivo transceptor realiza interfaz deberá ser la misma con la cual deban ser configurados los transceptores, ya sea operando en modo punto-a-punto o multi-punto).

NOTA: Si al teclear cualquier cosa en el editor VI sufre un error de tipografía, deberá eliminar la línea completa, presionando *esc + d + d*, y crear nuevamente una línea para insertar el texto nuevo, mediante *esc + o*.

## C.10. Transferencia de archivos

SCP (Secure CoPy) es un programa que permite transferir archivos entre computadoras o sistemas empujados de forma segura, ya que la información es encriptada para que no pueda ser intervenida durante su paso por el medio de transmisión. Para efectuar este proceso recuerde abrir una sesión **root** en la estación de trabajo y otorgar a los archivos atributos de lectura/escritura. Todos los procesos de transferencia de archivos se realizan en el modo superusuario. Siga el siguiente formato:

```
#scp <archivo_local> <usuario>@<maquina_remota>:<archivo_remoto>
```

Por ejemplo, para transferir el archivo `/home/user/termar/programas/LI-7500.exe` de la estación de trabajo (con IP 192.168.0.1) al directorio `/termar/programas/` de la Gumstix programada con la dirección IP 192.168.0.101 conservando el nombre original (si desea cambiarlo, solo modifique el nombre en el destino) debe emplearse la forma

```
#scp /home/user/termar/programas/LI-7500.exe .\  
root@192.168.0.101:/termar/programas/LI-7500.exe
```

Si desea copiar el directorio `/home/user/termar/programas/`, con su contenido completo, al directorio `/termar` de la Gumstix teclee:

```
#scp -r /home/user/termar/programas root@192.168.0.101:/termar/
```

Para descargar (proceso conocido como “*ordeñamiento de datos*”) todos los archivos contenidos en el directorio `/mnt/CF/irga/` al directorio `/home/user/datos_torres/gumstix101/irga/` debe emplear la forma siguiente:

```
#scp root@192.168.0.101:/mnt/CF/irga/* /home/user/datos_torres/gumstix101/irga/
```

Los archivos transferidos desde el usuario `root` de la microcomputadora se encuentran protegidos contra lectura, por lo cual esta característica debe ser retirada.

```
#chmod a+rw /home/user/datos_torres/gumstix101/*
```

## C.11. Acceso remoto

SSH (Secure SHell) es el nombre del programa que sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo el ordenador mediante un intérprete de comandos. SSH emplea técnicas de cifrado que permiten que la información no sea legible mientras viaja por el medio de transmisión. De esta puede para acceder a la Gumstix desde una terminal de consola en la estación de trabajo.

Empleando el ejemplo de acceso remoto a la Gumstix con IP 192.168.0.101 (G101):

```
#ifconfig usb0 192.168.0.1
#ssh root@192.168.0.101
```

## C.12. Configuración de arranque

Se requiere que algunos de los programas corran de forma automática en el momento en que la Gumstix encienda, con la finalidad de asegurar la funcionalidad del sistema de monitoreo. Para esto se debe crear en el directorio `/etc/init.d/` el archivo `S60local`, el cual tiene que contener los comandos de ejecución de los programas de adquisición de datos encargados de la lectura de los puertos `ttyS0` (TorresEF.exe) y `ttyS2` (LI-7500.exe) de la microcomputadora. Además debido a que los datos son almacenados en una memoria externa *Compact Flash*, esta debe ser montada también durante el arranque del sistema.

Accese a la Gumstix y cree el archivo `/etc/init.d/S60local`

```
#vi /etc/init.d/S60local
```

Que deberá contener:

```
#!/bin/bash
echo "Cargando CF y programas de adquisicion de datos de las Torres Eddy Flux"
mount -t vfat /dev/hda1 /mnt/CF
/termar/programas/LI-7500.exe &
/termar/programas/TorresEF.exe &
```

Una vez finalizado lo anterior, otorge el atributo de *ejecutable* a éste archivo desde la consola

```
#chmod a+rx /etc/init.d/S60local
```

## C.13. Liberación del puerto serie ttyS0

El puerto `ttyS0` es destinado por el fabricante como *puerto de consola*. Es a través de este puerto que se puede acceder al sistema de archivos, sin embargo para aplicaciones automatizadas que requieren también de este puerto, se debe silenciar el puerto para evitar que despliegue los mensajes del sistema, los cuales pueden ser confundidos con señales de información por los dispositivos conectados el puerto.

Una vez que los programas y la estructura de archivos se encuentra funcionando apropiadamente y la comunicación vía USBnet es satisfactoria, se debe liberar el puerto `ttyS0`. Estableciendo las variables de ambiente `silent` y `console` del u-boot logrará que los mensaje del u-boot no aparezcan.

Para liberar el puerto de consola se realizan los pasos que a continuación se enlistan:

1. Accesar al u-boot como se describe en el paso 9 de la sección C.5
2. Una vez accesado, para conocer los valores actuales que guardan los argumentos del boot, teclear el comando

```
GUM>pri
```

Aparecerá una lista de mensajes como la siguiente:

```
bootdelay=2
baudrate=115200
serial#=05000700BD3F4693
bootargs=console=ttyS0,115200n8 root=1f02 rootfstype=jffs2 reboot=cold,hard
bootcmd=fsload a2000000 boot/uImage; bootm a2000000
stdin=serial
stout=serial
stderr=serial
Environment size: 221/4092 bytes
```

3. Los comandos encargados de liberar el puerto:

```
GUM>setenv bootargs console=null root=1f02
GUM?setenv bootargs rootfstype=jffs2 reboot=cold,hard
GUM>setenv silent true
GUM>saveenv
GUM>boot
```

Espere un poco. Los mensajes de inicio se reducen a:

```
Uncompressing Linux.....done, booting the kernel
Welcome to the Gumstix Linux Distribution!
gumstix login:
```

El puerto de consola en modo silencioso permanecerá con la capacidad para detectar cuando una tecla es presionada durante los primeros 2 segundos del arranque del sistema.

La línea *"Uncompressing Linux.....done, booting the kernel"* requiere de un proceso un poco más complicado para ser eliminada, sin embargo su presencia no afecta la comunicación con los dispositivos debido a que los enlaces se establecen una vez que el sistema ha iniciado.

4. El resto de los mensajes pueden ser removidos editando el archivo `/etc/inittab`. Sustituya la línea

```
::respawn:/sbin/getty -L ttyS0 115200 vt100
```

por

```
#::respawn:/sbin/getty -L ttyS0 115200 vt100
```

Con lo anterior queda eliminada la posibilidad de acceder el sistema de archivos de la Gumstix.

## C.14. Recuperación del puerto serie ttyS0

El acceso al sistema de archivos de la Gumstix, vía serial al puerto de consola, puede ser recuperado reescribiendo los argumentos del boot:

```
GUM>setenv bootargs console=ttyS0,115200 root=1f02
GUM>setenv bootargs rootfstype=jffs2 reboot=cold,hard
GUM>setenv silent false
GUM>saveenv
GUM>boot
```

y reestableciendo la línea `::respawn:/sbin/getty -L ttyS0 115200 vt100` del archivo `/etc/inittab` retirando el símbolo `#`.

# Apéndice D

## Scripts

Los *scripts* son archivos que contienen uno o más comandos de consola. Estos programas permiten simplificar tareas repetitivas realizadas por el usuario al reemplazar dos o más comandos que son siempre ejecutados juntos, de manera secuencial, como una sola instrucción.

Para crear un *script* debe crear un archivo utilizando un editor de textos y escribir los comandos que desea ejecutar. En lugar de teclear todos los comandos cada vez que se requiera correr un proceso puede ejecutar un archivo que contiene todas estas instrucciones. Una vez que el archivo esté completo, se le debe otorgar un permiso como *ejecutable* para poder correr la aplicación: .

```
#chmod +x <archivo>
```

ahora puede correr el programa tecleando `.\` y el nombre del archivo que contiene las instrucciones deseadas:

```
#./<archivo>
```

### D.1. Adaptador USB-Serial

El adaptador USB a serial debe ser *montado* al sistema cada vez que éste sea conectado, debido a que la versión de Slackware 2.6.10 utilizada en las estaciones de trabajo no hace este proceso de forma automatizada. El archivo `Cargar_Adaptador_USB_Serial` debe crearse dentro del directorio de usuario `/home/user/` con los siguientes comandos:

```
#!/bin/bash
clear
echo 'Cargando convertidor USB-Serial'
rm /dev/modem
ln -s /dev/tts/USB0 /dev/modem
chmod 777 /dev/tts/USB0
```

```
chmod 777 /dev/modem
echo 'Convertidor USB-Serial cargado'
```

## D.2. Memoria USB

### D.2.1. Montaje de memorias USB

En ambientes donde las memorias de almacenamiento externas no son montadas automáticamente por el sistema, este proceso debe ser implementado por el usuario. Recuerde que el formato de almacenamiento empleado corresponde al FAT32. El archivo `Montar_Memoria_USB` contiene las instrucciones necesarias para ello:

```
#!/bin/bash
mount -t vfat /dev/uba1 /mnt/USB
echo 'Memoria USB montada'
```

### D.2.2. Desmontaje de memorias USB

Igualmente la memoria debe ser desmontada del sistema cuando desea desconectarla, de lo contrario errores graves pueden ocurrir en el sistema de archivos de la propia memoria. El archivo `Desmontar_Memoria_USB` contiene la instrucción requerida:

```
#!/bin/bash
umount /dev/uba1
echo 'Memoria USB desmontada'
```

## D.3. Acceso a la Gumstix

Por medio del USBnet puede accederse al sistema de archivos de la microcomputadora Gumstix para realizar procesos de lectura/escritura, así como ejecución de comandos y aplicaciones. Cada Gumstix debe ser previamente provista de una dirección IP (como se trata en la sección C.3) con la forma `192.168.0.10X`. El acceso vía SSH debe ser realizado en modo `root` desde una estación de trabajo. El archivo `Acceso_a_G100`, por ejemplo, contiene las siguientes instrucciones:

```
#!/bin/bash
ifconfig usb0 192.168.0.1
ssh root@192.168.0.100
```

Deberá crear un archivo idéntico para el acceso a cada Gumstix sustituyendo la variable `X` de la dirección IP por la correspondiente a la microcomputadora empleada.

## D.4. Descarga de datos

El archivo `bajar_datos_G10X-Laptop` contiene las instrucciones para descargar los archivos y/o subdirectorios contenidos en el directorio `/mnt/CF/` (la memoria CF debe estar correctamente montada al sistema de la Gumstix) al directorio `/home/user/datos_torres/gumstix10X/` donde son finalmente analizados y respaldados. Se presenta el *script* que permitirá bajar los datos desde la G100 (Gumstix con IP 192.168.0.100) a la estación de trabajo:

```
#!/bin/bash
ifconfig usb0 192.168.0.1
scp root@192.168.0.1:/mnt/CF/* /home/user/datos_torres/gumstix100/
chmod a+rw /home/user/datos_torres/gumstix100/*
```

Los archivos pueden ser directamente descargados a una memoria USB (previamente montada) empleando el *script* `bajar_datos_G100-USB`:

```
#!/bin/bash
ifconfig usb0 192.168.0.1
scp root@192.168.0.1:/mnt/CF/* /mnt/USB/G100/
chmod a+rw /mnt/USB/G100/*
```



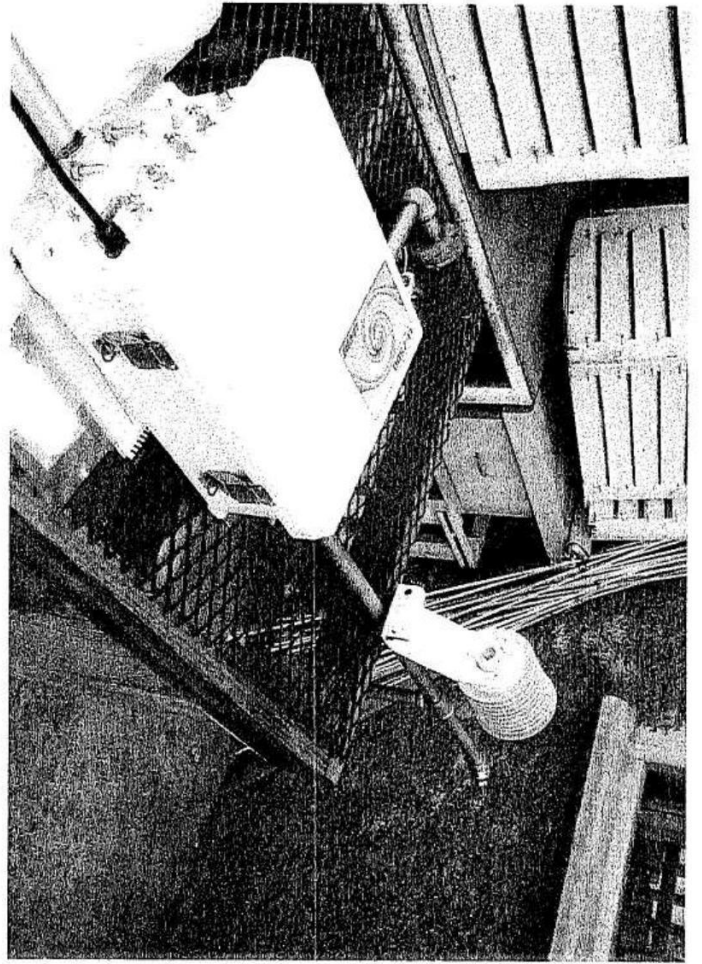
## Apéndice E

# Galería fotográfica TCT

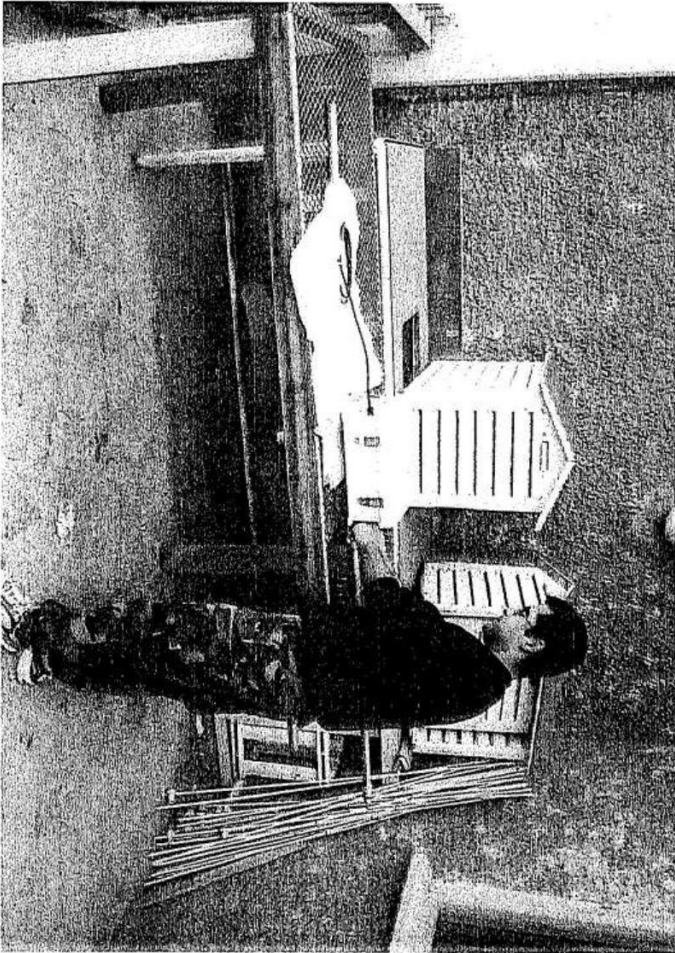
No. Fotografía	Descripción
<b>Seguimiento histórico del diseños de las Torre de Correlacion Turbulenta (TCT)</b>	
1	Obtención de las curvas de calibración para sensores de temperatura
2	Torre experimental antes de ser instalada
3	Ajustes mecánicos a la caja de instrumentación del IRGA
3	Instalación de torre experimental en el cerro del CICESE
5	Todos los dispositivos fueron previamente probados en laboratorio (en la fotografía se muestra el sistema IRGA-Anemómetro)
6	TCT experimental instalada en el cerro del CICESE
7	Torre de correlación turbulenta experimental con panel solar
8	Caja experimental de instrumentación
9	Estación de trabajo en el laboratorio de Electrónica del IA-UNAM
10	Calibración de sensores de temperatura (Edificio de Oceanología del CICESE)
<b>TCT – Rancho “El Mogor” (Valle de Guadalupe)</b>	
11	Armado de la caja de instrumentación que será colocada en la TCT del Rancho “El Mogor” en Valle de Guadalupe
12	Construcción de la caja donde serán guardadas las baterías de ciclo profundo empleadas por la TCT
13	Instalación de cables tensores de la TCT
14	Ajuste de tuercas de conexión de los tramos de la TCT
15	Vista del Valle de Guadalupe donde fue colocada la TCT
16	Mantenimiento a la estructura de la TCT
17	Mantenimiento a la estructura de la TCT
18	Colocación de extensiones de soporte para sensores
19	Transporte de material hacia el lugar de trabajo
20	Instalación de montura para panel solar y construcción de base para la caja que contendrá las baterías del sistema
<b>TCT – Isla de Todos Santos Norte</b>	
21	Transporte de personal y equipo a bordo del barco de apoyo oceanográfico “Calhypso”
22	Descarga a bordo de la panga “Genus” a la isla
23	Movilización de equipo y personal vía terrestre hacia el sitio de trabajo
24	Transporte de la estructura de la TCT por vía marítima
25	Vista hacia la Isla de Todos Santos Sur
26	Construcción de la base de la torre
27	Ensablaje de los tramos restantes de la TCT
28	Colocación de los cables tensores



1



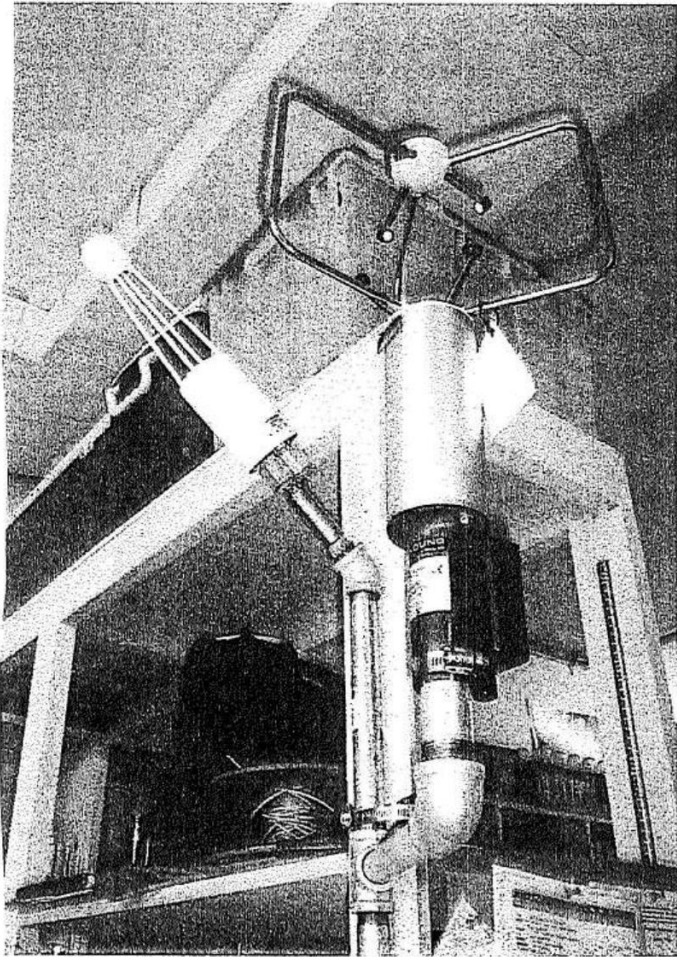
2



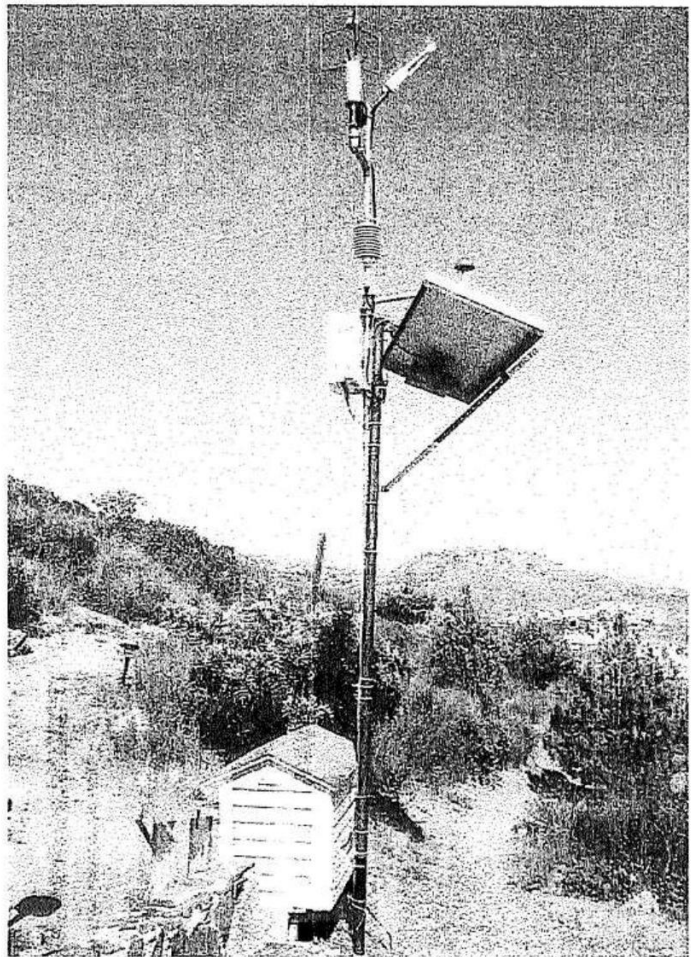
3



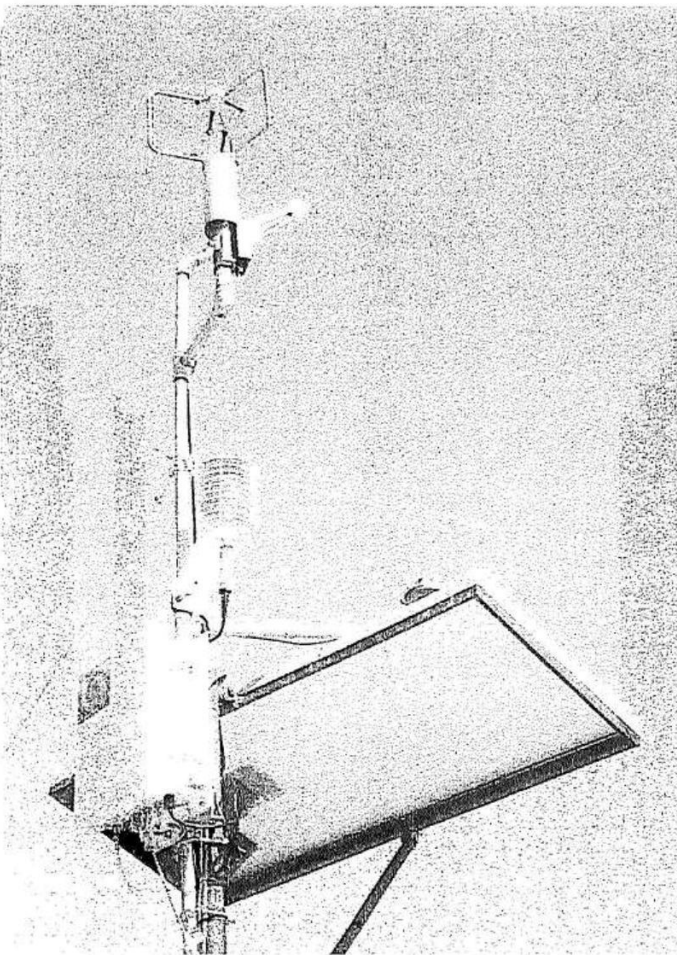
4



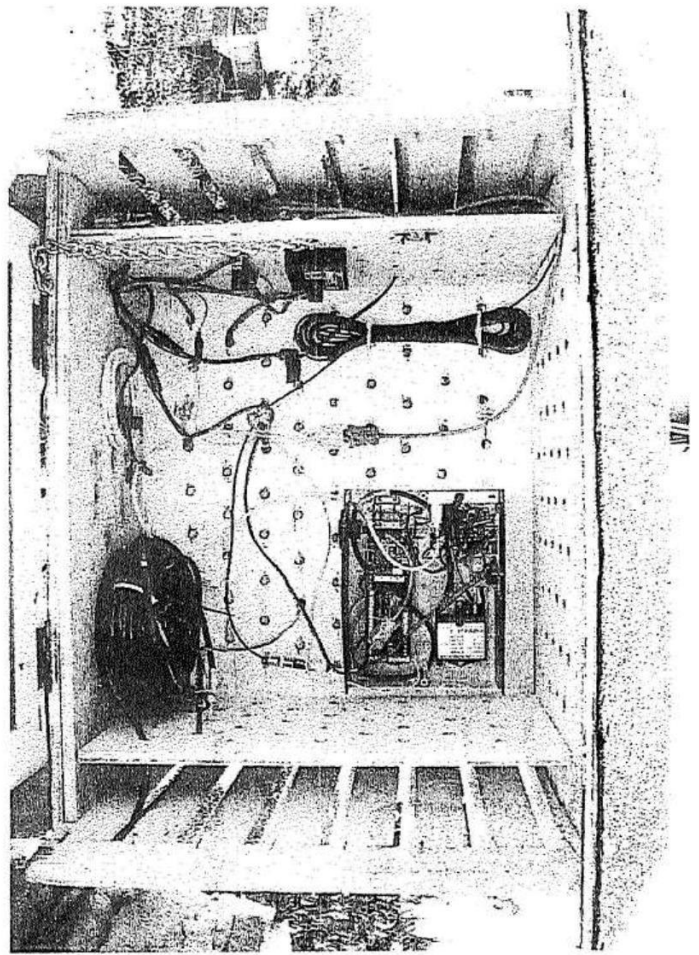
5



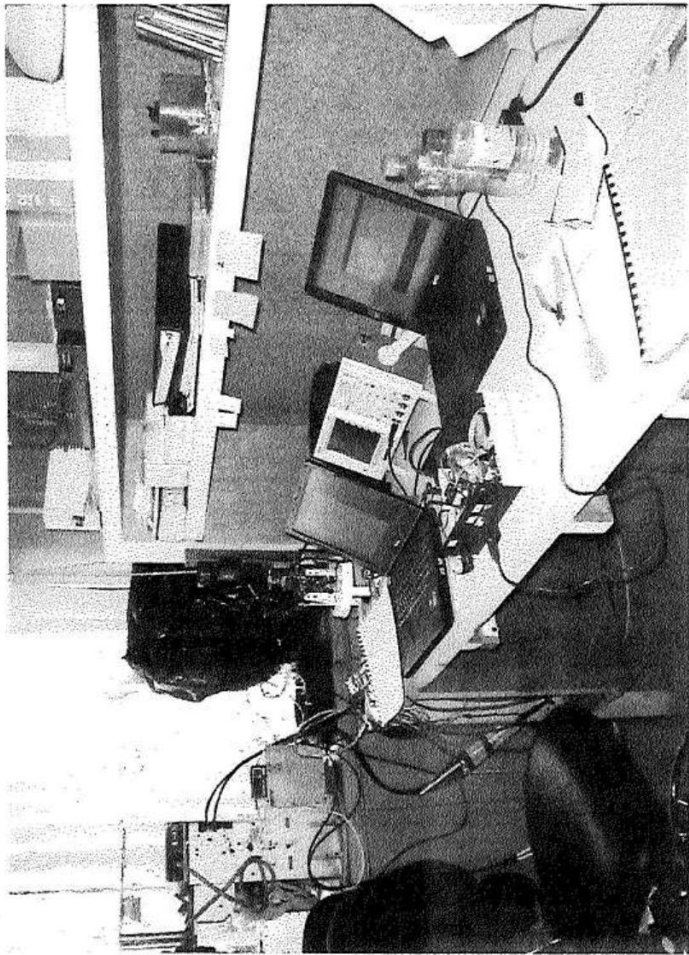
6



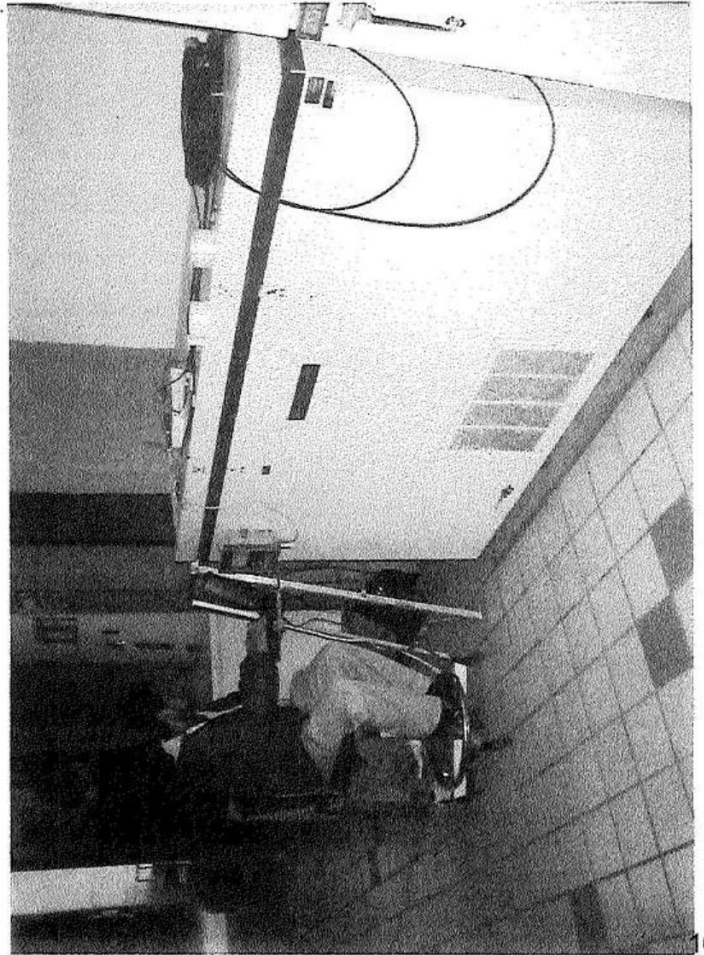
7



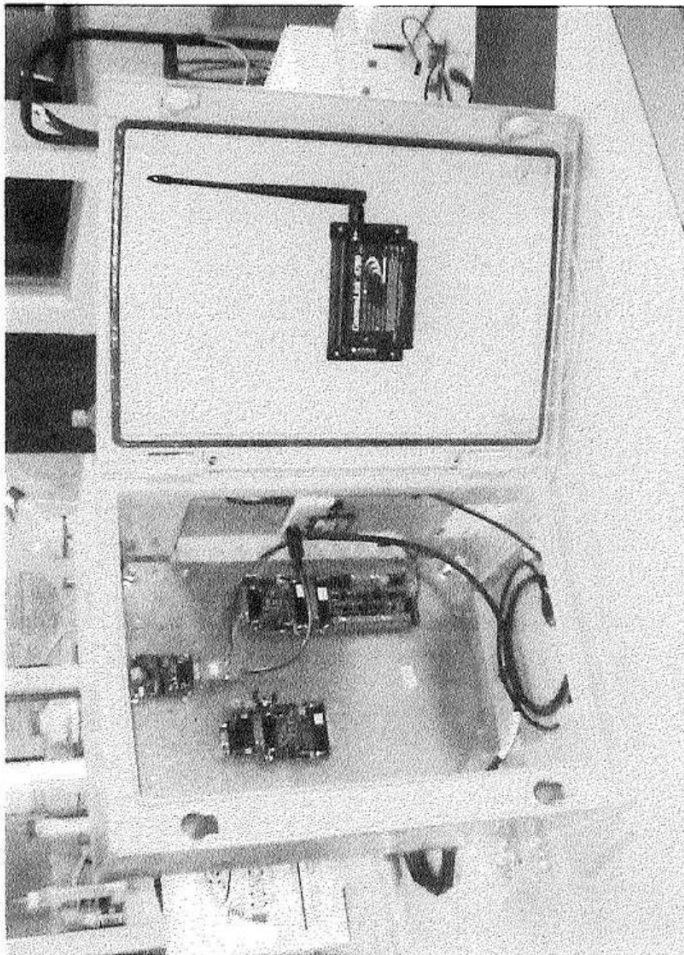
8



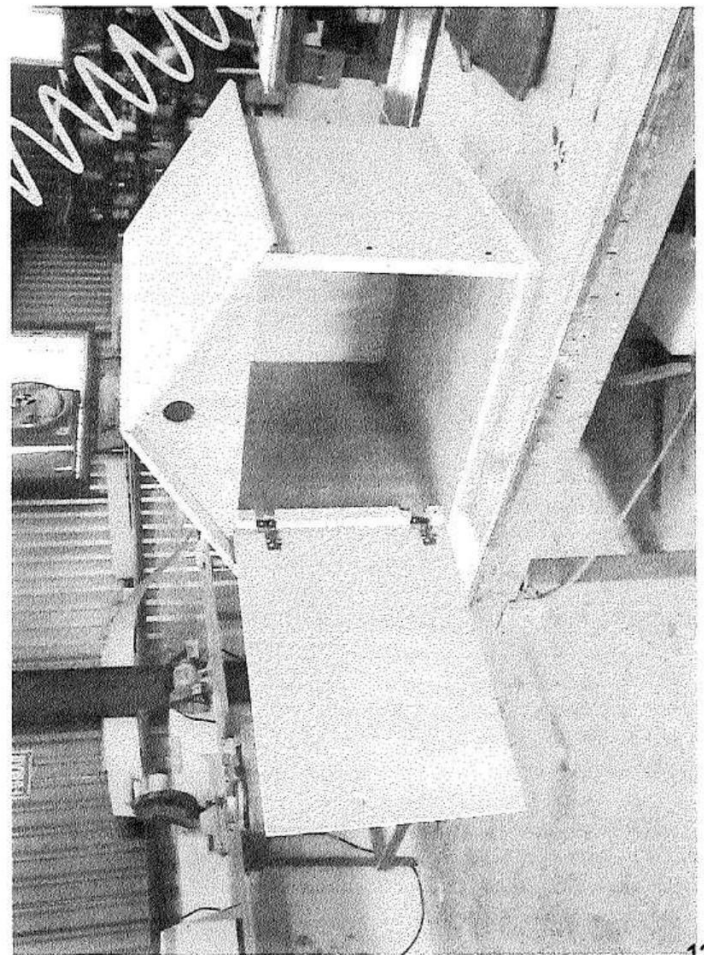
9



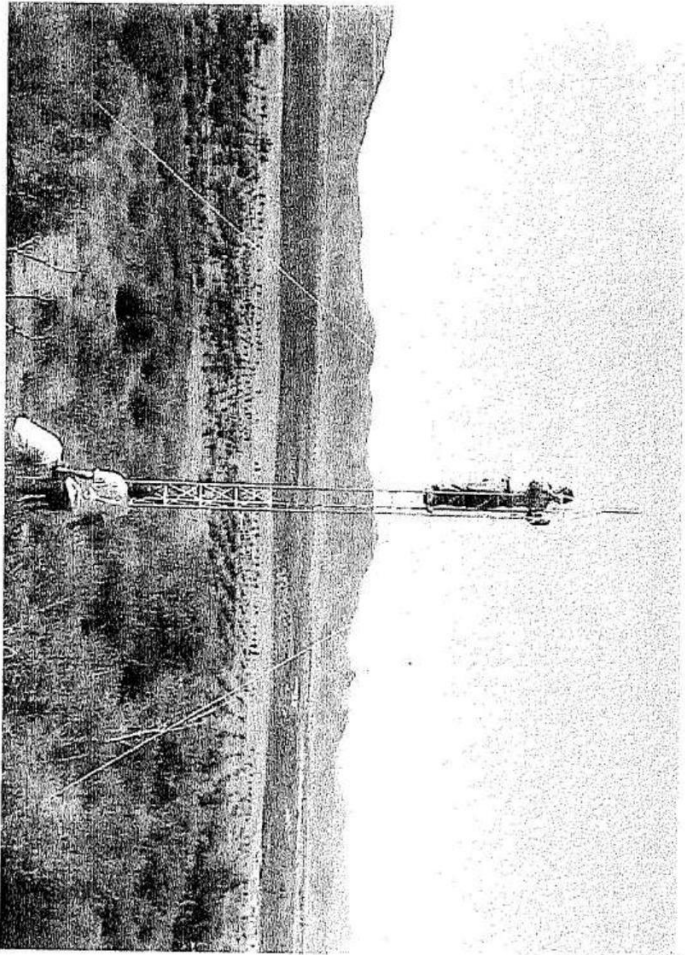
10



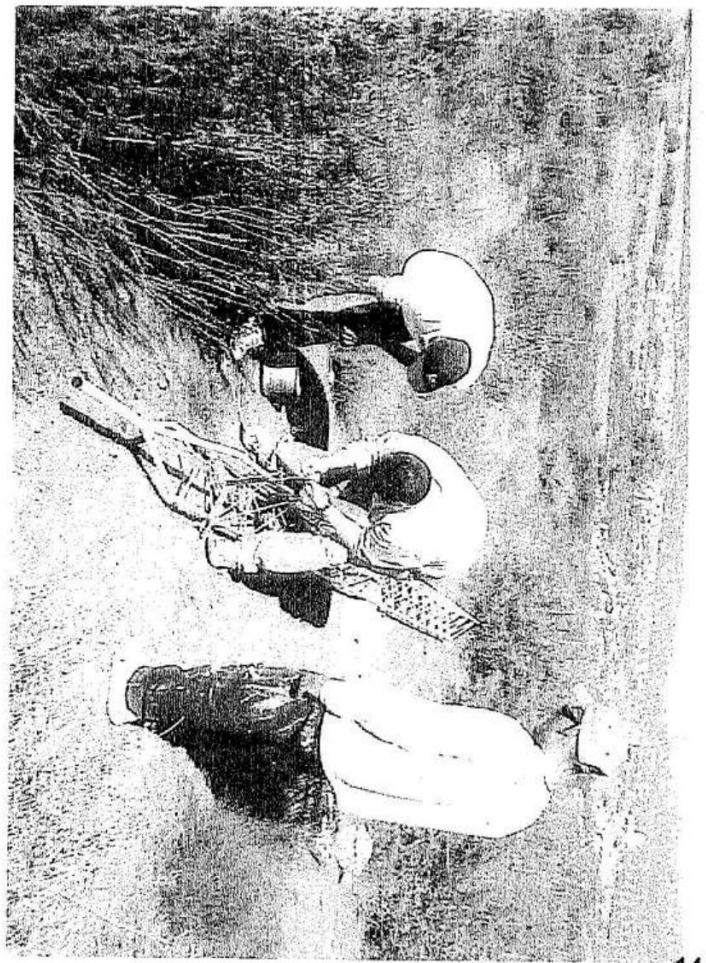
11



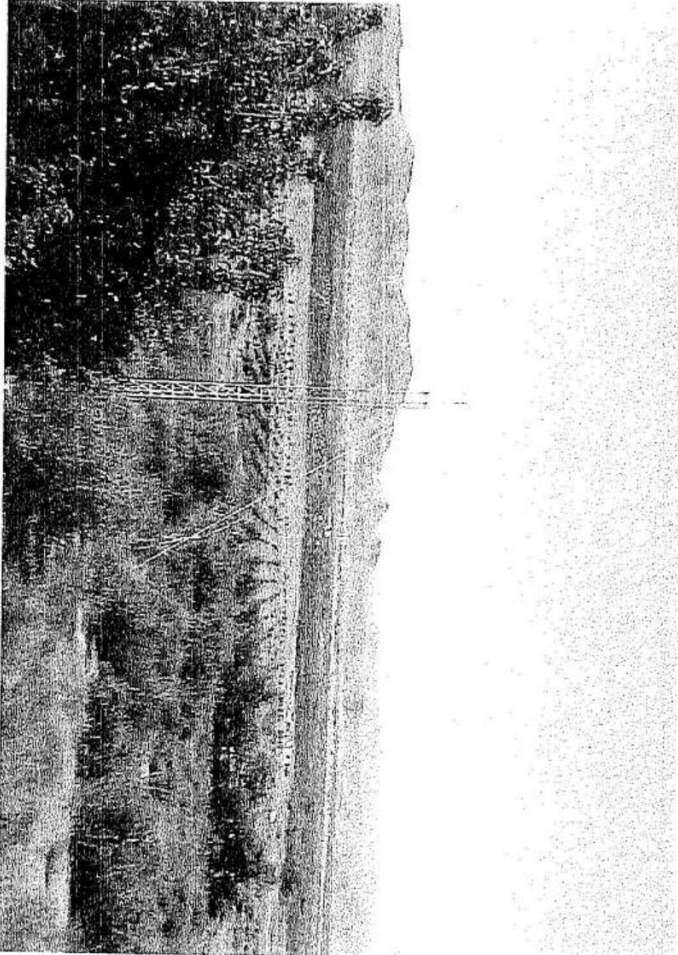
12



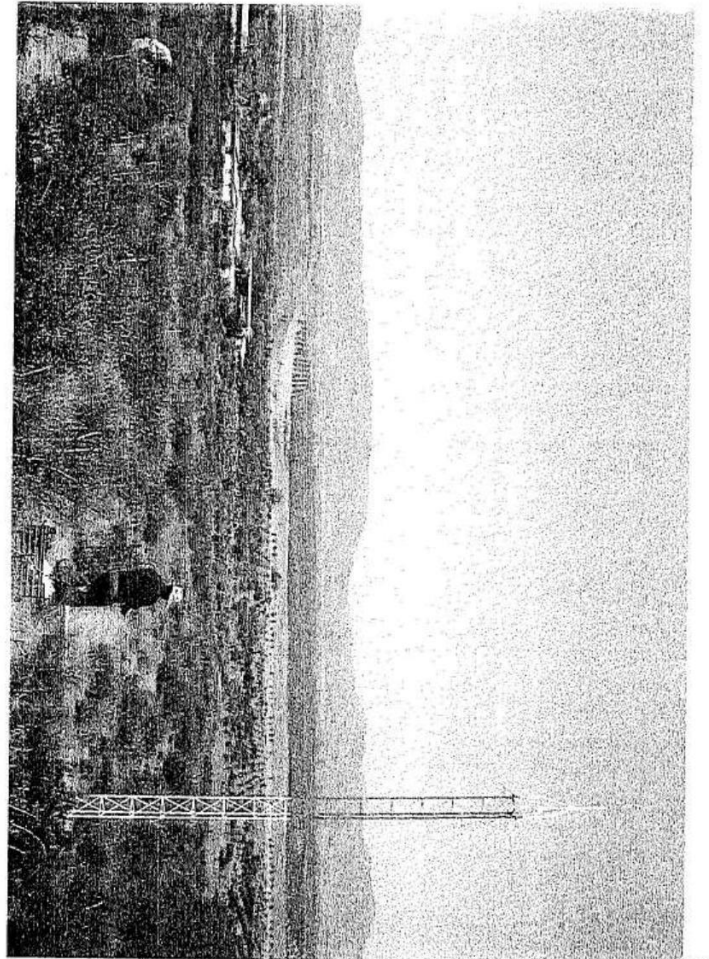
13



14



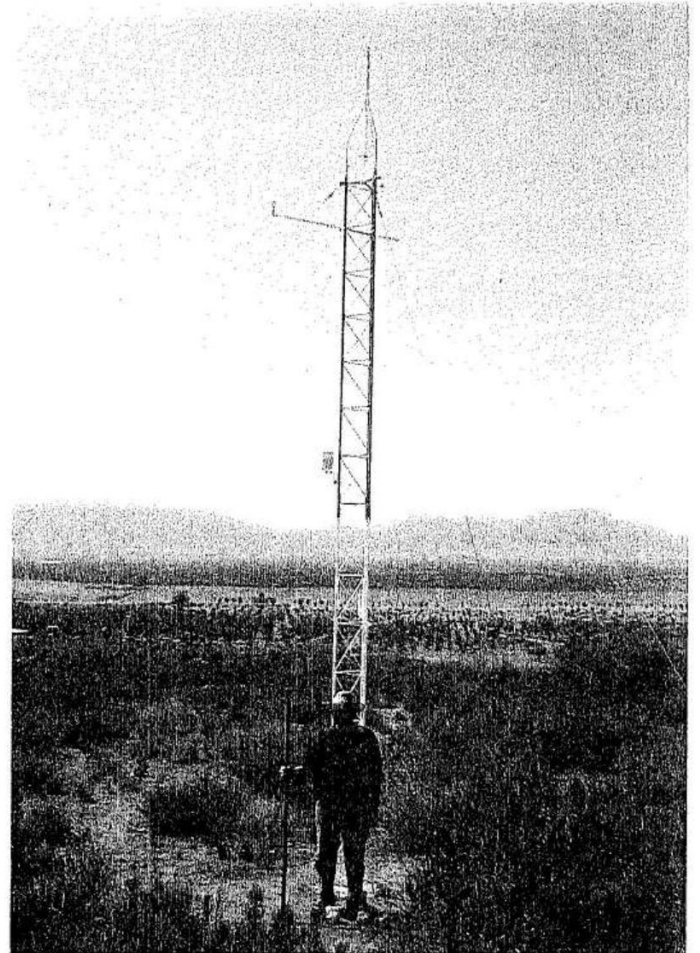
15



16



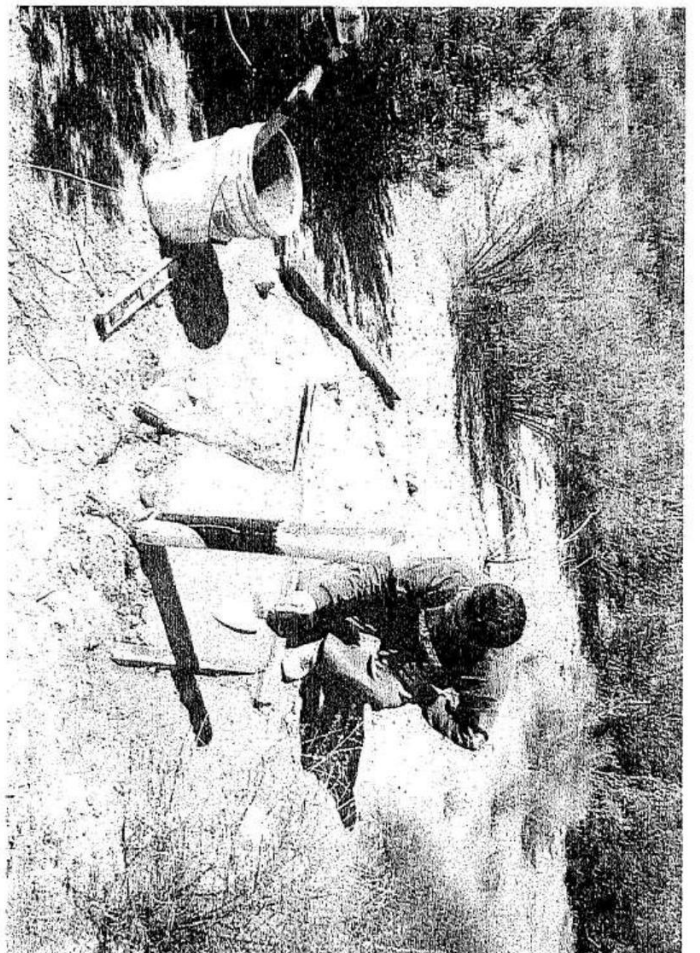
17



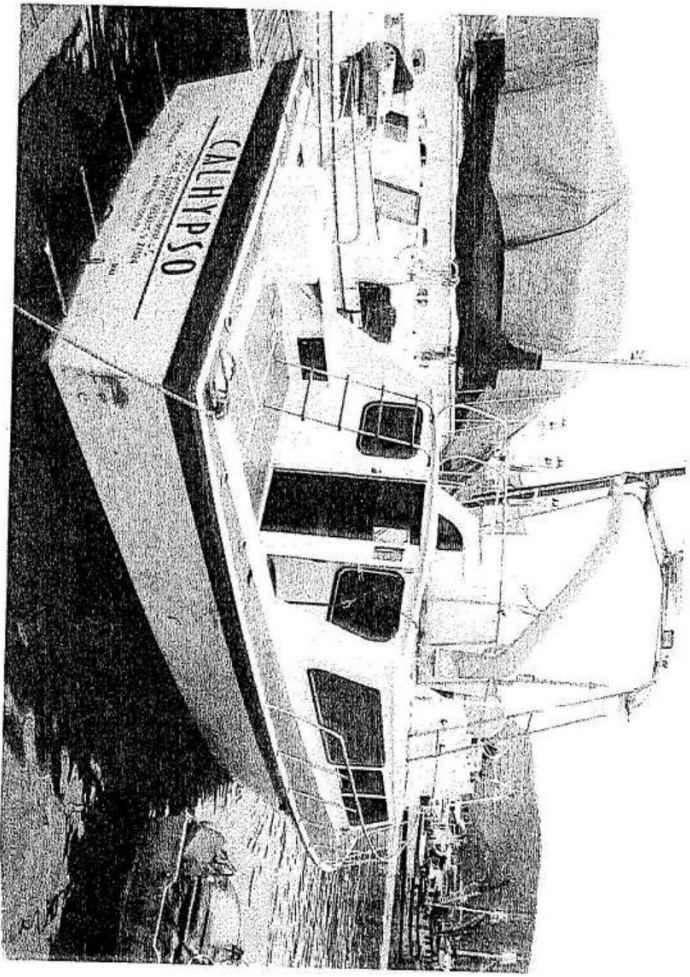
18



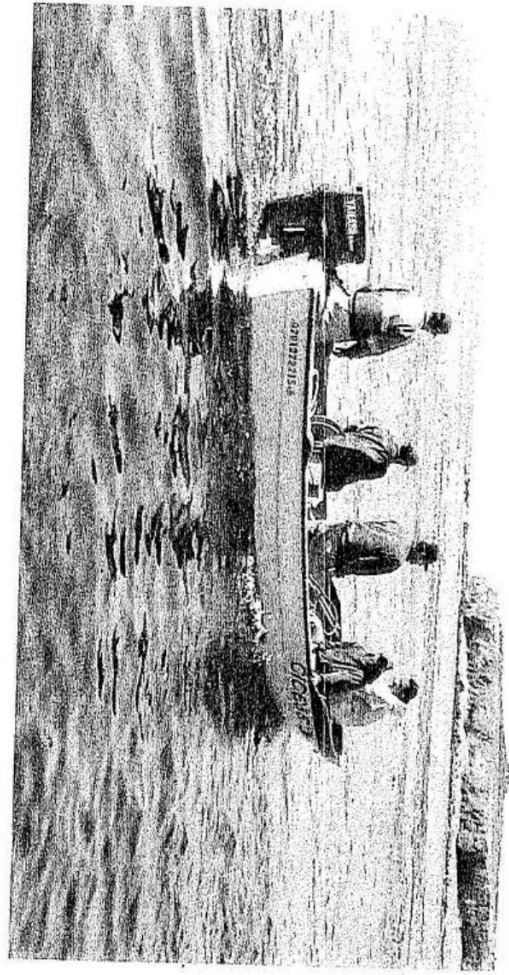
19



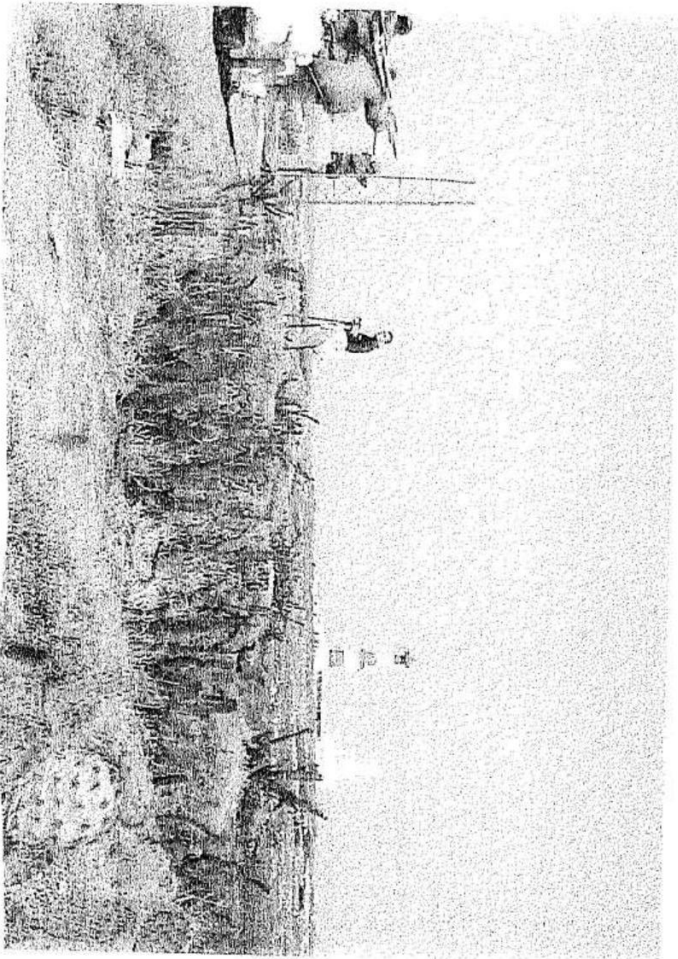
20



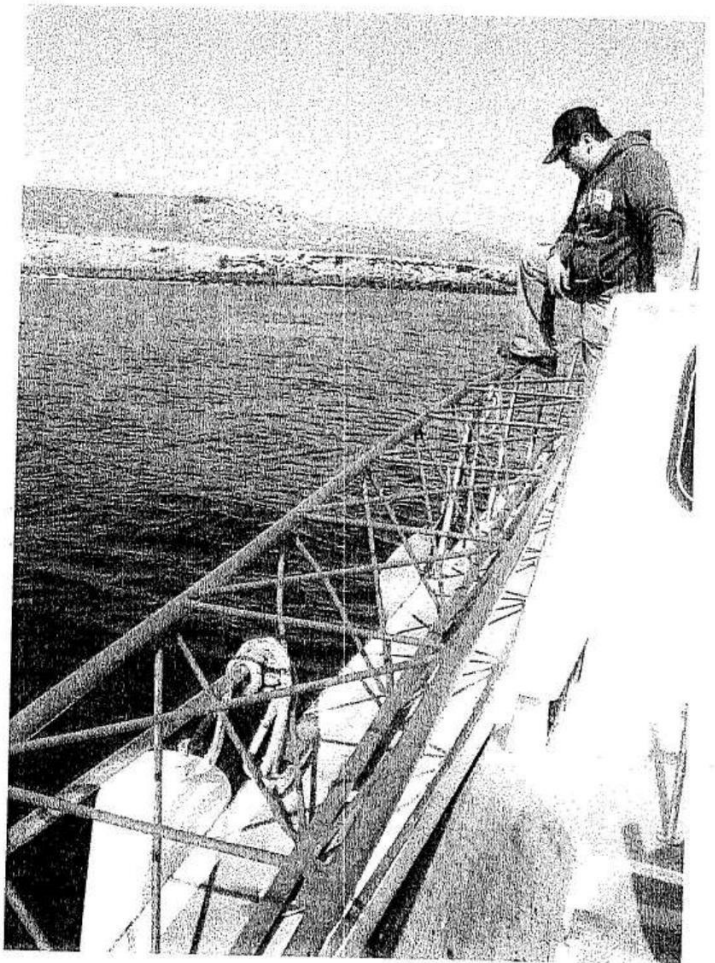
21



22



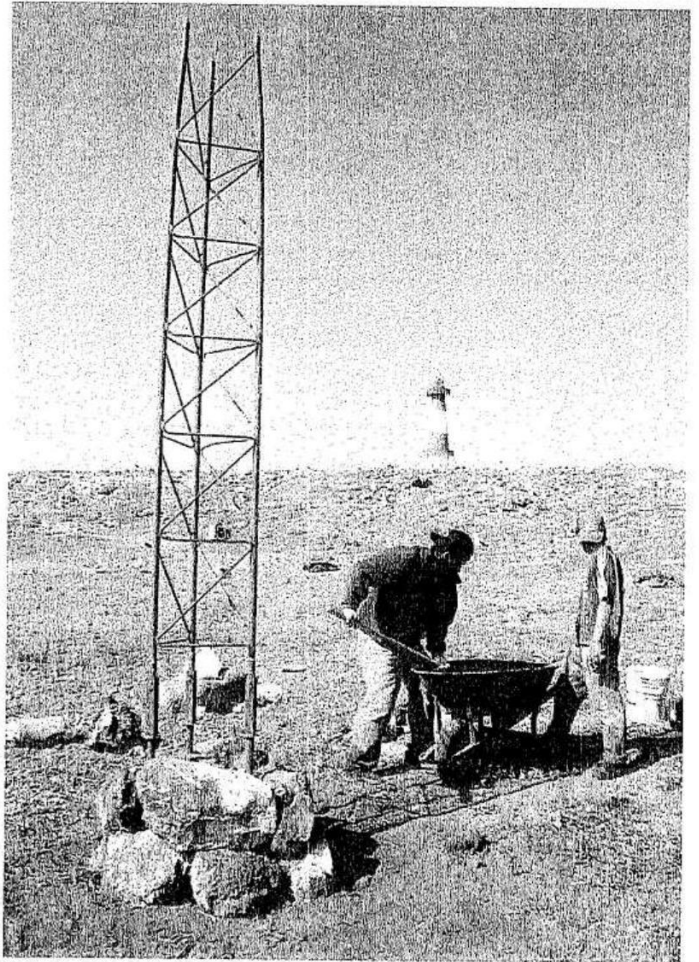
23



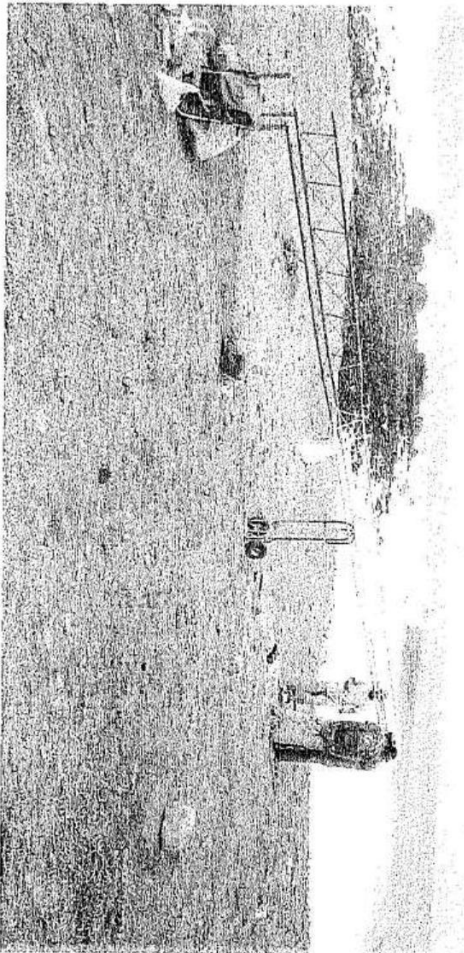
24



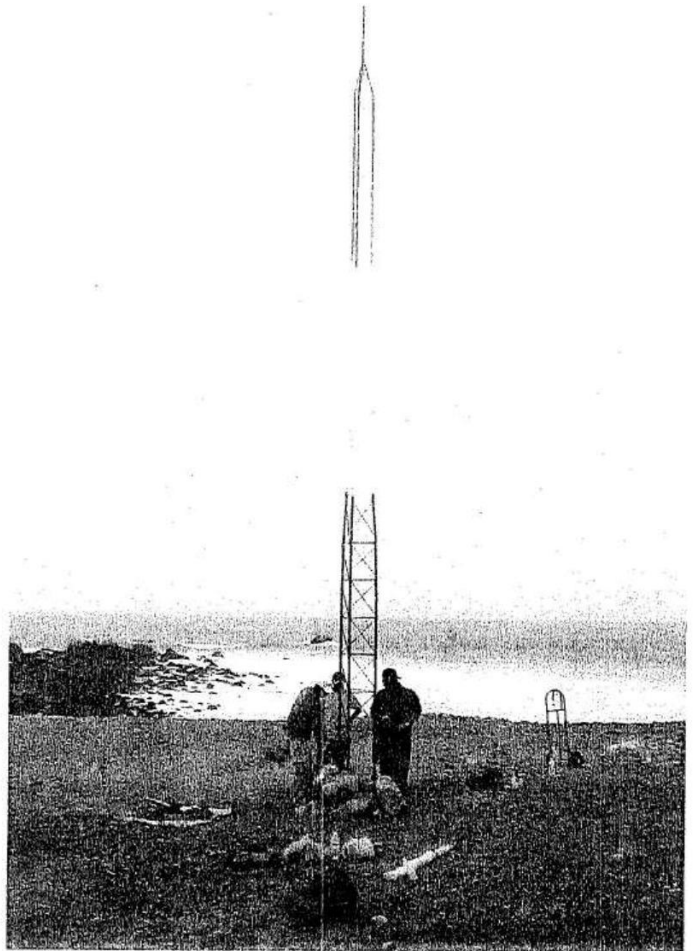
25



26



27



28