Practica Profesional Supervisada

Informe de las 100 horas

Plataforma concentradora de sensores y eventos digitales

Autores: Ignacio Sambataro, Luciano Mantovani

Tutor: PhD. Ing. Orlando Micolini
Supervisor: Ing. Maximiliano Eschoyez

Laboratorio de Arquitectura de Computadoras
Facultad de Ciencias Exactas, Fisicas y Naturales
Universidad Nacional de Cordoba

Año 2015

Resumen

Este informe contiene un resumen de las actividades realizadas para la practica profesional supervisada en las primeras 100 horas de trabajo. El proyecto a realizar es una plataforma concentradora de sensores y eventos digitales

Índice

1. Introduccion **3**

2. Requerimientos **3**
2.1. Principales . . . 3
2.2. Secundarios . . . 3

3. Investigacion **3**
3.1. Parametros tenidos en cuenta en la seleccion del microcontrolador . . . 5
3.2. Selección . . . 5

4. Silicon Labs *C8051F352* **6**
4.1. Conversor Analogico-Digital . . . 6
4.1.1. Modos de medicion . . . 6
4.1.2. Modos de conversion . . . 6
4.2. Contadores . . . 6
4.3. Interfaz Serial . . . 6
4.4. Flash . . . 7

5. Diseño de Software **7**
5.1. Descripcion grafica del sistema . . . 7
5.1.1. Diagrama de bloques . . . 7
5.1.2. Diagramas de caso de uso . . . 8
5.1.3. Diagramas de secuencia . . . 9

6. Diseño de Hardware **11**
6.1. Diagrama Esquematico . . . 11

7. Avances con respecto al diseño de software **11**

8. Estructura del software **11**
8.1. Partes que conforman el software . . . 11
8.1.1. Configurador . . . 12
8.1.2. Principal . . . 12
8.1.3. Conversor . . . 12
8.1.4. Interfaz de usuario . . . 12
8.1.5. Contador . . . 12
8.1.6. Interrupciones . . . 13

Appendices **14**

A. Instrucciones MML **14**
A.1. SSE (Set Single Ended) . . . 14
A.2. SDI (Set Diferencial) . . . 14
A.3. SGA (Set Ganancia) . . . 14
A.4. GT0 (Get Timer 0) . . . 14
A.5. GT2 (Get Timer 2) . . . 14
A.6. ST (Start) . . . 15

1. Introduccion

Esta practica esta orientada al diseño y la construccion de un sistema embebido que concentre las señales de varios sensores y varias fuentes de eventos digitales. La idea es que un sistema embebido de uso especifico pueda tercerizar la tarea de obtener, convertir y procesar una señal de uno o varios sensor o una o varias fuente de eventos digitales.

Surgio de la problematica de algunos proyectos dentro del Laboratorio de Arquitectura de Computadoras que compartian el mismo problema. La necesidad de un sistema que genericamente obtenga las señales de los sensores y la pueda transmitir al sistema principal, ya convertidas. Ademas de un contador de eventos que no requiera del uso de interrupciones por software.

En este informe cubrimos los avances hechos en las primeras 100 horas de trabajo en el proyecto.

2. Requerimientos

2.1. Principales

- Leer de 4 a 8 señales analógicas y convertirlas a digital.
- Contar eventos con 3 o 4 contadores distintos.
- Transmitir los datos digitales a través de un protocolo serial a alguna otra placa o procesador.

2.2. Secundarios

- Lograr el menor consumo posible.
- Buscar la mejor inmunidad al ruido, con una distancia de la placa a los sensores de hasta un máximo de 10 metros.
- Lograr un producto lo más pequeño posible.
- Lograr un producto programable

3. Investigacion

La etapa de investigacion consistio en encontrar un microcontrolador que satisfaga la mayor cantidad de requerimientos principales. El sistema entero consiste en interactuar con el nucleo, que es el microcontrolador, por lo que esta etapa requierio de analisis detallado de las opciones con las se contaba. En el cuadro1 se pueden ver los microcontroladores considerados en la etapa de seleccion.

Fabricante	Modelo	RAM(K)	canales ADC	Referencia	Resolucion	ganancia	Contadores	low power	puerto serie	Dimension (')	Pins	Us$
Intel	8XC51GB	256	8	GND	8 bits	no	3 (16 bits)	si	salida y entrada RS232	N	32	N
Silicon Labs	C8051F352	768	8	DIF/GND	24 bits	128x	4 (16 bits)	si	Smbus/I^2C, UART, SPI	0,35x0,35	32	2,3
Atmel	AT89C5115	256	8	GND	8/10 bits	no	3 (16 bits)	si	UART (3 modos Full Duplex)	0,34x0,34	32	10
Microchip	PIC18F4550	32	13	GND	10 bits	no	4(8 y 16 bits)	si	SPI, I^2C, UART/USART, USB	0,47x0,47	44	5,36
Nec	PD78C17	1024	8	GND	8 bits	no	2 (8 bits)	si	Msbus/I^2C	0,92x0,70	64	N
Maxim	DS4830	1024x16	16	DIF/GND	13 bits	no	2(16 bits)	si	SPI, I^2C	0,2x0,2	40	7,5
NXP	LPC1110	4	8	GND	10 bits	no	2(32 bits)	si	I^2C, UART, Soporte RS-485	0,42x0,51	20	2,5
Atmel	ATSAM3A8C	256	16	DIF/GND	12 bits	no	9(32 bits)	si	USB, SPI	0,63x0,63	63	2,4
Atmel	ATSAM3S1A	64	8	DIF/GND	10/12 bits	1x,2x,4x	3(16 bits)	si	USB, I^2C, SPI	0,35x0,35	44	2,5
Atmel	ATSAM3S1C	16	16	DIF/GND	10/12 bits	1x,2x,4x	6(16 bits)	si	USB, I^2C, SPI	0,63x0,63	74	2,5
Atmel	ATSAMD21J	256	20	DIF/GND	12 bits	16x	5 (16 bits)	si	1 USB 2.0 + 6 I^2C/USART/SPI	0,47x0,47	64	3
Atmel	ATSAMD21G	256	14	DIF/GND	12 bits	16x	3 (16 bits)	si	1 USB 2.0 + 6 I^2C/USART/SPI	0,35x0,35	48	2,5
Atmel	ATSAMD21E	256	10	DIF/GND	12 bits	16x	3 (16 bits)	si	1 USB 2.0 + 4 I^2C/USART/SPI	0,35x0,35	32	2,5
Texas Instr	MSP430F5340	64	9	GND	12 bits	2x	7 (distintas)	si	SPI, I^2C, UART	0,3x0,3	48	3,3
ST	STM32F373CX	256	4	GND	12, 16 bits	32x	17 (distintas)	si	2 I^2C, 3 SIP, 3 USART, 1 USB	0,35x0,35	48	2,5
Atmel AVR	ATmega128	128	8 (2 c/gain)	7 DIF, 8 GND	10 bits	1x, 10x, 200x	4 (8 y 16)	si	USART, SPI	0,6x0,6	64	8

Cuadro 1: Microcontroladores considerados

3.1. Parametros tenidos en cuenta en la seleccion del microcontrolador

- **RAM:** No es un requisito principal, pero en caso de tener que decidir entre dos micros similares, el tamaño de la memoria puede ser un factor para tomar la decision final
- **Cantidad de canales del ADC:** Mientras mas canales se tengan, mas señales analogicas de entrada pueden haber, y mas señales de sensores se podran procesar simultaneamente
- **Referencia:** Nos dice si los pines del ADC se pueden usar como entrada diferencial o unicamente con referencia a GND. Esto es porque en el caso que haya 16 pines para el ADC y puedan usarse todos como entrada diferencial, se podrann usar como maximo la mitad de los pines, es decir 8.
- **Resolucion:** Es la cantidad de bits con la que se representa el dato convertido. A mayor resolucion, mayor presicion de la conversion.
- **Ganancia:** Una buena ganancia interna en el micro es necesaria para una amplificacion de la señal. evitando la mayor cantidad de ruido posible. Este parametro es clave si se quiere trabajar con sensores que funcionan a voltajes muy pequeños en ambientes susceptibles al ruido electrico.
- **Contadores:** Cantidad de timers en el microcontrolador que se utilizarian como contadores de eventos(es necesario que puedan ser clockeados por fuente externa, es decir, que la fuente que incrementa el contador provenga de eventos externos y no interiores al microcontrolador).
- **Modos de bajo consumo:** Si tiene mas de un modo de bajo consumo, es mas simple lograr que el sistema se encuentre el mayor tiempo posible consumiendo lo menor posible.
- **Puerto serie:** Interfaces seriales que soporta el micro. Minimamente se necesita que soporten I^2C y UART.
- **Dimension:** Dimension del micro. El tamaño de la placa deberia ser lo menor posible por lo que mientras mas pequeño el micro, mejor.
- **Cantidad de pines:** Dependiendo del encapsulado, habra una cantidad de pines. La cantidad puede afectar el tamaño y la complejidad de la placa.
- **Precio:** Costo en dolares del integrado.

3.2. Selección

En el momento, habia en el laboratorio una placa de desarrollo de Silicon Labs con el microcontrolador C8051F352. Este mismo fue considerado dentro de las elecciones posibles, como puede verse en el cuadro 1.

Ademas de la ventaja de tenerlo en el mismo laboratorio, la placa de Silicon Labs tiene la particularidad de tener una buena ganancia maxima (128x) a la entrada del ADC, lo cual lo distingue del resto de los microcontroladores analizados. Ademas de esto, cumple con el resto de los requisitos propuestos por nuestro tutor, por lo que consistia en una buena eleccion.

Habiendo hecho este analisis, se decidio optar por utilizar el microcontrolador C8051F352 de Silicon Labs.

4. Silicon Labs *C8051F352*

En esta seccion se explicaran brevemente las funcionalidades que se utilizaron del microcontrolador elegido. Para informacion mas detallada referirse al datasheet del microcontrolador. [1]

4.1. Conversor Analogico-Digital

El C8051F350 incluye un ADC Sigma-Delta completamente diferencial de 16 bits con capacidad de calibracion interna, con capacidad de 8 mediciones simultaneas en modo singular, y 4 mediciones simultaneas en modo diferencial. Tiene dos filtros separados de decimacion programables con un throughput de hasta 1KHz. Tiene un voltaje de referencia interno de 2.5 V, y admite la administracion de voltajes de referencia externos. Cada canal puede ser amplificado 1, 2, 4, 8, 16, 32, 64, o 128 veces.

4.1.1. Modos de medicion

- **Singular**: Entrada independiente medida con respecto a GND.
- **Diferencial**: Dos entradas medidas con respecto una de otra.

4.1.2. Modos de conversion

- **Singular**: Indica al ADC que genere informacion suficiente para producir un resultado luego de un ciclo de conversion. El filtro puede ser *fast-filter* o *SINC3*. En el primero, el resultado esta disponible luego de un solo ciclo de conversion del ADC. En el segundo, espera 3 ciclos antes de que este disponible el resultado.
- **Continua**: El ADC comienza una nueva conversion inmediatamente despues de terminar la anterior. Los filtros funcionan de manera analoga al modo singular.

4.2. Contadores

El C8051F350 cuenta con 4 contadores: Timer0, Timer1, Timer2 y Timer3. Timer0 y Timer1 cuentan con 3 modos de operacion:

- Contador de 13 bits
- Contador de 16 bits
- Contador de 8 bits con valor de retorno

Timer0 cuenta con un modo mas que es el de separarse en dos contadores independientes de 8 bits. Timer2 y Timer3 tienen 2 modos:

- Contador de 16 bits con valor de retorno
- Dos contadores independientes de 8 bits con valor de retorno

Todos los timers pueden ser clockeados por fuente interna o externa. En el caso de los timers 2 y 3 la fuente externa se divide por 8.

4.3. Interfaz Serial

El C8051F352 cuenta con dos protocolos de transferencia serial de datos: SMBus y UART. **SMBus** es una interfaz de dos cables bi-direccional. Es compatible con I^2C, es por esto que se considera que este microcontrolador cuenta con I^2C por el hecho de tener SMBus. **UART** es una interfaz serial asincrona full-duplex.

4.4. Flash

Incluye una memoria Flash de 8 Kilobytes reprogramable dentro del chip para almacenamiento de datos no volatil. Es programable mediante interfaz C2 o por software

5. Diseño de Software

Luego de la eleccion de un microcontrolador, se inicio el proceso de diseño. las figuras , , , muestran los diagramas que describen el sistema segun los requerimientos.

5.1. Descripcion grafica del sistema

5.1.1. Diagrama de bloques

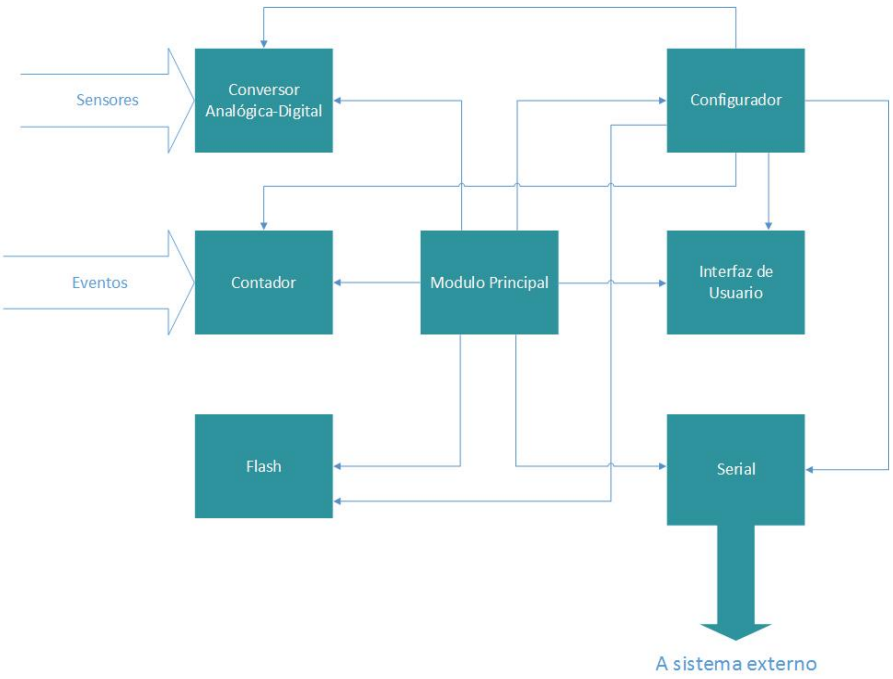

Figura 1: Diagrama de Bloques 1

El sistema esta compuesto por 7 bloques o modulos separados, manejados por un modulo principal. En la figura 1 se pueden observar estos bloques y la interaccion que tienen entre si.

- El **Bloque Principal** se encarga ejecutar las funciones del resto de los modulos para dar arranque y ejecucion al sistema.
- El **Bloque Conversor Analógco-Digital** principalmente obtiene los datos de los sensores, los procesa, y los envia al modulo principal. Ademas de esto, configura el funcionamiento del ADC segun los parametros dados por el usuario. El usuario puede

elegir la cantidad de pines que va a utilizar como entrada segun la cantidad de sensores que quiera medir, puede elegir un nivel de ganancia de amplificacion de la señal antes de la conversion, y puede tambien elegir el modo de obtencion de los datos (diferencial o single-ended).

- El **Bloque Contador** se encarga de obtener los valores en los contadores de eventos.
- El **Bloque de Interfaz de Usuario** en este modulo se levanta la interfaz con la que interactua el usuario para establecer los parametros configurables del sistema.
- El **Bloque Configurador** interactua directamente con el hardware del microcontrolador. Realiza todas las configuraciones necesarias para poder hacer funcionar cada modulo. Es por esto que en el diagrama de bloques se puede ver que este modulo interactua directamente con todos los demas. Inicializa todos los registros pertinentes, el clock del sistema y setea los puertos de entrada y salida.
- El **Bloque Serial** envia los datos por interfaz serial. Puede ser UART o I^2C.
- El **Bloque Flash** Maneja la unidad de memoria no volatil del sistema. Se utiliza para guardar y cargar configuraciones hechas por el usuario, de forma que puedan cargarse automaticamente al inicio del sistema sin necesidad de volver a configurarlo cada vez que se inicia.

5.1.2. Diagramas de caso de uso

En la figura 2 se muestra un diagrama de caso de uso del sistema. Los casos de uso son bastante intuitivos, el usuario debe poder configurar todos los aspectos claves del sistema. Este diagrama muestra a grandes rasgos las acciones posibles que pueden hacerse sobre el sistema desde el punto de vista del usuario.

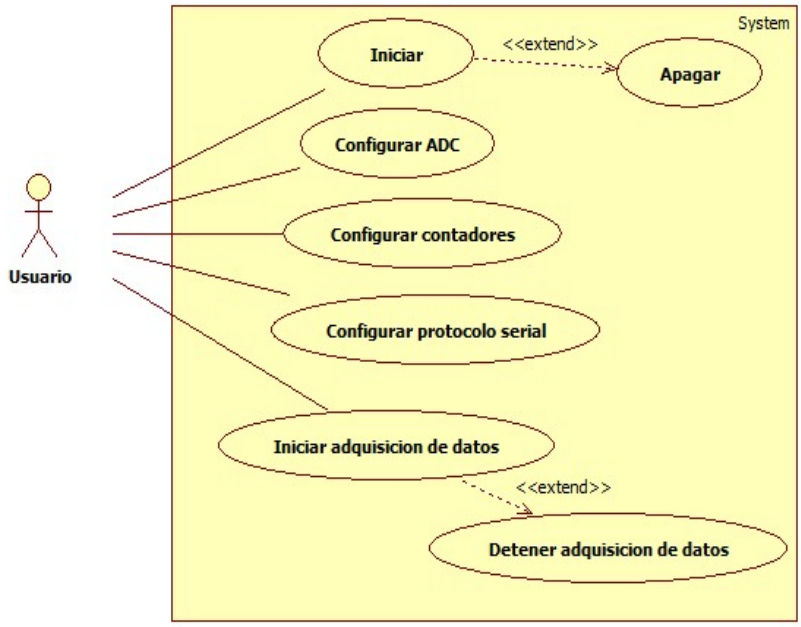

Figura 2: Caso de uso 1

5.1.3. Diagramas de secuencia

Los diagramas de secuencia modelan distintas interacciones entre los componentes de un sistema. En este caso, los dos componentes mas importantes del sistema son el usuario, y el programa principal que recibe las peticiones del usuario a traves de la interfaz grafica, y procesa los pedidos llamando a funciones de otros bloques del sistema. En el primer diagrama (figura 3) se modelo una configuracion de un pin especifico para medir una entrada analogica. El programa principal, en este caso, debe llamar a funciones del bloque conversor para configurar el pedido del usuario. Luego, el usuario habilita el comienzo de conversion para que el sistema envie los datos ya digitales por interfaz serial.

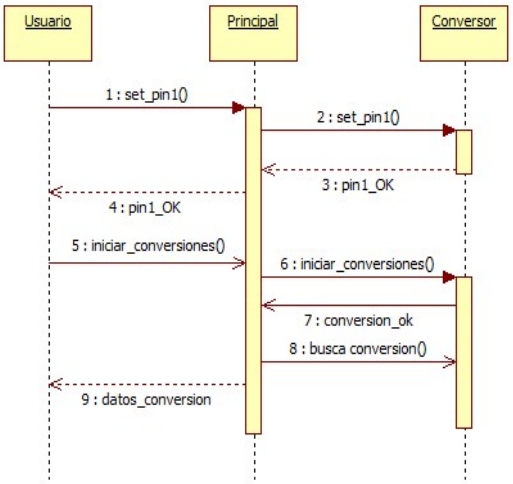

Figura 3: Diagrama de secuencia 1

El diagrama en la figura 4 muestra una configuracion de un contador. En este sistema, los contadores se inician junto con el arranque mismo del sistema y desde ahi mismo comienzan a contar los eventos que ocurran en el pin que tienen asignado. Es por esto que lo unico que hay que hacer es consultar el valor en los registros asociados al contador para obtener la cuenta actual.

El diagrama 5 muestra una obtencion y un guardado de datos de configuracion en la memoria flash del microcontrolador. Los datos de configuracion son unicamente los pertenecientes al conversor analogico-digital.

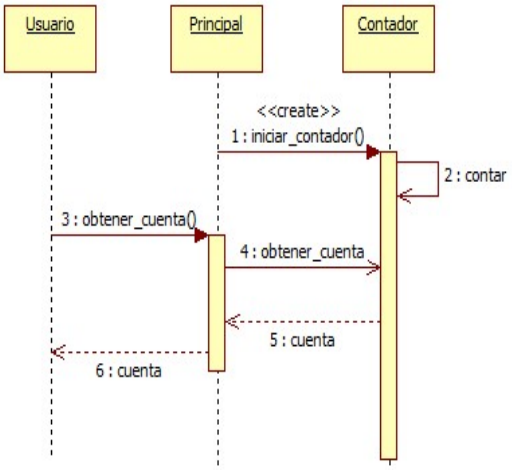

Figura 4: Diagrama de secuencia 2

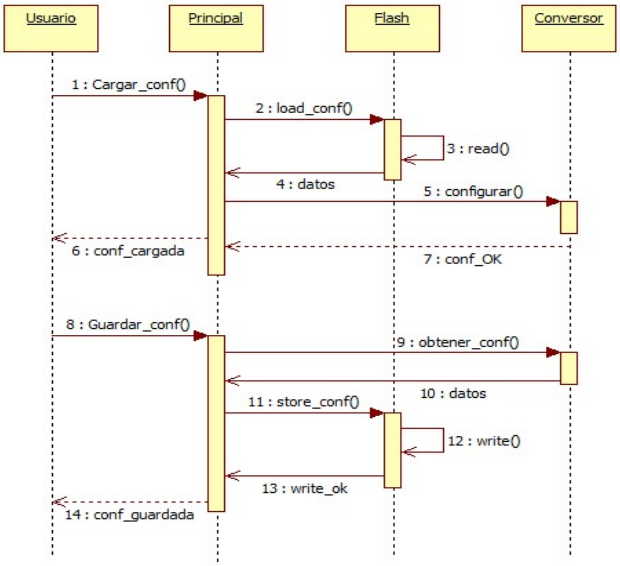

Figura 5: Diagrama de secuencia 3

6. Diseño de Hardware

Una vez seleccionado el microcontrolador, se inició el diseño. Gracias a que en el Laboratorio se encontraba la placa de Silicon Labs (C8051F350DK), con el micro que seleccionamos para trabajar. Procedimos a buscar y analizar el diagrama esquematico de la placa. Una vez determinado los bloques necesarios para nuestra placa, procedimos a realizar nuestro diagrama esquematico. Para esto se utilizó un programa llamado KiCad, ya que nos exporta los archivos (Gerbers) que posteriormente nos serviran para poder imprimir la placa en la fresadora.

6.1. Diagrama Esquematico

En la Figura NUMERO se muestra el esquematico del total a desarrollar. Podemos ver que la placa en principio tendria un microcontrolador C8051F352, que trabaja con un nivel de tension de 3,3V. La entrada de alimentacion tiene un Jack Power al que se le conectara una fuente de 9 voltios y 500 miliAmeperes, entonces se le agrega a la placa un regulador de tension LM2937. Desde el regulador sacamos 3 lineas distintas (con el mismo valor de tensión) de alimentación, una irá hacía Vdd del 8051, otra hacia AV del 8051 y una tercera que se utilizara para alimentacion de un MAX232 y un Led.

7. Avances con respecto al diseño de software

En las primeras 100 horas no fue posible cubrir los requerimientos principales. Los avances obtenidos hasta hoy son los siguientes

- Es posible configurar las entradas del ADC en modo single-ended y bipolar
- Es posible dar una ganancia de 2 a 128 para cada entrada del conversor
- Es posible contar eventos con dos contadores distintos
- Es posible modificar la tasa de envio de datos por UART para cada señal convertida por separado

8. Estructura del software

El programa esta dividido en 6 modulos que en la mayoria de los casos corresponde a cada bloque en el que esta dividido el sistema, como se muestra en la figura 1

8.1. Partes que conforman el software

En este momento, el programa entero esta compuesto por 6 modulos. Un modulo principal y otros 5 que definen funciones segun los bloques que describen el comportamiento del sistema. El modulo principal ejecuta las funciones de los modulos secundarios.

Modulos:

- Principal
- Conversor
- Contador
- Interrupciones
- Configurador
- Interfaz de Usuario

8.1.1. Configurador

El configurador se encarga de inicializar todos los parametros necesarios que permiten la operatividad del resto de los modulos. Establece los valores correspondientes para todos los registros pertinentes y configura los puertos seleccionados en el modulo del conversor (Sección 8.1.3) cuando se especifique por parte del usuario.

8.1.2. Principal

El modulo principal es la funcion main, inicializa todo el sistema usando funciones del configurador (Seccion 8.1.1) y hace el loop infinito que corre el sistema indefinidamente, obteniendo los comandos del usuario via la interfaz (Seccion 8.1.4).

8.1.3. Conversor

El modulo de conversion se encarga de la etapa de obtencion y procesado de las señales convertidas del ADC (Sección 4.1).

Las funciones de este modulo se encargan de que la obtencion de los datos se corresponda con la configuracion dada por el usuario. Es por esto que tiene funciones que se activan en la etapa de configuracion del sistema que preparan al mismo para una obtencion de datos conforme a los ajustes hechos por el usuario. Estos ajustes se realizan mediante las instrucciones MML mencionadas en la seccion 8.1.4 y explicadas con detalle en el apendice A.

8.1.4. Interfaz de usuario

En un principio, se comenzo con una interfaz de usuario pensada en forma de un menu principal. La idea era que un pueda acceder a todas las opciones de configuracion del sistema desde este menu, ingresando opciones por teclado y cambiando asi los parametros. La ventaja de este metodo es que los errores por parte del usuario se reducen significativamente, dado que no tiene otra opcion que elegir entre las opciones que muestra el menu. La desventaja es la complejidad que implica hacer un sistema altamente configurable con una interfaz de este tipo. Esta desventaja fue finalmente la que hizo que se cambiara la interfaz por completo, ya que luego de algunas semanas, las opciones de configuracion comenzaron a crecer y se hizo evidente que una interfaz de menu hacia la configuracion muy arduosa para el usuario y muy complicada de cambiar y hacer para el programador.Por lo que se cambio a un metodo con mas libertades para el usuario y menos complejidad para el programador.

La interfaz de usuario actual esta hecha con un lenguaje de especificacion denominado "Man Machine Languaje". MML es un lenguaje de especificacion que se usa tipicamente para estandarizar la interfaz de un sistema para el manejo del mismo desde una consola. Siguendo el paradigma de MML, lo que se hace es definir una serie de comandos que pueden aceptar una serie de argumentos. Con cada comando y sus argumentos se conforma una orden que ejecuta el procesador. De esta manera, se pueden lograr instrucciones simples que cambian distintos aspectos de la configuracion del sistema conforme a las intenciones del usuario. Una descripcion completa de las instrucciones hechas hasta el momento se encuentra en el apendice A. Con esto, es posible configurar todos los aspectos del sistema, sabiendo como operan todas las instrucciones y sus argumentos.

Este esquema de interfaz de usuario esta todavia en proceso. Hasta ahora, las instrucciones descriptas en el apendice A no cubre por completo todas las configuraciones que deberian poder hacerse teniendo en cuenta los requerimientos principales del sistema.

8.1.5. Contador

El modulo de contador contiene funciones que devuelven los valores de las cuentas actuales de los contadores en funcionamiento. Por una cuestion de simpleza, los contadores

siempre estan activos, ya sea que se usen o no. Actualmente se cuenta con tres contadores de eventos. El microcontrolador tiene cuatro timers (Sección4.2) pero obligadamente uno de ellos debe ser utilizado por el modulo de la interfaz serial UART (Sección4.3).

Cada contador utilizable tiene una funcion que simplemente se encarga de obtener el valor de la cuenta en su respectivo timer asociado. Las instrucciones definidas en MML(Seccion 8.1.4) descriptas en el apencice A incluyen las instrucciones GT0, GT1, y GT2, que se utilizan para la obtencion del valor de la cuenta actual.

8.1.6. Interrupciones

Este archivo define las rutinas de interrupcion para aquellas funcionalidades que se requiere que realicen interrupciones sobre el microcontrolador para condicionar el comportamiento del programa. Actualmente el conversor analogico-digital4.1 es el unico modulo que realiza interrupciones.

Appendices

A. Instrucciones MML

A.1. SSE (Set Single Ended)

Formato: SSE,[1]
Descripcion: Establece el pin ingresado en modo single ended.
Limitaciones:

- Un solo argumento
- El argumento es un byte par comprendido entre 0 y 7

Ejemplo:
SSE,4: Establece el pin 4 del ADC en modo single ended

A.2. SDI (Set Diferencial)

Formato: SDI,[1]
Descripcion: Establece el pin ingresado y el pin siguiente a ese en numero en modo diferencial
limitaciones:

- un solo argumento
- el argumento es un byte par comprendido entre 0 y 6

ejemplo:
SDI,2: setea los pines 2 y 3 en modo diferencial

A.3. SGA (Set Ganancia)

Formato: SGA,[1]
descripcion: Establece la ganancia del ADC segun el argumento a la potencia de 2
Limitaciones:

- Un solo argumento
- El argumento es un byte comprendido entre 0 y 7

Ejemplo:
SGA,3: Establece la ganancia en $2^3 = 8$

A.4. GT0 (Get Timer 0)

Formato: GT0[0]
Descripcion: Obtiene el valor actual de la cuenta de los eventos digitales monitoreados por timer0
Limitaciones:

- No lleva argumentos

A.5. GT2 (Get Timer 2)

Formato: GT2[0]
descripcion: Obtiene el valor actual de la cuenta de los eventos digitales monitoreados por timer2
Limitaciones:

- No lleva argumentos

A.6. ST (Start)

Formato: ST[0]
Descripcion: Guarda los cambios, sale del modo de configuracion y comienza a correr el programa
Limitaciones:

- No lleva argumentos

Referencias

[1] Silicon Laboratories *C8051F351/2/3* 8K ISP Flash MCU Family