Instituto Tecnológico de Tuxtla Gutiérrez

Ingeniería Electrónica

"IMPLEMENTACIÓN DE UN ROBOT MANIPULADO A CONTROL REMOTO CON PROTOCOLO DE SEGURIDAD EN CASO DE PÉRDIDA DE SEÑAL DE CONTROL"

❖ **Reporte Final de Residencia**

Alumnos	N. de control	Carrera	Semestre
Aguilar Gómez Ricardo de Jesús	09270433	Ingeniería Electrónica	9
Díaz Velázquez Jorge Ariosto	09270444	Ingeniería Electrónica	9

Asesor interno	M. en C. Raúl Moreno Rincón
Asesor externo	Dr. Madaín Pérez Patricio
Área donde se esta realizando la Residencia	División de estudios de Posgrado del ITTG

Tuxtla Gutiérrez, Chiapas a diciembre del 2013

INDICE

CAPITULO I

1.1 INTRODUCCIÓN

La robótica combina diversas disciplinas como son: la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control y la física.

Los sistemas de información en línea y el acceso remoto a la información serían imposibles sin las telecomunicaciones. Los grandes avances que se han dado en la industria de las telecomunicaciones han impactado de manera significativa a los sistemas de cómputo. Anteriormente las telecomunicaciones significaban únicamente transmisión de voz mediante líneas telefónicas. En la actualidad muchas de las transmisiones de telecomunicaciones son de forma digital utilizando computadoras o móviles para la transmisión de datos de un lugar a otro mediante lo que conocemos como comunicación inalámbrica.

Hoy en día la creación de los dispositivos controlados de manera inalámbrica ya no es una novedad para la mayor parte de las personas a nivel mundial, y estos a lo largo de su desarrollo se han ido innovando cada día con los avances de la ciencia y la tecnología, sin embargo día a día se van resolviendo problemas y al mismo tiempo se van derivando otros.

Un problema común es la pérdida de la señal inalámbrica entre el dispositivo emisor y el receptor, y al tiempo de ocurrir esto se presenta otro inconveniente de saber la localización del dispositivo controlado y de donde habrá quedado después de haber perdido señal con el emisor.

Es acá donde se presenta la aplicación de uso del hardware GPS (Sistema de Posicionamiento Global), el cual es utilizado por numerosos profesionistas y empresas, para muy diversas aplicaciones, en una de estas destacamos la aplicación en la robótica para poder ser rastreado, como lo es por ejemplo el caso de los celulares que cuentan con el sistema GPS implementado.
Por estos motivos hoy en día muchas organizaciones en muchos lugares del mundo reconocen los beneficios de establecer una estación de referencia GPS para el control y localización de diversos dispositivos robóticos

1.2 JUSTIFICACIÓN

Los dispositivos manejados a control remoto en ocasiones cuando pierden la señal entre emisor y receptor ya no son posibles recuperarlos.

El presente proyecto tiene como justificación principal lograr que el usuario que controle el robot al momento de que pierdan comunicación, cuente con un protocolo de seguridad que permita al dispositivo (robot) regresar a un punto donde pueda volver a tener comunicación con el emisor y mediante el hardware GPS poder tener la ubicación en todo momento del robot en tiempo real.

Esto traerá beneficios de manera económica ya que ayudara a reducir gastos por la pérdida de dispositivos a causa de problemas de comunicación entre el dispositivo emisor y receptor.
El proyecto se realizo en el ITTG como base de pruebas para futuros proyectos a realizarse en el área de posgrado para la SECRETARIA DE MARINA (SEMART).

1.3 OBJETIVOS

OBJETIVO GENERAL

Diseñar un robot capaz de regresar a su punto de origen en caso de perdida de señal con el emisor.

OBJETIVOS ESPECÍFICOS

- Lograr una comunicación inalámbrica entre el emisor y receptor.
- Programación de los dispositivos utilizados en el proyecto.
- Implementación y programación del GPS al proyecto
- Pruebas de funcionamiento

1.4 ALCANCES Y LIMITACIONES

Alcances

El dispositivo posee una interfaz fácil de manejar por el usuario a base de un Smartphone, y a base de un LCD nos proporciona las coordenadas en las que se encuentra.

Alcances que proporcione el uso del bluetooth:

- El bluetooth es una tecnología ampliamente usada para la comunicación inalámbrica desde un teléfono celular (Smartphone).

- El uso del bluetooth ofrece la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Alcances con el uso del sistema GPS.

- Obtención de coordenada para la localización la pronta del dispositivo al que se le implemento el sistema GPS.

- Localización en tiempo real del dispositivo.

- Registro de señales de los satélites y su codificación para su procesamiento.

- Se puede adaptar un GPS shield dotado de un lector de memoria microSD para descargar los archivos de datos de las coordenadas almacenadas en el dispositivo, para posteriormente comparar los datos con la ruta original visualizando tiempos y ubicación del dispositivo.

Limitaciones

- El uso de bluetooth solo nos proporciona pocos metros de comunicación inalámbrica

- No contar con recursos suficientes para una mejor opción de comunicación inalámbrica.

- Durante el proceso se pueden presentar obstrucciones en la recepción de señales.

- El GPS cuenta con un margen de error alrededor de 5 a 10 metros a la redonda (excepto los GPS diferenciales)

CAPITULO II

2.1 CARACTERIZACIÓN DEL ÁREA EN QUE SE PARTICIPO

2.1.1 Antecedentes de la Empresa

En la década de los 70s, se incorpora el estado de Chiapas al movimiento educativo nacional extensión educativa, por intervención del Gobierno del Estado de Chiapas ante la federación.

Esta gestión dio origen a la creación del Instituto Tecnológico Regional de Tuxtla Gutiérrez (ITRTG) hoy Instituto Tecnológico de Tuxtla Gutiérrez (ITTG). El día 23 de agosto de 1971 el Gobernador del Estado, Dr. Manuel Velasco Suárez, colocó la primera piedra de lo que muy pronto sería el Centro Educativo de nivel medio superior más importante de la entidad. El día 22 de octubre de 1972, con una infraestructura de 2 edificios con 8 aulas, 2 laboratorios y un edificio para talleres abre sus puertas el Instituto Tecnológico de Tuxtla Gutiérrez con las carreras de Técnico en Máquinas de Combustión Interna, Electricidad, Laboratorista Químico y Máquinas y Herramientas.

En el año 1974 dio inicio la modalidad en el nivel superior, ofreciendo las carreras de Ingeniería Industrial en Producción y Bioquímica en Productos Naturales.

En 1980 se amplió la oferta educativa al incorporarse las carreras de Ingeniería Industrial Eléctrica e Ingeniería Industrial Química.

En 1987 se abre la carrera de Ingeniería en Electrónica y se liquidan en 1989 las carreras del sistema abierto del nivel medio superior y en el nivel superior se reorientó la oferta en la carrera de Ingeniería Industrial Eléctrica y se inicia también Ingeniería Mecánica.

En 1991 surge la licenciatura en Ingeniería en Sistemas Computacionales. Desde 1997 el Instituto Tecnológico de Tuxtla Gutiérrez ofrece la

Especialización en Ingeniería Ambiental como primer programa de postgrado. En 1998 se estableció el programa interinstitucional de postgrado con la Universidad Autónoma de Chiapas para impartir en el Instituto Tecnológico la Maestría en Biotecnología.

En el año 1999 se inició el programa de Maestría en Administración como respuesta a la demanda del sector industrial y de servicios de la región. A partir de 2000 se abrió también la Especialización en Biotecnología Vegetal y un año después dio inicio el programa de Maestría en Ciencias en Ingeniería Bioquímica y la Licenciatura en Informática.

El instituto tecnológico de Tuxtla Gutiérrez está ubicado en la carretera panamericana Km. 1080 C.P 29050 como se puede observar en la *figura 2.1*

Figura 2.1 Ubicación geográfica del ITTG

2.1.2 Organigrama de la Empresa

Un organigrama es la representación gráfica de la estructura de una empresa o cualquier otra organización. Representa las estructuras departamentales y las personas que la dirigen.

En la figura 2.2 podemos observar las áreas que conforman el ITTG señalando en particular la División de Estudios Investigación y Posgrado y dentro de ésta la coordinación de la Maestría en Ciencias en Ing. Mecatrónica que fue donde se llevo acabo el proyecto de la residencia profesional.

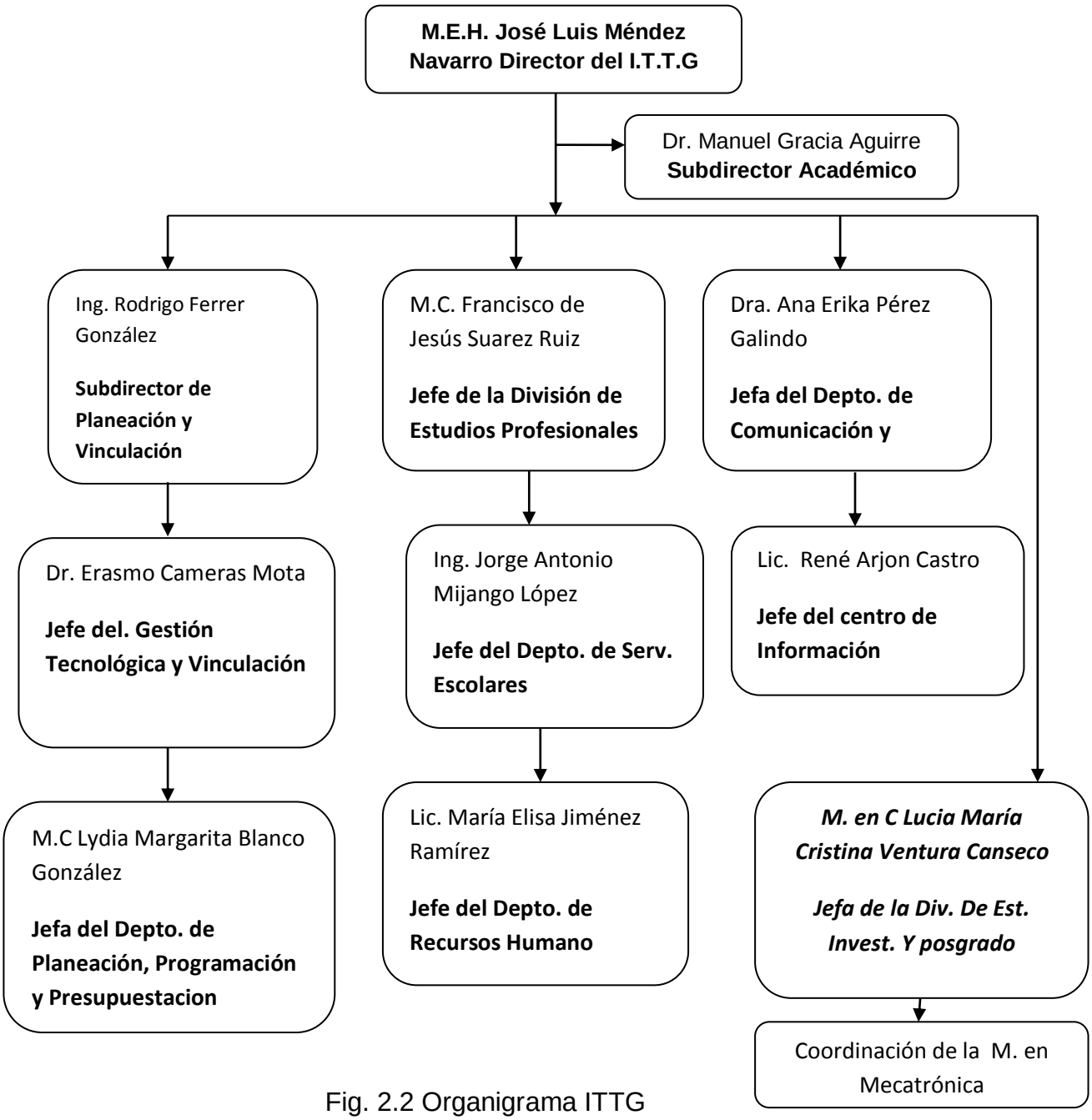


Fig. 2.2 Organigrama ITTG

2.1.3 Misión, Visión Y Valores

Misión

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora respeto al medio ambiente y apego a los valores éticos.

Visión

Ser una institución de excelencia en la educación superior tecnológica del sureste comprometido con el desarrollo socioeconómico sustentable de la región.

Valores

- El ser humano
- El espíritu de servicio
- El liderazgo
- El trabajo en equipo
- La calidad
- El alto desempeño
- Respeto al medio ambiente

2.1.4 Descripción Del Área Donde Se Realizó El Proyecto.

Como se presentó en el organigrama el ITTG cuenta con diversos departamentos y uno de estas áreas es el de Div. De Estudios de Posgrado e Investigación que fue donde se realizó el proyecto de residencia profesional.

El área de la Div. De Estudios de Posgrado e Investigación del ITTG tiene como jefa del departamento M. en C. Lucia María Cristina Ventura Conseco.
En el Instituto Tecnológico de Tuxtla Gutiérrez (ITTG) a través de ésta área, se encarga de la realización de proyectos de vanguardia, como un ejemplo se puede citar la elaboración de un prototipo denominado “Escáner Tridimensional Robótico para objetos de grandes dimensiones” para la SECRETARIA DE MARINA (SEMART).
En esta área se trabaja para satisfacer las demandas de dependencias Nacionales y Estatales con tecnología de punta.

Dentro del instituto los principales objetivos de este departamento son:

- Planear, coordinar, controlar y evaluar los estudios de posgrados que se imparten en el Instituto Tecnológico, así como los proyectos de desarrollo curricular y la atención a los alumnos, de conformidad con las normas y lineamientos emitidos por la Secretaría de Educación Pública.

- Organizar, controlar y evaluar los proyectos de evaluación y desarrollo curricular relacionados con los cursos de posgrados que se impartan en el Instituto Tecnológico, de acuerdo a los procedimientos establecidos.

- Supervisar y evaluar el funcionamiento de la división y con base en los resultados, proponer las medidas que mejoren su operación.

En la coordinación de la Maestría en Ciencias de la Ing. Mecatrónica que se encuentra dentro de la Div. De estudios de Posgrado e Investigación, como se pudo observar en la *figura 2.2,* fue la zona en particular donde se realizó la presente residencia profesional; ésta área en particular se encarga de la elaboración de prototipos para el desarrollo de grandes proyectos con la colaboración de residentes y/o estudiantes de la maestría de Mecatrónica, con el asesoramiento de diversos docentes.

En esta área se trabaja con tecnología de punta y en actualidad por citar algunos podemos mencionar XBee, bluetooth, Arduino, el Sistema de Posicionamiento Global (GPS), fibra óptica entre otros.

Se está en constante investigación para la realización de proyectos para la satisfacción de nuestro entorno, y en la solución de los problemas que se vayan presentando en la elaboración de estos, hasta llegar al objetivo deseado. Se estudian problemas que en la actualidad están presentes y se busca como resolverlos con la aplicación de tecnologías actuales, de acuerdo a algoritmos creados. Es aquí donde de acuerdo al algoritmo que se haya creado se va haciendo por partes, con colaboración de residentes y estudiantes de la maestría que con ayuda de algún asesor realizan prototipos base de

funcionamiento de las partes que mas adelante conformarán un proyecto de mayor escala y de mayor calidad el cual logrará satisfacer alguna necesidad.

2.1.4.1 Antecedentes de la problemática.

En el área antes mencionada (posgrado) se han realizado diversos prototipos de robots manipulados inalámbricamente que realizan diversas funciones, pero surge el problema de cómo hacer si en dado momento éste llegara a perder comunicación como saber dónde esta es aquí donde se piensa en la aplicación del Sistema GPS, para una pronta localización. Se pensó en el diseño de un robot e implementarle el Sistema GPS y mediante la ayuda de Google Search o maps hallar su ubicación y poder recuperarlo nuevamente.

En general ésta área esta enfocada en buscar prontas soluciones de manera viable a los problemas que se presenten en la creación de proyectos, y al mismo tiempo como se mencionó anteriormente solucionar y satisfacer las necesidades Nacionales y Estatales que se presenten.

2.2 PROBLEMAS A RESOLVER

Actividades	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Investigación	X	X	X												
Diseño			X	X	X										
Construcción						X	X	X							
Comunicación Inalámbrica								X	X	X					
Implementación del GPS										X	X	X			
Pruebas preliminares													X	X	
Doc. Final															X

Figura 2.3 Cronograma de actividades

En la *figura 2.3* se presenta el cronograma de actividades a realizarse ahí podemos ver los tiempos en que deben realizarse cada uno de ellos.

INVESTIGACION: Como se observa en el cronograma de actividades se empieza con la investigación, que servirá lograr entender lo que se pretende hacer, saber cómo hacerlo, y con qué medios. Es en esta parte donde el uso del internet y las bibliografías son de suma importancia para el desarrollo de cualquier proyecto que se pretenda realizar; se debe reunir la información necesaria que sirva como sustento para el desarrollo del proyecto.

DISEÑO: Ya que se obtuvo toda la información necesaria para la creación del proyecto se pasa a la etapa del diseño, de cómo estará conformado. El diseño viene siendo prácticamente la idea propia de cómo queremos que sea el prototipo a desarrollarse, la distribución de los componentes. En este punto se presenta la creación de diagramas los cuales se comprueba su funcionamiento mediante simuladores y posterior a su verificación, la creación del rutado para las conexiones físicas permanentes de los componentes en placas fenólica.

CONSTRUCCION: Una vez creados los rutados y comprobado mediante simuladores el funcionamiento correcto del circuito viene la construcción del prototipo robótico, en el cual se realizan todas las conexiones físicas, y se va comprobando cada parte del robot que sus componentes no hayan sido afectados por algún descuido y que todo esté correctamente conectado para no presentar problemas en un futuro.

COMUNICACIÓN INALÁMBRICA: Una vez realizado todo lo anterior se empieza con la parte de la programación, en este caso la comunicación inalámbrica median el uso del bluetooth; es acá donde se tiene que lograr una comunicación con un Smartphone, el cual ordenará las funciones que tiene que realizar el dispositivo robotizado, se debe tener cuidado con la conexión del bluetooth así como su alimentación ya que si no puede llegar a dañarse.

IMPLEMENTACIÓN DEL GPS: Esta será la parte más importante del proyecto ya que con el GPS se lleva acabo gran parte del protocolo de seguridad que se

le implementara al robot. De igual manera será la parte donde se llegara a presentar mayor conflicto ya que se debe programar para poder obtener la localización de forma precisa del robot. Se hará uso de librerías que no vienen incluidas en el Arduino.

Posterior a todo lo antes descrito se harán las pruebas para verificar que el dispositivo funcione de manera correcta, es en ésta parte donde los problemas o detalles que se presenten se tendrán que corregir para obtener funcionamiento esperado, que se puede decir, será en el ámbito de la programación que se realizará en la plataforma de Arduino.

Más adelante se abundará con más detalle en cada aspecto que conforma el cronograma de actividades.

CAPITULO III

3.1 FUNDAMENTO TEÓRICO

3.1.1 Conceptos básicos

Con la finalidad de que el lector tenga mas claridad y le sea mas entendible este trabajo, definiremos algunos conceptos básicos propios del tema que se esta desarrollado.

3.1.1.1 Puente H

Un Puente H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avance y retroceso. Son ampliamente usados en robótica y como convertidores de potencia. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos.

El término "puente H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores (mecánicos o mediante transistores). Cuando los interruptores S1 y S4 (*figura 3.1*) están cerrados (y S2 y S3 abiertos) se aplica una tensión positiva en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 (y cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

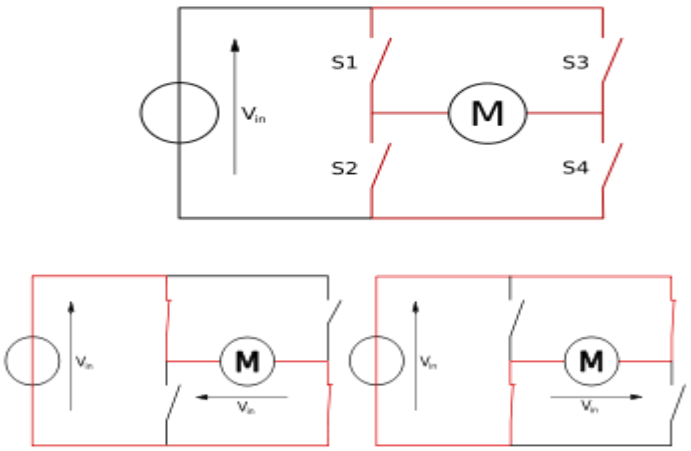


Figura 3.1 Estructura de un puente H y los 2 estados básicos del circuito.

Como se ha mencionado el puente H se usa para invertir el giro de un motor, pero también puede usarse para frenarlo (de manera brusca), al hacer un corto entre los bornes del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia, al desconectar el motor de la fuente que lo alimenta. En la tabla 3.1 se resumen las diferentes acciones.

Tabla 3.1 Giro de motores

S1	**S2**	**S3**	**S4**	**Resultado**
1	0	0	1	El motor gira en *avance*
0	1	1	0	El motor gira en *retroceso*
0	0	0	0	El motor se detiene bajo su inercia
1	0	1	0	El motor frena (*fast-stop*)

3.1.1.2 Doble Puente H (L293b)

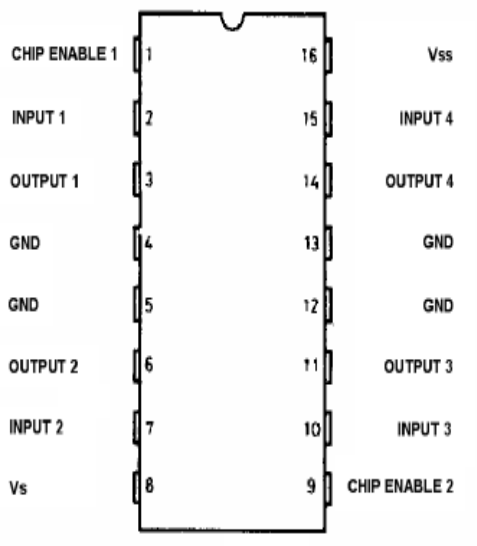
El circuito integrado L293B (*tabla 3.2 y figuras 3.2 y 3.3*) se ha diseñado con el propósito de realizar el control de los motores CC (DC) de manera óptima y económica. Está conformado por cuatro amplificadores push-pull capaces de entregar una corriente de salida de 1A por canal. Cada canal está controlado por entradas compatibles con los niveles TTL y cada par de amplificadores (un puente completo) está equipado con una entrada de habilitación, que puede apagar los cuatro transistores de salida. Tiene una entrada de alimentación independiente para la lógica, de manera que se puede polarizar con bajos voltajes para reducir la disipación de potencia. Los cuatro pines centrales se emplean para conducir el calor generado hacia el circuito impreso.

Sus características sobresalientes son las siguientes:

- Corriente de salida de 1A por canal.
- Corriente pico de salida 2A por canal (no repetitiva).
- Pines de Habilitación.
- Alta inmunidad al ruido.
- Fuentes de alimentación separadas.
- Protección contra exceso de temperatura.

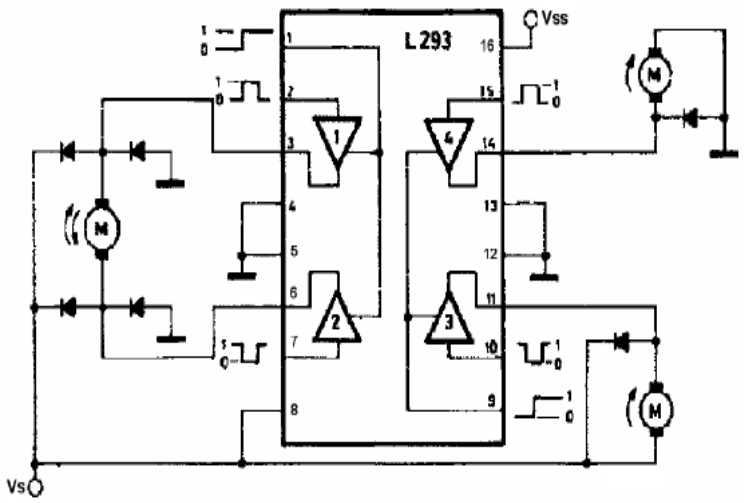
Tabla 3.2 Valores máximos absolutos del driver L293B

Símbolo	Significado	Valor máximo
Vs	Fuente de alimentación (motores)	36 V
Vss	Fuente de alimentación de la lógica	36 V
Vi	Voltaje de entrada	7 V
Vinh	Voltaje de habilitación	7 V
Iout	Corriente pico de salida	2 A
Ptot	Disipación de potencia	5 W



Pin	Función
1	Habilitación del puente 1
2	Entrada del amplificador 1
3	Salida del amplificador 1
6	Salida del amplificador 2
7	Entrada del amplificador 2
8	Fuente de alimentación (motores)
9	Habilitación del puente 2
10	Entrada del amplificador 3
11	Salida del amplificador 3
14	Salida del amplificador 4
15	Entrada del amplificador 4
16	Fuente de alimentación (lógica)

Figura 3.2 Distribución de terminales del driver L293B



Vi (cada canal)	Vo	Vinh
H	H	H
L	L	H
H	Z	L
L	Z	L

Figura 3.3 Diagrama de bloques del driver L293B y tabla de verdad (Z= Alta impedancia de salida). Se muestran diferentes tipos de conexión de motores CC DC).

Observe con cuidado la tabla de verdad de la figura 3.3 y note que si el voltaje de entrada de habilitación Vinh tiene un nivel ALTO el voltaje de salida Vo tendrá el mismo nivel (ALTO o BAJO), aunque no el mismo valor del nivel de entrada Vi. Algo que debe tenerse muy en cuenta es que los valores del voltaje de entrada Vi no son los mismos valores del voltaje de salida Vo, ya que Vi corresponde a valores TTL mientras que Vo es el voltaje de alimentación de los motores Vs.

Por otro lado, si Vinh tiene un valor BAJO, el pin de salida se pone en estado de alta impedancia (sin importar el valor del voltaje de entrada Vi).

3.1.1.3 Servomotores

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

Un servomotor es un motor eléctrico que puede ser controlado tanto en velocidad como en posición.

Los servos se utilizan frecuentemente en sistemas de radio control y en robótica, pero su uso no está limitado a éstos. Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM, que a continuación se explica) para controlar la dirección o posición de los motores de corriente continua.

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés depulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información

a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

$$D = \frac{\tau}{T}$$

D es el ciclo de trabajo

τ Es el tiempo en que la función es positiva (ancho del pulso)

T es el período de la función

3.1.1.4 Aplicación del LCD

El LCD tiene un interfaz paralelo, significando esto que el micro controlador tiene que manipular varios pines del interfaz a la vez para controlarlo. El interfaz consta de los siguientes pines *figura 3.4.*

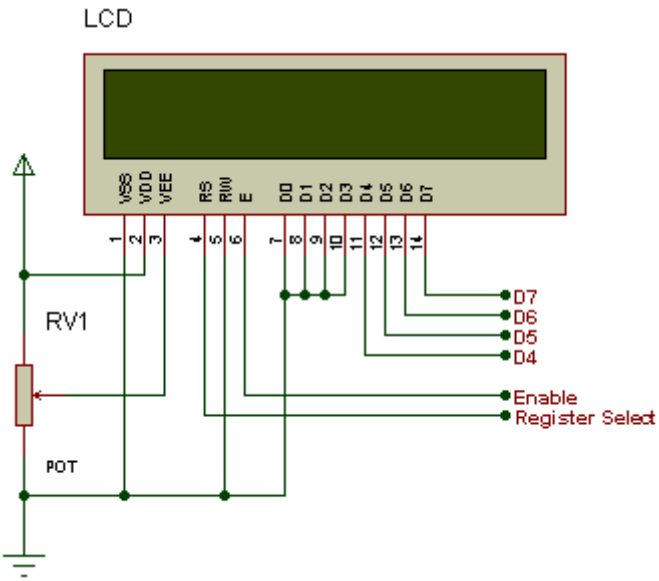


Figura 3.4 Configuración de pines del LCD

Un pin de selección de registro (RS) que controla en qué parte de la memoria del LCD estás escribiendo datos. Se puede seleccionar bien el registro de datos, que mantiene lo que sale en la pantalla, o un registro de instrucción, que es donde el controlador del LCD busca las instrucciones para saber cual es lo siguiente que hay que hacer.

El pin de lectura/escritura (R/W) que selecciona el modo de lectura o el de escritura.

Un pin para habilitar (enable) que habilita los registros.
8 pines de datos (D00-D07). Los estados de estos pines (nivel alto o bajo) son los bits que se están escribiendo a un registro cuando se escribes, o los valores de lectura cuando se esta leyendo.
Hay también un pin de contraste del display (Vo), pines de alimentación (+5V y GND) y pines de retro-iluminación (Bklt+ y Bklt-), que te permiten alimentar el LCD, controlar el contraste del display, o encender y apagar la retro-iluminación, respectivamente.

El proceso de controlar el display involucra la colocación de los datos que componen la imagen de lo que quieres mostrar, en los registros de datos, y luego, colocar las instrucciones, en el registro de instrucciones.

3.1.2 Robótica.

La robótica es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia. Las ciencias y tecnologías de las que deriva podrían ser: el álgebra, los autómatas programables, las máquinas de estados, la mecánica o la informática.

3.1.2.1 Aspecto histórico de la robótica

La historia de la robótica va unida a la construcción de "artefactos", que trataban de materializar el deseo humano de crear seres a su semejanza y que lo descargasen del trabajo. El ingeniero español Leonardo Torres Quevedo (GAP) que construyó el primer mando a distancia para su automóvil mediante telegrafía sin hilo, el ajedrecista automático, el primer transbordador aéreo y otros muchos ingenios, acuñó el término "automática" en relación con la teoría de la automatización de tareas tradicionalmente asociadas.
Karel Capek, un escritor checo, acuñó en 1921 el término "Robot" en su obra dramática Rossum's Universal Robots / R.U.R., a partir de la palabra checa robota, que significa servidumbre o trabajo forzado. El término

robótica es acuñado por Isaac Asimov, definiendo a la ciencia que estudia a los robots. Asimov creó también las Tres Leyes de la Robótica. En la ciencia ficción el hombre ha imaginado a los robots visitando nuevos mundos.

En la elaboración de prototipos robóticos intervienen muchas aplicaciones de la tecnología moderna como haremos mención a continuación.

3.1.3 Comunicación inalámbrica

Nuestra naturaleza humana nos hace desenvolvernos en situaciones donde se requiere comunicación. Para ello, es necesario establecer medios para que esto se pueda realizar. Uno de los medios más discutidos es la capacidad de comunicar computadores o cualquier otro dispositivo a través de redes inalámbricas.

La comunicación inalámbrica o sin cables (figura 3.5) es aquella en la que el emisor y el receptor no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.

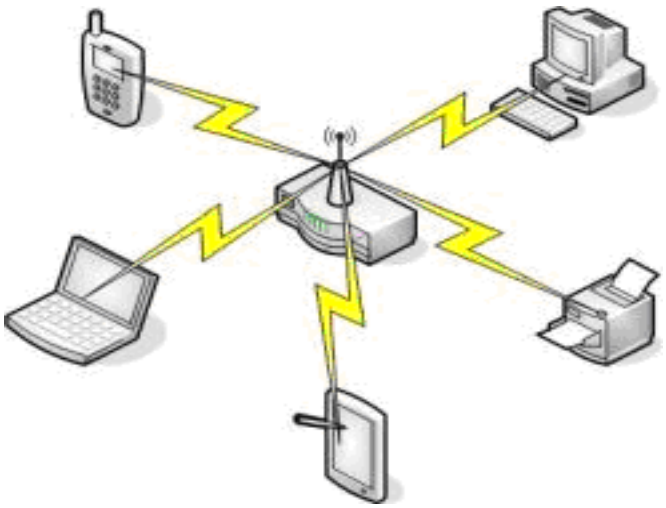


Figura 3.5 Esquema de comunicación inalámbrica

La comunicación inalámbrica, que se realiza a través de ondas de radiofrecuencia, facilita la operación en lugares donde el dispositivo o un PC no se encuentra en una ubicación fija (almacenes, oficinas de varios pisos, etc.). Actualmente se utiliza de una manera general y accesible para todo público. Cabe también mencionar que actualmente las redes cableadas presentan ventaja en cuanto a transmisión de datos sobre las inalámbricas. Mientras que las cableadas proporcionan velocidades de hasta 1 Gbps (Red Gigabit), las inalámbricas alcanzan sólo hasta 108 Mbps

Uno de los dispositivos más comunes para la comunicación inalámbrica es el *bluetooth* del cual se hará referencia a continuación.

3.1.4 Tecnología Bluetooth

El bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar los cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan está tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como PDA, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

3.1.4.1 Aspecto histórico del bluetooth

La historia de Bluetooth se remonta a mediados de la década del 90', cuando Ericsson se encontraba desarrollando una tecnología que permitiera comunicaciones a corto alcance con la bondad de ocupar muy poca energía en los dispositivos (principalmente móviles).

Con el paso del tiempo, los grandes de la tecnología mostraron interés por el producto y formaron una SIG (Special Interest Group), eso en el campo de la tecnología corresponde a una especie de grupo de trabajo conformado por diferentes empresas, quienes aportan capital monetario y humano. Entre esos grandes está: Apple, Ericsson, Intel, Lenovo, Microsoft, Motorola, Nokia, Nordic Semiconductor y Toshiba. (Hoy hay más de 14.000 empresas que lo conforman).

En 1998, cuando Bluetooth vio la luz, Wi-Fi (el estándar 802.11) era muy comentado por el público y muchos llegaron a pensar que Bluetooth era la competencia del Wi-Fi. Bluetooth tiene un norte muy bien definido y se basa en ser una tecnología que:

- Establece conexiones con poco gasto de energía.
- Establece enlaces por lo general de corta duración.
- Otorga seguridad mediante diversas maneras de cifrado de datos, además de exigir el uso de un PIN para establecer conexiones entre equipos.
- Soporta voz y datos
- Tiene un bajo costo de producción e implementación, se planteó que no sobrepasara los US$5 por dispositivo.

Cuando en 1999 Bluetooth lanzó su primera versión, permitía una velocidad de hasta 0.8 ~ 1Mbps a una distancia menor de 10 metros. Obviamente todo lo anterior era teórico y nunca se alcanzaban los 1Mbps y los supuestos 125 KB/s. Prontamente la versión 1.0, cambió a 1.1 y seguidamente a 1.2, la cual fue la más popular.

Con el paso del tiempo Bluetooth ha cambiado notablemente, pero su esencia sigue siendo la misma. Y prácticamente todos los teléfonos en el mercado actual poseen esta tecnología.

3.1.5 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 multidisciplinares.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son elAtmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños *ver anexo 1*. 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.

3.1.5.1 Arduino UNO

El Arduino Uno es una placa electrónica basada en el microprocesador Atmega328 (ficha técnica). Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, un resonador cerámico 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 PC con un cable USB, o alimentarla con un adaptador de corriente AC a DC para empezar.

El Arduino 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.

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.

En la alimentación externa (no USB) de potencia puede venir con un adaptador de AC-DC ó la batería.

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 el circuito 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. La tensión de entrada a la placa Arduino cuando se trata de utilizar una fuente de alimentación externa (en contraposición 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.
- 5V. Este pin como salida 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. patillas 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

Entrada y salida

Cada uno de los 14 pines digitales en el Arduino Uno se puede utilizar como una entrada o salida, utilizando pinMode () ,digitalWrite () , y () digitalRead funciones. 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 tienes funciones especializadas como las siguientes:

- De serie: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y de transmisión (TX) TTL datos en serie. Estos pines están conectados a los pines correspondientes del ATmega8U2 USB-to-TTL de chips de serie.
- 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. Ver el attachInterrupt () función para más detalles.
- PWM:. 3, 5, 6, 9, 10, y 11 proporcionan PWM de 8 bits con el () analogWrite función.
- SPI:. 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) Estos pines soportan comunicación SPI utilizando la biblioteca de SPI .
- LED:. 13 Hay un built-in LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido, cuando el pasador es bajo, es apagado.

El Arduino Uno tiene 6 entradas analógicas, etiquetados A0 a A5, cada uno de los cuales proporcionan 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto se miden desde el suelo a 5 voltios, aunque es posible cambiar el extremo superior de su rango utilizando el pin A_{REF} y la analogReference () función. Además, algunos pernos han especializado funcionalidad:

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

Hay un par de pines la placa:

- AREF. Voltaje de referencia para las entradas analógicas. Se utiliza con analogReference ().
- Restablecer. Con esta línea al pin BAJO para reiniciar el microcontrolador. Normalmente se utiliza para agregar un botón de reinicio para escudos que bloquean el Arduino uno en el tablero.

3.1.6 Sistema Operativo Android

Android (figura 3.6) es un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android, Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005. Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

Figura 3.6 Logotipo de android

Los componentes principales del sistema operativo de Android (cada sección se describe en detalle):

- **Aplicaciones:** Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y

otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

- **Marco de trabajo de aplicaciones:** Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo **permite que los componentes sean** reemplazados por el usuario.

- **Bibliotecas**: Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

3.1.6.1 Aspecto histórico de Android

Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43,6% en el tercer trimestre. A nivel mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (iOS de Apple, Inc.).

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se ha llegado ya al 1.000.000 de aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung. Google Playes la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java. No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.

3.1.7 Sistema de Posicionamiento Global

GPS es la solución para una de las incógnitas más antiguas que se ha planteado el hombre: el preguntarse “¿En que lugar de la Tierra me encuentro?” Durante muchos siglos, este problema fue resuelto empleando al Sol y las estrellas para navegar. Asimismo, en tierra, los topógrafos y los exploradores utilizaban puntos conocidos hacia los cuales hacían referencia para sus mediciones o para encontrar su camino.

3.1.7.1 Funcionamiento del GPS

El sistema GPS está constituido por 24 satélites operativos y 3 de respaldo *figura 3.7* y utiliza la triangulación para determinar en todo del globo la posición con una precisión de más o menos metros.

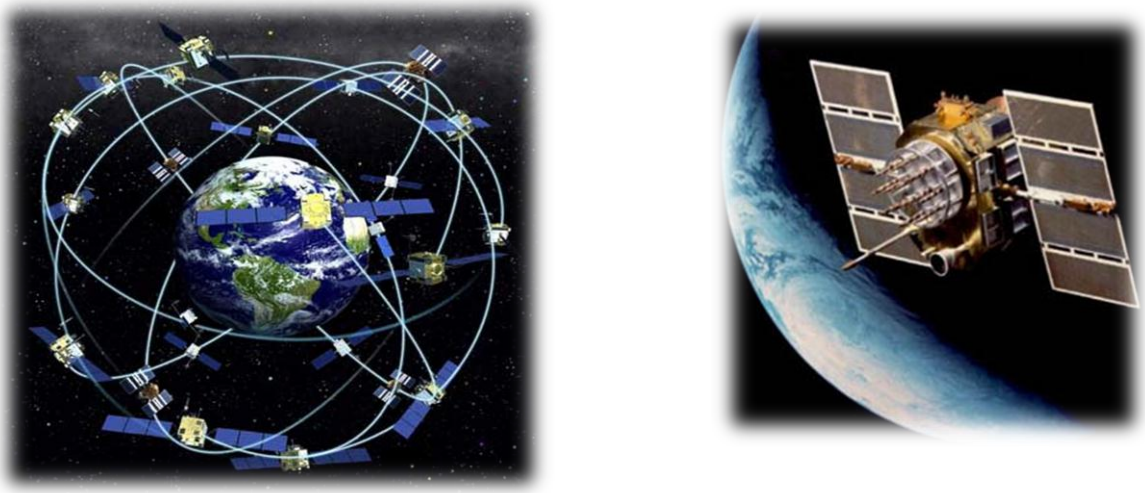


Figura 3.7 Satélites artificiales

Por triangulación calcula la posición en que éste se encuentra. La triangulación en el caso del GPS, se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenada 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.

Existen diferentes métodos para obtener una posición empleando el GPS. El método a utilizar depende de la precisión requerida por el usuario y el tipo de receptor disponible. Estas técnicas pueden ser clasificadas básicamente en tres clases:

- Navegación Autónoma. Empleando un solo receptor simple. Utilizado por excursionistas, barcos en alta mar, y las fuerzas armadas. La Precisión de la Posición es mejor que 100 metros para usuarios civiles y alrededor de 20 metros para usuarios militares.

- Posicionamiento Diferencial Corregido. Conocido comúnmente como DGPS, el cual proporciona precisiones del orden de 0.5-5m. utilizado

para la navegación costera, adquisición de datos para SIG (Sistemas de Información Geográfica GIS), agricultura automatizada.

- Posicionamiento Diferencial de Fase. Ofrece una precisión de 0.5-20mm. Utilizado para diversos trabajos de topografía, control de maquinaria, etc. (Fig. 3.8)

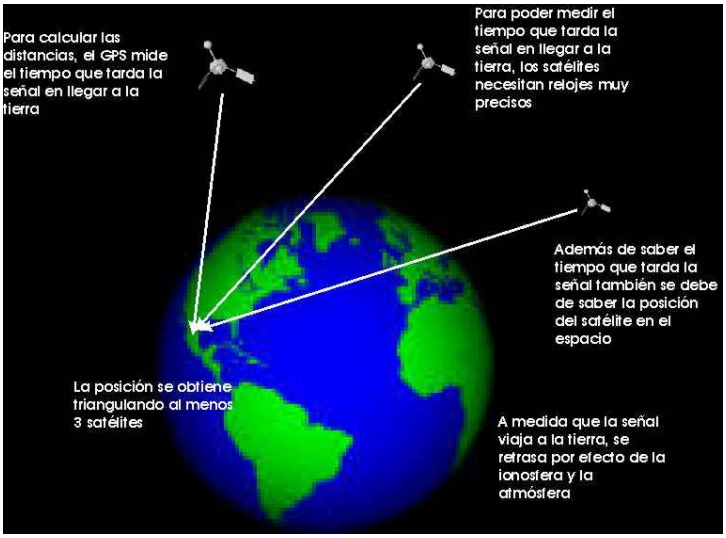


Figura 3.8 posicionamiento diferencial de fase

3.1.7.2 Aspecto histórico del GPS

En 1957, la Unión Soviética lanzó al espacio el satélite Sputnik I, que era monitorizado mediante la observación del efecto Doppler de la señal que transmitía. Debido a este hecho se comenzó a pensar que, de igual modo, la posición de un observador podría ser establecida mediante el estudio de la frecuencia Doppler de una señal transmitida por un satélite cuya órbita estuviera determinada con precisión.

La armada estadounidense rápidamente aplicó esta tecnología, para proveer a los sistemas de navegación de sus flotas de observaciones de posiciones actualizadas y precisas. Así surgió el sistema TRANSIT, que quedó en operación en 1964, y hacia 1967estuvo disponible, además, para uso comercial.

Las actualizaciones de posición, en ese entonces, se encontraban disponibles cada 40 minutos y el observador debía permanecer casi estático para poder obtener información adecuada.

Posteriormente, en esa misma década y gracias al desarrollo de los relojes atómicos, se diseñó una constelación de satélites, portando cada uno de ellos uno de estos relojes y estando todos sincronizados con base en una referencia de tiempo determinado.

En 1973 se combinaron los programas de la Armada y el de la Fuerza Aérea de los Estados Unidos (éste último consistente en una técnica de transmisión codificada que proveía datos precisos usando una señal modulada con un código de PRN (Pseudo-Random Noise: ruido pseudo-aleatorio), en lo que se conoció como Navigation Technology Program (programa de tecnología de navegación), posteriormente renombrado como NAVSTAR GPS.

Entre 1978 y 1985 se desarrollaron y lanzaron once satélites prototipo experimentales NAVSTAR, a los que siguieron otras generaciones de satélites, hasta completar la constelación actual, a la que se declaró con «capacidad operacional inicial» en diciembre de 1993 y con «capacidad operacional total» en abril de 1995.

En 2009, este país ofreció el servicio normalizado de determinación de la posición para apoyar las necesidades de la OACI, y ésta aceptó el ofrecimiento.

3.1.7.3 Descripción del GPS SKM53

El módulo GPS que se uso en este proyecto es el modelo SKM53 del fabricante SKYLAB. El módulo SKM53 de SkyLabs trabaja con el chip MediaTek3329 y viene con una antena integrada.

Es de un tamaño reducido por lo que es fácilmente implementado en la construcción de proyectos.

Se comunica a través de puerto serial UART. Con este módulo podrás medir la latitud, longitud y velocidad de desplazamiento.

Mide tan solo 30 x 20 x 8,5 (Ancho x Alto x Grosor) y mediante una sencilla comunicación serie a 9600 bps al microcontrolador, o lo que significa lo mismo; únicamente dos hilos, Rx y Tx, nos ofrece toda la potencia y las características de módulos destinados al mismo uso, de costos más elevados.
Como podemos ver en la *figura 3.9* a continuación dispone de una tira de 6 pines macho:

Figura 3.9 GPS SKM53

Solo interesan en principio 4 pines, dos destinados a la comunicación serie más los destinados a GND y VCC, resumidamente una descripción de las conexiones sería:

- **RXD:** Línea de entrada de datos de la conexión serie. Conectaremos este pin al controlador o sistema host en su línea de transmisión o TXD.

- **TXD:** Línea de transmisión de datos. Este pin debe ir conectado al de recepción de datos en el microcontrolador. La conexión seria emplea niveles TTL, por lo que no es necesaria su conversión al emplearlo en muchos sistemas microcontrolados.

- **GND:** Masa.

- **VCC:** En este pin conectamos la alimentación, a una tensión de 5 Voltios.

El conector identificado como RST se puede conectar si interesa tener un sistema de reinicio del dispositivo, si no, podemos dejarlo sin conexión.
Una vez hechas las conexiones el procedimiento es leer los datos que lleguen al microcontrolador por la línea RXD.

Características:

-Ultra sensibilidad: -165dBm
- 22 tracking/66 acquisition-channel receiver
- Estándares WAAS/EGNOS/MSAS/GAGAN
- Protocolo NMEA (a 9600bps)
- 01 puerto serial
- Antena incorporada de 18.2 x 18.2 x 4.0 mm
- Rango de temperatura: -40 to 85 C
- Cumple estándar RoHS
- Tamaño reducido 30mm x20mm x 11.4mm

3.1.8 Protocolo NMEA

NMEA 0183 (o NMEA de forma abreviada) es una especificación combinada eléctrica y de datos entre aparatos electrónicos marinos y receptores GPS.

El protocolo NMEA 0183 es un medio a través del cual los instrumentos marítimos y también la mayoría de los receptores GPS pueden comunicarse entre si. Ha sido definido, y está controlado, por la organización estadounidense National MarineElectronics Association.

CAPITULO IV

4.1 PROCEDIMIENTOS Y DESCRIPCION DE LAS ACTIVIDADES REALIZADAS

Para la elaboración y diseño del sistema se han descrito con anterioridad los componentes que han sido previamente seleccionados por su versatilidad y factibilidad de utilización dentro de una amplia variedad de componentes electrónicos.
A continuación se describen a detalle cada uno pasos que se llevaron acabo para el desarrollo del proyecto.

4.1.1 Control de giro de los motores con L293B

Para la operación continua y confiable del robot manipulado a control remoto se debe de contar con una base donde se montara el dispositivo como podemos ver en la *figura 4.1.*

Figura 4.1 base para el dispositivo

En la figura anterior se observa la colocación de los servomotores los cuales se les implemento ruedas en sus extremos.

Para la comprobación del giro de los motores se empleó el uso del puente H, es en esta parte donde entra la aplicación del uso del circuito integrado L293B que al igual que el L293D tiene la misma función con la única diferencia de que el segundo trae internamente diodos de protección en cada uno de los cuatro amplificadores.

El control de motores se realizó mediante la aplicación de la plataforma de Arduino, las conexiones de los pines del CI se puede apreciar en la figura 4.2 y la tabla 4.1

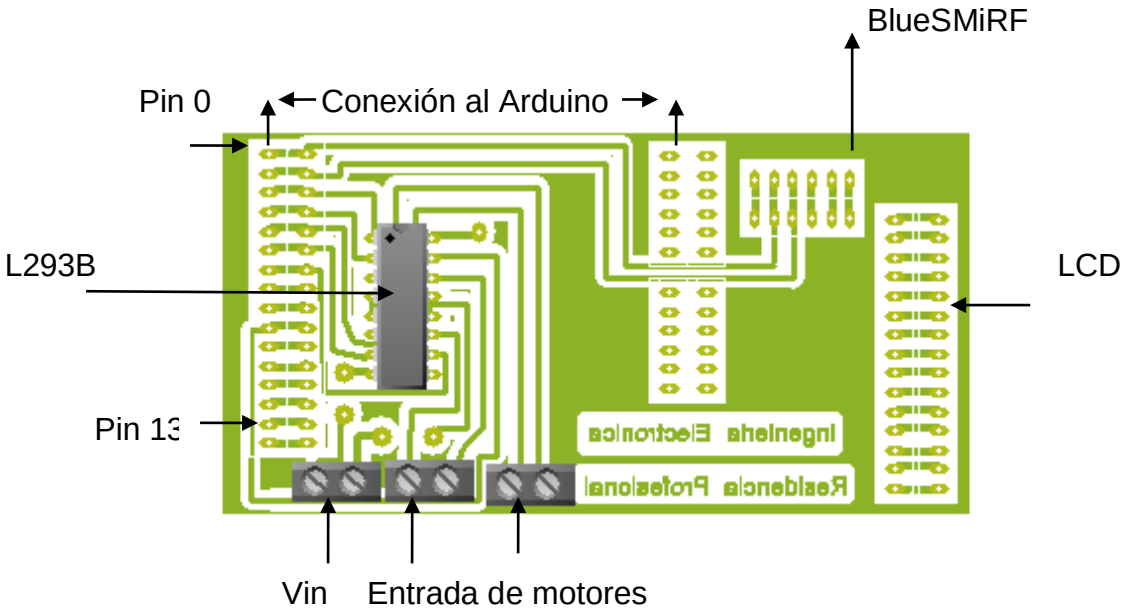


Figura 4. 2 Pre-diseño del circuito

Tabla 4.1 Tabla de conexión de L293B al Arduino

Pines del L293B	PINES DE ARDUINO UNO
1	2
2	3
3	Servomotor
4 y 5	GND
6	Servomotor
7	5
8	Alimentación motores
9	4
10	9
11	Servomotor
12	GND
13	GND
14	Servomotor
15	6
16	Voltaje lógico

Enseguida observamos parte del sketch *figura 4.3* donde se hace la declaración de la conexión de los pines.

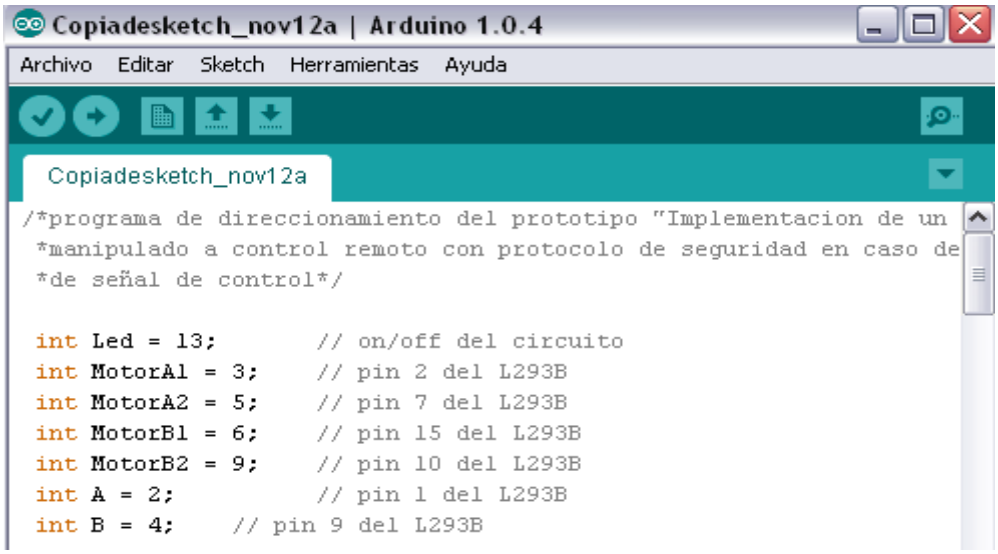


Figura 4.3 Sketch 1 declaración de pines

En el marco teórico se explicó el funcionamiento del puente H y la forma en cómo opera dentro del CI L293B para darle los giros que se deseen a los motores, de la misma forma dentro del sketch de Arduino se van creando bloques para el control de giro de los motores y darnos los sentidos que se requieren que son: Adelante, Atrás, Izquierda, Derecha, Paro (figura 4.4 y 4.5).

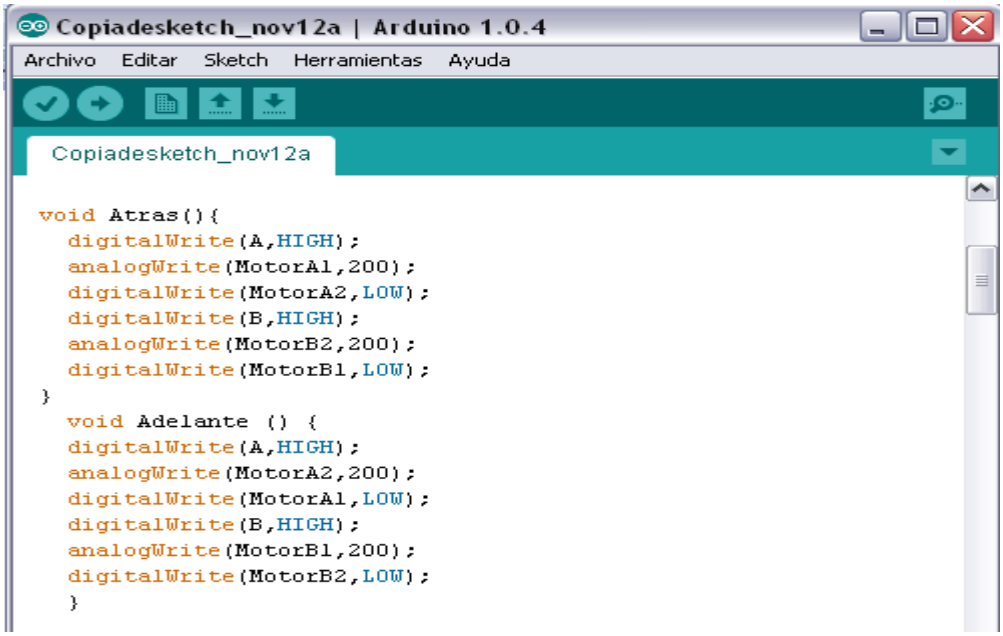


Figura 4.4 Sketch 1 bloques de giro de motores

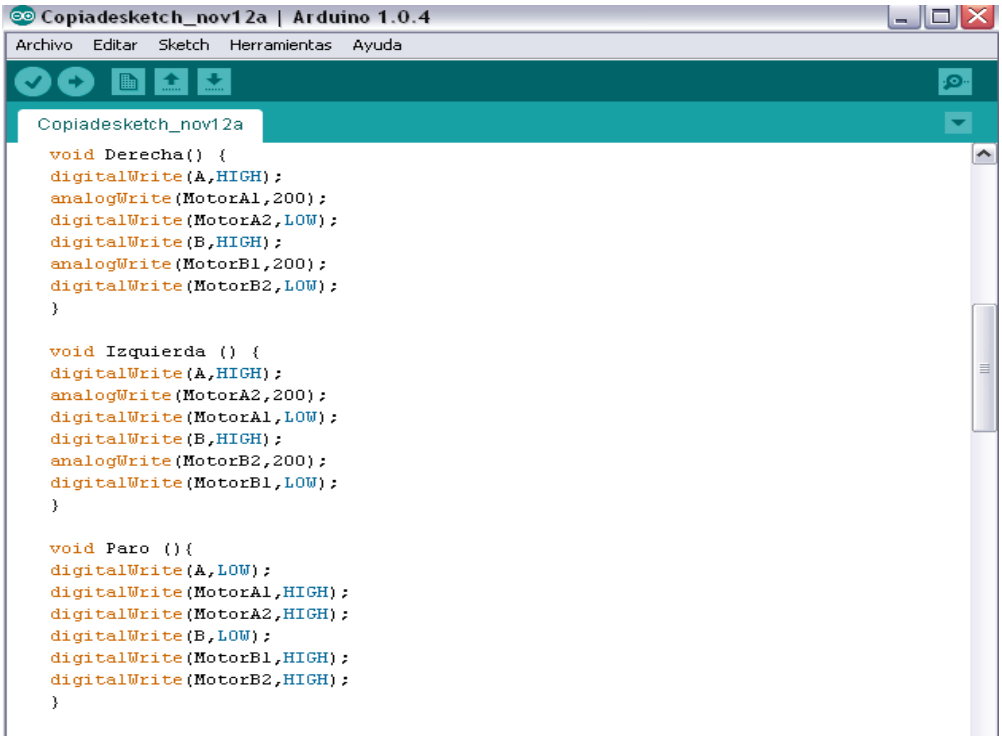


Figura 4.5 Sketch 1 bloques de giro de motores

Posterior a la creación de los bloques de dirección se realizó el complemento de la programación para la verificación del funcionamiento (el programa completo lo podrá observar en el *anexo 1)*

Una vez comprobado el funcionamiento del CI y realizada y probada la programación en Arduino de control de direcciones se construyó la primera placa del diagrama anteriormente presentada quedando como se muestra en la *figura 4.6.*

Figura 4.6 Primera placa de prueba

4.2 Implementación y del LCD para indicar dirección

Durante el desarrollo se utilizaron dos LCD, la primera para indicar la dirección a la cual el dispositivo se dirija, y el segundo en la aplicación del GPS del cual se hablará mas adelante.

Para el control del LCD se dispuso de los pines 7, 10, 11, 12, 14, 15 del Arduino para la su conexión, posterior a esto se realizó la programación la cual nos indicaba la posición a la que se esta moviendo “adelante, de reversa, derecha e izquierda” seguido de la leyenda “ING ELECTRONICA” el display implementado se observa en la *figura 4.7.* Ver anexo 4 para observar el dispositivo terminado.

Figura 4.7 Primer prototipo robótico creado

4.3 Comunicación inalámbrica Bluetooth y Arduino.

Inicialmente se pretendía realizar la comunicación inalámbrica con el uso del bluetooth BLUESMIRF RP-SMA, pero se descartó debido a problemas de compatibilidad con la antena que se le colocó, ya que esta no era la que originalmente venía con el bluetooth.

Por lo cual se empleó el bluetooth HC 06 la cual tiene antena interna. Fig. 4.8

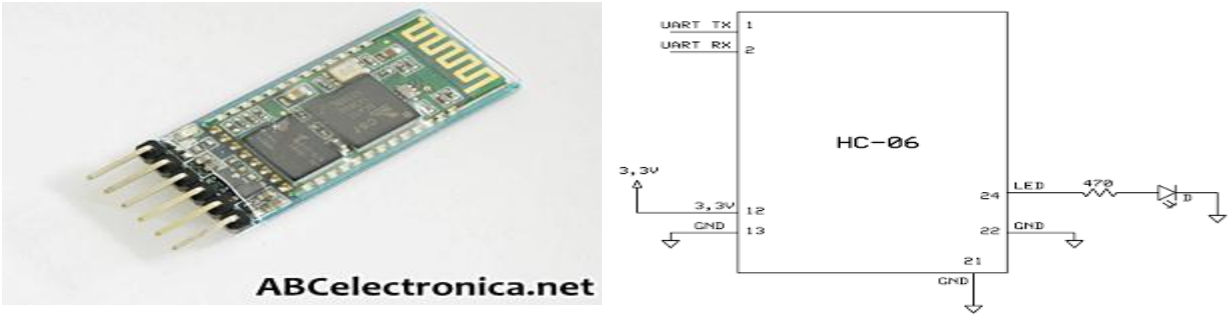


Figura 4.8 Modulo bluetooth HC-06

El módulo bluetooth HC-06 *figura 4.8* viene configurado de fábrica para trabajar como esclavo, es decir, preparado para escuchar peticiones de conexión.

Agregando este módulo a cualquier proyecto se podrá controlar a distancia desde un celular o una laptop todas las funcionalidades que se deseen.

Como se observa el dispositivo HC-06 posee los pines: VCC, GND, TxD, RxD, KEY, de los cuales solo se utilizaron 4 pines (Vcc, GND, TxD, RxD) los cuales fueron conectados como se muestra en la figura 4.9 (Ver anexo 4 para observar el dispositivo terminado). Se debe de tener cuidado en la alimentación de éste dispositivo ya que su voltaje de funcionamiento es de 3.3 a 5V y si éste se excede puede llegar a dañarse. La placa de Arduino proporciona una salida de 3.3V la cual fue utilizada para la alimentación del bluetooth.

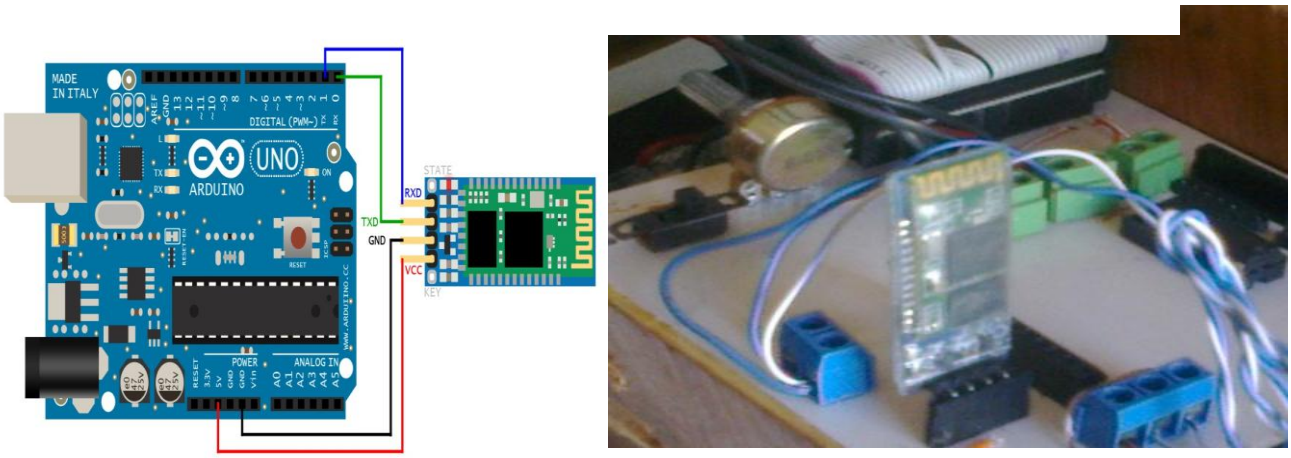


Figura 4.9 Conexión del HC – 06 al Arduino

4.4 Control con android

Se puede utilizar el monitor del puerto serie incorporado en el entorno Arduino para comunicarte con la placa Arduino y darle las instrucciones que debe realizar. Puede ser mediante cable AB o si la PC trae instalado un bluetooth.

La intención en el desarrollo de este proyecto es utiliza también los avances tecnológicos de la actualidad, es en esta parte donde entra el uso del Smartphone. Para las pruebas realizadas se utilizó el Samsung Galaxy, aunque

cabe mencionar que cualquier móvil con el sistema operativo android puede ser utilizado.

4.4.1 Uso de la Aplicación Blueterm de Android

Blueterm figura 4.10 es una aplicación gratuita exclusiva para dispositivos con sistema operativo Android, la cual permite una comunicación inalámbrica con cualquier dispositivo de bluetooth, es un emulador de terminal (terminal emulador vt100) para conectarse a cualquier dispositivo con puerto serie mediante un adaptador bluetooth-serie.

El uso de Blueterm es de gran beneficio ya que es una aplicación fácil de entender por ser similar a la conexión de dos dispositivos mediante bluetooth, en este caso desde el Smartphone localizar el bluetooth HC-O6 que será el esclavo para ser controlado, y que de acuerdo a la programación realizada desde el móvil se enviarán valores por lo general de tipo char o int para realizar las instrucciones que se hayan programado.

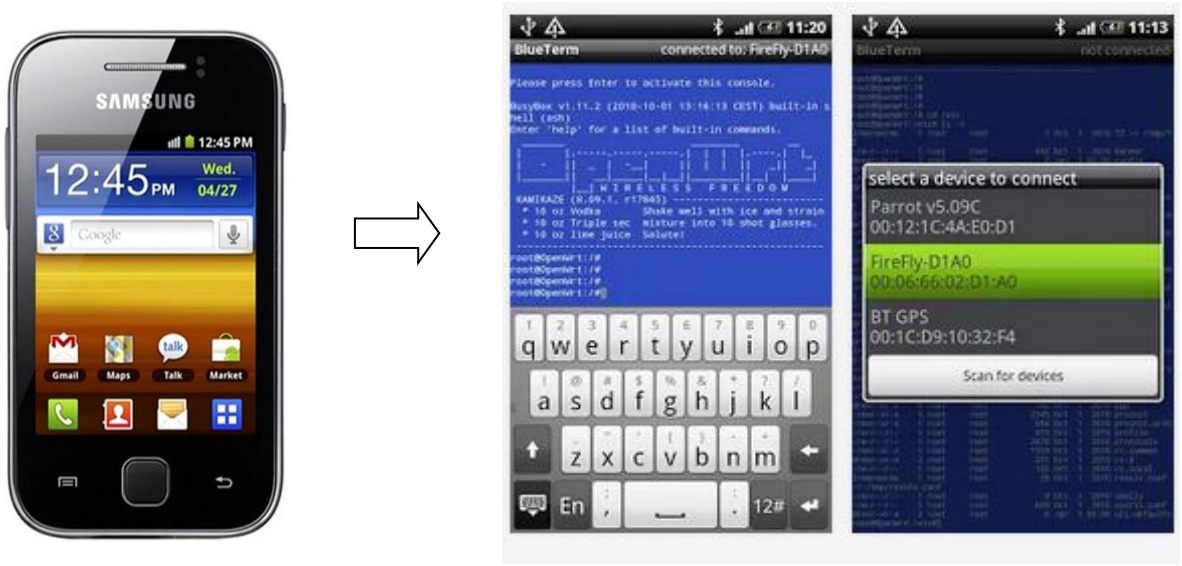


Figura 4.10 Aplicación Blueterm de android

La interfaz con el Smartphone se realiza mediante Available () que devuelve el número de bytes (caracteres) disponibles para ser leídos por el puerto serie. Se refiere a datos ya recibidos y disponibles en el buffer de recepción del puerto (que tiene una capacidad de 128 bytes).

En la programación para el desarrollo del dispositivo fue la siguiente *figura 4.11* (Ver anexo 1 para ver programación completa)

Figura 4.11Sketch 1 de lectura puerto serie

Se observa que mediante el puerto serie (Sintaxis Serial. Available ()) se verifica si hay datos para leer, si es así se guarda en la variable char llamado dato y de acuerdo a lo enviado comprueba lo que se le envió y si está dentro las variables de Switch case designados, realiza un bloque de funciones designadas.

4.5 Protocolo de seguridad

El protocolo de seguridad implementado en el proyecto se llevó a cabo mediante dos etapas importantes, la primera fue en la programación con la comunicación del Smartphone con el dispositivo controlado, en caso de que se perdiera comunicación el robot pueda regresar al último o penúltimo punto donde tuvo señal. La segunda etapa del protocolo de seguridad fue la aplicación y la implementación del hardware GPS (sistema de posicionamiento global) para su pronta localización en tiempo real.

El dispositivo se creo con la siguiente condición:

- Después de haber pasado por algún punto donde el dispositivo recibió señal se considerara como un espacio donde siempre podrá haber comunicación entre el emisor y receptor.

A continuación se describen ambas etapas.

4.5.1 Seguridad de comunicación Smartphone – Robot

El robot fue programado para poder ser usado con o sin el protocolo de seguridad en lo que respecta al regreso en caso de pérdida de señal, esta opción será a disposición del usuario.

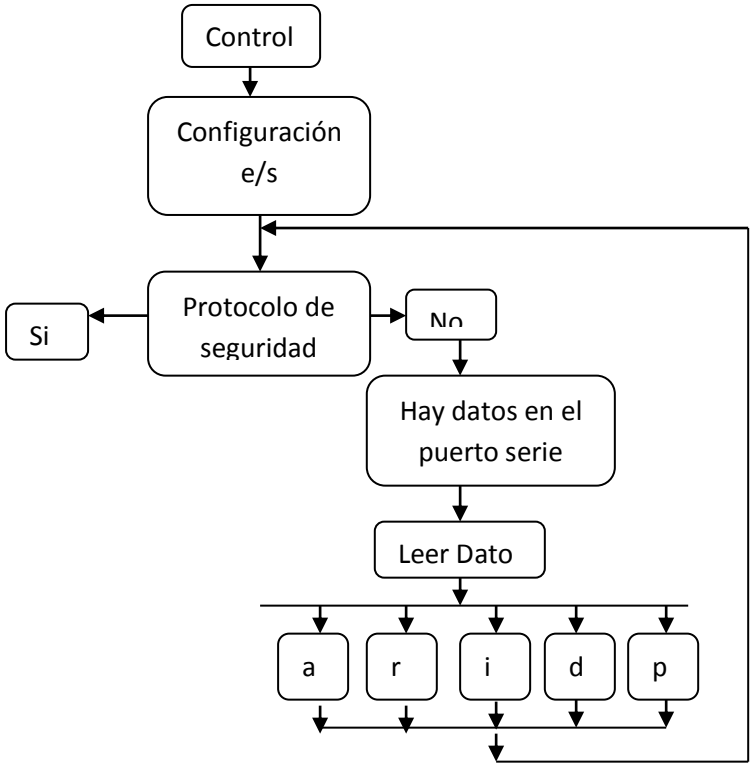
El uso del robot del robot sin el protocolo de seguridad tiene la desventaja de que caso de que se perdiera la señal con el emisor el dispositivo no se parará ya que se quedará con la ultima instrucción recibida excepto que la instrucción haya sido la de PARO pero sin embargo éste no podrá regresar.

Esta parte consistió en lo siguiente:

- Mandar a llamar los bloques void (); izquierda, derecha, adelante, atrás y paro figura 4.12 e ir utilizando libremente las direcciones.

Figura 4.12 Sketch 1Bloques de direcciones

El funcionamiento sin protocolo de seguridad se presenta en el siguiente diagrama.



Las instrucciones son las siguientes:

a= adelante, r= reversa, i=izquierda, d= derecha, p= paro.

Para el control sin el uso del protocolo de seguridad se sugieren las siguientes recomendaciones:

- Procurar utilizar el dispositivo en un lugar seguro.
- No llevarlo a los límites de alcances del bluetooth.
- Regresarlo constantemente hacia el punto de control

Utilizando la opción del protocolo de seguridad se debe de tener mucho cuidado con los valores que se le envíen al robot para que éste tome decisiones.

Dichas instrucciones son las siguientes (a pesar de la condición señalada el protocolo de seguridad se implemento en todos los casos solo para mayor seguridad del usuario):

Para regresar al último punto de recepción de señal:

DATO	INSTRUCCIONES
A	Adelante con protocolo de seguridad
R	Reversa con protocolo de seguridad
I	Izquierda con protocolo de seguridad
D	Derecha con protocolo de seguridad

Para regresar al penúltimo punto de recepción de señal (para uso exclusivo de giros derecha – izquierda):

DATO	INSTRUCCIONES
W	Adelante con protocolo de seguridad
X	Reversa con protocolo de seguridad
Y	Reversa con protocolo de seguridad
Z	Adelante con protocolo de seguridad

Los pasos del protocolo de seguridad fueron los siguientes:

- Activar protocolo de seguridad
- Leer instrucción mandada
- Comprobar si se encuentra dentro de los comando de seguridad
- Realizar la acción enviada durante el tiempo programado
- Transcurrido el tiempo detenerse para esperar la próxima instrucción
- Si se recibió algún dato ejecutarlo de no ser a si realizar la última o las últimas dos acciones antes del último punto de llegada.

A continuación se describirán los pasos de la programación realizada para el regreso al último punto de comunicación.

En la programación (ver anexo 1) para el uso del protocolo de seguridad, se empleo el uso de los tiempos delay como se observa en la figura 4.13

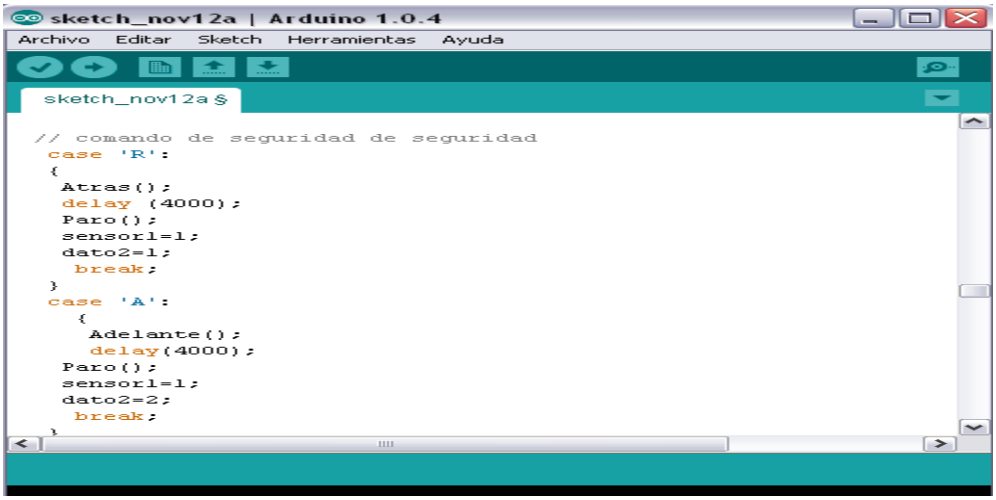


Figura 4.13 Parte del sketch 1 de la programación

En la figura anterior se observa que inicialmente se manda a llamar la opción deseada del movimiento que se requiere que haga el robot y ésta se ejecuta durante el tiempo que se le asigne, al término del tiempo esta acción se detiene con uso de void PARO ();

La variable sensor1 y la variable dato2, inicialmente tienen el valor de cero, y al darles una instrucción de movimiento toman valores para ejecutar acciones que a continuación se detallan.

Después de que se haya ejecutado alguna instrucción de movimiento con protocolo de seguridad la variable sensor1 siempre tomara el valor de 1 para poder entrar a los bloques de seguridad como se observa en la figura 4.14

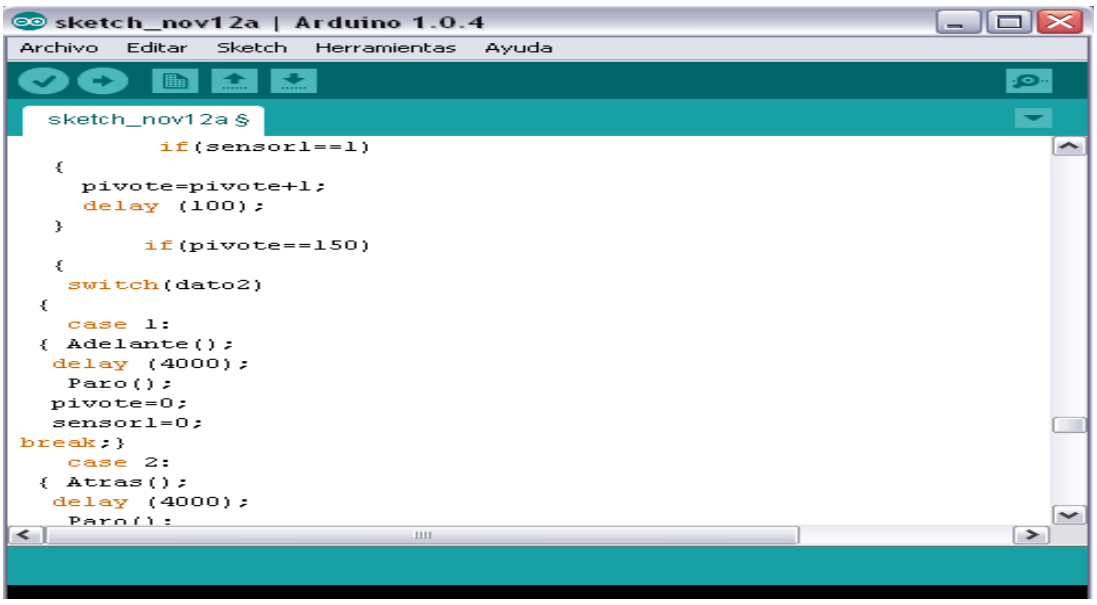


Figura 4.14 Parte del sketch 1 de la programación

En la figura anterior se presenta un retardo que se va guardando en la variable pivote, éste retardo es para que el dispositivo controlado verifique si durante el tiempo transcurrido se recibe algún dato, de no ser a si el robot considerara que se perdió la conexión con el emisor, y mande a llamar bloques de instrucciones de acuerdo al valor que dato2 haya tomado para regresar al robot al ultimo punto donde se recibió señal del emisor para volver a ser controlado.

Regreso al último punto de comunicación emisor - receptor:

Dato → Delay → dato delay → dato delay ↓ dato

Delay → delay

Regresa al ultimo punto de recepción de señal ← no re recibió dato

Esta secuencia de pasos muestra de manera simple como funciona parte de la programación sobre el regreso al ultimo punto de comunicación entre el emisor y receptor, y es de la siguiente manera: recibimos datos, realiza la función indicada, espera a que se reciba el siguiente dato, se vuelve a realizar la función indicada, a si sucesivamente hasta llegar el momento en que haya terminado el tiempo permitido para la recepción de datos, al ocurrir esto se considerara como perdida de señal y el dispositivo ara la ultima instrucción de manera inversa llegando al punto donde recibió el ultimo dato.

Como se menciono los puntos por donde pase el robot se consideran como áreas donde no habrá problemas de comunicación entre el emisor y receptor, tomando en cuenta esto, cuando el robot pierda la señal en algún giro de derecha o izquierda, el dispositivo esta programado para que realice un giro en sentido contrario y un retroceso.

Para los comandos de regreso al penúltimo punto de señal con el emisor se debe tomar en cuenta la siguiente condición:

- Solo se podrá utilizar esta parte del protocolo de seguridad si el dispositivo antes de darle la instrucción adelante o atrás viene de un giro de derecha o izquierda.

Básicamente es el mismo principio de funcionamiento explicado anteriormente.

Las instrucciones de los datos son las siguientes:

DATO	INSTRUCCIONES
W	Atrás – delay – paro - no se recibió dato – adelante –izquierda – atrás – paro
X	Adelante – delay – paro - no se recibió dato – atrás-giro izquierda – atrás – paro
Y	Atrás – delay – paro - no se recibió dato - adelante – derecha – atrás – paro
Z	Adelante – delay – paro - no se recibió dato - atrás – derecha – atrás – paro

Las primeras tres instrucciones se puede observar en la *figura 4.15* donde se aprecia que se manda a llamar el movimiento ordenado el cual se realiza por un determinado tiempo para después pararse y esperar la siguiente instrucción. Las variables sensor1 y dato2 toman valores para ejecutar acciones en caso de ya no recibir algún dato.

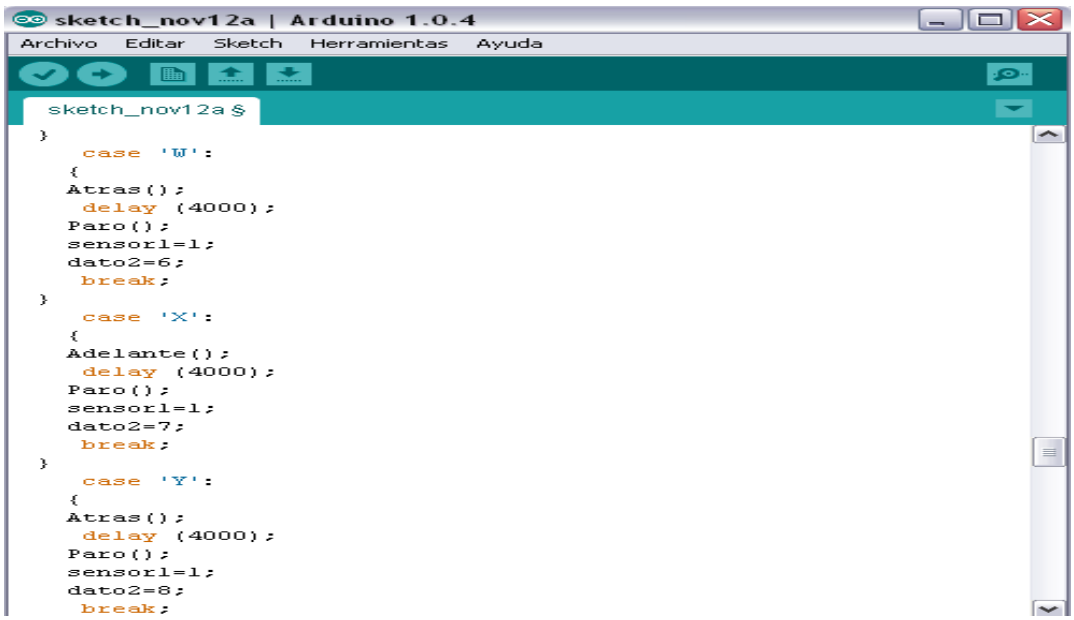


Figura 4.15 Parte del sketch 1 de la programación

En la figura 4.16 se aprecia que se ejecuta la acción del retardo para verificar si se siguen enviando datos, en caso de no recibir ninguna instrucción entra al bloque de seguridad para hacer el regreso.

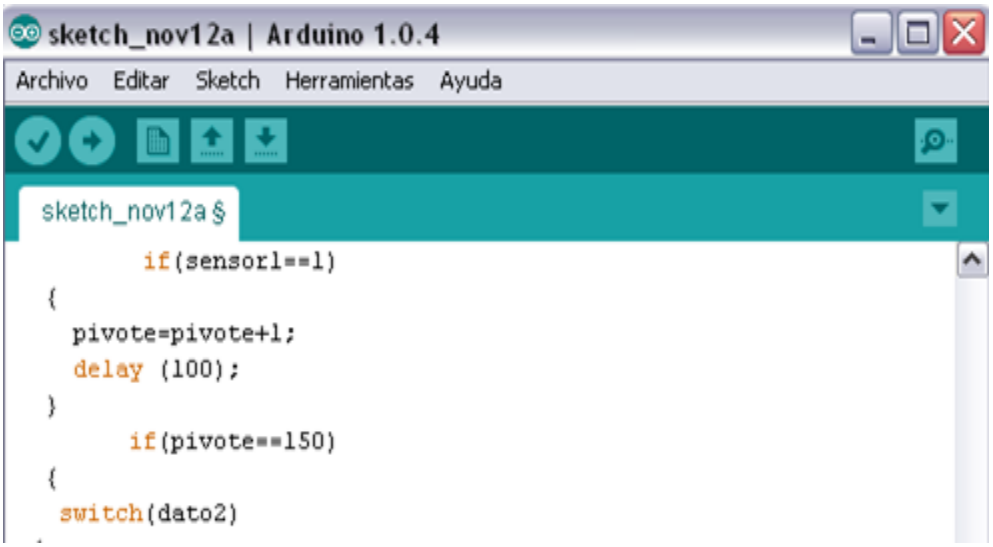


Figura 4.16 Parte del sketch 1 de la programación

De acuerdo ala instrucción de movimiento que se ordeno la variable dato2 tomo un valor, y de acuerdo a ese valor entra a un bloque de la sentencia case para realizar instrucciones de regreso como se observa en la *figura 4.17.* La programación completa se podrá observar en el *anexo 1*

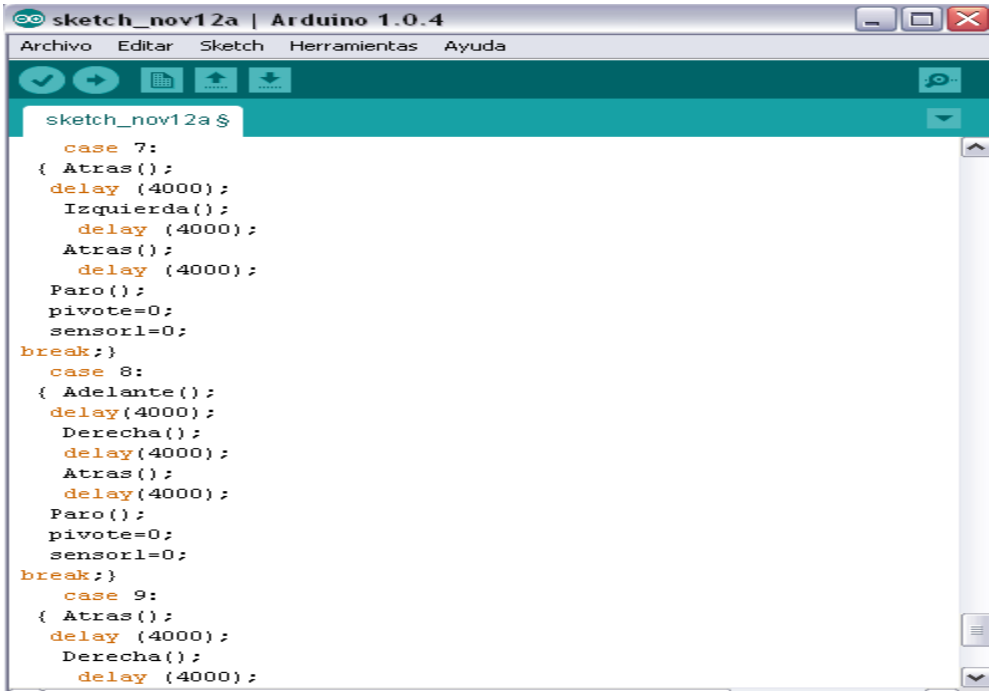
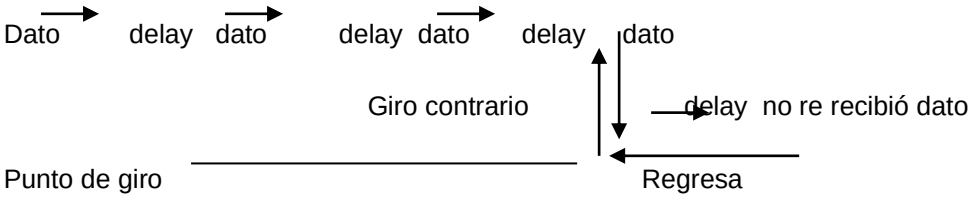


Figura 4.17 Parte del sketch 1 de la programación

Regreso al penúltimo punto de comunicación emisor - receptor:



Esta secuencia de pasos nos muestra como se lleva a cabo el regreso al penúltimo punto de comunicación entre el emisor y receptor, que viene siendo similar a lo que anteriormente se explico. Se recibe el dato, realiza la función, espera el siguiente dato, hasta llegar el momento en que no reciba el dato y ejecute las instrucciones de acuerdo ala instrucción del protocolo de seguridad seleccionado.

Al dispositivo se le implemento luces direccionales, secuenciales y rítmicas (ver anexo 4) para darle una mejor presentación esto fue con ayuda del Pic 16f877a con una programación en Mplab (ver anexo 6).

4.5.2 Implementación del GPS

Al inicio del proyecto se pretendía el uso del GPS SmartGPS Ublox LEA – 5s-0-004 (del cual no se hablara con profundidad) como se observa en la *figura 4.18*, de la compañía Microelectrónica pero debido a inconvenientes similares al del bluetooth BLUESMIRF RP-SMA por parte de la antena, ya que viene de manera externa se tuvieron problemas en la recepción de las coordenadas. Por lo que se optó por el uso del GPS SKM53 el cual se explicó de manera detallada en el marco teórico.

Figura 4.18 GPS de Smart GPS Ublox LEA – 5s-0-004

4.5.2.1 Conexión física del GPS al Arduino

El módulo GPS SKM53 figura 4.19, es un modelo con grandes beneficios, ya utiliza el protocolo NMEA (National Marine Electronics Association) para transmitir los datos de posición. (*Ver anexo 2*)

Figura 4.19 GPS SKM53

En la *figura 4.20* presentada se puede observar la conexión física del hardware GPS al Arduino (Ver anexo 4 para observar el dispositivo terminado). Los pines conectados del GPS al Arduino fue el de recepción (Rx) y transmisión (Tx) además del pin de alimentación y de referencia.

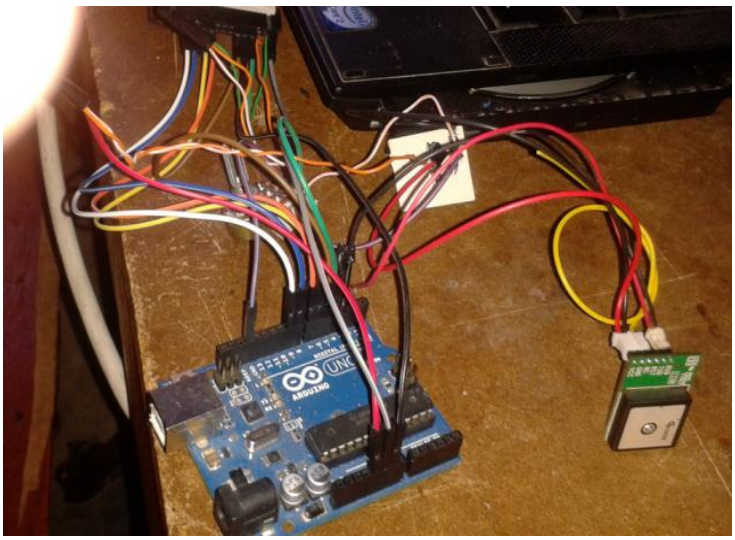


Figura 4.20 Conexiones del SKM53 al Arduino

Para la programación del GPS conectado al Arduino fue necesario el uso de la librería TinyGPS que se tuvo que descargar e incluirla ya que no viene incluida en las librerías predeterminadas de Arduino

4.5.2.12 Uso de la librería TinyGPS

Para el uso del GPS con el Arduino se incluyo la librería TinyGPS; que está diseñado para proporcionar la mayor parte de la funcionalidad GPS NMEA, esta librería permite extraer con mayor facilidad la información de las tramas NMEA.

El dispositivo GPS debe estar conectado a un puerto UART o creado con SoftwareSerial (librería incluida en Arduino), ya que la información se obtiene atravez de estos, la librería simplemente facilita la interpretación de la información recogida.

En la *figura 4.21* se observa el sketch donde indica que TinyGPS ya está incluida dentro de la librería de Arduino, ya que un error que se puede presentar es escribirla considerando que esta ya viene predeterminada como lo es por ejemplo la librería LiquidCrystal.h.

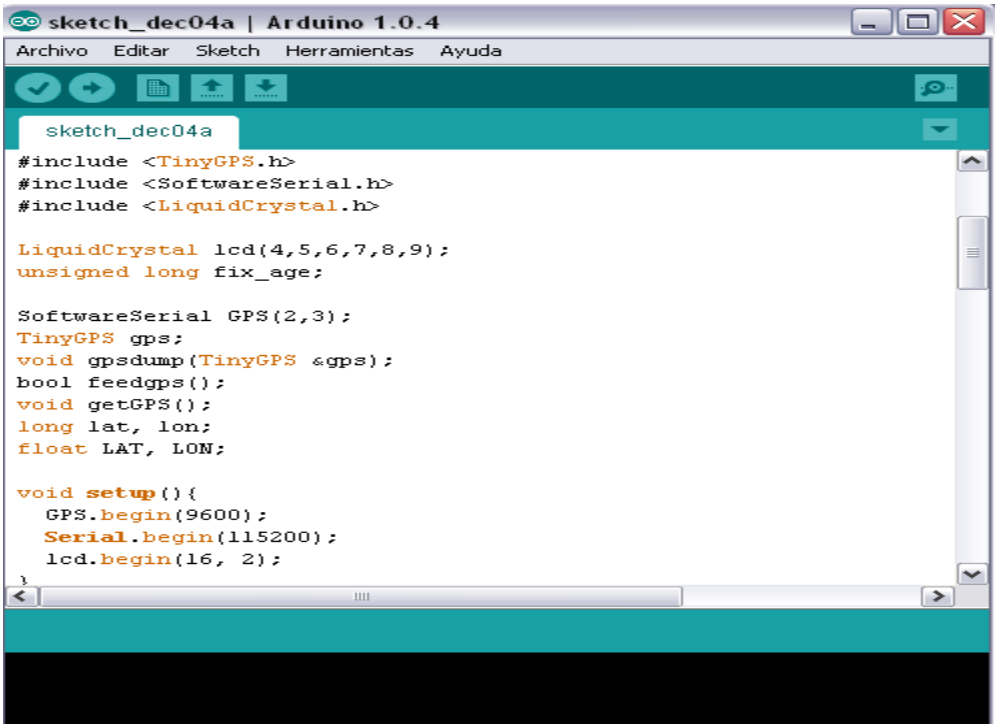


Figura 4.21 Sketch del GPS con librería TinyGPS

Se observa el uso de la librería SofwareSerial.h del que es de suma importancia por el motivo de saber que versión de Arduino se esta utilizando, ya que las versiones anteriores ala 1.0.0 no cuentan con esta librería, y tienen que hacer uso de la librería newsoftserial que tiene que ser descargada e incluirla.

Se observa en la imagen que se configuran los pines 2 y 3 como transmisor (Rx) y receptor (Tx) para la conexión con el GPS y que establece la velocidad de datos en bits por segundo (baudios) para la transmisión de datos en serie para el GPS y el ordenador. Para comunicarse con el computador u otro dispositivo, puede utilizarse las velocidades 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200. Sin embargo, puedes especificar otras velocidades. Para ver la *programación completa ver anexo 3.*

Se hizo uso de las palabras reservadas que son las “ordenes” que el compilador es capaz de ejecutar. Toda palabra reservada debe escribirse con minúscula, y como su nombre lo indica esta reservada por el compilador, y no puede utilizarse como nombres de variables. Se hizo uso de los modificadores de tipo que son: signed, unsigned, short, long. Estos modificadores se utilizan para alterar el significado del tipo base de modo que se ajuste mejor alas necesidades de diferentes situaciones, como lo es en este caso, la recepción de la latitud y longitud recibida mediante el GPS. Se le implemento al GPS un LCD para poder observar las coordenadas en las que el robot se encuentre, tanto en si mismo como en el ordenador, como se observa en la *figura 4.22*

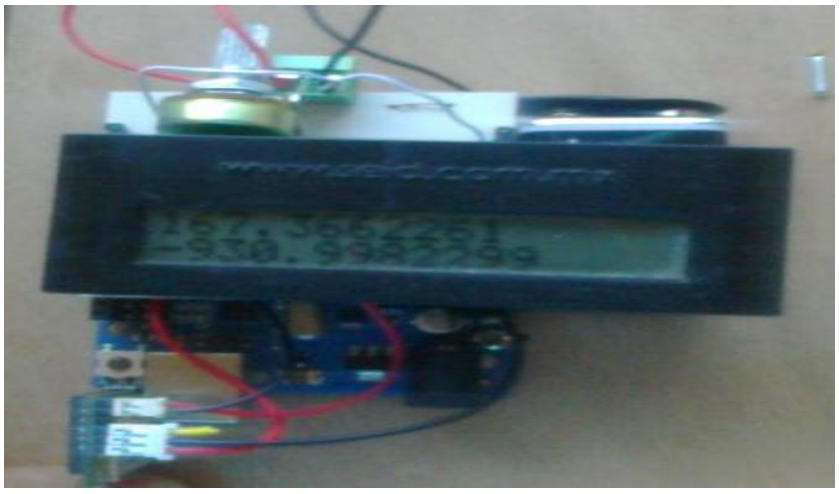


Figura 4.22 Coordenadas vistas en el LCD

CAPITULO IV

5.1 RESULTADOS

Los resultados obtenidos en el control mediante la comunicación inalámbrica fueron los que se esperaban con la desventaja de que el control fue a muy poca distancia debido al uso de bluetooth, las instrucciones se ejecutaron ala perfección de acuerdo ala programación creada (ver anexo 1) y la cual ya se ah explicado.

Los bloques de regreso de protocolo de seguridad se ejecutaron de manera correcta, presentándose ciertos detalles en el momento de los giros izquierda y derecha debido a ciertos detalles, por mencionar alguno el tipo de terreno, desgaste de pilas, y desgaste de los motores debido a muchas pruebas realizadas.

Cabe mencionar que hay que tener cierta atención en la alimentación de los motores, ya que si son alimentados por el pin de 5 V de la placa de Arduino puede ser poco a si que lo recomendable es conectar el cable de alimentación de los motores al terminal positivo de una fuente externa. Al hacer esto no olvidarse de unir las dos tierras existentes en el circuito a de la fuente externa y la de la placa de Arduino (si existen mas fuentes externas unir todas las tierras), ya que si no compartimos la dos tierras para que sean una sola el comportamiento del circuito puede ser errático.

Resultados obtenidos del GPS

A continuación tenemos los resultados obtenidos por el GPS en condiciones de terreno abierto con el área alrededor despejada donde no se ve muy afectado el GPS y los resultados obtenidos en un área rodeada de muros, en un cuarto encerrado en esta parte se conto con la ayuda del google search y google maps para la verificación de los resultados obtenidos por el GPS.

La programación para la recepción de datos del GPS esta programado para que cada 5 segundos se este actualizando, y a si la ubicación la tengamos siempre presente, actualizándose tanto en el display implementado como en el ordenador.

Como se puede observar en la figura 5.1 se nota que los datos están en cero ya que el arranque en frio del GPS dura aproximadamente de 36 segundos a 2 minutos en captar sus coordenadas.

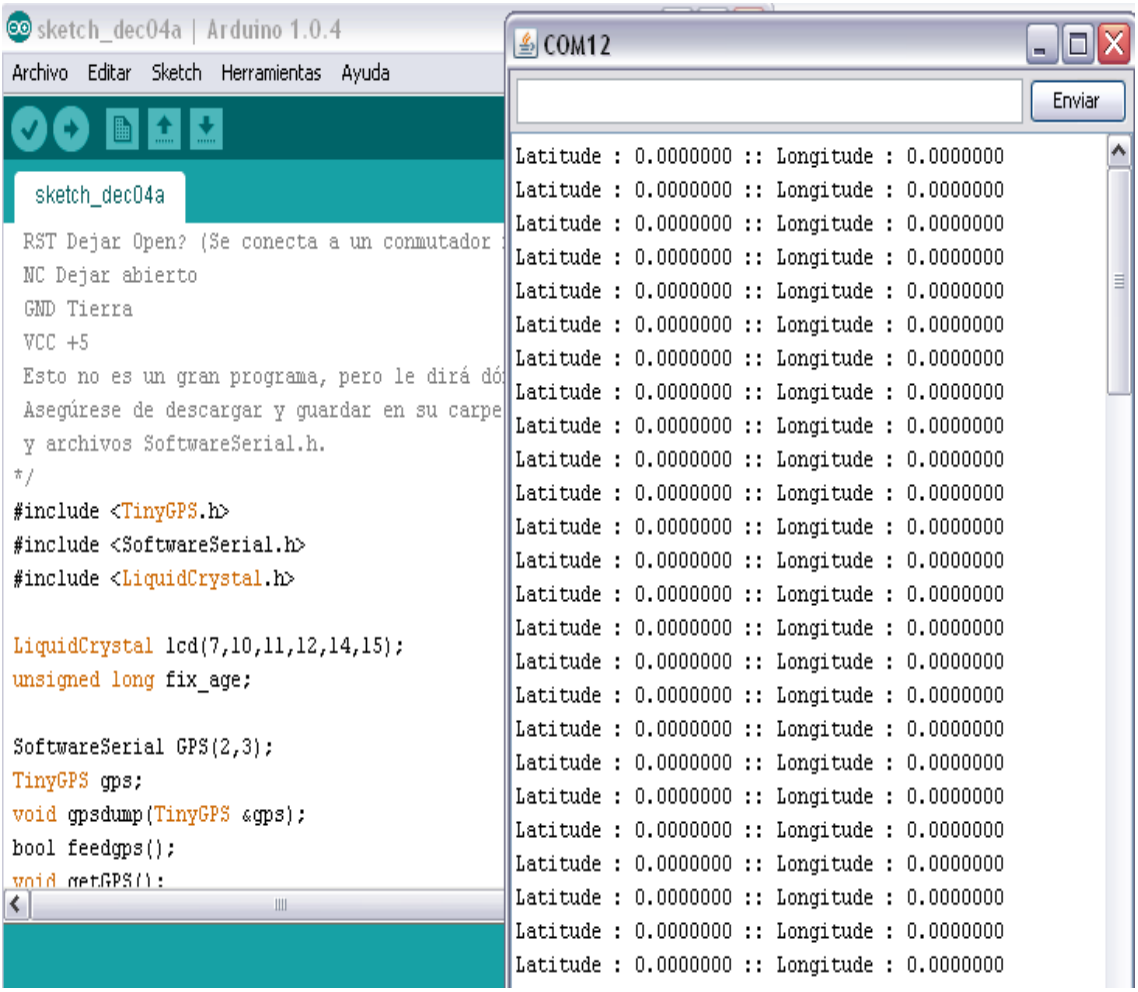


Figura 5.1 Primer envió de datos del GPS

Como se aprecia en la *figura 5.2* después de haber transcurrido cierto tiempo empieza a enviarse las coordenadas las cuales se actualizan de acuerdo al tiempo programado.

Los resultados vistos en la imagen son en un área libre, despejada por ejemplo el techo de un local.

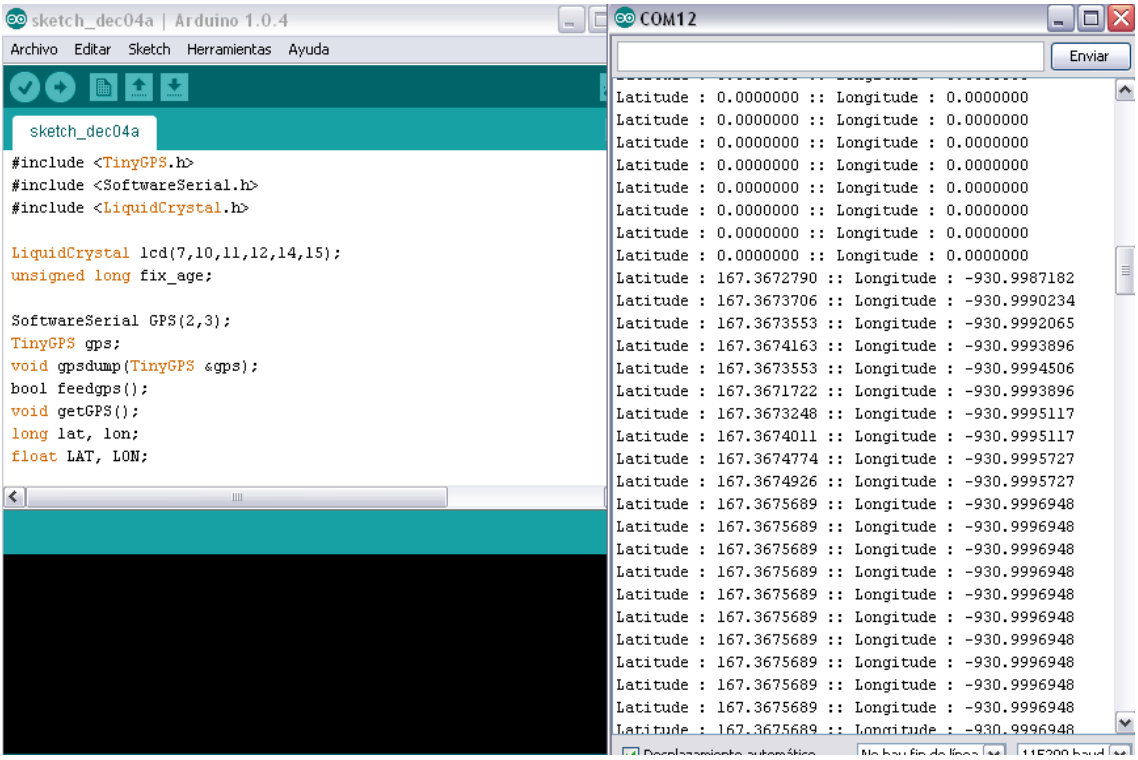


Figura 5.2 Captación de coordenadas en área despejada y comprobación mediante google search y google maps

En base a las pruebas de localización realizada en un área despejada en este caso el techo de una casa, se comprueba que nos da una ubicación exacta del

dispositivo considerando el margen de error que esta permitido por el GPS SKM53 que son de 10 metros.

Enseguida se presentan los resultados obtenidos en un espacio cerrado en este caso un cuarto completamente aislado *figura 5.3* dentro de la misma zona donde se hizo la prueba anterior.

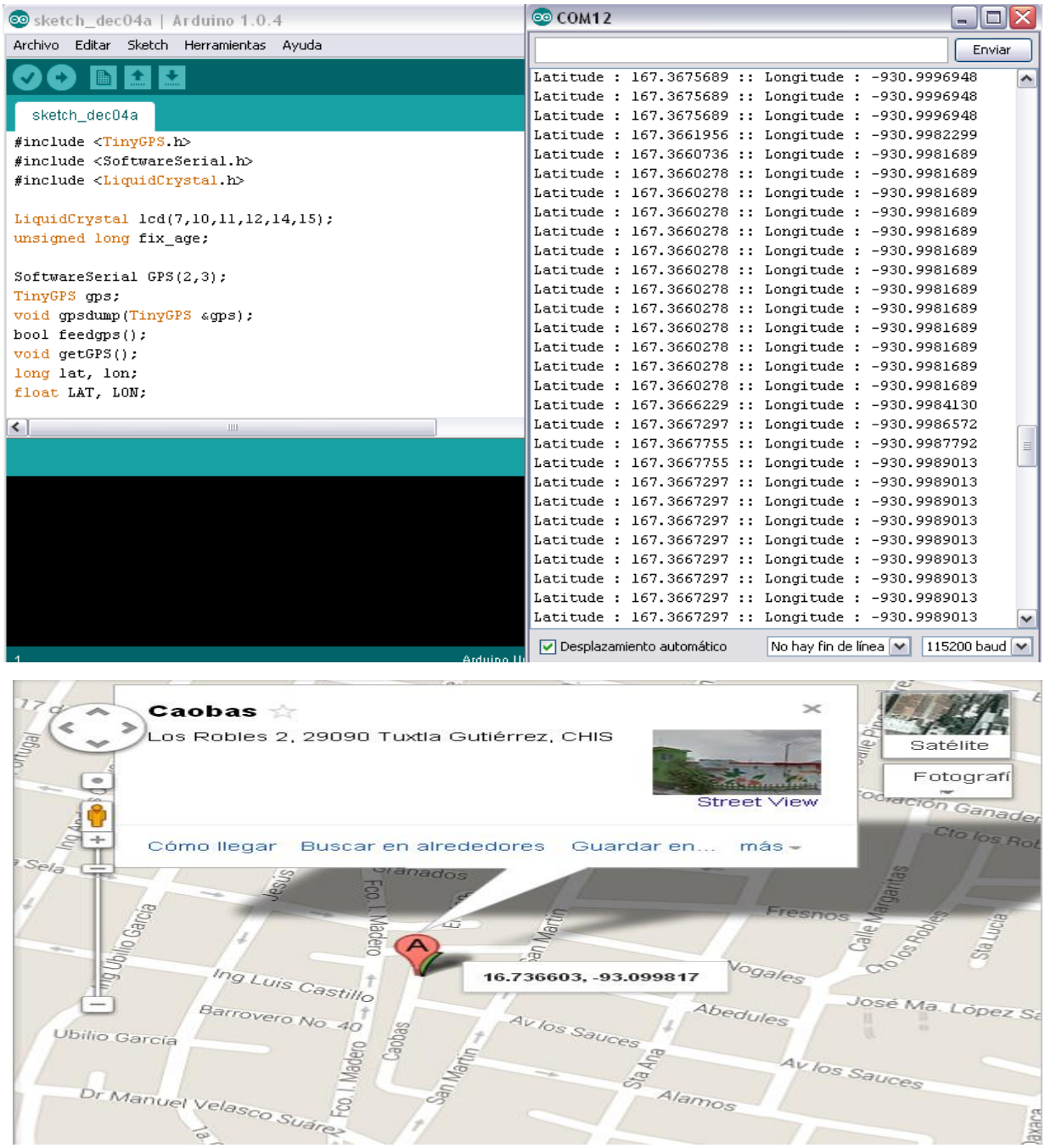
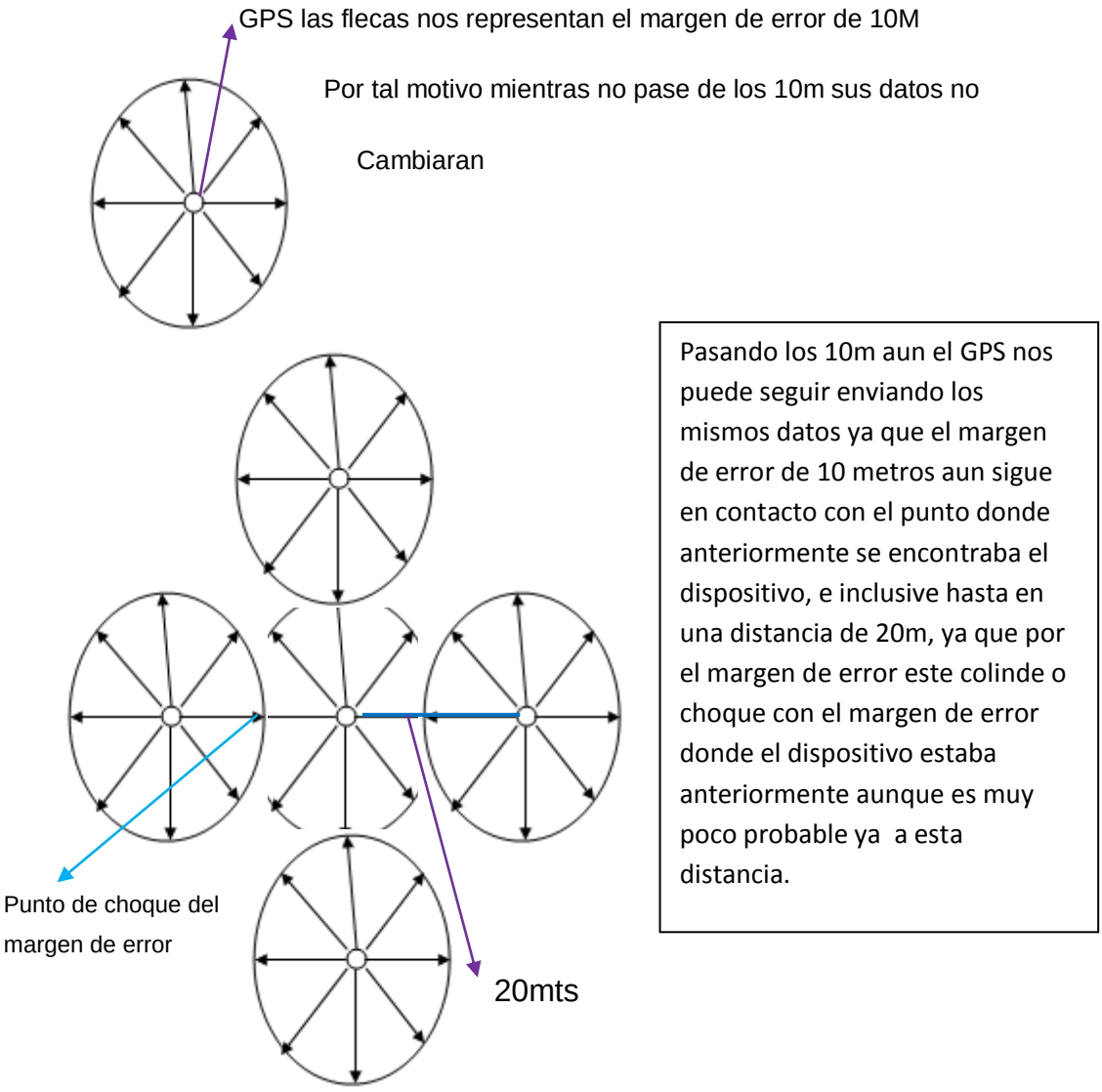


Figura 5.3 Captación de coordenadas en área aislada y comprobación mediante google search y google maps

Al realizar las pruebas en lugar rodeado en este caso un cuarto completamente cerrado, vemos que lo mismo sucede el margen de error esta dentro de lo permitido, con la única diferencia que ahora nos indica un punto distinto esto es a consecuencia del error de 10 metros que tiene permitido el GPS.

Cabe señalar que si se nota nos indica colonias diferentes esto es porque la calle francisco I madero divide la colonia El roble de la colonia Caminera.

Otros dato importante obtenido, en la pruebas realizadas fue que después de pasar un tiempo determinado los datos que recibe el GPS queda constante, ¿Por qué? Enseguida se explicara mediante un esquema.



Esquema de explicación del margen de erros del GPS

5.2 CONCLUSIONES Y RECOMENDACIONES.

Conclusiones

En base al desarrollo del proyecto, se pudo notar que los avances tecnológicos ofrecen una mayor facilidad para la realización de proyectos, uno de estos avances es la tecnología bluetooth la cual permite la comunicación inalámbrica a cualquier dispositivo.

Otro aspecto que fue de gran importancia es el avance tecnológico en los móviles como los son los Smartphone, los cuales cuentan con el sistema android, y mediante una aplicación exclusiva de android llamada Blueterm sirvió para el control sobre el robot, el cual tuvo como base el uso de la plataforma Arduino el cual presenta muchas ventajas al utilizarlo como lo es: la compatibilidad con el bluetooth mediante los pines Rx y Tx, el control de los motores, la comunicación con el LCD, la similitud de programación del lenguaje en C, entre otros.

Durante la etapa de implementación, programación y funcionamiento del SKM53 (GPS), se lograron localizaciones con mucha exactitud considerando lo que es su margen de error. Se hicieron las pruebas suficientes para lograr una confiabilidad en los datos que el GPS proporciona comprobándolo con la ayuda de google maps y google search.

Los resultados obtenidos nos indica que la implementación del hardware GPS para el rastreo de robots inalámbricos, los datos que nos arroja son verídicos y confiables, lo cual puede permitir poder ir tras el dispositivo perdido y volver a tener control sobre el y traerlo de vuelta. Mediante el display implementado se logro resolver la incógnita ¿me perdí? ¿Dónde estoy? Ya que con esto en cualquier parte se encuentre el display nos indicara la posición donde se este.

Se lograron cumplir los objetivos específicos, pero no cubrir el 100% el objetivo general, esto es debido a diversos factores, pero principalmente fue el uso del bluetooth el cual nos proporciona poca distancia de comunicación inalámbrica y se ve afectado por el margen de error que tiene el GPS.

Sin embargo la mayor parte de los aspectos que se pretendían lograr fue obtenido con resultados satisfactorios. En el desarrollo de este proyecto se puede considerar de gran confiabilidad para ser tomado como base de futuros proyectos de mayor escala.

Recomendaciones

Como se describió a lo largo de este trabajo el alcance del bluetooth es poco por lo cual presentaría una mayor ventaja sustituirlo por un dispositivo de mayor alcance como lo es el XBee y en la programación del GPS en el Arduino mejorara la programación ya que al usar XBee el efecto que tiene el margen de error del GPS será mínimo a comparación del alcance que posee el XBee.

Con esto se podría llegar a crear un algoritmo donde se puedan programar coordenadas para que el dispositivo pueda ir de manera autónoma a mayor distancia.

Se recomiendo leer las características eléctricas de todos los dispositivos que se empleen, principalmente la alimentación ya que cuentan con un margen específico de 3.3 a 5v ya que indica claramente que puede causar grave daño si se le aplica mayor tensión.

Se debe considerar que siempre que se trabaje con GPS de este tipo el margen de error estará siempre presente, por lo que hay que tomarlo en cuenta para la interpretación de los resultados de acuerdo al proyecto que se este realizando.

.

ANEXOS

ANEXO 1 Programación 1 en Arduino completa (control inalámbrico)

```
/*programa de direccionamiento del prototipo "Implementación de un robot
 *manipulado a control remoto con protocolo de seguridad en caso de perdida
 *de señal de control*/

#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 10, 11, 12, 14, 15);
int pivote=0,sensor1=0,dato2=0;
int Led = 13;       // on/off del circuito
int MotorA1 = 3;       // pin 2 del L293B
int MotorA2 = 5;       // pin 7 del L293B
int MotorB1 = 6;       // pin 15 del L293B
int MotorB2 = 9;       // pin 10 del L293B
int A = 2;          // pin 1 del L293B
int B = 4;    // pin 9 del L293B
int Entrada = 8;
int Valor;
int Der = 16; // luces derecha
int Izq = 17; // luces izquierda

void Atras(){
  digitalWrite(A,HIGH);
  analogWrite(MotorA1,190);
  digitalWrite(MotorA2,LOW);
  digitalWrite(B,HIGH);
  analogWrite(MotorB2,190);
  digitalWrite(MotorB1,LOW);
   digitalWrite(Led,HIGH);
  digitalWrite(Izq,LOW);
```

```
  digitalWrite(Der,LOW);

lcd.clear(); // borramos lo que hay el display

lcd.setCursor(0, 0); // Nos posicionamos en el primer caracter de la primer linea 0,0

lcd.print("....DE REVERSA!!!");// Escribimos en la primera linea el display

lcd.setCursor(0, 1); // Nos posicionamos en el primer caracter de la segunda linea 0,0

lcd.print("....ING ELECTRONICA"); // Escribimos en la segunda linea del display

delay(250); // retardamos la ejecucion del programa 250 milisegundos

 Serial.println("atras");

 }

void Adelante () {

  digitalWrite(A,HIGH);

  analogWrite(MotorA2,190);

  digitalWrite(MotorA1,LOW);

  digitalWrite(B,HIGH);

  analogWrite(MotorB1,190);

  digitalWrite(MotorB2,LOW);

lcd.clear(); // borramos lo que hay el display

lcd.setCursor(0, 0); // Nos posicionamos en el primer caracter de la primer linea 0,0

lcd.print("....ADELANTE!!!");// Escribimos en la primera linea el display

lcd.setCursor(0, 1); // Nos posicionamos en el primer caracter de la segunda linea 0,0

lcd.print("....ING ELECTRONICA"); // Escribimos en la segunda linea del display

delay(250); // retardamos la ejecucion del programa 250 milisegundos

  digitalWrite(Led,HIGH);

  digitalWrite(Izq,HIGH);

  digitalWrite(Der,HIGH);

    Serial.println("adelante");

  }

  void Derecha() {
```

```
  digitalWrite(A,HIGH);
  analogWrite(MotorA1,190);
  digitalWrite(MotorA2,LOW);
  digitalWrite(B,HIGH);
  analogWrite(MotorB1,190);
  digitalWrite(MotorB2,LOW);
lcd.clear(); // borramos lo que hay el display
lcd.setCursor(0, 0); // Nos posicionamos en el primer caracter de la primer linea 0,0
lcd.print("....DERECHA!!!");// Escribimos en la primera linea el display
lcd.setCursor(0, 1); // Nos posicionamos en el primer caracter de la segunda linea 0,0
lcd.print("....ING ELECTRONICA"); // Escribimos en la segunda linea del display
delay(250); // retardamos la ejecucion del programa 250 milisegundos
 Serial.println("derecha");
  digitalWrite(Led,HIGH);
  digitalWrite(Izq,LOW);
  digitalWrite(Der,HIGH);
 }
 void Izquierda () {
  digitalWrite(A,HIGH);
  analogWrite(MotorA2,190);
  digitalWrite(MotorA1,LOW);
  digitalWrite(B,HIGH);
  analogWrite(MotorB2,190);
  digitalWrite(MotorB1,LOW);
  lcd.clear(); // borramos lo que hay el display
lcd.setCursor(0, 0); // Nos posicionamos en el primer caracter de la primer linea 0,0
lcd.print("....IZQUIERDA!!!");// Escribimos en la primera linea el display
lcd.setCursor(0, 1); // Nos posicionamos en el primer caracter de la segunda linea 0,0
lcd.print("....ING ELECTRONICA"); // Escribimos en la segunda linea del display
```

```
delay(250); // retardamos la ejecucion del programa 250 milisegundos

  Serial.println("izquierda");

   digitalWrite(Led,HIGH);

    digitalWrite(Der,LOW);

    digitalWrite(Izq,HIGH);


  }

  void Paro (){

  digitalWrite(A,LOW);

  digitalWrite(MotorA1,HIGH);

  digitalWrite(MotorA2,HIGH);

  digitalWrite(B,LOW);

  digitalWrite(MotorB1,HIGH);

  digitalWrite(MotorB2,HIGH);

lcd.clear(); // borramos lo que hay el display

lcd.setCursor(0, 0); // Nos posicionamos en el primer caracter de la primer linea 0,0

lcd.print("....ALTO TOTAL!!!");// Escribimos en la primera linea el display

lcd.setCursor(0, 1); // Nos posicionamos en el primer caracter de la segunda linea 0,0

lcd.print("....ING ELECTRONICA"); // Escribimos en la segunda linea del display

delay(250); // retardamos la ejecucion del programa 250 milisegundos

  Serial.println("paro");

   digitalWrite(Led,LOW);

   digitalWrite(Izq,LOW);

   digitalWrite(Der,LOW);

  }


 void setup (){

  lcd.begin(16, 2);

  pinMode(14,OUTPUT);

  pinMode(15,OUTPUT);

  pinMode(Der,OUTPUT);

  pinMode(Izq,OUTPUT);
```

```
  pinMode(Led,OUTPUT);
  pinMode(MotorA1,OUTPUT);
  pinMode(MotorA2,OUTPUT);
  pinMode(MotorB1,OUTPUT);
  pinMode(MotorB2,OUTPUT);
  pinMode(A,OUTPUT);
  pinMode(B,OUTPUT);
  pinMode(Entrada,INPUT);
  Serial.begin(9600);
}
 void loop() {
  Valor=digitalRead(Entrada);
  if (Valor== HIGH)
  {
  while (Serial.available()>0)
  {

  char dato= Serial.read();
   switch(dato)
  // control simple
  {
    case 'r':
    {
    Atras();
     break;
  }
  case 'a':
  {
    Adelante();
    break;
  }
 case 'i':
  {
    Izquierda();
     break;
  }
  case 'd':
  {
    Derecha();
```

```
  break;
}
  case 'p':
  {
  Paro();
    sensor1=1;
  dato2=5;
   break;
}
// comando de seguridad de seguridad
  case 'R':
 {
  Atras();
  delay (4000);
  Paro();
  sensor1=1;
  dato2=1;
   break;
 }
 case 'A':
  {
   Adelante();
   delay(4000);
 Paro();
 sensor1=1;
 dato2=2;
  break;
}
case 'I':
  {
   Izquierda();
   delay(4000);
 Paro();
 sensor1=1;
 dato2=3;
  break;
}
case 'D':
  {
   Derecha();
```

```
    delay(4000);
  Paro();
  sensor1=1;
  dato2=4;
    break;
 }
    case 'P':
  {
  Paro();
  sensor1=1;
  dato2=5;
    break;
}
    case 'W':
  {
  Atras();
    delay (4000);
  Paro();
  sensor1=1;
  dato2=6;
    break;
}
    case 'X':
  {
  Adelante();
    delay (4000);
  Paro();
  sensor1=1;
  dato2=7;
    break;
}
    case 'Y':
  {
  Atras();
    delay (4000);
  Paro();
  sensor1=1;
  dato2=8;
    break;
}
```

```
   case 'Z':
  {
  Adelante();
   delay (4000);
  Paro();
  sensor1=1;
  dato2=9;
   break;
}
}
 }
      if(sensor1==1)
 {
   pivote=pivote+1;
   delay (100);
 }
     if(pivote==150)
 {
  switch(dato2)
{
   case 1:
 { Adelante();
  delay (4000);
   Paro();
  pivote=0;
  sensor1=0;
break;}
   case 2:
 { Atras();
  delay (4000);
   Paro();
  pivote=0;
  sensor1=0;
break;}
   case 3:
 { Derecha();
  delay (4000);
   Atras();
  delay(4000);
  Paro();
```

```
 pivote=0;
 sensor1=0;
 break;
}
  case 4:{
    Izquierda();
     delay (4000);
    Atras();
    delay(4000);
 Paro();
 pivote=0;
 sensor1=0;
break;}

 case 5:{
    Paro();
 pivote=0;
 sensor1=0;
 break;}
   case 6:
{ Adelante();
 delay (4000);
  Izquierda();
   delay (4000);
 Atras();
   delay (4000);
Paro();
pivote=0;
sensor1=0;
break;}
  case 7:
{ Atras();
 delay (4000);
  Izquierda();
   delay (4000);
 Atras();
   delay (4000);
Paro();
pivote=0;
sensor1=0;
```

```
break;}
 case 8:
{ Adelante();
 delay(4000);
  Derecha();
  delay(4000);
  Atras();
  delay(4000);
 Paro();
 pivote=0;
 sensor1=0;
break;}
  case 9:
{ Atras();
 delay (4000);
  Derecha();
  delay (4000);
  Atras();
  delay (4000);
 Paro();
 pivote=0;
 sensor1=0;
break;}
 }
  }}}
```

Anexo 2 Especificaciones del SKM 53

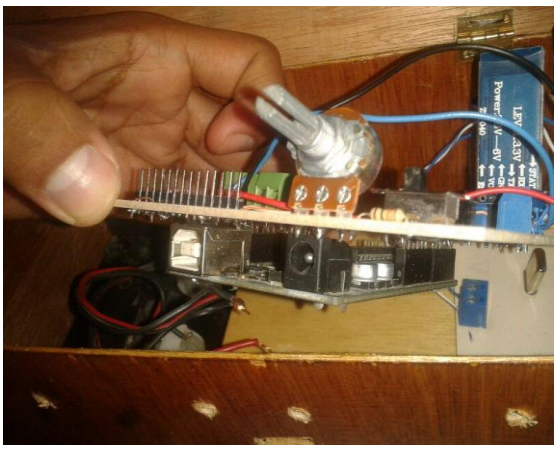
Tipo de receptor	L1 frecuncia de banda, C/A code 22 Tracking / 66 Canales de lectura	
Sensibilidad	Tracking Acquisition	-165dBm -148dBm
Precisión	Posición Velocidad Tiempo (PPS)	3mts. 3D RMS sin SA 0.1m/s sin SA 60ns RMS
Tiempo de lectura	Cold Start Warm Start Hot Start Re-adquisición	36s 33s 1s <1s
Consumo de energía	Tracking Adquisición Sleep/Standby	<30mA @ 3V Vcc 40mA TBD
Frecuencia de actualización de datos de navegación	1Hz	
Límites de operación	altitud velocidad Aceleración	Max 18,000m Max 515m/s Menor a 4g
Especificaciones de la antena	OutlineDimension Center Frequency Bandwidth Impedancia Axial Ratio Polarizacion	18.2 x 18.2 x 4.0 mm 1575 ± 3 MHz 10 MHz min 50 Ω 3 dB max RHCP
Dimensiones y peso	Dimensiones Peso	30mm x20mm x 11.4mm 9g
Fuente de poder	VCC Corriente	5V ±5% 55mA(typical)
Entorno	Temperatura de operación Temperatura de almacenamiento Humedad	40 ~ +85 (sin batería de respaldo) 0 ~ +125

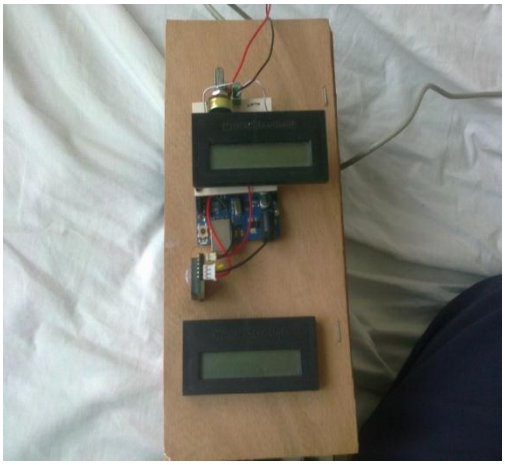
Anexo 3 Sketch 2 Programación del GPS SKM 53

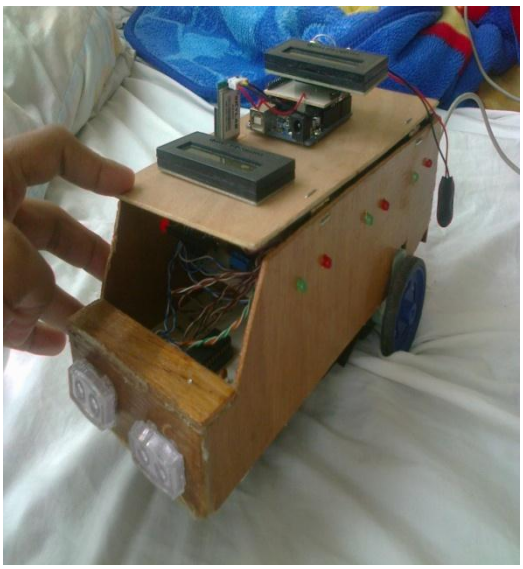
```
#include <TinyGPS.h>

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(7,10,11,12,14,15);

unsigned long fix_age;

SoftwareSerial GPS(2,3);

TinyGPS gps;

void gpsdump(TinyGPS &gps);

bool feedgps();

void getGPS();

long lat, lon;

float LAT, LON;

void setup(){

  GPS.begin(9600);

  Serial.begin(115200);

  lcd.begin(16, 2);

}


void loop(){

  long lat, lon;

  unsigned long fix_age, time, date, speed, course;

  unsigned long chars;

  unsigned short sentences, failed_checksum;

  gps.get_position(&lat, &lon, &fix_age);

  getGPS();

  lcd.setCursor(0,0);

  lcd.print("Lt: ");

  lcd.print(LAT/100000,7);

  lcd.setCursor(0,1);

  lcd.print("Ln: ");

  lcd.print(LON/100000,7);

  Serial.print("Latitude : ");

  Serial.print(LAT/100000,7);

  Serial.print(" :: Longitude : ");

  Serial.println(LON/100000,7);


  delay(1000);

}

void getGPS(){

  bool newdata = false;
```

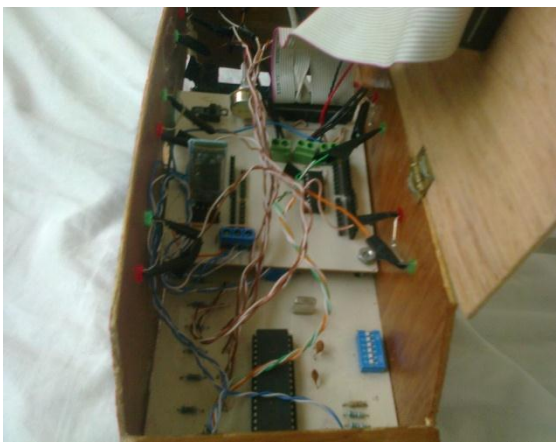
```
 unsigned long start = millis();
  while (millis() - start < 1000)
 {
   if (feedgps ()){
     newdata = true;
   }
 }
 if (newdata)
 {
   gpsdump(gps);
 }
}
bool feedgps(){
 while (GPS.available())
 {
   if (gps.encode(GPS.read()))
    return true;
 }
 return 0;
}
void gpsdump(TinyGPS &gps)
{
 gps.get_position(&lat, &lon);
 LAT = lat;
 LON = lon;
 {
  feedgps();
 }}
```

Anexo 4 Imágenes del dispositivo









Anexo 5 Programación en MPLAB del Pic

```
; Residencia profesional
;------------------
                #INCLUDE
        <P16F877A.INC>
                __CONFIG
        _WDT_OFF & _CP_OFF &
_CPD_OFF & _LVP_OFF & _XT_OSC
                LIST
        P=16F877A
;-------------------Declaracion de variables
        estado equ 0x03
        rp0 equ 0x05
        rp1 equ 0x06
        ptoa equ 0x05
        ptob equ 0x06
        adcon1 equ 0x1f
        aux equ 0x20
   aux1 equ 0x21
;------------------
        org 0x00
;-------------------Configuracion de
registros
        bsf estado, rp0
        bcf estado, rp1
        movlw 0xff
        movwf ptoa
        movlw 0x00
        movwf ptob
        movlw 0x06
        movwf adcon1
        bcf estado, rp0
;-------------------Bloque principal
ini clrf ptob
 chek btfsc ptoa,3
 goto chek3
btfss ptoa,0
 goto chek2
 incf aux,0
 movwf ptob
 movwf aux1
```

```
a btfss ptoa,1
goto chek
movf ptob, 0
movwf ptob
goto a
 chek2 btfsc ptoa,2
 goto ini
 movf aux1, 0
 movwf ptob
movwf aux
goto chek
chek3 btfss ptoa, 0
                goto chek2
                decf aux,0
                movwf ptob
                movwf aux1
                goto chek
;----
 end
```

Bibliografías y páginas WEB

[1] Arduino Curso Práctico de Formación
Oscar Torrente Artero
Editorial RC libros 2013

[2] GPS Fácil
Lawrence Letham
Editorial PAIDOTRIBO 2001

[3] Introducción al análisis de circuitos
Robert L. Boylestad
Editorial Pearson 2004

WEB grafia

[4] Mosqueda Barajas, Luis, *"Robot Detector de Obstáculos"*, (Web en línea), http://www.tlalpan.uvmnet.edu/oiid/download/Detector%20de%20obst%C3%A1culos_04_ING_ITE_PIT_E.pdf, Consulta 03/09/2013.

[5] "Servomotor", (Web en línea), http://es.wikipedia.org/wiki/Servomotor, Consulta 03/09/2013.

[6] "Guía Básica de arduino", (Documento en línea), http://www.slideshare.net/hugoangaritaespinosa/libro-kit-basico, Consulta 05/09/2013.

[7] "Generalidades Robótica", (Web en línea), http://generalidadesrobotica.blogspot.mx/p/que-es.html, Consulta 08/09/2013.

[8] Perdigon, Marilyn, *"Los primeros robots"*, Noviembre 14 de 2012, (Web en línea), http://marilynpg.blogspot.mx/2012/11/historia-de-la-robotica.html, Consulta 09/09/2013.

[9] Maroñas Molano, Lucas, *"Comunicaciones Inalámbricas"*, (Documento en línea), https://docs.google.com/document/preview?hgd=1&id=1m0yaPCa33a-zlSNQlIvfQ0R6xzQxFhHaG47rHl132Xo, Consulta 14/09/2013.

[10] Sulbaran, Heli, *"Nace la tecnología Bluetooth"*, Diciembre 8 de 2013. (Web en linea), http://helisulbaran.blogspot.mx/2013/12/07-de-diciembre-2000-nace-la-tecnologia.html, Consulta 25/09/2013.

[11] Plaza, Exequiel, *"La historia del Bluetooth"*, Septiembre 28 de 2011, (Web en línea), http://www.wayerless.com/2011/09/la-historia-del-nacimiento-de-bluetooth/, Consulta 02/10/2013.

[12] "Arduino más allá del hardware libre", (Web en línea), http://www.somoslibres.org/modules.php?name=News&file=article&sid=5581, Consulta 05/10/2013.

[13] "Placa Arduino Uno", Diciembre de 2012, (Web en línea), http://www.menosmedia.org/spip.php?article43, Consulta 05/10/2013.

[14] "Arduino", Abril 14 de 2012, (Web en línea), http://preciosanchez.blogspot.mx/2012/07/arduino.html, Consulta 05/10/2013.

[15] "Plataformas de desarrollo móviles actuales", (Documento en línea), http://es.scribd.com/doc/170157049/Plataformas-de-desarrollo-moviles-actuales-docx, Consulta 11/10/2013.

[16] "Que es Android", (Web en línea), http://yosoyandroid.com/diccionario-android/que-es-android-2/, Consulta 11/10/2013.

[17] "Android", (Web en línea), http://es.wikipedia.org/wiki/Android, Consulta, 14/10/2013.

[18] "GPS Basics", (Web en línea), https://www.u-cursos.cl/forestal/2011/1/EF016/1/material_docente/previsualizar?id_material=487198, Consulta 21/10/2013.

[19] Perdomo Ramírez, Douglas, *"Equipos de Posicionamiento por Satélite GPS y DGPS"*, Octubre 2009, (Web en línea), http://www.oocities.org/es/douglas_perdomo76/epps/eppstrabajo1.html, Consulta 22/10/2013.