

# INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ



ING. ELECTRÓNICA

INFORME TÉCNICO DE RESIDENCIA PROFESIONAL

NOMBRE DEL PROYECTO:

TECNOLOGÍA INALÁMBRICA PARA SU IMPLEMENTACIÓN EN ROBOTS PARA  
TRABAJO COLABORATIVO

---

Alumno

---

Nombre

Número de control

---

ESAU FLORES MARROQUIN

08270049

---

---

FECHA DE REPORTE ABRIL - 2014

---

# ÍNDICE

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Estado del arte.....</b>	<b>4</b>
<b>3. Justificación.....</b>	<b>6</b>
<b>4. Objetivos.....</b>	<b>7</b>
<b>5. Caracterización del área de trabajo.....</b>	<b>8</b>
<b>6. Problemas a resolver.....</b>	<b>9</b>
<b>7. Alcances y limitaciones.....</b>	<b>9</b>
<b>8. Marco Teórico.....</b>	<b>10</b>
<b>9. Desarrollo del proyecto.....</b>	<b>39</b>
<b>9.1.0. Fuente de voltaje regulada a +5 (Para pruebas).....</b>	<b>39</b>
<b>9.1.1. Diseño de la tarjeta principal.....</b>	<b>40</b>
<b>9.2. Comunicación Serial de comandos AT.....</b>	<b>45</b>
<b>9.2.1. Mensajes de Texto.....</b>	<b>47</b>
<b>9.2.2. Sistema de Posicionamiento Global.....</b>	<b>49</b>
<b>9.2.3. Visualización del posicionamiento Global.....</b>	<b>51</b>
<b>9.2.4. Interpretación de cadena de caracteres de GPS.....</b>	<b>55</b>
<b>9.2.5. Conversiones.....</b>	<b>56</b>
<b>9.2.6. Google Maps.....</b>	<b>57</b>
<b>9.2.7. Pruebas y análisis de errores en el GPS.....</b>	<b>58</b>
<b>9.3. Etapas de Programación.....</b>	<b>70</b>
<b>9.3.1 Implementación de módulo gm862-gps en el Robot.....</b>	<b>72</b>
<b>10. Conclusión y Recomendaciones.....</b>	<b>74</b>
<b>11. Bibliografía</b>	

# 1.-INTRODUCCIÓN

El desarrollo del proyecto de residencia profesional se basó básicamente en analizar, un módulo GM862- GPS, que será implementado en un robot de trabajo colaborativo. Este reporte enseñara a utilizar de manera adecuada nuestro modulo ya que este está basado específicamente en comandos AT y el lenguaje Phyton. Utilizando ciertas herramientas de la programación como el lenguaje arduino, para realizar un software que sea capaz de manejar el módulo de manera adecuada, realizando funciones como envió de mensajes, llamadas, pedir su posicionamiento global y enviar su posición a otro dispositivo. Una vez terminada estas actividades deberá implementarse en el robot de trabajo colaborativo. En donde el dispositivo GPS ayudara a visualizar la posición del robot de trabajo colaborativo por medio de la aplicación de google maps. . También se desarrolló el hardware necesario para la realización de pruebas físicas de manera adecuada.

## 2.-ESTADO DEL ARTE

- **Swarm-bots**

Swarm-bots es un proyecto patrocinado por el programa “The Future and Emerging Technologies” de la comisión europea (IST-2000-31010), con el objetivo de estudiar nuevos enfoques para el diseño e implementación de la auto-organización y auto-montaje de los robots. (Figura 1)



Figura 1.- **Swarm-bots**

El proyecto coordinado por el Dr. Marco Dorigo, ha sido galardonado en noviembre del 2003, por “The First EU Marie Curie Award”, y en noviembre del 2003, y en noviembre del 2005, por el prestigioso “FNRS Prize – Prix Dr. A De Leeuw-Damry-Bourlart”, con valor de 75,000 euros, por sus contribuciones a la inteligencia artificial y la robótica. (Figura 2)

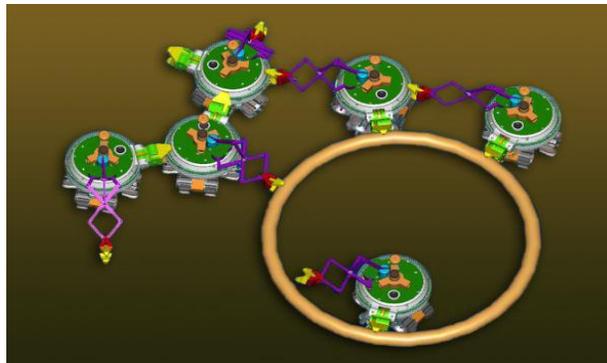


Figura 2- Estructura de **Swarm-bots**

- **Swarmanoid**

El proyecto Swarmanoid es un proyecto fundado por la comisión europea, donde robots trabajan en equipo de forma colaborativa. El objetivo principal de este proyecto de

investigación, es el diseño, ejecución y control de un novedoso sistema de distribución robótica.

El sistema se compone de heterogéneos, dinámicamente conectados con pequeños robots autónomos. ( Figura .3)

Marco Dorigo y Mauro Birattari, del laboratorio de investigación de inteligencia artificial de la universidad libre de Bruselas, desarrollan un equipo de 20 máquinas autónomas programadas para cooperar entre sí. En conjunto, estos robots forman lo que llamamos swarmanoid.

El swarmanoid tiene la intención de construirse en conjunto, constituyendo robots de 3 tipos: eye-bots, hand-bots y foot-bots (ojos, manos y pies).

- IRIDIA (Université Libre de Bruxelles )
- IDSIA (Università della Svizzera Italiana)
- LSRO (École Polytechnique Fédérale de Lausanne)
- LIS (École Polytechnique Fédérale de Lausanne)
- ISTC (Consiglio Nazionale delle Ricerche)



Figura.3.- Swarmanoid

La coordinación se basa en los sistemas de inteligencia de enjambre, disciplina que estudia las estructuras por las que muchos individuos se coordinan utilizando un control descentralizado, es decir, pueden auto organizarse. Este proyecto es el sucesor del proyecto Swarm-bots, y se basa en los resultados obtenidos durante el proyecto Swarm-bots.

### 3.-JUSTIFICACIÓN

Este proyecto se llevó a cabo con la finalidad de desarrollar los algoritmos más adecuados, basados en métodos de inteligencia artificial, que mejor resolvieran la problemática que se tiene en un sistema comprendido por robots con trabajo colaborativo que deban alcanzar un objetivo predeterminado, evadiendo obstáculos que se presentaran en el recorrido del robot. Para ello necesitamos el monitoreo constante de nuestro robot de trabajo colaborativo. En el momento que el robot encuentre un obstáculo o detecte la presencia de ciertos gases, este solicitará su posicionamiento global y lo enviara a otro dispositivo. Por el momento es un prototipo didáctico, pero tiene la finalidad de ser utilizado como un robot explorador de trabajo colaborativo, para detectar la presencia de gases nocivos para el ser humano. La utilidad más importante del módulo Gm862-Gps es monitorear al robot a grandes distancias desde una computadora con la aplicación de Google Maps, Además, se trabajó dentro de la plataforma Arduino, ya que brinda una completa flexibilidad en su uso, tanto a nivel físico, como a nivel software. Dicho entorno facilitará el uso tanto de sensores, comunicación inalámbrica, e incluso la implementación de la red dentro de la misma plataforma, pues cuenta con un lenguaje de programación Open Source bastante amigable, sencillo de comprender, y lo mejor de este sistema, cuenta con microcontroladores AVR, los cuales superan por mucho a los comúnmente conocidos PIC de Microchip, tanto en memoria, como en puertos analógicos, seriales.

## **4.-OBJETIVOS**

### ***OBJETIVO GENERAL***

Analizar la tecnología inalámbrica: el módulo gm862- GPS; con la finalidad de ser implementados en robots para trabajo colaborativo.

### ***OBJETIVO ESPECÍFICO.***

- 1.- Comunicación serial de comandos AT
- 2.- Reducir el margen de error del dispositivo GPS.
- 3.- Diseñar el hardware para las pruebas físicas.
- 4.- Realizar un programa que sea capaz de guardar la posición y enviarlo a otro dispositivo



## **6.- PROBLEMAS A RESOLVER**

- 1.- Desarrollar una fuente de voltaje para la alimentación del módulo.
- 2.- Desarrollo de una tarjeta de pruebas.
- 3.- Desarrollar funciones de comandos AT para verificar el módulo GM862-GPS.
- 4.- Desarrollar un software para el envío de mensaje.
- 5.- Desarrollar un software que proporcione el Posicionamiento Global, mostrarlo y enviarlo a otro dispositivo.
- 6.- Implementar el módulo Gm862-gps al Robot de trabajo colaborativo.

## **7.- ALCANCES Y LIMITACIONES**

El desarrollo del proyecto se enfocara en el análisis del módulo GM862-GPS, donde se realizara algunos software para la visualización del posicionamiento global y envío de este a otro dispositivo y se implementara a un robot de trabajo colaborativo para conocer su trayectoria u obstáculos que se le presenten con la finalidad de tener un robot explorador y poder alcanzar lugares donde sean de alto riesgo para la vida humana . Se limitará a la visualización de su posición por medio de la aplicación de google Maps , ya que el módulo Gm862-gps ira implementado sobre un robot de trabajo colaborativo para tener el monitoreo constante de su posicionamiento global.

## 8.-MARCO TEÓRICO

### SISTEMA DE POSICIONAMIENTO GLOBAL

El SPG o GPS (**Global Positioning Sistema: 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 situados 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, y de tal modo mide la distancia al satélite mediante "triangulación" (método de trilateración inversa), 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. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene las posiciones absolutas o coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites.

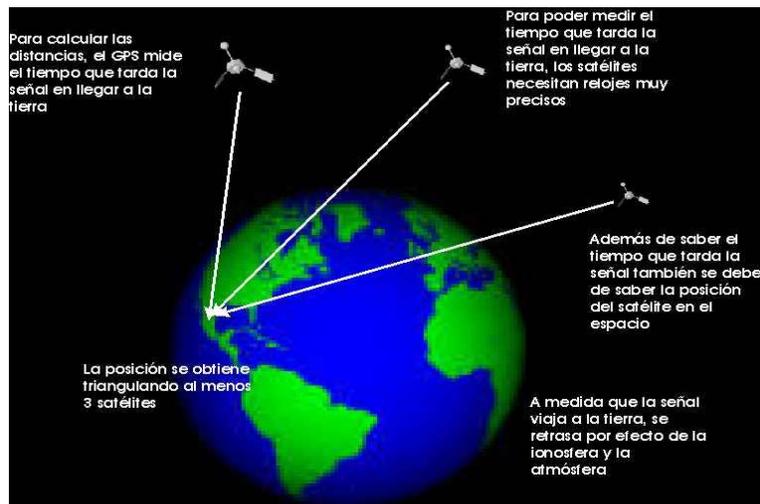


Figura 5.- La posición se obtiene triangulando al menos 3 satélites

## Características técnicas y prestaciones

**El Sistema Global de Navegación por Satélite lo componen:**

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

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

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

Segmento espacial

- Satélites en la constelación: 24 (4 x 6 órbitas )
- Altitud :20.200 km
- Período: 11 h 58min ( 12 horas sidéreas)
- Inclínación: 55 grados (respecto al ecuador terrestre)
- Vida útil : 7.5 años

Segmento de control

- Estación principal:1
- Antena de tierra:4

Estación monitora (de seguimiento): Colorado Springs, Hawaii, Kwajalein, Isla de Ascensión e Isla Diego García.

Señal RF (frecuencia portadora)

- Civil-1575,42 MHz (L.1) . Utiliza el código de adquisición aproximada (C/A)
- Militar-1227,60 MHz (L.2) utiliza el código de precisión (P), cifrado.
- Nivel de potencia de la señal : -160 dBW (en superficie de la tierra )
- Polarización: circular dextrógira.

Exactitud

- Posición : oficialmente indican aproximadamente 15m (en el 95% del tiempo), En realidad un GPS portátil monofrecuencia de 12 canales paralelos ofrece una precisión de 2.5 a 3 metros es más del 95% del tiempo. Con el WASS/EGNOS/MSAS activado, la precisión es de 1 a 2 metros.
- Hora 1ns
- Cobertura: mundial
- Capacidad de usuarios : ilimitada
- Sistema de coordenadas
- Sistema Geodésico Mundial 1984 (WGS84)
- Centrado en la tierra , fijo
- Disponibilidad: 24 satélites. no suficiente como medio primario de navegación.

## FUNCIONAMIENTO

La situación de los satélites puede ser determinada de antemano por el receptor con la información del llamado **almanaque** (un conjunto de valores con 5 elementos orbitales), parámetros que son transmitidos por los propios satélites. La colección de los almanaques de toda la constelación se completa cada 12-20 minutos y se guarda en el receptor GPS.

La información que es útil al receptor GPS para determinar su posición se llama efemérides. En este caso cada satélite emite sus propias efemérides, en la que se incluye la salud del satélite (si debe o no ser considerado para la toma de la posición), su posición en el espacio, su hora atómica, información doppler, etc.

El receptor GPS utiliza la información enviada por los satélites (hora en la que emitieron las señales, localización de los mismos) y trata de sincronizar su reloj interno con el reloj atómico que poseen los satélites. La sincronización es un proceso de prueba y error que en un receptor portátil ocurre una vez cada segundo. Una vez sincronizado el reloj, puede determinar su distancia hasta los satélites, y usa esa información para calcular su posición en la tierra.

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

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

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

## **Fiabilidad de los datos**

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

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

## **Fuentes de error**

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

Al introducir el atraso, el receptor compara una serie de bits (unidad binaria) recibida del satélite con una versión interna. Cuando se comparan los límites de la serie, las electrónicas pueden meter la diferencia a 1% de un tiempo BIT, o aproximadamente 10 nanosegundos por el código de Adquisición Aproximada. Desde entonces las señales GPS se propagan a la velocidad de luz, que representa un error de 3 metros. Este es el error mínimo posible usando solamente la señal GPS Código de Adquisición Aproximada. La precisión de la posición se mejora con una señal P (Y). Al presumir la misma precisión de 1% de tiempo BIT, la señal P(Y) (alta frecuencia) resulta en una precisión de más o menos 30 centímetros. Los errores en las electrónicas son una de las varias razones que perjudican la precisión (ver la tabla).

Tabla 8.1 Fuentes de error

FUENTE	EFEECTO
<b>Ionosfera</b>	$\pm 3$ m
<b>Efemérides</b>	$\pm 2.5$ m
<b>Reloj satelital</b>	$\pm 2$ m
<b>Distorsión multibandas</b>	$\pm 1$ m
<b>Troposfera</b>	$\pm 0.5$ m
<b>Errores numéricos</b>	$\pm 1$ m o menos



Figura 6- MODULO TELIT GM862-GPS.

El GM862-GPS combina el poderoso motor GSM del GM862 con un SiRF III de 20 canales de alta sensibilidad receptor GPS. Acceda al módulo, emitir el comando de consulta de GPS, y usted tendrá datos NMEA! Si esta unidad está dentro del rango de una torre celular, usted sabrá dónde se encuentra dentro de 9 metros de cualquier punto de la superficie de la tierra.

El GM862-GPS incluye un intérprete de comandos de Python, y la capacidad de cuádruple banda celular (frecuencias 850/900/1800/1900MHz). Los programas de Python pueden ser cargados en el GM862 el control de los GPIO y la emisión de órdenes globales celulares. Módulo de pin compatible con la memoria interna para actualizar los programas de usuario. El módulo GM862 se controla mediante comandos AT. Una vez que el módulo se ha

conectado a otro módulo o módem, una conexión en serie se realiza y los datos pueden ser transferidos tan simples como pasar y recibir cadenas de serie a través del módulo.



Figura 7.- Modulo GPS

#### Características:

- SiRF III de alta sensibilidad receptor GPS
- GSM de cuádruple banda
- El titular de la Junta de SIM
- Embedded pila TCP / IP
- GPRS Clase 10
- PYTHON intérprete de secuencias de comandos
- FTP integrado y el cliente SMTP
- 17mA promedio de stand-by, 3,5 mA en modo de bajo consumo
- Media de funcionamiento de 250 mA de corriente
- Datos, Voz, SMS y fax
- Los datos se transmisión a 57.6Kbps
- Tensión de alimentación: 3.4-4.2V
- cámara capaz
- 2 x conectores de antena MMCX
-

El módulo Gm862-gps permite integrar a los dispositivos capacidad de comunicaciones de voz y datos mediante el uso de la red de telefonía celular GSM. Mediante este módulo se pueden lograr varias formas de comunicación, como lo son los mensajes de texto (SMS), llamadas de voz y de datos (CSD), así como el acceso a redes de datos de conmutación de paquetes, como lo es internet a través del servicio GPRS (PSD). El módulo está pensado para poder ser usado en cualquier parte del mundo, ya que soporta las cuatro bandas de frecuencia de GSM (850, 900, 1800 y 1900 Mhz).

Este módulo Gm862-gps ofrece algunas ventajas que lo convierten en un producto bastante versátil:

## PLATAFORMA ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (*boot loader*) que corre en la placa.

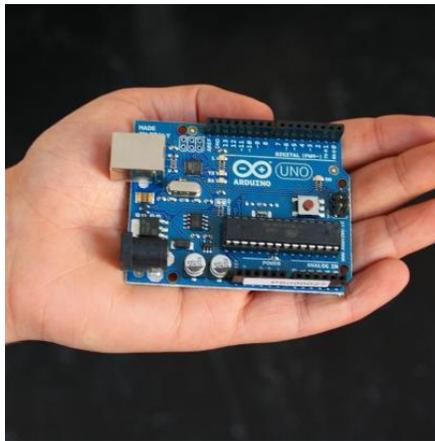


Figura 8-Arduino Uno

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador (por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente. Al ser open-hardware, tanto su diseño como su distribución es libre. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. El proyecto Arduino recibió una mención honorífica en la categoría de Comunidades Digital en el Prix Ars Electrónica de 2006.

El Arduino Uno es una placa electrónica basada en el microprocesador Atmega328 (datasheet). Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, un cristal de cuarzo 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio.

Contiene todo lo necesario para apoyar el microcontrolador , simplemente conectarlo a un ordenador con un cable USB , o alimentarla con un adaptador de corriente AC a DC para empezar.

El Uno se diferencia de todas las placas anteriores en que no utiliza el chip controlador de USB a serial FTDI . En lugar de ello , se cuenta con el Atmega16U2 ( Atmega8U2 hasta la versión R2) programado como convertidor USB a serie . Revisión 2 de la junta Uno tiene una resistencia tirando de la línea 8U2 HWB a tierra, por lo que es más fácil de poner en modo DFU. Revisión 3 de la placa tiene las siguientes características nuevas :

- 1.0 pinout : pines SDA y SCL cerca al pin AREF y otros dos nuevos pernos colocados cerca del pin RESET, el IOREF que permiten a los escudos para adaptarse al voltaje suministrado desde la pizarra añadido. En el futuro , los escudos serán compatibles tanto con la junta que utiliza el AVR , que funciona con 5V y con el Arduino Debido que opera con 3.3V . El segundo es un pin no está conectado, que se reserva para usos futuros .
- Más fuerte circuito de RESET.
- ATmega 16U2 reemplazar el 8U2 .

"Uno" significa uno en italiano y es nombrado para conmemorar el próximo lanzamiento de Arduino 1.0 . El Uno y la versión 1.0 será la versión de referencia de Arduino , moviéndose hacia adelante . El Uno es el último de una serie de placas Arduino USB y el modelo de referencia para la plataforma Arduino , para una comparación con las versiones anteriores , consulte el índice de la placa Arduino .

## Características Principales de Arduino uno

Microcontrolador	ATmega328
Voltaje de Operación	5v
Voltaje de entrada (recomendado)	7-12v
Voltaje de entrada (limite)	6-20v
Pines digital i/o	14 ( de los cuales 6 proporciona pwm)
Pines de entradas analógicas	6
Corriente continua para e/s de pin	40mA
Corriente de la cc para pin 3.3 V	50Ma
Memoria Flash	32 KB (ATmega328) de los cuales 0.5 KB son utilizados para el gestor de arranque.
SRAM	2 KB (ATmega328)
EPROM	1 KB (ATmega328)
Velocidad de Reloj	16 MHz



Fig. 9.- Muestra las terminales del arduino uno

## POTENCIA

El Arduino Uno puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

Externa (no USB) de potencia puede venir con un adaptador de AC -DC (pared) o la batería. El adaptador se puede conectar al conectar un centro positivo enchufe de 2,1 mm en el conector de alimentación de la placa. Conduce de una batería se pueden insertar en los encabezados pin GND y Vin del conector de alimentación.

La tarjeta puede funcionar con un suministro externo de 6 a 20 voltios. Si se proporcionan menos de 7V, no obstante, el pin de 5V puede suministrar menos de cinco voltios y la junta puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son como sigue:

-VIN . El voltaje de entrada de la placa Arduino cuando se trata de utilizar una fuente de alimentación externa (por oposición a 5 voltios de la conexión USB o de otra fuente de alimentación regulada ) . Usted puede suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de poder , acceder a ella a través de este pin.

Pin- 5V.This emite una 5V regulado por el regulador en el tablero. El tablero puede ser alimentado ya sea desde el conector de alimentación de CC ( 7 - 12 ) , el conector USB (5V ) o el pasador de VIN del tablero ( 7 -12V ) . El suministro de tensión a través de los 5V o 3.3V pins no pasa por el regulador , y puede dañar su tablero. No aconsejamos ella.

- 3V3 . Un suministro de 3,3 voltios generados por el regulador a bordo. El drenaje actual

máximo es de 50 mA.

-GND . Pins de tierra.

- IOREF . Este pin de la placa Arduino proporciona la referencia de tensión con la que opera el microcontrolador. Un escudo configurado puede leer el voltaje pin IOREF y seleccione la fuente de alimentación adecuada o habilitar traductores tensión en las salidas para trabajar con los 5V o 3.3V.

## Memoria

El ATmega328 tiene 32 KB (con 0,5 KB utilizado por el gestor de arranque). También dispone de 2 KB de SRAM y 1 KB de EEPROM (que puede ser leído y escrito con la librería EEPROM).

Entrada y salida

Cada uno de los 14 pines digitales en el Uno se puede utilizar como una entrada o salida, usando (funciones) `pinMode ( )` , `digitalWrite ( )` , y `digitalRead` . Operan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20-50 kOhms . Además , algunos pines han especializado sus funciones. :

- Serial: 0 ( RX) y 1 ( TX ) . Se utiliza para recibir (RX) y transmisión (TX) TTL datos en serie. Estos pines están conectados a los pines correspondientes de la USB -to- TTL de chip de serie ATmega328.

- Interrupciones Externas: 2 y 3 . Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.

Véase la función `attachInterrupt()` .  
- PWM : 3 , 5 , 6 , 9 , 10 , y 11 . Proporcionar salida PWM de 8 bits con la función `analogWrite()` .

- SPI : 10 ( SS ) , 11 ( MOSI ) , 12 ( MISO ) , 13 ( SCK ) . Estos pines soportan comunicación SPI usando la librería SPI .

- LED: 13. Hay un built-in LED conectado al pin digital 13. Cuando el pin tiene un estado alto, el LED está encendido, cuando el pin tiene un estado bajo, está apagado. El Uno tiene 6 entradas analógicas, etiquetados A0 a A5, cada uno de los cuales proporcionan 10 bits de resolución (es decir, 1.024 valores diferentes) . Por defecto se miden desde el 0 a 5 voltios, aunque es posible cambiar el extremo superior de su rango utilizando el pin AREF y la función `analogReference()` . Además, algunos pines han sido especializados para proporcionar funcionalidad:

- TWI : pin A4 o A5 SDA y SCL o pin. Soporte comunicación TWI utilizando la librería `Wire` .

Hay un par de otras patas de la placa :

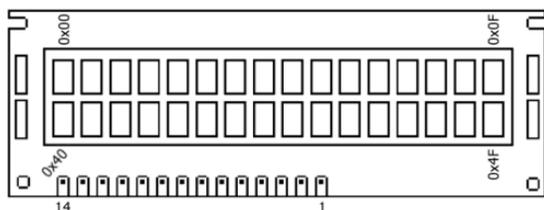
- AREF . Voltaje de referencia para las entradas analógicas. Se utiliza con `analogReference()` .

- Reset. Lleve esta línea BAJO para restablecer el microcontrolador. Normalmente se utiliza para agregar un botón de reinicio para escudos que bloquean el uno en el tablero. Ver también el mapeo entre los pines de Arduino y puertos ATmega328 . La asignación para el Atmega8 , 168, y 328 es idéntico.

## PANTALLA DE CRISTAL LÍQUIDO

Una pantalla de cristal líquido o *LCD* (sigla del inglés *liquid crystal display*) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

LCD Display with 2 lines x 16 characters :



Pin No	Name	Function	Description
1	Vss	Power	GND
2	Vdd	Power	+ 5 V
3	Vee	Contrast Adj.	(-2) 0 - 5 V
4	RS	Command	Register Select
5	R/W	Command	Read / Write
6	E	Command	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

Figura 10.- Pantalla LCD

## CARACTERÍSTICAS

Cuando se aplica un voltaje a través de los electrodos, una fuerza de giro orienta las moléculas de cristal líquido paralelas al campo eléctrico, que distorsiona la estructura helicoidal (esto se puede resistir gracias a las fuerzas elásticas desde que las moléculas están limitadas a las superficies). Esto reduce la rotación de la polarización de la luz incidente, y el dispositivo aparece gris. Si la tensión aplicada es lo suficientemente grande, las moléculas de cristal líquido en el centro de la capa son casi completamente desenrolladas y la polarización de la luz incidente no es rotada ya que pasa a través de la capa de cristal líquido. Esta luz será principalmente polarizada perpendicular al segundo filtro, y por eso será bloqueada y el pixel aparecerá negro. Por el control de la tensión aplicada a través de la capa de cristal líquido en cada píxel, la luz se puede permitir pasar a

través de distintas cantidades, constituyéndose los diferentes tonos de gris. El efecto óptico de un dispositivo twisted nematic (TN) en el estado del voltaje es mucho menos dependiente de las variaciones de espesor del dispositivo que en el estado del voltaje de compensación. Debido a esto, estos dispositivos suelen usarse entre polarizadores cruzados de tal manera que parecen brillantes sin tensión (el ojo es mucho más sensible a las variaciones en el estado oscuro que en el brillante). Estos dispositivos también pueden funcionar en paralelo entre polarizadores, en cuyo caso la luz y la oscuridad son estados invertidos. La tensión de compensación en el estado oscuro de esta configuración aparece enrojecida debido a las pequeñas variaciones de espesor en todo el dispositivo. Tanto el material del cristal líquido como el de la capa de alineación contienen compuestos iónicos. Si un campo eléctrico de una determinada polaridad se aplica durante un período prolongado, este material iónico es atraído hacia la superficie y se degrada el rendimiento del dispositivo. Esto se intenta evitar, ya sea mediante la aplicación de una corriente alterna o por inversión de la polaridad del campo eléctrico que está dirigida al dispositivo (la respuesta de la capa de cristal líquido es idéntica, independientemente de la polaridad de los campos aplicados).

## COMANDOS AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un Terminal MODEM. Los comandos AT fueron desarrollados en 1977 por Dennis Hayes como un interfaz de comunicación con un MODEM para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, con el avance del baudio, fueron las compañías Microcomm y US Robotics las que siguieron desarrollando y expandiendo expandiendo el juego de comandos hasta universalizarlo.

Los comandos AT se denominan así por la abreviatura de attention.

Aunque la finalidad principal de los comandos AT es la comunicación con modems, la telefonía móvil GSM también ha adoptado como estandar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.

Notación empleada en las definiciones:

Comando AT: [Definición técnica]

- Funcionalidad del comando
- Sintaxis: Petición | Respuesta
- Respuesta obtenida al comando

## Comandos AT para operaciones básicas

AT+CPAS: [Phone Activity Status]

### 1) AT+CPAS=?

- *Muestra la implementación del comando.*

- *Sintaxis: AT+CPAS=? / +CPAS: (lista de estados soportados)*

**0 - Ready (Encendido pero inactivo)**

**1 - Unavailable (No disponible)**

**2 - Unknown (Desconocido)**

**3 - Ringing (Llamada entrante en proceso)**

**4 - Call in progress (Llamada saliente en proceso)**

**5 - Asleep (Dormido)**

- **Respuesta:** +CMGD: (0,2,3,4)

### 2) AT+CPAS

- **Informa del estado de actividad del teléfono.**

- **Sintaxis:** AT+CPAS / +CPAS: <estado>

- **Respuesta:** +CPAS: 0, en estado normal de inactividad.

+CPAS: 3, si el teléfono atacado está sonando a causa de una llamada entrante.

### 3) ATD: [Dial Command]

- *Inicia una llamada telefónica*

- **Sintaxis:** ATD64612345 para una llamada de Datos.(se omite la coma)

**ATD64612345;** para una llamada de Voz. (Importante ;)

ATD>"Alberto"; para llamar al contacto almacenado en la agenda con el texto asociado Alberto.

#### 4) AT+CCFC: [Call Forwarding Number]

- **Gestiona el Desvío de Llamadas. Permite redireccionar llamadas entrantes a otro número de teléfono.**

- Syntax:  
AT+CCFC=<razón>,<modo>,<número>,<tipo>,<clase>,[<subaddr>,<satype>,[<time>]]

**<Razón> Razón por la cual entra en acción el desvío de llamada.**

- 0 - *incondicional*
- 1 - *si teléfono ocupado*
- 2 - *si no obtiene respuesta*
- 3 - *si inalcanzable*
- 4 - *todos los desvíos de llamadas*
- 5 - *todos los desvíos de llamadas condicionales*

**<Modo> Estado del desvío de llamada.**

- 0 - *deshabilitado*
- 1 - *habilitado*
- 2 - *query status*
- 3 - *registro*
- 4 - *erasure (borrado)*

**<Número>** Cadena de texto con el número de teléfono destino del desvío de llamada. Se especifica en el formato indicado en el campo <type>

**<tipo>** Tipo de código de dirección de teléfono:

- 145 - *para código internacional +*
- 129 - *en otro caso*

<clase> Código que representa la clase de información que contiene la llamada a desviar.

1 - voz

2 - datos

4 - fax

7 - cualquier clase (por defecto)

<time> Tiempo en segundos a esperar antes de desviar la llamada.

1..30 (por defecto, 20)

<Status> Estado de la opción desvío de llamadas. (Sólo en respuesta AT)

0 - no activo

1 - activo

5).-AT+CGMI: [Request Manufacturer Identification]

- **Petición de identificación del Fabricante** (Marca del teléfono).

- **Sintaxis:** AT+CGMM | <fabricante>

- **Respuesta:** Nokia Mobile Phones

6).-AT+CGMM: [Request Model Identification]

- **Petición de identificación del modelo de teléfono.**

- **Sintaxis:** AT+CGMM | <modelo>

- **Respuesta:** Nokia 6820

7).-AT+CGSN: [*Request Product Serial Number Identification*]

- **Petición de identificación del número de serie del producto.**

- **Sintaxis:** AT+CGSN | <IMEI>

- **Respuesta:** 1234567890987654321 (IMEI)

8).-AT+CBC: [*Battery Charge*]

- **Devuelve el estado de carga de la batería.**

- **Sintaxis:** AT+CBC | +CBC: <bcs>, <bcl>

<bcs> = 0 indica que el teléfono está conectado a una batería.

<bcl> = 0 indica que el teléfono tiene la batería agotada.

= 1..100 indica el porcentaje de carga que aún queda por agotar.

- **Respuesta:** +CBC:0,56

9).-AT+CSQ: [*Signal Quality*]

- **Devuelve el estado de calidad de la señal de cobertura.**

- **Sintaxis:** AT+CSQ | +CSQ: <rssi>,<ber>

<rssi> = 0 indica -113 dBm o menos

= 1 indica -111 dBm

= 2..30 indica -109..-53 dBm

= 31 indica -51dBm o más

= 99 indica desconocido

<ber> = 99 indica porcentaje desconocido

- **Respuesta:** +CSQ: 13,9

## Comandos AT para gestión de la agenda de contactos

### 1) AT+CPBS: [Select Phone Book Memory Storage]

AT+CPBS?

- **Informa de los dispositivos de memoria que soporta el teléfono para almacenar las distintas listas de contactos.**

- *Sintaxis: AT+CPBS? | +CPBS: "XX", donde "XX" se sustituye por el dispositivo de almacenamiento:*

"SM" - *SIM phonebook list* [**Lista de contactos de la agenda SIM**]

"TA" - *TA phonebook list* [**Lista de contactos del terminal**]

"LD" - *SIM last dialing list* [**Lista de números marcados**]

"DC" - *Dialled call list* [**Lista de llamadas realizadas**]

"RC" - *ME received calls list* [**Lista de llamadas recibidas**]

"MC" - *ME missed call list* [**Lista de llamadas perdidas**]

"EN" - *Emergency number list* [**Lista de números de emergencia**]

"FD" - *SIM fix dialing list*

"MT" - *ME + SIM combined list*

"ON" - *SIM o ME own number list*

- **Respuesta:** +CPBS: "SM"

## 2) AT+CPBS="XX"

- Selecciona por defecto uno de los dispositivos de memoria que soporta el teléfono para almacenar las listas de contactos.

- *Sintaxis: AT+CPBS="XX", donde "XX" se sustituye por el dispositivo de almacenamiento:*

"SM" - SIM phonebook list [Lista de contactos de la agenda SIM]

"TA" - TA phonebook list [Lista de contactos del terminal]

"LD" - SIM last dialing list [Lista de números marcados]

"DC" - Dialed call list [Lista de llamadas realizadas]

"RC" - ME received calls list [Lista de llamadas recibidas]

"MC" - ME missed call list [Lista de llamadas perdidas]

"MT" - ME + SIM combined list

"ON" - SIM o ME own number list

## 3) AT+CPBR: [Read Phone Book Entry]

**AT+CPBR=?**

- *Informa del tamaño de la agenda de contactos.*

- *Sintaxis: AT+CPBR=? | +CPBR: <(1-n)>,<nlen>,<tlen>*

*<(1-n)> indica el rango de índices que la agenda puede contener.*

*<nlen> indica la longitud máxima permitida para un número de teléfono.*

*<tlen> indica la longitud máxima permitida para el texto asociado a ese número (nombre del contacto).*

- **Respuesta:** +CPBR: (1-150),48,14

#### 4) AT+CPBR=<índice>

- Leer una entrada de la agenda de contactos.

- **Sintaxis:** AT+CPBR=<índice inicial> [, <índice final>] | +CPBR: <índice>, <número>, <tipo>, <texto>

<índice> indica el índice de la agenda de contactos.

<número> indica el número de teléfono almacenado en el índice.

<tipo> indica el tipo de número de teléfono 129 o 145 si incluye el prefijo internacional +.

<texto> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.

- Respuesta a AT+CPBR=8: +CPBR: 8,"646123456",129,"Pepito"

#### 5).-AT+CPBS;+CPBR [Leer una entrada de una lista de contactos seleccionada]

- Primero elegimos la lista de contactos a la que queremos acceder, y luego leemos una entrada por su índice.

**Sintaxis:** AT+CPBS="XX";+CPBR=<índice>, donde "XX" se sustituye por el dispositivo de almacenamiento:

"SM" - SIM phonebook list [Lista de contactos de la agenda SIM]

"TA" - TA phonebook list [Lista de contactos del terminal]

"LD" - SIM last dialing list [Lista de números marcados]

"DC" - Dialed call list [Lista de llamadas realizadas]

"RC" - ME received calls list [Lista de llamadas recibidas]

"MC" - ME missed call list [Lista de llamadas perdidas]

"EN" - Emergency number list [Lista de números de emergencia]

"FD" - SIM fix dialing list

"MT" - ME + SIM combined list

"ON" - SIM o ME own number list

**6).-AT+CPBF:** [Find Phone Book Entries]

- **Devuelve la entrada de la agenda de contactos cuyo texto asociado a un número contiene la cadena alfanumérica proporcionada.**

- **Sintaxis:** AT+CPBF="texto a encontrar" | +CPBR: <índice>, <número>, <tipo>, <texto>  
"texto a encontrar" es case-sensitive

<índice> indica el índice de la agenda de contactos.

<número> indica el número de teléfono almacenado en el índice.

<tipo> indica el tipo de tipo de número de teléfono. Por defecto, 129 o 145 si incluye el prefijo internacional +.

<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.

- **Respuesta** a AT+CPBF="Alberto": +CPBF: 19, "646987654", 129,"Alberto"

**7).-AT+CPBW:** [Write Phone Book Entry]

- **Escribe una entrada en la agenda de contactos.**

- **Sintaxis:** AT+CPBW = <índice>, <número>, <tipo>, <texto>

<índice> indica el índice de la agenda de contactos donde se creará la entrada de contacto. Si no se proporciona índice, se añade la entrada en el primer hueco libre.

<número> indica el número de teléfono almacenado en el índice.

<tipo> indica el tipo de tipo de número de teléfono. Por defecto, 129 o 145 si incluye el prefijo internacional +.

<text> indica el texto asociado al número de teléfono, normalmente, el nombre del contacto.

**Nota:** Si únicamente se proporciona el campo del índice (omitiendo el resto de campos), la entrada de la agenda asociada a ese índice se borrará.

## Comandos AT para gestión de mensajes SMS

**AT+CMGF:** [Message Format]

1) *AT+CMGF=?*

- **Informa de los formatos de mensaje soportados por el teléfono**

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

**modo = 0** *indica formato de mensajes en modo PDU*

**modo = 1** *indica formato de mensajes en modo TEXTO*

2) **AT+CMGF?**

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

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

**modo = 0** *indica formato de mensajes en modo PDU*

**modo = 1** *indica formato de mensajes en modo TEXTO*

3) **AT+CMGF=<modo>**

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

- *Sintaxis: AT+CMGF=<modo>*

**modo = 0** *indica formato de mensajes en modo PDU*

**modo = 1** *indica formato de mensajes en modo TEXTO*

4) **AT+CMGL:** [*List Messages*]

*AT+CMGL=?*

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

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

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

<estados>:

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

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

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

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

4 | "ALL": Todos los mensajes almacenados.

5) AT+CMGL=<estado>

- **Lista todos los mensajes almacenados correspondientes al estado especificado.**

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

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

<estado> Estado del mensaje.

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

<timestamp> Fecha y hora.

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

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

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

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

6).-AT+CMGR: [Read Message]

- **Permite leer mensajes SMS de la bandeja de entrada.**

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

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

<estado> Estado del mensaje.

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

<timestamp> Fecha y hora.

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

- **Respuesta a AT+CMGR=1:**

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

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

7) **AT+CMGD:** [Delete Message]

AT+CMGD=?

- **Muestra la implementación del comando.**

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

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

8) **AT+CMGD=<índice>**

- **Elimina el mensaje de índice especificado.**

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

**9) AT+CMGW:** *[Write Message]*

- **Permite escribir un mensaje SMS en memoria** (Bandeja de salida).

- **Sintaxis:** *AT+CMGW=<número> [Presionar CR]*

> **Escribimos el cuerpo del mensaje.** *[Presionar Ctrl-Z]*

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

- *Ejemplo:*

*AT+CMGW="679123456" [Presionar CR]*

> **Hola! Como estás? Hace mucho q no te veo. Ciao.** *[Ctrl-Z]*

**Respuesta:** *+CMGW: <índice que ocupará en la memoria>*

**10).-AT+CMGS:** *[Send Message]*

- **Permite enviar un mensaje SMS.**

- **Sintaxis:** *AT+CMGS=<número> [Presionar CR]*

> **Escribimos el cuerpo del mensaje.** *[Presionar Ctrl-Z]*

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

- *Ejemplo:*

*AT+CMGS="679123456" [Presionar CR]*

> **Hola! Como estás? Hace mucho q no te veo. Ciao.** *[Ctrl-Z]*

*Respuesta: +CMGS: 213*

## 9.-Desarrollo del proyecto

A continuación se hará mención del diseño de hardware necesario para llevar a cabo las pruebas del programa a desarrollar.

### 9.1.1 Fuente de voltaje regulada A +5V. (Para pruebas)

Se desarrolló una fuente de voltaje, con el objetivo de alimentar al módulo GM862-GPS, ya que utiliza un voltaje de +5 Volts.

La fuente consta de un transformador de relación 10:1 a 1 Amper, 2 diodos rectificadores a 1Amper, 1 capacitor de 1000  $\mu$ F, 1 capacitor de 100  $\mu$ F, y 1 reguladores de voltaje: 7805. Contiene una resistencia delimitadora que va conectada aun diodo led, la cual indica que nuestra fuente está encendida.

El funcionamiento es simple, primero se reduce el voltaje de 120 V ac a 12 V ac por medio del transformador. Posteriormente se rectifica la señal con los diodos y se filtra con el capacitor. Después pasa por el regulador de voltaje para mantener los 5V aunque su entrada este variando. Por si quedo algún ruido filtramos la señal con otro capacitor y obtenemos los +5V deseados.

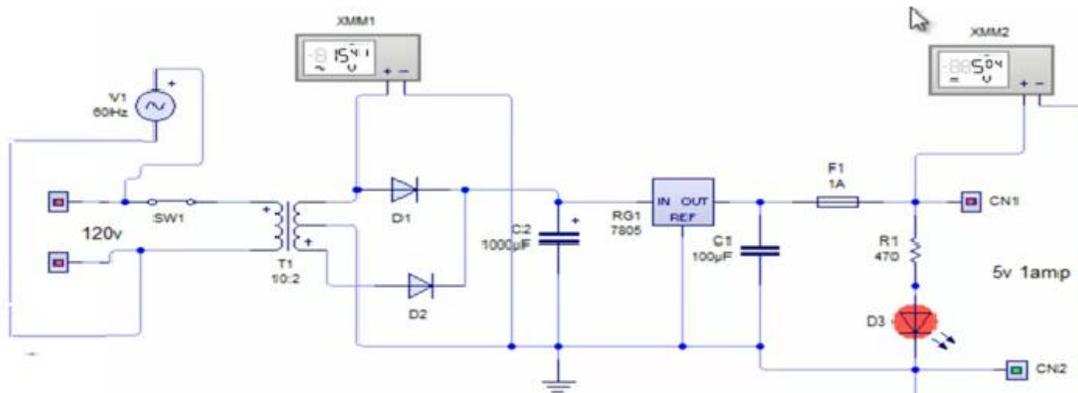


Figura 11.- Diagrama de Fuente Regulada

## 9.1.2 Diseño De La Tarjeta Principal

A continuación se presenta el diseño de una tarjeta que facilita la conexión entre el arduino y el módulo GM862-GPS.

### **Pines del Módulo GM862-GPS.**

La tarjeta principal se alimenta con una fuente de voltaje regulado, la cual va conectada a la bornera de alimentación.

El módulo Gm862-gps va insertado sobre los pines hembra, pero se debe verificar que Rx y Tx concuerden con el Tx y Rx de la tarjeta de pruebas.

El Rx del módulo Gm862-gps va conectado al Rx del arduino uno (pin 0) en su trayectoria pasa por un seguidor de voltaje, para que el voltaje sea el adecuado y no tenga pérdidas.

El Tx del arduino Uno (pin 1) va conectado al Tx del módulo, debido a que el módulo Gm862-gps necesita un voltaje de 3.7 V para la recepción, aplicamos un divisor de tensión con resistencias de 100k para obtener el voltaje deseado. Después de tener el voltaje adecuado se pasa por un seguidor de voltaje para que no existan pérdidas de voltaje y la comunicación sea la adecuada.

El Push button para obtener datos va conectado al pin 3 del arduino uno.

El Push button para el envío de mensajes va conectado al pin 2 del arduino uno.

La terminal 4 (RS) del módulo LCD va conectado al pin 12 del arduino uno.

La terminal 6 (E) del módulo LCD va conectado al pin 11 del arduino uno.

Los bus de datos del módulo LCD que abarcan de la terminal 11, 12,13, 14 van conectados a los pines 7, 6, 5,4 del arduino uno.

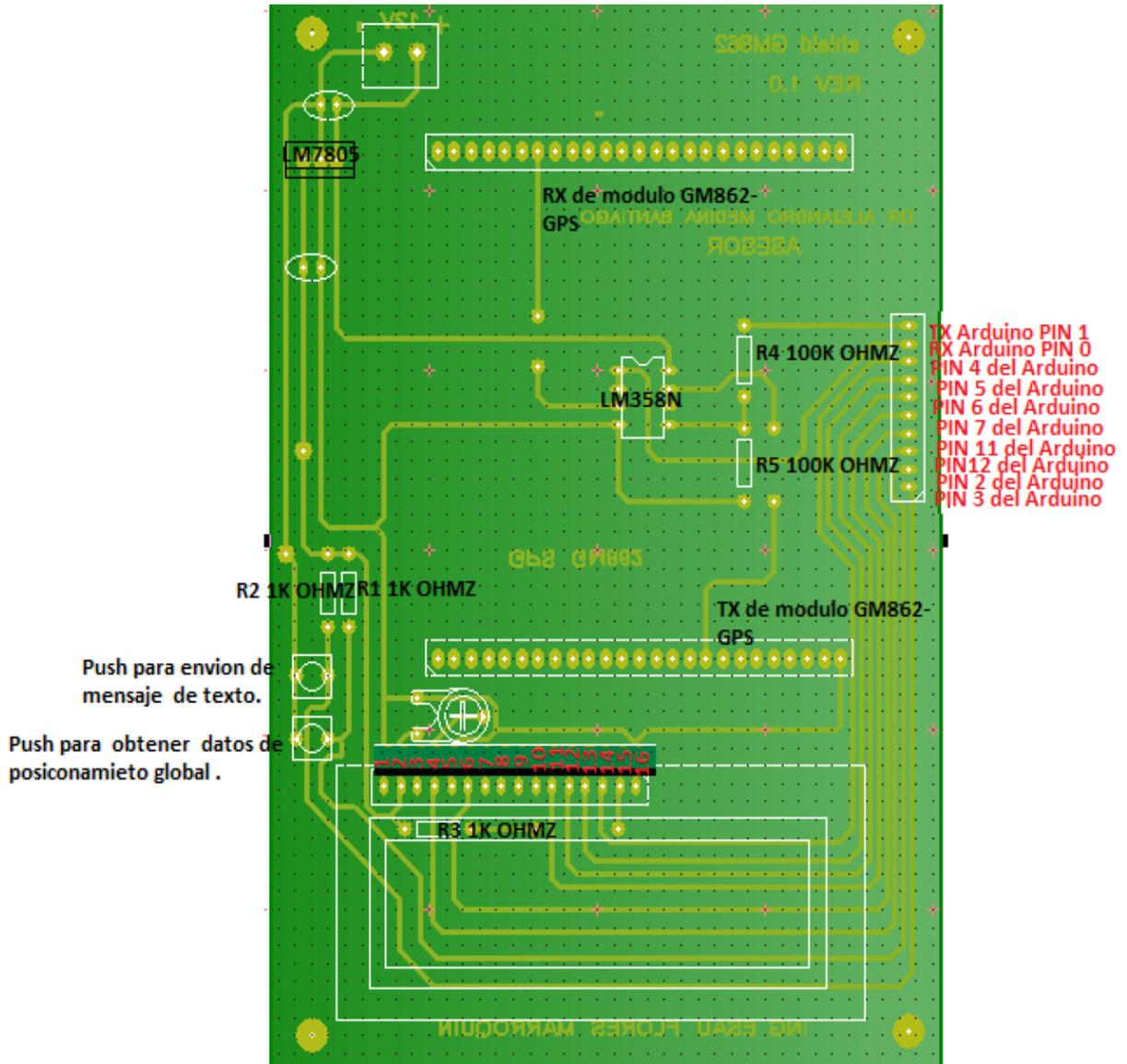


Figura.12-Diagrama de Conexión

**Tabla 9.1 Asignación de pines de LCD**

TERMINAL	ASIGNACION
1- LCD	GND (Tierra)
2- LCD	VDD (Alimentación a 5V )
3- LCD	Vo (Ajuste de contraste)
4- LCD	Rs(Selección de Registro)
5- LCD	R/W (Lectura y escritura )
6- LCD	E (habilitación )
8 a 16- LCD	Bus de datos
Push button( envió mensaje)	Entrada
Push button(obtener datos)	Interrupción
1 de arduino	TX
0 de arduino	RX
35 del módulo Gm862-gps	TX
41 del módulo Gm862-gps	RX
1,3,5,7 del módulo Gm862-gps	Voltaje de alimentación(3.8V)
G1,G2,G3,G4,G5,G6,G7, 2 ,4 del módulo Gm862-gps	GND (Tierra)

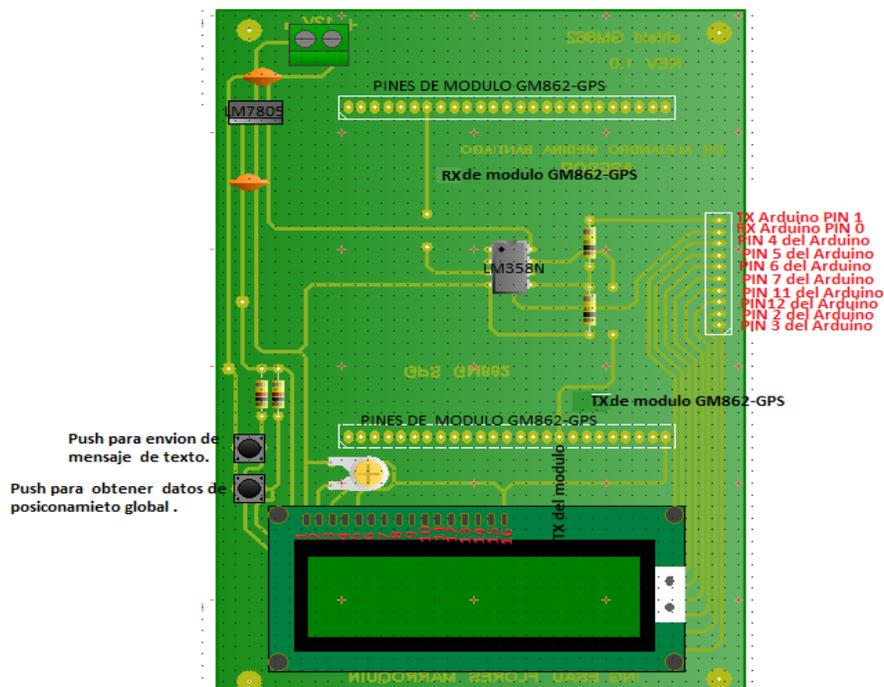


Figura.13-Diagrama de Asignación de terminales.



Una vez terminada la tarjeta se inserta en el módulo GM862-GPS, ya que esta ayudara a realizar las pruebas necesarias, se facilitara las conexiones entre el arduino, la fuente voltaje y el módulo gm862-gps.

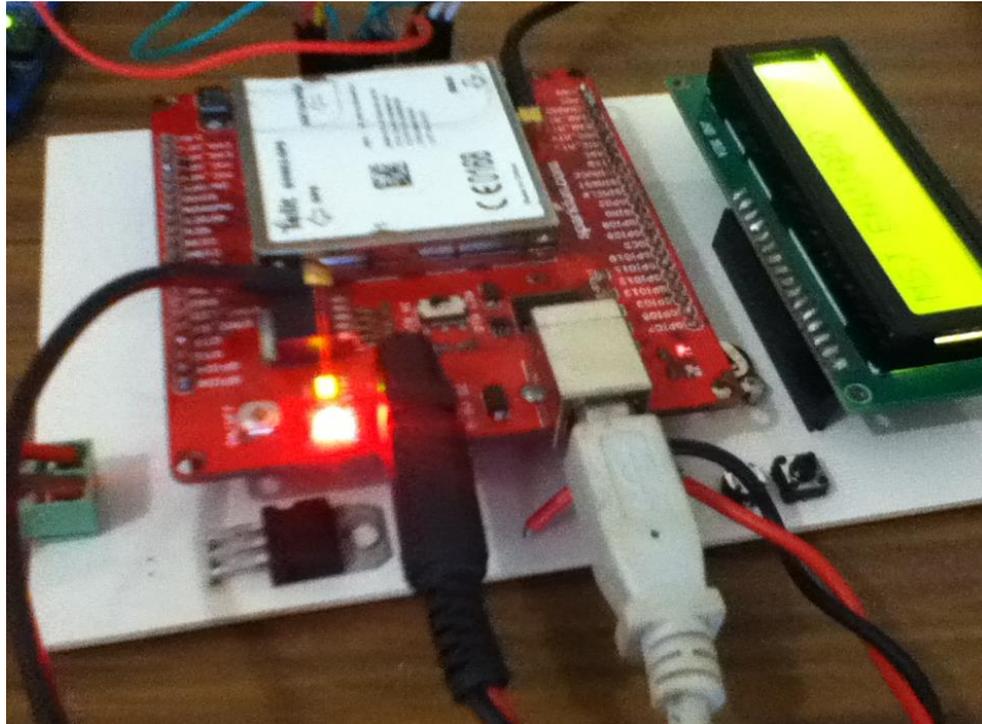


Figura 15- Módulo Telit GM862-GPS, Insertado en la tarjeta de pruebas

## 9.2 COMUNICACIÓN SERIAL DE COMANDOS AT

Se Realizó una serie de pruebas entre el hyperterminal y el módulo GM862-GPS, para comprobar que el modulo es compatible a los comando AT.

**Tabla 9.2- Muestra los comandos de configuración del módulo gm862-gps.**

COMANDOS A T	FUNCIÓN
AT	Hace que el dispositivo establezca la velocidad adecuada. También establece el formato de caracteres en el puerto serie.
AT + GMI	Nos proporciona la marca de nuestro dispositivo.
AT+GMM	Nos proporciona el modelo de nuestro dispositivo
AT+IPR=(300,1200,2400,9600.....57600)	Asignamos la velocidad del puerto de transmisión.
AT+CMME=(0,1)	0= código de error numérico 1= código de error en formato detallado
AT+CPIN?	Consulta la presencia de la tarjeta SIM
AT+CREG?	Comprueba si está listo para hacer llamadas
AT+CSQ	Comprueba el nivel de la señal recibida y calidad.
AT#MONI	Nos indica la compañía a la que estamos afiliados, calidad e intensidad.

Se realizó una simulación en proteus para comprobar módulo GM862-GPS, Funciona de manera correcta enviándole el comando AT para establecer el formato de caracteres en el puerto serie, después el comando AT+GMI el cual proporciona la marca del dispositivo. Después se envía AT+GMM proporciona el modelo del dispositivo, se configura a una velocidad de trasmisión con el comando AT+IPR=9600, se envía el comando AT+CPIN? Para verificar que el dispositivo contenga una tarjeta SIM si esto es verdadero, se podrá verificar la calidad de la señal con el comando AT+CSQ Y por último se le envía el comando AT#MONI la cual indica la compañía telefónica que pertenece.

### EJEMPLO.

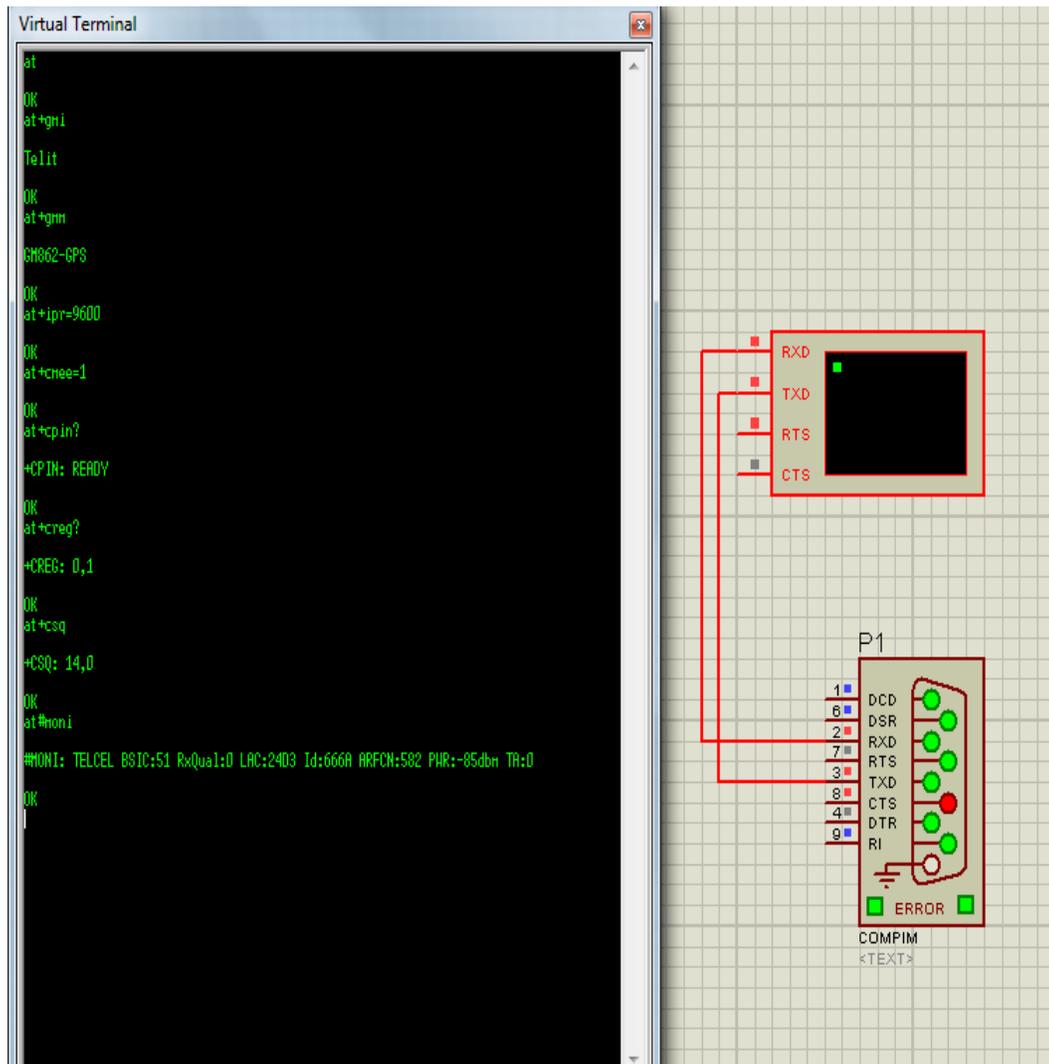


Figura .16.-Inicializacion del módulo Gm862-gps

## 9.2.1 MENSAJES DE TEXTO

El módulo también tiene la capacidad de poder enviar mensajes en modo texto o modo pdu (como una cadena de caracteres en hexadecimales o decimales), a continuación se presenta una serie de comandos para enviar mensajes, almacenarlos, e incluso leer mensajes guardados.

**Tabla 9.3 comandos para el envío de mensajes**

COMANDOS AT	FUNCION
AT+CMGF=(0,1)	Configuración de mensaje modo 0= modo pdu 1=modo Texto
AT+CMGW="NUMERO" + Ctrl-z	Guardar mensaje en el modulo
AT+CMGR="# DE MENSAJE"	Leer mensajes guardados en el modulo
AT+CMGS="NUMERO" + Ctrl-z	Enviar Mensaje
AT+CMGD="# DE MENSAJE"	Eliminar Mensaje
AT+WMSC="# DE MENSAJE"	Modificar el estado de un mensaje
AT+CMSS="# DEMENSAJE " + Ctrl-z	Envía mensajes almacenado en el modulo

## ENVIÓ DE MENSAJES

Por protocolo se debe realizar la configuración del módulo Gm862-gps como se muestra en la “figura. 9.5” para verificar que está funcionando de manera adecuada. La “figura 9.7” muestra el envío de mensajes. Para poder enviar mensajes de texto se realizó la siguiente secuencia de comandos. AT+CMGF=1 indica el tipo de mensaje a enviar en este caso se escogió mensajes de texto, después se introduce el comando AT+CMGS=”número telefónico” la cual despliega un espacio para poder escribir y por último se la da + Ctrl-Z. En la tabla 9.4 se puede ver otros comandos para eliminar, guardar o modifica los mensajes de texto.

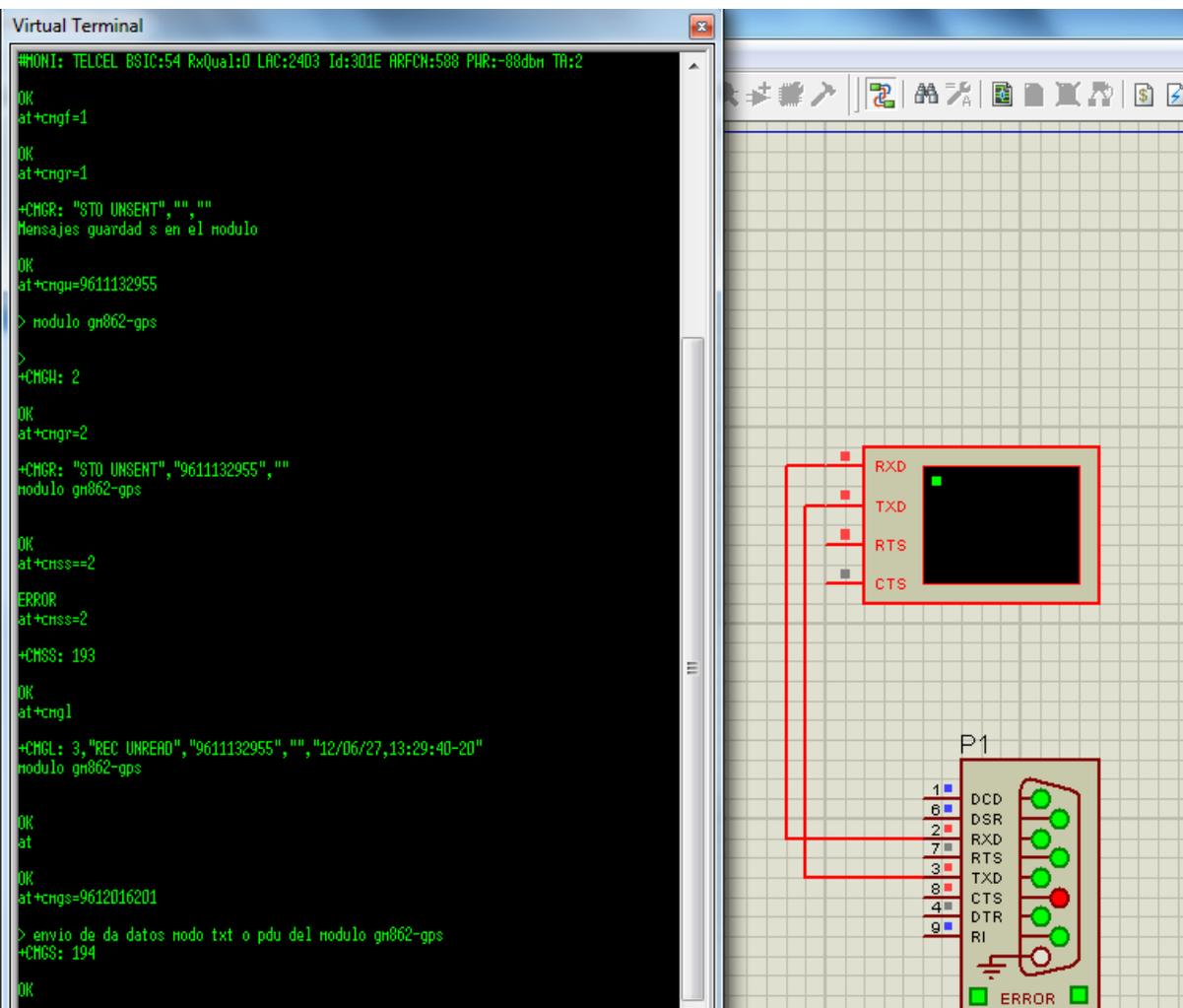


Figura.17.- envío de mensaje

### 9.2.2 GPS (SISTEMA DE POSICIONAMIENTO GLOBAL)

El módulo GM862- GPS contiene un comando que proporciona su posicionamiento global. El comando AT\$GPSACP es el más importante ya que se utiliza para pedir el posicionamiento global (coordenadas altitud, latitud y longitud). En la tabla se muestra los comandos más importantes y cada una de sus funciones.

**Tabla 9.4 comandos del sistema de posicionamiento global**

COMANDOS AT	FUNCIÓN
AT\$GPSACP	Posicionamiento global
AT\$GPSR=(0,1,2,3)	0=Restablecimiento del equipo, el receptor GPS se reinicia con los valores almacenados en la memoria. 1=Borra todos los datos almacenado en memoria 2=Borra todo los datos
AT\$GPSAT=(0,1)	0=Conectarse a la antena proporcionada por el modulo. 1=Antena GPS suministrada
AT#SHDN	El dispositivo se apaga con la siguiente secuencia de actividades, se desconecta de la red, y el modulo se apaga.

## EJEMPLO

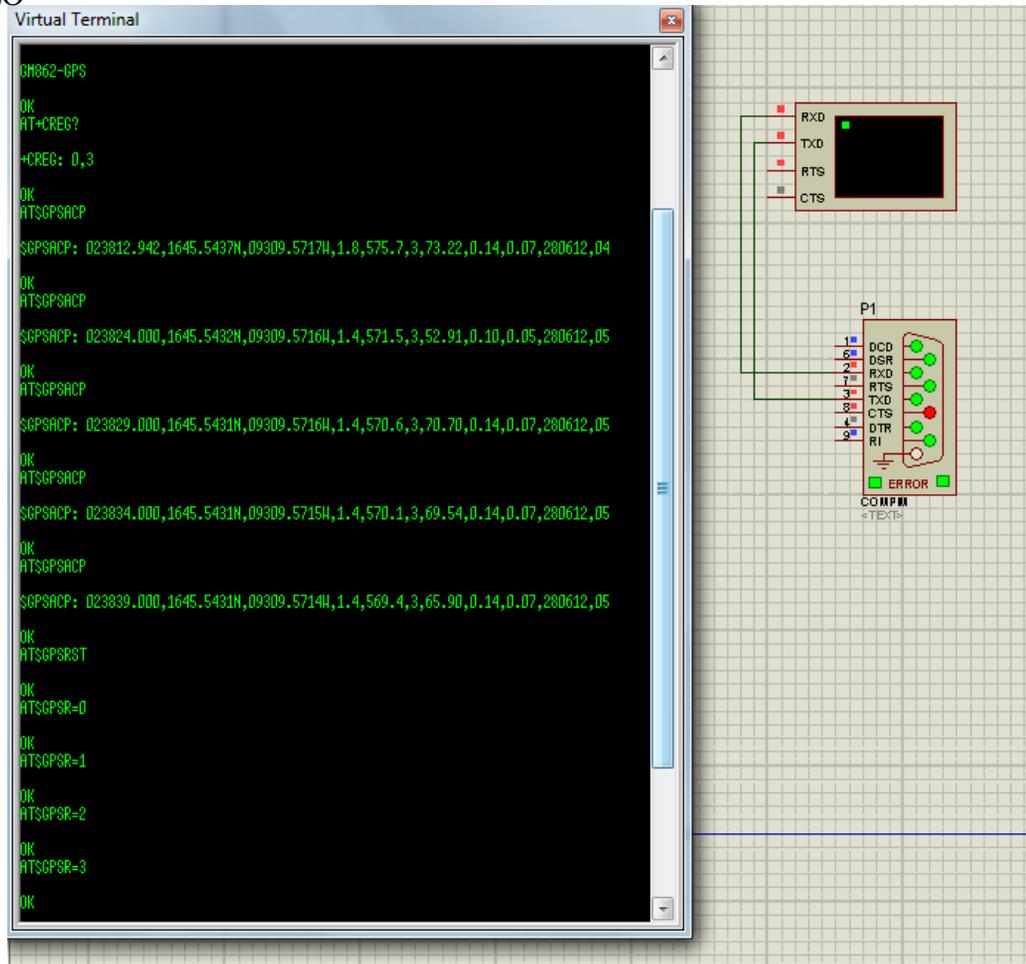


Figura.18.- Posicionamiento global

### 9.2.3 Visualización de Posicionamiento Global

A continuación se presenta el diseño del circuito principal “fig. 19”. El circuito está compuesto por un ARDUINO UNO, LCD y el módulo GM862-GPS.

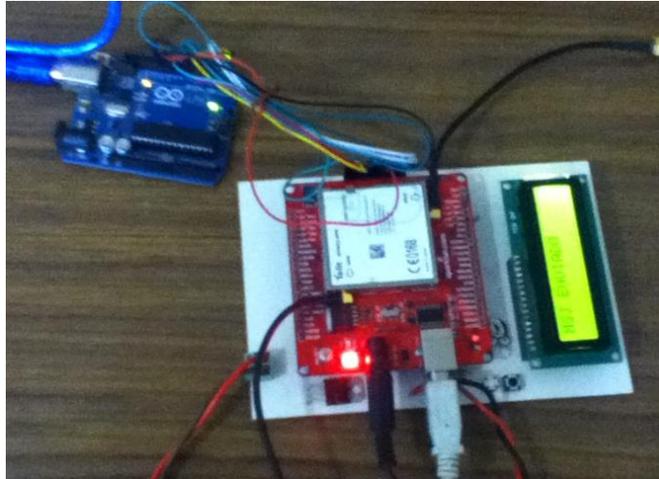


Fig. 19.-Muestra módulo Gm862-gps

Este circuito es diseñado para capturar la posición de GPS y mostrarla en una LCD.

El circuito trabaja de la siguiente manera. El arduino envía el comando “AT\$GPSACP” al módulo GM862-GPS. Este responde una cadena de caracteres, la cual es separa con el arduino, se toman los caracteres que interesaban (latitud y longitud), se desecha los caracteres que no serán útiles, se guarda la información en el arduino y por lo consiguiente se muestra en una LCD los caracteres que interesan. Ejemplo la “figura 9.11”



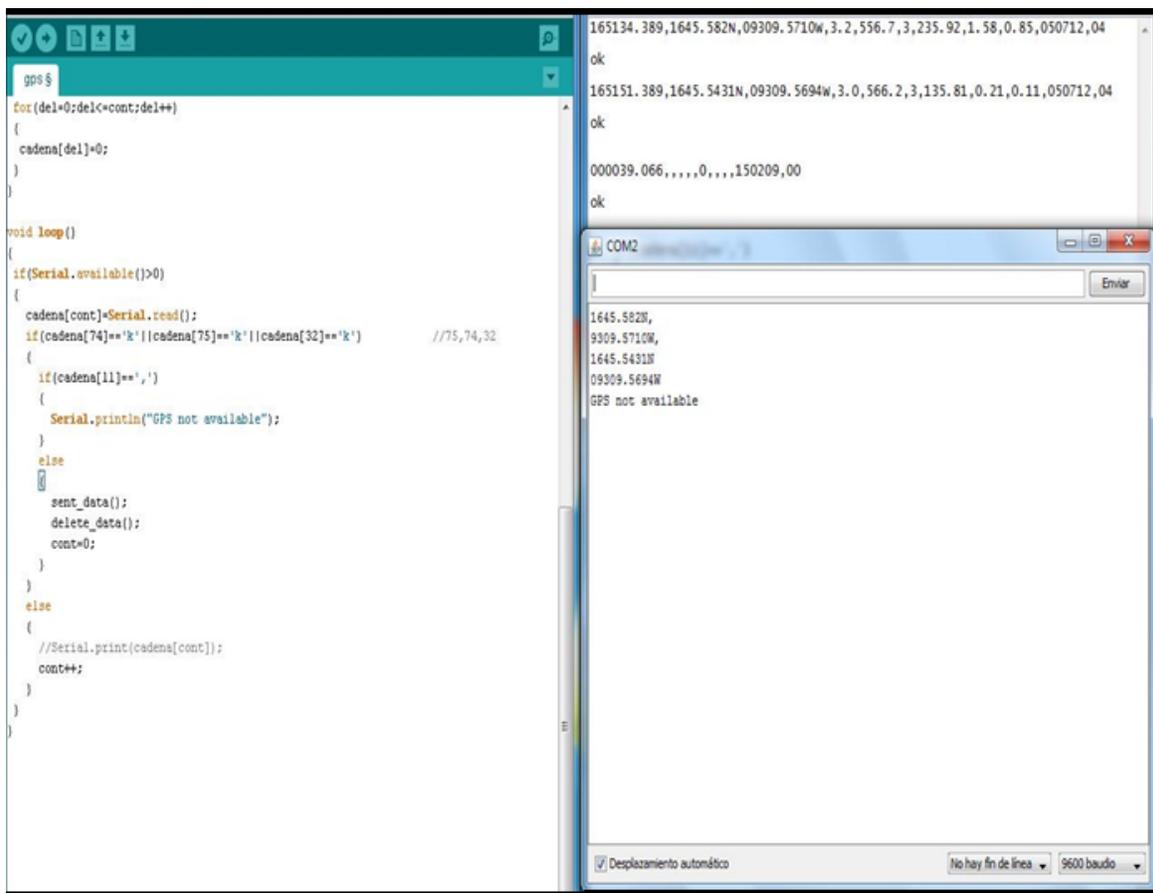


Fig. 21 separaciones de la cadena del Posicionamiento Global

En la “fig.21” se puede observar por medio de la terminal virtual del programa Arduino que el módulo Gm862-gps responde una cadena de caracteres (165134.389,1645.582N,09309.5710W,3.2,556.7,3,235.92,1.58,0.85,050712,04). El arduino separa los caracteres que nos interesan y los guarda en memoria (1645.582N,09309.5710W) que son latitud y longitud. Para posteriormente enviar por medio de un dispositivo móvil un mensaje de texto.

En la imagen se puede observar la cadena de caracteres (22541.000, 1646.1719N, 09306.5593W, 556.7,3, 235.92,1.58,0.85, 090114, 08 ). Se retiró los datos de Latitud, longitud y se muestra en una LCD. Además es guardado en la memoria del Arduino por si se desea enviar la posición a otro dispositivo móvil.



Figura 22.-LCD. Muestra la longitud y latitud

## 9.2.4 INTERPRETACIÓN DE CADENA DE CARACTERES DEL GPS

**Observación:** El posicionamiento global fue tomado bajo buenas condiciones como son: buen estado del tiempo, En un lugar despejado alejado de interferencias. Para tener una mayor exactitud en nuestra posición se recomienda realizar las pruebas por las tardes o noches debido a que en ese momento, los satélites tienen una mejor respuesta. En la (figura 9.12) podemos observar la conversión de nuestros grados decimales, grados, a minutos decimales o Al sistema de coordenadas Universal. Para realizar las conversiones se utilizó una aplicación del sistema de coordenadas Universal, pero para poder ingresarlas debemos recorrer 2 dígitos el punto de la longitud y latitud.

\$GPSACP: 22541.000, 1646.1719N, 09306.5593W, 556.7,3, , 090114, 08

**Tabla 9.5 Interpretación de cadena de caracteres de GPS.**

NOMBRE	EJEMPLO	DESCRIPCIÓN
TIEMPO	22541.000	<b>Hhmmss</b> (los primeros dos dígitos indican las horas, los siguientes dos indican los minutos y los dos últimos dígitos los segundos.
Latitud	1646.1719	<b>ddmm.mmmm</b> (los primeros dos dígitos indican los grados y los siguientes indican los minutos decimales
N/S INDICADOR	N	(Norte)
LONGITUD	09306.5593	<b>ddmm.mmmm</b> los primeros dos dígitos indican los grados y los siguientes indican los minutos decimales
Altitud	556.7	<b>mmm.d</b> los primeros dígitos indican los metros y los siguientes indican las decimas de metro.
E/W INDICADOR	W	(oeste )
POSICION DE REVISION	3	
FECHA	09/01/14	<b>D/M/A</b> (día, mes, año)
# SATELITE	08	(número del satélite al que está conectado)

Tabla 9.7 Muestra la interpretación de la cadena de posicionamiento global.

Ejemplo.

1645.5427N, 09309.5693W.....original

16.455427N, 093.095693W.....modificada

## 9.2.5 Conversiones

---

Results using Map Datum *NAD83/WGS84*

GRADOS, MINUTOS , SEGUNDOS

Latitude: 16° 27' 19.530" N  
Longitude: 93° 5' 44.495" E

GRADOS DECIMALES

Latitude: 16.455425  
Longitude: 93.095693

**Universal Transverse Mercator (UTM):**

Zone: 46  
Easting: 510215  
Northing: 1819315

MINUTOS DECIMAL

Latitude: 16°27.3255000000001  
Longitude: 93°5.741579999999983

---

Fig. 22 En la imagen se puede observar 4 tipos de conversiones de grado decimales, minutos decimales, grados UTM, grados. Todo depende del país donde se encuentre, en México de acuerdo a sus normas y estándares de medición se utilizan los grados decimales.

## 9.2.6 GOOGLE MAPS

Google maps es una aplicación que nos sirvió para poder visualizar de manera gráfica nuestro posicionamiento global. Pero para ello necesitamos conocer la longitud y latitud de nuestra cadena de caracteres del posicionamiento global. se realiza una pequeña modificación a la latitud y longitud, recorriendo 2 dígitos hacia la izquierda en cada una, la longitud se le introduce un signo negativo debido a que el Oeste en muchos mapas es substituida por una W y puede ser también expresada usando un signo negativo ya que es una longitud del Hemisferio Occidental. La parte modificada se introduce al google maps como verán en la figura

\$GPSACP: 165158.389, 1645.5427N, 09309.5693W, 3.0, 050712,04.....normal

\$GPSACP: 165158.389, 16.455427N, -093.095693W, 3.0, 050712,04.....modificado

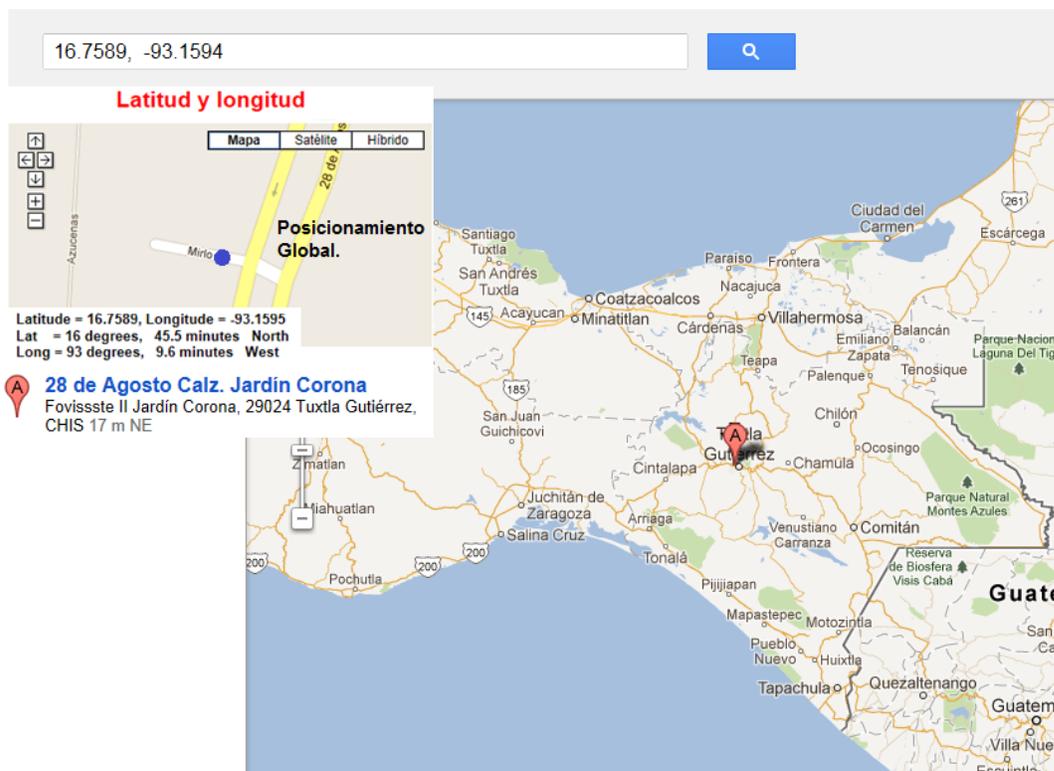


Figura .23.- Muestra el posicionamiento global de manera grafica

### 9.2.7 SE REALIZARON 2 PRUEBAS PARA CORROBORAR QUE EL MODULO GM862-GPS FUNCIONA DE LA MANERA CORRECTA.

La primera prueba fue realizada en el Instituto Tecnológico de Tuxtla Gutiérrez en la cual se obtuvieron los siguientes datos.

Tabla 9.6

Primer día	Tiempo	Posición N(latitud)	Posición W (longitud)	posición, altitud ,metros	Velocidad en k	fecha	Satélite
1	164436.000,	1645.4158N,	09310.3329W,	556.7,3,	235.92,1.58,0.85,	131213,	09
2	164436.000,	1645.4158N,	09310.3329W,	556.7,3,	235.92,1.58,0.85,	131213,	09
3	164438.000,	1645.4158N,	09310.3327W,	556.7,3,	235.92,1.58,0.85,	131213,	09
4	164436.000,	1645.4158N,	09310.3329W,	556.7,3,	235.92,1.58,0.85,	131213,	09
5	164436.000,	1645.4160N,	09310.3329W,	556.7,3,	235.92,1.58,0.85,	131213,	09

1.- consiste en obtener los datos (latitud y longitud)



Figura.24.-LCD. Muestra datos obtenidos

**2.- Se guarda en la memoria del arduino los datos obtenidos.**

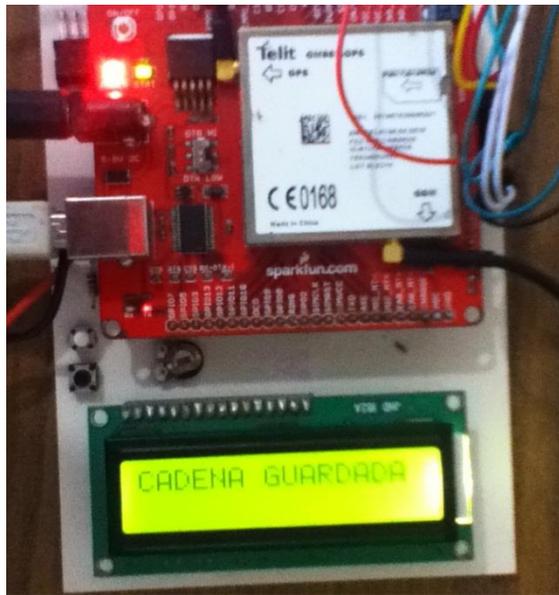


Figura .25.-LCD. Muestra que los datos se han guardado

**3.-Se envía los datos envía los datos de la cadena guarda por medio de un mensaje de texto a un dispositivo móvil**

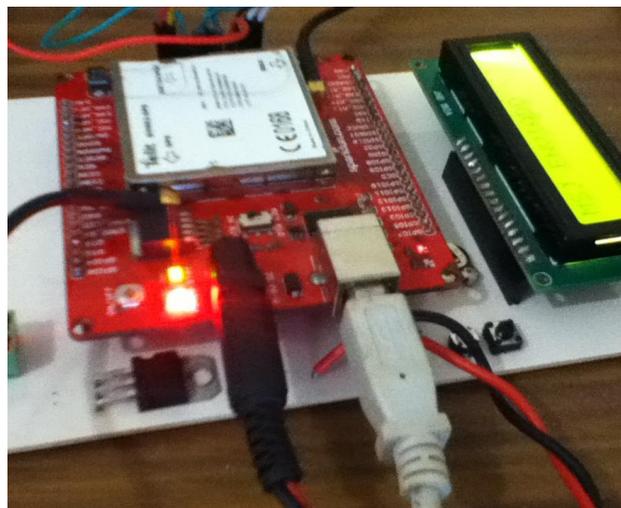


Figura.26.-LCD. Muestra que el mensaje de texto se ha enviado correctamente

#### 4.- Muestra los datos obtenidos en el dispositivo móvil.

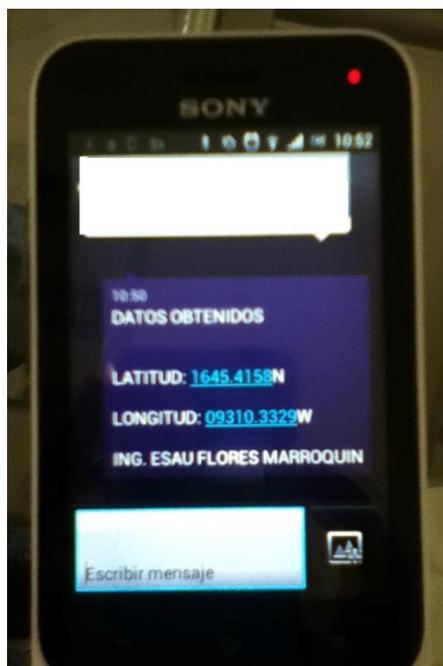


Figura.27.-LCD. Muestra que el mensaje de texto se ha enviado correctamente

## 5.-ingresamos latitud y longitud a google maps

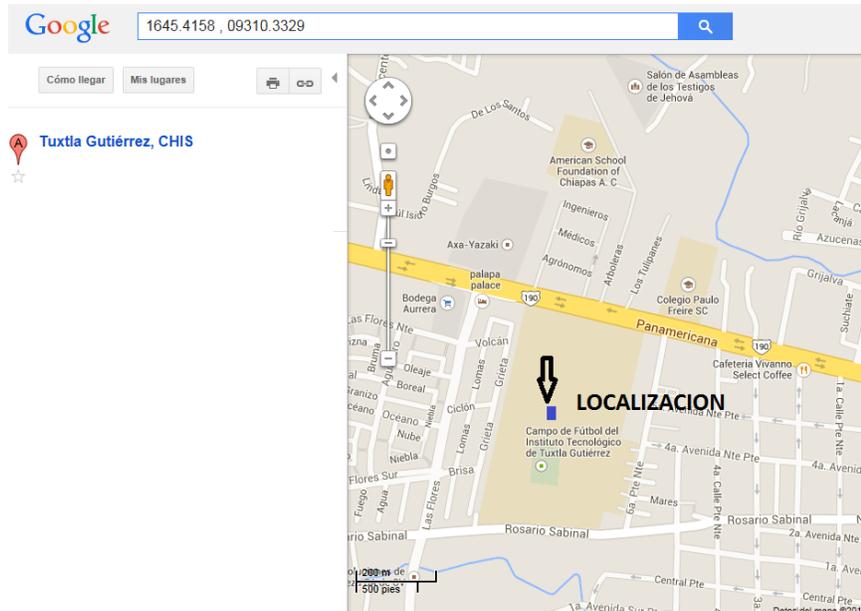


Figura .28.-Muestra la localización de nuestro Dispositivo

**La segunda prueba fue realizada en la 4ta Oriente y 17 norte en la cual se obtuvieron los siguientes datos.**

**Tabla 9.7**

Primer día	Tiempo	Posición N(latitud)	Posición W (longitud)	posición, altitud ,metros	Velocidad en k	fecha	Satélite
1	22541.000,	1646.1751N,	09306.5576W,	556.7,3,	235.92,1.58,0.85,	090114,	08
2	22541.000,	1646.1719N,	09306.5594W,	556.7,3,	235.92,1.58,0.85,	090114,	08
3	22541.000,	1646.1719N,	09306.5593W,	556.7,3,	235.92,1.58,0.85,	090114,	08
4	22541.000,	1646.1718N,	09306.5594W,	556.7,3,	235.92,1.58,0.85,	090114,	08
5	22541.000,	1646.1719N,	09306.5594W,	556.7,3,	235.92,1.58,0.85,	090114,	08

**1.- consiste en la separación de la cadena y obtener los datos (latitud y longitud)**



Figura .29.-LCD. Muestra datos obtenidos

2.- Se guarda en la memoria del arduino los datos obtenidos.



Figura .30.-LCD. Muestra que los datos se han guardado

3.-Se envía los datos de la cadena guarda por medio de un mensaje de texto a un dispositivo móvil

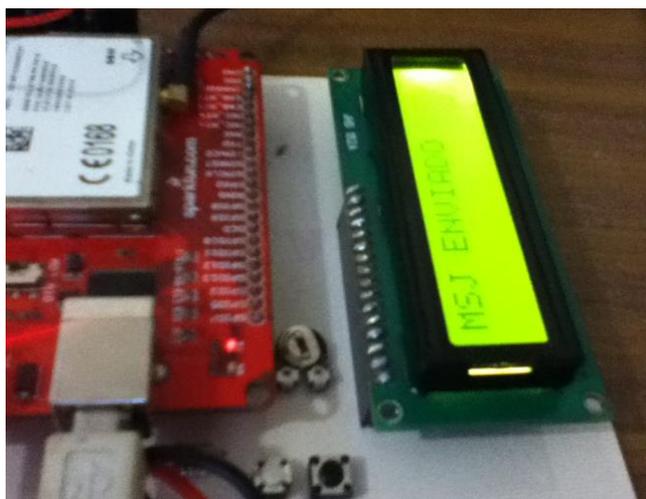


Figura .31.-LCD. Muestra que el mensaje de texto se ha enviado correctamente

4.- Se visualiza los datos enviados por el módulo Gm862-gps en nuestro dispositivo móvil.

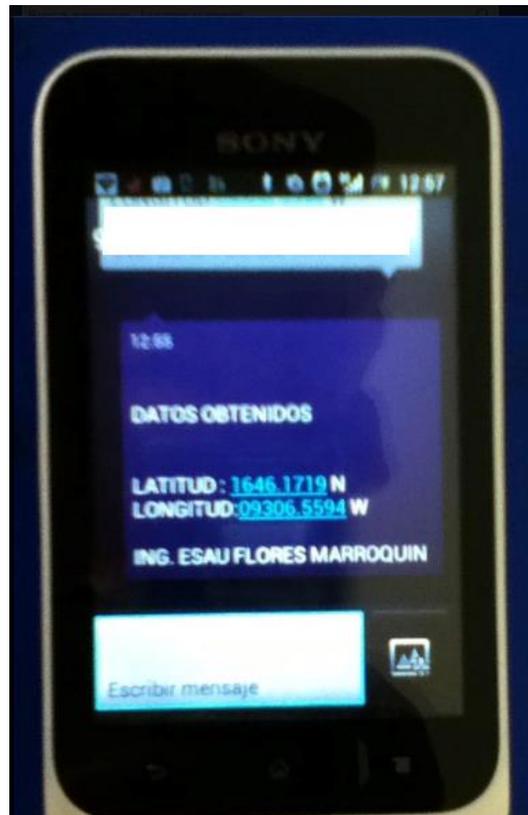


Figura.32.- El dispositivo celular muestra los datos (latitud y longitud).

## 5.-Se introduce los datos de latitud y longitud a google maps

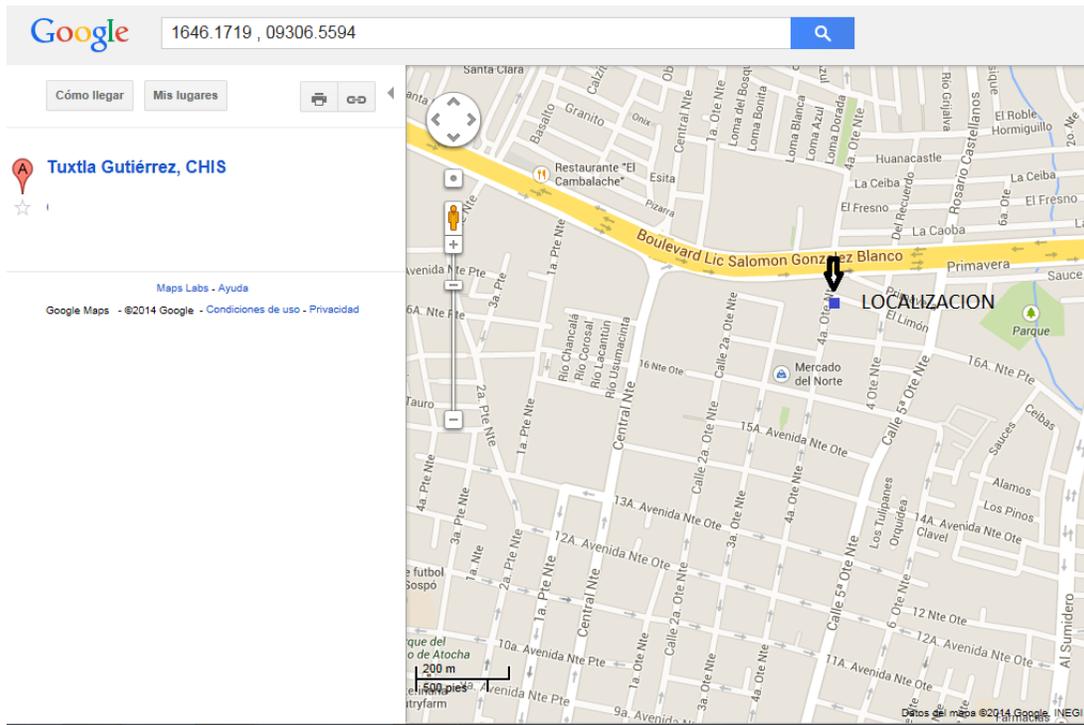


Figura .33.-Muestra la localización de nuestro Dispositivo

## Análisis para la reducción de error del posicionamiento global.

Se realizaron pruebas necesarias para obtener el margen de error del dispositivo GPS. En este caso les mostrare el análisis de 36 pruebas obtenidas.

**Tabla 9.8**

Primer día	Tiempo	Posición N(latitud)	Posición W (longitud)	posición, altitud ,metros	Velocidad en k	fecha	Satélite
1	165134.389,	1645.5421N,	09309.5610W,	556.7,3,	235.92,1.58,0.85,	050712,	04
2	165151.389,	1645.5431N,	09309.5694W,	566.2,3,	135.81,0.21,0.11,	050712,	04
3	165155.389,	1645.5425N,	09309.5694W,	566.1,3,	170.86,0.75,0.40,	050712,	04
4	165158.389,	1645.5427N,	09309.5693W,	566.2,3,	172.61,0.93,0.50,	050712,	04
5	165224.389,	1645.5428N,	09309.5687W,	564.4,3,	131.29,0.39,0.21,	050712,	04
6	165228.389,	1645.5430N,	09309.5688W,	564.8,3,	126.29,0.43,0.23,	050712,	04
7	165241.389,	1645.5433N,	09309.5695W,	566.4,3,	132.52,0.32,0.17,	050712,	04
8	165232.389,	1645.5431N,	09309.5690W,	565.4,3,	132.34,0.39,0.21,	050712,	04
9	165324.389,	1645.5437N,	09309.5603W,	568.2,3,	127.97,0.43,0.23,	050712,	04
10	165316.389,	1645.5437N,	09309.5603W,	568.1,3,	132.86,0.39,0.21,	050712,	04
11	165302.389,	1645.5436N,	09309.5601W,	568.0,3,	131.54,0.39,0.21,	050712,	04
12	165247.389,	1645.5434N,	09309.5697W,	567.1,3,	128.82,0.32,0.17,	050712,	04
13	190610.000,	1645.5417N,	09309.5623W,	557.7,2,	58.07,1.15,0.62,	050712,	03
14	190615.000,	1645.5412N,	09309.5615W,	558.0,2,	237.90,0.75,0.40,	050712,	03
15	190619.000,	1645.5410N,	09309.5609W,	558.3,2,	256.39,0.18,0.09,	050712,	03
16	190622.000,	1645.5409N,	09309.5607W,	558.4,2,	247.35,0.28,0.15,	050712,	03
17	190625.000,	1645.5409N,	09309.5608W,	558.5,2,	46.58,0.18,0.09,	050712,	03
18	190630.000,	1645.5410N,	09309.5699W,	559.1,2,	169.23,0.18,0.09,	050712,	03
19	190638.000,	1645.5407N,	09309.5666W,	561.7,2,	66.54,1.65,0.89,	050712,	03
20	190656.000,	1645.5410N,	09309.5660W,	563.1,2,	205.00,0.18,0.09,	050712,	03
21	190702.000,	1645.5415N,	09309.5639W,	564.6,2,	97.59,0.39,0.21,	050712,	03
22	190706.000,	1645.5414N,	09309.5636W,	565.0,2,	97.97,0.36,0.19,	050712,	03
23	190709.000,	1645.5414N,	09309.5636W,	565.1,2,	97.18,0.25,0.13,	050712,	03
24	190715.000,	1645.5412N,	09309.5636W,	565.8,2,	119.53,0.28,0.15,	050712,	03
25	165134.389,	1645.5822N,	09309.5710W,	556.7,3,	235.92,1.58,0.85,	050712,	04
26	165151.389,	1645.5431N,	09309.5694W,	566.2,3,	135.81,0.21,0.11,	050712,	04
27	165155.389,	1645.5425N,	09309.5694W,	566.1,3,	170.86,0.75,0.40,	050712,	04
28	165158.389,	1645.5427N,	09309.5693W,	566.2,3,	172.61,0.93,0.50,	050712,	04
29	165224.389,	1645.5428N,	09309.5687W,	564.4,3,	131.29,0.39,0.21,	050712,	04
30	165228.389,	1645.5430N,	09309.5688W,	564.8,3,	126.29,0.43,0.23,	050712,	04
31	190638.000,	1645.5407N,	09309.5666W,	561.7,2,	66.54,1.65,0.89,	050712,	03
32	190656.000,	1645.5410N,	09309.5660W,	563.1,2,	205.00,0.18,0.09,	050712,	03
33	190702.000,	1645.5415N,	09309.5639W,	564.6,2,	97.59,0.39,0.21,	050712,	03
34	190706.000,	1645.5414N,	09309.5636W,	565.0,2,	97.97,0.36,0.19,	050712,	03
35	190709.000,	1645.5414N,	09309.5636W,	565.1,2,	97.18,0.25,0.13,	050712,	03
36	190715.000,	1645.5412N,	09309.5636W,	565.8,2,	119.53,0.28,0.15,	050712,	03

De acuerdo a la tabla se puede observar que existe una pequeña variación en los minutos en las centesimales de la (longitud y latitud). Para realizar este pequeño análisis tuvimos que separar los datos de longitud y latitud, las primeras 2 cifras nos indican que son los grados, las 2 cifras siguientes indican los minutos, después del punto se encuentran 4 dígitos el primero indica los minutos decimales, el segundo indica los minutos centesimales, el tercero indica minutos millesimales y el cuarto indica los minutos diezmillesimales. Se realizó un análisis para saber su margen de error de esta pequeña variación.

## Análisis de resultados.

### Latitud:

Debido a que existe una pequeña variación en los minutos millesimales de Latitud. Se realizó un análisis para poder deducir el margen de error. Se sumó todas las latitudes de las 36 pruebas tomadas y con ello sacar una latitud promedio.

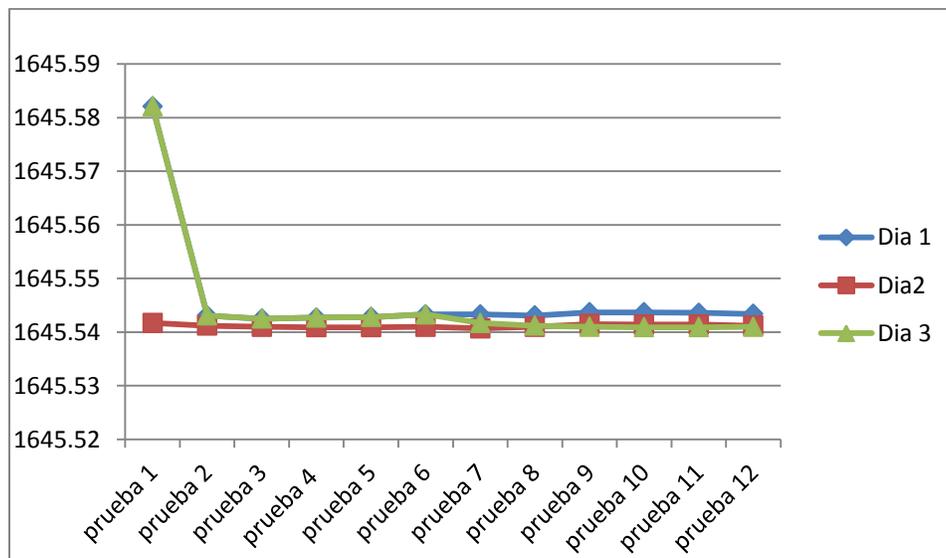


Figura .34.- Muestra la variación de la longitud en 36 pruebas obtenidas

$$\text{valor promedio de la latitud} = \frac{\text{suma de las latitudes}}{36} = 59239.5944/36 = 1645.544289$$

$$\begin{aligned} \text{Error Absoluto} &= \text{Valor Real} - \text{Valor promedio} &= 1645.5421 - 1645.544289 \\ &= -0.002189 \end{aligned}$$

$$Error\ Relativo = \frac{Error\ Absoluto}{Valor\ Real} = \frac{-0.002189}{1645.5421} = -0.0001330$$

$$Error\ Relativo\ Porcentual = Error\ Relativo * 100 = -0.0001330 * 100 \\ = -0.01330 \%$$

De acuerdo a los análisis realizados se dedujo que existe una variación de 0.01330 % más o menos de error porcentual. Con esto se puede decir que existe un error significativo. El valor absoluto de la coordenada de latitud fue tomada de la aplicación google maps, para poder tener un valor promedio de la latitud se tomaron 36 muestras y se obtuvo un promedio de ellas. Y al final se observó por medio de una comparación en google maps que existen un margen de error muy pequeño de 0 a 10 metros a la redonda.

## Longitud:

De acuerdo a los análisis obtenidos se dedujo que existe una variación de más de 0.00005306% o menos de error porcentual en longitud. Debido a que el Posicionamiento Global tiene pequeñas variaciones en los minutos millesimales, para poder tener una aproximación adecuada se introdujo las coordenadas promedio de (longitud y latitud) y las coordenadas máximas de (longitud y latitud) en la aplicación de Google Maps. Se observó que existe una variación de 0 a 10 metros a la redonda aproximados.

Se consideró los errores de longitud y latitud, para poder dar un área aproximada de nuestro robot de trabajo colaborativo.

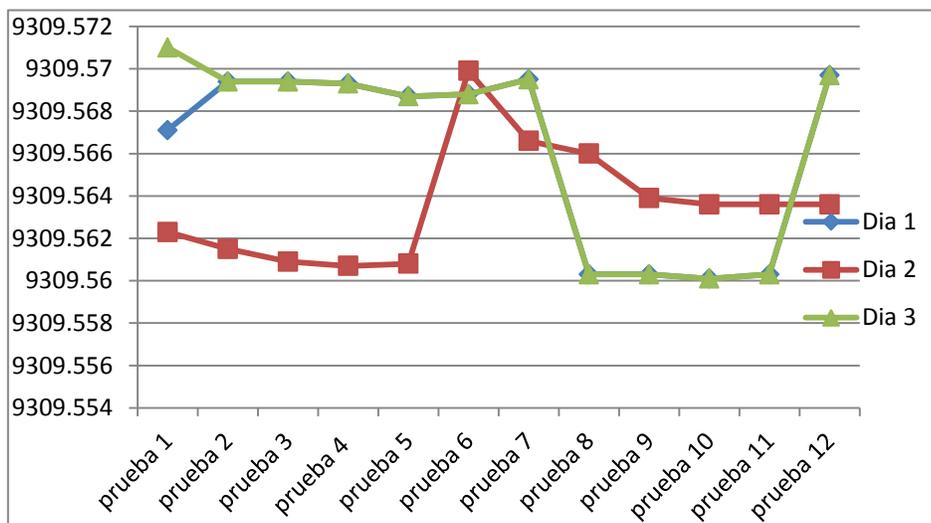


Figura.35.- Muestra la variación de la latitud en 36 pruebas obtenidas

Para obtener el error porcentual aproximado, se realizó la suma promedio de todas las latitudes y después se comparó con la latitud de valor real,

$$\text{Promedio de la longitud} = \frac{\text{suma de todas las longitudes}}{36} = \frac{335144.3531}{36} = 9309.565364$$

$$\text{Error Absoluto} = \text{Valor Real} - \text{Valor promedio} = 9309.5703 - 9309.565364 = 0.00494$$

$$\text{Error Relativo} = \frac{\text{Error Absoluto}}{\text{Valor Real}} = \frac{0.00436}{9309.5703} = 0.000005306$$

$$\text{Error Relativo Porcentual} = 0.00005306\%$$

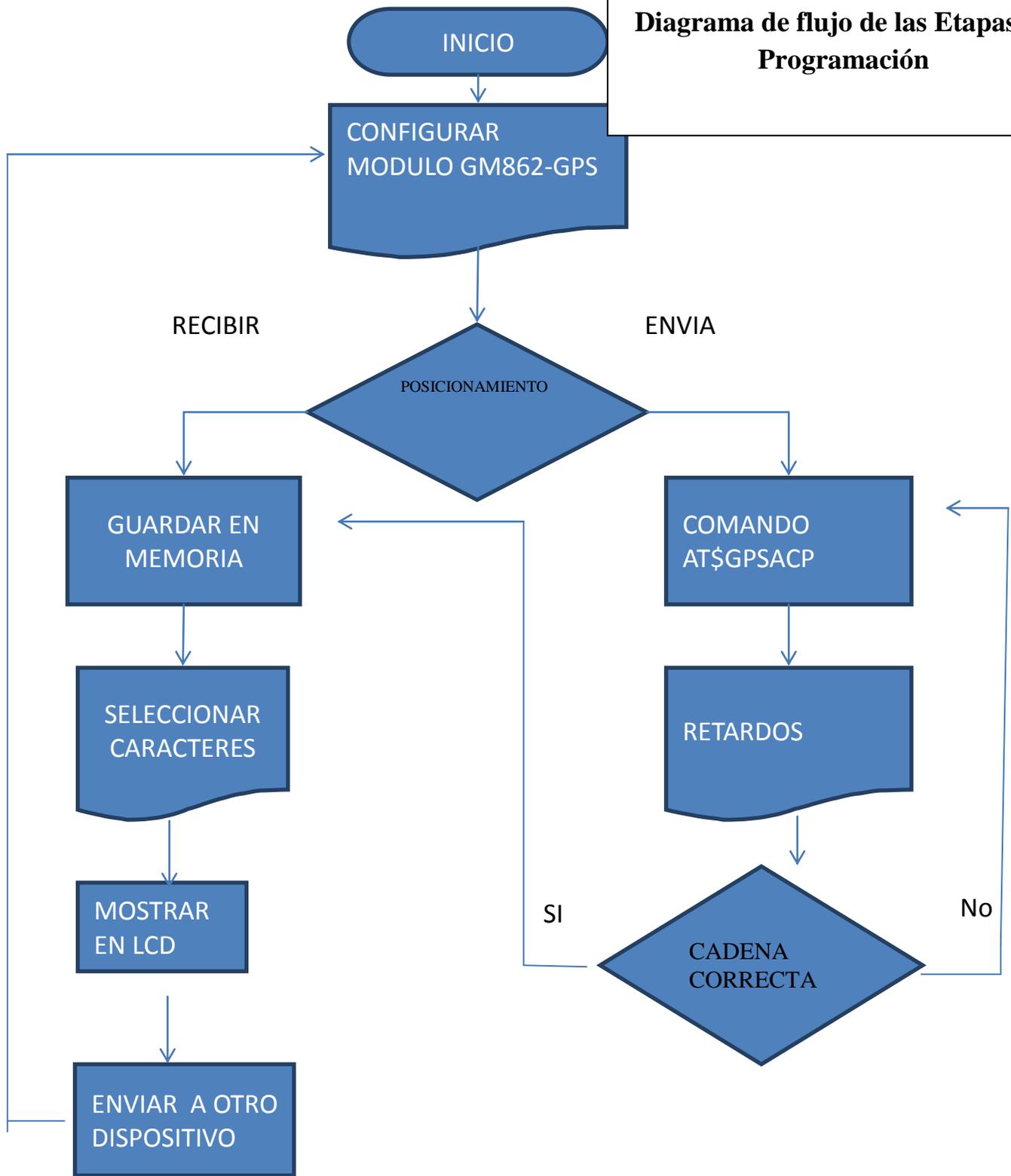
Este análisis no sirve para poder tener una idea más amplia y aproximada del robot de trabajo colaborativo y tomar en cuenta que existe un margen de error de 10 m a la redonda, Este error se podría eliminar con un GPS de pago ya que estos tienen una mayor precisión de su posicionamiento global.

### 9.3 Etapas de Programación

A continuación se presenta las etapas de programación y el diagrama de flujo. Las etapas de programación están basadas en la lógica siguiente:

- ✓ 1.- Configurar el módulo GM862-GPS. (Velocidad de transmisión de datos) desde el arduino.
  
- ✓ 2.- Enviar el comando At\$gpsacp para pedir la posición global
  
- ✓ 3.- Guardar los datos obtenidos en una memoria interna del arduino.
  
- ✓ 4.- Separar la cadena de caracteres y tomar los caracteres que interesan.
  
- ✓ 5.- Poder analizar los datos detalladamente y enviarlo a otro dispositivo móvil.

# Diagrama de flujo de las Etapas de Programación



### 9.3.1 Implementación del módulo Gm862-gps en robot de trabajo colaborativo

Para implementar el módulo gm862-gps al robot de trabajo colaborativo. Se tuvo que realizar algunas modificaciones al robot como fue la anexión de otro piso. Ya que en su parte inferior estaba ocupada por sensores y la batería encargada de alimentar al robot de trabajo colaborativo. El piso que se le anexo estaba constituido de una mica resistente, además se le puso unos tornillos para ajustar el piso y a la vez elevarlo, para que no tuviera interferencias con los sensores y la batería

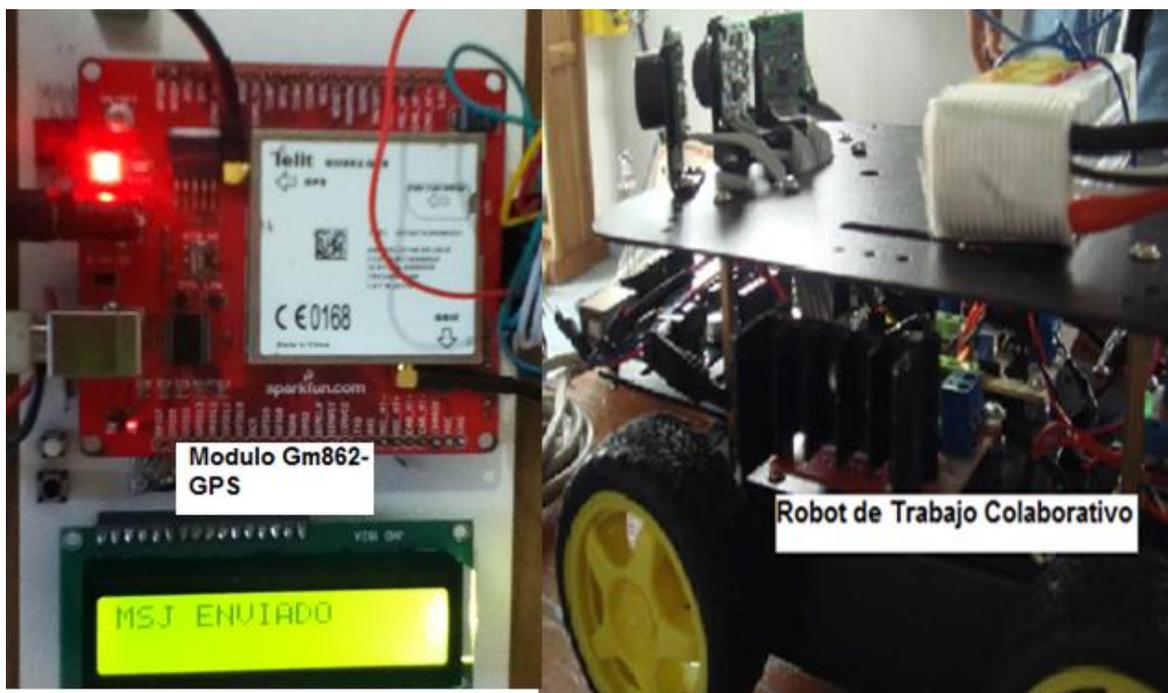


Figura 36.- Muestra el módulo GM862-GPS Y el robot de Trabajo colaborativo

A continuación se puede observar los arreglos realizados al robot de trabajo colaborativo, las modificaciones más importantes son las siguientes: el módulo gm862-gps va conectada

vía serial al arduino del robot de trabajo colaborativo. El arduino y el módulo gm862-gps va conectada a la fuente voltaje del robot de trabajo (batería Lipo).

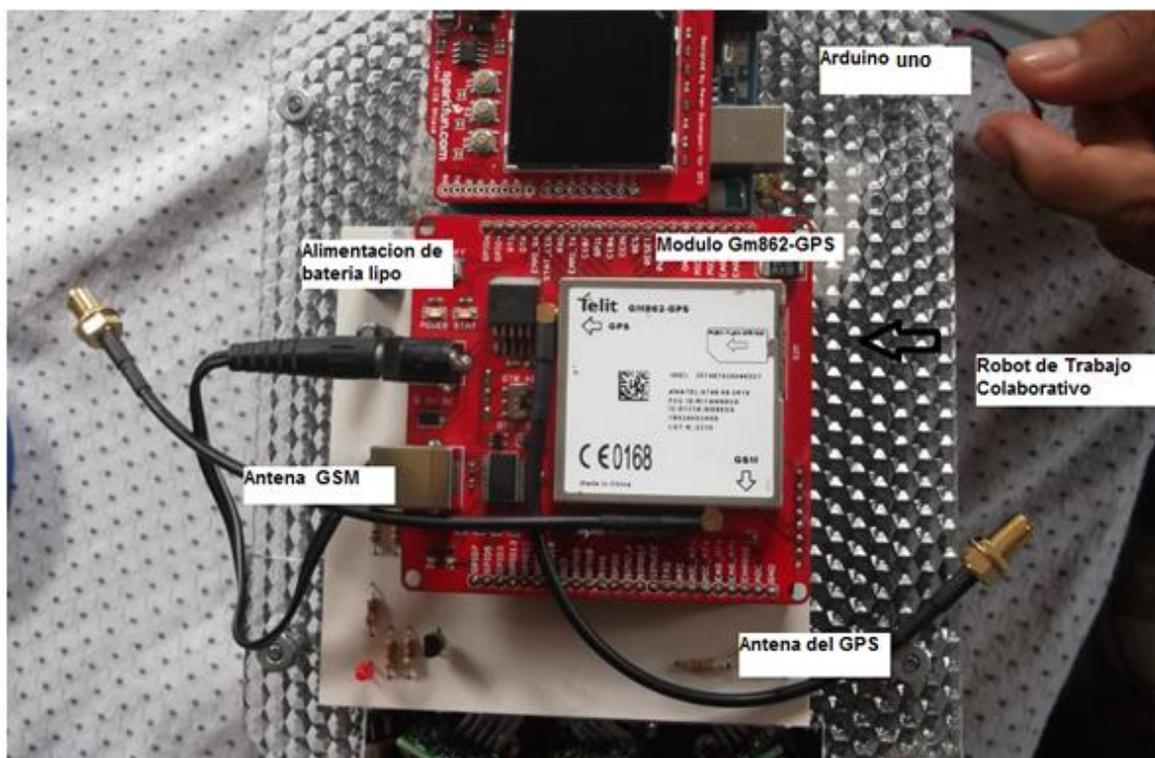


Figura. 37.- Muestra la implementación del módulo GM862-GPS en robot de trabajo Colaborativo

## 10.- CONCLUSIÓN

Los resultados que se obtuvieron físicamente fueron satisfactorios cumpliendo las expectativas buscadas:( Comunicación serial de comandos AT, Reducir el margen de error de nuestro dispositivo GPS, Diseñar el hardware para las pruebas físicas, Realizar un programa que sea capaz de guardar la posición y enviarlo a otro dispositivo). La prueba del sistema nos arrojó resultados satisfactorios ya que se logró obtener resultados similares a los de la simulación por software, aunque se tuvieron algunos problemas con respecto a la reacción del módulo Gm862-gps, debido a que en ocasiones tarda demasiado en proporcionar el posicionamiento global. Estos problemas se corrigieron de tal forma que el circuito funcione de la mejor manera posible. Cabe mencionar que el programa desarrollado fue para visualización y envío del posicionamiento global, de nuestro robot de trabajo colaborativo, para poderlo monitorear.

Con estos resultados obtenidos se concluyó que se realizó lo que se buscaba cumpliendo así los objetivos planteados.

## RECOMENDACIONES

Las recomendaciones de uso del sistema son las siguientes:

- 1.- Es importante verificar los voltajes de alimentación del sistema. Los cuales son +5 Volts, o por defecto +9 Volts y tierra, siempre y cuando estos vayan a nuestra tarjeta principal.
- 2.- Verificar que la tarjeta principal, entregue 3.8 V a nuestro modulo, para que trabaje adecuadamente y no sufra ningún daño.
- 3.- Si el sistema no inicializa correctamente, hay que proceder a reinicializarlo.
- 4.- Verificar el botón de encendido de nuestro módulo Gm862- GPS
- 5.-Alimentar nuestro módulo arduino con +5 V y en defecto +9V.
- 6.- Verificar la comunicación serial entre el arduino y nuestro módulo Gm862-gps.

Con respecto a la programación se pueden mejorar las funciones con un arduino Shell. Ya que este está diseñado especialmente para nuestro módulo gm862-gps.

## 12.- BIBLIOGRAFÍA

- 30 Arduino Projects for the Evil Genius  
Autor: Simon Monk  
Ed. Prentice-Hall International
- Manual Telit Gm862-gps  
Teléfonos, celulares y módulos Telit
- Manual de Comandos AT  
Autor: Dennis Hayes
- Manual de GPS Garmin  
Autor: Carlos Punch

Citas de internet

[http://www.telit.com/en/products.php?p\\_id=3&p\\_ac=show&p=7](http://www.telit.com/en/products.php?p_id=3&p_ac=show&p=7)

<http://bluehack.elhacker.net/proyectos/comandosat/comandosat.html>

[http://es.wikipedia.org/wiki/Sistema\\_de\\_posicionamiento\\_global](http://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global)

<http://www.arduino.cc/es/>

[http://www.arduino.cc/es\\_old/Tutoriales/GPS](http://www.arduino.cc/es_old/Tutoriales/GPS)

<http://www.telit.com/>

[http://es.wikipedia.org/wiki/Conjunto\\_de\\_comandos\\_Hayes](http://es.wikipedia.org/wiki/Conjunto_de_comandos_Hayes)

## ANEXOS

Código de Separación de datos (latitud y longitud) de la cadena de posicionamiento Global

```
char cadena[80]; // cadena de caracteres (array) , donde se guarda cada uno de los bytes que entrega el gps//
```

```
byte cont=0;// variable auxiliar utilizada como apuntador para especificar en que parte de la cadena se guardan los Bytes recibidos//
```

```
void setup() // Función para la inicialización de variables, librerías , pines//  
{  
  Serial.begin(9600); // Configuración de la velocidad de transmision //  
}
```

```
void sent_data74(void) // Función que se encarga de procesar las cadenas de 74 caracteres //
```

```
{  
  byte cont_lat; // Variable auxiliar para registrar la dirección donde se guardara latitud//
```

```
byte cont_long; // Variable auxiliar para registrar la dirección donde se guardara longitud//
```

```
for(cont_lat=11;cont_lat<=19;cont_lat++) // en este ciclo se envía los datos de la latitud//
```

```
{  
  Serial.print(cadena[cont_lat]);  
}  
Serial.println();
```

```
for(cont_long=21;cont_long<=31;cont_long++)// en este ciclo se envía los datos de la longitud//
```

```
{  
  Serial.print(cadena[cont_long]);  
}  
Serial.println();
```

```

}

void sent_data75(void) // Función que se encarga de procesar las cadenas de 75
caracteres //
{
  byte cont_lat; // Variable auxiliar para registrar la dirección donde se guardara
latitute//
  byte cont_long; // Variable auxiliar para registrar la dirección donde se guardara
longitud//

  for(cont_lat=11;cont_lat<=20;cont_lat++) // en este ciclo se envía los datos de la
latitute//
  {
    Serial.print(cadena[cont_lat]);
  }
  Serial.println();
  for(cont_long=22;cont_long<=32;cont_long++) //en este ciclo se envía los datos de la
longitud//
  {
    Serial.print(cadena[cont_long]);
  }
  Serial.println();
}

void delete_data() // esta función elimina los datos guardados en la cadena de caracteres
//
{
  byte del;
  for(del=0;del<=cont;del++)
  {
    cadena[del]=0;
  }
}

void loop()
{
  Serial.print("at$gpsacp"); // envía el comando at$gpsacp al modulo gm862-gps//
  delay (500);
  if(Serial.available()>0) // Función que detecta que ay datos en el buffer de l Puerto
serial del arduino//

```

```

{
  cadena[cont]=Serial.read();//guarda el dato recibido en el puerto //
  if(cadena[74]=='k' || cadena[75]=='k' || cadena[32]=='k') //cuando se reciba la
variable K, entran en función las subrutinas //
  {
    if(cadena[11]==',')// rutina de la cadena errónea //
    {
      Serial.println("GPS not available");
      delete_data();
      cont=0;
      delay(1000);
    }
    else
    {
      if(cadena[74]=='k')// rutina para cadena 74 datos //
      {
        sent_data74();
        delete_data();
        cont=0;
        delay(1000);
      }
      if(cadena[75]=='k')// rutina para cadena 75 datos //
      {
        sent_data75();
        delete_data();
        cont=0;
        delay(1000);
      }
    }
  }
  else
  {
    cont++;
  }
}
}

```

## Código de envío para mensajes de texto

A continuación se muestra un programa para el envío de mensajes de texto a otro Dispositivo Celular

- Éste código envía un mensaje SMS a un número definido por el programador, con un texto pre-definido.

```
int incomingByte = 0;  
void setup()// Función para la inicialización de variables, librerías , pines//  
  
{ Serial.begin(9600);// Configuración de la velocidad de transmision //  
  
  delay(1000); //retardo//  
  erial.println("AT+CMGF=1\r\n");  
  delay(1000); //retardo//  
  
  Serial.print("AT+CMGS=9611132955\r\n");// envía el numero predefinido//  
  delay(1000); delay(1000); //retardo//  
  
  Serial.println("mensaje a enviar .\r\n"); // envía el mensaje predefinido//  
  delay(1000);  
  Serial.print(0x1A,BYTE);// envía Ctrl-Z//  
  delay(1000);}  
void loop(){}
```

## Código de separación de datos (latitud y longitud) y envió a dispositivo móvil.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11,7,6,5,4);

byte conteo=0,m0=0,aux1=0;

char dato[128], cadena[50], entrada;

void setup()
{
  Serial.begin(9600);
  Serial.println("AT");
  attachInterrupt(0,envio,RISING);
  pinMode(3,INPUT);
  lcd.begin(16, 2);
  lcd.display();
}

void envio()
{ Serial.println("AT$GPSACP");
}

void lcd_1() //imprime lcd 1er dato
{
  delay(100);
  lcd.clear();
  lcd.print("ALTITUD:");
```

```

lcd.setCursor(0,1);

for(aux1=2;aux1<=11;aux1++)
{
    lcd.print(dato[aux1]);
}
}

void lcd_2()    //imprime lcd 2do dato
{
    delay(100);
    lcd.clear();
    lcd.print("LATITUD:");
    lcd.setCursor(0,1);
    for(aux1=13;aux1<=22;aux1++)
    {
        lcd.print(dato[aux1]);
    }
}

void lcd_3()    //imprime lcd 3er dato
{
    delay(100);
    lcd.clear();
    lcd.print("LONGITUD:");
    lcd.setCursor(0,1);
    for(aux1=24;aux1<=34;aux1++)

```

```

{
    lcd.print(dato[aux1]);
}
}

void lcd_c() //imprime lcd total de caracteres de la cadena
{
    delay(100);

    lcd.clear();

    lcd.print(conteo);
}

void lcd_e() //imprime un solo caracter especifico
{
    delay(100);

    lcd.clear();

    lcd.print(dato[13]);    //dato "13" de la cadena "dato"
}

void mensaje()
{
    Serial.println("AT+CMGF=1");

    delay(500);

    Serial.println("AT+CMGS=+529615931691");    //movil del operador

    delay(500);

    Serial.print("DATOS OBTENIDOS: ");

    //Serial.print("ALT:230446.000,LONG:1646.1730N,LAT: 09306.5532W");

    //ciclo for

```

```

Serial.print("ALTITUD: ");
for(aux1=2;aux1<=11;aux1++) //DIRECCION 1ER DATO
{
    Serial.print(dato[aux1]);
}
Serial.print(", LATITUD: ");
for(aux1=13;aux1<=22;aux1++) //DIRECCION 2DO DATO
{
    Serial.print(dato[aux1]);
}
Serial.print(", LONGITUD: ");
for(aux1=24;aux1<=34;aux1++) //DIRECCION 3ER DATO
{
    Serial.print(dato[aux1]);
}
Serial.print(". ING. ESAU FLORES MARROQUIN");
Serial.write(0x1A);
lcd.clear();
lcd.print("DATOS OBTENIDOS");
delay(1000);
lcd_1();
delay(2000);
lcd_2();
delay(2000);
lcd_3();

```

```
delay(2000);  
lcd.clear();  
lcd.print("MSJ ENVIADO");  
delay(1000);  
}
```

```
void loop()  
{  
  if(Serial.available()>0)  
  {  
    entrada=Serial.read();  
    if(entrada==':') //inicio de la cadena  
    {  
      m0=1;  
    }  
    if(entrada=='K') //fin de la cadena  
    {  
      lcd.clear();  
      lcd.print("CADENA GUARDADA");    //impresion de prueba en LCD  
      m0=0;  
      conteo=0;  
    }  
    if(m0==1)    //guardar los datos  
    {  
      dato[conteo]=entrada;
```

```
    conteo++;  
}  
}  
  
if(digitalRead(3)==LOW)  
{  
    mensaje();  
}  
}
```