

# APLICACIONES CON MINICOMPUTADORES RASPBERRY PI PROVISTO DE MÓDULO GPS Y ACELERÓMETRO PARA CONTROL DE VELOCIDAD Y POSICIONAMIENTO. (Noviembre 2013)

Fernando Chavez<sup>(1)</sup>, Christian Yuquilema<sup>(2)</sup>, Carlos Valdivieso<sup>(3)</sup>

Facultad de Ingeniería en Electricidad y Computación<sup>(1)(2)(3)</sup>

Escuela Superior Politécnica del Litoral (ESPOL)<sup>(1)(2)(3)</sup>

Campus Gustavo Galindo, Km 30.5 Vía Perimetral, Apartado 09-01-5863. Guayaquil, Ecuador<sup>(1)(2)(3)</sup>

fchavez@armada.mil.ec<sup>(1)</sup>, cyuquile@espol.edu.ec<sup>(2)</sup>, cvaldiv@fiee.espol.edu.ec<sup>(3)</sup>

## Resumen

El presente trabajo de investigación pretende realizar el control de velocidad y posicionamiento de un móvil mediante el uso del minicomputador moderno Raspberry Pi, el cual está provisto de un módulo GPS y acelerómetro para poder medir las variables antes señaladas.

Puntal fundamental para el desarrollo de este trabajo es la programación del minicomputador Raspberry Pi, y mediante la utilización del lenguaje de programación llamado Python, se ha logrado recibir y procesar los datos enviados desde el módulo GPS y acelerómetro.

La adquisición de datos desde el Sistema de Posicionamiento Global, hacia la mini computadora Raspberry Pi se lo ha realizado a través del puerto GPIO, el cual es un puerto de entrada y salida de propósitos generales. La adaptación del módulo GPS hacia el minicomputador se lo ha implementado mediante una tarjeta puente. Cabe mencionar que los datos enviados por el módulo GPS cumplen con el protocolo NMEA 0183.

### Palabras Claves:

Raspberry-Pi, GPS, Python, GPS CSC3 Vincotech.

## Abstract

The aim of this study is to evaluate the speed and position of a mobile using the modern Raspberry Pi minicomputer equipped with a GPS module and accelerometer to measure the variables mentioned above.

The key to execute this study is the programming of the minicomputer Raspberry Pi and through the use of a program language called Python, it has been possible to receive and process the data sent by the GPS module.

The acquisition of data from the Global Positioning System to the Raspberry Pi was made through the GPIO port, which is an general purposes input and output port. The GPS module adaptation to the minicomputer was implemented by the use of a bridge card. It is important to mention the GPS module sends data compliant with the NMEA 0183 protocol.

### Keywords:

Raspberry-Pi, GPS, Python, GPS CSC3 Vincotech..

## 1. Introducción.

Este proyecto se basa en el control de velocidad y posicionamiento de un móvil mediante la utilización de un módulo GPS y acelerómetro que se encuentra

conectado a una minicomputadora Raspberry Pi, para lo cual se ha elaborado un software que permite la adquisición y procesamiento de los datos antes indicados para su presentación en una interfaz hombre máquina implementado en el lenguaje Python.

## 1.1. Antecedentes.

El reto que se ha planteado para la ejecución del presente proyecto es realizar el control de posición y velocidad de un vehículo mediante la utilización del minicomputador Raspberry Pi, el mismo que interactúa con un módulo GPS para obtener las variables deseadas.

Esta minicomputadora soporta un sistema operativo desarrollado en Linux, lo que facilita la programación y soporte durante el desarrollo, especialmente por la existencia en la web de aplicaciones gratuitas que son de gran utilidad para ejecutar proyectos a un costo relativamente bajo.

La principal potencialidad de este minicomputador es su capacidad de comunicación con periféricos a través de sus diferentes puertos y en especial por el puerto de propósitos generales GPIO, que permite realizar aplicaciones para el control y manejo de dispositivos.



**Figura 1.1. Tarjeta Minicomputador Raspberry Pi Modelo B.**

## 1.2. Identificación del Problema.

Mediante este proyecto se desea alcanzar el conocimiento necesario para realizar el manejo del puerto de entrada y salida de propósitos generales del minicomputador Raspberry Pi, a fin de realizar el control de posición y velocidad de un móvil, utilizando para el efecto una interface hombre máquina para la presentación de los datos en mención.

Estos datos se obtendrán del módulo GPS y acelerómetro, lo que implica un conocimiento suficiente de este hardware para manejar sus entradas y salidas de datos.

De igual manera se requiere analizar y entender el protocolo de comunicaciones NMEA 0183 que emplea

el módulo GPS que se interconectará con el minicomputador.

Finalmente, una vez que se logre controlar y obtener la data desde el GPS, se desarrollará una HMI utilizando el lenguaje de programación Python para presentar los datos de posición y velocidad de un móvil, concretándose de esta manera el objetivo de esta tesina.

## 1.3 Objetivos Principales

- Realizar el control de posición y velocidad de un móvil, mediante la adquisición de datos desde el módulo GPS y a través del puerto de propósitos generales GPI del minicomputador Raspberry Pi.
- Conocer el funcionamiento del módulo GPS y su protocolo de comunicación a fin de interpretar la información que entrega mencionado módulo para la determinación de la posición y velocidad del móvil en tiempo real.
- Aprender el manejo de la herramienta Cutecom en modo local, el cual permitirá demostrar que la comunicación entre el módulo GPS y el minicomputador se encuentra correctamente establecida.
- Desarrollar capacidades para implementar interfaces Hombre-máquina, mediante la utilización del lenguaje Python, el mismo que permitirá realizar la visualización de los datos que ingresan en tiempo real a nuestro minicomputador, a través del puerto GPIO.

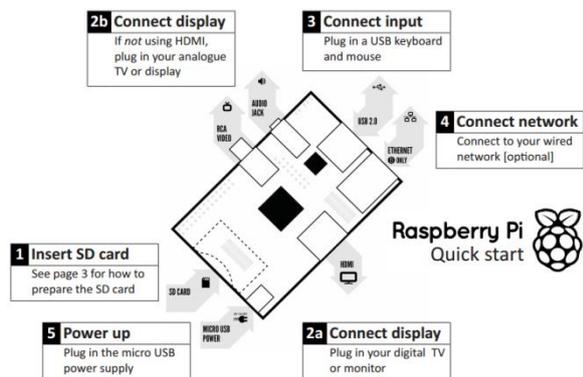
## 2. Fundamentos Teórico.

### 2.1. Minicomputador Raspberry – Pi

Raspberry-Pi es un minicomputador que posee un procesador central (CPU) ARM1176JZF-S a 700 MHz y una memoria de 512 MB de RAM. Posee 02 puertos USB, 02 salidas de video (RCA y HDMI), salida de audio, conectividad de red mediante el conector RJ-45 y ranura para la instalación de un dispositivo de almacenamiento del tipo SD.

Es una tarjeta de bajo consumo aproximadamente 1 Amp y requiere de un voltaje de alimentación de 5 voltios. Soporta lenguajes operativos tipo Linux. Esto implica que tiene la capacidad de realizar procesamiento gráfico, permitiendo la comunicación

con varios tipos de hardware. Se recomienda la instalación del sistema operativo llamado Raspbian.



1

Figura No. 2.1: Diagrama en Bloques de la Tarjeta Minicomputador Raspberry Pi Modelo B. [1]

### 2.1.1 Configuración del Raspberry Pi para Recepción de Data desde el módulo GPS.

El lenguaje de programación Python que se ha seleccionado para programar el Raspberry nos permitirá manejar el puerto de propósitos generales, el mismo que está compuesto de 26 pines que pueden ser configurados como entradas o salidas.

Para el caso de esta aplicación se conectará a éstos pines la tarjeta interface del módulo GPS, el mismo que nos enviará la información de posición y velocidad.

El puerto GPIO se encuentra ubicado en la parte superior izquierda del integrado, sus 26 pines se encuentran distribuidos en dos columnas de 13 pines cada una. Cada pin tiene su propósito específico el mismo que puede ser visualizado la figura 2.2.

En los pines 8 y 10, existe un TX/RX asincrónico UART, el mismo que consiste en un puerto serial, donde el pin 8 es utilizado para transmisión de data y el pin 10 para la recepción. Este puerto serial es configurado mediante la modificación del archivo cmdline.txt.

Para comunicarse con el módulo GPS GSC3 de la empresa Vicontech se debe setear una velocidad de transmisión de 4800 baudios.

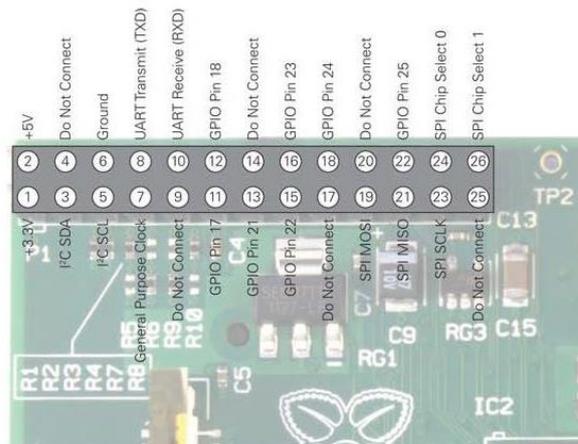


Figura No. 2.2: Distribución de los Pines del Puerto GPIO del Raspberry Pi Modelo B. [1]

### 2.1.2 Instrucciones en Python para programación del UART.

Para trabajar con el puerto serial se debe realizar la instalación de la librería Pyserial, para lo cual se debe descargar mencionada librería de la web y realizar el siguiente procedimiento:

Para descargar de la web la librería escribir:

```
http://sourceforge.net/projects/pyserial/files/latest/download?source=files
```

Para extraer el contenido del archivo escribir:

```
tar xvf pyserial-2.6.tar.gz
```

Para instalar la librería escribir:

```
Sudo python setup.py install
```

De igual manera se debe importar este módulo al programa mediante la instrucción:

```
import serial
```

Para establecer una comunicación standard con un baudrate de 4800, 8 bits, sin paridad y 01 bit de parada es suficiente con escribir las siguientes instrucciones:

```
ser=serial.Serial ()
ser.baudrate=4800
ser.port=0
ser
```

Para almacenar la data que es leída en el puerto del UART se puede utilizar la sentencia:

```
ser.open()  
GPS = ser.readline()
```

## 2.2 Protocolo de Comunicaciones NMEA 0183.

La asociación “National Electronics Association (NMEA)”, es una asociación sin fines de lucro de los fabricantes, distribuidores, institutos de educación y otros interesados en equipos periféricos relacionados con el área marítima. El protocolo NMEA 0183 se lo define como una interface eléctrica y protocolo de datos para la comunicación efectiva entre los diferentes equipos del área marítima

### 2.2.1 Interface Eléctrica del Protocolo.

Este protocolo está diseñado tanto para los equipos que transmiten como los que reciben data y emplea una comunicación asincrónica mediante la utilización de una interface serial. Esta interface debe ser configurada con los siguientes parámetros: Baudrate, Número de bits, Bit de Parada, Paridad y Handshake.

NMEA 0183 permite que existan una sola fuente de transmisión (Talker) y varias fuentes de recepción (listeners) en un mismo circuito. La conexión recomendada es a través de conductores que tenga protección electromagnética especialmente conductores de par trenzado con malla, la misma que debe estar aterrizada solamente del lado del Talker.

No se requieren conectores especiales sin embargo es recomendable que el talker cumpla con el standard RS-422 para la transmisión de datos, el mismo que es un sistema diferencial que utiliza dos líneas. Este es un formato más inmune al ruido y debe ser aterrizado. De igual forma se recomienda que al realizar la interconexión de dispositivos que cumplan con este protocolo, se instale en el receptor opto acopladores para establecer una protección al circuito.

## 2.3 Descripción de los Sistemas de Posicionamiento Satelital.

El SPG o GPS (Global Positioning System: sistema de posicionamiento global) o NAVSTAR-GPS es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión

hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de 24 satélites en órbita sobre el planeta tierra, a 20.200 km, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos.

Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, de tal modo que mide la distancia al satélite mediante triangulación, la cual se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites.

Mediante la triangulación, cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en el propio satélite y de radio equivalente a la distancia total hasta el receptor.

Si se obtiene la información de dos satélites queda determinada una circunferencia que resulta cuando se intersecan las dos esferas en algún punto de la cual se encuentra el receptor.

Mientras que si se dispone la información de un cuarto satélite, se elimina el inconveniente de la falta de sincronización entre los relojes de los receptores GPS y los relojes de los satélites. Y es en este momento cuando el receptor GPS puede determinar una posición 3D exacta (latitud, longitud y altitud).

## 2.4 Descripción del Firmware del Módulo GPS CSC3 Vincotech.

El módulo GPS CSC3 de Vincotech soporta una interface serial bidireccional. Esta es implementada utilizando la interface del procesador del GPS, utilizando una comunicación full dúplex del tipo UART.

La configuración por defecto de este módulo es una comunicación por puerto serial con los siguientes parámetros: 4800 baudios, 8 bits, No Paridad y 1 Bit de parada, no flujo de control.

El módulo en mención soporta 08 sentencias en formato NMEA 0183 relacionadas con la información que entrega el GPS, y son las siguientes:

- \$GPGGA (default: ON)
- \$GPVTG (default: OFF)
- \$GPRMC (default: ON)
- \$GPGSA (default: ON)
- \$GPGSV (default: ON, 0.2Hz)
- \$GPGLL (default: OFF)

#### 2.4.1 Mensajes de Inicialización del GPS.

Después del encendido del módulo GPS o reset del módulo, este dispositivo enviará el primer mensaje de inicialización en el cual se podrá apreciar las siguientes sentencias:

```
$PSRFTXT,Version: GSW3.5.0_3.5.00.00-C25P2.01
*03
$PSRFTXT,TOW: 0*25
$PSRFTXT,WK: 1517*67
$PSRFTXT,POS: 6378137 0 0*2A
$PSRFTXT,CLK: 96250*25
$PSRFTXT,CHNL: 12*73
$PSRFTXT,Baud rate: 4800*65
```

Estos mensajes tiene información de la versión del firmware del GPS, tiempo del GPS, posición entre otros. Esta información realmente no debe ser utilizada durante la implementación del programa por cuanto es únicamente informativa.

#### 2.4.2 Sentencias propietarias del Módulo GPS.

El módulo GPS permite la habilitación de mensajes de entrada en formato NMEA. Por defecto en el puerto 0 es configurado como modo NMEA. Los mensajes pueden ser enviados mediante la utilización de un terminal de programa. Los siguientes mensajes de entrada son soportadas:

Message	MID <sup>(1)</sup>	Description
Set serial port	100	Set Port 0 parameters and protocol
Reset Configuration	101	Initialize various start up behaviors
Query/rate control	103	Query standard NMEA message and/or set out-put rate
Development data	105	Development Data

On/Off		messages On/Off
Select Datum	106	Selection of datum to be used for coordinate transforming

#### (1) Message Identification (MID)

**Tabla No. 1: Mensajes de Entrada NMEA al Módulo GPS. [5]**

#### 2.4.3 Configuración del Puerto Serial del Módulo GPS.

El comando que es utilizado para configurar los parámetros de baud rate, data bits, stop bits y paridad es el siguiente:

```
$PSRF100,0,9600,8,1,0*0C
```

Name	Example	Description
Message ID	\$PSRF100	PSRF100 protocolheader
Protocol	0	0 SiRfbinary / 1 NMEA
Baud	9600	4800, 9600, 19200, 38400, 57600, 115200
DataBits	8	8, 7 <sup>(1)</sup>
StopBits	1	0, 1
Parity	0	0 none / 1 odd / 2 even
Checksum	*0C	End of message termination

**(1) SiRF protocol is only valid for 8 data bits, 1 stop bit and no parity**

**Tabla No. 2: Configuración del Puerto Serial [5]**

#### 2.4.4 Selección de las Sentencias que enviará el Módulo GPS.

Se requiere de comando especiales para habilitar o deshabilitar las diferentes sentencias que son enviadas por la salida serial del GPS. Como se mencionó anteriormente este módulo GPS en especial tiene la capacidad de enviar los mensajes GGA, GLL, GSA, GSV, RMC y VTG. Se puede de igual forma habilitar o no el envío del checksum en caso de ser necesario.

A continuación las instrucciones que son necesarias para habilitar el checksum y habilitar el envío de los datos de velocidad que por defecto se encuentra deshabilitada:

1. Mensaje GGA con checksum habilitado

- \$PSRF103,00,01,00,01\*25

2. Mensaje VTG para salida a 1 Hz con checksum habilitado

• \$PSRF103,05,00,01,01\*20

La interpretación de estos mensajes se puede visualizar en las siguientes tablas:

Name	Example	Units	Description
Message ID	\$PSRF103		PSRF103 protocolheader
Msg	00		Ver Tabla 4.
Mode	01		0=SetRate, 1=Query
Rate	00	seconds	Output rate 0 off Max 255
CksumEnable	01		0 Disable Checksum 1 EnableChecksum
Checksum	*25		End of messageterminati on

Tabla 3: Formato de la Sentencia de Control [5]

Value	Description
0	GGA
1	GLL
2	GSA
3	GSV
4	RMC
5	VTG
6	MSS (If internal beacon is supported)
7	Notdefined
8	ZDA (if 1PPS output is supported)
9	Notdefined

Tabla 4: Valor asignado a cada Mensaje de Control [5]

### 3.DISEÑO E IMPLEMENTACIÓN DEL PROYECTO.

#### 3.1 Conexión de Componentes.

Para realizar la conexión de los componentes, fue necesario identificar los pines de entrada y salida del módulo GPS, para lo cual con la ayuda de un osciloscopio, se determinó que el dispositivo se encontraba trabajando correctamente. En la figura 3.1, se puede apreciar la prueba realizada.



Figura No. 3.1: Determinación del Funcionamiento del Módulo mediante un Osciloscopio.

Los pulsos que se pueden visualizar en la Figura 3.1, corresponden a la data que está saliendo del dispositivo GPS. El tipo de transmisión de datos es serial, por lo que los pines del UART (Rx – Tx) se deberán conectar a los pines del Raspberry Pi, además de la alimentación respectiva que necesita el mencionado módulo.

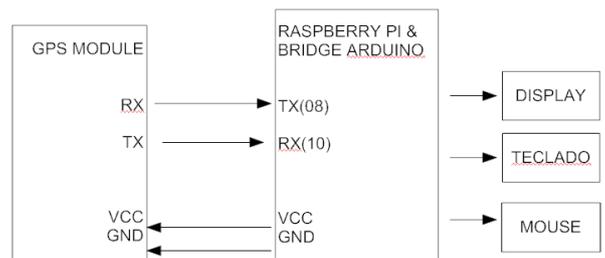


Figura No 3.2. Diagrama Básico de Interconexión.

En la figura 3.3 se puede visualizar la interconexión física de estos módulos:

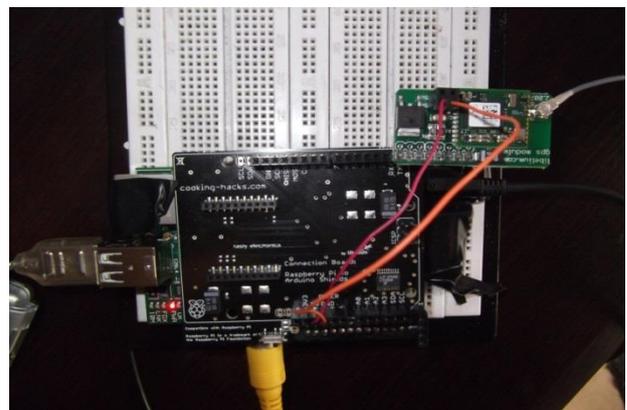
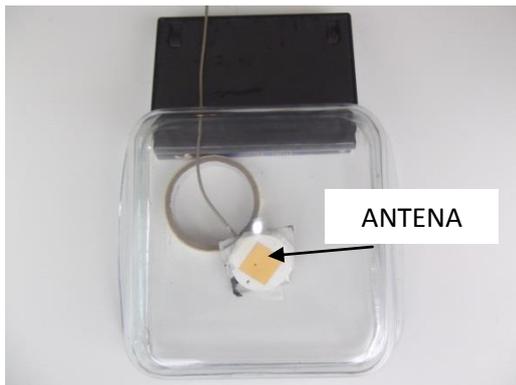


Figura No 3.3. Interconexión Física de Componentes.

Se debe tomar la precaución, que para recibir la información de los satélites, se requiere que la antena GPS se encuentre al exterior del edificio, sin embargo es factible colocar la antena en la cercanía de una ventana, a pesar que el vidrio puede degradar levemente las señales recibidas por los GPS, después de un cierto tiempo se consigue recibir señal desde los satélites. En la figura 3.4 se puede apreciar las pruebas de recepción realizadas con un pyrex de 5mm de grosor, con el cual se obtuvieron resultados exitosos.



**Figura No 3.4. Pruebas de Recepción con Vidrio de 5mm de Espesor.**

Se necesita al menos 03 señales de satélites con niveles de señal ruido sobre 18 para que el módulo pueda calcular los valores de posición y velocidad de un blanco.

### 3.2 Configuración del Raspberry para conexión con Módulo GPS y Simulador.

Una vez que el dispositivo que se encuentra conectado, este debe ser seteado vía comandos en el terminal de Linux. Las direcciones físicas de los dispositivos en mención son:

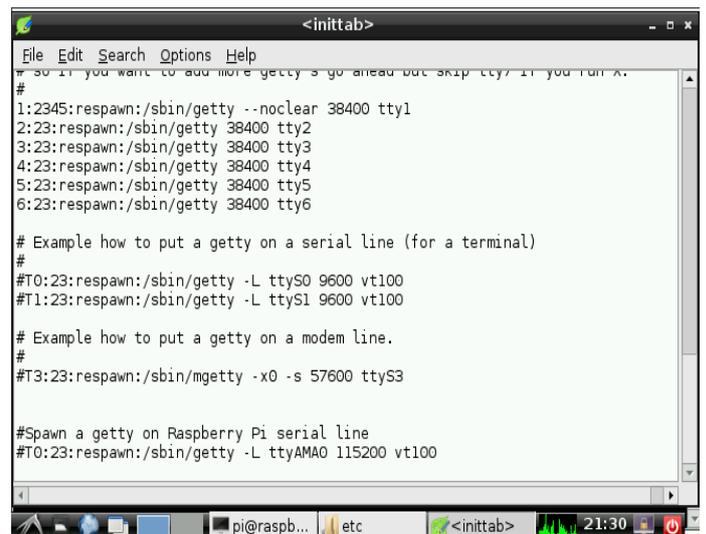
- 1.- MODULO GPS: `//dev/ttyAMA0`
- 2.- SIMULADOR GPS: `//dev/ttyUSB0`

Para poder establecer esas direcciones se debe modificar el archivo `/boot/cmdline.txt`, debiéndose visualizar la siguiente instrucción en el mencionado archivo.



**Figura No 3.5 . Edición del archivo cmdline.txt**

Mientras tanto en el archivo `/etc/inittab`, se debe comentar todo lo relacionado con el puerto serial. Debiéndose visualizarlo de la siguiente manera:



**Figura No 3.6. Edición del archivo Initab.txt**

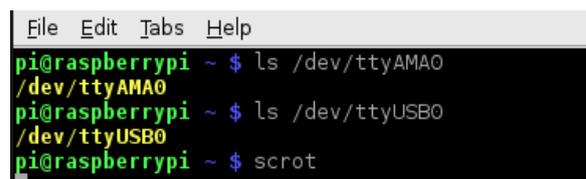
Es necesario mencionar que estos archivos son protegidos, razón por la cual se debe dar los privilegios de escritura, vía comando:

```
sudo chmod 666 <ruta y nombre del archivo>
```

Posteriormente se puede visualizar si los puertos están disponibles y reconocidos mediante la instrucción:

```
ls /dev/ttyUSB, para el caso del simulador y ls /dev/ttyAMA0 para el módulo GPS.
```

En la figura 3.7, se puede visualizar el resultado de estas instrucciones:

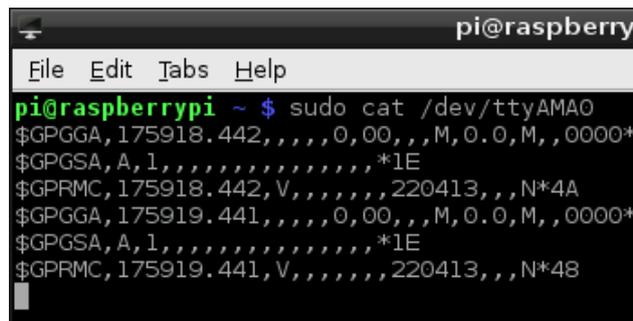


**Figura No 3.7 . Resultado del Comando ls en Consola**

Para ver rápidamente lo que está ingresando desde el GPS conectado se puede ingresar el siguiente comando, dependiendo del puerto en el que se encuentre el dispositivo:

```
sudo cat /dev/ttyAMA0 o sudo cat /dev/ttyUSB0
```

A continuación el resultado del comando “cat” en el terminal.



```
pi@raspberrypi ~ $ sudo cat /dev/ttyAMA0
$GPGGA,175918.442,,,,,0,00,,M,0.0,M,,0000*
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPRMC,175918.442,V,,,,,220413,,N*4A
$GPGGA,175919.441,,,,,0,00,,M,0.0,M,,0000*
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPRMC,175919.441,V,,,,,220413,,N*48
```

Figura No 3.8. Resultado del Comando “cat” en el Terminal

### 3.3 Instalación del Software GPS Daemon (gpsd).

El software GPS Daemon (gpsd), es el software mediante el cual vamos a poder entender la data que está enviando el GPS. Este software envía la información ordenada de tal manera que la interpretación de la información del blanco se lo hace de una manera sencilla.

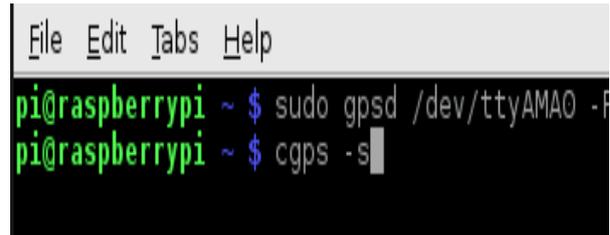
Este programa, como la mayoría de los desarrollados en Linux se encuentra de manera gratuita y se lo instala mediante la siguiente instrucción:

```
Sudo apt-get install gpsd-clients python-gps.
```

Con estas instrucciones estamos habilitando tanto el software GPSD y también la librería GPS de python, la misma que la utilizaremos más adelante.

Dependiendo del puerto en el que se encuentre nuestro dispositivo GPS, debemos habilitarlo para la lectura mediante el siguiente comando:

```
Sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock
```



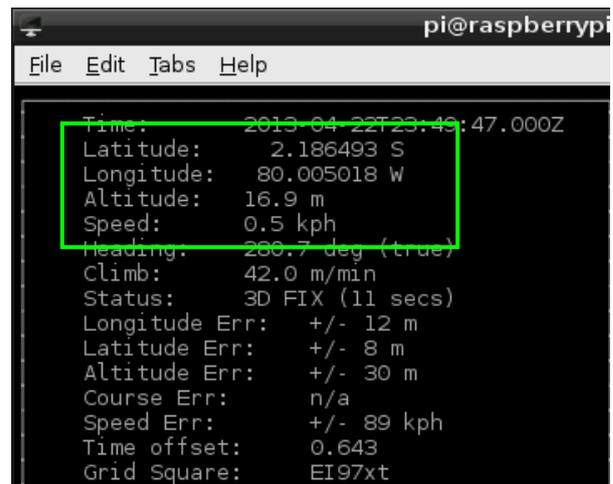
```
File Edit Tabs Help
pi@raspberrypi ~ $ sudo gpsd /dev/ttyAMA0 -F
pi@raspberrypi ~ $ cgps -s
```

Figura No.3.9: Comandos para ejecución del Software GPSD

Una vez que se tiene instalado el mencionado software, lo ejecutamos mediante el comando:

```
cgps -s
```

Como se puede visualizar en la figura 3.13, se despliega la información que entrega de manera ordenada el GPS. En esta figura se puede visualizar la información de posición y velocidad del blanco, objetivo del presente proyecto:



```
File Edit Tabs Help
Time: 2013-04-22T23:49:47.000Z
Latitude: 2.186493 S
Longitude: 80.005018 W
Altitude: 16.9 m
Speed: 0.5 kph
Heading: 280.7 deg (true)
Climb: 42.0 m/min
Status: 3D FIX (11 secs)
Longitude Err: +/- 12 m
Latitude Err: +/- 8 m
Altitude Err: +/- 30 m
Course Err: n/a
Speed Err: +/- 89 kph
Time offset: 0.643
Grid Square: EI97xt
```

Figura No. 3.10: Visualización de datos en el Software GPSD

Cabe mencionar que este programa puede ser utilizado únicamente cuando se tiene recepción de datos reales de GPS, por lo que para la utilización de simuladores se debe realizar una programa en Python, que permita interpretar la data simulada, la misma que es muy similar a la data real. La única diferencia es que no existe información de los satélites que entregan información, razón por la cual el software GPSD identifica que la data recibida no es válida.

### 3.4 Adquisición de datos desde el módulo GPS utilizando Python

A pesar que ya se ha logrado controlar la posición y velocidad de un blanco mediante el programa “GPSD”, se ha considerado pertinente desarrollar el código requerido para poder monitorear la data del GPS, mediante la utilización de un programa desarrollado en Python, el mismo que servirá para futuras implementaciones. Para realizar el desarrollo del software nos guiaremos del siguiente algoritmo, el mismo que será la base para la implementación.

#### Descripción del Algoritmo

1. Inicialización de los puertos
2. Definición de puertos a utilizar.
3. Ingreso en un lazo infinito
4. Captura de los datos entregados por el módulo GPS.
5. Si el dato recibido corresponde a una sentencia GGA o RMC, procesa la información y los muestra en la pantalla



**Figura No 3.11. Algoritmo para la Implementación del Software.**

Cabe mencionar que en Python, al igual que en otros lenguajes es necesario utilizar librerías que contienen comandos especiales que permiten el desarrollo del software de una manera más sencilla. Para el caso del

manejo del puerto UART del Raspberry Pi es necesario instalar y utilizar la librería serial. La instalación de esta librería fue explicada anteriormente.

De igual forma se requiere la instalación de la librería respectiva para la elaboración de la interfaz gráfica. La librería más utilizada es la denominada Tkinter y la forma de instalarla es mediante código en el terminal mediante la siguiente instrucción:

```
sudo apt-get install python-tk
```

Una vez que se instala esa librería se debe llamar a la misma en el inicio del programa, para que se habiliten todas sus funciones.

El programa desarrollado en Python, se compone de dos módulos internos, el primero que corresponde a la adquisición de datos y procesamiento de los mismos y el segundo que corresponde a la interfaz gráfica (HMI).

## 4. PRUEBAS Y SIMULACIONES

### 4.1 Procedimientos de Encendido e Inicialización.

Para llevar a cabo la adquisición de datos GPS utilizando el simulador o directamente desde el módulo GPS se deben realizar el siguiente procedimiento:

1. Se debe verificar las alimentaciones necesarias para realizar el proceso de encendido de los equipos. Para el caso del simulador, el voltaje requerido es de 115 V ACC, mientras que el Raspberry Pi requiere una alimentación de 5 Vdc. Para las pruebas que se realizarán en el auto se utilizará la alimentación desde el encendedor de cigarrillos.
2. Para el caso del simulador no se requiere la instalación de una antena, sin embargo para la prueba con el módulo GPS, se requiere que la antena se encuentre en un lugar donde reciba las señales entregadas por los satélites. Si la antena va a ser colocada al interior del auto, ésta debe estar ubicada en las cercanías del parabrisas.
3. Una vez que se realice la conexión del simulador o del módulo GPS, se debe verificar que el software Cutecom se encuentra recibiendo correctamente la información, para lo cual se

deberá tener precaución de la velocidad de transmisión de datos, así como la dirección del dispositivo.

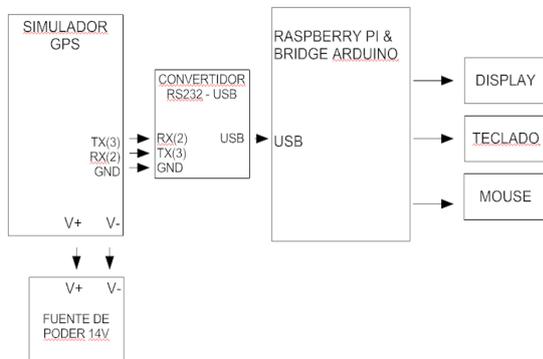
4. Luego de haber probado la comunicación con el dispositivo GPS, se procede a ejecutar en el caso del módulo GPS, el software de monitoreo GPSD, el mismo que permite visualizar en tiempo real la información recibida por los satélites. También se puede realizar las pruebas con el software desarrollado en Python.
5. En el caso del Simulador, se requiere la utilización del software desarrollado en Python, tomando la precaución de primero ejecutar el programa y luego el simulador para evitar que el buffer del simulador se sature de datos que son enviados mediante la secuencia FIFO.

#### 4.2 Pruebas realizadas con el Simulador GPS.

Siguiendo las instrucciones antes indicadas y luego de realizar la conexión adecuada del simulador al mini computador al Raspberry procedemos a realizar la simulación de la data para verificar que efectivamente estamos recibiendo la información y ésta es procesada por nuestro software.

Se debe recordar que para la prueba con el simulador no se puede ejecutar el software GPSD, por cuanto éste necesita información de los satélites para verificar si la data envía es válida o no.

La conexión del Simulador, hacia el Raspberry se lo va a realizar mediante el puerto RS-232, razón por la cual se debe verificar los pines de este simulador, los mismos que irán a un convertidos RS232 a USB. A continuación una figura de la conexión del simulador.



**Figura No. 4.1: Diagrama en Bloques de Conexión del Simulador de GPS**

Este diagrama en bloques ha sido implementado y en la figura 4.2 se puede visualizar la conexión de cada uno de sus componentes.

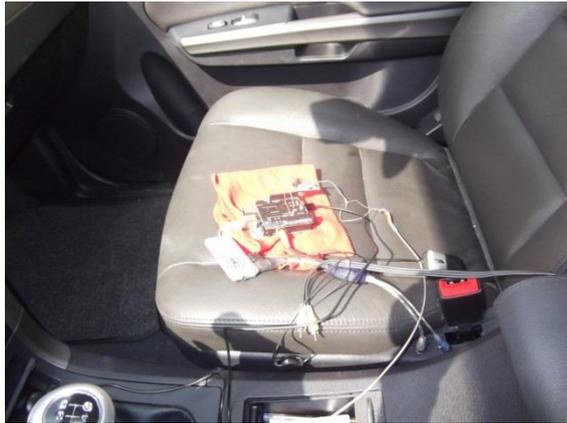
Cabe mencionar que este simulador de GPS, puede también funcionar como un GPS normal, por lo que se debe seleccionar en el menú respectivo la función de simulación para que el mismo entregue la data requerida.



**Figura No. 4.2: Interconexión Física del Simulador de GPS.**

#### 4.3 Pruebas realizadas con el módulo GPS en un vehículo.

Una vez que se ha realizado las pruebas con el simulador, procedemos a instalar el dispositivo en un vehículo para monitorear la velocidad y posición del auto en tiempo real. El poder de alimentación para el Raspberry Pi, se lo toma del encendedor de cigarrillos, teniendo la precaución de conectar el dispositivo una vez que el auto se encuentra arrancado, para evitar que existan picos de corriente que pueden dañar al dispositivo. En las figuras 4.9, 4.10 y 4.11, se puede ver la instalación del dispositivo en el automóvil.



**Figura No. 4.3 Conexión del Raspberry-Pi en el vehículo.**



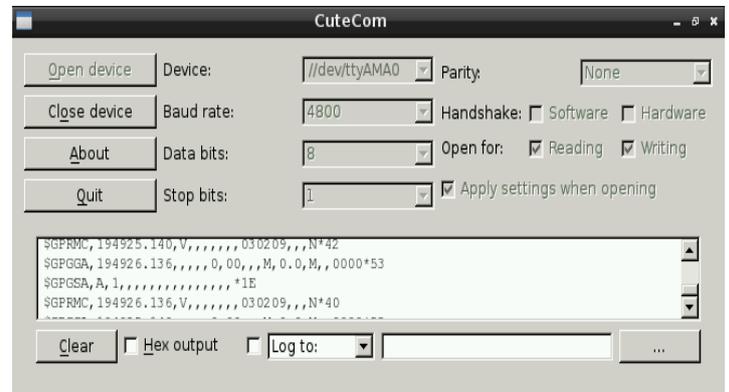
**Figura No. 4.4 Instalación de antena en el interior del vehículo.**



**Figura No. 4.5 Instalación de display en el interior del vehículo.**

Como se puede visualizar la antena es colocada detrás del parabrisas, el mismo que tiene la transparencia necesaria para captar las señales de los satélites.

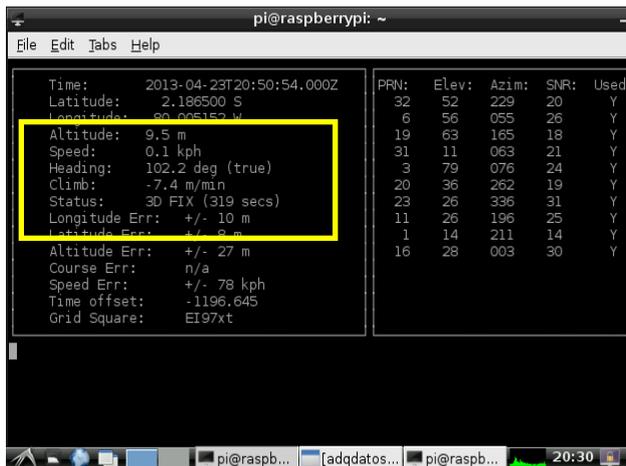
Posteriormente ejecutamos el programa CuteCom para visualizar si ha sido reconocido el módulo GPS. Se debe tener precaución de seleccionar bien el puerto mediante el cual se recibirá la data. Mencionado puerto es //dev/ttyAMA0. A continuación una gráfica de lo antes indicado:



**Figura No. 4.6 Adquisición de data del módulo GPS mediante el Software Cutecom.**

Como se puede visualizar, el dispositivo se encuentra conectado y enviado información al Raspberry PI, sin embargo no está enviando datos de posición y velocidad del blanco, esto es debido a que el dispositivo toma alrededor de 10 minutos para procesar la información de los satélites y entregar la información requerida.

Una vez que estamos seguros que estamos conectados al módulo GPS, verificamos el estado de los satélites mediante el software GPST, el cual lo ejecutamos desde el terminal y podemos determinar, que existen satélites detectados, sin embargo todavía no son utilizados por el módulo para entregar los valores de posición y velocidad. Después de transcurrido cerca de 10 minutos las señales recibidas de los satélites son procesadas por el módulo GPS para entregar valores ciertos de posición y velocidad, objetivos de este proyecto, sin embargo también se recibe información complementaria que puede ser utilizada en otro tipo de proyecto. A continuación una gráfica de la obtención de la data real desde el módulo GPS, una vez que el vehículo se encuentra detenido.



**Figura No. 4.7 Adquisición de datos del módulo GPS mediante el Software GPSD – Auto Detenido.**

Como se puede visualizar en la figura anterior los datos entregados por los satélites son procesados correctamente y existe presentación de valores de posición, velocidad y otros datos. Se puede visualizar que el auto está detenido por cuanto la velocidad indicada es 0.1 Kph.

Finalmente, lo que queda pendiente es poner el vehículo en movimiento para determinar que efectivamente se ha cumplido con el objetivo del proyecto que consistía en realizar el control de posición y velocidad de móvil. Se presentarán varias fotografías donde se visualiza el velocímetro del vehículo y se realiza una comparación con el valor entregado por el software GPSD.



Velocímetro en 40 Km/h



Velocímetro en 70 Km/h

**Figura No. 4.8 Adquisición de datos del módulo GPS mediante el Software Desarrollado en Python – Auto en Movimiento.**

## Conclusiones.

1. El minicomputador Raspberry Pi, utilizado para realizar la adquisición de datos en tiempo real, por su gran versatilidad ha facilitado la instalación del módulo GPS CSC3, del software especializado GPS Daemon y del programa desarrollado en el lenguaje de programación Python, herramientas utilizadas para realizar el control de la velocidad y posición de un vehículo en movimiento, objetivo principal de esta tesina.
2. La exactitud de los datos de posición y velocidad que son calculados por el módulo GPS CSC3, depende del número de señales válidas que son recibidas desde los satélites, siendo necesario al menos tres satélites para obtener una posición confiable del blanco, por lo tanto mientras más satélites se encuentren disponibles, más exactos serán los valores entregados por mencionado módulo.
3. A pesar que el módulo GPS CSC3 se encuentre recibiendo información correcta desde los satélites, es necesario que el móvil se mantenga a velocidad constante, a fin de que este dispositivo pueda calcular de manera exacta la velocidad de desplazamiento, en base a las señales que son recibidas desde los satélites.
4. Dentro de la etapa de desarrollo del proyecto, fue muy importante la utilización del simulador de GPS, el cual permitió validar el funcionamiento del software de adquisición de datos y del procesamiento de la información, por cuanto se pudo controlar y verificar la data enviada,

asegurando de esta forma el éxito en la integración del minicomputador Raspberry Pi con el módulo GPS CSC3.

5. Debido a que el minicomputador Raspberry Pi fue desarrollado para ser utilizado con el Sistema Operativo Linux y al existir programadores que promueven mediante aportes gratuitos el uso de este sistema operativo, abre las puertas para que en un futuro cercano mediante el uso de este dispositivo se encuentren soluciones a bajo costo tanto para la industria como para el uso diario de las personas.

### Recomendaciones.

1. Se debe tener precaución al momento de interconectar dispositivos a través de los puertos UART y GPIO del Minicomputador Raspberry, por cuanto estos puertos están diseñado para recibir voltajes de 3.3 voltios, en caso de conectar voltajes de mayor valor pueden afectar o dañar al minicomputador.
2. Se recomienda que la instalación del módulo electrónico se lo realice en una ubicación, la cual permita que la antena del GPS tenga una vista al cielo, lo que facilitará la triangulación. Además, se recomienda que su instalación no se realice en un lugar cerrado por metales, debido a que un sitio así produciría una jaula de Faraday, lo que provocaría el no funcionamiento del módulo GPS.
3. Debido a que el Minicomputador Raspeberry Pi está concebido para ejecutar tareas puntuales, se recomienda no recargar a este dispositivo con aplicaciones que se ejecuten al mismo tiempo, ya que esto puede elevar la temperatura del procesador y debido a que el Raspberry no cuenta con un dispositivo disipador de calor, ésto puede dañar de manera permanente e irresversible al minicomputador.
4. Previo a la conexión de cualquier tipo de hardware se debe verificar el datasheet de cada uno de estos componentes, para evitar malas conexiones que pueden dañar a los componentes, así como también se debe asegurar que las tierras de cada uno de estos se encuentren interconectadas para evitar diferenciales de potencial que pueden de igual forma afectar a los mismos.

### REFERENCIAS.

- [1] Eben UPTON y Gareth HALFACREE, Raspberry Pi – User Guide, Editorial Wiley, Inglaterra, 2012.
- [2] Wikipedia.com, Raspberry Pi, [http://es.wikipedia.org/wiki/Raspberry\\_Pi](http://es.wikipedia.org/wiki/Raspberry_Pi), Fecha de Consulta: 16/abril/2013.
- [3] Fundación Raspberry Pi Raspberry Pi, Quick Start Guide, <http://www.raspberrypi.org/wp-content/uploads/2012/12/quick-start-guide-v1.1.pdf>, Fecha de consulta: 19/abril/2013.
- [4] National Marine Electronics Association (NMEA), Protocolo NMEA 0183, <http://www.nmea.org/>, 2001, Fecha de consulta: 22/abril/2013.
- [5] VINCOTECH, GPS Firmware GSC3-Bases Product- User Manual, VICONTECH, 2009.
- [6] Simon MONK, Programming the Raspberry Pi – Getting Started With Python, Editorial Mc Graw Hill, New York, 2013
- [7] Karl WRIGHT, Robert CRUSE y Paul KINGETT, The Raspberry Pi – Educational Manual, Inglaterra,2012.
- [8] FURUNO Electric Co. LTD., Operator’s Manual “GPS NAVIGATOR Model GP-37/G-32, FURUNO, Nishinomiya City – Japón, 2010.