



CCADET

CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Informe Técnico

Comunicación entre Controladores Utilizando el Protocolo RS-485

**Rafael Prieto Meléndez, Alejandro Padrón Godínez,
Alberto A. Herrera Becerra, C. Jonathan Luna Jaimes,
Raúl Jiménez Hernández**

Diciembre 2009

Proyecto Interno

Financiamiento CCADET

Título: Comunicación entre Controladores Utilizando el Protocolo RS-485

Autores: Rafael Prieto Meléndez, Alejandro Padrón Godínez, Alberto A. Herrera Becerra, C. Jonathan Luna Jaimes, Raúl Jiménez Hernández.

Afiliación: Centro de Ciencias Aplicadas y Desarrollo Tecnológico, Universidad Nacional Autónoma de México.

Resumen

En este trabajo se presenta la implantación de la comunicación de controladores con una computadora personal utilizando el protocolo de comunicación serial RS-485. El desarrollo de estos controladores surge como consecuencia de la colaboración con la empresa QMatic de México SA de CV, la cual tiene la intención de sustituir a los elementos utilizados en sus sistemas, que son importados, por otros de fabricación nacional utilizando tecnología propia. Los controladores de sensores ópticos que actualmente utilizan se comunican entre si por medio de un puerto RS-485 formando una cadena de comunicación, en donde el primer controlador de la cadena se comunica con la computadora central por medio del puerto RS-232 con que cuenta la computadora. Los controladores que se desarrollaron tienen la función de controlar a un par de sensores ópticos para detectar el paso de personas así como la dirección en que pasan por una vía de acceso, y con esto llevar un control del número de personas que pasan por ese punto. La computadora central se comunica por poleo con cada uno de los controladores para conocer y actualizar el conteo general de personas que transitan por los puntos en observación.

Índice

Introducción	3
1. Comunicación Serial	4
1.1. Estándar RS-232	5
1.2. Estándar RS-422	5
1.3. Estándar RS-485	5
2. Comunicación entre Controladores	5
2.1. Prueba de Comunicación RS-485 entre Microcontroladores	6
2.2. Comunicación Serial de Computadora Personal a los Microcontroladores	7
Conclusiones	11
Referencias Bibliográficas	12

Introducción

La empresa QMatic de México SA de CV se dedica a proveer sistemas para el manejo de colas de personas, utilizando para ello dispositivos electrónicos de fabricación brasileña junto con sensores de origen chino (Prieto *et al*, 2007). En la operación de sus sistemas se encontraron con el problema de la falla recurrente de sus sensores bajo ciertas condiciones de operación, complicándose su situación al no contar con información técnica de los componentes que utilizaban. De ahí surgió la necesidad de sustituir esos elementos por otros de tecnología propia. Al grupo de Modelado y Simulación de Procesos del CCADET nos encargaron hacer una evaluación técnica de su sistema y proponer opciones para solucionar su problema. Como consecuencia de ello, primero se diseñó un circuito de protección para sus sensores con la idea de resolver el problema urgente (Prieto *et al*, 2006), y posteriormente se les diseñó un nuevo sensor óptico (Prieto *et al*, 2008). Para completar el proceso de sustitución de tecnología se propuso a la empresa el diseño de una tarjeta de control para los sensores.

Las tarjetas de control realizan diversas funciones, como son la de proporcionar el voltaje de alimentación para los sensores, y de recibir la señal proveniente de ellos, acondicionarla, enviar el estado de los sensores hacia la computadora en que se concentra la información y permitir la comunicación entre tarjetas. En el presente trabajo se muestra el desarrollo de los mecanismos de comunicación entre las tarjetas de control basados en el protocolo de comunicación serial RS-485.

1. Comunicación Serial

La comunicación serial es un protocolo muy común (no hay que confundirlo con el Bus Serial de Comunicación, o USB) para comunicación entre dispositivos que se incluye de manera estándar en prácticamente cualquier computadora. La mayoría de las computadoras incluyen un puerto serial RS-232. Este tipo de comunicación es comúnmente utilizado para intercambiar información entre la computadora y algunos de sus periféricos. Además, la comunicación serial puede ser utilizada para la adquisición de datos si se usa en conjunto con un dispositivo remoto de muestreo.

El concepto de comunicación serial es sencillo. El puerto serial envía y recibe bytes de información un bit a la vez. Aun y cuando esto es más lento que la comunicación en paralelo, que permite la transmisión de un byte completo por vez, este método de comunicación es más sencillo y puede alcanzar mayores distancias.

Típicamente, la comunicación serial se utiliza para transmitir datos en formato ASCII. Para realizar la comunicación se utilizan 3 líneas de transmisión: (1) Tierra (o referencia), (2) Transmisión, (3) Recepción. Debido a que la transmisión es asincrónica, es posible enviar datos por una línea mientras se reciben datos por otra. Existen otras líneas disponibles para realizar *handshaking*, o intercambio de pulsos de sincronización, pero dependiendo de la aplicación pueden no ser requeridas. Las características más importantes de la comunicación serial son la velocidad de transmisión, los bits de datos, los bits de parada, y la paridad. Para que dos puertos se puedan comunicar, es necesario que las características sean iguales. Estas características se describen a continuación.

- a. **Velocidad de transmisión (*baud rate*):** Indica el número de bits por segundo que se transfieren, y se mide en baudios (bauds). Por ejemplo, 300 baudios representa 300 bits por segundo. Cuando se hace referencia a los ciclos de reloj se está hablando de la velocidad de transmisión. Por ejemplo, si el protocolo hace una llamada a 4800 ciclos de reloj, entonces el reloj está corriendo a 4800 Hz, lo que significa que el puerto serial está muestreando las líneas de transmisión a 4800 Hz. Las velocidades de transmisión más comunes para las líneas telefónicas son de 14400, 28800, y 33600. Es posible tener velocidades más altas, pero se reduciría la distancia máxima posible entre los dispositivos. Las altas velocidades se utilizan cuando los dispositivos se encuentran uno junto al otro.
- b. **Bits de datos:** Se refiere a la cantidad de bits en la transmisión. Cuando la computadora envía un paquete de información, el tamaño de ese paquete no necesariamente será de 8 bits. Las cantidades más comunes de bits por paquete son 5, 7 y 8 bits. El número de bits que se envía depende en el tipo de información que se transfiere. Por ejemplo, el ASCII estándar tiene un rango de 0 a 127, es decir, utiliza 7 bits; para ASCII extendido es de 0 a 255, lo que utiliza 8 bits. Si el tipo de datos que se está transfiriendo es texto simple (ASCII estándar), entonces es suficiente con utilizar 7 bits por paquete para la comunicación. Un paquete se refiere a una transferencia de byte, incluyendo los bits de inicio/parada, bits de datos, y paridad. Debido a que el número actual de bits depende en el protocolo que se seleccione, el término paquete se usará para referirse a todos los casos.
- c. **Bits de parada:** Usado para indicar el fin de la comunicación de un solo paquete. Los valores típicos son 1, 1.5 o 2 bits. Debido a la manera como se transfiere la información a través de las líneas de comunicación y que cada dispositivo tiene su propio reloj, es posible que los dos dispositivos no estén sincronizados. Por lo tanto, los bits de parada no sólo indican el fin de la transmisión sino además dan un margen de tolerancia para esa diferencia de los relojes. Mientras más bits de parada se usen, mayor será la tolerancia a la sincronía de los relojes, sin embargo la transmisión será más lenta.
- d. **Paridad:** Es una forma sencilla de verificar si hay errores en la transmisión serial. Existen cuatro tipos de paridad: par, impar, marcada y espaciada. La opción de no usar paridad alguna también está disponible. Para paridad par e impar, el puerto serial fijará el bit de

paridad (el último bit después de los bits de datos) a un valor para asegurarse que la transmisión tenga un número par o impar de bits en estado alto lógico. Por ejemplo, si la información a transmitir es 011 y la paridad es par, el bit de paridad sería 0 para mantener el número de bits en estado alto lógico como par. Si la paridad seleccionada fuera impar, entonces el bit de paridad sería 1, para tener 3 bits en estado alto lógico. La paridad marcada y espaciada en realidad no verifican el estado de los bits de datos; simplemente fija el bit de paridad en estado lógico alto para la marcada, y en estado lógico bajo para la espaciada. Esto permite al dispositivo receptor conocer de antemano el estado de un bit, lo que serviría para determinar si hay ruido que esté afectando de manera negativa la transmisión de los datos, o si los relojes de los dispositivos no están sincronizados.

Para la comunicación serial existen diversos estándares que son de uso común para la comunicación de dispositivos con una computadora. Algunos de los más utilizados son los siguientes:

1.1. Estándar RS-232

RS-232 (Estándar ANSI/EIA-232) es el conector serial hallado en las PCs IBM y compatibles. Es utilizado para una gran variedad de propósitos, como conectar un ratón, impresora o modem, así como instrumentación industrial. Gracias a las mejoras que se han ido desarrollando en las líneas de transmisión y en los cables, existen aplicaciones en las que se aumenta el desempeño de RS-232 en lo que respecta a la distancia y velocidad del estándar. RS-232 está limitado a comunicaciones de punto a punto entre los dispositivos y el puerto serial de la computadora. El hardware de RS-232 se puede utilizar para comunicaciones seriales en distancias de hasta 15 metros.

1.2. Estándar RS-422

RS-422 (Estándar EIA RS-422-A) es el conector serial utilizado en las computadoras Apple de Macintosh. RS-422 usa señales eléctricas diferenciales, en comparación con señales referenciadas a tierra como en RS-232. La transmisión diferencial, que utiliza dos líneas para transmitir y recibir, tiene la ventaja que es más inmune al ruido y puede lograr mayores distancias que RS-232. La inmunidad al ruido y la distancia son dos puntos clave para ambientes y aplicaciones industriales.

1.3. Estándar RS-485

RS-485 (Estándar EIA-485) es una mejora sobre RS-422 ya que incrementa el número de dispositivos que se pueden conectar (de 10 a 32) y define las características necesarias para asegurar los valores adecuados de voltaje cuando se tiene la carga máxima. Gracias a esta capacidad, es posible crear redes de dispositivos conectados a un solo puerto RS-485. Esta capacidad, y la gran inmunidad al ruido, hacen que este tipo de transmisión serial sea la elección de muchas aplicaciones industriales que necesitan dispositivos distribuidos en red conectados a una PC u otro controlador para la colección de datos u otras operaciones. RS-485 es un conjunto que cubre RS-422, por lo que todos los dispositivos que se comunican usando RS-422 pueden ser controlados por RS-485. El hardware de RS-485 se puede utilizar en comunicaciones seriales de distancias de hasta 1200 metros de cable.

2. Comunicación entre Controladores

Debido a las ventajas ofrecidas por el protocolo, aunado al hecho de la conveniencia de usarlo para mantener la compatibilidad con las tarjetas existentes, se pretende utilizar el protocolo RS-485 tanto para comunicación entre las tarjetas, así como también para la comunicación con la computadora. Para hacer la implantación del mecanismo de comunicación, se dividió en dos partes el trabajo, por un lado la comunicación entre controladores, lo cual implica la programación de los microcontroladores para que puedan intercambiar información entre ellos, y por otro lado la comunicación con la PC, que en el sistema final va a tomar el papel del dispositivo maestro en la cadena de comunicación, y además en esta etapa se añade la programación correspondiente en la computadora.

2.1. Prueba de Comunicación RS-485 entre Microcontroladores

En esta primera fase se realizó el procedimiento necesario para comunicar dos PICs mediante el protocolo RS-485, para estas pruebas se utilizó el PIC16F628A, que es un PIC de la gama media, dándole los atributos de maestro al primer PIC y como esclavo al segundo. En la figura 1 se muestra el sistema armado para probar este tipo de comunicación.

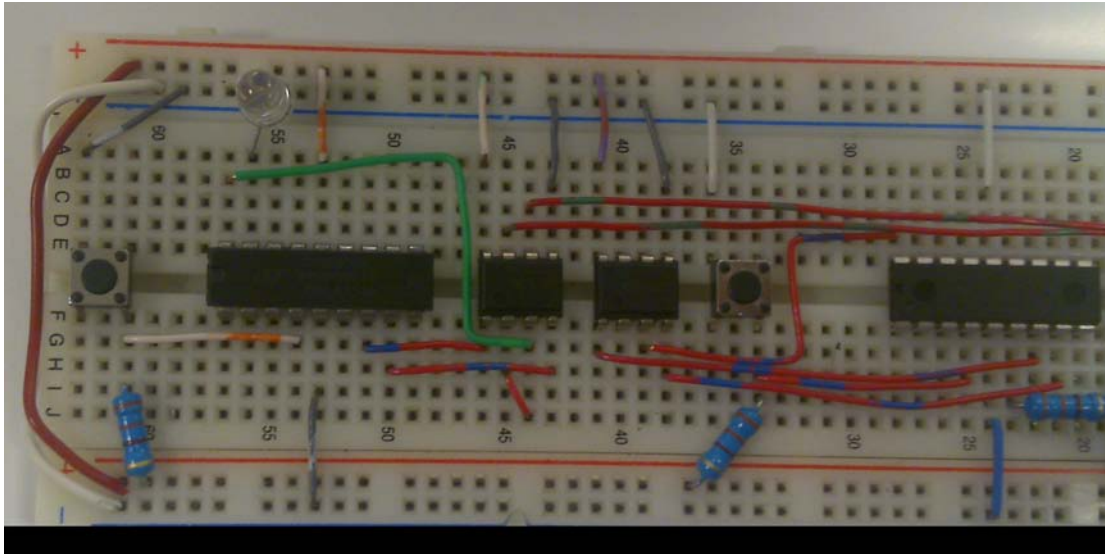


Figura 1. Circuito utilizado para evaluar la comunicación RS-485.

Para evaluar la comunicación entre controladores se hizo la programación de los microcontroladores diferenciando la función que cada uno realiza en el proceso, por lo que se tiene la programación del primer controlador como maestro, que es quien controla el bus y realiza el envío de datos, y el segundo controlador, que se encuentra configurado como esclavo, está escuchando en el bus para captar la información que viaja por ahí, y recibe el paquete de información que va dirigido a él.

Código en lenguaje C para el microcontrolador que actúa como maestro:

```
////////////////////////////////////
//TITLE: ENVIO DE BIT DE PIC A PIC
////////////////////////////////////

//CONFIGURACION////////////////////////////////
#include<16f628a.h>
#fuses INTRC_IO, NOWDT, NOLVP, MCLR, NOPROTECT
#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_B2, rcv=PIN_B1)
retorna(int c);
void main(void){
    int a;
    set_tris_b(0x00);
    set_tris_a(0x21);
    while(true){
        output_HIGH(PIN_A1);
        a=input(PIN_A0);
        putc(a);
    }
}
```

Código en lenguaje C para el microcontrolador que actúa como esclavo:

```
////////////////////////////////////
//TITLE: RECEPCION PIC A PIC
////////////////////////////////////

//CONFIGURACION////////////////////////////////
#include<16f628a.h>
#define INTRC_IO, NOWDT, NOLVP, MCLR, NOPROTECT
#define delay(clock=4000000)
#define rs232(baud=9600, xmit=PIN_B2, rcv=PIN_B1)
returna(int c);
int lectura;
#define INT_RDA //interrupcion para la recepcion del master
RDA_isr()
{
    lectura=getc(); //guarda el valor enviado por el RS-485
}
void main(void)
{
    set_tris_b(0x00);
    enable_interrupts(GLOBAL); //todas las interrupciones activadas
    enable_interrupts(INT_RDA); //interrupcion INT_RDA activada. comunicacion
    RS-485

    /*
    * Pin RC5 que habilita la transmisión por el canal RS485* \li LOW Habilita
    recepción RS485
    * \li HIGH Habilita transmisión RS485.*
    while(1){
        output_low(PIN_A1);
        //lectura=(PIN_A0);
        output_b(lectura);
    }
    }
    */
```

2.2. Comunicación Serial de Computadora Personal a los Microcontroladores

En la segunda fase de desarrollo se integró al sistema la comunicación con la computadora tipo PC, utilizando para ello el puerto serial con que cuenta este equipo. La computadora se utiliza como dispositivo maestro, es quién controla el acceso al bus y envía instrucciones a las diferentes tarjetas conectadas. Para verificar la autenticación de las tarjetas, se cuenta con el ingreso de una llave de identificación que nos va a identificar de forma única a cada una de las tarjetas, para que se pueda indicar la acción que se desea que realice una tarjeta en lo específico. Para el envío de comandos desde la PC se utilizó una hiper-terminal que nos permite enviar de forma directa comandos a las diferentes tarjetas y en donde también podemos visualizar la actividad en el bus para obtener el mensaje de respuesta de cualquier controlador. En la figura 2 se observa el circuito con las modificaciones para integrar la comunicación con la PC. En la figura 3 se muestra el diagrama esquemático del circuito para acoplar las señales provenientes de la PC, mientras que en la figura 4 se muestra la forma de integrarlo al sistema para comunicarlo con los microcontroladores.

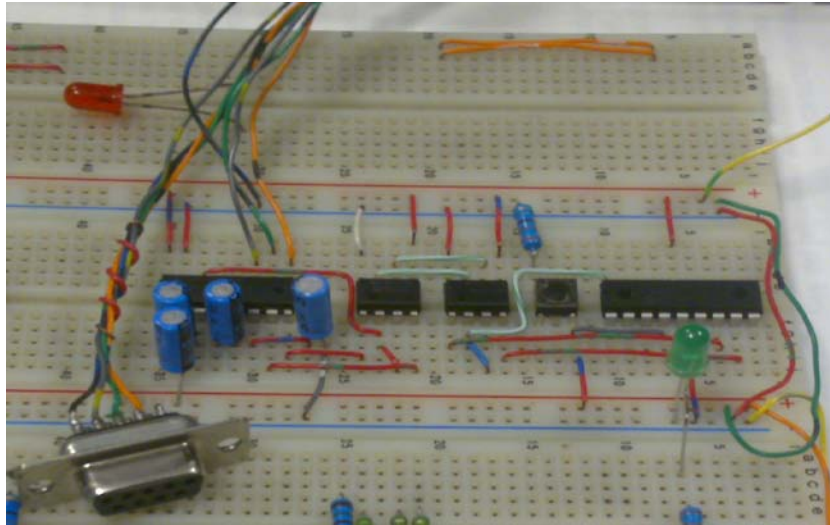


Figura 2. Circuito donde se agrega la bus RS-485 la comunicación con la PC.

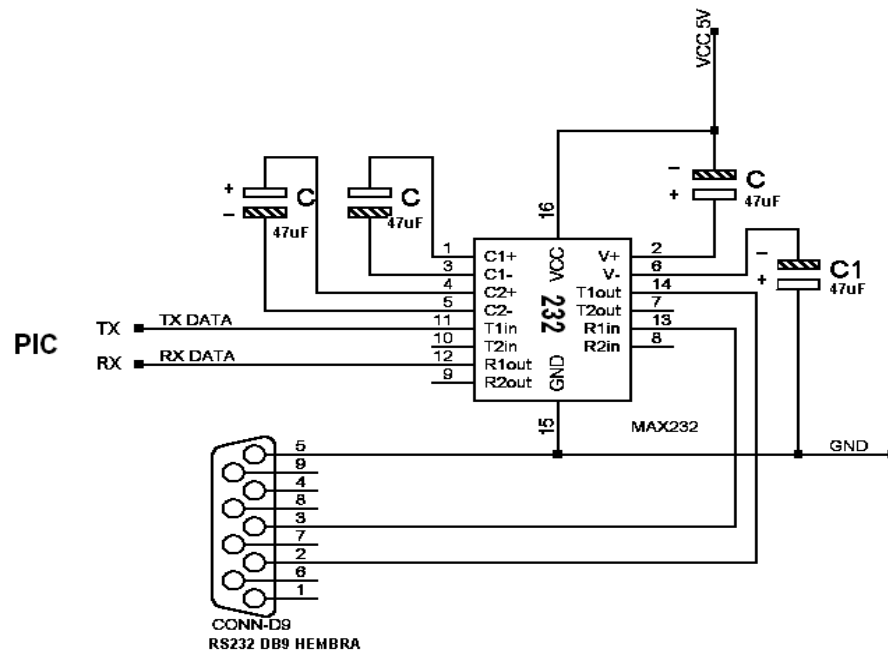


Figura 3. Diagrama del circuito para acoplar las señales de la PC.

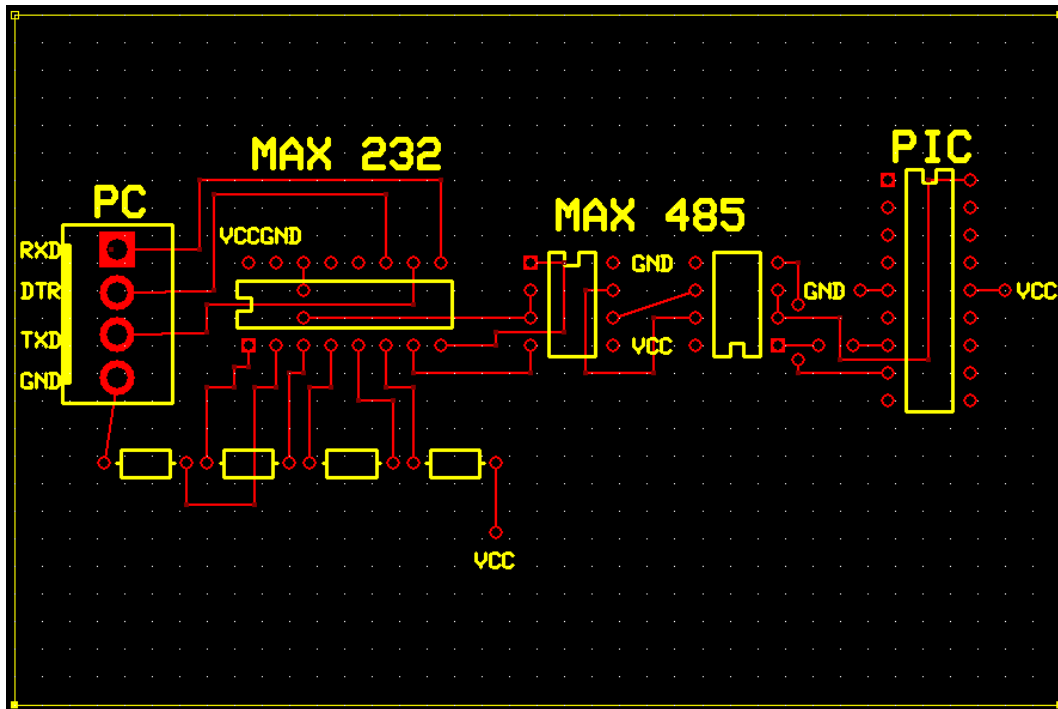


Figura 4. Diagrama esquemático de la integración de la PC con los controladores.

Código en lenguaje C para los microcontroladores con la autenticación de tarjeta:

```

////////////////////////////////////
//TITLE: AUTENTICACION DE TARJETAS
////////////////////////////////////

//CONFIGURACION////////////////////////////////
#include <16f628a.h>
#include <stdlib.h>
#include <string.h>
#fuses INTRC_IO,NOWDT,NOLVP,MCLR,NOPROTECT
#byte PORTB=0x06 //definicion del PORTB
#use delay(clock=4000000)
#use rs232(baud=9600,xmit=PIN_B2,rcv=PIN_B1)
//set_tris_b(0xF0); // entrada parte alta de B
//set_tris_a(0x00); // salida el puerto a
void main ()
{
    char buffer[20],clave[10],buffer1[10];
    int cont;
    cont=0;
    while(1){

        printf("\n\rEsperando clave de autenticacion: \n\r");
        gets(buffer);
        strcpy(buffer1, buffer);
        strcpy(clave, "31BF");
        if(strncmp(buffer1, clave,4) == 0){
            printf("\n\r La clave es correcta: %s\n\r",buffer);

```

```

while(1){
    #asm          //modulo asm para manejar
    SISTEMA:      BTFSS PORTB,4
                  goto INCREMENTA1
                  goto DECREMENTA1

    INCREMENTA1:  NOP
    ENTSEN1:      BTFSS PORTB,4      ;COMPARA EL BIT QUE LLEGA AL PUERTO
A EN LA POSICION CERO
                  GOTO SISTEMA      ;SE REGRESA SI ESTA EN CERO
                  GOTO ENTSEN2      ;SI ESTA ENCENDIDO SALTA A
                                      ; ENTSEN2
    ENTSEN2:      BTFSS PORTB,5      ;COMPARA EL BIT QUE LLEGA AL
                                      ; PUERTO A EN LA POSICION UNO
                  GOTO ENTSEN1      ;SI ESTA EN CERO SE REGRESA A
                                      ; ENTSEN1
                                      ;VERIFICA EL SENSOR 1
    APASEN1:      BTFSC PORTB,4
                  GOTO ENTSEN2
                  GOTO APASEN2
    APASEN2:      BTFSC PORTB,5
                  GOTO APASEN1
                  GOTO INCREMENTA
    INCREMENTA:   INCF cont
                  GOTO FIN
    DECREMENTA1:  NOP
    SEN2:         BTFSS PORTB,5
                  GOTO SISTEMA
                  GOTO SEN1
    SEN1:         BTFSS PORTB,4
                  GOTO SEN2
                  GOTO APAGSEN2
    APAGSEN2:     BTFSC PORTB,5
                  GOTO SEN1
                  GOTO APAGSEN1
    APAGSEN1:     BTFSC PORTB,4
                  GOTO APAGSEN2
                  GOTO DECREMENTA
    DECREMENTA:   DECF cont
                  GOTO FIN
    FIN:          NOP
    #endasm
    printf("\n Personas en la Fila: %d \n\r",cont);
}
}
else{
    printf("\n La clave que me diste es: %s \n\r",buffer);
    printf("\n NO ES CORRECTA \n\r");
}
}
}

```

Conclusiones

En el presente trabajo se muestra una implantación funcional del protocolo serial RS-485 utilizado para el intercambio de datos entre dos o más sistemas basados en un microcontrolador tipo PIC. Esta solución puede ser implantada en diferentes modelos de este microcontrolador, incluso de la gama menor, en donde la única limitante es la velocidad a la que se pretende establecer la comunicación, ya que el consumo de recursos del controlador es muy reducido, y solamente requiere que el microcontrolador tenga un puerto de comunicaciones UART (o USART).

La idea de utilizar esta forma de comunicación entre tarjetas se toma buscando que las tarjetas que se desarrollen sean totalmente compatibles con las existentes y puedan remplazarlas paulatinamente y según se vaya requiriendo. El único aspecto que se está evaluando y que diferenciaría a la nueva tarjeta con las adquiridas comercialmente que se actualmente se están utilizando es en lo relativo a la comunicación de la primer tarjeta con la computadora personal, ya que las tarjetas existentes cuentan con dos tipos de puertos, uno RS-232 y uno RS-485, utilizando el puerto RS-232 únicamente para enlazar a la primer tarjeta de la cadena RS-485 con la PC, tomando esta primer tarjeta la responsabilidad de hacer el puente de comunicación entre los dos protocolos. Para la tarjeta que se está diseñando se está evaluando la posibilidad de que la comunicación se haga de manera uniforme entre todos los elementos, incluyendo la PC, en donde ésta se encargaría de arbitrar el intercambio de información.

Queda como trabajo a futuro el completar ésta evaluación encaminada a eliminar la comunicación RS-232 sin perder la compatibilidad con la tarjetas existentes para hacer la integración con el control de los sensores y con esto poder ofrecer la tarjeta resultante a la empresa QMatic de México SA de CV como un sustituto viable para remplazar a las tarjetas importadas.

Referencias Bibliográficas

Prieto Meléndez, R., A. A. Herrera Becerra, A. Padrón Godínez (2007): *Evaluación de Funcionamiento del Sistema ADMINISTRACIÓN DEL FLUJO DE CLIENTES* (Informe técnico de final de proyecto), UNAM-CCADET, México D.F., 19 pp.

Prieto Meléndez, R., A. Padrón Godínez, A. A. Herrera Becerra (2006): *Diseño de un Circuito de Protección para Sensores Ópticos* (RTMSP-0603), UNAM-CCADET, México D.F., 11 pp.

_____ (2008): *Sensor Óptico Retrorreflexivo con Capacidad de Medición de Intensidad del Haz* (MOD_RT-0801), UNAM-CCADET, México D.F., 21 pp.