

---

# Capítulo 3

## Desarrollo del Hardware

---

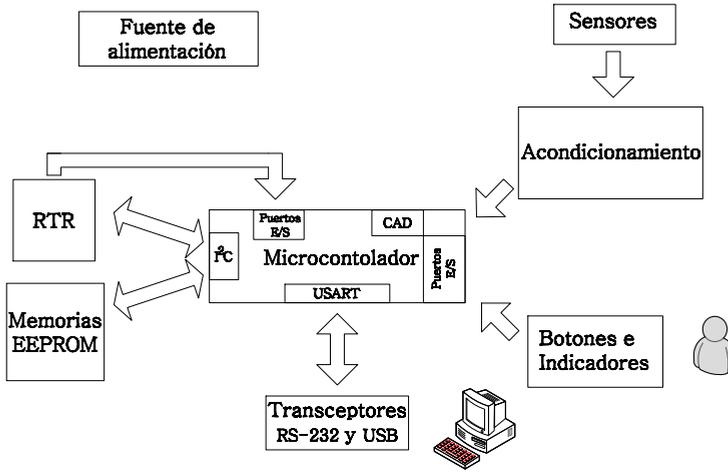
En este capítulo estudiaremos los elementos físicos que componen al *datalogger*, es decir su hardware, pero no se tratará nada acerca de la interacción que estos elementos realizan para desempeñar las funciones del sistema, esta parte se reservará para el siguiente capítulo.

En la figura 3.1 se observa el diagrama general de las partes que componen al *datalogger*. La figura muestra un diagrama de bloques del sistema, en el centro de éste se encuentra el microcontrolador. La función del microcontrolador es ejecutar el programa contenido en él de manera secuencial. A través de la ejecución de las instrucciones contenidas en este programa, el microcontrolador maneja la interacción tanto de sus módulos internos como los componentes externos a él. La interacción de todos estos elementos permite al *datalogger* realizar sus funciones.

En la parte superior derecha de la figura 3.1 se encuentra un bloque que representa a los sensores. La señal de éstos, una vez pasando por una etapa de acondicionamiento, es adquirida por el microcontrolador a través de dos de sus módulos; el convertidor analógico/digital (CAD) y las terminales de entrada/salida (E/S). Como ya se mencionó, el módulo usado para la adquisición de cada señal depende de la naturaleza de la misma, de esta manera, se utilizó el CAD para adquirir la señal de temperatura ambiente y los puertos de E/S para adquirir la señal de precipitación pluvial.

En la parte izquierda de la figura 3.1 se aprecian ciertos dispositivos que nos servirán para guardar la información y el manejo de las estampas de tiempo, estos son el reloj en tiempo real (RTR) y las memorias EEPROM externas. Ambos dispositivos realizan una comunicación bidireccional a través de la interfaz I<sup>2</sup>C con el microcontrolador. Esta conexión sirve, en el caso de las memorias EEPROM, para guardar y leer información de ellas y en el caso del RTR para configurarlo y leer el valor de tiempo y hora de éste.

Adicionalmente al manejo de la fecha y hora, el RTR tiene una función adicional; a través de sus alarmas programables proporciona al microcontrolador una señal externa para que éste realice determinadas funciones programadas.



**Fig. 3.1.** Diagrama de los elementos de hardware que compone al *datalogger*.

Lo anterior evita que el microcontrolador sea el encargado de llevar el control de tiempo de estas funciones. Estas señales son recibidas por el microcontrolador a través de terminales de sus puertos de E/S.

De igual forma, a través de la conexión de controles e indicadores en determinadas terminales de sus puertos de E/S el microcontrolador se comunica con el usuario, lo cual permite una interacción en sitio entre éste y el *datalogger*.

La unidad serie universal síncrona/asíncrona (USART) fue utilizada para implementar los protocolos RS-232 y USB que servirán para conectar al *datalogger* con una PC. Por último se implementó una fuente de alimentación que proporcionará energía a todos los componentes del *datalogger* a partir de una batería de 1.5 [V].

### 3.1. El microcontrolador

El microcontrolador, al ser el dispositivo de control del *datalogger* y él que proporciona a través de sus módulos gran parte de la funcionalidad de éste, es la primera parte a escoger en la realización del proyecto. Se eligió el microcontrolador ATmega16 de marca ATMEL de la familia AVR para integrar el núcleo del *datalogger*. La elección se basó en que este microcontrolador es

un dispositivo muy barato, disponible en el mercado nacional y que además disponían de las herramientas necesarias para el desarrollo de su programa.

En esta sección se tratará de manera general las características de este microcontrolador y de los módulos e interfaces que lo componen, aunque sólo se profundizara en aquellos que fueron utilizadas para el desarrollo de este proyecto.

### 3.1.1. Características generales

La familia AVR integra a microcontroladores de 8 bits contenidos en un solo chip basados en una arquitectura modificada Harvard. Esta familia de microcontroladores fue una de las primeras en tener la memoria de programa almacenada en una memoria interna flash.

La arquitectura modificada Harvard es una derivación de la arquitectura Harvard original. Esta arquitectura modificada mantiene la memoria de datos y la del programa separados en diferentes regiones, pero el CPU del microcontrolador tiene la habilidad de leer datos de la memoria del programa utilizando instrucciones especiales.

Dentro de las características de la familia AVR se encuentran:

- Contienen memoria del tipo flash, EEPROM y SRAM integradas, quitando la necesidad, para muchas aplicaciones, de utilizar memoria externa.
- Las instrucciones del programa están almacenadas en memoria no volátil flash (a pesar de ser microcontroladores de 8 bits cada instrucción ocupa una o dos palabras de 16 bits).
- Su espacio de dirección de datos consta de archivos de registro, registros de entrada/salida y la memoria SDRAM.
- Poseen 32 registros de trabajo de un solo byte.
- Tienen un diseño en base a un *pipeline* de dos niveles. Lo anterior significa que la siguiente instrucción a ejecutar es traída mientras que la actual es ejecutada por el CPU del microcontrolador.
- La mayoría de sus instrucciones se ejecutan en uno o dos ciclos de reloj. Esto los hace relativamente rápidos, en comparación con otros microcontroladores de 8 bits.
- Fueron diseñados teniendo en cuenta una ejecución eficiente, con código compilado de C y contienen varios apuntadores integrados para esta tarea.

- Soportan velocidades de reloj de 0-20 [MHz].

Dentro de las características específicas del modelo de microcontrolador usado se tienen las siguientes:

- 16 [kB] de memoria de programa flash
- 512 [Bytes] de memoria EEPROM
- 1 [kB] de memoria interna SRAM
- Cuatro puertos de entrada/salida (E/S) programables
- Cada puerto de E/S posee ocho terminales
- Dos contadores/temporizadores de 8-bits
- Un contador/temporizador de 16-bits
- Convertidor analógico-digital de ocho canales y 10-bits de resolución
- Interfaz de comunicación I<sup>2</sup>C
- Unidad de comunicación síncrona-asíncrona USART
- Interfaz serial maestra/esclava SPI
- Comparador analógico interno

No todos estos módulos del microcontrolador son utilizados por el *data-logger*. En la figura 3.2 se muestra una vista del microcontrolador y de los módulos usados de éste.

Para que el microcontrolador pueda operar sólo requiere de ser alimentado, proporcionarle una señal de reloj y conectar su terminal de *reset* a través de una resistencia de *pull-up* a  $V_{CC}$ . Esta resistencia debe de ser no menor a 4.7 [k $\Omega$ ] cuando se utiliza como fue en este caso un programador SPI. En la figura 3.3 se aprecian estos elementos; el cristal de cuarzo sirve para que el circuito oscilador interno del microcontrolador funcione y a partir de éste se genere la señal de reloj requerida por el microcontrolador. La terminal de *reset* está conectada a  $V_{CC}$  y a un interruptor para llevarla a tierra, aunque la terminal de *reset* posee internamente un filtro paso bajas, el capacitor de 22 [pF] proporciona inmunidad adicional al ruido mecánico del interruptor. Adicionalmente, a las terminales de salida del módulo SPI del microcontrolador, se colocó un conector para acoplar, en éstas, el programador que descarga el programa al microcontrolador.

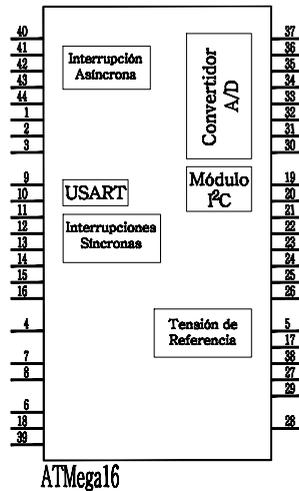


Fig. 3.2. Diagrama de los elementos del microcontrolador usados.

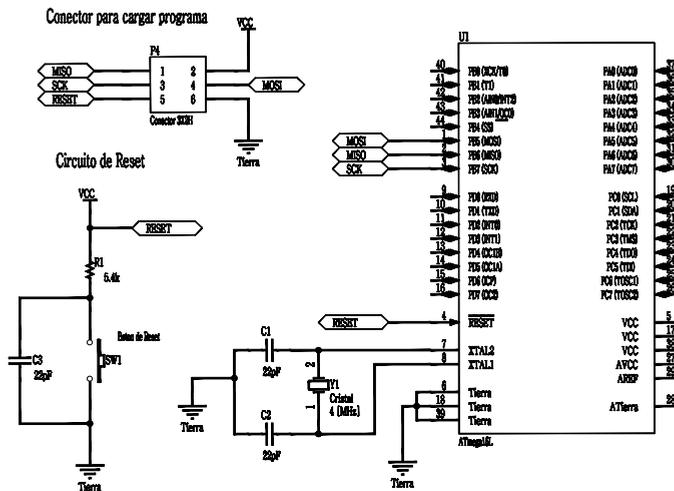


Fig. 3.3. Circuito básico del microcontrolador.

Antes de estudiar el funcionamiento de los módulos internos del microcontrolador, veremos las características generales de éste que ayudaron en la realización del *datalogger*.

### 3.1.2. La señal de reloj

Para la ejecución de su programa, el microcontrolador ATmega16 requiere de una señal de reloj. Esta señal puede ser provista de manera externa o

generarla al colocar un cristal de cuarzo o un circuito RC en las terminales que contiene el microcontrolador para ello. Las frecuencias de oscilación de estos elementos determinarán la frecuencia de la señal de reloj de salida del circuito oscilador.

Debido a la gran estabilidad en la oscilación de los cristales, se optó por usar uno de ellos con una frecuencia de 4 [MHz].

La frecuencia de reloj determina la velocidad a la que el CPU del microcontrolador funciona y también sirve como una señal de reloj base, para los distintos módulos del microcontrolador.

### 3.1.3. Modos de ahorro de energía

El microcontrolador ATmega16 incorpora distintas maneras de permitir un ahorro energético en su funcionamiento. En primer lugar, no todos los módulos que están integrados a éste se encuentran encendidos en todo momento. Estos son encendidos individualmente y pueden ser apagados en el instante en que no se requieran, para disminuir el consumo total de energía del microcontrolador.

Dependiendo de la aplicación en que el microcontrolador se use, es posible que éste no requiera estar funcionando en todo momento, con este fin el microcontrolador ATmega16 incorpora varios modos de funcionamiento de bajo consumo, que disminuyen su gasto energético al desactivar todos sus módulos o incluso su propio CPU. Estos modos de bajo consumo son los siguientes: modo inactivo (*idle mode*), modo de reducción de ruido del convertidor A/D (*ADC noise reduction mode*), modo apagado (*power-down mode*), modo de ahorro de energía (*power-save mode*), modo de espera (*standby mode*) y modo de espera extendido (*extended standby mode*).

De estos modos el único que fue usado fue el modo apagado, a continuación estudiaremos sus características.

#### **Modo apagado**

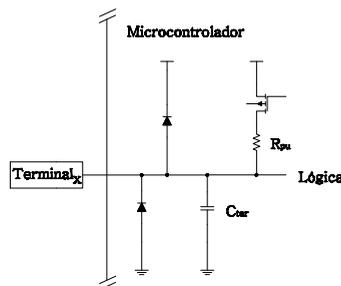
En el modo apagado el microcontrolador detiene todas las señales de reloj del sistema incluyendo la de su CPU, por lo que la ejecución de su programa se detiene. En este modo, el consumo de corriente del microcontrolador se reduce a menos de 1 [ $\mu\text{A}$ ], lo cual representa un consumo mínimo en comparación con su estado activo, en el cual tiene un consumo promedio de 1.1 [mA]. La desventaja de este modo es que, dado que ninguna señal de reloj está activa, existen muy pocas maneras de sacar al microcontrolador de este

estado. En el diseño del *datalogger* se utilizaron las interrupciones externas del microcontrolador para ese fin.

### 3.1.4. Puertos de E/S e interrupciones externas

Como ya se mencionó, en las características generales del ATmega16, este microcontrolador posee 32 terminales de E/S programables, distribuidas en 4 puertos. Sin embargo, no todas estas terminales están disponibles en todo momento. Lo anterior se debe a que éstas también desempeñan la función de servir como entradas y salidas a los distintos módulos internos del microcontrolador. Una vez que cierto módulo es activado, éste toma control del puerto y lo configura de acuerdo a sus necesidades.

Las líneas de E/S se configuran a través de registros del microcontrolador. Cada terminal de un puerto puede ser configurado de manera independiente, por lo cual se puede hacer una configuración mixta de entradas y salidas de un determinado puerto. En caso de configurar una terminal como entrada, a ésta se le puede activar su resistencia interna de *pull up*. Cada terminal cuenta con diodos de protección a tierra y al voltaje de alimentación y tiene la capacidad suficiente para que se drene de ellos hasta 40 [mA]. El diagrama interno de un terminal de un puerto se puede apreciar en la figura 3.4.



**Fig. 3.4.** Diagrama de un puerto E/S del microcontrolador.

Las terminales de E/S del microcontrolador, configuradas como entradas, permiten que el microcontrolador lea el valor de tensión que posea esta terminal. Si esta tensión se aplica a una terminal de entrada que pueda disparar una interrupción externa y dicha interrupción se encuentra configurada y activada, una interrupción en la ejecución del microcontrolador ocurre. Las interrupciones en el ATmega16, al igual que en cualquier microcontrolador, son señales que le indica a la ejecución de un programa que cierto evento ha ocurrido y que requiere de su atención. La ventaja del uso de interrupciones es que evitan gastar tiempo del procesador en estar revisando, constantemente, si algún tipo

de evento ha ocurrido. Las terminales con interrupción externa, de acuerdo al modo en que detectan las señales a su entrada, se dividen en dos tipos:

- Síncronas y
- Asíncronas

Una terminal con modo de detección síncrona requiere de una señal de reloj para detectar un cambio en la señal de entrada. En el caso del microcontrolador usado, esta señal de reloj es generada a través del reloj principal del sistema. Las terminales del microcontrolador ATMega16 que poseen un modo de detección síncrono son dos y se pueden configurar para lanzar una interrupción en caso de que detecten cualquiera de las siguientes señales:

- Un nivel de señal bajo en la terminal
- Cualquier cambio en el nivel de la terminal
- Un flanco de bajada en la terminal
- Un flanco de subida en la terminal

Una terminal con modo de detección asíncrona no requiere señal de reloj alguna para detectar cambios en su entrada. El microcontrolador ATMega16 sólo posee una terminal con este tipo de interrupción y ésta únicamente puede ser configurada para generar una interrupción mediante un flanco de bajada o de subida.

Todas las interrupciones externas pueden usarse para despertar al microcontrolador del modo de bajo consumo usado. Para realizar esto, en el caso de las interrupciones síncronas, sólo lo pueden realizar mediante la detección de un cambio en el nivel de la señal a la entrada, mismo que deberá de mantenerse hasta que el microcontrolador salga del estado de bajo consumo. Señales más cortas no garantizan despertar al microcontrolador. En el caso de la interrupción asíncrona la detección de un flanco de subida o bajada es suficiente para reanudar la operación normal del microcontrolador.

### 3.1.5. El convertidor analógico/digital

El convertidor analógico digital (A/D) del microcontrolador posee las siguientes características:

- Realiza conversiones a una resolución máxima de 10 bits.
- Su método de conversión es a través de aproximaciones sucesivas.

- Puede ser configurado para utilizar una tensión de referencia generada por el mismo microcontrolador o una generada de manera externa.

Un convertidor de aproximaciones sucesivas realiza una conversión a través de un reloj, un registro y un convertidor digital analógico, éste último requiere de una tensión de referencia. La salida del convertidor analógico digital es de 10 bits con un margen que va desde tierra ( $0000_{\text{H}}$ ) hasta su tensión de referencia ( $03\text{FF}_{\text{H}}$ ). En el caso del *datalogger*, para mantener un número mínimo de componentes externos, se decidió utilizar la tensión de referencia generada internamente por el microcontrolador, la cual tiene un valor de 2.56 [V]. Esta tensión está disponible en una terminal del microcontrolador con el fin de desacoplar la señal por medio de un capacitor y de esa manera obtener un mejor desempeño contra el ruido.

La palabra digital esperada a la salida del convertidor A/D está dada mediante la siguiente expresión:

$$CAD_{valor} = \frac{V_{ent} \cdot 1024}{V_{ref}} \quad (3.1)$$

donde:

- $CDA_{valor}$  es el valor de salida del convertidor A/D
- $V_{ent}$  es la tensión de la señal de entrada al convertidor
- $V_{ref}$  es la tensión de referencia usada para el convertidor A/D, que en nuestro caso tiene un valor de 2.56 [V]

Con un  $V_{ref}$  de 2.56 [V] y 10 bits de resolución, la resolución de salida del convertidor A/D en terminos de volts es de 2.5 [mV].

La velocidad de conversión del convertidor A/D está dada a partir de su frecuencia de reloj. Esta frecuencia se genera a través de escalar la frecuencia del reloj principal del microcontrolador. De acuerdo a las hojas de especificaciones del microcontrolador, para realizar una conversión a 10 bits el convertidor A/D requiere de una frecuencia de reloj que se encuentre entre 50 [kHz] y 200 [kHz]. En nuestro caso esta frecuencia es de 62.5 [kHz]<sup>1</sup>. El convertidor A/D del microcontrolador requiere de 14.5 ciclos entre conversión y conversión, por lo que la frecuencia de muestreo del convertidor A/D con esta configuración es de 4.310 [kHz].

<sup>1</sup> Considerando un reloj del sistema de 4 [MHz] y un preescalador de 64.

### 3.1.6. Unidad de transmisión/recepción síncrona/asíncrona (USART)

La USART es un dispositivo de comunicación *full-duplex*, muy flexible en cuanto a su configuración y uso. El módulo integrado en este microcontrolador posee las siguientes características:

- Posee líneas independientes de transmisión y recepción de datos
- Operación síncrona y asíncrona
- Soporta paquetes de cinco, seis, siete, ocho o nueve bits de datos y uno o dos bits de paro
- Generación y revisión de paridad par o impar por medio de hardware

El *datalogger* solamente emplea la USART para realizar una comunicación asíncrona. Utilizada de esta manera y con una configuración adecuada, los pulsos de salida de la USART son compatibles con el estándar de comunicación RS-232, aunque requieren de cierto acondicionamiento.

Un paquete de transmisión en modo asíncrono consiste en una palabra de datos con bits de sincronización (bits de inicio y paro) y opcionalmente un bit de paridad para la detección de error. La USART acepta todas las treinta combinaciones de los siguientes características como paquetes válidos:

- Un bit de inicio
- Cinco, seis, siete, ocho o nueve bits de datos
- Bit de paridad par, impar o ninguno
- Uno o dos bits de paro

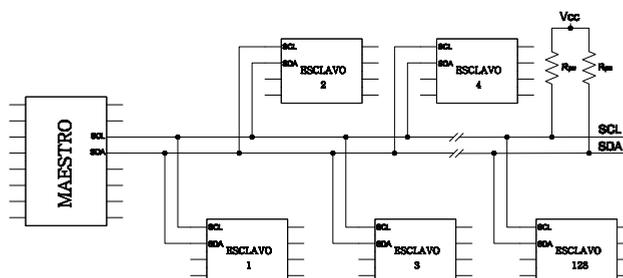
Un paquete de datos comienza con un bit de inicio, seguido por el bit menos significativo del dato, posteriormente le continúan los bits de datos hasta completar el número configurado, terminando con el bit más significativo. Si ha sido configurado, el bit de paridad es insertado después de los bits de datos y antes de los bits de paro, cuando una secuencia de datos es completamente transmitida puede inmediatamente seguir otra o la línea, puede quedar en estado inactivo (alto).

Ningún hardware externo es necesario para hacer funcionar este módulo y las líneas que se utilizan, la de transmisión (Tx) y recepción (Rx), se encuentran referenciadas a la tierra del circuito.

### 3.1.7. El módulo I<sup>2</sup>C

La interfaz de comunicación del I<sup>2</sup>C o *Two-Wire Interface* (TWI) es ideal para utilizarse en aplicaciones con microcontroladores, dado que con sólo dos líneas bidireccionales es posible comunicarse con hasta 128 dispositivos conectados al *bus*. Una de estas líneas es dedicada para la señal de reloj (SCL) y la otra para la señal de datos (SDA). El único hardware externo necesario para que el módulo de comunicación I<sup>2</sup>C funcione son dos resistencias de *pull-up*, una en cada línea del *bus*. Todos los dispositivos conectados al *bus* tienen direcciones independientes y el mecanismo para la conexión de los dispositivos es inherente al protocolo I<sup>2</sup>C.

En la figura 3.5 se muestra la conexión del módulo I<sup>2</sup>C a distintos dispositivos, se pueden apreciar las resistencias de *pull-up* necesarias para llevar a cabo la comunicación.



**Fig. 3.5.** Ejemplo de conexión del *bus* I<sup>2</sup>C.

El valor de las resistencias de *pull-up* dependen de la velocidad a la que se establece la comunicación I<sup>2</sup>C. Para una velocidad de comunicación de 100 [kHz], con la cual se configuró este módulo, un valor de resistencias de 10 [k $\Omega$ ] es adecuado<sup>2</sup>. La máxima capacitancia permitida para *bus* I<sup>2</sup>C es de 400 [pF] valor que limita tanto su longitud como el número de dispositivos que se pueden conectar a éste.

La terminología del protocolo I<sup>2</sup>C se refiere a un dispositivo MAESTRO como aquel dispositivo que inicia la comunicación en el *bus*, éste a su vez es el encargado de generar la señal de reloj. En el *bus* puede estar conectado más de un MAESTRO, pero sólo uno puede tomar el control del *bus* a la vez. Un dispositivo ESCLAVO es el dispositivo direccionado por el dispositivo MAESTRO. El TRASMISOR es el dispositivo que está poniendo información en el *bus* y el RECEPTOR es aquel dispositivo que recibe datos del *bus*.

<sup>2</sup> Microchip, *24LC1025 CMOS Serial EEPROM*. Hoja de datos del producto.

Cada bit de datos transmitidos en el *bus* del I<sup>2</sup>C es acompañado por un pulso en la línea del reloj. El nivel de la línea de datos debe de mantenerse estable cuando la línea de reloj esté en alto, la única excepción a esta regla es durante la generación de las señales de inicio y paro de la transmisión. Dichas operaciones las realiza un dispositivo MAESTRO, al cambiar el nivel de la línea de datos mientras la línea del reloj está en alto.

Un ejemplo de una comunicación típica de datos del protocolo I<sup>2</sup>C se puede apreciar en la figura 3.6. Una vez *iniciada la transmisión* por el MAESTRO, mediante una señal de inicio, manda un *paquete de direcciones* que consta de nueve bits, esto considerando siete bits de dirección, uno de lectura/escritura y uno de *acknowledge* (enterado)(ACK). Si el bit de lectura/escritura es un uno lógico, indica que el MAESTRO desea realizar una operación de lectura sobre el ESCLAVO direccionado, en caso contrario la operación requerida es de escritura. Cuando el dispositivo ESCLAVO reconoce que ha sido direccionado, éste debe de bajar la línea de datos en el bit nueve, si el dispositivo ESCLAVO está ocupado o por cualquier otra causa no puede atender el llamado del MAESTRO, éste debe dejar la línea datos en “alto” durante el noveno ciclo de reloj.

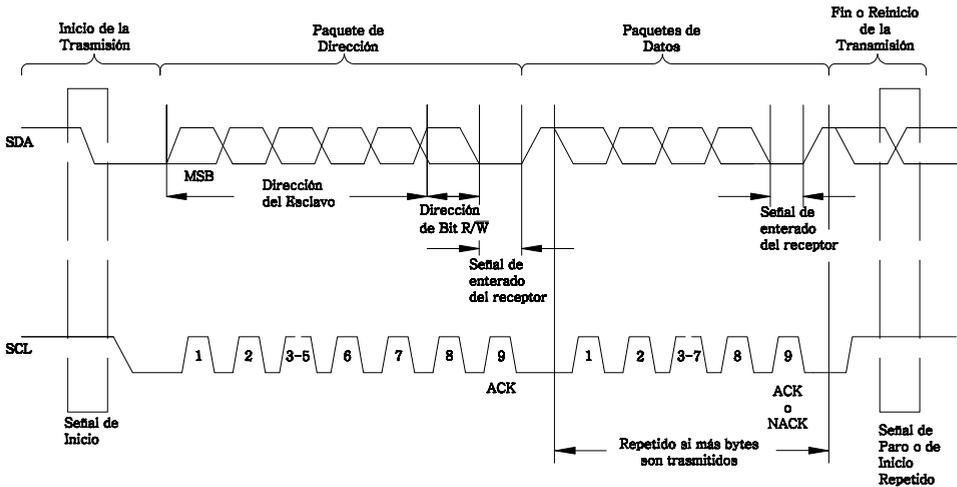


Fig. 3.6. Diagrama de tiempo del protocolo I<sup>2</sup>C.

Una vez que el paquete de dirección es reconocido por el ESCLAVO, el TRASMISOR envía los *paquetes de datos*. Todos los paquetes de datos transmitidos por el *bus* I<sup>2</sup>C son de nueve bits, considerando ocho bits de datos y un bit de enterado. El bit más significativo del byte de datos es transmitido primero. Durante

la transmisión de información el MAESTRO es el encargado de generar la señales de inicio y paro y la señal de reloj, mientras el RECEPTOR es el encargado de mandar las señales de enterado al final de cada paquete de información. Si el RECEPTOR mantiene la línea de datos “alta”, esto se considera una señal de *no acknowledge* (no enterado) (NACK). Cuando el RECEPTOR ha recibido el último bit o por alguna razón ya no le es posible recibir más bits, éste debe informárselo al TRASMISOR con una señal de NACK al final del último byte.

Si una señal NACK es recibida por el TRASMISOR, en caso de ser éste un dispositivo ESCLAVO, dejará de transmitir información o en caso de ser un dispositivo MAESTRO podrá enviar una señal de paro o un inicio repetido para reiniciar la transmisión. El *inicio repetido* ocurre cuando una nueva señal de inicio es hecha entre una condición de inicio y paro, y sucede cuando un MAESTRO desea realizar una nueva transmisión sin ceder el control de éste.

La *señal de paro* es transmitida por el MAESTRO cuando el intercambio de información ha terminado y éste desea liberar el *bus*. Entre una señal de inicio y una de paro el *bus* se considera ocupado y ningún otro MAESTRO puede intentar tomar el control de éste.

## 3.2. Adquisición de datos

La adquisición de datos es la parte más importante de un sistema de instrumentación, es aquí donde se determina gran parte de la funcionalidad y calidad del sistema de instrumentación. Un error en el diseño de esta parte conllevará a una salida llena de errores, que hará inútil todo el sistema en su conjunto. Es por ello que, un buen diseño debe comenzar por asegurar que las variables medidas están siendo adquiridas de manera adecuada y que éstas describen de la manera más próxima la variable a medir.

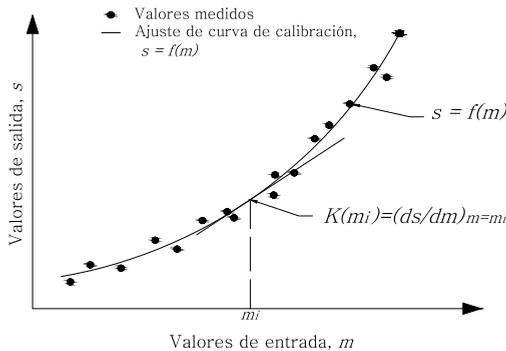
### 3.2.1. Sensores

Los sensores son el primer componente en la cadena de instrumentación. Estos convierten las variables físicas o químicas de un fenómeno en señales eléctricas, que casi siempre serán señales analógicas. Esta conversión debe de ser lo más fiel a la señal involucrada en la medición. Sólo un conocimiento minucioso de la respuesta de un sensor garantiza el éxito de una medición. En ocasiones los sensores producen señales corruptas debido a interferencia, las condiciones de uso o por el mismo proceso de medición.

Un sensor funciona convirtiendo una cantidad física medible ( $m$ ) en variable eléctrica ( $s$ ). Esta señal eléctrica puede ser impedancia, una carga eléctrica, una corriente o una diferencia de potencial. La relación entre  $s$  y  $m$  es una función  $s = F(m)$  y depende de:

- La ley física que determina el comportamiento del sensor
- La estructura interna del sensor
- El ambiente en el que se encuentra el elemento sensor del sensor

Mediante el uso de un estándar o unidades de medición, encontramos los valores de la señal eléctrica  $m$  ( $m_1, m_2...m_i...$ ) que son enviadas por el sensor en respuesta a la variable sensible ( $s_1, s_2...s_i...$ ). En una calibración el valor de entrada ( $m$ ) debe ser una variable controlada independiente, mientras que el valor de salida ( $s$ ) es la variable dependiente de la calibración. Una correlación tendrá la forma  $s = f(m)$  y se determina aplicando a la curva de calibración un razonamiento físico y técnicas de ajuste de curvas. En la figura 3.7 se muestra un ejemplo de la construcción de una curva de calibración, así como ejemplos de valores medidos que permitieron trazar esta curva. La correlación obtenida a partir de la calibración de un sensor, podrá posteriormente ser utilizada para encontrar el valor de entrada desconocido con base en el valor de salida indicado por el sistema de medición.



**Fig. 3.7.** Curva representativa de calibración.

Un parámetro importante de un sensor y que se obtiene a través de la curva de calibración es la sensibilidad. Llamaremos sensibilidad  $K$  a la razón de cambio de la señal de salida eléctrica del sensor  $m$  en relación con el cambio de la señal de entrada del sensor  $s$ , esta relación la denotaremos con una derivada:

$$K = K(m_i) = \left( \frac{ds}{dm} \right)_{s=s_1} \quad (3.2)$$

donde  $K$  es una función de  $m$ .

Puesto que las curvas de calibración pueden ser lineales o no lineales, según el sistema de medición y la variable que mide,  $K$  puede o no ser una constante en un intervalo de valores de entrada.

Como se mencionó en el capítulo introductorio, las variables medidas por el *datalogger* corresponden a las variables meteorológicas de temperatura ambiente y precipitación pluvial. Para realizar la medición de estas variables se eligieron, del universo de sensores disponibles para este fin, aquellos que se tenían disponibles y que adicionalmente estuvieran dentro de los recomendados por la Organización Meteorológica Mundial (OMM)<sup>3,4</sup>. Los sensores que se utilizaron fueron una sonda de temperatura basada en un termistor, para realizar la medición de la temperatura ambiente, y un pluviómetro de balancín, para la medición de la precipitación pluvial.

### ***La sonda de temperatura***

El sensor elegido para realizar las mediciones de temperatura ambiente es la sonda de temperatura 107 fabricado por la compañía Campbell Scientific. Una fotografía de la sonda 107 utilizada se muestra en la figura 3.8.



**Fig. 3.8.** Fotografía de la sonda de temperatura 107 usada.

La sonda tiene como características generales las siguientes:

- Es capaz de realizar mediciones de temperatura en suelo, agua y aire

<sup>3</sup> WMO, *Guide to Meteorological Instruments and Methods of Observation*, pág. 1.2-12.

<sup>4</sup> *ibid*, pág. 1.6-10.

- Su margen de medición es de los  $-35$  [°C] a los  $+50$  [°C]
- Los errores de precisión de la sonda se encuentran de  $\pm 0.4$  [°C], en el margen de los  $-24$  [°C] a los  $48$  [°C], y  $\pm 0.9$  [°C], en el margen de los  $-38$  a los  $53$  [°C]
- Posee una constante de tiempo entre 30 y 60 segundos para una velocidad del viento de  $5$  [ms<sup>-1</sup>]

El sensor de la sonda de temperatura es un termistor tipo *Negative Temperature Coefficient* (NTC). Un termistor es un sensor de temperatura el cual varía su resistencia de acuerdo a su temperatura, el término NTC se refiere a que es de coeficiente negativo, por lo cual su resistencia disminuye conforme su temperatura aumenta.

La exactitud general de la sonda está determinada por una combinación de las especificaciones de intercambiabilidad, la precisión del puente de resistencias y el error de la ecuación Steinhart-Hart. El mayor error es debido a las especificaciones de intercambiabilidad del termistor. En el margen de  $0$  [°C] a  $50$  [°C] el error de intercambiabilidad es mayoritariamente un *offset* que puede ser corregido con una calibración de un solo punto<sup>5</sup>.

La estructura interna de esta sonda se observa en la figura 3.9. Como se puede observar la resistencia  $R_1$  junto con el termistor forman un divisor de tensión con la resistencia  $R_2$ . El divisor está formado por resistencias de una tolerancia de 0.1%, con un coeficiente de temperatura de  $10$  [ppm/°C].

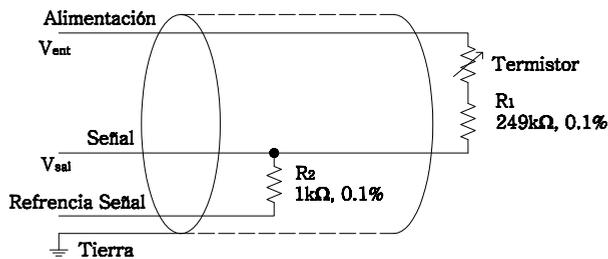


Fig. 3.9. Estructura de la sonda de temperatura 107.

La ecuación que determina el valor de tensión a la salida de la sonda está dado por:

$$V_{sal} = \frac{1000V_{ent}}{R_{termistor} + 250000} [V] \quad (3.3)$$

<sup>5</sup> Capbell Scientific, Inc., *User Manual Model 107 Temperature Probe*, pág. 2.

Para determinar la relación de resistencia del termistor ( $R_{termistor}$ ) con su temperatura es necesario utilizar la ecuación de Steinhart–Hart, cuya expresión se muestra en la ecuación 3.4.

$$\frac{1}{T} = A + B \log R + C(\log R)^3 \quad (3.4)$$

donde:

- $T$  es la temperatura en Kelvins
- $R$  es la resistencia a la temperatura  $T$  en Ohms y
- $A, B, C$  son los coeficientes de la ecuación de Steinhart-Hart que varían de acuerdo al modelo del termistor y el margen de temperatura de interés

Para este modelo de termistor los coeficientes  $A, B$  y  $C$  tienen los siguientes valores<sup>6</sup>:

$$\begin{aligned} A &= 8.271111 \times 10^{-4} \\ B &= 2.088020 \times 10^{-4} \\ C &= 8.059200 \times 10^{-8} \end{aligned}$$

Para que por medio de las ecuaciones 3.3 y 3.4 se puede determinar la tensión de salida a determinada temperatura de la sonda, es necesario solamente fijar una tensión de alimentación  $V_{ent}$ . Esta tensión de alimentación debe de ser una tensión constante y estable en todo momento, para asegurar obtener una medición libre de errores. En nuestro caso esta tensión es de 2.56 [V] y es generada internamente por el microcontrolador como un voltaje de referencia para el convertidor A/D.

Finalmente, con fines de caracterizar la sonda de temperatura, se puede utilizar la ecuación inversa de Steinhart–Hart, que se muestra en la ecuación 3.5, para obtener la resistencia del termistor dada una temperatura de entrada<sup>7</sup>.

$$R = \exp \left( \sqrt[3]{x - \frac{y}{2}} - \sqrt[3]{x + \frac{y}{2}} \right) \quad (3.5)$$

donde:

<sup>6</sup> ibid, pág. 9.

<sup>7</sup> **J. S. Steinhart and S. R. Hart**, *Calibration curves for thermistors*, Deep-Sea Res., 15:497, 1968.

$$y = \frac{A - \frac{1}{T}}{C}$$

$$x = \sqrt{\left(\frac{B}{3C}\right)^3 + \frac{T^2}{4}}$$

### El pluviómetro

El pluviómetro es un tipo de instrumento usado por los meteorólogos e hidrólogos para acumular y medir la cantidad de líquido precipitado en un determinado período de tiempo. La mayor parte de los pluviómetros tienen como unidad de medida los milímetros. Una precipitación de 5 [mm] indica que si toda el agua se acumulada en un terreno plano y sin escurrir de un metro cuadrado, la altura de la capa de agua sería de 5 [mm]. Por lo que los milímetros son equivalentes a litros por metro cuadrado. Para este proyecto se utilizó un pluviómetro de balancín, dado que su salida es bastante adecuada para la toma de mediciones por medio de un *datalogger*.

Los pluviómetros de balancín o de volcado consisten en un gran cilindro de cobre u otro material sujeto al suelo, en la parte de arriba de dicho cilindro se coloca un embudo que colecta y dirige la precipitación. La precipitación cae en unas pequeños cubos que se encuentran balanceados a un columpio o balancín. Estos cubos están calibrados para que después de que cierta cantidad de lluvia vuelquen, esta volcadura cierra momentáneamente un interruptor que sirve para poder detectar si un evento ocurrió. Una fotografía del pluviómetro utilizado se muestra en la figura 3.10.



**Fig. 3.10.** Fotografía del pluviómetro usado.

Para detectar el evento se introduce una resistencia conectada en un extremo a una fuente de tensión y en el otro extremo a una terminal del interruptor, en el momento en que ocurra un volcado, el circuito se cierra y la tensión de entrada se muestra en la terminal de salida del interruptor. Realizando el

procedimiento anterior, el pluviómetro de balancín se convierte en un sensor que convierte una señal mecánica a una señal eléctrica, la cual puede ser registrada.

La desventaja de este tipo de pluviómetros es que suelen no ser tan precisos como los pluviómetros estándar, dado que la precipitación puede terminar antes de que el cubo tenga la cantidad necesaria para volcar, de esta manera cuando otro evento de precipitación comience puede sólo requerir de unas cuantas gotas para volcar el balancín y se registre un evento. También tienden a subestimar la cantidad de precipitación, particularmente en eventos con granizo y nieve.

### 3.2.2. Caracterización de la señal de los sensores

Por caracterización de la señal de los sensores nos referimos a determinar la salida esperada de los sensores, una vez que éstos son instalados para medir las variables de interés. Esta información nos será de importancia para diseñar el acondicionamiento requerido por las señales antes de ser digitalizada.

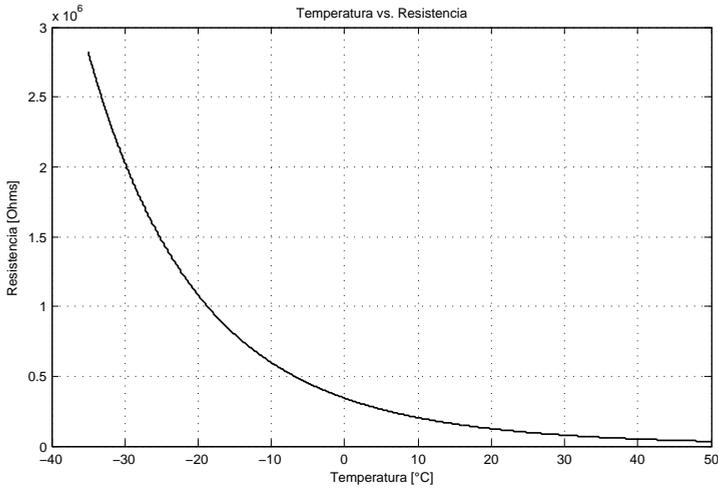
#### *La sonda de temperatura*

En el caso de la sonda de temperatura es necesario especificar el margen de mediciones en que la sonda estará sometida para de ahí determinar su salida.

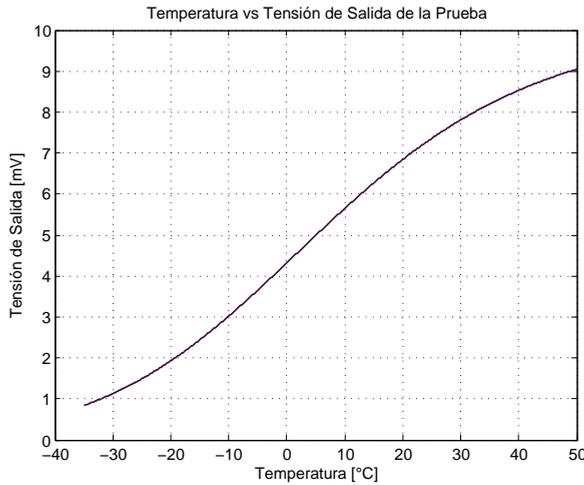
Para nuestra aplicación decidimos aprovechar todo el potencial de la sonda de temperatura y utilizar el margen de mediciones que le es posible realizar, es decir de los  $-35$  [°C] a los  $50$  [°C]. Utilizando la ecuación 3.5 obtenemos los valores de resistencia esperados para la sonda en el margen de temperatura mencionado, dichos valores se graficaron y se muestran en la figura 3.11.

Como se puede observar en la gráfica, la ecuación 3.5 describe un comportamiento no lineal sobre el margen de temperaturas medidos. Para el valor mínimo de temperatura ( $-35$  [°C]) tenemos una resistencia del termistor de  $2.826$  [M $\Omega$ ] y para el valor máximo ( $50$  [°C]) de  $33.124$  [k $\Omega$ ]. Ahora bien, mediante la ecuación 3.3 se determina la tensión de salida esperada por la sonda de temperatura en el margen de medición. Esta respuesta se puede apreciar en la gráfica de la figura 3.12. Se determinó que los valores de salida de la sonda van del margen de los  $0.833$  [mV] a los  $9.0420$  [mV] para los valores de temperatura mínimo y máximo respectivamente.

La variación de la señal de temperatura está dada por la constante de tiempo de la sonda. Este valor es el tiempo necesario para que la sonda de temperatura refleje a la salida el 63% de una nueva temperatura. En el caso



**Fig. 3.11.** Gráfica de la resistencia del termistor respecto a su temperatura para la sonda 107.



**Fig. 3.12.** Tensión de salida contra temperatura de la sonda 107.

de la sonda utilizada, este valor se encuentra entre los 30 y 60 segundos. Para obtener la frecuencia en la que esperamos varíe la señal de temperatura obtenemos el inverso de la constante de tiempo de la sonda, por lo tanto, la frecuencia de la señal de temperatura se encuentra entre los 0.033 [Hz] a los 0.017 [Hz]. Se considerará el valor más bajo de 0.017 [Hz].

En resumen, a la salida de la sonda de temperatura se espera una señal de tensión que se encuentra entre los 0.833 [mV] a los 9.0420 [mV] con una frecuencia de 0.017 [Hz].

### ***El pluviómetro de balancín***

El tipo de salida esperado para el pluviómetro depende totalmente de la entrada suministrada al interruptor del pluviómetro. Dado que se espera alimentar al circuito con una tensión de 3.3 [V], ésta será la tensión de salida del interruptor. En cuanto a la frecuencia, ésta no es constante y depende totalmente de la medición, de hecho la medición que se va a realizar del pluviómetro se considera un conteo de pulsos.

Una vez caracterizadas las señales se hablará del sistema de adquisición de las mismas.

### **3.2.3. Acondicionamiento de la señales de los sensores**

El acondicionamiento de las señales de los sensores es el proceso por el cual se transforma la señal de estos para poderla hacer compatible con la entrada de los sistemas de adquisición. Este acondicionamiento es necesario para obtener una medición confiable de las señales de entrada.

### ***La sonda de temperatura***

Las características de la salida de la sonda de temperatura, como ya se vio, son las siguientes: una señal analógica que varía de los 0.833 [mV] a los 9.0420 [mV] con una frecuencia de 0.017 [Hz].

Según la configuración realizada para el convertidor A/D del microcontrolador, a su entrada éste acepta señales desde 0 [V] hasta los 2.56 [V] con una resolución de  $2.5 \left[ \frac{\text{mV}}{\text{Bit}} \right]$  y muestrea las señales a 4.310 [kHz].

Si bien se puede apreciar que la frecuencia de muestreo del convertidor A/D es adecuada para la adquisición de la señal de temperatura, la tensión de salida de la misma no lo es, debido a que al ser digitalizada por el convertidor A/D se tendría una resolución pésima, en términos de  $\frac{^{\circ}\text{C}}{\text{Bit}}$ .

Para hacer frente a lo anterior se requiere de amplificar la señal de salida de la sonda de temperatura. Con el fin de tratar de obtener una resolución máxima, la amplificación debe de ser igual a la relación  $\frac{2.56[\text{V}]}{9.20268[\text{mV}]}$  la cual

arroja una amplificación de 283.12 que es redondea a 283. Con esta amplificación, se obtiene una señal de la sonda de temperatura que varía de los 0.2357 [V] a los 2.559 [V].

Empleando el método anterior, existe una pérdida de resolución partiendo del hecho de que aunque es posible digitalizar valores por abajo de 0.2757 [V], éstos nunca estarán presentes a la entrada del convertidor A/D. Una mejor aproximación consideraría en restar el *offset* de la señal de temperatura antes de amplificarla, para asegurar que el margen de entrada de tensiones que representan una temperatura varíe desde 0 [V] hasta 2.56[V]. En este proyecto no se consideró realizar esta adecuación dado que, como se verá más adelante, no ofrece mejoras significativas en la resolución de salida del *datalogger* y representa el uso de un mayor número de componentes.

La ganancia calculada se implementó mediante el circuito amplificador no inversor mostrado en la figura 2.10, donde la ganancia de éste está dada a partir de la ecuación 2.16. Proponiendo un valor de resistencia para  $R_1$  de 220 [k $\Omega$ ] y una ganancia (G) de 283, el valor de  $R_2$  se calcula mediante la ecuación 3.6.

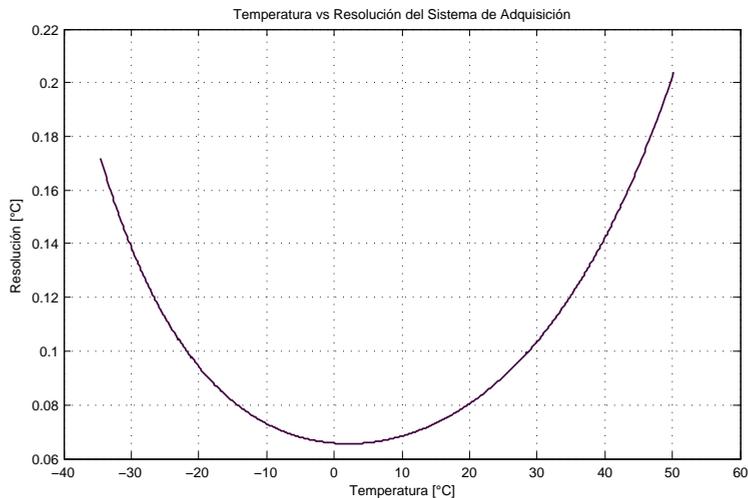
$$R_2 = \frac{R_1}{G - 1} = \frac{220[\text{k}\Omega]}{283 - 1} = 780.14[\Omega] \quad (3.6)$$

Dado que 780.14 [ $\Omega$ ] no es un valor comercial de resistencia, se selecciona una resistencia de 680 [ $\Omega$ ] más un potenciómetro de precisión de 220 [ $\Omega$ ], lo cual nos dará un margen de ganancias desde los 324.52 a los 244.44. El uso de un potenciómetro permitirá hacer un ajuste fino de la ganancia de la señal de temperatura.

Una vez que se ha establecido la ganancia y el procedimiento para implementarse, es necesario conocer cuál será la resolución de salida del sistema después de esta amplificación. Con una ganancia de la señal de 283, el intervalo de valores binarios de salida posibles del convertidor A/D serían de  $5E_H$  (0.2357 [V]) a  $3FF_H$  (2.559 [V]). Utilizando este intervalo, obtendremos la resolución de salida en términos de temperatura una vez hecha la digitalización, para ello seguiremos el siguiente procedimiento.

- Primeramente, del intervalo de valores esperados por el convertidor A/D, obtenemos los valores de tensión representado por éstos mediante la ecuación 3.1.
- Los valores de tensión obtenidos se dividen entre la amplificación dada a la señal (en nuestro caso 283).

- Se le aplica la ecuación 3.3, que relaciona la salida de la sonda de temperatura con respecto a la resistencia del termistor, para calcular el valor de resistencia de ésta en los valores discretos de salida del convertidor A/D.
- Mediante la ecuación de Steinhart–Hart (ecuación 3.4), se obtienen los valores de temperatura de los valores de resistencia calculados en el punto anterior.
- Finalmente, se obtiene la diferencia entre los valores continuos de temperatura. Esta diferencia es la resolución del sistema de instrumentación y su comportamiento se observa en la figura 3.13.



**Fig. 3.13.** Gráfica de la resolución de salida esperada.

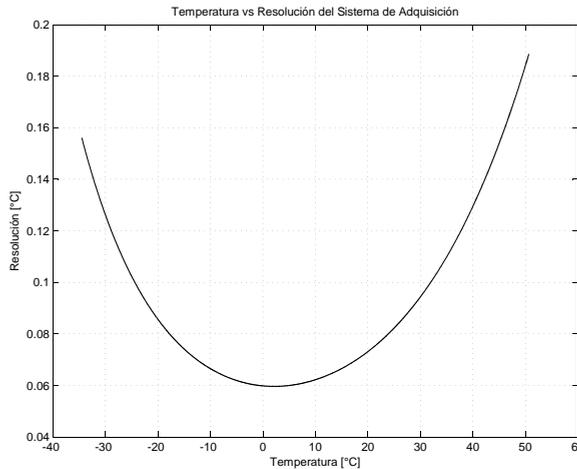
Como se puede observar en la gráfica, la resolución esperada por el sistema no es lineal debido a que la respuesta de la sonda de temperatura usada no lo es. La resolución promedio en el intervalo es de  $0.0914$  [°C], con una resolución máxima de  $0.2038$  en los  $50$  [°C] y una mínima de  $0.0655$  [°C] en el intervalo de los  $1.9578$  [°C] a los  $2.2856$  [°C].

En caso de haber implementado un acondicionamiento para eliminar el *offset* de la sonda de temperatura, el procedimiento que se hubiera seguido para obtener la resolución esperada por el *datalogger* sería el siguiente.

- Partiendo de la tensión de salida esperada por la sonda para el rango de temperatura medible por ésta ( $0.833$  [mV] y  $9.0420$  [mV] para  $-35$  [°C] y  $50$  [°C] respectivamente), eliminar mediante un circuito restador el *offset* de  $0.833$  [mV].

- Del nuevo margen de salida obtenido 0 [mV] para -35 [°C] y 8.209 [mV] para 50 [°C], calcular la ganancia más adecuada para cubrir el intervalo. Esta ganancia está dada por la relación  $\frac{2.56[V]}{8.209[mV]}$ .
- La ganancia resultante de 311.85 puede ser redondeada a 311 y con ésta obtener el margen de tensión esperado a la entrada del convertidor A/D, una vez que la señal fuera amplificada. Este margen de tensión sería de los 0 [V] a los 2.55 [V].
- Una vez obtenido este margen se puede utilizar el procedimiento anteriormente descrito para obtener la resolución esperada por el *datalogger*, a este procedimiento sólo se le debe hacer la adecuación para que, una vez recuperada la señal de la sonda, sumarle el *offset* eliminado.

La gráfica que muestra la resolución que se hubiera esperado obtener por medio del *datalogger*, si el *offset* de la señal de temperatura hubiera sido eliminado, se puede ver en la figura 3.14.



**Fig. 3.14.** Gráfica de la resolución de salida esperada sin *offset*.

Eliminando el *offset* de la señal de temperatura el valor de resolución promedio es de 0.834 [°C], la resolución máxima es de 0.1886 [°C] a los 50 [°C] y una mínima de 0.0596 [°C] en el intervalo entre 1.8366 [°C] y los 2.3774 [°C]. Aunque se aprecia que se puede obtener una mejora en resolución del *datalogger*, en caso de que el *offset* sea eliminado, debido a que este acondicionamiento requiere de un mayor número de componentes y que la ganancia de resolución no es significativa, se decidió no implementarla.

Una vez especificado el método para adecuar la amplitud de la salida de la sonda, adicionalmente se requiere eliminar el ruido de la señal para disminuir el efecto *aliasing* una vez que esta señal sea digitalizada. Para realizar lo anterior se propuso el diseño de un filtro paso bajas con aproximación de Butterworth como filtro *antialiasing*.

Los parámetros del filtro se consideran de la siguiente forma:

$$\begin{aligned} A_p &= 1 \text{ [dB]} \\ f_s &= 2.05 \text{ [kHz]} \\ A_s &= 62 \text{ [dB]} \\ f_p &= 10 \text{ [Hz]} \end{aligned}$$

La frecuencia de supresión del filtro es igual a la mitad de la frecuencia de muestreo del convertidor A/D (4.10 [kHz]). Para determinar la atenuación requerida en la banda de rechazo del filtro hacemos uso de la razón señal a ruido del convertidor A/D que se calcula a través de la ecuación 2.3, haciendo B igual a 10, el número de bits del convertidor A/D usado.

$$SNR = (1.763 + 6.02(10))[dB] = 62[dB]$$

Finalmente se propone que la atenuación en la banda de paso sea igual a 1 [dB] y dado que la señal de temperatura es basicamente DC se propone una frecuencia de corte del filtro sea de 10 [Hz].

Siguiendo el proceso para el diseño de un filtro Butterworth descrito en el capítulo anterior, primeramente calculamos el valor de  $\epsilon_1$  con la ecuación 2.10.

$$\epsilon_1 = \sqrt{10^{A_{m\acute{a}x}/10} - 1} = \sqrt{10^{0.1(1[ \text{dB}])} - 1} = 0.5088$$

Con la ecuación 2.11 calculamos  $\epsilon_2$  para la amplitud de la banda de supresión.

$$\epsilon_2 = \sqrt{10^{A_{m\acute{i}n}/10} - 1} = \sqrt{10^{0.1(62[ \text{dB}])} - 1} = 1258.92$$

Por medio de la ecuación 2.9 obtenemos el orden del filtro Butterworth.

$$n_B = \frac{\log \epsilon_2 / \epsilon_1}{\log f_s / f_p} = \frac{\log 1258.92 / 0.5088}{\log 2050 / 10} = 1.47$$

Empleando el valor entero más próximo hacia arriba usaremos  $n_B = 2$ . Una vez conocido el orden del filtro, utilizamos la figura 2.5 para determinar los modos naturales del filtro. Encontramos que estos son:

$$\begin{aligned} p_1 &= \omega_0(-\cos 45^\circ + j \sin 45^\circ) = \omega_0(-0.7071 + j0.7071) \\ p_2 &= \omega_0(-\cos 45^\circ - j \sin 45^\circ) = \omega_0(-0.7071 - j0.7071) \end{aligned}$$

Finalmente combinando ambos polos en la expresión del filtro (ecuación 2.12) y haciendo la ganancia ( $K$ ) unitaria obtenemos:

$$H(s) = \frac{\omega_0^N}{s^2 + 1.4242\omega_0 + \omega_0^2}$$

donde:

$$\omega_0 = \omega_p \left( \frac{1}{\epsilon_1} \right)^{1/N} = 2\pi \times 10 \text{ Hz} \times \left( \frac{1}{0.5088} \right)^{0.5} = 88.086$$

Con lo que finalmente, la función de transferencia del filtro queda como:

$$H(s) = \frac{7759.143}{s^2 + 125.452s + 7759.143} \quad (3.7)$$

Una vez obtenida la función de transferencia del filtro, se procede con su implementación a través de una topología Sallen-Key que se mostró en el capítulo anterior.

Partiendo de la función de transferencia del circuito (ecuación 2.15), se reescribe de la siguiente forma:

$$H(s) = \frac{1}{1 + RC(m+1)s + mnR^2C^2s^2} \quad (3.8)$$

donde:

$$\begin{aligned} R_1 &= mR \\ R_2 &= R \\ C_1 &= nC \\ C_2 &= C \end{aligned}$$

De la ecuación 3.7 se normaliza para dejarla darle la forma de la ecuación 3.8

$$H(s) = \frac{1}{1.2888 \times 10^{-4}s^2 + 0.0167s + 1} \quad (3.9)$$

Igualando las ecuaciones obtenemos que el término  $RC(m+1) = 0.0167$ , donde proponiendo  $R = 4.7[\text{k}\Omega]$  y  $C = 2.2[\mu\text{F}]$  obtenemos  $m$ :

$$m = \frac{0.01617}{(4.7[\text{k}\Omega])(2.2[\mu\text{F}])} - 1 = 0.5638$$

a partir del valor de  $m$  obtenemos  $n$ , partiendo de igualar el término  $mnR^2C^2 = 1.2888 \times 10^{-4}$

$$n = \frac{1.2888 \times 10^{-4}}{mR^2C^2} = \frac{1.2888 \times 10^{-4}}{0.5638 \times (4700[\Omega])^2 \times (2.2[\mu\text{F}])^2} = 2.1381$$

Por último a partir del valor de  $n$  y  $m$  calculamos  $C_1$  y  $R_1$ :

$$C_1 = nC = (2.1381)(2.2[\mu\text{F}]) = 4.703[\mu\text{F}]$$

$$R_1 = mR = (0.5638)(4.7[\text{k}\Omega]) = 2.649[\text{k}\Omega]$$

Con lo que obtenemos todos los valores de los componentes del filtro Sallen-Key. Ajustando estos valores a componentes comerciales finalmente obtenemos que:

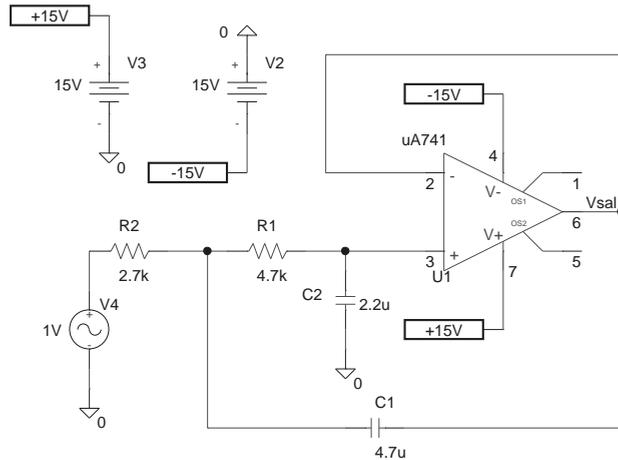
$$C_1 = 4.7 [\mu\text{F}]$$

$$R_1 = 2.7 [\text{k}\Omega]$$

$$C_2 = 2.2 [\mu\text{F}]$$

$$R_2 = 4.7 [\text{k}\Omega]$$

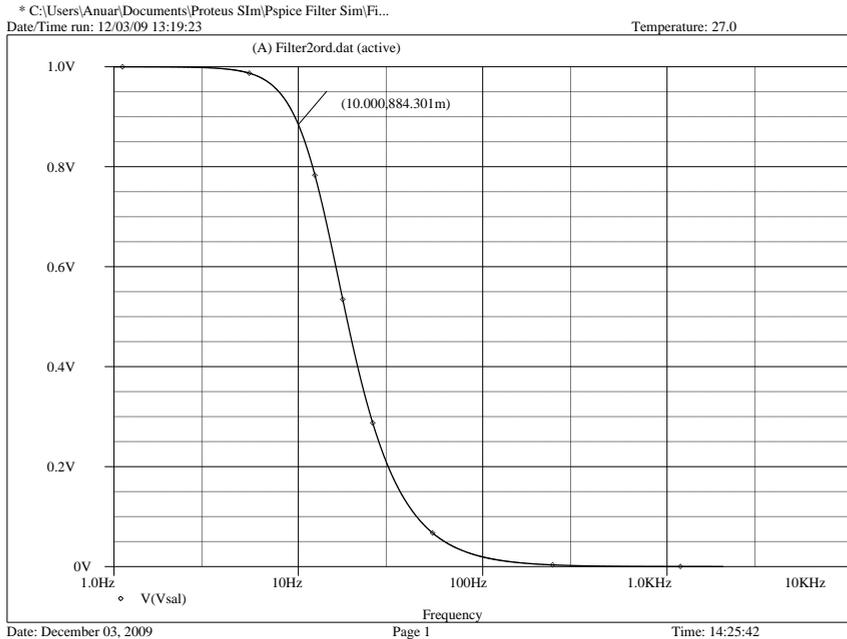
El circuito completo, implementado con un amplificador operacional uA741 y componentes con valores comerciales, se puede apreciar en la figura 3.15.



**Fig. 3.15.** Implementación del filtro Butterworth.

Aunque el amplificador uA741 no se utilizó en la implementación del filtro, dado que es el único amplificador disponible en el simulador usado y con el fin de dar una idea de la respuesta del filtro, este circuito se utilizó para hacer una simulación con barrido de A/C, dando como resultado la gráfica de la figura 3.16. Esta gráfica muestra la variación de la amplitud de la señal

de salida del filtro a diferentes frecuencias. Podemos apreciar en la simulación que a la frecuencia de corte de 10 [Hz] el filtro atenúa la señal de entrada 1.068 [dB] ( $20 \log \frac{0.884[V]}{1[V]}$ ), esta atenuación tiene un error de 6.8% con respecto a la atenuación definida, pero, a pesar de ello, se considera que el filtro es adecuado para ser implementado.



**Fig. 3.16.** Simulación del filtro Butterworth.

Como ya se había mencionado, la alimentación de la sonda de temperatura se realizó a través de una tensión de referencia. En nuestro caso se empleó la tensión de referencia interna del microcontrolador. Dado que esta salida no está diseñada para alimentar cargas, se empleó un amplificador operacional en configuración *buffer* para darle “potencia” y de esa manera poder alimentar la sonda de temperatura. La corriente necesaria a la salida del amplificador se encuentra dado por la carga que representa la sonda de temperatura, dicha carga está dada por el arreglo resistivo de la sonda de temperatura. En el peor de los casos la resistencia total de la sonda es de un poco menos de 1 [kΩ], siendo alimentado con una fuente de tensión de 2.56 [V], se obtiene que la corriente que circula por ella es de aproximadamente 2.56 [mA].

La selección del amplificador operacional debe contemplar características para implementar tanto el filtro, el circuito *buffer* y el circuito amplificador, para ello tomaremos los siguientes parámetros de selección:

- El tipo de alimentación
- El *gain bandwidth product* (GBWP)
- El *slew rate*
- La máxima corriente de salida

Dado que no se cuenta con una alimentación negativa en el circuito, es necesario considerar un amplificador con alimentación unipolar, lo cual se convierte en el primer parámetro de selección. Adicionalmente ya hemos calculado que la corriente mínima de salida que debe proporcionar el amplificador es de 2.56 [mA]. El GBWP necesario para un filtro con ganancia unitaria se encuentra dado por la ecuación 3.10

$$\text{GBWP}_{\min} = 100f_c \quad (3.10)$$

donde  $f_c$  es la frecuencia de corte del filtro en nuestro caso 13.7 [Hz], por lo que se requiere de un amplificador cuya GBWP es mayor o igual a:

$$\text{GBWP}_{\min} = 100 \times 13.7[\text{ Hz}] = 1.37[\text{ kHz}]$$

El *slew rate* que el amplificador debe poseer está dado por la ecuación 3.11

$$\text{Slew Rate} \geq 2\pi V_{\text{salPP}} f_c \quad (3.11)$$

donde  $f_c$  es la frecuencia de corte del amplificador y  $V_{\text{salPP}}$  es la variación pico a pico esperada a la salida del amplificador abajo de  $f_c$ . En nuestro caso esta variación es mínima, dado que la razón de cambio de la señal de temperatura es muy bajo, pero utilizaremos 0.1 [V] para realizar el cálculo. Tomando esta consideración el *slew rate* requerido es igual a:

$$\text{Slew Rate} \geq 2\pi \times 0.1[\text{V}] \times 13.7[\text{ Hz}] = 8.6[\text{ V/s}]$$

Como podemos ver, dado que la frecuencia de corte es muy pequeña y la potencia de salida necesaria también es pequeña, no se requiere un amplificador operación con características muy específicas y podría haberse usado para su implementación un amplificador operacional de propósito general. Sin embargo, dada que la señal proveniente de la sonda de temperatura está muy cercana al nivel de tierra, nivel con el que se pretende polarizar el amplificador operacional, fue necesario el uso de un amplificador operacional con salidas y

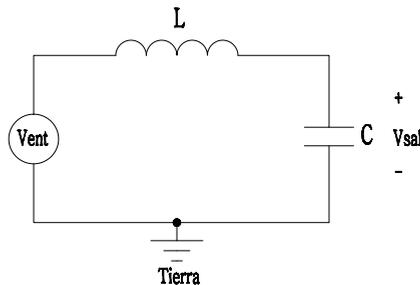
entradas *rail to rail*. Esta característica permite que tanto las entradas como las salidas de este amplificador puedan tener niveles muy cercanos a las tensiones de polarización del amplificador operacional. Dado que se requieren de tres amplificadores operacionales para realizar la etapa de acondicionamiento de la sonda de temperatura, se utilizaron el circuito integrado MAX4168 que incorpora dos amplificadores operacionales y el circuito integrado MAX4166 que incorpora uno. Adicionalmente a su alimentación unipolar y salidas y entradas *rail to rail*, los amplificadores operacionales en estos circuitos integrados tienen las siguientes características:

$$\begin{aligned} SlewRate &= 2 \text{ [V}/\mu\text{s]} \\ GBWP &= 5 \text{ [MHz]} \\ I_{sal \text{ máx}} &= 80 \text{ [mA]} \end{aligned}$$

A simple vista se puede apreciar que las características de los amplificadores superan los requerimientos de los circuitos por lo que se concluyó que el amplificador es adecuado.

Con el fin de asegurar un bajo consumo de energía del *datalogger* y evitar el ruido proveniente de los circuitos digitales, se realizaron consideraciones para la alimentación del amplificador operacional:

En primer lugar se realizó un circuito de tierra independiente y se unió con la tierra digital en un solo punto, además la alimentación se filtró mediante un filtro LC paso bajas pasivo. Este circuito se muestra en la figura 3.17.



**Fig. 3.17.** Filtro pasivo LC.

Este filtro tiene una frecuencia de corte dada por la ecuación ecuación 3.12.

$$f_0 = \frac{1}{2\pi \times \sqrt{LC}} \quad (3.12)$$

Sustituyendo los valores de  $L = 10[\mu\text{F}]$  y  $C = 100[\text{nF}]$ , con los que el fabricante del microcontrolador sugiere implementar este filtro, encontramos que la frecuencia de corte es de:

$$f_0 = \frac{1}{2\pi \times \sqrt{10[\mu\text{F}] \times 100[\text{nF}]}]} = 159.15[\text{kHz}]$$

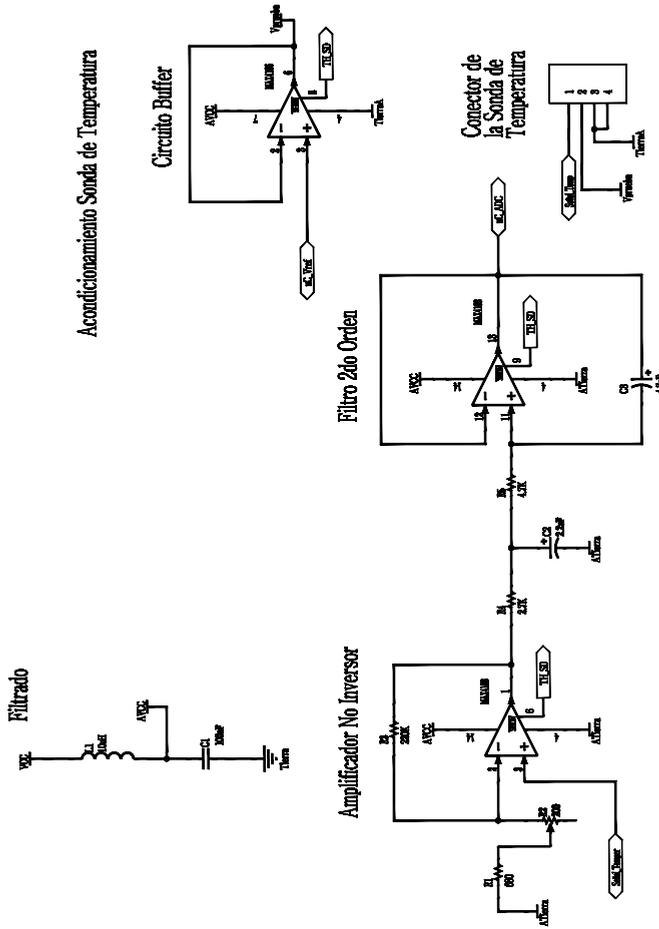
Dado que es un filtro que está diseñado para eliminar el ruido proveniente de los circuitos digitales con una frecuencia, en nuestro caso, de 4 [MHz] se determina, que si se considera una atenuación de 40 [dB/decada] para este filtro, éste es capaz de atenuar una señal de 4 [MHz] 56 [dB] lo cual se considera adecuado para la aplicación.

Adicionalmente, dado que durante el modo de adquisición del *datalogger* es necesario que el circuito de acondicionamiento de la sonda de temperatura opere en todo momento y con el fin de disminuir el consumo que éste representa, se decidió hacer uso de la terminal de encendido y apagado que los amplificadores operacionales usados incorporan. Haciendo uso de esta terminal se puede disminuir el consumo de energía de cada amplificador de 1.2 [mA] a 38 [μA].

Los circuitos involucrados en el acondicionamiento de la señal proveniente de la sonda de temperatura se muestran en la figura 3.18. Como se observa, la conexión de la sonda de temperatura se realiza a través de un conector de donde se alimenta (*Vprueba*) y se obtienen la señal de temperatura (*Señal\_Temp*). La alimentación se realiza a través del circuito *buffer* implementado con el amplificador operacional MAX4166 y que tiene por entrada la tensión de referencia del microcontrolador (*UC\_vref*). Posteriormente, la señal de temperatura se amplifica mediante el circuito amplificador no inversor y se filtra mediante el filtro Butterworth de segundo orden, ambos circuitos son implementados mediante el MAX4168. Después de este proceso la señal se encuentra lista para ser adquirida, por lo que se conecta a través del puerto *uC\_ADC* a la terminal del convertidor A/D del microcontrolador. Las terminales de control de encendido y apagado de los amplificadores operacionales son unidas y llevadas a una terminal del microcontrolador *TH\_SD*.

### ***El pluviómetro***

La señal proveniente del pluviómetro debe ser acondicionada, para evitar posibles problemas con el ruido de la señal que pudieran ocasionar que el microcontrolador detecte pulsos en falso o que, una vez que éstos se generen, sean detectados en más de una ocasión. Otro efecto que se desea evitar es que una tensión que provenga de la línea de señal del pluviómetro sea mayor a la que el sistema pueda manejar, por lo que está es también otra consideración de diseño. Si bien el efecto de rebote mecánico no es un problema grave para



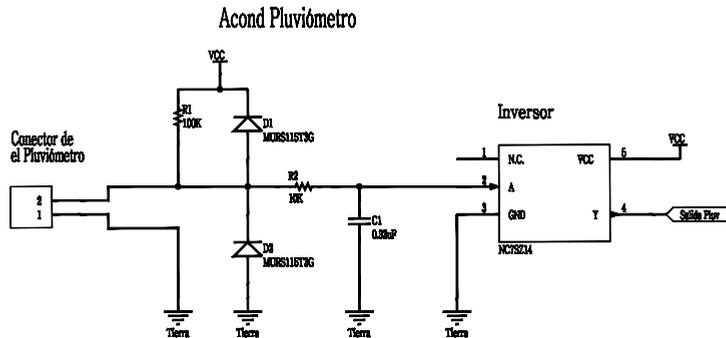
**Fig. 3.18.** Circuito de acondicionamiento de la señal de temperatura.

este tipo de pluviómetro, dado que su interruptor es de mercurio, se deben de considerar otras fuentes de ruido, como el ambiental.

El circuito que se diseñó para disminuir tanto el problema de ruido en la señal como de sobretensiones de la línea, es el mostrado en la figura 3.19.

En el circuito de la figura 3.19, un filtro pasivo RC paso bajas es implementado con la resistencia  $R_2$  y el capacitor  $C_1$ , la frecuencia de corte del filtro RC pasivo se encuentra dado por la ecuación 3.13.

$$f_0 = \frac{1}{2\pi RC} \tag{3.13}$$



**Fig. 3.19.** Circuito de acondicionamiento de la señal del pluviómetro

Sustituyendo el valor de los componentes usados obtenemos que la frecuencia de corte, para el filtro implementado es igual a:

$$f_0 = \frac{1}{2\pi \times 10[\text{ k}\Omega] \times 0.33[\text{ }\mu\text{F}]} = 48.23[\text{ Hz}]$$

La compuerta inversora con Schmitt Trigger NC7SZ14 posee, a una temperatura de operación de 25 [°C] y una tensión de 3.0 [V], con las siguientes características: una tensión de umbral positivo de 1.75 [V], una tensión de umbral negativo de 1 [V] y una tensión de histéresis de 0.75 [V].

Combinados el filtro RC y la compuerta inversora NC7SZ14 ayudan a evitar el ruido presente en la línea del pluviómetro. Dado que esta combinación de filtro RC y de compuerta con Schmitt Trigger es normalmente implementado para evitar el efecto rebote del interruptor, se considera que el circuito implementado es más que adecuado para evitar disparos en falso causados por ruido ambiental.

Los diodos en el circuito de la figura 3.19 sirven de protección por sobretensiones, en caso de que la tensión de línea supere  $V_{cc}$  más 0.58 [V] (la tensión de encendido del diodo) el diodo  $D_1$  entra en conducción y dirige la corriente a la fuente. Si por el contrario, la tensión de línea baja a -0.58 [V], el diodo  $D_2$  entra en conducción dirigiendo la corriente también a la fuente. Los diodos elegidos MURS115T3G tienen una capacidad de conducción de 1 [A] y se encuentran en un empaque en montaje superficial por lo que fueron elegidos para desempeñar esta función.

La resistencia  $R_1$  es una resistencia de *pull-up* que proporciona un nivel de tensión a la entrada al interruptor. A la salida de la compuerta inversora

(Salida\_Pluv) se obtiene una señal continua de tierra y pulsos en alto en cada cierre del interruptor.

### 3.3. El reloj en tiempo real

Un reloj en tiempo real es un tipo de dispositivo integrado que mantiene un registro de la hora y fecha actual. Los RTR's están presentes en cualquier dispositivo que necesiten mantener la fecha y hora de manera confiable, como lo son PC's, servidores, sistemas embebidos, etc.

Si bien mantener la hora y la fecha puede ser realizado sin la necesidad de un RTR, el uso de uno nos permite:

- Un menor consumo de energía
- Liberar al microcontrolador de la tarea de manter la hora y fecha
- Obtener mayor precisión que con otros métodos

Para realizar su función los RTR requieren de una señal de reloj. Normalmente esta fuente proviene de un cristal oscilador, si bien hay algunos que pueden utilizar la frecuencia de la línea de alimentación. La frecuencia a la que oscila este cristal es de 32.768kHz, la cual es la misma frecuencia que utilizan los relojes de cuarzo, esto es debido a que esta frecuencia es exactamente  $2^{15}$  ciclos por segundo, la cual es una razón conveniente para un contador binario.

El modelo de RTR usado en este *datalogger* fue el circuito integrado DS1337 de la empresa Maxim Dallas. Éste, a demás de poseer las funciones básicas de un RTR, incorpora dos alarmas programables con salidas independientes. Esta función nos permitirá que sea el RTR el encargado de llevar el período de muestreo de la señal de temperatura. Para su funcionamiento el DS1337 sólo requiere de ser alimentado y conectado al *bus* I<sup>2</sup>C. Dado que sus salidas son del tipo colector abierto, se requieren de resistencias de *pull-up* para que éstas operen de manera adecuada.

En la figura 3.20 se muestra un diagrama de la conexión del DS1337. En éste se pueden apreciar tanto las resistencias de *pull-up* en las terminales del *bus* I<sup>2</sup>C (SCL y SDA), como en las terminales de las alarmas (INTB e INTA). Estas resistencias con un valor de 12 [k $\Omega$ ] permiten al tener una alimentación

de 3.3 [V] un flujo de corriente de  $275[\mu\text{A}]$ , lo cual es muy adecuado para ser manejado por los puertos de E/S del microcontrolador y no desperdician tanta energía al llevar la señal de salida a tierra. El DS1337 incorpora terminales para conectar el cristal de 32.768 [kHz].

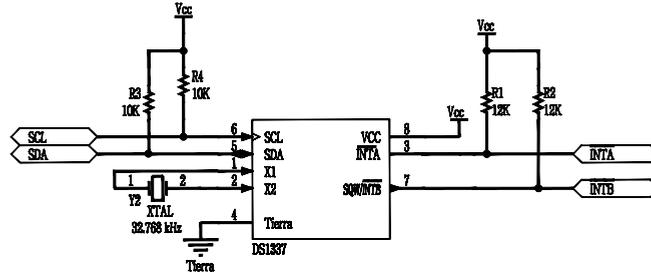


Fig. 3.20. Diagrama de conexión del RTR DS1337.

### 3.4. Las memorias EEPROM

Las memorias EEPROM usadas fueron las 24LC1025 de la compañía Microchip. Estas memorias tienen una capacidad de 1024 [kBits], lo cual representa 128 [kBytes]. La comunicación con ellas se realiza a través del *bus* I<sup>2</sup>C y es posible conectar a éste hasta cuatro circuitos integrados de estas memorias, número que fue colocado en el *datalogger*. Las cuatro memorias usadas nos dan una capacidad de 512 [kBytes], memoria que el *datalogger* puede utilizar para el almacenamiento de los datos adquiridos.

En la figura 3.21 se muestra el diagrama de conexión de las memorias. Para funcionar, éstas sólo requieren ser alimentadas y conectadas al *bus* I<sup>2</sup>C. Los circuitos integrados de memoria poseen dos terminales ( $E_0$  y  $E_1$ ) que deben ser conectadas a tierra o a  $V_{CC}$  para establecer su dirección dentro del *bus* I<sup>2</sup>C. La terminal  $E_2$  no es configurable y debe de ser conectada a  $V_{CC}$  para que la memoria pueda funcionar de manera correcta. Adicionalmente, las memorias cuentan con una terminal para habilitar o deshabilitar su operación de escritura (Modo), esta terminal puede ser controlada por medio del microcontrolador o interruptor mecánico, pero en esta aplicación dicha función no fue utilizada y la terminal fue conectada directamente a tierra para que la escritura de las memorias siempre se encontrara habilitada.

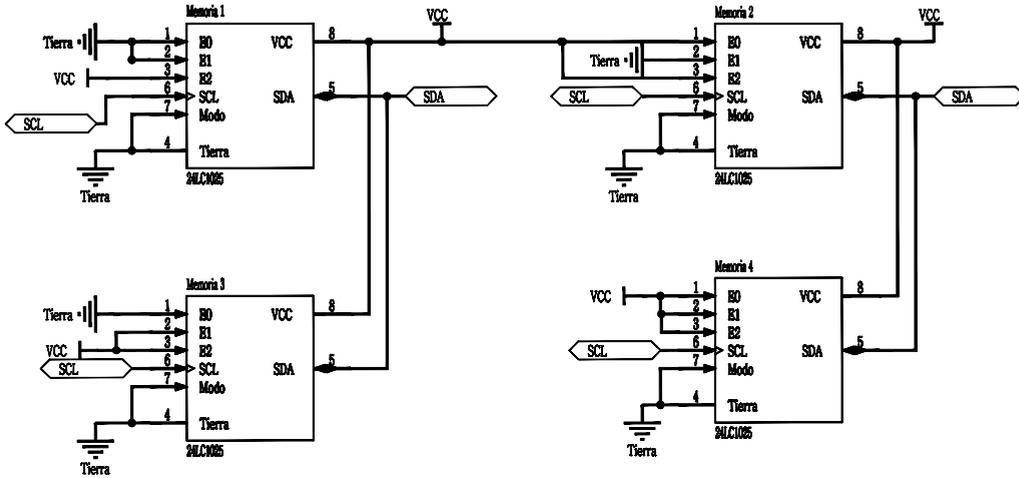


Fig. 3.21. Diagrama de conexión de las memorias EEPROM externas.

### 3.5. El transceptor RS-232

Debido a que las señales provenientes de la USART no cumplen con los requerimientos de tensión y polaridad del estándar RS-232, es necesario emplear un transeptor que haga dicha conversión.

Este transeptor funciona a través de dobladores de tensión con base en capacitores, por lo que para su funcionamiento éstos necesitan estar presentes. En el caso del proyecto en cuestión, se decidió utilizar el transeptor MAX3233 cuyos capacitores de carga se encuentran dentro del empaque, lo cual nos ayudó a ahorrar espacio en el PCB.

Adicionalmente a esto, este dispositivo tiene una terminal para su encendido y apagado, función que fue utilizada con miras en disminuir el consumo de energía del *datalogger*.

En la figura 3.22 se muestra el diagrama de conexión usado. Para funcionar el MAX3233 debe de ser alimentado y sus terminal de entrada deben ser conectadas a las terminales de la USART del microcontrolador (Tx y Rx). A la salida del MAX3233 se obtienen las señales listas para que, por medio de un conector DB9, éstas sean recibidas por la PC.



El transceptor utilizado en este proyecto fue el FT232RL de la compañía FTDI. Para funcionar, este dispositivo sólo requiere de ser conectado a las terminales de la USART del microcontrolador, unos cuantos componentes pasivos para filtrar las señales y de ser conectado al *bus* USB mediante un conector tipo B. No es necesario proporcionarle energía al circuito de manera interna, dado que éste se alimenta de la energía proporcionada por *host* USB. Los capacitores  $C_1$  y  $C_2$  y la ferrita sirven para suprimir el ruido proveniente de la señal de alimentación. El capacitor  $C_3$  desacopla la salida del regular interno de 3.3 [V] que posee el circuito integrado.

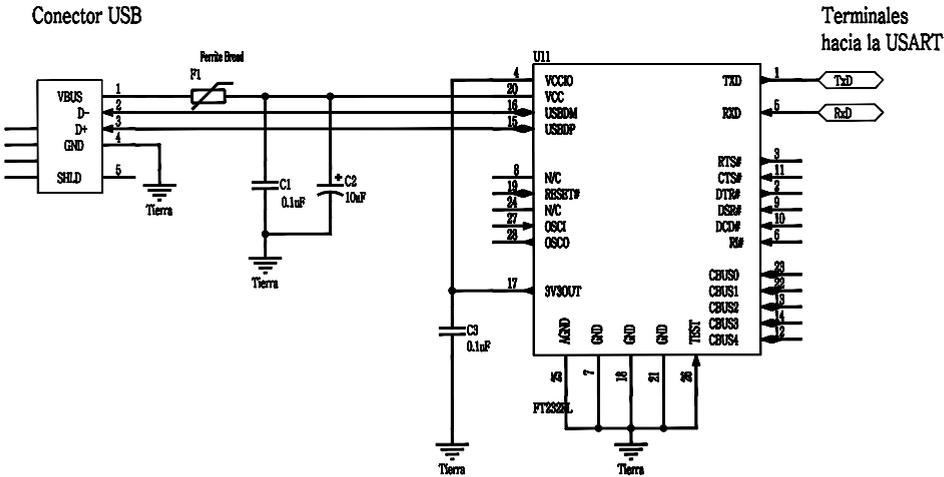


Fig. 3.23. Diagrama del transceptor USB.

### 3.7. La fuente de alimentación

La fuente de alimentación del *datalogger* se realizó con base en el MAX1675, el cual es un convertidor DC/DC de subida. Este circuito puede elevar una tensión de entrada desde 0.7 [V] hasta una tensión de salida ( $V_{sal}$ ) configurable, desde los 2 [V] hasta los 5 [V] con una corriente máxima de 500 [mA].

En nuestro caso, dado que, se desea que el *datalogger* sea alimentado a través de una batería de 1.5 [V], este convertidor es ideal para nuestra aplicación.

En la figura 3.24 se aprecia la conexión del MAX1675 para una salida de 3.3 [V], donde  $V_{CC}$  es la tensión de salida del convertido y  $V_{bat}$  es la tensión proveniente de la batería. Para su funcionamiento, el MAX1675 requiere de muy pocos componentes, siendo el inductor el más importante. Para un funcionamiento eficiente, se debe asegurar que el inductor usado posea una baja resistencia en DC y que adicionalmente soporte la corriente de salida que se desea.

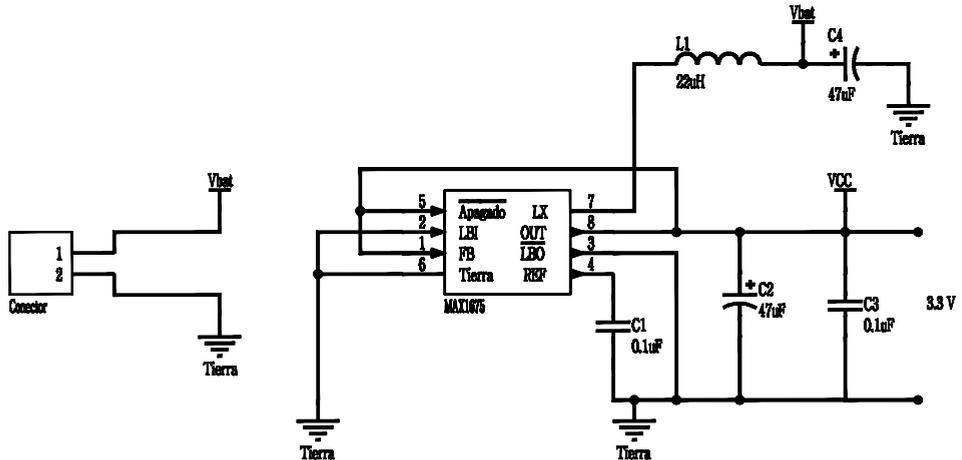


Fig. 3.24. Diagrama de la fuente de alimentación del *datalogger*.

### 3.8. Botones e indicadores

Con la finalidad de que el *datalogger* pueda tener una interacción con el usuario en sitio, se implementaron controles a través de botones de presión y de posición. De igual forma, para indicar acciones realizadas por el usuario, en el *datalogger* se colocó un LED que, haciendo usos de secuencias de parpadeo, puede indicarle rápidamente al usuario la operación que fue realizada por el *datalogger*, por ejemplo, el inicio o paro de la adquisición.

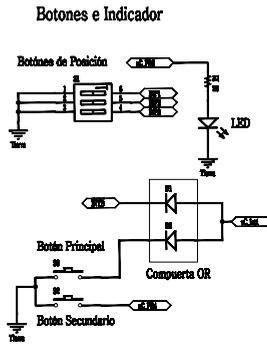
El diagrama de conexión de los botones y el LED se puede observar en la figura 3.25. Tanto los botones como el LED están conectados directamente a terminales de E/S del microcontrolador de acuerdo a la tabla 3.1. La única excepción es el botón principal, que se conectó mediante una compuerta OR

a una de las terminales con interrupción externa del microcontrolador. Esta compuerta se implementó de manera discreta con diodos. Por medio de esta compuerta OR, la señal del botón principal comparte la terminal de interrupción con una de las alarmas del reloj en tiempo real (INTB), esta interrupción también fue conectada a la terminal PD5 del microcontrolador con el fin de poder detectar que dispositivo ocasionó la interrupción.

Los botones están conectados a tierra y emplean la resistencia de *pull-up* interna de los puertos para evitar drenar corriente en exceso cuando éstos sean cerrados.

Botón	Terminal del uC
Botón de posición 1	PB0
Botón de posición 2	PB1
Botón de posición 3	PB3
Botón principal	NA
Botón secundario	PB4
LED	PD6

**Tabla 3.1.** Distribución de los botones y el LED en las terminales del uC.



**Fig. 3.25.** Diagrama de conexión de los botones y el LED del *datalogger*.

### 3.9. Integración del hardware

Una vez que todos los componentes del *datalogger* fueron probados en tarjetas prototipo, la integración del hardware se realizó mediante la creación de

circuitos impresos por partes de los componentes usados. Las tarjetas impresas creadas fueron las siguientes:

- Una tarjeta principal que integra el microcontrolador, el RTR y los botones e indicadores.
- Una tarjeta para agrupar las memorias EEPROM.
- Una tarjeta para realizar el acondicionamiento de la sonda de temperatura.
- Una tarjeta para el acondicionamiento del pluviómetro.
- Una tarjeta para la fuente de alimentación.
- Una tarjeta para el transceptor RS-232.
- Una tarjeta para el transceptor USB.

Todas las tarjetas se conectaron a la tarjeta principal del microcontrolador a través de cables planos, para integrar completamente el hardware del proyecto. La tarjeta principal es la que alimenta a los distintos módulos. Debido a que la tarjeta desarrollada encargada del acondicionamiento de la sonda de temperatura tuvo que ser adecuada, éste se muestra implementado en una tarjeta de prototipo. El circuito del *datalogger* completo se muestra en la fotografía de la figura 3.26.

Por último, en estos momentos se encuentra en desarrollo una tarjeta de circuito impreso que integra todos los elementos juntos. El diagrama de dicha tarjeta se muestra en las figuras 3.27 y 3.28. En esta tarjeta, la mayoría de los elementos usados, a excepción de algunos capacitores, son de montaje superficial y estos poseen una mayor integración. Por citar un ejemplo, el modelo de reloj en tiempo real usado en esta tarjeta incorpora el cristal dentro del empaque. Se espera que una vez que esta tarjeta se encuentre desarrollada, ésta se convierta en la tarjeta definitiva del *datalogger*.

Una vez que tenemos identificados todos los componentes de hardware del *datalogger*, en el siguiente capítulo estudiaremos la interacción que éstos llevan a cabo para realizar las funciones del *datalogger*.

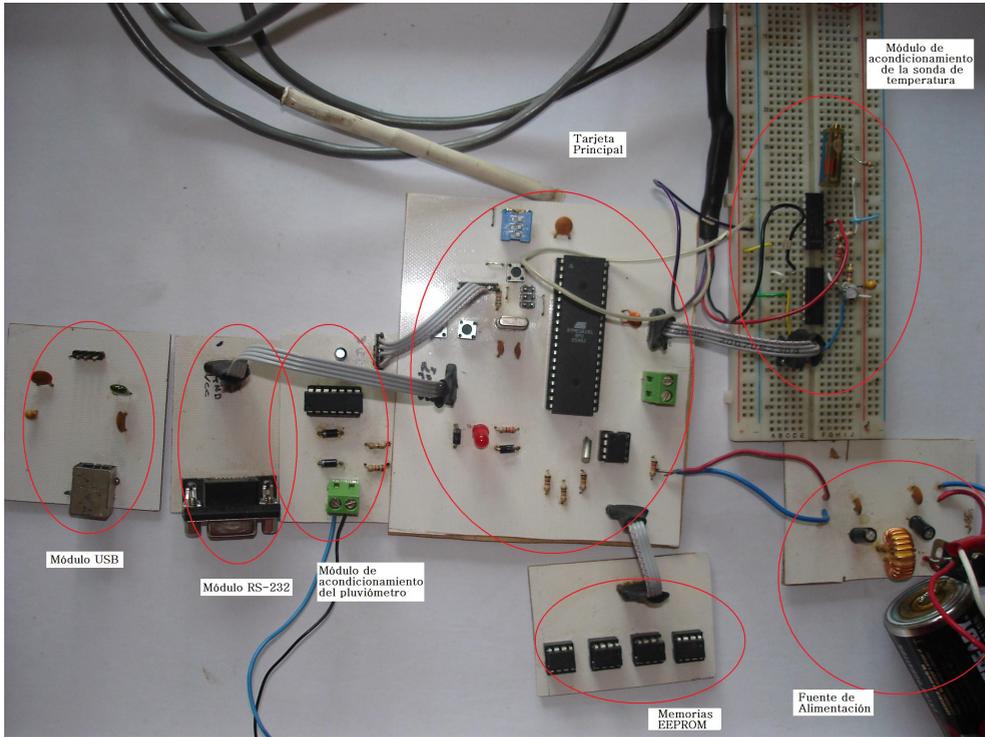


Fig. 3.26. Módulos de los circuitos del *datalogger* interconectados.

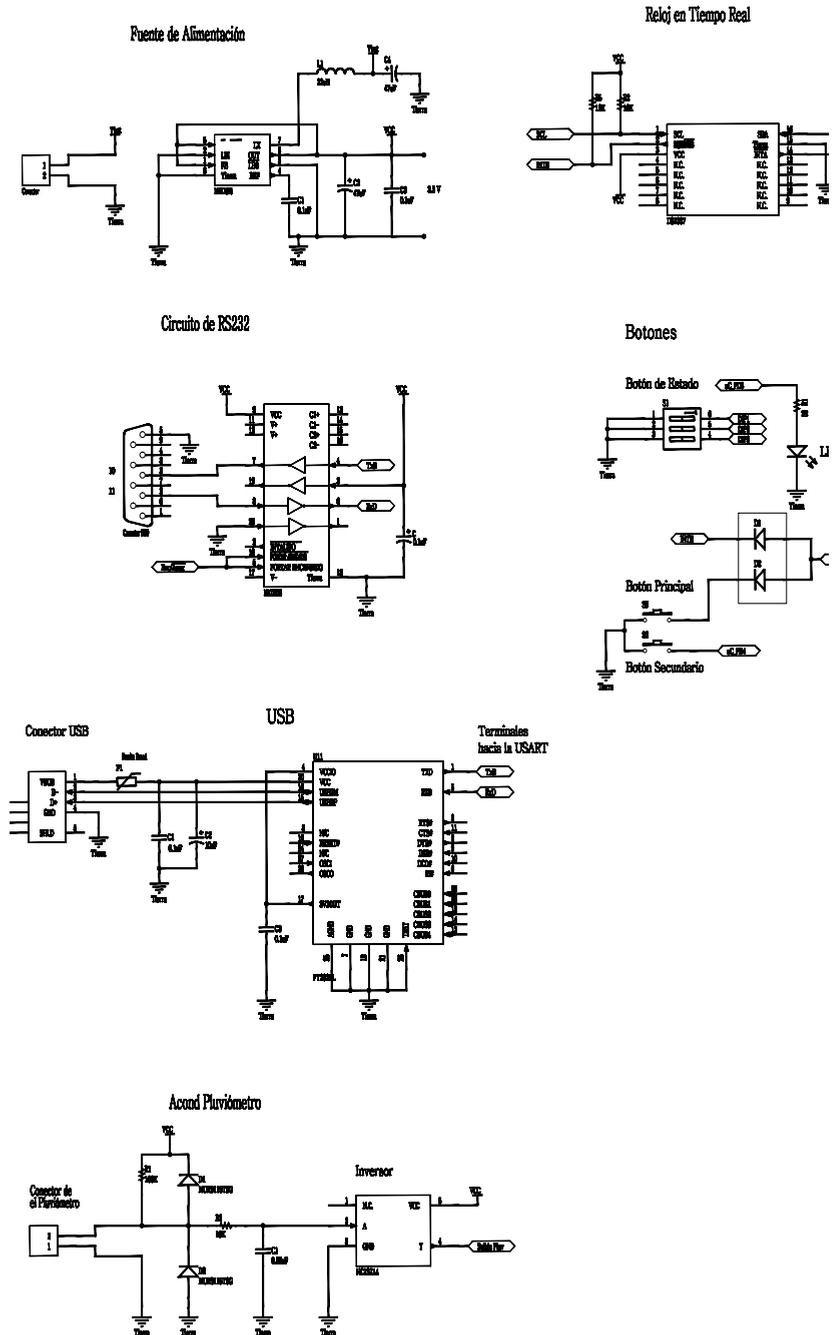


Fig. 3.27. Diagrama del *datalogger* parte 1.

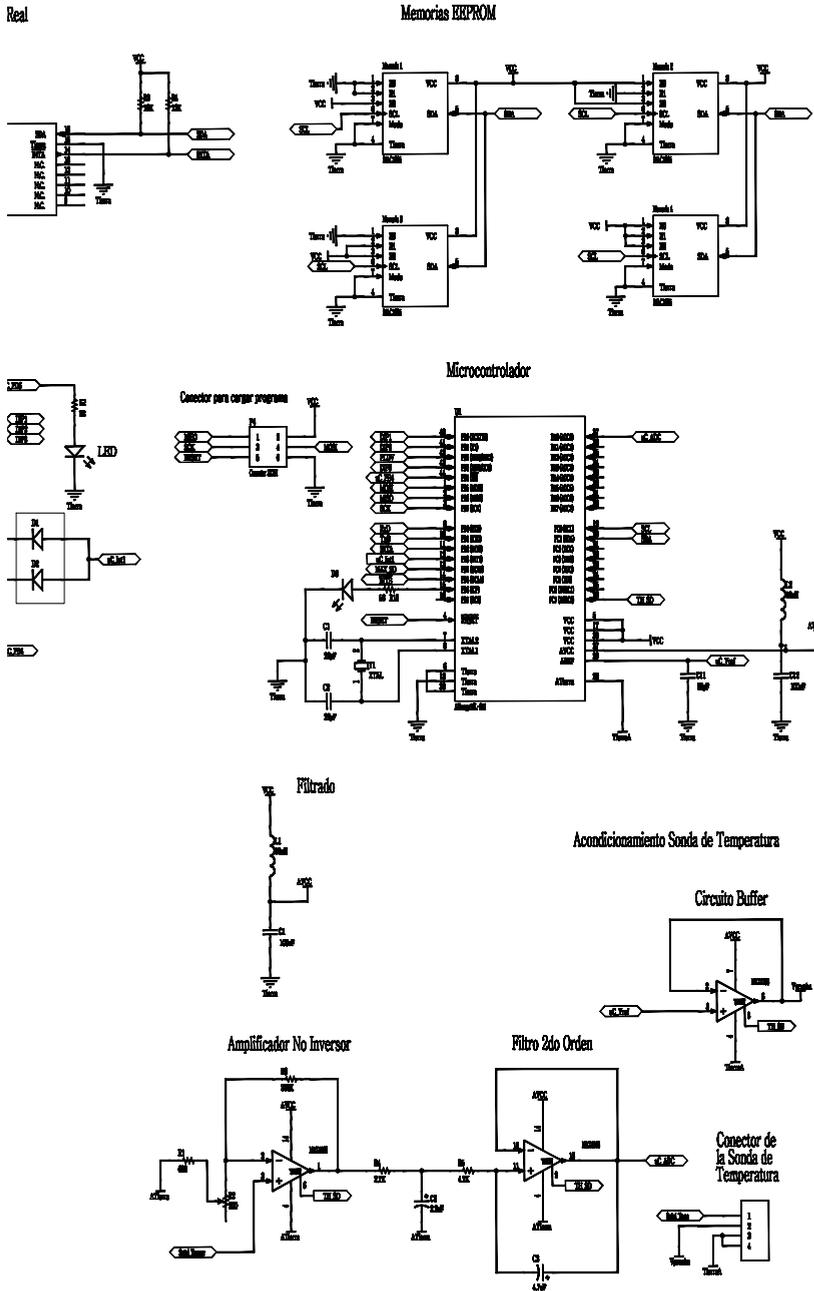


Fig. 3.28. Diagrama del datalogger parte 2.