UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INALÁMBRICO DE CONTROL DE ACCESO PARA PASAJEROS UTILIZANDO TECNOLOGÍAS RFID/NFC Y WI-FI

T E S I S

Que para obtener el título de

INGENIERO EN TELECOMUNICACIONES

PRESENTA

RICARDO AXEL PAREDES VÁZQUEZ

DIRECTOR DE TESIS: M.I. MARIO ALFREDO IBARRA CARRILLO

Ciudad Universitaria, Cd. Mx., 2017

"En la era de las telecomunicaciones, el que mejor comunica sigue siendo quien es capaz de mirar a los demás a los ojos"

Rafael Vidac

"La vida es y siempre seguirá siendo una ecuación incapaz de resolver, pero tiene ciertos factores que conocemos"

Nikola Tesla

Índice

Acrónimos

5C: 5 Criteria.
AIDC: Automatic Identification and Data Capture.
AIEE: American Institute of Electrical Engineers.
AM: Amplitude Modulation.
ASK: Amplitude Shift Keying.
AUI: Attachment Unit Interface.
CA: Collision Avoidance.
CD: Collision Detection.
CEPT: European Conference of Postal and Telecommunications Administrations.
CRC: Ciclic Redundancy Check.
CSMA/CD: Carrier Sense Multiple Access with Collision Detection.
DARPA: Defense Advanced Research Projects Agency.
DoD: Department of Defense.
DSSS: Direct Sequence Spread Spectrum.
EAS: Electronic Article Surveillance.
FFT: Fast Fourier Transform.
FHSS: Frequency Hopping Spread Spectrum.
FM: Frequency Modulation.
FSK: Frequency Shift Keying.
HTTP: Hypertext Transfer Protocol.
IAB: Internet Activities Board.
IEEE: Institute of Electrical and Electronics Engineers.
IFF: Identification Friend or Foe.
IRE: Institute of Radio Engineers.
I2C: Inter-Integrated Circuit
ISO: International Organization for Standardization.
LMSC: IEEE 802 LAN/MAN Standards Committee.
LRC: Longitudinal Redundancy Check.
MAC: Media Access Control.
MiFare: Tarjeta inteligente de lectura sin contacto para sistemas RFID\NFC
NDEF: NFC Data Exchange Format.
NFC: Near Field Communication
NeSCom: New Standards Committee.
OCR: Optical Character Recognition.

OFDM:	Orthogonal Frequency Division Multiplexing.
OSI:	Open System Interconnection.
PAR:	Project Authorization Request.
PLC:	Physical Medium Layer Convergence Procedure.
PM:	Phase Modulation.
PMD:	Physical Medium Dependent.
PoE:	Power over Ethernet
PSK:	Phase Shift Keying.
RFID:	Radio Frequency Identification.
RO:	Read-Only.
RTC:	Real Time Clock.
RW:	Read/Write.
SPI:	Serial Peripheral Interface
TCP/IP:	Transmission Control Protocol \Internet Protocol.
UPCC:	Uniform Product Code Council.
UTP:	Unshield Twisted Pair.
WG:	Working Group's.
WORM:	Write-Once-Read-Many.

Introducción

En los últimos años la tecnología ha avanzado de forma muy veloz en todas las áreas, una de las más notables es el área de las telecomunicaciones.

En un principio todas las comunicaciones se efectuaban por medio de una conexión física entre dispositivos, hoy en día muchas comunicaciones se efectúan de manera inalámbrica.

Las vías de comunicación han evolucionado a nuevas formas de interacción, anteriormente para una comunicación punto a punto se utilizaba el teléfono fijo, cartas, telegramas y para una comunicación masiva se utilizaba la radio y la televisión. Hoy en día se utiliza el celular, mensajes de texto, correos electrónicos, redes sociales y el internet.

Estas nuevas tecnologías inalámbricas han dotado a la sociedad de poder tener acceso a la red de comunicación, el internet o redes privadas desde casi cualquier lugar.

Wi-Fi y Bluetooth son tecnologías de comunicación inalámbrica muy populares hoy en día, pero existen otras tecnologías que comienzan a ser muy populares.

Los sistemas RFID han existido desde la mitad del siglo pasado pero debido a sus altos costos de producción y las limitantes tecnológicas no tenían un impacto tan grande como lo tiene hoy en día. Actualmente la gente utiliza estos sistemas inclusive sin darse cuenta que lo hacen. Se utilizan en los centros comerciales para etiquetar productos, para acceso a casas, oficinas, autopistas, túneles o estacionamientos, en el control de inventarios, entradas y salidas de algún producto, al ingresar al transporte público, para el rastreo de mascotas y personas, entre muchas otras aplicaciones.

Dado que RFID es una tecnología emergente hay muchos detalles técnicos de los que no estamos familiarizados, pero al llegar a comprenderlos podemos lograr una concepción mucha más amplia y nos incita a pensar y diseñar nuevas aplicaciones para el bienestar de la sociedad.

NFC es un subconjunto de RFID cuyo alcance no es mayor a los 10 cm. Esto nos brinda mayor seguridad para aplicaciones de identificación o transacciones monetarias.

Cuando comenzaron a desarrollarse los primeros sistemas de red para interconectar distintos puntos cada empresa, instituto o laboratorio tenía sus propios estándares y protocolos, esto generaba que solo los sistemas fueran compatibles entre los de su misma clase. Para evitar esto y lograr una interconectividad se crearon instituciones dedicadas a regular y estandarizar todas las tecnologías ya existentes y las que estén por venir. Estos estándares van desde la estructura de los sistemas hasta los equipos que se fabrican y distribuyen generando una garantía de operatividad a los usuarios.

La tesis se compone de los siguientes capítulos:

En el capítulo 1 se abordan antecedentes del área de telecomunicaciones y conceptos básicos sobre los elementos de una red para el mejor entendimiento sobre lo que se mencionará en los capítulos posteriores.

El capítulo 2 tratará sobre los estándares IEEE, sus orígenes y su organización. Se profundizara especialmente en los estándares IEEE 802.3 (Ethernet) e IEEE 802.11 (Wi-Fi) describiendo de una forma técnica los principios generales de su funcionamiento.

El capítulo 3 se enfocará en la tecnología RFID/NFC y dado que es la tecnología en la que se enfocará esta investigación se describirá de una forma más amplia la evolución de esta tecnología a la actualidad, los elementos que conforman un sistema RFID/NFC, las capas que hacen posible el funcionamiento del sistema, los protocolos utilizados y se espera lograr un panorama más favorable para esta tecnología emergente.

En el capítulo 4 se mostrara el proceso de diseño de todo el sistema, la recolección de datos por medio de tarjetas RFID/NFC, la programación del microcontrolador, la etapa de comunicación inalámbrica vía Wi-Fi, las pruebas realizadas al sistema y los resultados obtenidos.

Justificación

Una empresa privada de autobuses dedicada al transporte empresarial se acercó al Departamento de Telecomunicaciones de la Facultad de Ingeniería buscando asesoría para un proyecto de control de acceso de pasajeros a los autobuses. Se solicitaba el uso de Smart Cards para Lectores RFID/NFC, almacenamiento de datos en una memoria, la transmisión inalámbrica de datos por medio de Wi-Fi hacia un servidor y acceder a la información desde una terminal principal.

Se fijaron ciertas especificaciones técnicas a seguir así como una lista de material a utilizar. A partir de esto se comenzó el diseño del sistema logrando un prototipo funcional.

Esta investigación está enfocada para diferentes niveles de conocimientos dentro del área de las telecomunicaciones donde se profundiza en conceptos y desarrollos técnicos sobre estándares de tecnologías de vanguardia.

El sentido que se le da a este trabajo es a corto y mediano plazo ya que al finalizar las pruebas de laboratorio se comenzará a utilizar el prototipo en pruebas de campo dando lugar a continuar haciendo mejoras y actualizaciones.

Los alcances para este trabajo son el dar una visión técnica general de los estándares IEEE 802.3 (Ethernet) e IEEE 802.11 (Wi-Fi) y profundizar en aspectos técnicos y generales de las tecnologías RFID/NFC.

Objetivo General

El objetivo principal de esta tesis es generar un prototipo que ayude a administrar el control de aforo de personas en un autobús, el cual sea capaz de obtener y guardar los datos de los pasajeros que ingresen al mismo, mediante tecnología RFID/NFC para posteriormente transmitir la información obtenida (ID del chofer, ID del pasajero, estado del autobús, fecha y hora del abordaje) por medio de una red Wi-Fi.

Objetivos Específicos

- Comprender los estándares para redes alámbricas e inalámbricas.
- Evaluar las diferentes tecnologías disponibles en el mercado, para la realización del prototipo y generar una propuesta.
- Investigar y comprender los lenguajes de programación, así como los protocolos de comunicación que se utilizarán en la programación del prototipo.
- Generar conocimiento propio sobre los sistemas RFID/NFC y exponer a través de este trabajo una breve explicación sobre la funcionalidad de estos sistemas.
- Aplicar los conocimientos sobre redes y sistemas RFID/NFC para la solución a un problema real de Ingeniería.
- Mostrar la importancia de esta tecnología emergente y su prospección a futuro en la vida cotidiana.

Meta

Desarrollo de un prototipo para el control de acceso de pasajeros a autobuses utilizando tecnologías RFID/NFC y Wi-Fi.

Hipótesis

Mediante la metodología expuesta, se darán a conocer los pasos para poder evaluar la implementación de un sistema RFID/NFC para el control de acceso a pasajeros en un autobús, la recolección de datos y el envío de los mismos a través de la tecnología Wi-Fi. Al finalizar las pruebas se valorará el sistema y se dimensionará su viabilidad para su uso en el mercado.

Capítulo 1
Antecedentes

1.1 Redes de datos

Son un conjunto de equipos de comunicaciones, computadoras y dispositivos que pueden comunicarse entre sí a través de un medio particular. [2] [3]

Los objetivos de una red:

- Comunicación.
- Ahorro de tiempo.
- Disponibilidad de información.
- Reducción de costos.
- Compartir recursos (Programas, datos, información, impresoras, módem, etc.)

Podemos clasificar las redes de acuerdo a la tecnología de transmisión que emplean.

1.1.1 Redes de Difusión

Existe solamente un medio de transmisión que comparten todos los dispositivos de la red, se envían datos a todos los destinos con un código de dirección, al transmitir cada máquina de la red lo recibe y lo procesa.

1.1.2 Redes de Conmutación de Paquetes

Consisten en una serie de conexiones entre pares individuales de máquinas. Para enviar un paquete del origen al destino debe de pasar por una o más máquinas intermedias donde los algoritmos de ruteo toman el control. Este tipo de redes serán utilizadas para nuestro prototipo.

1.2 Medio de transmisión

Es necesario definir la forma en la que se va a conducir una señal o la información hacia su destino, por lo que podemos clasificarlo en dos grupos. Para la realización de nuestro sistema, se utilizarán ambos medios.

1.2.1 Medios de transmisión guiados

Son aquellos que tienen una interacción directa entre equipos, los más utilizados en las redes de datos son el cable coaxial grueso y delgado, par trenzado y fibra óptica.

1.2.2 Medios de transmisión no guiados (Inalámbricos)

Utilizar medios no guiados o inalámbricos nos brindan la posibilidad de movilidad sin perder la conexión; evitan la instalación de cable, canaletas, rosetas o alteraciones en la estructura, lo cual reduce el costo de la red.

1.3 Redes alámbricas

Utilizan como medio de transmisión distintos tipos de cables ya sea de cobre o de fibra óptica para comunicarse a otras redes o para unir sus nodos.

Se clasifican de la siguiente forma de acuerdo a su alcance.

1.3.1 Redes LAN (Local Area Network)

Se define como un sistema de comunicaciones que proporciona interconexión a varios dispositivos en un área restringida y no utiliza medios de telecomunicaciones externos. Se caracteriza por tres factores:

- Extensión: Algunos metros o incluso Kilómetros.
- Tecnología de transmisión: Cable de par trenzado UTP o coaxial, fibra óptica, infrarrojo, laser, radiofrecuencia y microondas.
- Topología: Anillo, bus, estrella, árbol e híbridas.

1.3.2 Redes MAN (Metropolitan Area Network)

Pueden ser consideradas como una red LAN en cuanto a su topología y medios de transmisión, pero de mayor alcance y más robusta.

Este tipo de redes pueden llegar a tener un alcance de 10 Km. hasta el abarcar toda una ciudad. Una red de datos, voz o video con una extensión de más de una decena de Km. puede considerarse una MAN.

1.3.3 WAN (Wide Area Network)

Es una red de comunicación de datos que tiene una cobertura geográfica muy grande y utiliza instalaciones de transmisión que ofrecen compañías portadoras de servicios como las telefónicas. Esta une nodos de usuarios llamada subred y abarca diversos equipos de red como pueden ser routers y líneas de comunicación que unen las redes. Su alcance puede ir desde todo un país hasta un continente.

1.4 Redes Inalámbricas

Nos brindan la posibilidad de conexión a redes públicas o privadas sin la necesidad de cables. Ofrecen la posibilidad de navegar por Internet, conectividad a la red privada de una empresa, consultar el correo electrónico y las redes sociales desde la escuela, aeropuertos, hoteles o en algunos espacios públicos. Las redes inalámbricas nos brindan la posibilidad de interactuar en casi todo momento y a cualquier hora.

Se clasifican de la siguiente forma de acuerdo a su alcance:

1.4.1 W-PAN (Wireless – Personal Area Network): Bluetooth

La tecnología Bluetooth es utilizada para las redes de área personal en dispositivos desde audífonos, bocinas, celulares, pantallas, automóviles y computadoras.

Otra tecnología que entra en esta categoría es RFID/NFC que permite una comunicación con un alcance aproximado de 10 cm. Utilizada principalmente para transmitir paquetes pequeños de información que son ligados principalmente a ID´s o números de identificación.

1.4.2 W-LAN (Wireless – Local Area Network)

El espacio de una red LAN abarca cualquier entorno de recintos como edificios, oficinas, escuelas, aeropuertos, hoteles o cualquier espacio público. Los estándares de la IEEE para esta tecnología inalámbrica son los 802.11.

1.4.3 W-MAN (Wireless – Metropolitan Area Network)

Estas redes se sitúan en rangos de cobertura de pocos kilómetros como conjuntos de fraccionamientos, colonias, pueblos o municipios pequeños. Las tecnologías de este grupo se conocen como Inalámbricas de Banda Ancha y una de las más conocidas es WiMax.

1.4.4 W-WAN (Wireless – Wide Area Network)

Es espacio de las redes WAN se mide por kilómetros y funciona en casi cualquier parte. Su conectividad es a través de un teléfono o una PC Card de una computadora.

Este tipo de redes utilizan tecnologías de red celular de comunicaciones móviles como GPRS, CDMA, GSM, 3G y 4G LTE logrando con esta última tener velocidades de transmisión de hasta 100 Mbit/s.

1.5 Modelo OSI

El modelo OSI[1] estandariza la comunicación entre sistemas en una arquitectura de red, brinda las funciones necesarias para la comunicación y las divide en siete capas de función simplificando todo en pequeñas unidades lógicas. Esto nos brinda una ventaja al momento de la comunicación debido a que entre capas es posible intercambiar la implementación técnica de una capa independiente desde otras capas. [2] [3]

1.5.1 Capas del Modelo OSI

Las primeras cuatro capas son llamadas "inferiores" y están orientadas a red. Las siguientes son llamadas "superiores" y están orientadas a la aplicación. Cada una de las capas inferiores provee sus servicios a la capa superior por medio de interfaces.

Capa 1 – Capa Física:

Representa una transferencia a través de un medio físico, donde los datos serán transmitidos bit por bit. Establece características mecánicas, eléctricas y de procedimiento para activar, desactivar y mantener conexiones entre dos equipos. En esta capa están incluidos los cables, la distribución de terminales, los niveles de voltaje y velocidad de transmisión.

Capa 2 – Capa de Enlace de Datos:

Maneja el acceso correcto al medio de transmisión y proporciona confiabilidad en el medio para la transmisión de datos que comparten el medio de comunicación. Controla el flujo de datos, realiza la detección de errores, recupera la trama y al haber un error hace una retransmisión de esta.

Capa 3 – Capa de Red:

Se utiliza el direccionamiento lógico para identificar un host destino en la transmisión de datos. Determina mecanismos de enrutamiento para poder transportar la información entre cualquier punto de una red, por lo que desempeña funciones de segmentación, de secuenciación, de encaminamiento, de control de flujo y de retransmisión.

[1] OSI: Open System Interconnection, "Sistema Abierto de Interconexión".

Capa 4 – Capa de Transporte:

Es responsable de que la transmisión esté libre de errores y que la secuencia de transmisión de datos sea compatible entre los dispositivos terminales. Determina el control de flujo en la transmisión, en la retransmisión y en el multiplexaje.

Capa 5 – Capa de Sesión:

Se dedica a establecer, liberar y terminar las conexiones entre los dispositivos, así como, permitir una organización y sincronización del diálogo entre ellas para intercambiar datos.

Capa 6 – Capa de Presentación:

Los datos al ser transmitidos son convertidos en un formato común y cambiados del lado del receptor en la sintaxis necesaria para él.

Capa 7 – Capa de Aplicación:

Se encarga de la interfaz de las aplicaciones con el usuario destino. Es el único medio para que el proceso de aplicación tenga acceso al entorno OSI debido a que es la capa de más alto nivel. Dentro de sus funciones se encuentra la transferencia de archivos, el correo electrónico, la búsqueda en directorios, el software de terminales virtuales y todo lo que no es realizado por las capas inferiores como aquellas funciones ejecutadas por programas o usuarios.

Fig. 1.1 Modelo OSI

1.6 Modelo TCP/IP

Este modelo fue desarrollado en una red de investigación de conmutación de paquetes nombrada ARPANET, una división de DARPA[2] a cargo del Departamento de Defensa de los Estados Unidos. [3]

Al no estar listo el modelo OSI, el Departamento de Defensa de los Estados Unidos tuvo que utilizar por la disponibilidad el modelo TCP/IP[3] sobre el modelo OSI y dio el inicio para fabricantes a desarrollar productos basándose en este modelo que actualmente es la base para el desarrollo de nuevos protocolos y estándares.

1.6.1 Arquitectura del protocolo TCP/IP

El protocolo TCP/IP es un conjunto de protocolos que se publicaron como estándares de internet por la IAB[4] que forma parte de la Sociedad de Internet encargada del desarrollo técnico del internet y la evolución del TCP/IP.

Su objetivo principal es soportar diferentes tipos de servicios e interconectar diferentes redes sin que la comunicación se pierda si un tramo de la subred llega a fallar. Cada una de las capas utiliza información de control para la transferencia de datos.

Otra característica es que permite conectar máquinas de diferentes tipos y con diferentes sistemas operativos en redes LAN y WAN.

No se reconoce como un modelo de referencia como el caso del modelo OSI debido a que primero se utilizó y después se estandarizó.

1.6.2 Capas de modelo TCP/IP

Al no tener un modelo de referencia oficial TCP/IP todas las funciones realizadas por el sistema de comunicación pueden ser clasificadas en cuatro capas, ya que este modelo es muy general y no hace una clara diferencia entre el servicio, interfaz y protocolo.

Se dice que está compuesto por cuatro capas:

- Capa de Aplicación.
- Capa de Transporte.
- Capa Internet.
- Capa de acceso a red.

[2] DARPA: *Defense Advanced Research Projects Agency*, “Agencia de Proyectos de Investigación Avanzada para la Defensa”.
[3] TCP/IP: *Transmission Control Protocol*, “Protocolo de Control de Transmisión” e *Internet Protocol*, “Protocolo de Internet”.
[4] IAB: *Internet Activities Board*, “Junta de Actividades de Internet”.

1.6.3 Capa de acceso a red.

Es la responsable de llevar y recibir datos de la red. Corresponde a la capa física y de enlace del modelo OSI. Se encarga de poner la información en el medio de comunicación y señala las características físicas del medio de transmisión, la naturaleza de las señales, la velocidad de los datos, entre otras. No incorpora funciones de control de flujo y errores.

Modelo TCP/IP
Aplicación
Transporte (origen-destino)
Internet
Acceso a la Red
Física

Fig. 1.2 Modelo TCP/IP

En la figura (Fig. 1.3) se muestra la funcionalidad de las capas del modelo TCP/IP con el OSI.

<table>
<tr><th>OSI</th><th>TCP/IP</th></tr>
<tr><td>Aplicación</td><td rowspan="3">Aplicación</td></tr>
<tr><td>Presentación</td></tr>
<tr><td rowspan="2">Sesión</td></tr>
<tr><td rowspan="2">Transporte</td><td rowspan="2">Transporte (origen-destino)</td></tr>
<tr><td rowspan="2">Red</td></tr>
<tr><td>Internet</td></tr>
<tr><td rowspan="2">Enlace de Datos</td><td rowspan="2">Acceso a la Red</td></tr>
<tr><td>Física</td><td>Física</td></tr>
</table>

Fig. 1.3 Comparación del Modelo OSI y el Modelo TCP/IP

1.7 Dispositivos de una red

1.7.1 Switch

También es conocido como “conmutador”, es un dispositivo que trabaja en la capa de enlace del modelo OSI, se le denomina un dispositivo inteligente ya que envía el paquete únicamente al host destino y no a todos los host como lo haría el Hub.

Realiza sus funciones mediante una matriz conmutada permitiendo la transmisión simultánea entre dos host, al igual que el bridge su principal función es segmentar dominios de colisión utilizando un ancho de banda dedicado para cada host conectado a uno de sus puertos limitando el tráfico únicamente al segmento que pertenece el frame y así aumentar la velocidad de transmisión.

Un Switch es considerado un bridge multipuerto, donde a diferencia del bridge, el switch al recibir el encabezado del frame inicia la transmisión y no necesita recibir toda la frame como lo hace el bridge.

1.7.2 Router

Este dispositivo trabaja sobre la capa de red del modelo OSI y capas inferiores, se utiliza para segmentar una red limitando el tráfico de broadcast, proporciona seguridad y control al dominio de broadcast utilizando filtros de paquetes en ambiente LAN y WAN. Otra de sus funciones principales es poder soportar rutas redundantes en la red y soportar diferentes tecnologías de enlace de datos como: Ethernet, Fast Ethernet, Token Ring, entre otros. El Router también se utiliza como firewall.

Al trabajar en la capa 3 tiene acceso a la dirección lógica de cada host utilizando algoritmos de enrutamiento para elegir la mejor trayectoria para transmitir un paquete.

Para la elección de dichas trayectorias realiza tablas de ruteo considerando factores como: el número de saltos, condiciones de tráfico, velocidad de línea y retraso; la desventaja es que el procesamiento adicional de paquete por el Router incrementa el tiempo de espera o reduce el desempeño del mismo al comparándolo con un Switch, pero a diferencia de este, el Router puede distinguir entre diferentes protocolos de red.

Las tablas de ruteo pueden ser creadas estáticamente o dinámicamente, las primeras necesitan que el administrador las cree manualmente y no son actualizadas automáticamente cuando se producen cambios en la red como lo hacen las tablas dinámicas.

1.7.3 Gateway

Es un dispositivo utilizado para realizar la conversión de protocolos entre sistemas de comunicación incompatibles. Se puede considerar al Gateway como un servidor que actúa como único para el cliente representando otros servidores que no son capaces de comunicarse directamente con el cliente.

1.8 Dirección IP

Dirección de 32 bits asignada a los hosts mediante TCP/IP. Una dirección IP corresponde a una de cinco clases (A, B, C, D o E) y se escribe en forma de 4 octetos separados por puntos (formato decimal con punto). Cada dirección consta de un número de red, un número opcional de subred, y un número de host. Los números de red y de subred se utilizan conjuntamente para el enrutamiento, mientras que el número de host se utiliza para el direccionamiento a un host individual dentro de la red o de la subred. Se utiliza una máscara de subred para extraer la información de la red y de la subred de la dirección IP. También denominada dirección de Internet (dirección IP).

1.9 Protocolos

Para que dos dispositivos se comuniquen es necesario que exista una serie de acuerdos entre ellos para establecer una comunicación, al conjunto de normas que deben seguir los dos host para que se establezca una comunicación desde que inicia hasta que termina de le llama protocolo.

Capítulo 2

Ethernet (IEEE 802.3) y Wi-Fi (IEEE 0802.11)

2.1 Estándares IEEE 802

En el año de 1963 el AIEE[5] y el IRE[6] se convirtieron en la IEEE[7] quien actualmente es la autoridad en estándares, investigación e intercambio científico en diversas áreas como la aeroespacial, computación, telecomunicaciones, ingeniería biomédica, energía eléctrica, consumidores electrónicas y muchas más. [1]

Dentro de la IEEE se encuentra el LMSC[8] y en diciembre de 1979 se propuso el inicio del Proyecto "IEEE 802" que alentó a la LMSC a desarrollar los estándares de las LAN y MAN que fue probada el 13 de Marzo de 1980. Actualmente hay 22 WG´s [9] en IEEE 802 considerando los dos niveles más bajos del modelo ISO/OSI como referencia.

La LMSC desarrolla y mantiene estándares sobre creación de redes y prácticas recomendadas para redes locales, metropolitanas y otras, utilizando procesos abiertos y acreditados a nivel mundial. Un WG se dedica específicamente a un área.

2.1.1 IEEE 802 Working Group´s activos

- **802.1** Normalización de interfaz
- **802.3** CSMA/CD (Ethernet)
- **802.11** Redes inalámbricas WLAN (Wi-Fi)
- **802.15** WPAN (Bluetooth) y ZigBee
- **802.16** Redes de acceso metropolitanas sin hilos de banda ancha (WIMAX)
- **802.18** Asesoría Técnica sobre Normativas de Radio
- **802.19** Asesoría Técnica sobre Coexistencia
- **802.21** Rechazo/interoperatividad entre redes
- **802.22** Redes de Área Regional Inalámbricas

2.1.2 IEEE 802 Working Group´s inactivos

Sus estándares están publicados pero ya no trabajan en ellos ni sostienen reuniones hasta que sean nuevamente reactivados:

- **802.2** Control de enlace lógico LLC
- **802.5** Token Ring
- **802.12** Acceso de Prioridad por demanda 100 Base VG-Any Lan

[5] AIEE: *American Institute of Electrical Engineers*, "Instituto Americano de Ingenieros Eléctricos".
[6] IRE: *Institute of Radio Engineers*, "Instituto de Ingenieros de Radio".
[7] IEEE: *Institute of Electrical and Electronics Engineers*, "Instituto de Ingenieros Eléctricos y Electrónicos".
[8] LMSC: *IEEE 802 LAN/MAN Standards Committee,* "Comité de estandares para IEEE 802 LAN/MAN".
[9] WG: *Working Group's,* "Grupos de Trabajo".

2.1.3 IEEE 802 Working Group´s disueltos

Se consideran disueltos debido a que no publicaron estándares o sus estándares fueron retirados.

- **802.4** Token Bus
- **802.6** Redes de Área Metropolitana (MAN) (ciudad) (fibra óptica)
- **802.7** Grupo Asesor en Banda ancha
- **802.8** Grupo Asesor en Fibras Ópticas
- **802.9** Servicios Integrados de red de Área Local (redes con voz y datos integrados)
- **802.10** Seguridad de Red
- **802.14** Módems de cable

Cada Work Group puede tener subgrupos a los que se les denomina *Task Groups*. Los Work Group´s participantes en la IEEE 802 deben iniciar la formación de un *Study Group*, para el estudio de nuevos problemas y futuras extensiones. Similar a los Task Group´s, el tiempo de vida de un Study Group es limitado. Su principal tarea es el desarrollo de 2 documentos, el PAR[10] y los 5C[11] , los cuales se utilizan para proponer el establecimiento de un nuevo *Task Group*. Los documentos son revisados por el NeSCom[12], siendo un organismo independiente a las 25 sociedades y consejos de la IEEE.

[10] PAR: *Project Authorization Request*, "Proyecto de Autorización de Solicitud".
[11] 5C: *5 Criteria*, "5 Criterios".
[12] NeSCom: *New Standards Committee*, "Comité de Nuevos Estándares".

2.2 Ethernet (IEEE 802.3)

En 1972 inició el desarrollo de una tecnología para redes conocida como Ethernet Experimental. El primer sistema Ethernet fue conocido como ALTO ALOHA, siendo ésta la primera red de área local (LAN) para computadoras. En mayo de 1973 funcionó por primera vez a una velocidad de 2.94 Mb/s. Las especificaciones formales para Ethernet de 10 Mb/s se desarrollaron en conjunto con las corporaciones DEC, Xerox e Intel publicándolas en el año de 1980 conocidas como el estándar DEC-Intel-Xerox (DIX) o el "Libro azul de Ethernet" haciendo Ethernet experimental a 10 Mb/s un estándar abierto. [4]

La tecnología Ethernet fue adoptada para su estandarización por el comité de redes locales (LAN) de la IEEE como el estándar IEEE 802.3 publicado por primera vez en 1985. Ethernet fue adoptado por la ISO[13] haciéndolo el estándar de redes internacionales

2.2.1 Tipos de Ethernet

Hoy en día podemos encontrar una gran variedad de implementaciones de IEEE 802.3. Para poder distinguirlas, se ha desarrollado una notación específica mediante las características de implementación.

- Método de señalamiento utilizado.
- Máxima longitud de segmento de cable en cientos de metros.
- La taza de transferencia de datos en Mb/s.

2.2.2 Estándares para Ethernet

Las redes originales de Ethernet trabajan a 10 Mbps tomando como referencia los estándares de la IEEE. Tomando de ejemplo 10BASE-T el número 10 indica la velocidad de la red, BASE indica el tipo de operación en este caso Banda base; es decir, la señal no está modulada. El último parámetro señala la distancia máxima de acuerdo al medio utilizado.

1BASE-5
El estándar IEEE para Ethernet en banda base a 1Mbps sobre cable trenzado a una distancia máxima de 250m.

10BASE-5
El estándar IEEE para Ethernet en banda base a 10Mbps sobre cable coaxial de 50 ohms troncal y AUI [14] de cable sobre par trenzado a una distancia máxima de 500m.

[13] ISO: *International Organization for Standardization,* "Organización Internacional de Estandarización".
[14] AUI: *Attachment Unit Interface, "Unidad Adjunta de Interfaz".*

10-base-2
El estándar IEEE para Ethernet en banda base a 10Mbps sobre cable coaxial delgado de 50 Ω con una distancia máxima de 185m.

10BROAD-36
El estándar IEEE para Ethernet en banda ancha a 10Mbps sobre cable coaxial de banda ancha de 75 Ω con una distancia máxima de 3600m.

10BASE-T
El estándar IEEE para Ethernet en banda base a 10Mbps sobre par trenzado sin blindaje UTP[15] con una distancia máxima de 100m desde una estación a un repetidor.

10BASE-F
El estándar IEEE para Ethernet en banda base a 10Mbps sobre fibra óptica con una distancia máxima de 2000m. (2 Km.)

2.2.3 Estándares para Fast Ethernet

Las redes Fast Ethernet si las comparamos con las redes Ethernet, tienen una velocidad 10 veces mayor. Las redes Fast Ethernet se conectan a 100 Mbps por lo que se reduce el tiempo de transmisión. También permite conectar dos Hub entre la ruta de comunicación de dos nodos y la conexión de 7 Switches entre la ruta de dicha comunicación de dos nodos.

100BASE-TX
El estándar IEEE para Ethernet en banda base a 100Mbps sobre 2 pares de cable UTP (de categoría 5 o superior) o dos pares de STP.

100BASE-T4
El estándar IEEE para Ethernet en banda base a 100Mbps sobre 4 pares de cable UTP (de categoría 3 o superior)

2.2.4 Estándares para Gigabit Ethernet

Gigabit Ethernet es una opción conveniente para la migración de redes LAN de alta velocidad, este estándar sigue utilizando el protocolo de acceso al medio CSMA/CD. LA velocidad de operación para estas redes es de 1000 Gbps. Sólo permite utilizar un Hub para unir dos segmentos.

1000BASE-SX
El estándar IEEE para Ethernet en banda base a 1000Mbps (1GBps) sobre 2 fibras multimodo de fibra óptica (50/125µm. o 62.5/125µm.)

1000BASE-T
El estándar IEEE para Ethernet en banda base a 1000Mb/s (1GBps) sobre 4 pares UTP (de categoría 5 o superior) con una distancia máxima de cableado de 100m.

[15] UTP: *Unshield Twisted Pair,* "Par Trenzado sin Blindaje".

2.2.5 Ethernet en el modelo OSI

Ethernet opera en dos capas del modelo OSI, en la capa física y la mitad inferior de la capa de enlace de datos también conocida como subcapa MAC.

En la Capa 1 se incluyen las interfaces con los medios, señalamientos, corrientes di bits transportadas en el medio, componentes que transmiten la señal a los medios y las topologías. Esta capa tiene un papel muy importante en la comunicación que se produce entre los dispositivos, pero cada una de estas funciones tiene limitaciones.

La Capa 2 se encarga de estas limitaciones, cuando la capa 1 no puede comunicarse con las capas superiores la Capa 2 lo hace con el Control de Enlace Lógico (LLC), utiliza un proceso de direccionamiento cuando no se puede identificar un dispositivo y aquí es donde se utiliza la dirección MAC para poder identificar al dispositivo que está transmitiendo.

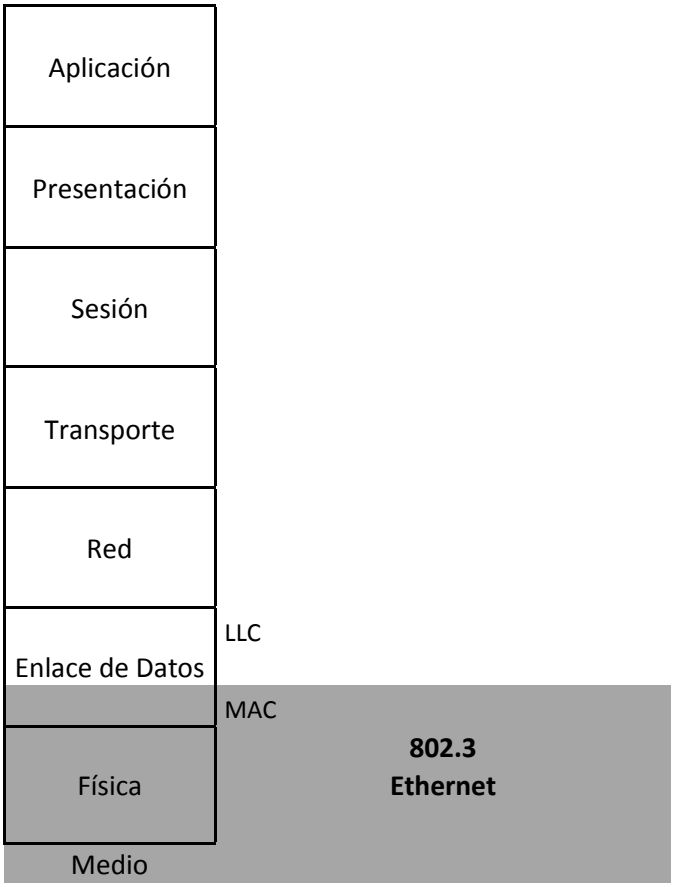


Fig. 2.1 Alcance del Estándar IEEE 802.3 dentro del modelo OSI

2.2.6 Elementos Básicos de Ethernet

Los sistemas Ethernet incluyen cuatro bloques de construcción que al combinarse hacen funcional Ethernet.

- **La Trama:** Es una serie de bits estandarizados utilizada para llevar los datos en el sistema.

- **La Señalización:** Consiste en dispositivos electrónicos estandarizados que mandan y reciben señales sobre un canal Ethernet.

- **El Medio Físico***:* Consiste en cables y otros elementos de Hardware usados para llevar las señales digitales de Ethernet entre computadoras de la misma red.

- **El Protocolo de Control de Acceso al Medio:** Consiste en una serie de reglas incrustadas en cada una de las interfaces Ethernet que permiten el acceso a múltiples computadoras compartiendo el canal Ethernet en una manera justa y controlada.

64 bits	48 bits	48 bits	16 bits	46 a 1500 bytes	32 bits
Preámbulo	Dirección de destino	Dirección de origen	Tipo/ Longitud	Datos	Frame Check Secuence (CRC)

Fig. 2.2 Trama Ethernet a nivel de Bits [Imagen Propia]

2.2.7 Detección de portadora con acceso múltiple y detección de colisiones CSMA/CD

Las señales Ethernet son transmitidas a cada uno de los host conectados a la LN mediante un conjunto especial de reglas que determinan cuál de todas las estaciones pueden transmitir en un momento.

Las redes Ethernet administran las señales de una red por medio del CSMA/CD[16] donde antes de transmitir, primero se escucha al medio de red. Si el medio está inactivo o en "silencio" se envían los datos. Después transmitir, los elementos de la red compiten por el siguiente tiempo de inactividad disponible para poder enviar otra trama. La competencia sobre el tiempo de inactividad denota que no existe una ventaja sobre algún otro elemento de la red.

Una colisión se produce cuando dos estaciones escuchan el medio para saber si hay tráfico en la red, si no lo detectan transmiten de forma simultánea. En este caso ambas transmisiones se dañan y se deben volver a transmitir cuando el canal este inactivo. Estas estaciones CSMA/CD tienen la capacidad para detectar las colisiones y saber que deben retransmitir.

Una estación al transmitir la señal analógica utilizada para la transmisión de la información definida como una "portadora". Si no hay una portadora, una estación en espera sabe que está libre para transmitir. Esto se le conoce como "Detección de portadora"

La posibilidad para detectar errores está limitada en cuanto a distancia en CSMA/CD debido a la atenuación, el debilitamiento de una señal transmitida a medida que se aleja del origen, el mecanismo deja ser efectivo a partir de 2,500 metros ya que los segmentos no pueden detectar señales a partir de esa distancia, por lo que no se puede asegurar que se esté transmitiendo.

A mayor cantidad de elementos de la red se genera mayor tráfico de red. A medida que aumenta el tráfico, aumentan las colisiones, haciendo más lenta la red convirtiendo el CSMA/CD en un método de acceso lento.

16 CSMA/CD: Carrier Sense Multiple Access with Collision Detection, "Acceso Múltiple con Escucha de Portadora y Detección de Colisiones".

2.2.8 Direccionamiento de la trama Ethernet.

Los bits transmitidos a través del protocolo Ethernet están organizados en tramas que contienen información del encabezado, direcciones origen/destino y los datos que se están transmitiendo.

Las comunicaciones en una red ocurren de tres maneras: Unicast, Broadcast y Multicast.

Unicast: Es la comunicación donde una trama es enviada desde un host y es direccionada hacia un destino específico. En una transmisión unicast, solo existe un emisor y un receptor.

Broadcast: Es la comunicación en la que una trama se envía desde una dirección a todas las demás direcciones. En este caso, existe un solo emisor que envía toda la información a todos los receptores conectados. Esta transmisión es esencial cuando se envía el mismo mensaje a todos los dispositivos de la red.

Multicast: Es la comunicación que envía información a un grupo específico de dispositivos o clientes. A diferencia de la transmisión broadcast, en la transmisión multicast los clientes deben ser miembros del grupo multicast para recibir la información.

2.2.9 Direcciones de Ethernet (MAC)

MAC[17] es un identificador hexadecimal de 48 bits organizada en pares o cuartetos que corresponden de forma única con una tarjeta o interfaz de red.

La dirección utilizada en una red Ethernet es el medio por el cual se dirigen los datos a la ubicación receptora correcta: dentro de la tarjeta de interfaz de red (NIC) se encuentra la dirección MAC. La subcapa MAC maneja los problemas de asignación de direcciones físicas, y la dirección física en un número en formato hexadecimal.

Cada dispositivo conectado en red debe tener una dirección MAC única para participar en la red. La dirección MAC identifica la ubicación de un dispositivo específico en una red. A diferencia con otros tipos de direcciones utilizadas para redes, la dirección MAC no se debe cambiar a menos que se requiera para una necesidad específica.

[17] MAC: *Media Access Control,* "Control de Acceso al Medio".

2.3 Wi-Fi (IEEE 802.11)

El origen de esta tecnología surgió por la necesidad de establecer mecanismos de conexión inalámbricos compatibles entre distintos tipos de dispositivos. En 1999 empresas como 3Com, Airones, Nokia, Lucent Technologies, Intersil y Symbol Technologies crearon la Wireless Ethernet Compatibility Alliance o WECA, actualmente llamada Wi-Fi Alliance cuyo objetivo fue designar una marca que fomentara fácilmente la tecnología inalámbrica y asegurar la compatibilidad entre equipos. [B]

Para Abril de 2000 WECA logra certificar la interoperabilidad de equipos con la norma IEEE 802.11b, bajo la marca Wi-Fi, con lo que los usuarios ya podían tener garantía de que todos los equipos con el sello Wi-Fi pueden conectarse y operar entre ellos sin problemas.

El término "Wi-Fi" surge de la marca comercial Wi-Fi de la Wi-Fi Alliance quien contrató una empresa publicitaria para nombrar su estándar de manera que fuera sencillo de entender y recordar.

Interbrand fue la empresa de publicidad que dio el nombre "Wi-Fi" y diseñó el "Style logo" del Yin Yang logrando así un nombre más llamativo y corto a diferencia de "IEEE 802.11 de Secuencia Directa" como originalmente fue nombrado.

Actualmente mucha gente confunde el significado del término Wi-Fi y creen que significa "Wireless Fidelity" debido a la similitud con el término Hi-Fi que quiere decir "High Fidelity", pero realmente no tiene un significado.

Por otro lado muchos asocian Wi-Fi con Internet erróneamente pensando que una conexión Wi-Fi es portadora de la señal de internet por sí sola, cuando Wi-Fi es simplemente un protocolo de interconexión inalámbrico entre dispositivos.

Las redes Wi-Fi emplean ondas de radio para conectar dispositivos entre ellos, a una red o a internet.

La señal de un Router de acceso Wi-Fi tiene un alcance promedio de 90 metros y mayormente se emplea para tener acceso a internet. Comenzó a utilizarse en lugares públicos como hoteles, cafés, aeropuertos y universidades. Posteriormente muchas empresas comenzaron a emplear redes Wi-Fi en sus edificios y oficinas para empleados e invitados, actualmente debido al auge de los dispositivos móviles la mayoría de los hogares cuentan con redes Wi-Fi.

Fig. 2.3 Logotipo de Wi-Fi [B]

Este tipo de redes puede extenderse a través de la colocación de puntos de acceso Wi-Fi adicionales en distintos puntos para ampliar el alcance y la potencia de la señal.

Los primeros equipos que se conectaron a las redes Wi-Fi necesitaban una tarjeta adaptadora de red inalámbrica que con el tiempo tuvo un bajo costo y de fácil instalación. En la actualidad todas las computadoras cuentan con estas tarjetas y son pocos los dispositivos móviles que no tienen integrados funciones Wi-Fi.

2.3.1 Wi-Fi y el estándar IEEE 802.11

La IEEE en 1997 agregó un nuevo miembro a la familia 802 que se ocuparía de definir las redes de área local inalámbricas denominándolo 802.11. [5]

La norma IEEE 802.11 se diseñó como un sustituto equivalente a la capa física y MAC de la norma 802.3 (Ethernet). Lo que diferencia una red Ethernet de una red Wi-Fi es la forma en la que se transmiten las tramas o paquetes de datos, por lo demás son idénticas. Esto logró que ambas fueran compatibles en todos sus servicios.

La primera norma 802.11 utilizó infrarrojos como medio de transmisión pero no tuvo una buena aceptación en el mercado, posteriormente se desarrollaron dos nuevas normas 802.11 que utilizaban la radiofrecuencia en la banda de 2.4 GHz y 5 GHz.

La familia 802.11 se fue fortaleciendo y aparecieron nuevas versiones del estándar, no todas están abiertas al uso comercial, algunas son experimentales o para mejoras y mantenimiento.

A continuación se mencionan las más populares.

- **IEEE 802.11 a.** Fue aprobada en 1999. Utiliza los mismos protocolos de base que el estándar original, opera en la banda de 5 GHz a una velocidad máxima de 54 Mb/s, tiene 12 canales, 8 para red inalámbrica y 4 para conexiones punto a punto.

- **IEEE 802.11 b.** Es una revisión del estándar original, fue ratificada en 1999. Trabaja en la banda de 2.4 GHz, tiene una velocidad máxima de transmisión de 11 Mb/s y utiliza el mismo método de acceso definido en el estándar original CSMA/CA.

- **IEEE 802.11 g.** En Junio de 2003 se ratificó un tercer estándar de modulación que es la evolución del 802.11 b. Utiliza la banda de 2.4 GHz y opera a una velocidad teórica máxima de 54 Mb/s que en velocidad real el promedio seria de 22 Mb/s.

- **IEEE 802.11 n.** Se dio a conocer en enero de 2004. Opera a una velocidad máxima de 600 Mb/s y en velocidad promedio a 300 Mb/s. Se ha implementado desde 2008. Es el primer estándar de la familia 802.11 que trabaja en dos bandas de frecuencias: 2.4 GHz (802.11b y 802.11g) y 5 GHz (802.11 a). Esto lo hace compatible con dispositivos basados en estándares anteriores.

Estos estándares son los más utilizados actualmente para un uso comercial, existen variaciones del estándar donde algunas son utilizadas sólo para investigación. A continuación se muestra una tabla donde se detalla brevemente cada una de ellas.

Las diferentes versiones del estándar 802.11 permiten actualizaciones, garantizan la compatibilidad entre equipos viejos y equipos emergentes.

Tabla 1 Estándares IEEE 802.11 [5] [C] [D] [E]

Estándar	Año	Descripción
802.11	1997	Define la capa física y MAC de las redes LAN inalámbricas (infrarrojo y radio a 2.4 GHz).
802.11a	1999	Especifica redes de alta velocidad (54Mbps) en la banda de 5GHz.
802.11b	1999	Especifica redes LAN inalámbricas de velocidades de 5.5 a 11 Mbps a 2.4 GHz.
802.11c	1998	Define las características necesarias de los puntos de acceso para actuar como puentes (Bridges).
802.11d	2001	Adapta los requerimientos regionales (Modo Mundial).
802.11e	2005	Calidad de servicio para aplicaciones en tiempo real (voz, video, etc.)
802.11f	2000	Proporciona interoperabilidad entre puntos de acceso de distintos fabricantes (Interfaces Point Protocol, IAPP) para permitir la itinerancia (Rooming).
802.11g	2003	Especifica redes de alta velocidad (54Mbps) en la banda de 2.4 GHz.
802.11h	2003	Mejora el uso del espectro mediante una selección dinámica de canal y control de potencia de transmisión.
802.11i	2004	Mejoras para seguridad y autentificación.
802.11j	2004	802.11a con canales adicionales por encima de 4.9 GHz (802.11ª en Japón).
802.11k	2002	Intercambio de información de capacidad entre clientes y puntos de acceso.
802.11m	2003	Propuesto para el mantenimiento de redes inalámbricas.
802.11n	2004	Nueva generación de redes inalámbricas de alta velocidad (hasta 540 Mbps teóricos) a 2.4 y 5 GHz.
802.11p	2008	Acceso inalámbrico para el entorno de vínculos (autos, ambulancias, etc.).
802.11r	2008	Establece protocolos de seguridad para identificar un dispositivo en un nuevo punto de acceso antes de abandonar el actual logrando pasar a él en menos de 50 milisegundos.
802.11v	2011	**Nivel Prueba**. Permitirá la configuración remota se los dispositivos cliente, incluirá mecanismos de ahorro de energía, mejor posicionamiento, temporización y coexistencia.
802.11w	2009	**Nivel Prueba.** Permitirá aumentar la seguridad de los protocolos de autentificación y codificación.

2.3.2 Funcionamiento de Wi-Fi

Una red Wi-Fi puede formarse por dos ordenadores o miles de ellos. Para que exista una comunicación inalámbrica entre ellos es necesario instalar un adaptador de red.

Un adaptador de red es un equipo de radio (Transmisor/Receptor) que está conectado con un equipo susceptible a formar parte de la red. A todos los equipos dentro de la red se les denomina "Terminales".

Las redes Wi-Fi necesitan de otros equipos llamados Access Points AP, los cuales son utilizados como estación base para gestionar las comunicaciones entre las distintas terminales. Los Access Points funcionan de forma autónoma sin necesidad de conectarse a un ordenador.

Las terminales y los Access Points se les conocen como estación. Las estaciones logran establecer una comunicación gracias a que utilizan la misma banda de frecuencias y que se rigen bajo los mismos protocolos. El estándar 802.11 sólo define las dos primeras capas (física y enlace).

2.3.3 Las capas del Estándar IEEE 802.11 (Wi-Fi)

La capa física del estándar IEEE 802.11 se divide en dos subcapas; una se denomina PLC[18] y PMD[19].

- **PLC:** Está encargado de convertir los datos a un formato compatible con el medio físico, esto se aplica si se trata de un medio físico infrarrojo o de radio.
- **PMD:** Es el encargado de la difusión de la señal.

A pesar que las especificaciones originales de IEEE 802.11 contemplaban utilizar infrarrojos como medio de transmisión, nunca llegó a desarrollarse un sistema con esas características debido al corto alcance que ofrece y no es posible utilizarse en el exterior por las interferencias producidas por la lluvia o niebla.

2.3.4 Spread Spectrum DSSS y FHSS

Para el uso del espectro radioeléctrico, la tecnología en la cual se basa el funcionamiento de los sistemas inalámbricos es conocida como Spread Spectrum (Espectro expandido). Este sistema consiste en que el ancho de banda real utilizado en la transmisión es superior al necesario para la transmisión de la información. Con esto se logra un sistema resistente a las interferencias de otras fuentes de radio, a los efectos de eco (multipath) y pueden coexistir otros sistemas de radiofrecuencia sin que se afecten o influyan en su actividad. Estas ventajas hacen que el Spread Spectrum sea la más adecuada en las bandas de frecuencia para las que no se necesita licencia.

IEEE 802.11 contemplaba sólo dos técnicas distintas de Spread Spectrum para la capa física pero con la versión IEEE 802.11a se desarrolló una nueva técnica conocida como OFDM la cual no es, por definición, una técnica de espectro extendido.

[18] PLC: *Physical Medium Layer Convergence Procedure,* "Procedimiento de Convergencia de la Capa Física".
[19] PMD: Physical Medium Dependent, "Dependencia del Medio Físico".

2.3.4.1 FHSS

El FHSS[20] divide la banda de frecuencias en una serie de canales y transmite la información saltando de un canal a otro de acuerdo con un patrón de saltos (*spread code* o *hopping code*) conocido por el emisor y el receptor. El tiempo máximo para permanecer en cada frecuencia está establecido para 400 ms.

La desventaja de FHSS es que necesita sincronizar el emisor y el receptor en la frecuencia a utilizar en cada momento.

En 1997 el estándar IEEE 802.11 definió que cada canal de FHSS tuviera un ancho de banda total disponible y el número total de canales sería variable de acuerdo con el marco regulatorio de cada país o área geográfica.

FHSS reduce las interferencias debido a que en el peor de los casos afectará exclusivamente a uno de los saltos de frecuencia. Liberándose a continuación de la interferencia al saltar a otra frecuencia distinta. El resultado de bits erróneos será extremadamente bajo.

Otra ventaja de FHHSS es que permite que varias comunicaciones coexistan en la misma banda de frecuencias. Para esto cada canal debe tener un patrón de saltos con distinta secuencia.

A pesar que el estándar original IEEE 802.11 incluye el sistema FHSS, no existe ninguna implementación real. Esto se debe a que la velocidad máxima conseguida con la técnica FHSS es de 3 Mbps, pero es posible que en un futuro se puedan conseguir velocidades superiores de hasta 15 Mbps.

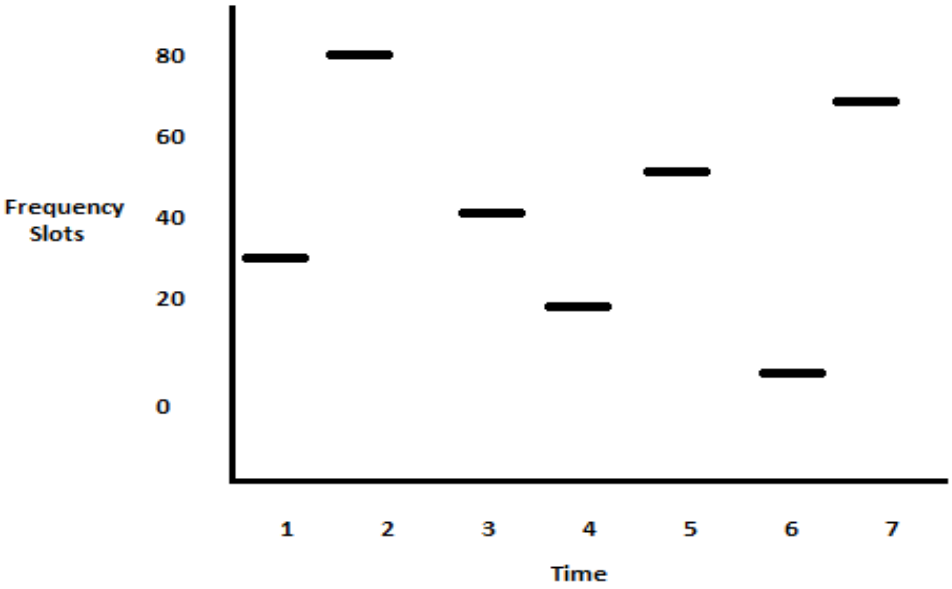


Fig. 2.4 Sistema FHSS [5]

[20] FHSS: *Frequency Hopping Spread Spectrum,* "Espectro Extendido por Salto de Frecuencia".

2.3.4.2 DSSS

El DSSS[21] sustituye cada bit de información por una secuencia de bits conocida como *chipping code* "código de chips". Estos códigos permiten a los receptores eliminar por filtrado las señales que no utilizan la misma secuencia de bits. Entre las señales que son eliminadas se encuentra el ruido y las interferencias.

El *chipping code* permite al receptor identificar los datos como pertenecientes a un emisor determinado. El emisor genera el *chipping code* y sólo los receptores que conocen del código pueden descifrar los datos. Por lo tanto, teóricamente DSSS permite que varios sistemas puedan funcionar en paralelo; cada receptor filtrará exclusivamente los datos que se corresponden con su código. Por otro lado cuanto más largo sea el *chipping code* más resistente será el sistema a las interferencias y más sistemas podrán coexistir simultáneamente. La norma IEEE 802.11 determina que la longitud mínima del *chipping code* debe ser de 11.

En la práctica, la coexistencia de sistemas no se consigue por el uso de distintos *chipping codes,* sino por el uso de distintas bandas de frecuencia. Un sistema DSSS de 11 Mbps (IEEE 802.11 b) necesita un ancho de banda de 22 MHz., esta es la distancia mínima entre portadoras de 30 MHz. Como el ancho de banda disponible en la banda de 2.4 GHz es de 83.5 MHz, solo pueden coexistir tres sistemas DSSS en un mismo lugar.

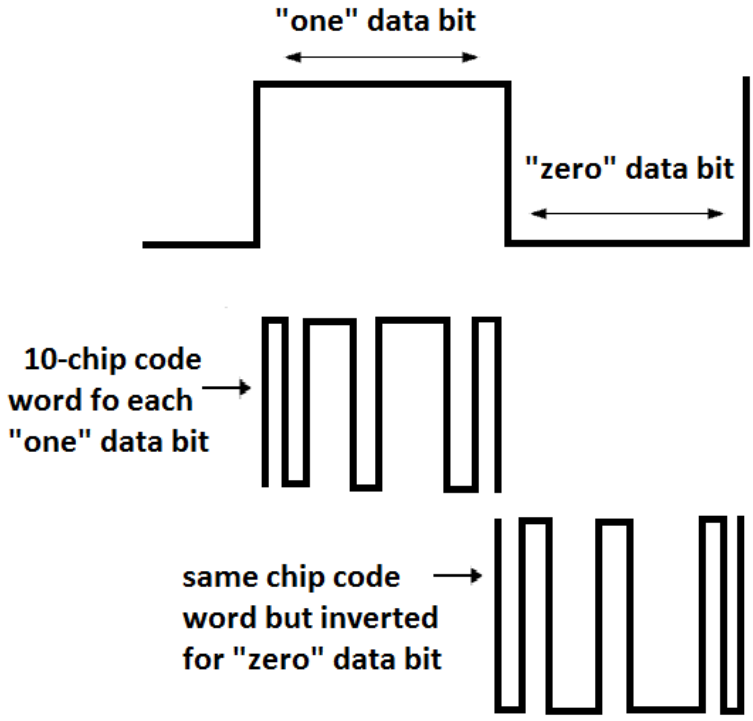


Fig. 2.5 Funcionamiento de DSSS [5]

[21] DSSS: *Direct Sequence Spread Spectrum,* "Espectro Extendido por Secuencia Directa".

2.3.5 OFDM

OFDM[22] es una técnica de gestión de frecuencias utilizada por IEEE 802.11a y 802.11g. Esta técnica divide el ancho de banda en subcanales más pequeños que operan en paralelo. De esta forma consiguen legar a velocidades de transmisión de hasta 54 Mbps.

La técnica OFDM fue patentada por Bell Labs en 1970, está basada en un proceso matemático llamado FFT[23]. OFDM divide la frecuencia portadora en 52 subportadoras solapadas, 48 de estas subportadoras son utilizadas para transmitir datos y las otras cuatro para poder alinear las frecuencias en el receptor. Este sistema consigue un uso muy eficiente del espectro radioeléctrico.

OFDM puede transmitir datos a distintas velocidades utilizando diversas técnicas modulación. Las velocidades normalizadas que admite OFDM son 6, 9, 12, 18, 24, 36, 48, y 54 Mbps.

Una ventaja de OFDM es la alta resistencia a las interferencias producidas por las ondas reflejadas en los objetos del entorno (eco o *multipath*). Estas ondas llegan al receptor con distintas amplitudes y a distinto tiempo que la señal principal produciendo interferencias. Estas interferencias son un problema a velocidades mayores a 4 Mbps; para evitar esto se utilizan técnicas como esta para mitigar esos efectos.

Tabla 2 Técnicas de modulación utilizadas por IEEE 802.11ª [5]

Velocidad	Técnica de Modulación	Bits por Señal
6 Mbps	BPSK	1
9 Mbps	BPSK	1
12 Mbps	QPSK	2
18 Mbps	QPSK	2
24 Mbps	QAM-16 (BPSK)	4
36 Mbps	QAM-16 (BPSK)	4
48 Mbps	QAM-64 QPSK	6
54 Mbps	QAM-64 QPSK	6

[22] OFDM: *Orthogonal Frequency Division Multiplexing,* "Multiplicación Ortogonal por División de Frecuencias".
[23] FFT: *Fast Fourier Transform,* "Transformada Rápida de Fourier".

2.3.6 Modulación de la señal

Para transmitir una señal vía radio, es necesario definir un método de difusión de la señal y un método de modulación. La modulación consiste en modificar una señal pura para incorporarle la información que se quiere transmitir. La señal base que se va a modular se le conoce como *carrier* "portadora". Lo que cambia en la portadora para poder modular puede ser su amplitud, frecuencia, fase o una combinación de estas. Mientras más velocidad de transmisión se necesite, más complejo será el sistema de modulación.

Las técnicas de modulación utilizadas en IEEE 802.11 son:

- **BPSK:** *Binary Phase-Shift Keying*, "Modulación Binaria por salto de fase"
- **QPSK:** *Quadrature Phase-Shift Keying*, "Modulación por salto de fase en cuadratura"
- **GFSP:** *Gaussian Frecuency-Shift Keying*, "Modulación Gaussiana por Salto de Frecuencia"
- **CCK:** *Complementary Code Keying* "Modulación de Código Complementario"

Al emitir la señal modulada, el receptor tiene que recibir la señal, sincronizar el código de difusión y demodular la información. Los sistemas FHSS son más difíciles de sincronizar que los sistemas DSSS ya que en los primeros se tiene que sincronizar en tiempo y frecuencia, mientras que en los segundos sólo en el tiempo.

2.3.7 Capa MAC (Medium Access Control)

La capa MAC define los procedimientos necesarios para que distintos dispositivos compartan el uso del espectro radioeléctrico. En distintas versiones del estándar IEEE 802.11 se utilizan distintos sistemas para difundir la señal (distinta capa física), la capa MAC es la misma para todo el estándar.

La capa MAC utilizada por estas redes es muy similar a la utilizada por la red Ethernet, ambas utilizan la técnica CSMA. Sin embargo Ethernet (alambrico) utiliza la tecnología CD [24] y Wi-Fi (inalámbrico) utiliza CA[25]. La tecnología CD detecta cuando ocurrió una colisión y retransmite los datos, CA cuenta con procedimientos para evitar que se produzcan colisiones.

Se necesitan dos sistemas diferentes debido a que al utilizar un cable como medio físico de transmisión, una terminal puede y recibir al mismo tiempo, esto genera colisiones. Por el medio radioeléctrico un terminal no puede transmitir y recibir al mismo tiempo por el mismo canal debido a que la transmisión opacaría a la recepción, al no poder detectar las colisiones lo único que se puede hacer es disponer de una técnica que las evite.

[24] CD: *Collision Detection,* "Detección de Colisión".
[25] CA: *Collision Avoidance,* "Evitación de Colisiones".

2.3.8 Trama de IEEE 802.11

Las tramas MAC contienen los siguientes componentes.

- Cabecera MAC: Comprende campos de control, duración, direccionamiento y control de secuencia.
- Cuerpo de trama de longitud variable: Contiene información específica del tipo de trama.
- Secuencia Checksum (FCS): Contiene un código de redundancia CRC de 32 bits.

Las tramas MAC se clasifican en tres tipos:

1. Tramas de datos.
2. Tramas de control.
3. Tramas de gestión.

La trama es muy parecida a la de la familia IEEE 802.3, siendo de 48 bits de longitud y con campos comunes a la trama Ethernet.

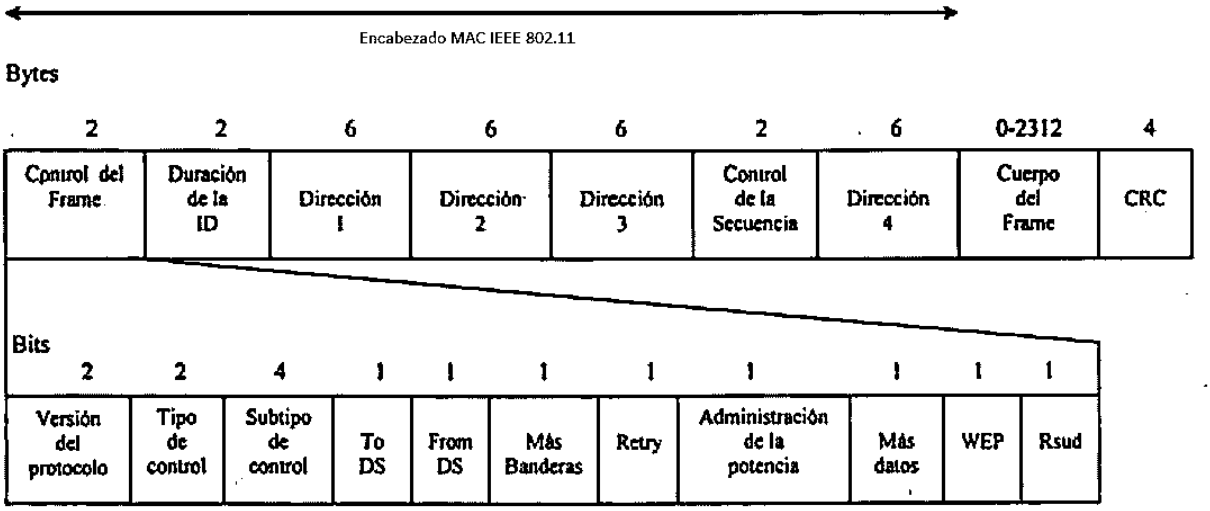


Fig. 2.6 Trama MAC IEEE 802.11 [5]

2.3.9 Seguridad

Actualmente la tecnología Wi-Fi enfrenta una saturación progresiva del espectro radioeléctrico, el porcentaje de redes instaladas es muy elevado y no siempre se tiene consideración de la seguridad dejando estas redes abiertas y vulnerables para acceder a ellas, dejando desprotegida la información que circula por ellas.

Existen protocolos de cifrado de datos para los estándares Wi-Fi como el WEP, el WPA y el WPA2 que se encargan de que la información transmitida se codifique.

WEP: Realiza un cifrado en los datos de la red a 64 y 128 bits para que sólo el destinatario deseado pueda tener acceso a ellos. La codificación se realiza por medio de una "clave" de cifrado antes de ser enviado. Este cifrado no es muy recomendado debido a las grandes vulnerabilidades que presenta ante cualquier *craker* que pueda conseguir la clave.

WPA: Presenta mejoras ante el WPE tales como una generación dinámica de la clave de acceso la cual se inserta como dígitos alfanuméricos.

WPA2: Es una mejora relativa a WPA. Se considera el protocolo de seguridad más efectivo para i-Fi actualmente, sin embargo es necesario hardware y software compatibles ya que dispositivos antiguos no lo son.

La seguridad dentro de una red Wi-Fi puede ponerse a prueba por medio de una auditora de Wi-Fi. Sin embargo, no existe ninguna alternativa completamente confiable ya que todas son susceptibles de ser vulneradas.

2.3.10 Ventajas y desventajas de Wi-Fi

Las redes Wi-Fi poseen una serie de ventajas, entre las cuales podemos destacar:

- La Wi-Fi Alliance asegura que exista una compatibilidad entre todos los dispositivos Wi-Fi.
- Las redes inalámbricas ofrecen la comodidad de conectarse desde distintos puntos dentro del rango de la red, proporcionando movilidad al usuario.
- Al configurar una red Wi-Fi se permite el acceso de múltiples ordenadores sin problemas ni gasto en infraestructura evitando el cableado estructural en un área donde se concentran muchos usuarios.

Al ser una tecnología para redes inalámbricas, Wi-Fi presenta una serie de problemas intrínsecos de cualquier tecnología de este tipo.

- El sistema Wi-Fi tiene una velocidad menor en comparación con una conexión cableada, eso se debe a las interferencias y pérdidas de señal que el ambiente proporciona.
- Una desventaja fundamental de estas redes es la seguridad ya que existen programas capaces de recapturar paquetes de forma que pueden calcular y decodificar la contraseña de red y acceder a ellas
- No se puede controlar el área de cobertura de una conexión, de manera que un receptor se puede conectar desde fuera de la zona de recepción prevista.
- Esta tecnología no es compatible con otros tipos de conexiones sin cables como Bluetooth, GPRS, UMTS, etc.
- La cobertura de un sistema Wi-Fi es afectada por los agentes físicos que se encuentran a nuestro alrededor, tales como: árboles, paredes, arroyos, una montaña, etc.

Capítulo 3

RFID

3.1 Sistemas RFID

RFID es un acrónimo de *Radio Frequency Identification*, "Identificación por Radio Frecuencia", es una tecnología de comunicación inalámbrica utilizada pata identificar objetos etiquetados o personas. Tiene muchas aplicaciones que actualmente están en uso:

- Sistemas de control de acceso por medios sin llaves e identificadores de personal.
- Sistemas automáticos de autentificación de entradas a puentes, túneles y torniquetes.
- Dispositivos de rastreo para animales fuera del cautiverio y últimamente se ha comenzado a utilizar en mascotas.
- Localización de vehículos e inmovilizadores.
- Pulseras de muñeca y tobillo para ID y seguridad de niños.
- Una cadena de suministro y seguimiento de objetos o mercancía de los proveedores.

Estas son unas pocas por mencionar las más importantes ya que en los próximos años las nuevas aplicaciones para RFID beneficiarán a una gran cantidad de industrias y agencias gubernamentales en maneras que ninguna otra tecnología podrá hacer. [7] [8]

3.2 Componentes de un sistema RFID

Un sistema RFID utiliza comunicaciones inalámbricas por radio frecuencia para identificar objetos etiquetados o personas. Se necesitan tres componentes básicos para un sistema RFID.

1. Un Tag (en algunas ocasiones llamado también Etiqueta), el cual está compuesto por un chip conductor, una antena y en algunos casos una batería.
2. Un Interrogador (también llamado Lector (así lo llamaremos a partir de aquí) o dispositivo de Lectura/Escritura), el cual está compuesto por una antena, un módulo de RF y un módulo electrónico de control.
3. Un Controlador (Llamado también Host), toma el lugar de una computadora y realiza el control del software.

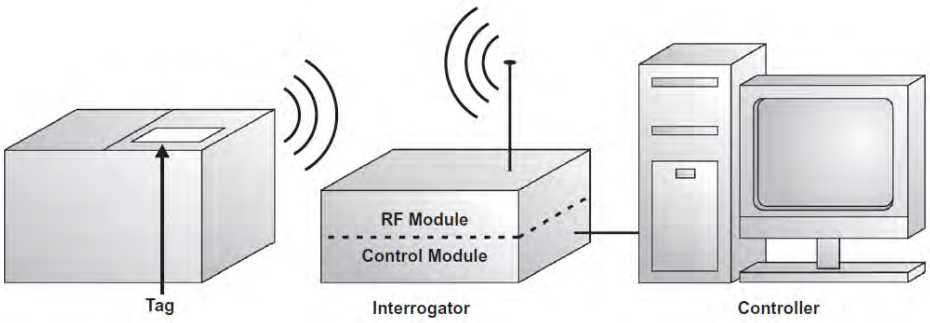


Fig. 3.1 Diagrama a bloques de una estructura básica de un sistema RFID [7]

EL Tag y el Lector transmiten información de uno a otro por medio de ondas de radio. Cuando el Tag entra en la zona de cobertura del Lector, El Lector detecta el tag para transmitir la información que lleva almacenada. Un Tag puede guardar muchos tipos de información como: números de serie, etiquetas de tiempo, instrucciones de configuración, etc. Una vez que el Lector recibió la información del Tag, la información es enviada al controlador por medio de una interface de red ya sea Ethernet o por medio de Internet. El controlador puede utilizar esa información para distintos propósitos. Por ejemplo, el controlador puede utilizar esa información para hacer un inventario y crear una base de datos o re direccionar a un objeto hacia una nueva ruta.

Un sistema RFID puede contar con varios Lectores distribuidos a través de una ruta a seguir o dentro de una serie de puntos estratégicos en una locación. Sin embargo todos los Interrogadores deben estar conectados en red a un solo controlador. Un Lector puede comunicare con varios Tags simultáneamente, actualmente se puede lograr la comunicación de 1,000 Tags por segundo, con un promedio de éxito de lectura de un 98%. Finalmente un Tag, puede ser instalado en cualquier objeto, una paleta, una computadora, un bebé, una caja, etc.

3.2.1 TAGS RFID

LA función básica de un Tag RFID es almacenar información y transmitirla al Lector. Un Tag básicamente está formado por un chip electrónico y una antena encapsulada en paquete dependiendo la forma y la aplicación para el Tag. Generalmente, el chip contiene una memoria donde la información será almacenada, leída o incluso se podrá sobre escribir. Un elemento adicional es una batería la cual diferenciará entre un Tag activo y un Tag pasivo. Otro factor que diferencia a los Tags radica en el tipo de memoria, hay dos tipos RO [26] y RW[27].

3.2.1.1 TAG Activo

Se denominan activos cuando contienen una fuente de poder integrada, como una batería. Cando el Tag necesita transmitir información al Lector, este utiliza esta fuente para generar la potencia necesaria para la transmisión, muy similar a lo que hace un celular con su batería. Debido a esto, los Tags Activos pueden comunicarse con Lectores de menos potencia y pueden transmitir en a distancias más largas. También este tipo de Tags tienen una memoria mayor (hasta 128Kb). Sin embargo son más grandes y más complejas de fabricar que las pasivas, fabricarlas cuesta mucho y esto las hace costosas. Las baterías de los Tags Activos pueden durar de dos a siete años.

3.2.1.2 TAG Pasivo

Los Tags pasivos no tienen una fuente de poder integrada. La energía necesaria para transmitir la obtienen del Lector. Como resultado de esto, los Tags pasivos son más pequeños y más económicos que los Tags activos. Sin embargo la distancia de transmisión es mucho más corta que un Tag activo, por lo que se requiere un Lector mucho más potente y por otro lado tienen menos capacidad de memoria.

[26] RO: *Read-Only*, "Solo Lectura".

[27] RW: *Read/Write*, "Lectura/Escritura".

Algunos Tags pasivos contienen baterías pero no las utilizan para asistir la transmisión de las ondas de radio. Este tipo de Tags pasivos utilizan la batería solamente para activar elementos electrónicos dentro del mismo. Estos elementos pueden ser luces, indicadores o sensores que se activan ante algún cambio (Temperatura, presión, movimiento, tiempo, etc.).

3.2.1.3 RO Tags

Los RO Tags contienen una memoria que solo puede leerse, algo similar a los códigos de barras, los cuales pueden programarse una sola vez y por lo tanto no pueden ser alterados. Estos Tags usualmente son programados con una información muy limitada la cual será estática, pueden ser un número de serie o un número de parte.

3.2.1.4 RW Tags "Smart Tags"

Generalmente llamados "Smart Tags", presentan más flexibilidad que los RO Tags. Pueden almacenar mayores cantidades de información y tienen una memoria que puede cambiarse fácilmente. En los Smart Tags puede borrarse o sobre-escribir la información muchas veces. Debido a esto los Tags pueden actuar como una base de datos móvil. Esta cualidad ha impulsado esta tecnología debido al gran número de aplicaciones que pueden realizarse.

Existen algunas variaciones de estos dos tipos de memorias. Existe una memoria denominada WORM[28]. Es similar a RO al ser programada con información estática. Este tipo de memoria se utiliza en una línea de ensamblado donde en un Tag se señala la fecha o la localización después de que el proceso de producción se completó.

Adicionalmente, algunos Tags pueden contener ambas memorias RO y RW al mismo tiempo. Por ejemplo un Tag RFID puede estar marcado con un número de serie, este estaría en la memoria RO. En la parte de la memoria RW puede indicar el contenido y en cualquier momento al ser leído puede sobre-escribirse en la información para reflejar un cambio.

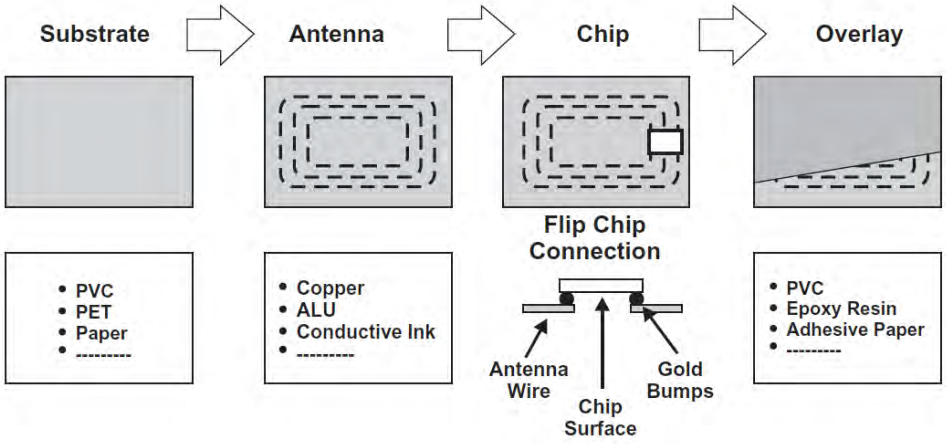


Fig. 3.2 Componentes de un Tag RFID [7]

[28] WORM: *Write-Once-Read-Many, "*Escrita una vez/Leida muchas veces".

3.2.2 Diseños y formatos comerciales de Tags

Los diseñadores y fabricantes de sistemas RFID se tienen implementar diferente tipos de formatos dependiendo de la aplicación para la cual está dirigido el sistema, estos diseños generalmente están montados sobre artículos de uso común para que pasen desapercibidos por las personas, esto genera un grado de seguridad para los usuarios y empresas.

3.2.2.1 Coin/Disk

Uno de los formatos más comunes son llamados *Coin* o *Disk* (moneda o disco), tienen diámetros no mayores al rango de los 10cm Usualmente tienen un hoyo en el centro para un tornillo que la fije**,** adicionalmente tiene un recubrimiento de polystryol o una resina epóxica para garantizar e rango de temperatura de operación.

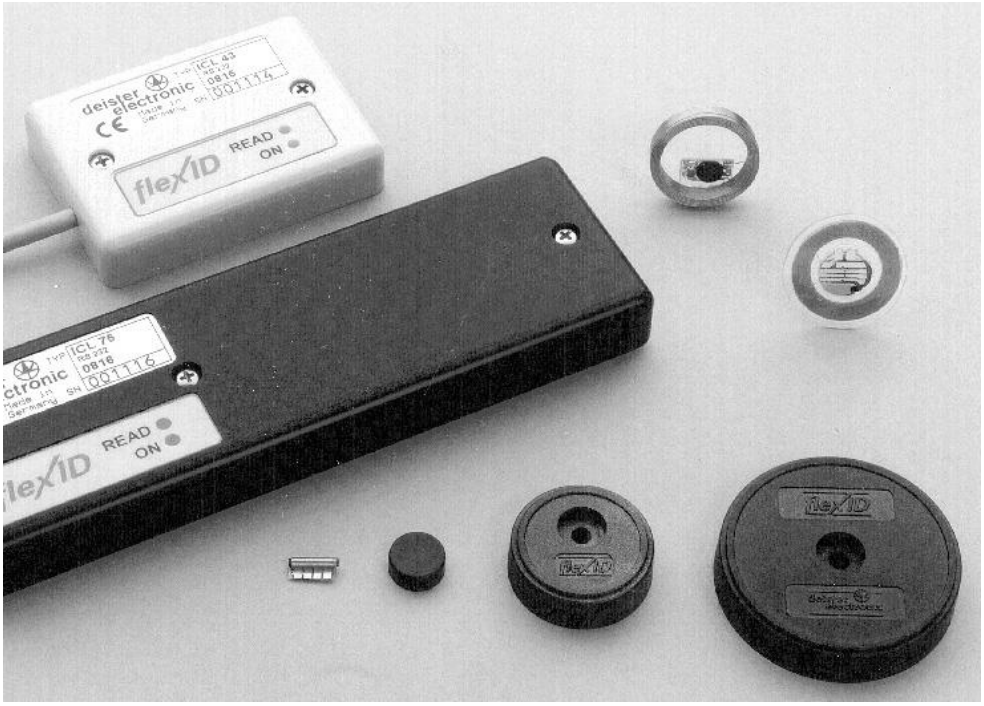


Fig. 3.3 Diferentes tipos de Tags *Coin/Disk* (moneda o disco). [Propiedad de Deister Electronic, Barsinghausen] [8]

3.2.2.2 Cápsulas de vidrio

Los Tags de vidrio fueron diseñados para ser insertados bajo la piel de animales para propósitos de identificación. Los tubos de vidrio de 12–32mm Contienen un microchip montado en una placa junto a un capacitor de chip para proporcionarle la corriente necesaria. Contiene una bobina de 0.03mm de espesor sobre un núcleo de ferrita. Los componentes internos están montados por un adhesivo suave que garantiza la estabilidad.

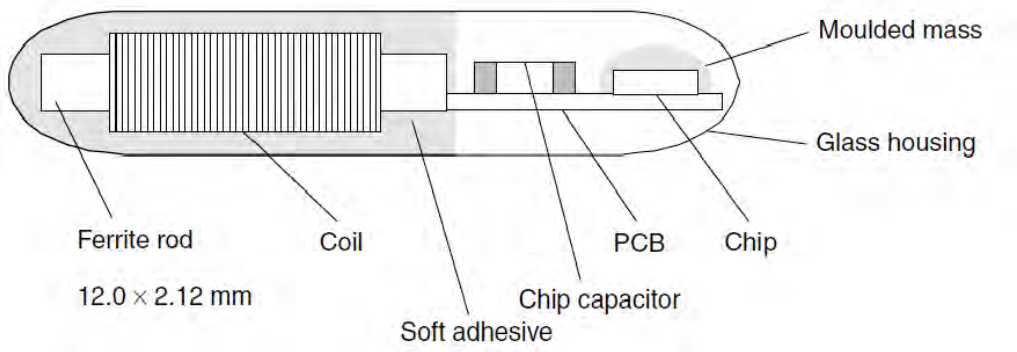


Fig. 3.4 Diseño mecánico de un Tag de vidrio [8]

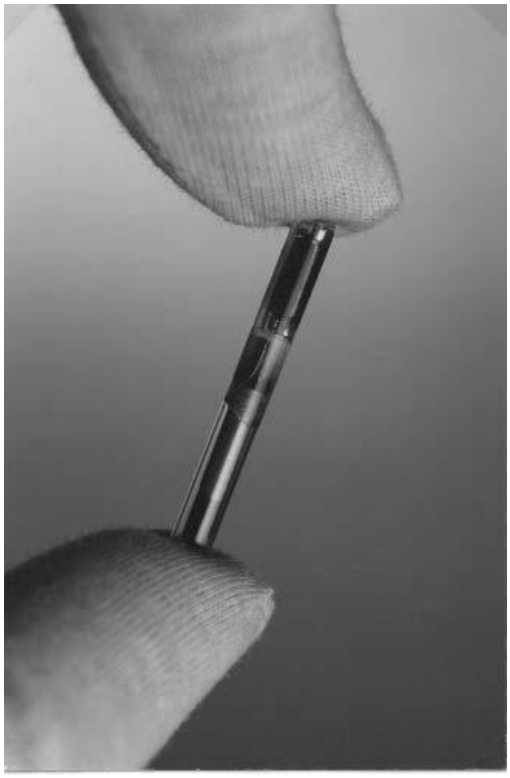


Fig. 3.5 Vista de un Tag de vidrio encapsulado de 32mm utilizado para la identificación de animales. [Propiedad de Texas Instruments] [8]

3.2.2.3 Cápsulas de plástico

Los Tags de plástico fueron desarrollados para aplicaciones que involucraban una alta demanda mecánica. Este diseño de cápsula puede ser integrado fácilmente a otros productos. Contiene una bobina más larga que le da una mayor funcionalidad, contiene casi los mismos componentes que el de vidrio. Otra ventaja es su capacidad de aceptar chips más grandes y tener una tolerancia mayor a las vibraciones mecánicas.

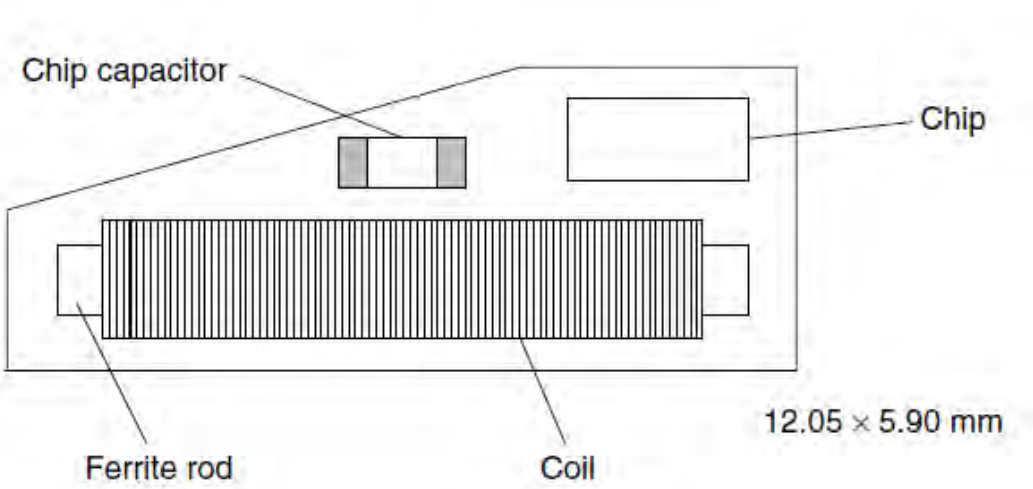


Fig. 3.6 Diseño mecánico de una cápsula de plástico. [8]

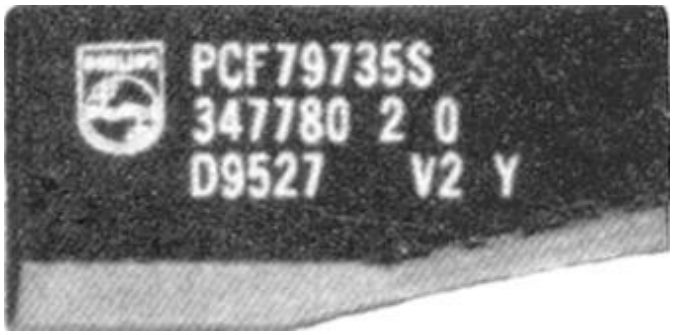


Fig. 3.7 Ejemplo de un Tag encapsulado en plástico. [Propiedad de Philips Electronics B.V.] [8]

3.2.2.4 Llaves y llaveros

Algunos Tags RFID están integrados en llaves para aplicaciones de alta seguridad de inmovilizadores o para bloqueos de puertas. Generalmente están basados en las capsulas de plástico en forma de llavero. El diseño de llavero ha sido muy popular para los sistemas de acceso a oficinas.

Fig. 3.8 Tag de Llavero para Sistemas de Acceso. [Propiedad de Intermarketing] [8]

3.2.2.5 Relojes

Estos diseños comenzaron a principios de la década de los 90´s por la compañía Austriaca *Ski-Data* y fue utilizada por primera vez en pases para los esquís. Estos relojes están ganando terreno dentro de los sistemas de acceso. El reloj contiene una antena impresa con devanados cobre un circuito delgado, utilizando la carcasa del reloj para maximizar el rango.

Fig. 3.9 Reloj con un Tag integrado para los Sistemas de Autorización de Acceso Contactless. [Propiedad de Junghans Uhren GmbH, Schramberg] [8]

3.2.2.6 Formato ID-1, Smart Cards

El formato ID-1 es muy familiar a las tarjetas de crédito y las tarjetas telefónicas (85.72mm x 54.03mm x 0.76mm +/- tolerancias) se han convertido cada vez más importantes en los sistemas RFID. Una ventaja de estos sistemas de RFID es la gran área de la bobina lo cual aumenta el rango de las Smart Cards.

Las *Smart Cards* son producto de la laminación de un transpondedor entre cuatro láminas de PVC. Las láminas individuales se cuecen en hornos a alta presión y a temperaturas mayores a 100°C para producir una unión permanente.

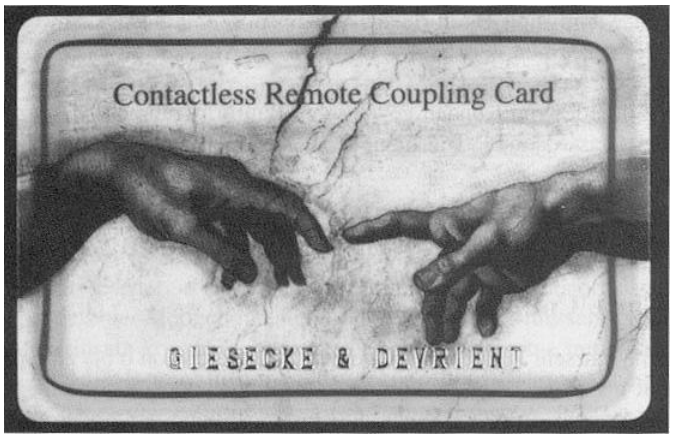


Fig. 3.10 *Smart Card* semitransparente. La antena puede verse claramente alrededor del borde de la tarjeta. [Propiedad de Giesecke & Devrient, Munich] [8]

3.2.2.7 Smart Label

El término de *Smart Label* (Etiqueta Inteligente) es un formato de Tag tan delgado como el papel. Este formato contiene una bobina integrada sobre una lámina de plástico de apenas 0.1mm de espesor por serigrafía o grabado. Tiene un recubrimiento de papel aluminio y una capa de papel adhesivo en la parte trasera. Son flexibles y pueden adherirse a muchas superficies.

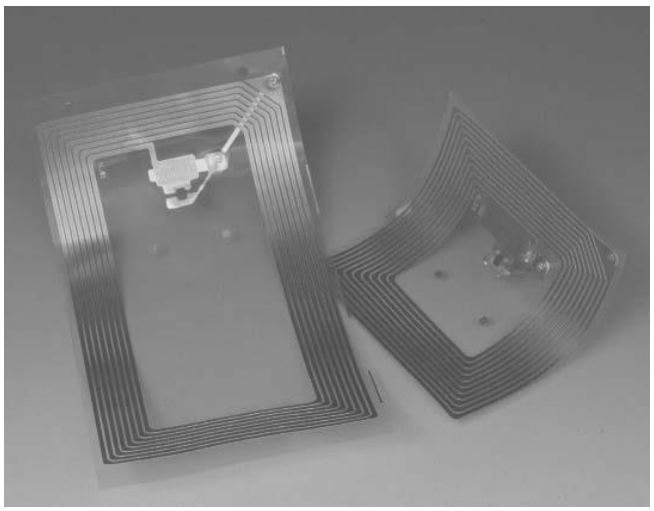


Fig. 3.11 Tag *Smart Label* (Etiqueta Inteligente). [Propiedad de Texas Instruments, Friesing] [8]

3.2.2.8 Coil-on-chip (Bobina en chip)

En la construcción de los formatos mencionados previamente el Tag consistía por separado en una bobina que funcionaba como antena y un chip. La bobina del transponedor en este formato está unido al chip, para lograr esto la bobina se coloca directamente sobre el aislador del chip de silicio como una capa extra. La pista del conductor está en el rango de los 5–10µm con un grosor de 15–30µm.

El tamaño del chip de silicio, junto con el transpondedor es de tan solo 3mm x 3mm. El transpondedor se encuentra embebido en una cubierta plástica de 0.6mm x 1.5mm siendo estos los Tags más pequeños disponibles en el mercado.

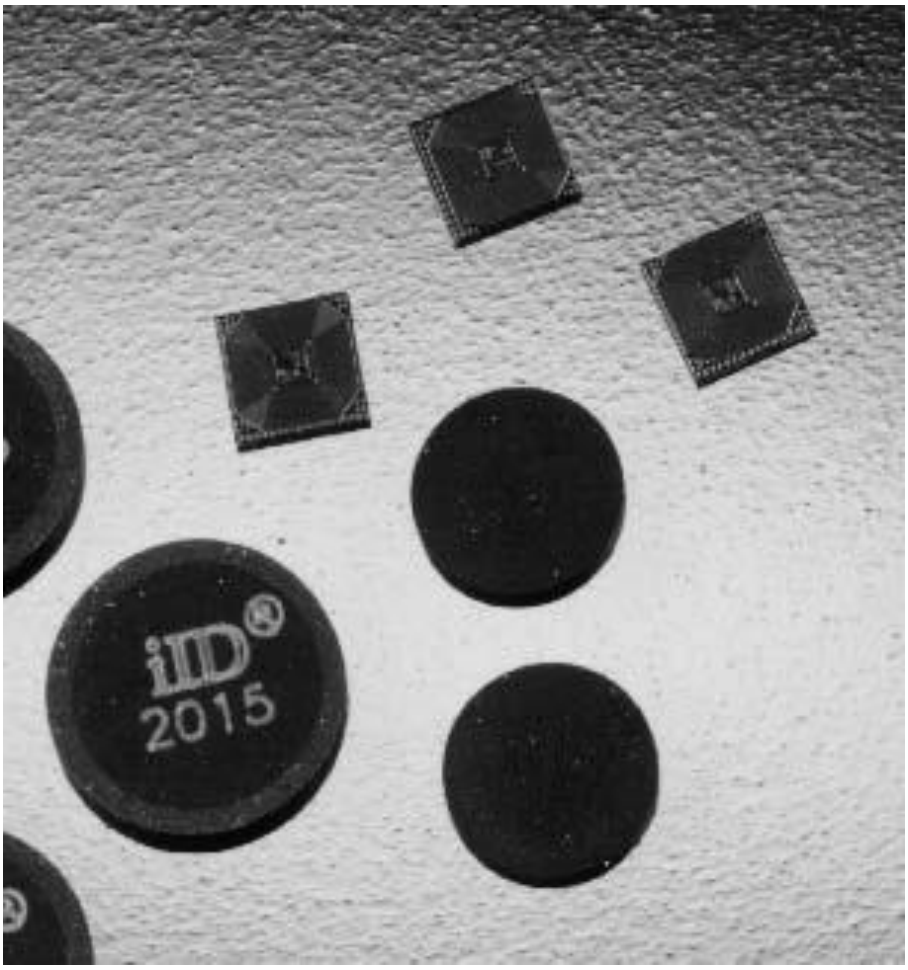


Fig. 3.12 Tag miniatura bobina en chip (*coil-on-chip*). [Propiedad de Micro Sensys, Erfurt] [8]

3.2.3 Lectores RFID

Un lector RFID actúa como un puente entre el Tag y el Controlador, el Lector tiene algunas funciones básicas.

- Leer la información contenida en el Tag RFID.
- Escribir información en el Tag (en el caso de los Smart Tags).
- Direccionar la información hacia el Controlador.
- Energizar al Tag (en el caso de Tags pasivos.

Los Lectores RFID básicamente son pequeñas computadoras. Principalmente están compuestas por tres partes: Una antena, un módulo de RF responsable de comunicarse con el Tag y un módulo electrónico de control.

Los Lectores RFID pueden realizar tareas más complejas que las cuatro básicas mencionadas previamente:

- Implementar medidas anti-colisiones para asegurar la comunicación simultánea con muchos Tags.
- Autentificar Tags para prevenir fraudes o acceso no autorizados al sistema.
- Encriptar y proteger la integridad de la información.

Fig. 3.13 Lector RFID comercial [8]

3.2.3.1 Medidas Anticolisiones.

Se han implementado algoritmos anticolisiones para habilitar al Lector a comunicarse con muchos Tags al mismo tiempo. Si el Lector no conociera el número de Tags que se encuentran en su zona de cobertura, no se sabría cómo enviar la información ya que todos pueden ser detectados y transmitir al mismo tiempo. Para esto se desarrolló un plan de contingencia denominado para RFID "*Anticollision*".

Existen tres tipos de técnicas para anticolisiones: espacial, frecuencia y por dominio de tiempo. Las tres se utilizan para establecer una prioridad o una serie de medidas aleatorias en el sistema para prevenir problemas de ocurrencia o por lo menos hacer que la ocurrencia sea estáticamente diferente**.**

3.2.3.2 Autentificación.

Se necesita un sistema de muy alta seguridad en el Lector para autentificar a los usuarios. Los sistemas de "Puntos de Venta", donde se maneja dinero a cambio de productos pueden ser propenso a fraudes si no se tuvieran estas medidas. En este ejemplo, el proceso de la autentificación se produce en dos partes, una ocurre en el Controlador y el otro proceso en el Lector.

Existen dos tipos de autentificación. Una se denomina *Mutual Symmetrical* ("Por Simetría Mutua") y *Derived Keys* ("Llaves derivadas"). En ambos sistemas, un Tag RFID provee un código de acceso al Lector, el cual contiene un algoritmo o un "candado" para determinar si la llave tiene acceso al sistema o no.

3.2.3.3 Encriptación y protección de la información

La encriptación de la información es otra medida de seguridad que se debe tomar en cuenta para prevenir ataques externos al sistema. En el ejemplo de los "Puntos de Venta" imaginemos a una tercera parte la cual intercepta la llave del usuario. Esta información puede ser utilizada para realizar algún fraude en las compras o en las tarjetas de crédito. Para poder proteger la integridad de la información transmitida inalámbricamente y para prevenir una intercepción de algún tercero se utiliza la encriptación. El Lector implementa la encriptación y desencriptación para garantizar la protección. La encriptación es utilizada para evitar fraudes, espionaje, sabotaje industrial y falsificación**.**

3.2.3.4 Ubicación de los Lectores y Factores de Forma

Los sistemas RFID no requieren línea de vista directa entre el Ta y el Lector como lo hacen los sistemas de códigos de barra. Por lo tanto el diseñador del sistema tiene mucha libertad para decidir dónde ubicar los Lectores. Una ubicación fija de los lectores puede ser en la cerradura de una puerta, en la hebilla de un cinturón y en las puertas de acceso para registrar el movimiento de objetos que entran y salen de un complejo. Algunas aplicaciones para almacenaje en bodegas cuelgan Lectores en el techo, los colocan en repisas o estantes para rastrear e movimiento del inventario.

Se pueden montar lectores portátiles en elevadores de carga**,** camiones y otros equipos de manejo de materiales para rastrear paletas y otros artículos en movimiento. Existen Lectores tan pequeños que caben en la mano, esto habilita a los usuarios para ir a locaciones remotas donde es complicado instalar Lectores fijos. También existen Lectores portátiles que se conectan a una PC o Laptop de forma alámbrica o inalámbrica para el envío de datos.

3.2.4 Controladores RFID

Un Controlador RFID es el "cerebro" de cualquier sistema RFID. Son utilizados para interconectar múltiples Lectores y así poder procesar la información. Un controlador en cualquier red es muy similar a una Pc, una base de datos inteligente, una aplicación de software para una red. Los Controladores pueden utilizar la información de los Lectores para:

- Mantener un inventario y alertar algún cambio o necesidad del sistema.
- Rastreas el movimiento de objetos a través de sistema y posiblemente re direccionarlos.
- Verificar la identidad y autorizar sistemas.
- Realizar una cuenta o total en algunos "Puntos de Venta".

Fig. 3.14 Sistema completo RFID [Propiedad de Datalogic] [8]

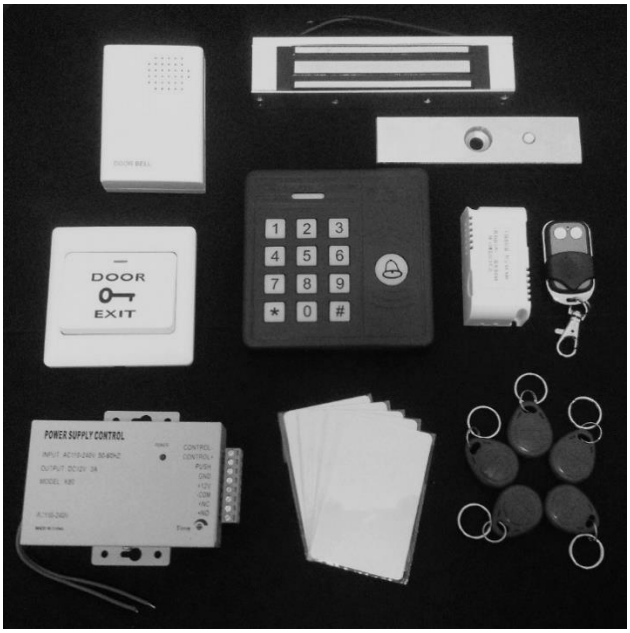


Fig. 3.15 Sistema de Acceso RFID [Propiedad de OEM] [8]

3.3 Historia y Evolución de la tecnología RFID

3.3.1 Sistemas AIDC

Los orígenes de los sistemas RFID provienen de los tradicionales *Paper Taggin* "Etiquetas de papel". Los sistemas *Paper Taggin* fueron desplazados en la década de 1970 por nuevas tecnologías denominadas AIDC [29]. RFID es parte de esta familia, otros partícipes de estas tecnologías son los Códigos de Barras bajo la denominación de OCR [30], Tecnologías Infrarrojas de Identificación y los Códigos QR. La tecnología RFID es la más prometedora de esta familia ya que ofrece muchos beneficios a comparación de las otras tecnologías.

Es difícil identificar un punto de partida de la tecnología RFID, sin embargo la historia de esta tecnología se cruza junto a otras tecnologías que se fueron desarrollando a lo largo del siglo XX. Estas tecnologías incluyen computadoras, tecnologías de la información, teléfonos celulares, redes LAN inalámbricas, comunicaciones por satélites, GPS, etc. RFID emerge como una tecnología por separado, en retrospectiva se sabe que muchos de los desarrolladores han realizado avances en otras tecnologías que han ayudado a la tecnología RFID en su investigación, desarrollo y despliegue.

3.3.2 Convergencia de tres tecnologías

La investigación y los avances en las siguientes áreas ayudaron a ser viable comercialmente esta tecnología.

- Electrónica de Radio Frecuencia: Investigaciones en este campo aplicadas a RFID comenzaron durante la Segunda Guerra Mundial y continuaron hasta la década de 1970. Los sistemas de antenas y de RF empleados en los Lectores RFID y Tags son posibles gracias a estas investigaciones.
- Tecnologías de la información: Las investigaciones en este campo comenzaron en la década de 1970. El Lector y el controlador utilizan esta tecnología. La red de Lectores RFID de un sistema es posible gracias a estas investigaciones.
- Ciencia de materiales: Esta área tuvo su auge en la década de 1990, con esto se lograron Tags RFID con costos de fabricación muy bajos, los que ha logrado que esta tecnología sea comercialmente viable.

[29] AIDC: *Automatic Identification and Data Capture*, "Identificación Automática y Captura de Datos".
[30] OCR: *Optical Character Recognition*, "Reconocimiento Óptico de Caracteres".

3.3.3 Hitos en RFID y el tiempo de adopción

Para poder definir mejor el desarrollo de la tecnología RFID se hará un recorrido a través de una línea del tiempo.

Antes de 1940

La última mitad del siglo XIX tuvo muchos avances en el área del electromagnetismo. Los trabajos de Faraday, Maxwell. Hertz, entre otros, ayudaron a completar leyes que describen su naturaleza. Comenzando en 1896, Marconi, Alexanderson, Baird y Watson apicararon estas leyes en las comunicaciones por radio y al radar, Los trabajos realizados en esta área fueron la base para el desarrollo de muchas tecnologías incluyendo el RFID.

1940 – La Segunda Guerra Mundial

La Segunda Guerra Mundial trajo muchos avances en las comunicaciones por radio frecuencia y radar. Después de la guerra, muchos científicos e ingenieros continuaron sus investigaciones en esta área para adaptarla al uso de civiles. En Octubre de 1948 Harry Stockman publicó un artículo en "*Proceedings of the IRE*" titulado "*Comunications by Means of Refleccted Power*" el cual se considera lo más cercano al nacimiento de la tecnología RFID.

1950 – Primeras exploraciones de la tecnología RFID

Durante la década de 1950 muchas tecnologías relacionadas con el RFID fueron exploradas por investigadores. Un par de artículos fueron publicados, uno por F.L. Vernon "*Applications of the Microwave Homodyne*" y otro por D.B. Harris "*Radio Transmissions Systems with Modulatable Passive Responders*". El Ejército de los Estados Unidos comenzó a implementar una temprana forma de tecnología RFID aérea llamada IFF[31].

1960 – Desarrollo de la teoría de RFID y las primeras pruebas de campo.

Esta década fue un preámbulo para la explosión del RFID que vendría en la siguiente década. R.F. Harrington realizo una gran investigación en el campo de la teoría electromagnética aplicada al RFID, la describió es un artículo llamado "*Field Mesurements Using Active Scatterers*" y en otro titulaado "*Theory of Loaded Scatterers*".

Algunos inventos de RFID comenzaron a surgir, al final de la década algunas empresas comenzaron a producir artículos y tecnologías. Sensormatic and Checkpoint fue fundada para desarrollar EAS[32], equipos anti robo y aplicaciones de seguridad. Sus sistemas eran simples basados en 1-bit, lo que significa que solo podían detectar la presencia del Tag RFID. Los EAS se convirtieron más adelante en lo que sería el uso comercial del RFID.

[31] IFF: *Identification Friend or Foe*, "Identificador de Amigo o Amenaza".

[32] EAS: *Electronic Article Surveillance*, ''Artículos Electrónicos de Vigilancia".

1970 – El auge en RFID y las primeras aplicaciones

Esta década atestiguó el crecimiento de esta tecnología. Compañías, Instituciones Académicas y laboratorios del gobierno comenzaron a involucrarse en la tecnología RFID.

El 1975 El Laboratorio Científico de Los Álamos, realizo un logro en su investigación publicando un artículo titulado "*Short-Range Radio-telemetry for Electronic Identification Using Modulated Backscatter*" publicado por Alfred Koelle, Steven Depp y Robert Freyman.

Compañías como Raytheon, RCA y Fairchild comenzaron a desarrollar sistemas electrónicos RFID. Para 1978 se había logrado el desarrollo de un Transponedor pasivo de microondas.

Muchas agencias del gobierno comenzaron a tener interés en esta tecnología. La Autoridad de puertos en Nueva York y Nueva Jersey experimentaron con aplicaciones de transporte desarrolladas por GE, Westinghouse, Philips y Glenayre, sin embargo esta tecnología no fue adoptada. La administración Federal de Autopistas de U.S. convocó a una conferencia para explorar el uso de RFID en aplicaciones para vehículos y transportes.

El número de compañías enfocadas en la tecnología RFID emergieron. Para el final de la década mucha de la investigación en Radio Frecuencia, electrónica y electromagnetismo se aplicó a RFID en conjunto con el desarrollo de las computadoras y las tecnologías de la información, naciendo así la PC y la red ARPENET.

1980- Comercialización

Durante esta década se dio el primer éxito comercial en los sistemas RFID. Eran sistemas muy sencillos como el manejo de inventarios, sistemas de acceso de personal y cerraduras sin llave. La Asociación Americana de Ferrocarriles y el Programa Cooperativo de Manejo de Contenedores se volvieron activas ante las iniciativas de RFID, instalando sistemas en los ferrocarriles.

La primera aplicación en el mundo se implementó en Normandía en 1987, seguida en Dallas en 1989. La Autoridad Portuaria en Nueva York y Nueva Jersey implementó un proyecto de tracio de pasajeros a través del Túnel Lincoln.

Todos los sistemas RFID implementados en esta década fueron sistemas propios, no existía interoperabilidad entre estos sistemas, lo que no dio una alta competitividad en comparación con otros sistemas y esto elevó sus costos haciendo que esta industria no creciera.

1990- RFID entra en el Mainstream

Los 90´s fueron buenos ya que RFID finalmente comenzó a entrar al mercado mainstream de los negocios y la tecnología. Para la mitad de la década los sistemas RFID podían operar en autopistas para controlar los límites de velocidad. Adicionalmente fue posible utilizar cámaras de video en conjunto con estos sistemas. Esto logró que se generara una mayor investigación dentro de los Estados Unidos.

Texas Instruments comenzaron su sistema TIRIS. Este sistema desarrollo nuevas aplicaciones para RFID para la recarga de combustible en expendios ExxonMobil´s Speedpass en donde se generaba un pase de acceso para pasajeros o para vehículos. Muchas compañías en Estados Unidos y Europa se involucraron en RFID como Philips, Mikron, Alcatel y Bosch.

Las investigaciones de las Tecnologías de la Información se desarrollaron en conjunto durante esta década proliferando el uso de la PC y el Internet. Esto dejo a la tecnología RFID con el único problema de que los costos de los Tags aún eran muy altos para que los sistemas fueran viables. Los avances en la tecnología de materiales durante esta década de relacionaron con trabajos de fabricación de chips de semiconductor por parte de IBM, Intel, AMD y Motorola, finalmente esto generó que los costos en los Tags bajaran. Las inversiones de capital dentro de estos sistemas comenzaron a fluir y se logró comenzar a realizar pruebas a grandes escalas para el final de esta década.

Hasta esta década todos los sistemas de RFID eran propios y no contaban con interoperabilidad, esto provocó que muchas compañías reconocieran esto como un barrera para su crecimiento y comenzaron a trabajar para estandarizar los sistemas. La CEPT[33] y la ISO fueron los organismos encargados de regular los estándares y se creó el Centro Auto-ID en el M.I.T. para este propósito. Actualmente estas son las organizaciones que trabajan sobre los nuevos estándares y el manejo de aplicaciones basados en esta tecnología.

2000 – Despliegue de RFID

Para esta década quedó claro que Tags de un valor de $0.05 hacían posible que esta tecnología pudiera remplazar al sistema de códigos de barra. Las implicaciones para la distribución y la rentabilidad de la industria generaron mula atención de muchos. El 2003

Fue un año clave para RFID, Wal-Mart y el DoD[34] quienes son grandes minoristas y la cadena de suministro más grande del mundo respectivamente, solicitaron que se emplearan sistemas RFID en sus operaciones para el 2005. El tamaño de estas operaciones constituye un enorme mercado para RFID. Otros minoristas y muchas industrias de manufactura como Target, Proctor & Gamble y Gillete las siguieron.

En 2003 el Centro de Auto-ID se fusionó con EPC global, uniendo las empresas con el UPCC[35], quienes hicieron en símbolo del código de barras. Finalmente RFID tenía una plataforma común, el estándar desarrollado por la EPC fue adoptado por ISO en 2006, esto le dio a la industria RFID una sola fuente para guiarse. La convergencia de todos los sistemas incrementó la competencia entre todas las compañías bajando así los costos de RFID y generando un rápido desarrollo de esta tecnología.

[33] CEPT: *European Conference of Postal and Telecommunications Administrations*, "Conferencia Europea de Correos y Administración de Telecomunicaciones".
[34] DoD: *Department of Defense*, "Departamento de Defensa".
[35] UPCC: *Uniform Product Code Council*, "Consejo de Código de Producto Uniforme".

Para 2004 el número de tecnologías piloto de RFID aumentó rápidamente y los participantes ganaron mucha experiencia en esta tecnología, a finales de ese año la ECP publicó el estándar Clase I para la Generación 2 y se creó una legislación para la banda UHF.

En 2005 el EPCglobal se volvió completamente operacional, se tuvo acceso a productos en la banda de VHF y los vendedores ofrecían tarjetas y placas con RFID integradas.

Para 2006 EPCglobal fue adoptado por ISO, todas las tecnologías pasadas de RFID fueron escaladas y reguladas. Aumentó el número de proyectos piloto y desarrollo de aplicaciones.

En 2007 los precios de los RFID pasivos continuaron bajando por debajo de los 5 centavos y las aplicaciones que comenzaron su desarrollo finalmente completaron sus investigaciones y se implementaron sus programas para aplicaciones de logística.

RFID en la Actualidad

Mejoran la experiencia del paciente con Tags que les comunican a los doctores los datos del tratamiento y progreso del paciente al mismo tiempo que almacenan información acerca de los gustos y preferencias del propio paciente.

RFID para gestión de documentos donde hay chips incrustados en los documentos de la oficina del Fiscal General de Florida.

Limpiar material peligroso con RFID donde los camiones que transportan material radioactivo llevan chips para ser rastreados.

La marca de tablas de snowboard Burton y Nokia se han unido para rastrear los datos de los trucos de snowboard recopilados a través de sensores RFID y compartirlos vía Twitter y otras redes sociales.

RFID y motores Mud en la extracción de petróleo en el derrame de BP.

Una ciudad Danesa ha colocado lectores de RFID en las intersecciones de tráfico más conflictivas para leer las etiquetas que los ciclistas de la ciudad llevan en el cuadro.

Sensores en tumores cancerígenos ayudan a que la cirugía sea más exacta.

3.4 Frecuencia

Una de las consideraciones de un sistema RFID es la frecuencia de operación. Tal como sucede en los sistemas de televisión, puede trabajar en las bandas de VHF y UHF.

En RFID hay dos frecuencias bajas y dos altas dentro del espectro radioeléctrico.

Bandas de frecuencia bajas para RFID

- Frecuencias Bajas (LF): 125-134 KHz
- Frecuencias Altas (HF): 13.56 MHz

Bandas de frecuencias altas para RFID

- Frecuencias Ultra Altas (UHF): 860-960 MHz
- Microondas: 2.5 GHZ en adelante

La selección de frecuencias afecta las características de cualquier sistema RFID

3.4.1 Rango de lectura

En frecuencias bajas los rangos de lectura de un Tag pasivo no llega a ser mayor que unos cuantos centímetros. En frecuencias mayores el rango aumenta y más si se utilizan Tags Activos, pero al tener consecuencias sobre la salud el uso de estas frecuencias se ha regulado un límite de potencia en las bandas UHF y las Microondas.

3.4.2 Tags Pasivos vs. Tags Activos

Históricamente los Tags pasivos operan en las Frecuencias Bajas, mientras que los Tags Activos operan en las Frecuencias Altas.

3.4.3 Interferencias con otros Sistemas de Radio

Los sistemas RFID están propensos a interferencias de otros sistemas. Los sistemas RFID que operan en la banda de Frecuencias Bajas (LF) son particularmente vulnerables debido a que no tienen muchas pérdidas en distancias cortas en comparación con frecuencias más altas. En la parte de las Microondas, los sistemas presentan mayores pérdidas y se necesita una interacción casi a línea de vista de otros sistemas para que interfieran.

3.4.4 Líquidos y Metales

El desempeño de los sistemas RFID puede ser afectado por el agua o la humedad en las superficies. Las señales HF en sus respectivas longitudes de onda, son más capaces de penetrar el agua que las UHF y las Microondas lo que produce que sean absorbidas por el líquido.

El metal es un reflector electromagnético, por lo que las señales electromagnéticas no pueden penetrarlo. Esto no solo genera una obstrucción entre el Tag y el lector, sino que un metal cerca del sistema puede alterar las características de propagación en las antenas. La presencia de metales afecta más a las Frecuencias Altas que a las Frecuencias Bajas.

3.4.5 Velocidades de Transmisión.

Los sistemas RFID que operan en Frecuencias Bajas tienen una velocidad de transmisión relativamente baja en el orden de Kb/s. La velocidad de transmisión aumenta con la frecuencia de operación llegando a velocidades de Mb/s en el rango de las Microondas.

3.4.6 Tamaño y precio en los Tags RFID

Los rimeros sistemas RFID fueron en la banda LF por que los Tags utilizados eran los más fáciles de fabricar. Se tuvieron muchos inconvenientes como el tamaño necesario, lo que incrementó los precios. La banda HF es la que más prevalece a nivel mundial debido a que los Tags para HF son más económicos de fabricar que en los Tags para LF. Actualmente los avances tecnológicos en el área han bajado los costos en los Tags UHF y en las Microondas ya que son más pequeños y económicas de producir.

Tabla 3 Caracterización de Sistemas RFID a diferentes Frecuencias [ABI Research.]

Banda de Frecuencia	LF 125KHz	HF 13.56MHz	UHF 860-960 MHz	Microondas 2.5 GHz en adelante
Rango de Lectura (Tag Pasivo)	< 60cm	< 91cm	< 305- 915cm	menor a 305cm
Fuente de alimentación	Pasiva	Pasiva	Activa/Pasiva	Activa/Pasiva
Costos	Costosa	Costosa (menos que LF)	Económica	Económica
Aplicaciones	Llaves de entrada, rastreo de animales, inmobilizadores de automoviles	*Smart Cards,* rastreo de equipajes y bibliotecas	Seguimiento de unidades, manejo de equipajes Telepeaje	Telepeaje
Velocidad de Datos	Lento ———			Rapido
Desempeño ante Metales o Líquidos	Bueno ———			Malo
Tamaño	Grande ———			Pequeño

3.5 Antenas

Debido a las longitudes de onda de las Frecuencias Bajas, el tamaño de las antenas de las bandas LF y HF son mucho más grandes que las de las bandas de UHF y las Microondas. Esto genere un problema para lograr que los Tags sean más pequeños y económicos. Sin embargo muchos diseñadores sacrifican la ganancia en la antena por un diseño donde el costo sea más bajo, donde los resultados inmediatos en un rango bajo de lectura para los sistemas LF y HF. Hay un límite de diseño para las antenas LF y HF, como resultado los Tags LF y HF tienen dimensiones más grandes que los Tags UHF y de Microondas.

La frecuencia de operación nos indicará el tipo de antena que se usará en el sistema RFID. Para la banda de Frecuencias LF y HF se utilizan antenas de acoplamiento inductivo y antenas inductivas, usualmente antenas de lazo**.** Para la banda de frecuencias UHF y Microondas, se utilizan acoplamiento capacitivo siendo antenas dipolo.

3.5.1 Antenas Inductivas

Usadas por las bandas de frecuencias LF y HF, operan por "*flooding"* saturando la zona de lectura con señales uniformes que no diferirán entre el comienzo y el final de otra señal.

3.5.2Antenas dipolo

Usadas por las bandas de frecuencias UHF y Microondas operan señales puntuales del transmisor al receptor. Dadas las cortas longitudes de onda de las señales de alta frecuencia UHF y las Microondas, da lugar a pequeñas ondulaciones en la zona de lectura del Lector, por lo que la potencia no será uniforme del transmisor al receptor apareciendo en algunos puntos atenuaciones a cero, creando **"***nulls***"** o manchas invisibles.

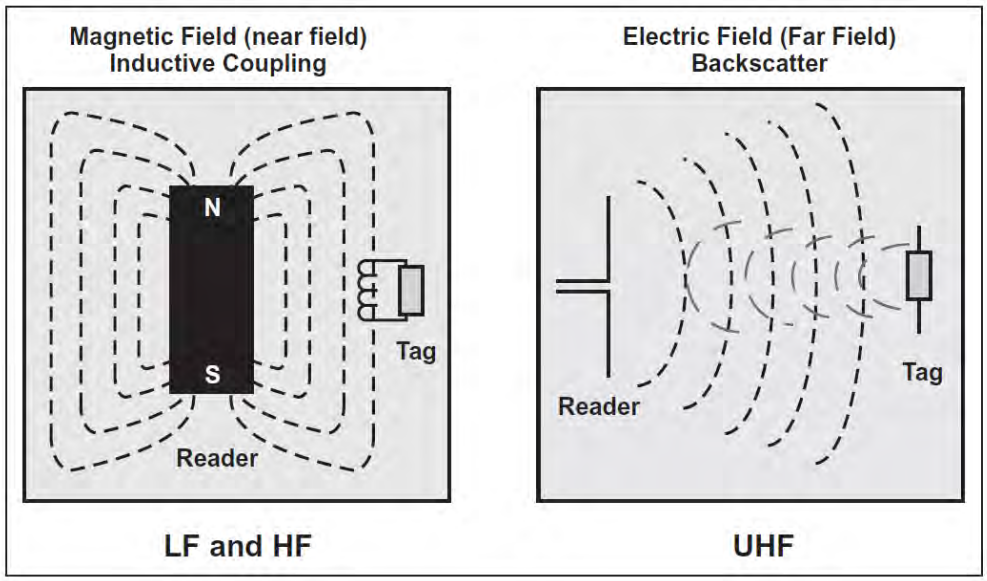


Fig. 3.16 Dos tipos diferentes de acoplamientos de Antenas/Tags. [7]

Los puntos nulos pueden ocurrir por la falta de sintonización de un Tag, lo cual ocurre cuando dos Tags son colocados muy cerca el uno del otro o muy cerca de líquidos, metales u otro material con una permitividad dieléctrica alta.

Los sistemas UHF y de Microondas son muy sensibles a las diferentes orientaciones de la antena. Las antenas inductivas tienen una ganancia direccional pequeña, lo que significa que la potencia de las señales puede presentarse en las mismas distancias sobre, por debajo, al frente o detrás de la antena, las antenas dipolo tienen una ganancia directiva mayor, lo que significa que la potencia de las señales tendrán diferentes potencias con respecto a la posición y dirección de la antena siendo la más óptima una posición frontal, ya que cualquier otra orientación no permitirá una comunicación.

Estos fenómenos requieren que en los sistemas UHF y de Microondas se implemente una forma de modulación más compleja para contrarrestar llamada *Frequency Hopping*.

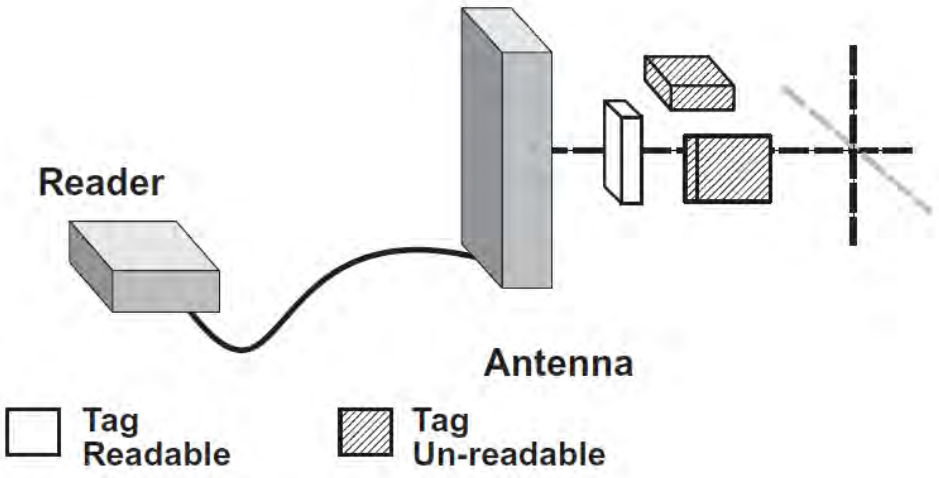


Fig. 3.17 Problemas de orientación de un Tag. [7]

3.6 Codificación

Un sistema codificado toma el mensaje que será transmitido en una señal que lo represente y lo empata en las óptimas características del canal de transmisión. Este proceso provee al mensaje de protección contra interferencias o colisiones y contra modificaciones intencionales de algunas características de la señal (Herter y Lörcher, 1987).

3.6.1 Codificación en Banda Base

Los uno y ceros binarios pueden se representados en varias líneas de código. Los sistemas RFID normalmente utilizan los siguientes códigos: NRZ, Manchester, Unipolar, RZ, DBP (*Differential Bi-Phase*), Miller, Differential Coding.

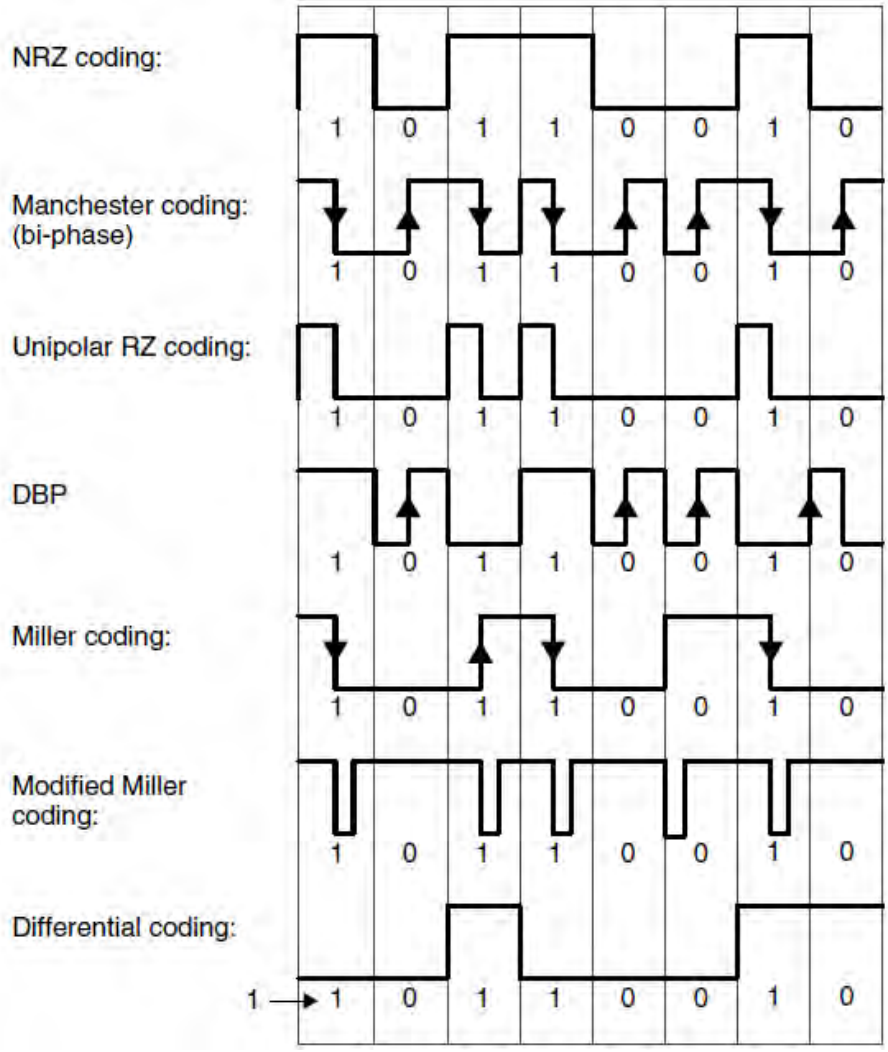


Fig. 3.18 Códigos utilizados en los sistemas RFID. [8]

3.7 Modulación

Es el proceso de alterar los parámetros de una señal con una portadora modificando su frecuencia, su amplitud, fase o una combinación de estas, en relación con la señal que se va a modular. Al analizar las características de una onda electromagnética en cualquier punto de un área podemos reconstruir el mensaje compensando los cambios que se realizaron a la señal sin modular (Portadora). Esto se le conoce como Demodulación.

3.7.1 Métodos de Modulación Digital

Las tecnologías de radio clásicas se manejaban bajo métodos de modulación analógicas tales como AM[36], FM[37] y PM[38]. Todos los demás métodos de modulación se derivan de estos tres.

Los métodos que utilizan los sistemas RFID son digitales como: ASK[39], FSK[40] y PSK[41]

3.7.1.1 Modulación ASK

La señal portadora modifica su amplitud en función de los valores de la amplitud de la señal moduladora para que al final, los valores binarios de la señal moduladora estén representados en la señal modulada por dos amplitudes diferentes de la señal portadora.

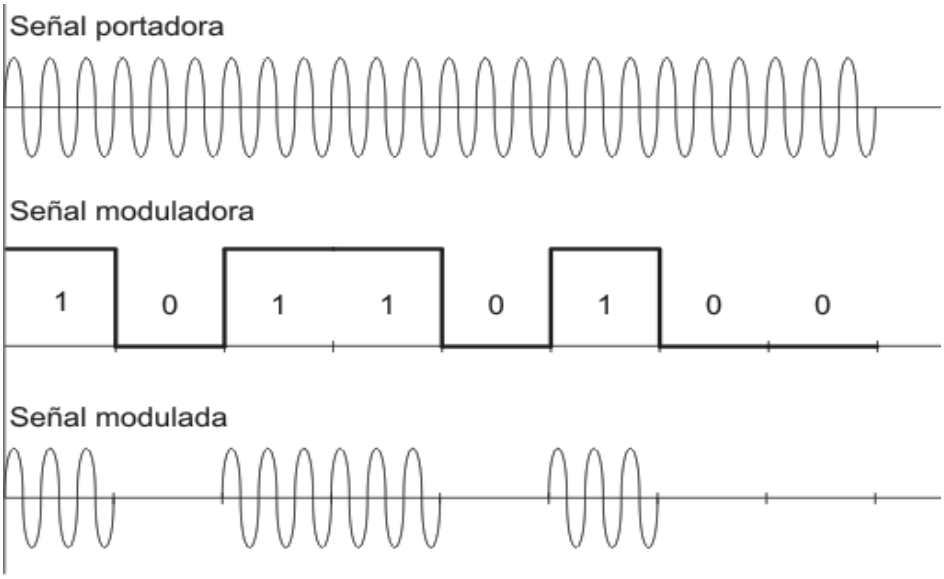


Fig. 3.19 Ejemplo de Modulación ASK [F]

36 AM: *Amplitude Modulation*, "Amplitud modulada".
37 FM: *Frequency Modulation*, "Frecuencia Modulada".
38 PM: *Phase Modulation,* "Fase Modulada".
39 ASK: *Amplitude Shift Keying*, "Desplazamiento en Amplitud".
40 FSK: *Frequency Shift Keying*, "Desplazamiento en Frecuencia."
41 PSK: *Phase Shift Keying*, "Desplazamiento en Fase".

3.7.1.2 Modulación FSK

En esta modulación se emplean dos señales portadoras con frecuencias distintas. Esto genera que los valores binarios de la señal moduladora estarán representados en la señal modulada por dos frecuencias diferentes, cada una de ellas corresponde a cada una de las dos señales portadoras.

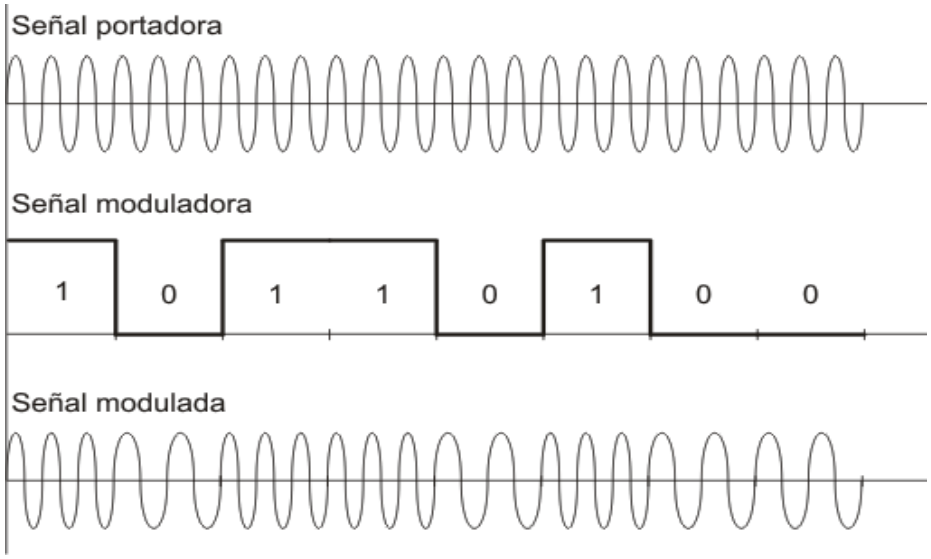


Fig. 3.20 Ejemplo de Modulación FSK [F]

3.7.1.3 Modulación PSK

En esta modulación se emplea una portadora que modifica su fase en función del valor de la señal moduladora que codifica “0” y “1”. Los valores binarios de la señal moduladora están representados en la señal modulada por dos fases diferentes de la señal portadora empleada.

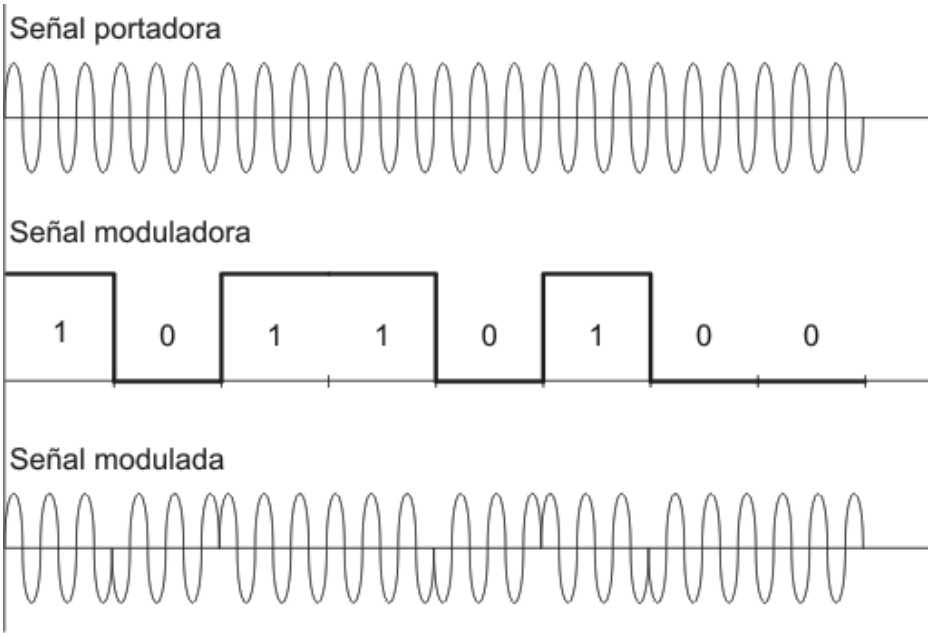


Fig. 3.21 Ejemplo de Modulación PSK [F]

3.8 Data Integrity – Integridad de los datos

3.8.1 Procedimiento de Checksum

Cuando transmitimos datos usando tecnologías Inalámbricas se está propenso a interferencias, esto causa efectos no deseados y cambios en la información al ser trasmitida por el medio, lo que genera pérdida de información y errores.

Un *Checksum* puede utilizarse para reconocer errores en la transmisión e iniciar medidas correctivas, un ejemplo es la retransmisión de los paquetes con errores. Los procedimientos *Checksum* más comunes son: *Parity Check* "Comprobación de Paridad", LRC y CRC.

3.8.1.1 Parity Checking

Este es un método simple y muy popular, se incorpora un bit en cada byte y se transmite con él, transmitiéndose al final un total de 9 bits por cada byte. Antes de que la información transmitida pase por una decisión se necesita verificar que quien envía y recibe la información lo harán bajo el mismo método.

El valor del bit de paridad se establece de tal forma que si hay paridad impar se utiliza un número impar de 9 bits tienen el valor de 1 y si la paridad utilizada da un número par de bits tiene el valor de 1. El bit de paridad par también se puede interpretar como la suma de control horizontal (modulo 2) de bits de datos. Esta suma de control horizontal también permite el cálculo de la puerta exclusiva OR lógica de la información.

La simplicidad de este método se compensa con la baja capacidad de reconocimiento de errores. Un número impar de bits invertidos (1, 3, 5,…) siempre será detectado, pero si hay un número par de bits invertidos (2, 4, 6,…) los errores cancelan cada uno de los otros y el Bit de paridad siempre aparentara ser correcto.

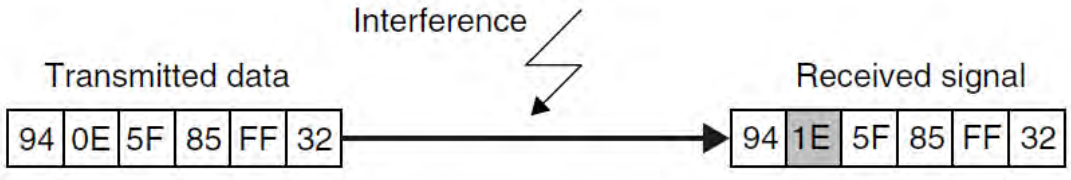


Fig. 3.22 La interferencia durante la transmisión causa errores. [8]

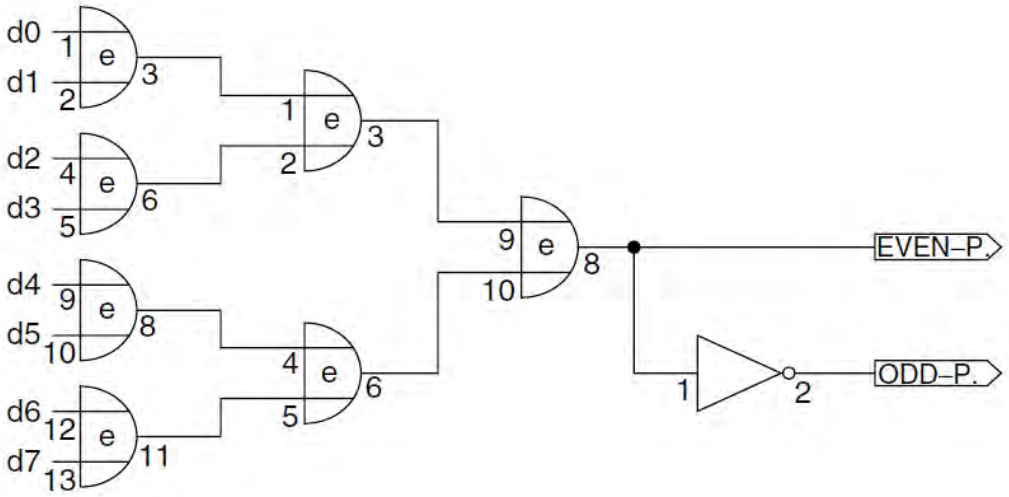


Fig. 3.23 El Bit de Paridad puede ser determinado bajo una serie de Compuertas Lógicas OR. [8]

3.8.1.2 Procedimiento LRC

El procedimiento LRC[42] también es conocido como XOR *Checksum*, es un método muy simple y rápido.

Se genera por la integración de compuertas lógicas XOR donde pasan los paquetes de información, El byte 1 realiza la operación lógica XOR con el byte 2, el resultado realiza una operación lógica con el byte 3, y así sucesivamente. Si el valor de LRC es agregado a la trama de datos y transmitida con ella, se realiza una simple verificación de errores en la transmisión por parte del receptor generando el LRC con la suma de los Paquetes de datos + Bytes LRC. El resultado de esta operación siempre será cero; otro resultado indicará errores en la transmisión.

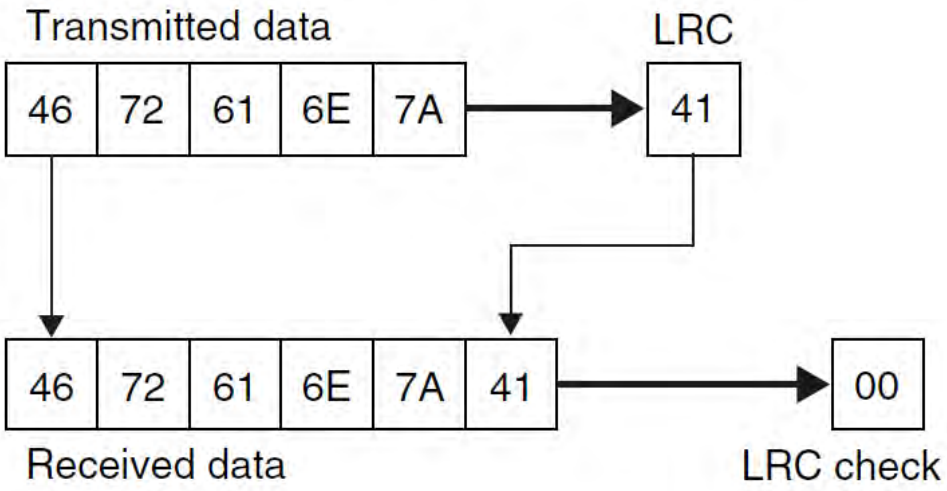


Fig. 3.24 Ejemplo de la verificación por medio del Procedimiento LRC. La suma será cero. [8]

[42] LRC: *Longitudinal Redundancy Check*¸ "Verificación de Redundancia Longitudinal".

Dada la simplicidad del algoritmo, el LRC puede calcularse muy rápido y de una forma muy simple. Sin embargo el LRC no es muy confiable debido a que es propenso a que múltiples errores se cancelen el uno al otro y la verificación no detecte los errores en los bytes transmitidos. LRC generalmente se usa para una verificación rápida en una pequeña trama de datos.

3.8.1.3 Procedimiento CRC

El procedimiento CRC[43]originalmente fue utilizado en unidades de disco**,** y generan un *Checksum* que es lo suficientemente confiable para tramas de datos muy largas. Es excelente para reconocimiento de errores en los datos transferidos por medio de par trenzado (teléfono) o interfaces inalámbricas (radio, RFID), sin embargo no corrige errores.

Como el nombre lo dice, el cálculo del CRC es un procedimiento cíclico. Para el cálculo del valor del CRC se incorpora el valor CRC de la trama de datos que será calculada más el valor CRC de las tramas de datos previas. Cada uno de los bytes en la trama es verificado para obtener el valor CRC de toda la trama.

Para calcular el valor CRC de toda la trama, el valor CRC del byte anterior es usado como el valor inicial de la subsecuente trama de datos.

Si el valor CRC que fue calculado es anexado al final de la trama de datos y se realiza un nuevo cálculo CRC, entonces el nuevo valor CRC obtenido será cero. Esta opción en particular del algoritmo CRC es utilizada para detectar errores en una transmisión de datos seriales.

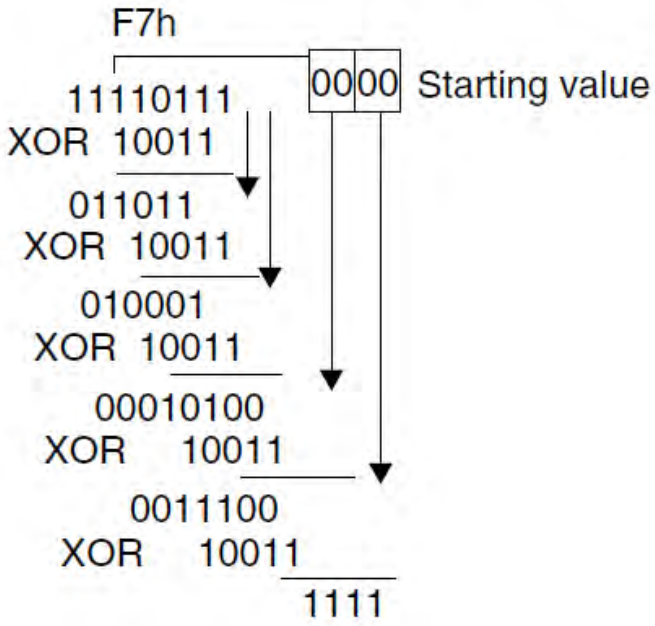


Fig. 3.25 Calculo de la verificación CRC paso a paso. [8]

[43] CRC: *Ciclic Redundancy Check*, "Verificación de Redundancia Cíclica".

Cuando una trama se transmite el valor de CRC es calculado junto con el transmisor y este valor se adjunta al final de la trama y se transmite. El valor CRC recibido incluido el byte CRC adjunto, es calculado en el receptor. El resultado siempre será cero, al menos que se presenten errores en la trama recibida. Verificar en cero es un método muy sencillo para analizar el *Checksum* CRC y evitar los procesos costosos de *Checksum* comparativos. Sin embargo es necesario asegurar que ambos cálculos CRC comienzan del mismo valor inicial.

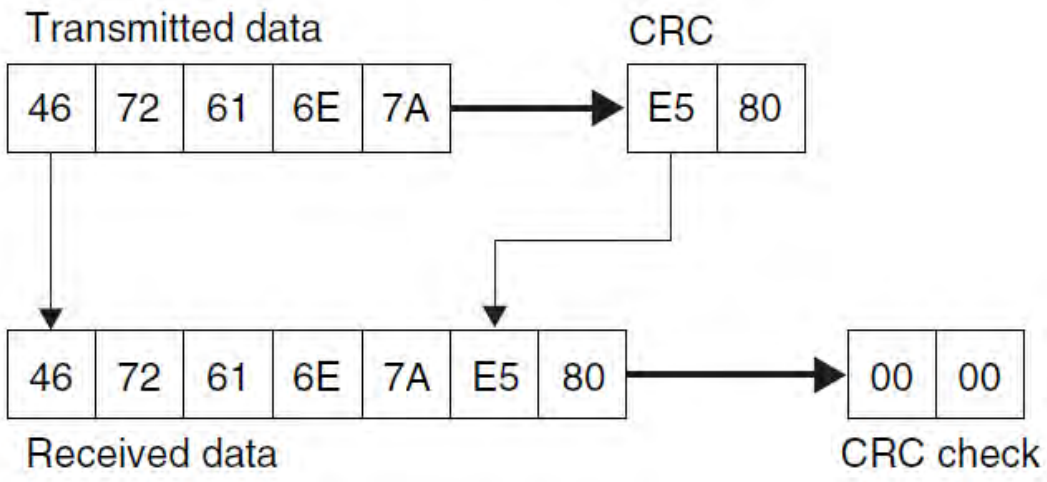


Fig. 3.26 Cálculo del CRC y la suma de comprobación. [8]

3.9 Métodos de Acceso Múltiple – Anticolisión

El uso de sistemas RFID involucran situaciones en las cuales varios Tags se encuentran presentes al mismo tiempo dentro de la zona de lectura de un mismo Lector. En estas situaciones podemos diferenciar dos formas de comunicación.

La primera consiste en transmitir información de los Lectores hacia los Tags. La información transmitida es recibida simultáneamente por todos los Tags. Esta recepción se le conoce como *Broadcast*.

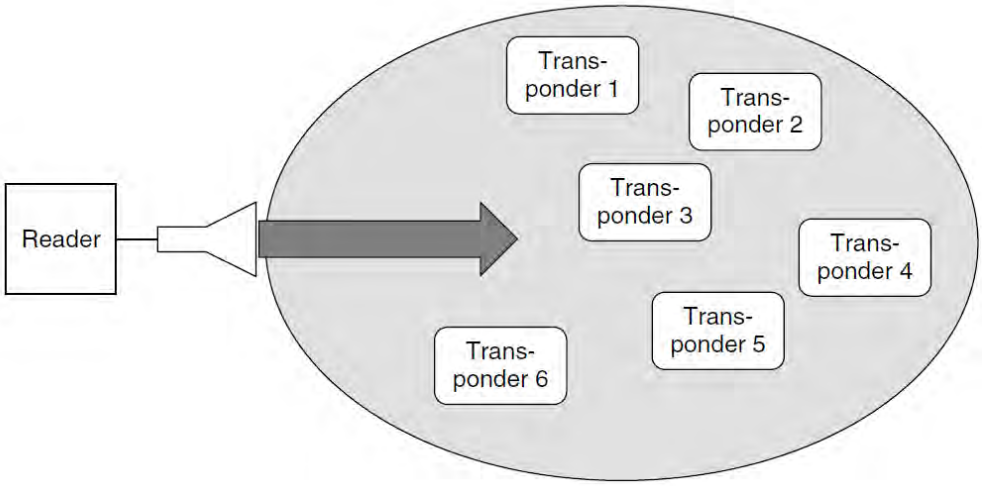


Fig. 3.27 Modo de transmisión tipo Broadcast. [8]

La segunda forma de comunicación involucra la transmisión individual de datos de muchos Tags dentro de la zona de lectura. Esta forma de comunicación se conoce como *Multi-access* "Acceso-múltiple".

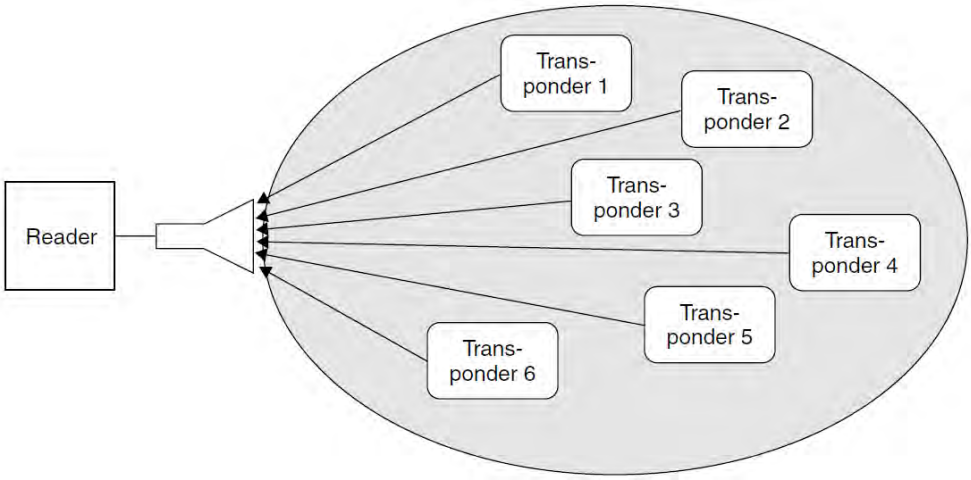


Fig. 3.28 Modo de transmisión tipo Multi-Access. [8]

Cada canal de comunicación tiene una capacidad de canal que determina la tasa máxima de transferencia datos para cada canal y el tiempo de disponibilidad. La capacidad disponible del canal se divide entre cada uno de los participantes (Tags) para que esa información pueda ser transferida de varios Tags hacia un solo Lector sin que se presente mutua interferencia (colisiones). La tasa máxima de transferencia de datos depende del ancho de banda efectivo de las antenas en el Tag y en el Lector.

El problema de los acceso múltiples ha estado presente desde hace mucho en las tecnologías de radio. Para evitar inconvenientes y poder separar cada una de las señales una de otra, se han desarrollado básicamente cuatro procedimientos diferentes:

- **SDMA** (*Space Division Multiple Access*, "Multiple acceso por División de Espacio")
- **FDMA** (*Frecuency Division Multiple Access*, "Multiple acceso por División de Frecuencia")
- **TDMA** (*Time Division Multiple Access*, "Multiple acceso por División de Tiempo")
- **CDMA** (*Code Division Multiple Access*, "Multiple acceso por División de Código")

Sin embargo estos procedimientos clásicos están basados en la asimilación ininterrumpida de datos de los participantes.

Los Tags están caracterizados para tener actividad un corto periodo de tiempo. Una *Smart Card* que entra en contacto con un Lector puede autentificarse, leerse y sobrescribir información en aproximadamente 10 mili segundos (dependiendo el Lector). Cuando se realizan lecturas de una sola tarjeta separada por intervalos cortos de tiempo no es necesario hablar de un sistema anticolisiones, pero cuando un usuario tiene 3 tarjetas o más del mismo tipo y entra dentro de la zona de lectura aquí se pueden presentar complicaciones. Un poderoso procedimiento para acceso múltiple es capaz de seleccionar la tarjeta correcta ante las otras sin ningún retraso.

La relación técnica de los procedimientos para el multi-acceso en los sistemas RFID contiene muchos retos para los Tags y los Lectores.

Hay que prevenir que la información del Tag no colapse con la información de otros Tags que van hacia el mismo Lector y se pierda en el camino. Para esto los sistemas RFID cuentan con protocolos de acceso que facilitan el manejo de accesos múltiples sin interferencias llamado *Anticollision System,* "Sistema Anticolisiones".

Al mandar información del lector hacia un solo Tag y que los otros Tag´s que se encuentran dentro de la zona de lectura no reciban esta información en otro reto importante dentro de todos los sistemas RFID. Un Tag en primera instancia no puede detectar la presencia de otros Tag´s dentro de la zona de lectura.

Por razones de competencia entre las empresas que desarrollan y manufacturan estos sistemas no se publican los procedimientos anticolisiones que manejan, poco de esto se conoce a nivel de literatura técnica, sin embargo se hablará sobre los procedimientos encontrados.

3.9.1 SDMA

Consiste en una técnica que utiliza cierta parte de los recursos del canal en áreas separadas espacialmente. Reduce significativamente el rango de un Lector compensándolo con un arreglo de varios Lectores y antenas para tener cobertura en el área necesaria. Como resultado la capacidad del canal al adjuntar lectores es repetidamente puesta a disposición**.** Estos procedimientos han sido utilizados con éxito en eventos de maratón, donde los corredores están equipados con Tags. Esta aplicación utiliza varias antenas lectoras en una alfombra de tartán. Un corredor viaja sobre la alfombra y pasa por varias zonas de lectura. Como resultado de la distribución espacial de los corredores durante todo el traslado se pueden leer de forma simultánea.

Otra opción es utilizar antenas direccionales controladas electrónicamente por el lector donde el haz puede apuntar direccionalmente a un Tag (SDMA Adaptativo). De esta manera varios Tags pueden ser diferenciados por su posición angular con respecto a la zona de lectura. Se utilizan antenas en fase como antenas direccionales controladas electrónicamente. Esto consiste en que varias antenas dipolo que al ser SDMA Adaptativo sólo se utiliza para aplicaciones RFID en frecuencias por encima de 850 MHz (2.45 Ghz) que se refleja en el tamaño de las antenas. En ciertas direcciones de los campos individuales de la antena dipolo se superponen en fase generando una amplificación en el campo. En otras direcciones las ondas borran total o parcialmente la una con otra. Para establecer la dirección los elementos individuales suministran una tensión y una fase variable controlada. Para hacer frente a un Tag, el espacio alrededor del Lector debe ser escaneado usando la antena direccional hasta que se detecte uno.

Una desventaja de la técnica SDMA es el alto costo del sistema de antenas utilizado. Por lo que el procedimiento anti colisión se limita solo a ciertas aplicaciones especializadas.

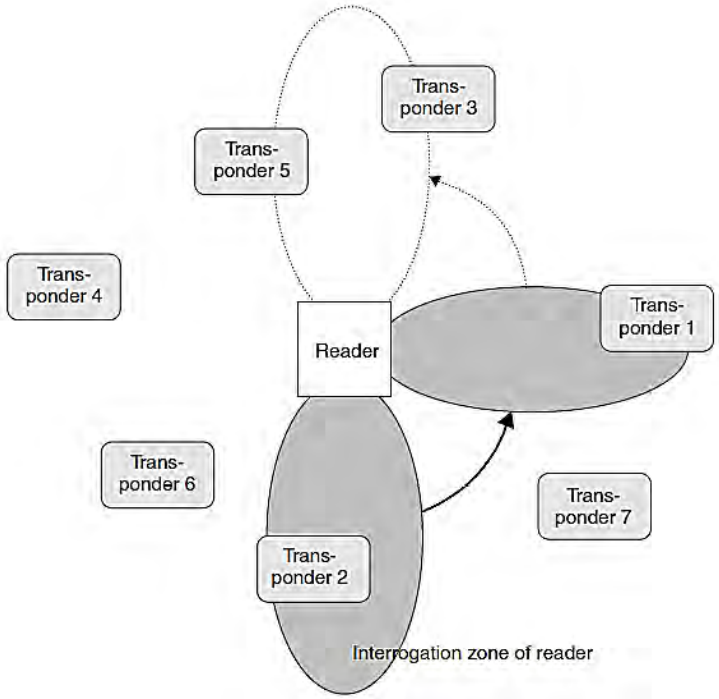


Fig. 3.29 SDMA Adaptativo con control direccional electrónico para la antena [8]

3.9.2 FDMA

Es una técnica en la que varios canales de transmisión están simultáneamente a disposición por el uso de portadoras con diferentes frecuencias a disposición de los participantes en la comunicación.

En los sistemas RFID, se puede lograr utilizando Tags con un ajustador libre de frecuencia inarmónica de transmisión. LA fuente de alimentación para el transpondedor y la transmisión de señales de control (broadcast) tienen lugar en el lector que está óptimamente adaptado a la frecuencia. Los Tags responden en una de las varias frecuencias de respuesta disponible, esto genera diferentes rangos de frecuencias que pueden ser utilizados pata la transferencia de datos hacia los Tags, esto se logra con rangos de frecuencia específicos para un enlace ascendente y una para el enlace descendente.

Una desventaja de este procedimiento es el alto costo de los Lectores, ya que desde un receptor específico se debe proporcionar un canal distinto. Este procedimiento sigue siendo muy limitado a unas pocas aplicaciones especializadas

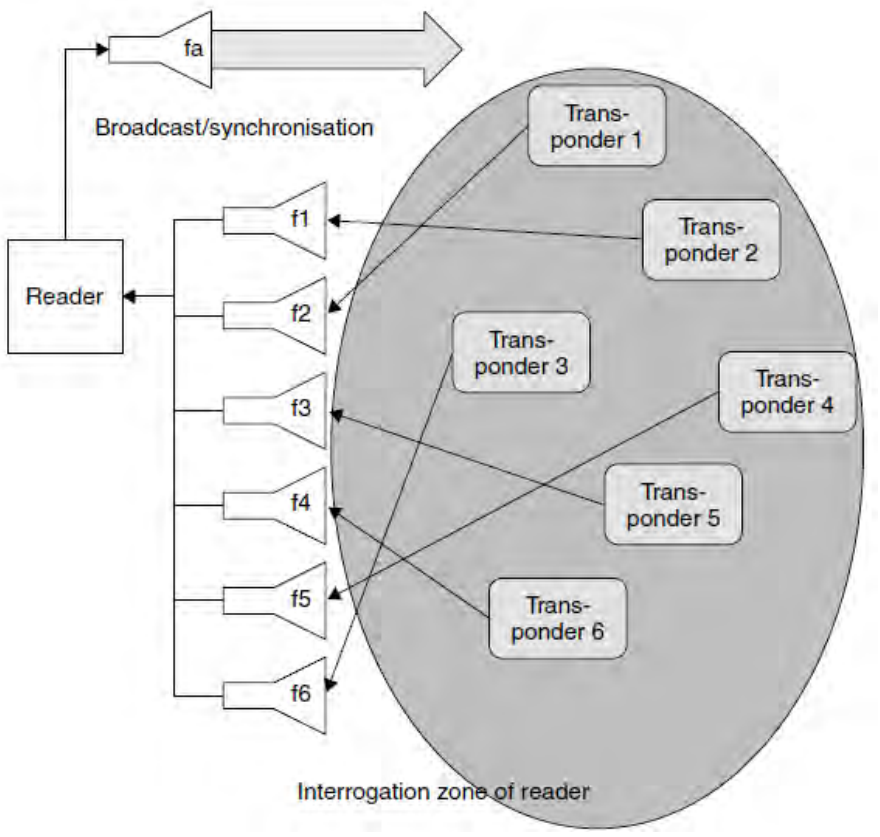


Fig. 3.30 En FDMA varios canales de frecuencia están disponibles para la transferencia de datos [8]

3.9.3 TDMA

Es una técnica donde toda la capacidad del canal es dividida cronológicamente entre los participantes. Los procedimientos TDMA son muy utilizados en el campo de los sistemas móviles digitales. En los sistemas RFID son los procedimiento anticolisión más utilizado. Existen procedimientos síncronos y asíncronos

Para Los Tags asíncronos el lector no controla la transferencia de datos por lo que se utiliza el procedimiento ALOHA. Dentro de esta clasificación diferenciamos entre "S*witched off*" y "N*on-switched*" dependiendo si el Tag se enciende/apaga por una señal del lector después de una transferencia exitosa.

Los Tags asíncronos naturalmente son muy lentos e inflexibles, la mayoría de las aplicaciones utilizan procedimientos controlados por el Lector (síncrono). Estos procedimientos son considerados síncronos ya que todos los Tags son controlados y verificados por el lector de forma simultánea. Un solo Tag es primero seleccionado mediante un algoritmo de entre un gran número de Tas dentro de la zona de lectura, se efectúa la acción (lectura/escritura) y posteriormente la comunicación entre ese Tag y el Lector termina para dar paso a la selección de otro Tag. Dado que la relación lectura/escritura se inicia en cualquier momento, los Tags pueden ser operados rápidamente, esto se le conoce como un procedimiento dúplex en el tiempo.

Los procedimientos síncronos de subdividen en "*Polling*" y "*Binary Search*". En estos procedimientos los Tags se identifican por un único número de serie.

Para el *Polling* se requiere una lista de todos los números de serie de los Tags que se ocuparan. Todos estos números de serie son solicitados una vez por el Lector uno tras otro hasta que un Tag con el número de serie idéntico responde. Este procedimiento puede ser muy lento dependiendo el número de posibles Tags, por lo que solo es adecuado para aplicaciones con pocos Tags.

El *Binary Search* es el procedimiento más flexible y por lo tanto el más común. En este procedimiento un Tag se selecciona de un grupo y se causa intencionalmente una colisión de los números de serie del Tag hacia el lector para que este realice un comando de solicitud. Si el procedimiento es exitoso, es fundamental que el Lector sea capaz de determinar la posición del bit de colisión utilizando un sistema de codificación de señal adecuado.

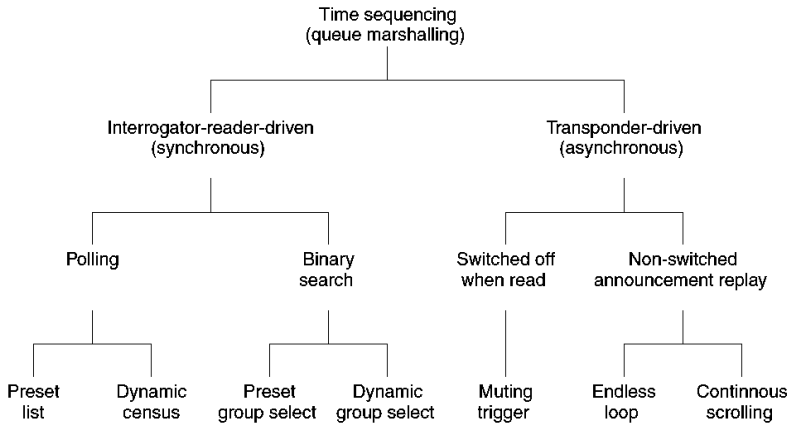


Fig 3.31 Clasificación de TDMA deacuerdo a Hawkes (1997) [8]

3.10 Diferencia entre NFC y RFID

NFC (Near Field Communications) es un subconjunto de RFID que limita el rango de alcance en 10cm.

RFID usa ondas de radio frecuencia que pueden ser pasivas, activas o una combinación de ambas. Los Tags RFID generan la corriente eléctrica necesaria para que los dispositivos pasivos que se encuentran en su alcance puedan recibirla, activarse y comenzar el proceso de transmisión de información.

Las ondas de radio pueden transmitir información a largas distancias y ya que RFID utiliza estas ondas como medio de propagación, esta funcionalidad puede ser útil en algunas situaciones, pero en otras como las tarjetas de identificación, tarjetas de crédito o los pasaportes digitales no es deseable que cualquier persona pueda recibir la información de un usuario y clonarla en su dispositivo, para estas circunstancias se diseñó NFC.

Las etiquetas basadas en NFC generalmente son pasivas, y no requieren mucha potencia eléctrica para funcionar, muchas de ellas se desarrollan con un "escudo" que impide que las ondas se propaguen. Esta limitación aporta una seguridad extra y está iniciando un cambio social que ayuda a la implementación masiva en un futuro muy cercano.

En la actualidad, cada vez más fabricantes están equipando sus teléfonos móviles con emisores/receptores activos, que pueden extender enormemente la popularidad y la gama de servicios que RFID puede aportar.

3.11 Near Field Communication (NFC)

La tecnología NFC de las siglas *Near Field Communication*, "Comunicación en Campo Cercano" ofrece nuevas funcionalidades hacia la tecnología RFID, ésta combina la etiqueta y un lector RFID en un mismo dispositivo. Este hecho facilita la comunicación bidireccional entre dos dispositivos actuando. [9]

NFC surgió en el año 2002 como resultado de la cooperación entre Philips, Sony y posteriormente Nokia. Se trata de un estándar ISO, ECMA y ETSI que trabaja en la banda de 13.56 MHz y con un rango de cobertura pequeño menor a 10 cm. Actualmente ofrece velocidades de transmisión de datos de 106Kbps, 212 Kbps y 424 Kbps. No está pensado para transmitir grandes volúmenes de información, sino para intercambiar información de forma rápida, eficiente y segura.

Al igual que las demás tecnologías RFID, el protocolo NFC cubre los modos de operación activo y pasivo.

El NFC Forum desarrolló 4 tipos de etiquetas que todo dispositivo NFC debe soportar:

- **Tipo 1:** Basado en ISO 14443A. Proporcionado por Innovision Research & Tecnology (Topaz). Posee una capacidad de hasta 1 Kb y velocidades de transmisión de 106 Kbps. Son etiquetas de bajo costo.

- **Tipo 2:** Basado en ISO 14443A. Proporcionado por NXP Semiconductors (MIFARE Ultraligth). Posee una capacidad de 0.5 Kb y velocidad similar a las tipo 1. Son de bajo costo.

- **Tipo 3:** Basada en FeliCa derivada de ISO 18092. Proporcionado por Sony. Posee capacidades de hasta 2Kb y velocidades de 212 Kbps. El costo es mayor aunque es muy útil para aplicaciones más complejas.
- **Tipo 4:** Basadas en ISO 14443 A/B. Proporcionado por varios fabricantes. Posee una capacidad de hasta 64 KB y velocidades entre 106 Kbps y 424 Kbps.

Comunicación Pasiva

Este sistema es utilizado para leer y escribir Tags RFID pequeños a 13.56MHz basados en el estándar ISO14443A.

Al operar a 13.56MHz está basado en el modelo de “iniciador” y un “objetivo” donde el iniciador genera un pequeño campo magnético que energiza al objetivo haciendo que este no requiera alguna fuente de alimentación.

Comunicación Activa

También conocida como “Peer-to-peer” es posible cuando ambos dispositivos tienen una fuente de poder y cada uno genera un campo magnético, esto propicia una rotación continua entre los dos dispositivos. Mientras dos dispositivos se encuentran comunicándose solo uno de ellos activa su campo magnético, al finalizar su envío de datos apaga su campo magnético y el segundo lo activa.

3.12 Tarjeta MiFare y Tags

NXP desarrolló la tarjeta MiFare MF1 IC S50 como una tarjeta de lectura sin contacto de acuerdo al ISO/IEC 144443A. La capa de comunicación (interface MiFare) cumple con la 2ª y 3ª parte del estándar ISO/IEC 144443ª. La capa de seguridad utilizada es Crypto1 el cual realiza un cifrado de flujo para el intercambio de datos seguros. [10]

MiFare es uno de los 4 protocolos para tarjetas a 13.56MHz (FeliCa es uno de los otros).

Tarjeta MiFare Clásica

Existen 2 variedades de esta tarjeta 1K y 4K. Dentro de la gran variedad de chips que existen estas tarjetas solo tienen estos dos:

- MF 1S503x MiFare Classic 1K
- MF 1S70yyX MiFare Classic 4K

Las tarjetas MiFare clásicas tienen un ID de 4 bytes para identificar la tarjeta. Es posible tener IDs de 7 bytes pero el más común para estos modelos es el de 4 bytes.

3.12.1 Transferencia de datos y Alimentación sin contacto

En el sistema MiFare el chip MF1 IC S50 está conectado a una bobina con pocas vueltas y se encuentra embebida en plástico para formar la tarjeta. No es necesario el uso de una batería. Cuanto la tarjeta se encuentra dentro del rango de la antena de un Lector la interfaz de comunicación permite transmitir datos a 106 Kbps.

Fig. 3.32 Tarjeta MiFare Clásica

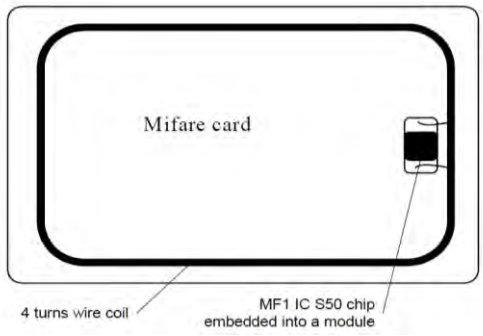


Fig. 3.33 Elementos dentro de una tarjeta MiFare

3.12.2 Anticolisión

Cuenta con un sistema inteligente anticolisiones que permite operar más de una tarjeta simultáneamente. El algoritmo anticolisiones selecciona cada tarjeta individualmente y asegura la ejecución de la transacción con la tarjeta seleccionada sin corrupción en los datos de otras tarjetas en la misma área.

3.12.3 Seguridad

Se ha puesto un gran énfasis en la seguridad contra fraudes. Se ha desarrollado una respuesta de autentificación, cifrado de datos y confirmaciones en mensajes de autentificación para proteger el sistema para cualquier tipo de infiltración, también se cuenta con un número de serie que no puede ser alterado y garantiza que cada tarjeta es única. Esto genera más seguridad en las aplicaciones de identificación y pago.

3.12.4 Interfaz de Radiofrecuencia MiFare (ISO/IEC 14443A)

- Transmisión sin contacto de datos y alimentación de energía (sin baterías)
- Distancia de operación: No mayor a 10 cm. (dependiendo la geometría de la antena)
- Frecuencia de Operación: 13.56 MHz.
- Rápida transferencia de datos: 106 Kbps.
- Alta integridad de datos: 16 Bit CRC, parity, bit coding, bit counting.

3.12.5 Memoria EEPROM

Las tarjetas MiFare Clásicas tienen una memoria EEPROM de 1K o 4 K. Cada bloque de memoria puede ser configurado con diferentes condiciones de acceso con dos llaves de autentificación presentes separadas en cada bloque.

Estas tarjetas están divididas en secciones llamados sectores o bloques. Cada sector tiene derechos de acceso individual y contiene un número fijo de bloques que son controlados para acceder a los derechos. Cada bloque contiene 16 bytes y cada sector contiene 4 bloques (1K/4K) dando un total de 64 bytes por sector o 16 bloques (solo las tarjetas 4K) para un total de 256 bytes por sector. Estas tarjetas se organizan de la siguiente forma.

- Tarjetas 1K - 16 sectores de 4 bloques cada uno (sectores 0…15).
- Tarjetas 4K – 32 sectores de 4 bloques cada uno (sectores 0…31) y 8 sectores de 16 bloques cada uno (sectores (32…39).

3.12.6 Sector de 4 Bloques

Las tarjetas 1K y 4K utilizan 16 sectores de 4 bloques cada uno. Estos 4 sectores de bloques individualmente contienen 64 bytes cada uno, tienen protocolos de seguridad que pueden ser configurados con accesos separados lectura/escritura y dos llaves de autentificación de 6 bytes (la llave puede ser diferente en cada sector). Dada esta seguridad (encontrada almacenada en el bloque 3 llamado "*Sector Trailer*") solo el fondo de cada sector queda disponible para el almacenamiento de datos, siendo esto 48 bytes cada sector de 64 bytes disponibles para uso.

Cada sector de bloques de 4 está organizado con 4 filas de 16 bytes cada uno para un total de 64 bytes por sector. Los primeros dos sectores de cualquier tarjeta se ve de la siguiente manera.

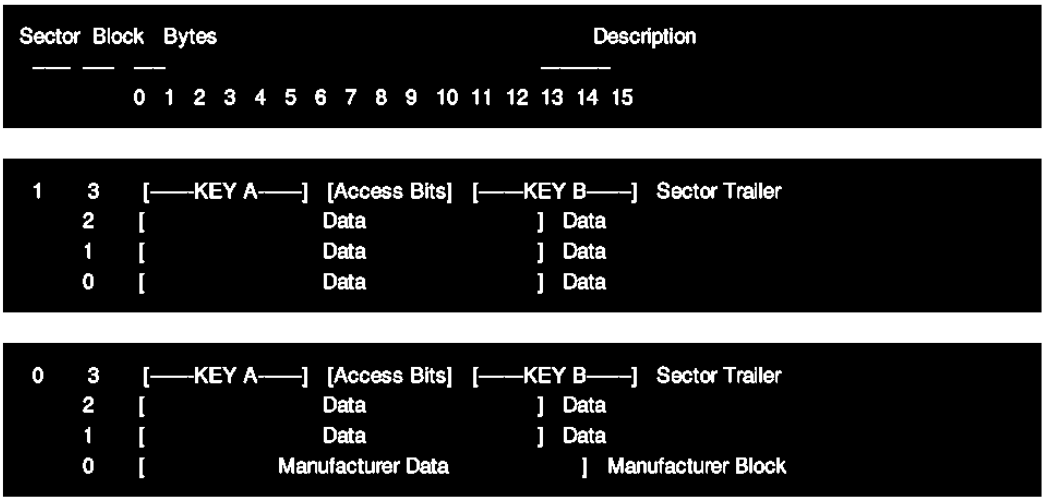


Fig. 3.34 Organización de un sector de 4 bloques

3.12.7 Sectores de 16 Bloques

Son idénticos a los sectores de 4 Bloques pero con más bloques de datos. Tienen la misma estructura descrita por los sectores de 4 Bloques.

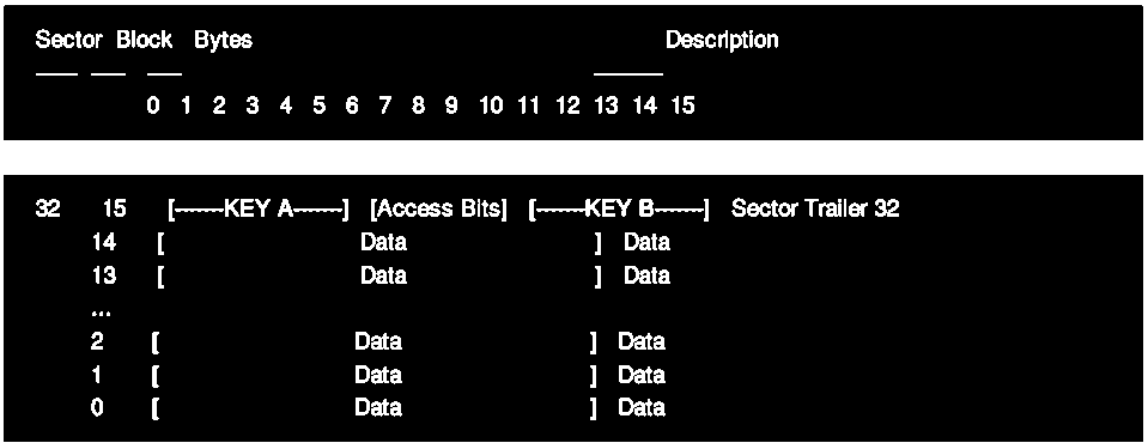


Fig. 3.35 Ejemplo de un sector de 16 Bloques

3.12.8 Ejemplo de una Tarjeta MiFare Clásica 1K Nueva

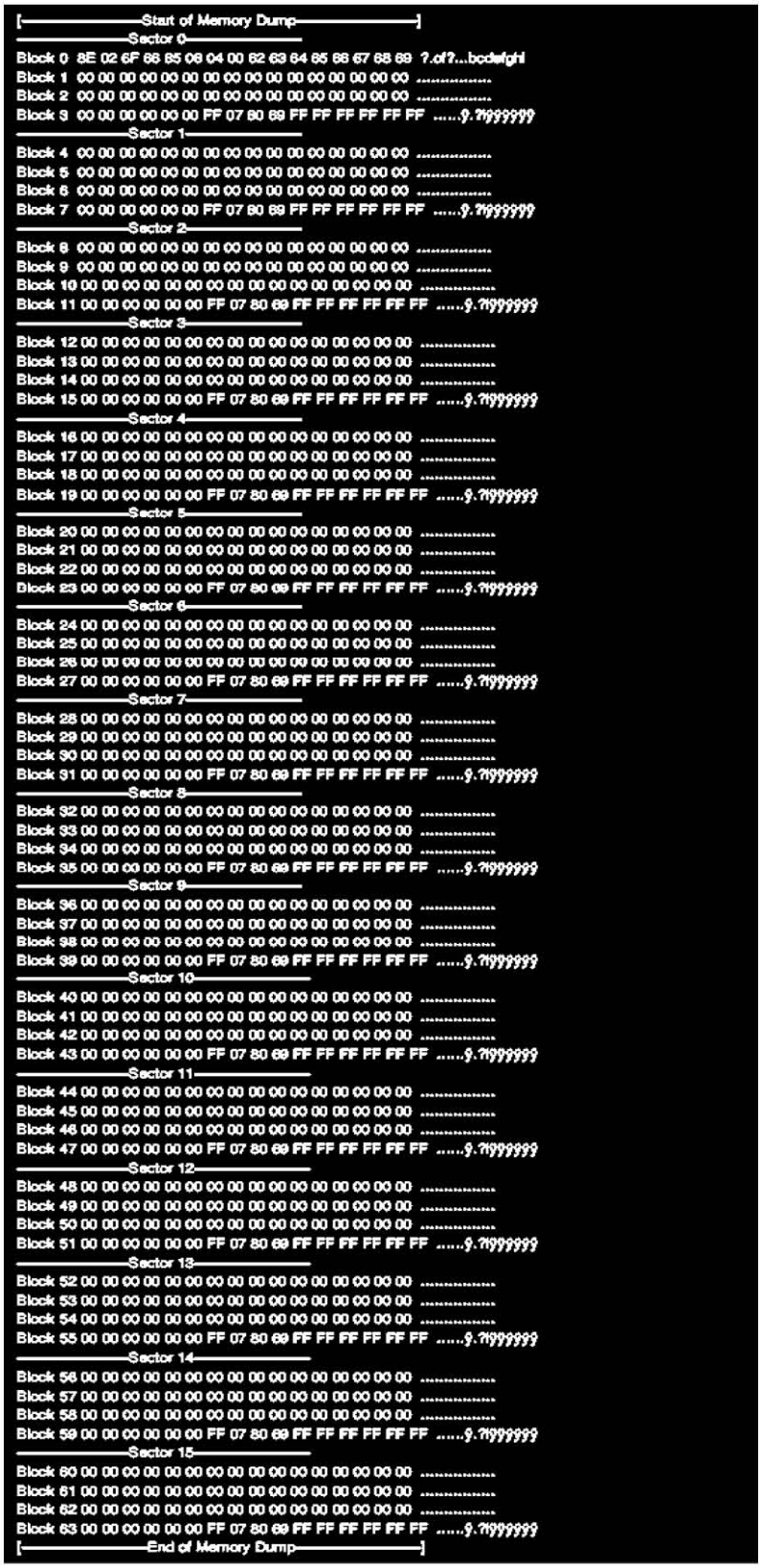


Fig.3.36 Organización de bloques en una tarjeta MiFare

3.12.9 NDEF (NFC Data Exchange Format)

El NDEF[44] es un formato estandarizado utilizado para el intercambio de información entre cualquier dispositivo NFC y otro dispositivo o Tag NFC compatible. El formato de los datos consiste en Mensajes NDEF y Registros NDEF. Este estandar se mantiene por el Forum NFC.

El formato NDEF es utilizado para almacenar e intercambiar información como URLs, textos, etc. Tags NFC como las tarjetas MiFare Clásicas pueden ser configuradas como Tags NDEF y escribir datos a ellos a travez por un dipositivo NFC (Registros NDEF) para que posteriormente pueda se accesado por otro dispositivo NDEF compatible. Los mensajes NDEF pueden se utilizados para intercambiar dato entre dos dispositivos activos en modo “peer-to-peer”. Con este formato durante la comunicación entre dos dispositivos se tiene un conocimiento de un lenguaje comun para compartir información de forma organizada.

De esta manera con los Registros NDEF es posible almacenar en las tarjetas algún identificador alfanúmerico.

[44] NDEF: *NFC Data Exchange Format*, “Formato de Intercambio de Datos NFC”.

Capítulo 4

Implementación de un Sistema RFID/NFC para el control de acceso a pasajeros en un autobús.

4.1 Descripción del problema

Una empresa privada de autobuses dedicada al transporte empresarial necesita más control en el acceso de pasajeros ya que su sistema actual consiste en entregar un ticket al abordar a la unidad y durante todos los viajes que se realizan al día, los tickets se pierden, se desorganizan o inclusive no se sabe si los choferes cumplen con la norma de entrega a los usuarios. La empresa cobra el servicio por pasajero y al no contar con un control preciso del aforo diario, no se tiene un registro confiable del número de los mismos y esto le puede generar pérdidas.

Al analizar el problema anterior se propuso que mediante tecnologías RFID/NFC y tarjetas MIFARE se capturen y se guarden en una memoria SD los datos del operador, el acceso de pasajeros y el estado del autobús agregando una etiqueta con la fecha y la hora del registro. Al llegar a la central destino se descargará por medio de tecnologías inalámbricas (Wi-Fi) esa información en un servidor y en una memoria SD para su posterior análisis.

4.1.1 Solución propuesta

Se sugirió el siguiente esquema de red para la solución del problema planteado en donde se dividirá en tres etapas.

- Sistema de lectura, almacenamiento y transmisión de datos. (Autobús)
- Sistema de recepción y almacenamiento de datos. (Central)
- Dispositivo Tag, tarjeta MIFARE. (Chofer/Maestra) y (Usuario)

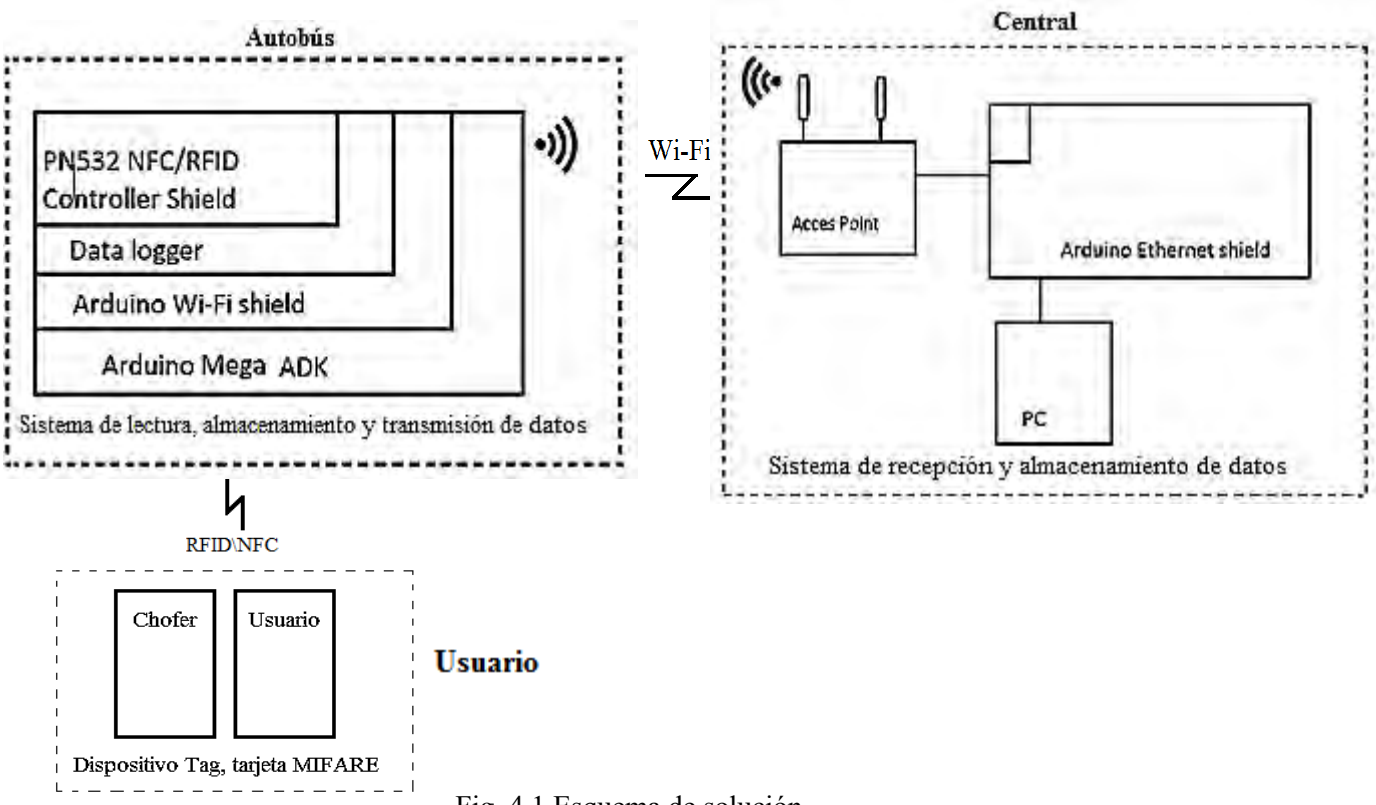


Fig. 4.1 Esquema de solución

4.1.2 Material

La empresa sugirió una lista de materiales con las cuales se esperaba generar la solución pero durante el desarrollo se evaluó el microcontrolador Arduino UNO y el módulo Wi-Fi, ambos no cumplieron con los resultados esperados. En la Tabla 4 y la Tabla 5 se muestran los parámetros que se tomaron en cuenta durante la evaluación.

Para realizar la transferencia de datos entre el autobús y la central, se requiere generar un servidor en donde se guardará momentáneamente la información que se transmite para posteriormente ser recibida y guardada por la central. Las librerías del shield Adafruit no cumplían con estor requerimientos por lo que se optó por utilizar el shield Arduino.

Tabla 4 Adafruit Wi-Fi CC3000 Vs. Arduino Wi-Fi Shield

Módulo Wi-Fi	**Adafruit**	**Arduino**
Tarjeta SD	✓	✓
CC30000 Wi-Fi Chip	✓	x
HDG204 Wi-Fi Chip	x	✓
Estándar 802.11b	✓	✓
Estándar 802.11g	✓	✓
Librería capaz de generar un servidor	**x**	✓
Compatibilidad con el servidor del Shield Ethernet	**x**	✓
Terminal para activar SD	10	4

La extensión del código de programación es de 38Kb y superan de los 24 Kb reales que permite el Arduino UNO, por esta razón se utilizó el Arduino Mega ADK

Tabla 5 Adafruit Wi-Fi CC3000 Vs. Arduino Wi-Fi Shield

Shield	**Mega ADK**	**UNO**
Microcontrolador	Atmega2560	ATmega328
Voltaje de Operación	5V	5V
Voltaje de Entrada	7-12V	7-12V
Terminales Digitales I/O	54 (14 proveen una salida PWM)	14 (6 proveen una salida PWM)
Terminales Analógicos	16	6
Memoria Flash	**256Kb** de los cuales 8 Kb son usados como gestor de arranque.	**32Kb** de los cuales 8 Kb son usados como gestor de arranque.
Velocidad de Reloj	16 MHz	16 MHz

Tomando en cuenta estos resultados se optó por utilizar el shield Wi-Fi de Arduino y el Arduino UNO con los cuales se generó una contrapropuesta.

Tabla 6 Propuesta Inicial Vs. Contrapropuesta

Propuesta inicial	Contrapropuesta
Arduino UNO	Arduino Mega-ADK
Adafruit Wi-Fi CC3300	Arduino Wi-Fi Shield
Adafruit Data Logger Shield	Adafruit Data Logger Shield
Adafruit PN532 NFC/RFID Controller Shield for Arduino	Adafruit PN532 NFC/RFID Controller Shield for Arduino
MiFare Classic 1K tag (Tarjeta)	MiFare Classic 1K tag (Tarjeta)
	Arduino Ethernet Board
	Access Point

A continuación se describirá la funcionalidad de cada dispositivo dentro del Sistema. Para mayor información pasar al Apéndice A.

Autobús:

- **Arduino Mega-ADK:** Es el microcontrolador responsable de mandar los comandos a cada una de las placas, desempeña la función de dispositivo integrador, suministra la energía a toda esta etapa, aquí se encuentra almacenado el código de programación y es utilizado como interfaz directa a la PC.
- **Arduino Wi-Fi Shield:** Este dispositivo es el encargado de generar un servidor y crear una conexión vía Wi-Fi para la transmisión de datos. También cuenta con una ranura para insertar una memoria micro SD donde se guardan los datos recopilados por el sistema.
- **Adafruit Data Logger Shield:** Su función es guardar la fecha y la hora para ser añadida en cada adquisición de datos mediante un RTC.
- **Adafruit PN532 NFC/RFID Controller Shield for Arduino:** Dispositivo utilizado como lector para las tarjetas.

Central:

- **Access Point:** Dispositivo que recibe los datos transmitidos mediante Wi-FI por el autobús. Para las pruebas realizadas se utilizó un Access Point Linksys (Broadband Router)
- **Arduino Ethernet Board:** Recupera los datos recibidos por el Access Point, guarda la información en una memoria micro SD, aquí está guardado el código de programación de esta etapa y cuenta con un microcontrolador.
- **PC:** Aquí podemos visualizar la información presentada a través de un archivo de Excel.

Usuario:

- **MiFare Classic 1K Tag (Tarjeta):** El chofer contará con una tarjeta maestra que al pasarla por el lector por primera vez activara el sistema de recolección de datos y el final del abordaje de todos los pasajeros volverá a pasar su tarjeta por el lector para cerrar el sistema y guardar esa información en la memoria micro SD del sistema del autobús. Los usuarios contaran con una tarjeta personalizada que tendrán que pasar por el lector instalado en el camión para registrar su ingreso.

4.1.3 Software

- **Arduino 1.0.6:** Es la interfaz de programación con el microcontrolador, mediante este software configuramos el puerto de comunicación y el tipo de tarjeta que se estará empleando. El código se escribe y/o modifica a través de este software, por lo que también nos lo compila y lo carga en nuestro microcontrolador. Como última función lo empleamos como un monitor serial para visualizar los procesos y resultados.

- **Microsoft Excel:** Programa utilizado para visualizar los datos guardados por el sistema**.**

4.1.4 Justificación de Dispositivos

Para esta primer etapa de desarrollo del primer prototipo, la empresa solicitó explicitamente el uso de estos dispositivos debido a que son fáciles de adquirir, cuentan con una interfáz de programación muy accesible para pruebas y son plataformas libres para desarrollo.

Dentro de esta familia de dispositivos tuvo que haber algunas modificaciones a lo largo del proceso de diseño debido a limitantes a nivel de hardware e incompatibilidades entre los dispositivos de Arduino y Adafruit.

El primer cambio fue en el Microcontrolador a utilizar, primero se pensó utilizar un Arduino UNO, pero debido a la extensión del código, éste no tenia la capacidad de memoria necesaria para guardar la extensión del código de programacion (32 KB), por lo que se optó por cambiar al Arduino Mega 256 ADK cuya capacidad (256 Kb) logró soportar el código.

El segundo cambio fue en la placa de Wi-Fi, la primer propuesta se desarrolló para una placa Adafruit Wi-Fi CC3300 pero debido a que esta placa no podía generar un servidor para la transmisión de archivos, se optó por utilizar la placa Wi-Fi de Arduino que cuenta con una opción programable de generar un servidor dentro de sus bibliotecas.

4.2 Bibliotecas y Protocolos

Las bibliotecas son archivos que contienen líneas de programación vinculadas a funciones específicas que hacen más sencillo el proceso de programación y simplifican las líneas de código.

Los protocolos son conjuntos de normas y procedimientos estandarizados que se establecen entre dos o más dispositivos para lograr una comunicación exitosa desde su inicio hasta su fin.

4.2.1 Biblioteca SPI

Permite comunicar mediante protocolo SPI el Arduino (al cual actúa siempre como "maestro") con otros dispositivos externos (los cuales actuarán como "esclavos"). La comunicación se establece por medio de:

ETHCS: Controla la conexión Ethernet	PIN 10
MOSI: Línea de envío de datos del maestro al esclavo	PIN 11
MISO: Línea de envió de datos del esclavo al maestro	PIN 12
SCK: Línea de reloj	PIN 13

Esta Biblioteca es relativamente compleja debido a que es capaz de controlar hasta el mínimo detalle del proceso de comunicación establecido con los periféricos SPI [G]

4.2.2 El protocolo SPI

El protocolo SPI (*Serial Peripheral Interface,* "Interfaz Serial Periférica") es un estándar de comunicaciones usado para la transferencia de información entre circuitos integrados. Se trata de un bus de datos en serie para la transferencia síncrona y bidireccional de información. En toda comunicación por SPI deberá haber al menos un dispositivo actuando como maestro, y uno o más actuando como esclavos. Para seleccionar a cada uno de los esclavos existe una línea, denominada "*Slave select*" o "C*hip select*".

Las señales del protocolo SPI son las siguientes:

- **SCLK:** Es la señal de reloj, impuesta por el dispositivo maestro.
- **MOSI:** Corresponde a las siglas "Master Output – Slave Input", es decir, el maestro enviará los datos a través de esta línea y el esclavo los recibirá.
- **MISO:** Corresponde a las siglas "Master Input – Slave Output", y es la línea por la que los esclavos enviarán datos al dispositivo maestro.
- **SS:** Es la señal de "Slave Select", es decir, la línea que el maestro activará para indicar al esclavo que se va a establecer la comunicación con él.

Habitualmente, la terminal MISO del maestro se conecta con la terminal MOSI del esclavo, y viceversa. Además, la señal de selección de esclavo suele ser activa a nivel bajo.

A continuación se va a describir con un poco más de detalle cómo funciona el protocolo:

- Para iniciar la comunicación, el maestro configura el reloj usando una frecuencia menos o igual a la frecuencia máxima que soporta el esclavo. Estas frecuencias suelen estar en el rango de 1 a 70 MHz.
- El maestro pone a nivel bajo la señal "*Slave Select*" del esclavo para indicarle que se va a comunicar con él. Si es necesario esperar un tiempo antes de iniciar la comunicación (por ejemplo para permitir una conversión analógico / digital), el maestro esperará al menos ese tiempo antes de proseguir con el intercambio de información.
- Durante cada ciclo de reloj se produce una comunicación en los dos sentidos, ya que por una parte el maestro va a mandar un bit a través de la terminal MOSI y el esclavo lo va recibir, mientras que a la vez el esclavo va a mandar un bit a través de la línea MISO para que el maestro lo reciba.
- Cuando ya no quedan datos que transmitir, el maestro deja de accionar la señal de reloj y normalmente vuelve a colocar a nivel alto la señal de "*Slave Select*" para así deseleccionar al dispositivo.
- Cualquier dispositivo que no tenga a nivel bajo su señal de selección, ignorará los movimientos que haya en las líneas MISO y MOSI, con lo que podemos tener distintos dispositivos conectados a esas mismas líneas sin que interfieran en la comunicación. Evidentemente, la señal de "*Slave Select*" sí debe ser propia de cada dispositivo.
- Además de la frecuencia de reloj, el maestro también puede configurar la polaridad y la fase de la señal de reloj con respecto a los datos. La polaridad se suele denominar usando las siglas CPOL, mientras que la fase se suele denominar como CPHA.

4.2.3 Protocolo HTTP

El protocolo HTTP[45] es un protocolo cliente-servidor que gestiona los intercambios de información entre los clientes Web y los servidores HTTP. [3]

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL.

[45] HTTP: *Hypertext Transfer Protocol*, "Protocolo de Transferencia de Híper Texto".

Etapas de una transacción HTTP

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo de localización del cliente Web.
- El cliente Web decodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.
- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente.
- Se realiza la petición enviando el comando necesario (GET, POST, HEAD,…), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor,…
- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.
- Se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP. Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.

4.2.4 Protocolo I2C

Se le denomina de esta forma por sus siglas de Inter-Integrated Circuit, es un bus de datos serial desarrollado en 1982 por Philips Semiconductors, principalmente es utilizado para la comunicación entre diferentes partes de un circuito ya sea un controlador o los periféricos.

Este protocolo permite que el microcontrolador pueda controlar toda una red de circuitos integrados con terminales I/O y un software muy simple. Es utilizado para la transmisión de datos de control y configuración.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

Descripción de las señales

- **SCL** (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema.
- **SDA** (System Data) es la línea por la que se mueven los datos entre los dispositivos.
- **GND** (Tierra) común de la interconexión entre todos los dispositivos "enganchados" al bus.

4.3 Ajustes de dispositivos a nivel Hardware

Algunos dispositivos de fábrica vienen con ciertas limitaciones, por lo que hay que ajustarlos a nuestras necesidades, principalmente soldar conectores de aguja paraque sea más facil el montaje entre las placas.

La placa PN532 RFID/NFC de Adafruit no cuenta con conectores de aguja de fábrica, por lo que hay que montarlos para su montaje.

Entre la familia Arduino y Adafruit hay ciertas terminales que no comparten las mismas funciones.

En el caso del Data-logger de Adafruit y la placa Wi-Fi de Arduino cuentan con una ranura para memorias SD y/o micro SD, el problema es que en el Data-logger la terminal que activa la comunicación con la SD es la terminal 10 y en la placa de Arduino Wi-Fi es la terminal 4.

Se debe tomar una desición sobre cual placa va a ser la responsable del almacenamiento de datos. Para nuestro prototipo se decidió utilizar la placa de Arduino Wi-Fi.

Para evitar interferencia entre las dos placas debemos dejar la placa del Data-logger aparte de las demás y hacer un puente con jumpers hacia 4 terminales indispensbles para la comunicación, los dos primeros son los de alimentación (5V y GND), los otros dos son los que realizan la comunicación del RTC con el microcontrolador (SCL y SDA). Otra alternativa es soldar conectores de aguja únicamente en estas terminales o evitar poner conectores de aguja en las terminales 4 y 10 del Data-logger.

Para el proceso de soldado de conectores de aguja para la placa PN532 RFID/NFC se requiere una tira de conectores de aguja, un cautín y soldadura.

4.3.1 Ajustes de la placa PN532 RFID/NFC

1. Cortamos tiras de 6, 8, 8 y 10 conectores de aguja

Fig. 4.2 [10]

2. Montamos la parte más larga de los conectores de aguja a una placa de arduino en su respectivo orden y sobremontamos nuestra placa PN532 RFID/NFC.

Fig. 4.3 [10]

3. Soldamos con mucho cuidado los conectores de aguja a la placa evitando que con la soldadura se unan las terminales.

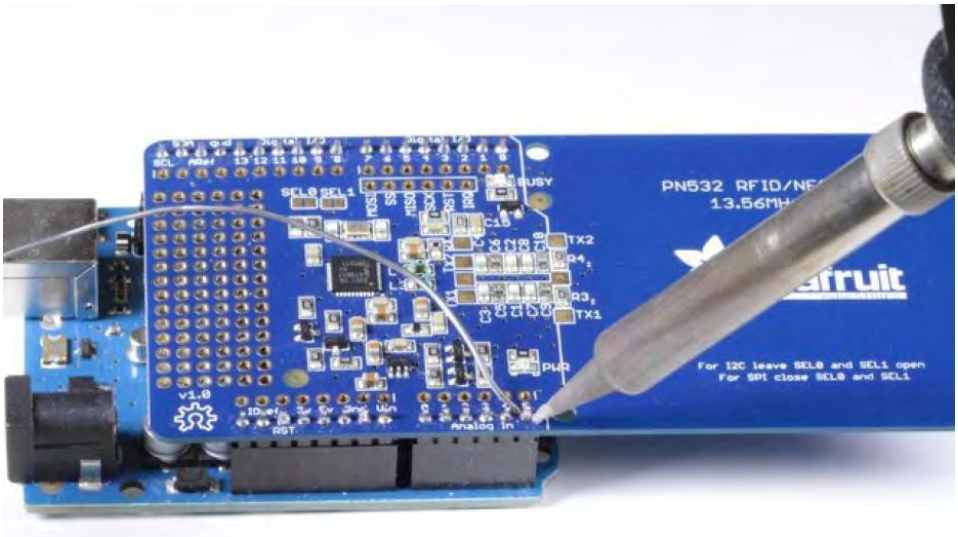


Fig. 4.4 [10]

4.3.2 Ajustes del Data-logger

Para el proceso de soldado de conectores para el Data-logger se requiere una tira de conectores de aguja para el montado superior e inferior (como se muestra en la Fig. 4.19), un cautín y soldadura.

1. Cortamos tiras de 6, 8, 8 y 10 conectores de aguja.

Fig. 4.5 [9]

2. Montamos los conectores de aguja con parte hembra hacia arriba de la placa.

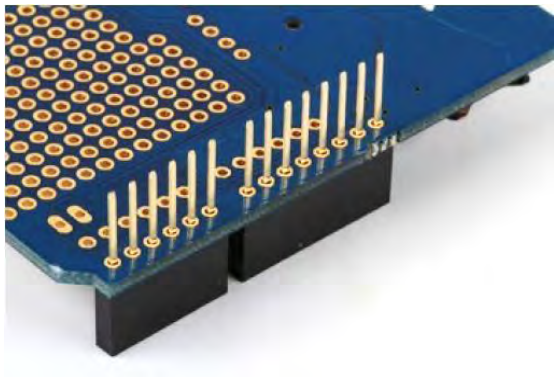


Fig. 4.6 [9]

3. Soldamos con mucho cuidado los conectores de aguja a la placa evitando que con la soldadura se unan las terminalnes.

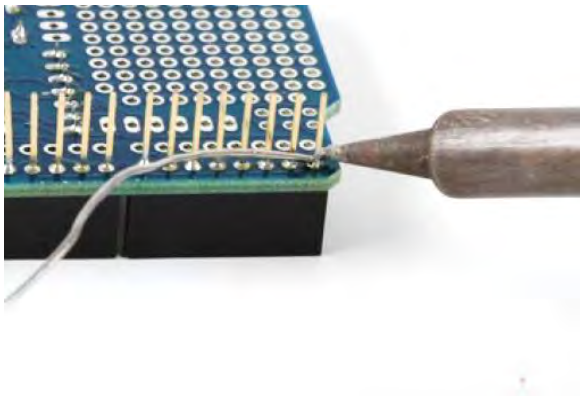


Fig. 4.7 [9]

4.4 Montado de placas para el Lector RFID/NFC

Para un mejor uso y debido al tipo de conectores de aguja instalados en la placa se propone el siguiente orden de ensamblado.

Recordemos que debido a la incopatibilidad entre el Data-logger y el Wi-Fi shield en las terminales para la SD, cortaremos la terminal 4 y 10 de nuestro Data-logger.

1. Arduino Mega

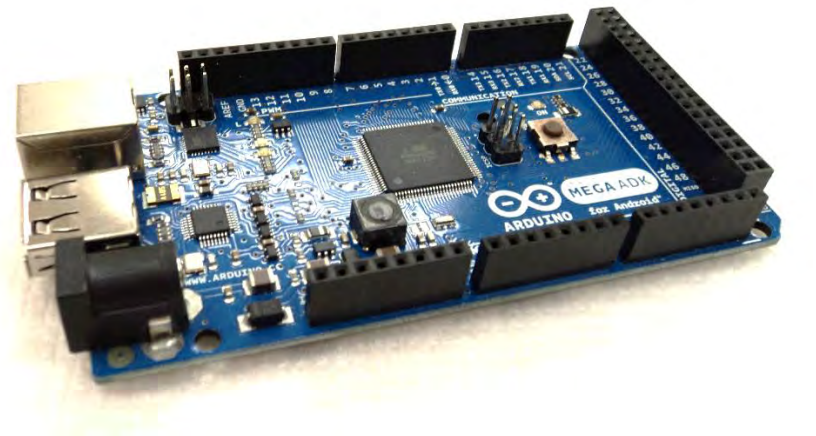


Fig. 4.8 Arduino Mega ADK

2. Arduino Wi-Fi Shield

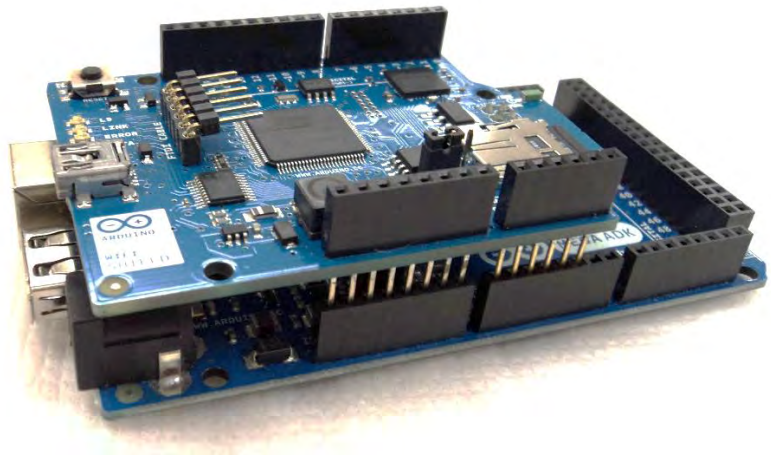


Fig. 4.9 Arduino Mega ADK + Arduino Wi-Fi Shield

3. Adafruit Data- Logger

Fig. 4.10 Arduino Mega ADK + Arduino Wi-Fi Shield + Data Logger

4. Adafruit PN532 RFID/NFC

Fig. 4.11 Arduino Mega ADK + Arduino Wi-Fi Shield + Data Logger + PN532 RFID/NFC

4.5 Instalación del Software de Programación

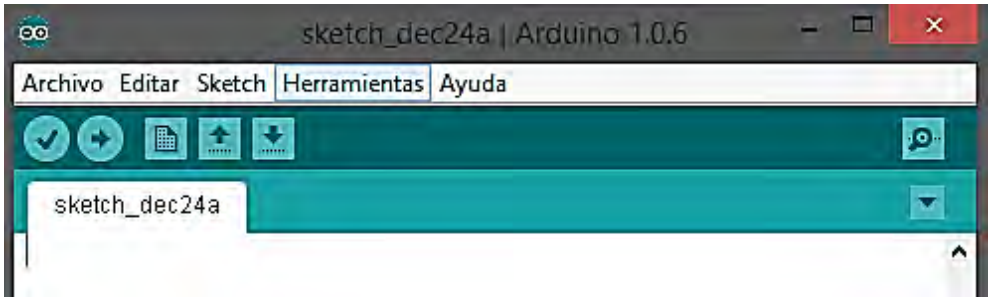
Para este prototipo se utilizó la versión 1.0.6 del compilador de Arduino. Los pasos para el sistema operativo Windows se detallan en el Apéndice C.

4.6 Instalación de Bibliotecas

Para el funcionamiento de los dispositivos de Adafruit es necesario descargar sus bibliotecas ya que sin estas, el compilador no reconocerá los comandos correspondientes a los dispositivos de Adafruit. Los pasos se detallan en el Apéndice D.

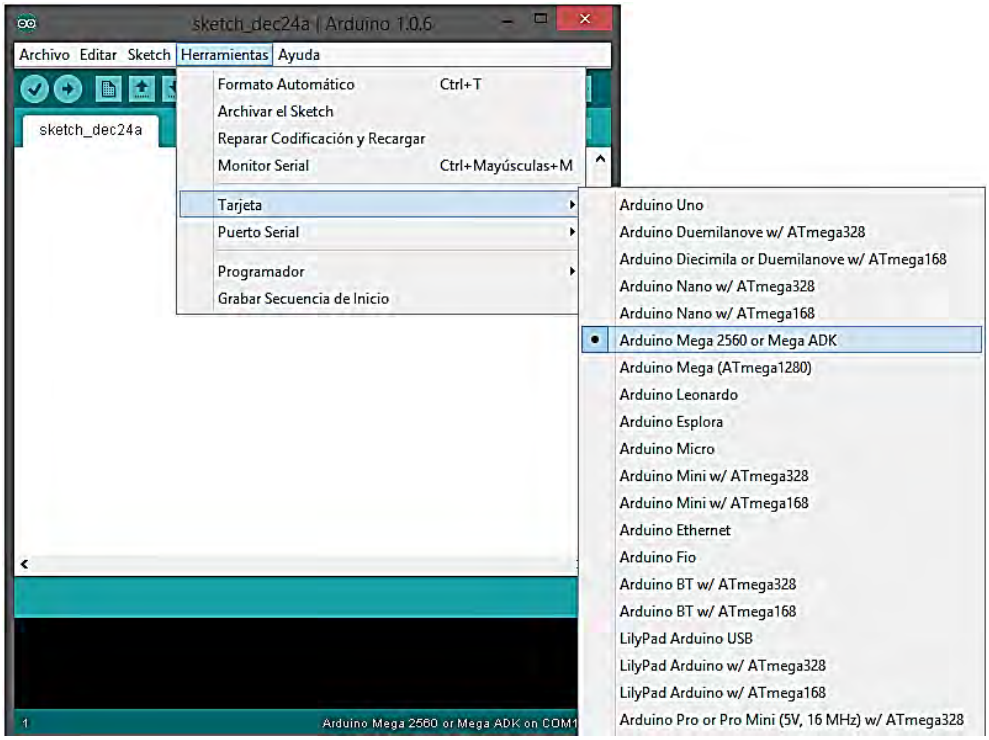
4.7 Ambiente de programación

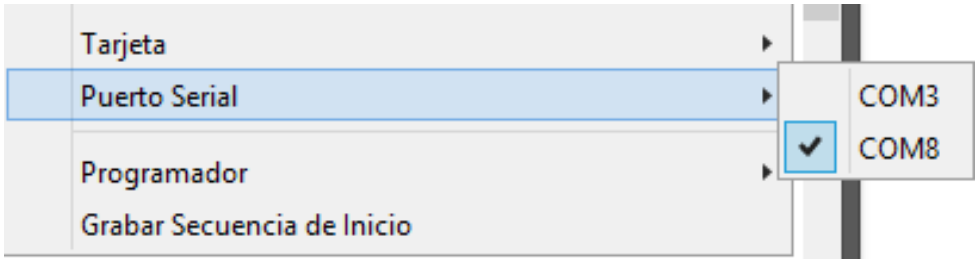
Abrimos nuestra aplicación de Arduino y en la parte superior seleccionamos la pestaña *Herramientas*.



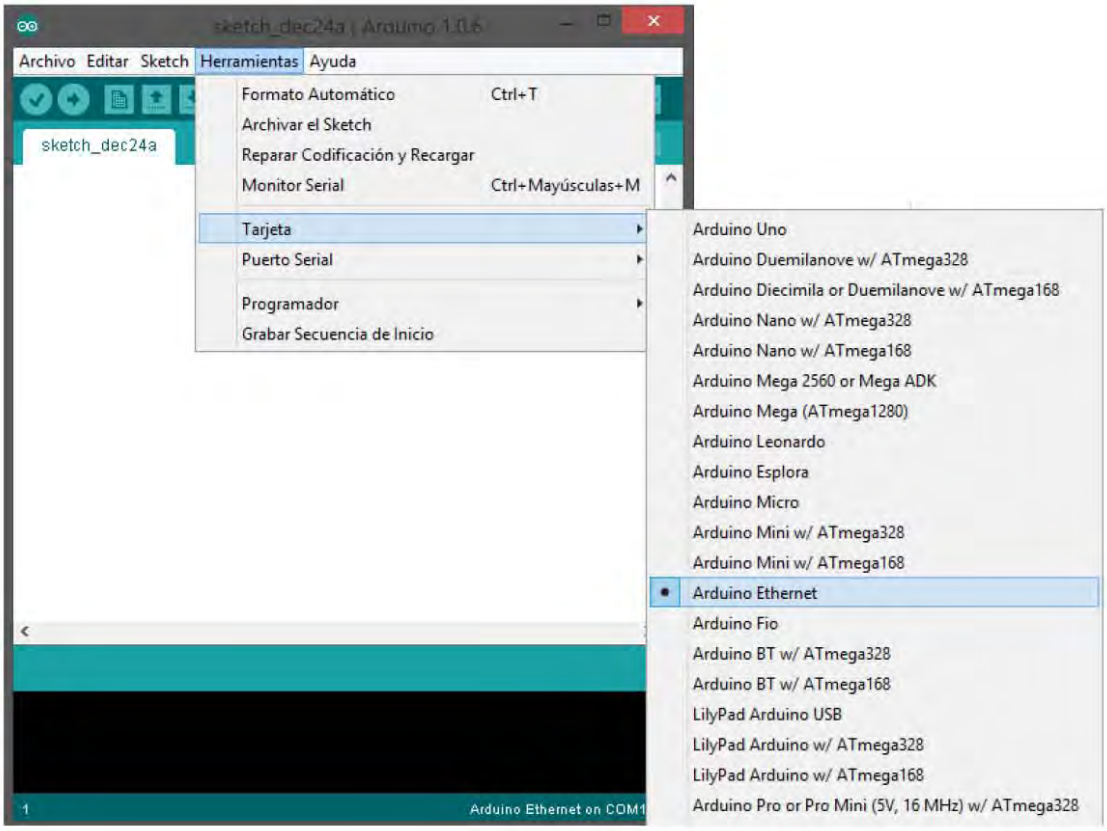
Debemos seleccionar nuestro puerto COM y el tipo de tarjeta que utilizaremos.

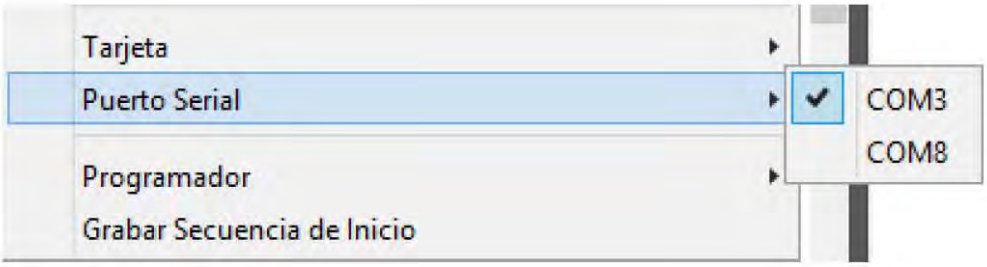
Para la parte de RFID/NFC utilizaremos la Tarjeta *Arduino Mega 2560 or Mega ADK* y el COM8.





Para la placa Ethernet utilizaremos la Tarjeta *Arduino Ethernet* y el COM 3.





4.8 Conexión y configuración del Arduino Ethernet

Existen dos tipos de dispositivos Ethernet; la placa y el Shield

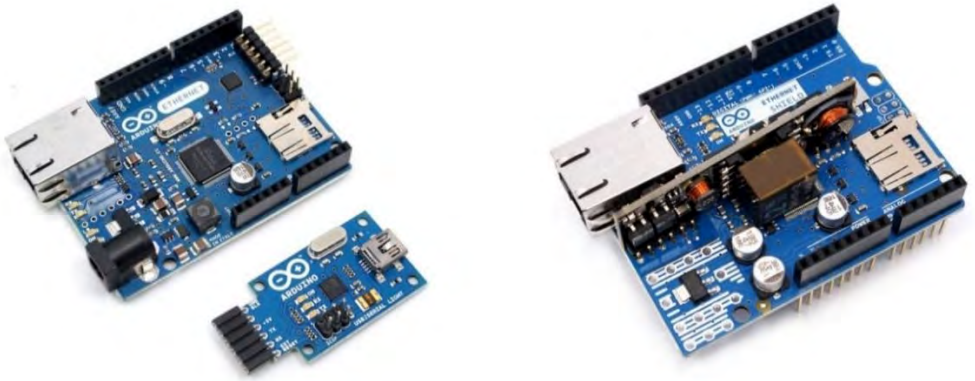


Fig. 4.12 Diferencia entre Arduino Ethernet Board y Arduino Ethernet Shield

Conexión a la PC/Laptop

1. La placa necesita un conector USB2SERIAL para alimentar la placa y conectarse con la computadora.

Fig. 4.13 USB Serial

2. Para conectarlos solo es necesario unirlos de la siguiente manera.

Fig. 4.14 Conexión del Arduino Ethernet Board y el USB Serial

3. Conectar el cable USB – mini USB a la PC/Laptop.
4. Para el Shield se necesita conectar a un Arduino (UNO; Duemilanove, Mega, etc) por la parte superior

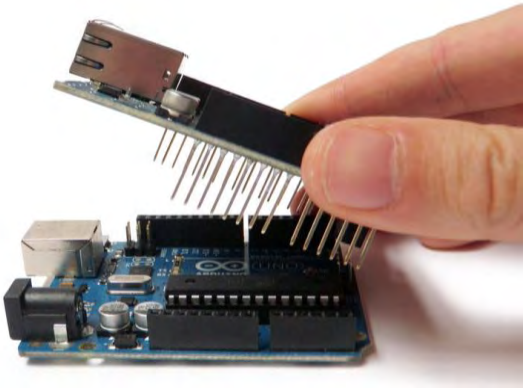


Fig. 4.15

5. Es indispensable revisar que todos los conectores concuerden en ambas placas.

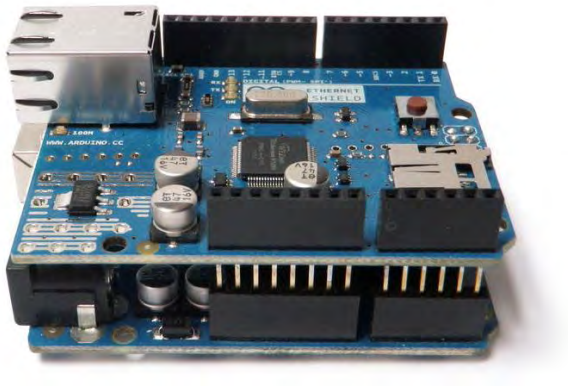


Fig. 4.16

6. Conectar el cable USB de la placa Arduino a la PC/Laptop

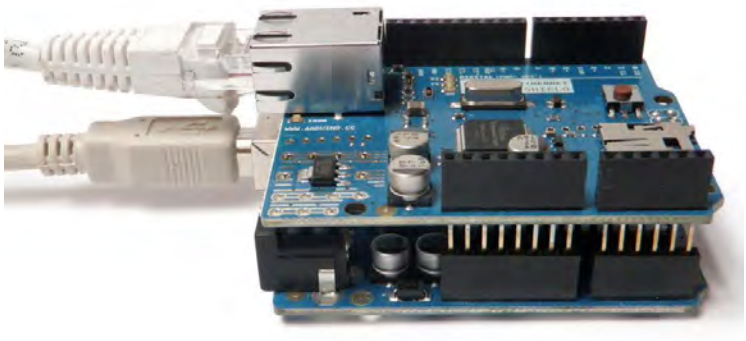


Fig. 4.17

7. Conectar la Placa/Shield al Acces Point mediante un Cable directo 568B y verificar que hay acceso a Internet.

4.9 Configuración Inicial de los parámetros de red

Para que una placa Arduino Ethernet pueda utilizar una red TCP/IP es necesario:

1. Asignar una serie de valores de configuración (dirección MAC y/o dirección IP)
2. Conocer la IP del servidor al cual nos vamos a conectar, este puede ser una computadora o una página de internet. Para mayor información del proceso pasar al Apéndice E.

4.10 Asignación y verificación de IP

Es necesario confirmar que nuestro Arduino Ethernet tiene asignada la IP correcta, podemos confirmar esta información mediante un código y una prueba de ping que encuentra detallado en el Apéndice F.

4.11 Memoria SD

La tarjeta SD es un dispositivo de almacenamiento: Es una memoria tipo Flash, la cual nos permite almacenar una gran cantidad de datos en comparación a otro tipo de memorias tales como las EEPROM. Para esto el ambiente de programación de Arduino tiene una biblioteca para el manejo de lectura y escritura de tarjetas SD:

Arduino tiene poca memoria para almacenar datos después del encendido y apagado, y para algunas aplicaciones se puede necesitar más espacio como al conectar un conjunto de sensores recolectado datos continuamente y puede que nos interese guardarlos para un posterior análisis.

Antes de iniciar a trabajar con Arduino es necesario que nuestra tarjeta SD se encuentre en formato FAT16 o FAT32. Si este no es el caso tenemos que formatear la tarjeta, no podemos abrirla desde el Arduino, tenemos que usar un adaptador para leerla, ahora vamos a “Mi pc” y localizamos el dispositivo, damos clic izquierdo y seleccionamos “formatear”.

Seleccionamos FAT o FAT 32 y damos iniciar.

4.12 Código de programación del Lector RFID

Declaración de bibliotecas

```
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <Adafruit_NFCShield_I2C.h>
#include <WiFi.h>
#include "RTClib.h"
#include <string.h>
#include "utility/debug.h"
#include <avr/pgmspace.h>
```

Parámetros del PN532 RFID/NFC

```
define IRQ 2
#define RESET 3
Adafruit_NFCShield_I2C nfc(IRQ, RESET);
uint8_t success;
uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };
uint8_t uidLength;
uint8_t nfctag[]={0, 0, 0, 0, 0, 0, 0};
```

Parámetros del lector de memoria SD

```
File myFile,myStatusFile;
const int chipSelect = 4;
```

Parámetros de conexión Wi-Fi para seguridad WEP

```
char ssid[] = "NOMBRE DE LA RED";
char key[] = "CONTRASEÑA";
int keyIndex = 0;
int status = WL_IDLE_STATUS;
IPAddress server(192,168,1,177); //Direccion del servidor (otro Arduino-escudo WiFi)
WiFiClient client;
IPAddress ip;
String cmd;
```

Parámetros de conexión Wi-Fi para seguridad WPA

```
char ssid[] = "NOMBRE DE LA RED";
char pass[] = " CONTRASEÑA”
int status = WL_IDLE_STATUS;;
IPAddress server(192,168,1,177); //Direccion del servidor (otro Arduino-escudo WiFi
WiFiClient client;
IPAddress ip;
String cmd;
```

Parámetros del RTC

```
RTC_DS1307 RTC;
int led = 13;
```

Reestablecimiento del contador a 0

```
uint8_t voltaje=0;
uint8_t cont;
boolean inicio_viaje=true;
int archivo_cerrado=0;
```

Asignación de mensaje a apuntadores en la memoria EEPROM

```
const char string_22[] PROGMEM =  "INICIO DE VIAJE";
const char string_23[] PROGMEM =  "OP:";
const char string_24[] PROGMEM =  "FECHA";
const char string_25[] PROGMEM =  "HORA";
const char string_26[] PROGMEM =  "USUARIO";
const char string_27[] PROGMEM =  "IGNICION";
const char string_28[] PROGMEM =  "FIN DE VIAJE";
const char string_29[] PROGMEM =  ": ";
const char string_30[] PROGMEM =  "NINGUN VIAJE ABIERTO";
const char string_31[] PROGMEM =  "ERROR AL ABRIR ARCHIVO";
```

Asignación de variables a apuntadores en la memoria EEPROM

```
PROGMEM const char *mensaje_22[] ={string_22};
PROGMEM const char *mensaje_23[] ={string_23};
PROGMEM const char *mensaje_24[] ={string_24};
PROGMEM const char *mensaje_25[] ={string_25};
PROGMEM const char *mensaje_26[] ={string_26};
PROGMEM const char *mensaje_27[] ={string_27};
PROGMEM const char *mensaje_28[] ={string_28};
PROGMEM const char *mensaje_29[] ={string_29};
PROGMEM const char *mensaje_30[] ={string_30};
PROGMEM const char *mensaje_31[] ={string_31};
```

Aasignación del tamaño de buffer

```
char buffer_mensaje_17[46];
char buffer_mensaje_22[15];
char buffer_mensaje_23[10];
char buffer_mensaje_24[6];
char buffer_mensaje_25[4];
char buffer_mensaje_26[7];
char buffer_mensaje_27[8];
char buffer_mensaje_28[12];
char buffer_mensaje_29[2];
char buffer_mensaje_30[20];
char buffer_mensaje_31[22];

char* ignicion[9];
char* num_tag_1;
char frasenula[]=" ";
char filename[] = "VIAJE00.CSV";
char delimitador[]=", ";
char slash[]="/";
char dos_puntos[]=":";
```

Configuración del RTC

```
Wire.begin();
RTC.begin();
RTC.adjust(DateTime(__DATE__, __TIME__))
DateTime now = RTC.now();
```

Iniciando Serial y chip PN532

```
void setup(void) {
   Serial.begin(38400);
   nfc.begin();
   nfc.SAMConfig();
   pinMode(10, OUTPUT);
```

Autentificación de memoria SD

```
Serial.print("Iniciando tarjeta SD...");
pinMode(4, OUTPUT);

if (!SD.begin(chipSelect)) {
     Serial.println("Problemas en la tarjeta SD!");
     Serial.println("Verificar que la tarjeta este incertada y/o con el formato
adecuado");
     return; }
Serial.println("OK");
```

Borrado de archivo pasado

```
Serial.println("Borrando VIAJE00.CSV...");
SD.remove("VIAJE00.CSV");
```

Lectura de tarjeta con etiqueta de Fecha y Hora

```
void loop(void) {
      voltaje=analogRead(0);
      DateTime now = RTC.now();
      success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength);
      delay(1000);
   if (success) {
            nfctag[0]=uid[0];
            nfctag[1]=uid[1];
            nfctag[2]=uid[2];
            nfctag[3]=uid[3];
String num_tag= String(nfctag[0])+String(nfctag[1])+String(nfctag[2])+String(nfctag[3]);
//Obtiene el No. de Serie del TAG
```

Comparación de lectura de voltaje de ignición

```
if (voltaje>150)      {
            ignicion[9]="ENCENDIDO";
            Serial.println(ignicion[9]); }
else       {
           ignicion[9]="APAGADO";
           Serial.println(ignicion[9]);  }
```

Validación de Tarjeta Maestra y creación de archivo

```
if ((num_tag=="237196149247")&&(inicio_viaje))
// Validacion del TAG asignado al Operador del Camion para el Inicio del Viaje
{
strcpy_P(buffer_mensaje_22, (char*)pgm_read_word(&(mensaje_22))); //INICIO DE VIAJE
Serial.println(buffer_mensaje_22);
inicio_viaje=false;

        if (! SD.exists(filename))  {
        myFile = SD.open("VIAJE00.CSV",FILE_WRITE);    }
```

Esccritura exitosa de Datos en el archivo

```
                    if (myFile)
                            {
                             myFile.println("Control de aforo");
                             myFile.println();
                                    strcpy_P(buffer_mensaje_22, (char*)pgm_read_word(&(mensaje_22)));
//INICIO DE VIAJE
                                    myFile.println(buffer_mensaje_22);
                                    strcpy_P(buffer_mensaje_23, (char*)pgm_read_word(&(mensaje_23)));
//OPERADOR
                                    myFile.print(buffer_mensaje_23);
                                    myFile.print(delimitador);
                                    myFile.println(num_tag);
                                    strcpy_P(buffer_mensaje_24, (char*)pgm_read_word(&(mensaje_24)));
//FECHA
                                    myFile.print(buffer_mensaje_24);
                                    strcpy_P(buffer_mensaje_29, (char*)pgm_read_word(&(mensaje_29)));
// :
                             myFile.print(buffer_mensaje_29);
                             myFile.print(delimitador);
                             myFile.print(now.day(), DEC);
                             myFile.print(slash);
                             myFile.print(now.month(), DEC);
                             myFile.print(slash);
                             myFile.println(now.year(), DEC);
                             strcpy_P(buffer_mensaje_25, (char*)pgm_read_word(&(mensaje_25)));
//HORA
                             myFile.print(buffer_mensaje_25);
                             strcpy_P(buffer_mensaje_29, (char*)pgm_read_word(&(mensaje_29)));
// :
                             myFile.print(buffer_mensaje_29);
                             myFile.print(delimitador);
                             myFile.print(now.hour(), DEC);
                             myFile.print(dos_puntos);
                             myFile.print(now.minute(), DEC);
                             myFile.print(dos_puntos);
                             myFile.println(now.second(), DEC);
                             myFile.println(frasenula);

                             myFile.close();
                             archivo_cerrado=0;   }
```

Condicional del fallo de escritura de datos

```
else   {
  strcpy_P(buffer_mensaje_31, (char*)pgm_read_word(&(mensaje_31)));
  //ERROR AL ABRIR ARCHIVO
  Serial.println(buffer_mensaje_31);    }    }
```

Condicional de lectura de Tarjeta Maestra

```
else   {
  if ((num_tag=="237196149247")&&(!inicio_viaje))
//Validacion del TAG asignado al Operador del Camion para el Fin del Viaje
```

Esccritura de datos de tarjeta de pasajeros en el archivo

```
{
                                myFile = SD.open("VIAJE00.CSV",FILE_WRITE);
                                strcpy_P(buffer_mensaje_28, (char*)pgm_read_word(&(mensaje_28)));
//FIN DE VIAJE
                                Serial.println(buffer_mensaje_28);
                                myFile.println(frasenula);
                                myFile.println(buffer_mensaje_28);
                                strcpy_P(buffer_mensaje_23, (char*)pgm_read_word(&(mensaje_23)));
//OPERADOR
                                myFile.print(buffer_mensaje_23);
                                myFile.print(delimitador);
                                myFile.println(num_tag);
                                strcpy_P(buffer_mensaje_24, (char*)pgm_read_word(&(mensaje_24)));
//FECHA
                                myFile.print(buffer_mensaje_24);
                                strcpy_P(buffer_mensaje_29, (char*)pgm_read_word(&(mensaje_29)));
// :
                                myFile.print(buffer_mensaje_29);
                                myFile.print(delimitador);
                                myFile.print(now.day(), DEC);
                                myFile.print(slash);
                                myFile.print(now.month(), DEC);
                                myFile.print(slash);
                                myFile.println(now.year(), DEC);
                                strcpy_P(buffer_mensaje_25, (char*)pgm_read_word(&(mensaje_25)));
//HORA
                                myFile.print(buffer_mensaje_25);
                                strcpy_P(buffer_mensaje_29, (char*)pgm_read_word(&(mensaje_29)));
 // :
                                myFile.print(buffer_mensaje_29);
                                myFile.print(delimitador);
                                myFile.print(now.hour(), DEC);
                                myFile.print(dos_puntos);
                                myFile.print(now.minute(), DEC);
                                myFile.print(dos_puntos);
                                myFile.println(now.second(), DEC);
                                myFile.close();   // cierre del archivo
                                archivo_cerrado=1;

                                  inicio_viaje=true; }
```

Cierre de lectura y archivo con llave maestra

```
                    else   {
                                if ((num_tag!="237196149247")&&(!inicio_viaje)) {
                                 myFile = SD.open("VIAJE00.CSV",FILE_WRITE);
                                 myFile.println(num_tag);
                                 myFile.print(now.hour(), DEC);
                                 myFile.print(dos_puntos);
                                 myFile.print(now.minute(), DEC);
                                 myFile.print(dos_puntos);
                                 myFile.println(now.second(), DEC);
                                 myFile.println(ignicion[9]);
                                 myFile.close();   // cierre del archivo
                                 archivo_cerrado=0;        }
                      else   {

                                 strcpy_P(buffer_mensaje_30, (char*)pgm_read_word(&(mensaje_30)));

// NINGUN VIAJE ABIERTO
                                 myFile.print(buffer_mensaje_30);
                                 Serial.println(buffer_mensaje_30);
                                      }
                                     }
                                           }

                       if (uidLength == 4)

                                { uint8_t keya[6] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
                                  success = nfc.mifareclassic_AuthenticateBlock(uid, uidLength, 4, 0, keya);

                       if (success)

                                {  uint8_t data[16];
                                   success = nfc.mifareclassic_ReadDataBlock(4, data);

                    if (success)
                            {  nfc.PrintHexChar(data, 16);
                                  }
                                        }
                                              }
                                                    }
```

Establecimiento de conexión Wi-Fi y envío de datos

```
char cc;
    if(archivo_cerrado)   {
            while (status != WL_CONNECTED) {
                Serial.print("Intentando conexion con SSID: ");
                Serial.print(ssid);
                Serial.print(" ...");
                status = WiFi.begin(ssid, keyIndex, key);
                    delay(500);   }
                Serial.println("OK");
                ip = WiFi.localIP();
                Serial.print("IP local :");
                Serial.println(ip);

            do {
                Serial.println("Intentando conexion con servidor");
                    delay(250);  }

            while (!client.connect(server, 80));
                Serial.println("Conectado con servidor");
                Serial.print("Enviando comando search...");
                client.println("GET /search?q=s?=VIAJE00.CSV HTTP/1.0");
                    delay(2500);
                Serial.println("OK");
                Serial.println("Enviando archivo...");
                myFile=SD.open("VIAJE00.CSV");

            while (myFile.available()) {
                cmd="";
            do {
                cc=(char) myFile.read();
                cmd=cmd + cc;      }
            while (cc!='\n');
                Serial.println(cmd);
                client.println("GET /search?q="+cmd);
                    delay(1000);         }
                client.println("GET /search?q=EOF");
                    delay(2000);
                myFile.close();
                Serial.println("OK");
                Serial.print("Desconectando del servidor...");
                client.flush();
                client.stop();
                Serial.println("OK");
                Serial.print("Desconectando wireless...");
                WiFi.disconnect();
                Serial.println("OK");
                    delay(5000);
                Serial.println("Borrando VIAJE00.CSV...");
                SD.remove("VIAJE00.CSV");
                inicio_viaje=true;   }
```

4.13 Código de programación del Servidor Ethernet

Declaración de bibliotecas

```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
```

Parámetros del lector de memoria SD

```
File myFile;
String linea;
const int chipSelect = 4;
```

Parámetros de Red

```
char fileName[9];
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE,
0xED};
IPAddress ip(192, 168, 1, 177);
EthernetServer server(80);
String cmd;
int led = 13;
```

Inicio del monitor Serial

```
void setup() {
  Serial.begin(38400);
```

Autentificación de memoria SIM

```
while (!Serial) {    ;  }
   Serial.print("Initializing SD card...");
   pinMode(10, OUTPUT);
   if (!SD.begin(chipSelect)) {
   Serial.println("Error en la SD.!");
   Serial.println("Verificar que la tarjeta esté instalada y con el formato
adecuado");
   return;  }
Serial.println("OK");
```

Borrado de archivo pasado

```
Serial.println("Borrando VIAJE00.CSV...");
SD.remove("VIAJE00.CSV");
```

Conexión al servido e impresión de dirección IP

```
Ethernet.begin(mac, ip);
server.begin();
   Serial.print("server is at ");
   Serial.println(Ethernet.localIP()); }
```

Busqueda de clientes disponibles

```
void loop()
{
   char cc;
   int k;
   Serial.println("Listen for incoming
clients");
   EthernetClient client = server.available();
if (client)
{
   Serial.println("new client");
   delay(100);
if (client.available()) {
   cmd="";
 do
{
      cc=(char) client.read();
      cmd=cmd + cc;
  }
 while (cc!='\n');
      Serial.println(cmd);
```

Creacion de archivo y conexión al servidor

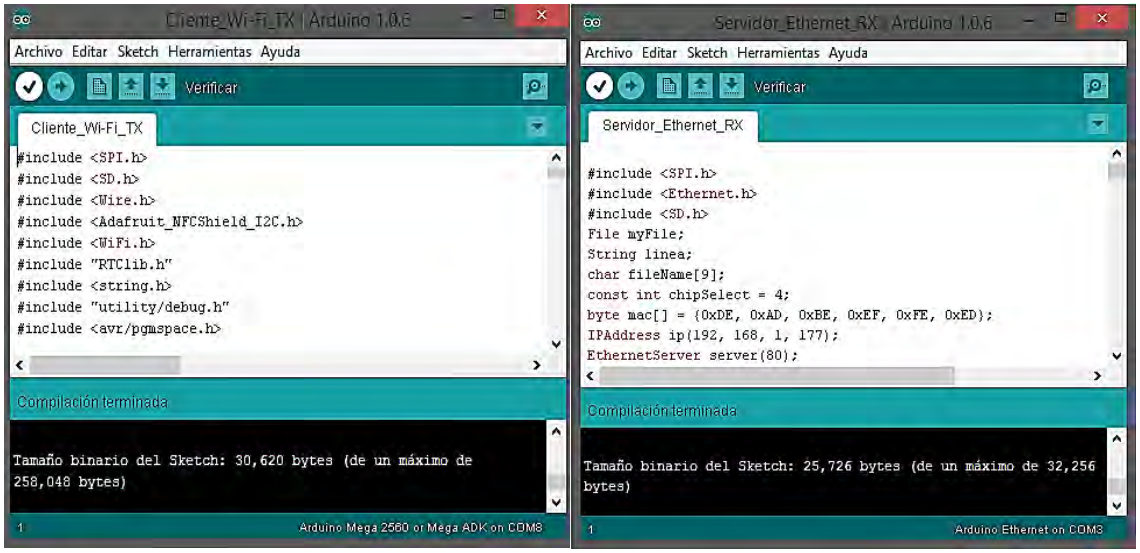
```
if (cmd.substring(5,8)=="q?=") {

            myFile=SD.open("VIAJE00.CSV");
            client.println("HTTP/1.1 200 OK");
            client.println("Content-Type: text/html");
            client.println("Connection: close");
            client.println();
            client.println("<!DOCTYPE HTML>");
            client.println("<html>");
            client.println("hello <br>");
            k=1;
            while (myFile.available()) {
                  cmd="";
                   do {
                      cc=(char) myFile.read();
                      cmd=cmd + cc;         }
                    while (cc!='\n');
                    client.print("# ");
                    client.print(k++);
                    client.print("...");
                    client.print(cmd);
                    client.println("<br />");    }
            client.println("</html>");
            Serial.println("final");
            myFile.close();    }
```

Escritura de datos en el archivo

```
else
     if (cmd.substring(14,17)=="s?=") {
          Serial.println("Recibiendo el archivo");

myFile=SD.open("VIAJE00.CSV",FILE_WRITE);
     do {
          while (!client.available());
          cmd="";
     do {
          cc=(char) client.read();
          cmd=cmd + cc;  }
     while (cc!='\n');
          Serial.print(cmd);
          delay(1000);

     if (!cmd.startsWith("EOF",14)) {
          myFile.print(cmd.substring(14)); }

     while (!cmd.startsWith("EOF",14));
           myFile.println();
           myFile.close();
           Serial.println("Recepcion finalizada");   }  }
   client.stop();
   Serial.println("client disconnected");
   }   }
```

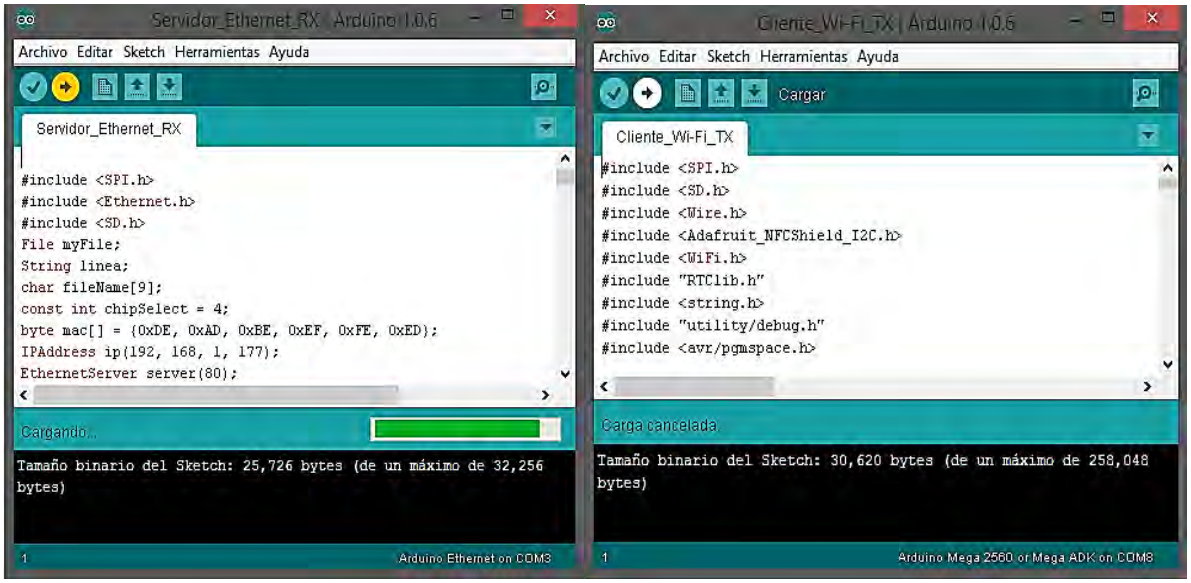
4.14 Compilación del código de programación

Despúes de escribir el código en el cuadro de texto del software, debemos compilarlo para comprobar que la sintaxís es correcta y no hay errores. Para realizar esta función hay que dar click al ícono check ✓ que se ecuentra en la parte superior izquierda. Si todo está bien en nuestro código nos mostrará el mensaje " Compilación terminada".



4.15 Carga del código de programación

Despúes de haber compilado nuestro código de programación sin ningún error, podemos proceder a cargarlo en el microcontrolador dando click en el ícono con una flecha que se ecuentra a la derecha del símbolo check ✓. Se desplegará una barra donde nos mostrará el proceso de carga, al cubrirse en su totalidad nos mostrará el mensaje "Carga finalizada" y nuestro dispositivo se encontrará listo para operar.



4.16 Pruebas y Ajustes de parámetros

Con el fin de realizar pruebas, se conectaron ambos módulos a una laptop para observar mediante el monitor serial los procesos de adquisición, envío y recepción de datos. Se ensambló con respecto a nuestro diagrama propuesto (Fig. 4.1) y se simuló el proceso del sistema.

El módulo del autobús se conectó en el puerto del lado derecho de la laptop.

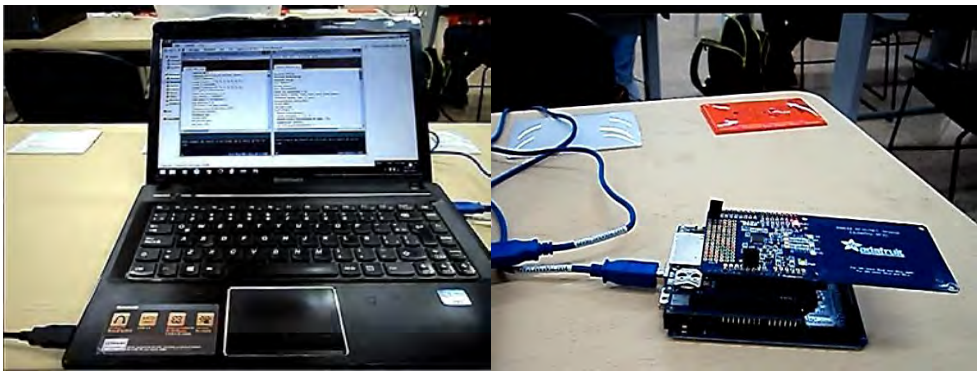


Fig. 4.18 Laptop y módulo del autobús

Para el módulo de la central se conectó en un puerto de lado izquierdo de la laptop, intentando alejar lo más posible el Access Point del módulo del autobús.

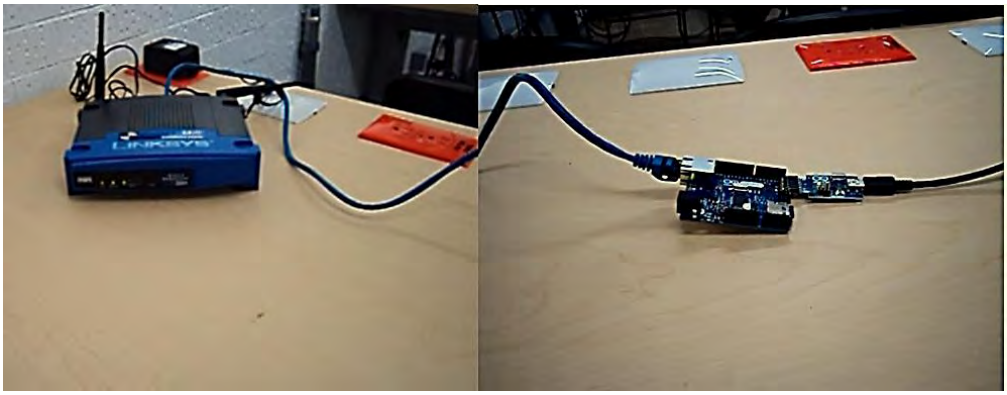


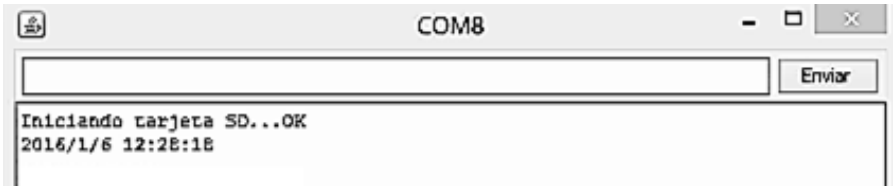
Fig. 4.19 Access Point y módulo de la Central

Para las pruebas se utilizaron 3 tarjetas,1 para el chofer y 2 para los usuarios.

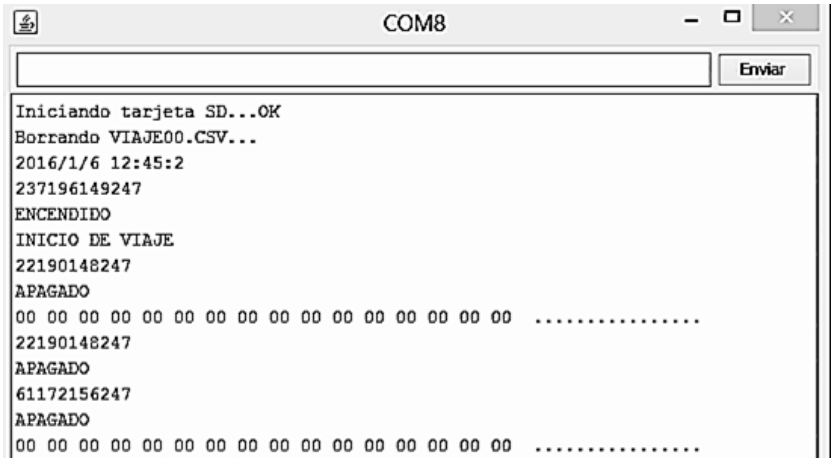
Fig. 4.20 Tarjetas Chofer/Usuarios

En la primera etapa se probó el módulo de lectura, almacenamiento y transmisión de datos (Autobús) y los resultados fueron en su mayoría favorables.

El reconocimiento de la tarjeta SD y la incorporación de la fecha y hora fueron exitosos.

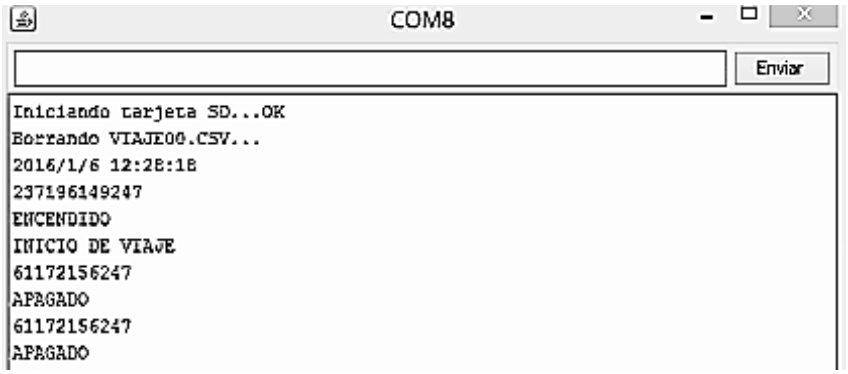


Para la recolección de datos de las tarjetas, se tuvo que hacer un ajuste en el parámetro que controla el tiempo entre cada lectura por parte del lector. Primero se estableció un valor delay=500 pero éste era muy rápido y se tenía que colocar y retirar la tarjeta rápidamente del lector para evitar errores en la lectura.



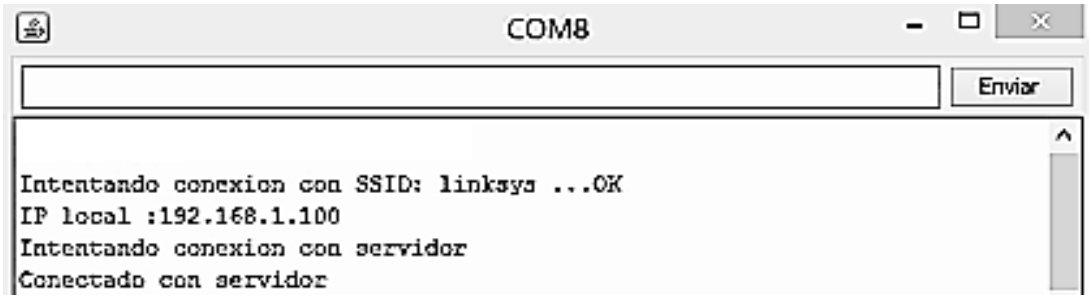
Para evitar estos errores en la lectura, se cambió el valor del delay a 1000, con lo que el tiempo entre cada muestra aumento y ya no se tenía que colocar y retirar rápidamente la tarjeta del lector.

```
void loop(void) {
      voltaje=analogRead(0);
      DateTime now = RTC.now();
      success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
&uidLength);
      delay(1000);
```

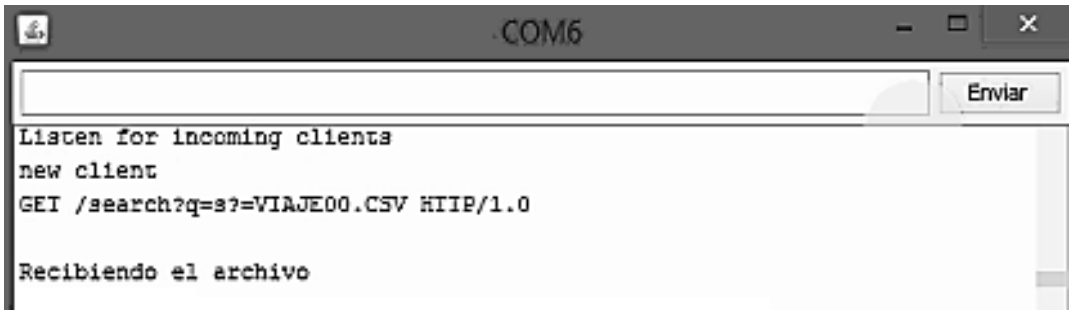


En la segunda etapa se probó la comunicación entre el módulo de lectura, almacenamiento y transmisión de datos (Autobús) y el módulo de recepción y almacenamiento de datos (Central).

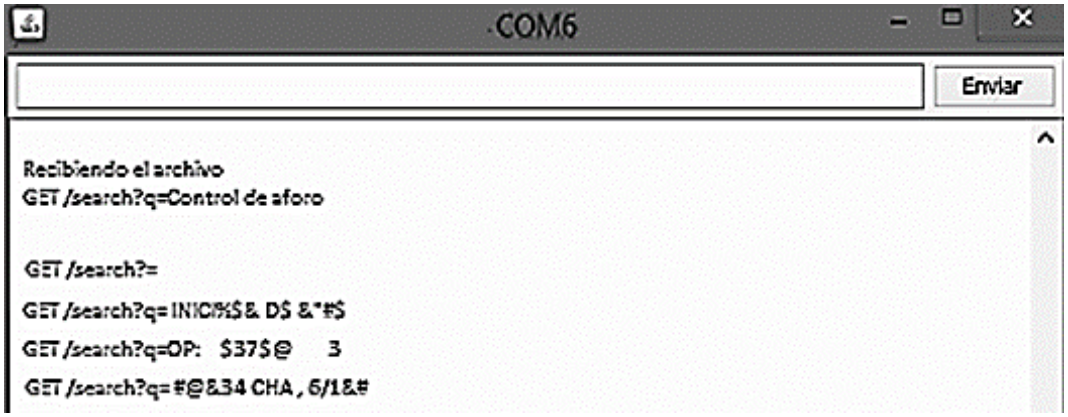
Por parte del módulo del autobús, se logró la conexión al Access Point.



Del lado del módulo de la Central, se confirmó la conexión con el módulo del autobús.



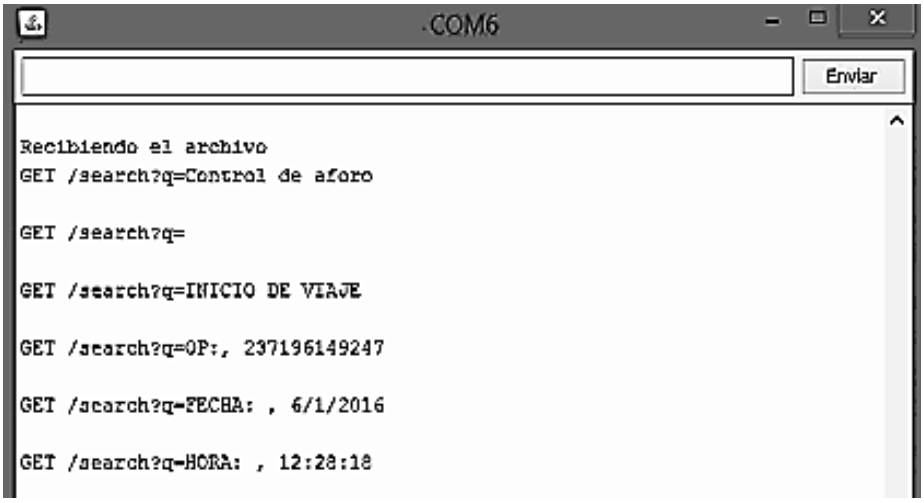
Para la tercera etapa, se comprobó que los datos eran recibidos por el módulo de la central, pero no se obtuvieron los resultados esperados ya que llegaban incompletos o con interferencia.



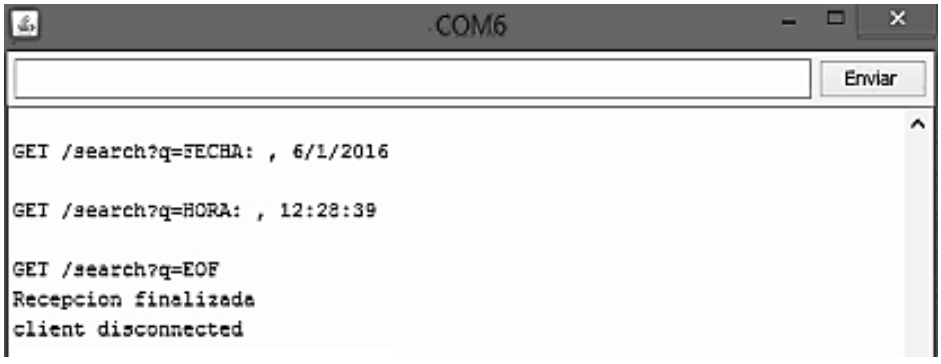
Se llegó a la conclusión, que no había una sincronía entre la transmisión y recepción, por lo tanto se realizó un ajuste en el intervalo de envío de datos por parte del módulo del autobús. Para que el módulo de la central pueda recibir y guardar los datos completos, se modificó los valores de delay a 1000 y 2000.

```
while (cc!='\n');
    Serial.println(cmd);
    client.println("GET /search?q="+cmd);
        delay(1000);        }
    client.println("GET /search?q=EOF");
        delay(2000);
```

Despúes de realizar los cambios en los intervalos de tiempo, de lado de la central se recibieron correctamente los datos de forma completa y sin interferencias.



Posteriormente se verificó que la recepción finalizó y se cerró el enlace de comunicación con el módulo del autobús.



Para finalizar se comprobó de manera exitosa, que los datos recabados se guardaron en un archivo de Excel, en donde se podrán ver los registros de cada viaje y generar tendencias y/o estadísticas con la información.

En la primera fila se muestra el título de nuestra tabla, de la fila 3 a 6 nos muestra el inicio de nuestro viaje, el ID de nuestro Operador (chofer), la fecha y la hora en que inició el abordaje. Las líneas posteriores nos muestran el ID de nuestros usuarios, la hora en que abordaron y el estado del autobús, dependiendo el número de pasajeros será el tamaño de nuestra tabla, en este ejemplo se realizaron 4 abordajes. Por último en la fila 21 se encuentran los datos que finalizan el sistema mostrando nuevamente el ID de nuestro Operador (chofer), la fecha y la hora en que finalizó el abordaje.

	A	B
1	Control de aforo	
2		
3	INICIO DE VIAJE	
4	ID OPERADOR:	237000000000
5	FECHA:	6/1/2017
6	HORA:	12:45:2
7		
8	22190148247	
9	12:45:09	
10	APAGADO	
11	22190148247	
12	12:45:11	
13	APAGADO	
14	61172156247	
15	12:45:13	
16	APAGADO	
17	61172156247	
18	12:45:16	
19	APAGADO	
20		
21	FIN DE VIAJE	
22	ID OPERADOR:	237000000000
23	FECHA:	6/1/2017
24	HORA:	12:45:18
25		
26		

VIAJE00

Una vez terminadas las pruebas, se cumple el objetivo general al lograr transmitir la información obtenida mediante RFID/NFC (ID del chofer, ID del pasajero, estado del autobús, fecha y hora del abordaje) por medio de una red Wi-Fi. Por lo tanto se concluye la hipótesis de la metodología y se comprueba que el sistema es funcional para su uso en el mercado.

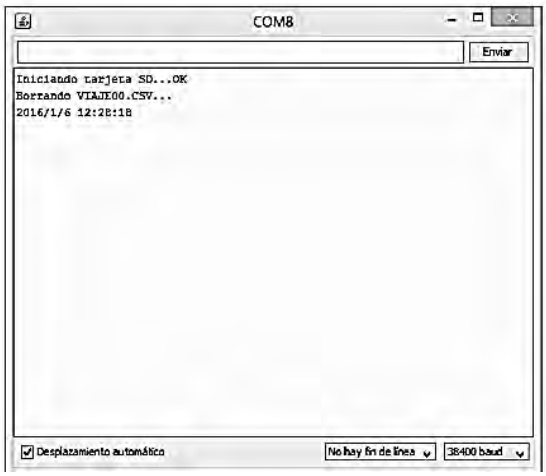
Resultados

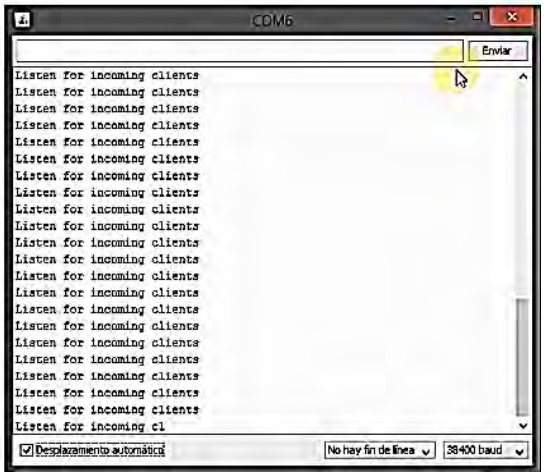
Al término de las pruebas se plantearon diversos escenarios en los que nuestro sistema está vulnerable a darnos falsos resultados (conectividad, reconocimiento de tarjeta SD, utilizar las tarjetas equivocadas, dejar las tarjetas mucho tiempo sobre el lector).

Esto nos generó 6 casos posibles los cuales se detallan a continuación.

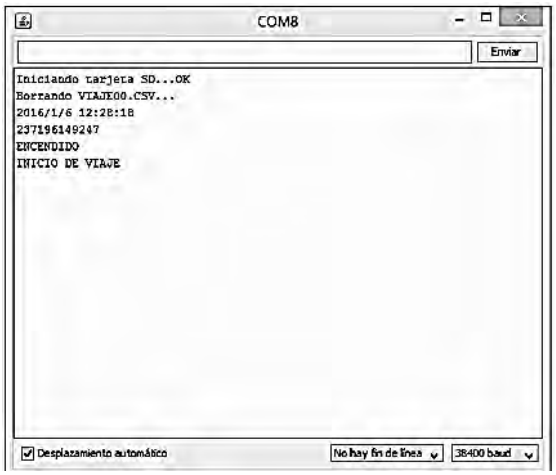
Caso exitoso

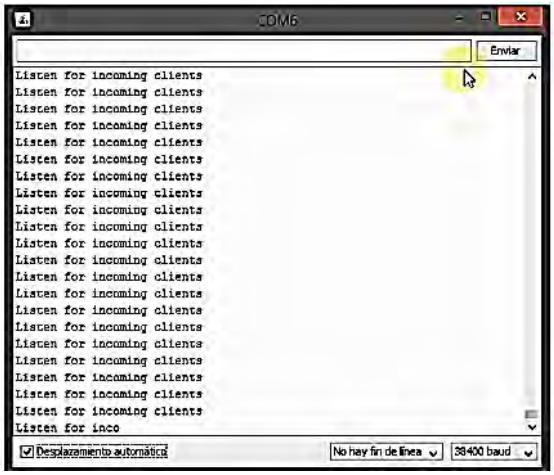
1.- Verifica el estado de la SD y borra el archivo viejo con la hora y fecha.



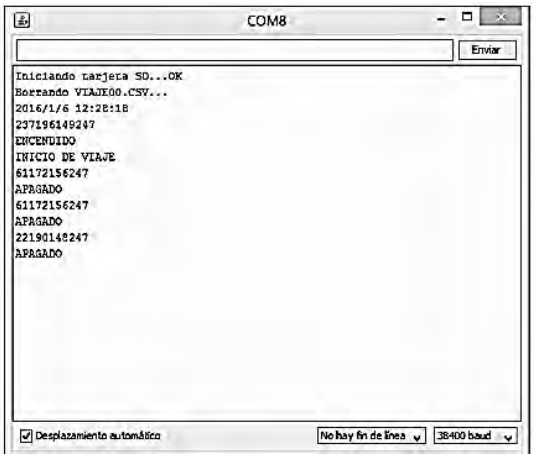


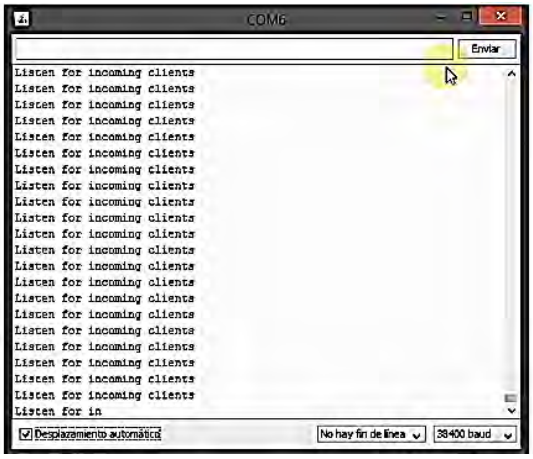
2.- Al pasar la tarjeta maestra se activa el sistema registrando la fecha y hora de inicio.



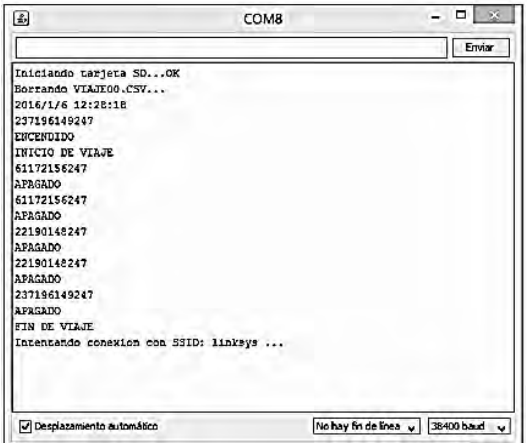


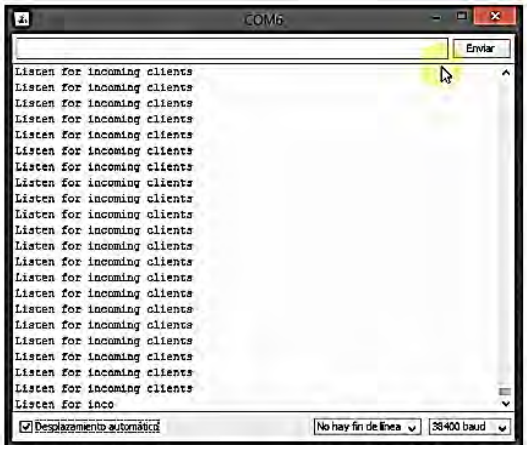
3.- Se recopila la información de las tarjetas indicando la fecha y hora de cada una.



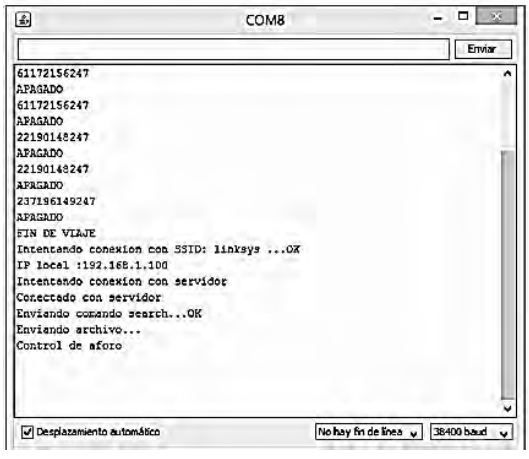


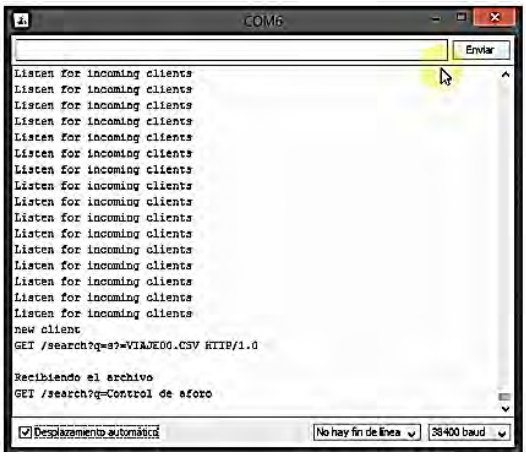
4.- Al pasar por segunda vez la llave maestra se cierra el sistema y comienza a buscar una red para conectarse.



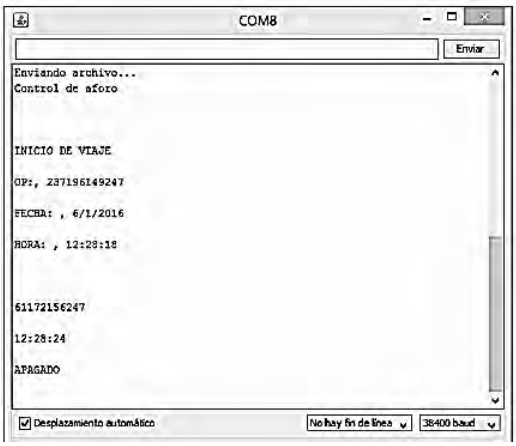


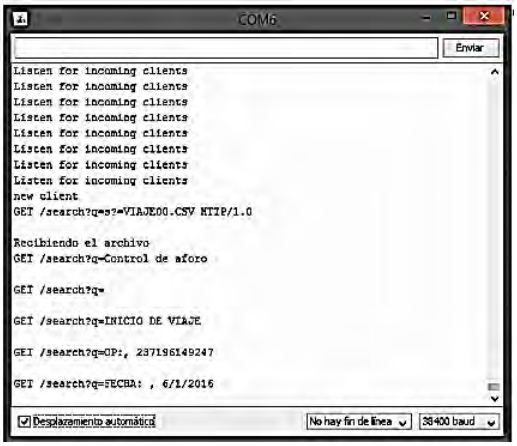
5.- Al encontrar la red Wi-Fi previamente configurada comienza la transferencia de datos.



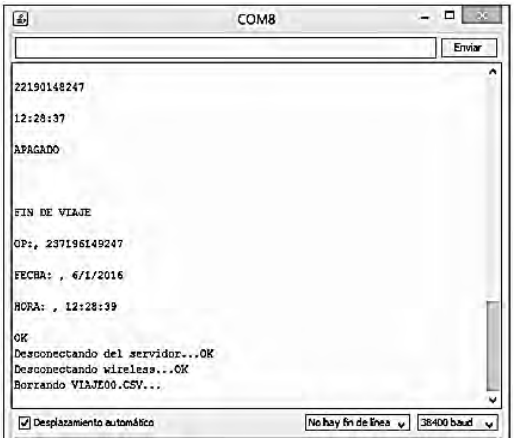


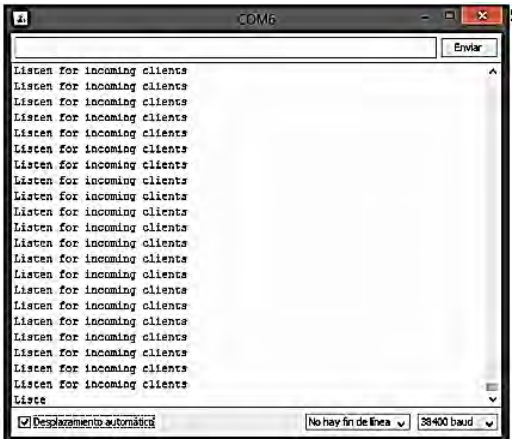
6.- Los datos son enviados línea por línea entre ambos módulos.





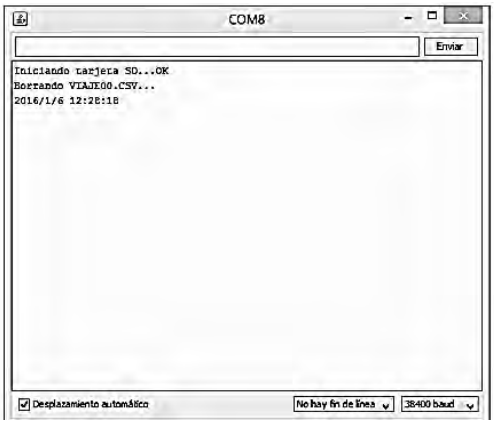
7.- Al finalizar el envío de datos se desconecta del servidor y nuevamente el sistema queda abierto para la lectura de tarjetas.

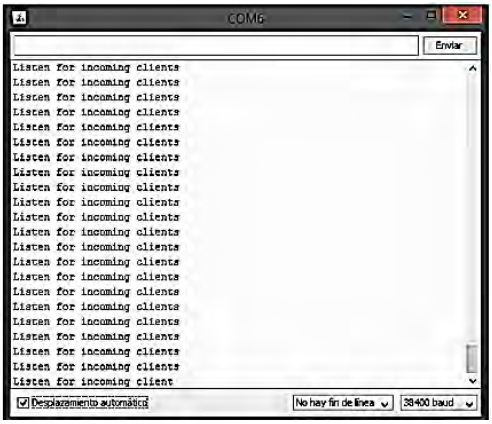




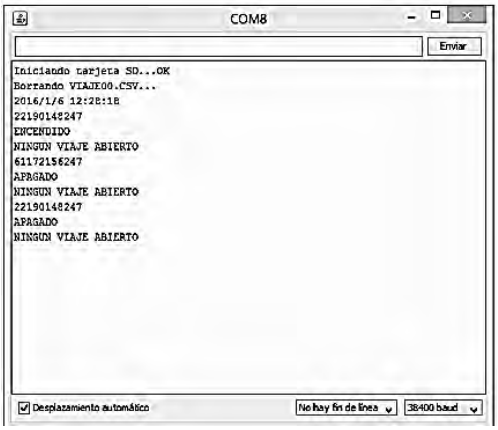
Caso otra tarjeta llave

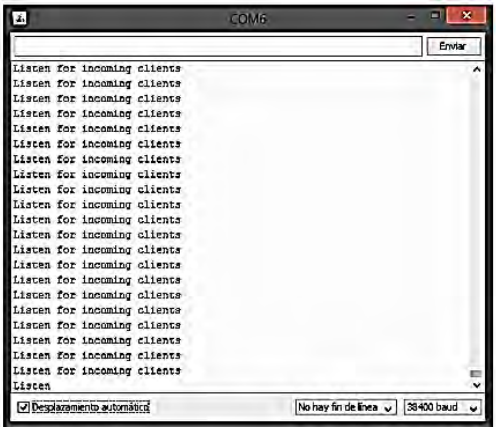
1.- Verifica el estado de la SD y borra el archivo viejo con la hora y fecha.



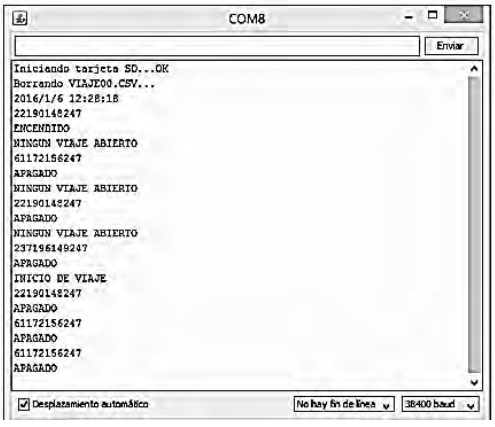


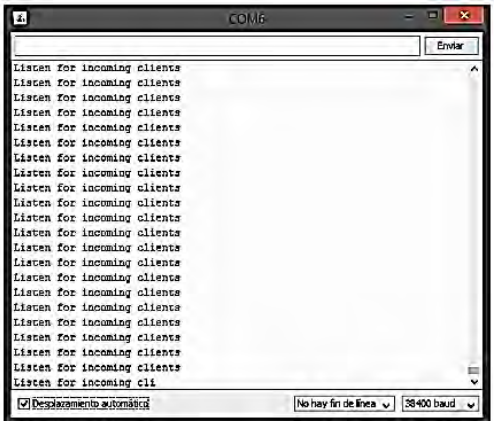
2.- Al pasar la tarjeta de pasajero se lee pero no se activa el sistema.



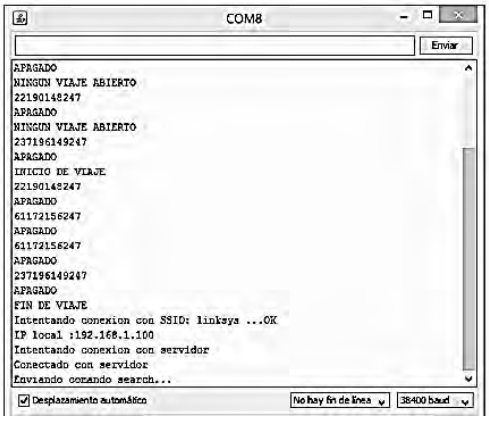


3.- Hasta que pasamos la tarjeta maestra se activa el sistema y las tarjetas de los pasajeros pueden ser guardadas en el sistema.



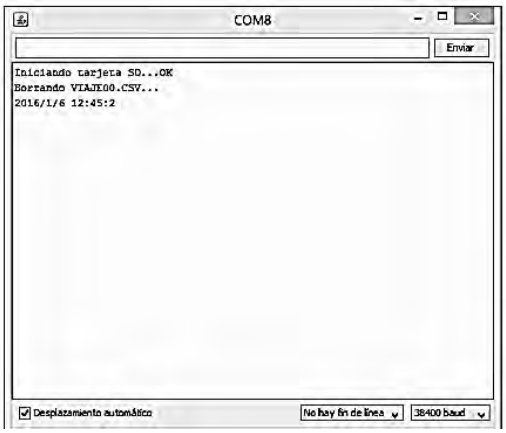


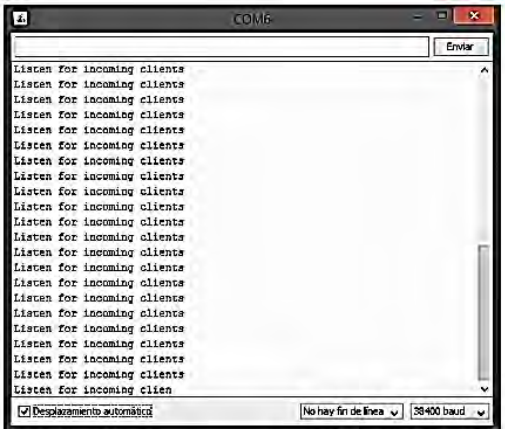
4.- Cuando pasamos la tarjeta maestra por segunda ocasión cerramos el sistema y se comienza con la búsqueda de la red Wi-Fi.



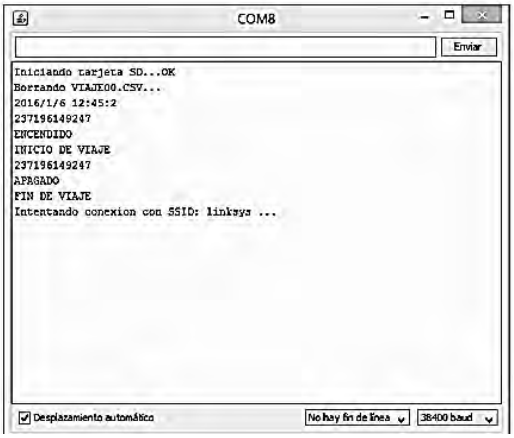
Caso no hay conexión

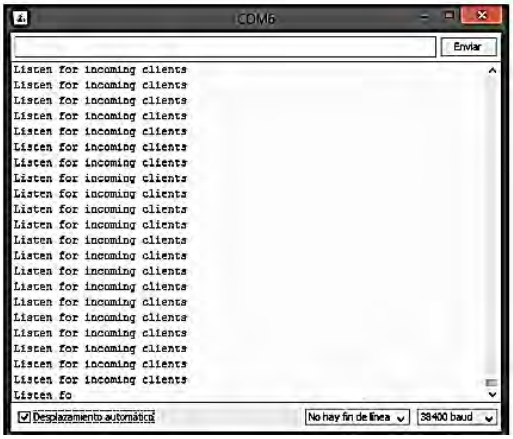
1.- Verifica el estado de la SD y borra el archivo viejo con la hora y fecha.





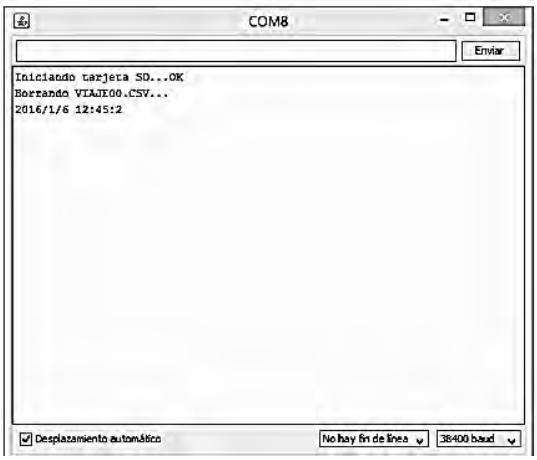
2.- Al pasar la tarjeta maestra se activa el sistema registrando la fecha y hora de inicio pero se queda intentando conectarse al servidor mostrando el mensaje en el monitor serial.

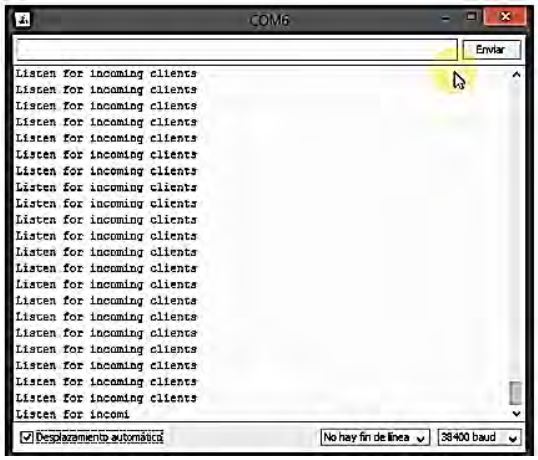




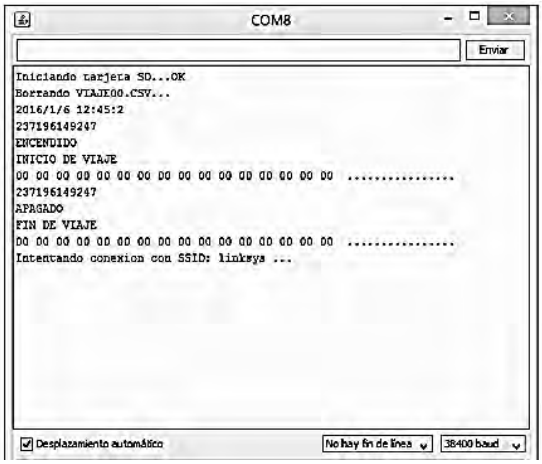
Caso llave maestra mucho tiempo

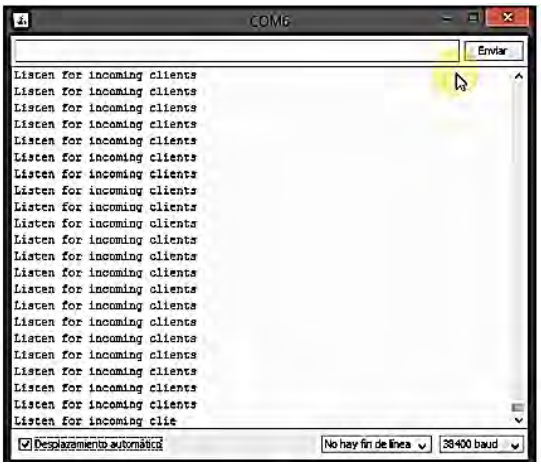
1.- Verifica el estado de la SD y borra el archivo viejo con la hora y fecha.



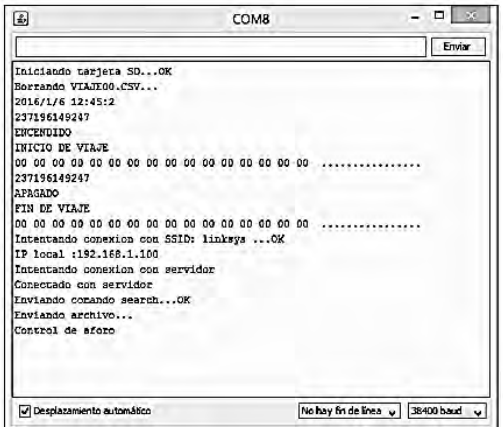


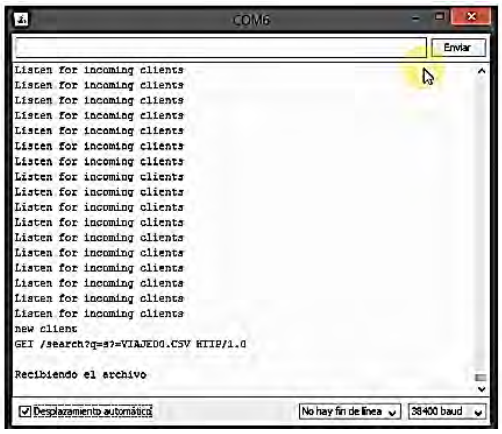
2.- Al dejar la tarjeta maestra colocada se activa el sistema y registra el ID de la tarjeta pero al estar mucho tiempo el ID mostrado es incorrecto, por otra parte da Inicio y Fin del viaje sin dar oportunidad de pasar otras tarjetas.



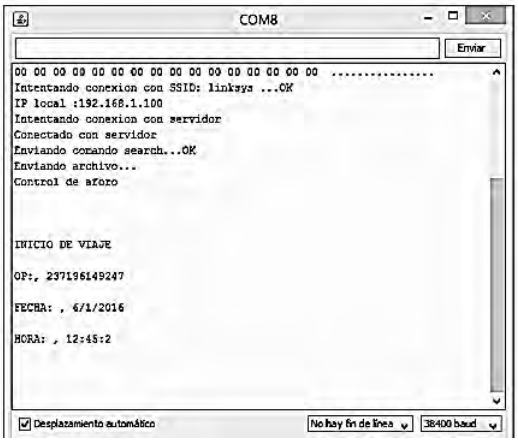


3.- Al leer por segunda vez la llave maestra se cierra el sistema y comienza a buscar una red para conectarse.



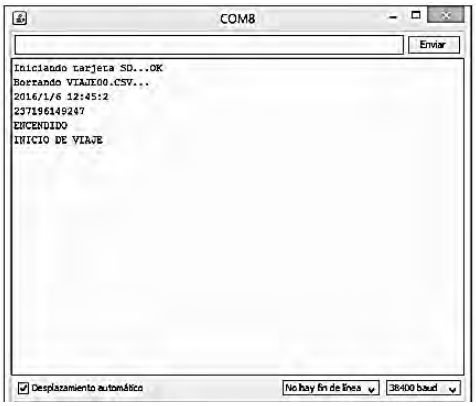


4.- Cuando encuentra la red Wi-Fi previamente configurada comienza la transferencia de datos, en este caso los datos que se transmitirán son los ID's erróneos de la tarjeta maestra y sin ningún otro ID.

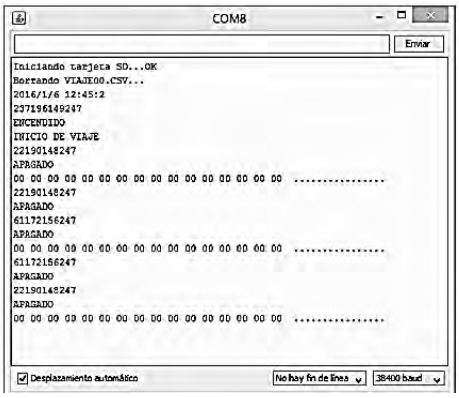


Caso llaves mucho tiempo

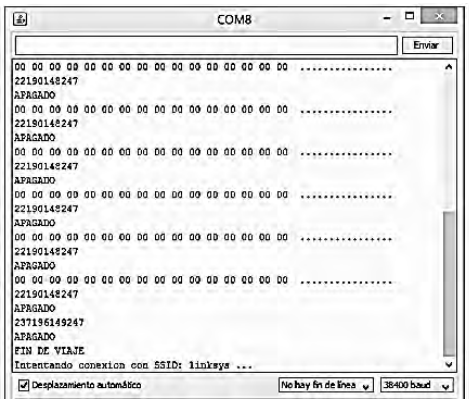
1.- Verifica el estado de la SD y borra el archivo viejo con la hora y fecha.



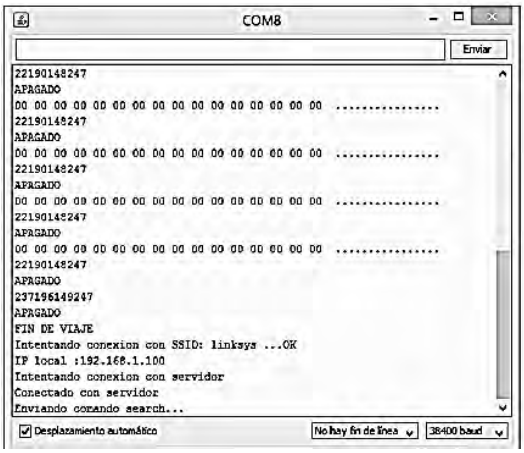
2.- Al pasar la tarjeta maestra se activa el sistema y permite el registro de las demás tarjetas, pero si dejamos una expuesta al lector por mucho tiempo registrará erróneamente los ID's junto con la fecha y hora de inicio.

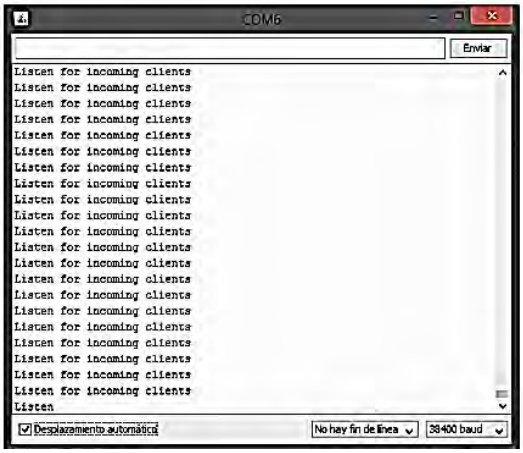


3.- Pasamos por segunda vez la llave maestra se cierra el sistema y comienza a buscar una red para conectarse.



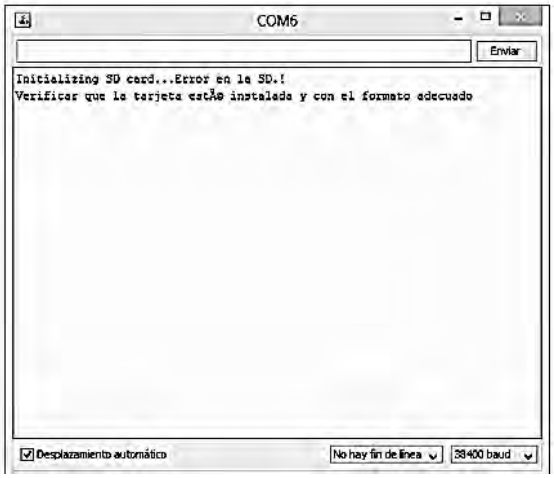
4.- Cuando encuentra la red Wi-Fi previamente configurada comienza la transferencia de datos, en este caso los datos que se transmitirán son los ID's erróneos de las tarjetas de los pasajeros.





Caso SD dañada o no insertada

1.- Al iniciar el sistema nos aparecerá un mensaje indicándonos un error con la SD.



Conclusiones

El objetivo principal de la tesis consiste en generar un prototipo capaz de adquirir datos (ID del chofer, ID del pasajero, estado del autobús, fecha y hora del abordaje) por medio de RFID/NFC, guardar y transmitir la información por Wi-Fi. Después de haber realizado pruebas se cumplieron los requerimientos solicitados y se comprobó que el sistema es funcional para su uso en el mercado.

Para poder lograr esto se comprendieron los estándares de las redes alámbricas e inalámbricas, los cuales sirvieron como modelos y referencias para poder generar la interconexión entre dispositivos de diferentes tecnologías y marcas.

Como parte de este trabajo se evaluaron diferentes tecnologías y proveedores en el mercado para hacer una nueva propuesta a la original que se había indicado por parte de la empresa teniendo que cambiar el Shield Wi-Fi de la empresa Adafruit por el Shield Wi-Fi de Arduino ya que la configuración física entraba en conflicto con la del microcontrolador Arduino y las bibliotecas no cumplían con las capacidades requeridas para la transmisión de datos como lo requería el sistema diseñado. Así mismo se logró comprender los lenguajes de programación y los protocolos de comunicación que se utilizaron para la realización del prototipo.

Al término de la investigación sobre la tecnología RFID/NFC se logró generar un conocimiento más detallado y profundo el cual se expone en este trabajo de modo que el lector pueda tener un acercamiento más amigable para comprender el trasfondo e importancia de esta tecnología así como un conocimiento técnico detallado y resumido de la misma.

Al trabajar en el desarrollo de este sistema se comprendieron a detalle y se ampliaron conocimientos en diversas áreas como lo son las redes de telecomunicaciones, la funcionalidad de cada una de las capas del modelo OSI, la importancia de los estándares para lograr una interconectividad entre dispositivos y tecnologías, las comunicaciones por medio de radiofrecuencia, la importancia de la modulación de una señal, procesamiento de la información, codificación de la información y la configuración de microcontroladores para poder realizar un proyecto de telecomunicaciones.

Personalmente creo que el poder aplicar los conocimientos obtenidos durante la realización de este trabajo para solucionar un problema de ingeniería es uno de los logros más importantes ya que al ser este trabajo la culminación de una etapa de formación me demuestra el potencial y la capacidad de poder hacer algo útil por la sociedad partiendo de un problema real el cual requiere de una solución inmediata,

Trabajo a futuro

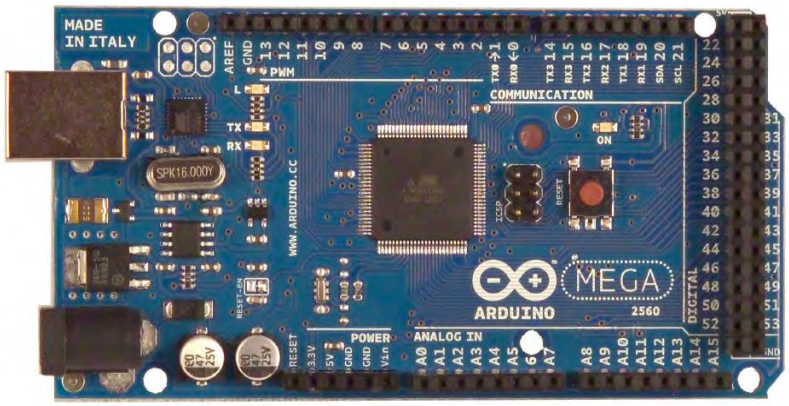
El sistema actualmente es funcional y cumple con los requisitos solicitados para el primer prototipo, como siguiente paso se necesitaría hacer la electrónica mínima del sistema realizando un análisis de cada shield y descartar los componentes adicionales propios de los shield de desarrollo. Teniendo el diseño nuevo se puede hacer una tarjeta con los componentes mínimos ahorrando costos de producción y disminuir el tamaño del dispositivo para poder introducirlo en un case y así generar un producto comercial.

Debido a las especificaciones solicitadas se utilizó la tecnología Wi-Fi y un servidor vía Ethernet como medio de comunicación, posteriormente se puede desarrollar una versión con GPRS, agregar GPS y ampliar la red de sensores al camión para tener un mayor control de lo que sucede en el camión en tiempo real.

Apéndice A

Especificaciones técnicas de los dispositivos utilizados

Arduino Mega 2560 ADK



Es un microcontrolador basado en el Atmega2560. Tiene 54 terminales digitales input/output de los cuales 14 pueden ser utilizados como salidas PWM, 16 terminales análogos input, 4 UARTs (puertos seriales para hardware), un oscilador de cristal a 16 MHz, un conector USB, un jack para alimentación, un header ICSP y un botón de reset. [G]

Resumen

Microcontrolador	Atmega2560
Voltaje de Operación	5V
Voltaje de Entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Terminales Digitales I/O	54 (14 proveen una salida PWM)
Terminales Input Analógicos	16
Corriente DC por la terminal I/O	40 Ma.
Corriente DC para la terminal a 3.3V	50 Ma.
Memoria Flash	256 Kb de los cuales 8 Kb son usados como gestor de arranque.
SRAM	8 KB
EEPROM	4 KB
Velocidad de Reloj	16 MHz

Alimentación

El Arduino Mega puede alimentarse mediante una conexión USB o con una alimentación externa con un eliminador.

Las terminales de alimentación internas son los siguientes:

- **VIN.** La terminal de entrada de voltaje para el Arduino cuando se utiliza una fuente de alimentación externa (cuando no se tiene una alimentación de 5 volts por el USB u otra fuente regulada). Uno puede suplir de voltaje a través de esta terminal o por el jack de alimentación que pasa por esta terminal.
- **5V.** Una fuente de alimentación regulada utilizada para el microcontrolador y otros componentes de la placa. Esta proviene de un regulador integrado en la placa.
- **3.3V.** Fuente de alimentación generada por un regulador integrado en la placa. La corriente máxima generada es de 50 mA.
- **GND.** Terminal de Tierra.

Memoria

El ATmega2560 tiene 256 KB de memoria Flash para almacenar el código (8KB son utilizados como gestor de arranque), 8 KB de SRAM y 4KB de EEPROM.

Input/Output

Cada uno de las 54 terminales digitales del Mega pueden ser utilizados de entrada o salida (input/output) mediante las funciones *pinMode( ), digitalWrite( )* y *digitalRead( )*.Cada terminal opera a 5 volts generando un máximo de 40mA.

- **Serial 0: (RX) y 1 (TX); Serial 1: 19 (RX) y 18 (TX); Serial 2: 17 (RX) y 16 (TX); Serial 3: 15 (RX) y 14 (TX).** Son utilizados para recibir (RX) y para transmitir (TX) datos TTL seriales. Las terminales 0 y 1 están conectados con las correspondientes del ATmega8U2 USB-to-TTL Serial Chip.
- **Interruptores Externos: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) y 21 (interrupt 2).** Estas terminales pueden ser configurados con un trigger e interrumpen un valor bajo, un alza, una caída o un cambio de valor.
- **PWM: 0 a 13.** Proveen un PWM de 8 bits de salida con la función *analogWrite( ).*
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)**. Estas terminales soportan comunicación SPI.
  LED: 13. Un LED incorporado que está conectado a la terminal digital 13. Cuando el valor de la terminal es HIGH el DED se enciende y cuando el valor es LOW se apaga.
- **I2C: 20 (SDA) y 21 (SCL).** Soporta I2C (TWI).

El Mega2560 tiene 16 entradas analógicas, cada una provee 10 bits de resolución. Por defecto miden de tierra a 5 volts siendo posible cambiar el valor límite mediante la terminal AREF y la función *analogReference ( ).*

Hay otro par de terminales en la placa:

- **AREF.** Hace una referencia al voltaje de entrada.
- **Reset.** Brinda un LOW para resetear el microcontrolador. Generalmente se agrega un botón a las placas.

Comunicación

El Arduino Mega2560 tiene muchas facilidades para comunicarse con una computadora, con otro Arduino o con otros microcontroladores. Cuenta con cuatro UARTs de hardware para comunicación serial TTL (5V). Un ATmega8U2 en los canales de la placa, uno sobre USB que provee un puerto virtual com port al software en la computadora donde reconocerán el Arduino automáticamente. El software incluye un monitor serial que permite ver datos textuales simples del Arduino. Los leds de TX y RX integrados en la placa perderán cuando se transmite información por medio del chip ATmega8U2 y la conexión USB a la computadora.

El ATmega2560 también soporta comunicación I2C (TWI) y SPI. El software de Arduino incluye Biblioteca para simplificar el uso del bus I2C.

Arduino Ethernet

El Arduino Ethernet es una placa con un microcontrolador ATmega328. Tiene 14 terminales input/output, 6 entradas analógicas, un oscilador de cristal a 16 MHz, entrada para un conector RJ45, un jack de alimentación, un header ICSP y un botón de reset. [G]

Se basa en el chip Wiznet W5100 Ethernet que proporciona una red (IP) por medio de la biblioteca de Ethernet, permitiendo escribir sketches que se conectan a Internet a través de comandos.

Dispone de una ranura para insertar una tarjeta micro-SD, que puede ser usada para almacenar archivos para servir a través de la red

Resumen

Microcontrolador	ATmega328
Voltaje de Operación	5V
Voltaje de Entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Terminal Digitales I/O	14 (4 proveen una salida PWM)
Terminal Input Analógicos	6
Corriente DC por la terminal I/O	40 mA.
Corriente DC para la terminal a 3.3V	50 mA.
Memoria Flash	32KB (ATmega 328) 0.5 Kb son usados como gestor de arranque.
SRAM	2 KB (ATmega 328)
EEPROM	1 KB (ATmega 328)
Velocidad de Reloj	16 MHz

Terminales Reservados:

- PIN10 al 13 usados para SPI
- PIN 4 usado para la memoria SD
- 2 Interruptor Ws100 (puenteado)

Alimentación

La placa puede alimentarse mediante un cable FTDI/cable serial USB, un módulo opcional PoE (Power-over-Ethernet) o con una alimentación externa con un eliminador.

Las terminales de alimentación internas son los siguientes:

- **VIN.** La terminal de entrada de voltaje para el Arduino cuando se utiliza una fuente de alimentación externa (cuando no se tiene una alimentación de 5 volts por el USB u otra fuente regulada). Uno puede suplir de voltaje a través de esta terminal o por el jack de alimentación que pasa por esta terminal.
- **5V.** Una fuente de alimentación regulada utilizada para el microcontrolador y otros componentes de la placa. Esta proviene de un regulador integrado en la placa.
- **3.3V.** Fuente de alimentación generada por un regulador integrado en la placa. La corriente máxima generada es de 50 mA.
- **GND.** Terminal de Tierra.

El módulo PoE opcional está diseñado para extraer la energía de un cable Ethernet trenzado categoría 5:

- IEEE802.3af
- Baja relación señal a ruido en la salida (100mVpp).
- Voltaje de entrada en el rango de 36V a 57V.
- Protección contra sobrecargas y descargas.
- Salida de 9V.
- Alta eficiencia de conversión DC/DC: typ 75% °50% de carga.

Memoria

El ATmega328 tiene 32KB de memoria (0.5KB son utilizados como gestor de arranque), 2KB de SRAM y 1KB de EEPROM.

Input/Output

Cada uno de las 14 terminales digitales del Mega pueden ser utilizados de entrada o salida (input/output) mediante las funciones *pinMode( ), digitalWrite( )* y *digitalRead( ).*Cada pin opera a 5 volts generando un máximo de 40mA.

- **Serial 0: (RX) y 1 (TX).** Son utilizados para recibir (RX) y para transmitir (TX) datos TTL seriales.
- **Interruptores Externos: 2 y 3.** Estas terminales pueden ser configurados con un trigger e interrumpen un valor bajo, un alza, una caída o un cambio de valor.
- **PWM: 3, 5, 6, 9 y 10.** Proveen un PWM de 8 bits de salida con la función *analogWrite( )*.
- **SPI: 10 (SS), 11 (MISO), 12 (MOSI) y 13 (SCK).** Estas terminales soportan comunicación SPI.
  LED: 9. Un LED incorporado que está conectado a la terminal digital 9. Cuando el valor de la terminal es HIGH el DED se enciende y cuando el valor es LOW se apaga.

La placa Ethernet tiene 6 entradas analógicas numeradas de A0 a A5, cada una provee 10 bits de resolución. Por default miden de tierra a 5 volts siendo posible cambiar el valor límite mediante la terminal AREF y la función *analogReference ( )*.

Hay otras terminales con funciones especiales:

- **TWI: A4 (SDA) y A5 (SCL).** Soportan comunicación TWI.
- **AREF.** Hace una referencia al voltaje de entrada.
- **Reset.** Brinda un LOW para resetear el microcontrolador. Generalmente se agrega un botón a las placas.

Comunicación

El Arduino Ethernet tiene un gran número de formas para comunicarse con una computadora, otro Arduino u otros microcontroladores.

El ATmega328 soporta comunicaciones I2C y SPI. Puede conectarse a una red alámbrica por medio de Ethernet. Cuando se conecta a una red, se necesita proveer una dirección IP y MAC para acceder. La Biblioteca Ethernet soporta estas funciones.

La placa contiene un lector de memorias SD que pueden acceder mediante la biblioteca SD dando entrada a la SD por medio de la terminal 4.

Arduino Wi-Fi Shield

El Arduino Wi-Fi Shield es una placa de adaptación del módulo Wi-Fi WIZ610wi de WIZnet. Éste shield da conectividad inalámbrica a una red Wi-Fi con o sin salida a internet, es compatible con las plataformas Duamilanove, Mega y Uno. El módulo Wi-Fi WIZ610wi posee el stack TCP/IP por hardware lo que lo hace ser una de las plataformas más estables del mercado, sin necesidad de ocupar recursos del procesador o microcontrolador en tareas de comunicación. [G]

Resumen

Microcontrolador	WIZ610wi
Voltaje de Operación	5V
Voltaje de Entrada (recomendado)	5-9V
Corriente de consumo	500 mA. aprox.

Terminales Reservadas

Dado que es un shield, necesita forzosamente de una placa o de un microcontrolador externo para realizar sus funciones, al estar sobre una placa Arduino comparte las terminales de salidas tanto analógicas como digitales salvo algunos reservados.

- **Digital 0: (Serial RX).** Conexión a Rx del puerto serial por hardware. No utilizado
- **Digital 1: (Serial TX).** Conexión a Tx del puerto serial por hardware. No utilizado
- **Digital 2: (SoftSerial RX).** Conexión a Rx del puerto serial por software
- **Digital 3: (SoftSerial TX).** Conexión a Tx del puerto serial por software
- **Digital 4: (PWR_OFF).** Conexión para encendido o apagado del shield
  ***Alto:** Apaga regulador de voltaje
  ***Bajo:** Enciende regulador de voltaje

- **Digital 7: (AUX_LED).** Conexión para LED auxiliar

- **Digital 12: (WIZ_CFG).** Terminal de control de modo de configuración.
  ***Alto:** Entra a modo de configuración
  ***Bajo:** Sale de modo de configuración

- **Reset: (RESET_ARD).** Reset de placa Arduino

Led, Botones y Jumpers

- **Led PWR:** Led indicador de que placa se encuentra energizada.
- **Led WIR:** Led indicador de actividad en la interfaz Wi-Fi del módulo.
  Estático: Sin transferencia de datos.
  Parpadeando: Transmitiendo o recibiendo datos.
- **Led RS232:** Led indicador de actividad en la interfaz RS232 del módulo.
  Estático: Sin transferencia de datos.
  Parpadeando: Transmitiendo o recibiendo datos.
- **Led AUX:** Led configurable por el usuario. Corresponde a la terminal Digital 7 de la placa Arduino.
- **Botón RST_ARD:** Botón de reset de placa base Arduino.
- **Botón RST_WIZ:** Botón de reset de módulo Wi-Fi.

Éste botón resetea al módulo Wi-Fi a sus valores de fábrica si se mantiene presionado por más de 3 [seg].

- **Jumper RX:** Jumper para usar puerto serial RX por hardware de placa Arduino.

Si utiliza éste puerto de comunicaciones, pierde la capacidad de programar la tarjeta Arduino con el Shield montado.

- **Jumper TX**: Jumper para usar puerto seria TX por hardware de placa Arduino.

Si utiliza éste puerto de comunicaciones, pierde la capacidad de programar la tarjeta Arduino con el Shield montado.

Adafruit Data Logger Shield

El Data Logger Shield de Adafruit permite guardar datos en archivos en cualquier formato FAT16 o FAT32 de memoria SD para una futura lectura o graficado para un análisis de tendencias. Incluye un RTC[46] para etiquetar los datos con el tiempo y fecha correctos y tener un mayor control del censado de datos. [9]

Alimentación

- **3V.** Es una terminal fuera del regulador con referencia a 3V y 50 mA.
- **SQ.** Es la salida Square wave (Onda Cuadrada) opcional de la RTC. Se tiene que enviar un comando para activar la salida y así conseguir opcionalmente una onda cuadrada de precisión. Principalmente se utiliza para pruebas.
- **WP.** Es una terminal de protección contra escritura de la tarjeta SD, se puede utilizar para detectar si la terminal de protección contra escritura está en la tarjeta mediante la comprobación de esta terminal.
- **CD.** Detecta la tarjeta SD. Cuando se inserta una tarjeta SD está conectado a tierra. Se sugiere usar el pull-up interno en una terminal de Arduino para esta plataforma.
- **CS.** Es la terminal que selecciona el chip para la tarjeta SD. Debido a que esto puede causar conflicto se recomienda cortar la pista a la terminal 10, se puede soldar a cualquier terminal digital y especificarlo por medio de software.
- **L2 y L1.** Son LED opcionales para los usuarios. Se conectan a cualquier terminal digital, se con figuran en alta para activar el LED correspondiente. Los LED tienen una resistencia de 470 ohms en serie.

[46] RTC: *Real Time Clock*, "Reloj en Tiempo Real".

Real Time Clock (RTC)

Cuando se obtienen datos es muy útil utilizar etiquetas de hora y fecha. De esta manera se puede tener un registro detallado del momento preciso en que se realizó la adquisición de los datos. [9]

Arduino contiene contadores y timers internos que pueden activar y desactivar el microcontrolador por periodos de tiempo de minutos o días. Estas funciones solo mantienen un registro del tiempo transcurrido desde que el Arduino fue apagado, por lo que al encenderse estos contadores vuelven a estar en 0. El Arduino no puede reconocer la fecha y hora exacta en la que nos encontramos, solamente reconoce cuantos milisegundos han transcurrido desde que se encendió por última vez.

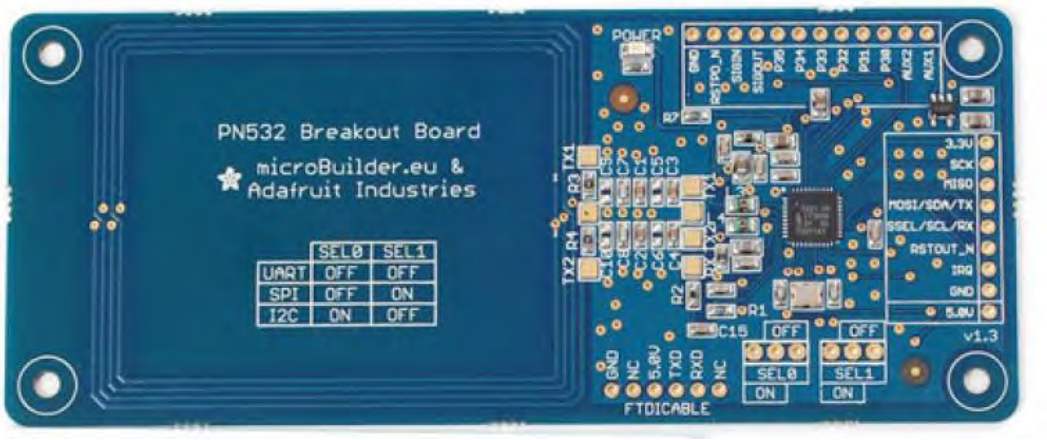
Si se desea programar la hora en el Arduino se pueden sincronizar estos contadores con una función que nos dé la hora, pero al momento en que se des energice la placa el contador volverá a 0 y la hora que indicará será 00:00.

Dada la importancia de tener una etiqueta de tiempo y fecha para algunos proyectos y aplicaciones, se utiliza un Data Logger que guardará la hora sin importar que la placa del Arduino se des energice. El chip RTC es el encargado de realizar esta función haciéndolo posible con una pila externa que puede alimentarlo. Este chip puede reconocer la fecha inicial programada y el paso de los días y los años en el área en la que fue programada, si se cambia de uso horario es necesario hacer un ajuste.

Para e RTC se utiliza el DS1307 ya que es económico, fácil de soldar y puede funcionar por años con una sola pila

Chip DS1307

Adafruit PN532 RFID/NFC



Es un módulo nos permite establecer una comunicación RFID/NFC entre Tags y el microcontrolador de Arduino. Se puede utilizar para aplicaciones de seguridad, acceso, monitoreo de personal o mercancía, registro de inventario, entre otras. [10]

El chip PN532 y su salida, está diseñado para sistemas de 3.3V. Para poder utilizarse en el Arduino que proporciona 5V es requerido un regulador interno que baje el voltaje a 3.3V. Este sistema se encuentra embebido en la placa, por lo que no es necesario hace algún cambio interno.

Comunicación NFC con I2C

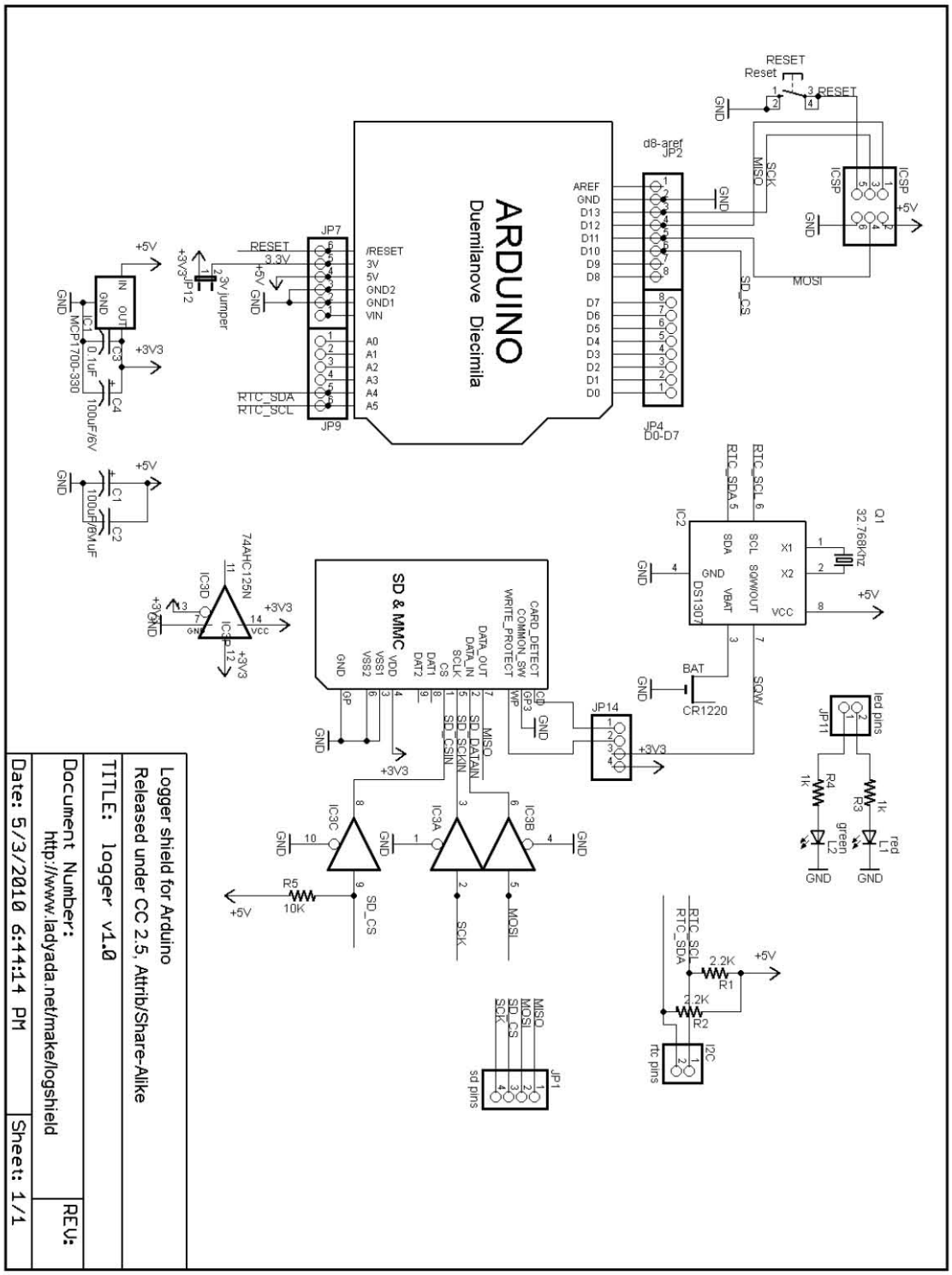
El shield está diseñada para ser utilizada con I2C por defecto. I2C es un protocolo que utiliza 2 terminales analógicas 4 y 5 integrados ya en hardware, lo que hace que no puedan cambiarse. De esta manera se comunican una terminal con otro utilizando una terminal de interrupción extra. La ventaja del I2C es que se genera un bus compartido que a diferencia de SPI y la serie TTL, se pueden utilizar el número de sensores que se desee todos comunicados a través de estas terminales siempre que las direcciones no colisionen o entren en conflicto. La terminal de interrupción es útil ya que en lugar de estar solicitando mensajes de confirmación a cada momento del sensor NFC, el chip nos avisa cuando un Tag NFC entra en alcance de la antena.

El shield es un drop-in compatible con cualquier Arduino Classic (UNO, Duemilanove, Diecimilla, etc. usando el ATmega168 o '328), así como cualquier Mega R3 o posterior.

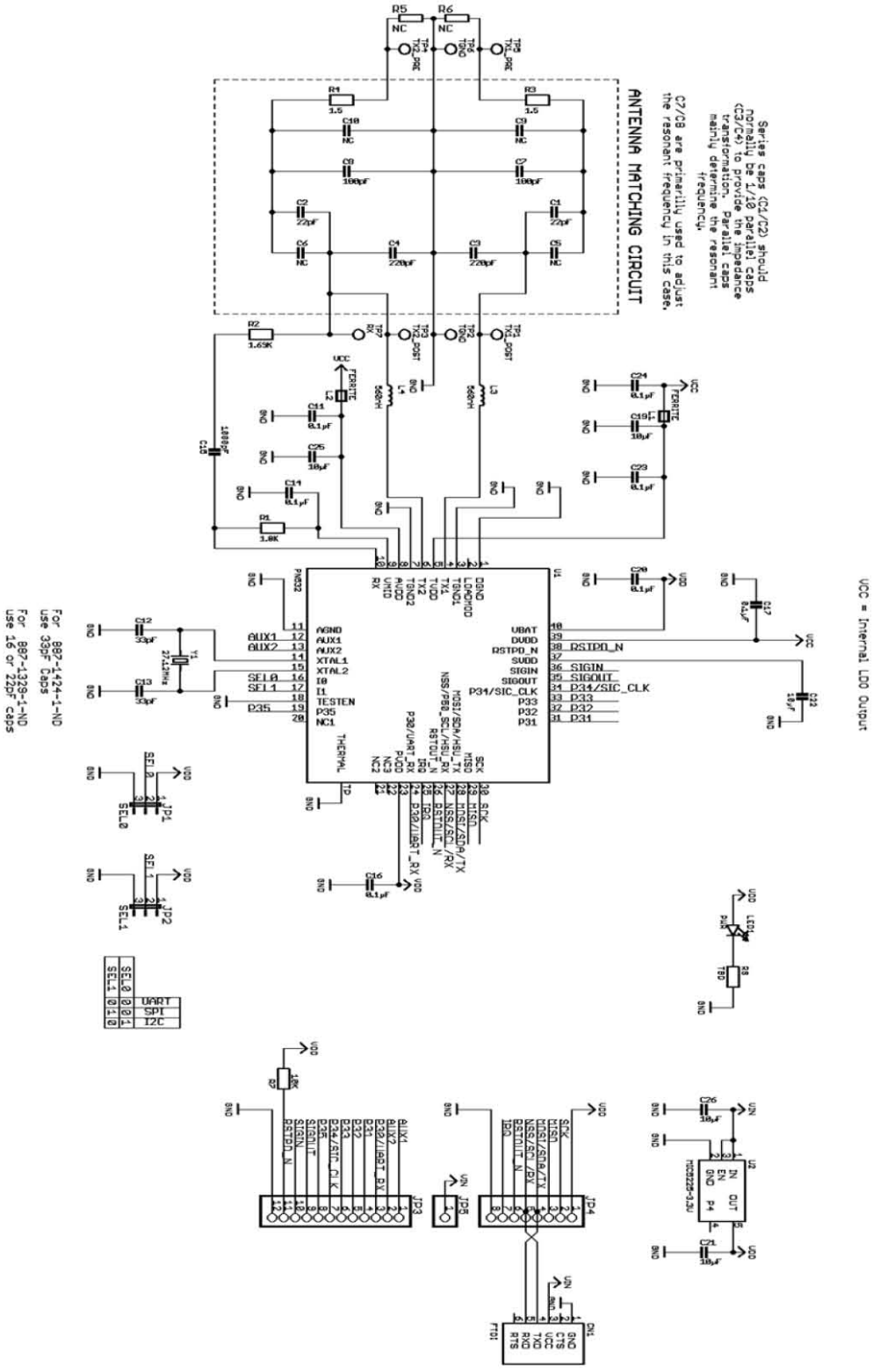
Apéndice B

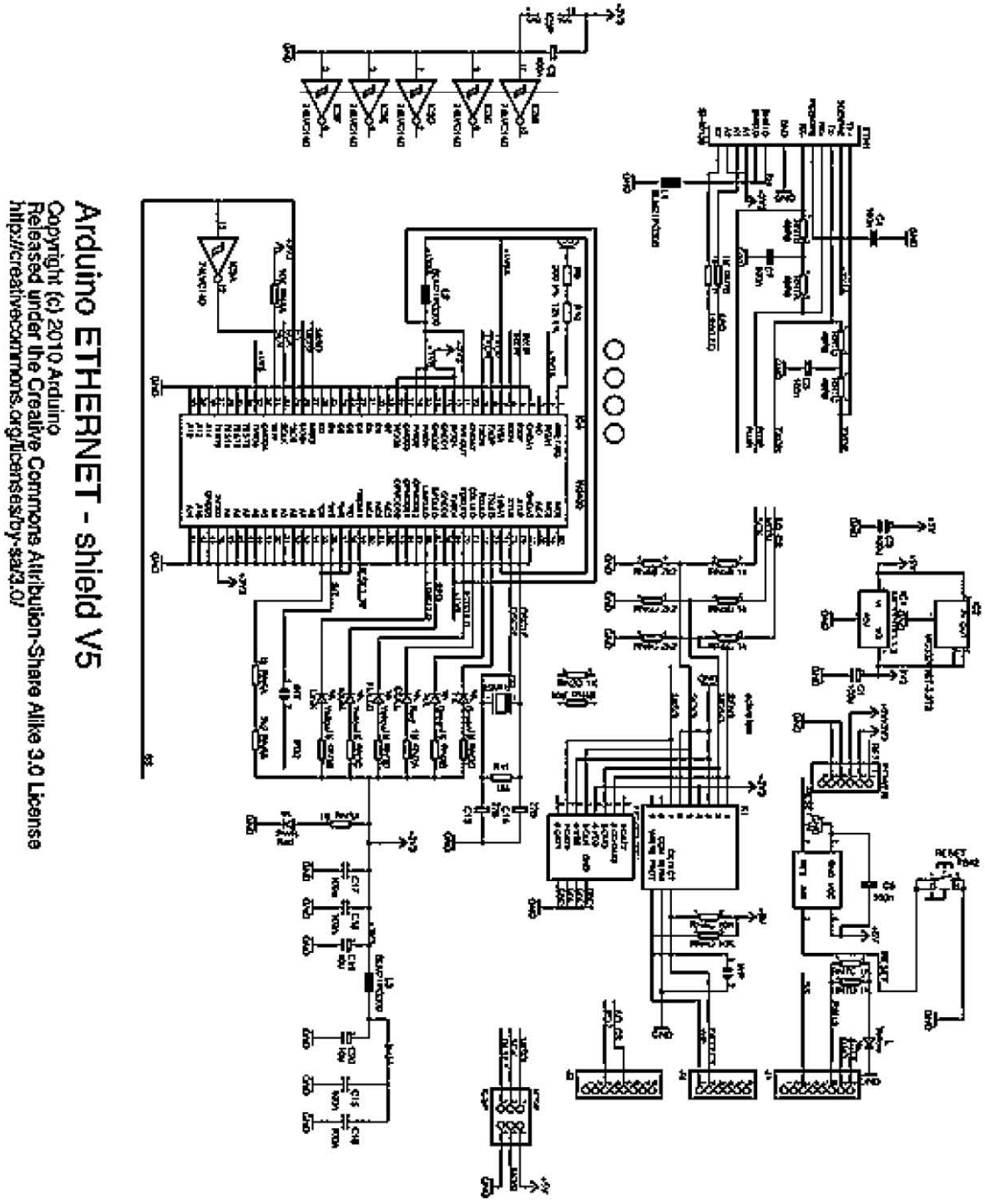
Esquemáticos de los dispositivos utilizados

Adafruit - Datalogger

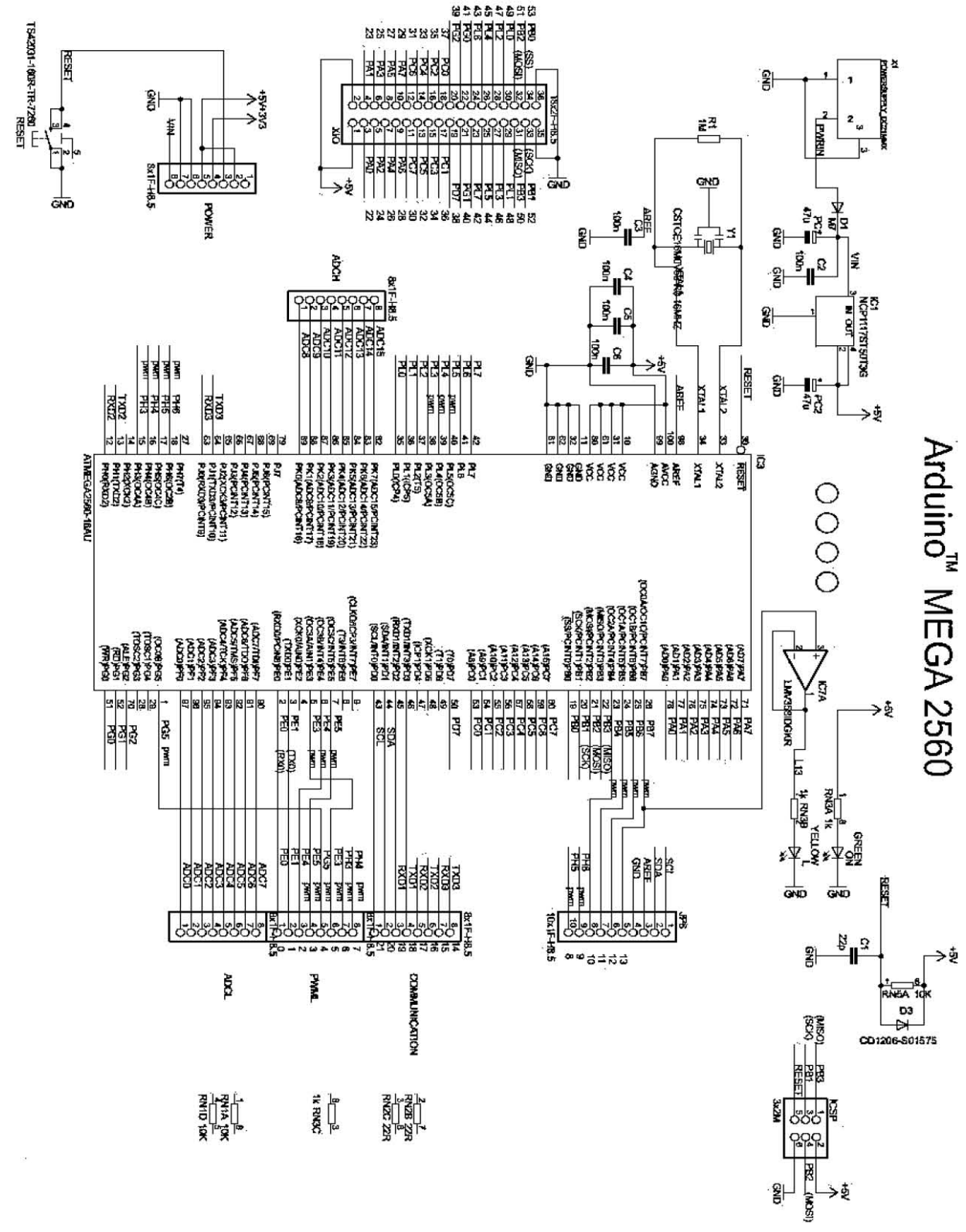


Adafruit PN532 NFC_RFID

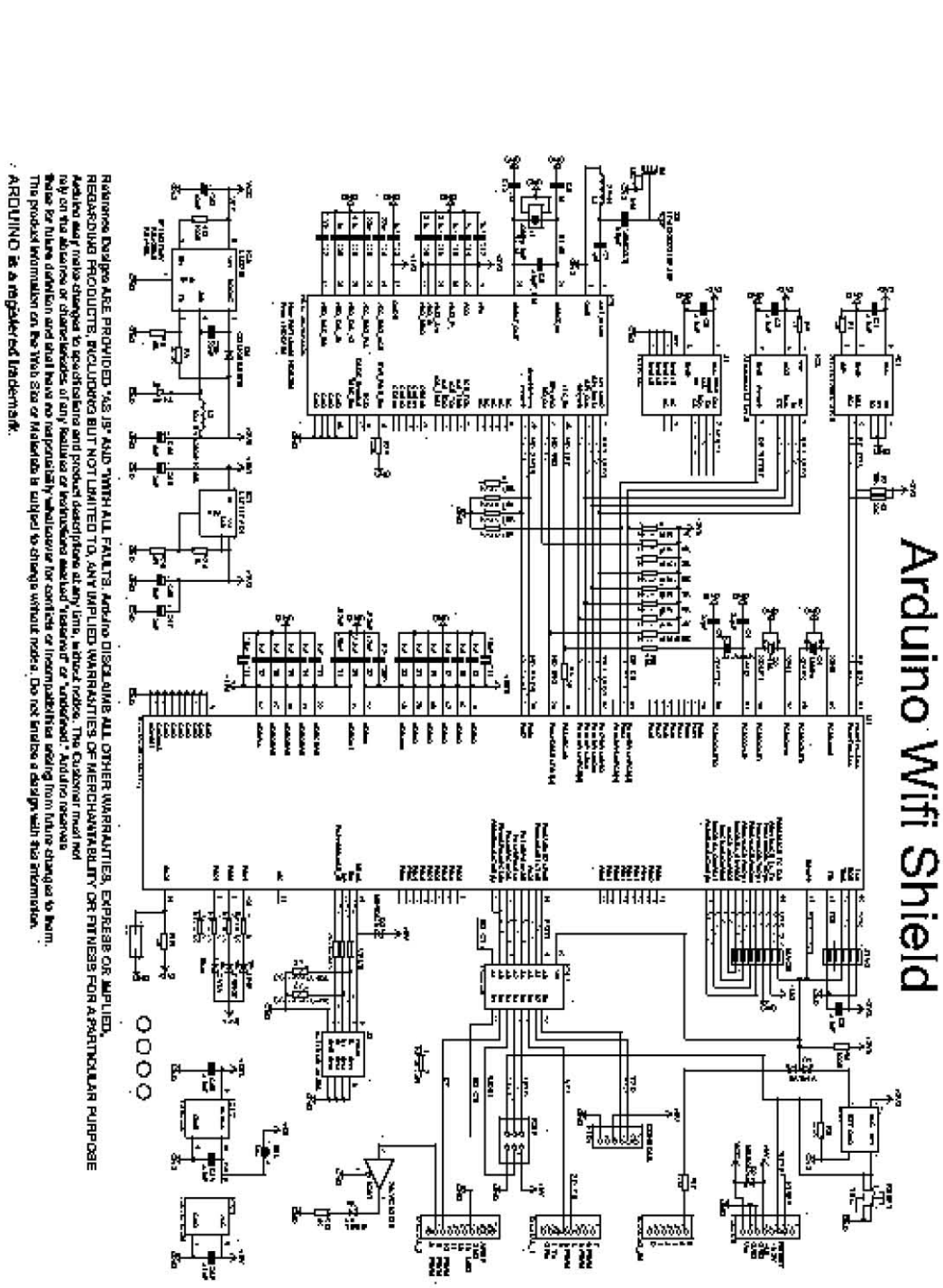



Arduino ETHERNET - shield V5
Copyright (c) 2010 Arduino
Released under the Creative Commons Attribution-Share Alike 3.0 License
http://creativecommons.org/licenses/by-sa/3.0/

Arduino - Mega2560 Schematic



Arduino - Wi-Fi shield Schematic



Apéndice C

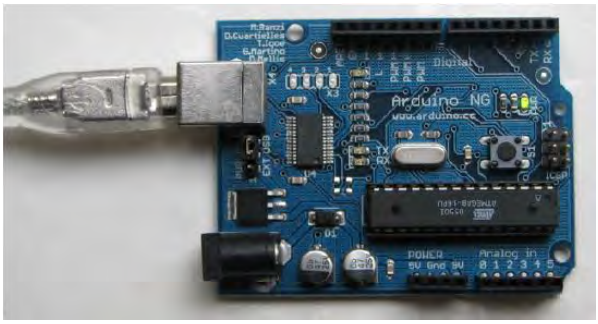
Instalación del Software de Programación

Para este prototipo se utilizó la versión 1.0.6 del compilador de Arduino. Todos los pasos que a continuación se detallan están descritos para el sistema operativo Windows.

Debemos ir a la página de Arduino “http://www.arduino.cc/en/Main/Software” y buscar la versión 1.0.6.

Al finalizar la descarga, descomprimimos el el archivo descargado. Es importante conservar la estructura de las carpetas.

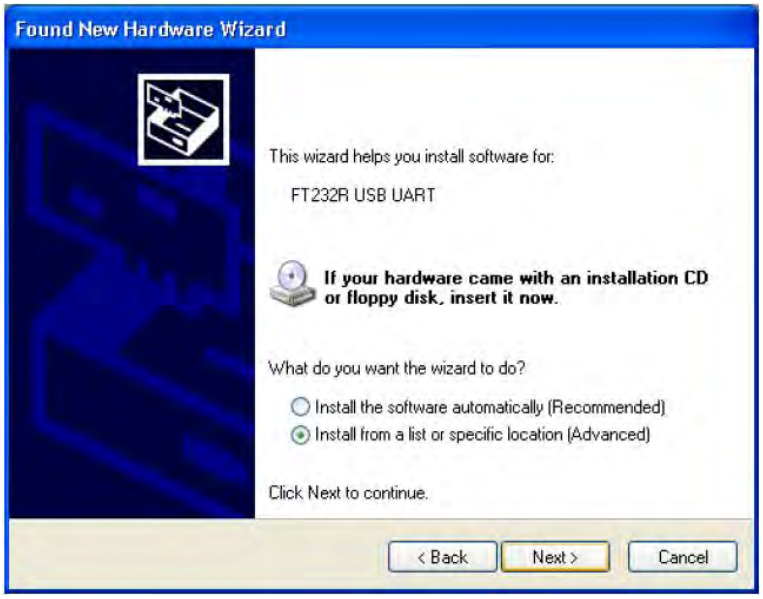
Se requiere la instalación de los drivers USB, por lo que sin ejecutar el compilador Arduino conectamos la placa con un cable USB.



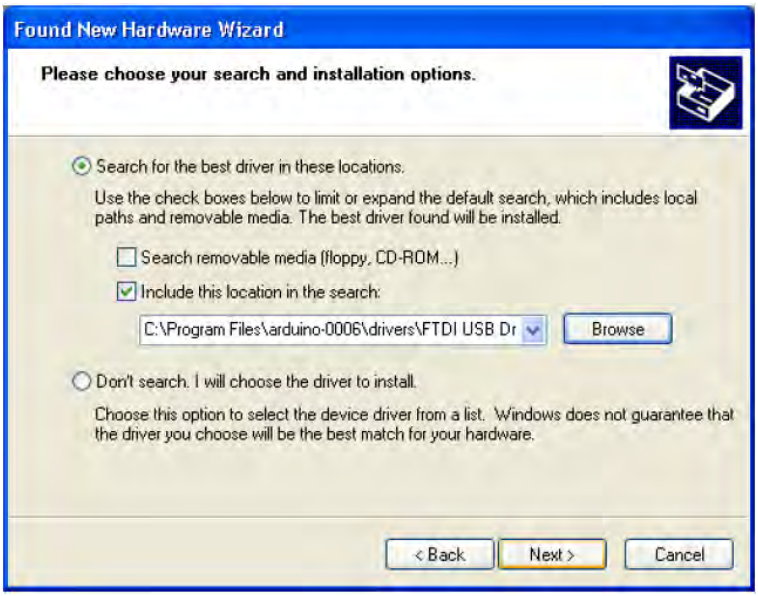
Conexión de USB Serial del Arduino

Esto nos despliegará el asistente para *Añadir Nuevo Hardware de Windows* y nos pedirá si queremos conectar *Windows Uptade*, nosostros seleccionamos *No*.

A continuación nos pedira el método de instalación de drivers. Seleccionamos la opción de *Instalación de una lista de una ubucación específica (Avanzado).*

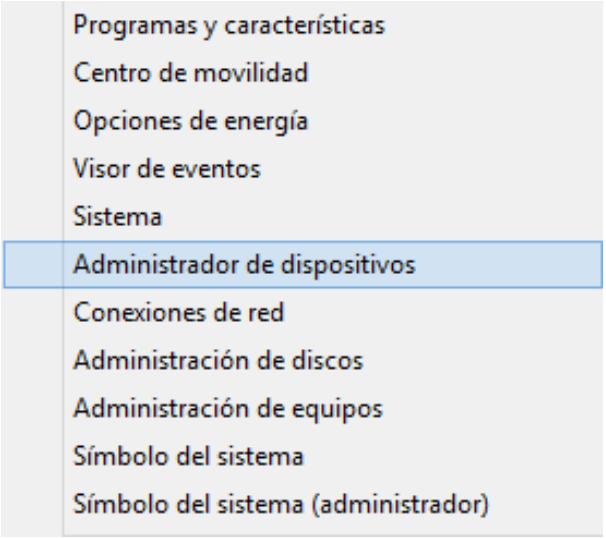


Se desplegará otra ventana donde debemos poner la ubicación de la carpeta de drivers. En el botón de *Buscar* damos click y ponemos la ubicación de nuestra carpeta que descomprimimos y damos aceptar.

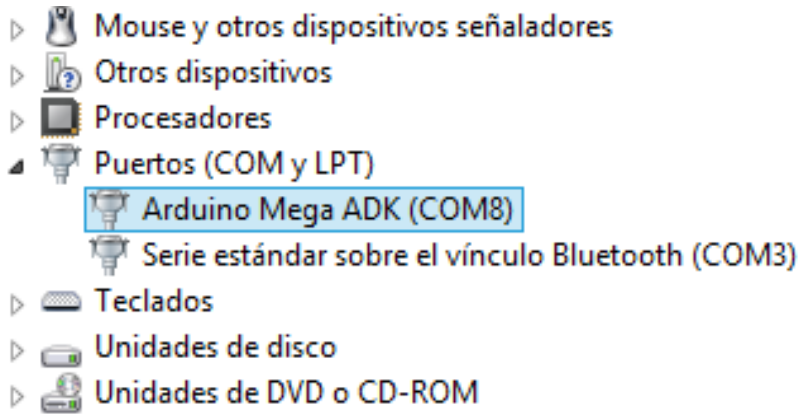


Si realizamos los pasos anteriores correctamente y las carpetas que descargamos estan íntegras nos aparecerá de nuevo la ventana del asistente para *Añadir Nuevo Hardware de Windows* con un mensaje que completo la instalación del software.

Para comprobar que los drivers están correctamente instalados debemos ir al *Administrador de Dispositivos*. Damos click derecho a la ventana de Windows de nuestra barra de tareas y seleccionamos *Administrador de Dispositivos.*

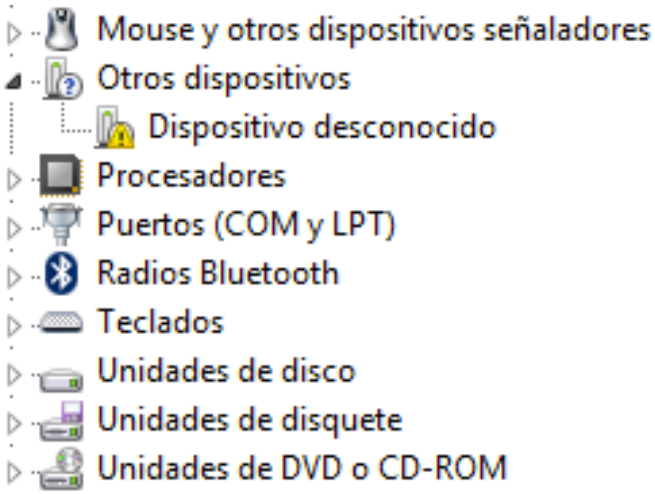


Una vez seleccionada la ventana Buscamos los puertos (COM y LPT) y desplegamos la pestaña.



Aquí nos indicara si los drivers fueron instalados correctamente y nos mostrará el numero de puerto COM asignado para nuestra placa Arduino.

Si nos aparece un símbolo de advertencia significa que los drivers no fueron instalados correctamente y nuestras placas no seran reconocidas por el compilador, debemos revisar nuevamente la carpeta descomprimida y revisar en el asistente para *Añadir Nuevo Hardware de Windows* que la direción donde se encentran los drivers sea correcta.



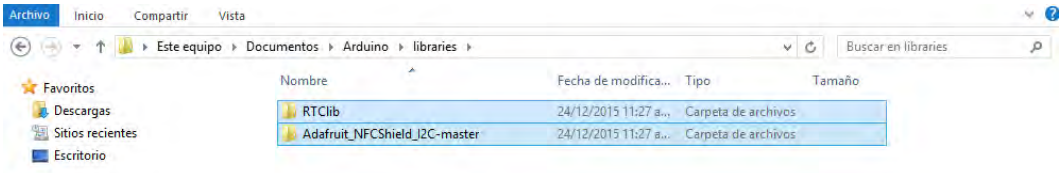
Apéndice D

Instalación de Bibliotecas

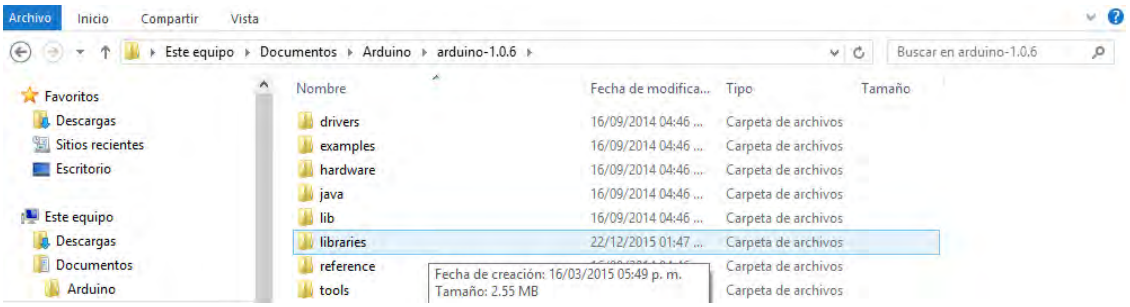
Para el funcionamiento de los dispositivos de Adafruit es necesario descargar sus bibliotecas ya que sin estas el compilador no reconocerá los comandos correspondientes a los dispositivos de Adafruit.

Debemos ir a las paginas de Adafruit (*https://learn.adafruit.com/adafruit-pn532-rfid-nfc* y *https://learn.adafruit.com/adafruit-data-logger-shield/overview)* para descargar las bibliotecas **"***Adafruit_NFCShield_I2C-master***"** y **"***RTClib***".**

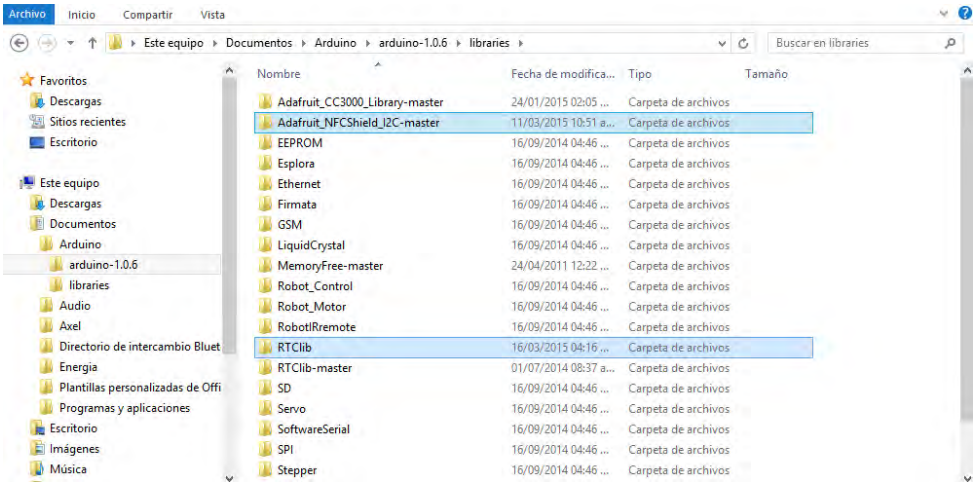
Una vez descargadas estas bibliotecas tenemos que agregarlas a la carpeta *"libraries"* que se encuentra dentro de nuestra carpeta Arduino.



Copiamos y pegamos ambas bibliotecas dentro de la carpeta "libraries".



En esta carpeta se encuentran todas las bibliotecas que puede soportar nuestro compilador.



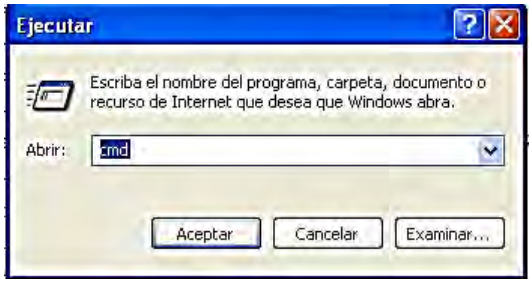
Apéndice E

Configuración Inicial de los parámetros de red

Para que una placa Arduino Ethernet pueda utilizar una red TCP/IP:

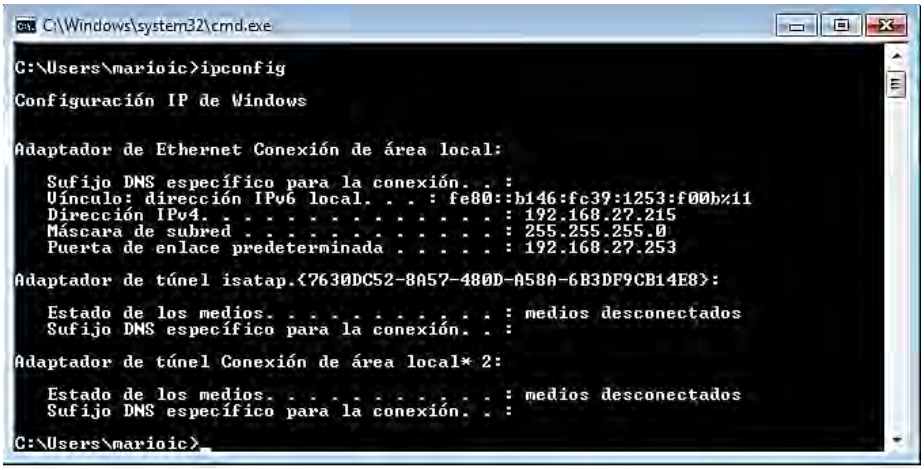
3. Se asignan una serie de valores de configuración (dirección MAC y/o dirección IP)
4. Se requiere conocer la IP del servidor al cual nos vamos a conectar, este puede ser una computadora o una página de internet. Para esto se necesitan seguir los siguientes pasos:

Entrar en la consola Windows “ejecutar” y escribir **cmd.**



Para conocer la IP de una computadora escribimos “*ipconfig*”

Este comando nos despliega una lista donde podemos ver cuál es el IP de la máquina dentro de una red. **ES IMPORTANTE RECORDAR EL NUMERO IP POR QUE SERÁ UTILIZADO POSTERIRMENTE**. En nuestro caso será **[192.168.27.215]**.



Apéndice F

Asignación y verificación de IP

1. Asignamos las bibliotecas SPI. Ethernet.h

```
#include <SPI.h>
#include <Ethernet.h>
```

2. Asignamos direcciones MAC e IP.

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,27, 217);
```

3. Indicamos el puerto asignado para comunicación HTTP (siempre es el 80)

```
EthernetServer server(80);
```

4. Abrimos comunicación con los puertos seriales de Arduino

```
void setup() {
Serial.begin(9600);
```

5. Abrimos comunicación Ethernet y con el servidor

```
Ethernet.begin(mac, ip);
server.begin();
```

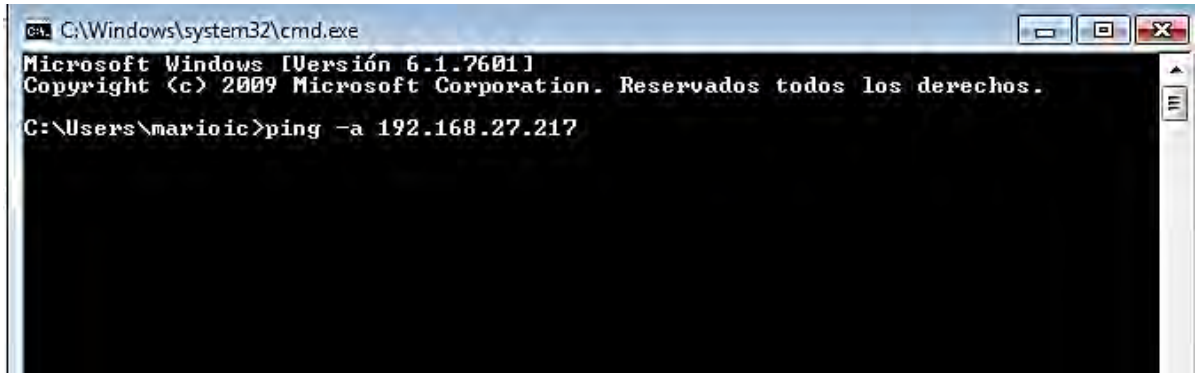
6. Imprimimos dirección IP por medio del comando Ethernet.localIP()

```
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
```

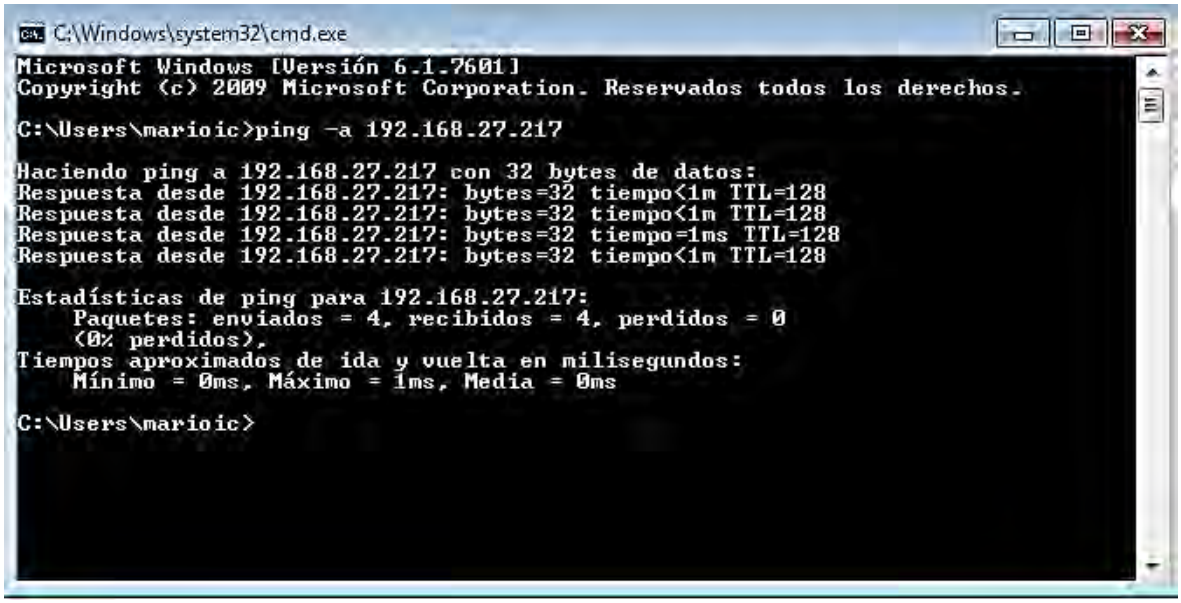
Donde el IP asignado es [192, 168, 27, 217] recordando el IP de nuestra máquina cambiamos sólo un número de la última tercia de números para asegurar que estaremos en la misma red

Ahora para verificar que estamos en la misma red nuevamente entrar en la consola Windows "ejecutar" y escribir "*cmd*"**.**

Escribir el comando ping –a “IP asignado al Arduino” en este caso [192.168.27.217].



Este comando envía cuatro paquetes de datos al IP asignado para verificar que están en red. Si esto es correcto nos indica que los paquetes fueron enviados con éxito.



Teniendo nuestra pc/laptop en red con nuestro Arduino Ethernet podemos ahora comenzar sin ningún problema. Si el Arduino Ethernet no está en red con la pc/ laptop no se puede seguir ya que nunca habrá comunicación entre ellos.

Bibliografía

[1] - Bernhard H. Walke, Stefan Mangold, Lars Berlemann, "*IEEE 802 Wireless Systems: Protocols, Multi-Hop Mesh/Relaying, Performance*", Ed. Wiley, 2007. pag. 4, 5 y 6.

[2] - Bruce A. Hallberg, "*Fundamentos de Redes*", Ed. Mc Graw Hill, 4ta Edición, 2006 pag. 28-32 y 69-75.

[3] - Stallings William, "*Comunicaciones y Redes de Computadores*", Ed. Prentice Hall, 6ta Edición, Madrid, 2004, pag. 41-53.

[4] - Charles E. Spurgeon, "*Ethernet: The Definitive Guide*", Ed. O´Relly, 1ª Edición 2000.

[5] - Houda Labiod, Hossam Afifi, Costantino de Santis, "*Wi-Fi, Bluetooth, Zigbee and Wi-Max*" Ed. Springer, 2007.

[6] - Tomasi Wayne, "*Sistemas de Comunicaciones Electrónicas*", Ed. Prentice Hall, 4ª Edición, 2003.

[7] - V. Daniel Hunt, Albert Puglia, Mike Puglia, "*RFID A guide to Radio Frequency Identification*", Ed. Wiley-Interscience, EUA, 2007.

[8] - Klaus Finkenzeller, Rote Muller, "*RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*", Ed. Wiley, 3ª Edición, Agosto 2010.

[9] – Bill Earl, "Adafruit Data Logger Shield", Adafruit Learning System, NY, EUA, 2015.

[10] – Lady Ada, "Adafruit PN532 RFID/NFC Breakout and Shield", Adafruit Learning System, NY, EUA, 2015.

[A] - http://www.ieee802.org (Consulta: 21/02/2016).

[B] - http://www.wi-fi.org/w (Consulta: 05/03/2016).

[C] - http://ieeexplore.ieee.org/document/4544755/ (Consulta: 09/11/2016).

[D] - http://ieeexplore.ieee.org/document/4573292/ (Consulta: 09/112016).

[E] - http://ieeexplore.ieee.org/document/5716530/?reload=true&arnumber=5716530 (Consulta: 09/11/2016).

[F] – http://www.textoscientificos.com/redes/modulacion/analogica-digital (Consulta: 29/03/2016).

[G] – https://www.arduino.cc/ (Consulta: 18/05/2016).