

Teclado con protocolo de comunicación universal aplicado a máquina de CNC

Keyboard with Universal Communication Protocol Applied to CNC Machine

Mejía-Ugalde Mario

*Facultad de Ingeniería
Universidad Autónoma de Querétaro
Correo: mmu92@hotmail.com*

Información del artículo: recibido: noviembre de 2012, reevaluado: febrero de 2013, aceptado: abril de 2013

Resumen

Este artículo, describe el uso de un protocolo de comunicación universal para teclado industrial basado en un microcontrolador aplicado a una máquina de *control numérico por computadora* (CNC). La diferencia principal entre los fabricantes de teclados es que cada fabricante tiene su propia programación de código fuente, produciendo un protocolo de comunicación diferente, generando una interpretación inadecuada de las funciones establecidas. Lo anterior crea que los teclados comerciales industriales sean costosos e incompatibles en su conexión con diferentes máquinas. En el presente trabajo el protocolo permite conectar el teclado universal diseñado y un teclado normal de la PC al mismo tiempo, este es compatible con todas las computadoras a través de la comunicación USB, AT o PS/2, para emplearse en máquinas de CNC, con extensión a otras máquinas como robots, soplado, máquinas de inyección, entre otras. Entre las ventajas de este diseño se encuentran: la fácil reprogramación, disminución de costos, manipulación de diversas funciones de máquina y facilidad de expandir las señales de entrada y salida. Los resultados obtenidos de las pruebas de funcionamiento fueron satisfactorios, debido a que cada tecla tiene la facilidad de programarse y reprogramarse de diferentes maneras, generando distintas funciones de códigos, dependiendo de la aplicación donde se requiera utilizar.

Descriptores:

- microcontrolador
- teclado
- USB
- AT
- PS/2
- protocolo

Abstract

This article describes the use of a universal communication protocol for industrial keyboard based microcontroller applied to computer numerically controlled (CNC) machine. The main difference among the keyboard manufacturers is that each manufacturer has its own programming of source code, producing a different communication protocol, generating an improper interpretation of the function established. The above results in commercial industrial keyboards which are expensive and incompatible in their connection with different machines. In the present work the protocol allows to connect the designed universal keyboard and the standard keyboard of the PC at the same time, it is compatible with all the computers through the communications USB, AT or PS/2, to use in CNC machines, with extension to other machines such as robots, blowing, injection molding machines and others. The advantages of this design include its easy reprogramming, decreased costs, manipulation of various machine functions and easy expansion of entry and exit signals. The results obtained of performance tests were satisfactory, because each key has the programmed and reprogrammed facility in different ways, generating codes for different functions, depending on the application where it is required to be used.

Keywords:

- microcontroller
- keyboard
- USB
- AT
- PS/2
- protocol

Introducción

El teclado es uno de los elementos más importantes dentro de la automatización de maquinaria, por medio de este, el operador introduce las órdenes de trabajo que la máquina debe realizar o el usuario puede editar el programa general que controla la máquina. Sin embargo, en el mercado existe una gran variedad de fabricantes de teclados industriales, y cada uno tiene una forma diferente de comunicar sus teclados a las máquinas, en algunos casos se tienen que instalar controladores especiales al ordenador para que pueda admitirlo o reconocerlo al conectarlo y muchas veces estos teclados no son compatibles entre sí, es decir, en caso de que un teclado ya instalado falle, no se podrá reemplazar por otro de diferente fabricante, ya que no suelen tener la misma configuración del protocolo de comunicación (García, 2009).

De acuerdo con los avances tecnológicos, la industria de procesos requiere sistemas con mayor cantidad de elementos o mecanismos que puedan controlarse de una forma más fácil, confiable y eficaz; por esta razón, se pensó en dispositivos más autónomos como es el uso de microcontroladores y la PC para el control. En el mercado encontramos diferentes fabricantes de teclados como son: Rafi, Schurter, Click Touch, Storm, Beltronic, Indukey, Siemens, Devlink, Ikey, Hitra, Hid y Cti; sin embargo, aunque existen algunas innovaciones y diseños nuevos de teclados en el mercado, es difícil

encontrar el específico para un proceso u automatización.

Estado del arte

La evolución de los teclados a lo largo de la historia ha cambiado notablemente en las áreas didácticas e industriales (Li, 2006).

Dentro de los trabajos realizados con respecto a los teclados, encontramos algunos artículos como el caso de Varela (2002), quien realiza un teclado matricial de 70 casillas, lo que significa que superó el tamaño estándar de 64 casillas (102 teclas) con la intención de obtener teclas adicionales con funciones específicas; en su trabajo el microcontrolador PIC16F84A fue insuficiente por el número reducido de pines (Manual, 2001) según el manual del microcontrolador y decidió utilizar el microcontrolador PIC16F873A. Domínguez *et al.* (1991) determinaron que la forma más común para reducir el número de líneas necesarias para la conexión en un teclado es una matriz. En cuanto a protocolos de comunicación encontramos el artículo de Suárez *et al.* (2005) quienes desarrollaron una interfaz de comunicación con la computadora a través del puerto USB con la alternativa de hacerlo a través del puerto serie RS232 basados en el Microprocesador MSP430F155.

Por otra parte Li (2006), presentó la realización de un teclado móvil inalámbrico aplicado a la industria, muestra que se puede llevar a cabo un dispositivo mó-

Nombre del teclado	Aplicación	Fabricante	Precio + Conexiones
FlexOhm flexible keyboards (teclado flexible a la resistencia)	Máquinas electrostáticas y generación de energía	Click Touch	\$ 4,700
Vandal-proof keypads (teclado de banda)	Máquinas transportadoras	Schurter	\$ 5,000
Electroluminescent lamps keyboards (teclado equipado con lámparas electro-luminiscentes)	Estudios de revelado	Devlink	\$ 6,200
Touch Screens keyboards (teclado táctil en pantalla)	Fabricación de productos tóxicos	Indukey	\$ 6,500
Industrial plastic keyboards (teclado industrial para plásticos)	Máquinas de inyección y soplado	Click Touch	\$ 4,400
Force sensors keyboards (CFSK) (teclado de sensor de fuerza)	Siderurgia y fundición	Siemens	\$ 5,200
Contact layers keyboards (teclado de contacto en capa)	Fabricación de cristales	Hid	\$ 3,800

Tabla 1. Características de los teclados comerciales

vil usando solamente una mano o un indicador como entrada de datos, pero la única desventaja es que utilizó un emisor y un receptor, por lo que su protocolo de comunicación puede funcionar solo para algunas máquinas. Navabi (2004), presenta comunicación serial de un núcleo de código (CORE) del teclado sobre el chip de Altera MAX 700 series EPM7128S PLD en VHDL (*Very High Speed Integrated Circuits Hardware Description Language*, lenguaje de descripción de hardware para circuitos de alta velocidad). Dogan (2006), mediante la integración de los PLD's (*Programmable Logic Device*, dispositivo lógico programable) en micro, presentó la realización de un decodificador para el código ASCII realizado en display alfanumérico y elaborado para visualizar los caracteres o números ingresados por el teclado, que opera como dispositivo de entrada para un CORE (decodificador de ASCII a display alfanumérico); Peng y En-Ping (2011), desarrollaron un teclado reuniendo varias letras o código de ellas en una sola tecla con el método llamado CyberSwarm.

La tabla 1 muestra una lista de fabricantes que ofrecen diferentes tipos de teclados para aplicaciones específicas, ocasionando que algunas industrias cambien su sistema de control para adecuarse a ese teclado y conexiones; generando con esto un incremento en el costo. A pesar de la diversidad de los teclados, la desventaja que se tiene radica en el requerimiento de accesorios adicionales y la interconectividad con todos los equipos. Por esta razón se desarrolla la siguiente investigación.

Propuesta general

Lo novedoso de este proyecto es que ofrece un teclado con ciertas características o ventajas respecto a los

comerciales en dos partes: interfaz con el usuario y funciones adicionales. La primera, permite la conexión directa mediante el protocolo de comunicación USB, AT o PS/2 en todos los ordenadores, sin la necesidad de instalar controladores y software especiales. La segunda, se adapta al tipo de proceso que se requiere automatizar usando funciones adicionales; esto quiere decir que si cambia o aumenta el número de herramientas o componentes en la máquina solo se necesita una sencilla reprogramación en el protocolo, logrando con esto su fácil control.

Teniendo un teclado universal se podrá aplicar en cualquier máquina automatizada, lo único que se necesita cambiar serán las funciones específicas de algunos caracteres para las diferentes actividades de cada máquina; y para su fabricación se utilizarán circuitos integrados programables (PIC) que no son costosos (entre \$80 y \$100) de fácil programación.

Dentro de las funciones adicionales de este proyecto se pueden programar y reprogramar las teclas para realizar movimientos especiales y generar caracteres aplicados a diferentes actividades en cada máquina. Este nuevo diseño cuenta con funciones especiales que son útiles en los procesos de maquinado para evitar que cada vez que se reemplaza un teclado comercial por otro, se tenga que cambiar la configuración de los códigos que la máquina tiene programada.

Desarrollo

La figura 1 muestra el diagrama general de protocolo de comunicación universal para teclado industrial basado en microcontrolador, el microcontrolador permite la comunicación entre el teclado de la PC y el teclado industrial hacia la computadora mediante el puerto

USB o PS/2 (Chung y Chih, 2007), el conector PS/2 se muestra en la figura 2 con los conectores típicos de teclados: el conector de 5-pines (AT) y el mini-dinar de 6-pines (PS/2). Ambos conectores transmiten en forma serial los datos y son por completo (eléctricamente) similares, la única diferencia práctica entre los dos es el arreglo de pernos y el tamaño; esto significa que los dos tipos de conectores se pueden cambiar fácilmente con los adaptadores hard-wired simples.

En la figura 2 se presenta el protocolo típico para transmisión de datos en teclados, donde la frecuencia del reloj es de 10 a 16.7 KHz. La transmisión de un bit se realiza en la transición positiva del reloj y el dato es enviado dentro de 11 transiciones del reloj. La información se envía desde el teclado de forma serial usando 11 bits sobre la línea de datos bi-direccional, la línea de datos permanece inactiva (nivel bajo) cuando ni el teclado ni el computador envían datos. El tiempo de transición de un pulso positivo a negativo de los datos debe ser por lo menos $5\mu\text{s}$ (García, 2009), el tiempo de una transición de datos que baja, debe ser por lo menos $5\mu\text{s}$ y no mayor que $25\mu\text{s}$. El tiempo de

datos de transmisión puede inhibir la comunicación en cualquier momento usando la línea hacia abajo del reloj, por lo menos $100\mu\text{s}$. Si una transmisión se inhibe antes de 11 pulsos del reloj, el dispositivo debe abortar la transmisión actual y prepararse para retransmitir el dato actual. Por ejemplo, si se interrumpe un teclado mientras envía el segundo paquete de 8 de un código de dos bits, necesitará retransmitir ambos octetos de ese código.

El protocolo de comunicación implementado sigue las reglas que permiten llevar a cabo la comunicación según Chung y Chih, (2007) para este dispositivo mediante el puerto físico USB, AT o PS/2 mostrado en la figura 2.

El diagrama a bloques de la figura 3 muestra el protocolo de comunicación que se realizó modificando los protocolos de comunicación existentes para teclado (estructura matricial), pero manteniendo las mismas conexiones y conectores mostrados en la figura 1. El protocolo se genera en las siguientes etapas: *escaneo de teclas y convertidores serie-paralelo* que se logra por medio de una señal de activación mandada desde el microcon-

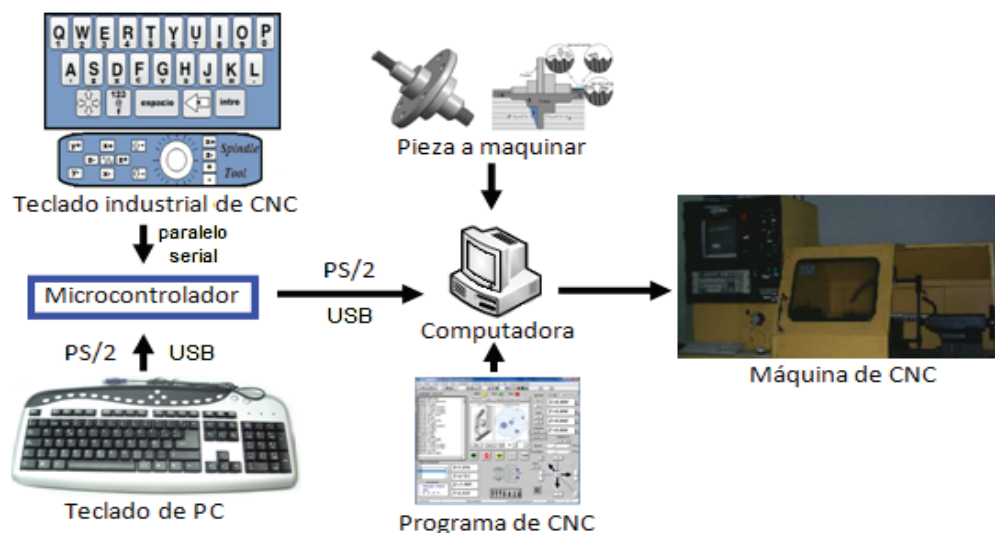


Figura 1. Diagrama general de protocolo de comunicación universal

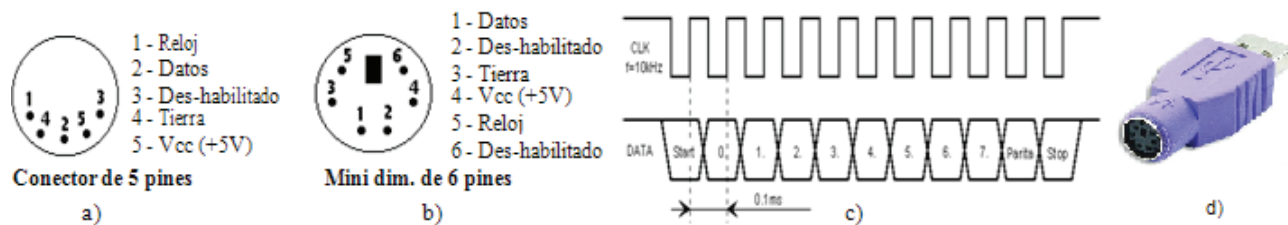


Figura 2. Conectores a) AT, b) PS/2, c) transmisión de datos y d) adaptador PS/2 a USB

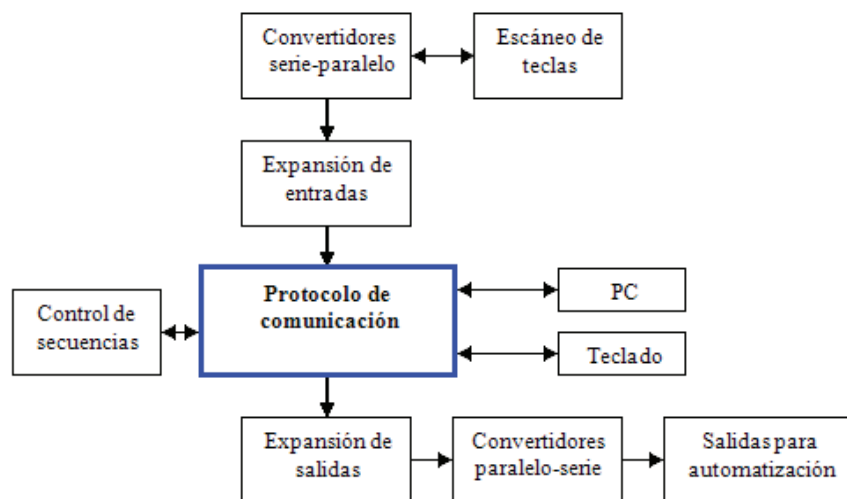


Figura 3. Diagrama a bloques del protocolo de comunicación

trolador cada 20 ms activando todos los circuitos integrados 74LS166 (convertidores serie-paralelo en donde las entradas son de 8 bits y el corrimiento de 1 bit), adquiriendo cada circuito las señales de 8 bits provenientes de las teclas que posteriormente con una señal de reactivación pasan los datos uno a uno en forma serial. El bloque de *expansión de entradas* realiza un corrimiento de forma serial sincrónico hacia el microcontrolador y hacia el protocolo de comunicación.

El módulo *protocolo de comunicación* permite la comunicación bi-direccional del teclado con la computadora. Este a su vez, puede estar conectado al ordenador (PC) que al mismo tiempo está conectado a un teclado normal. En el bloque *control de secuencias* se encuentra el microcontrolador, el cual realiza ciertas funciones de secuencia y control. *Expansión de salida*, este bloque se realizó por medio de los circuitos integrados 74LS199 (convertidores paralelo-serial, el corrimiento de entradas es de 1 bit y las salidas de 8 bits) en donde las salida puede ser desde 8 hasta n salidas (García, 2009). *Salidas para automatización* (Ramírez et al., 2011), es la última etapa digital para la automatización o para fines deseados, aquí es donde se requiere de un opto-acoplamiento si la aplicación necesita que la información sea transmitida entre dos circuitos eléctricamente aislados uno de otro. Este aislamiento se puede conseguir mediante relevadores, transformadores de aislamiento y receptores de línea. No obstante, otro dispositivo que se utiliza de manera igualmente efectiva para resolver estos problemas (Kangbin, 2008; Ramírez, 2011) es el opto-acoplador, su empleo es muy importante en aplicaciones en las que el aislamiento de ruido, alta tensión y el tamaño son características determinantes.

En la figura 4 se presenta de manera más detallada la funcionalidad del teclado implementado que consta

de tres módulos de registros serie-paralelo, cada uno de 8 entradas digitales, la conexión directa bidireccional del ordenador al microcontrolador y de éste al teclado normal; un módulo para expandir entradas, otro para las salidas, y el último módulo usado industrialmente para la automatización y control.

La figura 5 muestra las diferentes estructuras digitales programadas en el microcontrolador como son: *un contador módulo 3*, encargado de activar las líneas de selección de los multiplexores, *dos multiplexores de 7 a 1* usados para dirigir los bits uno a uno hacia la base de datos, para su posterior almacenamiento, *una compuerta xor* que es una segunda selección de datos antes de llegar a la base de las teclas y códigos, *una base de datos y código de teclas* donde se unen los 8 bits en 20 ms y se da el formato y código de la tecla oprimida, *un búfer circular* (Kangbin, 2008) donde se realiza la comparación de los paquetes de 8 bits en tres columnas, como lo muestra la tabla 2; si uno de estos paquetes cambia con relación al que se encuentra en el búfer, este entrará al búfer reemplazando el anterior y se sincroniza el microcontrolador mediante *máquinas de estado*.

Una compuerta XOR es un arreglo de tres compuertas básicas AND, OR y NOT que produce una salida '1' lógico si sus entradas son diferentes, se emplea un bloque digital XOR asíncrono para verificar qué tecla(s) se ha(n) activado, para llevar este proceso se toman los valores que contienen los estados actuales y anteriores de las teclas, proporcionados por los registros Q1 y Q2, al aplicar la operación XOR se genera una bandera de activación de la *tecla presionada* (TPS), cabe mencionar que los datos para cada tecla son transmitidos por la línea con etiquetas Q1 y M1 generando la *tecla de los datos internos* (TDI).

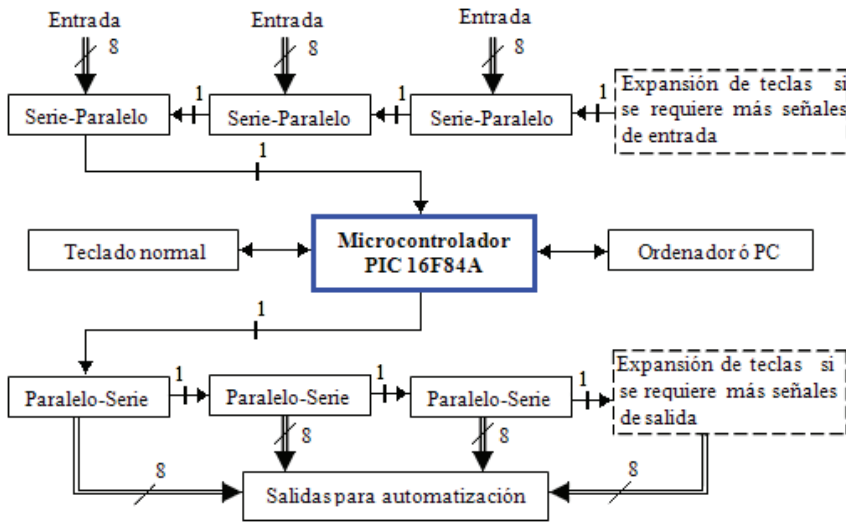


Figura 4. Diagrama funcional del protocolo de comunicación

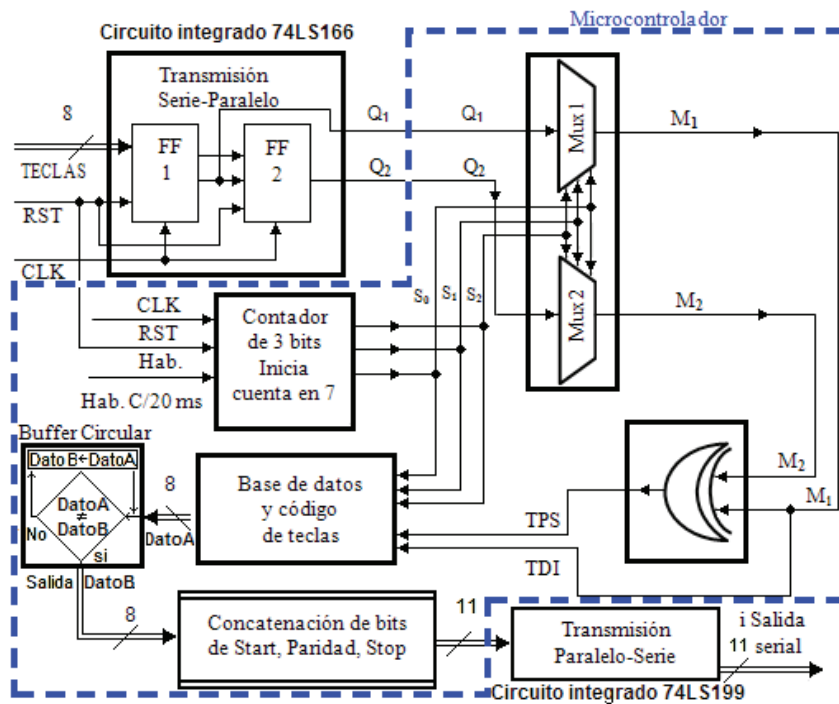


Figura 5. Estructuras digitales programadas en el microcontrolador

Dato A	Dato B	Condición	Salida
Dato A[0]=dato 0	Dato B[0]=dato 0	Si, dato A =dato B entonces	No
Dato A[1]=dato 9	Dato B[1]=dato 7	Si, dato A ≠ dato B entonces	Dato B
Dato A[2]=dato 2	Dato B[2]=dato 2	Si, dato A = dato B entonces	No
Dato A[3]=dato 3	Dato B[3]=dato 3	Si, dato A = dato B entonces	No
Dato A[4]=dato 9	Dato B[4]=dato 8	Si, dato A ≠ dato B entonces	Dato B
Dato A[5]=dato 4	Dato B[5]=dato 4	Si, dato A = dato B entonces	No
Dato A[6]=dato 5	Dato B[6]=dato 5	Si, dato A = dato B entonces	No
Dato A[7]=dato 6	Dato B[7]=dato 6	Si, dato A = dato B entonces	No
Dato A[8]=dato 8	Dato B[8]=dato 8	Si, dato A = dato B entonces	No

Tabla 2. Búfer circular

La tabla 3 muestra el programa principal que comienza escalando el reloj para estar sincronizados los datos al realizar la transmisión o recepción de datos con una velocidad de 12 a 20khz, el arranque o reinicio de la máquina, la presencia de datos con el escáner de teclas que identifica qué tecla presiona y genera el método de envío, recepción o almacenamiento de los datos detectados.

La tabla 4 muestra el programa de escaneo de teclas la cual identifica qué tipo de proceso se está llevando a cabo, recepción, transmisión o almacenamiento y gene-

rar las banderas necesarias para saber qué tipo de función se realiza, si la PC baja el reloj (cero lógico) la PC transmite y el teclado recibe, si el teclado baja el reloj (cero lógico) entonces el teclado trasmite y la PC recibe los datos.

La tabla 5 muestra la identificación y concatenación de la tecla que está presionada, la concatenación es el almacenamiento y el orden de los bits.

El control de tecla se muestra en la tabla 6, y esta permite saber los tiempos que una tecla se está presionando y a su vez saber si se está enviando o recibiendo datos.

```

Programa principal (recep , star_stop, ban_r, aux, cont, clk2, estado_proximo, rst)
reducción (clk, b, rst) %genera la velocidad de trasmisión de datos
If presencia (ack, recep, sal2, clk0, tecla_auxiliar, star_stop, ban_r) then
  escaneo_de_teclas (in_micro_clk, cuenta, in_micro_dato, start )
  if ACK_contestacion (clk2, ack_recepcion, ack_cuenta, star_ack) then
    swich (ack_recepcion)
    caso 0 is %Transición
    Control_de_transmision (Inicio_de_trasmision, Fin_de_trasmision)
    Envio de datos (clk_Pc_bi, dato_Pc_bi, dato0, dato1, dato2, dato3... dato9)
    Caso 1 is %Recepción
    condiciones_d_recepcion (tecla, dato_0, dato_1 , dato_2, dato_3, dato_4,
    dato_5,
    Envio de datos (clk_Pc_bi, dato_Pc_bi, dato0, dato1, dato2, dato3... dato9)
    Caso 2 is %Envío
    condiciones_d_envio (tecla_auxiliar)
    Envio de datos (clk_Pc_bi, dato_Pc_bi, dato0, dato1, dato2, dato3... dato9)
  End swich;
End if;
End if;

```

Tabla 3. Programa principal

```
escaneo_de_teclas (in_micro_clk, cuenta, in_micro_dato, start )
begin
```

Tabla 4. Identificación de la tecla presionada

```

  if (falling_edge(in_micro_clk)) then % Inicializa
    if cuenta=="00000" and in_micro_dato=='0' then %Espera bit start
      start='1'; Inicio_de_trasmision = '1';%inicia todo%Habilitador1 del 74245
      star_stop = '0';
    end if; if start=='1' then
      Inicio_de_trasmision= '1'; %Habilitador1 del 74245
      registro = in_micro_dato & registro(8 downto 1); %guarda datos(letra) en
      registro
      if cuenta=="01000" then %límite de datos 11
        star_stop='1'; dato_out=registro(8 downto 1); %manda el dato a la
        salida
      end if;
    end if;
  end if;
end;
```

```

%concatenación de la tecla escaneada (formato)
%D10=1 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0=0
% stop paridad D7 D6 D5 D4 D3 D2 D1 D0 star
concat_comercial(0) = '0'; %bit de star
concat_comercial(8 downto 1) = dato_out(7 downto 0); %bits de datos
concat_comercial(9) = not(dato_out(0) xor dato_out(1) xor dato_out(2) xor dato_out(3) xor dato_out(4) xor
dato_out(5) xor
dato_out(6) xor dato_out(7)); %bit de paridad
concat_comercial(10)= '1'; %bit de stop
tecla_presionada_dato(tecla_auxiliar3, ack, recep, si_ban, sal, ban_r, star_stop, concat_comercial, teclaF,
tecla1)
begin %identificación de tecla (teclado fabricado) % Bandera de recepción,
  if tecla1=='1' then sal = "10000101100" ; % número 1
  aux = '1';Led="11111001"; end if;
  if teclaF=='1' then sal = "11001010110" ; aux= '1'; % tecla F
  if (si_ban == '1') then aux= '1';else % espera para imprimir el primer carácter
  aux= '0'; % resetea el búfer de datos a enviar
  end if; end if;
end;
```

Tabla 5. Detección de tecla presionada

```
presencia (ack, recep, sal2, clk0, tecla_auxiliar, star_stop, ban_r)
```

Tabla 6. Control de la tecla

```

begin %tecla presionada fabricado
  if ((sal2 /= "1111111111") and (clk0=='1')) then
    tecla_auxiliar = sal2; if (star_stop=='1') then rst = '0'; end if;%teclado comercial
  if (ban_r=='1') then rst = '0'; end if; %teclado fabricado
  if (recep=='1') then rst = '0'; end if; %teclado fabricado
  if (ack=='1') then rst = '0'; end if;
  else tecla_auxiliar = tecla_auxiliar; rst = '1'; end if;%retiene el dato a enviar
end;
```

La tabla 7 permite enviar los datos de manera serial por el puerto PS/2 en tiempos iguales cuando la bandera tecla_auxiliar esta activada.

La tabla 8 envía los datos, pero aquí se reconoce qué dato se envía, por medio de un contador de 11 bits y banderas activadas.

La tabla 9 recibe el dato que transmite la computadora al teclado, si la computadora envía un error o está ocupada haciendo otro proceso, el teclado industrial le pide al búfer circular que reenvíe los datos.

Cuando una computadora inicia el proceso de arranque o de reinicio, esta envía datos de reconocimiento a todos los periféricos (teclado, mouse, escáner, pantalla, entre otros); estos periféricos deben recibir los datos almacenarlos e identificarlos para después con-

testarle a la computadora (PC). En la tabla 10 se muestra este proceso.

La tabla 11 identifica qué dispositivo está transmitiendo o recibiendo datos mediante un detector o bandera.

La transmisión de datos de la computadora al teclado y viceversa se realiza a una velocidad de reloj de 1 a 12 khz, si no se envían estos datos a esa velocidad la computadora no podrá almacenarlos, ocasionando un error o que no reconozca el dispositivo (teclado); para esto existe un código fuente mostrado en la tabla 12 para bajar o incrementar la frecuencia de reloj dependiendo del cristal de cuarzo o la velocidad de envío del microcontrolador.

```
condiciones_d_envio (tecla_auxiliar)
begin    %cambio de variable par enviar
    dato0= tecla_auxiliar(0); dato1= tecla_auxiliar(1); ...dato10= tecla_auxiliar(10);
end;
```

Tabla 7. Control de envío

```
%contador de 11 bits iguales (22 diferentes) para enviar 11 datos
Envío de datos (clk_Pc_bi, dato_Pc_bi, dato0, dato1... , dato10)
begin    swich (ban='1') %existe dato para enviarse
    case 0 is
        clk_Pc_bi='Z';    %alta impedancia dato_Pc_bi='Z'; %bit de star
        clk0='0'; %habilitación Fin_de_transmicion='1'; Recepcion='0';
        case 0 is %envía dado0, dato1...
end;
```

Tabla 8. Envío de datos

Tabla 9. Recepción de datos

```
condiciones_d_recepcion (tecla, dato_0, dato_1, dato_2, dato_3, dato_4, dato_5, dato_6, dato_7, dato_8, dato_9,
dato_10)
begin    %cambio de variable par enviar %ACK Stop P D7 D6 D5 D4 D3 D2 D1 D0
    tecla(0)=dato_0; tecla(1)=dato_1; tecla(2)=dato_2; tecla(3)=dato_3; tecla(4)=dato_4; tecla(5)= dato_5;
    tecla(6)=dato_6; tecla(7)=dato_7; tecla(8)=dato_8; tecla(9)=dato_9; tecla(10)=dato_10;
    if (tecla=="01111101101") then tecla_aux="11111110100"; end if;%( ACK)
end;
```

```
ACK_contestacion (clk2, ack_recepcion, ack_cuenta, star_ack)
begin
    if (rising_edge(clk2)) then;%ACK bandera contestación
        if ack_cuenta=="00000" and ack_recepcion='1' then star_ack= '1' end if;
        if (star_ack='1') then if ack_cuenta=="00010" then ack= '1'; end if;
        if ack_cuenta=="00011" then ack_cuenta="00000";star_ack = '0'; ack = '0';

        else ack_cuenta= ack_cuenta + 1; end if;
    end if;
end if;
end;
```

Tabla 10. Contesta los datos a la PC

Tabla 11. Determina quién está transmitiendo o recibiendo

```

Control_de_trasmision (Inicio_de_trasmision, Fin_de_trasmision)
begin
  if (Inicio_de_trasmision=='1' or Fin_de_trasmision=='1') then out_ban_detector_clk='1';
  else out_ban_detector_clk = '0'; end if;
end;

```

Tabla 12. Baja la frecuencia de 50 Mhz a 12khz

```

reduccion (clk, b, rst)
Begin
  if (rst=='0') then   clk2='1';
  elsif (rising_edge(clk)) then  b=b+1;
    if b==1 then clk2=='0'; end if;
    if b==2000 then clk2=='1'; end if;
    if b==4000 then b=0; end if;
  end if;
end;

```

Experimentación

La implementación física del prototipo de comunicación se muestra en la figura 6 y la conexión directa entre los dos teclados al mismo tiempo. Algunas pruebas se realizan con la finalidad de identificar el funcionamiento del nuevo diseño, entre las pruebas experimentales se encuentran el arranque y reinicio del ordenador para su detección y reconocimiento de los dos teclados. Por razones de seguridad hacia el proceso la prioridad total la tiene el teclado industrial; entre las funciones del teclado industrial están la de cambiar parámetros como puntos de ajuste, rangos de operación en tiempo, límites de variables, permisivos, protecciones y otras.

Las funciones principales del teclado PC son la puesta en marcha y el registro al programa (código G y M), y algunas pruebas de tecleo de botones al mismo tiempo y de forma consecutiva, como resultado se tiene un teclado universal industrial nuevo en perfectas condiciones y con pruebas satisfactorias.

Discusiones y análisis

En la figura 7 se muestran las señales obtenidas del protocolo de comunicación del teclado normal y el implementado (industrial). Las dos primeras líneas de la parte superior muestran los datos del teclado normal y las dos líneas de la parte inferior muestran los datos del teclado implementado, ambos envían el mismo dato al mismo tiempo, donde la prioridad es el teclado industrial por el botón de stop de máquina si es que hubiera un error en la máquina o humano, finalmente el teclado normal transmite su señal y esta es guardada en un búfer circular de datos y para terminar se transmite.

En la figura 7a se envía el dato 00010001011 que corresponde a la palabra "o" (código: cuando se presiona una tecla es 0x44 y cuando se libera la tecla es 0xF0, 0x44), donde la señal se envía desde el teclado normal pasando primero por el teclado implementado hacia el ordenador.

En la figura 7a y b se transmite el dato 00010101011 que corresponde al número "0", siguiendo la caracterís-

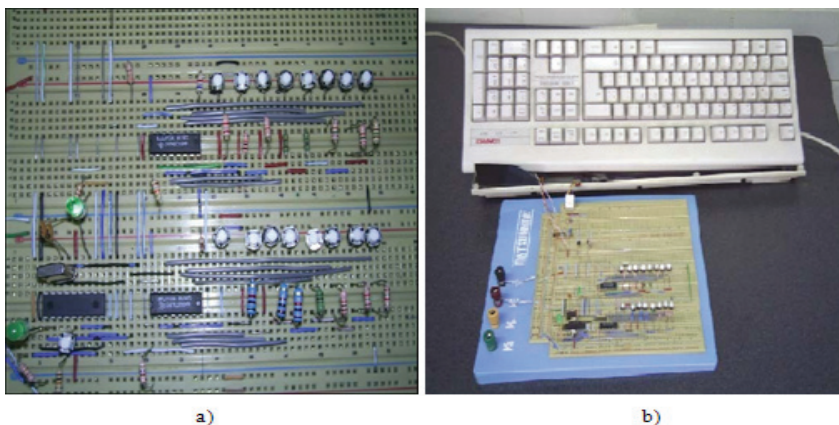


Figura 6. Implementación física del prototipo de comunicación, a) prototipo de implementación de 16 teclas, b) conexión directa entre los dos teclados hacia el ordenador

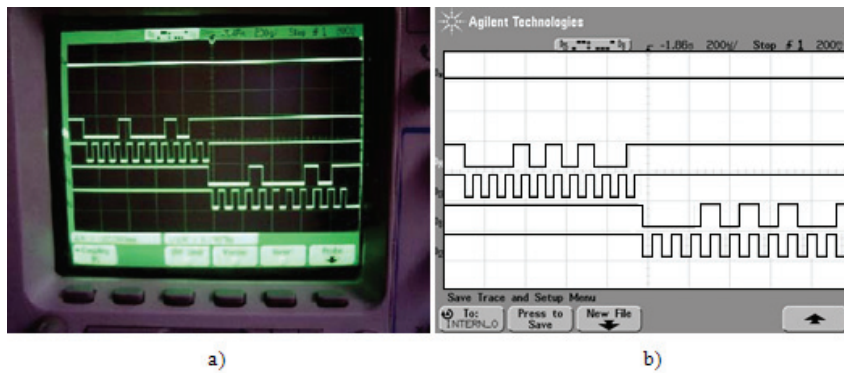


Figura 7. Señales seriales del protocolo de comunicación (teclado normal y el implementado), a) enviando la letra “o” al ordenador, b) se envía el número “0”al ordenador

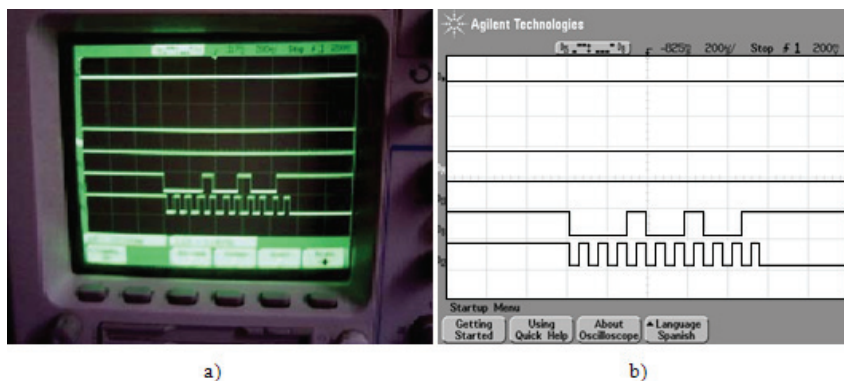


Figura 8. Señales seriales del protocolo de comunicación (sólo el teclado implementado), a) y b), muestran el mismo dato enviado al ordenador (letra “K”)

tica de transmisión de los 11 bits de comunicación; el primero es el bit de inicio, los 8 siguientes son los datos, el penúltimo es el de paridad (indica si se están enviando o recibiendo datos), y el último es el bit del paro (stop).

En la figura 8 se presenta la señal de datos enviada al ordenador desde el teclado implementado (00010010011 correspondiente a la palabra “K”, 0x42), se puede observar que las señales transmitidas desde el microcontrolador van directamente al ordenador de la computadora sin pasar por el teclado de la PC, que tiene un protocolo de comunicación matricial, las señales enviadas son las últimas dos líneas.

Las figuras 7 y 8 muestran que la comunicación es satisfactoria e idéntica tanto del protocolo de comunicación implementado como del comercial, logrando la obtención de un teclado multifuncional y con teclas para funciones específicas.

Conclusiones

Se diseñó e implementó un protocolo de bajo costo para el control de máquinas automatizadas, logrando todos los objetivos propuestos desde el prototipo de un teclado estándar con protocolo y tecnología propia (eléctrica, control y diseño físico) mostrado en la figura

6, aplicado a máquinas de control numérico y a la automatización de maquinaria, permitiendo la compatibilidad en cualquier computadora, con la finalidad de que los códigos programados en una máquina de control numérico no se vean afectados si se reemplaza un teclado por otro cuando falle.

La arquitectura diseñada queda abierta a nuevas aplicaciones e innovaciones, sin necesidad de cambiar el hardware.

Agradecimientos

Este artículo contó con el apoyo financiero de la Universidad Autónoma de Querétaro (UAQ) y por el Consejo Nacional de Ciencia y Tecnología (CONACYT) (número de concesión 209333).

Referencias

- Chung-Chen S. y Chih H. KVM Switch Cable for PS/2 and USB Signaling 2007: Patent. US 7, 281, 067, B2.
- Varela Á. Alternative and Augmentative Communication Based Support Technology 2002, 7-8 [en línea] [fecha de consulta agosto de 2012]. Disponible en: <http://www.iberchip.net/iberchip2004/articles/39-3-AL-VAREL-ALVAROVARELA.PDF>

- Dogan H. *Pic Basic Projects: 30 Projects Using Pic Basic and Pic Basic Pro*, 1a ed., Londres UK, Newnes MPG Books, 2006, pp. 9-71.
- Domínguez F.R., Palacios-Municio E., López-J. L. *Desarrollo de proyectos microcontrolador PIC16F84A*, México, editorial Alfaomega, 1991, pp. 289-294.
- García-Breijo E. *Compilador C CCS y Simulador proteus para microcontroladores pic*, 2a ed., Barcelona, Marcombo, 2009, pp. 167-190.
- Kangbin Y. A Fix to the HCI Specification to Evade ID and Password Eexposure by USB Sniff. *Proceedings of APIC-IST 2008*, volumen 12, 2008: 191-194.
- Li Y. A Heuristic-Based Approach to Optimize Keyboard Design for Single-Finger Keying Applications. *International Journal of Industrial Ergonomics*, volumen 36, 2006: 695-704.
- Manual PIC16F84A, Data Sheet, Microchip Technology Inc 2001, [en línea] [fecha de consulta: febrero de 2012]. Disponible en: <http://ww1.microchip.com/downloads/en/devicedoc/5007b.pdf>
- Navabi Z. *Digital Design and Implementation with Field Programmable Devices*, Information Technology: Transmission, Processing and Storage, Springer, 2004 [en línea]. Disponible en: <http://books.google.com.mx/books?id=cPv4fHOxcVoC&pg=PA260&dq=design+and++implementation+of+keyboard&hl=es&sa=X&ei=a7efUPeVMYXrrQGNyYCwBA&ved=0CDAQ6AEwAA#v=onepage&q&f=false>
- Peng-Yeng Y. y En-Ping S. Cyber Swarm Optimization for General Keyboard Arrangement Problem. *International Journal of Industrial Ergonomics*, volumen 41, 2011: 43-52.
- Ramírez-Cortés J.M., Gómez-Gil P., Martínez-Carballido J., López-Larios F. A LabVIEW-Based Autonomous Vehicle Navigation System using Robot Vision and Fuzzy Control. *Ingeniería Investigación y Tecnología*, volumen 12 (número 2), 2011: 129-136.
- Suárez-Domínguez E., González-Marrero F.A. Diseño y desarrollo de la interfaz de comunicación del sistema diramic. *Revista CENIC*, volumen 36, 2005: 1-10.

Este artículo se cita:

Citación estilo Chicago

Mejía-Ugalde, Mario. Teclado con protocolo de comunicación universal aplicado a máquina de CNC. *Ingeniería Investigación y Tecnología*, XV, 02 (2014): 209-220.

Citación estilo ISO 690

Mejía-Ugalde M. Teclado con protocolo de comunicación universal aplicado a máquina de CNC. *Ingeniería Investigación y Tecnología*, volumen XV (número 2), abril-junio 2014: 209-220.

Semblanza del autor

Mario Mejía-Ugalde. Es estudiante de doctorado y profesor en la Facultad de Ingeniería de la Universidad Autónoma de Querétaro (UAQ). Recibió la licenciatura como ingeniero electromecánico y la maestría en instrumentación y control automático por parte de la Universidad Autónoma de Querétaro. Sus intereses de investigación actuales son diseño implementación y optimización mediante software para la automatización de máquina-herramienta de fabricación.