

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

Manejo de un GPS con NIOS II

TESINA DE SEMINARIO

Previa a la obtención del Título de:

INGENIERO EN ELECTRONICA Y TELECOMUNICACIONES

Presentado por:

Diego Fernando Jácome Ruiz

Guayaquil – Ecuador

Año 2012

AGRADECIMIENTO

A la divinidad.

DEDICATORIAS

Para Ana, lamparita dulce de mi existencia.

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Ponguillo

PROFESOR DEL SEMINARIO DE GRADUACIÓN

Ing. María Antonieta Álvarez

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este trabajo, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”. (Reglamento de exámenes y títulos profesionales de la ESPOL)

Diego Fernando Jácome Ruiz

RESUMEN

Este proyecto pretende dar a conocer e implementar el manejo de un módulo GPS, a través del protocolo RS232, con un FPGA Cyclone II de Altera.

El presente trabajo se lo ha estructurado en 4 capítulos.

En el capítulo 1 se presentan los objetivos, identificación del problema y la metodología a seguir en el proyecto.

El capítulo 2 incluye el fundamento teórico sobre el que se basa el desarrollo del trabajo. Se revisan conceptos fundamentales de procesadores embebidos configurables y FPGAs. Así mismo incluye información acerca del funcionamiento del Sistema de Posicionamiento Global (GPS por sus siglas en inglés) y de todo lo necesario para tener claras las limitaciones y alcances de la aplicación. Se considera en este capítulo información relevante de la tecnología de almacenamiento de información basado en tarjetas SD que son utilizadas como parte del sistema a presentar.

En el capítulo 3 se describe cada una de las partes que componen el proyecto, separados en hardware y software a implementar. El hardware está comprendido principalmente por el FPGA que contiene a todos los drivers que se utilizarán más adelante con el resto del hardware externo al mismo. Se presenta además un detalle

del funcionamiento del modulo GPS escogido para realizar esta aplicación, en donde podemos ver detalles del protocolo de comunicación utilizado así como su funcionamiento en las diferentes condiciones que se utilizará el sistema, y también se detallan características de la tecnología de almacenamiento en tarjetas de memoria. El software del proyecto se lo desarrolla en lenguaje C para el núcleo de procesador NIOS II que está embebido en el FPGA configurado anteriormente.

Finalmente, en el capítulo 4, se hace un exhaustivo análisis del funcionamiento del sistema en condiciones reales donde se verifica su utilidad y limitaciones.

INDICE GENERAL

RESUMEN	vi
INDICE GENERAL	viii
ABREVIATURAS	xi
INDICE DE FIGURAS	xiii
INDICE DE TABLAS	xv
INTRODUCCION	xiv
CAPITULO I	1
1. GENERALIDADES	1
1.1 OBJETIVOS	1
1.1.1 Objetivos Generales	1
1.1.2 Objetivos Específicos	1
1.2 IDENTIFICACIÓN DEL PROBLEMA	2
1.3 METODOLOGÍA	2
CAPITULO II	4
2. MARCO TEÓRICO	4

2.1	FPGA.....	4
2.1.1	Definición y características	4
2.1.2	FPGA Cyclone II.....	6
2.1.3	NIOS II.....	7
2.1.4	Tarjetas de Desarrollo de Altera.....	8
2.1.5	Aplicaciones.....	10
2.2	GPS.....	11
2.2.1	Sistema.....	11
2.2.2	Módulos	14
2.2.3	Formato NMEA.....	15
2.3	Tarjetas de Memoria SD	17
2.3.1	Tipos.....	18
2.3.2	Características Tarjetas SD [9].....	20
CAPITULO III.....		23
3.	DISEÑO E IMPLEMENTACIÓN	23
3.1	Hardware.....	24
3.1.1	Componentes en la tarjeta de desarrollo	24
3.1.2	FPGA.....	26

3.1.3	Receptor GPS.....	31
3.1.4	Tarjeta de memoria SD	35
3.2	Programación del Procesador	36
3.2.1	Entorno de programación	36
CAPITULO 4.....		40
4.	PRUEBAS Y RESULTADOS.....	40
4.1	Almacenamiento de datos GPS.....	40
4.2	Visualización en programa Google Earth.....	42
CONCLUSIONES.....		46
RECOMENDACIONES.....		48
ANEXOS		49
ANEXO A.....		50
ANEXO B.....		55
ANEXO C.....		57
REFERENCIAS BIBLIOGRÁFICAS.....		59

ABREVIATURAS

FPGA	Field Programmable Gate Array
DE2	Development and Education Board
RS-232	Recommended Standard 232
Km/h	Kilómetros por hora
E/S	Entrada y Salida
GPIO	General Purpose Input/Output
USB	Universal Serial Bus
LCD	Liquid Crystal Display
JTAG	Joint Test Action Group
UART	Universal Asynchronous Receiver-Transmitter
ASIC	Application Specific Integrated Circuit
SOPC	System on a Programmable Chip

Mbits	Megabits
ADC	Analog-to-Digital Converter
TTL	Transistor-Transistor Logic
CMOS	Complementary metal-oxide-semiconductor
LED	Light-Emitting Diode
GPS	Global Position System
SD	Secure Data
KML	Keyhole Markup Language
UTC	Universal Time Coordinated

INDICE DE FIGURAS

Figura 1 Diagrama de bloques de sistema NIOS II.....	7
Figura 2 Tarjeta de Desarrollo DE2	8
Figura 3 Sistema de Satélites GPS.....	11
Figura 4 Módulo GPS FV-M5 marca SANAV	14
Figura 5 Tarjetas de Memoria SD	17
Figura 6 Diagrama de bloques General del sistema	24
Figura 7 Componentes utilizados de la Tarjeta DE2.....	25
Figura 8 Herramienta SoPC Builder	25
Figura 9 Entorno de Programación.....	27
Figura 10 Configuración de Puerto Serie	28
Figura 11 Programador de Quartus II	29
Figura 12 Diagrama General del Sistema.....	30
Figura 13 Receptor GPS SANAV GM-158	31

Figura 14 Tarjeta de memoria micro SD marca Sandisk	35
Figura 15 Entorno de programación Eclipse para NIOS II.....	37
Figura 16 Diagrama de Flujo de programa.....	38
Figura 17 Prototipo instalado en un vehículo	40
Figura 18 Archivo de texto almacenado en la memoria SD.....	41
Figura 19 Página web GSP Visualizer	42
Figura 20 Opciones a elegir en página GPS Visualizer	43
Figura 21 Archivo generado por página GPS Visualizer	44
Figura 22 Presentación del recorrido en Google Earth	45

INDICE DE TABLAS

Tabla I Características de FPGA EP2C35	6
Tabla II Campos de sentencia NMEA GPRMC	16
Tabla III Tipos de tarjetas de memoria SD	19
Tabla IV Canal de comunicación Tarjeta SD	22

INTRODUCCION

En los últimos años, la humanidad ha sido testigo de la impresionante velocidad a la que la tecnología va evolucionando, creando cada vez más y mejores dispositivos pensando en resolver problemas o simplemente en el confort de las personas que pueden acceder a dicha tecnología. Esto se evidencia en los dispositivos electrónicos de uso general como teléfonos móviles, tabletas y computadores. En esta línea todos los nuevos desarrollos están basados en tres ejes fundamentales: diseño, rapidez y funcionalidad. Todos basados en procesadores más versátiles, pequeños y rápidos.

Aquí es donde los denominados FPGAs (Field Programmable Gate Array) entran en acción, permitiendo a los usuarios programar tanto el hardware como el software de acuerdo a necesidades específicas.

Las soluciones basadas en esta tecnología son apreciadas por poseer una gran versatilidad a la hora de actualizar soluciones sin tener que cambiar de circuito integrado central, es decir, muy útiles cuando de prototipos se habla.

En este punto, al hacer aplicaciones con este tipo de circuitos integrados, existen muchas posibilidades de configurar el hardware de acuerdo a necesidades muy puntuales.

Hallar una aplicación con un modulo de GPS resulta bastante sencilla, dado lo extendido del uso de los mismo en diversos ámbitos como seguridad, logística y operaciones remotas. En este sentido, el presente trabajo desarrolla una aplicación sencilla pero eficaz para control y logística de recorridos de bienes o personas.

CAPITULO I

1. GENERALIDADES

Se inicia el documento identificando objetivos, problema y finalmente la metodología que se aplicará a lo largo del desarrollo del presente proyecto.

1.1 OBJETIVOS

1.1.1 Objetivos Generales

Aplicar los conocimientos adquiridos para la implementación de un hardware que integre un FPGA Cyclone II de Altera con un módulo receptor de GPS, y lograr una aplicación práctica que incluso servirá de base para desarrollos más complejos.

1.1.2 Objetivos Específicos

- Investigar la forma de usar el driver de comunicación serial del FPGA Cyclone II, para poder recibir información serial desde un módulo receptor de GPS.
- Interpretar protocolos de comunicación de receptores GPS, así procesar dicha información.

- Almacenar la información del Receptor GPS en una memoria externa.
- Interpretar los datos almacenados para poder ser visualizados en un mapa georeferenciado.

1.2 IDENTIFICACIÓN DEL PROBLEMA

Actualmente existe cada vez una mayor demanda por dispositivos para poder controlar la ubicación de todo tipo de bienes, vehículos y personas; que, dependiendo de la necesidad, pueden ser activos, en donde la información de posición es transmitida por algún medio, o pueden ser pasivos en donde la información es almacenada y posteriormente analizada.

Por estas razones se requieren de dispositivos cada vez más pequeños, inteligentes y con menor consumo de energía.

1.3 METODOLOGÍA

Este proyecto parte de un receptor GPS serial de marca SANNNAV y de una tarjeta de desarrollo DE2 con un FPGA Cyclone II. El almacenamiento de información se lo hace en una tarjeta de memoria SD.

Se aprovecha el hardware que viene incluido en la tarjeta DE2, específicamente el convertidor de niveles para poder comunicarse con el receptor GPS a través del protocolo serial RS-232, el conector DB9, la ranura para la tarjeta SD y el LCD de 2x16.

Se configura el hardware en el FPGA, de tal manera que se tenga, además de los módulos básicos, los módulos necesarios para manejar el puerto serial y la memoria SD.

Posteriormente se programa el procesador Nios II para que pueda interpretar la información proveniente, cada segundo, del receptor GPS y almacenar, cada 10 segundos, la información en la tarjeta de memoria SD.

Finalmente la información almacenada en la SD puede ser visualizada con la herramienta gratuita Google Earth, convirtiendo el archivo de texto almacenado al formato KML con una herramienta web.

CAPITULO II

2. MARCO TEÓRICO

En este capítulo se da a conocer el fundamento teórico y la generalidad de cada uno de los componentes del proyecto. Se presenta información acerca de FPGA's, el sistema GPS y la tecnología de almacenamiento de información en tarjetas de memoria SD.

2.1 FPGA

2.1.1 Definición y características

FPGA, que en castellano se lo podría traducir como Conjunto de Puertas Programables en Campo, es un tipo de circuito integrado que se utiliza ampliamente en circuitos electrónicos digitales. Es un dispositivo semiconductor que contiene interconexiones y lógica programables. Los componentes lógicos programables pueden ser configurados de acuerdo a necesidades específicas incluso si el producto ya ha sido instalado en campo, de ahí su nombre. Esta característica hace a los FPGAs muy útiles para ciertas aplicaciones.

Los FPGAs han evolucionado en el tiempo, pasando de usar entradas y salidas con lógica en interconexión programables, a consistir de varias combinaciones de Memorias SRAM configurables, transceptores de alta velocidad, entradas y salidas de alta velocidad, bloques lógicos, que incluyen elementos de memoria, y enrutamiento. Se pueden configurar desde puertas lógicas simples hasta complejas funciones combinatorias.

Con estos antecedentes fácilmente se puede determinar las ventajas de los FPGAs sobre otros circuitos integrados de aplicación específica:

- Menor tiempo para desarrollar prototipos
- Menor tiempo de comercialización
- Capacidad de ser reprogramados en campo
- Menor costo de investigación

Los FPGAs continúan evolucionando combinándolos con mejores procesadores, convirtiéndolos en sistemas en un solo chip. Varios fabricantes como Xilinx, Atmel y Altera fabrican este tipo de tecnología híbrida [1].

2.1.2 FPGA Cyclone II

La serie de FPGAs Cyclone del fabricante Altera está diseñada para cumplir con un bajo consumo de energía, necesidades de diseño sensibles al costo, permitiendo comercializar productos más rápido, mayor integración, mayor rendimiento. La serie Cyclone va por su quinta generación. En el presente trabajo se usará la segunda generación, denominada Cyclone II, la misma que fue introducida en el 2004, con una tecnología de proceso de 90nm y recomendado para nuevos diseños [11]. El Dispositivo Cyclone II utilizado es el EP2C35 que posee las siguientes características (ver Tabla I):

Característica	EP2C35
Elementos lógicos	33216
Bloques M4K RAM	105
Bits RAM Totales	483840
Multiplicadores Integrados	35
PLLs	4
Pines de E/S máximos	475

Tabla I Características de FPGA EP2C35

2.1.3 NIOS II

Nios II es un núcleo de procesador del fabricante Altera que combinado con un FPGA componen un sistema en un chip (SoC por sus siglas en Inglés). EL Procesador Nios II es un procesador de propósito general RISC con las siguientes características:

- Conjunto de instrucciones completas de 32 bits
- 32 registros de propósito general
- 32 fuentes de interrupción
- Desempeño de hasta 250 DMIPS
- Entorno de desarrollo basado en C/C++

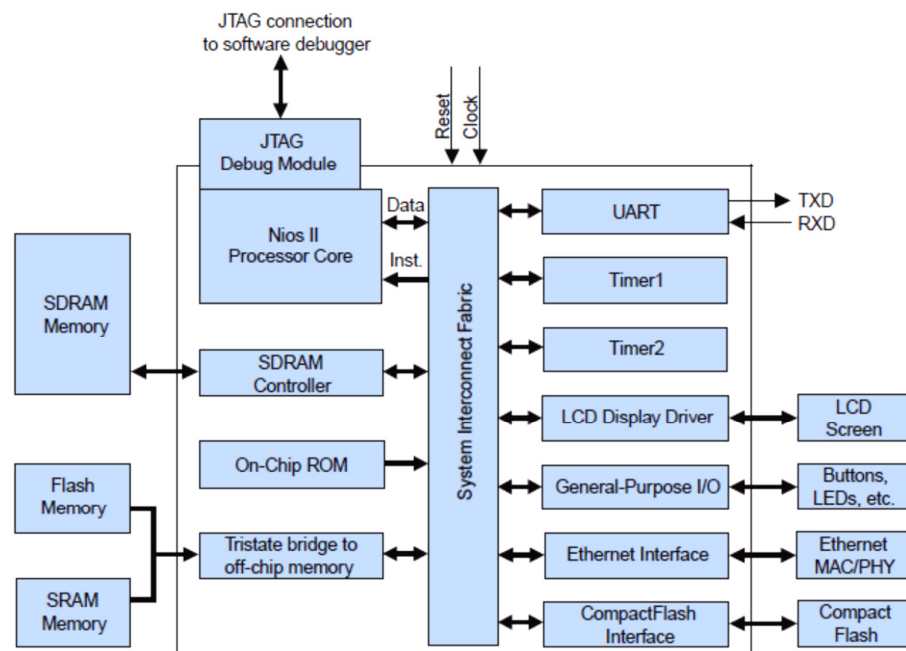


Figura 1 Diagrama de bloques de sistema NIOS II [12]

Un sistema de procesador Nios II es un microcontrolador que incluye un procesador y una combinación de periféricos y memoria en un solo chip. El sistema consiste en un núcleo de procesador Nios II, un conjunto de periféricos integrados, memoria interna, e interfaces para memoria externa, todo implementado en un solo dispositivo de Altera, como se muestra en la Figura 1[12].

2.1.4 Tarjetas de Desarrollo de Altera

El propósito de las tarjetas de desarrollo de Altera es proveer el medio ideal para hacer diseños de prototipos avanzados con multimedia, almacenamiento y redes. El modelo usado para el presente trabajo es el DE2 del proveedor TERCASIC (ver figura 2).

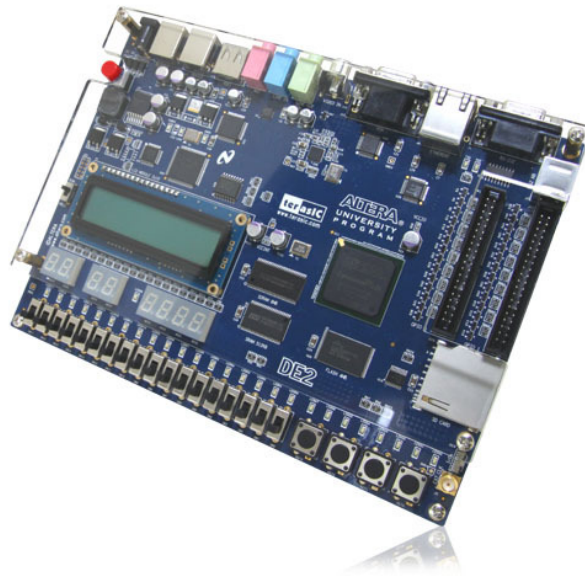


Figura 2 Tarjeta de Desarrollo DE2 [10]

La tarjeta DE2 provee a los usuarios de muchas características para realizar numerosos proyectos multimedia. La selección de componentes fue hecha de acuerdo a los diseños de productos más populares tales como DVD, VCD y reproductores de MP3. La plataforma DE2 permite a los usuarios adquirir rápidamente todas las destrezas para diseñar proyectos multimedia reales para la industria [3]. Incluye:

- FPGA Altera Cyclone II 2C35 con 35000 LEs
- Dispositivos Altera de configuración de serie (EPCS16) para Cyclone II 2C35
- USB Blaster integrado para programación
- Modo JTAG y modo AS compatibles
- 8MB de SDRAM
- 512Kbytes (256 X16) SRAM
- 4 MB de memoria flash
- Ranura para tarjeta SD
- 4 interruptores de pulsador
- 18 interruptores DPDT
- 9 LEDs verdes
- 18 LEDs rojos

- Módulo LCD 16 x 2
- Oscilador de 50-MHz y 27 MHz (del decodificador de TV) para fuentes de reloj
- Audio CODEC de 24-bit con calidad de CD con línea de entrada, línea de salida y conector para micrófono
- VGA DAC (10-bit de alta velocidad DAC triples) con conector de salida VGA
- Decodificador de TV (NTSC / PAL)
- Controlador 10/100 Ethernet con conector
- Controlador USB Host/Slave con conectores USB de tipo A y tipoB
- Transceptor RS-232 y conector de 9 pines
- Conector PS/2 para ratón / teclado
- Transceptor IrDA

2.1.5 Aplicaciones

Un sinnúmero de aplicaciones específicas pueden ser implementadas con un FPGA. Son comúnmente utilizados en DSP (procesamiento digital de señales), radio definido por software, sistemas aeroespaciales y de defensa, prototipos de ASICs, sistemas de imágenes para medicina, sistemas de visión para computadoras,

reconocimiento de voz, bioinformática, emulación de hardware de computadora, etc. Su uso en otras áreas es cada vez mayor, sobre todo en aquellas aplicaciones que requieren un alto grado de paralelismo.

2.2 GPS

2.2.1 Sistema

GPS son las siglas que en castellano se traduce como Sistema de Posicionamiento Global, constituye un sistema de radionavegación por satélite desarrollado y operado por el Departamento de Defensa de EE.UU. El GPS permite a usuarios en tierra, mar, y aire determinar su posición tridimensional, velocidad y tiempo las 24 horas del día, en

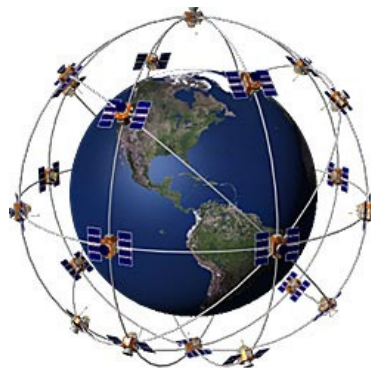


Figura 3 Sistema de Satélites GPS [4]

todo clima, en cualquier parte del mundo con una precisión y exactitud mucho mejor que otros sistemas de radionavegación disponibles en la actualidad o en un futuro previsible. GPS consta de tres segmentos: el espacio, el control y los usuarios.

El segmento espacial consiste en un mínimo de 24 satélites operacionales (ver figura 3) en seis órbitas circulares a 20.200 km por encima de la tierra en un ángulo de inclinación de 55 grados con un período de 12 horas. Los satélites en órbita están espaciados de modo que en cualquier momento un mínimo de 6 satélites estarán en vista a los usuarios en cualquier parte del mundo. Los satélites transmiten continuamente, en dos frecuencias la civil a 1575,42 MHz y la militar a 1227,60 MHz, datos de posición y tiempo a los usuarios en todo el mundo.

El segmento de control consta de una estación de control principal en tierra, en Colorado Springs, EE.UU., cinco estaciones de monitoreo y tres antenas en tierra ubicados alrededor el mundo. Las estaciones de monitoreo dan seguimiento de todos los satélites de la red GPS a la vista y recogen información enviadas por los satélites. Las estaciones de monitoreo envían la información que se obtiene de cada uno de los satélites a la estación de control principal, que calcula las órbitas de

satélite de manera muy precisa. La información se formatea en mensajes de navegación actualizadas para cada satélite. La información actualizada se transmite a cada satélite a través de las antenas de tierra, que también transmiten y reciben señales de control de satélites y de vigilancia. El segmento del usuario consiste de los receptores, procesadores, y las antenas que permiten a los operadores en tierra, mar o aire recibir las emisiones por satélite GPS y calcular su posición exacta, velocidad y tiempo [2].

El concepto de funcionamiento de GPS se basa en el recorrido de los satélites de la red. Los usuarios determinan su posición en la tierra mediante la medición de su distancia del grupo de satélites en el espacio. Los satélites actúan como puntos de referencia precisos [4].

Cada satélite GPS transmite una posición exacta y la señal de tiempo. El receptor del usuario mide el tiempo de retardo que le tomó a la señal en llegar al receptor, que es la medida directa de la aparente distancia para el satélite. Las mediciones recogidas de forma simultánea desde cuatro satélites son procesadas para resolver las tres dimensiones de la posición, velocidad y tiempo del receptor [5].

2.2.2 Módulos

El segmento del usuario del sistema GPS se compone de cientos de miles de usuarios en Estados Unidos y sus aliados militares, y decenas de millones de usuarios civiles, comerciales y científicos del Servicio de Posicionamiento Estándar. En general, los receptores GPS, como el que se muestra en la figura 5, se componen de una antena sintonizada a las frecuencias transmitidas por los satélites, procesadores de recepción, y un reloj muy estable (a menudo un oscilador de cristal). También pueden incluir una pantalla para proporcionar información de ubicación y velocidad para el usuario. Un receptor es a menudo descrito por su número de canales: esto significa cuántos satélites se puede controlar de forma simultánea. Originalmente se limitan a cuatro o cinco, esta ha aumentado progresivamente en los últimos años para que, a partir de 2007, los receptores suelen tener entre 12 y 20 canales [5].



Figura 4 Módulo GPS FV-M5 marca SANAV [6]

Muchos receptores GPS pueden enviar datos de la posición a una PC u otro dispositivo usando el protocolo NMEA 0183. Aunque este protocolo se define oficialmente por la *National Marine Electronics Association* (NMEA), las referencias a este protocolo han sido recopiladas de los registros públicos, permitiendo que las herramientas de código abierto puedan leer el protocolo, sin violar las leyes de propiedad intelectual. Otros protocolos propietarios también existen, como el SiRF y protocolos de MTK. Los receptores pueden interactuar con otros dispositivos que utilizan métodos que incluyen una conexión en serie, USB o Bluetooth.

2.2.3 Formato NMEA

La trama del formato NMEA utilizado es:

```
$GPRMC,081836,A,3751.65,S,14507.36,E,000.0,360.0,130998,011.3,E*62
```

Donde cada campo es descrito en la tabla II.

Parámetro	Ejemplo	Descripción
Identificador	\$GPRMC	Identifica el tipo de Cadena
Hora UTC	081836	Hora UTC en HHmmss
Validez del dato	A	A: válido; V: no válido
Latitud	3751.65	Latitud en GGGmm.mm
Norte / Sur	S	
Longitud	14507.36	Longitud en GGGmm.mm
Este / Oeste	E	
Velocidad en Nudos	000.0	Velocidad sobre la tierra
Dirección en grados	360.0	Curso de navegación
Fecha UTC	130998	Fecha en DDMMYY
Variación Magnética en grados	011.3	Variación del este que se resta del curso verdadero
Este/Oeste	E	
Checksum	*62	Control de errores

Tabla II Campos de sentencia NMEA GPRMC

Se utiliza este formato para enviar la información de posición a otro equipo desde un receptor GPS. Existen varios formatos NMEA, con

características muy similares dependiendo de la aplicación. En este caso usaremos el formato \$GPRMC que es una trama recomendada cuando se requiere datos mínimos [7].

2.3 Tarjetas de Memoria SD

Secure Digital (SD) es un tipo de tarjeta de memoria, inventado por el fabricante de tecnología Panasonic, que se utiliza como medio de almacenamiento externo específicamente diseñado para cumplir con los requisitos de seguridad, capacidad, rendimiento, y ambiente inherentes a dispositivos electrónicos de audio y vídeo de reciente aparición.

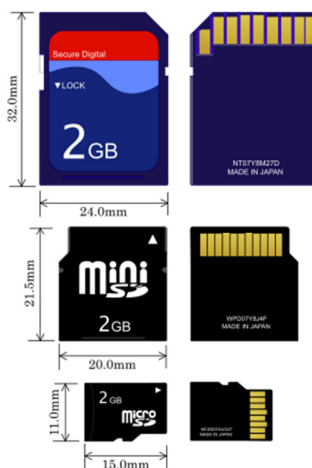


Figura 5 Tarjetas de Memoria SD [9]

Ampliamente utilizada en casi todo tipo de dispositivos portátiles como cámaras fotográficas digitales, PDA, teléfonos móviles, computadores, entre muchos otros. La figura 5 muestra los tamaños de los tipos de memorias SD.

2.3.1 Tipos

La tarjeta SD no es el único estándar de tarjetas de memoria, existen otros formatos como son el MiniSD y el MicroSD . Estas tarjetas más pequeñas se pueden utilizar en ranuras del mismo tamaño que MMC/SD/SDIO con un adaptador (que debe conectar los contactos eléctricos así como llevar a cabo el contacto físico). Es difícil ya crear dispositivos de E/S con el factor de forma del SD y esto será aún menos posible con tamaños más pequeños.

Cientos de fabricantes han construido miles de tarjetas de memoria y dispositivos con los estándares SD. Las tarjetas de memoria vienen en una variedad de capacidades de almacenamiento, las clases de velocidad y los tres tamaños físicos diferentes: tarjeta de memoria SD, tarjeta de memoria miniSD, y la tarjeta de memoria microSD. Con tantas opciones, es importante saber cuál es la tarjeta de memoria SD que funciona mejor con sus teléfonos móviles, tablets, computadoras

personales, cámaras de video, cámaras digitales y otros dispositivos electrónicos. Las características de los tipos de memorias SD se las enlista en la tabla III [8].









	Full SD	miniSD	microSD	Capacidad SD Card
D				Hasta 2GB
DHC				Alta Capacidad 4GB to 32GB
SDXC				Capacidad extendida de 32GB hasta 2TB

Tabla III Tipos de tarjetas de memoria SD

2.3.2 Características Tarjetas SD [9]

- Dirigida a aplicaciones portátiles y estacionarias
- Rango de operación de voltaje: 2.7-3.6 V
- Diseñada para sólo lectura y/o lectura / escritura.
- Capacidad de la Memoria: Capacidad estándar de tarjeta de memoria SD (SDSC): Hasta 2 GB; tarjeta de memoria SD de alta capacidad (SDHC): hasta 32 GB; tarjeta de memoria SD de capacidad ampliada (SDXC): hasta 2 TB.
- Modo de velocidad de bus (con 4 líneas de datos en paralelo) por defecto 3.3 V de señalización, frecuencia de hasta 25 MHz, hasta 12.5 MB / s.
SDR12: 1.8 V de señalización, de frecuencia de hasta 25 MHz, hasta 12.5MB/s
SDR25: 1.8 V de señalización, frecuencia de hasta 50 MHz, hasta 25MB/s
SDR50: 1.8 V de señalización, frecuencia de hasta 100 MHz, hasta 50MB/s
SDR104: 1.8 V de señalización, frecuencia de hasta 208 MHz, hasta 104MB/s
DDR50: 1.8 V de señalización, frecuencia de hasta 50 MHz, hasta50MB/s

- Corrección de errores de campo de memoria
- El retiro de la tarjeta durante la operación de lectura nunca dañará el contenido
- Mecanismo de Protección de Contenido - Cumple con la más alta seguridad del estándar SDMI.
- Protección por contraseña de las tarjetas (CMD42 - LOCK_UNLOCK)
- Protección contra escritura característica con interruptor mecánico
- Funciones incorporadas de protección de escritura (permanente y temporal)
- Detección de tarjeta (inserción / extracción)
- Características de Protocolo del canal de comunicación (ver tabla IV):

Canal de Comunicación de tarjeta SD

**Canal de comunicación de 6 cables (reloj,
comando, 4 líneas de datos)**

**Transferencia de datos con protección de
errores**

**Transferencia de datos orientada a bloque
simple y múltiple**

T
a

Tabla IV Canal de comunicación Tarjeta SD

- Tamaño estándar de grosor de la memoria SD se define como 2,1 mm (normal) y 1,4 mm (memoria SD delgada).

CAPITULO III

3. DISEÑO E IMPLEMENTACIÓN

En este capítulo se presenta al detalle el diseño e implementación del proyecto y de esta manera dar a conocer la forma como se relacionan los diferentes componentes mencionados en el capítulo 2. Se detalla el hardware y software utilizados, así como su interacción.

El hardware esta compuesto por la configuración del FPGA, el receptor GPS, la Memoria SD y la tarjeta DE2 con los elementos para la comunicación RS232, el slot de memoria SD y el display LCD de 2x16 que se lo usa como auxiliar para verificar el funcionamiento del sistema. Primero se debe configurar el hardware del FPGA con el software Quartus II, con la herramienta SOPC Builder se escogen los drivers para interactuar con el hardware externo.

EL software del sistema lo constituye la programación del núcleo de procesador NIOS II, desarrollado en lenguaje C en el entorno de programación Eclipse. Con esta programación se maneja los drivers del hardware configurados en el FPGA. LA figura 6 muestra el diagrama general utilizado en este proyecto.

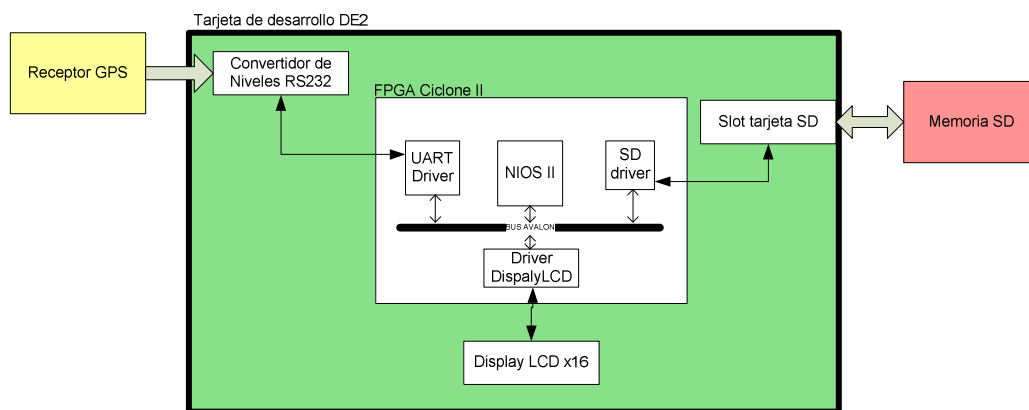


Figura 6 Diagrama de bloques General del sistema

3.1 Hardware

3.1.1 Componentes en la tarjeta de desarrollo

Para el correcto funcionamiento del sistema se deben escoger los elementos de la tarjeta de desarrollo DE2 necesarios. En principio son necesarios tres elementos, además del FPGA Cyclone II (ver figura 7).

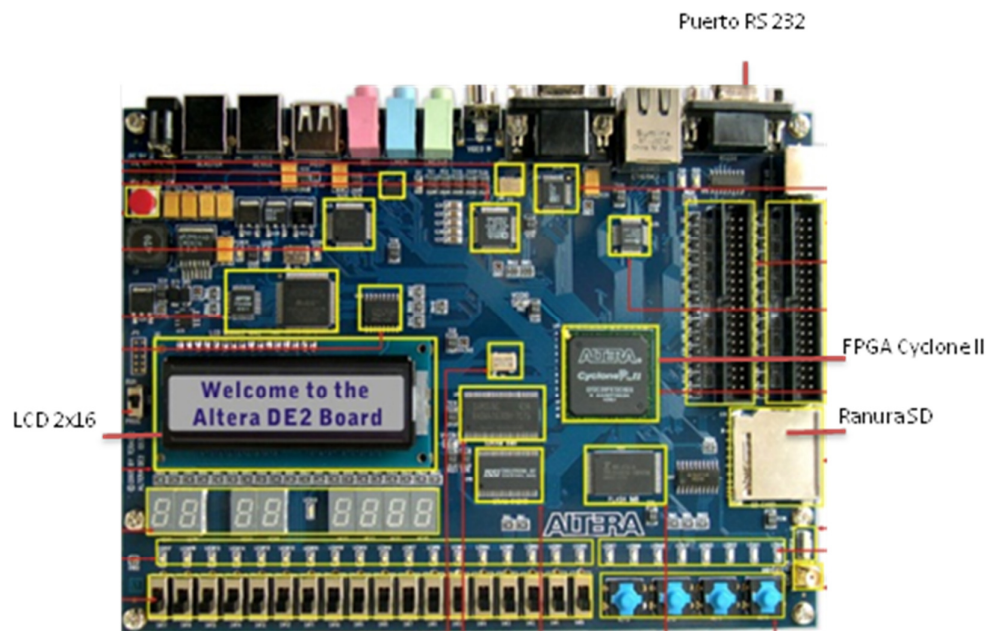


Figura 7 Componentes utilizados de la Tarjeta DE2 [10]

- Interfaz RS232: es la vía de entrada de la información que envía el Receptor GPS.
- Interfaz para tarjeta SD: ranura donde se inserta la tarjeta de Memoria SD que almacena la información de posición procesada.
- LCD 2x16: se presentan los mensajes indicando el estado del sistema y de los componentes conectados a la tarjeta de desarrollo DE2.

3.1.2 FPGA

Para configurar el hardware dentro del FPGA se usa el programa Quartus II, que contiene la herramienta SoPC Builder con la que se configura cada uno de los módulos. Los módulos necesarios para este proyecto son:

- CPU: Procesador NIOS II
- Serial Port: Driver para RS232 UART
- Char_LCD_16x2: Driver para Display de caracteres de 16x2
- AVALON_UP_SD_CARD: Driver para interface de tarjeta SD

Si se parte de configuraciones ejemplo como la DE2_Media_Computer, se debe agregar el módulo AVALON_UP_SD_CARD. Se debe revisar la configuración de pines luego de agregar el módulo de la tarjeta SD, por si hay algún conflicto con los módulos generales.

El programa Quartus II ofrece una interfaz amigable para esta tarea, abriendo en primer lugar un proyecto ya existente sobre el que se hacen los cambios requeridos.

Se escoge el ícono de la herramienta SOPC Builder, con la cual se abrirá una nueva ventana con el listado de los módulos disponibles en el lado izquierdo y en el derecho los módulos a configurar. Para escoger uno de los módulos disponibles basta con dar doble clic en el nombre para pasarlos al lado derecho. Por ejemplo el módulo AVALON_UP_SD_CARD, que se encuentra en la librería University Program/Memory, se lo agrega de esta manera y esta listo para ser configurado dentro del FPGA, como se muestra en la figura 9.

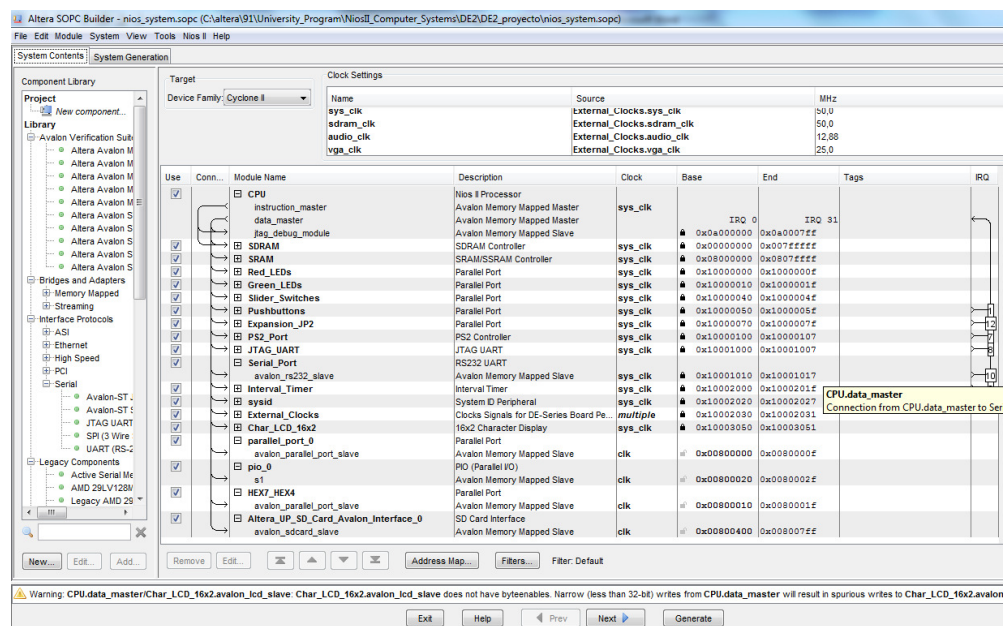


Figura 9 Entorno de Configuración de Hardware SOPC Builder

Se debe tener cuidado con la configuración del módulo Serial Port. Haciendo doble clic sobre el nombre, se abre una ventana en donde se debe escoger las opciones de velocidad 9600 baud, paridad

ninguna, bits de datos 7 y bits de parada 1. En a figura 10 se muestra la configuración para la comunicación serial del receptor GPS utilizado.

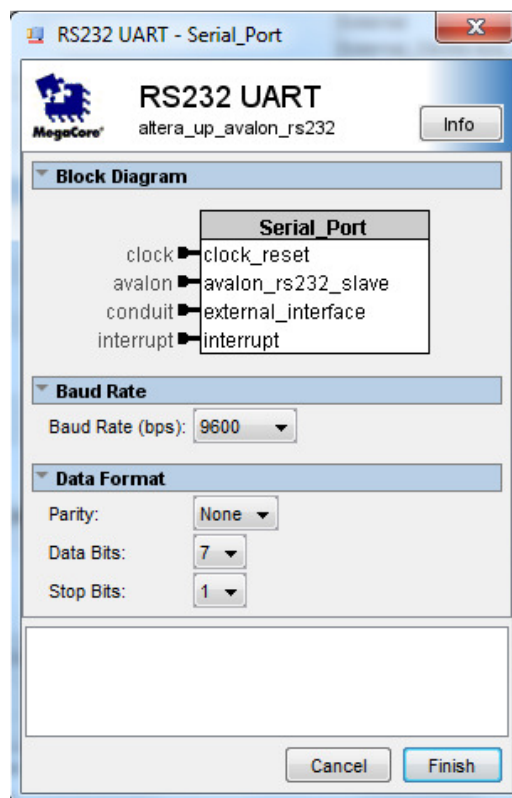


Figura 10 Configuración de Puerto Serie

Una vez terminado de configurar todos los módulos se procede a presionar el botón Generate, el mismo que crea los archivos *.vhd o *.v, que luego son pasados a Quartus II, compilados y cargados en el FPGA, a través de la interfaz USB Blaster.

Para cargar el archivo de configuración en el FPGA se debe encender la tarjeta DE2 y conectarla a un computador con el cable USB por la interfaz antes mencionada. En el programa Quartus II se selecciona la herramienta Programmer, en donde podemos seleccionar el archivo con extensión *.sof y verificando que haya conexión con la tarjeta, se presiona el botón start para que comience la carga del archivo de configuración al FPGA (ver figura 11). Este proceso toma alrededor de dos minutos.

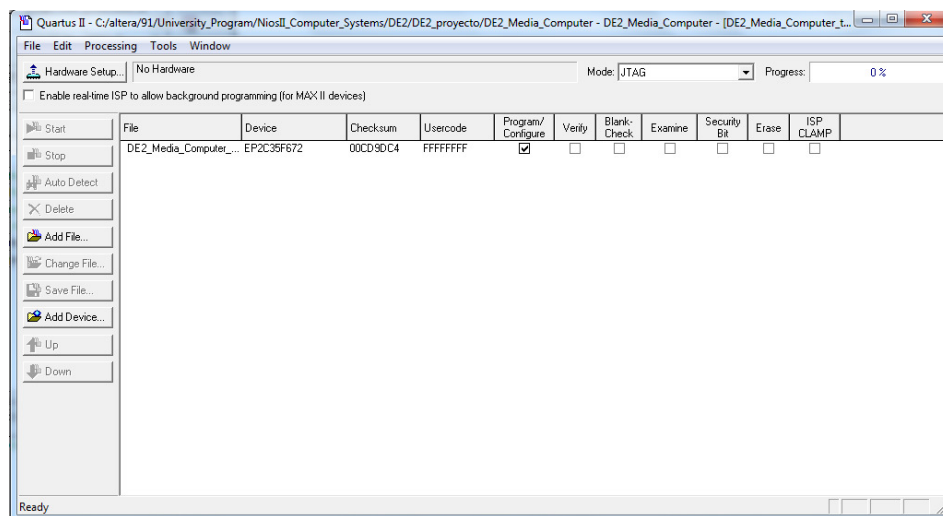


Figura 11 Programador de Quartus II

Una vez configurado el hardware en el FPGA el sistema NIOS queda de la forma como se muestra en la figura 12.

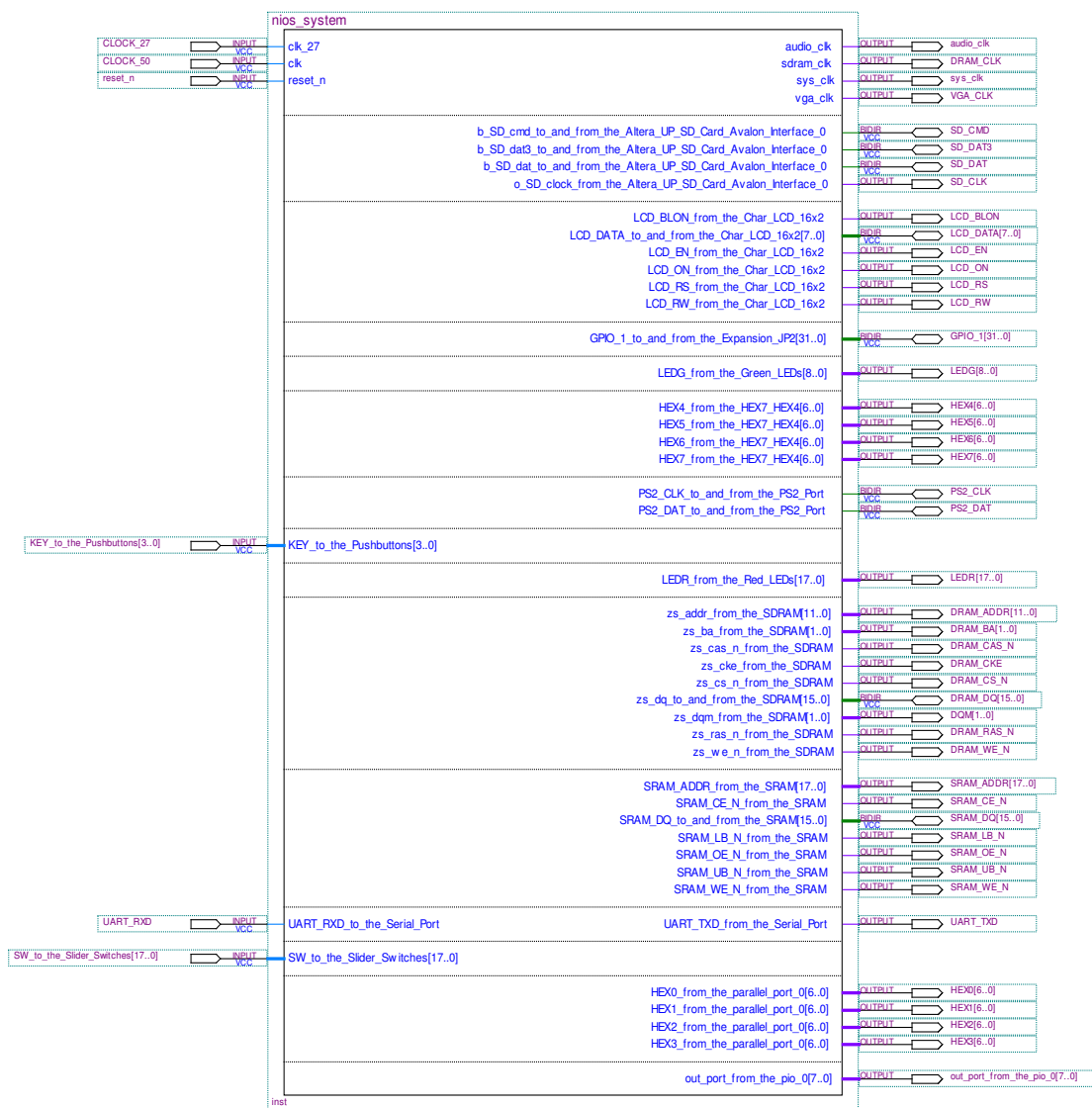


Figura 12 Diagrama del Sistema NIOS II

3.1.3 Receptor GPS

Para facilitar la integración se ha seleccionado un receptor que envíe la información con el protocolo RS 232, así se aprovecha el puerto que viene ya integrado en la tarjeta y no se invierten recursos en armar un hardware adicional para este efecto.

Se escogió al proveedor San Jose Navigation, del que ya se conocen productos de muy buen desempeño. Este proveedor ofrece el receptor GM-158 (ver figura 13), ideal para integraciones rápidas que cuenta con un conector serial para comunicación y un conector ps2 para alimentación de 5V [6]. Para efectos prácticos el conector PS2 se lo reemplazó por una pequeña fuente de 12V.



Figura 13 Receptor GPS SANAV GM-158 [6]

Para que el receptor funcione se debe energizar, conectar a un puerto serial con velocidad de 9600 baud N-7-1, además de poner la antena con vista directa al cielo. El tiempo que le toma sincronizar depende de las condiciones de cobertura, que en condiciones óptimas de cobertura y habiendo estado apagado, toma aproximadamente 50 segundos sincronizar. El Protocolo de envío de datos se muestra a continuación:

Datos de inicio	<pre>\$PMTK010,001*2E \$POLYN,TIME,RESTART_OCCURRED \$POLYN,EPH,0,00000000 \$POLYN,ALM,0,00000000</pre>
Datos no válidos (sin sincronizar)	<pre>\$GPRMC,235946.005,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*77 \$GPRMC,235947.004,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*77 \$GPRMC,235948.004,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*78 \$GPRMC,235949.004,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*79 \$GPRMC,235950.004,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*71 \$GPRMC,235951.004,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*70 \$GPRMC,235952.004,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*73 \$GPRMC,235953.004,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*72 \$GPRMC,235954.005,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*74 \$GPRMC,235955.005,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*75 \$GPRMC,235956.005,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*76 \$GPRMC,235957.005,V,8960.0000,N,00000.0000,E,0.00,0.00,050180,,,N*77 \$GPRMC,194830.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*75 \$GPRMC,194831.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*74 \$GPRMC,194832.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*77 \$GPRMC,194833.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*76 \$GPRMC,194834.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*71 \$GPRMC,194835.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*70 \$GPRMC,194836.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*73 \$GPRMC,194837.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*72 \$GPRMC,194838.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*7D \$GPRMC,194839.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*7C \$GPRMC,194840.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*72 \$GPRMC,194841.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*73 \$GPRMC,194842.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*70</pre>

33

Datos no válidos (sin sincronizar)

```

$GPRMC,194843.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*71
$GPRMC,194844.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*76
$GPRMC,194845.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*77
$GPRMC,194846.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*74
$GPRMC,194847.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*75
$GPRMC,194848.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*7A
$GPRMC,194849.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*7B
$GPRMC,194850.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*73
$GPRMC,194851.159,V,8960.0000,N,00000.0000,E,0.00,0.00,070180,,,N*72
$GPRMC,194852.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*73
$GPRMC,194853.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*72
$GPRMC,194854.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*75
$GPRMC,194855.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*74
$GPRMC,194856.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*77
$GPRMC,194857.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*76
$GPRMC,194858.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*79
$GPRMC,194859.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*78
$GPRMC,194900.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*75
$GPRMC,194901.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*74
$GPRMC,194902.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*77
$GPRMC,194903.159,V,8960.0000,N,00000.0000,E,0.00,0.00,180612,,,N*76
$GPRMC,194904.153,V,0202.6747,S,07954.9338,W,0.16,0.00,180612,,,N*78
$GPRMC,194906.000,V,0202.6748,S,07954.9278,W,0.22,0.00,180612,,,N*70
$GPRMC,194907.000,V,0202.6740,S,07954.9336,W,0.04,0.00,180612,,,N*76

```

Datos válidos

```

$GPRMC,194908.000,A,0202.6750,S,07954.9327,W,0.04,0.00,180612,,,A*60
$GPRMC,194909.000,A,0202.6752,S,07954.9326,W,0.11,0.00,180612,,,A*66
$GPRMC,194910.000,A,0202.6757,S,07954.9325,W,0.21,0.00,180612,,,A*6B
$GPRMC,194911.000,A,0202.6759,S,07954.9329,W,0.29,0.00,180612,,,A*60

```

Se puede ver claramente que primero se sincroniza la hora, luego la fecha y finalmente las coordenadas. Los datos son enviados exactamente cada segundo desde el receptor vía el puerto serie. Como se vio en el capítulo 2, cada campo separado por comas corresponde a un dato de posición y control, sabiendo que el tercer campo corresponde a la validez del dato, es decir, cuando el receptor GPS sincronizó con por lo menos 4 satélites de la red GPS. El análisis

del último dato válido del ejemplo anterior se muestra a continuación:

*\$GPRMC,194911.000,A,0202.6759,S,07954.9329,W,0.29,0.00,180612,,,A*60*

En donde:

\$: inicio de cadena

GPRMC: tipo de cadena NMEA

194911.0000: Hora UTC 19 horas, 49 minutos, 11.0 segundos

A: Dato Válido

0202.6759: Latitud 2° 2.6759'

S: Latitud Sur

07954.9329: Longitud 79° 54.9329'

W: Longitud Oeste

0.29: Velocidad 0.29 nudos

0.00: Curso 0.00°

180612: Fecha 18 de Junio de 2012

A*60: check sum 60

3.1.4 Tarjeta de memoria SD

Para el almacenamiento de información se escogió a la marca Sandisk en la versión microSD (ver figura 14) con un adaptador para ser insertado en la interfaz disponible para el efecto en la tarjeta de desarrollo DE2. Esta marca es una de las populares por su precio y calidad, siendo muy estables para aplicaciones en condiciones de alta temperatura como es el caso dentro de un vehículo. La versión microSD se la escogió por estar más al alcance al momento de desarrollar el presente proyecto. El almacenamiento se lo hará cada 10 segundos por lo que no es muy importante la velocidad de grabación y únicamente se almacenan los datos válidos.



Figura 14 Tarjeta de memoria microSD marca Sandisk [8]

3.2 Programación del Procesador

Una vez configurado y elegido todo el hardware se procede con la programación del núcleo de procesador Nios II que tiene el FPGA Cyclone II. Para esto se usó el programa Nios II basado en el entorno Eclipse.

3.2.1 Entorno de programación

En el entorno de Eclipse se puede fácilmente programar en lenguaje C las tareas que debe hacer el procesador. Se crea un nuevo proyecto basándose la plantilla “hello world”, seleccionando el archivo `nios_system.sopcinfo`, generado en SOPC Builder, que describe la configuración del hardware dentro del FPGA y así agrega las librerías con las funciones a utilizar.

Además se agregan dos librerías, la `259macros.h` y `addressmap.h`. Las dos se las utilizan en el manejo del LCD.

Una vez se tienen listas las librerías, se revisan los ejemplos de uso en la documentación disponible para partir de programaciones básicas. Es útil que el lenguaje sea C, ya que es conocido y la sintaxis resulta bastante familiar. En la figura 15 se observa el entorno de programación.

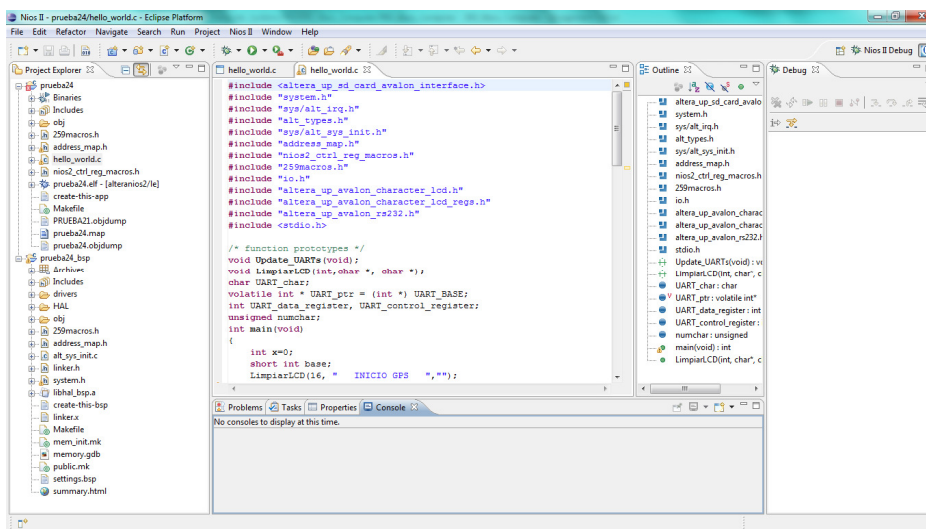


Figura 15 Entorno de programación Eclipse para NIOS II

Lo primero que se hace en el código del programa es verificar que, tanto el módulo GPS como la tarjeta SD, estén correctamente conectados y funcionando. Luego se verifica si los datos recibidos del módulo GPS son validos o no, para, finalmente, almacenarlo en un archivo de texto en la tarjeta de memoria SD. Se repite el ciclo infinitamente hasta que alguno de los componentes externos, módulo GPS o Tarjeta SD, sean desconectados. Al detectar que ya no se reciben datos seriales, se cierra el archivo de texto en la memoria SD y se termina el programa en el procesador. El archivo de texto en la SD, es utilizado para generar el recorrido en un mapa.

El hilo de ejecución de la programación en el procesador Nios II se lo muestra en un diagrama de flujo (ver figura 16).

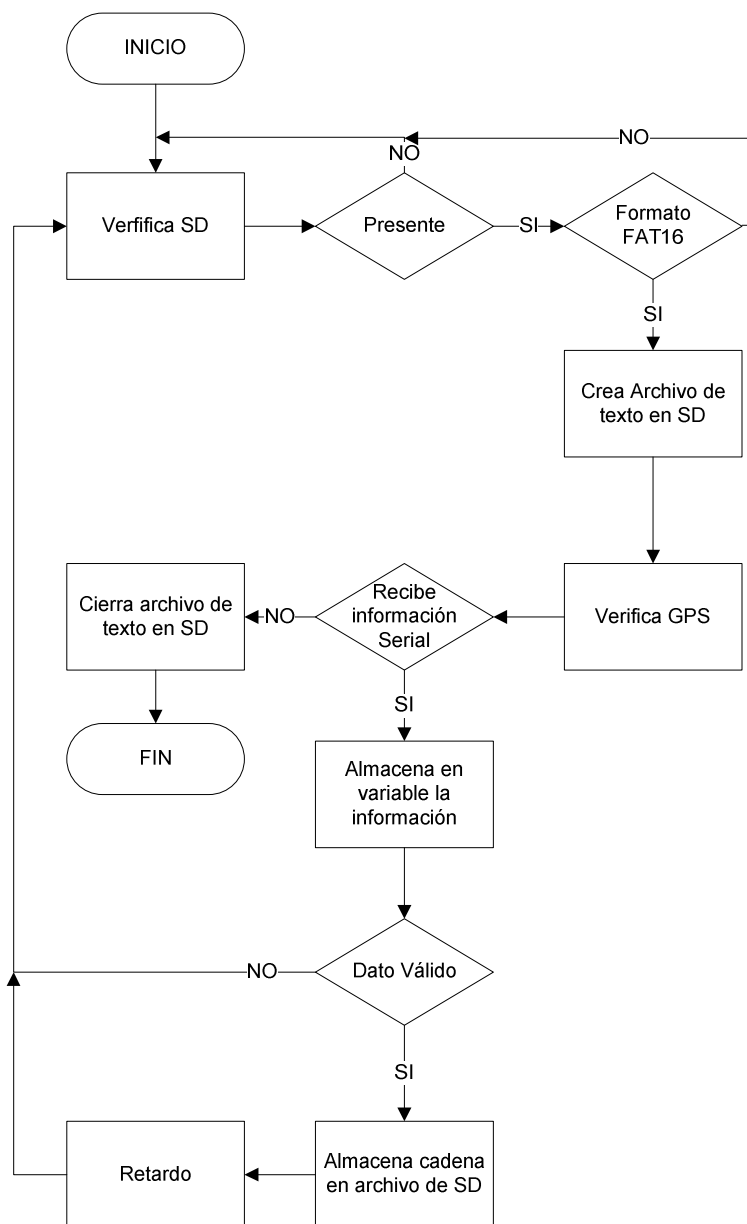


Figura 16 Diagrama de Flujo de programa

Bajo ese esquema se presenta todo el código fuente debidamente documentado en el anexo A.

CAPITULO 4

4. PRUEBAS Y RESULTADOS

Finalmente, se muestran los resultados prácticos del sistema en pleno funcionamiento en un vehículo haciendo un recorrido por algunos puntos de la ciudad de Guayaquil.

4.1 Almacenamiento de datos GPS

La instalación del sistema se la hizo con un inversor de voltaje que se

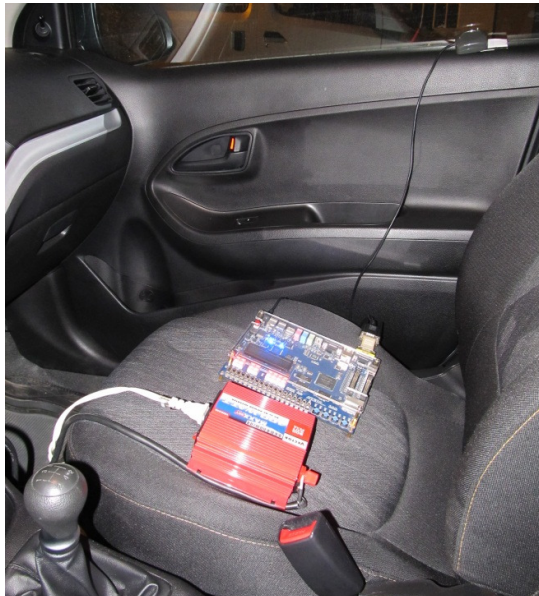
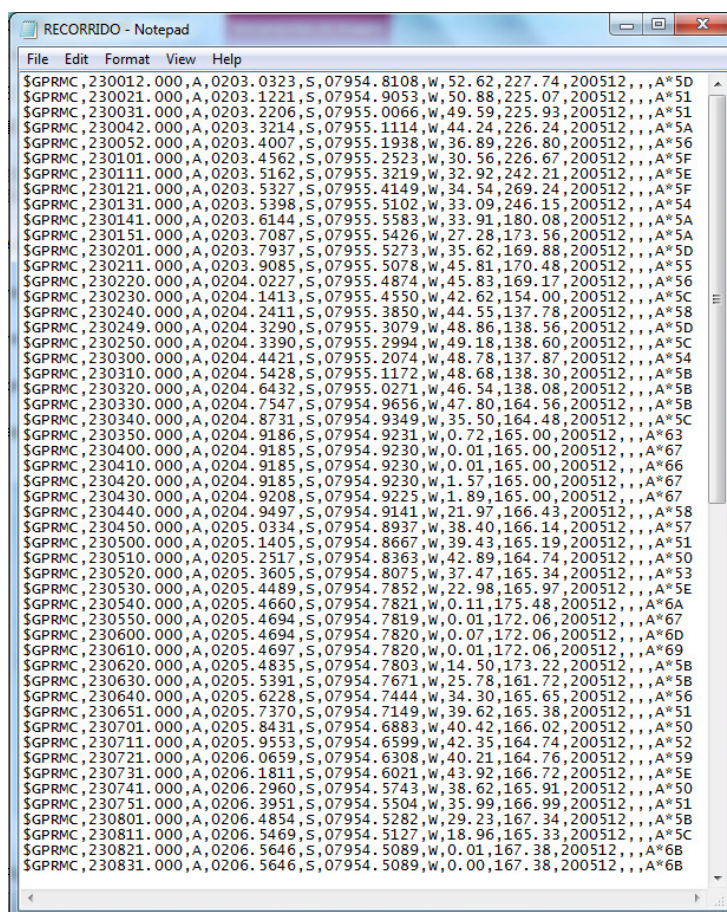


Figura 17 Prototipo instalado en un vehículo

conecta en la cigarrillera de un vehículo, que transforma los 12 VDC del sistema eléctrico a 110VAC para alimentar la tarjeta de prueba y el receptor GPS (ver figura 17).

Al momento de encender el vehículo, se energiza el sistema, y se espera a que el módulo se sincronice y comience a almacenar la información recibida cada 10 segundos. Esta información es procesada y almacenada en un archivo de texto (ver figura 18). El recorrido dura aproximadamente 15 minutos por el norte centro de ciudad.



```

$GPRMC,230012.000,A,0203.0323,S,07954.8108,W,52.62,227.74,200512,,A*5D
$GPRMC,230021.000,A,0203.1221,S,07954.9053,W,50.88,225.07,200512,,A*51
$GPRMC,230031.000,A,0203.2206,S,07955.0066,W,49.59,225.93,200512,,A*51
$GPRMC,230042.000,A,0203.3214,S,07955.1114,W,44.24,226.24,200512,,A*5A
$GPRMC,230052.000,A,0203.4007,S,07955.1938,W,36.89,226.80,200512,,A*56
$GPRMC,230101.000,A,0203.4562,S,07955.2523,W,30.56,226.67,200512,,A*5F
$GPRMC,230111.000,A,0203.5162,S,07955.3219,W,32.92,242.21,200512,,A*5E
$GPRMC,230121.000,A,0203.5327,S,07955.4149,W,34.54,269.24,200512,,A*5F
$GPRMC,230131.000,A,0203.5398,S,07955.5102,W,33.09,246.15,200512,,A*54
$GPRMC,230141.000,A,0203.6144,S,07955.5583,W,33.91,180.08,200512,,A*5A
$GPRMC,230151.000,A,0203.7087,S,07955.5426,W,27.28,173.56,200512,,A*5A
$GPRMC,230201.000,A,0203.7937,S,07955.5273,W,35.62,169.88,200512,,A*5D
$GPRMC,230211.000,A,0203.9085,S,07955.5078,W,45.81,170.48,200512,,A*55
$GPRMC,230220.000,A,0204.0227,S,07955.4874,W,45.83,169.17,200512,,A*56
$GPRMC,230230.000,A,0204.1413,S,07955.4550,W,42.62,154.00,200512,,A*5C
$GPRMC,230240.000,A,0204.2411,S,07955.3850,W,44.55,137.78,200512,,A*58
$GPRMC,230249.000,A,0204.3290,S,07955.3079,W,48.86,138.56,200512,,A*5D
$GPRMC,230250.000,A,0204.3390,S,07955.2994,W,49.18,138.60,200512,,A*5C
$GPRMC,230300.000,A,0204.4421,S,07955.2074,W,48.78,137.87,200512,,A*54
$GPRMC,230310.000,A,0204.5428,S,07955.1172,W,48.68,138.30,200512,,A*5B
$GPRMC,230320.000,A,0204.6432,S,07955.0271,W,46.54,138.08,200512,,A*5B
$GPRMC,230330.000,A,0204.7547,S,07954.9656,W,47.80,164.56,200512,,A*5B
$GPRMC,230340.000,A,0204.8731,S,07954.9349,W,35.50,164.48,200512,,A*5C
$GPRMC,230350.000,A,0204.9186,S,07954.9231,W,0.72,165.00,200512,,A*63
$GPRMC,230400.000,A,0204.9185,S,07954.9230,W,0.01,165.00,200512,,A*67
$GPRMC,230410.000,A,0204.9185,S,07954.9230,W,0.01,165.00,200512,,A*66
$GPRMC,230420.000,A,0204.9185,S,07954.9230,W,1.57,165.00,200512,,A*67
$GPRMC,230430.000,A,0204.9208,S,07954.9225,W,1.89,165.00,200512,,A*67
$GPRMC,230440.000,A,0204.9497,S,07954.9141,W,21.97,166.43,200512,,A*58
$GPRMC,230450.000,A,0205.0334,S,07954.8937,W,38.40,166.14,200512,,A*57
$GPRMC,230500.000,A,0205.1405,S,07954.8667,W,39.43,165.19,200512,,A*51
$GPRMC,230510.000,A,0205.2517,S,07954.8363,W,42.89,164.74,200512,,A*50
$GPRMC,230520.000,A,0205.3605,S,07954.8075,W,37.47,165.34,200512,,A*53
$GPRMC,230530.000,A,0205.4489,S,07954.7852,W,22.98,165.97,200512,,A*5E
$GPRMC,230540.000,A,0205.4660,S,07954.7821,W,0.11,175.48,200512,,A*6A
$GPRMC,230550.000,A,0205.4694,S,07954.7819,W,0.01,172.06,200512,,A*67
$GPRMC,230600.000,A,0205.4694,S,07954.7820,W,0.07,172.06,200512,,A*6D
$GPRMC,230610.000,A,0205.4697,S,07954.7820,W,0.01,172.06,200512,,A*69
$GPRMC,230620.000,A,0205.4835,S,07954.7803,W,14.50,173.22,200512,,A*5B
$GPRMC,230630.000,A,0205.5391,S,07954.7671,W,25.78,161.72,200512,,A*5B
$GPRMC,230640.000,A,0205.6228,S,07954.7444,W,34.30,165.65,200512,,A*56
$GPRMC,230651.000,A,0205.7370,S,07954.7149,W,39.62,165.38,200512,,A*51
$GPRMC,230701.000,A,0205.8431,S,07954.6883,W,40.42,166.02,200512,,A*50
$GPRMC,230711.000,A,0205.9553,S,07954.6599,W,42.35,164.74,200512,,A*52
$GPRMC,230721.000,A,0206.0659,S,07954.6308,W,40.21,164.76,200512,,A*59
$GPRMC,230731.000,A,0206.1811,S,07954.6021,W,43.92,166.72,200512,,A*5E
$GPRMC,230741.000,A,0206.2960,S,07954.5743,W,38.62,165.91,200512,,A*50
$GPRMC,230751.000,A,0206.3951,S,07954.5504,W,35.99,166.99,200512,,A*51
$GPRMC,230801.000,A,0206.4854,S,07954.5282,W,29.23,167.34,200512,,A*58
$GPRMC,230811.000,A,0206.5469,S,07954.5127,W,18.96,165.33,200512,,A*5C
$GPRMC,230821.000,A,0206.5646,S,07954.5089,W,0.01,167.38,200512,,A*6B
$GPRMC,230831.000,A,0206.5646,S,07954.5089,W,0.00,167.38,200512,,A*6B

```

Figura 18 Archivo de texto almacenado en la memoria SD.

Durante esta prueba el archivo de texto, con los datos válidos de posición identificados por la letra A de la cadena recibida, ocupa un espacio de 7 KB en la tarjeta de memoria en 15 minutos de recorrido, por lo que se podría estimar que la tarjeta de 2GB utilizada albergue hasta 70000 horas de información.

4.2 Visualización en programa Google Earth

Bajamos la información contenida en la memoria SD en un computador, y los cargamos con el programa en línea que se muestra en la figura 19, donde se escoge la opción Google Earth KML.

http://www.gpsvisualizer.com/map_input?form=googleearth

The screenshot shows the GPS Visualizer website interface. At the top, there is a navigation menu with links: About GPSV, DRAW A MAP, CONVERT A FILE, Atlas: Share a map, Geocode an address, Look up elevations, Google Earth overlays, Calculators, GPSbabel, Help/EAD, Examples, Partner sites: GlobalMotion.com, EveryTrail.com, and EveryTrail Guides. Below the menu, there are several sections: 'Satellite Maps' (Get Maps, Directions, Street Views & Traffic From Your Browser Free!), 'Satellite tracking Africa' (Personal satellite tracking Anywhere in Africa), 'mini GPS Vehicle Tracker' (Size is similar as one credit card, Support realtime tracking software), and 'Localizador para Celular' (localiza tu Novia o Amigos con el GPS locator en tu Cell \$0,25/dia). The main content area features a map titled 'GPS Visualizer: Do-It-Yourself Mapping' with a description: 'GPS Visualizer is a free, easy-to-use online utility that creates maps and profiles from GPS data (tracks and waypoints, including GPX files), driving routes, street addresses, or simple coordinates. Use it to see where you've been, plan where you're going, or visualize geographic data (business locations, scientific observations, events, customers, real estate, geotagged photos, "GPS drawing," etc.).' Below the map is a 'Get started now' section with a file upload field, a dropdown for 'Choose an output format: Google Maps', and a 'Go!' button. To the right of the map, there are links for 'To set more options, use the detailed input pages:' including 'Google Maps', 'Google Earth KML', 'Convert to GPX', 'Convert to plain text', 'JPEG/PNG/SVG maps', 'Plot data points', 'Profiles (elevation, etc.)', 'Geocoding', and 'Freehand drawing tool'. A 'Donate' button is also visible on the right side.

Figura 19 Página web GSP Visualizer

Se carga una nueva página donde se escogen las siguientes opciones (ver figura 20):

The screenshot shows the GPS Visualizer web application interface. The browser address bar displays 'http://www.gpsvisualizer.com/map_input/form-g...'. The page is divided into several sections with green headers:

- General map parameters:** Includes 'Output file type: .kml (uncompressed)', 'Units: Metric', 'Google Earth doc name:', 'Add DEM elevation data: No', and 'Time offset: -5 hrs'. There are also links to 'show advanced options [+]' and 'Add time stamps, if possible: Yes'.
- Track options:** Includes 'Altitude mode: Clamped to ground', 'Draw a shadow: No', 'Tickmark interval:', 'Draw as waypoints: Yes, named with sequential trackpoint numbers', 'Track opacity: 100%', 'Line width: 4', 'Colorize by: Track (recommended)', and 'Default color: Red'. There are also links to 'show advanced options [+]'.
- Waypoint options:** Includes 'Show waypoints: In bounds of track plus padding', 'Wpt. display padding: 10%', 'Altitude mode: Clamped to ground', 'Default icon color: white', 'Default icon: Small square', and 'Waypoint labels: Labels on waypoints + tickmarks + trackpoints'. There are also links to 'show advanced options [+]'.
- Contact information:** Includes 'Your e-mail: (OPTIONAL)' and a note: 'This is for impromptu tech support, NOT a mailing list!'.
- Upload your GPS data files here: ?** (Total size of all files cannot exceed 3 MB): Includes three 'Browse...' buttons for File #1, File #2, and File #3, and a link to 'Show additional file input boxes'.
- Or paste your data here: ?**: Includes a text area with the header 'name, desc, latitude, longitude' and a 'Force plain text to be this type: default' dropdown.
- Or provide the URL of data on the Web:** Includes a text input field.

At the bottom right, there is a 'Create KML file' button and a checkbox for 'Open in new window'. At the bottom center, there is a 'Reset the entire form' button.

Figura 20 Opciones a elegir en página GPS Visualizer

- Tipo de archivo de salida: .kml (descomprimido)
- Desviación de Hora: -5
- Adicionar etiquetas de tiempo: Si
- Dibujar como Puntos de ruta: Si, etiquetados con números de puntos de trayectoria secuenciales
- Etiquetas de Puntos de ruta: Etiquetas en puntos de ruta + Marcas + puntos de trayectoria.

- Se escoge el archivo en la ruta almacenada, que contiene la información en modo texto de los datos de posición en formato NMEA.
- Se presiona el botón Crear archivo KML

Con estas opciones el archivo KML resultante de la figura 21, una vez ejecutado, mostrará en la interface de Google Earth la trayectoria en rojo con puntos numerados donde al hacer clic en cada uno de los puntos se vera información de hora, fecha y velocidad (ver figura 22).

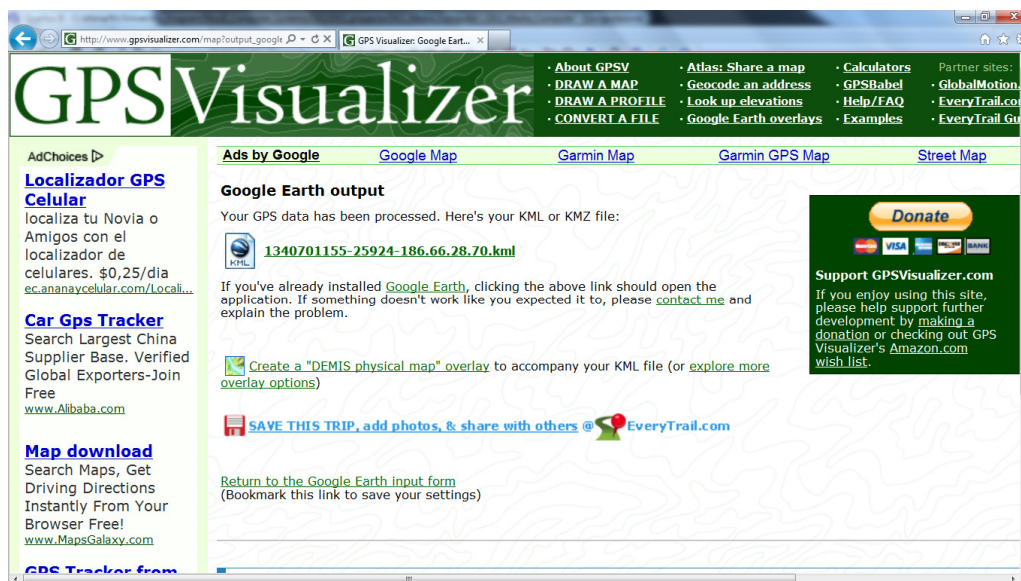


Figura 21 Archivo generado por página GPS Visualizer

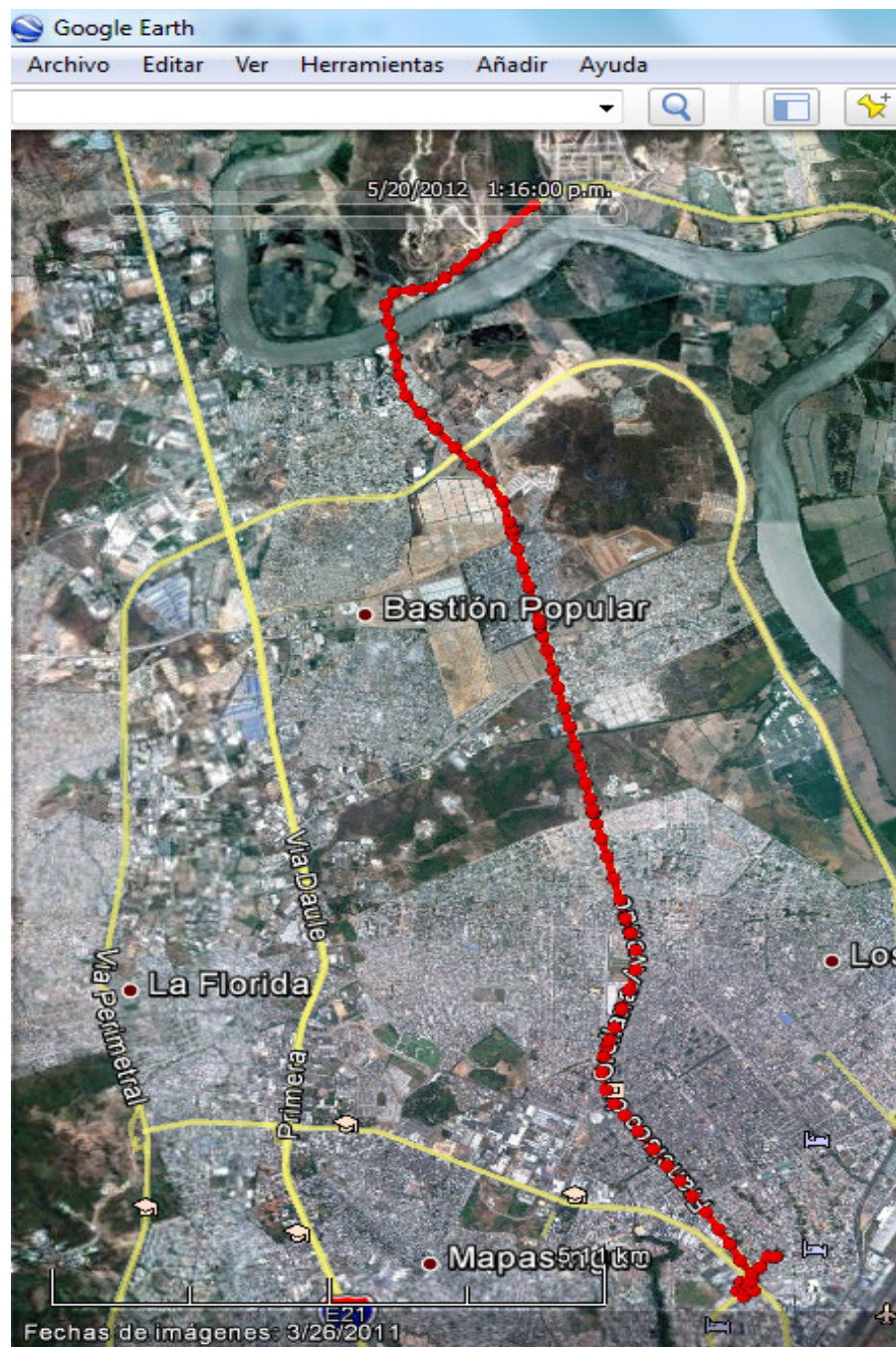


Figura 22 Presentación del recorrido en Google Earth

CONCLUSIONES

1. La tarjeta DE2 facilita mucho el desarrollo de cualquier prototipo donde el núcleo del procesador es el NIOS II y el hardware adicional de la tarjeta sea suficiente para el mismo. Sin embargo, información con respecto a cierto hardware está muy limitada en la página del fabricante para programar en lenguaje C.
2. Se pudo constatar la facilidad de configurar un nuevo hardware en un FPGA, aun luego de haber implementado el proyecto en la primera versión. Para aumentar el LCD de 16x2, luego de las primeras configuraciones del hardware en el FPGA, simplemente se configuró el módulo para controlar el LCD y se aumentaron las líneas de programación en el procesador para manejarlo, sin tener que cambiar de circuito Integrado central, como hubiese sido el caso si se trabaja con un microcontrolador y ya no se tuviesen mas puertos de comunicación.
3. Se concluye además que es posible leer y procesar datos provenientes de un receptor GPS. La información de esta tecnología es bastante amplia en la red, así como existe una importante cantidad de fabricantes de este tipo de dispositivos que

utilizan este servicio que actualmente es gratuito.

4. Las aplicaciones para el sistema GPS son muchas, y van creciendo a medida que los módulos son cada vez mas pequeños, llegando fácilmente a pensar que en pocos años se tendería a perder la privacidad al saber la ubicación de todo a nuestro alrededor.

5. Las tarjetas SD son una solución económica y popular para cualquier tipo de aplicación donde se requiera almacenar temporalmente información, debiendo tomar en cuenta las condiciones de uso como temperatura, salinidad y humedad.

6. Se verificó además las limitaciones de la tecnología GPS, como el hecho de no siempre tener vista al cielo y los márgenes de error al no contar con buena cobertura.

RECOMENDACIONES

1. Se recomienda, antes de comenzar a trabajar con un FPGA, tener conocimientos de programación en C, VHDL y electrónica digital para poder avanzar sin retrasos.
2. Es recomendable hacer las pruebas ubicando correctamente la antena para no perder datos durante la misma. También es recomendable alejar la antena de equipos que emita cualquier tipo de radiación electromagnética sea por potencia o en una frecuencia cercana.
3. Para hacer pruebas más prolongadas, se recomienda usar otra tarjeta de desarrollo más pequeña debido a las limitaciones de espacio dentro de un vehículo. Manipular el equipamiento necesario para las pruebas se vuelve complicado dentro de un vehículo en movimiento.
4. Al trabajar con equipos que intercambien información con RS232, se debe revisar muy bien la configuración de velocidad y trama de datos, ya que de no estar con la misma configuración, ninguno de los dos equipos podrá comunicarse.

ANEXOS

ANEXO A

Código Fuente Nios II

```
/*  
  
* Titulo: Programa para recibir y almacenar datos GPS con NIOS II  
* Auto: Diego Jácome  
*  
* Este programa verifica que estén correctamente conectados, a la tarjeta DE2,  
* tanto un modelo GPS como la tarjeta de memoria SD. Recibe los datos del modulo  
* GPS por el puerto serial RSR232, se analiza la información para saber si son datos  
* válidos o no. Si son datos validos se los almacena en la memoria SD en forma de texto.  
* Se utiliza la LCD de 2x16 para verificar en todo momento el estado y conexiones del  
* sistema.  
*  
*/  
  
#include <altera_up_sd_card_avalon_interface.h>  
#include "system.h"  
#include "sys/alt_irq.h"  
#include "alt_types.h"  
#include "sys/alt_sys_init.h"  
#include "address_map.h"  
#include "nios2_ctrl_reg_macros.h"  
#include "259macros.h"  
#include "io.h"  
#include "altera_up_avalon_character_lcd.h"  
#include "altera_up_avalon_character_lcd_regs.h"  
#include "altera_up_avalon_rs232.h"  
#include <stdio.h>  
  
//Declaración de Funciones  
void VerificarSD(void); //Funcion que verifica estado y conexión de SD  
void VerificarGPS(void); //Funcion que verifica estado y conexión de GPS  
void LimpiarBufferUART(void); //Limpia Buffer del puerto serial  
void LlenarCadena(void); //Almacena la información en forma de cadenas de  
//texto en la SD  
void LimpiarLCD(int,char *, char *); //Limpia la pantalla de la LCD y presenta nuevas  
//cadenas  
  
//Variables Globales  
char UART_char, cadena[80]; //Cadena donde se almacena la Informacion GPS  
int indice, caracter;  
volatile int * UART_ptr = (int *) UART_BASE;  
int UART_data_register, UART_control_register;  
unsigned numchar;  
alt_up_sd_card_dev *device_reference = NULL;  
short int base;  
int s =0, d=0,c=0;
```

```

//Funcion Principal
int main(void)
{
    indice=0;
    LimpiarLCD(16, "INICIO GPS  ", ""); // Muestra el mensaje de inicio en la LCD
    usleep(1000000);

    while (1) //Lazo infinito de operaci3n
    {
        if(d==0)
        {
            VerificarSD (); // Verifica estado y conexi3n de SD
            d=1;
        }
        VerificarGPS(); //Verifica estado y conexi3n de GPS
        if(c==1) break; //Almacena la informaci3n GPS en la SD
        LlenarCadena();
    }
    return 0;
}

void LimpiarBufferUART( )
{
    while(1)
    {
        UART_data_register = *(UART_ptr);
        if ( (UART_data_register& 0xFFFF0000))
        {
            UART_control_register = *(UART_ptr + 1);
        }
        else
        {
            break;
        }
    }
    return;
}

void VerificarSD( )
{
    int connected = 0;
    // verifica si existe el HW para leer/escribir memoria SD
    device_reference = alt_up_sd_card_open_dev("/dev/Altera_UP_SD_Card_Avalon_Interface_0");
    if (device_reference != NULL)
    {
        while(1) //Lazo infinito esperando que inserte un SD
        {
            //Verifica si existe una memoria SD insertada en el slot
            if ((connected == 0) && (alt_up_sd_card_is_Present()))
            {

```

```

        //presenta en la LCD Mensaje
        LimpiarLCD(16, " TARJETA SD OK ", "");
        usleep(500000);
        //Verifica formato de la SD
        if (alt_up_sd_card_is_FAT16())
        {
            LimpiarLCD(16, " FAT16 OK ", "");
            usleep(500000);
            // Crea el archivo donde se almacena la Informacion
            base = alt_up_sd_card_fopen("RECORRIDO.txt", true);
            break;
        }
        else
        {
            LimpiarLCD(16, " ERROR FAT16 ", "");
            usleep(500000);
        }
        connected = 1;
    }
    else if ((connected == 0) && (alt_up_sd_card_is_Present() == false))
    {
        LimpiarLCD(16, " SIN TARJETA SD ", "");
        usleep(500000);
        connected = 0;
    }
}
}
return;
}
//funcion que verifica estado y conexion de GPS
void VerificarGPS( )
{
    LimpiarBufferUART(); //Borra cualquier dato en el buffer del puerto serie
    UART_char=0;
    bool a;
    while(UART_char!='$') //Toma un solo dato GPS
    {
        usleep(1000000);
        UART_data_register = *(UART_ptr);
        if ( (UART_data_register&0xFFFF0000))
        //Verifica si el Buffer tiene caracteres
        {
            UART_char = (char) (UART_data_register & 0xFF);
            //Almacena el cada caracter un variable temporal
            UART_control_register = *(UART_ptr + 1);
            //aumenta direccion
        }
        else
        {
            LimpiarLCD(16, "SIN CONEXION GPS", "");
            a=alt_up_sd_card_fclose(base);
            // cierra el archivo apenas el gps es desconectado
            usleep(2000);
            c=1;
        }
    }
}

```

```

        return;
    }
}
LimpiarLCD(16, "CONEXION GPS OK ", "");
usleep(2000);
return;
}
// Selecciona los datos GPS validos y los almacena en la SD
void LlenarCadena()
{
    int x=0;
    int t=0;
    for(x=0;x=8;x++) cadena[x]=0; //Limpia cadena
    cadena[indice]=UART_char; //Almacena primer caracter
    while(t!=2)
    {
        UART_data_register = *(UART_ptr);
        if ( (UART_data_register & 0xFFFF0000)
        {
            UART_char = (char) (UART_data_register & 0xFF);
            if(UART_char=='*') t=t+1;
            UART_control_register = *(UART_ptr + 1);
            indice=indice+1;
            cadena[indice]=UART_char;

        }
        else
        {
            break;
        }
    }
    cadena[indice+1]=(char)(13);
    cadena[indice+2]=(char)(10);

    t=0;

    if (cadena[18]=='A') //Verifica si es dato válido
    {
        LimpiarLCD(16, "GPS SINCRONIZADO", "");
        x=0;
        do
        {
            alt_up_sd_card_write(base, cadena[x]);
            usleep(20000);
            x=x+1;
        }while(cadena[x]!=(char)(10));

        usleep(7000000); //Retardo para tener 10 segundos entre dato almacenado
    }
    else
    {
        LimpiarLCD(16, " PERDIDA GPS ", "");
    }
}

```

```

        usleep(2000);
    }
    indice=0;
    return;
}
//Limpia la LCD y coloca la caracteres del puntero *texto
void LimpiarLCD (int i, char *texto, char *texto2)
{
    int j=0;
    IOWR_ALT_UP_CHARACTER_LCD_COMMAND(CHAR_LCD_16X2_BASE, 0X38);
    usleep(2000);
    IOWR_ALT_UP_CHARACTER_LCD_COMMAND(CHAR_LCD_16X2_BASE, 0x0C);
    usleep(2000);
    IOWR_ALT_UP_CHARACTER_LCD_COMMAND(CHAR_LCD_16X2_BASE, 0X01);
    usleep(2000);
    IOWR_ALT_UP_CHARACTER_LCD_COMMAND(CHAR_LCD_16X2_BASE, 0X06);
    usleep(2000);
    IOWR_ALT_UP_CHARACTER_LCD_COMMAND(CHAR_LCD_16X2_BASE, 0X80);
    usleep(2000);
    for (j=0; j<i; j++)
    {
        IOWR_ALT_UP_CHARACTER_LCD_DATA(CHAR_LCD_16X2_BASE, texto[j]);
        usleep(2000);
    }
    for (j=10; j<i; j++)
    {
        IOWR_ALT_UP_CHARACTER_LCD_DATA(CHAR_LCD_16X2_BASE, texto2[j]);
        usleep(2000);
    }
    return;
}

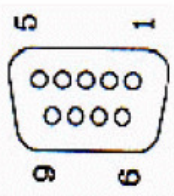
```

ANEXO B

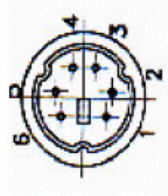
Módulo GPS

Model: GM-158

DB09 Female



DIN-6PIN



<u>pin1 = n/c</u>	<u>pin1 = n/c</u>
<u>pin2 = TXA</u>	<u>pin2 = GND</u>
<u>pin3 = RXA</u>	<u>pin3 = n/c</u>
<u>pin4 = n/c</u>	<u>pin4 = n/c</u>
<u>pin5 = GND</u>	<u>pin5 = +5VDC</u>
<u>pin6 = n/c</u>	<u>pin6 = n/c</u>
<u>pin7 = n/c</u>	
<u>pin8 = n/c</u>	
<u>pin9 = n/c</u>	

DB-9 Pin assignment



DB-9 interface
with RS-232
protocol



USB interface

Specifications:

PHYSICAL CONSTRUCTION		
Dimension	L46.2mm*W38mm*H16.1mm	
Weight	214 gram(with cable)	
Enclosure	Highly impact; corrosion-proof	
Mounting	Magnetic mount	
Construction	Ultrasonic welded, fully waterproof	
ENVIRONMENTAL CONDITIONS		
Temperature	Operating: -0 ~ +70 °C	
	Storage: -20 ~ +85 °C	
COMMUNICATION		
Protocol	NMEA V3.01	
Signal level	EIA/TIA-232 or USB 1.1	
INTERFACE CAPABILITY		
Output Sentences	Standard Sentences: GGA, RMC, GSV*5,VTG, GSA*5	
	Option: GLL, ZDA	
PERFORMANCE		
Sensitivity	-158dBm	
SBAS	1 channel (Support WAAS, EGNOS, MSAS)	
Receiver architecture	32 parallel channels	
Start-up time	3 sec. typical (hot start)	
	35 sec. typical (warm start)	
	41 sec. typical (cold start)	
Position accuracy	Without aid 3.3 m CEP	
	With SBAS 2.6 m	
Velocity	0.1 Knot RMS steady state	
Update Rate	1 Hz(Standard);5Hz optional	
Power Supply	RS-232 @ 3~5V	
	USB @ 5V	
Current Consumption	Acquisition	60mA (RS-232) (USB)
	Tracking	50mA (RS-232) (USB) (first 5 minutes)
		42mA (RS-232) (USB) (after 5 minutes)
		36mA (RS-232) (USB) (after 20 minutes)
Baud Rate	4800 bps (default)	
	Optional:9600/19200/38400/115200 bps are adjustable	

ANEXO C

Tarjeta de Memoria SD

Product Description



SanDisk's 2GB Micro SD card is extremely small in size but big in memory. The semi-removable memory card is based on the mini SD designs for the latest mobile phone products. Its possible to store personal data such as photos, MP3's, video games and other applications.

At roughly the size of a fingernail the Micro SD is designed for mobile phones which are compact, yet fully-featured with storage-intensive applications like cameras, video capture, MP3's, video games etc.

Features:

- Warranty: 5 years
- Can be used in SD compatible devices with use of adapter (*NOT included)
- Includes protective plastic case
- Smallest possible removable storage card
- Read speed up to 5 MB/sec
- Write speed up to 1.5 MB/sec
- Dimensions: 15mm (W) x11mm (H) x1mm (D)
- Weight: 0.40 g. max
- DC Input Voltage: 2.7V to 3.6V
- Capacity: 2GB

Please Note: that Micro SD cards, capacities 2GB and above, are not compatible with all Samsung D600, D900 and E900 mobile phones.

STORAGE CHART

	Tracks		Video Time (min)		Pictures		
	Type		Type		Size		
Card Capacity	MP3	WMA	Normal	HD	1MP	3MP	6MP
128MB	24	48	4	2	113	60	38
256MB	48	96	8	4	226	119	75
512MB	96	192	15	8	452	238	151
1GB	190	380	30	15	905	476	302
2GB	380	760	60	30	1808	952	603
4GB	760	1520	120	60	3619	1905	1206
6GB	1140	2280	180	90	5430	2856	1810
8GB	1520	3040	240	120	7238	3812	2413
12GB	2280	4560	360	180	10860	5712	3620
16GB	3040	6080	480	240	14480	7616	4827
32GB	6080	12160	960	480	28960	15232	9653

REFERENCIAS BIBLIOGRÁFICAS

- [1] Wikipedia, Field Programmable Gate Array,
http://es.wikipedia.org/wiki/Field_Programmable_Gate_Array, 15 de Junio 2012.
- [2] Centro de Navegación de la Guardia Costera de EE.UU., General Information on GPS, <http://www.navcen.uscg.gov/?pageName=GPSmain>, 15 de junio de 2012.
- [3] TERASIC, Altera DE2-70 Board, <http://www.terasic.com.tw/>, 15 de junio de 2012.
- [4] TECNoproject, ¿Qué es GPS?,
<http://www.tecnoprojectltda.com/QUEESGPS.htm>, 15 de junio de 2012.
- [5] WIKIPEDIA, Global Positioning System
http://en.wikipedia.org/wiki/Global_Positioning_System, 15 de junio de 2012.
- [6] San Jose Navigation, Mini GPS Locator,
http://www.sanav.com/gps_receivers/gps_locators/TK-158.htm, 15 de junio de 2012.

[7] Sistema Automático de Información de Posición de Holanda, NMEA Format, <http://aprs.gids.nl/nmea>, 15 de junio de 2012.

[8] SD Group, SD Specifications: Part 1 Physical Layer Simplified Specification, https://www.sdcard.org/downloads/pls/simplified_specs/Part_1_Physical_Layer_Simplified_Specification_Ver_3.01_Final_100518.pdf, Mayo 18 de 2010.

[9] SD Group, SD Choices, <https://www.sdcard.org/consumers/choices/>, 15 de junio de 2012.

[10] Altera Corporation, Altera University Program Media Computer Manual, <http://www.altera.com/literature/lit-index.html>, 28 de Julio de 2010.

[11] Altera Corporation, Cyclone II Device Handbook, http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf, 18 de febrero de 2008.

[12] Altera Corporation, Nios II Processor Reference Handbook, <http://www.altera.com/literature/lit-index.html>, 3 de Mayo de 2011.

[13] Altera Corporation, Altera University Program Secure Data Card IP Core, <http://www.altera.com/literature/lit-index.html>, 12 de Junio de 2009.

[14] Altera Corporation, Quartus II 7.1 Handbook, http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf, 25 de abril de 2007.

