UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

SISTEMA DE BLOQUEO DE ENCENDIDO PARA VEHÍCULOS MEDIANTE LECTOR BIOMÉTRICO Y AVISO MEDIANTE SMS

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRÓNICO

AUTORES:

Paúl Fernando San Martín Ledesma

Christian Esteban Serrano Cevallos

DIRECTOR:

Omar Santiago Alvarado Cando

CUENCA - ECUADOR

2014

DEDICATORIA

A mi Madre por haber sido el apoyo no solo durante mi carrera si no en toda mi vida, por toda su paciencia y dedicación durante todo este trayecto en el que ha hecho de mi una persona de bien. A toda mi familia que es un pilar fundamental en mi vida sobre todo a mi hermano con el cual hemos luchado siempre para conseguir nuestros objetivos. A todas las personas que siempre han estado apoyándome y han confiado en mí, en especial a Belén que ha sido un gran apoyo en este y muchos proyectos.

A Dios que es por quien todos los proyectos se lleva a cabo y quien guía mi vida.

Paúl

DEDICATORIA

Este trabajo quiero dedicar a mi familia que incondicionalmente me han estado apoyando día a día para alcanzar mis metas propuestas y de manera muy especial a mi madre que ha sido el pilar más importante para la culminación de esta importante etapa y a mi hija Ana Cristina quien representa lo más importante que existe en mi vida y para la cual me debo como ser humano y profesional.

Esteban

AGRADECIMIENTO

Agradezco de manera especial a Dios por todas las bendiciones recibidas en mi vida y permitirme culminar mi carrera. Mil gracias a todos los familiares, amigos y a todas las personas que durante este tiempo han sido mi apoyo para lograr mis objetivos. Al Lcdo. Wilson Chuquin que ha sido un amigo que ha estado motivándonos y pendiente de que cumplamos con el proyecto de Tesis. A todos los profesores que siempre han estado ahí para todas las inquietudes que se han generando a lo largo de este caminar ya que sin ellos no hubiera sido posible alcanzar esta meta.

Paúl

AGRADECIMIENTO

Mis más sinceros agradecimientos para la Universidad del Azuay con todo su personal docente y administrativo quienes me han brindado su ayuda y apoyo en cada momento a lo largo del buen camino de la enseñanza, de manera especial a nuestros directores de tesis que nos ayudaron a que este proyecto se lo realice de la mejor manera Ing. Leonel Pérez que por asares de la vida tuvo que retirarse de su función de profesor y nos dejo en manos de otra excelente persona como es el Ing. Omar Alvarado.

A mi familia y amigos que supieron entenderme y darme esos ánimos para no desfallecer en los momentos difíciles y no permitirme abandonar el sueño de alcanzar esta meta.

Esteban

INDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

SISTEMA DE BLOQUEO DE ENCENDIDO PARA VEHÍCULOS MEDIANTE LECTOR BIOMÉTRICO Y AVISO MEDIANTE SMS

Resumen

Debido al incremento de la delincuencia y el robo de autos en el Ecuador se busca la manera de incrementar la seguridad existente mediante un sistema de seguridad biométrico. Desarrollar el marco teórico, diseñar, construir y elaborar software y hardware de un sistema de seguridad con huella digital eficiente. Para el proyecto se empleó el método analítico que determina ventajas de seguridad del sistema comparado con sistemas de seguridad existentes y el método deductivo para analizar beneficios de usuarios con el aumento de la seguridad en el vehículo. Como resultado se obtuvo un dispositivo biométrico de seguridad que limita el uso del vehículo a personas autorizadas únicamente. En conclusión, el dispositivo puede ser muy útil como un sistema de seguridad de vehículos frente al nivel de robo en la actualidad.

Palabras Clave: Biométrico, Huella Digital, Lector, Alarma, Seguridad.

Ing. Omar Santiago Alvarado Cando — DIRECTOR TRABAJO DE GRADO

Ing. Francisco Eugenio Vásquez Calero — DIRECTOR DE ESCUELA

Paúl Fernando San Martín Ledesma — AUTOR

Cristian Esteban Serrano Cevallos — AUTOR

ABSTRACT

VEHICLES FIRE SUPPRESSION SYSTEM BY BIOMETRIC READER AND SMS ALERT SYSTEM

Due to crime increase and car theft in Ecuador, our objective was to find ways to increase the existing security through a biometric security system. The project aims to develop the theoretical framework, as well as to design, build and develop a software and hardware of a security system with efficient fingerprint. We used the analytical method to determine the system's security advantages compared to existing security systems; and the deductive method for analyzing users' benefits when the security in the vehicle was increased. As a result, a biometric security device that limits the use of the vehicle only to authorized persons was obtained. In conclusion, the device can be very useful as a vehicle security system against today's theft level.

Ing. Omar Santiago Alvarado Cando
THESIS DIRECTOR

Ing. Francisco Eugenio Vásquez Calero
SCHOOL DIRECTOR

Paul Fernando San Martín Ledesma
AUTHOR

Cristian Esteban Serrano Cevallos
AUTHOR

UNIVERSIDAD DEL AZUAY
DPTO. IDIOMAS

Translated by,
Lic. Lourdes Crespo

Paul Fernando San Martín Ledesma

Christian Esteban Serrano Cevallos

Trabajo de Graduación

Ing. Omar Alvarado

Enero 2014

SISTEMA DE BLOQUEO DE ENCENDIDO PARA VEHÍCULOS MEDIANTE LECTOR BIOMÉTRICO Y AVISO MEDIANTE SMS

Introducción

Debido a la delincuencia que existe en el Ecuador cada vez más se busca la manera de proteger los bienes personales, mejorando las seguridades existentes o aumentándolas con nuevos dispositivos y muchos de ellos tecnológicamente más avanzados, y de la misma manera los delincuentes van a la par con nuevas técnicas de robo, tomando como ejemplo los vehículos y sus alcances en seguridades partiendo desde la chapa simple que se acciona con una llave, los delincuentes han encontrado la manera de violentar esta seguridad realizando puentes entre las conexiones que llegan al encendido.

Las alarmas son otras seguridades que se han implementado, existen muchas variedades, a los controles de las primeras alarmas que salieron al mercado se las programaba mediante la variación de un potenciómetro hasta compatibilizar con la frecuencia del receptor instalado en el vehículo.

Estas han ido mejorando su tecnología cada vez mas encontrando alarmas que a mas de trabajar con una frecuencia determinada posee un código encriptación para validación, adicionalmente se encuentran alarmas que utilizan estos sistemas y que adicionalmente permiten ingresar una clave mediante teclados.

Es por esto que se ve la necesidad de seguir mejorando tecnológicamente estas seguridades y se ha presentado como alternativa para este proyecto el diseño de un sistema de seguridad biométrico mediante huella digital que no pretende reemplazar a los sistemas existentes y comúnmente utilizados, sino que brinda una alternativa adicional de seguridad mejorando las que los vehículos ya poseen.

Se ha optado por el sistema biométrico mediante la lectura de huella digital debido a la alta seguridad que ofrece ya que una huella dactilar es única para cada persona y casi imposible de copiar, a demás son sistemas ya probados y que los usuarios ya se encuentran familiarizados con su uso.

CAPÍTULO I

ANÁLISIS E INVESTIGACIÓN DEL MARCO TEÓRICO QUE SUSTENTA EL PROYECTO

1.1. Introducción.

En este primer capítulo se tratará sobre algunos aspectos generales y básicos que son necesarios conocerlos para tener un mejor entendimiento del proyecto, como son la problemática de la inseguridad en nuestro país, conocimientos de los sistemas biométricos, su seguridad y especificaciones, características de los microcontroladores y los métodos más usados para realizar el bloqueo de un vehículo.

1.2. Problemática.

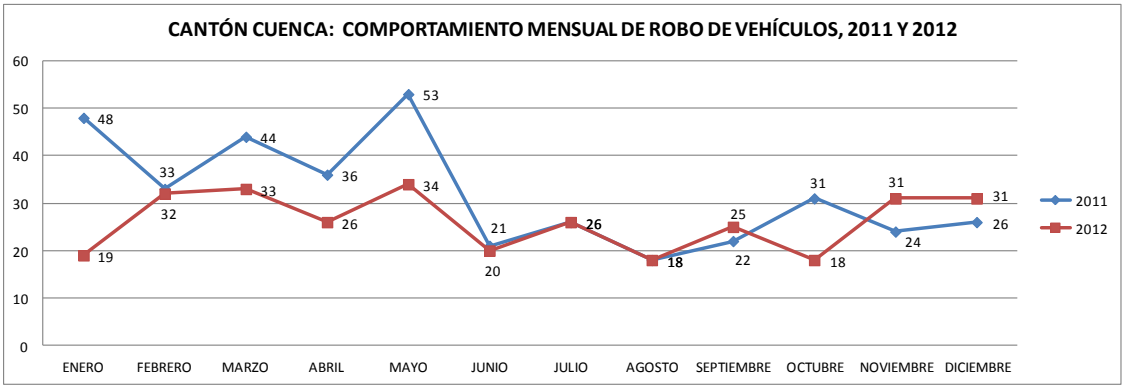
Cuenca se ha convertido en un importante centro de comercialización y circulación de vehículos robados o con registros adulterados; analizando las estadísticas que maneja el Consejo de Seguridad Ciudadana de Cuenca en la tabla 1.1, se puede dar cuenta de ello.

La delincuencia es un problema social muy complejo que afecta a todo nivel económico y probablemente en mayor medida a la gente de estrato económico más bajo que es un grupo social indefensos y es más complicado que puedan acceder a medios de seguridad o vigilancia de sus posesiones.

Es debido a esta necesidad de tener mayores seguridades para proteger nuestros bienes que se ha decidido proponer un sistema que proporcione una alta seguridad como se puede dar mediante un sistema biométrico y con un costo para el usuario no muy elevado.

Tabla 1.1. Cuadro estadístico del robo de vehículos en el cantón Cuenca, de los años 2011 y 2012.

Fuente: CONSEJO DE SEGURIDAD CIUDADANA, *Boletín estadístico del Consejo de Seguridad Ciudadana de Cuenca*/ Publicación 2013



	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
2011	48	33	44	36	53	21	26	18	22	31	24	26
2012	19	32	33	26	34	20	26	18	25	18	31	31

1.3. Biometría

El termino Biometría proviene de las palabras bios (vida) y metría (medida), por lo tanto se refiere a que todo equipo biométrico se basa en una tecnología de seguridad que mide e identifica alguna característica física morfológica única e intransferible que nos diferencia de otras personas, como por ejemplo, la forma de la cara, la geometría de partes de nuestro cuerpo como las manos, nuestros ojos y tal vez la más conocida, la huella digital, las características de seguridad de cada una de ellas se puede analizar en la Tabla 1.2.

Tabla 1.2. Comparación de sistemas biométricos. **Fuente:** BIOMETRIA BASICA, *Manual de Aplicación de Tecnologías Biométricas*, Estados Unidos 2008

Tabla Comparativa de las Tecnologías Biométricas					
Tipo	**Universalidad**	**Precisión**	**Facilidad de Uso**	**Aceptación de los Usuarios**	**Estabilidad a Largo Plazo**
Analisis de Firma Dinámica	Bajo	Bajo	Alto	Muy Alto	Medio
Imagén Facial	Bajo	Bajo	Medio	Medio	Medio
Huella Digital	Alto	Alto	Alto	Alto	Alto
Geometría de la Mano	Medio	Medio	Alto	Alto	Medio
Reconocimiento del Iris	Alto	Muy Alto	Alto	Medio	Alto
Teclado	Bajo	Bajo	Alto	Desconocido	Desconocido
Huella de la Palma	Medio/Alto	Medio/Alto	Alto	Desconocido	Desconocido
Escaneo de Retina	Muy Alto	Alto	Bajo	Bajo	Alto
Contacto de Piel	Alto	Desconocido	Desconocido	Desconocido	Desconocido
Verificación de Voz	Bajo	Bajo	Alto	Alto	Medio
Biométrica Vascular	Medio/Alto	Medio	Medio/Alto	Alto	Alto
ADN	Alto	Alto	Bajo	Alto	Muy Alto
Forma del Oído	Alto	Desconocido	Medio	Desconocido	Desconocido
Forma de Caminar	Medio	Desconocido	Alto	Desconocido	Bajo

Para este proyecto se ha analizado las ventajas y desventajas de cada uno de los sistemas existentes y se ha determinado como mejor opción la utilización de la huella digital debido a su alta seguridad, a la familiaridad que existe de la gente con

el uso de estos dispositivos, para esto se realizó el estudio de la información existente en la tabla 1.3 o ingresando en el enlace Tabla 1.3.

La identificación por medio de huellas digitales constituye una de las formas más representativas de la utilización de la biometría. Una huella digital está formada por una serie de surcos, las terminaciones o bifurcaciones de los mismos son llamados puntos de minucia, cada uno de estos puntos tiene una característica y una posición única, que puede ser medida. Comparando esta distribución es posible obtener la identidad de una persona que intenta acceder a un sistema en general, como se ilustra en la Figura 1.1, en donde se detalla estos puntos para la digitalización de la huella digital.

Figura 1.1. Puntos de minucia de huellas digitales.

Fuente: http://www.bixit.mx/2011/identificacion-biometrica

Comparación del algoritmo, Escenario, y las pruebas de funcionamiento, Parte I

Tipo	Medidas	Robusteces	Limitaciones	Tamaño de la	Aplicaciones
Analisis de Firma Dinámica	Como los usuarios firman su nombre	- Prácticamente no hay problemas de derechos de privacidad - Los usuarios pueden cambiar de firmas - Resistente a los impostores - Aprovecha los procesos existentes - Percibido como no invasiva - Aceptación alta del usuario, ya que es similar a la firma del	- Tamaño de la Firma es limitada - Los usuarios están acostumbrados a firmar - Tiene aplicaciones limitadas - Firmas cambian con el tiempo - Precisión baja estar en el mismo tipo de ambiente	500 – 1000 bytes	- Muy adecuado para aplicaciones en las que las firmas son aceptadas
Imagén Facial	Características y patrones Faciales	- Puede capturar imágenes desde la distancia - Hardware asequible - Percibido como menos intrusivo que otras tecnologías - Precisión Moderada - Puede aprovechar las bases de datos existentes, incluyendo	- Fácilmente eludido por encubrimiento y cosméticos - No se puede distinguir entre gemelos idénticos - Nicho de mercado para la autenticación de red - El problema PIEC degrada el rendimiento PIEC es la sensibilidad a las variaciones de ángulo	84 bytes – 3.5K	- Para algunos sistemas en aplicación de pasaporte y visado - En algunos sistemas de control de acceso
Huella Digital	Patrones de Huellas Digitales	- Sistema único incluso entre gemelos - Alto precisión - Tecnología básica bien desarrollada y probada - La tecnología es relativamente barata - Puede ser desarrollada en una variedad de entornos - Estable durante toda la vida (sujeto a las advertencias en la columna de las limitaciones) - Emplea dispositivos ergonómicos y fáciles de usar	- Tiene el estigma "delincuente común" - La impresión de las huellas digitales a menudo se dejan en el sensor - Sequedad de la piel, la suciedad, cortes, y la edad del usuario pueden provocar errores de identificación - Ciertas ocupaciones o actividades de forma temporal o permanente puede causar pérdida de la definición de huellas digitales que perjudica el funcionamiento	256 bytes – 2Kb	- Acceso a estaciones de trabajo - Sistemas internos donde los usuarios pueden ser entrenados apropiadamente, en un ambiente controlado
Geometría de la Mano	Forma y Tamaño de la Mano	- Precisión Moderada - Ofrece un buen equilibrio de características del rendimiento - Relativamente fácil de usar - Percibido por la mayoría como no intrusivo y no amenazante	- Lesión de las manos y la edad del usuario pueden producir errores - Limitaciones en la destreza manual pueden dar lugar a errores o no uso - Precisión limitada debido a las "simples" características - Las características pueden cambiar con el tiempo - Hardware de geometría de la mano tiene gran tamaño y no se puede utilizar en sistemas incrustados - Actualmente, sólo funciona en el modo de verificación - Algunos usuarios pueden sentirse incómodos tocando un dispositivo que mucha gente ha tocado previamente	9 bytes	- Control de acceso - Control de tiempo y asistencia

Tabla 1.3, Parte I. Comparación del algoritmo, Escenario, y las pruebas de funcionamiento

Tabla 1.3, Parte I. BIOMETRIA BASICA, *Manual de Aplicación de Tecnologías Biométricas*, Estados Unidos 2008

Comparación del algoritmo, Escenario, y las pruebas de funcionamiento, Parte II

Tipo	Medidas	Robusteces	Limitaciones	Tamaño de la Plantilla	Aplicaciones
Reconocimiento del Iris	Patrones del Iris	- Alta precisión - Utiliza lector basado en cámaras convencionales - Capaz de manejar grandes bases de datos - Distingue los gemelos idénticos - Alta velocidad, 1.000.000 comparaciones por segundo - Característica biométrica altamente distintivo - Funciona bien a través de anteojos y contactos, incluso con la diferencia de colores - Una de las pocas técnicas biométricas que funciona bien en el modo de identificación 1 a N	- Dispositivos de captura de alto costo - Erróneamente confundida con la iridología - No se ha demostrado que son adecuados para la vigilancia encubierta - Algunos usuarios no aceptan la tecnología eye-based	256 – 512 bytes	- Aplicaciones de alta seguridad - 1 a N registros sin PIN o P/W - Listas de observación - Beneficios de autorizaciones - Detección de duplicación de licencias de conducir - Apto para bases de datos muy grandes
Teclado	Se digita un Patrón	- Se puede ajustar los parámetros - No requiere hardware especializado - Combina la generación de contraseñas y la inscripción en una función simple	- No es único para cada individuo - Algunas personas no saben cómo escribir - Las grandes variaciones en los patrones de escritura de una persona - Baja precisión - Predominio de las tecnologías de biometría	84 – 2K bytes	- Seguridad para el ordenador o estaciones de trabajo
Huella de la Palma	Patrón de Huellas de la Palma	- Características únicas y estable a lo largo de la vida - Potencialmente tiene más funciones que las huellas digitales - Características más numerosas y la geometría de la mano es única	- El cristal de exposición debe estar limpio - Tocar donde otras personas han tocado - La aceptación del usuario es baja, debido al parecido con el fichado criminal - Hardware del sensor voluminosos	Datos no disponibles	
Escaneo de Retina	Patrones de los vasos sanguíneos de la Retina	- Alta precisión - Esta característica biométrica es única, es estable durante la vida, dificultad de falsificación	- Difícil de uso (proximidad, enfoque, sin gafas) - Los usuarios no se sienten cómodos con esta tecnología - las enfermedades como el glaucoma, la diabetes, la hipertensión y el SIDA, puede afectar el rendimiento - Disponibilidad comercial limitada - No es adecuado para aplicaciones encubiertas	96 bytes	- para aplicaciones de Alta seguridad, como por ejemplo accesos militares
Verificación de Voz	Patrón de código, tono y timbre de la voz	- La operación no requiere de ojos ni manos - Puede aprovechar la infraestructura telefónica - La flexibilidad lo hace adecuado para muchas aplicaciones - No requiere entrenamiento especial o equipo de usuario - Capas con contraseñas y PINs verbales	- Componente biométrico no distintivo y variará con la resfriado, dolor de garganta, el clima, el estado emocional y la edad - El ruido ambiental interfiere con el proceso - Variaciones de calidad en teléfonos, micrófonos y conexiones afectan a la precisión - Inscripción en varios canales y verificación afectan la precisión - No apto para sistemas de identificación 1: N - Potencialmente más vulnerable que otros sistemas biométricos - Utiliza más recursos de sistema que otros sistemas	6Kb-80Kb	- sistema de baja a mediana seguridad - Identificación de reclusos en aplicaciones de control telefonico de centros penitenciarios - 911 solicitudes - Aplicaciones de arresto domiciliario

Tabla 1.3, Parte II. Comparación del algoritmo, Escenario, y las pruebas de funcionamiento

Tabla 1.3, Parte II. BIOMETRIA BASICA, *Manual de Aplicación de Tecnologías Biométricas*, Estados Unidos 2008

1.3.1. Digitalización de huellas dactilares

El primer paso de un sistema de Biometría, es el registro de las características físicas y/o de conducta, estas son procesadas por un algoritmo numérico que depende del tipo de sistema biométrico, estos algoritmos pueden ser basados en puntos de minucia, en patrones o híbridos, en nuestro proyecto utilizamos este último ya que es más confiable al integrar la exactitud que brinda el algoritmo basado en puntos de minucia y la velocidad de comparación del algoritmo basado en patrones y para terminar esta información se almacena en una base de datos. Luego para el acceso, el sistema compara los algoritmos numéricos leídos con los existentes en la base de datos creada con anterioridad, si no concuerdan el sistema rechaza el acceso, caso contrario habilita el ingreso. Las tecnologías actuales tienen tasas de error variables que dependerán de la calidad y precisión de los dispositivos de lectura, los cuales oscilan entre 60% y 99,9%.

El rendimiento de una medida biométrica se define generalmente por la mínima relación de aceptación de falsos (*False Acceptance Rate* o FAR), es decir, el porcentaje de aceptación de una huella que no se encuentre en la base de datos como valida y la relación de rechazos de falsos (*False NonMatch Rate* o FNMR, también conocido como *False Rejection Rate* o FRR), es decir, la probabilidad de que un usuario verdadero pueda no ser reconocido y, por lo tanto, denegado su acceso.

En el proceso de autentificación (o verificación) los rasgos biométricos se comparan solamente con los de un patrón ya guardado, este proceso se conoce también como uno-a-uno (1:1); este proceso implica conocer presuntamente la identidad del

Fuente bibliográfica de la digitalización de huellas dactilares. BIOMETRIA BASICA, *Manual de Aplicación de Tecnologías Biométricas*, Estados Unidos 2008

individuo a autentificar, por lo tanto, dicho individuo ha presentado algún tipo de credencial, que después del proceso de autentificación biométrica será validada o no.

En el proceso de identificación los rasgos biométricos se comparan con los de un conjunto de patrones ya guardados, este proceso se conoce también como uno-a-N (1: N); este proceso implica no conocer la identidad presunta del individuo, la nueva muestra de datos biométricos es tomada del usuario y comparada una a una con los patrones ya existentes en el banco de datos registrados. El resultado de este proceso es la identidad del individuo, mientras que en el proceso de autentificación es un valor verdadero o falso.

El proceso de autentificación o verificación biométrica es más rápido que el de identificación biométrica, sobre todo cuando el número de usuarios (N) es elevado. Esto es debido a que la necesidad de procesamiento y comparaciones es más reducida en el proceso de autentificación. Por esta razón, es habitual usar autentificación cuando se quiere validar la identidad de un individuo desde un sistema con capacidad de procesamiento limitada o se quiere un proceso muy rápido.

1.3.2. Estándares de tecnologías biométricas

A nivel mundial el principal organismo que coordina las actividades de estandarización biométrica es el Sub-Comité 17 (SC17) del Joint Technical Committee on Information Technology (ISO/IEC JTC1), del International Organization for Standardization (ISO) y el International Electrotechnical Commission (IEC).

Existen además otros organismos no gubernamentales impulsando iniciativas en materias biométricas tales como: Biometrics Consortium, International Biometrics Groups y BioAPI. Este último se estableció en Estados Unidos en 1998 compuesto por las empresas Bioscrypt, Compaq, Iridiam, Infineon, NIST, Saflink y Unisis. El Consorcio BioAPI desarrolló conjuntamente con otros consorcios y asociaciones, un estándar que promoviera la interconexión entre los dispositivos biométricos y los diferentes tipos de programas de aplicación, además de promover el crecimiento de los mercados.

Los estándares más importantes son:

- Estándar ANSI / INCITS 358-2002

  Este estándar fue creado en el 2002 por ANSI y denominado BioAPI 1.1. (Biometric Application Programming Interface), es un interfaz de comunicación que fue desarrollado para garantizar que los producto y sistemas sean compatibles entre sí, utilizando los dispositivos existentes con programas de validación y además valida la interrelación entre el estándar BioAPI y otros estándares Biométricos.

Fuente bibliográfica de los estándares biométricos.
http://www.nist.gov/itl/csd/biometrics/bioapicts.cfm; abril del 2013

- CBEFF ó NISTIR 6529

  El estándar CBEFF (Common Biometric Exchange Formats Framework): este grupo de normas fue desarrollado por algunos organismos de normalización internacionales como son el Comité Internacional de Estándares de Tecnologías de la Información (INCITS) Comité Técnico M1 - Biometría y el Comité Técnico Conjunto ISO / IEC 1 (JTC 1) Subcomité SC 37 - Biometría, en donde se definen las estructuras de los datos básicos, conjuntos de elementos y los valores que ayudan el intercambio directo de datos biométricos en conformidad con el guardado de datos biométricos (BIRs), estos valores están diseñados para revelar el formato y otros atributos de los datos biométricos en el BIR sin exponer a los propios datos biométricos a las aplicaciones, apoyo a la seguridad de los datos biométricos.

  La versión original del este estándar CBEFF fue publicado como NISTIR 6529, fue desarrollado por un grupo de técnicos patrocinados por el NIST y el Consorcio Biométrico (BC) en coordinación con el Consorcio BioAPI, el Grupo de Trabajo X9F4, la Asociación Internacional de la Industria Biométrica y el grupo de interfaces de TeleTrusT.

- NISTIR 6529-A ó ANSI INCITS 398-2005

  El estándar NISTIR 6529-A fue publicado el 5 de abril del 2004 y es la versión mejorada del estándar NISTIR 6529 con mejor interoperabilidad y rendimiento, fue incorporada como Norma Nacional Americana a través del INCITS y fue publicada como ANSI INCITS 398-2005.

- ANSI 378

  Fue creado en el 2004 por la ANSI, el cual establece criterios para representar e intercambiar la información de las huellas dactilares a través del uso de minucias. El propósito de esta norma es que un sistema biométrico dactilar pueda realizar procesos de verificación de identidad e identificación, empleando información biométrica proveniente de otros sistemas.

- Estándares Internacionales

  En el 2003 se inicio el proyecto para desarrollar la versión internacional del CBEFF y algunas partes que conforman este estándar ya han sido publicadas en las normas ISO / IEC, las diferentes partes del estándar internacional ya publicadas son las siguientes:

  - Parte 1: ISO / IEC 19785-1, se refiere a las especificaciones de los elementos de datos, publicada en mayo del 2006, con enmienda del formato de intercambio de datos estándares ISO / IEC 197974-2.

  - Parte 2: Procedimientos para el funcionamiento de la autoridad de registro biométrico, publicado en mayo del 2006, con enmienda en los registros adicionales requeridos por las partes ISO / IEC 197974-2.

- Parte 3: Especificaciones del formato de patrones, publicado en diciembre del 2007, con enmienda referida a la representación de nuevos elementos de datos agregados por CBEFF en la parte 1.

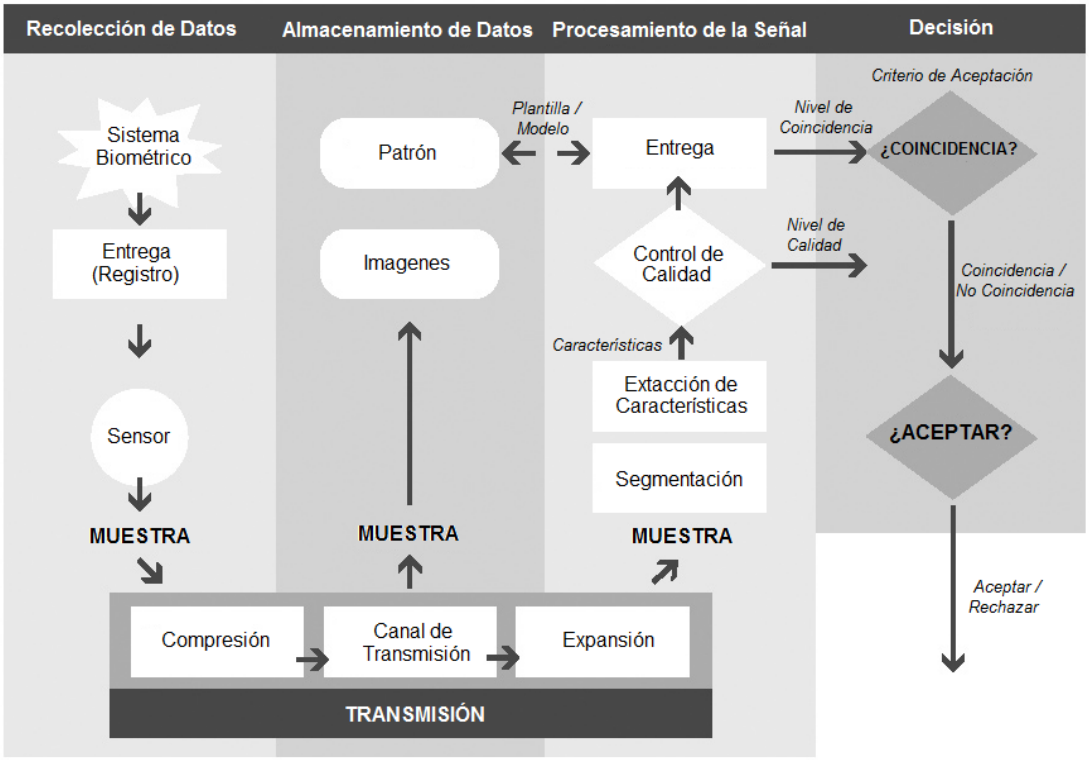
El lector escogido para la realización de este proyecto es de la marca Nitgen y el modelo es el FIM 5360 el cual cumple con los estándares vigentes al momento y soporta tanto el estándar ISO 197974-2 y el estándar ANSI 378.

1.3.3. Funcionamiento del sistema biométrico mediante huella digital

Todos los sistemas biométricos tienen una base de funcionamiento muy similar, para lo cual se toma como referencia la figura 1.2, en donde se ilustra las diferentes etapas de funcionamiento que poseen estos sistemas y se determinan cuatro procesos básicos que son la recolección de datos, el almacenamiento de datos, el procesamiento de la señal y la toma de decisión.

Cada una de estas etapas se detallan a continuación, haciendo referencia al sistema biométrico basado en huellas digitales que es la base de este proyecto:

Figura 1.2. Estructura de funcionamiento de un sistema biométrico general. **Fuente:** BIOMETRIA BASICA, *Manual de Aplicación de Tecnologías Biométricas*, Estados Unidos 2008

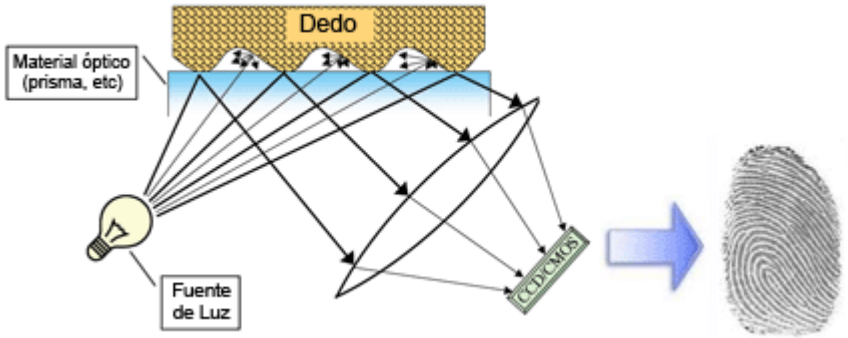


1) Recolección de datos: En esta primera etapa se determina el sistema biométrico a utilizar, el cual para este caso es mediante huella digital, mediante este sensor se genera un registro de la muestra.

Para esto el usuario coloca su dedo sobre el sensor, mediante una fuente de luz direccionada hacia el material óptico o prisma, refleja la imagen hacia el dispositivo que recibe la imagen, como se ilustra en la figura 1.3, esta información es comprimida y se la transmite dependiendo si el sensor se encuentra ingresando un nuevo usuario o validando la muestra con la información almacenada.

Figura 1.3. Captura de huellas digitales.

Fuente: http://www.squarenet.com.ec/conocimientocomofunciona.html



2) Almacenamiento de datos: para este proceso el sensor debe estar configurado como modo almacenamiento, luego toma la imagen obtenida y la transforma en datos matemáticos mediante un algoritmo que utiliza los puntos de minucia ya que tienen características únicas como tipo de punto, posición y dirección, es convertida en un modelo matemático llamado patrón.

Este patrón es almacenado en la memoria interna del dispositivo o en un dispositivo externo generando una base de datos, la misma que será donde se compare las huellas de los usuarios para su respectiva validación, como se aprecia en la figura 1.4.

Figura 1.4. Extracción o digitalización. **Fuente:** http://www.accesovip.co/biometria.html

3) Procesamiento de la señal: En esta etapa el sensor debe estar trabajando en el modo de verificación de datos, aquí los datos son segmentados y se extraen las características principales de las huellas.

Estas características son verificadas mediante un control de calidad de la muestra, de no existir una calidad óptima la misma es rechazada y no se realiza la validación, si la calidad de la muestra es buena se realiza la entrega a la etapa de decisión.

4) Decisión: Aquí se verifica si la información obtenida por la muestra tiene concordancia con alguna de las huellas guardadas previamente en la base de datos y se realiza la validación aceptando o rechazando la muestra.

1.4. Microcontroladores

El Microcontrolador es un Circuito integrado programable, que tiene las funciones básicas de un computador todo incorporado al interior del encapsulado, en donde consta el CPU (Unidad Central de Procesamiento), unidades de Entrada/Salida, memoria RAM y memoria ROM, adicionalmente posee un generador de reloj integrado; necesita una alimentación que oscila entre 3.3V y 5V.

En los microcontroladores encontramos terminales de entrada y salida, como puerto serial RS232, convertidor analógico - digital, temporizador.

Entre los microcontroladores más utilizados en nuestro medio se tiene los siguientes:

- PIC

- AVR
- ARM
- MSP430

Destacando principalmente el PIC que son los que se utilizará para la construcción de este proyecto.

1.4.1. PIC

Los microcontroladores PIC (*Peripheral Interface Controller*) son unos de los más populares de 8-bits, fabricados por Microchip Technology Inc. Microchip tiene diferentes tipos de microcontroladores desde pequeños en encapsulados SOT23 hasta los 84-PLCC, para el proyecto se utilizarán los PICs de las subfamilias 16F87XA y 16F8XA. Ellos pueden trabajar con frecuencias de hasta 40Mhz y ejecutar instrucciones cada 4 ciclos de reloj, las principales ventajas son:

- Software de desarrollo en assembler gratuito el MPLAB.
- Se encuentra una gran gama de dispositivos con diferentes tamaños de memorias y cantidad de periféricos.
- Los puertos GPIO pueden entregar hasta 20mA lo que permite manejar LED`s y otros dispositivos directamente.
- Hay disponibles para rangos de temperaturas extendidos de -40°C a +125ºC, lo cual es ideal para aplicaciones automotrices.

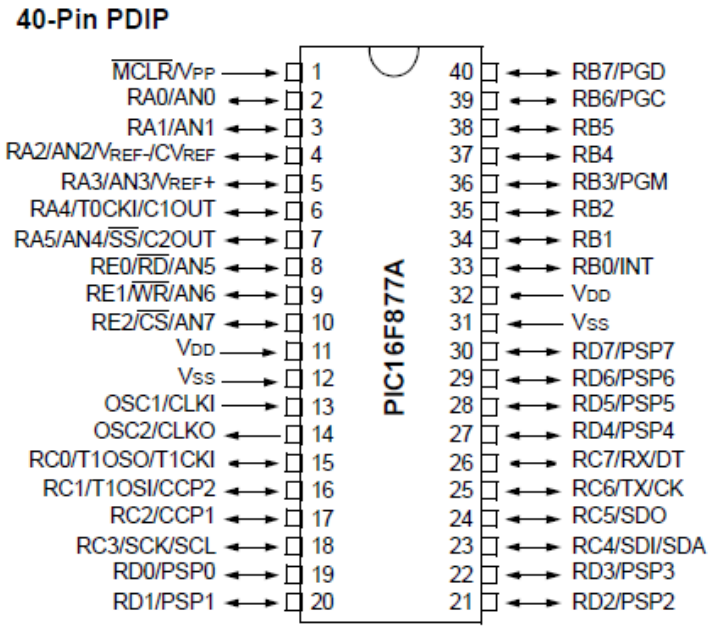
Fuente bibliográfica de los PIC. https://www.microchip.com/pagehandler/en-us/products/picmicrocontrollers; noviembre del 2012

1.4.1.A. PIC 16F877A

En el diseño de este proyecto se considero la utilización de dos microcontroladores, el más importante es el PIC16F877A que es donde se almacena el programa principal que controlará el sistema biométrico, por esto se debe analizar las características principales de este microcontrolador.

El PIC16F877A es el más completo de la familia PIC16F87XA de 8 bits, en encapsulado de 40 pines como se ilustra en la figura 1.5 con las funciones de cada pin, este brinda un alto rendimiento y facilidad en programación debido a que trabaja con solo 35 palabras de instrucciones, minimizando en gran medida el tiempo de programación, poseen módulos de comparación analógicos esto esta indicado por la letra A al final del código.

Figura 1.5. Encapsulado de 40 pines del PIC16F877A. **Fuente:** MICROCHIP, *Datasheet 16F87XA*/ Microchip Technology Inc. Dallas. 2010.



Fuente bibliográfica del PIC16F877A. MICROCHIP, *Datasheet 16F87XA*/ Microchip Technology Inc. Dallas. 2010

Las Características principales de este microcontrolador se detalla a continuación:

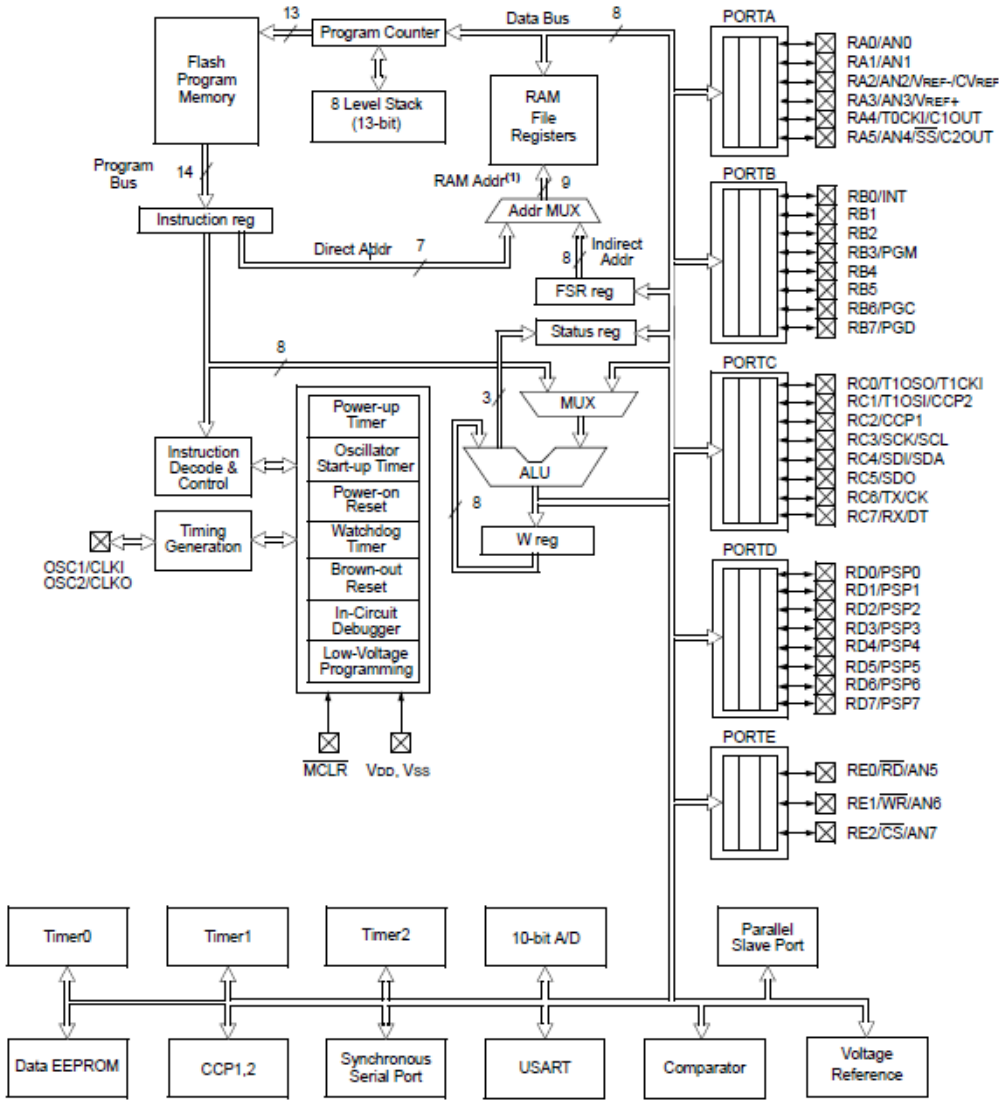
- Frecuencia de operación 20 MHz
- Tamaño de la memoria de programa de 14,3 KBytes
- Número de palabras de instrucción en la memoria de programa 8192
- Tamaño de la SRAM 368 x 8 Bytes
- Tamaño de la memoria flash EEPROM de alta velocidad 256 x 8 Bytes
- Número de interrupciones 15
- Puertos de entrada y salida A, B, C, D, E
- 3 Timers, 2 de 8 bits y 1 de 16 bits
- Puerto de comunicaciones seriales MSSP, USART
- Puerto de comunicaciones paralelo PSP
- Módulos de 10-bit analógico a digital, 8 canales de salida
- Comparadores analógicos 2
- Grupo de 35 instrucciones
- Modo de bajo consumo (Sleep).
- Tipo de oscilador seleccionable (RC, HS, XT, LP y externo).
- Rango de voltaje de operación desde 2,0V a 5,5V.
- Watchdog Timer o Perro Guardián.

1.4.1.A.1. Estructura interna

Este PIC tiene la memoria de datos y la memoria de programa por separado, esto se conoce como arquitectura Harvard, lo cual ayuda a acceder tanto a las instrucciones de programa como los datos simultáneamente mediante buses independientes, lo que

hace que la velocidad de procesos sea mucho mayor, esto se puede apreciar en la figura 1.6 en donde consta la estructura interna de este PIC, o en Figura 1.6.

Figura 1.6. Estructura interna del PIC16F877A. **Fuente:** MICROCHIP, *Datasheet 16F87XA*/ Microchip Technology Inc. Dallas. 2010



1.4.1.A.2. Circuitería externa adicional

Una parte importante para el funcionamiento del microcontrolador es la circuitería externa, como por ejemplo el circuito de alimentación, el reloj oscilador, el circuito de reinicio y circuitos de salida que comandan los relés para realizar el bloqueo del vehículo, los cuales se analizará brevemente.

- **Circuito de alimentación**

  Una ventaja importante de la familia PIC16F87X es que admiten un rango amplio para las tensiones de alimentación, el cual va desde 2,0 V a 5,5 V. Esta tensión determina la frecuencia máxima de trabajo.

  Para este proyecto, se trabajará con la tensión de 12 V existente en la batería de los vehículos, la cual pasa por un circuito regulador que hace que se disminuya la tensión a 5 V mediante un regulador 7805.

  Existe una fórmula para calcular la potencia máxima que requiere el sistema, esto se puede analizar en la ecuación 1.1, la misma la encontramos a continuación, es importante conocer que para este tipo de PICs la potencia máxima es de 1 W:

  Ecuación 1.1

$$P_{disipada} = V_{DD}\,(I_{DD} - \sum I_{OH}) + \sum [(V_{DD} - V_{OH})\, I_{OH}] + \sum (V_{OL}\ \ I_{OL})$$

donde,

- V_{DD} = Tensión de la fuente de alimentación, en voltios (V) .
- I_{OH} = Corriente entregada por las salidas del PIC en estado de plena carga, en amperios (A).
- I_{OL} = Corriente consumida por las salidas del PIC en estado de baja carga, en amperios (A).
- V_{OH} = Tensión entregada por los terminales en estado de plena carga, en voltios (V).
- V_{OL} = Tensión de los terminales en estado de baja carga, en voltios (V).

- **Circuito de reloj**

Los microcontroladores trabajan con un reloj u oscilador que se utiliza para generar la base de tiempo mediante pulsos de voltaje intermitentes, estos dispositivos relacionan estos pulsos con el código binario que es con el que trabaja el lenguaje de ensamblador, de no existir estos pulsos no se podría generar ningún comando. Para el ingreso de la conexión del oscilador se emplean los puertos OSC1 y OSC2 del PIC.

El microcontrolador PIC16F877A utiliza por cada ciclo de instrucción cuatro ciclos de reloj. Esto se interpreta de la siguiente manera ayudados con la ecuación 1.2, el tiempo total que se empleará para ejecutar un programa es por ejemplo, si el PIC es posee 1000 instrucciones, con un reloj de 20 MHz ó período del reloj de 50 ns, es decir, 1 sobre la frecuencia, se tendría.

Ecuación 1.2

$$T = \frac{I * Cr}{F}$$

donde,

T = Tiempo total de ejecución del programa, en microsegundos (µs).

I = Número de Instrucciones, valor entero

Cr = Número de ciclos del reloj, valor entero

F = Frecuencia, en megaherz (MHz)

reemplazando,

$$T = \frac{1000 * 4}{20 * 10^6} = 200\ \mu s$$

como resultado se tiene que el tiempo de ejecución del programa es de 200 µs y el tiempo de ejecución por ciclo es de 200 ns.

Para generar una señal de reloj se puede hacer mediante un circuito de resistencias-condensadores, resonadores cerámicos o cristales de cuarzo piezoeléctrico, con este último se obtienen frecuencias de oscilación muy exactas, lo cual permite calcular tiempos de ejecución de programas y temporizaciones precisas.

- **Circuito de reinicio**

El pin que se utiliza para realizar el reinicio del programa del PIC es el Master Clear (MCLR), este debe estar en valor lógico alto (1) para que funcione normalmente. Con un valor lógico bajo (0) el microprocesador se reinicia y se ejecuta nuevamente el programa grabado en el PIC desde el principio.

Para realizar el reinicio del PIC, lo más sencillo es un reinicio manual que puede ser mediante un pulsante en serie con una resistencia entre 50 a 100 Ω para prevenir que existan corrientes inducidas que pueden bloquear al microcontrolador, además se puede conectar un condensador en paralelo que estabilice el pulso, para evitar que al presionar el pulsante se genere rebotes que pueden afectar al PIC.

Adicionalmente para el reinicio se puede utilizar diferentes tipos de circuitos externos que se adecúen al diseño del proyecto y que por ejemplo también puede darse desde una opción del mismo programa cargado en el microcontrolador.

- **Circuito de comando de relés**

Una parte importante para la realización de este proyecto es el controlar los bloqueos del vehículo, para prevenir que se pueda encender sin la validación

del sistema biométrico y un bloqueo importante que se ha considerado es al encendido, esto se lo puede hacer mediante el comando de relés.

Para poder controlar los relés se utiliza los puertos de salida del PIC, pero debido a la corriente que se necesita para habilitar a estos, la interconexión se la realiza mediante circuitos integrados optoacopladores, que ayudan a manejar diferentes voltajes y corrientes sin afectar al funcionamiento del PIC.

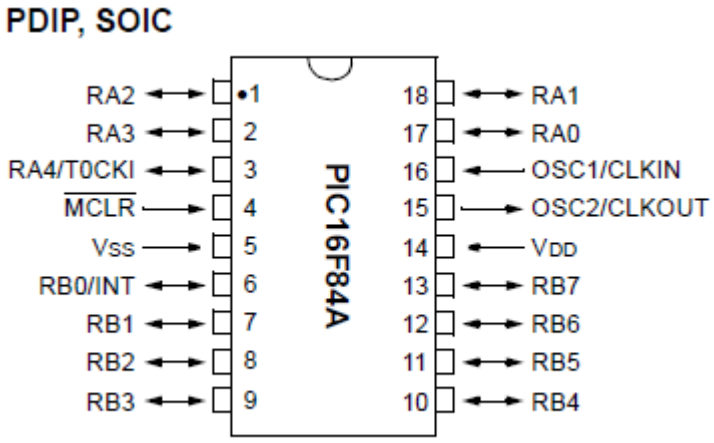
1.4.1.B. PIC 16F84A

El siguiente microcontrolador que se va a utilizar para el proyecto es el PIC16F84A, que tiene menos opciones que el PIC visto anteriormente, pero que consta con características similares, las cuales se analizaran brevemente.

Este PIC es de 8 bits que permite una comunicación optima con el sistema, el encapsulado es de 18 pines, como se muestra en la figura 1.7, en donde se indica también la función de cada uno de los pines.

Fuente bibliográfica del PIC16F84A. MICROCHIP, *Datasheet 16F84A*/ Microchip Technology Inc. New York. 2009

Figura 1.7. Encapsulado de 18 pines del PIC16F84A. **Fuente:** MICROCHIP, *Datasheet 16F84A*/ Microchip Technology Inc. New York. 2009



A continuación se detalla las características principales de este microcontrolador:

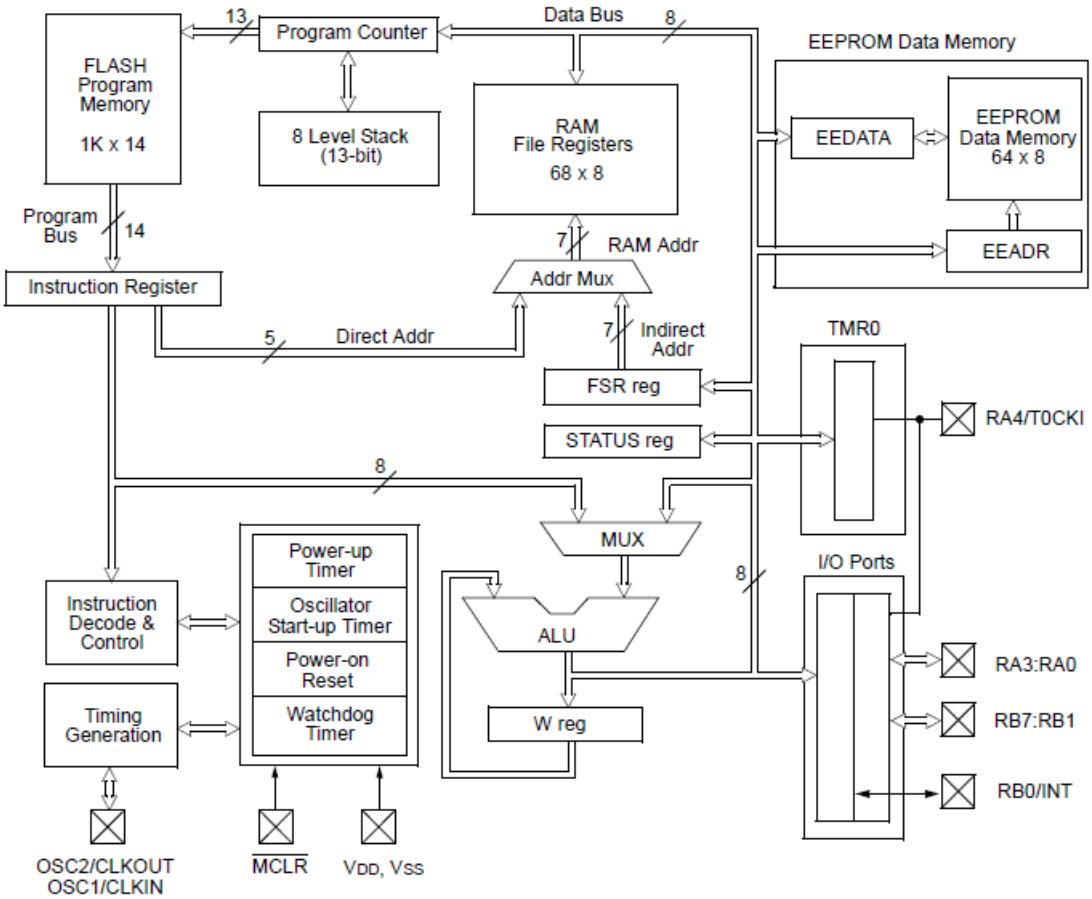
- Frecuencia de operación 20 MHz
- Número de palabras de instrucción en la memoria de programa 1024
- Tamaño de la SRAM 68 Bytes
- Tamaño de la memoria flash EEPROM de alta velocidad 64 Bytes
- Número de interrupciones 15
- Puertos de entrada y salida A, B
- 1 Timers de 8 bits
- Puerto de comunicaciones seriales MSSP, USART
- Módulos de 10-bit analógico a digital
- Grupo de 35 instrucciones
- Modo de bajo consumo (Sleep).
- Tipo de oscilador seleccionable (RC, HS, XT, LP y externo).
- Rango de voltaje de operación desde 2,0V a 5,5V.
- Watchdog Timer o Perro Guardián.

Como se puede apreciar, este PIC en relación al PIC16F877A, tiene opciones similares pero con limitaciones sobre todo en los puertos de entrada y salida, la capacidad de memoria RAM y EEPROM, que para las aplicaciones destinadas en este proyecto cumple con los requerimientos necesarios.

1.4.1.B.1. Estructura interna

De igual manera como se pudo apreciar en las características de este PIC que tiene menos opciones que el microcontrolador principal del proyecto, la estructura interna es también más simple como se aprecia en la figura 1.8.

Figura 1.8. Estructura interna del PIC16F84A. **Fuente:** MICROCHIP, *Datasheet 16F84A*/ Microchip Technology Inc. New York. 2009



1.5. Elementos a bloquear en un vehículo

Existen muchas maneras de bloquear un vehículo utilizando técnicas tanto mecánicas como de control eléctrico y electrónico, en la actualidad los dispositivos más utilizados a bloquear en un vehículo son:

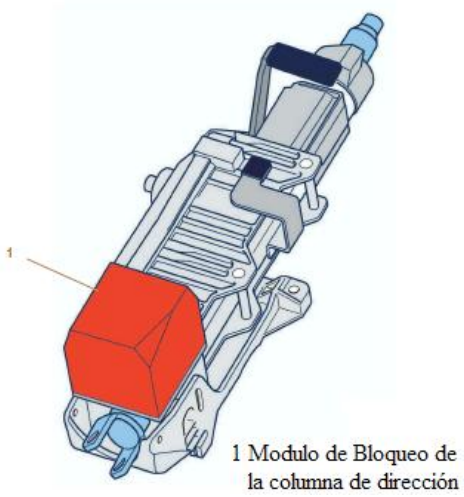
1.5.1. Columna de la dirección

Este sistema mecánico cumple con la función de bloquear el volante a través de un pasador metálico, ubicado en la columna de dirección, y se activa en el momento en que se gira el volante en cualquier dirección. Para desbloquearlo es necesario introducir la llave y girarla dentro del interruptor, al mismo tiempo que se mueve el volante, para una mejor comprensión se puede tomar como referencia las figuras 1.9 y 1.10.

Existen dispositivos que realizan el bloqueo y desbloqueo de la columna de dirección mediante mandos de control eléctrico. Estos sistemas constan de un computador de control de carrocería, de un modulo de bloqueo que se encuentra instalado sobre la columna de dirección.

Fuente bibliográfica de los elementos a bloquear en un vehículo. NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010

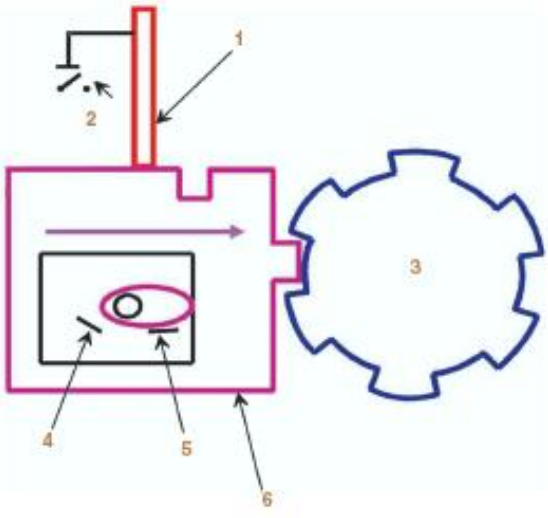
Figura 1.9. Bloqueo de la columna de dirección. **Fuente:** NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010



Funcionamiento

Posición reposo (antiarranque activo, columna bloqueada)

Figura 1.10. Columna de dirección bloqueada. **Fuente:** NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010

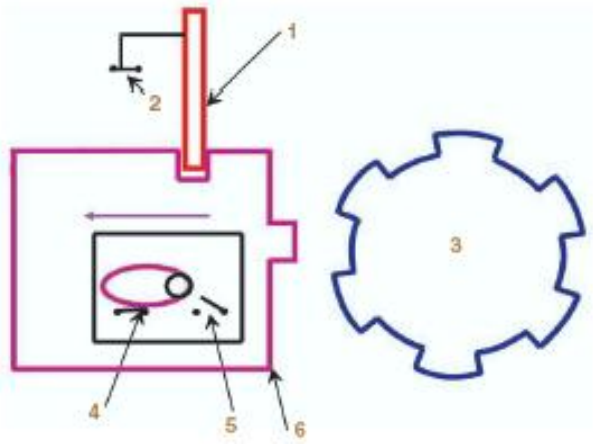


1 Bloqueo del pivote
2 Contactor de bloqueo del pivote
3 Columna de dirección
4 Contactor de pivote metido
5 Contactor de pivote armado
6 Pivote del bloqueo de la columna

Cuando el sistema está en reposo y el antiarranque del vehículo se encuentra activo, el pivote (6) bloquea la columna de dirección (3). El contactor de pivote armado (5) está cerrado para informar al computador de control que el sistema se encuentra en posición de reposo (columna bloqueada).

Posición columna desbloqueada

Figura 1.11. Columna de dirección desbloqueada. **Fuente:** NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010



1 Bloqueo del pivote
2 Contactor de bloqueo del pivote
3 Columna de dirección
4 Contactor de pivote metido
5 Contactor de pivote armado
6 Pivote del bloqueo de la columna

El computador de control de carrocería envía una orden al modulo de bloqueo de columna de dirección de desbloquearse. Esta orden corresponde a la emisión de un código enviado cuando el antiarranque ha sido desactivado; si el código es reconocido por el modulo de bloqueo, este último libera la columna de dirección, como se aprecia en la figura 1.11.

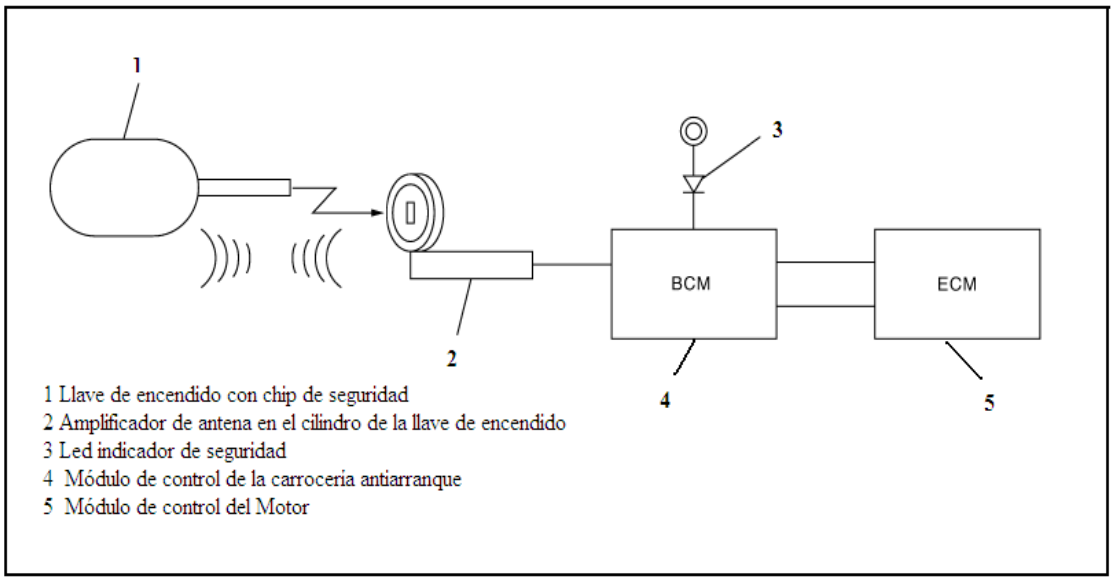
El contactor pivote metido (4) está cerrado indicando que el sistema se encuentra en posición desbloqueado, el pivote está bloqueado (1) y el contactor de bloqueo del pivote (2) está cerrado.

1.5.2. Bloqueo al encendido

Existe un mecanismo de bloqueo al encendido denominado antiarranque, compuesto por varios elementos electrónicos que se comunican entre sí para evaluar si la llave que ha entrado en el cilindro de encendido es la que está codificada para ese vehículo.

Para poder realizar la tarea de bloquear el encendido del motor, el sistema antiarranque cuenta con unos elementos básicos: emisor, amplificador, decodificador y calculador de inyección, los cuales se pueden ver en la figura 1.12.

Figura 1.12. Sistema de antiarranque. **Fuente:** NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010



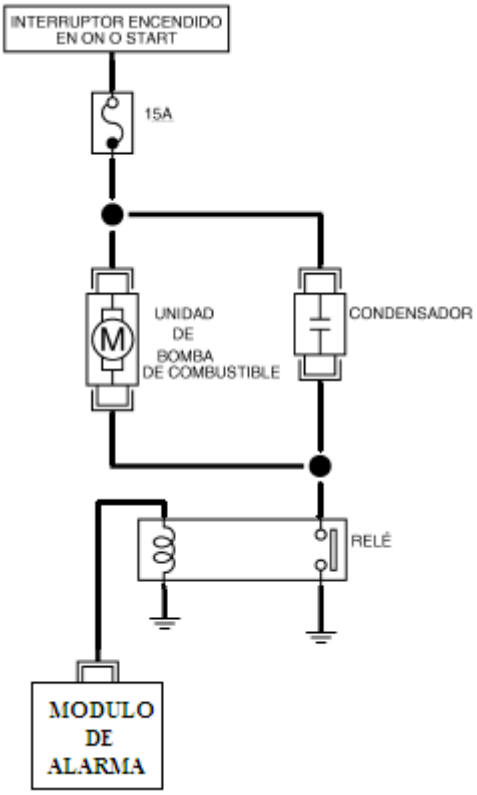
El sistema trabaja cuando la llave de encendido (1) envía una señal mediante el chip encriptado que se encuentra en la misma hacia la antena (2) que se encuentra en el cilindro de la llave, esta señal es amplificada por la antena y es enviada al Modulo de control del motor (4) para ser decodificada. Esta señal al ser decodificada es

comprobada por el modulo y si es la correcta se envía la orden al Modulo de control del motor (5) de autorizar el encendido, activa los relés de bomba de gasolina y el relé de control de inyección lo que permite el encendido del vehículo. El sistema cuenta con un led indicador de seguridad (3) el cual informa el estado del sistema ya sea activo o inactivo. Las llaves que se necesiten usar en este sistema tendrán que ser registradas es decir que su identificación tendrá que ser aprendida por el BCM y ECM para poder encender el vehículo, el máximo de llaves a ser aprendidas dependerá de cada sistema y de las configuraciones de sus diseñadores.

1.5.3. Bloqueo a la alimentación de combustible

Este sistema de bloqueo es comúnmente utilizado por diferentes tipos de alarmas, mediante el comando de relés se realiza el corte directo a la línea negativa del circuito de alimentación de la bomba de combustible, provocando ausencia de carburante en la rampa de inyección. Este es un método de bloqueo fácil de realizar por lo que es uno de los más utilizados en instalaciones de alarmas pero no es el más seguro ya que se puede detectar fácilmente y no garantiza la seguridad del vehículo, este tipo de bloqueo se lo puede apreciar en la figura 1.13.

Figura 1.13. Corte a la bomba de combustible. **Fuente:** NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010

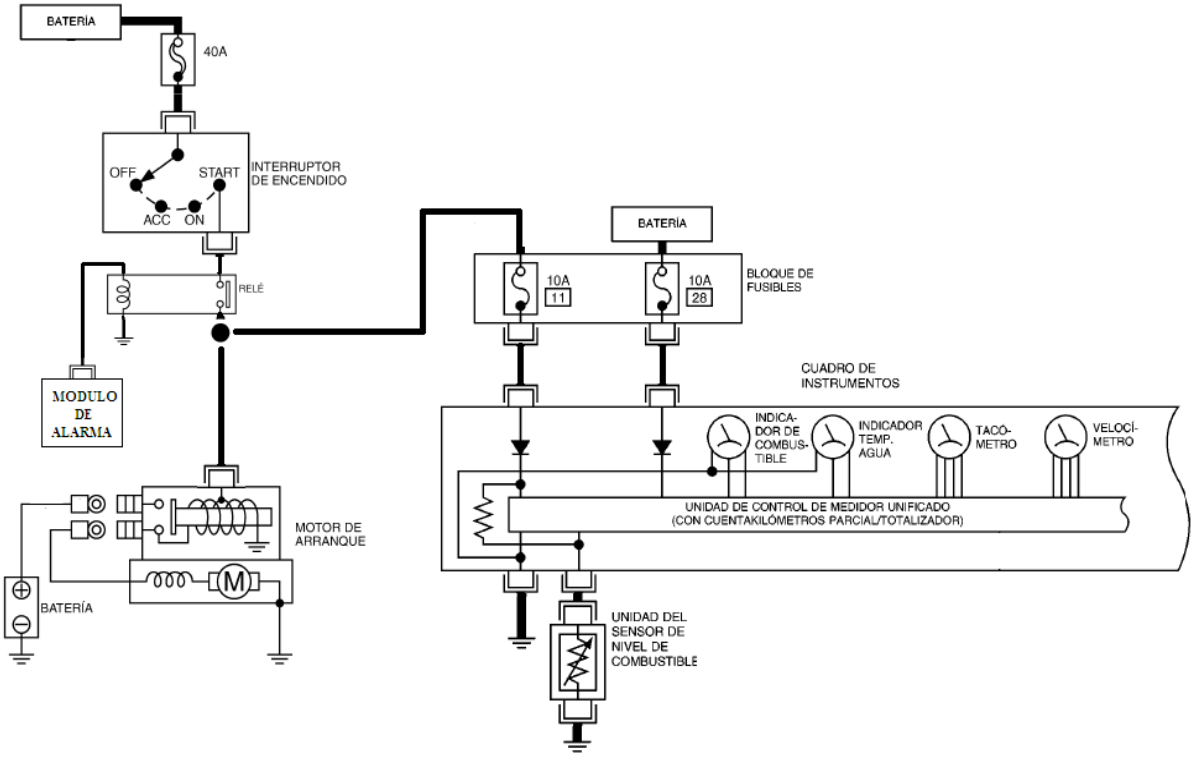


1.5.4. Bloqueo al arranque

Es el método de bloqueo más utilizado en la actualidad en sistemas de alarmas para vehículos ya que presenta esquemas eléctricos de fácil instalación y de difícil localización para ser desactivado por lo que presenta un grado de seguridad mayor. Estas ventajas han hecho que este sistema sea el que se va a utilizar para el desarrollo del proyecto, su instalación se realiza mediante el corte de energía en la alimentación de 12 V después del contacto lo que impide el accionamiento de elementos eléctricos como el motor de arranque, tablero de instrumentos, etc. Los mismos que se pondrán en funcionamiento solo cuando haya la autorización del modulo de alarma.

El diagrama eléctrico de este sistema está representado en la figura 1.14:

Figura 1.14. Bloqueo al arranque y cuadro de instrumentos. **Fuente:** NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010



1.6. Conclusiones

En este capítulo se ha analizado los aspectos generales de los sistemas biométricos, las ventajas y desventajas entre los diferentes sistemas que existen y cuál es la mejor alternativa de sistema biométrico para este proyecto, su funcionamiento y características, se hablo sobre los microcontroladores y se determinó los mejores dispositivos que respaldan al proyecto, así como también las maneras de poder bloquear un vehículo.

CAPÍTULO II

DISEÑO DEL SISTEMA DE SEGURIDAD BIOMÉTRICO

2.1. Introducción.

Existen varios sistemas de identificación biométrica utilizados en la actualidad que van a depender del propósito requerido, en general todos se utilizan para garantizar mayor seguridad y confianza en temas de identificación y validación de usuarios.

Es necesario realizar un análisis que permita determinar los parámetros de diseño según los alcances delimitados para este proyecto cuyo objetivo principal es mejorar los sistemas de seguridad ya existentes en los vehículos.

2.2. Sensor de huella digital

La biometría se basa en la verificación de la identidad de una persona dependiendo de características únicas de cuerpo o de comportamiento, como se ha revisado, son varios los métodos utilizados en sistemas biométricos.

Era necesario analizar detenidamente los parámetros de cada sistema para poder determinar un sistema en base a fiabilidad, ventajas, desventajas y prestaciones. De todos los sistemas de identificación biométrica analizados las huellas dactilares es el

Fuente bibliográfica del sensor de huella digital. NITGEN CO. LTD, *Datasheet Nitgen Fim5360* Version 1.02, Korea 2011

que se adapta a los parámetros de diseño planteados ya que además de ser efectivo, es cómodo de aplicar y la autentificación se obtiene rápidamente.

2.2.1. Fiabilidad.

Una de las maneras más comunes de distinguir huellas digitales es a través del patrón que siguen sus líneas y surcos y se los puede clasificar según tres rasgos mayores, que se indican en la figura 2.1.

Figura 2.1. Patrones de las clasificaciones de huellas digitales. **Fuente:** BIOMETRIA BASICA, *Manual de Aplicación de Tecnologías Biométricas*, Estados Unidos 2008.

Por otro lado, en determinados puntos las líneas de huellas dactilares se cortan bruscamente o se bifurcan, esto da lugar a la formación de puntos conocidos como minucias y juntos llegan a formar casi el 80% de los elementos singulares de una huella que da lugar a un patrón único para cada individuo que es distinto incluso en gemelos idénticos, se estima que la probabilidad que una huella se repita es de 1 en 64.000 millones por lo que sistemas de huellas dactilares son muy fiables pues son prácticamente inviolables, un parámetro básico a tener en consideración para elegir este sistema biométrico en la delimitación de este proyecto.

En el proceso de diseño del proyecto es fundamental el análisis de ventajas y desventajas del sistema de huella digital que es la mejor manera de definir si el sistema elegido es adecuado a las características de diseño planteadas.

2.2.2. Ventajas.

- No se puede adivinar un patrón de huella digital como se puede adivinar una contraseña.
- Universalidad alta ya que es casi improbable la ausencia de algún dedo o de una o ambas manos.
- Sensores de huella digital de bajo costo y gran variedad.
- Alta aceptabilidad por los usuarios debido a la larga tradición de uso de huellas dactilares.
- Alta permanencia pues es casi improbable el deterioro de una huella digital con el pasar del tiempo.
- Buenas prestaciones debido a que se cuenta con algoritmos eficientes y precisos de comparación entre huellas dactilares y no se requiere gran cantidad de espacio para el almacenamiento de los puntos de minucias.
- Alta precisión, seguridad y fiabilidad.
- Sistemas de fácil ergonomía y utilización.

2.2.3. Desventajas.

- Existe la probabilidad de que un corte, resequedad de la piel, suciedad, contacto de las manos con agentes químicos puedan ocasionar un error en la lectura de la huella, por eso se recomienda que un mismo usuario tenga al menos dos huellas registradas.
- En casos esporádicos se puede dar la posibilidad que se asocie el uso de huellas dactilares con criminalidad o invasión a la intimidad.

2.2.4. Prestaciones.

Los sistemas de identificación por huellas dactilares se han venido utilizando desde el siglo pasado y por su efectividad en la actualidad se han desarrollado ampliamente ya que es una tecnología altamente confiable ya que es casi imposible su falsificación.

Al ser una tecnología en desarrollo los costos de los sistemas de huella digital son relativamente baratos, se trata de una tecnología básica bien desarrollada y probada con buenos resultados en los que la autentificación y verificación de usuarios se refiere.

Son sistemas ergonómicos por lo que pueden ser desarrollados en una variedad de entornos que fue unas de las razones por las que se eligió este sistema para el proyecto.

2.3. Características del sensor de huella digital utilizado

En base a este análisis y luego de buscar las mejores opciones se eligió el modulo FIM 5360 de la compañía Coreana NITGEN, que es líder en la industria del reconocimiento de huellas digitales.

FIM 5360 es un dispositivo de identificación de huella dactilar de excelentes características y buenos beneficios como alta calidad de identificación, consumo de bajo poder e interfaces seriales UART de comandos de fácil integración con un amplio rango de aplicaciones. Es un dispositivo resistente y compacto con un modulo de identificación de huella digital que contiene un sensor óptico dentro, las características principales se aprecian en las tablas detalladas a continuación.

Tabla 2.1. Especificaciones principales del sensor FIM 5360. **Fuente:** NITGEN CO. LTD, *Datasheet Nitgen Fim5360* Version 1.02, Korea 2011

ESPECIFICACIONES PRINCIPALES DEL SENSOR FIM 5360		
ITEM		**FIM5360**
Especificaciones de Memoria	CPU	S3C2410 (ARM9 266Mhz)
	DRAM	16MByte SDRAM
	FLASH ROM	8Mbyte
Dimensiones		43 x 60 [mm^2]
Sensor		NITGEN OPP06
Voltaje de Alimentación		5 / 3.3 [V]
Consumo de Corriente	Normal	70 [mA]
	Máxima	220 [mA]
Temperatura de operación		-20 ~ 60 [°C]
Humedad		~ 90 [% RH]
Canal de Comunicación		RS-232 level UART
		Speed: 9600 ~ 115200 [bps]
		(1 start bit, 8 data bit, no parity, 1 stop bit)
Almacenamiento máximo de usuarios		1000 Usuarios

Tabla 2.2. Especificaciones de operación. **Fuente:** NITGEN CO. LTD, *Datasheet Nitgen Fim5360* Version 1.02, Korea 2011

ESPECIFICACIONES DE OPERACIÓN	
ITEM	**FIM5360**
Velocidad de Captura	0.2 [s]
Velocidad de Verificación	Menor a 1 [s]
Tiempo de Arranque	0.4 [s] para 100 usuarios
	0.5 [s] para 1000 usuarios
Método de Encriptación de Datos	AES para guardar datos
	AES para comunicación DB

Tabla 2.3. Características del sensor. **Fuente:** NITGEN CO. LTD, *Datasheet Nitgen Fim5360* Version 1.02, Korea 2011

CARACTERÍSTICAS DEL SENSOR	
OPP – 6	
Nombre del Sensor	OOP - 6
Tipo de Detección	Óptico
Área de Detección	15.0mm x 18.5mm
Resolución de Imagen	500 DPI
Tamaño de Imagen	260 x 300

2.4. Método de programación

Existe gran cantidad de compiladores de alto rendimiento utilizados para Microcontroladores PIC de MICROCHIP, por lo que se ve la necesidad de escoger un compilador de fácil aprendizaje y que cuente con todas las características necesarias para la realización del software.

Por esta razón se decidió utilizar el compilador MikroBasic de la empresa Mikroelectronic, debido a que basa su lenguaje de programación en BASIC que está

diseñado para desarrollar, construir y depurar aplicaciones basadas en PIC. Cuenta con una gran variedad de características como: Un código muy compacto y eficiente, variedad de bibliotecas de software, documentación completa, un simulador de software, un depurador de hardware, generación de archivos *.COF (*Code Object File Format. Es generado para fines de simulación o depuración de software del programa*.), IDE fácil de usar (*Integrated Device Electronics. Estándar de interfaz para la comunicación de dispositivos de almacenamiento de datos.*), además de una sintaxis BASIC es de fácil aprendizaje que incluye varios ejemplos prácticos que permiten un rápido inicio en la programación de microcontroladores PIC.

2.5. Microcontroladores utilizados

Existe una gran diversidad de microcontroladores desarrollados hoy en día, por lo que a la hora de escoger un microcontrolador se debe tener en cuenta los requisitos de diseño y considerar parámetros fundamentales como:

2.5.1. Procesamiento de datos

Se debe analizar la precisión de datos al manejar y considerar si es necesario que el microcontrolador realice cálculos críticos en tiempos limitados, ya que si no es suficiente un microcontrolador de 8 Bits, será necesario acudir a uno de 16 y 32 Bits lo que elevaría el coste del diseño.

2.5.2. Entrada - Salida

La mejor manera de verificar este parámetro es realizando un diagrama de bloques en donde se pueda identificar la cantidad y tipo de señales a controlar. De este análisis se puede determinar si es necesario añadir periféricos de hardware o cambiar a otro microcontrolador más adecuado al diseño.

2.5.3. Consumo

En el caso de que se incorporen microcontroladores alimentados por baterías y en el que su funcionamiento sea vital como activar una alarma antirrobo, el microcontrolador puede encontrarse en estado de bajo consumo y que despierte ante la activación de una señal (*una interrupción*) y ejecute el programa adecuado para procesarla.

2.5.4. Ancho de palabra

Se maneja el criterio de diseño en el que se debe seleccionar el microcontrolador con menor ancho de palabra que satisfaga los requerimientos de la aplicación, ya que el uso de un microcontrolador de 4 Bits significaría una reducción importante en los costos, y uno de 8 Bits sería el más adecuado si el ancho de los datos es de un Byte.

2.5.5. Memoria

Para analizar las necesidades de memoria de nuestro diseño se debe primero separar los tipos de memoria en: memoria volátil (RAM), memoria no volátil (ROM, EPROM, etc.) y memoria no volátil modificable (EEPROM). Este último tipo de memoria puede ser muy útil en el diseño para incluir información específica como un número de serie o parámetros de calibración.

Es imprescindible realizar una versión preliminar del programa y a partir de ella realizar una determinación de cuanta memoria volátil y no volátil se necesita y si es conveniente disponer de memoria no volátil modificable.

2.5.6. Determinación de diseño

Luego de considerar todos los parámetros que requiere el proyecto se determina que la mejor opción es trabajar con dos microcontroladores PICs, el primero es el PIC16F877A, que es el que manejara el programa principal administrando los periféricos e integrando todos los módulos que conforman el sistema de seguridad; definimos para el segundo PIC el PIC16F84A, que es el que manejará el módulo celular para notificaciones.

2.6. Diseño de la placa

En el tema de diseño es muy importante considerar este parámetro ya que la elección de un microcontrolador puede condicionar el diseño de la placa de circuitos. Se debe pensar que tal vez la elección de un microcontrolador barato, encarezca el resto de componentes del diseño, así como también se debe tener en consideración el espacio físico con el que se cuenta ya que si se tiene que colocar en espacios reducidos como es el caso de este proyecto la optimización de espacio es una necesidad.

En el diseño de la placa de circuitos es necesario que sea lo más pequeño posible de tal manera que se pueda instalar en cualquier lugar en un vehículo y que el espacio no sea un impedimento.

2.7. Conclusiones

Para el diseño del sistema de seguridad biométrico se ha considerado todas las variables que se necesitan tener en cuenta para poder realizar un diseño adecuado de un proyecto, se tomo en cuenta el lector de huella digital más apropiado para este proyecto, con respecto a los microcontroladores se ha considerado las características de memoria, capacidad, dispositivos de entrada y salida, que permitan una excelente comunicación con el lector de huella digital y demás periféricos, para la programación de los PICs se analizó de la misma manera un software versátil y con buenas prestaciones de trabajo.

CAPÍTULO III

DISEÑO, CONSTRUCCIÓN Y ELABORACIÓN DEL HARDWARE Y SOFTWARE DEL SISTEMA DE SEGURIDAD BIOMÉTRICO

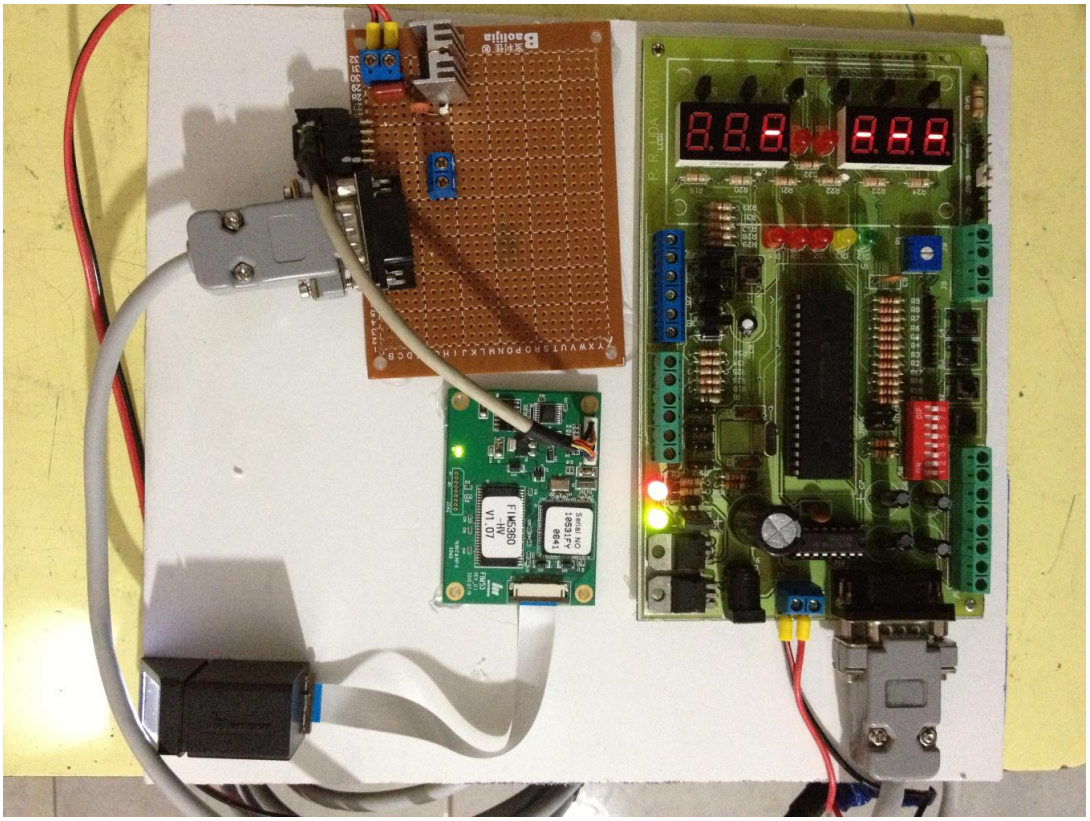
3.1. Introducción

Luego del análisis de los parámetros de diseño que sustentan este proyecto realizado en el capítulo anterior, en donde se definió los dispositivos más adecuados y con las mejores prestaciones para lograr un sistema de seguridad robusto y eficiente; para este capítulo se diseñará los módulos de hardware que conforman todo el proyecto, el software de programación de los PICs y se realizará el diseño de la placa definitiva que integra todos los módulos del sistema biométrico.

3.2. Diseño del hardware

Para el diseño del hardware, se comenzó definiendo los diferentes módulos que conforman el proyecto y procedimos a elaborar cada uno de ellos para poder realizar las diferentes programaciones y pruebas respectivas antes de diseñar la placa definitiva del proyecto, este prototipo se encuentra en la figura 3.1.

Figura 3.1. Prototipo de prueba



Los diferentes módulos que se han diseñado son los siguientes:

- Módulo central
- Módulo de alimentación
- Módulo de mando para bloqueos
- Módulo de control celular
- Módulo del lector de huella digital

3.2.1. Módulo central

El módulo central es el más importante debido a que este maneja todo el sistema y es aquí donde se encuentra el microcontrolador principal el PIC 16F877A, para realizar este diseño aprovechamos el módulo de entrenamiento que ya se tenía elaborado para los proyectos de laboratorio de microcontroladores, este se encuentra en la figura 3.2.

Una vez realizadas las pruebas del sistema de seguridad biométrico, se continua con el diseño para construir la placa principal, en donde está el PIC 16F877A con las interconexiones a todos los módulos que conforman el proyecto, se utiliza un cristal de cuarzo para generar el pulso de reloj, que para este caso es de 16MHz debido a que se requiere que el programa corra de forma rápida y como se analizó anteriormente que a mayor frecuencia, se obtiene menor tiempo de ciclo de instrucción, además se encuentra también la entrada de teclado para lo cual se han colocado las resistencias de 10 KΩ mediante el puerto B del PIC, esto se lo realiza debido a que se necesita un 0 lógico para el pulso del teclado, pero si se conecta directamente se produciría un corto circuito entre los 5 Vcc del puerto y tierra, para esto se coloca estas resistencias, adicionalmente están los indicadores mediante diodos leds y los puertos de entrada y salida.

Existen los Input 1, 2 y 3, que para este proyecto se usaran de la siguiente manera el input 1 para recibir la señal del encendido del vehículo que es el que habilita la lectura del sensor para la comparación de los usuarios, los input 2 y 3 quedan de reserva para alguna aplicación adicional que se desee implementar, estos trabajan

con el puerto A del PIC, adicionalmente y mediante el puerto C se comanda las salidas para el módulo de bloqueos, este también se usa para comandar el módulo celular y debido a que posee la conexión serial se comanda también el módulo del lector de huella digital, como se puede apreciar en la figura 3.3, o en Figura 3.3.

Figura 3.2. Módulo de entrenamiento

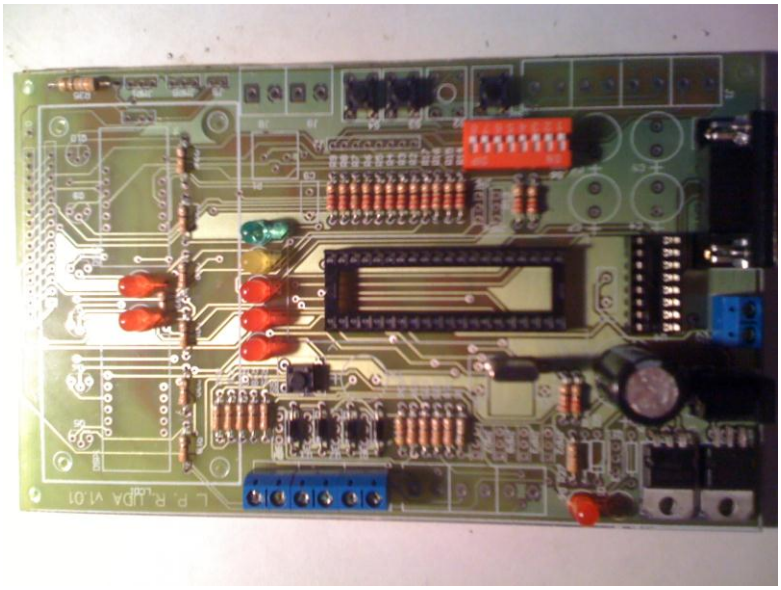
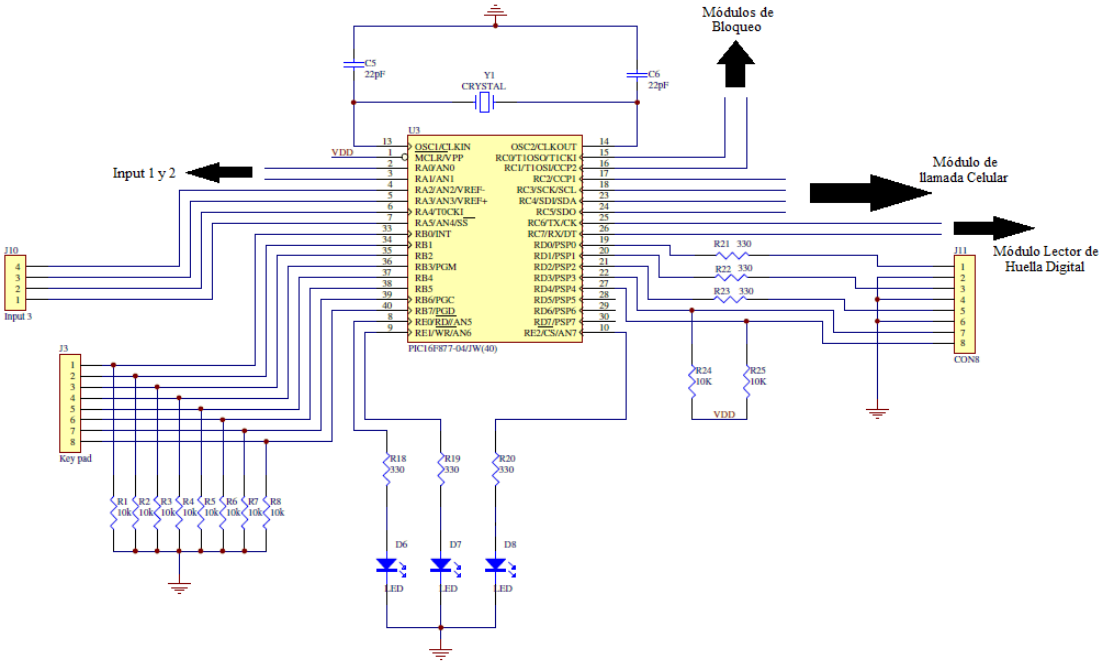


Figura 3.3. Diagrama de conexión del módulo central



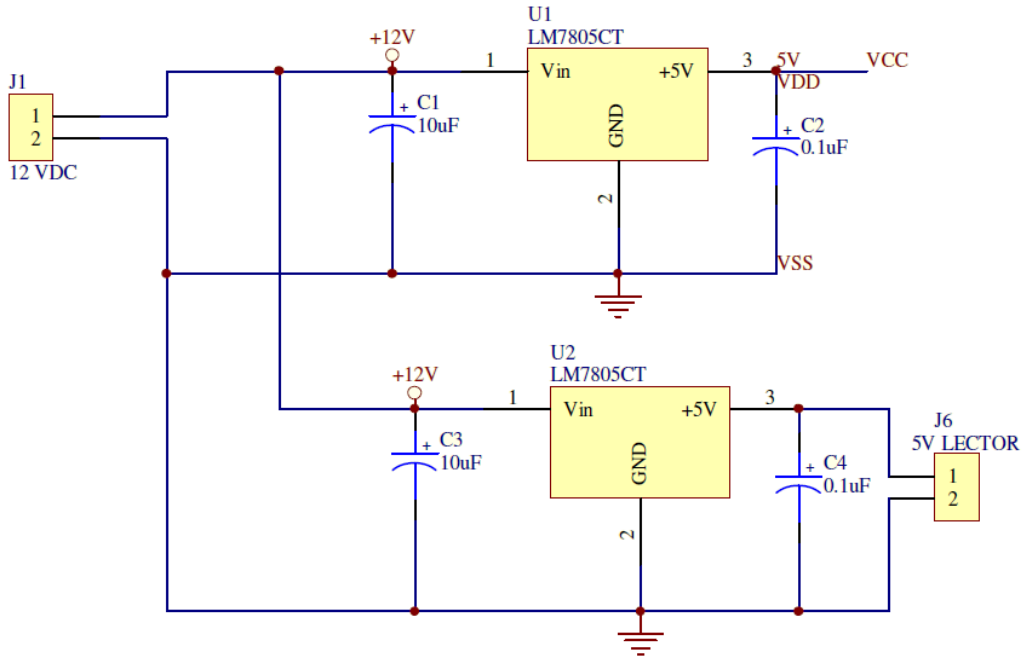
3.2.2. Módulo de alimentación

Es necesario para el funcionamiento de todo circuito poseer una fuente de poder, para este proyecto la alimentación se la toma de la batería de 12 V existente en cualquier vehículo, pero debido a que el sistema trabaja a un potencial de 5 V, es necesario colocar un regulador de voltaje, como se indica en la figura 3.4, adicionalmente permite reducir en parte el ruido que puede ocasionar las variaciones de voltaje producidas por el resto de sistemas del vehículo.

Figura 3.4. Módulo de alimentación

Para el módulo de alimentación se utiliza el circuito integrado LM7805 que es un regulador de voltaje a 5 V, en la figura 3.5 se muestra el uso de este regulador el cual se lo instala con los condensadores conectados entre el ingreso y tierra, y la salida de 5 V y tierra, esto sirve como protección contra sobretensiones transitorias, para el proyecto se alimenta de esta manera todo el sistema incluido el lector de huella digital.

Figura 3.5. Esquema del módulo de alimentación

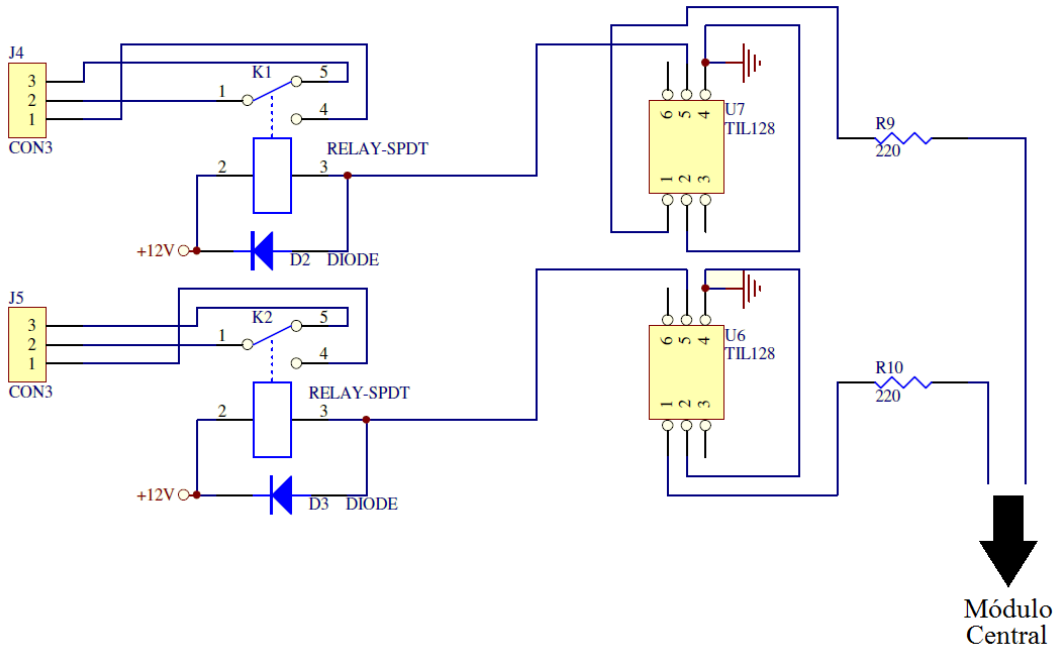


3.2.3. Módulo de mando para bloqueos

En este módulo se va a comandar relés de 12 V uno de ellos bloquea al encendido del vehículo, el segundo se puede utilizar para realizar un segundo bloqueo como por ejemplo a la bomba del combustible, esto puede ser opcional, para este proyecto solo se realiza el bloqueo al encendido.

Para poder controlar a los relés se utiliza circuitos integrados optoacopladores, debido a que la salida del microcontrolador es de 5 V y no es suficiente para realizar la activación, adicionalmente nos ayuda a proteger al sistema de posibles picos de voltajes que puede producir la bobina del relé, esto se puede apreciar en la figura 3.6 para un mejor entendimiento.

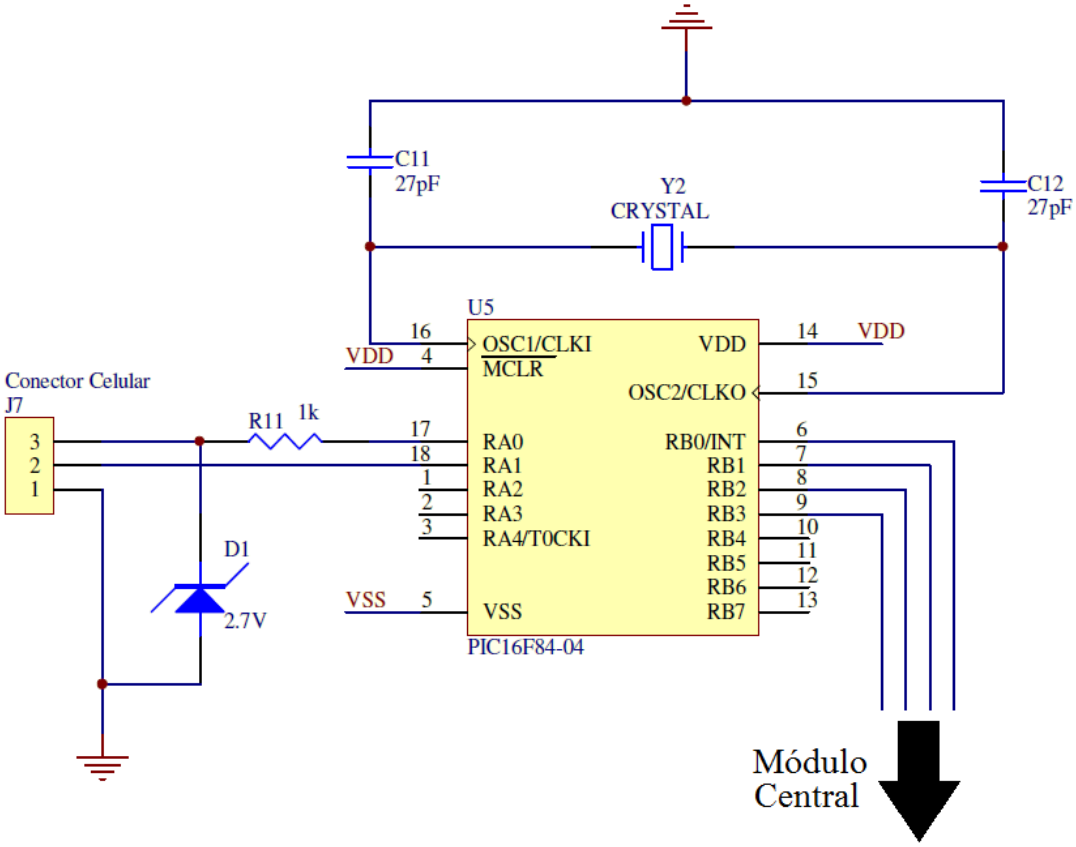
Figura 3.6. Esquema del módulo de mando para bloqueos



3.2.4. Módulo de control celular

Si el vehículo intenta ser encendido por más de 3 intentos se realiza una notificación de seguridad mediante un dispositivo celular, la comunicación con este es mediante un puerto serial, para esto se lo va a comandar mediante el PIC16F84A, el dispositivo celular trabaja a un voltaje de 2,7 V por lo que se conecta el puerto de comunicación en paralelo con un diodo zener y la resistencia de 1 KΩ, como se ilustra en la figura 3.7, adicionalmente tal como en el PIC del módulo central es necesario la utilización del cristal de cuarzo que produce los pulsos para generar la base de tiempo de los códigos del PIC y para finalizar se interconecta con el módulo central.

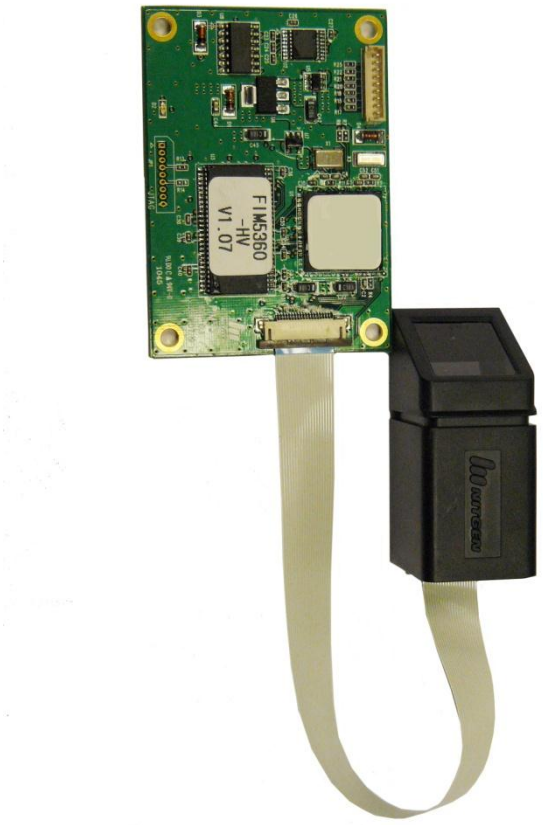
Figura 3.7. Esquema del módulo de control celular



3.2.5. Módulo del lector de huella digital

El lector de huella digital con el que se diseño el sistema es de la marca Nitgen y es el modelo FIM 5360, el mismo ya fue analizado en los capítulos anteriores, en donde se justifica su uso por las buenas prestaciones que posee y la robustez del mismo, se lo puede apreciar en la figura 3.8.

Figura 3.8. Lector de huella digital FIM 5360 de Nitgen. **Fuente:** http://www.nitgen.com



En las siguientes figuras se apreciar el diagrama de bloques del lector de huella digital (figura 3.9) y el diagrama de comunicación que interconecta el lector con el módulo de control central (figura 3.10).

Para esta interconexión se utiliza el circuito integrado MAX232ACPE, que permite una mejor comunicación del puerto RS232 y ayuda a mantener el buen rendimiento del sistema biométrico.

Figura 3.9. Diagrama de bloques del lector de huella digital FIM 5360. **Fuente:** NITGEN CO. LTD, *Datasheet Nitgen Fim5360* Version 1.02, Korea 2011

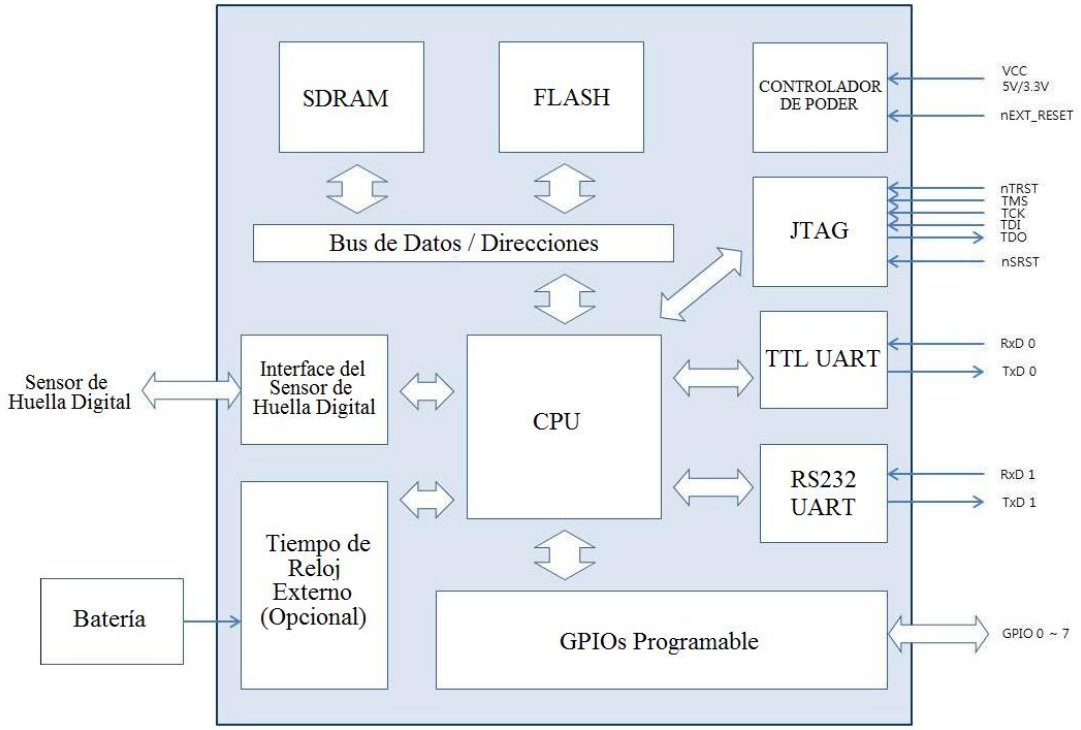
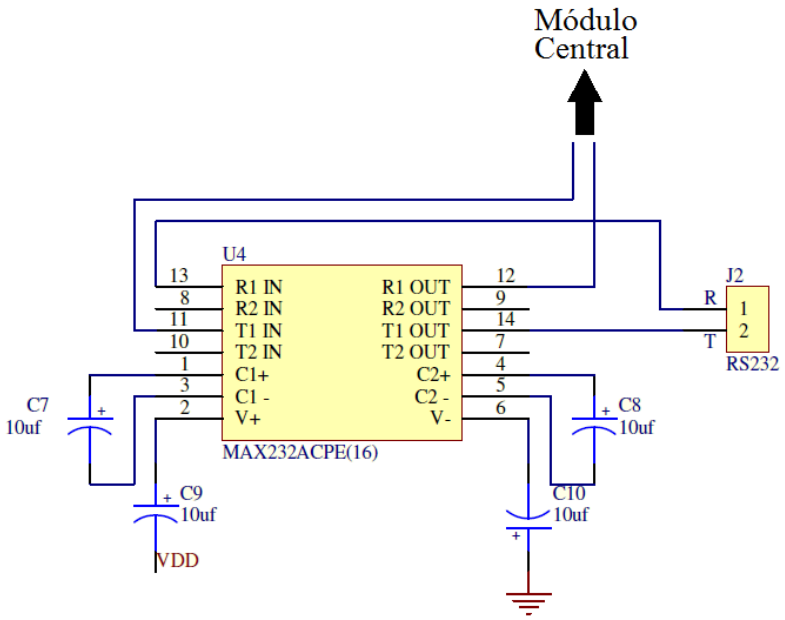


Figura 3.10. Módulo de comunicación del lector FIM 5360 con el módulo central



3.3. Diseño del software

Teniendo en cuenta los alcances determinados para el proyecto, se comienza ya con el desarrollo del software para la programación de los microcontroladores, tanto del PIC principal, así como el PIC del módulo de control celular, mediante el uso del programa MikroBasic, se obtuvo como resultado el conjunto de comandos que se adjunta en los anexos 1 y 2; aunque a continuación se analizara brevemente las subrutinas más importantes.

3.3.1. Programa del PIC principal PIC16F877A

Luego de realizar la programación básica de un microcontrolador como por ejemplo la declaración de variables, subrutinas de inicialización, configuraciones de los puertos de entrada y salida, se procede a analizar las líneas de comando más importantes.

Básicamente se pondrá principal atención en dos subrutinas que son el ingreso de nuevos usuarios y la validación de estos.

3.3.1.A. Subrutina para ingreso de nuevos usuarios

Tal como los microcontroladores que requieren de comandos de inicialización, de igual manera para el lector de huella digital se comienza generando los códigos propietarios que arrancan al dispositivo biométrico configurando al puerto de comunicación, esto se puede apreciar en las primeras líneas de esta subrutina.

Luego de la inicialización se habilita al lector biométrico con la opción de ingreso de usuarios, en esta configuración el dispositivo realizará una doble lectura de la huella digital, la primera vez codifica lo leído en un algoritmo numérico y en la segunda lectura realiza un comparativo con la muestra anterior, en el caso de que este sea positivo se guarda en la memoria del dispositivo, al no ser así se rechaza la lectura inicial reiniciándose y es necesario comenzar el proceso nuevamente.

```
396: '*****************************************************************
397: 'proceso para nuevo usuario
398: '*****************************************************************
399: case 6 '------------------------------
400: 'Emvia el comando 0x01 CMD_REQUEST_CONNECTION
401: Comando= $01
402: Param1= $00
403: Param2= $00
404: Heder_Checksum= $01
405: Escritura()
406: Swt=7
407: case 7 '------------------------------
408: Long_Dat=24
409: Swt_Lectura=8
410: Lectura()
411: case 8 '------------------------------
412: if Matriz[3]= $01 then
413: 'Envia el comando 0x2F CMD_ENTER_MASTER_MODE2 master null=3
414: Comando= $2F
415: Param1= $03
```

```
416: Param2= $00
417: Heder_Checksum= $32
418: Escritura()
419: Swt=9
420: else
421: Swt=0
422: end if
423: case 9 '------------------------------
424: Long_Dat=24
425: Swt_Lectura=10
426: Lectura()
427: case 10 '------------------------------
428: if Matriz[7]= $01 then
429: 'Envia comando 0x38 CMD_ENROLL_FP 1
430: Comando= $38
431: Param1= $00
432: Param2= $01
433: Heder_Checksum= $39
434: Escritura()
435: Swt=11
436: else
437: Swt=0
438: end if
439: case 11 '------------------------------
440: Long_Dat=24
441: Swt_Lectura=12
442: Lectura()
443: case 12 '------------------------------
```

```
444: if Matriz[7]= $01 then
445: 'Envia comando 0x68 CMD_GET_IMAGE_QAULITY
446: Comando= $68
447: Param1= $00
448: Param2= $00
449: Heder_Checksum= $68
450: Escritura()
451: Swt=13
452: else
453: Swt=0
454: end if
455: case 13 '------------------------------
456: Long_Dat=24
457: Swt_Lectura=14
458: Lectura()
459: case 14 '------------------------------
460: Delay_ms(2000)
461: if Matriz[7]= $01 then
462: 'Envia comando 0x38 CMD_ENROLL_FP 2
463: Comando= $38
464: Param1= $00
465: Param2= $02
466: Heder_Checksum= $3A
467: Escritura()
468: Swt=15
469: else
470: Swt=0
471: end if
```

```
472: case 15 '------------------------------
473: Long_Dat=24
474: Swt_Lectura=16
475: Lectura()
476: case 16 '------------------------------
477: if Matriz[7]= $01 then
478: 'Envia comando 0x68 CMD_GET_IMAGE_QAULITY
479: Comando= $68
480: Param1= $00
481: Param2= $00
482: Heder_Checksum= $68
483: Escritura()
484: Swt=17
485: else
486: Swt=0
487: end if
488: case 17 '------------------------------
489: Long_Dat=24
490: Swt_Lectura=18
491: Lectura()
492: case 18 '------------------------------
493: if Matriz[7]= $01 then
494: 'Envia comando 0x38 CMD_ENROLL_FP Final guarda usuario nuevo
495: Comando= $38
496: Param1= $00
497: Param2= $04
498: Heder_Checksum= $3C
499: Escritura()
```

```
500: Swt=19
501: else
502: Swt=0
503: end if
504: case 19 '------------------------------
505: Long_Dat=24
506: Swt_Lectura=20
507: Lectura()
508: case 20 '------------------------------ termina nuevo usuario
509: if Matriz[7]= $01 then 'Correcto nuevo usuario Ingresado
510: i=0
511: for i = 0 to 8
512: SetBit(PORTE,1)
513: Delay_ms(300)
514: ClearBit(PORTE,1)
515: Delay_ms(300)
516: next i
517: end if
518: Swt=0
```

3.3.1.B. Subrutina de identificación de usuarios

El comienzo de esta subrutina es igual que en el caso anterior, la diferencia se presenta cuando se habilita al lector para que trabaje en modo de identificación de usuarios, en esta configuración las líneas de comandos son más reducidas debido a que el dispositivo lee una sola vez y realiza el comparativo 1 a N con la base de datos generada con los usuarios ingresados, esta subrutina se adjunta a continuación.

```
'*********************************************************************
'proceso para identificacion Usuario
'*********************************************************************
case 1
Inc(Inten_Errados)
'------------------------------
'Envia el comando 0x01 CMD_REQUEST_CONNECTION
Comando= $01
Param1= $00
Param2= $00
Heder_Checksum= $01
Escritura()
Swt=2
case 2 '------------------------------
'lectura
Long_Dat=24
Swt_Lectura=3
Lectura()
case 3 '------------------------------
```

```
366: if Matriz[3]= $01 then
367: 'Envia el comando 0x12 CMD_IDENTIFY_FP
368: Comando= $12
369: Param1= $00
370: Param2= $00
371: Heder_Checksum= $12
372: Escritura()
373: Swt=4
374: else
375: Swt=0
376: end if
377:
378: case 4 '-------------------------------
379: 'lectura
380: Long_Dat=39
381: Swt_Lectura=5
382: Lectura()
383: case 5 '-------------------------------
384: if Matriz[7]= $01 then 'correcto Usuario identificada
385: SetBit(PORTC,1) 'Rele Encendido
386: Inten_Errados=0
387: i=0
388: for i = 0 to 8
389: SetBit(PORTE,0)
390: Delay_ms(300)
391: ClearBit(PORTE,0)
392: Delay_ms(300)
393: next i
```

```
394: end if
395: Swt=0
```

3.3.2. Programa del PIC del módulo de control celular PIC16F84A

El programa que se ha cargado en este microcontrolador, permite realizar la intercomunicación entre el módulo central con el dispositivo celular, por lo que en las líneas de comando se encontrará la inicialización del PIC, la configuración de los puertos y la rutina de comunicación celular, este software se encuentra en el anexo 2.

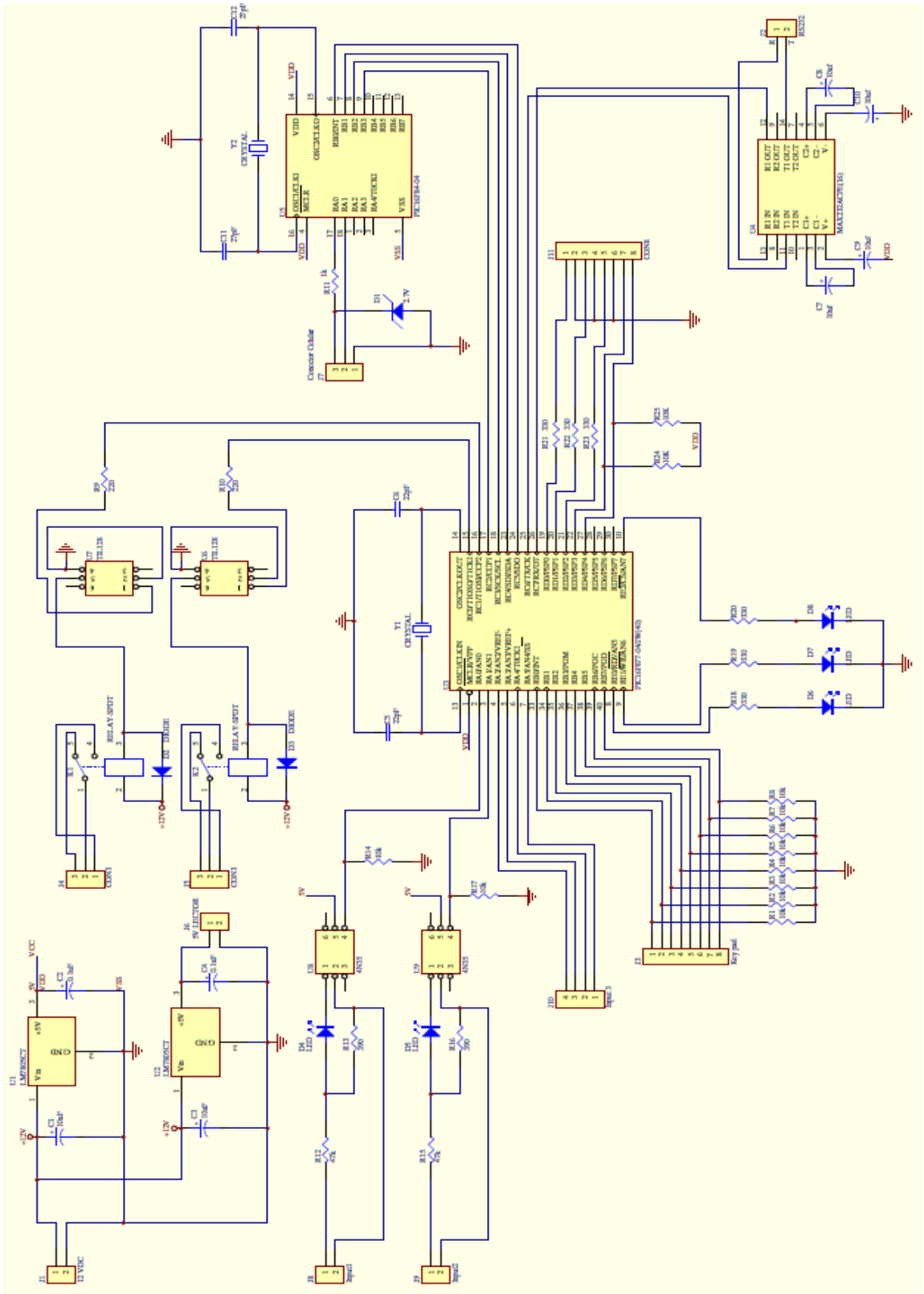
La parte más importante para la programación del sistema es el conocer los comandos de programación e intercomunicación del lector de huella digital, por lo que se adjunta a este proyecto en el anexo 3.

3.4. Diseño de la placa del sistema de seguridad biométrico

Una vez realizadas las pruebas y comprobado el correcto funcionamiento del sistema biométrico, se continua con la elaboración de la placa del hardware integrando todos los módulos, para lograr un diseño más compacto, ya que por seguridad esté sistema debe ser instalado en lugares pequeños y que no esté a la vista de los delincuentes para que sea más difícil su vulnerabilidad.

Se parte por realizar la interconexión de todos los componentes electrónicos en un mismo diseño de circuitos, como se muestra en la figura 3.11, o en Figura 3.11.

Figura 3.11. Esquema de diseño con todos los componentes integrados



Como resultado de este diseño se obtiene las diagramaciones de las pistas superiores e inferiores, así como también las disposiciones de los componentes en la placa, estas se muestran en las figuras 3.12 a 3.14 y se detalla de la misma manera la lista de materiales que se utilizaron para la construcción del proyecto.

Figura 3.12. Imagen superior del diseño de la placa

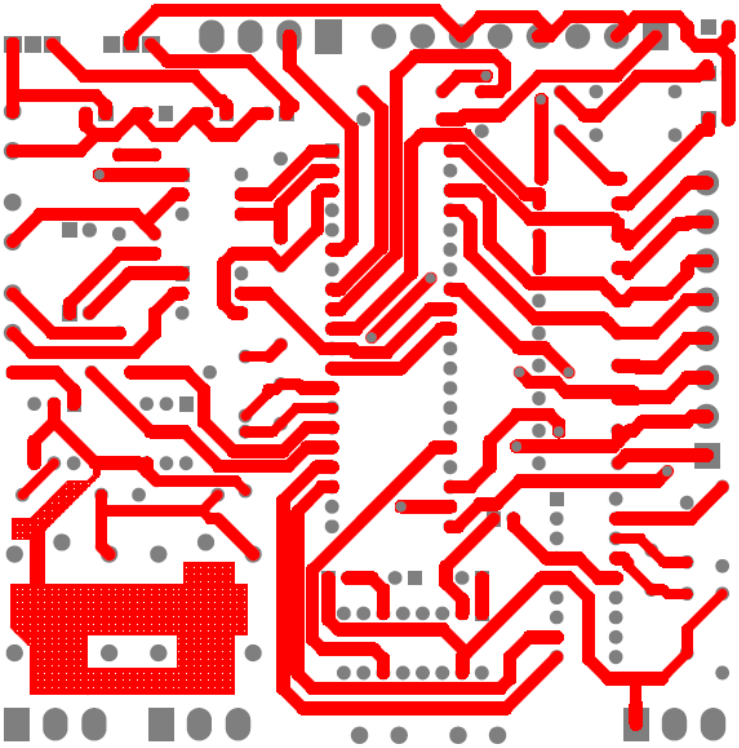


Figura 3.13. Imagen Inferior del diseño de la placa

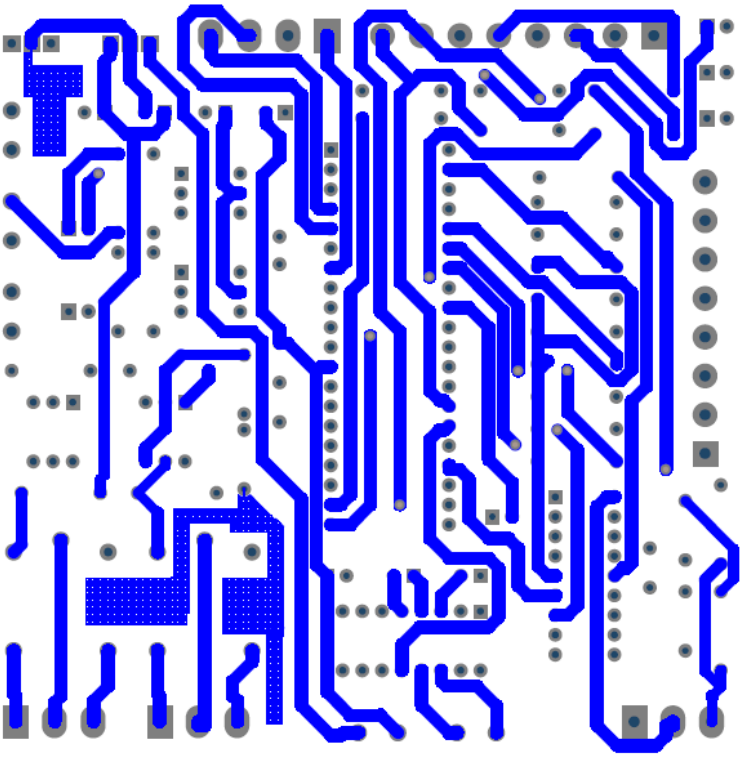
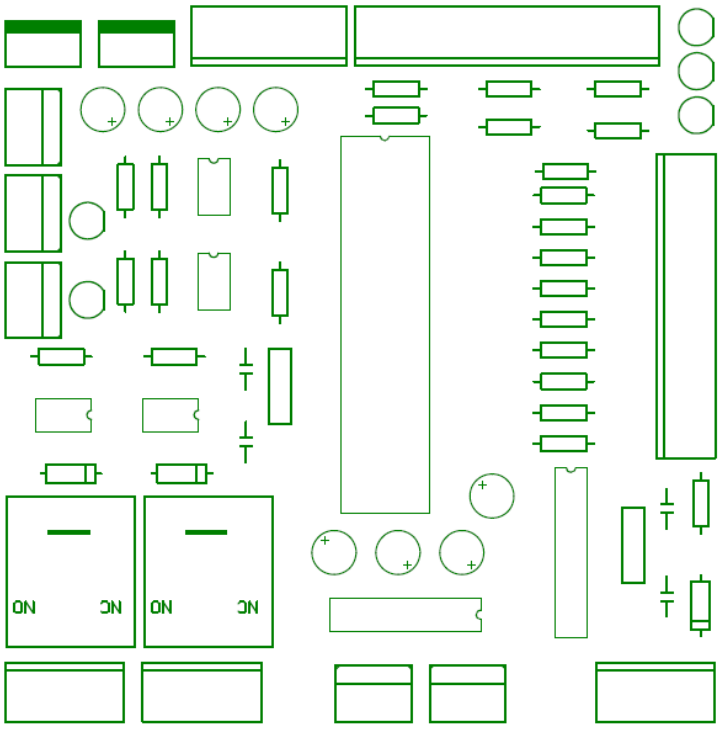


Figura 3.14. Imagen de la ubicación de los componentes en la placa



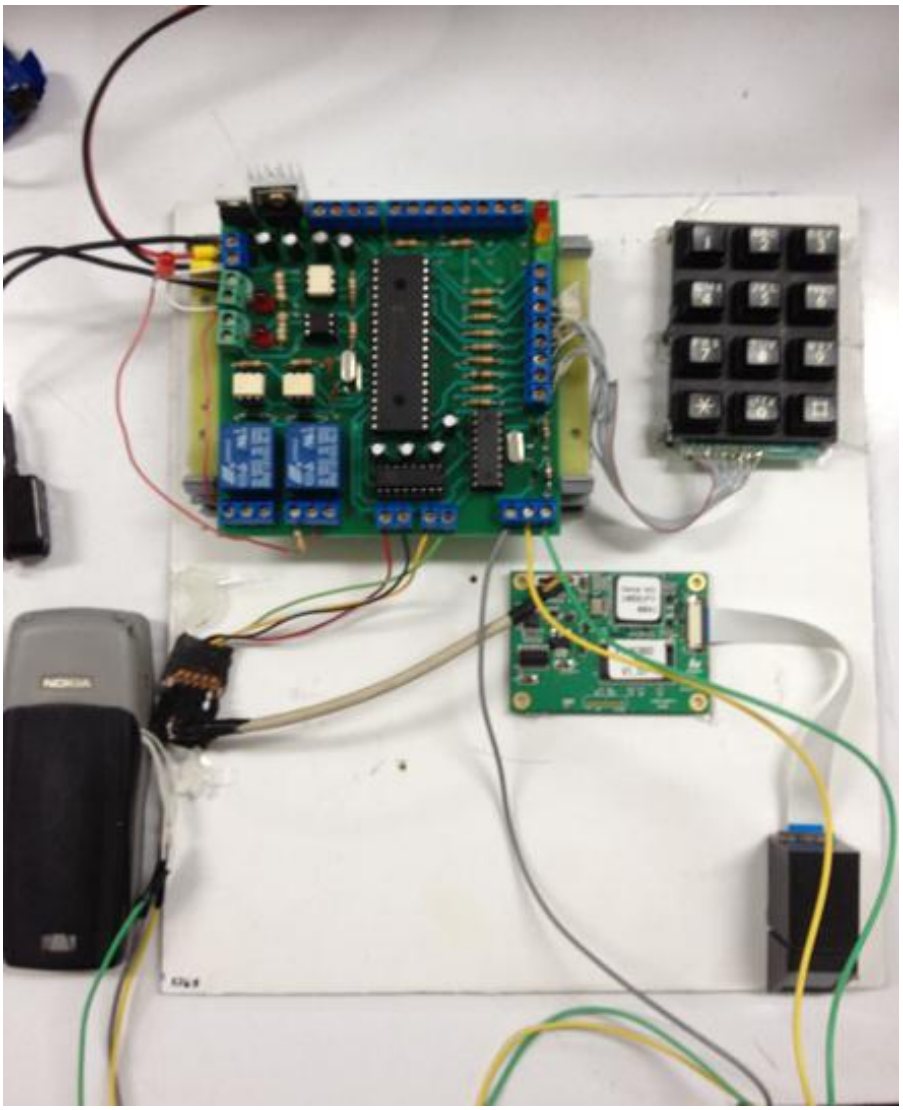
A continuación en la tabla 3.1 se detalla la lista de componentes que se utilizaron en el diseño del hardware:

Tabla 3.1. Lista de materiales utilizados para la construcción del sistema biométrico

ITEM	DESCRIPCION	CANTIDAD
Capacitor	0.1uF	2
Capacitor	10uF	6
Capacitor	22pf	2
Capacitor	27pf	2
Resistecia	1k	1
Resistecia	10k	12
Resistecia	47k	2
Resistecia	220	2
Resistecia	330	6
Resistecia	390	2
Diodo zener	2.7 voltios	1
Diodo	1A	2
Led	normal	5
Optoacoplador	4N35	4
BORNERA	2 PINES 5 MM AZUL	14
BORNERA	3 PINES 5 MM AZUL	3
CRYSTAL	16 Mhz	1
CRYSTAL	4 Mhz	1
REGULADOR	LM7805	2
Circuito Integrado	Max 232	1
Circuito Integrado	PIC16f84A	1
Circuito Integrado	PIC16f877	1
Socalo	40 pines	1
Socalo	18 pines	1
Socalo	16 pines	1
Socalo	6 pines	4
RELE NORMAL	5 PINES BOBINA 12Vdc	2

Una vez realizada la placa definitiva se procedió armarla con los componentes electrónicos detallados anteriormente, se realizaron las interconexiones de los dispositivos que conforman el proyecto y las pruebas pertinentes antes de la instalación en el vehículo, como se observa en la fotografía 3.15.

Figura 3.15. Integración de los componentes definitivos del sistema de seguridad biométrico



3.5. Conclusiones

Luego de haber logrado una correcta integración de todos los módulos del proyecto, realizar las pruebas de funcionamiento del mismo en donde se obtuvo muy buenos resultados y cumpliendo con los alcances y expectativas planteadas para el sistema de seguridad biométrico; se determina que se ha podido lograr un correcto desempeño acatando con todos los parámetros propuestos, por lo que satisfactoriamente se procedió a armar el primer prototipo denominado SSATBS-01 (Sanmartín - Serrano Anti Theft Biometric Sistem), que va a ser instalado en diferentes vehículos donde se determinará el desempeño del sistema en un ambiente real de trabajo.

CAPÍTULO IV

VALIDACIÓN DEL PROTOTIPO

4.1. Introducción

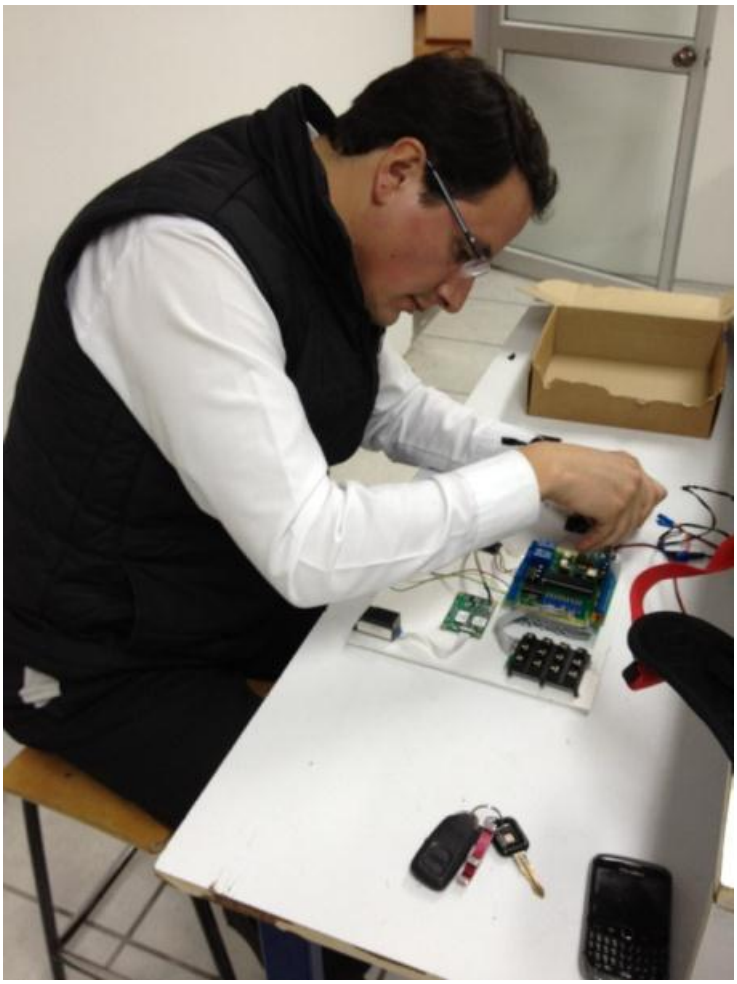
Para la validación del proyecto es importante comprobar el funcionamiento del prototipo en condiciones reales poniéndolo a prueba en vehículos con uso diario y es en donde se podrá determinar parámetros como rendimiento, seguridad del sistema, facilidad de adaptación de los usuarios, metodología de instalación en el vehículo y comprobar si el diseño es idóneo para trabajar en un ambiente que estará expuesto a variables como temperatura, humedad, ruido externo, etc. y de esta manera poder garantizar el correcto funcionamiento de la alarma biométrica SSATBS-01 (Sanmartín - Serrano Anti Theft Biometric Sistem), figura 4.1.

Figura 4.1. Prototipo SSATBS-01

4.2. Pruebas de funcionamiento en laboratorio

Han sido necesarias varias pruebas de funcionamiento en el laboratorio para ir corrigiendo los errores que se presentaron a lo largo del diseño, como agregar funciones que son necesarias para que el usuario tenga mayor seguridad y facilidad en la utilización del sistema, es importante comprender la universalización del proyecto pues tiene que ser utilizado por todo tipo de usuarios, por lo que es necesario el fácil manejo del sistema, en la figura 4.2 se están realizando estas pruebas.

Figura 4.2. Imagen de la pruebas realizadas en el laboratorio



Del resultado de estas pruebas se realizó algunos cambios como mejorar el método de ingreso de nuevos usuarios, así como también se vio la necesidad de agregar avisos luminosos que ayuden a entender al usuario cuando el sistema este activo y cuando se tiene que ingresar la huella digital para desactivar el anti atraco y poder poner en marcha al vehículo.

Adicionalmente se vio la necesidad de crear una clave de valet parking que se ingrese mediante teclado para casos en que el usuario lleve su vehículo a mantenimiento o tenga que ser manipulado por un usuario al cual no se lo va a registrar en la base de datos de huellas digitales, pues su utilización es temporal, de igual manera se presento la necesidad de incorporar una clave maestra que será utilizada para borrar todos los usuarios en caso de que el vehículo cambie de dueño y de esta manera poder ingresar las huellas de los nuevos usuarios.

4.3. Determinación del rendimiento y seguridad del sistema

Una vez que se han realizado rigurosas pruebas de funcionamiento se ha podido determinar que el sistema biométrico presenta un excelente rendimiento y con un nivel de seguridad muy alto, para llegar a esta determinación se analizaron los siguientes parámetros.

4.3.1. Rendimiento

Para determinar el rendimiento del sistema biométrico se partió instalando el prototipo en diferentes vehículos y se lo hizo trabajar durante varios días en donde se pudo comprobar la robustez del proyecto al trabajar durante varios días sin generar inconvenientes, de igual manera se probó todas las aplicaciones planteadas en el proyecto, como ingreso de nuevos usuarios, validación de los mismos, eliminación de usuarios, configuración en modo valet parking, en donde se obtuvo los mejores resultados.

4.3.2. Seguridad

Como parte de las pruebas realizadas se verificó si se cumple efectivamente las seguridades propuestas y se pudo comprobar la eficacia del prototipo, evitando el encendido del vehículo al no validar una huella dactilar errónea, realizó el aviso celular al obtener tres ingresos erróneos, de esta manera se comprobó la alta eficiencia del prototipo.

4.4. Uso del sistema de alarma en un vehículo

Es importante definir el modo de uso y la forma de programar el sistema de seguridad biométrico, con esto se demuestra la facilidad para el usuario de aprender a utilizar el sistema y la confiabilidad que brinda.

4.4.1. Usuario máster

El sistema tiene por defecto programada la clave "1234" que será la clave de usuario principal que permite acceder a funciones como ingreso de usuarios encendido del vehículo mediante teclado, y borrado de usuarios, dicho código podrá ser cambiado por el nuevo usuario. Para cambiar la clave maestra se tiene que ingresar "1234*" y a continuación digitar la nueva clave de 4 dígitos, el LED verde se encenderá para indicar que la nueva clave ha sido aceptada, ejemplo: "1234*" "5555".

4.4.2. Encendido mediante teclado

Esta función fue necesaria ya que se presenta ocasiones en las que el usuario tiene que dejar su vehículo para lavado o mantenimiento y el vehículo va a ser operado temporalmente por un usuario al que no se necesita registrar en la base de datos de huellas digitales, entonces se dispone de un teclado en el que se ingresa un código definido por el usuario en el cual desbloquea el sistema y permite el arranque, esta función también podrá ser usada en casos en las que las huellas digitales no puedan ser leídas ya sea por daño del lector o por deterioro de la huella programada. Para ingresar esta clave se tiene que ingresar la clave maestra seguida de asterisco y a

continuación la clave de encendido, el LED verde se encenderá para indicarnos que ha sido ingresado el código, ejemplo: “1234*” “4444”.

4.4.3. Borrado de usuarios:

Es necesario también ingresar una clave con la cual se borre todos los usuarios programados ya que en el caso de que el vehículo cambie de dueño, el usuario actual tendrá que borrar su base de datos para que el nuevo usuario ingrese su información, para programar la clave de borrado se tiene que ingresar la clave maestra seguida de asterisco y a continuación la clave de borrado el LED verde se encenderá para indicarnos que la clave ha sido aceptada, al digitar dicha clave toda la base de datos de huellas será borrada por lo que tiene que ser usada solo en casos determinados en los que se desee ingresar usuarios totalmente diferentes, ejemplo: “1234*” “5678”.

4.4.4. Ingreso de usuarios:

Va a ser siempre necesario ingresar usuarios que vayan a utilizar el vehículo con un mínimo de dos huellas por cada usuario, esto es para casos en los que la huella que se usa comúnmente este inutilizable o el lector no la haya detectado como válida, además siempre existirá la necesidad del ingreso de usuarios por lo que contar con una clave para ingresar huellas será fundamental, de la misma manera que se ha hecho el ingreso de claves se programará una clave que nos permita ingresar nuevas huellas a la base de datos.

Para ingresar esta clave se tecleará la clave maestra seguida de asterisco y a continuación la nueva clave, el led verde se enciende para indicarnos que ha sido programada, ejemplo: “1234*” “7777”. Cuando se ingresar una nueva huella el proceso que se realiza es el que se detalla a continuación:

Se ingresa la clave maestra “7777*" se enciende el lector de huella digital lo que indica que se debe colocar la huella digital sin presionar con fuerza, luego de colocada la huella digital el lector se apaga durante un segundo en el que se tiene que retirar el dedo y a continuación se enciende nuevamente, este proceso lo realiza para validar la huella antes puesta y si es correcta se encenderá el led verde lo que indica al usuario que una nueva huella ha sido ingresada a la base de datos. En el sistema se puede almacenar hasta 100 huellas digitales de alta definición, lo que hace poco probable que se dé el error en el reconocimiento de huellas.

Cada vez que sea necesario usar las claves ya sea para ingreso de huellas, desbloqueo del encendido o borrado de usuarios será necesario usar * al final. ejemplo: “7777*”.

Se puede apreciar a continuación en la figura 4.3 y 4.4 el panel frontal del prototipo, en donde se ilustra la ubicación física del lector de huella digital, el puerto del teclado y los leds indicadores.

Figura 4.3. Diagrama del panel frontal del prototipo SSATBS-01

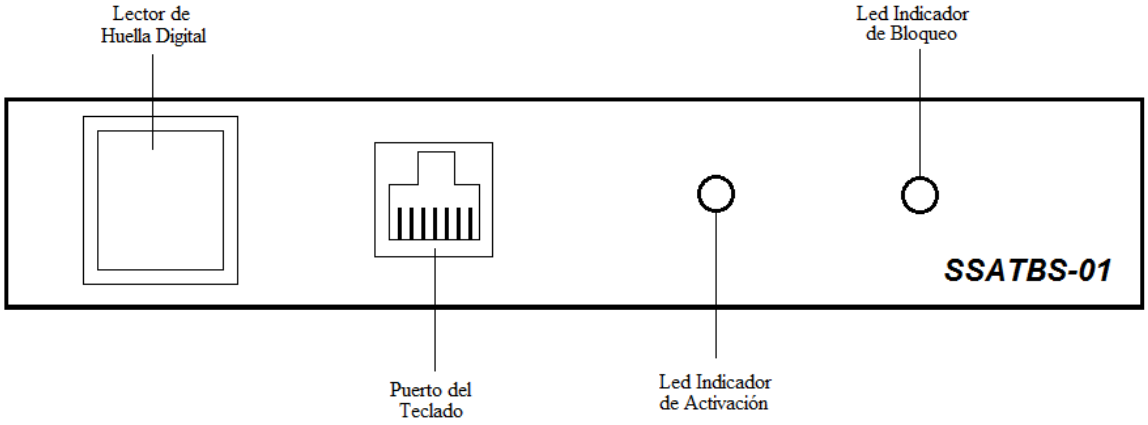


Figura 4.4. Panel frontal del prototipo SSATBS-01

4.5. Diagrama de instalación

Para realizar la instalación del prototipo en el vehículo se dispone de un puerto en la parte posterior con un conector de 8 pines donde se tiene la alimentación, el puerto para el bloqueo en el encendido y se encuentra también los pines para el módulo de comunicación celular, esto se ve en la figura 4.5 y la 4.6.

Figura 4.5. Diagrama del panel posterior del prototipo SSATBS-01.

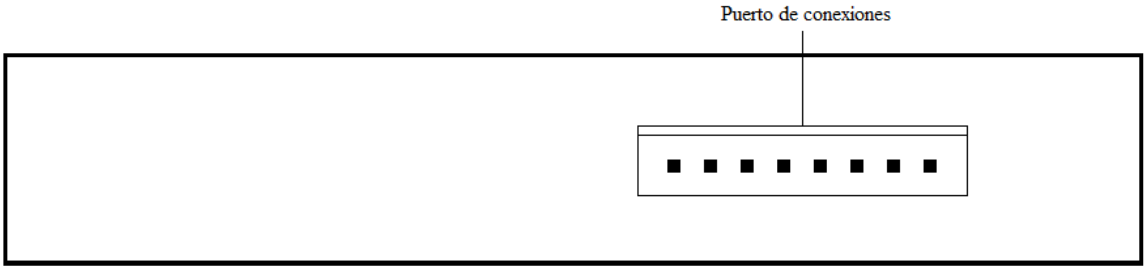
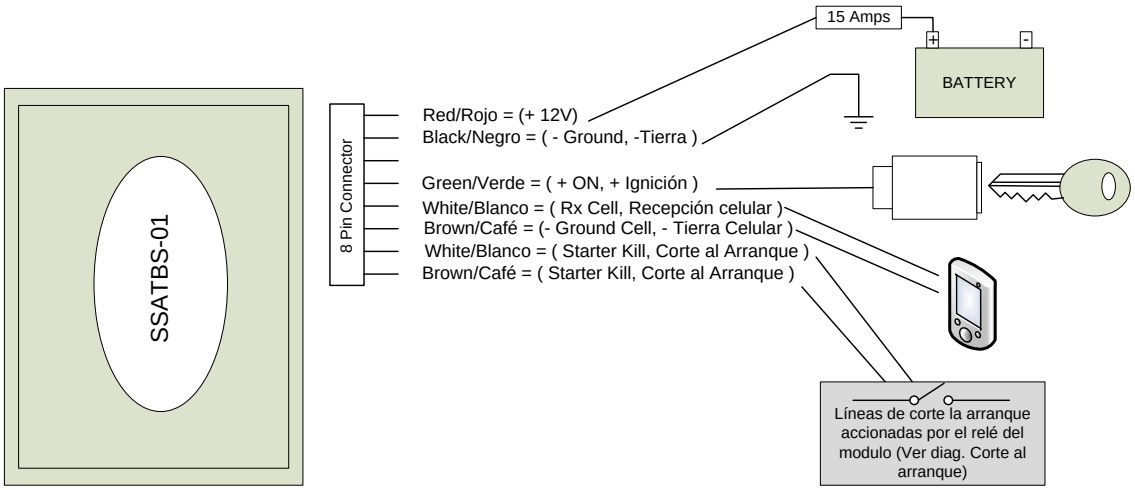


Figura 4.6. Panel posterior del prototipo SSATBS-01.

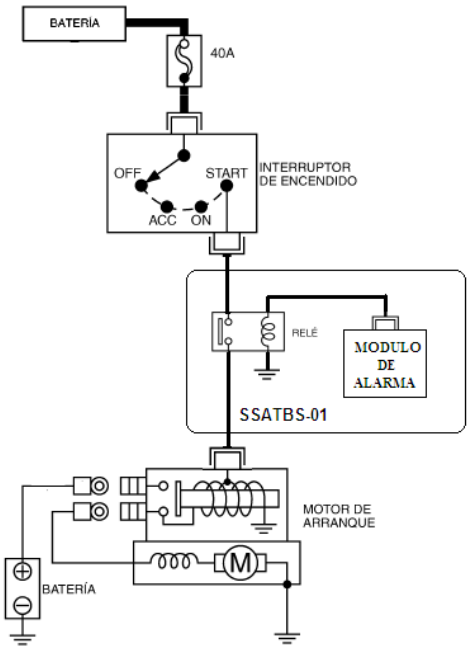
A continuación en la figura 4.7 se detalla las funciones de cada uno de los pines del conector existente en la parte posterior.

Figura 4.7. Detalle de pines del conector posterior.



Un diagrama importante de instalación es el bloqueo al encendido por lo que se muestra este esquema en la figura 4.8, esto permitirá realizar la instalación con una mayor facilidad.

Figura 4.8. Esquema para el bloqueo al encendido. **Fuente:** NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010



4.6. Pruebas prácticas de funcionamiento en vehículos

Se realizó la instalación del sistema biométrico en varios vehículos de diferentes marcas y modelos, esto con la finalidad de poder comprobar de mejor manera el correcto desempeño del prototipo en ambientes reales de funcionamiento y a la vez garantizar la generalidad en la instalación sin tener limitaciones por las diferencias que existen entre vehículos, comprobando todos los parámetros propuestos en el proyecto, se pudo instalar en tres vehículos diferentes, el primero fue de marca Trooper, modelo 1999 (Figuras de la 4.9 a 4.11), el segundo fue un automóvil Chevrolet, modelo 2008 (Figuras 4.12 y 4.13) y el tercer vehículo fue un Suzuki Forsa I, modelo 1987 (Figuras 4.14 y 4.15).

Figura 4.9. Instalación y pruebas en vehículo Trooper, año 1999, primera fotografía.

Figuraa 4.10. Instalación y pruebas en vehículo Trooper, año 1999, segunda fotografía.

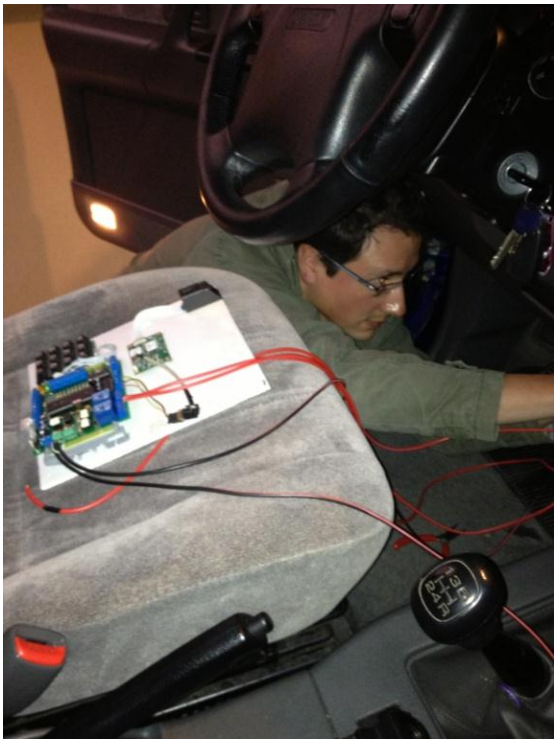


Figura 4.11. Instalación y pruebas en vehículo Trooper, año 1999, tercera fotografía.

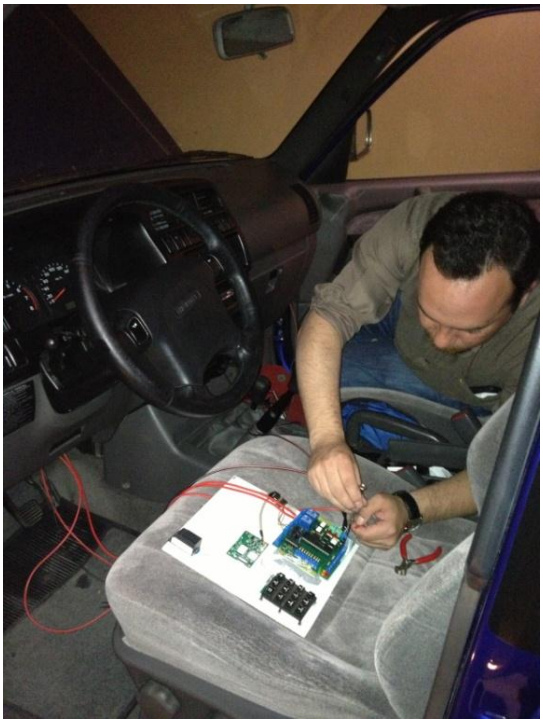


Figura 4.12. Instalación y pruebas en vehículo Chevrolet Aveo Activo, año 2008, primera fotografía.

Figura 4.13. Instalación y pruebas en vehículo Chevrolet Aveo Activo, año 2008, segunda fotografía.

Figura 4.14. Instalación y pruebas en vehículo Suzuki Forsa I, modelo 1987, primera fotografía.

Figura 4.15. Instalación y pruebas en vehículo Suzuki Forsa I, modelo 1987, segunda fotografía.

En el primer y tercer vehículo las pruebas se realizaron por varios días para de esta manera validar el prototipo SSATBS-01 con funcionamiento del día a día y poder verificar si presenta algún inconveniente en el funcionamiento o si existe alguna

observación que se pueda considerar para realizar mejoras y por ende brindar al cliente un producto de muy alta calidad y seguridad garantizada.

Se consideró la instalación en el vehículo Suzuki Forsa ya que a mas de verificar una vez más el funcionamiento del sistema biométrico, el dueño del mismo es una persona ajena al proyecto y puede brindar una opinión imparcial del desempeño del prototipo SSATBS-01, así como también recomendaciones que permitan mejorar el proyecto para posteriores diseños.

Una vez que se ha utilizado por varios días el sistema de seguridad, se toma el testimonio y las recomendaciones los cuales se adjuntan a continuación:

El sistema de seguridad mediante la huella dactilar me parece un sistema muy confiable y seguro para proteger un automóvil.

El sistema es muy amigable y fácil de utilizar solo se necesita la huella dactilar para desbloquear al automóvil y listo. Una ventaja importante del sistema es que un usuario puede ingresar varias huellas y así evitar depender de una solo huella para activar el automóvil esto permite que el sistema sea robusto.

El equipo es compacto por tal motivo fácil de instalar y no requiere de mucho espacio para la instalación se lo puede ubicar en cualquier lugar de fácil acceso para el usuario.

Para ingresar un nuevo usuario el sistema solo requiere de una clave y el reconocimiento de la huella dactilar este proceso no dura más de 3 minutos

Otro punto a destacar es la alerta que emite al celular de un usuario, al tener más de 5 intentos fallidos esto permite estar siempre en contacto con el automóvil.

Recomendaciones.

Interconectar la alarma del automóvil y el sistema de huella dactilar, con esto se podría monitorear también el estado completo del automóvil en cuanto a seguridad.

4.7. Conclusiones

Una vez que se ha podido verificar el correcto funcionamiento del prototipo SSATBS-01 con las pruebas en los vehículos, se concluye que se a logrado cumplir con los parámetros planteados, se pudo determinar también que el sistema es robusto y brinda una alta seguridad que es el objetivo principal del diseño del proyecto.

Otro aspecto muy importante a considerar con las pruebas realizadas, es que este sistema de seguridad puede ser instalado en vehículos de cualquier marca, modelo y año de fabricación, lo que vuelve a este sistema muy versátil y sin limitaciones de uso para clientes que lo deseen adquirir.

De esta manera el prototipo SSATBS-01 proporciona una muy buena opción para implementar una industria productora de estos sistemas de seguridad.

CONCLUSIONES Y RECOMENDACIONES

Según los objetivos planteados para el desarrollo de este proyecto se ha determinado que este sistema proporciona un alto nivel de seguridad para los vehículos ya que los usuarios lo pueden instalar como una seguridad adicional a la existente, independientemente del tipo de automotor que sea, siendo esto una de las características más importantes que hacen de este prototipo muy versátil, confiable y sobre todo seguro.

Se lo pudo probar en diferentes vehículos de distintas marcas y modelos, obteniendo en todos el mismo desempeño.

Es un proyecto económicamente factible para producirlo en gran medida generando una industria nacional sustentable, brindando costos bajos para los usuarios que deseen mejorar las seguridades para sus vehículos.

BIBLIOGRAFÍA

CONSEJO DE SEGURIDAD CIUDADANA, *Boletín estadístico del Consejo de Seguridad Ciudadana de Cuenca*/ Publicación 2013

GOMEZ VIEITES, ALVARO. *Enciclopedia de la seguridad informática*/ Alfaomega. México. 2. ed. 2011.

BIOMETRIA BASICA, *Manual de Aplicación de Tecnologías Biométricas,* Estados Unidos 2008

REYES, CARLOS. *Aprenda rápidamente a programar microcontroladores*/ AYERVE. Quito. 2004.

TOKHEIM, ROGER L.; SANCHEZ, JUAN MANUEL; VAQUERO, ANTONIO. *Fundamentos de los microprocesadores* / MacGraw Hill. México. 2002.

MICROCHIP, *Datasheet 16F87XA*/ Microchip Technology Inc. Dallas. 2010

MICROCHIP, *Datasheet 16F84A*/ Microchip Technology Inc. New York. 2009

NISSAN MOTOR CO. LTD, Manuales *Eléctricos de Servicio*, Japón 2010

NITGEN CO. LTD, *Datasheet Nitgen Fim5360* Version 1.02, Korea 2011

Páginas web

Figura 1.1. Puntos de minucia de huellas digitales [en línea]. México, D.F.: Biometría aplicada, [fecha de consulta: 15 de octubre del 2012].

Disponible en: http://www.bixit.mx/2011/identificacion-biometrica

Estándares de tecnologías biométricas [en línea]. Estados Unidos: The National Institute of Standards and Technology (NIST), [fecha de consulta: 7 de abril del 2013].

Disponible en: http://www.nist.gov/itl/csd/biometrics/bioapicts.cfm

Figura 1.3. Captura de huellas digitales [en línea]. Ecuador: Square Net software solutions, [fecha de consulta: 9 de diciembre 2012].

Disponible en : http://www.squarenet.com.ec/conocimientocomofunciona.html

Figura 1.4. Extracción o digitalización [en línea]. Colombia: Acceso VIP, Biometría y Tecnología, [fecha de consulta: 10 de diciembre 2012].

Disponible en: http://www.accesovip.co/biometria.html

Información del PIC [en línea]. Shanghai: Microchip Technology Inc., [fecha de consulta: 16 de noviembre 2012].

Disponible en:

https://www.microchip.com/pagehandler/en-us/products/picmicrocontrollers

Figura 3.8. Lector de huella digital FIM 5360 de Nitgen [en línea]. Korea: Nitgen, [fecha de consulta: 21 de mayo del 2013].

Disponible en: http://www.nitgen.com

Anexo I

Programa del PIC principal PIC16F877A

```
1: program Tesis_S
2: '-----------------------------------------------------------------
3: 'Declaracion de variables
4: dim i,Comando,Param1,Param2,Heder_Checksum, Inicio,n,Swt as byte
5: dim Swt_Lectura, Long_Dat,kp,n_pulso,oldstate,Inten_Errados as byte
6: dim Matriz as byte[39]
7: dim teclado as word
8: dim ctr_sup as longword
9:
10: '-----------------------------------------------------------------
11:
12: '-----------------------------------------------------------------
13: sub procedure interrupt
14:
15:
16:
17: TMR0 = 0 ' Clear TMRO
18: INTCON = $20 ' Clear TMR0IF and set TMR0IE
19: end sub
20: '-----------------------------------------------------------------
21:
22: '-----------------------------------------------------------------
23: sub procedure Escritura()
```

```
24: USART_Write($7E)
25: USART_Write($00)
26: USART_Write($00)
27: USART_Write($00)
28: USART_Write(Comando)
29: USART_Write($00)
30: USART_Write($00)
31: USART_Write($00)
32: USART_Write(Param1)
33: USART_Write($00)
34: USART_Write($00)
35: USART_Write($00)
36: USART_Write(Param2)
37: USART_Write($00)
38: USART_Write($00)
39: USART_Write($00)
40: USART_Write($00)
41: USART_Write($00)
42: USART_Write($00)
43: USART_Write($00)
44: USART_Write($00)
45: USART_Write($00)
46: USART_Write($00)
47: USART_Write($00)
48: USART_Write(Heder_Checksum)
49: end sub
50: '----------------------------------------------------------------
51: sub procedure Lectura()
```

```
52: if Usart_Data_Ready = 1 then ' if data is received
53: i = USART_Read() ' read the received data
54: if i= $7E then
55: Inicio = 1
56: n = 0
57: else
58: if Inicio = 1 then
59: Matriz[n] = i
60: n=n+1
61: if n >= Long_Dat then
62: Swt=Swt_Lectura
63: end if
64: end if
65: end if
66: else
67: Inc(ctr_sup)
68: if ctr_sup >= 140000 then 'es para salir de un lazo indefinido
69: Swt=0
70: ctr_sup = 0
71: end if
72:
73: end if
74: end sub
75: '-----------------------------------------------------------------
76: '-----------------------------------------------------------------
77: sub procedure keypad()
78: kp = 0
79: while kp = 0
```

```
80: kp = Keypad_Released
81: if TestBit(PORTA, 0) = 1 then
82: if oldstate = 0 then
83: Delay_ms(600)
84: Swt=1
85: kp = 13
86: end if
87: oldstate = 1
88: end if
89: if TestBit(PORTA, 0) = 0 then
90: oldstate = 0
91: ClearBit(PORTC,1) ' Rele apagado
92: end if
93: '-----------------------------------
94: if Inten_Errados > 3 then
95: ClearBit(PORTC,5) 'llamar
96: Delay_ms(300)
97: SetBit(PORTC,5)
98: Delay_ms(12000) 'espera 12 seg
99: ClearBit(PORTC,4) 'Cortar
100: Delay_ms(400)
101: SetBit(PORTC,4) 'Cortar
102: Delay_ms(600)
103: ClearBit(PORTC,4) 'Cortar
104: Delay_ms(400)
105: SetBit(PORTC,4) 'Cortar
106: Inten_Errados=0
107: end if
```

```
108: '-----------------------------------
109: wend
110: Delay_ms(300)
111: select case Kp
112: case 1 ' *
113:
114: case 2 ' 7
115: select case n_pulso
116: case 0
117: teclado=teclado+(7*1000)
118: Inc(n_pulso)
119: case 1
120: teclado=teclado+(7*100)
121: Inc(n_pulso)
122: case 2
123: teclado=teclado+(7*10)
124: Inc(n_pulso)
125: case 3
126: teclado=teclado+(7*1)
127: Inc(n_pulso)
128: end select
129: case 3 ' 4
130: select case n_pulso
131: case 0
132: teclado=teclado+(4*1000)
133: Inc(n_pulso)
134: case 1
135: teclado=teclado+(4*100)
```

```
136: Inc(n_pulso)
137: case 2
138: teclado=teclado+(4*10)
139: Inc(n_pulso)
140: case 3
141: teclado=teclado+(4*1)
142: Inc(n_pulso)
143: end select
144:
145: case 4 ' 1
146: select case n_pulso
147: case 0
148: teclado=teclado+(1*1000)
149: Inc(n_pulso)
150: case 1
151: teclado=teclado+(1*100)
152: Inc(n_pulso)
153: case 2
154: teclado=teclado+(1*10)
155: Inc(n_pulso)
156: case 3
157: teclado=teclado+(1*1)
158: Inc(n_pulso)
159: end select
160: case 5 ' 0
161: select case n_pulso
162: case 0
163: teclado=teclado+(0*1000)
```

```
164: Inc(n_pulso)
165: case 1
166: teclado=teclado+(0*100)
167: Inc(n_pulso)
168: case 2
169: teclado=teclado+(0*10)
170: Inc(n_pulso)
171: case 3
172: teclado=teclado+(0*1)
173: Inc(n_pulso)
174: end select
175: case 6 ' 8
176: select case n_pulso
177: case 0
178: teclado=teclado+(8*1000)
179: Inc(n_pulso)
180: case 1
181: teclado=teclado+(8*100)
182: Inc(n_pulso)
183: case 2
184: teclado=teclado+(8*10)
185: Inc(n_pulso)
186: case 3
187: teclado=teclado+(8*1)
188: Inc(n_pulso)
189: end select
190:
191: case 7 ' 5
```

```
192: select case n_pulso
193: case 0
194: teclado=teclado+(5*1000)
195: Inc(n_pulso)
196: case 1
197: teclado=teclado+(5*100)
198: Inc(n_pulso)
199: case 2
200: teclado=teclado+(5*10)
201: Inc(n_pulso)
202: case 3
203: teclado=teclado+(5*1)
204: Inc(n_pulso)
205: end select
206: case 8 ' 2
207: select case n_pulso
208: case 0
209: teclado=teclado+(2*1000)
210: Inc(n_pulso)
211: case 1
212: teclado=teclado+(2*100)
213: Inc(n_pulso)
214: case 2
215: teclado=teclado+(2*10)
216: Inc(n_pulso)
217: case 3
218: teclado=teclado+(2*1)
219: Inc(n_pulso)
```

```
220: end select
221: case 9 ' #!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
222: n_pulso =0
223: if teclado = 1234 then
224: i=0
225: for i = 0 to 8
226: SetBit(PORTE,0)
227: Delay_ms(300)
228: ClearBit(PORTE,0)
229: Delay_ms(300)
230: next i
231: SetBit(PORTC,1) 'Rele Encendido
232: Swt=0
233: end if
234: if teclado = 5678 then
235: for i = 0 to 5
236: SetBit(PORTE,1)
237: Delay_ms(300)
238: ClearBit(PORTE,1)
239: Delay_ms(300)
240: next i
241: Swt=6
242: end if
243: if teclado = 2769 then
244: for i = 0 to 5
245: SetBit(PORTE,2)
246: Delay_ms(300)
247: ClearBit(PORTE,2)
```

```
248: Delay_ms(300)
249: next i
250: Swt=21
251: end if
252: teclado =0
253: case 10 ' 9
254: select case n_pulso
255: case 0
256: teclado=teclado+(9*1000)
257: Inc(n_pulso)
258: case 1
259: teclado=teclado+(9*100)
260: Inc(n_pulso)
261: case 2
262: teclado=teclado+(9*10)
263: Inc(n_pulso)
264: case 3
265: teclado=teclado+(9*1)
266: Inc(n_pulso)
267: end select
268: case 11 ' 6
269: select case n_pulso
270: case 0
271: teclado=teclado+(6*1000)
272: Inc(n_pulso)
273: case 1
274: teclado=teclado+(6*100)
275: Inc(n_pulso)
```

```
276: case 2
277: teclado=teclado+(6*10)
278: Inc(n_pulso)
279: case 3
280: teclado=teclado+(6*1)
281: Inc(n_pulso)
282: end select
283: case 12 ' 3
284: select case n_pulso
285: case 0
286: teclado=teclado+(3*1000)
287: Inc(n_pulso)
288: case 1
289: teclado=teclado+(3*100)
290: Inc(n_pulso)
291: case 2
292: teclado=teclado+(3*10)
293: Inc(n_pulso)
294: case 3
295: teclado=teclado+(3*1)
296: Inc(n_pulso)
297: end select
298: case 13
299: end select
300: end sub
301: '------------------------------------------------------------
302: main:
303: PORTC = 0 ' Initialize PORTC
```

```
TRISA = 255 ' PORTA es entrada
TRISD = 0 ' PORTD es salida
TRISE = 0 ' PORTE es salida
TRISC = 128 ' PORTC is output solo RC7 es entrada
ADCON1= 7
OPTION_REG = $80 ' Enable T0IE pull ups disable
Usart_init(9600) ' Initialize USART module
INTCON = $A0 ' Enable T0IE
Keypad_Init(PORTB)
Inicio = 0
Swt=0
kp = 0
n_pulso =0
teclado=0
ctr_sup = 0
oldstate = 0
Inten_Errados=0
i=0
SetBit(PORTC,5) 'llamar
SetBit(PORTC,4) 'Cortar
for i = 0 to 2
SetBit(PORTE,0)
Delay_ms(100)
ClearBit(PORTE,0)
Delay_ms(100)
SetBit(PORTE,1)
Delay_ms(100)
ClearBit(PORTE,1)
```

```
Delay_ms(100)
SetBit(PORTE,2)
Delay_ms(100)
ClearBit(PORTE,2)
Delay_ms(100)
next i


while TRUE

select case Swt
case 0 '---------------------- Pregunta por el pulsante RB0,RB1,RB2

keypad()

'*****************************************************************
'proceso para identificacion Usuario
'*****************************************************************
case 1
Inc(Inten_Errados)
'-------------------------------
'Emvia el comando 0x01 CMD_REQUEST_CONNECTION
Comando= $01
Param1= $00
Param2= $00
Heder_Checksum= $01
Escritura()
Swt=2
```

```
360: case 2 '------------------------------
361: 'lectura
362: Long_Dat=24
363: Swt_Lectura=3
364: Lectura()
365: case 3 '------------------------------
366: if Matriz[3]= $01 then
367: 'Emvia el comando 0x12 CMD_IDENTIFY_FP
368: Comando= $12
369: Param1= $00
370: Param2= $00
371: Heder_Checksum= $12
372: Escritura()
373: Swt=4
374: else
375: Swt=0
376: end if
377:
378: case 4 '------------------------------
379: 'lectura
380: Long_Dat=39
381: Swt_Lectura=5
382: Lectura()
383: case 5 '------------------------------
384: if Matriz[7]= $01 then 'correcto Usuario identificada
385: SetBit(PORTC,1) 'Rele Encendido
386: Inten_Errados=0
387: i=0
```

```
388: for i = 0 to 8
389: SetBit(PORTE,0)
390: Delay_ms(300)
391: ClearBit(PORTE,0)
392: Delay_ms(300)
393: next i
394: end if
395: Swt=0
396: '*****************************************************************
397: 'proceso para nuevo usuario
398: '*****************************************************************
399: case 6 '-------------------------------
400: 'Emvia el comando 0x01 CMD_REQUEST_CONNECTION
401: Comando= $01
402: Param1= $00
403: Param2= $00
404: Heder_Checksum= $01
405: Escritura()
406: Swt=7
407: case 7 '-------------------------------
408: Long_Dat=24
409: Swt_Lectura=8
410: Lectura()
411: case 8 '-------------------------------
412: if Matriz[3]= $01 then
413: 'Envia el comando 0x2F CMD_ENTER_MASTER_MODE2 master null=3
414: Comando= $2F
415: Param1= $03
```

```
416: Param2= $00
417: Heder_Checksum= $32
418: Escritura()
419: Swt=9
420: else
421: Swt=0
422: end if
423: case 9 '------------------------------
424: Long_Dat=24
425: Swt_Lectura=10
426: Lectura()
427: case 10 '------------------------------
428: if Matriz[7]= $01 then
429: 'Envia comando 0x38 CMD_ENROLL_FP 1
430: Comando= $38
431: Param1= $00
432: Param2= $01
433: Heder_Checksum= $39
434: Escritura()
435: Swt=11
436: else
437: Swt=0
438: end if
439: case 11 '------------------------------
440: Long_Dat=24
441: Swt_Lectura=12
442: Lectura()
443: case 12 '------------------------------
```

```
444: if Matriz[7]= $01 then
445: 'Envia comando 0x68 CMD_GET_IMAGE_QAULITY
446: Comando= $68
447: Param1= $00
448: Param2= $00
449: Heder_Checksum= $68
450: Escritura()
451: Swt=13
452: else
453: Swt=0
454: end if
455: case 13 '-------------------------------
456: Long_Dat=24
457: Swt_Lectura=14
458: Lectura()
459: case 14 '-------------------------------
460: Delay_ms(2000)
461: if Matriz[7]= $01 then
462: 'Envia comando 0x38 CMD_ENROLL_FP 2
463: Comando= $38
464: Param1= $00
465: Param2= $02
466: Heder_Checksum= $3A
467: Escritura()
468: Swt=15
469: else
470: Swt=0
471: end if
```

```
472: case 15 '------------------------------
473: Long_Dat=24
474: Swt_Lectura=16
475: Lectura()
476: case 16 '------------------------------
477: if Matriz[7]= $01 then
478: 'Envia comando 0x68 CMD_GET_IMAGE_QAULITY
479: Comando= $68
480: Param1= $00
481: Param2= $00
482: Heder_Checksum= $68
483: Escritura()
484: Swt=17
485: else
486: Swt=0
487: end if
488: case 17 '------------------------------
489: Long_Dat=24
490: Swt_Lectura=18
491: Lectura()
492: case 18 '------------------------------
493: if Matriz[7]= $01 then
494: 'Envia comando 0x38 CMD_ENROLL_FP Final guarda usuario nuevo
495: Comando= $38
496: Param1= $00
497: Param2= $04
498: Heder_Checksum= $3C
499: Escritura()
```

```
Swt=19
else
Swt=0
end if
case 19 '------------------------------
Long_Dat=24
Swt_Lectura=20
Lectura()
case 20 '------------------------------ ternima nuevo usuario
if Matriz[7]= $01 then 'Correcto nuevo usuario Ingresado
i=0
for i = 0 to 8
SetBit(PORTE,1)
Delay_ms(300)
ClearBit(PORTE,1)
Delay_ms(300)
next i
end if
Swt=0
'*****************************************************************
'proceso para borrar todos los usuarios
'*****************************************************************
case 21 '------------------------------
'Emvia el comando 0x01 CMD_REQUEST_CONNECTION
Comando= $01
Param1= $00
Param2= $00
Heder_Checksum= $01
```

```
528: Escritura()
529: Swt=22
530: case 22 '------------------------------
531: Long_Dat=24
532: Swt_Lectura=23
533: Lectura()
534: case 23 '------------------------------
535: if Matriz[3]= $01 then
536: 'Envia el comando 0x2F CMD_ENTER_MASTER_MODE2 master null=3
537: Comando= $2F
538: Param1= $03
539: Param2= $00
540: Heder_Checksum= $32
541: Escritura()
542: Swt=24
543: else
544: Swt=0
545: end if
546: case 24 '------------------------------
547: Long_Dat=24
548: Swt_Lectura=25
549: Lectura()
550: case 25 '------------------------------
551: if Matriz[7]= $01 then
552: 'Envia comando 0x23 CMD_DELETE_ALL_FP
553: Comando= $23
554: Param1= $00
555: Param2= $00
```

```
556: Heder_Checksum= $23
557: Escritura()
558: Swt=26
559: else
560: Swt=0
561: end if
562: i=0
563: for i = 0 to 8
564: SetBit(PORTE,2)
565: Delay_ms(300)
566: ClearBit(PORTE,2)
567: Delay_ms(300)
568: next i
569:
570: case 26 '------------------------------ Usuarios Borrados
571: Long_Dat=24
572: Swt_Lectura=27
573: Lectura()
574: case 27 '------------------------------
575: Swt=0
576: case 28 '------------------------------
577:
578: case 29 '------------------------------
579:
580: case 30 '------------------------------
581:
582: end select
583:
```

584: **wend**

585: **end**.

Anexo II

Programa del PIC del módulo de control celular PIC16F84A

```
1: program Tesis_E
2: ' Discador para nokia 1100 con pic 16f84a con protocolo fbus
3: ' Este programa puede llamar a un número guardado en la memoria 2 y
4: ' cortar a través del protocolo fbus de nokia, se utiliza un cristal
5: ' de 4 Mhz.
6: '--------------------------------------------------------------------
7: 'Declaracion de variables
8: dim i as byte
9: '--------------------------------------------------------------------
10: sub procedure Reset_celular()
11: Soft_Uart_Write ($1E)
12: Soft_Uart_Write ($00)
13: Soft_Uart_Write ($0C)
14: Soft_Uart_Write ($40)
15: Soft_Uart_Write ($00)
16: Soft_Uart_Write ($06)
17: Soft_Uart_Write ($00)
18: Soft_Uart_Write ($01)
19: Soft_Uart_Write ($64)
20: Soft_Uart_Write ($03)
21: Soft_Uart_Write ($01)
22: Soft_Uart_Write ($60)
23: Soft_Uart_Write ($77)
```

```
24: Soft_Uart_Write ($24)
25: end sub
26: '----------------------------------------------------------------
27: main:
28: trisa = $85
29: porta = $05
30: trisb = $86
31: portb = $06
32: trisa = 0 'se programa el puerto A como salida
33: trisb = $0F 'se programa los 4 primeros bit del puerto B como entrada
34: OPTION_REG = 7 'se habilitan resistencias de Pull Up
35: Soft_Uart_Init(PORTA, 0, 1, 9600, 0)'Configura el puerto para envio de datos
software
36: 'reset celular
37: Soft_Uart_Write ($1E)
38: Soft_Uart_Write ($00)
39: Soft_Uart_Write ($0C)
40: Soft_Uart_Write ($40)
41: Soft_Uart_Write ($00)
42: Soft_Uart_Write ($06)
43: Soft_Uart_Write ($00)
44: Soft_Uart_Write ($01)
45: Soft_Uart_Write ($64)
46: Soft_Uart_Write ($03)
47: Soft_Uart_Write ($01)
48: Soft_Uart_Write ($60)
49: Soft_Uart_Write ($77)
50: Soft_Uart_Write ($24)
```

```
51: Delay_ms(100)
52: 'reset celular
53: Soft_Uart_Write ($1E)
54: Soft_Uart_Write ($00)
55: Soft_Uart_Write ($0C)
56: Soft_Uart_Write ($40)
57: Soft_Uart_Write ($00)
58: Soft_Uart_Write ($06)
59: Soft_Uart_Write ($00)
60: Soft_Uart_Write ($01)
61: Soft_Uart_Write ($64)
62: Soft_Uart_Write ($03)
63: Soft_Uart_Write ($01)
64: Soft_Uart_Write ($60)
65: Soft_Uart_Write ($77)
66: Soft_Uart_Write ($24)
67: Delay_ms(100)
68: 'reset celular
69: Soft_Uart_Write ($1E)
70: Soft_Uart_Write ($00)
71: Soft_Uart_Write ($0C)
72: Soft_Uart_Write ($40)
73: Soft_Uart_Write ($00)
74: Soft_Uart_Write ($06)
75: Soft_Uart_Write ($00)
76: Soft_Uart_Write ($01)
77: Soft_Uart_Write ($64)
78: Soft_Uart_Write ($03)
```

```
79: Soft_Uart_Write ($01)
80: Soft_Uart_Write ($60)
81: Soft_Uart_Write ($77)
82: Soft_Uart_Write ($24)
83: Delay_ms(100)
84:
85: 'Lazo principal
86: while TRUE
87:
88: if TestBit(PORTB, 0) = 0 then 'testea si el pulsador para llamar está
89: ' precionado llamar
90: Soft_Uart_Write ($1E)
91: Soft_Uart_Write ($00)
92: Soft_Uart_Write ($10)
93: Soft_Uart_Write ($01)
94: Soft_Uart_Write ($00)
95: Soft_Uart_Write ($11)
96: Soft_Uart_Write ($00)
97: Soft_Uart_Write ($01)
98: Soft_Uart_Write ($00)
99: Soft_Uart_Write ($01)
100: Soft_Uart_Write ($01)
101: Soft_Uart_Write ($00)
102: Soft_Uart_Write ($32)
103: Soft_Uart_Write ($05)
104: Soft_Uart_Write ($01)
105: Soft_Uart_Write ($05)
106: Soft_Uart_Write ($00)
```

```
107: Soft_Uart_Write ($02)
108: Soft_Uart_Write ($00)
109: Soft_Uart_Write ($00)
110: Soft_Uart_Write ($01)
111: Soft_Uart_Write ($01)
112: Soft_Uart_Write ($45)
113: Soft_Uart_Write ($00)
114: Soft_Uart_Write ($78)
115: Soft_Uart_Write ($13)
116: end if
117:
118: if TestBit(PORTB, 1) = 0 then 'estea si el pulsador para cortar está precionado
119: Soft_Uart_Write ($1E)
120: Soft_Uart_Write ($00)
121: Soft_Uart_Write ($0C)
122: Soft_Uart_Write ($01)
123: Soft_Uart_Write ($00)
124: Soft_Uart_Write ($07)
125: Soft_Uart_Write ($00)
126: Soft_Uart_Write ($01)
127: Soft_Uart_Write ($00)
128: Soft_Uart_Write ($08)
129: Soft_Uart_Write ($60)
130: Soft_Uart_Write ($01)
131: Soft_Uart_Write ($63)
132: Soft_Uart_Write ($00)
133: Soft_Uart_Write ($11)
134: Soft_Uart_Write ($0E)
```

```
135: Delay_ms(100)
136: 'reset celular
137: Soft_Uart_Write ($1E)
138: Soft_Uart_Write ($00)
139: Soft_Uart_Write ($0C)
140: Soft_Uart_Write ($40)
141: Soft_Uart_Write ($00)
142: Soft_Uart_Write ($06)
143: Soft_Uart_Write ($00)
144: Soft_Uart_Write ($01)
145: Soft_Uart_Write ($64)
146: Soft_Uart_Write ($03)
147: Soft_Uart_Write ($01)
148: Soft_Uart_Write ($60)
149: Soft_Uart_Write ($77)
150: Soft_Uart_Write ($24)
151: Delay_ms(100)
152: 'reset celular
153: Soft_Uart_Write ($1E)
154: Soft_Uart_Write ($00)
155: Soft_Uart_Write ($0C)
156: Soft_Uart_Write ($40)
157: Soft_Uart_Write ($00)
158: Soft_Uart_Write ($06)
159: Soft_Uart_Write ($00)
160: Soft_Uart_Write ($01)
161: Soft_Uart_Write ($64)
162: Soft_Uart_Write ($03)
```

```
163: Soft_Uart_Write ($01)
164: Soft_Uart_Write ($60)
165: Soft_Uart_Write ($77)
166: Soft_Uart_Write ($24)
167: Delay_ms(100)
168: 'reset celular
169: Soft_Uart_Write ($1E)
170: Soft_Uart_Write ($00)
171: Soft_Uart_Write ($0C)
172: Soft_Uart_Write ($40)
173: Soft_Uart_Write ($00)
174: Soft_Uart_Write ($06)
175: Soft_Uart_Write ($00)
176: Soft_Uart_Write ($01)
177: Soft_Uart_Write ($64)
178: Soft_Uart_Write ($03)
179: Soft_Uart_Write ($01)
180: Soft_Uart_Write ($60)
181: Soft_Uart_Write ($77)
182: Soft_Uart_Write ($24)
183: Delay_ms(100)
184: end if
185:
186: wend
187: end.
```

Anexo III

Comandos de programación e intercomunicación del lector de huella digital

Code	System Information	Value Range	Default Value	
0x02	SI_USING_LOG	True/False	False	
0x17	SI_IDENTIFY_TIMEOUT	255 or 10~250	30	100ms tick
0x18	SI_RELAY_TIME	0 or 1~100	10	100ms ticks
0x19	SI_CAPTURE_TIMEOUT	More than 10	50	100ms ticks
0x20	SI_IMAGE_BRIGHTNESS	0~100	45	100 - brightest
0x21	SI_IMAGE_GAIN	1,2,4,8	2	
0x22	SI_IMAGE_CONTRAST	0~100	20	
0x28	SI_ADAPTIVE_CAPTURE	True/False	False	
0x30	SI_VERIFY_SECURITY_LEVEL	1~9	5	
0x31	SI_IDENTIFY_SECURITY_LEVEL	6~9	8	
0x32	SI_REGISTER_QUALITY	30~100	40	
0x33	SI_VERIFY_QUALITY	10~100	30	
0x49	SI_CHANNEL1_BAUDRATE	0 – 115200 1 – 57600 2 – 38400 3 – 19200 4 – 9600	4	
0x4A	SI_CURR_CHANNEL_BAUDRATE			
0x50	SI_MAX_USER			
0x51	SI_FP_FULL_ROTATION	True/False	False	
0x52	SI_LENGTH_OF_USER_ID	4~15	10	
0x53	SI_NUM_OF_ADAPTIVA_CAP	1~10	5	
0x54	SI_MAX_TEMPLATE			Read Only