



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES**

**CAMPUS ARAGÓN**

**“Diseño del hardware para una interfaz de control digital utilizando el puerto USB y su implementación en un interruptor controlado inalámbricamente para aplicaciones domóticas”**

**T E S I S**

**Que para obtener el título de:  
Ingeniero Mecánico Electricista**

**P R E S E N T A**

**César Joaquín Rodríguez Cruz**

**DIRECTOR DE TESIS**

**M. EN I. Fernando Macedo Chagolla**

**Ciudad Nezahualcóyotl, Estado de México, 2018**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

<b>OBJETIVO GENERAL:</b> .....	<b>4</b>
<b>INTRODUCCIÓN</b> .....	<b>5</b>
<b>CAPÍTULO 1.- DOMÓTICA</b> .....	<b>7</b>
<b>1.1.- Definición de la domótica</b> .....	<b>7</b>
<b>1.2.- Elementos que componen un sistema de domótica</b> .....	<b>7</b>
<b>1.3.- Topología</b> .....	<b>8</b>
<b>1.4.- Aplicaciones de la domótica</b> .....	<b>10</b>
<b>1.5.- Beneficios generados con el uso de instalaciones domóticas</b> .....	<b>13</b>
<b>1.6.- Importancia de los sensores e interruptores controlados inalámbricamente y su aplicación en construcciones existentes</b> .....	<b>14</b>
<b>CAPÍTULO 2.- EL PUERTO USB</b> .....	<b>16</b>
<b>2.1.- El puerto USB</b> .....	<b>16</b>
<b>2.2.- Como Funciona</b> .....	<b>16</b>
<b>2.3.- Topología</b> .....	<b>17</b>
<b>2.4.- Concentradores o HUBS</b> .....	<b>18</b>
<b>2.5.- Periféricos</b> .....	<b>19</b>
<b>2.6.- Diagrama de capas</b> .....	<b>19</b>
<b>2.7.- Ventajas del puerto USB</b> .....	<b>20</b>
2.7.1.- Una interfaz para muchos dispositivos .....	<b>21</b>
2.7.2.- Configuración automática .....	<b>21</b>
2.7.3.- Facilidad de conexión.....	<b>22</b>
2.7.4.- Múltiples Velocidades.....	<b>23</b>
2.7.5.- Confiabilidad .....	<b>24</b>
<b>2.8.- Usos y límites</b> .....	<b>24</b>
2.8.1.- Distancia .....	<b>24</b>
2.8.2.- Peer-to-peer .....	<b>24</b>
2.8.3.- USB OTG .....	<b>25</b>
2.8.4.- Broadcasting .....	<b>25</b>
<b>2.9.- Componentes de un dispositivo USB</b> .....	<b>25</b>
<b>2.10.- Cables y conectores</b> .....	<b>26</b>
<b>CAPÍTULO 3.- HARDWARE PROPUESTO</b> .....	<b>30</b>
<b>3.1.- Micro-controladores PIC</b> .....	<b>30</b>
3.1.1.- Estructura del PIC .....	<b>30</b>
<b>3.2.- Microprocesador PIC 18F4550</b> .....	<b>31</b>
<b>3.3.- Diagrama a bloques del PIC 18F4550</b> .....	<b>32</b>
<b>3.4.- Descripción Funcional</b> .....	<b>33</b>
<b>3.5.- Aplicaciones prácticas</b> .....	<b>35</b>
<b>3.6.- El Controlador USB</b> .....	<b>36</b>
3.6.1.- Dentro de un controlador USB.....	<b>36</b>
3.6.2.- El transceptor o transceiver.....	<b>36</b>
3.6.3.- El motor de la interfaz serial .....	<b>37</b>

3.6.4.- Buffers.....	37
3.6.5.- Configuración, estado y control de la información .....	38
3.6.6.- Reloj .....	38
3.6.7.- CPU .....	38
3.6.8.- Memoria del programa .....	38
3.6.9.- Memoria de datos .....	39
3.6.10.- Requerimientos del dispositivo .....	39
<b>3.7.- Velocidad.....</b>	<b>39</b>
<b>3.8.- End-Points.....</b>	<b>39</b>
<b>3.9.- Actualizaciones de Firmware.....</b>	<b>40</b>
<b>3.10.- Tipos de controlador.....</b>	<b>40</b>
<b>3.11.- Clases de Dispositivos .....</b>	<b>40</b>
<b>3.12.- Descriptores.....</b>	<b>41</b>
3.12.1.- Dispositivo de Interfaz Humana (HID).....	41
<b>3.13.- Transferencias y comunicación de la interfaz.....</b>	<b>42</b>
3.13.1.- Control de transferencias y tipos de transferencias .....	43
3.13.2.- Manejo de los datos en el BUS .....	43
<b>CAPÍTULO 4.- DISEÑO DEL PRODUCTO .....</b>	<b>46</b>
<b>4.1.- Diseño Conceptual.....</b>	<b>46</b>
<b>4.2.- Diseño del sistema de domótica .....</b>	<b>47</b>
4.2.1.- Modulo de RF .....	48
4.2.2.- Etapa de potencia para controlar una carga conectada a AC .....	49
4.2.3.- Diseño de las placas de circuito impreso .....	50
<b>CAPÍTULO 5.- IMPLEMENTACIÓN .....</b>	<b>54</b>
<b>CONCLUSIONES .....</b>	<b>61</b>
<b>TRABAJO A FUTURO .....</b>	<b>61</b>
<b>BIBLIOGRAFÍA.....</b>	<b>62</b>
<b>ANEXOS.....</b>	<b>63</b>

## **OBJETIVO GENERAL:**

Diseñar un hardware para una interfaz de control digital utilizando el puerto USB e implementarlo en un interruptor controlado inalámbricamente para aplicaciones domóticas

## **OBJETIVOS PARTICULARES:**

- Describir los elementos que componen la domótica.
- Describir el puerto USB y los elementos que lo componen.
- Describir el PIC 18F4550 para la implementación de la interfaz propuesta en esta tesis.
- Diseñar el hardware que sirve como interfaz de control digital utilizando el puerto USB.
- Implementar la interfaz en un interruptor.

## INTRODUCCIÓN

El uso del puerto USB y tecnologías inalámbricas no se encuentran muy difundidas como herramienta para sistemas de control debido a que aún es una tecnología en constante desarrollo y de la cual no hay mucha información. Con la interfaz de control digital se pretende controlar dispositivos haciendo uso de un solo puerto USB, generando así una mayor cantidad de posibilidades que se incrementan a mayor cantidad de puertos disponibles en la una PC, para hacer de cualquier computadora portátil o personal una estación de control que nos brinde señales digitales para activar periféricos acoplados a un sistema, siendo esta capaz de interactuar con sistemas de RF para mayor versatilidad.

En esta tesis se explica y describe el diseño e implementación de una interfaz de control digital haciendo uso del puerto USB que transmite las instrucciones de encendido y apagado de un interruptor inalámbrico para aplicaciones de domótica.

El Capítulo 1 tiene por objetivo describir los elementos que componen la domótica, destacando la importancia y los beneficios obtenidos al ser implementada en construcciones existentes o por existir.

En el Capítulo 2 se describe el puerto USB y su topología, se explican las ventajas de este puerto, así como también sus usos y límites pasando así a la evolución de esta tecnología y describiendo sus formas de comunicación y transferencias de datos.

En el Capítulo 3 se habla de la familia de micro controladores utilizados para comunicación mediante el puerto USB, así como de la familia que se utilizó para desarrollar la interfaz de control digital; se describe el PIC 18F4550 para la implementación de la interfaz propuesta en esta tesis. También se hace mención de los componentes que conforman a un dispositivo USB, explicando cada uno de los principales componentes que le brindan comunicación e interfaz hacia la computadora donde se tenga conectada, de igual manera se explican las clases de dispositivos y tipos de controlares usados con este tipo de tecnología.

En el Capítulo 4 se diseña el hardware que sirve como interfaz de control digital utilizando el puerto USB, desde los planos esquemáticos hasta la técnica utilizada para la realización de la placa de circuito impreso.

En el Capítulo 5 se realiza la implementación de la interfaz del interruptor, mencionando los resultados obtenidos.

Al final se presentan las conclusiones y trabajo a futuro.

# CAPÍTULO 1

# CAPÍTULO 1.- DOMÓTICA

## 1.1.- Definición de la domótica

En su más básico nivel, es un producto o servicio que brinda un nivel de interacción o mensaje en el ambiente del hogar, un evento que será generado sin la intervención directa del usuario, también llamada automatización del hogar.

Es la automatización y control centralizado de aparatos y sistemas eléctricos y electrónicos dentro de un hogar.

Dentro de sus principales aplicaciones contempla aumentar el confort, el ahorro y uso eficiente de la energía y la seguridad.

Es la tecnología de la información aplicada al hogar, utilizando diversos dispositivos para interconectarse y retroalimentar una central que toma decisiones para lograr un resultado final que busca la autonomía, aumentar el confort, el ahorro y uso eficiente de la energía, sin embargo, el encendido de luces y control de electrodomésticos ya ha sido utilizado por años (como el X10), domótica es otro término para el hogar digital, incluyendo las redes, y dispositivos que agregan seguridad.

## 1.2.- Elementos que componen un sistema de domótica

Un sistema de domótica básicamente incluye un CPU, una red y dispositivos periféricos que se controlaran desde el CPU u otros dispositivos portátiles.

Muchos estándares de comunicación coexisten usando el sistema eléctrico de los edificios u hogares como lo es en el uso de la señalización de red (X10, CEBus), periféricos de bajo consumo inalámbrico como ZigBee e INSTEON o en un sistema con un BUS dedicado como EIB que también incluye tecnología inalámbrica. Todos estos estándares requieren un transceiver externo conectado al CPU y la red de computadoras y tienen una configuración estática, como resultado hay un incremento en el manejo de la instalación y costo del sistema de domótica.

**Controlador:** Gestiona el sistema según la programación y la información que reciben. Puede haber uno o más en el mismo sistema, de acuerdo a la arquitectura elegida.

**Actuador:** Ejecuta una orden que recibe del controlador para realizar una acción sobre un aparato o sistema.

**Sensor:** Monitorea el entorno obteniendo información que transmite a un sistema de control, tomando datos para la toma de decisiones.



### 1.3.- Topología

Existen distintas arquitecturas dentro de la domótica que hacen referencia a la estructuración de una red, se clasifican de acuerdo al lugar donde reside el control del sistema.

**Arquitectura centralizada:** El controlador (Figura 1.0) está al centro, envía la información a los actuadores e interfaces según el programa, la configuración y la información que recibe de los sensores, otros sistemas o el usuario.

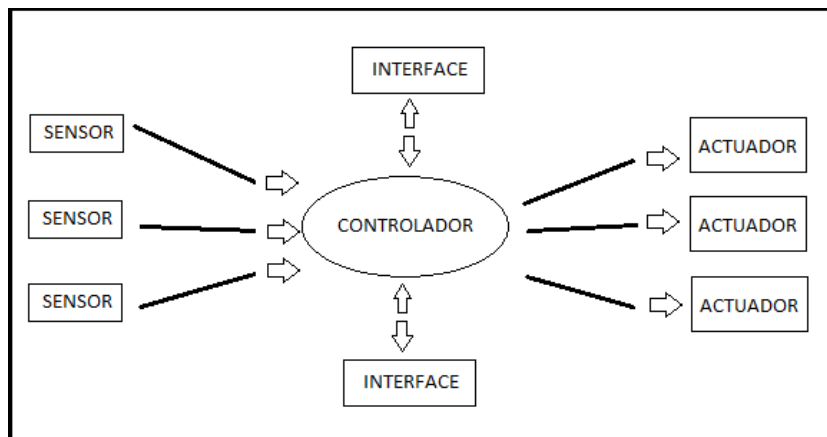


Figura 1.0.- Arquitectura centralizada (elaboración propia).

**Arquitectura descentralizada:** Existen varios controladores interconectados por un bus (Figura 1.1) que envía información entre ellos y a los actuadores e interfaces.

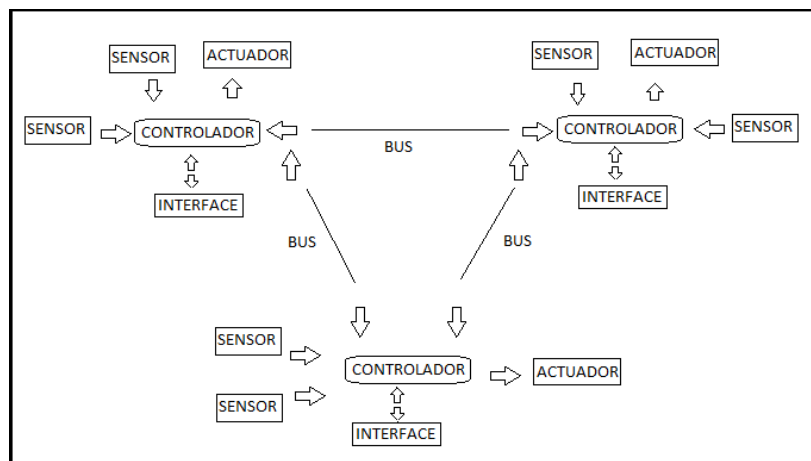


Figura 1.1.- Arquitectura descentralizada (elaboración propia).

**Arquitectura distribuida:** Cada sensor y actuador es también un controlador capaz de actuar y enviar información al sistema según el programa, la configuración, la información que capta por sí mismo y la que recibe de los otros dispositivos.

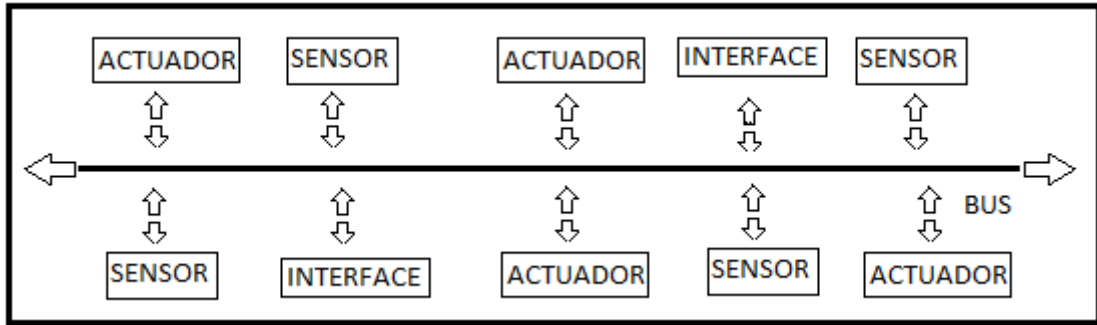


Figura 1.2.- Arquitectura distribuida (elaboración propia).

**Arquitectura híbrida:** Se combinan las anteriores, a la par, se puede disponer de un control central.

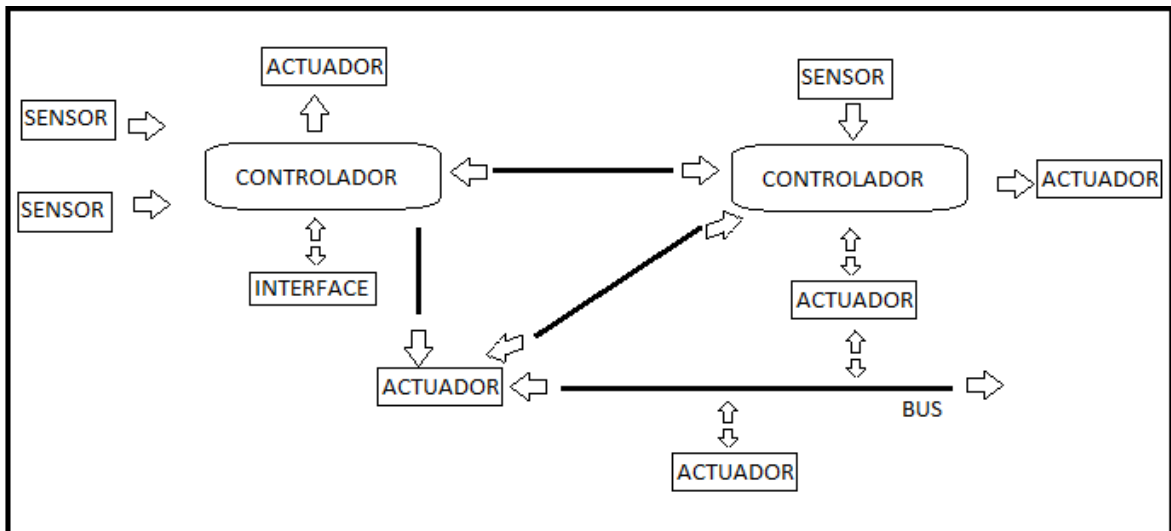


Figura 1.3.- Arquitectura híbrida (elaboración propia).

## 1.4.- Aplicaciones de la domótica

Se han hecho un gran número de intentos de estandarizar la automatización de los hogares estableciendo protocolos de comunicación que han tenido un avance lento en este campo, uno de estos es el X10 que ofrece soluciones económicas en automatización del hogar.

El control de encendido de calentadores, aire acondicionado, cámaras de seguridad, preparación de comida, encendido de TV, estéreos, luces, y otros electrodomésticos son un ejemplo del gran número de dispositivos que se pueden integrar a un sistema de domótica.

Entre las aplicaciones se pueden encontrar:

El control de encendido de calentadores, aire acondicionado, cámaras de seguridad, preparación de comida, encendido de TV, estéreos, luces, y otros electrodomésticos.

En la industria se puede aplicar este sistema de control y automatización a control de líneas de producción, supervisión de equipos, y aplicaciones de inmótica.

Los servicios que ofrece la domótica se pueden agrupar según cinco aspectos principales

- Ahorro energético
- Confort
- Seguridad
- Comunicaciones
- Accesibilidad

### Ahorro energético

En muchos casos no es necesario sustituir los aparatos o sistemas del hogar por otros que consuman menos energía, solo es necesaria una gestión eficiente de los mismos.

Algunas aplicaciones a considerar:

- Climatización: programación y zonificación.
- Gestión eléctrica. Racionalización de cargas eléctricas, desconexión de equipos de uso no prioritario en función del horario en un momento dado.
- Uso de energías renovables.

## **Confort**

El confort conlleva todos los cambios o adecuaciones para mejorar el bienestar y la estancia dentro de una vivienda o centro de convivencia.

Dichas adecuaciones pueden ser de carácter tanto pasivo, como activo o mixtas.

- Iluminación. Apagado general de todas las luces de la vivienda, automatización del apagado y encendido en cada punto donde haya luminarias, regulación de la iluminación según el nivel de luminosidad ambiental.
- Automatización de los sistemas, instalaciones y equipos, para tener un control eficiente de los mismos y facilitar su utilización.
- Integración del video portero a un dispositivo móvil o a la misma unidad de control del sistema de encendido de luces.
- Control vía internet o bluetooth.

## **Seguridad**

Consiste en una red de seguridad encargada de proteger tanto los bienes patrimoniales como la seguridad personal.

- Alarmas de intrusión. Se utilizan para detectar o prevenir la presencia de personas ajenas a la vivienda, entre las cuales se pueden implementar aplicaciones como: la detección de un posible intruso, cierre de ventanas y puertas, simulación de presencia ante la ausencia prolongada de los ocupantes de la vivienda.
- Alarmas de detección de incendios, fugas de gas, fugas de agua, húmedas en áreas no deseadas.
- Alerta médica, tele asistencia, monitoreo en tiempo real de funciones básicas de supervivencia en adultos mayores.
- Acceso remoto a cámaras incluyendo comunicación bidireccional para evitar la entrada de personas ajenas a la vivienda.

## **Comunicaciones**

Son los sistemas o infraestructuras de comunicaciones que posee el hogar, generalmente control de acceso interno, o control remoto desde internet, integrando también sistemas de monitoreo de alarmas.

- Control general de la vivienda tanto externo como interno, control remoto desde cualquier lugar donde se tenga acceso a internet, o control por mandos inalámbricos implementados para equipos móviles.
- Informes de consumo de energía, gas, agua y sus costos.
- Transmisión de alarmas e intercomunicación con autoridades ante la presencia de algún evento.

## **Accesibilidad**

Se enfoca principalmente a las personas con algún tipo de capacidad diferente, para hacer de las tareas comunes algo más sencillas y accesibles de utilizar.

Se pueden implementar controles remotos centralizados para el monitoreo de alarmas y las funciones básicas o principales del hogar, así como el monitoreo del estado de salud de los ocupantes que presentes capacidades diferentes dentro del hogar para avisar oportunamente ante algún malestar que requiera atención médica.

Las características del sistema propuesto plantean un sistema de control de encendido de luces a distancia utilizando la interfaz desarrollada para el puerto USB, siendo este tipo de sistema centralizado al utilizar como mando de control la PC que integrara la GUI y el hardware de transmisión por RF, brindando así al usuario una estación central donde se pueden activar o desactivar distintas funciones de dispositivos en el hogar, dando oportunidad a conectar dispositivos que requieran un solo pulso de encendido o apagado para su funcionamiento sin la necesidad de integrar algún sensor externo.

### **1.5.- Beneficios generados con el uso de instalaciones domóticas**

Los beneficios derivados del uso de instalaciones domóticas son diversos y van desde el confort al ahorrar tiempo en tareas de control de acceso y vigilancia, a continuación se mencionan algunos de ellos:

- Ahorro energético implementando una gestión adecuada de los recursos.
- Potenciación y enriquecimiento de la red de comunicaciones.
- Incremento en seguridad y monitoreo en tiempo real.
- Gestión remota en tiempo real de funciones principales del hogar y sistemas de seguridad.

## **1.6.- Importancia de los sensores e interruptores controlados inalámbricamente y su aplicación en construcciones existentes**

Las tendencias actuales del ahorro y uso eficiente de la energía, exigen cada día nuevas y mejores alternativas para hacer un uso más racional de los recursos, con los avances tecnológicos y científicos siempre se han encontrado nuevas opciones para tales fines.

La domótica como una aplicación de tecnologías de vanguardia dentro de la vivienda es una herramienta bastante amplia y de gran aplicación para la población tanto con fines de confort y bienestar, así como el consecuente ahorro energético al hacer un uso más inteligente de los recursos dentro de una construcción existente.

Con el fin de continuar con las tendencias del ahorro de energía es necesaria la creación de un producto, auxiliado de los principios que componen la domótica, capaz de sustituir elementos tradicionales en las casas, como son los interruptores eléctricos de las luminarias, el desarrollo que se propone es con la finalidad de controlar luminarias de manera inalámbrica para generar mayor confort y seguridad al usuario.

Con el sistema propuesto no solo se pueden controlar luminarias, también se pueden controlar en un sistema cualquier equipo que requiera una conexión a AC logrando así ampliar el marco de aplicación para ahorro en la implementación del sistema a futuro.

La importancia que tienen los sensores e interruptores controlados es vital para la implementación de este proyecto ya que mediante el uso de estos es que se puede tener cargas conectadas al sistema de control que permite procesar la información recibida mediante la interfaz de RF para así activar los equipos conectados a dichos interruptores, pudiendo ser estos equipos desde luminarias a sistemas de calefacción o de seguridad permitiéndonos simular la presencia de personas en el interior de la vivienda ante una ausencia prolongada.

# CAPÍTULO 2



## **CAPÍTULO 2.- EL PUERTO USB**

### **2.1.- El puerto USB**

El Universal Serial Bus (USB) es una interface plug and play entre la computadora y ciertos dispositivos, tales como teclados, mouse, scanner, impresoras, módems, placas de sonido, cámaras, etc.)

Una característica importante es que permite a los dispositivos trabajar a velocidades mayores, en promedio a unos 12 Mbps, esto es más o menos de 3 a 5 veces más rápido que un dispositivo de puerto paralelo y de 20 a 40 veces más rápido que un dispositivo de puerto serial.

### **2.2.- Como Funciona**

Trabaja como interfaz para transmisión de datos y distribución de energía, que ha sido introducida en el mercado de computadoras y periféricos para mejorar las lentas interfaces serie (RS-232) y paralelo. Esta interfaz de 4 hilos, 12Mbps y “plug and play”, distribuye 5V para alimentación, transmite datos y está siendo adoptada rápidamente por la industria informática.

Es un bus basado en el paso de un testigo, semejante a otros buses como los de las redes locales en anillo con paso de testigo. El controlador USB distribuye testigos por el bus. El dispositivo cuya dirección coincide con la que porta el testigo responde aceptando o enviando datos al controlador. Este también gestiona la distribución de energía a los periféricos que lo requieran.

Emplea una topología de estrellas apiladas que permite el funcionamiento simultáneo de 127 dispositivos a la vez. En la raíz o vértice de las capas, está el controlador anfitrión o host que controla todo el tráfico que circula por el bus. Esta topología permite a muchos dispositivos conectarse a un único bus lógico sin que los dispositivos que se encuentran más abajo en la pirámide sufran retardo. A diferencia de otras arquitecturas, USB no es un bus de almacenamiento y envío, de forma que no se produce retardo en el envío de un paquete de datos hacia capas inferiores.

El sistema de bus serie universal USB consta de 3 componentes:

- **Controlador**
- **HUBS o Concentradores**
- **Periféricos**

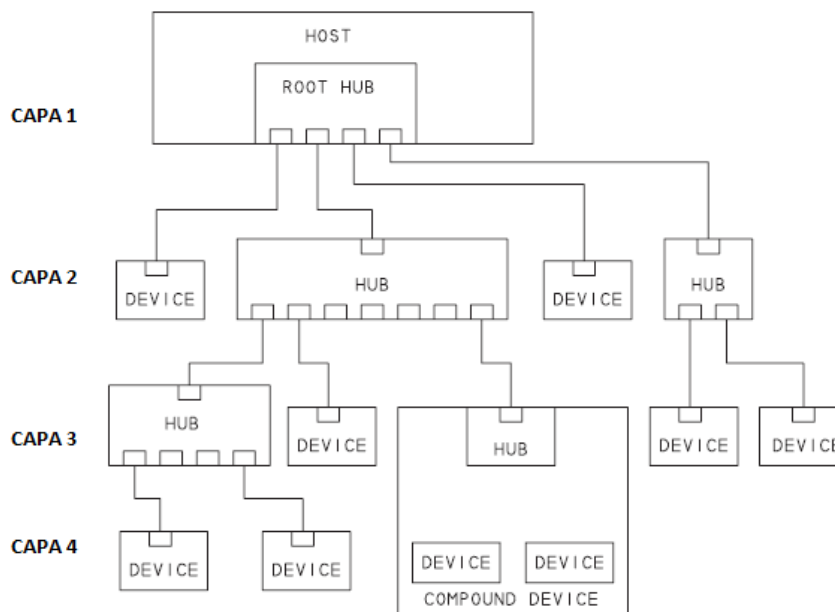
### 2.3.- Topología

La topología, o arreglo de conexiones, en el bus es una estrella en capas. (Figura 2.0). En el centro de la estrella esta un HUB, y cada conexión al HUB es un punto en la estrella. El HUB raíz está en el Host. Un HUB externo tiene un conector upstream (en el lado del host), para comunicarse con el Host y uno más downstream (en el lado del dispositivo) para comunicarse con conexiones internas con dispositivos embebidos.

Un HUB típicamente tiene dos, 4, o 7 puertos. Cuando múltiples HUBS se conectan en serie, se pueden tomar como capas, y así consecutivamente.

La estrella en capas describe únicamente las conexiones físicas. En programación, todo lo que importa es la conexión lógica. Las aplicaciones Host y el firmware del dispositivo no necesitan conocerse o tomarse en cuenta cuando la comunicación pasa a través de un HUB o hasta 5.

Hasta 5 HUBS pueden conectarse en serie con un límite de 127 periféricos y HUBS, incluyendo el HUB Raíz. Sin embargo, el ancho de banda y los límites de tareas pueden prevenir un único controlador de Host para comunicar con varios dispositivos. Para incrementar el ancho de banda con dispositivos USB, algunas computadoras tienen múltiples controladores de Host, cada uno controlando un bus independiente.



**Figura 2.0.-** EL puerto USB utiliza una topología de estrella, donde cada HUB externo tiene un puerto upstream y uno o más para downstream. (Axelson, USB COMPLETE FOURTH EDITION, 2009)

## 2.4.- Concentradores o HUBS

Son distribuidores inteligentes de datos y alimentación, y hacen posible la conexión a un único puerto USB de entre 127 dispositivos. De una forma selectiva reparten datos y alimentación hacia sus puertos descendentes y permiten la comunicación hacia el puerto de retorno o ascendente. Un HUB de 4 puertos, por ejemplo, acepta datos de la computadora para un periférico por su puerto de retorno o ascendente y los distribuye a los 4 puertos descendentes si fuera necesario.

Los concentradores también permiten las comunicaciones desde el periférico hacia la computadora, aceptando datos en los 4 puertos (Figura 2.1) descendentes y enviándolos hacia la computadora por la puerta de retorno.

Además del controlador, la computadora también contiene al concentrador raíz. Este es el primer concentrador de toda la cadena que permite a los datos y a la energía pasar a uno o dos conectores USB de la computadora, y de allí a los 127 periféricos que, como máximo, puede soportar el sistema claro está utilizando varios HUBS para las conexiones.

Un ejemplo de esto sería si la computadora tiene un único puerto USB y a este le conectamos un HUB o concentrador de 4 puertos, la computadora se queda sin más puertos disponibles. Sin embargo, el HUB de 4 puertos permite realizar 4 conexiones descendentes. Conectando otro HUB de 4 puertos a uno de los 4 puertos del primero, habremos creado un total de 7 puertos a partir de un puerto de la computadora. De esta forma, es decir, añadiendo concentradores, la computadora puede soportar hasta 127 periféricos USB.

La mayoría de los concentradores se encontrarán incorporados en los periféricos. Por ejemplo, un monitor USB puede contener un concentrador de 7 puertos incluido dentro de su chasis. El monitor utilizará uno de ellos para sus datos y control y le quedarán 6 para conectar allí otros periféricos.

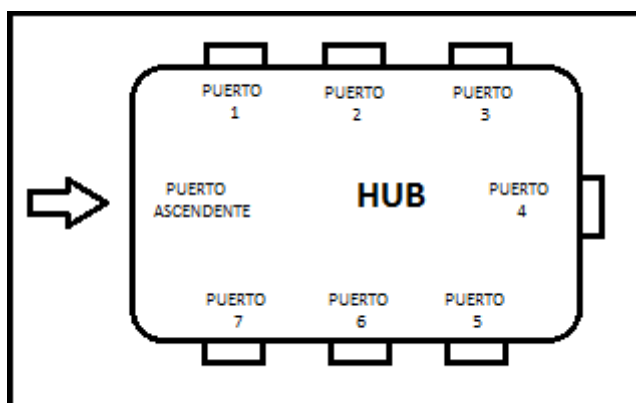


Figura 2.1.- Diagrama de un HUB USB (elaboración propia).

## 2.5.- Periféricos

USB soporta periféricos de baja y media velocidad y actualmente con el nuevo High Speed se integran los dispositivos de alta velocidad que son retro compatibles con puertos que no manejen el estándar High Speed. Empleando dos velocidades para la transmisión de datos de 1.5 y 12 Mbps se consigue una utilización más eficiente de sus recursos. Los periféricos de baja velocidad tales como teclados, ratones, joysticks, y otros periféricos para juegos, no requieren 12 Mbps. Empleando para ellos 1,5 Mbps, se puede dedicar más recursos del sistema a periféricos tales como monitores, impresoras, módems, scanner, etc. que precisan de velocidades más altas para transmitir mayor volumen de datos o datos cuya dependencia temporal es más estricta.

## 2.6.- Diagrama de capas

En el diagrama de capas de la (Figura 2.2) podemos ver cómo fluye la información entre las diferentes capas a nivel real y a nivel lógico.

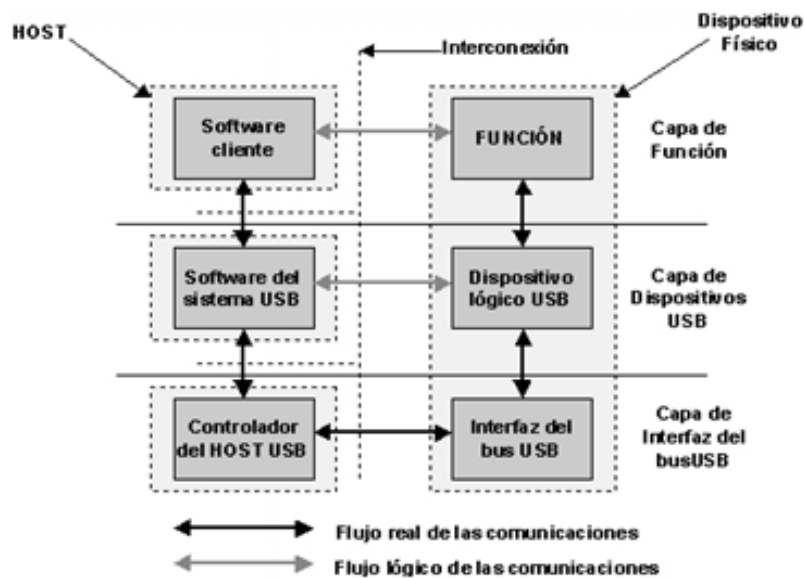


Figura 2.2.- Capas del sistema de comunicaciones USB (Axelson, USB COMPLETE FOURTH EDITION, 2009)

En la figura anterior está materializada la conexión entre el controlador anfitrión o host y un dispositivo o periférico. Este está constituido por hardware al final de un cable USB y realiza alguna función útil para el usuario.

El software cliente se ejecuta en el host y corresponde a un dispositivo USB; se suministra con el sistema operativo o con el dispositivo USB. El software del sistema USB, es el que soporta USB en un determinado sistema operativo y se suministra con el sistema operativo independientemente de los dispositivos USB o del software cliente.

El controlador anfitrión USB está constituido por el hardware y el software que permite a los dispositivos USB ser conectados al anfitrión. Como se muestra en la figura 3, la conexión entre un host y un dispositivo requiere la interacción entre las capas. La capa de interfaz de bus USB proporciona la conexión física entre el host y el dispositivo.

La capa de dispositivo USB es la que permite que el software del sistema USB realice operaciones genéricas USB con el dispositivo.

La capa de función proporciona capacidades adicionales al host vía una adecuada capa de software cliente. Las capas de función y dispositivos USB tienen cada una de ellas una visión de la comunicación lógica dentro de su nivel, aunque la comunicación entre ellas se hace realmente por la capa de interfaz de bus USB.

## **2.7.- Ventajas del puerto USB**

Desde la perspectiva del usuario, los beneficios del USB son que es un puerto fácil de utilizar, rápido y transferencias de datos confiables, bajo costo, y estabilidad en el voltaje de alimentación. La facilidad de uso fue el mayor éxito en el diseño del puerto USB, y el resultado es como tal una interfaz que al usuario le resulta muy cómoda de utilizar.

### 2.7.1.- Una interfaz para muchos dispositivos

El USB es versátil lo suficiente para que casi cualquier función de un periférico estándar para la computadora (Figura 2.3). Aun teniendo diferentes conectores y tipos de cables para cada función de los periféricos, una interfaz sirve para todos debido a su retro compatibilidad.



Figura 2.3.- Dispositivos con interfaz USB (recopilado de internet)

### 2.7.2.- Configuración automática

Cuando un usuario conecta un dispositivo USB a la computadora, el sistema operativo detecta el dispositivo y carga el software del controlador adecuado, muchas veces la primera vez que es conectado un dispositivo, el sistema operativo pregunta al usuario que inserte el disco con el driver adecuado, pero en otras ocasiones la instalación es automática y esto beneficia al usuario porque no es necesario que reinicie el sistema antes de usar el dispositivo USB.

### 2.7.3.- Facilidad de conexión

Una computadora común, tiene múltiples puertos USB, que van desde los 4 como mínimo hasta los 16 en computadoras de gama alta, por otra parte, en las portátiles tenemos desde 1 hasta 4 o más puertos disponibles, pero los HUBS hacen que sea fácil agregar más puertos sin necesidad de una modificación de hardware en el interior de la computadora.

#### Cables más convencionales

Los conectores USB son pequeños y compactos comparados con los conectores usados por otras interfaces como el RS-232. Para asegurar la confiabilidad en el funcionamiento, el USB especifica y define los requerimientos eléctricos para los cables a utilizar en los dispositivos dedicados a esta tecnología. Un segmento de cable puede ser hasta de 5 m dependiendo de la velocidad del bus, lográndose comunicar hasta 9 m máximo para estándares USB 2.0.

#### Conexión en encendido (hot pluggable).

El usuario puede conectar y desconectar un dispositivo USB cuando él quiera, sin importar si el sistema está encendido o no, todo esto sin dañar el dispositivo o la computadora. Para algunos dispositivos como lo son los de almacenamiento será importante si el sistema lo detecta como un disco de lectura o medio de almacenamiento externo retirarlo de forma adecuada evitando dañar las tablas de partición de archivo en dicho dispositivo de almacenamiento.

- **No requiere de configuraciones por parte del usuario (para la mayoría de dispositivos de uso básico).**

Los dispositivos USB no tienen configuraciones seleccionables por el usuario como direcciones de puerto o (IRQ), por lo tanto, los usuarios no tienen que configurar jumpers o utilidades para hacer uso del dispositivo.

- **No es necesaria una fuente de alimentación (algunas veces).**

La interfaz USB incluye fuente de alimentación y líneas de tierra (GND) que proveen un voltaje nominal de +5V desde la computadora o un HUB. Un dispositivo que requiere hasta 500mA (USB 2.0) o 900mA (USB 3.0) puede obtener todo el voltaje desde el bus en lugar de utilizar una fuente de alimentación dedicada. En contraste, los dispositivos que usan otras interfaces deben obtener una fuente de alimentación interna o una fuente de alimentación externa. Aun para algunos dispositivos USB que necesitan energía extra para incrementar la fiabilidad en la información a transferirse sobre todo utilizados para medios de almacenamiento digital, es necesario el proveer de una fuente de alimentación externa al sistema que conforma el dispositivo de almacenamiento USB como lo son las unidades RAID.

## 2.7.4.- Múltiples Velocidades

El USB soporta cuatro velocidades: Super Speed a 5Gbps, High Speed a 480Mbps, Full Speed a 12Mbps y Low Speed a 1.5Mbps. Para poder hacer uso de Super Speed se requiere un controlador host USB 3.0 en la computadora que vaya a ser utilizada como host. Los controladores host USB 2.0 soportan low, full y high speed.

En la (Tabla 2.0) se pueden observar las comparativas del Puerto USB con otros tipos de puertos más comúnmente utilizados.

Interfaz	Tipo	Numero de dispositivos (incluyendo PC) (máx.)	Distancia (máx. pies)	Velocidad (máx. bps)	Uso Típico
USB 3.0	Dual simplex serie	127 (por bus)	9 (típico) (hasta 49 pies con 5 HUBS)	5 G	Almacenamiento masivo, video
USB 2.0	Half duplex serie	127 (por bus)	16 (98 pies con 5 HUBS)	1.5 M, 12 M, 480 M	Teclado, mouse, pen drive, bocinas, impresora, cámara
eSATA	Serie	2 (con multiplicador de puertos 16)	6	3 G	Discos duros
Ethernet	Serie	1024	1600	10 G	Comunicaciones generales de red
IEEE-1394b (Firewire 800)	Serie	64	300	3.2 G	Almacenamiento masivo, video
IEEE-488 (GPIB)	Paralelo	15	60	8 M	Instrumentación
I <sup>2</sup> C	Síncrona serial	40	18	3.4 M	Comunicaciones en micro controladores
MIDI	Serial continua	2(mas con modo flow-through)	50	31.5 k	Música, control en eventos
Puerto paralelo	Paralelo	2(8 con soporte daisy-chain)	10-30	8 M	Impresoras, scanners, discos duros
RS-232 (EIA/TIA-232)	Serial asíncrono	2	50-100	20 k (115 k con algunos tipos de hardware)	Modem, mouse, instrumentación
RS-485(TIA/EIA-485)	Serial asíncrono	32 unidades (algunos chips permiten hasta 256 dispositivos)	4000	10 M	Adquisición de datos y control de sistemas
SPI	Serial síncrono	8	10	2.1 M	Comunicaciones en micro controladores

**Tabla 2.0.-** El USB es más flexible que otras interfaces que ofrecen un uso específico. (Axelson, USB COMPLETE FOURTH EDITION, 2009)



### **2.7.5.- Confiabilidad**

La confiabilidad del puerto USB se debe al hardware y los protocolos utilizados para este tipo de puerto. Las especificaciones de hardware para USB, receptores, y cables aseguran una interfaz eléctricamente silenciosa que elimina la mayor parte del ruido que pudiera causar errores en los datos. Los protocolos USB habilitan detección de errores en los datos recibidos y notifican al mensajero si puede retransmitirlos. El hardware mejora la detección, notificación y retransmisión sin hacer uso de un software o intervención del usuario.

### **2.8.- Usos y límites**

Los límites de esta interfaz se refieren básicamente a la distancia, protocolos de comunicación del tipo peer-to-peer y transmisión en vivo, de igual forma se tiene una limitante debido a versiones viejas de sistemas operativos como lo son desde Windows 95 hacia abajo.

#### **2.8.1.- Distancia**

EL USB fue diseñado como bus para expansión de la computadora de escritorio donde los dispositivos están relativamente cercanos al usuario, otras interfaces, incluyendo RS-232, RS-485, IEEE-1394b, y Ethernet permiten manejar distancias mayores en los cables que los comunican con el sistema.

Para extender la distancia entre un dispositivo y la computadora host, una opción es utilizar el puerto USB a un dispositivo cercado que funcione como puente para retransmitir los datos a una distancia mayor a donde el circuito termina y así su señal es amplificada.

#### **2.8.2.- Peer-to-peer**

Cada comunicación por USB es entre una computadora host y un dispositivo (excepto por una opción nueva introducida con la tecnología USB 3.0). El host es una computadora u otro dispositivo con hardware del tipo host-controller. El host no puede hablar con el otro host directamente, y los dispositivos no pueden hablar con cualquier otro directamente, Otras interfaces como el IEEE-1394, permiten comunicación directa del tipo device-to-device.

### **2.8.3.- USB OTG**

Un dispositivo con función OTG (On-The-Go) puede funcionar como ambos, es decir dispositivo y host, pero con una capacidad limitada como host para poder comunicarse con otros dispositivos, es actualmente utilizado para transferir información básicamente entre tablets y smart phones.

### **2.8.4.- Broadcasting**

El puerto USB no soporta envío de comunicación simultánea a múltiples dispositivos (excepto por el USB 3.0 que incluye un protocolo llamado time stamp packets). El host entonces debe enviar los datos a cada dispositivo individualmente. Si es necesario tener la disponibilidad de broadcasting se pueden utilizar los puertos IEEE-1394 o Ethernet.

## **2.9.- Componentes de un dispositivo USB**

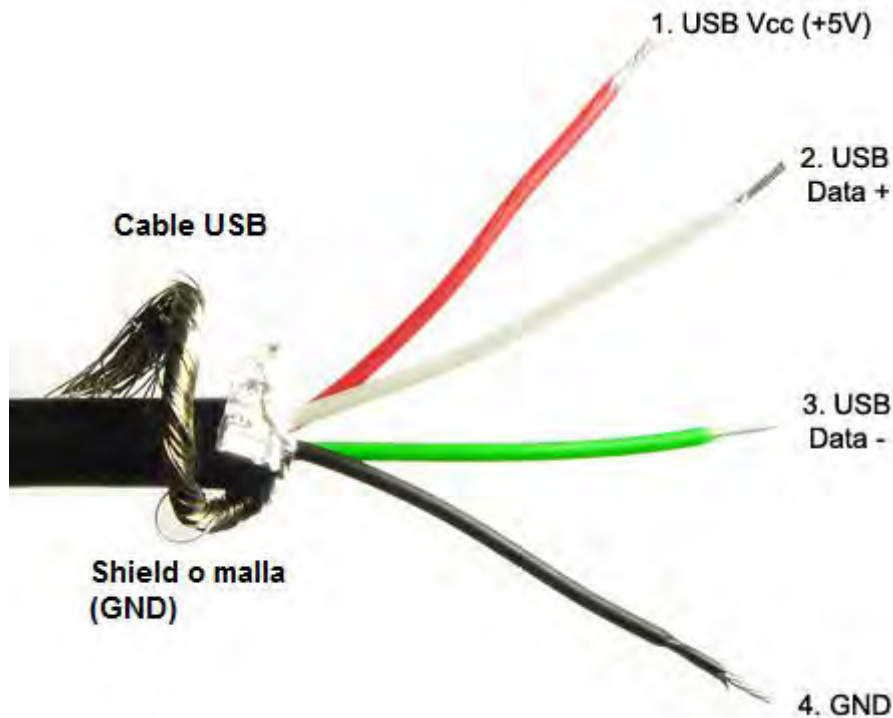
Cada dispositivo tiene la “inteligencia” de detectar y responder a las solicitudes y otros eventos en el puerto. Un micro controlador o un ASIC (Application Specific Integrated Circuit) pueden llevar a cabo estas funciones en un dispositivo. El chip controlador-de dispositivo varía en cómo se implementen las comunicaciones USB y en cómo o que tanto el firmware soporte las comunicaciones requeridas. Algunos controladores requieren un poco más de accesibilidad a los buffers que proveen y reciben datos a través del puerto USB.

Otros requieren de un firmware más complejo para manejar más el protocolo, incluyendo manejo de envío de descriptors al host, configurando valores de manejo de datos y asegurando que los end points regresen apropiadamente los paquetes de hand shake. En general el firmware de bajo nivel no es portable entre chips con diferentes arquitecturas, pero las compañías de chips proveen ejemplos de firmware para tareas comunes y aplicaciones.

- **Transceptor o transceiver**
- **El motor de la interfaz serial**
- **Buffers**
- **Configuración, estado y control de información**
- **Reloj**
- **CPU**
- **Memoria del programa**
- **Memoria de datos**

## 2.10.- Cables y conectores

USB transfiere señales y energía a los periféricos utilizando un cable de 4 hilos, blindado para transmisiones a 12Mbps y no blindado para transmisiones a 1.5Mbps. En la (Figura 2.4) se muestra un esquema del cable, con dos conductores para alimentación y los otros dos para señal, debiendo estos últimos ser trenzados o no según la velocidad de transmisión.



**Figura 2.4.-** Pin out de un cable USB (artechulate.info, 2018)

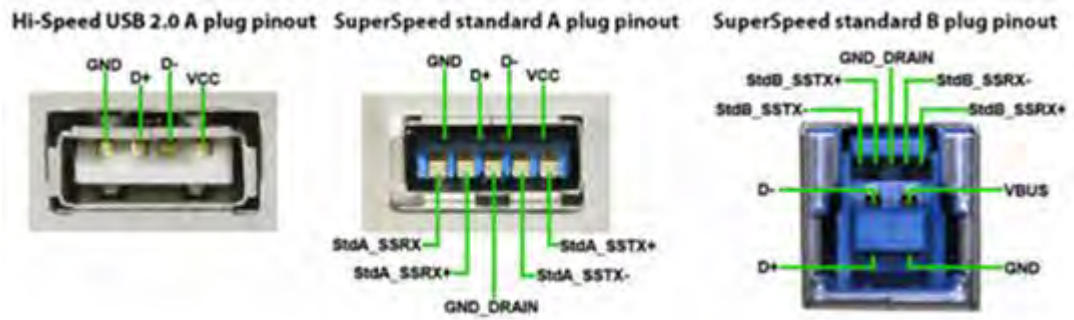
El calibre de los conductores destinados a alimentación de los periféricos varía desde 20 a 26 AWG, mientras que el de los conductores de señal es de 28 AWG. La longitud máxima de los cables es de 5 metros, aunque con la práctica en este proyecto se determinó que es viable un conductor de hasta 10 m.

Por lo que respecta a los conectores hay que decir que son del tipo ficha o conector hembra y macho, y son de dos tipos: serie A y serie B (Figura 2.5). Los primeros presentan los cuatro pines correspondientes a los cuatro conductores alineados en un plano.

Los conectores de la serie B presentan los contactos distribuidos en dos planos paralelos, dos en cada plano, y se emplean en los dispositivos que deban tener un receptáculo al que poder conectar un cable USB. Por ejemplo, impresoras, scanner, y módems.



**Figura 2.5.-** Distintos tipos de conectores USB (Jan, 2006)



**Figura 2.6.-** Pin out de los conectores más comunes usados para comunicación por USB (artechulate.info, 2018)

# CAPÍTULO 3

## **CAPÍTULO 3.- HARDWARE PROPUESTO**

### **3.1.- Micro-controladores PIC**

Las dos principales ventajas de los PIC (micro-controladores) sobre los microprocesadores típicos, son la facilidad de programación y la reducción en los requerimientos de hardware necesario para su implementación, así como el costo requerido para lograr mantener funcional un sistema basado en microprocesadores.

Sin embargo, los dispositivos basados en PIC no son propios para realizar todo tipo de tareas, por ejemplo: una PC sigue requiriendo un microprocesador capaz de direccionar grandes cantidades de memoria y necesita un bus de datos más amplio, por ejemplo de (16 o 64 bits) acompañado de un extendido rango de instrucciones.

Los PICs (micro-controladores) son computadoras en un chip con una memoria ROM y memoria RAM implementados en un encapsulado. Se encuentran además agregadas una serie de funciones como: conversión A/D, comunicación serial I/O, reloj, contadores, etc.

Los PICs pueden acoplarse con interfaces directamente de hardware, por ejemplo: LEDs, módulos Bluetooth, RF, switches, etc.

#### **3.1.1.- Estructura del PIC**

Un microprocesador está constituido por un cierto número de dispositivos lógicos (gates) encapsulados en un único chip de silicón que es capaz de responder a una serie de instrucciones llamadas (programa). El programa define que tareas le son solicitadas al microprocesador a realizar. La operación de un microprocesador puede ser demostrada utilizando un simple modelo que consista en registros de entradas y salidas (I/O), y lectura de memoria ROM, que retendrá el código del programa en un área de almacenamiento llamada Memoria de Acceso Aleatorio (RAM).

Las instrucciones que van a ser transportadas son permanentemente almacenadas en la memoria ROM y ejecutadas una a la vez incrementando el contador del programa.

### 3.2.- Microprocesador PIC 18F4550

A continuación, se describen las características principales de la familia PIC 18 y en la (Figura 3.0) se muestra el Pin out del PIC 18F4550:

- Arquitectura RISC avanzada Harvard: 16 bit con 8 bit de datos
- 77 Instrucciones
- Desde 18 a 80 pines
- Hasta 64Kbytes de programa (hasta 2Mbytes en ROM less)
- Multiplicador Hardware 8x8
- Hasta 3968 bytes de RAM y 1Kbyte de EEPROM
- Frecuencia máxima de reloj 40Mhz, hasta 10MIPS
- Pila de 32 niveles
- Múltiples fuentes de interrupción
- Periféricos de comunicación avanzados (CAN y USB)

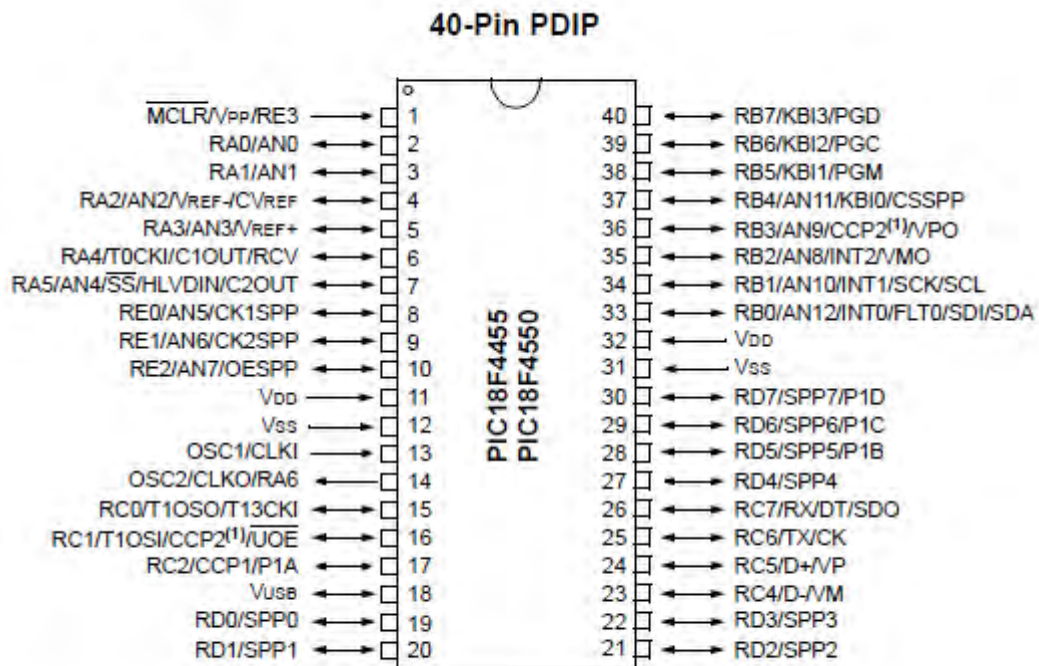


Figura 3.0.- Pin out del PIC 18F4550 (DIP) (Technology, 2004)



### 3.3.- Diagrama a bloques del PIC 18F4550

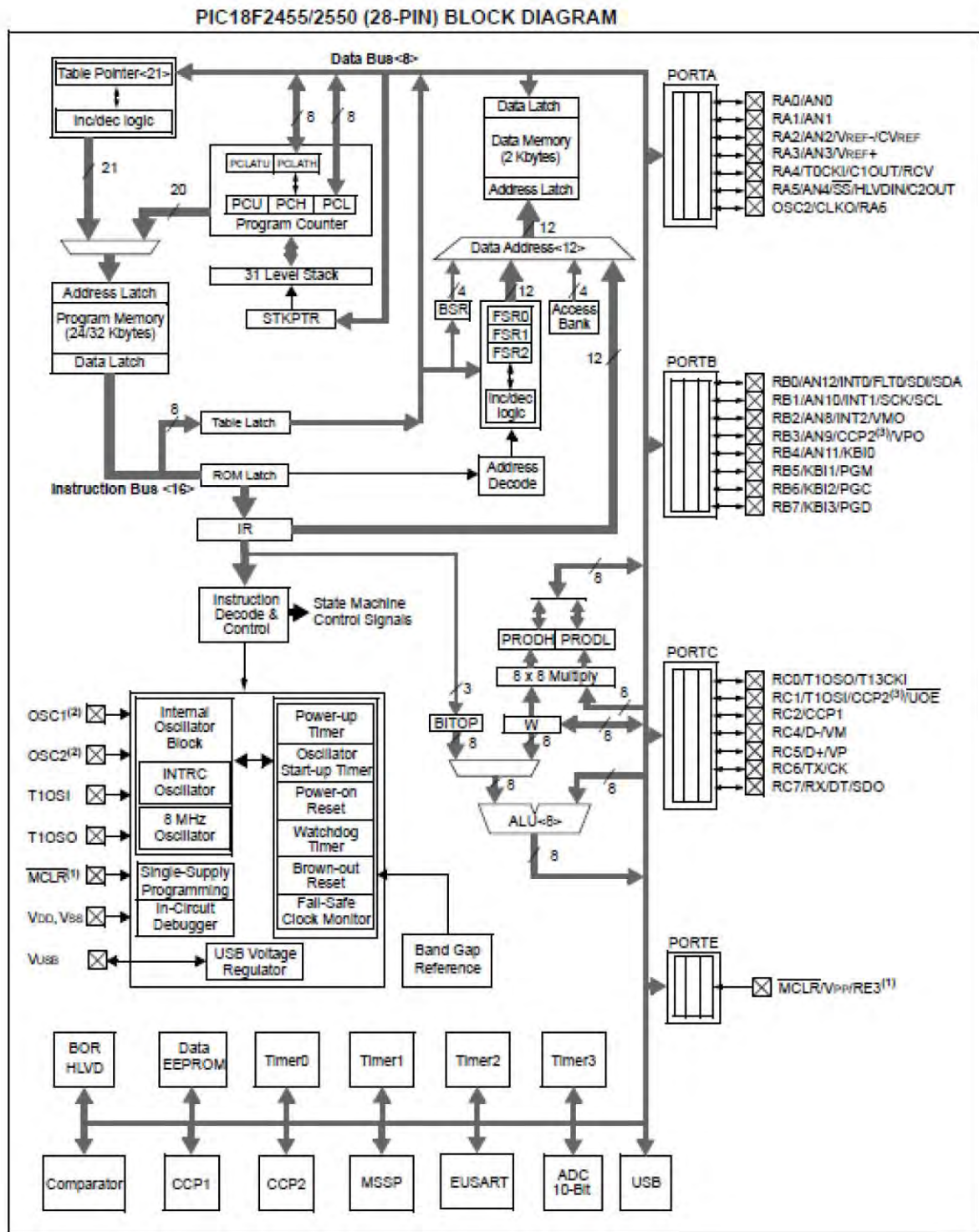


Figura 3.1.- Diagrama a bloques del PIC 18F4550 (Technology, 2004)

### 3.4.- Descripción Funcional

A continuación, se mencionan los principales rasgos funcionales del microcontrolador PIC 18F4550:

- **Memoria de programa**

Está constituida por una memoria flash interna de 32.768 bytes y almacena instrucciones y constantes/datos, puede ser escrita/leída mediante un programador externo o durante la ejecución del programa mediante unos punteros.

- **Memoria RAM**

La memoria RAM de datos, la memoria SRAM interna cuenta con 2048 bytes en la que están incluidos los registros de función especial, esta almacena datos de forma temporal durante la ejecución del programa y puede ser de lectura/escritura en tiempo de ejecución mediante diversas instrucciones.

- **Memoria EEPROM**

Es una memoria no volátil de 256 bytes y almacena datos que se deben conservar aun en ausencia de voltaje de alimentación, puede ser de lectura/escritura en tiempo de ejecución a través de registros.

- **Memoria de configuración**

Memoria en la que se incluyen los bits de configuración (12 bytes de memoria flash) y los registros de identificación (2 bytes de memoria de solo lectura).

- **MEMORIA DE PROGRAMA**

El PIC18F4550 dispone una memoria de programa de 32.768 bytes (0000H-7FFFH). Las instrucciones ocupan 2 bytes (excepto CALL, MOVFF, GOTO y LSFR que ocupan 4). Por lo tanto, la memoria de programa puede almacenar hasta 16.384 instrucciones.

La operación de lectura en posición de memoria por encima de 7FFFH da '0' como resultado (equivalente a la instrucción NOP).

- **RELOJ**

Otra parte importante de nuestro microcontrolador es el reloj, en este microcontrolador el reloj puede ser externo micros y también se puede hacer uso de su oscilador interno, el circuito del oscilador se rige como se muestra en la (Figura 3.2).

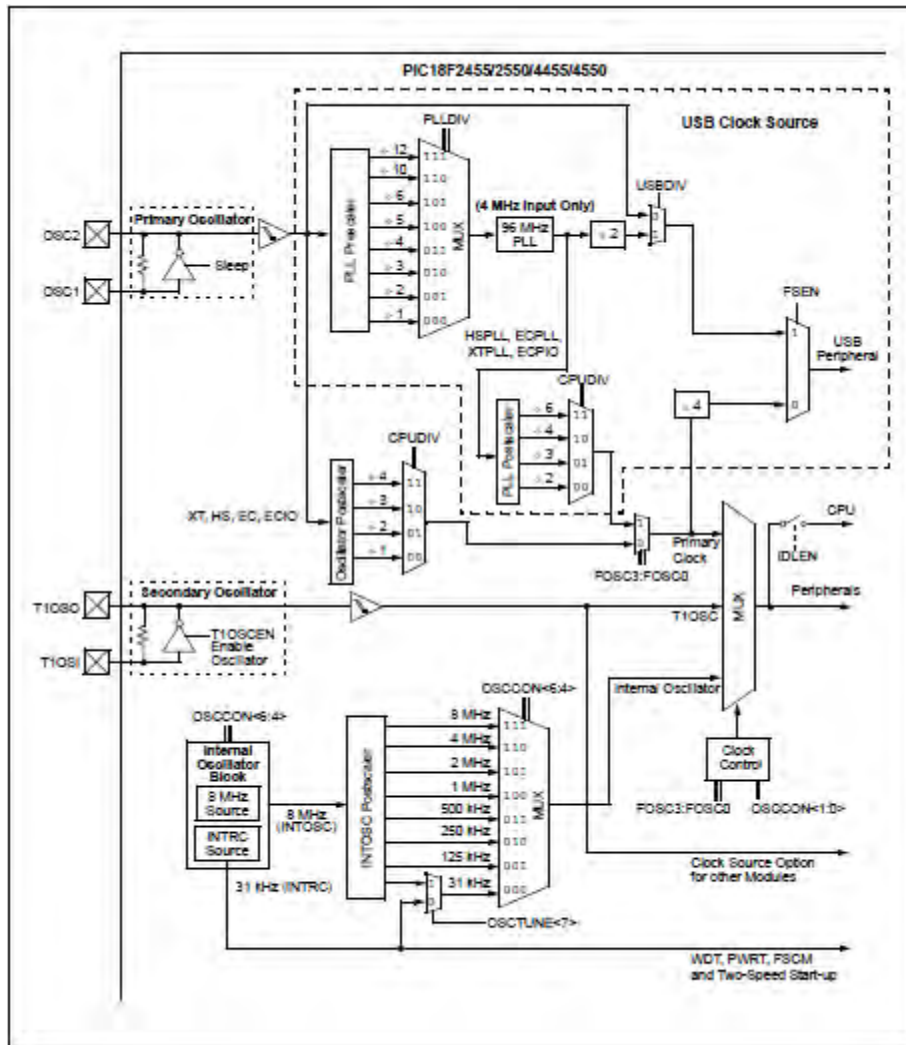


Figura 3.2.- Diagrama a bloques del Reloj del PIC18F4550 (Technology, 2004)

### 3.5.- Aplicaciones prácticas

El campo de aplicaciones prácticas de los PIC en general es muy amplio ya que los podemos encontrar en aparatos electrodomésticos, el automóvil, periféricos de equipo de cómputo, sistemas de domótica, instrumentación, aeronaves tanto de uso civil como militar, drones, equipos que requieran de un sistema que los controle y dote de ciertas decisiones a tomar basadas en una adquisición de datos tomada desde algún transductor o instrucción que le sea enviada a través de puertos de comunicación.

En los aparatos electrodomésticos los podemos encontrar en microondas, refrigeradores, televisores, grabadores y reproductores multimedia, etc.

En el automóvil se pueden encontrar desempeñando funciones de control de frenado como lo es en los ABS, sistema de fuel injection, controles de luces automáticos, apertura y encendido a distancia, etc.

En los sistemas de domótica se pueden encontrar en los controles de acceso, sistemas de alarma, aire acondicionado, alarmas contra incendio, encendido remoto de luces, sistemas de alimentación para mascotas, etc.

### **3.6.- El Controlador USB**

Reside dentro del PC y es responsable de las comunicaciones entre los periféricos USB y la CPU del PC. Es también responsable de la admisión de los periféricos dentro del bus, tanto si se detecta una conexión como una desconexión. Para cada periférico añadido, el controlador determina su tipo y le asigna una dirección lógica para utilizarla siempre en las comunicaciones con el mismo.

Si se producen errores durante la conexión, el controlador lo comunica a la CPU, que, a su vez, lo transmite al usuario. Una vez que se ha producido la conexión correctamente, el controlador asigna al periférico los recursos del sistema que éste precise para su funcionamiento. El controlador también es responsable del control de flujo de datos entre el periférico y la CPU.

Un controlador típico USB 2.0 contiene un transceiver, un motor de interfaz serial, buffers para mantener los datos USB, registros que guardan la configuración, estado, e información de control relacionada con las comunicaciones.

#### **3.6.1.- Dentro de un controlador USB**

Un controlador típico de USB 2.0 contiene un módulo transceptor, un motor de interface serial, buffers para manejar los datos, y registros que almacenan la configuración, estado, y control de la información relacionada con las comunicaciones del puerto.

#### **3.6.2.- El transceptor o transceiver**

El transceiver es la interfaz de hardware entre el conector de los dispositivos y los circuitos que controlan las comunicaciones USB. El transceiver esta típicamente embebido en un chip, pero algunos controladores permiten la interfaz como si fueran un transceiver externo.

### 3.6.3.- El motor de la interfaz serial

Los circuitos que hacen la interfaz al transceiver forman una unidad llamada SIE (Serial Interface Engine). La SIE maneja, envía y recibe los datos en las transacciones. La SIE no interpreta o usa los datos, pero solo pone los datos proveídos en el bus y almacena cualquier dato recibido. Una típica SIE hace lo siguiente:

- Detecta los paquetes entrantes
- Envía paquetes
- Detecta y genera Start-of-Packet, EOP, Reset y reinicializa la señalización
- Codifica y decodifica los datos del bus usando NRZI
- Checa y genera valores CRC
- Checa y genera ID de Paquetes
- Convierte entre datos seriales USB y datos en paralelo en registros o memoria.

### 3.6.4.- Buffers

Los controladores USB usan buffers para almacenar los datos recibidos y los datos que están listos para transmitirse al bus. EN algunos chips, como el NET2272, el CPU accesa a los buffers leyendo y escribiendo en los registros, como otros, como la serie de chips EZ-USB, que reservan una porción de memoria para los buffers.

Para habilitar transferencias rápidas, algunos chips tienen dobles buffers que pueden guardar dos sets completos de datos en cada dirección. Mientras un bloque está transmitiendo, la firma puede recibir en el siguiente bloque de datos en el otro buffer mientras que los datos están listos para pasar al primer bloque y terminar la transmisión. En la dirección de recepción, un buffer extra habilita una nueva transacción de datos para que llegue antes de que el firmware reciba datos de una transacción previa. El hardware automáticamente switchea, o recibe y regresa, datos entre los dos buffers, algunos controladores soportan buffers cuádruples.

### **3.6.5.- Configuración, estado y control de la información**

Un chip de controlador típicamente tiene registros que mantienen la información acerca de que end points están habilitados, el número de bites recibidos, el número de bites listos para transmitir, estado de suspensión, información de chequeo de errores, y otros estados y control de la información. El número de registros, sus contenidos, y como los registros acezan varía dependiendo de la arquitectura del chip utilizado. Estas diferencias son una de las razones del porque el firmware de bajo nivel para comunicaciones USB típicamente no es portable entre familias de chips.

### **3.6.6.- Reloj**

Las comunicaciones requieren una fuente que delimite el tiempo, típicamente proveído por un cristal oscilador, debido a que velocidades bajas en USB permiten más variación en la velocidad del reloj, los dispositivos de bajas velocidades pueden algunas veces usar un resonador cerámico menos costoso, Algunos controladores tienen un oscilador on-chip que quiere decir está integrado en el chip y no requiere una fuente de tiempo externo.

### **3.6.7.- CPU**

Un CPU on-chip en un micro controlador puede estar basado en arquitectura de propósito general como el 8051, o el CPU puede tener una arquitectura desarrollada específicamente para aplicaciones USB. Un controlador de interfaz únicamente USB puede comunicarse con cualquier CPU que tenga una interfaz compatible.

### **3.6.8.- Memoria del programa**

La memoria del programa retiene el código que el CPU ejecuta, incluyendo código para las comunicaciones USB y cualquiera otra tarea que de las que el chip sea responsable. Esta memoria puede estar en el micro controlador o en un chip separado.

El almacenamiento del programa puede usar ROM, memoria FLASH, EEPROM, EPROM, o RAM, todos los tipos de memoria excepto la RAM son no volátiles: la memoria retiene sus datos después de que el sistema es apagado.

Otro nombre para el código almacenado en la memoria del programa es FIRMWARE. El término sugiere que la memoria es no volátil y que no es fácilmente cambiado como un código de programa que puede ser cargado en la RAM, editado, y re-salvado en disco.

### **3.6.9.- Memoria de datos**

La memoria de datos provee almacenamiento temporal durante la ejecución del programa. El contenido de memoria de datos puede incluir datos recibidos por el puerto, y datos que van a ser enviados al puerto, valores para ser usados en cálculos, o cualquier otra cosa que el chip necesite para mantener un estado. La memoria de datos es RAM.

### **3.6.10.- Requerimientos del dispositivo**

Antes de seleccionar un dispositivo para algún proyecto, es recomendable considerar las prestaciones que tendremos con el dispositivo a desarrollar, tomando en cuenta distancias de transmisión de los datos, cantidad de información que se va a transmitir y las características de expansión y actualización que requiera el desarrollo para futuras expansiones.

### **3.7.- Velocidad**

El data rate del dispositivo depende de las velocidades soportadas en el bus del dispositivo, el tipo de transferencia y que tan ocupado estará en bus para manejar los datos. Si el producto requiere solo manejar interrupciones a low speed y control de transferencias, un chip low speed puede ahorrarnos algo de dinero en el diseño, componentes y cables. Pero los dispositivos low speed pueden transferir únicamente hasta 8 bytes de datos por transacción y que la especificación USB limita el ancho de banda garantizado para una interrupción de end point a 800 bytes por segundo, mucho menos que la velocidad del bus de 1.5Mbps.

Comparado con low y full speed, el diseño de las placas de circuito para dispositivos high speed es más crítico y puede agregar mayor costo al producto que se busca diseñar. Si es posible, los dispositivos que soportan high speed deben también soportar full speed, entonces deberán por lo tanto trabajar con velocidades de low speed 1.x en HUBS y hosts.

### **3.8.- End-Points**

Cada dirección de end point soporta un tipo de transferencia y dirección. Un dispositivo que solo usa transferencias de control necesita solo el end point por default. Las transferencias de interrupción, volcado, o asíncronas requieren direcciones adicionales de end point. No todos los chips soportan todos los tipos de transferencias. No todos los tipos de controladores soportan el número máximo posible de direcciones de end point, pero solo algunos dispositivos necesitan el máximo.



### **3.9.- Actualizaciones de Firmware**

Las actualizaciones de firmware son necesarias para corregir el funcionamiento del dispositivo y en muchos casos optimizarlo y hacerlo compatible con más ambientes de software y hardware. Normalmente las actualizaciones se realizan almacenando el código del programa en una memoria flash, EEPROM, o una memoria RAM cargada desde el host conectado.

Existe una especificación para la actualización del firmware de un dispositivo USB que está regulada por la USB.org que define un protocolo acerca de cómo cargar el firmware en el dispositivo host.

### **3.10.- Tipos de controlador**

Si el dispositivo coincide con una clase de las soportadas por el sistema operativo que el host USB utilice, no será necesario escribir u obtener el controlador para el dispositivo. Por ejemplo, las aplicaciones pueden acceder a dispositivos tipo HID-class utilizando funciones API estándar para comunicarse con los controladores HID incluidos en Windows.

Algunas compañías de chips proveen drivers genéricos que se pueden utilizar al momento de desarrollar una aplicación, este se puede utilizar para intercambiar datos con los dispositivos. Cypress Semiconductor, Microchip Technology y Silicon Laboratories todos ellos tienen controladores de propósito general. Los dispositivos desarrollados para sistemas operativos Windows, también tienen la opción de utilizar el controlador genérico Win USB de Microsoft.

### **3.11.- Clases de Dispositivos**

La mayoría de los dispositivos USB tienen mucho en común con otros dispositivos que realizan funciones similares. Los mouses envían datos acerca de los movimientos del mouse, los botones y los clicks. Las memorias flash o “memorias USB” por igual envían y reciben datos de cualquier tipo que quiera ser almacenado e informan al sistema si la transferencia fue realizada, las impresoras reciben datos para imprimir e informan al host cuando se les terminó el papel.

Cuando muchos dispositivos proveen o solicitan servicios similares, tiene sentido definir protocolos para todos los dispositivos a utilizar. Una especificación de clase “class specification” sirve como guía para los programadores acerca de cómo escribir el firmware del dispositivo o los controladores para el host. Los controladores incluidos en un sistema operativo eliminan la necesidad de los vendedores de proveer controladores con sus dispositivos.

Cuando un dispositivo de una clase soportada tiene características o habilidades no incluidas en un controlador estándar que define una clase de dispositivo, el dispositivo algunas veces necesita usar un controlador diseñado para agregar las capacidades y características distintas con las que cuenta.

Las clases no soportadas actualmente por un sistema operativo se espera siempre que vayan siendo soportadas con el paso del tiempo cuando salen actualizaciones del sistema operativo que integran una mayor base de datos de controladores a los archivos del sistema, de igual manera un firmware que es compatible con una especificación de clase de dispositivo igual se espera que con las actualizaciones posteriores del sistema sean compatibles.

### **3.12.- Descriptores**

Como en cualquier dispositivo USB, los descriptores HID le dicen al host que necesita saber para poder comunicarse con el dispositivo. El host aprende acerca de la interfaz HID durante la enumeración enviando una solicitud de Descriptor para la configuración que contiene la interfaz HID. Un descriptor de interfaz especifica dicha interfaz HID. Un descriptor de clase HID especifica el número combinado o reporte y descriptores físicos soportados por la interfaz. Durante la enumeración, el driver de la HID solicita un descriptor de reporte y cualquier otro descriptor físico.

En PICBASIC PRO, las tablas de descriptores están en código ensamblador. Cada tabla es una lista de valores cada una precedida por una instrucción `retlw`, que pone el valor de la literal que siguen en el registro de trabajo y regresa al código de inicio. No es necesario saber código ensamblador para componer un descriptor, solo es necesario iniciar un ejemplo y editar los valores que se necesiten.

#### **3.12.1.- Dispositivo de Interfaz Humana (HID)**

En el descriptor de interface, `bInterfaceClass = 03h` se utiliza para identificar la interfaz como HID. Otros campos que contienen información de la especificación HID en el descriptor de la interfaz son los campos de `bInterfaceSubclass` y `bInterfaceProtocol`, que pueden especificar una interfaz de arranque.

Si `bInterfaceSubclass = 01h`, el dispositivo soporta una interfaz de arranque, Una HID con una interfaz de arranque puede comunicarse con el host incluso cuando el host no ha cargado sus controladores de HID. Esta situación puede ocurrir cuando la computadora inicia directamente de MS-DOS o cuando se visualizan las ventanas de configuración cuando se accede desde el boot, o cuando se utiliza Windows en modo a prueba de fallos para solucionar problemas del sistema.

Un teclado o un mouse con una interfaz de arranque pueden usar un protocolo simplificado soportador por el BIOS en muchos hosts. El BIOS carga de la ROM u otra memoria no volátil en el inicio y si está disponible en cualquier modo del sistema operativo.

La especificación HID define los protocolos de la interfaz de arranque para teclados y mouse, si un dispositivo tiene una interfaz de arranque, el `InterfaceProtocol` indica si el HID soporta un teclado con la función (01h) o un mouse (02h).

### **3.13.- Transferencias y comunicación de la interfaz**

Para enviar o recibir datos, un host debe iniciar una transferencia USB, cada transferencia usa un formato definido de envío de datos, direccionamiento, información, bits de detección de errores, e información de control y estado. El formato varía con el tipo de transferencia y la dirección.

Cada comunicación USB (con excepción del USB 3.0) es entre el host y el dispositivo. El host controla el tráfico del bus, y el dispositivo responde a las comunicaciones del host. Un end point es el buffer del dispositivo que almacena los datos recibidos o datos transmitidos. Cada end point tiene un número, dirección, y un máximo número de bits de datos que el end point puede enviar o recibir en una transacción.

Cada transferencia USB consiste en una o más transacciones que pueden llevar datos a o de un end point. Una transacción USB 2.0 comienza cuando el host envía paquetes de token en el bus. El paquete de token contiene el número de destino del end point y la dirección. Un paquete de token entrante pide un paquete de datos de un end point, un paquete de salida de token precede el paquete de datos del host. Como complemento en los datos, cada paquete de datos contiene bits de verificación de errores y un identificador de paquete (Packet ID PID) con un valor de datos secuencial. Muchas transacciones también tienen un paquete hand shake donde el receptor de los datos reporta si la transacción fue satisfactoria o presentó una falla. Para las transacciones USB 3.0, los tipos de paquetes y protocolos difieren, pero las transacciones contienen direccionamiento similar, así como verificación de errores, y valores de secuencia de datos.

Las comunicaciones en el puerto USB pueden agruparse en dos categorías generales: comunicaciones que ayudan a identificar y configurar el dispositivo y las comunicaciones que se encargan del propósito del dispositivo. Durante la enumeración, el host aprende acerca del dispositivo y solicita una configuración que prepara al dispositivo para optimizar su función.

Cuando la enumeración está completa, el host puede enviar una petición de datos como sea necesaria para hacer funcionar el dispositivo de acuerdo a su propósito. Una especificación del vendedor también puede definir las peticiones.

El control de transferencias tiene que pasar por tres estados: Configuración, Datos (opcional), y estado. La configuración contiene la petición. Cuando está presente, la etapa de datos contiene datos del dispositivo host, dependiendo de la petición. La etapa de estado contiene información acerca de si la transferencia se llevó a cabo. En una transferencia de control de lectura, el dispositivo envía datos a la etapa de Datos. En una transferencia de control de escritura, el host envía datos a la etapa de Datos, o la etapa de datos está ausente.

Los otros tipos de transferencia no tienen etapas definidas. Por lo tanto, el software de alto nivel define como interpretar los datos.

### **3.13.1.- Control de transferencias y tipos de transferencias**

El puerto USB soporta cuatro tipos de transferencias: control, transferencia masiva, interrupción y asíncrona.

En la transferencia de control, el host envía una solicitud definida al dispositivo. Cuando el dispositivo es conectado, el host utiliza transferencias de control para solicitar una serie de estructuras de datos llamadas descriptores al dispositivo. Los descriptores proporcionan información acerca de las capacidades del dispositivo y ayudan al host a decidir que driver asignar al dispositivo, Una especificación del vendedor también puede definir las solicitudes.

### **3.13.2.- Manejo de los datos en el BUS**

El host gestiona las transferencias en el BUS. Un controlador USB 2.0 maneja el tráfico dividiendo el tiempo en tramas de 1 ms en low speed y en full speed y 125us para high speed. El host aloja una porción de cada (micro) trama para cada transferencia. Cada (micro) trama empieza con un tiempo de referencia de Inicio de trama. Con USB 3.0 el BUS no usa en inicio de transferencia, pero maneja las transferencias en Super Speed con intervalos de 125us. Un host USB 3.0 también puede enviar paquetes cada que todos los puertos Super Speed no estén en un estado de low -power.

Cada transferencia consiste en una o más transacciones, Las transferencias de control siempre tienen múltiples transacciones porque tienen múltiples etapas, cada una consistente de una o más transacciones.

Dependiendo de cómo el host gestiona las transacciones y la velocidad de respuesta del dispositivo, las transacciones en una transferencia puede que todas estén en una (micro) trama sencilla o en un intervalo del BUS, o las transacciones sean separadas en múltiples (micro) tramas o intervalos del BUS.

Cada dispositivo tiene una dirección única asignada por el host, y todos los datos que viajan al host o desde este. Excepto por la señalización wakeup, cada dispositivo envía esto en respuesta si recibe un paquete enviado por el HOST. Debido a que múltiples dispositivos pueden compartir una ruta de datos en el BUS, cada transacción USB 2.0 incluye una dirección de dispositivo que identifica el destino de la transacción.

Los dispositivos Super Speed pueden enviar información de control y estado al HOST sin esperar a que el host solicite la información. Cada paquete de datos Super Speed y paquete de transacción incluye una dirección de dispositivo.

Cada transferencia USB consiste en una o más transacciones, y cada una de esas transacciones en turno contiene paquetes de información.

# CAPÍTULO 4

## CAPÍTULO 4.- DISEÑO DEL PRODUCTO

### 4.1.- Diseño Conceptual

El sistema estará compuesto por una tarjeta controladora para poder enviar comandos desde la computadora y a través de esta tarjeta controladora USB poder enviarlos hacia un módulo transmisor por RF utilizando la frecuencia de los 433MHz tendiendo un alcance promedio de 30m permitiéndonos cubrir un área aproximadamente del frente de un hogar para poder accionar las luces, en la (Figura 4.0) se describe el diagrama a bloques del sistema de interconexión de los módulos.

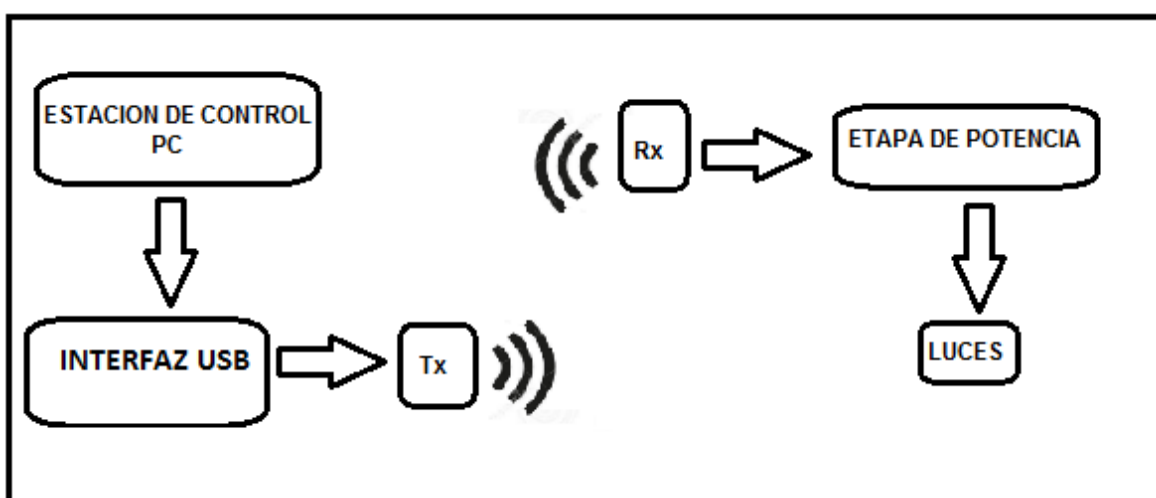
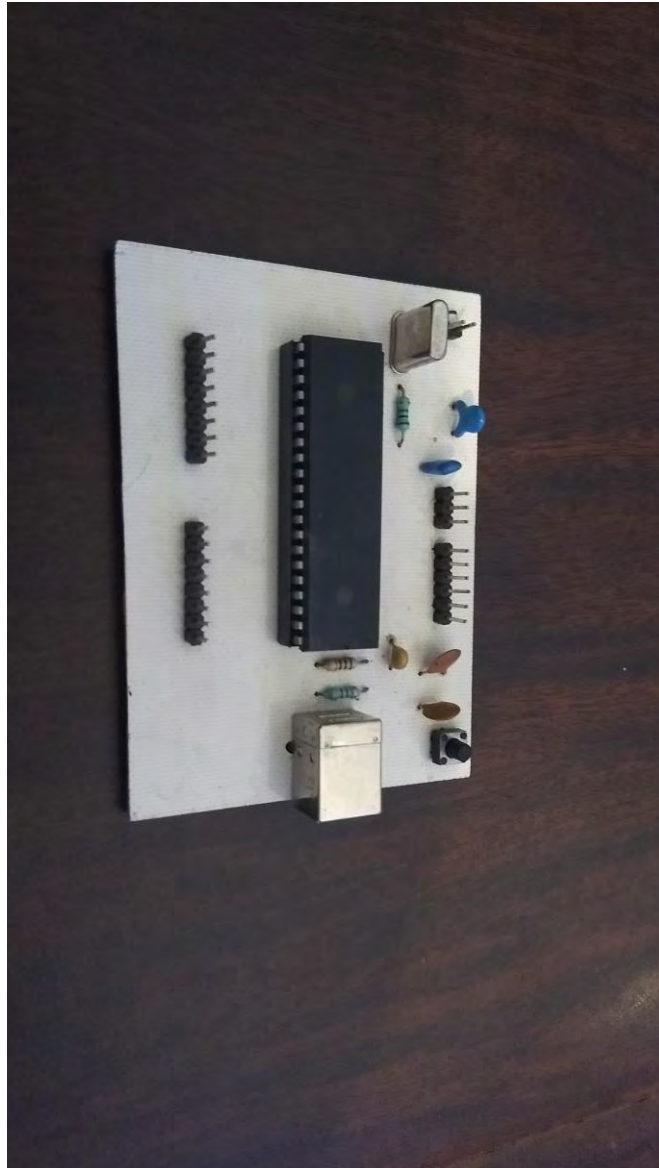


Figura 4.0.- Diagrama a bloques del sistema propuesto.

## 4.2.- Diseño del sistema de domótica

El sistema se compondrá de las siguientes 4 etapas:

- Interfaz de control por USB (Figura 4.1)
- Tarjeta codificadora de RF (Figura 4.2)
- Tarjeta decodificadora de RF (Figura 4.3)
- Interfaz de potencia (Figura 4.4)



**Figura 4.1.-** Interfaz de control por USB.



#### 4.2.1.- Modulo de RF

El sistema está compuesto por dos módulos de RF un Tx (Figura 4.2) y un Rx (Figura 4.3); utilizando la frecuencia de 433MHz para poder establecer una comunicación con un alcance promedio de 30m, para la etapa de codificación y decodificación se utilizaron encoders y decoders MOTOROLA modelo MC145026 y MC145027 respectivamente, utilizando únicamente 4 bits para la transmisión de los datos hacia la etapa de potencia y con esto lograr el encendido y apagado a distancia del sistema de iluminación.



Figura 4.2.- Tarjeta Tx de RF



Figura 4.3.- Tarjeta Rx de RF.

#### 4.2.2.- Etapa de potencia para controlar una carga conectada a AC

Para la etapa de potencia (Figura 4.4) se eligieron opto acopladores MOC3041 y los TRIAC BT138 para lograr la interfaz entre la etapa receptora de RF y el control de potencia y con esto lograr el control del encendido y apagado de las luces u otro elemento que requiera ser conectado al mando a distancia sin sobre pasar los 12 Amperes que soportan los TRIAC.

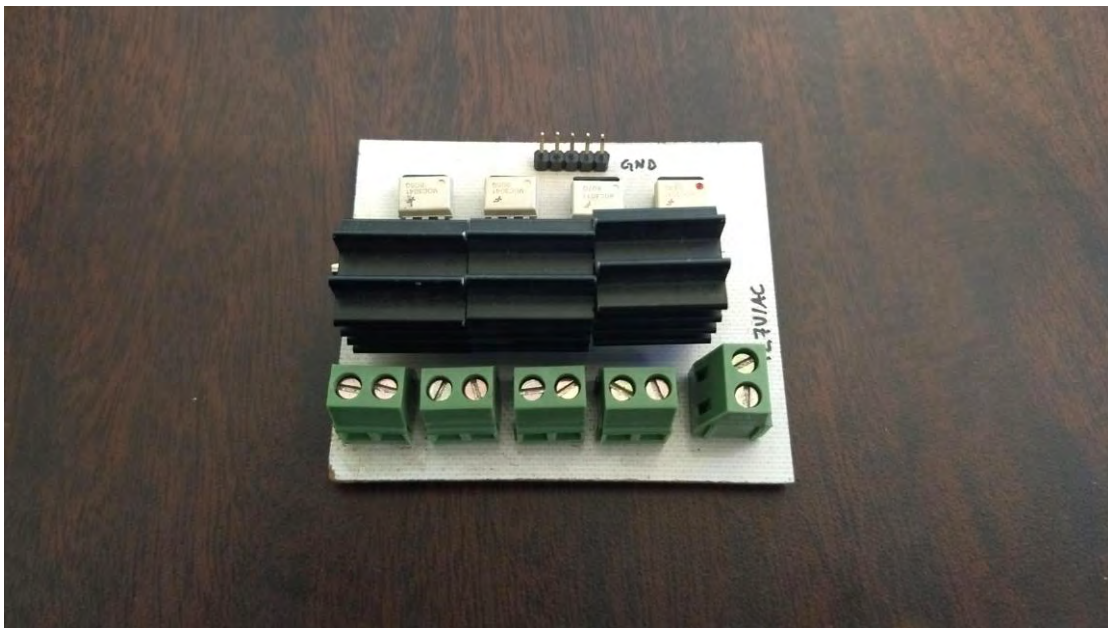


Figura 4.4.- Interfaz de potencia.

### 4.2.3.- Diseño de las placas de circuito impreso

Para el diseño de las placas de circuito impreso se utilizó el software de uso libre Eagle, logrando así desarrollar las mallas de tierra necesarias para las etapas de RF logrando con esto una reducción en las interferencias que afecten al sistema, a continuación, se muestran los esquemáticos de los circuitos implementados para este sistema:

- Esquemático de la interfaz de control por USB (Figura 4.5)
- Esquemático de tarjeta codificadora o Tx de RF (Figura 4.6)
- Esquemático de tarjeta decodificadora o Rx de RF (Figura 4.7)
- Esquemático de interfaz de potencia (Figura 4.8)

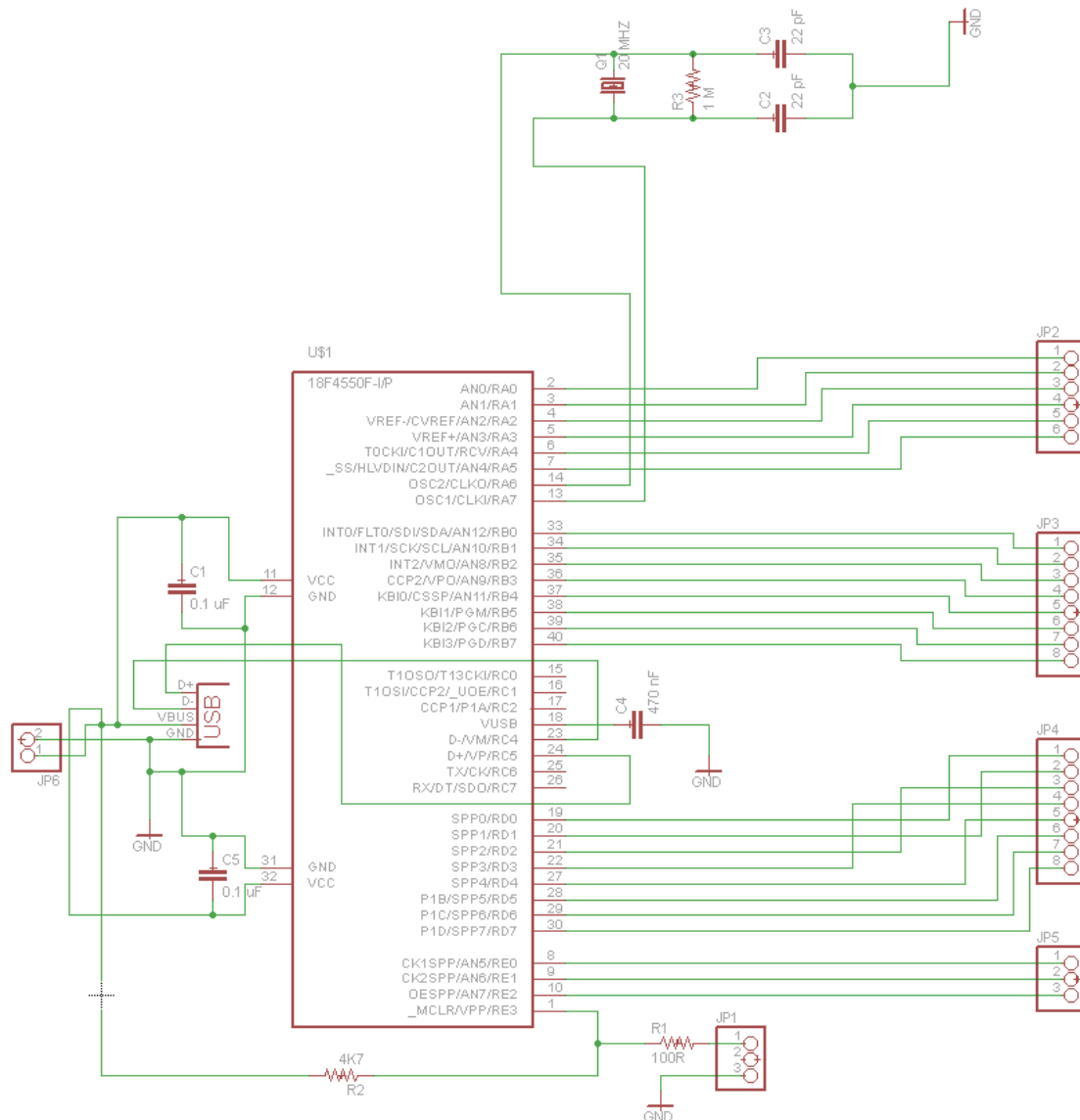


Figura 4.5.- Esquemático de la interfaz de control por USB.

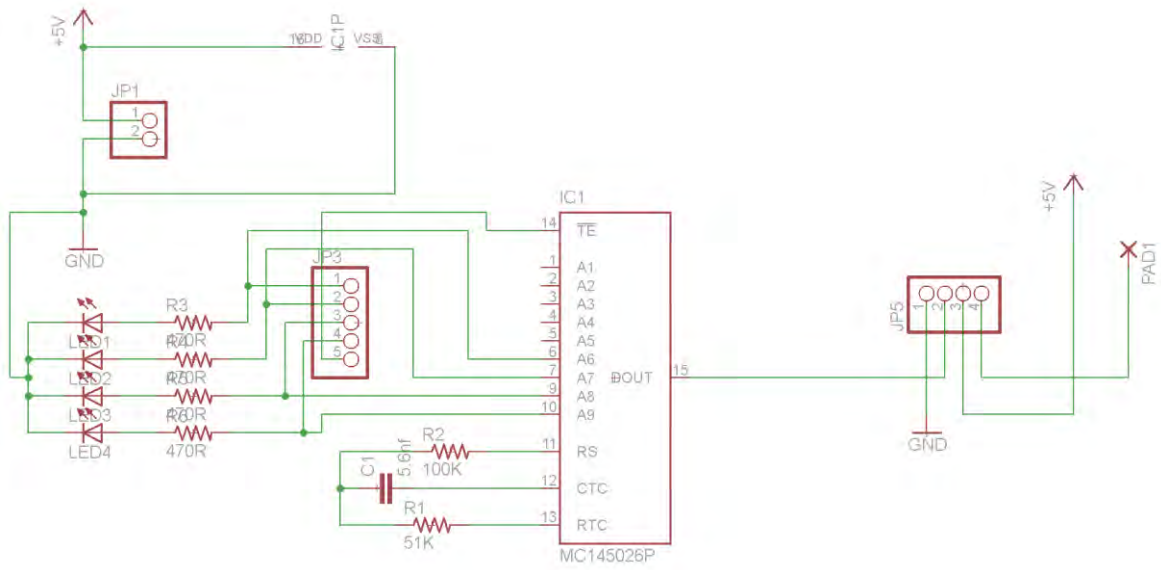


Figura 4.6.- Esquemático de tarjeta codificadora o Tx de RF.

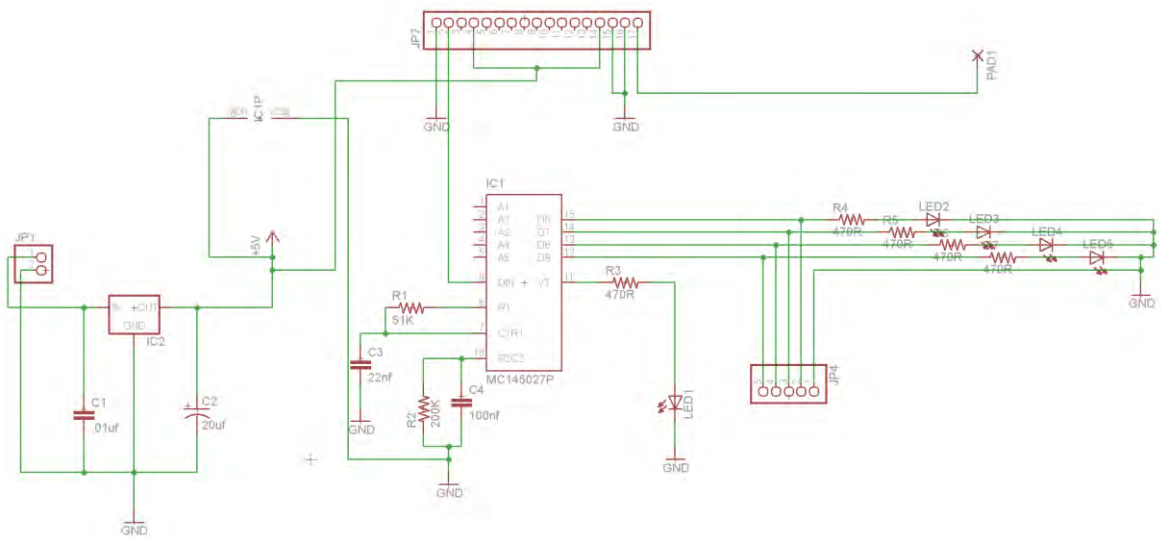
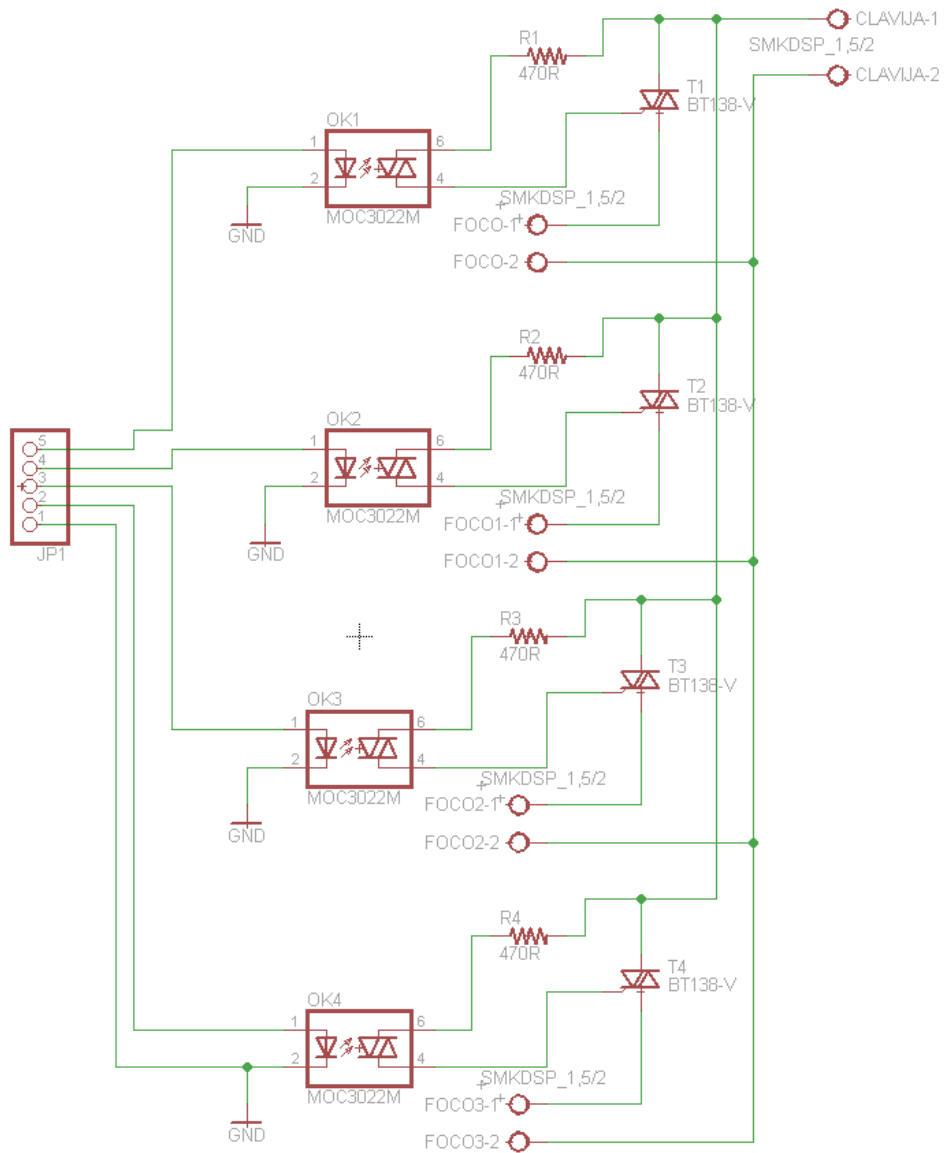


Figura 4.7.- Esquemático de tarjeta decodificadora o Rx de RF.

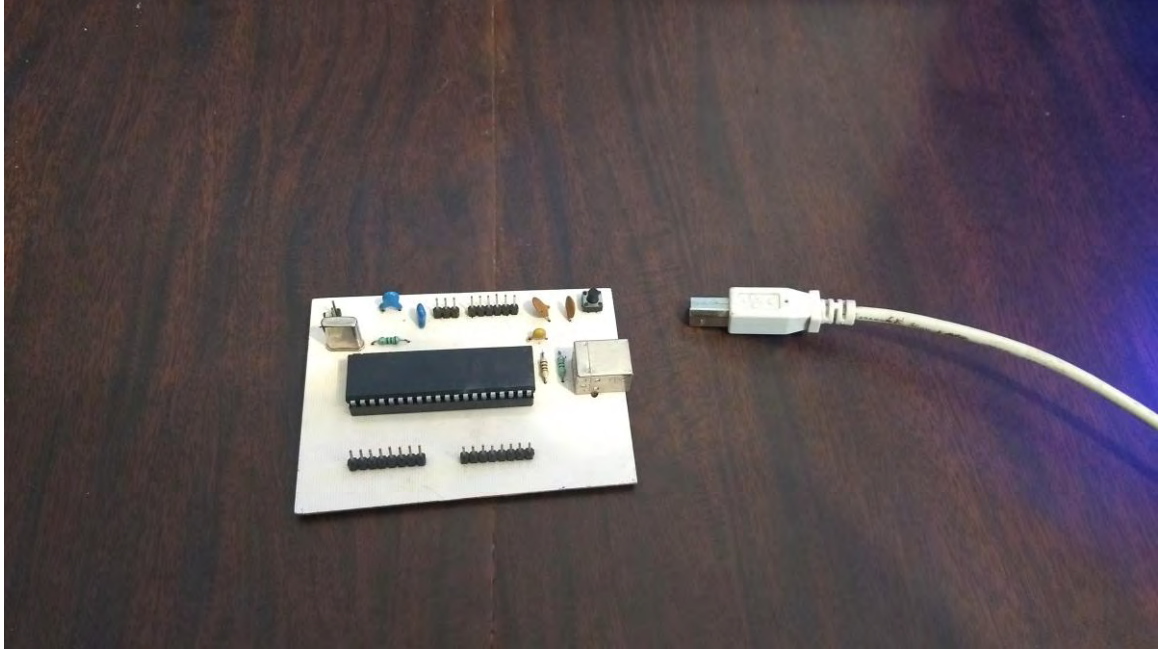


**Figura 4.8.-** Esquemático de interfaz de potencia.

# CAPÍTULO 5

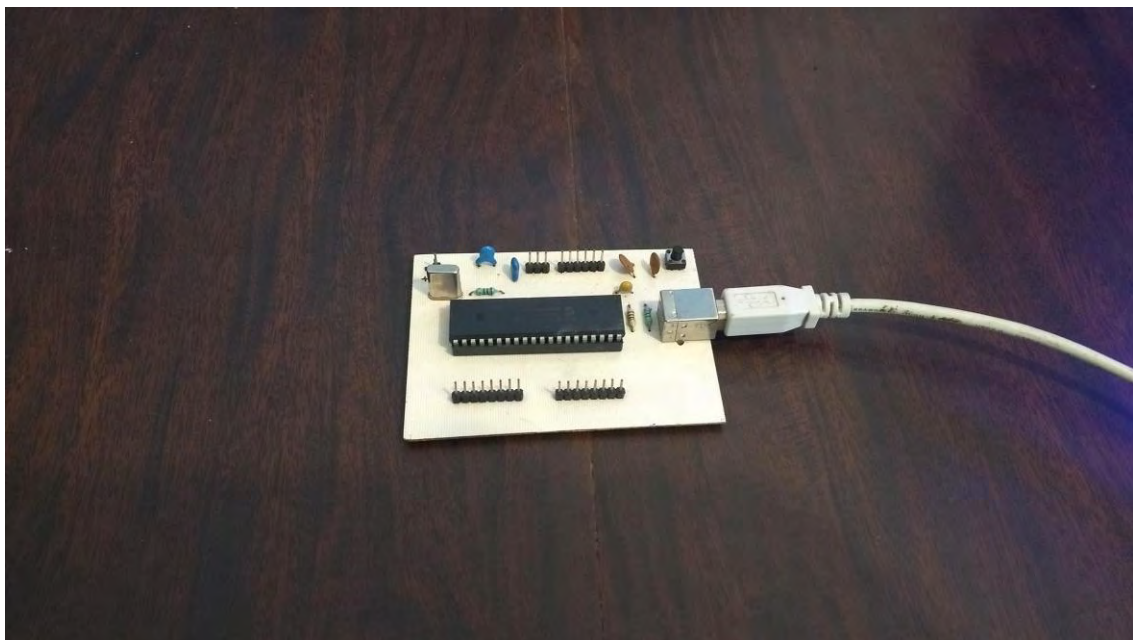
## CAPÍTULO 5.- IMPLEMENTACIÓN

A continuación se describe gráficamente como se llevó a cabo la implementación. La interfaz se conecta al puerto USB mediante un cable tipo B, tipo A (Figura 5.0).



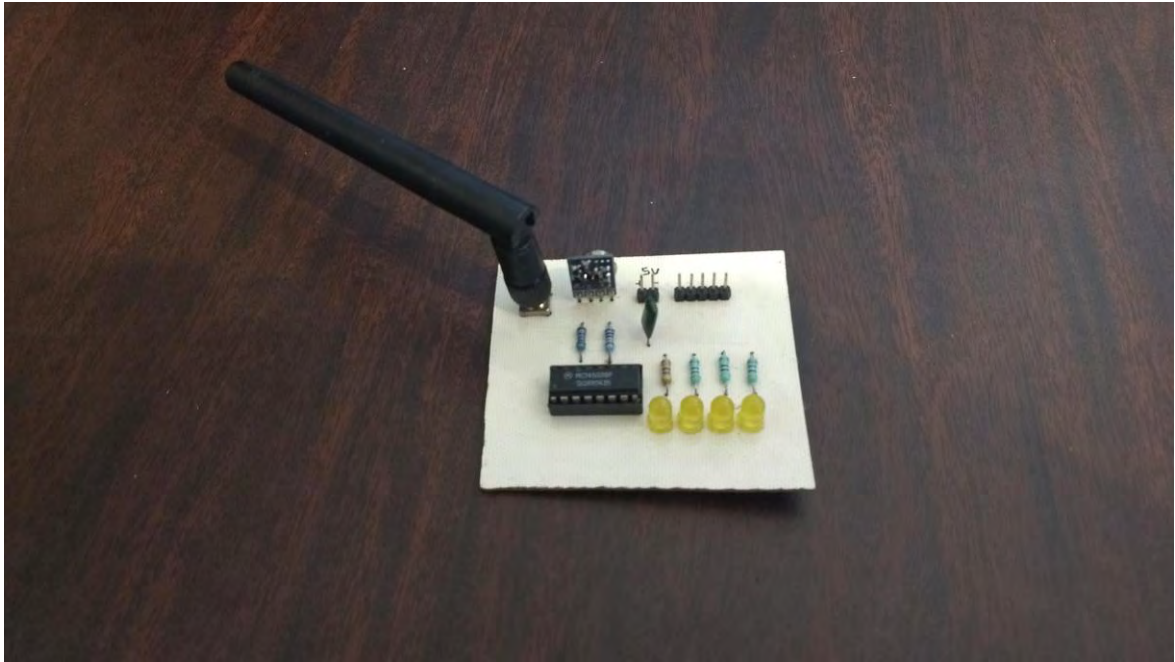
**Figura 5.0.-** Cable USB tipo B para conectar la interfaz a la PC.

Una vez conectada a la PC se reconoce mediante plug and play y está lista para proporcionar voltaje a la etapa de Tx de RF y poder enviar los datos (Figura 5.1).



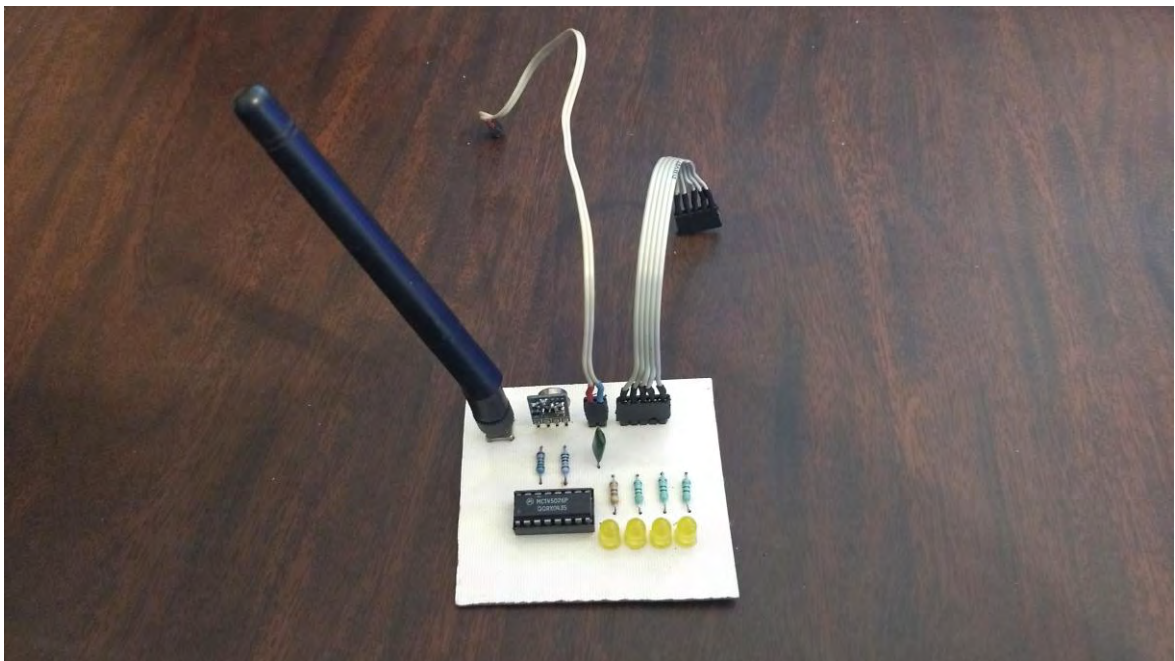
**Figura 5.1.-** Interfaz de control USB conectada mediante el cable tipo B, a tipo A.

La etapa transmisora de datos por RF a 433MHz (Figura 5.2).



**Figura 5.2.-**Tx utilizando el integrado MC145026 habilitado para 4 bits.

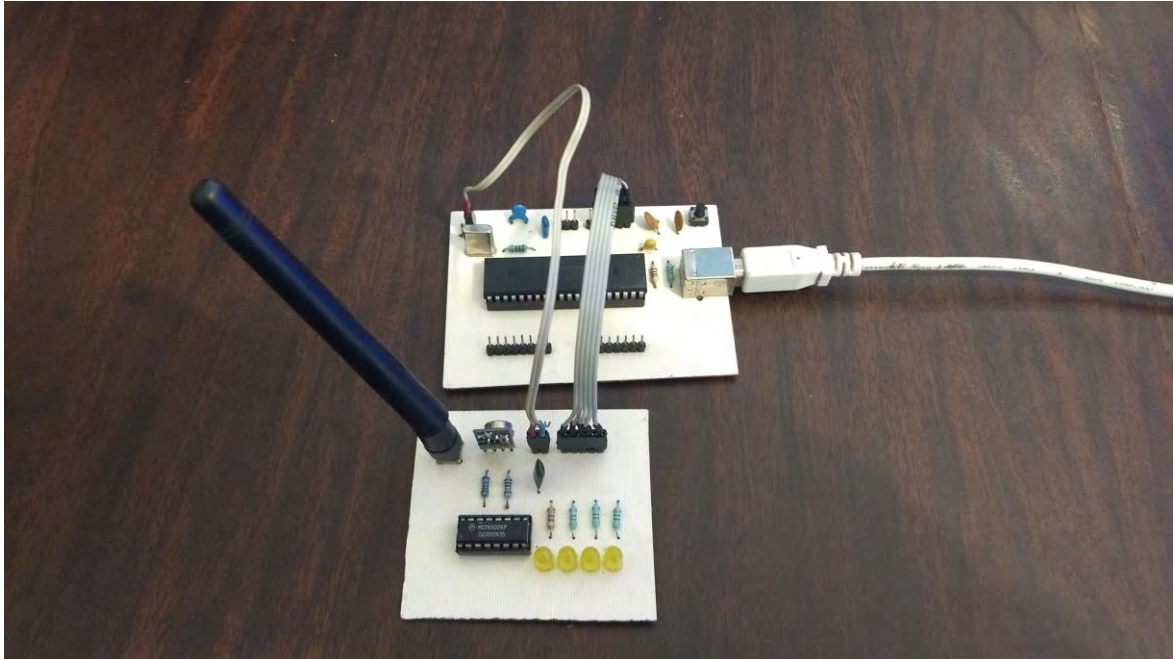
La tarjeta Tx necesita cables para alimentación y los datos que serán enviados al Rx, los cuales obtiene directamente de la interfaz USB (Figura 5.3).



**Figura 5.3.-** Tx con los cables que le proporcionan voltaje y señal a transmitir.

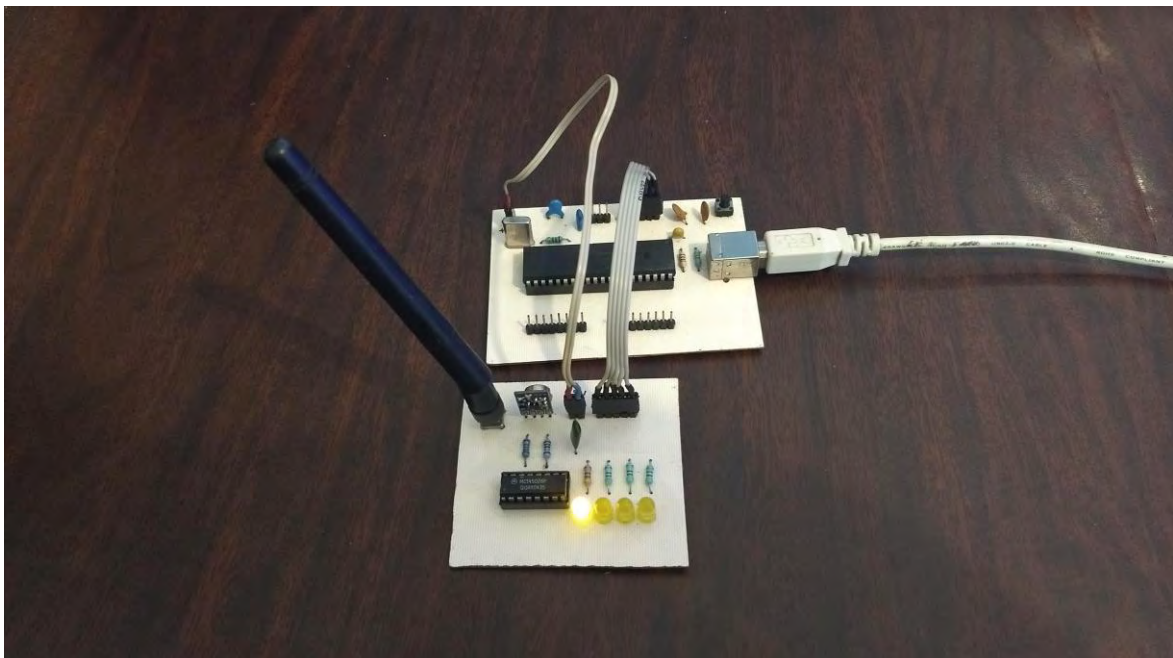


El Tx es conectado a la interfaz USB de donde se alimenta directamente del puerto USB y también recibe los datos a ser enviados (Figura 5.4).



**Figura 5.4.-** La interfaz de control USB es conectada a la etapa de Tx de datos.

Conexión del Tx recibiendo un dato desde la interfaz USB que en ese momento es enviado al Rx (Figura 5.5).



**Figura 5.5.-** Se envía un bit a través del puerto A de la interfaz de control.

Interfaz Gráfica del Usuario (GUI) donde se está enviando un bit a través del puerto A de la interfaz (Figura 5.6).



**Figura 5.6.-** Interfaz gráfica el usuario (GUI) con el envío de un bit en el puerto A.

Una vez enlazado el sistema Tx y Rx se enciende un LED en el Rx solo para verificar que el estado de la transmisión sea válido (Figura 5.7).



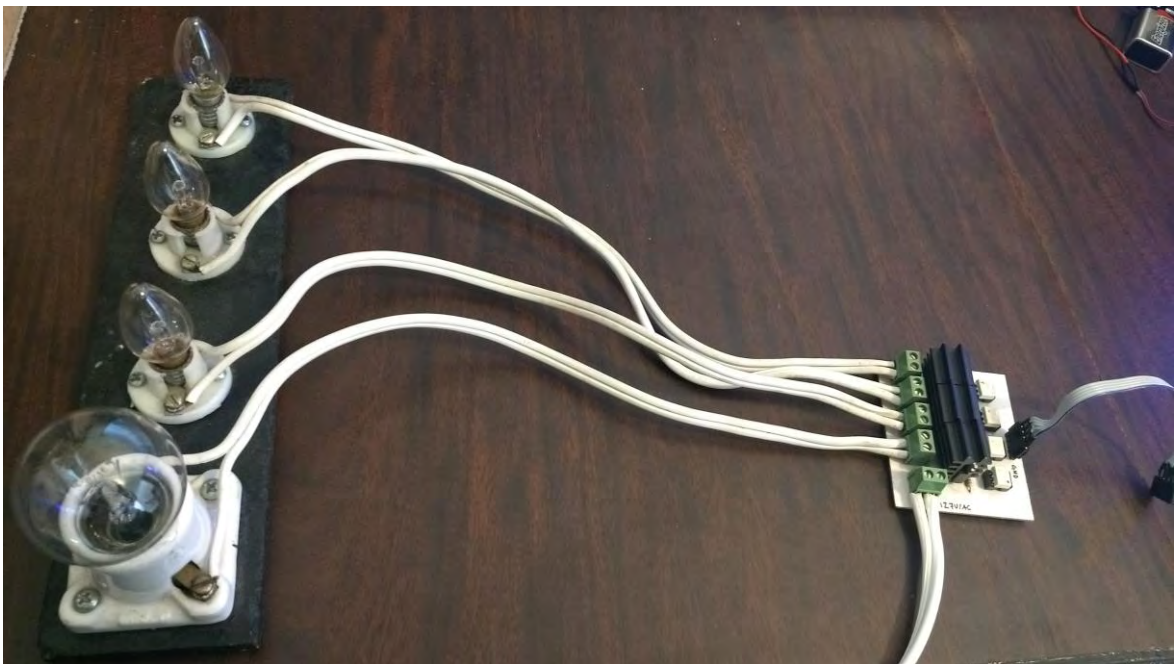
**Figura 5.7.-** Siempre y cuando haya enlace con el Tx se enciende el bit de estado VT.

Muestra del bit enviado por la interfaz USB y el Tx a la derecha, y el bit recibido por el Rx a la izquierda (Figura 5.8).



**Figura 5.8.-** Bit enviado por el Tx a través de la interfaz a la derecha de la foto y recibido por el Rx a la izquierda.

Conexión de la etapa de potencia, se conectó una barra con 4 focos de distinto wataje para simular la carga conectada al interruptor controlado (Figura 5.9).



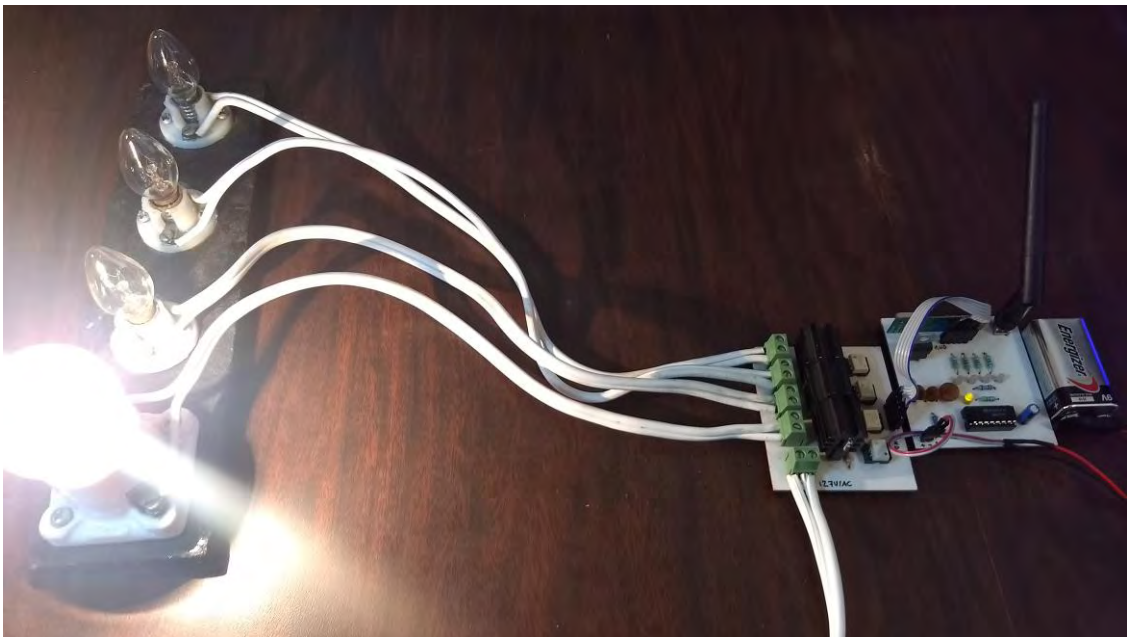
**Figura 5.9.-** Se muestra la conexión de las cargas conectadas al interruptor controlado por la interfaz USB a través de RF.

Se conecta la etapa del receptor que recibe los bits enviados por la interfaz de control de manera inalámbrica (Figura 5.10).



**Figura 5.10.-** Se muestra la conexión del interruptor controlado por los bits recibidos a través de la tarjeta Rx.

En la siguiente foto se muestra el primer bit enviado (Figura 5.11) en este caso representado por el foco de 40W conectado al primer bit del Rx.



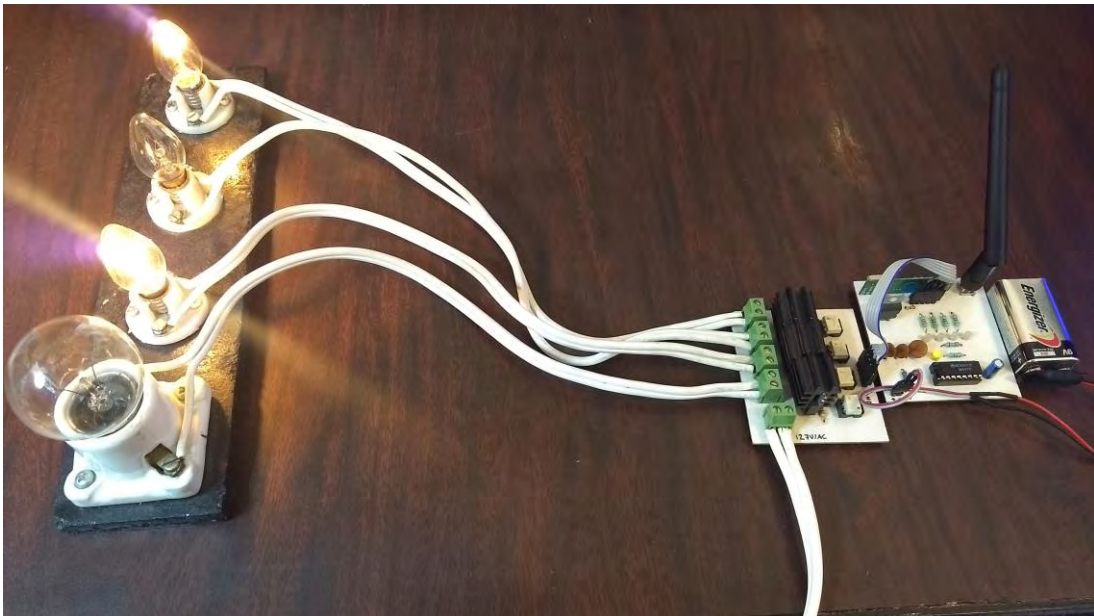
**Figura 5.11.-** Se activa el primer interruptor con el envío del primer bit desde la interfaz USB.

Se prueba una combinación distinta de bits (Figura 5.12) enviados a través de la interfaz USB.



**Figura 5.12.-** Se envía una combinación distinta de señales al interruptor controlado.

Se ven reflejados los bits enviados al recibirlos en la etapa Rx (Figura 5.13) que se encuentra conectada al interruptor, para esta demostración enciende dos focos de 5W.



**Figura 5.13.-** Se reciben 2 bits a través del puerto A de la interfaz de control que activan 2 focos de 5W.

## CONCLUSIONES

Con el desarrollo de este prototipo se comprendió que este tipo de aplicaciones con interfaces son de gran beneficio, integrando este tipo de interfaz a un sistema de domótica se comprendió el alcance y las aplicaciones técnicas de un sistema de domótica, el centralizar el control dentro de una computadora de características comunes sin la necesidad de ser un equipo costoso permite potencializar el uso de estas interfaces de control para un hogar completo al hacer uso de sus 22 bits de salida y tomando en cuenta que por cada puerto USB se conectaría una interfaz, las posibilidades de expansión son mayores de lo esperado. La importancia de haber descrito los elementos que componen la domótica nos permitió visualizar las posibles aplicaciones para el uso de esta interfaz, la cual quedo orientada a un interruptor.

Al describir los elementos que componen al puerto USB y conocer las velocidades y distintos tipos de conexión del puerto USB permite saber la posible compatibilidad con microcontroladores y así elegir el más adecuado para este tipo de interfaces, en este caso por sus características se realizó el diseño de la interfaz de control digital con un PIC 18F4550.

Los resultados obtenidos fueron los esperados logrando controlar hasta 4 dispositivos de manera inalámbrica mediante la interfaz de control digital USB aplicándola a un sistema de domótica, se logró superar la expectativa de las 4 luces desarrollando al máximo la interfaz pudiendo obtener así hasta 22 bits que permiten enviar señales para ser transmitidas mediante la interfaz de control por RF maximizando las capacidades de control y periféricos que se pueden conectar a futuro en la interfaz. Con esto se cumplió con el objetivo general planteado en esta tesis.

## TRABAJO A FUTURO

Se tiene la propuesta de reducir el tamaño de la interfaz y del interruptor controlado inalámbricamente con la etapa de potencia integrando estas dos etapas en una sola tarjeta, al igual que se tiene contemplado el rediseño de la interfaz en una tarjeta más pequeña miniaturizándola con componentes SMD para reducir el consumo de energía, tamaño y para facilitar el manejo, para futuras versiones de esta interfaz se tiene previsto el manejo de más bits de control que pueden ser multiplexados elevando así la cantidad de dispositivos que se pueden conectar del lado del receptor, por otra parte también se pretende diseñar e implementar un sistema de adquisición de datos para poder lograr un sistema de telemetría y que nos permita tener comunicación bi-direccional con el entorno donde esté instalado.

## BIBLIOGRAFÍA

- artechulate.info*. (11 de 06 de 2018). Obtenido de <http://www.artechulate.info/usb-power-wiring-diagram/>
- Axelson, J. (2009). *USB COMPLETE FOURTH EDITION* (Cuarta Edición ed.). Madison, Wisconsin: Lakeview Research LLC.
- Axelson, J. (2011). *USB EMBEDDED HOSTS*. Lakeview Research LLC .
- Humphris, C. W. (2014). *PIC MICRO PRINCIPLES ON YOUR MOBILE*. eptsoft.
- Humphris, C. W. (2014). *PIC MICRO PRINCIPLES V10*. eptsoft.
- Jan, A. (2006). *USB MASS STORAGE*. Lakeview Research LLC.
- Lunt, B. D. (2013). *USB: THE UNIVERSAL SERIAL BUS*. Forever Young Software.
- McGraw Hill. (1994). Metodología de Programación, Programación en C, Aplicaciones electrónicas. *PC World*, 1-16.
- Minns, P. D. (2013). *C PROGRAMMING*. AuthorHouse.
- Mueller, S. (2013). *UPGRADING AND REPAIRING PCs, 21st EDITION*. QUE.
- Riley, M. (2012). *PROGRAMMING YOUR HOME*. The Pragmatic Programmers, LLC.
- Scanail, M. J. (2014). *SENSOR TECHNOLOGIES*. Apress Media, LLC.
- Steven McDowell, M. D. (1998). *USB explained*. Prentice Hall.
- Technology, M. (2004). *PIC 18F2455/2550/4455/4550 Data Sheet*. Retrieved April 19, 2018, from [http://www.datasheetcatalog.com/datasheets\\_pdf/P/I/C/1/PIC18F4550.shtml](http://www.datasheetcatalog.com/datasheets_pdf/P/I/C/1/PIC18F4550.shtml)
- usb.org. (s.f.). *usb.org*. Recuperado el 20 de Julio de 2016, de [http://www.usb.org/developers/docs/usb20\\_docs/#usb20spec](http://www.usb.org/developers/docs/usb20_docs/#usb20spec)
- Wilcox, J. (2005-2014). *PIC MICROCONTROLLER PROGRAMMING COURSE WITH C & ASSEMBLY*. Yankee Bush Software LLC.
- Wordpress.com*. (2013). Obtenido de <https://jesusmelendresg.wordpress.com/2013/04/18/puerto-serial-com/>

## ANEXOS

### ALGUNOS TIPOS DE PUERTOS

#### Puerto paralelo

La comunicación en paralelo se realiza mediante la transferencia simultánea de tramas de bits que constituyen el dato (byte o palabra) y presenta la ventaja de que la transmisión puede ser más rápida. Sin embargo, las comunicaciones en paralelo no pueden ser implementadas para grandes distancias debido a que no es viable la conexión física de todas las líneas necesarias.

Las comunicaciones en paralelo propiamente dichas no han sido normalizadas, lo que sí se reconoce es la norma Centronics, para la conexión de la PC a la impresora, mediante el envío simultáneo de 8 bits de datos (un byte), además de un conjunto de líneas de protocolo (hand shake o intercambio).

Los antiguos circuitos integrados que se incluían en las tarjetas de interfaz de puerto paralelo no permitían la recepción de datos, sólo estaban diseñados para el envío de información al exterior.

La norma Centronics hace referencia a las características de la conexión entre una interface de puerto paralelo y una impresora. Las líneas son latcheadas, esto es, mantienen siempre el último valor establecido en ellas mientras no se cambien expresamente y los niveles de voltaje y de corriente coinciden con los niveles de la lógica TTL.

La norma Centronics establece el nombre y las características de 36 líneas para la conexión entre la computadora y la impresora, en realidad, para la transferencia de las señales de datos y de control a través de la tarjeta de interfaz paralelo sólo se requieren 18 líneas, las restantes son líneas de tierra (GND) que se enrollan alrededor de los cables de señal para proporcionarles apantallamiento y protección contra interferencias.

Por esto, dichas tarjetas suelen incorporar un conector hembra DB-25, mientras que prácticamente todas las impresoras incorporan un conector hembra tipo Centronics macho de 36 pines, los cables comerciales para la conexión paralelo entre la PC y la impresora tienen una longitud de 2 metros, aunque no es recomendable que tengan una longitud superior a 5 metros si se desea una conexión fiable y sin interferencias.

(McGraw Hill, 1994)



Tanto en el conector tipo Centronics de 36 pines como en el conector DB-25 se tienen 8 líneas correspondientes a los bits de datos (D0 a D7) que son líneas de salida, así lo establece el estándar Centronics, sin embargo y sobre todo en las implementaciones más recientes, la electrónica asociada al interfaz del puerto paralelo puede ser tal que las líneas de datos pueden ser leídas desde la computadora y, por lo tanto, ser consideradas como líneas bidireccionales, aunque sea en determinadas condiciones y con el software adecuado.

En MS-DOS se pueden gestionar hasta cuatro interfaces de puertos paralelo, LPT1, LPT2, LPT3 y LPT4, además, reserva las siglas PRN como sinónimo del LPT1, de modo que puede ser tratado como un archivo genérico.

### **Entradas y salidas por el puerto paralelo**

El puerto paralelo se desarrolló de acuerdo con el estándar Centronics con el fin, casi exclusivo, de que la computadora pudiese enviar datos en paralelo a la impresora conectada, no se pensó en la posibilidad inversa: que la computadora pudiese recibir datos a través de ese puerto.

Las operaciones de entrada y salida de información a través del puerto paralelo en una computadora las realizaremos gestionando el puerto paralelo en el nivel de registros, es decir, programando directamente los circuitos integrados que constituyen la tarjeta de interfaz, lo cual permitirá aprovechar al máximo todas las posibilidades que ofrezca realmente el hardware de la tarjeta de interface.

### **Características E/S**

Cuando usamos el puerto paralelo para otro cometido distinto al original, solo podemos hablar de 12 líneas de salida de información desde el ordenador:

- pines del 2 al 9 → registro de datos
- pines 1,14, 16 y 17 → registro de control 15 líneas de entrada al mismo:
- pines 2-9→ registro de datos

Esto hace del puerto paralelo un interfaz de comunicación con el exterior bastante flexible. El registro de estado es de sólo lectura. Cuando se lee este registro, lo que se recibe es el estado lógico de los pines 10, 11, 12, 13 y 15 del conector DB-25.

En este caso, el hecho de que las operaciones de entrada y salida se hagan por las mismas líneas, condiciona notablemente el proceso de lectura, ya que con esa configuración electrónica de las líneas de datos (D0 - D7), los valores lógicos leídos dependerán del nivel lógico presente en el registro y del valor de tensión en la línea.

(McGraw Hill, 1994)

## **Puerto Serie RS-232**

Puerto serial, puerto COM, puerto de comunicaciones y puerto RS-232 (Recommended Standard-232), hacen referencia al mismo puerto. Se le llama serial, porque permite el envío de datos, uno detrás de otro, mientras que un paralelo se dedica a enviar los datos de manera simultánea.

El acrónimo COM es debido al término ("*COMmunications*"), que traducido significa comunicaciones. Es un conector semi trapezoidal de 9 terminales, que permite la transmisión de datos desde un dispositivo externo (también llamado periférico), hacia la computadora; por ello es denominado puerto, compitió directamente en el mercado contra el puerto LPT.

Este puerto está siendo reemplazado actualmente casi en su totalidad por el puerto USB para el uso en tablets, dispositivos móviles, joysticks, etc. pero aún viene integrado en la denominada placa base (mother board) en algunos equipos.

### **Características:**

1. Normalmente estos suelen ser 2 en una placa base y son denominados COM 1 y COM
2. Estos puertos funcionan con un chip llamado UART, que es un controlador serie.
3. El puerto serie utiliza direcciones y una línea de señales, un IRQ para llamar la atención del procesador. Además, el Software de control debe conocer la dirección.
4. La mayoría de los puertos serie utilizan direcciones Standard predefinidas. Éstas están descritas normalmente en base hexadecimal.
5. Para el protocolo de transmisión de datos, sólo se tiene en cuenta dos estados de la línea, 0 y 1, también llamados Low y High.
6. El conector tiene sus extremos en ángulo de manera que el enchufe podrá introducirse de una manera solamente.

(Wordpress.com, 2013)

Se utilizaba principalmente para la conexión del mouse, algunos tipos antiguos de escáneres y actualmente para dispositivos de consolas de control y medición, cada puerto, permite conectar solamente 1 dispositivo. Para conectar y desconectar los dispositivos, así como para que la computadora los reconozca de manera correcta, es necesario apagar y reiniciar la computadora.

### **Velocidad de la transmisión del puerto serial**

La forma de medir la velocidad de transmisión del puerto serial es en Kilo Bytes/segundo (KB/s):

SERIAL COM = 112KB/S

### **USOS ESPECÍFICOS DEL PUERTO SERIAL**

El uso principal que se le asignaba era para conectar el mouse, e incluso escáneres, pero con la salida al mercado del puerto USB se dejó de utilizar con este fin. Un uso actual es para conectar algunos tipos de agendas electrónicas, equipos de recolección de datos ya sea en ventas o equipo de laboratorio como equipos de instrumentación y conexiones directas entre computadoras.

(Wordpress.com, 2013)

## USB

El Universal Serial Bus o (bus serie universal) USB es una arquitectura de bus desarrollado por un grupo de siete empresas (Compaq, Digital Equipment Corp., IBM PC Co., Intel, Microsoft, NEC y Northern Telecom) que forma parte de los avances plug-and-play y permite instalar periféricos sin tener que abrir la computadora para instalarle hardware, es decir, basta con que se conecte dicho periférico en el puerto USB y en algunos caso instalar el driver para poder utilizar el dispositivo correctamente.

Así nació el USB (Universal Serial Bus) con una velocidad de 12Mb/seg y su evolución el USB 2.0, apodado USB de alta velocidad, con velocidades en este momento de hasta 480 Mb/seg, es decir, 40 veces más rápido que las conexiones mediante cables USB 1.1.

El USB es la solución preferida si se quiere comunicar una computadora con un dispositivo externo como discos duros, impresoras, así como también lectores de huellas digitales pueden hacer uso de esta tecnología por igual. Esta interfaz de comunicación entre la computadora y dispositivos externos es la preferida por la gran mayoría de fabricantes para desarrollar productos de forma masiva de casi cualquier tipo, como también se han desarrollado productos de bajo volumen de mercado para algunos usos específicos.

Según estadísticas cerca de un millón de computadoras vendidas anualmente consideran el puerto USB como la interfaz con más éxito jamás diseñada. Cada computadora diseñada recientemente tiene puertos USB para conectar teclados, mouse, controles para juegos, scanners, cámaras, impresoras, discos duros y más. El USB es confiable, rápido, versátil, estable en sus niveles de consumo, económico, y compatible por la gran mayoría de sistemas operativos.

(Axelson, USB COMPLETE FOURTH EDITION, 2009)

## **USB 1.0**

El Universal Serial Bus revisión 1.0 fue desarrollado en enero de 1996. La capacidad de USB estuvo disponible primero en las PC con el desarrollo de Windows 95 OEM Service Release 2, disponible únicamente para fabricantes de equipo instalando Windows 95 en equipos que ellos vendían. En estas versiones el soporte y uso del USB no era tan bueno y era limitado, y por lo tanto no había muchos periféricos USB disponibles en el mercado, por lo tanto el uso de USB en esta época era limitado.

Esta situación mejoro cuando se lanzó al mercado el Windows 98 en Junio de 1998, para este año, había disponibles mucho más vendedores y desarrolladores de periféricos USB, y el USB comenzaba a tomar fuerza como una interface popular entre los usuarios. Windows 98 Second Edition (SE) corrigió muchos de los fallos y soporte de los controladores USB. La edición original de Windows 98 fue llamada Windows 98 Gold Edition para distinguirlo del Windows 98 SE.

## **USB 1.1**

EL Universal Serial Bus revisión 1.1 fue dado a conocer en Septiembre de 1998 y agrego un nuevo tipo de transferencia (interrupciones de salida).

## **USB 2.0**

Mientras el USB fue ganando popularidad, y las PC comenzaban a ser más poderosas, se comenzó a demandar un mayor crecimiento de la velocidad para el bus.

Investigaciones mostraron que una velocidad 40x más rápida que la velocidad alta podía mantener retro-compatibilidad con interfaces de baja y alta velocidad. En Abril del 2000 se pudo ver el lanzamiento de la revisión 2.0 del USB, que agrego una velocidad alta de 480Mbps. La velocidad alta hizo que el USB más atractivo para periféricos como las impresoras, discos duros y video cámaras. Por su parte Windows agrego soporte para USB 2.0 en Windows XP SP2, fue entonces que el USB 2.0 reemplazo al USB 1.1.

Excepto por los HUBS, un dispositivo USB 2.0 puede soportar bajas velocidades, o altas velocidades de transferencia. La habilidad de comunicarse en cualquier velocidad incremento la complejidad de los HUBS pero conservo el ancho de banda en el bus y elimino el uso de diferentes HUBS para diferentes velocidades.

El USB 2.0 es retro compatible con el USB 1.x, en otras palabras, los dispositivos USB 2.0 pueden usar los mismos conectores y cables que los dispositivos 1.x, y un dispositivo USB 2.0 funciona cuando es conectado a una PC que soporta tanto USB 1.x como 2.0, excepto por unos cuantos dispositivos que funcionan únicamente con un bus de alta velocidad requiriendo así el soporte obligatorio de USB 2.0.

(Axelson, USB COMPLETE FOURTH EDITION, 2009)

Cuando los dispositivos USB 2.0 comenzaron a estar disponibles para los usuarios, hubo algo de confusión sobre si todos los dispositivos USB 2.0 soportarían alta velocidad. Para reducir la confusión, la USB-IF desarrollo recomendaciones acerca del nombre y empaque para enfatizar la velocidad y compatibilidad más que la versión de USB. Un producto que soporta alta velocidad es etiquetado como “High-Speed USB”, y un mensaje en el empaque nuevamente muestra que es completamente compatible con el USB en su versión original y compatible con el USB 2.0.

Para productos que soportaban baja velocidad únicamente, se recomendaron mensajes en el empaque que dijeran: (Compatible con USB 2.0 y funciona con USB y sistemas USB Hi-Speed, así como periféricos y cables). La recomendación advertía referencias sobre baja o alta velocidad en el empaque que era presentado al consumidor final.

Para usar altas velocidades, un dispositivo debe conectarse a una computadora con puertos USB 2.0 o HUBS USB 3.0 entre el dispositivo receptor. Los hosts y HUBS USB 2.0 y USB 3.0 pueden también comunicarse con dispositivos USB 1.x.

### **USB 3.0**

La revisión 1.0 del Universal Serial Bus 3.0 fue lanzada en Noviembre del 2008, con el primer dispositivo-controlador que había sido esperado desde ya hacía un año antes. Windows estaría interesado en soportar USB 3.0 un poco después del lanzamiento del Windows 7, el sucesor de Windows Vista.

El USB 3.0 redefinió una nueva arquitectura de doble-bus con dos buses físicamente operando en paralelo. El USB 3.0 proveyó un par de cables para el tráfico de USB 2.0 y cables adicionales para el soporte de la nueva Súper Velocidad (Super Speed) del bus de hasta 5Gbps. La Súper Velocidad ofrece un incremento de más de 10x sobre la velocidad del USB 2.0. Como un extra, la Súper Velocidad tiene un par de cables para cada dirección de transferencia de datos en ambas direcciones al mismo tiempo.

El USB 3.0 también incrementa el número de corriente que el bus puede soportar para algunos dispositivos y define protocolos para un ahorro de energía más eficaz y transferencias más eficientes.

(Axelson, USB COMPLETE FOURTH EDITION, 2009)

## **USB OTG (On-The-Go)**

El USB se vuelve la interfaz elegida por todos los tipos de periféricos, los desarrolladores comienzan a preguntarse sobre el acceso de periféricos USB a otros dispositivos USB. Por ejemplo, un usuario quiere agregar una impresora a una cámara o un teclado a un PDA o Tablet, el complemento USB OTG (On-The-Go o en el camino) del puerto USB 2.0 define los límites y capacidades de los dispositivos de funcionar como un host que los dispositivos pueden implementar para habilitar comunicaciones con periféricos USB.

## **Wireless USB (USB inalámbrico)**

Los desarrolladores que quieren diseñar dispositivos con interfaces inalámbricas tienen muchas opciones. El Grupo Promotor del USB Inalámbrico en su especificación Wireless Universal Serial Bus Specification define los certificados para dispositivos USB inalámbricos (WUSB) una interfaz de comunicación de hasta 480 Mbps. Los semiconductores Cypress Wireless USB permiten implementar dispositivos inalámbricos que funcionan como dispositivos USB de baja velocidad. Otra opción es el uso de un adaptador que convierte las señales entre el dispositivo USB y una interfaz inalámbrica como los módulos Zigbee, Bluetooth o Wi-Fi.

(usb.org, s.f.)