



CCADET

CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Informe Técnico

SISTEMA DE ADQUISICIÓN DE DATOS MEDIANTE USB

Autores:

Juan Ricardo Damián Zamacona

Daniel Hernández Rivera

Sergio Quintana Thierry

José Castillo Hernández

Febrero de 2012

Proyecto Interno

SISTEMA DE ADQUISICIÓN DE DATOS MEDIANTE USB

Juan Ricardo Damián Zamacona, Daniel Hernández Rivera, Sergio Quintana Thierry, José Castillo Hernández

Centro de Ciencias Aplicadas y Desarrollo Tecnológico, Circuito Exterior S/N, C.P.
04510, A.P.70-186, Universidad Nacional Autónoma de México, D.F., México.

RESUMEN

El presente trabajo describe el diseño y construcción de un equipo que registra y almacena la información de diferentes sensores empleados en estudios de sueño en una memoria USB. Dado que en algunos estudios del sueño no es necesario capturar la señal EEG, el sistema de captura se reduce ya que en ocasiones solamente se requiere la captura de algunas señales tales como el ronquido, la frecuencia cardíaca, la frecuencia respiratoria, la expansión torácica y la oximetría de pulsos, es por ello que a estos sistemas se les conoce como sistemas simplificados, estos sistemas por lo general son portátiles y el paciente los puede llevar a su hogar para monitorear su sueño cómodamente; es por esto que se hace necesario contar con un sistema que almacene la información necesaria en una memoria y que después se pueda desplegar en una computadora para su posterior estudio. En este trabajo se muestra el diseño de un sistema de adquisición de datos en el que los datos obtenidos del exterior son almacenados en una memoria flash USB, aún sin tener una computadora en el lugar de captura, para lograr este tipo de almacenamiento, se utilizó un módulo del fabricante FTDI el cual nos proporcionó la herramienta necesaria para lograr que un microcontrolador almacene datos en la memoria mencionada. Este proyecto, es una continuación de un proyecto que se desarrolló en conjunto entre la clínica del sueño de la Facultad de Medicina de la UNAM y el CCADET; dado que ellos tienen la necesidad de realizar estudios ambulatorios que en principio no requiere de un estudio más completo, el proyecto mencionado, consistió en el desarrollo de un polisomnógrafo para estudios del sueño. El sistema simplificado presentado en este trabajo, ha dado buenos resultados ya que se han podido almacenar las variables básicas en una memoria USB.

Índice

Introducción.....	4
1. Sistema de Adquisición de Datos.....	5
2 . Sistema Propuesto.....	6
3 . Memoria USB.....	8
4. Circuito Integrado FTDI.....	9
5. Etapas del Sistema Propuesto.....	10
6. Sensado de Variables.....	10
6.1 Flujo respiratorio.....	11
6.2 Expansión Torácica.....	11
6.3 Sensor de Saturación de Oxígeno en Sangre.....	12
6.4 Ronquido.....	13
6.5 Sensores ECG.....	14
7. Acondicionamiento de Señal.....	15
7.1 Circuito Electrónico de Acondicionamiento.....	16
8. Conversión A/D y Envío de Datos.....	21
8.1 Selección del Microprocesador.....	21
8.2 Multiplexado y Conversión A/D.....	22
9. Almacenamiento de Datos.....	23
10. Comunicación y Programación del Módulo VDIP1.....	24
11. Visualización de Datos.....	26
12. Resultados.....	27
12.1 Interfaz de Usuario.....	29
12.2 Prototipo Logrado.....	30
Apéndice A.....	33
Apéndice B.....	40
Apéndice C.....	56
Referencias Bibliográficas.....	76

INTRODUCCIÓN

Dado que los sistemas de adquisición de datos son esenciales en diversos ámbitos tales como en áreas industriales, de investigación, salud, seguridad, medioambiente, etc., es de gran importancia que cada sistema se adecue a las necesidades propias que se requieran.

Los sistemas de adquisición de datos pueden ser muy diferentes dependiendo del tipo de señales involucradas, primordialmente atendiendo a su frecuencia, amplitud, fase, si es digital o analógica, etc. de acuerdo a éstas características, se presentan necesidades diferentes en la adquisición, estas necesidades definirán los elementos y funciones que tendrá el sistema de adquisición de datos.

En este trabajo se presenta una alternativa basada en un medio digital como lo es una memoria USB para almacenar datos sin contar con una PC en el lugar de captura, posteriormente esta información es vaciada mediante un programa adecuado en una PC para su análisis. Una aplicación directa es la captura de señales biológicas generadas por personas que se encuentran durmiendo, son varias las señales que requieren ser conocidas por el médico especialista para poder hacer un diagnóstico adecuado de cada paciente y de esta manera detectar enfermedades del sueño tales como las apneas, trastornos respiratorios como el ronquido, etc. Esto hace necesario sensor diferentes señales biológicas tales como el flujo respiratorio, el cual puede ser nasal o bucal, el ronquido, la expansión torácica, la posición corporal, el ritmo cardiaco, la saturación de oxígeno en sangre, además de la señal ECG.

Para la correcta presentación de las variables mencionadas al especialista, es necesario un tratamiento adecuado de estas, lo cual involucra varios pasos tales como:

- Sensado de la variable, involucra la selección del elemento transductor adecuado así como su caracterización.
- Acondicionamiento, lo cual es necesario dadas las características de voltaje y corriente que entregan los sensores.
- Amplificación, esta etapa es importante dado que para la correcta visualización de los resultados es necesario un nivel adecuado a todas las señales.
- Filtrado, con esta etapa se eliminan todas las componentes de frecuencia indeseables, las cuales pueden ocasionar lecturas erróneas.
- Aislamiento, con esta etapa se protege al paciente de alguna descarga eléctrica peligrosa.

Finalmente estas señales son almacenadas para posteriormente realizar su lectura, es en esta etapa en la que se presentan los resultados de una manera adecuada para su correcta interpretación mediante un software diseñado especialmente para este sistema.

Es así como en este trabajo se trata la manera en la que cada una de las variables del sueño es procesada siguiendo los pasos anteriores y generando un equipo portátil que integra todos los pasos mencionados.

1. SISTEMAS DE ADQUISICIÓN DE DATOS

De manera general, se pueden utilizar de acuerdo al tipo de señal a capturar, dos tipos de sistemas de adquisición de datos, analógicos o digitales, estos últimos son los más utilizados en la actualidad dada su versatilidad. En la figura 1 (a) y (b) se muestran los bloques básicos que podrían tener un sistema analógico y digital respectivamente.

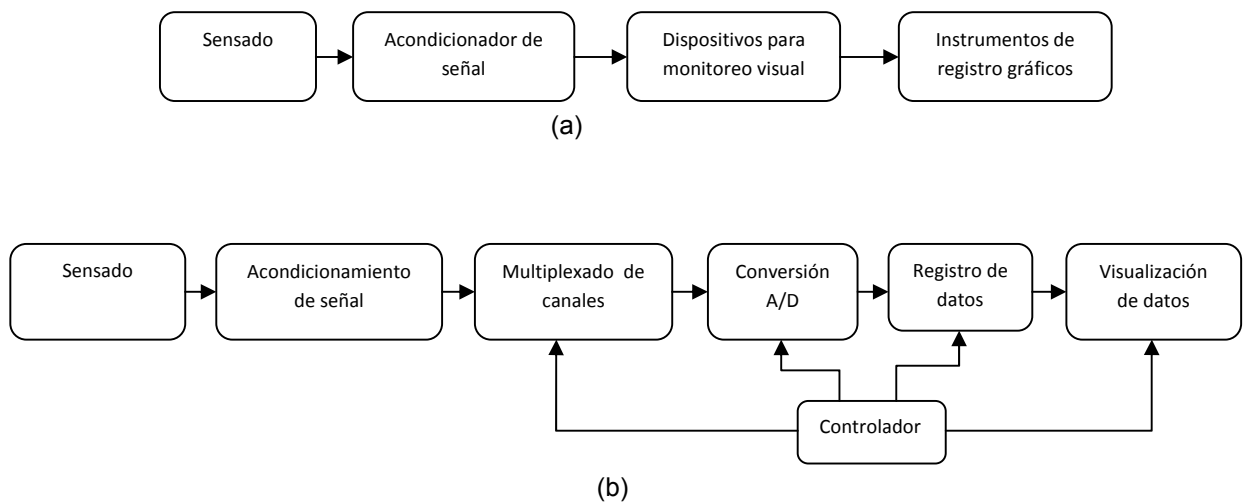


Figura 1. Diagrama a bloques de un sistema de adquisición de datos analógico (a) y digital (b)

En la actualidad se le da mayor uso a los sistemas de adquisición de datos digitales por los beneficios que presenta en cuanto al manejo de ruido ya que se puede regenerar la señal ante la pérdida de información, además se pueden emplear técnicas de detección y corrección de errores, por otro lado cualquier operación que se requiera realizar sobre la señal es muy práctico dadas las técnicas computacionales existentes, todo esto facilita el procesamiento de este tipo de señales (Cooper W.,1991). Por su parte, los sistemas analógicos, son sistemas con aplicaciones donde principalmente se requiere un ancho de banda grande, además tienen la ventaja de no requerir conversión A/D ó D/A con lo cual se pierde cierto grado de información en el proceso.

2. SISTEMA PROPUESTO

El sistema aquí propuesto cuenta con todas las etapas de un sistema digital mencionadas anteriormente, poniendo especial énfasis en el bloque de registro de datos ya que éste, se realiza mediante el uso de una memoria flash USB, esto nos permitirá coleccionar información por tiempo prolongado sin hacer uso de una PC en el mismo lugar donde se tiene el sistema de adquisición de datos, como se muestra en la figura 2. Posteriormente haciendo uso de una interfaz, se visualizan los datos en una computadora.

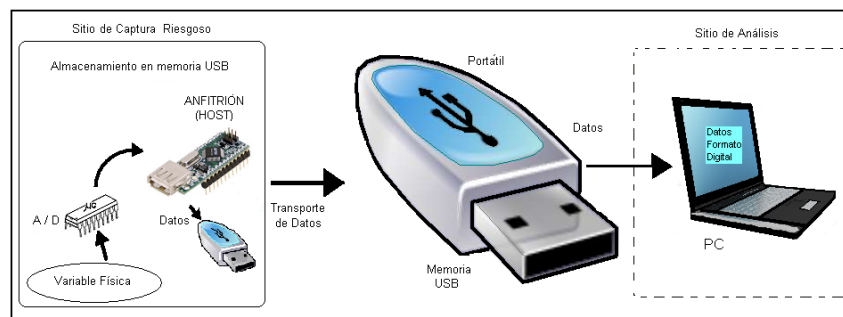


Figura 2 Esquema propuesto

Al comparar el sistema propuesto con otros similares, como el mostrado en la figura 3. Se emplea también la memoria USB como medio de almacenamiento, o en su defecto, el disco duro. En este esquema, se tiene una PC en el mismo sitio de captura que funciona como “Anfitrión”, con los consecuentes riesgos que esto acarrea, tales como el daño permanente de la PC debido a inclemencias del ambiente e incluso el extravío de la misma, dado su elevado costo. En este caso, al igual que en el anterior, también es necesario digitalizar los datos para poder procesarlos con la computadora. Finalmente, y una vez que se tienen capturadas las señales, la computadora completa se traslada al sitio en el que se analizarán los datos de manera más detallada.

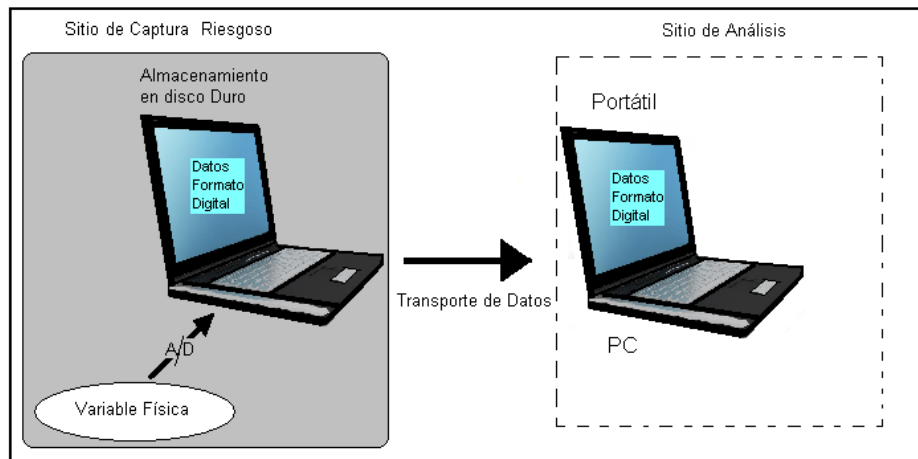


Figura 3. Captura de información usando una PC.

Otro esquema que es muy similar a los anteriores, sería el mostrado en la figura 4. Como se puede ver en este esquema también es necesario el uso de una PC como "Anfitrión" para el almacenamiento de la información en la memoria USB, solamente que ésta información, es transportada al sitio de análisis por medio de la misma memoria USB, permaneciendo la PC en el lugar de captura. En el sitio de análisis se debe contar con otra PC para poder leer los datos almacenados en la memoria.

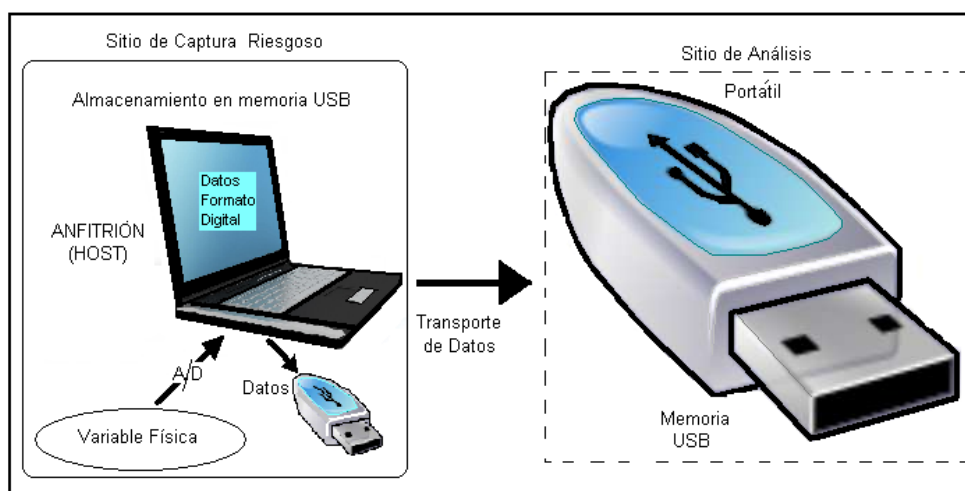


Figura 4. Transporte de información en memoria USB

3. MEMORIA USB

La memoria flash USB actualmente es muy práctica, de amplia distribución y bajo costo, además, es de completa compatibilidad con los equipos de computo actuales. El inconveniente que se tiene con este tipo de sistemas y el cual es uno de los problemas a resolver, es que no cualquier dispositivo puede escribir en una memoria USB, para esto se necesita tener generalmente un dispositivo anfitrión (HOST) como lo es una PC, ya que solo este dispositivo maestro tiene el control en una comunicación USB con dispositivos USB esclavos. Este esquema de almacenamiento típico de información se muestra en la figura 5.

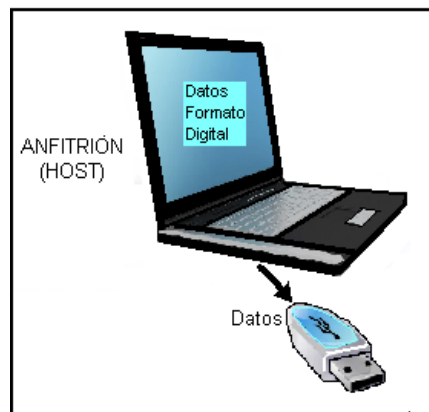


Figura 5. Memoria flash USB

Para que la información pueda ser grabada correctamente en la memoria para el sistema aquí propuesto es necesario formatearla previamente.

5. ETAPAS DEL SISTEMA PROPUESTO

El esquema de la figura 7, muestra de manera más específica las etapas del sistema de adquisición de datos propuesto.

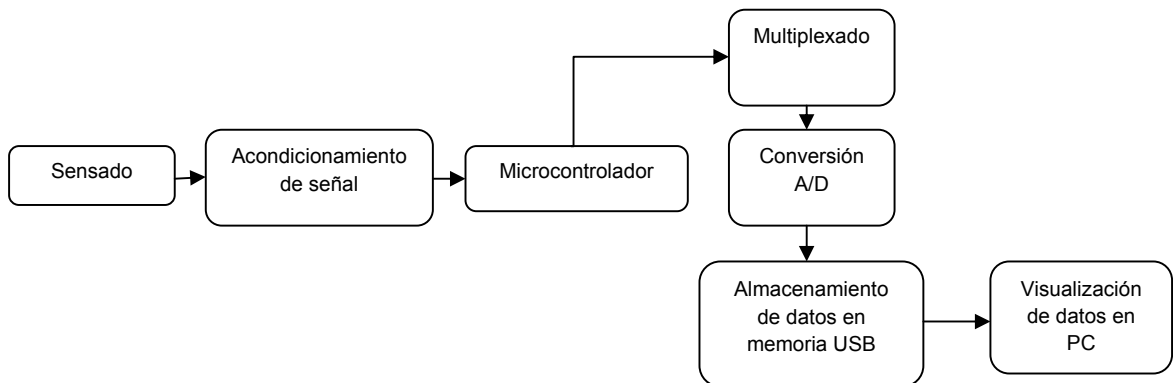


Figura 7. Diagrama general del sistema de adquisición de datos

6. SENSADO DE VARIABLES

La importancia de los sistemas de adquisición de datos radica en poder obtener, manipular y analizar datos que nos proporcionan las características de las señales que estamos estudiando, el primer bloque del sistema mostrado en el apartado anterior, requiere de dispositivos transductores adecuados a las señales de interés.

Los sensores captan magnitudes físicas y las transforman en magnitudes eléctricas que podemos manipular en nuestro sistema, de esta manera necesitamos sensores especiales para la magnitud que deseamos obtener.

Hay una gran variedad de sensores disponibles en el mercado, estos sensores nos entregan una señal eléctrica de diferentes magnitudes y frecuencias, a su vez estas señales deben ser acondicionadas a nuestras necesidades para posteriormente manipularlas de manera adecuada (Sapiensman, 2011).

El sistema aquí presentado está diseñado para señales de baja frecuencia menores a 100 Hz, como podrían ser señales biológicas (ecg, flujo respiratorio, expansión torácica etc.), señales meteorológicas (humedad, velocidad del viento, precipitación etc.), señales geológicas (hundimiento, humedad etc.) y algunas otras señales de algún proceso industrial.

En particular el sistema se probó con señales biológicas tales como flujo respiratorio, expansión torácica, frecuencia cardíaca, saturación de oxígeno y ronquido.

Para tener una idea del tipo de señales generadas por estos sensores, se caracterizaron de tal forma que se pudieran conocer sus características principales.

6.1. Flujo respiratorio

Para el sensado de esta variable se tienen disponibles sensores del tipo termopar. Este elemento es colocado a la entrada de las fosas nasales por lo que se aprovecha el cambio de temperatura al inhalar y exhalar aire para monitorear el ritmo respiratorio. Esta señal es enviada para su acondicionamiento a un amplificador de instrumentación.

La sensibilidad del termopar es de 2 a 30 $\mu\text{V}/^\circ\text{C}$, la amplitud de la señal puede variar de manera considerable dependiendo de la posición del paciente, el flujo de aire del paciente, la posición al dormir, la temperatura de la habitación etc.

La amplitud típica de la señal a la salida del sensor es del orden de 400 μV pico a pico, la resistencia de salida es de 7ohms, y el conector es del tipo estándar 1.5mm, además de 2.13 m de longitud en el cable. En la figura 8 se muestra el sensor utilizado para detectar el flujo respiratorio.

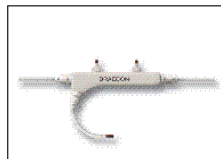


Figura 8. Sensor de flujo respiratorio

6.2. Expansión torácica

Para detectar la expansión torácica se utilizó un sensor de esfuerzo respiratorio piezoeléctrico con una sensibilidad de 2-30 $\mu\text{V}/\text{mm}$, la amplitud típica a la salida del sensor es de 600 μV pico a pico y para su conexión tiene un cable de 2.13m., además cuenta con un conector “plug” invertido macho de 1.5mm. La figura 9 muestra el sensor empleado.



Figura 9. Sensor de expansión Torácica

6.3. Sensor de saturación de oxígeno en sangre

Para poder tener una lectura de SpO_2 se utilizó un circuito integrado comercial que ya tiene su propia curva de calibración, este circuito se conoce como “ChipOx”. Este es un circuito integrado de propósito específico al cual por un lado se le conecta el sensor dactilar mostrado en la figura 10 y por otro lado nos entrega resultados de SpO_2 , para ello es necesario programar el protocolo de comunicación para poder obtener la información necesaria de SpO_2 y presentarla al usuario. Además de esta información, el circuito nos puede proporcionar otros datos tales como la frecuencia cardiaca, el cual es también un dato de vital interés. La figura 10 muestra una imagen del dispositivo electrónico “ChipOx” montado en una tarjeta y el sensor dactilar.

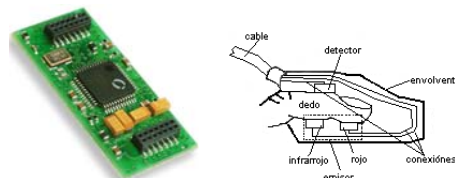


Figura 10 Módulo para monitoreo de SpO_2 y sensor dactilar

Para su funcionamiento, el ChipOx requiere de una fuente externa de 3V aproximadamente y el sensor (dactilar). El ChipOx tiene dos conectores denominados BU1 y BU2 con los cuales se comunica al exterior, la tabla 1 especifica el tipo de información presente en cada Pin del conector BU1 y BU2.

Tabla 1 . Pines de conexión de BU1 y BU2

Pin	Conector BU1	Conector BU2
1	3V	NC
2	GND	aislamiento
3	RESET	I.D. sensor
4	I/O-0 (TXD)	A. Reseptor
5	I/O-1 (RXD)	K. Reseptor
6	I/O-2(Adr-AO-0)	A. Rojo
7	I/O-3(Adr-AO-1)	A,IR
8	I/O-4(Adr-AO-2)	AIN-0
9	I/O-5(AO-Strobe)	AIN-1
10	I/O-INT	AIN-2
11	Salida Analog.	NC
12	NC	NC
13	NC	NC
14	NC	NC

Como se puede ver de la tabla 1, un conector es de entradas y otro de salidas. De estas terminales se toman las conexiones que van al sensor dactilar y también las salidas de información. Adicionalmente se tienen salidas analógicas multiplexadas en un conector disponible el cual es mostrado en la tabla 2 y que se identifica como X13. Estas señales provienen de un circuito multiplexor SMP08 el cual se alambra en conjunto con el ChipOx, las señales analógicas se toman de la terminal 11 del conector BU1.

Tabla 2. Conexión de señales analógicas al exterior del chipOx

Parámetro	Intervalo	Resolución	Terminales(X13)header
GND			#1
SpO2	0-100%	10mV/%SpO2	#2
Frecuencia de pulso	0-300bpm	3.33mV/bpm	#3
Calidad de señal	0-100 en 10 pasos	100mV/paso	
Pletismograma	0-255	3.9mV/LSB	#5

6.4. Ronquido

Para sensar esta variable se empleó un micrófono de tipo “electreto” encapsulado de tal manera que no moleste al paciente durante el sueño dado que se debe colocar en la garganta. La figura 11 muestra una imagen del sensor empleado.

Este sensor proporciona una salida típica de +/- 5mV la cual es amplificada y filtrada para eliminar todo el ruido no deseable.



Figura 11. Micrófono para detectar ronquido

6.5. Sensores ECG

Los sensores empleados para detectar las señales de electrocardiografía (ECG) son de tipo concha (para estudios de larga duración) como los mostrados en la figura 12, tienen un diámetro de aproximadamente 10 mm y cuentan con una perforación en el centro para facilitar la aplicación de gel conductor.



Figura 12. Sensores de ECG

Los electrodos de ECG no solamente capturan esta señal, sino también, el ruido electromagnético presente en el ambiente, la fuente más importante de ruido es por las líneas eléctricas las cuales introducen una componente de 60Hz. Este ruido normalmente es más grande que la señal de ECG y suele ser del orden de entre 10mV y 1 V mientras que la señal de ECG es del orden de entre 1 a 3 mV.

7. ACONDICIONAMIENTO DE SEÑAL

La magnitud de las señales eléctricas que generan los sensores, es de baja amplitud, por lo que hacen difícil su procesamiento, adicionalmente a este hecho, se pueden tener otras causas que dificultan el procesamiento tales como la presencia de ruido, la atenuación de la señal por la distancia recorrida, el rango de voltaje de los dispositivos controladores, etc. Es por estas razones que se hace necesaria una etapa de acondicionamiento que nos permita manipular de la mejor manera nuestras señales, para que al final se tengan los datos lo más confiables posibles (Sapiensman, 20011).

La amplificación de la señal, eliminación de ruido e interferencias y el manejo del offset, son las principales características que se requieren en el bloque de acondicionamiento y que son necesarias debido a que las señales serán introducidas a un microcontrolador por lo que se hace especial énfasis en el rango de voltajes que maneja este, de 0 a 5 Volts.

La etapa de acondicionamiento cuentan básicamente con un amplificador de instrumentación, un filtro pasivo paso altas y otro paso bajas, una etapa más de amplificación junto con el manejo de offset.

La amplificación de la señal es un método efectivo para mejorar la relación señal a ruido (S/R), generalmente se logra haciendo uso de amplificadores operacionales de instrumentación. Estos amplificadores se utilizan para señales de baja intensidad generalmente provenientes de sensores (Saravia E., 2004). En este sistema se utilizó el Circuito integrado AD621 que contiene en un solo chip un amplificador de instrumentación. En la siguiente etapa, se utilizó filtro activo paso bandas para eliminar el ruido y limitar el ancho de banda a las frecuencias de las señales obtenidas.

La última parte cuenta con una etapa más de amplificación que nos permite obtener la magnitud adecuada junto con el manejo de offset proporcionado por un amplificador en configuración sumador. En la figura 13 se muestra el diagrama general de la etapa de acondicionamiento.

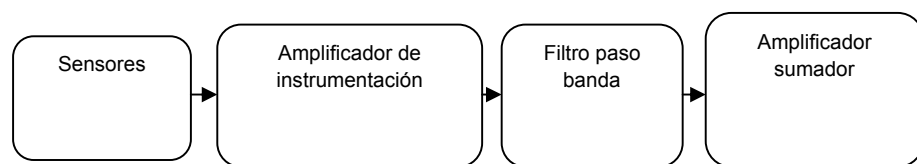


Figura 13. Diagrama a bloques de la etapa de acondicionamiento

7.1. Circuitos Electrónicos de Acondicionamiento

En este apartado se presentan los diferentes circuitos empleados para acondicionar cada una de las señales provenientes de los sensores empleados.

En la figura 14 se muestra el circuito de acondicionamiento de la señal que proviene del sensor de flujo nasal. Como se presentó anteriormente, esta etapa está integrada por un Amplificador de Instrumentación AD621 ajustado con una Ganancia de 10, la entrada de señal del sensor está referenciada a tierra. Las características eléctricas del AD621 se muestran en la tabla 3 así como sus terminales de conexión en la figura 15.

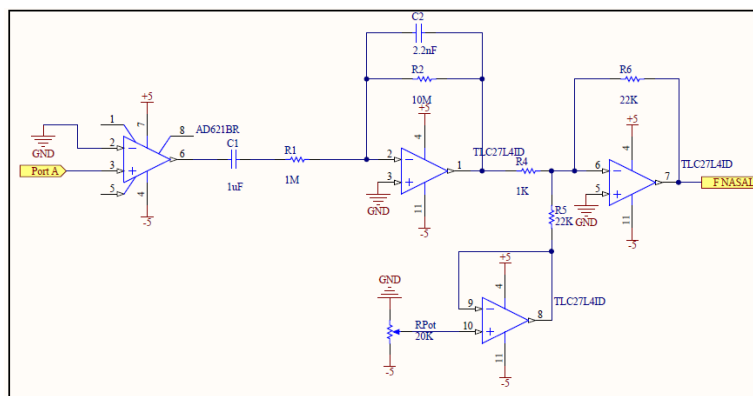


Figura 14. Acondicionamiento de flujo nasal

Después del AD621 se tiene un filtro activo conocido como de banda ancha con frecuencias de corte f_L de 0.159Hz y f_H de 7.23 Hz. La ganancia del filtro es de 10. Para el cálculo de las frecuencias se empleó la ecuación 1.

$$f_C = \frac{1}{2\pi RC} \dots\dots\dots(1)$$

La siguiente etapa del circuito acondicionador como se puede ver en la figura 14, es una combinación de nivel de offset con un amplificador operacional en configuración inversora. Éste último tiene una ganancia de 22 para con esto tener una ganancia total de 2200 a la salida de ésta etapa.

Empaque plástico Mini-Dip (N) Cerdip (Q) y SOIC (R), 8 terminales

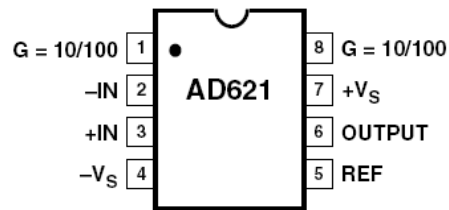


Figura 15. Terminales de conexión del AD621

La figura 16 muestra el circuito electrónico para el acondicionamiento de la señal que proviene del sensor de flujo bucal, este es prácticamente el mismo que se utilizó para el acondicionamiento de la señal de flujo bucal por lo que la ganancia total de esta etapa también es de 2200.

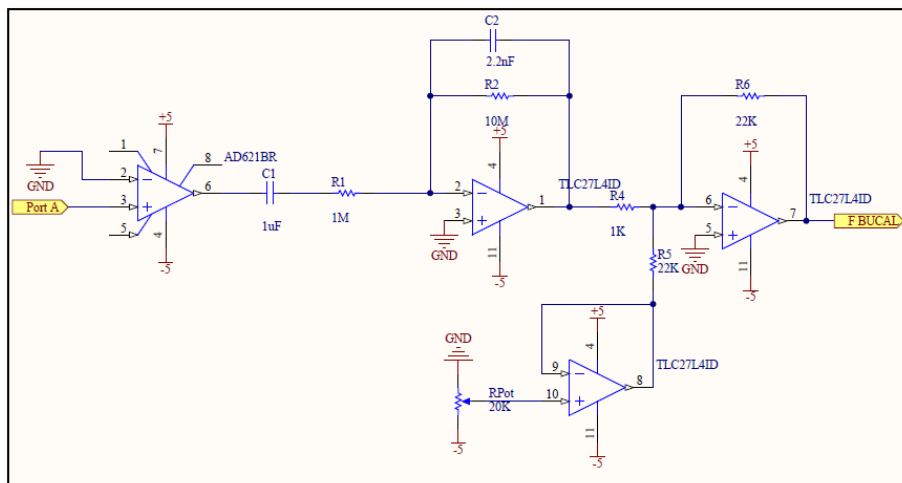


Figura 16. Acondicionamiento de flujo bucal

Tabla 3. Características eléctricas del AD621

Característica AD621	Valor
Ganancia	10 a 100
Fuente de alimentación	(+/-2.3 a +/-18V)
Consumo	1.3mA Max
Disipación de potencia	650mW

Error en la ganancia total	0.15% Máx
Deriva en la ganancia total	(+/- 5PPM/°C)
Voltaje de offset total	125uV Max.
Deriva en el voltaje de offset máxima	1uV/°C Max.
Voltaje de ruido de entrada	9nV/Hz ^{1/2} a 1kHz
Ruido	0.28uV p-p (0.1Hz a 10 Hz)
Ruido en la corriente de entrada	0.1pA/Hz ^{1/2}
Corriente de bias	1.5nA Max.
BW	800kHz a G=10, 200kHz a G=100
CMRR	110 dB
Impedancia de entrada diferencial	10 GΩ//2pF
Impedancia de entrada común	10GΩ//2pF

El hecho de monitorear flujo bucal y nasal es debido a que algunas personas respiran inconscientemente por la boca y otras lo hacen por la nariz. De esta forma, siempre se tendrá el dato de la respiración de la persona en estudio.

La figura 17 muestra el circuito de acondicionamiento de la señal de expansión torácica. Se puede ver que el frente de entrada es similar a los anteriores circuitos, incluyendo las frecuencias de corte y las ganancias totales, esto es debido a que estas tres señales presentan características muy similares a la salida de sus respectivos sensores, por lo que su tratamiento es igual.

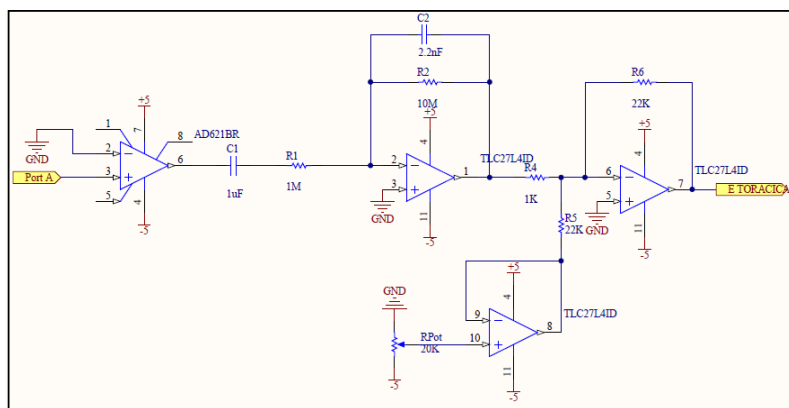


Figura 17. Acondicionamiento de expansión torácica

El acondicionamiento de la señal que proviene del sensor de audio, usado para detectar ronquido se presenta en la figura 18.

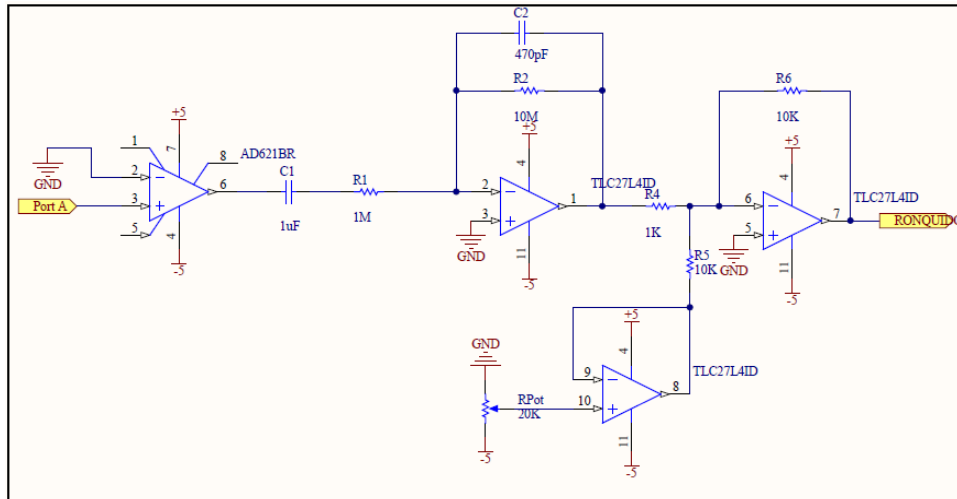


Figura 18. Circuito acondicionador de ronquido

En este caso las frecuencias f_L y f_H del filtro de banda ancha están ajustadas a 0.159 Hz y 33.8 Hz respectivamente, esto con la finalidad de no perder tanta información en el proceso de acondicionamiento. La ganancia total a la salida de esta etapa para este circuito es de 1000.

El circuito empleado para acondicionar la señal de ECG se muestra en la figura 19. Como se puede ver en esta figura, la entrada que proviene de los electrodos de prueba es conectada en forma diferencial al AD621. El tercer electrodo que en este caso es la referencia se conecta directamente al la tierra del circuito.

La ganancia de la primera etapa en este circuito es de 10, igual que en los circuitos anteriores pero en este caso se le agregó un circuito restaurador de línea base con la finalidad de tener una señal que no este variando al montarse sobre una componente de DC. Para tal fin se toma una muestra de la señal de salida del AD621 y se inyecta a la terminal de referencia del mismo circuito, integrando previamente esta señal. Las frecuencias f_L y f_H para este caso están ajustadas a 0.48Hz y 7.23Hz dado que se ha visto que con estos valores la forma de onda de la señal ECG se aprecia claramente. La ganancia total a la salida de esta etapa también es de 1000 como en el caso de la señal de ronquido.

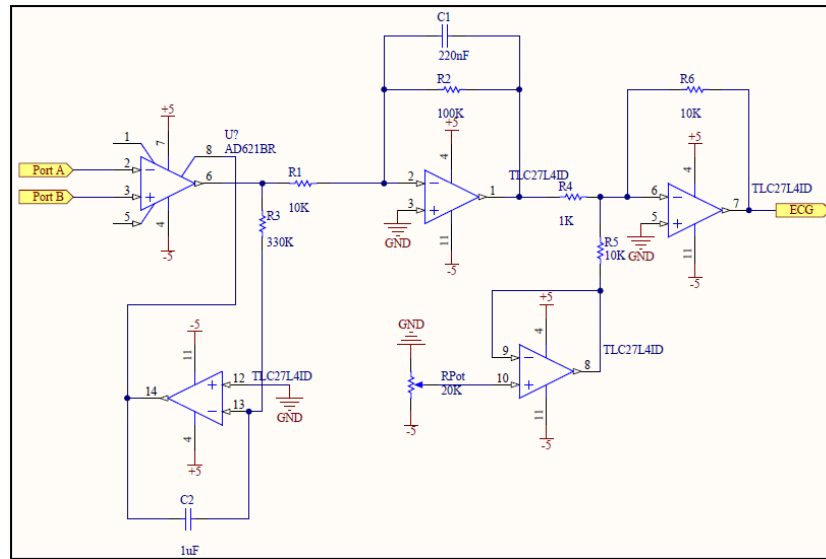


Figura 19. Circuito acondicionador de señal ECG

En el **Apéndice A** muestran estos mismos diagramas con más detalle.

8. CONVERSIÓN A/D Y ENVÍO DE DATOS

8.1. Selección del microprocesador (anfitrión)

Dado que en el laboratorio de electrónica del CCADET se cuenta con las herramientas adecuadas para el manejo de microcontroladores PIC de Microchip, software de programación, compiladores, software de simulación, dispositivos programadores, bibliografía etc. Se eligió usar el microcontrolador PIC de clase alta 18F2550, adicionalmente a lo anterior, este circuito nos proporciona ciertas características deseables para el diseño de nuestro sistema, algunas de las cuales son:

- Bajo costo
- Módulos varios
- Múltiples entradas analógicas y digitales
- Conversión analógica digital de mayor velocidad que los PIC's de clase media
- Comunicación serial, paralela, SPI
- Programación en lenguaje C

La figura 20 muestra los pines de conexión de este circuito integrado y la forma en la que se conectaron las señales provenientes de la etapa de acondicionamiento de cada sensor. En el **Apéndice C** se muestran las hojas de datos de los circuitos empleados en este diseño.

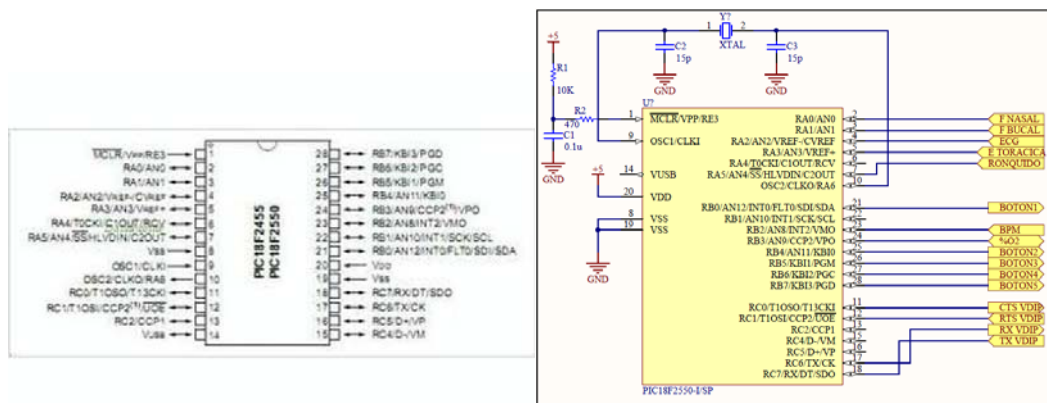


Figura 20. Pines de conexión del PIC18F2550

Todo el programa que se cargó al microcontrolador para realizar el envío de comandos, datos, conversión A/D y demás procedimientos algebraicos fue hecho en lenguaje C ya que este lenguaje nos permite realizar programas muy complejos de manera más práctica de lo que sería programar en lenguaje ensamblador, pero tenemos la

desventaja de perder velocidad en la ejecución de instrucciones. A pesar de esto fue más conveniente utilizar el lenguaje C ya que la frecuencia de la señales de interés es baja y no requerimos un programa demasiado optimizado.

8.2. Multiplexado y conversión A/D

Este microcontrolador cuenta con un módulo convertidor A/D con 13 entradas analógicas multiplexadas. En nuestro sistema solo se utilizaron 7 entradas para hacer pruebas, se utilizó multiplexado para poder obtener 7 señales digitales. En el sistema se puede aumentar la cantidad de señales, pero esto provocaría una disminución en la frecuencia de muestreo ya que el microcontrolador solo tiene un convertidor A/D.

El tiempo de conversión fue aproximadamente de 20 μ s más un tiempo extra de 10 μ s que proporcionamos por el multiplexado ya que se realizan cambios del canal a convertir. Al final de cada conversión se realiza tratamiento algebraico para tener los datos listos para el envío a la memoria flash USB.

La figura 21 muestra un listado del programa desarrollado en C para la conversión A/D y el multiplexado de señales.

```
.....
.....

setup_adc_ports(AN0_TO_AN9);           //C
setup_adc(ADC_CLOCK_DIV_16);           //
.....
set_adc_channel(canal);                 // C
delay_us(10);                           // R
q= read_adc();
p=( (5.0*q/1024)); //+1.564);           // M
EnviaDatoAD(p);                         // E
canal++;                                // C

.....
.....
```

Figura 21. Parte del listado del programa para la conversión A/D

9. ALMACENAMIENTO DE DATOS

El almacenamiento de datos en la memoria flash USB se realizó por medio del módulo VDIP1 del fabricante FTDI el cual contiene el circuito integrado VNC1L que proporciona la característica de anfitrión a un microcontrolador. La elección del microcontrolador como ya se mencionó fue con base en las señales que tenemos, que en nuestro caso son señales de baja frecuencia, a la disponibilidad del CI, entorno de programación, versatilidad del dispositivo y las propias necesidades que requiere el módulo VDIP1. La figura 22 muestra el modulo VDIP1.

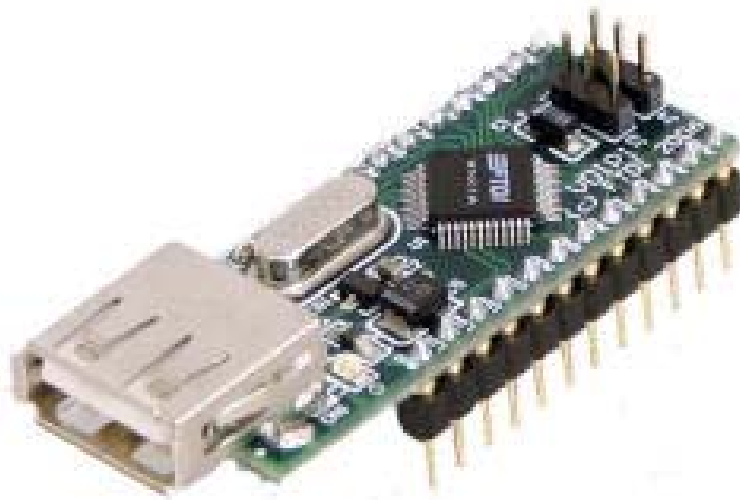


Figura 22. Modulo VDIP1

10. COMUNICACIÓN Y PROGRAMACIÓN DEL MÓDULO VDIP1

Para poder utilizar el módulo VDIP1 se necesita de un dispositivo controlador que en este caso será el PIC 18F2550, que envíe comandos y reciba órdenes o avisos. El dispositivo controlador debe poder comunicarse en cualquiera de los siguientes tipos de comunicación:

- Serial UART
- Paralelo FIFO
- SPI

La más conveniente para nuestro sistema por practicidad es la comunicación serial UART, ya que solo utiliza un canal de comunicación y podemos obtener buena velocidad de transmisión, alrededor de 1 Mbps.

La comunicación SPI es mas compleja de realizar, además que utiliza más líneas de comunicación y tiene velocidades parecidas a la de la comunicación serial. La comunicación paralela tiene la desventaja de usar muchas líneas de comunicación, y como nuestras señales no necesitan de una mayor velocidad, este tipo de comunicación esta sobrada.

Para una configuración inicial del módulo el fabricante proporciona un firmware que se encarga de programarlo para una cierta velocidad de transmisión, tipo de datos, control de flujo de datos, etc.

El manejo del módulo VDIP1 se realiza mandando comandos por medio de la comunicación elegida. Estos comandos son referentes de las siguientes instrucciones principales.

- Configuración de la velocidad de transmisión
- Tipo de datos ya sean ASCII o binarios
- Sincronía
- Búsqueda de dispositivos USB
- Creación de archivo
- Apertura de archivo
- Escritura en archivo
- Lectura de archivo
- Cierre de archivo

Cada comando da una instrucción al módulo y este la realiza si el comando es correcto o en su defecto manda un mensaje de error al microcontrolador si la instrucción esta mal escrita o no es posible utilizarla en ese momento (Vinculum, 2008).

De esta manera nos permitió crear un archivo dentro de la memoria flash USB para ir almacenando los datos provenientes del módulo A/D. La figura 23 muestra el programa realizado en C para la comunica entre el PIC y el módulo VDIP1.

```

if (VNC1L_EncontrarDisco() == D                                // Ch
{
    serial_enviarbyte('S');                                     // E
    serial_enviarbyte('B');                                     // E
    serial_enviarbyte('D');                                     // E
    serial_enviarbyte(' ');                                     // E
    serial_enviarbyte(0x03);                                     // E
    serial_enviarbyte(0x00);                                     // E
    serial_enviarbyte(0x00);                                     // E
    serial_enviarbyte(0x0D);                                     // E
    delay_ms(1000);

    set_timer1(0x0000);                                         // R

    serial_resetrxfifo();

    VNC1L_CerrarArchivo();

        serial_enviarbyte('S');                                 // E
        serial_enviarbyte('B');                                 // E
        serial_enviarbyte('D');                                 // E
        serial_enviarbyte(' ');                                 // E
        serial_enviarbyte(0x34);                                 // E
        serial_enviarbyte(0xC0);                                 // E
        serial_enviarbyte(0x00);                                 // E
        serial_enviarbyte(0x0D);                                 // E

        delay_ms(100 );
        set_uart_speed(576);                                     //R

}

while(1)
{
    if (VNC1L_Encontrar () != DiscoObtenido)                   // Ch
    {
        break;                                                  // A
    }
}

```

Figura 23. Parte del programa que controla la comunicación y el almacenamiento de datos en la USB

11. VISUALIZACIÓN DE DATOS

Un sistema de adquisición de datos debe contar con una interfaz que permita visualizar los datos obtenidos, ya que esto permitirá corroborarlos, analizarlos y usarlos para los fines deseados.

Para el despliegue de la información almacenada en la memoria USB, se desarrollo un programa en ambiente LabView, esto debido a que es un ambiente que nos genera interfaces amigables con el usuario de tal forma que resulta muy sencillo operarlas por cualquier persona incluso si no se tienen muchos conocimientos de computación, éste programa generado nos permiten visualizar los datos de manera rápida y eficaz. La figura 24 muestra parte del código generado en LabView para poder visualizar los datos en una PC.

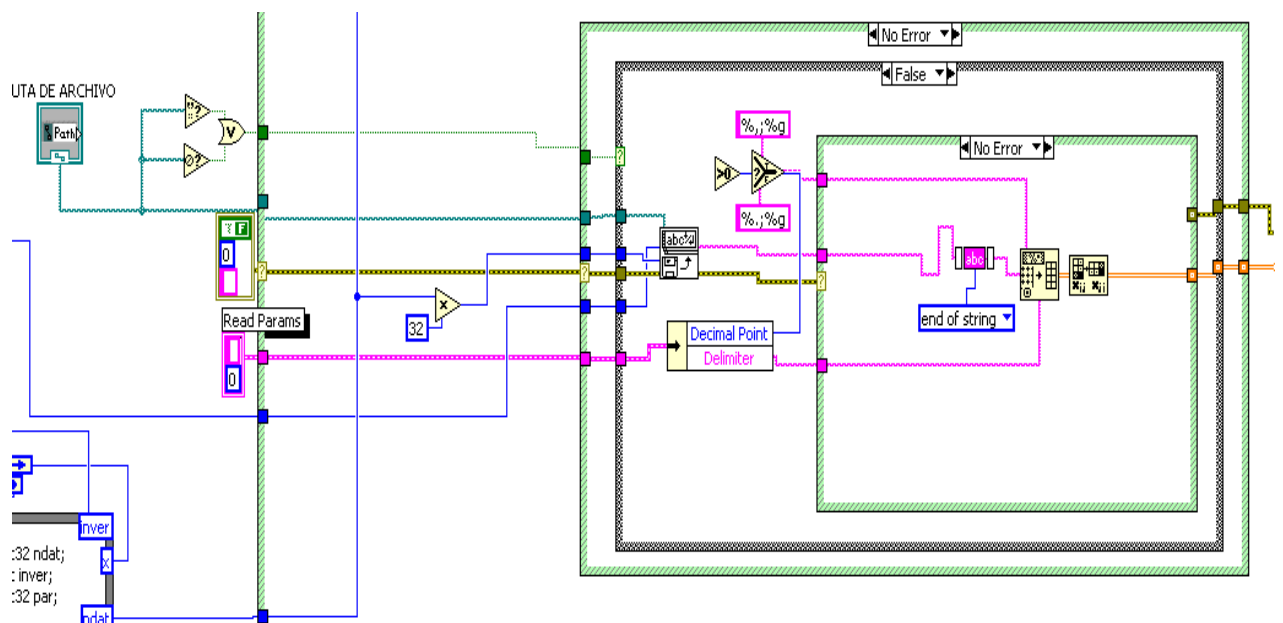


Figura 24. Programa generado en LabView para visualización de datos almacenados en la memoria USB

En el **Apéndice B** se muestran los listados de los programas de la interfaz de usuario así como los generados en C para realizar la comunicación entre el PIC y el modulo VDIP1, además de los necesarios para realizar la conversión A/D.

12. RESULTADOS

La prueba de captura se realizó con 6 señales, la frecuencia de cada una de estas fue entre 0 y 3Hz, las señales fueron acondicionadas para poderlas introducir a los canales analógicos del PIC y posteriormente convertirlas a señales digitales.

Con los comandos necesarios se mandaron los datos uno por uno al módulo para poder así almacenarlos en la memoria flash USB, vale la pena mencionar que las 6 señales utilizaron solo un canal de conversión A/D así como solo un canal para el almacenamiento en la memoria USB, esto nos permitió tener un sistema de bajo costo, ya que no se necesitaron módulos externos CAD ni otros dispositivos además del microcontrolador y el módulo VDIP1, pero limitó la velocidad de almacenamiento lo que en este caso no es crítico.

La comunicación entre el microcontrolador y el módulo VDIP1 se realizó a 1 Mbps, pero la velocidad máxima que se alcanzó para el almacenamiento de datos fue de 1449 datos/segundo dividido en las 6 señales de prueba, en una de estas señales vienen enlazados los datos de %O₂ y BPM, por lo que en total son 7 señales. Las razones principales de que esta velocidad sea mucho más baja que la alcanzada en la transmisión de datos fue que:

- Cada dato es de 32 bits
- Se necesita tiempo de adquisición y conversión en el módulo A/D
- Tratamiento algebraico de los datos
- Tiempo de ejecución de las instrucciones en lenguaje C
- Multiplexado en tiempo

A pesar de estos inconvenientes la velocidad alcanzada es adecuada para las señales adquiridas, ya que como hemos mencionado antes son de baja frecuencia.

Dependiendo de la resolución que se desee dar y la frecuencia de la señal, nosotros podemos aumentar o disminuir la cantidad de canales disponibles, así como utilizar microcontroladores más potentes o dispositivos externos que nos permitan aumentar la velocidad de los procesos de conversión y transmisión de datos.

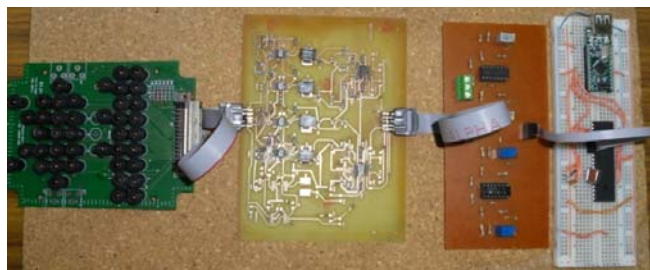


Figura 25. Sistema prototipo de adquisición de datos montado en un PCB

La figura 25 muestra una vista general de un primer prototipo del sistema montado en circuito impreso, la figura 26 muestra un oscilograma de la señal que entra al sistema proveniente del paciente y la figura 27 muestra una pantalla de recuperación de datos mediante el uso de una PC y empleado el programa Excel a manera de prueba.

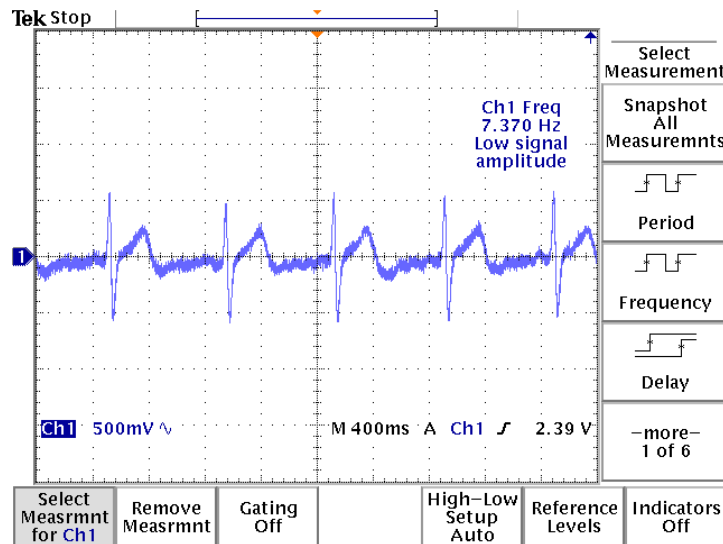


Figure 26. Señal de prueba proveniente del paciente

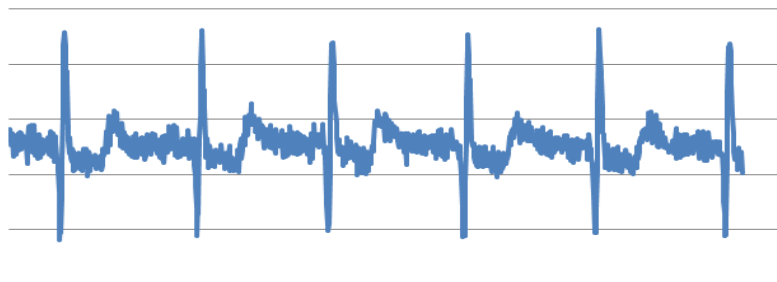


Figure 27. Señal de prueba de la frecuencia cardiaca obtenida de los datos almacenados y vistos en Exel.

12.1. Interfaz de usuario

Las figuras 28 y 29 muestran las imágenes que los usuarios ven en la PC en el momento que abren el programa diseñado finalmente en LabView, este programa es empleado para visualizar los datos almacenados en la memoria USB. Fue necesario desarrollar un programa a la medida de los requerimientos, dado que de esta forma se le presenta al especialista toda la información requerida de manera más amigable.

La pantalla puede mostrar 7 variables:

- Expansión Torácica
- Flujo Bucal
- Flujo Nasal
- Ronquido
- Señal ECG
- Porcentaje de oxígeno en sangre
- Frecuencia cardiaca

El porcentaje de oxígeno y la frecuencia cardiaca se muestran como valor numérico, adicionalmente se presentan sus gráficas como el resto de las señales. El programa tiene la opción de poder seleccionar la cantidad de datos que se ven en la pantalla, por lo que se pueden seleccionar 1, 5, 10 y 30 segundos y 1, 3 y 5 minutos de tal forma que el especialista lo ajuste al valor que más le acomode.

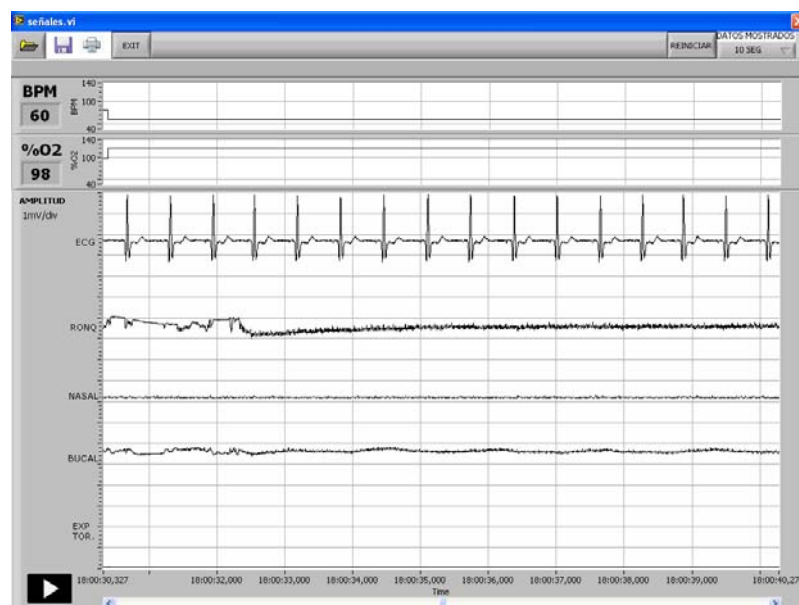


Figura 28 Pantalla de la interfaz de usuario para lectura de datos

El programa también cuenta con otras teclas de control de tal forma que se puede seleccionar el archivo que se quiere ver en ese momento o también se tiene la opción de almacenar la imagen actual en un archivo. Por otro lado, al momento de estar analizando los datos se tiene la opción de poder desplazarse en el tiempo, a cualquier punto de las señales grabadas (se puede adelantar o atrasar) mediante una barra de desplazamiento. Otras opciones adicionales son el que se cuenta con un botón de pausa para congelar la imagen y un botón de inicio para empezar nuevamente la lectura desde cero.

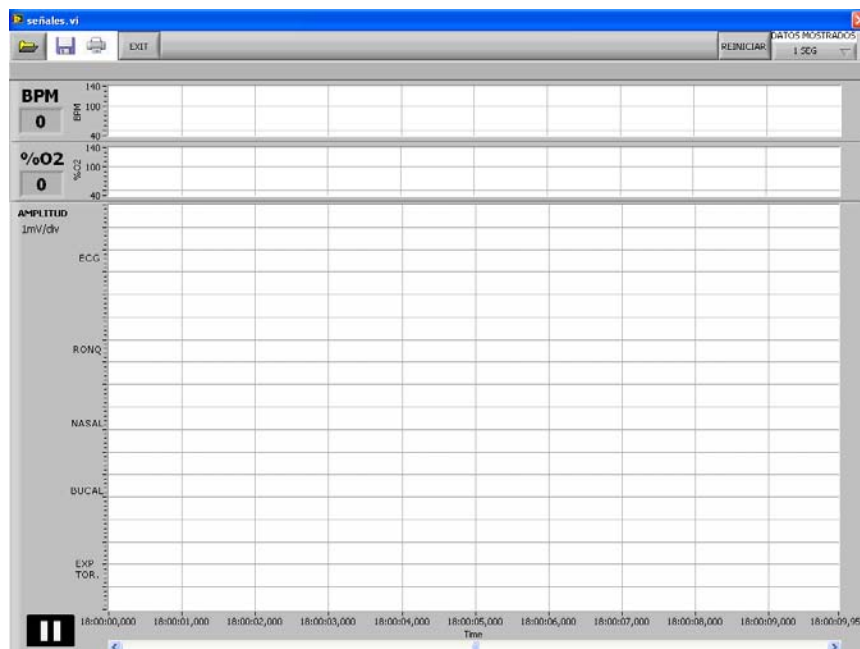


Figura 29. Interfaz de usuario mostrando los botones de control.

12.2. Prototipo logrado

Dados los adelantos que se han tenido en la adquisición de componentes electrónicos cada vez mas pequeños,

El circuito impreso final para este prototipo en particular, se realizó completamente en montaje superficial, integrando todas las etapas mostradas en la figura 25 básicamente en una sola tarjeta PCB. Con este hecho se podrá introducir todo el sistema en una caja de dimensiones pequeñas para hacerlo lo mas portátil posible.

La figura 30 muestra el esquemático completo y la figura 31 muestra una imagen del PCB logrado para este prototipo.

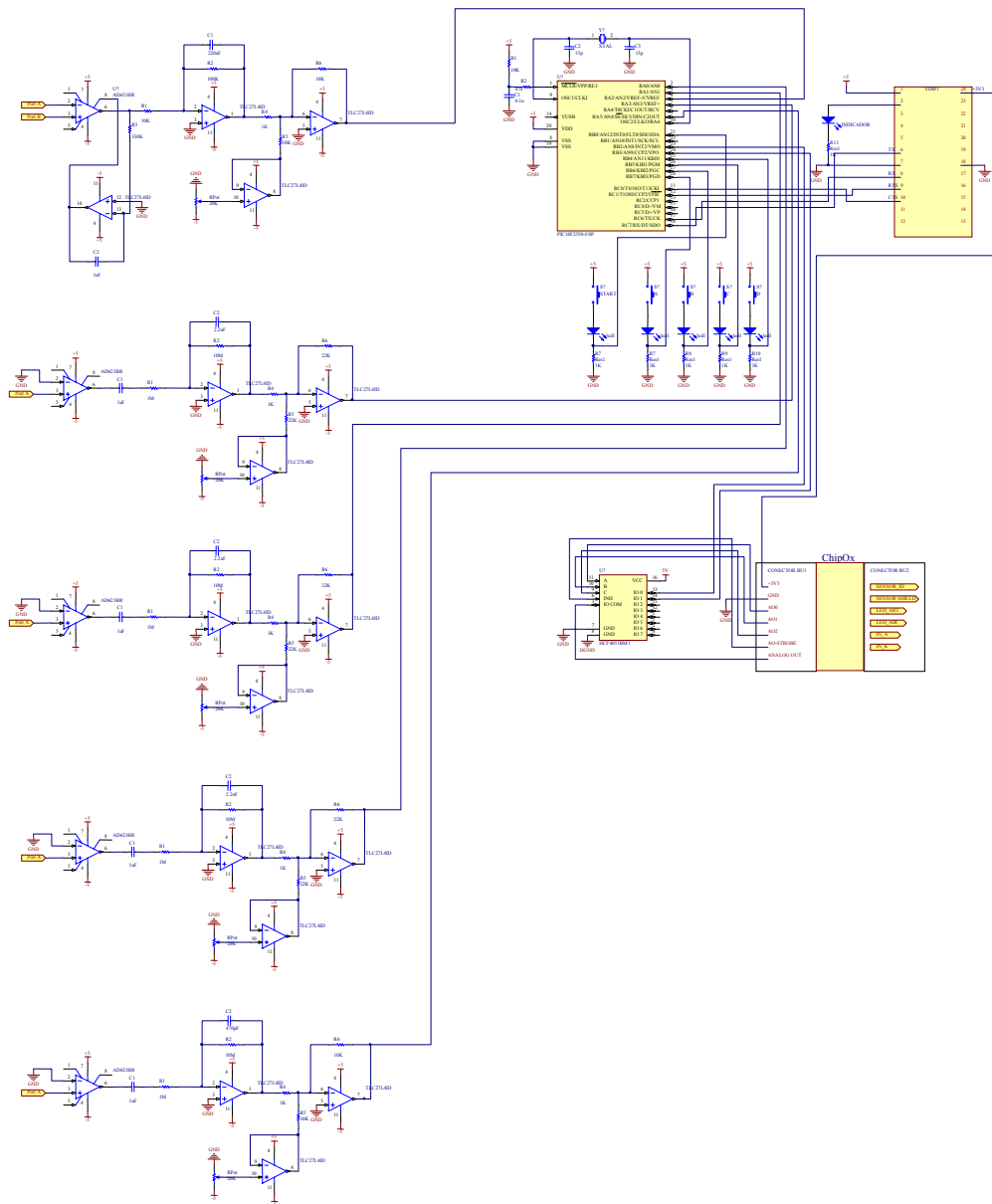


Figura 30. Esquemático completo del sistema

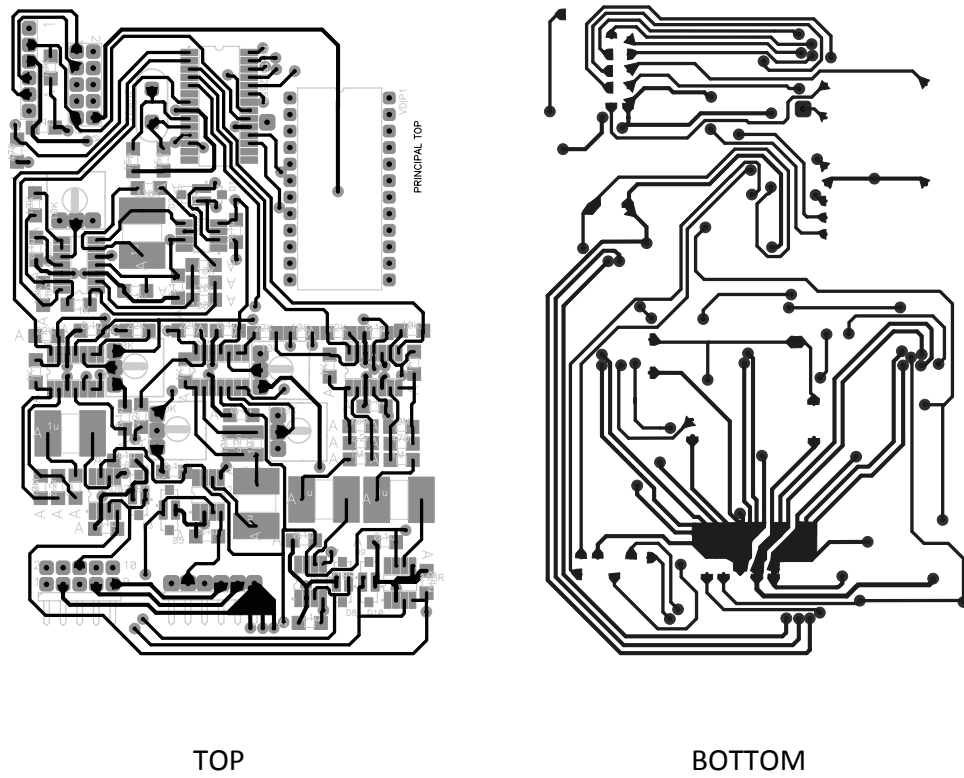


Figura 31 PCB (placa principal) del sistema en montaje superficial

APÉNDICE A

Diagramas Y Circuitos Impresos

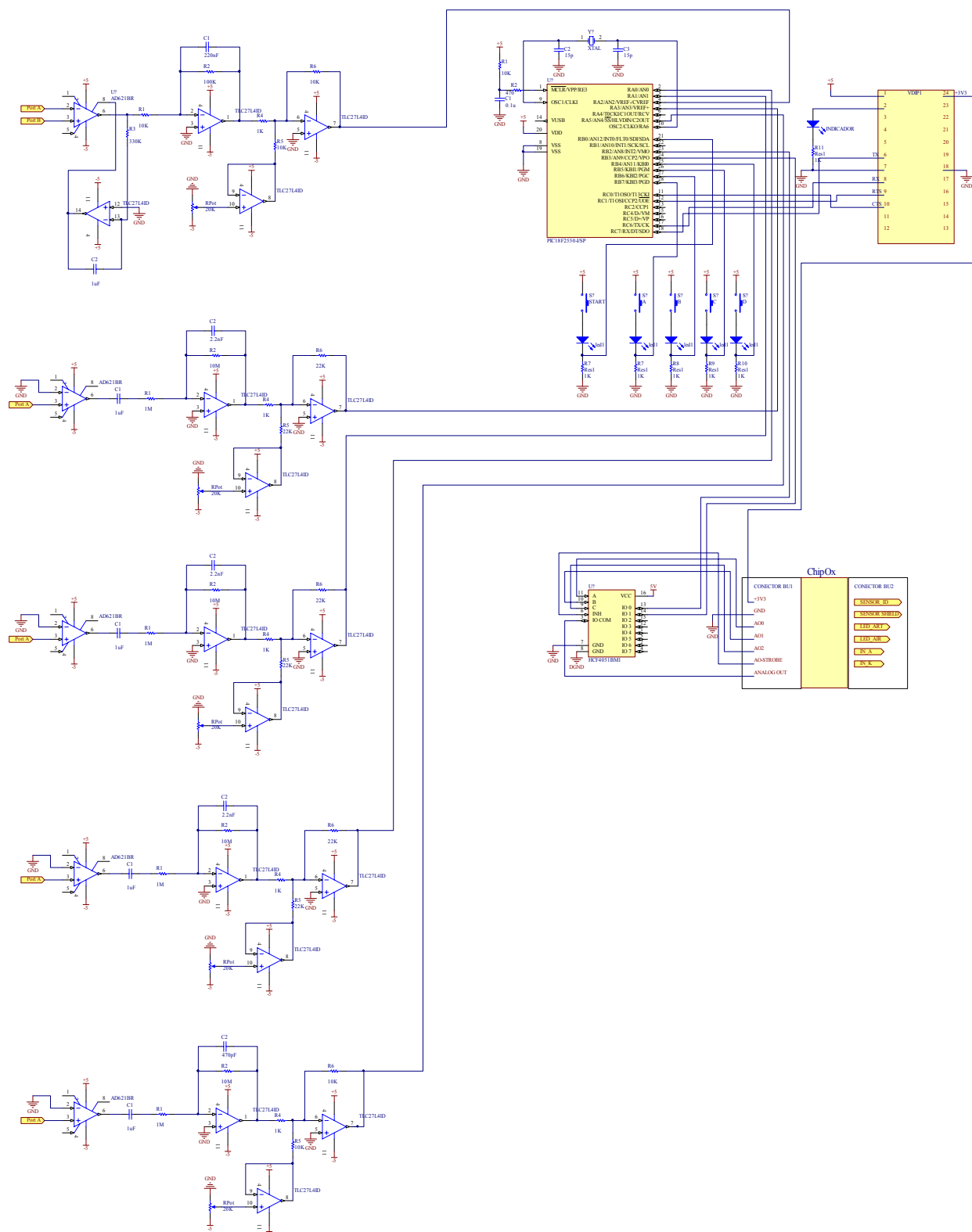
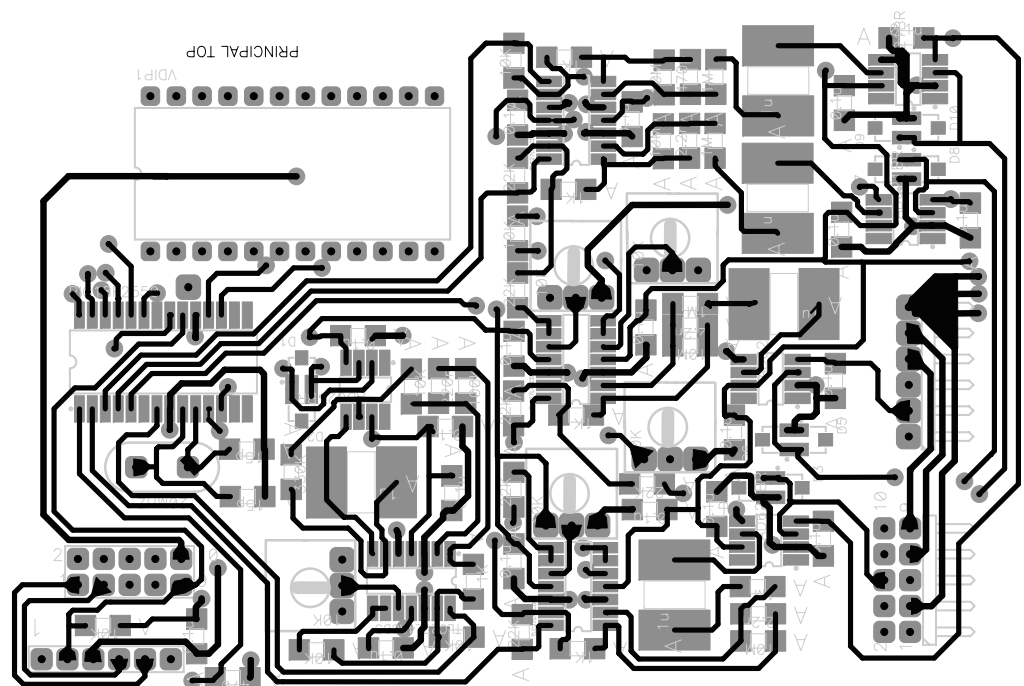
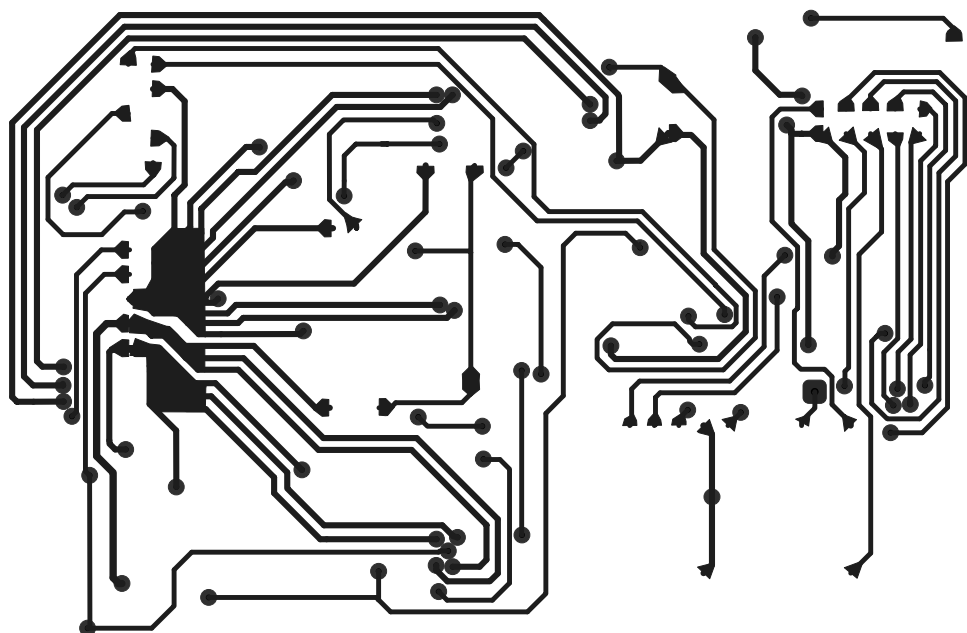


Figura 32. Esquema general del sistema

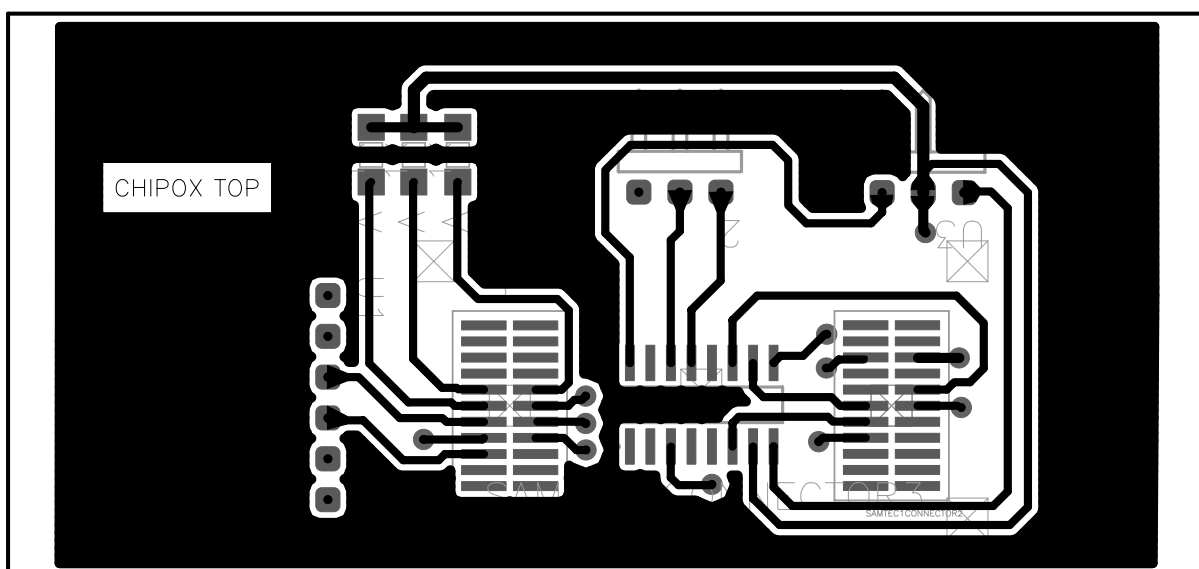


TOP



BOTTOM

Figura 33.PCB de placa principal



TOP

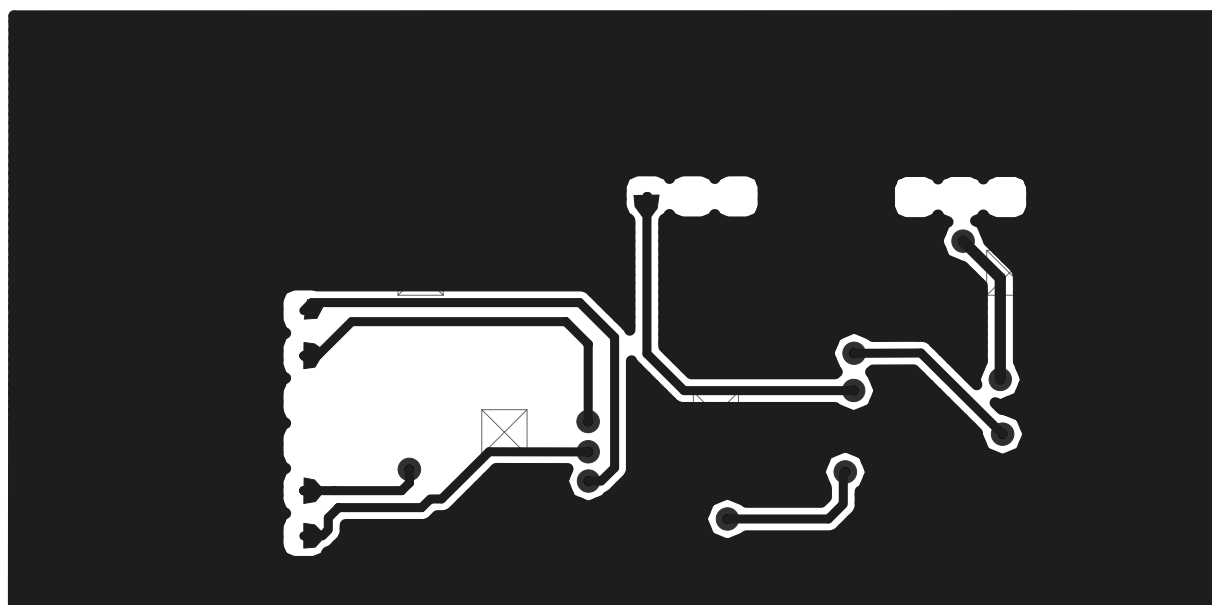
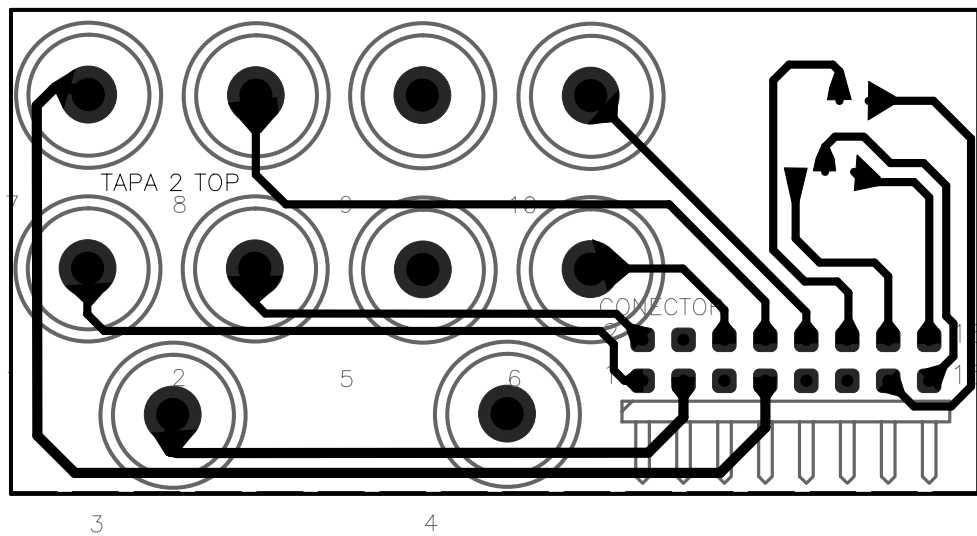
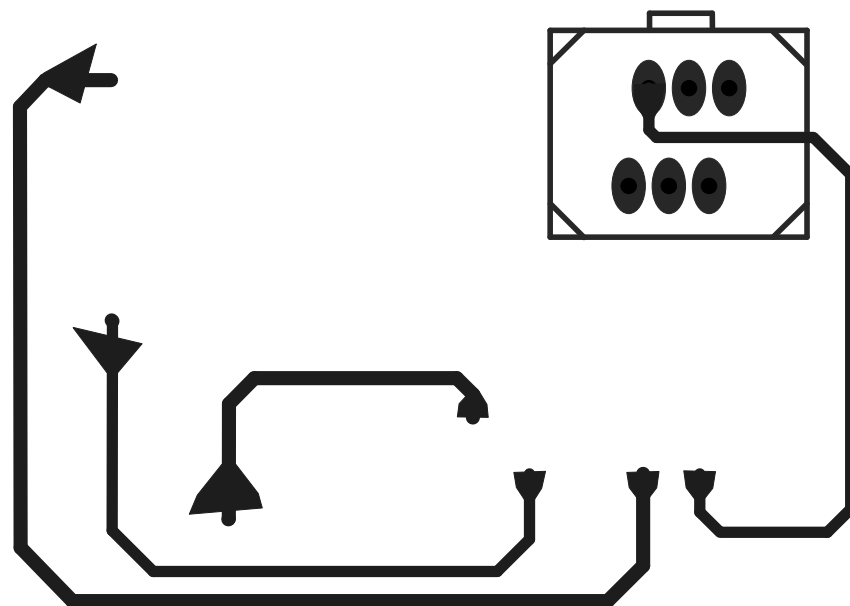


Figura 34. PCB para ChipOx

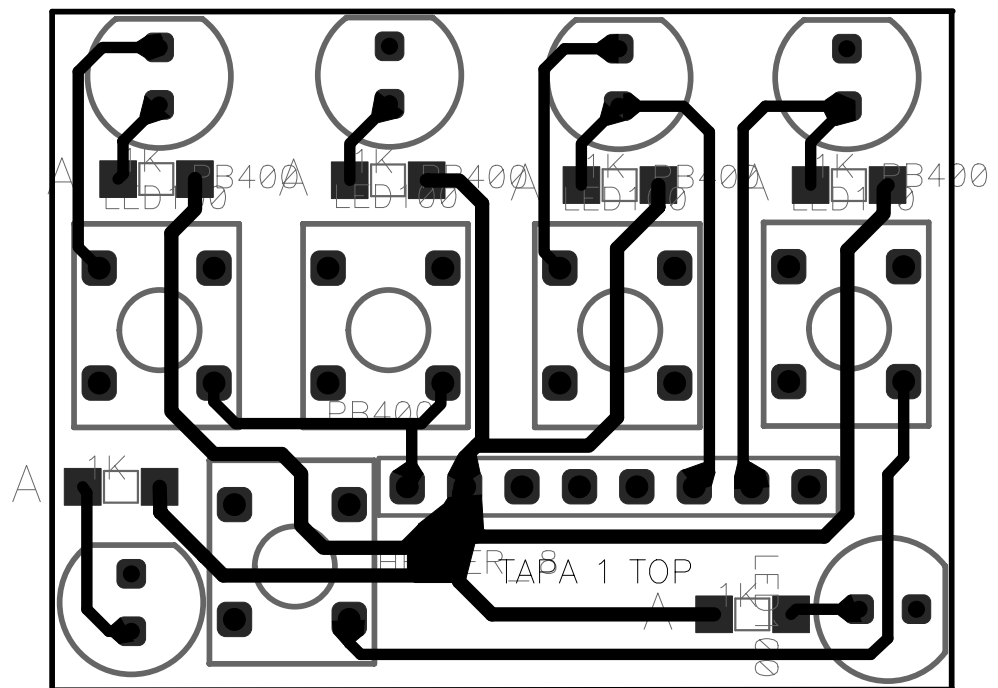


TOP

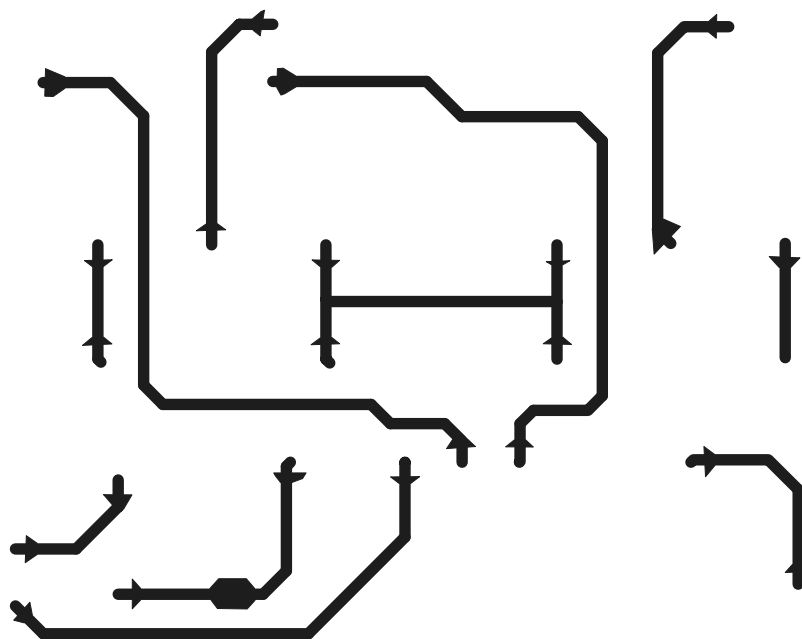


BOTTOM

Figura 35. PCB para conectores



TOP



BOTTOM

Figura 36. PCB para botones

APÉNDICE B

Programas

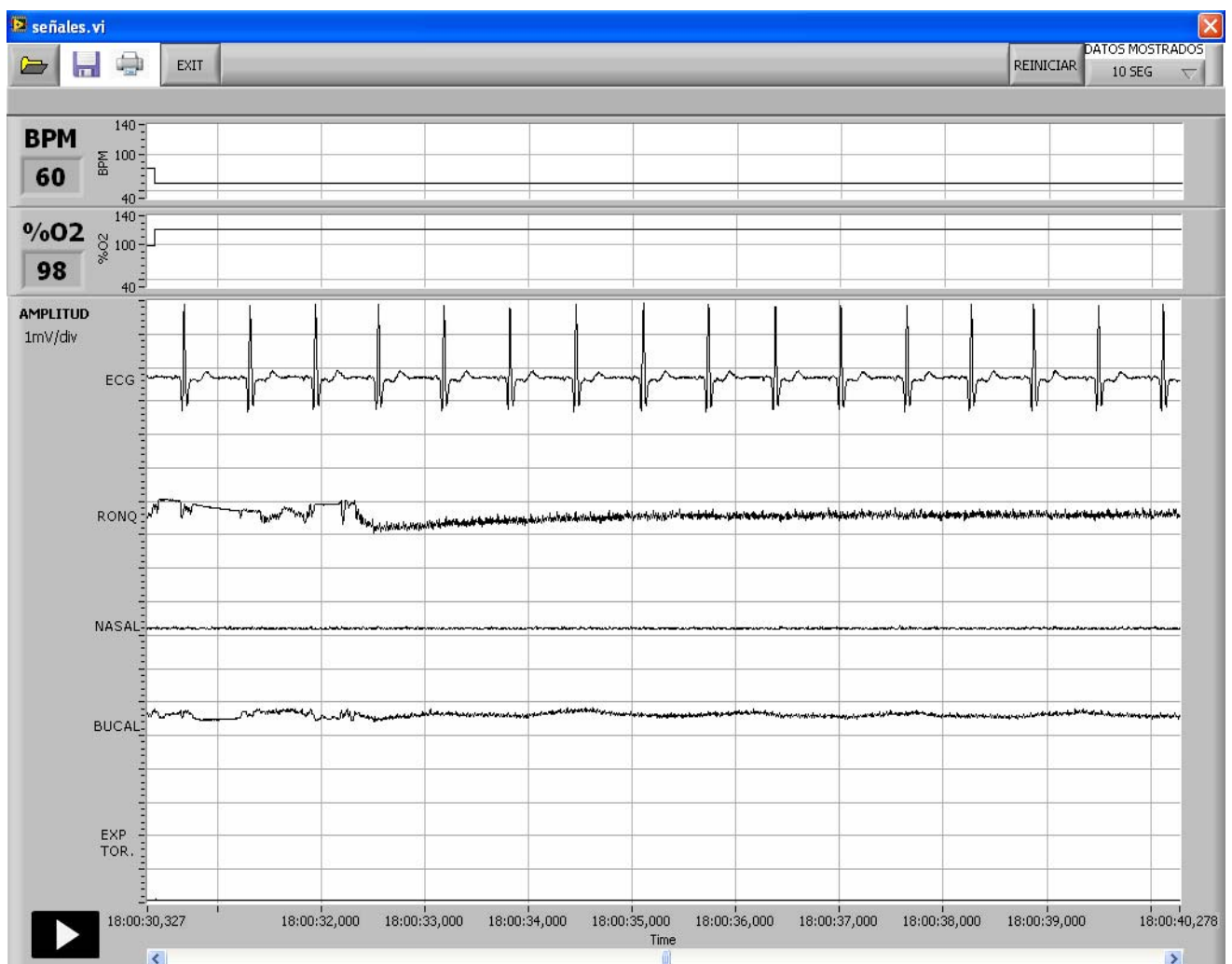


Figura 37. INTERFAZ DE USUARIO MOSTRANDO SEÑALES PREVIAMENTE GRABADAS

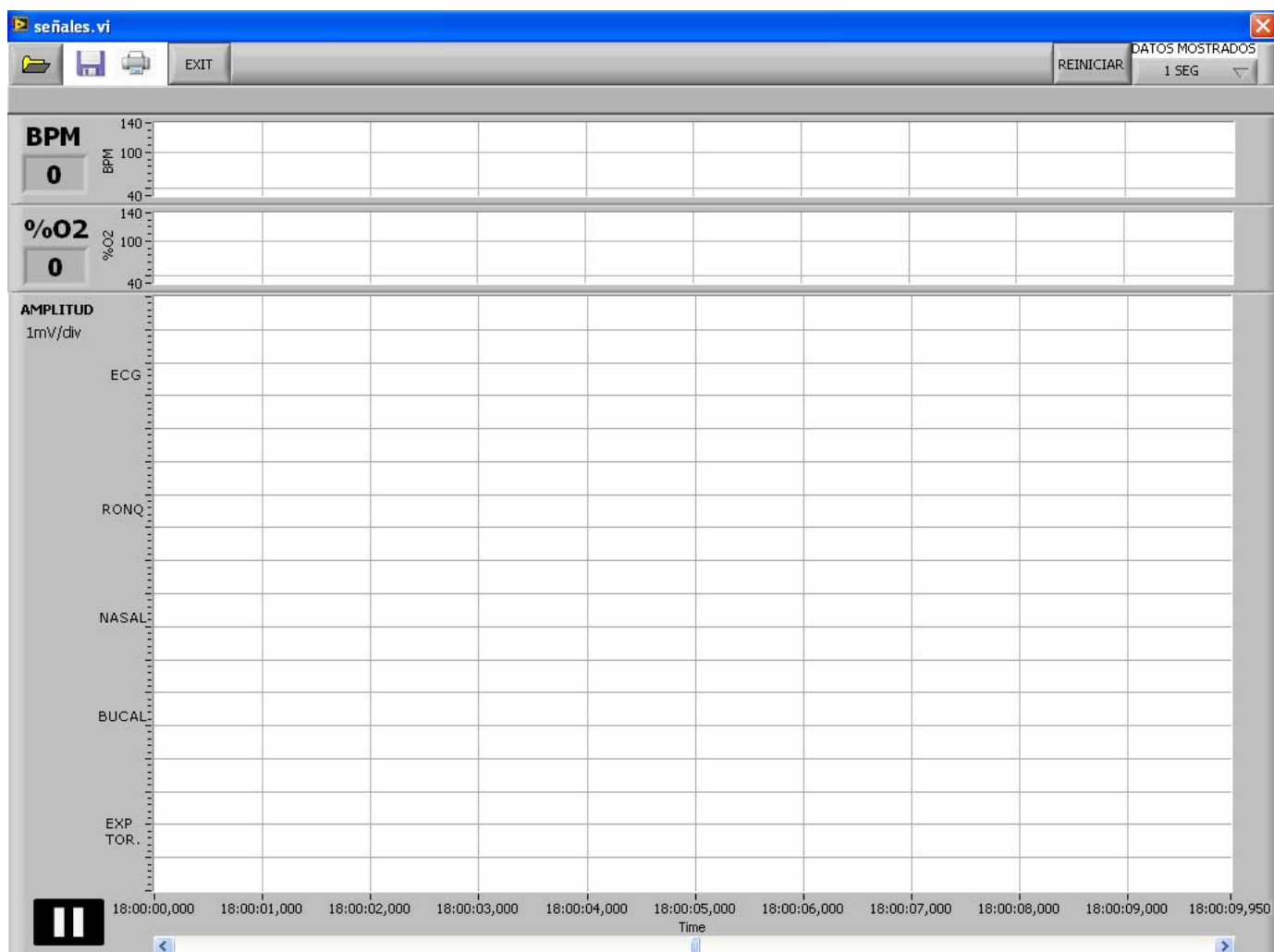


Figura 38. BOTONES DE CONTROL EN LA INTERFAZ

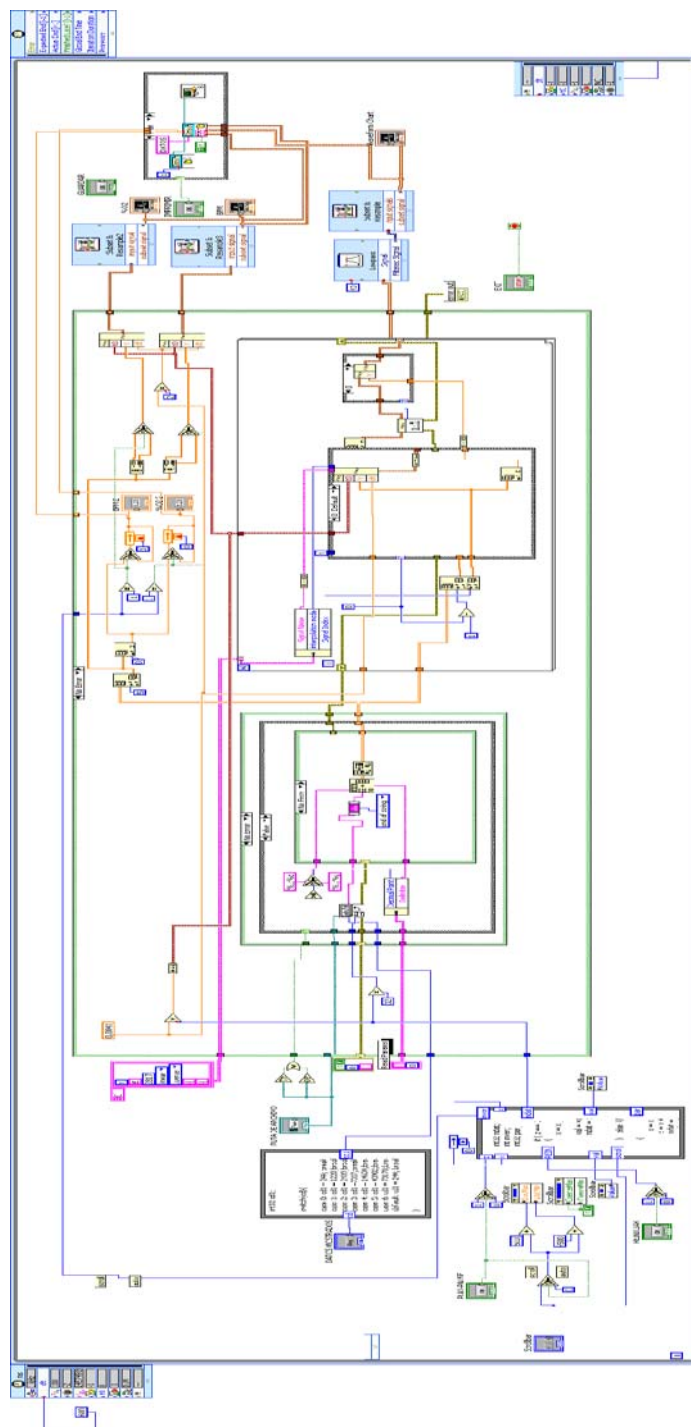


Figura 39. PROGRAMA COMPLETO EN LABVIEW

PROGRAMA EN C PARA ALMACENAMIENTO EN MEMORIA USB

```
#include <18f2550.h>
#device adc=10
#fuses nowdt,hs,cpudiv1,nowrt
#use standard_io(b)
#use delay(clock = 20000000)
#use rs232(xmit = pin_c6, rcv = pin_c7, bits= 8, parity= n)
#use fast_io(c)
#BYTE portc = 0xF82
#BYTE trisc = 0xF94
#BYTE baudcon = 0xFB8
#BYTE txreg = 0xFAD
#BYTE rcreg = 0xFAE
#BYTE spbrgh = 0xFB0
#BYTE spbrg = 0xFAF
#BIT cren = 0xFAB.4
#BIT trmt = 0xFAC.1
#BIT txen = 0xFAC.5
#BIT rcif= 0xF9E.5
#BIT brgh = 0xFAC.2

//*****
//Constantes
//*****
char _RTS = 0x01; // RC0
char _CTS = 0x02; // RC1

// Variables buffer del puerto serial
char SERIAL_RX_FIFO[0x20]; // Asigna
char RX_FIFO_SIZE = sizeof(SERIAL_RX_FIFO); // Inicializa
char RX_FIFO_COUNT = 0x00; // Restablece
char * RX_FIFO_HEAD_PTR = &SERIAL_RX_FIFO[0x00]; // Inicializa
char * RX_FIFO_TAIL_PTR = &SERIAL_RX_FIFO[0x00]; // Inicializa
char * RX_FIFO_START = RX_FIFO_TAIL_PTR; // Establece
char * RX_FIFO_END = RX_FIFO_START + RX_FIFO_SIZE; // Establece
char RX_STATUS = 0x00; // Restablece
char RX_FIFO_OVERRUN = 0x01;
char RX_FIFO_UPPER_LIMIT = 0x03 *(RX_FIFO_SIZE / 0x04); // Limite
char RX_FIFO_LOWER_LIMIT = (RX_FIFO_SIZE / 0x04); // Limite

//*****
// Variables
//*****
int d;
int control;
int a;
int x=0;
int y=1;

int16 q;

long ciclo;
long n=1;
long m=1;

float i=0;
float p;
float val;
float time = 0;

//*****
```

```

// Define
//*****

enum {
    VNC1L_Idle,
    VNC1L_DiscoDisponible,
    VNC1L_CrearArchivo,
    VNC1L_EscribirArchivo,
    VNC1L_FinalizarArchivo,
    VNC1L_EsperaParaEliminar
};

char VNC1L_Estado = VNC1L_Idle;

// Inicializa

//*****
// Constantes
//*****
char Sincronizado = 001;
char DiscoObtenido = 001;

//*****
// Rutinas
//
// *****
//*****

//*****
//
// Restablece Rx FIFO
//
//*****

void serial_resetrxfifo()
{
    disable_interrupts(INT_RDA);
    disable_interrupts(GLOBAL);

    RX_FIFO_COUNT = 0x00;
    RX_STATUS = 0x00;
    RX_FIFO_HEAD_PTR = (&SERIAL_RX_FIFO[0x00]);
    RX_FIFO_TAIL_PTR = (&SERIAL_RX_FIFO[0x00]);

    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    portc &= ~_RTS;

}

//*****
//
// Inicializa
//
//*****
void serial_init(long BaudRate)
{
    disable_interrupts(INT_RDA);
    disable_interrupts(GLOBAL);

    BAUDCON = 0x00;

```

```

    TRISC = 0xBE;                                // Bits
    portc |= _RTS;                                // Establece
    set_uart_speed(5760);                          // Establece
    portc &= ~_RTS;                                // Establecer

    serial_resetrxfifo();                          // Restablece

    clear_interrupt(int_rda);
    enable_interrupts(INT_RDA);                    // Habilita
    enable_interrupts(GLOBAL);
    CREN = 001;                                    // Habilita
}

//*****
//
// Enviar un byte
//
//*****
void serial_enviarbyte(char TX_BYTE)
{
    while (portc & _CTS);                          // Espera

    while (!TRMT);                                  // Espera
    TXREG = TX_BYTE;                                // Mover byte
    TXEN = 1;                                       // Comenzar
    while (TRMT);                                  // Espera
}

//*****
//
// Agrega
//
//*****
void serial_agregararxfifo()
{
    if (RX_FIFO_COUNT >= RX_FIFO_)                  // esta lleno
    {
        RX_STATUS = RX_STATUS | RX_FIFO_OVERRUN;    // Establecer

        return;
    }

    if (RX_FIFO_COUNT >= RX_FIFO_UPPER_LIMIT)        // Mantener
                                                    // bytes
    {
        portc |= RTS;                                // Establecer
    }

    disable_interrupts(GLOBAL);                      // Deshabilitar

    *RX_FIFO_HEAD_PTR = RCREG;                       // Agregar
                                                    // puntero
    RX_FIFO_COUNT;                                    // Incrementa

    if (RX_FIFO_HEAD_PTR >= RX_FIFO_END)             // A
                                                    // dirección
    {
        RX_FIFO_HEAD_PTR = RX_FIFO_START;           // circular
    }
}

```

```

        if (RX_FIFO_COUNT < RX_FIFO_UPPER_LIMIT)                // lleno
        {
            portc &= ~_RTS;                                       // Establecer
        }

        enable_interrupts(GLOBAL);                                // Habilitar
    }

    /*******
    //
    // Lee
    /*******
void serial_leerderxfifo(char * RX_FIFO_BYTE)
{
    if (RX_FIFO_COUNT 0)                                         // vacío
    {
        return;                                                  // Salir
    }

    disable_interrupts(GLOBAL);                                   // Deshabilita

    *RX_FIFO_BYTE = *RX_FIFO_PTR++;                             // Lee
    RX_FIFO_COUNT--;                                             // incrementa
                                                                // Decrementa

    if (RX_FIFO_TAIL_PTR >= RX_FIFO_END)                         // puntero
                                                                // final
    {
        RX_FIFO_TAIL_PTR = RX_FIFO_START;                       // circular
    }

    enable_interrupts(G);

    if (RX_FIFO_COUNT < RX_FIFO_LOWER_LIMIT)
    {
        portc &= ~_RTS;                                         // Establecer
    }
}

////////////////////////////////////
////////////////////////////////////
//
// Comandos
//
////////////////////////////////////
////////////////////////////////////

/*******
//
// Comando E
//
/*******
void VNC1L_Big_E()
{
    serial_enviarbyte("");
    serial_enviarbyte(00D);                                     // Envía
                                                                // Envía
}

/*******
//
// Comando e
//
/*******
void VNC1L_Small_e()

```

```

{
    serial_enviarbyte('e');
    serial_enviarbyte(00D);
}

//*****
//
// Sincronizar
//
//*****
char VNC1L_Sinc()
{
    char LoopCount = 000;
    char FIFOByte1 = 000;
    char FIFOByte2 = 000;
    char Got_Big_E = 000;
    char Got_Small_e = 000;

    serial_resetrxfifo(); // Reset

    while ((Got_Big_E == 0x00) && (LoopCount < 0xFF))
    {
        VNC1L_Big_E(); // Envía
        LoopCount++; // Incrementa
        while (RX_FIFO_COUNT < 0x02)
        {
            delay_ms(10); // Espera
        }

        if (RX_FIFO_COUNT > 0x00) // datos
        {
            while (RX_FIFO_COUNT > 0x02) // Leer
            { // últimos
                serial_leerderxfifo(&FIFOByte1); // leer
                // e

            }
            serial_leerderxfifo(&FIFOByte1); // Checar
            // 00D
            serial_leerderxfifo(&FIFOByte2);
            if ((FIFOByte1 == 'E') && (FIFOByte2 == 0x0D))
            {
                Got_Big_E = 0x01;
            }
            else
            {
                delay_ms(10); // Espera
            }
        }
    }

    if (Got_Big_E == 0x01)
    {
        VNC1L_Small_e();
        while (RX_FIFO_COUNT < 0x02)
        {
            delay_ms(1); // Espera
        }
        serial_leerderxfifo(&FIFOByte1); // Checar
        serial_leerderxfifo(&FIFOByte2);
        if ((FIFOByte1 == 'e') && (FIFOByte2 == 0x0D))
        {
            Got_Small_e = 0x01; // encuentra
        }
    }
}

```

```

    }

    return Got_Small_e;
}

//*****
//
// Buscar
//
//*****
char VNC1L_Encontrar () // Regresa
                        //regresa
{
    char FIFOByte1 = 0x00;
    char FIFOByte2 = 0x00;
    char FIFOByte3 = 0x00;
    char FIFOByte4 = 0x00;
    char FIFOByte5 = 0x00;
    serial_resetrxfifo();

    serial_enviarbyte(0x0D); // Envía

    while (RX_FIFO_COUNT < 0x05); // Esperar
    serial_leerderxfifo(&FIFOByte1); // Leer
    serial_leerderxfifo(&FIFOByte2);
    serial_leerderxfifo(&FIFOByte3);
    serial_leerderxfifo(&FIFOByte4);
    serial_leerderxfifo(&FIFOByte5);

    if ((FIFOByte1 == 'D') && (FIFOByte2 == ':') && (FIFOByte3 == 0x5C) && (FIFOByte4 == '>') && (FIFOByte5 ==
0x0D))
    {
        return 0x01; // regresa
    }
    else
    {
        while (RX_FIFO_COUNT > 0x00)
        {
            serial_leerderxfifo(&FIFOByte1); // Lee
        }

        return 0x00; // Regresa
    }
}

//*****
//
// Comando abrir
//
//*****
void VNC1L_AbrirArchivo()
{
    char FIFOByte = 0x00;

    serial_enviarbyte('O'); // E
    serial_enviarbyte('P'); // E
    serial_enviarbyte('W'); // E
    serial_enviarbyte(' '); // E
    serial_enviarbyte('d'); // E
    serial_enviarbyte('a'); // E
    serial_enviarbyte('t'); // E
    serial_enviarbyte('o'); // E
    serial_enviarbyte('s'); // E
    serial_enviarbyte('.'); // E
    serial_enviarbyte('t'); // E

```

```

        serial_enviarbyte('x');
        serial_enviarbyte('t');
        serial_enviarbyte(0x0D);
        while (RX_FIFO_COUNT < 0x05);

        while (RX_FIFO_COUNT > 0x00)
        {
            serial_leerderxfifo(&FIFOByte);
        }
    }

    /*******
    // Separa
    /*******

void EnviaDatoAD(float z)
{
    a = (int)z;
    serial_enviarbyte(a+48);

    z = (z - a)*10;
    a = (int)z;
    serial_enviarbyte(a+48);

    z = (z-a)*10;
    a = (int)z;
    serial_enviarbyte(a+48);

    z = (z-a)*10;
    a = (int)z;
    serial_enviarbyte(a+48);
}

    /*******
    //Configuración
    /*******
void DuracionEscritura(int d)
{
    switch (d)
    {
    case 1:

        serial_enviarbyte("W");
        serial_enviarbyte("R");
        serial_enviarbyte("F");
        serial_enviarbyte(' ');
        serial_enviarbyte(0x00);
        serial_enviarbyte(0x08);
        serial_enviarbyte(0x3D);
        serial_enviarbyte(0x5A);
        serial_enviarbyte(0x0D);
        break;

    case 2:

        serial_enviarbyte("W");
        serial_enviarbyte("R");
        serial_enviarbyte("F");
        serial_enviarbyte(' ');
        serial_enviarbyte(0x00);
        serial_enviarbyte(0x10);
        serial_enviarbyte(0x7A);
        serial_enviarbyte(0xB4);
        serial_enviarbyte(0x0D);
        break;
    }
}

```

```

case 3:
    serial_enviarbyte('W');           // E
    serial_enviarbyte('R');           // E
    serial_enviarbyte('F');           // E
    serial_enviarbyte(' ');
    serial_enviarbyte(0x00);           // E
    serial_enviarbyte(0x18);           // E
    serial_enviarbyte(0xB8);           // E
    serial_enviarbyte(0x0E);           // E
    serial_enviarbyte(0x0D);           // E
    break;
case 4:
    serial_enviarbyte('W');           // E
    serial_enviarbyte('R');           // E
    serial_enviarbyte('F');           // E
    serial_enviarbyte(' ');
    serial_enviarbyte(0x00);           // E
    serial_enviarbyte(0x20);           // E
    serial_enviarbyte(0xF5);           // E
    serial_enviarbyte(0x68);           // E
    serial_enviarbyte(0x0D);           // E
    break;
}

}

//*****
//
// Comando
//
//*****
void VNC1L_Escribira()
{
    char FIFOByte = 0x00;

    DuracionEscritura(d);
    while(m<ciclo)
    {
        n=1;
        while(n<12858)                // B
        {
            if(canal==5)
            {
                if(y==1)
                {
                    canal =8;
                    y=2;
                }
                else
                {
                    canal=9;
                    y=1;
                }
            }
        }

        set_adc_channel(canal);        // Conversión
        delay_us(10);
        q= read_adc();
        p=( (5.0*q/1024));              // Manipulación
        EnviaDatoAD(p);
    }
}

```

```

n++;
canal++;                                // Cambio

if(x>8)                                  // Espera
{

    canal=0;

    serial_enviarbyte(0x20);              // Envía
    serial_enviarbyte(0x20);              // Envía

    val = get_timer1();                   // Obtiene
    val = val/10;
    time = time + (val*0.0000008);
    EnviaDatoAD(time);
    set_timer1(0x0000);                   // Restablece

    serial_enviarbyte(0x0D);               // Envía
    serial_enviarbyte(0x0A);               // Salto
}
else
{

    serial_enviarbyte(0x20);               //Envía
    serial_enviarbyte(0x20);               //Envía

}

}

m++;
}
    while (RX_FIFO_COUNT < 0x05);          // Espera

    while (RX_FIFO_COUNT > 0x00)
    {
        serial_leerdxfifo(&FIFOByte);      // Lee
    }

}

//*****
//
// Comando
//
//*****
void VNC1L_Cerrar ()
{
    char FIFOByte = 0x00;

    serial_enviarbyte('C');                 // E
    serial_enviarbyte('L');                 // E
    serial_enviarbyte('F');                 // E
    serial_enviarbyte(' ');                 // E
    serial_enviarbyte('d');                 // E
    serial_enviarbyte('a');                 // E
    serial_enviarbyte('t');                 // E
    serial_enviarbyte('o');                 // E
    serial_enviarbyte('s');                 // E
    serial_enviarbyte('.');                 // E
    serial_enviarbyte('t');                 // E
    serial_enviarbyte('x');                 // E
    serial_enviarbyte('t');                 // E
    serial_enviarbyte(0x0D);                 // E

```

```

        while (RX_FIFO_COUNT < 0x05);           // Espera

        while (RX_FIFO_COUNT > 0x00)
        {
            serial_leerderxfifo(&FIFOByte);      // Lee
        }
    }

    /*******
    /*******
    // interrupción
    // Usar
    /*******
    /*******

    #int_rda

    void rutina_rda(void)
    {
        if(RCIF)
        {
            serial_agregararxfifo();
            RCIF=000;
        }
    }

    /*******
    /*******
    // Linea
    /*******
    /*******

    void main()
    {
        char sinc = 000;
        set_tris_b(0F1);           //C

        ciclo = 7;
        d=1;

        while(input(PIN_B0) == 0)   //Espera
        {

            control = input_b();
            control = control&0xf0;

            switch (control)         //Establece
            {
                case 12:

                    ciclo = 7;
                    d=1;
                    break;

                case 64:

                    ciclo = 1;
                    d=2;

```

```

        break;

    case 32:

        ciclo = 19;
        d=3;
        break;

    case 16:

        ciclo = 25;
        d=4;
        break;
    }
}

setup_adc_ports(AN0_TO_AN9);           //Configura
setup_adc(ADC_CLOCK_DIV_16);           //Configura
setup_timer_1(T1_INTERNAL|T1_DIV_BY_4);

set_timer1(0x0000);
VNC1L_Estado = VNC1L_Idle;              // Inicializa
serial_init(57600);                     // Inicializa
delay_ms(100);                          // Proporciona

while(1){
    i=0;
    if (VNC1L_Sinc() == Sincronizado)    // Espera
    {

        sinc = VNC1L_Sinc();

        if (VNC1L_Encontrar() == DiscoObtenido)    // Checa
        {

            serial_enviarbyte('S');                // E
            serial_enviarbyte('B');                // E
            serial_enviarbyte('D');                // E
            serial_enviarbyte(' ');                // E
            serial_enviarbyte(0x03);                // E
            serial_enviarbyte(0x00);                // E
            serial_enviarbyte(0x00);                // E
            serial_enviarbyte(0x0D);                // E
            delay_ms(1000);

            BRGH=1;                                //Establece
            BAUDCON = 0x08;
            SPBRGH = 0x00;
            SPBRG=0x04;

            set_timer1(00000);

            VNC1L_AbrirArchivo();

            serial_resetrxfifo();

            VNC1L_EscribirArchivo();
            n=1;                                    //Inicializa
            m=1;

            VNC1L_CerrarArchivo();

            serial_enviarbyte('S');                // E

```

```

        serial_enviarbyte('B');           // E
        serial_enviarbyte('D');           // E
        serial_enviarbyte(' ');           // E
        serial_enviarbyte(0x34);           // E
        serial_enviarbyte(0xC0);           // E
        serial_enviarbyte(0x00);           // E
        serial_enviarbyte(0x0D);           // E

        delay_ms(1000);
        set_uart_speed(57600);             //Restablece
    }

    while(1)
    {
        if (VNC1L_EncontrarDisco() != DiscoObtenido) // Checar
        {
            break;                                     // Actualizar
        }
    }
}
}
}

```

APÉNDICE C

Hojas de Datos



Low Drift, Low Power Instrumentation Amplifier

AD621

FEATURES

EASY TO USE

Pin-Strappable Gains of 10 and 100

All Errors Specified for Total System Performance

Higher Performance than Discrete In Amp Designs
Available in 8-Lead DIP and SOIC

Low Power, 1.3 mA Max Supply Current

Wide Power Supply Range (± 2.3 V to ± 18 V)

EXCELLENT DC PERFORMANCE

0.15% Max, Total Gain Error

± 5 ppm/ $^{\circ}$ C, Total Gain Drift

125 μ V Max, Total Offset Voltage

1.0 μ V/ $^{\circ}$ C Max, Offset Voltage Drift

LOW NOISE

9 nV/ $\sqrt{\text{Hz}}$, @ 1 kHz, Input Voltage Noise

0.28 μ V p-p Noise (0.1 Hz to 10 Hz)

EXCELLENT AC SPECIFICATIONS

800 kHz Bandwidth (G = 10), 200 kHz (G = 100)

12 μ s Settling Time to 0.01%

APPLICATIONS

Weigh Scales

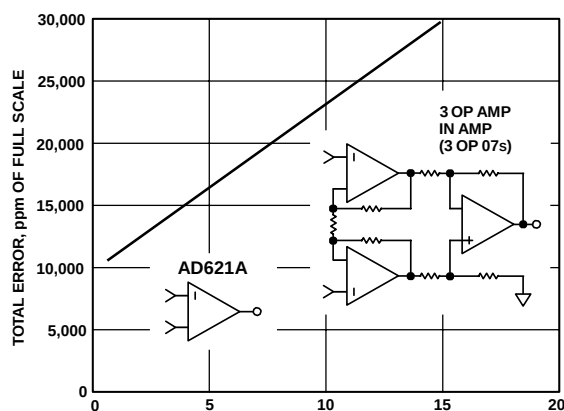
Transducer Interface and Data Acquisition Systems

Industrial Process Controls

Battery-Powered and Portable Equipment

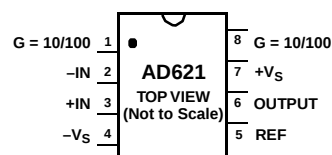
PRODUCT DESCRIPTION

The AD621 is an easy to use, low cost, low power, high accuracy instrumentation amplifier that is ideally suited for a wide range of applications. Its unique combination of high performance, small size and low power, outperforms discrete in amp implementations. High functionality, low gain errors, and low



CONNECTION DIAGRAM

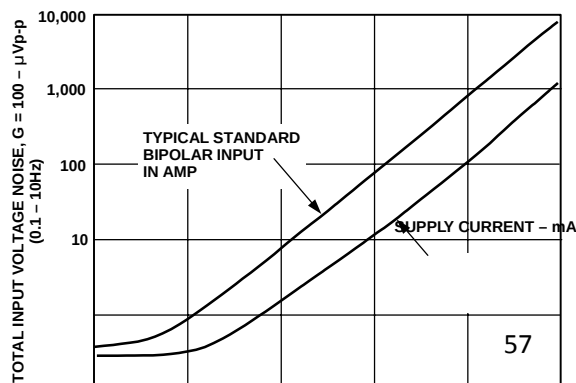
8-Lead Plastic Mini-DIP (N), Cerdip (Q)
and SOIC (R) Packages



gain drift errors are achieved by the use of internal gain setting resistors. Fixed gains of 10 and 100 can easily be set via external pin strapping. The AD621 is fully specified as a total system, therefore, simplifying the design process.

For portable or remote applications, where power dissipation, size, and weight are critical, the AD621 features a very low supply current of 1.3 mA max and is packaged in a compact 8-lead SOIC, 8-lead plastic DIP or 8-lead cerdip. The AD621 also excels in applications requiring high total accuracy, such as precision data acquisition systems used in weigh scales and transducer interface circuits. Low maximum error specifications including nonlinearity of 10 ppm, gain drift of 5 ppm/ $^{\circ}$ C, 50 μ V offset voltage, and 0.6 μ V/ $^{\circ}$ C offset drift ("B" grade), make possible total system performance at a lower cost than has been previously achieved with discrete designs or with other monolithic instrumentation amplifiers.

When operating from high source impedances, as in ECG and blood pressure monitors, the AD621 features the ideal combination of low noise and low input bias currents. Voltage noise is specified as 9 nV/ $\sqrt{\text{Hz}}$ at 1 kHz and 0.28 μ V p-p from 0.1 Hz to 10 Hz. Input current noise is also extremely low at 0.1 pA/ $\sqrt{\text{Hz}}$. The AD621 outperforms FET input devices with an input bias current specification of 1.5 nA max over the full industrial temperature range.



AD621 SUPER β ETA BIPOLAR INPUT
IN AMP

Figure 1. Three Op Amp IA Designs vs. AD621

REV. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

0.1 1k 10k 100k 1M 10M 100M
SOURCE RESISTANCE – K

Figure 2. Total Voltage Noise vs. Source Resistance

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 World Wide Web Site: <http://www.analog.com>
Fax: 781/326-8703 © Analog Devices, Inc., 2001

AD621-SPECIFICATIONS

Gain = 10 (Typical @ 25°C, $V_S = \pm 15$ V, and $R_L = 2$ k Ω , unless otherwise noted.)

Model	Conditions	AD621A			AD621B			AD621S ¹			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
GAIN											
Gain Error	$V_{OUT} = \pm 10$ V			0.15			0.05			0.15	%
Nonlinearity, $V_{OUT} = -10$ V to $+10$ V	$R_L = 2$ k Ω	2	10		2	10		2	10		ppm of FS
Gain vs. Temperature		-1.5	± 5		-1.5	± 5		-1	± 5		ppm/°C
TOTAL VOLTAGE OFFSET											
Offset (RTI)	$V_S = \pm 15$ V		75	250		50	125		75	250	μ V
Over Temperature	$V_S = \pm 5$ V to ± 15 V			400			215			500	μ V
Average TC	$V_S = \pm 5$ V to ± 15 V		1.0	2.5		0.6	1.5		1.0	2.5	μ V/°C
Offset Referred to the Input vs. Supply (PSR) ²	$V_S = \pm 2.3$ V to ± 18 V	95	120		100	120		95	120		dB
Total NOISE											
Voltage Noise (RTI)	1 kHz		13	17		13	17		13	17	nV/ $\sqrt{\text{Hz}}$
RTI	0.1 Hz to 10 Hz		0.55			0.55	0.8		0.55	0.8	μ V p-p
Current Noise	f = 1 kHz		100			100			100		fA/ $\sqrt{\text{Hz}}$
	0.1 Hz–10 Hz		10			10			10		pA p-p
INPUT CURRENT	$V_S = \pm 15$ V										
Input Bias Current			0.5	2.0		0.5	1.0		0.5	2	nA nA
Over Temperature				2.5			1.5			4	pA/°C
Average TC			3.0			3.0			8.0		nA
Input Offset Current			0.3	1.0		0.3	0.5		0.3	1.0	nA
Over Temperature				1.5			0.75			2.0	pA/°C
Average TC			1.5			1.5			8.0		
INPUT											
Input Impedance											
Differential			10 ¹²			10 ¹²			10 ¹²		G Ω pF
Common-Mode			10 ¹²			10 ¹²			10 ¹²		G Ω pF
Input Voltage Range ³	$V_S = \pm 2.3$ V to ± 5 V	- $V_S + 1.9$		+ $V_S - 1.2$	- $V_S + 1.9$		+ $V_S - 1.2$	- $V_S + 1.9$		+ $V_S - 1.2$	V
Over Temperature		- $V_S + 2.1$		+ $V_S - 1.3$	- $V_S + 2.1$		+ $V_S - 1.3$	- $V_S + 2.1$		+ $V_S - 1.3$	V
	$V_S = \pm 5$ V to ± 18 V	- $V_S + 1.9$		+ $V_S - 1.4$	- $V_S + 1.9$		+ $V_S - 1.4$	- $V_S + 1.9$		+ $V_S - 1.4$	V
Over Temperature		- $V_S + 2.1$		+ $V_S - 1.4$	- $V_S + 2.1$		+ $V_S - 1.4$	- $V_S + 2.3$		+ $V_S - 1.4$	V
Common-Mode Rejection Ratio DC to 60 Hz with 1 k Ω Source Imbalance	$V_{CM} = 0$ V to ± 10 V	93	110		100	110		93	110		dB
OUTPUT											
Output Swing	$R_L = 10$ k Ω , $V_S = \pm 2.3$ V to ± 5 V	- $V_S + 1.1$		+ $V_S - 1.2$	- $V_S + 1.1$		+ $V_S - 1.2$	- $V_S + 1.1$		+ $V_S - 1.2$	V
Over Temperature		- $V_S + 1.4$		+ $V_S - 1.3$	- $V_S + 1.4$		+ $V_S - 1.3$	- $V_S + 1.6$		+ $V_S - 1.3$	V
	$V_S = \pm 5$ V to ± 18 V	- $V_S + 1.2$		+ $V_S - 1.4$	- $V_S + 1.2$		+ $V_S - 1.4$	- $V_S + 1.2$		+ $V_S - 1.4$	V
Over Temperature		- $V_S + 1.6$		+ $V_S - 1.5$	- $V_S + 1.6$		+ $V_S - 1.5$	- $V_S + 2.3$		+ $V_S - 1.5$	V
Short Current Circuit		± 18			± 18			± 18			mA
DYNAMIC RESPONSE											
Small Signal, -3 dB Bandwidth			800			800			800		kHz
Slew Rate		0.75	1.2		0.75	1.2		0.75	1.2		V/ μ s
Settling Time to 0.01%	10 V Step		12			12			12		μ s
REFERENCE INPUT											
R_{IN}			20			20			20		k Ω
I_{IN}	$V_{IN} +, V_{REF} = 0$		50	60		50	60		+50	+60	μ A
Voltage Range		- $V_S + 1.6$		+ $V_S - 1.6$	- $V_S + 1.6$		+ $V_S - 1.6$	$V_S + 1.6$		+ $V_S - 1.6$	V
Gain to Output			1 ± 0.0001			1 ± 0.0001			1 ± 0.0001		
POWER SUPPLY											
Operating Range		± 2.3		± 18	± 2.3		± 18	± 2.3		± 18	V
Quiescent Current	$V_S = \pm 2.3$ V to ± 18 V		0.9	1.3		0.9	1.3		0.9	1.3	mA
Over Temperature			1.1	1.6		1.1	1.6		1.1	1.6	mA
TEMPERATURE RANGE	For Specified Performance		-40 to +85			-40 to +85			-55 to +125		°C

NOTES

¹See Analog Devices' military data sheet for 883B tested specifications.

²This is defined as the supply range over which PSRR is defined.

³Input Voltage Range = CMV + (Gain $\times$ V_{DIFF}).

Specifications subject to change without notice.

Gain = 100 (Typical @ 25°C, $V_S = \pm 15$ V, and $R_L = 2$ k Ω , unless otherwise noted.)

AD621

Model	Conditions	AD621A			AD621B			AD621S ¹			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
GAIN											
Gain Error	$V_{OUT} = \pm 10$ V			0.15			0.05			0.15	%
Nonlinearity, $V_{OUT} = -10$ V to $+10$ V	$R_L = 2$ k Ω	2		10	2		10	2		10	ppm of FS
Gain vs. Temperature		-1		± 5	-1		± 5	-1		± 5	ppm/°C
TOTAL VOLTAGE OFFSET											
Offset (RTI)	$V_S = \pm 15$ V		35	125		25	50		35	125	μ V
Over Temperature	$V_S = \pm 5$ V to ± 15 V			185			215			225	μ V
Average TC	$V_S = \pm 5$ V to ± 15 V		0.3	1.0		0.1	0.6		0.3	1.0	μ V/°C
Offset Referred to the Input vs. Supply (PSR) ²	$V_S = \pm 2.3$ V to ± 18 V	110	140		120	140		110	140		dB
Total NOISE											
Voltage Noise (RTI)	1 kHz		9	13		9	13		9	13	nV/ $\sqrt{Hz}$
RTI	0.1 Hz to 10 Hz		0.28			0.28	0.4		0.28	0.4	μ V p-p
Current Noise	f = 1 kHz		100			100			100		fA/ $\sqrt{Hz}$
	0.1 Hz–10 Hz		10			10			10		pA p-p
INPUT CURRENT	$V_S = \pm 15$ V										
Input Bias Current			0.5	2.0		0.5	1.0		0.5	2	nA nA
Over Temperature				2.5			1.5			4	pA/°C
Average TC			3.0			3.0			8.0		nA
Input Offset Current			0.3	1.0		0.3	0.5		0.3	1.0	nA
Over Temperature				1.5			0.75			2.0	pA/°C
Average TC			1.5			1.5			8.0		
INPUT											
Input Impedance											
Differential			10 ¹²			10 ¹²			10 ¹²		G Ω /pF
Common-Mode			10 ¹²			10 ¹²			10 ¹²		G Ω /pF
Input Voltage Range ³	$V_S = \pm 2.3$ V to ± 5 V	- $V_S + 1.9$		$+V_S - 1.2$	- $V_S + 1.9$		$+V_S - 1.2$	- $V_S + 1.9$		$+V_S - 1.2$	V
Over Temperature		- $V_S + 2.1$		$+V_S - 1.3$	- $V_S + 2.1$		$+V_S - 1.3$	- $V_S + 2.1$		$+V_S - 1.3$	V
	$V_S = \pm 5$ V to ± 18 V	- $V_S + 1.9$		$+V_S - 1.4$	- $V_S + 1.9$		$+V_S - 1.4$	- $V_S + 1.9$		$+V_S - 1.4$	V
		- $V_S + 2.1$		$+V_S - 1.4$	- $V_S + 2.1$		$+V_S - 1.4$	- $V_S + 2.3$		$+V_S - 1.4$	V
Over Temperature											
Common-Mode Rejection											
Ratio DC to 60 Hz with 1 k Ω Source Imbalance	$V_{CM} = 0$ V to ± 10 V	110	130		120	130		110	130		dB
OUTPUT											
Output Swing	$R_L = 10$ k Ω , $V_S = \pm 2.3$ V to ± 5 V	- $V_S + 1.1$		$+V_S - 1.2$	- $V_S + 1.1$		$+V_S - 1.2$	- $V_S + 1.1$		$+V_S - 1.2$	V
Over Temperature		- $V_S + 1.4$		$+V_S - 1.3$	- $V_S + 1.4$		$+V_S - 1.3$	- $V_S + 1.6$		$+V_S - 1.3$	V
	$V_S = \pm 5$ V to ± 18 V	- $V_S + 1.2$		$+V_S - 1.4$	- $V_S + 1.2$		$+V_S - 1.4$	- $V_S + 1.2$		$+V_S - 1.4$	V
Over Temperature		- $V_S + 1.6$		$+V_S - 1.5$	- $V_S + 1.6$		$+V_S - 1.5$	- $V_S + 2.3$		$+V_S - 1.5$	V
Short Current Circuit		± 18			± 18			± 18			mA
DYNAMIC RESPONSE											
Small Signal, -3 dB Bandwidth			200			200			200		kHz
Slew Rate		0.75	1.2		0.75	1.2		0.75	1.2		V/ μ s
Settling Time to 0.01%	10 V Step		12			12			12		μ s
REFERENCE INPUT											
R_{IN}			20			20			20		k Ω
I_{IN}	$V_{IN+}, V_{REF} = 0$		50	60		50	60		50	60	μ A
Voltage Range		- $V_S + 1.6$		$+V_S - 1.6$	- $V_S + 1.6$		$+V_S - 1.6$	$V_S + 1.6$		$+V_S - 1.6$	V
Gain to Output			1 ± 0.0001			1 ± 0.0001			1 ± 0.0001		
POWER SUPPLY											
Operating Range		± 2.3		± 18	± 2.3		± 18	± 2.3		± 18	V
Quiescent Current	$V_S = \pm 2.3$ V to ± 18 V		0.9	1.3		0.9	1.3		0.9	1.3	mA
Over Temperature			1.1	1.6		1.1	1.6		1.1	1.6	mA
TEMPERATURE RANGE											
For Specified Performance			-40 to +85			-40 to +85			-55 to +125		°C

NOTES

¹See Analog Devices' military data sheet for 883B tested specifications.

²This is defined as the supply range over which PSEE is defined.

³Input Voltage Range = CMV + (Gain $\times$ V_{DIFF}).

Octal Sample-and-Hold with Multiplexed Input

SMP08*

FEATURES

- Internal Hold Capacitors
- Low Droop Rate
- TTL/CMOS Compatible Logic Inputs
- Single or Dual Supply Operation
- Break-Before-Make Channel Addressing
- Compatible With CD4051 Pinout
- Low Cost

APPLICATIONS

- Multiple Path Timing Des skew for ATE
- Memory Programmers
- Mass Flow/Process Control Systems
- Multichannel Data Acquisition Systems
- Robotics and Control Systems
- Medical and Analytical Instrumentation
- Event Analysis
- Stage Lighting Control

GENERAL DESCRIPTION

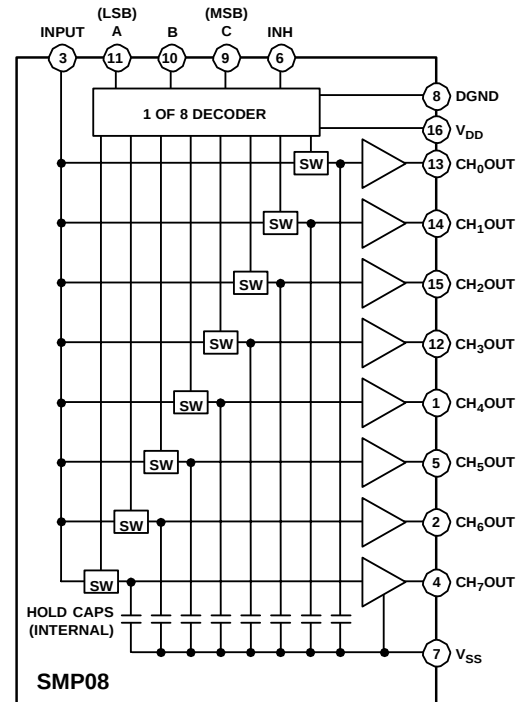
The SMP08 is a monolithic octal sample-and-hold; it has eight internal buffer amplifiers, input multiplexer, and internal hold capacitors. It is manufactured in an advanced oxide isolated CMOS technology to obtain high accuracy, low droop rate, and fast acquisition time. The SMP08 has a typical linearity error of only 0.01% and can accurately acquire a 10-bit input signal to

1/2 LSB in less than 7 microseconds. The SMP08's output swing includes the negative supply in both single and dual supply operation.

The SMP08 was specifically designed for systems that use a calibration cycle to adjust a multiple of system parameters. The low cost and high level of integration make the SMP08 ideal for calibration requirements that have previously required an ASIC, or high cost multiple D/A converters.

*Protected by U.S. Patent No. 4,739,281.

FUNCTIONAL BLOCK DIAGRAM



The SMP08 is also ideally suited for a wide variety of sample-and-hold applications including amplifier offset or VCA gain adjustments. One or more SMP08s can be used with single or multiple DACs to provide multiple set points within a system.

The SMP08 offers significant cost and size reduction over discrete designs. It is available in a 16-pin plastic DIP, or surface-mount SOIC package.

REV. D

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 617/329-4700 **World Wide Web Site: <http://www.analog.com>**
Fax: 617/326-8703 **© Analog Devices, Inc., 1996**

SMP08-SPECIFICATIONS

ELECTRICAL CHARACTERISTICS

(@ $V_{DD} = +5$ V, $V_{SS} = -5$ V, $DGND = 0$ V, $R_L = \text{No Load}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$ for SMP08F, unless otherwise noted)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Linearity Error		-3 V V_{IN} $+3$ V		0.01		%
Buffer Offset Voltage	V_{OS}	$T_A = +25$ C, $V_{IN} = 0$ V		2.5	10	mV
		-40 C T_A $+85$ C, $V_{IN} = 0$ V		3.5	20	mV
Hold Step	V_{HS}	$V_{IN} = 0$ V, $T_A = +25$ C to $+85$ C		2.5	4	mV
		$V_{IN} = 0$ V, $T_A = -40$ C			5	mV
Droop Rate	V_{CH}/t	$T_A = +25$ C, $V_{IN} = 0$ V		2	20	mV/s
Output Source Current	I_{SOURCE}	$V_{IN} = 0$ V ¹	1.2			mA
Output Sink Current	I_{SINK}	$V_{IN} = 0$ V ¹	0.5			mA
Output Voltage Range		$R_L = 20$ k	-3.0		+3.0	V
LOGIC CHARACTERISTICS						
Logic Input High Voltage	V_{INH}		2.4			V
Logic Input Low Voltage	V_{INL}				0.8	V
Logic Input Current	I_{IN}	$V_{IN} = 2.4$ V		0.5	1	A
DYNAMIC PERFORMANCE ²						
Acquisition Time ³	t_{AQ}	$T_A = +25$ C, -3 V to $+3$ V to 0.1%		3.6	7	s
Hold Mode Settling Time	t_H	To 1 mV of Final Value		1		s
Channel Select Time	t_{CH}			90		ns
Channel Deselect Time	t_{DCS}			45		ns
Inhibit Recovery Time	t_{IR}			90		ns
Slew Rate	SR			3		V/ s
Capacitive Load Stability		<30% Overshoot		500		pF
Analog Crosstalk		-3 V to $+3$ V Step		-72		dB
SUPPLY CHARACTERISTICS						
Power Supply Rejection Ratio	PSRR	$V_S = 5$ V to 6 V	60	75		dB
Supply Current	I_{DD}	$T_A = +25$ C		4	7.5	mA
		-40 C T_A $+85$ C		5	9.5	mA

ELECTRICAL CHARACTERISTICS

(@ $V_{DD} = +12$ V, $V_{SS} = 0$ V, $DGND = 0$ V, $R_L = \text{No Load}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$ for SMP08F, unless otherwise noted)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Linearity Error		60 mV V_{IN} 10 V		0.01		%
Buffer Offset Voltage	V_{OS}	$T_A = +25$ C, $V_{IN} = 6$ V		2.5	10	mV
		-40 C T_A $+85$ C, $V_{IN} = 6$ V		3.5	20	mV
Hold Step	V_{HS}	$V_{IN} = 6$ V, $T_A = +25$ C to $+85$ C		2.5	4	mV
		$V_{IN} = 6$ V, $T_A = -40$ C			5	mV
Droop Rate	V_{CH}/t	$T_A = +25$ C, $V_{IN} = 6$ V		2	20	mV/s
Output Source Current	I_{SOURCE}	$V_{IN} = 6$ V ¹	1.2			mA
Output Sink Current	I_{SINK}	$V_{IN} = 6$ V ¹	0.5			mA
Output Voltage Range		$R_L = 20$ k	0.06		10.0	V
		$R_L = 10$ k	0.06		9.5	V
LOGIC CHARACTERISTICS						
Logic Input High Voltage	V_{INH}		2.4			V
Logic Input Low Voltage	V_{INL}				0.8	V
Logic Input Current	I_{IN}	$V_{IN} = 2.4$ V		0.5	1	A
DYNAMIC PERFORMANCE ²						
Acquisition Time ³	t_{AQ}	$T_A = +25$ C, 0 V to 10 V to 0.1%		3.5	4.25	s
		-40 C T_A $+85$ C		3.75	6.00	s
Hold Mode Settling Time	t_H	To 1 mV of Final Value		1		s
Channel Select Time	t_{CH}			90		ns
Channel Deselect Time	t_{DCS}			45		ns
Inhibit Recovery Time	t_{IR}			90		ns
Slew Rate	SR	$R_L = 20$ k ⁴	3	4		V/ s
Capacitive Load Stability		<30% Overshoot		500		pF
Analog Crosstalk		0 V to 10 V Step		-72		dB
SUPPLY CHARACTERISTICS						
Power Supply Rejection Ratio	PSRR	10.8 V V_{DD} 13.2 V	60	75		dB
Supply Current	I_{DD}	$T_A = +25$ C		6.0	8.0	mA
		-40 C T_A $+85$ C		8.0	10.0	mA

NOTES

¹Outputs are capable of sinking and sourcing over 20 mA but offset is guaranteed at specified load levels.

²All input control signals are specified with $t_r = t_f = 5$ ns (10% to 90% of +5 V) and timed from a voltage level of 1.6 V.

³This parameter is guaranteed without test.

⁴Slew rate is measured in the sample mode with 0 V to 10 V step from 20% to 80%.

Specifications subject to change without notice.

SMP08

ABSOLUTE MAXIMUM RATINGS

V_{DD} to DGND	−0.3 V, 17 V
V_{DD} to V_{SS}	−0.3 V, 17 V
V_{LOGIC} to DGND	−0.3 V, V_{DD}
V_{IN} to DGND	V_{SS} , V_{DD}
V_{OUT} to DGND	V_{SS} , V_{DD}
Analog Output Current	20 mA

(Not Short-Circuit Protected)

Operating Temperature Range

FP, FS	−40 C to +85 C
Junction Temperature	+150 C
Storage Temperature	−65 C to +150 C
Lead Temperature (Soldering, 60 sec)	+300 C

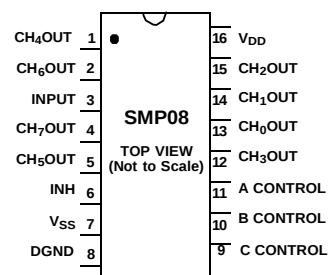
Package Type	θ_{JA}^*	θ_{JC}	Units
16-Pin Plastic DIP (P)	76	33	C/W
16-Pin SOIC (S)	92	27	C/W

* θ_{JA} is specified for worst case mounting conditions, i.e., θ_{JA} is specified for device in socket for plastic DIP package; θ_{JA} is specified for device soldered to printed circuit board for SO package.

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
SMP08FP	−40 C to +85 C	Plastic DIP	N-16
SMP08FS	−40 C to +85 C	SO-16	R-16A

PIN CONNECTIONS



CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the SMP08 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.





MICROCHIP PIC18F2455/2550/4455/4550

28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant SIE
- Low-speed (1.5 Mb/s) and full-speed (12 Mb/s)
- Supports control, interrupt, isochronous and bulk transfers
- Supports up to 32 endpoints (16 bidirectional)
- 1-Kbyte dual access RAM for USB
- On-board USB transceiver with on-chip voltage regulator
- Interface for off-chip USB transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μ A typical
- Sleep current down to 0.1 μ A typical
- Timer1 oscillator: 1.1 μ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Five Crystal modes, including High-Precision PLL for USB
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 MHz
- Internal oscillator block:
 - 8 user selectable frequencies, from 31 kHz to 8 MHz
 - User tunable to compensate for frequency drift
- Secondary oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High current sink/source: 25 mA/25 mA
- Three external interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 6.25 ns ($T_{CY}/16$)
 - Compare is 16-bit, max. resolution 100 ns (T_{CY})
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead-time
 - Auto-Shutdown and Auto-Restart
- Addressable USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI™ (all 4 modes) and I²C™ Master and Slave modes
- 10-bit, up to 13-channels Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Dual analog comparators with input multiplexing

Special Microcontroller Features:

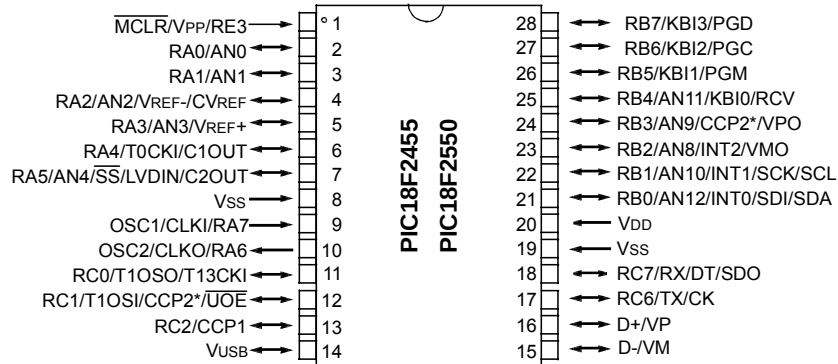
- C compiler optimized architecture with optional extended instruction set
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle data EEPROM memory typical
- Flash/data EEPROM retention: > 40 years
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Wide operating voltage range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-bit
	FLASH (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I ² C			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

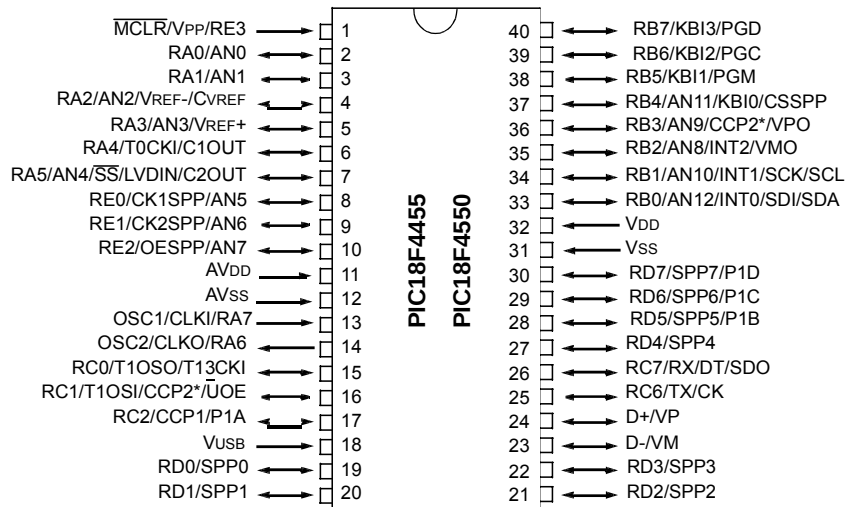
PIC18F2455/2550/4455/4550

Pin Diagrams

28-Pin SDIP, SOIC



40-Pin PDIP



Note: Pinouts are subject to change.

* Assignment of this feature is dependent on device configuration.

TMFutureTechnology
Devices International Ltd.



VINCULUM
BINDING USB TECHNOLOGIES

VDIP1

Vinculum VNC1L-1A Prototyping Module



<http://www.vinculum.com>

Copyright © Future Technology Devices International Ltd. 2006

1. Introduction and Features

1.1 Introduction

The VDIP1 module is an MCU to embedded USB host controller development module for the VNC1L-1A I.C. device. The VDIP1 is supplied on a PCB designed to fit into a 24 pin DIP socket, and provides access to the UART, parallel FIFO, and SPI interface pins on the VNC1L-1A device, via its AD and AC bus pins. Not only is it ideal for developing and rapid prototyping of VNC1L-1A designs, but also an attractive quantity discount structure makes this module suitable for incorporation into low and medium volume finished product designs.

The Vinculum VNC1L-1A is the first of F.T.D.I.'s Vinculum family of Embedded USB host controller integrated circuit devices. Not only is it able to handle the USB Host Interface, and data transfer functions but owing to the inbuilt MCU and embedded Flash memory, Vinculum can encapsulate the USB device classes as well. When interfacing to mass storage devices such as USB Flash drives, Vinculum also transparently handles the FAT File structure communicating via UART, SPI or parallel FIFO interfaces via a simple to implement command set. Vinculum provides a new cost effective solution for providing USB Host capability into products that previously did not have the hardware resources available. The VNC1L-1A is available in Pb-free (RoHS compliant) compact 48-Lead LQFP package.

1.2 Features

- Uses F.T.D.I.'s VNC1L-1A embedded USB host controller I.C. device.
- USB 'A' type socket to interface with USB peripheral devices
- Second USB interface port available via module pins, if required.
- Jumper selectable UART, parallel FIFO, or SPI MCU interfaces.
- Single 5V supply input.
- Auxiliary 3.3 V / 200 mA power output to external logic.
- Power indicator, and USB traffic indicator LED's.
- Program or update firmware via USB Flash disk or via UART / parallel FIFO / SPI interface.
- VNC1L-1A firmware programming control pins PROG# and RESET# brought out onto jumper interface
- VDIP1 is a Pb-free, RoHS complaint development module.
- Schematics, and firmware files available for download from the [Vinculum website](#).

2. Pin Out and Signal Descriptions

2.1 Module Pin Out

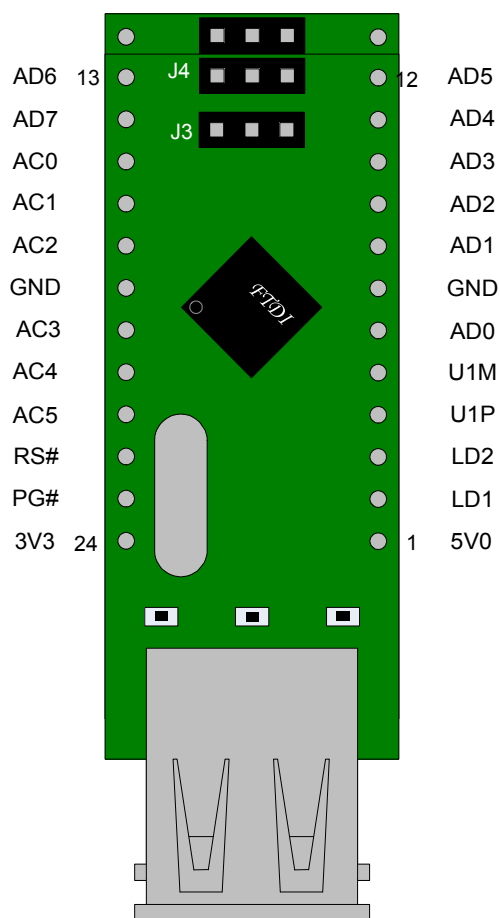


Figure 1 - VDIP1 Module Pin Out

2.2 Pin Signal Descriptions

Table 1 - VDIP1 module pin descriptions

Pin No.	Name	Pin Name on PCB	Type	Description
1	5V0	5V0	PWR Input	5.0 V module supply pin. This pin provides the 5.0V output on the USB 'A' type socket, and also the 3.3V supply to VNC1L-1A, via an on-board 3.3 V L.D.O.
2	LED1	LD1	Output	USB port 1 traffic activity indicator LED. This pin is hard wired to a green LED on board the PCB. It is also brought out onto this pin which allows for the possibility of bringing out an additional LED traffic indicator out of the VDIP1 board. For example, if the VDIP1 USB connector is brought out onto an instrument front panel, an activity LED could be mounted along side it.
3	LED2	LD2	Output	USB port 2 traffic activity indicator LED. This pin is hard wired to a green LED on board the PCB. It is also brought out onto this pin which allows for the possibility of bringing out an additional LED traffic indicator out of the VDIP1 board. For example, if the VDIP1 USB connector is brought out onto an instrument front panel, an activity LED could be mounted along side it.
4	USBD1P	U1P	I/O	USB host / slave port 1 - USB Data Signal Plus with integrated pull up / pull down resistor. Module has on board 27 Ω USB series resistor. This pin can be brought out along with pin 5 to provide a second USB port, if required.
5	USBD1M	U1M	I/O	USB host / slave port 1 - USB Data Signal Minus with integrated pull up / pull down resistor. Module has on board 27 Ω USB series resistor. This pin can be brought out along with pin 4 to provide a second USB port, if required.
6	ADBUS0	AD0	I/O	5V safe bidirectional data / control bus, AD bit 0
7	GND	GND	PWR	Module ground supply pin
8	ADBUS1	AD1	I/O	5V safe bidirectional data / control bus, AD bit 1
9	ADBUS2	AD2	I/O	5V safe bidirectional data / control bus, AD bit 2
10	ADBUS3	AD3	I/O	5V safe bidirectional data / control bus, AD bit 3
11	ADBUS4	AD4	I/O	5V safe bidirectional data / control bus, AD bit 4
12	ADBUS5	AD5	I/O	5V safe bidirectional data / control bus, AD bit 5
13	ADBUS6	AD6	I/O	5V safe bidirectional data / control bus, AD bit 6
14	ADBUS7	AD7	I/O	5V safe bidirectional data / control bus, AD bit 7
15	ACBUS0	AC0	I/O	5V safe bidirectional data / control bus, AC bit 0
16	ACBUS1	AC1	I/O	5V safe bidirectional data / control bus, AC bit 1
17	ACBUS2	AC2	I/O	5V safe bidirectional data / control bus, AC bit 2
18	GND	GND	PWR	Module ground supply pin
19	ACBUS3	AC3	I/O	5V safe bidirectional data / control bus, AC bit 3
20	ACBUS4	AC4	I/O	5V safe bidirectional data / control bus, AC bit 4
21	ACBUS5	AC5	I/O	5V safe bidirectional data / control bus, AC bit 5
22	RESET#	RS#	Input	Can be used by an external device to reset the VNC1L-1A. This pin can be used in combination with PROG# and the UART / parallel FIFO / SPI interface to program firmware into the VNC1L-1A.
23	PROG#	PG#	Input	This pin is used in combination with the RESET# pin and the UART / parallel FIFO / SPI interface to program firmware into the VNC1L-1A.
24	3V3	3V3	PWR Output	3.3V output from VDIP1's on board 3.3V L.D.O.

2.3 I/O Configuration Using The Jumper Pin Header

Two three way jumper pin headers are provided to allow for simple configuration of the I/O on data and control bus pins of the VDIP1. This is done by a combination of pulling up or pulling down the VNC1L-1A's ACBUS5 (pin 46) and ACBUS6 (pin 47). The relevant portion of the VDIP1 module schematic is shown in figure 7, below.

Figure 2 - VDIP1 On-board jumper pin configuration.

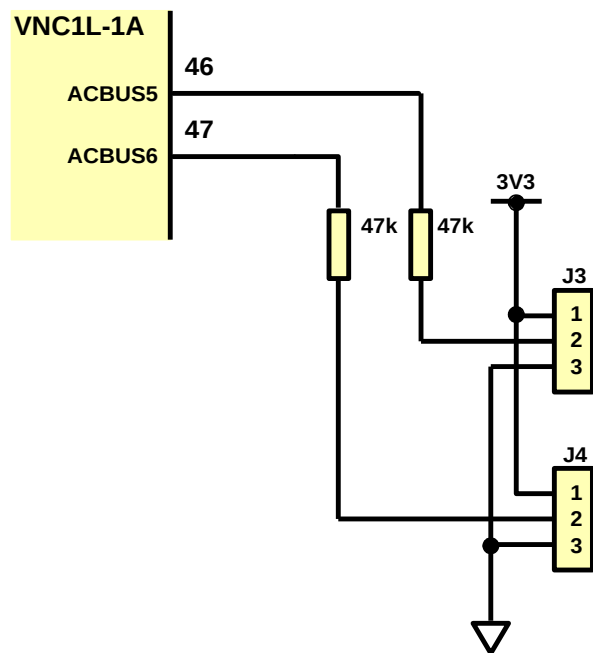


Table 2 - Port Selection Jumper Pins

ACBUS6 (VNC1L-1A pin 47)	ACBUS5 (VNC1L-1A pin 46)	I/O Mode
Pull-Up	Pull-Up	Serial UART
Pull-Up	Pull-Down	SPI
Pull-Down	Pull-Up	Parallel FIFO
Pull-Down	Pull-Down	Serial UART

Pin No.	Name	Pin Name on PCB	Type	Description	Data and Control Bus Configuration Options			
					UART Interface	Parallel FIFO Interface	SPI Slave Interface	I/O Port
6	ADBUS0	AD0	I/O	5V safe bidirectional data / control bus, AD bit 0	TXD	D0	SCLK	PortAD0
8	ADBUS1	AD1	I/O	5V safe bidirectional data / control bus, AD bit 1	RXD	D1	SDI	PortAD1
9	ADBUS2	AD2	I/O	5V safe bidirectional data / control bus, AD bit 2	RTS#	D2	SDO	PortAD2
10	ADBUS3	AD3	I/O	5V safe bidirectional data / control bus, AD bit 3	CTS#	D3	CS	PortAD3
11	ADBUS4	AD4	I/O	5V safe bidirectional data / control bus, AD bit 4	DTR#	D4		PortAD4
12	ADBUS5	AD5	I/O	5V safe bidirectional data / control bus, AD bit 5	DSR#	D5		PortAD5
13	ADBUS6	AD6	I/O	5V safe bidirectional data / control bus, AD bit 6	DCD#	D6		PortAD6
14	ADBUS7	AD7	I/O	5V safe bidirectional data / control bus, AD bit 7	RI#	D7		PortAD7
15	ACBUS0	AC0	I/O	5V safe bidirectional data / control bus, AC bit 0	TXDEN#	RXF#		PortAC0
16	ACBUS1	AC1	I/O	5V safe bidirectional data / control bus, AC bit 1		TXE#		PortAC1
17	ACBUS2	AC2	I/O	5V safe bidirectional data / control bus, AC bit 2		WR		PortAC2
19	ACBUS3	AC3	I/O	5V safe bidirectional data / control bus, AC bit 3		RD#		PortAC3
20	ACBUS4	AC4	I/O	5V safe bidirectional data / control bus, AC bit 4				PortAC4

2.4 UART Interface Signal Descriptions

Table 4 - Data and Control Bus Signal Mode Options - UART Interface

Table 3 - Data and control bus configuration options

<i>Pin No.</i>	<i>Name</i>	<i>Type</i>	<i>Description</i>
6	TXD	Output	Transmit asynchronous data output
8	RXD	Input	Receive asynchronous data input
9	RTS#	Output	Request To Send Control Output / Handshake signal
10	CTS#	Input	Clear To Send Control Input / Handshake signal
11	DTR#	Output	Data Terminal Ready Control Output / Handshake signal
12	DSR#	Input	Data Set Ready Control Input / Handshake signal
13	DCD#	Input	Data Carrier Detect Control Input
14	RI#	Input	Ring Indicator Control Input. When the Remote Wake up option is enabled in the EEPROM, taking RI# low can be used to resume the PC USB Host controller from suspend.
15	TXDEN#	Input	Enable Transmit Data for RS485 designs

TLC271, TLC271A, TLC271B

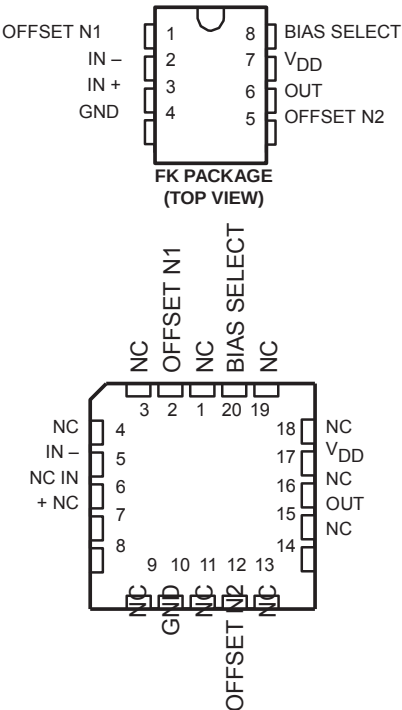
LinCMOS □ PROGRAMMABLE LOW-POWER OPERATIONAL AMPLIFIERS

SLOS090C – NOVEMBER 1987 – REVISED AUGUST 1997
D, JG, OR P PACKAGE
(TOP VIEW)

Input Offset Voltage Drift . . . Typically
0.1 μ V/Month, Including the First 30 Days
Wide Range of Supply Voltages Over

–40°C to 85°C . . . 4 V to 16 V
–55°C to 125°C . . . 5 V to 16 V

Single-Supply Operation
Common-Mode Input Voltage Range Extends Below the Negative Rail (C-Suffix and I-Suffix Types)
Low Noise . . . 25 nV/ $\sqrt$ Hz Typically at f = 1 kHz (High-Bias Mode)
Output Voltage Range includes Negative Rail
High Input Impedance . . . 10¹² Ω Typ
ESD-Protection Circuitry
Small-Outline Package Option Also Available in Tape and Reel
Designed-In Latch-Up Immunity



description

The TLC271 operational amplifier combines a wide range of input offset voltage grades with low offset voltage drift and high input impedance. In addition, the TLC271 offers a bias-select mode

that allows the user to select the best combination of power dissipation and ac performance for a particular application. These devices use Texas Instruments silicon-gate LinCMOS □ technology, which provides offset voltage stability far exceeding the stability available with conventional metal-gate processes.

NC – No internal connection

AVAILABLE OPTIONS					
	V _{IO} ma	SMALL OUTLI	CHIP CARRI	CERA MIC	PLAS TI
0°	2 mV	TLC271B			TLC271B
C	5 mV	CD			CP
–	2 mV	TLC271			TLC271
40°C	5 mV	BID			BIP
–					
55°C	10 mV	TLC271M	TLC271MF	TLC271M	TLC271MF

The D package is available taped and reeled. Add R suffix to the device type (e.g., TLC271BCDR).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.
is a trademark of Texas Instruments Incorporated.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include

PARAMETER†	BIAS-SELECT MODE			UNIT
	HIGH	MEDIUM	LOW	
P_D	3375	525	50	μW
SR	3.6	0.4	0.03	V/ μs
V_n	25	32	68	nV/ $\sqrt{Hz}$
B_1	1.7	0.5	0.09	MHz
A_{VD}	23	170	480	V/mV

† Typical at $V_{DD} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

description (continued)

Using the bias-select option, these cost-effective devices can be programmed to span a wide range of applications that previously required BiFET, NFET or bipolar technology. Three offset voltage grades are available (C-suffix and I-suffix types), ranging from the low-cost TLC271 (10 mV) to the TLC271B (2 mV) low-offset version. The extremely high input impedance and low bias currents, in conjunction with good common-mode rejection and supply voltage rejection, make these devices a good choice for new state-of-the-art designs as well as for upgrading existing designs.

In general, many features associated with bipolar technology are available in LinCMOS □ operational amplifiers, without the power penalties of bipolar technology. General applications such as transducer interfacing, analog calculations, amplifier blocks, active filters, and signal buffering are all easily designed with the TLC271. The devices also exhibit low-voltage single-supply operation, making them ideally suited for remote and inaccessible battery-powered applications. The common-mode input voltage range includes the negative rail.

A wide range of packaging options is available, including small-outline and chip-carrier versions for high-density system applications.

The device inputs and output are designed to withstand -100-mA surge currents without sustaining latch-up. The TLC271 incorporates internal ESD-protection circuits that prevent functional failures at voltages up to 2000 V as tested under MIL-STD-883C, Method 3015.2; however, care should be exercised in handling these devices as exposure to ESD may result in the degradation of the device parametric performance.

The C-suffix devices are characterized for operation from 0°C to 70°C . The I-suffix devices are characterized for operation from -40°C to 85°C . The M-suffix devices are characterized for operation over the full military temperature range of -55°C to 125°C .

bias-select feature

The TLC271 offers a bias-select feature that allows the user to select any one of three bias levels depending on the level of performance desired. The tradeoffs between bias levels involve ac performance and power dissipation (see Table 1).

REFERENCIAS BIBLIOGRÁFICAS

- [1] Cooper W.; Helfrick A., (1991), ***Instrumentación Electrónica Moderna y Técnicas de Medición***, México, Prentice Hall.
- [2] Franco S., (2002), ***Diseño con amplificadores operacionales y circuitos integrados analógicos***, 3ª edición, México, Mc Graw Hill.
- [3] Sapiensman, (2011), ***Sistemas de adquisición de datos***, [internet], disponible desde: <www.sapiensman.com/control_automatico/control_automatico7.htm>, [acceso en agosto 2011].
- [4] Saravia E., (2004), ***Sistema autónomo de adquisición de datos portátil***, [internet], Facultad de Ingeniería, disponible desde: <www.3.fi.mdp.edu.ar/electronica/articulos/>, [acceso en agosto 2011].
- [5] Vinculum, (2008), ***Vinculum Firmware User Manual***, [internet], Version 2.05, Future Technology Devices International Ltd., disponible desde: <http://www.ftdichip.com/Firmware/Precompiled/UM_VinculumFirmware_V205.pdf>, [acceso en julio 2011].