

## Desarrollo de aplicaciones con Microsoft office para control de prototipo de circuito integrado a través de puerto USB

ABRIL-GARCÍA, José Humberto\*†, MEZA-IBARRA, Iván Dostoyewski, ELIZARRARÁS-UIROZ, José de Jesús y GARCÍA-JUÁREZ, Alejandro

*Universidad de Sonora. Blvd. Luis Encinas y Rosales S/N, Col. Centro, C.P. 8300, Hermosillo, Sonora, México.*

Recibido Enero 28, 2014; Aceptado Mayo 29, 2014

### Resumen

Actualmente se está reduciendo la brecha de comunicación entre hardware y software, específicamente con circuitos integrados basados en puerto USB o metodologías que ofrezcan conocer su comportamiento interno con fines didácticos, asimismo no se encuentran librerías para integrar aplicaciones de Microsoft Office para enviar o recibir datos a través del puerto USB.

El objetivo de este trabajo es desarrollar diferentes aplicaciones para el control de un prototipo de un circuito integrado utilizando Programable Integrated Circuit (PIC) que pueda ser controlado mediante una interfaz USB, desde una computadora, que apoye y permita el uso didáctico, así como el desarrollo de macros en Microsoft Office, que se integren a proyectos de automatización, control, monitoreo, domótica, entre otros.

**Prototipo, PIC, USB, DLL, VB.Net, Macros.**

### Abstract

At present the communication breach is diminishing between hardware and software, especially with integrated circuits based on port USB or methodologies that offer to know its internal behavior with didactic ends, also bookstores are not to integrate applications of Microsoft Office to send or to receive information across the port USB.

The target of this one work is to develop different applications for the control of a prototype of an integrated circuit using Programmable Integrated Circuit (PIC) that interface USB could be controlled by means of one, from a computer, which supports and allows the didactic use, as well as the macros development in Microsoft Office, which integrate to projects of automation, control, monitoring, domótica, between others.

**Prototype, PIC, USB, DLL, VB.Net, Macros.**

**Citación:** ABRIL-GARCÍA, José Humberto, MEZA-IBARRA, Iván Dostoyewski, ELIZARRARÁS-UIROZ, José de Jesús y GARCÍA-JUÁREZ, Alejandro. Desarrollo de aplicaciones con Microsoft office para control de prototipo de circuito integrado a través de puerto USB. Revista de Sistemas Computacionales y TIC's. 2015, 1-1:41-51

\* Correspondencia al Autor (Correo Electrónico: jhabril@gmail.com)

† Investigador contribuyendo como primer autor.

## Introducción

Esta propuesta presenta una metodología con para desarrollar un aplicaciones para control de un circuito integrado, en donde se aprenda a implementar una Interfaz Gráfica de Usuario en un lenguaje de programación VB.Net, C++ . Net, así como la Librería de Enlace Dinámico mediante la cual se logrará integrar, hardware y software. Este documento consta de tres apartados, Definición del problema, Metodología de desarrollo y Resultados, que ayudará a un formador a comprender cómo aplicar teorías, modelos y metodologías que involucren el desarrollo de Circuitos electrónicos que pueden implementarse en un Circuito Integrado Programable, así como el desarrollo de software, en el que se podrá hacer énfasis en una o varias áreas temáticas.

El objetivo del siguiente trabajo es desarrollar diferentes aplicaciones para el control de un prototipo de un circuito integrado utilizando Programmable Integrated Circuit (PIC) que pueda ser controlado mediante una interfaz USB, desde una computadora, que apoye y permita el uso didáctico, así como el desarrollo de macros en Microsoft Office, que se integren a proyectos de automatización, control, monitoreo, domótica, entre otros.

## Revisión de literatura

En la actualidad, no es fácil implementar actividades prácticas que permitan desarrollar el aprendizaje en cuanto a la interacción de dispositivos periféricos a través del puerto Universal Serial Bus (USB) y la computadora, mediante el uso de un software, con fines didácticos y de manera económica. Para desarrollar una actividad práctica con fines didácticos y que los estudiantes puedan culminar con éxito, es importante determinar los componentes y herramientas que estén a su alcance. Se requiere que el alumno desarrolle conocimientos en cuanto a la implementación de Circuitos Integrados Programables.

Por ello se dio a la tarea de investigar microcontroladores o microprocesadores en el mercado. Se hallaron varios tipos y marcas, entre los que destacan Arduino (2014) y sus múltiples clones Ktuluino como se puede observar en (Jeff, 2013), \$9 Arduino (Timmis, 2013), Moteino (Lab, 2013), Prototino (SpikenzieLabs, 2011), Paperduino (Picuino, 2012), Boarduino (Adafruit, 2013), Rasberry Pi (2014), UDOO (2014) y microcontrolador PIC (2014), las iniciales se refieren a Peripheral Interface Controller, por sus siglas en Inglés, se determinó utilizar este último, por ser una opción económica. Éste ofrece las bases para conocer las funciones y capacidades de un Circuito Integrado Programable e inclusive representa una opción como solución real, hecho que no se pondrá en discusión en este artículo. Antes de enfocarse a la adquisición e implementación de un Circuito electrónico, es altamente recomendable utilizar un software en el que se pueda simular y hacer variaciones al diseño, así como calcular o determinar diferentes parámetros. Para el desarrollo de la Interfaz Gráfica de Usuario (GUI, por sus siglas en Inglés), se tiene que diseñar una aplicación que represente las entradas y salidas del circuito diseñado, hay diferentes entornos de desarrollo de software, una clasificación que debemos considerar es el tipo de licenciamiento ya sea Opensource o de Costo, Microsoft Visual Studio Community 2013, representa una opción libre de costo para aprender o perfeccionar el desarrollo de software, así como para comprender mejor algún modelo de desarrollo de software.

En investigación anterior, los investigadores (Meza Ibarra, I. D.; Abril García, J. H.; y Elizarrarás Quiroz, J. J. , 2014) presentaron resultados sobre .Net. Ahora se ha trabajado en el sistema operativo a utilizar como propuesta inicial debe ser una versión de Microsoft Windows.

## Diseño e implementación de prototipo de circuito integrado

Un circuito integrado, es un chip que está compuesto principalmente por cristal semiconductor de silicón, que a su vez contiene elementos eléctricos como condensadores, diodos, resistencias y transistores conectados entre sí, un circuito integrado, cuenta con extremidades metálicas, por las cuales entra o sale señales que representan información. Un Programmable Integrated Circuit (PIC), es un dispositivo que combina memoria, unidad central de procesamiento y periféricos de entrada y salida para realizar una serie de tareas, Predko (2001). La interfaz Universal Serial Bus (USB) es un medio de comunicación entre una computadora y diversos dispositivos periféricos, su nombre implica utilizar un sólo tipo de conector en lugar de tener diferentes conectores y protocolos Axelson (2005). Una GUI es una aplicación diseñada e implementada para un usuario pueda interactuar fácilmente con una computadora, algunas aplicaciones requieren el uso de una librería de enlace dinámico (dll) para poder comunicarse con un dispositivo externo a una computadora. La arquitectura .NET, es una plataforma de desarrollo para la implementación de aplicaciones.

El circuito integrado, puede ser implementado en 3 etapas:

### Paso uno

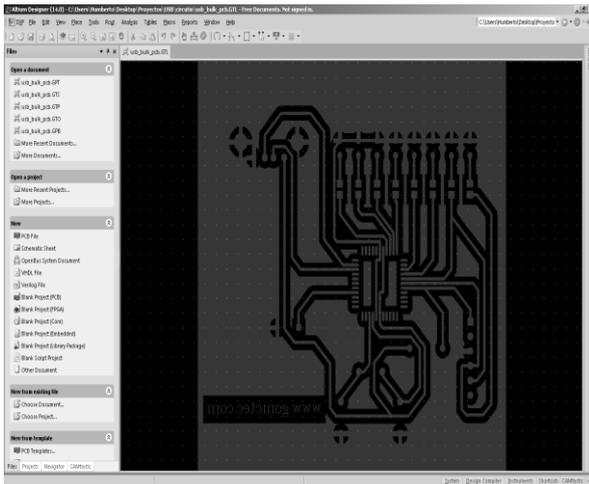
Simulado en un software: en este caso la herramienta que se utilizó Multisim (National Instruments Corporation, 2014), el objetivo es identificar y confirmar los componentes que formarán el circuito, aquí se simulan diferentes esquemas y escenarios. Posteriormente, se mencionan los componentes que se proponen para el circuito, cabe aclarar que esta es una propuesta para un circuito solamente, pero se pueden diseñar diferentes esquemas y valores, el objetivo es comprender el funcionamiento del circuito.

En este caso se propone utilizar un microcontrolador PIC, de acuerdo con las configuraciones propuestas por Predko (2001), Iovine, (2000) y Lynch, (1991), considerando la ventaja económica que ofrece diseño con este tipo de circuitos integrados. Este diseño es circuito básico, el microcontrolador PIC es programado para que según lo que reciba por el Puerto USB, encienda los diferentes LEDs.

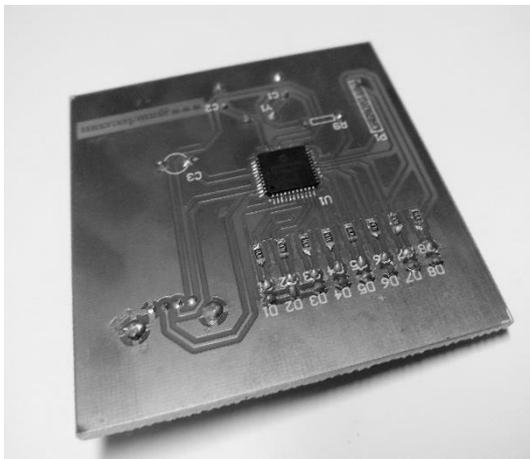
Los componentes que se utilizaron en este proyecto, son los siguientes: Microcontrolador PIC18F4550-I/PT, capacitores, resistencias, cristal, LEDs, set de pines, conector USB.

### Pasos 2 y 3

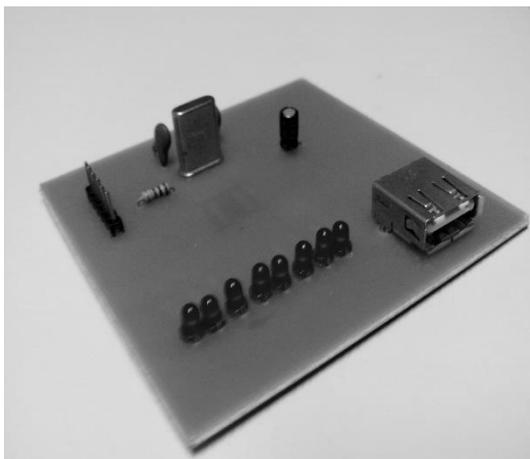
Realizar el circuito real utilizando un protoboard y/o implementarlo en una tarjeta de circuito impresa (PCB, Printed Circuit Board). Una vez que se realizó el diseño del circuito y se analizó su comportamiento, se procede a conseguir los componentes, se puede implementar el circuito en un protoboard, o se puede implementar ya en una tarjeta de PCB, ambas propuestas proporcionan habilidades prácticas y conocimientos para los estudiantes, que a su vez puede sugerir a variaciones en el diseño como lo sugieren Boylestad, Robert., Nashelsky, Louis. & Mendoza Barraza, C. (2003). En caso de querer utilizar una herramienta de software para el diseño del circuito integrado en una tarjeta PCB, se puede utilizar Altium Designer (2014), ZenitPCB (Stortini, 2012), Eagle (CadSoftusa, 2013), entre otros. En las figuras de la uno a la tres, se muestran los diseños que representan el circuito integrado.



**Figura 1** Diseño de PCB en software.



**Figura 2** Diseño en una PCB (chasis).



**Figura 3** Diseño en una PCB (componentes).

## Implementación de librería en .NET para uso de puerto USB

Para poder integrar el circuito integrado con la interfaz gráfica, se desarrolló una Librería de Enlace Dinámico (dll), en nuestro caso se llamó picusbcom.dll, en el mismo entorno de desarrollo de software, esta librería se debe almacenar en el directorio de trabajo donde se encuentra la aplicación, esta es un punto de referencia ya que la librería es la que recibe las instrucciones de la GUI, y de ahí será la encargada de comunicarse con la librería que el sistema operativo utiliza para manipular físicamente el puerto USB, para que las instrucciones sean enviadas electrónicamente, a través de un medio físico (cable) al circuito. La librería usada es mpusbapi.dll, cuya función es enviar y recibir datos a través del puerto USB. En este caso es muy importante analizar el proceso de interacción entre un el Sistema Operativo y un dispositivo periférico:

Cualquier Sistema Operativo generalmente recurre al uso de librería, estas cuentan con las instrucciones de bajo nivel para interactuar con cada dispositivo externo, lo que implica tener la librería adecuada para cada dispositivo. Dado lo anterior se puede recomendar buscar si el proveedor del dispositivo cuenta con alguna aplicación de software, que permita comprender el uso del dispositivo y su librería, el microcontrolador PIC, nos ofrece una aplicación de este tipo, lo que beneficia bastante, esto nos permitió poder analizar y conocer a mejor detalle el uso del Circuito Integrado para la etapa de análisis del desarrollo del software, con el objetivo de poder desarrollar una librería que enlace la GUI con la librería del dispositivo (mpusbapi.dll).

La interacción con el circuito primero se logró mediante el diseño de una Interfaz Gráfica de Usuario, la cuál puede ser mediante herramienta de Microsoft Office, que resulte simple para usar y entender el potencial de un PIC, este proceso se describirá más adelante. Se encontró que con Microsoft Visual C# 2013, se puede desarrollar un componente que puede ser utilizado en las herramientas de Microsoft Office, lo que da las capacidades necesarias para el desarrollo de la aplicación, un factor relevante es que esta plataforma de desarrollo no representa un costo significativo para los estudiantes (es de libre uso).

Para fines didácticos se puede adaptar un modelo de desarrollo de software, ya que se tiene como análisis y requerimiento el prototipo del circuito diseñado, como diseño la representación gráfica de las funciones del circuito, se lleva a cabo la implementación de la aplicación, se pasa por el proceso de pruebas y finalmente el mantenimiento, que puede incluir adecuaciones. El lenguaje de desarrollo por el que se optó fue C#, ya que históricamente, el desarrollo que implica interacciones con periféricos han sido solucionados en algún estándar derivado de C#, como lo ha propuesto Ceballos (2007 y 2011), se puede apreciar la interfaz gráfica así como parte del código en las figuras cinco y seis.

## Metodología

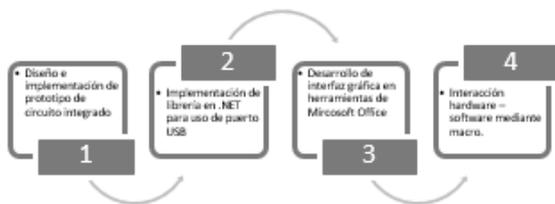


Figura 4 Metodología de desarrollo

Para el cumplir con el objetivo, se desarrolló una librería, con el objetivo de ser utilizada con un componente COM. A continuación se describen las acciones realizadas.

1. La librería se denomina picusbcom.
2. Se renombra el namespace a picusb. namespace picusb
3. Se renombra la clase a picusbcomclass unsafe public class picusbcomclass
4. Se agrega las siguientes líneas de código para hacer compatible el código con los componentes COM.

```
[ComVisible(true)]
[ClassInterface(ClassInterfaceType.AutoDual
)]
```

Antes de la definición de la clase.

```
[ComVisible(true)]
```

Antes de la definición de la función puerto esto con el fin de hacerla visible desde fuera de la clase.

5. En las propiedades del proyecto y en la pestaña "Build" marcamos las opciones "Allow unsafe code" y "Register for COM interop".

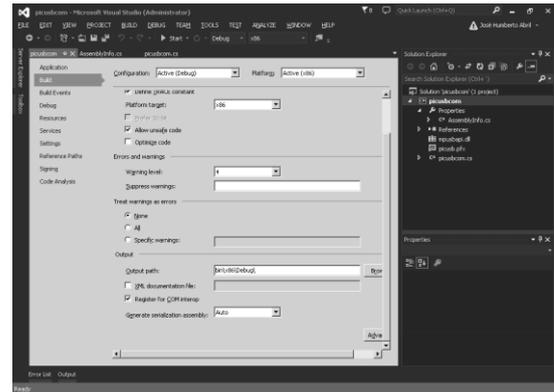


Figura 5 Register for COM interop.

6. Generar la librería.

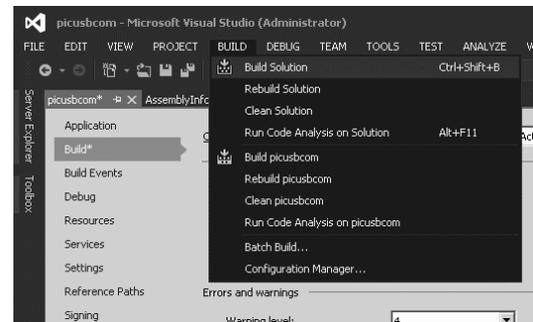


Figura 6 Generar la librería.

## Integración de herramientas de Microsoft Office mediante la creación de macros para interactuar con prototipo de Circuito Integrado

1. Crear un archivo Excel compatible con macros extensión xlsm.

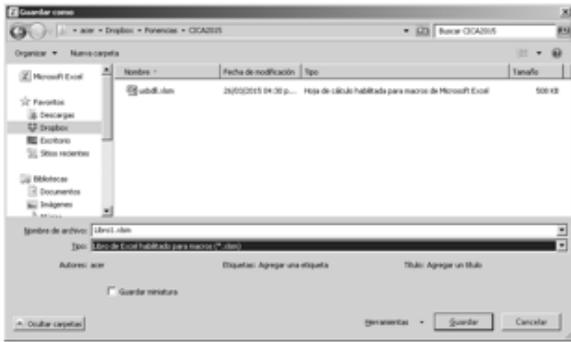


Figura 7 Excel compatible con macros extensión xlsxm

2. En caso de que la opción programador no este habilitada, se procede a mostrarla dando clic derecho sobre la cinta, luego en la opción “Personalizar la cinta de opciones...”, después se marca la casilla “Programador” y clic en Aceptar, de esta manera ya tenemos habilitada la pestaña Programador que es donde se encuentran los elementos necesarios para realizar el diseño de nuestra aplicación.



Figura 8 Personalizar la cinta de opciones.



Figura 9 Cinta de Opciones

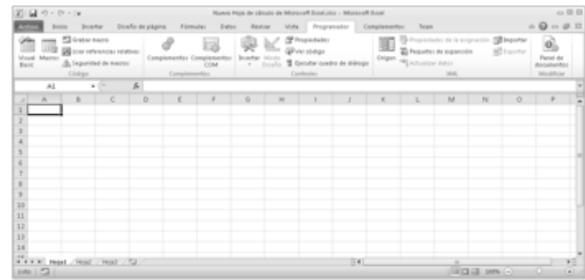


Figura 10 Menú Programador

3. El siguiente paso es desarrollar el diseño de la interface, para esto eliminamos el color de borde de todas las celdas, agregamos dos cuadros de grupo con los títulos de “Control de Leds” y “Diseño”.

4. Al control Cuadro de grupo “Control de Leds” se le agregan nueve objetos de tipo CommandButton, a los objetos del 1 a 8 se les modifica su propiedad Caption a “Led 1 On”, “Led 2 On”, “Led 3 On”, “Led 4 On”, “Led 5 On”, “Led 6 On”, “Led 7 On” y “Led 8 On”, respectivamente, al objeto 8 la propiedad Caption se le da un valor de restart.

5. Al control cuadro de grupo “Control de Leds” se le agrega un objeto de tipo Label, a su propiedad Caption se le da el valor de “Salida: 0”.

6. Al control cuadro de grupo denominado “Diseño” se le agregan dos imágenes, en nuestro caso hacen alusión al prototipo de circuito, quedando el diseño como se ve en la figura 8.

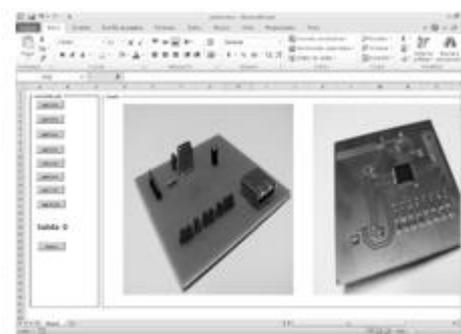


Figura 11 Diseño

7. El siguiente paso es la programación del macro que cargara la librería dll, en la pestaña programador hacemos clic en el botón “Visual Basic”.

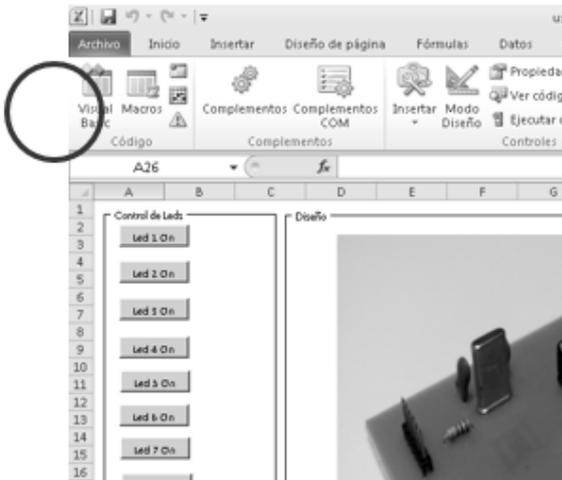


Figura 12 Visual Basic

8. Por otro lado, en la ventana del editor de macros, agregamos la referencia a la biblioteca picusbcom.dll.

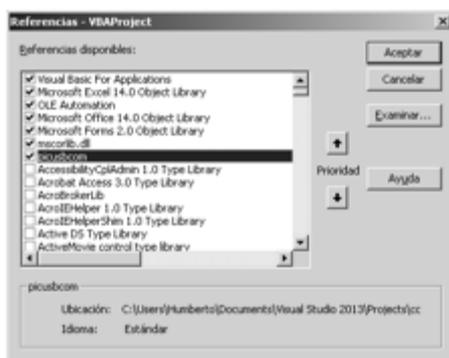


Figura 13 Macros

9. A continuación declaramos un variable de tipo entera llamada salida, que controlará el dato a enviar al puerto usb.

Dim salida As Integer

10. Declaramos una variable llamada “o” de tipo picusbcom la cual permite usar las funciones de la librería.

Dim o As New picusbcom.picusbcomclass

11. Se programa la función cambia que recibe como parámetro una referencia a un objeto de tipo CommandButton, y no regresa ningún valor, el objetivo de la función es aumentar o disminuir el valor de la variable salida, enviar el valor al puerto, modificar la etiqueta del botón y el texto del objeto Label.

Private Sub cambia(ByRef b As CommandButton)

Dim a() As String: a = Split(b.Caption, " ")

If a(2) = "On" Then

salida = salida + (2 ^ (CInt(a(1)) - 1))

b.Caption = a(0) & " " & a(1) & " " & "Off"

Else

salida = salida - (2 ^ (CInt(a(1)) - 1))

b.Caption = a(0) & " " & a(1) & " " & "On"

End If

Label1.Caption = "Salida: " & salida

o.Puerto (salida)

End Sub

12. Se programa la función CommandButton9\_Click que no recibe ningún parámetro y tiene como objetivo reiniciar la variable salida, restablecer las etiquetas de los botones y mandar el valor de 0 al puerto usb limpiando así los datos del mismo.

Private Sub CommandButton9\_Click()

CommandButton1.Caption = "Led 1 On"

CommandButton2.Caption = "Led 2 On"

CommandButton3.Caption = "Led 3 On"

CommandButton4.Caption = "Led 4 On"

CommandButton5.Caption = "Led 5 On"

CommandButton6.Caption = "Led 6 On"

CommandButton7.Caption = "Led 7 On"

CommandButton8.Caption = "Led 8 On"

Label1.Caption = "Salida: 0"

salida = 0

o.Puerto (salida)

End Sub

13. Se programa una función por cada botón del 1 al 8, estas funciones no reciben parámetros ni regresan datos, su objetivo es llamar a la función cambia.

```
Private Sub CommandButton1_Click()
    cambia CommandButton1
End Sub
Private Sub CommandButton2_Click()
    cambia CommandButton2
End Sub
Private Sub CommandButton3_Click()
    cambia CommandButton3
End Sub
Private Sub CommandButton4_Click()
    cambia CommandButton4
End Sub
Private Sub CommandButton5_Click()
    cambia CommandButton5
End Sub

Private Sub CommandButton6_Click()
    cambia CommandButton6
End Sub
Private Sub CommandButton7_Click()
    cambia CommandButton7
End Sub
Private Sub CommandButton8_Click()
    cambia CommandButton8
End Sub
```

14. Se graba la aplicación.

15. Se repiten los pasos del 1 al 14 para Word, PowerPoint y Access.

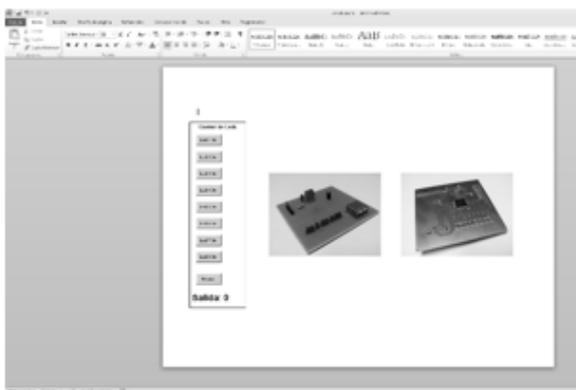


Figura 14 Diseño para Microsoft O. Word

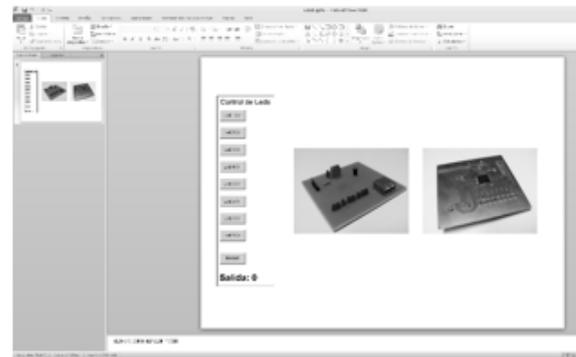


Figura 15 Diseño para Microsoft O. Power Point.

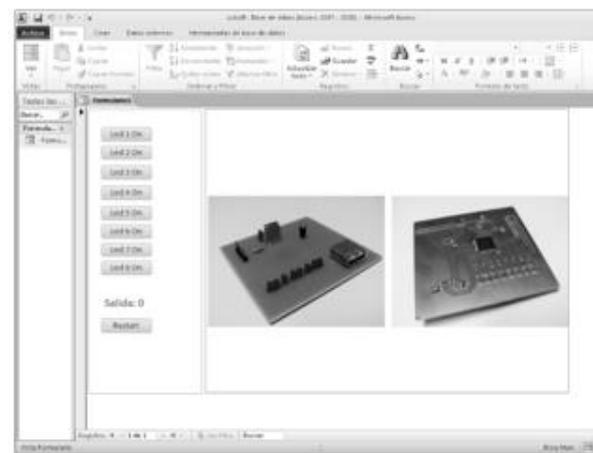


Figura 16 Diseño para Microsoft O. Access

## Resultados

Como resultado se tiene una guía para llevar a cabo una didáctica para el desarrollo del aprendizaje enfocado a la teoría de Circuitos Integrados Programables, Librerías para el manejo de dispositivos periféricos a través del puerto USB, así como el desarrollo de software.

1. Se programan aplicaciones para controlar con PowerPoint, Excel Access y Word. Se desarrolló un software cuya Interfaz Gráfica representa el circuito y su función, mandar encender una serie de LEDs.

2. Se desarrolló una librería a la cual se conecta la Interfaz Gráfica, para que ésta envíe información a la librería que utilice el Sistema Operativo para enviar señales al Circuito Integrado.

3. El circuito integrado que se desarrolló tiene como entrada señales a través un puerto USB, según la instrucción que recibe, manda a encender LEDs.

4. Se creó un proyecto que puede ser usado como base para el desarrollo de circuitos o programas más complejos y robustos que dan mayor funcionalidad.

5. Se diseña una metodología de trabajo integradora que permite que varias áreas de las ingenierías se integren en un sólo proyecto, por lo que cada área puede trabajar directamente los temas de enfocadas en su especialidad.

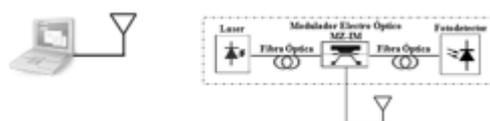
6. Diagrama a bloques de las etapas o módulos del proyecto finalizado. El desarrollo de la aplicación es estructura .NET, GUI; DLL para interactuar con USB; Librería del Sistema Operativo para uso de USB; y por último, Circuito electrónico, como se aprecia en la figura 17.



**Figura 17** Diagrama a bloques de ejecución del proyecto finalizado.

7. Como aplicaciones potenciales del desarrollo de este tipo de trabajos, se plantea posteriormente controlar sistemas optoelectrónicos inalámbricos y de fibra óptica que permitan la generación y distribución de señales de microondas tal y como se observa en el diagrama a bloques de la figura 18.

El esquema estará conformado por un láser mono modo, un modulador de intensidad electróptico, un Fotodetector de respuesta rápida y un par de antenas que podrán ser diseñadas para las frecuencias de operación del modulador electróptico



**Figura 18** Diagrama a bloques del modelo propuesto.

## Conclusión

El proyecto da como resultado una dinámica de clase en la que los estudiantes identifican fácilmente las funciones de un Circuito Integrado Programable, además pueden interactuar con él, de tal manera que pueden crear diferentes escenarios, reforzando el las áreas de electrónica y mecatrónica; por otro lado, el área de desarrollo de software, es beneficiada ya que de igual manera se pueden plantear diferentes diseños o problemáticas, integrando finalmente en el área de comunicaciones, todo esto impactando fuertemente en el área académica. La actividad permite plantear diversos escenarios, entre los que podemos ejemplificar: domótica, administración y manipulación de sensores, control y automatización de manera remota, entre otros.

Se recomienda realizar un trabajo multidisciplinar de las carreras de electrónica, informática, mecánica y mecatrónica para llevar el proyecto mediante una materia Integradora, se aplican competencias genéricas, transversales y específicas.

## Referencias

Adams, J. (23 de Enero de 2014). Compute Module development kits now available! Recuperado el 7 de Julio de 2014, de Raspberry Pi Foundation: <http://www.raspberrypi.org/compute-module-development-kits-now-available/>

- Altium. (2014). <http://www.altium.com/en/products/altium-designer>. Recuperado el 7 de Julio de 2014, de Altium: <http://techdocs.altium.com/display/ALIVE/AltiumLive>
- Adafruit. (16 de Julio de 2013). Adafruit. Recuperado el 1 de Abril de 2015, de Adafruit: <https://learn.adafruit.com/boarduino-kits>
- Arduino. (2014). What Arduino can do. Recuperado el 7 de 07 de 2014, de Arduino: <http://arduino.cc/>
- Axelsson, J. (2005). USB Complete: Everything You Need to Develop Custom USB Peripherals (3ª ed.). USA: Lakeview Research.
- Boylestad, R., Nashelsky, L. & Mendoza Barraza, C. (2003). Electrónica: teoría de circuitos y dispositivos electrónicos. México, D.F., México: Pearson Educación de México.
- CadSoftusa. (8 de Enero de 2013). EAGLE VERSION 6.4. Recuperado el 7 de Julio de 2014, de CadSoft USA: <http://www.cadsoftusa.com/2013/01/eagle-version-6-4-ab-sofort-erhaltlich/>
- Ceballos, F. J. (2007). Microsoft C#: Curso de programación (3ª ed.). México, D.F.: Alfaomega, Ra-Ma.
- Ceballos, F. J. (2011). Enciclopedia De Microsoft Visual C# (3ª ed.). México: Alfaomega, Ra-Ma.
- Iovine, J. (2000). PIC Microcontroller Project Book. USA: McGraw-Hill.
- Jeff. (31 de Agosto de 2013). Ktuluino - How to Build Your Own Arduino Clone. Recuperado el 1 de Abril de 2015, de Techunboxed: <http://www.techunboxed.com/2013/08/ktuluino.html>
- Lab, L. P. (Marzo de 2013). All about Moteino. Recuperado el 1 de Abril de 2015, de Low Power Lab: <http://lowpowerlab.com/moteino/>
- Lynch, M. L. (1991). Computer Numerical Control for Machining (First Edition) (1ª ed.). USA: McGraw-Hill.
- Mano, M. M. (1982). Lógica Digital y Diseño de Computadores. (J. A. Valbuena Z., Trad.) Naucalpan de Juárez, Edo. de México, México: Prentice Hall.
- Meza Ibarra, I. D.; Abril García, J. H.; y Elizarrarás Quiroz, J. J. . (2014). Desarrollo de circuito integrado como prototipo e Implementación de librería en .net para uso de puerto usb con fines didácticos. Congreso Universitario Ciudad Juárez, 336. doi:ISBN 9786078262052
- Microchip Technology Inc. (2014). Why Buy PIC® Microcontrollers (MCUs) And. Recuperado el 7 de Julio de 2014, de Microchip Technology Inc.: <http://www.microchip.com/pagehandler/en-us/products/picmicrocontrollers>
- National Instruments Corporation. (2014). NI Multisim. Recuperado el 7 de Julio de 2014, de National Instruments Corporation: <http://www.ni.com/multisim/esa/>
- Predko, M. (2001). Programming & Customizing PICmicro Microcontrollers (2 ed.). New York, USA.: McGraw-Hill.
- Pressman, R. S. (2002). Ingeniería de Software, un enfoque práctico. México: McGraw-Hill. ISBN 0-07-709677-0
- Picuiño. (2 de Enero de 2012). Picuiño. Recuperado el 1 de Abril de 2015, de Picuiño: <https://sites.google.com/site/picuiño/>

SpikenzieLabs. (2011). What is a Prototino. Recuperado el 1 de Abril de 2015, de SpikenzieLabs:  
<http://spikenzielabs.com/SpikenzieLabs/Prototino.html>

Stortini, M. (2012). Tutorial : New PCB Project. Recuperado el 7 de Julio de 2014, de ZenitPCB:  
[http://www.zenitpcb.com/Tutorial/New\\_Project.html](http://www.zenitpcb.com/Tutorial/New_Project.html)

Timmis, H. (Agosto de 2013). \$9 ARDUINO Compatible STARTER KIT - Anyone can learn Electronics. Recuperado el 2 de Abril de 2015, de Indiegogo:  
<https://www.indiegogo.com/projects/9-arduino-compatible-starter-kit-anyone-can-learn-electronics>

UDOO. (s.f.). Tutorials. Recuperado el 7 de Julio de 2014, de UDOO:  
<http://www.udoo.org/tutorials/>