"Responsabilidad con pensamiento positivo"

UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN

CARRERA:

ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA:

MODELO DE SISTEMA DE CONTROL PARA MEJORAR LA SEGURIDAD EN EL SERVICIO DE TRANSPORTES DE TAXI USANDO TECNOLOGÍA NFC.

AUTOR:

ORBE PAZMIÑO FERNANDO RAFAEL

TUTOR:

Mg. ARMANDO MÉNDEZ

AÑO 2014

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación certifico:

Que el trabajo de titulación **"MODELO DE SISTEMA DE CONTROL PARA MEJORAR LA SEGURIDAD EN EL SERVICIO DE TRANSPORTES DE TAXI USANDO TECNOLOGÍA NFC."**, presentado por el Sr. Fernando Rafael Orbe Pazmiño, estudiante de la Carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D. M., Septiembre de 2014

TUTOR

Mg. Armando Méndez

UNIVERSIDAD TECNOLÓGICA ISRAEL

AUTORÍA DE TRABAJO DE TITULACIÓN

Yo Fernando Rafael Orbe Pazmiño, en calidad de estudiante de la Carrera de Ingeniería en Electrónica Digital y Telecomunicaciones, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Quito D. M., Septiembre de 2014

Fernando Rafael Orbe Pazmiño

CC: 172103915-2

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado, aprueban el Trabajo de Titulación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D. M., Septiembre de 2014

Para constancia firman:

TRIBUNAL DE GRADO

PRESIDENTE

MIEMBRO 1

MIEMBRO 2

AGRADECIMIENTO

Agradezco a mis padres que confiaron en mí y siempre me brindaron su apoyo en todas mis decisiones, también me enseñaron a luchar por mis sueños Y persistir en cumplir los mismos. Agradezco también a las personas que fueron y son parte de mi vida e inspiración de crecimiento para la misma.

DEDICATORIA

Dedico el proyecto a mis padres por ser la mayor inspiración de esfuerzo y constancia en el desarrollo de toda mi carrera guiándome por un buen camino lleno de paciencia, amor, comprensión. También lo dedico a Alexandra que siempre insistió día a día para que terminara mi carrera y seguir en el camino del aprendizaje.

RESUMEN

El presente proyecto consiste en el análisis, diseño implementación de un dispositivo modelo de seguridad que pueda ser instalado en las unidades de taxis y sea de fácil manejo para el usuario.

Para el análisis del proyecto obtuvo información acerca de los dispositivos y temas que se manejan para su elaboración como es; Dispositivos electrónicos de Arduino, Tecnología NFC y comunicación (RS-232) de la misma con modem GSM. También la programación, requerida para el correcto funcionamiento del proyecto, sé la realizo de forma que sea amigable con el usuario y programador de tag´s.

Con la implementación y montaje del proyecto, este dispositivo cumplirá dos diferentes tareas según el uso que le quiera dar el propietario. La primera tarea, en la que el propietario puede aplicar este dispositivo de seguridad, es la de grabar en una tag dos números de teléfonos requeridos por el usuario. La segunda tarea, en la que el propietario puede aplicar este dispositivo de seguridad, es en una unidad de taxi, por lo que este dispositivo servirá como lector de la tag que contiene los dos números de teléfono del usuario a los mismos que inmediatamente se envía un SMS con la información de la unidad de taxi que abordó este usuario. De esa manera esta información facilita la investigación y denuncia del algún acto delictivo que haya tenido en alguna unidad. Además la misma información puede ser usada como un lazo de confianza entre el chofer de taxi y pasajero, ya que si el cliente recibió una buena atención, el mismo tendrá en cuenta la información adquirida para otras oportunidades llamar a su taxi de preferencia.

ABSTRACT

This project involves the analysis, design implementation of a model safety device that can be installed in the units of taxis and is easy to use for the user.

For analysis of the project obtained information about the devices and themes that are handled for processing as it is; Electronic Devices Arduino, NFC technology and communication (RS-232) with the same GSM modem. Also the programming required for the proper functioning of the project, I know the way I make it user friendly and programmer's tag.

With the implementation and installation of the project, this device will be two different tasks depending on the use you want to give the owner. The first task, which the owner can apply this security device is to tax the two numbers in a phone tag required by the user. The second task, which the owner can apply this security device is in a taxi drive, so this device will serve as a reader of the tag containing the two phone numbers to the same user is sent immediately an SMS to the unit information taxi approached this person. Thus this information facilitates the investigation and reporting of any criminal act that has been in any unit. In addition, the same information can be used as a bond of trust between the taxi driver and passenger, because if the client received good care, it will consider the information gathered for other opportunities call your taxi preference.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE TE TABLAS

CAPÍTULO 1

1.1. INTRODUCCIÓN

1.1.1. PLANTEAMIENTO DEL PROBLEMA

La mayoría de los asaltos y secuestros exprés se los realiza en unidades de taxis por lo que se requiere un método económico y practico de prevenir este problema para los usuarios

Figura 1.1 Imagen ilustrando el problema de delincuencia en taxis.
Fuente: (RUEDA, 2013)

1.1.1.1. ANTECEDENTES

La delincuencia es uno de los problemas más fuertes para el país y sobre todo en ciudades grandes de uso masivo de medios de transporte sobre todo el uso de taxis donde en la actualidad ha aumentado tanto en número de cooperativas, unidades y a la vez en delincuencia usando la confianza del usuario hacia este medio de transporte.

Este problema cada día crece en mayor magnitud en las diferentes ciudades del país y principalmente es más notorio en las ciudades principales.

La falta de información y precaución del usuario al momento de tomar una unidad de taxi permite que sea mucho más fácil para los delincuentes inclinarse en mayor cantidad

hacia esa forma de delito. Después de que el usuario ya ha sido víctima de secuestro o asalto el mismo que al realizar la denuncia, no presenta o no tiene las suficientes pruebas para una mejor investigación de parte de la policía para estos casos.

NFC (Near field comunication) Comunicación de campo cercano , la tecnología NFC es una tecnología de corto alcance y de alta frecuencia que permite el intercambio de datos entre dispositivos, Requiere típicamente una distancia de 4 cm o menos para iniciar una conexión. NFC permite compartir pequeñas cargas útiles de datos entre una etiqueta NFC y un dispositivo NFC.

Esta tecnología ha sido poco explotada en Ecuador, siendo esta una tecnología de bajo costo, comunicación garantizada, segura y de fácil uso o acceso ya que NFC es una tecnología que se comunica mediante inducción en un campo magnético, en donde dos espiras de antenas son colocadas dentro de sus respectivos campos cercanos para transmitir datos.

1.1.1.2. INVESTIGACIÓN APLICADA A LOS TAXISTAS, REFERENTE A LA SEGURIDAD EN LAS UNIDADES DE TAXIS.

En la investigación, se obtuvo la ayuda de los socios y de la cooperativa de taxis "Comité del Pueblo". Esta cooperativa cuenta con más de 150 unidades de taxi dentro de la Ciudad de Quito.

Dentro de la investigación, se encontró que los taxistas se sienten más seguros si el servicio es solicitado por central o llamada a domicilio, también muchos de los taxistas tienen una preferencia a ciertos clientes ya conocidos y que muchos de ellos diariamente solicitan sus servicios de transporte. También en las entrevistas realizadas a los choferes de las unidades de taxis también se habló acerca de los dispositivos de seguridad implementados en los taxis para su seguridad y tenemos algunos como:

GPS: En el caso de la Coop. Comité del Pueblo, estos dispositivos están instalados en el 80% de sus unidades, pero los mismos son usados en mayor frecuencia para control de actividades de los choferes en las unidades de taxis y poco usados para seguridad de los mismos.

También otro dispositivo de seguridad mencionado en las entrevistas son las cámaras de seguridad las cuales solo están instaladas en el 25% de las unidades de taxis, lo que indica que aún no se puede demostrar la efectividad de este método de seguridad.

El más importante y conocido método seguridad implementado en los taxis del D.M. Quito es Taxi Seguro, el cual fue presentado por: El Ministerio Coordinador de Seguridad, MICS, el ECU 911 y la Agencia Nacional de Tránsito, ANT.

Figura 1.2 Imagen de cámara Taxi Seguro
Funete: (El Universo, 2013)

Taxi Seguro, El proyecto representativo es el de las cámaras de seguridad implementadas en las unidades de taxis, las cuales permiten identificar las personas que se encuentran dentro de las unidades en el momento en que se presiona un botón (botón de pánico). También taxi Seguro es parte de una aplicación creada en Brasil y extendida en algunos países de Latinoamérica. Esta aplicación ayuda a la información de la unidad que tomará por teléfono, pero para solicitar una unidad de taxi se requiere un teléfono Smartphone y con plan de datos o conexión wifi tanto para el usuario como también

para el chofer de taxi. Por lo cual este método no es tan utilizado en Quito ni en el país, sin contar que muy pocas unidades de taxi tienen conocimiento de esta aplicación.

Dentro de las entrevistas a usuarios de las unidades de taxis, pocos conocen los métodos de seguridad implementados para taxis. El 60% tenía conocimiento del botón de pánico y el 1% supo acerca de la aplicación de Taxi Seguro para celulares.

Además la mayoría de usuarios, toma cualquier unidad de taxi sin alcanzar a ver la placa o los datos del mismo, por lo que al tomar una unidad tiene un mayor riesgo de ser víctima de delincuencia y lo que es peor no tener información de que unidad o como realizar una denuncia para empezar una investigación del caso.

1.1.2. FORMULACIÓN DEL PROBLEMAS

Problema Principal

- No existe un modelo de dispositivo de seguridad que permita al usuario adquirir, guardar y dar a conocer al usuario a una persona externa, la información inmediata de la unidad de taxi que abordó.

Problemas Secundarios

- Actualmente en el Ecuador no se dispone de un análisis de la tecnología NFC aplicado en el uso de los medios de transporte.
- No existe el diseño y programación para un dispositivo que brinde información de la unidad de taxi abordada mejorando así la seguridad para usuario y conductor de la unidad.
- No se posee un dispositivo que sea de fácil implementación, montaje y manejo para los usuarios.

1.1.3. SISTEMATIZACIÓN

1.1.3.1. Diagnóstico

Estos problemas cada día más frecuentes en las ciudades principales de Ecuador como: Quito, Cuenca y Guayaquil, han causado un gran temor y riesgo en los usuarios al momento de tomar una unidad de taxi

- Riesgo de tomar una unidad de taxi ilegal
- Riesgo de tomar una unidad te taxi que ha sido robada y estar siendo usada para fines delictivos.
- Falta de información de las unidades o compañías de las mismas.

1.1.3.2. Pronóstico

Con la falta de exigencia y seguridad para el uso de estas unidades de taxis los usuarios serán más vulnerables a ser víctimas de actos delictivos disminuyendo así también la garantía de seguridad dentro de estas unidades.

1.1.3.3. Control de Pronostico

Con la implementación de este dispositivo NFC instalado en la unidad permitirá al usuario obtener información inmediata de la unidad que tomó y así:

- En caso de problemas con el conductor o acto delictivo dentro de la unidad, tenga la información necesaria para una denuncia y empezar una investigación.
- También guardar la información de la unidad en caso de pérdida u olvido de objetos importantes o si también le pareció una unidad muy confiable y requiera en un futuro nuevamente de sus servicios.

1.1.4. JUSTIFICACIÓN

El estudio de este proyecto permitirá conocer el funcionamiento y aplicación de tecnología NFC acoplándolo con una memoria de control datos los mismos que serán transmitidos por medio de un dispositivo GSM. Este prototipo servirá como base para mejorar seguridad y confiabilidad de los usuarios en el momento de tomar un taxi.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Estudiar, diseñar e implementar un modelo de control para mejorar la seguridad en el servicio de transportes de taxi usando tecnología NFC

1.2.2. OBJETIVOS ESPECÍFICOS

1.- Analizar las características de los dispositivos requeridos para diseñar e implementar un hardware que permita la comunicación entre los dispositivos NFC y GSM.

2.- Diseñar una programación adecuada para que el dispositivo de seguridad cumpla con las exigencias de leer o grabar datos en las tag NFC y enviar un SMS al teléfono del usuario y otro a una persona externa, con las descripciones del taxi abordado.

3.- Realizar el acoplamiento de los dispositivos y programación que permita el fácil uso del modelo de seguridad NFC.

CAPÍTULO 2

2.1. MARCO DE REFERECIA

En este capítulo se encuentra información y descripción de los dispositivos tecnología que se aplicará al proyecto

2.1.1. MARCO TEÓRICO

2.1.1.1. Tecnología NFC

Introducción a la tecnología NFC

NFC (Near Field Comunication) o Comunicación de Corto Alcance. NFC fue creado en el año del 2002 con un proyecto encabezado por Nokia, Philips y Sony las mismas que componen la asociación NFC Fórum. De esta forma NFC aparece como una evolución en el uso de aplicaciones dentro de la telefonía móvil. (NFC Forum, 2014)

Figura 2.1 Ideas de aplicaciones de NFC Forum
Fuente: (NFC Forum, 2014)

La base de la tecnología NFC parte de otras tecnologías de radiofrecuencia ya existentes como RFID, por lo cubre los protocolos de comunicación y formatos de intercambio de datos basados en normas de identificación como ISO/IEC 14443 y FeliCa. Los estándares incluyen ISO/IEC 18092. RFID permite el uso de una pequeña memoria llamado tag RFID que se la ubica en cualquier objeto aplicable, con el objetivo de identificación y seguimiento usando ondas de radio. (NFC Forum, 2014)

Características

NFC es una tecnología de comunicaciones inalámbrica de corto alcance y alta frecuencia, derivada de la tecnología RFID, que permite el intercambio de datos cercanos. Estos dispositivos se llaman Iniciador (el que origina la transmisión) y Tag (el receptor). NFC funciona en la banda de frecuencia de 13,56 MHz, y un ancho de banda que oscila 7 kHz a cada lado y trabaja a una distancia máxima de 20 cm. El funcionamiento de NFC se basa en el principio de inducción electromagnética, en el cual dos circuitos inductivos cercanos comparten energía por lo que pueden transmitir datos a distancia de pocos centímetros (CASE Congreso Argentimo de Sistemas Embebidos, 2011, pág. 97)

NFC está definido en dos modos de operación que son activos y pasivos por lo que la característica principal del NFC es combinar ambas funciones de tag, lectura / escritura.

Actualmente NFC ofrece tasas de transferencia de datos de 106, 212 y 424 Kbps.

Un dispositivo NFC que comienza la comunicación y controla el intercambio de comunicación es conocido como iniciador y el que responde al iniciador es conocido como objeto. La comunicación puede realizarse de modo pasivo o activo como se puede observar en la Tabla. (Foro de Nuevas Tecnologías en el Transporte, ITS España, 2013)

	Iniciador	Tag
Activo	Posible	Posible
Pasivo	No Posible	Posible

Tabla 1 Tipo de enlace NFC
Fuente: (Idoneum Electronic Identity, 2011)

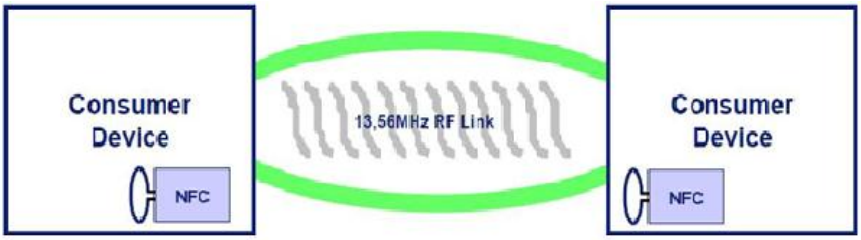


Figura 2.2 Ilustración de comunicación NFC
Fuente: (Josef Noll, 2005, pág. 17) NFC standardisation.

2.1.1.2. Aplicación de la Tecnología NFC

Este tipo de tecnología podemos encontrarle en ciertos dispositivos móviles los mismos que están programados para realizar actividades de NFC según el usuario lo requiera. También NFC es una tecnología de comunicación inalámbrica que cuenta con el desarrollo de aplicaciones Java para estos dispositivos a nivel software.

La tecnología NFC nos permite facilitar o agilitar actividades en diversas áreas como:

- Control de acceso
- Organización
- Intercambio de información
- Pagos
- Transporte
- Seguridad

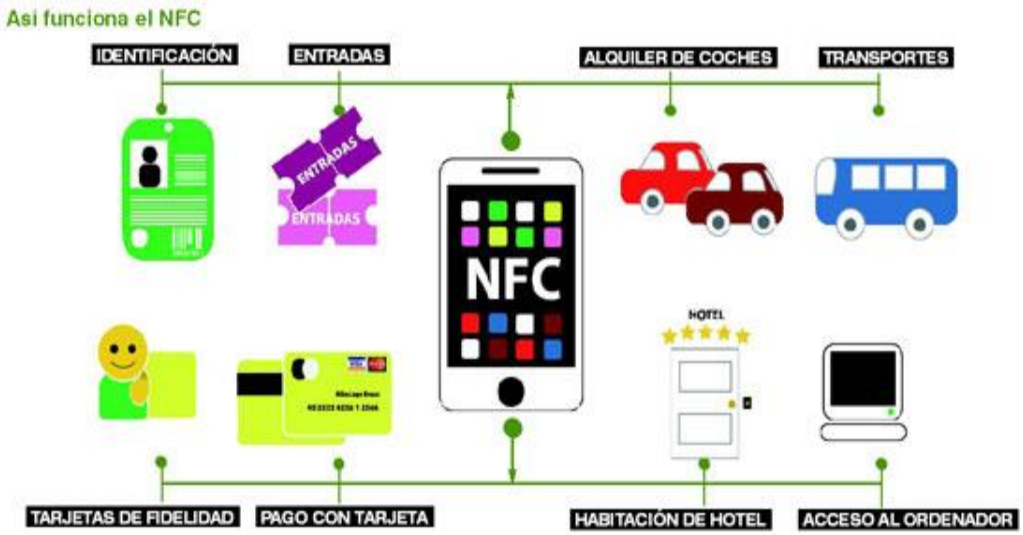


Figura 2.3 Aplicaciones de tecnología NFC
Fuente: (Telecomunicaciones, 2006)

2.1.1.2.3. Dispositivos NFC

La tecnología NFC utiliza los siguientes dispositivos inteligentes:

- Teléfono móvil con NFC: Los móviles NFC son los dispositivos más importantes en la integración de la NFC.
- Lector NFC: Un lector NFC es capaz de transmitir datos con otro componente NFC como por ejemplo el más común es el terminal de venta (POS), que se puede realizar pagos cuando el dispositivo NFC se lo acerca hacia el lector NFC.
- NFC tag: una etiqueta NFC es en realidad una etiqueta RFID que no tiene una fuente de alimentación integrada y toda tag contiene una (ID) identificación única para cada una

La comunicación NFC entra en funcionamiento en el momento mismo que dos dispositivos NFC son acercados mutuamente, uno funcionan como iniciador y otro receptor de datos según el modo de operación. (NFC Forum, 2014)

Iniciador	Tag
Celular con NFC	Tarjeta NFC
Celular con NFC	Celular con NFC
Lector NFC	Tarjeta NFC

Tabla 2 Tipo de comunicación NFC
Fuente: (Idoneum Electronic Identity, 2011, pág. 2)

2.1.1.2.4. Modos de operación NFC

Teniendo en cuenta el modo de operación los protocolos más significativos son:

- NFCIP-1: Este modo de operación combina dos protocolos de comunicación que pertenecen al RFID como son el MIFARE y el FeliCa e incluye en ellos nuevos protocolos de transporte.
- NFCIP-2: Hace posible la compatibilidad de la combinación del NFC con lectores RFID. (CASE Congreso Argentimo de Sistemas Embebidos, 2011, pág. 98)

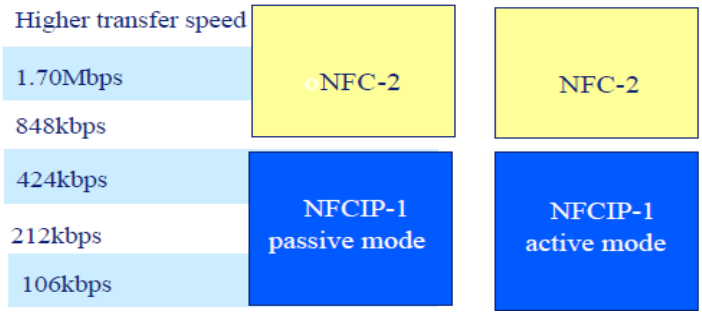


Figura 2.4 Tipos de operación NFC
Fuente: (ECMA International, 2002, pág. 13)

Para establecer una comunicación de NFC existen dos modos:

- **Modo pasivo:** En este modo debe haber existir un dispositivo receptor y otro emisor, el emisor dispone de una fuente eléctrica propia para funcionar, y debe generar una señal para el intercambio de datos. El

dispositivo receptor no posee baterías y debe aprovechar el campo incidente del emisor para su funcionamiento.

- **Modo activo:** En este modo los dispositivos poseen energía propia por lo que ambos generan el campo electromagnético para la transferencia de datos. (CASE Congreso Argentimo de Sistemas Embebidos, 2011, pág. 98)

En los dispositivos NFC es posible hacer la comunicación con otro dispositivo NFC actuando como Tag o como lector/escritor para ello el NFC Fórum define los siguientes modos de operación:

- **Peer to Peer:** Dos dispositivos NFC se comunicarán entre ellos en forma activa o pasiva según el modelo de maestro/esclavo, el iniciador o maestro inicia la transferencia la transferencia de datos que el objetivo o receptor esclavo espera para dar la respuesta. Este modo es utilizado cuando surge la necesidad de transmitir una reducida cantidad de datos.
- **Lectura/ escritura**: El dispositivo NFC actúa como un lector de tarjetas activo, por lo que generara un campo de RF para comunicarse con otros dispositivos como Tag's. En este, modo se tiene la capacidad de leer y escribir las Tag's. Una vez establecida la comunicación es posible el intercambio de texto en pequeñas cantidades, una dirección de internet o un número de teléfono.
- **Emulación de tarjeta inteligente**: El dispositivo NFC actúa como una Tag o etiqueta NFC sin contacto pasiva. Como dispositivo pasivo no debe generar ningún campo de radiofrecuencia. Un lector puede identificar a un dispositivo NFC como si este fuera una etiqueta NFC o tarjeta inteligente de este modo se puede ser usado para medios de pagos y transacciones bancarias, control de

accesos e intercambio de datos. (CASE Congreso Argentimo de Sistemas Embebidos, 2011, pág. 98)

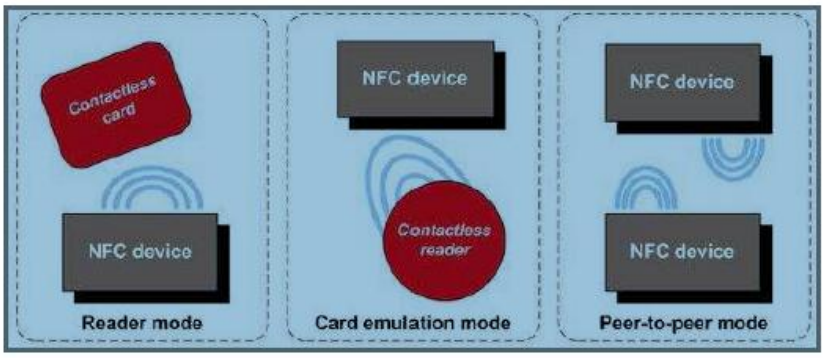


Figura 2.5 Modos de conexión NFC
Fuente: (ECMA International, 2002)

2.1.1.2.5. Composición de un dispositivo NFC

- **Chip NFC:** Este es la memoria en la cual contiene una ID de identificación y un espacio para guardar información mínima en Bits.
- **Antena:** Es la que permite la comunicación y el intercambio de datos con otro dispositivo NFC a muy poca distancia. esta antena es la que genera el campo magnético para el intercambio de datos.
- **Etiquetas NFC:** Implementan un almacenamiento pasivo en la espera de que algún lector NFC requiera información que retiene la etiqueta. (Mundo NFC, 2012)

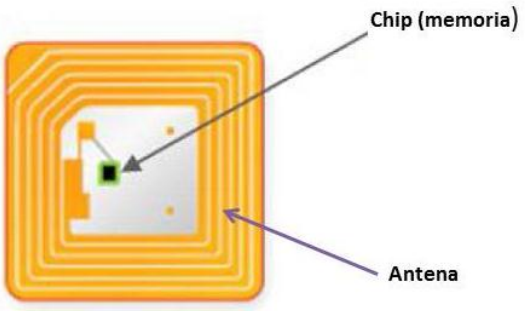


Figura 2.6 Composición interna de una tag NFC o RFID
Fuente: (Mundo NFC, 2012)

2.1.1.2.6. Comunicación NFC

Existen cuatro etapas importantes para establecer intercambio de datos o comunicación de los dispositivos NFC que son:

- **Descubrimiento:** Es la fase donde se rastrean mutuamente los dispositivos NFC para luego iniciar el reconocimiento.
- **Autenticación:** Los dispositivos están autorizados para iniciar una comunicación entre ellos o establecer una conexión a través de un cifrado.
- **Negociación:** Se definen parámetros como: tasa de transmisión, identificación, aplicación, y acción solicitada.
- **Transferencia:** Se realiza el intercambio de datos
- **Confirmación:** se confirma el estado y transferencia de datos.

Formato de comunicación

Para que exista compatibilidad y comunicación entre dispositivos NFC y RFID de diferentes fabricantes, NFC Fórum define un formato de datos estandarizados.

- **NFC Data Exchange Format (NDEF):** Es un formato de datos para el intercambio de información entre dispositivos NFC activo y una etiqueta pasiva o entre dispositivos NFC activos. NDEF es un formato de mensaje binario que encapsula una o más cargas útiles de aplicación definidas en un solo mensaje.
- **NFC Record Type Definition (RTD):** Especifica tipos de registro estándar que pueden ser enviados en los mensajes intercambiados entre los dispositivos NFC.
    - **Smart Poster RTD:** Este formato define un tipo de NFC Fórum de cómo establecer las URL, SMS o números de teléfono en una etiqueta NFC o como transportarlos entre dispositivos. Por lo que es capaz de almacenar información adicional en una etiqueta NFC.

- **Text RTD:** Este formato es para registro que solo contienen texto de forma libre o sea sin formato.
- **Uniform Resource Identifier:** Para registros que se refieren a un recurso de internet.

  (CASE Congreso Argentimo de Sistemas Embebidos, 2011, pág. 99)

2.1.1.3. Tecnología GSM

Es el sistema global para las comunicaciones móviles (Global System for Movile Comunications), es un estándar para comunicación utilizando teléfonos móviles que incorporan tecnología digital. Este tipo de tecnología móvil es de segunda generación (2G) la cual es: de alta velocidad y mayor cobertura en transmisión y recepción de voz y datos, envío y recepción de mensajes cortos (SMS) también por ser digital cualquier cliente de GSM puede conectarse a través de su teléfono con su ordenador y puede hacer, enviar y recibir mensajes por e-mail, navegar por Internet, llamadas en espera, identificador entre otros servicios. (GSM and TDMA Technology, pág. 320)

El sistema GSM requiere de un módulo de identidad del suscriptor, la cual se denomina tarjeta SIM. Esta es una tarjeta inteligente que contiene la información del usuario, parámetros de referencia y directorio telefónico.

El estándar GSM para la mayor parte del mundo está en el rango de 900MHz y 1800MHz, en el Ecuador se utilizan las bandas de frecuencia de 850MHz y 1900MHz, (GSM and TDMA Technology, pág. 321)

2.1.1.3.1. Servicio de mensajes cortos de texto SMS

El servicio de SMS permite enviar un mensaje de texto entre una Estación Móvil (MS) y otra entidad denominada SME (Entidad de mansajes cortos) por medio de un centro de servicio SMSC (Centro de servicio de mensajes cortos). (Ingeniatic, 2011)

Formato del SMS

Dentro de las especificaciones de los SMS existe la posibilidad de realizar el envío de mensajes de dos maneras, las mismas que son:

- **Modo PDU,** esta estructura de mensaje lleva consigo bits de información específica, además de funciones de control para presentación del mensaje.

- **Modo Texto,** esta estructura está conformada por caracteres de texto, números y símbolos, es un modo de gama media que no se encuentra en todos los terminales MS.
  (Ingeniatic, 2011)

2.1.1.4. Modem GSM

Estos son dispositivos que se conectan a la red telefónica celular, como cualquier otro móvil de igual manera permite el uso de tarjeta SIM, también tiene la característica de conectarse a un computador o dispositivos que posean comunicación a través de un puerto serial.

Para la comunicación entre un modem GSM con un computador este dispositivo acepta comandos para la recepción y transmisión de datos. (Fernando Orbe, 2014)

2.1.1.4.1. Comunicación del Modem GSM

Existen comandos que requieren el uso del puerto serial por lo que se usan comandos AT Cuando una aplicación recibe una notificación, debe enviar un comando AT para leer el contenido del mensaje. Es posible, sin embargo, configurar el módem para que envíe el contenido del mensaje junto con la notificación. (Fernando Orbe, 2014)

2.1.1.5. Comandos AT

Es un estándar de comunicación entre el modem y el usuario permitiéndole configurarlos o programarlos según lo requiera.

Este lenguaje fue desarrollado por la compañía Hayes Comunicatios en 1977. Los caracteres (AT), que contienen todos los comandos, significa (Atenttion). (LUCKYSTAR'S OPINIONS, 2009)

El objetivo principal de los comandos AT es la comunicación con los modem sin embargo la tecnología GSM adoptó este lenguaje para poder comunicarse con sus terminales, de esta manera a todos los teléfonos GSM utilizando los comandos AT, se pueden configurar o proporcionar instrucciones tales como realizar llamadas de datos, voz , leer o escribir y enviar mensajes entre otros. (Manualslib, 2008)

2.1.1.5.1 Nomenclatura de comandos AT para comunicación con modem GSM

Para ejecutar un comando AT se debe primeramente ante poner el prefijo AT.

AT +	CGMI=1	<CR>
PREFIJO	COMANDO	SUFIJO

• El prefijo de los comandos AT debe ser la cadena de caracteres "AT", el signo "+" también debe ser colocado después de estos caracteres.

• El sufijo de los comandos AT debe ser <CR> (Retorno de carro), que es equivalente a ENTER.

• Cuando se coloca el signo "=" a un comando, se está configurando un parámetro, cuando se coloca el signo de interrogación "?" se está pidiendo información.

La respuesta del modem ante un comando es la siguiente:

<CR><LF> OK <CR><LF>

PREFIJO COMANDO SUFIJO

• Los caracteres "OK" corresponden a una conexión exitosa, caso contrario aparecerá "ERROR".

<CR><LF> ERROR <CR><LF>

PREFIJO COMANDO SUFIJO

• El prefijo y sufijo consta de <CR> (Retorno de carro) y <LF> (Salto de línea).

- Existen comandos que deben escribirse por la puerta serial. Por ejemplo, para iniciar el envío de un mensaje al usuario, se escribe:

  AT+CGMS="número del destinatario"

- El envío de un comando siempre implica una respuesta del módem, que debe leerse de la puerta serial

- Existen comandos que deben leerse de la puerta serial, los que indican que un mensaje ha llegado, por ejemplo:

  AT+CMTI="SM", 1

  - Configuración de aviso sobre nuevo mensaje

AT+CNMI= 1,2,0,0,0

Cuando una aplicación recibe una notificación, debe enviar un comando AT para leer el contenido del mensaje. Es posible, sin embargo, configurar el módem para que envíe el contenido del mensaje junto con la notificación. (Manualslib, 2008)

Comandos AT para envío de SMS

Para envío de un SMS se requiere los siguientes comandos:

AT + CMGF = 1 Para dar formato SMS como mensaje de texto

Para enviar un SMS, el comando AT que se utiliza es AT + CMGS

AT + CMGS = "+ xxxx" <Intro>

Donde "+ xxxx" es el número de móvil al cual se enviará. (blogElectronica.com, 2008) (LUCKYSTAR'S OPINIONS, 2009)

2.1.1.5.2. Comandos AT más utilizados.

- Comandos para información del equipo

AT+CMGI: Información el fabricante.

AT+CGNS: Obtener # de serie.

AT+CPAS: Leer estado del modem.

- Comandos de servicio de red

AT+CSQ: Obtener calidad de la señal

AT+COPS: Selección de un operador.

AT+WOPN: Lee nombre de operadora.

- Comandos para SMS

AT+CMGF: Selecciona el formato de los mensajes de texto.

AT+CMGR: Lee un mensaje de texto almacenado.

AT+CMGL: Lista los mensajes almacenados.

AT+CMGS: Envía un SMS.

AT+CMGW: Almacena mensaje en la memoria interna.

AT+CMSS: Envía un mensaje almacenado.

(Manualslib, 2008, pág. 147)

2.2. MARCO CONCEPTUAL DE LOS ELEMENTOS UTILIZADOS EN EL PROYECTO.

Aquí se encuentra las características técnicas y descripciones de los dispositivos electrónicos usados en el proyecto modelo.

2.2.1. ESPECIFICACIONES DEL NFC SHIELD V1.0

NFC Shied es un dispositivo de comunicación NFC diseñado con el circuito integrado NXP PN532 fusionado con una tecnología que permite la comunicación en un radio de corto alcance. Este dispositivo tiene fácil conectividad con el shield de Arduino UNO y diseñada para acoplarse con la placa del mismo. (Enlace de ARDUINO - Seeedstudio wiki, 2014)

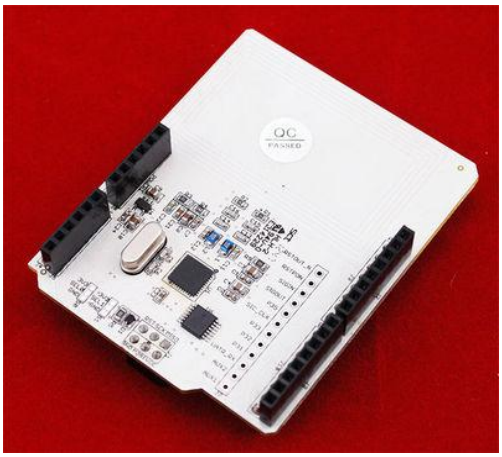


Figura 2.7 Shield NFC V1.0
Fuente: (Enlace de ARDUINO - Seeedstudio wiki, 2014)

2.2.1.1 Características

- Compatible con Shield Arduino (No se requiere soldadura)
- Socket para conectar con otros escudos.
- Cuenta Interfaz SPI ,por lo tanto la mayoría de los pines de Arduino están disponibles para otras aplicaciones
- Construido con antena PCB.
- El rango máximo de comunicación del NFC es de unos 5 cm.
- capaz de leer / escribir chip Mifare tag's
- Voltaje 5v – min4.3v max 5,7v
- Corriente 90mA – min 80mA max 100mA

  (Enlace de ARDUINO - Seeedstudio wiki, 2014)

2.2.2. Especificaciones del Arduino UNO

Es una placa con un microcontrolador diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares este dispositivo consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. (ARDUINO, 2013)

Figura 2.8 Placa Arduino UNO
Fuente: (ARDUINO, 2013)

2.2.2.1. Arduino

Esta es una plataforma electrónica libre que ayuda a la implementación de prototipos de dispositivos electrónicos los cuales se puede modificar en su software y hardware.

Las Shield de Arduino se puede encontrar ensamblada de fábrica y según los costos y el riesgo se la encuentra también en partes para ensamblarla personalmente. Esta placa de Arduino maneja un lenguaje de comunicación con el usuario fundamentado en Wirin y el entorno de desarrollo es IDE que se fundamenta en Processing. (ARDUINO, 2013)

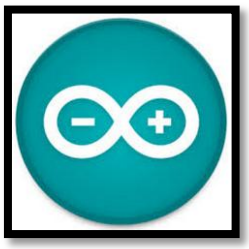


Figura 2.9 Símbolo de Arduino
Fuente: (ARDUINO, 2013)

2.2.2.2. Placa Arduino UNO

Es una versión de placa Arduino que permite una comunicación segura contra variaciones de voltaje que puede producirse por el puerto USB y así afectar a su Pc.

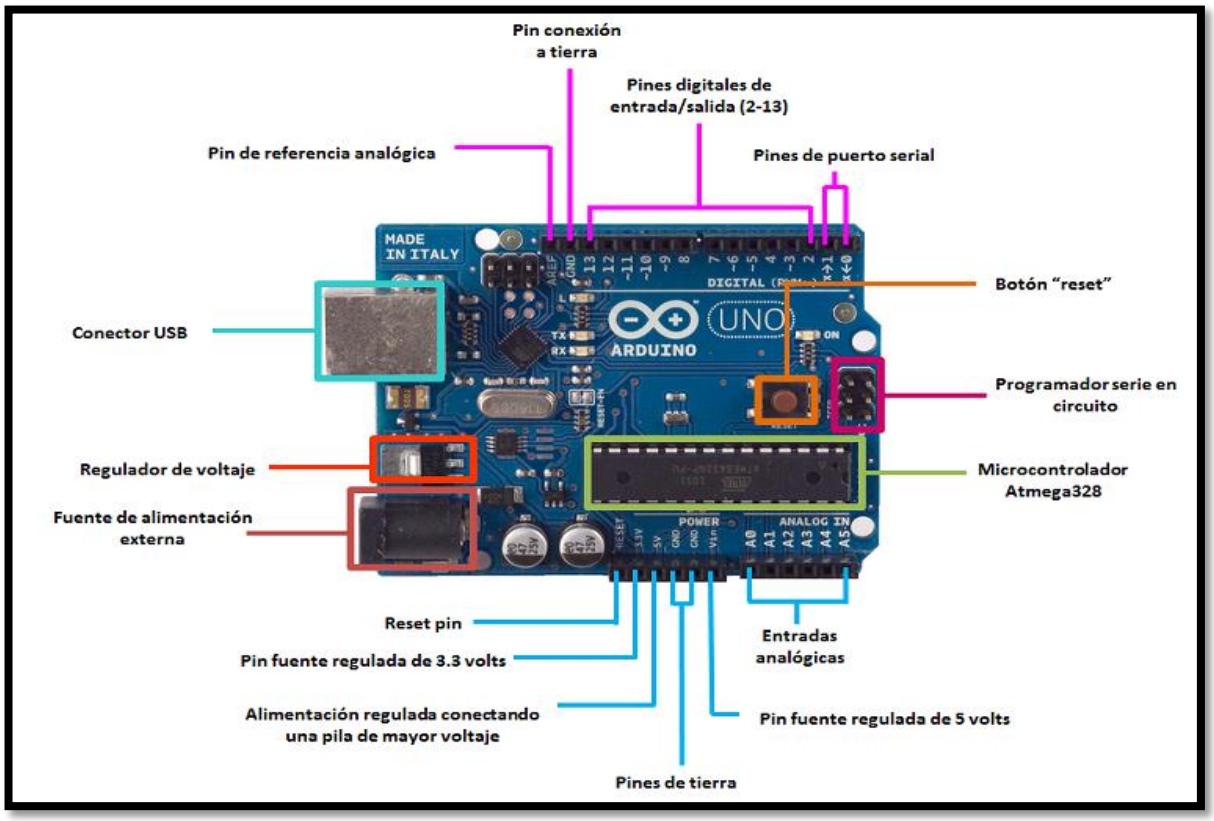


Figura 2.10 Entradas y salidas de la placa Arduino UNO
Fuente: (BLAM electronics)

2.2.2.2.1. Características de la placa Arduino UNO

La placa electrónica de Arduino UNO consta de:

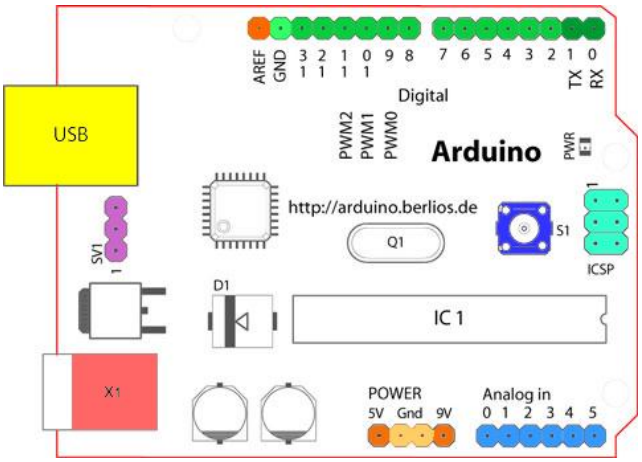


Figura 2.11 Elementos Importantes de la placa Arduino UNO
Fuente: (ARDUINO, 2013) Disabling Auto Reset On Serial Connection

Características de los terminales de Arduino Uno

· De 2-13 Terminales digitales de color verde
· De 0-5 Terminales analógicas de color azul claro (ANALOGIN)
· Terminal de color naranja es de referencia analógica (AREF)
· Terminal de tierra digital verde claro (GND)
· Terminales digitales 0-1 / E/S serie – TX/RX color verde oscuro
· Botón de reset de azul oscuro (S1)
· Terminales de comunicación serial de celeste (ICSP)
· Terminales de alimentación externa (9-12 VDC) de color rosa (X1)
· Terminales de alimentación y tierra de azul claro (POWER)

Tabla 3. Terminales de Arduino Uno
Fuente: (ARDUINO, 2013)

ATmega 328

Arduino Uno presenta un microcontrolador (IC1) ATmega 328 le permite utilizar el chip con el entorno Arduino en los proyectos, sin necesidad de utilizar la placa. Es una forma de darle un formato más profesional a los proyectos con arduino.

- Terminales de E/S digital 23 (6 de ellos pueden proporcionar salidas PWM)
- Terminales de entrada analógicos (ADC) 6 canales de 10 bits
- 20MHz de velocidad
- 3 temporizadores
- Memoria Flash 32 KB
- RAM 2 KB
- EEPROM 512 bytes

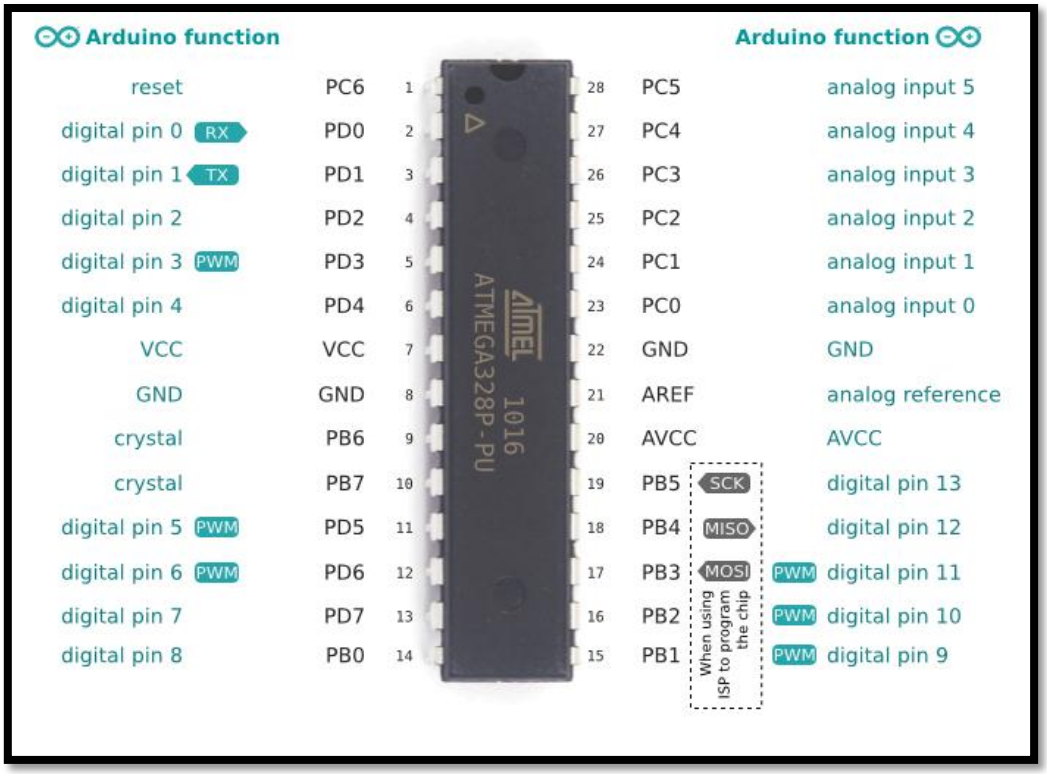


Figura 2.12 Pines del I.C. ATMEGA328
Fuente: (Comparduino electronics y openhardware, 2014)

2.2.3. TAG NFC MIFARE

Es una tarjeta de tecnología NFC en forma de llavero y compatible con todos los dispositivos de NFC y celulares con este tipo de tecnología

Figura 2.13 Smart tag's Mifare
Fuente: (NXP(Philips), 2007)

2.2.3.1. Características de las tag NFC Mifare:

- Memoria de 1KB (768bytes)
- Compatible con S.O. Android
- Compatible con Shield NFC y Arduino
- Dimensiones: 40mm x 32mm x4mm
- Material plástico PVC resistente al agua

(NXP(Philips), 2007)

2.2.4. ESPECIFICACIONES DEL MÓDULO GSM

El modem GSM utilizado pertenece al fabricante ZTE, modelo MG3006 debido a que se acopla perfectamente a la placa electrónica Arduino Uno y a la plataforma de programación del mismo. Este modelo consta de antena integrada fácil de manipular, es pequeño y potente. El modelo del modem GSM ZTE-MG3006 encapsulado se puede observar en la Figura 2.14.

Figura 2.14 Modem GSM, ZTE-MG3006
(ZTE, 2007)

Las características principales de este equipo son:

- Quad-band 850/900/1800/1900 MHz
- Tecnología GSM
- Acceso a través de comandos AT
- Socket para tarjeta SIM
- Rango de alimentación de voltaje: 7V – 25V
- Puerto serial RS232 (BB9)
- Dimensiones:75mm*50mm*16mm
- Peso < 200G
- Rango de temperatura de operación: -30°C – 75°C

  (E-Lins Technology)

2.2.5. Software Arduino

Es una plataforma electrónica libre de fácil uso q permite programar y facilitar la comunicación con los dispositivos Arduino. Es compatible con sistemas operativos en Windows, Linux y Android. (ARDUINO, 2013)

Figura 2.15 Ícono del programa Arduino
Fuente: Fernando Orbe

Este software es de fácil uso y programación pero para agregar ciertos dispositivos extras se requieren ciertas librerías para darle una mejor funcionalidad a la aplicación de Arduino. (ARDUINO, 2013)

El software Arduino se le visualiza en la Figura 2.16:

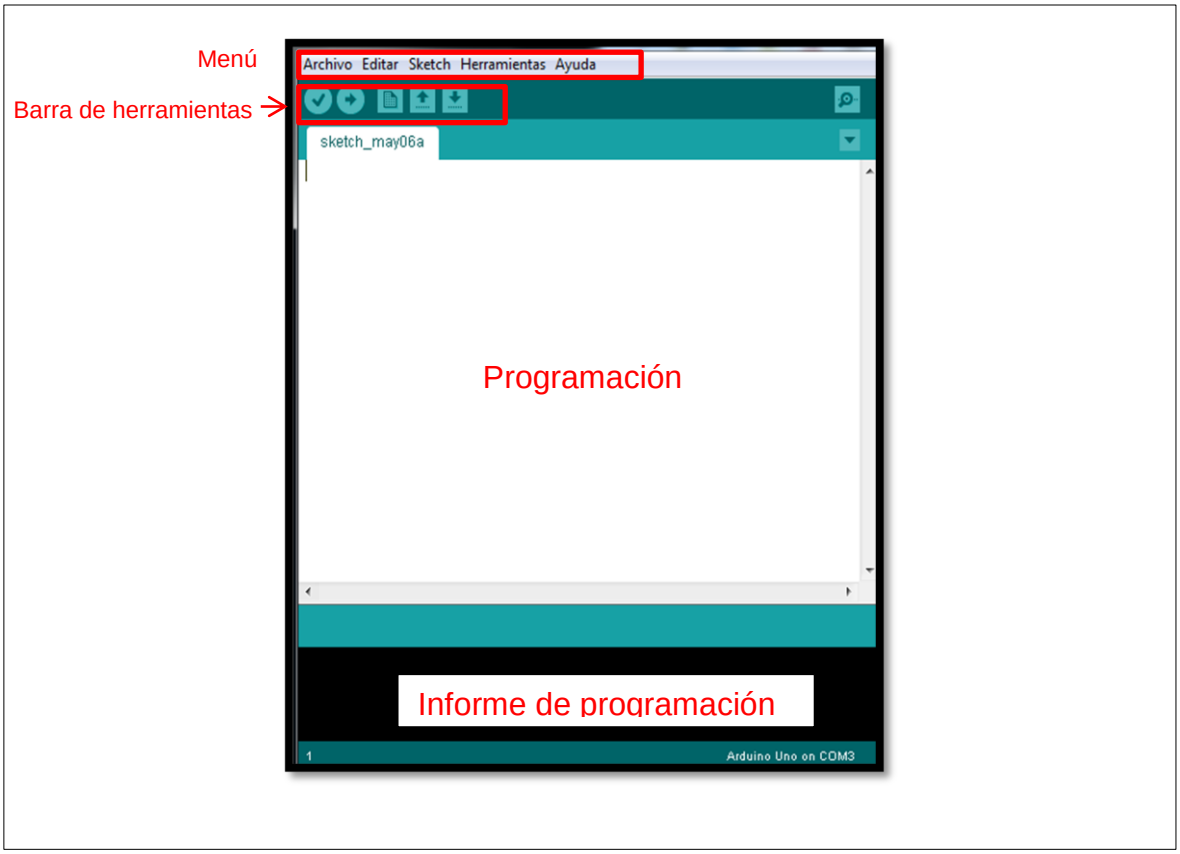


Figura 2.16 Partes de la interface Arduino
Fuente: Fernando Orbe

El sofware de arduino presenta un espacio para realizar la programacion para o se determina las caracteristicas que el usuario desea que su dispositivo arduino realice. Encontamos tambien un espacio de informe de programacion el mismo que indica si algo no se encuentra bien dentro de la programacion o funcionamiento del sofware.

En la barra de Menú se cambia las caracteristicas de uso del software, también se encuentra librerías y ejemplos basicos para uso de los dispositivos Arduino.

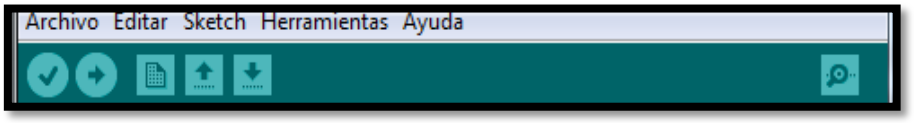


Figura 2.17 Barra de Menú del programa Arduino
Fuente: Fernando Orbe

En la barra de herramientas tenemos en un orden de izquierda a derecha:

Verificar: Sirve para revisar si existe errores en la programación

Cargar: Sirve para enviar los datos hacia la memoria del dispositivo

Nuevo: Abre una nueva página de programación

Abrir: Abre un documento ya realizado

Guardar: Guarda la programación dentro de un documento formato (.ino)

También se encuentra un ícono dentro del software llamado (monitor serial), la cual se observa en la Figura 2.18, este ícono aparte de monitorear la comunicación de la PC con el dispositivo, también permite interactuar con el mismo y manejar sus condiciones de funcionamiento desde el computador.

Figura 2.18 Ícono del monitor serial
Fuente: Fernando Orbe

2.2.4.1. Librería NFC para Arduino <PN532.h>

Para el uso de dispositivos NFC con el software Arduino, se requiere librerías dependiendo del tipo de dispositivo NFC como por ejemplo el NFC Shield V1.0 el mismo que requiere de la librería <PN532.h> para que funcione con Arduino UNO e identifique las Tag Mifare. (Fernando Orbe, 2014)

En esta librería vienen ciertos ejemplos y sublibrerías que ayudan a la lectura, escritura de las tag como se puede ver en la Figura 2.19.

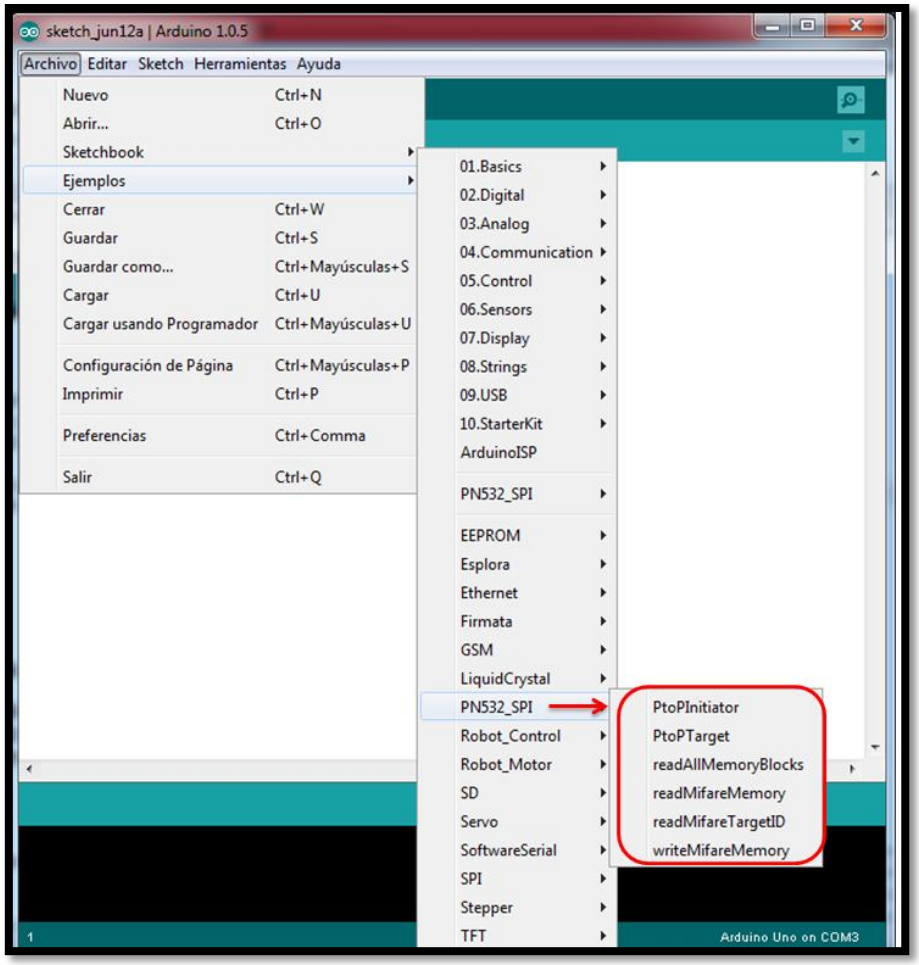


Figura 2.19 Librería <PN532_SPI> para NFC
Autor: Fernando Orbe

Entre los ejemplos que brinda la librería de NFC, un ejemplo completo de lectura es el de “readAllMemoryBlocks” el mismo que permite visualizar la ID, cantidad del tag's y datos de la misma, además presenta de forma ordenada los bloques de memoria dentro de las Tag Mifare como se puede observar en la Figura 2.20 se ordenan en 64 bloques numerados cada bloque tiene 16 bytes. También es importante mencionar que en la línea de comando que contiene una llave de seguridad para dar paso a las tag nuevas (KEY_A) o usadas (KEY_B). Esta debe ser cambiada según el tipo de tag se requiera leer sus bloques caso contrario los mismos no podrán ser visualizados.

nfc.authenticateBlock(1, id ,3,KEY_A,keys);

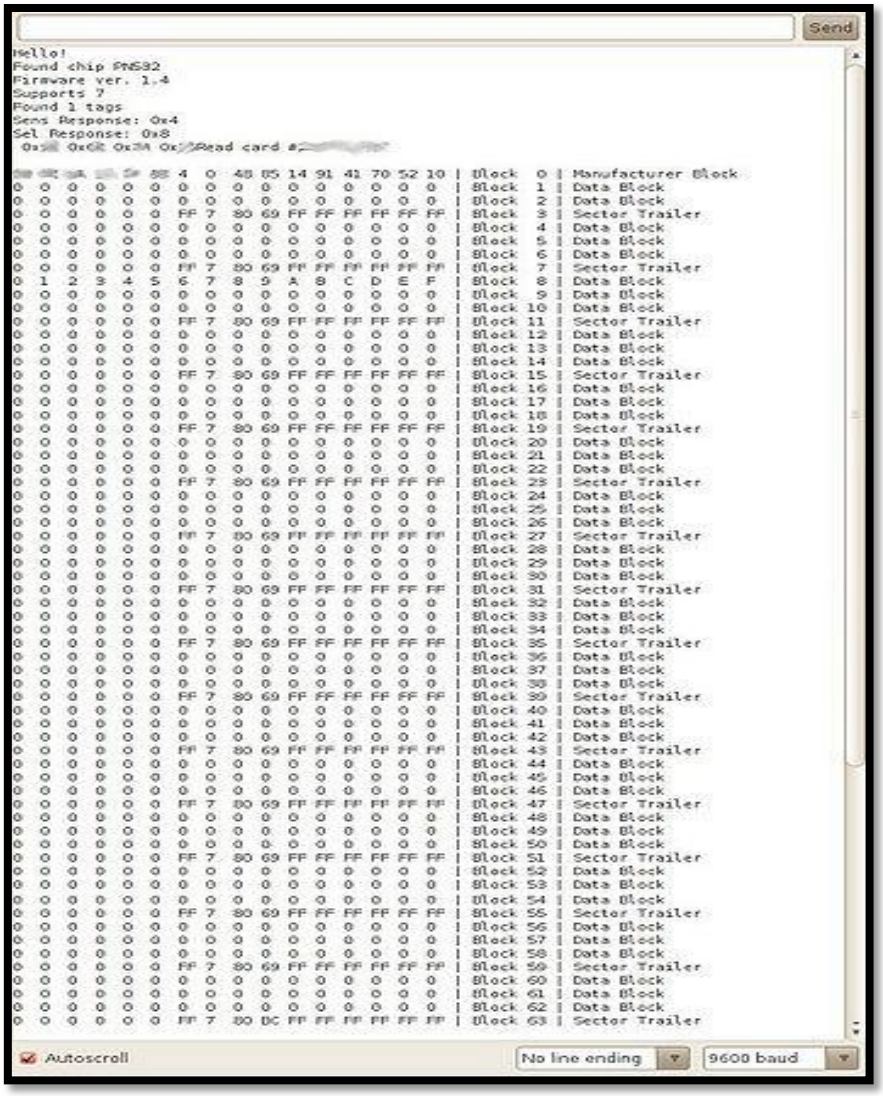


Figura 2.20 Bloques de memoria de las tag Mifare
Fuente: (Enlace de ARDUINO - Seeedstudio wiki, 2014)

También se encuentra entre los ejemplos de la librería un programa base e importante que permite la escritura en las memorias tag Mifare. Este programa se lo encuentra con el nombre de "writeMifareMemory".

Este programa está diseñado para identificar los 64 bloques de memorias de las tag Mifare y guardar información dentro de las mismas, para ello con la ayuda de la librería tiene una línea de comando donde se puede escoger el bloque en el cual se requiere guardar datos. (Enlace de ARDUINO - Seeedstudio wiki, 2014)

nfc.writeMemoryBlock(1,0x08,writeBuffer);

Como se observa en el ejemplo de la línea de comando indica que los datos se guardaran en el bloque de memoria 8 (0x008). Es importante mencionar que los bloques de memoria se dividen en tres partes que son: (Enlace de ARDUINO - Seeedstudio wiki, 2014)

Manufacturer Block : Solo para lectura, en este bloque se guarda la ID

Data Block : Estos son grabables o escritura de datos

Sector trailer: Sirve para autenticación y acceso bits de del sector

(Enlace de ARDUINO - Seeedstudio wiki, 2014)

CAPÍTULO 3

ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE UN MODELO DE CONTROL PARA MEJORAR LA SEGURIDAD EN EL SERVICIO DE TRANSPORTES DE TAXI USANDO TECNOLOGÍA NFC

En este capítulo se encuentra el diseño, montaje e implementación tanto en hardware, software y también funcionamiento modelo del dispositivo prototipo NFC.

Para el diseño de este prototipo y funcionamiento del mismo se toma en cuenta los siguientes requerimientos

- Una fuente de poder o regulación de voltaje que proporcione 5Vdc
- Una fuente de poder interna para mantener activo el dispositivo en caso de desconexión de la fuente principal.
- Activación de mensajes del dispositivo GSM
- Un circuito de comunicación serial 232 que permita la comunicación entre los dispositivos de Arduino y GSM
- Un computador que contenga el software Arduino para la programación de los dispositivos.

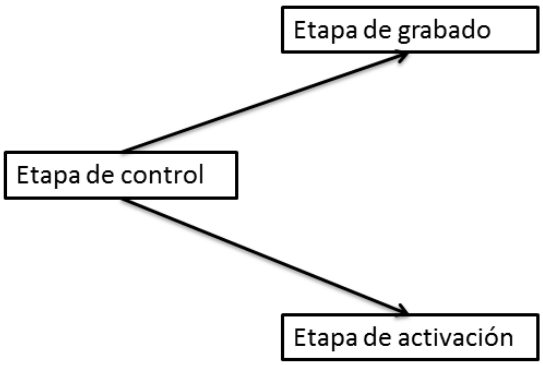
Con los requerimientos solicitados listos, para el diseño del proyecto este se divide en dos partes que funcionaran conjuntamente

- Grabación de tag's
- Envío de información

Para el funcionamiento de estas dos partes mencionadas se requiere desarrollar el hardware y software de dispositivo

3.1 ANÁLISIS Y DISEÑO DEL HARDWARE APLICADO PARA LA COMUNICACIÓN Y FUNCIONAMIENTO DE DISPOSITVOS DE NFC Y GSM CONJUNTAMENTE.

En el diseño de hardware se encuentran los dispositivos electrónicos que serán usados en el diseño del proyecto. Dentro del diseño de hardware encontramos la etapa de control que cumple con dos etapas diferentes una de grabado y otra etapa diferente que es la de activación.



Mapa de etapas de funcionamiento
Fuente: Fernando Orbe

3.1.1. ETAPA DE CONTROL

La etapa de control es una de las principales etapas para la el funcionamiento del dispositivo ya que en la misma están todos los elementos electrónicos los mismos que actúan en conjunto para cumplir con el objetivo

Dentro de la implementación se usó dispositivos que trabajan con hardware y software libre por lo que después de varias investigaciones existen varios dispositivos NFC en el mercado sin embargo el uso de los dispositivos shield de Arduino UNO y NFC son de fácil uso, programación y sobre todo tiene fácil comunicación con otros dispositivos requeridos en el objetivo del proyecto.

En la Figura 3.1 y 3.2 encontramos los diagramas circuitales de los dispositivos de Arduino UNO y NFC Shield PN-532 utilizados en el proyecto.

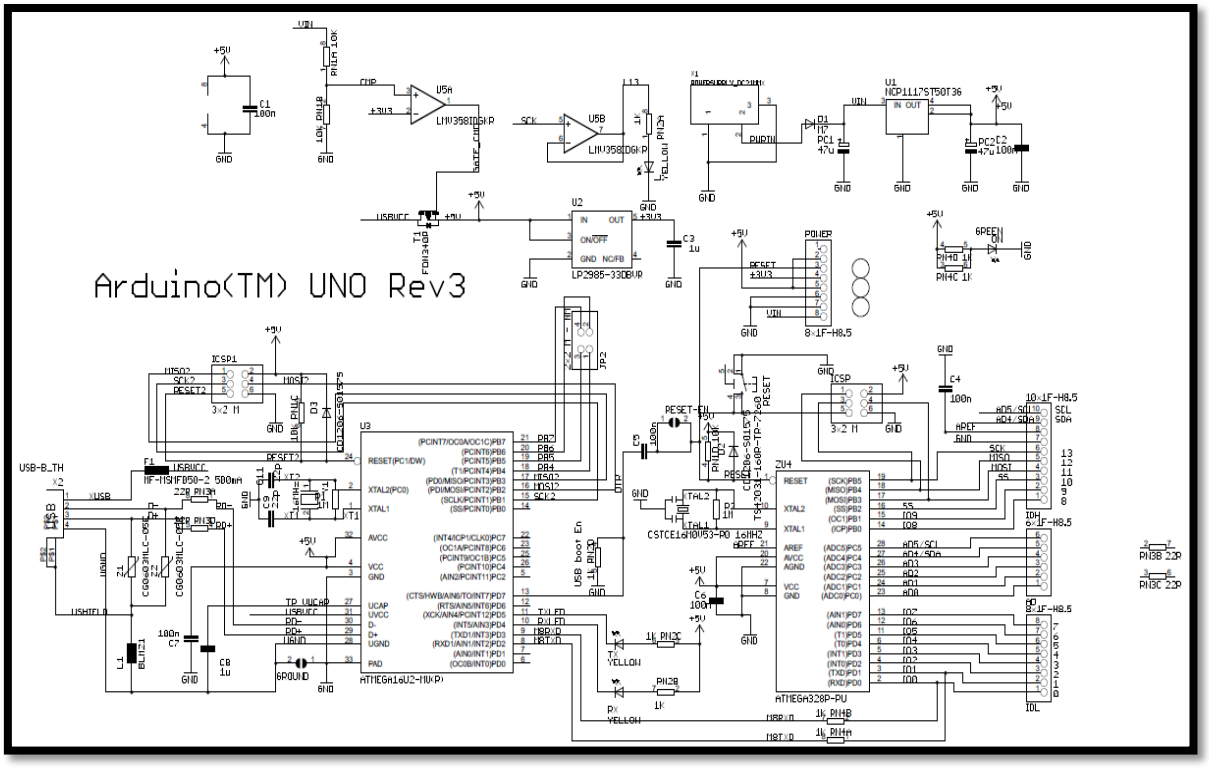


Figura 3.1 Diagrama circuital de Arduino UNO
Fuente: (ARDUINO, 2013)

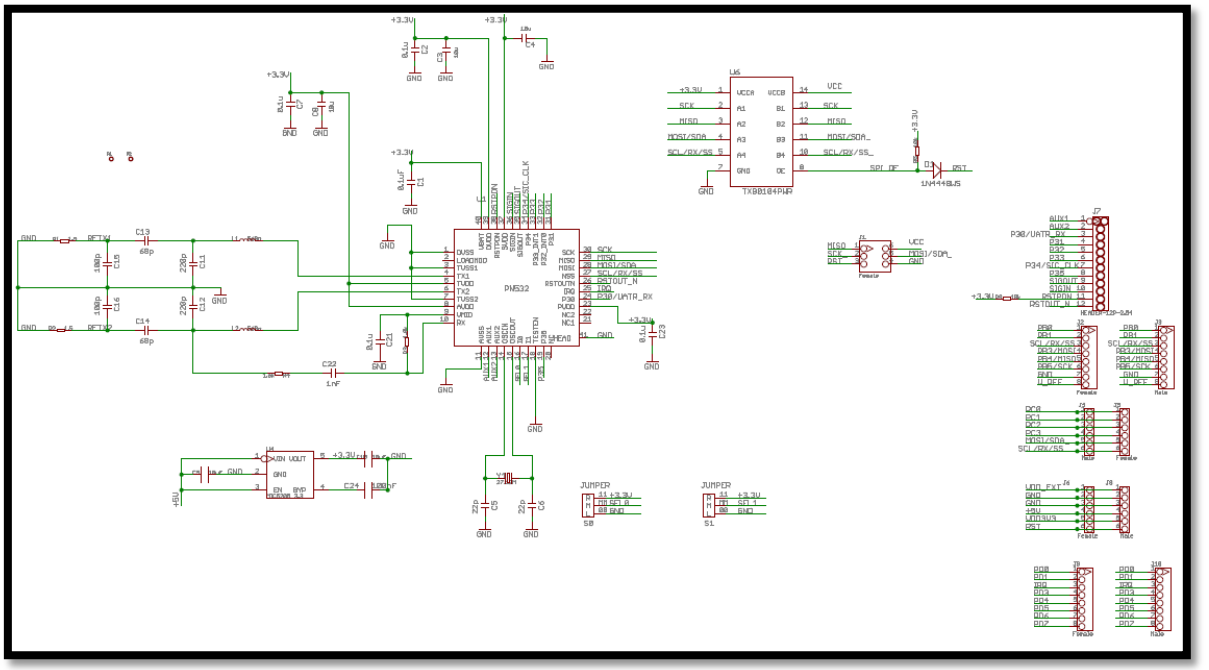


Figura 3.2 Diagrama circuital de NFC Shield V1.0
Fuente: (Enlace de ARDUINO - Seeedstudio wiki, 2014)

3.1.1.1. Configuración de pines de Arduino UNO

La configuración de los pines de utilizados son:

Pin A0, A2 y A4, Estos pines son conectados a 3 de led's indicadores de funcionamiento.

Pin 1 y 2, En estos pines se encuentra las señales de RX y TX del NFC Shield

Pin 8 y 9, En estos pines se encuentra las señales de RX y TX serial pala comunicación con el modem GSM.

Pin 10, 11, 12 y 13, Estos pines están distribuidos de la siguiente manera para comunicación SPI; (D10 – CS), (D11 – MOSI), (D12 – MISO), (D13-SCK). (ARDUINO, 2013) (Enlace de ARDUINO - Seeedstudio wiki, 2014)

También en la etapa de control se encuentra el NFC Shield el cual se basa en el integrado PN532. Esta placa está diseñada para acoplarse con los pines de comunicación de la placa de Arduino UNO. Para mayor facilidad, NFC Shield PN-532 es compatible con la distribución de pines de Arduino UNO por lo que únicamente se acoplan las placas como se observa en el la Figura 3.3.

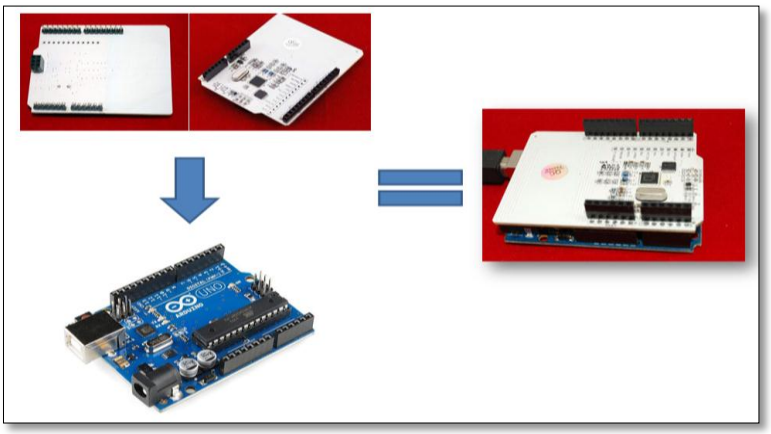


Figura 3.3 Arduino UNO + NFC Shield
Fuente: Fernando Orbe

3.1.2. ETAPA DE VERIFICACIÓN Y GRAVADO DE TAG´S

Dentro de las especificaciones de dispositivos que se requieren para la implementación del sistema de seguridad en taxis, las tag's son principalmente las que el usuario de las unidades de transporte debe conseguir para poder almacenar en estas los números de teléfono solicitados.

Para el prototipo se utiliza las tag´s Mifare, las mismas que siendo compatibles con NFC PN-532 y por sus características técnicas permiten lectura y escritura de datos en su memoria.

Para el sistema de seguridad en los taxis se requiere una persona asignada únicamente para la tarea de grabar los dos números de teléfono que solicite el usuario que sean escritos dentro de su tag.

Esta etapa se la debe realizar mediante un computador el mismo que debe tener instalado la aplicación de Arduino. El puerto USB es el medio por el cual interactúa la PC con los dispositivos de NFC Shield y Arduino UNO.

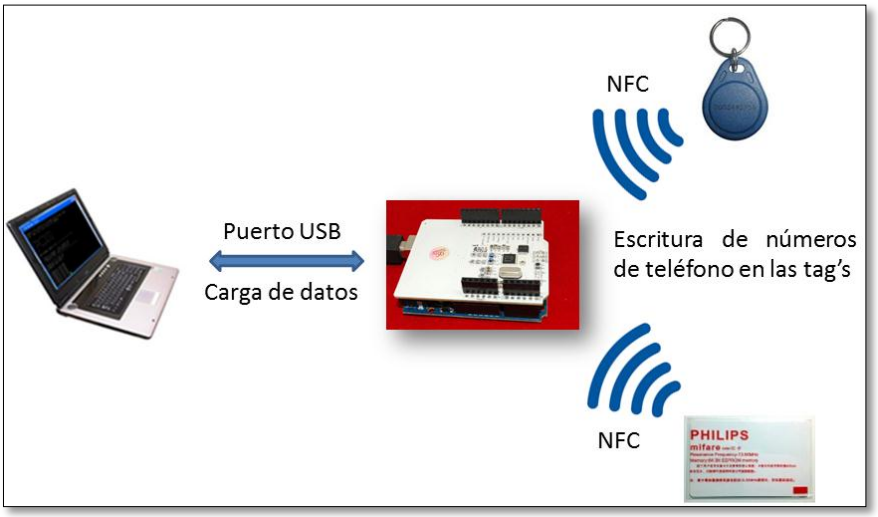


Figura 3.4 Presentación gráfica para escritura de tag
Fuente: Fernando Orbe

Una vez que se establece la comunicación entre los dispositivos de Arduino y la PC, se procede a cargar el programa el mismo que está diseñado para leer y escribir datos sobre las memorias de las tag´s NFC.

El programa cargado está diseñado para poder interactuar por medio del monitor serial de la aplicación Arduino donde se ingresará manualmente los dos números diferentes de teléfonos que luego serán escritos y guardados en las memorias tag NFC al momento de acercarlos. Este procedimiento de gravado en las tag`s es explicado con más detalle en la etapa de software.

3.1.3. ETAPA DE ACTIVACIÓN:

La etapa de activación es una etapa independiente a la de gravado ya que para la misma se requiere la instalación de unidad de dispositivo de seguridad NFC para cada taxi.

Figura 3.5 Dispositivo de seguridad NFC
Fuente: Fernando Orbe

Para la instalación de este dispositivo se requiere una fuente de voltaje mínima 8v a máxima de 16v la misma que la podemos obtener dentro del mismo auto ya que todos tienen baterías de 12VDC

En esta etapa se requiere cargar el programa diseñado para comunicar el Arduino UNO y NFC Shield con el modem GSM.

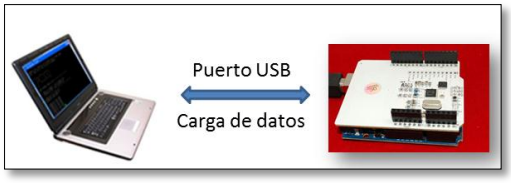


Figura 3.6 Comunicación entre PC y Arduino UNO
Fuente: Fernando Orbe

En el momento en que una tag programada con los dos números de teléfono la Shield NFC la detecta, lee si es compatible, revisa los datos de la memoria y envía los datos (números de teléfono) de la misma por medio de la comunicación serial RS-232 que conecta con el modem GSM.

Figura 3.7 Elementos del dispositivo de seguridad NFC
Fuente: Fernando Orbe

El modem GSM para comunicarse con los dispositivos Arduino Y NFC requiere una únicamente de tres entradas RX, TX y GND que presenta el conector DB9. Como se muestra en la Figura 3.8

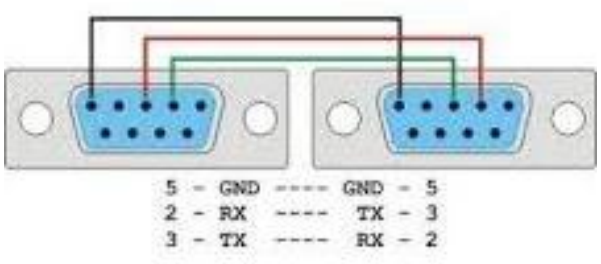


Figura 3.8 Configuración RS-232 cruzada
Fuente: (WordPress Arduino, 2012)

Esta comunicación serial RS-232 tiene la característica de utilizar los pines 8 y 9 del Arduino UNO donde representa RX y TX respectivamente

Para la comunicación entre las placas de Arduino UNO y modem GSM se implementó una pequeña placa para facilitar la conexión serial como se muestra en la Figura 3.9

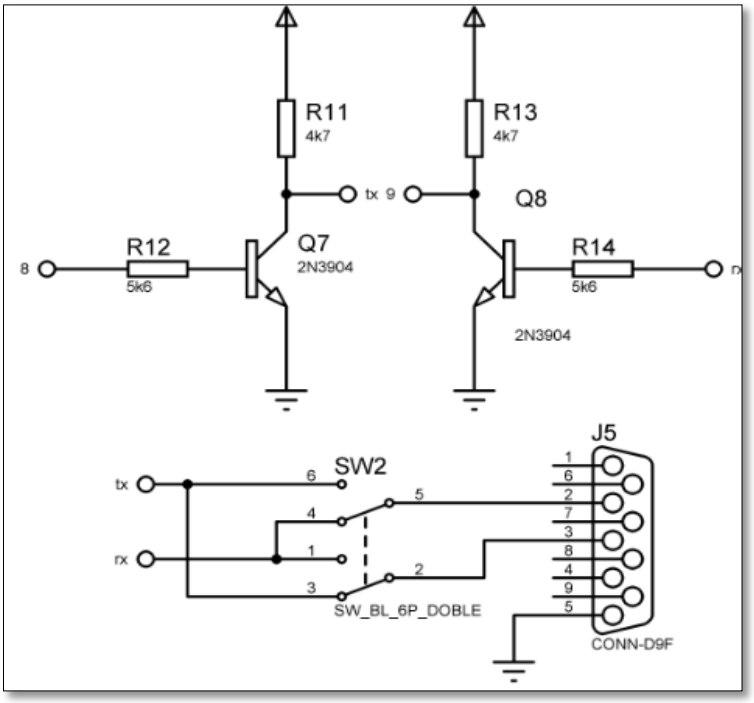


Figura 3.9 Diagrama circuital de la placa de comunicación serial
Fuente: (WordPress Arduino, 2012)

En el diagrama circuital de la Figura 3.9 se encuentran un SW2 que servirá para cruzar la comunicación del RS232 en caso se lo requiera, además Q7 y Q8 ayudan amplificando la señales recibidas y enviadas y de esta manera podremos garantizar el correcto envío de datos, además se agregaron 3 led de indicadores que van directamente a las entradas de A0, A2 y A4 del Arduino Uno.

Al final de esta etapa el modem GSM envía un mensaje contenido en la memoria del Arduino a los números registrados en las tag's, como se muestra en el ejemplo de la Figura 3.10.

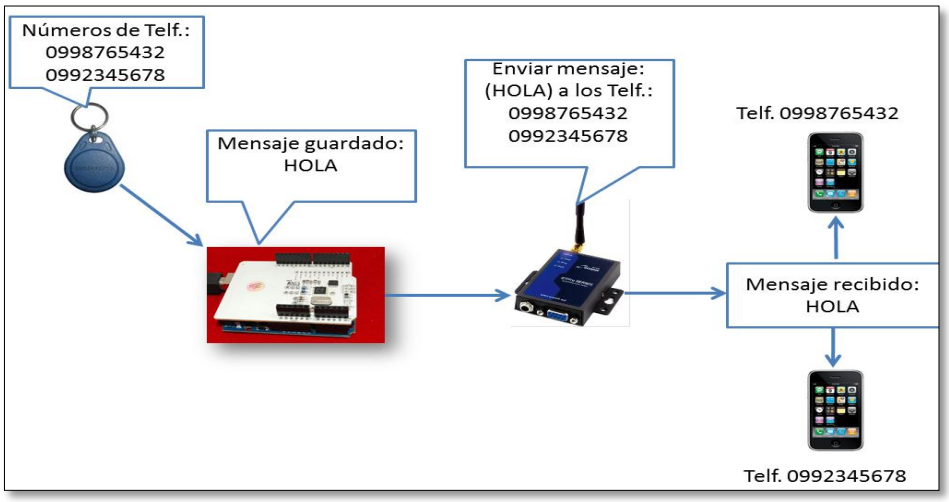


Figura 3.10 Ejemplo de funcionamiento de los elementos del dispositivo de seguridad.
Autor: Fernando Orbe

3.2. ESTUDIO, DISEÑO Y PROGRAMACION DE SOFTWARE DE ACUERDO A LOS REQUERIMIENTOS DEL DISPOSITIVO DE SEGURIDAD NFC.

3.2.1. DIAGRAMA DE FLUJO DEL DISEÑO DE SOFTWARE ARDUINO UNO Y NFC SHIELD PARA GRAVADO Y ESCRITURA DE DATOS

El diseño de software de esta etapa es el que permite la escritura sobre la memoria de la tag, y de esta manera grabar los números de teléfono. Se puede observar en la Figura 3.11 el diagrama de flujo del procedimiento de esta etapa.

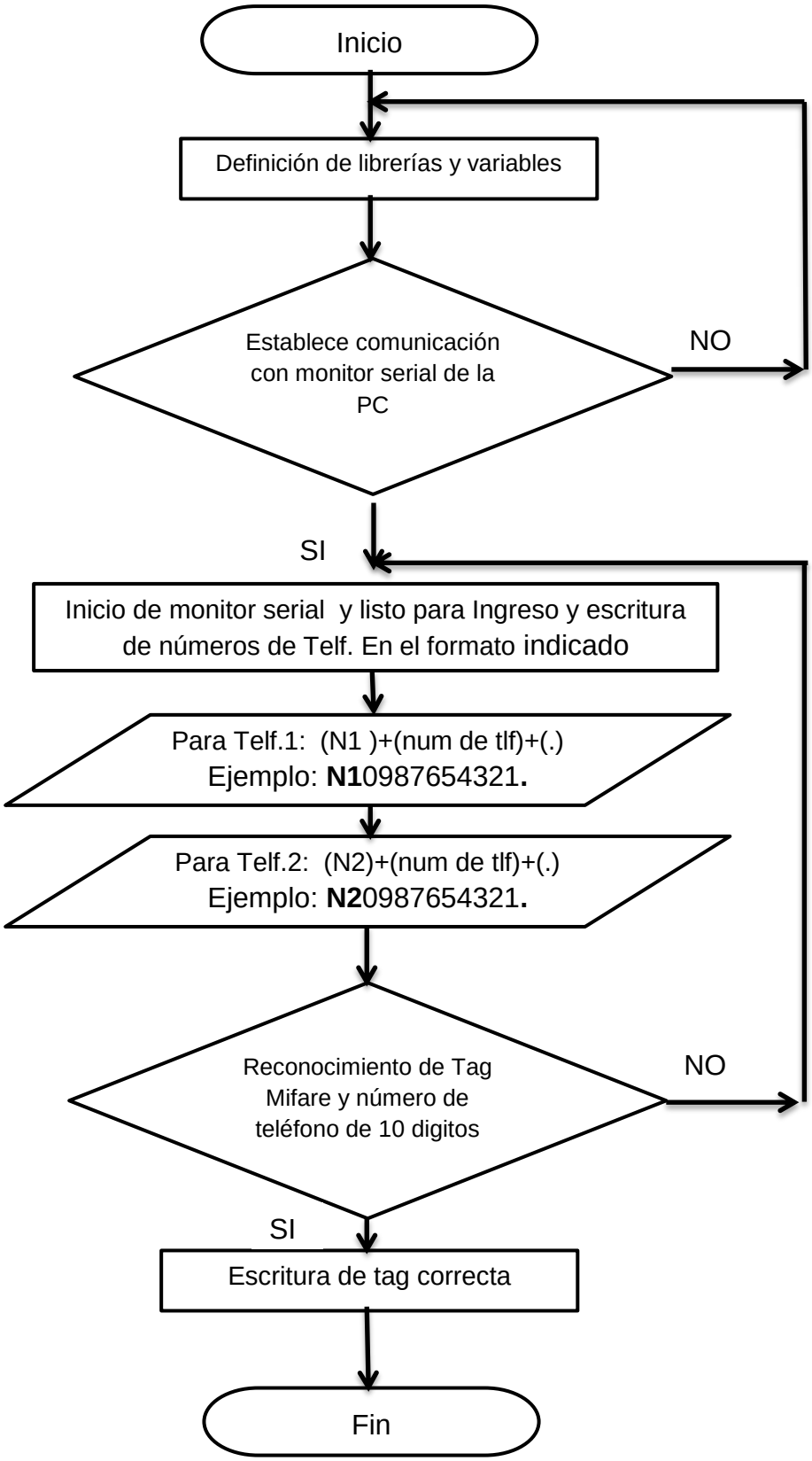


Figura 3.11 Diagrama de flujo de Escritura y gravado de datos.
Fuente: Fernando Orbe

3.2.2. DIAGRAMA DE FLUJO DEL DISEÑO DE SOFTWARE ARDUINO UNO Y NFC SHIELD PARA LECTURA Y ENVÍO DE MENSAJES POR EL MODEM GSM

Para esta etapa se debe primeramente cargar el programa diseñado para el funcionamiento del dispositivo de seguridad NFC dentro de la memoria de Arduino UNO.

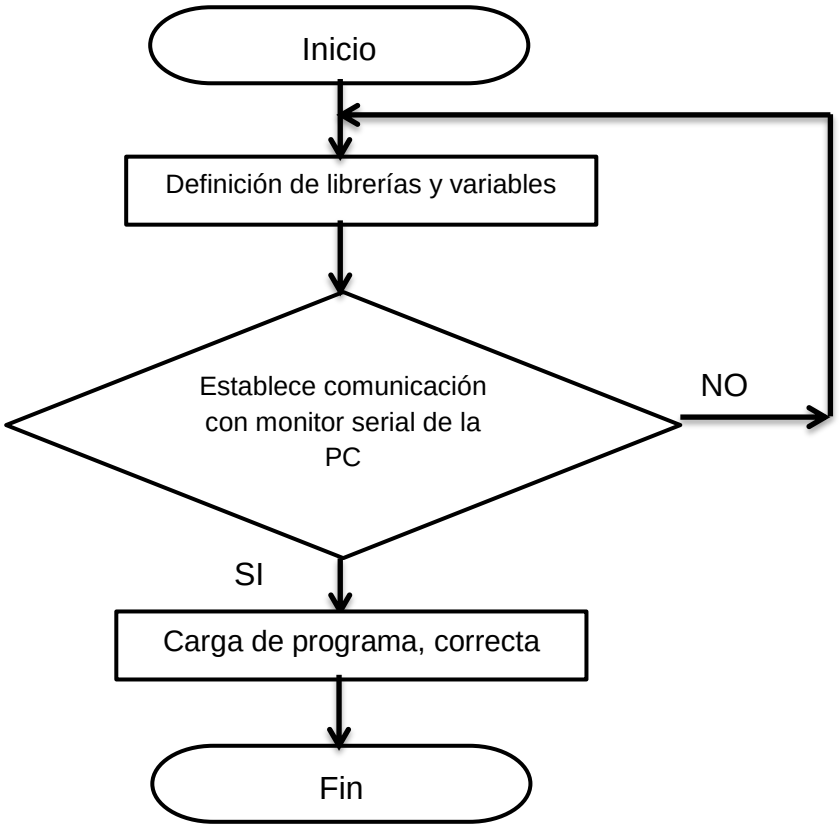


Figura 3.14 Diagrama de flujo de carga del programa para envío de SMS
Fuente: Fernando Orbe

Una vez cargado el programa dentro de la memoria Arduino UNO se establece la conexión serial con el modem GSM. El modem GSM envía mensajes en el momento en que el Arduino UNO junto al NFC Shield detectan cerca una tag previamente configurada con 2 números de teléfono en los espacios de memoria establecidos. En la Figura 3.15 se presenta el diagrama de flujo de envío de SMS.

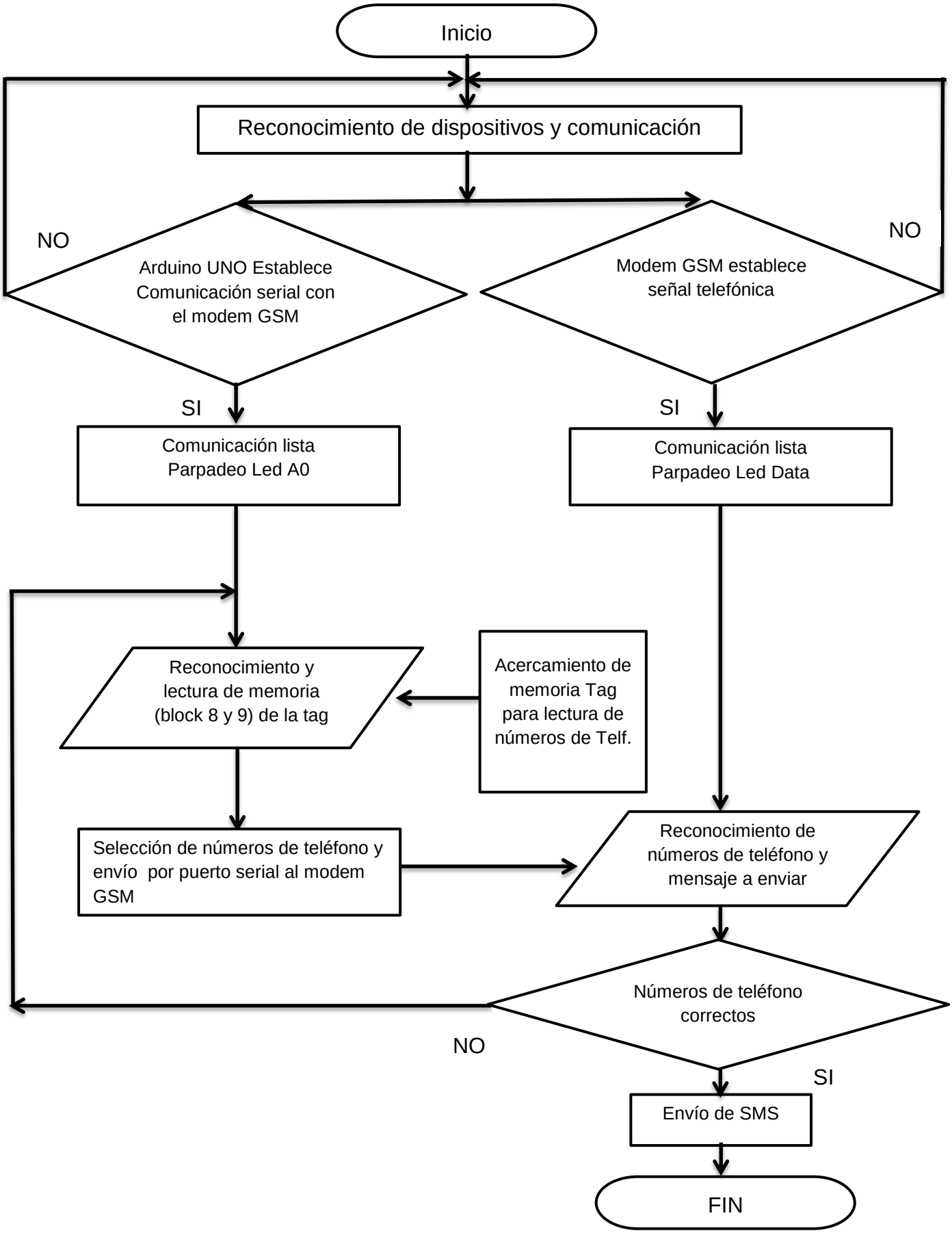


Figura 3.15 Diagrama de flujo de Lectura y envío de SMS
Fuente: Fernando Orbe

3.3. ACOPLAMIENTO DE LOS DISPOSITIVOS Y PROGRAMACIÓN.

3.3.1 MONTAJE DE DISPOSITIVOS

Como ya se había mencionado en las características de Arduino y NFC Shield, estas placas están diseñadas para facilitar el montaje y manejo entre ellas únicamente con las librerías para Arduino y NFC.

Para la comunicación serial RS232 con el modem GSM, se requiere de una placa para facilitar la conexión con el conector DB9 como se muestra en la Figura 3.16

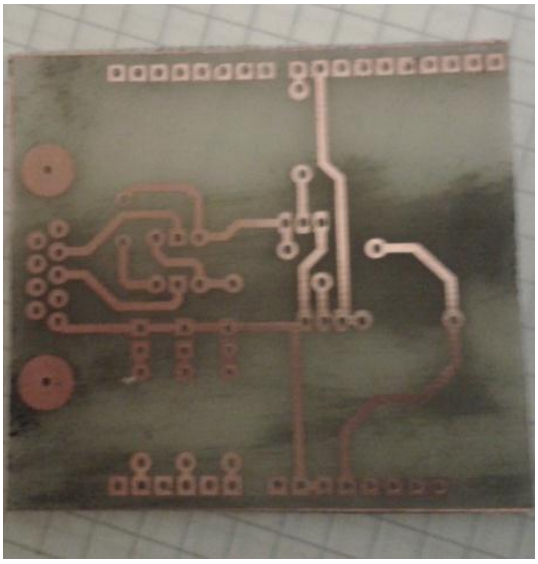
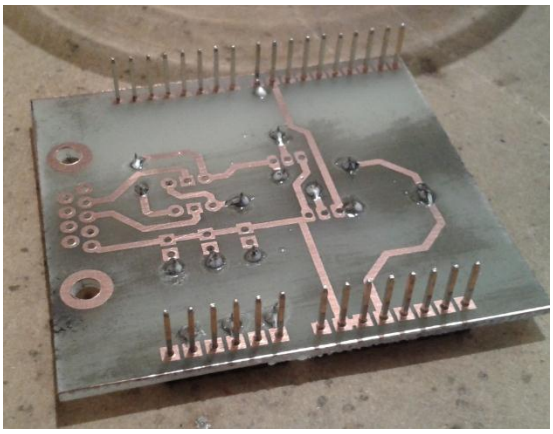
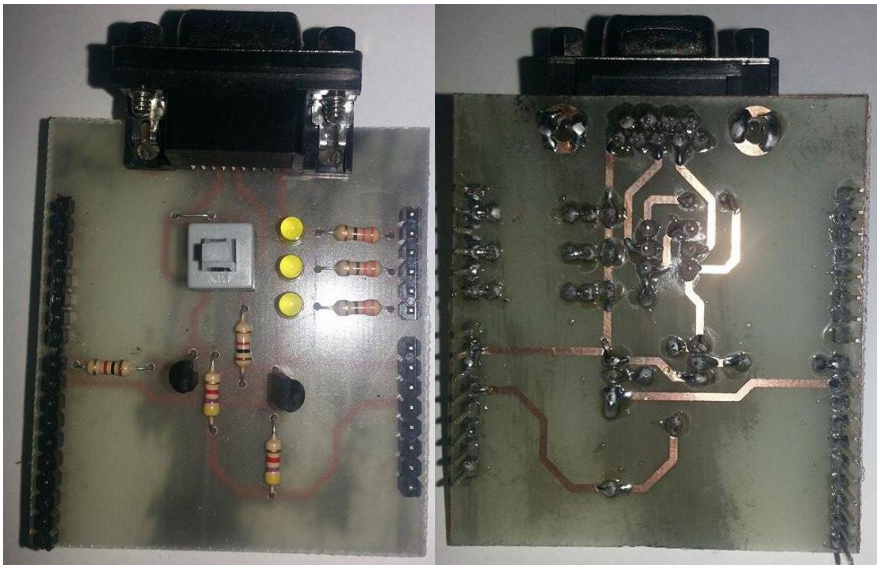


Figura 3.16 Placa de COM RS-232
Autor: Fernando Orbe

Con la impresión de pistas en la placa lista, se agregan los elementos en la misma y nos queda como podemos observar en las Figuras 3.17 (a,b,c)



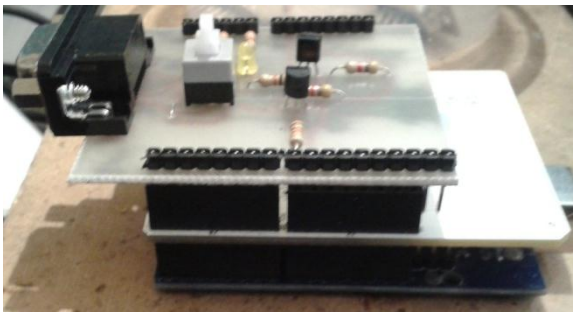
(a)



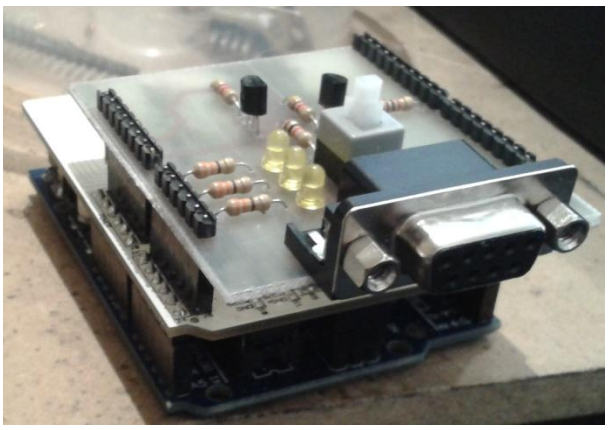
(b) **(c)**

Figura 3.17 Diseño de placa de COM RS-232
Fuente: Fernando Orbe

Por lo tanto la placa de comunicación serial es acoplable con RS-232 y el montaje de los dispositivos NFC y Arduino UNO quedaría como indica la Figura 3.18. (a,b)



(a)



(b)

Figura 3.18 Montaje de dispositivos NFC, Arduino UNO y COM RS-232.
Fuente: Fernando Orbe

Una vez terminado el montaje de las placas de Arduino UNO, NFC Shield y la placa de comunicación serial se procede al montaje en la caja diseñada para el dispositivo de seguridad NFC

3.3.1. 1. Diseño y montaje de hardware

Inicialmente se realiza el diseño estándar de un plano indicando las medidas de las medidas y características que se requiere para que puedan ser colocados los dispositivos electrónicos.

Como se puede observar en la Figura 3.19 se presentan tres vistas del diseño, características y medidas de la caja como son: vista frontal (VF), vistan superior (VS), y vista lateral (VL).

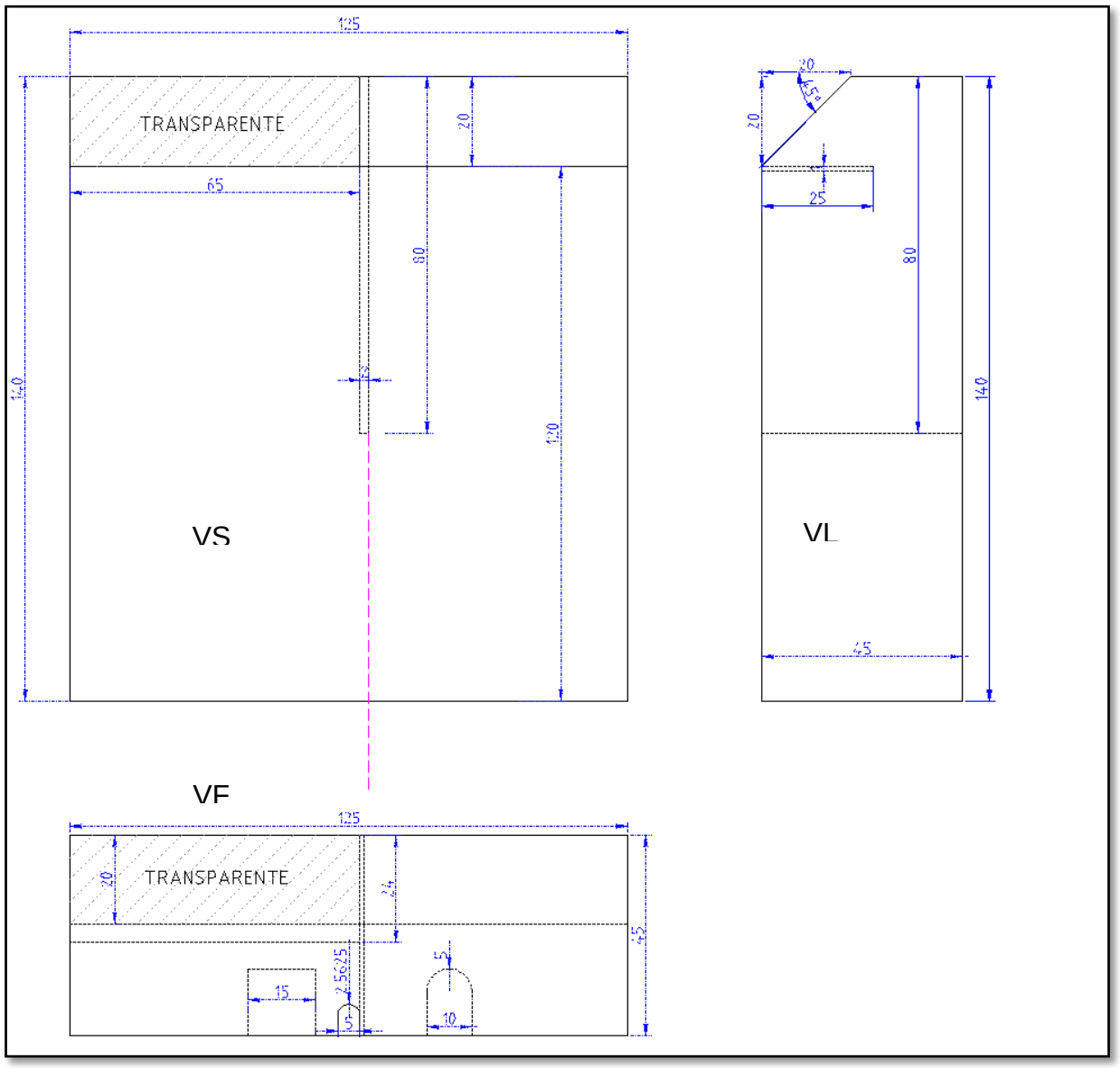


Figura 3.19 Plano de la caja para el dispositivo de seguridad NFC
Fuente: Fernando Orbe

Una vez determinadas las medidas de la caja, el modelo quedaría como se muestra en la Figura 3.20, que tiene una vista de la caja en una posición de 45º, donde la línea completa representa una vista frontal y la línea entre cortada representa lo que está detrás o internamente de la caja.

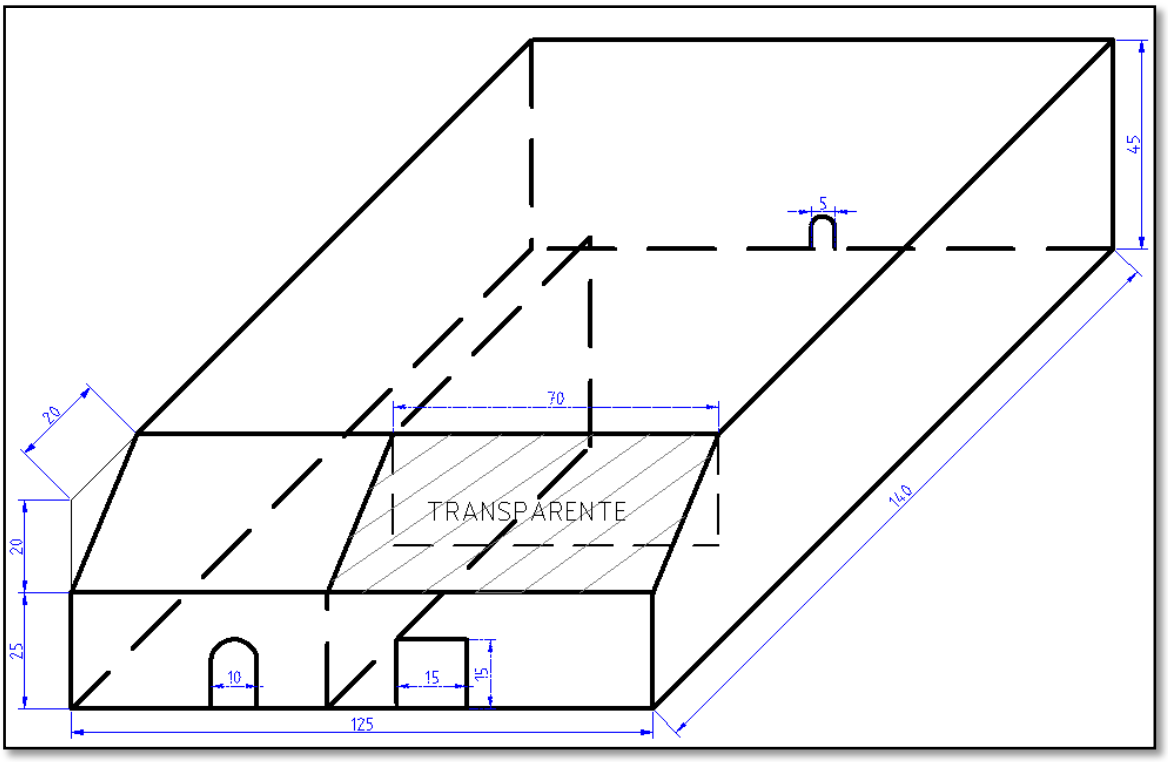


Figura 3.20 Vista a 45º de inclinación del plano del diseño de la caja.
Fuente: Fernando Orbe

Para el diseño del dispositivo se considera una caja que lo contenga y proteja de cualquier mala manipulación de estos. Las características de la caja se presentan en la Figura 3.21. (a) base ,(b) cubierta

(a) Base de la caja **(b) Cubierta de la caja**

Figura 3.21 Elaboración de la caja
Fuente: Fernando Orbe

La caja del dispositivo está diseñada con acrílico tomando en cuenta las especificaciones y dimensiones que se acomode acorde al tamaño de los dispositivos, conexiones y requerimientos de funcionamiento

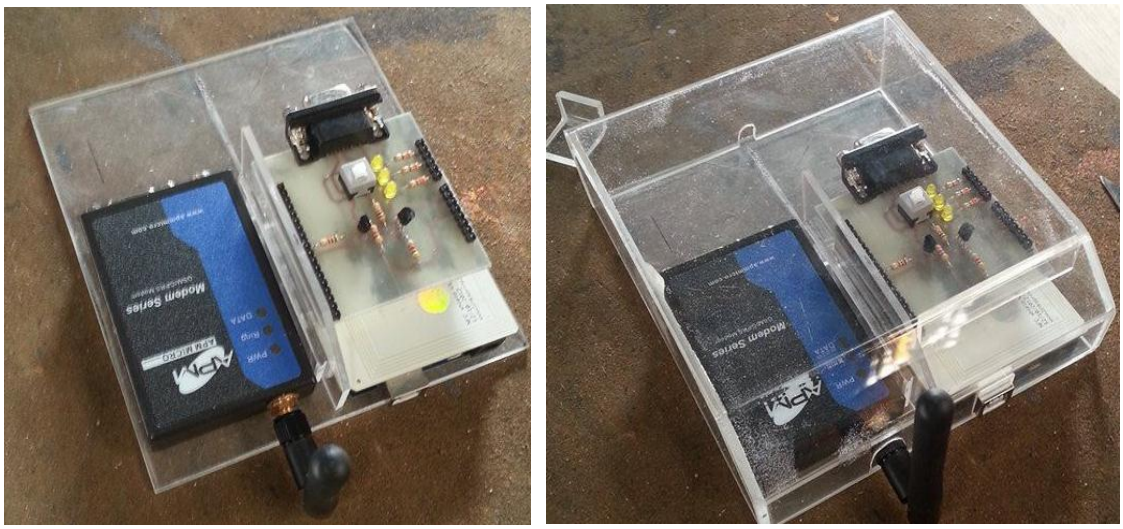


Figura 3.22 Montaje de elementos en la caja.
Fuente: Fernando Orbe

Al terminar el montaje de los dispositivos dentro de la caja podemos observar cómo queda el dispositivo terminado interna y externamente

Figura 3.23 Vista interna del montaje
Fuente: Fernando Orbe

Figura 3.24 Vista externa del montaje forrado en color negro
Fuente: Fernando Orbe

3.3.2. DISEÑO Y MONTAJE DE SOFTWARE

De igual manera como en la etapa de hardware, también la etapa de software comprende dos etapas que son de programación de Software de Lectura y escritura de datos en la tag y la otra etapa que es de activación y envío de SMS a los números de teléfono asignados en la tag.

Como se había comentado en las características de implementación del proyecto se requiere la aplicación de Arduino instalada en su PC para con la misma asignar la programación para los dispositivos. También se requiere para esto descargar la librería del escudo Shield NFC la cual se llama <PN532.h> que la puede encontrar en la página oficial de Arduino.

Para la programación necesaria para el cumplimiento del objetivo de funcionamiento del dispositivo de seguridad NFC se utiliza; C++, Java y comandos AT.

3.3.2.1. Programaciones diseñadas para el dispositivo de seguridad NFC

3.3.2.1.1. Programación de lectura y escritura de datos en la tag

Para esta programación se requiere las librerías de <PN532.h>, ya que dentro de ella se encuentra pequeñas librerías de lectura y escritura de las tag Mifare en los bloques de memoria programados para guardar los números de teléfono que en este caso se asignaron los bloques 8 y 9, además <SPI.h> también nos permite iniciar una comunicación externa en este caso con el Shield NFC.

ESCRITURA_DE_TAG

```
#include <PN532.h>      // Librería Shield NFC
#include <SPI.h>      // Librería Comunicación con otros dispositivos

#include <stdlib.h> //
#define Envio     A0

#define SCK 13
#define MOSI 11
#define SS 10
#define MISO 12
PN532 nfc(SCK, MISO, MOSI, SS);

uint8_t written=0;
uint32_t id;

String NumeroA = "";
String NumeroB = "";

int contA;
int contB;

boolean state;
String inputString = "";
boolean stringComplete=false;

int timer1_counter;
int contin=0;
#define ledPin A0

String a,b,c,d,e,f,g,h,i,j;
uint8_t a1,b1,c1,d1,e1,f1,g1,h1,i1,j1,dato;
```

```
void setup(void)
{
          Serial.begin(9600);
          inputString.reserve(200);
          Serial.println("Inicializando...");
           nfc.begin();
           uint32_t versiondata = nfc.getFirmwareVersion();
           if (!versiondata)
                    {
                     Serial.print("Didn't find PN53x board");
                    while (1); // halt
                    }
         Serial.print("Found chip PN5");
         Serial.println((versiondata>>24) & 0xFF, HEX);
         Serial.print("Firmware ver. ");
         Serial.print((versiondata>>16) & 0xFF, DEC);
         Serial.print('.');
         Serial.println((versiondata>>8) & 0xFF, DEC);
         Serial.print("Supports ");
         Serial.println(versiondata & 0xFF, HEX);
         nfc.SAMConfig();
         Serial.println(" ");
         Serial.println(" ");
         Serial.println("Agregue (N1 o N2)+(num de tlf)+(.) ");
         Serial.println(" ");
         Serial.println("  ejemplo: N10987654321. ");
         pinMode(Envio,    OUTPUT);

         digitalWrite(Envio,    0);
         }


void loop(void)
         {

          if (stringComplete)
          {
           RDWR();
         }
}


void serialEvent() {

          while (Serial.available()) {
         char inChar = (char)Serial.read();
          inputString += inChar;

          if (inChar == '.') {
```

```
                if(inputString.length() == 13)
                {
                Serial.println("RESULTADOS DE ESCRITURA");
                Serial.println(inputString.length());
                Serial.println(inputString);


                 }
                 else
                        {

                        stringComplete = false;
                         }

                stringComplete = true;


                }
        }
}

void escritura1()
        {
        Serial.println("Numero 1");
         id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);
          if (id != 0)
          {
                Serial.println();
                Serial.print("Read card #");
                Serial.println(id);
                Serial.println();
                uint8_t keys[]= {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
                uint8_t writeBuffer[10];

                writeBuffer[0]=a1;
                writeBuffer[1]=b1;
               writeBuffer[2]=c1;
               writeBuffer[3]=d1;
               writeBuffer[4]=e1;
               writeBuffer[5]=f1;
               writeBuffer[6]=g1;
               writeBuffer[7]=h1;
               writeBuffer[8]=i1;
               writeBuffer[9]=j1;

                 if(nfc.authenticateBlock(1, id ,0x08,KEY_B,keys)) //authenticate block
0x08
                        {
```

```
                        if(written == 0) //Not written
                        {
                        written = nfc.writeMemoryBlock(1,0x08,writeBuffer); // Write
                       writeBuffer[] to block 0x08

                        if(written)
                                {
                                  Serial.println("Write 1 Successful");
                                 }
                        }
                }
          }
}




void escritura2()
{
 Serial.println("Numero 2");
 id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);
         if (id != 0)
          {
                Serial.println();
                  Serial.print("Read card #");
                  Serial.println(id);
                 Serial.println();
                 uint8_t keys[]= {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
                 uint8_t writeBuffer[10];

                  writeBuffer[0]=a1;
                  writeBuffer[1]=b1;
                  writeBuffer[2]=c1;
                 writeBuffer[3]=d1;
                 writeBuffer[4]=e1;
                 writeBuffer[5]=f1;
                writeBuffer[6]=g1;
                 writeBuffer[7]=h1;
                writeBuffer[8]=i1;
                writeBuffer[9]=j1;

                 if(nfc.authenticateBlock(1, id ,0x09,KEY_B,keys)) //authenticate block
0x08
                 {

                 if(written == 0) //Not written
                {
          written = nfc.writeMemoryBlock(1,0x09,writeBuffer); // Write writeBuffer[] to
block 0x08
```

```
        if(written)
                    {
          Serial.println("Write 2 Successful");
        }
      }
    }
  }
}

void RDWR()
{

    inputString.trim();
    if(inputString.substring(0,2)=="N1")
    {

      String aux=inputString.substring(0,2);
      NumeroA=inputString.substring(2,12);

a=inputString.substring(2,3);b=inputString.substring(3,4);c=inputString.substring(4,5);d
=inputString.substring(5,6);e=inputString.substring(6,7);

f=inputString.substring(7,8);g=inputString.substring(8,9);h=inputString.substring(9,10);
i=inputString.substring(10,11);j=inputString.substring(11,12);
      a.trim();b.trim();c.trim();d.trim();e.trim();f.trim();g.trim();h.trim();i.trim();j.trim();

a1=conversion(a);b1=conversion(b);c1=conversion(c);d1=conversion(d);e1=conversion
(e);

f1=conversion(f);g1=conversion(g);h1=conversion(h);i1=conversion(i);j1=conversion(j
);
      aux.trim();
      NumeroA.trim();
      Serial.println(aux);
      Serial.println(NumeroA);

Serial.println(a1);Serial.println(b1);Serial.println(c1);Serial.println(d1);Serial.println(e1
);

Serial.println(f1);Serial.println(g1);Serial.println(h1);Serial.println(i1);Serial.println(j1);
      escritura1();
      delay(1000);
      asm("jmp 0x0000");

    }

    if(inputString.substring(0,2)=="N2")

    {
```

```
      String aux=inputString.substring(0,2);
      NumeroB=inputString.substring(2,12);

a=inputString.substring(2,3);b=inputString.substring(3,4);c=inputString.substring(4,5);d
=inputString.substring(5,6);e=inputString.substring(6,7);

f=inputString.substring(7,8);g=inputString.substring(8,9);h=inputString.substring(9,10);
i=inputString.substring(10,11);j=inputString.substring(11,12);
      a.trim();b.trim();c.trim();d.trim();e.trim();f.trim();g.trim();h.trim();i.trim();j.trim();

a1=conversion(a);b1=conversion(b);c1=conversion(c);d1=conversion(d);e1=conversion
(e);

f1=conversion(f);g1=conversion(g);h1=conversion(h);i1=conversion(i);j1=conversion(j
);
      aux.trim();
      NumeroB.trim();
      Serial.println(aux);
      Serial.println(NumeroB);

Serial.println(a1);Serial.println(b1);Serial.println(c1);Serial.println(d1);Serial.println(e1
);

Serial.println(f1);Serial.println(g1);Serial.println(h1);Serial.println(i1);Serial.println(j1);
      escritura2();
      delay(1000);
      asm("jmp 0x0000");
    }


    Serial.println(inputString);

     inputString = "";
     stringComplete = false;

}

int conversion(String var)
{
  if(var=="0"){dato=0;return(dato);}
  if(var=="1"){dato=1;return(dato);}
  if(var=="2"){dato=2;return(dato);}
  if(var=="3"){dato=3;return(dato);}
  if(var=="4"){dato=4;return(dato);}
  if(var=="5"){dato=5;return(dato);}
  if(var=="6"){dato=6;return(dato);}
  if(var=="7"){dato=7;return(dato);}
  if(var=="8"){dato=8;return(dato);}
  if(var=="9"){dato=9;return(dato);}
}
```

3.3.2.2.2. Programación de Lectura y envío de SMS

La programación de esta etapa requiere de las librerías de NFC Shield y SPI ya mencionadas y a estas se agrega la librería de comunicación serial.

#include <SoftwareSerial.h> // Librería de comunicación Serial

SoftwareSerial mySerial(9, 8); // RX, TX

La declaración de pines 8 y 9 para RX y TX respectivamente sirve para que no exista una confusión de comunicación TX/RX de los pines 0 y 1 que están ocupados por la comunicación entre Arduino UNO Y NFC Shield

Para esta programación se requiere los comandos AT enviados por la comunicación serial la cual está conectada al modem GSM.

LECTURA_Y_ENVÍO_DE_SMS

```
#include <PN532.h>              //Librería Shield NFC
#include <SPI.h>                // Librería Comunicación con otros dispositivos
#include <SoftwareSerial.h>     // Librería de comunicación Serial

SoftwareSerial mySerial(9, 8); // RX, TX

#define Envio     A0
#define Lectura   A2
#define Escritura A4

#define SCK 13
#define MOSI 11
#define SS 10
#define MISO 12

PN532 nfc(SCK, MISO, MOSI, SS);
uint8_t written=0;

uint32_t id;

String NumeroA = "";
String NumeroB = "";

String smsA="Unidad de taxi 1234 con codigo 33 ";
```

```
String smsB="Cooperativa Comite del Puebo";

boolean state;
String inputString = "";
boolean stringComplete = false;

int timer1_counter;

#define ledPin A0

void setup(void)
{
  noInterrupts();           // disable all interrupts
  TCCR1A = 0;
  TCCR1B = 0;


  TCNT1 = timer1_counter;   // preload timer
  TCCR1B |= (1 << CS12);    // 256 prescaler
  TIMSK1 |= (1 << TOIE1);   // enable timer overflow interrupt
  interrupts();


  Serial.begin(9600);
  inputString.reserve(200);
  mySerial.begin(4800);

  Serial.println("Inicializando...");
  nfc.begin();
  uint32_t versiondata = nfc.getFirmwareVersion();
  if (!versiondata)
  {
    Serial.print("Didn't find PN53x board");
    while (1); // halt
  }
  Serial.print("Found chip PN5");
  Serial.println((versiondata>>24) & 0xFF, HEX);
  Serial.print("Firmware ver. ");
  Serial.print((versiondata>>16) & 0xFF, DEC);
  Serial.print('.');
  Serial.println((versiondata>>8) & 0xFF, DEC);
  Serial.print("Supports ");
  Serial.println(versiondata & 0xFF, HEX);
  nfc.SAMConfig();
  initGSM();
  pinMode(Envio,    OUTPUT);

}

ISR(TIMER1_OVF_vect)        // interrupt service routine
```

```
{
  TCNT1 = timer1_counter;   // preload timer
  digitalWrite(ledPin, digitalRead(ledPin) ^ 1);
}

void loop(void)
{
  lectura1();
  lectura2();
  delay(1000);
  if(NumeroA!="" && NumeroB!="")
  {
    envioSMS1();
    envioSMS2();
    delay(20000);
    delay(20000);
    delay(20000);
    Serial.println("siguiente-A");
    Serial.println("siguiente-B");
    NumeroA="";
    NumeroB="";
  }

  while(mySerial.available())
  {
    Serial.write(mySerial.read());
  }
  if (stringComplete)
  {
    RDWR();
  }


}


void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    inputString += inChar;
    if (inChar == '.') {
      stringComplete = true;
    }
  }
}

void lectura1()
{
id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);
  if (id != 0)
```

```
  {
    Serial.println();
    Serial.print("Read card #");
    Serial.println(id);
    Serial.println();
    uint8_t keys[]= {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

    if(nfc.authenticateBlock(1, id ,0x08,KEY_B,keys)) //authenticate block 0x08
    {

      uint8_t block[10];
      //read memory block 0x08
      if(nfc.readMemoryBlock(1,0x08,block))
      {
        Serial.println("Read block 0x08:");
        //if read operation is successful
        for(uint8_t i=0;i<10;i++)
        {
          //print memory block

          NumeroA += (int)block[i];
          Serial.print(block[i],HEX);
          Serial.print(" ");
        }
        Serial.println();
        NumeroA.trim();
        NumeroA=NumeroA.substring(0,10);
        Serial.println(NumeroA);
        Serial.println();

      }

    }
  }

}

void lectura2()
{
id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);
  if (id != 0)
  {
    Serial.println();
    Serial.print("Read card #");
    Serial.println(id);
    Serial.println();
    uint8_t keys[]= {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};

    if(nfc.authenticateBlock(1, id ,0x09,KEY_B,keys)) //authenticate block 0x08
    {
```

```
      uint8_t block[10];
      //read memory block 0x08
      if(nfc.readMemoryBlock(1,0x09,block))
      {
        Serial.println("Read block 0x09:");
        //if read operation is successful
        for(uint8_t i=0;i<10;i++)
        {
          //print memory block
          NumeroB += (int)block[i];
          Serial.print(block[i],HEX);
          Serial.print(" ");
        }
        Serial.println();
        NumeroB.trim();
        NumeroB=NumeroB.substring(0,10);
        Serial.println(NumeroB);
        Serial.println();

      }

    }
  }

}

void RDWR()
{

    inputString.trim();
    if(inputString=="N1")
    {
     lectura1();
    }

    if(inputString=="N2")
    {
     lectura2();
    }

    Serial.println(inputString);

     inputString = "";
     stringComplete = false;

}

void initGSM()
{
```

```
   delay(20000);
   mySerial.println("AT");
   Serial.println("AT");
   delay(5000);
   mySerial.println("AT+CNMI=1,2,0,0,0");
   Serial.println("AT+CNMI=1,2,0,0,0");
   delay(5000);
   mySerial.println("AT+CMGF=1");
   Serial.println("AT+CMGF=1");
   delay(5000);
}

void envioSMS1()
{
  delay(1000);

mySerial.print("AT+CMGS=");mySerial.print(char(34));mySerial.print(NumeroA);mySerial.print(char(34));mySerial.print(char(10));mySerial.print(char(13));

Serial.print("AT+CMGS=");Serial.print(char(34));Serial.print(NumeroA);Serial.print(char(34));Serial.print(char(10));Serial.print(char(13));
  delay(4000);

mySerial.print(smsA);mySerial.print(NumeroA);mySerial.print(char(26));mySerial.print(char(10));mySerial.print(char(13));

Serial.print(smsA);Serial.print(NumeroA);Serial.print(char(26));Serial.print(char(10));Serial.print(char(13));
  delay(1000);
}

void envioSMS2()
{
  delay(1000);


mySerial.print("AT+CMGS=");mySerial.print(char(34));mySerial.print(NumeroB);mySerial.print(char(34));mySerial.print(char(10));mySerial.print(char(13));

Serial.print("AT+CMGS=");Serial.print(char(34));Serial.print(NumeroB);Serial.print(char(34));Serial.print(char(10));Serial.print(char(13));

  delay(4000);

mySerial.print(smsB);mySerial.print(NumeroB);mySerial.print(char(26));mySerial.print(char(10));mySerial.print(char(13));
```

```
Serial.print(smsB);Serial.print(NumeroB);Serial.print(char(26));Serial.print(char(10));S
erial.print(char(13));
  delay(1000);
}
```

3.3.2.2 Montaje de software

Para la programación se requiere agregar las librerías de NFC, SPI y de comunicación serial, generalmente la librería que no viene por defecto en el programa Arduino es el de NFC <PN532.h>

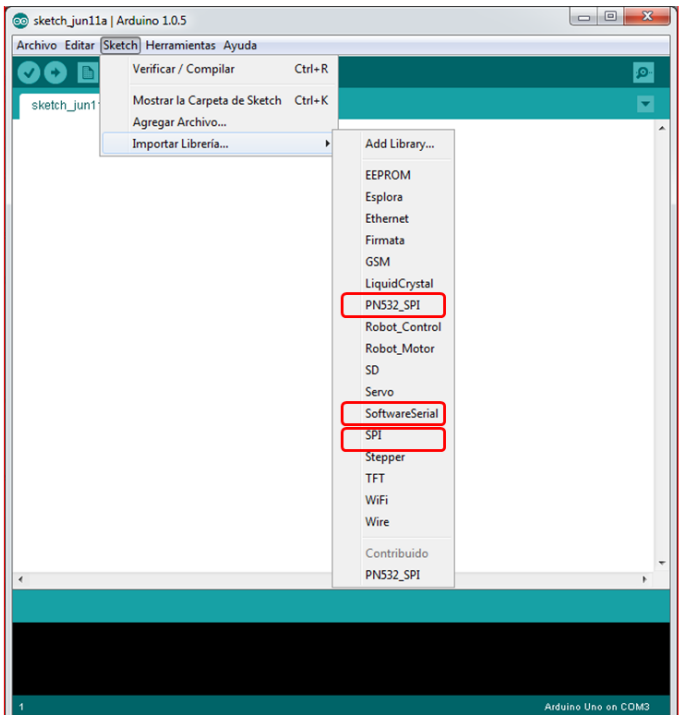


Figura 3.25 Librerías requeridas para el proyecto
Fuente: Fernando Orbe

Con la programación lista, la cual debe estar en el programa Arduino, se la debe correr y poner en funcionamiento mediante la comunicación del monitor serial como se puede ver en la Figura 3.26.

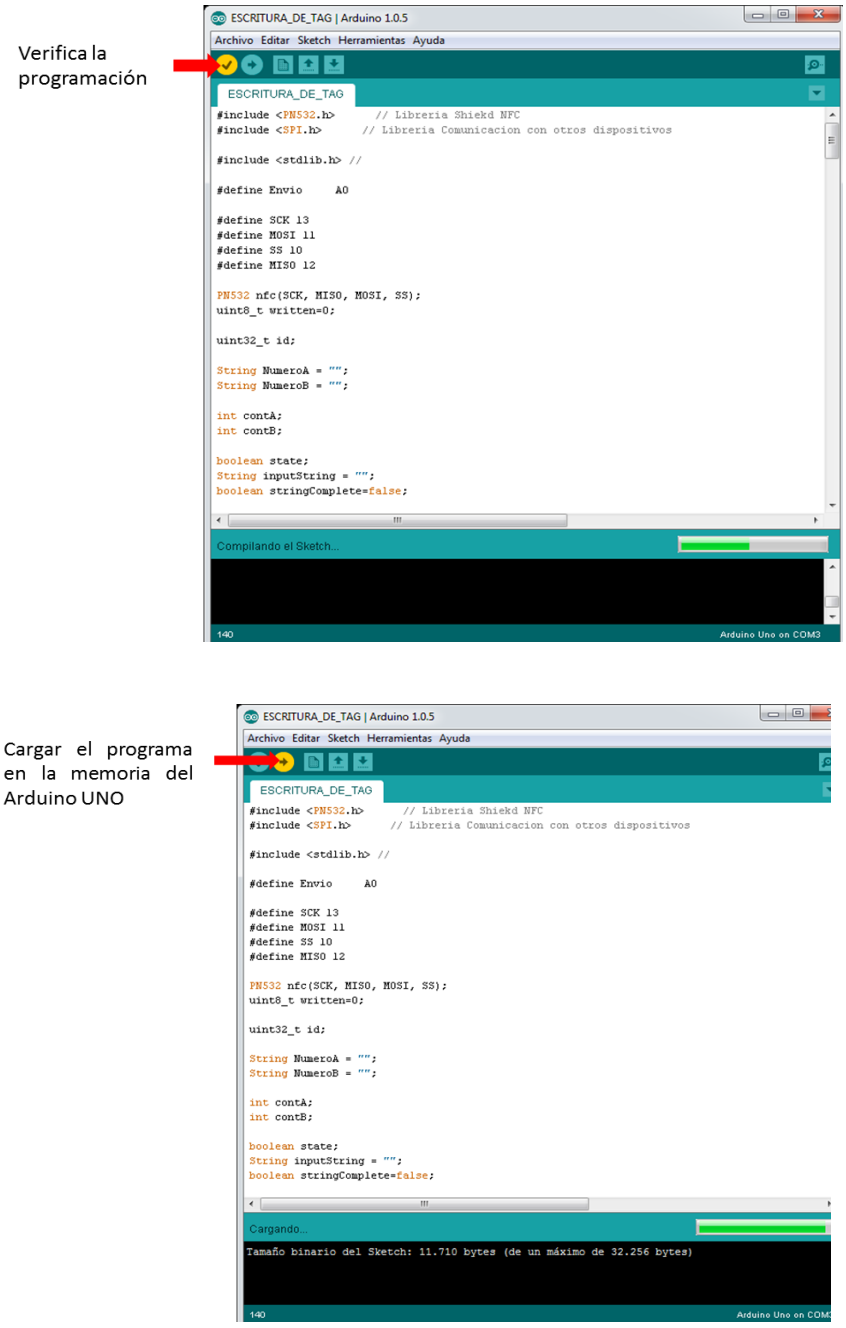


Figura 3.26 Pasos para cargar el programa de ESCRITURA DE TAG
Fuente: Fernando Orbe

Para la Escritura de las tag, el programa diseñado que se debe cargar en la memoria de Arduino UNO en este caso se llama (ESCRITURA_DE_TAG). Una vez cargado el programa abrimos el ícono de Monitor Serial, el mismo que despliega una ventana donde se podrá interactuar con el dispositivo y en este caso escribir los números de

teléfono deseados en las tag adicionalmente debemos cambiar o verificar que la velocidad de comunicación serial sea la adecuada, en este caso es de (9600).

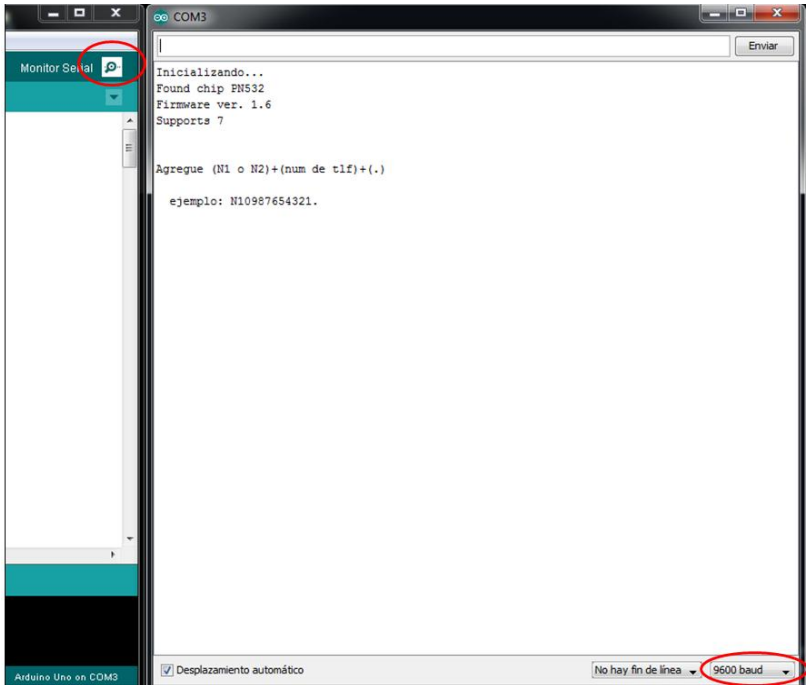


Figura 3.27 Como se presenta por monitor serial, el programa ESCRITURA DE TAG. Fuente: Fernando Orbe

Para demostrar o verificar que el programa de ESCRITURA DE TAG funciona correctamente se puede cargar el programa que nos permite ver lo que está escrito dentro de los bloques de memoria de las tag. El programa mencionado es un ejemplo que viene en la librería del <PN532.h>

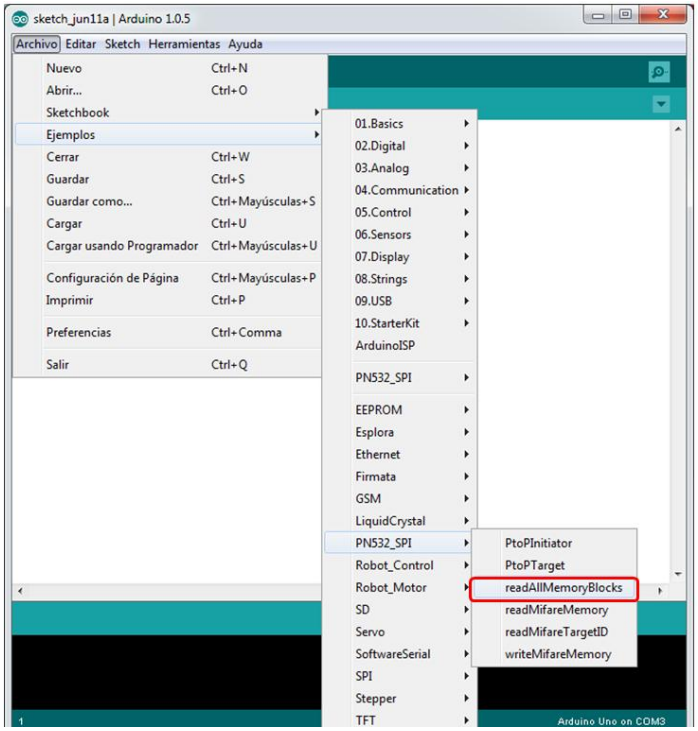


Figura 3.28 Programa para leer los bloques de memoria de la tag
Fuente: Fernando Orbe

Como se puede ver en la Figura 3.29 se verifica en el monitor serial la correcta escritura de los números de teléfono en los bloques de memoria 8, y 9 de la tag.

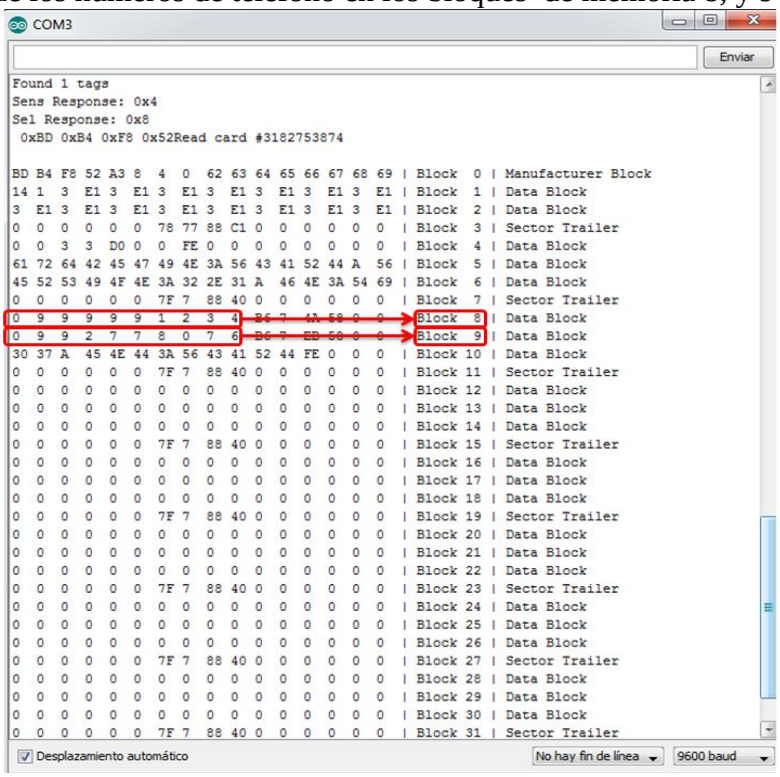


Figura 3.29 Presentación en monitor seria del bloque 8 y 9 de la tag
Fuente: Fernando Orbe

3.3.2.2.1. Montaje de software de lectura de Tag y envío de SMS

De igual manera para la Lectura y envío de mensajes se carga el programa diseñado que en este caso esta con el nombre de (LECTURA_Y_ENVIO_DE_SMS).

Pero antes de cargar el programa al dispositivo es importante resaltar que el mensaje (los datos de la unidad de taxi), que será enviado los teléfonos guardados en las tag debe ser escrito en la programación del diseño (LECTURA_Y_ENVIO_DE_SMS) en la línea de comando que se muestra en la Figura 3.30

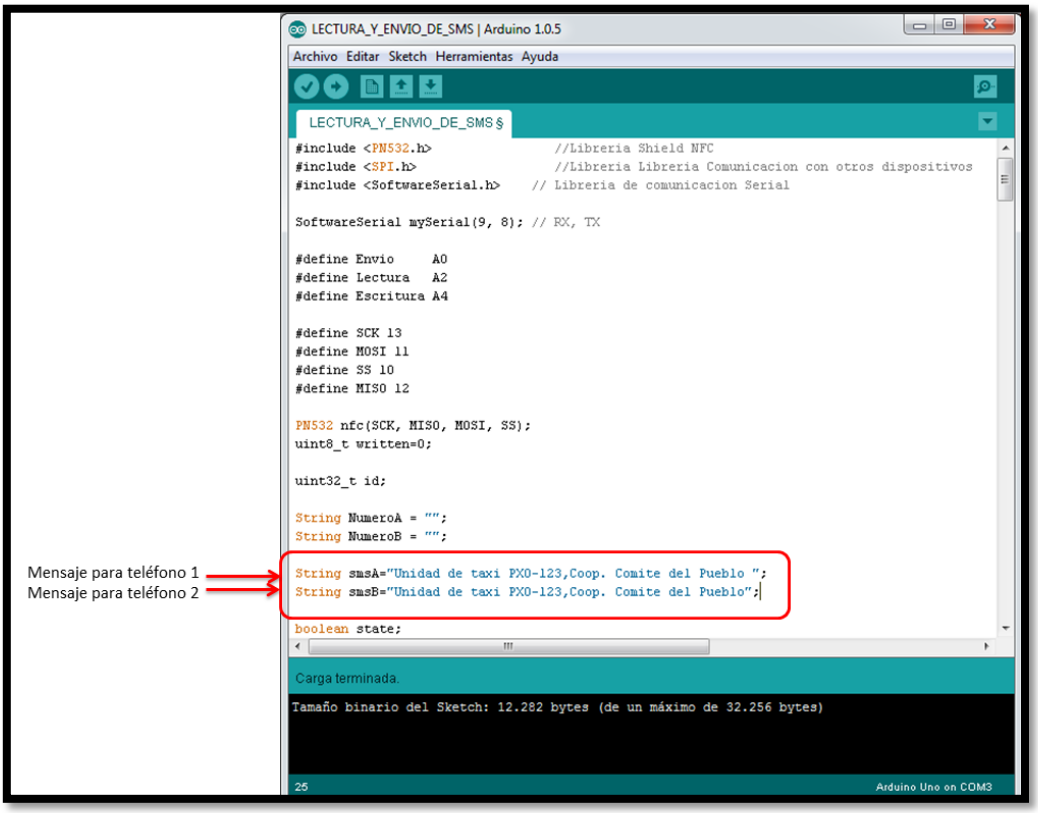


Figura 3.30 Como programar el texto del mensaje que será enviado.
Fuente: Fernando Orbe

Este programa funciona de forma física y sus funciones son activadas automáticamente al momento en que el dispositivo de seguridad NFC detecta el acercamiento de la tag NFC gravada con los números de teléfono.

Para el funcionamiento del dispositivo no requiere la interacción con el monitor serial, pero si se requiere ver cómo funciona podemos visualizar su actividad por medio de este monitor serial como se puede ver en la Figura 3.31.

También en el monitor serial se puede observar los resultados de establecimiento de comunicación serial, comunicación con el dispositivo GSM, después el reconocimiento del acercamiento de la tag y la confirmación y el mensaje (SMS) enviado por el dispositivo y a que números de teléfono se envió el mismo.

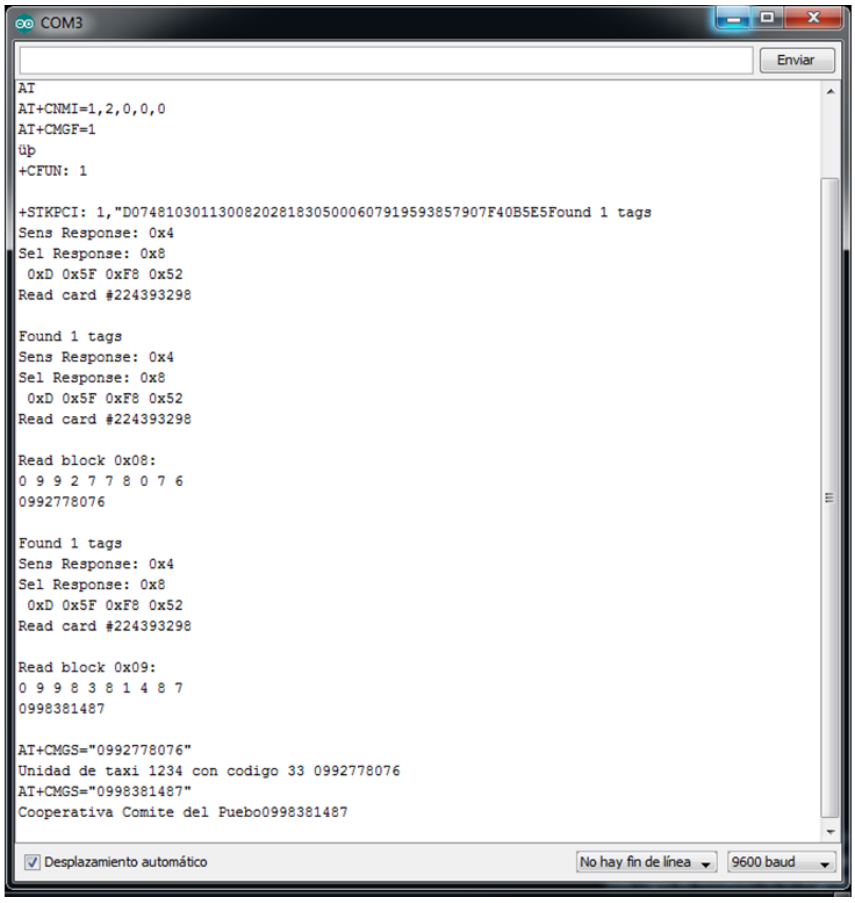


Figura 3.31 Respuesta del monitor serial al programa de LECTURA Y ENVÍO DE SMS
Fuente: Fernando Orbe

CAPÍTULO 4

4.1. INDICACIONES DE MANEJO, RESULTADOS DE FUNCIONAMIENTO, Y COSTOS DEL DISPOSITIVO DE SEGURIDAD NFC.

4.1.1. INDICACIONES DE MANEJO Y RESULTADOS DE FUNCIONAMIENTO DEL DISPOSITIVO DE SEGURIDAD NFC

Una vez establecidos el montaje y diseño del dispositivo de seguridad NFC, podemos ponerlo en pruebas de funcionamiento para lo cual se dará indicaciones del manejo del dispositivo paso a paso.

4.1.1.1. Procedimientos para grabar dos números de teléfono en la tag

Para el proceso de gravado se requiere la interacción con una pc por medio de la comunicación USB

Figura 4.1 conexión por puerto USB para el gravado de números de teléfono en las tag
Fuente: Fernando Orbe

1) Se debe abrir y cargar el programa "***ESCRITURA_DE_TAG"*** en el dispositivo de seguridad NFC.

Figura 4.2 Paso 1, Cargar el programa ESCRITURA DE TAG en Arduino UNO
Fuente: Fernando Orbe

2) Con el programa cargado, se procede a activar la comunicación del monitor serial el cual despliega la inicialización del programa y muestra la forma adecuada de ingresar los números de teléfono para ser gravados.

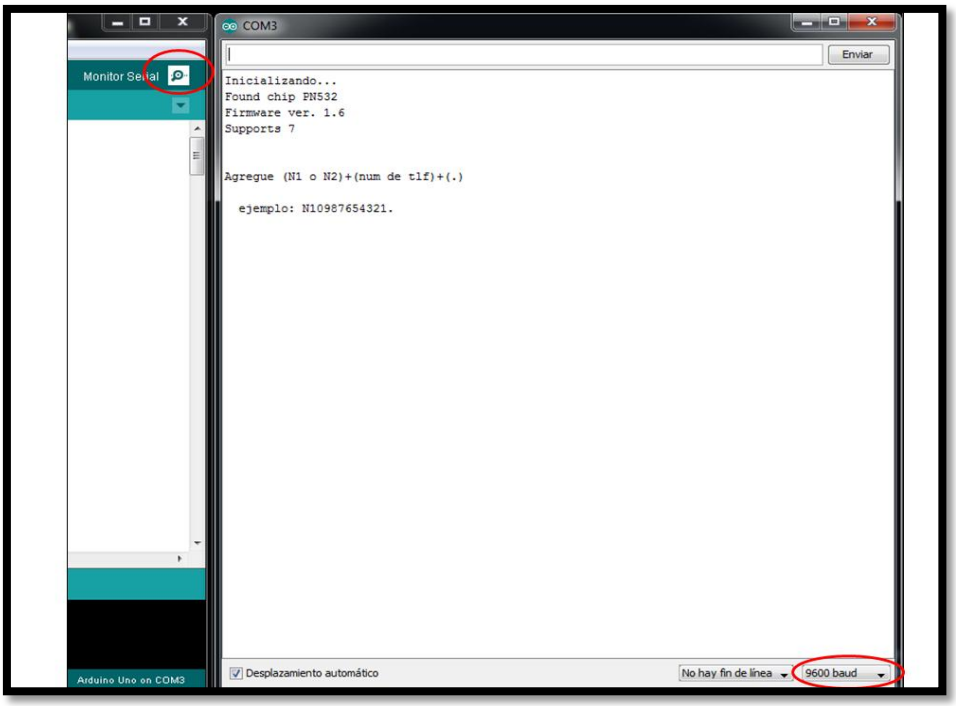


Figura 4.3 Paso 2, Iniciar monitor serial
Fuente: Fernando Orbe

Como se muestra en la Figura 4.3 el dispositivo está listo para grabar, y como se observa en el monitor serial se despliega un mensaje con el formato adecuado q se debe ingresar los números de teléfono

Agregue: **(N1 o N2) + (num de tlf) + (.)**

Ejemplo: **N1 0987654321 .**

3) Para proceder a grabar los números de teléfono, acercamos la tag que deseamos grabar hacia el dispositivo de seguridad NFC.

Figura 4.4 Paso 3, Acercar la tag Al dispositivo de seguridad NFC
Fuente: Fernando Orbe

Manteniendo el acercamiento de la tag hacia el dispositivo NFC se procede a ingresar por el monitor serial el primer número de teléfono, como se muestra en la Figura 4.5.

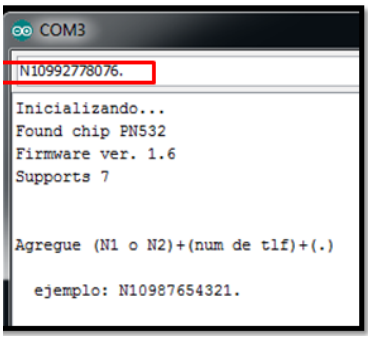


Figura 4.5 Ingreso del primer nuero de teléfono (N1)
Fuente: Fernando Orbe

Si el proceso de gravado es correcto, en el monitor serial se despliega un mensaje indicando que (Escritura del teléfono 1, correcta) como se muestra en la Figura 4.6, caso contrario mientras la tag no se encuentre cerca no obtendremos este mensaje.

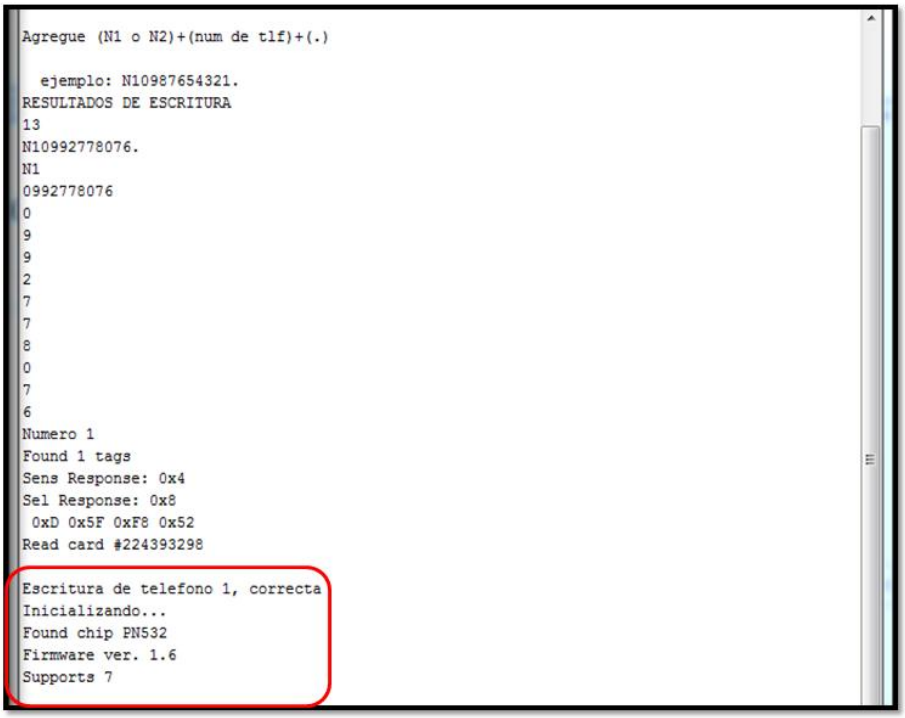


Figura 4.6 Mensaje de escritura de tag correcta
Fuente: Fernando Orbe

4) Una vez gravado el primer teléfono quedara habilitado nuevamente el programa para rectificar o agregar el número de teléfono 2, como se muestra en la Figura 4.7.

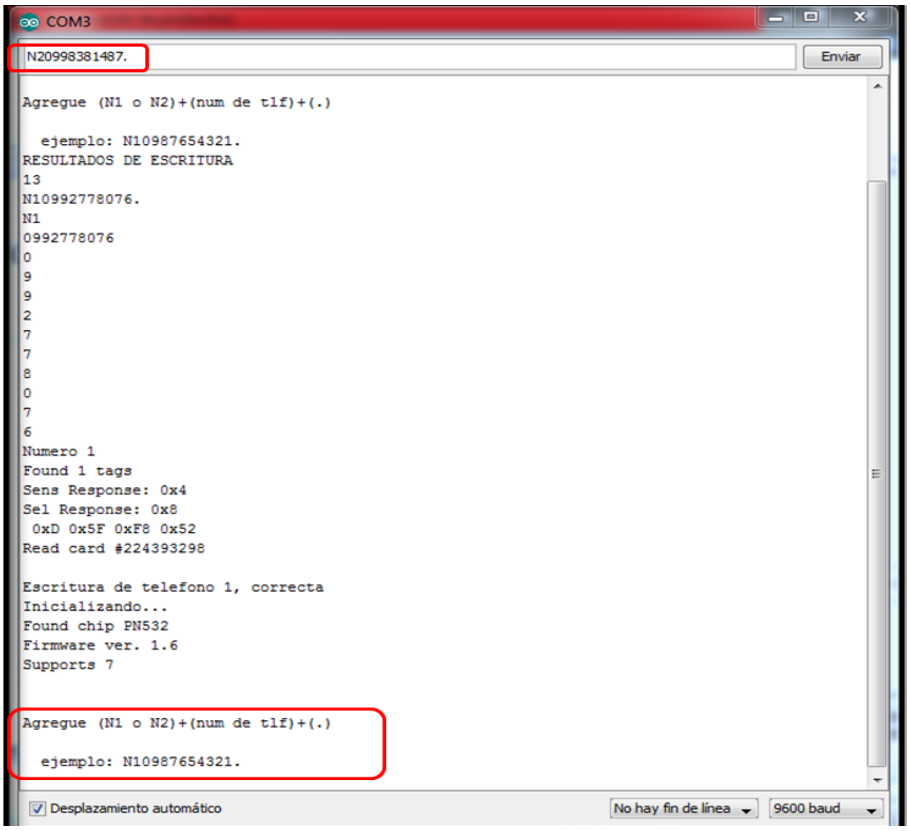


Figura 4.7 Paso 4, Ingreso y mensaje de escritura correcta del segundo número de teléfono (N2)
Fuente: Fernando Orbe

5) Después de concluir de grabar los dos números de teléfono en la tag, el programa queda habilitado para grabar dos números de teléfono más dentro de una nueva o siguiente tag. Por lo tanto, de esta forma mientras no se cierre el monitor serial ni se interrumpa la comunicación USB con el dispositivo NFC, este se mantendrá listo para grabar a indefinido número de tag´s los dos números de teléfono requeridos por el usuario en cada tag mencionada.

4.1.1.2. Procedimientos para lectura de los números de teléfono gravados en la tag y envío de SMS.

El proceso de envío de mensajes del dispositivo de seguridad NFC en un proceso automático en el instante mismo en que detecta cerca una tag programada con los números de teléfono.

1) Primero en el dispositivo de seguridad NFC debe estar cargado el programa (***LECTURA_Y_ENVIO_DE_SMS),*** el mismo que está programado para un funcionamiento automático que en el momento de encendido quedara listo para recibir la señal de las tag.
2) Para el funcionamiento del dispositivo, este es alimentado con un regulador o una instalación dirigida hacia la batería del automóvil el mismo que proporciona 12V que están dentro del estándar del dispositivo.

Como se presenta en la Figura 4.8 la instalación eléctrica es de fácil conexión y se la realiza de acuerdo al auto móvil ya que varía en cierta forma en sus instalaciones. Se toma una señal eléctrica de 12v de la batería (A) pero sin antes verificar que pase por la caja de fusibles (B) y de esta forma proteger al dispositivo (C).

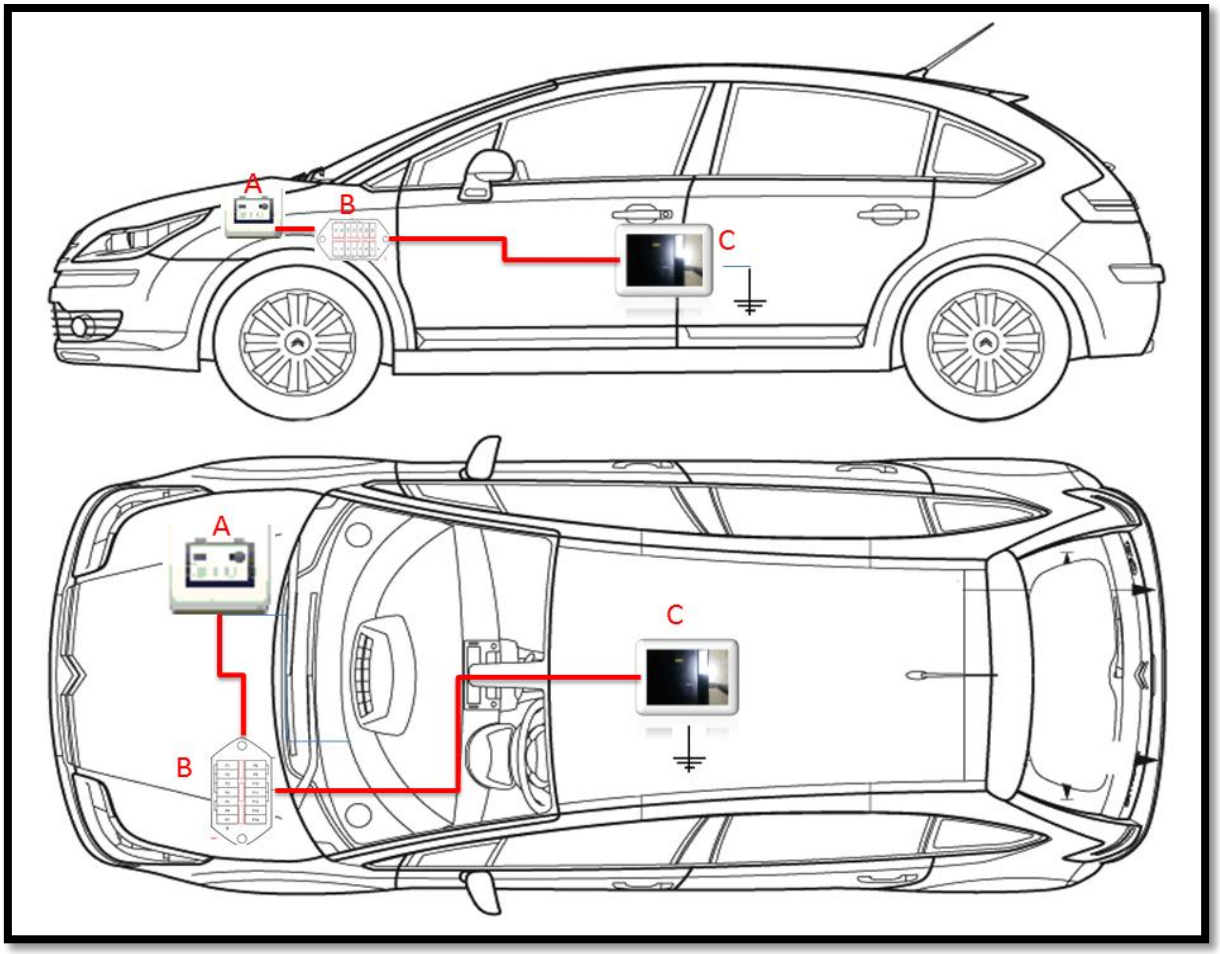
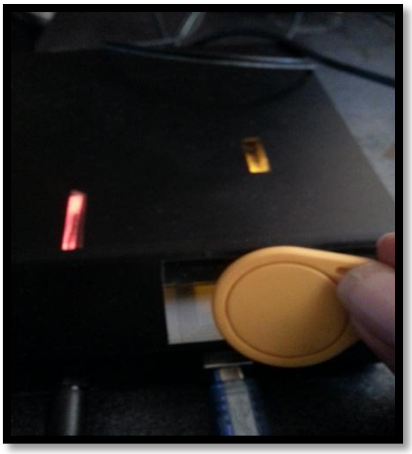


Figura 4.8 Paso 2, Instalación del dispositivo de seguridad NFC en el taxi.
Fuente: Fernando Orbe

3) Una vez activado y encendido el dispositivo, el proceso de envío de mensajes se realizara en el momento en que el dispositivo detecte el acercamiento de una tag en la que se encuentran los números de teléfono y donde será enviado los datos de la unidad de taxi. En la Figura 4.9 se presenta un ejemplo del funcionamiento del dispositivo de seguridad NFC. Como se había mencionado anteriormente el mensaje de los datos del Taxi deben ser programados antes de cargar este programa al Dispositivo de Seguridad NFC.



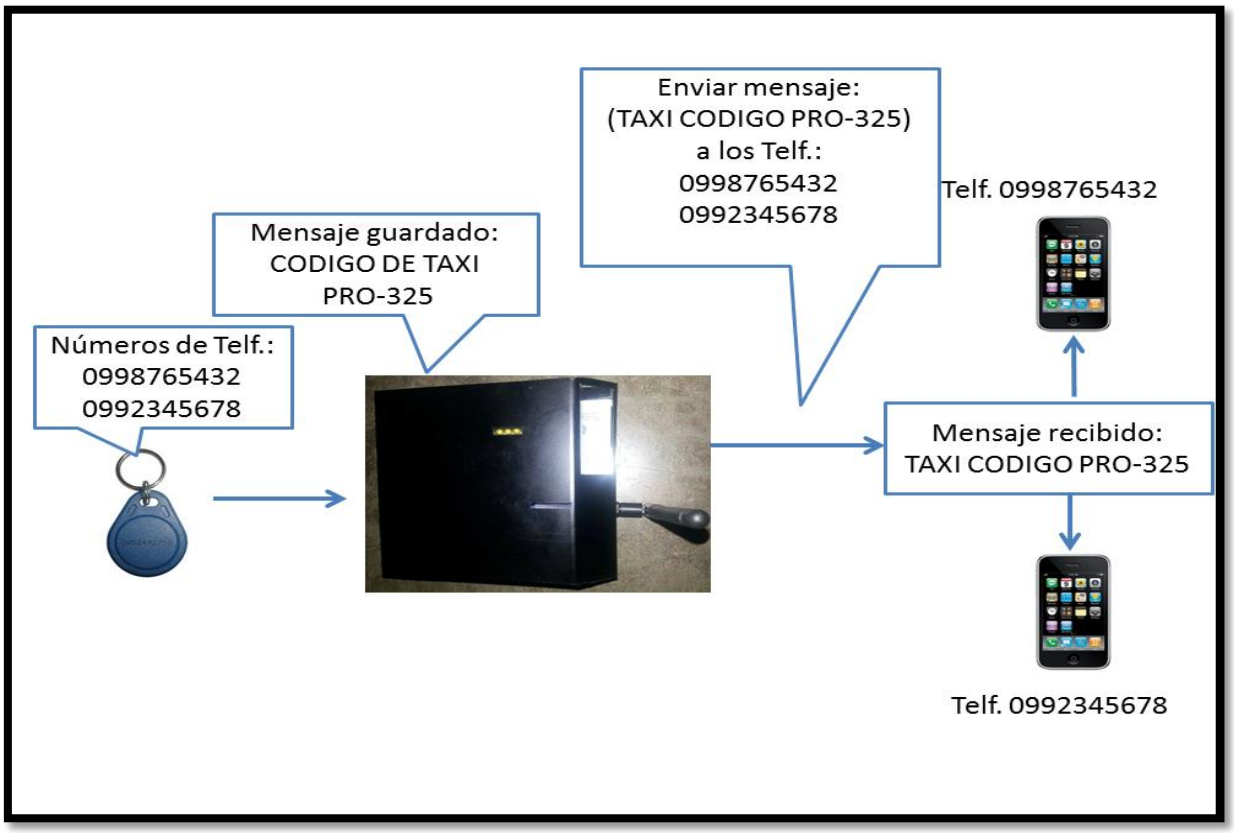


Figura 4.9 Paso 3, Demostración y ejemplo de funcionamiento de envío de SMS.
Autor: Fernando Orbe

4.2. ANÁLISIS FODA

Al iniciar a estructurar el análisis FODA del "Dispositivo de Seguridad NFC" hay que tener claro que este análisis consta de factores internos y externos es así que para los internos que son las Fortalezas y Debilidades y externos las Oportunidades y Amenazas.

Figura 4.10 Cuadro de análisis FODA
Fuente: Fernando Orbe

Mediante el análisis FODA realizado se puede ver que la principal causa de dar a conocer este dispositivo es brindar un servicio de seguridad tanto para las personas que laboran manejando taxis como para aquellas que utilizan este servicio para movilizarse, en cuanto las debilidades y las amenazas siempre va a estar expuesto a los cambios repentinos de tecnología , es por ello que como oportunidad se tiene que puede ir mejorando con el tiempo para que así satisfaga las necesidades del cliente.

4.3. ANÁLISIS FINANCIERO

En el presente análisis se tiene un desglose de los precios de los elementos electrónicos utilizados para la elaboración del prototipo y otros componentes. Cabe mencionar que este análisis se lo realiza para conocer cuál será el precio final del dispositivo.

TABLAS DE ELEMENTOS DEL DISPOSITIVO DE SEGURIDAD

Elementos	Cantidad	Costo c/u	Costo total	Costo al por mayor c/u	Costo total al por mayor
Tag's NFC	4	5	20	1	4
Placa Arduino	1	30	30	20	20
Placa NFC Shield	1	20	20	15	15
MODEM GPRS- GSM ZTE	1	70	70	50	50
TOTAL		**$ 125,00**	**$ 140,00**	**$ 86,00**	**$ 89,00**

Tabla 4 Elementos usados para el dispositivo de seguridad
Fuente: Fernando Orbe

OTROS MATERIALES

Elementos	Cantidad	Costo c/u	Costo total	Costo al por mayor c/u	Costo total al por mayor
Cable COM USB	1	3	3	2	2
Transformador 9V , 1A	1	5	5	3	3
TOTAL		$8,00	$8,00	$5,00	$5,00

Tabla 5 Costos de material adicional.
Fuente: Fernando Orbe

CAJA ACRÍLICA

Elementos	Cantidad	Costo c/u	Costo total	Costo al por mayor c/u	Costo total al por mayor
Plancha de acrílico 50cm x 50cm	1	5	5	3	3
Forro negro 50 cm cuadrados	1	4	4	2	2
Silicón Transparente 310ML	1	3,53	3,53	1	1
pega acrílica	1	1,5	1,5	0,5	0,5
mano de obra	1	20	20	10	10
Tornillos	4	0,1	0,4	0,05	0,2
TOTAL		**$34,13**	**$34,43**	**$16,55**	**$16,70**

Tabla 6 Costos del diseño e implementación de la caja.
Fuente: Fernando Orbe

PLACA DE COMUNICACIÓN RS232

Elementos	Cantidad	Costo c/u $	Costo total $	Costo al por mayor c/u $	Costo total al por mayor $
Dispositivos electrónicos					
Led	3	0,10	0,3	0,5	1,5
Swich GR BL	1	0,65	0,65	0,4	0,4
regleta de conectores	1	0,45	0,45	0,2	0,2
R 1/4W	3	0,27	0,81	0,1	0,3
Resistencia 1/4 4.7 KOhm	2	0,27	0,54	0,1	0,2
Resistencia 1/4 1 KOhm	2	0,27	0,54	0,1	0,2
1N3904	2	0,36	0,72	0,25	0,5
DB9 Placa Hembra	1	0,55	0,55	0,4	0,4
DB9 Macho Cable	2	0,9	1,8	0,6	1,2
Baquelita	1	0,6	0,6	0,4	0,4
bus de datos	1	0,45	0,45	0,3	0,3
TOTAL		**$4,87**	**$7,41**	**$3,35**	**$5,60**

Tabla 7 Elementos usados para placa de comunicación RS-232
Fuente: Fernando Orbe

MANO DE OBRA DEL SISTEMA = $30,00

Para encontrar el costo total se procede a sumar los subcostos de cada tabla de materiales, a continuación se dará a conocer tanto los costos totales calculados al por menor como el costo total al comprar por mayor.

DETALLE	Al por menor	Al por mayor
Tabla del dispositivo de seguridad	$ 140,00	$ 89,00
Placa de comunicación rs232	$ 7,41	$ 5,60
Caja acrílica	$ 34,43	$ 16,70
Otros materiales	$ 8,00	$ 5,00
Mano de obra del sistema	$ 30,00	$ 30,00
COSTO TOTAL…	**$ 219,84**	**$ 146,30**

Tabla 8 Costo final del prototipo del dispositivo de seguridad NFC
Autor: Fernando Orbe

Como se muestra en los resultados presentados en la tabla de costos totales vemos que es una gran opción al hablar de seguridad ya que el costo es accesible al contar con la compra al por mayor teniendo en cuenta que las empresas tienen como objetivo brindar un mejor servicio que satisfaga al cliente y que no implique un costo alto que perjudique su rentabilidad.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Las pruebas de funcionamiento de; escritura, lectura de NFC y envío de SMS, fueron exitosas cumpliendo de esta forma los objetivos planteados.
- El dispositivo de seguridad NFC, brinda mayor confianza tanto para el usuario como para el conductor de la unidad
- El dispositivo de seguridad NFC contara con la comunicación USB libre en caso de que se requiera modificar el tipo de mensaje a enviar o cargar nuevamente la programación si en caso existe algún error o falla del mismo
- Para evitar que se reitere el envío de mensajes y estos puedan gastarse tiene un tiempo de retardo antes de que el dispositivo pueda enviar nuevamente los SMS.
- Los conductores de taxis entrevistados, se mostraron muy asertivos y optimistas con la presentación del proyecto, ya que el mismo les ayuda a que sus pasajeros sean en su mayor parte personas que ellos ya conocen.
- Los usuarios encuestados en su 98% están de acuerdo con el método de seguridad presentado, ya que no requieren comprar o tener dispositivos de alta tecnología y de mayor costo.
- Arduino aparte de ser una solución libre cuenta también con múltiples formas de implementación de proyectos haciendo estos mucho más fáciles de realizarlos.

- Las placas diseñadas por Arduino son acoplables y de fácil programación.
- NFC es una tecnología muy poco explotada en el país y que puede mejorar la seguridad y facilitar muchos proyectos de seguridad
- Las tag NFC tiene suficiente espacio de memoria para guardar varios datos o números de teléfono en todos sus bloques de memoria.
- El tiempo de respuesta de envío de los SMS, una vez detectada la tag depende de la operadora.
- Para la instalación del dispositivo en cualquier tipo de automóvil, cuenta con la facilidad de que sus características técnicas soporten de una forma adecuada el voltaje de la batería del automóvil que es de 12VDC, sin contar que el consumo de corriente de los dispositivos son bajos.

5.2. RECOMENDACIONES

- Se recomienda mantener siempre el dispositivo con saldo y paquete de mensajes, ya que el chip del prototipo es de movistar y se requiere saldo para enviar mensajes hacia otra operadora diferente.
- Una fuente de energía que en la mayoría de los autos es la más adecuada para utilizar dispositivos externos, es la fuente del encendedor eléctrico.
- Este prototipo está proyectado a ser mejorado ya que el mismo puede ser la base para utilizar varios métodos electrónicos conjuntamente y así controlar no solo información por SMS sino puede llegar a ser un proyecto con datos, control de usuarios y unidades de taxis u otros tipos de transporte.

- Se recomienda utilizar un modem GSM de trabajo pesado, en el caso del modem GSM de Arduino es vulnerable y su comunicación es inestable después de varias horas de trabajo.
- La instalación del dispositivo debe ser sellada para que los cables de energía no puedan ser manipulados.
- Es recomendable las Tag de Mifare por su capacidad de gravables y resistente a mala manipulación, variaciones de temperatura y humedad.
- Se debe utilizar tag's de Mifare debido a que está programado para el reconocimiento de este tipo de tag's, caso contrario con otras tag no obtendrá ningún resultado.

BIBLIOGRAFÍA Y LINKOGRAFÍA

ARDUINO. (2013). *Pagina oficial*. Recuperado el 2014, de http://arduino.cc/es/Main/Software

Arduino, T. (28 de Junio de 2013). *Arduino pinout y conexiones básicas*. Obtenido de http://tallerarduino.com/tag/shield/

ATMEL. (Febrero de 2009). *Datasheet ATmega3*. Obtenido de http://www.atmel.com/Images/doc8161.pdff

B.V. NXP. (10 de Noviembre de 2011). *data sheet NFC cintroller PNX532*. Obtenido de http://www.nxp.com/documents/short_data_sheet/PN532_SDS.pdf

BLAM electronics. (s.f.). *COMPONENTES DE TARJETA ARDUINO*. Recuperado el 2014, de http://mikiblam.blogspot.com/p/blog-page.html

blogElectronica.com. (2008). *SMS formato PDU*. Obtenido de file:///C:/Users/FERCHOXD/Desktop/TESIS/FOTOS%20TAG/Fernando%20Orbe/SMS%20en%20formato%20PDU%20_%20blogElectronica.com.htm

CASE Congreso Argentimo de Sistemas Embebidos. (4 de 3 de 2011). *Libro de Trabajos de NFC*. Obtenido de Tecnología inalámbrica Near Field Communication: http://www.sase.com.ar/2011/files/2011/01/CASE2011_Libro_de_Trabajos.pdf

Comparduino electronics y openhardware. (31 de Marzo de 2014). *Microcontrolador Atmega 328P-PU*. Obtenido de http://comparduino.com/product_info.php/products_id/231

Dynamo Electronics. (2014). *Arduino RFID Shield*. Obtenido de http://www.dynamoelectronics.com/descargas/rfid%20shield.pdf

ECMA International. (Octubre de 2002). *Near Field Communication* . Obtenido de Standardizing Information and Communication System: http://www.ecma-international.org/activities/Communications/2002tg19-010.pdf

El Universo. (24 de Junio de 2013). Taxis con tecnología y seguridad. págs. http://www.eluniverso.com/noticias/2013/06/24/nota/1060951/taxis-tecnologia-seguridad.

E-Lins Technology. (s.f.). *M100g GPRS Modem* . Obtenido de http://www.szelins.com/Wireless_GPRS_Modem.html

Enlace de ARDUINO - Seeedstudio wiki. (21 de 04 de 2014). *NFC Shield V2.0*. Recuperado el 2014, de http://www.seeedstudio.com/wiki/NFC_Shield_V2.0

Escrito por Vedat Coskun, K. O. (2012). Near Field Communication (NFC): From Theory to Practice. En K. O. Escrito por Vedat Coskun. WILEY.

Foro de Nuevas Tecnologías en el Transporte, ITS España. (2013). Libro Blanco Sobre la Aplicacion de la Tecnoñogía NFC en Transporte Público. En I. E. (Henri Dunant) Foro de Nuevas Tecnologías en el Transporte, *Libro Blanco* (pág. 88). España: ISBN – 10: 84-616-4714-9 .

GSM and TDMA Technology. (s.f.). Obtenido de http://www.cwins.wpi.edu/publications/pown/chapter_7.pdf

Idoneum Electronic Identity. (2011). *Identificación y seguridad en NFC*. Obtenido de http://slideplayer.es/slide/114533/

Ingeniatic. (2011). *SMS*. Obtenido de http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/600-sms-servicio-de-mensajes-cortos

Josef Noll, J. C. (11 de Mayo de 2005). *Business through Mobile Phone initiated Near Field Communication*. Obtenido de NFC standardisation: http://wiki.unik.no/media/Unik/200505NUf-NFC-Mobile-Noll-Calvet.pdf

LinkSprite. (9 de 2 de 2013). *NFC PN532 Shield*. Recuperado el 2014, de file:///C:/Users/FERCHOXD/Desktop/archivos%20de%20escritorio/GSM/Shield_NFC/NFC%20PN532%20Shield%20-%20LinkSprite%20Playgound.htm

LUCKYSTAR'S OPINIONS. (16 de 06 de 2009). *Tutorial :: Using AT commands to Send and Receive SMS*. Recuperado el 2014, de http://oldlight.wordpress.com/2009/06/16/tutorial-using-at-commands-to-send-and-receive-sms/

Manualslib. (15 de 4 de 2008). *Motorola AT Commands G24-L Technical Information*. Recuperado el 2014, de http://www.manualslib.com/manual/130670/Motorola-At-Commands-G24-L.html?page=147#manual

Mundo NFC. (9 de Febrero de 2012). *Funcionamiento de NFC*. Obtenido de http://mundonfc.wordpress.com/category/uncategorized/

NFC Forum. (2014). *NFC and Contactless Technologies*. Obtenido de http://nfc-forum.org/what-is-nfc/about-the-technology/

NXP(Philips). (15 de Junio de 2007). *data sheet Tag Mifare*. Obtenido de http://www.stronglink-rfid.com/download/M001052.pdf

RUEDA, R. (19 de Mayo de 2013). En Quito también hay zozobra por los secuestros express. *El Universo*, págs. http://www.eluniverso.com/noticias/2013/05/18/nota/935226/quito-tambien-hay-zozobra-secuestros-express.

Telecomunicaciones, E. N. (13 de Junio de 2006). *La tecnología NFC entra de lleno en nuestro móvil*. Obtenido de http://e-volucion.elnortedecastilla.es/actualidad-digital/movilidad/la-tecnologia-nfc-entra-de-lleno-en-nuestro-movil-13062012.html

WordPress Arduino. (28 de Marzo de 2012). *RS232 level converter for Arduino*. Obtenido de http://arduinodiy.wordpress.com/2012/03/28/rs232-levelconverter-for-arduino/

ZTE, C. (2007). *AT command Manual for ZTE Modem*. Obtenido de http://www.bointec.com/060.support/010.download/010.cellular_converter/manual/MC221-Z/ZTE%20MG3006%20AT-CMD.pdf

ANEXOS

ANEXO 1

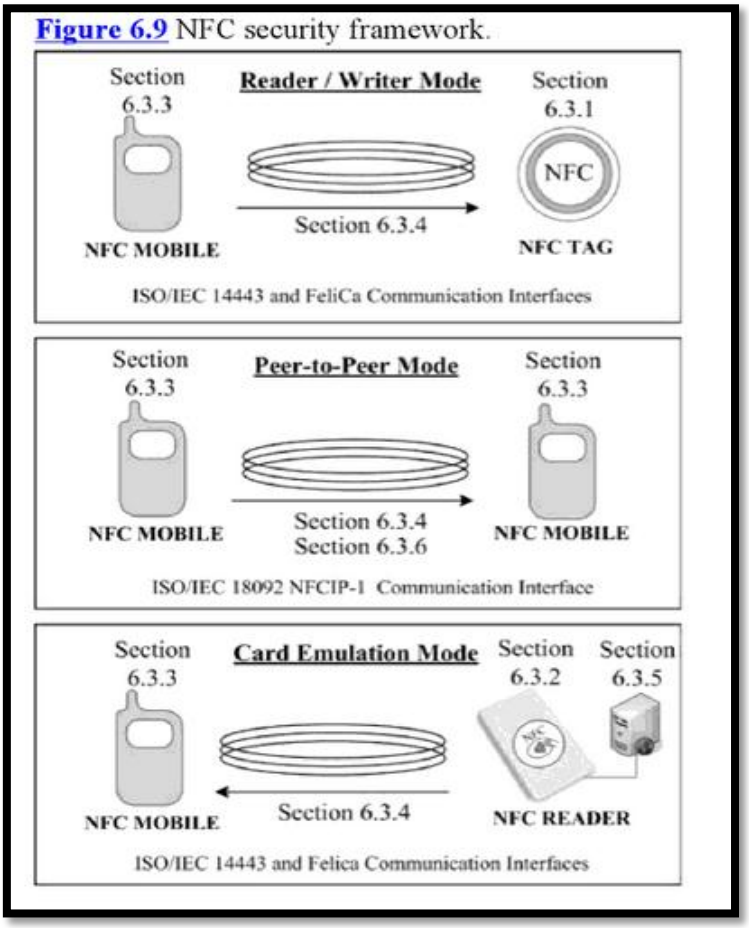
Datos de Tecnología NFC

TABLA I. COMPARACIÓN ENTRE TECNOLOGÍAS INALÁMBRICAS

	NFC	RFID	WiFi	Bluetooth	ZigBee	IrDA
Estándar	ISO/IEC 18092	ISO/IEC 14443	IEEE 802.11	IEEE 802.15.1	IEEE 802.15.4	IrDA
Tasa de transferencia	106-424 Kbps	106-424 Kbps	11-200 Mbps	1-480 Mbps	20-250 kbps	1 Kbps – 100 Mbps
Frecuencia de funcionamiento	13,56 MHz	13,56 MHz	2.4, 5.25, 5.6, 5.8 GHz	2.4 GHz	868/915 MHz 2.4 GHz	
Cantidad máxima de dispositivos que pueden interactuar	2	2	Indefinida	8	Indefinida	2
Tiempo de inicialización	< 0,1 ms	< 0,1 ms	< 0,1 ms	6 s	< 0,1 ms	0,5 ms

Datos de Tecnología NFC

Fuente: (Escrito por Vedat Coskun, 2012)



STANDARDIZATION BODY	STANDARD	DESCRIPTION
ISO/IEC	ISO/IEC 18092	Near Field Communication Interface and Protocol (NFCIP-1)
	ISO/IEC 21481	Near Field Communication Interface and Protocol (NFCIP-2)
	ISO/IEC 28361	Near Field Communication Wired Interface (NFC-WI)
	ISO/IEC 14443	Contactless Proximity Smart Cards and their technical features
	ISO/IEC 15693	Contactless Vicinity Smart Cards and their technical features
ETSI	ETSI TS 102 190	Near Field Communication Interface and Protocol (NFCIP-1)
	ETSI TS 102 312	Near Field Communication Interface and Protocol (NFCIP-2)
	ETSI TS 102 541	Near Field Communication Wired Interface (NFC-WI)
	ETSI TS 102 613	Contactless front end (CLF) interface for UICC, physical and data link layer characteristics; Single Wire Protocol (SWP)
	ETSI TS 102 622	Contactless front end (CLF) interface for UICC, Host Controller Interface (HCI)
ECMA	ECMA 340	Near Field Communication Interface and Protocol (NFCIP-1)
	ECMA 352	Near Field Communication Interface and Protocol (NFCIP-2)
	ECMA 356	NFCIP-1 - RF Interface Test Methods
	ECMA 362	NFCIP-1 - Protocol Test Methods
	ECMA 373	Near Field Communication Wired Interface (NFC-WI)
	ECMA 385	NFC-SEC: NFCIP-1 Security Services and Protocol
	ECMA 386	NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES
	ECMA 390	Front-End Configuration Command for NFC-WI

Datos de Tecnología NFC

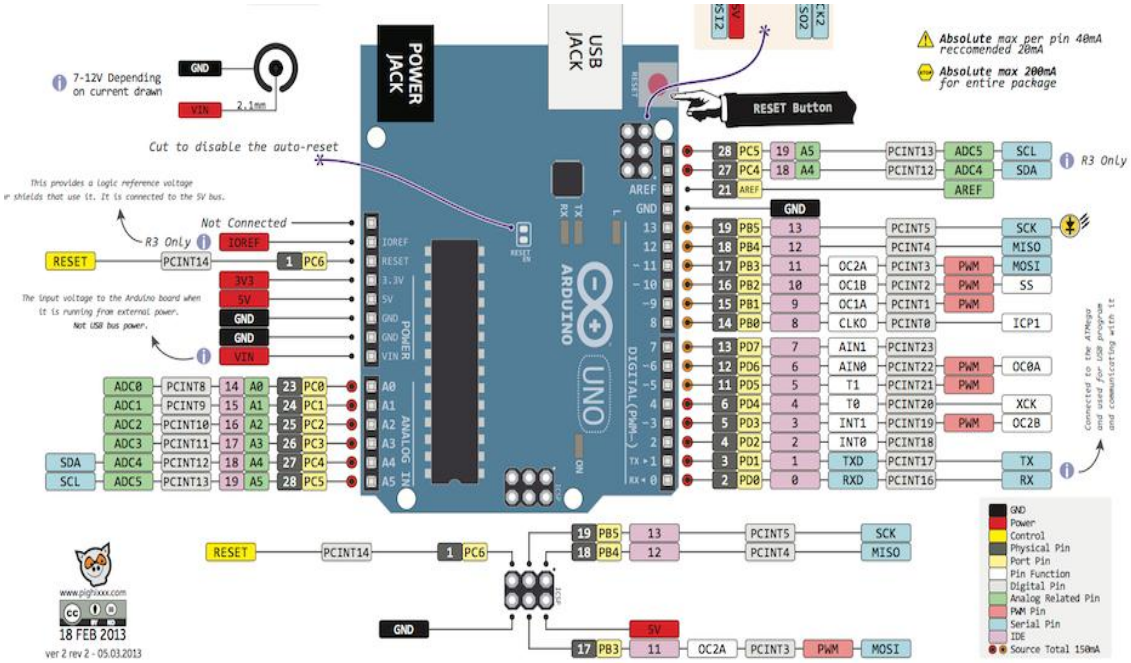
Fuente: (Escrito por Vedat Coskun, 2012)

ANEXO 2

Datos técnicos de la placa electrónica Arduino UNO

2 TARJETA ARDUINO

La tarjeta Arduino, contiene para interacción con el usuario trece entradas/salidas digitales, seis entradas analógicas y un puerto serial que permite realizar comunicación con periférico, además de un puerto serial una conexión USB, en la figura podemos observar la localización de las entradas analógicas y digitales como los pines de alimentación. También tiene un pulsador para resetear cualquier fallo que exista en los procesos que se vayan a realiza con la tarjeta Arduino.



Datos técnicos de la placa electrónica Arduino UNO

Fuente: (Arduino, 2013)

2 HARDWARE

El microprocesador ATmega328
• 32 kbytes de memoria Flash
• 1 kbyte de memoria RAM
• 16 MHz • Entradas y salidas
• 13 pins para entradas/salidas digitales (programables)
• 5 pins para entradas analógicas
• 6 pins para salidas analógicas (salidas PWM)
• Completamente autónomo: Una vez programado no necesita estar conectado al PC

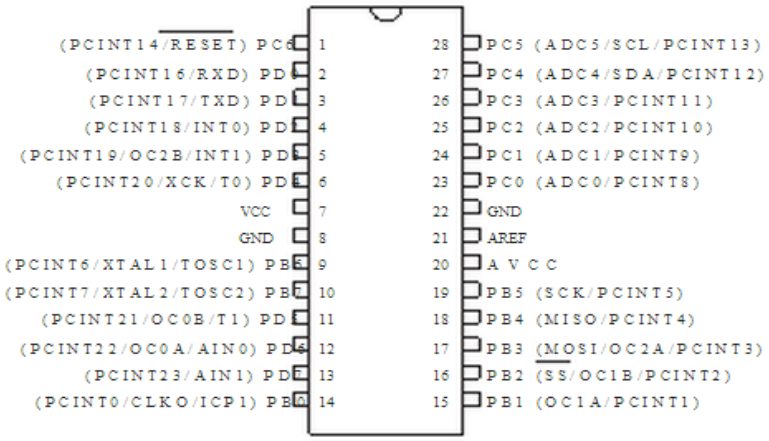
Datos técnicos de la placa electrónica Arduino UNO

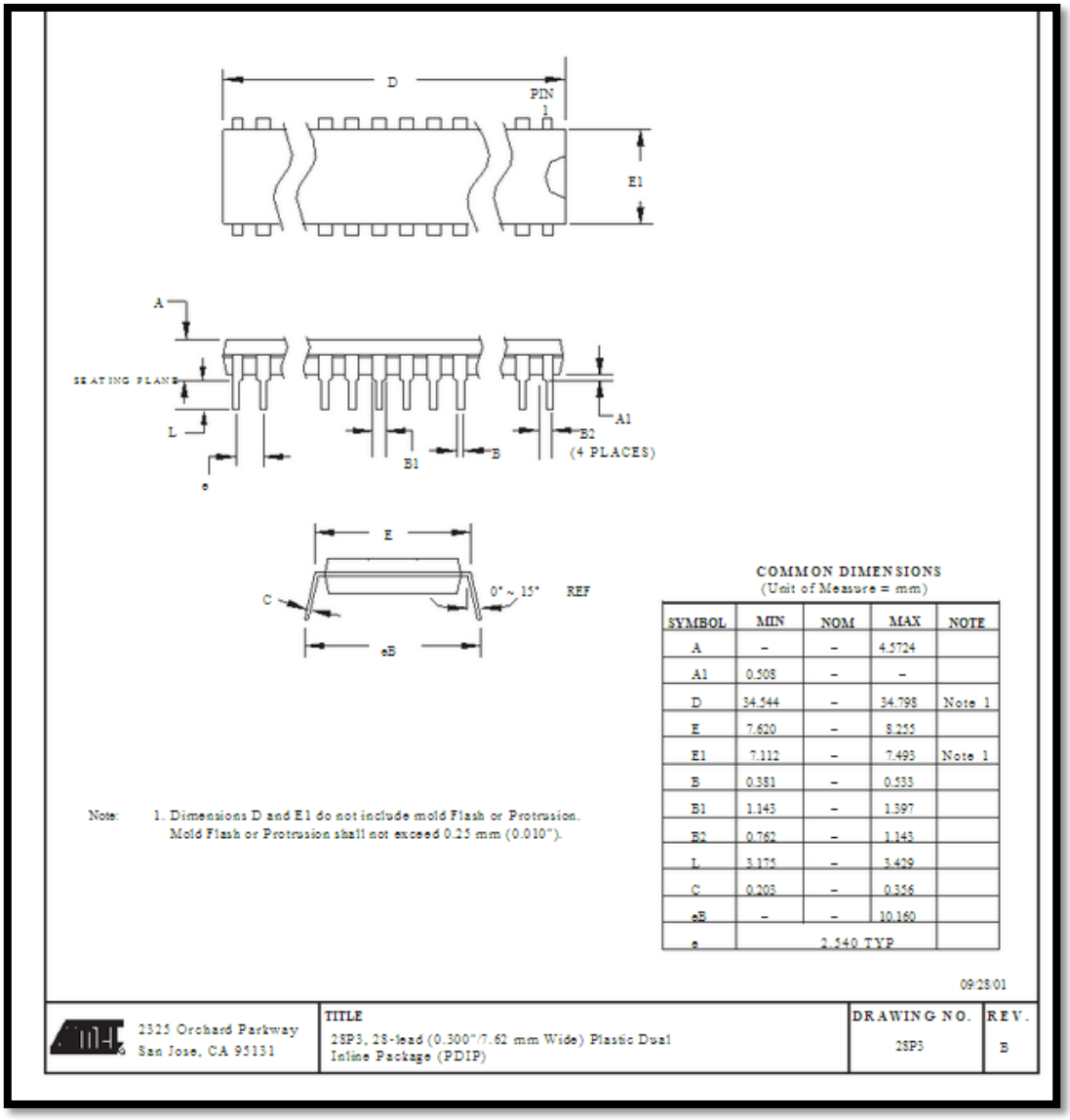
Fuente: (Arduino, 2013)

ANEXO 3

Datos técnicos del

ATMEGA 328P





Datos técnicos del ATMEGA 328P
Fuente: (ATMEL, 2009)

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 88.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

Datos técnicos del ATMEGA 328P
Fuente: (ATMEL, 2009)

Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P

The ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

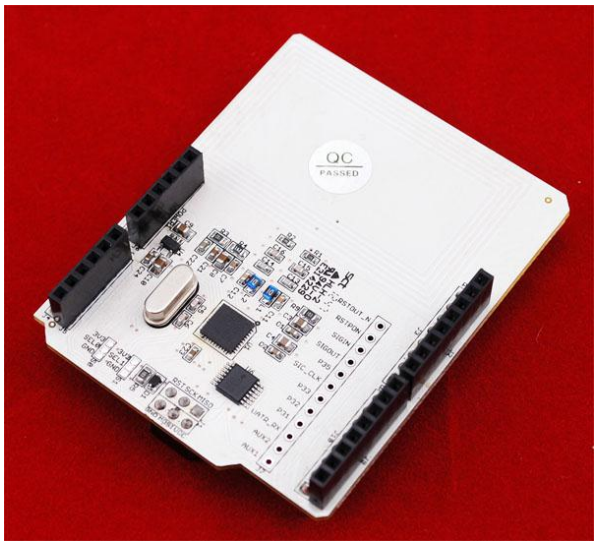
ATmega88PA, ATmega168PA and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48PA, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

Datos técnicos del ATMEGA 328P
Fuente: (ATMEL, 2009)

ANEXO 4

Datos técnicos de la Shield

NFC V1.0 y PN-532DS



Schematic

schematics of NFC shield

Specification

Item	Min	Typical	Max	Unit
Voltage	4.3	5.0	5.7	V
Current	80.0	90.0	100.0	mA
Maximum Communication Distance		2.9		cm
Dimension	69.1x55.7x17.8			mm
Supported Card Type	Mifare One			/
Net Weight	18.5			g

Application Programming Interfaces

NFC is a secure technology (meaning the communication between NFC reader/writer and NFC card/tag happens in an encrypted and authenticated manner). The security and other complex handshaking are handled by PN532 firmware provided by NXP.

The APIs make use of the commands to invoke the interfaces provided by PN532 firmware via SPI. All these commands are documented in PN532 User Manual. The following APIs are provided in PN532 library.

Datos técnicos de la Shield NFC V1.0 y PN-532DS
Fuente: (Enlace de ARDUINO - Seeedstudio wiki, 2014)

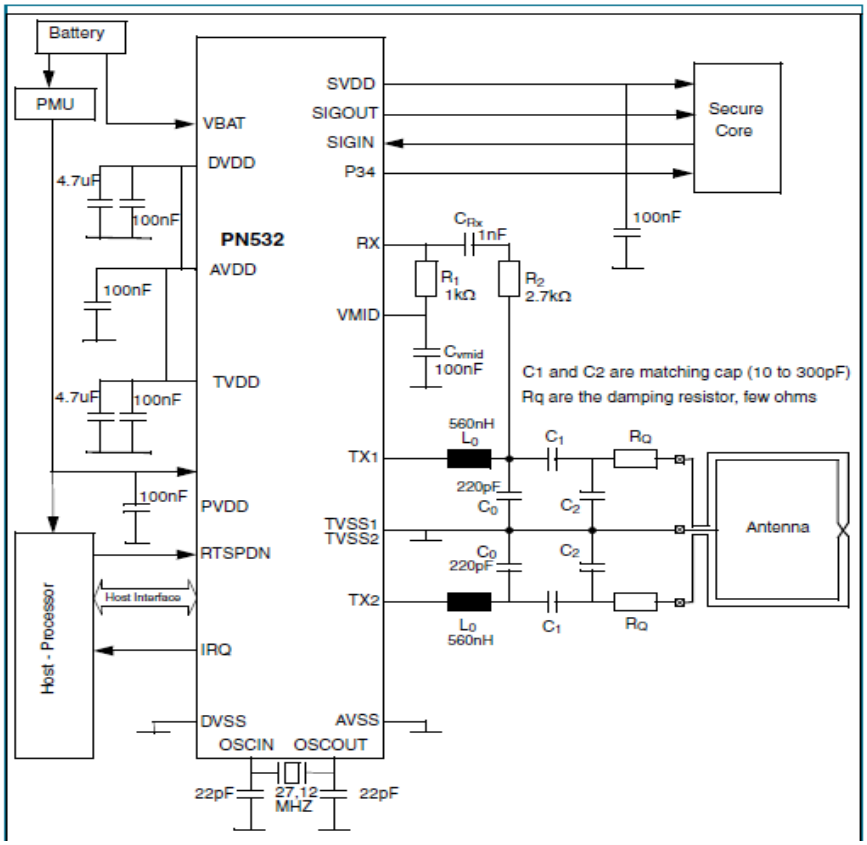


Table 3: PN532 Pin description ...*continued*

Symbol	Pin	Type	Pad Ref Voltage	Description
P31	31	IO	PVDD	General purpose IO signal.Can be configured to act either as TX line of the second serial interface or general purpose IO. In test mode this signal is used as input and output test signal.
P32_INT0	32	IO	PVDD	General purpose IO signal. Can be used to generate an HZ state on the output of the selected interface for the Host communication and to enter PN532 into powerdown mode without reseting the internal state of PN532. In test mode this signal is used as input and output test signal.
P33_INT1	33	IO	PVDD	General purpose IO signal. Can also be used as an interrupt source In test mode this signal is used as input and output test signal.
P34	34	IO	SVDD	General purpose IO signal or clk signal for the SAM
SIGOUT	35	O	SVDD	Contactless communication interface output: delivers a serial data stream according to NFCIP-1 and output signal for the SAM. In test mode this signal is used as test signal output.
SIGIN	36	I	SVDD	Contactless communication interface input: accepts a digital, serial data stream according to NFCIP-1 and input signal from the SAM. In test mode this signal is used as test signal input.
SVDD	37	O		Output power for SAM power supply. Switched on by Firmware with an overload detection. Used as a reference voltage for SAM communication.
RSTPDN	38	I	PVDD	Reset and Power Down: When Low, internal current sources are switched off, the oscillator is inhibited, and the input pads are disconnected from the outside world. With a negative edge on this pin the internal reset phase starts.
DVDD	39	PWR		Internal Digital Power Supply
VBAT	40	PWR		Main external power supply.

Datos técnicos de la Shield NFC V1.0 y PN-532DS
Fuente: (B.V. NXP, 2011)

Table 2. TX framing and TX speed in RFfieldON configuration

TX framing – TX speed	Selection Pins	
	P33_INT1 (pin #33)	P34/SIC_CLK (pin #34)
Mifare - 106 kbps	1	1
	0	0
FeliCa - 212 kbps	0	1
FeliCa - 424 kbps	1	0

2. Configuration Modes

The PN532 has 3 possible modes that can be chosen by using two GPIOs during the reset phase of the IC:

Table 1. Configuration modes

Mode	Selection Pins	
	P70_IRQ (pin #25)	P35 (pin #19)
Standard	1	1
	0	1
PN512 emulation	1	0
RF field ON	0	0

Datos técnicos de la Shield NFC V1.0 y PN-532DS
Fuente: (B.V. NXP, 2011)

ANEXO 5

Datos técnicos del Modem

GSM ZTE MG3006

3 Appearance and framework

Appearance of MG3006 is as following figure 3-1:

Figure 3-1 appearance of MG3006 module

- Dimension（length x width x height）：44.0 mm x 28.0mm x 7.6mm
- Weight：8g

4 Functions and interfaces

The basic functions of module are as below:

- Support Quad Band：*GSM 850/EGSM 900/DCS 1800/PCS 1900*
- Support packet data service
- Support circuit switched data service
- Support SMS service
- Support standard AT commands and extended AT commands
- Support standard UART interface
- Support dual-path audio interface
- Supplementary service functions：incoming call display, call forward, call maintenance, call stand by, triple call service and so on.
- Support TCP/IP protocol

5 Technical specifications

5.1 Communication protocols and technical specifications

The communication protocols and technical specifications of MG3006 modules is as following table 5-1:

Table 5-1 communication protocols and technical specifications

Access mode	GSM
Tech-spec	GSM phase 2/2+
Rx/Tx frequency interval	45MHz for GSM 850 45MHz for EGSM 900 95MHz for DCS 1800 80MHz for PCS 1900
Voice encoding	- Half rate (HR) - Full rate (FR) - Enhanced Full rate (EFR) - Adaptive Multi-Rate (AMR)

- MG3006 frequency band：GSM 850/EGSM 900/DCS 1800/PCS 1900 MHz. There frequency bands are shown in table 5-2. Data transmission rate depends on interval assignment and channel encoding of GPRS.

Table5-2 frequency band

name	Tx frequency band(MHz)	Rx frequency band (MHz)
GSM 850	824~849 MHz	869~894MHz
EGSM 900	880~915 MHz	925~960MHz
DCS 1800	1710~1785MHz	1805~1880MHz
PCS 1900	1850~1910MHz	1930~1990MHz

MG3006 module supports CLASS 10，the interval assignment is as following tableTable5-3:

Table5-3 interval assignment

down	up	Maximum supported interval at the same time
4	2	5

GPRS encoding modes supported by MG3006 module are as following table 5-4:

Datos técnicos del Modem GSM ZTE MG3006

Fuente: (ZTE, 2007)

5.4 Recommendation of antenna specs

The recommended antenna specs are as following table 5-18:

Table5-18 recommended antenna specs

VSWR	1.5:1 maximum
gain	At least 0 dBi in one direction
Input impedance	50Ω
Polarized form	Vertical polarizing

The requirements for antenna's gain are different in different environment. Commonly, in used frequency range, the larger gain, the better capability; otherwise, out of this range, the smaller gain, the better capability.

The antenna seat's type of MG3006 module is MM9329-2700B.

5.5 Power supply

5.5.1 Input voltage

The input voltage is shown in table 5-19:

Table5-19 input voltage

state	Max. voltage	Typical voltage	Min. voltage
Power supply	4.25 VDC	3.90 VDC	3.30 VDC

5.6 Working conditions

- Working temperature:-20℃ ~ +80℃
- Storage temperature:-40℃ ~ +85℃
- humidity：0% ~ 95%

Datos técnicos del Modem GSM ZTE MG3006

Fuente: (ZTE, 2007)

2.5.3 +CMGF: set SMS mode

Description	This command is used to set SMS input mode.	
Format	AT+CMGF=< num>	
Example	AT+CMGF=1 OK AT+CMGF? +CMGF:1 AT+CMGF=? +CMGF=(0-1)	OK Set SMS input mode as text input Query current input mode setting Current setting as text mode Query current setting range
Parameters	0: PDU mode; 1: Text mode.	

2.5.4 +CNMI: set SMS indicator format

Description	This command is used to set SMS indicator format.
Format	AT+CNMI=<mode>,<mt>,<bm>,<ds>,<bfr>

Example	AT+CNMI=? +CNMI: (0-3),(0-3),(0,2,3),(0-1),(0) OK	Query the range for current settings
	AT+CNMI=3,1,0,0,0 OK +CMTI: "SM",19	Set SMS receiving mode as +CMTI: men, index format Receive new messages
	AT+CNMI=3,2,0,0,0 OK AT+CMGF=1 OK +CMT: "+86130********","","07/02/14,10:29:04+32" text	Set SMS receiving mode Set current setting as Text Mode Receive SMS text from 130********
Returned Results	+CMTI:<mem>,<index>: indicate receipt of new message. +CMT:,<length><CR><LF><pdu>: directly output received message (PDU mode). +CBM:<length><CR><LF><pdu>: directly output cell broadcast info (PDU mode).	

Datos técnicos del Modem GSM ZTE MG3006

Fuente: (ZTE, 2007)

ANEXO 6

Certificado de la Cooperativa

de Taxis Agua Clara