UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL

Facultad de Educación Técnica para el Desarrollo

TESIS DE GRADO

Previo a la obtención del título de:

Ingeniero en Telecomunicaciones
Con Mención en Gestión Empresarial

TÍTULO:

Investigación y Desarrollo de un Sistema de Seguridad Electrónica Personal Utilizando el Microcontrolador ATMEL AT1NN84 y el Modulo GPS V23993

Realizado por:

Xavier Ulloa Bustos
Jennifer Villavicencio B.

Director:

Ing. Luís Córdova Rivadeneira

Guayaquil – Ecuador
2008 - 2009

TESIS DE GRADO

Titulo:

Investigación y Desarrollo de un Sistema de Seguridad Electrónica Personal Utilizando el Microcontrolador ATMEL AT1NN84 y el Modulo GPS V23993

Presentada a la Facultad de Educación Técnica para el Desarrollo, Carrera de Ingeniería en Telecomunicaciones de la Universidad Católica de Santiago de Guayaquil

Realizado por:

Xavier Ulloa Bustos
Jennifer Villavicencio Bravo

Para dar cumplimiento con uno de los requisitos para optar por el titulo de:

Ingeniero en Telecomunicaciones con Mención en Gestión Empresarial

Ing. Luis Córdova Rivadeneira

Director de Tesis

Ing.……
Vocal

Ing.…….
Vocal

Ing. Héctor Cedeño
Decano de la Facultad

Ing. Luis Tutiven
Director de Carrera

DEDICATORIA

Este proyecto va dedicado a nuestros padres, ya que gracias a su esfuerzo y constancia hemos podido llegar a este punto culmínate de nuestras carreras.

A nuestros maestros que con su dedicación y esfuerzo formaron en nosotros profesionales capaces de enfrentar las diversas circunstancias y retos que el mundo actual nos plantea, sin olvidar los valores que nos fueron inculcados a lo largo de nuestra permanencia en la universidad.

AGRADECIMIENTO

Agradecemos ante todo a Dios que nos ha permitido concluir con satisfacción ésta etapa de nuestras vidas, convirtiéndonos en profesionales con todos los principios morales y éticos que una educación Católica puede brindar.

A nuestros padres que son los verdaderos artífices de nuestros éxitos y triunfos, que con su constante lucha y guía nos han sido de ejemplo de superación ante toda adversidad.

A nuestros maestros que han sabido compartir con sabiduría sus conocimientos y experiencias en especial a los Ingenieros: Luis Córdova, Pedro Tutiven, Hector Cedeño, Efrain Suarez, Miguel Torres, Manuel Romero y Jaime Monteverde (✞) que más que ser profesores, se convirtieron en verdaderos amigos que ayudaron a fortalecer nuestro carácter para nuestro desarrollo personal y profesional.

Gracias.

PROLOGO

Esta obra es el respaldo textual de la tesis “INVESTIGACIÓN Y DESARROLLO DE UN SISTEMA DE SEGURIDAD ELECTRÓNICA PERSONAL UTILIZANDO EL MICROCONTROLADOR ATMEL AT1NN84 Y EL MODULO GPS V23993” efectuada bajo la dirección del profesor ING. LUIS CORDOVA RIVADENEIRA, que se pondrá en funcionamiento como una aplicación seguridad electrónica personal.

El dispositivo nos permitirá conocer la ubicación de una determinada persona u objeto dentro del territorio continental ecuatoriano, el cual dispondrá de su respectivo software de operación con un mínimo margen de error y bajo tiempo de respuesta.

INDICE GENERAL

INDICE DE ILUSTRACIONES

INDICE DE TABLAS

RESUMEN

Se presentan en este trabajo, los principales aspectos relacionados con el análisis, implementación y uso correcto de un dispositivo de seguridad electrónica GPS, exponiendo con detalles sus componentes y funcionamiento.

El principal componente del dispositivo es el microcontrolador que es un pequeño dispositivo con la capacidad de gobernar uno o varios procesos el cual consta de procesador, memoria de datos e instrucciones, líneas de entrada y salida, oscilador de reloj, bus interno de dato para la interconexión interna de estos bloques y módulos controladores de periféricos.

Estos dispositivos son programados a medida de las necesidades del usuario usando los diferentes lenguajes que se han desarrollado para los microcontroladores, los más usados son el Ensamblador, el BASIC y el C.

Los principales fabricantes de estos dispositivos son Intel, Motorola, ATMEL AVR y PIC, cada uno con modelos con similares características haciendo este mercado muy competitivo.

El proceso de geo-referenciacion de el equipo lo confiamos al Sistema de Posicionamiento Global el cual es un sistema global de navegación por satélite, que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros.

Cuando deseamos determinar la posición, el receptor capta señales indicando la posición y el reloj de múltiples satélites geoestacionarios para luego el sincronizar el reloj y calcular el retraso de las mismas, es decir la distancia a estos satélites, seguido procede a triangular para determinar su posición.

La etapa de comunicación confía de las redes celulares GSM, la cual es una manera confiable, robusta y estable de transportar los datos, misma que en nuestro país se encuentra cubriéndolo casi en su totalidad haciendo de la misma la mejor elección sobre otras opciones, que requerirían mayor implementación de equipos y harían los costos del proyecto prohibitivos.

El protocolo que utilizaremos es el sistema binario, ya que el módem que desarrollaremos es un sistema digital, el cual se comunicara por medio de impulsos eléctricos de 1's y 0's provenientes del puerto USB del PC, y de igual manera se receptaran los datos en el PC desde el módem, para que luego estos sean interpretados y procesados por la aplicación del Sistema

Para nuestro propósito hemos diseñado un sistema de gestión y monitoreo, el cual hemos denominado LocalGPS, para el mismo contamos con la ayuda de los scripts del ya conocido Google Maps en el que se encuentra cartografiado la mayor parte del mundo.

Las ventajas del uso del scripts Google Maps, es sencillamente la rápida obtención de los mapas que se encuentran en línea, la veracidad de los datos ya que esta empresa ha invertido millones de dólares en procurar que todos estos sean correctos, para nuestro propósito el mapa de la Ciudad de Guayaquil.

ANTECEDENTES

Nuestro trabajo consiste en implementar un dispositivo de seguridad electrónica GPS ya que como se ha expuesto en la estadística de secuestros express son cada vez más frecuentes en el país.

DELITO	Año	DENUNCIAS	ESTIMACION
Secuestro Express	2008	203	734
Secuestro Express	2009	252	931

Tabla. Estadisticas de los años 2008 - 2009

De esta estadística se deriva la importancia de crear este dispositivo el cual mediante el uso del sistema de posicionamiento global y las redes de telefonía celular nos permite conocer con exactitud la ubicación de un objeto o persona brindando a la ciudadanía un mecanismo de seguridad más exacto que los comercializados en la actualidad, con el objetivo de aportar a la disminución del auge delincuencial que aqueja a nuestro país.

OBJETIVO GENERAL

Diseñar y construir un sistema de localización satelital usando las tecnologías GSM y GPS para reducir la incidencia de secuestros Express en la ciudad de Guayaquil.

OBJETIVOS ESPECIFICOS

- Reconocer los componentes y herramientas de instalación del dispositivo que hemos diseñado.
- Realizar un prototipo funcional.
- Investigar la factibilidad de comercialización.

CAPÍTULO 1

Generalidades del Microcontroladores ATMEL

1.1. Definición

La palabra “Microcontrolador” define a un pequeño dispositivo que tiene la capacidad de gobernar uno o varios procesos. Por ejemplo el controlador que dispone el sistema de control de un robot, que regula el funcionamiento con la ayuda de sensores en las distintas articulaciones que miden su movilidad con la finalidad de evitar que se des coordine y se pierda el control del mismo.

Para cumplir con el objetivo de controlar desde hace algunas décadas, se ha empleado diferentes dispositivos, desde componentes de lógica discreta hasta los microprocesadores, que se rodeaban con chips de memoria y puertos de entrada y salidas sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de **Microcontrolador**. Realmente consiste en un sencillo pero completo computador contenido dentro de un circuito integrado.

1.2. Recursos comunes a todos los microcontroladores

El empleo de recursos de los microcontroladores al estar integrados en un chip, su estructura fundamental básicamente posee las mismas características. Todos deben disponer de los bloques esenciales: El procesador, memoria de datos e instrucciones, líneas de entrada y salida, oscilador de reloj, bus interno de dato para la interconexión interna de estos bloques y módulos controladores de periféricos.

1.3. Arquitectura básica

La arquitectura de los microcontroladores es la **Arquitectura Harvard**. Dispone de dos memorias independientes: una, que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias, con lo cual la velocidad del sistema aumenta.

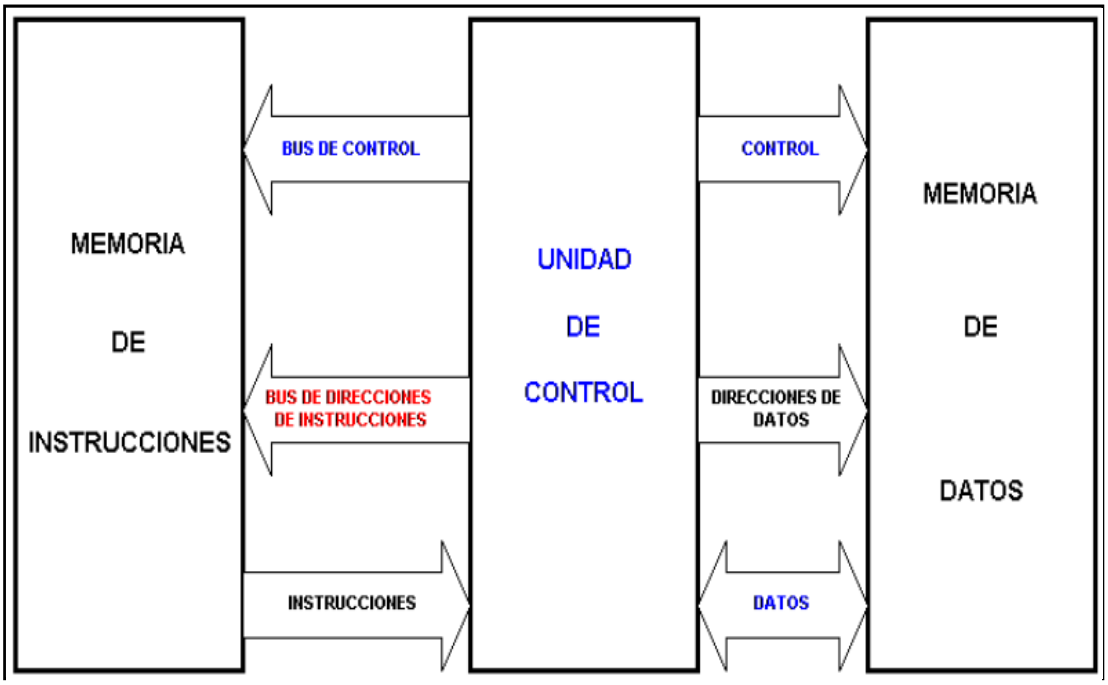

Fig. 1.1 Representación de la Arquitectura Harvard

1.4. Procesador CPU

Es la parte central e importante del microcontrolador y determina sus principales características a nivel de hardware y software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP (lenguaje de maquina) de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y en

almacenamiento del resultado. Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales:

1.4.1. RISC

(Computadores de Juego de Instrucciones Reducido) Tanto la industria de los computadores comerciales como la de los microcontroladores están orientadas hacia la filosofía RISC. En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

1.4.2. SISC

(Computadores de Juego de Instrucciones Específico) En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es “específico”, es decir, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC.

1.4.3. CISC

Computadores de Juego de Instrucciones Complejo: Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

1.5. Memoria

En los microcontroladores disponen de dos memorias integradas en el propio chip. La primera que es la memoria de programa que debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación.

La segunda memoria que es de tipo RAM, volátil, que se la emplea para guardar las variables y los datos.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM. Existen diferentes memorias no volátiles que pueden poseer los microcontroladores puesto que la aplicación y utilización de los mismos es diferente. A continuación se describen las cinco versiones de memoria no volátil.

1.5.1. ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado costo del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

1.5.2. EPROM

(Erasable Programmable Read Only Memory)

Los microcontroladores que disponen de memoria EPROM pueden borrarse y grabarse varias veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, se emplea luz Ultravioleta y se la expone a una ventana de cristal en su superficie donde somete a la EPROM durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

1.5.3. EEPROM

(Electrical Erasable Programmable Read Only Memory)

Se trata de memorias de sólo lectura, escritura, programables y borrables eléctricamente EEPROM. Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la operación de grabado y la de borrado.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se requiera sin ser retirados de dicho circuito. Para ello se usan “procesos de grabado interno” que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

En la actualidad la mayoría de los fabricantes de microcontroladores le incluye en un área de la memoria EEPROM con el objetivo de almacenar datos que no se pierda aun sin fluido eléctrico.

1.5.4. FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña. A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias **EEPROM y FLASH** son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados “en circuito”, es decir, sin tener que sacar el circuito integrado de la tarjeta.

1.6. Puertos de Entrada y Salida

La principal utilidad de los pines que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

1.7. Reloj principal

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

1.8. Recursos especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo

que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el costo, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

Temporizadores o “Timers”:

Se emplean para controlar periodos de tiempo y para llevar la cuenta de acontecimientos que suceden en el interior.

Perro guardián o “Watchdog”:

Es un temporizador que cuando se desborda y pasa por 0 provoca un reset automáticamente en el sistema.

Protección ante fallo de alimentación o “Brownout”:

Se trata de un circuito que genera un reset cuando el voltaje de alimentación V_{DD} es inferior a un voltaje mínimo establecido.

Estado de reposo o de bajo consumo:

Es un estado del sistema donde se detiene el reloj principal y sus circuitos asociados con el objetivo de ahorrar energía en periodos de tiempo donde el microcontrolador se mantiene en espera de instrucciones.

Conversor A/D:

Procesa señales analógicas convirtiéndolas en señales digitales.

Comparador analógico:

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula.

La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

Modulador de anchura de impulsos o PWM:

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

1.9. Puertos de Comunicación:

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

-**UART,** adaptador de comunicación serie asíncrona.

-**USART,** adaptador de comunicación serie síncrona y asíncrona.

-**PUERTO PARALELO**, esclavo para poder conectarse con los buses de otros microprocesadores.

-**USB** (Universal Serial Bus), que es un moderno bus serie para los PC.

-**BUS IC,** que es un interfaz serie de dos hilos desarrollado por Philips.

-**BUS CAN** (Controller Area Network), para permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles.

1.10. Lenguajes de programación

Se han desarrollado todo tipo de lenguajes para los microcontroladores, pero los más usados son el Ensamblador, el BASIC y el C. Los programas escritos en Ensamblador son compactos y rápidos, sin embargo, utiliza nemónicos inteligibles y si no están bien confeccionados resultarán de gran tamaños y lentos. Los lenguajes de alto nivel como el BASIC y el C son más fáciles de comprender y por tanto de diseñar ya que poseen grandes librerías que en su interior podremos encontrar funciones útiles para el desarrollo de nuestros proyectos.

1.11. Fabricantes

En la actualidad, gran parte de los fabricantes de circuitos integrados disponen de su propia línea de microcontroladores. Así tendremos **Intel**, que ha ido siempre por delante presentando nuevos productos, así por ejemplo el 8048 se considera el primer microcontrolador de 8 bits y lo fabricó **Intel** en la década de los 70.

Otra de las principales empresas del mundo de los dispositivos programables es **Motorola**, los microcontroladores **ATMEL AVR Solution** y los microcontroladores **PIC** de la empresa americana **Microchip** han sido conocidos en los últimos años. Su popularidad avanza día a día, siendo incluidos en la mayoría de proyectos debido a su bajo costo, reducido consumo, pequeño tamaño, fácil programación y abundancia de herramientas económicas de soporte. Otras empresas como **Hitachi, Texas, Toshiba** y **Zilog** abarcan pequeñas partes del mercado.

Todos los microcontroladores que se fabrican en el presente son buenos y el mejor no siempre es el mismo. Cambian el modelo y fabricante según la aplicación y las circunstancias que lo envuelven.

1.12. Aplicaciones

Cada vez existen más productos que incorporan un microcontrolador con el fin de aumentar sustancialmente sus prestaciones, reducir su tamaño y costo, mejorar su fiabilidad y disminuir el consumo.

Algunos fabricantes de microcontroladores superan el millón de unidades de un modelo determinado producidas en una semana. Este dato puede dar una idea de la masiva utilización de estos componentes.

Los microcontroladores están siendo empleados hablando solo en general en los campos tales como: Automoción, Aparatos portátiles, Instrumentación, Electromedicina, Domótica, Electrodomésticos, Maquinas expendedoras, Control Industrial, Sistemas de Seguridad, Sistemas de Navegación, Jugueterías y hasta los dispositivos y periféricos Informáticos, inclusive en el control de sistemas de una nave espacial.

1.13. Microcontroladores Atmel AVR.

Esta familia está basada en una nueva arquitectura RISC que incorpora memoria Flash para el programa y memoria EEPROM para los datos. Además esta arquitectura fue diseñada para ser totalmente compatible con lenguaje C, permitiendo trabajar en alto nivel

Fig. 1.2 Mic. ATMEL en Tarjeta Madre

Capítulo 2

Sistema de Posicionamiento Global - GPS

El **Global Positioning System** (**GPS**) o **Sistema de Posicionamiento Global** (más conocido con las siglas GPS, aunque su nombre correcto es **NAVSTAR-GPS**) es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros, usando GPS diferencial, aunque lo habitual son unos pocos metros. Aunque su invención se les atribuye a los gobiernos de Francia y belga, el sistema fue desarrollado e instalado, y actualmente es operado por el Departamento de Defensa de los Estados Unidos.

Fig. 2.1 Satélite NAVSTAR GPS

El GPS funciona mediante una red de 27 satélites (24 operativos y 3 de respaldo) en órbita sobre el globo, a 20.200 km, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los

que recibe unas señales indicando la posición y el reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el retraso de las señales; es decir, la distancia al satélite. Por "triangulación" calcula la posición en que éste se encuentra. La triangulación en el caso del GPS, a diferencia del caso 2-D que consiste en averiguar el ángulo respecto de puntos conocidos, se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta, o coordenadas reales del punto de

Fig. 2.2 Lanzamiento de satélites para la constelación NAVSTAR-GPS mediante un cohete Delta

Medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites.

La antigua Unión Soviética tenía un sistema similar llamado GLONASS, ahora gestionado por la Federación Rusa.

Actualmente la Unión Europea está desarrollando su propio sistema de posicionamiento por satélite, denominado Galileo.

2.1. Historia

En 1957 la Unión Soviética lanzó al espacio el satélite Sputnik I, que era monitorizado mediante la observación del Efecto Doppler de la señal que transmitía. Debido a este hecho, se comenzó a pensar que, de igual modo, la posición de un observador podría ser establecida mediante el estudio de la frecuencia Doppler de una señal transmitida por un satélite cuya órbita estuviera determinada con precisión.

La armada estadounidense rápidamente aplicó esta tecnología, para proveer a los sistemas de navegación de sus flotas de observaciones de posiciones actualizadas y precisas. Así surgió el sistema TRANSIT, que quedó operativo en 1964, y hacia 1967 estuvo disponible, además, para uso comercial.

Las actualizaciones de posición, en ese entonces, se encontraban disponibles cada 40 minutos y el observador debía permanecer casi estático para poder obtener información adecuada.

Posteriormente, en esa misma década y gracias al desarrollo de los relojes atómicos, se diseñó una constelación de satélites, portando cada uno de ellos uno de estos relojes y estando todos sincronizados con base en una referencia de tiempo determinada.

Fig. 2.3 Operadora de satélites controlando la constelación NAVSTAR-GPS, en la Base Aérea de Schriever

En 1973 se combinaron los programas de la Armada y el de la Fuerza Aérea de los Estados Unidos (este último consistente en una técnica de transmisión codificada que proveía datos precisos usando una señal modulada con un código de ruido pseudo-aleatorio (PRN = Pseudo-Random Noise), en lo que se conoció como Navigation Technology Program, posteriormente renombrado como NAVSTAR GPS.

Entre 1978 y 1985 se desarrollaron y lanzaron once satélites prototipo experimentales NAVSTAR, a los que siguieron otras generaciones de satélites, hasta completar la constelación actual, a la que se declaró con «capacidad operacional inicial» en diciembre de 1993 y con «capacidad operacional total» en abril de 1995.

En 1994, este país ofreció el servicio normalizado de determinación de la posición para apoyar las necesidades de la OACI, y ésta aceptó el ofrecimiento.

2.2. CARACTERÍSTICAS TÉCNICAS Y PRESTACIONES

El Sistema Global de Navegación por Satélite lo componen:

Sistema de satélites: Está formado por 24 unidades con trayectorias sincronizadas para cubrir toda la superficie del globo terráqueo. Más concretamente, repartidos en 6 planos orbitales de 4 satélites cada uno. La energía eléctrica que requieren para su funcionamiento la adquieren a partir de dos paneles compuestos de celdas solares adosados a sus costados.

Estaciones terrestres: Envían información de control a los satélites para controlar las órbitas y realizar el mantenimiento de toda la constelación.

Terminales receptores: Indican la posición en la que están; conocidas también como Unidades GPS, son las que podemos adquirir en las tiendas especializadas.

2.3. Segmento espacial

Satélites en la constelación: 24 (4 x 6 órbitas)

Altitud: 20.200 km

Período: 11 h 56 min (12 horas sidéreas)

Inclinación: 55 grados (respecto al ecuador terrestre).

Vida útil: 7,5 años

Segmento de control (estaciones terrestres)

Estación principal: 1

Antena de tierra: 4

Estación monitora (de seguimiento): 5

Señal RF

Frecuencia portadora:

Civil - 1575,42 MHz (L1). Utiliza el Código de Adquisición Aproximativa (C/A).

Militar – 1227,60 MHz (L2). Utiliza el Código de Precisión (P), cifrado.

Nivel de potencia de la señal: -160 dBW (en superficie tierra).

Polarización: circular dextrógira.

Exactitud

Posición: aproximadamente 15 m (el 95%)

Hora: 1 ns

Cobertura: mundial

Capacidad de usuarios: ilimitada

Sistema de coordenadas:

Sistema Geodésico Mundial 1984 (WGS84).

Centrado en la Tierra, fijo.

Integridad: tiempo de notificación 15 minutos o mayor.

Disponibilidad: 24 satélites - 70% y 21 satélites - 98%.

Fig. 2.4 Estación y receptor GPS profesionales para precisiones centimétricas

2.4. Evolución del sistema GPS

El GPS está evolucionando hacia un sistema más sólido (GPS III), con una mayor disponibilidad y que reduzca la complejidad de las aumentaciones GPS. Algunas de las mejoras previstas comprenden:

Incorporación de una nueva señal en L2 para uso civil.

Adición de una tercera señal civil (L5): 1176,45 MHz

Protección y disponibilidad de una de las dos nuevas señales para servicios de Seguridad Para la Vida (SOL).

Mejora en la estructura de señales.

Incremento en la potencia de señal (L5 tendrá un nivel de potencia de -154 dB).

Mejora en la precisión (1 – 5 m).

Aumento en el número de estaciones monitorizadas: 12 (el doble)

Fig. 2.5 Receptor GPS

Permitir mejor interoperabilidad con la frecuencia L1 de Galileo

El programa GPS III persigue el objetivo de garantizar que el GPS satisfará requisitos militares y civiles previstos para los próximos 30 años. Este programa se está desarrollando para utilizar un enfoque en 3 etapas (una de las etapas de transición es el GPS II); muy flexible, permite cambios futuros y reduce riesgos. El desarrollo de satélites GPS II comenzó en 2005, y el primero de ellos estará disponible para su lanzamiento en 2012,con el objetivo de lograr la transición completa de GPS III en 2017.

Los desafíos son los siguientes:

Representar los requisitos de usuarios, tanto civiles como militares, en cuanto a GPS.

Limitar los requisitos GPS III dentro de los objetivos operacionales.

Proporcionar flexibilidad que permita cambios futuros para satisfacer requisitos de los usuarios hasta 2030.

Proporcionar solidez para la creciente dependencia en la determinación de posición y de hora precisa como servicio internacional.

2.5. Funcionamiento

La situación de los satélites es conocida por el receptor con base en las efemérides (5 elementos orbitales), parámetros que son transmitidos por los propios satélites. La colección de efemérides de toda la constelación se completa cada 12 minutos y se guarda en el receptor GPS.

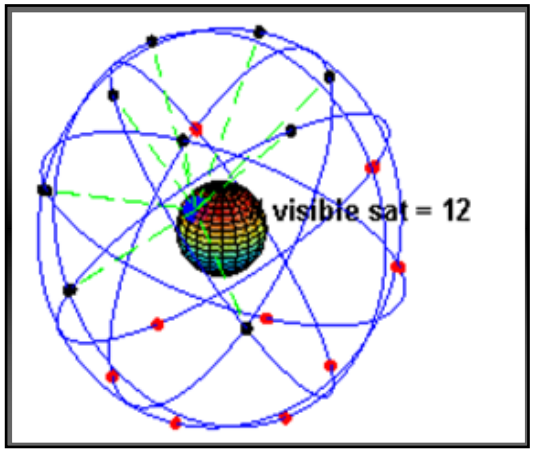

Fig. 2.6 Un ejemplo visual de la constelación GPS en conjunción con la rotación de la Tierra. Obsérvese como el número de satélites visibles en un determinado punto de la superficie de la Tierra, en este ejemplo a 45° N, cambia con el tiempo

El receptor GPS funciona midiendo su distancia a los satélites, y usa esa información para calcular su posición. Esta distancia se mide calculando el tiempo que la señal tarda en llegar al receptor. Conocido ese tiempo y basándose en el hecho de que la señal viaja a la velocidad de la luz (salvo algunas correcciones que se aplican), se puede calcular la distancia entre el receptor y el satélite.

Cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en el propio satélite y de radio la distancia total hasta el receptor. Obteniendo información de dos satélites se nos indica que el receptor se encuentra sobre la circunferencia que resulta cuando se interceptan las dos esferas.

Si adquirimos la misma información de un tercer satélite notamos que la nueva esfera solo corta la circunferencia anterior en dos puntos. Uno de ellos se puede descartar porque ofrece una posición absurda. De esta manera ya tendríamos la posición en 3-D. Sin embargo, dado que el reloj que incorporan los receptores GPS no está sincronizado con los relojes atómicos de los satélites GPS, los dos puntos determinados no son precisos.

Teniendo información de un cuarto satélite, eliminamos el inconveniente de la falta de sincronización entre los relojes de los receptores GPS y los relojes de los satélites. Y es en este momento cuando el receptor GPS puede determinar una posición 3-D exacta (latitud, longitud y altitud). Al no estar sincronizados los relojes entre el receptor y los satélites, la intersección de las cuatro esferas con centro en estos satélites es un pequeño volumen en vez de ser un punto. La corrección consiste en ajustar la hora del receptor de tal forma que este volumen se transforme en un punto.

2.6. Fiabilidad de los datos

Debido al carácter militar del sistema GPS, el Departamento de Defensa de los EE. UU. se reservaba la posibilidad de incluir un cierto grado de error aleatorio, que podía variar de los 15 a los 100 m. La llamada **disponibilidad selectiva** (S/A) fue eliminada el 2 de mayo de 2000. Aunque actualmente no aplique tal error inducido, la precisión intrínseca del sistema GPS depende del número de satélites visibles en un momento y posición determinados.

Con un elevado número de satélites siendo captados (7, 8 ó 9 satélites), y si éstos tienen una geometría adecuada (están dispersos), pueden obtenerse precisiones inferiores a 2,5 metros en el 95% del tiempo. Si se activa el sistema DGPS llamado SBS (WAAS-EGNOS-MSAS), la precisión mejora siendo inferior a un metro en el 97% de los casos. (Estos sistemas SBS no aplican en Sudamérica, ya que esta parte del mundo no cuenta con este tipo de satélites geoestacionarios).

2.7. Fuentes de error

La posición calculada por un receptor GPS requiere el instante actual, la posición del satélite y el retraso metido de la señal recibido. La precisión es dependiente en la posición y el retraso de la señal.

Al introducir el atraso, el receptor compara una serie de bits (unidad binaria) recibida del satélite con una versión interna. Cuando se comparan los límites de la serie, las electrónicas pueden meter la diferencia a 1% de un tiempo BIT, o aproximadamente 10 nanosegundos por el código C/A. Desde entonces las señales GPS se propagan a la velocidad de luz, que representa un error de 3 metros. Este es el error mínimo posible usando solamente la señal GPS C/A.

La precisión de la posición se mejora con una señal P(Y). Al presumir la misma precisión de 1% de tiempo BIT, la señal P(Y) (alta frecuencia) resulta en una precisión de más o menos 30 centímetros. Los errores en las electrónicas son una de las varias razones que perjudican la precisión.

2.7.1. Fuentes de Errores de posicionamiento

Fuente	Efecto
Ionosfera	± 5 m
Efemérides	± 2,5 m
Reloj satelital	± 2 m
Distorsión multibandas	± 1 m
Troposfera	± 0,5 m
Errores numéricos	± 1 m o menos

Tabla 2.1 Fuentes de Errores de posicionamiento

Retraso de la señal en la ionosfera y la troposfera.

Señal multirruta, producida por el rebote de la señal en edificios y montañas cercanos.

Errores de orbitales, donde los datos de la órbita del satélite no son completamente precisos.

Número de satélites visibles.

Geometría de los satélites visibles.

Errores locales en el reloj del GPS.

2.8. GPS DIFERENCIAL

El DGPS (Differential GPS), o GPS diferencial, es un sistema que proporciona a los receptores de GPS correcciones de los datos recibidos de los satélites GPS, con el fin de proporcionar una mayor precisión en la posición calculada.

Se concibió fundamentalmente debido la introducción de la disponibilidad selectiva (SA).El fundamento radica en el hecho de que los errores producidos por el sistema

GPS afectan por igual (o de forma muy similar) a los receptores situados próximos entre sí. Los errores están fuertemente correlacionados en los receptores próximos.

Fig. 2.7 Receptor GPS Navcom SF-2040G StarFire montado sobre un mástil

Un receptor GPS fijo en tierra (referencia) que conoce exactamente su posición basándose en otras técnicas, recibe la posición dada por el sistema GPS, y puede calcular los errores producidos por el sistema GPS, comparándola con la suya, conocida de antemano. Este receptor transmite la corrección de errores a los receptores próximos a él, y así estos pueden, a su vez, corregir también los errores producidos por el sistema dentro del área de cobertura de transmisión de señales del equipo GPS de referencia.

En suma, la estructura DGPS quedaría de la siguiente manera:

Estación monitorizada (referencia), que conoce su posición con una precisión muy alta.

Esta estación está compuesta por:

Un receptor GPS.

Un microprocesador, para calcular los errores del sistema GPS y para generar la estructura del mensaje que se envía a los receptores.

Transmisor, para establecer un enlace de datos unidireccional hacia los receptores de los usuarios finales.

Fig. 2.8 Estación de referencia DGPS

Equipo de usuario, compuesto por un receptor DGPS (GPS + receptor del enlace de datos desde la estación monitorizada).

Existen varias formas de obtener las correcciones DGPS. Las más usadas son:

Recibidas por radio, a través de algún canal preparado para ello, como el RDS en una emisora de FM.

Descargadas de Internet, o con una conexión inalámbrica.

Proporcionadas por algún sistema de satélites diseñado para tal efecto. En Estados Unidos existe el WAAS, en Europa el EGNOS y en Japón el MSAS, todos compatibles entre sí.

En los mensajes que se envían a los receptores próximos se pueden incluir dos tipos de correcciones:

Una corrección directamente aplicada a la posición. Esto tiene el inconveniente de que tanto el usuario como la estación monitora deberán emplear los mismos satélites, pues las correcciones se basan en esos mismos satélites.

Fig. 2.9 Vehículo de la empresa Tele Atlas con GPS cartografiando y fotografiando las carreteras en Rochester, Nueva York (EE.UU.)

Una corrección aplicada a las pseudodistancias de cada uno de los satélites visibles. En este caso el usuario podrá hacer la corrección con los 4 satélites de mejor relación señal-ruido (S/N). Esta corrección es más flexible.

El error producido por la disponibilidad selectiva (SA) varía incluso más rápido que la velocidad de transmisión de los datos. Por ello, junto con el mensaje que se envía de correcciones, también se envía el tiempo de validez de las correcciones y sus tendencias. Por tanto, el receptor deberá hacer algún tipo de interpolación para corregir los errores producidos.

Si se deseara incrementar el área de cobertura de correcciones DGPS y, al mismo tiempo, minimizar el número de receptores de referencia fijos, será necesario modelar las variaciones espaciales y temporales de los errores. En tal caso estaríamos hablando del GPS diferencial de área amplia.

Con el DGPS se pueden corregir en parte los errores debidos a:

Disponibilidad selectiva (eliminada a partir del año 2000).

Propagación por la ionosfera - troposfera.

Errores en la posición del satélite (efemérides).

Errores producidos por problemas en el reloj del satélite.

Para que las correcciones DGPS sean válidas, el receptor tiene que estar relativamente cerca de alguna estación DGPS; generalmente, a menos de 1.000 km. La precisión lograda puede ser de unos dos metros en latitud y longitud, y unos 3 m en altitud.

2.9. Integración con telefonía móvil

Algunos celulares pueden vincularse a un receptor GPS diseñado a tal efecto. Suelen ser módulos independientes del teléfono que se comunican vía inalámbrica bluetooth, o implementados en el mismo terminal móvil, y que le proporcionan los datos de posicionamiento que son interpretados por un programa de navegación. Esta aplicación del GPS está particularmente extendida en los teléfonos móviles que operan con el sistema operativo Symbian OS, y PDAs con el sistema operativo Windows Mobile, aunque varias marcas han lanzado modelos con un módulo GPS integrado con software GNU/Linux.

2.10. Aplicaciones

2.10.1. Civiles

Navegación terrestre (y peatonal), marítima y aérea. Bastantes automóviles lo incorporan en la actualidad, siendo de especial utilidad para encontrar direcciones o indicar la situación a la grúa.

Teléfonos móviles

Topografía y geodesia.

Localización agrícola (agricultura de precisión), ganadera y de fauna.

Salvamento y rescate.

Deporte, acampada y ocio.

Para localización de enfermos, discapacitados y menores.

Aplicaciones científicas en trabajos de campo.

Geocaching, actividad deportiva consistente en buscar "tesoros" escondidos por otros usuarios.

Se utiliza para rastreo y recuperación de vehículos.

Navegación Deportiva.

Deportes Aéreos: Parapente, Ala delta, Planeadores, etc.

Existe quien dibuja usando tracks o juega utilizando el movimiento como cursor (común en los GPS garmin).

Fig. 2.10 Navegador GPS de pantalla táctil de un vehículo con información sobre la ruta, así como las distancias y tiempos de llegada al punto de destino

2.10.2. Militares

Navegación terrestre, aérea y marítima.

Guiado de misiles y proyectiles de diverso tipo.

Búsqueda y rescate.

Reconocimiento y cartografía.

Detección de detonaciones nucleares.

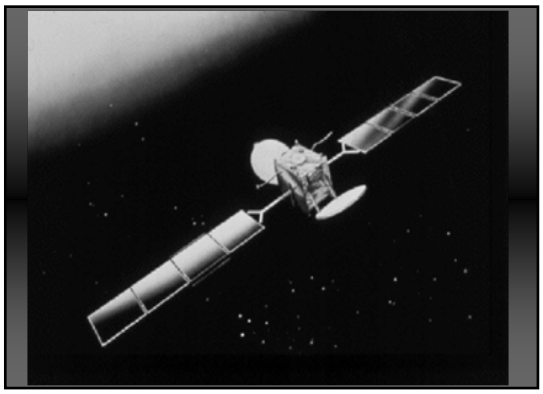

Fig. 2.11 Satélite Inmarsat-3

2.11. EGNOS

El sistema **EGNOS** (*European Geostationary Navigation Overlay Service*) es un Sistema de Aumentación Basado en Satélites desarrollado por la Agencia Espacial Europea (ESA), la

Comisión Europea (institución de la Unión Europea) y Eurocontrol. Está ideado como un complemento para las redes GPS y GLONASS para proporcionar una mayor precisión y seguridad en las señales, permitiendo una precisión inferior a dos metros.

Consiste en una red de tres satélites geoestacionarios y en una red de estaciones terrestres encargadas de monitorizar los errores en las señales de GPS y actualizar los mensajes de corrección enviados por EGNOS.

El sistema empezó a emitir de formal operacional (*initial operation phase*) en julio de 2005 mostrando unas prestaciones excelentes en términos de precisión y disponibilidad. El inicio oficial de operaciones fue anunciado por la Comisión Europea para el 1 de octubre de 2009. El sistema debería ser cualificado para su uso en aplicaciones de seguridad (*safety of life*) en el año 2010 por la Agencia de supervisión GNSS (*GNSS Supervisory Agency*).

El sistema EGNOS es completamente compatible con el sistema de Estados Unidos llamado WAAS, operativo desde el año 2003. También existe otro igual en Japón llamado MSAS, que debería empezar a operar en el año 2007, y la Agencia India del Espacio (ISRO) está actualmente desarrollando el sistema GAGAN.

2.11.1. ELEMENTOS DEL SISTEMA EGNOS

Fig. 2.12 Sector terrestre de EGNOS

El sistema EGNOS está compuesto por 34 Estaciones de Referencia y de Supervisión de Integridad (RIMS) desplegadas para supervisar los satélites de las constelaciones

GNSS. Cada satélite tiene que ser supervisado por múltiples RIMS antes de que se generen las correcciones y los mensajes de integridad.Existen cuatro Centros de Control de Misión (MCC), que procesan los datos de las RIMS para generar las

Correcciones WAD (*Wide Aérea Differential*) y mensajes de integridad para cada satélite. Solo uno de estos MCCs está activo y operacional, los otros MCCs permanecen como "reserva caliente" que pueden activarse si ocurre algún problema.

Las Estaciones Terrestres de Navegación (*Navigation Land Earth Stations*, NLES) transmiten los mensajes de corrección e integridad desde el MCC a los satélites

geoestacionarios, encargados de radiar finalmente la señal SBAS a los usuarios finales. El sistema desplegará dos NLES (una principal y una de reserva), y una tercera NLES con fines de pruebas y validación.

El sector espacial de EGNOS está compuesto por tres satélites geoestacionarios con cobertura global terrestre: dos satélites Inmarsat-3 (AOR-E e IOR), así como el satélite ESA Artemis. Los usuarios EGNOS pueden rastrear dos satélites geoestacionarios por lo menos.

Capítulo 3

GSM Y SUS CARACTERISTICAS

GSM significa Global System for Mobile Communications, lo que en español es Sistema Global para Comunicaciones Globales. Ya en el nombre está implícita la clara intención de comunicar a las personas incluso en distintos países, crear algo mundial.

Lo que se pretende lograr con la tecnología GSM es una especie de roaming internacional, algo más global, que no sólo abarque un país o ciertas zonas específicas del mismo. Es algo así como tener el mismo número para más de 150 países ya que es una tecnología satelital. A pesar de que empezó a desarrollarse desde hace más de 10 años, hasta ahora es que está empezando a ser utilizada en todo el mundo.

Esta es una breve reseña de la evolución de la comunicación inalámbrica, la cual involucra las tecnologías GSM, GPRS, EDGE y 3GSM.

3.1. TECNOLOGÍA GSM:

La plataforma actual de GSM es un suceso tecnológico y un logro sin precedentes en la historia mundial. En menos de 10 años desde que se instalo la primera red GSM, esta tecnología se convirtió en el estándar, mostrando los más altos índices de crecimiento y de penetración en el mercado, pues ya se encuentra disponible en más de 200 países.

Actualmente, la tecnología GSM es utilizada por 1/6 de la población mundial y se estima que cuenta con más de 1 billón de usuarios, acreditando un crecimiento de 160 millones de usuarios en los últimos 12 meses.

Fig. 3.1 Comunicaciones GSM

3.2. LA TECNOLOGÍA GPRS:

La tecnología GPRS (General Packet Radio Service) ofrece una conexión siempre activa a los servicios en red y una mayor capacidad de manejo de transmisión de d

Fig. 3.2 Comunicaciones GPRS

atos. Esta evolución permite navegación en Internet a color, servicio de email, MMS y servicios de acuerdo a la cobertura y equipo que tenga el usuario.

3.3. TECNOLOGÍA EDGE:

Llega como una evolución de GPRS, manejando la transmisión de datos por el núcleo de la red GSM, lo cual permitirá el uso de servicios avanzados como la descarga de videos, música, MMS con videos, navegación por Internet visualizadas a todo color, y obviamente servicio de email.

EDGE ofrece servicios de navegación similares a los de banda ancha, lo cual le permitirá a los usuarios conectarse a Internet, enviar y recibir información, archivos, páginas web a una velocidad 3 veces mayores a la obtenida con una red GSM convencional. Por otro lado, a los operadores les brinda la posibilidad de ofrecer mayores velocidades de navegación en los móviles, atender a un mayor número de clientes que utilicen este tipo de servicios, y liberar espectro de su banda para dedicarlo a las llamadas de voz.

Fig. 3.3 Comunicaciones EDGE

3.4. TECNOLOGIA 3GSM:

3GSM es la última adición a la familia GSM. 3GSM busca ofrecer servicios de tercera generación a nivel mundial. La tecnología en la cual los servicios 3GSM serán ofrecidos esta soportada en la tecnología GSM con una interface Wideband-CDMA (CDMA de banda ancha), la cual está siendo desarrollada como un estándar abierto a los operadores.

Actualmente, más del 85% de los operadores mundiales han escogido la plataforma 3GSM.

Fig. 3.4 Comunicaciones 3GSM

3.5. PROTOCOLO DE COMUNICACIÓN ENTRE PC Y DISPOSITIVO MÓVIL.

El protocolo que se va a utilizar es el sistema binario, ya que el módem que desarrollaremos es un sistema digital, el cual se comunicara por medio de impulsos eléctricos de 1's y 0's provenientes del puerto USB del PC, y de igual manera se receptaran los datos en el PC desde el módem, para que luego estos sena interpretados y procesados por la aplicación del Sistema. Este protocolo nos servirá

para envío y recepción de información; para esto utilizaremos diagramas esquemáticos de equipos celulares de prueba.

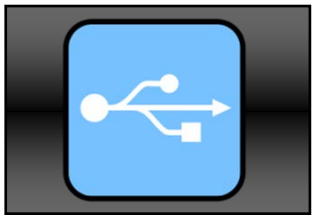

Fig. 3.5 Símbolo de USB

3.5.1. PUERTO USB

El **Universal Serial Bus** (bus universal en serie) es un puerto que sirve para conectar periféricos a una computadora. Fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

Fig. 3.6 Conector USB tipo A

El estándar incluye la transmisión de energía eléctrica al dispositivo conectado. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios sin necesitar fuentes de alimentación extra. La gran mayoría de los concentradores incluyen fuentes de alimentación que brindan energía a los dispositivos conectados a ellos, pero algunos dispositivos consumen tanta energía

que necesitan su propia fuente de alimentación. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites).

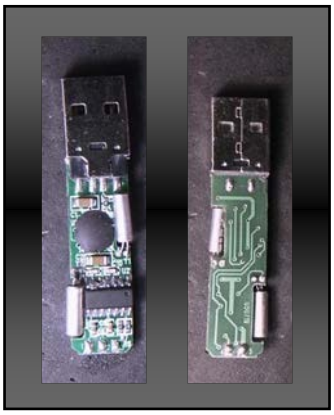

Fig. 3.7 Dispositivos USB

El diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades plug-and-play permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar. Cuando se conecta un nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar.

El USB puede conectar los periféricos como mouse, teclados, escáneres, cámaras digitales, teléfonos celulares, reproductores multimedia, impresoras, discos duros externos, tarjetas de sonido, sistemas de adquisición de datos y componentes de red. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha empezado a desplazar a los puertos paralelos porque el

USB hace sencillo el poder agregar más de una impresora a una computadora personal.

Fig. 3.8 Tarjeta PCI-USB 2.0

En el caso de los discos duros, el USB es poco probable que reemplace completamente a los buses como el ATA (IDE) y el SCSI porque el USB tiene un rendimiento un poco más lento que esos otros estándares. El nuevo estándar Serial ATA permite tasas de transferencia de hasta aproximadamente 150/300 MB por segundo. Sin embargo, el USB tiene una importante ventaja en su habilidad de poder instalar y desinstalar dispositivos sin tener que abrir el sistema, lo cual es útil para dispositivos de almacenamiento externo. Hoy en día, una gran parte de los fabricantes ofrece dispositivos USB portátiles que ofrecen un rendimiento casi indistinguible en comparación con los ATA (IDE).

El USB no ha remplazado completamente a los teclados AT y mouse PS/2, pero virtualmente todas las placas base de PC traen uno o más puertos USB.

3.6. CARACTERÍSTICAS DE TRANSMISIÓN

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

Baja Velocidad (1.0): Bitrate de 1.5Mbit/s (192KB/s). Utilizado en su mayor parte por Dispositivos de Interfaz Humana (HID) como los teclados, los ratones y los joysticks.

Velocidad Completa (1.1): Bitrate de 12Mbit/s (1.5MB/s). Esta fue la más rápida antes de que se especificara la USB 2.0 y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos, dividen el ancho de banda de la conexión USB entre ellos basados en un algoritmo FIFO (First In First Out).
Alta Velocidad (2.0): Bitrate de 480Mbit/s (60MB/s).

Súper Velocidad (3.0): Actualmente en fase experimental. Bitrate de 4.8Gbit/s (600MB/s). Las velocidades de los buses serán 10 veces más rápidas que la de USB 2.0 debido a la inclusión de un enlace de fibra óptica que trabaja con los conectores tradicionales de cobre.

Las señales del USB son transmitidas en un cable de data par trenzado con impedancia de 90Ω ±15% llamados D+ y D-. Éstos, colectivamente utilizan señalización diferencial en half dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- usualmente operan en conjunto y no son conexiones simplex. Los niveles de transmisión de la señal varían de 0 a 0.3V para bajos (ceros) y de 2.8 a 3.6V para altos (unos) en las versiones 1.0 y 1.1, y en ±400mV en Alta Velocidad (2.0). En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una

terminación de 45 Ω a tierra o un diferencial de 90 Ω para acoplar la impedancia del cable. Este puerto sólo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100mA por cada puerto, pero en caso que estuviese conectado un dispositivo que permite 4 puertos por cada salida USB (extensiones de máximo 4 puertos) entonces la energía del USB se asigna en unidades de 100mA hasta un máximo de 500mA por puerto.

Pin	Nombre	Color del Cable	Descripción
1	VCC	Rojo	+5V
2	D−	Blanco	Data −
3	D+	Verde	Data +
4	GND		Tierra

Tabla 3.1 Descripcion de los Pines del USB

3.7. COMPATIBILIDAD Y CONECTORES

El estándar USB especifica tolerancias para impedancia y de especificaciones mecánicas relativamente bajas para sus conectores, intentando minimizar la incompatibilidad entre los conectores fabricados por distintas compañías. Una meta a la que se ha logrado llegar. El estándar USB, a diferencia de otros estándares también

define tamaños para el área alrededor del conector de un dispositivo, para evitar el bloqueo de un puerto adyacente por el dispositivo en cuestión.

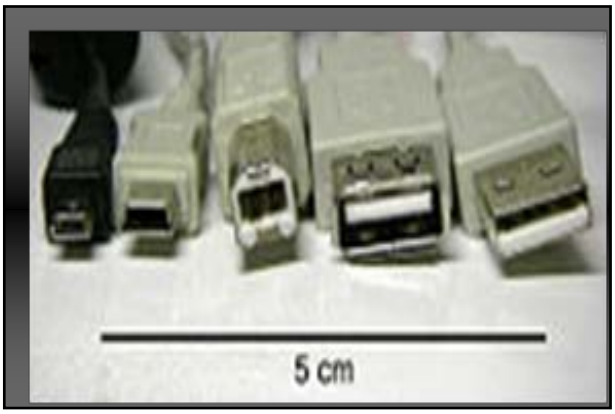

Fig. 3.9 Tipos diferentes de Conectores USB: Micro USB, Mini USB, Tipo B, Hembra tipo A, Tipo A

Las especificaciones USB 1.0, 1.1 y 2.0 definen dos tipos de conectores para conectar dispositivos al servidor: A y B. Sin embargo, la capa mecánica ha cambiado en algunos conectores. Por ejemplo, el IBM UltraPort es un conector USB privado localizado en la parte superior del LCD de las computadoras portátiles de IBM. Utiliza un conector mecánico diferente mientras mantiene las señales y protocolos característicos del USB. Otros fabricantes de artículos pequeños han desarrollado también sus medios de conexión pequeños, y ha aparecido una gran variedad de ellos, algunos de baja calidad.

Una extensión del USB llamada USB-On-The-Go (*sobre la marcha*) permite a un puerto actuar como servidor o como dispositivo esto se determina por qué lado del cable está conectado al aparato. Incluso después de que el cable está conectado y las

unidades se están comunicando, las 2 unidades pueden "cambiar de papel" bajo el control de un programa.

Esta facilidad está específicamente diseñada para dispositivos como PDA, donde el enlace USB podría conectarse a un PC como un dispositivo, y conectarse como servidor a un teclado o ratón. El USB-On-The-Go también ha diseñado 2 conectores pequeños, el mini-A y el mini-B, así que esto debería detener la proliferación de conectores miniaturizados de entrada.

3.8. EQUIPO CELULAR GSM

El equipo celular a utilizar en la recepción de datos en el computador es el Nokia 3220, escogido por ser uno de los más económicos que responden a los comandos AT y utilizan cable de datos USB, en nuestro caso es el USB CA-42.

Fig. 3.10 Pines de conexión pop-port Nokia 3220

Cabe recalcar como ya es de conocimiento de la mayoría de los usuarios de teléfonos celulares Nokia el modelo Nokia 3220 es en base similar al modelo Nokia 6020, con las únicas diferencias en el diseño de su presentación además que en el modelo Nokia 6020 viene incluido un dispositivo de comunicación infrarrojo el cual en nuestros días se encuentra en estado de desuso, puesto que la tecnología Bluetooth está ganando gran cantidad de terreno.

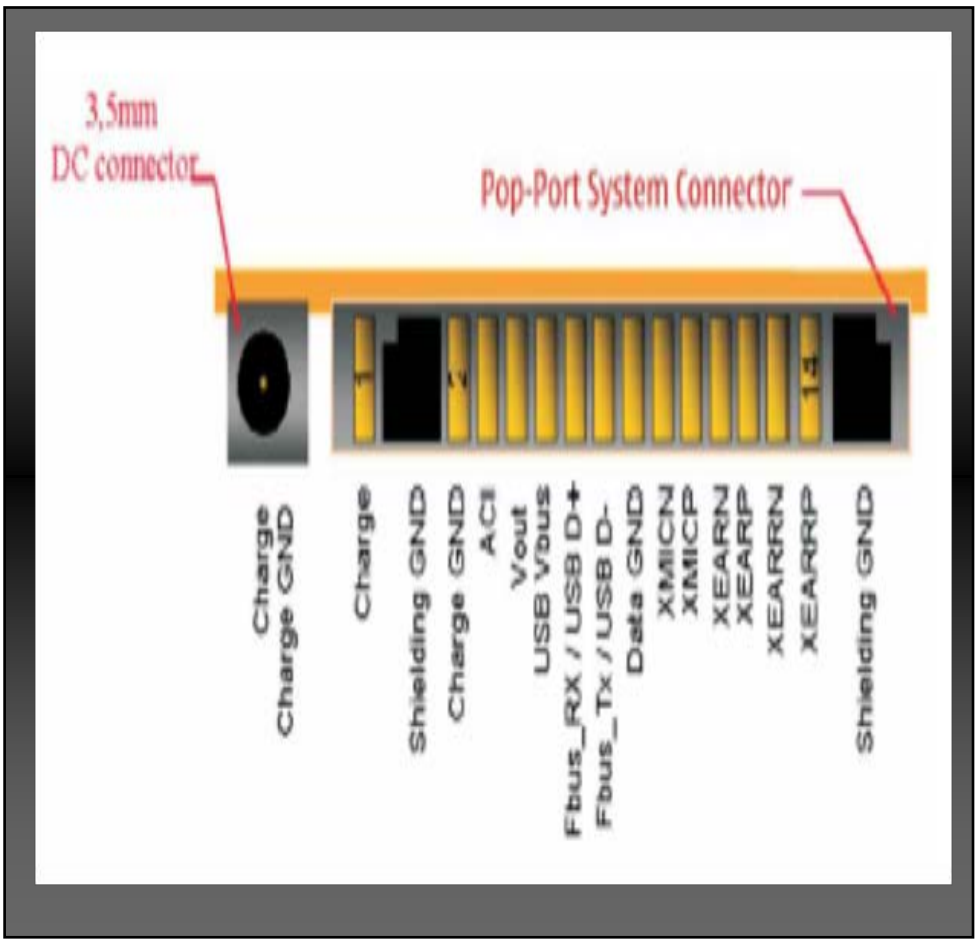

Fig. 3.11 14 pines pop-port utilizados en el Nokia 3220

Lo más importante entre estos dos modelos de telefonía celular es el orden de sus pines de conexión son similares, siendo posible el poder utilizar el mismo cable de comunicación USB CA-42.

Pin	Nombre	Descripción
1	Vin	Entrada de carga
2	GND	Tierra de carga
3	ACI	Interfaz de control de accesorios
4	V Out	Conectado al pin 3 en cable USB de datos CA-42
5	USB Vbus	Debe ser conectado al pin 1 cable USB de datos
6	FBus Rx/USB D+	Debe ser conectado al pin 2 cable USB de datos
7	FBus Tx/USB D-	Debe ser conectado al pin 3 cable USB de datos
8	GND	Tierra de datos
9	X Mic-	Entrada externa de audio – entrada negativa de micrófono
10	X Mic+	Entrada externa de audio – entrada positiva de micrófono
11	HS Ear L-	salida externa de audio – salida negativa lado izquierdo
12	HS Ear L+	salida externa de audio – salida positivo lado izquierdo
13	HS Ear R-	salida externa de audio – salida negativa lado derecho
14	HS Ear R+	salida externa de audio – salida positivo lado derecho
	GND	Tierra del shield

Tabla 3.2 Pines de Salida del Equipo Celular

3.9. CABLE NOKIA CA – 42

Este cable el cual usaremos para el modulo de recepción de datos, que se encontrara ubicado en el PC monitor de las acciones de nuestra Tesis.

Para el uso adecuado de este dispositivo USB, se debe instalar el Connectivity Cable Drivers para CA-42, el cual viene en un disco de instalación que adjunta el dispositivo al adquirirlo.

Fig. 3.12 Cable Adaptador de Conectividad Nokia CA-42

3.10. REQUISITOS

Para instalar Nokia Connectivity Cable Drivers en su ordenador deberá tener:

• Un mínimo de 30 MB de espacio libre en el disco

• Uno de los siguientes sistemas operativos:

• Windows 2000 con Microsoft SP (Service Pack) 3 o SP 4

• Windows XP con Microsoft SP 1 o SP 2 (Professional o Home Edition, no Media Center Edition)

• Un puerto USB libre para conectar el cable.

3.11. INSTALACIÓN DE NOKIA CONNECTIVITY CABLE DRIVERS ANTES DE LA INSTALACIÓN

Si en su PC ya ha instalado antes Nokia Connectivity Cable Drivers, deberá desinstalarlo manualmente. Para hacerlo:

1. Haga clic en **Inicio > (Configuración) > Panel de control** para abrir el Panel de control.

2. Haga doble clic en **Agregar o quitar programas**. Si no encuentra Nokia Connectivity Cable Driver o DKU-2 Cable Driver en la lista, significa que no tiene ninguna versión instalada y puede seguir con la instalación de Nokia Connectivity Cable Drivers.

3. Seleccione Nokia Connectivity Cable Driver o DKU-2 Cable Driver de la lista y haga clic en **Cambiar o quitar**.

Resolución de problemas

La conexión no funciona

Para volver a establecer la conexión, siga los siguientes pasos:

• Asegúrese de que el teléfono es compatible con el cable: consulte la guía del usuario del teléfono o la página Web www.nokia.com/pcsuite > Features Supported by Your Phone (Funciones compatibles con su teléfono).

• Reinicie el teléfono y el PC y, a continuación, intente volver a establecer la conexión.

• Si tiene otros dispositivos USB conectados al PC, desconéctelos del PC e intente volver a establecer la conexión. Compruebe que no ha desconectado el ratón o el teclado.

• Asegúrese de que el tipo de conexión está activo en Nokia Connection Manager: **Inicio > (Configuración) Panel de control > Nokia Connection Manager.** Para CA-42 el tipo seleccionado debe ser “Cable serie” y para DKU-2, CA-53 y DKE-2 el tipo debe ser “USB”.

• Si utiliza un cable CA-42 asegúrese de que el puerto COM correcto está definido en Nokia Connection Manager: **Inicio > (Configuración) Panel de control > Nokia Connection Manager > Configurar…**

Si todos los pasos anteriores se han realizado sin problemas, compruebe que la instalación es correcta. Para hacerlo:

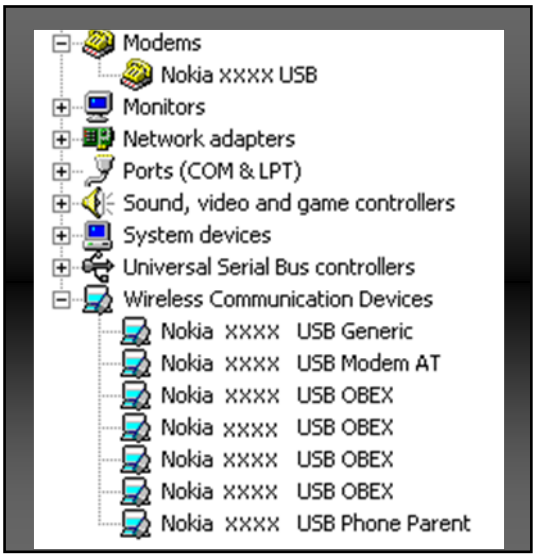

Fig. 3.13 Instalación correcta (los dígitos xxxx representan el número de modelo de cuatro cifras del teléfono)

1. Conecte el teléfono al PC con el cable y espere hasta que Windows haya encontrado todo el hardware nuevo.

2. Vaya a **Panel de control > Sistema > Hardware > Administrador de dispositivos...** (Si va a utilizar Windows XP, utilice la Vista clásica del Panel de control.)

3. Haga clic en **Wireless Communication Devices** (Dispositivos de comunicación inalámbrica). Si los dispositivos de Nokia están en la lista sin un signo de admiración junto al icono, la instalación se ha realizado correctamente. La lista debería incluir dispositivos como “Nokia CA-42 USB Phone Parent” o “Nokia xxxx USB Phone Parent” (los dígitos xxxx representan el número de modelo de cuatro cifras del teléfono). El número de dispositivos en la lista puede variar dependiendo del modelo de teléfono.

4. Haga clic en **Módems**. Si el “módem Nokia xxxx” (los dígitos xxxx representan el número de modelo de cuatro cifras del teléfono) está en la lista sin un signo de admiración junto al icono, la instalación se ha realizado correctamente.

Tenga en cuenta que el número de dispositivos en la lista Wireless Communication Devices (Dispositivos de comunicación inalámbrica) puede variar dependiendo del modelo de teléfono.

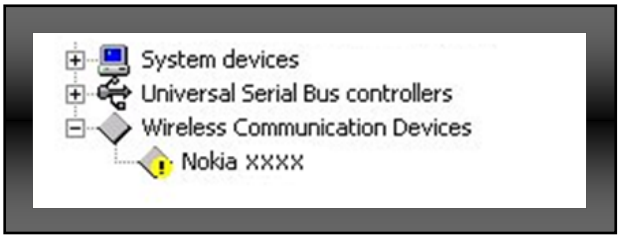

Fig. 3.14 Instalación incorrecta (los dígitos xxxx representan el número de modelo de cuatro cifras del teléfono)

5. Si los dispositivos Nokia no están en la lista o están pero con un signo de admiración, la instalación no se ha realizado correctamente. En este caso, desinstale los controladores, reinicie el ordenador y vuelva a instalarlos (consulte las instrucciones de instalación en este documento).

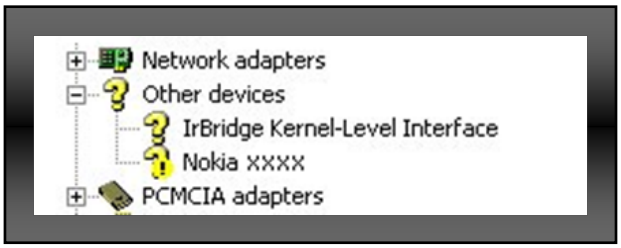

Fig. 3.15 Dispositivo desconocido (los dígitos xxxx representan el número de modelo de cuatro cifras del teléfono)

He conectado el cable durante/antes de instalar Nokia Connectivity Cable Drivers y ahora la conexión no funciona.

1. Conecte el teléfono al PC.

2. Vaya a **Panel de control** > **Sistema** > **Hardware** > **Administrador de dispositivos...**

3. Haga clic en **Wireless Communication Devices** (Dispositivos de comunicación inalámbrica).

4. Desinstale cualquier dispositivo desconocido (dispositivos con un signo de interrogación o admiración junto a él): haga clic con el botón derecho del ratón y haga clic en **Desinstalar**, o seleccione el dispositivo y pulse Supr en el teclado.

5. Asegúrese de que Nokia Connectivity Cable Drivers está instalado.

6. Desenchufe y vuelva a enchufar el cable.

7. Si esto no funciona, vuelva a instalar Nokia Connectivity Cable Drivers y vuelva a conectar el cable cuando finalice la instalación.

3.12. DIAGRAMA DE BLOQUE DEL SISTEMA MODULAR BÁSICO DE UN CELULAR

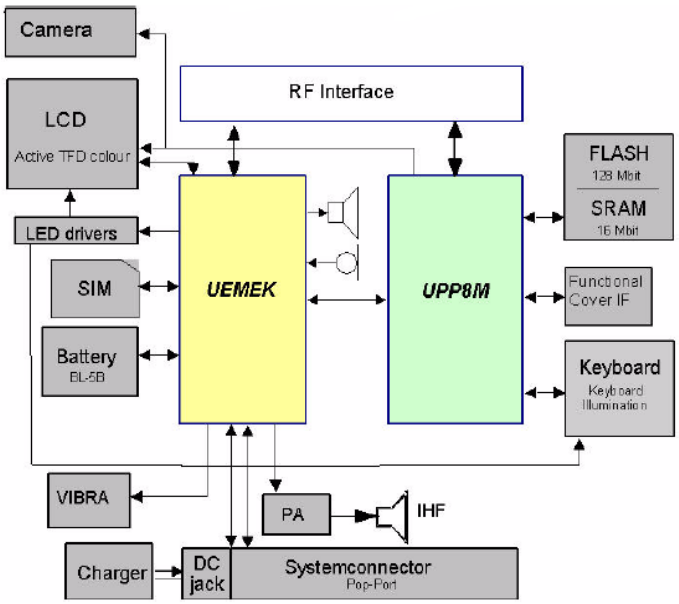

Fig. 3.16 Diagrama de Bloque del Sistema Modular Basico de un Celular

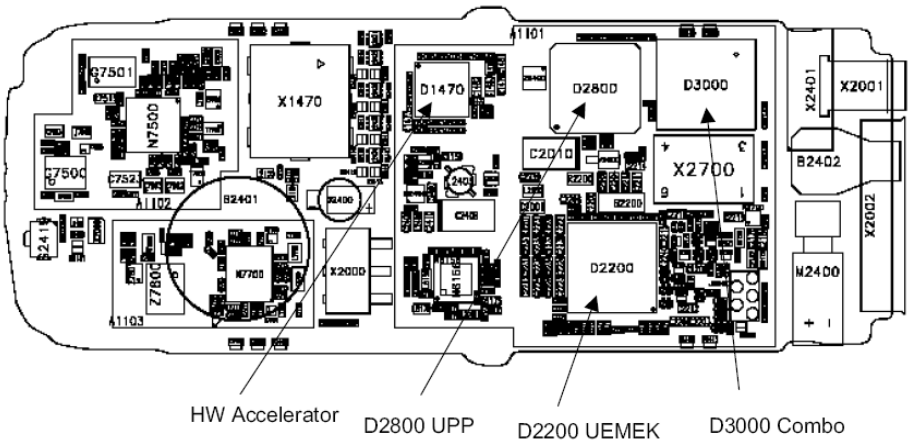

Fig. 3.17 Interior de Un Celular

Este diagrama consiste de una radio frecuencia y banda base de un celular.

3.13. Ubicación Componentes

UEMEK alimenta a la banda base y RF con una vía de poder construida con reguladores de voltaje, los cuales son conectados a la batería. La interfaz de RF usa principalmente 2.78V y la parte de banda base 1.8V I/o voltajes. El UPP es alimentado con un voltaje programado de 1.0V, 1.3V o 1.5V. UEMEK incluye 7 líneas reguladoras de LDO (Low Drop-Out) para la banda base y 7 reguladores para RF, este también incluye 4 fuentes de corriente para propósitos básicos y usos internos. El UEMEK además alimenta a la interfaz de SIM con voltajes programables de 1.8V o 3V.

El UPP opera a 26MHz de reloj del RF ASIC, la señal de reloj es dividida en dos por sistema nominal de reloj de 13 MHz. El DSP y el MCU contienen las PLLs, los cuales multiplican el sistema de reloj para elevar la frecuencia.

A tiempo real la función de reloj es integrada dentro del UEMEK, las cuales utilizan los mismos 32KHz de reloj que cuando el reloj esta en reposo.

El Dbus es controlado por el MCU y opera a la velocidad de 13 MHZ ambos procesadores están localizados en el UPP.

El UEMEK ASIC maneja la interfaz análoga entre la banda base y la sección RF. UEMEK provee conversiones A/D y D/A a la fase de entrada, al cuadrante de recepción y transmite paquetes de señal, también conversiones A/D y D/A de recepción y transmisión de señales de audio para y desde la interfaz de usuario. El UEMEK alimenta las señales análogas TXC y AFC de acuerdo a la sección RF de la

señal de control del UPP. Hay también señales separadas por el codificador de audio PDM. La velocidad digital es manejada por el DSP que está dentro del UPP ASIC.

UEMEK es un circuito dual de voltaje, las partes digitales son corridas desde la banda base alimentada por 1.8V y las partes análogas son corridas desde la alimentación análoga de 2.78V. (El vibrador, la cámara, los leds, reguladores) usan directamente el voltaje de la batería.

La banda base alimenta las entradas de micrófono tanto interno como externo y las salidas del parlante, los tonos del teclado, DMTF, y otros tonos de audio son generados y codificados por el UPP y los transmite al UEMEK para decodificarlos. Las señales externas de alerta vibratoria son generadas por el UEMEK con salidas separadas PWM.

3.14. ESPECIFICACIONES AMBIENTALES DE FUNCIONAMIENTO

Rangos Máximos Absolutos de Equipo Celular

Señal	Nota
Voltaje Batería	-0.3V….5.5V
Voltaje Batería (llamada)	Max 4.8V
Voltaje de entrada de carga	**-0.3V….16V**

Tabla 3.3 Voltajes de Funcionamiento de Equipo Celular

Condiciones de Temperatura

Condición	Min	Max
Operando en temperatura normal	-10º C	+55ª C
Funcionalidad reducida	-25ª C	+75ª C
Equipo almacenado	**-40ª C**	**+85ª C**

Tabla 3.4 Condiciones de Temparatura de Equipo Celular

Humedad y Resistencia al Agua

Condición	Min	Max
Humedad relativa	**5%**	**95%**

Tabla 3.5 Humedad y Resistenciasal Agua

Nota: Este modulo no está protegido contra los daños que pueda causar la humedad.

3.15. SMS

SMS son las siglas de *Servicio de Mensaje Corto*. Disponible en redes digitales GSM permitiendo enviar y recibir mensajes de texto de hasta 160 caracteres a teléfonos móviles GSM vía el centro de mensajes de un operador de red (como porta, movistar o alegro). También se pueden enviar mensajes cortos a través de Internet, usando un sitio web de SMS. Si el teléfono al que se envía el mensaje está apagado o fuera de cobertura el mensaje se almacena en la red y se entrega en cuanto el teléfono se conecta de nuevo a la red.

Uso de SMS en la actualidad, tabla comparativa con otras tecnologías:

	Relación Compañías a compañías				Relación Compañías a usuario			
Acciones	**Fax**	**e-mail**	**SMS**	**voz**	**fax**	**e-mail**	**SMS**	**voz**
Lanzamiento de productos	•	•	•	•		•	•	•
Comunicados de prensa	•	•						
Boletín informativo	•	•	•	•		•	•	•
Promoción con cupones de respuesta	•	•				•	•	•
Transacciones	•					•	•	•
Elaboración de informes	•	•				•		
Promoción de un sitio Web	•	•	•	•		•	•	•
Tarifa	•	•				•		
Notificación de facturas pendientes	•					•		
Extracto de cuentas	•	•	•	•			•	•
Ofertas comerciales	•	•	•	•		•	•	•
Cuestionarios	•	•	•	•		•	•	•

Tabla 3.6 Comparativas de Uso de Tecnologias de Comunicación

Existen dos tipos de diferentes de SMS:

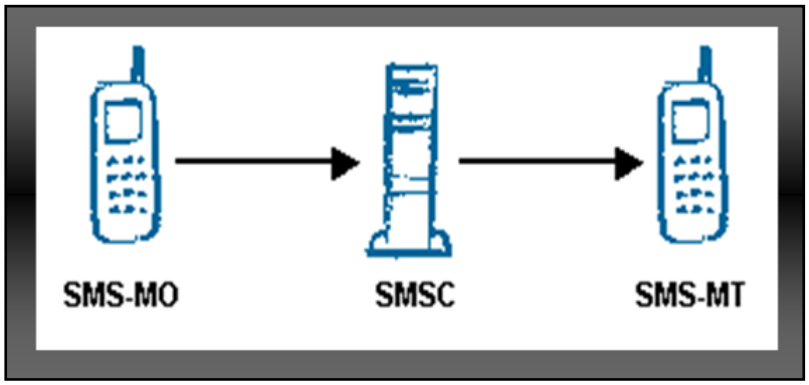

Fig. 3.18 Esquema de envió de un sms en sistema SMSC

SMS Point to Point (SMS/PP)

Permite enviar un mensaje de texto entre dos teléfonos GSM.

El envío más corriente de mensajes (SMS PP) desde un teléfono consiste en la transmisión del mensaje hasta un centro de mensajes, que es el que después realiza la entrega y la gestión de incidencias. Por ejemplo: si el móvil está apagado, guarda el mensaje e intenta entregarlo más tarde. Tal como se muestra en el esquema:

SMS Cell Broadcast (SMS/CB)

Permitirá enviar uno o más mensajes entre el teléfono emisor y teléfonos que estén en una zona de cobertura abarcada por un emisor de radio. Este emisor puede enlazar con otro emisor y los teléfonos de su zona, y así sucesivamente.

3.16. COMANDOS AT PARA GESTIÓN DE MENSAJES SMS

AT+CMGF: [Mensaje Formato]

1) AT+CMGF=?

- Informa de los formatos de mensaje soportados por el teléfono

- Sintaxis: AT+CMGF=? | +CMGF: (0,1)

modo = 0 indica formato de mensajes en modo PDU

modo = 1 indica formato de mensajes en modo TEXTO

2) AT+CMGF?

- Informa del formato de mensajes que está siendo utilizado actualmente para los comandos enviar, listar, leer y escribir.

- Sintaxis: AT+CMGF? | +CMGF: <modo>

modo = 0 indica formato de mensajes en modo PDU

modo = 1 indica formato de mensajes en modo TEXTO

3) AT+CMGF=<modo>

- Establace el formato a utilizar para la entrada y salida de mensajes.

- Sintaxis: AT+CMGF=<modo>

modo = 0 indica formato de mensajes en modo PDU

modo = 1 indica formato de mensajes en modo TEXTO

AT+CMGL: [Lista Mensajes]

1) AT+CMGL=?

- Informa de los posibles estados de un mensaje en la memoria que el teléfono puede soportar.

- Sintaxis: (+CMGF=0) AT+CMGL=? | +CMGL: (0-4)

(+CMGF=1) AT+CMGL=? | +CMGL: ("REC UNREAD", "REC READ", "STO UNSENT", "STO SENT", "ALL")

<estados>:

0 | "REC UNREAD": Almacenado en Bandeja de entrada y sin leer.

1 | "REC READ": Almacenado en Bandeja de entrada y leído.

2 | "STO UNSENT": Almacenado en Bandeja de salida y sin enviar.

3 | "STO SENT": Almacenado en Bandeja de salida y enviado.

4 | "ALL": Todos los mensajes almacenados.

2) AT+CMGL=<estado>

- Lista todos los mensajes almacenados correspondientes al estado especificado.

- Sintaxis: AT+CMGL=<estado> | +CGML: <índice>, <estado>, <número>, [otros parámetros opcionales] , <timestamp><CR><LF><Cuerpo del mensaje>

<indice> Posición que ocupa el mensaje SMS en la memoria.

<estado> Estado del mensaje.

<número> Cadena de texto con el número de teléfono origen del mensaje.

<timestamp> Fecha y hora.

<CR><LF> Retorno de carro y salto de linea.

- Resultado a AT+CMGL="STO UNSENT":

+CMGL: 16,"STO UNSENT","679123456",,

Hola! Esto es un sms almacenado en memoria. Luego puede ser enviado... Salu2

AT+CMGR: [Lee Mensajes]

- Permite leer mensajes SMS de la bandeja de entrada.

- Sintaxis: AT+CMGR=<índice> | +CMGR: <estado>, <número>, [otros parámetros] , <timestamp><CR><LF><Cuerpo del mensaje>

<indice> Posición que ocupa el mensaje SMS en la memoria.

<estado> Estado del mensaje.

<número> Cadena de texto con el número de teléfono origen del mensaje.

<timestamp> Fecha y hora.

<CR><LF> Retorno de carro y salto de linea

- Respuesta a AT+CMGR=1:

+CMGR: "REC READ","227",,"05/07/12,14:13:02+08"

Movistar info: Ahora, GRATIS, tus Llamadas Perdidas vienen con el NOMBRE de la persona que te llamo, si esta en tu agenda. Para volver al SMS del 200 llama 283

AT+CMGD: [Borra Mensajes]

AT+CMGD=?

- Muestra la implementación del comando.

- Sintaxis: AT+CMGD=? | +CMGD: (lista de índices soportados)[,(lista de delflags soportadas)]

- Respuesta: +CMGD: (1-15),(0-4) //1-15 indica que la memoria SIM puede almacenar de 1 a 15 mensajes SMS

2) AT+CMGD=<índice>

- Elimina el mensaje de índice especificado.

- Sintaxis: AT+CMGD=1 elimina el mensaje con índice 1, es decir, el primer mensaje de la bandeja de entrada de mensajes SMS.

AT + CPMS = “ME”,”SM”

Este uso para cambiar de “banco” de memoria, (memoria de teléfono o SIM).

Seleccione el mensaje en el área guardada, se usa para leer el SMS, borrarlo y recibirlo

+CPMS= [mensaje_almacenado1] [, mensaje_almacenado2 [mensaje_almacenado2]

Para tener una idea antes describir los parámetros el comando, quedaría como sigue:

AT+CPMS="ME","SM","MT"

El [mensaje_almacenado1]

El primer parámetro, +CPMS= [mensaje_almacenado1], especifica que el mensaje almacenado en el área determinada, será usado cuando se lee o se borra el SMS.

El [mensaje_almacenado2]

El primer parámetro, +CPMS= [mensaje_almacenado2], especifica que el mensaje almacenado en el área determinada, será usado cuando se envíe el SMS.

El [mensaje_almacenado3]

El primer parámetro, +CPMS= [mensaje_almacenado3], especifica que el mensaje almacenado en el área determinada, será usado cuando se recibe el SMS.

Los valores que pueden ser asignados al comando +CPMS AT son los valores definidos en la especificación del SMS que puede ser asiganada a los parámetros mensaje_almacenado1, mensaje_almacenado2, mensaje_almacenado3.

SM. Se refiere al mensaje almacenado en el área del SIM card.

ME. Se refiere al mensaje almacenado en el área del modem GSM/GPRS o del equipo celular.

MT. Se refiere al mensaje almacenado en el área asociada con el modem GSM/GPRS o del equipo se combina el MT con el ME, cuando no hay espacio en cualquiera de ellos.

BM. Se refiere al mensaje almacenado en el área de información.

SR. Se refiere al mensaje almacenado en el área de reportes.

TA. Se refiere al mensaje almacenado en el área de algún terminal de un adaptador.

Capítulo 4

Sistema de Gestión y Monitoreo Local GPS – Google Maps

Para nuestro propósito hemos diseñado un sistema de gestión y monitoreo, el cual lo hemos denominado **LocalGPS – GoogleMaps**, del cual deducimos **LocalGPS** Localizador de con dispositivo de Sistema de Posicionamiento Global, y **GoogleMaps** porque en el sistema también contamos con la ayuda de los scripts del ya conocido Google Maps, el cual es un sistema en línea (Online) que se encuentra cartografiado la mayor parte del mundo y además su uso es gratuito hasta el momento.

Las ventajas del uso del scripts Google Maps, es sencillamente la rápida obtención de los mapas que se encuentran en línea, para nuestro propósito el mapa de la Ciudad de Guayaquil.

Este sistema está diseñado en una plataforma de programación llamada Microsoft Visual Studio 6.0, y específicamente en lenguaje de programación Microsoft Visual Basic 6.0.

Las herramientas de desarrollo empleadas en el sistema **LocalGPS – GoogleMaps**, son entre las principalmente indispensables:

MS COMM.

Para la comunicación con los puertos seriales en nuestro caso el Dispositivo USB CA-42.

Web Browser.

Para la comunicación Online con el servidor de Google Maps, utilizando de por medio el scripts de Google Maps.

Common Dialog.

Para la comunicación de los componentes comunes de Windows.

Windows Media Player.

Para la reproducción de sonidos multimedios, utilizados por nuestro sistema.

Timer.

Para la sincronización e interacción de nuestro sistema con sus puertos de puertos de comunicación que se encuentra utilizando para el propósito ya establecido.

OLE.

Para la comunicación con la Base de Datos, en la cual se encuentra almacenada la información que se necesita para el acceso, configuración y monitoreo de las diferentes opciones que se encuentran en el sistema.

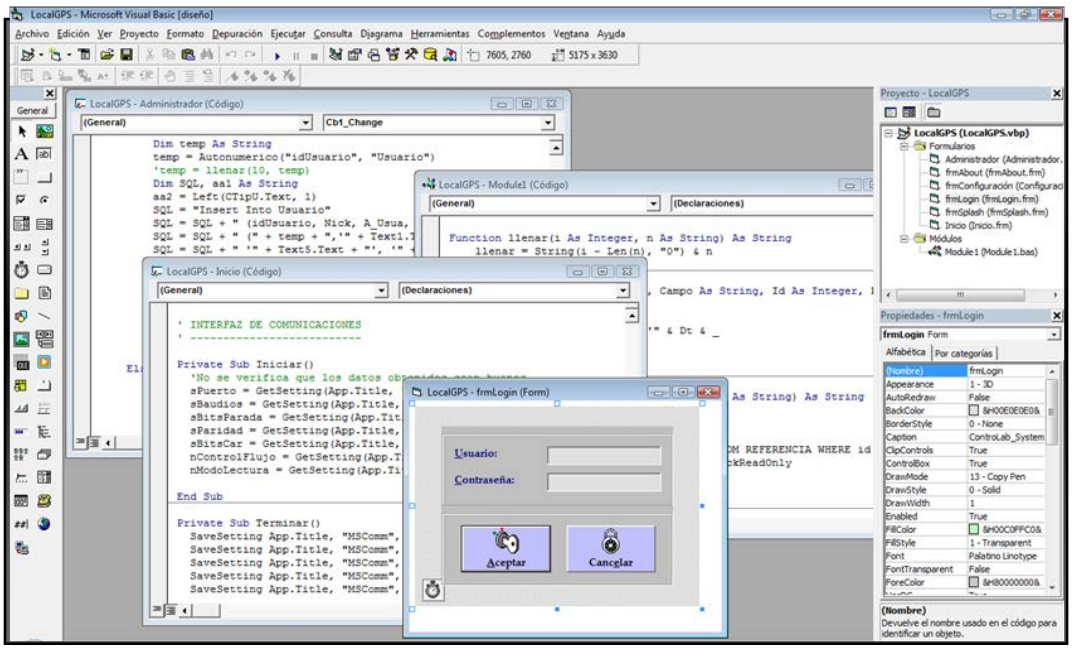

Fig. 4.1 Proceso de desarrollo del Sistema LocalGPS – GoogleMaps.

OTRAS

Además con la ayuda del Gestor de Base de Datos Microsoft Office ACCESS 2007, que nos ayuda con el almacenamiento crucial de la información que estamos usando en tiempo de ejecución del sistema **LocalGPS – GoogleMaps,** siendo también de vital importancia para el acceso seguro al sistema, puesto que en ella se guarda la información personal del usuario tales como:

Nombre de Usuario.

Contraseña ó Clave de Acceso.

En cuanto al diseño de la Base de datos utilizamos el Gestor de Base de Datos Microsoft Office ACCESS 2007, el cual se encuentra constituido por múltiples herramientas necesarias para el diseño, uso y optimización de los recursos de la Base de Datos.

Para tal propósito declaramos tres Tablas de Datos indispensables, las cuales hemos denominado como:

Principal.

Es la Tabla de Datos que contiene la información actualizada de las diferentes acciones que se realizan en tiempo de ejecución.

Usuario.

En esta Tabla de Datos se encuentra almacenada la información personal de cada uno de los usuarios, con fines de seguridad.

Referencia

Esta Tabla de Datos se guardan parámetros de referencia de las localizaciones Realizadas por el sistema GPS-Móvil.

4.1 Sistema Local GPS - GoogleMaps

Este sistema específicamente construido con fines educativos es el resultado de varios meses de investigación y esfuerzo de las partes que hacemos posible esta Tesis.

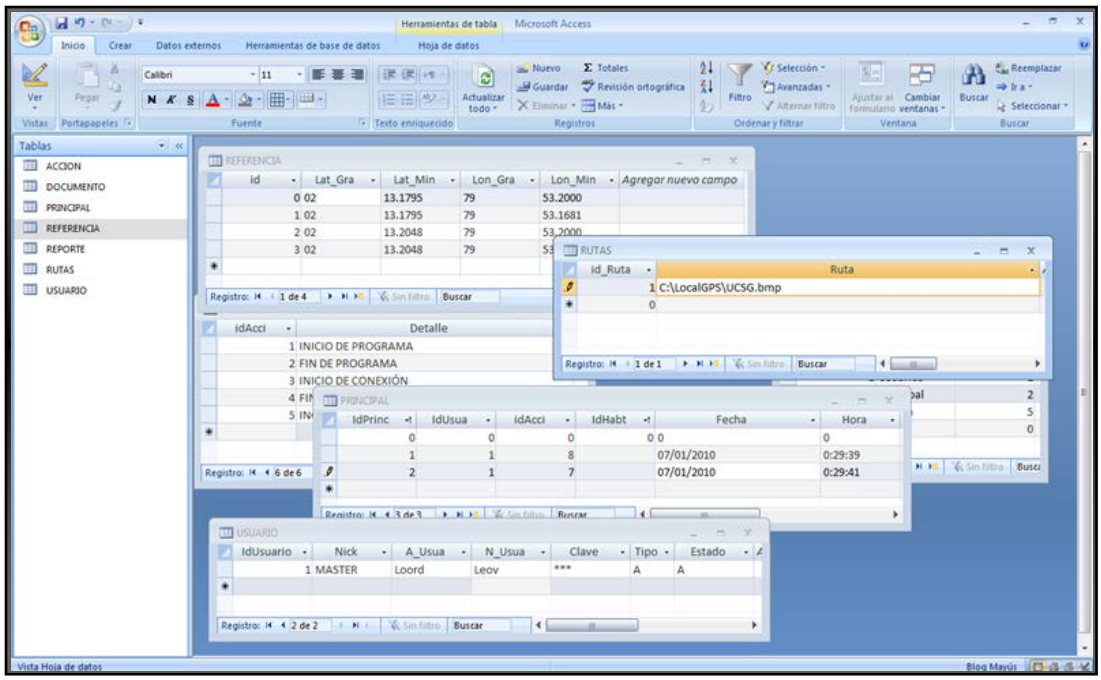

Fig. 4.2 Proceso de desarrollo de la Base de Datos para el Sistema LocalGPS – GoogleMaps.

Para poder utilizar el sistema se requiere instalarlo en un PC que tenga como sistema operativo Microsoft Windows XP en adelante.

Luego de haber instalado uno a uno los componentes necesarios para el óptimo funcionamiento del sistema nos disponemos al reconocimiento y la utilización del mismo.

Buscamos el icono de acceso al Sistema LocalGPS – GoogleMap.exe y lo ejecutamos dando doble clic en el icono e inmediatamente comienza a ejecutarse la aplicación.

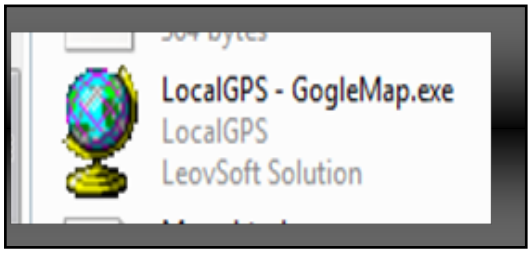

Fig. 4.3 Icono de Acceso a LocalGPS – GoogleMaps.

Al comienzo de la ejecución de la aplicación se visualiza la ventana de acceso en la cual debemos escribir el nombre de usuario y la contraseña, para poder ingresar.

El usuario por default es "**master**" y la contraseña es "**gps**", se puede crear varios usuarios más, luego de haber ingresado por primera vez al sistema.

Fig. 4.4 Ventana de acceso a LocalGPS – GoogleMaps.

Al aceptar el acceso al Sistema se presenta la ventana de bienvenida al sistema en el cual se detalla a manera de información sobre el motivo del desarrollo del proyecto.

Tales como los nombre de los desarrolladores, la plataforma de programación empleada para el desarrollo del mismo, el número de versión del producto, el año de lanzamiento del producto y la autorización de su uso de forma exclusiva a la Universidad Católica Santiago de Guayaquil (UCSG).

Fig. 4.5 Ventana de presentación del Sistema LocalGPS – GoogleMaps.

Luego de un lapso de tiempo de carga controlado de los diversos dispositivos y herramientas útiles para el desenvolvimiento apropiado del sistema.

Se carga la pantalla principal de monitoreo en la cual podemos divisar varios de sus principales características desarrolladas para la configuración y monitoreo del dispositivo GPS-Móvil.

A continuación un vistazo a la ventana principal de monitoreo del sistema, en la cual se alcanza a divisar que no se encuentra enlazado el sistema a la internet y desde luego tampoco al modem GSM.

Se alcanza a reconocer los varios parámetros que rodean al propósito de nuestra tesis. La ubicación de cada una de las diferentes informaciones que se presentan en esta Ventana está debidamente planificada para el sencillo uso del operador del mismo, dejando como parámetros de configuración y administración del mismo sistema ocultos con la finalidad de no perder información de tipo valioso para el correcto funcionamiento de las diferentes herramientas encargadas del monitoreo del Modulo GPS-Móvil, hasta del mismo dispositivo.

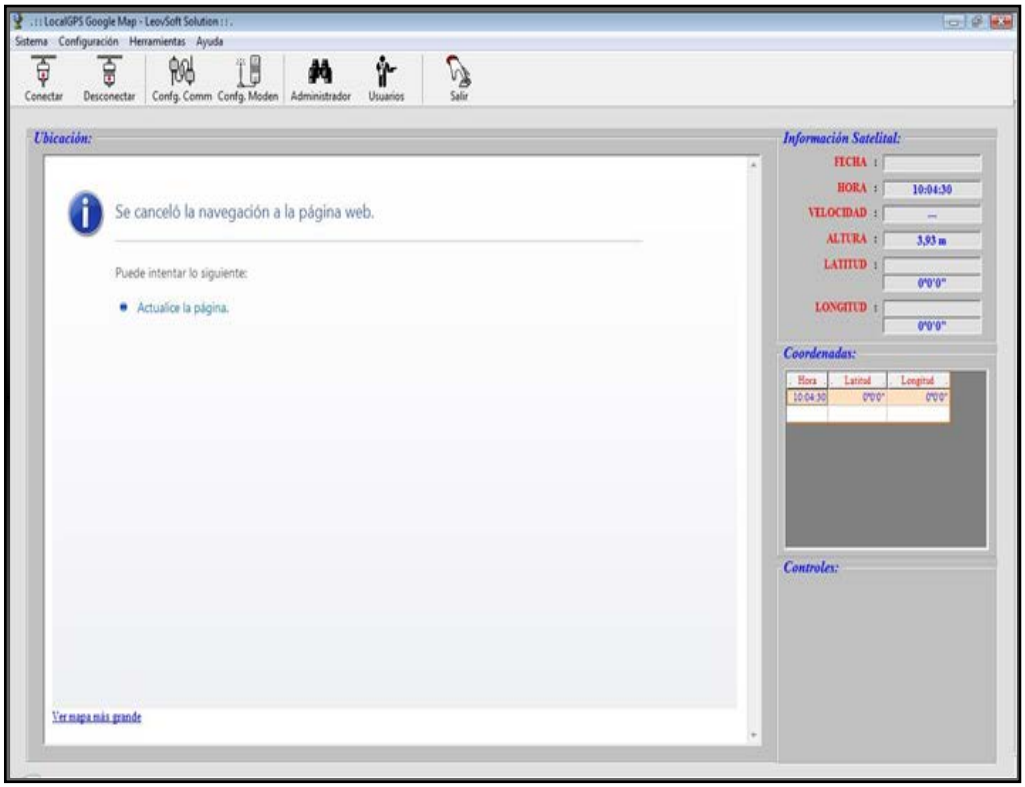

Fig. 4.6 Ventana principal del Sistema LocalGPS – GoogleMaps

La barra botones de acceso inmediato a las funciones hábiles para el usuario se encuentran en la parte superior de la ventana justamente en la segunda fila después de la fila de Menús.

Detallamos uno a uno los diferentes botones:

Conectar (Conectar Comm).

Sirve para establecer o abrir la comunicación entre el puerto serial del PC en el que se encuentra instalado el modem GSM y el Sistema LocalGPS - GoogleMaps.

Desconectar (Desconectar Comm).

Deshabilita la comunicación que ya se encuentra establecida entre el puerto serial del PC en el que se encuentra instalado el modem GSM y el Sistema LocalGPS – GoogleMaps, de esta forma ahorra energía, y baja el uso de recursos de memoria y procesamiento de datos.

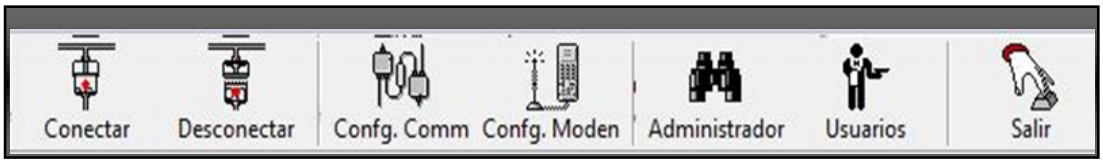

Fig. 4.7 Barra de Acceso rápido a las funciones: Conectar COMM, Desconectar COMM, Configurar COMM, Configurar modem, Administrador de conexión, Administrador de Usuarios y Salir.

Confg. Comm (Configurar Comm).

Presenta una ventana la cual se deben administrar los diferentes parámetros de comunicación.

Confg. Modem (Configurar Modem).

Presenta una ventana en la cual se puede definir el protocolo de comunicación con el Modem que se encuentra conectado al Pc y el Dispositivo GPS-Móvil.

Administrador (Administrador de Comunicación).

Visualiza un formulario el cual se encuentra una opción de envió y recepción de prueba entre los Dispositivos ya configurados en los Botones anteriores.

Usuario (Administrador de Usuarios).

Despliega un formulario en el cual nos permite el ingreso de nuevos usuarios al Sistema LocalGPS – GoogleMaps, por motivos de seguridad del mismo y para llevar un control de usuarios.

Salir (Salir del Sistema).

Con este botón podemos salir de una manera segura, guardando los parámetros necesarios para el correcto reinicio del Sistema LocalGPS – GoogleMaps, cerrando las diferentes comunicaciones y la Base de Datos.

Además contamos con el menú tradicional que tiene en su mayoría las mismas funciones antes expuestas, pero con la opción de teclas de acceso rápido.

A continuación detallamos uno a uno los diferentes menús que están diseñados en el Sistema LocalGPS – GoogleMaps:

Establecer Conexión (F2).- Al igual que el botón de acceso rápido del mismo nombre, sirve para establecer la conexión con el puerto serial, y pone en modo atento cualquier evento que suceda en el mismo.

Cortar Conexión (F3).- Cumple la misma función del botón Deshabilitar, pues corta la conexión que ha sido habilitada anteriormente en el puerto serial y ayuda al ahorro significativo de recursos de memoria y procesamiento de su sistema operativo.

Salir (Ctrl+Q).- Este menú realiza varias acciones antes de salir del Sistema LocalGPS – GoogleMaps. cerrando las acciones pendientes y permitiendo liberar memoria que estuvo usando el sistema en tiempo de ejecución.

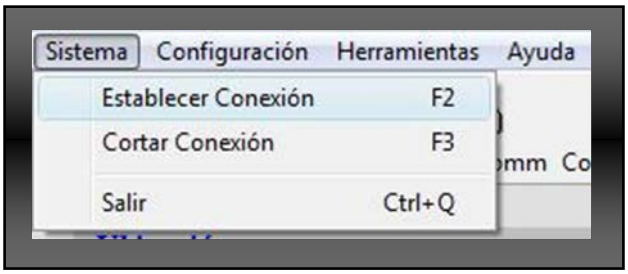

Fig. 4.8 Menú Sistema: Establecer Conexión, Cortar Conexión y Salir.

Configuración.

Modem GSM (F5).- Despliega un formulario en el cual se configura los parámetros de la conexión con los dispositivos GSM que son necesarios para la comunicación vía SMS.

Parámetros RS232 (F4).-

Visualiza una ventana en la que se configura uno a uno los parámetros del tipo de comunicación con el puerto serial, en nuestro caso utilizares el protocolo de comunicación RS232, puesto que el más apropiado para los fines de nuestra Tesis.

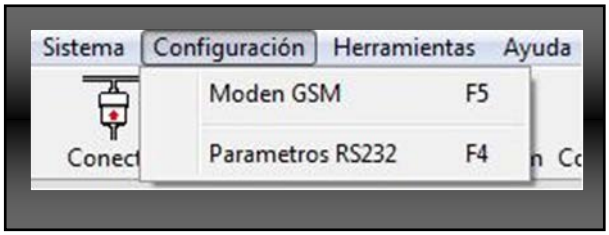

Fig. 4.9 Menú Configuración: Modem GSM, Parámetros RS232.

Herramientas.

Administrador (Ctrl+A).- Este menú el cual tiene la misma función que el botón antes descrito del mismo nombre, administra la comunicación de entrada y salida e interacción del puerto serial, con la finalidad de revisar si hubiese perdida de datos.

Usuarios (Ctrl+U).- Del mismo modo las funciones programadas en este menú tienen relación con el botón del mismo nombre, siendo estas del ingreso de nuevos usuario al Sistema LocalGPS – GoogleMaps, con mayor seguridad, evitando posibles infiltraciones por parte de personas no autorizadas al acceso del mismo.

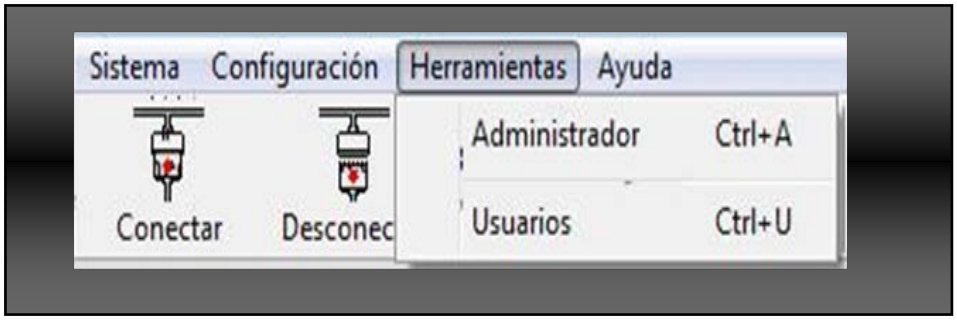

Fig. 4.10 Menú Herramientas: Administrador, Usuarios.

Ayuda.

Soporte Técnico (F1).- Eta parte del Sistema LocalGPS – GoogleMaps, nos brindara la información necesaria para poder contactar a los encargados del desarrollo del Sistema, y así poder solucionar conflictos dentro del sistema que no se han puesto por definido.. Siendo este un Sistema con fines Educativos.

Fig. 4.11 Menú Ayuda: Soporte Técnico, Acerca de LocalGPS - GoogleMaps.

Acerca de Local GPS – GoogleMaps.- En este enlace nos desplegara una pantalla de información más detallada de las características del sistema LocalGPS – GoogleMaps, además cuenta con un botón el cual nos mostrara la información

abstracta del sistema operativo que estamos utilizando, con fines de seguimiento de errores en tiempo de ejecución.

4.2 Formularios anexados

Los formularios anexados en el Sistema LocalGPS – GoogleMaps, son de gran importancia para el correcto desenvolvimiento del mismo, ya que en ellos debemos configurar los diferentes parámetros debidamente señalizados, con la finalidad de llevar a cabo las diferentes funciones y eventos que se generan en el transcurso de las diferentes tipos de comunicaciones que estamos empleando.

Para ello detallaremos a continuación los tres formularios bases del Sistema LocalGPS – GoogleMaps, los cuales tienen sus funciones específicas y a la vez comparten funciones entre sí, los formularios son:

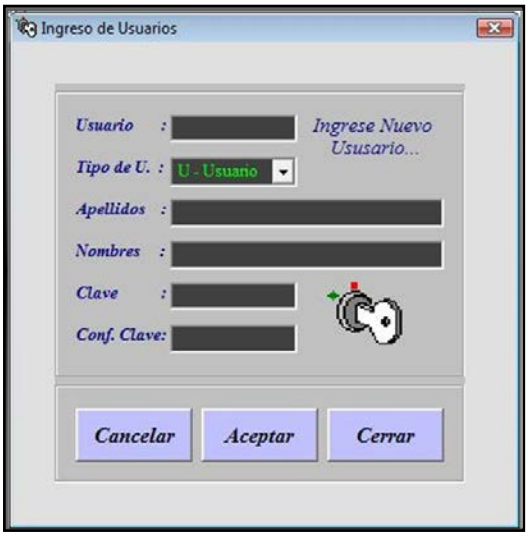

Fig. 4.12 Formulario de Ingresos de Usuarios

Ingreso de Usuarios.

Con este formulario podemos ingresar nuevos usuarios a la Base de Datos del Sistema.

Para tal motivo debemos registrar la información que esta detallada en el formulario, como razón principal debemos escoger los privilegios de usuarios, ya que hay dos tipos de usuario; el tipo Administrador y el tipo Usuario.

Configuración Comum

En este formulario debemos especificar los parámetros de configuración para el tipo de comunicación que se va a emplear en la comunicación del puerto serial con el dispositivo USB CA-42.

Siendo por consideración de las reglas de comunicación previamente establecidas en el modem a utilizar, en nuestro caso es un teléfono celular Nokia 3220, el cual tiene un protocolo de comunicación bastante sensible y trabaja a 9600 baudios, con un bit de parada y ocho bits por cada carácter enviado y recibido, además el control debe ser Ninguno como también la paridad, el modo de lectura de los caracteres recibidos tiene que ser Modo Texto, como ya lo establece el protocolo de comunicación RS232 anteriormente analizado.

Existe un botón de valores predeterminados para la comunicación RS232, el cual configura automáticamente los valores no obstante se debe seleccionar de manera manual y cuidadosamente el número de Puerto serie tal como el COM1, debido que en el momento de instalación del Dispositivo USB CA-42, se establece para el mismo.

Administrador.

Este formulario principalmente nos ayuda a la verificación de los distintos eventos que se producen en el puerto de comunicación serial el cual está siendo utilizado con el Modem en nuestro caso con el teléfono celular Nokia 3220, por su costo y fácil comercialización en el mercado.

Alcanzamos a visualizar el Cuadro de texto en el cual se recibirá los datos que lleguen al Modem que se encuentra conectado al PC, también tenemos el recuadro de texto de envió de las diferentes instrucciones de prueba para el Modem conjuntamente con el botón enviar, el cual ejecuta la acción de envió del comando AT directamente al puerto Serial.

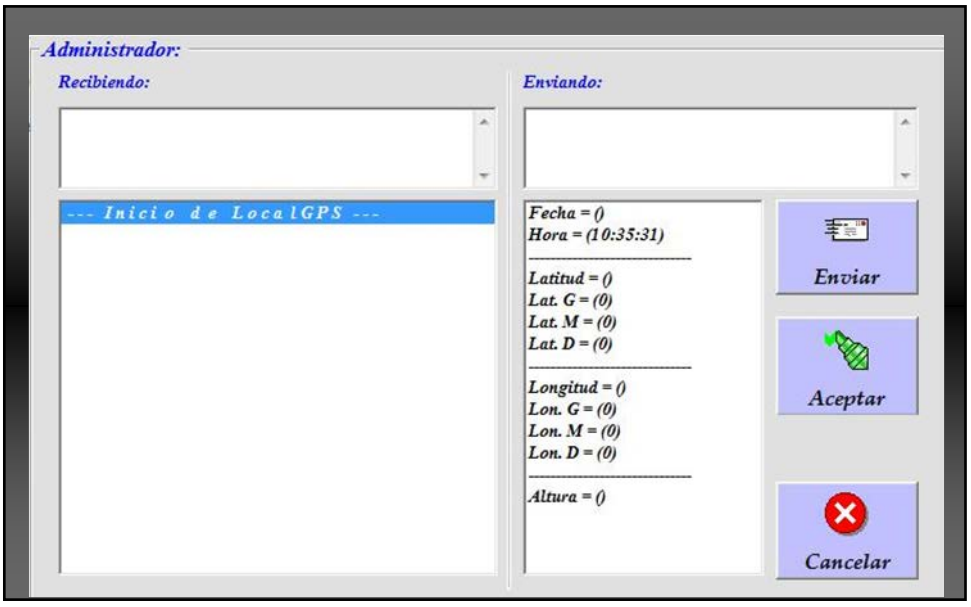

Fig. 4.13 Formulario de Administración de la comunicación.

Tenemos también una lista del historial de recepción del puerto serial, y lo complementa el recuadro detallado del lado derecho del formulario.

Fig. 4.14 Formulario de Configuración Comm

Estándar de comunicaciones RS-232

El puerto serie RS-232C, presente en todos los ordenadores actuales, es la forma más comúnmente usada para realizar transmisiones de datos entre ordenadores. El RS-232C es un estándar que constituye la tercera revisión de la antigua norma RS-232, propuesta por la EIA (Asociación de Industrias Electrónicas), realizándose posteriormente un versión internacional por el CCITT, conocida como V.24. Las diferencias entre ambas son mínimas, por lo que a veces se habla indistintamente de V.24 y de RS-232C (incluso sin el sufijo "C"), refiriéndose siempre al mismo estándar.

El RS-232C consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, mas barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC). En cualquier caso, los PCs no suelen emplear más de 9 pines en el conector DB-25. Las señales con las que trabaja este puerto serie son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V.

Dependiendo de la velocidad de transmisión empleada, es posible tener cables de hasta 15 metros.

Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Las más importantes son:

Pin	Función
TXD	(Transmitir Datos)
RXD	(Recibir Datos)
DTR	(Terminal de Datos Listo)
DSR	(Equipo de Datos Listo)
RTS	(Solicitud de Envío)
CTS	(Libre para Envío)
DCD	(Detección de Portadora)

Tabla. 4.1 Funciones de los Pines del RS-232

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS y DCD son de entrada. La masa de referencia para todas las señales es SG (Tierra de Señal). Finalmente, existen otras señales como RI (Indicador de Llamada), y otras poco comunes que no se explican en este artículo por rebasar el alcance del mismo.

Numero	de Pin	Señal	Descripción	E/S
En DB-25	En DB-9			
1	1	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC(*)	Transmit Clock	S
17	-	RxC(*)	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC(*)	Transmit/Receive Clock	S

Tabla. 4.2 Descripción de los Pines del RS-232

(*) = Normalmente no conectados en el DB-25

4.3 Puerto Serial RS 232

El ordenador controla el puerto serie mediante un circuito integrado específico, llamado UART (Transmisor-Receptor-Asíncrono Universal). Normalmente se utilizan los siguientes modelos de este chip: 8250 (bastante antiguo, con fallos, solo llega a 9600 baudios), 16450 (versión corregida del 8250, llega hasta 115.200 baudios) y 16550A (con buffers de E/S). A partir de la gama Pentium, la circuiteria UART de las placa base son todas de alta velocidad, es decir UART 16550A. De hecho, la mayoría de los módems conectables a puerto serie necesitan dicho tipo de UART, incluso algunos juegos para jugar en red a través del puerto serie necesitan de este tipo de puerto serie. Por eso hay veces que un 486 no se comunica con la suficiente velocidad con un PC Pentium... Los portátiles suelen llevar otros chips: 82510 (con buffer especial, emula al 16450) o el 8251 (no es compatible).

Para controlar al puerto serie, la CPU emplea direcciones de puertos de E/S y líneas de interrupción (IRQ). En el AT-286 se eligieron las direcciones 3F8h (o 0x3f8) e IRQ 4 para el COM1, y 2F8h e IRQ 3 para el COM2. El estándar del PC llega hasta aquí, por lo que al añadir posteriormente otros puertos serie, se eligieron las direcciones 3E8 y 2E8 para COM3-COM4, pero las IRQ no están especificadas. Cada usuario debe elegirlas de acuerdo a las que tenga libres o el uso que vaya a hacer de los puertos serie (por ejemplo, no importa compartir una misma IRQ en dos puertos siempre que no se usen conjuntamente, ya que en caso contrario puede haber problemas). Es por ello que últimamente, con el auge de las comunicaciones, los fabricantes de PCs incluyan un puerto especial PS/2 para el ratón, dejando así libre un puerto serie.

Mediante los puertos de E/S se pueden intercambiar datos, mientras que las IRQ producen una interrupción para indicar a la CPU que ha ocurrido un evento (por ejemplo, que ha llegado un dato, o que ha cambiado el estado de algunas señales de entrada). La CPU debe responder a estas interrupciones lo más rápido posible, para que dé tiempo a recoger el dato antes de que el siguiente lo sobrescriba. Sin embargo, las UART 16550A incluyen unos buffers de tipo FIFO, dos de 16 bytes (para recepción y transmisión), donde se pueden guardar varios datos antes de que la CPU los recoja. Esto también disminuye el número de interrupciones por segundo generadas por el puerto serie.

El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a unas velocidades determinadas (normalmente, 9600 bits por segundo o más). Después de la transmisión de los datos, le sigue un bit opcional de paridad (indica si el numero de bits transmitidos es par o impar, para detectar fallos), y después 1 o 2 bits de Stop. Normalmente, el protocolo utilizado ser 8N1 (que significa, 8 bits de datos, sin paridad y con 1 bit de Stop).

Una vez que ha comenzado la transmisión de un dato, los bits tienen que llegar uno detrás de otro a una velocidad constante y en determinados instantes de tiempo. Por eso se dice que el RS-232 es asíncrono por carácter y síncrono por bit. Los pines que portan los datos son RXD y TXD. Las demás se encargan de otros trabajos: DTR indica que el ordenador esta encendido, DSR que el aparato conectado a dicho puerto esta encendido, RTS que el ordenador puede recibir datos (porque no está ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación, presencia de datos.

Tanto el aparato a conectar como el ordenador (o el programa terminal) tienen que usar el mismo protocolo serie para comunicarse entre sí. Puesto que el estándar RS-232 no permite indicar en que modo se está trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes. Como ya se ha visto, los parámetros que hay que configurar son: protocolo serie (8N1), velocidad del puerto serie, y protocolo de control de flujo. Este ultimo puede ser por hardware (el que ya hemos visto, el handshaking RTS/CTS) o bien por software (XON/XOFF, el cual no es muy recomendable ya que no se pueden realizar transferencias binarias). La velocidad del puerto serie no tiene porque ser la misma que la de transmisión de los datos, de hecho debe ser superior. Por ejemplo, para transmisiones de 1200 baudios es recomendable usar 9600, y para 9600 baudios se pueden usar 38400 (o 19200).

Este es el diagrama de transmisión de un dato con formato 8N1. El receptor indica al emisor que puede enviarle datos activando la salida RTS. El emisor envía un bit de START (nivel alto) antes de los datos, y un bit de STOP (nivel bajo) al final de estos.

4.4 Construcción y medios generales para ensamblaje de GPS-Móvil

El dispositivo lo construimos sobre una tarjeta de circuito impreso a partir del diagrama esquemático detallado en el apéndice A la conexión del GPS es la única parte que podría representar dificultad debido a que únicamente acepta cable 26.

Para proceder a programar debemos soldar la corriente al IC y las dos hileras de tres pines para el puerto, el módulo sera fijado a la tarjeta con tornillos.

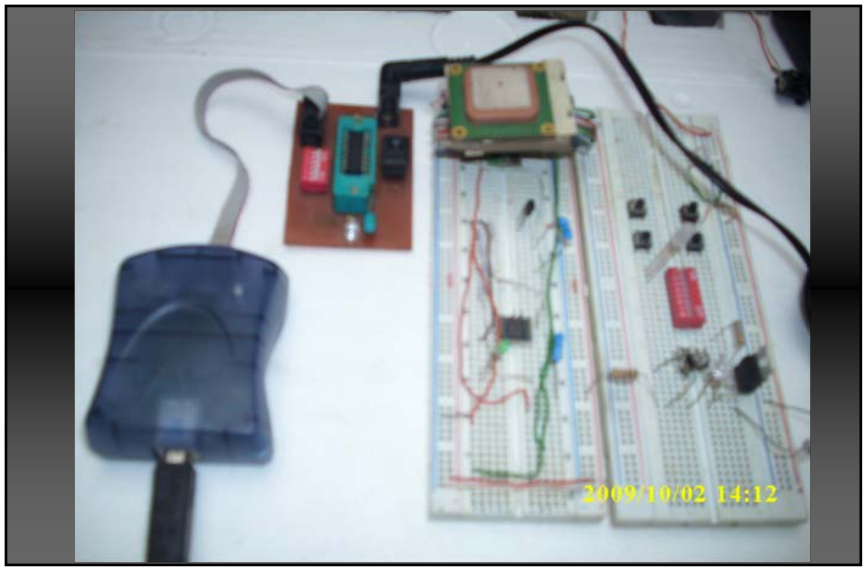

Fig. 4.15 Montaje de los Dispositivos en proceso de pruebas.

4.5 Materiales a Utilizar en el Modulo GPS-Móvil

Parte	Descripción	Cantidad
556 - ATTINY84 - 20PU	Atmel microcontrolador de AVR	2
827 - AME8811AEATZ	3.3V regulador de voltaje	2
522 - ZTX1151A	Transistor PnP con el voltaje de saturación bajo	2
581 - SA105E104M	Condensador de cerámica de 0.1 uF	10
581 - TAP106K006SCS	Condensador de tantalum de 10 uF	4
855 - M50 – 3501242	Conector 27 mm	3
171-0025 – ex	Enchufe de 2.5 mm para teléfono	2
571-1-390261 - 3 o 571-26415994	Toma de corriente de IC 14 pines	1
604 - WP937EGW	LED Rojo/Verde	1
604 - WP132XID	LED rojo	2
660MF1 / 4DC3321F	Resistencia de 3.32K Ohms	5
660MF1 / 4DC4700F	Resistencia 470 Ohms	5
12BH431	Base para batería AAA	1
340 - V23993 - A1035D	Modulo GPS Vincotech	1

Tabla. 4.3 Listado de Materiales a Utilizar en el Modulo GPS-Movil

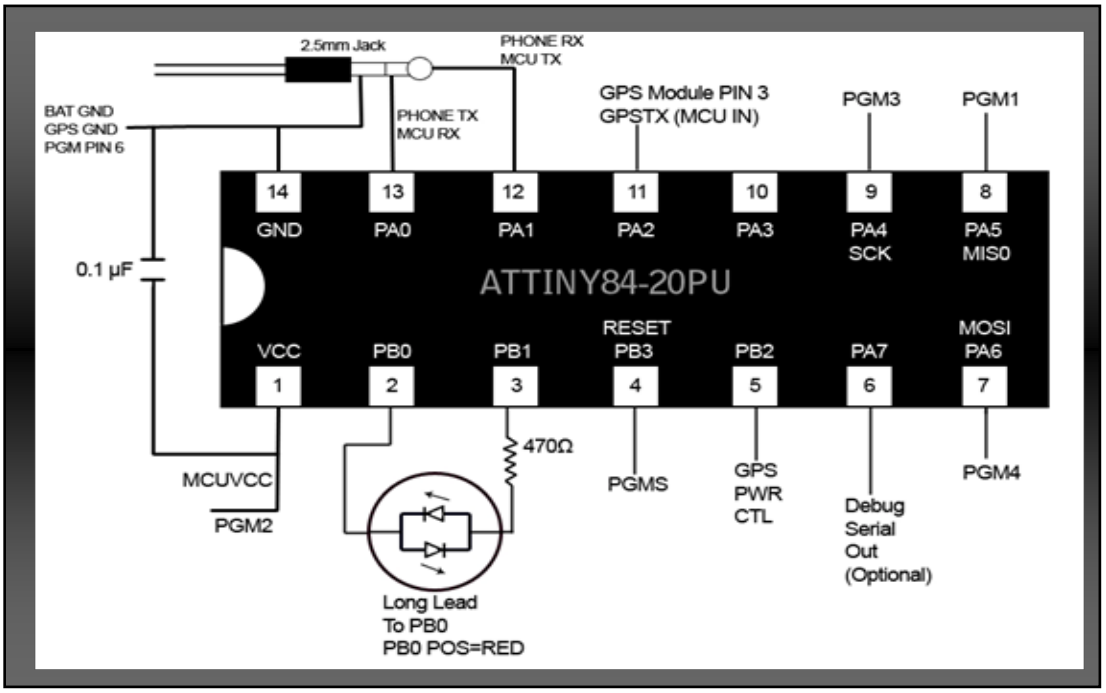

Fig. 4. 16 Esquema del Circuito del Micro controlador ATMEL ATTINY84-20PU

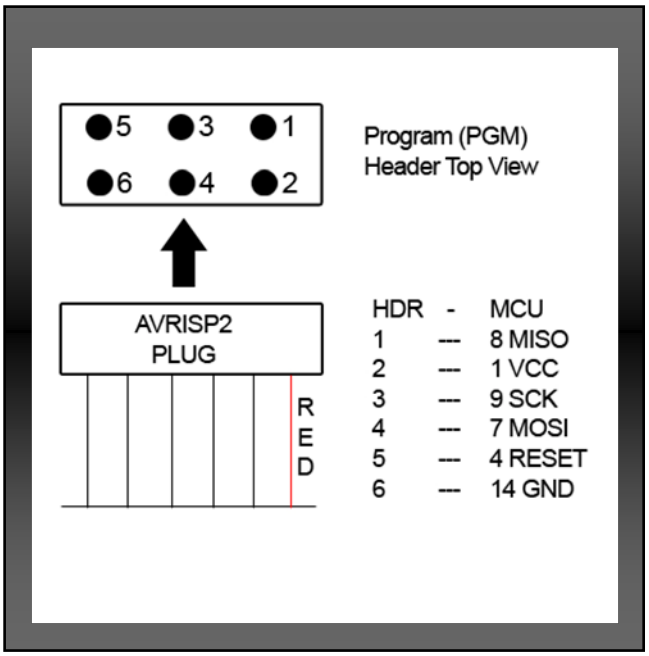

Fig. 4.17 Esquema de conexión al Programador de 4.5V.

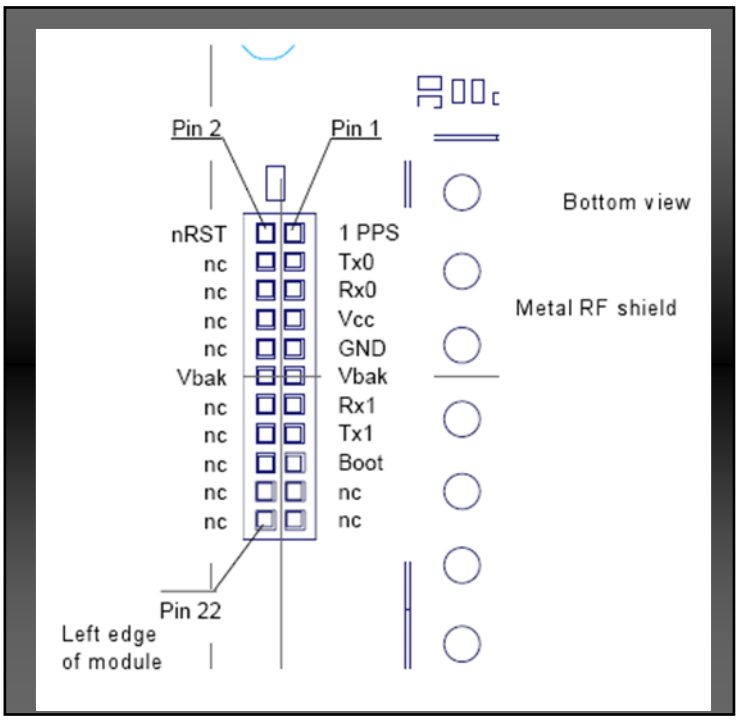

Fig. 4.18 Conector del Dispositivo Vincotech GPS.

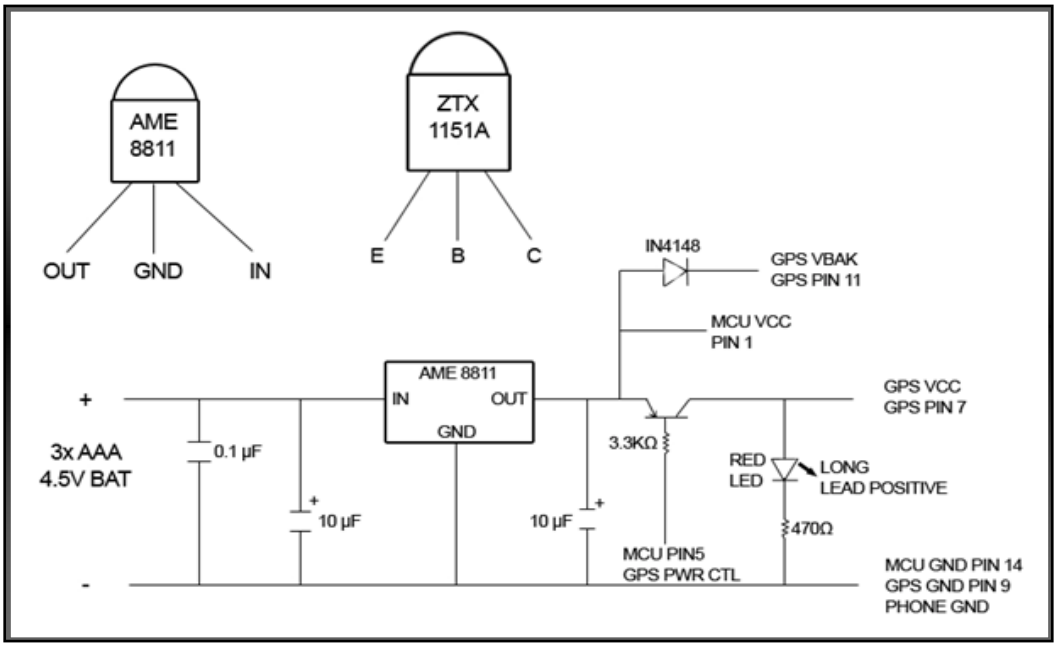

Fig. 4.19 Esquemático para toma de corriente del módulo GPS-Móvil.

Capítulo 5

Puesta en marcha y Pruebas

5.1 REVISION DEL EQUIPO CON LOS COMPONENTES INSTALADOS

Revisar la continuidad del terminal negativo de la batería al microcontrolador, al conector del teléfono, al regulador y GPS.

Revisar la continuidad del terminal positivo a la entrada del regulador.

Revisar cortocircuitos en los pines adyacentes del controlador, los pines para programacion, conector del teléfono y modulo GPS.

Ponga el modulo GPS sobre su receptáculo para asegurarse que la instalacion eléctrica sea la correcta.

Procedemos a suministrar energía a nuestro dispositivo revisamos el voltaje de alimentación del microcontrolador entre los pines 1 y 14 este debe ser 3.3 V, el del GPS debe acercarse a 3 V.

5.2 CODIGOS DE ESTADO

Los led de estado parpadean para reportar los códigos de estado.Cada parpadeo nos indica el digito del código que buscaremos en la siguiente tabla:

Clave	Descripción
11	Teléfono activo
12	Fallo envío de mensaje
13	Telefono inactivo
14	Ningún número de teléfono definido
21	Contraseña inválida
31	Reset
32	Eeprom inicializada con valores preestablecidos

Tabla. 5.1 Codigos de Estado de los Leds

5.3 CONFIGURACION FINAL

Instale el microcontrolador Atmel, y ponga la batería.

Proceda a programar en microcontrolador conectado el dispositivo al computador. Ejecute AVR Studio, ir a herramientas escoja mkII de AVRISP y USB luego de clic en conectar.

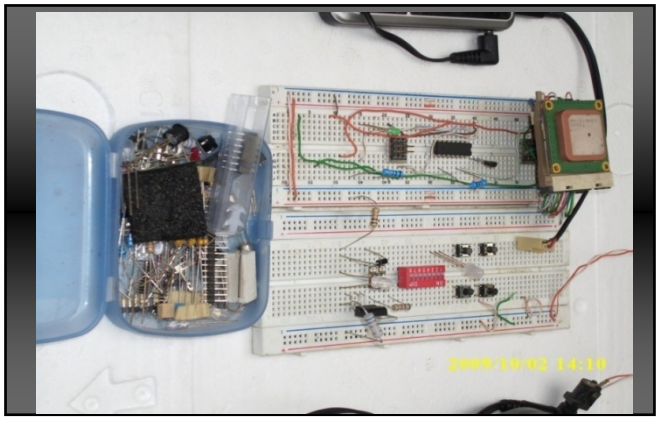

Fig. 5. 1 Micro controlador Atmel en tiempo de Pruebas del modulo GPS

Escoja el programa y ponga a Frecuencia de ISP a 125 kHz.

El dispositivo debe ser ATT84. El modo de programación es ISP.

En el flash escoja su archivo hexadecimal.

El LED de estado empezará a brillar intermitentemente poco después de que la programación termine, sin modulo GSM estos deben mostrar 31, 11 y 13 una SIM recientemente programado también exhibirá 33 de clave (EEPROM inicializado) una vez.

Instalar las baterías luego el plug de 2.5 mm al modulo GSM.

Usted debe recibir código 11 (revisando teléfono).

Fig. 5.2 Programando el Micro controlador y realizando las primeras pruebas en el Modulo GPS-Móvil.

5.4 CONFIGURACION DEL DISPOSITIVO

El rastreador GPS almacena una dirección de respuesta y envía los mensajes a la dirección de respuesta sin importar desde dónde vienen. La dirección de respuesta debe ser un número de teléfono móvil activo.

Para determinar este numero de respuesta enviar un mensaje al dispositivo con este código:

GPS SETADDRESS 59393402218

Donde 59393402218 es el número celular al que usted quiere el rastreador GPS-Móvil envié las respuestas. Las palabras GPS y SETADDRESS deben ser mayúsculas, para confirmar esto debemos recibir un mensaje diciendo COMANDO EJECUTADO.

5.5 FUNCIONAMIENTO DEL DISPOSITIVO

Una vez hemos realizado la configuración correspondiente podemos ya poner en funcionamiento el dispositivo, para esto enviaremos el mensaje:

GPS LOCATE

Donde GPS debe estar en la mayúscula.

Esta será la respuesta que recibirá nuestro receptor conectado al computador:

LOCATE POS 34 05.8779 N 118 20.6368 W ALT 377 M SPEED 0.0 KPH COURSE 11.05 AT 08/04/05 22:31:51 UTC SATS 04

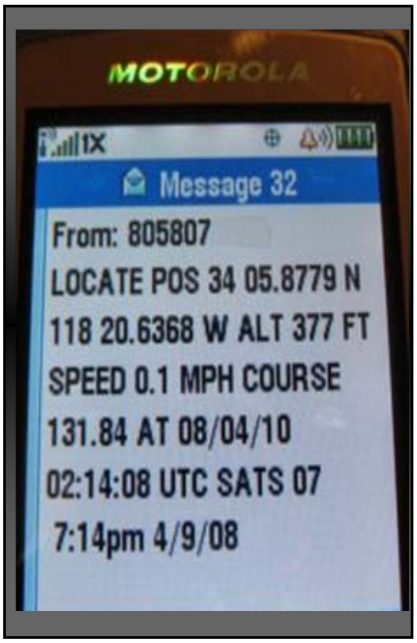

Fig. 5.3 Mensaje de prueba del Modulo GPS-Móvil.

Los campos lo son:

LOCATE: el informe es el resultado de una solicitud manual.

Pos 3405.8779 N 11820.6368 W: latitud, longitud en grados y los minutos de decimal.

Alt 377 M: Altura a la que se encuentra el dispositivo.

Speed 0.0 kph: Velocidad a la que se mueve el dispositivo.

AT 08/11/09 22:31:51: Mes, día, la hora, minuto y segundos en el que la medición fue tomada.

SATS 04: Número de satélites que proveen la medicion.

Fig. 5.4 Montaje final del dispositivo

Conclusiones

Una vez terminado el presente proyecto de tesis y luego de analizar el mismo de una forma detenida sabiendo que todo el conocimiento adquirido y requerido abarca multiples areas del campo científico hemos llegado a las siguientes conclusiones:

Cosideramos el trabajo de Titulacion o Tesis, el compendio de todos los conocimientos y destrezas que el estudiante adquiere a lo largo de su carrera universitaria los cuales serán las herramientas para su futuro desarrollo como profesional.

El diseño de sistemas de cualquier índole, basándonos en microcontroladores dan como resultado dispositivos practicos, resistentes, versátiles y económicos, debido a la facilidad y flexibilidad con la que estos son programados, existiendo gran variedad de los mismos en el mercado con multiples características y aplicaciones.

El dispositivo desarrollado en esta Tesis resulta muy económico algo que es un argumento muy validero en nuestra región ya que hace uso de una tecnología desarrollada para ser usada de forma gratuita como es el GPS.

El uso de Basic un lenguaje de alto nivel nos permite el desarrollo de aplicaciones para microcontroladores de alta fiabilidad y eficacia, gracias a las funciones con las

que dispone además nos permite realizar mejoras y mantenimientos con sencillez debido a que es un lenguaje estructurado.

El dispositivo desarrollado cumple su objetivo principal de proveer de una herramienta tecnológica a la comunidad que permita disminuir la cantidad de secuestros, robos de vehículos y otros tipos de ilícitos en donde una rápida respuesta de las autoridades sea crucial.

Recomendaciones

Concluida la tesis, consideramos las siguientes recomendaciones:

En el armado prestar especial atención a la posición del transistor de alimentación debe estar correctamente instalado. Si se invierte el emisor y el colector, el dispositivo dara como error que no existe flujo de datos.

En el momento de colocar el modulo GPS en su receptáculo debemos prestar especial atención con los pines ya que estos son muy fáciles de destruir, provocando perdidas costosas.

Si bien es cierto que el lenguaje de programación que usamos simplifica el código es necesario evitar el abuso de las funciones, ya que esto terminaría ralentizando en general la aplicación desperdiciando los recursos de sistema disponibles.

Antes de realizar la comunicación entre los modulos, es conveniente conocer las normas y trama de datos a las que se sujetan los protocolos de comunicaciones, a fin de poder usarlas correctamente durante la programación y que sirvan de guía para la Recepción o envio de datos.

GLOSARIO

A

B

BRG (Bearing): el rumbo entre dos puntos de pasos intermedios (*waypoints*)

C

CMG (Course Made Good): rumbo entre el punto de partida y la posición actual

---

D

DOP (Dilution Of Precision): medida de la precisión de las coordenadas obtenidas por GPS, según la distribución de los satélites, disponibilidad de ellos.

E

EPE (Estimated Postion Error): margen de error estimado por el receptor

ETE (Estimated Time Enroute): tiempo estimado entre dos waypoints

ETA (Estimated Time to Arrival): tiempo estimado de llegada al destino

---

F

I

L

LEDs: Son fuentes de luz con emisión espontánea o natural (no coherente), son diodos semiconductores de unión p-n que para emitir luz se polarizan directamente

---

M

Microcontrolador: es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y unidades de E/S (entrada/salida).

O

P

R

S

Semiconductor: Un semiconductor es un material sólido que conduce eléctricamente bajo ciertas circunstancias y no conduce en otras. Los semiconductores se utilizan en circuitos electrónicos. Son materiales que tienen propiedades eléctricas entre los óxidos y los metales.

Señal Analógica: Aquella **función matemática continua** en la que es variable su amplitud y periodo (representando un dato de información) en función del tiempo.

Señal Digital: Se dice que una señal es digital cuando las magnitudes de la misma se representan mediante valores discretos en lugar de variables continuas.

T

Transistor: El transistor es un dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador. El término "transistor" es la contracción en inglés de transfer resistor ("resistencia de transferencia").

V

W

Waypoint: Son coordenadas para ubicar puntos de referencia tridimensionales utilizados en la navegación fundamentada en GPS (Global Positioning System). La palabra viene compuesta del inglés way (camino) y point (punto), en realidad se emplean para trazar rutas mediante agregación secuencial de puntos.

PRESUPUESTO

CONCEPTO	VALOR
Elementos electrónicos	$ 960
Textos de referencia	$ 30
Servicios telefónicos y computacionales	$ 50
Transporte	$ 50
TOTAL	**$ 1090**

BIBLIOGRAFIA

SAPAG CHAIN Nassir, SAPAG CHAIN Reinaldo Preparación y evaluación de proyectos Editorial McGraw-Hill Interamericana, 2003

VALENCIA Ramiro, Aplicaciones electronicas con microcontroladores Editorial Microtel Julio 2006

MORTON, John, AVR: An Introductory Course Editorial Newnes Diciembre 2002

KÜHNEL, Claus, Avr Microcontrollers Handbook Editorial Newnes Agosto , 1998

BIBLIOGRAFÍA ELECTRÓNICA

Atmel	http://www.atmel.com/
Hojas de Datos	http://www.datasheetcatalog.org
API's Google Maps	http://code.google.com/intl/es-EC/apis/maps/documentation/javascript/v2/reference.html
Dontronics.	http://www.dontronics.com
Mouser	http://www.mouser.com/atmel/
AVR Assembler User Guide	http://www.atmel.com/dyn/resources/prod_documents/doc1022.pdf

Anexos

APENDICE A

DIAGRAMA ESQUEMATICO

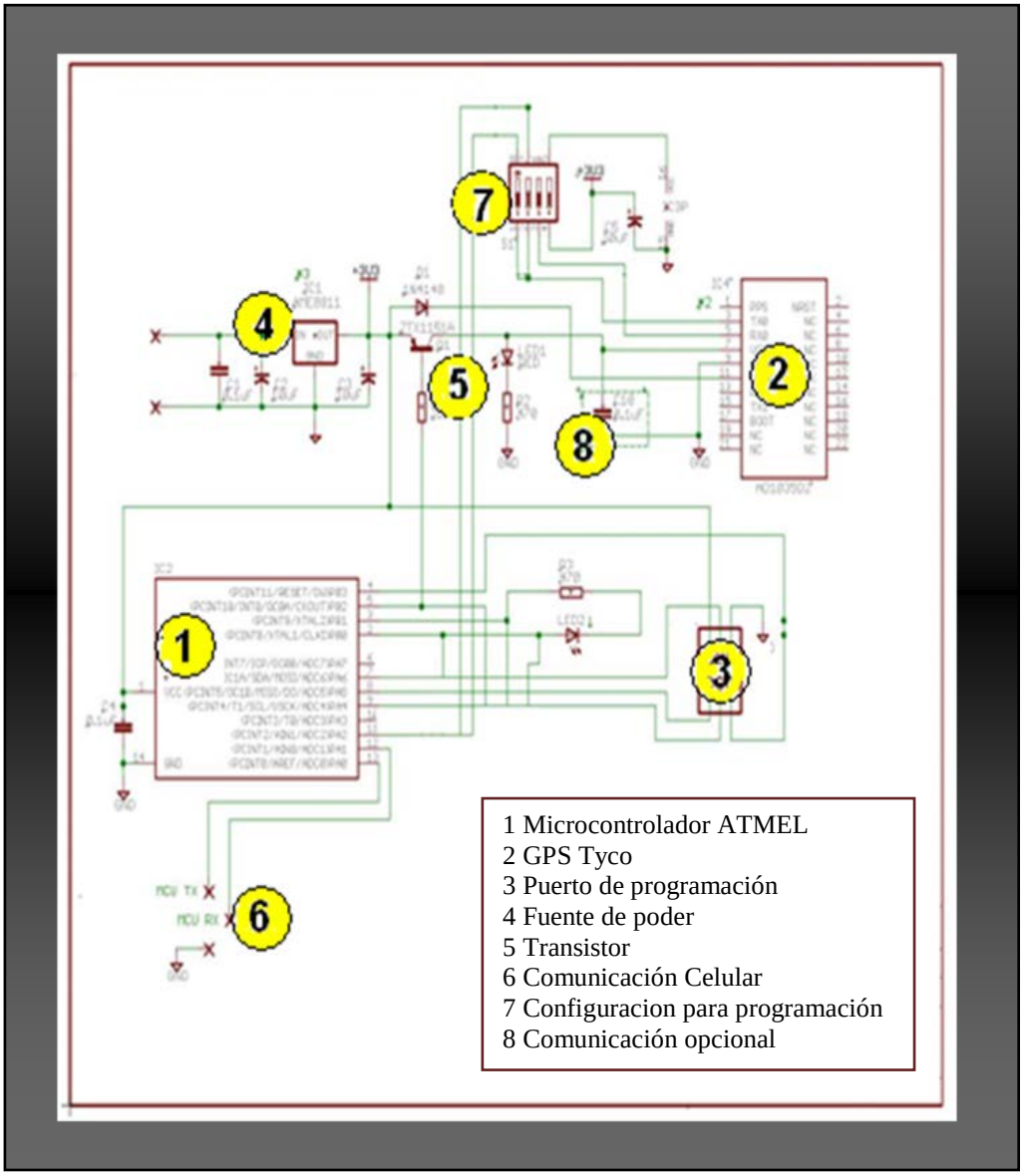

Fig. DIAGRAMA ESQUEMATICO

APENDICE B

CIRCUITO IMPRESO DISPOSITIVO

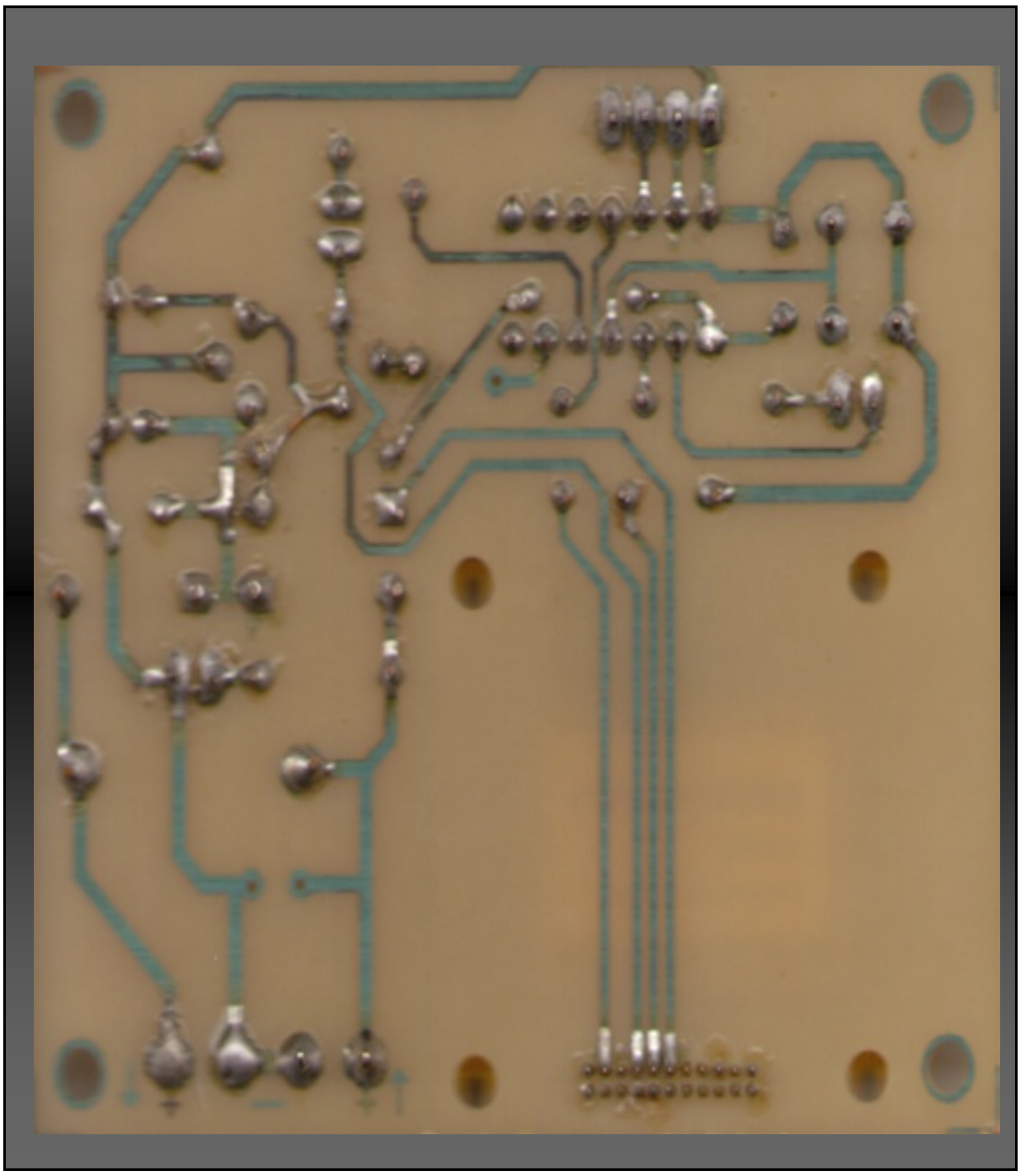

Fig. CIRCUITO IMPRESO DEL DISPOSITIVO

APENDICE C

Funciones Básicas definidas y utilizadas dentro del programa

SETADDRESS.- Determina la dirección de respuesta

LOCATE.- Hace el requerimiento inmediato de la ubicación del dispositivo.

SETSPEED.- Límite de velocidad, por omisión valor 0

SETNAME.- Nombre del Rastreador.

SETPASSWORD.- Contraseña del Modulo GPS.

SETADDRESS.- Número de teléfono del Receptor de datos.

TRACKON.- Activa la localización.

TRACKOFF.- Desactiva la localización.

POWERSAVE.- Modo ahorro de energía.

POWERON.- Arranca el dispositivo y lo pone en modo normal.

REINIT.- Reinicio del Modulo, en caso de mal funcionamiento.

Codigos de estado:

11 => Teléfono detectado

12 => Falló envio de datos

13 => No se detecta conexión con teléfono

14 => Definir numero de telefono

21 => La contraseña es inválida

31 => Encender

32 => Reiniciar

```
Inclusión "Tn84def.Inc"        ; Definimos Librerías para el Microcontrolador a ser utilizado
;TX_OUT_TO_DEBUG
;#Defina SERIAL_DEBUG 1
;;#Define DEBUG_MESSAGE_COUNT_ENABLE 1
;#Define DEBUG_KEEP_INVALID_MESSAGES 1
;#Defina DEBUG_SEND_MESSAGES 1
#Defina RESET_PHONE_PERIODICALLY 1
.Equ      = 208 de BAUD_RATE_COUNTER; 4800 baudio
.Equ      = 8 de INIT_RECEIVE_TIMEOUT; el tiempo muerto breve para ATE0 (2 segundos)
.Equ      = 20 de DEFAULT_RECEIVE_TIMEOUT
.Equ      = de SMS_SEND_RECEIVE_TIMEOUT que 120; el tiempo muerto para SMS envía (24 segundos)
.Equ      = 900 de DEFAULT_PHONE_POLL_INTERVAL;
.Equ      = 18 de DEFAULT_FOUR_SAT_WAIT_TIME; 72 segundos (x4)
.Equ      DEFAULT_GPS_FIX_WAIT_TIME = 34; 136 segundos (x4)
.Equ      = de DEFAULT_BLOCKED_GPS_FIX_WAIT_TIME 24; 96 segundos
.Equ      = 0 de DEFAULT_SPEED_LIMIT; dejar incapacitado
.Equ      = de DEFAULT_TRACK_STOPPED_FIX_INTERVAL 120; dos minutos
.Equ      = 2 de DEFAULT_TRACK_STOPPED_NOTIFY_DELAY; 2 intervalos de busqueda
.Equ      = de DEFAULT_TRACK_BLOCKED_FIX_INTERVAL 600; diez minutos
.Equ      = 2 de DEFAULT_TRACK_BLOCKED_NOTIFY_DELAY; 2 intervalos de busqueda
.Equ      = de DEFAULT_TRACK_MOVING_FIX_INTERVAL 120; dos minutos
.Equ      = 5 de DEFAULT_TRACK_MOVING_NOTIFY_FREQ; 5 intervalos de busqueda
.Equ      = de DEFAULT_POWERSAVE_PHONE_OFF_INTERVAL 3600; una hora
.Equ      = de DEFAULT_POWERSAVE_PHONE_ON_INTERVAL 600; 10 minutos
.Equ      = 200 de DEFAULT_TRACK_MIN_DISPLACEMENT;0.2 minuto de lat / lon
.Equ      = 10 de GPS_LINES_AFTER_FIX;
.Equ      = 100 de PHONE_RESET_INTERVAL
.Equ      = 45 de PHONE_RESET_WAKEUP_DELAY
.Equ      = 160 de PHONE_MAX_CHECKSUM_LENGTH
.Equ      TX_OUT_TO_PHONE        = PORTA1
.Equ      RX_IN_FROM_PHONE = PORTA0
```

```
.Equ     RX_IN_FROM_GPS = PORTA2
.Equ     TX_OUT_TO_DEBUG = PORTA7
.Equ     LED_OUT_RED = PORTB0
.Equ     LED_OUT_GREEN = PORTB1
.Equ     GPS_POWER_OUT = PORTB2
.Equ     REINIT_CHECK_IN = PORTA4
.Equ     REINIT_CHECK_OUT = PORTA6
.Equ     = 5 de MAX_PHONE_ERRORS; AT que el mando vuelve a procesar
.Derrotar FLAG_REGISTER = r16
.Equ     = 0 de FLAG_SERIAL_SEND; compuesto
.Equ     FLAG_PARSE_NMEA = 2; set poner PARSE_LINE en el modo de NMEA
.Equ     = 3 de FLAG_SKIP_REPEATED_DELIMITER; el indicador usado en PARSE_LINE
.Equ     FLAG_GOT_RECEIVED_MESSAGE_NUMBER = 4; demuestra + que CMGL captó
.Equ     = 5 de FLAG_NEED_TO_POLL_MESSAGES; CMTI recibido, la interrupción, etcétera.
.Equ     = 6 de FLAG_PHONE_FULL; la necesidad de limpiar el espacio en teléfono
.Equ     = 7 de FLAG_MESSAGE_SENT; mensaje enviado
Derrotar  REG_ZERO = r0; registro siempre cero;
.Derrotar SERIAL_PORT = r1; bit determinado que corresponde a la línea de E/S que usted quiere usar.
.Derrotar SERIAL_DATA = r2; el registro de Shift para byte de datos.
.Derrotar SERIAL_COUNT = r17; los recuentos que las partes descendentes enviaron / recibieron.
.Derrotar SERIAL_POINTER = r3; Pointer respecto a la serie 256 byte recibe la memoria intermedia.
.Derrotar SERIAL_SCRATCH = r4; el registro de Scratch utilizado por la E/S por entregas
.Derrotar SERIAL_INTERRUPT_SREG = r5; el almacenamiento para banderas de estado durante la interrupción
por entregas
.Derrotar TIMER_INTERRUPT_SREG = r6; el almacenamiento para banderas de estado durante interrupción de
reloj automático
.Derrotar La demora = r23; este registro es contada por una interrupción 4x / seca, parando en el cero
.Derrotar RETRY_COUNT = r9; el recuento de error para las operaciones por entregas
; Asignaciones de registro de analizador sintáctico
; Y tiene puntero en la memoria intermedia
; El indicador de T es el error
```

```
.Derrotar El carácter = r18; la calidad actual
.Derrotar Pat = r7; calidad de dibujo actual
.Derrotar La base = r8; la base de  que es combinado
.Derrotar R10 de = de suma de control
.Derrotar DELIM = r11
; 26-27, 28-29, 30-31 es punteros
.Derrotar SCRATCH0 = r19
.Derrotar SCRATCH1 = r20
.Derrotar SCRATCH2 = r21
.Derrotar SCRATCH3 = r22
.Derrotar READ_FIELD = r19
.Derrotar PARSE_FIELD = r20
.Derrotar FIELD_LENGTH = r21
.Derrotar VERIFY_CHECKSUM = r12
#Ifdef     SERIAL_DEBUG
.Derrotar DEBUG_POINTER = r13
#Endif
; Las claves de estado devolver por GPS_LOCATE
.Equ       = de GPS_STATUS_DATASTREAM_TIMEOUT 0
.Equ       = de GPS_STATUS_FIX_TIMEOUT 1
.Equ       = de GPS_STATUS_FIX_STOPPED 2
.Equ       = de GPS_STATUS_FIX_MOVING 3
; Rastreador valores de estado federal
.Equ       = de TRACK_STATE_OFF 0
.Equ       = de TRACK_STATE_BLOCKED 1
.Equ       = de TRACK_STATE_STOPPED 2
.Equ       = de TRACK_STATE_MOVING 3
.Equ       = de TRACK_STATE_INITIAL 4
.Equ       = 5 de TRACK_STATE_POWERSAVE; el valor de powersave más bajo
.Equ       = de TRACK_STATE_POWERSAVE_PHONE_OFF 5
.Equ       = de TRACK_STATE_POWERSAVE_PHONE_ON 6
```

```
; Descifre los modos
.Equ    = de DECODE_SETTING_TWO_BYTE 7
.Equ    = de DECODE_SETTING_ONE_BYTE 6
.Equ    = de DECODE_SETTING_ONE_BYTE_X4 5
.Equ    = de EEADR_GPS_FOUR_SAT_WAIT_TIME 0
.Equ    = 1 de EELEN_GPS_FOUR_SAT_WAIT_TIME;
.Equ    = EEADR_GPS_FOUR_SAT_WAIT_TIME + EELEN_GPS_FOUR_SAT_WAIT_TIME de
EEADR_GPS_FIX_WAIT_TIME
.Equ    = 1 de EELEN_GPS_FIX_WAIT_TIME;
.Equ    = EEADR_GPS_FIX_WAIT_TIME + EELEN_GPS_FIX_WAIT_TIME de
EEADR_BLOCKED_GPS_FIX_WAIT_TIME
.Equ    = 1 de EELEN_BLOCKED_GPS_FIX_WAIT_TIME;
.Equ    EEADR_SPEED_LIMIT = EEADR_BLOCKED_GPS_FIX_WAIT_TIME + \
            EELEN_BLOCKED_GPS_FIX_WAIT_TIME
.Equ    = 1 de EELEN_SPEED_LIMIT; en unidades seleccionadas
.Equ    EEADR_GPS_MOVING_THRESHOLD = EEADR_SPEED_LIMIT + \
                EELEN_SPEED_LIMIT
.Equ    = de EELEN_GPS_MOVING_THRESHOLD 1; en unidades seleccionadas * 10
.Equ    EEADR_GPS_MOVING_RETRY_THRESHOLD = EEADR_GPS_MOVING_THRESHOLD + \
                    EELEN_GPS_MOVING_THRESHOLD
.Equ    = de EELEN_GPS_MOVING_RETRY_THRESHOLD 1; en unidades seleccionadas * 10
.Equ    EEADR_TRACK_STOPPED_FIX_INTERVAL = EEADR_GPS_MOVING_RETRY_THRESHOLD
+ \
                    EELEN_GPS_MOVING_RETRY_THRESHOLD
.Equ    = 2 de EELEN_TRACK_STOPPED_FIX_INTERVAL;
.Equ    EEADR_TRACK_STOPPED_NOTIFY_DELAY = EEADR_TRACK_STOPPED_FIX_INTERVAL +
\
                    EELEN_TRACK_STOPPED_FIX_INTERVAL
.Equ    EELEN_TRACK_STOPPED_NOTIFY_DELAY que = 1;
```

```
.Equ    EEADR_TRACK_BLOCKED_FIX_INTERVAL = EEADR_TRACK_STOPPED_NOTIFY_DELAY
+ \
                        EELEN_TRACK_STOPPED_NOTIFY_DELAY
.Equ    = 2 de EELEN_TRACK_BLOCKED_FIX_INTERVAL; cuán a menudo intentar la  generales posición
when bloquear
.Equ    EEADR_TRACK_BLOCKED_NOTIFY_DELAY = EEADR_TRACK_BLOCKED_FIX_INTERVAL
+ \
                        EELEN_TRACK_BLOCKED_FIX_INTERVAL
.Equ    EELEN_TRACK_BLOCKED_NOTIFY_DELAY que = 1; cuántos intervalos después bloquear to
notificar y usar
                        ; Intervalo de  bloquear - modo
.Equ    EEADR_TRACK_MOVING_FIX_INTERVAL = EEADR_TRACK_BLOCKED_NOTIFY_DELAY
+ \
                        EELEN_TRACK_BLOCKED_NOTIFY_DELAY
.Equ    = 2 de EELEN_TRACK_MOVING_FIX_INTERVAL;
.Equ    EEADR_TRACK_MOVING_NOTIFY_FREQ = EEADR_TRACK_MOVING_FIX_INTERVAL + \
                        EELEN_TRACK_MOVING_FIX_INTERVAL
.Equ    EELEN_TRACK_MOVING_NOTIFY_FREQ = 1
.Equ    EEADR_POWERSAVE_PHONE_OFF_INTERVAL = EEADR_TRACK_MOVING_NOTIFY_FREQ
+ \
                        EELEN_TRACK_MOVING_NOTIFY_FREQ
.Equ    = 2 de EELEN_POWERSAVE_PHONE_OFF_INTERVA
.Equ    EEADR_POWERSAVE_PHONE_ON_INTERVAL =
EEADR_POWERSAVE_PHONE_OFF_INTERVAL + \
                        EELEN_POWERSAVE_PHONE_OFF_INTERVAL
.Equ    = 2 de EELEN_POWERSAVE_PHONE_ON_INTERVAL;

.Equ    EEADR_TRACK_MIN_DISPLACEMENT = EEADR_POWERSAVE_PHONE_ON_INTERVAL + \
                EELEN_POWERSAVE_PHONE_ON_INTERVAL
.Equ    = 2 de EELEN_TRACK_MIN_DISPLACEMENT;
```

```
.Equ    EEADR_PHONE_NUMBER = EEADR_TRACK_MIN_DISPLACEMENT + \
                EELEN_TRACK_MIN_DISPLACEMENT
.Equ    = de EELEN_PHONE_NUMBER 32
.Equ    = EEADR_PHONE_NUMBER + EELEN_PHONE_NUMBER de EEADR_EMAIL_ADDR
.Equ    = de EELEN_EMAIL_ADDR 64
.Equ    = EEADR_EMAIL_ADDR + EELEN_EMAIL_ADDR de EEADR_PASSWORD
.Equ    = de EELEN_PASSWORD 16
.Equ    = EEADR_PASSWORD + EELEN_PASSWORD de EEADR_DEVICENAME
.Equ    = de EELEN_DEVICENAME 16
.Equ    = de EEADR_CHECKCODE 510
.Equ    = de EELEN_CHECKCODE 2
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; Direcciones de RAM estática
.Dseg
CMGL_SAFETY_COUNT.1 de byte;
WDR_COUNT.Byte que 1;
SEND_CHECKSUM.1 de byte;
SEND_CHECKSUM_COUNTER;
SEND_CHECKSUM_BYTES.
#DEBUG_MESSAGE_COUNT_ENABLE de ifdef
DEBUG_MESSAGE_COUNT.2 de byte;
#Endif
#RESET_PHONE_PERIODICALLY de ifdef
PHONE_RESET_COUNT.Byte 1
#Endif
#DEBUG_KEEP_INVALID_MESSAGES de ifdef
DEBUG_INVALID_MESSAGE_FLAG.Byte 1
#Endif
SERIAL_SEND_DELAY.1 de byte; la cantidad de bit mide la duración por byte para la serie, minuto es 11
RECEIVE_TIMEOUT.1 de byte; la demora para recibir los datos
PHONE_STRING.Byte 2
```

```
SEND_STRING_OFFSET_0.Byte 1
SEND_STRING_OFFSET_1.Byte 1
CURRENT_SMS.Byte 4
COMMAND_PENDING.Byte 2
GPS_TIMER.1 de byte; recuentos up si volver a poner al cero, a un recuento por 4 segundos, a los altos en 0xff
GPS_STATUS.Byte 1
SUBSECONDS_COUNT.Byte 1
BLINK_CODES.Byte 8
BLINK_CODES_INDEX.1 de byte; halfbyte alto - orden es el puntero de Insertar.  La bajo - orden es el puntero de función.
TRACK_STATE.Byte 1
TRACK_COUNT.¡1 de byte; ¡recuentos up when GPS_STATUS!= TRACK_STATE hasta el cambio estatal
CHECKSUM_RETRY_COUNT.Byte 1
MOVING_LOCATE_COUNT.1 de byte; recuentos up para notificar mientras conmovedor
POLL_INTERVAL.2 de byte; la demora entre requerimientos de teléfono /  generales
FASTEST_SPEED.2 de byte; velocidad más rápida por encima del límite de velocidad durante esta sesión de agrupación según rendimiento
.Equ       = 27 de LAST_GOOD_FIX_LEN; FLDLEN_GPRMC_TIME + FLDLEN_GPRMC_LATITUDE + \
;                  FLDLEN_GPRMC_LATDIR + FLDLEN_GPRMC_LONGITUDE + \
;                  FLDLEN_GPRMC_LONGDIR
LAST_GOOD_FIX.LAST_GOOD_FIX_LEN de byte

; Ajustes, y s de NMEA
.Equ       = de PARSE_FIELDS_LENGTH 64
PARSE_FIELDS.PARSE_FIELDS_LENGTH de byte
;DEBUG_LAST_BYTE.Byte 1
RECEIVE_BUFFER.Byte    256
; Compensaciones respecto a la memoria intermedia de LAST_GOOD_FIX
.Equ       LAST_GOOD_FIX_TIME = LAST_GOOD_FIX
.Equ       LAST_GOOD_FIX_LATITUDE = LAST_GOOD_FIX + 6
.Equ       LAST_GOOD_FIX_LATDIR = LAST_GOOD_FIX_LATITUDE + 9
```

```
.Equ	LAST_GOOD_FIX_LONGITUDE = LAST_GOOD_FIX_LATDIR + 1
.Equ	LAST_GOOD_FIX_LONGDIR = LAST_GOOD_FIX_LONGITUDE + 10
; Campo para descomponer gramaticalmente DEVICENAME
.Equ	FLDADR_DEVICENAME = PARSE_FIELDS + 0
.Equ	= de FLDLEN_DEVICENAME 16
; Campo para descomponer gramaticalmente CMGL
.Equ	FLDADR_CMGL_UNSENT = PARSE_FIELDS + 0
.Equ	FLDADR_CMGL_SENT = PARSE_FIELDS + 32
.Equ	FLDADR_CMGL_STOP = PARSE_FIELDS + 56
.Equ	= 8 de FLDLEN_CMGL_MAXMSGS; en realidad lo limita a 7 de cada uno
; Para y para descomponer gramaticalmente SETPASSWORD
.Equ	= de FLDADR_PASSWORD1 + de PARSE_FIELDS 0
.Equ	= de FLDLEN_PASSWORD1 16
.Equ	FLDADR_PASSWORD2 = + de PARSE_FIELDS 16
.Equ	= de FLDLEN_PASSWORD2 16
.Equ	= de MIN_PASSWORD_LENGTH 3
; Para descomponer gramaticalmente  generales producto
.Equ	FLDADR_GPRMC_FIXVALID = PARSE_FIELDS + 0
.Equ	= de FLDLEN_GPRMC_FIXVALID 1
.Equ	= FLDADR_GPRMC_FIXVALID + FLDLEN_GPRMC_FIXVALID de FLDADR_GPRMC_TIME
.Equ	= de FLDLEN_GPRMC_TIME 6
.Equ	= FLDADR_GPRMC_TIME + FLDLEN_GPRMC_TIME de FLDADR_GPRMC_LATITUDE
.Equ	= de FLDLEN_GPRMC_LATITUDE 9
.Equ	= FLDADR_GPRMC_LATITUDE + FLDLEN_GPRMC_LATITUDE de
FLDADR_GPRMC_LATDIR
.Equ	= de FLDLEN_GPRMC_LATDIR 1
.Equ	= FLDADR_GPRMC_LATDIR + FLDLEN_GPRMC_LATDIR de
FLDADR_GPRMC_LONGITUDE
.Equ	= de FLDLEN_GPRMC_LONGITUDE 10
.Equ	= FLDADR_GPRMC_LONGITUDE + FLDLEN_GPRMC_LONGITUDE de
FLDADR_GPRMC_LONGDIR
```

```
.Equ    = de FLDLEN_GPRMC_LONGDIR 1
.Equ    = FLDADR_GPRMC_LONGDIR + FLDLEN_GPRMC_LONGDIR de FLDADR_GPRMC_SPEED
.Equ    = de FLDLEN_GPRMC_SPEED 6
.Equ    = FLDADR_GPRMC_SPEED + FLDLEN_GPRMC_SPEED de FLDADR_GPRMC_COURSE
.Equ    = de FLDLEN_GPRMC_COURSE 6
.Equ    = FLDADR_GPRMC_COURSE + FLDLEN_GPRMC_COURSE de FLDADR_GPRMC_DATE
.Equ    = de FLDLEN_GPRMC_DATE 6
.Equ    = FLDADR_GPRMC_DATE + FLDLEN_GPRMC_DATE de FLDADR_GPGGA_FIXVALID
.Equ    = de FLDLEN_GPGGA_FIXVALID 1
.Equ    = FLDADR_GPGGA_FIXVALID + FLDLEN_GPGGA_FIXVALID de FLDADR_GPGGA_SATS
.Equ    = de FLDLEN_GPGGA_SATS 2
.Equ    = FLDADR_GPGGA_SATS + FLDLEN_GPGGA_SATS de FLDADR_GPGGA_ALT
.Equ    ¿= 5 de FLDLEN_GPGGA_ALT; demasiado largo?
.Equ    = FLDADR_GPGGA_ALT + FLDLEN_GPGGA_ALT de FLDADR_GPRMC_SPEED_NUMERIC
.Equ    = de FLDLEN_GPRMC_SPEED_NUMERIC 2
.Equ    = FLDADR_GPRMC_SPEED_NUMERIC + FLDLEN_GPRMC_SPEED_NUMERIC de
FLDADR_GPGGA_ALT_NUMERIC
.Equ    = de FLDLEN_GPGGA_ALT_NUMERIC 2
.Equ    FLDADR_GPS_LINE_COUNT_FLAGS = FLDADR_GPGGA_ALT_NUMERIC + \
                    FLDLEN_GPGGA_ALT_NUMERIC
.Equ    = de FLDLEN_GPS_LINE_COUNT_FLAGS 1
.Equ    = de GPS_FLAG_LOW_SPEED_RETRY 7
.Equ    = de GPS_FLAG_DISPLACED 6
; Para y para el mensaje de estado
.Equ    STATUSMSG_SPEED_LIMIT = PARSE_FIELDS + 0
.Equ    STATUSMSG_GPS_MOVING_THRESHOLD = PARSE_FIELDS + 2
.Equ    STATUSMSG_GPS_MOVING_RETRY_THRESHOLD = PARSE_FIELDS + 4
.Equ    STATUSMSG_TRACK_STOPPED_FIX_INTERVAL = PARSE_FIELDS + 6
.Equ    STATUSMSG_TRACK_STOPPED_NOTIFY_DELAY = PARSE_FIELDS + 8
.Equ    STATUSMSG_TRACK_BLOCKED_FIX_INTERVAL = PARSE_FIELDS + 10
.Equ    STATUSMSG_TRACK_BLOCKED_NOTIFY_DELAY = PARSE_FIELDS + 12
```

```
.Equ    STATUSMSG_TRACK_MOVING_FIX_INTERVAL = PARSE_FIELDS + 14
.Equ    STATUSMSG_TRACK_MOVING_NOTIFY_FREQ = PARSE_FIELDS + 16
.Equ    STATUSMSG_MIN_DISPLACEMENT = PARSE_FIELDS + 18
.Equ    STATUSMSG_POWERSAVE_PHONE_OFF_INTERVAL = PARSE_FIELDS + 20
.Equ    STATUSMSG_POWERSAVE_PHONE_ON_INTERVAL = PARSE_FIELDS + 22
.Equ    STATUSMSG_GPS_FOUR_SAT_WAIT_TIME = PARSE_FIELDS + 24
.Equ    STATUSMSG_GPS_FIX_WAIT_TIME = PARSE_FIELDS + 26
.Equ    STATUSMSG_BLOCKED_GPS_FIX_WAIT_TIME = PARSE_FIELDS + 28
.Equ    STATUSMSG_WDR_COUNT = PARSE_FIELDS + 30
.Equ    STATUSMSG_BATTERY_CHARGE = PARSE_FIELDS + 32
.Equ    STATUSMSG_SIGNAL_STRENGTH = PARSE_FIELDS + 34
; Para y para CPMS
.Equ    FLDADR_CPMS_USED = PARSE_FIELDS + 8
.Equ    FLDADR_CPMS_TOTAL = PARSE_FIELDS + 16
.Equ    FLDADR_CPMS_USED_NUMERIC = PARSE_FIELDS + 24
.Equ    FLDADR_CPMS_TOTAL_NUMERIC = PARSE_FIELDS + 32
.Cseg
; Vectores de interrupción
        Rjmp    V_RESET
        Rjmp    V_INT0
        Rjmp    V_PCINT0
        Rjmp    V_PCINT1
        Rjmp    V_WDT
        Rjmp    V_TIM1_CAPT
        Rjmp    V_TIM1_COMPA
        Rjmp    V_TIM1_COMPB
        Rjmp    V_TIM1_OVF
        Rjmp    V_TIM0_COMPA
        Rjmp    V_TIM0_COMPB
        Rjmp    V_TIM0_OVF
V_PCINT1:
```

```
V_TIM1_CAPT:
V_TIM1_COMPB:
V_TIM1_OVF:
V_TIM0_COMPA:
V_TIM0_COMPB:
        Cli
        Rjmp    V_INT0
V_RESET:
; Puntero de pila
        Ldi     SCRATCH0, alto (RAMEND)
        Ldi     SCRATCH1, bajo (RAMEND)
        Afuera  SPH, SCRATCH0
        Afuera  SPL, SCRATCH1
; Registros de Init
        Clr     REG_ZERO; se quedar cero
        Clr     FLAG_REGISTER
        Afuera  EECR, REG_ZERO; EEPM1 y EEPM0 deben estar despejados para la escritura atómica
        Sts     TRACK_STATE, REG_ZERO
        Sts     COMMAND_PENDING, REG_ZERO
        Sts     + 1 de COMMAND_PENDING, REG_ZERO
        Sts     BLINK_CODES_INDEX, REG_ZERO
; Configuración para la E/S por entregas
        Ldi     SCRATCH0, 0xff sobre el que - que ((1 < < RX_IN_FROM_PHONE) + (1 < <
RX_IN_FROM_GPS)) el hombre fuera por entregas dejan sin trabajo la máxima, y la vuelta tirón - aumentar
sobre líneas sin usar para reducir el poder
        Afuera  PORTA, SCRATCH0
#Ifdef  SERIAL_DEBUG
        Ldi     SCRATCH0, (1 < < TX_OUT_TO_PHONE) |(1 < < REINIT_CHECK_OUT) |(1 < <
TX_OUT_TO_DEBUG)
#Más
        Ldi     SCRATCH0, (1 < < TX_OUT_TO_PHONE) |(1 < < REINIT_CHECK_OUT)
```

```
#Endif
        Afuera  DDRA, SCRATCH0; puesto al producto
;
        Ldi     SCRATCH0, 0xff del que - que ((1 < < LED_OUT_RED) + (1 < < LED_OUT_GREEN));
        Ldi     SCRATCH0, (1 < < LED_OUT_RED) |(1 < < LED_OUT_GREEN) |(1 < <
GPS_POWER_OUT)
        Afuera  DDRB, SCRATCH0; permite productos entonces/luego
;
        Sbi     TIMSK1, OCIE1A; permite A de interrupción como el reloj automático por entregas
        Hacia dentro        SCRATCH0, GIMSK
        Ori     SCRATCH0, 1 < < PCIE0; permite la interrupción de Cambiar de  mundialmente
        Afuera  GIMSK, SCRATCH0
; Permitir reloj automático del guardián con el tiempo muerto de 8 segundos (asumir el nivel de seguridad1)
        Ldi     SCRATCH0, (1 < < WDP3) + (1 < < WDP0) + (1 < < WDE)
        Wdr
        Afuera  WDTCSR, SCRATCH0
; Active Timer0 para la interrupción de reloj
        Ldi     SCRATCH0, (1 < < CS02) + (1 < < CS00); prescaler / 1024
        Afuera  TCCR0B, SCRATCH0; 4x por segundo para 1MHz
        Ldi     SCRATCH0, 1 < < TOIE0; la interrupción de exceso
        Afuera  TIMSK0, SCRATCH0; permitir
; Fije la velocidad de bauds
        Ldi     SCRATCH0, alto (BAUD_RATE_COUNTER)
        Ldi     SCRATCH1, bajo (BAUD_RATE_COUNTER)
        Afuera  OCR1AH, SCRATCH0
        Afuera  OCR1AL, SCRATCH1
; Fije el cronometraje por entregas
        Ldi     SCRATCH0, 12
        Sts     SERIAL_SEND_DELAY, SCRATCH0
; Ajustes de baja potencia
        Sbi     ACSR, ACD; comparator de análogo saliendo
```

```
; Verifique los datos de EEPROM y, si faltante, cargue de los incumplimientos el archivo
        Ldi     XH, alto (EEADR_CHECKCODE)
        Ldi     XL, bajo (EEADR_CHECKCODE)
        Rcall   EEPROM_READWORD_X_TO_SCRATCH10
        Cpi     SCRATCH1'', m'
        Brne    INIT_EEPROM_BAD; no ya arreglar
        Cpi     SCRATCH0'', yo'
        Breq    INIT_EEPROM_DONE; ya arreglado
INIT_EEPROM_BAD:
        Ldi     SCRATCH0, 0x33; enciende y apaga EEPROM clave inicializar
        Rcall   SEND_BLINK_CODE
        Ldi     ZH, alto (EEPROM_DEFAULTS < <1)
        Ldi     ZL, bajo (EEPROM_DEFAULTS < <1)
INIT_EEPROM_LOOP1:
        Lpm     XH, + de Z; aborda duración de + alto (< < 2 )
        Tst     XH; los medios cero hechos
        Breq    INIT_EEPROM_DONE
        Lpm     XL, + de Z; Address bajos
        Mov     SCRATCH0, XH
        Andi    XH, 3; dos partes bajas
        Lsr     SCRATCH0
        Lsr     SCRATCH0; la duración
INIT_EEPROM_LOOP2:
        Lpm     El carácter, Z+
        Rcall   EEPROM_WRITEBYTE
        Dec     SCRATCH0
        Brne    INIT_EEPROM_LOOP2
        Rjmp    INIT_EEPROM_LOOP1
INIT_EEPROM_DONE:
        Rcall   CMD_TRACKOFF_ATSTART;
        Rcall   SAFE_DELAY
```

```
        Clr     SCRATCH0
        Clr     SCRATCH1
MAIN_REINIT_LOOP:
        Sbi     PORTA, REINIT_CHECK_OUT; fijó el cheque de reinit a gran altura
        Nop
        Sbis    PINA, REINIT_CHECK_IN; ve si la línea de contribución sigue
        Rjmp    MAIN_REINIT_JUMPER_NOT_PRESENT
        Cbi     PORTA, REINIT_CHECK_OUT; fijó el cheque de reinit bajo
        Nop
        Sbic    PINA, REINIT_CHECK_IN; ve si la línea de contribución sigue
        Rjmp    MAIN_REINIT_JUMPER_NOT_PRESENT
        Dec     SCRATCH0
        Brne    MAIN_REINIT_LOOP
        Wdr
        Tst     ¿SCRATCH1; primer pase?
        Brne    MAIN_REINIT_LOOP;
        Inc     SCRATCH1
        Ldi     SCRATCH0, 0x34; al que (capseq) traslada a saltador reinit
        Rcall   SEND_BLINK_CODE
        Rjmp    MAIN_REINIT_LOOP
MAIN_REINIT_JUMPER_NOT_PRESENT:
        Tst     SCRATCH1
        Breq    MAIN_NOT_REINIT
        Rcall   CLEAR_EEPROM_FLAG
        Rjmp    CLEAR_RAM_AND_REBOOT
MAIN_NOT_REINIT:
        Cbi     DDRA, REINIT_CHECK_OUT;
        Sbi     PORTA, REINIT_CHECK_OUT;
;
        Hacia dentro        SCRATCH1, MCUSR
```

```
        Sbrc    SCRATCH1, WDRF
        Rjmp    MAIN_WATCHDOG_RESET
MAIN_NOT_WATCHDOG:
        Sts     ¡CMGL_SAFETY_COUNT, REG_ZERO;
#Ifdef  RESET_PHONE_PERIODICALLY
        Sts     PHONE_RESET_COUNT, REG_ZERO
#Endif#DEBUG_MESSAGE_COUNT_ENABLE de ifdef
        Sts     DEBUG_MESSAGE_COUNT, REG_ZERO
        Sts     + 1 de DEBUG_MESSAGE_COUNT, REG_ZERO
#Endif
        Sts     WDR_COUNT, REG_ZERO; el recuento guardián volver a poner
        Ldi     SCRATCH0,'-'
        Ldi     XH, alto (LAST_GOOD_FIX)
        Ldi     XL, bajo (LAST_GOOD_FIX)
        Ldi     SCRATCH1, LAST_GOOD_FIX_LEN
MNW_CLGF_LOOP:
        St      X +, SCRATCH0
        Dec     SCRATCH1
        Brne    MNW_CLGF_LOOP
;
        Ldi     SCRATCH0, 0x31;
        Rjmp    MAIN_AFTER_WATCHDOG_CHECK
MAIN_WATCHDOG_RESET:
        Cbr     SCRATCH1, 1 < < WDRF
        Afuera  MCUSR, SCRATCH1; claro la bandera
        Lds     SCRATCH0, WDR_COUNT
        Inc     SCRATCH0
        Brne    MAIN_WDR_COUNTNOTOVF
        Dec     SCRATCH0; el do no envuelve al cero
MAIN_WDR_COUNTNOTOVF:
        Sts     WDR_COUNT, SCRATCH0
```

```
        Ldi     SCRATCH0, 0x32; el parpadeo que guardián de clave volvió a poner
MAIN_AFTER_WATCHDOG_CHECK:
        Rcall   SEND_BLINK_CODE; la puesta en marcha

#SERIAL_DEBUG de ifdef
        Ldi     ZH, alto (DEBUG_STARTUP_MESSAGE < <1)
        Ldi     ZL, bajo (DEBUG_STARTUP_MESSAGE < <1)
Rcall   SERIAL_SEND_STRING_TO_DEBUG
#Endif
;
;       Rcall   GPS_LOCATE; depure
;       Rcall   CMD_STATUS; depuren
;
; Encienda el teléfono si es apagado
#Ifdef  RESET_PHONE_PERIODICALLY
        Rcall   PHONE_RESET_IF_OFF
#Más
        Ldi     ZH, alto (PHONE_POWERON_STRING < <1)
        Ldi     ZL, bajo (PHONE_POWERON_STRING < <1)
        Rcall   DO_PHONE_OPERATION; suministre energía al teléfono on
        Ldi     Demora, (* de PHONE_RESET_WAKEUP_DELAY4)
        Rcall   SAFE_DELAY
#Endif

;; Notificar si WDR -
;       Lds     SCRATCH0, WDR_COUNT
;       Cpi     SCRATCH0, 1
;       Brne    MAIN_DEBUG_NOT_WDR
;       Ldi     SCRATCH0, PHONE_STRING_WATCHDOG_RESET - PHONE_STRING_BASE
;       Rcall   MULTI_SEND_DO
;MAIN_DEBUG_NOT_WDR:
```

```
;


; Bucle principal
        Lds     SCRATCH0, TRACK_STATE
        Cpi     SCRATCH0, TRACK_STATE_POWERSAVE
        Brsh    MAIN_TRK_SKIP; ninguna agrupación según rendimiento en el modo de powersave
        Cpi     SCRATCH0, TRACK_STATE_OFF
        Brne    MAIN_TRK_LOCATE
MAIN_TRK_SKIP:
Rjmp    MAIN_TRK_END
MAIN_TRK_LOCATE:


        Rcall   GPS_LOCATE; el resultado que la clave devolvió en SCRATCH0 y en GPS_STATUS
        Cpi     SCRATCH0, GPS_STATUS_FIX_STOPPED
        Lds     SCRATCH0, TRACK_STATE; no afecta banderas
        Breq    MAIN_TRK_GPS_STATUS_NOT_MOVING
        Brsh    MAIN_TRK_GPS_STATUS_MOVING
        Rjmp    MAIN_TRK_GPS_STATUS_BLOCKED
;
MAIN_TRK_GPS_STATUS_NOT_MOVING:
        Cpi     SCRATCH0, TRACK_STATE_STOPPED
        Breq    MAIN_TRK_CHECK_DISPLACED;  generales = parado, que = de TRACK_STATE paró
        Brsh    MAIN_TRK_TO_STATE_STOPPED;  generales = parado, TRACK_STATE = la mudanza
        Rjmp    MAIN_TRK_CHECK_DISPLACED;  generales = paró, = de TRACK_STATE bloqueó
;
MAIN_TRK_GPS_STATUS_MOVING:
        Cpi     SCRATCH0, TRACK_STATE_STOPPED
        Breq    MAIN_TRK_SEND_STARTED;  generales = la mudanza, = de TRACK_STATE parado
        Brsh    MAIN_TRK_CONTINUE_MOVING;  generales = la mudanza, TRACK_STATE = la
mudanza
        Rjmp    MAIN_TRK_SEND_STARTED;  generales = la mudanza, = de TRACK_STATE obstruido
```

```
;
MAIN_TRK_GPS_STATUS_BLOCKED:
        Cpi     SCRATCH0, TRACK_STATE_STOPPED
        Breq    MAIN_TRK_TO_STATE_BLOCKED;  generales = obstruido, que = de TRACK_STATE
paró
        Brsh    MAIN_TRK_TO_STATE_BLOCKED;  generales a quienes = bloqueó, TRACK_STATE =
la mudanza
        Rjmp    MAIN_TRK_END;  generales = bloqueó, = de TRACK_STATE bloqueó
;

MAIN_TRK_CHECK_DISPLACED:
        Lds     SCRATCH0, FLDADR_GPS_LINE_COUNT_FLAGS
        Sbrs    SCRATCH0, GPS_FLAG_DISPLACED
        Rjmp    MAIN_TRK_END
        Ldi     SCRATCH2, FIX_TYPE_MOVED - FIX_TYPE_BASE
        Rjmp    MAIN_TRK_SEND

MAIN_TRK_SEND_STARTED:
        Ldi     SCRATCH2, FIX_TYPE_STARTED - FIX_TYPE_BASE
        Rcall   MAIN_TRK_CHECK_SPEEDING
MAIN_TRK_SEND_MOVING:
        Ldi     SCRATCH3, TRACK_STATE_MOVING
        Ldi     XH, alto (EEADR_TRACK_MOVING_FIX_INTERVAL)
        Ldi     XL, bajo (EEADR_TRACK_MOVING_FIX_INTERVAL)
        Rjmp    MAIN_TRK_SAVE_AND_SEND

MAIN_TRK_TO_STATE_STOPPED:
        Lds     SCRATCH1, TRACK_COUNT
        Inc     SCRATCH1
        Sts     TRACK_COUNT, SCRATCH1
;
```

```
        Ldi     XH, alto (EEADR_TRACK_STOPPED_NOTIFY_DELAY)
        Ldi     XL, bajo (EEADR_TRACK_STOPPED_NOTIFY_DELAY)
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0
;
        Cp      SCRATCH1, SCRATCH0
        Brsh    MAIN_TTSS_SEND
        Rjmp    MAIN_TRK_END
MAIN_TTSS_SEND:
        Ldi     XH, alto (EEADR_TRACK_STOPPED_FIX_INTERVAL)
        Ldi     XL, bajo (EEADR_TRACK_STOPPED_FIX_INTERVAL)
        Ldi     SCRATCH2, FIX_TYPE_STOPPED - FIX_TYPE_BASE
        Ldi     SCRATCH3, TRACK_STATE_STOPPED
        Rjmp    MAIN_TRK_SAVE_AND_SEND
MAIN_TRK_TO_STATE_BLOCKED:
        Lds     SCRATCH1, TRACK_COUNT
        Inc     SCRATCH1
        Sts     TRACK_COUNT, SCRATCH1
;
        Ldi     XH, alto (EEADR_TRACK_BLOCKED_NOTIFY_DELAY)
        Ldi     XL, bajo (EEADR_TRACK_BLOCKED_NOTIFY_DELAY)
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0
;
        Cp      SCRATCH1, SCRATCH0
        Brsh    MAIN_TTSB_SEND
        Rjmp    MAIN_TRK_END
MAIN_TTSB_SEND:
        Ldi     XH, alto (EEADR_TRACK_BLOCKED_FIX_INTERVAL)
        Ldi     XL, bajo (EEADR_TRACK_BLOCKED_FIX_INTERVAL)
        Ldi     SCRATCH2, FIX_TYPE_STOPPED - FIX_TYPE_BASE
        Ldi     SCRATCH3, TRACK_STATE_BLOCKED
        Rjmp    MAIN_TRK_SAVE_AND_SEND
```

```
MAIN_TRK_CONTINUE_MOVING:
        Lds     SCRATCH1, MOVING_LOCATE_COUNT
        Inc     SCRATCH1
        Sts     MOVING_LOCATE_COUNT, SCRATCH1
        Sts     TRACK_COUNT, REG_ZERO
;
        Ldi     SCRATCH2, FIX_TYPE_MOVING - FIX_TYPE_BASE
        Ldi     XH, alto (EEADR_TRACK_MOVING_NOTIFY_FREQ)
        Ldi     XL, bajo (EEADR_TRACK_MOVING_NOTIFY_FREQ)
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0
;
        Lds     SCRATCH1, MOVING_LOCATE_COUNT
        Cp      SCRATCH1, SCRATCH0
        Brsh    MAIN_TRK_SEND_MOVING
        Rjmp    MAIN_TRK_END

; Pone FIX_TYPE_SPEEDING si el límite más arriba

MAIN_TRK_CHECK_SPEEDING:
        Mov     El carácter, SCRATCH2
;
        Ldi     XH, alto (EEADR_SPEED_LIMIT)
        Ldi     XL, bajo (EEADR_SPEED_LIMIT)
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0
        Tst     SCRATCH0
        Breq    MAIN_TRK_SPEEDING_OUT; si cero, límite de velocidad disabled
;
        Clr     SCRATCH1
        Rcall   MULX10_SCRATCH10
        Lds     SCRATCH3, FLDADR_GPRMC_SPEED_NUMERIC
        Lds     SCRATCH2, + 1 de FLDADR_GPRMC_SPEED_NUMERIC
```

```
;
        Cp      ¿SCRATCH0, SCRATCH2; más alto que salvar el límite de velocidad?
        Cpc     SCRATCH1, SCRATCH3
        Brsh    MAIN_TRK_SPEEDING_OUT; no ir a exceso de velocidad
;
        Ldi     El carácter, FIX_TYPE_SPEEDING - FIX_TYPE_BASE
;
        Lds     SCRATCH0, FASTEST_SPEED; visto hasta ahora
        Lds     SCRATCH1, + 1 de FASTEST_SPEED
        Cp      ¿SCRATCH0, SCRATCH2; se ir más rápido?
        Cpc     SCRATCH1, SCRATCH3
        Brsh    MAIN_TRK_SPEEDING_OUT
;
        Sts     FASTEST_SPEED, SCRATCH2
        Sts     + 1 de FASTEST_SPEED, SCRATCH3
        Rjmp    MAIN_TRK_SPEEDING_FASTEST; lleve set ya
MAIN_TRK_SPEEDING_OUT:
        Clc
MAIN_TRK_SPEEDING_FASTEST:
        Mov     SCRATCH2, el carácter
        Ret


MAIN_TRK_SAVE_AND_SEND:
        Sts     TRACK_COUNT, REG_ZERO
        Sts     MOVING_LOCATE_COUNT, REG_ZERO
        Sts     TRACK_STATE, SCRATCH3
        Rcall   EEPROM_READWORD_X_TO_SCRATCH10
        Sts     POLL_INTERVAL, SCRATCH1
        Sts     + 1 de POLL_INTERVAL, SCRATCH0
MAIN_TRK_SEND:
        Rcall   GPS_SEND_LOCATION
```

```
MAIN_TRK_END:


MAIN_POLL_PHONE:


        Ldi     SCRATCH0, 0x11; enciende y apaga el sondeo de clave
        Rcall   SEND_BLINK_CODE
        Cbr     FLAG_REGISTER, (1 < < FLAG_GOT_RECEIVED_MESSAGE_NUMBER) |(1 < <
FLAG_NEED_TO_POLL_MESSAGES) |(1 < < FLAG_MESSAGE_SENT) |(1 < < FLAG_PHONE_FULL)
        Ldi     ZH, alto (PHONE_INIT_POLL_STRING < <1)
        Ldi     ZL, bajo (PHONE_INIT_POLL_STRING < <1)
        Rcall   DO_PHONE_OPERATION
        Sbrc    FLAG_REGISTER, FLAG_GOT_RECEIVED_MESSAGE_NUMBER
        Sbr     FLAG_REGISTER, 1 < < FLAG_NEED_TO_POLL_MESSAGES;
        Brtc    MAIN_AFTER_CHECK_FAILED
        Ldi     SCRATCH0, 0x13;
        Rcall   SEND_BLINK_CODE
MAIN_AFTER_CHECK_FAILED:
;
#Ifdef  SERIAL_DEBUG
        Ldi     ZH, alto (SPACE_USED_MESSAGE < <1)
        Ldi     ZL, bajo (SPACE_USED_MESSAGE < <1)
        Rcall   SERIAL_SEND_STRING_TO_DEBUG
#Endif
;
#Ifdef  RESET_PHONE_PERIODICALLY
        Lds     SCRATCH0, PHONE_RESET_COUNT;
        Tst     SCRATCH0; si un mensaje ha sido recibido
        Breq    MAIN_SKIP_INC_PHONE_RESET;
        Inc     SCRATCH0; no hace increment
        Sts     PHONE_RESET_COUNT, SCRATCH0
MAIN_SKIP_INC_PHONE_RESET:
```

```
#Endif
;
        Sbrs    FLAG_REGISTER, FLAG_GOT_RECEIVED_MESSAGE_NUMBER
        Rjmp    MAIN_NO_MESSAGE
;
        Lds     SCRATCH0, CMGL_SAFETY_COUNT
        Cpi     SCRATCH0, 4
        Brsh    MAIN_FAULT_SKIP_READ;
#Ifdef  DEBUG_KEEP_INVALID_MESSAGES
        Sts     DEBUG_INVALID_MESSAGE_FLAG, REG_ZERO
#Endif
        Ldi     ZH, alto (PHONE_READ_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_READ_MESSAGE_STRING < <1)
        Rcall   DO_PHONE_OPERATION
;
MAIN_FAULT_SKIP_READ:
        Ldi     ZH, alto (PHONE_DELETE_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_DELETE_MESSAGE_STRING < <1)
#Ifdef  DEBUG_KEEP_INVALID_MESSAGES
        Lds     SCRATCH0, DEBUG_INVALID_MESSAGE_FLAG
        Tst     SCRATCH0
        Brne    MAIN_NO_DELETE
#Endif
        Rcall   DO_PHONE_OPERATION
MAIN_NO_DELETE:
        Cbr     FLAG_REGISTER, 1 < < FLAG_GOT_RECEIVED_MESSAGE_NUMBER; o envío
fallarán
        Brts    MAIN_FAULT_NO_CLEAR;
        Sts     CMGL_SAFETY_COUNT, REG_ZERO
MAIN_FAULT_NO_CLEAR:
;
```

```
#Ifdef    RESET_PHONE_PERIODICALLY
          Lds       SCRATCH0, PHONE_RESET_COUNT
          Inc       SCRATCH0; teléfono recuento vuelto a poner de inc después de recibir un mensaje
          Sts       PHONE_RESET_COUNT, SCRATCH0
#Endif
;
;
MAIN_NO_MESSAGE:
;
;         Sbrc      FLAG_REGISTER, FLAG_PHONE_FULL
          Rcall     MSD_CLEAR_SENT_AND_SEND_PENDING; limpia FLAG_PHONE_FULL
;
; Comande pendiente
          Lds       ZH, COMMAND_PENDING
          Lds       ZL, COMMAND_PENDING + 1
          Tst       ZH
          Brne      MAIN_HAVE_COMMAND_PENDING
          Tst       ZL
          Breq      MAIN_AFTER_COMMAND_PENDING
MAIN_HAVE_COMMAND_PENDING:
          Sts       COMMAND_PENDING, REG_ZERO
          Sts       + 1 de COMMAND_PENDING, REG_ZERO
          Icall; Z
MAIN_AFTER_COMMAND_PENDING:
;
#Ifdef    RESET_PHONE_PERIODICALLY
          Lds       SCRATCH0, PHONE_RESET_COUNT
          Cpi       SCRATCH0, PHONE_RESET_INTERVAL; la cantidad de / de requerimientos que los
mensajes antes volvieron a poner
          Brlo      MAIN_NO_PHONE_RESET
          Rcall     PHONE_RESET
```

```
        Rjmp    MAIN_POLL_PHONE; la busqueda arreglar teléfono otra vez
MAIN_NO_PHONE_RESET:
#Endif
;
Ldi     SCRATCH1, BAUD_RATE_COUNTER / 9; espera impedir retriggering aquí
MAIN_DONE_PAUSE1:
        Clr     SCRATCH0
MAIN_DONE_PAUSE2:
        Dec     SCRATCH0
        Brne    MAIN_DONE_PAUSE2; espera tiempo un caracter (1MHz)
        Dec     SCRATCH1
        Brne    MAIN_DONE_PAUSE1
        Rcall   SERIAL_RECEIVE_DISABLE
;
MAIN_FINISH_BLINKING_WDR:
        Wdr; si interrupciones no están respondiendo, wdr no será declarado repetidamente
        Mov     SCRATCH2, SCRATCH0
MAIN_FINISH_BLINKING:
        Lds     SCRATCH0, BLINK_CODES_INDEX
        Mov     SCRATCH1, SCRATCH0
        Intercambie     SCRATCH1
        Cp      SCRATCH0, SCRATCH1
        Breq    MAIN_FINISHED_BLINKING  Cp      SCRATCH0, SCRATCH2
        Brne    MAIN_FINISH_BLINKING_WDR
        Rjmp    MAIN_FINISH_BLINKING
MAIN_FINISHED_BLINKING:
;
        Sbrc    FLAG_REGISTER, FLAG_NEED_TO_POLL_MESSAGES
        Rjmp    MAIN_POLL_PHONE; busque los mensajes otra vez
MAIN_SLEEP:
#SERIAL_DEBUG de ifdef
```

```
        Ldi     SCRATCH0, (1 < < TX_OUT_TO_PHONE) |(1 < < TX_OUT_TO_DEBUG)
        Ldi     ZH, alto (POLL_INTERVAL_MESSAGE < <1)
        Ldi     ZL, bajo (POLL_INTERVAL_MESSAGE < <1)
        Rcall   SERIAL_SEND_STRING_TO_DEBUG
        Rcall   SERIAL_RECEIVE_DISABLE
#Endif
        Lds     SCRATCH2, POLL_INTERVAL
        Lds     SCRATCH1, + 1 de POLL_INTERVAL; está en segundo
        Wdr     ; Reloj automático de inicio en el alto
;
MAIN_BACK_TO_SLEEP:
        Hacia dentro    SCRATCH0, WDTCSR
        Sbr     SCRATCH0, 1 < < WDIE; permite la interrupción del guardián, esto get limpiar cuando
ocurre
Afuera  WDTCSR, SCRATCH0
;
Ldi     SCRATCH0, (1 < < el SE) + (1 < < SM1)
        Afuera  MCUCR, SCRATCH0; permite el modo de Reposo
        Duerma
        Sbci    SCRATCH2, 0
        Brcc    ¿MAIN_BACK_TO_SLEEP
        Lds     SCRATCH0, TRACK_STATE
        Cpi     SCRATCH0, TRACK_STATE_POWERSAVE_PHONE_OFF
        Brne    MAIN_PHONE_WAS_NOT_OFF
        Ldi     SCRATCH0, 0xff sobre el que - que ((1 < < RX_IN_FROM_PHONE) + (1 < <
RX_IN_FROM_GPS))
        Afuera  PORTA, SCRATCH0
#Ifdef  SERIAL_DEBUG
        Ldi     SCRATCH0, (1 < < TX_OUT_TO_PHONE) |(1 < < TX_OUT_TO_DEBUG)
#Más
        Ldi     SCRATCH0, 1 < < TX_OUT_TO_PHONE
```

```
#Endif
        Afuera  DDRA, SCRATCH0;
#Ifdef  RESET_PHONE_PERIODICALLY
        Rcall   PHONE_RESET
#Más
        Ldi     ZH, alto (PHONE_POWERON_STRING < <1)
        Ldi     ZL, bajo (PHONE_POWERON_STRING < <1)
        Rcall   DO_PHONE_OPERATION; suministre energía al teléfono on
        Ldi     Demora, (* de PHONE_RESET_WAKEUP_DELAY4)
        Rcall   SAFE_DELAY
#Endif
        Ldi     SCRATCH0, TRACK_STATE_POWERSAVE_PHONE_ON
        Sts     TRACK_STATE, SCRATCH0
        Ldi     XH, alto (EEADR_POWERSAVE_PHONE_ON_INTERVAL)
        Ldi     XL, bajo (EEADR_POWERSAVE_PHONE_ON_INTERVAL)
        Rcall   EEPROM_READWORD_X_TO_SCRATCH10
        Sts     POLL_INTERVAL, SCRATCH1
        Sts     + 1 de POLL_INTERVAL, SCRATCH0; intervalo de busqueda determinado
        Rjmp    MAIN_POLL_PHONE
MAIN_PHONE_WAS_NOT_OFF:
        Lds     SCRATCH0, TRACK_STATE
        Cpi     SCRATCH0, TRACK_STATE_POWERSAVE_PHONE_ON
        Brne    MAIN_NOT_PHONEOFF
        Ldi     ZH, alto (PHONE_POWEROFF_STRING < <1)
        Ldi     ZL, bajo (PHONE_POWEROFF_STRING < <1)
        Rcall   DO_PHONE_OPERATION
        Rcall   SERIAL_RECEIVE_DISABLE
        Afuera  PCMSK0, REG_ZERO
        Afuera  DDRA, REG_ZERO; deassert producto por entregas
        Ldi     SCRATCH0, 0xff - ((1 < < RX_IN_FROM_PHONE) + (1 < < RX_IN_FROM_GPS) + (1 <
        Ldi     SCRATCH0, TRACK_STATE_POWERSAVE_PHONE_OFF
```

```
        Sts     TRACK_STATE, SCRATCH0
;
        Ldi     XH, alto (EEADR_POWERSAVE_PHONE_OFF_INTERVAL)
        Ldi     XL, bajo (EEADR_POWERSAVE_PHONE_OFF_INTERVAL)
        Rcall   EEPROM_READWORD_X_TO_SCRATCH10
        Sts     POLL_INTERVAL, SCRATCH1
        Sts     + 1 de POLL_INTERVAL, SCRATCH0
        Rjmp    MAIN_SLEEP
;
MAIN_NOT_PHONEOFF:
MAIN_SLEEP_OUT:
        #Ifdef  RESET_PHONE_PERIODICALLY
PHONE_RESET_IF_OFF:
        Ldi     SCRATCH0, 3;
        Rjmp    PRS_E1
PHONE_RESET:
        Ldi     SCRATCH0, 3
        Empuje  SCRATCH0
        Rjmp    PRS_NOW
;
PRS_LOOP:
        Reviente SCRATCH0; vuelve a procesar el recuento
        Tst     SCRATCH0
        Brne    PRS_CONT
        Ret; vuelva a procesar el recuento expired
PRS_CONT:
        Dec     SCRATCH0
PRS_E1:
        Empuje  SCRATCH0
;
        Ldi     ZH, alto (PHONE_CHECK_STATUS_STRING < <1)
```

```
        Ldi     ZL, bajo (PHONE_CHECK_STATUS_STRING < <1)
        Sts     SEND_STRING_OFFSET_1, REG_ZERO; será puesto si CFUN: 0 de + de teléfono
        Rcall   DO_PHONE_OPERATION
        Lds     SCRATCH1, SEND_STRING_OFFSET_1
        Sbrs    SCRATCH1, 0; será 0xff si el teléfono se va
        Reviente SCRATCH0
        Sbrs    SCRATCH1, 0
        Ret
;
PRS_NOW:
        Ldi     ZH, alto (PHONE_RESET_STRING < <1)
        Ldi     ZL, bajo (PHONE_RESET_STRING < <1)
        Rcall   DO_PHONE_OPERATION;
        Adiw    ZH:ZL, 1
        GNC_WAITCHAR:
        Cp      ¿YL, SERIAL_POINTER; la calidad pendiente?
        Brne    GNC_HAVECHAR; sí
        Cp      Carácter, demórese; ¿ha contado regresivamente?
        Breq    GNC_NOTICK
        Wdr
        Mov     El carácter, la demora
GNC_NOTICK:
        Tst     Demora
        Brne    ¿GNC_WAITCHAR; tiempo up?
GNC_TIMEDOUT:
        Juego   ; Error de informe
        Ret
GNC_HAVECHAR:
        Ld      El carácter, Y+
        Ldi     YH, 1; cruzado
SERIAL_SEND_DONE:
```

```
        Wdr
        Ret
#Ifdef  SERIAL_DEBUG
SERIAL_SEND_STRING_TO_DEBUG:
        Ldi     SCRATCH0, 1 < < TX_OUT_TO_DEBUG
        Mov     SERIAL_PORT, SCRATCH0
        Rjmp    SERIAL_SEND_STRING
#Endif
SERIAL_SEND_STRING_TO_PHONE:
#Ifdef  SERIAL_DEBUG
        Ldi     SCRATCH0, (1 < < TX_OUT_TO_PHONE) |(1 < < TX_OUT_TO_DEBUG)
#Más
        Ldi     SCRATCH0, 1 < < TX_OUT_TO_PHONE
#Endif
        Mov     SERIAL_PORT, SCRATCH0
SERIAL_SEND_STRING:
        Lpm     SCRATCH0, + de Z; consigue un byte e puntero de increment
        Cpi     ¿SCRATCH0, 0xf7; la función especial?
        Brsh    SSS_SPECIAL; consiguió una función especial
        Tst     ¿SCRATCH0; final de ?
        Breq    SERIAL_SEND_DONE; ret
        Mov     SERIAL_DATA, SCRATCH0
        Rcall   SERIAL_SEND_BYTE
        Rjmp    SERIAL_SEND_STRING
SSS_SPECIAL:
        Cpi     SCRATCH0, 0xfd
        Brsh    SSS_FIELD; RAM estática o campo de EEPROM
        Cpi     SCRATCH0, 0xfa
        Brsh    SSS_SUBSTRING
        Cpi     SCRATCH0, 0xf8
        Brsh    SSS_DECIMAL
```

```
; Captura de suma de control
        Lds     SCRATCH0, SEND_CHECKSUM_COUNTER
        Sts     SEND_CHECKSUM_BYTES, SCRATCH0; la cantidad de la memoria de bytes
checksummed
        Sts     SEND_CHECKSUM_COUNTER, REG_ZERO
        Lds     VERIFY_CHECKSUM, SEND_CHECKSUM
        Sts     SEND_CHECKSUM, REG_ZERO
        Rjmp    SERIAL_SEND_STRING
SSS_DECIMAL:
        Lpm     XH, + de Z; máxima de dirección
        Lpm     XL, + de Z; nivel más bajo de dirección
        Ld      SCRATCH1, + de X; byte alto
        Ld      XL, X; byte bajo
        Mov     XH, SCRATCH1
        Andi    SCRATCH0, 1; aísla el entero / xxx.Bandera de x
        Rcall   PRINT_DECIMAL
        Rjmp    SERIAL_SEND_STRING; sin enteros de campo
SSS_SUBSTRING:
; Esto es 0xfa - 0xfc de llamada de subcadena
        Lpm     SCRATCH2, Z+
        Lpm     SCRATCH1, + de Z; consiguió dirección de base de la subcadena
        Empuje  ZH
        Empuje  ZL; memoria dirección llamado de subcadena anterior
        Lds     ZL, SEND_STRING_OFFSET_0
        Sbrc    SCRATCH0, 0; = 11111010 de 0xfa, = 11111011 de 0xfb
        Lds     ZL, SEND_STRING_OFFSET_1
        Sbrc    SCRATCH0, 2; = 11111100 de 0xfc
        Clr     ZL
        Clr     ZH
; Añada el offset a dirección de base y subcadena
        Lsl     ZL; debido a que el  está en el ROM debe ser palabra alinear
```

```
        Rol     ZH; duplicar el offset da el alcance 512 byte
        Añada   ZL, SCRATCH1
        Adc     ZH, SCRATCH2
        Rcall   SERIAL_SEND_STRING; envie la subcadena
        Reviente ZL
        Reviente ZH; consigue dirección vieja
        Rjmp    SERIAL_SEND_STRING; vuelva a este
SSS_FIELD:
        Clr     SCRATCH2; use demostrar si necesitamos un espacio a la cola
        Lpm     XH, Z+
        Lpm     XL, Z+
        Mov     SCRATCH1, XH; la duración de campo
        Andi    XH, 3; bits de orden superior de dirección
        Andi    SCRATCH1, 0xfc; quita dos partes bajas
        Mov     SCRATCH3, SCRATCH0; usado para EEPROM abajo ya que EEPROM leído sobreescribe
SCRATCH0
        Cpi     ¿SCRATCH0, 0xfe; campo de eeprom?
        Brsh    SSS_FIELD_EEPROM_LOOP
SSS_FIELD_SRAM_LOOP:
        Ld      SERIAL_DATA, X+
        Tst     SERIAL_DATA
        Breq    SSS_RJMP_SERIAL_SEND_STRING; ¡fuera! sobre la terminación nula
        Rcall   SERIAL_SEND_BYTE
        Subi    SCRATCH1, 4; campo la duración es * 4 atribuible a su puesto de bit
        Brne    SSS_FIELD_SRAM_LOOP
SSS_BACKTOTOP:
        Rjmp    SERIAL_SEND_STRING; sin enteros de campo
SSS_FIELD_EEPROM_LOOP:
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0; X de incrementos
        Mov     SERIAL_DATA, SCRATCH0
        Tst     SERIAL_DATA
```

```
        Breq    SSS_FIELD_EEPROM_DONE; ¡fuera! sobre la terminación nula
        Rcall   SERIAL_SEND_BYTE
        Sbr     SCRATCH2, 1; el espacio a la cola requerido
        Subi    SCRATCH1, 4; campo la duración es * 4 atribuible a su puesto de bit
        Brne    SSS_FIELD_EEPROM_LOOP
SSS_FIELD_EEPROM_DONE:
        Añada   SCRATCH2, SCRATCH3; si el tipo es 0xff y one or more bytes enviado,
        Brne    SSS_BACKTOTOP; el indicador de cero es puesto y ir a la zaga del espacio debe ser enviado
        Ldi     SCRATCH2,' '
        Mov     SERIAL_DATA, SCRATCH2
        Rcall   SERIAL_SEND_BYTE
SSS_RJMP_SERIAL_SEND_STRING:

        Rjmp    SERIAL_SEND_STRING; sin enteros de campo

; Envie un byte en SERIAL_DATA
; Puso SERIAL_SEND_DELAY to por lo menos11. la cantidad más alta da deadtime pre- datos.
SERIAL_SEND_BYTE_FROM_CHAR:
        Mov     SERIAL_DATA, el carácter
SERIAL_SEND_BYTE:
        Empuje  SCRATCH0
        Lds     SCRATCH0, SEND_CHECKSUM_COUNTER
        Cpi     SCRATCH0, PHONE_MAX_CHECKSUM_LENGTH
        Brsh    SERIAL_SEND_BYTE_NO_CHECKSUM
        Inc     SCRATCH0
        Sts     SEND_CHECKSUM_COUNTER, SCRATCH0
        Lds     SCRATCH0, SEND_CHECKSUM
        Eor     SCRATCH0, SERIAL_DATA
        Sts     SEND_CHECKSUM, SCRATCH0
SERIAL_SEND_BYTE_NO_CHECKSUM:
        Lds     SERIAL_COUNT, SERIAL_SEND_DELAY
```

```
        Sbr     FLAG_REGISTER, 1 < < FLAG_SERIAL_SEND
        Cbr     FLAG_REGISTER, 1 < < FLAG_SERIAL_RECEIVE
CTC
        Afuera  TCCR1B, SCRATCH0; pone en marcha el reloj por entregas
SERIAL_SEND_BYTE_WAIT:
        Tst     SERIAL_COUNT
        Brne    SERIAL_SEND_BYTE_WAIT
        Reviente SCRATCH0
        Ret
SERIAL_RECEIVE_ENABLE_FROM_GPS:
        Ldi     SCRATCH0, 1 < < RX_IN_FROM_GPS
        Mov     SERIAL_PORT, SCRATCH0
        Rjmp    SERIAL_RECEIVE_ENABLE
SERIAL_RECEIVE_ENABLE_FROM_PHONE:
        Ldi     SCRATCH0, 1 < < RX_IN_FROM_PHONE
        Mov     SERIAL_PORT, SCRATCH0
SERIAL_RECEIVE_ENABLE:
        Cbr     FLAG_REGISTER, 1 < FLAG_SERIAL_SEND de <; selecto reciben el modo
        Sbr     FLAG_REGISTER, 1 < FLAG_SERIAL_RECEIVE de <; selecto reciben el modo
        Afuera  PCMSK0, SERIAL_PORT;
SERIAL_RECEIVE_DISABLE:
        Cli     ;
        Ldi     SCRATCH0, 1 < < RX_IN_FROM_PHONE
        Mov     SERIAL_PORT, SCRATCH0
        Cbr     FLAG_REGISTER, (1 < < FLAG_SERIAL_RECEIVE) |(1 < < FLAG_SERIAL_SEND)
        Afuera  PCMSK0, SERIAL_PORT
        Afuera  TCCR1B, SERIAL_SCRATCH; pare el reloj por entregas
V_TIM1_COMPA:
        Hacia dentro    SERIAL_INTERRUPT_SREG, SREG
        Sbrs    FLAG_REGISTER, FLAG_SERIAL_SEND
        Rjmp    SERIAL_RECEIVE_INTERRUPT
```

```
;
SERIAL_SEND_INTERRUPT:
        Breq    SERIAL_SEND_START_BIT
        Brcc    SERIAL_DEC_AND_RET
SERIAL_SEND_DATA_BIT:
        Ror     SERIAL_DATA; deje caer bit bajo en indicador de alcance, 1 de alcance into bit de orden
superior
SERIAL_SEND_START_BIT:
        Hacia dentro    SERIAL_SCRATCH, PORTA; que IN y OR no cambian lleva el indicador
        O       SERIAL_SCRATCH, SERIAL_PORT; pone a 1 la parte para el puerto seleccionado
        Brcs    SERIAL_SEND_DATA_ONE
        Submarino       SERIAL_SCRATCH, SERIAL_PORT; si es uno 0, ponga bit to zero en el producto
SERIAL_SEND_DATA_ONE:
        Afuera  PORTA, SERIAL_SCRATCH
SERIAL_DEC_AND_RET:
        Dec     SERIAL_COUNT
        Brne    SERIAL_SEND_NOT_DONE
; Envío por entregas hecho
        Afuera  TCCR1B, REG_ZERO; pare el reloj por entregas
SERIAL_SEND_NOT_DONE:
        Afuera  SREG, SERIAL_INTERRUPT_SREG
        Reti
;
; Interrumpir por el primer bit (la demora larga después de que el inicio mordió)
SERIAL_RECEIVE_INTERRUPT:
        Hacia dentro    SERIAL_SCRATCH, PINA; recibe el registro
        Y       SERIAL_SCRATCH, SERIAL_PORT; aisle  de contribución activo
        Neg     SERIAL_SCRATCH; esto pone indicador de alcance a menos que el registro es cero
        Ror     SERIAL_DATA; haga rodar el bit de bandera de alcance into registro de cambio
        Dec     SERIAL_COUNT
        Brne    ¿SERIAL_RECEIVE_RETI; último bit?
```

```
; Último bit
        Afuera  TCCR1B, REG_ZERO; pare el reloj por entregas
        Empuje  ZH; se presta el puntero de Z
        Empuje  ZL
        Ldi     ZH, 1; la memoria intermedia es 0x100 - 0x1ff
        Mov     ZL, SERIAL_POINTER
        St      Z, SERIAL_DATA
        Inc     SERIAL_POINTER; tope contra el puntero después de guardar; el puntero demuestra next
byte
        Reviente ZL
        Reviente ZH
        Afuera  PCMSK0, SERIAL_PORT; permite la interrupción de Cambiar de  para next calidad
SERIAL_RECEIVE_RETI:
        Afuera  SREG, SERIAL_INTERRUPT_SREG
        Reti
; La interrupción de Cambiar de  que 0 use detectar el inicio mordido on recibe
V_PCINT0:
        Hacia dentro    SERIAL_INTERRUPT_SREG, SREG
        Hacia dentro    SERIAL_SCRATCH, PINA; recibe el registro
        Y       SERIAL_SCRATCH, SERIAL_PORT; aisle  de contribución activo
        Brne    PCINT0_IGNORE; mordido debe ser cero si cayendo el borde, por lo demás hacer caso
omiso
        Sbrs    FLAG_REGISTER, FLAG_SERIAL_RECEIVE; recibe el modo permitido?
        Rjmp    PCINT0_UNSOLICITED_MESSAGE; no
        Afuera  PCMSK0, SERIAL_SCRATCH; cerró la interrupción de Cambiar de  mientras recibir
        Sbc     SERIAL_SCRATCH, SERIAL_DATA; la resta completa para el byte alto
        Afuera  TCNT1H, SERIAL_SCRATCH; el byte alto primero
        Afuera  TCNT1L, SERIAL_COUNT; ahora el byte bajo

        Ldi     SERIAL_COUNT, (1 < < WGM12) |(1 < < CS10); el inicio que el por entregas registra
        Afuera  TCCR1B, SERIAL_COUNT; usar Serial lo cuenta porque enlatamos LDI
```

```
        Ldi     SERIAL_COUNT, 8; póngase listo para recibir el byte
PCINT0_IGNORE:
        Afuera  SREG, SERIAL_INTERRUPT_SREG
        Reti
PCINT0_UNSOLICITED_MESSAGE:
        Sbr     FLAG_REGISTER, 1 < < FLAG_NEED_TO_POLL_MESSAGES
        Rjmp    PCINT0_IGNORE



; Envie el clave de parpadeo en SCRATCH0
; Riesgo pequeño de una condición de la competencia aquí que repetiría una clave, interrumpe si el reloj
automático
; Suena y termina una clave durante esta rutina. Si lo molesta, suspender el
; Interrupción de reloj automático durante esta rutina.
SEND_BLINK_CODE:
        Lds     XL, BLINK_CODES_INDEX
        Empuje  XL
        Intercambie         XL
        Andi    XL, 7; halfbyte alto cambiar de lugado a las primeras tres partes bajas y aisladas
        Subi    XL, 0 - bajo (BLINK_CODES); el offset respecto a la memoria intermedia
        Ldi     XH, máxima (BLINK_CODES); no asume BLINK_CODES sobre el límite de 256 bytes
        St      X, SCRATCH0
        Reviente XL
        Subi    XL, - 16; puntero de Insertar de increment
        Andi    XL, 0x77; bits de orden superior de máscara
        Sts     BLINK_CODES_INDEX, XL
        Ret
V_TIM0_OVF:
        Ballena sei         ; Permita interrupciones para la E/S por entregas
        Hacia dentro        TIMER_INTERRUPT_SREG, SREG
        Empuje  ZH
```

```
        Empuje  ZL
        Empuje  SCRATCH0
;
        Tst     Demora
        Breq    CLK_DELAY_AT_ZERO
        Dec     Demora
CLK_DELAY_AT_ZERO:
;
        Lds     SCRATCH0, SUBSECONDS_COUNT
        Inc     SCRATCH0
        Sts     SUBSECONDS_COUNT, SCRATCH0
        Mov     ZL, SCRATCH0; para el parpadeo codifica abajo
        Andi    SCRATCH0, 15; para 1MHz
        Brne    CLK_NOT_FOUR_SECONDS
; Increment se detener en 0xff cada cuatro segundos
        Lds     SCRATCH0, GPS_TIMER
        Inc     SCRATCH0
        Brne    CLK_NOT_ROLLOVER
        Dec     SCRATCH0
CLK_NOT_ROLLOVER:
        Sts     GPS_TIMER, SCRATCH0
CLK_NOT_FOUR_SECONDS:
;
; Generar claves de parpadeo (el verde, entonces/luego rojo)
        Sbrc    ZL, 0; pasa por alto los ciclos de reloj alternativos (1MHz)
        Rjmp    CLK_BLINK_OUT
        Bst     ZL, 1; que el indicador de T ahora demuestra sobre / de la fase del parpadeo pasa por un ciclo

; Halfbyte alto - orden es el puntero de Insertar.  La bajo - orden es el puntero de función.
; Cargue el clave de parpadeo actual.
        Lds     ZL, BLINK_CODES_INDEX
```

```
        Mov     ZH, ZL
        Intercambie     ZL
        Andi    ZH, 7
        Andi    ZL, 7
        Cp      ZH, ZL
        Breq    CLK_BLINK_OUT; si Insertar y punteros de función son lo mismos, ningunas claves
        Ldi     ZL, bajo (BLINK_CODES); dirección de orden baja de selección de claves
        Añada   ZL, ZH; añade el offset de puntero
        Ldi     ZH, alto (BLINK_CODES); la toma que BLINK_CODES no sobre el límite de 256 bytes
        Ld      SCRATCH0, Z; consigue el clave de parpadeo
        Brtc    CLK_BLINK_OFF
;
CLK_BLINK_ON:
        Cpi     ¿SCRATCH0, 16; cualquier (primer dígito) verde que queda?
        Brlo    CLK_BLINK_ON_SECOND
        Sbi     PORTB, LED_OUT_GREEN; tropieza con el verde
        Subi    SCRATCH0, 16; disminuye el primer recuento de dígito
        St      Z, SCRATCH0
        Rjmp    CLK_BLINK_OUT
;
CLK_BLINK_ON_SECOND:
        Tst     ¿SCRATCH0; cualquier (segundo dígito) rojo que queda?
        Breq    CLK_BLINK_OUT; no
        Sbi     PORTB, LED_OUT_RED; tropieza con el rojo
        Subi    SCRATCH0, 1; disminuye el segundo recuento de dígito
        St      Z, SCRATCH0
        Rjmp    CLK_BLINK_OUT
;
CLK_BLINK_OFF:
        Cbi     PORTB, LED_OUT_GREEN
        Cbi     PORTB, LED_OUT_RED; se enciende saliendo
```

```
        Tst     ¿SCRATCH0; la clave terminada?
        Brne    CLK_BLINK_OUT; no
        Lds     SCRATCH0, BLINK_CODES_INDEX; puntero de función de increment
        Inc     SCRATCH0
        Andi    SCRATCH0, 0x77; bits de orden superior de máscara de ambo halfbytes
        Sts     BLINK_CODES_INDEX, SCRATCH0; salva el puntero de función
CLK_BLINK_OUT:
;
        Reviente SCRATCH0
        Reviente ZL
        Reviente ZH
        Afuera  SREG, TIMER_INTERRUPT_SREG
        Reti
PRINT_DECIMAL:
        Empuje  ZH
        Empuje  ZL
;
        Ldi     ZH, alto (PD_TABLE < <1)
        Ldi     ZL, bajo (PD_TABLE < <1)
PD_DIGITLOOP:
        Ldi     El carácter", 0'
        Lpm     SCRATCH1, + de Z; bajo
        Cpi     SCRATCH1, 0xfe
        Breq    PD_OUT
        Brcc    PD_LASTDIGIT
        Lpm     SCRATCH2, + de Z; alto
PD_COUNTLOOP:
        Submarino       XL, SCRATCH1
        Sbc     XH, SCRATCH2
        Brcs    PD_UNDERFLOW
        Inc     Carácter
```

```
        Sbr     SCRATCH0, 1 < < 1; el indicador de suppresion de cero conductor determinado
        Rjmp    PD_COUNTLOOP
PD_UNDERFLOW:
        Añada   XL, SCRATCH1
        Adc     XH, SCRATCH2
PD_SEND:
        Sbrc    SCRATCH0, 1; sofoca los ceros precedentes
        Rcall   SERIAL_SEND_BYTE_FROM_CHAR
        Rjmp    PD_DIGITLOOP
PD_LASTDIGIT:
        Cpi     SCRATCH0, 1; xxx.Formato de x y cero hasta ahora
        Brne    PD_NO_EXTRA_ZERO
        Ldi     El carácter", 0'
        Rcall   SERIAL_SEND_BYTE_FROM_CHAR
PD_NO_EXTRA_ZERO:
        Sbr     SCRATCH0, 1 < < 1; permite la impresión de un cero solo si el valor es 0
        Ldi     El carácter".'
        Sbrc    SCRATCH0, 0
        Rcall   SERIAL_SEND_BYTE_FROM_CHAR; punto de decimal de letra en xxx.Modo de x
        Rjmp    PD_DIGITLOOP
PD_OUT:
        Reviente ZL
        Reviente ZH
        Ret
; Las constantes que 10000,1000,100,10,1 little endian, 0xff antes de que las últimas marcas de dígito, 0xfe
terminen
PD_TABLE:
        .Db     0x10, 0x27, 0xe8, 0x03, 0x64, 0x00, 0x0a, 0x00, 0xff, 0x01, 0x00, 0xfe
PARSE_DECIMAL:
        Clt     ; El indicador de T caracteriza el último dígito después ".'
        Clr     SCRATCH0
```

```
        Clr     SCRATCH1
PSD_LOOP:
; Consiga dígito
        Ld      El carácter, X+
        Tst     DELIM
        Brne    PSD_NOT_SKIP_DECIMAL
        Cpi     El carácter'.' ; Pase por alto el modo decimal
        Breq    PSD_NEXTCHAR
PSD_NOT_SKIP_DECIMAL:
        Cp      El carácter, DELIM
        Brne    PSD_NOT_DECIMAL
        Juego
        Rjmp    PSD_NEXTCHAR
PSD_NOT_DECIMAL:
        Subi    El carácter", 0'
        Brlo    PSD_DONE
        Cpi     El carácter, 10
        Brsh    PSD_DONE
; Multiplique antes de las 10
        Rcall   MULX10_SCRATCH10
; Añada el dígito
        Clr     SCRATCH2
        Añada   SCRATCH0, el carácter
        Adc     SCRATCH1, SCRATCH2
;
        Brts    PSD_DONE; ¡fuera! si el decimal y one more dígito
PSD_NEXTCHAR:
        Dec
        Brne    PSD_LOOP
PSD_DONE:
        Ret
```

```
; Multiplicar antes de las 10 (X * * 8 de + X de 10 = * de X 2 )
MULX10_SCRATCH10:
        Lsl     SCRATCH0
        Rol     SCRATCH1
        Mov     SCRATCH3, SCRATCH0
        Mov     SCRATCH2, SCRATCH1
        Lsl     SCRATCH3
        Rol     SCRATCH2
        Lsl     SCRATCH3
        Rol     SCRATCH2
        Añada   SCRATCH0, SCRATCH3
        Adc     SCRATCH1, SCRATCH2
        Ret
CMD_SETNAME:
        Rcall   CLEAR_PARSE_FIELDS
        Ldi     ZH, alto (SETNAME_PARSE_STRING < <1)
        Ldi     ZL, bajo (SETNAME_PARSE_STRING < <1)
        Cbr     FLAG_REGISTER, 1 < < FLAG_PARSE_NMEA
        Ldi     Carácter,' '
        Mov     DELIM, el carácter
        Rcall   PARSE_NMEA_OR_COMMAND_LINE
;
        Ldi     ZH, alto (FLDADR_DEVICENAME)
        Ldi     ZL, bajo (FLDADR_DEVICENAME)
        Ldi     XH, alto (EEADR_DEVICENAME)
        Ldi     XL, bajo (EEADR_DEVICENAME)
        Ldi     SCRATCH0, FLDLEN_DEVICENAME
CSN_WRITE_LOOP:
        Ld      El carácter, Z+
        Rcall   EEPROM_WRITEBYTE
```

```
        Dec     SCRATCH0
        Brne    CSN_WRITE_LOOP; el campo ya era la duración puesto fin o correcta nula
;
        Rjmp    CMD_SAVE_COMMAND_EXECUTED_AND_RET; respuesta con el mensaje comando
ejecutar

; Llamar para analizar gramaticalmente y ejecutar el mando de SETADDRESS
; : uno o dos parámetros esperaban, telefonee la cantidad y envie por correo electrónico la dirección
opcionalmente
; Salva el número de teléfono y la dirección de correo electrónico a EEPROM
; Llamar mientras comm por entregas es corrido
; SALOCATE variante hace uno LOCATE después de la terminación
CMD_SALOCATE:
        Rcall   CMD_LOCATE
        Rjmp    CSA_COMMON
CMD_SETADDRESS:
        Rcall   CMD_SAVE_COMMAND_EXECUTED_AND_RET
CSA_COMMON:
        Rcall   SKIP_WHITESPACE; pase por alto el espacio después del mandato
        Ldi     XH, alto (EEADR_PHONE_NUMBER)
        Ldi     XL, bajo (EEADR_PHONE_NUMBER)
        Ldi     SCRATCH0, EELEN_PHONE_NUMBER
        Rcall   EEPROM_WRITE_FIELD; número de teléfono de escritura
        Ldi     XH, alto (EEADR_EMAIL_ADDR)
        Ldi     XL, bajo (EEADR_EMAIL_ADDR)
; ¿Hay otro campo para la dirección de correo electrónico?
CSA_SKIP:
        Rcall   GET_NEXT_CHAR
        Brts    CSA_DONE
        Cpi     ¿El carácter, 32; el espacio?
        Brlo    CSA_NO_EMAIL; inferior que el espacio debe ser Cr así que ningún correo electrónico addr
```

```
        Breq    CSA_SAVE_EMAIL; medios de espacio al que la dirección de correo electrónico sigue
        Rjmp    CSA_SKIP el espacio más arriba - guarde pasar por alto las calidades
CSA_SAVE_EMAIL:
        Ldi     SCRATCH0, EELEN_EMAIL_ADDR
        Rcall   EEPROM_WRITE_FIELD; dirección de correo electrónico de escritura
        Rjmp    CSA_DONE
CSA_NO_EMAIL:
        Clr     El carácter; el cero afuera de la dirección de correo electrónico
        Rcall   EEPROM_WRITEBYTE
CSA_DONE:
        Ret
; Los  generales poner se establecen para ocurrir después del procesamiento de mensaje
CMD_LOCATE:
        Ldi     SCRATCH1, alto (GPS_LOCATE_AND_SEND)
        Ldi     SCRATCH0, bajo (GPS_LOCATE_AND_SEND)
        Rjmp    CMD_SAVE_PENDING_AND_RET
CMD_SETPASSWORD:
        Rcall   CLEAR_PARSE_FIELDS
        Ldi     ZH, alto (SETPASSWORD_PARSE_STRING < <1)
        Ldi     ZL, bajo (SETPASSWORD_PARSE_STRING < <1)
        Cbr     FLAG_REGISTER, 1 < < FLAG_PARSE_NMEA
        Ldi     Carácter,' '
        Mov     DELIM, el carácter
        Rcall   PARSE_NMEA_OR_COMMAND_LINE
; Compare los dos campos de contraseña
        Ldi     XH, alto (FLDADR_PASSWORD1)
        Ldi     XL, bajo (FLDADR_PASSWORD1)
        Ldi     ZH, alto (FLDADR_PASSWORD2)
        Ldi     ZL, bajo (FLDADR_PASSWORD2)
        Ldi     SCRATCH0, FLDLEN_PASSWORD1
CSP_COMPARE_LOOP:
```

```
        Ld      SCRATCH1, X+
        Ld      SCRATCH2, Z+
        Cp      SCRATCH1, SCRATCH2
        Brne    CSP_PASSWORD_NOGOOD
        Tst     SCRATCH1
        Breq    CSP_COMPARE_GOOD
        Dec     SCRATCH0
        Brne    CSP_COMPARE_LOOP
CSP_COMPARE_GOOD:
        Cpi     ¿SCRATCH0, FLDLEN_PASSWORD1 - MIN_PASSWORD_LENGTH + 1; la contraseña
larga lo suficientemente?
        Brsh    CSP_PASSWORD_NOGOOD; demasiado bajo
; Escriba la contraseña a EEPROM
        Ldi     ZH, alto (FLDADR_PASSWORD1)
        Ldi     ZL, bajo (FLDADR_PASSWORD1)
        Ldi     XH, alto (EEADR_PASSWORD)
        Ldi     XL, bajo (EEADR_PASSWORD)
        Ldi     SCRATCH0, FLDLEN_PASSWORD1
CSP_WRITE_LOOP:
        Ld      El carácter, Z+
        Rcall   EEPROM_WRITEBYTE
        Dec     SCRATCH0
        Brne    CSP_WRITE_LOOP; el campo ya era la duración puesto fin o correcta nula
;
        Ldi     SCRATCH1, alto (MULTI_SEND_PASSWORD_CHANGED)
        Ldi     SCRATCH0, bajo (MULTI_SEND_PASSWORD_CHANGED)
        Rjmp    CMD_SAVE_PENDING_AND_RET
CSP_PASSWORD_NOGOOD:
        Ldi     SCRATCH1, alto (MULTI_SEND_PASSWORD_CHANGE_FAILED)
        Ldi     SCRATCH0, bajo (MULTI_SEND_PASSWORD_CHANGE_FAILED)
        Rjmp    CMD_SAVE_PENDING_AND_RET
```

```
; Apague el modo de agrupación según rendimiento
; Esto también es llamado en la nueva empresa para poner los parámetros de incumplimiento
CMD_POWERON:
CMD_TRACKOFF:
        Ldi     SCRATCH1, alto (MULTI_SEND_COMMAND_EXECUTED)
        Ldi     SCRATCH0, bajo (MULTI_SEND_COMMAND_EXECUTED)
        Rcall   CMD_SAVE_PENDING_AND_RET
CMD_TRACKOFF_ATSTART:
        Ldi     SCRATCH0, alto (DEFAULT_PHONE_POLL_INTERVAL)
        Sts     POLL_INTERVAL, SCRATCH0
        Ldi     SCRATCH0, bajo (DEFAULT_PHONE_POLL_INTERVAL)
        Sts     + 1 de POLL_INTERVAL, SCRATCH0
        Ldi     SCRATCH0, TRACK_STATE_OFF
        Sts     TRACK_STATE, SCRATCH0
        Ret
; Encienda el modo de agrupación según rendimiento
; Ninguna respuesta transmitió aquí. Locate debe ser enviado en breve.
CMD_TRACKON:
        Sts     POLL_INTERVAL, REG_ZERO
        Sts     POLL_INTERVAL que + 1, REG_ZERO; debe re- sondear inmediatamente
        Sts     FASTEST_SPEED, REG_ZERO
        Sts     + 1 de FASTEST_SPEED, REG_ZERO
        Ldi     SCRATCH0, TRACK_STATE_INITIAL
        Sts     TRACK_STATE, SCRATCH0
        Ldi     SCRATCH0, 0xf0
        Sts     TRACK_COUNT, SCRATCH0; make sure que notifica inmediatamente
        Sts     MOVING_LOCATE_COUNT, SCRATCH0; whether en movimiento o parar
        Ret
CMD_POWERSAVE:
        Sts     POLL_INTERVAL, REG_ZERO
        Sts     POLL_INTERVAL que + 1, REG_ZERO; debe re- sondear inmediatamente
```

```
        Ldi     SCRATCH0, TRACK_STATE_POWERSAVE_PHONE_ON
        Sts     TRACK_STATE, SCRATCH0
        Rjmp    CMD_SAVE_COMMAND_EXECUTED_AND_RET


CMD_SETSPEED:
        Ldi     ZH, alto (SETSPEED_DECODE_STRING < <1)
        Ldi     ZL, bajo (SETSPEED_DECODE_STRING < <1)
        Rjmp    CMD_SETTRACK_SETPOWER_SETSPEED
CMD_SETTRACK:
        Ldi     ZH, alto (SETTRACK_DECODE_STRING < <1)
        Ldi     ZL, bajo (SETTRACK_DECODE_STRING < <1)
        Rjmp    CMD_SETTRACK_SETPOWER_SETSPEED
CMD_SETPOWER:
        Ldi     ZH, alto (SETPOWER_DECODE_STRING < <1)
        Ldi     ZL, bajo (SETPOWER_DECODE_STRING < <1)
CMD_SETTRACK_SETPOWER_SETSPEED:
        Rcall   SAVE_SETTINGS
; Genere y mande ajustes y mensaje de estado
CMD_STATUS:
        Ldi     SCRATCH1, alto (MULTI_SEND_STATUS)
        Ldi     SCRATCH0, bajo (MULTI_SEND_STATUS)
        Rjmp    CMD_SAVE_PENDING_AND_RET


; Rechace el mando irreconocido
CMD_UNRECOGNIZED:
        Rcall   SKIP_TO_NEXT_LINE
#Ifdef  DEBUG_KEEP_INVALID_MESSAGES
        Ldi     SCRATCH0, 0xff
        Sts     DEBUG_INVALID_MESSAGE_FLAG, SCRATCH0
        Ret
#Endif
```

```
        Ldi     SCRATCH1, alto (MULTI_SEND_UNKNOWN_COMMAND)
        Ldi     SCRATCH0, bajo (MULTI_SEND_UNKNOWN_COMMAND)
        Rjmp    CMD_SAVE_PENDING_AND_RET

; Reinicialice o reinit el sistema
CMD_REINIT:
        Ldi     SCRATCH1, alto (CMD_REINIT_DO)
        Ldi     SCRATCH0, bajo (CMD_REINIT_DO)
        Rjmp    CMD_SAVE_PENDING_AND_RET
CMD_REBOOT:
        Ldi     SCRATCH1, alto (CMD_REBOOT_DO)
        Ldi     SCRATCH0, bajo (CMD_REBOOT_DO)
        Rjmp    CMD_SAVE_PENDING_AND_RET
CMD_SAVE_COMMAND_EXECUTED_AND_RET:
        Ldi     SCRATCH1, alto (MULTI_SEND_COMMAND_EXECUTED)
        Ldi     SCRATCH0, bajo (MULTI_SEND_COMMAND_EXECUTED)
CMD_SAVE_PENDING_AND_RET:
        Sts     COMMAND_PENDING, SCRATCH1
        Sts     + 1 de COMMAND_PENDING, SCRATCH0
        Ret
CMD_REINIT_DO:
        Rcall   CLEAR_EEPROM_FLAG
CMD_REBOOT_DO:
        Rcall   MULTI_SEND_COMMAND_EXECUTED
#Ifdef  RESET_PHONE_PERIODICALLY
        Ldi     Demora, (* de PHONE_RESET_WAKEUP_DELAY4)
        Rcall   SAFE_DELAY; espere que el mensaje sea enviado
        Rcall   PHONE_RESET
#Endif
CLEAR_RAM_AND_REBOOT:
        Cli
```

```
        Wdr
        Ldi     SCRATCH1, alto (SRAM_SIZE)
        Ldi     SCRATCH0, bajo (SRAM_SIZE)
        Ldi     ZH, alto (SRAM_START)
        Ldi     ZL, bajo (SRAM_START)
CMD_REBOOT_LOOP:
        St      + de Z, REG_ZERO; borra toda RAM estática
        Dec     SCRATCH0
        Brne    CMD_REBOOT_LOOP
        Dec     SCRATCH1
        Brne    CMD_REBOOT_LOOP
        Rjmp    V_RESET; adiós
CLEAR_EEPROM_FLAG:
        Ldi     XH, alto (EEADR_CHECKCODE)
        Ldi     XL, bajo (EEADR_CHECKCODE)
        Clr     Carácter
        Rcall   EEPROM_WRITEBYTE
        Rcall   EEPROM_WRITEBYTE; pase un trapo al EEPROM buen indicador
        Ret
CMD_CAUSEWDR:
        Cli
        Rjmp    CMD_CAUSEWDR


GPS_LOCATE:
        Sts     FLDADR_GPS_LINE_COUNT_FLAGS, REG_ZERO; el recuento de líneas después de la
posición legítimo
        Sts     GPS_TIMER, REG_ZERO; cuenta up at los intervalos de 4 segundos
;
        Cbi     PORTB, GPS_POWER_OUT;  generales poder on
        Ldi     YH, alto (RECEIVE_BUFFER)
        Mov     YL, SERIAL_POINTER; se deshace de todos caracteres en la memoria intermedia
```

```
        Ldi     SCRATCH0, GPS_DATASTREAM_WAIT_TIME
        Sts     RECEIVE_TIMEOUT, SCRATCH0
        Rcall   CLEAR_PARSE_FIELDS
        Rcall   SERIAL_RECEIVE_ENABLE_FROM_GPS
;
GPS_NEXT_LINE:
        Wdr
        Ldi     ZH, alto (GPS_MATCH_PATTERN < <1)
        Ldi     ZL, bajo (GPS_MATCH_PATTERN < <1)
        Clt     ; Limpie la bandera de error
        Rjmp    BRANCH_ON_STRING
;
GPS_PARSE_OTHER:
        Brts    GPS_DATASTREAM_TIMEOUT
        Rcall   SKIP_TO_NEXT_LINE
        Rjmp    GPS_CHECK_TIMEOUT_AND_NEXT_LINE
;
GPS_DATASTREAM_TIMEOUT:
        Ldi     SCRATCH0, GPS_STATUS_DATASTREAM_TIMEOUT
        Rjmp    GPS_SAVE_AND_OUT
;
GPS_PARSE_GPRMC:
        Ldi     ZH, alto (GPRMC_PARSE_STRING < <1)
        Ldi     ZL, bajo (GPRMC_PARSE_STRING < <1)
        Rjmp    GPS_PARSE_GO
;
GPS_PARSE_GPGGA:
        Ldi     ZH, alto (GPGGA_PARSE_STRING < <1)
        Ldi     ZL, bajo (GPGGA_PARSE_STRING < <1)
;       Rjmp    GPS_PARSE_GO
;
```

```
GPS_PARSE_GO:
        Mov     YL, BASE; descompone gramaticalmente
        Sbr     FLAG_REGISTER, 1 < < FLAG_PARSE_NMEA
        Ldi     Carácter,','
        Mov     DELIM, el carácter
        Rcall   PARSE_NMEA_OR_COMMAND_LINE
        Rcall   SKIP_WHITESPACE
        Brts    GPS_DATASTREAM_TIMEOUT
        Cp      Suma de control, VERIFY_CHECKSUM
        Brne    GPS_FIX_NOT_VALID; inválido de suma de control, guarda tratar
        Lds     SCRATCH0, FLDADR_GPRMC_FIXVALID
        Cpi     SCRATCH0", a'; ¿usted ve? si la posición es legítima aún
        Brne    GPS_FIX_NOT_VALID
        Lds     SCRATCH0, FLDADR_GPGGA_FIXVALID
        Cpi     SCRATCH0", 1'
        Brsh    GPS_FIX_VALID
GPS_FIX_NOT_VALID:


GPS_CHECK_TIMEOUT_AND_NEXT_LINE:
; Tiempo muerto de cheque
        Ldi     XH, alto (EEADR_GPS_FIX_WAIT_TIME)
        Ldi     XL, bajo (EEADR_GPS_FIX_WAIT_TIME)
Lds     SCRATCH1, TRACK_STATE
        Cpi     SCRATCH1, TRACK_STATE_BLOCKED
        Brne    GPS_NOT_BLOCKED_1
        Ldi     XL, bajo (EEADR_BLOCKED_GPS_FIX_WAIT_TIME); mismo toma de página
GPS_NOT_BLOCKED_1:
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0
        Lds     SCRATCH1, GPS_TIMER
        Cp      SCRATCH1, SCRATCH0
        Brlo    GPS_NEXT_LINE
```

```
; Tiempo muerto
        Ldi     SCRATCH0, GPS_STATUS_FIX_TIMEOUT
        Rjmp    GPS_SAVE_AND_OUT
GPS_FIX_VALID:
; ¿Tenemos enough líneas? (Deje la posición se estabilizar)
        Lds     SCRATCH0, FLDADR_GPS_LINE_COUNT_FLAGS
        Inc     SCRATCH0
        Sts     FLDADR_GPS_LINE_COUNT_FLAGS, SCRATCH0; el recuento de líneas después de la
posición legítimo
        Andi    SCRATCH0, 0x3f; bit 7 = de la mudanza despacio la vez pasada, bit 6 que = reemplazó
        Cpi     ¿SCRATCH0, GPS_LINES_AFTER_FIX; conseguir suficientemente?
        Brlo    GPS_CHECK_TIMEOUT_AND_NEXT_LINE; guarde mirar
; ¿Es cuatro o mejor arreglo se presentado?
        Ldi     XH, alto (FLDADR_GPGGA_SATS)
        Ldi     XL, bajo (FLDADR_GPGGA_SATS)
        Ldi     SCRATCH0, FLDLEN_GPGGA_SATS
        Mov     Pat, SCRATCH0
        Ldi     SCRATCH0, 0xff
        Mov     DELIM, SCRATCH0; no es analizable gramaticalmente.
        Rcall   PARSE_DECIMAL
        Cpi     SCRATCH0, 4
        Brsh    GPS_FIX_GOOD
; ¿Es el tiempo de aceptar unos less than 4 arreglo se presentado?
        Lds     SCRATCH1, TRACK_STATE
        Cpi     SCRATCH1, TRACK_STATE_BLOCKED
        Breq    GPS_FIX_GOOD; acepte la posición se presentada por 3 siempre si bloquear antes
;
        Ldi     XH, alto (EEADR_GPS_FOUR_SAT_WAIT_TIME)
        Ldi     XL, bajo (EEADR_GPS_FOUR_SAT_WAIT_TIME)
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0
        Lds     SCRATCH1, GPS_TIMER
```

```
        Cp      SCRATCH1, SCRATCH0
        Brsh    GPS_FIX_GOOD
        Rjmp    GPS_CHECK_TIMEOUT_AND_NEXT_LINE


GPS_FIX_GOOD:
; Si el curso es nulo, ponga un cero
        Lds     SCRATCH0, FLDADR_GPRMC_COURSE
        Tst     SCRATCH0
        Brne    GPS_COURSE_NOT_NULL
        Ldi     SCRATCH0", 0'
        Sts     FLDADR_GPRMC_COURSE, SCRATCH0
        Sts     + 1 de FLDADR_GPRMC_COURSE, REG_ZERO
GPS_COURSE_NOT_NULL:
; Verifique desplazado y vuelva a procesar the first time si es así
; GPS devuelve un arreglo desenfrenado a veces, causando muchos informes "Cambiado de lugar" falsos
        Lds     SCRATCH0, TRACK_STATE
        Cpi     SCRATCH0, TRACK_STATE_MOVING
        Brsh    GPS_DISPLACED_OUT; solamente preocúpese por el desplazamiento hacia dentro parado y
bloqueado
;
        Ldi     XH, alto (+ de FLDADR_GPRMC_LATITUDE2)
        Ldi     XL, bajo (+ de FLDADR_GPRMC_LATITUDE2)
        Ldi     ZH, alto (+ de LAST_GOOD_FIX_LATITUDE2)
        Ldi     ZL, bajo (+ de LAST_GOOD_FIX_LATITUDE2)
        Rcall   CHECK_FOR_DISPLACEMENT; busque el desplazamiento de latitud
        Brcc    GPS_DISPLACED; afectado desde la última posición
;
        Ldi     XH, alto (+ de FLDADR_GPRMC_LONGITUDE3)
        Ldi     XL, bajo (+ de FLDADR_GPRMC_LONGITUDE3)
        Ldi     ZH, alto (+ de LAST_GOOD_FIX_LONGITUDE3)
        Ldi     ZL, bajo (+ de LAST_GOOD_FIX_LONGITUDE3)
```

```
        Rcall   CHECK_FOR_DISPLACEMENT; examine el desplazamiento en busca de la longitud
        Brcc    GPS_DISPLACED; afectado desde la última posición
; No desplazar
        Lds     SCRATCH0, FLDADR_GPS_LINE_COUNT_FLAGS; bit 6 es desplazaron el indicador
        Cbr     SCRATCH0, 1 < < GPS_FLAG_DISPLACED; el indicador desplazado claro
        Sts     FLDADR_GPS_LINE_COUNT_FLAGS, SCRATCH0
        Rjmp    GPS_DISPLACED_OUT
;
GPS_DISPLACED:
        Lds     SCRATCH0, FLDADR_GPS_LINE_COUNT_FLAGS; bit 6
        Rjmp    GPS_DISPLACED_OUT;
GPS_MASK_FLAGS_AND_RETRY:
        Andi    SCRATCH0, 0xC0;
        Sts     FLDADR_GPS_LINE_COUNT_FLAGS, SCRATCH0
        Sts     GPS_TIMER, REG_ZERO; evite time out
        Rjmp    GPS_NEXT_LINE; consiga más datos
GPS_DISPLACED_OUT:
        Ldi     XH, alto (FLDADR_GPGGA_ALT)
        Ldi     XL, bajo (FLDADR_GPGGA_ALT)
        Ldi     SCRATCH0, FLDLEN_GPGGA_ALT
        Mov     Pat, SCRATCH0
        Ldi     SCRATCH0, 0xff
        Mov     DELIM, SCRATCH0; no descompone gramaticalmente el punto decimal
        Rcall   PARSE_DECIMAL
;
#Ifdef  ALT_IN_FEET
        Ldi     XH, alto (CF_M_FT)
        Ldi     XL, bajo (CF_M_FT)
        Rcall   UNITS_CONVERT_X4; multiplica SCRATCH1:SCRATCH0 antes de las cuatro
        Sts     FLDADR_GPGGA_ALT_NUMERIC, ZH
        Sts     + 1 de FLDADR_GPGGA_ALT_NUMERIC, ZL
```

```
#Endif
;
#Ifdef    ALT_IN_METERS
          Sts       FLDADR_GPGGA_ALT_NUMERIC, SCRATCH1
          Sts       + 1 de FLDADR_GPGGA_ALT_NUMERIC, SCRATCH0
#Endif
;
; Velocidad de análisis sintáctico
          Ldi       XH, alto (FLDADR_GPRMC_SPEED)
          Ldi       XL, bajo (FLDADR_GPRMC_SPEED)
          Ldi       SCRATCH0, FLDLEN_GPRMC_SPEED
          Mov       Pat, SCRATCH0
          Ldi       SCRATCH0''.'
          Mov       DELIM, SCRATCH0;

#Ifdef    SPEED_IN_MPH
          Ldi       XH, alto (CF_KNOT_MPH)
          Ldi       XL, bajo (CF_KNOT_MPH)
          Rcall     UNITS_CONVERT_X2; multiplica SCRATCH1:SCRATCH0 antes de las dos
          Sts       FLDADR_GPRMC_SPEED_NUMERIC, ZH
          Sts       + 1 de FLDADR_GPRMC_SPEED_NUMERIC, ZL
#Endif
;
#Ifdef    SPEED_IN_KPH
          Ldi       XH, alto (CF_KNOT_KPH)
          Ldi       XL, bajo (CF_KNOT_KPH)
          Rcall     UNITS_CONVERT_X2; multiplica SCRATCH1:SCRATCH0 antes de las dos
          Sts       FLDADR_GPRMC_SPEED_NUMERIC, ZH
          Sts       + 1 de FLDADR_GPRMC_SPEED_NUMERIC, ZL
#Endif
;
```

```
#Ifdef    SPEED_IN_KNOTS
          Sts       FLDADR_GPRMC_SPEED_NUMERIC, SCRATCH1
          Sts       + 1 de FLDADR_GPRMC_SPEED_NUMERIC, SCRATCH0
          Mov       ZH, SCRATCH1
          Mov       ZL, SCRATCH0
#Endif
;
          Ldi       XH, alto (EEADR_GPS_MOVING_THRESHOLD)
          Ldi       XL, bajo (EEADR_GPS_MOVING_THRESHOLD)
          Rcall     EEPROM_READWORD_X_TO_SCRATCH10;
        ; SCRATCH1 MOVING_THRESHOLD de =, SCRATCH0 = MOVING_RETRY_THRESHOLD
          Tst       ZH; byte alto
          Brne      GPS_MOVING
          Cp        ZL, SCRATCH0
          Brsh      GPS_MOVING
          Cp        ZL, SCRATCH1
          Brlo      GPS_NOT_MOVING
          Lds       SCRATCH0, FLDADR_GPS_LINE_COUNT_FLAGS
          Sbr       SCRATCH0, 1 < < GPS_FLAG_LOW_SPEED_RETRY
          ;
GPS_MOVING:
          Ldi       SCRATCH0, GPS_STATUS_FIX_MOVING
          Rjmp      GPS_SAVE_AND_OUT
GPS_NOT_MOVING:
          Ldi       SCRATCH0, GPS_STATUS_FIX_STOPPED
GPS_SAVE_AND_OUT:
          Sts       GPS_STATUS, SCRATCH0
          Sbi       PORTB, GPS_POWER_OUT;  generales poder saliendo
; Salve la última buena posición si consiguiéramos un arreglo legítimo
          Cpi       SCRATCH0, GPS_STATUS_FIX_STOPPED
          Brlo      GPS_SLGF_OUT; no salve datos de  inválidos
```

```
        Ldi     XH, alto (FLDADR_GPRMC_TIME)
        Ldi     XL, bajo (FLDADR_GPRMC_TIME)
        Ldi     ZH, alto (LAST_GOOD_FIX)
        Ldi     ZL, bajo (LAST_GOOD_FIX)
        Ldi     SCRATCH1, LAST_GOOD_FIX_LEN
GPS_SLGF_LOOP:
        Ld      SCRATCH2, X+
        St      + de Z, SCRATCH2
        Dec     SCRATCH1
        Brne    GPS_SLGF_LOOP
GPS_SLGF_OUT:
        Ret
CHECK_FOR_DISPLACEMENT:
; Consiga lat / lon actual primero
        Empuje  ZH
        Empuje  ZL; última conocida buena dirección
        Ldi     SCRATCH2, 6; la duración = seis calidades (xx.Xxx)
        Mov     Pat, SCRATCH2
        Clr     DELIM; contenedor. Y tratar como el entero
        Rcall   PARSE_DECIMAL
;
        Reviente XL
        Reviente XH; consigue última conocida buena dirección
        Empuje  SCRATCH1
        Empuje  SCRATCH0; tienda valor de lat / lon actual
; Consiga último buen lat / lon de  ahora
        Ldi     SCRATCH2, 6; la duración = seis calidades (xx.Xxx)
        Mov     Pat, SCRATCH2
        Clr     DELIM; contenedor. Y tratar como el entero
        Rcall   PARSE_DECIMAL
        Reviente SCRATCH2
```

```
        Reviente SCRATCH3; consigue corriente en SCRATCH3:2
; Ahora SCRATCH3:SCRATCH2 tiene corriente y SCRATCH1:SCRATCH0 ha sabido el bien en último lugar
; Restar más pequeño de más grande, resultado terminar en SCRATCH3:SCRATCH2
        Cp      SCRATCH0, SCRATCH2
        Cpc     SCRATCH1, SCRATCH3
        Brcs    CPD_SFL1
        Submarino       SCRATCH0, SCRATCH2
        Sbc     SCRATCH1, SCRATCH3
        Mov     SCRATCH2, SCRATCH0
        Mov     SCRATCH3, SCRATCH1
        Rjmp    CPD_SFL2
CPD_SFL1:
        Submarino       SCRATCH2, SCRATCH0
        Sbc     SCRATCH3, SCRATCH1
CPD_SFL2:
; El resultado es entre 0 y 59,999
; Si el resultado > = 30000, suponga que es un rollover y tome 60,000 - el resultado
        Ldi     SCRATCH1, alto (30000)
        Ldi     SCRATCH0, bajo (30000); ningún cpci
        Cp      SCRATCH2, SCRATCH0
        Cpc     SCRATCH3, SCRATCH1
        Brlo    CPD_NOT_ROLLOVER
        Mov     SCRATCH0, SCRATCH2
        Mov     SCRATCH1, SCRATCH3
        Ldi     SCRATCH3, alto (60000)
        Ldi     SCRATCH2, bajo (60000)
        Submarino       SCRATCH2, SCRATCH0
        Sbc     SCRATCH3, SCRATCH1
CPD_NOT_ROLLOVER:
; Obtenga desplazamiento mínimo en SCRATCH1:SCRATCH0
        Ldi     XH, alto (EEADR_TRACK_MIN_DISPLACEMENT)
```

```
        Ldi     XL, bajo (EEADR_TRACK_MIN_DISPLACEMENT)
        Rcall   EEPROM_READWORD_X_TO_SCRATCH10
; Compárese con el desplazamiento calculado
        Cp      SCRATCH2, SCRATCH0
        Cpc     SCRATCH3, SCRATCH1
; Aquí el alcance es fijado si no hemos se movido
        Ret

#Ifdef  ALT_IN_FEET
#Definir NEED_UNITS_CONVERT 1
#Endif
#Ifdef  SPEED_IN_MPH
#Definir NEED_UNITS_CONVERT 1
#Endif
#Ifdef  SPEED_IN_KPH
#Definir NEED_UNITS_CONVERT 1
#Endif
#Ifdef  NEED_UNITS_CONVERT
UNITS_CONVERT_X4:
        Lsl     SCRATCH0
        Rol     SCRATCH1
UNITS_CONVERT_X2:
        Lsl     SCRATCH0
        Rol     SCRATCH1
UNITS_CONVERT:
        Ldi     SCRATCH2, 16
        Clr     ZH
        Clr     ZL
UC_LOOP:
        Lsr     XH
        Ror     XL
```

```
        Brcc    UC_ZERO
        Añada   ZL, SCRATCH0
        Adc     ZH, SCRATCH1
UC_ZERO:
        Ror     ZH; el alcance de la adición previa se hace MSB
        Ror     ZL
        Dec     SCRATCH2
        Brne    UC_LOOP
        Ret
#Endif

GPS_LOCATE_AND_SEND:
;       Lds     SCRATCH0, TRACK_STATE
;       Cpi     SCRATCH0, TRACK_STATE_POWERSAVE
;       Brsh    GLS_LOCATE_REQUIRED
;       Cpi     SCRATCH0, TRACK_STATE_OFF
;       Breq    GLS_LOCATE_REQUIRED
;       Rjmp    GLS_ALREADY_LOCATED
;GLS_LOCATE_REQUIRED:
        Rcall   GPS_LOCATE
;GLS_ALREADY_LOCATED:
        Ldi     SCRATCH0, FIX_TYPE_LOCATE - FIX_TYPE_BASE
        Sts     SEND_STRING_OFFSET_1, SCRATCH0
GPS_SEND_LOCATION:
        Lds     SCRATCH1, GPS_STATUS
        Ldi     SCRATCH0, PHONE_STRING_GPS_DATASTREAM_TIMEOUT -
PHONE_STRING_BASE
        Cpi     SCRATCH1, GPS_STATUS_DATASTREAM_TIMEOUT
        Breq    MULTI_SEND_DO
        Ldi     SCRATCH0, PHONE_STRING_GPS_FIX_TIMEOUT - PHONE_STRING_BASE
        Cpi     SCRATCH1, GPS_STATUS_FIX_TIMEOUT
```

```
        Breq    MULTI_SEND_DO
        Ldi     SCRATCH0, PHONE_STRING_GPS_LOCATION - PHONE_STRING_BASE
        Rjmp    MULTI_SEND_DO
MULTI_SEND_COMMAND_EXECUTED:
        Ldi     SCRATCH0, PHONE_STRING_COMMAND_EXECUTED - PHONE_STRING_BASE
        Rjmp    MULTI_SEND_DO
MULTI_SEND_PASSWORD_CHANGED:
        Ldi     SCRATCH0, PHONE_STRING_PASSWORD_CHANGED - PHONE_STRING_BASE
        Rjmp    MULTI_SEND_DO
MULTI_SEND_PASSWORD_CHANGE_FAILED:
        Ldi     SCRATCH0, PHONE_STRING_PASSWORD_CHANGE_FAILED -
PHONE_STRING_BASE
        Rjmp    MULTI_SEND_DO
MULTI_SEND_UNKNOWN_COMMAND:
        Ldi     SCRATCH0, PHONE_STRING_UNKNOWN_COMMAND - PHONE_STRING_BASE
        Rjmp    MULTI_SEND_DO
MULTI_SEND_STATUS:
        Rcall   CLEAR_PARSE_FIELDS
        Ldi     YH, alto (PARSE_FIELDS)
        Ldi     YL, bajo (PARSE_FIELDS)
        Ldi     ZH, alto (SETSPEED_DECODE_STRING < <1)
        Ldi     ZL, bajo (SETSPEED_DECODE_STRING < <1)
        Clr     El carácter; para determinar sobre qué  somos
SSM_COPY_LOOP:
        Lpm     SCRATCH2, Z+
        Tst     ¿SCRATCH2; listo con el ?
        Breq    SSM_DONE_WITH_STRING
        Mov     XH, SCRATCH2
        Andi    XH, 15
        Lpm     XL, + de Z; tiene dirección valiosa ahora en X
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0; X incrementado aquí
```

```
;
        Clr     SCRATCH1
        Sbrs    SCRATCH2, DECODE_SETTING_ONE_BYTE_X4
        Rjmp    SSM_NOT_X4
        Lsl     SCRATCH0
        Rol     SCRATCH1
        Lsl     SCRATCH0
        Rol     SCRATCH1; X4
SSM_NOT_X4:
        Sbrs    SCRATCH2, DECODE_SETTING_TWO_BYTE
        St      + de Y, SCRATCH1; escribe el byte alto si el valor un byte
        St      + de Y, SCRATCH0
;
        Sbrs    SCRATCH2, DECODE_SETTING_TWO_BYTE
SSM_DONE_WITH_STRING:
        Inc     Carácter
        Cpi     El carácter, 2; ¿sobre qué  somos?
        Breq    SSM_NOW_DO_POWER
        Brsh    SSM_DONE_WITH_ALL_STRINGS
; Haga la pista ahora
        Ldi     ZH, alto (SETTRACK_DECODE_STRING < <1)
        Ldi     ZL, bajo (SETTRACK_DECODE_STRING < <1)
        Rjmp    SSM_COPY_LOOP
SSM_NOW_DO_POWER:
        Ldi     ZH, alto (SETPOWER_DECODE_STRING < <1)
        Ldi     ZL, bajo (SETPOWER_DECODE_STRING < <1)
        Rjmp    SSM_COPY_LOOP
SSM_DONE_WITH_ALL_STRINGS:
        St      + de Y, REG_ZERO; byte alto del recuento de WDR
        Lds     SCRATCH0, WDR_COUNT
        St      + de Y, SCRATCH0; byte bajo del recuento de WDR
```

```
;
        Ldi     ZH, alto (PHONE_GET_BATTERY_AND_SIGNAL_STRING < <1)
        Ldi     ZL, bajo (PHONE_GET_BATTERY_AND_SIGNAL_STRING < <1)
        Rcall   DO_PHONE_OPERATION
        Ldi     SCRATCH0, PHONE_STRING_STATUS_MESSAGE - PHONE_STRING_BASE
;       Rjmp    MULTI_SEND_DO
MULTI_SEND_DO:
#DEBUG_MESSAGE_COUNT_ENABLE de ifdef
        Lds     SCRATCH1, + 1 de DEBUG_MESSAGE_COUNT
        Inc     SCRATCH1
        Sts     + 1 de DEBUG_MESSAGE_COUNT, SCRATCH1
        Brne    MSD_NO_INCHIGH
        Lds     SCRATCH1, DEBUG_MESSAGE_COUNT
        Inc     SCRATCH1
        Sts     DEBUG_MESSAGE_COUNT, SCRATCH1
MSD_NO_INCHIGH:
#Endif
;
        Sts     SEND_STRING_OFFSET_0, SCRATCH0
;
        Ldi     XH, alto (EEADR_PHONE_NUMBER)
        Ldi     XL, bajo (EEADR_PHONE_NUMBER)
        Rcall   EEPROM_READBYTE_X_TO_SCRATCH0
        Tst     SCRATCH0
        Brne    MSD_PHONENUMBER
; Ningún número de teléfono definir no
        Ldi     SCRATCH0, 0x14; clave de parpadeo que ningún número de teléfono definió
        Rcall   SEND_BLINK_CODE
        Rjmp    MSD_END
MSD_PHONENUMBER:
; Tienda
```

```
        Ldi     SCRATCH0, 3; cuántas veces volver a procesar
MSD_STORE_RETRY:
        Sts     CHECKSUM_RETRY_COUNT, SCRATCH0
        Ldi     ZH, alto (PHONE_SEND_STORE_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_SEND_STORE_MESSAGE_STRING < <1)
        Rcall   La suma de control de DO_PHONE_OPERATION; envío está en VERIFY_CHECKSUM
        Brts    MSD_FAILED_128
; Verifique la suma de control y la longitud
        Ldi     ZH, alto (PHONE_READ_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_READ_MESSAGE_STRING < <1);
        Rcall   DO_PHONE_OPERATION; suma de control leída en VERIFY_CHECKSUM
        Brts    MSD_FAILED_64
;
        Cp      Suma de control, VERIFY_CHECKSUM
        Brne    MSD_STORE_CHECKSUM_BAD
        Lds     SCRATCH0, SEND_CHECKSUM_BYTES
        Lds     SCRATCH1, READ_CHECKSUM_BYTES
        Cp      SCRATCH0, SCRATCH1
        Breq    MSD_STORE_DONE
;
MSD_STORE_CHECKSUM_BAD:
        Ldi     ZH, alto (PHONE_DELETE_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_DELETE_MESSAGE_STRING < <1)
        Rcall   DO_PHONE_OPERATION; supresión corrupto mensaje
;
        Lds     SCRATCH0, CHECKSUM_RETRY_COUNT
        Dec     SCRATCH0
        Breq    MSD_FAILED_32;
        Rjmp    MSD_STORE_RETRY
MSD_STORE_DONE:
; Envío
```

```
        Cbr     FLAG_REGISTER, 1 < < FLAG_MESSAGE_SENT
        Ldi     ZH, alto (PHONE_SEND_TRANSMIT_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_SEND_TRANSMIT_MESSAGE_STRING < <1)
        Rcall   DO_PHONE_OPERATION
        Brts    MSD_FAILED_16
; Aclárese y transmita pendiente
        Sbrc    FLAG_REGISTER, FLAG_MESSAGE_SENT; salte si sin la cobertura
        Rcall   MSD_CLEAR_SENT_AND_SEND_PENDING
        Ret
;
; Begin depura
MSD_FAILED_128:
#Ifdef  DEBUG_SEND_MESSAGES
        Lds     SCRATCH0, WDR_COUNT
        Ori     SCRATCH0, 128
        Sts     WDR_COUNT, SCRATCH0
        Rjmp    MSD_FAILED
#Endif
MSD_FAILED_64:
#Ifdef  DEBUG_SEND_MESSAGES
        Lds     SCRATCH0, WDR_COUNT
        Ori     SCRATCH0, 64
        Sts     WDR_COUNT, SCRATCH0
        Rjmp    MSD_FAILED
#Endif
MSD_FAILED_32:
#Ifdef  DEBUG_SEND_MESSAGES
        Lds     SCRATCH0, WDR_COUNT
        Ori     SCRATCH0, 32
        Sts     WDR_COUNT, SCRATCH0
        Rjmp    MSD_FAILED
```

```
#Endif
MSD_FAILED_16:
#Ifdef    DEBUG_SEND_MESSAGES
          Lds       SCRATCH0, WDR_COUNT
          Ori       SCRATCH0, 16
          Sts       WDR_COUNT, SCRATCH0
          Rjmp      MSD_FAILED
#Endif
; El Fin depura
MSD_FAILED:
          Ldi       SCRATCH0, 0x12; el parpadeo que el mensaje de envío de clave reprobó
          Rcall     SEND_BLINK_CODE
MSD_END:
          Ret


UNSENT,
MSD_CLEAR_SENT_AND_SEND_PENDING:
          Ldi       ZH, alto (PHONE_LIST_STO_STRING < <1)
          Ldi       ZL, bajo (PHONE_LIST_STO_STRING < <1)
          Rcall     DO_PHONE_OPERATION
;
          Ldi       ZH, alto (FLDADR_CMGL_UNSENT)
          Ldi       ZL, bajo (FLDADR_CMGL_UNSENT)
          Sbr       FLAG_REGISTER, 1 < < FLAG_MESSAGE_SENT; transmite hasta que uno falla
MSD_CSSP_LOOP:
          Ldi       XH, alto (CURRENT_SMS)
          Ldi       XL, bajo (CURRENT_SMS)
          Ldi       SCRATCH0, 4
MSD_CSSP_COPY:
          Ld        SCRATCH1, Z+
          St        X +, SCRATCH1
```

```
        Dec     SCRATCH0
        Brne    MSD_CSSP_COPY
;
        Lds     SCRATCH0, CURRENT_SMS
        Tst     SCRATCH0
        Breq    MSD_CSPP_RECORD_NULL
;
        Empuje  ZH
        Empuje  ZL
;
        Cpi     ¿ZL, bajo (FLDADR_CMGL_SENT); enviar la sección? ( Asuma campos sobre la misma página )
        Sbrs    FLAG_REGISTER, FLAG_PHONE_FULL; elimine el mensaje de unsent si el teléfono está lleno
        Brlo    MSD_CSPP_SENDING
;
        Ldi     ZH, alto (PHONE_DELETE_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_DELETE_MESSAGE_STRING < <1)
        Cbr     FLAG_REGISTER, 1 < < FLAG_PHONE_FULL; solamente elimina uno de ellos
        Rjmp    MSD_CSPP_DO_PHONE_OP
MSD_CSPP_SENDING:
        Sbrs    FLAG_REGISTER, FLAG_MESSAGE_SENT; salte si en último lugar uno fallara
        Rjmp    MSD_CSPP_SKIP_PHONE_OP
        Cbr     FLAG_REGISTER, 1 < < FLAG_MESSAGE_SENT
        Ldi     ZH, alto (PHONE_SEND_TRANSMIT_MESSAGE_STRING < <1)
        Ldi     ZL, bajo (PHONE_SEND_TRANSMIT_MESSAGE_STRING < <1)
MSD_CSPP_DO_PHONE_OP:
        Rcall   DO_PHONE_OPERATION
MSD_CSPP_SKIP_PHONE_OP:
        Reviente ZL
        Reviente ZH
```

```
MSD_CSPP_RECORD_NULL:
;
        Cpi     ZL, bajo (FLDADR_CMGL_STOP)
        Brlo    MSD_CSSP_LOOP
        Ret
EEPROM_READWORD_X_TO_SCRATCH10:
        Afuera  EEARH, XH; EEPROM registro alto
        Afuera  EEARL, XL; EEPROM registro bajo
        Sbi     EECR, EERE; señal de lectura de EEPROM
        Hacia dentro    SCRATCH1, EEDR; el registro de datos de EEPROM
        Adiw    XH:XL, 1
EEPROM_READBYTE_X_TO_SCRATCH0:
        Afuera  EEARH, XH; EEPROM registro alto
        Afuera  EEARL, XL; EEPROM registro bajo
        Sbi     EECR, EERE; señal de lectura de EEPROM
        Hacia dentro    SCRATCH0, EEDR; el registro de datos de EEPROM
        Adiw    XH:XL, 1
        Ret

EEPROM_WRITE_FIELD:
        Rcall   GET_NEXT_CHAR
        Brts    EWF_DONE_WRITING; no debe ocurrir
        Cpi     El carácter, 33
        Brlo    EWF_DONE_WRITING; espacio o Cr - listo con campo
        Rcall   EEPROM_WRITEBYTE
        Dec     SCRATCH0; el recuento de byte
        Brne    EEPROM_WRITE_FIELD
EWF_DONE_WRITING:
        Dec     YL; de regreso hasta arriba de un carácter
        Clr     El carácter; to nulo se terminar
        Tst     ¿SCRATCH0; campo lleno?
```

```
        Brne    EEPROM_WRITEBYTE; la escritura nula y regreso
        Ret

; Escriba el carácter para abordar X en EEPROM X post- inc
EEPROM_WRITEBYTE:
        Afuera  EEARH, XH; máxima de dirección de EEPROM
        Afuera  EEARL, XL; nivel más bajo de dirección de EEPROM
        Afuera  EEDR, el carácter; la calidad recibida
        Hacia dentro    El carácter, SREG
        Cli     ; Next instrucciones tienen que ser atómico
        Sbi     EECR, EEMPE
        Sbi     EECR, EEPE; escribiendo byte ahora
        Afuera  SREG, el carácter; termina la sección atómica
EWB_WAIT:
        Sbic    EECR, EEPE; espera que la escritura termine
        Rjmp    EWB_WAIT
        Adiw    XH:XL, 1; puntero de EEPROM de increment
        Ret

SAVE_SETTINGS:
        Empuje  ZH
        Empuje  ZL
        Rcall   CLEAR_PARSE_FIELDS
        Ldi     ZH, alto (SETTINGS_PARSE_STRING < <1)
        Ldi     ZL, bajo (SETTINGS_PARSE_STRING < <1)
        Cbr     FLAG_REGISTER, 1 < < FLAG_PARSE_NMEA
        Ldi     Carácter,' '
        Mov     DELIM, el carácter
        Rcall   PARSE_NMEA_OR_COMMAND_LINE
        Reviente ZL
        Reviente ZH; consigue dirección de dibujo
```

```
        Ldi     XH, alto (PARSE_FIELDS); dirección de primera base
        Ldi     XL, bajo (PARSE_FIELDS); la cadena numérica (cada 8 bytes)
;
SSET_LOOP:
        Empuje  XH
        Empuje  XL; dirección de torreón de  de decimal
        Ld      SCRATCH0, X
        Tst     SCRATCH0; la marca sure que hay un  ahí
        Breq    SSET_OUT; si no, estamos listos
;
        Ldi     SCRATCH0, 8; la duración de campo
        Mov     Pat, SCRATCH0
        Ldi     SCRATCH0, 0xff
;
        Mov     XH, SCRATCH2
        Andi    XH, 15; dirección de EEPROM
        Lpm     XL, + de Z; nivel más bajo de dirección de EEPROM
;
        Sbrs    SCRATCH2, DECODE_SETTING_ONE_BYTE_X4
        Rjmp    SSET_NO_DIV_X4
        Lsr     SCRATCH1
        Ror     SCRATCH0
        Lsr     SCRATCH1; Divide antes de las 4 si X4 configuración
        Ror     SCRATCH0
SSET_NO_DIV_X4:
;
        Mov     El carácter, SCRATCH1; byte alto
        Sbrc    SCRATCH2, DECODE_SETTING_TWO_BYTE
        Rcall   EEPROM_WRITEBYTE; escriba sobre el byte alto si éste es un ajuste de dos bytes
        Mov     El carácter, SCRATCH0; byte bajo
        Rcall   EEPROM_WRITEBYTE; escritura bajo byte
```

```
;
        Reviente XL
        Reviente XH; puntero de cadena de decimal
        Adiw    XH:XL, 8; next campo
        Rjmp    SSET_LOOP
;
SSET_OUT:
        Reviente XL
        Reviente XH; descarte de la pila
        Ret

#Ifdef  SERIAL_DEBUG
DEBUG_STARTUP_MESSAGE:
        .Db     13, "La computadora empezó en el modo de DEBUG", 13,10,0
TIMEOUT_MESSAGE:
        .Db     "La operación time out", 13,10,0
FAILED_MESSAGE:
        .Db     "La operación falló", 13,10,0
SUCCEEDED_MESSAGE:
        .Db     "La operación dio resultado", 13,10,0
POLL_INTERVAL_MESSAGE:
        .Db     "Sondear IS de intervalo ", 0xf8, alto (POLL_INTERVAL), nivel más bajo
(POLL_INTERVAL), 13,10,0
SPACE_USED_MESSAGE:
        .Db     "El espacio de teléfono usó IS ", 0xf8, alto (FLDADR_CPMS_USED_NUMERIC), \
                (FLDADR_CPMS_USED_NUMERIC) bajo, "OF", 0xf8, \
                (FLDADR_CPMS_TOTAL_NUMERIC) alto, bajo
(FLDADR_CPMS_TOTAL_NUMERIC), \
                13,10,0
CRLF:
        .Db     13,10,0
```

```
#Endif

; Los mandatos nulos poner fin, doble nulo al final de último uno
PHONE_CHECK_STATUS_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, 26, "ATE0", 13,0, "ATE0", 13,0, \
                0xf0, DEFAULT_RECEIVE_TIMEOUT", ¿AT + CFUN?",13,0,0

#Ifdef  DEBUG_KEEP_INVALID_MESSAGES
PHONE_INIT_POLL_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0, \
                0xf0, DEFAULT_RECEIVE_TIMEOUT", AT + CMGF = 1 ", 13,0, \
                "AT + CNMI = 2,1,0,0,0 ", 13,0, \
                0xf1, "= de CPMS de + de AT", 34, "Mí", 34 "", 34, "Mí", 34 "", 34, "ME", 34,13,0, \
                "En = de CMGL de + ", 34 ", REC sin leer", 34,13,0,0
#Más
PHONE_INIT_POLL_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0, \
                0xf0, DEFAULT_RECEIVE_TIMEOUT", AT + CMGF = 1 ", 13,0, \
                "AT + CNMI = 2,1,0,0,0 ", 13,0, \
                0xf1, "= de CPMS de + de AT", 34, "Mí", 34 "", 34, "Mí", 34 "", 34, "ME", 34,13,0, \
                "En = de CMGL de + ", 34", REC leyó", 34,13,0, \
                "En = de CMGL de + ", 34 ", REC sin leer", 34,13,0,0
#Endif

PHONE_GET_BATTERY_AND_SIGNAL_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0, \
                0xf0, DEFAULT_RECEIVE_TIMEOUT, "CBC de + de AT", 13,0, "AT + CSQ", 13,0,0

PHONE_READ_MESSAGE_STRING:
```

```
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0,0 xf0, DEFAULT_RECEIVE_TIMEOUT,
\
        "AT + = de CMGR", 0xfd (4 < <2) + alto (CURRENT_SMS), nivel más bajo (CURRENT_SMS),
13,0,0
PHONE_DELETE_MESSAGE_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0,0 xf0, DEFAULT_RECEIVE_TIMEOUT,
\
                "AT + = de CMGD", 0xfd (4 < <2) + alto (CURRENT_SMS), nivel más bajo
(CURRENT_SMS), 13,0,0

PHONE_LIST_STO_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0, \
                0xf1, "= de CMGL de + de AT", 34, "UNSENT de STO", 34,13,0, \
                "En = de CMGL de + ", 34", STO transmitió", 34,13,0,0

PHONE_POWEROFF_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0,0 xf0,
SMS_SEND_RECEIVE_TIMEOUT, \
                "AT + CFUN = 0 ", 13,0,0

#Ifdef  RESET_PHONE_PERIODICALLY
; Note el cero doble en medio de esto
PHONE_RESET_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0,0, "AT * SWRESET", 13,0
#Más
PHONE_POWERON_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0,0 xf0,
SMS_SEND_RECEIVE_TIMEOUT,
                "AT + CFUN = 1 ", 13,0,0
#Endif
PHONE_MATCH_PATTERN:
```

```
        .Db     Nivel más bajo (PHONE_INIT_ERROR), máxima (PHONE_INIT_ERROR), el "Error",
13,0, \
                Nivel más bajo (PHONE_INIT_OK), máxima (PHONE_INIT_OK), "OK", 13,0, \
                (PHONE_INIT_ISOFF) bajo, máxima (PHONE_INIT_ISOFF)", CFUN de +: 0 ", 0, \
                (PHONE_INIT_IGNORE) bajo, máxima (PHONE_INIT_IGNORE)", CFUN de +: 1 ", 0, \
                Bajos (PHONE_MSG_SENT), máximo (PHONE_MSG_SENT), "CMSS de +", 0, \
                Nivel más bajo (PHONE_INIT_IGNORE), máxima (PHONE_INIT_IGNORE), "Error de
CMS de +", 0, \
                (PHONE_INIT_CPMS) (PHONE_INIT_CPMS) bajos altos, "CPMS de +", 0, \
                Bajos (PHONE_SEND_CMGW), máximo (PHONE_SEND_CMGW), "CMGW de +", 0, \
                Bajos (PHONE_INIT_CMGL), máximo (PHONE_INIT_CMGL), "CMGL de +", 0, \
                Bajos (PHONE_READ_CMGR), máximo (PHONE_READ_CMGR), "CMGR de +", 0, \
                Bajos (PHONE_GOT_CMTI), máximo (PHONE_GOT_CMTI), "CMTI de +", 0, \
                Nivel más bajo (PHONE_SEND_PROMPT), máxima (PHONE_SEND_PROMPT), ">", 0, \
                Nivel más bajo (PHONE_GOT_BATTERY_CHARGE), altos
(PHONE_GOT_BATTERY_CHARGE), "CBC: 0 de +", 0, \
                Nivel más bajo (PHONE_GOT_BATTERY_CHARGE), altos
(PHONE_GOT_BATTERY_CHARGE), "CBC: 1 de +", 0, \
                Bajos (PHONE_GOT_SIGNAL_STRENGTH), máximo
(PHONE_GOT_SIGNAL_STRENGTH), "CSQ de +", 0, \
                Nivel más bajo (PHONE_INIT_ERROR), máxima (PHONE_INIT_ERROR), 0

; Estas funciones no deben mess el puntero de Y.
COMMAND_MATCH_PATTERN:
        .Db     Nivel más bajo (CMD_SETADDRESS), máxima (CMD_SETADDRESS), "SETADDRESS",
0, \
                Nivel más bajo (CMD_SETNAME), máxima (CMD_SETNAME), "SETNAME", 0, \
                Nivel más bajo (CMD_SETPASSWORD), máxima (CMD_SETPASSWORD),
"SETPASSWORD", 0, \
                Nivel más bajo (CMD_SETTRACK), máxima (CMD_SETTRACK), "SETTRACK", 0, \
                Nivel más bajo (CMD_SETPOWER), máxima (CMD_SETPOWER), "SETPOWER", 0, \
```

```
                    Nivel más bajo (CMD_SETSPEED), máxima (CMD_SETSPEED), "SETSPEED", 0, \
                    Nivel más bajo (CMD_STATUS), máxima (CMD_STATUS), el "Estado", 0, \
                    Nivel más bajo (CMD_LOCATE), máxima (CMD_LOCATE), "Establézcase", 0, \
                    Nivel más bajo (CMD_SALOCATE), máxima (CMD_SALOCATE), "SALOCATE", 0, \
                    Nivel más bajo (CMD_TRACKOFF), máxima (CMD_TRACKOFF), "TRACKOFF", 0, \
                    Nivel más bajo (CMD_TRACKON), máxima (CMD_TRACKON), "TRACKON", 0, \
                    Nivel más bajo (CMD_POWERSAVE), máxima (CMD_POWERSAVE), "POWERSAVE",
0, \
                    Nivel más bajo (CMD_POWERON), máxima (CMD_POWERON), "POWERON", 0, \
                    Nivel más bajo (CMD_REBOOT), máxima (CMD_REBOOT), "Reinicialice", 0, \
                    Nivel más bajo (CMD_REINIT), máxima (CMD_REINIT), "REINIT", 0, \
                    Nivel más bajo (CMD_CAUSEWDR), máxima (CMD_CAUSEWDR), "CAUSEWDR", 0, \
                    Nivel más bajo (CMD_UNRECOGNIZED), máxima (CMD_UNRECOGNIZED), 0
CMGL_MATCH_PATTERN:
          .Db       Precio mínimo (PPMN_CSMS_MATCH), máximo (PPMN_CSMS_MATCH) "", 0, \
                    Nivel más bajo (PPMN_CSMS_MATCH), máxima (PPMN_CSMS_MATCH), 34,0, \
                    Nivel más bajo (PPMN_CSMS_SENT), máxima (PPMN_CSMS_SENT), "STO enviado", 0, \
                    Nivel más bajo (PPMN_CSMS_UNSENT), máxima (PPMN_CSMS_UNSENT), "UNSENT
de STO", 0, \
                    Nivel más bajo (PPMN_CSMS_RECEIVED), máxima (PPMN_CSMS_RECEIVED), "REC",
0, \
                    Nivel más bajo (PPMN_OUT), máxima (PPMN_OUT), 0

; <La cantidad de campo > < la duración de máximo y las partes de dirección altas > < bajo el byte de dirección>
SETNAME_PARSE_STRING:
          .Db       1, (FLDLEN_DEVICENAME < <2) + (FLDADR_DEVICENAME) alto, bajo
(FLDADR_DEVICENAME), \
                    0xff

SETPASSWORD_PARSE_STRING:
```

```
        .Db     1, (FLDLEN_PASSWORD1 < <2) + (FLDADR_PASSWORD1) alto, bajo
(FLDADR_PASSWORD1), \
                2, (FLDLEN_PASSWORD2 < <2) + (FLDADR_PASSWORD2) alto, bajo
(FLDADR_PASSWORD2), \
                0xff
#DEBUG_MESSAGE_COUNT_ENABLE de ifdef
PHONE_SEND_STORE_MESSAGE_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0, \
                0xf0, DEFAULT_RECEIVE_TIMEOUT, \
                "AT + = de CMGW", 34,0 xfe, \
                + de (EEADR_PHONE_NUMBER) alto (EELEN_PHONE_NUMBER < <2), \
                Nivel más bajo (EEADR_PHONE_NUMBER), 34,13,0, \
                0xf7, 0xff + de (EEADR_EMAIL_ADDR) alto (EELEN_EMAIL_ADDR < <2), \
                Bajo (EEADR_EMAIL_ADDR), \
                + de 0xff, máxima (EEADR_DEVICENAME) (EELEN_DEVICENAME < <2), \
                Bajo (EEADR_DEVICENAME), \
                "= de MC", 0xf8, alto (DEBUG_MESSAGE_COUNT), bajo
(DEBUG_MESSAGE_COUNT)," ", \
                0xfa, alto (PHONE_STRING_BASE < <1), bajo (PHONE_STRING_BASE < <1), 0xf7,
26,13,0,0
#Más
PHONE_SEND_STORE_MESSAGE_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0, \
                0xf0, DEFAULT_RECEIVE_TIMEOUT, \
                "AT + = de CMGW", 34,0 xfe, \
                + de (EEADR_PHONE_NUMBER) alto (EELEN_PHONE_NUMBER < <2), \
                Nivel más bajo (EEADR_PHONE_NUMBER), 34,13,0, \
                0xf7, 0xff + de (EEADR_EMAIL_ADDR) alto (EELEN_EMAIL_ADDR < <2), \
                Bajo (EEADR_EMAIL_ADDR), \
                + de 0xff, máxima (EEADR_DEVICENAME) (EELEN_DEVICENAME < <2), \
                Bajo (EEADR_DEVICENAME), \
```

```
                    0xfa, alto (PHONE_STRING_BASE < <1), bajo (PHONE_STRING_BASE < <1), 0xf7,
26,13,0,0
#Endif

PHONE_SEND_TRANSMIT_MESSAGE_STRING:
        .Db     0xf0, INIT_RECEIVE_TIMEOUT, "ATE0", 13,0,0 xf0,
SMS_SEND_RECEIVE_TIMEOUT, \
                    "AT + = de CMSS", 0xfd (4 < <2) + alto (CURRENT_SMS), nivel más bajo
(CURRENT_SMS),
 13,0,0
PHONE_STRING_BASE:
PHONE_STRING_COMMAND_EXECUTED:
        .Db     "El comando ejecutado", 0

PHONE_STRING_PASSWORD_CHANGED:
        .Db     "La contraseña cambiada", 0

PHONE_STRING_PASSWORD_CHANGE_FAILED:

        .Db     "El cambio de contraseña reprobado", 0
PHONE_STRING_UNKNOWN_COMMAND:
        .Db     "Comando desconocido", 0

PHONE_STRING_GPS_DATASTREAM_TIMEOUT:
        .Db     "Ningún flujo de datos de  generales dispositivo", \
                    0xfc, alto (LAST_GOOD_FIX_STRING < <1), bajo (LAST_GOOD_FIX_STRING < <1),0

PHONE_STRING_GPS_FIX_TIMEOUT:
        .Db     " generales time out esperando la posición", \
                    0xfc, alto (LAST_GOOD_FIX_STRING < <1), bajo (LAST_GOOD_FIX_STRING < <1),0
```

```
#SPEED_IN_MPH de ifdef

#ALT_IN_FEET de ifdef

PHONE_STRING_GPS_LOCATION:

        .Db     0xfb, alto (FIX_TYPE_BASE < <1), bajo (FIX_TYPE_BASE < <1), \

                " POS ", 0xfd (2 < <2) + (FLDADR_GPRMC_LATITUDE) alto, bajo

(FLDADR_GPRMC_LATITUDE)," ", \

                0xfd, ((- de FLDLEN_GPRMC_LATITUDE2)<<2) + alto (+ de

FLDADR_GPRMC_LATITUDE2), bajo (+ de FLDADR_GPRMC_LATITUDE2), \

                " ",0xfd, (FLDLEN_GPRMC_LATDIR < <2) + (FLDADR_GPRMC_LATDIR) alto, bajo

FLDADR_GPRMC_LATDIR), \

                " ",0xfd, (3 < <2) + (FLDADR_GPRMC_LONGITUDE) alto, bajo

(FLDADR_GPRMC_LONGITUDE), \

                " ",0xfd, ((- de FLDLEN_GPRMC_LONGITUDE3)<<2) + alto (+ de

FLDADR_GPRMC_LONGITUDE3), bajo (+ de FLDADR_GPRMC_LONGITUDE3), \

                " ",0xfd, (FLDLEN_GPRMC_LONGDIR < <2) + (FLDADR_GPRMC_LONGDIR) alto,

bajo (FLDADR_GPRMC_LONGDIR), \

                " El Alt ", 0xf8, el máximo (FLDADR_GPGGA_ALT_NUMERIC), precio mínimo

(FLDADR_GPGGA_ALT_NUMERIC)", pies", \

                " La velocidad ", 0xf9, el máximo (FLDADR_GPRMC_SPEED_NUMERIC), precio mínimo

(FLDADR_GPRMC_SPEED_NUMERIC)", mph", \

        " El curso ", 0xfd (FLDLEN_GPRMC_COURSE < <2) + (FLDADR_GPRMC_COURSE) alto, bajo

(FLDADR_GPRMC_COURSE), \

                " AT ", 0xfd (2 < <2) + alto (+ de FLDADR_GPRMC_DATE4), bajo (+ de

FLDADR_GPRMC_DATE4),"/", \

                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_DATE2), bajo (+ de

FLDADR_GPRMC_DATE2),"/", \

                0xfd, (2 < <2) + (FLDADR_GPRMC_DATE) alto, bajo (FLDADR_GPRMC_DATE)," ", \

                0xfd, (2 < <2) + (FLDADR_GPRMC_TIME) alto, bajo (FLDADR_GPRMC_TIME),":", \

                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME2), bajo (+ de

FLDADR_GPRMC_TIME2),":", \
```

```
                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME4), bajo (+ de
FLDADR_GPRMC_TIME4), la "UTC", \
                " SAT ", 0xfd (FLDLEN_GPGGA_SATS < <2) + alto (FLDADR_GPGGA_SATS), nivel
más bajo (FLDADR_GPGGA_SATS), 0
#Endif
#Endif


#SPEED_IN_KPH de ifdef
#ALT_IN_FEET de ifdef
PHONE_STRING_GPS_LOCATION:
        .Db     0xfb, alto (FIX_TYPE_BASE < <1), bajo (FIX_TYPE_BASE < <1), \
                " POS ", 0xfd (2 < <2) + (FLDADR_GPRMC_LATITUDE) alto, bajo
(FLDADR_GPRMC_LATITUDE)," ", \
                0xfd, ((- de FLDLEN_GPRMC_LATITUDE2)<<2) + alto (+ de
FLDADR_GPRMC_LATITUDE2), bajo (+ de FLDADR_GPRMC_LATITUDE2), \
                " ",0xfd, (FLDLEN_GPRMC_LATDIR < <2) + (FLDADR_GPRMC_LATDIR) alto, bajo
(FLDADR_GPRMC_LATDIR), \
                " ",0xfd, (3 < <2) + (FLDADR_GPRMC_LONGITUDE) alto, bajo
(FLDADR_GPRMC_LONGITUDE), \
                " ",0xfd, ((- de FLDLEN_GPRMC_LONGITUDE3)<<2) + alto (+ de
FLDADR_GPRMC_LONGITUDE3), bajo (+ de FLDADR_GPRMC_LONGITUDE3), \
                " ",0xfd, (FLDLEN_GPRMC_LONGDIR < <2) + (FLDADR_GPRMC_LONGDIR) alto,
bajo (FLDADR_GPRMC_LONGDIR), \
                " El Alt ", 0xf8, el máximo (FLDADR_GPGGA_ALT_NUMERIC), precio mínimo
(FLDADR_GPGGA_ALT_NUMERIC)", pies", \
                " La velocidad ", 0xf9, el máximo (FLDADR_GPRMC_SPEED_NUMERIC), precio mínimo
(FLDADR_GPRMC_SPEED_NUMERIC)", km/h", \
        " El curso ", 0xfd (FLDLEN_GPRMC_COURSE < <2) + (FLDADR_GPRMC_COURSE) alto, bajo
(FLDADR_GPRMC_COURSE), \
                " AT ", 0xfd (2 < <2) + alto (+ de FLDADR_GPRMC_DATE4), bajo (+ de
FLDADR_GPRMC_DATE4),"/", \
```

```
                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_DATE2), bajo (+ de
FLDADR_GPRMC_DATE2),"/", \
                0xfd, (2 < <2) + (FLDADR_GPRMC_DATE) alto, bajo (FLDADR_GPRMC_DATE)," ", \
                0xfd, (2 < <2) + (FLDADR_GPRMC_TIME) alto, bajo (FLDADR_GPRMC_TIME),":", \
                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME2), bajo (+ de
FLDADR_GPRMC_TIME2),":", \
                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME4), bajo (+ de
FLDADR_GPRMC_TIME4), la "UTC", \
                " SAT ", 0xfd (FLDLEN_GPGGA_SATS < <2) + alto (FLDADR_GPGGA_SATS), nivel
más bajo (FLDADR_GPGGA_SATS), 0
#Endif
#Endif

#SPEED_IN_KNOTS de ifdef
#ALT_IN_FEET de ifdef
PHONE_STRING_GPS_LOCATION:
        .Db     0xfb, alto (FIX_TYPE_BASE < <1), bajo (FIX_TYPE_BASE < <1), \
                " POS ", 0xfd (2 < <2) + (FLDADR_GPRMC_LATITUDE) alto, bajo
(FLDADR_GPRMC_LATITUDE)," ", \
                0xfd, ((- de FLDLEN_GPRMC_LATITUDE2)<<2) + alto (+ de
FLDADR_GPRMC_LATITUDE2), bajo (+ de FLDADR_GPRMC_LATITUDE2), \
                " ",0xfd, (FLDLEN_GPRMC_LATDIR < <2) + (FLDADR_GPRMC_LATDIR) alto, bajo
(FLDADR_GPRMC_LATDIR), \
                " ",0xfd, (3 < <2) + (FLDADR_GPRMC_LONGITUDE) alto, bajo
(FLDADR_GPRMC_LONGITUDE), \
                " ",0xfd, ((- de FLDLEN_GPRMC_LONGITUDE3)<<2) + alto (+ de
FLDADR_GPRMC_LONGITUDE3), bajo (+ de FLDADR_GPRMC_LONGITUDE3), \
                " ",0xfd, (FLDLEN_GPRMC_LONGDIR < <2) + (FLDADR_GPRMC_LONGDIR) alto,
bajo (FLDADR_GPRMC_LONGDIR), \
                " El Alt ", 0xf8, el máximo (FLDADR_GPGGA_ALT_NUMERIC), precio mínimo
(FLDADR_GPGGA_ALT_NUMERIC)", pies", \
```

```
                    " La velocidad ", 0xf9, el máximo (FLDADR_GPRMC_SPEED_NUMERIC), precio mínimo
(FLDADR_GPRMC_SPEED_NUMERIC)", nudos", \
          " El curso ", 0xfd (FLDLEN_GPRMC_COURSE < <2) + (FLDADR_GPRMC_COURSE) alto, bajo
(FLDADR_GPRMC_COURSE), \
                    " AT ", 0xfd (2 < <2) + alto (+ de FLDADR_GPRMC_DATE4), bajo (+ de
FLDADR_GPRMC_DATE4),"/", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_DATE2), bajo (+ de
FLDADR_GPRMC_DATE2),"/", \
                    0xfd, (2 < <2) + (FLDADR_GPRMC_DATE) alto, bajo (FLDADR_GPRMC_DATE)," ", \
                    0xfd, (2 < <2) + (FLDADR_GPRMC_TIME) alto, bajo (FLDADR_GPRMC_TIME),":", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME2), bajo (+ de
FLDADR_GPRMC_TIME2),":", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME4), bajo (+ de
FLDADR_GPRMC_TIME4), la "UTC", \
                    " SAT ", 0xfd (FLDLEN_GPGGA_SATS < <2) + alto (FLDADR_GPGGA_SATS), nivel
más bajo (FLDADR_GPGGA_SATS), 0
#Endif
#Endif


#SPEED_IN_MPH de ifdef
#ALT_IN_METERS de ifdef
PHONE_STRING_GPS_LOCATION:
          .Db       0xfb, alto (FIX_TYPE_BASE < <1), bajo (FIX_TYPE_BASE < <1), \
                    " POS ", 0xfd (2 < <2) + (FLDADR_GPRMC_LATITUDE) alto, bajo
(FLDADR_GPRMC_LATITUDE)," ", \
                    0xfd, ((- de FLDLEN_GPRMC_LATITUDE2)<<2) + alto (+ de
FLDADR_GPRMC_LATITUDE2), bajo (+ de FLDADR_GPRMC_LATITUDE2), \
                    " ",0xfd, (FLDLEN_GPRMC_LATDIR < <2) + (FLDADR_GPRMC_LATDIR) alto, bajo
(FLDADR_GPRMC_LATDIR), \
                    " ",0xfd, (3 < <2) + (FLDADR_GPRMC_LONGITUDE) alto, bajo
(FLDADR_GPRMC_LONGITUDE), \
```

```
                    " ",0xfd, ((- de FLDLEN_GPRMC_LONGITUDE3)<<2) + alto (+ de
FLDADR_GPRMC_LONGITUDE3), bajo (+ de FLDADR_GPRMC_LONGITUDE3), \
                    " ",0xfd, (FLDLEN_GPRMC_LONGDIR < <2) + (FLDADR_GPRMC_LONGDIR) alto,
bajo (FLDADR_GPRMC_LONGDIR), \
                    " El Alt ", 0xf8, el máximo (FLDADR_GPGGA_ALT_NUMERIC), precio mínimo
(FLDADR_GPGGA_ALT_NUMERIC)", m", \
                    " La velocidad ", 0xf9, el máximo (FLDADR_GPRMC_SPEED_NUMERIC), precio mínimo
(FLDADR_GPRMC_SPEED_NUMERIC)", mph", \
          " El curso ", 0xfd (FLDLEN_GPRMC_COURSE < <2) + (FLDADR_GPRMC_COURSE) alto, bajo
(FLDADR_GPRMC_COURSE), \
                    " AT ", 0xfd (2 < <2) + alto (+ de FLDADR_GPRMC_DATE4), bajo (+ de
FLDADR_GPRMC_DATE4),"/", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_DATE2), bajo (+ de
FLDADR_GPRMC_DATE2),"/", \
                    0xfd, (2 < <2) + (FLDADR_GPRMC_DATE) alto, bajo (FLDADR_GPRMC_DATE)," ", \
                    0xfd, (2 < <2) + (FLDADR_GPRMC_TIME) alto, bajo (FLDADR_GPRMC_TIME),":", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME2), bajo (+ de
FLDADR_GPRMC_TIME2),":", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME4), bajo (+ de
FLDADR_GPRMC_TIME4), la "UTC", \
                    " SAT ", 0xfd (FLDLEN_GPGGA_SATS < <2) + alto (FLDADR_GPGGA_SATS), nivel
más bajo (FLDADR_GPGGA_SATS), 0
#Endif
#Endif


#SPEED_IN_KPH de ifdef
#ALT_IN_METERS de ifdef
PHONE_STRING_GPS_LOCATION:
          .Db       0xfb, alto (FIX_TYPE_BASE < <1), bajo (FIX_TYPE_BASE < <1), \
                    " POS ", 0xfd (2 < <2) + (FLDADR_GPRMC_LATITUDE) alto, bajo
(FLDADR_GPRMC_LATITUDE)," ", \
```

```
                    0xfd, ((- de FLDLEN_GPRMC_LATITUDE2)<<2) + alto (+ de
FLDADR_GPRMC_LATITUDE2), bajo (+ de FLDADR_GPRMC_LATITUDE2), \
                    " ",0xfd, (FLDLEN_GPRMC_LATDIR < <2) + (FLDADR_GPRMC_LATDIR) alto, bajo
(FLDADR_GPRMC_LATDIR), \
                    " ",0xfd, (3 < <2) + (FLDADR_GPRMC_LONGITUDE) alto, bajo
(FLDADR_GPRMC_LONGITUDE), \
                    " ",0xfd, ((- de FLDLEN_GPRMC_LONGITUDE3)<<2) + alto (+ de
FLDADR_GPRMC_LONGITUDE3), bajo (+ de FLDADR_GPRMC_LONGITUDE3), \
                    " ",0xfd, (FLDLEN_GPRMC_LONGDIR < <2) + (FLDADR_GPRMC_LONGDIR) alto,
bajo (FLDADR_GPRMC_LONGDIR), \
                    " El Alt ", 0xf8, el máximo (FLDADR_GPGGA_ALT_NUMERIC), precio mínimo
(FLDADR_GPGGA_ALT_NUMERIC)", m", \
                    " La velocidad ", 0xf9, el máximo (FLDADR_GPRMC_SPEED_NUMERIC), precio mínimo
(FLDADR_GPRMC_SPEED_NUMERIC)", km/h", \
          " El curso ", 0xfd (FLDLEN_GPRMC_COURSE < <2) + (FLDADR_GPRMC_COURSE) alto, bajo
(FLDADR_GPRMC_COURSE), \
                    " AT ", 0xfd (2 < <2) + alto (+ de FLDADR_GPRMC_DATE4), bajo (+ de
FLDADR_GPRMC_DATE4),"/", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_DATE2), bajo (+ de
FLDADR_GPRMC_DATE2),"/", \
                    0xfd, (2 < <2) + (FLDADR_GPRMC_DATE) alto, bajo (FLDADR_GPRMC_DATE)," ", \
                    0xfd, (2 < <2) + (FLDADR_GPRMC_TIME) alto, bajo (FLDADR_GPRMC_TIME),":", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME2), bajo (+ de
FLDADR_GPRMC_TIME2),":", \
                    0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME4), bajo (+ de
FLDADR_GPRMC_TIME4), la "UTC", \
                    " SAT ", 0xfd (FLDLEN_GPGGA_SATS < <2) + alto (FLDADR_GPGGA_SATS), nivel
más bajo (FLDADR_GPGGA_SATS), 0
#Endif
#Endif
#SPEED_IN_KNOTS de ifdef
```

```
#ALT_IN_METERS de ifdef
PHONE_STRING_GPS_LOCATION:
        .Db     0xfb, alto (FIX_TYPE_BASE < <1), bajo (FIX_TYPE_BASE < <1), \
                " POS ", 0xfd (2 < <2) + (FLDADR_GPRMC_LATITUDE) alto, bajo
(FLDADR_GPRMC_LATITUDE)," ", \
                0xfd, ((- de FLDLEN_GPRMC_LATITUDE2)<<2) + alto (+ de
FLDADR_GPRMC_LATITUDE2), bajo (+ de FLDADR_GPRMC_LATITUDE2), \
                " ",0xfd, (FLDLEN_GPRMC_LATDIR < <2) + (FLDADR_GPRMC_LATDIR) alto, bajo
(FLDADR_GPRMC_LATDIR), \
                " ",0xfd, (3 < <2) + (FLDADR_GPRMC_LONGITUDE) alto, bajo
(FLDADR_GPRMC_LONGITUDE), \
                " ",0xfd, ((- de FLDLEN_GPRMC_LONGITUDE3)<<2) + alto (+ de
FLDADR_GPRMC_LONGITUDE3), bajo (+ de FLDADR_GPRMC_LONGITUDE3), \
                " ",0xfd, (FLDLEN_GPRMC_LONGDIR < <2) + (FLDADR_GPRMC_LONGDIR) alto,
bajo (FLDADR_GPRMC_LONGDIR), \
                " El Alt ", 0xf8, el máximo (FLDADR_GPGGA_ALT_NUMERIC), precio mínimo
(FLDADR_GPGGA_ALT_NUMERIC)", m", \
                " La velocidad ", 0xf9, el máximo (FLDADR_GPRMC_SPEED_NUMERIC), precio mínimo
(FLDADR_GPRMC_SPEED_NUMERIC)", nudos", \
        " El curso ", 0xfd (FLDLEN_GPRMC_COURSE < <2) + (FLDADR_GPRMC_COURSE) alto, bajo
(FLDADR_GPRMC_COURSE), \
                " AT ", 0xfd (2 < <2) + alto (+ de FLDADR_GPRMC_DATE4), bajo (+ de
FLDADR_GPRMC_DATE4),"/", \
                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_DATE2), bajo (+ de
FLDADR_GPRMC_DATE2),"/", \
                0xfd, (2 < <2) + (FLDADR_GPRMC_DATE) alto, bajo (FLDADR_GPRMC_DATE)," ", \
                0xfd, (2 < <2) + (FLDADR_GPRMC_TIME) alto, bajo (FLDADR_GPRMC_TIME),":", \
                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME2), bajo (+ de
FLDADR_GPRMC_TIME2),":", \
                0xfd, (2 < <2) + alto (+ de FLDADR_GPRMC_TIME4), bajo (+ de
FLDADR_GPRMC_TIME4), la "UTC", \
```

```
                    " SAT ", 0xfd (FLDLEN_GPGGA_SATS < <2) + alto (FLDADR_GPGGA_SATS), nivel
más
bajo (FLDADR_GPGGA_SATS), 0#Endif
#Endif
PHONE_STRING_STATUS_MESSAGE:
        .Db     "SPDLMT=", \
                0xf8, alto (STATUSMSG_SPEED_LIMIT), \
                (STATUSMSG_SPEED_LIMIT) bajo, "STOPD=", \
                0xf8, alto (STATUSMSG_TRACK_STOPPED_FIX_INTERVAL), \
                Bajo (STATUSMSG_TRACK_STOPPED_FIX_INTERVAL),"/", \
                0xf8, alto (STATUSMSG_TRACK_STOPPED_NOTIFY_DELAY), \
                Nivel más bajo (STATUSMSG_TRACK_STOPPED_NOTIFY_DELAY)", BLKD=", \
                0xf8, alto (STATUSMSG_TRACK_BLOCKED_FIX_INTERVAL), \
                Bajo (STATUSMSG_TRACK_BLOCKED_FIX_INTERVAL),"/", \
                0xf8, alto (STATUSMSG_TRACK_BLOCKED_NOTIFY_DELAY), \
                Nivel más bajo (STATUSMSG_TRACK_BLOCKED_NOTIFY_DELAY)", MOVNG=", \
                0xf8, alto (STATUSMSG_TRACK_MOVING_FIX_INTERVAL), \
                Bajo (STATUSMSG_TRACK_MOVING_FIX_INTERVAL),"/", \
                0xf8, alto (STATUSMSG_TRACK_MOVING_NOTIFY_FREQ), \
                Nivel más bajo (STATUSMSG_TRACK_MOVING_NOTIFY_FREQ)", PSV=", \
                0xf8, alto (STATUSMSG_POWERSAVE_PHONE_OFF_INTERVAL), \
                Bajo (STATUSMSG_POWERSAVE_PHONE_OFF_INTERVAL),"/", \
                0xf8, alto (STATUSMSG_POWERSAVE_PHONE_ON_INTERVAL), \
                Nivel más bajo (STATUSMSG_POWERSAVE_PHONE_ON_INTERVAL)", 3D /  /
BLKD=", \
                0xf8, alto (STATUSMSG_GPS_FOUR_SAT_WAIT_TIME), \
                Bajo (STATUSMSG_GPS_FOUR_SAT_WAIT_TIME),"/", \
                0xf8, alto (STATUSMSG_GPS_FIX_WAIT_TIME), \
                Bajo (STATUSMSG_GPS_FIX_WAIT_TIME),"/", \
                0xf8, alto (STATUSMSG_BLOCKED_GPS_FIX_WAIT_TIME), \
```

```
                    Nivel más bajo (STATUSMSG_BLOCKED_GPS_FIX_WAIT_TIME)", Doctor en Medicina
/ MS=", \
                    0xf8, alto (STATUSMSG_MIN_DISPLACEMENT), \
                    Bajo (STATUSMSG_MIN_DISPLACEMENT),"/", \
                    0xf8, alto (STATUSMSG_GPS_MOVING_THRESHOLD), \
                    Bajo (STATUSMSG_GPS_MOVING_THRESHOLD),"/", \
                    0xf8, alto (STATUSMSG_GPS_MOVING_RETRY_THRESHOLD), \
                    Nivel más bajo (STATUSMSG_GPS_MOVING_RETRY_THRESHOLD)", murciélago=", \
                    0xf8, alto (STATUSMSG_BATTERY_CHARGE), \
                    Nivel más bajo (STATUSMSG_BATTERY_CHARGE)", SIG=", \
                    0xf8, alto (STATUSMSG_SIGNAL_STRENGTH), \
                    Nivel más bajo (STATUSMSG_SIGNAL_STRENGTH)", WDR=", \
                    0xf8, alto (STATUSMSG_WDR_COUNT), \
                    Nivel más bajo (STATUSMSG_WDR_COUNT)", = de V generales rastreador de Abierto de
0.17A", 0


;PHONE_STRING_WATCHDOG_RESET:
;           .Db
FIX_TYPE_BASE:
FIX_TYPE_STOPPED:
           .Db      "Parar", 0
FIX_TYPE_STARTED:
           .Db      "Empezar", 0
FIX_TYPE_MOVING:
           .Db      "La mudanza", 0
FIX_TYPE_MOVED:
           .Db      "Cambiar de lugar", 0
FIX_TYPE_SPEEDING:
           .Db      "El exceso de velocidad", 0
FIX_TYPE_LOCATE:
           .Db      "LOCATE", 0
```

```
LAST_GOOD_FIX_STRING:
        .Db     ", Último buen  ", \
                0xfd, (2 < <2) + (LAST_GOOD_FIX_LATITUDE) alto, bajo
(LAST_GOOD_FIX_LATITUDE)," ", \
                0xfd, ((- de FLDLEN_GPRMC_LATITUDE2)<<2) + alto (+ de
LAST_GOOD_FIX_LATITUDE2), bajo (+ de LAST_GOOD_FIX_LATITUDE2)," ", \
                0xfd, (FLDLEN_GPRMC_LATDIR < <2) + (LAST_GOOD_FIX_LATDIR) alto, bajo
(LAST_GOOD_FIX_LATDIR)," ", \
                0xfd, (3 < <2) + (LAST_GOOD_FIX_LONGITUDE) alto, bajo
(LAST_GOOD_FIX_LONGITUDE)," ", \
                0xfd, ((- de FLDLEN_GPRMC_LONGITUDE3)<<2) + alto (+ de
LAST_GOOD_FIX_LONGITUDE3), bajo (+ de LAST_GOOD_FIX_LONGITUDE3)," ", \
                0xfd, (FLDLEN_GPRMC_LONGDIR < <2) + alto (LAST_GOOD_FIX_LONGDIR), nivel
más bajo (LAST_GOOD_FIX_LONGDIR)", AT ", \
                0xfd, (2 < <2) + (LAST_GOOD_FIX_TIME) alto, bajo (LAST_GOOD_FIX_TIME),":", \
                0xfd, (2 < <2) + alto (+ de LAST_GOOD_FIX_TIME2), bajo (+ de
LAST_GOOD_FIX_TIME2),":", \
                0xfd, (2 < <2) + alto (+ de LAST_GOOD_FIX_TIME4), bajo (+ de
LAST_GOOD_FIX_TIME4), la "UTC", 0

GPS_MATCH_PATTERN:
        .Db     Nivel más bajo (GPS_PARSE_GPRMC), máxima (GPS_PARSE_GPRMC), "$ GPRMC", 0,
\
                Nivel más bajo (GPS_PARSE_GPGGA), máxima (GPS_PARSE_GPGGA), "$ GPGGA", 0, \
                Nivel más bajo (GPS_PARSE_OTHER), máxima (GPS_PARSE_OTHER), 0

GPRMC_PARSE_STRING:
        .Db     1, (FLDLEN_GPRMC_TIME < <2) + (FLDADR_GPRMC_TIME) alto, bajo
(FLDADR_GPRMC_TIME), \
```

```
                2, (FLDLEN_GPRMC_FIXVALID < <2) + (FLDADR_GPRMC_FIXVALID) alto, bajo
(FLDADR_GPRMC_FIXVALID), \
                3, (FLDLEN_GPRMC_LATITUDE < <2) + (FLDADR_GPRMC_LATITUDE) alto, bajo
(FLDADR_GPRMC_LATITUDE), \
                4, (FLDLEN_GPRMC_LATDIR < <2) + (FLDADR_GPRMC_LATDIR) alto, bajo
(FLDADR_GPRMC_LATDIR), \
                5, (FLDLEN_GPRMC_LONGITUDE < <2) + (FLDADR_GPRMC_LONGITUDE) alto,
bajo (FLDADR_GPRMC_LONGITUDE), \
                6, (FLDLEN_GPRMC_LONGDIR < <2) + (FLDADR_GPRMC_LONGDIR) alto, bajo
(FLDADR_GPRMC_LONGDIR), \
                7, (FLDLEN_GPRMC_SPEED < <2) + alto (FLDADR_GPRMC_SPEED), bajo
(FLDADR_GPRMC_SPEED), \
                8, (FLDLEN_GPRMC_COURSE < <2) + (FLDADR_GPRMC_COURSE) alto, bajo
(FLDADR_GPRMC_COURSE), \          9, (FLDLEN_GPRMC_DATE < <2) +
(FLDADR_GPRMC_DATE) alto, bajo (FLDADR_GPRMC_DATE), \
                0xff
GPGGA_PARSE_STRING:
        .Db     6, (FLDLEN_GPGGA_FIXVALID < <2) + (FLDADR_GPGGA_FIXVALID) alto, bajo
(FLDADR_GPGGA_FIXVALID), \
                7, (FLDLEN_GPGGA_SATS < <2) + (FLDADR_GPGGA_SATS) altos, bajo
(FLDADR_GPGGA_SATS), \
                9, (FLDLEN_GPGGA_ALT < <2) + (FLDADR_GPGGA_ALT) alto, bajo
(FLDADR_GPGGA_ALT), \
                0xff

; Usar para SETSPEED, SETTRACK, y SETPOWER
SETTINGS_PARSE_STRING:
        .Db     1,(8<<2) + (PARSE_FIELDS) altos, bajo (PARSE_FIELDS), \
                2,(8<<2) + alto (+ de PARSE_FIELDS8), bajo (+ de PARSE_FIELDS8), \
                3,(8<<2) + alto (+ de PARSE_FIELDS16), bajo (+ de PARSE_FIELDS16), \
                4,(8<<2) + alto (+ de PARSE_FIELDS24), bajo (+ de PARSE_FIELDS24), \
```

```
                    5,(8<<2) + alto (+ de PARSE_FIELDS32), bajo (+ de PARSE_FIELDS32), \
                    6,(8<<2) + alto (+ de PARSE_FIELDS40), bajo (+ de PARSE_FIELDS40), \
                    7,(8<<2) + alto (+ de PARSE_FIELDS48), bajo (+ de PARSE_FIELDS48), \
                    8,(8<<2) + alto (+ de PARSE_FIELDS56), bajo (+ de PARSE_FIELDS56), \

SETSPEED_DECODE_STRING:
        .Db     + de (EEADR_SPEED_LIMIT) alto (1 < < DECODE_SETTING_ONE_BYTE), \
                Bajo (EEADR_SPEED_LIMIT), \
                + de (EEADR_GPS_MOVING_THRESHOLD) alto (1 < <
DECODE_SETTING_ONE_BYTE), \
                Bajo (EEADR_GPS_MOVING_THRESHOLD), \
                + de (EEADR_GPS_MOVING_RETRY_THRESHOLD) alto (1 < <
DECODE_SETTING_ONE_BYTE), \
                Bajo (EEADR_GPS_MOVING_RETRY_THRESHOLD), \
SETTRACK_DECODE_STRING:
        .Db     + de (EEADR_TRACK_STOPPED_FIX_INTERVAL) alto (1 < <
DECODE_SETTING_TWO_BYTE), \
                Bajo (EEADR_TRACK_BLOCKED_FIX_INTERVAL), \
                + de (EEADR_TRACK_BLOCKED_NOTIFY_DELAY) alto (1 < <
DECODE_SETTING_ONE_BYTE), \
                Bajo (EEADR_TRACK_BLOCKED_NOTIFY_DELAY), \
                + de (EEADR_TRACK_MOVING_FIX_INTERVAL) alto (1 < <
DECODE_SETTING_TWO_BYTE), \
                Bajo (EEADR_TRACK_MOVING_FIX_INTERVAL), \
                + de (EEADR_TRACK_MOVING_NOTIFY_FREQ) alto (1 < <
DECODE_SETTING_ONE_BYTE), \
                Bajo (EEADR_TRACK_MOVING_NOTIFY_FREQ), \
                + de (EEADR_TRACK_MIN_DISPLACEMENT) alto (1 < <
DECODE_SETTING_TWO_BYTE), \
                Bajo (EEADR_TRACK_MIN_DISPLACEMENT), \
        0
```

```
SETPOWER_DECODE_STRING:
        .Db     + de (EEADR_POWERSAVE_PHONE_OFF_INTERVAL) alto (1 < <
DECODE_SETTING_TWO_BYTE), \
                Bajo (EEADR_POWERSAVE_PHONE_OFF_INTERVAL), \
                + de (EEADR_POWERSAVE_PHONE_ON_INTERVAL) alto (1 < <
DECODE_SETTING_TWO_BYTE), \
                Bajo (EEADR_POWERSAVE_PHONE_ON_INTERVAL), \
                + de (EEADR_GPS_FOUR_SAT_WAIT_TIME) alto (1 < <
DECODE_SETTING_ONE_BYTE_X4), \
                Bajo (EEADR_GPS_FOUR_SAT_WAIT_TIME), \
                + de (EEADR_GPS_FIX_WAIT_TIME) alto (1 < <
DECODE_SETTING_ONE_BYTE_X4), \
                Bajo (EEADR_GPS_FIX_WAIT_TIME), \
                + de (EEADR_BLOCKED_GPS_FIX_WAIT_TIME) alto (1 < <
DECODE_SETTING_ONE_BYTE_X4), \
                Bajo (EEADR_BLOCKED_GPS_FIX_WAIT_TIME), \
                0
EEPROM_DEFAULTS:
        .Db     + de (EEADR_PASSWORD) alto (4 < < 2 ), \        Nivel más bajo
(EEADR_PASSWORD), " generales", 0, \
                + de (EEADR_DEVICENAME) alto (1 < < 2 ), \
                Nivel más bajo (EEADR_DEVICENAME), 0, \
                + de (EEADR_EMAIL_ADDR) alto (1 < < 2 ), \
                Nivel más bajo (EEADR_EMAIL_ADDR), 0, \
                + de (EEADR_GPS_FOUR_SAT_WAIT_TIME) alto (21 < < 2 ), \
                Bajo (EEADR_GPS_FOUR_SAT_WAIT_TIME), \
                DEFAULT_FOUR_SAT_WAIT_TIME, \
                DEFAULT_GPS_FIX_WAIT_TIME, \
                DEFAULT_BLOCKED_GPS_FIX_WAIT_TIME, \
                DEFAULT_SPEED_LIMIT, \
```

```
                    DEFAULT_GPS_MOVING_THRESHOLD, \
                    DEFAULT_GPS_MOVING_RETRY_THRESHOLD, \
                    Alto (DEFAULT_TRACK_STOPPED_FIX_INTERVAL), \
                    Bajo (DEFAULT_TRACK_STOPPED_FIX_INTERVAL), \
                    DEFAULT_TRACK_STOPPED_NOTIFY_DELAY, \
                    Alto (DEFAULT_TRACK_BLOCKED_FIX_INTERVAL), \
                    Bajo (DEFAULT_TRACK_BLOCKED_FIX_INTERVAL), \
                    DEFAULT_TRACK_BLOCKED_NOTIFY_DELAY, \                Alto
(DEFAULT_TRACK_MOVING_FIX_INTERVAL), \
                    Bajo (DEFAULT_TRACK_MOVING_FIX_INTERVAL), \
                    DEFAULT_TRACK_MOVING_NOTIFY_FREQ, \
                    Alto (DEFAULT_POWERSAVE_PHONE_OFF_INTERVAL), \
                    Bajo (DEFAULT_POWERSAVE_PHONE_OFF_INTERVAL), \
                    Alto (DEFAULT_POWERSAVE_PHONE_ON_INTERVAL), \
                    Bajo (DEFAULT_POWERSAVE_PHONE_ON_INTERVAL), \
                    Alto (DEFAULT_TRACK_MIN_DISPLACEMENT), \
                    Bajo (DEFAULT_TRACK_MIN_DISPLACEMENT), \
                    + de (EEADR_PHONE_NUMBER) alto (1 < < 2 ), \
                    Nivel más bajo (EEADR_PHONE_NUMBER), 0, \                + de
(EEADR_EMAIL_ADDR) alto (1 < < 2 ), \
                    Nivel más bajo (EEADR_EMAIL_ADDR), 0, \
                    + de (EEADR_CHECKCODE) alto (2 < < 2 ), \            Nivel más bajo
(EEADR_CHECKCODE), "Milla", 0
; EOF
}
```

APENDICE D

CÓDIGO FUENTE DEL SISTEMA LOCAL GPS –GOOGLE MAPS.

```
Option Explicit

'Private WithEvents PuertoCom As MSComm

Dim M_Pro() As ProporcionesControl
Dim LFecha, LHora, LLatitud, LLongitud, LAltura As String
Dim LLatG, LLatM, LLatD As String
Dim LLonG, LLonM, LLonD As String
Dim PorcLat, PorcLon As Double
Dim ContFila As Integer

Private vPuntero As Integer

Private sPuerto As String
Private sBaudios As String
Private sParidad As String
Private sBitsCar As String
Private sBitsParada As String
Private nControlFlujo As HandshakeConstants 'entero
Private nModoLectura As InputModeConstants  'entero

'Tamaño de las colas de recepción y de transmisión
Const COLARX As Integer = 4096
Const COLATX As Integer = 4096

Sub Camb_Map()

End Sub

Private Sub cmdEnviar_Click()

   Call imprime

End Sub

Private Sub Command1_Click()
   'Regresar todo al valor inicial
   Me.FrmComunicacion.Visible = False
End Sub


Private Sub ConfigParams_Click()
   If PuertoCom.PortOpen = True Then
   MsgBox "Cierre primero la conexión"
   Exit Sub
  End If
```

```
  ' Visualizar en los controles los parámetros de comunicación
  ' actuales. Inicialmente fueron recuperados del registro
  frmConfiguración.sPuerto = sPuerto
  frmConfiguración.sBaudios = sBaudios
  frmConfiguración.sParidad = sParidad
  frmConfiguración.sBitsCar = sBitsCar
  frmConfiguración.sBitsParada = sBitsParada
  frmConfiguración.nControlFlujo = nControlFlujo
  frmConfiguración.nModoLectura = nModoLectura

  ' Visualizar el formulario Configuración
  frmConfiguración.Show vbModal, Me
  ' Si se pulsó el botón Aceptar ...
  If frmConfiguración.bDatosVálidos = True Then
    ' Asignar los nuevos valores a las variables correspondientes
    sPuerto = frmConfiguración.sPuerto
    sBaudios = frmConfiguración.sBaudios
    sParidad = frmConfiguración.sParidad
    sBitsCar = frmConfiguración.sBitsCar
    sBitsParada = frmConfiguración.sBitsParada
    nControlFlujo = frmConfiguración.nControlFlujo
    nModoLectura = frmConfiguración.nModoLectura
    ' Descargar el formulario
    Unload frmConfiguración

    ' Establecer la conexión con los parámetros establecidos
    If EstablecerConexion = True Then
      ' Habilitar el botón de Enviar
      Me.TEnvio = True
      cmdEnviar.Enabled = True
    End If
    Me.txtRX.Text = ""
    Me.TRecibo.Enabled = True
    Me.TPuntero.Enabled = True
  End If
End Sub


Private Sub Form_Resize()
    Dim C As Integer
    For C = 0 To Controls.Count - 1
        If TypeOf Controls(C) Is Timer Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is CommonDialog Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is MSComm Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is Winsock Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is ImageList Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is Line Then
            Controls(C).X1 = M_Pro(C).ProporcionesIzquierda * ScaleWidth
            Controls(C).Y1 = M_Pro(C).ProporcionesSuperior * ScaleHeight
            Controls(C).X2 = M_Pro(C).ProporcionesAnchura * ScaleWidth
```

```
            Controls(C).Y2 = M_Pro(C).ProporcionesAltura * ScaleHeight
        ElseIf TypeOf Controls(C) Is Menu Then
            'No hacer nada
        Else
            Controls(C).Move M_Pro(C).ProporcionesIzquierda * ScaleWidth,
M_Pro(C).ProporcionesSuperior * ScaleHeight, M_Pro(C).ProporcionesAnchura *
ScaleWidth, M_Pro(C).ProporcionesAltura * ScaleHeight
        End If
    Next C

End Sub

Sub Iniciar_M()
    Dim C As Integer
    For C = 0 To Controls.Count - 1
        If TypeOf Controls(C) Is Timer Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is CommonDialog Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is MSComm Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is Winsock Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is ImageList Then
            'No hacer nada
        ElseIf TypeOf Controls(C) Is Line Then
            With M_Pro(C)
                .ProporcionesIzquierda = Controls(C).X1 / ScaleWidth
                .ProporcionesSuperior = Controls(C).Y1 / ScaleHeight
                .ProporcionesAnchura = Controls(C).X2 / ScaleWidth
                .ProporcionesAltura = Controls(C).Y2 / ScaleHeight
            End With
        ElseIf TypeOf Controls(C) Is Menu Then
            'No hacer nada
        Else
            With M_Pro(C)
                .ProporcionesAnchura = Controls(C).Width / ScaleWidth
                .ProporcionesAltura = Controls(C).Height / ScaleHeight
                .ProporcionesIzquierda = Controls(C).Left / ScaleWidth
                .ProporcionesSuperior = Controls(C).Top / ScaleHeight
            End With
        End If
    Next C
End Sub

Private Sub ConexionCortar_Click()
    ' Si la conexión está cerrada, la orden Cortar
    ' está inhabilitada
    Dim p As Integer
    Me.TEnvio = False
    Me.TRecibo = False
    'For p = 0 To 33
        'Me.lblPuntos(p) = "0"
        'Bandera(p) = 0
        'Bandera1(p) = 2
```

```
    'Next p
    'M = 1
    Terminar
    Call CortarConexion
    cmdEnviar.Enabled = False
    Me.TPuntero.Enabled = False
    Puntero.Picture = LoadPicture(App.Path & "\Pun2.ico")
End Sub

Private Sub ConexionEstablecer_Click()
    ' Si la conexión ya estaba establecida, la orden Establecer
    ' está inhabilitada
    If EstablecerConexion = True Then
        Me.TEnvio = True
        cmdEnviar.Enabled = True

    End If
    Me.txtRX.Text = ""
    Me.TRecibo.Enabled = True
    Me.TPuntero.Enabled = True
End Sub

Private Sub MkB1_Change(Index As Integer)
    If Index = 0 Then
        Me.MkB1(1).Text = Me.MkB1(0).Text
    ElseIf Index = 2 Then
        Me.MkB1(3).Text = Me.MkB1(2).Text
    End If
End Sub

Private Sub MkB2_Change(Index As Integer)
    If Index = 0 Then
        Me.MkB2(2).Text = Me.MkB2(0).Text
    ElseIf Index = 1 Then
        Me.MkB2(3).Text = Me.MkB2(1).Text
    End If

End Sub

Private Sub mnAcerca_Click()
    frmAbout.Show vbModal, Me
End Sub

Private Sub mnAdministrador_Click()
    If FrmComunicacion.Visible Then
        FrmComunicacion.Visible = False
    Else
        FrmComunicacion.Visible = True
    End If
End Sub

Private Sub mnMapa_Click()
    Call Camb_Map
End Sub
```

```
Private Sub mnSalir_Click()
   If PuertoCom.PortOpen Then
      CortarConexion
   End If
   Unload Me
End Sub

Private Sub mnUsuarios_Click()
   Administrador.Show vbModal, Me
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
   If Button.Caption = "Salir" Then
      Call mnSalir_Click
   ElseIf Button.Caption = "Conectar" Then
      Call ConexionEstablecer_Click
   ElseIf Button.Caption = "Desconectar" Then
      Call ConexionCortar_Click
   ElseIf Button.Caption = "Confg. Comm" Then
      Call ConfigParams_Click
   ElseIf Button.Caption = "Confg. Moden" Then
      Call mnMapa_Click
   ElseIf Button.Caption = "Administrador" Then
      Call mnAdministrador_Click
   ElseIf Button.Caption = "Usuarios" Then
      Call mnUsuarios_Click
   End If

End Sub

' RESPONDER A LOS EVENTOS GENERADOS EN EL PUERTO
' ----------------------------------------------

Private Sub PuertoCom_OnComm()
  Dim sEvento As String, sError As String, sRecibida As String

  ' Controlar cada evento o error escribiendo
  ' código en cada caso
  Select Case PuertoCom.CommEvent
    ' Eventos
    Case comEvCD
      sEvento = "Cambio en la línea CD."
    Case comEvCTS
      sEvento = "Cambio en la línea CTS."
    Case comEvDSR
      sEvento = "Cambio en la línea DSR."
    Case comEvRing
      sEvento = "Cambio en el indicador de llamadas."
    Case comEvReceive
      sEvento = "Recibido(s) " & PuertoCom.RThreshold & _
              " carácter/caracteres."
      ' Leer caracteres del puerto
      If LeerCarsPuerto(sRecibida) > 0 Then
        txtRX.Text = txtRX.Text & sRecibida
      End If
```

```
    Case comEvSend
      sEvento = "Hay SThreshold = " & PuertoCom.SThreshold & _
              " carácter/caracteres en el búfer de transmisión."
    Case comEvEOF
      sEvento = "Se ha encontrado un carácter EOF en la entrada."

    ' Errores
    Case comBreak
      sError = "Se ha recibido una interrupción."
    Case comEventFrame
      sError = "Error de trama."
    Case comEventOverrun
      sError = "Datos perdidos."
    Case comEventRxOver
      sError = "Desbordamiento del búfer de recepción."
    Case comEventRxParity
      sError = "Error de paridad."
    Case comEventTxFull
      sError = "Búfer de transmisión lleno."
    Case comEventDCB
      sError = "Error inesperado al recuperar el DCB."
  End Select

  If Not IsEmpty(sEvento) Then
    StatusBar1.SimpleText = sEvento
  ElseIf Not IsEmpty(sError) Then
    Dim vr As VbMsgBoxResult
    Beep
    sError = sError & vbNewLine & "Aceptar para ignorar. " & _
             "Cancelar para salir"
    vr = MsgBox(sError, vbOKCancel + vbExclamation, App.Title)

    If vr = vbCancel Then
      ' Cerrar el puerto
      PuertoCom.PortOpen = False
      ConexionEstablecer.Enabled = True
      ConexionCortar.Enabled = False
      'UtilsEnviarFichero.Enabled = False
    End If
  End If
End Sub

' INTERFAZ DE COMUNICACIONES
' --------------------------

Private Sub Iniciar()
  'No se verifica que los datos obtenidos sean buenos
  sPuerto = GetSetting(App.Title, "MSComm", "Puerto", "1")
  sBaudios = GetSetting(App.Title, "MSComm", "Baudios", "9600")
  sBitsParada = GetSetting(App.Title, "MSComm", "BitsParada", "1")
  sParidad = GetSetting(App.Title, "MSComm", "Paridad", "None - Ninguna")
  sBitsCar = GetSetting(App.Title, "MSComm", "BitsCar", "8")
  nControlFlujo = GetSetting(App.Title, "MSComm", "ControlFlujo", "0")
  nModoLectura = GetSetting(App.Title, "MSComm", "ModoLectura", "0")
```

```
End Sub

Private Sub Terminar()
  SaveSetting App.Title, "MSComm", "Puerto", sPuerto
  SaveSetting App.Title, "MSComm", "Baudios", sBaudios
  SaveSetting App.Title, "MSComm", "BitsParada", sBitsParada
  SaveSetting App.Title, "MSComm", "Paridad", sParidad
  SaveSetting App.Title, "MSComm", "BitsCar", sBitsCar
  SaveSetting App.Title, "MSComm", "ControlFlujo", nControlFlujo
  SaveSetting App.Title, "MSComm", "ModoLectura", nModoLectura
End Sub

Private Function EstablecerConexion() As Boolean
  On Error Resume Next
  With PuertoCom
    ' Cerrar el control si estuviera abierto
    If .PortOpen = True Then .PortOpen = False
    ' Especificar el puerto COM que se desea abrir
    .CommPort = sPuerto ' número del puerto (1, 2, ...)
    ' Establecer el tamaño de las colas de recepción y transmisión
    .InBufferSize = COLARX  ' cola de recepción
    .OutBufferSize = COLATX ' cola de transmisión
    ' Limpiar las colas Rx y Tx
    .InBufferCount = 0
    .OutBufferCount = 0

    ' Establecer los parámetros de la comunicación
    Dim sSettings As String
    ' Baudios, paridad, número de bits de datos y de parada
    ' Longitud del bit de paro:
    sSettings = sBaudios & "," & Left(sParidad, 1) & "," & _
                sBitsCar & "," & sBitsParada
    .settings = sSettings

    ' Establecer el control de flujo
    .Handshaking = nControlFlujo
    ' Cómo se leerán los datos del puerto
    .InputMode = nModoLectura

    ' Caracteres que puede admitir el buffer de transmisión antes
    ' de que el control genere el evento OnComm.
    ' Su valor predeterminado es 0
    .SThreshold = 1
    ' Caracteres que se van recibir antes de que el control genere
    ' el evento OnComm. Su valor predeterminado es 0.
    .RThreshold = 1

    ' Abrir el puerto de comunicaciones
    .PortOpen = True
    If .PortOpen = False Then
      ' Error al abrir el puerto (verifique la configuración)
      Beep
      MsgBox "Error: No se puede abrir el puerto COM" & _
             sPuerto, vbOKOnly + vbCritical, App.Title
      If sBitsParada > "1" Then
```

```
        MsgBox _
          "1 bit en cualquier longitud de carácter, o bien " & _
          vbCrLf & "1.5 bits en longitud de carácter 5, y " & _
          vbCrLf & "2 bits en longitud de carácter 6 a 8", _
          vbOKOnly + vbInformation, App.Title
      End If
      EstablecerConexion = False
      Exit Function
    End If
  End With
  ' El puerto se abrió con éxito
  EstablecerConexion = True
  StatusBar1.SimpleText = "Puerto de comunicaciones abierto"
  ' Habilitar/Inhabilitar órdenes de ls menús
  ConexionEstablecer.Enabled = False
  ConexionCortar.Enabled = True
  'UtilsEnviarFichero.Enabled = True
End Function

Private Function LeerCarsPuerto(ByRef sRecibida As String) As Long
  sRecibida = PuertoCom.Input
  LeerCarsPuerto = Len(sRecibida)
End Function

Private Function EscribirCarsPuerto(str As String) As Boolean
  PuertoCom.Output = str
  EscribirCarsPuerto = True
End Function

Private Function CortarConexion() As Boolean
  If ConexionCortar.Enabled = True Then
    Dim bTiempoSobrepasado As Boolean, Tiempo As Long

    ' Establecer un periodo de 10 segundos a partir de la hora
    ' actual antes de cerrar el puerto, por seguridad
    bTiempoSobrepasado = False
    Tiempo = Now

    StatusBar1.SimpleText = "Cerrando la conexión..."
    While PuertoCom.OutBufferCount > 0
      ' Permitir procesar mensajes pendientes
      DoEvents
      If DateDiff("s", Now, Tiempo) > 10 Or _
        bTiempoSobrepasado = True Then
        Dim vr As VbMsgBoxResult
        vr = MsgBox("Datos no enviados", vbAbortRetryIgnore, _
                App.Title)
        Select Case vr
          ' Intentar enviar los datos durante otros 10 segundos
          Case vbRetry
            Tiempo = Now
          ' Ignorar el tiempo límite
          Case vbIgnore
            bTiempoSobrepasado = True
          Case vbAbort
```

```
            StatusBar1.SimpleText = ""
            CortarConexion = False
            Exit Function
        End Select
      End If
    Wend
    ' Tx vacío. Cerrar el puerto.
    If PuertoCom.PortOpen Then
       PuertoCom.PortOpen = False
    End If
    ConexionEstablecer.Enabled = True
    ConexionCortar.Enabled = False
    'UtilsEnviarFichero.Enabled = False
  End If
  StatusBar1.SimpleText = "Conexión concluida"
  CortarConexion = True

  'Conexion Concluida

End Function

Private Sub TPuntero_Timer()
    If vPuntero = 0 Then
       Puntero.Picture = LoadPicture(App.Path & "\Pun0.cur")
       vPuntero = 1
    ElseIf vPuntero = 1 Then
       Puntero.Picture = LoadPicture(App.Path & "\Pun1.ico")
       vPuntero = 0
    Else
       Puntero.Picture = LoadPicture(App.Path & "\Pun2.ico")
       vPuntero = 0
    End If
End Sub

Private Sub TRecibo_Timer()
    If Me.txtRX.Text = "" Then

    Else
       If Not Val_Ascii(Me.txtRX.Text) Then
          Me.txtRX.Text = ""
          Exit Sub
       End If
       If Len(Me.txtRX.Text) > 55 Then
          Me.txtRX.Text = ""
       Else
          Call localizar(Me.txtRX.Text)
       End If
    End If
    'Call imprime
End Sub

Sub localizar(cadena As String)
    Dim LinT, Opc As String
    Dim MPos, CntC As Integer
    Dim Kar() As String
```

```
   ReDim Kar(Len(cadena))

   List2.AddItem (cadena)

   If Len(cadena) < 41 Then
      Exit Sub
   End If

   For CntC = 1 To Len(cadena)
       Kar(CntC - 1) = Mid(cadena, CntC, 1)
   Next CntC

   For CntC = 0 To Len(cadena) - 1
      If Kar(CntC) = "F" Then
         LFecha = Kar(CntC + 1) & Kar(CntC + 2) & Kar(CntC + 3) & Kar(CntC + 4) &
Kar(CntC + 5) & Kar(CntC + 6) & Kar(CntC + 7) & Kar(CntC + 8)
         CntC = 9

      ElseIf Kar(CntC) = "H" Then
         LHora = Kar(CntC + 1) & Kar(CntC + 2) & Kar(CntC + 3) & Kar(CntC + 4) & Kar(CntC
+ 5) & Kar(CntC + 6) & Kar(CntC + 7) & Kar(CntC + 8)
         CntC = 19
On Error GoTo banda2
      ElseIf Kar(CntC) = "Y" Then
            LLatG = Kar(CntC + 1)
            If Kar(CntC + 2) = "g" Then
               CntC = CntC + 2
            Else
               LLatG = LLatG & Kar(CntC + 2)
               CntC = CntC + 3
            End If
            LLatM = Kar(CntC + 1)
            If Kar(CntC + 2) = "." Then
               CntC = CntC + 2
            Else
               LLatM = LLatM & Kar(CntC + 2)
               CntC = CntC + 3
            End If
            LLatD = Kar(CntC + 1)
            If Kar(CntC + 2) = "m" Then
               CntC = CntC + 2
            Else
               LLatD = LLatD & Kar(CntC + 2)
               If Kar(CntC + 3) = "m" Then
                  CntC = CntC + 3
               Else
                  LLatD = LLatD & Kar(CntC + 3)
                  If Kar(CntC + 4) = "m" Then
                     CntC = CntC + 4
                  Else
                     LLatD = LLatD & Kar(CntC + 4)
                     CntC = CntC + 5
                  End If
               End If
            End If
```

```
        LLatitud = Kar(CntC + 1)

ElseIf Kar(CntC) = "X" Then
        LLonG = Kar(CntC + 1)
        If Kar(CntC + 2) = "g" Then
           CntC = CntC + 2
        Else
           LLonG = LLonG & Kar(CntC + 2)
           CntC = CntC + 3
        End If
        LLonM = Kar(CntC + 1)
        If Kar(CntC + 2) = "." Then
           CntC = CntC + 2
        Else
           LLonM = LLonM & Kar(CntC + 2)
           CntC = CntC + 3
        End If
        LLonD = Kar(CntC + 1)
        If Kar(CntC + 2) = "m" Then
           CntC = CntC + 2
        Else
           LLonD = LLonD & Kar(CntC + 2)
           If Kar(CntC + 3) = "m" Then
              CntC = CntC + 3
           Else
              LLonD = LLonD & Kar(CntC + 3)
              If Kar(CntC + 4) = "m" Then
                 CntC = CntC + 4
              Else
                 LLonD = LLonD & Kar(CntC + 4)
                 CntC = CntC + 5
              End If
           End If
        End If
        LLongitud = Kar(CntC + 1)

ElseIf Kar(CntC) = "A" Then
        LAltura = Kar(CntC + 1)
        If Kar(CntC + 2) = "m" Then
           CntC = CntC + 2
        Else
           LAltura = LAltura & Kar(CntC + 2)
           If Kar(CntC + 3) = "m" Then
              CntC = CntC + 3
           Else
              LAltura = LAltura & Kar(CntC + 3)
              If Kar(CntC + 4) = "m" Then
                 CntC = CntC + 4
              Else
                 LAltura = LAltura & Kar(CntC + 4)
                 CntC = CntC + 5
              End If
           End If
        End If
     End If
```

```
    Next CntC
    Call imprime
banda2:
    'FDD/MM/AA HHH:MM:SS Yººg"".####m(N/S) Xººg"".####m(E/O) A####m
End Sub

Sub imprime()
    Dim LtG, LtM, LnG, LnM As Integer
    Dim LtS As Double
    Dim LnS As Double
    List1.Clear
    List1.AddItem ("Fecha = (" & LFecha & ")")
    List1.AddItem ("Hora = (" & LHora & ")")
    List1.AddItem ("------------------------------")
    List1.AddItem ("Latitud = (" & LLatitud & ")")
    List1.AddItem ("Lat. G = (" & LLatG & ")")
    List1.AddItem ("Lat. M = (" & LLatM & ")")
    List1.AddItem ("Lat. D = (" & LLatD & ")")
    List1.AddItem ("------------------------------")
    List1.AddItem ("Longitud = (" & LLongitud & ")")
    List1.AddItem ("Lon. G = (" & LLonG & ")")
    List1.AddItem ("Lon. M = (" & LLonM & ")")
    List1.AddItem ("Lon. D = (" & LLonD & ")")
    List1.AddItem ("------------------------------")
    List1.AddItem ("Altura = (" & LAltura & ")")
    Me.txtRX.Text = ""

    If LLonM & LLonD <> "00" Then
        'Calculo de Latitud
        LtS = Val(Mid((Me.MkB1(2).Text), 4, 7)) - Val(Mid((Me.MkB1(0).Text), 4, 7))
        LtS = (Val(LLatM & "." & LLatD) - Val(Mid((Me.MkB1(0).Text), 4, 7))) / LtS
        PorcLat = Format(LtS, "0.##") * 1
        'Calculo de Longitud
        LnS = Val(Mid((Me.MkB2(0).Text), 4, 7)) - Val(Mid((Me.MkB2(1).Text), 4, 7))     'Invertir
((Me.MkB2(1).Text) X (Me.MkB2(0).Text)) Y ((Me.MkB2(0).Text) (Me.MkB2(1).Text)
        LnS = (Val(LLonM & "." & LLonD) - Val(Mid((Me.MkB2(1).Text), 4, 7))) / LnS
'Cambiar (Me.MkB2(1).Text) X (Me.MkB2(0).Text)
        PorcLon = Format(LnS, "0.##") * 1
    End If
    Me.Puntero.Left = (Me.Hubikt.Width - 400) - Int(PorcLon * (Me.Hubikt.Width - 400))
    Me.Puntero.Top = Int(PorcLat * (Me.Hubikt.Height + 400))

    LtG = Int(Val(LLatG))
    LtM = Int(Val(LLatM))
    LtS = Format(Val("0." & LLatD) * 60, "00.##")

    LnG = Int(Val(LLonG))
    LnM = Int(Val(LLonM))
    LnS = Format(Val("0." & LLonD) * 60, "00.##")

    MSF1.Rows = ContFila
    MSF1.TextMatrix(ContFila - 1, 0) = LHora
    MSF1.TextMatrix(ContFila - 1, 1) = LtG & "º" & LtM & "'" & LtS & Chr(34)
    MSF1.TextMatrix(ContFila - 1, 2) = LnG & "º" & LnM & "'" & LnS & Chr(34)
```

```
    If LLongitud = "E" Then
       LLongitud = "ESTE"
    ElseIf LLongitud = "O" Then
       LLongitud = "OESTE"
    End If

    If LLatitud = "S" Then
       LLatitud = "SUR"
    ElseIf LLatitud = "N" Then
       LLatitud = "NORTE"
    End If
    Randomize
    LAltura = Format(350 + (50 * Rnd), "0000")
    If LAltura <> "" Then
       LAltura = Mid(LAltura, 1, 2) & "." & Mid(LAltura, 3, Len(LAltura) - 1)
    End If
    Me.Label6(0).Caption = LFecha
    Me.Label6(1).Caption = LHora
    Me.Label6(2).Caption = Val(LAltura) & " m"
    Me.Label6(3).Caption = LLatitud
    Me.Label6(4).Caption = LtG & "º" & LtM & "'" & LtS & Chr(34)
    Me.Label6(5).Caption = LLongitud
    Me.Label6(6).Caption = LnG & "º" & LnM & "'" & LnS & Chr(34)

    Call Ultimo
    ContFila = ContFila + 1
End Sub

Function Val_Ascii(cadena As String) As Boolean
    Dim ran1_1, ran1_2 As Integer
    Dim ran2_1, ran2_2 As Integer
    Dim ran3_1, ran3_2 As Integer
    Dim AscII As Integer
    Dim lc As Integer
    Dim i As Integer
    Dim caracter As String
    Dim resp As String
    'Rango de números desde 0 - 9 y signos (.) , (/) y (:)
    ran1_1 = 46
    ran1_2 = 58
    'Rango de Letras Mayusculas desde A - Z
    ran2_1 = 65
    ran2_2 = 90
    'Rango de Letras Minusculas desde a - z
    ran3_1 = 97
    ran3_2 = 122
    'Longitud de Cadena de Caracteres
    lc = Len(cadena)

    For i = 1 To lc
       caracter = Mid(cadena, i, 1)
       AscII = Asc(caracter)
       If AscII = 32 Then
          resp = True "'¡Es un espacio")!!!..."
       ElseIf AscII >= ran1_1 And AscII <= ran1_2 Then
```

```
            resp = True "'¡Esta en rango (" & ran1_1 & " - " & ran1_2 & ")!!!..."
        ElseIf AscII >= ran2_1 And AscII <= ran2_2 Then
            resp = True "'¡Esta en rango (" & ran2_1 & " - " & ran2_2 & ")!!!..."
        ElseIf AscII >= ran3_1 And AscII <= ran3_2 Then
            resp = True "'¡Esta en rango (" & ran3_1 & " - " & ran3_2 & ")!!!..."
        Else
            Val_Ascii = False "'¡No está en rango!!!!..."
            Exit Function
        End If
    Next i
    Val_Ascii = True
End Function

Sub Ultimo()
    Dim clm As Integer
    clm = Me.List2.ListCount - 1
    Me.List2.Selected(clm) = True
    clm = ContFila
    Me.MSF1.SelectionMode = flexSelectionByRow
    MSF1.RowPosition(clm - 1) = 1

    MSF1.Row = 1
    MSF1.Col = 0
    MSF1.CellBackColor = &HC0E0FF
    MSF1.Col = 1
    MSF1.CellBackColor = &HC0E0FF
    MSF1.Col = 2
    MSF1.CellBackColor = &HC0E0FF

    MSF1.Row = 2
    MSF1.Col = 0
    MSF1.CellBackColor = &HFFFFFF
    MSF1.Col = 1
    MSF1.CellBackColor = &HFFFFFF
    MSF1.Col = 2
    MSF1.CellBackColor = &HFFFFFF
    MSF1.Row = 1
    MSF1.Col = 0
End Sub
```

APÉNDICE E

Código Fuente del Script Google Maps.

```
<iframe
 width="870"
 height="540"
 frameborder="0"
 scrolling="no"
 marginheight="0"
 marginwidth="0"

src="http://maps.google.es/maps?f=q&amp;source=s_q&amp;hl=es&amp;geocode=&amp;q
=02+14.9730+S+079+54.4034+W&amp;ie=UTF8&amp;t=h&amp;ll=-2.239096,-
79.902534&amp;spn=0.002316,0.005493&amp;z=14&amp;iwloc=near&amp;output=embed"
>
</iframe>
<br />
<small>
<a
href="http://maps.google.es/maps?f=q&amp;source=embed&amp;hl=es&amp;geocode=&a
mp;q=02+14.9730+S+079+54.4034+W&amp;ie=UTF8&amp;t=h&amp;ll=-2.239096,-
79.902534&amp;spn=0.002316,0.005493&amp;z=14&amp;iwloc=near"
style="color:#0000FF;text-align:left">Ver mapa mßs grande</a></small>
```

APÉNDICE F

Manual de Comandos AT-GSM.

Comandos AT y AT+:

Por ejemplo, si quieres enviar un sms via comandos AT, te conectas via hyperterminal (windows) o minicom (linux) al modem y envias el siguiente comando:

AT+CMGS="numero de telefono" + CR

el modem debera responder ">"

ahora introduces el texto a enviar + CONTROL-Z

Un resumen de los comandos para GSM es:

1 Comandos generales

a) AT+CGMI: Identificación del fabricante

b) AT+CGSN: Obtener número de serie

c) AT+CIMI: Obtener el IMSI.

d) AT+CPAS: Leer estado del modem

2. Comandos del servicio de red

a) AT+CSQ: Obtener calidad de la señal

b) AT+COPS: Selección de un operador

c) AT+CREG: Registrarse en una red

d) AT+WOPN: Leer nombre del operador

3. Comandos de seguridad:

a) AT+CPIN: Introducir el PIN

b) AT+CPINC: Obtener el número de reintentos que quedan

c) AT+CPWD: Cambiar password

4. Comandos para la agenda de teléfonos

a) AT+CPBR: Leer todas las entradas

b) AT+CPBF: Encontrar una entrada

c) AT+CPBW: Almacenar una entrada

d) AT+CPBS: Buscar una entrada

5. Comandos para SMS

a) AT+CPMS: Seleccionar lugar de almacenamiento de los SMS

b) AT+CMGF: Seleccionar formato de los mensajes SMS

c) AT+CMGR: Leer un mensaje SMS almacenado

d) AT+CMGL: Listar los mensajes almacenados

e) AT+CMGS: Enviar mensaje SMS

f) AT+CMGW: Almacenar mensaje en memoria

g) AT+CMSS: Enviar mensaje almacenado

h) AT+CSCA: Establecer el Centro de mensajes a usar

i) AT+ WMSC: Modificar el estado de un mensaje.