

TRABAJO ESPECIAL DE GRADO

DESARROLLO DE SISTEMA MODULAR MULTIPROPÓSITO DE LECTURA/ESCRITURA DE TARJETAS INTELIGENTES EN MODO AUTÓNOMO

Presentado ante la Ilustre
Universidad central de Venezuela
Por el Br. Daniel Alejandro Pérez Cáceres
Para Optar al título de
Ingeniero electricista

Caracas, 2008

TRABAJO ESPECIAL DE GRADO

DESARROLLO DE SISTEMA MODULAR MULTIPROPÓSITO DE LECTURA/ESCRITURA DE TARJETAS INTELIGENTES EN MODO AUTÓNOMO

Prof. Guía: Ing Pedro Pinto.
Tutor Industrial: Ing José David Sánchez

Presentado ante la Ilustre
Universidad central de Venezuela
Por el Br. Daniel Alejandro Pérez Cáceres
Para Optar al título de
Ingeniero electricista

Caracas, 2008

CONSTANCIA DE APROBACIÓN

Caracas, 13 de junio de 2008

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Daniel A. Pérez C., titulado:

“DESARROLLO DE SISTEMA MODULAR MULTIPROPOSITO DE LECTURA/ESCRITURA DE TARJETAS INTELIGENTES EN MODO AUTÓNOMO”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.


Prof. Carolina Regoli
Jurado




Prof. Rafael Rivero
Jurado


Prof. Pedro Pinto
Prof. Guía

DEDICATORIA

**A Maria del Pilar Mesa
Y a Servando Eugenio Álvarez Chaina**

Y todos aquellos otros que, de diversas maneras, hicieron posible este trabajo

RECONOCIMIENTOS Y AGRADECIMIENTOS

En agradecimiento general, deseo nombrar a las siguientes personas como reconocimiento a su contribución significativa al fin de este trabajo:

Primeramente a Dios, por permitirme llegar hasta aquí

A las personas que me dieron su invaluable apoyo durante este largo camino: María del Pilar Mesa, Yamileth Meléndez, Andrea Haeik, Nadia Intriago, y Servando Eugenio Álvarez Chaina, quien además de mucho ánimo, aportó toneladas de consejos y ayuda en los momentos difíciles.

Al valioso apoyo de José David Sánchez (y su invaluable orientación en programación), al Profesor Pedro Pinto, Damarys Laporte, Darwin Arenas y Carlos Borre.

A los UNBÍCOLAS, y en especial a Albert Chávez, Boris Roa y también a Orlando Fajardo, visitante ocasional del UNBI.

Especial mención hago de Ana Rosa Hernández, quien me ayudó en gran parte de la carrera, al principio de la misma, y me ayudo de múltiples maneras.

A mis padres, por el apoyo prestado, a Carolina Régoli, a Mariana Rodríguez y Dorimar Balza.

Y a aquellos otros que por motivos de tiempo y espacio, no incluyo explícitamente, pero que recuerdo siempre y agradezco profundamente todo su aporte

A todos, muchas gracias

Pérez Cáceres, Daniel Alejandro

**DESARROLLO DE SISTEMA MODULAR MULTIPROPOSITO DE
LECTURA/ESCRITURA DE TARJETAS INTELIGENTES EN MODO
AUTÓNOMO**

Profesor Guía: Pedro Pinto. Tutor Industrial: Ing. José David Sánchez. Tesis. Caracas. Universidad Central de Venezuela. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Mención: Electrónica. Institución: E-Smart Consulting, C.A. 2008. 98h + Anexos

Palabras Claves: Sistema Lectura/Escritura de Tarjetas Inteligentes, Circuitos Electrónicos. Módulo Gemcore500.

Resumen: Se planteó la necesidad de elaborar un nuevo tipo de sistema de desarrollo de aplicaciones de lectura e interacción con tarjeta inteligente. Esto surge de una infructuosa búsqueda de equipos que puedan ofrecer las características adecuadas para operaciones no cubiertas por equipos tipo Punto de Venta, realizada por E-Smart Consulting. Debido a ello se requirió la elaboración, en torno a un módulo OEM adquirido, de un equipo electrónico que sea capaz de interaccionar con las tarjetas inteligentes (lectura, escritura y activación de aplicaciones) que ofrezca las virtudes que no proveen los puntos de venta, como simplicidad, flexibilidad de modificación, que sea modular y multipropósito. De esta manera, en base al módulo Gemcore500 MS4, se desarrollaron 2 tarjetas de circuito impreso básicas, más las correspondientes a otros periféricos, y la programación de los dispositivos de control, según los protocolos involucrados, que convierten al desarrollo en un equipo operacional y funcional, según los requerimientos establecidos.

ÍNDICE GENERAL

	Pág.
CONSTANCIA DE APROBACIÓN	
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
RESUMEN.....	vii
ÍNDICE.....	vii
LISTA DE FIGURAS.....	x
LISTA DE TABLAS.....	xi
INTRODUCCIÓN.....	1
CAPÍTULO I	4
1. FORMULACIÓN DEL PROBLEMA	4
1.1. Planteamiento del problema.....	4
1.2. OBJETIVOS.....	6
1.2.1. Objetivo general.....	6
1.2.2. Objetivos específicos.....	6
CAPÍTULO II	
2. CONCEPTOS TEÓRICOS	9
2.1. Definición de tarjeta.....	9
2.2. Definición de tarjeta inteligente.....	13
2.3. Normalización de las tarjetas Inteligentes.....	15
2.3.1. Sección 7816-1: Características físicas.....	15
2.3.2. Sección 7816-2: Dimensiones y localización de los contactos.....	16
2.3.3. Sección 7816-3: Características eléctricas y señales.....	18
2.3.4. Sección 7816-4: Organización interna y comandos.....	24
2.3.5. Sección 7816-5: Registro de proveedores de la Aplicación.....	27
2.3.6. Sección 7816-6: Interoperabilidad en los elementos de datos para el intercambio.....	27
2.3.7. Sección 7816-7: Interoperabilidad en los comandos de la tarjeta (SCQL).....	28
2.3.8. Sección 7816-8: Comandos para operaciones de Seguridad.....	28
2.3.9. Sección 7816-9: Comandos para la gestión de la Tarjeta.....	28
2.3.10. Sección 7816-10: Señales electrónicas para operación sincrónica...	28
2.3.11. Sección 7816-11: Verificación de la identidad personal a través de métodos biométricos.....	28
2.3.12. Sección 7816-12 Tarjetas con contactos. Interfaz eléctrica USB y procedimientos operativos.....	29
2.3.13. Sección 7816-13: Comandos de administración de aplicaciones en un entorno multi-aplicación.....	29

2.3.14. Sección 7816-15: Aplicación de información Criptográfica.....	29
2.4. Aplicaciones típicas.....	30
2.5. Composición básica.....	31
2.6. Seguridad.....	32
CAPITULO III	33
3. DESCRIPCIÓN DEL MÓDULO GEMCORE.....	33
3.1. Muelle de Inserción.....	33
3.2. Interfaz IC100.....	35
3.3. Microcontrolador Central.....	37
3.4. Otros componentes.....	40
3.4.1. Memoria de código externa.....	40
3.4.2. Memoria RAM externa.....	41
3.4.3. Memoria RAM estática.....	41
3.4.4. Dispositivos Selectores.....	42
3.4.5. Reloj de tiempo real.....	42
3.4.6. Gestor de Alimentación.....	43
3.4.7. Conectores de salida.....	43
3.5. Funcionamiento del módulo.....	44
3.6. Protocolo de comandos en Bloque.....	48
3.6.1. Campo NAD.....	49
3.6.2. Campo PCB.....	49
3.6.3. Campo LEN.....	50
3.6.4. Campo DAT.....	50
3.6.5. Campo EDC.....	51
CAPITULO IV	52
4. METODOLOGIA.....	52
CAPITULO V	
5. SISTEMA DESARROLLADO.....	58
5.1.Tarjeta Principal de Interfaz.....	60
5.1.1.Sistema de Gestión de señales.....	61
5.1.2.LCD.....	64
5.1.3.Teclado.....	64
5.1.4.Buzzer.....	65
5.1.5.Presentador 7 Segmentos.....	65
5.2.Programación Módulo GemCore 500-MS4.....	68
5.3.Tarjeta General de Comando.....	71
5.3.1.Control de interfaz.....	72
5.3.2.Módulo de comando General.....	74
5.3.3.Fuente de alimentación.....	76
CAPITULO VI	
6. RESULTADOS.....	79
CONCLUSIONES.....	84
BIBLIOGRAFIA.....	86

GLOSARIO	88
ANEXOS	94

ÍNDICE DE FIGURAS

FIGURAS	Pág.
1. Esquema que ilustra los distintos tipos de tarjeta	12
2. Tipos y medidas de las tarjetas inteligentes	16
3. Fijado de la posición y espacio ocupado por los contactos de la tarjeta	17
4. Circuito que protege a ambos terminales de errores	22
5. Diagrama de bloques de los elementos que conforman al módulo Gemcore	34
6. Esquema del circuito que se usa para activar cada uno de los terminales	36
7. Esquema del procedimiento de lectura escritura.	39
8. Esquema de las señales que salen del módulo Gemcore500	44
9. Esquema de las distintas capas una aplicación con el Gemcore	45
10. Esquema de la distribución de memoria del módulo Gemcore	47
11. Esquema de los elementos internos del GemCore500	59
12. Distribución espacial de los bloques Tarjeta de Interfaz.	62
13. Esquema de disposición y señales de la tarjeta principal de interfaz	64
14. Esquema del funcionamiento del sistema de visualización en 7 segmentos	67
15. Esquema de la interacción entre capas de programación del GemCore500	70
16. Esquema que ilustra el cambio decidido de orientación del proyecto.	72
17. Esquema de la conformación de los elementos del prototipo	74
18. Esquema de la estructura en bloques de la Tarjeta de Comando General	76
19. Circuito que se aplicó para la fuente de alimentación del circuito	77
20. Esquema del prototipo, con las distintas capas que lo componen	82
21. Pantalla del programa desarrollado en Visual Basic	82
24. Fotografías del prototipo culminado	83

ÍNDICE DE TABLAS

TABLAS.	Pág.
Tabla 1. Detalles sobre las zonas conductoras (contactos) de la superficie de la tarjeta inteligente y su significado y función (especificados por la sección 2 de la ISO7816)	17
Tabla 2. Características eléctricas de la señal I/O de intercambio de información entre la tarjeta y el sistema lector, tal como lo especifica la sección 3 de la norma ISO7816	20
Tabla 3. Características eléctricas de la señal VPP como lo define la parte 3 de la norma 7816.....	20
Tabla 4. Características eléctricas de la señal de reloj que recibe la tarjeta, según la sección 3 de la norma 7816.....	21
Tabla 5. Formato estipulado por la sección 3 de la norma ISO7816 para el mensaje de ATR.....	24
Tabla 6. Formato designado por la sección 4 de la norma ISO7816 para el protocolo de mensajes APDU	26
Tabla 7. Formato de la respuesta que se debe dar a un comando APDU, al recibir un mensaje como el mostrado en la tabla 6, tal como lo define la seccion4 de la norma ISO7816	27

ACRÓNIMOS

ABS: Acrilonitrilo Butadieno Estireno.
APDU: En inglés Application Protocol Data Unit.
API: En Inglés Aplicacion Program Interface.
ASN1: En inglés Abstract Syntax Notation 1
GSM: En inglés Global System Mobile.
GPB: En Inglés Gemplus Block Protocol
IDE: En inglés Integrated Drive Electronics
IEC: En Inglés International Electrotechnical Commission
ISO: En Inglés International Standard Organization
OEM: En inglés Original Equipment Manufacturer
PC: En Inglés Personal Computer
PIN: En Inglés Personal Identification Number.
PVC: En inglés PolyVinyl Chloride
POS: En inglés Point Of Sale
RAM: En Inglés Random Access Memory
ROM: En Inglés Read Only Memory.
SAM: En Inglés Security Authentication Module
SIM: En Inglés Subscriber Identity Module

INTRODUCCIÓN

La empresa E-Smart Consulting C.A. es una empresa que se fundó en el año 2002, con el objeto de idear, crear, desarrollar, implementar y comercializar aplicaciones donde se utilicen tarjetas inteligentes para solventar diversas necesidades existentes. Dichas aplicaciones engloban diversos ámbitos, como atención a obligaciones laborales, clubes de lealtad, controles de acceso y otros. La empresa siempre se ha distinguido por la incursión, por medio de la tarjeta inteligente, en nuevas áreas, ofreciendo soluciones a diversos sectores de la vida nacional. Tales desarrollos, dependen de equipos y suministros de diversos fabricantes, los cuales ofrecen diversidad de puntos de venta de variadas características, que interaccionan con la tarjeta para dar vida a la aplicación.

Sin embargo, los puntos de venta, son diseñados y contruidos contando con características idóneas para su operación comercial, lo que a la vez conlleva a un problema, cuando se desea implementar con ellos otras soluciones. Tanto técnicamente, como económicamente, se presentan variados inconvenientes cuando se utiliza un punto de venta para otras aplicaciones que no sean pagos, consumos y otras similares. Tales aplicaciones como por ejemplo control de acceso a instalaciones, cobro de pasaje estudiantil (para sólo mencionar dos) impulsó a la empresa a buscar otro tipo de sistemas de interacción con la tarjeta inteligente. Luego de cierto tiempo, la búsqueda de equipos diferentes quedó desierta, sólo obteniéndose módulos OEM de la casa Gemplus. Estos módulos, pueden hacerse cargo de la parte mas pesada de la interacción con la tarjeta inteligente, pero no constituyen un equipo funcional en sí. Requieren de un sistema que controle y opere para poder utilizarse.

Siendo como es, vital para la empresa expandir las operaciones a nuevos campos aún no explotados, y contando con la falta de equipos idóneos para cubrir estas necesidades, E-Smart Consulting decidió involucrarse en el desarrollo en si de tal equipo, percatándose luego de las ventajas que proveería tal proyecto. En tal sentido, se decidió iniciar la realización de un sistema de desarrollo general, que sea capaz de interaccionar con la tarjeta inteligente. Este sistema debería ser Autónomo

(es decir, poder funcionar sin estar adherido a un computador), modular (constituyendo su estructura de diversos módulos que puedan ser incluidos o no al ensamblar), económico (que su construcción sea competitiva con la adquisición de un punto de venta), adaptable y multipropósito.

Adicionalmente, dicha decisión implica que la empresa es capaz de proveerse a sí misma de su equipamiento, conoce y domina los dispositivos que comercializa, puede mantenerlos sin involucrar a terceros, lo que le da independencia tecnológica y ayuda al fomento de la producción nacional. Tal provisoria visión, alentó aún más a la consecución de la meta propuesta. Cabe destacar, que aunque es un sistema de desarrollo basado en la interacción total con la tarjeta inteligente, en el argot de la industria se denomina simplemente “lector”, a pesar de que este equipo es capaz de leer, escribir y acceder a aplicaciones que se ejecutan dentro de la propia tarjeta. En comunión con dicha costumbre, de aquí en adelante a los equipos se les denominará simplemente “Lector”.

El desarrollo se llevó a cabo, primeramente estudiando la normativa internacional vigente, para asegurar la estandarización (y profundizar más en el entorno, conceptos y conocimientos relevantes) y luego en los manuales y documentación suministrados por el fabricante Gemplus, dado que el sistema a desarrollar se haría alrededor de su módulo OEM. Esta fase de investigación de la teoría pertinente relacionada con el subsiguiente diseño derivó en el planteamiento de lineamientos generales de diseño, y perspectivas adicionales.

En gran parte del trabajo, la inspiración para las soluciones se basó en características obtenidas de esta investigación, con lo cual se realizó un avance significativo en el planteamiento del esbozo de la solución. Posteriormente se inició el desarrollo electrónico como tal, que derivaría en los circuitos y tarjetas de circuito impreso a construir. Aunque durante el desarrollo se hallaron ciertos problemas y de hecho, en cierto momento fue necesario reconducir la estrategia de diseño, el planteamiento original no varió y se logró, paso a paso, ir cumpliendo con los objetivos, tomando como referencia la importancia de la simplicidad, economía y modularidad.

Luego del desarrollo de circuitos electrónicos y tarjetas de circuito impreso, se procedió a desarrollar la programación que gobernaría el funcionamiento de aquellos sistemas inteligentes que se encontraban presentes en el diseño y que en esencia sería la base para crear la plataforma del sistema de desarrollo, con la idea de que al estar a punto el sistema, desarrolladores incluyeran sus programas y algoritmos para dar vida a la aplicación particular que se han propuesto implementar. De esta manera, el prototipo, constituye en sí, una plataforma sobre la cual, otros especialistas crearían aplicaciones particulares, aprovechando las bondades descritas, que no sería posible con otros equipos adquiribles.

Sin embargo, se desarrolló una aplicación demostrativa, que no es funcional en el sentido completo de un funcionamiento operativo, pero que permite ver en acción (genérica) a todos los elementos que conforman el fin de este trabajo. Una vez llevado todo a término, se obtiene un prototipo operacional, que cumple con las metas planteadas y que de manera satisfactoria, puede ser la base de muchas nuevas aplicaciones, incluso algunas aún no imaginadas.

CAPÍTULO I

1. PRELIMINARES

1.1. Planteamiento del problema

E-Smart Consulting, es una empresa creada para desarrollar soluciones basadas en el mundo inteligente, lo que se refiere a aplicaciones en torno a tarjetas inteligentes. Tales aplicaciones constituyen soluciones a diversas debilidades, necesidades y problemas que presentan empresas actualmente. E-Smart Consulting fue constituida en el año 2002, ante la creciente necesidad en el mercado, de alternativas que permitan la solución de tales requerimientos empresariales.

Aunque siempre surgen nuevas posibilidades no imaginadas al inicio, la mayoría de las soluciones implementadas hasta el momento son de ámbito laboral. A manera de ejemplo, se puede enumerar algunas de ellas, como lo son, control de acceso a instalaciones de las empresas, registro y auditoria de horarios laborados, beneficios de alimentación para trabajadores. Sin embargo, también se ha incursionado en el área comercial y servicios, implementándose numerosos clubes de fidelidad, por medio de una tarjeta inteligente, en un comercio o cadena de ellos. Esta aplicación permite al comerciante (o prestador de servicios) ofrecer ventajas a los consumidores, si ellos eligen con frecuencia adquirir sus productos y/o servicios, gracias a la tarjeta inteligente que se les otorga. De la misma manera, el comerciante, con estas ventajas o premios, consigue atraer mayor clientela. Luego, han surgido numerosas aplicaciones que no se habían previsto, como control y registros de servicios relacionados con la adquisición de vehículos, tarjetas de licencia para activar software vendido, evitando el cracking (suplantado o falsificación) de contraseñas, y otras.

E-Smart Consulting siempre se encuentra a la búsqueda de nuevos campos donde puedan crear una manera de insertar aplicaciones basadas en tarjetas inteligentes. Asimismo, E-Smart Consulting posee proveedores confiables de lectores de tarjeta inteligente (además de las tarjetas en sí), aunque la mayoría de los equipos

recibidos cuentan con las características para ser incluidos dentro de la categoría de “Puntos de Venta” (POS por Point Of Sales). A estos equipos lectores, se les da la condición de POS porque tienen alta seguridad, se encuentran diseñados para ser imposible su modificación física (por seguridad), cuentan con un gran poder de procesamiento, aunque son físicamente delicados y se averían con facilidad si se les somete a una operación diferente a la planeada. Por supuesto, son costosos, lo cual regularmente no es problema para operaciones comerciales, que producen ganancias que permiten cubrir dicho costo. Si exceptuamos las aplicaciones de beneficios de alimentación y clubes de lealtad, buena parte de las otras aplicaciones, no requieren tanta seguridad, ni tanto poder de procesamiento ni rapidez y debería ser posible agregar o suprimir componentes de más a los equipos para adecuarlos a las aplicaciones.

En ciertos casos, se ha resuelto adaptando puntos de venta de menor seguridad a dichas aplicaciones, pero los resultados, aunque han sido adecuados técnicamente, no son económicamente muy satisfactorios (por ejemplo en el caso de control de acceso). En resumen, se requiere de un nuevo tipo de lector, listo para ser adaptado a las aplicaciones presentes, y otras que se presentarán en el futuro cercano. Por esto, se hizo evidente que había que hallar un equipamiento distinto de lectura de tarjetas inteligentes, más adecuado y apegado a las necesidades y flexibilidades de esas otras operaciones.

Al acudir a los proveedores requiriendo una plataforma de lectura de tarjeta inteligente más sencilla, simple y económica, para utilizarlos en estas otras aplicaciones, se obtuvo una serie de módulos OEM, es decir, tarjetas de circuito impreso especiales para la lectura, pero que requieren de desarrollo adicional para poderse integrar en un lector como tal. Ante tal evento, E-Smart Consulting aceptó el reto y pensó en la idoneidad de esta posibilidad, pues permite a la empresa crear un lector a la medida de las necesidades, y abre la posibilidad de atender nuevos campos, y atender nuevas expectativas del mercado. Siendo así, este desarrollo permitiría posteriormente a un equipo multidisciplinario, crear software y añadir o eliminar fácilmente componentes para cada nueva posibilidad de atender necesidades en el

mercado. Por ello, se planteó la necesidad de generar todo lo necesario en circuitos y programas que permitiesen crear fácilmente y económicamente, a partir de un trazado básico, ejemplares del mismo lector con una mayor o menor complejidad y otras variadas características. Todo esto alrededor de los módulos que ya se poseían y estudiando su composición y funcionamiento. Esto implicó, que el desarrollo a efectuar debía ser un equipo lector de tarjetas inteligentes modular, para que se pudiesen modificar rápidamente, conectando o desconectando submódulos (algunos de los cuales pueden ser diseñados más adelante); debería ser multipropósito, para que con solo modificarle la programación interna, y agregando o quitando módulos, se pueda cambiar para adaptar al propósito; que pudiese operar en modo autónomo, sin requerir la presencia de algún equipo computacional que dirigiese su funcionamiento. Cabe destacar, que al ser propietaria e inventora de su diseño, la empresa está en capacidad de proporcionarse a sí misma la capacidad de reparación, actualización, recuperación y modificación del diseño. Esto hace más robusta la capacidad de prestación de servicios.

1.2. OBJETIVOS

1.2.1. Objetivo general

Desarrollar un sistema de lectura y gestión de tarjetas inteligentes (Smart Card) multipropósito, modular, programable y adaptable a las distintas posibles aplicaciones de los clientes de E-Smart Consulting C.A, basándose en un subsistema comercial OEM¹.

1.2.2. Objetivos específicos

- Estudiar la norma ISO 7816, así como otros parámetros y estándares que permitan comprender el subsistema OEM comercial (módulo base) a fondo para desarrollar el sistema íntegro.

¹ Manufactura de Equipo Original, dispositivos o partes expendidas al mayor para ensamblar equipos de forma personalizada que otras empresas presentan con su propio nombre.

- Estudiar las características del módulo básico OEM de lectura de tarjetas para posteriormente desarrollar el programa residente que interacciones tanto con las tarjetas insertadas como con los módulos periféricos a desarrollar.
- Desarrollar un circuito que sea capaz de administrar las señales producidas por el módulo base OEM (el lector de tarjetas en sí) y gestione la comunicación con los distintos módulos periféricos (módulo de interfaz).
- Proveer la capacidad de acoplar y desacoplar fácilmente los distintos módulos con el módulo base por medio del módulo de interfaz (o sistema gestor de señales).
- Desarrollar un módulo estándar de comunicación visual al usuario que muestre mensajes para varias aplicaciones.
- Desarrollar un módulo de inserción de datos/comandos por medio de teclas o botones. Desarrollar un módulo asimismo con ausencia de teclas, que pueda sustituir al módulo con teclas en caso de que no se requiera en alguna aplicación particular la presencia de teclas o botones.
- Desarrollar un módulo de alimentación o fuente (para el sistema entero) que cuente con sencillos sistemas de detección de la alimentación, entre posibilidades de alimentación de la red o alimentación autónoma.
- Desarrollar un módulo de aviso auditivo, que produzca señales audibles para eventos y alertas al usuario. Asimismo, prever la ausencia de esta posibilidad para ciertas aplicaciones por medio de un módulo sustitutivo nulo.

- Desarrollar un módulo que prevea la posibilidad de entrada de otros dispositivos de identificación de usuario y la adición de otros periféricos aún no especificados
- Desarrollar un módulo de manejo de potencia, que permita realizar acciones como: apagado o encendido de sistemas de cierto nivel de potencia según los eventos adquiridos por el resto del sistema (ejemplo, apertura de puertas).
- Desarrollar la programación del dispositivo completo, de modo que soporte las distintas posibilidades de ausencia o presencia de periféricos al arranque

CAPÍTULO II

2. CONCEPTOS TEÓRICOS

Como se puede apreciar, dada la necesidad ya expuesta de realizar un desarrollo alrededor del módulo adquirido por E-Smart Consulting, es relevante realizar una breve síntesis de los conceptos envueltos en este proyecto. Básicamente, los elementos previos que se encuentran vinculados con el desarrollo son la tarjeta inteligente y el Módulo Lector adquirido para solventar el problema planteado. Por ello, en las subsiguientes páginas se presentará alguna información relevante relacionada con ambos tópicos.

2.1. Definición de tarjeta

Llevado a los términos más básicos, una tarjeta es una pieza de material resistente, de forma rectangular con ciertas medidas especificadas, la cual tiene una función de relevancia para el portador. En tal sentido, actualmente casi todas las tarjetas, son de material plástico, y cumplen con unas medidas fijadas en la especificación ISO-7816 con bordes redondeados para evitar constituirse en objetos punzantes. Estas tarjetas, poseen muchas funciones, pero regularmente contienen variados tipos de información recuperable por medio de diversas tecnologías, las cuales varían según las necesidades, abarcando desde la más primitiva hasta la más moderna.

En tal sentido, las tarjetas han venido evolucionando y aunque en el momento actual todas las formas están vigentes, su aplicación varía según la tecnología presente en cada tipo de tarjeta. La utilidad principal de una tarjeta es contener una información en sí misma, lo que da vida a alguna aplicación de diversa índoles, aunque frecuentemente se encuentran englobadas entre seguridad, identificación, o finanzas. Dada esta utilidad o función, se podría clasificar a las tarjetas en general, atendiendo a la tecnología de almacenaje de tal información, de la siguiente manera:

- *Sin modificación.* Este tipo de tarjetas son las que no poseen ninguna modificación física del plástico, es decir, no presentan la incrustación de ningún elemento ajeno al plástico del que están hechas. Los datos se pueden encontrar contenidos en ellas de tres maneras distintas.
 - a. Impresión simple: Consiste en imágenes, cifras y letras impresas sobre su superficie. Es muy común en carnets de identificación y otros similares. La información se recupera observando los datos impresos a simple vista y la interpretación corre a cargo de la persona que requiere conocer algún dato del portador.
 - b. Troqueladas: Idéntica a la anterior, sólo que algunos datos relevantes se encuentran grabados en relieve, es decir, sobresalen de la superficie lisa de la tarjeta. La información se registra por medio de la presión mecánica sobre un trozo de papel que está encima de un papel carbón, y éste último dispuesto sobre la tarjeta, para lograr un calcado de la silueta de estos datos importantes. Se usó en tarjetas de crédito por algunos años, como única manera de validación.
 - c. Código de barra. Se basa en la impresión en la superficie de un código de líneas verticales, cuyo grosor y distancia entre sí es significativo para reconocer patrones asociados a cifras y letras basadas en estándares. Una máquina proyecta un haz de luz sobre el código y con esto lo adquiere o recupera. La interpretación la ejecuta un equipo computacional que realiza comparaciones y validaciones según la aplicación.
- *Con incrustación de elementos.* Este tipo de tarjetas posee en su interior o exterior un elemento distinto al plástico con que se fabricó, el cual contiene los datos grabados en el interior de esta incrustación, por métodos electromagnéticos. Existen tres tipos básicos de estas tarjetas
 - a. Banda magnética. Es una banda que se incrusta en la parte superior del anverso de la tarjeta, compuesta por partículas ferromagnéticas embebidas en una matriz de resina (generalmente epoxi) y que

almacenan cierta cantidad de información mediante una codificación determinada que polariza dichas partículas. Esta banda contiene datos almacenados electro-magnéticamente de manera similar a la que se utilizaba en las viejas cintas de audio o video (Cassette). Generalmente, la banda contiene de dos a tres pistas con los datos duplicados. Al deslizarla por una ranura contentiva de un lector (muy similar también al proceso de reproducción de cassettes) se recuperan los datos y se interpretan y validan. Es la tecnología actual de tarjetas de crédito en muchos países.

- b. Tarjeta inteligente de contacto. Esta tarjeta tiene incrustado en su interior un sistema microprocesador y/o memoria minúsculo. Se conecta con el exterior por medio de una serie de contactos metálicos que se aprecian en la superficie. Los datos (y posiblemente la ejecución de aplicaciones sencillas en su interior) se recuperan por medio del acoplamiento de los contactos de la tarjeta con los contactos del lector. Generalmente esto se realiza por medio de la inserción de la tarjeta en una ranura del lector especialmente diseñada para tal fin. Ambos sistemas interactúan entre sí (el sistema lector y el sistema interno de la tarjeta) al tener contacto. Muy utilizada en telefonía y en aplicaciones bancarias en muchos países.
- c. Tarjeta sin contacto (o de Proximidad). Esta tarjeta también tiene incrustado en su interior un sistema microprocesador y memoria minúsculo. Adicionalmente tiene un sistema de comunicación por radiofrecuencia. Al aproximarse la tarjeta a un lector, las emisiones de radiofrecuencia de éste energizan y proporcionan la portadora de comunicación a la tarjeta. La tarjeta establece una comunicación inalámbrica con el lector. Por ese medio se recupera la información presente en la tarjeta.

Esta clasificación de las tarjetas en general no se encuentra estrictamente delimitada. Muchas tarjetas tienen una combinación en sí misma de varias de estas características antes descritas. Es posible apreciar las características de cada tipo de tarjeta en la figura 1. De hecho, es posible que tengan todas simultáneamente. El desarrollo de la presente tesis se va concentrar alrededor de la tarjeta inteligente de contacto y su lectura.

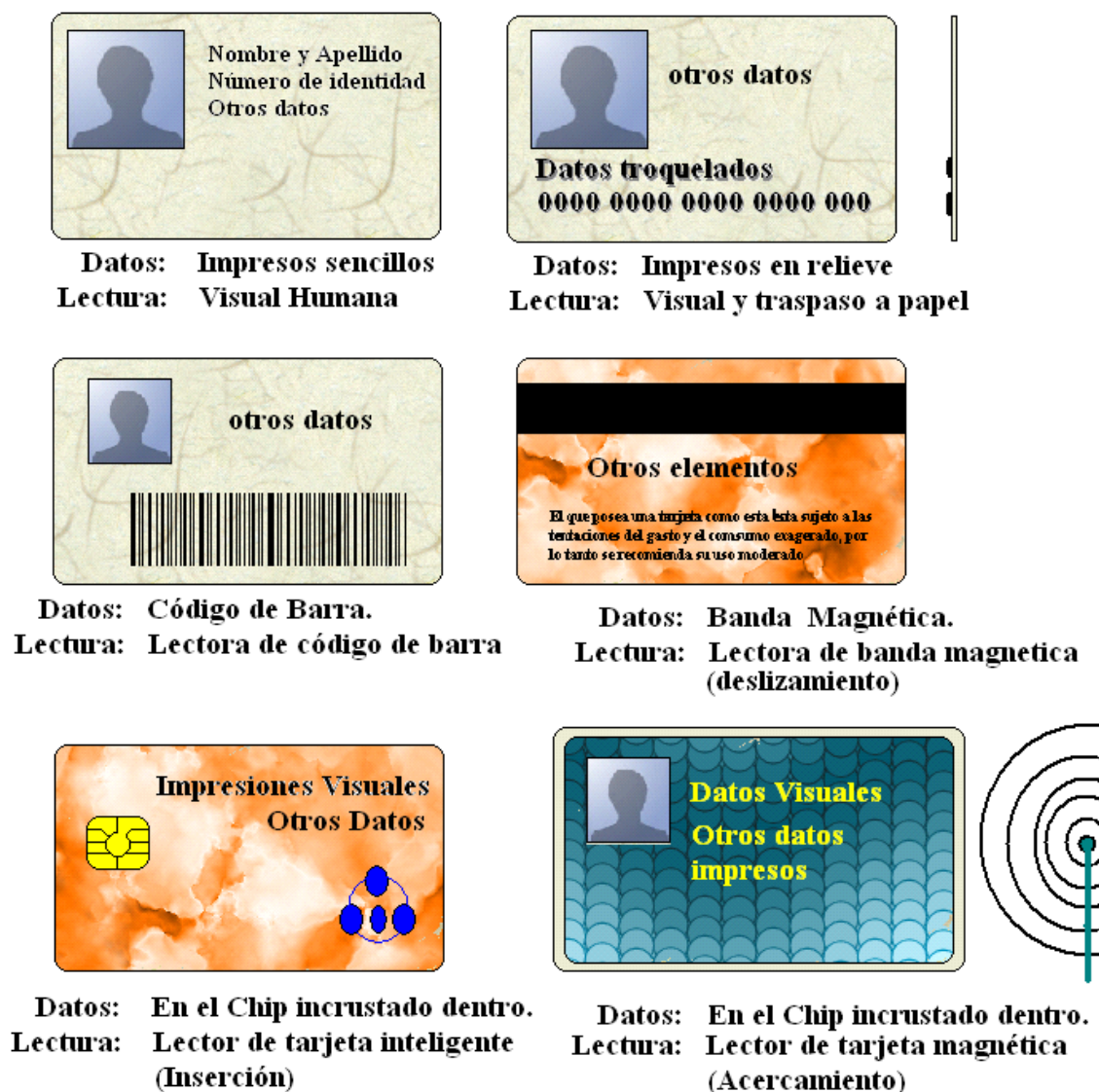


Figura 1. Esquema que ilustra los distintos tipos de tarjeta que actualmente se hallan en circulación, según sus características físicas.

2.2. Definición de tarjeta inteligente

Una tarjeta inteligente (smart card), o tarjeta con circuito integrado (TCI), es cualquier tarjeta de dimensiones adecuadas para portarla en un bolsillo, con circuitos integrados incluidos que permiten la ejecución de cierta lógica programada. La percepción estándar de una tarjeta inteligente es una tarjeta microprocesadora (es decir, con un microprocesador adentro) de las dimensiones de una tarjeta de crédito con varias propiedades especiales (ej. un procesador criptográfico seguro, sistema de archivos seguro, características legibles por humanos) y es capaz de proveer servicios de seguridad (ej. confidencialidad de la información en la memoria). Las tarjetas no contienen baterías; por lo que la energía es suministrada por los lectores de tarjetas.

Aunque existe un diverso rango de aplicaciones, suele decirse que, a grandes rasgos, hay dos tipos principales de tarjetas inteligentes. Sin embargo, las tarjetas inteligentes, a pesar que conservan múltiples rasgos invariables entre sí, pueden tener algunas diferencias que dan lugar a varias clasificaciones. Según las capacidades de su sistema electrónico interno (que suele denominarse chip), las tarjetas más habituales son:

- *Tarjetas de Memoria:* Tarjetas que contienen mayormente componentes de memoria no volátil y cierta lógica de seguridad y acceso. Suelen usarse únicamente como un contenedor de información pero no albergar aplicaciones ejecutables, dado que no tienen una elevada capacidad de procesamiento. Suelen utilizarse en aplicaciones de identificación y control de acceso sin altos requisitos de seguridad, o como medio de transportar información entre un lector autónomo y un lector esclavo de un PC.
- *Microprocesadora:* Tarjetas con una estructura análoga a la de un ordenador (procesador, memoria volátil, memoria persistente). Éstas albergan alguna información y aplicaciones ejecutables, siendo la mayor capacidad la de ejecutar aplicaciones, lo que sugiere una capacidad de procesamiento elevada. Suelen usarse para identificación y pago con monederos electrónicos.

- *Criptográficas:* Tarjetas microprocesadas avanzadas en las que hay módulos especialmente creados para la ejecución de algoritmos usados en cifrados y firmas digitales. En estas tarjetas se puede almacenar de forma segura un certificado digital (y su clave privada) y firmar documentos o autenticarse con la tarjeta sin que el certificado salga de la tarjeta (sin que se instale en el almacén de certificados de un navegador Web, por ejemplo) ya que es el procesador de la propia tarjeta el que realiza la firma. Suelen usarse en aplicaciones de alta seguridad.

Como se puede notar, estas diferencias son sutiles, pues los tres tipos de tarjeta pueden contener algo de las tres características (memoria, procesamiento y Criptografía). La diferencia estriba en la robustez y potencia que alcanza cada característica en cada tipo de tarjeta inteligente. También se suelen diferenciar las tarjetas según la estructura de su sistema operativo. Atendiendo a esta característica, existen:

- *Tarjetas de memoria.* Tarjetas que únicamente son un contenedor de archivos pero que no albergan aplicaciones ejecutables. Disponen de un sistema operativo limitado con una serie de comandos básicos de lectura y escritura de las distintas secciones de memoria y pueden tener capacidades de seguridad para proteger el acceso a determinadas zonas de memoria.
- *Basadas en sistemas de archivos, aplicaciones y comandos.* Estas tarjetas disponen del equivalente a un sistema de archivos compatible con el estándar ISO 7816 parte 4 (del cual se hablará más adelante) y un sistema operativo en el que se incrustan una o más aplicaciones (durante el proceso de fabricación) que exponen una serie de comandos que se pueden invocar a través de APIs de programación.
- *Java Cards.* Una Java Card es una tarjeta capaz de ejecutar mini-aplicaciones Java. En este tipo de tarjetas el sistema operativo es una pequeña máquina virtual Java (JVM) y en ellas se pueden cargar dinámicamente aplicaciones desarrolladas específicamente para este entorno.

2.3. Normalización de las tarjetas Inteligentes

Como se puede deducir, el correcto funcionamiento de este vasto mundo de las tarjetas inteligentes, se basa en la posibilidad de que los lectores de diversos fabricantes, sean compatibles con tarjetas de diversos fabricantes. Para este cometido, es necesaria una regulación internacional, y para eso existen diversas normas que aseguran la estandarización.

Como ya se mencionó, las tarjetas inteligentes se encuentran regidas por la norma ISO 7816, que regula los aspectos más relevantes de ellas. Debido a esto, se considera adecuado proporcionar una síntesis de los contenidos de esta norma. La norma ISO 7816 es un estándar internacional relacionado con las tarjetas electrónicas, en especial las tarjetas inteligentes, gestionado conjuntamente por la Organización Internacional de Normalización (ISO) y Comisión Electrotécnica Internacional (IEC). Se trata de una extensión de la ISO 7810. Está editada por el Comité Técnico Conjunto (JTC) 1 / Subcomité (SC) 17, Tarjetas de identificación personal.

Dicha norma se divide en varias partes o secciones, cada una de las cuales se especializan en aspectos básicos que permiten la interoperatividad entre fabricantes. Hasta el momento, se conocen 15 secciones, redactadas entre 1987 y 2004, las cuales son descritas a continuación.

2.3.1. Sección 7816-1: Características físicas

Esta sección de la norma fue creada en 1987, actualizada en 1998, modificada en 2003. En esta sección se definen las características físicas de la tarjeta en sí, y aunque se refieren a la tarjeta inteligente, son comunes a todos los otros tipos de tarjetas mencionadas, por lo menos en cuanto al tipo ID 1. Se definen así mismo, tres clases o tipos de tamaño de tarjeta inteligente, adecuadas para distintas aplicaciones, que son denominadas ID 000, ID 00 e ID 1.

- **Tarjetas ID 000:** es el tamaño de las tarjetas SIM usadas para teléfonos móviles GSM. También acostumbran a tener este formato las tarjetas SAM

(*Security Access Module*) utilizadas para la autenticación criptográfica mutua de tarjeta y Terminal. Las medidas se aprecian en la figura 2.

- **Tarjetas ID 00:** un tamaño intermedio poco utilizado comercialmente.
- **Tarjetas ID 1:** el más habitual, tamaño típico de las tarjetas en general.

Cabe destacar que esta sección se refiere únicamente al tamaño, grosor y demás dimensiones (así como forma) del plástico que constituye la tarjeta.

2.3.2. Sección 7816-2: Dimensiones y localización de los contactos.

Sección creada en 1988, actualizado en 1999, modificado en 2004. Esta sección se basa esencialmente en la descripción de los contactos eléctricos visibles en la superficie de la tarjeta. Describe las dimensiones, posición y función de cada uno de

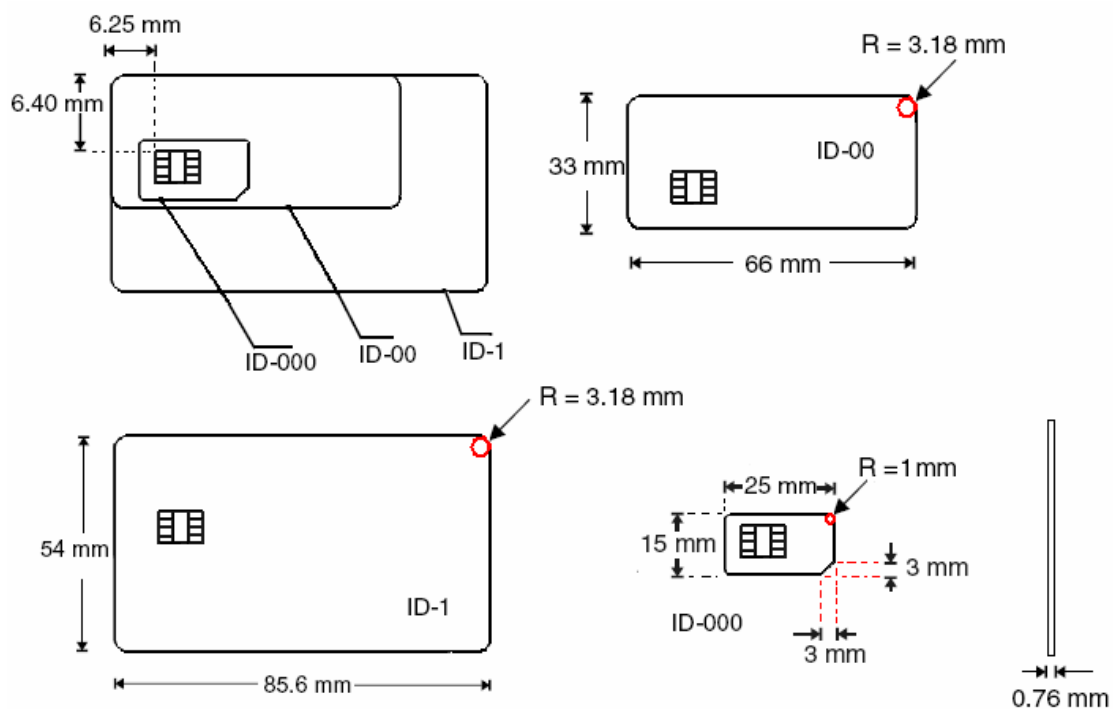


Figura 2. Tipos y medidas de las tarjetas inteligentes, tal como son definidas por la sección 1 de la norma ISO 7816.

los contactos. A modo de resumen, se presenta a continuación en la tabla 1 donde se precisa el nombre y función de cada uno de los contactos de la tarjeta. Igualmente, en

la figura 3 se aprecia una ilustración de los contactos, su disposición y distancias fijadas por esta sección e la norma.

Tabla 1. Detalles sobre las zonas conductoras (contactos) de la superficie de la tarjeta inteligente y su significado y función (especificados por la sección 2 de la ISO7816)

Contacto	Designación	Uso
C1	VCC	Alimentación: Contacto a través de la cual la tarjeta es provista de la energía necesaria para alimentar todo el sistema interno del chip (microprocesador, memoria, etc.)
C2	RST	Línea de Reset: Contacto a través de la cual se le indica al microprocesador de la tarjeta que debe iniciar una secuencia de reset.
C3	CLK	Señal de Reloj: contacto a través de la cual es provista de la tarjeta con una señal de reloj al microprocesador interno. Esta señal controla la velocidad de operación del chip, y establece una estructura común de comunicación entre el lector y la tarjeta, sincronizándolas adecuadamente.
C4	RFU	Reservadas para Uso Futuro (sin uso actualmente)
C5	GND	Tierra: contacto con que se provee una referencia común entre el equipo lector y la tarjeta.
C6	VPP	Línea de Voltaje de Programación: contacto con que se proveía un voltaje de programación, para los EPROMS de la primera generación de tarjetas. Actualmente o no se requiere, o lo produce un sistema interno. (sin uso actualmente)
C7	I/O	Entrada/Salida: Contacto con que se provee la comunicación tipo serial half-duplex entre el lector y la tarjeta inteligente
C8	RFU	Reservadas para Uso Futuro (sin uso actualmente)

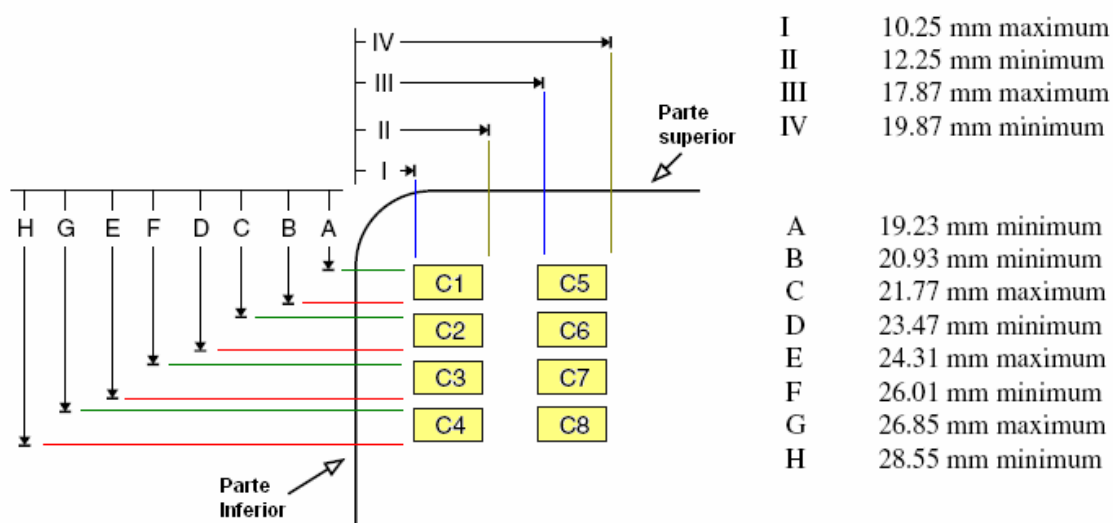


Figura 3. Fijado de la posición y espacio ocupado por los contactos de la tarjeta según la parte 2 de la ISO7816. No son parámetros fijos, pues el tamaño y forma de los contactos, varía según el fabricante y modelo de la tarjeta. Sólo se debe asegurar un tamaño

2.3.3. Sección 7816-3: Características eléctricas y señales

Sección creada en 1989, actualizada en 1997, modificado en 2002 y en 2006. En esta sección se especifican las características eléctricas de las señales que se producen para lograr un correcto intercambio de información, cuando se establece un adecuado contacto entre el lector y las áreas conductoras externas de la tarjeta. Dado que esta sección se considera importante, se proporcionará mayor cantidad de información al respecto, en comparación con otras secciones. Estas señales se corresponden con cada uno de los contactos descritos en la sección 2 de la Norma.

Adicionalmente, se establecen elementos protocolo de comunicación e intercambio de data. De especial importancia es la respuesta al Reset (ATR), que se expondrá más adelante. A continuación se proporciona una descripción de las características más relevantes de cada señal regulada por la presente Norma.

2.3.3.1. Señal VCC.

Esta línea constituye la alimentación del circuito interno de la tarjeta. La tensión de alimentación para las tarjetas inteligentes fue inicialmente de 5 volts, con una tolerancia máxima de $\pm 10\%$. Esta tensión, que es la misma que la utilizada para los circuitos TTL convencionales, es el valor comercial estándar para todas las tarjetas inteligentes y todas las aplicaciones. Al igual que con otros componentes, la tendencia a la miniaturización, ha obligado a reducir notablemente el rango de voltaje de funcionamiento de los equipo. Sobre todo en los teléfonos móviles o celulares, por la demanda para reducir su peso.

Esto provocó el cambio de baterías de 6V hacia las de 3V, y dado que todos los demás componentes para los teléfonos móviles están disponibles en tecnología 3V, se hizo necesario adecuar la tensión de las tarjetas inteligentes ID000 a 3V. Por ello la norma fue revisada y extendida la alimentación de 3 a 5V con 10%. Sin embargo, al final, se prefirió dividir las tarjetas en clases. La norma ISO / IEC 7816-3 y su modificación definen tres categorías para caracterizar las gamas de tensión de las tarjetas inteligentes. La Clase A abarca el rango de tensión de $5\text{ V} \pm 10\%$, la Clase B abarca la gama de $3\text{ V} \pm 10\%$ y Clase C abarca la gama de $1,8\text{ V} \pm 10\%$.

Las tres clases pueden ser utilizadas individualmente o en cualquier combinación deseada. Por ejemplo, si una tarjeta inteligente reúne los requisitos para la clase A y clase B, puede ser utilizada con cualquiera de las dos alimentaciones (5V y 3V) de tensión. La norma especifica un procedimiento particular para la selección de la tensión de alimentación, que en esencia equivale a probar los voltajes para cada una de las tres clases a la vez. Tan pronto como el Terminal puede recibir un ATR, se analiza para ver si la tarjeta inteligente prefiere una clase particular. Si es así, el Terminal inicia una nueva secuencia de activación utilizando la clase deseada.

Si el ATR no incluye ninguna información sobre la gama de tensión, se utilizará el voltaje con el que el primer ATR pudo ser recibido. Por otro lado, la primera versión de la norma especifica una corriente máxima de 200mA para 5V y de 5 MHz de reloj, pero incluso entonces que era excesivo. Desde ese momento, los valores se han reducido significativamente y dependen de las distintas clases de tensión de alimentación. El consumo de un microcontrolador es directamente proporcional a la frecuencia de reloj aplicada y la tensión de alimentación. También dependen de la temperatura del microcontrolador. La versión actual de la Norma ISO / IEC 7816-3 especifica un máximo actual de 60mA para tensión de clase A (5 V) en un máximo de frecuencia de reloj de 5 MHz y una temperatura ambiente máxima de 50°C.

2.3.3.2. Señal I/O

Este señal se utiliza como entrada (modalidad de recepción) o salida (de transmisión) para el intercambio de datos. Existen dos estados posibles para I/O:

- Marca o nivel alto (Estado Z), si la tarjeta y el dispositivo de interfaz están en modalidad de recepción o si el estado es impuesto por el transmisor.
- Espacio o nivel bajo (Estado A), si este estado es impuesto por el transmisor.

Cuando los dos extremos de la línea están en modalidad de recepción, la línea se mantendrá en estado Z. Cuando los dos extremos se encuentran en modo de transmisión no acoplado, el estado lógico de la línea puede ser indeterminado. Durante las operaciones, la interfaz del dispositivo y la tarjeta no pueden estar ambos en modo de transmisión. Sin embargo, para proteger eléctricamente a los terminales

(tarjeta y lector) de desacoples de comunicación, se coloca un circuito como el mostrado en la figura 4. En la tabla 2 se aprecian las principales características eléctricas de la transmisión de datos entre la tarjeta y el lector.

Tabla 2. Características eléctricas de la señal I/O de intercambio de información entre la tarjeta y el sistema lector, tal como lo especifica la sección 3 de la norma ISO7816

Símbolo	Condición	Mínimo	Máximo	Unidad
V _{pp}	Estado Inactivo	0.95*V _{cc}	1.05*V _{cc}	V
I _{pp}	(Programación no activa)		20	mA
V _{pp}	Estado Activo	0.975*P	1.025*P	V
I _{pp}	Programación en progreso		I	mA
La tarjeta provee la interfaz con los valores de P e I (donde, por defecto, P=5, I=50)				

2.3.3.3. Señal VPP

Esta señal constituye el suministro de tensión necesaria para programar o borrar el interior de memoria no volátil que poseen las tarjetas. Existen dos posibles estados para VPP: estado de reposo y estado activo, tal como se definen en la tabla 3. El estado inactivo será mantenido por el dispositivo de interfaz a menos que surja la necesidad de cambiar el estado a activo.

Tabla 3. Características eléctricas de la señal VPP como lo define la parte 3 de la norma 7816

Parámetro		Condiciones	Mínimo	Máximo	Unidad
Voltaje de entrada en nivel lógico alto (Vih)	Uno u	Iih max = +/- 500uA	2,00	Vcc	Volts
	otro (1)	Iih max = +/- 50uA	0,7*Vcc	Vcc (3)	Volts
Voltaje de entrada en nivel lógico alto (Vil)		Iil max = 1mA	0	0,8	Volts
Voltaje de salida en nivel lógico alto (Voh) (2)	Uno u	Iol max = +/- 100uA	2,4	Vcc	Volts
	otro	Iol max = +/- 20uA	3,8	Vcc	Volts
Voltaje de salida en nivel lógico alto (Vol)		Iol max = 1mA	0	0,4	Volts
tr, tf	Cin = 30pF; Cout = 30pF			1	us
(1) Para el circuito de interfaz, es conveniente usar ambas condiciones.					
(2) Se asume que se usa una resistencia de pull up en el dispositivo de interfaz (20K Ω)					
(3) el voltaje en I/O se mantendrá entre 0.3V y VCC+0.3V.					

Esto es lo que dice la especificación original, en el presente, regularmente no se usa, y el estado es típicamente inactivo. Esto es debido a que en la primera tecnología, se utilizaban EPROM, en lugar de las EEPROM que se utilizan actualmente. Así, luego de 1990, muchas tarjetas tenían circuitos “bombas de carga” que llevaban a cabo esta función. Esta señal, no puede usarse para nada más, por lo cual, se mantiene el contacto, sin uso, sólo con el fin de no contradecir la norma.

2.3.3.4. Señal CLK

Esta es la señal de sincronización del sistema tarjeta lector, es la base de sincronización, aunque no es necesariamente la velocidad del microprocesador interno, pues algunas tarjetas tienen circuitos modificadores (multiplicadores o divisores) de frecuencia. La frecuencia real, emitida por el dispositivo de interfaz en CLK se designa, ya sea por la frecuencia inicial (f_i) durante la respuesta a reset (ATR), o la frecuencia posterior (f_s) durante la transmisión posterior. El ciclo de trabajo para operaciones asíncronas será de entre 45% y el 55% del período del funcionamiento estable. Se pondrá cuidado al cambiar las frecuencias (de f_i a f_s) para garantizar que ningún pulso es más corto que el 45% del plazo más breve. En la tabla 4 se aprecia las características eléctricas de esta señal

Tabla 4. Características eléctricas de la señal de reloj que recibe la tarjeta, según la sección 3 de la norma 7816

Símbolo	Condiciones		Mínimo	Máximo	Unidad
Vih	Una u otra (1)	Iih max = +/- 200uA	2.4	VCC (2)	Volt
		Iih max = +/- 20uA	0.7*VCC	VCC (2)	Volt
		Iih max = +/- 10uA	VCC-0.7	VCC (2)	Volt
Vil	Iil max = +/-200 uA		0 (2)	0.5	Volt
tr, tf	Cin = 30pF		9% del periodo	Con un max:0.5	µs
(1) Para el dispositivo de interfaz, tomar en cuenta las tres condiciones posibles					
(2) El voltaie en CLK debe permanecer entre los 0.3 y VCC+0.3V.					

2.3.3.5. Procedimientos

Además de las señales que se deben proveer a las tarjetas, existen ciertas configuraciones de procedimientos que se encuentran estipulados por la Norma, y específicamente, por la presente sección de la Norma. De estas se hace una síntesis a continuación.

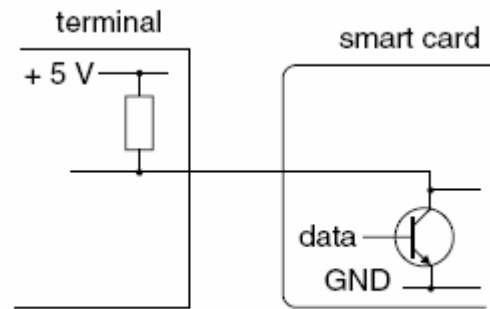


Figura 4. Circuito que protege a ambos terminales (o extremos) de la comunicación de daños por errores de sincronización.

2.3.3.5.1. Procedimientos de activación y desactivación.

Para prevenir errores en la comunicación y posibles daños en los componentes involucrados, la norma estipula una secuencia de eventos previos a considerar iniciada la comunicación interfaz de lectura-tarjeta. Esto debe iniciar con el contacto físico entre las minúsculas clavijas que suele tener el lector y las correspondientes secciones de superficie conductoras de la superficie del chip de la tarjeta. Al ocurrir esto se debe observar la siguiente serie de eventos:

- Estabilización de la referencia o tierra.
- Estabilización de la señal de alimentación (VCC).
- Luego de esto (un tiempo después) se empieza a proveer la señal pulsante de reloj. De aparecer esta señal primero que la alimentación, el microcontrolador podría sufrir daños irreversibles. Para este momento, aparte de una frecuencia estable de CLK, RST debe estar en nivel bajo, I/O en modo de recepción y VPP en estado inactivo.
- Al llegar este momento, el microprocesador interno ya se encuentra en condiciones de operar, por lo que se aplica un reset y se inicia la operación, con el flanco de subida de esta señal.
- Luego del reset, la tarjeta debe enviar una respuesta al lector, que proporciona cierta información concerniente a la subsiguiente operación. Éste es el ATR.

- Luego se inicia formalmente el intercambio de información.

Cabe destacar que todo lo antes descrito es un proceso del cual se asegura la interfaz lectora de tarjetas sea llevado a cabo, paso por paso, para iniciar la comunicación. De no efectuarse algún paso correctamente, se reintenta, y se considera un error, si no se efectúa correctamente luego de varios intentos. El proceso es similar para la desconexión (claro está, en orden de secuencia inverso). De cualquier manera, el sistema está diseñado para que en virtud de un retiro repentino de la tarjeta, ésta pierda tanto la comunicación como la alimentación al primer desplazamiento, apagándola automáticamente.

2.3.3.5.2. Respuesta al Reset (ATR)

El ATR (por *Answer To Reset*, en inglés) es sumamente importante para la comunicación subsiguiente. Es básicamente una cadena de hasta 33 Bytes. Raramente un ATR utiliza la longitud máxima admitida, regularmente son sólo unos pocos bytes. El ATR debe ser recibido antes de 40.000 ciclos de reloj, y luego de 400 ciclos de reloj, lo cual comprueba que se ha establecido la comunicación de manera satisfactoria. Regularmente el envío del ATR y demás mensajes subsiguientes es asincrónico, pues cada componente solicita el canal y transmite en lo que le es posible. El ATR también contiene diversos elementos que indican el modo de comunicación ulterior. La composición a grandes rasgos se muestra en la tabla 5

Tabla 5. Formato estipulado por la sección 3 de la norma ISO7816 para el mensaje de ATR

Byte	Nombre	Contenido
TS	Carácter inicial	0x3B para convención directa 0x3F para convención inversa
TO	Carácter de Formato	Nibble alto: Caracteres de Interfaz presentes Nibble Bajo: Número de Caracteres Históricos presentes
TA1, TB1, TC1	Caracteres de Interfaz	Definen los parámetros del protocolo (sin embargo, existen configuraciones por defecto, pueden ser omitidos)
TD1		
...		
T1....Tk	Caracteres Históricos	Caracteres adicionales que generalmente se omiten o contienen en ASCII un nombre o numero de versión
TCK	Carácter de comprobación	Carácter que se calcula a partir de operaciones entre todos los bytes enviados y sirve de verificación de si no hubo error de envío.
CARACTERES DE INTERFAZ		
TA1: define el nuevo tiempo de bit por medio de 2 valores (D velocidad de transmisión y FI factor de conversión de la frecuencia de reloj), los cuales por medio de la ecuación $ETU=(D/FI)*(1/fr)$ dan el tiempo de cada bit (ETU), por defecto $(D/FI)=372$ (fr es la frecuencia de reloj). TB1: define el valor de tensión que se debe enviar por VPP, por medio de un factor con respecto a la alimentación. Dado que VPP cayó en desuso, igual con TB1. TC1: define un tiempo de guarda (o extensión en tiempo del bit de stop). TD1: contiene (idéntico a TO, si existen más caracteres de interfaz como TA2, TB2, TC2 y TD2). El nibble alto dice el protocolo de comunicación T=0 o T=1; Los siguientes caracteres de interfaz indican la clase de alimentación, el estado preferido como bit de stop (TA2), valor exacto para VPP (TB2), máximo tiempo de espera (TC2) y otros El protocolo T=0 es half duplex, orientado a bytes El protocolo T=1 es half duplex orientado a bloques		

2.3.4. Sección 7816-4: Organización interna y comandos

Sección creada en 1995, actualizada en 2005. Esta sección especifica la organización interna de la memoria de la tarjeta con respecto al sistema operativo, así como la base de los comandos que entienden las tarjetas. La organización interna de la tarjeta inteligente se basa en sistemas de archivos o archivos organizados de forma jerárquica como en la mayoría de los sistemas operativos modernos. Los archivos se nombran por un identificador de archivo de dos bytes.

Las tarjetas inteligentes contienen 3 tipos principales de archivos:

- **Archivo Principal (MF):** es la raíz de la jerarquía. Seleccionado por defecto al inicio de la comunicación. Se identifica por *3F 00* y contiene la

información y la lista de los archivos contenidos dentro de ella. Tiene cierta analogía con un directorio raíz.

- **Archivo Dedicado (DF):** son análogos a directorios en las tarjetas inteligentes, subdividen secciones de las tarjetas para contener los archivos descritos a continuación (EFs), aunque pueden contener en su interior igualmente otros DF (tal como los directorios). Normalmente contienen los datos relativos a una aplicación.
- **Archivo Elemental (EF):** en los que se almacenan los datos realmente. Pueden encontrarse en cualquier posición, directamente del directorio raíz (MF) o dentro de algún DF. Pueden ser de cuatro tipos:

- o *Archivo transparente* o binario: sin estructura interna (a veces se les denomina estructura amorfa), son almacenes de bytes (o bloques de bytes) con un tamaño máximo. Los datos se pueden direccionar con un indicando la diferencia que hay con el primer dato (es decir, en inglés un offset).
- o *Archivo lineal de registros de longitud variable:* archivos con una estructura prefijada consistente en una lista de registros individualmente identificables donde cada uno puede tener una longitud variable (el tamaño se fija en el momento en el que se crea). Los registros son direccionados según el orden de su creación y su número (o índice) no se puede modificar posteriormente. Son semejantes a las estructuras (struct) de lenguaje C.
- o *Archivo lineal de registros de longitud fija:* semejantes a los anteriores, sólo que consisten en una lista de registros individualmente identificables con una longitud fija e igual para cada registro. Los registros son direccionados según el orden de su creación y su número no se puede modificar ya posteriormente.
- o *Archivo cíclico de registros:* archivos con una estructura prefijada consistente en un 'anillo' (lista enlazada circular) de registros individualmente identificables cada uno con un ancho fijo. Los

registros son direccionados en orden inverso a su creación/modificación. Es decir el puntero señala el último registro escrito. Cuando se alcanza el máximo, la siguiente escritura ocurre en el primer registro.

Esta sección de la norma también fija la estructura interna de los archivos, como su nombre y organización, y manera de accederlos crearlos y modificarlos. Sin embargo, esta sección 7816-4 también regula la parte básica de los comandos recibidos por la tarjeta, es decir su formato. Como ya se ha establecido, una vez recibido el ATR, la interfaz de lectura se encuentra capacitada para iniciar el tráfico de comandos que el sistema operativo de la tarjeta debe ejecutar. Dado la relación Maestro-Esclavo que existe en esta comunicación, los comandos siempre son enviados por la interfaz lectora. Por medio de ellos se le solicita a la tarjeta realizar ciertas acciones. La data se envía dentro de contenedores denominados APDU (del inglés Application Protocol Data Unit). Dicho contenedor está conformado por 4 bytes obligatorios de cabecera y tres bytes más opcionales. Éstos se muestran en la tabla 6.

Tabla 6. Formato designado por la sección 4 de la norma ISO7816 para el protocolo de mensajes APDU

Cabecera		Cuerpo	
CLA INS P1 P2		[campo Lc] [Campo de Datos] [Campo Le]	
Significado de los campos			
Nombre	Bytes	Nombre	Descripción
CLA	1	Clase	Clase o tipo de instrucción (mensaje seguro, autenticado, propietario, etc.)
INS	1	Instrucción	Instrucción en sí (Leer, escribir, seleccionar, etc.)
P1	1	Parámetro1	Primer parámetro de la instrucción.
P2	1	Parámetro2	Segundo Parámetro de la instrucción.
Campo Lc	1 a 3	Longitud	Numero de bytes presentes en el campo de datos
Campo datos	Lc	Datos	Cadena de datos que vienen con el comando
Campo Le	1 a 3	Longitud	Número de bytes esperados como respuesta

Esta sección de la norma también define la respuesta que debe recibirse de la tarjeta, la cual se muestra en la tabla 7

Tabla 7. Formato de la respuesta que se debe dar a un comando APDU, al recibir un mensaje como el mostrado en la tabla 6, tal como lo define la seccion4 de la norma ISO7816

Código	Nombre	Longitud	Descripción
Campo de Datos	Datos	Le	Cadena de bytes recibidas en el campo de datos de respuesta
SW1	byte de Estatus 1	1	Estado de procesamiento del comando (respuesta de la tarjeta = correcto, error, etc)
SW2	byte de Estatus 2	1	Estado de procesamiento del comando

Esta sección contiene cierta cantidad de otros temas, que en función de la brevedad, serán omitidos, por ejemplo cómo utilizar una solicitud de identificación para determinar la presencia de y/o realizar la recuperación de una aplicación en una tarjeta. Sin embargo, cabe destacar que esta sección es bastante amplia en cuanto a la cantidad de elementos que regula

2.3.5. Sección 7816-5: Registro de proveedores de la aplicación

Creado en 1995, actualizado en 2004. Esta sección muestra cómo conceder unicidad (o exclusividad) de los identificadores de aplicación por medio del registro internacional de una parte de este identificador, y define el procedimiento de registro, Las autoridades a cargo, y la disponibilidad de registro. De esta manera, se puede identificar la presencia de una aplicación en una tarjeta, dado que sus identificadores son exclusivos.

2.3.6. Sección 7816-6: Interoperabilidad en los elementos de datos para el intercambio

Creado en 1996, actualizado en 2004. Esta sección especifica los Elementos de Datos utilizados para la interoperatividad industrial, de modo que haya compatibilidad. Se basa en las tarjetas con circuito integrado, ya sea de contacto o proximidad. Se regula el identificador, el nombre, la descripción, el formato, la codificación y el diseño de cada elemento de datos y define los medios de recuperación de éstos de la tarjeta.

2.3.7. Sección 7816-7: Interoperabilidad en los comandos de la tarjeta (SCQL)

Creado en 1999. Define comandos para interoperatividad y especifica los métodos estándar para mantener y consultar bases de datos manera similar al SQL, sólo que limitada a selección simple y operaciones de proyecto. También proveer definiciones de formato

2.3.8. Sección 7816-8: Comandos para operaciones de seguridad

Creado en 1995, actualizado en 2004. Especifica los comandos de las tarjetas (ya sea con o sin contacto) que pueden utilizarse para operaciones criptográficas. Estos comandos son complementarios, y sobre la base de los comandos enumerados en la norma ISO/IEC 7816-4. La elección y condiciones de utilización de los mecanismos criptográficos pueden afectar a la posibilidad de exportar la tarjeta (debido a las restricciones impuestas en algunos países). La evaluación de la idoneidad de los algoritmos y protocolos está fuera del alcance de la norma ISO/IEC 7816-8.

2.3.9. Sección 7816-9: Comandos para la gestión de la tarjeta

Creado en 1995, actualizado en 2004. Especifica los comandos de las tarjetas (con contactos y sin contactos) para la gestión de archivos, por ejemplo la creación y borrado de archivos. Estos comandos abarcan todo el ciclo de vida de la tarjeta y, por consiguiente, algunos comandos pueden ser usados antes de que la tarjeta haya sido expedida a su titular o después de que ésta haya caducado.

2.3.10. Sección 7816-10: Señales electrónicas para operación sincrónica

Creado en 1999. Especifica las señales si la operación de la tarjeta se va a efectuar sincrónicamente

2.3.11. Sección 7816-11: Verificación de la identidad personal a través de métodos biométricos

Creado en el año 2004. Especifica el uso de los comandos y de los datos relacionados con la verificación de la identidad de una persona a través de los métodos biométricos en las tarjetas de circuito integrado. Los comandos utilizados se definen en la norma ISO/IEC 7816-4. Los datos se definen parcialmente en esta norma y en parte importados de la norma ISO/IEC 19785-1.

2.3.12. Sección 7816-12 Tarjetas con contactos. Interfaz eléctrica USB y procedimientos operativos

Creado en el año 2005. Desde su resumen, especifica las condiciones de funcionamiento de una tarjeta de circuito integrado a través de una interfaz USB. La ISO/IEC 7816-12:2005 proporciona dos protocolos para controlar las transferencias. Se trata de soportar el protocolo T=0 (versión A) o utilizar la transferencia de APDU (versión B).

2.3.13. Sección 7816-13: Comandos de administración de aplicaciones en un entorno multi-aplicación.

Creado 2006, Especifica los comandos para el manejo de aplicaciones en un ambiente multi-aplicación. Similar a la sección 7816-9.

2.3.14. Sección 7816-15: Aplicación de información criptográfica

Creado en el año 2004. Especifica una aplicación para tarjetas que contiene información sobre la funcionalidad criptográfica. Por otra parte, define una sintaxis común (en ASN.1) y el formato de codificación de la información y los mecanismos para compartir esta información cuando sea apropiado. Es compatible con las siguientes capacidades:

- Almacenamiento de múltiples claves de criptográficas información en una tarjeta;
- Uso de la información criptográfica;
- Recuperación de la información criptográfica;

- Referencias cruzadas de la información criptográfica con denominaciones definidos en la norma ISO/IEC 7816, cuando proceda;
- Diferentes mecanismos de autenticación; y
- Múltiples algoritmos criptográficos.

Todas las características definidas por la principal normativa que rige el diseño y funcionamiento de sistemas alrededor de las tarjetas inteligentes, provee una cierta cantidad de utilidades generales, entre las que se incluyen algunas ya descritas y aparecen otras aún no consideradas

2.4. Aplicaciones típicas

La mayoría de las aplicaciones fundamentales de las tarjetas inteligentes pueden concentrarse en tres renglones, que son

- Identificación del titular (portador) de la misma.
- Pago electrónico de bienes o servicios mediante dinero virtual.
- Almacenamiento seguro de información asociada al titular.

Algunas aplicaciones no mencionadas aún, de las tarjetas inteligentes incluyen su uso como tarjeta de crédito, SIM para telefonía móvil, tarjetas de autorización para televisión por pago, identificación de alta seguridad y como tarjetas de pago del transporte público. Las tarjetas inteligentes son muy utilizadas como un monedero electrónico. Estas aplicaciones disponen normalmente de un archivo protegido que almacena un contador de saldo y comandos especiales para decrementar e incrementar el saldo. Con esta aplicación, a la tarjeta inteligente se le puede incluir un monto de dinero que puede ser utilizado en parquímetros, máquinas expendedoras u otros elementos de cobro automático. Protocolos criptográficos protegen el intercambio de dinero entre la tarjeta inteligente y la máquina receptora.

Cuando las tarjetas son criptográficas las posibilidades de identificación y autenticación se multiplican ya que se pueden almacenar de forma segura certificados digitales o características biométricas en archivos protegidos dentro de la tarjeta de

modo que estos elementos privados nunca salgan de la tarjeta y las operaciones de autenticación se realicen a través del propio procesador criptográfico de la tarjeta.

Una aplicación novedosa es el almacenamiento de un certificado digital de forma segura dentro de la tarjeta y poder firmar con él documentos electrónicos sin que en ningún momento el certificado (y más concretamente su clave privada) salgan del lugar confinados. Con estas aplicaciones se abre todo un abanico de posibilidades en el campo de la Administración electrónica. Una aplicación adicional son las Tarjetas, sistema de identificación de pacientes y almacenamiento de los principales datos de la historia clínica de los mismos en tarjetas inteligentes para agilizar la atención.

Una vez conocida la amplia gama de aplicaciones existentes hasta el momento, se procederá a realizar una síntesis de la composición interna del chip de la tarjeta.

2.5. Composición básica

Internamente, el chip de una tarjeta inteligente microprocesadora se compone de:

- CPU: el procesador de la tarjeta; suelen ser de 8 bits, a 5 MHz. Y un valor de alimentación dependiente de la clase. Algunas soportan varios niveles. Pueden tener opcionalmente módulos de procesamiento exclusivos para operaciones criptográficas.
- ROM: memoria interna (normalmente entre 12 y 30 KB) en la que se ubica el sistema operativo de la tarjeta, las rutinas del protocolo de comunicaciones y los algoritmos de seguridad de alto nivel por software. Esta memoria, como su nombre indica, no se puede reescribir y se personaliza durante el proceso de fabricación.
- EEPROM: memoria de almacenamiento en el que está grabado el sistema de archivos, los datos usados por las aplicaciones, claves de seguridad y las propias aplicaciones que se ejecutan en la tarjeta. El acceso a esta memoria está protegido a distintos niveles (según la naturaleza del dato en cuestión) por el sistema operativo de la tarjeta.

- RAM: memoria volátil de trabajo del procesador. Para uso del sistema operativo.

Una de las propiedades más importantes de las tarjetas inteligentes, es la seguridad que es posible aplicar, residente en la propia tarjeta. Antes de pasar a otros tópicos, se presentará una breve síntesis.

2.6. Seguridad

La seguridad es una de las propiedades más importantes de las tarjetas inteligentes y se aplica a múltiples niveles y con distintos mecanismos. Cada archivo lleva asociadas unas condiciones de acceso y deben ser satisfechas antes de ejecutar un comando sobre ese archivo.

En el momento de pre-personalización de la tarjeta (paso final de la fabricación) se puede indicar qué mecanismos de seguridad se aplican a los archivos. Normalmente se definirán:

- *Archivos de acceso libre*: Seleccionables sin necesidad de contraseñas
- *Archivos protegidos por claves*: Pueden definirse varias claves con distintos propósitos. Normalmente se definen claves para proteger la escritura de algunos archivos y claves específicas para los comandos de consumo y carga de las aplicaciones de monedero electrónico. De ese modo la aplicación que intente ejecutar comandos sobre archivos protegidos tendrá que negociar previamente con la tarjeta la clave oportuna.
- *Archivos protegidos por PIN*: El PIN es un número secreto que va almacenado en un archivo protegido y que es solicitado al usuario para acceder a este tipo de archivos protegidos. Es diferente al anterior, porque es de procedencia externa.

Frecuentemente, los procesos con claves o contraseñas se realiza habitualmente apoyándose en un Módulo SAM, alojada en un lector interno propio dentro de la carcasa del lector principal que contiene bases criptográficas Operando de este modo se está autenticando todos los elementos involucrados en cada operación.

CAPITULO III

3. DESCRIPCIÓN DEL MÓDULO GEMCORE

El módulo base de interfaz, fue diseñado basándose en la configuración de la familia de módulos OEM de Gemplus². Específicamente, se dispone de un módulo Gemcore500 MS-4. Luego de una investigación en las hojas de datos del fabricante y de un análisis del circuito del módulo OEM, se realizó un análisis exhaustivo de su estructura y se logró obtener un bosquejo del plano de la circuitería, lo que facilita enormemente la comprensión del mismo. El módulo OEM Gemcore500 MS4 tiene cierta cantidad de componentes siendo tres de ellos los fundamentales:

- El muelle de inserción de tarjeta inteligente estándar.
- El microcontrolador Phillips 83C524.
- El Integrado de control IC100.
- En la figura 5 se aprecia un esquema de la distribución general del módulo GemCore500-MS4

3.1. Muelle de Inserción.

En tal sentido, el muelle es un dispositivo que consta de una ranura para inserción de la tarjeta inteligente, la cual da acceso a un receptáculo o recámara que recubre una sección de la tarjeta inteligente (aproximadamente un 40% de su área total) durante la conexión con el lector. Este receptáculo asegura que la tarjeta se encuentre en la correcta posición, haciendo coincidir áreas conductoras de la tarjeta con los respectivos contactos del lector, y así garantizar la correcta conexión eléctrica.

² Gemplus, compañía Francesa especializada en el mercado de las tarjetas inteligentes y lectores de tarjetas. En el año 2007 se fusionó con otra compañía Axalto, creando Gemalto, quien a partir de este momento lleva las operaciones anteriores de ambas empresas

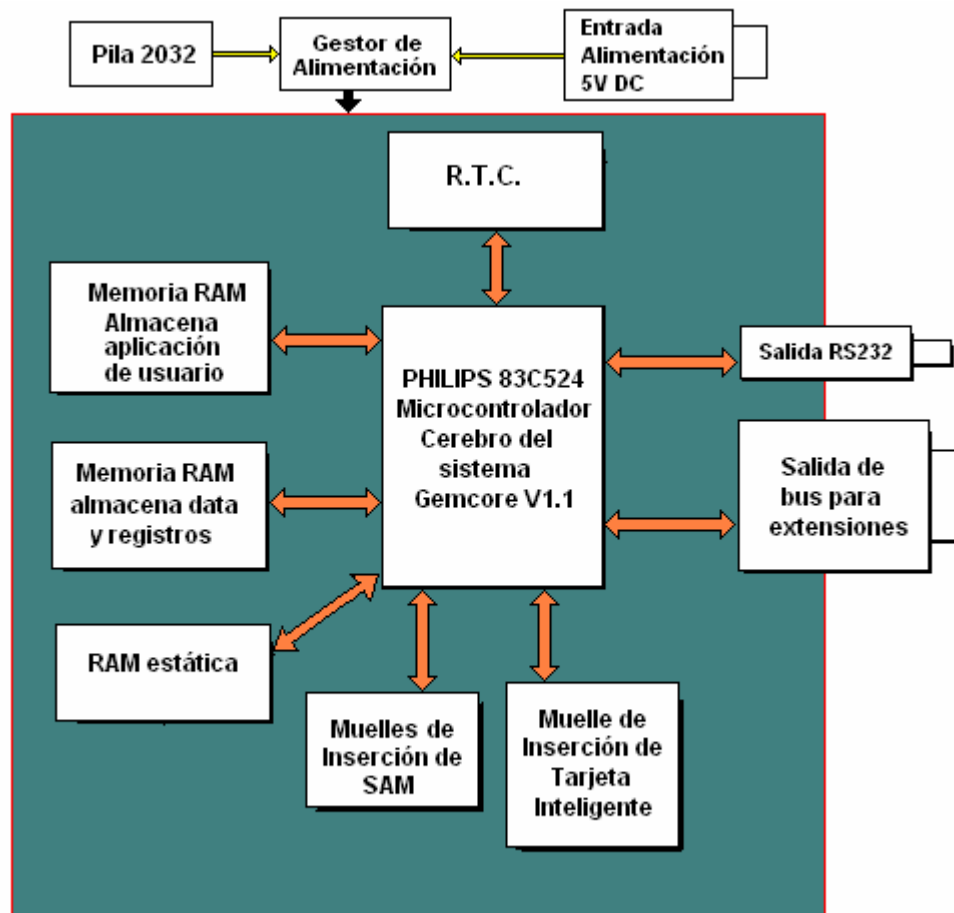


Figura 5. Diagrama de bloques de los elementos que conforman al módulo OEM Gemcore

El muelle está provisto de un sistema mecánico tal que, al alcanzar la tarjeta el tope de inserción dentro del receptáculo, genera la extensión de los contactos metálicos para lograr la conexión. Este procedimiento es más seguro y garantiza durabilidad, ya que protege a los delicados contactos de los movimientos de inserción o extracción de la tarjeta dentro del muelle, ya que en estos eventos, el conjunto de contactos se repliega y separa de la tarjeta en movimiento. En otros sistemas lectores, donde los contactos se encuentran fijos, se ha observado casos de ruptura o deformación de estos contactos, por tracción mecánica en el momento de inserción o extracción.

Una vez insertada la tarjeta inteligente, en la posición adecuada, los contactos del lector establecen una conexión directa entre los contactos del chip de la tarjeta inteligente y los pines de salida del muelle. A partir del momento en que la tarjeta está en posición, también se pulsa un micro interruptor que indica al sistema que maneja la comunicación, la presencia de una tarjeta.

Luego de ese momento, el sistema se encarga del resto de las gestiones. Lo primero que se establece con la tarjeta es una referencia o tierra. Luego, se activa la alimentación, y posteriormente, la señal de sincronización o reloj. Es después de estos pasos iniciales, que se establece la comunicación, de tipo serial por una única línea de señal. El lenguaje de esta comunicación está basado en comandos codificados en series de bytes con significado estándar, de los que se hablará más adelante.

Adicionalmente, sistema se encuentra proveído de una salida adicional, por medio de un peine de pines, para conectar un muelle lector externo independiente, lo que alarga la vida útil del módulo completo, y amplía su versatilidad.

3.2. Interfaz IC100

La gestión de tráfico de información entre la tarjeta inteligente y el resto del sistema corre a cargo del IC100, dispositivo especialmente diseñado como interprete entre el Gemcore y el conector de recepción de tarjeta inteligente. Este Circuito Integrado funge de interfaz entre el muelle y el microcontrolador. La comunicación con la tarjeta se realiza a través de una comunicación serial half duplex (por el Canal I/O) sincronizada con reloj (del canal CLK). Existe un dispositivo IC100 por cada elemento de tarjeta inteligente a leer. En el módulo OEM hay posibilidad de leer una tarjeta inteligente, y hasta cuatro minitarjetas SAM a la vez, lo que implica la presencia de 5 integrados IC100 en el circuito general del módulo OEM.

Todos los integrados realizan el tráfico de datos de un bus común de 8 señales proveídas por el microcontrolador central. Estas líneas de señal se corresponden con la programación del código ejecutable residente en el microcontrolador reservado para responder a eventos relacionados con la tarjeta inteligente. Un esquema de la

interconexión de cada IC100, obtenida a partir del análisis del módulo OEM, se presenta en la figura 6.

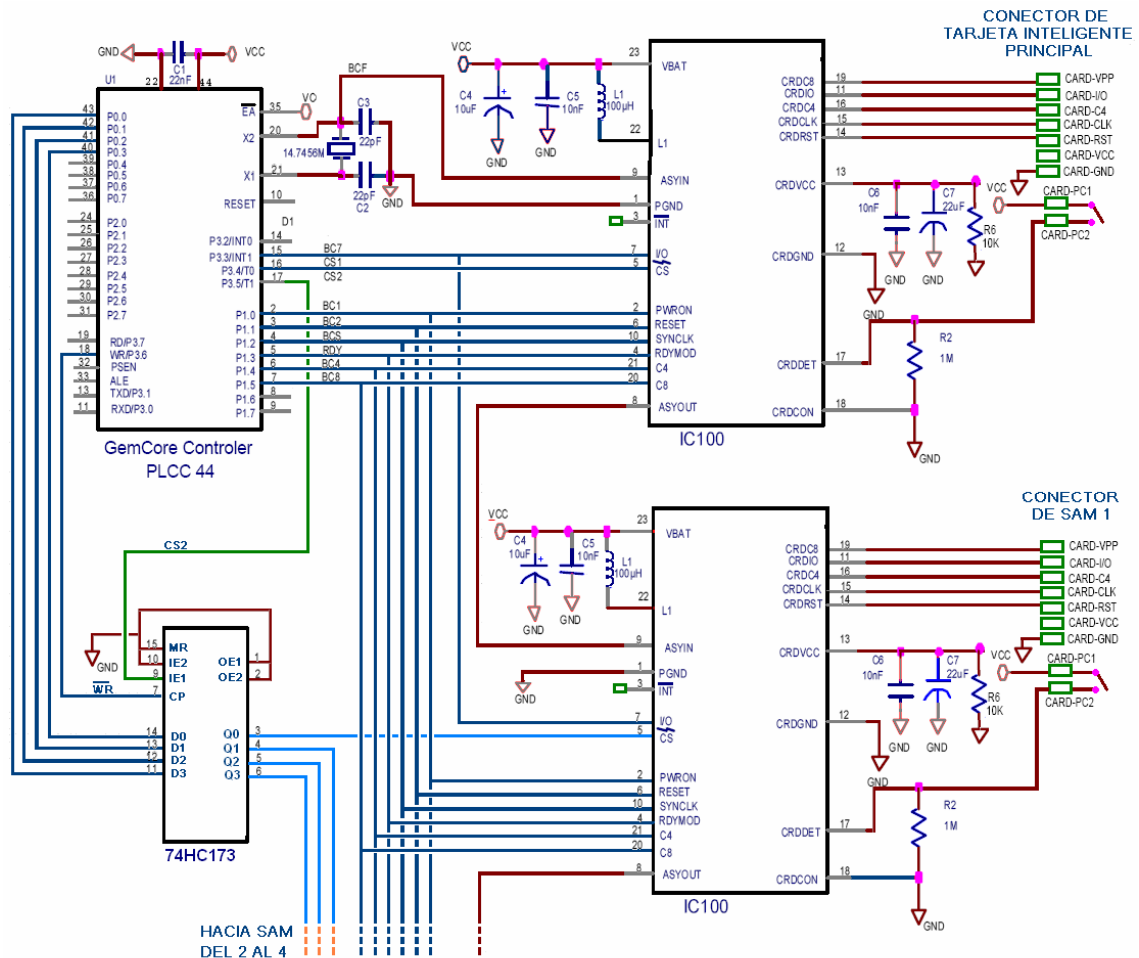


Figura 6. Esquema del circuito que se usa para activar cada uno de los terminales lectores de tarjeta inteligente por medio del integrado IC100

En tal figura se aprecia que la manera de seleccionar cual de los cinco posibles elementos de tarjeta inteligente se desea acceder es por medio de las señales CS1 y CS2, ambas provenientes del microcontrolador. La primera (CS1) está reservada exclusivamente para el muelle de inserción de tarjeta inteligente y su respectivo IC100. La otra señal (CS2) activa un 74HC173, el cual simultáneamente, recibe una dirección del nibble bajo de bus de datos/direcciones del microcontrolador (además de la señal WR), y de esta manera se selecciona entre los cuatro IC100 (correspondiente a SAMs), el que se desea acceder.

Otras características importantes que provee esta interfaz son: capacidad de monitorear la alimentación de la tarjeta y elegir entre 3 y 5V; detección de corto circuito; Control para asegurar la secuencia de encendido y apagado según la norma; apagado automático al retirar la tarjeta y protección contra fenómenos de estática.

3.3. Microcontrolador Central.

El elemento central del módulo OEM es el microcontrolador central. Este microcontrolador (Philips 83C524) se constituye en el “cerebro” del módulo. Es una versión del microcontrolador 8051 de Intel realizada por Philips, con algunas mejoras. El módulo GemCore500 MS4, viene programado con un código residente, que se carga inmediatamente se enciende o se aplica un RESET, y realiza tareas de arranque del sistema. Este código residente (firmware en inglés) tiene en sí las funciones de *bootloader* (algo así como *cargador de arranque* en inglés) y *bootstrap*. Para ser precisos, estos dos términos en inglés designan al proceso completo en que se carga un entorno de programación de bajo nivel, que está íntimamente relacionado con los periféricos presentes, detectados al arrancar. Este entorno de programación se mantiene activo para servir de interfaz entre dichos periféricos y un entorno distinto de más alto nivel.

En tal sentido, al culminar todo este proceso de arranque, entra en estado de espera de comandos externos o, de existir, carga un programa de más alto nivel que se ubica en memoria externa y lo ejecuta. Esto lo hace, salvando las diferencias, recibir el apelativo de Sistema Operativo. En todo caso, los comandos que recibe (del programa de más alto nivel, o de procedencia externa vía serial) sólo indican la tarea a ejecutar y los parámetros que delimitan las posibilidades de la tarea, y esa capa de software de bajo nivel, interpreta los parámetros y los convierte en acciones de control a circuitos electrónicos. Para ello, las secuencias de programación de este firmware deben ser coherentes con el trazado del circuito. En el módulo OEM en cuestión, esta capa de programa de bajo nivel, se la denomina GemCore (homónima con el nombre del módulo en general).

El fabricante desarrolló un protocolo de comandos reconocibles por el firmware GemCore, y con ellos es posible, en base a otras herramientas (que se encuentran incluidas en el paquete de adquisición del módulo), tanto controlar el módulo remotamente por vía serial, como descargar en memoria externa de código, un programa que, en base a librerías, genera los mismos comandos, pero internamente. Más adelante se especificará acerca de esta configuración de comandos.

Siendo la arquitectura del microcontrolador central del Módulo OEM, la de un 8051, dispone de un conjunto de señales cuyas funciones son bastante familiares. Las más relevantes son los puertos P0 y P2 (de 8 bits cada uno) los cuales en conjunto constituyen el bus compartido de datos/direcciones. Al indicarse en la ejecución del programa la lectura o escritura en periféricos externos, en un primer ciclo, la parte baja (menos significativa) del Bus de direcciones se corresponde con el puerto P0. En un siguiente ciclo, el puerto P0 toma la función de bus de datos.

La parte alta del bus de direcciones se corresponde con el puerto P2. Esto es relevante, pues en la arquitectura del 8051, todo periférico o componente dirigido por él, se accede como si fuese una memoria externa a través del bus compartido de datos/direcciones. Por ello, se aprovechan ciertas señales del bus para codificar por medio de selectores, la correcta activación de uno de los componentes entre todos los que tienen acceso al bus compartido. En tal sentido, el bit más significativo de una dirección colocada en el bus, es el pin de puerto P2.7 y el bit menos significativo es P0.0. Luego de colocar la dirección (6 pulsos de reloj después), se mantiene la parte alta de la dirección (el puerto P2) y el puerto P0 se convierte en datos. Según el comando interno, se establece la dirección de los datos (entrantes o salientes) y se encienden las señales de WR (escritura, es decir, salientes) o RD (lectura, es decir, entrantes). Igualmente se activa una señal específicamente diseñada para manejar un dispositivo Latch, que preserve en el lado del periférico, la parte baja de la dirección, sin que haya colisión mientras los datos viajan por el mismo bus. Dicha señal, se denomina EA (Habilitación del Latch en inglés).

Para diferenciarse la señal de habilitación de memoria RAM, existe una señal para memoria de código externa llamada PSEN (Program Store Enable). Todas las mencionadas anteriormente, son señales relacionadas con el tráfico de información con los componentes externos o periféricos. El comportamiento se aprecia mejor en un diagrama de tiempo de lectura o escritura en memoria externa, propio del funcionamiento de la arquitectura 8051 (que se ofrece en la figura 7). Estos procesos son realizados por el firmware Gemcore, cuando algún comando interpretado así lo requiere.

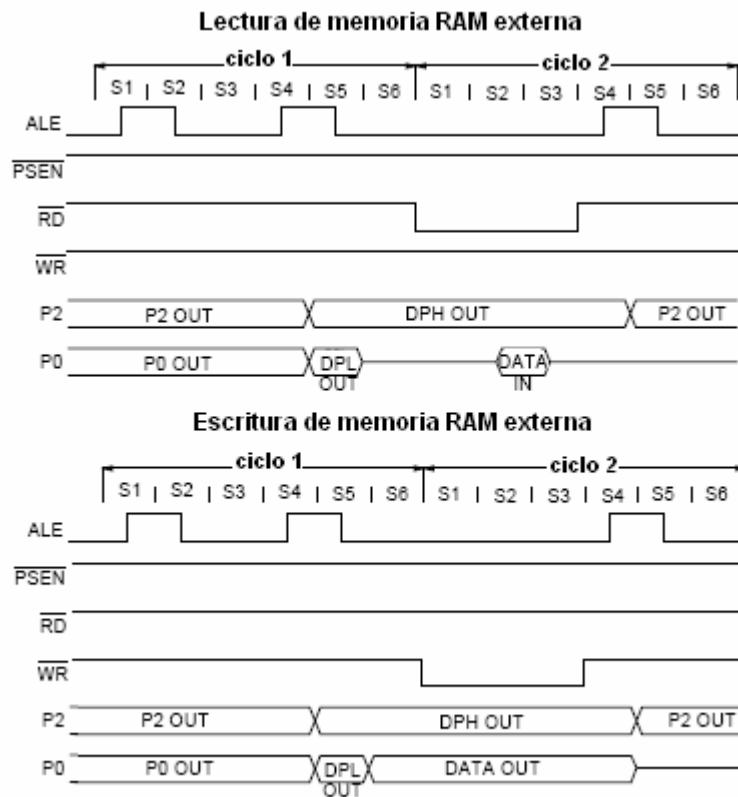


Figura 7. Esquema del procedimiento de lectura escritura. Este diagrama lo provee el fabricante, y demuestra la manera en que se acceden los periféricos que se configuran como si fuesen secciones de RAM externa.

Adicionalmente existen varias entradas de interrupción, las cuales fueron utilizadas por los diseñadores del módulo OEM para interaccionar con otros de los dispositivos presentes. Otras las señales relevantes son las de comunicación serial, TXD y RXD y las de I2C (SDA y SCL) disponibles para comunicarse con el exterior. La comunicación serial es importante, pues el firmware Gemcore, si se encuentra en estado de espera, verifica si se reciben vía serial una serie de bytes correspondientes con comandos, que ejecuta y devuelve el resultado por la misma vía. Más adelante se explicara en más detalles estos comandos y la posibilidad de proveniencia externa o interna de los mismos.

3.4. Otros componentes

El módulo GemCore500, cuenta con otros componentes importantes, entre los que se cuentan:

- Memoria de código externa.
- Memoria RAM externa.
- Memoria RAM estática.
- Dispositivos selectores
- Reloj de tiempo Real.
- Gestor de alimentación
- Conectores de salida.

3.4.1. Memoria de código externa.

La memoria de código externa se encuentra ubicada en el circuito integrado AM29F040 de AMD, y en ella se almacena eventualmente el programa creado por el usuario. Este programa corresponde con el que se ha llamado hasta el momento “de más alto nivel” en el presente texto (“de más alto nivel” en comparación con el firmware Gemcore, que probablemente sea programado en assembler e interacciona directamente con la circuitería del Módulo).

Cabe destacar que se realiza en lenguaje C y gracias a un compilador y un programa que se incluye al momento de adquirir el módulo, permite desarrollar aplicaciones versátiles para el uso del módulo, a condición que se disponga del formato y de cada uno de los comandos creados para tal fin. Tal como se ha mencionado anteriormente, el firmware Gemcore, grabado en la memoria interna del Philips 83C524, accede a esta memoria externa de programa, interpreta los comandos y los transforma en los comandos que generan la realización de la tarea deseada.

En caso de encontrarse esta memoria vacía (ausencia de programa en esta memoria), el firmware GemCore se queda en estado de espera, aguardando que lleguen comandos válidos vía serial. Es una memoria de 512Kbytes, asequible gracias a mecanismos de paginación, lo que le da cierta capacidad de memoria. Esta dividida en cuatro bloques de aplicación de longitud definida. El primer bloque tiene 64Kbytes, el segundo 64Kbytes, el tercer bloque 128Kbytes y el cuarto bloque 256Kbytes.

3.4.2. Memoria RAM externa

La memoria RAM Externa se encuentra ubicada en un circuito integrado AT29C04 de ATMEL. Esta memoria contiene datos (variables, arreglos, estructuras y otros) pertenecientes al programa insertado por el desarrollador que hace uso del módulo. Es una memoria de 512KBytes, la cual por medio de un mecanismo de paginación de memoria, es posible accederla, dado que el bus de direcciones es capaz de direccionar directamente hasta 64Kbytes. Sin embargo, el usuario no interviene en este proceso de selección, excepto en lo que refiere a la distribución de las aplicaciones en bloques, lo que asimismo condiciona la distribución de la data generada y manejada por la aplicación.

3.4.3. Memoria RAM estática

Es un espacio de memoria de longitud 8Kbytes, correspondiente con una memoria RAM estática, presente en un circuito integrado CY6264 de CYPRESS. Esta memoria se constituye en una serie de posiciones que no se borran al des-

energizar o apagar el módulo, donde se almacena información necesaria, como podría ser datos del equipo, o registro de transacciones

3.4.4. Dispositivos Selectores.

Dado la complejidad de la variedad de circuitos presentes y posibles que pueden compartir el bus de direcciones/datos más las señales de control, se aprecia que el fabricante diseñó una serie de interconexiones basadas en lógicas digitales de control de señales, para permitir la correcta selección de dispositivos y su posterior transmisión y tráfico de datos. Dicho sistema de selección y control de señales se encuentra compuesto por:

En la parte de selección de dispositivos: un decodificador (o selector) 74HC138, una compuerta lógica NOR de dos entradas 74HC02, una compuerta lógica OR de dos entradas 74HC32. Esto en la parte de selección de dispositivos de Memoria y RTC. En la parte de la selección de los IC100, se realiza la selección por medio de un F/F tipo D 74HC173.

En la parte de control de señales, existe un LATCH transparente Octal tipo D 74HC573 y un F/F tipo D 74HC273. Estos dispositivos previenen la colisión de información, preservando la dirección en el dispositivo seleccionada, aislada de los datos que viajan por la misma línea. La parte de los IC100 no requiere de control de señales, pues éstas son exclusivamente dedicadas a la tarjeta inteligente. No usa bus de direcciones/datos.

3.4.5. Reloj de tiempo real.

Este circuito integrado, que corresponde con un M6242B de OKI, tiene por función, en base a un valor pre-ajustado, llevar la cuenta del tiempo transcurrido, permitiendo obtener en cualquier momento solicitado la hora y fecha con un buen margen de exactitud (menos de 50 PPM de desviación a una temperatura ambiente 25°C y menos de 4 PPM si la tensión de alimentación permanece por encima de 2V).

Funciona conectándose directamente del bus compartido de direcciones/datos y las señales de control del mismo. Solo requiere de 4 señales del bus (un nibble) para

operar. Funciona en todo momento gracias a la pila CR2430 que viene adosada al sistema, la que mantiene el funcionamiento interno aun al apagarse todo el módulo, lo que permite no perder la correcta hora y fecha.

3.4.6. Gestor de Alimentación

Se trata de un MAX691 de MAXIM, el cual es un circuito supervisor, que gestiona la alimentación entre la batería CR2430 y la alimentación externa VCC. Realiza la selección de la fuente de alimentación de los elementos que requieren seguir funcionando, como la memoria y el RTC, permitiendo el ahorro de energía de la pila cuando la energía externa de alimentación se encuentra conectada

3.4.7. Conectores de salida.

Dado que el presente módulo GemCore es un módulo OEM, que no puede funcionar por sí solo, el módulo tiene tres conectores de peine de pines de salida, por donde se le conecta con el desarrollo externo que le dará vida a la aplicación. El primer conector, llamado K2, está conformado por dos hileras de 6 pines (12 pines en total). Este conector tiene conexión directa a las entradas de las señales de interfaz con la tarjeta inteligente principal. Está pensado para poder conectar un muelle externo de lectura de tarjeta inteligente, dejando el que se encuentra incluido en latencia.

Esto implica que solo puede haber un solo muelle de tarjeta inteligente operativo a la vez. La conexión de uno externo, desactiva al interno. El siguiente conector, denominado K3, esta compuesto por dos hileras de 13 pines (26 en total) y contiene todas las señales necesarias para conectar dispositivos periféricos, incluyendo el bus de datos direcciones. Entre las señales que contiene esta la alimentación y referencia para esos dispositivos externos periféricos (tales como LCD, teclado, Buzzer, etc.).

El tercer conector, denominado K4, contiene el acceso las señales seriales de los puertos Seriales e I2C del microcontrolador central, en niveles TTL. Adicionalmente, contiene la entrada de la alimentación y referencia del módulo en general. Un grafico de las señales de los distintos conectores y sus señales se aprecia en la figura 8.

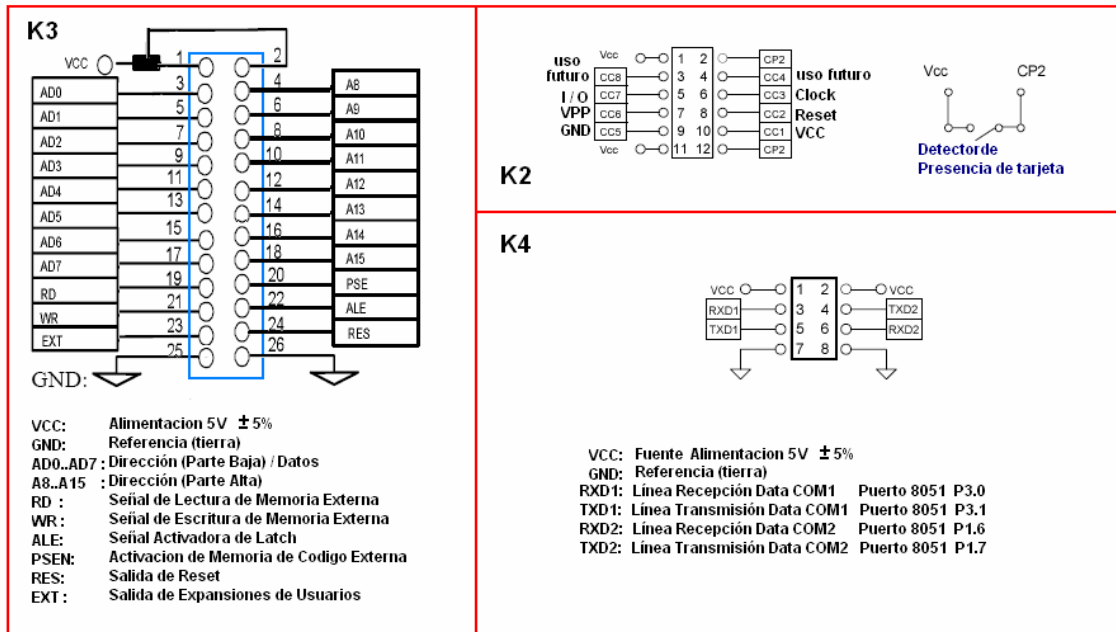


Figura 8. Esquema de las señales que son proveídas al exterior del módulo Gemcore500 por medio de los 3 conectores que dispone

3.5. Funcionamiento del módulo

Tal como se ha descrito en los capítulos anteriores, el firmware GemCore presente en el microprocesador central, es un código residente en la memoria interna del microcontrolador, que forma una de las capas del sistema de aplicaciones GemCore. Este sistema esta formado por la capa de comunicaciones, La capa de aplicaciones de usuario, la capa de Librería (llamada Librería de Interfaz con el lector GemCore), la capa de Firmware GemCore en sí y la capa física (o circuitos electrónicos). Es posible apreciar la distribución de estas capas en la figura 9. En esta figura se muestra un esbozo del funcionamiento en caso de utilizarse un programa interno.

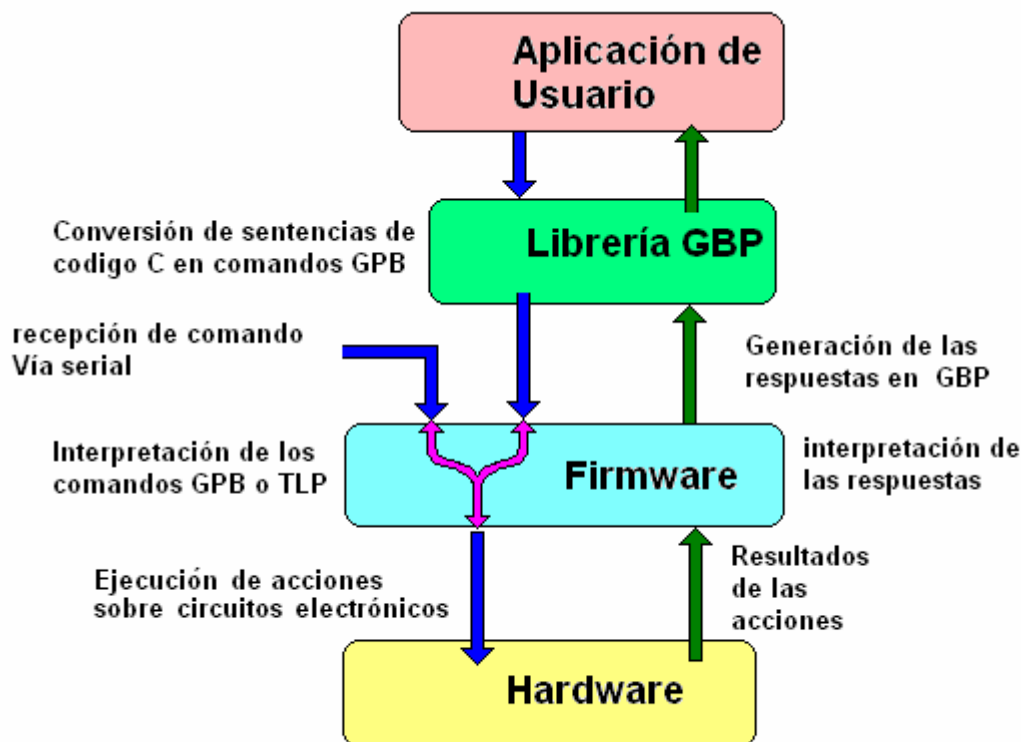


Figura 9. Esquema de la interacción de las distintas capas que conforman una típica aplicación con el Gemcore.

La aplicación de usuario, hasta ahora referida como “aplicación de más alto nivel”, es un conjunto de sentencias de lenguaje C que se compilan con las herramientas proporcionadas por GemPlus y se descargan vía serial, siendo el Firmware GemCore el que se encarga de localizar el programa ya traducido a lenguaje de máquina en la memoria CODE externa del módulo GemCore.

De esta manera, las sentencias de código en C que tienen relación con acciones a ser tomadas por el módulo, son traducidas a comandos en Bloques de uno de dos formatos Posibles: TLP o GPB. Dado que el TLP no se va a utilizar, no se explicará, contrario con el GPB, que se explicará más adelante. Una vez convertidos en comandos en bloques asimilables por el firmware, éste los reconoce (o en caso contrario reintenta o devuelve un mensaje de error) e inicia la ejecución de las acciones de control de capa física (Hardware) que son requeridos.

Muy probablemente, esta última ejecución sea una secuencia de sentencias en lenguaje ensamblador de acceso a memoria externa, determinados primeramente a

seleccionar el dispositivo adecuado y luego a enviarle o solicitarle los datos que son necesarios para cumplir la labor especificada por el comando en bloques recibido. Dado que el GemCore es un Firmware Propietario realizado por GemPlus (ahora Axalto), no se dispone de más información que las direcciones con que se encuentran asociados los distintos periféricos, siendo una incógnita los algoritmos utilizados. El propio fabricante sólo provee de una ilustración de cómo estructuró en forma de direcciones de memoria externa los distintos elementos conformantes del sistema. Tal distribución de memoria se ofrece en la figura 10

De esta manera, en resumen, el funcionamiento completo del sistema es una secuencia que se inicia en líneas de programa en lenguaje C que se compilan relacionándolas con la librería de Gemplus, y se descarga al módulo. Al encender el módulo se le indica que se recibirá un programa en lenguaje de maquina y el firmware se encarga de recibir el archivo de programa y lo almacena y ordena en la memoria de código externa. Una vez re-arrancado el módulo, al hallar este programa almacenado en memoria de código, lo ejecuta, transformándose cada instrucción escrita por el usuario en un comando estructurado en bloques (gracias a la compilación). En tal sentido, si por ejemplo se trata de letras en el LCD, el dispositivo hace una escritura en memoria externa, direccionando a la posición 2300 y escribiendo datos.

De este modo, se selecciona el LCD, habilitándose, y se le transcriben datos que han sido ensamblados según la metodología de activar y dirigir dicho dispositivo. Tal es el funcionamiento con los otros periféricos. De igual manera, es posible que algunos dispositivos, sólo para activarse o seleccionarse, requieran de más de una dirección y data, antes de que la verdadera transferencia de información ocurra. En tal sentido, los periféricos fundamentales como LCD, Teclado Matricial y Buzzer se encuentran pre-considerados y un código se ha escrito para tal fin. En este caso, resta diseñar el circuito adecuado que los maneje correctamente, pues dichos elementos no se encuentran presentes en el circuito interno del módulo.

De la misma manera, dado esta metodología, si se va a introducir un periférico adicional, se le debe direccionar por medio de la posición 2700 a la 27FF. En

		tipo de memoria		
Posición (Todas las direcciones en Hexadecimal)		IDATA	XDATA	CODE
	0000	Registros Internos del GemCore	Buffer de Intercambio	Sistema Operativo GemCore
	00FF			
	0100		Espacio de Usuario # 1	
	1FFF			
	2000		Extensiones (Periféricos) *	
	3FFF			
	4000		Espacio de Usuario # 2	Reservado Para Extensiones Gemplus
	7FFF			
	8000		Espacio de Usuario #0	Aplicación del Usuario
FFFF				

Valores para Periféricos *	{	2000	Manejo de VPP
		2100	Manejo de Memoria
		2200	Manejo de teclado
		2300	Manejo de Pantalla LCD
		2400	Manejo de Reloj de Tiempo Real
		2500	Manejo de Entradas/Salidas
		2600	Reservado para Uso Futuro
		2700	Acceso a Periféricos externos creados por el usuario

Figura 10. Esquema ofrecido por el fabricante que ilustra la distribución de memoria del módulo. Gracias a estrategia de paginación (división en páginas) se puede direccionar hasta 512K en el espacio de 8000 hasta FFFF

consecuencia, debe desarrollarse un circuito adicional que sea activado por cada una de las direcciones estipuladas, tanto las preconcebidas por el fabricante, como las desarrolladas en este trabajo, y así asegurar la unicidad de la activación de cada componente. Es evidente la necesidad de cohesión entre la aplicación de usuario con el diseño del circuito que operaría dichos periféricos.

Igualmente, el acceso a aquellos componentes que sí se encuentran incluidos en la superficie física del módulo, tal como memoria y Reloj de tiempo real, se realizan de forma similar, por medio de un proceso de lectura/escritura de memoria externa dentro del núcleo del GemCore, que se traduce en la realidad en la selección de cada componente, siendo más compleja para la memoria, dado que por medio de mecanismos de paginación de memoria se expandió hasta lograr direccionar hasta 512Kbytes.

Hasta el momento se ha pasado por alto la capa de comunicaciones. Esta capa se basa en un puerto que usa un protocolo serial asincrónico Full duplex de dos líneas, manejados por el UART interno del microcontrolador central, y controlado por el GemCore. Al encenderse, tiene una configuración por defecto, que se encuentra fijada a 9600 Baudios mensaje de 8 bits, y bit de parada y no considera la paridad. En el caso de que el módulo GemCore no halle información de aplicación de usuario descargada, o que la propia aplicación no bloquee esta posibilidad, el firmware GemCore explora este puerto continuamente a la espera de bytes que conformen comandos en Bloque. En caso de recibirlos, los ejecuta y retorna una respuesta. Estos comandos en bloques se denominan GBP.

3.6. Protocolo de comandos en Bloque

Los comandos en bloque que comprende y ejecuta el firmware Gemcore están diseñados en una forma similar y simplificada del protocolo T=1 de comunicación con tarjeta inteligente. Este protocolo se denomina GBP por Protocolo de Bloques de Gemplus en Inglés. Bajo este protocolo los datos y órdenes transmitidos son intercambiados bloques o series de bytes entre la fuente y el destino, es decir, entre el elemento que dirige las acciones (ya sea Aplicación de Usuario o Puerto Serial) y elemento ejecutor (Firmware Gemcore). Según este protocolo, existen tres tipos de mensajes.

- Bloques I (de Información): Contiene datos que son intercambiados entre la fuente y el destino

- Bloques R (de Respuesta): Contiene una respuesta positiva o negativa de reconocimiento a los elementos de transmisión.
- Bloques S (de supervisión): Sincroniza las transmisiones en un momento dado (por ejemplo, cuando se pierde la secuencia o se repite un error).

Los datos intercambiados entre la fuente y el destino Tiene el siguiente formato (siendo cada ítem un byte o serie de bytes):

NAD	PCB	LEN	DAT	EDC
------------	------------	------------	------------	------------

Este es el formato general de mensajería, dentro del cual, en el DAT va una clase interna de comando homónimas con el módulo y el firmware (es decir, también llamado GemCore, comandos Basados en Gemcore) y el resto es el contenedor que garantiza que el correcto intercambio.

3.6.1. Campo NAD

En el GBP, NAD es un byte que indica el sentido del mensaje, es decir, si va de fuente a destino o de destino a fuente. El nibble alto indica el emisor del mensaje y el bajo indica el receptor. La fuente (el ente generador de los comandos, es decir, el elemento externo que comanda vía serial, o la aplicación de usuario) se codifica como 2 y el destino (ente ejecutor Gemcore) se codifica como 4. Así, un comando que parte de la fuente al destino tendría un NAD = 0x42, por el contrario, un mensaje de respuesta del destino a la fuente, tendría un NAD = 0x24.

3.6.2. Campo PCB

El PCB es un byte que indica el tipo de bloque que se ha recibido. Si pertenece al tipo I, R o S. Si es un bloque I, engloba todas las comunicaciones normales de envío de comandos y respuestas satisfactorias. En tal caso solo presenta un bit de paridad, que se alterna con cada nuevo mensaje, empezando con el primer mensaje enviado desde el encendido. En este primer mensaje el bit P esta en cero, el siguiente en 1, el próximo mensaje en cero y así. El bit de paridad se ubica en el séptimo bit del byte, estando los demás bits a cero, lo que genera una alternancia de 0x40 y 0x00.

Si es tipo R, alguno de los dos terminales ha detectado una anomalía e indica error, y el tipo de error. El código de Error recibido puede ser:

7	6	5	4	3	2	1	0
1	0	0	S	0	0	E	V

V=1 Error de Comprobación
E=1 Otro tipo de Error
S=1 Error en la secuencia de paridad

Cabe destacar que los errores aparecidos en este campo son inherentes a anomalías en el cuerpo del mensaje. Errores en la ejecución del comando como tal, si el cuerpo del mensaje está correcto, son devueltos como códigos informativos dentro de un bloque tipo I. Los bloques S se envían cuando la secuencia de los bits de paridad se ha perdido y se requiere resincronizar. Esto indica que el siguiente mensaje (luego de la respuesta) va a tener un PCB con paridad 0 y se inicia de nuevo la secuencia. El formato solicitud de resincronización y respuesta es:

7	6	5	4	3	2	1	0	Solicitud	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0		Respuesta	1	1	1	0	0	0	0	0

3.6.3. Campo LEN

El LEN es un byte que indica el número de elementos que se recibirán como parte del siguiente campo DAT.

3.6.4. Campo DAT

Este Campo DAT está compuesto por la cantidad de bytes que indica la cifra presente en el campo LEN. En este campo viene el Comando en sí o la respuesta a un comando. En caso de ser el comando basado en GemCore, se encuentra compuesto por un primer byte, el cual es un código de operación, tomado de una lista de valores significativos proporcionado por el fabricante. Los siguientes Bytes son opcionales (según la naturaleza del comando) y forman parte de los parámetros del comando. De ser necesario, le siguen otra serie de bytes que serían los datos asociados.

En caso de ser la respuesta a un comando, se encuentra compuesto por un byte con un código de estatus y de ser el caso, una serie de bytes con data. En el Anexo 1 se ofrece una lista de la codificación y sintaxis de estos comandos

3.6.5. Campo EDC

Este campo EDC es un byte cuyo contenido que se obtiene a partir de la realización de la operación lógica O exclusivo (XOR) de cada uno de los bytes que componen el mensaje, desde NAD hasta el último byte de DAT. Constituye un elemento de detección de errores de envío, por lo cual, debe ser calculado tanto por la fuente como por el destino y verificado comparándolo con el recibido.

CAPITULO IV

4. METODOLOGIA

El presente trabajo de grado, se desarrolló siguiendo la siguiente secuencia de pasos que condujeron a la obtención de la solución planteada y construcción del prototipo de equipo lector de tarjetas inteligentes

- Se investigó los conceptos básicos que se relacionan con las tarjetas inteligentes y su entorno. Se realizó una investigación sobre resúmenes de la norma ISO7816 y los aspectos que regulaba, así como los elementos más relevantes que resultaran pertinentes para una correcta aproximación al tema.
- Se realizó una búsqueda de bibliografía relacionada con tarjetas inteligentes y sus procesos, características y particularidades, así como datos técnicos pertinentes, que permitiesen complementar y aumentar la información y comprensión ya lograda con la investigación alrededor de los estándares internacionales.
- Se realizó una búsqueda de todos los documentos técnicos, hojas de datos, notas de aplicación y material informativo e instructivo, disponibles del fabricante, que permitieron comprender y conocer el funcionamiento y estructura del módulo GemCore. Se estudio a fondo la información recabada y se inicio el establecimiento de relaciones con respecto a los conocimientos teóricos previamente recabados.
- Se procedió a contactar al proveedor y a la oficina de soporte técnico del fabricante del Módulo, en la persona del distribuidor autorizado para la región andina, con la idea de solicitar mayor información y apoyo en el conocimiento del módulo Gemcore y sus características.

- Se precisó y delimitó más específicamente las características y particularidades que se esperaban del producto a desarrollar, así como el ámbito general de operación y aplicación.
- Se investigó acerca de las aplicaciones de la empresa que ya se encontraban operando, y de los equipos que actualmente se están utilizando. Se participó en algunos proyectos de desarrollo y se observó la operación de controles de acceso, donde se presentaron fallas y comportamientos no óptimos, que fueron base para la necesidad de este trabajo.
- Se estudió las características técnicas y electrónicas de otros lectores y equipos que ya se encuentran en operación, así como su margen de precio, su fiabilidad, resistencia y operatividad.
- Se estudió las distintas variantes de programación interna del módulo. Esto incluye, el lenguaje de programación, las librerías, archivos relacionados, entorno de programación, compilación; método de descarga de aplicación al lector, método de control vía serial desde el PC o un controlador externo. Adicionalmente, el estudio de los comandos especiales y variables, estructuras, punteros y demás conceptos relacionados con la programación del lector.
- Se diseñó el primer diagrama de bloques a grandes rasgos de elementos conceptuales que deberían ir desembocando en el dimensionamiento del lector a desarrollar y su culminación en un prototipo.
- Se procedió a determinar los elementos que deberían ser primeramente resueltos y diseñados, tomándose la decisión de empezar con la sección de interfaz hombre-máquina. Esto es, los dispositivos que toman datos del usuario y retornan data al usuario.
- Se realizó la investigación de la manera de interconexión electrónica y circuital de los elementos seleccionados de interfaz. Esto implicó cierto margen de espera y algunas dificultades, lo que motivó a una investigación aun más profunda acerca del módulo en sí.

- Se estudió a fondo la estructura del módulo Gemcore 500, obteniéndose un esbozo de plano general, con lo cual se procedió a realizar comparaciones con sistemas similares en la red (Internet) y con ejemplos ya conocidos (como el microsistema utilizado en la materia Microprocesadores I, dado que el módulo guarda una lejana semejanza con él) para intentar deducir la circuitería a utilizar. Esta debería ser coherente el programa residente en memoria (firmware) del control central del Gemcore, dado que el funcionamiento de los elementos conectados depende de la correcta conexión y correspondencia con los algoritmos de activación diseñados por el fabricante.
- Al obtener esta configuración del fabricante, y aprovechando los conocimientos adicionales adquiridos, se realizó el primer esquemático de una tarjeta de circuito impreso que contuviese dichos circuitos, en conjunción con otros elementos más. Esto fue desarrollado tomando en cuenta las necesidades planteadas, (modularidad, expansibilidad y capacidad de adherir/quitar elementos sin tener que realizar modificaciones circuitales).
- Se continuó con la refinación y optimización de este circuito, así como la selección y adquisición de los componentes y materiales necesarios para su implantación en un prototipo.
- Se plantearon los esquemas de diseño de equipo necesarios para la correcta construcción del circuito impreso (inclusión de planos de tierra, grosor de las pistas, posición física de los componentes, distancia entre sí, tipo de trazado, etc.).
- Se realizó la primera prueba del circuito, colocando los componentes en una tarjeta de ensayos (protoboard) y cableando el circuito completo. Se construyeron los cables planos necesarios para la conexión de este circuito experimental con el módulo Gemcore. Se probó utilizando la estrategia de enviar comandos simples hallados en los manuales al módulo desde la PC por el puerto serial (utilizando una aplicación simple proveída por el fabricante), para observar el correcto manejo de los componentes de interfaz (Escritura en

LCD, Buzzer, lectura de teclado, y colocación de valores en puertos de expansión).

- Una vez ajustado y corregidos los errores, se procedió a diseñar el trazado de la tarjeta, siguiendo los lineamientos establecidos. Se realizó el proceso de disposición de los elementos, ruteo y optimización. Luego de esto, se revisó de cerca el trazado obtenido y se eliminaron errores y posibles fuentes de distorsión (cercanía de las pistas, planos de tierra, correcto trazado, etc.).
- Se imprimió y realizó el traspaso del circuito a la placa y se aplicaron los métodos más efectivos para llevar a cabo la creación de del circuito en físico. Se pasó por la fase de traspaso impresión, eliminación del cobre, apertura de agujeros, estañado, soldadura de los componentes y se obtuvo así una primera placa operativa para conectar por un lado con el módulo, y por otro lado con los periféricos de interfaz de usuario a manejar.
- Se realizó la programación en las herramientas proveídas, de las rutinas, funciones y procedimientos. Este código de programación se realizó en forma de archivos de cabecera, que realizan por sí solos el manejo de los comandos especiales del fabricante, de modo que al ser incluidos, el usuario solo llame funciones estructuradas y pase parámetros, que dichas funciones se encargan de convertir en tareas para el módulo Gemcore. Con esto se busca crear una interfaz para que luego, programadores creasen aplicaciones sobre la base de estos archivos incluidos.
- Se constató que no se conocían ciertos comandos especiales adicionales, relacionados con la comunicación externa, necesarios para crear un archivo incluíble de lectura y escritura de datos de comunicación en el puerto serial y/o el I2C. Luego de muchos inconvenientes y espera, se optó por abandonar la solución de programa hasta ahora desarrollada, pues hasta este punto dejaba varias expectativas sin cumplir. Se abandonó a favor de desplazar la ubicación del programa central, del microcontrolador central del módulo Gemcore, a un microcontrolador externo, que emulara el protocolo de comunicación serial de la PC (y el encapsulado de los mensajes). De esta manera, este nuevo control,

enviaría al módulo los comandos conocidos, y realizaría por sí mismo las otras tareas que no se podían implementar en el módulo, por no conocer los comandos adecuados

- Se realizó la selección del sistema de control externo del módulo, fijándose en un sistema distribuido en dos microcontroladores independientes, uno de los cuales, comandaría al módulo y serviría de interfaz, cumpliendo las funciones de los archivos incluidos de la solución desechada (además de facilitando el esquema de comunicación con el otro microcontrolador). El otro contendría la aplicación de usuario descargable que programarían mas adelante desarrolladores, conforme se aplique este sistema en alguna solución concreta.
- Se implemento el protocolo de comunicación comprendido por el Gemcore en el primer ente de control, y se realizo la programación de la comunicación con el segundo ente de control, así como la creación de las funciones estructuradas que permitiesen, que a un comando simple recibido desde el segundo ente de control, el primer ente se encargase de interpretar, crear y estructurar el mensaje, así como de traducir la respuesta obtenida.
- Se seleccionaron los componentes físicos que harían tal función y se le descargo el programa desarrollado, probándose en protoboard, con el módulo GemCore y la tarjeta con interfaces ya construida. Se afinó y optimizó el circuito. Adicionalmente se le agregaron nuevos componentes, como transceptor para poder comunicarse con PC y un dispositivo de memoria no volátil.
- Se desarrolló una aplicación demo simple, para el segundo ente de control, para comprobar la funcionalidad del sistema. Esta aplicación, al momento de entrar en producción seria sustituida, por una aplicación desarrollada posteriormente por desarrolladores en respuesta a una solución que s desee implementar.
- Se realizó el dimensionamiento y diseño de la fuente de tensión que alimenta correctamente a todo el circuito diseñado, así como aquellos periféricos que

no entran en los alcances de la tesis, pero que seguramente surgirán al entrar este diseño en producción.

- Se realizó el diseño del esquemático, y el trazado de esta segunda tarjeta, donde se colocarían todos los sistemas adicionalmente desarrollados como entes de control, almacenaje no volátil y comunicación. Asimismo se incluyó la fuente de tensión diseñada.
- Se construyó la segunda tarjeta y se realizó la construcción de una estructura donde se colocarían todos los circuitos, para darle forma de alguna manera al prototipo. Esta estructura es solo demostrativa y aunque conlleva la filosofía del proyecto, es demostrativa y rudimentaria.
- Se realizaron numerosas pruebas y ajustes de programa y del circuito electrónico.
- Se realizó la escritura del documento escrito que informe sobre el trabajo realizado.

CAPITULO V

5. SISTEMA DESARROLLADO

Tal como se ha planteado, se deseaba desarrollar una plataforma (tanto de aplicaciones como de circuitos electrónicos) que permita convertir al módulo OEM GemCore500 en un lector funcional multipropósito, modular y autónomo, para suplir la necesidad de la empresa E-smart Consulting de un dispositivo de estas características cada vez que surgiese alguna posible aplicación en la que se requiriese tal equipamiento. Por ello se inició con el desarrollo de circuitos electrónicos, evidentemente basados en principios de sistemas digitales inherentes al manejo, control y gestión de tráfico de señales eléctricas de naturaleza binaria, coincidentes con la programación interna tanto de las rutinas de control de periféricos existentes en el firmware GemCore (preconcebidas por el fabricante), como para las programadas durante el desarrollo de esta plataforma general.

Dado que el Firmware GemCore es propietario y el fabricante no ofrece visos sobre el algoritmo de control de estos circuitos, se le solicitó información acerca de la conexión de las líneas de datos con los periféricos preconcebido (a saber, LCD, teclado y Buzzer). Esto requirió de una espera prolongada y numerosas cartas y comunicaciones, durante las cuales se realizó, paralelamente, un estudio a fondo de la circuitería del módulo, llegándose al extremo de realizar el levantamiento de un esbozo del plano del módulo. Este esbozo (cuyo diagrama de bloques se aprecia en la figura 11 y un circuito aproximado se presenta en el anexo 3, Ilustración 1 de los Anexos) permitió comprender mejor la configuración de la organización de los elementos que conforman el módulo.

Simultáneamente, se realizaban numerosas pruebas para determinar, sabiendo la dirección de memoria asignada a los periféricos preconcebidos la estrategia de exploración y configuración utilizada. Este procedimiento se llevó a cabo con el

se encargaría de ser el nodo de señales que se encontrarían con la posible interfaz de usuario y otros periféricos externos.

5.1. Tarjeta Principal de Interfaz.

Este circuito impreso tendría como función obtener las señales que proporciona el conector K3 del módulo GemCore500 y gestionarlas, de modo que se tengan conexiones independientes donde se puedan colocar adecuadamente tanto los periféricos preconcebidos, como los que se pueden ir agregando posteriormente. Esta tarjeta tiene una arquitectura de bus de datos direcciones basadas en el bus del 8051, por lo que favorece la comunicación con elementos de manera paralela. En tal sentido, la primera necesidad a implementar, era el transporte de las señales del conector K3 hasta esta tarjeta. Dada la cantidad de señales (26) se pensó en un cable plano tipo IDE.

Este cable es idóneo, pues presenta baja resistencia interna, soporta niveles de corriente relativamente elevados (en teoría hasta 1A, pero en ensayos se llegó a comprobar que con hasta 3A no se observaba falla o discrepancia alguna). Además es una tecnología probada y usada frecuentemente. Adicionalmente, los conectores compatibles con los peine de pines son fácilmente instalables a presión en los extremos del cable. Una vez realizada la decisión del cable a usar, se hizo obvia la presencia de siete elementos en la primera selección de dispositivos en la Tarjeta Principal de Interfaz.

En esta tarjeta se colocaría el peine de pines de conexión del cable plano tipo IDE proveniente del conector K3 del módulo Gemcore, el acceso a LCD, el control de contraste y luz trasera del LCD, el acceso a teclado, el acceso a buzzer, los accesos a otros periféricos que se comuniquen vía paralela y el sistema de gestión de señales. Una vez enumerados todos los subsistemas presentes en esta primera tarjeta, se realizó la selección inicial de los componentes físicos a usar, y su distribución espacial en el trazado. Como punto de partida, y luego de un recorrido por los expendedores de suministros electrónicos, se determinó que existía mayor probabilidad de adquirir placas de circuito impreso (backelita) de una medida de 9cm

de ancho por 15cm de alto, lo que aunado a que armoniza con la medida del módulo Gemcore500 (8cm de ancho por 9,5cm de alto) influyó en la selección de tal medida como tamaño físico de la tarjeta. En la Figura 12 se ilustra esta distribución inicial de componentes.

Una vez determinada la disposición de los subsistemas, se procedió a idear la manera de lograr la conectividad con el resto de los elementos. Esta necesidad se suplió con la adquisición de conectores hembra lineales de 100 milésimas, para acoplarlo con peine de pines macho de 100 milésimas. De esta manera, se asegura la posibilidad de extraer y volver a incluir los periféricos sin necesidad de modificaciones en el circuito. Dado esto, el desarrollo de los demás periféricos esta condicionado a tener compatibilidad con estos pines de conexión. Por ello se siguió el circuito recomendado por el fabricante para los periféricos preconcebidos, y utilizándolo como base se amplió y agregó el resto. Un hecho relevante observado en la disposición de señales que se obtienen del conector K3 es que no se tiene acceso externo a ninguna interrupción externa. Eso implica que la interfaz de usuario no se maneja por interrupción sino por exploración.

5.1.1. Sistema de Gestión de señales

El circuito recomendado por el fabricante se basa en la lógica de tráfico de señales, sólo que la lógica puede parecer engorrosa, sin embargo es la base para toda la gestión posterior. Básicamente, un selector de datos (Decoder) toma las direcciones (parte alta) del bus y con ella activa un Seguidor (o buffer) octal (con la dirección 2200 en hexadecimal) o un Flip Flop tipo D octal (con la dirección 2300 en hexadecimal). De cualquier manera, el procedimiento elegido por el fabricante, para escribir en LCD (en formato de medio bus o por nibble) es colocar el dato en el nibble bajo del bus y modificar los dos bytes menos significativos del nibble alto en escrituras sucesivas para operar las líneas de control del LCD.

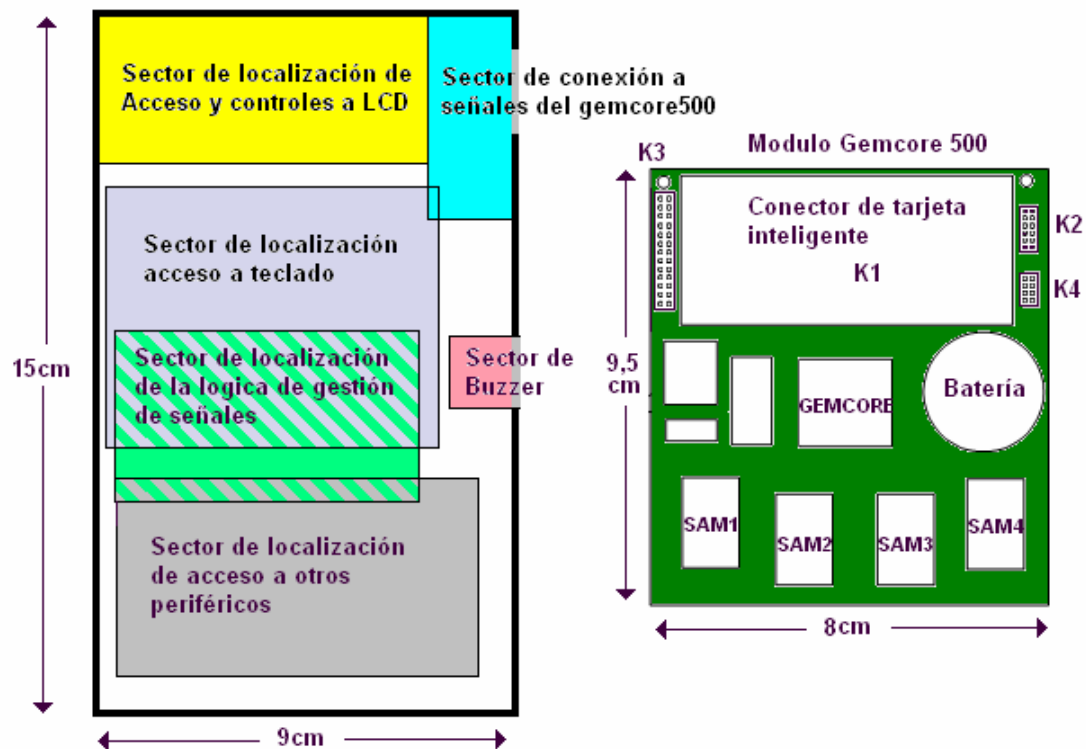


Figura 12. Distribución espacial elegida para situar los bloques de manejo de señales en la tarjeta de interfaz. Esta disposición fue elegida para coincidir con las posiciones adecuadas del resto del circuito.

Según esta topología no se contempla la lectura de la bandera de ocupado del LCD. La escritura de datos para el LCD queda registrada en el Flip Flop. Igualmente, a través del mismo Flip Flop se realizan las exploraciones del teclado matricial a conectar, solo que los resultados se recogen a través del Buffer octal cuando se realiza lectura en memoria 2200 hexadecimal. Adicionalmente, el buzzer se adquiere escribiendo directamente en la posición más significativa del Flip Flop. El Flip Flop es activado también por la señal de escritura WR (además de la señal del selector). El buffer es activado por la señal de lectura RD (además de la señal del selector).

Una vez implementado este circuito, del mismo selector, se extrae la selección para la posición 2700 hexadecimal (que indica periféricos no preconcebidos por el fabricante) para activar un segundo selector en cascada y a su vez un Latch transparente tipo D que permita almacenar la parte baja de la dirección, para especificar de esta manera el dispositivo periférico externo al cual se va a acceder. En

tal sentido, si desde el microcontrolador central se procede a leer la dirección de memoria externa 2701, esta configuración activa la opción más significativa del primer selector, que a su vez realiza la activación del segundo selector (en cascada) y media selección del Match transparente.

La señal ALE, que se activa precisamente con el fin de conservar la parte baja de la dirección (al cambiar la información del Puerto P0 del controlador de direcciones, a datos) completa la selección del match y por lo tanto almacena esta parte baja, para así constituirse esta parte baja en señales de control del segundo selector. De esta manera se seleccionaría el periférico externo 1, de los periféricos no preconcebidos por el fabricante. En consecuencia, por las dimensiones físicas de la tarjeta de circuito impreso sólo se planteó la salida para seis periféricos de este estilo.

De la misma manera, cada salida se realiza por medio de un peine de pines de modo que se conecta directamente ese nuevo periférico, y se le provee de la señal de datos/direcciones (parte baja del bus), señales ALE, WR, RD y señal de selección particular (generadas por el segundo selector) además de alimentación y referencia. Uno de estos periféricos fue desarrollado e implementado y es un módulo independiente de presentación en 7 segmentos de 3 cifras. El resto de los componentes son el circuito de control de contraste del LCD, que es un potenciómetro que permite generar una escala de tensión para así introducirla en el pin que gradúa el contraste de la pantalla; un jumper para darle alimentación o no al pin del LCD donde se enciende la luz posterior, y un segundo jumper que coloca la parte no utilizada del bus de datos del LCD a tensión de fuente, para algunos LCD que así lo requieren. Adicionalmente unos diodos de señal 1N4148 y resistencias de 10K Ω para complementar el sistema de exploración de teclado. Un esquema de bloques del sistema presente en esta tarjeta se ofrece en la figura 13.

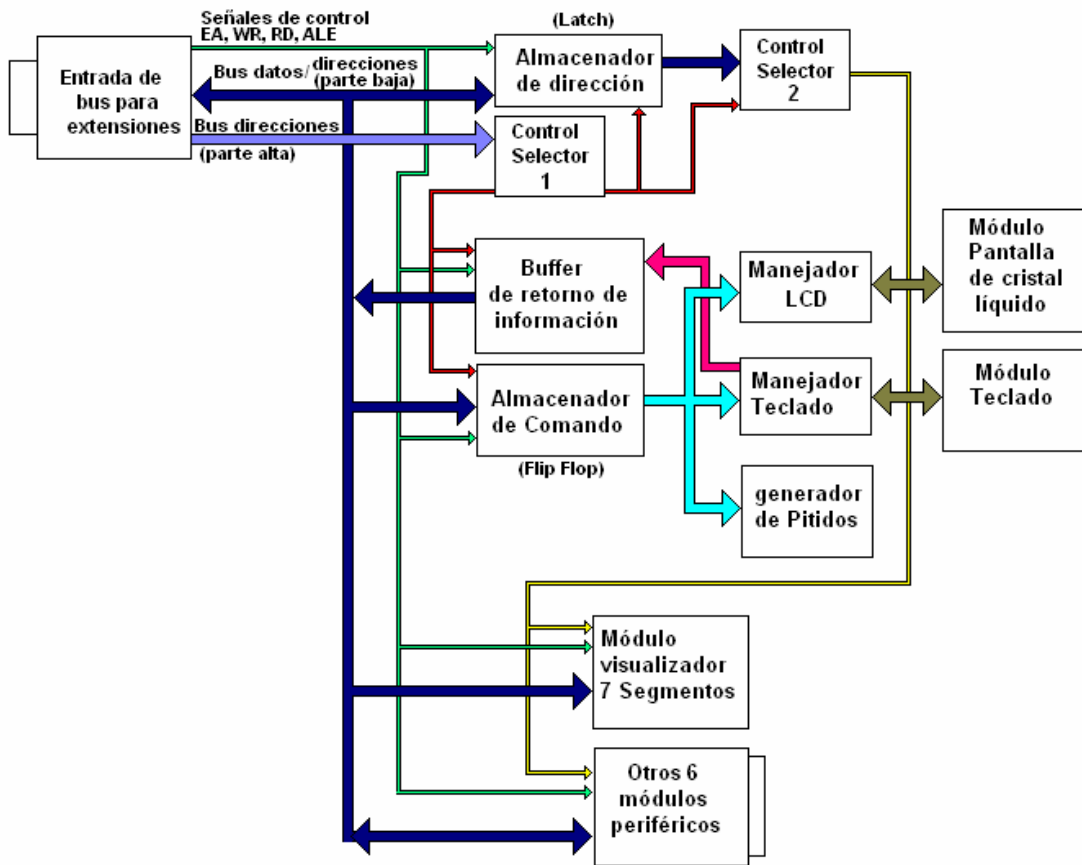


Figura 13. Esquema de disposición y señales que se encuentran en la tarjeta principal de interfaz.

5.1.2. LCD

El LCD preestablecido es un dispositivo alfanumérico de 2 líneas y 16 caracteres por líneas, sin embargo, viendo la configuración y estudiando los comandos basados en GemCore usados para manejarlo, se le pueden adaptar otros tamaños (como por ejemplo el de 20 caracteres 4 líneas) sin problema alguno, dado que mantienen la disposición y protocolo de control. En tal caso, lo único que se hace es soldar un peine de pines compatible con los conectores soldados en la Tarjeta Principal de Interfaz.

5.1.3. Teclado

El teclado fue diseñado de la típica forma de teclado matricial, de cuatro filas por cuatro columnas, de modo que por las filas se coloca un nivel lógico y se escrutan las columnas buscando que ese valor lógico se reproduzca. De cualquier manera, según las necesidades de pueden omitir teclas de no ser necesario. También este diseño cuenta en su parte posterior con un peine de pines compatible con el conector hembra dispuesto en la Tarjeta Principal de Interfaz. Igualmente se buscaron pulsadores rotulables para aumentar la posibilidad de configurar las funciones de las teclas.

5.1.4. Buzzer.

Este dispositivo es directamente controlado y manejado por el pin correspondiente al bit más significativo de la parte baja del bus. El código preestablecido del fabricante dentro del firmware GemCore se encarga de generar las frecuencias y duración de las señales de este dispositivo, según los parámetros indicados por el programador de la aplicación de usuario. Puede ser un buzzer genérico de PC, o un speaker.

5.1.5. Presentador 7 Segmentos.

Este dispositivo es un periférico adicional y sirve de modelo para cualquier otro dispositivo periférico direccionable vía paralelo. En tal caso, se creo un protocolo propio para recibir la información que se requiere mostrar. Cuenta con una inteligencia propia que le permite ser relativamente autónomo del resto del sistema, recibiendo los datos y realizando independientemente la visualización y retirada de los valores, siguiendo los comandos recibidos.

5.1.5.1. Circuito.

El circuito del presentador 7 segmentos se diseñó en base a la característica de recibir valores y comandos vía paralelo y no distraer (del sistema de comando central) capacidad de procesamiento en la generación de las cifras y valores de cada 7 segmentos. Bajo tal premisa, se consideraron diversas topologías de circuitos

digitales y maquinas de estado con compuertas u otros componentes lógicos. Sin embargo, en última instancia, dado el alto costo y la magnitud del tamaño físico que el circuito podría alcanzar, se optó por utilizar un microcontrolador.

El microcontrolador seleccionado fue elegido por su capacidad de adquirir datos (por interrupción) desde el bus paralelo sin demasiado esfuerzo de programación, a través de una característica de los PIC de microchip, llamada Parallel Slave Port, en el cual viene pre-implementado dicha comunicación. Observando la relación coste-eficiencia, se decidió utilizar el 16F877A, que cuenta con esta característica, de buen precio y alta fiabilidad. En tal sentido, se decidió utilizar una configuración que permitiese hasta 4 dígitos 7 segmentos ser manejados, aunque se proveyó el sistema prototipo de sólo 3 (y es factible reducir la cantidad de ellos solo dejando sin soldar tales componentes). Se seleccionaron dígitos 7 segmentos cátodo común, de modo que la mayor parte de la energía confluya en un único canal de control, lo que permite controlar los niveles de potencia por medio de máximo 4 transistores.

Es así como la señal de cada ánodo de cada uno de los dígitos se encuentra conectada a una señal común saliente del microcontrolador PIC directamente. Todas estas señales desembocan al cátodo común el cual va al colector de un transistor por medio de una resistencia y cuya base va a un pin de control que puede encender o anular el dígito en cierto momento. Del emisor va a referencia. De esta manera se controla la cantidad de corriente que entra al pin de control al éste colocarse en nivel bajo y activar al dígito.

El microcontrolador utiliza adicionalmente un oscilador de 4MHz como señal de reloj y provee control hasta de cuatro leds con su respectiva resistencia limitadora. Este sistema permite realizar las funciones de multiplexado en el tiempo de la presencia de imagen en cada led, para asegurar la eficiencia del sistema, así como permite codificar hasta 6 caracteres adicionales a los 10 números básicos, tan solo cambiando cierta codificación de su programación interna. Un diagrama de bloques que muestra esquemáticamente el funcionamiento del módulo se presenta en la figura 14.

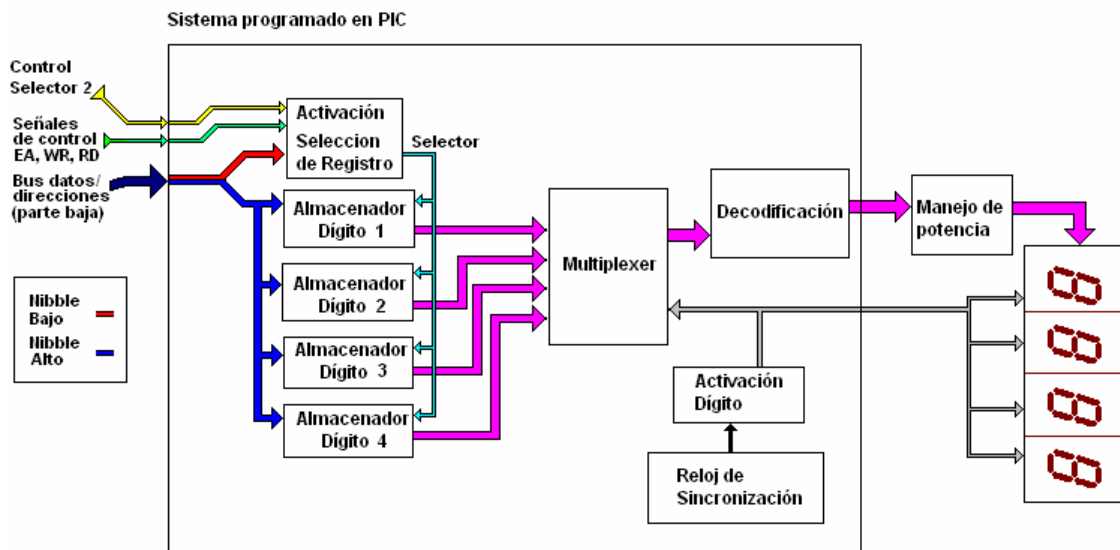


Figura 14. Esquema del funcionamiento del sistema de visualización en 7 segmentos. De manera esquemática, se presenta el proceso llevado a cabo por el microcontrolador.

5.1.5.2. Programación interna.

El programa interno desarrollado para controlar este sistema se basa en los principios presentados en la ilustración de la figura 14. Básicamente, se diseñó el sistema de interfaz de modo que se envíen, en formato byte a través del Parallel Slave Port ciertos números con significado propio para el sistema de manejo de 7 segmentos. Se tomaron los números 0x22 como código de autorización de arranque y el 0x56 como comando para limpiar el presentador (apagarlo). Cuando hay un dato para mostrar, se debe recibir en el siguiente formato:

- El nibble bajo indica la posición en que se va a mostrar una cifra. Si es 0x00 va en el dígito 7 segmentos ubicado en la posición más significativa (unidades de mil). Si es 0x04, la cifra va en el dígito que ocupa las centenas, si es 0x08, en el dígito que representa las decenas, y por último, si es 0x0C (el código es en hexadecimal) se refiere a las unidades. Si el nibble bajo no contiene ninguna de estas cuatro cifras, el sistema verifica si es un comando o es un valor desconocido.

- El nibble alto viene el valor o cifra a mostrar en la posición indicada por el nibble bajo que lo acompaña. De esta manera, al querer mostrarse un valor deben recibirse las 4 posiciones consecutivamente, para poder completar la cifra.

Por ejemplo, si se desea mostrar el número 0045 en el presentador deben recibirse los siguientes números hexadecimales: 0x00, 0x04, 0x48, 0x5C. Esto inmediatamente colocará en pantalla la cifra deseada, debiéndose recibir, en el tiempo considerado como prudencial, el valor 0x56 que indicara que se puede borrar el número y colocar el presentador en estado de reposo.

Otra particularidad del programa desarrollado es que al encender el sistema completo, el módulo de manejo de 7 segmentos aguarda la señal 0x22, que es un código de autorización para iniciar, lo que permite ejecutar una pequeña rutina de prueba que enciende en secuencia todos los leds presentes y permiten al usuario verificar que ninguna ha sido dañado, pasando luego a estado de espera.

La codificación de los segmentos ha sido realizada por medio de un arreglo, donde la cifra recibida apunta a una posición de ese arreglo, donde esta almacenado un valor coincidente con la distribución de puerto que enciende los led adecuados. Dada la estrategia de transmisión de data, los caracteres a mostrar son cifras de 0 al 9, quedando además de la A a la F libres para codificar otros signos para mostrar, uno de ellos es el elemento vacío, que mantiene todos los leds apagados. Adicionalmente, tiene un elemento de programación que permite seguir operando y no repetir la secuencia de diagnostico si se ha reiniciado automáticamente si la ejecución falla. Un diagrama de flujo del programa interno se muestra en el anexo 2, Ilustración 2 de los anexos.

5.2. Programación Módulo GemCore 500-MS4

Una vez dimensionada y desarrollada la Tarjeta Principal de Interfaz, se procede a desarrollar la programación del módulo gemCore500, basándose en el ambiente de programación C Keil CA51 y las aplicaciones Gemplus Builder/Loader que permiten crear el proyecto, compilarlo y descargarlo. Según el fabricante, la

programación se puede realizar incluyendo (como archivos de cabecera) ciertas librerías desarrolladas por el fabricante, en las que internamente se programaron rutinas que se convierten en sentencias (o instrucciones) con las cuales se hace interfaz entre el código escrito y la sentencia generadora de comandos basados en GemCore.

Es decir, se programaron funciones en las cuales, realizando su llamado y pasando los argumentos adecuados, se encargaba internamente de realizar todos los procesos necesarios. Sin embargo, no se poseían ciertos archivos claves que permitían de esta forma facilitar el trabajo. Por ello, después de una larga espera, sin obtener respuesta del fabricante, se decidió utilizar la sentencia generadora de comandos basados en Gemcore, y desarrollar la programación creando archivos de cabecera propios que fungiesen de librerías para sobre ellos, generar el código de aplicación.

En esencia, la sentencia generadora de comandos basados en GemCore es una función, que vinculada a una librería básica (archivo *.lib) que al compilar, genera simplemente los comandos basados en Gemcore, más su respectivo encapsulado en formato de bloques. Esta sentencia tiene la siguiente sintaxis:

```
Char sysct_exec(  
    char *buffer,  
    char Narg,  
    char presp,  
    char CodOp,  
    Word timeot  
)
```

Donde *buffer* apunta a un arreglo donde se colocan los demás parámetros de un comando basado de GemCore, y adicionalmente, donde se almacena la respuesta; *Narg* es una variable cuyo numero indica el numero de elementos (parámetros) que se encuentran en el arreglo *buffer*; *presp* indica cual es el máximo disponible de espacio para recibir data; *CodOp* es el código de operación, tomado del comando basado en GemCore. Es fácil deducir que por medio de esta instrucción SYSCT_EXEC, se está realizando la creación del comando por bloques para el firmware GemCore. Tomando la lista de comandos basados en GemCore del manual

del fabricante (y que se encuentran en el anexo 1) se procedió a programar funciones generales, incluidas y ordenadas que permitiesen realizar las labores requeridas. De esta manera se desarrollaron funciones y procedimientos de Lectura de tarjeta inteligente, escritura en LCD, lectura de teclado, generación de pitidos, y dichas funciones se incluyeron en archivos, pensadas para que fueran incluida, y llamadas por un usuario programador como si fuesen funciones, operando en su interior el SYSCT_EXEC. El proceso se ejemplifica en la figura 15

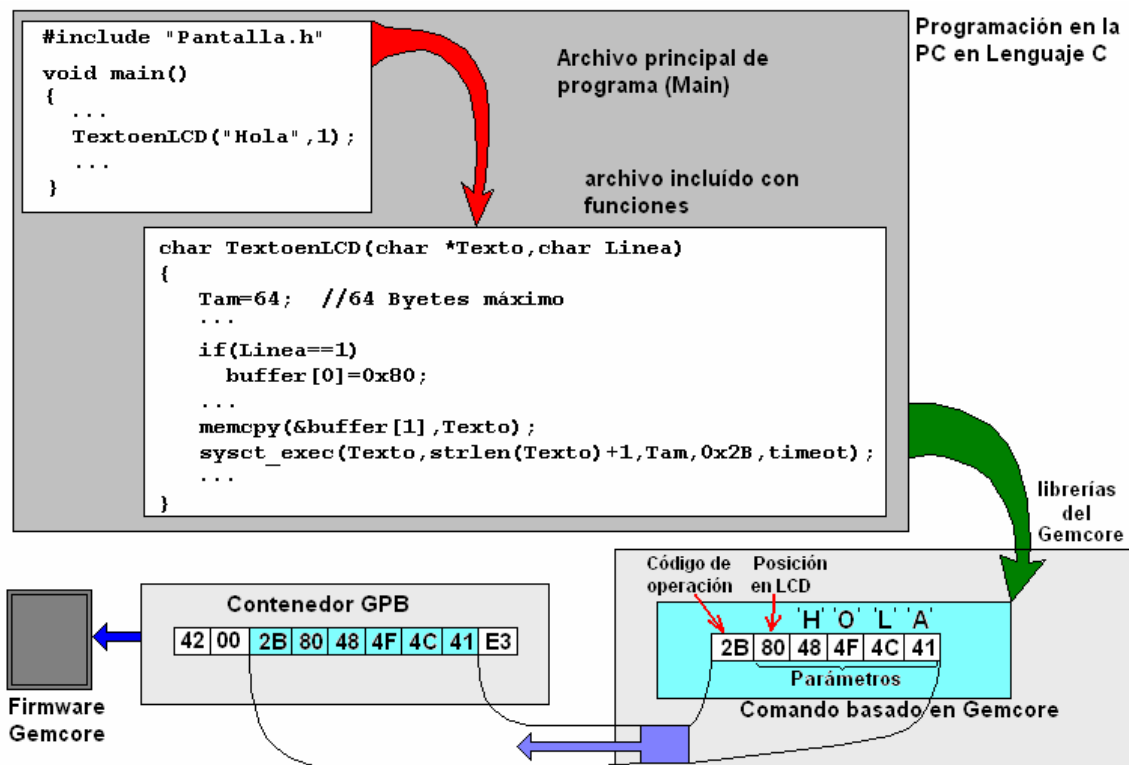


Figura 15. Esquema que ilustra el funcionamiento de la interacción entre capas de programación del GemCore500. Se observa un programa principal (obra del desarrollador de aplicaciones) que interacciona capa por capa hasta la ejecución de la tarea

De tal manera se realizó el desarrollo para un primer prototipo. Sin embargo, la culminación satisfactoria del sistema de esta manera estaba condicionada a conocer la totalidad de los comandos encapsulados en Gemcore. Desafortunadamente, el manual obtenido no contenía los comandos de recepción y envío de datos vía serial y vía I2C. Esto representaba un obstáculo importante para el desarrollo del resto de los

objetivos, pues se planteaba como necesidad la comunicación externa también vía serial (para agregar nuevos componentes o comunicación externa).

Igualmente, se trató de contactar al fabricante para solicitar mayor información sobre los comandos faltantes, y de nuevo, luego de un tiempo de espera no se obtuvo respuesta. Conociendo la arquitectura interna del 8051, se intentó de múltiples maneras acceder directamente los sistemas de envío y recepción serial, por medio de librerías que permitían utilizar directamente el conjunto de interrupciones, pero los resultados no fueron nada alentadores, dado que probablemente se estaba interfiriendo con el funcionamiento del Firmware Gemcore, al tratar de acceder directamente al 8051, dado que la aplicación fallaba y se congelaba la ejecución, siendo necesario el reencendido. Debido a esto, se decidió abandonar esta estrategia, y se decidió trasladar la inteligencia de programación de la memoria interna del módulo hacia un dispositivo externo que comande al módulo emulando a un PC que envía los comandos vía serial.

De esta manera, emulando la producción de comandos en bloques GBP, y enviándolos al módulo Gemcore500, se le utiliza como esclavo (que a su vez maneja a la tarjeta Principal de Interfaz). De esta manera, se ejecutan en el módulo GemCore los comandos que efectivamente se conocen (enviados desde un control externo) y el resto de las acciones la ejecuta el elemento de inteligencia externo. Un esquema que ilustra esta estrategia se aprecia en la figura 16. Esta estrategia implicó el desarrollo de una tarjeta adicional donde se encuentra físicamente el ente de control externo. Esta nueva tarjeta de circuito impreso, se le denominó Tarjeta General de Comando, pues en ella reside la inteligencia del sistema.

5.3. Tarjeta General de Comando

Esta tarjeta de circuito impreso está compuesta por el sistema de comando y registro de transacciones. En primer lugar, se decidió distribuir el control del sistema en dos subsistemas, uno más pequeño, encargado del control directo y exclusivo del módulo, y otro a nivel más global que controla al primer control y a otros subsistemas que puedan ser agregados.

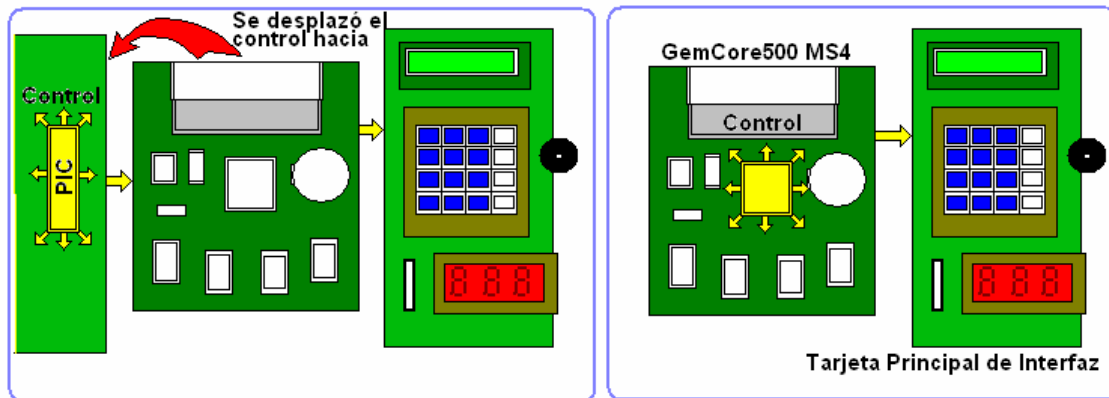


Figura 16. Esquema que ilustra el cambio decidido de orientación del proyecto. A la derecha, la lógica original, donde un programa de usuario se descargaba en la memoria CODE del Gemcore500. Al la izquierda, la estrategia aplicada, desplazar el control a un dispositivo PIC que funge de Host

5.3.1. Control de interfaz.

El subsistema de control directo de GemCore tiene la implementación del protocolo de comunicaciones por bloques comentado previamente. Se realizó el desarrollo del algoritmo de programación de dicho protocolo basándose en un análisis de la comunicación con un programa de demostración de la PC conectado al módulo GemCore y con información extraída de los manuales disponibles. El desarrollo se hizo creando un archivo de código C básico donde se implementa la encapsulación y generación del protocolo de comunicaciones, así como la recepción de y gestión de respuestas.

Este archivo maneja el puerto serial en esquema TTL, comunicándose por medio de un cable plano de 8 líneas con el conector K4 del módulo GemCore. Este archivo (donde se implemente el algoritmo GPB y al cual se le denominó “básico”), se incluye en el proyecto general de programa del compilador en C y pasa a ser un procedimiento similar al `sysct_exec`, mencionado en el ítem anterior. Por medio de esta función, se delega totalmente en él la responsabilidad de realizar la comunicación, solo recibiendo por paso de parámetros el arreglo de comandos basados en GemCore. Adicionalmente, se creó otro archivo de herramientas, donde

en base a la función de gestión de comunicación, se desarrollan procedimientos que se encargan a su vez de realizar las tareas comunes (como escritura en LCD, comunicación con tarjeta, lectura de teclado, etc.).

De esta manera, a su vez, se delega en estas funciones el conocimiento y uso de los comandos basados en Gemcore, siendo estos gestionados por estas funciones “herramientas” que a su vez se basan en la comunicación implementada por la función “básica”. Luego, en tercer lugar, se desarrollo un archivo principal, donde se programó el inicio y arranque del módulo GemCore, así como el estado de espera. Adicionalmente, se desarrolló el programa de manera que los mensajes que se envíen a un sistema comandante superior, sean previamente transformados en cadenas de caracteres ASCII. De esta manera, éste ente de control superior tendría funciones de comunicación más sencillas basadas en cadenas de caracteres. Una vez tomado todos estos elementos en cuenta, se decidió utilizar el PIC16F877. La selección estuvo influenciada por los siguientes aspectos:

El PIC16F877A es un modelo muy versátil, de amplias posibilidades (Puertos, comunicaciones), económico, fácil de obtener en los proveedores locales, y de amplia utilización en la industria. Adicionalmente tiene la capacidad de memoria adecuada al cometido designado. La plataforma de programación, compilación y descarga de programa en PIC seleccionada PIC C CCS®, dada su interfaz amigable, fácil de aprender y comprender y su gran cantidad de herramientas pre-diseñadas. Esto ofrece la ventaja de que cualquier desarrollador puede aprender rápidamente esta herramienta y podría programar actualizaciones y/o modificaciones sin un arduo proceso de aprendizaje.

Una vez seleccionados estos aspectos, se procedió al desarrollo general de la plataforma de programa descrita, con la filosofía que constituiría en un intermediario que traduciría y gestionaría el módulo independientemente para que otro sistema se libere de tal obligación. En tal sentido, este desarrollo solo recibiría comandos en forma de cadena de caracteres y respondería de la misma manera, asegurándose de realizar las tareas de manera satisfactoria. Un diagrama de flujo del programa interno

creado se aprecia en la Ilustraciones 4 a la 6 del anexo 2. Esto implicaba que el sistema desarrollado hasta el momento quedaría como lo ilustrado en la figura 17

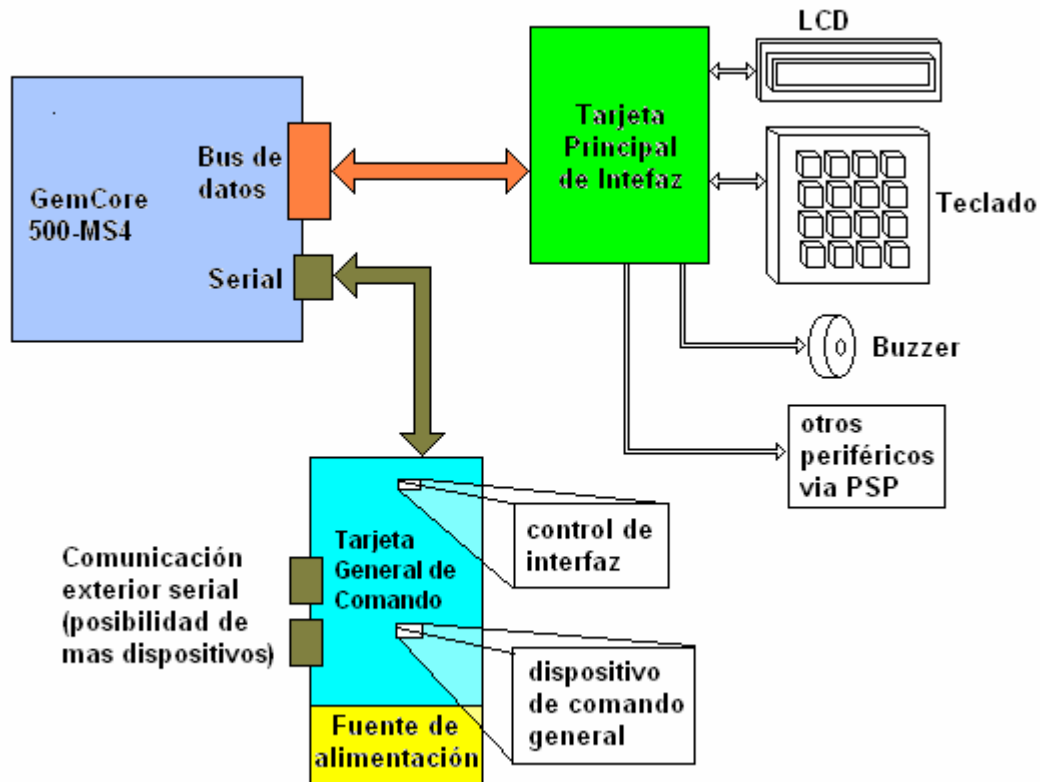


Figura 17. Esquema de la conformación general de los elementos que conforman el prototipo del lector desarrollado

Cabe destacar que el PIC tiene posibilidad de manejar varios puertos seriales a la vez, por lo cual, se designó un segundo puerto serial como comunicación entre este módulo de control de interfaz y el módulo de comando general.

5.3.2. Módulo de comando General.

Este dispositivo, debería tener características superiores al módulo de control de interfaz, ya que esta pensado para tener una aplicación más completa que, por medio de una comunicación serial con el módulo de control de interfaz (donde solo se le tenga que informar las tareas y delegue su ejecución) y otros periféricos, se constituya en la verdadera aplicación de usuario. De esta manera, este módulo no tendría que desviar memoria ni capacidad de procesamiento en escrutar los periféricos del

módulo Gemcore, ni en encapsular o gestionar comunicación. De todo esto ya se encargaría continuamente el módulo de interfaz, generando una interrupción al módulo de comando, cuando ocurra un evento relevante.

Sin embargo, dado que este es el módulo que va a contener la aplicación de usuario, quien es libre de programar, tan solo siguiendo unas pocas limitaciones en su comunicación, se requiere de mayor memoria de programa y RAM para asegurar la comodidad de desarrollo. Es por ello que se decidió seleccionar el PIC 18F4550, que tiene una adecuada capacidad de memoria, además de otras características importantes. Este módulo, al comunicarse con el módulo de control de interfaz, debe enviar un carácter definido que relaciona la acción con algún elemento en particular (LCD, tarjeta, etc.) y seguidamente, una cadena de caracteres.

Adicionalmente, si algo ocurre en el teclado, o una tarjeta se introduce, es interrumpido para avisar tal eventualidad y realizar el manejo de datos y gestiones de tareas. Adicionalmente, se colocó una memoria EEPROM accesible por I2C, con su respectiva comunicación con este módulo (de 64KBytes) para asegurar un espacio de memoria no volátil (aparte de la memoria interna EEPROM del propio PIC de 256 bytes) donde almacenar datos importantes que se debe conservar (como nombres, parámetros de lector o transacciones). Dicha memoria fue elegida I2C (ATMEL 24C512) por la simplicidad de circuito y desarrollo que implica, además del hecho de ya disponerse de un archivo de código con la interfaz de programación ya realizada.

Por último, se incluyó un circuito max232, para permitir, por un segundo acceso serial del PIC de comando general, proveer una salida de comunicación en formato RS232 con periféricos totalmente externos al sistema general (tal como PC y otros dispositivos). Adicionalmente se incluyeron pines para garantizar la fácil conexión de alguna otra posibilidad en puertos que quedaron libres del 18F4550 (tales como la posibilidad de comunicación USB u otras comunicaciones seriales con dispositivos adicionales aún no contemplados. Una representación esquemática de la estructura de la tarjeta de comando general, se aprecia en la figura 18.

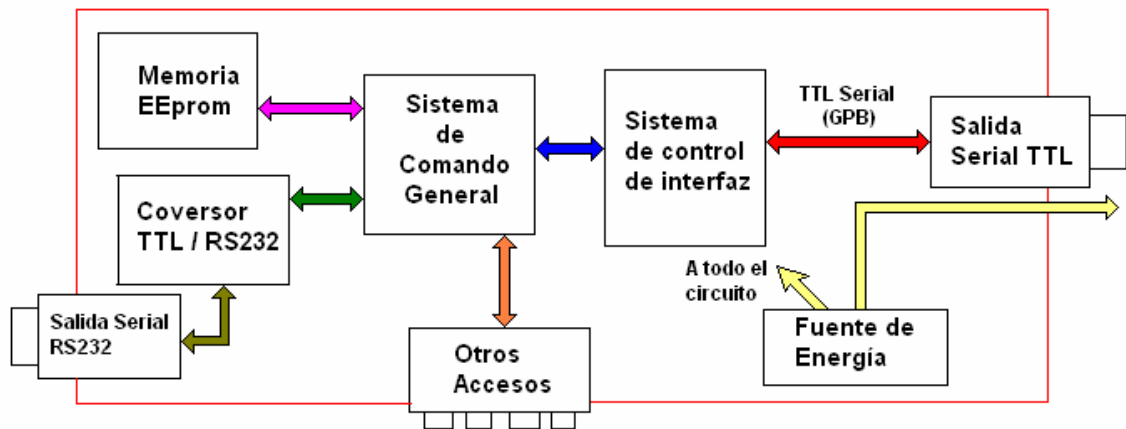


Figura 18. Esquema de la estructura en bloques de la Tarjeta de Comando General, con las señales que intercambian

5.3.3. Fuente de alimentación.

Un componente muy relevante del desarrollo general del problema es la alimentación de energía a todo el prototipo. Básicamente, todo el prototipo requiere una tensión de 5V (con 10% de tolerancia). Para conocer la potencia disipada, se buscaron diversos datos de consumo de los distintos componentes.

- Módulo GemCore: promedio 50mA, 150 con Tarjeta conectada.
- LCD: máximo 4mA.
- 7 segmentos: máximo 70 mA
- 2 PIC16F877A: máximo 35mA c/u = 70
- 18LF4550: 15 uA
- Teclado: máximo 2mA
- 4 Leds: 40mA
- AT24C512: máximo 3 mA
- Integrados Lógicos 74HC: máximo $50\text{mA} \cdot 4 = 200\text{mA}$

Total = 554mA, considerando posibles pérdidas de 35% = 749,4mA

Sin embargo, considerando que pueden ser adicionados muchos otros elementos, se estimó que sería adecuado poder proveer hasta 3 veces ese valor máximo. En tal sentido, se dimensionó a una fuente que fuese capaz de proveer 2,5A. Para ello se consideró una amplia gama de posibilidades. Al final, tomando en cuenta la simplicidad del circuito, fiabilidad eficiencia (del 77%), disponibilidad en los proveedores locales, y reducido número de componentes, se eligió al regulador switchado LM2576. El circuito empleado de observa en la figura 19.

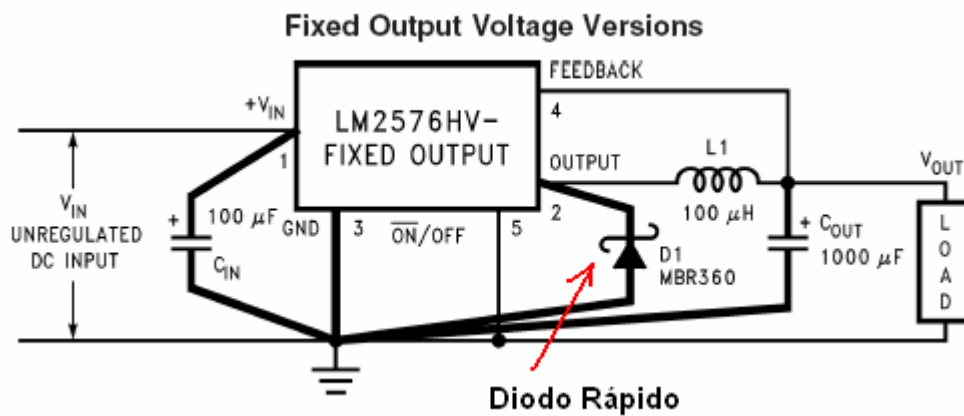


Figura 19. Circuito que se aplicó, tal como lo recomienda el fabricante, para la fuente de alimentación del circuito

Este circuito (el propuesto por el fabricante), como se observa en la figura, es bastante simple, lográndose una buena estabilidad (cerca del 5%) de la tensión de alimentación, y pudiendo recibir a la entrada un rango aceptable de tensiones DC no reguladas (acepta entradas de 8V a 40V), siendo importante la potencia que puede proveer el adaptador³ a utilizar. También se le podría adicionar un transformador y un rectificador, de modo que no requiriese de la adquisición de un adaptador. Sin embargo, se estimó que esta estrategia implicaría tener tensiones de 120V 60Hz

³ Adaptador se refiere a dispositivo adquirible en diversos comercios, generalmente pequeña caja plástica que contiene un transformador y circuito rectificador, del cual sale un cable con una tensión DC no regulada. En algunos casos, tiene un selector para elegir el valor DC.

presentes en el circuito, lo que podría conllevar a ciertos riesgos (fallas destructivas, electrocución del operador, etc.).

De la misma manera se incrementaría el peso, volumen y costo del módulo. Por ello se optó por el uso de adaptador de corriente externo, de los cuales se consiguen en el mercado abundantemente, con variedad de costos y capacidades. De hecho, se estima que para la mayoría de las aplicaciones, un adaptador de 4,5W (9V, 0,5A) es suficiente. Como elementos más importantes, el circuito utilizado requiere de una bobina de 100uH (Wilco W060, la cual soporta más de 2,5A) y un diodo rápido (Fast recovery) que en este caso es un zero recovery de la casa Cree (diodo modelo CSD04060).

CAPITULO VI

6. RESULTADOS

Tal como se describe en la sección de programas y diagramas de flujos, primeramente se realizo la programación básica del Módulo de control de Interfaz. Esta programación implica inicializar la interfaz de usuario, presentar continuamente la hora, fecha y un mensaje en el LCD. Adicionalmente, por vía serial recibe parámetros para convertir esos parámetros en acciones concretas, por medio de una comunicación basada en cadenas de caracteres ASCII.

Esta característica desarrollada, facilita enormemente el diseño de los desarrolladores de aplicaciones. Es posible comprobar la funcionalidad de este sistema encendiendo el prototipo, si la presencia del dispositivo de Comando General. En el caso del módulo de comando general, se realizó la programación de un algoritmo adicional para descargarlo en el Módulo de Comando General. Este programa, constituye una aplicación Demostrativa, que permite al usuario comprobar la operación de todos los componentes periféricos desarrollados, bajo el comando de este módulo.

De tal manera, el programa demo realiza tareas que comprueban la correcta operación y accesibilidad a todos los elementos desarrollados. Dicha aplicación verifica e informa que se ha pulsado una tecla del teclado, así como se encarga de especificar en el LCD cuál se ha pulsado. Esto lo hace continuamente, demostrando la funcionalidad tanto del LCD, como del teclado, así como del buzzer, que efectúa sonidos de pitido cada vez que una tecla es pulsada. Todas estas tareas son efectuadas por medio de la simple indicación en forma de cadena de caracteres enviada al módulo de control de interfaz vía serial en el momento oportuno, luego, éste ultimo se encarga de realizar la tarea. Típicamente el proceso se inicia con el envío de un carácter (que especifica simultáneamente un llamado al módulo de control de interfaz y una indicación de cual labor se desea ejecutar), a la cual el módulo de interfaz

contesta con una indicación de disponibilidad y entonces se produce el envío del resto de los parámetros.

Otras tareas produce un efecto al contrario (ejemplo: al pulsar una tecla especial o insertar tarjeta). A partir de un evento externo provocado por el usuario sobre el módulo de control de interfaz, éste es capaz de informarle al módulo de comando que posee algo para él, y al recibir la autorización, se produce el envío de la data recibida. Adicionalmente, Se dispone de un puerto RS232, por medio del cual es posible que esta aplicación demo reenvíe esta data al PC o que permitan realizar tareas comandadas desde una PC. Tales tareas ordenadas externamente pueden ser, tanto enviar una cifra y mostrarla en el visualizador 7 segmentos como enviar la orden de limpiar el visualizador 7 segmentos y colocarlo en espera; requerir inserción de tarjeta inteligente, mostrar mensaje en LCD, reproducir sonido o requerir solicitud de inserción de cifras. Esencialmente, son las mismas labores que el módulo de comando esta en capacidad de requerir, solo que estas son solicitadas eventualmente desde el exterior (por ejemplo desde una PC).

Otra manera de que el módulo de comando interrumpa el estado de espera (mostrado de mensaje y hora/fecha) es pulsando la tecla con el logo de una persona (verde) lo cual hace acceder a un menú donde las opciones posibles se encuentran de la 1 a la 7 y se ejecuta pulsando el botón rotulado con un signo de “Visto” (Check mark). El efecto de cada opción debe ser conocido por el usuario (usualmente se asume que el usuario se encuentra al corriente del funcionamiento). De esta manera:

- Pulsando 1, se muestra un mensaje predeterminado en pantalla LCD que sólo desaparece al presionarse Cancelar (tecla con flecha Roja)
- Pulsando 2, se solicita la inserción de una tarjeta inteligente, la cual al ser insertada, lee el serial de la misma y lo devuelve por Pantalla LCD y por el puerto RS232. Adicionalmente realiza una transformación de los tres últimos números del serial y los presenta en el visualizador 7 segmentos por unos instantes al extraer la tarjeta.
- Pulsando 3, se accede a una rutina donde se solicita un dato (número byte) y una posición de memoria y se almacena dicho dato en la posición de memoria

especificada. Esta memoria corresponde con la memoria EEPROM, por lo que el dato estará disponible aun cuando se reinicie el equipo.

- Pulsando 4, se solicita una cifra, que al terminar de introducirla, se presenta en el visualizador 7 segmentos
- Pulsando 5, se solicita una posición de memoria EEPROM y de ella se obtiene el dato almacenado, que se muestra en LCD. Es un claro complemento de la opción 3.
- Con la opción 6, se solicita un número que corresponde con el código para actualizar la hora y fecha. Se introduce una cadena de números que cumplen con el patrón AAMMDDHHMMSS (Año, Mes, Día, Hora militar 24h, Minuto, Segundo). A culminar el proceso, automáticamente se mostrará el cambio hacia la nueva/hora fecha especificada.
- Pulsando la opción 7, se reproduce un sonido preestablecido en el Buzzer. Cabe destacar que la programación permite corregir la opción, y ésta se considera aceptada al presionar la tecla con la marca de visto (check mark en ingles).

Adicionalmente, Introduciendo la tarjeta inteligente en cualquier momento, (mientras la ejecución del programa se encuentra en estado de espera) se produce su lectura, extrayendo el serial de la misma y lo devuelve por Pantalla LCD y por el puerto RS232. Adicionalmente realiza una transformación de los tres últimos números del serial y los presenta en el visualizador 7 segmentos por unos instantes al extraer la tarjeta. Básicamente, se realizó este programa demostrativo, que basado en la comunicación establecida entre módulo de interfaz y módulo de comando, sirve de modo de ensayo a las capacidades especificadas. De esta manera, cuando el prototipo vaya a ser utilizado comercialmente, los desarrolladores tendrán en cuenta estas características, y realizarán programaciones de este módulo, que adecuado a la solución que busquen ofrecer, realizará el manejo de cada uno de los elementos periféricos para ejecutar las tareas adecuadas.

Por último, se realizó una programación (también demostrativa, en Visual Basic 6) de una interfaz que permite a la PC, comandar ciertas tareas directamente, tal

como se ha referido anteriormente que es posible hacer. Las tarjetas y el respectivo cableado fueron montados en una armazón de PVC temporal, que le da cuerpo al sistema, el cual aún no dispone de diseño de carcasa o estructura que recubra y proteja los circuitos. Sin embargo, el diseño se realizó tomando en cuenta la disposición ideada en el desarrollo de modo que las tarjetas se encuentren dispuestas verticales, una tras otras, con una separación aproximada de 1 cm. La estructura por capas obtenida se presenta en la Figura 20 y una fotografía del prototipo se muestra en la Figura 22. En la Figura 23 se muestra la pantalla del programa de visual Basic desarrollado para interactuar con la aplicación demo. Cada una de las casillas emula la misma opción que se puede ejecutar desde el menú del propio lector, solo

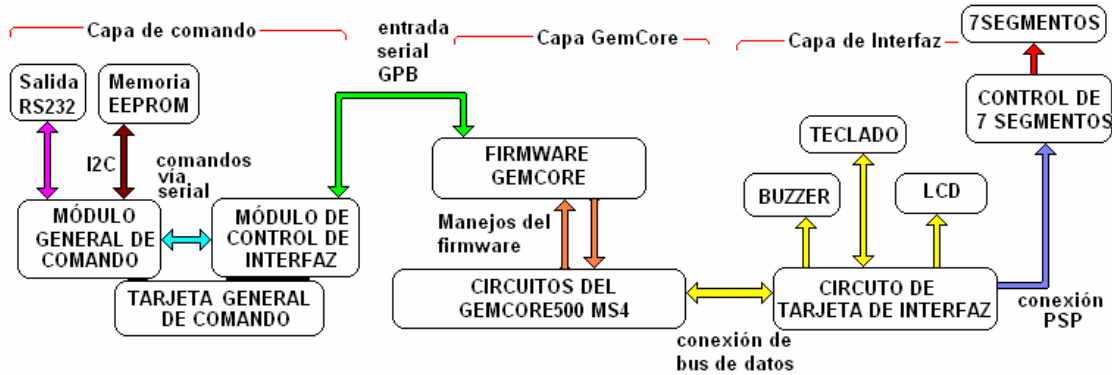


Figura 20. Esquema del prototipo, visto por las distintas capas que lo componen que ordenado desde la PC.

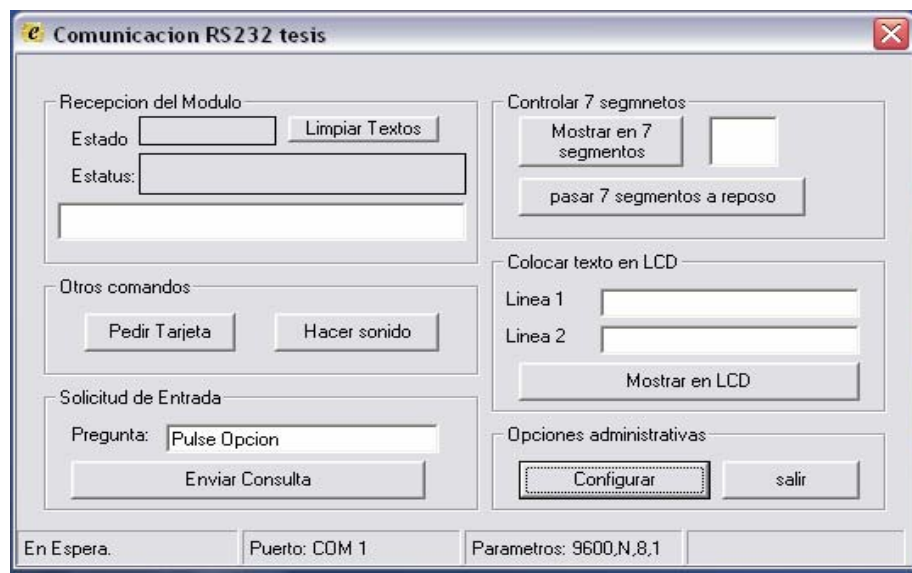


Figura 21. Imagen de la pantalla del programa realizado en Visual Basic para comandar el lector externamente



Figura 22. Fotografías del prototipo culminado, tanto de las tarjetas componentes por partes (foto extremo superior), de la armazón temporal del prototipo (Foto superior Izquierda, sobre estas líneas) y del prototipo ensamblado y operando (Foto derecha)

CONCLUSIONES

Una vez concluido la totalidad del desarrollo y luego de un análisis detallado de la teoría estudiada, se pueden obtener cierta cantidad de conclusiones que provienen de los diferentes ámbitos que acometió el presente trabajo.

En primer lugar, se apreció que la normativa y teoría que circunscribe a las tarjetas inteligentes, los lectores y sistemas de gestión de tarjetas inteligentes, se encuentra en continua revisión y actualización, evidentemente influidas por la tendencia de cambios en cuanto a avances de la tecnología, minituarización de componentes y aparición de nuevas necesidades y posibilidades. Asimismo, dada las características principales del prototipo realizado, que son simplicidad, economía, modularidad y aun muchas posibilidades (totalmente abiertas) sin explotar, se aprecia que se encuentra en armonía con esta tendencia teórica al cambio, permitiendo a la Empresa E-Smart Consulting, insertar nuevos componentes y subsistemas, rediseñar secciones, sustituir módulos, etc. Todo esto, garantiza la perfecta armonización del sistema a los patrones cambiantes, tanto de la tecnología como de los posibles escenarios de aplicación y utilidad.

En segundo lugar, Se evidenció la facilidad de desarrollo general, pues el prototipo fue montado con las técnicas aprendidas en el Laboratorio de Desarrollo de la Escuela de Ingeniería Eléctrica UCV y en la asignatura Diseño de Proyectos, lo que garantiza su aún más fácil producción industrial en serie. Los componentes seleccionados se pueden adquirir fácilmente en expendios de componentes electrónicos locales o con mayores ventajas económicas a proveedores en línea (web).

En tercer lugar, el cambio de orientación de la estrategia de diseño (es decir, la decisión de ubicar el control central en varios dispositivos PICs), provee varias ventajas adicionales que no se habían previsto inicialmente. Estas ventajas se circunscriben a tres ámbitos, que son Programación, Economía e Independencia. En cuanto a la programación, la ventaja es que se conoce a fondo el sistema de

programación del PIC (o cualquier otro microcontrolador a usar) y no se encuentra el futuro desarrollador limitado a los estilos y algoritmos impuestos por el fabricante del módulo GemCore. En lugar de ello, con solo respetar el protocolo GBP (de lo cual se puede encargar una rutina de ese microcontrolador), el futuro desarrollador se encuentra en plena libertad de elegir el lenguaje de programación, el compilador y la manera de descargarlo. Como la comunicación es serial, cualquier microcontrolador o lenguaje están capacitados de llevar a cabo esta comunicación con el módulo GemCore, tomando en cuenta el formato GBP. Esto potencia aun más el proyecto, pues le da posibilidades casi infinitas de expansión, actualización y modificación. En el ámbito económico, le da una perspectiva aún más ventajosa, pues el módulo GemCore dispone de elementos que están diseñados para descargar almacenar y correr un programa en su interior. Con la estrategia elegida, de control externo, esta capacidad queda ociosa. Entonces, en los catálogos del fabricante GemPlus se halló un producto similar, con todos los protocolos y disposiciones idénticos, sólo que sin las memorias y otros elementos de almacenaje de aplicaciones (es decir la sección que queda ociosa) y cuyo costo es 1/5 del Modelo GemCore 500. En Resumen, se podría sustituir al GemCore 500 por este otro modelo (GemCore410) sin ningún cambio en la programación ni circuital (excepto en la adición de un RTC) y se reducen los costos a 1/5 del total inicial. En el ámbito de la independencia, esta estrategia adoptada permite a E-Smar Consulting la capacidad de ser casi totalmente autónoma, siendo capaz de cambiar el sistema de dos módulos de control, por varios más pequeños, por uno más poderoso, o diversas alternativas. Es decir, siempre que se garantice una señal serial cumpliendo con el protocolo de comunicaciones GBP, podría E-Smar Consulting modificar o potenciar el proyecto a su antojo.

Todo esto, permite concluir que el desarrollo del sistema de gestión de tarjetas obtenido, aunque ya es una base funcional, puede dar lugar a una dirección nueva para las políticas de la empresa y dar inicio a múltiples desarrollos futuros. Dado todas estas reflexiones, es factible afirmar que el proyecto ha sido satisfactorio y ha concluido con resultados bastante alentadores con respecto a las necesidades y problemas planteados en un primer momento

BIBLIOGRAFIA

Libros

Rankl, Wolfgang y Effing, Wolfgang. Smart Card Handbook, Inglaterra: Editorial John Wiley & Sons Ltd, 2003.

Jurgensen, Timothy y Guthery, Scott. Smart Cards: The Developer's Toolkit (Essential Guide), EE.UU: Editorial Prentice Hall PTR, 2002.

Ferrari, Jorge; Mackinnon, Robert; Poh, Susan; Yatawara, Lakshman. Smart Cards: A Case Study. EE.UU: Editado por International Technical Support Organization & IBM. 1988

Manuales

Datasheet Philips P83C524. Sin Version. Philips Semiconductors. Holanda, 1997. 76 Pags

GCI410/GCI400-MS Development Board Reference Manual. Version 1.0. Gemplus. Cerdex France. 1999. 19 pags

GemCore 410 Technical Specifications. Version 2.0. Gemplus. Cerdex France. 2002. 12 pags

GemCore 500-MS4 Technical Specifications. version 2.0. Gemplus. Cerdex France. 2002. 20 pags

GemCore-Based Rader Interface Library Programmers Guide. Version 1.0. Gemplus. Cerdex France. 1997. 69 pags

GemCore Based Reader Reference Manual. Version 1.0. Gemplus. Cerdex France. 1998. 85 pags

GemCore Stand-Alone Application Programming Guide. Version 1.11. Gemplus. Cerdex France. 1999. 79 pags

GemCore Technical Specifications. Version 2.0. Gemplus. Cerdex France. 2000. 31 pags

OROS Reference Manual. Version 2.33. Gemplus. Cerdex France. 1997. 71 pags
PIC18F2455/2550/4455/4550 Data Sheet. Sin version. Microchips 2007. EE.UU. 2007. 430 Pags

PIC16F87XA Data Sheet. Sin version. Microchips 2003. EE.UU. 2007. 234 Pags

Paginas web

Cardwerk On-line

<http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-3.aspx>

(consulta agosto 2007)

GLOSARIO

ABS: Acrilonitrilo Butadieno Estireno o ABS es un plástico muy resistente al impacto (golpes) muy utilizado en automoción y otros usos tanto industriales como domésticos. Es un termoplástico amorfo. Se le llama plástico de ingeniería, debido a que es un plástico cuya elaboración y procesamiento es más complejo que los plásticos comunes, como son las polioleofinas (polietileno, polipropileno).

APDU: (En ingles Application Protocol Data Unit) es una unidad de comunicación entre un lector y una tarjeta inteligente, la cual está constituida como un contenedor de datos que tiene un formato específico para garantizar la correcta la comunicación. La estructura está definida por la norma ISO7816, de dos maneras distintas: Comando APDU y Respuesta APDU. El primero es el mensaje enviado del lector a la tarjeta, formado por 5 bytes obligatorios y un máximo de 255 bytes de datos opcionales. La respuesta es el mensaje de la tarjeta al lector con dos bytes (que forman una palabra de estatus) y hasta 255 bytes de datos.

API: Application Program Interface. Interfaz para la programación de aplicaciones. Representa una interfaz de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas librerías que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio

ASN: Abstract Syntax Notation 1 (notación de sintaxis abstracta nº 1) es una norma de representación de datos. Es una norma para representar datos independientemente de la máquina que se esté usando y sus formas de representación internas.

Bootloader: Se define como un código residente en memoria, que se activa inmediatamente al encender un equipo (o al reiniciarlo), realiza algunas tareas inaplazables, y da paso al siguiente proceso, generalmente un bootstrap. Entre ambos, constituyen el arranque de un equipo con cierto nivel de inteligencia.

Bootstrap: verdadero arranque del sistema que realiza todas las tareas de arranque, detección de dispositivos dependientes o periféricos y cargado de rutinas que se mantienen como interfaz entre los periféricos detectados y los comandos a recibir luego. De hecho, bootstrap significa “independiente”.

CODE.: Espacio de memoria, generalmente no volátil (o volátil respaldada por pila, o flash) que conserva, en lenguaje de máquina, los programas compilados y ensamblados de algún elemento electrónico programable. Cuenta generalmente con un apuntador que señala el código de operación que se va a ejecutar.

DUPLEX: en General, Capacidad de un dispositivo para operar de dos maneras. Particularmente en comunicaciones, indica la capacidad o incapacidad de realizar la Transmisión/Emisión simultáneamente.

EEPROM: Es un tipo de memoria ROM que puede ser programado, borrado y reprogramado eléctricamente, a diferencia de la EPROM que ha de borrarse mediante rayos ultravioletas. Aunque una EEPROM puede ser leída un número ilimitado de veces, sólo puede ser borrada y reprogramada entre 100.000 y 1.000.000 de veces. Conserva la característica de mantener sus datos intactos (por periodos de hasta 50 años) sin necesidad de fuente de alimentación, con la ventaja que no requiere de elementos externos (o extraerla del circuito), para modificar su contenido.

EPROM: Memoria ROM borrrable y programable (Erasable Programmable ROM). Se borran con luz ultravioleta (una exposición de unos 15-20 minutos). Cuenta con una cubierta de cuarzo al vacío de forma que el chip pueda ser alcanzado por las radiaciones ultravioletas. Mantiene la característica de preservar los datos luego de retirado el suministro de energía.

Firmware: Programación en Firme, es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria tipo ROM, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación. Funcionalmente, el firmware es el intermediario (interfaz) entre las órdenes externas que recibe el dispositivo y su electrónica, ya que es el encargado de controlar a ésta última para ejecutar correctamente dichas órdenes externas. Muchos de los Firmwares almacenados en ROM están protegidos por Derechos de Autor.

FLASH: es una forma desarrollada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación mediante impulsos eléctricos, frente a las anteriores que sólo permite escribir o borrar una única celda cada vez. Por ello, la memoria flash permite funcionar a velocidades muy superiores cuando los sistemas emplean lectura y escritura en diferentes puntos de esta memoria al mismo tiempo. Las memorias flash son de carácter no volátil (la información que almacena no se pierde en cuanto se desconecta la energía).

GSM: (Sistema Global para comunicaciones Móviles). Sistema de telefonía celular digital para comunicaciones móviles desarrollado en Europa con la colaboración de operadores, Administraciones Públicas y empresas. Estándar europeo que opera en las bandas de 900 y 1800 Mhz. Hace uso de las tarjetas inteligentes en forma de SIM

IDE: Siglas de Integrated Drive Electronics (electrónica de unidad integrada), que denota la interfaz estándar usada para conectar fundamentalmente unidades de disco y CD-ROM a un ordenador. Se basa en un cable de cinta plano de 40 hilos (generalmente Gris con uno de las líneas rojo o rosado) de formato de transmisión paralelo. Soporta típicamente hasta 66 MegaBits/s aunque se han alcanzado velocidades de 133 MBit/s.

IEC: La Comisión Electrotécnica Internacional es una organización de normalización en los campos eléctrico, electrónico y tecnologías relacionadas. Numerosas normas se desarrollan conjuntamente con la ISO. Fue Fundada en 1904, y en 1948 se trasladó a Ginebra.

ISO: Organización Internacional Para la estandarización. Fundada en 1946, es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional. También busca garantizar la compatibilidad de los elementos fabricados (de la misma naturaleza) en distintos lugares entre sí. La ISO es una red de los institutos de normas nacionales de 157 países, sobre la base de un miembro por el país, con una Secretaría Central en Ginebra, Suiza, que coordina el sistema. Las normas emitidas generalmente están referidas por las siglas, acompañados con un número.

Lenguajes Declarativos: En ciencias computacionales, Los lenguajes declarativos son aquellos lenguajes de programación en los cuales se le indica a la computadora que es lo que se desea obtener o que es lo que se está buscando.

OEM: Abreviatura del inglés Original Equipment Manufacturer, en español sería Fabricante de Equipos Originales. Sistemas vendidos al por mayor para que terceros ensamblen equipos de forma personalizada que utilicen este subsistema. Muchos fabricantes de hardware industrial ofrecen esta opción para que los clientes integren los equipos en sus propios envoltorios, cajas o sistemas.

PC: Personal Computer. En español más formal Ordenador de Escritorio, más frecuentemente Computador de escritorio. Todo dispositivo que tenga las características físicas de Monitor, Procesador Tarjeta madre y otros que permita correr sistemas operativos como Windows, Linux, Unix, Mac u otros y sobre ellos se ejecuten Programas y aplicaciones. Generalmente se asume que dispone de varios tipos de puertos (tanto de comunicación como conexión de dispositivos).

PIN: Número Personal de Identificación. Número secreto asociado a una persona o usuario de un servicio mediante el cual se accede al mismo. Se podría decir que es una contraseña numérica, que típicamente se acepta como de 4 cifras.

PVC: Cloruro de Polivinilo. Polímero orgánico obtenido por polimerización del cloruro de vinilo. Sus propiedades de resistencia a ácidos y bases, estabilidad y plasticidad hacen que sea uno de los productos más utilizados de la industria del plástico, con numerosas aplicaciones.

POS: Punto de Venta (o Punto de Servicio). Dispositivo electrónico integrado cuya función es recibir tarjetas de diversos tipos para leerlas y realizar gestiones financieras, principalmente pago de productos y servicios. Cuenta con una seguridad especialmente enfocada en su inviolabilidad. Esto se logra proveyendo al equipo de diseños que permitan que la violación (intento de modificación) se note claramente por medio de la inevitable destrucción parcial o total en caso de intentarlo.

RAM: Sigla de Random Access Memory (Memoria de acceso aleatorio). La RAM se usa para mantener los programas o sus variables mientras se están ejecutando, y los datos mientras se los procesa. La RAM es volátil, lo que significa que la información escrita en la RAM desaparecerá cuando se apague la alimentación de energía. Se dice que el acceso es aleatorio porque permite al usuario acceder a la información de manera aleatoria, es decir que se puede acceder a cualquier byte de la memoria sin tocar los bytes que lo preceden.

ROM: Memoria de sólo lectura. Memoria que contiene datos que no pueden ser modificados una vez grabados en ella. Después de programar guarda sus datos sin suministro de energía

SAM: un tipo de tarjeta ID 000 semejante al SIM que se utiliza generalmente para almacenar llaves criptográficas en las que se basa el principio de encriptación y seguridad

SIM: acrónimo de Subscriber Identity Module, ‘Módulo de Identificación del Suscriptor’. Es una tarjeta inteligente desmontable usada en teléfonos móviles que almacena de forma segura la clave de servicio del suscriptor usada para identificarse ante la red, de forma que sea posible cambiar la línea de un Terminal a otro simplemente cambiando la tarjeta. El uso de la tarjeta SIM es obligatorio en las redes GSM.

SQL: Lenguaje de consulta estructurado (SQL Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre la misma.

ANEXOS

ÍNDICE DE ANEXOS

ANEXO 1. LISTADO DE COMANDOS BASADOS EN GEMCORE	2
Sección de configuración	3
Comandos de Interfaz con Tarjeta.	4
Comandos de manejo de Memoria	7
Comandos de la Pantalla de Cristal Líquido	10
Comandos de Teclado y Buzzer	11
Comandos del Reloj de tiempo Real	12
 ANEXO 2. DIAGRAMAS DE FLUJO	 13
Programa Residente en Módulo Principal de Interfaz	14
Ilustración 1. Diagrama de flujo del procedimiento principal de espera	14
Ilustración 2. Diagrama de Flujo de la rutina básica que implementa el Protocolo de Comunicación	15
Ilustración 3. Diagramas de flujo (varios) de los respectivos procedimientos de herramientas	16
Ilustración 4. Diagrama de flujo de la rutina de "Herramientas" de manejo de teclado	17
Ilustración 5. Rutinas y procedimientos, que formando parte de "Herramientas" manejan 7 segmentos y Tarjeta Inteligente	18
Ilustración 6. Procedimiento que interpreta los comandos seriales, referenciado en la ilustración 1	19
Programa Residente en Módulo de Comando General	20
Ilustración 7. Ciclo Principal del programa residente en el dispositivo	20
Ilustración 8. Diagrama de flujo de la respuesta a la interrupción generada por el módulo Principal de interfaz	21
Ilustración 9. Diagrama de flujo del procedimiento de menú	22
Programa Residente en Visualizador 7 segmentos	23
Ilustración 10. Diagrama de Flujo del programa que reside en el control independiente que gobierna al visualizador	23
 ANEXO 3. PLANOS DE CIRCUITOS	 24
Ilustración 11. Esbozo de plano que describe los circuitos apreciados en el módulo Gemcore500	25
Ilustración 12. Plano de la Tarjeta de Interfaz	26
Ilustración 13. Plano de La tarjeta de comando general (donde reside la inteligencia del sistema)	27
Ilustración 14. Plano del Circuito del Teclado	28
Ilustración 15. Plano de la tarjeta del visualizador 7 segmentos	29

ANEXO 1

LISTADO DE COMANDOS BASADOS EN GEMCORE

COMANDOS BASADOS EN GEMCORE

Sección de configuración

Configurar Línea de Entrada/Salida Serial

Fija la configuración serial del módulo, como paridad, velocidad, y número de bits por carácter. Luego de encendido, por defecto no hay paridad, son 8 bits por carácter y 9600 baudios. Repetido dos veces se utiliza como inicialización.

Formato: **0x0A [BC]**

BC: Byte de Configuración.

BIT	Valor	Opción seleccionada
7 al 5	000	Sin uso
4	0	Sin paridad
	1	Paridad Par
3	0	8 bits por carácter
	1	7 bits por carácter
2 a 0	XXX	Fija la velocidad de transmisión
		000 Reservado para Uso Futuro
		001 76800
		010 38400
		011 19200
		100 9600
		101 4800
		110 2400
		111 1200

Fijar Modo

Este comando indica el modo de operación, con respecto al sistema operativo, el cual puede ser de tres maneras: Nativo, ROS y TLP (código OP = 00001001 = 0x09); sólo nativo y ROS (Código OP = 00001000 = 0x04); o sólo modo nativo (Código OP = 00000000 = 0x00)

Formato: **0x01 0x00 [OP]**

Retorna: **[S]** <Modo> [s] es Estatus.

Configurar Retardo

Si el elemento de control externo (originalmente pensado para PC) es más lento, se le puede definir al sistema que espere antes de dar la respuesta, para que el sistema de control externo no pierda la secuencia.

Formato: **0x23 0x01 0x00 0x67 0x01 [RE]**

Retrazo entre 0 y 255, por defecto 0.

Leer la versión del Firmware

Retorna en código ASCII la versión del firmware presente en el Módulo gemCore

Formato: **0x22 0x05 0x3F 0xE0 0x10**

Retorna: **[S]** <Versión>

Reiniciar

Reinicia el Sistema Operativo GemCore

Formato: **0x0C 0x00 0x00 0x00**

Reiniciar y correr aplicación específica

Al reiniciar, corre una aplicación especificada. Si no existiese, retornará mensaje de error

Formato: **0x0C 0x00 0x00 [AP]**

[AP] es el número de la aplicación

Retorna: **[s]** (0x03 si no existe)

Comandos de Interfaz con Tarjeta.

Apagado de la tarjeta

Este comando apaga la tarjeta que se encuentra insertada en el lugar seleccionado. Esto implica realizar el procedimiento de desconexión descrito en la norma ISO7816. Sin embargo la tarjeta es apagada automáticamente al retirarse. Puede usarse como detección de que la tarjeta ha sido extraída.

Formato (tarjeta Principal): **0x11**

Formato (tarjeta auxiliar seleccionada): **0x19**

[s] si no hay tarjeta presente se obtiene un 0xFB

Encendido de la tarjeta

Este comando enciende la tarjeta que se encuentra insertada en el lugar seleccionado. Esto implica realizar el procedimiento de encendido descrito en la norma ISO7816.

Formato (tarjeta Principal): **0x12 [CFG] <PTS>**

Formato (tarjeta auxiliar seleccionada): **0x1A [CFG] <PTS>**

Donde CFG son byte de configuración específica (Clase, Compatibilidad del sistema operativo interno, Manejo PTS) y los PTS adicionales

Salida ISO

Este comando envía comandos ISO que solicitan data que se encuentran en la tarjeta

Formato (tarjeta Principal): **0x13 [CLA] [INS] [A1] [A2] [LN]**

Formato (tarjeta auxiliar seleccionada): **0x1B [CLA] [INS] [A1] [A2] [LN]**

Que son las 5 cabeceras del protocolo ISO para gestión de tarjetas.

Retorna: **[s] <Data> [SW1] [SW2]**

Entrada ISO

Este comando envía comandos ISO que envían data hacia la tarjeta.

Formato (tarjeta Principal): **0x14 [CLA] [INS] [A1] [A2] [LN] <data>**

Formato (tarjeta auxiliar seleccionada): **0x1C [CLA] [INS] [A1] [A2] [LN] <data>**

Que son las 5 cabeceras del protocolo ISO para gestión de tarjetas.

Retorna: **[s] [SW1] [SW2]**

Intercambio APDU

Envía un comando APDU y espera la respuesta. La tarjeta debe soportar Protocolo T=1.

Formato (tarjeta Principal): **0x15 [APDU]**

Formato (tarjeta auxiliar seleccionada): **0x1D [APDU]**

El APDU debe cumplir con lo descrito en la sección 4 de la norma 7816

Retorna: **<Data> [SW1] [SW2]**

Define el tipo de la tarjeta Principal

El lector GemCore no tiene un algoritmo de reconocimiento del tipo de tarjeta inteligente. El tipo más frecuente viene precargado por defecto. Sin embargo, si se va a usar otro tipo, se debe especificar.

Formato: **0x17 [T] <00h [P]>**

Donde T es el byte de selección del tipo de tarjeta, especificado en la siguiente tabla

Código T	Tipo Correspondiente
0x01	Tarjetas sincrónicas
0x02	Tarjetas ISO7816-3 (T0= y T=1) de Velocidad estándar, frecuencia de reloj estándar (3,6864MHz)
0x12	Tarjetas ISO7816-3 (T0= y T=1) al doble de Velocidad con frecuencia de reloj 7,3728MHz
0x03	Tipo GPM256 (sólo lectura)
0x04	Tipo GPM416/GPM896 en modo estándar
0x14	Tipo GPM416/GPM896 en modo Personalizado
0x06	GFM2K/GFM4K
0x07	GPM103
0x08	GPM8K (SLE4418/4428)
0x09	GPM2K (SLE4432/4442 o PCB1032/2042)
0x0D	GPM276/GAM275
0x0E	GPM271/GAM273
0x0F	GAM226

Y el byte P define la identificación de presencia. Es opcional y se describe en la siguiente tabla

Código T	Tipo Correspondiente
0x00	En P1.6 (línea SCL) Tarjeta presente=1
0x01	En P1.6 (línea SCL) Tarjeta presente=0
0x02	En P3.1 (línea TxD) Tarjeta presente=1
0x03	En P3.1 (línea TxD) Tarjeta presente=0

Retorna: **[s]**

Define el tipo y selecciona tarjeta auxiliar

Idéntico al anterior pero aplicado a una tarjeta auxiliar

Formato: **0x1F [T] [N]**

Donde T es el código de la tabla anterior y N el número de la tarjeta auxiliar (de 1 a 8)

Retorna: **[s]**

Estatus de Tarjeta

Este comando es usado para obtener el estatus de la tarjeta principal o auxiliar que se encuentra conectada.

Formato (tarjeta Principal): **0x17**

Formato (tarjeta auxiliar seleccionada): **0x1F**

Retorna: **[s] [ES] [TI] [CNF1] [CNF2] [CNF3] [CNF4]**

Byte	Posición/Valor	Significado
ES	NNNNXXXX	NNNN es el numero en binario de la tarjeta seleccionada 0000 es 0x00 (tarjeta N° 0), 0001 = 0x01 (Tarjeta N° 1) y así
	XXXXXXXX0	Nivel de tensión 5V
	XXXXXXXX1	Nivel de tensión 3V
	XXXXXX0X	Tarjeta No encendida
	XXXXXX1X	Tarjeta Encendida
	XXXXX0XX	Tarjeta No insertada
	XXXXX1XX	Tarjeta insertada
	XXXX0XXX	Protocolo T=0
	XXXX1XXX	Protocolo T=1
TI	El tipo de tarjeta	
CNF1 CNF2 CNF3 CNF4	CNF1=TA1 (FI/DI) CNF2=TC1 (EGT) CNF3=WI CNF4=00	Tarjeta T=0 tal como lo especifica la norma 7816 sección 3
CNF1 CNF2 CNF3 CNF4	CNF1=TA1 (FI/DI) CNF2=TC1 (EGT) CNF3=IFSC CNF4=TB3 (BWI/CWI)	Tarjeta T=1 tal como lo especifica la norma 7816 sección 3
CNF1 CNF2 CNF3 CNF4	CNF1=00 CNF2=00 CNF3=00 CNF4=00	Tarjeta inteligente sincrónica

Directorio

Este comando se utiliza para obtener los tipos de tarjeta manejables, y el número de versión de cada controlador

Formato: **0x17 0x00**

Retorna: **[s] [TI] [CMD] [RV] ... [TI] [CMD] [RV]**

Donde [TI] es el tipo de tarjeta, CMD es 0x00 si es ISO Entrada/Salida, 0x01 si es APDU y 0x03 si es ambas simultáneamente; REV es la versión.

Comandos de manejo de Memoria

Leer Memoria

Lee el contenido de memoria de cualquiera de las áreas accesibles. De esta manera, áreas que no son memoria sino periféricos, también puede ser accedida.

Formato: **0x22 [Tip] <[Pag]> [DA] [DB] [LN]**

Donde [Tip] es el tipo de memoria leer entre IData (RAM Interna), XData(RAM externa) y CODE (Memoria de código). Además indica si se va a cambiar la sección de memoria actualmente en uso (página)⁴. De tal manera, se compone de la manera mostrada en la tabla.

Valor	Significado
0100XXXX	Se leerá por Páginas, indicado por el parámetro opcional [PAG] que se encuentra presente
0000XXXX	Se leerá byte por byte. No hay [PAG]
0X00.0001	IDATA
0X00.0010	XDATA
0X00.0101	CODE

[DA] y [DB] es la parte Alta y Baja de la dirección de 16Bits con que se accede a memoria. [LN] es la cantidad de Bytes a leer. [PAG] indicaría la página de memoria a referirse el comando.

Retorna: **[s] <Bytes con Datos>**

Escribir Memoria

Escribe en posiciones de memoria de cualquiera de las áreas accesibles. De esta manera, áreas que no son memoria sino periféricos, también puede ser accedida.

Formato: **0x23 [Tip] <[Pag]> [DA] [DB] [LN] <Datos>**

Valor	Significado
0100XXXX	Se leerá por Páginas, indicado por el parámetro opcional [PAG] que se encuentra presente
0000XXXX	Se leerá byte por byte. No hay [PAG]
0X00XXXX	Memoria no Flash
1X00XXXX	Memoria Flash (Sólo XData o Code)
0X00.0001	IDATA
0X00.0010	XDATA
0X00.0101	CODE

⁴ Cabe destacar que la memoria, para posibilitar el direccionamiento de 512KBytes con sólo 16 bit de direcciones, está dividido en 4 paginas de 8Kbytes (cada uno contentivo de dos sectores). Una página se encuentra elegida por defecto, pero si se desea acceder a otra se debe especificar.

[DA] y [DB] es la parte Alta y Baja de la dirección de 16Bits con que se accede a memoria. [LN] es la cantidad de Bytes a escribir. [PAG] indicaría la página a referirse el comando.

Retorna: [s]

Protección de Lectura y escritura de memoria

Tanto la memoria de programa como la de datos pueden ser protegidas contra lectura o escritura. Se debe incluir un código de 8 Bytes por cada una de las dos posibles memorias a proteger.

Formato (Memoria de Código): 0x23 X5 0xFF 0xB0 0x08 <código de 8 bytes>

Formato (memoria de datos): 0x23 X2 0xFF 0xA0 0x08 <código de 8 bytes>

Retorna: 0x1F

Borrar memoria Flash

Este comando es capaz de borrar por completo o parcialmente (por sectores) el contenido de la memoria flash. Solo es posible en memoria no protegida contra escritura.

Formato: 0x26 [Tip] <[Pag]> [DA] [DB] <Código>

El tipo es idéntico a la tabla anterior.

7	6	5	4	3	2	1	0
1	P	0	0	T	T	T	T

Donde P es 1 si se desea acceder por páginas (bloques) y TTTT es 0010 (binario) si se desea acceder a XDATA o 0101 (binario) si se va a borrar memoria de código.

Si <código> es 0x10 se borrará toda la memoria (para ello se debe colocar la dirección en 0xD555), o sólo un sector será borrado si es 0x30 (en tal sentido debería la dirección ser la inicial del sector). DA y DB indican la parte alta y baja de la posición de memoria a acceder, [Pag] indica la página a la cual se refiere, si es distinto del actual. En la tabla 1 (Anexos) se muestra la distribución de memoria del módulo según el esquema de paginación (en caso de memoria de 128K como en el Gemcore500 MS1 en el MS4 son 16 páginas).

Tabla 1 (Anexos) Distribución de memoria según el esquema de paginación				
	Página 0	Página 1	Página 2	Página 3
0x8000	Sector 1	Sector 1	Sector 1	Sector 1
0xBFFF				
0xC000	Sector 2	Sector 2	Sector 2	Sector 2
0xFFFF				

Retorna: [s]

Seleccionar Página de memoria actual

Tal como se especificó en el ítem anterior, para poder acceder la totalidad de los 512Kbytes de memoria con tan solo un bus de 16bits, se ha dividido en páginas de 32Kbytes (cada uno con sectores de 16Kbytes). Se acceden individualmente, y por defecto se encuentra seleccionada la página 0. Sin embargo, en ciertos momento (por ejemplo cuando se llena la pagina 0) se hace necesario empezar a trabajar con la siguiente página, para no necesitar especificarlo, se puede cambiar la selección de la pagina activa con este comando y luego todos los siguientes comandos donde no se especifique la página, asumirán que se refieren a la pagina seleccionada. En memoria de Código, también se puede referirse a las aplicaciones.

Formato: **0x27 [Pag] <[App]>**

En este caso

7	6	5	4	3	2	1	0
0	C	C	C	0	D	D	D

Donde CCC es la página de memoria CODE a seleccionar, en caso contrario, DDD es el número de la página de memoria XRAM a seleccionar. El byte [APP] es opcional e indica el número de la aplicación.

Retorna: **[s]**

Leer Puerto del Microcontrolador.

Lee directamente el estado del puerto del microcontrolador.

Formato: **0x24 [Puerto]**

Donde Puerto 0 es 0x00, Puerto 1 es 0x01, Puerto 2 es 0x02 y Puerto 3 es 0x03

Retorna: **[s] [Valor]**

Escribir en Puerto del Microcontrolador

Escribe directamente un valor en el puerto del microcontrolador.

Formato: **0x25 [Puerto] [Valor]**

Donde Puerto 0 es 0x00, Puerto 1 es 0x01, Puerto 2 es 0x02 y Puerto 3 es 0x03

Retorna: **[s]**

Comandos de la Pantalla de Cristal Líquido

Encender y limpiar LCD

Enciende el LCD con la rutina que indica el fabricante (secuencia de inicialización) si está encendido, lo borra y retorna al punto inicial.

Formato: **0x2A**

Retorna: **[s]**

Mostrar texto en pantalla.

Muestra una cadena de caracteres en el LCD.

Formato: **0x2B <[POS]> [Datos...]**

Donde [POS] es un Byte que indica la posición a la cual se moverá el cursor antes de escribir. La posición cero de la línea 1 es 0x80 (termina en la posición 0x8F), y la posición cero de la línea 2 es 0xC0 (termina en la posición 0xCF).

Los [datos] son los códigos ASCII de cada una de las letras contentivas del texto a mostrar.

Retorna: [s]

Mostrar un solo carácter

Muestra un solo carácter a partir de la posición actual del cursor en la pantalla.

Formato: 0x2C [Car]

Donde [Car] es el código ASCII del carácter a mostrar

Retorna: [s]

Enviar Comando al LCD

Envía uno de los comandos prediseñados por el fabricante del LCD.

Formato: 0x2D [Cod]

Donde Código es uno de los comandos mostrados en la tabla.

Código	Acción
0x01	Limpia la pantalla (borra todo)
0x02	Cursor a la posición letra 1, línea 1
0x04	Mueve el cursor a la izquierda luego de mostrar un carácter
0x05	Mueve el texto completo a la derecha luego de mostrar un carácter
0x06	Mueve el cursor a la derecha luego de mostrar un carácter
0x08	Mueve el texto completo a la izquierda luego de mostrar un carácter
0x09	Apaga el LCD
0x0C	LCD encendido pero sin cursor
0x0D	LCD encendido con carácter titilando en la posición del cursor
0x0E	LCD encendido con cursor fijo
0x0F	LCD encendido con cursor titilante
0x10	Mueve el cursor a la izquierda
0x14	Mueve el cursor a la derecha
0x18	Mueve el texto a la izquierda
0x1C	Mueve el texto a la derecha

Retorna: [s]

Comandos de Teclado y Buzzer

Fijar tiempo de espera por tecla pulsada.

Este comando fija el número de segundos durante los cuales el lector escrutará el teclado para ver si se ha pulsado una tecla. Es la única manera de leerlo, pues el módulo no trabaja por interrupción.

Formato: 0x32 [Tiem] [beep]

Donde [Tiem] es el número de segundos en unidades de 100ms que el lector explora el teclado en espera de una tecla pulsada. Si [Beep]=0x00, al detectar tecla pulsada no emite sonido, en caso contrario [Beep]=0x01, la detección de tecla produce un sonido en el buzzer

Retorna: [s] [Tecl]

Donde [Tecl] es el código devuelto según cada caso de la tabla.

Tecla	Código	Tecla	Código	Tecla	Código	Tecla	Código
'1'	0x11	'2'	0x21	'3'	0x31	F1	0x41
'4'	0x12	'5'	0x22	'6'	0x32	F2	0x42
'7'	0x13	'8'	0x23	'9'	0x33	F3	0x43
'<'	0x14	'0'	0x24	'>'	0x34	F4	0x44

En caso de no haberse pulsado alguna tecla se retorna 0xCF

Sonar buzzer

Realiza sonidos en el buffer con la frecuencia y duración especificada.

Formato: 0x33 [Tiem] <[Frec]>

[Tiem] es la duración del sonido en el buzzer La unidad de tiempo está en función de la frecuencia. [FREC] es un parámetro opcional y especifica la frecuencia del sonido (entre 1.183 Hz y 68.287 Hz), por defecto es 3.600 Hz. Ambos parámetros están relacionados por:

Tiempo= T (ms) * N (Hz) / 36000

Frecuencia = 307200 / N (Hz) – 4

Retorna: [s]

Comandos del Reloj de tiempo Real

Leer fecha y hora

Obtiene la hora y fecha del Reloj de Tiempo real (RTC)

Formato: 0x3A

Retorna: [s] AÑO MES DIA HORA MINUTO SEGUNDO

- AÑO en BCD (ejemplo 0x08 = 2008)
- MES en BCD (Ejemplo MAYO= 0x05)
- DIA día del mes en BCD (EJEMPLO 0x20 es el 20)
- HORA la hora en formato militar y en BCD (Ejemplo 0x14 = 2 de la tarde)
- MINUTO en BCD (Ejemplo 0x31, es decir 31 minutos pasada la hora)
- SEGUNDO en BCD (Ejemplo 0x45 es 45 segundos)

Actualizar hora y fecha

Guarda un nuevo valor de arranque con el cual el reloj seguirá sincronizado.

Formato: **0x3A AÑO MES DIA HORA MINUTO SEGUNDO**
Retorna: **[s]**

ANEXO 2

DIAGRAMAS DE FLUJO DE LOS PROGRAMAS DESARROLLADOS

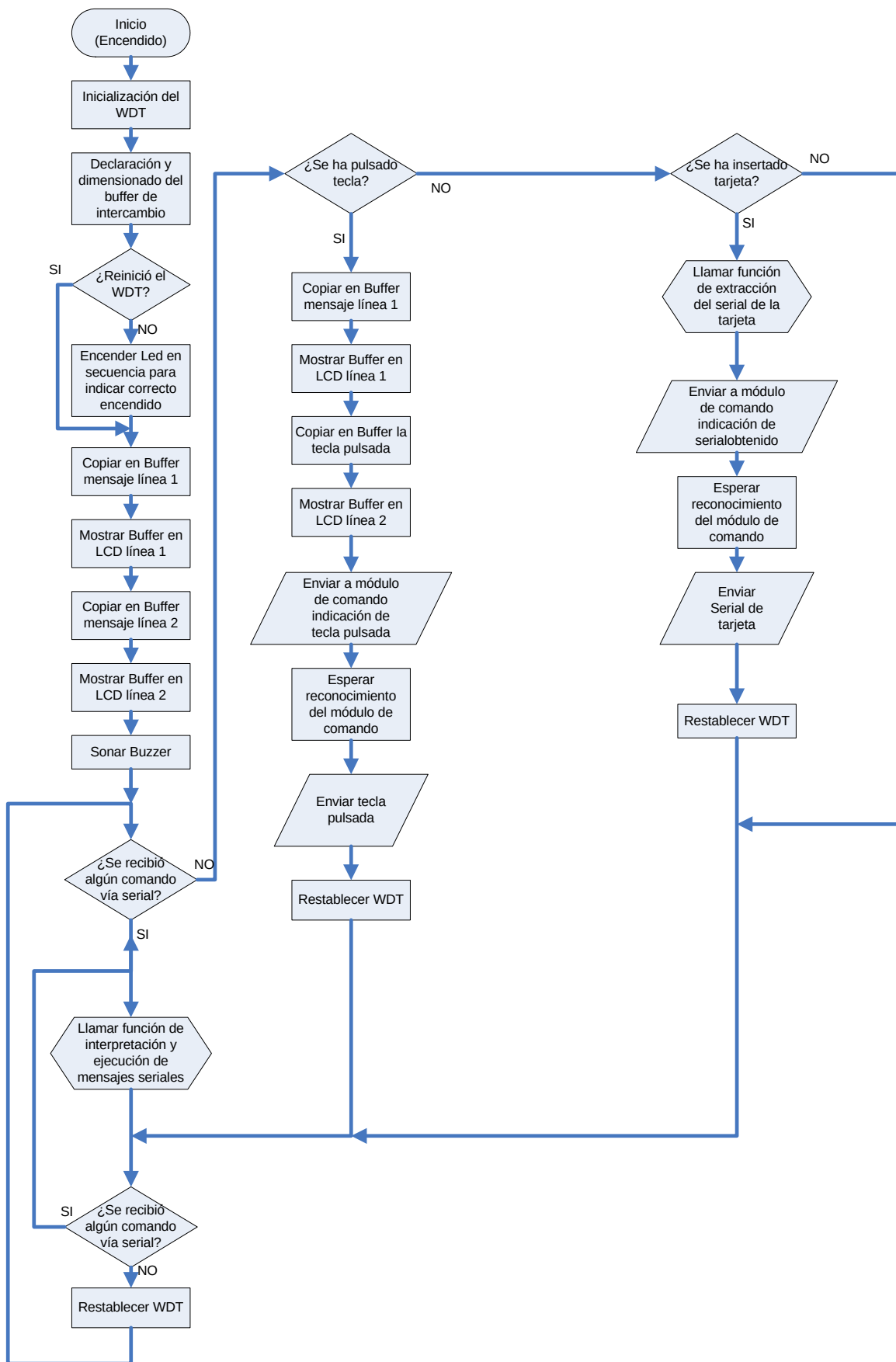


Ilustración 1 (Anexos) Diagrama de flujo del procedimiento principal de espera del programa residente en el módulo de control de interfaz. Esta rutina se ejecuta continuamente mientras no ocurra evento alguno.

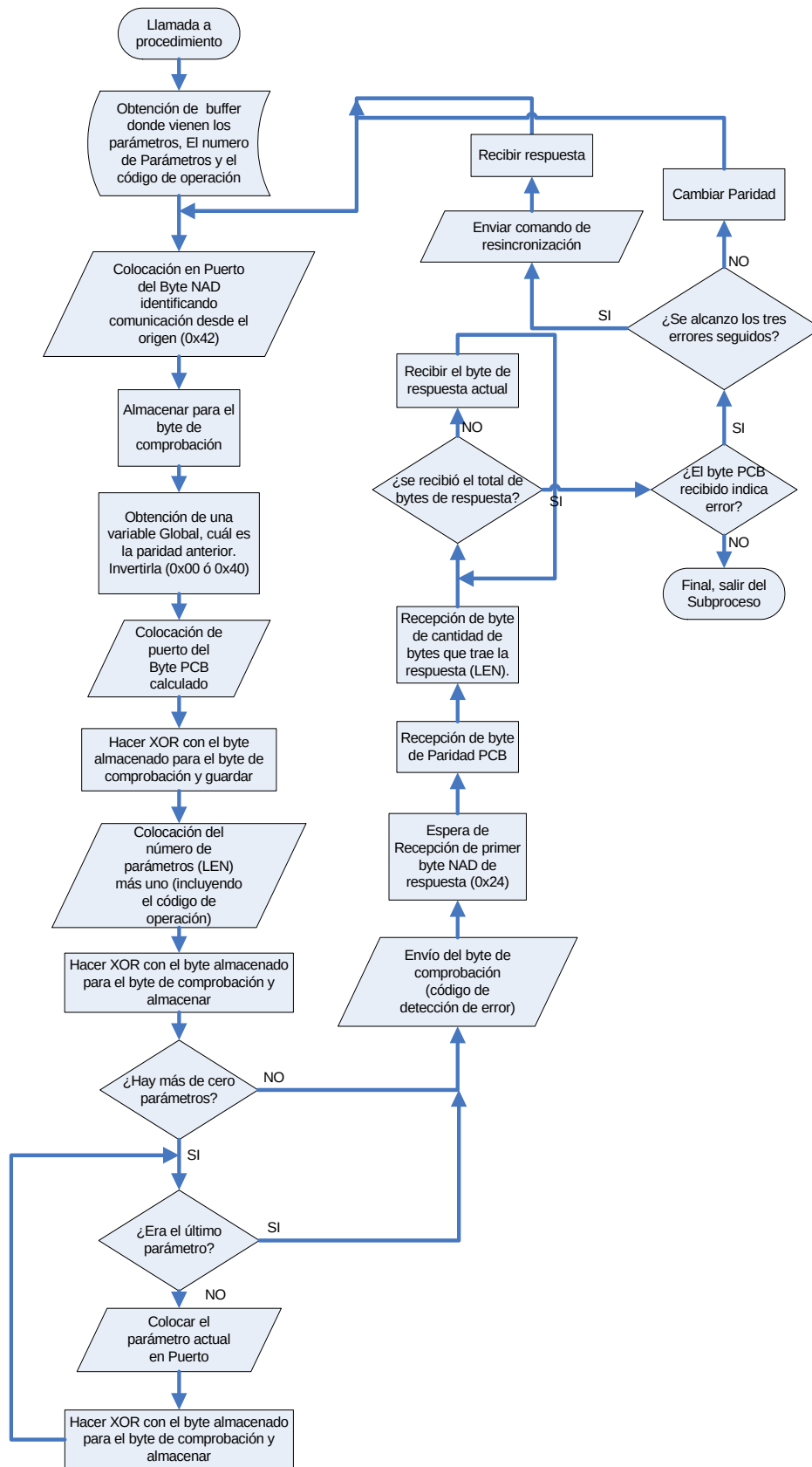
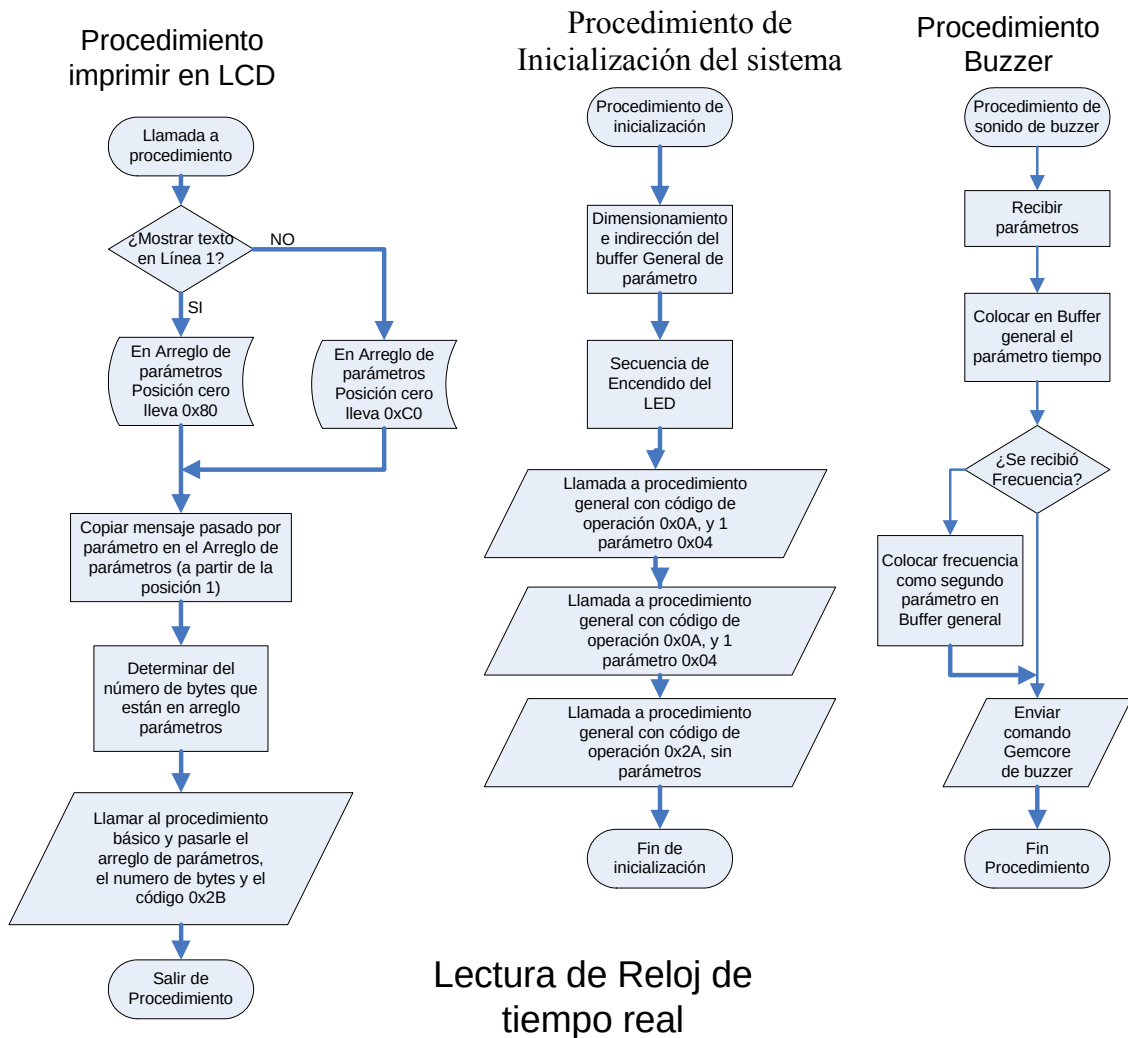


Ilustración 2 (Anexos) Diagrama de Flujo de la rutina básica que implementa el Protocolo de Bloque de Gemplus GBP, es decir, realiza la interfaz entre el resto del sistema y el protocolo que comprende el Módulo GemCore500. Sobre esta rutina, se basa todo los demás procedimientos de la programación del Módulo Principal de Interfaz, en lo que a comunicación con el GemCore500 se refiere.



Lectura de Reloj de tiempo real

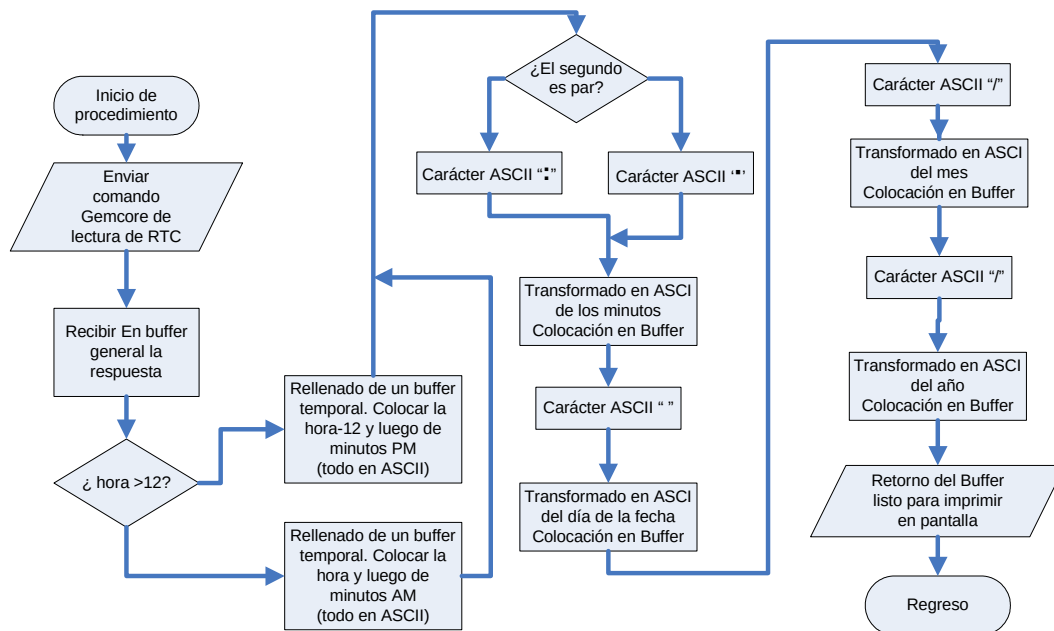


Ilustración 3 (Anexos). Diagramas de flujo (varios) de los respectivos procedimientos de "herramientas" que basados en el procedimiento básico de la Ilustración 2, proveen el manejo de interfaz entre el protocolo de comunicaciones ya implementado y el control de los posibles periféricos presentes

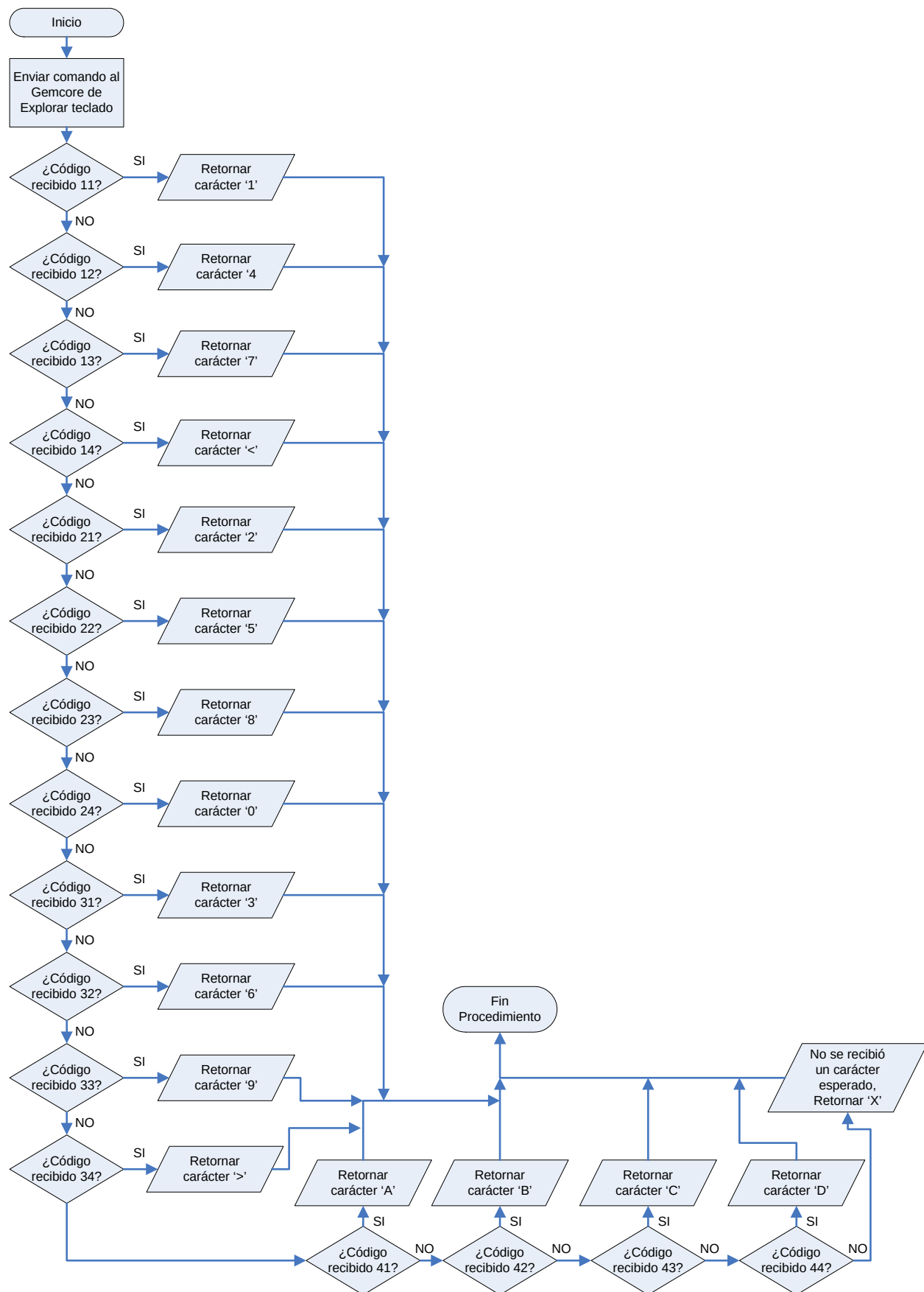


Ilustración 4 (Anexos). Diagrama de flujo de la rutina de "Herramientas", que sobre la base de la rutina "básica" de la ilustración 2, realiza la exploración del teclado por solicitud

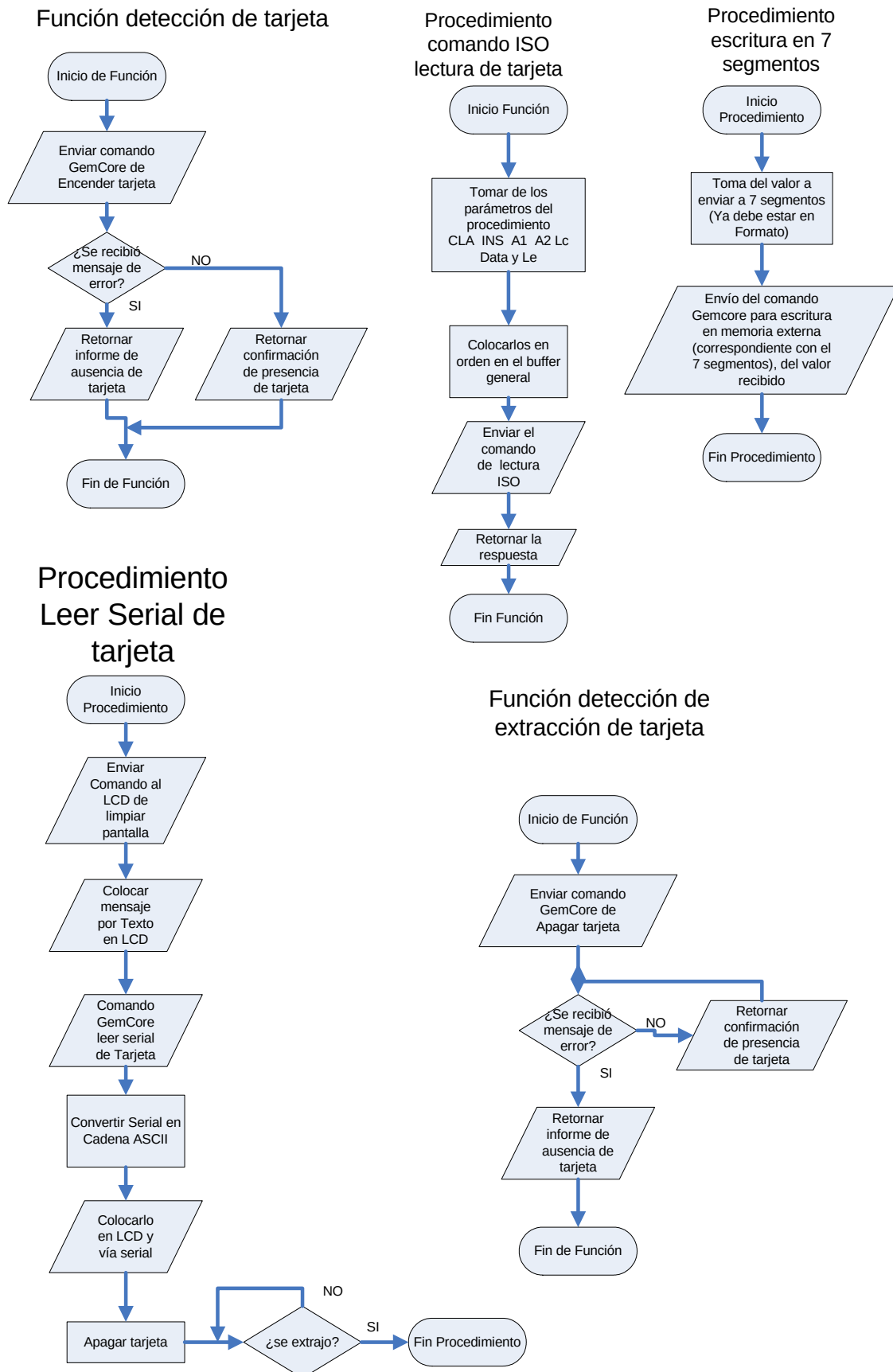


Ilustración 5 (Anexos). Rutinas y procedimientos, que formando parte de "Herramientas" y usando el procedimiento básico de la Ilustración 2, accionan tanto al presentador 7 segmentos como a la rutina de lectura del Serial de la tarjeta inteligente y detectar su inserción o extracción

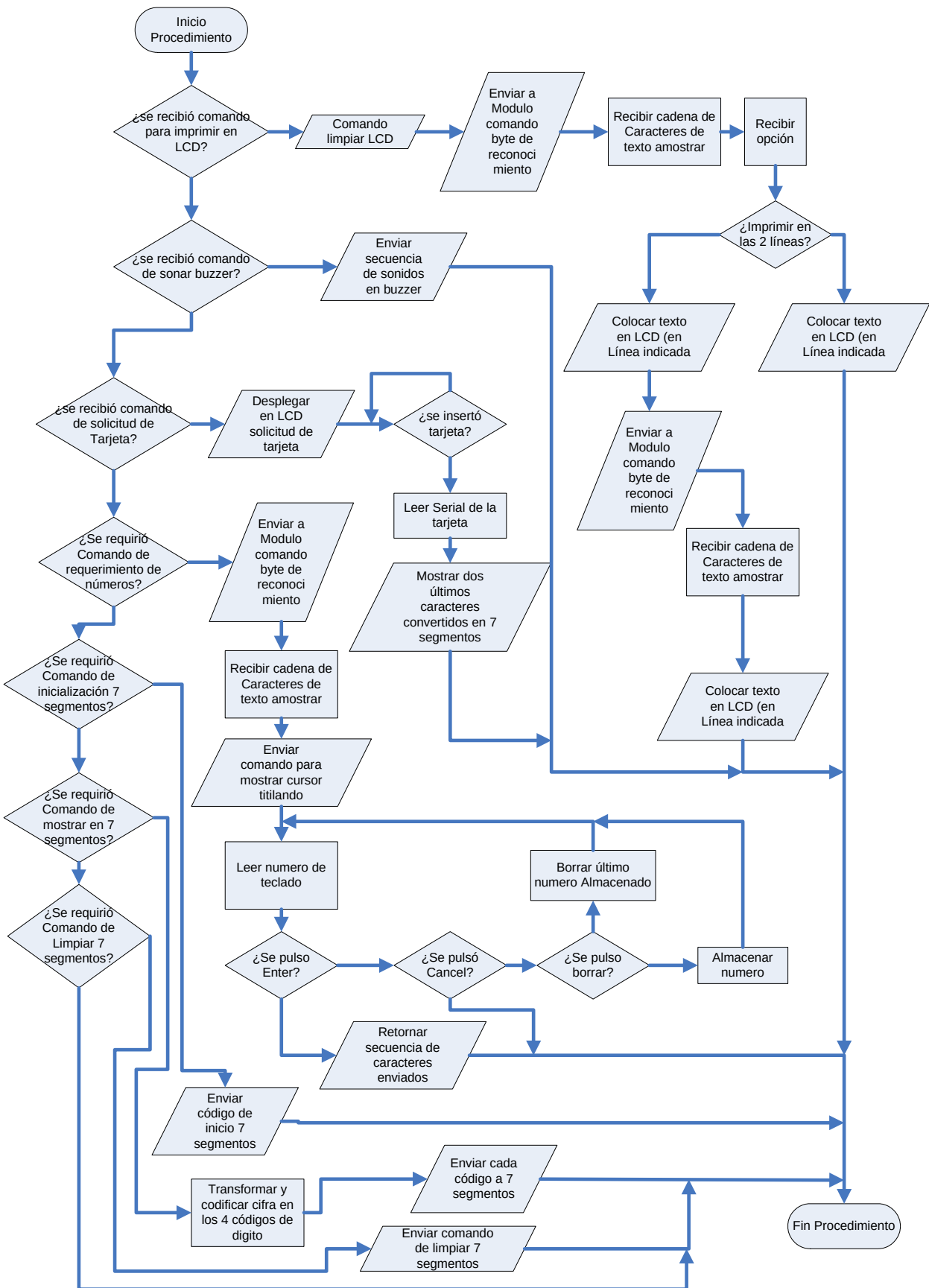


Ilustración 6 (Anexos) Procedimiento que interpreta los comandos seriales, referenciado en la ilustración 1 de los presentes anexos, es llamada al recibir un comando externo del módulo general de comando

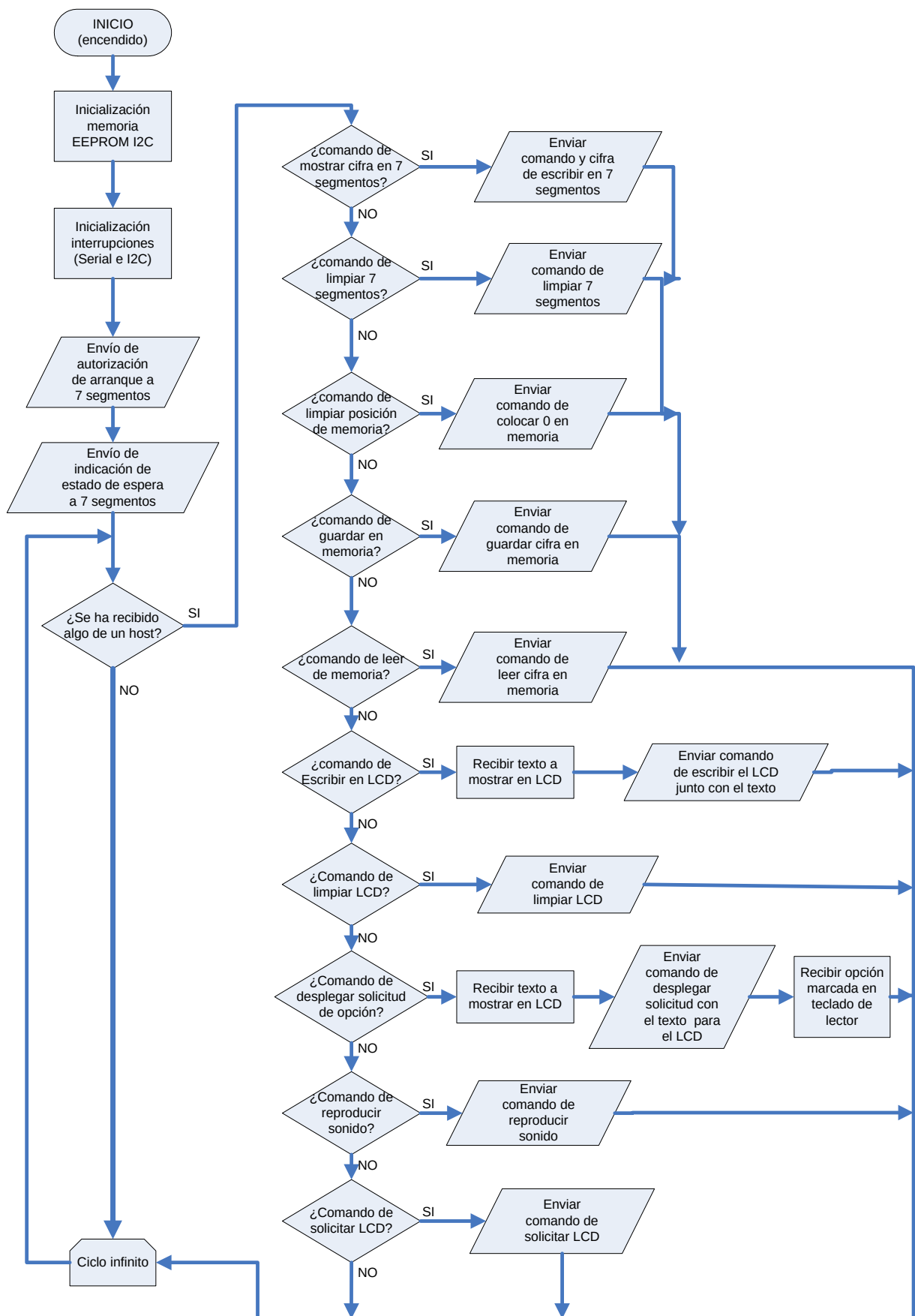


Ilustración 7 (Anexos) Ciclo Principal del programa residente en el dispositivo denominado Módulo de Comando General. Este programa, comprende la aplicación demostrativa desarrollada para demostrar la correcta funcionalidad del prototipo

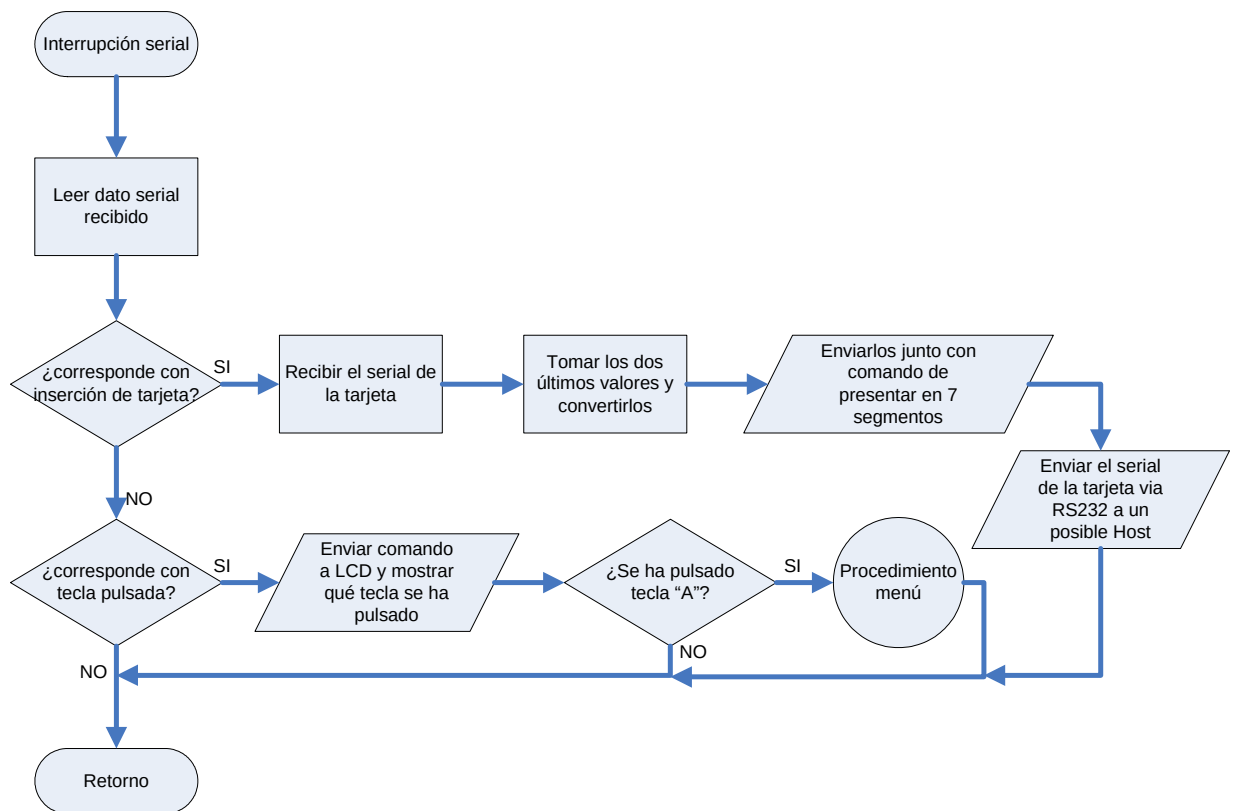


Ilustración 8 (Anexos) Diagrama de flujo de la respuesta a la interrupción generada por el módulo Principal de interfaz, en caso de producirse un evento informarlo al presente módulo (módulo de comando)

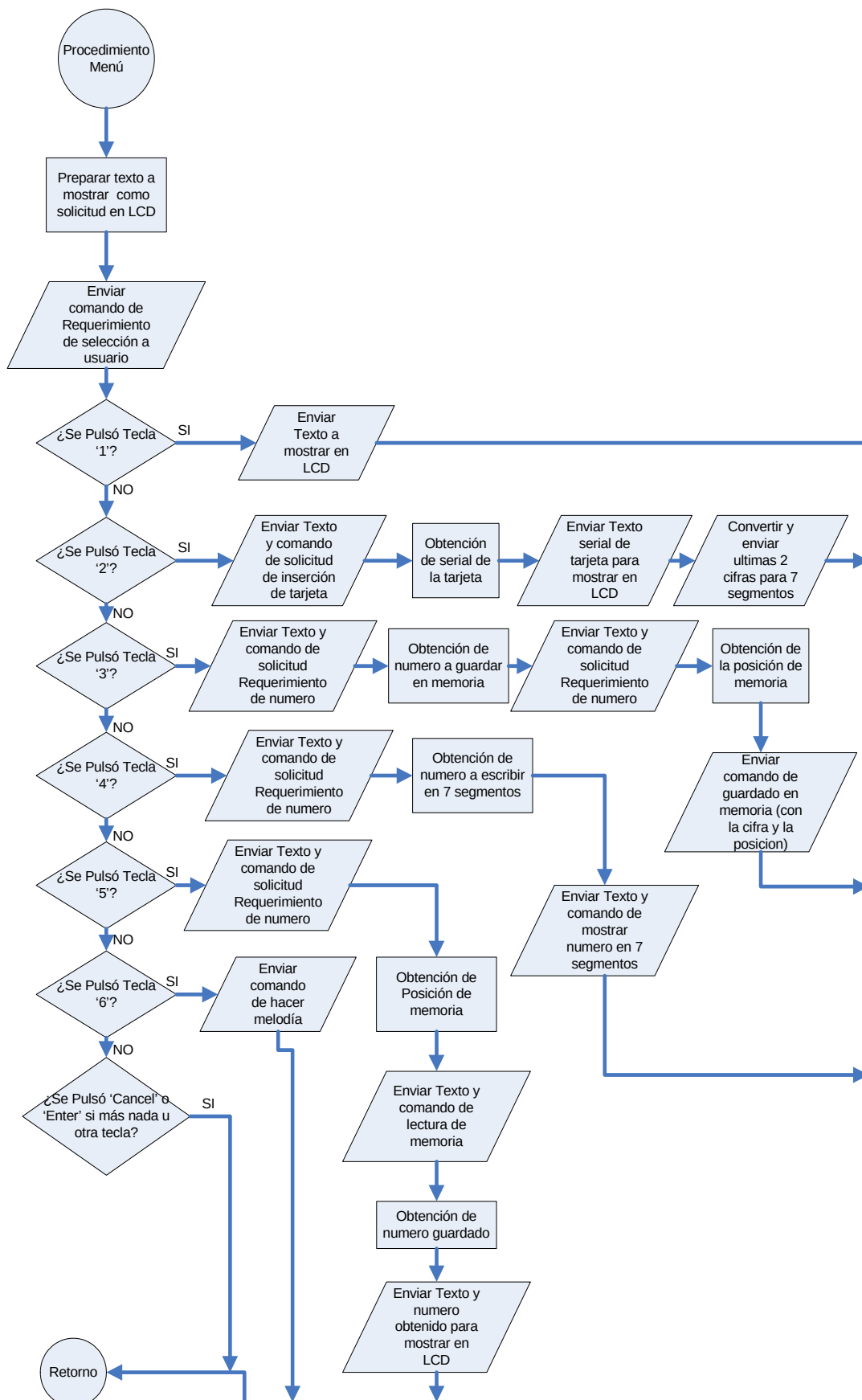


Ilustración 9 (Anexos). Diagrama de flujo del procedimiento de menú, que genera el requerimiento al usuario de una opción y gestiona la orden de ejecución al módulo principal de Interfaz

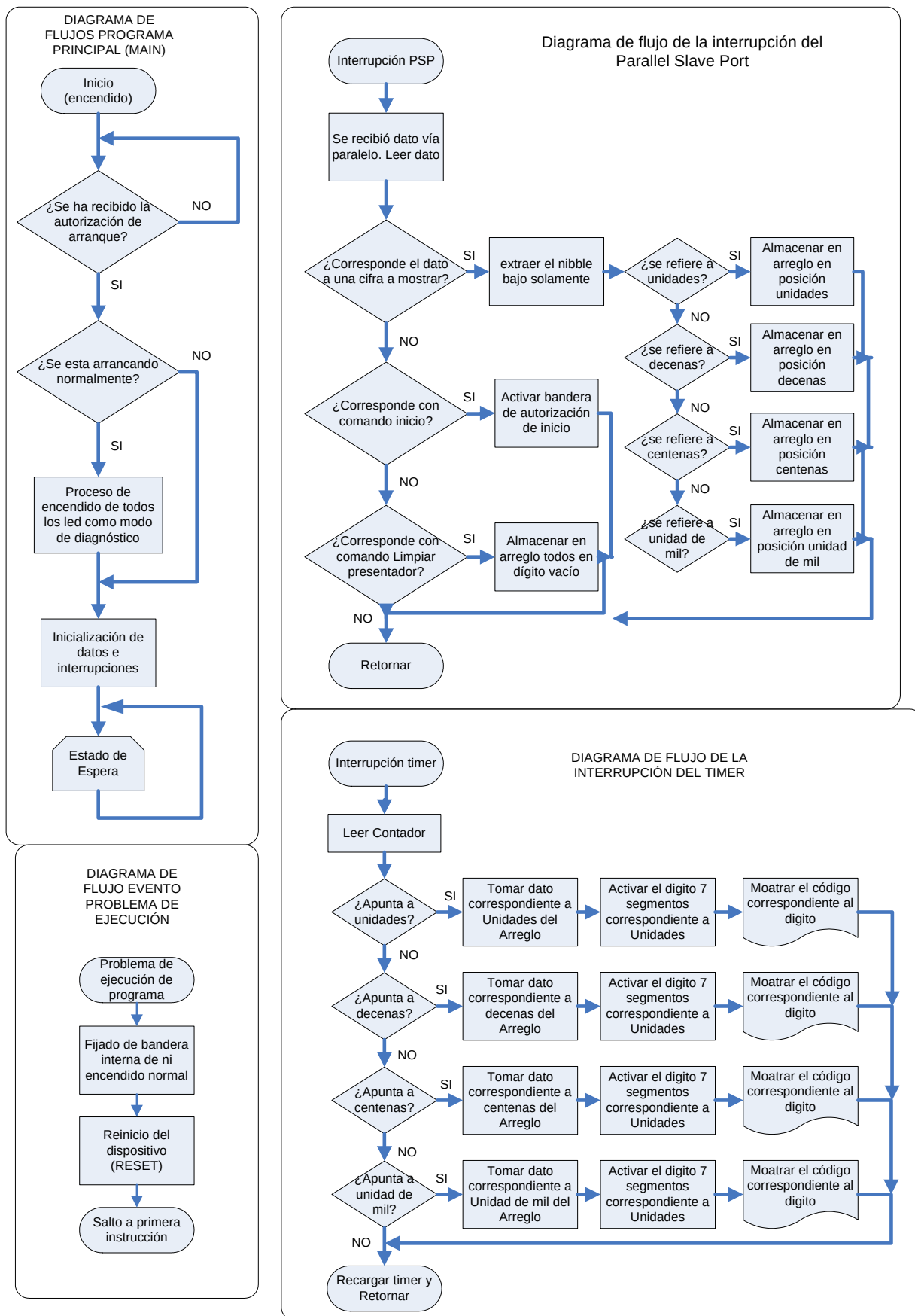


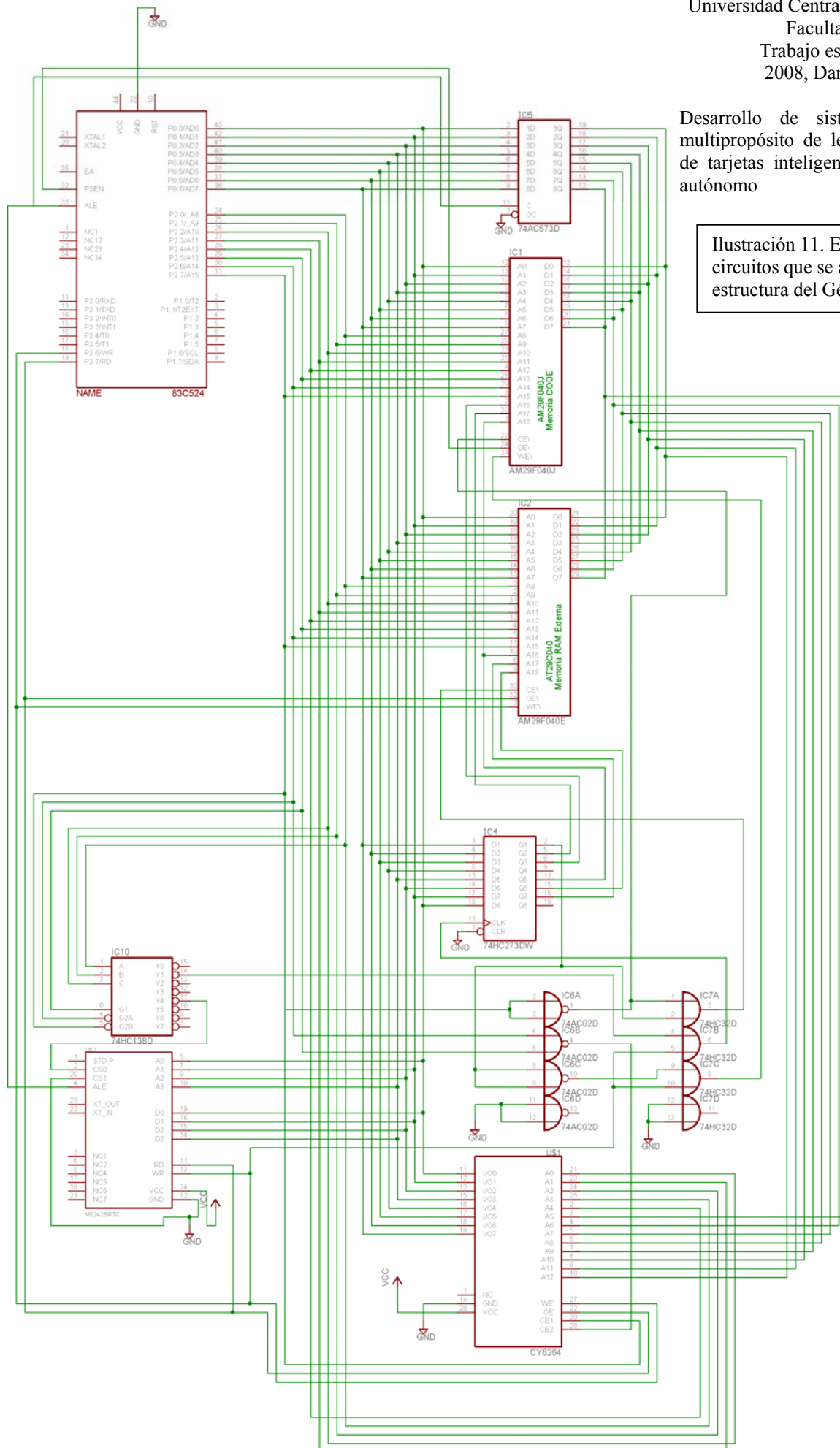
Ilustración 10 (Anexos). Diagrama de Flujo del programa que reside en el control independiente que gobierna al visualizador 7 segmentos, conformado por una rutina central y dos procedimientos activados por interrupción.

ANEXO 3

ESQUEMATICO DE LOS CIRCUITOS DEL PROYECTO

Desarrollo de sistema Modular
multipropósito de lectura escritura
de tarjetas inteligentes en modulo
autónomo

Ilustración 11. Esbozo de los
circuitos que se aprecian en la
estructura del Gémcore 500



Desarrollo de sistema Modular
multipropósito de lectura escritura
de tarjetas inteligentes en modulo
autónomo

Ilustración 12.
Plano de los
circuitos de la
tarjeta de
conexión de
interfaces

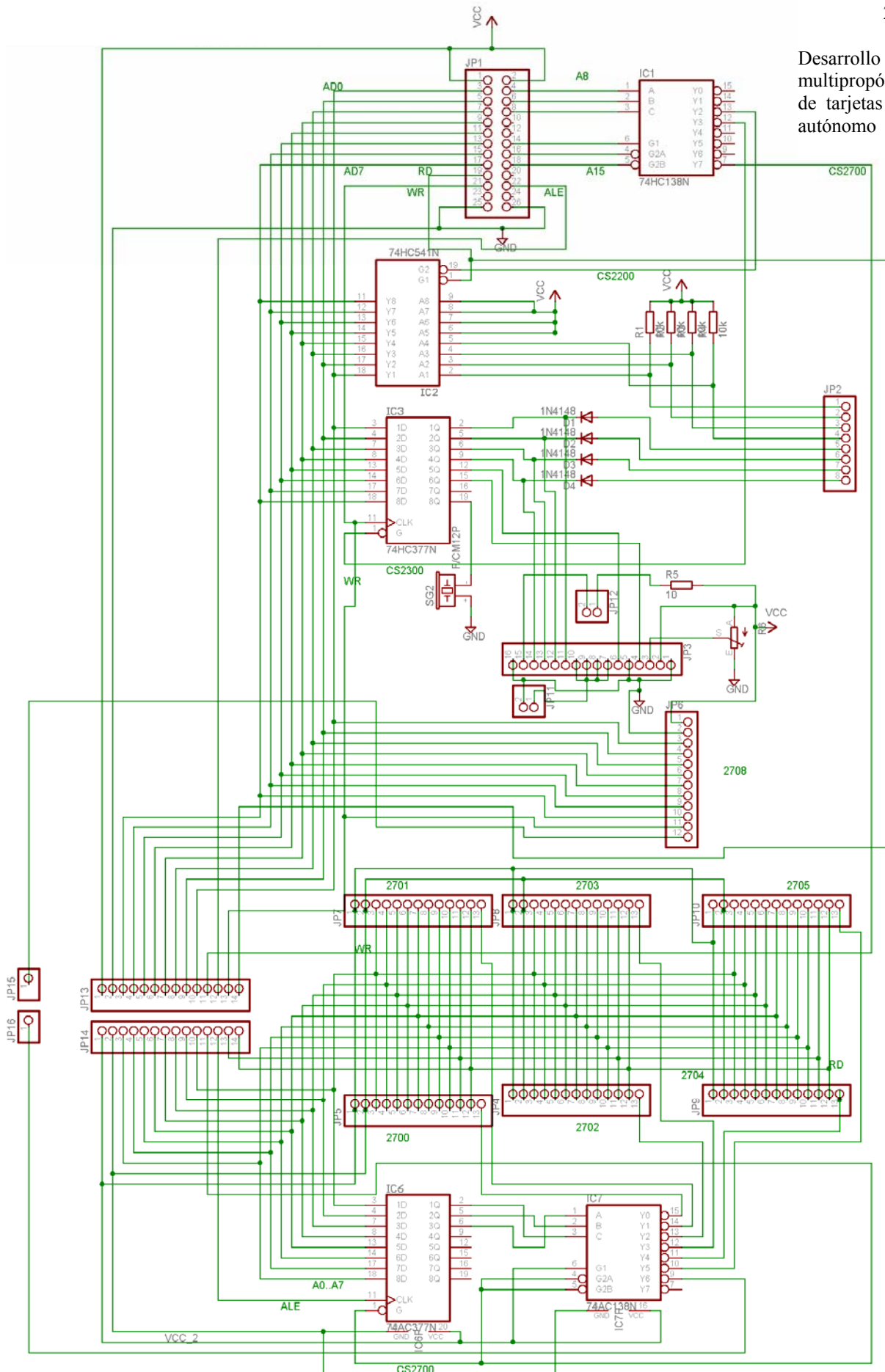
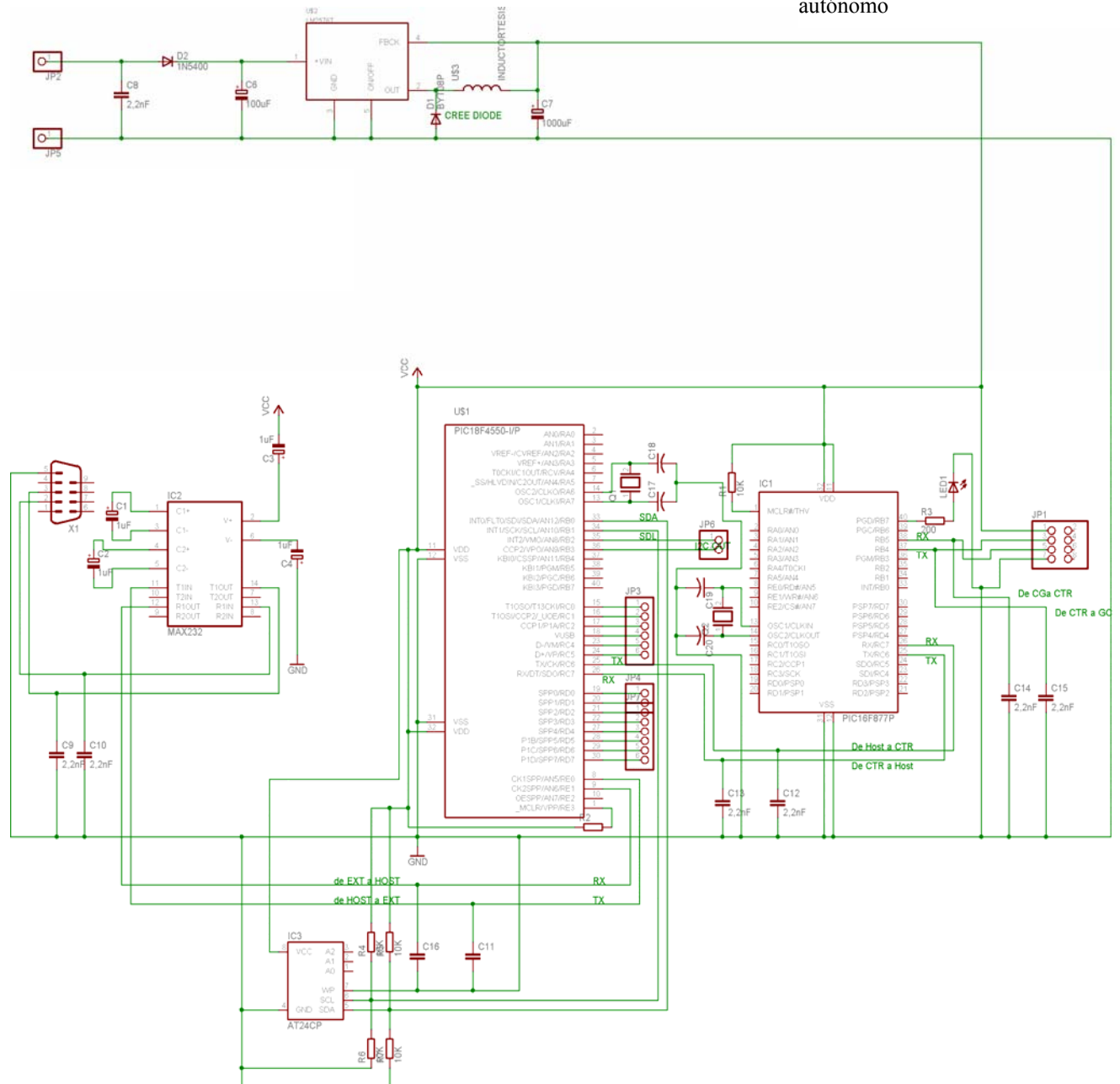


Ilustración 13. Plano de la Tarjeta de Comando General (Donde reside la Inteligencia del sistema)

Desarrollo de sistema Modular
multipropósito de lectura escritura de
tarjetas inteligentes en modulo
autónomo



Desarrollo de sistema Modular multipropósito de lectura escritura de etiquetas inteligentes en modulo autónomo

Ilustración 5. Plano de la tarjeta del visualizador 7 segmentos (Arriba) y del adaptador de dígitos (abajo)

Desarrollo de sistema Modular
 multipropósito de lectura escritura de
 tarjetas inteligentes en modulo autónomo

