

Servocontrol reprogramable USB-HID para articulaciones de manipuladores y robots móviles

M. Fajardo Rendón, F. Herrera Armendia, C.A. Osorio Ayala y M. Benítez Benítez
CIUSC, Departamento de control, Universidad San Carlos y E.N.S.M., Departamento de Matemáticas
Edo. de México 02600, México
marktrony2k@gmail.com
Teléfono: 55-57700824 Ext. 176

Resumen—Los sistemas mecatrónicos requieren de interfaces de control que logren una precisión, paralelismo y portabilidad tanto en software como en hardware, por lo que el presente proyecto surge como una solución a esta demanda al permitir gobernar paralelamente en tiempo real hasta 30 servomotores que puedan servir en articulaciones de manipuladores industriales de distinta configuración. Así mismo, para robots móviles el sistema se encuentra diseñado para reprogramar su propia memoria Flash con rutinas configurables desde el software host de control mediante la tecnología USB Human Interface Device que no requiere drivers de instalación y al ser desconectado de la PC trabajar en modo autónomo para aplicaciones. Por último se explican tres aplicaciones en donde se ha comprobado el sistema.

Palabras clave: Servocontrol, robótica, USB, HID, Plug and Play, microcontroladores.

I. INTRODUCCIÓN

Para poder lograr que un robot humanoide de la categoría RoboONE logre tirar una patada, camine y se equilibre, además de sensores de retroalimentación, se requiere de un sistema controlador de servomotores que permita la manipulación de estos simultánea y coordinadamente mediante una tarjeta dedicada para un control digital directo.

Humanoides como Kondo KHR-2, Robonova y otros requieren un promedio de 24 servos para hacer uso de las variables de posición independientes que les permitan coordinar movimientos complejos logrando la combinación en espacio y tiempo de las posibles configuraciones de las articulaciones del robot para moverse, por lo que el número de articulaciones del robot será el número de grados de libertad.

Los servomotores son un tipo de motor usado en aplicaciones que requieran un posicionamiento exacto como los antes descritos y sobre todo en manipuladores industriales que necesitan llegar a una posición sin importar la actual. El efecto de carga en la dinámica de los servomotores radica en la máxima aceleración que puede obtenerse. Estos sistemas trabajan con pulsos que son variaciones de tiempo en tiempo que generan una aceleración y desaceleración del subsistema rotor, por lo que se considera a este como un mecanismo de absorción y generación de energía mecánica como el frenado.

Un servomotor estándar tiene una alimentación y es controlado mediante una señal PWM a una frecuencia de

oscilación, el tiempo comprendido entre ambos flancos sucesivos es igual al inverso de la frecuencias diversas, las comprendidas en el rango de 50 Hertz se trabajan 0.02 segundos, por lo que se cuenta con un periodo de trabajo de 20 milisegundos, y a su vez una variación en el periodo alto entre el 6% y el 15% que es una duración comprendida en un espectro de 0.5 a 2.5 milisegundos.

Las tarjetas comerciales estándar como PhidgetAdvanced servo 81 no permiten la variación del Ciclo de trabajo, por ejemplo los servomotores de la marca Futaba varían ente 0.3 a 2.3 milisegundos y la Hitec de 0.5 a 2.5 milisegundos.

La conversión PWM a posiciones del servomotor se logra mediante el envío de un pulso entre 0.7 y 2.3 milisegundos aproximadamente para cada extremo respectivamente. Cualquier otro punto geométrico se obtiene entre los 1.6 comprendidos que forman el espectro de operación, por lo que 1.5 milisegundos posiciona el centro.

En un manipulador industrial o cualquier configuración del robot se requiere mantener paralelamente a cada servomotor en una posición en tiempo real, por lo que se debe refrescar esta trama de PWM, de lo contrario perdería fuerza y la gravedad actuaría exponencialmente, se ocuparía más tiempo de proceso en el microcontrolador y se puede llegar a perder la precisión por la inestabilidad en los anchos de los pulsos recibidos. De manera que un sistema adecuado debe coordinar, calcular y ajustar dinámicamente los pulsos. Todo esto es embebido aún más al requerir un sistema desconectado de la computadora que se encuentre leyendo su propia memoria Flash, o en el momento en que deba grabar y acomodar las direcciones, pulsos y tiempos correspondientes.

Los microcontroladores permiten el uso de sus interrupciones por temporizadores, por lo que se pueden armar bancos de 8 servomotores por cada temporizador en un periodo de 20 milisegundos, con lo que se obtiene 2.5 milisegundos por cada servomotor y el objetivo en tarjetas reprogramables como la propuesta es calcular los altos y los bajos que forman el PWM y grabarlos en la memoria del controlador.

La descripción de las denominadas ventanas de tiempo de un temporizador del microcontrolador por servomotor se muestra en la figura 1.

La tecnología USB ha venido a revolucionar y “unificar en sistemas de cómputo la comunicación serial”⁴, al grado que ya se encuentran en proceso los sistemas de carga por entradas de este tipo.

Una de las definiciones de clase soportadas por la tecnología USB es la denominada Human Interface Device, la cual identifica al dispositivo conectado en el sistema como un hardware de interacción I/O con un usuario humano, su mayor ventaja es el simplificado proceso de instalación respecto a la unificación de protocolos y drivers mediante un paquete de autodescripción y la tecnología de autoasociación dinámica de comunicación bidireccional mediante un solo driver.

El protocolo HID se encuentra formado por el software de control denominado host y el hardware con su respectivo firmware HID que funge de interfaz con un humano.

Todo esto permite el auténtico concepto de “Plug and Play“, es decir, la nulificación del uso de drivers de instalación y el uso de un simple archivo ejecutable que es la aplicación host del mismo, además de la autonumeración e identificación en el sistema operativo por lectura del dispositivo de una combinación de único Vendor ID y Product ID.

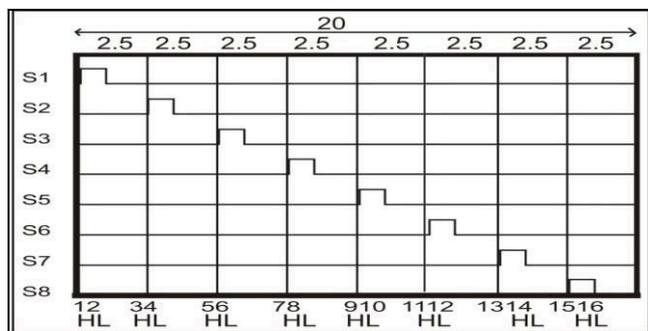


Fig. 1. Ventanas de tiempo del Temporizador1 del microcontrolador por servomotor

Para un usuario u operador de un sistema robótico es más intuitivo, cómodo y simple la conexión en una computadora con un dispositivo nuevo sin necesidad de realizar configuraciones e instalaciones, tal y como si se tratara de un mouse o teclado, además el programa no requiere por consiguiente de un registro en el sistema operativo y puede ser ejecutado en cualquier medio de almacenamiento al ser un programa portable.

El microcontrolador PIC 24FJ64GA004 cuenta con puertos específicos para lograr una comunicación USB, arquitectura de 16 bits, 35 pines I/O, 5 temporizadores de 5x16 bits, frecuencia de trabajo de 8 MHz a 32 KHz y una memoria reprogramable tipo Flash, misma que puede ser utilizada para el almacenamiento de programas o rutinas a realizar, por lo que el programa solo se encarga de acceder al banco de memoria correspondiente para recuperar su contenido.

II. EL SISTEMA DE SERVOCONTROL REPROGRAMABLE

Actualmente existen en el mercado tarjetas controladoras de servomotores, pero no se cuenta con una que permita una instalación fácil e intuitiva para el operador, quien muchas veces no es un ingeniero.

La tecnología USB de clase HID hace de esta un sistema fácil de instalar en computadoras con Windows XP ó superior.

Desde su hardware la tarjeta presenta un diseño compacto, no requiere alimentación externa al usar el voltaje de la computadora a través del USB y trabaja a una frecuencia de operación de 48Mhz, lo cual permite contar con los parámetros de un USB 2.0 a nivel arquitectura. La tecnología USB full speed de este microcontrolador soporta “Low-Speed (1.5 Mb/s) and Full-Speed (12 Mb/s) USB Funcionando como Host y Full-Speed USB como dispositivo usb”2

La característica más importante del sistema servocontrolador radica en que ésta cuenta con un software de control que no requiere de instalación, previniendo las políticas de algunas industrias de mantener la integridad del sistema operativo, además de no necesitar de drivers al ser un HID.

El software permite manejar en tiempo real desde una computadora los 30 servomotores mediante el arrastre intuitivo de los sliders, posteriormente se puede grabar una rutina para manipular un sistema mecatrónico mediante el almacenamiento en una pila de datos de cada posición a grabar.

Una vez grabado, el sistema puede ser desconectado de la computadora y alimentado independientemente por un voltaje externo, lo cual permite independencia y autonomía. Así mismo esta tarjeta puede ser implementada en sistemas mecatrónicos móviles o rutinarios como máquinas secuenciales industriales.

La reprogramación de la memoria Flash mediante el software host en la computadora permitirá entonces realizar al operador una calibración, prueba e implementación sin PC en un sistema mecatrónico, como un manipulador ó robot móvil y, si así se requiere, reprogramaciones continuas con un límite de 100,000 veces debido a las restricciones de una memoria Flash.

III. FIRMWARE

El kernel del sistema yace en el control embebido del microcontrolador, para lograr la comunicación I/O se establecieron las variables de flujo de datos denominadas Datain[64] y Dataout[64], encargadas de almacenar el tramado que contendrá los valores a manejar tanto del control por computadora como el autónomo de la memoria Flash.

Respecto a los servomotores se diseñaron las variables ServoPWM y encargada de almacenar el ancho de pulso modulado por servo, y ServoIDX, el cual permite numerar a los servos a utilizar por temporizador.

Se estableció que el Polling del sistema trabajara a 1 milisegundo, lo que lo hace ser de tipo USB 2.0, aunque la clase HID lo reduce a 10 milisegundos, por lo que el sistema se encuentra listo para trabajar a una mayor velocidad si los protocolos de velocidad en los descriptores USB fueran actualizados por el sistema operativo en un futuro.

Para usar la clase HID en los puertos V+ y V- del microcontrolador que fungen como TX y RX

respectivamente en los USB, se estableció un tamaño de buffer de 64 bits para manejar los Data I/O antes mencionados, las librerías PIC24_USB.H ó PIC18_USB.H respectivamente y USB.C, encargado de la descripción del driver modo USB.

Para la clase HID se hace uso de la librería USB_DES_HID.H, encargada del buffer, velocidad y el descriptor HID.

Internamente se realiza una declaración encargada de los 5 temporizadores mediante la función void Timer1_isr() que mediante el recorrido de una matriz de ocho elementos por temporizador calculando la diferencia entre las dos posiciones alto y bajo entre los intervalos de 2.5 milisegundos en PWM que presenta una onda cuadrada, ejecutándose recursivamente n veces la interrupción para cada servo .

El estado en tiempo real de cada canal de servomotores es controlado por la función booleanas Servo_ON[1,0], en donde tomará el valor 1 si se trabajará como PWM y 0 si se encuentra deshabilitado.

Las tres combinaciones posibles en la escritura de la EEPROM vienen dadas por la función de interrupción externa Int_Ext, asociadas a los puertos RB0, RB1 y RB2, donde para el primer puerto se asigna en la memoria la dirección hexadecimal 22, esto le indica al sistema que ejecute o no el programa, para el segundo la 5, la cual permite escribir rutinas y se cicla, para RB2 se utiliza la 55 que no ejecuta al cero y en 1 la rutina de movimiento por servomotor en bucle.

Para el manejo de control de USB se utiliza la función usb_int_cs(), habilita el hardware USB, revisa conexión y permite el uso de usb_task(), encargado de revisar la conexión, habilita el USB.

Cuando un dispositivo USB es conectado a cualquier puerto de una PC, este es enumerado mediante un USB raíz, si el sistema lo ha reconocido y la GUI de control indica que el nuevo hardware ha sido conectado correctamente, se ha autoinstalado y se encuentra listo para usarse mediante la función usb_enumerated() en verdadera, ejecutando la función apli_with_usb(), y en caso contrario se mantiene el apli_without_usb().

Para el primer caso con USB, se remite nuevamente a saber si recibió algo el dispositivo numerado y si existe algo en ese puerto mediante la función usb_kbhit(). En caso de existir, se obtienen los datos mediante la función que llama a la adquisición de datos antes mencionada usb_get_packet(1, Datain, 64) heredado de la librería de la clase USB.C.

Mediante el buffer Datain[] se logra una conmutación según el evento del hardware con las combinaciones 0x00:grabado, 0x01:revisión del serial del dispositivo, 0x02:grabación de la posición de memoria, la cuales es variable y dinámica respecto a su posición en la Flash, 0x03: graba las cuatro rutinas anteriores juntas, 0x04:apagado de servo por el usuario, la cual es una gran ventaja cuando este se encuentra simulando o ensamblando y requiere hacer una prueba de grados de libertad sin forzar a un servo posicionado, y la 0x05: realiza un movimiento continuo mediante el refrescado

cada 20 milisegundos desde la PC, lo que permite manipular en tiempo real desde el host los 30 servos.

La memoria Flash mediante el tramado permite grabarle hasta un total de 1600 posiciones por 30 servos manejados paralelamente.

El tiempo es un entero de 16 bits y es dividido ocupando 2 bytes, y reensamblado nuevamente para su manipulación.

El proceso de grabado consiste en el envío de un comando que contiene la posición de memoria Flash a donde se colocará basado en un contador de programa que mantiene la última posición grabada y grabado de rutinas con un ciclo de 0 a 3, grabando 6 posiciones simultáneamente con un total de 60 bytes.

Pic C no contiene un comando para realizar PWM en pines no designados, y ningún PIC es de propósito general para servomotores, en otros lenguajes como PICBASIC PRO, se cuenta con instrucciones dedicadas como PULSOOUT, por lo que se construyó una función.

El PWM es calculado mediante su equivalente en ticks de reloj, donde un tick es igual a 0.0833 microsegundos, por lo que el $PWM=150*120$, dando el número total de ticks necesarias para un pulso de 1.5 milisegundos con un incremento en uno en i, donde mediante la función ServoPWM[i] realiza una lectura secuencial a DataIn, multiplicada por 120 y decrementada en uno, logrando un complemento al cálculo del pulso, actualizando el valor de cada temporizador.

El tramado de recepción de datos está basado en la función antes mencionada DataIn que recibe la trama, la graba y compara, ya que en el sistema ha sido implementado un manejador de errores que consiste en verificar cada trama mediante la lectura del paquete enviado desde la Flash.

El modo de trabajo sin USB permite al operador, una vez grabadas las posiciones, que el sistema sea desconectado del USB de la PC y trabaje de modo autónomo para repetir secuencias, por ejemplo, de ensamblado. La función apli_without_usb() es encargada de la lectura EEPROM de las 4 rutinas programadas, si es enciende un led y conmuta a uno de los cuatro estados hexadecimales descritos anteriormente. Las rutinas trabajan mediante la variable Estado, la cual en estado cero, carga los valores de control como número de movimiento a realizar y la posición inicial de memoria de trabajo a la cual puede realizar un bucle infinito; en caso uno, ejecuta el número de movimientos guardados. El otro modo permite hacer una cuenta de movimientos, divide la duración en milisegundos entre veinte y obtiene el número de repeticiones, posteriormente carga estos valores por cada ocho servos por temporizador, y llama a Servo_PWM para ejecutarlas paralelamente las ventanas con su tic alto y baja repetidamente.

Para el caso en donde se conecta a la PC, y el sistema se encuentra haciendo rutinas, este es detenido mediante una función encargada de centrar servos y cargar todas la variables en default.

IV. EL SOFTWARE GUI DE CONTROL HOST

Como se describió en las características de un USB HID, estos se encuentran formados por un dispositivo de hardware y un software interfaz de control y comunicación para IBM PC en Visual C# .NET, como se muestra en la figura 2.

Uno de los valores agregados a esta GUI, además de su autodetección, es el llamado de las APIS con `system.timers.time` para mayor exactitud en la rutina.

Para asegurar la estructura de las tarjetas USB-HID, se asigna en cada una de ellas un número de serie único, el cual es verificado por la función `void verificar_serial()`.

La librería encargada de la identificación a la API de Windows correspondiente es `Mccphidclass`, que hace uso de las funciones de escritura-lectura.

Una característica única de la GUI es que evita la duplicidad de proceso, evitando que se tengan abiertas dos o más interfaces de control, mediante el uso de del MUTEX del sistema operativo mediante `User32()`.

La autodetección USB del dispositivo de interfaz humana viene dada mediante un temporizador dedicado cuando la `MCHPHIDClass.USBHIDIsConnected()` es verdadera mostrando una etiqueta con el texto ON-Line, apareciendo como conectado o desconectado mediante el uso del Hypertreading del sistema operativo.

La aplicación permite la modificación del timing de los servos para una calibración, así se podrá dar una exactitud y control según el modelo. Estos están en el rango de milisegundos mediante la función `servpro_Load()`.

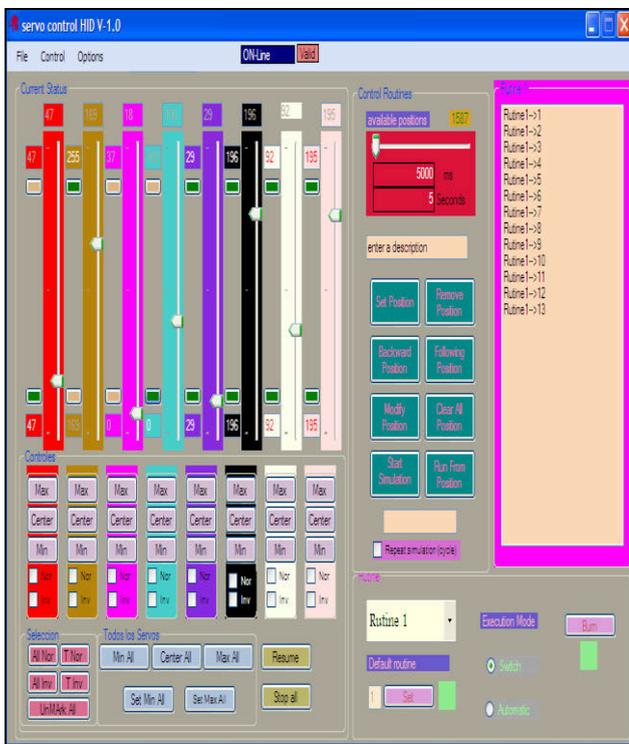


Fig. 2. Software interfaz de control para la tarjeta controladora de servomotores, donde se muestra la GUI del subsistema Host con un canal para 8 servomotores, 13 rutinas, sliders en distintas posiciones, velocidad, posiciones libres e identificación del dispositivo HID con número de serie correcto

Los scrollbars permiten una exactitud mediante la aplicación de la interpolación de Lagrange que permite calcular el equivalente proporcional del valor del pulso PWM a cada servomotor usando la función `trackserv1_Scroll()`.

La función `Duracion_scroll()` permite indicar la duración de cada posición antes de cambiar a la siguiente y la simulación cíclica en tiempo real de toda la rutina o desde una posición en particular.

Para aplicaciones en donde se diseñan dispositivos mecatrónicos con distintos tipos de servomotores como los robots móviles de tipo ápedo o hexápodos, la GUI permite el desplazamiento conjunto y proporcional entre los sliders mediante el anclamiento de uno, además de la modificación del timing, todo esto en tiempo real.

El sistema detecta automáticamente si está conectado a la PC, en caso contrario entra en modo autónomo y ejecuta infinitamente las rutinas y tiempos programados.

V. LA TARJETA DE SERVOCONTROL USB-HID

La tarjeta diseñada se muestra en la figura 3, y está basado en un microcontrolador Microchip modelo PIC 24FJ64GA004 para 30 servomotores, debido a que este cuenta con 5 temporizadores, uno para cada canal de 8 servomotores, sobrándole un temporizador de control, pero puede ser implementado como en los experimentos descritos en el artículo con microcontroladores menos robustos como el 18F4550, al ser portable el firmware.

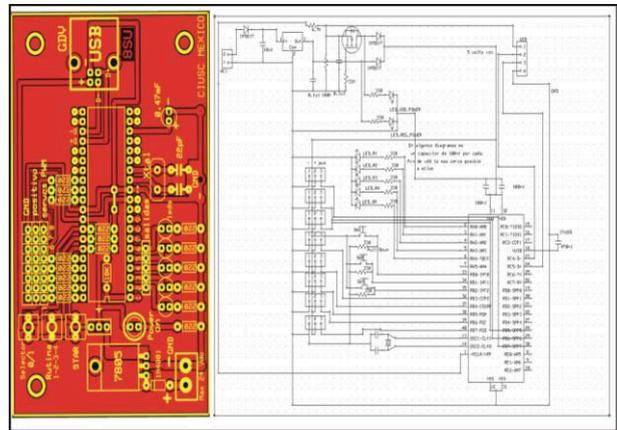


Fig. 3. PCB y diagrama esquemático de la tarjeta USB dispositivo de interfaz humana basada en microcontrolador PIC 18F4550 para 8 canales y PIC 24FJ64GA004 para 30 servomotores

VI. EXPERIMENTACIÓN

Para comprobar el correcto funcionamiento del sistema de servocontrol reprogramable para articulaciones de manipuladores y robots móviles USB-HID, se realizaron tres experimentos.

El primero consistió en la aplicación en un manipulador de 8 grados de libertad construido con servomotores Hitec HS-311 como se describe en la figura 4, almacenando en la memoria interna del PIC1600 posiciones. El sistema se mantuvo trabajando durante 48 horas en modo autónomo sometido a la carga específica

del fabricante de los servomotores. Cuando se requiera una carga superior se deberá diseñar un manipulador con servomotores con la capacidad deseada, ya que el fin de este trabajo es la de proveer una solución electrónico-computacional de control de lazo abierto.

En el tercer caso práctico se propone el uso de una tarjeta convertidora de motoredutores a servomotores para brindar una mayor carga a las aplicaciones del sistema de servocontrol dedicado exclusivamente al envío de señales PWM.



Fig. 4. Brazo robot con gripper utilizando servomotores Hitec HS-311

El segundo experimento consistió en una implementación en la empresa Bio-Raid de México por un miembro del equipo de investigación en el área de medios de cultivo donde se requería del suministro de una sustancia en una oblea permeable con área de 1cm de diámetro, por lo que se construyó una bomba peristáltica y un prototipo que permite ajustar y reprogramar la velocidad de goteo al conectarse a la computadora para dosificar de 0 a 0.5 microlitros, como se describe en la figura 5.



Fig. 5. Prototipo peristáltico para el suministro de una sustancia con 2 grados de libertad y servomotores Vigor, se aprecia el prototipo de bomba peristáltica desensamblada y el multidisco BIO-RAID de México, permite ajustar la velocidad de suministro al conectarse a la computadora para dosificar 0 a 0.5 microlitros

El tercer caso práctico requirió de una tarjeta de control externa debido a que se trabajó con motoredutores de corriente directa en conversión como servo a un máximo

de 4A, esto permite demostrar que el sistema no solo es funcional con servomotores de radio control, sino con sistemas que incluyan la conversión servo-amplificadora mediante la comparación de la señal PWM respecto a una analógica como la transmisión 1:10 de potenciómetro a 2 Kilo Ohms.

Las características de trabajo de un sistema mecatrónico como el descrito anteriormente estarán en función de las tarjetas de comparación analógico-PWM y no de la tarjeta de control descrita.

Para el experimento se diseñó una tarjeta que recibe la señal PWM de la tarjeta USB-HID a 5V, el pulso a recibir debe estar comprendido entre 1 y 2 milisegundos y su frecuencia debe ser de 50 veces por segundo, así mismo el ancho de pulso entregado determina la posición.

En la figura 6 se describe la señal de entrada del sistema servocontrolador reprogramable USB-HID y la salida convertida a voltaje mediante la tarjeta de control externa de conversión motoreductor a servomotor.

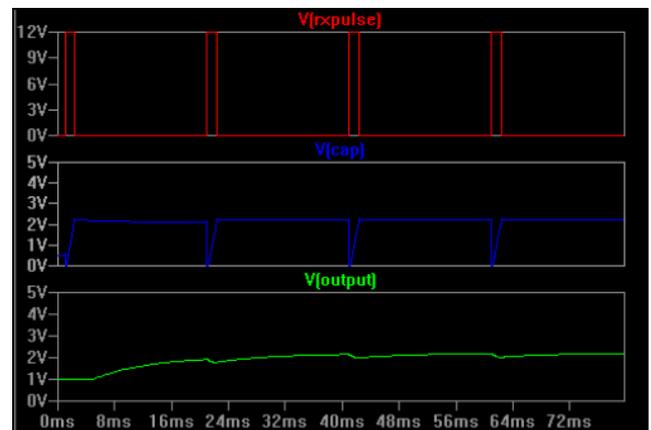


Fig. 6. Señal de entrada del sistema servocontrolador reprogramable USB-HID y salida convertida a voltaje mediante la tarjeta de control externa de conversión motoreductor a servomotor mediante la comparación de la señal PWM respecto a una analógica como la transmisión 1:10 de potenciómetro a 2 Kilo Ohms sin carga

Al utilizar el sistema descrito en tarjetas convertidoras motoreductor a servomotor, se incrementa el espectro de trabajo en ambientes industriales, ya que el servocontrolador puede ser utilizado en cualquier motoreductor de industrial que incorpore una tarjeta de conversión PWM a C.D. como la descrita anteriormente o industriales como el controlador de motores PWM/SCR/4Q/SERVO DC 48RT20BL de la firma Keya e inclusive variadores vectoriales como el V1000 de Yasaka que reciben una señal PWM como la del presente trabajo y permite controlar motores trifásicos de C.A.

Para cada diseño que utilice el sistema servocontrolador, se deberá tomar en cuenta que la tarjeta de conversión como las antes descritas cumpla con las necesidades requeridas como caballos de fuerza, torque o velocidad, así mismo para sistemas de lazo cerrado se recomienda integrar subsistemas de ajuste como PID, ya que el sistema electrónico-computacional descrito en este

trabajo se centra exclusivamente en el envío de una señal PWM y el almacenamiento de rutinas mediante la tecnología USB-HID.

Para la aplicación del experimento tres se desarrolló el sistema descrito en la figura 7.

El sistema comienza con la tarjeta de servocontrol reprogramable para articulaciones de manipuladores y robots móviles mediante tecnología USB-HID y una computadora IBM-PC con Windows 7, el PWM proveniente de la tarjeta es enviado hacia la tarjeta de conversión motoreductor-servomotor que cuenta con un segundo microcontrolador encargado de convertir el valor proporcional de la escala de un giro del motoreductor respecto a 10 vueltas del potenciómetro de 2 Kilo Ohms y compararlo con el PWM recibido mediante un diseño paralelo de control PID, permitiendo así convertir motoreductores industriales en servomotores.

Se utilizó una fuente externa para alimentar todos los motoreductores de un manipulador industrial Kuka 363.

Una vez programada una secuencia de 25 movimientos simulando un ensamblaje, el sistema fue removido de la computadora convirtiéndose en modo autónomo.

Para tener un control de errores de seguimiento, se puede hacer un sistema de lazo cerrado en la tarjeta de conversión motoreductor a servomotor que incorpore la lectura de posiciones y realice un ajuste PID. Algunas tarjetas de conversión motoreductor a servomotor son capaces de realizar esto, pero todas requieren de una señal PWM como la que se propone en este trabajo.

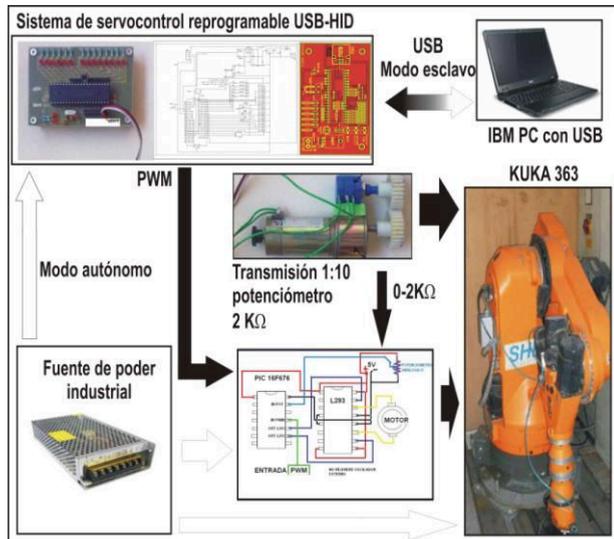


Fig. 7. Diagrama de bloques de la implementación del sistema de Servocontrol para articulaciones de manipuladores y robots móviles USB-HID, donde se acopló una transmisión del movimiento de los motoreductores de corriente directa del manipulador industrial KUKA 363 mediante un potenciómetro de 2 Kilo Ohms a una tarjeta convertidora a servomotores encargada de hacer el ajuste de los valores leídos de la posición del motoreductor respecto al PWM recibido por la tarjeta en modo autónoma

VII. CONCLUSIONES

El sistema permite controlar 30 servomotores y ser implementado en microcontroladores diversos según la necesidad, así mismo permite el grabado de rutinas, cuenta con autodetección, anulación de drivers, portabilidad, trabajo con motoreductores C.A. o C.D. por conversión a servomotor y una potencia máxima de suministro externo USB de 1 amperio.

La GUI utiliza una interpolación basada en Lagrange para los sliders y permite su aplicación en manipuladores industriales de distinta configuración o robots móviles en su modo de trabajo autónomo para aplicaciones en sistemas embebidos.

Actualmente se trabaja en el desarrollo de una tarjeta USB HID-Mass storage compuesto, la cual incluya una GUI embebida el Hardware, permitiendo ser detectado como USB-HID y memoria de datos extraíble simultáneamente.

Así mismo el equipo programa una GUI en Java y se encuentra migrando al PIC 24fj256gb110, el cual cuenta con 100 pines y 5 temporizadores internos de 16 bits que permitirán el control de 40 servomotores paralelamente.

VIII. AGRADECIMIENTOS

El equipo del centro de investigación CIUSC agradece a "Laboratorio BIO-RAD S.A" por su disponibilidad a realizar el prototipo y experimentación del mismo en su planta. .

REFERENCIAS

- Phidgetadvanced servo 8-motor (2012) <http://www.phidgets.com>
- Pic24fj64ga004 family data sheet, p.3
- Jan axelson, usb complete, oxford, 2009.
- Usb org (2012) http://www.usb.org/usb_overview
- Hi tec hs-311 data sheet, p. 1
- Jenan keya electronic pwm/scr/4q/servo 48rt20bl data sheet, p. 4
- Yasaka V1000 data sheet, p. 2