



CardSharing Project

Según ya nos indica su propio nombre, el circuito trata de compartir algo, en este caso, la comunicación de una tarjeta de acceso inteligente. Se trata de llevar al mismo tiempo las respuestas de la tarjeta a dos o mas lectores. Una tarjeta inteligente de acceso esta diseñada para ser usada desde un lector único. Pero, que sucede si miembros de una misma familia, pero separadas geográficamente en un momento dado, desean compartir los mismos datos? La respuesta se llama CardSharing.

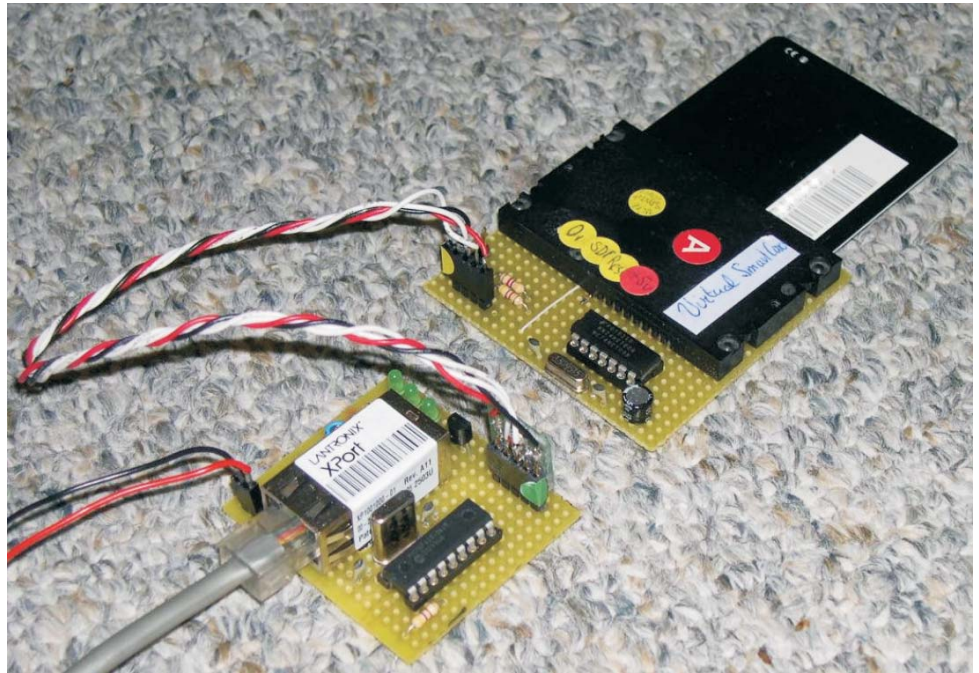
Disfrutar de la suscripción a un canal en otra parte de la vivienda, cuando se trata del mismo abonado no es ningún delito, como tampoco lo es compartir esta suscripción con otro Receptor Descodificador situado en otra superficie siempre que éste, sea del propio abonado, véase casa de verano. Que sucede cuando miembros de una familia están separados por la geografía?. Pues que no pueden tener la tarjeta de acceso condicional en ambos sitios, ya que la Plataforma a la que se está abonada solo distribuye una sola Card. Para solventar este problema

EL CIRCUITO CLIENTE

Un cliente, es un módulo encargado de recibir información del exterior, de una ReD o de un enlace, pero que provenga del lado Server. El modulo Cliente puede ser por Hardware o por Software. Entonces sabemos que podemos hablar de concepto P2P. Aplicaciones para compartir. En CardSharing, un cliente selecciona y recibe una serie de datos idéntico al que estaría suministrando una tarjeta de Acceso Inteligente a su lector respectivo. Por ejemplo si una cabecera esta compuesta por los valores C1 3A 58 00, esta deberá ser idéntica en ambos lados del CardSharing, es decir, cliente y server.

Pero teniendo en cuenta los respectivos tiempos de espera y respuesta, véase sincronización. Sin este valor añadido un Cliente no podría funcionar correctamente en otro lector de tarjetas. Para sincronizar estos tiempos el cliente debe "simular" correctamente la tarjeta desde el propio ATR, "respuesta de la tarjeta nada mas ser insertada al lector.

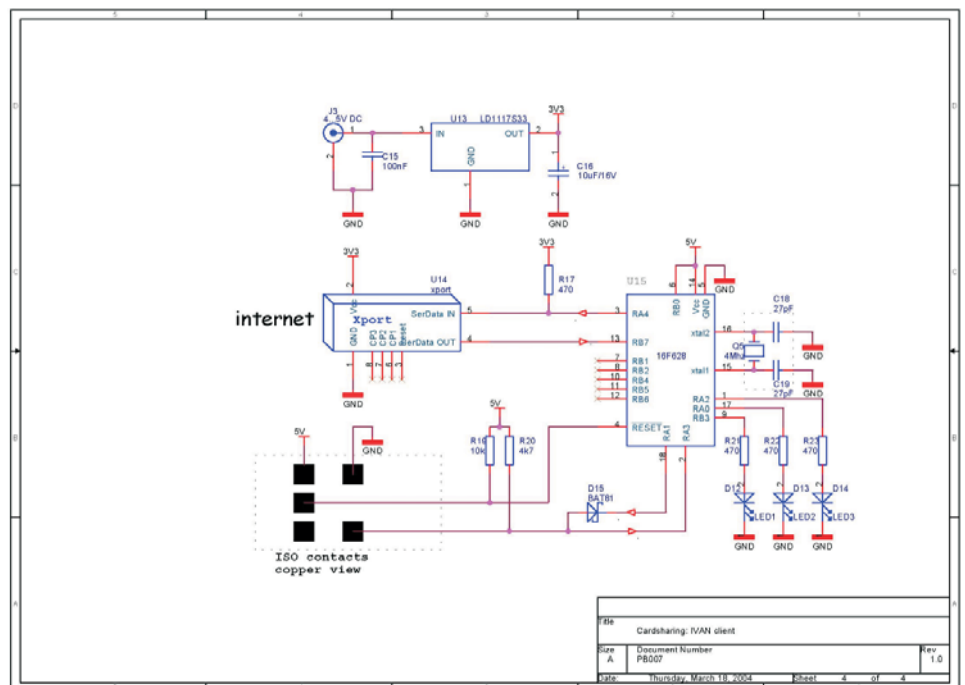
Además y solucionado esta fase del sistema, debemos tener en cuenta que los datos viajan a través de una ReD, ya sea interna o Extranet. En cualquier caso, debemos encriptar estos datos para que no



sean espiados por terceros. Desencriptarlos y suministrarlos de forma sincronía al lector. Un lector en este caso es un receptor de satélite con CAS integrado. Un CAS es un sistema de acceso condicional empujado en el firmware del receptor. Por lo tanto, esta unidad "interrogará" a la supuesta tarjeta que esta conectada al lector, en este caso el Cliente del CardSharing. Resumido el proceso vayamos a concretar el esquema; como se puede apreciar en el esquema, el circuito es bastante sencillo, tanto como mencionar sólo dos componentes importantes aquí. Se trata del

módulo Xport y el Pic 16F628. Por lo que los secretos en este circuito están entre el Firmware del PIC y el Xport.

- Los valores de ATR, Cabecera o demás son valores ficticios y no corresponden a ningún sistema de acceso condicional actual.
- El Xport es el componente mas caro del circuito, pero también encierra en su interior toda una circuiteria necesaria como para ser conectado como un Device mas en la ReD de Internet.



EL LADO SERVIDOR

El lado servidor no es mucho más complejo que el lado Cliente. También aquí los valores reposan sobre el PIC y el módulo Xport de Lantronix. Si en el lado cliente la velocidad de trabajo está en base a un cristal externo de 4 Mhz, aquí el cristal externo debe ser de 3.57 Mhz. En realidad el valor del cristal en el lado Server va en función de la tarjeta de acceso inteligente que se desea compartir. Otra cosa determinante en el lado Server es la combinación Receptor/tarjeta que podemos tener. Nos referimos a que es posible hacer funcionar la tarjeta de forma independiente para varios lectores a la vez, lo que nos lleva en realidad a un enjambre de conexiones entre PC – Receptor – Ethernet.

EL XPORT, ESA PIEZA TAN IMPORTANTE

El corazón de todo este invento en realidad es el Xport, ya que este dispositivo nos permite enviar y recibir los datos necesarios entre dos dispositivos interconectados en la distancia. Este módulo se entrega totalmente sellado. En su interior reposa toda una compleja circuitería que nos permite interconectar diferentes dispositivos en red. El Xport funciona bajo el modo Tunneling o modificando su Firmware. En Internet existen numerosas páginas que ofrecen códigos libres para trabajar con este dispositivo. En nuestra ocasión nos bastará con el efecto Tunneling, es decir, tal cual se nos en-



trega el Xport. Recordemos que aquí los datos se generan desde y hasta la tarjeta de acceso inteligente.

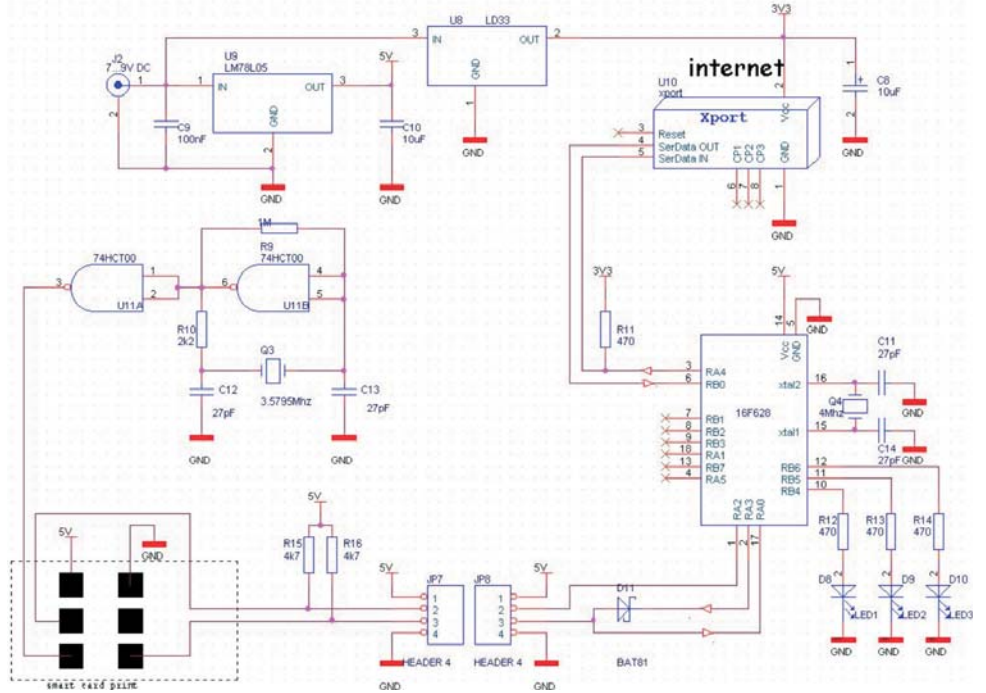
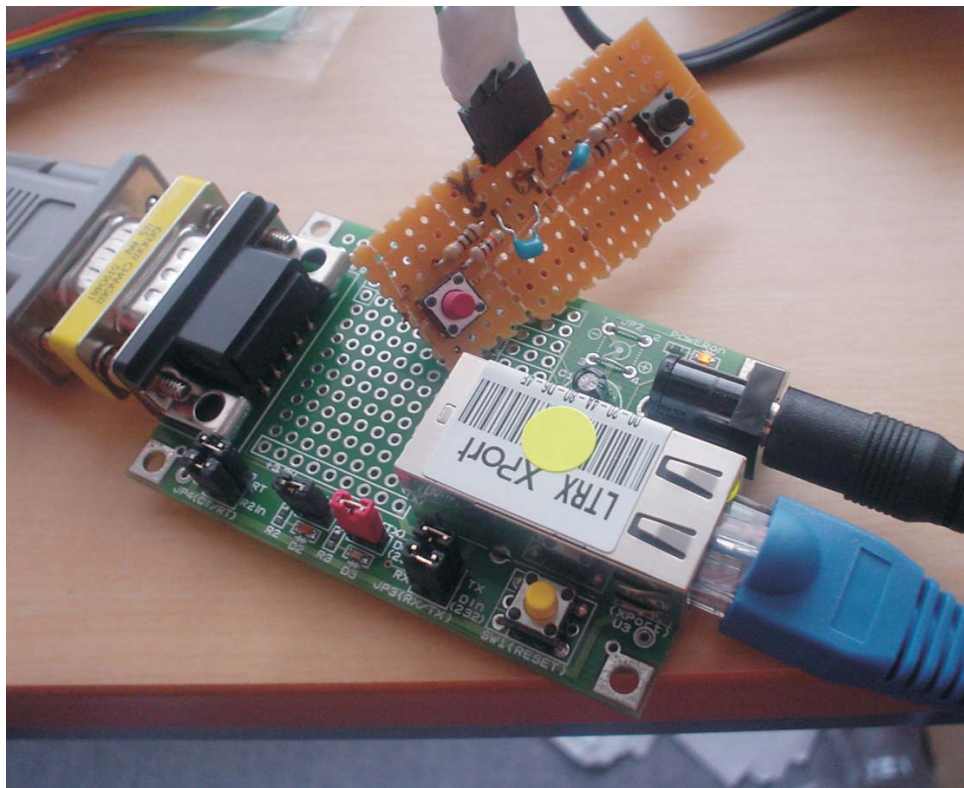
El presente módulo pues, puede funcionar en serie con un Pic o bien directamente desde el propio PC a través de un convertidor de tensiones RS232. Normalmente se vende en forma de Kit un circuito que adopta la solución RS232. Adaptar este circuito a nuestras necesidades es tarea bastante fácil.

En esta imagen vemos el módulo Xport sobre una placa de desarrollo. Como se puede apreciar en la fotografía el Xport es totalmente autónomo. Para comunicar-

se con el PC solo basta añadir un circuito RS232. Recuerde que todos estos módulos pueden funcionar como Devices independientes y que todos pueden ser redirigidos desde un solo PC a través de una Interfaz de Web. Si por el contrario no quiere emplear el PC, puede asignarles a cada Device una IP estática.

- Puede adquirir el presente Kit en Coelma
- Puede adquirir el Kit CardSharing en Coelma
- Puede adquirir los módulos Cliente y Servidor en Coelma

Imágenes del prototipo de desarrollo;





CardSharing II

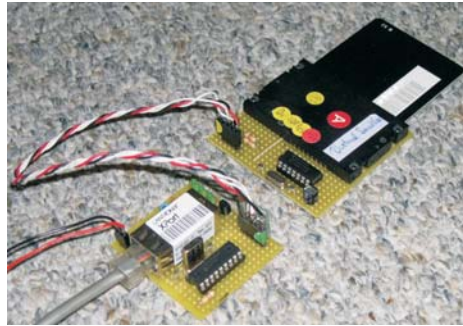
En la anterior ocasión hablamos del concepto del CardSharing entre comillas, si bien se podían ver los esquemáticos de cada circuito que la componen, no estaba muy del todo claro, como funcionan las diferentes formas de hacer CardSharing. En buena parte, para conocer el correcto funcionamiento de esto, debemos saber primero como manejar el Xport. En el presente circuito vamos a ver como es posible reenviar en la distancia las respuestas de una tarjeta de acceso condicional.

El circuito es tan básico como interconectar en ambos lados el modulo Xport y el Pic. Esto significa que tenemos el modulo Xport en modo Tunneling y que es el Pic quien toma el control de los datos que entran y salen de estos circuitos. Nuestra idea es realizar un circuito que nos permita conectar la tarjeta de Acceso Condicional en un lugar diferente de donde esta realmente el receptor. Esto nos permite de esta forma tener separados la tarjeta y el receptor, e incluso compartir esta misma en dos Receptores Digitales diferentes.

Para ello tenemos que tener dos módulos, el cliente y el Server. Con el presente circuito no será necesario tener al lado el PC de marras. En esta ocasión vamos a funcionar sin su presencia. Esto no quiere decir que no podamos usarlo nunca, podemos hacerlo por ejemplo, para programar el propio Xport.

El Cliente será aquí una DummyCard estándar, en lo que se refiere a un extremo de la tarjeta, en el otro lado de la misma tendremos el PIC y el modulo Xport. En este extremo estaremos conectados a una ReD local o Externa “*Internet”. Este circuito se alimenta con una tensión continua de 5 voltios para el PIC y de 3.3 voltios para el Xport. Para este ultimo tendremos que reducir la tensión en la propia placa a través de un regulador fijo.

El Clock necesario para la tarjeta “emulada aquí” se obtiene del propio PIC a partir de un oscilador externo de 4 Mhz. El Reset de la tarjeta también es tomada y/o enviada al PIC antes mencionado. Esto permite decidir al PIC cuando empezar a enviar las respuestas correctas que le vienen desde la Red. El pin de datos “siempre hablando del extremo de la tarjeta que posee los 5 contactos de cobre” es enviado a través de un diodo 1N4148 a dos pines del PIC con objeto de identificar entrada y salida de datos. 3 Leds conectados asimismo al PIC mediante una resistencia nos informará de los estados de comuni-



cación interna o mejor dicho, de los estados de respuesta/espera y Recepción. Volviendo a las líneas de Reset y Datos I/O recordamos que estas deben estar polarizadas a positivo desde los 5 voltios a través de una resistencia, para permitir una compatibilidad TTL adecuada en este punto. El completo contenido de este circuito lo podemos ver en la figura adjunta;

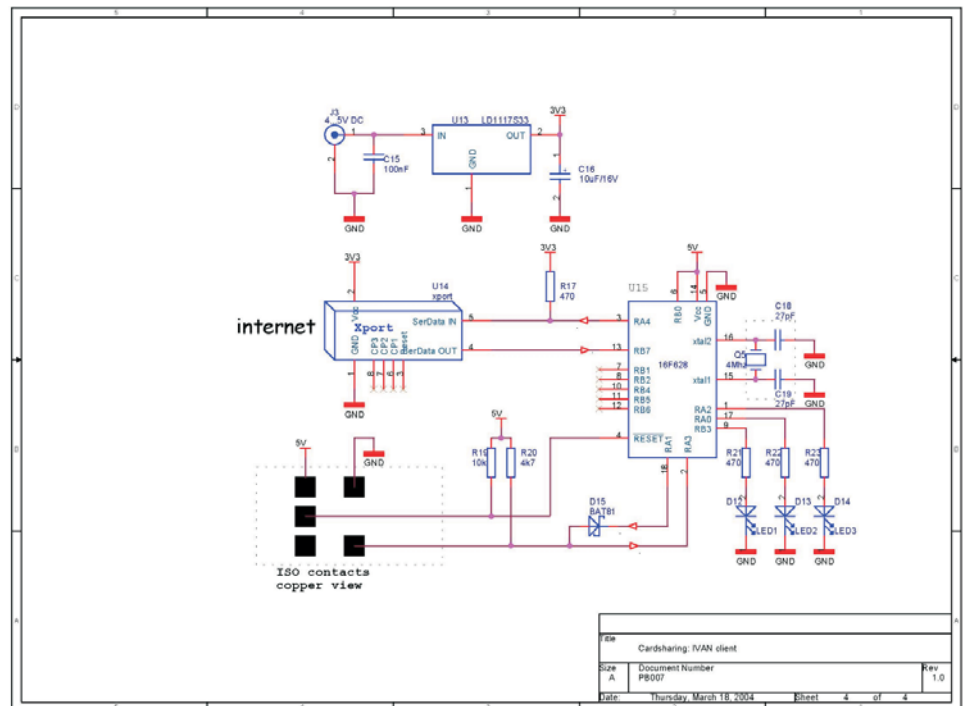
- *.Para ello será necesario que tengamos en la ReD un Router para acceder a lo que es Internet.

En el lado Server las cosas no se complican mucho mas. Según se puede apreciar en el esquema, de nuevo tenemos el Xport y el PIC. El trabajo aquí, de estos dos componentes es el de recibir los datos de la tarjeta y reenviarlos a través de la ReD. Para ello vemos como el Pin de datos I/O del lector de tarjetas se envía directamente sobre dos entradas del PiC. Aquí también enviamos el Reset, momen-

to en el cual la tarjeta responde con su ATR, lo que marca el inicio de las comunicaciones. Para el reloj se ha elaborado un subcircuito independiente al que se emplea para el PIC. La frecuencia aquí es de 3.57 Mhz creada a partir del propio cristal de cuarzo y las puertas NAND para su conversión TTL. La frecuencia de reloj del PiC es de 4 Mhz tal y como sucede en el lado Cliente. A través del modulo Xport recibimos las respuestas del otro “extremo de la ReD” mientras que lo mas importante aquí es el envío de los datos a la propia ReD. Como se puede apreciar el pin de datos de salida esta en modo Pus Pull, es decir polarizada a positivo a través de una resistencia. Las tensiones de alimentación aquí también son 2, los 5 voltios del circuito en si y los 3.3 voltios para el modulo Xport.

Ahora que ya sabemos como son las entrañas de este circuito solo queda saber como funciona realmente. Esto nos despertara mas de un interés por este circuito. La idea aquí es dejar el modulo Xport en modo tunneling, y seleccionarlo a través de una IP local. Este proceso se realiza desde el PC y a través del modo WEB Site. Esto se realiza con su Software específico.

El que se gestionen a través de una IP de identificación o redireccionamiento de direcciones, nos permite poder colocar tantos Clientes como deseemos. Si por ejemplo disponemos de una única tarjeta

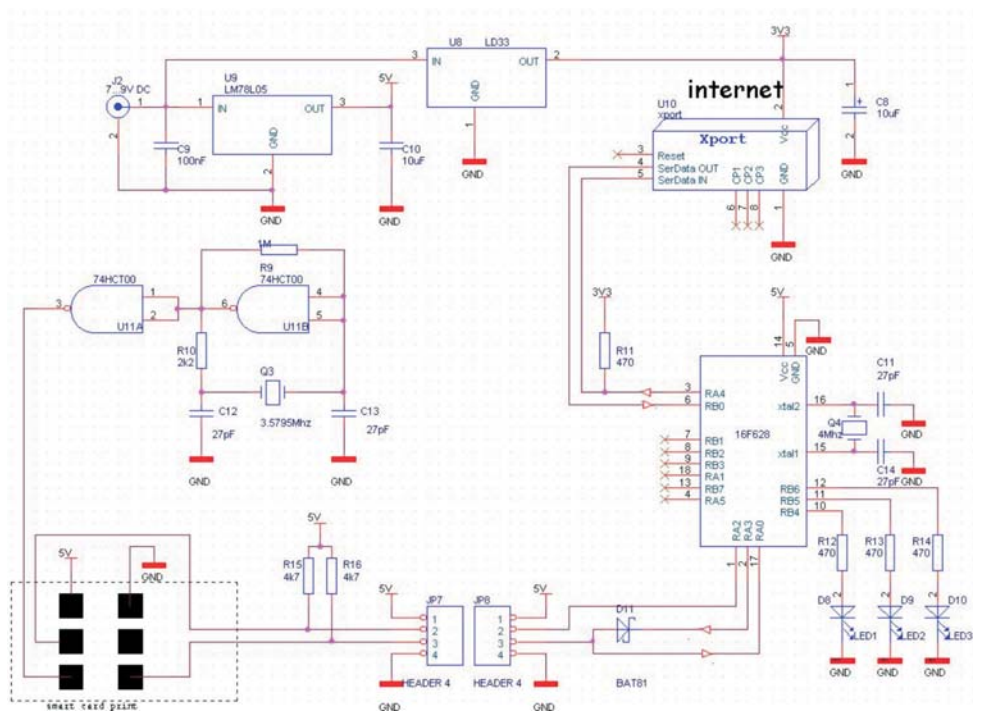


File	Cardsharing: IVAN client	Rev	1.0
Size	Document Number		
A	PB007		
Date	Thursday, March 18 2004	Sheet	4 of 4

de acceso condicional y deseamos compartirla entre las diferentes habitaciones de la misma casa, tenemos que insertar dicha tarjeta en el Server, que será el que distribuya los datos en la ReD LAN. Y poner tantos Clientes como nos hagan falta en la propia LAN.

Cuando insertamos la tarjeta de acceso condicional en el lector este produce un reset, siendo el PiC en encargado de recibir ahora el ATR de la tarjeta y redireccionarlo a la ReD. Normalmente tras un ATR, el lado Receptor empezara a solicitar una serie de datos de la tarjeta, de ahí el doble envío de datos en ambos sentidos, cliente/servidor Servidor/cliente. Los contenidos del PiC son los que gestionan esta comunicación de forma separada en la distancia.

Esto se consigue a través de los tan populares módulos Xport. Estos Convierten los datos TTL en datos de ReD y bajo los estándares del mismo. A partir de aquí se selecciona la IP a la que estará destinada y el lado Cliente recibirá dichas comunicaciones realizando el proceso inverso antes descrito. Llegados a este punto ya sabemos que lo que se pretende aquí es



tener el lector de tarjetas lo mas retirado posible del Receptor, sea este de cualquier naturaleza. Las comunicaciones entre tarjeta y Receptor deben ser tan fluidas y

transparentes como si esta ReD no existiera de por medio. Y por ultimo debe saber que la ReD puede ser interna LAN o Externa Internet.

Codigo en Assembler del Server;

```

*****
;
; Filename: iserver
; Date: 2004/2005
; File Version: V1.1: release with 16F628
; Smart Card Server
; Author:
; Company:
; Platform: 16F628 with oscillator of 4.0 Mhz
; Dev. platform: MPLAB 6.10,
*****

```

```

__CONFIG    0x3f09    ;no watchdog, internal reset, Ext. resona-
;
_IDLOCS     0x1234    ;
LIST P=16F628, R=DEC
#include P16F628.inc

```

```

counter     equ 0x20
TxByte      equ 0x21
delcnt      equ 0x22
bitcount    equ 0x23
rxreg       equ 0x24
maxadres    equ 0x25
Parity      equ 0x26
ident       equ 0x28
instr       equ 0x29
P1          equ 0x2A
P2          equ 0x2B
LEN         equ 0x2C

```

```

init:       bsf      org      0
            movlw   STATUS,RP0    ; switch to bank1
            movlw   0x54          ; OPTION register:
            movwf   OPTION_REG    ; - Prescaler rate 1:256
            movlw   0xEC          ; - Inputs: all; except

```

```

RA0,RA1,RA4
            movwf   TRISA         ;
            movlw   0x8F         ;
            movwf   TRISB         ; - Port B: output

```

```

bank0
            bcf      STATUS,RP0    ; switch to

```

```

            movlw   7
            movwf   CMCON         ; disable comparator,

```

```

make digital input
            clrf     INTCON
            movlw   0x11          ; RA0=1;RA4=1
            movwf   PORTA        ; Set serial port to xport

ready
            movlw   0x70
            movwf   PORTB        ;initiate lamptest

            movlw   108
            movwf   counter
            call    delay104us
            decfsz  counter,F     ;reset pulse 10.6ms low
            goto   $-2

men
            bsf     PORTA,1       ;reset wegne-

            movlw   0x32
            call    delayWms      ;400 ms ledtest
            clrf    PORTB        ;leds uit

; ***** READY FOR USE *****

MAIN:
            btfss  PORTB,0       ; character received
            from Xport?
            call   xport_rx      ; haal UDParray uit Xport
            goto  MAIN          ; main loop

xport_rx:
            movlw  0x28          ; Start state0
            movwf  FSR
            tmr0
            clrf
            bcf   INTCON,TOIF
            clrf  maxadres
            btfsc INTCON,TOIF   ;8 ms timeout
            goto  state1
            btfsc PORTB,0
            goto  $-3
            call  Rx_Xport
            movwf INDF
            incf  FSR,F
            movf  FSR,W
            xorlw 0x80
            btfsc STATUS,Z
            bsf   FSR,5          ;jump from 0x80—>0xA0

            incf  maxadres,F
            clrf  TMRO
            goto  $-13

state1:
            bsf   PORTB,4
            bcf   PORTB,5
            bcf   PORTB,6

            movf  ident,W
            ;packet check on

```