



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE INGENIERÍA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

LUIS ESTEBAN SERRANO BERMÚDEZ

TUTOR

DR. ... TUTOR



CIUDAD UNIVERSITARIA, CD. Mx., 2020

*Dedicatoria ...*

# Agradecimientos

# Resumen

# Índice general

<b>Agradecimientos</b>	<b>II</b>
<b>Resumen</b>	<b>III</b>
<b>Prólogo</b>	<b>VIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Computadora . . . . .	1
1.1.1. Procesador . . . . .	1
1.1.2. Arquitecturas de Procesadores . . . . .	2
1.2. Microprocesador . . . . .	4
1.3. Microcontrolador . . . . .	5
1.3.1. Elementos Principales . . . . .	6
1.3.2. Arquitecturas CISC y RISC . . . . .	9
1.4. PIC32MZ2048EFM100 . . . . .	10
1.5. Medidor de Energía ADE7880 . . . . .	12
1.5.1. Gestion de Energía . . . . .	12
1.5.2. Procedimiento de Encendido . . . . .	14
1.5.3. Hardware Reset . . . . .	15
1.5.4. Software Resest . . . . .	15
1.5.5. Operación del ADE7880 . . . . .	16
1.5.6. Canal de Corriente del ADC . . . . .	18
1.5.7. Canal de Voltaje del ADC . . . . .	20

1.5.8. Procesador Digital de Señales . . . . .	21
1.5.9. Interrupciones . . . . .	22
1.5.10. Interfaces de Comunicación . . . . .	23
1.6. Entorno de Desarrollo MPLABX . . . . .	25
1.6.1. Herramienta de configuración MPLAB Harmony . . . . .	25
1.7. Firmware . . . . .	25
1.7.1. Inicialización . . . . .	25
1.7.2. Configuración del puerto I <sup>2</sup> C . . . . .	26
1.7.3. Configuración del puerto HSDC . . . . .	26
<b>2. Resultados</b>	<b>29</b>
2.1. Lectura de Datos Usando el Canal SPI . . . . .	29
2.2. Lectura de Datos usando el canal HSDC . . . . .	30
<b>3. Conclusiones</b>	<b>33</b>
<b>4. Glosario</b>	<b>34</b>

# Índice de figuras

1.1. Arquitectura Harvard . . . . .	2
1.2. Arquitectura Von Neumann . . . . .	3
1.3. Puertos de Entrada y Salida . . . . .	3
1.4. Memoria y Puertos I/O . . . . .	4
1.5. Esquema básico general de un microprocesador . . . . .	5
1.6. Esquema de bloques de un microprocesador . . . . .	6
1.7. Bloque WatchDog Timer . . . . .	7
1.8. Encendido del ADE7880 . . . . .	14
1.9. Entradas analógicas de corriente . . . . .	16
1.10. Amplificadores de Ganancia . . . . .	16
1.11. Entradas analógicas de Voltaje . . . . .	17
1.12. Convertidor Analógico Digital . . . . .	17
1.13. Canal de Corriente . . . . .	19
1.14. Canal de Voltaje . . . . .	21
1.15. Interrupciones . . . . .	23
1.16. Configuración SPI del PIC32MZ . . . . .	27
1.17. Configuración I2C del PIC32MZ . . . . .	27
2.1. Comunicación SPI con los registros de voltaje y corriente . . . . .	30
2.2. Gráficas de la primer prueba de los Canales de Voltaje . . . . .	31
2.3. Gráficas de la primer prueba de los Canales de Corriente . . . . .	32

# Índice de tablas

1.1. Especificaciones PIC32MZ2048EFM100. . . . .	11
1.2. Modos de Suministro de Energía . . . . .	12
1.3. Registros de última escritura . . . . .	23
1.4. Valores de bloqueo del puerto I <sup>2</sup> C . . . . .	26



# Prólogo

Este proyecto se ha desarrollado con la finalidad de añadir una mejora a los medidores de energía previamente realizados en el Instituto de Ingeniería de la UNAM. La finalidad de esta mejora es agregar una función de lectura de datos de registros del medidor de energía ADE7880 para poder analizar una gran cantidad de datos en poco tiempo.

# Capítulo 1

## Introducción

### 1.1. Computadora

Una computadora esta conformada por hardware y software. La parte de Hardware consta de 4 componentes: el procesador, que funciona como el cerebro; la unidad de entrada, por la cual los programas y los datos son ingresados; la unidad de salida por donde son presentados los los resultados y la memoria que es en donde se almacena el software y los datos.

#### 1.1.1. Procesador

El procesador, tambien llamada como la Unidad Central de Procesamiento se puede clasificar en tres partes:

- **Registros:** Es una locación de almacenamiento dentro de la CPU en la cual se mantienen los datos y las direcciones de memoria durante la ejecución de una instrucción. Accesar a los registros de datos es más rápido que acceder a los datos en la memoria externa. Estos registros varían dependiendo del modelo del procesador.
- **Unidad Lógica Aritmética:** Es la calculadora numérica y evaluadora lógica de operaciones. Aquí se reciben los datos provenientes de la memoria principal

o de los registros, realiza una operación lógica y si es necesario reescribe el resultado de vuelta al registro o memoria.

- **Unidad de Control:** Contiene las instrucciones lógicas del hardware, esta se encarga de decodificar y monitorear la ejecución de las instrucciones. También funciona como árbitro de varios de los servicios del CPU, los cuales se encuentran sincronizados por un reloj de sistema.

### 1.1.2. Arquitecturas de Procesadores

#### Arquitectura Harvard

El término de Arquitectura Harvard se le conoce a aquellas configuraciones donde las instrucciones y los datos de entrada/salida se almacenan por separado en 2 memorias, este término se le conoce por la primer computadora digital automática el "*Harvard Mark I*" (1944) diseñado por IBM y la Universidad de Harvard [Cuéllar, 2008].

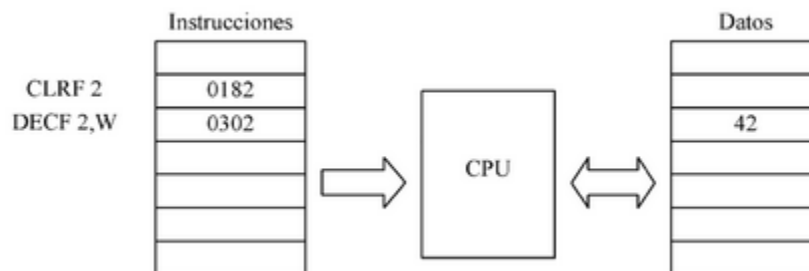


Figura 1.1: Arquitectura Harvard.

La arquitectura Harvard duplica efectivamente el espacio accesible de memoria y la velocidad teórica de ejecución, ya que el procesador puede realizar las operaciones de lectura y escritura sobre los 2 buses a la vez y leer datos e instrucciones en el mismo ciclo [Caprile, 2012].

#### Arquitectura Von Neumann

Esta arquitectura usa la misma memoria para instrucciones como para datos, es decir, utiliza una memoria de lectura/escritura para estos procesos. En esta arquitec-

tura no hay diferencia entre datos e instrucciones, todos son números y solo depende del uso que se le asigne. Esto ha permitido el desarrollo de procesadores simples y relativamente eficientes.

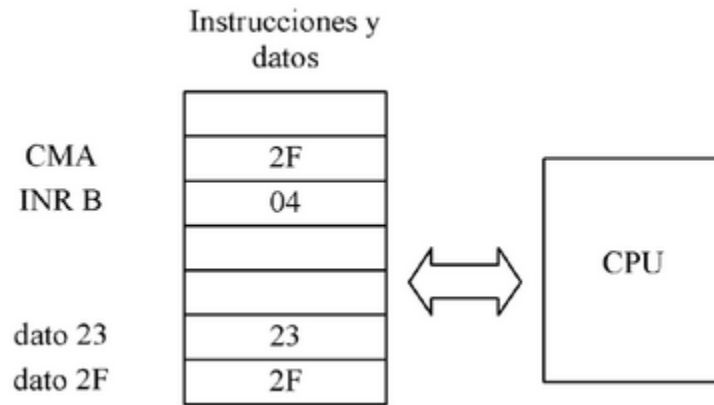


Figura 1.2: Arquietctura Von Neumann.

En estos casos es común el tener un programa para tomar datos externos, almacenarlos como datos en la memoria y ejecutarlos como instrucciones.

Actualmente se tienen microprocesadores de ambas arquitecturas y depende de las necesidades de cada situación en la que se beneficiaá del uso de una u otra arquitectura.

## Puertos

Los puertos son un conjunto de conexiones con las que el procesador puede interactuar con el mundo exterior a través de señales digitales.

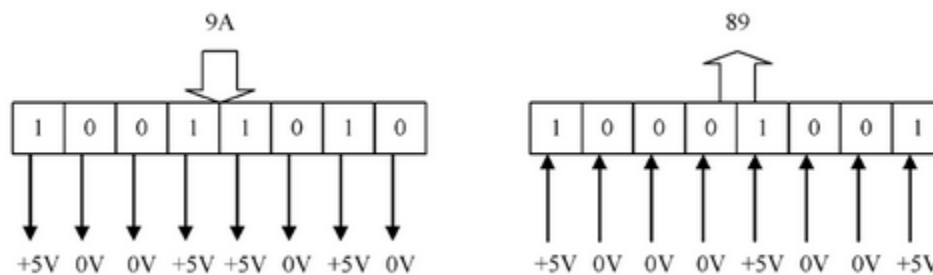


Figura 1.3: Puertos de Entrada y Salida.

Un procesador considera a estos como locaciones de memoria que se escriben o

leen datos. A un puerto que toma datos del exterior se le conoce como puertos de entrada (input) y si envía información al exterior entonces es de salida (output). Generalmente a estos puertos se les conoce como Puertos I/O.

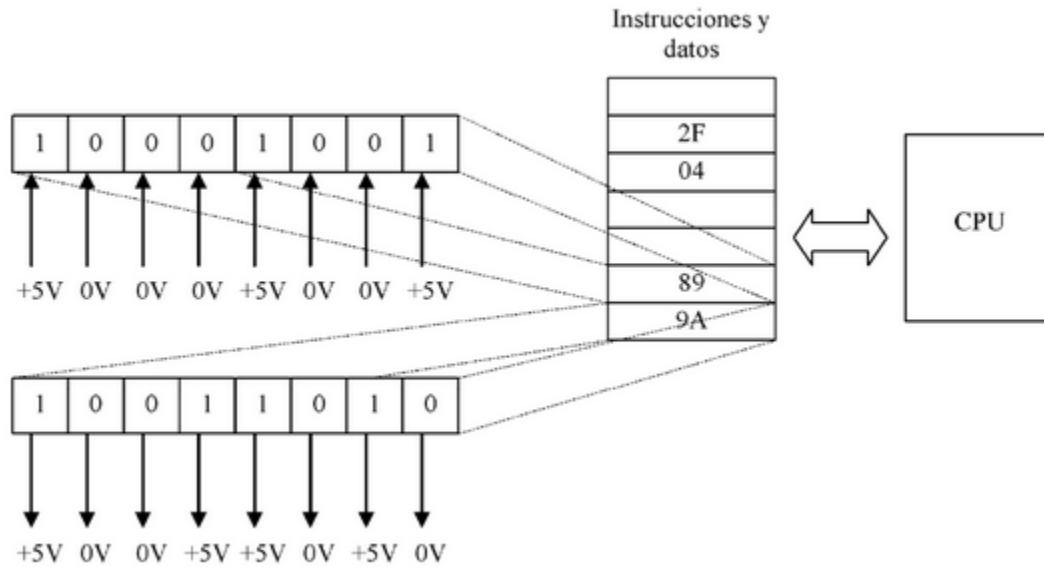


Figura 1.4: Interacción de puertos con la memoria y con el procesador.

## 1.2. Microprocesador

El procesador en una computadora, esta comprendido de varios circuitos integrados, mientras que un microprocesador es un procesador empaquetado en un único circuito integrado. Una microcomputadora usa un microprocesador como su CPU [Valdes and Areny, 2007].

Los microprocesadores vienen en diferentes presentaciones, 4-Bits, 8-Bits, 16-bits, 32-Bits e incluso en 64-Bits aunque estos últimos no son demasiado comunes como los anteriores. El número de bits corresponde al número de dígitos binarios que el microprocesador puede manipular en las operaciones.

El acceso de la memoria principal toma mucho más tiempo que el tiempo de reloj disponible por el CPU, por ello es que los microprocesadores de 32 y 64 Bits poseen una memoria caché de alta velocidad.

### 1.3. Microcontrolador

Una microcomputadora se compone de 3 bloques fundamentales: CPU, Memoria y los puertos de entrada/salida. Estos se conectan entre si mediante grupos de líneas eléctricas denominados buses; los cuales pueden ser de direcciones, si transportan direcciones de memoria, o de datos, si en cambio transportan datos o instrucciones, o de control si estos conducen diversas señales de control.

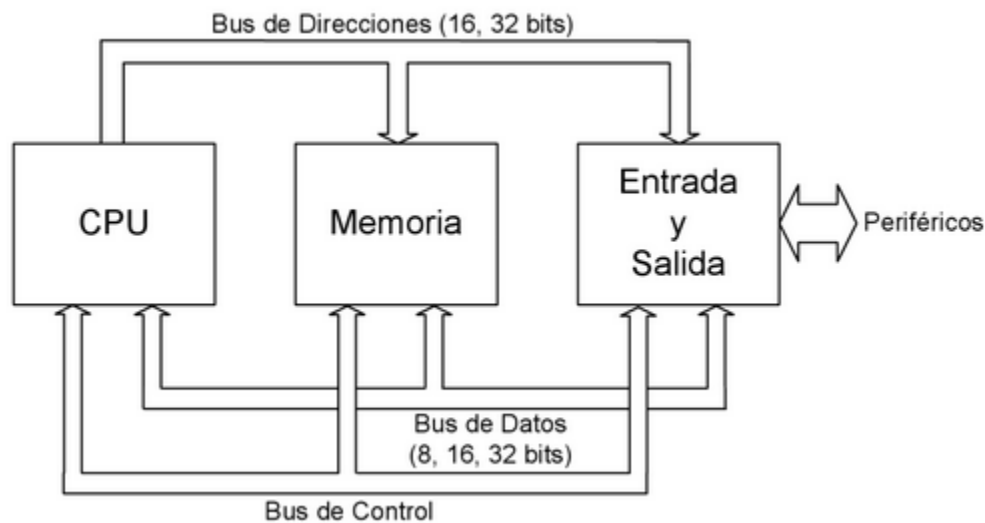


Figura 1.5: Esquema básico general de un microprocesador.

El CPU es el encargado en traer las instrucciones alojadas en la memoria, interpretarlas y hacer que se ejecuten. En una microcomputadora la CPU es el microprocesador por lo cual un Microcontrolador es una Microcomputadora fabricada en un Circuito Integrado.

Los microcontroladores son construidos fundamentalmente para aplicaciones puntuales en automoción, equipos de comunicaciones y telefonía, instrumentación electrónica, equipos médicos e industriales, electrodomésticos, etc. donde se deben realizar un pequeño número de tareas al menor costo posible. En estas el microcontrolador ejecuta un programa almacenado permanentemente en la memoria interactuando con datos almacenados temporalmente e interactúa con el exterior a través de las líneas de entrada/salida. El microcontrolador es parte de la aplicación (Controlador Embebido).

### 1.3.1. Elementos Principales

Además de la microcomputadora (CPU, memoria y líneas de entrada/salida) los microcontroladores disponen de más componentes para poder realizar el funcionamiento requerido por las aplicaciones.

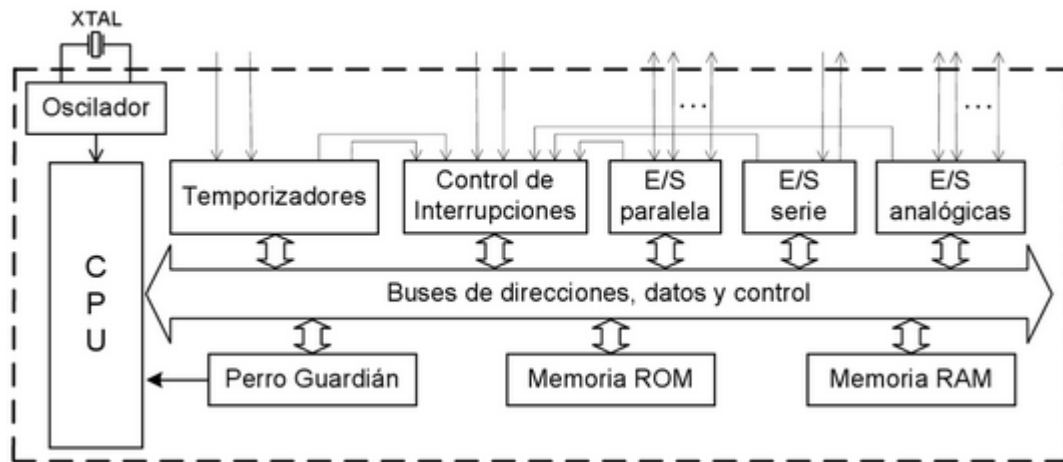


Figura 1.6: Esquema de bloques de un microprocesador.

#### Oscilador

Para poder realizar las operaciones internas el microcontrolador depende de un oscilador que genera los impulsos que permiten la sincronización de todas ellas. En la mayoría de los microcontroladores se utiliza como oscilador a un cristal de cuarzo debido a la gran estabilidad de frecuencia que ofrecen estos cristales ya que de estos dependen la velocidad de ejecución de las tareas.

#### Watchdog Timer (Perro Guardián)

Es un recurso disponible en la mayoría de los microcontroladores, consta de un oscilador (puede ser el principal) y de un contador binario de N-bits. La salida del contador va conectada al circuito *reset* del microcontrolador(1.7). El funcionamiento se define como un contador de pulsos que al llegar al desbordamiento reinicia el dispositivo, el fin de este bloque es que el programador evite el desbordamiento y

por ende el reinicio del dispositivo, por ello se debe reiniciar el contador desde el programador. En cambio si existe un error en alguna parte del código que evita que se reinicie el contador del perro guardián este reiniciará el microcontrolador y será posible retomar el control y redirigir el programa por la ruta correcta.

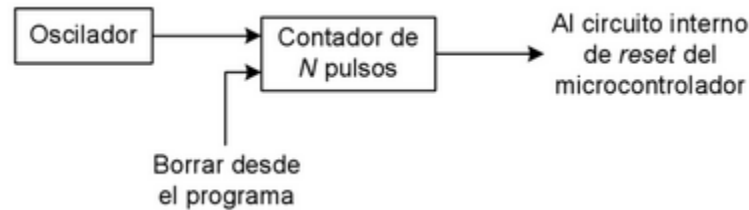


Figura 1.7: Bloque WatchDog Timer.

Es importante que cuando no se limpie el contador WDT a tiempo, se realice la acción de *reset* ya que entonces significa que se ha encontrado un fallo en la secuencia de las instrucciones y para remediarlo se redirecciona a una dirección de memoria determinada y no una aleatoria como podría suceder en estos casos.

En algunos microcontroladores como los PIC se señala la causa del reset por medio de bits de un registro del microcontrolador con lo cual se puede remediar el origen del fallo.

## Reset

El reset es una función con la cual se inician los microprocesadores y los microcontroladores. Esta acción se ejecuta cuando una señal de reset es aplicada de manera manual a una terminal en el CI con lo cual se pone el contador de programa (PC) a un valor predeterminado comunmente 0 haciendo que el microprocesador comience a ejecutar las instrucciones a partir de esa posición.

En un microcomputador la señal de *reset* se genera e forma manual al pulsar un botón o al iniciar el sistema (*Reset* por encendido), sin embargo tambien pueden haber otras fuentes de *reset* como lo son por "Fallo de Alimentación." por "Desbordamiento del WatchDog Timer".

El segundo caso, como se explicó anteriormente sucede cuando el contador WDT se desborda al no reiniciar el contador por lo que el microcontrolador ha perdido



la secuencia de las instrucciones o ha entrado en un bucle demasiado largo. El primer caso sucede cuando el valor de voltaje cae por debajo del umbral de operación momentáneamente con lo que el capacitor se descarga parcialmente y se produce el reinicio del dispositivo.

### **Estado de Bajo Consumo**

Una parte importante en los microcontroladores es la característica de tener un bajo consumo de corriente ya que hay un gran número de aplicaciones que requieren el uso de baterías como medio de alimentación por lo que un bajo consumo de corriente ayuda a una prolongada duración de la batería.

El consumo de corriente en el circuito se sustenta en el uso de compuertas CMOS que al mantenerse en un nivel lógico estático el consumo es prácticamente 0 y solo aumenta cuando el oscilador aumenta la frecuencia de las conmutaciones de los estados de las compuertas lógicas.

Los microcontroladores comunmente se encuentran en este estado y únicamente se activan al ocurrir un evento externo, realizan una tarea y regresan a este estado por lo que se recomienda tener el microcontrolador con bajo consumo de corriente hasta que ocurra un evento que lo saque de ese letargo.

### **Memoria**

La memoria en un microcontrolador es el lugar donde se almacenan el programa que se va a ejecutar y los datos y/o variables que va a utilizar. La memoria de datos es de lectura y escritura y los datos no permanecen en ella una vez que se suspende la alimentación al microcontrolador, es decir, la memoria es volátil (Memorias RAM estáticas). Algunos microcontroladores usan una memoria adicional externa de lectura y escritura no volátil como parte de la memoria de datos, para permitir almacenamiento de datos fijos, para este tipo de memorias se utilizan comunmente EEPROM.

- **RAM:** Esta es un tipo de memoria de lectura y escritura de alta velocidad. En la memoria RAM la información almacenada permanece estable indefinidamente

mientras no se suprima la alimentación del microcontrolador.

- **ROM:** Si se usa memoria ROM ésta se graba durante la fabricación del dispositivo y no se puede alterar una vez almacenada, por ello el programa debe ser depurado. Los microcontroladores que utilizan este tipo de memorias son aquellos que se fabrican en grandes cantidades ya que su fabricación es bastante rentable gracias a su bajo costo.
- **EEPROM:** Las memorias EEPROM son memorias no volátiles de lectura y escritura, donde la escritura se realiza por medios eléctricos y se puede lograr individualmente sin la necesidad de un borrado previo, sin embargo a pesar de ser reprogramables cuentan con un número finito de veces.
- **FLASH:** En las memorias FLASH se pueden realizar las operaciones de lectura y escritura celda por celda, pero a diferencia de las anteriores se requiere primero borrar la información de la celda antes de escribir en ella y estas deben borrarse por bloques de celdas colocando a 0. A menudo las operaciones de escritura de información se debe realizar un proceso de lectura-borrado-escritura del bloque de celdas donde se desea escribir la información. Todas las operaciones de borrado, lectura y escritura se realizan con la misma tensión de voltaje

### 1.3.2. Arquitecturas CISC y RISC

CISC (Complex Instruction Set Computer) y RISC (Reduced Instruction Set Computer) son dos modelos de computadoras visto desde el repertorio de instrucciones que repercuten en el modelo de arquitectura de la CPU y como lo dice su nombre las computadoras CISC tiene una mayor cantidad de instrucciones mientras que el RISC posee una menor cantidad.

Las arquitecturas CISC predominaba en un inicio debido a la ambición de hacer los microprocesadores y microcontroladores los más potente posibles. Debido a la complejidad que fue aumentando lo hizo también la complejidad de la CPU y por ende se le debió dedicar un mayor espacio en el circuito integrado a la decodificación

y ejecución de las instrucciones.

Por el contrario las RISC tienen un repertorio corto de instrucciones sencillas que pueden realizar operaciones simples pero a alta velocidad. La complejidad del CPU disminuye, así que a frecuencia del oscilador puede aumentar y así mejorar la velocidad de ejecución de las operaciones. Esto también apoya en su fabricación siendo más sencillos y baratos de producir, por ello esta arquitectura ha sido la predominante en microcontroladores PIC.

## 1.4. PIC32MZ2048EFM100

Es un microcontrolador de 32-bits que permite la conectividad embebida con una unidad de Punto Flotante, esta familia de microcontroladores poseen una gran cantidad de puertos de comunicación lo que permite que sea una opción bastante adecuada para proyectos en los que se necesite el intercambio de información entre varios dispositivos.[Connectivity, ]

La especificaciones del microcontrolador se pueden ver en la tabla 1.1.

<b>Nombre</b>	<b>Valor</b>
Familia	PIC32MZE
Velocidad Máx. CPU [MHz]	200
Tamaño de Memoria Programable [KB]	2048
SRAM [KB]	512
Auxiliary Flash [KB]	160
Crypto Engine	Sí
Range Temperatura [°C]	-40 to 125
Rango de Voltaje Operacional [V]	2.2 to 3.6
Canales De Acceso Directo a la Memoria	8
Canales SPI	6
Canales I2C	5
Interfaz CODEC (I2S,AC97)	Sí
Selección de Pines Perifericos / Pin Muxing	Sí
Ethernet	10/100 Base-TX Mac
Número de Puertos Ethernet	1
Número de Modulos USB	1
Interfaz USB	Alta Velocidad
Números de Modulos CAN	2
Tipo de Modulos CAN	CAN
Entrada ADC	40
Resolución Máx. ADC (Bits)	12
Máx. Rango de Muestreo ADC [ksps]	18000
Entradas de Captura	9
Comparación de Salida Independiente/PWM Estándar	9
Núm. Máx. de Timers Digitales de 16-bit	9
Puerto Paralelo	PMP
Número de Comparadores	2
Oscilador Interno	8 [MHz], 32 [kHz]
Hardware RTCC/RTC	Sí
Máx. I/O Pins	78
Núm. de Pines	100
Interfaz Quad Serial	Sí

Tabla 1.1: Especificaciones PIC32MZ2048EFM100.

En particular esta familia de microcontroladores posee perifericos que operan a una mayor frecuencia que los microcontroladores tipicos usados para los sistemas embebidos. Esto es posible gracias al microprocesador que posee la CPU el cual contiene varios bloques lógicos que trabajan en conjunto en paralelo, proviendo un alto rendimiento de procesamiento.

## 1.5. Medidor de Energía ADE7880

Este Circuito Integrado es un medidor de energía eléctrica trifásica de alta precisión, con interfaces de comunicación serial I<sup>2</sup>C y SPI. Es adecuado para la medición de energía eléctrica activa, reactiva y aparente en varias configuraciones trifásicas, además posee registros de muestreo de formas de onda para todas las salidas del Convertidor Analógico Digital incluido en el CI<sup>1</sup>.

### 1.5.1. Gestion de Energía

Este CI posee 4 modos de energía determinados por el estado de los pines PM0 y PM1 estos proveen total control sobre el chip y gracias a que poseen resistencias pull-up internas se pueden conectar fácilmente al microcontrolador.

Los modos de operación son:

Modo de Energía	PM1	PM0
PSM0	0	1
PSM1	0	0
PSM2	1	0
PSM3	1	1

Tabla 1.2: Configuración de los pines PM para los modos de suministro de energía.

- **PSM0 (Normal Power Mode):** Este es el modo de operación completamente funcional. Si el medidor de energía se encuentra en cualquiera de los otros modos y se cambia a este todos los registros automáticamente son reiniciados a sus valores iniciales, con excepción del registro LPOILVL<sup>2</sup> y el registro CONFIG2<sup>3</sup>.

Cuando se realiza la transición de un modo de energía a otro el CI cambia el estado del pin  $\overline{IRQ1}$  a bajo y el bit 15 (RSTDONE) en el registro STATUS1 a alto, especialmente este último es el que indica el fin de la transición ya que durante es 0.

---

<sup>1</sup>Circuito Integrado ADE7880 renombrado así a partir de aquí para simplificar.

<sup>2</sup>Registro del límite de sobrecorriente durante el modo PSM2.

<sup>3</sup>Registro de configuración de armónicos.

- **PSM1 (Reduced Power Mode):** En este modo el CI mide los valores absolutos promedio de las 3 fases de corriente para almacenarlos en los registros  $xIMAV$ <sup>4</sup>. Este modo es útil al usar una batería externa. Los puertos de comunicación serial están activos y se puede utilizar para leer los registros  $xIMAV$ , sin embargo, a pesar de poder leer estos registros para los demás registros del CI no se garantizan que los valores sean correctos.

Al entrar en este modo después de haber estado en el PSM0 entonces el cálculo de valor absoluto promedio inicia sin retrasos. Los registros  $xIMAV$  son accesibles en cualquier momento después de que el pin  $\overline{TRQ1}$  ha cambiado a un valor de 0 (indicando el inicio del cómputo de valores absolutos promedio).

- **PSM2 (Low Power Mode):** Los puertos de comunicación no son funcionales en este modo, se reduce el consumo requerido para monitorear la corriente cuando no hay entrada de voltaje y la fuente de voltaje es provista por una batería externa.

Si el pin  $\overline{TRQ0}$  está en un nivel lógico bajo al acabar el periodo de medición, entonces significa que todas las fases de corriente se mantuvieron por debajo del límite y, por lo tanto, no hay corriente fluyendo por el sistema, en este punto el microcontrolador externo pone el CI en modo de espera. Si en cambio el pin  $\overline{TRQ1}$  es el que se encuentra en bajo, entonces significa que por lo menos una entrada de corriente está por encima del límite definido y hay corriente fluyendo por el sistema a pesar de que no hay voltaje presente en los pines del CI. Esta situación se conoce como falta neutra, en este punto el microcontrolador externo coloca el CI en el modo PSM1. No es recomendable usar este modo si los registros de ganancia<sup>5</sup> son diferentes a 1 ó 2.

- **PSM3 (Sleep Mode):** En este modo la mayoría de los circuitos internos se encuentran apagados y el consumo de corriente se mantiene a su más bajo nivel. Los puertos I<sup>2</sup>C, SPI Y HSDC no son funcionales, además los pines de  $\overline{RESET}$ ,

---

<sup>4</sup> $x$ : Canal de corriente A, B, C.

<sup>5</sup>PGA1[2:0].

MOSI/SDA, SCLK/SCI y  $\overline{SS}$ /HSA deben permanecer en alto.

### 1.5.2. Procedimiento de Encendido

El ADE7880 posee un chip interno que monitorea la fuente de alimentación (VDD). En el encendido el dispositivo permanece inactivo hasta que VDD alcanza los  $2.0[V] \pm 10\%$ . Al sobrepasar este límite el monitor mantiene el dispositivo en inactivo por  $26[ms]$  para permitir alcanzar el voltaje de  $3.3[V] \pm 10\%$  a la fuente de voltaje.

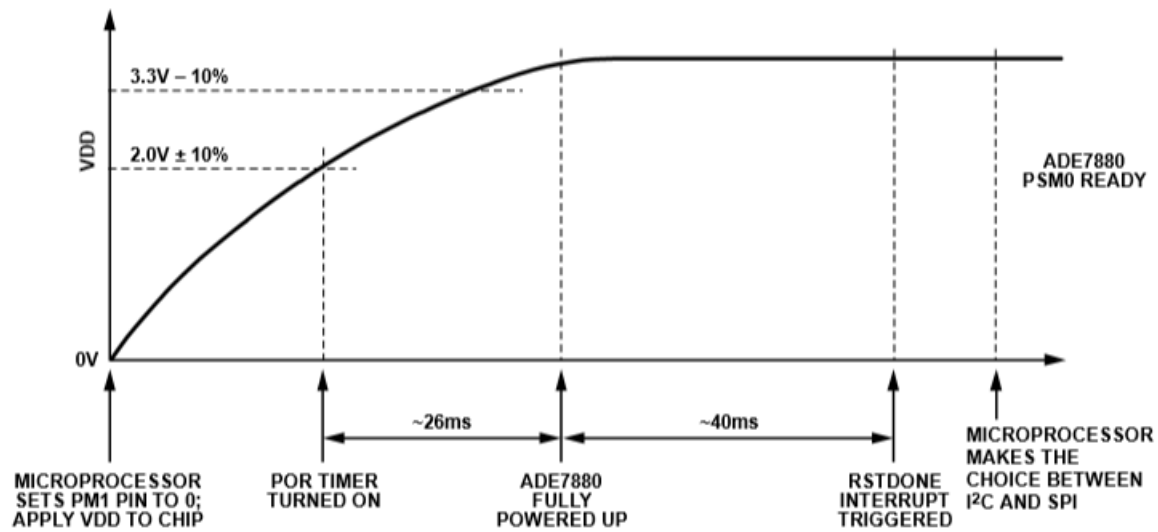


Figura 1.8: Curva de Voltaje de encendido.

Los pines PM0 y PM1 tienen resistencias pull-up, pero es necesario el colocar PM1 en un nivel lógico bajo ya sea por hardware (colocando a tierra) ó a travez del microcontrolador antes de encender el chip, para asegurar que el dispositivo se inicie en PSM0. El tiempo de encendido es de al rededor de  $40[ms]$  tiempo en el cual el pin de  $\overline{RESET}$  debe permanecer en un nivel lógico alto.

Al iniciar en en modo PSM0 el puerto activo es el I²C, si se desea cambiar el puerto de comunicación se tiene que alternar el pin  $\overline{SS}$ /HSA 3 veces de un nivel lógico de voltaje alto a un nivel de voltaje bajo. Cuando se decide usar un puerto de comunicación este debe bloquearse para evitar que durante la comunicación ó cambio de modo de energía se alterne por el otro puerto y se pierda la comunicación. Para ello si el puerto activo es el I²C se debe escribir un 1 el bit 1 (I2C\_LOCK) del registro

CONFIG2 en caso contrario si el puerto activo es el SPI entonces cualquier escritura al registro CONFIG2 bloqueará este puerto como el activo. En ambos casos el cambio por el otro puerto de comunicación no es posible hasta que se haga un reinicio de hardware o se apague y vuelva a encender el sistema.

Inicialmente el CI se encuentra en modo de reposo y, por lo tanto, no ejecuta ninguna instrucción, es en este punto donde todos los registros necesarios para la ejecución del programa deben iniciarse. Si el voltaje cae por debajo de los  $2.0[V] \pm 10\%$  este entra en estado inactivo y ninguna medición ó cómputo es ejecutada.

### 1.5.3. Hardware Reset

Este tipo de reset se logra al tener el CI en el modo PSM0 y colocando el pin de  $\overline{RESET}$  en un nivel de voltaje bajo, en otros modos de energía el pin no tiene función. Si entonces el pin es alternado de alto-bajo-alto con 10[ms] de retraso entre cambio. Durante el proceso de reinicio el pin  $\overline{TRQ1}$  permanece en 1, al termino del proceso este pin es colocado a 0 y todos los registros son devueltos a sus valores iniciales de fábrica.

El reinicio por Hardware devuelve el puerto I<sup>2</sup>C como canal principal de comunicación.

### 1.5.4. Software Resest

Al igual que el reset por Hardware esta opción solo esta disponible durante el modo PSM0. Para iniciar el reset por Software se utiliza el bit 7 en el registro CONFIG, inicialmente este bit se encuentra en 0, y si es cambiado a 1 entonces entra en modo de Reset, el cual regresa a valores iniciales a la mayoría de los registros, la seleccion de puerto de comunicación se mantiene si se ha bloqueado de manera correcta. Los únicos valores que mantienen sus valores son CONFIG2 y LPOILVL.



### 1.5.5. Operación del ADE7880

#### Entradas analógicas

Se tienen 7 entradas analógicas que forman los canales de corriente y voltaje. Los canales de corriente constan de 4 pares de entradas  $I_{xP}/I_{xN}$ <sup>6</sup> diferenciales los cuales tienen un máximo de señal  $\pm 0.5[V]$  e igual con respecto a la señal en AGND<sup>7</sup>.

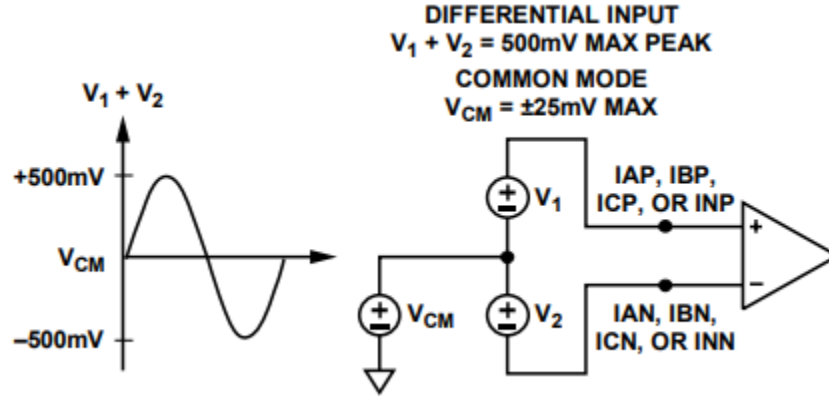


Figura 1.9: Esquema de entradas analógicas de corriente.

Todas las entradas poseen un Amplificador Programable de Ganancia (PGA por sus siglas en inglés) con una selección de 1, 2, 4, 8 ó 16. Las ganancias presentes en IA, IB e IC se colocan en un registro diferente a IN<sup>8</sup> para obtener una ganancia diferente a las 3 primeras.

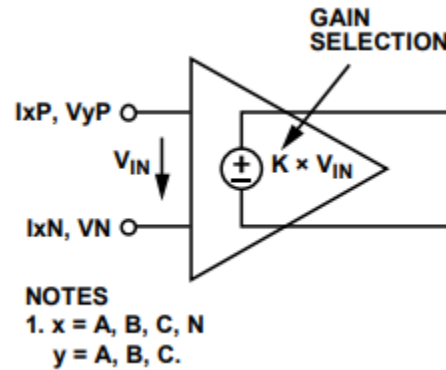


Figura 1.10: Esquema de los amplificadores de ganancia en las entradas analógicas.

<sup>6</sup>x: Canales A, B, C y N respectivamente

<sup>7</sup>Referencia a tierra para el circuito analógico.

<sup>8</sup>Entrada analógica del canal de corriente neutral.

El canal de voltaje tiene 3 entradas de voltaje de un sólo extremo, con una entrada máxima de  $\pm 0.5[V]$  con respecto a  $V_N^9$ , de igual manera estas entradas tienen una entrada de señal máxima de  $\pm 0.5[V]$  con respecto a AGND.

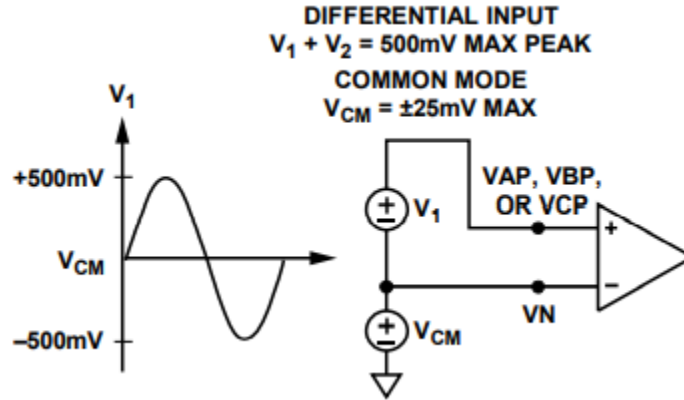


Figura 1.11: Esquema de entradas analógicas de voltaje.

## Convertidor Analógico Digital

A las entradas de los canales de corriente y voltaje se tienen Convertidores Analógicos Digitales  $\Sigma\text{-}\Delta$ , estos se encuentran activos en el modo PSM0, y en el PSM1 solo los que miden los canales de corriente se encuentran activos a excepcion del canal de corriente neutral. En los otros modos no estan activos ninguno de los ADC<sup>10</sup>.

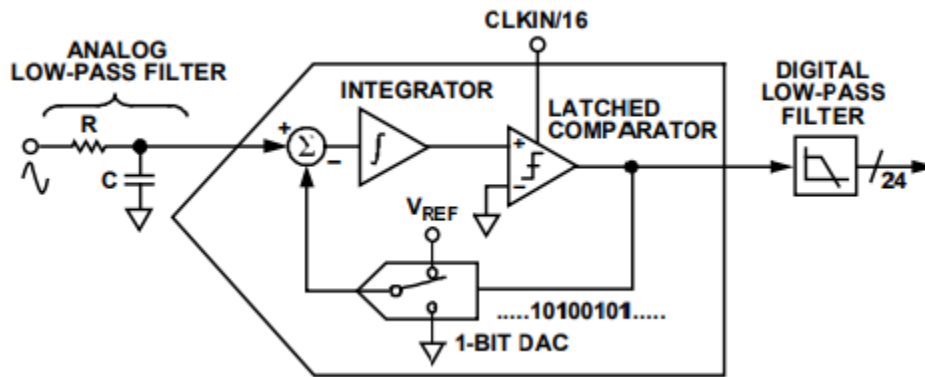


Figura 1.12: Convertidor Analógico Digital.

<sup>9</sup>Entrada analógica del canal de voltaje neutral.

<sup>10</sup>Convertidor Analógico Digital.

El módulo  $\Sigma\text{-}\Delta$  convierte la señal de entrada en un flujo en serie de "1s" "0s".<sup>a</sup> un tiempo determinado por el reloj de muestreo (En el ADE7880 este reloj es de 1.024[MHz]). El DAC<sup>11</sup> de 1-bit en el bucle de retroalimentación es llevado por el flujo de datos en serie, entonces la salida del DAC es sustraida de la señal de entrada. Si el bucle de ganancia es lo suficientemente alto el valor promedio de la salida del DAC puede aproximarse al nivel de la señal de entrada. El promedio es pasado entonces a un filtro digital pasa-bajas que produce una palabra de 24-bits proporcional al nivel de la señal de entrada.

### Filtro Antialias

Este filtro se encuentra a la entrada del ADC y su función es la de prevenir el solapamiento que se presenta en todos los sistemas de muestreo.

Para sensores de corriente convencionales es recomendable usar filtros RC<sup>12</sup> con una frecuencia de corte de 5[kHz] para que la atenuación sea suficiente para eliminar los efectos de solapamiento para sensores de corriente convencionales.

### Función de Transferencia de los ADC

Todos los ADC en el CI están diseñados para producir el mismo nivel de señal de la salida en el código de 24-bits con signo a la salida. El código del ADC puede variar entre 0x800000 (-8,388,608) y 0x7FFFFFFF (8,388,607), esto es equivalente a una señal de entrada de  $\pm 0.787\text{[V]}$ , por lo tanto no debe excederse el rango nominal especificado de  $\pm 0.5\text{[V]}$ . Las señales se muestrean a una velocidad de 8[kSPs]<sup>13</sup>.

#### 1.5.6. Canal de Corriente del ADC

Las salidas que producen el ADC en el canal de corriente son 2 palabras de 24-bits signadas en complemento y se encuentra disponible a 8[kSPs]. Con la señal analógica

---

<sup>11</sup>Convertidor Digital Analógico.

<sup>12</sup>RC: Resistor Capacitor.

<sup>13</sup>kSPs: Mil muestras por segundo (Kilo Samples per second).

específica de entrada de  $\pm 0.5[V]$ , el ADC produce un valor máximo que oscila entre -5,326,737 (0xAEB86F) y +5,326,737 (0x514791).

La entrada IN corresponde a la entrada de corriente neutral de un sistema trifásico. En caso de no contar con línea neutral de corriente se debe conectar a AGND.

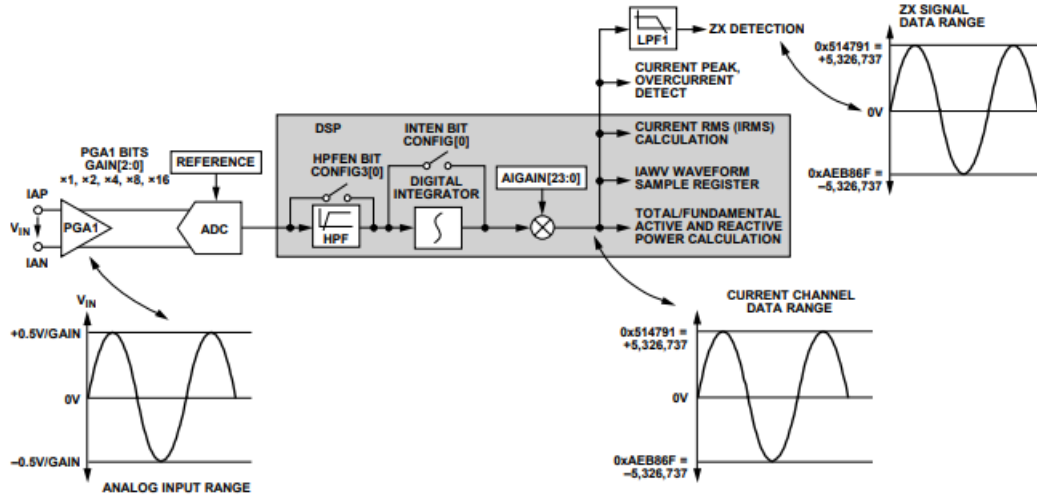


Figura 1.13: Ruta de la señal del Canal de Corriente.

## Registros de Ganancia de Ondas de Corriente

En cada fase hay un multiplicador de señal. La onda de corriente puede ser cambiada a  $\pm 100$  escribiendo en el registro correspondiente. Cuando se cambian estos registros por ende los cálculos de la corriente son modificados, esto es, se afectan las correspondientes fases activa, reactiva ó aparente y el cálculo de corriente RMS así como también se escalonan las muestras de onda.

El DSP<sup>14</sup> trabaja en 28-bits y los registros xIGAIN en 24-bits se acceden como registros de 32-bits y ya que los puertos de comunicación serie trabajan en 8, 16, 32 bits, los registros se accesan como si fueran registros de 32-bits y en el caso del DSP los 4-bits más significativos son colocados con "0"; en el otro caso, los registros xIGAIN además de poner los últimos bits en "0" los bits 24-27 son colocados con el mismo valor que el bit 23.

<sup>14</sup>DSP: Procesador Digital de Señales.

## Filtro Pasa Altas del Canal de Corriente

La salida del ADC pueden tener un offset, el cual puede crear errores en el cálculo de la potencia y los valores RMS de las señales. Los filtros pasa altas (HPF<sup>15</sup>) son colocados en la ruta de la señal de la corriente de fase y neutro y de los voltajes de fase. Si se habilita el HPF elimina cualquier offset de DC en el canal de corriente.

## Muestreo de Canal de Corriente

Las ondas de muestreo del canal de corriente son tomadas de la salida del HPF y almacenados en registros de 24-bits signados IxWV y a una velocidad de 8[kSPs]. Todos los cálculos de RMS y de potencia deben permanecer ininterrumpidos durante este proceso.

El bit 17 del registro STATUS0 es colocado en 1 cuando los registros IxWV están disponibles para la lectura a través de los canales de comunicación serial. Este CI posee un puerto de capturar de datos de alta velocidad (HSDC) diseñado específicamente para proveer rápido acceso a los registros de muestreo de las formas de onda.

### 1.5.7. Canal de Voltaje del ADC

Las salidas del ADC de los canales de voltaje son similares a las de los canales de corriente con 2 palabras en complemento signados a 24-bits. Con la señal analógica de entrada específica de  $\pm 0.5[V]$ , el ADC produce un valor máximo que oscila entre -5,326,737 (0xAEB86F) y +5,326,737 (0x514791).

## Registro de Ganancia de Ondas Voltaje

Al igual que con los registros de ganancia de corriente, existe un multiplicador para la señal que puede cambiado por  $\pm 100$  escribiendo los correspondientes 2 números signados en complemento de 24-bits a los registros de ganancia de ondas de voltaje.

---

<sup>15</sup>HPF: High Pass Filter

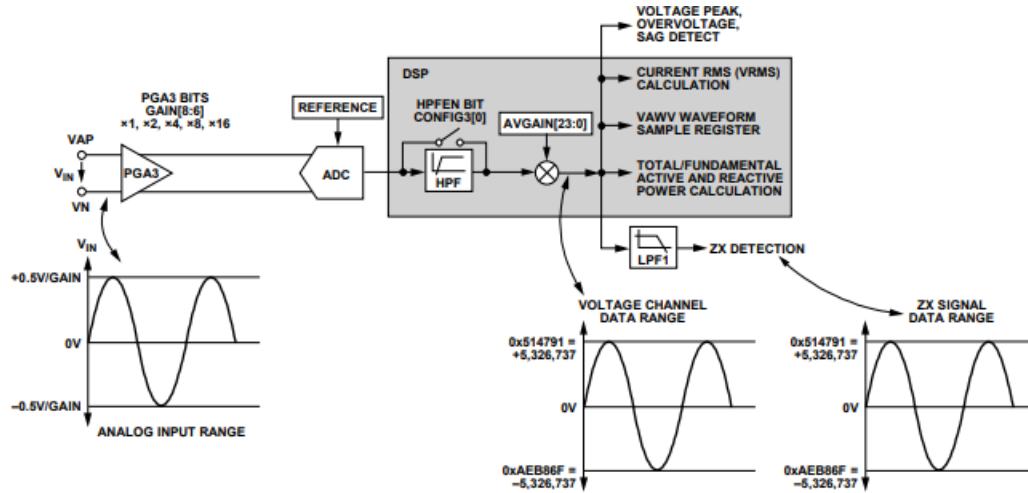


Figura 1.14: Ruta de la señal del Canal de Voltaje.

### Filtro Paso Altas del del Canal de Voltaje

Como se explico con el HPF del canal de corriente, las salidas del ADC pueden contener offsets de DC que pueden crear errores en los cálculos de RMS y potencia. Los HPFs son colocados en en las rutas de la señal de la fase de voltaje, similar a los del canal de corriente. Bit0 (HPFEN) del registro CONFIG3 activa ó desactiva los filtros.

### Muestreo del Canal de Voltaje

Las muestras del canal de voltaje son tomadas a la salida del HPF y almacenados en registros de 24-bits signados a una velocidad de 8[kSPs]. Del mismo modo que con el canal de corriente los cálculos permanecen ininterrumpidos durante el muestreo y se señala en el bit 17 del registro STATUS0 cuando los registros de VxWV están disponibles para ser consultados por los canales de comunicación serial.

### 1.5.8. Procesador Digital de Señales

El ADE7880 posee un Procesador Digital de Señales de función fija que computa todos los valores de potencia y RMS. Contiene memoria de programa ROM y memoria de datos RAM. El programa que se usa para calcular estos valores se guarda en la

memoria ROM y el procesador lo ejecuta cada 8[kHz] y para señalar que ha acabado el compute se coloca el bit 17 en "1.<sup>en</sup> el registro STATUS0, tambien se puede colocar el mismo bit en el registro MASK0 para habilitar la interrupción  $\overline{IRQ0}$ . Los registros usados para el DSP estan localizados en la memoria RAM entre las direcciones 0x438 a 0x43BE y un ancho de 28-bits.

Al usarse un pipelinede 2 etapas es necesario escribir 2 veces la inicialización de un registro para asegurarse que se escriba correctamente en la memoria RAM. Si se requieren 2 ó más registros, para inicializarlos se debe esscribir el último registro 2 veces. Al inicializar el medidor de energía el DSP se encuentra en reposo al igual que los otros con los registros inicializados con "0", para inicializar el DSP se escribe en el registro RUN 0x0001.

Como protección a la integridad de los datos almacenados en la memoria RAM, se tiene un mecanismo de protección contra escritura, deshabilitado inicialmente, para habilitarlo es necesario escribir 0xAD a la dirección 0xE7FE seguido de 0x80 en la dirección 0xE7E3. Para deshabilitarla de nuevo se escribe 0xAD a la dirección 0xE7FE seguido de 0x00 en la dirección 0xE7E3. Cuando el CI sale del modo PSM0 es recomendable desactivar el DSP escribiendo 0x0000.

### 1.5.9. Interrupciones

Cuenta con 2 pines de interrupción  $\overline{IRQ0}$  E  $\overline{IRQ1}$  cada uno manejado por registros de 32-bits de enmascaramiento activando los registros MASK0 y MASK1 estos registros activan la interrupción colocando en 1 el bit correspondiente de la funcion de la cual se desae conocer el estado.

Si el bit de enmascaramiento se encuentra con un valor de "1.<sup>en</sup> el registro correspondiente esto, entonces producirá una salida en el pin  $\overline{IRQx}$  de activo en bajo.

Para determinar la fuente de la interrupción el microcontrolador debe realizar una lectura en los registros e identificar el bit en alto correspondiente. Para borrar la bandera del registro STATUSx se debe escribir de nuevo en el registro con la bandera con la bandera establecida en "1". Después de una interrupción el pin cambia a un valor lógico bajo, el registro de STATUSx es leído y la fuente de la interrupción es

identificada. Entonces el registro es escrito de vuelta sin ningun cambio para limpiar el estado de la bandera de vuelta a 0. El estado del pin  $\overline{IRQx}$  se mantiene en bajo hasta que el estado de la bandera es cancelado.

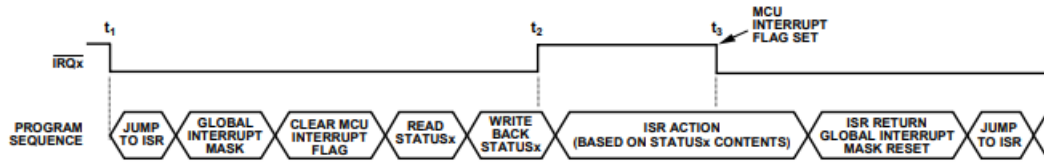


Figura 1.15: Interrupciones.

Inicialmente todas las interrupciones se encuentran desactivadas por lo que la interrupcion RSTDONE es una excepción.

### 1.5.10. Interfaces de Comunicación

El ADE7880 posee 3 puertos de comunicación serial SPI, I<sup>2</sup>C Y HSDC, sin embargo por la configuración de los pines solo se pueden usar 2 tipos de configuraciones, una utilizando el puerto SPI únicamente y la segunda es usando los puertos I<sup>2</sup>C Y HSDC.

Se incluyen un conjunto de 3 registros que permiten verificar la comunicación con el CI. Los registros LAST\_OP (Address 0xEA01), LAST\_ADD (Address 0xE9FE) and LAST\_RWDATA se encargan de almacenar los datos, la dirección de registro y la naturaleza del acceso al registro (Lectura/Escritura) de la última comunicación exitosa respectivamente. El registro LAST\_RWDATA tiene 3 diferentes direcciones dependiendo de la longitud del registro de la última comunicación.

Tipo de comunicación	Dirección de registro
Lectura/ Escritura de 8-Bit	0xE7FD
Lectura/ Escritura de 16-Bit	0xE9FF
Lectura/ Escritura de 32-Bit	0xE5FF

Tabla 1.3: Direcciones de los registros de última escritura (LAST\_RWDATA).

Después de cada comunicación exitosa los registros se actualizan con la información de la operación, 0xCA si fue escritura y 0x35 si es lectura y el dato que se leyó o escribió. Las operaciones no completadas no se reflejan en estos registros así como cuando se leen estos registros.



### **SPI (Serial Peripheral Interface)**

El SPI de este CI siempre es esclavo en la comunicación y consiste en 4 pines de comunicación SCLK, MOSI, MISO Y SSA.

El reloj para la transferencia de datos debe aplicarse al pin SCLK y todas las transferencias de datos se sincronizan con este reloj. El intercambio de información entrante en el CI se realiza en el pin MOSI en los flancos descendentes del reloj serial y el CI lo muestrea en los flancos ascendentes del reloj. Por el contrario la información saliente del CI se transfiere por el pin MISO en los flancos descendentes del reloj serial y el dispositivo maestro muestrea la información en los flancos ascendentes. El bit más significativo de la palabra es intercambiado a la entrada. La máxima frecuencia del reloj serial soportada por esta interfaz es de 2.5[MHz], el pin MISO se mantiene en alta impedancia mientras no hay intercambio de información.

EL pin  $\overline{SS}$  es utilizado para seleccionar entra más de un dispositivo si es que los hay. Para poder utilizar el dispositivo y realizar la comunicación entre ete y el dispositivo maestro este pin debe llevarse a un voltaje lógico bajo y permanecer así mientras dure la comunicación. Al cambiar el nivel lógico de este pin aborta la transferencia y para iniciar una nueva este se debe volver a poner a un voltaje lógico bajo.

### **I<sup>2</sup>C (Inter Integrated Circuits)**

Este CI posee una interfaz I<sup>2</sup>C para la comunicación, implementada completamente como un hardware esclavo. El pin de intercambio de información SDA se encuentra localizado en el pin 38 y se encuentra compartido con el pin MOSI del puerto SPI, de igual manera el reloj serial SCL se encuentra compartido con el reloj serial del puerto SPI en el pin 36. La máxima frecuencia soportada por el reloj serial es de 400[kHz].

La secuencia de transferencia del sistema I<sup>2</sup>C consiste en el dispositivo maestro generando un condición de inicio para la transferencia cuando el bus se encuantra desocupado. El dispositivo maestro entonces transmite la direccion del dispositivo esclavo (0x70) y la dirección del registro de la transferencia de datos en la transferencia

de dirección inicial si el dispositivo esclavo reconoce entonces empieza la transferencia de datos. Esto continua hasta que el maestro emite una condición de parada y el bus queda de nuevo desocupado.

### **HSDC (High Speed Data Capture)**

Debido a que para este proyecto es necesario el recopilar una gran cantidad de muestras a alta velocidad la documentación del Circuito Integrado ADE7880 recomienda, para la lectura de estos registros en específico, utilizar una interfaz propia de Analog Devices para este circuito llamada High Speed Data Capture (Captura de Datos a Alta Velocidad) por lo que es necesario configurar el microprocesador de manera que se pueda usar la comunicación I<sup>2</sup>C como interfaz serial principal y la comunicación HSDC como secundaria usando el canal SPI del microcontrolador como esclavo, que recibirá toda la información de los registros de voltaje y corriente enviados por esta interfaz.

## **1.6. Entorno de Desarrollo MPLABX**

Microchip provee un entorno de desarrollo y un compilador

### **1.6.1. Herramienta de configuración MPLAB Harmony**

## **1.7. Firmware**

### **1.7.1. Inicialización**

La programación del proyecto se realiza por parte del entorno de desarrollo de Microchip MPLAB X, mientras que la configuración de los puertos se realiza con la herramienta proporcionada por el mismo MPLAB Harmony debido a que la configuración manual sería demasiado complicada debido a la cantidad de puertos, pines y servicios disponibles por parte del microcontrolador. ya que esta es una actualización

del módulo de medición de energía por lo que la inicialización esta parcialmente completa, sin embargo al realizar el cambio de canal de comunicación principal de SPI (1.16) a I2C(1.17) y HSDC(Este puerto utiliza la interfaz SPI del microcontrolador por lo que se reconectan los pines pero se deja la configuración ) se cambió las configuraciones de estos canales. Todas las demás siguieron igual a como se tenían en la primer versión del módulo.

### 1.7.2. Configuración del puerto I<sup>2</sup>C

Este puerto se activa al encender el dispositivo o al realizar un reinicio de hardware, entonces el puerto I<sup>2</sup>C queda seleccionado como interfaz de comunicación. Para evitar que al hacer cambios en el nivel de voltaje del pin  $\overline{SS}$ /HSA se cambie a la interfaz de comunicación SPI, se debe bloquear el puerto I<sup>2</sup>C para ello se debe poner a 1 el Bit 1 (I2C\_LOCK) del registro CONFIG2, esto previene que al realizar cambios de voltaje en el pin  $\overline{SS}$ /HSA y el cambio a SPI no es posible hasta que se realice un reinicio de hardware o se reinicie todo el sistema.

Bit	Mnemonic	Valor predeterminado
0	EXTREFEN	0
1	I2C_LOCK	1
7:2	Reservados	0

Tabla 1.4: Valores del registro CONFIG2 que se deben transmitir para bloquear el puerto I<sup>2</sup>C.

Este registro se accesa como si fuera un registro de 8-Bits por lo que los 6 Bits más significativos deben permanecer como 0. En este caso en particular se transmite 0x02 a la dirección de registro 0xEC01.

### 1.7.3. Configuración del puerto HSDC

Para configurar el puerto HSDC se debe escribir primero al registro HSDC\_CFG [0xE706] a traves del puerto I<sup>2</sup>C la configuración con la que se van a estar enviando los datos a traves del puerto HSDC. Ya que se configura el puerto se habilita colocando el bit 6 (HSDCEN) en el registro CONFIG [0xE618] a 1, con esto se activa la

**SPI Driver Instance 1**

- SPI Module ID: SPI\_ID\_6
- Driver Mode
  - TX Interrupt Priority: INT\_PRIORITY\_LEVEL3
  - TX Interrupt Sub-priority: INT\_SUBPRIORITY\_LEVEL0
  - RX Interrupt Priority: INT\_PRIORITY\_LEVEL3
  - RX Interrupt Sub-priority: INT\_SUBPRIORITY\_LEVEL0
  - Error Interrupt Priority: INT\_PRIORITY\_LEVEL3
  - Error Interrupt Sub-priority: INT\_SUBPRIORITY\_LEVEL0
- Master/Slave Mode
- Data Width
- Buffer Mode
  - ☐ Allow Idle Run
- Protocol Type: DRV\_SPI\_PROTOCOL\_TYPE\_STANDARD
- Baud Clock Source: SPI\_BAUD\_RATE\_PBCLK\_CLOCK
- Clock/Baud Rate - Hz: 150000
- Clock Mode: DRV\_SPI\_CLOCK\_MODE\_IDLE\_HIGH\_EDGE\_FALL
- Input Phase: SPI\_INPUT\_SAMPLING\_PHASE\_IN\_MIDDLE
- Dummy Byte Value: 0xFF
- Max Jobs In Queue: 10
- Minimum Number Of Job Queue Reserved For Instance: 1

Figura 1.16: Configuración SPI del PIC32MZ.

**I2C**

- ☒ Use I2C Driver?
  - Driver Implementation: DYNAMIC
  - ☒ Interrupt Mode
  - Number of I2C Driver Clients: 1
  - Number of I2C Driver Instances: 1
  - ☐ Include Force Write I2C Function (Master Mode Only - Ignore NACK from Slave)
- ☒ I2C Driver Instance 0
  - ☐ Use Bit Bang I2C Implementation?
  - I2C Module ID: I2C\_ID\_2
  - Operation Mode: DRV\_I2C\_MODE\_MASTER
  - Master Interrupt Priority: INT\_PRIORITY\_LEVEL5
  - Master Interrupt Sub-priority: INT\_SUBPRIORITY\_LEVEL0
  - Error Interrupt Priority: INT\_PRIORITY\_LEVEL5
  - Error Interrupt Sub-priority: INT\_SUBPRIORITY\_LEVEL0
  - Baud Rate Generator Clock: 100000000
  - I2C CLOCK FREQUENCY (Hz): 400000
  - ☐ Slew Rate Control
  - Power State: SYS\_MODULE\_POWER\_RUN\_FULL

Figura 1.17: Configuración I2C del PIC32MZ.

comunicación.

La configuracion que se ha decidido usar es tener el reloj a 8[MHz] con la transmision de registros en paquetes de 32-bits, no se agrega una brecha de 7 ciclos de reloj entre transmisiones y unicamente se transmitira el contenido de los registros de voltaje y corriente: IAWV, VAWV, IBWV, VBWV, ICWV, VCWV, e INWV, lo que permite que se realice la comunicación de manera más rápida. El pin de seleccion de esclavo ( $\overline{SS}$  ó Chip Select) se mantiene como activo en bajo.

# Capítulo 2

## Resultados

Como se explicó al principio el desarrollo de este proyecto se pensó como una mejora a los medidores de energía anteriormente fabricados por el Instituto de Ingeniería por lo que este, solo se centró únicamente en la lectura de muestras a alta velocidad por parte del microcontrolador.

### 2.1. Lectura de Datos Usando el Canal SPI

En un principio el sistema medidor de energía estaba pensado para realizar la comunicación entre el microprocesador y el medidor ADE7880 con interfaz SPI sin embargo al momento de realizar las primeras pruebas, los registros correspondientes a los canales de medición de voltaje y corriente, pasado un tiempo los valores que se recuperan de los registros pierden coherencia lo que hace que los resultados sean inútiles para su análisis.

La lectura de los registros se configuró para que el microprocesador lo realizara a la mayor velocidad posible utilizando un reloj de 150000[Hz] por el canal 6 SPI 1.16.

En la siguiente gráfica 2.1 se puede ver como la velocidad de comunicación es bastante rápida al leer los valores de registro a pesar de tener que leer el registro DREADY para asegurarse que los registros de voltaje y corriente esten listos se obtenían demasiado rápido, pero como se había mencionado estos valores no demostraban valores que puedan ser útiles.

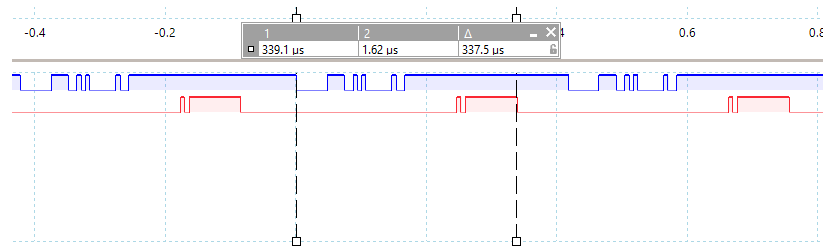


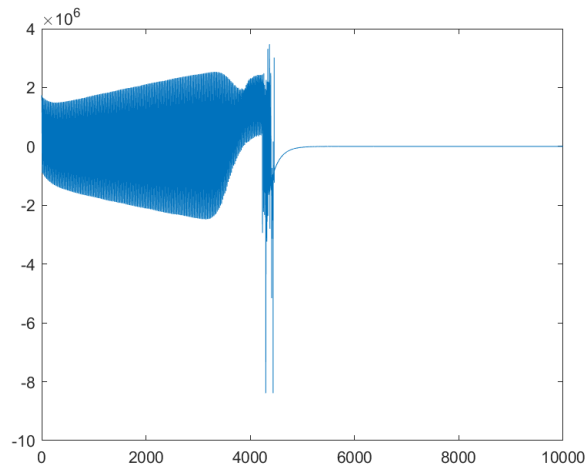
Figura 2.1: Comunicación SPI con los registros de voltaje y corriente.

Como se pueden ver en las graficas de voltaje (Graficas 2.2) y corriente (Graficas 2.3) al usar el canal SPI para muestrear los registros de voltaje y corriente de los 3 canales estos entregaban valores que no reflejaban la energía que se suministraba y en algunos casos ni siquiera se actualizaban los valores que el dispositivo tiene inicialmente, ya que las pruebas que se hacían se realizaban con un simulador de voltaje y corriente controlado con lo cual era posible conocer el valor que se deseaba obtener.

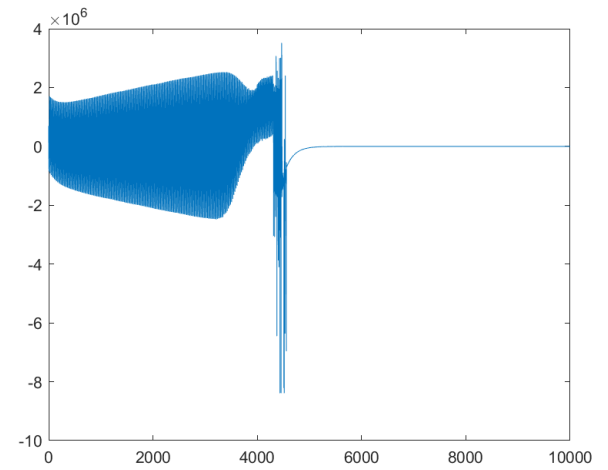
Después de realizar un par de pruebas más y obtener valores similares en todas las pruebas y siguiendo la recomendación que el manual del dispositivo para la lectura de esos registros se decidió cambiar el canal de comunicación de SPI por el HSDC con el cual se planea tener una lectura más rápida de los registros así como poder mantener la comunicación únicamente con los registros de voltaje y corriente de manera continua sin interacciones con los otros módulos del sistema de medición de energía.

## 2.2. Lectura de Datos usando el canal HSDC

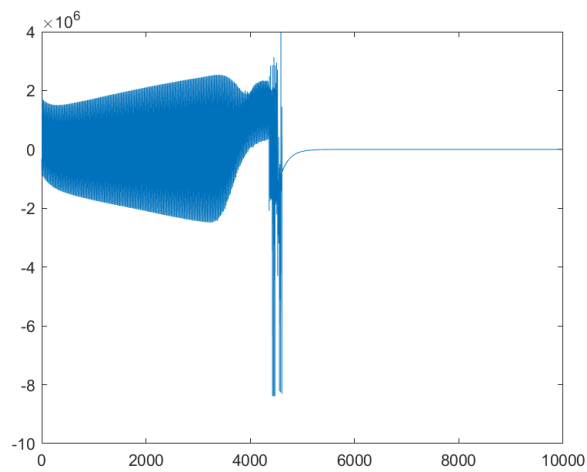
Las primeras pruebas que se realizaron usando este canal de comunicación inicia de manera correcta al configurar los valores necesarios en el registro correspondiente del ADE7880, se comprobó usando el osciloscopio que los paquetes de datos se envíen y reciban de manera correcta por el canal I2C para despues comprobar que el puerto HSDC realmente envíe información. Al iniciar el DSP se comprobó que el puerto HSDC enviaba la información proveniente de los registros de corriente y voltaje del medidor de energía.



(a) Canal de Voltaje A



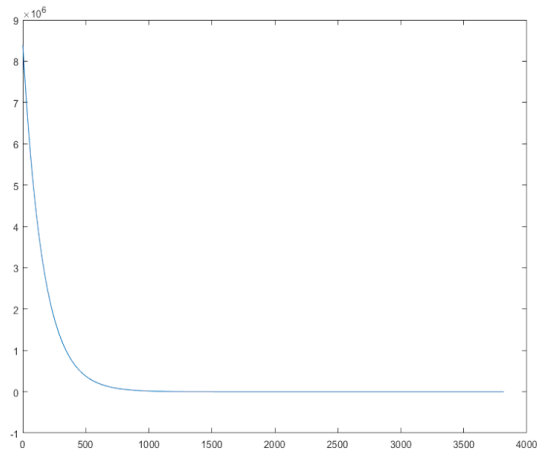
(b) Canal de Voltaje B



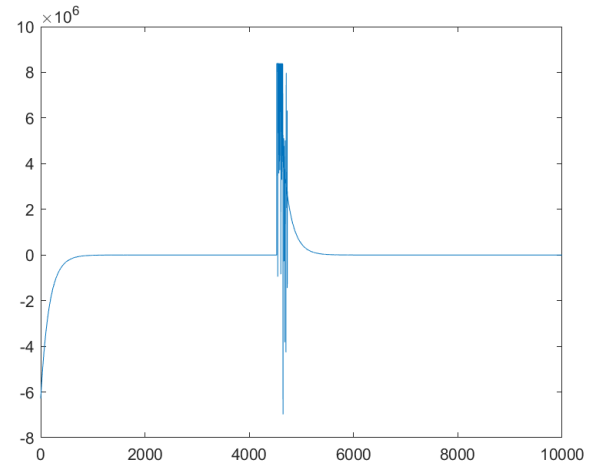
(c) Canal de Voltaje C

Figura 2.2: Gráficas de la primera prueba de los Canales de Voltaje.

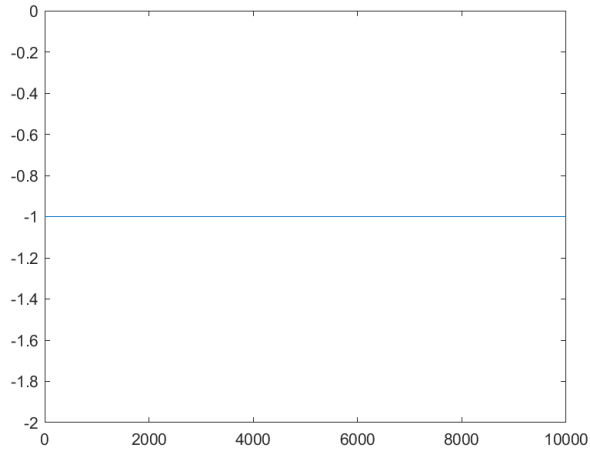




(a) Canal de Corriente A



(b) Canal de Corriente B



(c) Canal de Corriente C

Figura 2.3: Gráficas de la primer prueba de los Canales de Corriente.

## Capítulo 3

### Conclusiones

Hasta el momento el canal que se tenía seleccionado como principal en un inicio (SPI) no cumplió con las funciones requeridas para el proyecto por lo que se tuvo que cambiar por el  $I^2C$  y HSDC, en un inicio y debido a la configuración se tuvo que reestructurar las conexiones de comunicación, sin embargo esto también ocasionó contratiempos en la comunicación con el microcontrolador ya que la interfaz de programación añadía unas líneas para verificar que el canal de comunicación se encuentre disponible para la transmisión de datos.

# Capítulo 4

## Glosario

### B

- Bandera: Se le conoce como bandera a un bit en un registro que denota un valor binario que tiene un significado para el programa.

### F

- Frecuencia de corte ( $\omega C$ ) ó también llamada frecuencia de esquina o frecuencia crítica: Es la frecuencia donde la respuesta en amplitud está 3[dB] por abajo del valor de la banda de paso.

### O

- Offset: Tensión de voltaje de desequilibrio que se presenta debido a el uso de amplificadores operacionales cuando la señal de entrada es muy baja.[Areny, 2005]

### P

- Pipeline ó Registro de Arquitectura Paralela: Registro de datos en los que se permite la ejecución de las instrucciones de manera simultanea con la siguiente instrucción. Esta configuración requiere de un reloj de dos fases donde una se aplica al registro de direccionamiento y una al registro de datos.[Mano and Sarmiento, 1994]

# V

- Valor Lógico de Voltaje: En electrónica digital se utilizan sistemas y circuitos en los que solo existen 2 estados posibles representados mediante 2 niveles de tensión diferentes Alto (1) y Bajo (0). En circuitos digitales TTL (Transistor-Transistor) pueden variar entre 2.8 y 5.0[V] mientras que los valores bajos varían entre 0 y 0.8[V]. Los valores entre 0.8 y 2.8[V] no están definidos y nunca deben utilizarse.

# Bibliografía

- [Areny, 2005] Areny, R. (2005). *Sensores y Acondicionadores de Señal 4a.* ACCESO RÁPIDO. Marcombo.
- [Caprile, 2012] Caprile, S. (2012). *Desarrollo con microcontroladores ARM Cortex-M3.* Puntolibro.
- [Connectivity, ] Connectivity, P. E. *Technical Reference Manual [Electronic resource]*.
- [Cuéllar, 2008] Cuéllar, A. (2008). *Sistemas de Procesamiento Digital.* Delta Publicaciones.
- [Mano and Sarmiento, 1994] Mano, M. and Sarmiento, M. (1994). *Arquitectura de computadoras.* Prentice Hall.
- [Valdes and Areny, 2007] Valdes, F. and Areny, R. (2007). *Microcontroladores Fundamentos y Aplicaciones con PIC.* Alfaomega. Marcombo. Marcombo.