

Comunicación Serial entre un Robot Móvil y la PC Mediante LabWindows/CVI

Serial communication between a mobile robot and PC using LabWindows / CVI

Luis A. Arellano Zea, Daniel G. Martínez Martínez, Bruno E. Vargas Tamani

Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos

Resumen- El presente trabajo consiste en establecer una comunicación serial bidireccional entre un robot móvil y la computadora. Esta comunicación tiene como objetivo la interacción de datos en tiempo real entre ambos de tal manera que la PC pueda mediante una interfaz gráfica registrar los parámetros del robot (ángulo, coordenadas, velocidad, etc.) y además pueda recibir órdenes (comandos de función) desde la PC. Para cumplir con dicho objetivo el robot cuenta con un microcontrolador dsPIC30F4011 como unidad de procesamiento, el cual se encarga de comunicar en forma serial con la PC. Los datos enviados desde el robot hacia la PC son visualizados mediante una interfaz gráfica desarrollada en LabWindows/CVI 2010. Así mismo, se enviarán datos al robot desde la PC los cuales serán observados mediante leds instalados en el robot móvil.

Abstract- This paper is to establish a bidirectional serial communication between a mobile robot and computer. This communication aims at data interaction in real time between the two so that the PC through a graphical interface can record robot parameters (angle, coordinates, speed, etc) and also through a graphic interface, the robot can also receive commands (command functions) from the PC. To comply with this goal the robot has a microcontroller dsPIC30F4011 as processing, which is responsible for communicating with the PC serially. The data sent from the robot to the PC are visualized through a graphic interface developed in LabWindows/CVI2010. Likewise, data sent from the robot which will be observed by means of LEDs installed on the mobile robot.

Palabras Claves - Robot, comunicación serial, UART, LabWindows, dsPIC30F4011.

Key Words - Robot, serial communication, UART, LabWindows, dsPIC30F4011.

I. INTRODUCCIÓN

Los robots móviles deben ser monitoreados en su etapa de implementación y operación, dado que si no se visualizan los parámetros del proceso en la fase de pruebas o durante la ejecución no se podrá hacer un seguimiento en tiempo real del estado del sistema. El monitoreo se puede realizar con un LED, display, LCD, PC o cualquier otro dispositivo de visualización. El monitoreo mediante PC es una de las más ventajosas ya que permite visualizar los valores de los parámetros en la pantalla y dichos valores pueden ser representados mediante gráficos, cuadros, etc. Además no sólo sirve para monitorear el proceso sino también poder controlarlo mediante un programa desarrollado en él mismo. Otra de las ventajas de utilizar una PC es que podemos almacenar la información monitoreada en una base de datos o en archivo de texto.

Para monitorear un hardware externo mediante PC se debe establecer una comunicación entre ambos. En nuestro caso el robot cuenta con un microcontrolador dsPIC30F4011 con capacidad de comunicación serial, que es uno de los tipos de comunicación más usados para este propósito y además ha sido estandarizado tanto a nivel comercial como industrial en los últimos años.

Se requiere un programa tanto en la PC como en el microcontrolador para la transmisión y recepción de datos. En el caso de la PC el programa puede ser desarrollado en cualquier lenguaje de programación que soporte dichas características. Siendo el lenguaje C uno de los más estables y robustos para dicha aplicación. Además del lenguaje se necesita un buen entorno de desarrollo, uno de ellos es LabWindows CVI 2010 el cual está diseñado para el monitoreo y control de procesos, además posee una gran variedad de funciones para la implementación de cualquier tipo de aplicación. Para el caso del microcontrolador se

¹Luis Alberto Arellano Zea, e-mail: luis.arellano09@gmail.com

²Daniel Gustavo Martínez Martínez, e-mail: D.GMartinez.M@gmail.com

³Bruno Elio Vargas Tamani, e-mail: bvargast@unmsm.edu.pe

Recibido: Noviembre 2011 / Aceptado: Diciembre 2011

utilizó el lenguaje C con el compilador C30 en el entorno de MPLAB de Microchip.

II. ESQUEMA GENERAL

Se cuenta con un robot móvil capaz de posicionarse en cualquier coordenada de un plano determinado. El funcionamiento del sistema de comunicación entre el robot y la PC se muestra en la Fig.1. El robot en todo instante actualiza los valores de los parámetros (ángulo, posición, velocidad, etc.) mediante un programa de control que ejecuta el microcontrolador. Estos datos son los que se requieren para monitorear el sistema y enviados vía comunicación serial a la PC y son leídos mediante LabWindows, el cual muestra los valores de los parámetros recibidos en la Interfaz.

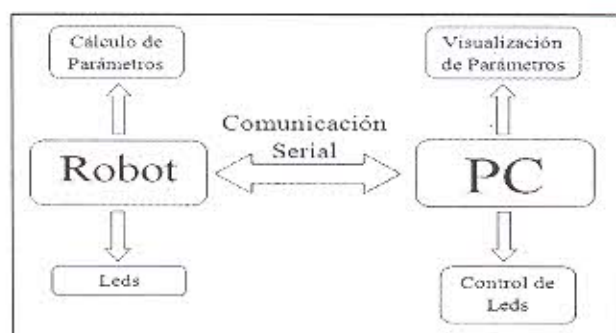


Fig. 1. Esquema general de funcionamiento

Además desde la PC requerimos enviar datos al robot, como son enviar coordenadas, consignas de control, instrucciones de movimiento, etc. que serán verificados mediante el encendido y apagado de leds instalados en el robot.

III. CARACTERÍSTICAS DE LA COMUNICACIÓN SERIAL DEL DSPIC30F4011

El dsPIC30F4011, [1], posee un módulo UART (Universal Asynchronous Receiver Transmitter), el cual se encarga de la comunicación serial del microcontrolador. Este módulo tiene las siguientes características [2]:

- Sistema full-dúplex asíncrono.
- Compatibilidad con interfaces RS-232.
- Datos transmisión de 8 o 9 bits.
- Paridad par, impar o ninguna (para datos de 8 bits).
- 1 ó 2 bits de parada.
- Generador de tasa de baudios con prescaler de 16 bit totalmente integrado.

- Rango de tasa de baudios entre 29bps a 1.875Mbps.

En la Fig.2 se puede observar el diagrama de bloques simplificado del UART, donde el transmisor y el receptor del UART son independientes entre sí, ya que es una comunicación full-dúplex. Además ambos dependen del generador de tasa de baudios.

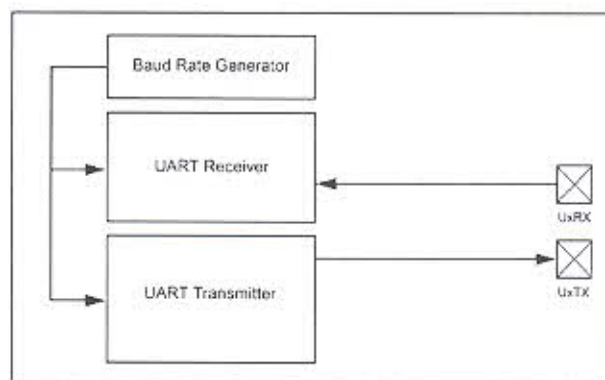


Fig. 2. Diagrama de bloques simplificado del UART

IV. CONFIGURACIÓN Y PROGRAMACIÓN DEL DSPIC30F4011

A. Configuración del sistema, UART, puertos y Timer.

La configuración del microcontrolador dsPIC30F4011 se realizó en función a los registros de control de sus módulos (UART, I/O, Timer, etc.) que se encuentran en su datasheet.

Para que el microcontrolador pueda funcionar se tiene que incluir la librería del microcontrolador (1), la librería del UART (2), donde se encuentran las funciones para la transmisión y recepción serial, y la librería STRING (3) que permite trabajar con arreglos tipo string (cadena de caracteres):

```

#include <p30f4011.h>           (1)
#include "UART.h"              (2)
#include "string.h"            (3)
  
```

Las características de ejecución del microcontrolador se hacen a través de los fuses. El primer fuse que se configura es el PLL (Phase Locked Loop), el cual multiplica el valor de la frecuencia del oscilador (para nuestro caso será 10MHz) para obtener frecuencias mayores [4], además dicho fuse forma parte del cálculo de la frecuencia de trabajo del microcontrolador, la cual se haya manera siguiente:

$$\text{frecuencia de trabajo} = F_{cy} = \frac{F_{osc} \times PLL}{4}$$

$$F_{cy} = \frac{10MHz \times 8}{4}$$

$$F_{cy} = 20MHz$$

Además se configura el watch dog timer como deshabilitado (5), el MCLR estará habilitado (6) y el código no estará protegido (7):

```
_FOSC(CSW_FSCM_OFF & XT_PLL8); (4)
```

```
_FWDTP(WDT_OFF); (5)
```

```
_FBORPOR(PBOR_OFF & MCLR_EN); (6)
```

```
_FGS(CODE_PROT_OFF); (7)
```

Es necesario definir contantes asignados a los nombres de los pines de I/O (8 a 10) para que al momento de programar sea sean llamados de una manera más amigable:

```
#define output_led1 _LATD2 (8)
```

```
#define output_led2 _LATF0 (9)
```

```
#define output_led3 _LATF1 (10)
```

Para la configuración del puerto serial del microcontrolador se utilizaron los registros del módulo UART [4]. La tasa de baudios requerida fue de 115200 y se configuró mediante el registro BRG (12) el cual se obtiene de la siguiente manera:

$$UxBRG = ((FCY/Baud Rate deseado)/16) - 1$$

$$UxBRG = ((20000000/115200)/16) - 1$$

$$UxBRG = [9.85] = 10$$

La paridad y el bit de parada se configuración en (13) y (14) respectivamente:

```
void init_uart() (11)
```

```
{
```

```
U2BRG = 10; (12)
```

```
U2MODEbits.PDSEL = 0b00; (13)
```

```
U2MODEbits.STSEL = 0; (14)
```

```
}
```

Los puertos I/O se configuran mediante el registro _TRISX [5], cuando un bit del registro es igualado a 0 significa que será una salida digital tal como se muestra en (16 a 18). Estos pines son utilizados para los leds y requieren ser inicializados, esto se hará mediante el registro _LATX tan como se indica en (19 a 21):

```
void init_puertos() (15)
```

```
{
```

```
_TRISD2 = 0; (16)
```

```
_TRISF0 = 0; (17)
```

```
_TRISF1 = 0; (18)
```

```
_LATD2 = 0; (19)
```

```
_LATF0 = 0; (20)
```

```
_LATF1 = 0; (21)
```

```
}
```

El módulo Timer [6] se encarga de la temporización dentro del microcontrolador, el cual se empleará para generar interrupciones de tiempo para transmitir datos desde el robot a la PC. Para su configuración se seleccionó el clock interno (23), se habilitó el modo Gate (24), se escogió el prescaler de 256 (25), se inicializó el contador en 0 (26), se seleccionó el periodo del Timer mediante el registro PR1 (27), el cual se haya de la siguiente manera:

$$\text{Periodo } T: \quad PR1 = T \times F_{cy} / \text{Prescaler}$$

$$\text{Periodo 20ms: } PR1 = (20ms) (20MHz) / 256$$

$$\text{Periodo 20ms: } PR1 = 1562$$

Luego se le asignó prioridad de interrupción 1 (28), se fijó el Flag de interrupción en 0 (29) y se habilitó la interrupción del timer (30).

```
void init_timer1() (22)
```

```
{
```

```
T1CONbits.TCS = 0; (23)
```

```
T1CONbits.TGATE = 0; (24)
```

```
T1CONbits.TCKPS=3; (25)
```

```
TMR1=0; (26)
```

```
PR1=1562; (27)
```

```
_T1IP = 1; (28)
```

```
_T1IF = 0; (29)
```

```
_T1IE = 1; (30)
```

```
}
```

A. Programación de la transmisión y recepción serial.

Este robot móvil posee 5 parámetros de medición, los cuales son obtenidos con el microcontrolador. Estos parámetros son:

- Ángulo
- Coordenada X
- Coordenada Y
- Velocidad instantánea
- Nivel de batería

Los parámetros son medidos en todo instante durante el funcionamiento del robot y son guardados en variables tipo float (reales de 32 bits). Estos datos

registrados deben de ser transmitidos por el puerto serial del microcontrolador y posteriormente recibidos por la PC. Como el dsPIC debe realizar funciones de control periódicamente, se elige un tiempo de 20ms como periodo de envío de información del robot a la PC. Para controlar el flujo de datos se implementó un comando de habilitación de la transmisión. El objetivo de este comando es que el robot no transmita información cuando no hay una conexión con la PC a través de la interfaz gráfica. Cuando se ejecute la interfaz gráfica se enviará un comando de función desde la PC al robot habilitando la transmisión de datos.

Desde la PC se puede enviar información al microcontrolador, estos datos enviados (encendido de leds y en una ampliación de funciones de movimiento y coordenadas de posición) accionarán instrucciones o comandos ya predefinidos dentro del robot. La información se envía por el puerto serial de la PC, el cual es controlado por la interfaz gráfica. El contenido de la información dependerá de los eventos realizados dentro de la interfaz (presionar un botón, Temporización por Timer, etc.), dicha información consistirá en caracteres alfanuméricos (char de 16 bits). Para que el robot pueda realizar estas tareas se deben implementar algoritmos dentro del microcontrolador. Dichos algoritmos se detallan en el siguiente diagrama de flujo mostrado en el Fig. 3.



Fig. 3. Diagrama de flujo general del dsPIC.

La recepción de información proveniente de la PC se ejecuta de acuerdo al algoritmo de la Fig. 4.

Primero se pregunta si hay un nuevo caracter en el buffer de recepción serial mediante una función de la librería UART. Si la sentencia es verdadera, se leerá el valor del buffer. Luego se procederá a desarrollar un selector con el dato obtenido y se ejecutará las sentencias de acuerdo al valor del caracter. Estas sentencias pueden ser encender o apagar los leds y habilitar o deshabilitar la transmisión de información.

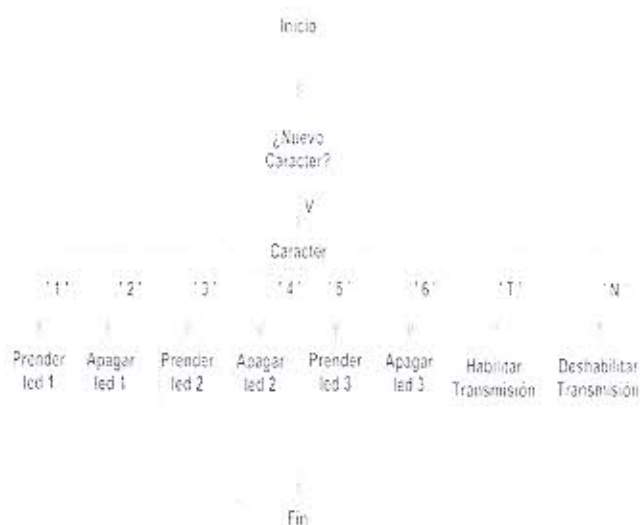


Fig. 4. Diagrama de flujo de la recepción del dsPIC

Para realizar las pruebas de comunicación asignamos valores conocidos a los parámetros del robot para que puedan ser verificados en el proceso de lectura en la PC por su similitud. Estos valores son asignados según la Fig. 5.



Fig. 5. Diagrama de flujo de cálculo de parámetros

La transmisión serial de datos del robot consiste en enviar los parámetros actualizados mediante la generación de una trama de información. Para generar esta trama se utiliza el algoritmo de la Fig. 6. Se crea un vector de 37 elementos del tipo alfanumérico (char de 16 bits). Como los valores de los parámetros son reales y poseen valores decimales se multiplica por 100 para que las dos primeras cifras decimales se

vuelvan enteras. Luego se procede a la formación de la trama asignando caracteres a los elementos del vector. Estos caracteres son los dígitos de los valores de los parámetros y también letras como cabecera de cada valor. Estas letras señalan el inicio y el final tanto de la trama como de los valores de los parámetros. Otra de las funciones de las letras es la validación de la trama al momento de recibir la información en la PC. Quiere decir que sintonizarán la información.

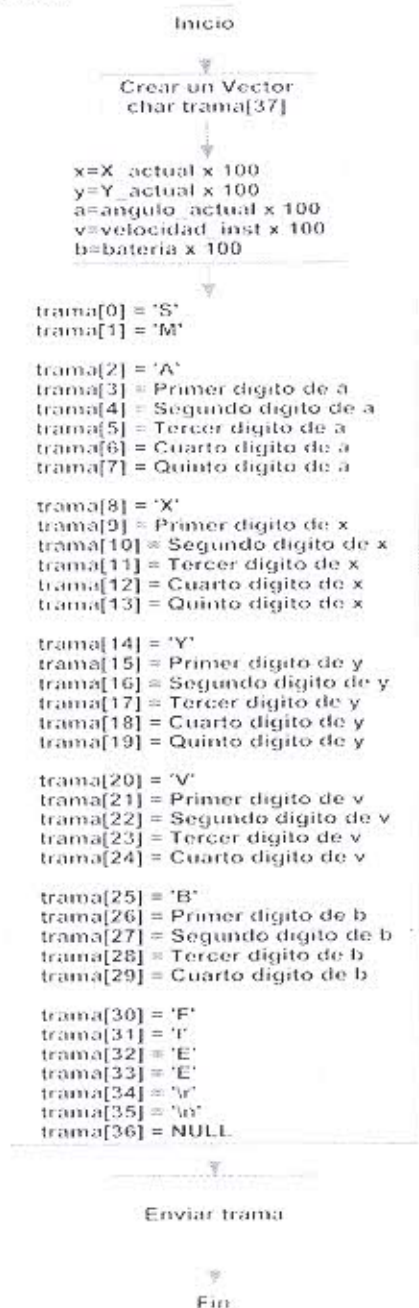


Fig. 6. Diagrama de flujo de la transmisión del dsPIC.

Las tramas tendrán la siguiente apariencia:

```

SMA10000X05000Y07000V0500B1300FIEE
SMA10000X05000Y07000V0500B1300FIEE
SMA10000X05000Y07000V0500B1300FIEE
SMA10000X05000Y07000V0500B1300FIEE
  
```

IV. CONFIGURACIÓN Y PROGRAMACIÓN DE LA INTERFAZ EN LABWINDOWS/CVI2010

LabWindows/CVI [7] es un entorno de desarrollo ANSI C integrado y comprobado que brinda una extensa variedad de herramientas de programación para crear aplicaciones de pruebas y control. Combina la durabilidad y capacidad de reutilización de ANSI C con una funcionalidad específica para ingeniería, diseñada para control de instrumentos, adquisición de datos, análisis y desarrollo de interfaz de usuario.

A. Configuración de la comunicación serial

La configuración de la comunicación serial del LabWindows se realiza mediante la siguiente función [8]:

```

OpenComConfig (           (31)
4, // Puerto              (32)
"COM4", // Nombre         (33)
115200, // Baudios         (34)
0, // Paridad              (35)
8, // Data bits            (36)
1, // Stop Bit             (37)
512, // Tamaño de buffer de entrada (38)
512 // Tamaño de buffer de salida (39)
);
  
```

Para la configuración se asignaron el puerto serial (32), el nombre del puerto (33), la tasa de baudios (34), la paridad (35), el tamaño de dato enviado (36), el bit de parada (37), el tamaño del buffer de entrada (38) y el tamaño del buffer de salida (39). Estos parámetros coinciden con los configurados en el dsPIC.

B. Programación de la interfaz usuario

LabWindows es un entorno de desarrollo que permite crear interfaces gráficas mediante el archivo con extensión .uir donde se diseña la apariencia de la GUI (Graphic User Interface). Luego se genera el código y se procede a programar. La programación se realiza en función a eventos, quiere decir que se ejecutan las instrucciones mediante una acción del usuario, como por ejemplo hacer click en un botón o la temporización de Timer.

La interfaz diseñada tiene dos funciones. La primera es la verificación de entrada de datos seriales

(trama de información enviada por el robot) para luego ser procesadas que se realiza mediante un evento tipo Timer cada 10ms. Este tiempo de temporización es el doble que el configurado en el robot para evitar errores de retardo. La segunda es la transmisión de datos desde la PC al robot. La interfaz tiene la apariencia que se observa en la Fig. 7. Los parámetros del robot se muestran en cada una de las casillas de texto, además se cuentan con botones para prender o apagar los leds en la placa del robot. Para un comprobar que los datos transmitidos por el robot son conformes se cuenta con un contador de tramas recibidas, contador y tasa de errores en el caso que los valores de los parámetros recibidos por la PC con concuerden que los programados en el dsPIC.



Fig. 7. Interfaz gráfica en LabWindows/CVI

Para que la GUI pueda funcionar se necesita tener un algoritmo, el cual en función a los eventos ejecute las operaciones deseadas. Este algoritmo se presenta en el diagrama de flujo de la Fig. 8. Cuando la interfaz inicia solo están habilitados los botones "Conectar" y "QUIT". Cuando se presiona el botón "Conectar" se hace la configuración del puerto serial de la PC y se habilitan los botones de "prender" y "apagar" leds. Además se envía serialmente el carácter "T" para habilitar el envío de datos del robot y se habilita el Timer para la recepción serial de datos cada 10ms. El programa estará constantemente preguntando si hay un evento y lo ejecutará. Cuando se presione el botón de "prender" o "apagar" se enviará un carácter ya predefinido el cual será recibido e interpretado por el robot de acuerdo a lo mostrado en la Fig. 4. De igual forma la recepción de datos se ejecutará cada 10 ms y realizará el algoritmo mostrado en la Fig. 9. Este algoritmo consiste en obtener primeramente el tamaño del dato que está en el buffer de entrada serial de la PC ya que no se cuenta con una función para indicar si hay un nuevo carácter en dicho buffer, si el tamaño del dato es igual a 36 se obtiene ese dato asignándolo a una variable (cadena de caracteres de 16 bit). Si el dato no está vacío y los valores de los caracteres de validación son

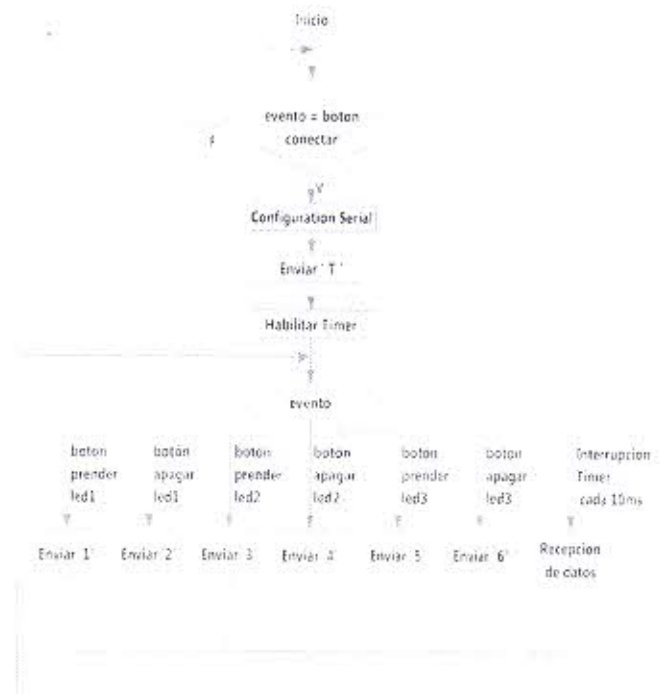


Fig. 8. Diagrama de flujo general en el Labwindows



Fig. 9 Diagrama de flujo de la recepción serial en Labwindows.

correctos entonces se extraerán los parámetros de la siguiente manera:

SMA10000X05000Y07000V0500B1300FIEE

A10000	Angulo	= 100.00 grados
X05000	X	= 50.00 cm
Y07000	Y	= 70.00 cm
V0500	Velocidad	= 05.00 m/s
B1300	Batería	= 13.00 volts

SM, A, X, Y, V, B, y FIEE son los caracteres de validación de la trama.

V. RESULTADOS

Para poder observar la trama enviada por el robot se utilizó el Hyperterminal, el cual es una aplicación de Windows para visualizar la comunicación serial de la PC. En la Fig. 10 se observa las tramas recibidas.



Fig. 10 Vista a través del HyperTerminal.

Luego de observar que los datos enviados por el robot son correctos se procederá a ejecutar la interfaz por 10 min donde se visualizarán los parámetros del robot, la cantidad de tramas recibidas y los errores si se diera el caso tal como se observa en la Fig. 11.

Luego de 10 min de operación no se obtuvieron errores en los valores de los parámetros recibidos, lo cual comprueba la fiabilidad del sistema.

La transmisión de datos desde la PC al robot encendiendo los leds de la placa del robot desde la interfaz. La placa se puede observar en la Fig. 12, donde se señala la posición de los leds. Se comprobó que todos los leds prenda independientemente y en conjunto tal como se muestra en la Fig. 13.

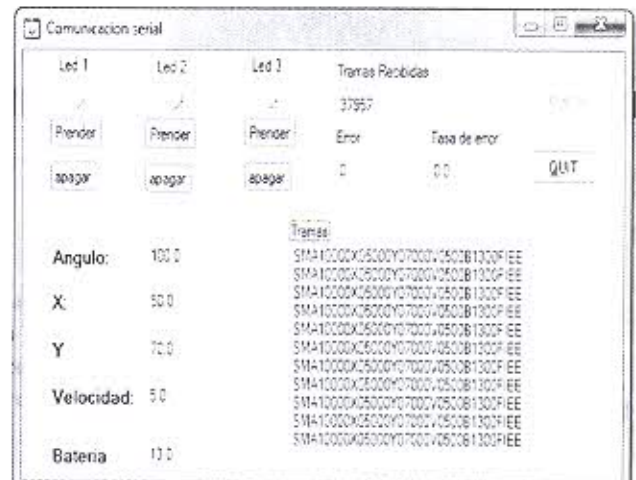


Fig. 11. Vista de la interfaz en funcionamiento

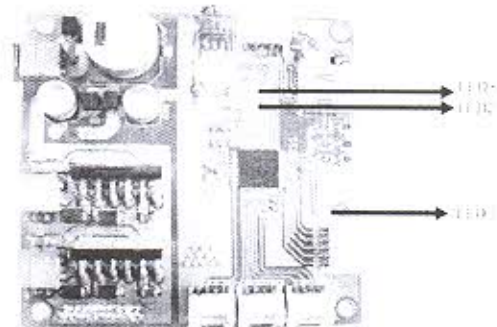


Fig. 12. Placa del Robot

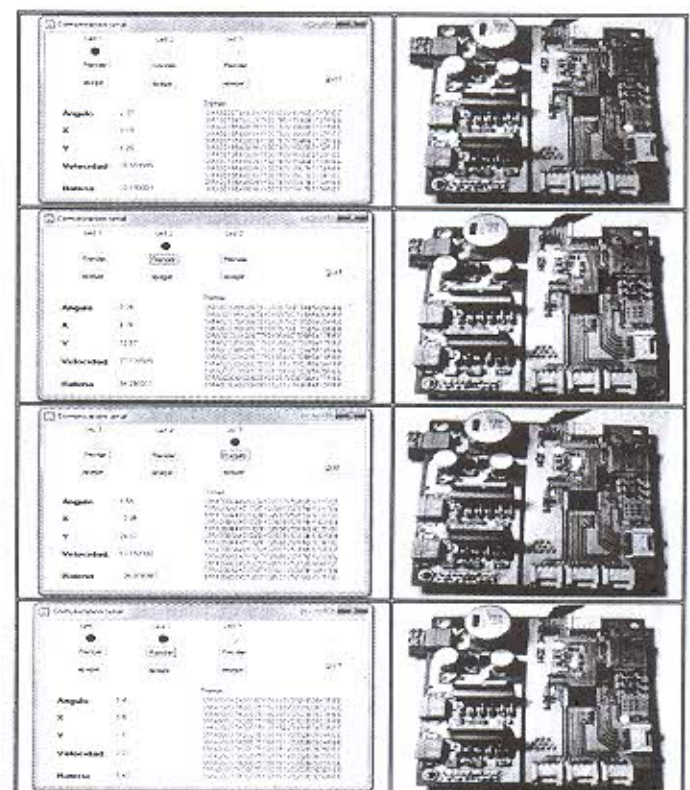


Fig. 13. Funcionamiento de los botones

VI. CONCLUSIONES

De acuerdo a los datos obtenidos por la PC y mediante los leds instalados en el robot se comprobó la factibilidad de la comunicación entre el robot y la PC de manera bidireccional tanto a nivel de monitoreo del robot como de control desde la PC.

El uso del dsPIC30F4011 como medio de comunicación del robot fue satisfactorio ya que no se presentaron problemas durante las pruebas. Además no se registraron retardos ni errores en la transmisión de los parámetros desde el robot a la PC.

El uso de Labwindows para implementar la interfaz gráfica dio muchas facilidades a nivel de soporte y robustez durante la ejecución del programa.

VII. REFERENCIAS

- [1] Microchip. "Datasheet dsPIC30F4011", pp 1-5. 2005.
- [2] Microchip. "Datasheet dsPIC30F4011, Section 19. UART", pp 1. 2004.
- [3] Microchip. "Datasheet dsPIC30F4011", pp 179. 2005.
- [4] Microchip. "Datasheet dsPIC30F4011, Section 19. UART", pp 3-7. 2004.
- [5] Microchip. "Datasheet dsPIC30F4011, Section 11. I/O", pp 3. 2005.
- [6] Microchip. "Datasheet dsPIC30F4011, Section 12. Timers", pp 2-8. 2005.
- [7] LabWindows/CVI. "Getting Started with LabWindows/CVI", pp 1. 2010.
- [8] LabWindows/CVI. "Standard Libraries Reference Manual", pp 256. 1998.