Capítulo 4

Desarrollo del Programa del Microcontrolador

En este capítulo nos centraremos en la descripción de la interacción que tienen los componentes de hardware del *datalogger*. Toda esta interacción, como ya se mencionó, se realiza a través del programa del microcontrolador que, a partir de este momento, denominaremos *firmware*. El desarrollo del *firmware* del microcontrolador fue realizado a través del compilador AVR GCC, este compilador, de distribución libre, compila código programado en C y lo transforma en un código objeto que puede ser programado en el microcontrolador. La programación que se realizó fue hecha a partir de programar funciones simples, que posteriormente pudieran ser integradas en funciones más complejas hasta componer el programa principal, que realiza las funciones del sistema.

El diseño de esta parte del proyecto se dividió en las tres funciones principales que el *datalogger* debe realizar, y que son: adquisición de datos, guardado de éstos en memoria y la trasmisión de los mismos a una computadora personal (PC).

La adquisición de datos corresponde a la recepción, muestreo y tratamiento de la información proveniente de los sensores previo a ser guardados. La etapa de guardado corresponde al uso, manejo y organización de las memorias para permitir el almacenamiento de la información adquirida. Por último, la etapa de transmisión de datos corresponde a la interacción que se lleva a cabo entre el *datalogger* y la PC, para el envío de los datos guardados en memoria. Adicionalmente a estas etapas básicas, se deben considerar funciones adicionales que lleva a cabo el *datalogger* y que complementan su función principal, como son su calibración, configuración, etc.

En las siguientes secciones se estudiará de manera general la estructura del *firmware*, para posteriormente particularizar en cada una de las funciones que se implementaron en el *datalogger*.

4.1. Estructura del *firmware*

La estructura del *firmware* se desarrolló teniendo como segunda prioridad, sólo después de la realización de las funciones principales del *datalogger* anteriormente mencionadas, el garantizar un bajo consumo de energía del circuito, con el fin de lograr el mayor tiempo posible de autonomía en la adquisición de datos. Para realizar lo anterior es necesario considerar que el *datalogger* hará mediciones de variables cuya velocidad de cambio es muy lenta respecto a su velocidad de procesamiento. Lo anterior significa que el *datalogger*, de su tiempo total de funcionamiento, sólo estará trabajando de manera activa en pequeños intervalos mientras que, la mayor parte del tiempo, se encontrará en espera. La manera en que el *datalogger* permanece en espera es un componente esencial en el manejo eficiente de la cantidad de energía disponible por éste.

Los elementos de los que consta el *datalogger* y su consumo energético alimentados con 3.3 [V] se enlistan en la tabla 4.1.

Componente	**Consumo en operación** [mA]	**Consumo en espera** [μA]
Microcontrolador	5	<1
Circuito acond. sonda temp.	6.16	114
Circuito del pluv.	0.061	NA
Memorias EEPROM	20	20
Reloj en Tiempo Real	5	<1
Transceptor RS-232	5	<1
Transceptor USB	5	<1
LED	5	NA

Tabla 4.1. Consumo energético de los componentes del *datalogger*.

Con la información presentada en la tabla 4.1 nos damos cuenta que un buen diseño del *firmware* intentará que los componentes se encuentren en bajo consumo la mayor parte del tiempo. Tomando en cuenta esto, se buscó que los elementos que componen al hardware del *datalogger* incorporan medios para ser encendidos y apagados por medio del microcontrolador.

El microcontrolador es un elemento que se pone en estado de bajo consumo a través de su *firmware* pero, para sacarlo de este estado, se requiere que un dispositivo externo que así se lo indique. En el capítulo de generalidades vimos

que la manera en que se implementó la salida del microcontrolador de su estado de bajo consumo fue a través de sus tres interrupciones externas.

En el diagrama de flujo mostrado en la figura 4.1 se presenta la estructura general del *firmware*. La ejecución del microcontrolador comienza en el bloque de inicio una vez que el microcontrolador es energizado o cuando el botón de *reset* de éste es presionado. Antes de proseguir con la inicialización del sistema, el programa lee los botones de posición, con los cuales el usuario le indica al sistema que desea reiniciar parámetros o formatear la memoria de datos. Si alguna de estas funciones es seleccionada el *datalogger* las realiza y a continuación se mantiene en un ciclo infinito, con lo cual detiene su ejecución hasta que éste sea reinicializado. Lo anterior asegura que el *datalogger* no sea operado dejando estas opciones activadas.

Si ninguna de estas opciones fueron seleccionadas, el microcontrolador realiza un proceso de configuración del sistema. Esta configuración permite que el *datalogger* inicie sus componentes con valores y estados conocidos y cargue en memoria los parámetros de operación que quedaron guardados durante su última configuración. En caso de que el *datalogger* sea puesto a adquirir sin ser previamente configurado por medio de una computadora, realizará el proceso de adquisición siguiendo estos parámetros.

Una vez configurado el microcontrolador, éste activa sus interrupciones y entra en estado de bajo consumo. En este estado, el microcontrolador queda en espera de interrupciones externas que lo despierten y reanuden su operación.

Como ya se había mencionado, las terminales con interrupción externa del microcontrolador son tres y están divididas en dos con interrupción síncronas y una con interrupción asíncrona.

Para saber a qué señales conectar a estas terminales con interrupción, debemos regresarnos a las funciones que el *datalogger* realizará. En primer lugar, el *datalogger* deberá adquirir dos tipos de variables. Cada variable deberá poseer una señal que indique en qué momento ésta deba ser adquirida. Estas señales pueden ser conectadas a una interrupción externa del microcontrolador para que, una vez dicha señal esté presente, el microcontrolador reanude su operación y adquiera la variable. En segundo lugar, el *datalogger* requiere de una interacción con el usuario, interacción que no se puede dar si el microcontrolador se encuentra en bajo consumo, por lo que, con el fin de que el microcontrolador pueda ser despertado por el usuario, se conectó un botón a la última terminal con interrupción externa disponible.

Una función adicional que se le dio a la última terminal de interrupción ya mencionada, es la de servir como una señal para iniciar la adquisición de las variables meteorológicas de manera programada por tiempo, a través de

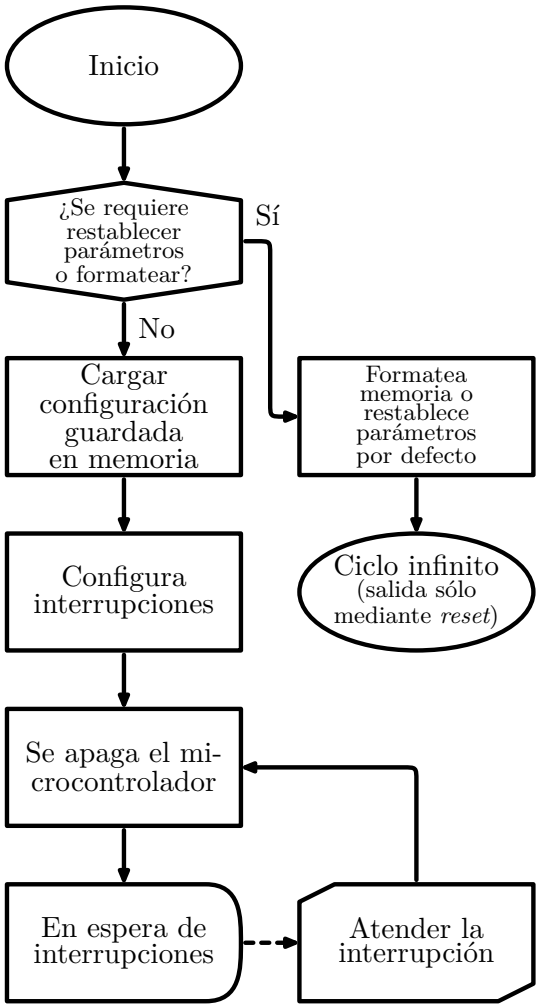

Fig. 4.1. Diagrama de operación del *datalogger*.

una señal de alarma del RTR. De esta manera el usuario puede indicarle al *datalogger* cuando comenzar con la adquisición de las variables. Dado que la señal del botón actívado por el usuario y la salida de la alarma del RTR son dos señales diferentes, se requiere realizar la conexión entre éstas y la terminal de la interrupción del microcontrolador a través de una compuerta OR. Adicionalmente a esto, se debe contar con un mecanismo para identificar la señal que ocasionó la interrupción. Esto se realizó de manera muy sencilla, como se mencionó en el capítulo anterior, al conectar la alarma del RTR a

otra terminal de entrada. De esta manera, en cuanto la interrupción se active, se lee el valor de la otra terminal de entrada y dependiendo de su valor se determina el elemento que ocasionó la interrupción.

Una vez explicado las interrupciones y su origen, en el diagrama de la figura 4.2 podemos apreciar las interrupciones que atenderá el microcontrolador durante su funcionamiento. La interrupción causada por el inicio programado y por el botón principal es básicamente la misma en el circuito pero, para fines del *firmware*, se manejarán como interrupciones distintas.

El inicio programado debe de ser activado mediante la PC y sólo es posible que éste suceda antes de que el *datalogger* comience a adquirir. En cuanto a la interrupción del botón principal, ésta depende del usuario y puede suceder en cualquier momento. Una vez que el usuario presione el botón principal, le indica al microcontrolador que salga de su estado de bajo consumo y espere algún tipo de acción sobre él. Esta acción puede ser: iniciar o parar la adquisición o, iniciar la comunicación con la PC. A través de una conexión con una PC es posible hacer las acciones que se muestran en el diagrama y que son iniciar la adquisición, configurar el *datalogger* y vaciar los datos adquiridos por éste.

Si el sistema se encuentra en un estado de adquisición dos interrupciones más son posibles. Estas son las interrupciones para adquirir las variables medidas por el *datalogger*.

Todas las interrupciones finalizan después de que el microcontrolador ha ejecutado la función programa dentro de la interrupción. En el caso de la interrupción por medio de un botón, ésta posee un tiempo de espera para que el usuario realice una acción, si después de dos minutos no se ha realizado acción alguna o se ha establecido una conexión con la computadora, ésta interrupción también termina. Una vez que las interrupciones han terminado, el microcontrolador vuelve a entrar en estado de bajo consumo esperando a que ocurra otra interrupción que lo haga despertar.

Una vez conocido el proceso general que sigue el *datalogger* en su ejecución, en las siguientes secciones ahondaremos en determinadas funciones del *datalogger*.

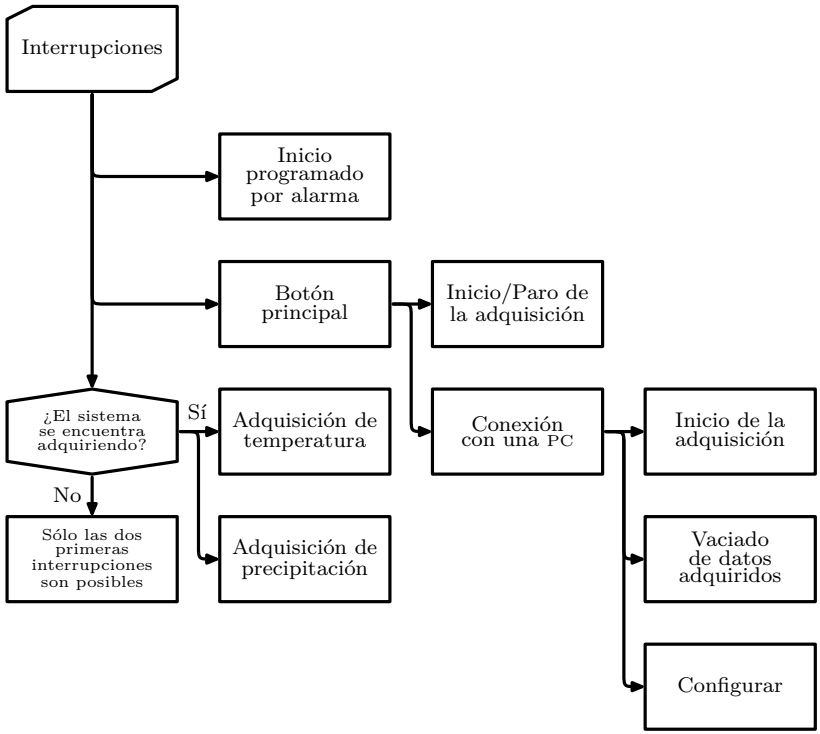

Fig. 4.2. Diagrama de interrupciones del *datalogger*.

4.2. Adquisición de datos

Antes de entrar de lleno en la adquisición de datos debemos conocer el funcionamiento del reloj en tiempo real, ya que éste será el encargado de entregarnos una referencia de tiempo de las señales adquiridas.

4.2.1. Uso del reloj en tiempo real

Como ya se estudió en la capítulo anterior, el RTR DS1337 tiene como función principal llevar el manejo de la hora y fecha del sistema, adicionalmente este dispositivo cuenta con alarmas programables que activan interrupciones en el microcontrolador. Esto convierte al reloj en tiempo real no sólo en una fuente de de información sobre la hora y la fecha sino también en el controlador de tiempos del *datalogger*.

La forma de comunicación entre el RTR y el microcontrolador es a través del *bus* I^2C. La transmisión de información entre ellos se realiza para que el microcontrolador escriba o lea sobre los registros internos del RTR. Estos registros tienen dos propósitos principales: almacenar la fecha y hora, controlar el funcionamiento de las alarmas. La localización de estos registros y la forma de mandar información y recibirla se puede ver de manera detallada en la hojas de especificaciones del DS1337. Para nuestros fines, sólo es necesario saber que la información concerniente a la hora y fecha almacenados por el DS1337 tiene el formato mostrado en la tabla 4.2 y es en éste mismo formato como el microcontrolador recibe la información de la hora y fecha del RTR.

Resistro	**Rango**
Segundos	0-60
Minutos	0-60
Horas	0-12+AM/PM o 0-24
Día de la Semana	1-7
Día del Mes	1-31
Mes	1-12
Año	0-99

Tabla 4.2. Fomato de la fecha y hora del RTR (DS1337).

Todos los registros mostrados tiene un byte de longitud y están en código BCD. Adicionalmente a los registros de fecha y hora existen otros que controlan la configuración y activación de las alarmas. Cada alarma posee un número determinado de registros donde se puede especificar la hora y la fecha en la que se desea que la alarme se active. Las alarmas del RTR se pueden configurar de la siguiente manera:

Como podemos apreciar en la tabla, la única diferencia entre la alarma 1 y la alarma número 2 del RTR es que la primera posee una resolución de segundos. Ambas alarmas pueden programarse para accionarse hasta un mes después de ser establecidas. Una vez que se han programado ya activado sus alarmas, el RTR comparará los registros de tiempo y hora con el de éstas cada segundo, en caso de que exista una coincidencia, el RTR elevará la tensión de la terminal de salida de la alarma correspondiente.

El nivel de salida de esta terminal se mantendrá en alto hasta que la bandera correspondiente a esta alarma sea limpiada por el microcontrolador. Esta es la razón por la cual se conectaron las salidas de las alarmas del RTR a las terminales de interrupciones síncronas del microcontrolador. Debido a que

Alarma 1
Cada segundo
Cada minuto
Cuando los segundos coincidan
Cuando los segundos y minutos coincidan
Cuando los segundos, minutos y la hora coincidan
Cuando los segundos, minutos, hora y día de la semana coincidan
Cuando los segundos, minutos, hora y día del mes coincidan
Alarma 2
Cada minuto, (segundo 0 de cada minuto)
Cuando los minutos coincidan
Cuando los minutos y la hora coincidan
Cuando los minutos, hora y día de la semana coincidan
Cuando los minutos, hora y día del mes coincidan

Tabla 4.3. Opciones de los menús.

la señal de interrupción se debe de mantener hasta que el microcontrolador salga de su estado de bajo consumo y es el microcontrolador el que controla cuando baja esta señal, la señal de alarma del RTR siempre reanudará la operación del microcontrolador.

4.2.2. Recepción de las señales del pluviómetro

Como ya sabemos, el pluviómetro es un instrumento mecánico que trasmite un pulso de señal en el momento en que ha ocurrido el volcado del balancín. Estos pulsos son recibidos a través de la interrupción asíncrona del microcontrolador, el flanco de subida de este pulso activa la interrupción que indica al programa que un evento ha ocurrido. Recibida la señal, el *datalogger* registra inmediatamente este evento en memoria valiéndose de la fecha y hora procedentes del RTR.

Cada evento del pluviómetro, registrado por el *datalogger*, representa cierto volumen de agua precipitado. Para poder convertir un determinado número de eventos en el volumen de precipitación colectada en cierto período de tiempo,

es necesario saber cuál es la cantidad de lluvia con la que el balancín del pluviómetro vuelca. Esta cantidad de lluvia está dada mediante una calibración que el fabricante realiza al equipo antes de venderlo y es un valor constante para cada modelo de pluviómetro. Debido a que este valor no cambia entre mediciones, no es necesario guardarlo junto con los registros, una localidad especial en la memoria EEPROM del microcontrolador está reservada para este fin y su valor se establece durante el proceso de calibración de los sensores.

4.2.3. Recepción de las señales de temperatura

La señal de la sonda de temperatura es recibida a través de una terminal del convertidor analógico digital (A/D), esta señal, dada su naturaleza, puede ser adquirida y registrada en cualquier momento por el microcontrolador. Pero para realizar un buen proceso de adquisición una frecuencia de muestreo y registro deben ser establecidas. Adicionalmente a la frecuencia de muestreo, el microcontrolador sólo obtendrá el valor de su convertidor analógico digital, valor que, para ser transformado en un valor de temperatura ambiente, requiere de un determinado procesamiento.

Recuperación del valor de temperatura

El proceso de conversión del valor del convertidor A/D a un valor de temperatura es el siguiente:

En primer lugar la palabra digital de salida del convertidor A/D se debe adecuar para que represente el valor de tensión de la terminal de entrada del convertidor A/D mediante la ecuación 3.1 que define la salida del convertidor A/D. Una vez obtenido este valor, se debe obtener la tensión a la salida de la sonda de temperatura dividiendo el valor de entrada del convertidor A/D entre la ganancia del amplificador usado en el acondicionamiento. Posteriormente, con este valor de tensión se debe obtener la resistencia del termistor y finalmente mediante la ecuación de Steinhart-Hart obtener la temperatura.

Dado que el proceso descrito es difícilmente implementable en un microcontrolador de 8 bits, por la necesidad de utilizar complejas operaciones matemáticas y datos con parte decimal, se decidió no realizar este procesamiento mediante el microcontrolador del *datalogger* y dejárselo realizar a la computadora, una vez que ésta reciba los datos adquiridos.

Para que la computadora pueda transformar el valor del convertidor A/D en un valor de temperatura, ésta requiere de ciertos valores procedentes de las características tanto de la sonda, como del acondicionamiento que recibió su señal de salida. De estos los que pueden ser considerados como constantes son:

el valor de tensión de referencia del convertidor A/D, dado que, en teoría, este valor no cambia y los parámetros de la ecuación Steinhart-Hart, debido a que el *datalogger* fue desarrollado para operar sólo con la sonda de temperatura 107 y los parámetros de esta ecuación no cambian para diferentes sondas del mismo modelo.

De los parámetros que se consideran variables son: la ganancia dada a la señal de salida de la sonda, debido a que, como ya se mencionó, ésta puede ser ajustada y el *offset* de la señal de la temperatura la cual puede variar debido a factores como el envejecimiento de la sonda. Un proceso para la obtención de dichos parámetros debe de ser considerado dentro de la calibración de los sensores.

Todas las variables de calibración de los sensores deben de estar presentes una vez que éste trasmita la información colectada a la PC. El *datalogger* obtendrá estas variables por parte del usuario antes de adquirir los datos y las almacenará en localidades de su memoria EEPROM hasta el momento en que los datos sean colectados por la PC.

Frecuencia de muestreo

Parte del proceso de adquisición de datos conlleva a considerar la velocidad de adquisición de éstos. Como se vio en la sección de adquisición de señales analógicas, en el capítulo de generalidades, la única restricción para fijar esta velocidad es el criterio de Nyquits.

En el caso del pluviómetro la velocidad de adquisición de los datos está definida por la naturaleza del sensor, sólo obtendremos una medición si el balancín ha volcado. En el caso del termistor la frecuencia de muestreo debe de ser definida por nosotros, para ello es necesario conocer la frecuencia de variación de la temperatura ambiente.

Tomando como ejemplo las estaciones meteorológicas automatizadas (EMAs) del Servicio Meteorológico Nacional (SMN) que toman mediciones de la temperatura ambiente cada minuto y las envían cada diez minutos[1] y de otro tipo de estaciones automatizadas que registran la información adquirida cada hora[2], se decidió que, para el caso del *datalogger*, la tasa de adquisición fuera en múltiplos de un minuto hasta un máximo de treinta y el registro de los datos fuera configurable por el usuario.

[1] http://smn.cna.gob.mx/productos/emas/

[2] **Alcinova S.**, *Automatization of Meteorological Measurements in Republic of Macedonia for the Needs of Environmental Sustainability.*

Para obtener la tasa de adquisición de la señal de temperatura se toma como base la tasa de registro que el usuario quiera y se divide entre treinta, el número máximo de registros que se pueden realizar. Lo anterior nos da un número fraccionario, así que se toma el valor entero más alto y se obtiene la tasa de adquisición de la temperatura. Por ejemplo, si se desea registrar cada 30 minutos, se adquirirá un valor de temperatura cada minuto pero si se desea registrar cada 45 minutos se adquirirá un valor de temperatura cada dos minutos.

La elección de un máximo de treinta muestras se realizó tomando en cuenta que se puede hacer una medición cada minuto en una tasa de registro de 30 minutos, tasa de registro comúnmente usada, y que la suma de 30 registros a 10 bits no sobrepasa el valor máximo almacenable en un entero.

La base de tiempo de un minuto para la medición y registro de la temperatura se obtuvo configurando la alarma dos del RTR para activarse cada minuto. De esta manera el proceso de adquisición y registro de la señal de temperatura se realiza de acuerdo al diagrama de flujo de la figura 4.3

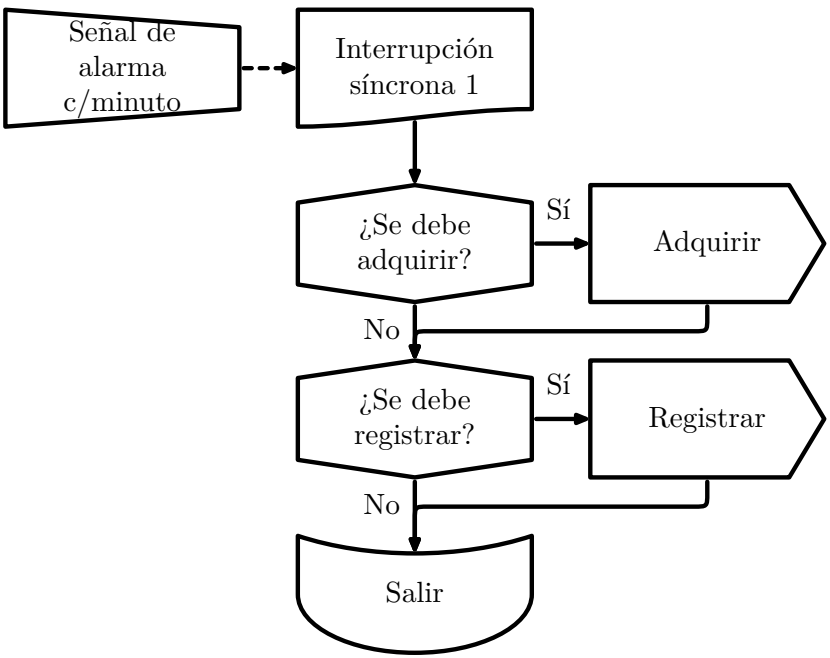

Fig. 4.3. Diagrama del proceso de adquisición y registro de la temperatura ambiente.

Como se muestra en el diagrama, todo el proceso comienza al recibir la señal de alarma del RTR. Una vez que esto sucede, el microcontrolador determina si es momento de adquirir o de registrar la temperatura, si es así, la acción se realiza y se sale de la interrupción. La adquisición se refiere al proceso de leer un valor de temperatura mientras que el registro es guardar el valor promedio de mediciones hechas durante el periodo de registro en las memorias EEPROM.

El proceso de registro de la temperatura ambiente se realiza de acuerdo a la secuencia que se indica en el diagrama de la figura 4.4. Este proceso obtiene una muestra de temperatura ambiente realizando cinco mediciones consecutivas del convertidor A/D y obteniendo el promedio de las mismas, previo a descartar el valor más alto y más bajo de éstas. El promedio obtenido es guardado para ser considerado al momento de registrar esta variable en las memorias, una vez que el período de registro establecido por el usuario haya sido alcanzado.

4.3. Guardado de los datos

4.3.1. Manejo de las memorias de registro de datos

Las memorias que se utilizaron para el registro de datos fueron las memorias EEPROM 25LC1024. Estas memorias se comunican con el microcontrolador a una velocidad de 100 [Hz] a través del *bus* I^2C. A este *bus* se encuentran conectadas cuatro memorias con una capacidad de almacenamiento total de 4.096 [MBits] o 512 [kBytes]. Cada memoria tiene una capacidad de 1024 [kBits] divididos en dos bloques de 512 [kBits]. Cada bloque a su vez se encuentra dividido en 500 páginas de 128 [Bytes] cada una.

Dentro del *bus* cada memoria es identificada a través de una dirección, que se forma basándose en la estructura mostrada en la figura 4.5

El bit R/W selecciona el tipo de operación que se desea realizar con las memorias, lectura o escritura. Los parámetros A son establecidos por hardware, a través de las conexiones que se realizaron de las termínales E_0 y E_1 del circuito integrado de memoria. El bit de bloque B_0 puede ser un uno o cero binario, de acuerdo al bloque dentro del circuito integrado que se quiera seleccionar. Finalmente, el código de control es un valor fijo que comparten todos los circuitos integrados de memoria de esta familia.

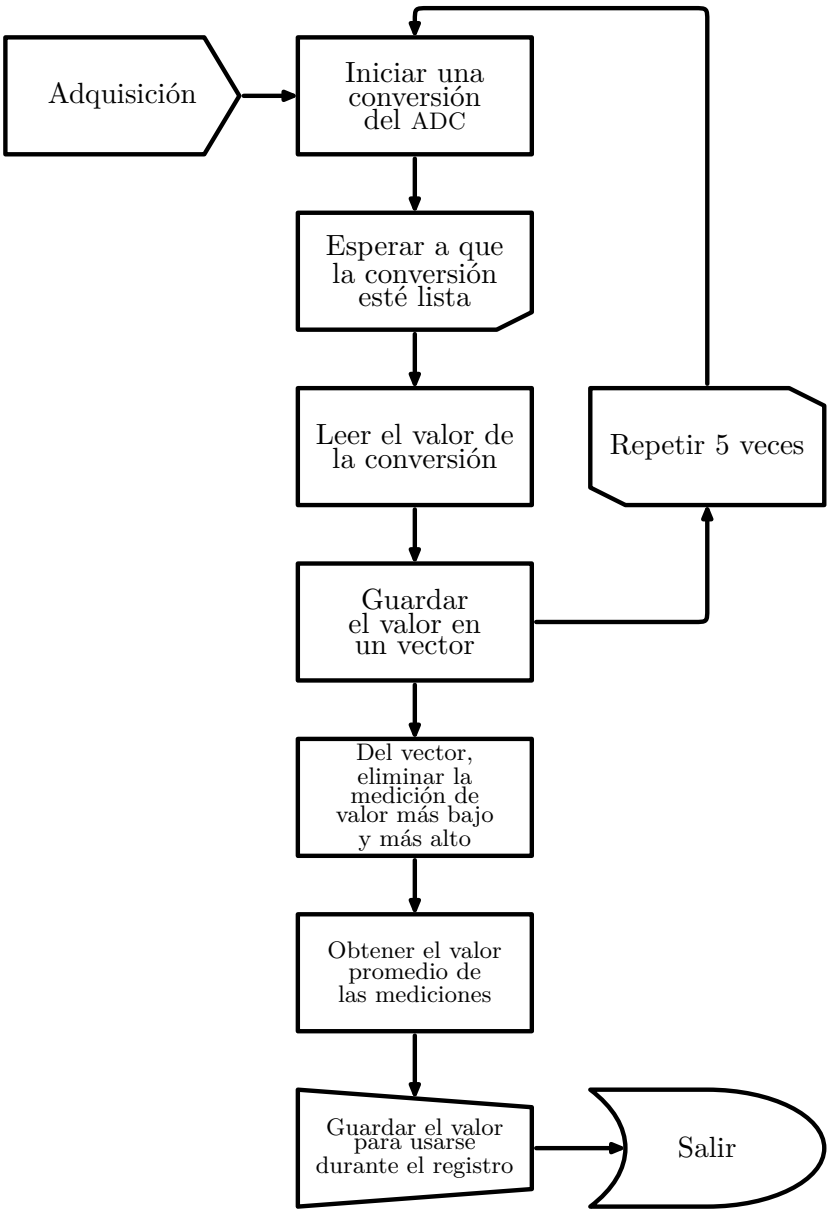

Fig. 4.4. Proceso de la adquisición de la temperatura ambiente.

Solamente dos funciones son posibles de realizar con las memorias, guardar datos en ellas y leer los datos almacenados en ellas. La escritura o lectura se realiza en dos pasos; en el primer paso se especifica la dirección de memoria

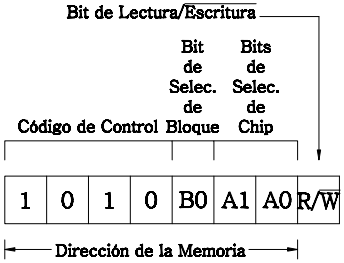

Fig. 4.5. Estructura del byte de dirección de las memorias 24LC1025.

que se quiere leer o escribir, y en el segundo paso se escribe o se lee esa localidad de memoria apuntada.

La dirección de memoria se especifica a través de dos bytes, según la estructura mostrada en la figura 4.6. Esta dirección es la de un byte en memoria y puede varias desde el 0000_{H} hasta el $\mathrm{F9FF}_{\mathrm{H}}$. Esta dirección sólo especifica una dirección dentro del bloque seleccionado en el byte de dirección.

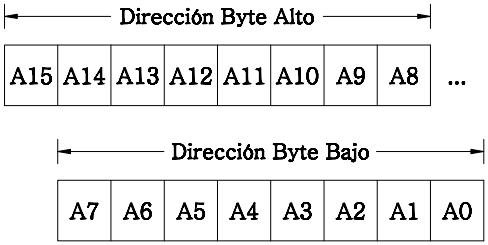

Fig. 4.6. Estructura de los bytes de dirección.

El proceso de escritura se realiza a través de la secuencia en tiempo mostrada en la figura 4.7. Primeramente se inicializa la comunicación $\mathrm{I}^2\mathrm{C}$, mediante una señal de inicio. Una vez iniciada la comunicación, a través del byte de control, se selecciona la memoria y el bloque que se desea escribir. Después se envían los dos bytes de la localidad de memoria donde la operación de escritura se desea realizar, y finalmente se envían los bytes a escribir en dicha localidad. Se pueden continuar enviando más bytes y estos serán guardados en las localidades continuas hasta alcanzar el final de una página, momento en el cual el apuntador interno de la memoria se reiniciará a la dirección de inicio de la página, con lo cual se podrían sobreescribir valores previamente enviados.

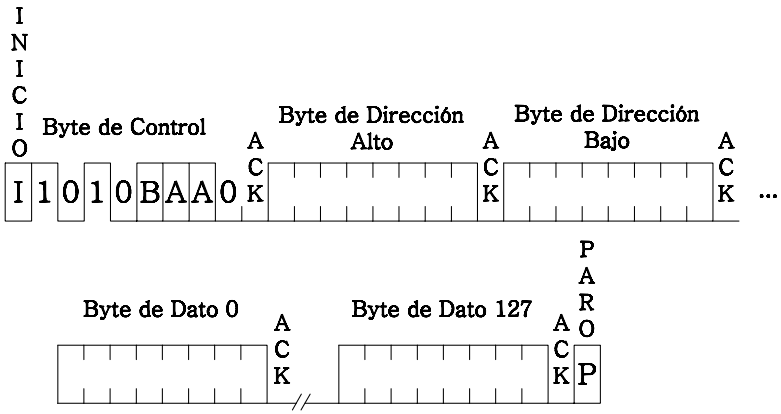

Fig. 4.7. Proceso de escritura de múltiples bytes en memoria.

El proceso de lectura es similar al de escritura pero éste se realiza en dos pasos: primeramente se establece el apuntador de memoria con una operación de escritura y después se reinicializa la comunicación y se inicia una operación de lectura. Mediante este procedimiento la memoria envía el byte de información cuya dirección fue especificada en la operación de escritura previa. Si el microcontrolador sólo desea ese byte, éste puede enviar una señal de no enterado (NACK) y parar la comunicación. Si por el contrario, el microcontrolador desea leer los bytes siguientes a dicha dirección, éste puede responder al byte recibido con una señal de enterado (ACK), entonces la memoria enviará el byte siguiente a la dirección especificada. Si el microcontrolador continúa respondiendo con señales ACK a cada byte recibido, la memoria continuará enviando el contenido de los bytes en las direcciones continuas hasta que una señal NACK es enviada. Se puede hacer lectura de bytes continuos que se encuentren entre el inicio y la mitad del bloque de memoria y entre la mitad del bloque de memoria y el final. En el caso en el que se sobrepasen estas fronteras el apuntador de lectura se regresará al principio del subbloque de memoria. Este proceso de lectura de múltiples bytes se puede ver en la figura 4.8.

4.3.2. Manejo del apuntador de memoria

El manejo de la memoria de datos se realizó mediante la implementación de dos apuntadores de dirección, uno que apunta a la localidad donde los datos empezaron a guardarse y otra que se va incrementando conforme los datos se guardan, esta idea es mostrada en la figura 4.9.

El uso de estos apuntadores permite mantener los datos útiles entre los límites de estas direcciones. Existen dos ventajas en realizar esto:

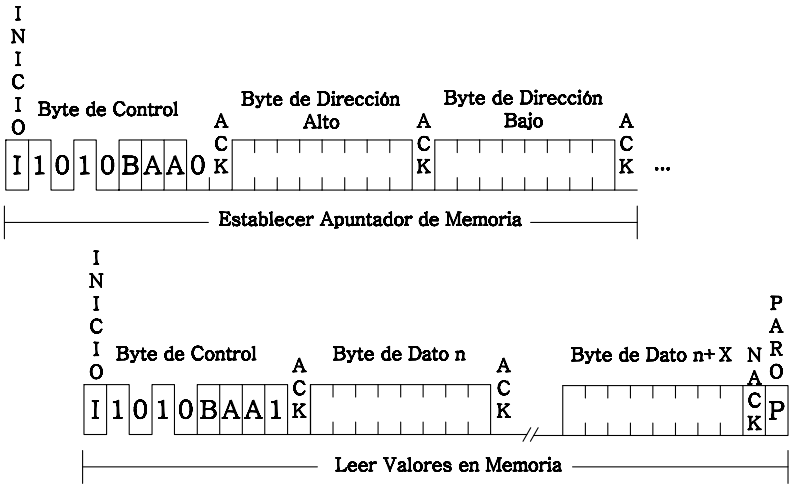

Fig. 4.8. Proceso de lectura de múltiples bytes en memoria.

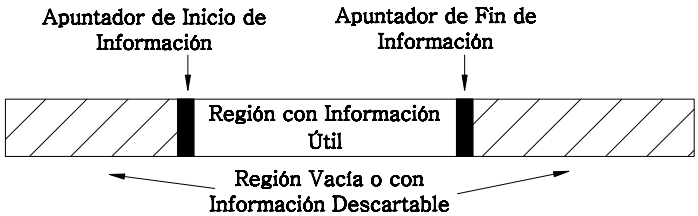

Fig. 4.9. Estructura de la memoria de datos.

- Se evitan borrar físicamente los datos cuando ya no sean útiles y
- Se permite un gasto uniforme de las memorias disponibles

La primera ventaja radica en que si se determina que ciertas localidades ya no tienen información útil, simplemente se dejan fuera de la región entre los apuntadores de inicio y fin del datos. Lo anterior logra que el sistema, en vez de borrar la información, la coloque fuera de su región de datos útiles, acción que tiene el mismo efecto que borrarlos pero que evita gastar de más las celdas de memoria.

La segunda ventaja se basa en que el apuntador de inicio puede ser movido y no siempre debe de permanecer en la primera localidad de la primera memoria. Esto hace posible que los datos puedan ser escritos de manera uniforme sobre todas las memorias, después de múltiples secciones de medición y no sólo en las primeras localidades. Sin implementar esta función, a la larga se

ocasionaría un fallo por ciclos de escritura en las primeras memorias, mientras que la última podría nunca haber sido utilizada.

Para realizar la implementación anterior se deben considerar dos apuntadores: uno para la localidad de inicio de los datos y otra para el final de los mismos. El apuntador más difícil de implementar es el de fin de datos, ya que éste tiene que ser incrementado de manera tentativa cada que se grabe una nueva medición en las memorias. El apuntador de inicio de datos no presenta tal dificultad, dado que sólo se escribe al inicio de una sesión de adquisición. Se debe de tener en cuenta que esta implementación requiere que el microcontrolador maneje la memoria como un anillo, para que, una vez que ha escrito en la dirección final de la última memoria, éste pase de manera automática a la primera.

Dado que el diseño del apuntador de fin de datos determina la manera de implementar ambos apuntadores de dirección, se comenzará con la descripción de éste.

Apuntador de fin de datos

El apuntador de fin de datos es una variable, dentro del programa del microcontrolador, que apunta a la siguiente localidad disponible para el guardado de la información adquirida. Esta variable, junto con la dirección donde se inició el registro de las mediciones, nos ayuda a delimitar la información que se ha sido guardado en la presente sesión de medición. El microcontrolador mantendrá esta variable en memoria volátil SRAM para utilizarla al momento de registrar las mediciones. El problema de esta variable se presenta cuando la alimentación del microcontrolador falla o éste es reinicializado. Cuando esto sucede el valor de esta variable se pierde y por ende perdemos el apuntador de la siguiente dirección de memoria disponible para el registro de las variables.

El problema anterior se puede solucionar guardando una copia de este apuntador en alguna localidad en memoria no volátil, en nuestro caso memoria EEPROM, y que ésta sea actualizado con cada medición. Existe un problema a la solución anterior, las memorias EEPROM sólo pueden ser escritas un determinado número de veces (aproximadamente 1 millón de veces) antes de que estas dejen de retener su valor guardado. Si utilizamos un solo registro y éste se actualiza en cada medición, la vida útil del *datalogger* estaría limitada a 1 millón de mediciones antes de que éste fallara al no poder guardar este apuntador. La solución a este problema se logra mediante la distribución de este apuntador en varias localidades de memoria, de esta manera por cada

localidad extra para guardar este apuntador se obtienen un millón más de mediciones posibles. Si bien esta solución es bastante cercana a lo que queremos, es posible mejorarla, según se describe a continuación.

Como ya se mencionó, las memorias EEPROM utilizadas están organizadas internamente a través de páginas de memoria. Estas páginas de memoria tienen una longitud de 128 bytes, comenzando en la dirección cero y están espaciadas cada 128 localidades. El manejo de la memoria a través de páginas nos da la oportunidad de tener una mejor organización y nos permiten realizar una mejor implementación del apuntador de fin de datos. Esta mejoría se logra si en vez de incrementar el apuntador de memoria en cada registro, se incrementa cada que una nueva página es utilizada. Con el método anterior se logra que el número de veces que se debe de actualizar el apuntador en memoria no volátil sea considerablemente disminuido.

Adicionalmente a tener un apuntador de página, una mejor solución conlleva que su registro no sólo se realice en una localidad de memoria sino en múltiples localidades con el fin de repartir el desgaste de las celdas.

Para realizar lo anterior se utilizó la primera página de la primera memoria para crear una tabla de apuntadores de página. Esta tabla permite organizar las distintas localidades donde el apuntador de fin de datos será almacenado. La estructura de ésta se puede ver en la figura 4.10.

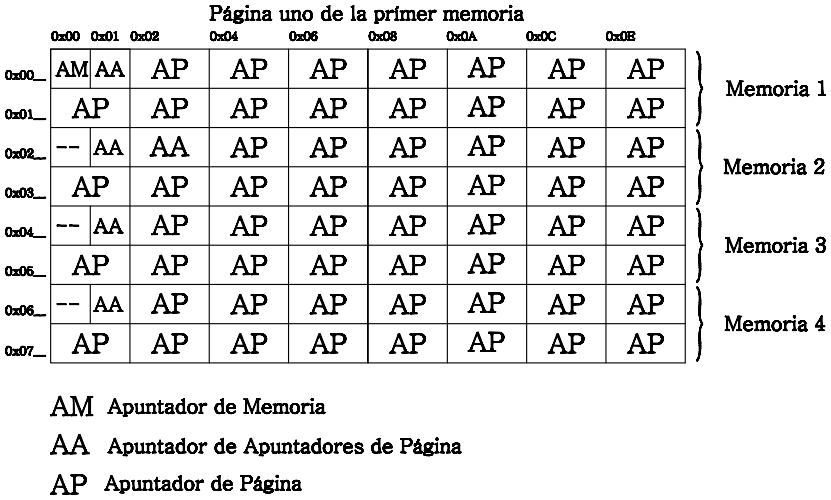

Fig. 4.10. Estructura de la tabla del apuntador de fin de datos.

La tabla está integrada para que el microcontrolador, una vez que la lea, pueda armar una dirección de una localidad en cualquiera de las cuatro memorias

disponibles. Para realizar esto, la tabla mantiene un registro tanto del valor de la memoria usada como de la página dentro de esta memoria. Cada memoria posee un conjunto de quince celdas para almacenar los apuntadores de sus páginas, pero sólo una celda está activa a la vez. El contenido de los apuntadores AM y AA son direcciones que, al ser leídas e interpretadas, originan direcciones sobre otras celdas de la tabla hasta originar la dirección del byte de inicio de la siguiente página disponible para el guardado de datos.

El apuntador de memoria (AM) indica cuál es la memoria que se está utilizando para guardar datos y por lo tanto que apuntador se encuentra en uso. El apuntador de apuntador de página (AA) especifica, de la lista de los quince apuntadores, que esa memoria dispone cuál es el que se encuentra en uso, y finalmente, el apuntador de página (AP), especifica dentro de esa memoria que página se está utilizando. Cada apuntador tiene una longitud de dos bytes, dado que el número de páginas por memoria requiere de dos bytes para ser almacenada: el apuntador de memoria y el apuntador de apuntadores de página tienen una longitud de un solo byte.

El proceso para determinar la página de memoria en uso se realiza de la siguiente manera. Se lee el contenido del primer byte de la primera memoria donde se localiza el apuntador de memoria. Con ese valor se localiza el apuntador de apuntador de página de esa memoria, al leer su contenido, se localiza el apuntador de página en uso. Una vez leído el contenido de ese apuntador se puede localizar la página en uso en determinada memoria. Para obtener la dirección del byte correspondiente al inicio de la página, sólo basta multiplicar el valor del apuntador de página por el tamaño de página (128) y se encuentra la dirección de la siguiente localidad disponible.

Una vez que una página se ha terminado de escribir, el programa escribirá en la siguiente página disponible, pero antes deberá de reportar el cambio en la tabla de apuntadores. Para realizar esto, el microcontrolador debe escribir la nueva página que se usará en el apuntador de página. En caso de que éste haya alcanzado el máximo valor del AP, que corresponde al máximo número de páginas por memoria, se deberá de escribir en otra memoria. Para hacer lo anterior, tanto el apuntador de memoria como el apuntador de apuntadores de esa memoria deberán de ser incrementados.

Con el proceso anterior se logra que el apuntador de memoria se reescriba cada que se ha llenado una memoria, el apuntador de apuntadores de página cada que se llena la memoria que representan y los apuntadores de página, aunque se reescriben con cada nueva página usada, sólo se vuelvan a utilizar cada que la memoria que representan se haya llenado quince veces.

De esta manera, cuando el microcontrolador deje de ser alimentado o sea reinicializado y su apuntador en SDRAM se pierde, éste podrá leer la tabla de apuntadores de la memoria EEPROM y a través de ésta fijar su apuntador en SDRAM.

Apuntador de inicio de datos

Adicionalmente al apuntador de fin de datos, para enmarcar la información registrada por el *datalogger* se requiere de un apuntador que guarda la dirección de la primera página donde se comenzó el registro de los datos. Este valor es establecido al inicio de una sesión de adquisición y sólo se sobrescribe si se requiere que cierta cantidad de información ya no se encuentre disponible. Para realizar lo anterior, bastará con escribir, en este apuntador, la dirección de la última página donde se guardaron datos en esa sección, es decir el contenido del apuntador de fin de datos. De esta manera, el sistema considerará que los datos anteriores son localidades disponibles.

Este apuntador, dado que no requiere una escritura de manera recurrente, se fijó en una única posición de memoria no volátil dentro de la memoria EEPROM del microcontrolador.

4.3.3. Escritura de datos en página

El uso de la memoria por páginas nos permite, además de un mejor uso de los apuntadores de dirección, organizar de una manera más ordenada la memoria. Lo anterior se logra al especificar una página por tipo de variable de medición e identificar cada una de ellas a través de una cabecera. Esto nos da una ventaja adicional, nos permite agrupar la información importante para un grupo de mediciones al inicio de la página y no tener que repetir esta información al guardar cada medición. De esta manera, toda página con información, ya sea de precipitación pluvial o de temperatura ambiente, está compuesta de dos partes:

- Una cabecera y
- La información de las mediciones

Esta estructura se puede apreciar en la figura 4.11

La información guardada en la cabecera y la composición de los datos de las mediciones varían de acuerdo a la variable medida.

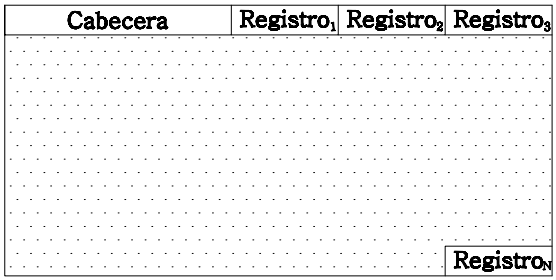

Fig. 4.11. Estructura de una página de datos en memoria EEPROM.

Registro de datos del pluviómetro

La estructura de la cabecera de una página de precipitación pluvial se muestra en la figura 4.12, el identificador que se utiliza es el valor hexadecimal 10_{H}, los dos valores siguientes FF_{H} sólo sirven para mantener la región de datos con las localidades exactas para mantener un número entero de registros.

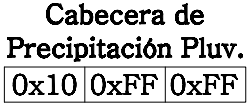

Fig. 4.12. Estructura de la cabecera de precipitación pluvial.

La estructura de los datos de precipitación pluvial se muestran en la figura 4.13, como se puede apreciar, sólo se almacena la fecha y hora en que el evento ocurrió. Dado que el valor de volcado del pluviómetro es guardado durante el momento de la calibración, ningún otro dato se necesita registrar.

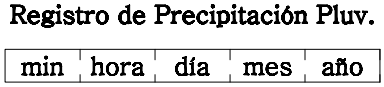

Fig. 4.13. Estructura de un registro de precipitación pluvial.

Registro de datos de la temperatura ambiente

La estructura de la cabecera de una página que guarda datos de temperatura ambiente se muestra en la figura 4.14, como se puede ver en la figura, mayor información es guardada en esta cabecera. Esto se debe a que en todo momento sabemos con qué frecuencia se registran los datos de temperatura, por lo tanto, no es necesario guardar la estampa de tiempo de cada medición

hecha. Solamente cuando se registra el primer valor de temperatura en la página, se escribe en su cabecera el valor del periodo de registro y la fecha y hora proveniente del RTR. Las mediciones consecutivas podrán ser estampadas con la información de la cabecera una vez que se reciban los datos por la PC. Una página de temperatura es identificada mediante el valor hexadecimal de 25_{H} en su primer byte.

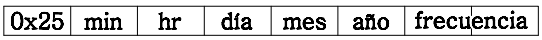

Fig. 4.14. Estructura de la cabecera de temperatura ambiente.

En la figura 4.15 se muestra la estructura de la parte de datos de una página que almacena temperatura ambiente. La información que se almacena para cada medición, es el valor promedio de las mediciones adquiridas mediante el convertidor A/D en cada período de registro. Como ya se mencionó, con este valor y las variables de calibración la PC será capaz de convertir estos valores en mediciones de temperatura al momento de recibirlos.

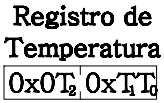

Fig. 4.15. Estructura de un registro de temperatura ambiente.

Al registrar los datos de la manera descrita, se obtiene que por cada página usada se puedan almacenar hasta 60 registros de temperatura ambiente por página y 25 registros de precipitación pluvial. Dado que ninguna variable posee memoria reservada, si el *datalogger* se configura para adquirir tanto precipitación pluvial como temperatura ambiente, ambas variables compiten por espacio en memoria. Si sólo se utilizara el *datalogger* para adquirir precipitación pluvial, la cantidad de registros que podría guardar serían 99,975. Si sólo se guardaran datos de temperatura ambiente se podrían almacenar 239,940 registros. En caso de que cada registro se realizara cada treinta minutos se puede almacenar con el *datalogger* hasta trece años y medio de mediciones. Como se puede apreciar, la mayor limitante en cuanto a la autonomía del *datalogger* es la capacidad de su fuente de energía.

4.3.4. Modos de manejo de la memoria

Los modos de guardado de datos son dos y están pensados para el caso en que las memoria agoten su capacidad durante una sesión de adquisición.

Con la primera opción, el *datalogger* para la adquisición de las variables en el momento en que la memoria se llene. El otro modo permite que la adquisición continúe, pero los nuevos registros serán guardados en el lugar de los registros más antiguos. Estas opciones son configuradas sólo mediante una conexión con la PC.

4.4. Envío de datos

Para realizar la comunicación entre el microcontrolador y la PC se hizo uso de los transceptores USB y RS-232, estos reciben señales de la USART del microcontrolador y los acondicionan para poder ser recibidos por la computadora mediante los puertos correspondientes a cada estándar.

La comunicación con la USART se realiza a una velocidad de 19.2 [kbps]. Esta velocidad tienen un error en una comunicación asíncrona de 0.2%, error que está por abajo del máximo aceptable para este tipo de velocidad y estructura del marco de datos, el cual es del 2%. Con esta velocidad se espera poder trasmitir toda la información almacenada en las memorias en aproximadamente cinco minutos considerando que 19.2 [kbps] representa una velocidad de 1920 bytes por segundo y que la capacidad máxima de las memorias es de 512 mil bytes.

El envió de datos se basa en implementar los protocolos de comunicación y enviar las páginas de información una por una cuando el programa de la computadora las pida. Previo a la petición de envió de datos, la PC solicita las direcciones de inicio y fin de los datos colectados y a partir de estas direcciones calcula cuántas páginas deben ser solicitadas al *datalogger* y la cantidad de memoria libre y en uso.

La información que es trasmitida es aquella que se encuentra entre el apuntador de inicio de datos y el de fin de datos, excepto si se seleccionó el modo de grabación de reescritura de datos y se están guardando los registros sobre registros antiguos. Cuando esto sucede el microcontrolador determina que el apuntador de inicio de información ya no es válido, por lo que envía toda la información guardada en memoria, comenzando con la página siguiente del apuntador de fin de memoria hasta la dirección de la página apuntada por éste.

Una vez que el usuario, mediante el botón principal, ha despertado al microcontrolador, el microcontrolador espera dos minutos para recibir un comando desde la PC por medio de la USART, si un comando no es recibido dentro de ese período de tiempo el microcontrolador entre en estado de bajo consumo. Si el microcontrolador recibió un comando válido, este permanecerá encendido hasta que un comando de término de la comunicación haya sido recibido. Comandos adicionales están disponibles para realizar la configuración del *datalogger*, calibrar sus sensores y habilitar el modo de prueba de éste.

4.5. Funciones adicionales

A parte de las funciones ya mencionadas en los apartados anteriores, existen otras que hacen que el *datalogger* pueda interactuar con el usuario para que éste lo pueda poner en funcionamiento, configurarlo, etc. Estas funciones están divididas en dos: las que se pueden realizar en el circuito y aquellas que requieren que el sistema esté conectado a la computadora.

Primeramente se mencionarán aquellas realizables con el circuito y posteriormente se indicarán las que requieren una conexión con la PC. Las que se pueden realizar con el sistema son las siguientes:

- Inicio y paro de la adquisición del *datalogger*
- Formateo de la memoria de registros
- Establecer parámetros de operación por defecto
- Encender o apagar la opción de indicación de cada adquisición

Cada una de estas funciones requiere de controles para poder ser indicadas, ya sean botones de acción momentánea o de posición. Estos controles se muestran en la figura 4.16.

4.5.1. Inicio y paro de la operación del *datalogger*

Como su nombre lo dice, esta función inicia o detiene la adquisición de datos del *datalogger*. Estas acciones se realizan mediante los dos botones de acción momentánea. El botón principal sirve para solicitar la opción y el secundario para ejecutarla. En el mecanismo de inicio y paro de la adquisición

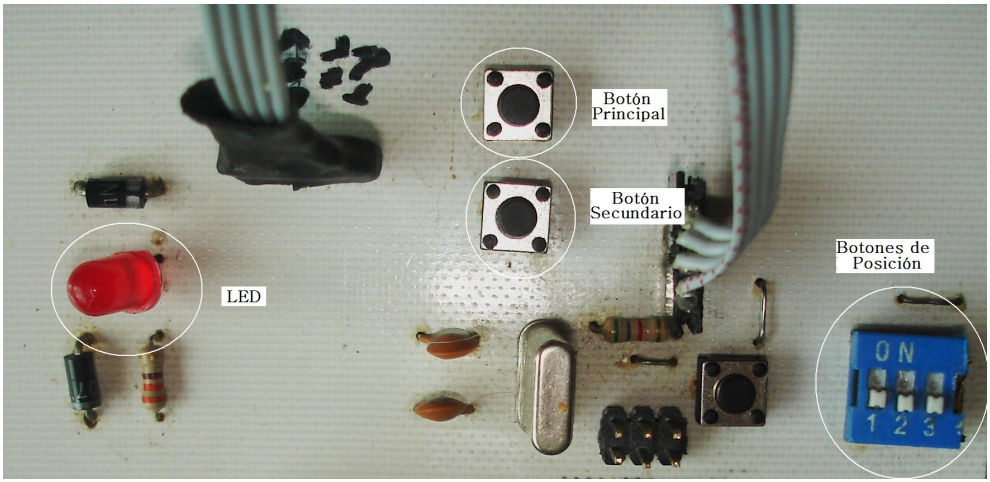

Fig. 4.16. Controles presentes en el *datalogger*.

de datos es el mismo y sólo realiza una acción o la otra, dependiendo del estado de operación en que se encuentre el sistema; si el *datalogger* se encuentra adquiriendo la operación detiene la adquisición y, si ésta se encuentra detenida, la inicializa. El sistema responde a través del parpadeo del LED con una diferente secuencia de acuerdo a la función que realizó.

El inicio del sistema se realiza con los últimos parámetros configurados y en caso de que haya sido restablecida su configuración con los parámetros por defecto.

A parte del inicio por botones se puede iniciar la adquisición del *datalogger* programado por fecha y hora. Esto requiere de una conexión con la PC, pero en caso de que se quiera cancelar este modo de inicio, se puede realizar esto al presionar el botón principal y mantener presionado el botón secundario por cinco segundos.

4.5.2. Formateo de la memoria de registros

Mediante el interruptor de posición número uno, se puede borrar todos los registros guardados en la memoria. Esta operación sólo puede ser realizada si el *datalogger* es puesto en funcionamiento con este interruptor puesto en su estado lógico "alto", el *datalogger* indica la ejecución de esta función manteniendo el LED encendido y ninguna operación más puede realizarse. Para volver a la operación normal tiene que ser apagado y reencendido el *datalogger*.

La función de formateo reescribe la tabla de apuntadores de memoria, que se creo para manejar el apuntador de fin de datos, lo cual hace que este

apuntador de fin de datos se establezca en la segunda página de la primera memoria. Adicionalmente, el apuntador de inicio de datos es igualmente establecido en este valor. Estas dos acciones ocasionan que el microcontrolador determine que ningún dato ha sido guardado en memoria. El proceso de formateado realmente nunca borra las memorias, simplemente le indican al *datalogger* que las localidades se encuentran disponibles para ser escritas.

4.5.3. Establecer los parámetros de operación por defecto

El establecimiento de los parámetros de operación por defecto se realiza colocando el interruptor de posición número dos en su estado lógico "alto" antes de que el *datalogger* entre en funcionamiento. Mediante esta opción se establecen los valores de configuración por defecto que se enlistan más adelante.

4.5.4. Encender o apagar la opción de indicación de cada registro

La opción de indicación del registro de datos le permite al usuario decidir si desea ver un parpadeo del LED cada que un valor es registrado o no. Esta opción puede ser encendida o apagada durante la adquisición de datos mediante el interruptor de posición número tres y permite indicarle al usuario que se están realizando el registro de datos. Por fines de eficiencia en el uso de energía es preferible dejarla apagada.

Otras funciones del *datalogger* están disponibles pero sólo pueden ser accedidas a través de una conexión con la computadora, éstas son:

- Calibración de los sensores
- Modo de prueba de los sensores
- Configuración del *datalogger*

4.5.5. Calibración de los sensores

La calibración de los sensores es necesaria para obtener mediciones precisas de estos por parte del *datalogger*. En el caso del pluviómetro la calibración consiste en saber cuál es el valor con el que éste vuelca. Este valor varía de acuerdo al modelo del pluviómetro y es proporcionado por el fabricante.

En el caso del termistor, la parte de acondicionamiento es tan específica para este tipo de sonda de temperatura que sólo se puede asegurar el uso del

termistor para dicho modelo de sonda. De acuerdo al fabricante, la calibración que es necesario hacer a la sonda es sólo del desplazamiento del valor medido al real (*offset*). Aún así, en la etapa de acondicionamiento, el valor de amplificación que sufre la señal proveniente de la sonda de temperatura puede variar. Por lo tanto, para realizar la calibración de la sonda de temperatura, se requerirán de dos valores: el *offset* y la ganancia del amplificador.

Una vez que la información sobre la calibración es recibida por el *datalogger*, éste almacena dicha información en memoria hasta que, previo al envío de los datos almacenados, le sea requerida por la PC. Con este fin, el *datalogger* tiene localidades reservadas dentro de la memoria EEPROM del microcontrolador para el almacenamiento de estas variables. Las variables de calibración son guardadas en un tipo de datos flotante por el *datalogger* pero nunca son realmente utilizadas por éste. La posición en memoria que estas variables tienen se puede ver en la tabla 4.4.

Parámetro	**Tipo de Dato**	**Dirección**
Dirección de inicio de datos	Entero	0x00
Número de memoria de inicio de datos	Chart	0x02
Tipo de adquisición	Chart	0x04
Tipo de guardado	Chart	0x05
Bandera de fin de memoria alcanzado	Chart	0x06
Período de registro temperatura	Chart	0x10
Identificador del *datalogger*	Arreglo chart	0x16
Capacidad del pluviómetro	Float simple	0x25
Ganancia del amplificador	Float simple	0x29
Offset de la sonda de temperatura	Float simple	0x33
Bandera de calibración	Chart	0x38

Tabla 4.4. Estructura de los datos almacenados en la memoria EEPROM del microcontrolador.

4.5.6. Modo de prueba

El modo de prueba le permitirá al usuario verificar que los sensores se encuentren funcionando y que estén obteniendo mediciones válidas antes de

que el *datalogger* sea puesto a adquirir. Lo anterior adicionalmente ayuda al usuario a detectar posibles problemas con los sensores.

Este modo de prueba se realiza, para la sonda de temperatura, leyendo el valor del convertidor A/D cada vez que se recibe un comando de la computadora y enviando el resultado de dicha conversión a ésta. En cuanto al pluviómetro, un comando es enviado a la PC cada que el balancín vuelca indicándole a ésta que un evento de precipitación pluvial ha ocurrido.

4.5.7. Configuración del *datalogger*

A través de la computadora es el único medio por donde el usuario puede configurar los parámetros de operación del sistema. Estos parámetros son los siguientes:

- Establecer la hora del reloj en tiempo real (RTR)
- Tipo de variables a adquirir: sólo temperatura, sólo precipitación pluvial o ambas
- Período de registro de la temperatura
- Modo de operación de la memoria: parar adquisición al llenarse o reescribir registros antiguos
- Modo de inicio: inmediato, al presionar el botón principal o programado por tiempo

Aún así, como ya se mencionó, para el inicio en sitio se establecieron parámetros predefinidos que son:

- Fecha y hora del RTR: 01 de Enero del 2009 a las 0:00:00 hrs
- Tipo de variables a adquirir: ambas
- Período de registro de la temperatura: cada cinco minutos
- Modo de operación de la memoria: parar adquisición cuando ésta se llene
- Modo de inicio: al presionar el botón principal

Todos estos parámetros son cargados una vez que el *datalogger* es energizado por primera vez, por lo que se almacena en la memoria EEPROM del microcontrolador, ver tabla 4.4.

4.6. Integración del *firmware*

En la figura 4.17 se muestra la estructura que se siguió para la programación del *firmware*. Partiendo desde la parte de arriba del diagrama, se desarrollaron funciones para el manejo de los módulos del microcontrolador: el convertidor A/D, la memoria EEPROM interna del microcontrolador y los módulos de comunicación I^2C y USART. A partir de las funciones que hacen el manejo de estos módulos se desarrollaron funciones más complejas, tanto para el manejo de los componentes externos al microcontrolador como para el desarrollo de las funciones de organización de memoria, registro de datos, etc. Una vez que las funciones más generales fueron creadas, se integraron al archivo principal. Es en este archivo en donde se definen los vectores de interrupción, comandos de comunicación y funciones adicionales, éstas últimas dada su sencillez no requirieron desarrollarse a partir de otras.

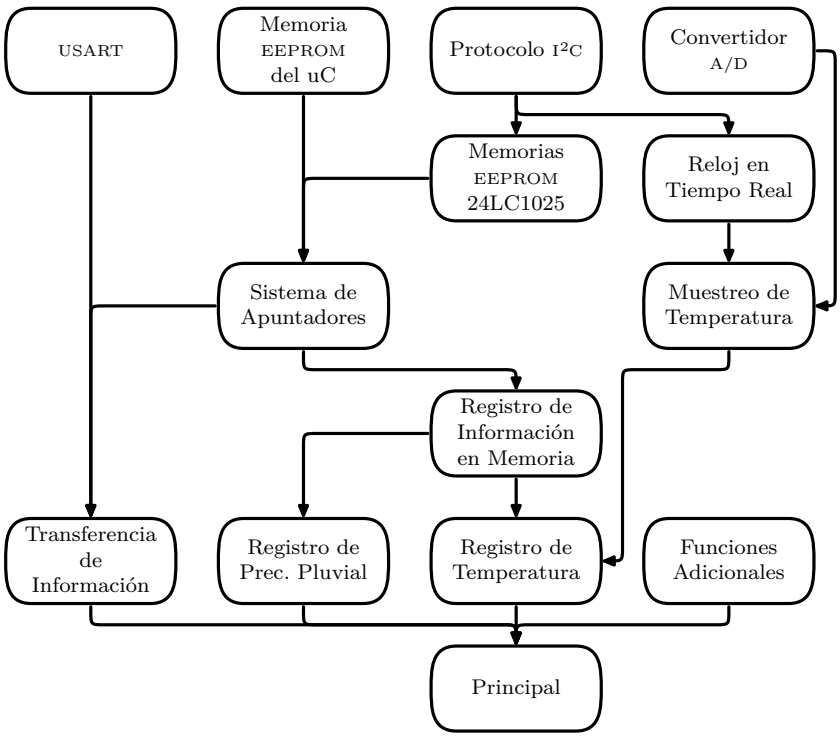

Fig. 4.17. Estructura del *firmware*.

Una vez hecho lo anterior, se compiló el programa y se programó en el microcontrolador para que a partir de la ejecución de éste el microcontrolador pusiera en funcionamiento al *datalogger*.

En el siguiente capítulo estudiaremos el programa de la computadora que al ejecutarse en una PC realizará la recepción de los datos del *datalogger* y los manejará para poder ser presentados al usuario.