|  |  |
| --- | --- |
|  | **Laboratorio de Microprocesadores (86.07)**  **Laboratorio de Microcomputadoras (66.09)** |
|  | |
| **Informe de Proyecto** | |

|  |  |  |
| --- | --- | --- |
| Anteproyecto: | **DataLogger de Posicionamiento** | |
| Autores: | Monpelat, Facundo | 92716 |
|  | Pisarew, Victor | 84303 |
|  | Palazzo Leandro | 97055 |
| Turno de TP: | 1 (Martes 19 a 22hs) | |
| Docente Guía: | Stola Gerardo, Cofman Fernando, Guido Salaya | |

|  |  |
| --- | --- |
| Observaciones generales | |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Aceptación del docente: |  |
| Fecha |  |

**Índice**

1. **Introducción**
2. **Objetivo del Proyecto**
3. **Descripción del Proyecto**
4. **Diagrama en Bloques (hardware)**
5. **Diagrama de Flujo (firmware)**
6. **Circuito Esquemático**
7. **Listado de Componentes y Costos**
8. **Resultados**

**8.1. *Modo de Operación del GPS***

**8.2. *Modo de Operación del LCD***

**8.3. *Modo de Operación del módulo SD***

**8.4. *Modo de Operación del Microcontrolador (AVR328p)***

***6.5 Especificación del formato de Salida de la SD.***

1. **Conclusiones**

**10. Bibliografía**

**11. Código Fuente**

**12. Apéndice**

**12.1. Cálculo del Baud Rate**

**12.2. Software**

**12.3. Hojas de Datos**

1. **Introducción**

El siguiente documento contiene el desarrollo del proyecto final de la asignatura 66.09 Laboratorio de Microcomputadoras. Este trabajo consistió en la aplicación de los conocimientos adquiridos en la materia para la investigación, diseño y construcción de un dispositivo electrónico que permita automatizar y facilitar una tarea en particular.

1. **Objetivo del Proyecto**

Actualmente, los vehículos o medios, son controlados mediante distintos tipos de dispositivos de rastreo para su mayor seguridad y/o control del mismo. El objetivo principal del proyecto es implementar un dispositivo que permita loguear las distintas ubicaciones que el vehículo transita, como también informacion GIS (“Sistema de información Geográfica” en donde tendríamos longitud, latitud, altura, velocidad, etc) del mismo en forma satelital.

1. **Descripción del Proyecto**

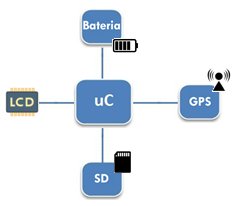
El proyecto consiste en un sistema electrónico diseñado para ser instalado en cualquier paquete o vehículo, capaz de loguear distintas informaciones de rastreabilidad, trayectoria en su recorrido y almacenar dicha información (histórico) en una memoria.

El dispositivo guardará en tiempos determinados (previamente configurados por el usuario), la ubicación y la orientación además de su velocidad.

El usuario podrá luego extraer su memoria para visualizar los datos en una computadora.

1. **Diagrama de Bloques (hardware)**

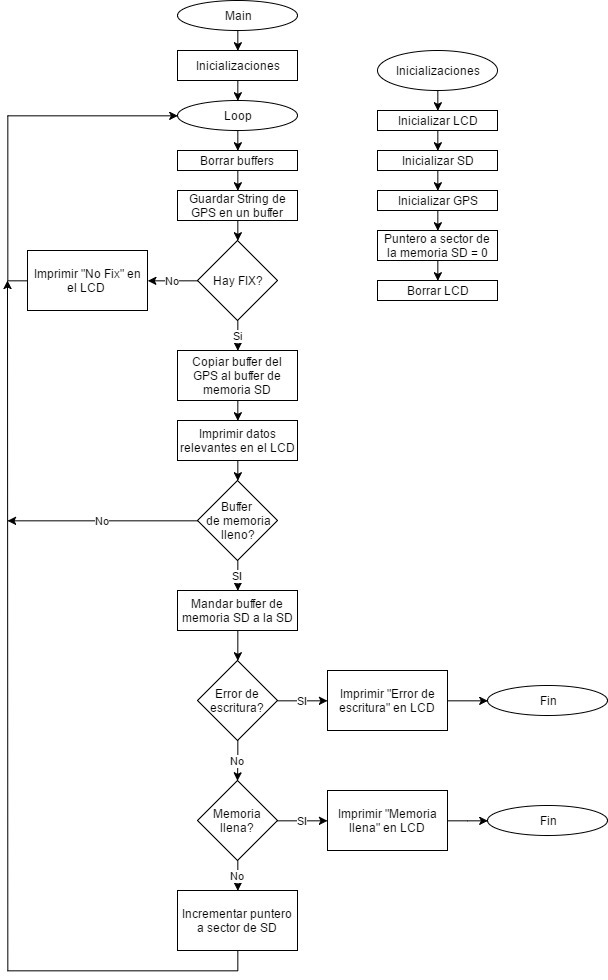
A continuación, se muestra el esquema general del dispositivo.



**Figura 1. Esquema del Dispositivo**

1. **Diagrama de Flujo (Firmware)**

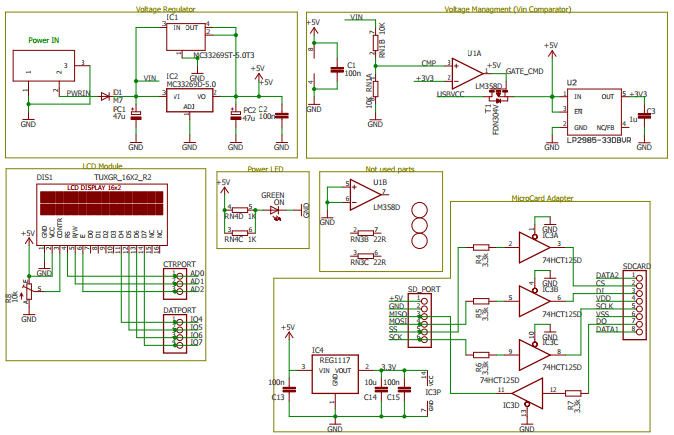
El presente diagrama ilustra la interacción entre los distintos bloques.

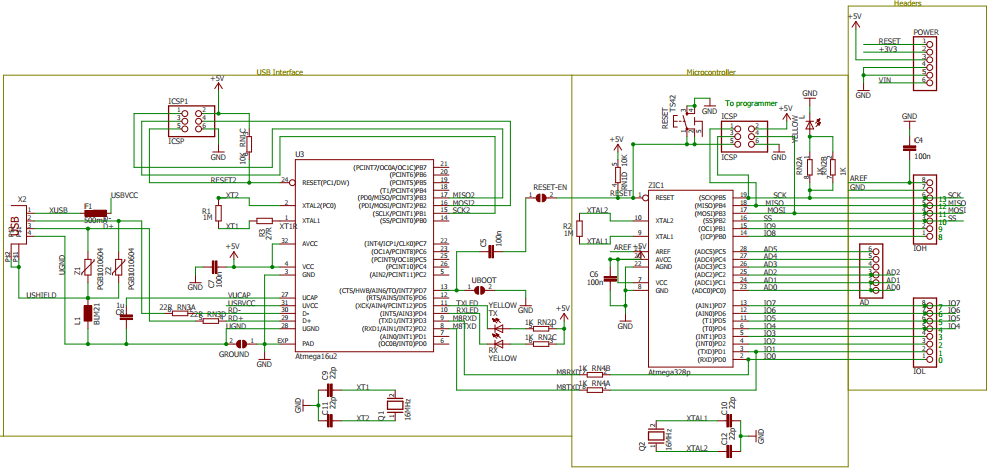


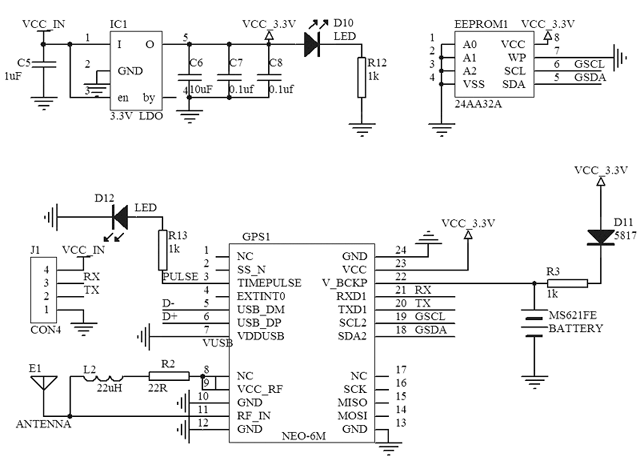
**Figura 2. Diagrama de Flujo**

1. **Circuito Esquemático**

A continuación, se exhibe el circuito eléctrico final de las distintas partes del prototipo.







1. **Listado de Componentes y Costos**

En esta sección se presentan los costos de los componentes más significativos.

|  |  |
| --- | --- |
| Componentes | Precio |
| Microcontrolador ATmega 328p |  |
| Modulo de GPS Ublox Neo 6M 0-001 |  |
| Programador Arduino UNO R3 |  |
| LCD 16x2 HITACHI |  |
| Programador AVR ICSP |  |
| Protoboard de 880 puntos |  |
| Módulo Lector de Tarjeta SD |  |
| Componentes Varios (resistores, capacitores, LEDs, Presets, conectores, puentes, cables) |  |
| Total |  |

**Tabla 1. Listado de Componentes y Costos**

Se le ha dedicado al proyecto aproximadamente horas-hombre. Esto corresponde a 3 días completos por semana durante 6 semanas.

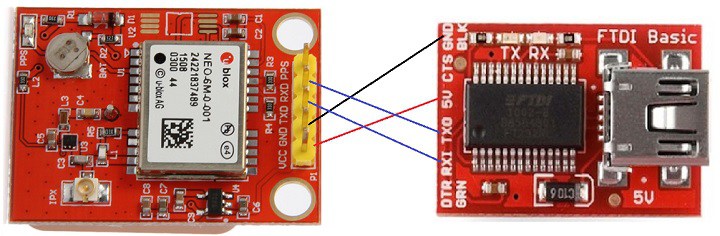
1. **Resultados**

Se han cumplido los objetivos originales, es decir, el sistema electrónico permite loguear toda la información de rastreabilidad satelital sin importar donde se encuentre instalado.

**Configuración del GPS:**

Se realizó la configuración del GPS utilizando el programa llamado “**u-center**” que nos provee la marca UBLOX. Dicho programa puede ser descargado desde su página oficial de ubox en forma gratuita junto con su respectiva documentación.

El módulo de GPS (UBLOX NEO 6M) no cuenta con interfaz USB, por lo tanto se utilizó un conversor de puerto serie a USB llamado comúnmente “FTDI”. Previamente al conexionado del módulo con la PC se bajaron los drivers mencionados en el apéndice

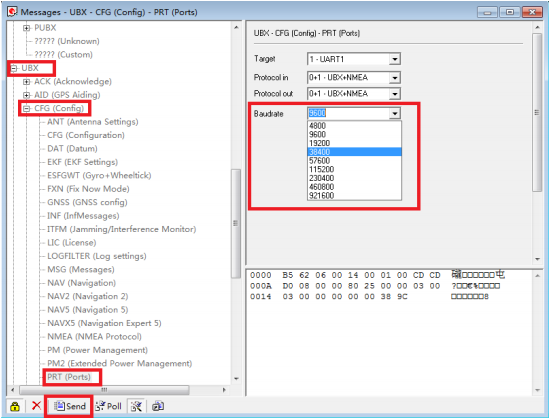
En la siguiente imagen podemos ver la conexión del módulo de GPS contra el de FTDI (derecha), cabe aclarar que para cualquier puerto serie cuando se interconectan 2 módulos se debe intercambiar los pines de E/S RX y TX,entonces para el GPS la salida TX se debe conectar en el FTDI en el pin RX y asimismo con el RX (GPS) a TX (FTDI).

Una vez instalado, se configurará de la siguiente forma:

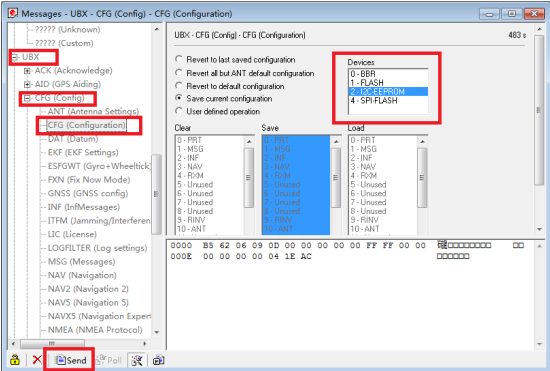
1. Se abre el programa **u-center** y se conecta dicho módulo (FTDI) vía USB a la PC.
2. Para que el GPS trabaje en forma correcta se modificará el Baud Rate y la salida de mensajes (muestras tomadas por dicho módulo).
   1. **Configuración del Baud Rate:**

El GPS por default viene configurado a 38.400 baudios. Para nuestro proyecto se modificó y configuró a 9.600 baudios para poder lograr compatibilizar el GPS con el microcontrolador en el envío de información.

Para ello, se seleccionó **View -> Messages View** y dentro de la configuración **UBX->CFG(Config)->PRT(Ports)** se selecciona nuestra elección (9.600 baudios) y se aprieta “Send” quedando así configurado dicho módulo.



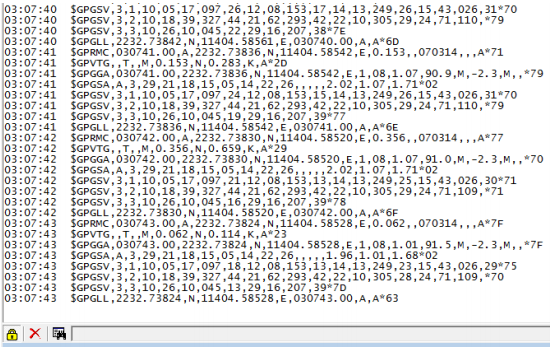
Por último, todos los cambios se guardarán en la memoria interna EEPROM del mismo GPS.

Para esto, se seleccionará en **UBX->CFG(Config)->CFG(Configuration)**, la opción **2-I2C EEPROM** y dando por último “Send” para el grabado del mismo.

* 1. **Configuración de mensajes:**

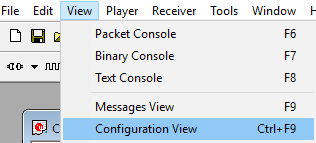
Dentro del software se configuró las opciones de salida que el GPS entregaría como información a utilizar.

Por default, el GPS (una vez inicializado correctamente) entrega distintos tramas de caracteres (utilizando el protocolo NMEA/UBX) como mensajes de salida cada vez que realiza un muestreo de la posición actual.

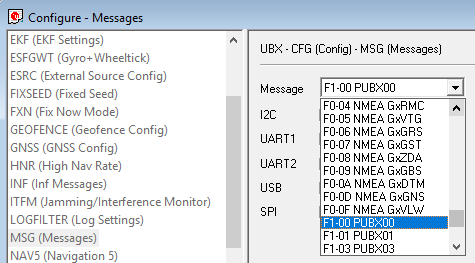


En nuestro caso solamente utilizaremos, de las muestras tomadas por el GPS, aquellos con el ID del mensaje “PUBX”. Dicha trama nos entregará la información que precisamos.

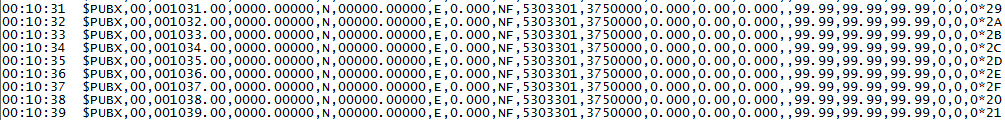
Internamente se realizó la configuración en **View -> Configuration View**



Dentro de la misma se desactiva todas las tramas de caracteres salvo la de “F1-00 PUBX00” que es la de navegación que utilizaremos.

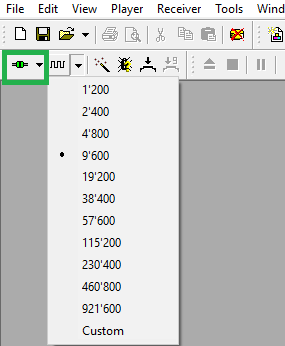
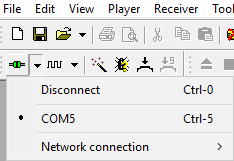


Dándonos así como resultado el siguiente mensaje de salida cada 1 segundo de muestreo, el cual utilizaremos en dicho proyecto.

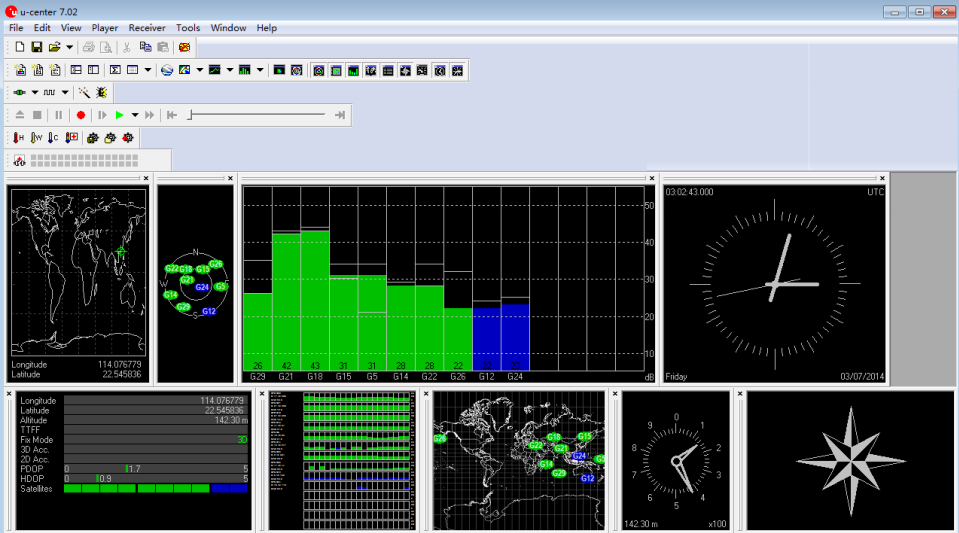


Se aplica nuevamente el guardado de dicha configuración en la **2-I2C EEPROM** del módulo (aplicando los mismos pasos ya mencionados anteriormente).

Por último, se analiza si el GPS se comunica correctamente con la configuración ya previamente grabada. Para ello, se lo conectará de la siguiente forma:



El icono  asegura que la configuración fue exitosa para después ser utilizado con el microcontrolador.



* ***Modo de Operación del GPS:***

El GPS NEO-6M es compatible y trabaja con el protocolo NMEA y uno propietario llamado UBX.



Una vez conectado a la interfaz serial y configurado como se dijo anteriormente, éste enviará cada 1 segundo una serie de comandos del tipo UBX por la salida serie y el microcontrolador lo recibiría con el siguiente formato:

$PUBX,00,001142.00,0000.00000,N,00000.00000,E,0.000,NF,5303301,3750000,0.000,0.00,0.000,,99.99,99.99,99.99,0,0,0\*2C

Cada comando comienza con el símbolo ‘$’ y finaliza con un checksum y salto de línea (line feed y return carriage). Es importante señalar que el GPS únicamente proporcionará datos válidos de manera de hardware si el led verde en el módulo del gps se encuentra encendiendo a lapsos de 1 segundos aproximadamente y en forma de software el campo número 9 del formato de salida nos entregará 2 caracteres según la tabla:

|  |  |
| --- | --- |
| Navigation Status | Description |
| NF | No Fix |
| DR | Dead reckoning only solution |
| G2 | Stand alone 2D solution |
| G3 | Stand alone 3D solution |
| D2 | Differential 2D solution |
| D3 | Differential 3D solution |
| RK | Combined GPS + dead reckoning |
| TT | Time only solution |

Se considera un fix válido en cualquier estado distinto al “NF” (No fix).

En caso de que el led no esté encendiendo y/o se de que la trama indique “NF” se puede deber a que el GPS no encuentra satelites disponibles ya que el sistema se puede encontrar operando dentro de un espacio cerrado (las paredes y edificios grandes obstruyen la señal captada por los satélites) en esos casos hay que apartar el equipo a alguna locación en donde se vea el cielo a simple vista.

***Ejemplo informativo de los 23 campos internos que posee el mensaje con ID “$PUBX”:***

**$PUBX,00,081350.00,4717.113210,N,00833.915187,E,546.589,G3,2.1,2.0,0.007,77.52,0.007,,0.92,1.19,0.7 7,9,0,0\*5F**

|  |  |
| --- | --- |
| **$PUBX** | ID del mensaje |
| **0** | Identificador de la propiedad dl mensaje |
| **2 081350.00** | Muestra tomada en tiempo UTC |
| **4.717.113.210** | Latitud 47 grados 17.11321′ |
| **N** | Norte |
| **833.915.187** | Longitud 08 grados 33.915187´ |
| **E** | Este |
| **546.589** | Altitud Estimada |
| **G3** | Status de Navegación |
| **2.1** | Precisión Horizontal Estimada |
| **2.0** | Precisión Vertical Estimada |
| **0.007** | Velocidad terrestre (km/h) |
| **77.52** | Cursor terrestre (Grados) |
| **0.007** | Velocidad vertical |
| **-** | Correcciones recientes del DGPS |
| **0.92** | HDOP, Precisión Horizontal |
| **1.19** | VDOP, Precisión Vertical |
| **0.77** | TDOP, Precisión de Tiempo |
| **9** | Número de GPS satélites usados en la solución de navegación |
| **0** | Número de GLONASS satélites usados en la solución de navegación |
| **0** | DR usados |
| **\*5B** | Checksum |
| **-** | Retorno de Acarreo y Final de Línea |

Para mayor información sobre los distintos comandos y especificaciones se anexa en la bibliografía el protocolo (UBX - página 69):

* ***Modo de Operación del LCD:***

El LCD trabaja con el integrado HD44780U, este integrado tiene la capacidad de mostrar en una pantalla led alfanumérica datos que le ingresa a través de una serie de 8 pines (con la posibilidad de utilizar solo 4) con 3 pines de control.

Mediante la pantalla display LCD se mostrará los distintos errores (en el caso que ocurran) del dispositivo.También mostrará el sector de memoria actual de la memoria SD (4 chars), la hora en formato UTC, el COG y el SOG.

* ***Modo de Operación del módulo SD:***

Una vez encendido el proyecto el sistema va a probar si puede leer la tarjeta de memoria, si no puede imprimirá un mensaje de error en el LCD. Si en algún momento ocurre una falla de escritura, se terminara el programa con un mensaje de error y se tendrá que reiniciar el dispositivo.

ADVERTENCIA: Cada vez que se inicia el programa SE PISAN LOS DATOS de la tarjeta SD. Por lo que es conveniente siempre BORRAR la tarjeta de memoria antes de iniciar el dispositivo. (por falta de tiempo no se pudo implementar otra forma de implementación).

Cada vez que el GPS toma una muestra de la posición actual, se almacena en un buffer en donde se guardan hasta 4 muestras, esto es porque cada muestra que manda el gps son de 23 campos con un máximo de caracteres de 115 bytes, y los bloques de la tarjeta SD comprada son de 512 bytes por lo que cabe un máximo de 4 tramas de gps en cada bloque, esto quiere decir que el programa no utiliza todo el tamaño del bloque.

* ***Modo de Operación del Microcontrolador (AVR328p):***

El microcontrolador es el encargado de leer la trama que envía el GPS por el puerto serie, interpetarla para mostrarla en el LCD, y enviarla a la memoria SD.

Tambien se encarga de imprimir los errores en el LCD si es que ocurren.

***6.5 Especificación del formato de Salida de la SD.***

El formato de salida especifica cómo guardamos la información del gps a la SD.

Se guardan 4 tramas UBX en cada sector de la SD, como el sector es de 512 bytes y las tramas suman a lo mayor 460 bytes el resto del espacio queda sin utilizar por ende al ubicar las tramas por sector se debería de buscar por $PUBX y el fin de línea “carriage return” y “line feed”. Al utilizar un contador (puntero) de memoria de 16 bytes se puede llegar a un máximo de (0xFFFF) 65535 sectores de la SD, con lo cual si cada sector contiene 4 tramas (y las tramas están configuradas cada 1 segundo) se podrá guardar la información un máximo de 72 horas aproximadamente (65535 muestras por segundo/3600=~ 72 horas de muestras por segundo).

1. **Conclusiones**

El presente trabajo de laboratorio nos permitió aplicar los conocimientos adquiridos en la materia para la investigación, diseño, construcción y depuración de un dispositivo electrónico digital utilizando un microcontrolador.

9.1 Posibles mejoras:

Hay varias mejoras para introducirle al proyecto que no pudieron ser implementadas por varios factores (falta de tiempo, difícil implementación en assembler, falta de presupuesto, etc). Las mejoras posibles pueden ser:

* Incluir un modulo de GSM para poder enviar cada determinado tiempo (por ejemplo 1 hora) un mensaje de texto con la trama o con datos importantes de la localización del vehiculo/paquete. También se podría hacer lo mismo pero utilizando con internet móvil (4G por ejemplo) y mandarlo a un host. Obviamente este proyecto en assembler seria mas difícil ya que no podríamos contar con las librerias que ya están hechas en C.
* Aumentar el tiempo de batería mediante la implementación de la función de sleep del microcontrolador. Asi la autonomía del dispositivo es mayor.
* Agrandar el tamaño del puntero al sector de la memoria SD, de forma tal que se puedan cubrir todos los sectores y no los primeros 65535 como esta implementado actualmente. De esta manera podríamos agrandar la cantidad de horas en la que el dispositivo puede guardar la información.
* Optimizar el espacio que se utiliza en la memoria SD, ya que con la actual implementación hay aproximadamente un 10% de la memoria que no se utiliza por sector.
* Tener un display de mayor tamaño (6x4 por ejemplo) para poder visualizar mayor información en tiempo real, o sino utilizar la función display shift del LCD.
* Utilizar el primer sector de la memoria para guardar información importante de configuración. Por ejemplo: en este momento el dispositivo siempre comienza a escribir desde el primer sector en adelante hasta que se apaga o se queda sin espacio. Implementando esta función se podría guardar el último sector donde se guardaron datos, de esta manera no se sobreescribiría la información cada vez que se reinicia el aparato.
* Utilizar una batería tipo Li-Po y un módulo cargador para generar un sistema completo sin elementos externos. Ya que actualmente dicho proyecto está en funcionamiento mediante una batería (power bank) externa.

Para las simulaciones se utilizó el software Atmel Studio 7.0. El aprendizaje y la utilización de este programa nos parece de vital importancia, no sólo para el desarrollo de este trabajo, sino para el desarrollo profesional.

Como conclusión final, podemos decir que este proyecto nos sirvió para conocer las diferentes etapas en la construcción de un prototipo, desde la formulación del proyecto, la investigación, el diseño y su construcción.

**11. Código Fuente**

Este se adjunta al trabajo en la carpeta Software y contiene los siguientes archivos:

* GPS.atsln //Proyecto de atmel studio
* GPS (Carpeta)
  + Makefile\_linux
  + delay.s //rutinas de delay de distintos tiempos
  + GPS.componentinfo.xml
  + GPS.cproj
  + lcd.s //rutinas del LCD
  + main.s //main del programa
  + SD\_routines.c //libreria de rutinas de la memoria SD
  + SD.s //rutinas de la memoria SD
  + USART.s //rutinas de comunicación UART y GPS
  + Debug (Carpeta)
    - (archivos de compilación varios)
    - GPS.hex (programa que se manda al microcontrolador)

Para compilar en linux se adjunta el makefile llamado “Makefile\_linux” en donde se debe de programar metiendo el makefile renombrandolo como Makefile dentro de la carpeta Debug y ahí ejecutando “make flash”.

Si utilizamos Linux se debe de utilizar el programa avrdude ejecutandolo desde consola “cmd” como: “avrdude.exe -p m328p -c usbtiny -U flash:w:GPS.hex:i”.

**12. Apendice**

**12.1. Cálculo del Baud Rate (Velocidad de Transmisión)**

Para el cálculo del Baud Rate para la comunicación por puerto serie se realizó el siguiente cálculo:

En primer lugar, se encontró el valor de UBRR para un valor de 103 (en decimal) a una frecuencia de oscilación de 16Mhz.

Aplicando la fórmula: UBRR = ( fosc /(16 \* BAUD) ) – 1 **= 103,1667**

Este valor se redondea a 103. Dicho valor se coloca en la ecuación inicial y se obtiene el BAUD como:

BAUD = fosc / 16 \* (UBRR + 1) **= 9615,384 bps**

Se calcula el error como:

Error (%) = ((BAUDcalc/BaudRate) - 1) \* 100

Error (%) = ((9615,384 bps / 9600 bps) - 1) \* 100

**Error (%) = 0,16**

Se recomienda que el porcentaje de error sea menor al 1%. Dado que en este caso se cumple, se utilizó un cristal de **16MHz**.

**12.2. Bibliografía**

* Mazidi, Muhammad & Naimi, Sarmad. The AVR Microcontroller and Embedded System. Editorial Pearson Prentice Hall
* Campligio, Guillermo. Filminas de la materia 66.09 Laboratorio de Microcomputadoras
* Descripcion del protocolo UBX del GPS:

<https://www.u-blox.com/sites/default/files/products/documents/u-blox6_ReceiverDescrProtSpec_%28GPS.G6-SW-10018%29_Public.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2Fu-blox6_ReceiverDescriptionProtocolSpec_%28GPS.G6-SW-10018%29.pdf>

* SD Specifications Part 1:

<http://users.ece.utexas.edu/~valvano/EE345M/SD_Physical_Layer_Spec.pdf>

* Hojas de Datos:
  + Datasheet Chip Atmega328P
  + Datasheet LCD Hitachi HD44780U (LCD-II)
  + Datasheet GPS Ublox 6M.

**12.3. Software utilizado**

* Atmel Studio 7.0
* Eagle 8.2: Utilizado para el esquemático.
* U-center: Utilizado para configurar el GPS.
* Avrdude/Winavr: utilizado para enviar el programa al microcontrolador
* Drivers FTDI: utilizado para leer y escribir en el puerto serie con USB

<http://www.ftdichip.com/FTDrivers.htm>