



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons  
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite  
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE CIENCIAS E INGENIERÍA**



**DISEÑO DEL SISTEMA DE ILUMINACIÓN AUTOMATIZADO PARA UNA  
OFICINA EN UN EDIFICIO INTELIGENTE, BASADO EN TECNOLOGÍA  
INALÁMBRICA ZIGBEE**

**Tesis para optar el Título de Ingeniero Electrónico**

**Presentado por:**

**MIGUEL EDUARDO HIDALGO ALCALÁ**

**Lima - Perú**

**2007**

<u>ÍNDICE</u>	Pág.
INTRODUCCIÓN: .....	5
<b>CAPÍTULO 1: EL MANEJO DE LA ILUMINACIÓN EN OFICINAS DE LA CIUDAD DE LIMA.</b> .....	7
<b>1.1. Proceso actual.</b> .....	7
<b>1.1.1. Infraestructura.</b> .....	7
<b>1.1.2. Políticas.</b> .....	8
<b>1.2. Deficiencias del manejo de la iluminación en edificaciones.</b> .....	8
<b>1.3. Declaración del marco problemático.</b> .....	9
<b>CAPÍTULO 2: MÉTODOS Y TECNOLOGÍAS DISPONIBLES PARA CONTROLAR LA INTENSIDAD DE LA ILUMINACIÓN.</b> .....	11
<b>2.1. Estado del arte.</b> .....	11
<b>2.1.1. Tipos de sistemas.</b> .....	12
<b>2.1.2. Tecnologías de comunicación disponibles.</b> .....	14
<b>2.2. Conceptualizaciones generales.</b> .....	17
<b>2.2.1. Sistemas de iluminación</b> .....	17
<b>2.2.1.1 Definición</b> .....	17
<b>2.2.1.2 Modos de control</b> .....	17
<b>2.2.1.3 Productos utilizados actualmente</b> .....	18
<b>2.2.2 Edificio Inteligente</b> .....	18
<b>2.2.2.1 Definición</b> .....	19
<b>2.2.2.2 Subsistemas</b> .....	19
<b>2.2.3 Tecnología ZigBee</b> .....	20
<b>2.2.3.1 Definición</b> .....	20
<b>2.2.3.2 Características</b> .....	20

2.2.3.3 Frecuencia, ancho de banda y número de canales .....	20
2.2.3.4 Características de los paquetes transmitidos .....	21
<b>CAPÍTULO 3: REQUERIMIENTOS PARA EL DISEÑO DEL SISTEMA DE CONTROL BASADO EN TECNOLOGÍA INALÁMBRICA.....</b>	<b>22</b>
3.1. Hipótesis de la investigación.....	22
3.1.1. Hipótesis principal .....	22
3.1.2. Hipótesis secundarias .....	22
3.2. Objetivos de la investigación. ....	23
3.2.1. Objetivo General.....	23
3.2.2. Objetivos Específicos.....	23
3.3. Universo y muestra. ....	24
3.4. Situación actual y tecnologías disponibles para la optimización.....	24
3.4.1. Distribución actual de las luminarias.....	24
3.4.2. Detalle de las luminarias actuales .....	28
3.4.3. Balastos y balastos electrónicos.....	29
3.4.4. Sensor de luz, fotosensor o fotocélula .....	31
3.5. Interpretando los datos: Determinación de la solución .....	32
3.5.1. Luminarias .....	32
3.5.2. Balastos electrónicos .....	32
3.5.3. Fotocélula.....	37
3.5.4. Comunicación inalámbrica.....	40
<b>CAPÍTULO 4: DISEÑO DEL SISTEMA DE REGULACIÓN PARA UNA OFICINA EN LIMA.....</b>	<b>44</b>
4.1. Consideraciones preliminares. ....	44

<b>4.2. Análisis y diseño del sistema.....</b>	<b>44</b>
4.2.1. El microcontrolador.....	45
4.2.2. Transmisor-receptor de radiofrecuencia de 2.4 GHz IEEE 802.15.4.....	46
4.2.3. La antena .....	48
<b>4.3. Desarrollo de la solución .....</b>	<b>51</b>
4.3.1. Módulo de control maestro (transmisor).....	53
<b>4.4. Módulo de regulación de la iluminación (receptor).....</b>	<b>55</b>
4.4.1. Multiplexor analógico.....	56
4.4.2. Acondicionamiento de la señal del conversor digital/análogo .....	57
4.4.3. Acondicionamiento de la señal de control de encendido y apagado de las luminarias.....	60
4.4.4. Fuentes de alimentación.....	61
<b>4.5. Simulación de la instalación y funcionamiento.....</b>	<b>63</b>
4.5.1. Lógica del programa .....	64
4.5.1.1. Programa de transmisión .....	66
4.5.1.2. Programa de recepción .....	70
4.5.2. Pruebas de los programas .....	73
4.6. Presupuesto .....	81
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>85</b>
<b>FUENTES .....</b>	<b>88</b>
Bibliografía.....	88
Documentos .....	89

## INTRODUCCIÓN:

En la actualidad, en la ciudad de Lima hay un sin número de oficinas de diversa índole para atender los distintos niveles y rubros de la industria.

Sin embargo, sólo las nuevas construcciones tienen cierto grado de automatización y normalmente está orientado al control de la temperatura (aire acondicionado). Prácticamente ninguna oficina cuenta con automatización en la iluminación y, las que lo hacen, lo limitan al encendido y apagado automático de las luces de los baños, cuando el gasto de energía eléctrica más grande lo están realizando la iluminación de la oficina en sí y los monitores de las computadoras.

El manejo de los monitores es sencillo debido a que se pueden configurar fácilmente para que se apaguen luego de un tiempo específico de inactividad, pero en cuanto a la iluminación, lo más cercano a ahorro que se ha pretendido lograr a un nivel macro es instalar luces fluorescentes de menor consumo de energía, distribuir las de manera que las luces estén sectorizadas y que cada sector esté comandado por un interruptor diferente. Si bien estas medidas ayudan, no son del todo eficientes, debido a que es común que alguna luz este prendida innecesariamente.

En nuestros días, donde la tecnología la llevamos a donde sea que vayamos, ya sea en un celular, en una memoria flash USB de gran capacidad, en un reproductor de música MP3, etc.; lo más lógico es que la tecnología esté donde sea que lleguemos y aquí es donde empieza a formarse el concepto de los edificios inteligentes.

El presente documento pretende mostrar una manera de ahorrar la energía eléctrica utilizada para la iluminación en una oficina, administrando su uso de

acuerdo a la demanda, no sólo en algo tan básico como encenderla sólo cuando va a ser utilizada y apagarla en caso contrario, sino en algo más elaborado como regular su intensidad de acuerdo a la necesidad y lograr realizarlo por sectores y de manera remota.



## **CAPÍTULO 1: EL MANEJO DE LA ILUMINACIÓN EN OFICINAS DE LA CIUDAD DE LIMA.**

### **1.1. Proceso actual.**

El proceso actual del manejo de la iluminación es una actividad simple y cotidiana que se realiza de manera manual mediante el uso de interruptores. Básicamente lo que sucede es que se observa el ambiente en general y, sobretodo, la zona de trabajo. Luego, se procede a encender luces o apagarlas de acuerdo a la necesidad, procurando lograr un ambiente adecuado y lo suficientemente cómodo como para realizar una tarea específica. En ocasiones, de ser el caso, suelen abrirse o cerrarse persianas y/o cortinas para ayudar a conseguir la intensidad luminosa deseada. Por lo tanto, la activación y desactivación se produce bajo demanda y de manera aleatoria en el tiempo.

En resumen, lo que se tiene es un sistema basado en prueba y error, que se regula luego de largos intervalos de tiempo y que, muchas veces, es regulado de una manera inadecuada.

#### **1.1.1. Infraestructura.**

Existen diferentes tipos y niveles de iluminación para distintos tipos de oficinas. En el caso de esta oficina, en la que se trabaja en cubículos frente a un computador la mayor parte del tiempo, la iluminación es controlada por varios interruptores correspondientes a distintos sectores. Además presenta diferentes ambientes que requieren distinta iluminación.



### 1.1.2. Políticas.

El Código Nacional de Electricidad del Perú estipula cómo debe ser la instalación y el cableado de las luminarias, así como cuántas luminarias y de qué tipo deben usarse para conseguir la iluminación adecuada según el tipo de trabajo a realizarse.

Sin embargo, la oficina en cuestión está en funcionamiento desde mediados del año 2006 y ha sido revisada por organismos reguladores que inspeccionaron las instalaciones y autorizaron su funcionamiento. Además, lo que más interesa para efectos del presente proyecto son los estándares y las normas reguladoras existentes en lo referente a la comunicación entre dispositivos de control. Desde detalles acerca de cómo debe ser el cableado o instalación de los mismos, hasta los formatos de las tramas de información de los diferentes protocolos que existen.

El objetivo de revisar los documentos reguladores es poder conocer mejor los estándares y las normas que deben cumplirse para poder desarrollar un sistema más flexible y adaptable, tratando de reducir al mínimo los problemas de interoperabilidad que puedan presentarse.

### 1.2. Deficiencias del manejo de la iluminación en edificaciones.

En el proceso actual se han observado algunas deficiencias:

- Necesariamente tiene que haber alguien alrededor para poder modificar el estado de la iluminación: Alguien tiene que encender o apagar las luces para controlar la intensidad luminosa, ya sea para incrementarla o para disminuirla.

- Como el manejo de la intensidad de la iluminación es manual y mediante interruptores, se obliga al usuario a que el control tenga que ser desde el punto en el que está ubicado dicho interruptor.
- Por el mismo hecho que la iluminación sólo puede controlarse desde el interruptor, es usual que haya luminarias encendidas innecesariamente porque a los usuarios les cuesta acercarse a apagarlas o simplemente porque no se dan cuenta que la iluminación es escasa hasta que dicho suceso se hace realmente molesto.

### 1.3. Declaración del marco problemático.

Actualmente, el sistema de iluminación en las oficinas es controlado de manera manual y centralizado, lo que produce que, en muchas ocasiones, la intensidad de la iluminación no sea la adecuada para el tipo de trabajo que se va a realizar. Cabe destacar además, que el control actual no se ajusta a la tendencia que tiene la sociedad al ahorro y al cuidado del ahorro.

Los dos posibles problemas frente a esto son: que la iluminación sea escasa, en cuyo caso el trabajo de oficina puede producir incomodidad, jaquecas y cansancio excesivo innecesario; o que la iluminación sea abundante, lo que produce un consumo energético y desgaste de luminarias innecesario. En ambos casos, los problemas producen un desperdicio económico para la empresa. En el primer caso, es una disminución en el rendimiento del empleado y en el

segundo, es un monto mayor el que se pagará en el recibo de la luz y las luminarias se tendrán que cambiar más seguido.



## **CAPÍTULO 2: MÉTODOS Y TECNOLOGÍAS DISPONIBLES PARA CONTROLAR LA INTENSIDAD DE LA ILUMINACIÓN.**

### **2.1. Estado del arte.**

La energía eléctrica es la forma de energía más utilizada. Gracias a la flexibilidad en la generación y a que el transporte por líneas de alta tensión es sumamente ventajoso, se ha convertido para la industria en la forma más grande de consumo de energía.

Los inconvenientes de esta forma de energía son la imposibilidad de almacenamiento en grandes cantidades y que las líneas de transmisión son muy costosas y por estos motivos, el ahorro de energía eléctrica es un tema que siempre es de interés.

Actualmente, existen diversas maneras de regular el uso de la energía eléctrica, sin embargo, en el Perú, específicamente en Lima, aún no se ha logrado hacer de una manera realmente eficiente. Además, según un estudio realizado en Venezuela y plasmado en el “Manual de diseño para edificaciones energéticamente eficientes”, se sabe que a nivel de toda Latino América cerca del 60% de la energía eléctrica generada es destinada al sector comercial y residencial y cerca del 40% de esta es destinada a iluminación. En este sentido el control de la iluminación es un tema importante para conseguir un ahorro significativo de energía.

De lo que estamos hablando ahora es de domótica (edificios y viviendas inteligentes), aunque realmente sólo estamos entrando en la base de esta, porque es la integración de sistemas automatizados la que realmente se define como tal. Sin embargo, el principio es el mismo debido a que en la actualidad contamos con estándares que hacen posible dicha integración entre diversos sistemas automatizados, así que nos referiremos a domótica porque justamente el desarrollo de este diseño estará buscando la posterior integración a un sistema de mayor envergadura.

### 2.1.1. Tipos de sistemas.

Existen claramente dos tipos de sistemas domóticos: Los sistemas basados en corrientes portadoras, utilizando la red de baja tensión ya instalada y los sistemas que utilizan su propia red para la interconexión de dispositivos.

- Sistemas mediante corrientes portadoras: Tienen la gran ventaja de no necesitar un cableado adicional para su instalación y que su aplicación en edificios ya construidos es inmediata. La desventaja es la complejidad de sus dispositivos, debido a que son necesarios moduladores y demoduladores en todos los elementos del sistema ya que la información va superpuesta a la red eléctrica. Además, aunque se adicionen filtros para recuperar claramente los datos enviados, la misma red eléctrica utilizada como soporte de transmisión puede influir

en el correcto funcionamiento porque introduce numerosas interferencias y, debido a la necesidad física de conexión para el transporte de los datos, tampoco es posible hacer que la información se transmita a través de transformadores.

Existen principalmente tres sistemas comerciales: X-10 de Home Systems, que fue la primera tecnología en la industria para comunicaciones entre dispositivos domóticos y aún está ampliamente disponible (fue desarrollada en 1975 por Pico Electronics - Escocia); CAD de Legrand y Global Home System de Landis & Gyr.

- Sistemas que utilizan su propia red de datos: Pueden ser alámbricos o inalámbricos, pero en cualquiera de los dos casos pueden ser sistemas centralizados, donde existe un dispositivo encargado de procesar la información recibida de los sensores y transmisores para enviar órdenes a los actuadores correspondientes, o sistemas distribuidos, donde cada elemento tiene la capacidad de tratar la información que recibe y actuar de forma autónoma. En la **Tabla 1** se muestran las principales ventajas y desventajas de cada uno de estos sistemas.

	<b>Ventajas</b>	<b>Desventajas</b>
<b>Sistema centralizado</b>	<i>Bajo costo</i> - No necesita módulos especiales de direccionamiento.	<i>Flexibilidad limitada</i> - Las posibles reconfiguraciones serán muy costosas.
	<i>Instalación sencilla</i> - Posibilita la utilización de gran cantidad de elementos convencionales.	<i>Dependencia de la unidad de control</i> - Si esta falla, todo el sistema deja de funcionar. <i>Cableado</i> - Se necesitan más metros de cable que en el caso del sistema distribuido.
<b>Sistema distribuido</b>	<i>Sistemas más potentes</i> - Permiten implementar gran cantidad de aplicaciones.	<i>Programación compleja</i> - Utiliza software especializado.
	<i>Cableado</i> - Se necesitan menos metros de cable.	<i>Es costoso</i> - Debido a la necesidad de incluir protocolos CSMA/CD y poseer capacidad de procesamiento, lo que implica necesidad de compatibilidad.
	<i>Mayor flexibilidad</i> - Facilidad de reconfiguración.	

**Tabla 1: Ventajas y desventajas de los sistemas con red de datos propia**

### 2.1.2. Tecnologías de comunicación disponibles.

En lo referente a la comunicación, el método utilizado por el sistema X-10 (que comunica los dispositivos por la red eléctrica), ya no es lo suficientemente robusto para las demandas actuales de los consumidores y las compañías que se disputan el mercado, tanto en los medios alámbricos como inalámbricos, son la Alianza ZigBee (que utiliza el estándar Zigbee), Smarthome (que utiliza Insteon) y Zensys (que utiliza Z-Wave):

**Insteon:** Utiliza una tecnología que se basa tanto en radiofrecuencia (RF) (entre 902-924MHz), como en el sistema mediante corrientes portadoras (ya sea a 110 o 220 V).

El vicepresidente de Smarthome, Ken Fairbanks [10], dice que cada mensaje o transmisión es propagada a través de RF y la línea de alimentación. Explica que las viviendas son lugares en los cuales es difícil hacer que la RF trabaje de manera confiable y la naturaleza dual del Insteon crea una red robusta. Su RF tiene alcances de 45 mts., es compatible con dispositivos X-10 y permite la integración con computadoras.

**Z-Wave:** Es un protocolo completamente inalámbrico que utiliza banda angosta para transmitir sus comandos de control y data menor (no tiene suficiente ancho de banda para audio o video). Trabaja a una frecuencia de 908MHz y permite la formación de redes, en las que todos los dispositivos apoyan al sistema actuando cada uno como su propio repetidor.

Según Michael Einstein, vocero de marketing en Z-Wave, una de las principales ventajas sobre los productos X-10, es la seguridad. Afirma que mientras que con X-10 sólo se pueden tener 256 identificadores (IDs) únicos y una persona podría estar prendiendo o apagando la luz de un vecino o abriendo y cerrando la puerta del garaje de otro por errores de IDs duplicados, Z-Wave tiene cerca de cuatro billones de IDs únicos. Este protocolo también permite integración con



computadoras mediante comunicación serial (RS232 y USB) para control remoto.

**ZigBee:** Es un estándar de comunicación inalámbrica que está siendo muy utilizada para las aplicaciones de domótica, ya que, entre otras cosas, puede formar su propia red con gran número de dispositivos y prescindir del cableado, además de posibilitar el control remoto.

También permite formar redes de tipo estrella, rejilla o árbol; mediante la utilización de tres tipos de dispositivos [8]:

- Coordinador de red: Encargado de mantener información general de la red y tiene los mayores requisitos de memoria y capacidad de procesado.
- Función completa: Soporta todas las funciones definidas por el estándar IEEE 802.15.4 (Redes WPAN) y ZigBee. Además, puede funcionar como coordinador de red.
- Función reducida: Implementan sólo una parte de las funciones de acuerdo con su tarea. Esta funcionalidad permite que sea de menor costo y complejidad y, por lo tanto, de mayor consumo.

El protocolo ZigBee fue creado con la intención de satisfacer las necesidades del mercado de conexiones inalámbricas de bajo costo, bajo consumo de energía, bajas velocidades de transmisión y seguridad moderada. Los dispositivos Zigbee cuentan con una dirección MAC única, lo que evita que haya problemas de IDs duplicados y hace que las redes que se quieran formar puedan tener

un número de dispositivos muy elevado. Además, tiene un alcance entre 10 y 75 metros dependiendo del entorno y opera en tres bandas libres para cumplir con los requerimientos mundiales: En la banda de 2.4GHz. a 250kbps (global), en la de 915MHz. a 40kbps (EE.UU.) y en la de 868MHz (Europa). a 20kbps. Es importante destacar que el método de modulación es distinto si se opera en la banda de 2.4GHz (O-QPSK) o en las bandas 915/868MHz (BPSK) [2].

## 2.2. Conceptualizaciones generales.

### 2.2.1. Sistemas de iluminación

Las estrategias para el control de iluminación pueden jugar un papel significativo en la reducción del costo de energía [11].

#### 2.2.1.1 Definición

Sistema compuesto por fuentes de luz instaladas y distribuidas de tal manera que se optimice el confort visual para la adecuada realización de una determinada actividad [15].

#### 2.2.1.2 Modos de control

Existen dos modos de control de un sistema de iluminación. Los de alto voltaje y los de bajo voltaje:

- Alto voltaje: Normalmente utilizan temporizadores de alto voltaje, llaves de alta tensión y contactores. Son costosos y requieren un amplio margen de seguridad [11].

- Bajo voltaje: Utilizan interruptores comunes y en ocasiones se automatiza el encendido y apagado mediante el uso de sensores. Son más seguros y más costo-efectivos [11].

### **2.2.1.3 Productos utilizados actualmente**

Principalmente son cuatro:

- Lámparas incandescentes: Termo-radiadores compuestos por un filamento metálico en forma de espiral, alojado en el interior de una ampolla de vidrio [12].
- Lámparas de vapor de mercurio: Lámparas eléctricas de descarga en las que la emisión de radiaciones luminosas se debe a la excitación de los átomos de mercurio contenidos en el tubo de descarga [12].
- Lámparas de vapor de sodio: Fuentes luminosas en las que la producción de luz se efectúa como consecuencia a la excitación de los átomos de sodio contenidos en el tubo de descarga [12].
- Lámparas Fluorescentes: Aquellas lámparas de descarga en atmósfera de mercurio a baja presión en las que la generación de luz se debe principalmente, al fenómeno de fotoluminiscencia [12].

### **2.2.2 Edificio Inteligente**

Los edificios cada vez son más complejos y sus regulaciones de gobierno más rectas, especialmente con respecto a la seguridad de la vida [12].

### **2.2.2.1 Definición**

Se define como edificio inteligente a aquel que provee un ambiente productivo y costo-efectivo a través de la optimización de su estructura, sistema, servicio y administración así como a la interrelación entre ellos [11].

También se define como el edificio que incluye todo el hardware, software, cableado e incidentes que son parte del sistema de supervisión, control y funciones de monitoreo necesarias para su automatización integral [12].

### **2.2.2.2 Subsistemas**

Existen varios tipos de subsistemas que pueden o no, estar incluidos en un edificio inteligente y se mencionan a continuación:

- *Control de energía* – Que incluye administración y regulaciones en el uso de la energía eléctrica en todos los artefactos que la utilizan [9][11][12].
- *Calefacción, ventilación y aire acondicionado (HVAC)* – Para mantener un ambiente y una temperatura apropiadas de acuerdo a las necesidades del local [9][11][12].
- *Incendios* – Manejo de alarmas y rociadores para advertir, contener y extinguir un incendio [9][11][12].
- *Seguridad* – Con el uso de cámaras, alarmas y control de accesos ante intrusiones no deseadas [9][11][12].
- *Transporte y tráfico* – Brindándole inteligencia a los ascensores para una mejor administración de su uso [9][12].

- *Contaminación* – Calidad del aire en el exterior e interior y manejo adecuado de desechos tóxicos [9][12].
- *Comunicación* – Calidad y seguridad en la misma, ya sea cableada o inalámbrica [12].
- *Plomería* – Haciendo posible la automatización del suministro de agua y activación del drenaje [11][12].

### **2.2.3 Tecnología ZigBee**

#### **2.2.3.1 Definición**

Estándar de comunicación inalámbrica de corto alcance, destinada a la electrónica de consumo y al control industrial; permitiendo el control remoto, sensado y uso de aplicaciones [2][13].

#### **2.2.3.2 Características**

Establece comunicaciones inalámbricas entre dispositivos y está centrado en aplicaciones de bajo coste y consumo [2], es ideal para locales aislados y ha sido implementado con el estándar 802.15.4 que define las capas física y MAC (Medium Access Control) [13]. Además puede conectar hasta 255 nodos por red y es un fácil protocolo de definición porque es implementable en microcontroladores de bajo costo [2].

#### **2.2.3.3 Frecuencia, ancho de banda y número de canales**

Estos tres parámetros varían dependiendo de la ubicación geográfica y se resumen en la **Tabla 2**.

	Frecuencia	BW	Nº de canales
Europa	868 MHz	20 Kbps	1
EE.UU.	915 MHz	40 Kbps	10
Resto del mundo	2.4 GHz	250 Kbps	16

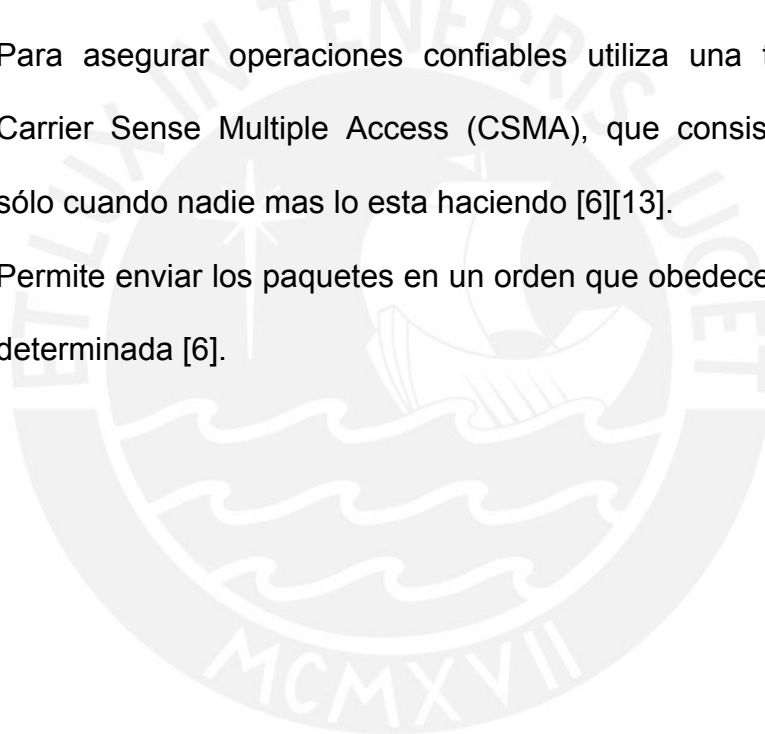
**Tabla 2: Características de la señal Zigbee**

#### **2.2.3.4 Características de los paquetes transmitidos**

Tamaño máximo: 128 bytes (incluyendo) el encabezado [6].

Para asegurar operaciones confiables utiliza una técnica llamada Carrier Sense Multiple Access (CSMA), que consiste en transmitir sólo cuando nadie mas lo esta haciendo [6][13].

Permite enviar los paquetes en un orden que obedece a una prioridad determinada [6].



## **CAPÍTULO 3: REQUERIMIENTOS PARA EL DISEÑO DEL SISTEMA DE CONTROL BASADO EN TECNOLOGÍA INALÁMBRICA.**

### **3.1. Hipótesis de la investigación.**

#### **3.1.1. Hipótesis principal**

Dado que en Lima el control de la iluminación es manual, se regula ocasionalmente (generando en muchas ocasiones abundancia o escasez de luz) y no se automatiza porque la interconexión de dispositivos requiere una serie de esfuerzos que suelen considerarse innecesarios; entonces un sistema de regulación de la iluminación, que se pueda controlar de manera inalámbrica, permitirá optimizar el uso de los recursos para generar un ahorro en el consumo de energía e incrementar el grado de confort, sin requerir de un cableado adicional.

#### **3.1.2. Hipótesis secundarias**

- 1) Utilizar un estándar en la comunicación entre dispositivos permitirá la posterior integración con otros sistemas automatizados para lograr generar un sistema de mayor envergadura.
- 2) Los sistemas automatizados permiten que el proceso de control sea continuo y reprogramable ante algún cambio en cuanto a la intensidad luminosa necesaria en un determinado sector.
- 3) El uso de un sistema automatizado para el control de la iluminación debería evitar el desperdicio de energía eléctrica, optimizando el uso de los recursos disponibles para lograr un ahorro económico.

- 4) El empleo de un sistema de control inalámbrico posibilita la implementación de dichos sistemas en ambientes que originalmente no fueron pensados para tales fines, eliminando el trabajo de cableado necesario.
- 5) El uso de un sistema de control que brinde la cantidad de luz necesaria en todo momento, incrementará el grado de confort de las personas que trabajen en el área.

### **3.2. Objetivos de la investigación.**

#### **3.2.1. Objetivo General.**

Diseñar un sistema regulable e inalámbrico para controlar la intensidad luminosa en un ambiente de trabajo de acuerdo a sus propias necesidades, maximizando el grado de confort y reduciendo el desperdicio de la energía eléctrica.

#### **3.2.2. Objetivos Específicos.**

- 1) Evitar el exceso o la falta de iluminación; por lo tanto, proporcionar sólo la intensidad luminosa necesaria para la actividad a realizar, a menos que el usuario desee graduar la intensidad manualmente.
- 2) Sistematizar y automatizar el proceso de regulación de la intensidad luminosa valiéndose del control retroalimentado.
- 3) Analizar si la intensidad luminosa proporcionada realmente maximiza el grado de confort.
- 4) Realizar el programa que posibilite el control inalámbrico mediante el uso del estándar ZigBee.



5) Diseñar el circuito que permitirá al microcontrolador Zigbee, realizar de manera adecuada la regulación de la iluminación.

### **3.3. Universo y muestra.**

El universo está formado por las oficinas que controlan su intensidad luminosa por sectores, de forma manual y por medio de interruptores.

La muestra que se utilizará para llevar a cabo este trabajo será una replica del piso 13 de la Torre Interbank ubicada en el distrito de La Victoria en la ciudad de Lima, correspondiente a una oficina dedicada a la venta de computadoras.

### **3.4. Situación actual y tecnologías disponibles para la optimización.**

#### **3.4.1. Distribución actual de las luminarias.**

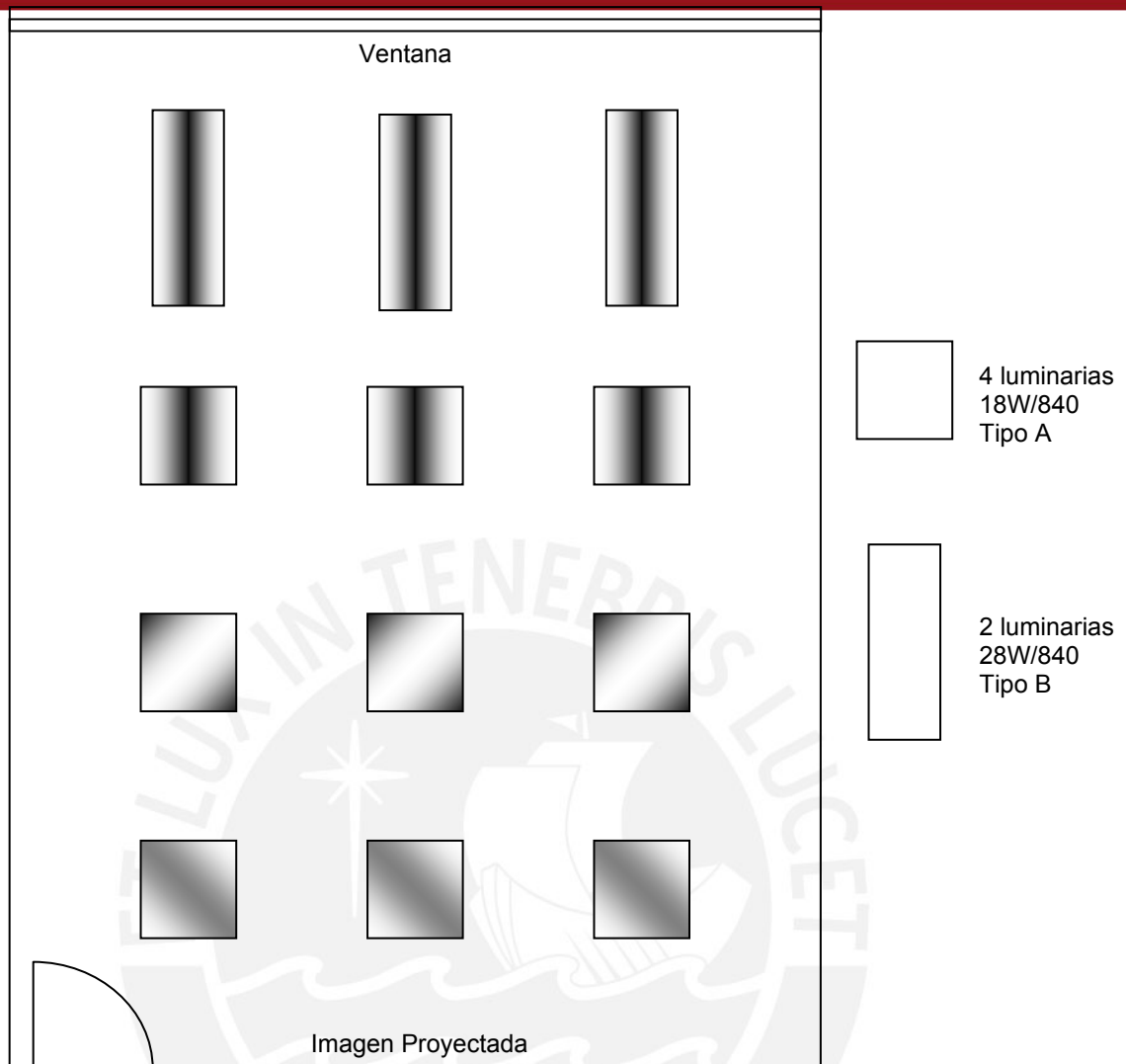
Para poder empezar a plantear de una manera adecuada una buena solución para los requerimientos de iluminación de la oficina, es importante recalcar que el estudio para la determinación del actual manejo de la misma se llevó a cabo no hace mucho, a mediados del año 2006.

La oficina mencionada opera actualmente en el piso trece de la Torre Interbank, en la calle Carlos Villarán #140, distrito de La Victoria. Está ubicada en ese lugar desde agosto del mismo año y tiene varias áreas con distintos requerimientos de iluminación. Por ejemplo, tiene una sala de exhibición, en la cual se exponen las computadoras y accesorios que venden; un área de entrenamiento, en la cual se realizan capacitaciones de personal y se dictan diversos cursos; varias

salas de reuniones de distintos tamaños; un pequeño comedor; etc. Sin embargo, para efectos de esta tesis nos vamos a concentrar en dos zonas principalmente: la sala de entrenamiento y el ambiente de cubículos, que es donde están la mayor cantidad de luminarias. Además, se detallará por zonas las luminarias que controla cada interruptor porque los usuarios consideran que la distribución actual es la adecuada y sólo se procederá a realizar el diseño para regular adecuadamente la intensidad luminosa según el área de trabajo.

Todo el local utiliza lámparas fluorescentes tubulares de 18 y 28 watts; además, siempre se instalan las de 18 watts en grupos de cuatro (de aquí en adelante se hará referencia a este conjunto de luminarias como **luminarias tipo A**) y las de 28 watts en grupos de dos (de aquí en adelante se hará referencia a este conjunto de luminarias como **luminarias tipo B**).

La sala de entrenamiento utiliza 36 lámparas de 18 watts y seis lámparas de 28 watts. En esta sala es importante implementar la regulación bajo demanda debido a que una de las paredes es, en realidad, una ventana muy grande y por ella ingresa la luz del ambiente de manera distinta a lo largo del día. Además, como es una sala en donde se llevan a cabo capacitaciones, hay un proyector instalado y es usual que el brillo que entra por la ventana afecte la visualización de lo proyectado, también de manera distinta a lo largo del día.



**Figura1: Sala de entrenamiento**

La **Figura 1** es una representación de la distribución de las luminarias de la sala, en donde cada cuadrado representa luminarias tipo A y cada rectángulo luminarias tipo B. Además, cada uno de los tres diseños distintos, representa las luminarias que controla cada uno de los tres interruptores de la sala. Los dos primeros controlan tres grupos de luminarias tipo A cada uno y el tercero controla tres grupos de luminarias tipo A y tres grupos de luminarias tipo B. El resumen de todo lo aquí descrito se muestra en la **Tabla 3**.

		Luminarias Tipo A (4 fluorescentes de 18W)	Luminarias Tipo B (2 fluorescentes de 28W)
Sala de entrenamiento	Interruptor 1	3	-
	Interruptor 2	3	-
	Interruptor 3	3	3

**Tabla 3: Control de iluminación actual en la sala de  
entrenamiento**

En lo que a iluminación respecta, el área común de cubículos de trabajo se divide en tres sectores y cada uno de ellos es operado desde un lugar distinto. En el primer ambiente se controlan, mediante seis interruptores, 56 luminarias de 18 watts y 12 luminarias de 28 watts. El primer interruptor controla cinco grupos de luminarias tipo A, el segundo interruptor controla tres grupos de luminarias tipo A, el tercer interruptor controla seis grupos de luminarias tipo A y los tres interruptores restantes controlan dos grupos de luminarias tipo A y dos grupos de luminarias tipo B, cada uno.

El segundo ambiente, mediante seis interruptores, utiliza 92 luminarias de 18 watts y 16 luminarias de 28 watts. Los dos primeros interruptores controlan seis grupos de luminarias tipo A cada uno, el tercer interruptor controla tres grupos de luminarias tipo A, otros dos interruptores controlan tres grupos de luminarias tipo A y tres grupos de luminarias tipo B cada uno y el sexto interruptor controla dos grupos de luminarias tipo A y dos grupos de luminarias tipo B.

El tercer y último ambiente, mediante dos interruptores, utiliza 36 luminarias de 18 watts. El primer interruptor controla cinco grupos de

luminarias tipo A y el segundo interruptor controla cuatro grupos de luminarias tipo A. El detalle del control descrito está resumido en la **Tabla 4** y se puede apreciar mejor en el **Anexo N°1**:

		Luminarias Tipo A (4 fluorescentes de 18W)	Luminarias Tipo B (2 fluorescentes de 28W)
Primer ambiente	Interruptor 1	5	-
	Interruptor 2	3	-
	Interruptor 3	6	-
	Interruptor 4	2	2
	Interruptor 5	2	2
	Interruptor 6	2	2
Segundo ambiente	Interruptor 1	6	-
	Interruptor 2	6	-
	Interruptor 3	3	-
	Interruptor 4	3	3
	Interruptor 5	3	3
	Interruptor 6	2	2
Tercer ambiente	Interruptor 1	5	-
	Interruptor 2	4	-

**Tabla 4: Control de iluminación actual en el área de cubículos**

### 3.4.2. Detalle de las luminarias actuales

Los dos tipos de luminarias usadas son de la marca PHILIPS. Las de 18 watts obedecen al estándar T8 y las de 28 watts obedecen al estándar T5. El detalle de las características de las luminarias se muestran en la **Tabla 5**.

Se ha considerado conveniente mantener el uso de las lámparas fluorescentes tubulares debido a que ya se cuenta con todos los dispositivos necesarios instalados. Sin embargo, es necesario adicionar algunos componentes al sistema de iluminación actual.

Descripción comercial	Pot. (W)	Base	Temp. de color (K)	Flujo lumin. (lm)	Eficacia lumin. (lm/W)	Índice reprod. de color (IRC)	Diám. (mm)	Largo (mm)
MASTER TLD18W/840	18	G13	4.000	1.350	75	85	28.0	604.0
MASTER TL5-28W-HE/840	28	G5	4.000	2.900	103	85	17.0	1163.2

**Tabla 5: Detalle de las luminarias empleadas**

### 3.4.3. Balastos y balastos electrónicos

Los balastos son los componentes auxiliares de las lámparas fluorescentes, que sirven para estabilizar la descarga en el interior del tubo y, en definitiva, la emisión de luz. Hasta hace pocos años, los balastos habituales eran las conocidas reactancias (formadas por una bobina de tipo inductivo) que satisfacían los requisitos de funcionamiento, aunque de forma poco eficiente desde el punto de vista energético. Las investigaciones realizadas en este campo permitieron averiguar que al incrementar la frecuencia de operación del tubo, aumentaba su eficacia luminosa y esto significa mayor cantidad de luz aportada por cada unidad de potencia eléctrica. Luego, para poder hacer funcionar el tubo a alta frecuencia, nacieron los balastos electrónicos.

Se ha decidido utilizar balastos electrónicos porque son los únicos dispositivos que existen hoy en día, que permiten regular la cantidad de luz que proporciona una lámpara fluorescente. Además, frente a las reactancias convencionales, los balastos electrónicos presentan las siguientes ventajas [16]:

- Ahorran energía, hasta un 25%, para la misma emisión de luz.

- Alargan la vida útil de la lámpara hasta 12.000 horas, es decir, un 50% más.
- Cuentan con encendido instantáneo, sin parpadeo.
- Permiten la desconexión automática en caso de lámpara defectuosa, lo que incluye fin de vida útil de la misma.
- Consiguen un factor de potencia próximo a la unidad.
- Existen balastos con regulación de luz, continua desde 1% a 100%, incluso en función de la aportación de luz natural (equipos especiales).
- Alcanzan la rentabilidad alrededor de las 5.000 horas de funcionamiento, por lo que se recomiendan especialmente en usos de conexión prolongada, típicos del sector comercial.

Por otro lado, en comparación con las reactancias convencionales, los balastos electrónicos presentan las siguientes desventajas:

- Sus dimensiones son mayores, lo que puede dificultar en algunos casos la sustitución directa y requerir, por tanto, el cambio del aparato de iluminación (luminaria).
- Su precio es, todavía, elevado; alrededor de los 30€ - 60€ (\$40 - \$80 dólares americanos), según modelos.

Sin embargo, como en la oficina de muestra se cuenta con una buena cantidad de espacio en el techo falso (1.05 mts. de altura), el hecho que haya o no una dimensión mayor en el balasto es irrelevante para la decisión. Respecto al precio elevado, como aún este proyecto está limitado al diseño y el precio de los balastos electrónicos del mercado está disminuyendo rápidamente, sí se justifica su uso, porque cumple

con las necesidades y porque en un futuro no muy lejano sí será fácilmente accesible.

#### **3.4.4. Sensor de luz, fotosensor o fotocélula**

Un fotosensor es un dispositivo que convierte la luz en una carga eléctrica, de manera que cuanto más brillante sea la luz, mayor será la carga proporcionada.

Los sensores de luz son utilizados para control, regulación y automatización de la iluminación en diferentes tipos de ambientes y se rigen por ciertos estándares, como por ejemplo el IEC 1547/EN 50082-1 (estándar genérico de inmunidad electromagnética [20]). También están los que definen características como el rango de temperatura ambiente de trabajo (5°C a 55°C), la humedad relativa (15% a 90%), el voltaje de control (1 – 10V), el voltaje de alimentación (12V), etc. para poder ser fácilmente integrables a otros sistemas.

Desde un comienzo, el diseño de este tipo de sensores para aplicaciones de iluminación ha tenido como objetivo principal regular la cantidad de luz proporcionada en un determinado ambiente. Obviamente, lo que se busca es lograr un ahorro energético, que se traduce en un ahorro económico, y esto es posible compensando los niveles excesivos de iluminación causados por el sobre-diseño de los sistemas (por ejemplo, 500 lux instalados, 400 lux requeridos) y/o la luz de día que ingresa. El ahorro energético se logra reduciendo la salida de luz artificial de manera que se logre mantener un nivel de iluminación preestablecido y constante en todo momento.



### 3.5. Interpretando los datos: Determinación de la solución

#### 3.5.1. Luminarias

Debido a que las características eléctricas de las luminarias actuales son adecuadas para el diseño que más adelante se detallará, no se adquirirán nuevas y de esa manera se puede ahorrar el costo de los fluorescentes y del cambio de infraestructura.

Además, comparando con otras marcas como Osram, se aprecia que los fluorescentes de 18 watts son iguales a los actualmente utilizados en la oficina, pero los de 28 watts tienen sólo 93 lm/W de eficacia luminosa y 2600 lm de flujo luminoso [17], contra los 103 lm/W y 2900 lm de Philips. Por otro lado, General Electric trabaja con fluorescentes de 25 watts y de 32 watts que no son fácilmente adaptables a los requerimientos actuales porque tienen distintas características [18], por lo que no se podrá trabajar más adelante de manera adecuada con los demás dispositivos elegidos.

#### 3.5.2. Balastos electrónicos

Para poder regular la intensidad de luz de un fluorescente, es necesario el uso de balastos electrónicos regulables y, de manera similar al caso de las luminarias, no podemos usar los de General Electric porque están diseñados para trabajar con fluorescentes de 25 y 32 watts.

En el caso de OSRAM es distinto porque las características de sus balastos electrónicos si se ajustan a las características eléctricas de las luminarias instaladas y obedecen normas y estándares:

- Tensión de alimentación: 220 a 240 V.
- Frecuencia de red: 0,5 a 60 Hz.
- Corte sobre el 80% del flujo luminoso nominal.
- Regulable desde 1% al 100%.
- Desconexión automática de seguridad de las lámparas en el caso de defecto o final de la vida (EOL).
- Reencendido automático después de reemplazar la lámpara.
- Índice de Eficiencia Energética EEI = A1, que es lo mejor existente para luminarias T8.
- Seguridad según IEC 61347-2-3, que es referente al modo seguro del uso de balastos electrónicos alimentados con corriente alterna.
- Rendimiento de la lámpara según EN 60929: Prescripción del funcionamiento de los estos balastos electrónicos con corriente alterna.
- Supresión de radiointerferencias según EN 55015, EN 55022.
- Contenido en armónicos según EN 61000-3-2.
- Requerimientos de inmunidad de equipos para iluminación en general ante interferencias por la red eléctrica o por campos interferentes según EN 61547.
- Control a través de interfaz desde 1 hasta 10 V (Estándar europeo).



**Figura 2: Balasto electrónico marca OSRAM**

Otra posibilidad es el balasto electrónico de Especialidades Luminotécnicas S.A. (ELT), que es una empresa española. Tiene más o menos las mismas características como son:

- Encendido por recalentamiento de cátodos.
- Dimensiones reducidas.
- Control de regulación por voltaje de 1 a 10 V.
- Rango de regulación de 1% a 100%.
- Ahorro de energía de hasta un 25%.
- Aumento del tiempo de vida de la lámpara en un 50%.
- Alto factor de potencia.
- Resistencia a la vibración según la norma IEC 68-2-6: 10 a 55Hz – 0.35mm.
- Certificación ENEC (Normas Europeas de Certificación Eléctrica).

En la **Tabla 6** se muestra un resumen de todas sus características según los modelos disponibles.

TIPO/REFERENCIA	DBT 118-W1	DBT 136-W1	DBT 158-W1	DBT 218-W1	DBT 236-W1	DBT 258-W1
Potencia lámpara (W.)	1x18 (T8)	1x36 (T8)	1x58 (T8)	2x18 (T8)	2x36 (T8)	2x58 (T8)
Tensión (V.)		220-240			220-240	
Tensión C.A. (V.)		198-264			198-264	
Frecuencia (Hz.)		0-50/60			0-50/60	
Frecuencia de funcionamiento (kHz.)	48-68	48-74	45-78	45-77	52-76	42-82
Corriente (A.)	0,1-0,04	0,16-0,04	0,26-0,05	0,18-0,05	0,33-0,07	0,49-0,09
Potencia absorbida (W.)	21-5	36-6	58-8	40-9	76-16	112-19
Índice de Eficiencia Energética (EEI)						
Factor de Potencia (λ)	0,96		0,98		0,98	
Armónicos en la red		conforme a EN 61000-3-2			conforme a EN 61000-3-2	
Supresión de RFI		conforme a EN 55015			conforme a EN 55015	
Homologaciones		CE			CE	
Tiempo de encendido (seg.)		<2			<2	
Temperatura máx. envolvente (tc °C)		70			70	
Temperatura de funcionamiento (ta °C)		+5...+50			+5...+50	
Temperatura de almacenamiento (°C)		-40...+85			-40...+85	
Ensayos vibración		IEC 68-2-6			IEC 68-2-6	
Corriente de dispersión (mA.)		<0,5			<0,5	
Peso (Kg.)		0,39			0,43	
Unidades por caja		28			28	
Código	9614000	9614010	9614020	9624000	9624010	9624020

**Tabla 6: Resumen de las características de los balastos electrónicos marca ELT**

Finalmente, el balasto electrónico de Philips también cumple con los requerimientos y se detallan a continuación:

- Arranque programado; sin parpadeo ideal para áreas con varios ciclos de apagado.
- Eleva la durabilidad hasta un 50% más que con balastos convencionales.
- Entrada del control de uno a diez voltios (estándar europeo).
- Frecuencia de operación mayor a 42 KHz.
- Voltaje de alimentación nominal entre 220 y 240 voltios.
- Frecuencia de alimentación 50/60 Hz.
- Frecuencia de operación mayor 42 KHz.
- Factor de potencia 0.95 en un 100% de energía.
- Voltaje de control entre 1 y 10 voltios.
- Regulación del flujo luminoso desde 3% hasta 100%

- Cumple con las normas: EN61000-3-2 (restricción del contenido armónicos), EN61547 (inmunidad ante interferencias de otros equipos), EN60928 (requerimientos generales y de seguridad para el uso de balastos electrónicos AC), EN60929 (prescripción del rendimiento de los balastos electrónicos para el correcto funcionamiento de la lámpara), IEC68-2-6 (resistencia a la vibración), IEC6-8-2-29 (resistencia a impactos).
- Tiene las siguientes aprobaciones: ISO 9001 (estándar de calidad), ISO 14001 (estándar ambiental), aprobación ENEC (Normas Europeas de Certificación Eléctrica), aprobación EMV (estándar de interoperabilidad de tarjetas) y marcación CE (Conformidad Europea).

Para uniformizar el sistema de regulación, con los componentes que ya están en funcionamiento, se ha decidido utilizar los balastos electrónicos de la marca Philips. En la **Tabla 7** se detallan las características de los modelos específicos que se van a usar:

HF-R 418 TLD	HF-R 228 TL5
Arranque programado	Arranque programado
Número de lámparas: 4 x 18W	Número de lámparas: 2 x 28W
Voltaje: 220V	Voltaje: 220V
Frecuencia: 50/60 Hz.	Frecuencia: 50/60 Hz.
Corriente de red: 0,34 (A)	Corriente de red: 0,28 (A)
Potencia total: 83(W)	Potencia total: 62(W)
Factor de potencia: 0,99	Factor de potencia: 0,99
Factor del flujo luminoso: 0,03 - 1,00	Factor del flujo luminoso: 0,03 - 1,00
Distorsión armónica: 10%	Distorsión armónica: 10%

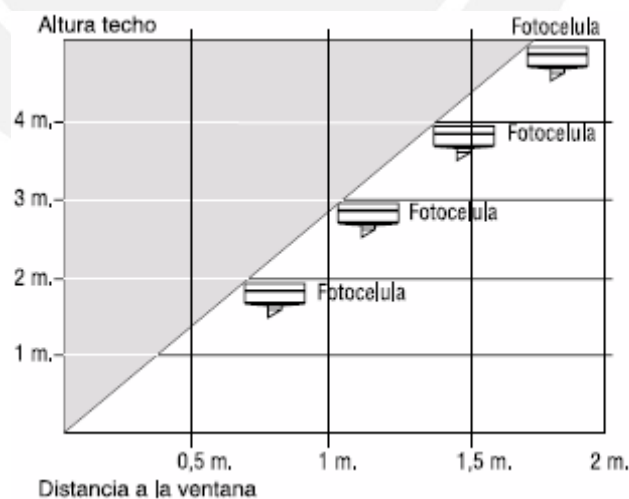
**Tabla 7: Características eléctricas de los balastos electrónicos**

**marca Philips**

Estos se han escogido principalmente debido a que en la oficina siempre se trabaja con las lámparas fluorescentes de 18 watts en grupos de cuatro (luminaria tipo A) y con las de 28 watts en grupos de 2 (luminaria tipo B). De los balastos electrónicos encontrados, el único que maneja exactamente esos números de fluorescentes por luminaria instalada es el de la marca Philips.

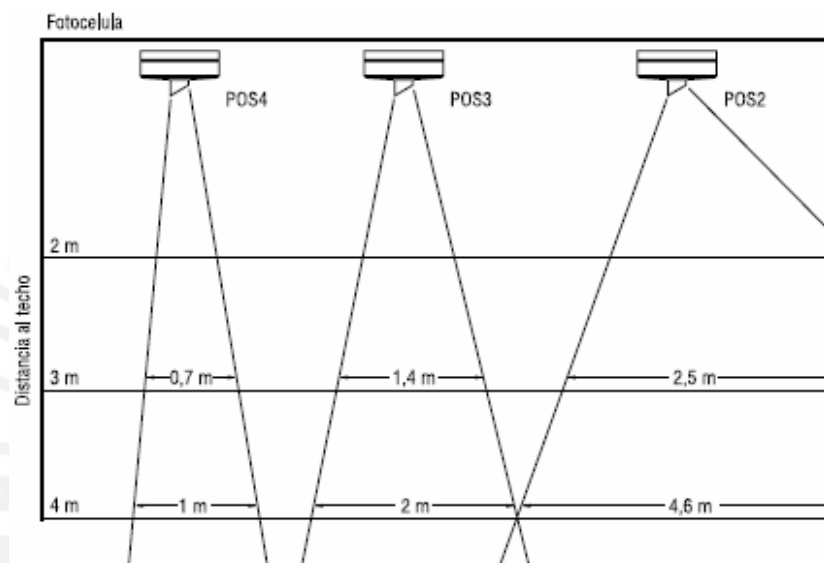
### 3.5.3. Fotocélula

La primera que se considerará es la fotocélula SDB de la compañía española ELT. Esta permite mantener constante el nivel de iluminación mediante un sistema basado en la lectura del nivel de luz del día. Además, si es utilizada en conjunto con un amplificador y un potenciómetro que ellos venden, puede controlar de 1 a 100 balastos electrónicos regulables.



**Figura 4: Distancia ventana/altura del techo**

Para el uso correcto de la fotocélula es muy importante el posicionamiento de la misma. El ángulo de visión no debe estar influenciado por otros puntos de luz o por cambios localizados en el nivel de iluminación. Así, al momento de diseñar una instalación, deben tenerse en cuenta los diagramas correspondientes a las **Figuras 4 y 5** [19].



**Figura 5: Posición fotocélula/ventana**

Otro detalle es que cuando el control de la iluminación lo está ejerciendo la fotocélula, se activa el diodo LED rojo [19] que se muestra en la **Figura 6**:



**Figura 6: Fotocélula marca ELT**

Otra fotocélula muy interesante es la LuxSense de Philips que se conecta directamente al balasto electrónico. Ofrece salidas de 1 a 10 voltios DC, puede controlar hasta 20 balastos electrónicos de Philips sin necesidad de amplificador alguno, se puede ajustar el “setpoint” de manera manual con tan sólo rotar el diafragma y su ligero peso de 20 gramos hace que sea posible la instalación en la misma luminaria.

En realidad, este dispositivo es parte del sistema de control propio de Philips, llamados Tríos. Ha sido fabricada pensando justamente en poder utilizarla para controlar la iluminación en oficinas donde aproximadamente se considera que debe haber 500 lux sobre un escritorio de 70 centímetros de altura estando el techo a unos 2.5 metros de altura. Además de esta consideración, cuenta con una perilla que permite terminar de regular el flujo luminoso proporcionado de manera que se logre obtener los 500 lux en el área determinada. Finalmente, el hecho de que cumpla las normas ENEC y la marcación CE, hace que sea fácilmente adaptable a otros sistemas de control.





**Figura 7: Fococélula marca Philips**

De la misma manera, la fotocélula elegida también será Philips (ver **Figura 7**), así logramos tener un sistema más robusto, cuyas partes han sido diseñadas para trabajar en conjunto. Además, al menos por el momento, Philips es la única empresa en el Perú que puede proporcionarnos este tipo de equipos en poco tiempo.

#### **3.5.4. Comunicación inalámbrica**

El principal motivo por el que se está escogiendo la comunicación inalámbrica es para evitar el tedioso proceso del adecuado cableado estructurado. Lo que queda ahora es determinar que solución inalámbrica se va a emplear.

Si bien los sistemas mediante corrientes portadoras no son exactamente inalámbricos, pueden considerarse como tal porque no requieren de cableado específico para el control. Pero ahora se debe tomar en cuenta la complejidad de los dispositivos que se van a emplear, porque debemos recordar que se necesitan moduladores y demoduladores en cada punto. Este tipo de sistemas era sumamente útil en el pasado cuando no se contaba con la tecnología con la que se cuenta actualmente. Si lo que se busca es un sistema que sea robusto, no se puede correr el riesgo de que la propia red eléctrica sea la que afecte el funcionamiento del mismo (por las numerosas interferencias que introduce). También podría ocurrir que la comunicación entre dispositivos se vea truncada por falta de una conexión física, por ejemplo, si se pasa por un transformador.

Debido a los detalles expuestos, ahora se pasará a analizar los sistemas que utilizan su propia red de datos, pero sin olvidar que vamos a prescindir del cableado. Además, como lo que estamos buscando es un sistema potente, deberíamos utilizar un sistema distribuido y como sus dos desventajas frente a los sistemas centralizados son la programación compleja y el costo, debemos buscar dispositivos baratos y de programación simple.

Hace ya algún tiempo, los fabricantes de controladores, venían buscando la manera de cubrir la actual creciente demanda del mercado de baja frecuencia de datos, de una manera económica.

Aquí es precisamente donde entra el estándar Zigbee, porque por ser una estándar (lo que implica compatibilidad) y por la gran preocupación de los fabricantes por hacer un dispositivo de bajo costo, nos ayudan a superar una de las desventajas. La segunda desventaja se superó prácticamente de una manera automática debido a que el lenguaje de programación que utilizan muchos de los controladores Zigbee actuales es el de la familia 8051 para ser de bajo consumo, y prácticamente cualquier programador conoce dicho lenguaje, pero lo interesante está en que ya está disponible en muchos otros lenguajes así que se puede programar usando controladores de la familia ATMEL y también con algunos PICs.

También cabe destacar que, si bien Zigbee está destinada a la electrónica de consumo y al control industrial, recién es ahora que está empezando a tomar fuerza y esto hace que al plantear la solución de la regulación de iluminación basada en este estándar, nos dé como resultado un sistema moderno. Posteriormente, se podrá integrar a sistemas de mayor orden sin necesidad de hacer grandes cambios a lo que ya esté instalado y operativo.

El módulo de Zigbee escogido para efectos de la simulación de esta tesis será el módulo de desarrollo Zigbee de 2.4 GHz. de Silicon Labs, porque cumple con el requerimiento de dispositivos integrados necesarios para poder realizar las pruebas de funcionamiento

pertinentes para la red de control deseada y porque la Pontificia Universidad Católica del Perú ya cuenta con el módulo.



## **CAPÍTULO 4: DISEÑO DEL SISTEMA DE REGULACIÓN PARA UNA OFICINA EN LIMA.**

### **4.1. Consideraciones preliminares.**

Es importante tener en cuenta que, debido a la similitud de todas las áreas de la oficina en cuestión, sólo se realizará y detallará el diseño de una de ellas, la cual será considerada genérica y facilitará la adaptación para cada una de las áreas reales. Adicionalmente, se debe considerar que el control central no estará empotrado en la pared, sino que será móvil y funcionará con baterías, similar a un control remoto de un televisor. Los por menores y alcances del mismo se detallarán más adelante.

### **4.2. Análisis y diseño del sistema.**

Lo primero que se debe analizar es qué componentes se van a utilizar y el precio de los mismos. En este caso en particular, se buscará reducir lo mayor posible el tamaño del control remoto y por este motivo se utilizarán componentes de montaje superficial; por otro lado, no es necesario reducir el tamaño de las tarjetas receptoras debido a que ellas estarán ubicadas en el falso techo y ese espacio es bastante amplio por lo que nos centraremos en disminuir el costo de los componentes de las tarjetas y no tanto en el tamaño.

A continuación se describirán algunas características con las que deben cumplir el microcontrolador utilizado y el transmisor-receptor de radiofrecuencia. Luego, se enumerarán los principales

requerimientos, tanto de las tarjetas maestras de control, como de las tarjetas de recepción, cuya acción en sí será la del control de la intensidad de la iluminación. Debemos recordar también, que tratándose de la sala de capacitaciones, son tres los interruptores que se están reemplazando; en el caso de la primera y segunda parte del área de cubículos, son seis los interruptores que se están reemplazando en cada una y en el caso de la tercera parte del área de cubículos, son sólo dos los interruptores que se van a reemplazar.

#### **4.2.1. El microcontrolador.**

Se debe utilizar un microcontrolador que sea capaz de manejar salidas digital/análogas y además que consuma poca corriente para que vaya de acuerdo con lo que se pretende lograr con el uso del Zigbee en cuanto al ahorro energético.

También debe ser capaz de establecer comunicación de tipo SPI, es decir que debe tener los pines de comunicación de datos CSN, SCLK, MOSI y MISO, porque son los que manejan los transmisores-receptores de radiofrecuencia Zigbee.

Debido a estos requerimientos es que se ha escogido el microcontrolador C8051F121 de la marca Silicon Laboratories: Este cuenta con dos convertidores digital-análogo, es de bajo consumo de potencia y tiene los pines que permiten la comunicación SPI.

## 4.2.2. Transmisor-receptor de radiofrecuencia de 2.4 GHz IEEE

### 802.15.4

Este dispositivo es quien realmente se encarga de la transmisión de datos utilizando el estándar Zigbee. Su funcionamiento consiste básicamente en transmitir la trama que uno le envíe desde el microcontrolador mediante las señales CSN, SCLK, MOSI y MISO (comunicación SPI). El formato de la trama debe corresponder con el formato de la trama Zigbee para lograr la correcta recepción en el módulo destino.

El dispositivo elegido es el decodificador CC2420 de marca Chipcon, debido a que este transmite los datos utilizando el protocolo Zigbee a 2.4 GHz., también es de bajo consumo de energía, se alimenta con 3.3 voltios (igual que el microcontrolador elegido) y además agrega de manera automática el Preamble Sequence y el Start of Frame Delimiter (SFD) como encabezado a la trama de datos y el Frame Check Sequence (FCS) al final de la trama. Estos tres segmentos que adiciona son los que ayudan a lograr la correcta transmisión de los datos en el formato Zigbee. Sin embargo, si bien facilita la comunicación Zigbee, primero se debe configurar, y para lograr dicho fin se deben inicializar los registros internos mediante la reinicialización del decodificador. Esto se consigue mandando un cero lógico por un periodo aproximado de 1ms.

Después, se deben configurar los registros y espacios de memoria RAM. En primer lugar, se configuran los registros los cuales son de dos tipos: STROBES y REGISTROS de lectura y escritura. Los

registros de tipo STROBES son comandos que se le envían al decodificador para que realice una acción específica, es así que se usan los siguientes:

- SXOSCON: Prender el oscilador interno
- STXCAL: Habilita y calibra el oscilador interno
- SRXON: Habilita la recepción
- STXON: Habilita la transmisión
- SFLUSHRX: Vacía el buffer RX FIFO
- SFLUSHTX: Vacía el buffer TX FIFO

En cuanto a los registros de lectura y escritura, además de los valores del registro por defecto se van a configurar los siguientes:

- MDMCTRL0: Se habilita el AUTOACK, se deshabilita el ADR\_DECODE que sirve para decodificar la dirección de hardware.
- MDMCTR1: CORR\_THR=20 – Valor umbral del demodulador, requerido antes de empezar a buscar el SFD. El valor asignado es una recomendación del fabricante.
- IOCFG0: FIFOP\_THR=127 - número de periodos que va a esperar a que el FIFOP se ponga en alta.
- SECCTRL0: RXFIFO\_PROTECTION=0 porque no se utiliza seguridad a nivel MAC.

Por último, se configura el Short Address (dirección propia del dispositivo en una red personal) y el PAN identifier (nombre de la red personal) para esto se hace uso de la memoria RAM. Para cambiar

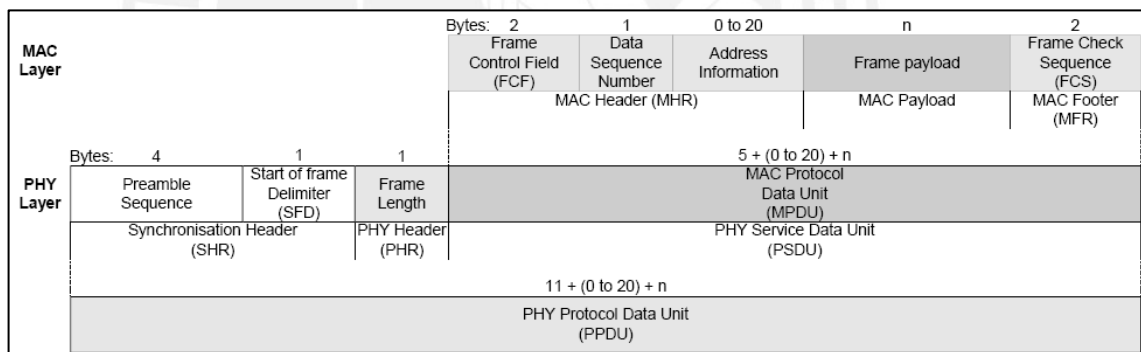


dichos valores se accede a espacios de memoria específicos siendo estos los siguientes:

- SHORTADR (0x16A) = 0xFFFF
- PANID (0x168) = 0xFFFF

Estos valores se toman para que reconozca cualquier dispositivo los datos enviados.

Un detalle adicional de este dispositivo es que algunos de sus pines de conexión necesitan un voltaje de referencia de 1.8 voltios, pero el mismo integrado tiene un regulador de voltaje de ese valor que sale por su pin 42, sin embargo, para poder utilizarlo debe ser habilitado alimentando con 3.3 voltios su pin 43 **(ver anexos 2 y 3)**.

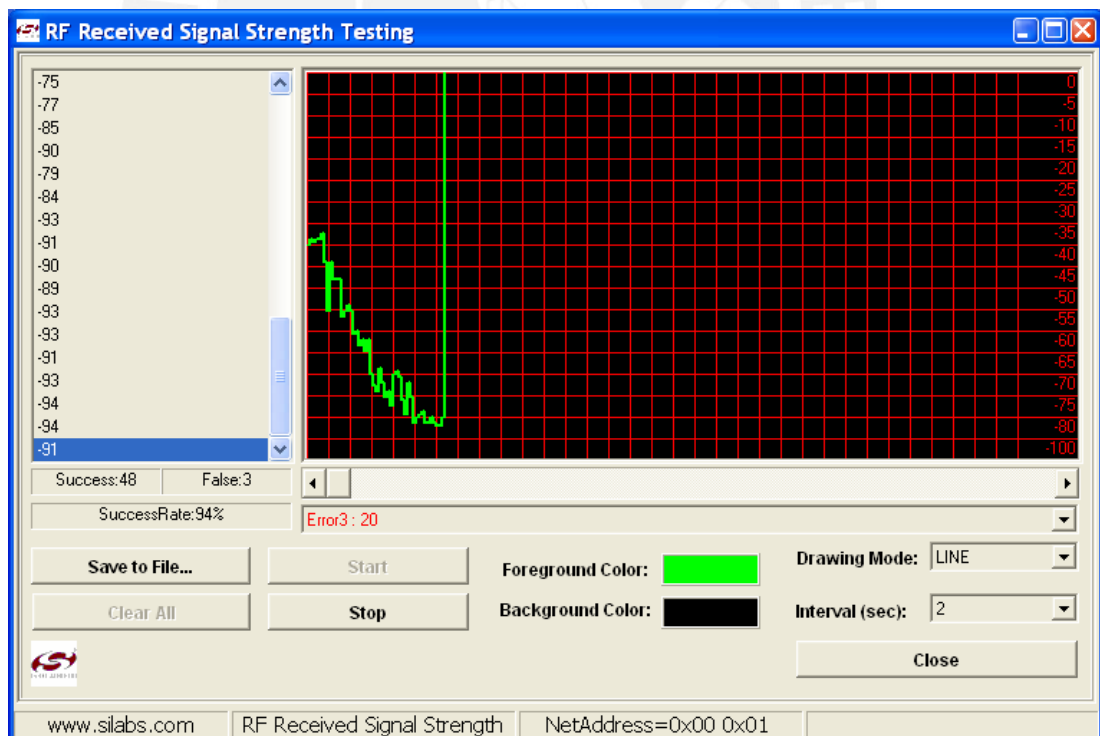


**Figura 8: Formato de la trama 802.15.4 (Fuente: Hoja de datos de Chipcon)**

### 4.2.3. La antena

La antena seleccionada es la de Linux Technologies modelo 2.4GHZ 1/4 wave whip debido a que sólo se puede utilizar esa o la de tipo SMA (en chip integrado) y además del hecho de ser más barata, tiene un muy buen alcance. Se probó la transmisión-recepción en la oficina del diseño, primero con la antena horizontal y luego con la antena

vertical, para los casos máximos de extremo a extremo del área de cubículos (38 metros aprox.) y la intensidad de la señal presentaba un porcentaje de éxito de la comunicación de 94% (**ver Figura 9**) para la antena horizontal y 100% (**ver Figura 10**) para la antena vertical. También se hizo la prueba para el caso máximo de la distancia del área de control (entre 8 y 10 metros) en cuyo caso el porcentaje de éxito de la señal es de 100% sin importar la dirección de la antena (**ver Figuras 11 y 12**). También es importante dar a conocer que, en el caso de esta oficina, todos los muros interiores son prefabricados y en el área de cubículos ni siquiera hay estos muros, por lo que la intensidad de la señal no se ve perjudicada.



**Figura 9: Prueba de distancia máxima con la antena horizontal**

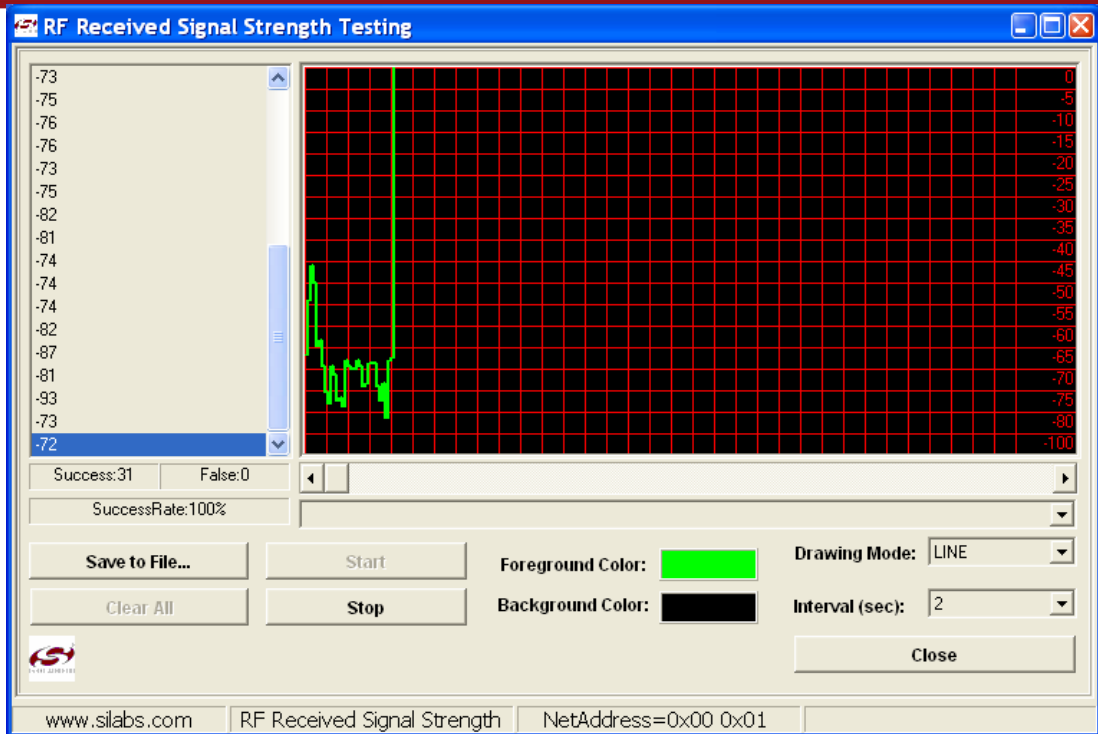


Figura 10: Prueba de distancia máxima con la antena vertical

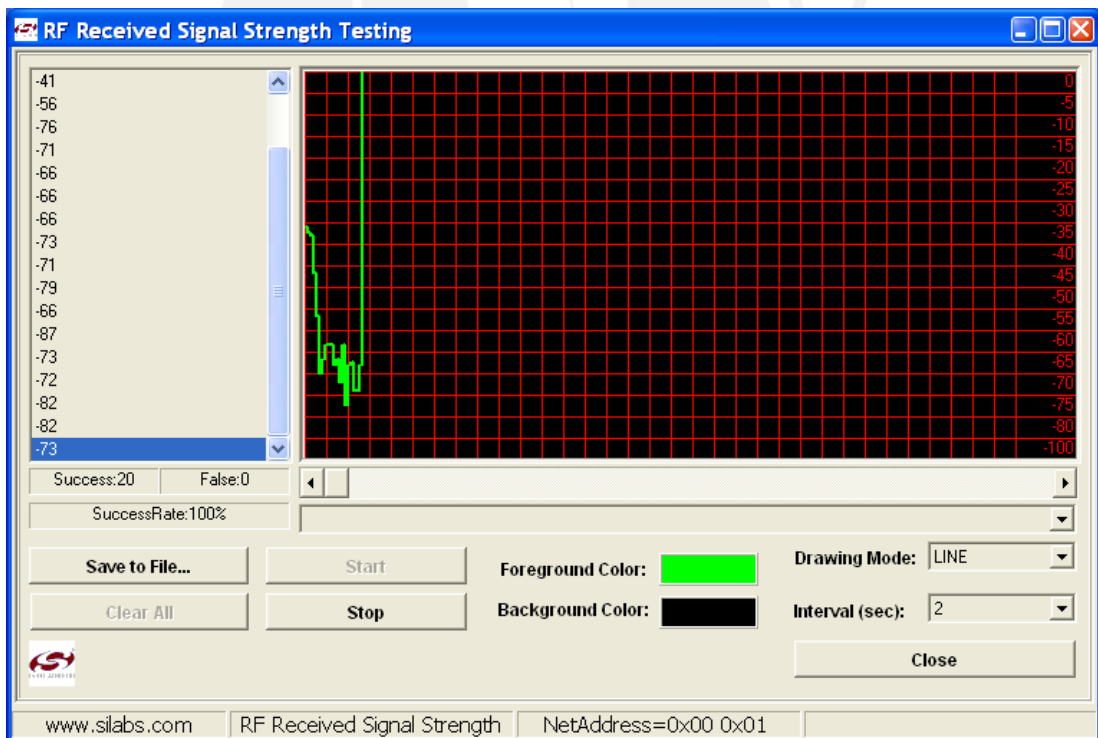
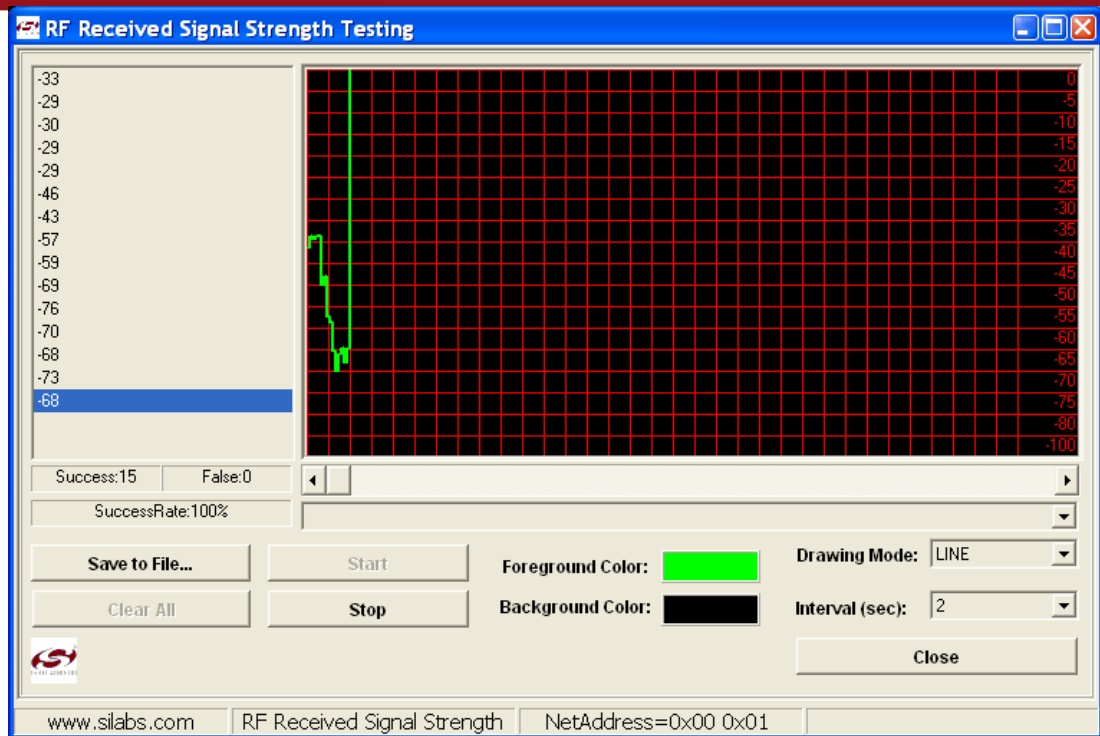


Figura 11: Prueba de distancia máxima de trabajo con la antena horizontal



**Figura 12: Prueba de distancia máxima de trabajo con la antena vertical**

En todas las pruebas se empezaba con el transmisor junto al receptor y se empezaba a alejar el mismo hasta el punto más distante posible.

### 4.3. Desarrollo de la solución

Lo que se va a implementar es un sistema que permita reemplazar los interruptores actuales por “interruptores Zigbee” con intensidad de luz regulable.

En la sala de entrenamiento el control se hará manual y no estará disponible la opción de regulación automática debido a que los usuarios del local están de acuerdo en que sea el expositor el que pueda regular la iluminación en todo momento. Por otro lado, en el área de cubículos normalmente se utilizará el control automático, pero puede darse el caso de que requieran regular la intensidad de forma

manual, motivo por el cual estarán disponibles ambos métodos. La distribución de las luminarias se está manteniendo, al igual que la distribución de las luminarias controladas por cada interruptor. En el **anexo N°1** se puede apreciar cuáles son las luminarias controladas por cada interruptor (áreas pequeñas) y también se ha indicado cuales de estas zonas serán controladas por cada fotocélula, así como la ubicación de las mismas (áreas ovaladas). Si bien la fotocélula realmente no puede “ver” la totalidad del área cobertura seleccionada, se ha escogido la distribución según los diferentes ambientes de trabajo. Así cada usuario distinto puede ajustar la iluminación según su necesidad y la regulación automática funcionará adecuadamente debido a que dentro de cada uno de estos ambientes los requerimientos de luz son iguales. Finalmente, se concluyo que la zona de los cubículos se controlará con dos módulos de control. Uno se encargará de los seis “interruptores” (áreas pequeñas) más cercanos a la entrada y el otro de los ocho “interruptores” más alejados.

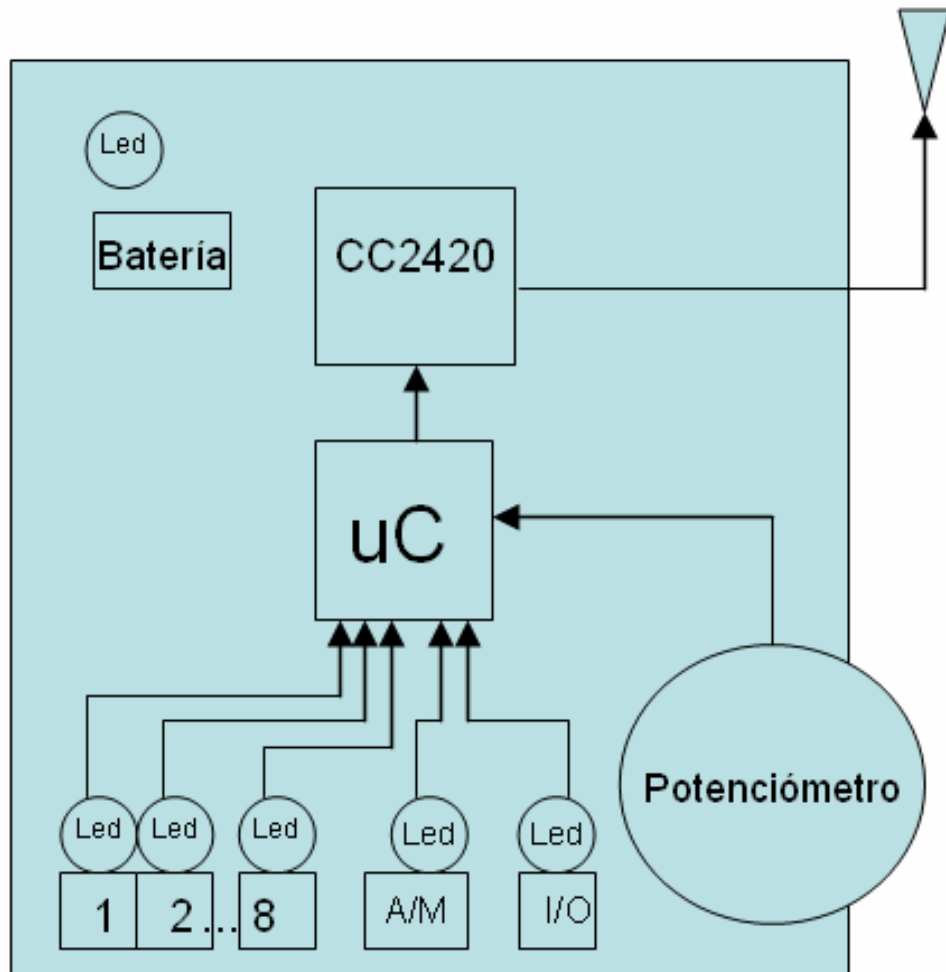
Como cada módulo receptor tiene un microcontrolador con dos conversores digital/análogo, cada módulo podrá controlar hasta dos interruptores, de manera que se necesitarán un total de tres módulos transmisores (uno para la sala de entrenamiento y dos para el área de los cubículos) y nueve módulos receptores (dos para la sala de entrenamiento, tres para la zona de ingreso a los cubículos y cuatro para la zona de los cubículos más alejada). A continuación se detallarán los circuitos utilizados para lograr esta solución.

#### 4.3.1. Módulo de control maestro (transmisor).

Como su nombre lo indica, es aquí donde está el control central principal. Desde este módulo el usuario será capaz de encender, apagar y regular la intensidad luminosa, ya sea de manera manual o automática y por sectores.

Este módulo deberá incluir los siguientes puertos de entrada (**ver Figura 13**):

- Un pulsador conectado a un led para comprobar que la batería se encuentra en buen estado.
- Ocho pulsadores y cada uno servirá para seleccionar un interruptor diferente de entre los actualmente usados. Cada uno de estos pulsadores tendrá un led asociado para saber cuál de ellos está seleccionado.
- Un pulsador para la el encendido o apagado del conjunto de luminarias seleccionadas. Habrá un led asociado a este pulsador.
- Un pulsador para la selección del modo automático/manual del conjunto de luminarias seleccionadas. Habrá un led asociado a este pulsador.
- Un potenciómetro que servirá para regular la intensidad de la iluminación. Por este requerimiento va a ser necesario que el microcontrolador elegido tenga una entrada análoga/digital.



**Figura 13: Módulo de transmisión**

Esta tarjeta es alimentada con una batería de 9 voltios y tiene un pulsador que se usará para probar que la batería esté alimentando correctamente a la tarjeta. Tiene también un regulador de voltaje (LM2936) que sirve para obtener un voltaje regulado a 3.3V y es con este voltaje con el que se alimentará tanto el microcontrolador como el decodificador CC2420. Además, por una de las entradas análogo/digital del microcontrolador ingresará un voltaje que se variará con un potenciómetro y que servirá para regular la intensidad luminosa de las luminarias cuando se esté en modo manual. También

se tienen ocho pulsadores que servirán para seleccionar cuál grupo de luminarias es el que se va a controlar y cada uno, al ser seleccionado, enciende un led indicador para una fácil identificación. Finalmente, hay otros dos pulsadores (también con un led indicador cada uno) que sirven para el encendido y apagado de cada grupo de luminarias en conjunto y para la selección del modo en el que actuarán, ya sea automático o manual (**ver Anexo N°2**).

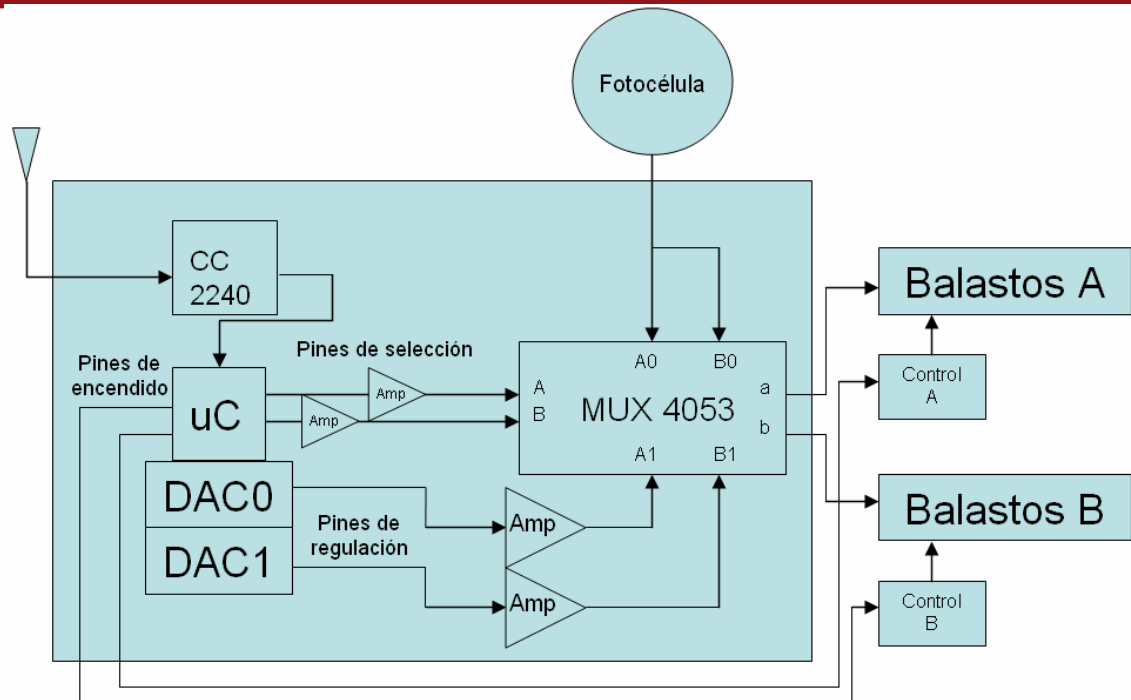
#### 4.4. Módulo de regulación de la iluminación (receptor).

Este módulo va a recibir órdenes enviadas por el módulo de control maestro y con esto procederá a regular la intensidad luminosa según lo requerido.

Incluirá lo siguiente (**ver Figura 14**):

- Una etapa de acondicionamiento de señal a la salida de cada conversor digital/análogo.
- Una etapa de acondicionamiento de señal a la salida de cada pin para la selección del modo de regulación, ya sea automático o manual.
- Una entrada para la señal de la fotocélula.
- Un multiplexor analógico que servirá para la conmutación entre el modo de regulación automático y manual.
- Una etapa de acondicionamiento para el control del encendido y apagado de cada grupo de luminarias.





**Figura 14: Módulo de recepción**

El funcionamiento detallado de cada una de las partes se presenta a continuación y se puede apreciar en el **Anexo N°3**.

#### 4.4.1. Multiplexor analógico.

La necesidad de utilizar un multiplexor analógico se debe a que la fotocélula nos proporciona señales DC que varían entre 1 y 10 voltios y estos valores de voltaje son muy elevados para que el microcontrolador pueda manejarlos, sin embargo, es exactamente el rango de voltaje DC que necesita el balasto electrónico. Por lo tanto, la solución más práctica fue generar una salida de control en el microcontrolador que logre manejar el mismo rango y luego, agregar al módulo un multiplexor analógico que pueda manejar ese rango de

voltaje y facilite la selección entre el modo de regulación automático (mediante la fotocélula) o manual (mediante el potenciómetro). El multiplexor elegido es el CD4053BC que puede seleccionar entre dos señales para tres entradas distintas y funciona hasta con 15 voltios.

La ventaja es que con tan sólo 3mA se pueden controlar hasta 20 balastos electrónicos y en este caso el multiplexor puede soportar en su salida más de 9 miliamperios, lo que permitiría en un caso ideal controlar hasta 60 balastos, pero en el caso de la aplicación presente como máximo se requiere que controle hasta 12 balastos electrónicos.

#### 4.4.2. Acondicionamiento de la señal del conversor digital/análogo

Un detalle adicional es que la salida del conversor digital/análogo del microcontrolador no puede generar directamente un rango de 1 a 10 voltios DC así que debe amplificarse. La salida real del pin digital/análogo varía de 0 a 2.43 voltios así que para llegar a la salida real requerida primero se amplificará con un factor de 3.7 para lograr un rango de 0 a 9 voltios. Esto se logró utilizando un amplificador operacional (OPAMP) con un factor de ganancia que obedece a la fórmula  $G = 1 + R2/R1$ , donde  $R1 = 10K\Omega$  y  $R2 = 27K\Omega$  (ver **Figura 16**). Ahora lo único que falta es elevar un voltio a todo el rango de la salida actual para lograr el rango de salida deseado de 1 a 10 voltios. Para esto se requiere un circuito sumador de un voltio y se consigue con otro amplificador operacional y 4 resistencias de  $10K\Omega$  (ver **Figura 16**).

Ahora se presenta otro inconveniente y es la necesidad de generar un voltio regulado. Para esto se utiliza el circuito mostrado en la **Figura 15**. En donde el potenciómetro de  $100\text{K}\Omega$  (R19) sirve para eliminar el offset del OPAMP en configuración seguidor de voltaje y el diodo D5 es un Zener de 2.7 voltios que ayuda a regular, mediante un divisor de voltaje, que se entregue 1 voltio a la entrada del OPAMP seguidor.

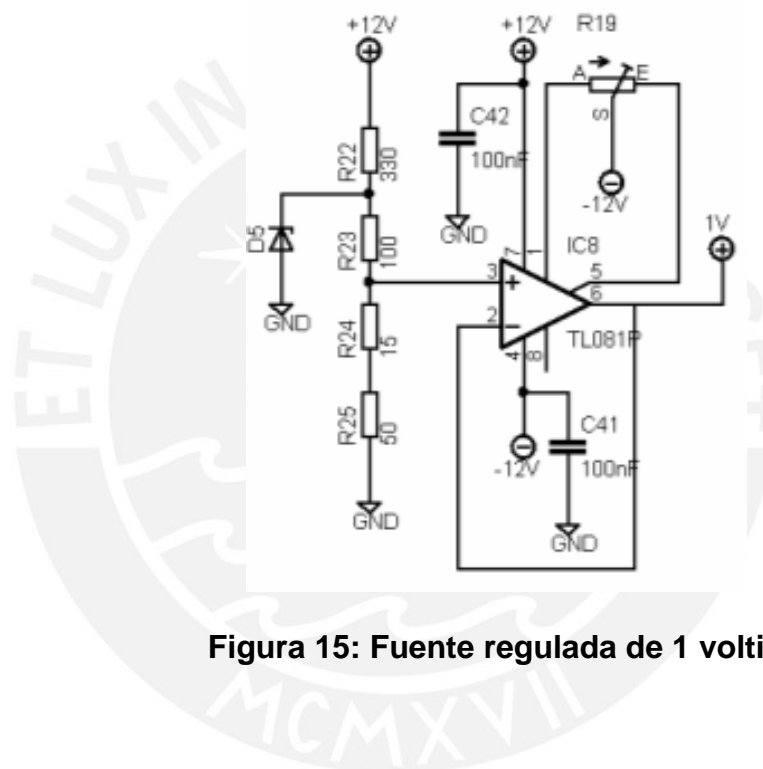
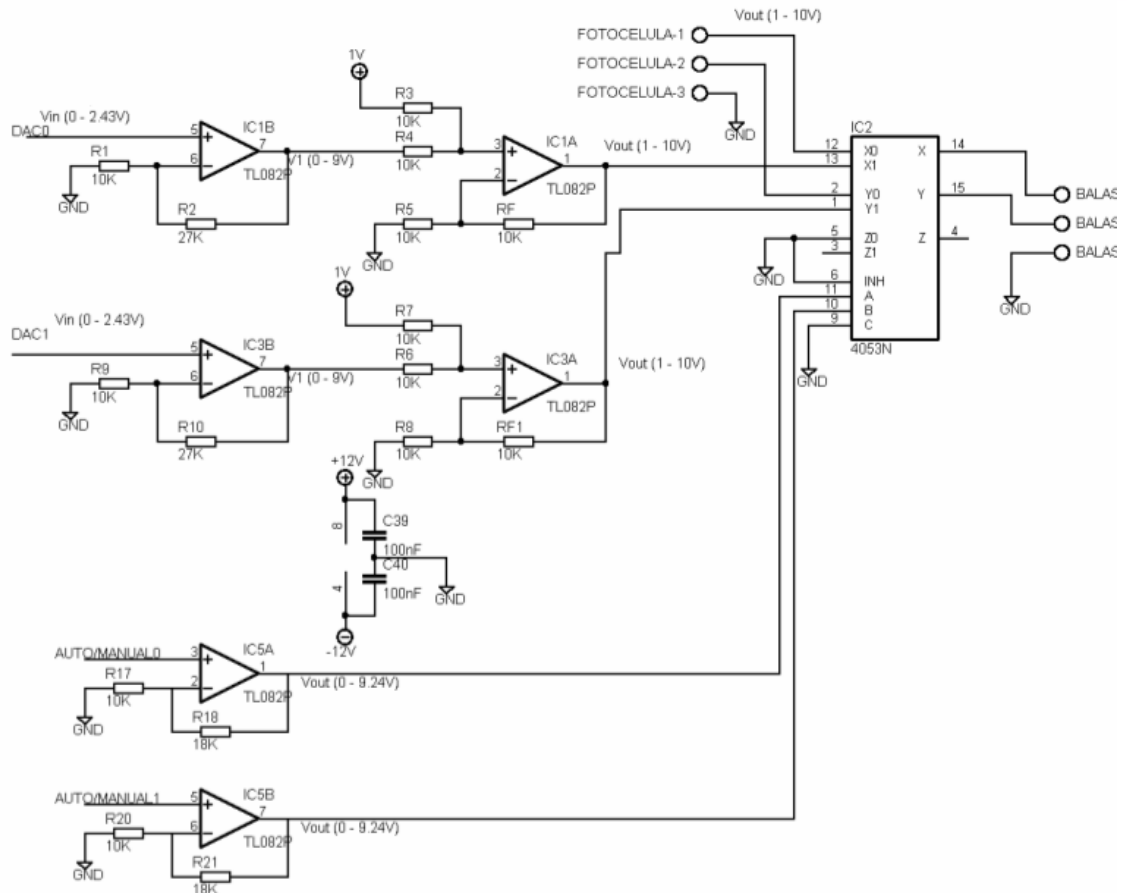


Figura 15: Fuente regulada de 1 voltio



**Figura 16: Diagrama esquemático del multiplexor**

Finalmente, en la parte inferior de la **Figura 16**, se puede apreciar el amplificador para el control de selección entre automático y manual del multiplexor. En este caso la señal proporcionada por el puerto de salida del microcontrolador necesita ser adaptada desde 0 a 3.3 voltios hasta un valor de 0 a 8 voltios por lo menos para lograr conmutar la entrada del multiplexor. En este caso la ganancia escogida tiene un factor de 2.8 (utilizando la misma fórmula que se utilizó para adaptar la señal del DAC), logrando un rango de 0 a 9.24 voltios, lo que es más que suficiente.

Con esto el multiplexor puede seleccionar entre el modo automático (fotocélula) o el modo manual (microcontrolador) y además se está

tomando en cuenta la posibilidad de que cada entrada del multiplexor pueda tener una fotocélula diferente conectada.

#### 4.4.3. Acondicionamiento de la señal de control de encendido y apagado de las luminarias

Para lograr controlar el encendido y apagado de las luminarias es necesario el uso de un relé que pueda manejar los niveles de corriente adecuados, de acuerdo a los interruptores actuales, como se detalla en la **Tabla 8**:

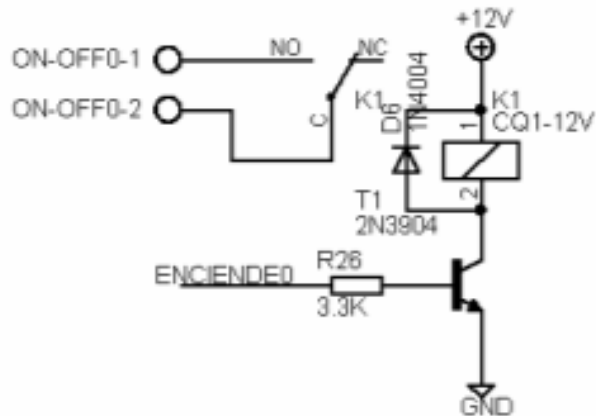
		Luminarias Tipo A - 0.34A (4 fluorescentes de 18W)	Luminarias Tipo B - 0.18A (2 fluorescentes de 28W)	Corriente Total (A)
Primer ambiente de cubículos	Int. 1	1.7	-	1.7
	Int. 2	1.02	-	1.02
	Int. 3	2.04	-	2.04
	Int. 4	0.68	0.36	1.04
	Int. 5	0.68	0.36	1.04
	Int. 6	0.68	0.36	1.04
Segundo ambiente de cubículos	Int. 1	2.04	-	2.04
	Int. 2	2.04	-	2.04
	Int. 3	1.02	-	1.02
	Int. 4	1.02	0.54	1.56
	Int. 5	1.02	0.54	1.56
	Int. 6	0.68	0.36	1.04
	Int. 7	1.7	-	1.7
	Int. 8	1.36	-	1.36
Sala de entrenamiento	Int. 1	1.02	-	1.02
	Int. 2	1.02	-	1.02
	Int. 3	1.02	1.02	2.04

**Tabla 8: Consumo de corriente por relé**

El relé seleccionado opera con 12 voltios y puede controlar hasta 5 amperios en el lado de los 220VAC, lo que nos da un margen más que

suficiente ante el caso de que se presente una sobrecarga en el encendido de cualquiera de ellos.

El circuito de control utiliza un transistor NPN (2N3904) en modo corte y saturación como se muestra en la **Figura 17**:



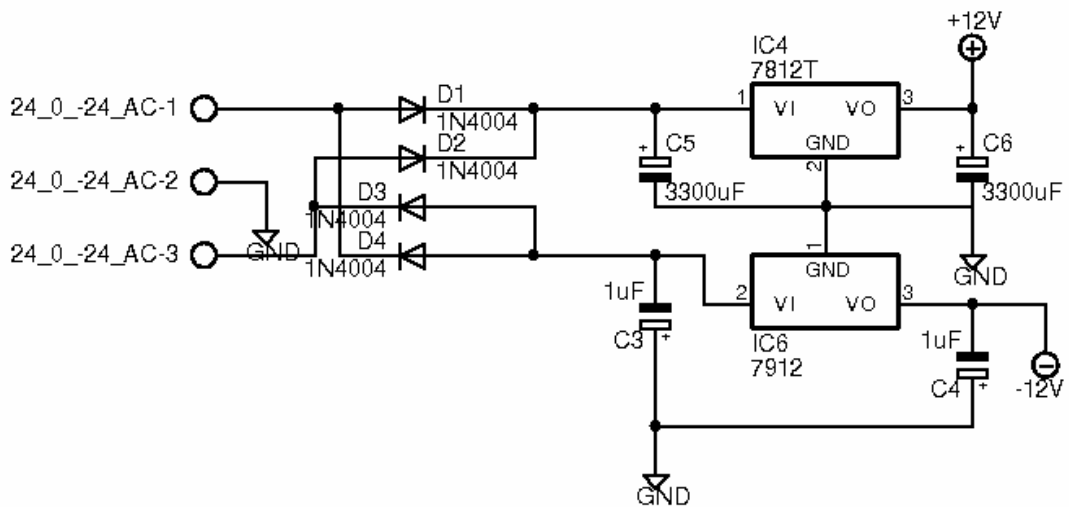
**Figura 17: Control de encendido y apagado de las luminarias**

Con la resistencia de  $3.3K\Omega$  el microcontrolador sólo tiene que entregar  $750\ \mu A$ . y el transistor puede entregar sin problemas a la bobina del relé los  $29\ mA$ . que requiere. El diodo en paralelo con la bobina es para proteger al resto del circuito ante sobrecargas en la bobina dado que permite que esta se descargue con facilidad.

#### 4.4.4. Fuentes de alimentación.

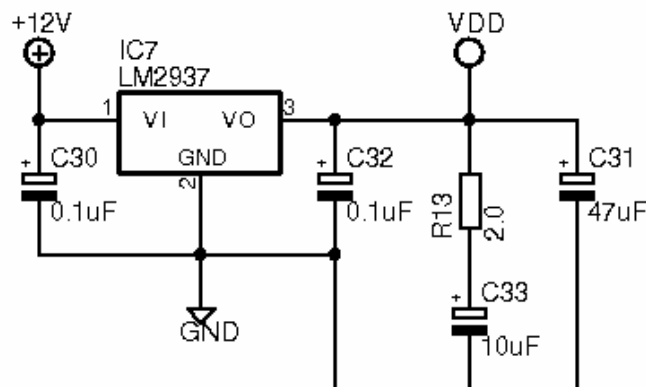
En la **Figura 18** se detallan los componentes de la fuente de alimentación que proporciona los  $\pm 12\ VDC$  a partir de un transformador AC de  $-24 - 0 - 24$  voltios y 2 amperios. Se decidió utilizar una fuente DC de  $\pm 12\ V$  debido a que la fuente regulada de un voltio (que se necesita para el circuito sumador de  $1V$ ) sólo trabaja

correctamente si se elimina el voltaje de offset y para realizar dicha tarea se debe alimentar el amplificador operacional, que se utiliza como regulador, con un voltaje positivo y otro negativo de la misma magnitud. Luego, con un potenciómetro se termina de ajustar el amplificador operacional para eliminar por completo dicho offset (**ver Figura 15**).



**Figura 18: Fuente de alimentación de -12 – 0 – +12 VDC**

Además, en la **Figura 19**, también se detalla el circuito regulador de voltaje de 3.3 voltios que es igual al que se usa en el circuito transmisor.



**Figura 19: Fuente de alimentación de 3.3 VDC**

#### 4.5. Simulación de la instalación y funcionamiento.

Para la simulación de la instalación se construyó un circuito de prueba **(ver Figura 20)** en el que se comprobó que los valores seleccionados para los componentes en el diseño eran los correctos. Con esto se pueden dar unos ajustes finales a los circuitos para finalmente lograr el diseño final tal y como se ha expuesto a lo largo de este capítulo. Lo que no se podía probar directamente por no tener los componentes, como la entrada de la fotocélula y la salida hacia los balastos, se simuló con una fuente de voltaje (limitando la corriente proporcionada) en el primer caso y con una carga similar a la de la salida para el segundo caso.

Las pruebas realizadas han sido satisfactorias para los valores de los componentes que se indican en los diagramas esquemáticos de los **anexos 2 y 3**. Lo único que resta es simular el programa del diseño real, lo que se logrará con un programa de prueba para la transmisión y un programa de prueba para la recepción, en los cuales se utilizan los mismos componentes que los considerados en el diseño como son los pulsadores con sus leds y el potenciómetro.



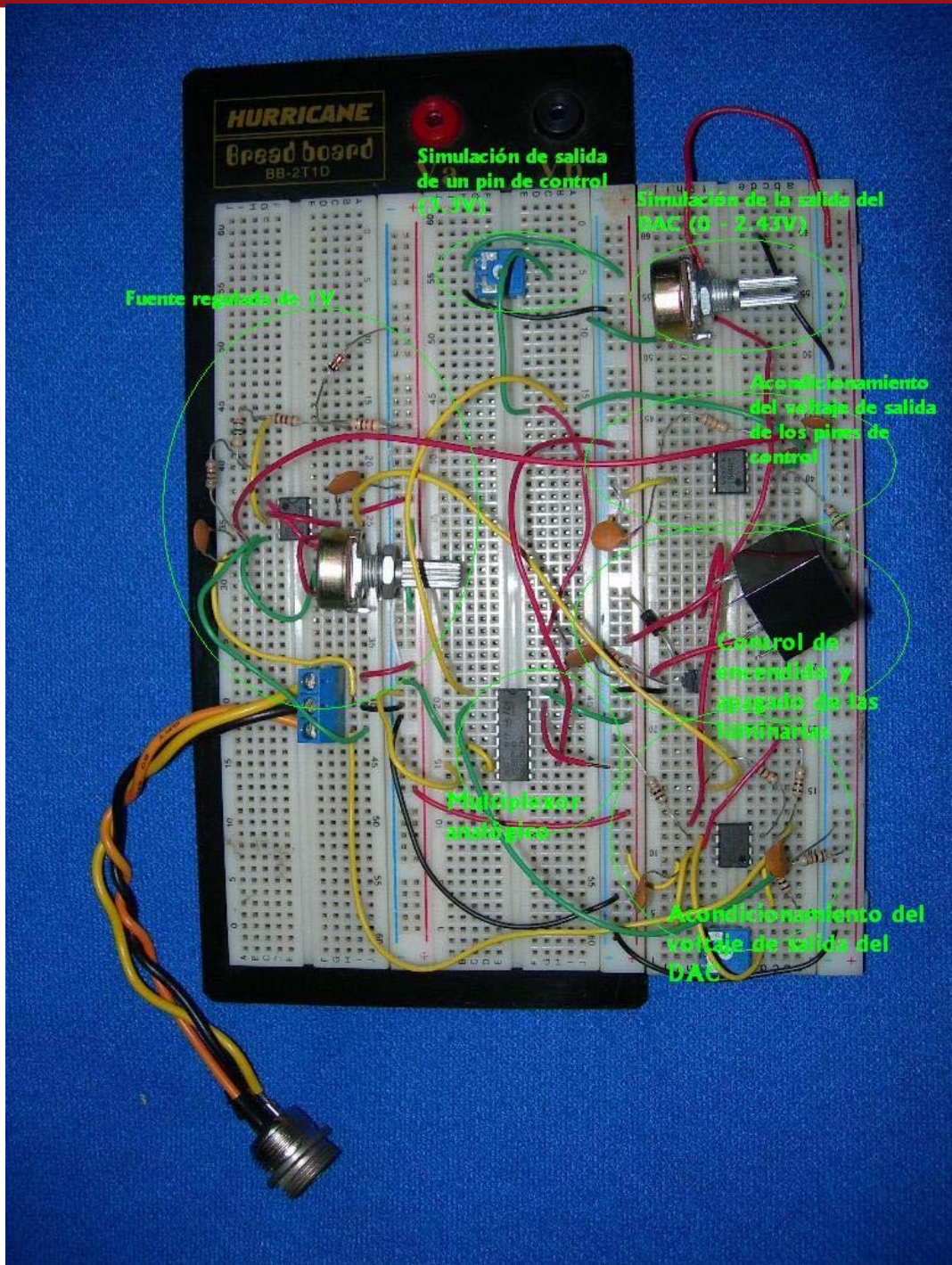


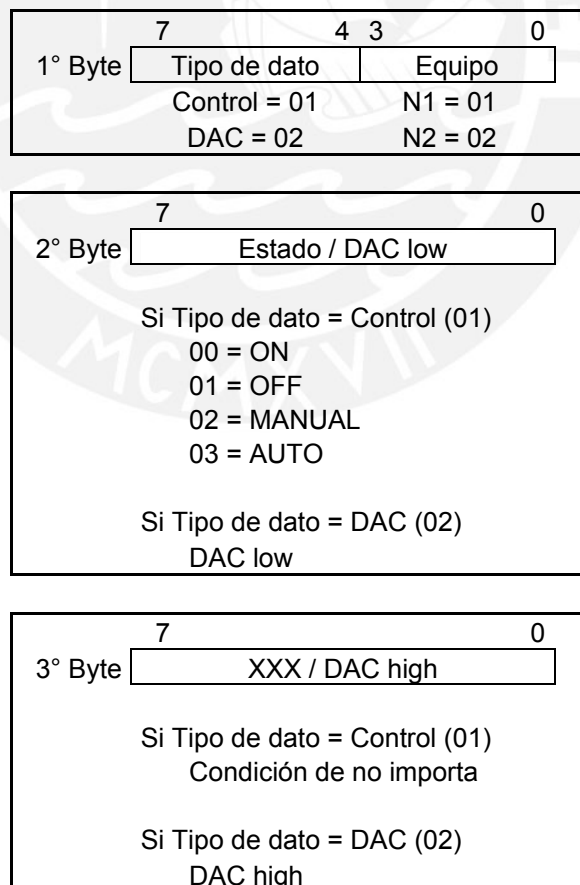
Figura 20: Circuito de prueba

#### 4.5.1. Lógica del programa

La lógica de interpretación del dato que se envía y recibe utilizando el protocolo Zigbee se describe a continuación y se ilustra en la **Figura 21**.

Siempre se envían tres datos de 8 bits de la siguiente manera:

El primero se divide en dos partes de 4 bits. La parte más significativa identifica el tipo de dato que se está transmitiendo, ya sea de control (para encender o apagar y controlar de manera automática o manual las luminarias) o de DAC (para regular la intensidad de la luz con el potenciómetro). La parte menos significativa es para identificar a qué equipo está dirigido el dato (N1, N2, N3...). No se debe olvidar que como máximo se necesitan 17 interruptores, 3 en la sala de capacitaciones y 14 en el área de cubículos, y como con el diseño del programa actual sólo se pueden direccionar hasta 16 interruptores se tendrían que separar de todas maneras ambas áreas.



**Figura 21: Lógica del programa**

El segundo dato depende del primero. Si se está en tipo de dato de control, aquí se especifica si se quiere apagar, encender, regular de manera manual o regular de manera automática las luminarias. Si se está en tipo de dato DAC, aquí se encuentra la parte menos significativa del valor digital/análogo.

El tercer y último dato también depende del primero. Si se está en tipo de dato de control, el dato es recibido, pero no se hace nada con él. Si se está en tipo de dato DAC, aquí se encuentra la parte más significativa del valor digital/análogo.

#### **4.5.1.1. Programa de transmisión**

Este programa se encarga de transmitir la trama de datos, dependiendo del tipo de estado seleccionado (ON, OFF, MANUAL o AUTOMATICO). Primero se barre el estado de los pulsadores de selección y dependiendo de la lógica se envía la trama correspondiente al pulsador seleccionado.

La lógica del funcionamiento del programa es la siguiente: Para seleccionar un equipo remoto al cual se quiere enviar un tipo de control (ON, OFF, MANUAL o AUTOMATICO) se presiona un pulsador que inmediatamente, mediante un LED, indica que está activo dicho equipo. Luego se procede a seleccionar el tipo de control que se desee, teniendo en cuenta que para seleccionar el tipo de control MANUAL o AUTOMATICO, el equipo seleccionado debe estar también encendido (ON). Si el equipo está seleccionado, pero no en ON, no le hará caso a la instrucción

AUTO o MANUAL. Por lo tanto, debe estar habilitado el estado ON del equipo remoto para que si se selecciona el modo automático (LED indicador A/M apagado), realmente se empiece a controlar la iluminación de manera automática y si se selecciona modo manual (LED indicador A/M prendido), se empiece a leer el estado del potenciómetro para transmitirlo y que la regulación manual se realice de manera correcta. Por otro lado, si se procede a mandar la señal de apagado del equipo remoto, OFF (LED apagado) se deshabilitará el estado MANUAL o AUTOMATICO que se haya seleccionado anteriormente. Además si se selecciona nuevamente un equipo que ya estaba seleccionado, este se deseleccionará y el LED indicador que le corresponde se apagará de inmediato (**Ver Figuras 22a y 22b**).

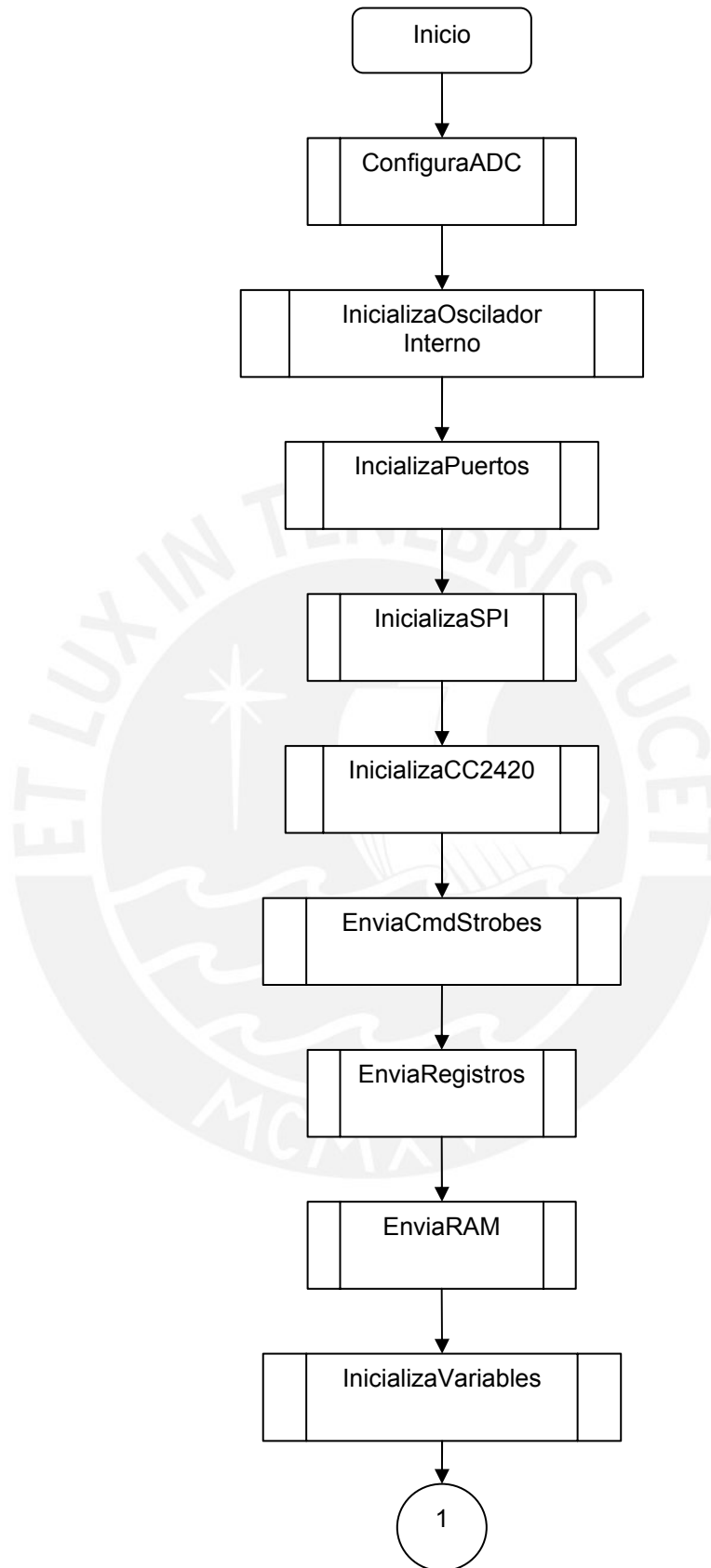


Figura 22a: Diagrama de bloques del programa de transmisión

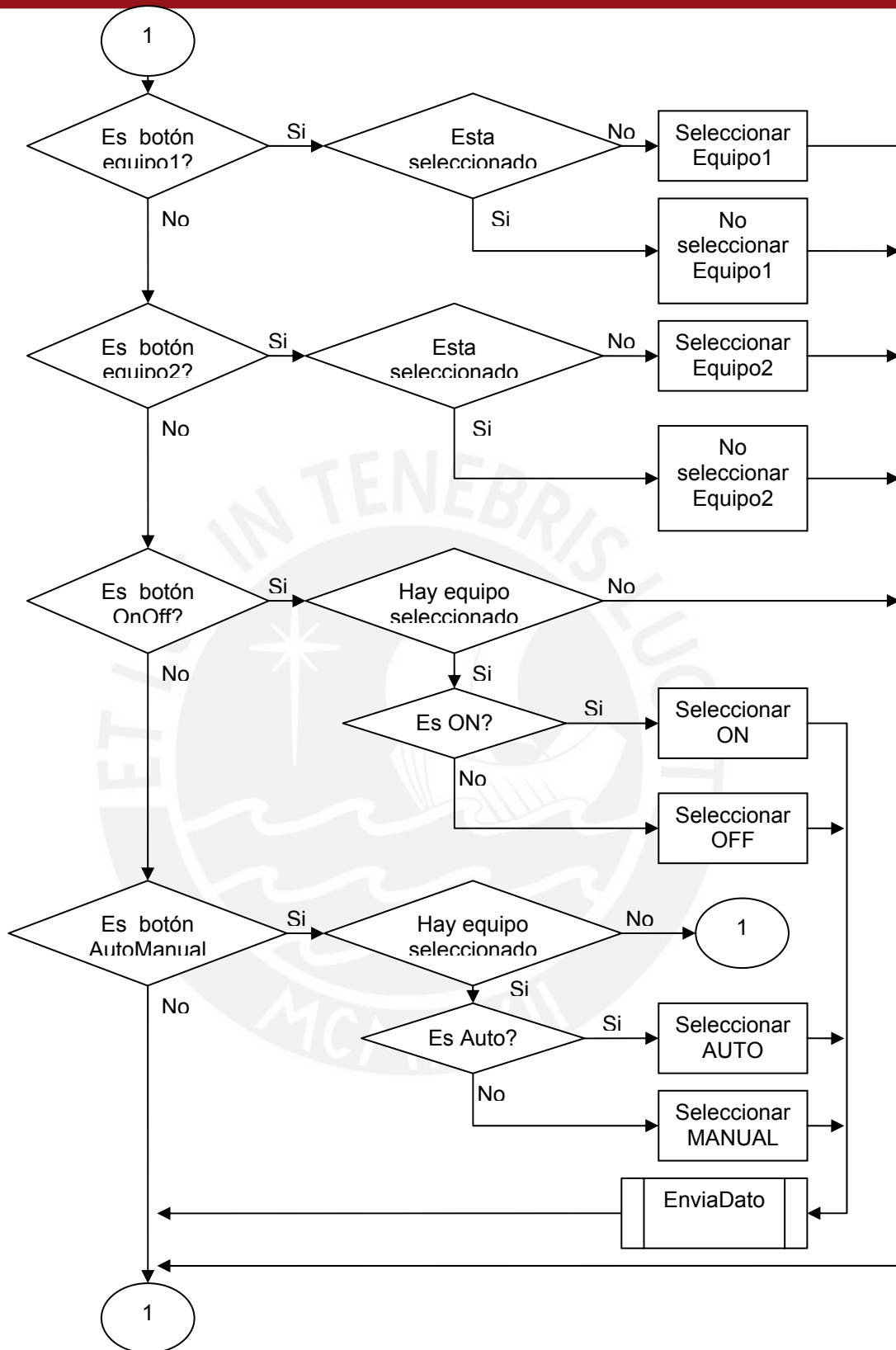


Figura 22b: Diagrama de bloques del programa de transmisión

#### 4.5.1.2. Programa de recepción

Este programa se encarga de recibir los datos enviados por el control remoto (transmisor) y accionar las salidas de control determinadas.

El programa espera recibir un paquete y una vez que ya se tiene la trama, se procede a sacar de esta el dato enviado, el cual es verificado para que cumpla con el protocolo establecido (**ver Figuras 8 y 21**). Una vez validado el dato, se procede a determinar a que equipo va dirigido, luego se revisa la señal de control para diferenciar si es para encender o apagar el sistema o para funcionar en modo automático o manual y regular la intensidad de la luz de manera adecuada (**Ver Figuras 23a y 23b**).

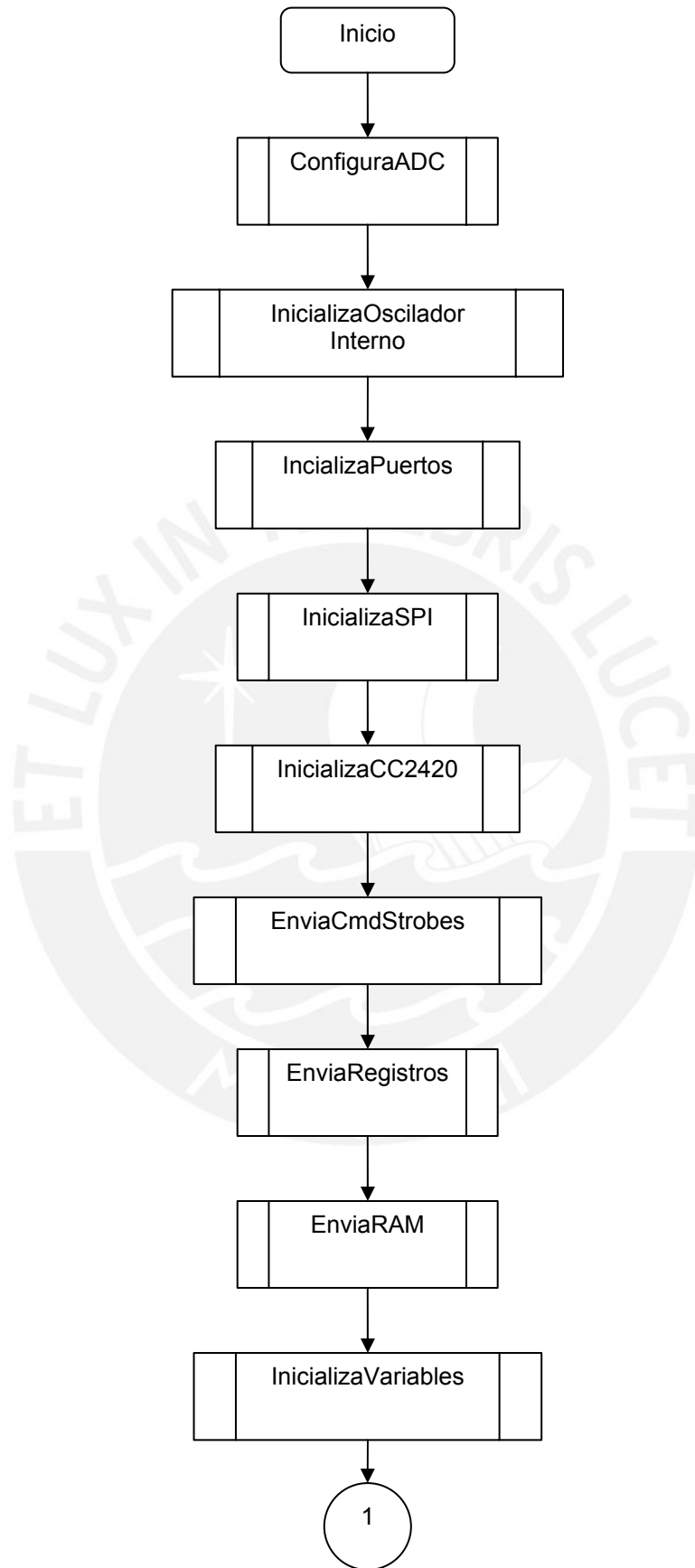


Figura 23a: Diagrama de bloques del programa de recepción



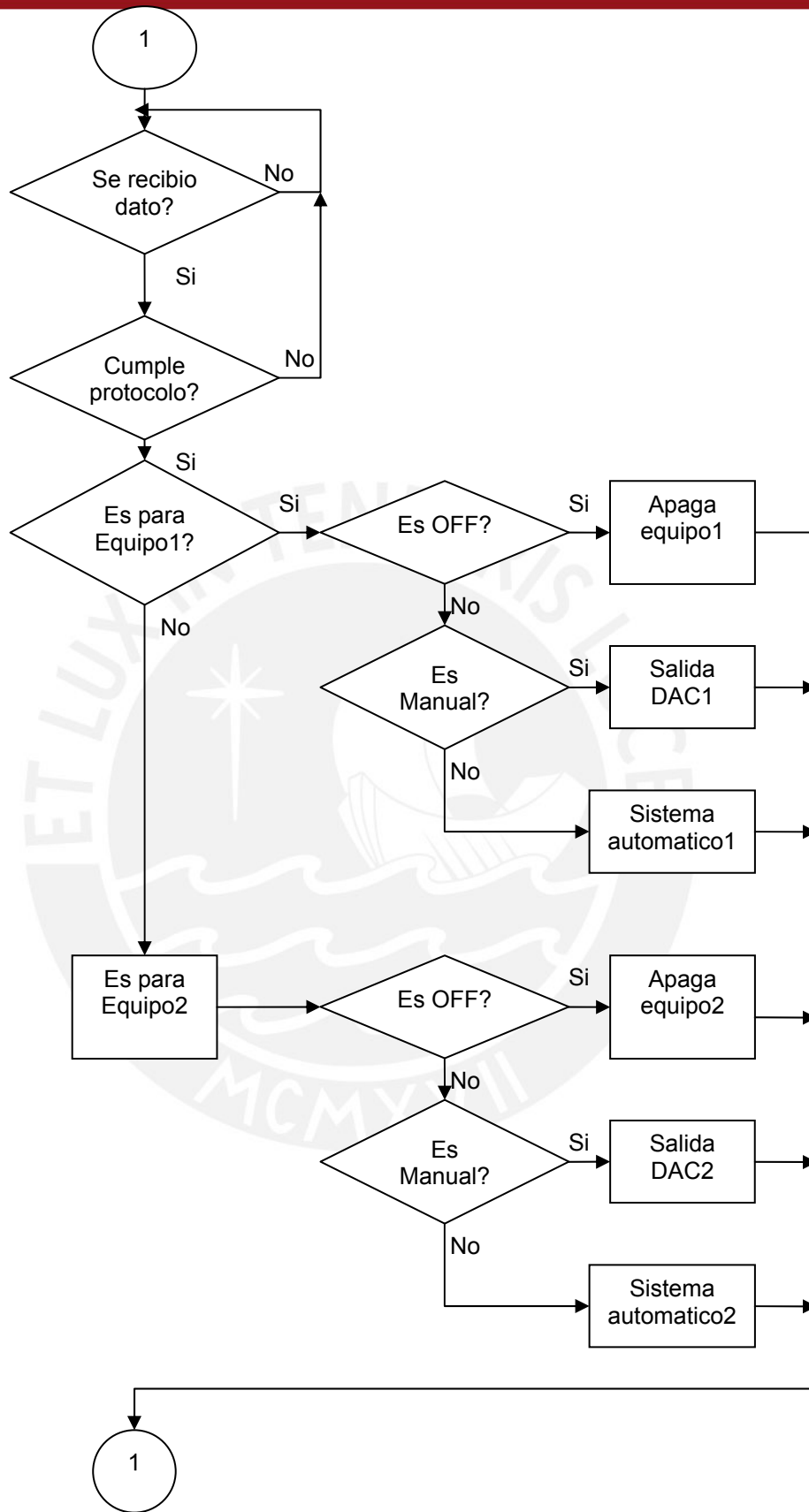


Figura 23b: Diagrama de bloques del programa de recepción

#### 4.5.2. Pruebas de los programas

Para verificar el correcto funcionamiento de los programas, se utilizó el USB Debug Adapter (que sirve para simular programas estando conectado al hardware en tiempo real) y el programa Silicon Laboratories IDE, ambos incluidos en el módulo de desarrollo Zigbee de 2.4 GHz. de Silicon Labs.

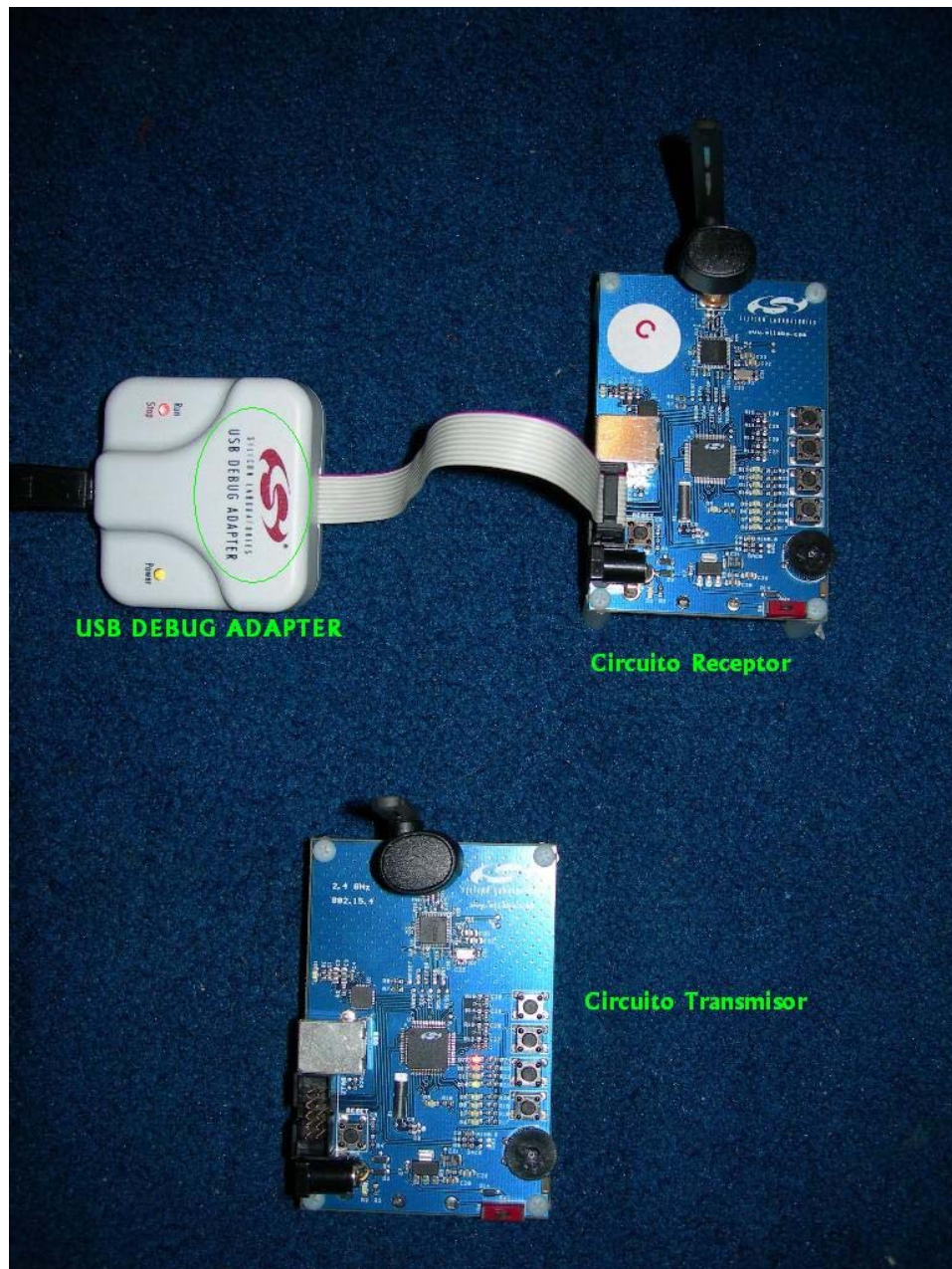
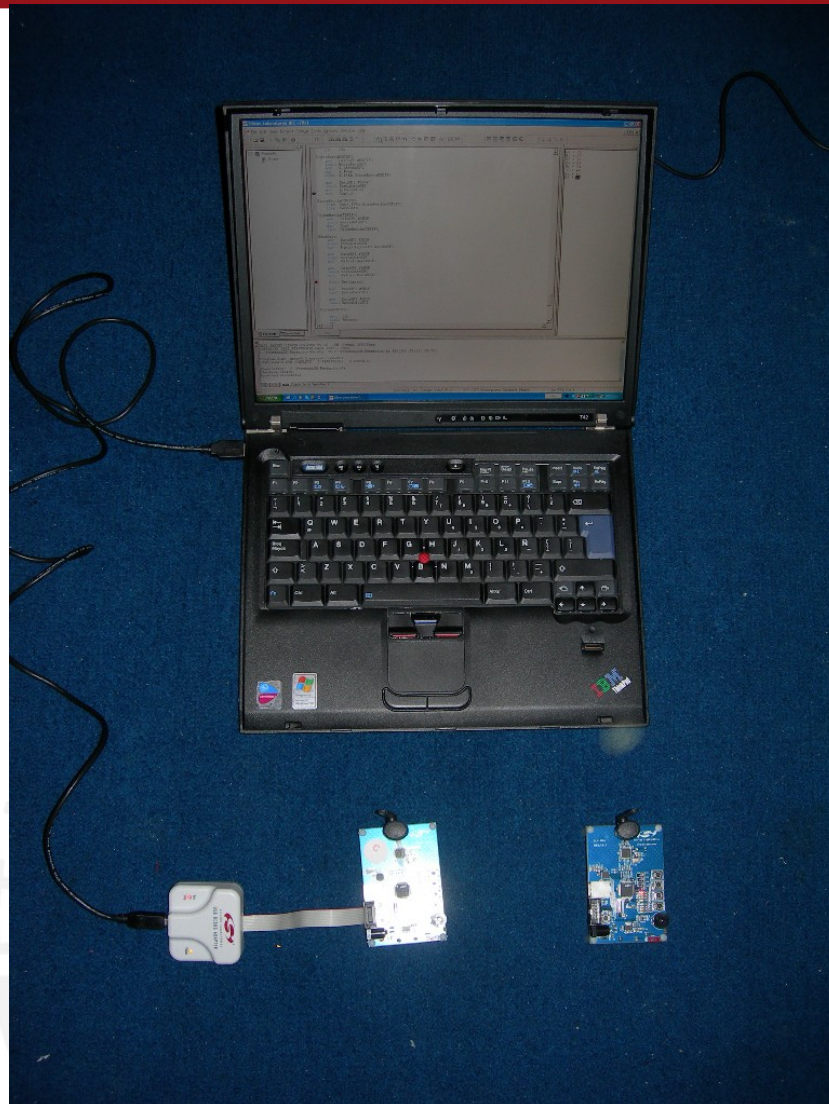
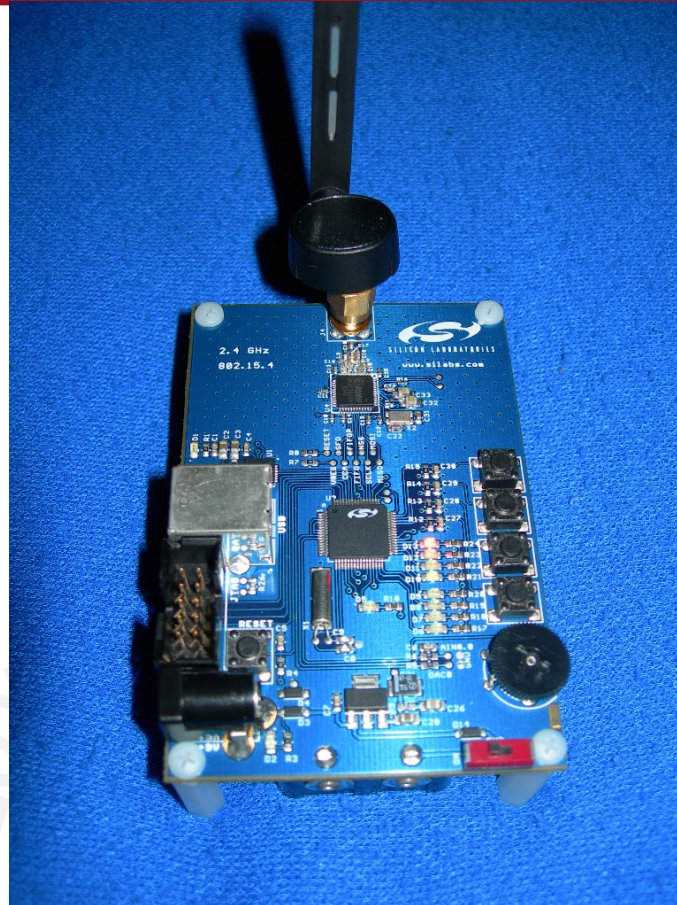


Figura 24: USB Debug Adapter



**Figura 25: Prueba del programa de transmisión-recepción ZigBee**

Si bien los programas son un poco extensos y la inicialización de los dispositivos es algo compleja, se mostrarán los resultados de la prueba de encendido del balasto N1. Para esta prueba, el circuito receptor estará conectado al USB Debug Adapter y se colocarán breakpoints adecuados para observar la variación de los valores en los registros que corroborarán el correcto funcionamiento de ambos programas (**Ver Figuras 24 y 25**).



**Figura 26: Circuito de prueba de transmisión-recepción**

Según la **Figura 8** se debe enviar el *Frame Length*, que para el caso de este sistema será de 14 bytes (0E en hexadecimal), y es este valor quien le indica al receptor hasta cuántos bytes va a leer antes de que lleguen los dos bytes del *Frame Check Sequence (FCS)*. Para este caso se utilizan dos bytes de *Frame Control Field (FCF)*, uno de Data Sequence Number, dos de Destination PAN ID, dos de Destination Address, dos de Source Address, tres de datos y los dos del FCS ya mencionados:

$$2 + 1 + 2 + 2 + 2 + 3 + 2 = 14 = 0E \text{ (hexadecimales).}$$

Como los datos que me interesan son justamente los tres bytes de datos, que empiezan 5 bytes antes que la trama que se recibe concluya, el algoritmo para la decodificación del dato recibido cuenta cuantos bytes se han ido recibiendo, empezando en trece y decrementando el valor hasta llegar a cinco y en ese momento empieza a decodificar la información recibida. El valor de la cuenta estará almacenado en el registro R3 y el valor que se está recibiendo se almacena en el registro R2.

Luego, según la **Figura 21**, el encendido del balasto N1 hace que el primer byte tome el valor de 11 hexadecimal porque la parte más significativa (Tipo de dato) es de control y no de DAC, por lo tanto su valor es 1 y la parte menos significativa (Equipo) es de N1 y no de N2, por lo tanto su valor también es 1. El segundo byte tomará el valor de Estado puesto que el Tipo de dato fue de control y su valor será 0 porque la instrucción es un ON. Finalmente, el tercerbyte es recibido, pero no se hace nada con él porque el Tipo de dato no fue DAC y el tercer byte corresponde a la parte alta del valor del DAC que en este momento no está en uso.

```

//***** Stop (F4) *****
LeeRXFIFO:
    clr    CSn

SigueEnviaRXFIFO:
    mov   DatoSPI, #RXFIFO
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    anl   A, #46h
    cjne  A, #46h, SigueEnviaRXFIFO

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    mov   Cont, A

SigueRecibeTXFIFO:
    cjne  Cont, #05h, SigueRecibeTXFIFO1
    ljmp  SacaData

SigueRecibeTXFIFO1:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    dec   Cont
    ljmp  SigueRecibeTXFIFO

SacaData:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   EquipoTipoDato, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor1, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor2, DatoRxSPI

    lcall SetControl

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI

FinLeeRXFIFO:
    seth  CSn
    
```

R0 = 00  
R1 = 00  
R2 = 3E  
R3 = 08  
R4 = 74  
R5 = 44  
R6 = 4C  
R7 = 00

Figura 27: Esperando órdenes

```

//*****
LeeRXFIFO:
    clr    CSn

SigueEnviaRXFIFO:
    mov   DatoSPI, #RXFIFO
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    anl   A, #46h
    cjne  A, #46h, SigueEnviaRXFIFO

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    mov   Cont, A

SigueRecibeTXFIFO:
    cjne  Cont, #05h, SigueRecibeTXFIFO1
    ljmp  SacaData

SigueRecibeTXFIFO1:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    dec   Cont
    ljmp  SigueRecibeTXFIFO

SacaData:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   EquipoTipoDato, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor1, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor2, DatoRxSPI

    lcall SetControl

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI

FinLeeRXFIFO:
    
```

R0 = FF  
R1 = 00  
R2 = 0E  
R3 = 08  
R4 = 74  
R5 = 44  
R6 = 4C  
R7 = 00

Figura 28: Órdenes recibidas - Valor de la cuenta listo para cargarse

```

//*****
LeeRXFIFO:
  clr  CSn

SigueEnviaRXFIFO:
  mov  DatoSPI, #RXFIFO
  lcall EnviaDatoSPI
  mov  A, DatoRxSPI
  anl  A, #46h
  cjne A, #46h, SigueEnviaRXFIFO

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  A, DatoRxSPI
  mov  Cont, A

SigueRecibeTXFIFO:
  cjne Cont, #05h, SigueRecibeTXFIFO1
  ljmp SacaData

SigueRecibeTXFIFO1:
  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  dec  Cont
  ljmp SigueRecibeTXFIFO

SacaData:
  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  EquipoTipoDato, DatoRxSPI

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  Valor1, DatoRxSPI

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  Valor2, DatoRxSPI

  lcall SetControl

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  
```

R0	=	FF
R1	=	00
R2	=	61
R3	=	0E
R4	=	74
R5	=	44
R6	=	4C
R7	=	00

Figura 29: Primer byte recibido - La cuenta va a decrementarse

```

//*****
LeeRXFIFO:
  clr  CSn

SigueEnviaRXFIFO:
  mov  DatoSPI, #RXFIFO
  lcall EnviaDatoSPI
  mov  A, DatoRxSPI
  anl  A, #46h
  cjne A, #46h, SigueEnviaRXFIFO

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  A, DatoRxSPI
  mov  Cont, A

SigueRecibeTXFIFO:
  cjne Cont, #05h, SigueRecibeTXFIFO1
  ljmp SacaData

SigueRecibeTXFIFO1:
  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  dec  Cont
  ljmp SigueRecibeTXFIFO

SacaData:
  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  EquipoTipoDato, DatoRxSPI

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  Valor1, DatoRxSPI

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  mov  Valor2, DatoRxSPI

  lcall SetControl

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI

  mov  DatoSPI, #SNOP
  lcall EnviaDatoSPI
  
```

R0	=	FF
R1	=	00
R2	=	89
R3	=	0D
R4	=	74
R5	=	44
R6	=	4C
R7	=	00

Figura 30: Segundo byte recibido – La cuenta va a decrementarse

```

//*****Go (F5)*****
LeeRXFIFO:
    clr    CSn

SigueEnviaRXFIFO:
    mov   DatoSPI,#RXFIFO
    lcall EnviaDatoSPI
    mov   A,DatoRxSPI
    anl  A,#46h
    cjne A,#46h,SigueEnviaRXFIFO

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   A,DatoRxSPI
    mov   Cont,A

SigueRecibeTXFIFO:
    cjne Cont,#05h,SigueRecibeTXFIFO1
    ljmp  SacaData

SigueRecibeTXFIFO1:
    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    dec   Cont
    ljmp  SigueRecibeTXFIFO

SacaData:
    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   EquipoTipoDato,DatoRxSPI

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   Valor1,DatoRxSPI

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   Valor2,DatoRxSPI

    lcall SetControl

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    
```

R0 = FF  
R1 = 00  
R2 = 01  
R3 = 0C  
R4 = 74  
R5 = 44  
R6 = 4C  
R7 = 00

Figura 31: Tercer byte recibido y así sucesivamente

```

//*****Go (F5)*****
LeeRXFIFO:
    clr    CSn

SigueEnviaRXFIFO:
    mov   DatoSPI,#RXFIFO
    lcall EnviaDatoSPI
    mov   A,DatoRxSPI
    anl  A,#46h
    cjne A,#46h,SigueEnviaRXFIFO

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   A,DatoRxSPI
    mov   Cont,A

SigueRecibeTXFIFO:
    cjne Cont,#05h,SigueRecibeTXFIFO1
    ljmp  SacaData

SigueRecibeTXFIFO1:
    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    dec   Cont
    ljmp  SigueRecibeTXFIFO

SacaData:
    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   EquipoTipoDato,DatoRxSPI

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   Valor1,DatoRxSPI

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    mov   Valor2,DatoRxSPI

    lcall SetControl

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI

    mov   DatoSPI,#SNOP
    lcall EnviaDatoSPI
    
```

R0 = FF  
R1 = 00  
R2 = 11  
R3 = 05  
R4 = 74  
R5 = 44  
R6 = 4C  
R7 = 00

Figura 34: Décimo byte recibido – Primer dato de control recibido



```

//*****
LeeRXFIFO:
    clr    CSn

SigueEnviaRXFIFO:
    mov   DatoSPI, #RXFIFO
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    anl  A, #46h
    cjne A, #46h, SigueEnviaRXFIFO

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    mov   Cont, A

SigueRecibeTXFIFO:
    cjne Cont, #05h, SigueRecibeTXFIFO1
    ljmp  SacaData

SigueRecibeTXFIFO1:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    dec   Cont
    ljmp  SigueRecibeTXFIFO

SacaData:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   EquipoTipoDato, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor1, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor2, DatoRxSPI

    lcall SetControl

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    
```

R0	=	FF
R1	=	00
R2	=	00
R3	=	05
R4	=	74
R5	=	44
R6	=	4C
R7	=	00

Figura 35: Undécimo byte recibido – Segundo dato de control recibido

```

igueEnviaRXFIFO:
    mov   DatoSPI, #RXFIFO
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    anl  A, #46h
    cjne A, #46h, SigueEnviaRXFIFO

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   A, DatoRxSPI
    mov   Cont, A

igueRecibeTXFIFO:
    cjne Cont, #05h, SigueRecibeTXFIFO1
    ljmp  SacaData

igueRecibeTXFIFO1:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    dec   Cont
    ljmp  SigueRecibeTXFIFO

acaDato:
    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   EquipoTipoDato, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor1, DatoRxSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    mov   Valor2, DatoRxSPI

    lcall SetControl

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI

    mov   DatoSPI, #SNOP
    lcall EnviaDatoSPI
    
```

R0	=	FF
R1	=	00
R2	=	30
R3	=	05
R4	=	74
R5	=	44
R6	=	4C
R7	=	00

Figura 36: Duodécimo byte recibido – Último byte de control recibido

Observar que el valor de la cuenta en R3 ya no se decrementa a partir de 5 porque sólo se usaba esta cuenta para determinar cuando

llegaba el primer byte de control (ver Figuras 34, 35 y 36), además los valores de los datos de control que se observan en R2, coinciden con lo esperado (ver Figuras 34 y 35)

#### 4.6. Presupuesto

A continuación se detalla el presupuesto necesario para la implementación de una tarjeta de transmisión, una tarjeta de recepción y todos los balastos y fotocélulas necesarios:

Componente	Descripción	Cantidad	Precio Unitario (US\$)	Total (US\$)
C8051F121	uC	1	23.740	23.740
CC2420	RF TXRX ZIGBEE 2.4GHZ	1	6.600	6.600
Antena	2.4 GHz 1/4 Wave Whip	1	4.740	4.740
Switch		12	0.300	3.600
Cristal	16 MHz	1	0.530	0.530
LED		11	0.110	1.210
LM2936	Regulador	1	2.650	2.650
Conector ISP		1	0.600	0.600
Condensador	27pF	6	0.036	0.216
	68pF	4	0.030	0.120
	0.01uF	1	0.024	0.024
	0.1uF	24	0.462	11.088
	10uF	3	0.880	2.640
	47uF	1	0.650	0.650
Resistencia	2	2	0.100	0.200
	270	10	0.069	0.690
	1.5K	1	0.072	0.072
	10K	14	0.076	1.064
Inductor	7.5nF	2	0.430	0.860
	5.6nF	1	0.400	0.400
Tarjeta		1	5.600	5.600
			<b>TOTAL</b>	<b>67.30</b>

Tabla 9: Presupuesto para una tarjeta de transmisión

Componente	Descripción	Cantidad	Precio Unitario (US\$)	Total (US\$)
C8051F121	uC	1	23.740	23.740
CC2420	RF TXRX ZIGBEE 2.4GHZ	1	6.600	6.600
Antena	2.4 GHz 1/4 Wave Whip	1	4.740	4.740
Cristal	16 MHz	1	0.530	0.530
LM2936	Regulador	1	2.650	2.650
7812	Regulador	1	0.620	0.620
7912	Regulador	1	0.620	0.620
TL082	OPAM	3	0.310	0.930
TL081	OPAM	1	0.310	0.310
2N3904	Transistor	2	0.156	0.312
1N4004	Diodo	6	0.093	0.558
4053BC	Mux Analógico	1	0.780	0.780
Relé	JQX-3F	2	1.100	2.200
Bornera	x2	2	0.310	0.620
Bornera	x3	3	0.310	0.930
Potenciómetro	100K	1	0.310	0.310
Transformador	(12-0-12)@2A	1	6.250	6.250
Conector ISP		1	0.600	0.600
Condensador	27pF	6	0.031	0.186
	68pF	4	0.031	0.124
	100nF	9	0.031	0.279
	0.01uF	1	0.031	0.031
	0.1uF	14	0.031	0.434
	1uF	1	0.031	0.031
	10uF	3	0.031	0.093
	47uF	1	0.031	0.031
Resistencia	3300uF	2	0.94	1.880
	2	2	0.015	0.030
	270	10	0.015	0.150
	1.5K	1	0.015	0.015
Inductor	10K	14	0.015	0.210
	7.5nF	2	0.430	0.860
Tarjeta	5.6nF	1	0.400	0.400
		1	11.500	11.500
			<b>TOTAL</b>	<b>69.60</b>

Tabla 10: Presupuesto para una tarjeta de recepción

Componente	Descripción	Cantidad	Precio Unitario (US\$)	Total (US\$)
Balasto	HFR-418TLD	59	48.00	2832.00
	HFR-228TL5	19	57.00	1083.00
Fotocélula	Luxsense	10	28.00	280.00
			<b>TOTAL</b>	<b>4195.00</b>

**Tabla 11: Presupuesto de los balastos y fotocélulas necesarias**

Considerando que en total se necesitan tres tarjetas de transmisión, nueve tarjetas de recepción, que el costo por el trabajo de ingeniería ascenderá a \$2 500.00 (dos mil quinientos dólares americanos) y que el costo de mano de obra ascenderá a \$500.00 (quinientos dólares americanos); el total del presupuesto será:

$$67.3 * 3 + 69.6 * 9 + 4195 + 2500 + 500 =$$

$$201.9 + 626.4 + 4195 + 2500 + 500 = \$8 023.30 \text{ (ocho mil veintitrés dólares americanos con treinta centavos).}$$

Además, el promedio mensual del consumo de energía eléctrica en dicha oficina es de aproximadamente \$2 100 (dos mil cien dólares americanos) y junto con la gente de finanzas se ha calculado que aproximadamente el 40% de ese monto es lo que se consume en iluminación. Por lo tanto, considerando que con una adecuada configuración del sistema automatizado se espera generar aproximadamente un 60% de ahorro energético, el tiempo de

recuperación total de la inversión sería como se detalla a continuación:

$$2100 * 0.4 = \$ 840.00 \text{ (Consumo mensual en iluminación)}$$

$$840 * 0.6 = \$ 504.00 \text{ (Ahorro mensual)}$$

$$8023.3 / 504 = 15.92 \text{ (Tiempo de recuperación en meses)}$$

Finalmente, concluimos que el tiempo total para lograr recuperar la inversión será de 16 meses.



## CONCLUSIONES Y RECOMENDACIONES

Las pruebas realizadas para la verificación del alcance entre los módulos de recepción y transmisión fueron exitosas, cumpliendo con los requerimientos según la ubicación de los dispositivos dentro del ambiente donde van a ser distribuidos.

La parte del paquete enviado por el transmisor correspondiente al dato se forma correctamente según el tipo de comando que el usuario selecciona (ON, OFF, AUTMOMATICO o MANUAL), cumpliéndose así con el adecuado protocolo de comunicación. Esto da la seguridad de que no cualquier dispositivo Zigbee pueda comunicarse con los módulos de recepción ya que éstos poseen su propio lenguaje.

Actualmente el CC2420 es el único dispositivo de fácil acceso para aplicaciones Zigbee y por su bajo precio, se recomienda la adaptación del mismo para trabajos con otros microcontroladores que puedan manejar comunicación SPI. En el caso particular de la Pontificia Universidad Católica del Perú, puede ser con el microcontrolador ATmega8 de la familia ATMEL debido a que cuenta con los requerimientos técnicos y los alumnos están muy familiarizados con él.

El microcontrolador empleado (C8051F212) es el único que contaba con dos conversores digital-análogo y con 8 conversores análogo-digital en el mercado. Además puede entregar y recibir mucho más corriente que los 8051 comunes, sin que esto signifique que genera

un desperdicio de energía, por lo que es una buena opción (aunque un poco cara) para aplicaciones Zigbee.

El USB Debug Adapter es una muy buena herramienta para identificar problemas en los programas creados, pero se debe tener cuidado con los tiempos de retardo para que no haya conflictos cuando se quiera simular paso a paso y cuando se quiera simular en tiempo real.

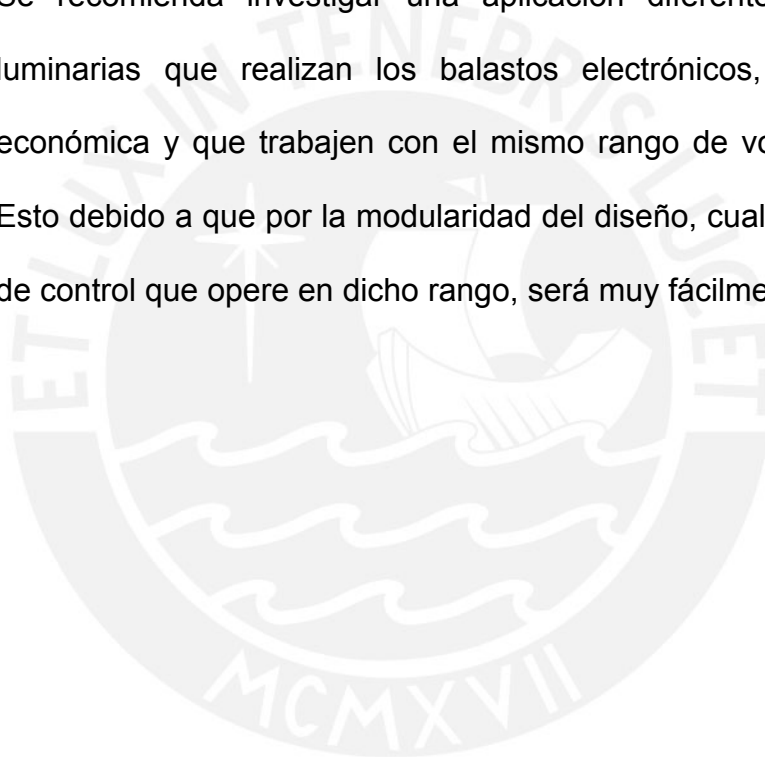
El voltaje a la salida del DAC del microcontrolador es acondicionada correctamente según las pruebas realizadas para todo el rango máximo de voltaje que el DAC del microcontrolador puede dar (0 – 2.43V). Cada etapa cumple con su requerimiento es así que la etapa de amplificación cumple con aumentar el rango de salida del DAC a un rango máximo de 0 – 9V y la etapa de offset cumple con llevar ese rango de voltaje a un rango entre 1 – 10V que es el indicado para el control de los balastos. Todas estas etapas cumpliendo con los niveles de corriente necesarios para la activación de los dispositivos (balastos, relés y pines de control del multiplexor analógico) teniendo en cuenta los niveles de corriente entregados por el microcontrolador.

El presupuesto calculado es elevado y el tiempo de recuperación no es tan corto, sin embargo, cabe destacar que es el precio de los balastos electrónicos regulables es lo que realmente está elevando el presupuesto final de esta manera. Es importante considerar que estos son dispositivos nuevos en el mercado y que su precio esta disminuyendo de una manera rápida por lo que este diseño podrá ser

implementado en un menor costo en un tiempo relativamente corto y ser igualmente moderno.

Se recomienda investigar y desarrollar diversas aplicaciones para adaptarlas al modulo de recepción con el fin de poder controlar otros dispositivos. Por ejemplo control de persianas, sistema de aire acondicionado, etc.

Se recomienda investigar una aplicación diferente al control de luminarias que realizan los balastos electrónicos, que sea más económica y que trabajen con el mismo rango de voltaje (1 – 10V). Esto debido a que por la modularidad del diseño, cualquier dispositivo de control que opere en dicho rango, será muy fácilmente adaptado.





## FUENTES

### Bibliografía

- [1] IEEE  
1997 “Engineering Manual of Automatic Controls”  
EE.UU.
- [3] FERNÁNDEZ VALDIVIESO, C.  
1999 “La domótica: Esencia de un edificio inteligente”.  
Madrid
- [5] GAITÁN, Gabriel  
2006 “Volviendo Inteligente nuestro hogar”.  
España
- [8] NAVARRO, María.  
2004 “ZigBee: Nuevo estándar de tecnología inalámbrica”.  
España
- [9] ATKIN, B.  
1988 “Intelligent Buildings: Applications of IT and Building Automation  
to High Technology Building Projects”.  
EE.UU.
- [10] DRITSAS, David  
2005 “Beyond X-10: The Future of Home Automation”.  
Philadelphia
- [11] BICSI  
2003 “Telecommunications and Distribution Methods Manual” 10<sup>th</sup>  
Edition  
Tampa
- [12] IEEE  
1991 “Electric Power Systems in Commercial Buildings”.  
EE.UU.
- [14] RUIZ CELMA, Antonio; ROJAS RODRIGUEZ, Sebastián.  
1998 “Instalaciones de Iluminación”  
Madrid
- [15] GUASCH FARRÁS, Juan  
2001 “Enciclopedia de salud y seguridad en el trabajo”  
España

## Documentos

- [2] ZIGBEE ALLIANCE  
2004 “ZigBee Specification” Versión 1.0  
<http://www.zigbee.org/en/index.asp>
- [4] BANDA, Marco  
2004 “CD: Sistemas Eléctricos”.  
Lima
- [6] WIKIPEDIA:  
<http://en.wikipedia.org/wiki/Domotics>  
<http://en.wikipedia.org/wiki/Zigbee>  
<http://es.wikipedia.org/wiki/X10>
- [7] IEEE 802.15.4:  
<http://www.ieee802.org/15/pub/TG4.html>
- [13] ZIGBEE, A TECHNICAL OVERVIEW OF WIRELESS TECHNOLOGY  
<http://zigbee.hasse.nl/>
- [16] Energuía  
<http://www.energuia.com/es/reportajes.aspx>
- [17] OSRAM  
<http://catalog.myosram.com/>
- [18] General Electric  
<http://genet.gelighting.com/LightProducts>
- [19] Especialidades Luminotécnicas  
<http://www.elt.es/home/princi.htm>
- [20] Magnetek  
[www.magnetek.hu/Light.pdf](http://www.magnetek.hu/Light.pdf)

# ANEXO N°1

## PLANO DE LUMINARIAS

# ANEXO N°2

## DIAGRAMA ESQUEMÁTICO DEL CIRCUITO TRANSMISOR

# ANEXO N°3

## DIAGRAMA ESQUEMÁTICO DEL CIRCUITO RECEPTOR

# ANEXO N°4

## HOJAS DE DATOS DEL TRANSMISOR-RECEPTOR CC2420 DE CHIPCON

# ANEXO N°5

## HOJAS DE DATOS DEL MICROCONTROLADOR C8051F212


**SmartRF® CC2420**

#### 4-wire Serial Configuration and Data Interface

**CC2420** is configured via a simple 4-wire SPI-compatible interface (pins  $SI$ ,  $SO$ ,  $SCLK$  and  $CS_n$ ) where **CC2420** is the slave. This interface is also used to read and write buffered data (see page 37). All address and data transfer on the SPI interface is done most significant bit first.

##### Register access

There are 33 16-bit configuration and status registers, 15 command strobe registers, and two 8-bit registers to access the separate transmit and receive FIFOs. Each of the 50 registers is addressed by a 6-bit address. The RAM/Register bit (bit 7) must be cleared for register access. The Read/Write bit (bit 6) selects a read or a write operation and makes up the 8-bit address field together with the 6-bit address.

In each register read or write cycle, 24 bits are sent on the  $SI$ -line. The  $CS_n$  pin (Chip Select, active low) must be kept low during this transfer. The bit to be sent first is the RAM/Register bit (set to 0 for register access), followed by the R/W bit (0 for write, 1 for read). The following 6 bits are the address-bits ( $A5:0$ ).  $A5$  is the most significant bit of the address and is sent first. The 16 data-bits are then transferred ( $D15:0$ ), also MSB first. See Figure 8 for an illustration.

The configuration registers can also be read by the microcontroller via the same configuration interface. The R/W bit must be set high to initiate the data read-back. **CC2420** then returns the data from the addressed register on the 16 clock cycles following the register address. The  $SO$  pin is used as the data output and must be configured as an input by the microcontroller.

The timing for the programming is also shown in Figure 8 with reference to Table 4. The clocking of the data on  $SI$  into the **CC2420** is done on the positive edge of  $SCLK$ . When the last bit,  $D0$ , of the 16 data-bits has been written, the data word is loaded in the internal configuration register.

Multiple registers may be written without releasing  $CS_n$ , as described in the Multiple SPI access section on page 29.

The register data will be retained during power down mode, but not when the power-supply is turned off (e.g. by disabling the voltage regulator using the  $VREG\_EN$  pin). The registers can be programmed in any order.



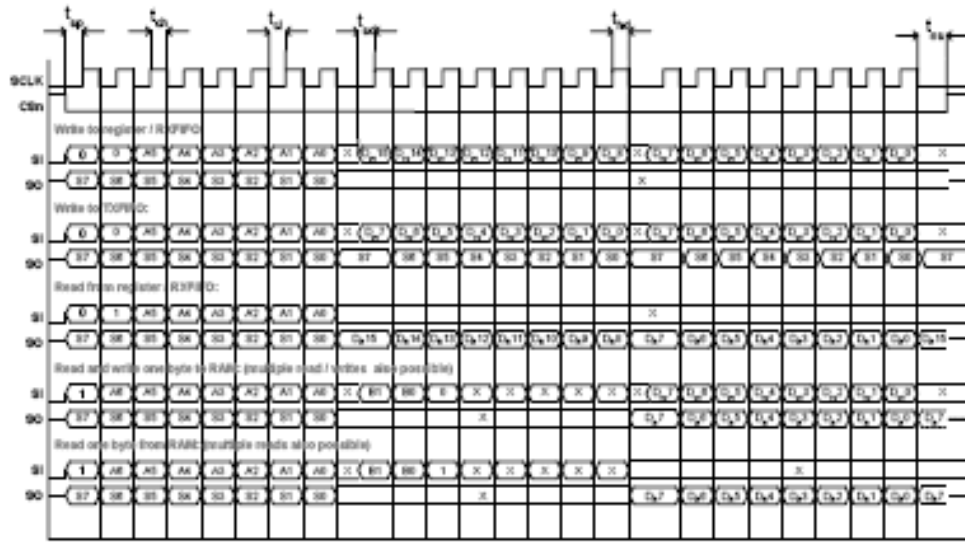


Figure 8. SPI timing diagram

Parameter	Symbol	Min	Max	Units	Conditions
SCLK, clock frequency	$F_{SCLK}$		10	MHz	
SCLK low pulse duration	$t_{L}$	25		ns	The minimum time SCLK must be low.
SCLK high pulse duration	$t_{H}$	25		ns	The minimum time SCLK must be high.
CSn setup time	$t_{sp}$	25		ns	The minimum time CSn must be low before the first positive edge of SCLK.
CSn hold time	$t_{sh}$	25		ns	The minimum time CSn must be held low after the last negative edge of SCLK.
SI setup time	$t_{sd}$	25		ns	The minimum time data on SI must be ready before the positive edge of SCLK.
SI hold time	$t_{sh}$	25		ns	The minimum time data must be held at SI, after the positive edge of SCLK.
Rise time	$t_{rise}$		100	ns	The maximum rise time for SCLK and CSn
Fall time	$t_{fall}$		100	ns	The maximum fall time for SCLK and CSn

Note: The set-up- and hold-times refer to 50% of VDD.

Table 4. SPI timing specification

### Status byte

During transfer of the register access byte, command strobes, the first RAM address byte and data transfer to the TXFIFO, the **CC2420** status byte is returned on the **SO**

pin. The status byte contains 8 status bits which are described in Table 5.

Issuing a **SNOP** (no operation) command strobe may be used to read the status byte. It may also be read during access to


**SmartRF® CC2420**

chip functions such as register or FIFO access.

Bit #	Name	Description
7	-	Reserved, ignore value
6	XOSC16M_STABLE	Indicates whether the 16 MHz oscillator is running or not 0: The 16 MHz crystal oscillator is not running 1: The 16 MHz crystal oscillator is running
5	TX_UNDERFLOW	Indicates whether an FIFO underflow has occurred during transmission. Must be cleared manually with a SFLUSHTX command strobe. 0: No underflow has occurred 1: An underflow has occurred
4	ENC_BUSY	Indicates whether the encryption module is busy 0: Encryption module is idle 1: Encryption module is busy
3	TX_ACTIVE	Indicates whether RF transmission is active 0: RF Transmission is idle 1: RF Transmission is active
2	LOCK	Indicates whether the frequency synthesizer PLL is in lock or not 0: The PLL is out of lock 1: The PLL is in lock
1	RSSI_VALID	Indicates whether the RSSI value is valid or not. 0: The RSSI value is not valid 1: The RSSI value is valid, always true when reception has been enabled at least 8 symbol periods (128 us)
0	-	Reserved, ignore value

Table 5. Status byte returned during address transfer and TXFIFO writing

### Command strobes

Command strobes may be viewed as single byte instructions to **CC2420**. By addressing a command strobe register internal sequences will be started. These commands must be used to enable the crystal oscillator, enable receive mode, start decryption etc. All 15 command strobes are listed in Table 11 on page 61.

When the crystal oscillator is disabled (Power Down state in Figure 24 on page 43), only the SXOSCON command strobe may be used. All other command strobes will be ignored and will have no effect. The crystal oscillator must stabilise (see the XOSC16M\_STABLE status bit in Table 5) before other command strobes are accepted.

The command strobe register is accessed in the same way as for a register write operation, but no data is transferred. That is, only the RAM/Register bit (set to 0),

R/W bit (set to 0) and the 6 address bits (in the range 0x00 through 0x0E) are written. A command strobe may be followed by any other SPI access without pulling CSn high, and is executed on the last falling edge on SCLK.

### RAM access

The internal 388 byte RAM may be accessed through the SPI interface. Single or multiple bytes may be read or written sending the address part (2 bytes) only once. The address is then automatically incremented by the **CC2420** hardware for each new byte. Data is read and written one byte at a time, unlike register access where 2 bytes are always required after each address byte.

The crystal oscillator must be running when accessing the RAM.

The RAM/Register bit must be set high to enable RAM access. The 9 bit RAM



# SmartRF<sup>®</sup> CC2420

address consists of two parts, B1:0 (MSB) selecting one of the three memory banks and A6:0 (LSB) selecting the address within the selected bank. The RAM is divided into three memory banks: TXFIFO (bank 0), RXFIFO (bank 1) and security (bank 2). The FIFO banks are 128 bytes each, while the security bank is 112 bytes.

A6:0 is transmitted directly after the RAM/Register bit as shown in Figure 8. For RAM access, a second byte is also required before the data transfer. This byte contains B1:0 in bits 7 and 6, followed by the R/W bit (0 for read+write, 1 for read). Bits 4 through 0 are don't care as shown in Figure 8.

For RAM write, data to be written must be input on the SI pin directly after the second address byte. RAM data read is

output on the SO pin simultaneously, but may be ignored by the user if only writing is of interest.

For RAM read, the selected byte(s) are output on the SO pin directly after the second address byte.

See Figure 9 for an illustration on how multiple RAM bytes may be read or written in one operation.

The RAM memory space is shown in Table 6.

As with register data, data stored in RAM will be retained during power down mode, but not when the power-supply is turned off (e.g. by disabling the voltage regulator using the VREG\_EN pin).

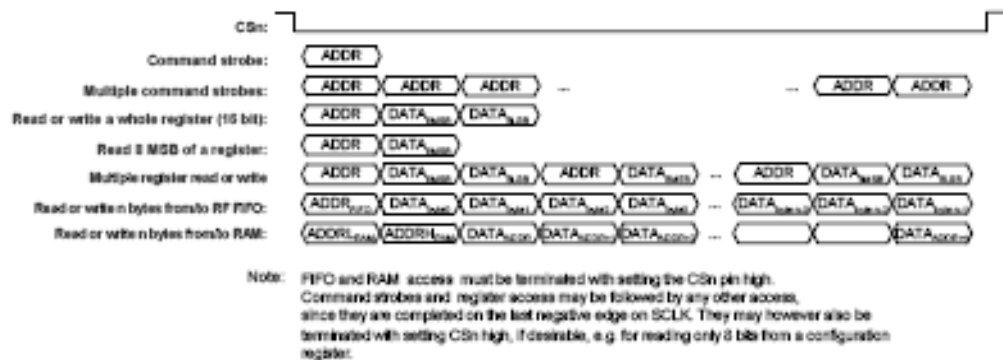


Figure 9. Configuration registers write and read operations via SPI


**SmartRF® CC2420**

Address	Byte Ordering	Name	Description
0x16F – 0x16C	-	-	Not used
0x16B – 0x16A	MSB LSB	SHORTADR	16-bit Short address, used for address recognition.
0x169 – 0x168	MSB LSB	PANID	16-bit PAN Identifier, used for address recognition.
0x167 – 0x160	MSB LSB	IEEEADR	64-bit IEEE address of current node, used for address recognition.
0x15F – 0x150	MSB LSB	CBCSTATE	Temporary storage for CBC-MAC calculations
0x14F – 0x140	MSB (Flags) LSB	TXNONCE / TXCTR	Transmitter nonce for in-line authentication and transmitter counter for in-line encryption.
0x13F – 0x130	MSB LSB	KEY1	Encryption key 1
0x12F – 0x120	MSB LSB	SABUF	Stand-alone encryption buffer, for plaintext input and ciphertext output
0x11F – 0x110	MSB (Flags) LSB	RXNONCE / RXCTR	Receiver nonce for in-line authentication or receiver counter for in-line decryption.
0x10F – 0x100	MSB LSB	KEY0	Encryption key 0
0x0FF – 0x080	MSB LSB	RXFIFO	128 bytes receive FIFO
0x07F – 0x000	MSB LSB	TXFIFO	128 bytes transmit FIFO

Table 6. CC2420 RAM Memory Space

#### FIFO access

The TXFIFO and RXFIFO may be accessed through the TXFIFO (0x3E) and RXFIFO (0x3F) registers.

The TXFIFO is write only, but may be read back using RAM access as described in the previous section. Data is read and written one byte at a time, as with RAM access. The RXFIFO is both writeable and readable. Writing to the RXFIFO should however only be done for debugging or for using the RXFIFO for security operations (decryption / authentication).

The crystal oscillator must be running when accessing the FIFOs.

When writing to the TXFIFO, the status byte (see Table 5) is output for each new data byte on SD, as shown in Figure 8. This could be used to detect TXFIFO underflow (see section RF Data Buffering section on page 37) while writing data to the TXFIFO.

Multiple FIFO bytes may be accessed in one operation, as with the RAM access.

FIFO access can only be terminated by setting the CS<sub>n</sub> pin high once it has been started.

The FIFO and FIFOB pins also provide additional information on the data in the receive FIFO, as will be described in the Microcontroller Interface and Pin Description section on page 30. Note that the FIFO and FIFOB pins only apply to the RXFIFO. The TXFIFO has its underflow flag in the status byte.

The TXFIFO may be flushed by issuing a `3FLUSHTX` command strobe. Similarly, a `3FLUSHRX` command strobe will flush the receive FIFO.

#### Multiple SPI access

Register access, command strobes, FIFO access and RAM access may be issued continuously without setting CS<sub>n</sub> high. E.g. the user may issue a command strobe, a register write and writing 3 bytes to the TXFIFO in one operation, as illustrated in Figure 10. The only exception



is that FIFO and RAM access must be terminated by setting CS<sub>n</sub> high.



Figure 10. Multiple SPI Access Example

### Microcontroller Interface and Pin Description

When used in a typical system, **CC2420** will interface to a microcontroller. This microcontroller must be able to:

- Program **CC2420** into different modes, read and write buffered data, and read back status information via the 4-wire SPI-bus configuration interface (SI, SO, SCLK and CS<sub>n</sub>).
- Interface to the receive and transmit FIFOs using the FIFO and FIFO<sub>W</sub> status pins.
- Interface to the CCA pin for clear channel assessment.
- Interface to the SFD pin for timing information (particularly for beaconing networks).

#### Configuration interface

A **CC2420** to microcontroller interface example is shown in Figure 11. The microcontroller uses 4 I/O pins for the SPI

configuration interface (SI, SO, SCLK and CS<sub>n</sub>). SO should be connected to an input at the microcontroller. SI, SCLK and CS<sub>n</sub> must be microcontroller outputs. Preferably the microcontroller should have a hardware SPI interface.

The microcontroller pins connected to SI, SO and SCLK can be shared with other SPI-interface devices. SO is a high impedance output as long as CS<sub>n</sub> is not activated (active low).

CS<sub>n</sub> should have an external pull-up resistor or be set to a high level when the voltage regulator is turned off in order to prevent the input from floating. SI and SCLK should be set to a defined level to prevent the inputs from floating.

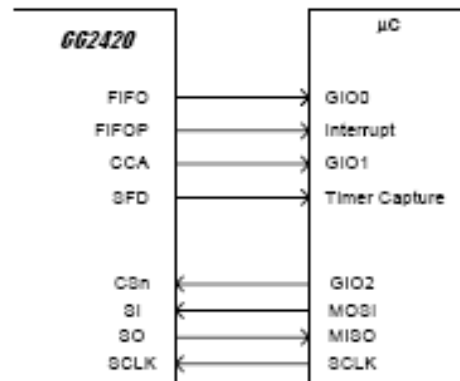


Figure 11. Microcontroller interface example

**Receive mode**

In receive mode, the SFD pin goes high after the start of frame delimiter (SFD) field has been completely received. If address recognition is disabled or is successful, the SFD pin goes low again only after the last byte of the MPDU has been received. If the received frame fails address recognition, the SFD pin goes low immediately. This is illustrated in Figure 12.

The FIFO pin is high when there is one or more data bytes in the RXFIFO. The first byte to be stored in the RXFIFO is the length field of the received frame, i.e. the FIFO pin is set high when the length field is written to the RXFIFO. The FIFO pin then remains high until the RXFIFO is empty.

If a previously received frame is completely or partially inside the RXFIFO, the FIFO pin will remain high until the RXFIFO is empty.

The FIFOP pin is high when the number of unread bytes in the RXFIFO exceeds the threshold programmed into IOCFG0.FIFOP\_THR. When address recognition is enabled the FIFOP pin will not go high until the incoming frame passes address recognition, even if the number of bytes in the RXFIFO exceeds the programmed threshold.

The FIFOP pin will also go high when the last byte of a new packet is received, even

if the threshold is not exceeded. If so the FIFOP pin will go back to low once one byte has been read out of the RXFIFO.

When address recognition is enabled, data should not be read out of the RXFIFO before the address is completely received, since the frame may be automatically flushed by CC2420 if it fails address recognition. This may be handled by using the FIFOP pin, since this pin does not go high until the frame passes address recognition.

Figure 13 shows an example of pin activity when reading a packet from the RXFIFO. In this example, the packet size is 8 bytes, IOCFG0.FIFOP\_THR = 3 and MODEMCTRL0.AUTOCRC is set. The length will be 8 bytes, RSSI will contain the average RSSI level during receiving of the packet and FCS/corr contain information of FCS check result and the correlation levels.

**RXFIFO overflow**

The RXFIFO can only contain a maximum of 128 bytes at a given time. This may be divided between multiple frames, as long as the total number of bytes is 128 or less. If an overflow occurs in the RXFIFO, this is signalled to the microcontroller by setting the FIFO pin low while the FIFOP pin is high. Data already in the RXFIFO will not be affected by the overflow, i.e. frames already received may be read out.

A SFLUSHRX command strobe is required after a RXFIFO overflow to enable reception of new data. Note that the



# SmartRF® CC2420

SFLUSHRX command strobe should be issued twice to ensure that the SFD pin goes back to its idle state.

For security enabled frames, the MAC layer must read the source address of the received frame before it can decide which key to use to decrypt or authenticate. This data must therefore not be overwritten

even if it has been read out of the RXFIFO by the microcontroller. If the SECCTRL0.RXFIPO\_PROTECTION control bit is set, **CC2420** also protects the frame header of security enabled frames until decryption has been performed. If no MAC security is used or if it is implemented outside the **CC2420**, this bit may be cleared to achieve optimal use of the RXFIFO.

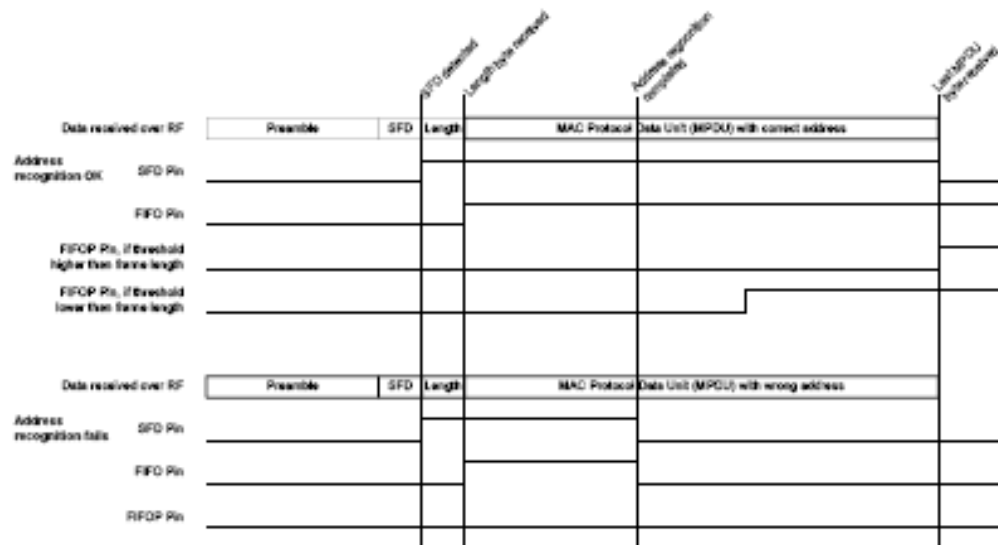


Figure 12. Pin activity examples during receive

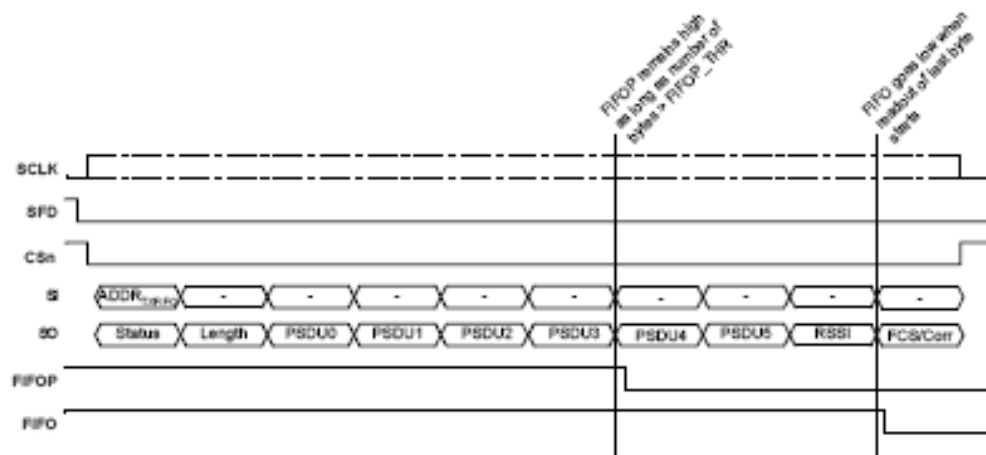


Figure 13. Example of pin activity when reading RXFIFO.


**SmartRF® CC2420**

### Transmit mode

During transmit, the FIFO and FIFOP pins are still only related to the RXFIFO. The SFD pin is however active during transmission of a data frame, as shown in Figure 14.

The SFD pin goes high when the SFD field has been completely transmitted. It goes low again when the complete MPDU (as defined by the length field) has been transmitted or if an underflow is detected.

See the RF Data Buffering section on page 37 for more information on TXFIFO underflow.

As can be seen from comparing Figure 12 and Figure 14, the SFD pin behaves very similarly during reception and transmission of a data frame. If the SFD pins of the transmitter and the receiver are compared during the transmission of a data frame, a small delay of approximately 2  $\mu$ s can be seen because of bandwidth limitations in both the transmitter and the receiver.

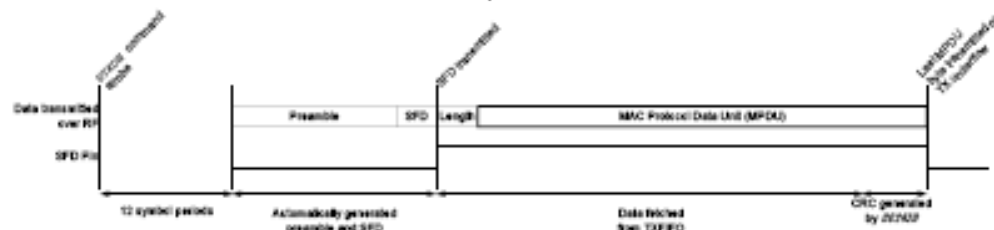


Figure 14. Pin activity example during transmit

### General control and status pins

In receive mode, the FIFOP pin can be used to interrupt the microcontroller when a threshold has been exceeded or a complete frame has been received. This pin should then be connected to a microcontroller interrupt pin.

In receive mode, the FIFO pin can be used to detect if there is data at all in the receive FIFO.

The SFD pin can be used to extract the timing information of transmitted and

received data frames. The SFD pin will go high when a start of frame delimiter has been completely detected / transmitted. The SFD pin should preferably be connected to a timer capture pin on the microcontroller.

For debug purposes, the SFD and CCA pins can be used to monitor several status signals as selected by the IOCFG1 register. See Table 12 and Table 13 for available signals.

The polarity of FIFO, FIFOP, SFD and CCA can be controlled by the IOCFG0 register (address 0x1C).

### Demodulator, Symbol Synchroniser and Data Decision

The block diagram for the **CC2420** demodulator is shown in Figure 15. Channel filtering and frequency offset compensation is performed digitally. The signal level in the channel is estimated to generate the RSSI level (see the RSSI / Energy Detection section on page 47 for more information). Data filtering is also included for enhanced performance.

With the  $\pm 40$  ppm frequency accuracy requirement from [1], a compliant receiver

must be able to compensate for up to 80 ppm or 200 kHz. The **CC2420** demodulator tolerates up to 300 kHz offset without significant degradation of the receiver performance.

Soft decision is used at the chip level, i.e. the demodulator does not make a decision for each chip, only for each received symbol. De-spreading is performed using over sampled symbol correlators. Symbol synchronisation is achieved by a





continuous start of frame delimiter (SFD) search.

When a SFD is detected, data is written to the RXFIFO and may be read out by the microcontroller at a lower bit rate than the 250 kbps generated by the receiver.

The **CC2420** demodulator also handles symbol rate errors in excess of 120 ppm without performance degradation.

Resynchronisation is performed continuously to adjust for error in the incoming symbol rate.

The `MDMCTRL1.CORR_THR` control bits should be written to 20 to set the threshold for detecting IEEE 802.15.4 start of frame delimiters.

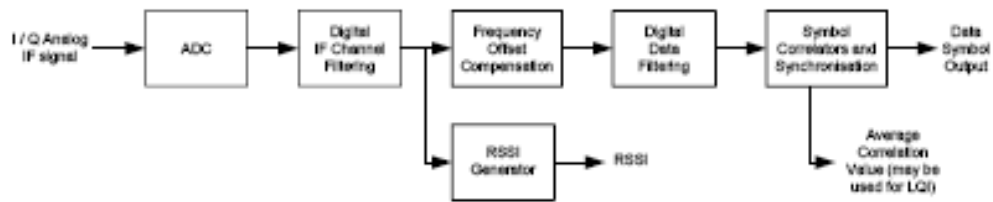


Figure 15. Demodulator Simplified Block Diagram

### Frame Format

**CC2420** has hardware support for parts of the IEEE 802.15.4 frame format. This section gives a brief summary to the IEEE 802.15.4 frame format, and describes how **CC2420** is set up to comply with this.

Figure 16 [1] shows a schematic view of the IEEE 802.15.4 frame format. Similar figures describing specific frame formats (data frames, beacon frames, acknowledgment frames and MAC command frames) are included in [1].

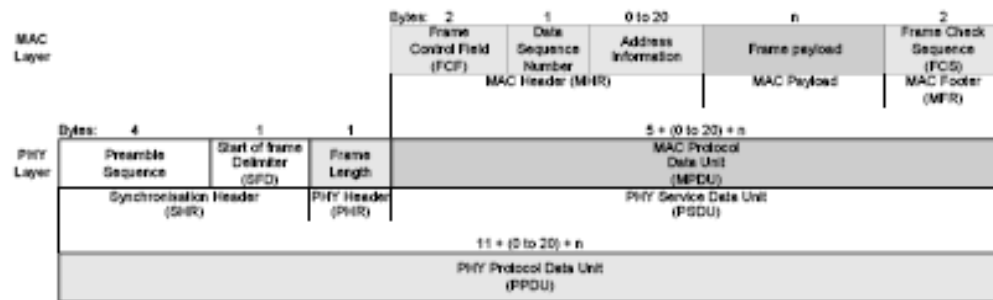


Figure 16. Schematic view of the IEEE 802.15.4 Frame Format [1]

#### Synchronisation header

The synchronisation header (SHR) consists of the preamble sequence

followed by the start of frame delimiter (SFD). In [1], the preamble sequence is defined to be 4 bytes of 0x00. The SFD is one byte, set to 0xA7.



# SmartRF® CC2420

In **CC2420**, the preamble length and SFD is configurable. The default values are compliant with [1]. Changing these values will make the system non-compliant to IEEE 802.15.4.

A synchronisation header is always transmitted first in all transmit modes.

The preamble sequence length can be set by `MDMCTRL0.PREAMBLE_LENGTH`, while the SFD is programmed in the `SYNCWORD` register. `SYNCWORD` is 2 bytes long, which gives the user some extra flexibility as described below. Figure 17 shows how the **CC2420** synchronisation header relates to the IEEE 802.15.4 specification.

The programmable preamble length only applies to transmission, it does not affect receive mode. The preamble length should not be set shorter than the default value. Note that 2 of the 8 zero-symbols in the preamble sequence required by [1] are included in the `SYNCWORD` register so that

the **CC2420** preamble sequence is only 6 symbols long for compliance with [1]. Two additional zero symbols in `SYNCWORD` make **CC2420** compliant with [1].

In reception, **CC2420** synchronises to received zero-symbols and searches for the SFD sequence defined by the `SYNCWORD` register. The least significant symbols in `SYNCWORD` set to 0xF will be ignored, while symbols different from 0xF will be required for synchronisation. The default setting of 0xA70F thereby requires one additional zero-symbol for synchronisation. This will reduce the number of false frames detected due to noise.

In receive mode **CC2420** uses the preamble sequence for symbol synchronisation and frequency offset adjustments. The SFD is used for byte synchronisation, and is not part of the data stored in the receive buffer (RXFIFO).

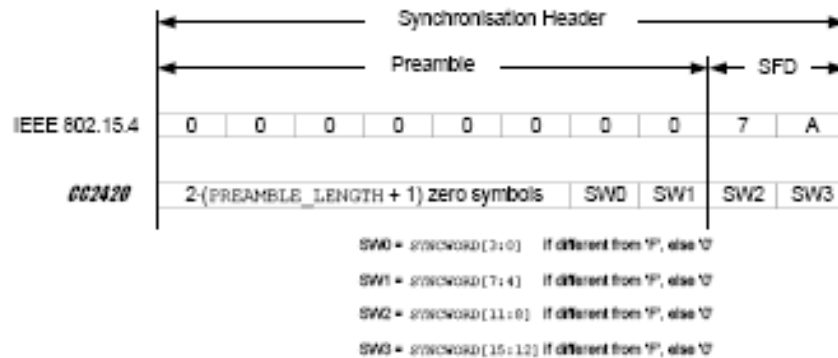


Figure 17. Transmitted Synchronisation Header

### Length field

The frame length field shown in Figure 16 defines the number of bytes in the MPDU. Note that the length field does not include the length field itself. It does however include the FCS (Frame Check Sequence), even if this is inserted automatically by **CC2420** hardware. It also includes the MIC if authentication is used.

The length field is 7 bits and has a maximum value of 127. The most

significant bit in the length field is reserved [1], and should be set to zero.

**CC2420** uses the length field both for transmission and reception, so this field must always be included. In transmit mode, the length field is used for underflow detection, as described in the FIFO access section on page 29.

### MAC protocol data unit

The FCF, data sequence number and address information follows the length field



## SmartRF® CC2420

as shown in Figure 18. Together with the MAC data payload and Frame Check Sequence, they form the MAC Protocol Data Unit (MPDU).

The format of the FCF is shown in Figure 18. Please refer to [1] for details.

There is no hardware support for the data sequence number, this field must be inserted and verified by software.

**CC2420** includes hardware address recognition, as described in the Address Recognition section on page 39.

Bits: 0-2	3	4	5	6	7-9	10-11	12-13	14-15
Frame Type	Security Enabled	Frame Pending	Acknowledge request	Intra PAN	Reserved	Destination addressing mode	Reserved	Source addressing mode

Figure 18. Format of the Frame Control Field (FCF) [1]

### Frame check sequence

A 2-byte frame check sequence (FCS) follows the last MAC payload byte as shown in Figure 18. The FCS is calculated over the MPDU, i.e. the length field is not part of the FCS. This field is automatically generated and verified by hardware when the `MODEMCTRL0.AUTOCRC` control bit is set. It is recommended to always have this enabled, except possibly for debug purposes. If cleared, CRC generation and verification must be performed by software.

The FCS polynomial is [1]:

$$x^{16} + x^{12} + x^5 + 1$$

The **CC2420** hardware implementation is shown in Figure 19. Please refer to [1] for further details.

In transmit mode the FCS is appended at the correct position defined by the length field. The FCS is not written to the TXFIFO, but stored in a separate 16-bit register.

In receive mode the FCS is verified by hardware. The user is normally only interested in the correctness of the FCS,

not the FCS sequence itself. The FCS sequence itself is therefore not written to the RXFIFO during receive.

Instead, when `MODEMCTRL0.AUTOCRC` is set the two FCS bytes are replaced by the RSSI value, average correlation value (used for LQI) and CRC OK/not OK. This is illustrated in Figure 20.

The first FCS byte is replaced by the 8-bit RSSI value. See the RSSI section on page 47 for details.

The 7 least significant bits in the last FCS byte are replaced by the average correlation value of the 8 first symbols of the received PHY header (length field) and PHY Service Data Unit (PSDU). This correlation value may be used as a basis for calculating the LQI. See the Link Quality Indication section on page 48 for details.

The most significant bit in the last byte of each frame is set high if the CRC of the received frame is correct and low otherwise.

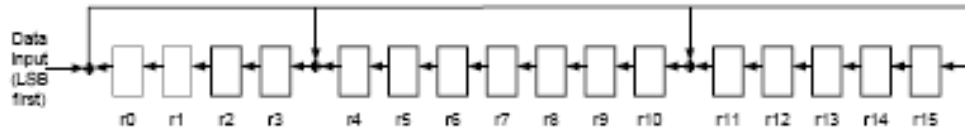


Figure 19. *CC2420* Frame Check Sequence (FCS) hardware implementation [1]

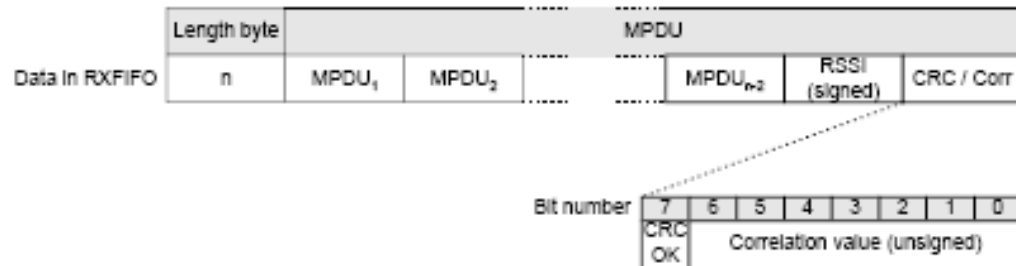


Figure 20. Data in RXFIFO when `MDMCTRL0.AUTOCRC` is set

### RF Data Buffering

*CC2420* can be configured for different transmit and receive modes, as set in the `MDMCTRL1.TX_MODE` and `MDMCTRL1.RX_MODE` control bits. Buffered mode (mode 0) will be used for normal operation of *CC2420*, while other modes are available for test purposes.

#### Buffered transmit mode

In buffered transmit mode (`TX_MODE 0`), the 128 byte TXFIFO, located in *CC2420* RAM, is used to buffer data before transmission. A preamble sequence (defined in the Frame Format section below) is automatically inserted before the length field during transmission. The length field must always be the first byte written to the transmit buffer for all frames.

Writing one or multiple bytes to the TXFIFO is described in the FIFO access section on page 29. Reading data from the TXFIFO is possible with RAM access, but this does not remove the byte from the FIFO.

Transmission is enabled by issuing a `STXON` or `STXONCCA` command strobe. See the Radio control state machine

section on page 42 for an illustration of how the transmit command strobes affect the state of *CC2420*. The `STXONCCA` strobe is ignored if the channel is busy. See the



## SmartRF® CC2420

Clear Channel Assessment section on page 49 for details on CCA.

The preamble sequence is started 12 symbol periods after the command strobe. After the programmable start of frame delimiter has been transmitted, data is fetched from the TXFIFO.

A TXFIFO underflow is issued if too few bytes are written to the TXFIFO. Transmission is then automatically stopped. The underflow is indicated in the TX\_UNDERFLOW status bit, which is returned during each address byte and each byte written to the TXFIFO. The underflow bit is only cleared by issuing a SFLUSHTX command strobe.

The TXFIFO can only contain one data frame at a given time.

After complete transmission of a data frame, the TXFIFO is automatically refilled with the last transmitted frame. Issuing a new STXON or STXONCCA command strobe will then cause **CC2420** to retransmit the last frame.

Writing to the TXFIFO after a frame has been transmitted will cause the TXFIFO to be automatically flushed before the new byte is written. The only exception is if a TXFIFO underflow has occurred, when a SFLUSHTX command strobe is required.

### Buffered receive mode

In buffered receive mode (RX\_MODE 0), the 128 byte RXFIFO, located in **CC2420** RAM, is used to buffer data received by the demodulator. Accessing data in the RXFIFO is described in the FIFO access section on page 29.

The FIFO and FIFOP pins are used to assist the microcontroller in supervising the RXFIFO. Please note that the FIFO and FIFOP pins are only related to the RXFIFO, even if **CC2420** is in transmit mode.

Multiple data frames may be in the RXFIFO simultaneously, as long as the total number of bytes does not exceed 128.

See the RXFIFO overflow section on page 31 for details on how a RXFIFO overflow is detected and signaled.

### Un-buffered, serial mode

Un-buffered mode should be used for evaluation / debugging purposes only. Buffered mode is recommended for all applications.

In un-buffered mode, the FIFO and FIFOP pins are reconfigured as data and data clock pins. The TXFIFO and RXFIFO buffers are not used in this mode. A synchronous data clock is provided by **CC2420** at the FIFOP pin, and the FIFO pin is used as data input/output. The FIFOP clock frequency is 250 kHz when active. This is illustrated in Figure 21.

In serial transmit mode (MDMCTRL1.TX\_MODE=1), a synchronisation sequence is inserted at the start of each frame by hardware, as in buffered mode. Data is sampled by **CC2420** on the positive edge of FIFOP and should be updated by the microcontroller on the negative edge of FIFOP. See Figure 21 for an illustration of the timing in serial transmit mode. The SFD and CCA pins retain their normal operation also in serial mode. **CC2420** will remain in serial transmit mode until transmission is turned off manually.

In serial receive mode (MDMCTRL1.RX\_MODE=1) byte synchronisation is still performed by **CC2420**. This means that the FIFOP clock pin will remain idle low until a start of frame delimiter has been detected.

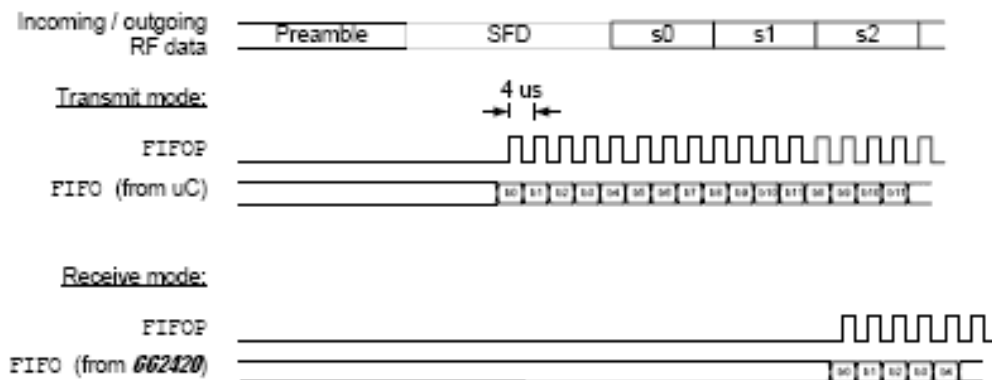

**SmartRF® CC2420**


Figure 21. Un-buffered test mode, pin activity

### Address Recognition

**CC2420** includes hardware support for address recognition, as specified in [1]. Hardware address recognition may be enabled / disabled using the `MDMCTRL0.ADR_DECODE` control bit.

Address recognition is based on the following requirements, listed from section 7.5.6.2 in [1]:

- The frame type subfield shall not contain an illegal frame type
- If the frame type indicates that the frame is a beacon frame, the source PAN identifier shall match `macPANId` unless `macPANId` is equal to `0xFFFF`, in which case the beacon frame shall be accepted regardless of the source PAN identifier.
- If a destination PAN identifier is included in the frame, it shall match `macPANId` or shall be the broadcast PAN identifier (`0xFFFF`).
- If a short destination address is included in the frame, it shall match either `macShortAddress` or the broadcast address (`0xFFFF`). Otherwise if an extended destination address is included in the frame, it shall match `aExtendedAddress`.

- If only source addressing fields are included in a data or MAC command frame, the frame shall only be accepted if the device is a PAN coordinator and the source PAN identifier matches `macPANId`.

If any of the above requirements are not satisfied and address recognition is enabled, **CC2420** will disregard the incoming frame and flush the data from the RXFIFO. Only data from the rejected frame is flushed, data from previously accepted frames may still be in the RXFIFO.

The `IOCFG0.BCN_ACCEPT` control bit must be set when the PAN identifier programmed into **CC2420** RAM is equal to `0xFFFF` and cleared otherwise. This particularly applies to active and passive scans as defined by [1] which requires all received beacons to be processed by the MAC sublayer.

Incoming frames with reserved frame types (FCF frame type subfield is 4, 5, 6 or 7) is however accepted if the `RESERVED_FRAME_MODE` control bit in `MDMCTRL0` is set. In this case, no further address recognition is performed on these frames. This option is included for future expansions of the IEEE 802.15.4 standard.



## SmartRF® CC2420

If a frame is rejected, **CC2420** will only start searching for a new frame after the rejected frame has been completely received (as defined by the length field) to avoid detecting false SFDs within the frame.

The `MDMCTRL0.PAN_COORDINATOR` control bit must be correctly set, since parts of the address recognition procedure requires knowledge about whether the current device is a PAN coordinator or not.

### Acknowledge Frames

**CC2420** includes hardware support for transmitting acknowledge frames, as specified in [1]. Figure 22 shows the format of the acknowledge frame.

If `MDMCTRL0.AUTOACK` is enabled, an acknowledge frame is transmitted for all incoming frames accepted by the address recognition with the acknowledge request flag set and a valid CRC. `AUTOACK` therefore does not make sense unless

also `ADR_DECODE` and `AUTOCRC` are enabled. The sequence number is copied from the incoming frame.

`AUTOACK` may be used for non-beacon systems as long as the frame pending field (see Figure 18) is cleared. The acknowledge frame is then transmitted 12 symbol periods after the last symbol of the incoming frame. This is as specified by [1] for non-beacon networks.

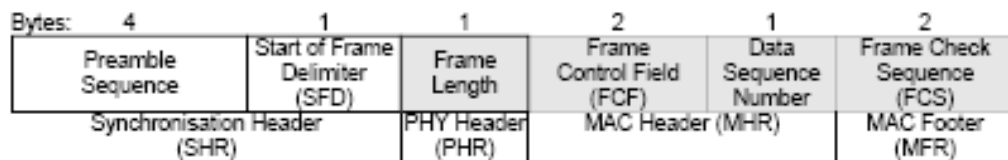


Figure 22. Acknowledge frame format [1]

Two command strobos, `SACK` and `SACKPEND` are defined to transmit acknowledge frames with the frame pending field cleared or set, respectively. The acknowledge frame is only transmitted if the CRC is valid.

For systems using beacons, there is an additional timing requirement that the acknowledge frame transmission should be started on the first backoff-slot boundary (20 symbol periods) at least 12 symbol periods after the last symbol of the incoming frame. This timing must be controlled by the microcontroller by issuing the `SACK` and `SACKPEND` command strobe 12 symbol periods before the following backoff-slot boundary, as illustrated in Figure 23.

If a `SACK` or `SACKPEND` command strobe is issued while receiving an incoming frame, the acknowledge frame is transmitted 12 symbol periods after the last symbol of the incoming frame. This should be used to transmit acknowledge frames in non-beacon networks. This timing is also illustrated in Figure 23.

Using `SACKPEND` will set the pending data flag for automatically transmitted acknowledge frames using `AUTOACK`. The pending flag will then be set also for future acknowledge frames, until a `SACK` command strobe is issued.

Acknowledge frames may be manually transmitted using normal data transmission if desired.



## Configuration Registers

The configuration of **CC2420** is done by programming the 16-bit configuration registers. Complete descriptions of the registers are given in the following tables. After chip reset (from the **RESETn** pin or programmable through the **MAIN.RESETn** configuration bit), all the registers have default values as shown in the tables.

Note that the **MAIN** register is only reset by using the pin reset **RESETn**. When writing to this register, all bits will get the value written, not the default value. This also means that the **MAIN.RESETn** bit must be written both low and then high to perform a chip reset through the serial interface.

15 registers are Strobe Command Registers, listed first in Table 11 below. Accessing these registers will initiate the change of an internal state or mode. There

are 33 normal 16-bits registers, also listed in Table 11. Many of these registers are for test purposes only, and need not be accessed for normal operation of **CC2420**.

The FIFOs are accessed through two 8-bit registers, **TXFIFO** and **RXFIFO**. The **TXFIFO** register is write only. Data may still be read out of the **TXFIFO** through regular RAM access (see section RAM access section on page 27), but data is then not removed from the FIFO. Note that the crystal oscillator must be active for all FIFO and RAM access.

During the address transfer and while writing to the **TXFIFO**, a status byte is returned on the serial data output pin **SDO**. This status byte is described in Table 5 on page 27.

All configuration and status registers are described in the tables following Table 11.

Address	Register	Register type	Description
0x00	SNOP	S	No Operation (has no other effect than reading out status-bits)
0x01	SXOSCCON	S	Turn on the crystal oscillator (set <b>XOSC16M_PD</b> = 0 and <b>BIAS_PD</b> = 0)
0x02	STXCAL	S	Enable and calibrate frequency synthesizer for TX; Go from RX / TX to a wait state where only the synthesizer is running.
0x03	SRXON	S	Enable RX
0x04	STXON	S	Enable TX after calibration (if not already performed) Start TX in-line encryption if <b>SPI_SEC_MODE</b> ≠ 0
0x05	STXONCCA	S	If CCA indicates a clear channel: Enable calibration, then TX. Start in-line encryption if <b>SPI_SEC_MODE</b> ≠ 0 else do nothing
0x06	SRFOFF	S	Disable RX/TX and frequency synthesizer
0x07	SXOSCOFF	S	Turn off the crystal oscillator and RF
0x08	SFLUSHRX	S	Flush the RX FIFO buffer and reset the demodulator. Always read at least one byte from the <b>RXFIFO</b> before issuing the <b>SFLUSHRX</b> command strobe
0x09	SFLUSHTX	S	Flush the TX FIFO buffer
0x0A	SACK	S	Send acknowledge frame, with pending field cleared.
0x0B	SACKPEND	S	Send acknowledge frame, with pending field set.
0x0C	SRXDEC	S	Start <b>RXFIFO</b> in-line decryption / authentication (as set by <b>SPI_SEC_MODE</b> )
0x0D	STXENC	S	Start <b>TXFIFO</b> in-line encryption / authentication (as set by <b>SPI_SEC_MODE</b> ), without starting TX.




**SmartRF® CC2420**

Address	Register	Register type	Description
0x0E	SAES	S	AES Stand alone encryption strobe. SPI_SEC_MODE is not required to be 0, but the encryption module must be idle. If not, the strobe is ignored.
0x0F	-	-	Not used
0x10	MAIN	R/W	Main Control Register
0x11	MEMCTRL0	R/W	Modem Control Register 0
0x12	MEMCTRL1	R/W	Modem Control Register 1
0x13	RSSI	R/W	RSSI and CCA Status and Control register
0x14	SYNCWORD	R/W	Synchronization word control register
0x15	TXCTRL	R/W	Transmit Control Register
0x16	RXCTRL0	R/W	Receive Control Register 0
0x17	RXCTRL1	R/W	Receive Control Register 1
0x18	FSCTRL	R/W	Frequency Synthesizer Control and Status Register
0x19	SECCTRL0	R/W	Security Control Register 0
0x1A	SECCTRL1	R/W	Security Control Register 1
0x1B	BATTMON	R/W	Battery Monitor Control and Status Register
0x1C	IOCFG0	R/W	Input / Output Control Register 0
0x1D	IOCFG1	R/W	Input / Output Control Register 1
0x1E	MANFIDL	R/W	Manufacturer ID, Low 16 bits
0x1F	MANFIDH	R/W	Manufacturer ID, High 16 bits
0x20	FSM7C	R/W	Finite State Machine Time Constants
0x21	MANAND	R/W	Manual signal AND override register
0x22	MANOR	R/W	Manual signal OR override register
0x23	AGCCTRL	R/W	AGC Control Register
0x24	AGCTST0	R/W	AGC Test Register 0
0x25	AGCTST1	R/W	AGC Test Register 1
0x26	AGCTST2	R/W	AGC Test Register 2
0x27	FSTST0	R/W	Frequency Synthesizer Test Register 0
0x28	FSTST1	R/W	Frequency Synthesizer Test Register 1
0x29	FSTST2	R/W	Frequency Synthesizer Test Register 2
0x2A	FSTST3	R/W	Frequency Synthesizer Test Register 3
0x2B	RXBPF7ST	R/W	Receiver Bandpass Filter Test Register
0x2C	FSMSTATE	R	Finite State Machine State Status Register
0x2D	ADCTST	R/W	ADC Test Register
0x2E	DACTST	R/W	DAC Test Register
0x2F	TOP7ST	R/W	Top Level Test Register
0x30	RESERVED	R/W	Reserved for future use control / status register
0x31- 0x3D	-	-	Not used
0x3E	TXFIFO	W	Transmit FIFO Byte Register
0x3F	RXFIFO	R/W	Receiver FIFO Byte Register

R/W - Read/write (control/status), R - Read only, W - Write only, S - Command Strobe (perform action upon access)

**Table 11. Configuration registers overview**

C8051F212  
ADC

C8051F120/1/2/3/4/5/6/7  
C8051F130/1/2/3

5. ADC0 (12-Bit ADC, C8051F120/1/4/5 Only)

The ADC0 subsystem for the C8051F120/1/4/5 consists of a 9-channel, configurable analog multiplexer (AMUX0), a programmable gain amplifier (PGA0), and a 100 ksps, 12-bit successive-approximation-register ADC with integrated track-and-hold and Programmable Window Detector (see block diagram in Figure 5.1). The AMUX0, PGA0, Data Conversion Modes, and Window Detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. The voltage reference used by ADC0 is selected as described in [Section "9. Voltage Reference" on page 113](#). The ADC0 subsystem (ADC0, track-and-hold and PGA0) is enabled only when the AD0EN bit in the ADC0 Control register (AD0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

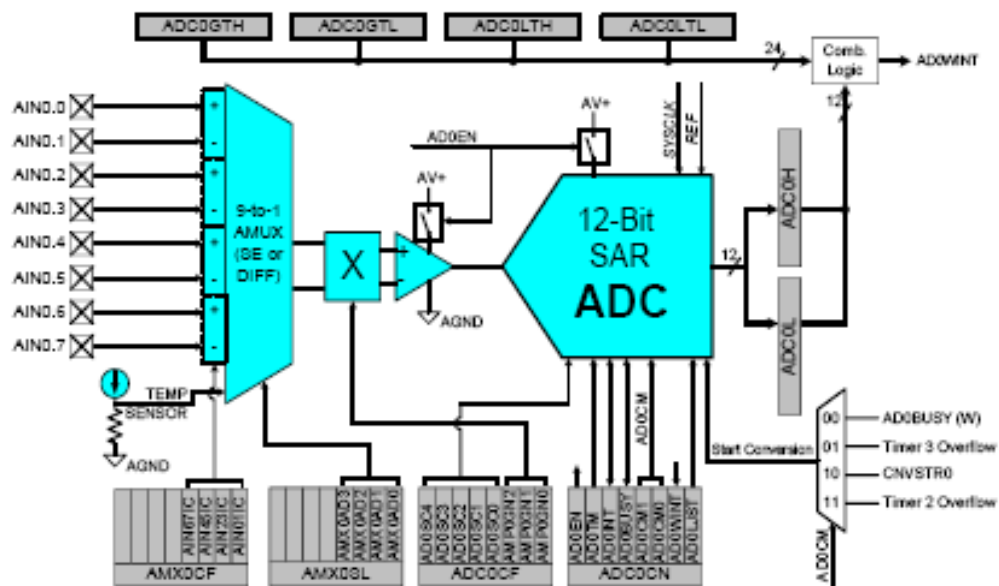


Figure 5.1. 12-Bit ADC0 Functional Block Diagram

5.1. Analog Multiplexer and PGA

Eight of the AMUX channels are available for external measurements while the ninth channel is internally connected to an on-chip temperature sensor (temperature transfer function is shown in Figure 5.2). AMUX input pairs can be programmed to operate in either differential or single-ended mode. This allows the user to select the best measurement technique for each input channel, and even accommodates mode changes "on-the-fly". The AMUX defaults to all single-ended inputs upon reset. There are two registers associated with the AMUX: the Channel Selection register AMX0SL (SFR Definition 5.2), and the Configuration register AMX0CF (SFR Definition 5.1). The table in SFR Definition 5.2 shows AMUX functionality by channel, for each possible configuration. The PGA amplifies the AMUX output signal by an amount determined by the states of the AMP0GN2-0 bits in the ADC0 Configuration register, AD0CF (SFR Definition 5.3). The PGA can be software-programmed for gains of 0.5, 2, 4, 8 or 16. Gain defaults to unity on reset.

## C8051F120/1/2/3/4/5/6/7

## C8051F130/1/2/3

The Temperature Sensor transfer function is shown in Figure 5.2. The output voltage ( $V_{TEMP}$ ) is the PGA input when the Temperature Sensor is selected by bits AMX0AD3-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings. Typical values for the Slope and Offset parameters can be found in Table 5.1.

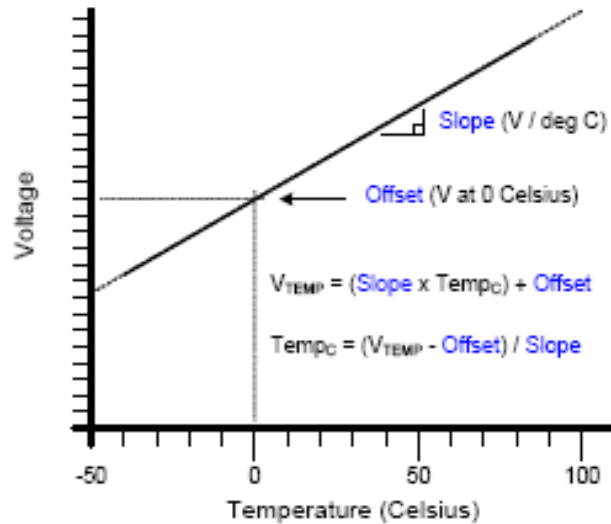


Figure 5.2. Typical Temperature Sensor Transfer Function

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### 5.2. ADC Modes of Operation

ADC0 has a maximum conversion speed of 100 ksp/s. The ADC0 conversion clock is derived from the system clock divided by the value held in the ADCSC bits of register ADC0CF.

#### 5.2.1. Starting a Conversion

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM1, AD0CM0) in ADC0CN. Conversions may be initiated by:

1. Writing a '1' to the AD0BUSY bit of ADC0CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR0;
4. A Timer 2 overflow (i.e. timed continuous conversions).

The AD0BUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the AD0INT interrupt flag (ADC0CN.5). Converted data is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L. Converted data can be either left or right justified in the ADC0H:ADC0L register pair (see example in Figure 5.5) depending on the programmed state of the AD0LJST bit in the ADC0CN register.

When initiating conversions by writing a '1' to AD0BUSY, the AD0INT bit should be polled to determine when a conversion has completed (ADC0 interrupts may also be used). The recommended polling procedure is shown below.

- Step 1. Write a '0' to AD0INT;
- Step 2. Write a '1' to AD0BUSY;
- Step 3. Poll AD0INT for '1';
- Step 4. Process ADC0 data.

When CNVSTR0 is used as a conversion start source, it must be enabled in the crossbar, and the corresponding pin must be set to open-drain, high-impedance mode (see [Section "18. Port Input/Output"](#) on [page 235](#) for more details on Port I/O configuration).

# C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

## 5.2.2. Tracking Modes

The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked when a conversion is not in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR0 signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR0 is low; conversion begins on the rising edge of CNVSTR0 (see Figure 5.3). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX or PGA settings are frequently changed, to ensure that settling time requirements are met (see [Section "5.2.3. Settling Time Requirements"](#) on page 59).

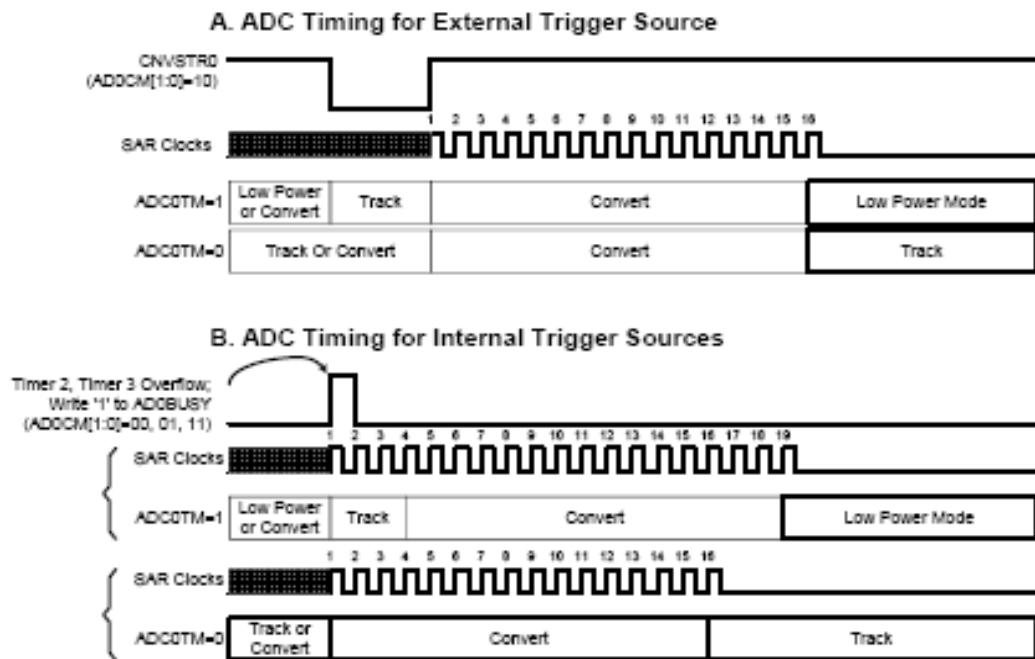


Figure 5.3. ADC0 Track and Conversion Example Timing

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### 5.2.3. Settling Time Requirements

A minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the ADC0 MUX resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 5.4 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required settling time for a given settling accuracy (SA) may be approximated by Equation 5.1. When measuring the Temperature Sensor output,  $R_{TOTAL}$  reduces to  $R_{MUX}$ . An absolute minimum settling time of 1.5  $\mu$ s is required after any MUX or PGA selection. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the tracking requirements.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Equation 5.1. ADC0 Settling Time Requirements

Where:

SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)

t is the required settling time in seconds

$R_{TOTAL}$  is the sum of the ADC0 MUX resistance and any external source resistance.

n is the ADC resolution in bits (12).

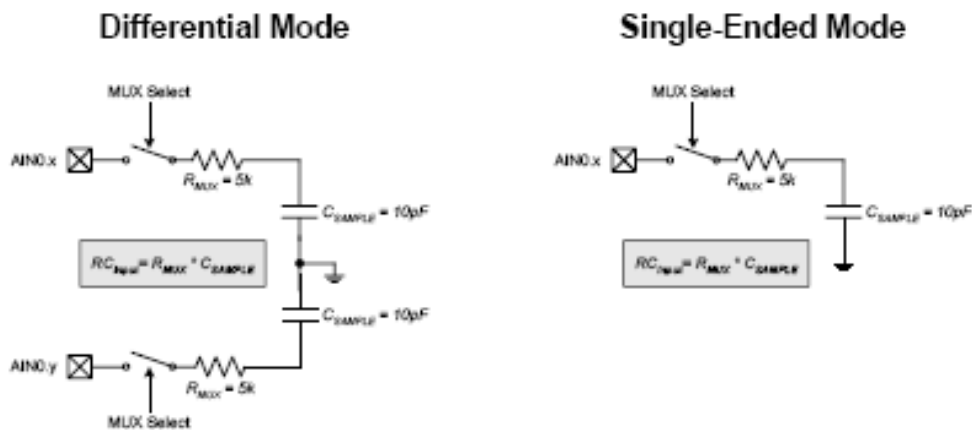


Figure 5.4. ADC0 Equivalent Input Circuits

## C8051F120/1/2/3/4/5/6/7

## C8051F130/1/2/3

### SFR Definition 5.1. AMX0CF: AMUX0 Configuration

SFR Page: 0								Reset Value
SFR Address: 0xBA								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	00000000
-	-	-	-	AIN67IC	AIN45IC	AIN23IC	AIN01IC	
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	

Bits7-4: UNUSED. Read = 0000b; Write = don't care.

Bit3: AIN67IC: AIN0.6, AIN0.7 Input Pair Configuration Bit.  
0: AIN0.6 and AIN0.7 are independent single-ended inputs.  
1: AIN0.6, AIN0.7 are (respectively) +, - differential input pair.

Bit2: AIN45IC: AIN0.4, AIN0.5 Input Pair Configuration Bit.  
0: AIN0.4 and AIN0.5 are independent single-ended inputs.  
1: AIN0.4, AIN0.5 are (respectively) +, - differential input pair.

Bit1: AIN23IC: AIN0.2, AIN0.3 Input Pair Configuration Bit.  
0: AIN0.2 and AIN0.3 are independent single-ended inputs.  
1: AIN0.2, AIN0.3 are (respectively) +, - differential input pair.

Bit0: AIN01IC: AIN0.0, AIN0.1 Input Pair Configuration Bit.  
0: AIN0.0 and AIN0.1 are independent single-ended inputs.  
1: AIN0.0, AIN0.1 are (respectively) +, - differential input pair.

Note: The ADC0 Data Word is in 2's complement format for channels configured as differential.

# C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

## SFR Definition 5.2. AMX0SL: AMUX0 Channel Select

SFR Page: 0  
SFR Address: 0xBB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	AMX0AD3	AMX0AD2	AMX0AD1	AMX0AD0	00000000
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	

Bits7-4: UNUSED. Read = 0000b; Write = don't care.  
Bits3-0: AMX0AD3-0: AMX0 Address Bits.  
0000-1111b: ADC Inputs selected per chart below.

		AMX0AD3-0								
		0000	0001	0010	0011	0100	0101	0110	0111	1xxx
AMX0CF Bits 3-0	0000	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	TEMP SENSOR
	0001	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	AIN0.6	AIN0.7	TEMP SENSOR
	0010	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	TEMP SENSOR
	0011	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	AIN0.6	AIN0.7	TEMP SENSOR
	0100	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TEMP SENSOR
	0101	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TEMP SENSOR
	0110	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TEMP SENSOR
	0111	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		AIN0.6	AIN0.7	TEMP SENSOR
	1000	AIN0.0	AIN0.1	AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TEMP SENSOR
	1001	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TEMP SENSOR
	1010	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TEMP SENSOR
	1011	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		AIN0.4	AIN0.5	+(AIN0.6) -(AIN0.7)		TEMP SENSOR
	1100	AIN0.0	AIN0.1	AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TEMP SENSOR
	1101	+(AIN0.0) -(AIN0.1)		AIN0.2	AIN0.3	+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TEMP SENSOR
	1110	AIN0.0	AIN0.1	+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TEMP SENSOR
	1111	+(AIN0.0) -(AIN0.1)		+(AIN0.2) -(AIN0.3)		+(AIN0.4) -(AIN0.5)		+(AIN0.6) -(AIN0.7)		TEMP SENSOR



# C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

## SFR Definition 5.3. ADC0CF: ADC0 Configuration

SFR Page: 0  
SFR Address: 0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AMP0GN2	AMP0GN1	AMP0GN0	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

**Bits7–3:** AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.  
The SAR Conversion clock is derived from system clock by the following equation, where AD0SC refers to the 5-bit value held in AD0SC4–0, and  $CLK_{SAR0}$  refers to the desired ADC0 SAR clock (Note: the ADC0 SAR Conversion Clock should be less than or equal to 2.5 MHz).

$$AD0SC = \frac{SYSCLK}{2 \times CLK_{SAR0}} - 1 \quad (AD0SC > 00000b)$$

When the AD0SC bits are equal to 00000b, the SAR Conversion clock is equal to SYSCLK to facilitate faster ADC conversions at slower SYSCLK speeds.

**Bits2–0:** AMP0GN2–0: ADC0 Internal Amplifier Gain (PGA).  
000: Gain = 1  
001: Gain = 2  
010: Gain = 4  
011: Gain = 8  
10x: Gain = 16  
11x: Gain = 0.5

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### SFR Definition 5.4. ADC0CN: ADC0 Control

SFR Page: 0								Reset Value
SFR Address: 0xE8 (bit addressable)								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0CM1	AD0CM0	AD0WINT	AD0LJST	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit7:	AD0EN: ADC0 Enable Bit. 0: ADC0 Disabled. ADC0 is in low-power shutdown. 1: ADC0 Enabled. ADC0 is active and ready for data conversions.
Bit6:	AD0TM: ADC Track Mode Bit. 0: When the ADC is enabled, tracking is continuous unless a conversion is in process. 1: Tracking Defined by AD0CM1-0 bits.
Bit5:	AD0INT: ADC0 Conversion Complete Interrupt Flag. This flag must be cleared by software. 0: ADC0 has not completed a data conversion since the last time this flag was cleared. 1: ADC0 has completed a data conversion.
Bit4:	AD0BUSY: ADC0 Busy Bit. Read: 0: ADC0 Conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY. 1: ADC0 Conversion is in progress. Write: 0: No Effect. 1: Initiates ADC0 Conversion if AD0CM1-0 = 00b.
Bits3-2:	AD0CM1-0: ADC0 Start of Conversion Mode Select. If AD0TM = 0: 00: ADC0 conversion initiated on every write of '1' to AD0BUSY. 01: ADC0 conversion initiated on overflow of Timer 3. 10: ADC0 conversion initiated on rising edge of external CNVSTR0. 11: ADC0 conversion initiated on overflow of Timer 2. If AD0TM = 1: 00: Tracking starts with the write of '1' to AD0BUSY and lasts for 3 SAR clocks, followed by conversion. 01: Tracking started by the overflow of Timer 3 and lasts for 3 SAR clocks, followed by conversion. 10: ADC0 tracks only when CNVSTR0 input is logic low; conversion starts on rising CNVSTR0 edge. 11: Tracking started by the overflow of Timer 2 and lasts for 3 SAR clocks, followed by conversion.
Bit1:	AD0WINT: ADC0 Window Compare Interrupt Flag. This bit must be cleared by software. 0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared. 1: ADC0 Window Comparison Data match has occurred.
Bit0:	AD0LJST: ADC0 Left Justify Select. 0: Data in ADC0H:ADC0L registers are right-justified. 1: Data in ADC0H:ADC0L registers are left-justified.

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### SFR Definition 5.5. ADC0H: ADC0 Data Word MSB

SFR Page: 0								Reset Value
SFR Address: 0xBF								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	00000000
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	

Bits7-0: ADC0 Data Word High-Order Bits.  
 For AD0LJST = 0: Bits 7-4 are the sign extension of Bit3. Bits 3-0 are the upper 4 bits of the 12-bit ADC0 Data Word.  
 For AD0LJST = 1: Bits 7-0 are the most-significant bits of the 12-bit ADC0 Data Word.

### SFR Definition 5.6. ADC0L: ADC0 Data Word LSB

SFR Page: 0								Reset Value
SFR Address: 0xBE								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	00000000
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	

Bits7-0: ADC0 Data Word Low-Order Bits.  
 For AD0LJST = 0: Bits 7-0 are the lower 8 bits of the 12-bit ADC0 Data Word.  
 For AD0LJST = 1: Bits 7-4 are the lower 4 bits of the 12-bit ADC0 Data Word. Bits 3-0 will always read '0'.

DAC

**C8051F120/1/2/3/4/5/6/7**  
**C8051F130/1/2/3**

**8. DACs, 12-Bit Voltage Mode (C8051F12x Only)**

The C8051F12x devices include two on-chip 12-bit voltage-mode Digital-to-Analog Converters (DACs). Each DAC has an output swing of 0 V to (VREF-1LSB) for a corresponding input code range of 0x000 to 0xFFF. The DACs may be enabled/disabled via their corresponding control registers, DAC0CN and DAC1CN. While disabled, the DAC output is maintained in a high-impedance state, and the DAC supply current falls to 1  $\mu$ A or less. The voltage reference for each DAC is supplied at the VREFD pin (C8051F120/2/4/8 devices) or the VREF pin (C8051F121/3/5/7 devices). Note that the VREF pin on C8051F121/3/5/7 devices may be driven by the internal voltage reference or an external source. If the internal voltage reference is used it must be enabled in order for the DAC outputs to be valid. See [Section "9. Voltage Reference" on page 113](#) for more information on configuring the voltage reference for the DACs.

**8.1. DAC Output Scheduling**

Each DAC features a flexible output update mechanism which allows for seamless full-scale changes and supports jitter-free updates for waveform generation. The following examples are written in terms of DAC0, but DAC1 operation is identical.

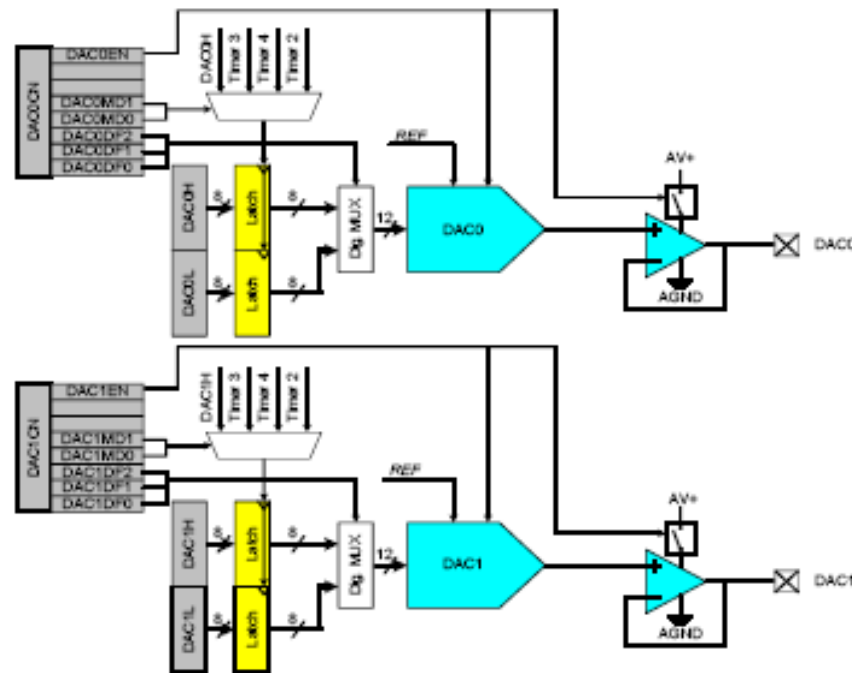


Figure 8.1. DAC Functional Block Diagram

## C8051F120/1/2/3/4/5/6/7

### C8051F130/1/2/3

#### 8.1.1. Update Output On-Demand

In its default mode (DAC0CN.[4:3] = '00') the DAC0 output is updated "on-demand" on a write to the high-byte of the DAC0 data register (DAC0H). It is important to note that writes to DAC0L are held, and have no effect on the DAC0 output until a write to DAC0H takes place. If writing a full 12-bit word to the DAC data registers, the 12-bit data word is written to the low byte (DAC0L) and high byte (DAC0H) data registers. Data is latched into DAC0 after a write to the corresponding DAC0H register, so the write sequence should be DAC0L followed by DAC0H if the full 12-bit resolution is required. The DAC can be used in 8-bit mode by initializing DAC0L to the desired value (typically 0x00), and writing data to only DAC0H (also see [Section 8.2](#) for information on formatting the 12-bit DAC data word within the 16-bit SFR space).

#### 8.1.2. Update Output Based on Timer Overflow

Similar to the ADC operation, in which an ADC conversion can be initiated by a timer overflow independently of the processor, the DAC outputs can use a Timer overflow to schedule an output update event. This feature is useful in systems where the DAC is used to generate a waveform of a defined sampling rate by eliminating the effects of variable interrupt latency and instruction execution on the timing of the DAC output. When the DAC0MD bits (DAC0CN.[4:3]) are set to '01', '10', or '11', writes to both DAC data registers (DAC0L and DAC0H) are held until an associated Timer overflow event (Timer 3, Timer 4, or Timer 2, respectively) occurs, at which time the DAC0H:DAC0L contents are copied to the DAC input latches allowing the DAC output to change to the new value.

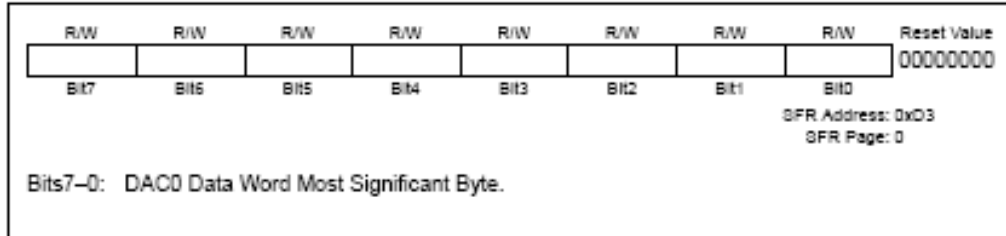
#### 8.2. DAC Output Scaling/Justification

In some instances, input data should be shifted prior to a DAC0 write operation to properly justify data within the DAC input registers. This action would typically require one or more load and shift operations, adding software overhead and slowing DAC throughput. To alleviate this problem, the data-formatting feature provides a means for the user to program the orientation of the DAC0 data word within data registers DAC0H and DAC0L. The three DAC0DF bits (DAC0CN.[2:0]) allow the user to specify one of five data word orientations as shown in the DAC0CN register definition.

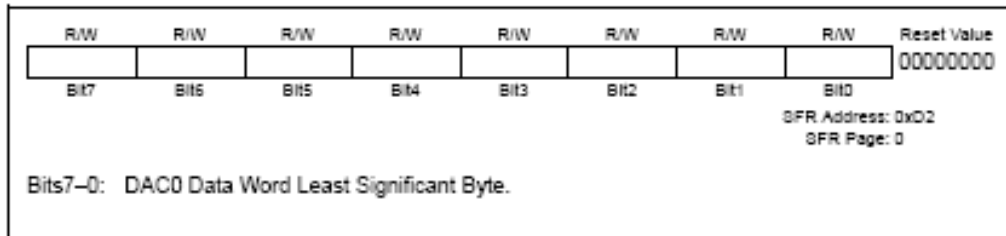
DAC1 is functionally the same as DAC0 described above. The electrical specifications for both DAC0 and DAC1 are given in [Table 8.1](#).

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### SFR Definition 8.1. DAC0H: DAC0 High Byte



### SFR Definition 8.2. DAC0L: DAC0 Low Byte



## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### SFR Definition 8.3. DAC0CN: DAC0 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
DAC0EN	-	-	DAC0MD1	DAC0MD0	DAC0DF2	DAC0DF1	DAC0DF0		00000000
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0		

SFR Address: 0xD4  
SFR Page: 0

Bit7: DAC0EN: DAC0 Enable Bit.  
0: DAC0 Disabled. DAC0 Output pin is disabled; DAC0 is in low-power shutdown mode.  
1: DAC0 Enabled. DAC0 Output pin is active; DAC0 is operational.

Bits6-5: UNUSED. Read = 00b; Write = don't care.

Bits4-3: DAC0MD1-0: DAC0 Mode Bits.  
00: DAC output updates occur on a write to DAC0H.  
01: DAC output updates occur on Timer 3 overflow.  
10: DAC output updates occur on Timer 4 overflow.  
11: DAC output updates occur on Timer 2 overflow.

Bits2-0: DAC0DF2-0: DAC0 Data Format Bits:

000: The most significant nibble of the DAC0 Data Word is in DAC0H[3:0], while the least significant byte is in DAC0L.

DAC0H				DAC0L			
		MSB					LBB

001: The most significant 5-bits of the DAC0 Data Word is in DAC0H[4:0], while the least significant 7-bits are in DAC0L[7:1].

DAC0H					DAC0L		
		MSB					LBB

010: The most significant 6-bits of the DAC0 Data Word is in DAC0H[5:0], while the least significant 6-bits are in DAC0L[7:2].

DAC0H						DAC0L	
		MSB					LBB

011: The most significant 7-bits of the DAC0 Data Word is in DAC0H[6:0], while the least significant 5-bits are in DAC0L[7:3].

DAC0H							DAC0L
		MSB					LBB

1xx: The most significant 8-bits of the DAC0 Data Word is in DAC0H[7:0], while the least significant 4-bits are in DAC0L[7:4].

DAC0H								DAC0L	
		MSB						LBB	

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

**Table 8.1. DAC Electrical Characteristics**
 $V_{DD} = 3.0\text{ V}$ ,  $AV+ = 3.0\text{ V}$ ,  $VREF = 2.40\text{ V}$  ( $REFBE = 0$ ), No Output Load unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
<b>Static Performance</b>					
Resolution			12		bits
Integral Nonlinearity		—	±1.5	—	LSB
Differential Nonlinearity		—	—	±1	LSB
Output Noise	No Output Filter	—	250	—	$\mu\text{Vrms}$
	100 kHz Output Filter		128		
	10 kHz Output Filter		41		
Offset Error	Data Word = 0x014	—	±3	±30	mV
Offset Tempco		—	6	—	ppm/°C
Full-Scale Error		—	±20	±60	mV
Full-Scale Error Tempco		—	10	—	ppm/°C
$V_{DD}$ Power Supply Rejection Ratio		—	-60	—	dB
Output Impedance in Shutdown Mode	DACnEN = 0	—	100	—	k $\Omega$
Output Sink Current		—	300	—	$\mu\text{A}$
Output Short-Circuit Current	Data Word = 0xFFFF	—	15	—	mA
<b>Dynamic Performance</b>					
Voltage Output Slew Rate	Load = 40 pF	—	0.44	—	V/ $\mu\text{s}$
Output Settling Time to 1/2 LSB	Load = 40 pF, Output swing from code 0xFFFF to 0x014	—	10	—	$\mu\text{s}$
Output Voltage Swing		0	—	$VREF - 1\text{LSB}$	V
Startup Time		—	10	—	$\mu\text{s}$
<b>Analog Outputs</b>					
Load Regulation	$I_L = 0.01\text{ mA to }0.3\text{ mA}$ at code 0xFFFF	—	60	—	ppm
<b>Power Consumption (each DAC)</b>					
Power Supply Current ( $AV+$ supplied to DAC)	Data Word = 0x7FF	—	110	400	$\mu\text{A}$



SILICON LABORATORIES

Rev. 1.4

111





Oscilador Interno

**C8051F120/1/2/3/4/5/6/7**  
**C8051F130/1/2/3**

**14. Oscillators**

The devices include a programmable internal oscillator and an external oscillator drive circuit. The internal oscillator can be enabled, disabled, and calibrated using the OSCICN and OSCICL registers, as shown in Figure 14.1. The system clock can be sourced by the external oscillator circuit, the internal oscillator, or the on-chip phase-locked loop (PLL). The internal oscillator's electrical specifications are given in Table 14.1 on page 185.

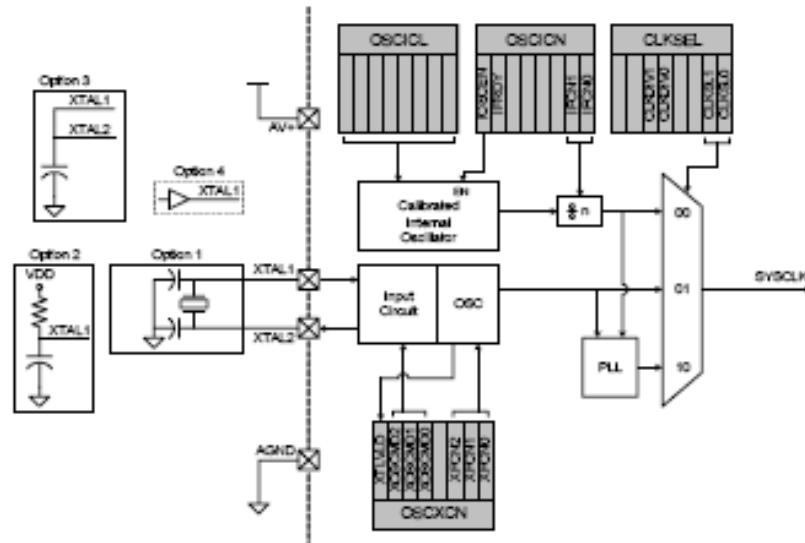


Figure 14.1. Oscillator Diagram

Table 14.1. Oscillator Electrical Characteristics

-40°C to +85°C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Calibrated Internal Oscillator Frequency		24	24.5	25	MHz
Internal Oscillator Supply Current (from V <sub>DD</sub> )	OSCICN.7 = 1	—	400	—	μA
External Clock Frequency		0	—	30	MHz
T <sub>XCH</sub> (External Clock High Time)		15	—	—	ns
T <sub>XCL</sub> (External Clock Low Time)		15	—	—	ns

**14.1. Internal Calibrated Oscillator**

All devices include a calibrated internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be adjusted via the OSCICL register as defined by SFR Definition 14.1. OSCICL is factory calibrated to obtain a 24.5 MHz frequency.

## C8051F120/1/2/3/4/5/6/7

### C8051F130/1/2/3

Electrical specifications for the precision internal oscillator are given in Table 14.1. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN.

#### SFR Definition 14.1. OSCICL: Internal Oscillator Calibration.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Variable

SFR Address: 0x8B  
SFR Page: F

Bits 7–0: OSCICL: Internal Oscillator Calibration Register.  
This register calibrates the internal oscillator period. The reset value for OSCICL defines the internal oscillator base frequency. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

#### SFR Definition 14.2. OSCICN: Internal Oscillator Control

R/W	R	R/W	R	R/W	R/W	R/W	R/W	Reset Value
IOSCEN	IFRDY	-	-	-	-	IFCN1	IFCND	11000000
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	

SFR Address: 0x8A  
SFR Page: F

Bit 7: IOSCEN: Internal Oscillator Enable Bit.  
0: Internal Oscillator Disabled.  
1: Internal Oscillator Enabled.

Bit 6: IFRDY: Internal Oscillator Frequency Ready Flag.  
0: Internal Oscillator not running at programmed frequency.  
1: Internal Oscillator running at programmed frequency.

Bits 5–2: Reserved.

Bits 1–0: IFCN1-0: Internal Oscillator Frequency Control Bits.  
00: Internal Oscillator is divided by 8.  
01: Internal Oscillator is divided by 4.  
10: Internal Oscillator is divided by 2.  
11: Internal Oscillator is divided by 1.

SPI

**C8051F120/1/2/3/4/5/6/7**  
**C8051F130/1/2/3**

**20. Enhanced Serial Peripheral Interface (SPI0)**

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

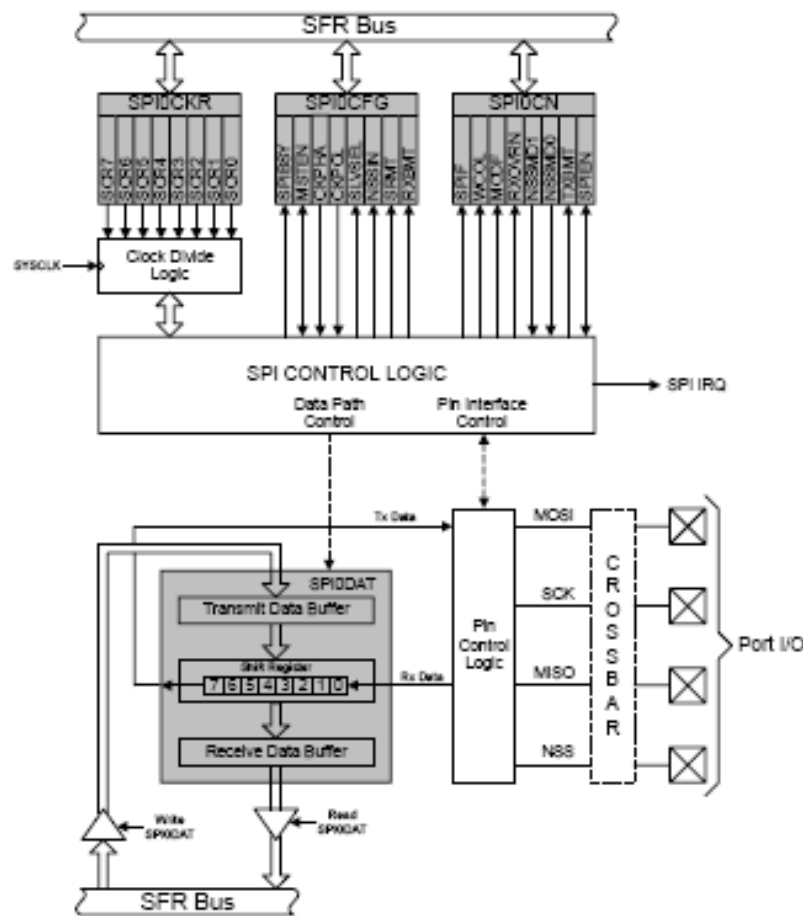


Figure 20.1. SPI Block Diagram

## C8051F120/1/2/3/4/5/6/7

### C8051F130/1/2/3

#### 20.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

##### 20.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

##### 20.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

##### 20.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

##### 20.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 20.2, Figure 20.3, and Figure 20.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section "18. Port Input/Output" on page 235 for general purpose port I/O and crossbar information.

## C8051F120/1/2/3/4/5/6/7

### C8051F130/1/2/3

#### 20.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 20.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 20.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 20.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

**C8051F120/1/2/3/4/5/6/7**  
**C8051F130/1/2/3**

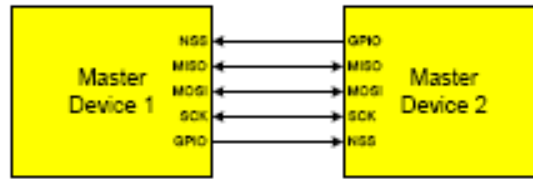


Figure 20.2. Multiple-Master Mode Connection Diagram

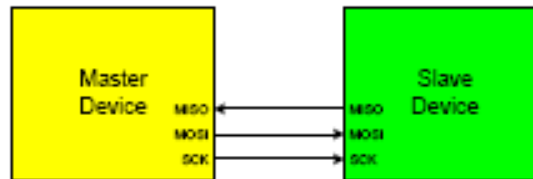


Figure 20.3. 3-Wire Single Master and Slave Mode Connection Diagram

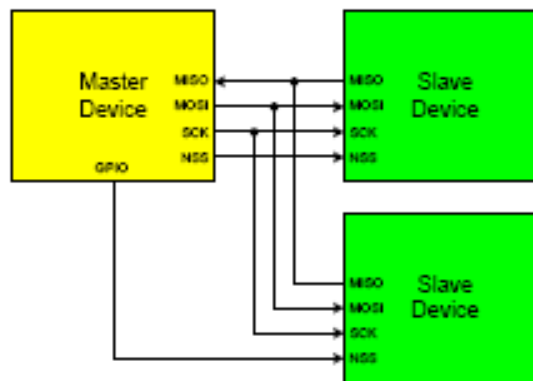


Figure 20.4. 4-Wire Single Master and Slave Mode Connection Diagram

## C8051F120/1/2/3/4/5/6/7

### C8051F130/1/2/3

#### 20.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 20.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 20.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

#### 20.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

*Note that all of the following bits must be cleared by software.*

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### 20.5. Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 20.5. For slave mode, the clock and data relationships are shown in Figure 20.6 and Figure 20.7. Note that CKPHA must be set to '0' on both the master and slave SPI when communicating between two of the following devices: C8051F04x, C8051F06x, C8051F12x/13x, C8051F31x, C8051F32x, and C8051F33x

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 20.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

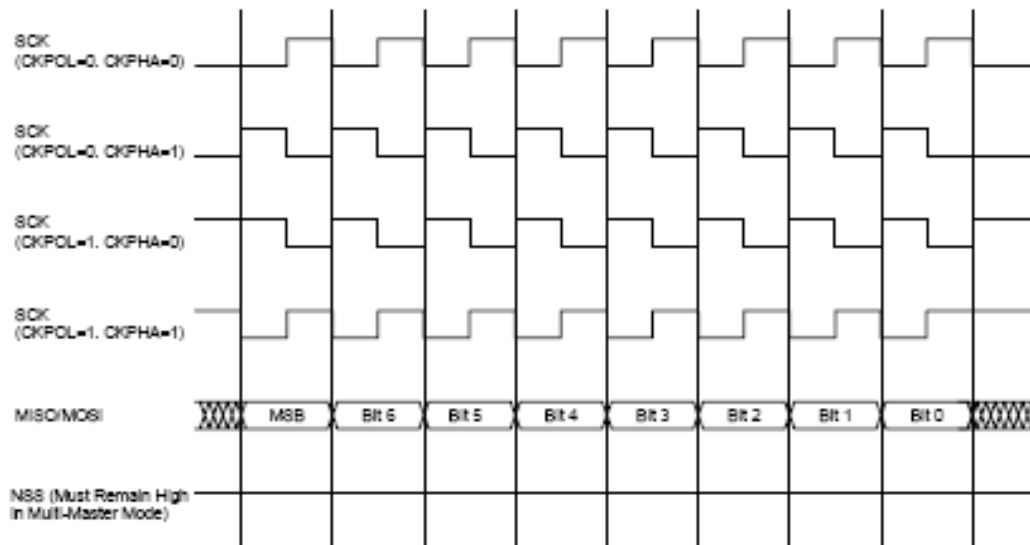


Figure 20.5. Master Mode Data/Clock Timing



**C8051F120/1/2/3/4/5/6/7**  
**C8051F130/1/2/3**

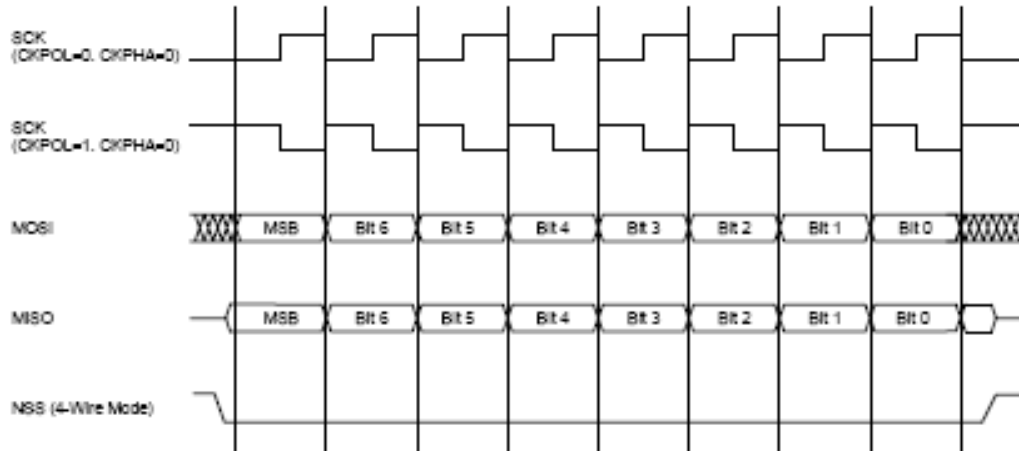


Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)

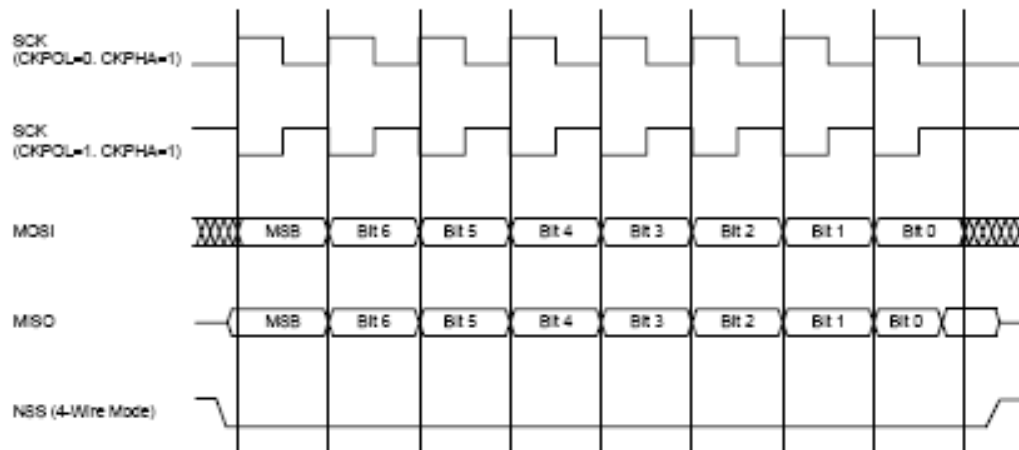


Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)

## C8051F120/1/2/3/4/5/6/7

### C8051F130/1/2/3

#### 20.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 20.1. SPI0CFG: SPI0 Configuration

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
								SFR Address: 0x9A SFR Page: 0
Bit 7:	SPIBSY: SPI Busy (read only). This bit is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).							
Bit 6:	MSTEN: Master Mode Enable. 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.							
Bit 5:	CKPHA: SPI0 Clock Phase. This bit controls the SPI0 clock phase. 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*							
Bit 4:	CKPOL: SPI0 Clock Polarity. This bit controls the SPI0 clock polarity. 0: SCK line low in idle state. 1: SCK line high in idle state.							
Bit 3:	SLVSEL: Slave Selected Flag (read only). This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.							
Bit 2:	NSSIN: NSS Instantaneous Pin Input (read only). This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.							
Bit 1:	SRMT: Shift Register Empty (Valid in Slave Mode, read only). This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. NOTE: SRMT = 1 when in Master Mode.							
Bit 0:	RXBMT: Receive Buffer Empty (Valid in Slave Mode, read only). This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. NOTE: RXBMT = 1 when in Master Mode.							
*Note:	In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 20.1 for timing parameters.							

## C8051F120/1/2/3/4/5/6/7 C8051F130/1/2/3

### SFR Definition 20.2. SPI0CN: SPI0 Control

	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
	SPIF	WCOL	MODF	RXOVRN	NSSMD1	NSSMD0	TXBMT	SPIEN	00000110
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable
									SFR Address: 0xF8 SFR Page: 0
Bit 7:	SPIF: SPI0 Interrupt Flag. This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.								
Bit 6:	WCOL: Write Collision Flag. This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) to indicate a write to the SPI0 data register was attempted while a data transfer was in progress. It must be cleared by software.								
Bit 5:	MODF: Mode Fault Flag. This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). This bit is not automatically cleared by hardware. It must be cleared by software.								
Bit 4:	RXOVRN: Receive Overrun Flag (Slave Mode only). This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. This bit is not automatically cleared by hardware. It must be cleared by software.								
Bits 3–2:	NSSMD1–NSSMD0: Slave Select Mode. Selects between the following NSS operation modes: (See <a href="#">Section "20.2. SPI0 Master Mode Operation" on page 275</a> and <a href="#">Section "20.3. SPI0 Slave Mode Operation" on page 277</a> ). 00: 3-Wire Slave or 3-wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is always an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.								
Bit 1:	TXBMT: Transmit Buffer Empty. This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.								
Bit 0:	SPIEN: SPI0 Enable. This bit enables/disables the SPI. 0: SPI disabled. 1: SPI enabled.								

## C8051F120/1/2/3/4/5/6/7

### C8051F130/1/2/3

#### SFR Definition 20.3. SPI0CKR: SPI0 Clock Rate

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SCR7	SCR8	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x9D  
SFR Page: 0

Bits 7–0: SCR7–SCR0: SPI0 Clock Rate.  
These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for  $0 \leq SPI0CKR \leq 255$

Example: If *SYSCLK* = 2 MHz and *SPI0CKR* = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

$$f_{SCK} = 200kHz$$

#### SFR Definition 20.4. SPI0DAT: SPI0 Data

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x9B  
SFR Page: 0

Bits 7–0: SPI0DAT: SPI0 Transmit and Receive Data.  
The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.