



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA Y ELECTRICIDAD Y COMPUTACIÓN**

“Almacenamiento masivo de datos de posicionamiento GPS en unidad flash  
USB para su posterior utilización con mapas de trayectoria compatibles con  
Google Earth”

**TESINA DE SEMINARIO**

PREVIA A LA OBTENCIÓN DE LOS TÍTULOS DE:

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA**  
**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y**  
**AUTOMATIZACION INDUSTRIAL**

PRESENTADA POR:

**Erwin José Segura Espinoza**

**Gonzalo Eloy Morán Tumbaco**

**GUAYAQUIL-ECUADOR**

**2010**

## AGRADECIMIENTO

A Dios, por siempre estar cerca, aun en mis errores;

A mis padres, Juan y Elena, porque de lo poco que han poseído mucho me han dado;

A mis maestros, por la noble tarea de guiarnos por la senda del conocimiento;

Al MSc. Carlos Valdivieso, por su guía y colaboración en la consecución de este proyecto.

Erwin José Segura Espinoza

## DEDICATORIA

A Dios por guiar nuestros pasos y ayudarnos a superar los obstáculos que se presentan a lo largo de la vida;

A mis padres, Juan y Elena, por la educación en el hogar que me han dado;

A mi Abuela Germania y a mi tía María, por toda su ayuda mientras estudié.

Erwin José Segura Espinoza

## AGRADECIMIENTO

A Dios, por darnos Vida y Salud.

A mis padres, por el ejemplo de lucha y perseverancia para salir adelante.

A todos los profesores por su guía y colaboración durante el transcurso de esta carrera.

A todas aquellas personas que de una u otra manera han hecho posible que se cumpla esta meta.

Gonzalo Eloy Morán Tumbaco

## DEDICATORIA

A Dios por haberme dado la oportunidad de vivir.

A mis padres, Luisa y Jorge, que con mucho esfuerzo y sacrificio siempre estuvieron pendientes que tuviera lo necesario.

A mis profesores esperando que sirva como estímulo en la formación de nuevos profesionales

A mis compañeros de trabajo que me ayudaron a cumplir mis jornadas laborales sin descuidar mis estudios.

Gonzalo Eloy Morán Tumbaco

# TRIBUNAL DE SUSTENTACIÓN



MSc. Carlos Valdivieso

Profesor de Seminario de Graduación



MSc. Hugo Villavicencio.

Delegado del Decano

## DECLARACIÓN EXPRESA

“La responsabilidad por los derechos, ideas y doctrinas expuestas en esta tesina, nos corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)

A handwritten signature in black ink, appearing to read "Erwin Segura", written over a horizontal line.

Erwin José Segura Espinoza.

A handwritten signature in black ink, appearing to read "Gonzalo Morán", written over a horizontal line.

Gonzalo Eloy Morán Tumbaco.

## RESUMEN

El proyecto consiste en almacenamiento masivo de datos de posición GPS en una unidad Flash USB para su posterior utilización con mapas de trayectoria compatibles con Google Earth.

El objetivo principal es registrar datos de posición proporcionados por la tarjeta receptora GPS  $\mu$ -Blox RCB-4H en una unidad USB Flash a través del microcontrolador AT90USB1287 que dispone la funcionalidad USB On-The-Go y trae implementado el conector USB en la tarjeta de desarrollo AT90USBKEY de ATMEL®. Se utiliza el ambiente de desarrollo AVR STUDIO y el programador JTAGICE mkII.

El microcontrolador es el encargado de recibir los datos proporcionados por el receptor GPS mediante el puerto serial al que se accede a través del conector J4. La interacción del usuario con el dispositivo se la realiza presentando mensajes para su uso, a través de una pantalla LCD de 16 caracteres por 2 líneas, controlada a través del conector J5 de la tarjeta de desarrollo utilizada. Y se reciben las órdenes a través de la palanca de mando y otros botones. Se implementa un puerto serial a nivel de 5 voltios así como a nivel de RS232.

## ÍNDICE GENERAL

<b>AGRADECIMIENTO .....</b>	<b>II</b>
<b>DEDICATORIA .....</b>	<b>III</b>
<b>AGRADECIMIENTO .....</b>	<b>IV</b>
<b>DEDICATORIA .....</b>	<b>V</b>
<b>TRIBUNAL DE SUSTENTACIÓN .....</b>	<b>VI</b>
<b>DECLARACIÓN EXPRESA .....</b>	<b>VII</b>
<b>RESUMEN.....</b>	<b>VIII</b>
<b>ÍNDICE GENERAL .....</b>	<b>IX</b>
<b>ABREVIATURAS.....</b>	<b>XII</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>XIII</b>
<b>INTRODUCCIÓN .....</b>	<b>XV</b>
<b>CAPITULO 1</b>	
<b>DESCRIPCIÓN GENERAL DEL PROYECTO .....</b>	<b>1</b>
1.1    Antecedentes. ....	1
1.2    Descripción del Proyecto.....	2
1.3    Aplicaciones. ....	4
1.4    Proyectos Similares. ....	4
1.4.1    G35L Data Logger. ....	5
1.4.2    Holux M-1000C Bluetooth GPS Data Logger.....	6
1.4.3    Wintec WPL-2000 Auto-Show.....	7
<b>CAPÍTULO 2</b>	
<b>FUNDAMENTOS TEÓRICOS. ....</b>	<b>8</b>
2.1    GPS.....	8
2.1.1    Historia de GPS. ....	8
2.1.2    Funcionamiento del GPS. ....	9
2.2    USB .....	11
2.2.1    Evolución del USB. ....	12
2.2.2    Tipos de conectores de USB.....	13
2.2.3    USB mass storage device class (USB MSDC). ....	14
2.3    Componentes del diseño.....	15
2.3.1    Tarjeta GPS $\mu$ -Blox RCB 4H. ....	15
2.3.2    Tarjeta de Desarrollo AT90USBKEY.....	17
2.3.3    Pantalla LCD.....	19
2.3.4    Fuente de alimentación. ....	20

2.4	Herramientas de Diseño. ....	23
2.4.1	Programador JTAGICE mkII.....	23
2.4.2	Proteus. ....	24
2.5	XML. ....	31
2.6	GPX. ....	32

### **CAPÍTULO 3**

<b>DISEÑO E IMPLEMENTACIÓN DEL PROYECTO. ....</b>	<b>36</b>	
3.1	Diseño Preliminar. ....	36
3.2	Pruebas Preliminares. ....	36
3.2.1	Prueba de conexión de USB flash drive. ....	37
3.2.2	Prueba de escritura de archivo. ....	37
3.3	Descripción del proyecto final.....	37
3.3.1	Diagrama de bloques. ....	37
3.4	Algoritmos de control.....	39
3.4.1	Ciclo Principal. ....	39
3.4.2	Tarea de Selección de modo de funcionamiento. ....	40
3.4.3	Tarea de manejo de Anfitrión de Almacenamiento Masivo. ....	42
3.4.4	Tarea de manejo de la memoria del dispositivo unidad USB Flash. ....	44
3.4.5	Inicialización de tarea de registro de datos. ....	45
3.4.6	Tarea de Registro de datos.....	45
3.4.7	Manejo de la interrupción del puerto serial. ....	50
3.5	Programa principal del microcontrolador.....	51
3.5.1	Función datalogging_task_init.....	54
3.5.2	Función datalogging_task.....	54
3.5.3	Función datalogging_start.....	55
3.5.4	Función datalogging_stop.....	57
3.5.5	Función de Manejo de la Interrupcion del Puerto Serial. ....	58
3.5.6	Función ProcesaSentenciaNMEA.....	59
3.5.7	Funciones complementarias.....	64
3.6	Modelo 3D.....	74
3.7	Ensamblado de tarjetas.....	75
3.7.1	Tarjeta Principal.....	75
3.7.2	Fuente de poder.....	75

<b>CAPÍTULO 4</b>	
<b>SIMULACIONES Y PRUEBAS.....</b>	<b>77</b>
4.1 Pruebas de Funcionamiento.....	77
4.1.1 Grabación en formato NMEA.....	77
4.1.2 Grabación en formato GPX .....	78
<b>CONCLUSIONES Y RECOMENDACIONES.</b>	
<b>ANEXOS</b>	
ANEXO I Manual del Usuario del Equipo.	
ANEXO II Lista de Materiales y Presupuesto.	
ANEXO III Diagramas Esquemáticos.	
ANEXO IV Diagramas Circuitos impresos.	
ANEXO V Hoja de datos RCB-4H.	
ANEXO VI Resumen Hoja de datos AT90USB1287.	
ANEXO VII Guía de Usuario AT90USBKEY.	
ANEXO VIII Hoja de datos LM2576.	
ANEXO IX Hoja de datos del módulo TMR-3WI.	
<b>BIBLIOGRAFIA.</b>	

## ABREVIATURAS

<b>CBW:</b>	Contenedor del Bloque de comandos (Command Block Wrapper).
<b>CSW:</b>	Contenedor del Bloque de Estado (Command Status Wrapper)
<b>GPS:</b>	Sistema de Posicionamiento Global (Global Positioning System).
<b>GPX:</b>	Formato de intercambio de datos de GPS(GPS eXchange Format).
<b>LCD:</b>	Pantalla de cristal líquido (Liquid Crystal Display).
<b>NMEA:</b>	Asociación Nacional de Electrónica Marina (National Marine Electronics Association).
<b>MSD:</b>	Dispositivo de almacenamiento masivo ( <b>M</b> ass <b>S</b> torage <b>D</b> evice).
<b>ROHS</b>	Restriccion de sustancias peligrosas ( <b>R</b> estriction <b>O</b> f <b>H</b> azardous <b>S</b> ubstances).
<b>RS232:</b>	Estándar Recomendado 232 ( <b>R</b> ecommended <b>S</b> tandard 232).
<b>TTL:</b>	Logica de transistor a transistor ( <b>T</b> ransistor to <b>T</b> ransistor <b>L</b> ogic).
<b>USB:</b>	Bus Serial Universal( <b>U</b> niversal <b>S</b> erial <b>B</b> us).
<b>USB-IF:</b>	Forum de Implementadores de USB ( <b>U</b> SB <b>I</b> mplementers <b>F</b> orum).
<b>XML:</b>	lenguaje de marcado ampliable (e <b>X</b> tensible <b>M</b> arkup <b>L</b> anguage).

## ÍNDICE DE FIGURAS

<b>Figura 1:</b>	Descripción del Proyecto.....	3
<b>Figura 2:</b>	G35L Data Logger.....	5
<b>Figura 3:</b>	Holux M-1000C Bluetooth GPS Data Logger .....	6
<b>Figura 4:</b>	GPS Wintec WPL-2000 .....	7
<b>Figura 5:</b>	Constelación de Satélites en Órbita.....	10
<b>Figura 6:</b>	Ilustración de Triangulación.....	10
<b>Figura 7:</b>	Interfaz USB .....	12
<b>Figura 8:</b>	Tipos de Conectores USB .....	13
<b>Figura 9:</b>	Receptor GPS $\mu$ -Blox RCB-4H.....	15
<b>Figura 10:</b>	Diagrama de Bloques del Receptor $\mu$ -Blox RCB-4H .....	16
<b>Figura 11:</b>	Recursos de la Tarjeta de Desarrollo AT90USBKEY .....	17
<b>Figura 12:</b>	Microcontrolador AT90USB1287.....	18
<b>Figura 13:</b>	Pantalla LCD 16X2 .....	19
<b>Figura 14:</b>	Modulo ConvertidorTMR 3WI .....	20
<b>Figura 15:</b>	Regulador de Voltaje LM 2576.....	22
<b>Figura 16:</b>	Esquemático para la versión ajustable del regulador LM 2576 .....	22
<b>Figura 17:</b>	JTAGICE mkII .....	23
<b>Figura 18:</b>	Ambiente de ISIS .....	24
<b>Figura 19:</b>	ISIS, Seleccionando Componentes.....	25
<b>Figura 20:</b>	Búsqueda de elementos en Librería.....	25
<b>Figura 21:</b>	Componentes en Fondo ISIS .....	26

<b>Figura 22:</b>	Diagrama Esquemático del Proyecto.....	27
<b>Figura 23:</b>	Llamando la Aplicación ARES de PROTEUS.....	28
<b>Figura 24:</b>	Ubicación de los componentes .....	28
<b>Figura 25:</b>	Iniciando el AUTO ROUTER para el Circuito Impreso .....	29
<b>Figura 26:</b>	Configuración de Auto Router .....	29
<b>Figura 27:</b>	Resultado del Auto Router del Circuito Impreso .....	30
<b>Figura 28:</b>	Formato del Archivo GPX.....	33
<b>Figura 29:</b>	Puntos de Interés, Ruta y Pistas.....	34
<b>Figura 30:</b>	Diagrama de Bloques Diseño .....	38
<b>Figura 31:</b>	Diagrama de flujo Ciclo de Operación del microcontrolador .....	40
<b>Figura 32:</b>	Diagrama de flujo Selección de modo de funcionamiento.....	41
<b>Figura 33:</b>	Diagrama de flujo de conexión de la Unidad USB Flash.....	43
<b>Figura 34:</b>	Diagrama de flujo Manejo de la memoria USB Flash. ....	44
<b>Figura 35:</b>	Diagrama de flujo Configuración de memoria utilizada .....	45
<b>Figura 36:</b>	Diagrama de flujo Registro de Datos (a).....	47
<b>Figura 37:</b>	Diagrama de flujo Registro de datos (b).....	48
<b>Figura 38:</b>	Algoritmo de manejo de la interrupción del puerto serial .....	50
<b>Figura 39:</b>	Vista Frontal del Proyecto .....	74
<b>Figura 40:</b>	Vista Posterior del Proyecto .....	74
<b>Figura 41:</b>	Fotografía Frontal del Circuito.....	76
<b>Figura 42:</b>	Fotografía Posterior del Circuito .....	76
<b>Figura 43:</b>	Datos GPS importados en Google Earth®.....	79

## INTRODUCCIÓN

El objetivo de este proyecto es diseñar e implementar un dispositivo que reciba datos de posición proporcionados por una tarjeta GPS y almacenarlos en un USB Flash Drive para su posterior utilización con mapas de trayectoria compatibles con Google Earth. Se utiliza la tarjeta receptor GPS de  $\mu$ -Blox RCB-4H. El encargado de almacenar estos datos en el USB Flash Drive es el microcontrolador AT90USB1287 gracias a que trae incluido en sus funciones el manejo USB Anfitrión necesarios para poder almacenar los datos proporcionados por el receptor GPS. Estos dos dispositivos es decir el microcontrolador y el puerto USB con función de anfitrión son dos recursos muy importantes que vienen incorporados en la tarjeta de desarrollo AT90USBKEY.

En el primer capítulo, se menciona una descripción general del proyecto, las partes que lo constituyen, su funcionamiento, ciertas aplicaciones más comunes y proyectos similares como: G35L Data Logger, el Holux M-1000C Bluetooth GPS Data Logger y el Wintec WPL-2000 Auto-Show.

En el segundo capítulo muestra en detalle las características de las partes que se han utilizado para la implementación del proyecto, es decir: el Receptor GPS de  $\mu$ -Blox RCB-4H, tarjeta de desarrollo AT90USBKEY con el microcontrolador AT90USB1287, pantalla LCD 16X2 así como el programador JTAGICE mKII.

También el software utilizado como: AVR Studio y las aplicaciones de Proteus ISIS y ARES. Para la realización del Circuito Esquemático y el Circuito Impreso para la placa.

El tercer capítulo contiene información sobre el diseño e implementación del circuito, diagrama de bloques del sistema, los algoritmos del programa del microcontrolador.

Finalmente, en el capítulo cuarto se muestran las pruebas al equipo ya ensamblado. Se muestran datos de archivos generados por el equipo en sus dos formatos de grabación, texto NMEA y GPX.

# CAPITULO 1

## DESCRIPCIÓN GENERAL DEL PROYECTO

### ***1.1 Antecedentes.***

Existe la necesidad de conocer el recorrido realizado por móviles como por ejemplo transporte de mercaderías así como la ruta que siguió un agente de ventas o cualquier persona que desee registrar su trayectoria durante sus viajes de turismo. Actualmente ya existen una infinidad de aplicaciones del GPS, en tal virtud en las diferentes Universidades del país y del mundo se han desarrollado temas relacionados a este sistema, cabe destacar que la ESCUELA SUPERIOR POLITECNICA DEL LITORAL en los últimos años ha venido implementando diferentes proyectos relacionados al GPS, destacando el uso de tarjetas de recepción de GPS, como es la SmartGPS que en un Proyecto es utilizada para recibir datos de posición con microcontrolador el cual grafica la trayectoria del receptor en una pantalla de GLCD. También se han desarrollado proyectos en los que se utiliza módulos para almacenamiento masivo de información, como es el caso de la tarjeta DATALOGGER de Parallax que es la encargada de almacenar en una unidad USB Flash.

Específicamente se destacan las tesis:

“Aplicación de Módulo Receptor de GPS para el posicionamiento de robots manejados a control remoto con capacidad de comunicación serial a Datalogger e Interfaz Gráfica”

“Datalogger compacto, con tiempo real, para almacenaje de grandes cantidades de información provenientes de sensores externos en una memoria USB, con capacidad de comunicación serial. Fuente de energía: 4 pilas recargables AA”

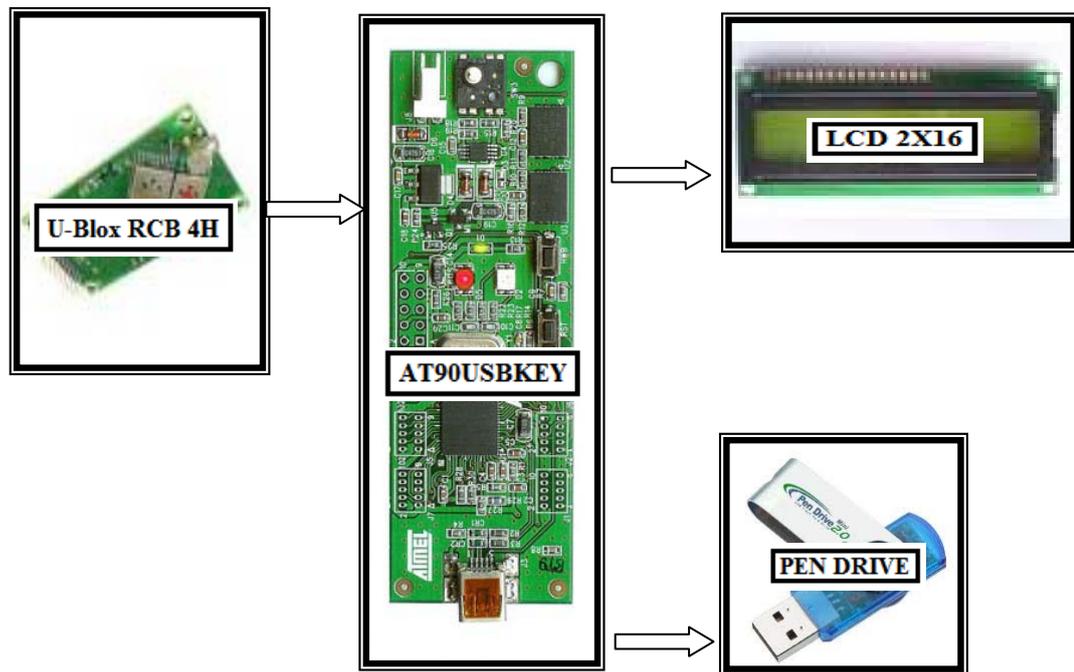
## ***1.2 Descripción del Proyecto.***

El Proyecto consiste en la adquisición de datos de GPS y su registro en archivo en una unidad Flash Drive. Para lograr ese cometido, se utiliza un módulo de GPS que nos sirve para recibir la información de posición, luego esta información es almacenada por un microcontrolador en la unidad de almacenamiento masivo USB.

En la realización de este proyecto utilizaremos la tarjeta  $\mu$ -Blox RCB 4H para la adquisición y envío continuo de datos de posicionamiento GPS, estos datos serán recibidos por la tarjeta de desarrollo AT90USBKEY de Atmel para su procesamiento. La tarjeta AT90USBKEY como parte fundamental está diseñada con un micro controlador AT90USB1287 el cual será el encargado de procesar los datos y almacenarlos en un Flash Drive. Los datos a almacenar en el Flash Drive serán en formato sentencias “NMEA” o en formato XML GPX. La selección del formato en el que se grabaran los datos será realizada por el micro controlador AT90USB1287. El

proyecto contará con una pantalla LCD 16 x 2 que permitirá al usuario interactuar con el dispositivo.

A continuación se muestra un diagrama de bloques de la descripción del proyecto.



**Figura 1:** Descripción del Proyecto

### **1.3 Aplicaciones.**

Este dispositivo podrá ser utilizado en sistemas de supervisión de recorrido, ya que podrá almacenar los datos obtenidos de la ruta que realice el vehículo o la persona que lleve el dispositivo, y saber si este se desvió o no de su ruta establecida.

Otra aplicación de este proyecto es el almacenamiento de datos del recorrido de viajeros que al regresar a sus hogares quisieran mostrar a sus familiares y amigos los lugares del mundo que visitó durante su viaje.

De manera general este proyecto podrá ser aplicado en cualquier medio de movilización del que se necesite obtener información del recorrido que realice, ya que esta información será almacenada en un USB Flash Drive y podrá ser visualizada y/o analizada con la ayuda de un computador que tenga instalado GOOGLE EARTH® ya que este programa es el que nos facilitará la visualización de los lugares por donde se realizó el recorrido.

### **1.4 Proyectos Similares.**

En esta sección presentaremos distintos proyectos relacionados al almacenamiento de datos GPS. Presentaremos ejemplos de GPS que fueron diseñados ya hace un tiempo atrás y GPS que se utilizan en la actualidad.

### 1.4.1 G35L Data Logger.

Este sistema utilizado especialmente en vehículos graba los datos de posicionamiento, velocidad, dirección, fecha, hora, en una memoria interna, para luego ser procesados una vez que el vehículo retorna a su base de trabajo. Esto permite un control detallado de actividades y recorridos.



a) Vista en perspectiva



b) Vista Frontal



c) Vista Posterior

**Figura 2:** G35L Data Logger

### Características Principales

Receptor GPS SiRF III

3 Leds para estados de alimentación, GPS y memoria

Alimentación de entrada entre 7 - 60V DC

Conector Mini USB para descarga de datos y configuración.

Salidas PS2 con nivel RS232/ DC power (adecuada para BlueTooth)

### 1.4.2 Holux M-1000C Bluetooth GPS Data Logger.

Este sistema está diseñado para las personas que quieren usar el GPS en combinación con una PDA, un teléfono móvil o un ordenador portátil, ya que por un lado dispone de la capacidad para almacenar el recorrido y por otro permite enviar la posición actual vía Bluetooth a una PDA, un teléfono móvil o un ordenador.

El Holux M-1000C Bluetooth GPS Logger viene de fábrica con el software exTour, el mismo que acompaña al Holux GPSport 245 y que sirve para analizar las rutas en el ordenador y exportarlas a Google Earth



**Figura 3:** Holux M-1000C Bluetooth GPS Data Logger

#### **Características principales**

Bluetooth™ : V1.2 compliant, Class2

USB : USB 2.0 High speed (mini USB Connector)

Rutas almacenadas compatibles con Google Earth

Indicador LED 3 colores:

Rojo: Navegación GPS

Azul: Bluetooth

Verde: Cargando Batería.

### 1.4.3 Wintec WPL-2000 Auto-Show.

El Wintec WPL-2000 Auto-Show es un GPS Datalogger que no solo nos permite registrar nuestro recorrido en actividades de ocio o deporte, sino que también es ideal para pequeñas aplicaciones de control de flotas, de empresas de seguridad, detectives y otras de seguimiento y control.



**Figura 4:** GPS Wintec WPL-2000

La información registrada en el GPS **Wintec WPL-2000** puede ser descargada y visualizada en nuestro ordenador de un modo sencillo y rápido, gracias a su conector USB y sin necesidad de drivers o controladores.

#### **Características principales.**

Fuente de alimentación: 2 baterías Alcalinas AAA - 1300 mA

Corriente promedio en operación completa: 70 mA

Tiempo de funcionamiento: más de 19 hours

Dimensiones 85\*31\*23 mm

Peso sin baterías: 39 g

## **CAPÍTULO 2**

### **FUNDAMENTOS TEÓRICOS.**

En este capítulo se muestra la mayor cantidad de información de los componentes utilizados para la realización del proyecto, empezando con su parte más esencial el GPS.

#### **2.1 GPS.**

##### **2.1.1 Historia de GPS.**

El GPS o NAVSTAR-GPS es un GNSS (Global Navigation Satellite System o sistema global de navegación por satélite) que permite determinar en todo el mundo la posición de un receptor de GPS.

El GPS, fue creado en la década de 1960 por el Departamento de Defensa de los Estados Unidos con fines militares, para conocer la posición geográfica de sus tropas (aviones, buques y submarinos) y aunque en aquella época sólo podían saber su ubicación en intervalos de tiempo de una a dos horas y con una exactitud con error de 250m, estos sistemas eran muy útiles para determinar la posición geográfica de

aviones, buques y submarinos, por obvias razones de tamaño. Con el paso del tiempo y gracias al avance tecnológico se fue modernizando el GPS, disminuyendo el intervalo de tiempo y aumentando la exactitud de posición geográfica, es así que para el año 1986 las fuerzas militares de los EE.UU. contaron con la posibilidad de determinar su posición geográfica de forma continua y con exactitud de alrededor de un metro.

Sin embargo, solo en el año 1983 a raíz del derribo de un avión de pasajeros coreano (vuelo 007) por parte de la Unión Soviética “URSS”, el presidente de los EE.UU. de esa época Ronald Reagan permite la utilización del GPS para uso civiles, en aviones, embarcaciones y medios de transporte de todo el mundo con el fin de evitar desviarse involuntariamente e invadir límites extranjeros. A partir de esto es que en la actualidad el GPS es utilizado en muchas aplicaciones como: la búsqueda de rutas para conductores, la creación de mapas, el rastreo de vehículos, teléfonos celulares entre otras.

### **2.1.2 Funcionamiento del GPS.**

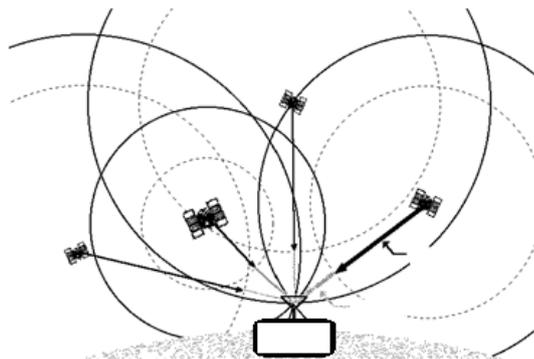
GPS viene de sus siglas en ingles de *Global Positioning System* (Sistema de posicionamiento global) y sirve para determinar la posición, la velocidad de desplazamiento de un vehículo y la hora precisa del día de algún lugar de nuestro interés en cualquier parte del mundo. Se trata de una constelación de 27 satélites impulsados por energía solar que orbitan la Tierra en órbitas casi circulares a una altitud de aproximadamente 20.000 kilómetros, estas orbitas están dispuestas de tal

forma que cualquier receptor GPS en cualquier lugar de la Tierra y en cualquier momento pueda captar la señal de por lo menos 4 satélites. 3 de los 27 satélites actualmente en órbita están de respaldo y listos para reemplazar cualquier satélite que llegara a fallar.



**Figura 5:** Constelación de Satélites en Órbita

Aunque solo son necesarias las señales de tres satélites para calcular la posición en la Tierra, según la distancia entre los tres satélites, la señal del cuarto satélite se la utiliza para confirmar los resultados del cálculo inicial.



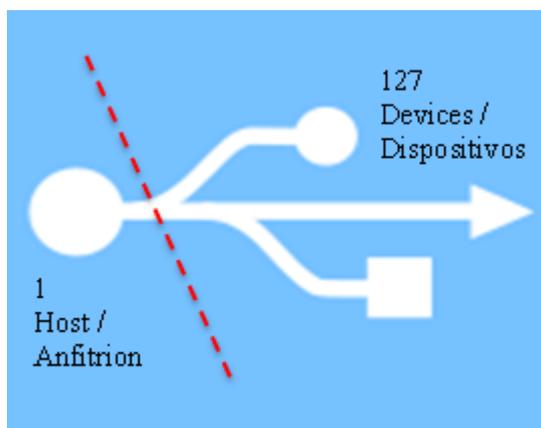
**Figura 6:** Ilustración de Triangulación

El proceso de medición de la distancia del satélite al receptor GPS se denomina triangulación y consiste en señales cronometradas, por ejemplo a una hora exacta determinada los satélites empiezan a emitir su señal. El receptor GPS empieza a realizar el proceso de recepción de señales de los satélites a la misma hora local, cuando el receptor capte la señal de los distintos satélites existirá un tiempo de retardo entre las señales de cada satélite en llegar al receptor, determinando así la distancia de cada satélite y estos retrasos entre señales de cada receptor son utilizadas para calcular la posición del receptor.

## **2.2 USB**

**Bus Universal Serie**, más comúnmente conocido por sus siglas en inglés como **USB** ("Universal Serial Bus"). Es una interfaz orientada principalmente a la integración de periféricos con la computadora. Es un bus serie bidireccional y de bajo costo, diseñado como una extensión en la arquitectura estándar del PC, fue creado por 7 compañías que actualmente forman el Consejo Directivo, las mismas que se nombran a continuación: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

Esta interfaz está definida como 1 Anfitrión y hasta 127 Dispositivos, tal como se muestra en la Figura 8, la misma que ha logrado desplazar a los puertos paralelos, Serie, PS2.



**Figura 7:** Interfaz USB

Existe una nueva especificación denominada USB On-The-Go que permite que un aparato funcione como Anfitrión o como Dispositivo dependiendo de dónde se conecte el pin de configuración del conector USB. 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.

### 2.2.1 Evolución del USB.

El estándar USB ha evolucionado con el transcurrir de los años, acomodándose a las necesidades y a los avances tecnológicos:

USB 0.9: Primer borrador, publicado en Noviembre de 1995.

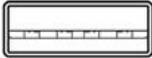
USB 1.0: Publicada en 1996 establece dos tipos de conexión: La primera, denominada **velocidad baja** ("Low speed"), ofrece 1.5 Mbps, y está pensada para periféricos que no requieren un gran ancho de banda, como ratones o joysticks. La

segunda, denominada **velocidad completa** ("Full speed"), es de 12 Mbps, y está destinada a los dispositivos más rápidos.

USB 1.1: Publicada en 1998, añade detalles y precisiones a la norma inicial; es el estándar mínimo que debe cumplir un dispositivo USB.

USB 2.0: Su versión final fue publicada en Abril del 2000; es una extensión de la norma compatible con las anteriores. Permite velocidades de hasta 480 Mbps, denominada **alta velocidad** ("High speed").

### 2.2.2 Tipos de conectores de USB.

Tipo	Imagen de puerto	Imagen de conector
Tipo A	4.5mm x 12.0mm 	
Tipo B	7.3mm x 8.5mm 	
Mini A	3.0mm x 6.8mm 	
Mini B	3.0mm x 6.8mm 	

**Figura 8:** Tipos de Conectores USB

### 2.2.3 USB mass storage device class (USB MSDC).

*Clase de dispositivo de almacenamiento masivo USB*, es un conjunto de protocolos de comunicación definido por la USB Implementers Forum que funciona sobre USB.

El **USB Implementers Forum** (Forum de Implementadores de USB, USB-IF), es una organización sin propósito de lucro con el objetivo de promover y dar soporte a la tecnología USB. Sus principales actividades son la promoción y márketing de USB, Wireless USB y USB On-The-Go, y el mantenimiento de sus especificaciones y el programa de certificación.

El estándar proporciona una interfaz para una variedad de dispositivos de almacenamiento tales como:

- Discos duros magnéticos externos.
- Dispositivos ópticos externos, incluyendo lectores y grabadoras de DVD.
- Memorias flash portátiles, particularmente memorias USB.
- Cámaras digitales.
- Reproductores multimedia.
- Lectores de tarjetas.

Esta interfaz reutilizó el protocolo de comunicación SCSI (Small Computer System Interface) que es el protocolo de control y comunicación de discos duros, para crear la nueva interfaz física. La comunicación se la realiza a través de Paquetes de Comandos donde se inicia la comunicación con los **CBW**, dentro de este paquete de comandos se encuentran los comando del Protocolo SCSI, luego de esto se realiza transferencia de datos y se finaliza la comunicación con el paquete **CSW**.

## **2.3 Componentes del diseño.**

### **2.3.1 Tarjeta GPS $\mu$ -Blox RCB 4H.**

Es una tarjeta receptora GPS de ultra bajo consumo de energía, diseñada con la tecnología de ANTARIS®4 y SuperSense®Indoor GPS (sensores para interiores). Esta tarjeta está equipada con un conector MCX / OSX RF y un conector de 20 pines para alimentación y E/S digitales.

El GPS ANTARIS ® 4 ofrece un alto rendimiento de navegación incluso en interiores y en otros entornos de señal débil.



**Figura 9:** Receptor GPS  $\mu$ -Blox RCB-4H

#### **2.3.1.1 Aspectos destacados.**

SuperSense®Indoor GPS (Sensores para interiores).

Ultra bajo consumo de energía.

A-GPS y el funcionamiento autónomo GPS.

Soporta AssistNow® Online.

Compatible AssistNow® Offline.

Flash EPROM: Programable y Configurable.

2 Puertos UART.

### 2.3.1.2 Características.

16 canales ANTARIS®4 con máquina de posicionamiento.

Compatible con DGPS, WAAS, EGNOS y MSAS (Sistemas de aumentación).

Modos de ahorro de energía.

Compatible con antenas pasivas y activas.

Antena con detección y protección contra cortocircuito y circuito abierto.

Amplio rango de tensión de alimentación: 3,15 a 5,25 V.

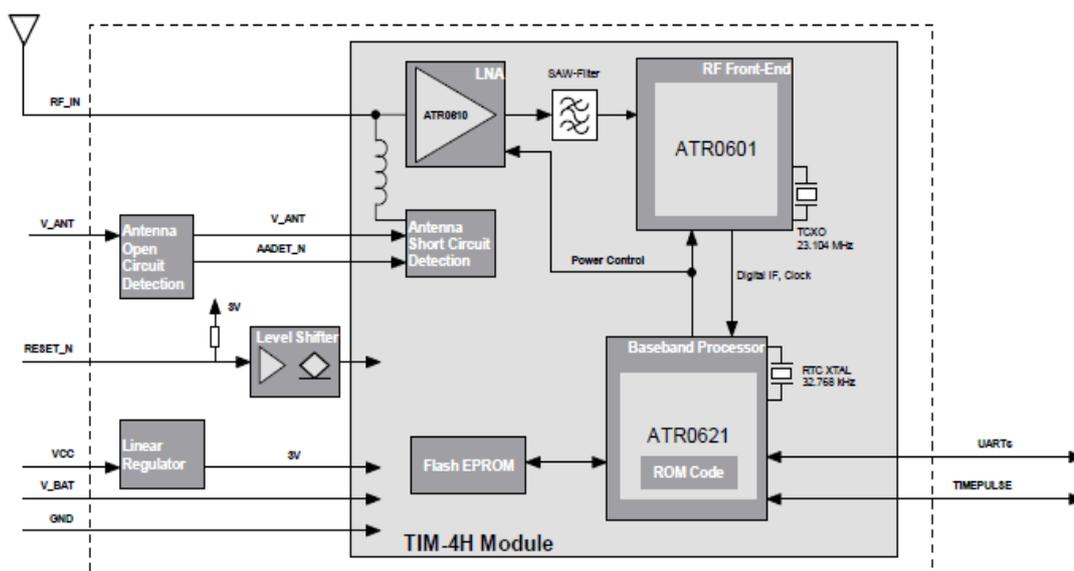
Pin de Voltaje de alimentación por batería para respaldo de la memoria interna y reloj en tiempo real.

Conector estándar para la integración de esta tarjeta con circuito impreso.

Temperatura de funcionamiento: -40 a 85 ° C.

Cumple con RoHS (sin plomo).

Conector de antena en ángulo recto MCX / conector OSX.

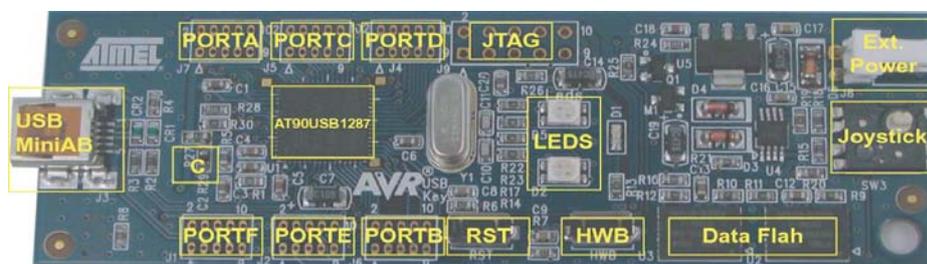


**Figura 10:** Diagrama de Bloques del Receptor  $\mu$ -Blox RCB-4H

### 2.3.2 Tarjeta de Desarrollo AT90USBKEY.

Esta tarjeta fue creada para dar a los diseñadores una guía de inicio rápido para el desarrollo de código AVR® y la creación de prototipos y pruebas de nuevos diseños con la familia de microcontroladores AT90USB.

Para obtener un mayor campo de utilización, esta tarjeta cuenta con: puerto mini USB, mando de control (joystick), memoria data-flash y un sensor de temperatura.



**Figura 11:** Recursos de la Tarjeta de Desarrollo AT90USBKEY

#### 2.3.2.1 Componentes.

01 Microcontrolador AT90USB1287

02 Memorias AT45DB642D

01 Palanca de mando 4 direcciones +1Seleccionador

02 LED Bi-color

01 Sensor de temperatura

01 pulsador RESET

01 pulsador HWB para forzar la ejecución del gestor de arranque (bootloader)

Acceso a todos los puertos de Entrada/Salida por medio de conectores de 2x5 pin

Sistema de reloj con cristal de 8 MHz.

### 2.3.2.2 Principales Características.

Funcionalidad USB On-the-Go

Programación a través de JTAG ICE

Entrada de Voltaje de poder 8 -15 VDC 100mA mínimo

Alimentación desde la interfaz USB cuando está en modo dispositivo.

Almacenamiento de datos en memoria Flash 16MB.

Manejo de sensor de temperatura con coeficiente negativo.

### 2.3.2.3 Características del AT90USB1287.

Microcontrolador de la marca ATMEL el cual fue escogido por tener como recurso el manejo de USB On-the-Go y viene incorporado en la tarjeta de desarrollo AT90USBKEY.



**Figura 12:** Microcontrolador AT90USB1287

### 2.3.3 Pantalla LCD.

Se hace necesaria la interacción del usuario con el proyecto, y para esto existen muchas maneras de hacerlo ya sea por medio de LED's que indican alguna alarma o etapa de funcionamiento del proyecto o por medio de pantallas de 7 segmentos que indican alguna cantidad que sea necesario saber por parte del usuario. En este proyecto se hace necesaria la visualización de mensajes que faciliten e indiquen sobre el funcionamiento del dispositivo, para esto utilizaremos una pantalla LCD de 16 caracteres por 2 líneas.



**Figura 13:** Pantalla LCD 16X2

Para su funcionamiento la pantalla requiere un Voltaje de polarización en los pines 2 y 1, además de conectar el pin 3 a través de una resistencia a Vcc para controlar el Brillo de la pantalla. Por otro lado también podemos controlar el encendido del LED entre los pines 15(BLA) y 16(BLK) con lo cual controlamos la luz de fondo de la pantalla y así obtener una mejor visualización de los caracteres mostrados e través de la pantalla.

Los valores de resistencias utilizadas y la conexión de las mismas en la pantalla se las muestra en el diagrama esquemático del proyecto.

### 2.3.4 Fuente de alimentación.

En el presente proyecto se vió la necesidad de utilizar una fuente de voltajes múltiples, debido que la tarjeta receptora GPS necesita ser energizada con 3.3 VDC, la pantalla LCD 16X2 con 5 VDC, mientras que la tarjeta AT90USBKEY se la puede energizar directamente desde la batería de 9V, ya que esta tarjeta trae internamente un circuito regulador de voltaje.

Para obtener 3.3 VDC necesarios para energizar la tarjeta receptora GPS se utilizó el módulo Convertidor DC-DC TMR 3WI, y el regulador LM2576 para obtener 5VDC necesarios para energizar la pantalla LCD 16X2 y el circuito de Comunicación RS232.

#### 2.3.4.1 Módulo Convertidor TMR 3WI.



**Figura 14:** Modulo ConvertidorTMR 3WI

La serie de TMR-3WI es una nueva familia de Módulos Convertidores de 3W DC/DC con salida regulada, con amplio rango de voltaje de entrada, viene en un encapsulado compacto que ocupa sólo 2,0 cm<sup>2</sup> de espacio en la placa en la que se la vaya a instalar, muestra excelente rendimiento en un rango de temperaturas de

operación muy amplio, incorpora un pin de control de encendido / apagado, protección continua del cortocircuito. Las dimensiones muy compactas de estos convertidores lo convierten en una solución ideal para aplicaciones de espacio reducido, en los equipos a baterías e instrumentación.

### **Características**

Relación Voltaje de entrada vs Voltaje de salida 4:1

Dimensiones: 21.8 x 9.2 mm

Rango de temperatura de operación  $-40^{\circ}$  a  $+85^{\circ}\text{C}$

Eficiencia 81%

Protección contra corcho circuito

Pin de Control Encendido/Apagado

Corriente de Salida a V salida = 3.3V : 700mA

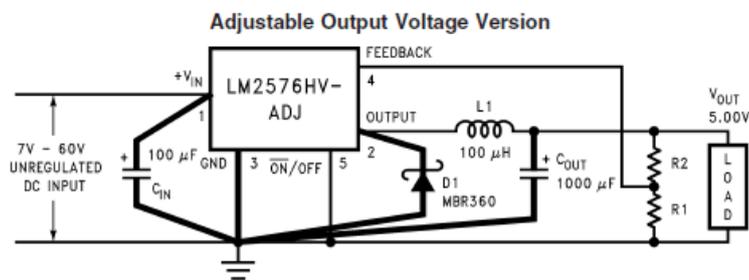
### **2.3.4.2 Regulador LM 2576.**

Regulador de voltaje de amplio rango de alimentación y al mismo tiempo un amplio rango de regulación en la salida capaz de proporcionar 3 A. La serie LM2576 ofrece un reemplazo de alta eficiencia para los populares reguladores lineales de tres terminales ya que reduce sustancialmente el tamaño del disipador de calor, y en algunos casos no es necesario el uso de disipadores de calor. Estos reguladores están disponibles en voltajes de salida fija de 3,3V 5V 12V y 15V y una versión ajustable.



**Figura 15:** Regulador de Voltaje LM 2576

A continuación se muestra el diagrama esquemático para el regulador ajustable y el modelo matemático utilizado para obtener el voltaje adecuado en la salida.



**Figura 16:** Esquemático para la versión ajustable del regulador LM 2576

$$V_{out} = V_{ref} \left(1 + \frac{R_2}{R_1}\right) \quad \text{Ecuación 1}$$

$$R_2 = R_1 \left(\frac{V_{out}}{V_{ref}} - 1\right) \quad \text{Ecuación 2}$$

Donde  $V_{ref} = 1.23V$ .

En este caso utilizamos  $R_1 = 22.1K\Omega$  y  $R_2 = 68.K\Omega$  para obtener en la salida 5V

## **2.4 Herramientas de Diseño.**

### **2.4.1 Programador JTAGICE mkII.**

Es un equipo fabricado por ATMEL para programar y depurar aplicaciones para microcontroladores AVR. Se conecta al PC por puerto serial o USB y desde AVR Studio se puede realizar la grabación y depuración. Este equipo ha sido utilizado en la fase de diseño para programar la tarjeta AT90USBKEY.



**Figura 17: JTAGICE mkII**

#### **Características**

Herramienta de desarrollo Rango Medio

AVR 8bits y AVR 32bits

Depurador a nivel de Código

#### **Modos de Programación**

SPI

JTAG

PDI

aWire

## Modos de Depuración

JTAG

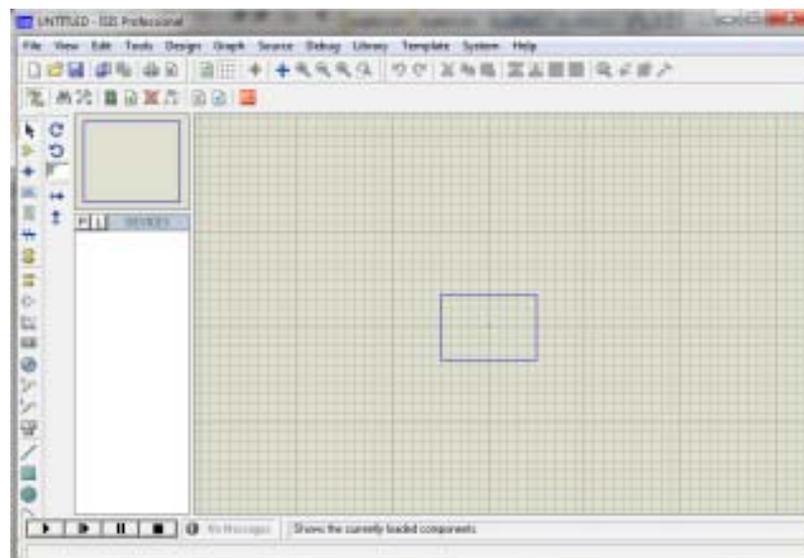
PDI

aWire

### 2.4.2 Proteus.

Este programa es de gran utilidad en el diseño electrónico con microcontroladores. Cuenta con aplicaciones como ISIS que simula el funcionamiento de circuitos que utilizan microcontroladores, puesto que permite cargar el programa del microcontrolador y observar su operación mediante la simulación; y ARES que asiste en el diseño del circuito impreso para la fabricación de la placa.

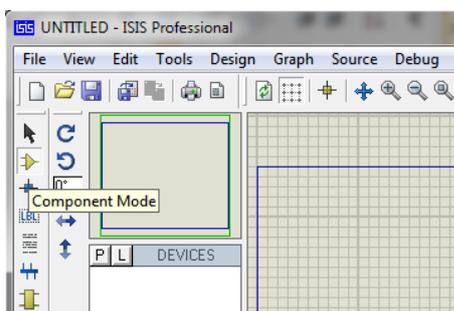
#### 2.4.2.1 ISIS.



**Figura 18:** Ambiente de ISIS

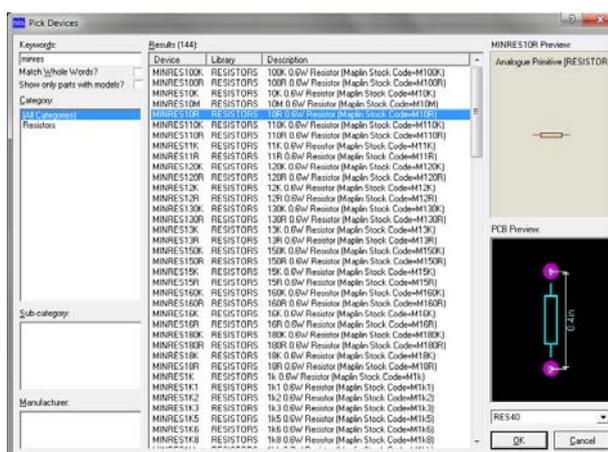
Con la ayuda de esta aplicación del programa Proteus se dibuja el circuito esquemático. A continuación se muestra una de las herramientas más importantes de esta aplicación.

Para escoger los componentes con los que trabajará el circuito esquemático seleccionamos el icono **Component Mode**.



**Figura 19:** ISIS, Seleccionando Componentes

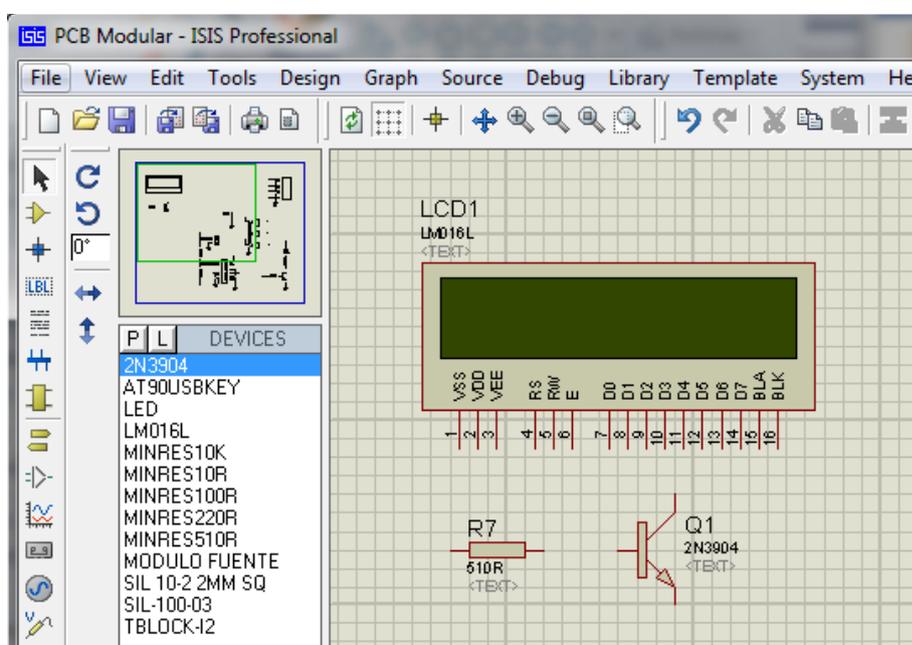
Una vez seleccionado este icono aparecerá debajo del recuadro de pantalla las letras P y L. Al hacer un click en la letra P se abrirá una ventana donde se puede seleccionar los componentes a utilizar.



**Figura 20:** Búsqueda de elementos en Librería

En este ejemplo se escoge una resistencia de 10 Ohm para lo cual deberá tiparse parte del nombre del componente, en este caso se digita “minres”, en la parte superior izquierda. Aparecerán cierta cantidad de resistencias que se pueden utilizar. Para el ejemplo mostrado se ha seleccionado MINRES10R.

En la parte superior derecha de la pantalla aparece el símbolo que utilizara ISIS para la elaboración del circuito esquemático y en la parte inferior derecha se muestra la huella del encapsulado del componente que utilizará Ares para la elaboración del circuito impreso, esto se mostrará más adelante. Una vez seleccionado el componente, al hacer click en OK se sale de esta pantalla y el componente aparece en la pantalla principal de ISIS, listo para insertarlo en el Diagrama Esquemático. De esta manera se escoge los componentes que se necesitan para hacer el diagrama y esta aparecerán en columna para su utilización como lo muestra la fig.



**Figura 21:** Componentes en Fondo ISIS

Luego se puede ir interconectando los componentes según el diseño que se desee implementar.

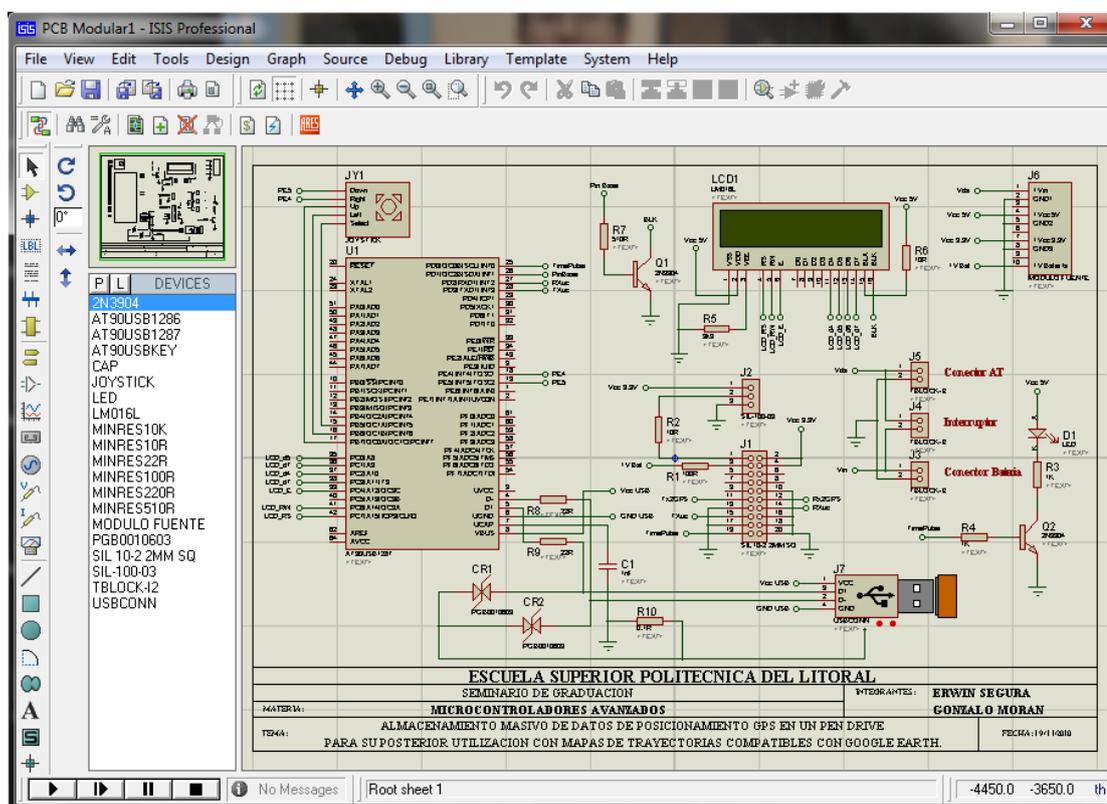
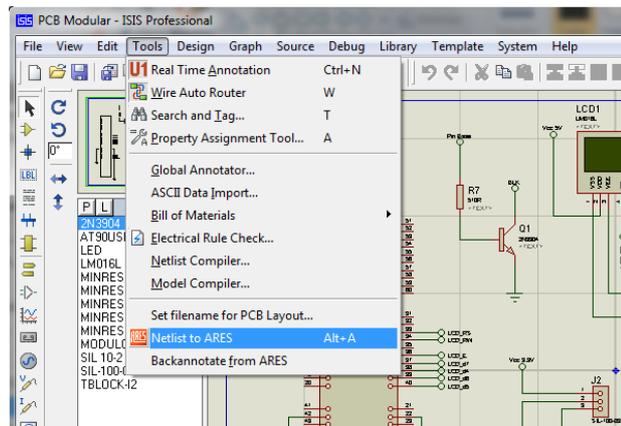


Figura 22: Diagrama Esquemático del Proyecto

En este proyecto no se puede realizar la simulación del circuito ya que no está disponible el microcontrolador AT90USB1287 de ATMEL.

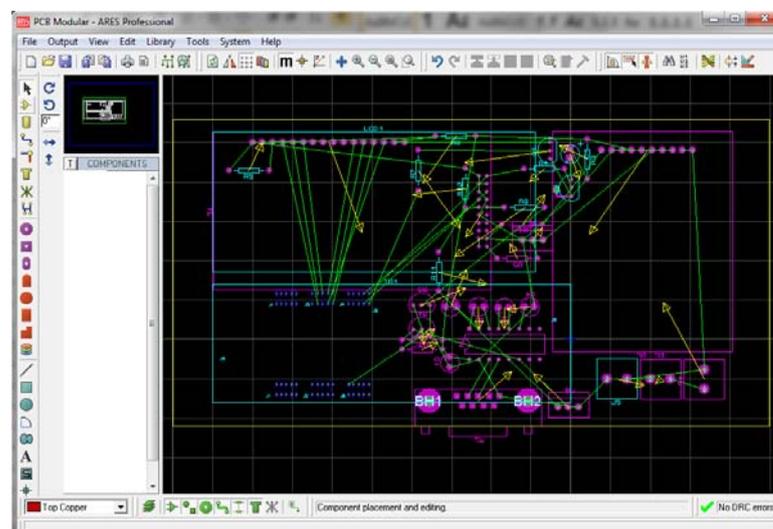
Para la realización del circuito impreso se hace click en el icono Tools y se despliega el menú donde se selecciona *Netlist to ARES*.



**Figura 23:** Llamando la Aplicación ARES de PROTEUS

#### 2.4.2.2 ARES.

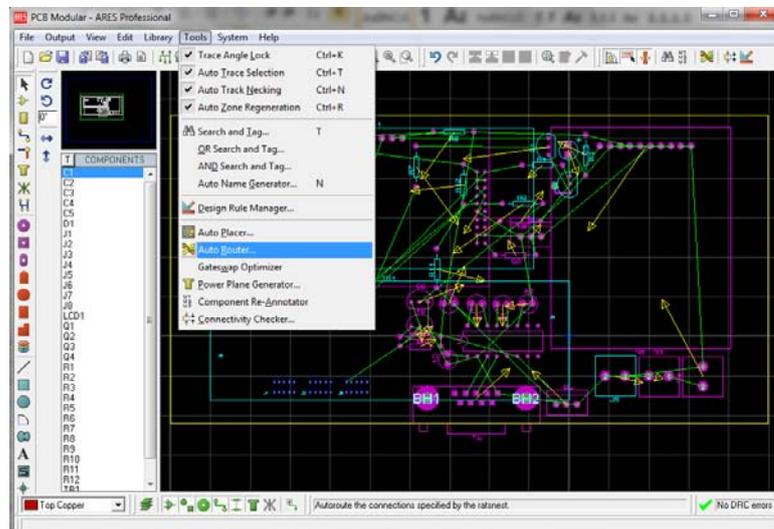
Esta aplicación de Proteus utiliza el archivo Netlist donde esta la informacion de los componentes, como su huella, y las conexiones entre ellos para realizar el diseño del circuito impreso



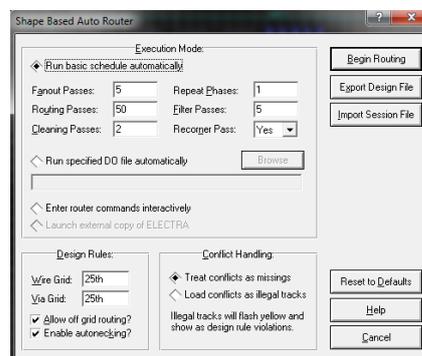
**Figura 24:** Ubicación de los componentes

Al cargar un archivo de netlist, se mostrará los componentes del diseño y se debe elegir cada componente para agregarlo al area de trabajo, como puede notarse hay líneas que unen a los componentes entre si, estas son las redes de conexión.

Para realizar el Ruteo de las pistas del circuito impreso se elige el menú Tools y se selecciona el ítem *Auto Router*, que mostrará la ventana de configuración del Ruteador Automatico.

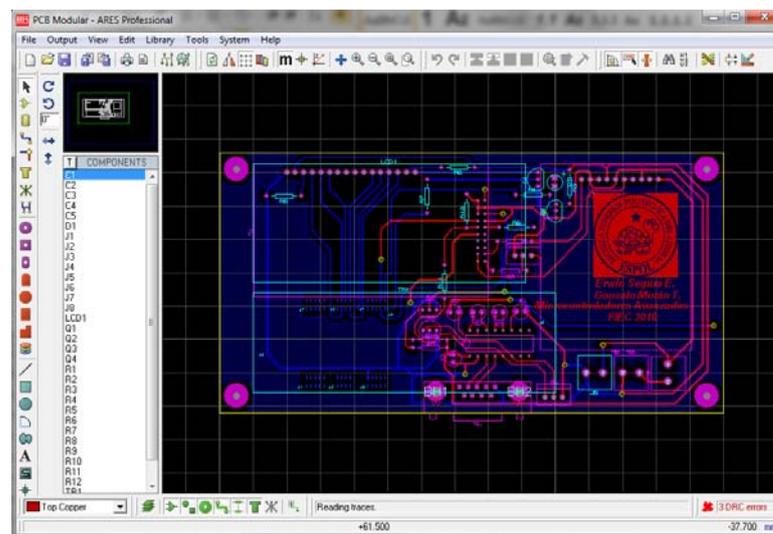


**Figura 25:** Iniciando el AUTO ROUTER para el Circuito Impreso



**Figura 26:** Configuración de Auto Router

Básicamente, si se hace click en *Begin Routing* el programa iniciara el proceso de conectar los pines de cada componente. Según el diseño se deben cambiar parámetros para obtener un circuito impreso que cumpla los requerimientos de diseño



**Figura 27:** Resultado del Auto Router del Circuito Impreso

Otra herramienta de esta aplicación es la visualización en 3D, con la cual se puede simular el acabado final del diseño, mostrándose los componentes en el circuito impreso. Para obtener esta visualización se elige El menú *Output* y se escoje *3D Visualization*.

## **2.5 XML.**

Este meta-lenguaje se utilizará para poder convertir los datos envía el receptor GPS en formato NMEA, a formato GPX para su vinculación directa con Google Earth.

XML no es un lenguaje de marcado como el lenguaje HTML, es un meta-lenguaje que nos permite definir lenguajes de marcado adecuado a usos determinados. Los elementos que lo componen pueden dar información sobre lo que contienen, no necesariamente sobre su estructura física o presentación, como ocurre en HTML, **XML no ha nacido sólo para su aplicación en Internet, sino que se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación), para intercambio de información estructurada entre diferentes plataformas.**

Un documento XML tiene dos estructuras, una lógica y otra física. Físicamente, el documento está compuesto por unidades llamadas entidades, una entidad puede hacer referencia a otra entidad, causando que ésta se incluya en el documento, cada documento comienza con una entidad documento, también llamada raíz. Lógicamente, el documento está compuesto de: Declaraciones, Elementos, Comentarios, referencias a caracteres e instrucciones de procesamiento

Todos los componentes están indicados por una marca explícita, éstas son las partes del documento que el procesador XML espera entender. El resto del documento que se encuentra entre las marcas son los datos que resultan entendibles por las personas. Los elementos XML pueden tener contenido (más elementos, caracteres, o ambos a la vez), o bien ser elementos vacíos.

Un elemento con contenido es, por ejemplo:

```
<nombre>Carlos Aveiga</nombre>  
<aviso tipo="emergencia" gravedad="mortal">No pasa nada, tranquilidad</aviso>
```

Siempre empieza con una `<etiqueta>` que puede contener atributos o no, y termina con una `</etiqueta>` que debe tener el mismo nombre. Al contrario que HTML, en XML siempre se debe "cerrar" un elemento.

El símbolo "<" siempre se interpreta como inicio de una etiqueta XML/

## **2.6 GPX.**

GPX, o GPS eXchange Format es un esquema XML diseñado como un formato común de datos GPS para aplicaciones de software. Puede ser utilizado para describir Puntos de interés, pistas y rutas. El formato es abierto y puede ser utilizado sin la necesidad de pagar derechos de licencia.

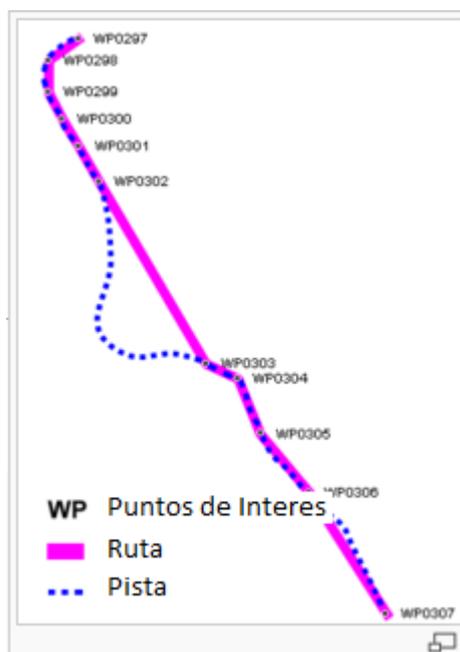
Las Etiquetas guardan: la ubicación, la elevación y el tiempo, y pueden ser utilizados para el intercambio de datos entre los dispositivos GPS y paquetes de software. Estos programas informáticos permiten a los usuarios, por ejemplo, ver sus recorridos, planificar sus rutas en imágenes de satélite (en Google Earth), Apuntar en Mapas, y etiquetar fotografías con la localización geográfica en la etiqueta Exif.

GPS eXchange Format	
Nombre del archivo de extensión	.gpx
los medios de comunicación de Internet tipo	application/gpx+xml
Versión inicial	2002
Última versión	1.1 <sup>[1]</sup> / 09 de agosto 2004
Extendido de	XML
Sitio web	<a href="http://www.topografix.com/gpx.asp">www.topografix.com / gpx.asp</a> 

**Figura 28:** Formato del Archivo GPX

En GPX, una colección de puntos, sin relación secuencial (las ciudades del condado de Inglaterra, por ejemplo, o todos los rascacielos de Nueva York), se considera una colección de puntos de control individuales. Una colección ordenada de puntos se puede expresar como una pista o una ruta. Conceptualmente, las pistas son un registro de cuando una persona ha ido, las rutas son sugerencias acerca de dónde podrían ir en el futuro. Por ejemplo, cada punto en una pista puede tener una marca de tiempo (porque alguien estuvo grabando donde y cuando ellos estuvieron allí), pero los puntos en una ruta es improbable que tengan marcas de tiempo, porque el autor está sugiriendo una ruta que nadie podría haber viajado.

Las características mínimas de un archivo GPX son la latitud y la longitud de un único punto de control. Todas las demás variables son opcionales.



**Figura 29:** Puntos de Interés, Ruta y Pistas

El siguiente es un ejemplo truncado de un archivo GPX producida por un [Garmin Oregon 400t](#) de la unidad de ayuda GPS. Este documento no muestra todas funciones que se pueden almacenar en el formato GPX, por ejemplo no hay puntos de referencia o ampliaciones y esto es parte de una pista, no una ruta. Pero el propósito es mostrar un ejemplo rápido.

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "no"?>
<gpx xmlns = "http://www.topografix.com/GPX/1/1"
xmlns:gpwx = "http://www.garmin.com/xmlschemas/GpxExtensions/v3" xmlns:gpwtpx =
"http://www.garmin.com/xmlschemas/TrackPointExtension/v1" creator = "Oregon
400t" version = "1.1"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd
http://www.garmin.com/xmlschemas/GpxExtensions/v3
http://www.garmin.com/xmlschemas/GpxExtensionsv3.xsd
http://www.garmin.com/xmlschemas/TrackPointExtension/v1
http://www.garmin.com/xmlschemas/TrackPointExtensionv1.xsd" >
  <metadata >
    <link href = "http://www.garmin.com" >
      <text > Garmin International </text >
    </link>
```

```
<time > 2009-10-17T22:58:43Z </time >
</metadata >
<trk >
  <name > Example GPX Document </name >
  <trkseg >
    <trkpt lat = "47.644548" lon = "-122.326897" >
      <ele > 4.46 </ele >
      <time > 2009-10-17T18:37:26Z </time >
    </trkpt >
    <trkpt lat = "47.644548" lon = "-122.326897" >
      <ele > 4.94 </ele >
      <time > 2009-10-17T18:37:31Z </time >
    </trkpt >
    <trkpt lat = "47.644548" lon = "-122.326897" >
      <ele > 6.87 </ele >
      <time > 2009-10-17T18:37:34Z </time >
    </trkpt >
  </trkseg >
</trk >
</gpx >
```

## **CAPÍTULO 3**

### **DISEÑO E IMPLEMENTACIÓN DEL PROYECTO.**

#### ***3.1 Diseño Preliminar.***

Para el diseño de este proyecto se seleccionó un microcontrolador con función de anfitrión USB y el AT90USB1287 de ATMEL posee funciones de USB On-the-go.

Se utiliza la tarjeta de recepción de GPS  $\mu$ -Blox RCB-4H.

Para interfaz gráfica se dispone de un módulo de pantalla de 16 caracteres por 2 líneas.

Se diseñó un módulo de fuente de voltajes para la circuitería, teniendo en cuenta que los voltajes de operación son de 3.3V para el receptor de GPS, 5V para la pantalla y 9 voltios para la tarjeta AT90USBKEY.

#### ***3.2 Pruebas Preliminares.***

Para confirmar la acertada selección del microcontrolador se procedió a evaluar las notas de aplicación que ofrece el fabricante para demostrar las funcionalidades del dispositivo, a continuación se detalla cada prueba:

### **3.2.1 Prueba de conexión de USB flash drive.**

Se utilizó la nota de aplicación AVR287, que como demostración realiza la copia de archivos desde la unidad USB flash hasta la memoria interna de la tarjeta y viceversa.

*AVR287 demostración de anfitrión USB para dispositivo de interfaz humana y almacenamiento masivo.*

### **3.2.2 Prueba de escritura de archivo.**

En la nota de aplicación AVR115 se detalla el proceso de registro en archivo de datos pero usando otro controlador del mismo fabricante.

*AVR115 registros de datos con sistema de archivos ATMEL sobre ATmega32U4*

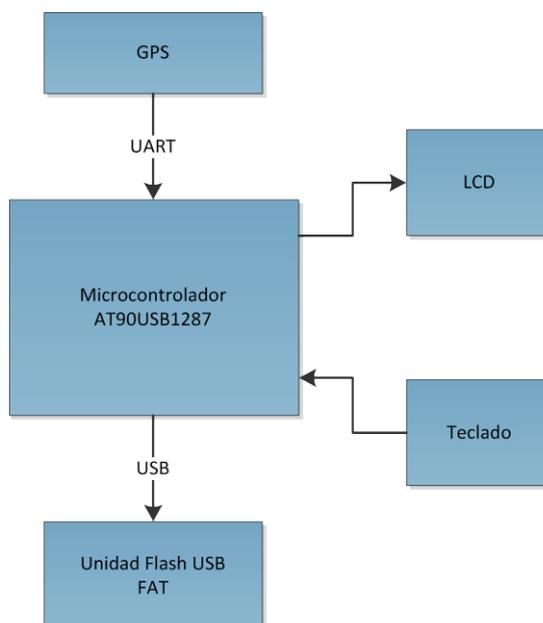
Se adaptó el código fuente de la nota de aplicación para realizar la misma función en el AT90USB1287

Estas pruebas permitieron comprobar las funcionalidades de mayor prioridad, como son el poder conectar un dispositivo de almacenamiento masivo; el de interactuar con su sistema de archivos; así como también, abrir un archivo y registrar información en éste, dentro de la unidad USB flash.

## **3.3 Descripción del proyecto final.**

### **3.3.1 Diagrama de bloques.**

En la siguiente grafica se detalla la interconexión de circuitos para el funcionamiento del proyecto:



**Figura 30:** Diagrama de Bloques Diseño

Como se aprecia en el diagrama de bloques, las partes principales de este proyecto son:

Microcontrolador Atmel® AT90USB1287

Receptor de GPS  $\mu$ -Blox RCB-4H

LCD de caracteres 16 x 2

El receptor de GPS provee los datos de posición por medio de un puerto serial al microcontrolador y este los procesa para realizar el archivado en la unidad USB Flash.

El voltaje de operación del receptor de GPS como de la tarjeta del controlador AT90USBKEY es de 3.3V, por lo cual la interconexión de señales no presenta inconvenientes, es así que el puerto serial que los une se conecta directamente.

Los datos a mostrarse en la pantalla de caracteres son enviados desde el microcontrolador, en este caso hay una diferencia en el voltaje de operación de estos componentes, puesto que el LCD trabaja a 5.0V, y como se ha indicado previamente, el microcontrolador funciona con 3.3V. Como las señales van desde el elemento que trabaja a menor voltaje hacia el otro, no hay inconvenientes de operación puesto que el voltaje mínimo en alto de las entradas del LCD es 2.4V, de esta forma un voltaje alto en las salidas del microcontrolador, es detectado como un voltaje alto por las entradas del LCD.

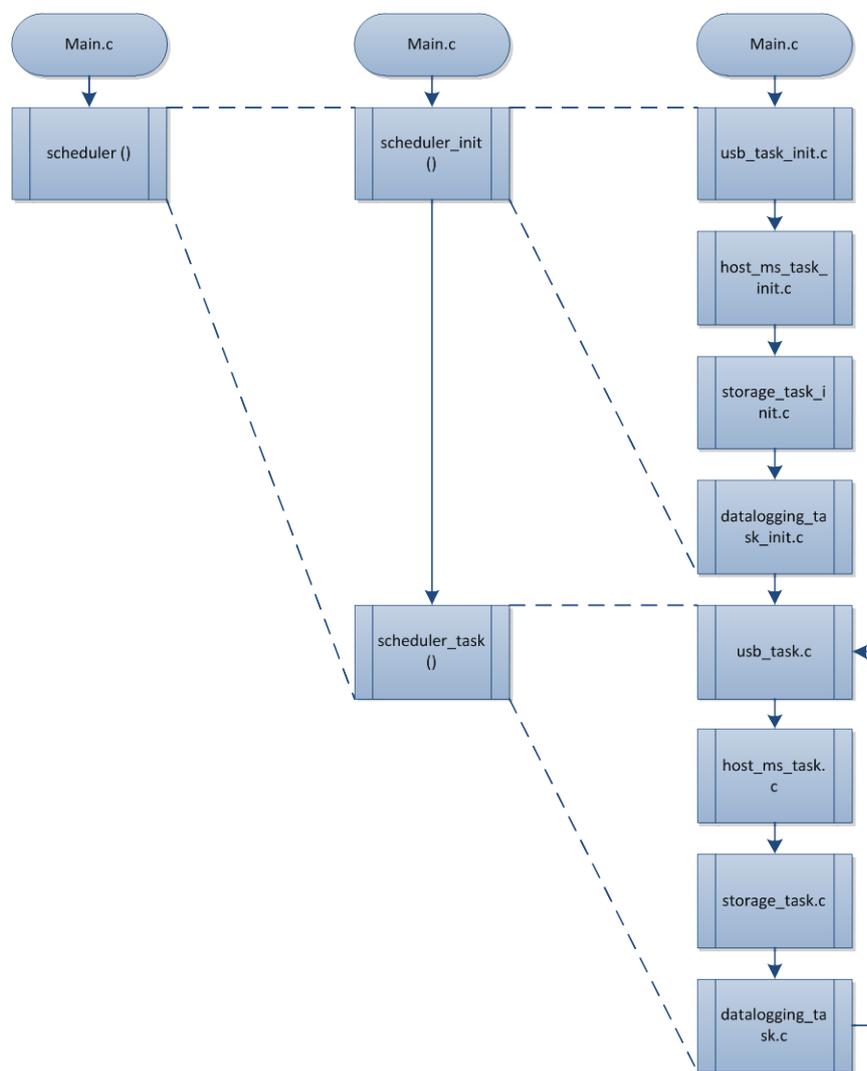
### ***3.4 Algoritmos de control.***

A continuación se hace referencia al bosquejo de la programación del microcontrolador en sus diferentes tareas para la ejecución del proyecto, como se detalla a continuación:

#### **3.4.1 Ciclo Principal.**

El microcontrolador utiliza un Ciclo principal de tareas. Primero se ejecuta una función de inicialización de cada tarea y luego secuencialmente pasa cada una de las tareas de ejecución normal, de forma que va verificando la conexión o desconexión

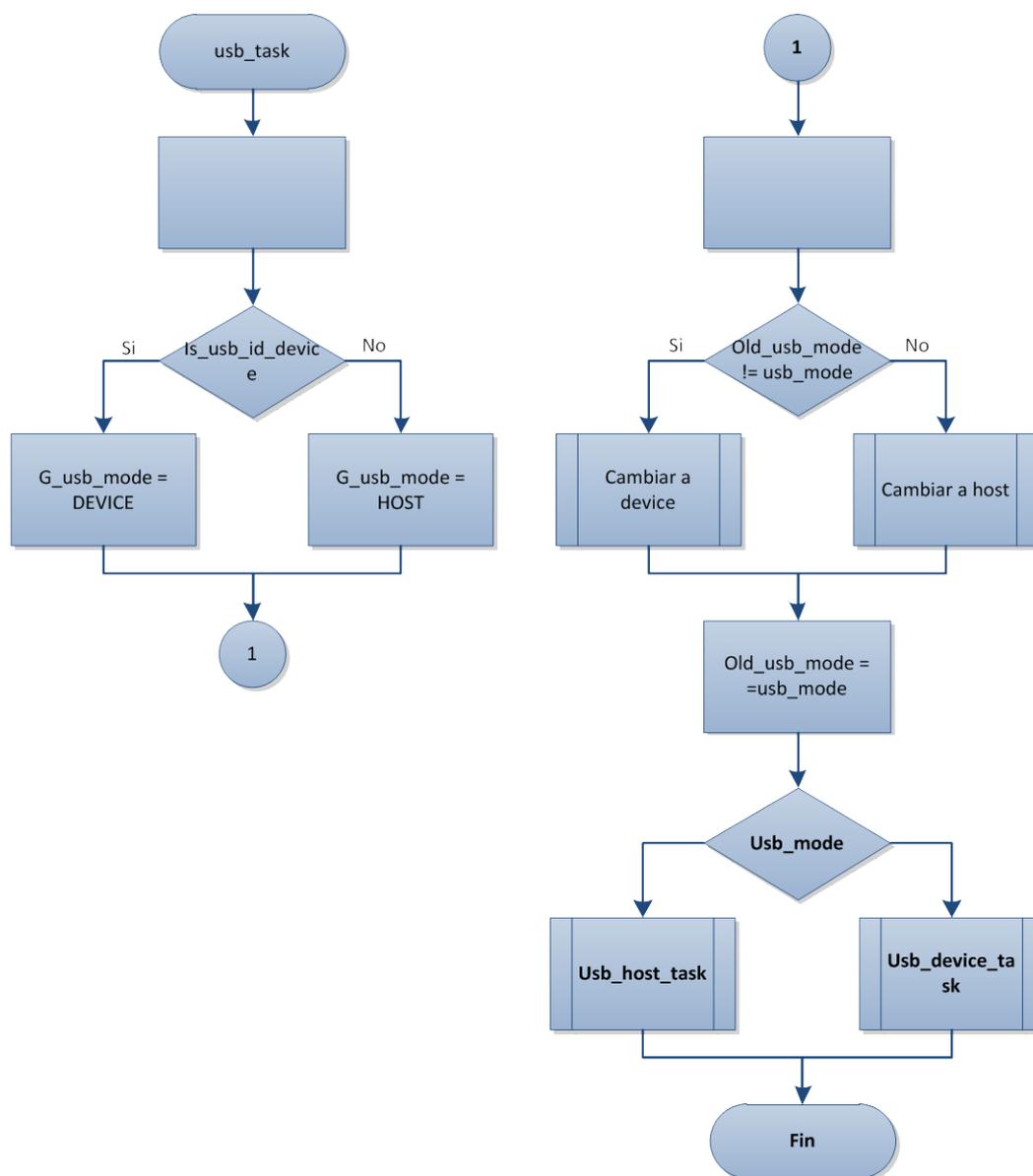
de la unidad USB Flash; así como también la utilización de la memoria de este dispositivo, y cómo manejar el sistema de archivos contenido en la misma.



**Figura 31:** Diagrama de flujo Ciclo de Operación del microcontrolador

### 3.4.2 Tarea de Selección de modo de funcionamiento.

La tarea `usb_task()` verifica el pin de id del puerto USB para determinar si se recibe en el conector USB un anfitrión o un dispositivo, y almacena el estado en la variable `g_usb_mode`.



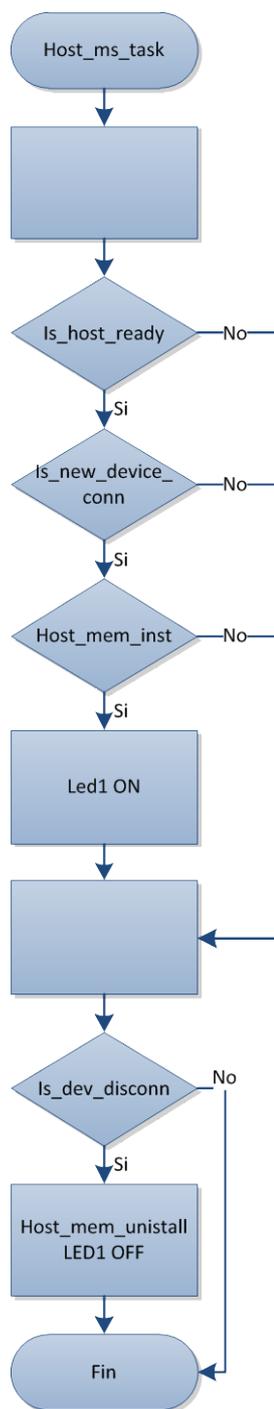
**Figura 32** Diagrama de flujo Selección de modo de funcionamiento

Luego se verifica si ha ocurrido un cambio en el modo de funcionamiento verificando el valor de la variable `g_old_usb_mode`. Si ha existido cambio se realiza la reconfiguración del microcontrolador para funcionar en su nuevo modo, sea este anfitrión o dispositivo. Se prosigue guardando el valor de `g_usb_mode` en

`g_old_usb_mode` para cuando se realice la siguiente inspección. Finalmente según el valor de `g_usb_mode` se ejecuta la tarea de anfitrión o la de dispositivo. En este caso el sistema funcionará como anfitrión. El grafico a continuación detalla lo descrito:

### **3.4.3 Tarea de manejo de Anfitrión de Almacenamiento Masivo.**

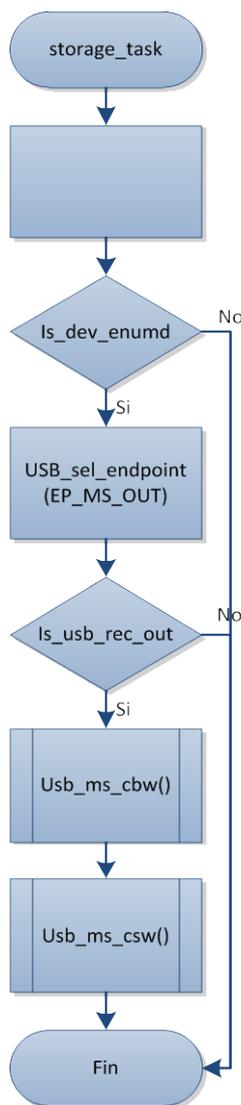
En esta tarea se verifica que el sistema está funcionando como anfitrión, luego si se ha conectado un nuevo dispositivo de almacenamiento USB Flash y si así ha ocurrido se procede a configurar la memoria disponible para usarse. Al terminar de realizar este proceso se enciende el LED1, el cual indica al usuario que la memoria de la unidad USB Flash ha sido reconocida y está lista para utilizarse. Al final se verifica si se ha desconectado algún dispositivo de almacenamiento previamente instalado, de ser así, se realiza la remoción de los datos de la memoria y se apaga el LED1, indicándole así al usuario, que ya no está disponible la memoria.



**Figura 33** Diagrama de flujo de conexión de la Unidad USB Flash

### 3.4.4 Tarea de manejo de la memoria del dispositivo unidad USB Flash.

La tarea `storage_task()` se encarga de la interfaz a la unidad USB Flash, esto es, envía los comandos SCSI para el manejo del dispositivo.



**Figura 34:** Diagrama de flujo Manejo de la memoria USB Flash.

### 3.4.5 Inicialización de tarea de registro de datos.

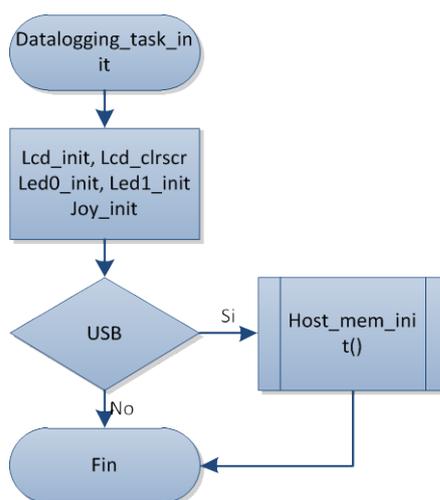
Datalogging\_task\_init()

En esta tarea se inicializa la pantalla LCD, y se muestra información descriptiva del proyecto:

GPSlogUSB v1.0

Segura / Moran

Luego se inicializa la memoria de la unidad USB Flash que ha sido conectada.



**Figura 35:** Diagrama de flujo Configuración de memoria utilizada

### 3.4.6 Tarea de Registro de datos.

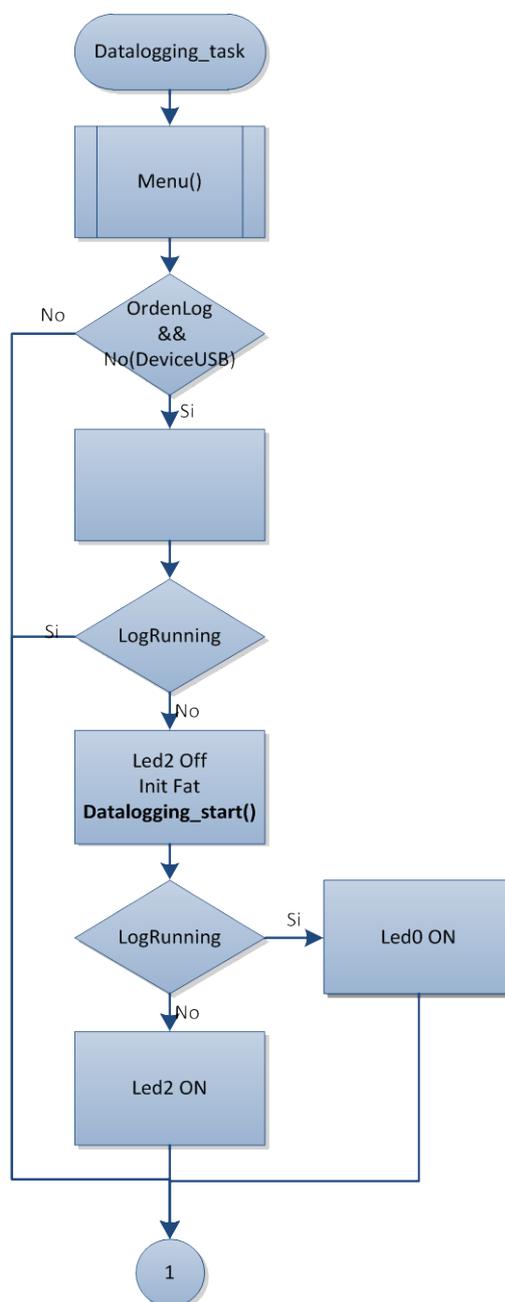
En esta tarea se realiza la recepción de comandos de parte del usuario, y el proceso de almacenamiento de los datos en un buffer intermedio previo a la grabación en la memoria del dispositivo de almacenamiento masivo USB Flash.

El proceso de recepción de comandos se realiza con la ayuda de información mostrada en la pantalla LCD. Cuando el dispositivo se inicializa muestra una pantalla de inicio donde se puede observar información del nombre del proyecto y los apellidos de los integrantes del grupo de desarrollo de éste. Cuando el usuario activa una de las señales de la palanca de mando, se muestra en pantalla dos opciones, que son grabar y configurar. La opción de grabar activa el proceso de grabación y también que se muestre otra pantalla donde se da la opción de parar la grabación. La opción de configurar permite al usuario cambiar el formato de grabación de los datos en el archivo, aquí hay dos opciones que son TXT, donde las tramas de datos que se reciben por el puerto serial son íntegramente guardadas en el archivo y la segunda opción, GPX, donde los datos son grabados con el formato de archivo GPX.

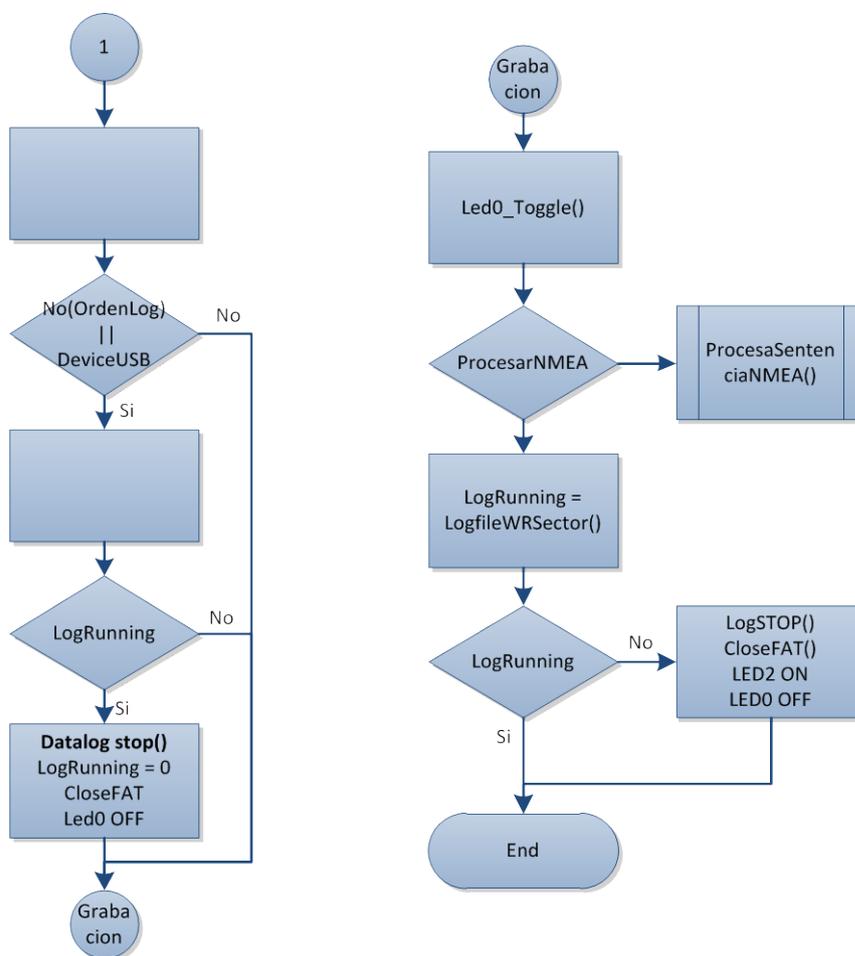
En la tarea `datalogging_task()` se manejan 3 etapas: Inicio de grabación, parar grabación y grabado

En la etapa de inicio de grabación se verifica por la variable `OrdenLog` y también que el sistema no esté en modo dispositivo. Luego de esto se observa si ya se ha estado grabando por medio de la variable `LogRunning`, si no ha estado grabando se ejecutan procesos de inicio de la grabación, que son inicializar el sistema de archivos de la memoria disponible, apagar el LED indicador que muestra que se está grabando, y la creación, si no existe, de la carpeta donde se almacenaran los archivos de registro y de allí la creación y apertura para grabación del archivo donde se van registrar los datos. Al terminar cada rutina de inicialización de uso de la memoria se manda a activar la variable `LogRunning`, que luego es verificada si esta activa para continuar el proceso.

Cuando se revisa el estado de LogRunning se vuelve a encender el LED0. Si no se enciende LED2 que es el indicador de Error en la grabación.



**Figura 36:** Diagrama de flujo Registro de Datos (a)



**Figura 37:** Diagrama de flujo Registro de Datos (b)

Se continua con una etapa de parado de grabación, donde se evalúa si se ha desactivado la variable OrdenLog o si se ha conectado como dispositivo el aparato, si una de estas opciones ocurre, se verifica si aún está activa la variable LogRunning, de estar activa LogRunning, se ejecuta el proceso de parado de la grabación, que es grabar los últimos datos del buffer a la memoria no volátil, de allí cerrar el archivo

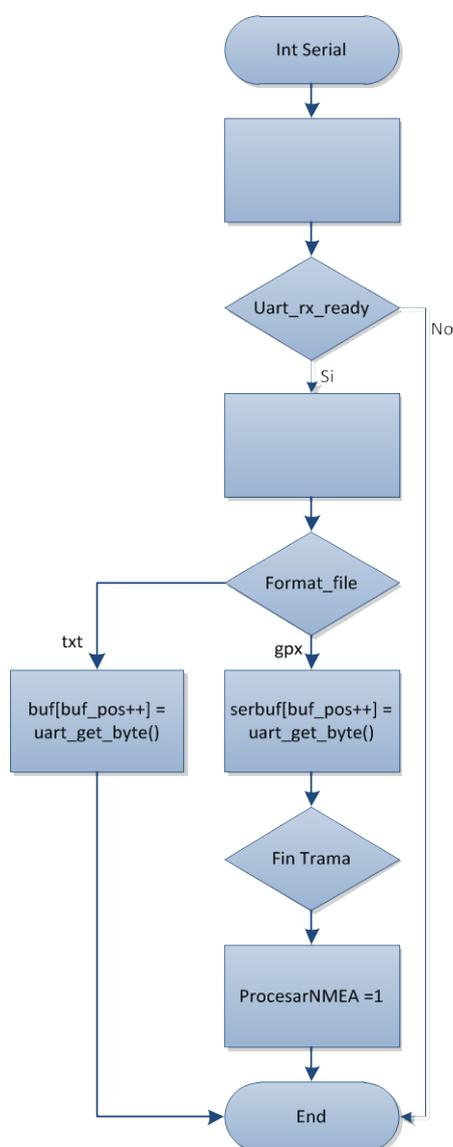
donde se está registrando la información, y por último descargar el sistema de archivos.

Finalmente está la etapa de grabación, aquí se hace un destello del LED0 para indicar al usuario que se está grabando en la memoria. Se pregunta por la variable ProcesarNMEA que es activada por otro proceso que se explica más adelante. Esta variable indica la orden de ejecutar el proceso de la trama que se encuentra en el buffer del puerto serial, esto para ponerlo en el formato de archivo GPX.

Luego se realiza el grabado de datos en la memoria no volátil con una función que devuelve el estado de la grabación, en la variable LogRunning y de aquí si la grabación falla, se ordena la detención de la grabación.

### 3.4.7 Manejo de la interrupción del puerto serial.

Con la activación de la interrupción del puerto serial se ejecuta el código de almacenamiento de cada carácter que llega al puerto serial. Se va verificando cada carácter recibido hasta encontrar el carácter de fin de trama del mensaje NMEA, y al encontrarse este carácter se activa la variable ProcesarNMEA().



**Figura 38:** Algoritmo de manejo de la interrupción del puerto serial

### 3.5 Programa principal del microcontrolador.

La parte principal del código de control se encuentra en el archivo `datalogging_task()`, la cual se detalla a continuación:

`Datalogging_task.c`

```

/*This file is prepared for Doxygen automatic documentation generation.*/
//! \file *****
//!
//! \brief This file manages the mass storage task.
//!
//! - Compiler:          IAR EWAVR and GNU GCC for AVR
//! - Supported devices: ATmega32U4
//!
//! \author             Atmel Corporation: http://www.atmel.com \n
                        Support and FAQ: http://support.atmel.no/
//!
//! *****

/* Copyright (c) 2009 Atmel Corporation. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 *
 * 3. The name of Atmel may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 *
 * 4. This software may only be redistributed and used in connection with an Atmel
 * AVR product.
 *
 * THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED
 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY AND
 * SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

// _____ I N C L U D E S _____

#include "config.h"
#include "conf_usb.h"
#include "conf_access.h"
#include "lib_mcu/timer/timer8_drv.h"
#include "lib_mcu/adc/adc_drv.h"
#include "lib_board/usb_key/usb_key.h"
#include "modules/usb/device_chap9/usb_standard_request.h"

```

```

#include "modules/control_access/ctrl_access.h"
#include "modules/file_system/navigation.h"
#include "modules/file_system/file.h"
#include "modules/lcd/hd44780.h"
#include "lib_mcu/uart/uart_lib.h"
#include "datalogging_task.h"
#include "stdio.h"
#include "stddef.h"
#include "string.h"
#include "stdlib.h"
#include <stdint.h>
#include <math.h>

//! Buffer management
#define NB_DATA_BUF 2
#define DATA_BUF_SIZE (NB_DATA_BUF*FS_SIZE_OF_SECTOR)
#define SERIAL_DATA_BUF_SIZE 200
// File alloc space (unit sector 512B)
#define FILE_ALLOC_SIZE ((1024L*1024L)/512L) // 1MB
#define FILE_ALLOC_SIZE_MIN ((100*1024L)/512L) // 100KB

//_____ D E C L A R A T I O N S _____

typedef enum {AAM=1,
ALM, APA, APB, BOD, BWC, DTM, GGA, GLL, GRS, GSA, GST, GSV,
MSK, MSS, RMA, RMB, RMC, RTE, TRF, STN, VBW, VTG, WCV, WPL,
XTC, XTE, ZTG, ZDA } enumTipoNMEAmsg;

Bool g_b_datalogging_running;
Bool g_b_ProcesarNMEA;
Bool g_b_datoValido = false;
Bool g_b_OrdenGrabar = false;
Bool g_b_GrabarFechaCreacion = true;
enumTipoNMEAmsg enumNMEARec;
U16 g_u16_buf_pos_wr; // (No need to be volatile)
U8 g_u8_cur_buf; // Current buffer waiting (No need to be volatile)
U8 g_u8_cur_buf_wr; // Current buffer waiting (No need to be volatile)
U8 g_u8_cur_buf_ser; // Current buffer waiting (No need to be volatile)
U8 g_u8_cur_buf_ser_proc; // Current buffer waiting (No need to be volatile)

Bool g_b_buf_full[NB_DATA_BUF]; // Signal state of buffers (No need to be volatile)
U16 g_u16_GrabarCada = 0;
U8 g_u8_formatFile = 2;
//1 nmea
//2 gpx

U8 g_u8_screen = 0;
U8 g_u8_opcion = 0;

char linea[20];
char *g_list_txt_GrabarCada[] = {" 1 seg", " 5 seg", "10 seg", "15 seg", "30 seg", " 1 min", " 5
min", "10 min", "15 min", "30 min", "60 min"};
U16 g_list_val_GrabarCada[] = { 1, 5, 10, 15, 30,
60, 300, 600, 900, 1800, 3600};
U16 g_list_val_calc[] = {1000, 5000, 10000, 15000, 30000, 60000, 60000, 60000, 60000, 60000,
60000};
U8 g_list_val_nav[] = { 1, 1, 1, 1, 1,
1, 5, 10, 15, 30, 60};

U8 g_u8_sel_GrabarCada = 0;

U8 _MEM_TYPE_SLOW_ FechaArchivo[] = "20";

```

```

char g_data_buf[DATA_BUF_SIZE];
char g_ser_data_buf[SERIAL_DATA_BUF_SIZE];
char g_ser_proc_buf[SERIAL_DATA_BUF_SIZE];
char *horaGPS;
char *EstadoDisponible;
union {
    int32_t b32;
    int8_t b8[4];
} Latnum;
char Lat[12];
char *LatSigno;
union {
    int32_t u32;
    int8_t b8[4];
} Lonnum;
char Lon[14];
char *LonSigno;
char *velKnot;
char *CoG;
char *FechaGPS;
char *vacio;
char *CRCMsg;

char *running;
char *Altitud;
U8 _MEM_TYPE_SLOW_date_create[20];
char date_write[20];
char g_str_temp[18];
char *g_ptr_temp;
long g_u32_temp;
U8 lenlinea;
U8 temp;
U8 lenlineafijo;

Bool datalogging_start (void);
void datalogging_stop (void);
Bool datalogging_open_disk (void);
Bool datalogging_create_path_file (void);
Bool datalogging_create_file (void);
Bool datalogging_alloc_file_space (void);
Bool datalogging_file_write_sector (void);
Bool datalogging_file_write_all (void);
void datalogging_file_init_wav_header (void);
Bool datalogging_file_update_gpx_header (void);
Bool datalogging_file_update_wav_header (void);
U16 datalogging_get_adc_ext_val (void);
void datalogging_file_init_gpx_header (void);
void datalogging_file_init_gpx_track (void);
void datalogging_file_init_gpx_tracksegment(void);
void datalogging_file_end_gpx_tracksegment(void);
void datalogging_file_end_gpx_track (void);
void datalogging_file_end_gpx (void);
void ProcesaSentenciaNMEA (void);
void Menu (void);
void str2Buf (char* strAdd);
void uart_puts (const char *s );

```

### 3.5.1 Función datalogging\_task\_init.

```

void datalogging_task_init(void)
{
    lcd_init();
    lcd_clrscr();
    lcd_puts(" GPSlogUSB v1.0");
    lcd_goto(0x40);
    lcd_puts(" Segura / Moran");
    g_b_datalogging_running = FALSE;
    Led0_init();
    Led2_init();
    Joy_init();
    Hwb_button_init();
    #if (LUN_USB == ENABLE)    // USB
        host_mem_init();    // Inicializa los recursos de hw/sw. requeridos para manejar el USB MSD
    #endif
    //#if (LUN_2 == ENABLE)    // Data Flash
    //    df_mem_init();    // Init the hw/sw resources required to drive the DF.
    //#endif
    //#if (LUN_3 == ENABLE)    // MMC / SD
    //    Mmc_init_spi();    // Init the hw/sw resources required to drive the MMC/SD.
    //#endif
}

//! @brief Entry point of the datalogging task management
//!

```

### 3.5.2 Función datalogging\_task.

```

void datalogging_task(void)
{
    //Manejo de Menu
    Menu();
    // Change the datalogging state
    if( g_b_OrdenGrabar    // If DataLogging started by user
    && (!Is_device_enumerated())) // and if USB Device mode stopped
    {
        if( !g_b_datalogging_running )
        {
            // Start data logging
            Led2_off();    // Clear flag error
            nav_reset();    /** Init File System
            g_b_datalogging_running = datalogging_start();
            if( g_b_datalogging_running )
            {
                Led0_on();
            }else{
                Led2_on();    // Datalogging startup fail
            }
        }
    }
    if( !g_b_OrdenGrabar    // If DataLogging stopped by user
    || Is_device_enumerated() ) // or if USB Device mode started
    {
        if( g_b_datalogging_running )
        {
            // Stop data logging

            datalogging_stop();
        }
    }
}

```

```

        g_b_datalogging_running = FALSE;
        nav_exit();           /*** Exit of File System module
        Led0_off();
    }
}

// Execute data logging background task
if( g_b_datalogging_running )
{
    Led0_toggle();

    if (g_b_ProcesarNMEA)
    {
        ProcesaSentenciaNMEA();
    }
    g_b_datalogging_running = datalogging_file_write_sector();
    if( !g_b_datalogging_running )
    {
        datalogging_stop();
        nav_exit();           /*** Exit of File System module
        Led2_on();           // Error during write data process
        Led0_off();
    }
}
}
}

```

### 3.5.3 Función datalogging\_start.

```

/** @brief Start the datalogging
/**
/** @return FALSE, in case of error
/**
Bool datalogging_start(void)
{
    // Open disk
    if( !datalogging_open_disk() )
        return FALSE;
    // Create file
    if( !datalogging_create_path_file() )
        return FALSE;
    // Alloc a cluster list to increase the bandwidth during data logging
    datalogging_alloc_file_space();
    // return TRUE; // File created and FAT space allocated
    // if( datalogging_alloc_file_space() )
    // return TRUE; // File created and FAT space allocated

    // Init buffer control
    for( g_u8_cur_buf=0; g_u8_cur_buf<NB_DATA_BUF; g_u8_cur_buf++) {
        g_b_buf_full[g_u8_cur_buf] = FALSE;
    }
    g_u8_cur_buf = 0;
    g_u8_cur_buf_wr = 0;
    g_u16_buf_pos_wr = 0;
    g_u8_cur_buf_ser = 0;

    // Add header data in file
    uart_init();//velocidad en config.h
    Uart_enable_it_rx();
    //Configura el GPS para la frecuencia de envio de tramas
    uart_putchar(0xB5);
    uart_putchar(0x62);
    uart_putchar(0x06);
    uart_putchar(0x08);

```

```

uart_putchar(0x06);
uart_putchar(0x00);
switch (g_u8_sel_GrabarCada)
{
case 0:
    uart_putchar(0xE8); //1000
    uart_putchar(0x03);
    uart_putchar(0x01); //1
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0x00); //CHK
    uart_putchar(0x37);
case 1:
    uart_putchar(0x88); //5000
    uart_putchar(0x13);
    uart_putchar(0x01); //1
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0xB0); //CHK
    uart_putchar(0x47);
case 2:
    uart_putchar(0x10); //10000
    uart_putchar(0x27);
    uart_putchar(0x01); //1
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0x4C); //CHK
    uart_putchar(0xDB);
case 3:
    uart_putchar(0x98); //15000
    uart_putchar(0x3A);
    uart_putchar(0x01); //1
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0xE7); //CHK
    uart_putchar(0x6A);
case 4:
    uart_putchar(0x30); //30000
    uart_putchar(0x75);
    uart_putchar(0x01); //1
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0xBA); //CHK
    uart_putchar(0x21);
case 5:
    uart_putchar(0x60); //60000
    uart_putchar(0xEA);
    uart_putchar(0x01); //1
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0x5F); //CHK
    uart_putchar(0x8A);
case 6:
    uart_putchar(0x60); //60000
    uart_putchar(0xEA);
    uart_putchar(0x05); //5
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);

```

```

    uart_putchar(0x63); //CHK
    uart_putchar(0x9A);
case 7:
    uart_putchar(0x60); //60000
    uart_putchar(0xEA);
    uart_putchar(0x0A); //10
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0x68); //CHK
    uart_putchar(0xAE);
case 8:
    uart_putchar(0x60); //60000
    uart_putchar(0xEA);
    uart_putchar(0x0F); //15
    uart_putchar(0x00);
    uart_putchar(0x00); //UTC
    uart_putchar(0x00);
    uart_putchar(0x6D); //CHK
    uart_putchar(0xC2);
}
switch ( g_u8_formatFile )
{
//case 1: //txt
// break;
case 2: //gpx
    {
        datalogging_file_init_gpx_header();
        datalogging_file_init_gpx_track();
        datalogging_file_init_gpx_tracksegment();
        break;
    }
}
return TRUE;
}

```

### 3.5.4 Función datalogging\_stop.

```

//! @brief Stop the datalogging
//!
void datalogging_stop(void)
{
    // Stop acquisition interrupt
    Timer8_compare_it_disable();

    // Close file
    switch (g_u8_formatFile )
    {
        //case 1: //txt
        // break;
        case 2: //gpx cierra los tag del xml
        {
            datalogging_file_end_gpx_tracksegment();
            datalogging_file_end_gpx_track();
            datalogging_file_end_gpx();
            break;
        }
    }
    g_b_datalogging_running = datalogging_file_write_sector();

    datalogging_file_write_all();

    //Para poner fecha de modificacion al archivo

```

```

date_write[0] = '2';
date_write[1] = '0';
date_write[2] = FechaGPS[4];
date_write[3] = FechaGPS[5];
date_write[4] = FechaGPS[2];
date_write[5] = FechaGPS[3];
date_write[6] = FechaGPS[0];           //dia
date_write[7] = FechaGPS[1];
date_write[8] = horaGPS[0];
date_write[9] = horaGPS[1];
date_write[10] = horaGPS[2];
date_write[11] = horaGPS[3];
date_write[12] = horaGPS[4];
date_write[13] = horaGPS[5];
date_write[14] = '0';
date_write[15] = '0';
date_write[16] = 0;

if( !nav_file_dateset( date_write , FS_DATE_LAST_WRITE) )
    return ;
/*if( !nav_file_dateset( date_create , FS_DATE_CREATION) )
    {}
*/
file_close();
}

```

### 3.5.5 Función de Manejo de la Interrupcion del Puerto Serial.

```

#ifdef __GNUC__
    ISR(USART1_RX_vect)
#else
#pragma vector = USART1_RX_vect
__interrupt void usart_receive_interrupt()
#endif
{
    U8 extern g_u8_cur_buf_ser;

    if(Uart_rx_ready())
    {
        switch (g_u8_formatFile )
        {
            case 1: //txt
            {
                g_data_buf[g_u16_buf_pos_wr] = Uart_get_byte();
                g_u16_buf_pos_wr++;
                if( 0 == (g_u16_buf_pos_wr%FS_SIZE_OF_SECTOR) )
                {
                    g_b_buf_full[g_u8_cur_buf_wr] = TRUE;
                    if( DATA_BUF_SIZE == g_u16_buf_pos_wr )
                    {
                        g_u16_buf_pos_wr = 0;
                        g_u8_cur_buf_wr = 0;
                    }else{
                        g_u8_cur_buf_wr++;
                    }
                }
            }
            break;
            case 2: //gpx
            {
                g_ser_data_buf[g_u8_cur_buf_ser] = Uart_get_byte();
                if (g_ser_data_buf[g_u8_cur_buf_ser] == 10)
                {

```



```

case 71:/'G':
{
  if (g_ser_proc_buf[4] == 'G')
  {
    enumNMEARec = GGA;
    break;
  }else{
    if (g_ser_proc_buf[4] == 'L')
    {
      enumNMEARec = GLL;
      break;
    }else{
      if (g_ser_proc_buf[4] == 'R')
      {
        enumNMEARec = GRS;
        break;
      }else{//el ultimo ya es s directo
        if (g_ser_proc_buf[5] == 'A')
        {
          enumNMEARec = GSA;
          break;
        }else{
          if (g_ser_proc_buf[5] == 'T')
          {
            enumNMEARec = GST;
          }else{
            //if (g_ser_data_buf[5] == 'V')
            enumNMEARec = GSV;
          }
        }
      }
    }
  }
}
}
}
case 77:/'M':
{
  if (g_ser_proc_buf[5] == 'K')
  {
    enumNMEARec = MSK;
  }else{
    enumNMEARec = MSS;
  }
}
break;
}
case 82:/'R':
{
  if (g_ser_proc_buf[5] == 'A')
  {
    enumNMEARec = RMA;
    break;
  }else{
    if (g_ser_proc_buf[5] == 'B')
    {
      enumNMEARec = RMB;
      break;
    }else{
      if (g_ser_proc_buf[5] == 'C')
      {
        enumNMEARec = RMC;
      }else{
        enumNMEARec = RTE;
      }
    }
  }
  break;
}
}

```

```

    }
}
case 83://'S':
{
    enumNMEARec = STN;
    break;
}
case 84://'T':
{
    enumNMEARec = TRF;
    break;
}
case 86://'V':
{
    if (g_ser_proc_buf[4] == 'B')
    {
        enumNMEARec = VBW;
    }else{
        enumNMEARec = VTG;
    }
    break;
}
case 87://'W':
{
    if (g_ser_proc_buf[4] == 'C')
    {
        enumNMEARec = WCV;
    }else{
        enumNMEARec = WPL;
    }
    break;
}
case 88://'X':
{
    if (g_ser_proc_buf[5] == 'C')
    {
        enumNMEARec = XTC;
    }else{
        enumNMEARec = XTE;
    }
    break;
}
case 90://'Z':
{
    if (g_ser_proc_buf[4] == 'T')
    {
        enumNMEARec = ZTG;
    }else{
        enumNMEARec = ZDA;
    }
    break;
}
};
switch (enumNMEARec)
{
case RMC:
{
    vacio = strtok(g_ser_proc_buf, ",");
    horaGPS = strtok(NULL, ",");
    EstadoDisponible = strtok(NULL, ",");
    if(EstadoDisponible[0] == 'V')
    {
        g_b_datoValido = false;
        goto salida;
    }
}
}

```

```

else
{
    g_b_datoValido = true;
}
g_ptr_temp = (char*)strtok(NULL, ",");
strcpy(Lat,g_ptr_temp);
LatSigno = strtok(NULL, ",");

g_ptr_temp = (char*)strtok(NULL, ",");
strcpy(Lon,g_ptr_temp);

LonSigno = strtok(NULL, ",");
velKnot= strtok(NULL, ",");
CoG= strtok(NULL, ",");
FechaGPS = strtok(NULL, ",");
vacio = strtok(NULL, ",");
vacio = strtok(NULL, ",");
vacio = strtok(NULL, ",");
CRCMsg= strtok(NULL, ",");
break;
}
default:
{
    goto salida;
    break;
}
};//end switch
if (g_b_datoValido)
{
    if(g_b_GrabarFechaCreacion)
    {
        date_write[0] = '2';
        date_write[1] = '0';
        date_write[2] = FechaGPS[4];
        date_write[3] = FechaGPS[5];
        date_write[4] = FechaGPS[2];
        date_write[5] = FechaGPS[3];
        date_write[6] = FechaGPS[0]; //dia
        date_write[7] = FechaGPS[1];
        date_write[8] = horaGPS[0];
        date_write[9] = horaGPS[1];
        date_write[10] = horaGPS[2];
        date_write[11] = horaGPS[3];
        date_write[12] = horaGPS[4];
        date_write[13] = horaGPS[5];
        date_write[14] = '0';
        date_write[15] = '0';
        date_write[16] = 0;
        if( !nav_file_dateset( date_write , FS_DATE_CREATION) )
            return;
        g_b_GrabarFechaCreacion = false;
    }
    str2Buf( "<trkpt lat=");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    if (LatSigno[0] == 'S')
        str2Buf("-");

    linea[0] = Lat[0];
    linea[1] = Lat[1];
    linea[2] = '.';
    linea[3] = 0;
    str2Buf(linea);

```

```

Lat[4] = Lat[3];
Lat[3] = Lat[2];
Lat[2] = '0';
Lat[1] = '0';
Lat[0] = '0';
Lat[10] = '0';
Lat[11] = 0;

g_u32_temp = atol(Lat);
g_u32_temp = strtol(Lat,NULL , 10);
g_u32_temp = g_u32_temp / 6;
ltoa(g_u32_temp,linea,10);
str2Buf(linea);

linea[0]= 34;
linea[1]= 0;
str2Buf(linea);
str2Buf(" lon=");
linea[0]= 34;
linea[1]= 0;
str2Buf(linea);

if (LonSigno[0] == 'W')
    str2Buf("-");

linea[0] = Lon[0];
linea[1] = Lon[1];
linea[2] = Lon[2];
linea[3] = '.';
linea[4] = 0;
str2Buf(linea);

Lon[5] = Lon[4];
Lon[4] = Lon[3];
Lon[3] = '0';
Lon[2] = '0';
Lon[1] = '0';
Lon[0] = '0';
Lon[11] = '0';
Lon[12] = 0;

g_u32_temp = atol(Lon);
g_u32_temp = strtol(Lon,NULL , 10);
g_u32_temp = g_u32_temp / 6;
ltoa(g_u32_temp,linea,10);
str2Buf(linea);
linea[0]= 34;
linea[1]= 0;
str2Buf(linea);

str2Buf(">");
linea[0]= 10;
linea[1]= 32;
linea[2]= 32;
linea[3]= 32;
linea[4]= 0;
str2Buf(linea);
str2Buf("<ele>");
str2Buf("0.0"); //elevacion 0.00
str2Buf("</ele>");
linea[0]= 10;
linea[1]= 32;
linea[2]= 32;
linea[3]= 32;
linea[4]= 0;

```

```

    str2Buf(linea);
    str2Buf("<time>20");

    linea[0] = FechaGPS[4];
    linea[1] = FechaGPS[5];
    linea[2] = '-';
    linea[3] = FechaGPS[2];
    linea[4] = FechaGPS[3];
    linea[5] = '-';
    linea[6] = FechaGPS[0];
    linea[7] = FechaGPS[1];
    linea[8] = 'T';
    linea[9] = horaGPS[0];
    linea[10] = horaGPS[1];
    linea[11] = ':';
    linea[12] = horaGPS[2];
    linea[13] = horaGPS[3];
    linea[14] = ':';
    linea[15] = horaGPS[4];
    linea[16] = horaGPS[5];
    linea[17] = 0;

    str2Buf(linea);
    str2Buf("Z</time>" );
    linea[0]= 10;
    linea[1]= 32;
    linea[2]= 32;
    linea[3]= 32;
    linea[4]= 0;
    str2Buf(linea);
    str2Buf("</trkpt>" );
    linea[0]= 10;
    linea[1]= 32;
    linea[2]= 32;
    linea[3]= 32;
    linea[4]= 0;
    str2Buf(linea);
}
salida:
g_b_ProcesarNMEA = 0;
//Uart_enable_it_rx();
}

```

### 3.5.7 Funciones complementarias.

```

/*! @brief Open disk
/*!
/*! @return FALSE, in case of error
/*!
Bool datalogging_open_disk(void)
{
    U8 u8_i;
    nav_select(FS_NAV_ID_DEFAULT);

    #if( (LUN_2 == ENABLE) && (LUN_USB == ENABLE)) // Configuration set in conf_access.h
        // Select and try to mount disk MMC/SD (lun 1) or DataFlash (lun 0)
        // Try first the MMC/SD
        for( u8_i=1; u8_i!=0xFF; u8_i-- )
    #else
        // There is only one memory MMC/SD or DataFlash (lun 0)
        for( u8_i=0; u8_i!=0xFF; u8_i-- )
    #endif
    {

```

```

    if( nav_drive_set(u8_i) )           // Select driver (not disk)
    {
        // Driver available then mount it
        if( !nav_partition_mount() )
        {
            // Error during the mount then check error status
            if( FS_ERR_NO_FORMAT != fs_g_status )
                continue;           // Disk fails (not present, HW error, system
error, ...)
            // Disk no formatted then format it
            if( !nav_drive_format(FS_FORMAT_DEFAULT) )
                continue;           // Format fails
        }
        return TRUE;                 // Here disk mounted
    }
    return FALSE;                     // No valid disk found
}

//! @brief Create path to store datalog
//!
//! @return FALSE, in case of error
//!
Bool datalogging_create_path_file(void)
{
    char ascii_name[15];
    U16 u16_dir_num = 0;

    if( !nav_dir_root() )
        return FALSE;               // Error FS

    while( u16_dir_num < 30 ) // The limitation of number of directories is just an example
    {
        sprintf( ascii_name, "./log%03d/", u16_dir_num);

        // Enter in sub directory and eventually create it if don't exist
        if( !nav_setcwd( ascii_name, FALSE, TRUE) )
            return FALSE;           // Error FS

        // Create a file
        if( datalogging_create_file() )
            return TRUE;            // File created

        // Here, the directory is full then go to parent directory to create the next sub
        // directory
        if( !nav_dir_gotoparent() )
            return FALSE;           // Error FS
        u16_dir_num++;
    }
    return FALSE;                   // Too many log directories and files
}

//! @brief Create file to store datalog
//!
//! @return FALSE, in case of error
//!
Bool datalogging_create_file(void)
{
    char ascii_name[15];
    U16 u16_file_num = 0;

    while( u16_file_num < 10 )
    {

```

```

    // Create file

    switch (g_u8_formatFile )
    {
        case 1: //txt
            sprintf( ascii_name, "log%03d.txt", u16_file_num);
            break;
        case 2: //gpx
            sprintf( ascii_name, "log%03d.gpx", u16_file_num);
            break;
    }
    if( nav_file_create( ascii_name ))
        return TRUE; // Here, the file is created, closed and emptied
    // Error during creation then check error
    if( fs_g_status != FS_ERR_FILE_EXIST )
        return FALSE; // Error FS (Disk full, directory full, ...)
    // The file exist then increment number in name
    u16_file_num++;
}
return FALSE; // Too many log files in current directory
}

//! @brief Alloc in FAT a cluster list file to increase the bandwidth during data logging
//!
//! @return FALSE, in case of error
//!
Bool datalogging_alloc_file_space(void)
{
    Fs_file_segment g_recorder_seg;

    // Open the file created in write mode
    if( !file_open(FOPEN_MODE_W))
        return FALSE;

    // Define the size of segment to alloc (unit 512B)
    // Note: you can alloc more in case of you don't know total size
    g_recorder_seg.u16_size = FILE_ALLOC_SIZE;

    // Alloc in FAT a cluster list equal or inferior at segment size
    if( !file_write( &g_recorder_seg ))
    {
        file_close();
        return FALSE;
    }

    // If you want then you can check the mimimun size allocated
    if( g_recorder_seg.u16_size < FILE_ALLOC_SIZE_MIN )
    {
        file_close();
        nav_file_del(FALSE);
        return FALSE;
    }

    // Close/open file to reset size
    // Note: This sequence doesn't remove the previous FAT allocation
    file_close(); // Closes file
    if( !file_open(FOPEN_MODE_W)) // Opens file in write mode and forces the size
        to 0
        return FALSE;
    return TRUE; /** File open and FAT allocated
}

//! @brief Store buffers avialable in file

```

```

//!
//! @return FALSE, in case of error
//!
Bool datalogging_file_write_sector(void)
{
    // !!!! Note :
    // if the written buffer size has a multiple of 512 B
    // and if the current file position is a multiple of 512 B
    // then the "file_write_buf()" routine is very efficient.
    if( g_b_buf_full[g_u8_cur_buf] )
    {
        if( !file_write_buf( &g_data_buf[g_u8_cur_buf*FS_SIZE_OF_SECTOR], FS_SIZE_OF_SECTOR ) )
            return FALSE; // Error write
        g_b_buf_full[g_u8_cur_buf] = FALSE;
        // Now wait new buffer
        g_u8_cur_buf++;
        if( NB_DATA_BUF == g_u8_cur_buf )
            g_u8_cur_buf = 0;
    }
    return TRUE;
}

//! @brief Store buffers available in file
//!
//! @return FALSE, in case of error
//!
Bool datalogging_file_write_all(void)
{
    // Write buffers which are full
    while( g_b_buf_full[g_u8_cur_buf] ) {
        if( !datalogging_file_write_sector() )
            return FALSE; // Error on disk
    }
    // Write data remaining
    if( g_u16_buf_pos_wr != (g_u8_cur_buf*FS_SIZE_OF_SECTOR) )
    {
        if( !file_write_buf( &g_data_buf[g_u8_cur_buf*FS_SIZE_OF_SECTOR], g_u16_buf_pos_wr-
(g_u8_cur_buf*FS_SIZE_OF_SECTOR) ) )
            return FALSE; // Error write
    }

    return TRUE;
}

//! @brief Update GPX header file before close it
//!
//! @return FALSE, in case of error
//!
Bool datalogging_file_update_gpx_header(void)
{
    U32 u32_data;
    if( !file_seek( 4 , FS_SEEK_SET ) )
        return FALSE;
    u32_data = nav_file_lgt() -8;
    file_putc( LSB0(u32_data) );
    file_putc( LSB1(u32_data) );
    file_putc( LSB2(u32_data) );
    file_putc( LSB3(u32_data) );
    if( !file_seek( 40 , FS_SEEK_SET ) )
        return FALSE;
    u32_data -= (44-8);
    file_putc( LSB0(u32_data) );
    file_putc( LSB1(u32_data) );
}

```

```

    file_putc( LSB2(u32_data) );
    file_putc( LSB3(u32_data) );
    return TRUE;
}

//! @brief Init GPX header file
//!

void datalogging_file_init_gpx_header(void)
{
    // Write the GPX header file in buffer
    // "<?xml version='1.0' encoding='UTF-8' standalone='no' ?>gpx
    xmlns='http://www.topografix.com/GPX/1/1' creator='' version='1.1'
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xsi:schemaLocation='http://www.topografix.com/GPX/1/1
    http://www.topografix.com/GPX/1/1/gpx.xsd'>";
    str2Buf("<?xml version=");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf("1.0");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf(" encoding=");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf("UTF-8");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf(" standalone=");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf("no");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf(" ?>");
    linea[0]= 10;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf("<gpx xmlns=");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf("http://www.topografix.com/GPX/1/1");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf(" creator=");
    linea[0]= 34;
    linea[1]= 34;
    linea[2]= 0;
    str2Buf(linea);
    str2Buf(" version=");
    linea[0]= 34;
    linea[1]= 0;
    str2Buf(linea);
    str2Buf("1.1");
    linea[0]= 34;

```

```

        linea[1]= 0;
        str2Buf(linea);
        str2Buf(" xmlns:xsi=");
        linea[0]= 34;
        linea[1]= 0;
        str2Buf(linea);
        str2Buf("http://www.w3.org/2001/XMLSchema-instance");
        linea[0]= 34;
        linea[1]= 0;
        str2Buf(linea);
        str2Buf(" xsi:schemaLocation=" );
        linea[0]= 34;
        linea[1]= 0;
        str2Buf(linea);
        str2Buf("http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd");
        linea[0]= 34;
        linea[1]= '>';
        linea[2]= 10;
        linea[3]= 32;
        linea[4]= 0;
        str2Buf(linea);
    }

void datalogging_file_init_gpx_track(void)
{
    // Write the GPX header file in buffer
    char texto[20];
    str2Buf("<trk>");
    linea[0]= 10;
    linea[1]= 32;
    linea[2]= 32;
    linea[3]= 0;
    str2Buf(linea);
    str2Buf("<name>LOG ");

    texto[0] = '2';
    texto[1] = '0';
    texto[2] = FechaGPS[4]; //año
    texto[3] = FechaGPS[5];
    texto[4] = '-';
    texto[5] = FechaGPS[2]; //mes
    texto[6] = FechaGPS[3];
    texto[7] = '-';
    texto[8] = FechaGPS[0]; //dia
    texto[9] = FechaGPS[1];
    texto[10] = ' ';
    texto[11] = horaGPS[0];
    texto[12] = horaGPS[1];
    texto[13] = ':';
    texto[14] = horaGPS[2];
    texto[15] = horaGPS[3];
    texto[13] = ':';
    texto[14] = horaGPS[4];
    texto[15] = horaGPS[5];
    texto[16] = 0;

    str2Buf(texto);

    str2Buf("</name>");
}

```

```

void datalogging_file_init_gpx_tracksegment(void)
{
    // Write the GPX header file in buffer
    // Update data buffer position
    str2Buf("<trkseg>");
}

void datalogging_file_end_gpx_tracksegment(void)
{
    // Write the GPX header file in buffer
    // Update data buffer position
    str2Buf("</trkseg>");
}

void datalogging_file_end_gpx_track(void)
{
    // Write the GPX header file in buffer
    // Update data buffer position
    str2Buf("</trk>");
}

void datalogging_file_end_gpx(void)
{
    /*
    // Write the GPX header file in buffer
    char texto[20] = "</gpx>";
    strcat( g_data_buf, texto );
    // Update data buffer position
    g_u16_buf_pos_wr = g_u16_buf_pos_wr + 6;
    */

    str2Buf("</gpx>");
}

void str2Buf(char* strAdd)
{
    do
    {
        g_data_buf[g_u16_buf_pos_wr] = *strAdd++;
        g_u16_buf_pos_wr++;
        if( 0 == (g_u16_buf_pos_wr%FS_SIZE_OF_SECTOR) )
        {
            g_b_buf_full[g_u8_cur_buf_wr] = TRUE;
            if( DATA_BUF_SIZE == g_u16_buf_pos_wr )
            {
                g_u16_buf_pos_wr = 0;
                g_u8_cur_buf_wr = 0;
            }else{
                g_u8_cur_buf_wr++;
            }
        }
    }
    while(*strAdd != 0);
}

```

```

/*****
Function: uart_puts()
Purpose: transmit string to UART
Input: string to be transmitted
Returns: none
*****/
void uart_puts(const char *s )
{
    while (*s)
        uart_putchar(*s++);
}/* uart_puts */
void Menu(void)
{
    switch (g_u8_screen)
    {
        case 0:
            if(Is_hwb() || Is_joy_up() || Is_joy_down() || Is_joy_left() || Is_joy_right() ||
Is_btn_middle())
            {
                lcd_clrscr();
                lcd_goto(0x00);
                lcd_puts("> Grabar");
                lcd_goto(0x40);
                lcd_puts(" Configurar");
                g_u8_screen = 1;
                g_u8_opcion = 0;
            }
            while(Is_hwb() || Is_joy_up() || Is_joy_down() || Is_joy_left() || Is_joy_right() ||
Is_btn_middle());
            break;
        case 1:
            if( Is_joy_up() || Is_joy_down() || Is_joy_left() || Is_joy_right() ||
Is_btn_middle())
            {
                if (g_u8_opcion == 0)
                {
                    if(Is_joy_up() || Is_joy_down())
                    {
                        lcd_clrscr();
                        lcd_goto(0x00);
                        lcd_puts(" Grabar");
                        lcd_goto(0x40);
                        lcd_puts("> Configurar");
                        g_u8_screen = 1;
                        g_u8_opcion = 1;
                        while (Is_joy_up() || Is_joy_down());
                        break;
                    }
                }
                if (Is_btn_middle())
                {
                    lcd_clrscr();
                    lcd_goto(0x00);
                    lcd_puts("> Parar ");
                    g_u8_screen = 2;
                    g_u8_opcion = 0;
                    g_b_OrdenGrabar = true;
                    while (Is_btn_middle());
                    break;
                }
            }
            if (g_u8_opcion == 1)
            {
                if(Is_joy_up() || Is_joy_down())
                {

```

```

        lcd_clrscr();
        lcd_goto(0x00);
        lcd_puts("> Grabar");
        lcd_goto(0x40);
        lcd_puts(" Configurar");
        g_u8_screen = 1;
        g_u8_opcion = 0;
        while (Is_joy_up() || Is_joy_down());
        break;
    }
    if (Is_btn_middle())
    {
        lcd_clrscr();
        lcd_goto(0x00);
        lcd_puts("> Formato Cada");
        lcd_goto(0x42);
        if (g_u8_formatFile == 1)
        {
            lcd_puts("NMEA");
        }
        else
        {
            lcd_puts("GPX ");
        }
        lcd_goto(0x49);
        lcd_puts(g_list_txt_GrabarCada[g_u8_sel_GrabarCada]);
        g_u8_screen = 3;
        g_u8_opcion = 0;
        while (Is_btn_middle());
        break;
    }
}
}
break;
case 2:
if( Is_hwb())
{
    lcd_clrscr();
    lcd_goto(0x00);
    lcd_puts("> Grabar");
    lcd_goto(0x40);
    lcd_puts(" Configurar");
    g_b_OrdenGrabar = false;
    g_u8_screen = 1;
    g_u8_opcion = 0;
}
while(Is_hwb());
break;
case 3:
if(Is_hwb())
{
    lcd_clrscr();
    lcd_goto(0x00);
    lcd_puts("> Grabar");
    lcd_goto(0x40);
    lcd_puts(" Configurar");
    g_u8_screen = 1;
    g_u8_opcion = 0;
}
while(Is_hwb());
switch (g_u8_opcion)
{
case 0:
{

```

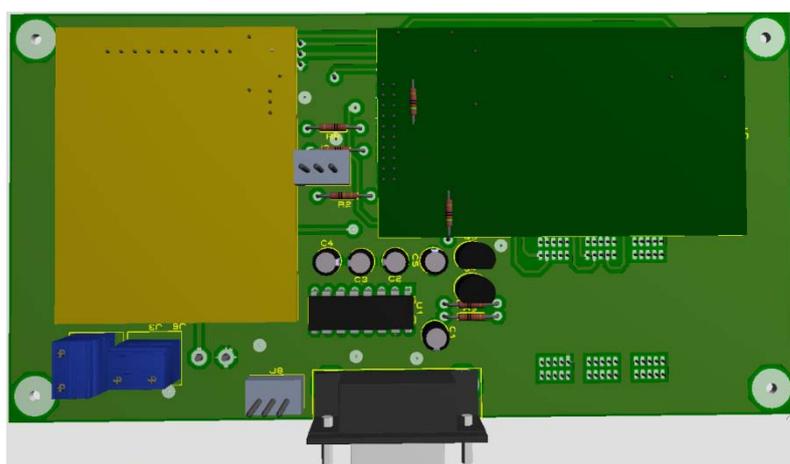


### 3.6 Modelo 3D.

A continuación se muestra la visualización en 3D de las dos caras del circuito impreso, donde se observa la pantalla LCD, la tarjeta AT90USBKEY en la vista frontal, mientras en la otra vista se divisa la tarjeta del receptor GPS y la fuente de poder además del circuito de conexión RS232.



**Figura 39:** Vista Frontal del Proyecto



**Figura 40:** Vista Posterior del Proyecto

### **3.7 Ensamblado de tarjetas.**

#### **3.7.1 Tarjeta Principal.**

Esta tarjeta tiene los conectores necesarios para acoplar la tarjeta de fuente de poder, el receptor de GPS y la tarjeta AT90USBKEY.

Luego de obtenida la PCB se comprobó que no tenga errores de fabricación, luego se la ensambló con todos sus componentes y conectores se procedió a verificar que las soldaduras se hayan realizado sin errores, como cortocircuitos.

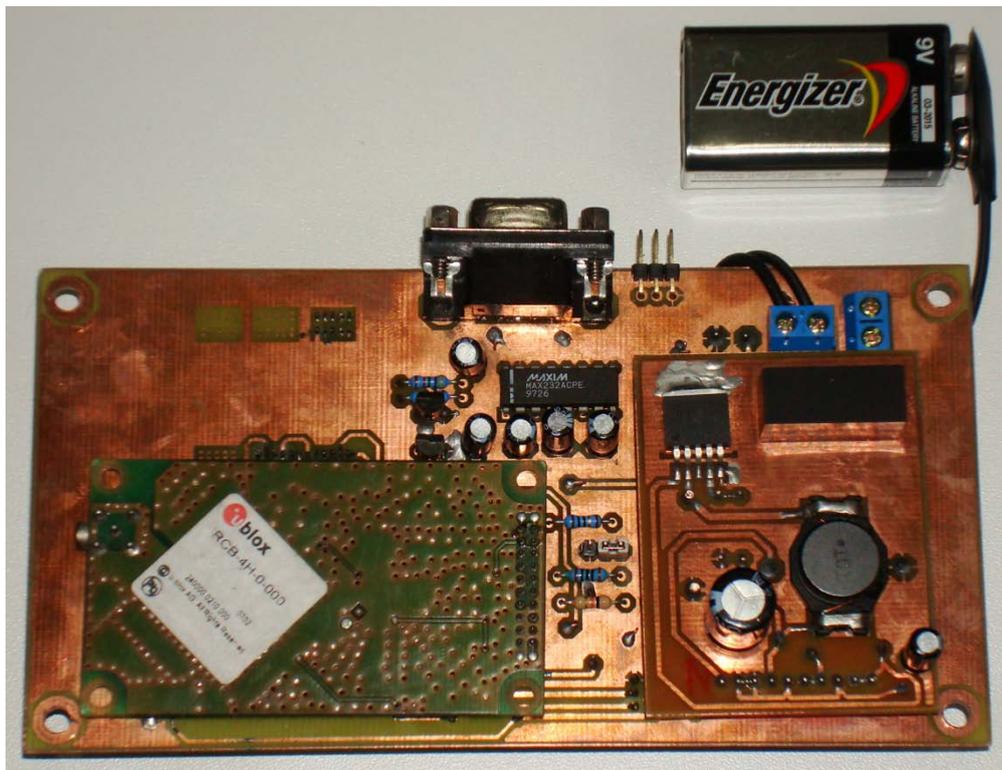
#### **3.7.2 Fuente de poder.**

Una vez desarrollado el PCB de esta tarjeta se procedió a verificar el funcionamiento de las salidas de voltaje que provee, obteniéndose los voltajes esperados

Se diseñó un módulo de fuente de voltajes para la circuitería, teniendo en cuenta que los voltajes de operación son de 3.3V para el receptor de GPS, 5V para la pantalla y 9 voltios para la tarjeta AT90USBKEY.



**Figura 41: Fotografía Frontal del Circuito.**



**Figura 42: Fotografía Posterior del Circuito.**

## **CAPÍTULO 4**

### **SIMULACIONES Y PRUEBAS.**

Debido a que Proteus no tiene soporte para simular el microcontrolador AT90USB1287 no se realizaron simulaciones y se procedió a ensamblar la tarjeta diseñada para el proyecto. A continuación se describe las pruebas realizadas:

#### ***4.1 Pruebas de Funcionamiento.***

Para el desarrollo del proyecto primero se desarrolló la tarjeta principal, puesto que al no poder hacer simulaciones, se realizaron depuraciones directamente gracias a que el JTAG ICE mkII ofrece la funcionalidad.

##### **4.1.1 Grabación en formato NMEA.**

Cuando el equipo es configurado para grabar datos en formato NMEA, se toman los datos que provee el receptor de GPS y se los guarda en la unidad USB Flash.

La extensión del archivo donde se graban los datos es txt, para que el usuario pueda accederlos fácilmente a través de una aplicación de sencilla de edición de texto.

Se muestran datos grabados por el equipo en modo TXT:

```
$GPRMC,145251.00,A,0211.48729,S,07953.34606,W,0.326,211.50,301110,,D*6D
$GPRMC,145252.00,A,0211.48479,S,07953.34501,W,0.950,218.36,301110,,D*6E
$GPRMC,145253.00,A,0211.48249,S,07953.34347,W,1.162,227.35,301110,,D*69
$GPRMC,145254.00,A,0211.48263,S,07953.34212,W,0.202,216.59,301110,,D*6B
$GPRMC,145255.00,A,0211.48114,S,07953.34080,W,0.373,222.74,301110,,D*6F
$GPRMC,145256.00,A,0211.47682,S,07953.33767,W,0.384,231.09,301110,,D*62
$GPRMC,145257.00,A,0211.47368,S,07953.33627,W,0.693,237.52,301110,,D*6C
$GPRMC,145258.00,A,0211.47076,S,07953.33512,W,1.103,241.09,301110,,D*6A
$GPRMC,145259.00,A,0211.46792,S,07953.33363,W,0.943,246.27,301110,,D*61
$GPRMC,145300.00,A,0211.46515,S,07953.33230,W,1.180,252.43,301110,,D*67
$GPRMC,145301.00,A,0211.46362,S,07953.33157,W,1.471,256.31,301110,,D*68
$GPRMC,145302.00,A,0211.45936,S,07953.32949,W,1.905,260.65,301110,,D*6F
```

#### 4.1.2 Grabación en formato GPX

Si el equipo es configurado para grabar en formato GPX, se crea el archivo con extensión .gpx y se van grabando los datos secuencialmente, cumpliendo las especificaciones del archivo.

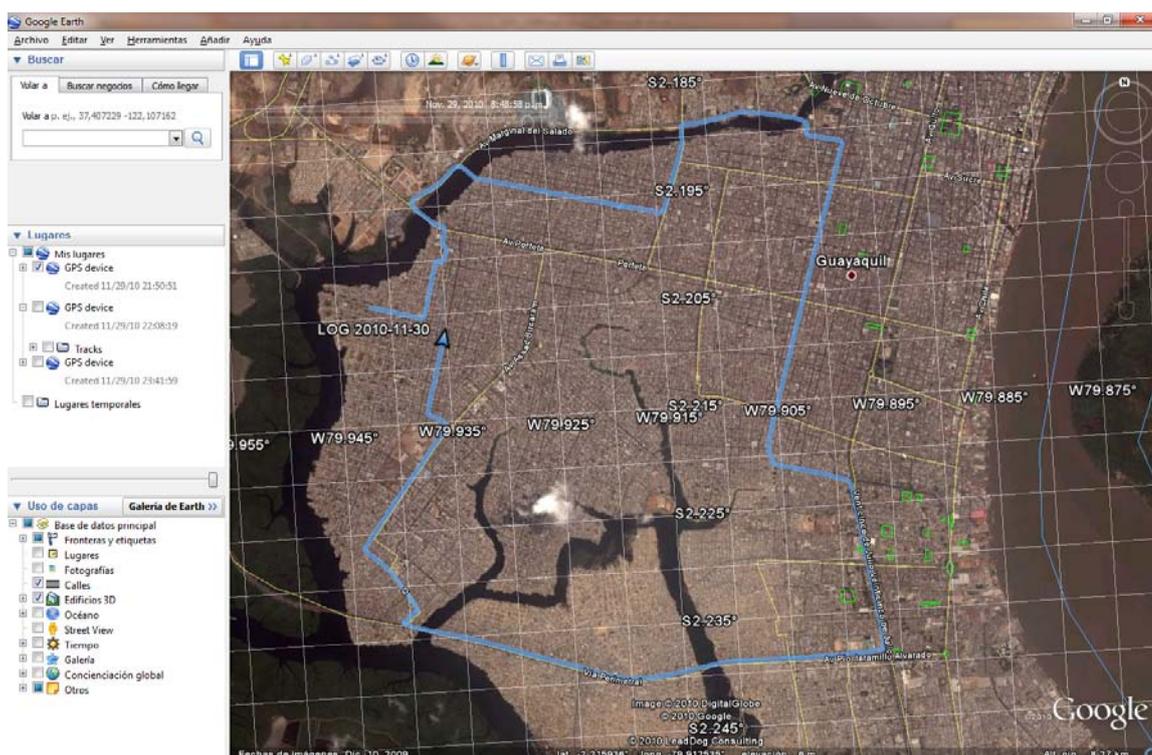
A continuación se muestran datos contenidos en un archivo de registro del GPS en formato GPX, más adelante se mostrarán estos datos ya importados en la aplicación Google Earth.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<gpx xmlns="http://www.topografix.com/GPX/1/1" creator="" version="1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd">
  <trk>
    <name>LOG 2010-11-30</name><trkseg>
```

```

<trkpt lat="-02.2039198" lon="-079.9420183"> <ele>0.0</ele> <time>2010-11-
30T01:00:57Z</time> </trkpt>
<trkpt lat="-02.2039486" lon="-079.9419245"> <ele>0.0</ele> <time>2010-11-
30T01:00:58Z</time> </trkpt>
<trkpt lat="-02.2039770" lon="-079.9418295"> <ele>0.0</ele> <time>2010-11-
30T01:00:59Z</time> </trkpt>
<trkpt lat="-02.2040033" lon="-079.9417348"> <ele>0.0</ele> <time>2010-11-
30T01:01:00Z</time> </trkpt>
.
.
.
<trkpt lat="-02.2027411" lon="-079.9364493"> <ele>0.0</ele> <time>2010-11-
30T01:52:11Z</time> </trkpt>
<trkpt lat="-02.2027460" lon="-079.9364390"> <ele>0.0</ele> <time>2010-11-
30T01:52:12Z</time> </trkpt>
<trkpt lat="-02.2027478" lon="-079.9364381"> <ele>0.0</ele> <time>2010-11-
30T01:52:13Z</time> </trkpt>
<trkpt lat="-02.2027390" lon="-079.9364496"> <ele>0.0</ele> <time>2010-11-
30T01:52:14Z</time> </trkpt> </trkseg></trk></gpx>

```



**Figura 43:** Datos GPS importados en Google Earth®

# CONCLUSIONES Y RECOMENDACIONES.

Las conclusiones son:

1. Se cumplió con el objetivo de almacenar la información obtenida del receptor GPS en el USB Flash Drive, utilizando la tarjeta de desarrollo AT90USBKEY y principalmente el microcontrolador AT90USB1287 mostrando información al usuario en la pantalla LCD.
2. El desarrollo de la aplicación en lenguaje de programación de alto nivel, como es C, permitió la fácil migración de rutinas de un compilador a otro, además de un menor tiempo necesario para el aprendizaje del manejo de otro compilador en el mismo lenguaje.
3. Al utilizar tarjetas de desarrollo se acortó el tiempo de prueba del prototipo ya que traen incorporadas ciertas herramientas, como en el caso de la tarjeta AT90USBKEY que contiene al microcontrolador principal AT90USB1287 y viene con un joystick (5 botoneras), 4 LED, memoria Flash SPI 16MB, 1 sensor de Temperatura y botón de RESET.

4. Los microcontroladores han evolucionado y cada vez realizan tareas de mayor complejidad que antes se las realizaba solamente con computadoras, en este proyecto ya no se necesita una computadora para acceder a una Unidad USB Flash con sistema de archivos FAT12, FAT16 o FAT32.

Las recomendaciones son:

1. Verificar que las entradas y salidas del módulo de GPS sea compatible con el voltaje de operación de la tarjeta AT90USBKEY, esto es 3.3 V.
2. Siempre antes de poner a funcionar un equipo debe de leerse el manual de usuario, para evitar daños en equipos o personal.
3. Cuando se utilice tarjetas de desarrollo, debe verificarse la compatibilidad de ésta con los otros circuitos que se vaya a interconectar en el sistema. Una diferencia de voltajes de funcionamiento, puede conllevar a la destrucción de algún elemento.
4. Implementar más proyectos en esta área que permiten al humano tener datos de posición que pueden ser utilizados en la automatización de procesos.

5. Continuar con la evolución de estos proyectos, puesto que al ser prototipos se pueden agregar más funciones, que pueden implementarse en los siguientes cursos. Una opción de mejora puede ser el utilizar un protocolo de comunicación binario que son más compactos que el NMEA. Otro proyecto que podría implementarse es complementar los datos de GPS a los de acelerómetros para estimar la posición de un vehículo con receptor GPS cuando se pierde señal de satélites.
  
6. Continuar con la integración de estos proyectos, para generar sistemas más complejos como podría ser utilizar el equipo que se ha implementado para registrar rutas y que estos datos sean utilizados para controlar a un vehículo para que siga la ruta, verificando cada cierto tiempo su posición actual detectada por GPS y acelerómetros contra la ruta grabada.

## **ANEXOS**

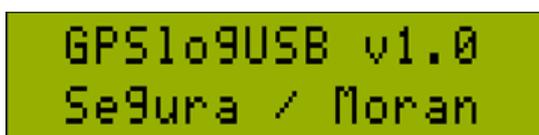
## ANEXO I

### **Manual del Usuario del Equipo.**

Conecte con cuidado la fuente de poder. El equipo no tiene protección contra conexión invertida de voltaje.

La fuente de poder debe estar en el rango 8 – 15V.

Al encender el aparato se observa en la pantalla LCD La siguiente información



**Figura 1: Pantalla de Presentación**

Se recomienda insertar el dispositivo de almacenamiento masivo USB Flash.

La unidad USB Flash debe tener el formato de archivos FAT, sea este FAT16 o FAT32.

Verifique que el Indicador de Unidad USB Flash se encienda.

Para continuar se presiona una de las direcciones de la palanca de mando, posterior a esta acción se mostrará la siguiente pantalla



**Figura 2: Pantalla para Seleccionar Grabar o Configurar**

En esta pantalla el usuario podrá iniciar la grabación de datos, con los valores por defecto, o si lo desea, configurar el equipo.

Al presionar la palanca de mando, cuando está seleccionada la opción de Grabar, el equipo inicia la grabación de la información, que lo muestra encendiendo intermitentemente el LED indicador de Grabación (Instalado en la tarjeta AT90USBKEY). Además se muestra la siguiente opción de Parar, como muestra la figura:



**Figura 3: Pantalla que se muestra mientras está Almacenando datos**

Mientras el equipo se encuentra grabando información, no puede desconectarse el dispositivo de almacenamiento masivo USB Flash, puesto que se dañaría el archivo donde se está grabando la información.

Si el usuario desea detener la grabación, debe presionar la palanca de mando.

Para retirar la unidad USB Flash, luego de parar la grabación debe esperarse a que el equipo libere el sistema de archivos



**Figura 4: Pantalla Seleccionar Configurar**

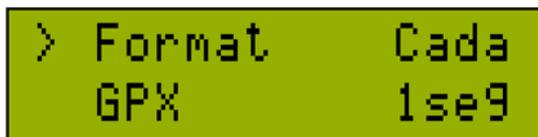
Para cambiar las opciones de grabación, se elige configurar y se presiona la palanca de mando.

Se muestra la facilidad para cambiar el formato de grabación del archivo y las opciones de grabación disponibles son:

GPX

NMEA

Para elegir una de estas opciones se presiona la palanca de mando hacia arriba o hacia abajo



```
> Format      Cada
  GPX        1se9
```

**Figura 5: Pantalla de Configuración del Formato de Datos**

Para cambiar la frecuencia de grabación se mueve la palanca de mando hacia la derecha y se observa que se ha seleccionado la opción “Cada”, de aquí al presionar la palanca hacia arriba o hacia abajo se van cambiando el Periodo de grabación de datos de posición.



```
Format > Cada
  GPX    1se9
```

**Figura 6: Pantalla Selección Frecuencia de Grabación**

## ANEXO II

### ***Lista de Materiales y Presupuesto.***

<b>Cant.</b>	<b>Descripción</b>	<b>Precio U (USD)</b>	<b>Total (USD)</b>
01	Módulo u-blox RCB-4H	179,00	<b>179,00</b>
01	Pantalla LCD 16X2 TS1620A	25,00	<b>25,00</b>
01	MAX 232	2,00	<b>2,00</b>
01	Antena GPS / VHF	119,00	<b>119,00</b>
04	2N3904	0,10	<b>0,40</b>
01	Diodo LED	0,10	<b>0,10</b>
01	Diodo 1N4148	0,05	<b>0,05</b>
14	Resistencias ¼ W	0,10	<b>1,40</b>
01	Capacitores Electroliticos 0.1uF	0,25	<b>0,25</b>
03	Conectores 1,27 x 2,54 mm 5 x 2 Macho	5,00	<b>15,00</b>
03	Conectores 1,27 x 2,54 mm 5 x 2 Hembra	5,00	<b>15,00</b>
01	Conectores 2 x 2mm 10 x 2 Hembra	5,00	<b>5,00</b>
01	Conectores 2,54 mm 16 x 1 Macho	1,00	<b>1,00</b>
01	Conectores 2,54 mm 16 x 1 Hembra	1,00	<b>1,00</b>
01	Conectores 2,54 mm 10 x 1 Hembra	1,00	<b>1,00</b>
01	LM2576HVS-ADJ	9,13	<b>9,13</b>
01	MBR360RLG	0,39	<b>0,39</b>
01	Capacitores Electroliticos 0.1uF	0,25	<b>0,25</b>
01	Capacitores Electroliticos 100uF	0,40	<b>0,40</b>
01	Capacitores Electroliticos 1000uF	1,20	<b>1,20</b>
01	Inductor 100uH	2,25	<b>2,25</b>
01	Bateria CR-2032	1,50	<b>1,50</b>
01	Bateria 9V	3,50	<b>3,50</b>
01	Conectores 2,54 mm 10 x 1 Macho	1,00	<b>1,00</b>
01	Soporte Bateria CR2032	1,50	<b>1,50</b>
01	Tarjeta Principal	32,00	<b>32,00</b>
01	Tarjeta Fuente de poder	7,00	<b>7,00</b>
<b>Total</b>			<b>\$ 425,32</b>

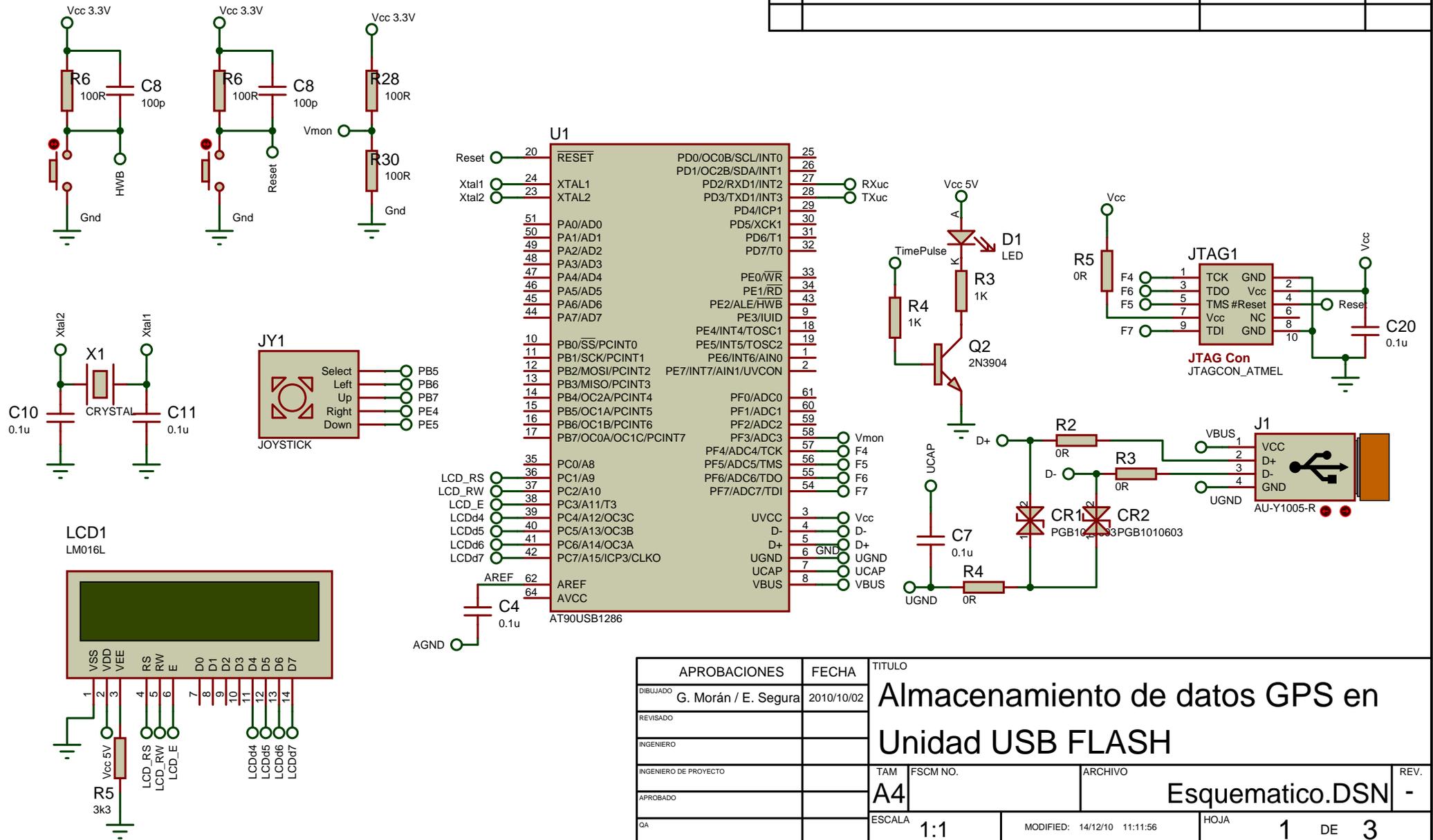
*ANEXO III*

***Diagramas Esquemáticos.***

NOTAS:

HISTORIAL DE REVISION

REV	DESCRIPCION	APROBADO	FECHA

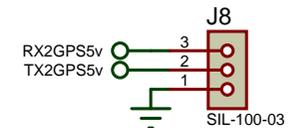
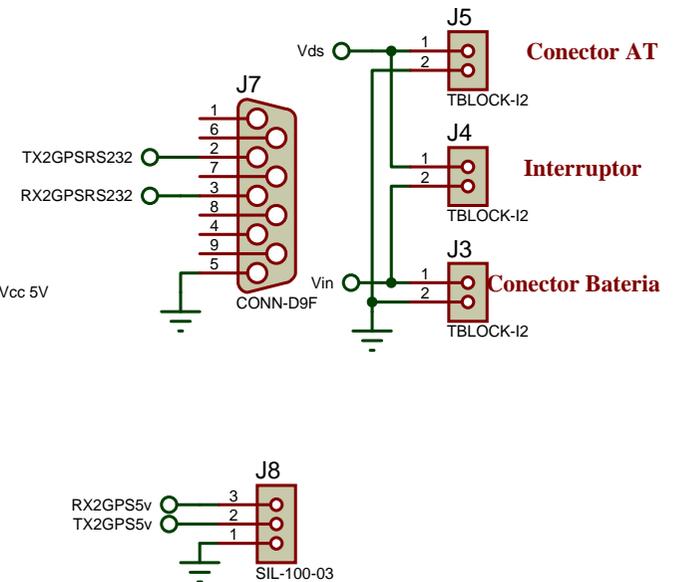
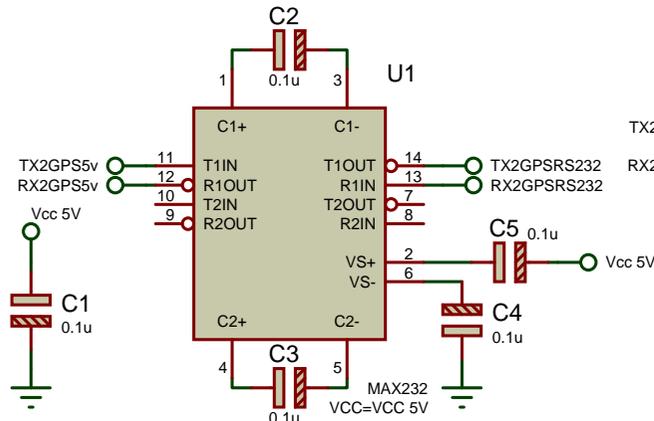
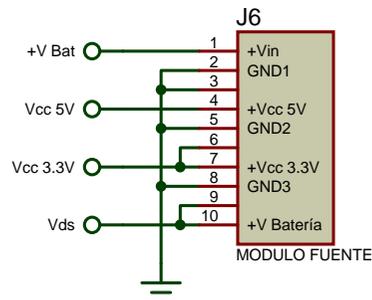
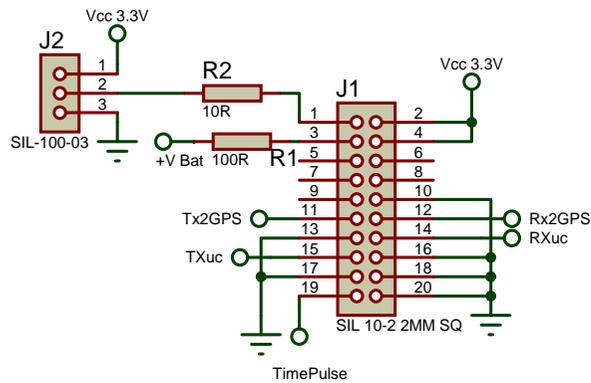
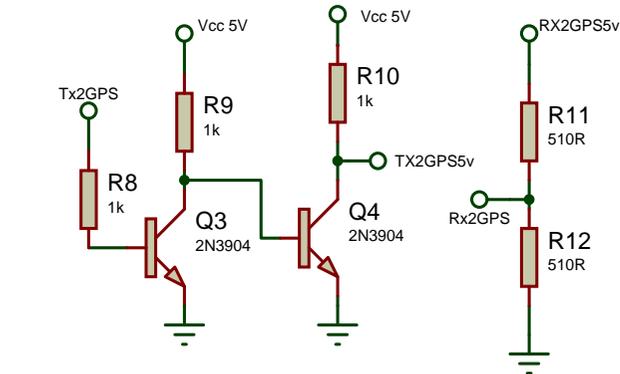


APROBACIONES		FECHA	TITULO	
DIBUJADO	G. Morán / E. Segura	2010/10/02	Almacenamiento de datos GPS en Unidad USB FLASH	
REVISADO				
INGENIERO				
INGENIERO DE PROYECTO			TAM	FSCM NO.
APROBADO			A4	ARCHIVO
QA			ESCALA	MODIFIED: 14/12/10 11:11:56
			1:1	HOJA 1 DE 3

NOTAS:

HISTORIAL DE REVISION

REV	DESCRIPCION	APROBADO	FECHA

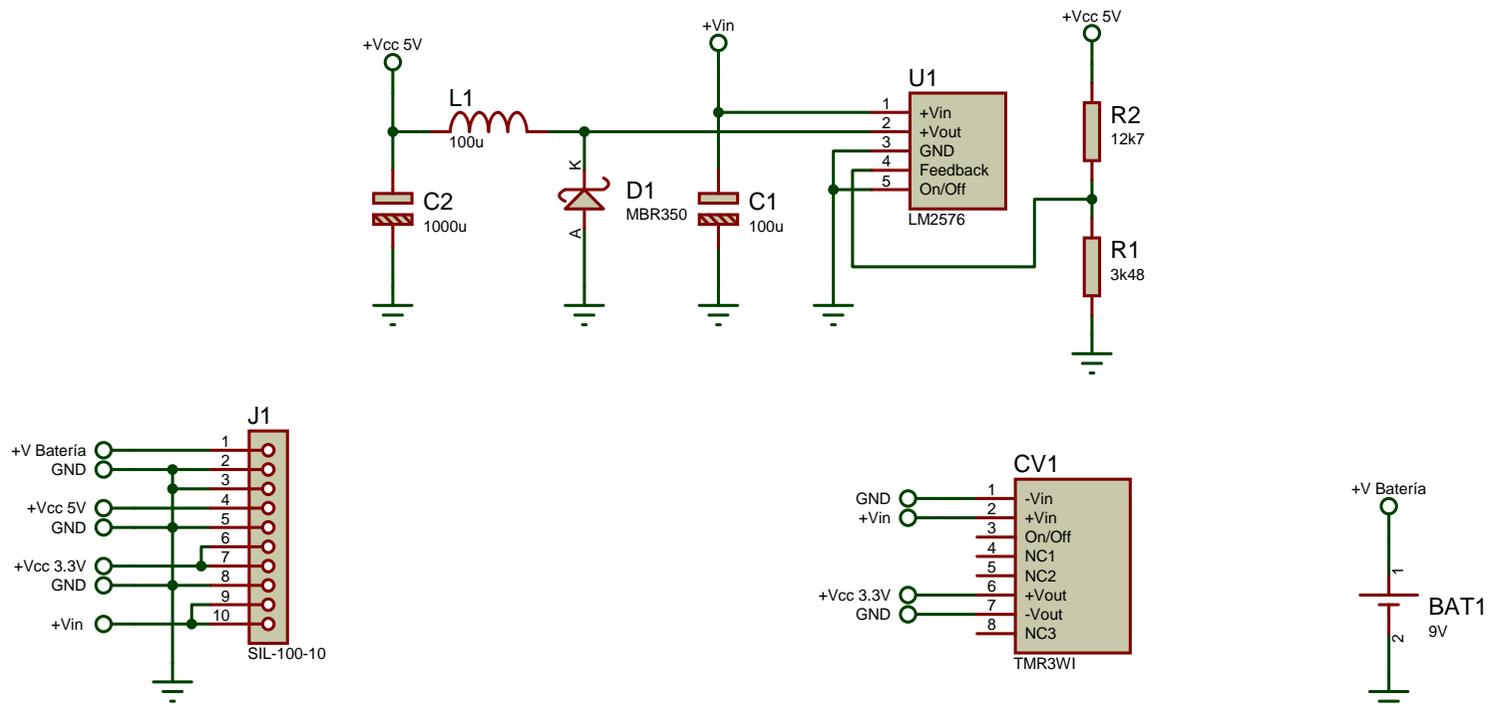


APROBACIONES		FECHA	TITULO		
DIBUJADO	G. Morán / E. Segura	2010/10/02	Almacenamiento de datos GPS en Unidad USB FLASH		
REVISADO					
INGENIERO					
INGENIERO DE PROYECTO			TAM	FSCM NO.	ARCHIVO
APROBADO			A4		Esquematico.DSN
QA			ESCALA	MODIFIED: 14/12/10 11:11:56	HOJA
			1:1		2 DE 3

NOTAS:

HISTORIAL DE REVISION

REV	DESCRIPCION	APROBADO	FECHA



APROBACIONES		FECHA	TITULO		
DIBUJADO	G. Morán / E. Segura	2010/10/02	Almacenamiento de datos GPS en Unidad USB FLASH		
REVISADO					
INGENIERO					
INGENIERO DE PROYECTO			TAM	FSCM NO.	ARCHIVO
APROBADO			A4		Esquematico.DSN
QA			ESCALA	MODIFIED: 14/12/10 11:11:56	HOJA
			1:1		3 DE 3

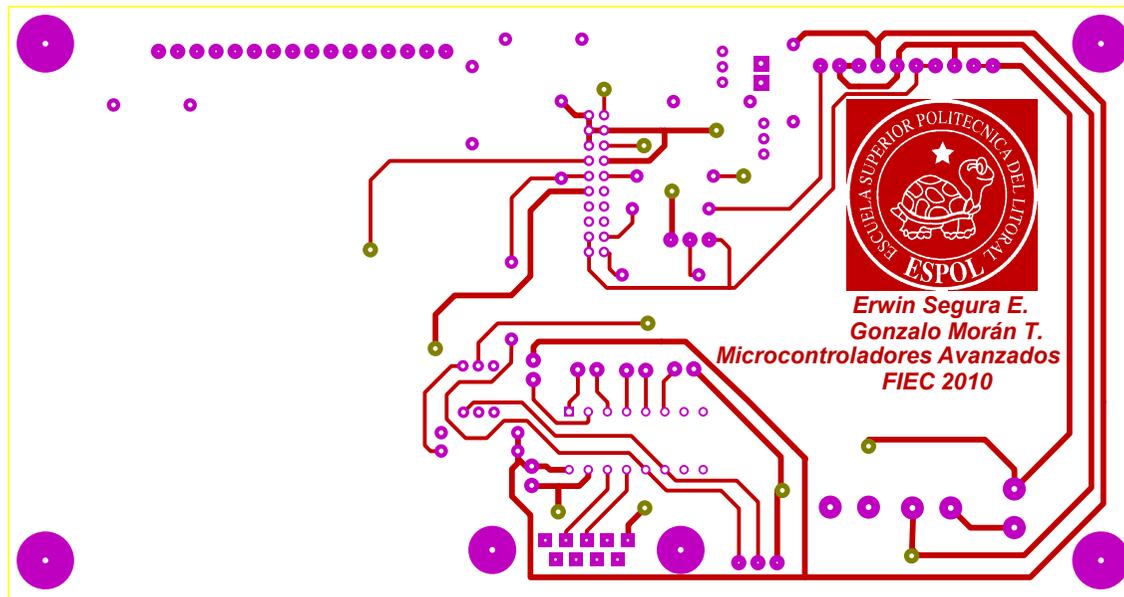
ANEXO IV

***Diagramas Circuitos impresos.***

NOTAS:

HISTORIAL DE REVISION

REV	DESCRIPCION	APROBADO	FECHA

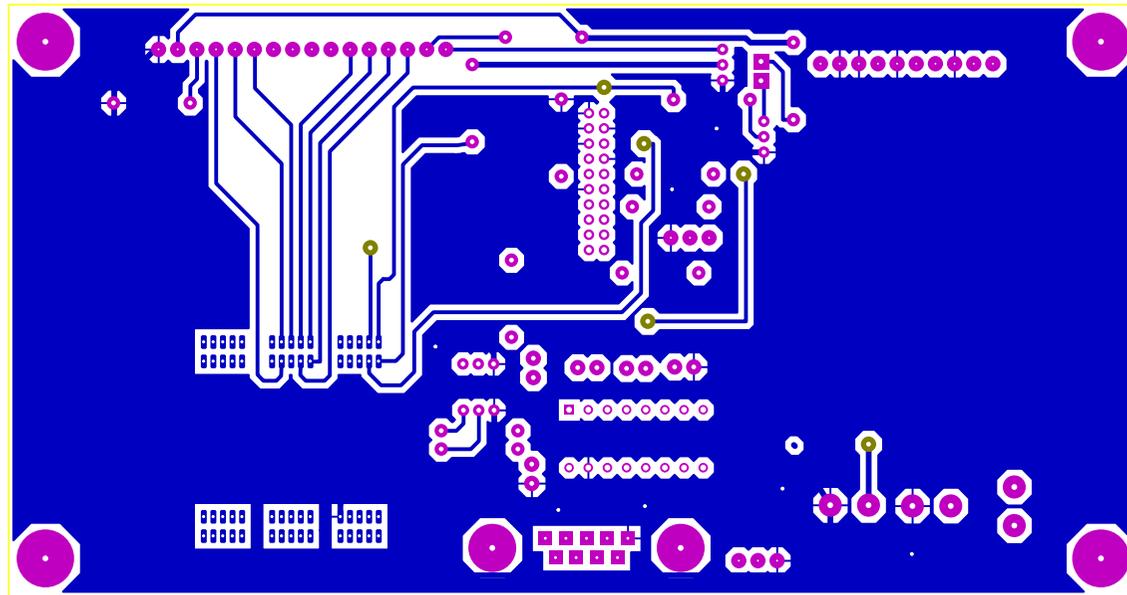


APROBACIONES		FECHA	TITULO	
DIBUJADO	G. Morán / E. Segura	2010/10/02	Almacenamiento de datos GPS en	
REVISADO			Unidad USB FLASH	
INGENIERO			TAM	FSCM NO.
INGENIERO DE PROYECTO			A4	
APROBADO			ARCHIVO	PCB.DSN
QA			ESCALA	REV.
			1:1	-
			MODIFIED: 14/12/10 14:49:49	HOJA 1 DE 2

NOTAS:

HISTORIAL DE REVISION

REV	DESCRIPCION	APROBADO	FECHA



APROBACIONES		FECHA	TITULO		
DIBUJADO G. Morán / E. Segura		2010/10/02	Almacenamiento de datos GPS en		
REVISADO			Unidad USB FLASH		
INGENIERO			TAM	FSCM NO.	ARCHIVO
INGENIERO DE PROYECTO			A4		PCB.DSN
APROBADO			ESCALA	MODIFIED: 14/12/10 14:49:49	REV. -
QA			1:1	HOJA	2 DE 2

ANEXO V

***Hoja de datos RCB-4H.***



**u-blox AG**  
Zürcherstrasse 68  
8800 Thalwil  
Switzerland  
www.u-blox.com

Phone +41 44722 7444  
Fax +41 44 722 7447  
info@u-blox.com

# RCB-4H ANTARIS<sup>®</sup> 4 Programmable GPS Receiver Board with SuperSense<sup>®</sup>

## Data Sheet



### Abstract

This document describes the features and specifications of the GPS receiver board RCB-4H. Based on the 16-channel ANTARIS<sup>®</sup> 4 GPS technology with SuperSense<sup>®</sup>, it offers best GPS performance at very low power consumption. This pluggable board is easy to integrate

*your position is our focus*

Data Sheet

<b>Title</b>	RCB-4H		
<b>Subtitle</b>	ANTARIS® 4 Programmable GPS Receiver Board with SuperSense®		
<b>Doc Type</b>	Data Sheet		
<b>Doc Id</b>	GPS.G4-MS4-06034		
<b>Revision Index</b>	<b>Date</b>	<b>Name</b>	<b>Status / Comments</b>
P1	11. May 2006	GzB	Preliminary Release
-	24.Nov 2006	TN	Storage temperature
A	12. Feb 2007	TG	Release

We reserve all rights to this document and the information contained therein. Reproduction, use or disclosure to third parties without express permission is strictly prohibited.

For most recent documents, please visit [www.u-blox.com](http://www.u-blox.com)

<b>Data Sheet Revisions</b>	<b>Identification of applicable hardware</b>	<b>Comments</b>
P1	All data codes	
-	All data codes	
A	All data codes	

	<p>Products marked with this lead-free symbol on the product label comply with the "Directive 2002/95/EC of the European Parliament and the Council on the Restriction of Use of certain Hazardous Substances in Electrical and Electronic Equipment" (RoHS).</p>
-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Performance characteristics shown in this document are estimates only and do not constitute a warranty or guarantee of product performance. u-blox does not support any applications in connection with weapon systems. Since u-blox' products are not designed for use in life-support and commercial aviation applications they shall not be used in such products. In devices or systems whereby malfunction of these products can be expected to result in personal injury and casualties, u-blox customers using or selling these products do so at their own risk and agree to keep u-blox harmless from any consequences. u-blox reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice.

u-blox makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. u-blox assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

u-blox integrated circuits, software and designs are protected by intellectual property laws in Switzerland and abroad. u-blox, the u-blox logo, the TIM-type GPS module, Antaris, SuperSense®, "your position is our focus", NavLox, u-center, AssistNow, AlmanacPlus, FixNow and EKF are (registered) trademarks of u-blox AG. This product may in whole or in part be subject to intellectual property rights protection. Please contact u-blox for any additional information. Copyright © 2007, u-blox AG.

# 1 Functional Description

## 1.1 Overview

The RCB-4H is an ultra-low power GPS receiver board featuring the ANTARIS®4 positioning engine with SuperSense® Indoor GPS. It is the pin-compatible successor product of the RCB-LJ. The RCB-4H is equipped with a MCX/QSX RF connector and a 20-pin connector for power and digital I/O. The supported A-GPS (Assisted GPS) function provides fast time to first fix even during difficult signal conditions.

The ANTARIS®4 GPS Engine provides high navigation performance even indoors and in other weak signal environments. Satellite based augmentation systems like WAAS, EGNOS and MSAS are fully supported. The combination of high performance and flexibility fulfill the requirements for cost efficiency as well as fast and straightforward plug-in system integration.

## 1.2 Block Diagram

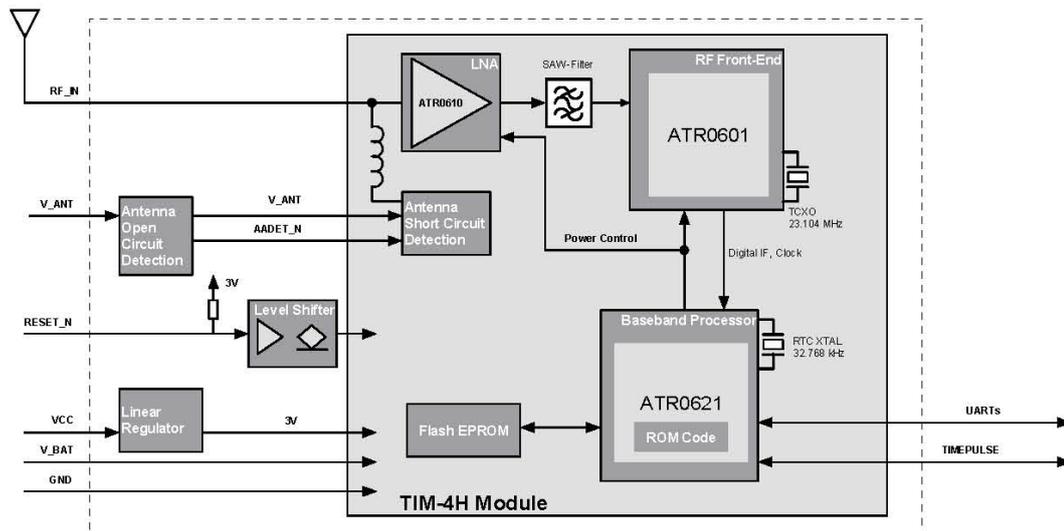


Figure 1: Block Diagram

### 1.3 Highlights

- **SuperSense® Indoor GPS**
- **Ultra low power consumption**
- **A-GPS and Autonomous GPS operation**
- **Supports AssistNow® Online**
- **AssistNow® Offline capable**
- **Flash EPROM: programmable and configurable**
- **4 Hz position update rate**
- **2 UART ports**

### 1.4 Features

- 16 channel ANTARIS®4 positioning engine
- Supports DGPS, WAAS, EGNOS and MSAS
- Power saving modes
- Supports passive and active antennas
- Antenna short and open circuit detection and protection
- Wide supply voltage range: 3.15 to 5.25 V
- Battery voltage supply pin for internal backup memory and real time clock
- Power brown-out protection: No external reset hardware needed
- Industry standard PCB form factor for easy plug-in integration
- Operating temperature range: -40 to 85°C
- RoHS compliant (lead-free)
- Right-angle or straight MCX/OSX connector

## 1.5 Operating Modes

The ANTARIS®4 GPS Technology supports following Operating Modes:

Operating Modes	Description
<b>Continuous Tracking Mode (CTM)</b>	In this mode, the Autonomous Power Management (APM) automatically optimizes power consumption. It powers off parts of the receiver when they are not used. Also, the CPU speed is reduced when the CPU workload is low.
<b>Power Saving Modes</b>	A configurable power saving mode is available where the GPS function is put into sleep mode and activated up on a selectable time interval or upon external request (signal activity on serial port or EXTINT input). This mode is ideally suited in applications with stringent power budget requirements in mobile and battery operated end products.

**Table 1: Operating Modes**

For details, please refer to the *ANTARIS®4 System Integration Manual* [1].

## 1.6 Protocols

The RCB-4H supports different serial protocols. These can be assigned to any serial interface port.

Protocol	Type	Runs on
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0) Standard <b>\$GPxxx</b> and u-blox-specific <b>\$PUBX</b> messages	All serial ports
UBX	Input/output, binary, u-blox proprietary	All serial ports
RTCM	Input, messages 1,2,3,9	All serial ports

**Table 2: Available Protocols**

For specification of the various protocols see the *Protocol Specification* [2].

## 1.7 Assisted GPS (A-GPS)

The ANTARIS® 4 GPS engine supports MS based (accelerated acquisition and position computation in the GPS receiver) A-GPS. Supply of aiding information like ephemeris, almanac, rough last position and time and satellite status reduce time to first fix significantly.

The RCB-4H supports the u-blox AssistNow® Online A-GPS service, and is AssistNow® Offline ready.

## 1.8 Antenna

This GPS receiver is designed for use with passive and active antennas. An antenna supervisor is provided. If activated, the GPS receiver is capable of detecting open or short circuits at the active antenna by checking the bias voltage level and can shut down the voltage bias immediately. UBX and NMEA messages are provided to report the condition of the antenna supply. Open circuit detection is built-in but needs to be activated with a configuration message. For more information see the *ANTARIS<sup>®</sup>4 System Integration Manual* [1].

Parameter	Specification	
Antenna Type	Passive and active antenna	
Active Antenna Recommendations	Minimum gain	15 - 20 dB (to compensate signal loss in RF cable)
	Maximum noise figure	1.5 dB
	Maximum gain	50 dB
Antenna Supply	External	
Antenna Supervisor	Short circuit detection	Built-in
	Open circuit detection	Built-in

**Table 3: Antenna Specification**

## 2 GPS Performance

Parameter	Specification				
Receiver Type	L1 frequency, C/A Code, 16-Channels				
Max Navigation Update Rate	4 Hz				
Accuracy	Position	2.5 m CEP <sup>2</sup>	5.0 m SEP <sup>3</sup>		
	Position DGPS / SBAS <sup>1</sup>	2.0 m CEP	3.0 m SEP		
Acquisition <sup>4,5</sup>	GPS Mode (UBX-CFG Msg):	Fast Acquisition Mode	Normal Mode	High Sensitivity Mode	Auto Mode
	Cold Start	34 s	36 s	41 s	34 s
	Warm Start	33 s			
	Hot Start	<3.5 s			
	Reacquisition	<1 s			
Sensitivity <sup>6</sup>	Tracking	-158 dBm			
	Acquisition & Reacquisition	-148 dBm	Includes aided starts (A-GPS)		
	Cold Starts	-142dBm			
Accuracy of Timepulse Signal	RMS	50 ns			
	99%	<100 ns			
	Granularity	43 ns			
Dynamics	Strong signals	≤ 4 g			
	Weak signals	typ. 1 g			
Operational Limits	Velocity	515 m/s			

**Table 4: GPS Performance Specification**

<sup>1</sup> Depends on accuracy of correction data of DGPS or SBAS service

<sup>2</sup> CEP = Circular Error Probability: The radius of a horizontal circle, centered at the antenna's true position, containing 50% of the fixes.

<sup>3</sup> SEP = Spherical Error Probability. The radius of the sphere, centered at the true position, contains 50% of the fixes.

<sup>4</sup> The different start-up modes like cold, warm and hot start are described in the *ANTARIS<sup>4</sup> System Integration Manual* [1]

<sup>5</sup> Measured with good visibility and -125 dBm signal strength

<sup>6</sup> Demonstrated with a good active antenna

### 3 Mechanical Specification

#### 3.1 Dimensions

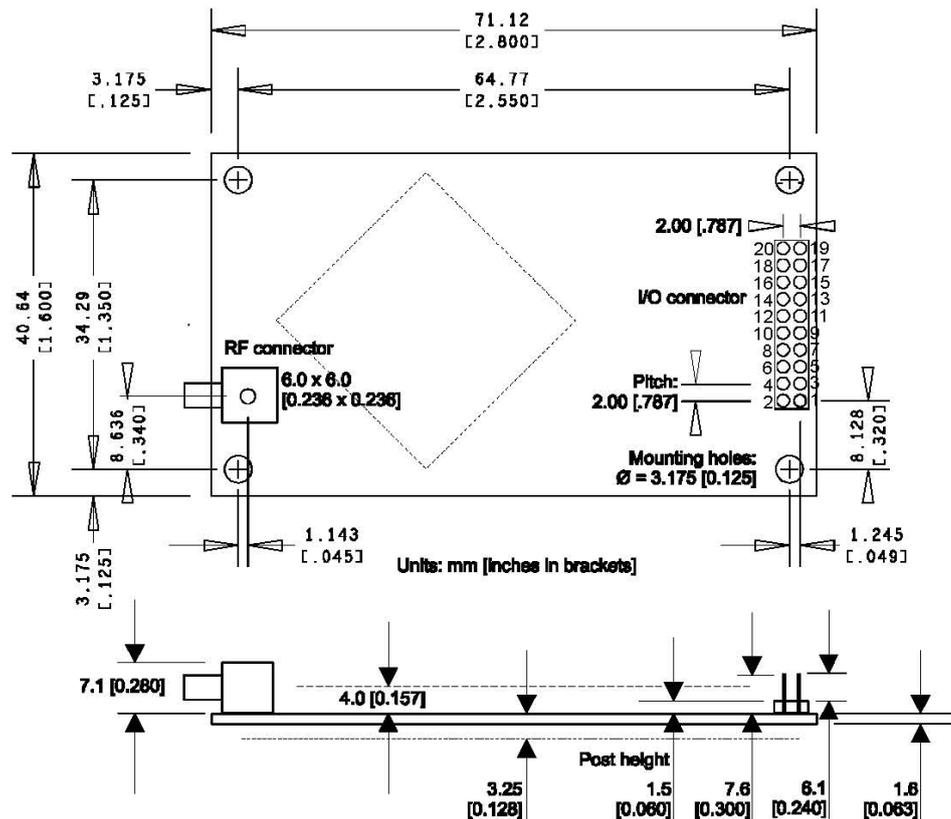


Figure 2: RCB-4H Dimensions (Height of RF connector may vary)

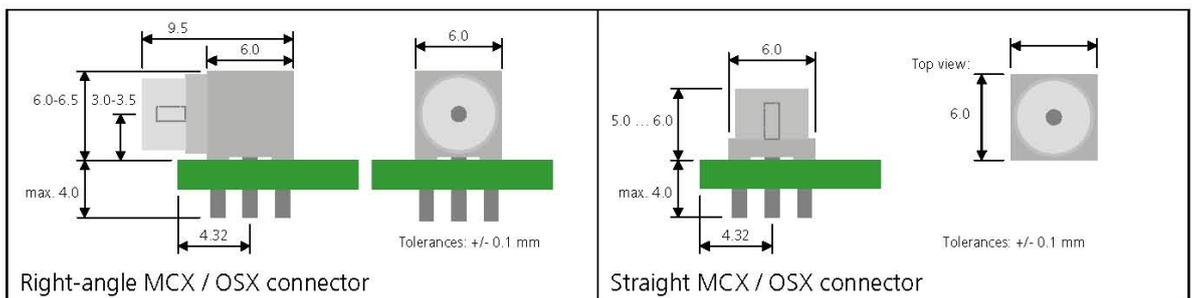


Figure 3: RF Connector Dimensions

### 3.2 Specification

Parameter	Specification	Tolerance	Unit
Length	71.12	±0.3	mm
Width	40.64	±0.3	mm
Thickness	11.0	±0.3	mm
Weight	17		g

**Table 5: Mechanical Specification**

### 3.3 Pinout

RF interface: MCX/OSX, right angle with stand-off, 50 Ohm

Standard Function				Remarks
Pin	Name	I/O	Description	
1	RF_IN	I	GPS signal input	Apply no DC through this pin
-	GND		Shield: Ground	

**Table 6: Pinout of RF Jack**

Digital interface: 20 pin 2mm pitch header

Standard Function				Remarks
Pin	Name	I/O	Description	
1	V_ANT	I	Antenna Bias voltage	Connect to GND if not used
2	VCC	I	Supply Voltage	Internally shorted with pin 4
3	V_BAT	I	Backup voltage supply	Connect to GND if not used
4	VCC	I	Supply Voltage	Internally shorted with pin 2
5	RESET_N	I	Reset (Active low)	
6	Reserved		Not connected	
7	Reserved		Not connected	
8	Reserved		Not connected	
9	Reserved		Not connected	
10	GND	I	Ground	
11	TxD1	O	Serial Port 1	
12	RxD1	I	Serial Port 1	Internal pull-up
13	GND	I	Ground	
14	TxD2	O	Serial Port 2	
15	RxD2	I	Serial Port 2	Internal pull-up
16	GND	I	Ground	
17	GND	I	Ground	
18	GND	I	Ground	
19	TIMEPULSE	O	Timepulse signal	
20	GND	I	Ground	

**Table 7: Pinout of I/O Connector**

## 4 Electrical Specification

### 4.1 Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Units
<b>Power supply</b>				
Power supply voltage (VCC)	Vcc	-0.3	7.0	V
Battery backup supply voltage (V_BAT)	Vbat	-0.3	3.6	V
<b>Digital Inputs (RxD1, RxD2)</b>				
Digital Input pin voltage	Vin	-0.3	5	V
<b>RESET_N</b>				
Reset pin input voltage	VRST	-0.3	6.0	V
<b>RF Input</b>				
Antenna bias voltage (applied via Vant)	Vant	0	6	V
Antenna bias current (applied via Vant)	Iant		100	mA
Input power at RF_IN (source impedance 50Ω, continuous wave)	Prfin		-5	dBm
<b>Environment</b>				
Storage temperature	Tstg	-40	85°	°C

**Table 8: Absolute Maximum Ratings**

- ! Warning** Stressing the device beyond the “Absolute Maximum Ratings” may cause permanent damage. These are stress ratings only. The product is not protected against overvoltage or reversed voltages. If necessary, voltage spikes exceeding the power supply voltage specification, given in table above, must be limited to values within the specified boundaries by using appropriate protection diodes.

## 4.2 Operating Conditions

Parameter <sup>7</sup>	Symbol	Condition	Min	Typ	Max	Units
<b>Power Supply</b>						
Power supply voltage	Vcc		3.15		5.25	V
Power supply voltage ripple	Vcc_PP				50	mV
Sustained Supply Current <sup>8</sup>	Icc	Vcc = 3.3 V		39		mA
Peak Supply Current <sup>9</sup>	Iccp	Vcc = 3.3 V			70	mA
Sleep Mode current	Iccs	Vcc = 3.3 V Vant = 0 V		170		µA
Backup battery voltage (without and with voltage limiter)	Vbat		1.5		3.6	V
Backup battery current	Ibat	Vbat = 3.3 V		5		µA
<b>UART and all I/O Signals<sup>10</sup></b>						
Input pin voltage range	Vin		0		5	V
Input pin low voltage	Vin_low				0.41	V
Input pin high voltage	Vin_high		1.46			V
Output pin voltage range	Vout		0		3.1	V
Output pin low voltage	Vout_low	Iout = 1.5 mA			0.4	V
Output pin high voltage	Vout_high	Iout = 1.5 mA	2.5			V
<b>RESET_N Input</b>						
Input pin voltage range	VinR		0		5.5	V
Input pin low voltage	Vin_lowR				0.9	V
Input pin high voltage	Vin_highR		2.1			V
<b>RF input</b>						
Antenna gain	Gant				50	dB
V_ANT antenna bias voltage (must connect to ground if not used)	Vant		2		6	V
Antenna bias voltage drop	Vant_drop	Iant=10mA		130		mV
Antenna bias current drawn by the RCB-4H, antenna not connected	Iant	Vant = 5 V	1.5		2.0	mA
<b>Environment</b>						
Operating temperature	Topr		-40		85	°C

**Table 9: Operating Conditions**

Running this device beyond the "Operating Conditions" is not recommended and extended exposure beyond them may affect its reliability.

<sup>7</sup> All specification are at an ambient temperature of 25°C.

<sup>8</sup> Average current drawn during Continuous Tracking Mode with 1 Hz update rate, using 9 channels for tracking and navigation and 3 channels for searching satellites (=acquisition). Use this figure to determine required battery capacity

<sup>9</sup> Peak current drawn during initial acquisition phase. Use this figure to dimension maximum current capability of power supply

<sup>10</sup> RxD1, RxD2 provide internal pull-up to V\_BAT18 (Battery supply regulated to 1.8V) and not VCC.

## 5 Environmental Specification

Detailed description of the test series:

Test	Standard
Visual inspection	IPC-A-610 D, class 2 "Acceptability of electronic assemblies"
Temperature step -40°C...+85°C, steps of 5K, operating	ISO 16750-4, IEC68-2-1, IEC68-2-2
Metallographic investigations	IPC-A-600 F, Class 2 "Acceptability of printed boards"
Dry heat +60°C, 5% rH, 1000 hours, operating	IEC68-2-2
Damp heat +60°C, 95% rH, 1000 hours, operating	IEC68-2-3
Thermal shock -40°C...+125°C, 100 cycles	IEC 68-2-14
Lifespan test +85°C/1000 hours, operating	IEC 68-2-2
Dry heat +125°C, 1000 hours, non-operating	IEC68-2-2
Damp heat, cyclic +25°C...+55°C; >90% rH, operating	IEC 68-2-30Db variation 1
Vibration 5-500 Hz, 5g, 2.5 hours/axis at -40°C, +25° and 85°C, operating	IEC 68-2-6
Shock 30g/11ms (half sine), 3 shocks/axis; non-operating	IEC 68-2-27

**Table 10: Environmental Specification**

## 6 Settings

### 6.1 Default Settings

Please refer to the *ANTARIS®4 System Integration Manual* [1] for information about further settings.

Interface	Settings
Serial Port 1 Output	9600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: <b>GGA, GLL, GSA, GSV, RMC, VTG, ZDA, TXT</b> Additional messages can be activated with appropriate input messages.
Serial Port 1 Input	9600 Baud, 8 bits, no parity bit, 1 stop bit, Autobauding disabled Automatically accepts following protocols without need of explicit configuration: <b>UBX, NMEA, RTCM</b> The GPS receiver supports interleaved UBX and NMEA messages.
Serial Port 2 Output	57600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following UBX and no NMEA messages have been activated at start-up: <b>NAV-POSLH, NAV-SOL, NAV-SVINFO, NAV-STATUS</b> MON-IO, MON-SCHD, MON-TXBUF, INF-Warning, INF-Error, INF-Notice Additional messages can be activated with appropriate input messages.
Serial Port 2 Input	57600 Baud, 8 bits, no parity bit, 1 stop bit, Autobauding disabled Automatically accepts following protocols without need of explicit configuration: <b>UBX, NMEA, RTCM</b> The GPS receiver supports interleaved UBX and NMEA messages.
TIMEPULSE	1 pulse per second, synchronized at rising edge, pulse length 100ms

**Table 11: Available Protocols**

Sensitivity Mode	Remark
Auto	The sensitivity mode can be changed with the CFG-RXM message.

**Table 12: Sensitivity Settings**

## 7 Ordering Information

### 7.1 Ordering Information

Ordering No.	Product
RCB-4H-0-000-0	RCB-4H Programmable GPS Receiver Board with SuperSense®
	<u>Delivery Packing</u> 0 = Single samples 1 = Packing unit (100 pieces)
	<u>RF Connectors</u> 0 = Right-angle MCX/OSX 1 = Straight MCX/OSX

Table 13: Ordering Information

## Related Documents

- [1] ANTARIS®4 System Integration Manual, Doc No GPS.G4-MS4-05007
- [2] ANTARIS®4 Protocol Specification, Doc No GPS.G3-X-03002

All these documents are available on our homepage (<http://www.u-blox.com>).



*your position is our focus*

## Contact

For further info, please contact us:

### Headquarters

#### u-blox AG

Zuercherstrasse 68  
CH-8800 Thalwil  
Switzerland

Phone: +41 44 722 74 44

Fax: +41 44 722 74 47

E-mail: [info@u-blox.com](mailto:info@u-blox.com)

[www.u-blox.com](http://www.u-blox.com)

### Sales Offices

#### North, Central and South America

##### u-blox America, Inc.

1902 Campus Commons Drive  
Suite 310  
Reston, VA 20191  
USA

Phone: +1 (703) 483 3180

Fax: +1 (703) 483 3179

E-mail: [info\\_us@u-blox.com](mailto:info_us@u-blox.com)

##### Regional Office West Coast:

8600 Lemon Ave #1  
La Mesa, CA 91941  
USA

Phone: +1 (619) 741 3011

Fax: +1 (619) 741 4334

E-mail: [info\\_us@u-blox.com](mailto:info_us@u-blox.com)

##### Technical Support:

Phone: +1 (703) 483 3185

E-mail: [support\\_us@u-blox.com](mailto:support_us@u-blox.com)

#### Europe, Middle East, Africa

##### u-blox AG

Zuercherstrasse 68  
CH-8800 Thalwil  
Switzerland

Phone: +41 44 722 74 77

Fax: +41 44 722 74 47

E-mail: [info@u-blox.com](mailto:info@u-blox.com)

##### Technical Support:

Phone: +41 44 722 74 74

E-mail: [support@u-blox.com](mailto:support@u-blox.com)

#### Asia, Australia, Pacific

##### u-blox Singapore Pte. Ltd.

435 Orchard Road  
#19-02, Wisma Atria,  
Singapore 238877

Phone: +65 6734 3811

Fax: +65 6736 1533

E-mail: [info\\_ap@u-blox.com](mailto:info_ap@u-blox.com)

Support: [support\\_ap@u-blox.com](mailto:support_ap@u-blox.com)

##### Regional Office China:

Room 716-718  
No. 65 Fuxing Road  
Beijing, 100036, China

Phone: +86 10 68 133 545

Fax: +86 10 68 217 890

E-mail: [info\\_cn@u-blox.com](mailto:info_cn@u-blox.com)

Support: [support\\_cn@u-blox.com](mailto:support_cn@u-blox.com)

##### Regional Office Korea:

1258 Ho, Shinhan MetroKhan  
#1115, BisanDong  
DongAn-Ku  
Anyang-si  
KyungKi-do, Korea

Phone: +82 31 383 2584

Fax: +82 31 386 2584

E-mail: [info\\_kr@u-blox.com](mailto:info_kr@u-blox.com)

Support: [support\\_kr@u-blox.com](mailto:support_kr@u-blox.com)

##### Regional Office Taiwan:

Room 305  
3F, #181, ZouTze Street  
Neihu Dis.  
Taipei, Taiwan

Phone: +886 2 2657 1090

Fax: +886 2 2657 1097

E-mail: [info\\_tw@u-blox.com](mailto:info_tw@u-blox.com)

Support: [support\\_tw@u-blox.com](mailto:support_tw@u-blox.com)

*ANEXO VI*

***Resumen Hoja de datos AT90USB1287.***

---

## Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-Chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
  - 64/128K Bytes of In-System Self-Programmable Flash
    - Endurance: 100,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    - USB Bootloader programmed by default in the Factory
    - In-System Programming by On-chip Boot Program hardware activated after reset
    - True Read-While-Write Operation
    - All supplied parts are preprogrammed with a default USB bootloader
  - 2K/4K (64K/128K Flash version) Bytes EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 4K/8K (64K/128K Flash version) Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- USB 2.0 Full-speed/Low-speed Device and On-The-Go Module
  - Complies fully with:
    - Universal Serial Bus Specification REV 2.0
    - On-The-Go Supplement to the USB 2.0 Specification Rev 1.0
    - Supports data transfer rates up to 12 Mbit/s and 1.5 Mbit/s
- USB Full-speed/Low Speed Device Module with Interrupt on Transfer Completion
  - Endpoint 0 for Control Transfers : up to 64-bytes
  - 6 Programmable Endpoints with IN or Out Directions and with Bulk, Interrupt or Isochronous Transfers
  - Configurable Endpoints size up to 256 bytes in double bank mode
  - Fully independent 832 bytes USB DPRAM for endpoint memory allocation
  - Suspend/Resume Interrupts
  - Power-on Reset and USB Bus Reset
  - 48 MHz PLL for Full-speed Bus Operation
  - USB Bus Disconnection on Microcontroller Request
- USB OTG Reduced Host :
  - Supports Host Negotiation Protocol (HNP) and Session Request Protocol (SRP) for OTG dual-role devices
  - Provide Status and control signals for software implementation of HNP and SRP
  - Provides programmable times required for HNP and SRP
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Two 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode



---

**8-bit AVR<sup>®</sup>  
Microcontroller  
with  
64/128K Bytes  
of ISP Flash  
and USB  
Controller**

---

**AT90USB646  
AT90USB647  
AT90USB1286  
AT90USB1287**

7593K-AVR-11/09

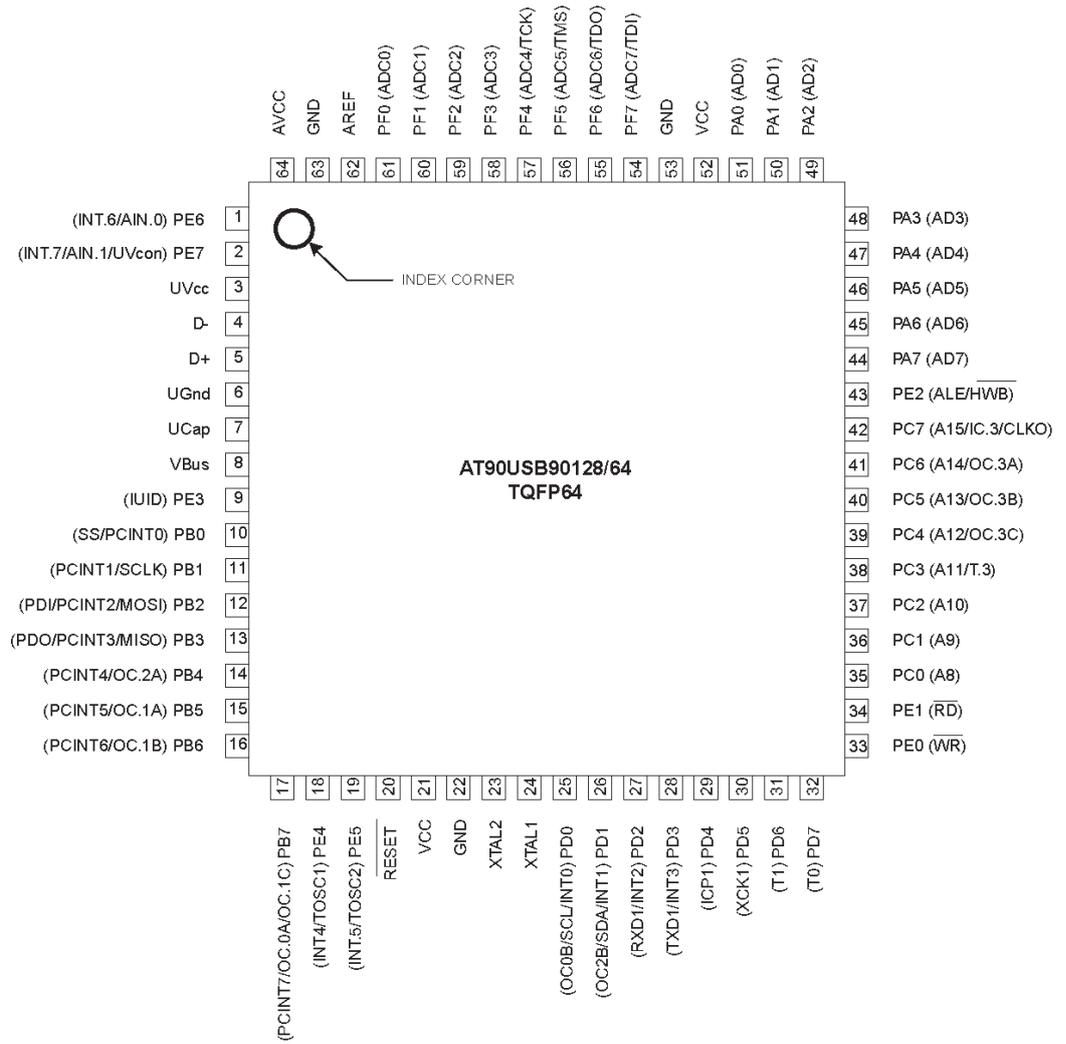




- Real Time Counter with Separate Oscillator
- Four 8-bit PWM Channels
- Six PWM Channels with Programmable Resolution from 2 to 16 Bits
- Output Compare Modulator
- 8-channels, 10-bit ADC
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Byte Oriented 2-wire Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 48 Programmable I/O Lines
  - 64-lead TQFP and 64-lead QFN
- Operating Voltages
  - 2.7 - 5.5V
- Operating temperature
  - Industrial (-40°C to +85°C)
- Maximum Frequency
  - 8 MHz at 2.7V - Industrial range
  - 16 MHz at 4.5V - Industrial range

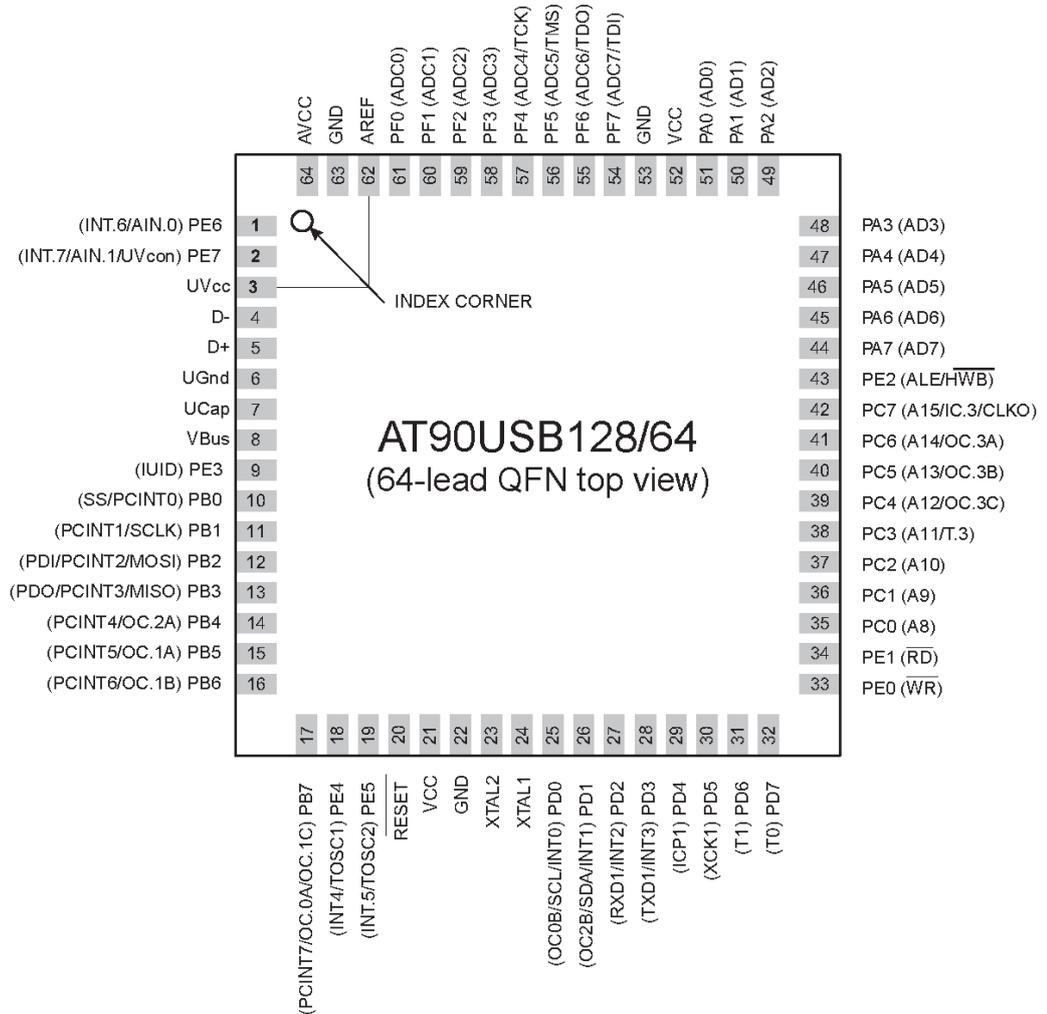
# 1. Pin Configurations

Figure 1-1. Pinout AT90USB64/128-TQFP





**Figure 1-2.** Pinout AT90USB64/128-QFN



Note: The large center pad underneath the MLF packages is made of metal and internally connected to GND. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board.

### 1.1 Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

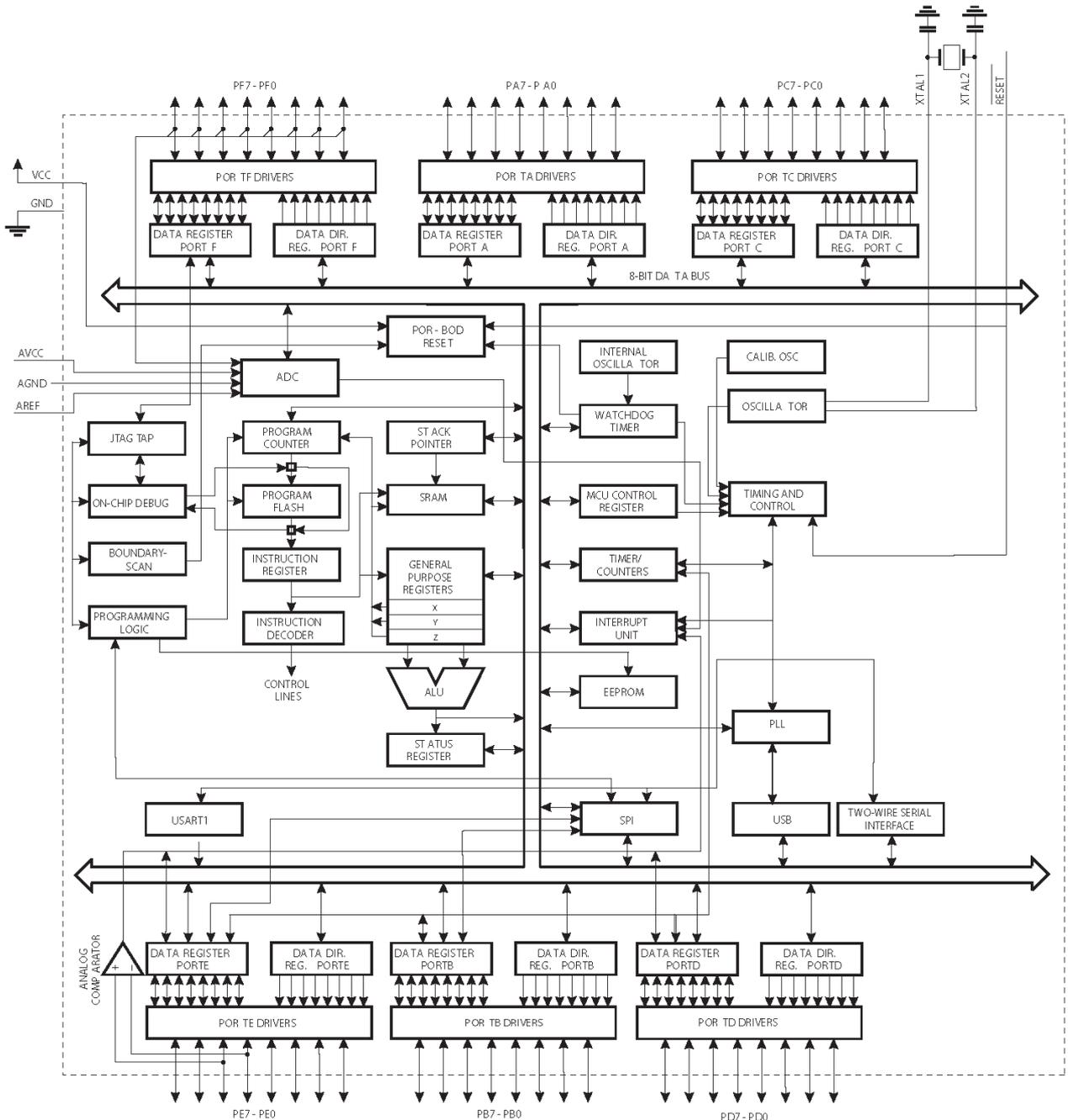
## 2. Overview

The AT90USB64/128 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the

AT90USB64/128 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## 2.1 Block Diagram

Figure 2-1. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The AT90USB64/128 provides the following features: 64/128K bytes of In-System Programmable Flash with Read-While-Write capabilities, 2K/4K bytes EEPROM, 4K/8K bytes SRAM, 48 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible Timer/Counters with compare modes and PWM, one USART, a byte oriented 2-wire Serial Interface, a 8-channels, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel AT90USB64/128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The AT90USB64/128 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## 2.2 Pin Descriptions

### 2.2.1 VCC

Digital supply voltage.

### 2.2.2 GND

Ground.

### 2.2.3 AVCC

Analog supply voltage.

### 2.2.4 Port A (PA7..PA0)

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

Port A also serves the functions of various special features of the AT90USB64/128 as listed on [page 79](#).

### 2.2.5 Port B (PB7..PB0)

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

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the AT90USB64/128 as listed on [page 80](#).

### 2.2.6 Port C (PC7..PC0)

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

Port C also serves the functions of special features of the AT90USB64/128 as listed on [page 83](#).

### 2.2.7 Port D (PD7..PD0)

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

Port D also serves the functions of various special features of the AT90USB64/128 as listed on [page 84](#).



### 2.2.8 Port E (PE7..PE0)

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

Port E also serves the functions of various special features of the AT90USB64/128 as listed on [page 87](#).

### 2.2.9 Port F (PF7..PF0)

Port F serves as analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

Port F also serves the functions of the JTAG interface.

### 2.2.10 D-

USB Full speed / Low Speed Negative Data Upstream Port. Should be connected to the USB D- connector pin with a serial 22 Ohms resistor.

### 2.2.11 D+

USB Full speed / Low Speed Positive Data Upstream Port. Should be connected to the USB D+ connector pin with a serial 22 Ohms resistor.

### 2.2.12 UGND

USB Pads Ground.

### 2.2.13 UVCC

USB Pads Internal Regulator Input supply voltage.

### 2.2.14 UCAP

USB Pads Internal Regulator Output supply voltage. Should be connected to an external capacitor (1 $\mu$ F).

### 2.2.15 VBUS

USB VBUS monitor and OTG negotiations.

### 2.2.16 $\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in [Table 8-1 on page 58](#). Shorter pulses are not guaranteed to generate a reset.

### 2.2.17 XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

**2.2.18 XTAL2**

Output from the inverting Oscillator amplifier.

**2.2.19 AVCC**

AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter.

**2.2.20 AREF**

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

### 3. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBRC", "SBR", and "CBR".

*ANEXO VII*

***Guía de Usuario AT90USBKEY.***

**AT90USBKey**

.....  
**Hardware User Guide**





---

<b>Section 1</b>	
Introduction .....	1-3
1.1 Overview .....	1-3
1.2 AT90USBKey Features.....	1-4

---

<b>Section 2</b>	
Using the AT90USBKey .....	2-5
2.1 Overview .....	2-5
2.2 Power Supply .....	2-6
2.3 Reset.....	2-8
2.4 On-board Resources.....	2-9
2.5 In-System Programming .....	2-13
2.6 Debugging .....	2-14

---

<b>Section 3</b>	
Troubleshooting Guide .....	3-15

---

<b>Section 4</b>	
Technical Specifications .....	4-16

---

<b>Section 5</b>	
Technical Support.....	5-17

---

<b>Section 6</b>	
Complete Schematics .....	6-18



# Section 1

---

## Introduction

Congratulations on acquiring the AVR® AT90USBKey. This kit is designed to give designers a quick start to develop code on the AVR® and for prototyping and testing of new designs with the AT90USB microcontroller family.

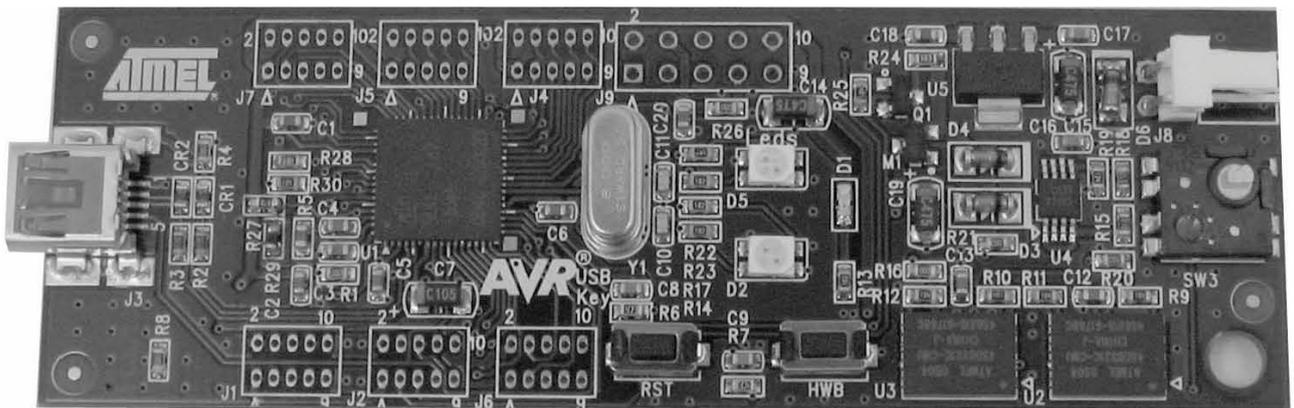
---

### 1.1 Overview

This document describes the AT90USBKey dedicated to the AT90USB AVR microcontroller. This board is designed to allow an easy evaluation of the product using demonstration software.

To increase its demonstrative capabilities, this stand alone board has numerous on-board resources: USB, joystick, data-flash and temperature sensor.

**Figure 1-1 .** AT90USBKey



---

## 1.2 AT90USBKey Features

The AT90USBKey provides the following features:

- AT90USB QFN64
- AVR Studio® software interface <sup>(1)</sup>
- USB software interface for Device Firmware Upgrade (DFU bootloader) <sup>(2)</sup>
- Power supply flagged by "VCC-ON" LED:
  - regulated 3.3V
  - from an external battery connector (for reduced host or OTG operation)
  - from the USB interface (USB device bus powered application)
- JTAG interface (connector not mounted):
  - for on-chip ISP
  - for on-chip debugging using JTAG ICE
- Serial interfaces:
  - 1 USB full/low speed device/host/OTG interface
- On-board resources:
  - 4+1-ways joystick
  - 2 Bi-Color LEDs
  - temperature sensor
  - serial dataflash memories
  - all microcontroller I/O ports access on 2x8pin headers (not mounted)
- On-board RESET button
- On-board HWB button to force bootloader section execution at reset.
- System clock:
  - 8 MHz crystal

- Notes:
1. The AVRUSBKey is supported by AVR Studio®, version 4.12 or higher. For up-to-date information on this and other AVR tool products, please consult our web site. The most recent version of AVR Studio®, AVR tools and this User Guide can be found in the AVR section of the Atmel web site, <http://www.atmel.com>.
  2. ATMEL Flip®, In System Programming Version 3 or Higher shall be used for Device Firmware Upgrade. Please consult Atmel web site to retrieve the latex version of Flip and the DFU bootloader Hex file if needed.



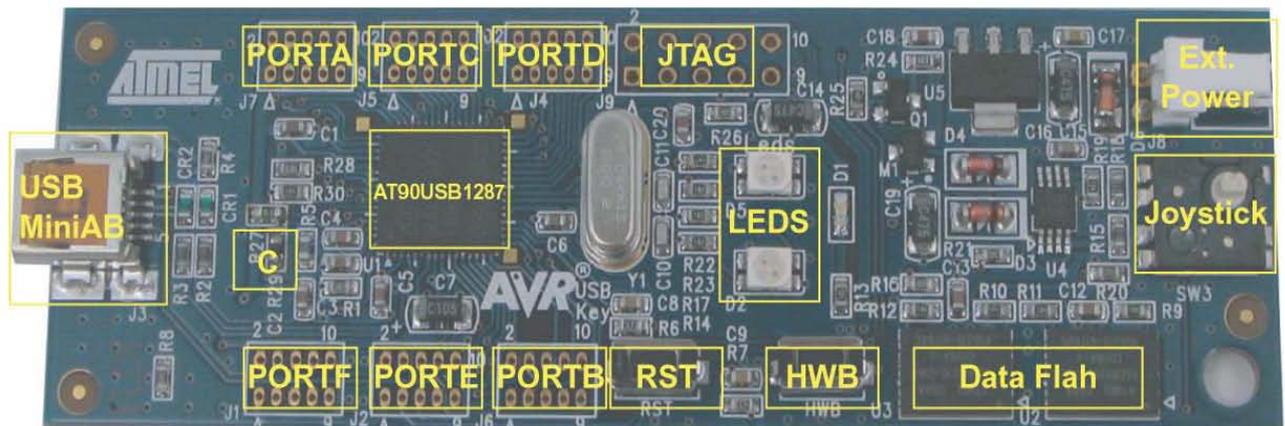
## Section 2

# Using the AT90USBKey

This chapter describes the AVRUSBKey and all its resources.

### 2.1 Overview

Figure 2-1 . AT90USBKey Overview



## 2.2 Power Supply

### 2.2.1 Power Supply Sources

The on-board power supply circuitry allows two power supply configurations:

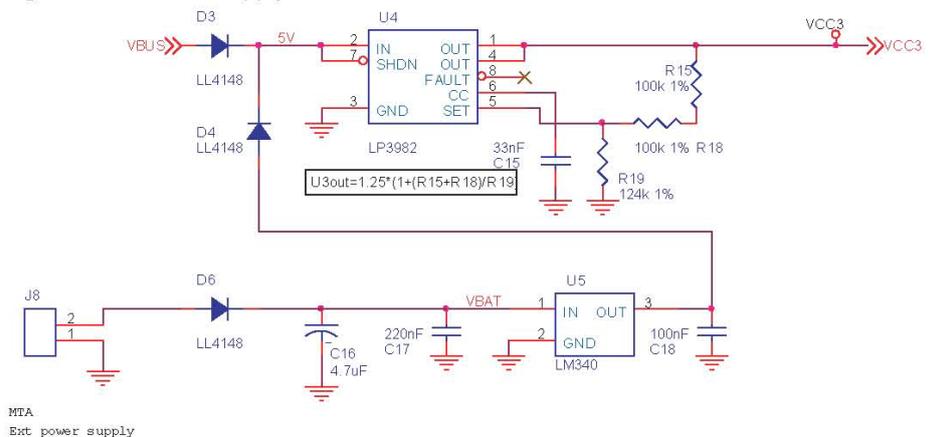
- from USB connector
- from battery connector

*USB powered* When used as a USB device bus powered application, the AVRUSBKey can be directly powered via the USB VBUS power supply line.

*Battery powered* The external battery connector should be used when the AT90USBKey is used as a USB host. This mode allows the AT90USBKey to provide a 5V power supply from its VBUS pin.

- Need of a female battery clip
- Input supply from 8 up to 15V DC (min. 100mA)

**Figure 2-2 .** Power supply schematic

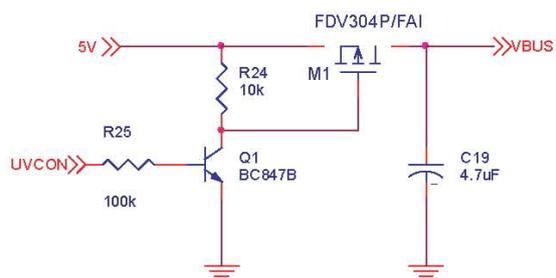


### 2.2.2 VBUS Generator

When using the AT90USB microcontroller in USB host mode, the AT90USBKey should provide a 5V power supply over the VBUS pin of its USB mini AB connector.

A couple of transistors allows the UVCON pin of the AT90USB to control the VBUS generation (See Figure 2-3). In this mode the AT90USBKey is powered by external battery power supply source.

**Figure 2-3 .** VBUS generator schematic



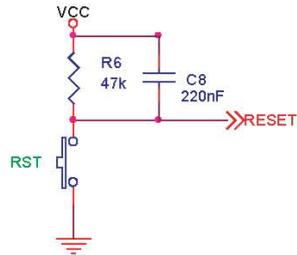
### 2.2.3 “POWER-ON“ LED

The POWER-ON LED (“D1”) is always lit when power is applied to AVRUSBKey regardless of the power supply source.

## 2.3 Reset

Although the AT90USB has its on-chip RESET circuitry (c.f. AT90USB Datasheet, section "System Control and Reset), the AVRUSBKey provides to the AT90USB a RESET signal witch can come from two different sources:

**Figure 2-4 .** Reset Implementation



### 2.3.1 Power-on RESET

The on-board RC network acts as power-on RESET.

### 2.3.2 RESET Push Button

By pressing the RESET push button on the AVRUSBKey, a warm RESET of the AT90USB is performed.

### 2.3.3 Main Clock XTAL

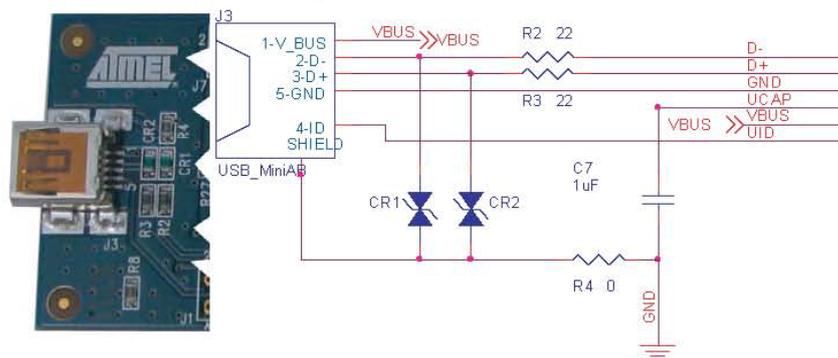
To use the USB interface of the AT90USB, the clock source should always be a crystal or external clock oscillator (the internal 8MHz RC oscillator can not be used to operate with the USB interface). Only the following crystal frequency allows proper USB operations: 2MHz, 4MHz, 6MHz, 8MHz, 12MHz, 16MHz. The AT90USBKey comes with a default 8MHz crystal oscillator.

## 2.4 On-board Resources

### 2.4.1 USB

The AVRUSBKey is supplied with a standard USB mini A-B receptacle. The mini AB receptacle allows to connect both a mini A plug or a mini B plug connectors.

**Figure 2-5 . USB mini A-B Receptacle**



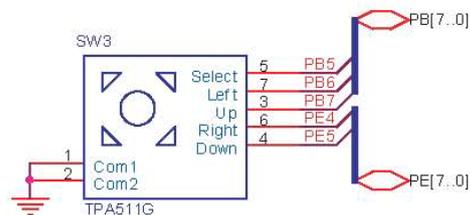
When connected to a mini B plug, the AT90USB operates as an “USB device” (the ID pin of the plug is unconnected) and when connected to a mini A plug, the AT90USB operates as a “USB host” (the ID pin of the A plug is tied to ground).

### 2.4.2 Joystick

The 4+1 ways joystick offers an easy user interface implementation for a USB application (it can emulate mouse movements, keyboard inputs...).

Pushing the push-button causes the corresponding signal to be pulled low, while releasing (not pressed) causes an H.Z state on the signal. The user must enable internal pull-ups on the microcontroller input pins, removing the need for an external pull-up resistors on the push-button.

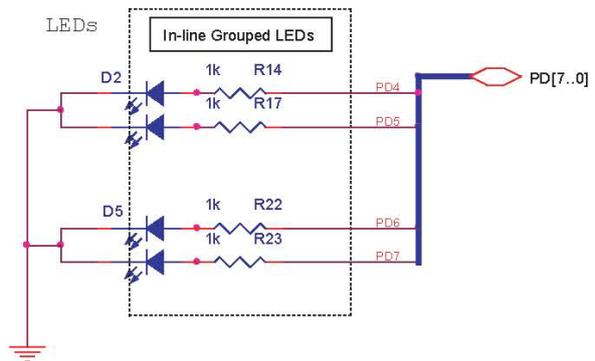
**Figure 2-6 . Joystick Schematic**



### 2.4.3 LEDs

The AT90USBKey includes 2 bi-color LEDs (green/red) implemented on one line. They are connected to the high nibble of “Port D” of AT90USB (PORTD[4..7]). To light on a LED, the corresponding port pin must drive a high level. To light off a LED, the corresponding port pin must drive a low level.

**Figure 2-7 .** LEDs Implementation schematic



**Table 2-1 .** Leds references

LED Reference	AT90USB Connection	Color
D2	PORTD.4	Red
	PORTD.5	Green
D5	PORTD.6	Green
	PORTD.7	Red

### 2.4.4 Temperature Sensor

The temperature sensor uses a thermistor (R29), or temperature-sensitive resistor. This thermistor have a negative temperature coefficient (NTC), meaning the resistance goes up as temperature goes down. Of all passive temperature measurement sensors, thermistors have the highest sensitivity (resistance change per degree of temperature change). Thermistors do not have a linear temperature/resistance curve.

The voltage over the NTC can be found using the A/D converter (connected to channel 0). See the AT90USB Datasheet for how to use the ADC. The thermistor value (RT) is calculate with the following expression:

$$R_T = (R_H \cdot V_{ADC0}) / (V_{CC} - V_{ADC0})$$

- Where:**
- RT** = Thermistor value ( $\Omega$ ) at T temperature ( $^{\circ}$ Kelvin)
  - RH** = Second resistor of the bridge - 100 K $\Omega$   $\pm$ 10% at 25 $^{\circ}$ C
  - VADC0** = Voltage value on ADC-0 input (V)
  - VCC** = Board power supply

The NTC thermistor used in AT90USBKey has a resistance of 100 K $\Omega$   $\pm$ 5% at 25°C (T<sub>0</sub>) and a beta-value of 4250  $\pm$ 3%. By the use of the following equation, the temperature (T) can be calculated:

$$T = \frac{\beta}{\left(\ln \frac{R_T}{R_0}\right) + \frac{\beta}{T_0}}$$

Where: R<sub>T</sub> = Thermistor value ( $\Omega$ ) at T temperature (°Kelvin)  
 $\beta$  = 4250  $\pm$ 3%  
 R<sub>0</sub> = 100 K $\Omega$   $\pm$ 5% at 25°C  
 T<sub>0</sub> = 298 °K (273 °K + 25°K)

The following cross table also can be used. It is based on the above equation.

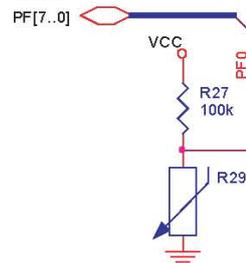
**Table 2-2.** Thermistor Values versus Temperature

Temp. (°C)	R <sub>T</sub> (K $\Omega$ )	Temp. (°C)	R <sub>T</sub> (K $\Omega$ )	Temp. (°C)	R <sub>T</sub> (K $\Omega$ )	Temp. (°C)	R <sub>T</sub> (K $\Omega$ )
-20	1263,757	10	212,958	40	50,486	70	15,396
-19	1182,881	11	201,989	41	48,350	71	14,851
-18	1107,756	12	191,657	42	46,316	72	14,329
-17	1037,934	13	181,920	43	44,380	73	13,828
-16	973,006	14	172,740	44	42,537	74	13,347
-15	912,596	15	164,083	45	40,781	75	12,885
-14	856,361	16	155,914	46	39,107	76	12,442
-13	803,984	17	148,205	47	37,513	77	12,017
-12	755,175	18	140,926	48	35,992	78	11,608
-11	709,669	19	134,051	49	34,542	79	11,215
-10	667,221	20	127,555	50	33,159	80	10,838
-9	627,604	21	121,414	51	31,840	81	10,476
-8	590,613	22	115,608	52	30,580	82	10,128
-7	556,056	23	110,116	53	29,378	83	9,793
-6	523,757	24	104,919	54	28,229	84	9,471
-5	493,555	25	100,000	55	27,133	85	9,161
-4	465,300	26	95,342	56	26,085	86	8,863
-3	438,854	27	90,930	57	25,084	87	8,576
-2	414,089	28	86,750	58	24,126	88	8,300
-1	390,890	29	82,787	59	23,211	89	8,035
0	369,145	30	79,030	60	22,336	90	7,779
1	348,757	31	75,466	61	21,498	91	7,533
2	329,630	32	72,085	62	20,697	92	7,296
3	311,680	33	68,876	63	19,930	93	7,067
4	294,826	34	65,830	64	19,196	94	6,847

Using the AT90USBKey

Temp. (°C)	R <sub>T</sub> (KΩ)						
5	278,995	35	62,937	65	18,493	95	6,635
6	264,119	36	60,188	66	17,820	96	6,430
7	250,134	37	57,576	67	17,174	97	6,233
8	236,981	38	55,093	68	16,556	98	6,043
9	224,606	39	52,732	69	15,964	99	5,860

Figure 2-8 . Thermistor Schematic

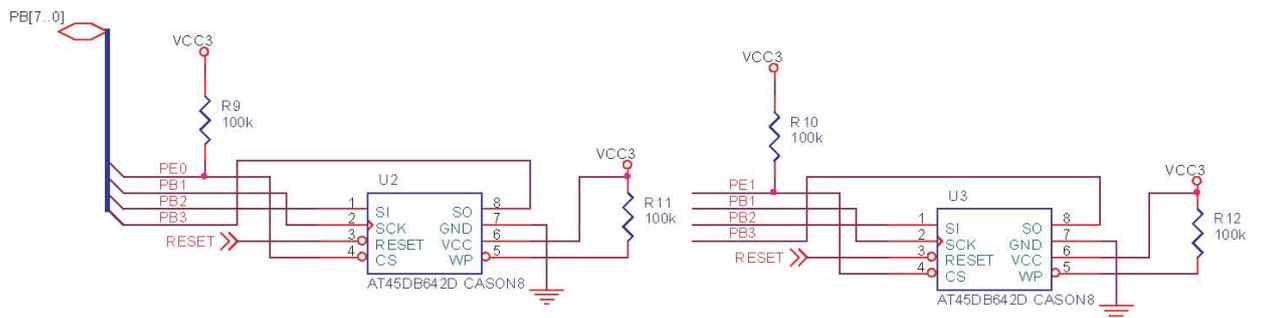


2.4.5 Data Flash memory

For mass-storage class demonstration purpose, the AT90USBKey provides two on-chip serial Flash memories (AT45DB642D) connected to the AT90USB Serial Port Interface (SPI).

The data-flash chip select signals are connected to PortE bit 0 and bit 1 of the AT90USB (See Figure 2-9).

Figure 2-9 . On-board data flash schematic



---

## 2.5 In-System Programming

### 2.5.1 Programming with USB bootloader: DFU (Device Firmware Upgrade)

AT90USB part comes with a default factory pre-programmed USB bootloader located in the on-chip boot section of the AT90USB. This is the easiest and fastest way to reprogram the device directly over the USB interface. The “Flip” PC side application, is available from the Atmel website, offers a flexible and user friendly interface to reprogram the application over the USB bus.

The HWB pin of the AT90USB allows to force the bootloader section execution after reset. (Refer to AT90USB datasheet section “boot loader support”). To force bootloader execution, operate as follow:

- Press both “RST” and “HWB” push buttons
- First release the “RST” push button
- Release the “HWB” push button

For more information about the USB bootloader and “Flip” application, please refer to the “USB bootloader datasheet document” and “Flip’s user manual”.

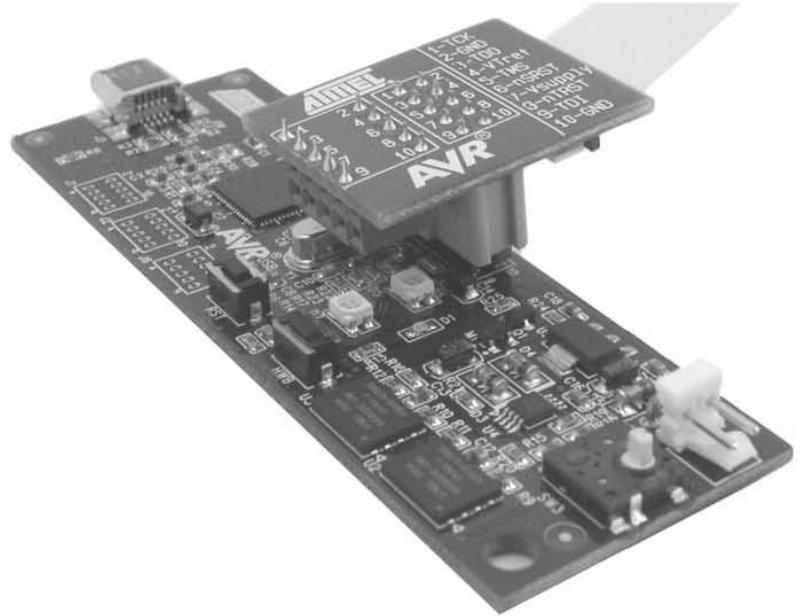
Note: The HWB pin is active only if the HWBE fuse is set (default factory configuration).

### 2.5.2 Programming with AVR JTAGICEmkII

The AT90USB can be programmed using specific JTAG link. To use the AVR JTAGICEmkII with an AT90USBKey an optional HE10 connector should be soldered to J9 footprint. Then the JTAG probe can be connected to the AT90USBKey as shown in Figure 2-10.

Note: When the JTAGEN Fuse is unprogrammed, the four TAP pins are normal port pins, and the TAP controller is in reset. When programmed, the input TAP signals are internally pulled high and the JTAG is enabled for Boundary-scan and programming. The AT90USB device is shipped with this fuse programmed.

**Figure 2-10 .** Connecting AVR JTAG ICE to AVRUSBKey



The Flash, EEPROM and all Fuse and Lock Bit options ISP-programmable can be programmed individually or with the sequential automatic programming option.

Note: See AVR Studio® on-line Help for information.

---

## 2.6 Debugging

### 2.6.1 Debugging with AVR JTAG ICE mkII

The AT90USBKey can be used for debugging with JTAG ICE MK II.

Connect the JTAG ICE mkII as shown in Figure 2-10, for debugging, please refer to AVR Studio® Help information.

When using JTAG ICE MK II for debugging, and as AT90USB parts are factory configured with the higher security level set, a chip erase operation will be performed on the part before debugging. Thus the on-chip flash bootloader will be erased. It can be restored after the debug session using the bootloader hex file available from ATMEL website.



## Section 3

# Troubleshooting Guide

**Figure 3-1 .** Troubleshooting Guide

Problem	Reason	Solution
The Green "VCC-ON" LED is not on	No power supply	Verify the power supply source (check battery charge or USB connection).
AVRUSBKey does not work		
The AT90USB cannot be programmed	The AVR JTAG ICE probe is not connected	Connect the JTAG ICE 10-PIN header to the correct AVRUSBKey JTAG header (page 13)
	The memory lock bits are programmed	Erase the memory before programming with JTAG ICE.
	The fuse bits are wrongly programmed	Check the fuse bits with JTAG ICE
	Can not connect to USB bootloader	Force bootloader execution with HWB under reset. USB bootloader erased after a JTAG debugging session: reprogram the USB bootloader with JTAG.
AVR Studio does not detect the AVR JTAG ICE.	Serial/USB cable is not connected, or power is off	Connect serial cable to RS232 (STK500 - AVR ISP) and check power connections
		Connect serial cable to USB (JTAG ICE MKII, AVR ISPMkIII) and check power connections
	PC COM port is in use	Disable other programs that are using PC COM port.
		Change PC COM port
AVR Studio does not detect COM port.		Disable COM port auto-detection in AVR Studio file menu. Force COM port to correct COM port



## Section 4

---

# Technical Specifications

### ■ System Unit

- Physical Dimensions ..... L=90 x W=30 x H=8 mm
- Weight ..... 12 g

### ■ Operating Conditions

- Internal Voltage Supply ..... 3.3V
- External Voltage Supply ..... 8V -15V

### ■ Connections

- USB Connector ..... Mini AB receptacle
- USB Communications ..... Full speed/low speed
- JTAG Connector ..... Footprint for HE10 connector
- All ports connectors ..... J1, J2, J4, J5, J6, J7
- Battery connector ..... MTA right angle



## Section 5

---

# Technical Support

For Technical support, please contact [avr@atmel.com](mailto:avr@atmel.com). When requesting technical support, please include the following information:

- Which target AVR device is used (complete part number)
- Target voltage and speed
- Clock source and fuse setting of the AVR
- Programming method (ISP, JTAG or specific Boot-Loader)
- Hardware revisions of the AVR tools, found on the PCB
- Version number of AVR Studio. This can be found in the AVR Studio help menu.
- PC operating system and version/build
- PC processor type and speed
- A detailed description of the problem



## Section 6

---

# Complete Schematics

On the next pages, the following documents of AT90USBKey are shown:

- Complete schematics,
- Bill of materials.

Figure 6-1. Schematics, 1 of 2

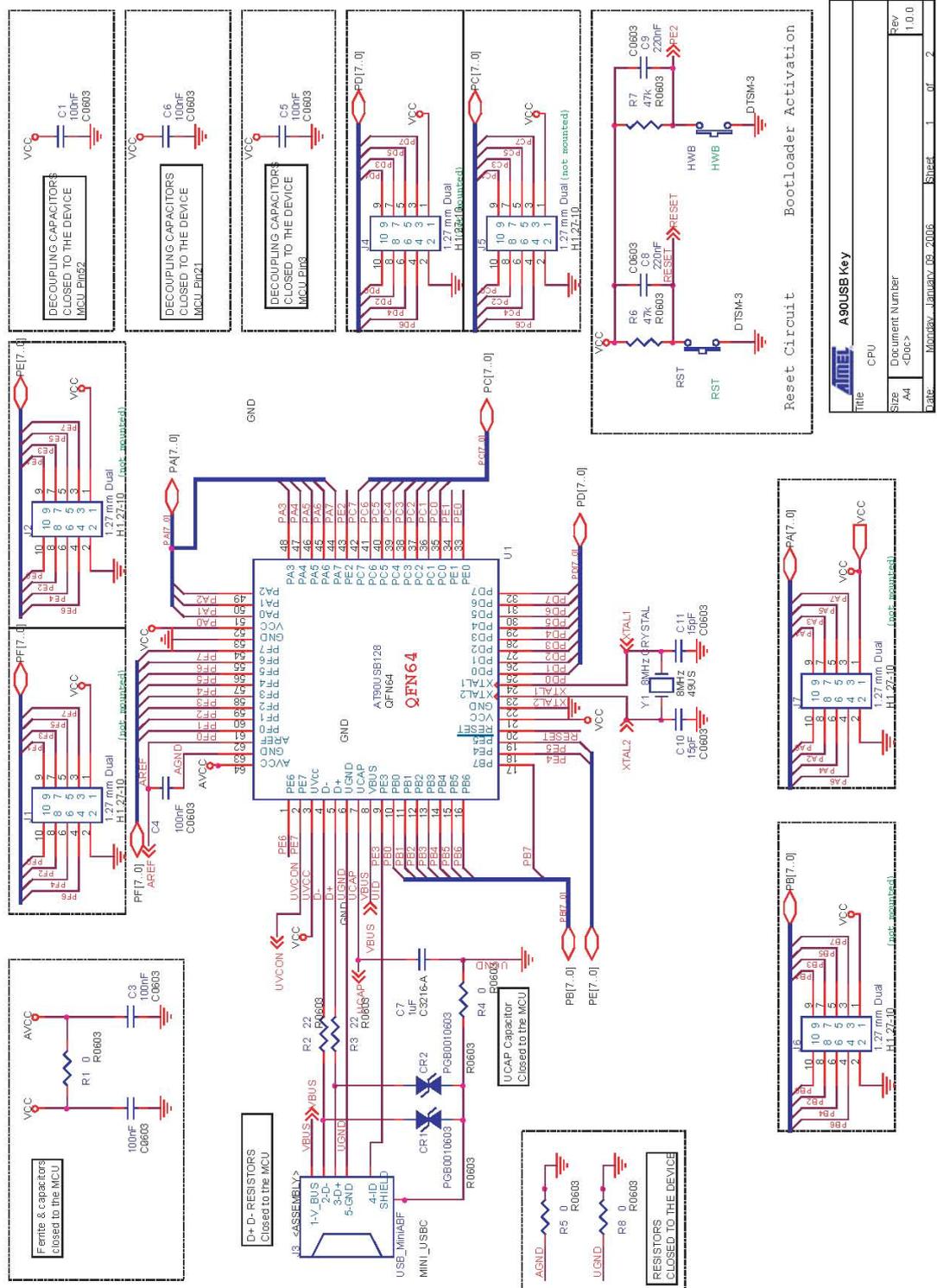
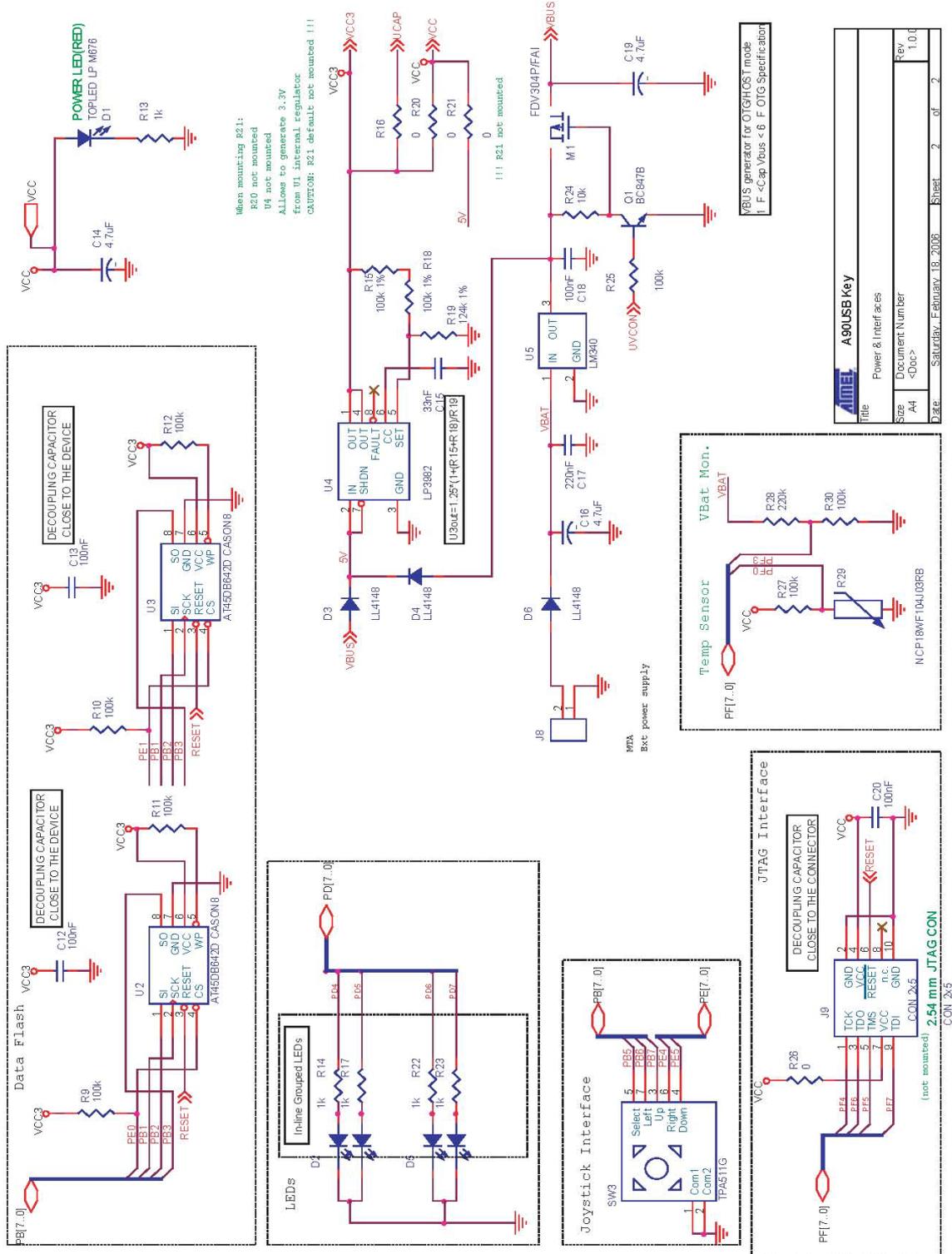


Figure 6-2 . Schematics, 2 of 2



ATMEL A90USB Key	
Title	Power & Interfaces
Size	Document Number
A4	<Doc>
Date	Saturday, February 19, 2006
Sheet	2 of 2
Rev	1.0.0

Table 6-1. Bill of material

Item	Q.ty	Reference	Part	Tech. Characteristics	Package
1	2	CR1,CR2	ESD protection (PGB0010603)		
2	10	C1,C2,C3,C4,C5,C6,C12,C13,C18,C20	100nF	50V-10% Ceramic	CASE 0603
3	1	C7	1uF	10Vmin ±10%	EIA/IECQ 3216
4	3	C8,C9,C17	220nF	50V-10% Ceramic	CASE 0603
5	2	C10,C11	15pF	50V-5% Ceramic	CASE 0603
6	3	C14,C16,C19	4.7uF	10Vmin ±10%	EIA/IECQ 3216
7	1	C15	33nF	50V-5% Ceramic	CASE 0603
8	2	D2,D5	LED BI-COLOUR/ LSGT670	I=10 mA	PLCC-4
9	1	D1	TOPLED/ LPM676-K2M1	I=10 mA	PLCC-2
10	2	D3,D4	DII LL4148-7	i=200mA max	LL-34
11	1	J3	USB mini AB receptacle Surface mount		
12	0	J9	CON 2x5 (2.54mm)	(Not Mounted)	
13	0	J1,J2,J4,J5,J6,J7	1.27 mm Dual header	(Not Mounted)	
14	1	J8	Connector MTA 2 cts right angle		
15	1	M1	FDV304P/FAI		SOT23
16	1	Q1	BC847B	IC peak=200mA	SOT23
17	2	R2,R3	22	1/16W-5% SMD	CASE 0603
18	1	R5	68k	1/16W-5% SMD	CASE 0603
19	2	R6,R7	47k	1/16W-5% SMD	CASE 0603
20	7	R1,R4,R5,R8,R16,R20,R26	0		CASE 0603
21	0	R21	0	(Not Mounted)	CASE 0603
22	1	R28	220k	1/16W-5% SMD	CASE 0603
23	7	R9,R10,R11,R12,R25,R27,R30	100k	1/16W-5% SMD	CASE 0603
24	5	R13,R14,R17,R22,R23	1k	1/16W-5% SMD	CASE 0603
25	2	R24,R28	10k	1/16W-5% SMD	CASE 0603
26	1	R29	NCP18WF104J03RB	100K - β=4250	CASE 0603
27	2	R15,R18	100k 1%	1/16W-1% SMD	CASE 0603
28	1	R19	120k 1%	1/16W-1% SMD	CASE 0603
29	2	SW1,SW2	PUSH-BUTTON / DTSM31N	6x3.5mm - 1.6N	See DS
30	1	SW3	TPA511G 4 ways joystick + center CMS mount		
31	1	U1	AT90USB1287		QFN64
32	2	U2,U3	AT45DB642D		CASON8
33	1	U4	LP3982IMM-ADJ	Vin Max 6V, 300mA	MSOP8
34	1	U6	LM340MP5.0		
35	1	Y1	8MHz CRYSTAL	H=4mm	HC49/4H



## Atmel Corporation

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 487-2600

## Regional Headquarters

### Europe

Atmel Sarl  
Route des Arsenaux 41  
Case Postale 80  
CH-1705 Fribourg  
Switzerland  
Tel: (41) 26-426-5555  
Fax: (41) 26-426-5500

### Asia

Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimshatsui  
East Kowloon  
Hong Kong  
Tel: (852) 2721-9778  
Fax: (852) 2722-1369

### Japan

9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
Tel: (81) 3-3523-3551  
Fax: (81) 3-3523-7581

## Atmel Operations

### Memory

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

### Microcontrollers

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

La Chanterrie  
BP 70602  
44306 Nantes Cedex 3, France  
Tel: (33) 2-40-18-18-18  
Fax: (33) 2-40-18-19-60

### ASIC/ASSP/Smart Cards

Zone Industrielle  
13106 Rousset Cedex, France  
Tel: (33) 4-42-53-60-00  
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park  
Maxwell Building  
East Kilbride G75 0QR, Scotland  
Tel: (44) 1355-803-000  
Fax: (44) 1355-242-743

### RF/Automotive

Theresienstrasse 2  
Postfach 3535  
74025 Heilbronn, Germany  
Tel: (49) 71-31-67-0  
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

### Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
Tel: (33) 4-76-58-30-00  
Fax: (33) 4-76-58-34-80

---

## Literature Requests

[www.atmel.com/literature](http://www.atmel.com/literature)

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© Atmel Corporation 2006. All rights reserved. Atmel®, logo and combinations thereof, are registered trademarks, and Everywhere You Are® are the trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.



Printed on recycled paper.

7627A-AVR-04/06

fxM

*ANEXO VIII*

***Hoja de datos LM2576.***

## LM2576/LM2576HV Series SIMPLE SWITCHER® 3A Step-Down Voltage Regulator

### General Description

The LM2576 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving 3A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5V, 12V, 15V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation and a fixed-frequency oscillator.

The LM2576 series offers a high-efficiency replacement for popular three-terminal linear regulators. It substantially reduces the size of the heat sink, and in some cases no heat sink is required.

A standard series of inductors optimized for use with the LM2576 are available from several different manufacturers. This feature greatly simplifies the design of switch-mode power supplies.

Other features include a guaranteed  $\pm 4\%$  tolerance on output voltage within specified input voltages and output load conditions, and  $\pm 10\%$  on the oscillator frequency. External shutdown is included, featuring 50  $\mu\text{A}$  (typical) standby current. The output switch includes cycle-by-cycle current limiting, as well as thermal shutdown for full protection under fault conditions.

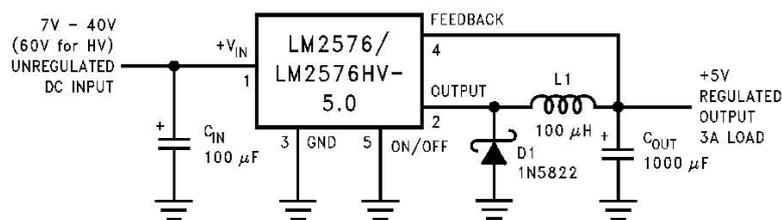
### Features

- 3.3V, 5V, 12V, 15V, and adjustable output versions
- Adjustable version output voltage range, 1.23V to 37V (57V for HV version)  $\pm 4\%$  max over line and load conditions
- Guaranteed 3A output current
- Wide input voltage range, 40V up to 60V for HV version
- Requires only 4 external components
- 52 kHz fixed frequency internal oscillator
- TTL shutdown capability, low power standby mode
- High efficiency
- Uses readily available standard inductors
- Thermal shutdown and current limit protection
- P+ Product Enhancement tested

### Applications

- Simple high-efficiency step-down (buck) regulator
- Efficient pre-regulator for linear regulators
- On-card switching regulators
- Positive to negative converter (Buck-Boost)

### Typical Application (Fixed Output Voltage Versions)

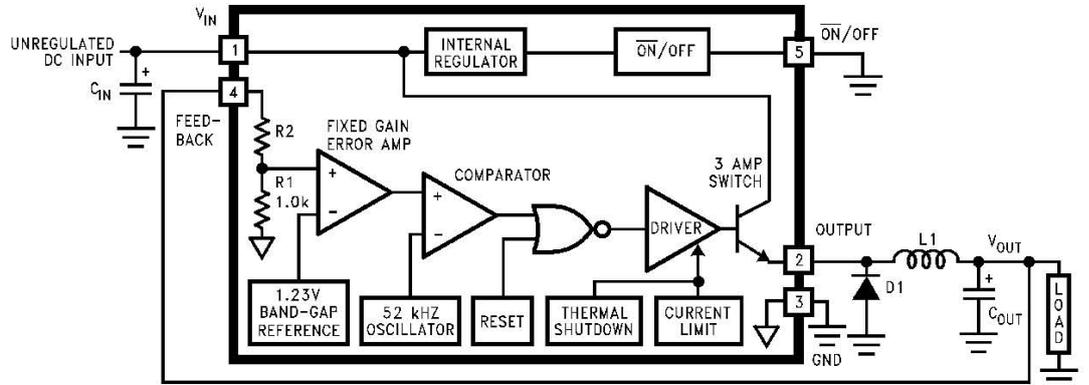


01147601

FIGURE 1.

SIMPLE SWITCHER® is a registered trademark of National Semiconductor Corporation.

### Block Diagram



01147602

3.3V, R2 = 1.7k  
 5V, R2 = 3.1k  
 12V, R2 = 8.84k  
 15V, R2 = 11.3k  
 For ADJ. Version  
 R1 = Open, R2 = 0Ω  
 Patent Pending

### Ordering Information

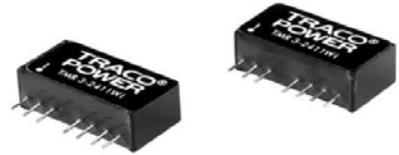
Temperature Range	Output Voltage					NS Package Number	Package Type
	3.3	5.0	12	15	ADJ		
-40°C ≤ T <sub>A</sub> ≤ 125°C	LM2576HVS-3.3	LM2576HVS-5.0	LM2576HVS-12	LM2576HVS-15	LM2576HVS-ADJ	TS5B	TO-263
	LM2576S-3.3	LM2576S-5.0	LM2576S-12	LM2576S-15	LM2576S-ADJ		
	LM2576HVSX-3.3	LM2576HVSX-5.0	LM2576HVSX-12	LM2576HVSX-15	LM2576HVSX-ADJ	Tape & Reel	
	LM2576SX-3.3	LM2576SX-5.0	LM2576SX-12	LM2576SX-15	LM2576SX-ADJ		
	LM2576HVT-3.3	LM2576HVT-5.0	LM2576HVT-12	LM2576HVT-15	LM2576HVT-ADJ	T05A	TO-220
	LM2576T-3.3	LM2576T-5.0	LM2576T-12	LM2576T-15	LM2576T-ADJ		
	LM2576HVT-3.3	LM2576HVT-5.0	LM2576HVT-12	LM2576HVT-15	LM2576HVT-ADJ	T05D	
	Flow LB03	Flow LB03	Flow LB03	Flow LB03	Flow LB03		
LM2576T-3.3	LM2576T-5.0	LM2576T-12	LM2576T-15	LM2576T-ADJ			
Flow LB03	Flow LB03	Flow LB03	Flow LB03	Flow LB03			

*ANEXO IX*

***Hoja de datos del módulo TMR-3WI.***

#### Features

- ◆ Highest power density in SIP package
- ◆ Ultra wide 4:1 input range
- ◆ Small footprint: 21.8 x 9.2 mm
- ◆ Temperature range  $-40^{\circ}$  to  $+85^{\circ}\text{C}$
- ◆ High efficiency up to 81%
- ◆ Excellent load and line regulation
- ◆ Short-circuit protection
- ◆ I/O isolation 1500 VDC
- ◆ Remote On/Off control
- ◆ 3-year product warranty



The TMR-3WI series is a new family of isolated 3W DC/DC converters with regulated output, featuring ultra-wide 4:1 input voltage range. The product comes in an ultra-compact SIP plastic package with a small footprint occupying only 2.0 cm<sup>2</sup> (0.3 square in.) of board space. An excellent efficiency allows  $-40^{\circ}$  to  $+85^{\circ}\text{C}$  operation temperatures.

Further features include remote On/Off control and continuous short circuit protection. The very compact dimensions of these converters make them an ideal solution for many space critical applications in battery-powered equipment and instrumentation.

Models				
Order code	Input voltage	Output voltage	Output current max.	Efficiency typ.
TMR 3-1210WI	4.5 – 18 VDC (12 VDC nominal)	3.3 VDC	700 mA	74 %
TMR 3-1211WI		5 VDC	600 mA	78 %
TMR 3-1212WI		12 VDC	250 mA	80 %
TMR 3-1213WI		15 VDC	200 mA	80 %
TMR 3-1221WI		$\pm 5$ VDC	$\pm 300$ mA	80 %
TMR 3-1222WI		$\pm 12$ VDC	$\pm 125$ mA	80 %
TMR 3-1223WI		$\pm 15$ VDC	$\pm 100$ mA	80 %
TMR 3-2410WI	9 – 36 VDC (24 VDC nominal)	3.3 VDC	700 mA	75 %
TMR 3-2411WI		5 VDC	600 mA	80 %
TMR 3-2412WI		12 VDC	250 mA	81 %
TMR 3-2413WI		15 VDC	200 mA	81 %
TMR 3-2421WI		$\pm 5$ VDC	$\pm 300$ mA	79 %
TMR 3-2422WI		$\pm 12$ VDC	$\pm 125$ mA	80 %
TMR 3-2423WI		$\pm 15$ VDC	$\pm 100$ mA	81 %
TMR 3-4810WI	18 – 75 VDC (48 VDC nominal)	3.3 VDC	700 mA	74 %
TMR 3-4811WI		5 VDC	600 mA	79 %
TMR 3-4812WI		12 VDC	250 mA	79 %
TMR 3-4813WI		15 VDC	200 mA	79 %
TMR 3-4821WI		$\pm 5$ VDC	$\pm 300$ mA	79 %
TMR 3-4822WI		$\pm 12$ VDC	$\pm 125$ mA	79 %
TMR 3-4823WI		$\pm 15$ VDC	$\pm 100$ mA	80 %

**Input Specifications**

Input current at full load	12 Vin models: 340 mA max. 24 Vin models: 170 mA max. 48 Vin models: 85 mA max.
Input current at no load	12 Vin models: 40 mA max. 24 Vin models: 25 mA typ. 48 Vin models: 15 mA typ.
Surge voltage (100 msec. max.)	12 Vin models: 25 V max. 24 Vin models: 50 V max. 48 Vin models: 100 V max.
Input filter	inernal capacitor
ESD (electrostatic discharge)	EN 61000-4-2, air ±8 kV, contact ±6 kV, perf. criteria A
Radiated immunity	EN 61000-4-3, 10 V/m, perf. criteria A
Fast transient / Surge	EN 61000-4-4, ± 2 kV, perf. criteria A EN 61000-4-5, ± 1 kV perf. criteria A With external input capacitor e.g. Nippon chemi-con KY 100 µF, 100 V, ESR 110 mOhm
Conducted immunity	EN 61000-4-6, 10 Vrms, perf. criteria A

**Output Specifications**

Voltage set accuracy	±1 % max
Regulation	– Input variation Vin min. to Vin max. 0.2 % max. – Load variation 0 – 100% single output models: 1.0 % max. dual output models: 1.0 % max. balanced load – Load cross regulation 25/100% 5.0 % max. (dual output models)
Minimum load	not required
Temperature coefficient	0.02 %/K
Ripple and noise (20 MHz Bandwidth)	30 mVpk-pk max.
Start up time (constant resistive load)	– Power On 30 ms typ. – Remote On 30 ms typ.
Transient response setting time (25% load step change)	250 µs typ.
Short circuit protection	continuous, automatic recovery
Capacitive load	3.3 VDC models: 1'760 µF max. 5 VDC models: 1'000 µF max. 12 VDC models: 170 µF max. 15 VDC models: 110 µF max. ±5 VDC models: ±470 µF max. ±12 VDC models: ±100 µF max. ±15 VDC models: ±47 µF max.

**General Specifications**

Temperature ranges	– Operating –40°C to +85°C – Case temperature +100°C max. – Storage –55°C to +125°C
Load derating	3.3 %/K above 70°C
Humidity (non condensing)	95 % rel. H max.
Reliability, calculated MTBF (MIL-HDBK-217F, @ 25°C, ground benign)	>1.7 Mio h

All specifications valid at nominal input voltage, full load and +25°C after warm-up time unless otherwise stated.

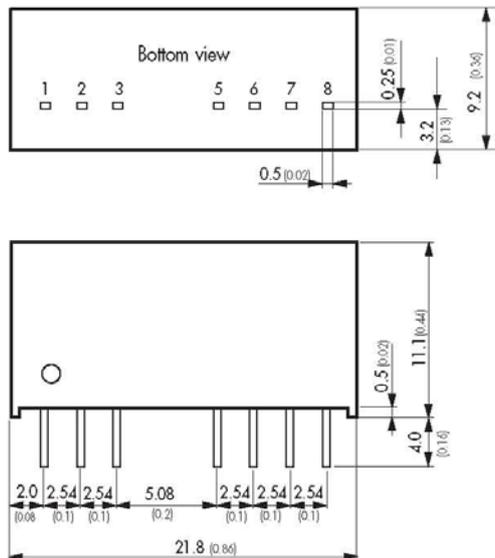
**General Specifications**

Isolation voltage [60 sec]	- Input/Output	1'500 VDC
Isolation capacity	- Input/Output	200 pF max.
Isolation resistance	- Input/Output [500 VDC]	>1 GOhm
Switching frequency		100 kHz (PWM)
Remote On/Off	- On: - Off: - Off stand by input current	open or high impedance 2...4 mA to applied via 1 kOhm resistor 2.5 mA max.
Vibration and thermal shock		MIL-STD-810E
Safety standards		UL /cUL 60950-1, IEC/EN 60950-1
Safety approvals	- UL/cUL	pending
Environmental compliance	- Reach - RoHS	<a href="http://www.tracopower.com/products/tmr3wi-reach.pdf">www.tracopower.com/products/tmr3wi-reach.pdf</a> RoHS Directive 2002/95/EC
Application notes		<a href="http://www.tracopower.com/products/tmr3wi-application.pdf">www.tracopower.com/products/tmr3wi-application.pdf</a>

**Physical Specifications**

Casing material	non-conductive plastic
Potting material	silicon, UL 94V-0 rated
Weight	4.8 g (0.17oz)

**Outline Dimensions**



Pin-Out		
Pin	Single	Dual
1	-Vin (GND)	-Vin (GND)
2	+Vin (Vcc)	+Vin (Vcc)
3	Remote On/Off	Remote On/Off
5	No con.	No con.
6	+Vout	+Vout
7	-Vout	Common
8	No con.	-Vout

Dimensions in [mm], ( ) = Inch  
Pin dimension tolerances 0.1 (0.004)  
Pin pitch tolerances: ±0.25 (±0.01)  
Tolerances: ±0.5 (±0.02)

Specifications can be changed any time without notice.

## BIBLIOGRAFIA.

- [1] Ariel Palazzesi, Historia del GPS: Cómo el mundo dejó de perderse, <http://www.neoteo.com/historia-del-gps-como-el-mundo-dejo-de-perderse.neo>; **Fecha de consulta:** 20 de septiembre de 2010.
- [2] Mio Technology UK Ltd., El GPS a fondo, [http://eu.mio.com/es\\_es/sistema-posicionamiento-global\\_4978.htm](http://eu.mio.com/es_es/sistema-posicionamiento-global_4978.htm), **Fecha de consulta:** 20 de septiembre 2010.
- [3] Jesús Adolfo Anaya Acevedo, Programa Académico, [http://www.unalmed.edu.co/~janaya/JAnayaCont05\\_Clase04.html](http://www.unalmed.edu.co/~janaya/JAnayaCont05_Clase04.html), **Fecha de consulta:** 21 de septiembre de 2010
- [4] Laipac Technology Inc., G35L Data Lgger, [http://www.laipac.com/g35l\\_datalogger.htm](http://www.laipac.com/g35l_datalogger.htm), **Fecha de consulta:** 21 de septiembre del 2010.
- [5] HOLUX Technology Inc., Holux M-1000C Bluetooth GPS Data Logger, [http://www.holux.com/JCore/en/products/products\\_content.jsp?pno=351](http://www.holux.com/JCore/en/products/products_content.jsp?pno=351), **Fecha de consulta:** 21 de septiembre del 2010.

- [6] Wintec Co., Ltd., Wintec WPL-2000 Auto-Show,  
<http://www.wintec.com.tw/en/faq/manual/WPL-2000.pdf>, **Fecha de consulta:** 21 de septiembre del 2010.
- [7] Wikimedia Foundation, Universal Serial Bus,  
[http://es.wikipedia.org/wiki/Universal\\_Serial\\_Bus](http://es.wikipedia.org/wiki/Universal_Serial_Bus), **Fecha de consulta:** 22 de septiembre de 2010
- [8] USB Implementers Forum, Inc., Universal Serial Bus Mass Storage Class Control/Bulk/Interrupt (CBI) Transport,  
[http://www.usb.org/developers/devclass\\_docs/usb\\_msc\\_cbi\\_1.1.pdf](http://www.usb.org/developers/devclass_docs/usb_msc_cbi_1.1.pdf), **Fecha de consulta:** 22 de septiembre de 2010.
- [9] Intel Corporation, Tipos de Conectores USB,  
<http://www.intel.com/support/sp/motherboards/desktop/sb/CS-023466.htm>,  
**Fecha de consulta:** 22 de septiembre del 2010.
- [10] USB Implementers Forum, Inc., USB On-The-Go and Embedded Host,  
<http://www.usb.org/developers/onthego/>, **Fecha de consulta:** 22 de septiembre de 2010.
- [11] Wikimedia Foundation, Mass Storage Device Class,  
[http://es.wikipedia.org/wiki/USB\\_mass\\_storage\\_device\\_class](http://es.wikipedia.org/wiki/USB_mass_storage_device_class), **Fecha de consulta:** 22 de Septiembre de 2010.

- [12] USB Implementers Forum, Inc., Universal Serial Bus, Mass Storage Class, Bulk-Only Transport, [http://www.usb.org/developers/devclass\\_docs/usbmassbulk\\_10.pdf](http://www.usb.org/developers/devclass_docs/usbmassbulk_10.pdf), **Fecha de consulta:** 22 de septiembre de 2010.
- [13] u-blox AG, ANTARIS®4 GPS Modules System Integration Manual (SIM) (incl. Reference Design), [http://www.u-blox.com/images/downloads/Product\\_Docs/ANTARIS4\\_Modules\\_SIM%28GPS.G4-MS4-05007%29.pdf](http://www.u-blox.com/images/downloads/Product_Docs/ANTARIS4_Modules_SIM%28GPS.G4-MS4-05007%29.pdf), **Fecha de consulta:** 23 de septiembre de 2010
- [14] u-blox AG, ANTARIS Protocol Specification(GPS.G3-X-03002), [http://www.u-blox.com/images/downloads/Product\\_Docs/ANTARIS\\_Protocol\\_Specification%28GPS.G3-X-03002%29.zip](http://www.u-blox.com/images/downloads/Product_Docs/ANTARIS_Protocol_Specification%28GPS.G3-X-03002%29.zip), **Fecha de consulta:** 23 de septiembre de 2010
- [15] u-blox AG, GPS  $\mu$ -Blox RCB 4H, [http://www.alphamicro.net/resources/u-blox/RCB-4H\\_Data\\_Sheet\(GPS.G4-MS4-06034\).pdf](http://www.alphamicro.net/resources/u-blox/RCB-4H_Data_Sheet(GPS.G4-MS4-06034).pdf), **Fecha de consulta:** 23 de septiembre de 2010
- [16] Atmel Corporation, Atmel Products - AVR Solutions - AT90USBKEY, [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=3879](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3879) , **Fecha de consulta:** 23 de septiembre de 2010.

- [17] Atmel Corporation, AT90USBKEY, Hardware User Guide,  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc7627.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc7627.pdf), **Fecha de consulta:** 23 de septiembre de 2010
- [18] Atmel Corporation, 8-bit AVR Microcontroller with 64/128K Bytes of ISP Flash and USB Controller , AT90USB646 AT90USB647 AT90USB1286 AT90USB1287,  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc7593.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc7593.pdf), **Fecha de consulta:** 23 de septiembre de 2010.
- [19] Atmel Corporation, AVR287:USB Host HID and Mass Storage Demonstration,  
[www.atmel.com/dyn/resources/prod\\_documents/doc8229.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8229.pdf) , **Fecha de consulta:** 23 de septiembre de 2010.
- [20] Atmel Corporation, AVR115: Data Logging with Atmel File System on ATmega32U4,  
[www.atmel.com/dyn/resources/prod\\_documents/doc8202.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8202.pdf) **Fecha de consulta:** 23 de septiembre de 2010.
- [21] Atmel Corporation, Atmel Products - AVR Solutions - AVR Studio 4,  
<http://www.atmel.com/avrstudio>, **Fecha de consulta:** 23 de septiembre de 2010.

- [22] Eric Weddington, WinAVR™ ,<http://winavr.sourceforge.net/> , **Fecha de consulta:** 23 de Septiembre de 2010.
- [23] Atmel Corporation, AVR JTAGICE mkII,  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc2489.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2489.pdf), **Fecha de consulta:** 24 de septiembre de 2010
- [24] Traco Electronic AG., TMR 3-WI Series Application Note,  
<http://www.tracopower.com/products/tmr3wi-application.pdf>, **Fecha de consulta:** 24 de septiembre de 2010.
- [25] National Semiconductor, Regulador de Voltaje LM2576ADJ,  
<http://www.national.com/ds/LM/LM2576.pdf>, **Fecha de consulta:** 24 de septiembre de 2010
- [26] TopoGrafix, GPX: the GPS Exchange Format,  
<http://www.topografix.com/gpx.asp>, **Fecha de consulta:** 04 de octubre de 2010.
- [27] Wikimedia Foundation, GPS eXchange Format.  
[http://en.wikipedia.org/wiki/GPS\\_eXchange\\_Format](http://en.wikipedia.org/wiki/GPS_eXchange_Format), **Fecha de consulta:** 04 de octubre de 2010.
- [28] Christian Cercado Suarez y Victor Touriz Plua, Datalogger compacto, con tiempo real, para almacenaje de grandes cantidades de información provenientes de sensores externos en una memoria USB, con capacidad de

comunicación serial. Fuente de energía: 4 pilas recargables AA, Facultad de Ingeniería en Electricidad y Computación, ESPOL, 2010.

- [29] Martin Avilés Avilés y Juan Bajaña Gutiérrez, Aplicación de Módulo Receptor de GPS para el posicionamiento de robots manejados a control remoto con capacidad de comunicación serial a Datalogger e Interfaz Gráfica, Facultad de Ingeniería en Electricidad y Computación, ESPOL, 2010.