



UNIVERSIDAD AMERICANA DE ACAPULCO

“EXCELENCIA PARA EL DESARROLLO”

FACULTAD DE INGENIERÍA EN COMPUTACIÓN

INCORPORADA A LA UNIVERSIDAD NACIONAL

AUTÓNOMA DE MÉXICO

CLAVE DE INCORPORACIÓN 8852-16

**“SISTEMA DE MONITOREO USANDO
COMUNICACIÓN POR RADIOFRECUENCIA PARA EL
CONTROL DE UN ROBOT EXPLORADOR”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA

EDGAR FIERRO ROJAS

FRANCISCO JAVIER MONTERO JIMÉNEZ

JUAN RAÚL SÁNCHEZ VILLA

DIRECTOR DE TESIS

M.C. DAVID JAIME GONZÁLEZ MAXINEZ



ACAPULCO, GUERRERO ABRIL DE 2015



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

Gracias a la Universidad Americana de Acapulco por brindarnos excelentes servicios en nuestro aprendizaje.

A la Facultad de Ingeniería la cual nos brindó apoyo y herramientas en el transcurso de la carrera.

Al Dr. David G. Maxinez que siempre me ha motivado a superarme gracias por su dedicación y entusiasmo.

Al Profesor Arturo Ibarra Colín que en este momento ya no se encuentra con nosotros, fue un gran maestro y amigo, siempre paciente y siempre sonriente, en paz descanse Profesor.

A todos nuestros profesores gracias por su tiempo, apoyo y su sabiduría que transmitieron para lograr este gran paso.

DEDICATORIA

Quiero agradecer a mi familia por todo el apoyo que me brindaron a través de mis estudios hasta la culminación de mi carrera profesional.

A mi madre Linda.

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi padre Juan.

Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

A mi hermana Yareli.

Por estar conmigo, apoyarme siempre y más que nada ser un gran ejemplo a seguir para mí.

A mi abuelita Ruperta.

Por quererme y apoyarme siempre, esto también se lo debo a usted.

A mi novia Paola.

Por estar conmigo, apoyarme siempre en los buenos y malos momentos, te agradezco todo ese apoyo

Juan Raúl Sánchez Villa

DEDICATORIA

Le doy gracias a mis padres, a mis maestros y a mis compañeros por participar y hacer realidad una de las etapas más importantes de mi vida, como lo es mi formación profesional.

Francisco Montero

DEDICATORIA

Dedico esta Tesis a mi Madre quien me apoyo durante toda la carrera, siempre estuvo a mi lado sin importar las dificultades que pasamos juntos.

A mi familia por brindarme su apoyo constante, consejos y darme entusiasmo.

A Dios que todos los días me dio fortaleza y paciencia para seguir adelante y me permitió haber llegado hasta este momento tan importante de mi vida.

A mis compañeros Juan Raúl y Francisco Montero quienes fueron un gran apoyo en el desarrollo de esta tesis.

Edgar Fierro Rojas

ÍNDICE GENERAL

| | Página |
|---|-----------|
| ÍNDICE DE FIGURAS | 9 |
| INTRODUCCIÓN..... | 11 |
| PLANTEAMIENTO DEL PROBLEMA..... | 12 |
| JUSTIFICACIÓN..... | 13 |
| HIPÓTESIS..... | 14 |
| OBJETIVO GENERAL..... | 15 |
| OBJETIVOS ESPECÍFICOS..... | 15 |
| | |
| CAPÍTULO.1 ROBÓTICA DE EXPLORACIÓN ESTADO DEL ARTE..... | 17 |
| 1.1 ¿Que es la robótica? | 18 |
| 1.2 Historia de la robótica..... | 18 |
| 1.3 Robots exploradores..... | 20 |
| 1.3.1 Exploración de minas..... | 21 |
| 1.3.2 Robot de exploración espacial..... | 22 |
| 1.3.3 Robot de exploración militar | 23 |
| 1.3.4 Robot de exploración arqueológica..... | 24 |
| 1.4 Control por radiofrecuencia..... | 25 |
| | |
| CAPÍTULO 2 RADIOFRECUENCIA CONCEPTOS Y DEFINICIONES BASICAS..... | 27 |
| 2.1 ¿Qué es la comunicación?..... | 28 |
| 2.1.1 ¿Que es la comunicación a distancia?..... | 28 |

| | |
|--|----|
| 2.1.2 ¿Que es la Radiofrecuencia?..... | 30 |
| 2.1.3 Frecuencias de operación..... | 31 |
| 2.1.4 Control mediante sistemas de radiofrecuencias..... | 33 |
| 2.1.5 Tipos de comunicación en serie..... | 34 |
| 2.2 Robótica móvil..... | 36 |
| 2.2.1 Locomoción..... | 37 |
| 2.2.2 Diseño de ruedas..... | 39 |
| 2.2.3 Diferencial..... | 39 |
| 2.2.4 Diseño sincronizado..... | 40 |
| 2.2.5 Diseño de triciclo y coche..... | 41 |
| 2.2.6 Cinemática..... | 41 |
| 2.2.7 Forma del robot..... | 42 |
| 2.2.8 Adaptación de plataformas móviles..... | 44 |
| 2.3 ¿Qué es una cámara ip? | 46 |
| 2.3.1 Características de una cámara ip..... | 47 |
| 2.4 Microprocesadores..... | 48 |
| 2.4.1 Microcontroladores..... | 50 |
| 2.4.2 ¿Que es un microcontrolador?..... | 51 |
| 2.4.3 Arquitecturas de computadora..... | 52 |
| 2.4.3.1 Arquitectura Von Neumann..... | 52 |
| 2.4.3.2 Arquitectura Harvard..... | 54 |
| 2.4.4 Memoria..... | 61 |
| 2.4.5 Interrupciones..... | 65 |
| 2.4.6 Entradas y salidas de propósito general..... | 66 |
| 2.4.7 Temporizadores y contadores..... | 66 |
| 2.4.8 Conversor analógico/digital..... | 67 |

| | |
|---|-----|
| 2.4.9 Puertos de comunicación..... | 68 |
| 2.4.10 Comparadores..... | 70 |
| 2.4.11 Modulador de ancho de pulsos..... | 71 |
| 2.4.12 Memoria de datos no volátil..... | 71 |
| 2.4.13 Familias de microcontroladores..... | 72 |
| 2.5 Elección de un microcontrolador..... | 74 |
| 2.5.1 Reglas básicas..... | 74 |
| 2.6 Arduino..... | 77 |
| 2.6.1 Entradas y salidas..... | 78 |
| 2.6.2 Funciones básicas y operadores..... | 79 |
| 2.7 Control de motores..... | 79 |
| 2.7.1 Descripción L293B..... | 85 |
| 2.7.2 Diagrama de bloques..... | 86 |
| 2.8 Caso de estudio..... | 87 |
| 2.9 Sistema mínimo..... | 88 |
| 2.10 Diseño de interfaz de radiofrecuencia..... | 88 |
| | |
| CAPÍTULO 3 PROGRAMACIÓN E INTEGRACIÓN DEL SISTEMA DE APLICACIÓN...90 | |
| 3.1 Integración de los Sistemas..... | 91 |
| 3.2 Desarrollo de interfaz gráfica..... | 114 |
| 3.3 Programación en Android..... | 118 |
| 3.4 Conectar cámara ip al navegador..... | 122 |
| 3.5 Conectar cámara ip al celular..... | 125 |

| | |
|--|-----|
| CAPÍTULO 4 RESULTADOS Y PRUEBAS | 129 |
| 4.1 Pruebas del sistema de control..... | 130 |
| 4.2 Pruebas del sistema de monitoreo..... | 134 |
| 4.3 Pruebas del control remoto..... | 137 |
| 4.4 Pruebas con la aplicación android..... | 139 |
| | |
| Conclusiones y Trabajo a Futuro..... | 141 |
| Referencias..... | 142 |
| Apéndice..... | 144 |

ÍNDICE DE FIGURAS

| CAPÍTULO 1 | Página |
|--|---------------|
| Figura 1. Robot curiosity en la superficie marciana..... | 17 |
| Figura 2. Robótica y Áreas de apoyo..... | 18 |
| Figura 3. Estructuras del diseño de androides..... | 19 |
| Figura 4. Robot de Exploración Espacial..... | 20 |
| Figura 5. Robot de Exploración..... | 21 |
| Figura 6. Robot Curiosity..... | 22 |
| Figura 7. Robot de exploración militar..... | 23 |
| Figura 8. Robot Tláloc II usado en exploraciones arqueológicas del Templo de la Serpiente Emplumada en México..... | 24 |
| Figura 9. Chip transmisor y receptor para comunicaciones inalámbricas..... | 25 |
| | |
| CAPÍTULO 2 | |
| Figura 10. Sistema de comunicación de radiofrecuencia..... | 27 |
| Figura 11. Sistema de Comunicación..... | 28 |
| Figura 12. Sistema de comunicación alámbrica..... | 28 |
| Figura 13. Comunicación inalámbrica..... | 29 |
| Figura 14. Longitud de ondas..... | 29 |
| Figura 15. Espectro electromagnético..... | 30 |
| Figura 16. Fórmula para calcular la distancia y velocidad de la frecuencia de onda..... | 31 |
| Figura 17. Sistema típico de comunicación por radiofrecuencia..... | 33 |
| Figura 18. Tipo de comunicación simplex..... | 34 |
| Figura 19. Tipo de comunicación dúplex..... | 34 |
| Figura 20. Tipo de comunicación full dúplex..... | 35 |
| Figura 21. Aplicaciones en la Robótica..... | 36 |
| Figura 22. Locomoción tipo Ruedas..... | 37 |
| Figura 23. Locomoción tipo cadenas o skidsteer..... | 38 |
| Figura 24. Diseño de ruedas..... | 39 |
| Figura 25. Diseño Diferencial..... | 40 |
| Figura 26. Diseño de Triciclo..... | 41 |
| Figura 27. Cinemática de Ángulos del Robot..... | 42 |
| Figura 28. Robot Cilíndrico..... | 43 |
| Figura 29. Robot Cuadrado..... | 43 |
| Figura 30. Diseño de Proyecto con un solo motor..... | 45 |
| Figura 31. Descripción del funcionamiento de una cámara IP..... | 46 |
| Figura 32. Cámara IP..... | 47 |
| Figura 33. Tipos de microcontroladores más usados..... | 50 |
| Figura 34. Arquitectura Von Neumann..... | 52 |
| Figura 35. Chip Intel 8051..... | 54 |
| Figura 36. Arquitectura Harvard..... | 54 |
| Figura 37. Unidad de control en un procesador..... | 56 |
| Figura 38. Unidad aritmética lógica..... | 57 |
| Figura 39. Puerto serie para comunicaciones de la PC..... | 68 |
| Figura 40. Puerto de comunicación USB..... | 69 |
| Figura 41. Puerto Jack tipo Ethernet..... | 69 |
| Figura 42. Datasheet Arduino (Modelo Atmega 328)..... | 77 |
| Figura 43. Arduino modelo uno (Chip Atmega 328), Tipos de Funciones..... | 78 |
| Figura 44. Datasheet de Entradas y salidas Arduino..... | 79 |
| Figura 45. (Chip L293B) Descripción..... | 85 |

| | |
|---|----|
| Figura 46. (Diagrama L293B)..... | 86 |
| Figura 47. Diseño a Seguir del Modo de Comunicación del Proyecto..... | 88 |

CAPÍTULO 3

| | |
|---|-----|
| Figura 48. Entornos de desarrollo a usar en el proyecto..... | 90 |
| Figura 49. Armado Virtual de Comunicación del Transmisor..... | 92 |
| Figura 50. Armado físico del Circuito..... | 92 |
| Figura 51. Selección del Programa..... | 98 |
| Figura 52. Tipo de placa para Programar..... | 100 |
| Figura 53. Selección del Puerto Serial..... | 100 |
| Figura 54. Introducción del Código al Compilador..... | 101 |
| Figura 55. Compilación..... | 101 |
| Figura 56. Proceso de compilación..... | 101 |
| Figura 57. Transferencia del Código a la placa Arduino..... | 102 |
| Figura 58. Diseño Virtual de Comunicación del Receptor..... | 102 |
| Figura 59. Montado Electrónico del Proyecto..... | 103 |
| Figura 60. Montaje de placas de Transmisión y Recepción..... | 113 |
| Figura 61. Ventana principal del entorno de desarrollo..... | 114 |
| Figura 62. Selección de Aplicación de Windows Forms..... | 114 |
| Figura 63. Selección del componente puerto serial..... | 115 |
| Figura 64. Librerías necesarias..... | 115 |
| Figura 65. Implementación de botones..... | 116 |
| Figura 66. Entorno de desarrollo MIT App Inventor..... | 118 |
| Figura 67. Implementación de botones en la aplicación..... | 119 |
| Figura 68. Diseño de bloques para la programación de la aplicación..... | 120 |
| Figura 69. Instrucción para enviar datos por el puerto serial..... | 121 |
| Figura 70. Robot monitoreado con cámara IP..... | 122 |
| Figura 71. Dirección URL de cámara IP..... | 123 |
| Figura 72. Ingreso de usuario y contraseña de la cámara IP..... | 123 |
| Figura 73. Monitoreo de la cámara IP..... | 124 |
| Figura 74. Dirección de cámara ip en el celular..... | 125 |
| Figura 75. Introducción de usuario y contraseña..... | 126 |
| Figura 76. Transmisión en vivo de la cámara ip al celular..... | 127 |

CAPÍTULO 4

| | |
|---|-----|
| Figura 77. Interfaz de Control..... | 130 |
| Figura 78. Interfaz establecida con el puerto serial..... | 130 |
| Figura 79. Envío de datos mediante el puerto serial..... | 131 |
| Figura 80. Comunicación por medio de radiofrecuencia..... | 132 |
| Figura 81. Encendido de leds para verificar transmisión de la interfaz de la CPU..... | 132 |
| Figura 82. Instrucción de ir hacia la izquierda con indicadores leds..... | 133 |
| Figura 83. Apagado de leds indica detener motores..... | 133 |
| Figura 84. Enlace de cámara IP con interfaz de monitoreo del robot..... | 134 |
| Figura 85. Sistema del control de la cámara y control del Robot..... | 135 |
| Figura 86. Cámara Infrarroja para acceder a zonas oscuras..... | 136 |
| Figura 87. Control de mando a distancia del robot..... | 137 |
| Figura 88. Control del robot y enlace de cámara ip..... | 138 |
| Figura 89. Abrir aplicación en el celular..... | 139 |
| Figura 90. Aplicación android para el control del robot..... | 140 |

INTRODUCCIÓN

En esencia la robótica es la ciencia y la tecnología de los robots. Se ocupa del diseño, manufactura y aplicaciones de los mismos combinando para diversas disciplinas, algunas de ellas son: mecánica, electrónica, informática, inteligencia artificial, ingeniería de control, autómatas programables, máquinas de estados...etc.

En la actualidad, los robots tanto comerciales como industriales son utilizados para realizar tareas de forma automática y exacta a través de la repetición de la tarea de cuestión.

En este trabajo de tesis se construye un robot móvil con capacidad de desplazamiento, controlado por un sistema de radio frecuencia, para aplicaciones de monitoreo y exploración.

En el capítulo 1 se hablara sobre la historia de la robótica y sus antecedentes así también como los robots exploradores y su uso en las diferentes áreas.

En el capítulo 2 se hablara sobre las herramientas a usar para la realización y programación del proyecto así también como una introducción de sus funciones básicas.

En el capítulo 3 explicaremos como se implementaron todos los códigos de programación usados con una breve explicación y funcionamiento.

En el capítulo 4 mostraremos las pruebas de funcionamiento del robot.

PLANTEAMIENTO DEL PROBLEMA

Hoy en día la exploración y monitoreo de lugares de difícil acceso se ha venido realizando a través de mecanismos o sistemas robotizados que minimizan el riesgo del ser humano en estas tareas. El desarrollo de nuevas tecnologías y disciplinas de ingeniería permiten actualmente que los robots realicen tareas que anteriormente recaían en la pericia y destreza de los seres humanos con los riesgos inherentes a este tipo de trabajo.

Nuestro proyecto de tesis desarrolla un prototipo de robot móvil con la capacidad de realizar la exploración de forma inalámbrica y segura para el ser humano.

JUSTIFICACIÓN

Un robot móvil controlado por radiofrecuencia tiene la capacidad no solo de desplazarse sobre un espacio o ambiente de trabajo no ideal para un ser humano sino también puede ser utilizado como un sistema de transporte de materiales o sustancias tóxicas o peligrosas.

El sistema de comunicación por radiofrecuencia es uno de los métodos más económicos y eficientes para el envío de datos ya que considera al aire como el elemento de transmisión de información.

HIPÓTESIS

Determinar que la radiofrecuencia es un tipo de comunicación muy utilizado que permitirá una comunicación rápida y eficaz entre un transmisor y un receptor. Consideramos que el robot móvil podrá ser monitoreado y controlado a través de radiofrecuencia sin ningún tipo de interferencia en un rango de distancia de 30 metros, al contar con una interfaz gráfica de monitoreo y un control remoto inalámbrico independiente, el cual nos permitirá explorar más fácilmente terrenos con muy poca accesibilidad y condiciones inapropiadas que se vayan presentando.

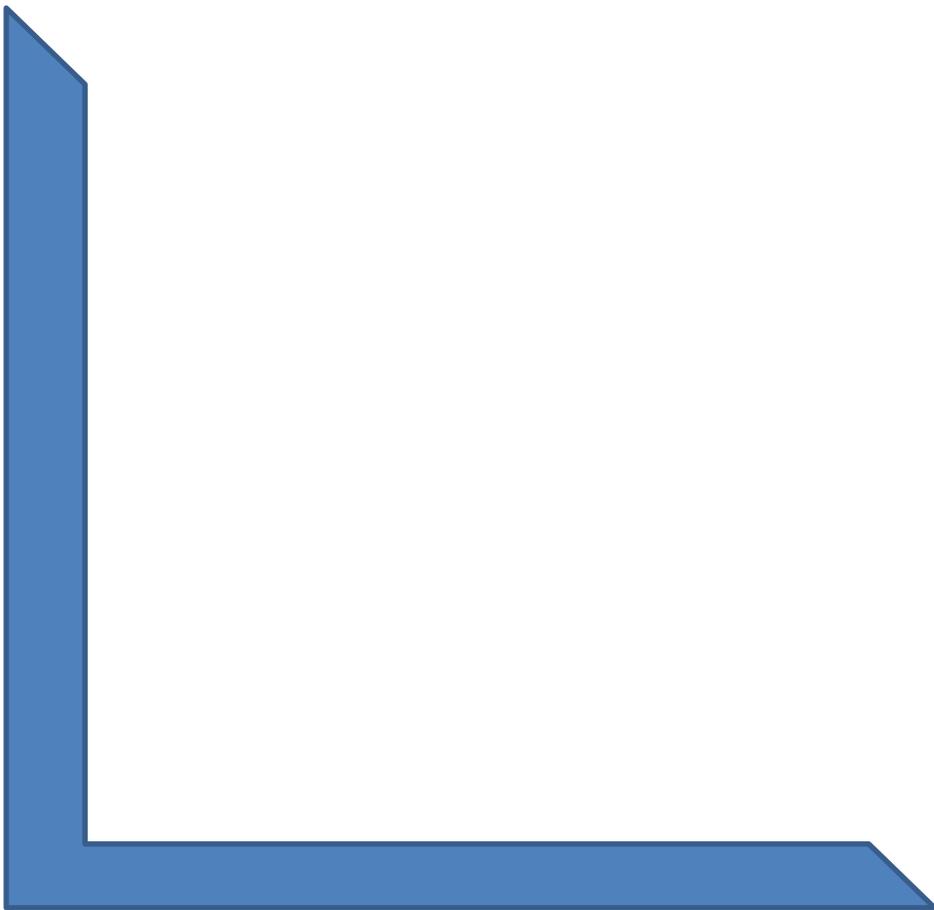
OBJETIVO GENERAL

Realizar una interfaz de comunicación de radiofrecuencia para controlar y monitorear un robot móvil que permita la transmisión de imagen y exploración de diversas situaciones ocurridas de propósito general.

OBJETIVOS ESPECÍFICOS

- Analizar el tipo de comunicación a utilizar probando su eficacia y rapidez de transmisión y recepción.
- Analizar e implementar el tipo de cámara para la función de monitoreo.
- Diseñar un diagrama de comunicación del proyecto.
- Construir una plataforma móvil controlada por radiofrecuencia.
- Realizar la programación del microcontrolador AVR de Arduino UNO para el control por radiofrecuencia.
- Realizar una interfaz de comunicación gráfica la cual tendrá el control del robot y control de la cámara para la navegación del robot.
- Realizar un control independiente a la PC como sistema de apoyo o sistema de emergencia.

CAPÍTULO 1



Capítulo I

ROBÓTICA DE EXPLORACIÓN ESTADO DEL ARTE

INTRODUCCIÓN

Las funciones que desempeñan hoy en día los robots exploradores son cada vez más satisfactorios para los seres humanos ya que ayudan en diversas tareas que pueden ser de investigación, exploración, análisis de riesgo del lugar, así también pueden ser de ayuda o monitoreo de situaciones a larga distancia sin poner en riesgo la vida de la persona que lo está controlando, como el caso del robot curiosity en la superficie del planeta marte, Figura 1. Existen varias formas de establecer comunicación a distancia por ejemplo el uso de Wi-Fi, Bluetooth, sistemas de detección Infrarrojo, etc.

Sin embargo en la actualidad la forma de comunicación a larga distancia que se utiliza más comúnmente no solo en el control de robots sino también en diferentes aparatos ya sea la televisión, satélites, aviación, etc. Es la radiofrecuencia. Este sistema de comunicación a larga distancia presenta algunas ventajas que lo hacen diferente a otros sistemas por ejemplo, que viaja a través del aire, pueden estar en lugares cerrados sin problemas de transmisión un ejemplo el acceso a cuevas.



Figura 1. Robot curiosity en la superficie marciana

1.1 ¿QUÉ ES LA 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 que le son de ayuda en el diseño son: el álgebra, los autómatas programables, las máquinas de estados, la mecánica o la informática Figura 2.¹

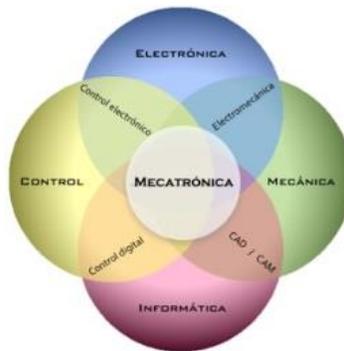


Figura 2. Robótica y Áreas de apoyo

1.2 HISTORIA DE LA ROBÓTICA

Desde la antigüedad el hombre siempre ha buscado la forma de diseñar mecanismos inteligentes que de una u otra forma puedan influir en el desarrollo y comportamiento del ser humano, los egipcios y los griegos desarrollaron sistemas manuales o hidráulicos para mover parcialmente extremidades de sus ídolos “dioses”[1], controlados mecánicamente para fascinar a los adoradores de los templos, posteriormente los hombres buscaron la forma de desarrollar máquinas encargadas de hacer el trabajo repetitivo y desgastante del ser humano; la palabra robot tiene su antecedente a partir de la palabra checa robot , que significa servidumbre o trabajo forzado, establecido por el dramaturgo Karl Capek en 1921 en su obra dramática “Rossum`s Universal robots/R.U.R “[2], , sin embargo se debe al célebre escritor de

¹ <https://robotica.wordpress.com/about/>

ciencia ficción Dr. Isaac Asimov a finales de la década de los años treinta quien establece el termino de robótica[3], como la ciencia encargada del estudio de los robots, posteriormente el termino de robótica se extiende y se establece como el campo encargado del estudio de la tecnología robotizada que considera aspectos de: diseño, manufactura y aplicación, combinando para su desarrollo disciplinas como mecánica, matemáticas, control, electrónica, computación, etc., actualmente la robótica se encuentra inmersa dentro del concepto denominado “ Mecatrónica”, sin embargo debemos de considerar que el desarrollo de máquinas que asemejan las funciones del ser humano no deben de limitarse solo al aspecto físico, sino también, considerar la relación con la programación e incorporación de inteligencia artificial, tal es el caso de los ro-BOTS , entidades artificiales programadas que pueden desarrollarse dentro de un ambiente físico o virtual . Aunque aún estamos lejos de poder desarrollar BOTS que puedan mantener una conversación lógica humana, el avance ha sido considerable y el límite, controlado por el test de Turing [4], que con lleva al interlocutor a no asegurar si mantiene una conversación con un ser humano o bien con un programa informático. En la Figura 3 por ejemplo se muestran algunas de las estructuras del diseño de androides hasta llegar a la última generación de asimov.

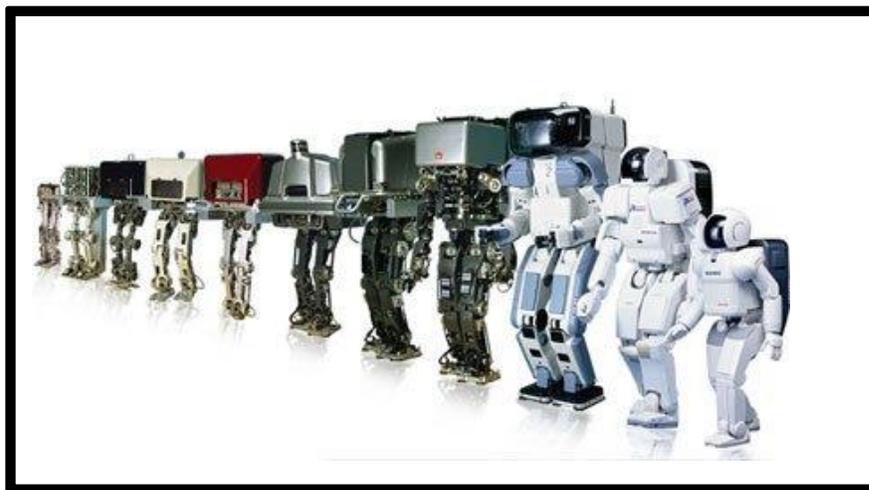


Figura 3. Estructuras del diseño de androides

1.3 ROBOTS EXPLORADORES

Los robots exploradores son aquellos que se han diseñado para conocer y explorar un lugar en concreto con el fin de evitar poner en peligro la vida de personas o conseguir acceso a entornos en los que el hombre no llega. Este tipo de robots cuentan con cámaras integradas para capturar imágenes que posteriormente son retransmitidas para su posterior análisis.

Como se puede observar en la Figura 4. Aunque los robots exploradores pueden ser controlados de forma remota, también existen unidades capaces de tomar decisiones gracias a que cuentan con inteligencia artificial.²

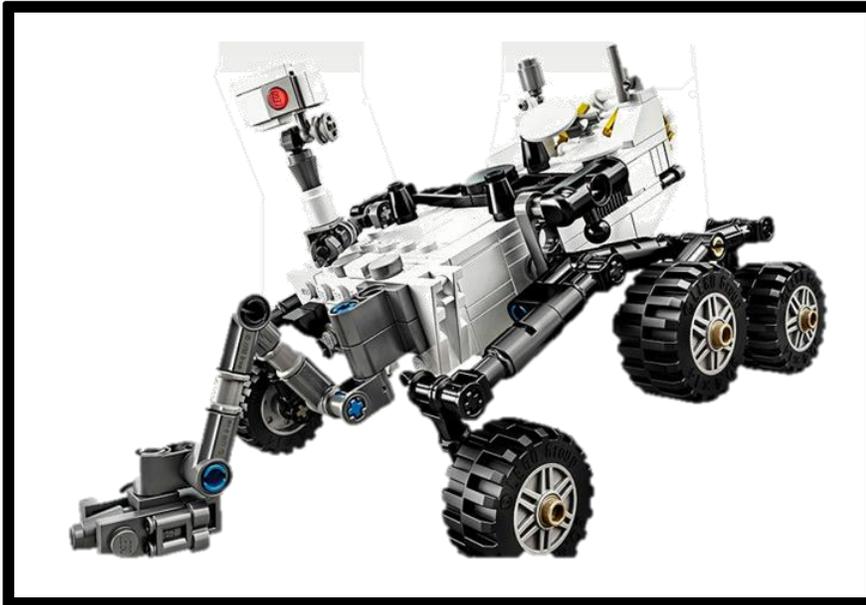


Figura 4. Robot de Exploración Espacial

El campo de los robots exploradores es muy grande actualmente algunos ejemplos son:

- Exploración en minas
- exploración espacial
- exploración militar
- exploración arqueológica

² Robots Exploradores. <http://www.robotikka.com/category/robots-exploradores/>

1.3.1 EXPLORACIÓN DE MINAS

La actividad minera está caracterizada por una exploración de alto riesgo, debido a que la persona que se dedican a la exploración de minerales que trabajan en lugares subterráneos. La oscuridad dentro de las minas, y la complejidad de las estructuras geológicas y geotécnicas sumadas al peligro de accidentes como caída de rocas, derrumbes, etc., hacen que las condiciones del trabajo sean extremadamente peligrosas.³ En la Figura 5 se muestra un robot de exploración a punto de ser ingresado a una cámara estrecha con el fin de entrar a un lugar subterráneo.

En un caso de derrumbe o inspección, donde es arriesgado enviar a una persona, se plantea la utilización de robots (Figura 5). Esta idea no es nueva y tiene antecedentes en su uso cada vez más constante en la industria, exploraciones espaciales, militares, etc.

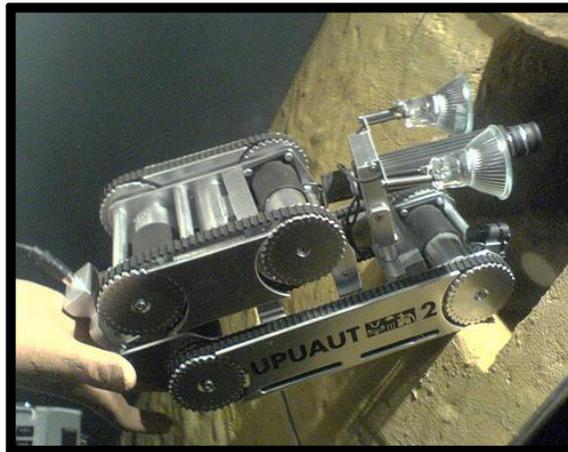


Figura 5. Robot de Exploración

³Robots en el uso de la minería. <http://djanho.blogspot.mx/2009/01/robot-movil-de-exploracion-y.html>

1.3.2 ROBOT DE EXPLORACIÓN ESPACIAL

La Mars Science Laboratory (abreviada MSL), conocida como Curiosity, del inglés 'curiosidad', es una misión espacial que incluye un móvil de exploración marciana dirigida por la NASA. La misión en si consiste en situar sobre la superficie marciana un móvil explorador, que lleva sobre su plataforma instrumentos científicos más avanzados para realizar un examen exhaustivo sobre el planeta, Curiosity Figura 6 es un vehículo tres veces más pesado y dos veces más grande que los vehículos utilizados en la misión Mars Exploration Rover, que aterrizaron en el año 2004.

El robot curiosidad fue lanzado mediante un cohete (Atlas V 541). Con el objeto de tomar muestras de suelo y polvo rocoso marciano para su análisis. La duración prevista de la misión es de 1 año marciano (1,88 años terrestres), con un radio de exploración mayor a los de los vehículos enviados anteriormente, e investigará la capacidad pasada y presente de Marte para alojar vida.⁴

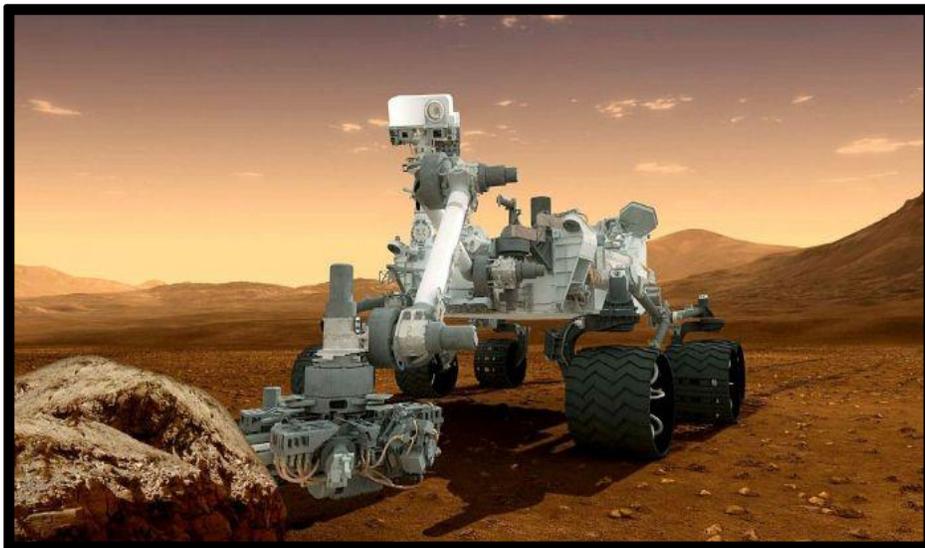


Figura 6. Robot Curiosity

⁴ Robot de Exploración espacial Curiosity <http://es.wikipedia.org/wiki/Curiosity>

1.3.3 ROBOT DE EXPLORACIÓN MILITAR

El iRobot 110 FirstLook, es un robot de exploración militar fabricado por la empresa que diseñó la aspiradora autónoma Roomba.

Una forma de poder acercarse y manipular un artefacto que puede estallar sin necesidad de exponer a una persona es a través de los robots terrestres no tripulados según iRobot, en todo el mundo se han suministrado miles y miles de UGV "que es" a fuerzas civiles y militares de todo el mundo con el objetivo de ayudar en las tareas de búsqueda, reconocimiento, desactivación de explosivos y todo tipo de misiones que pondrían la integridad de una persona en peligro. Entre ellos el iRobot 110 FirstLook.⁵

En cuanto a su funcionamiento, el mando permite maniobrar casi en cualquier situación el vehículo a una distancia considerable. Incorpora una pequeña pantalla (Figura 7) de 5 pulgadas para poder ver en todos los ángulos, gracias a las cuatro cámaras infrarrojas del iRobot 110 FirstLook.



Figura 7. Robot de exploración militar

⁵ <http://hipertextual.com/archivo/2014/03/irobot-110-firstlook/>

1.3.4 ROBOT DE EXPLORACIÓN ARQUEOLÓGICA

Una de las áreas en la cual los robots exploradores se desempeñan y utilizan más frecuentemente es en el área de investigación arqueológica.

Por ejemplo el robot explorador Tláloc II Figura 8, diseñado y construido por Ingenieros del Instituto Politécnico Nacional que ha aportado información importante para los arqueólogos a través de su sistema de transmisión de datos. Este robot a parte de sus cámaras de monitoreo cuenta con una tracción de orugas que permite una mayor movilidad en áreas rocosas.



Figura 8. Robot Tláloc II usado en exploraciones arqueológicas del Templo de la Serpiente Emplumada en México.

1.4 CONTROL POR RADIOFRECUENCIA

Una de las maneras más prácticas de trabajar con un robot a puertas cerradas y/o en espacios limitados, es programar sus funciones en una computadora y comandar el robot de manera remota por medio de algún enlace. Los datos recogidos por los sensores del robot llegan a la computadora central para que a través de ella se pueda entonces manipular los movimientos del robot. Para nuestro caso el enlace será a través de un sistema de radiofrecuencia. (Otra frecuencia, si hablamos de RF, porque la interconexión se puede hacer también por un enlace de infrarrojos). Para comunicarse por medio de un enlace de RF con un robot es posible usar receptores integrados muy pequeños y de bajo costo (Figura 9), como el RWS-433, o el RXLC-434, y otros similares, que trabajan en frecuencias de entre 303 y 433 MHz. La elección de los transmisores dependerá de la distancia entre receptor y emisor, pero si se trata de áreas acotadas es posible utilizar los módulos transmisores hermanados con los anteriores, como el TWS-433 y el TXLC-434.⁶

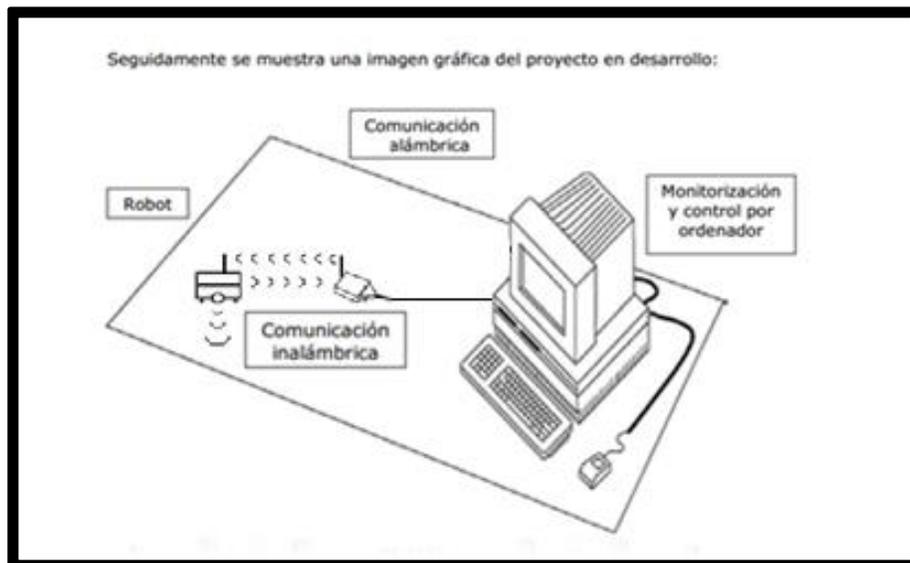
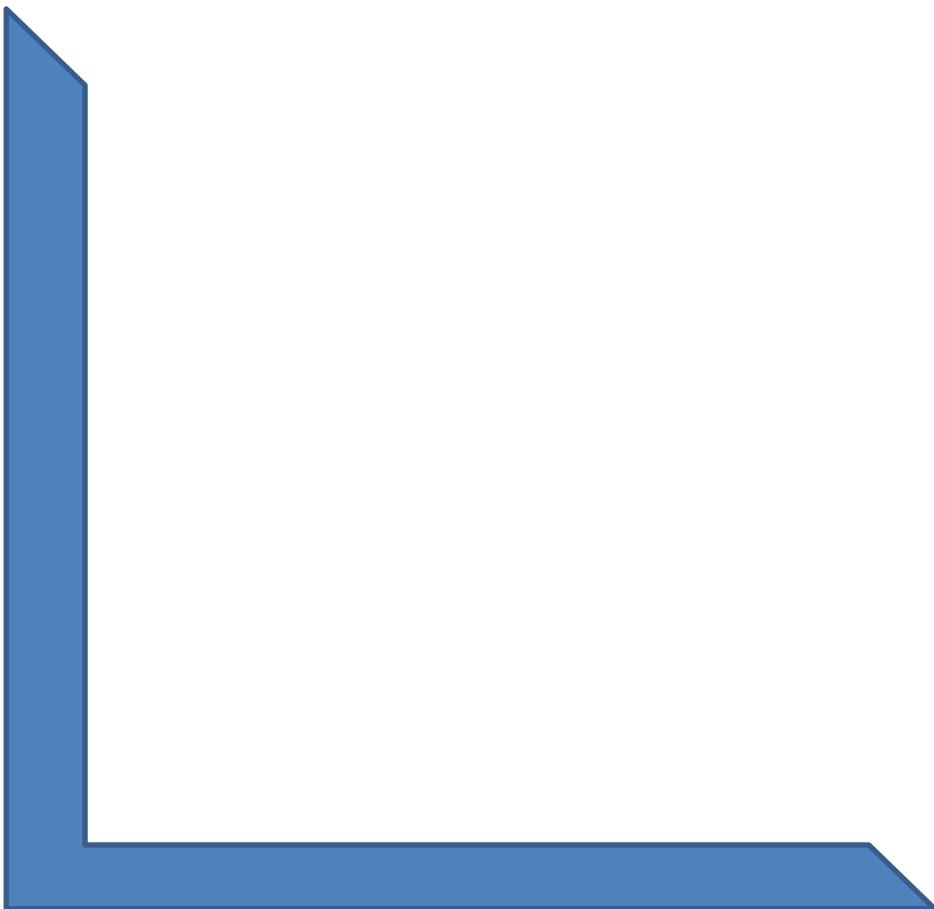


Figura 9. Chip transmisor y receptor para comunicaciones inalámbricas

⁶ Comunicación Radiofrecuencia

CAPÍTULO 2



Capítulo 2

INTEGRACIÓN DE MODULOS

CONCEPTOS Y DEFINICIONES BASICAS

INTRODUCCIÓN

Cuando uno enciende un aparato de radio se pueden sintonizar diversos canales “comúnmente conocidos como estaciones”, las cuales transmiten señales “información”, platicas, canciones, entrevistas, concursos, etc. Figura 10. La selección de la señal de radio entonces, depende del usuario y de lo que se desea escuchar, cada estación a su vez se encuentra vinculada con la frecuencia en la que se envía la información “señal”.

La radiofrecuencia no solamente se utiliza para obtener o enviar un señal en sus diversas modalidades, también se utiliza para manipular o accionar a distancia diversos sistemas, por ejemplo el movimiento de un robot, el movimiento de una cámara de seguridad, el analizar señales provenientes del espacio —radio astronomía—, el monitoreo de la calidad del aire, etc. Este capítulo detalla el concepto de la radiofrecuencia y sus diferentes usos, considerando los avances tecnológicos actuales.

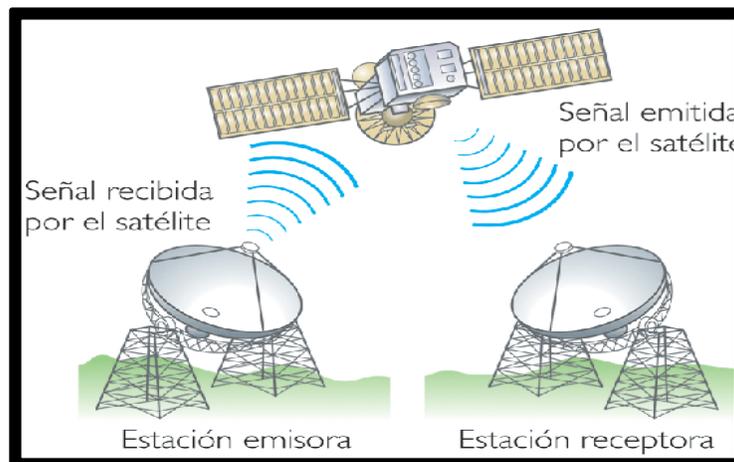


Figura 10. Sistema de comunicación de radiofrecuencia

2.1 ¿QUÉ ES LA COMUNICACIÓN?

La comunicación es el proceso mediante el cual se puede transmitir información de un punto a otro punto generalmente se requieren de un emisor, el mensaje y el receptor.

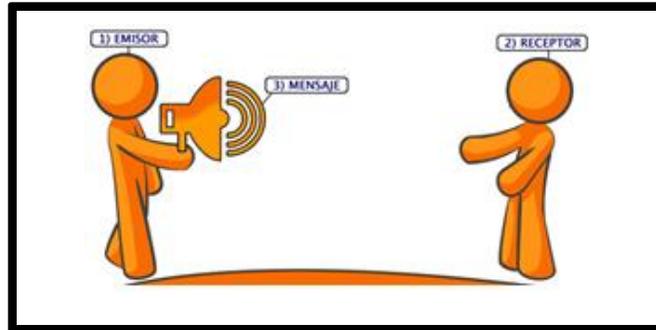


Figura 11. Sistema de Comunicación

En la figura 11. Se observa como la señal que proviene del emisor es canalizada a través de un medio de comunicación, en nuestro caso el aire y recibida por el receptor.

2.1.1 ¿QUÉ ES LA COMUNICACIÓN A DISTANCIA?

Es aquella que se da por medios alámbricos o inalámbricos, la comunicación alámbrica (por cable) tiene lugar a través de líneas o cables como se puede apreciar en la Figura 12, que unen al emisor y receptor, la información se transmite mediante impulsos eléctricos.

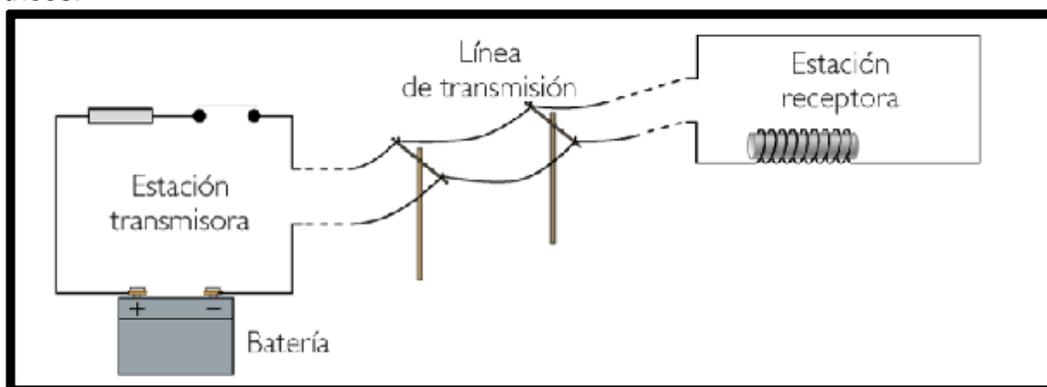
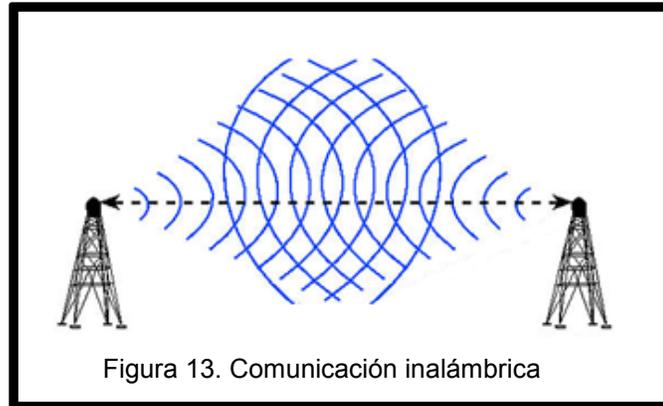


Figura 12. Sistema de comunicación alámbrica.

La comunicación inalámbrica es aquella en la cual la información se transmite mediante ondas de radio. Al igual que la comunicación alámbrica la inalámbrica también consta de un emisor y un receptor el ejemplo se muestra en la Figura 13.



Las ondas que se propagan a través del aire se denominan ondas electromagnéticas o también llamadas ondas de radio. En consecuencia una onda electromagnética es aquella que no necesita un medio material para propagarse incluyendo la luz visible y **se propagan en el vacío a una velocidad constante mediante oscilaciones⁷** de campos eléctricos y magnéticos, (Figura 14).

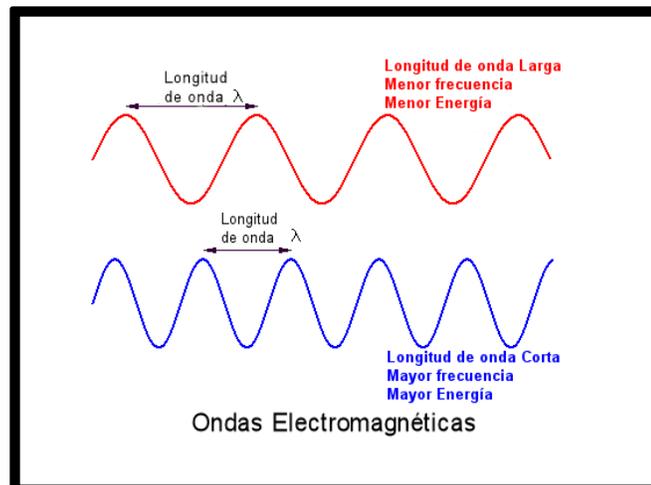


Figura 14. Longitud de ondas

⁷ Perturbación o fluctuación en el tiempo de un medio o sistema.

2.1.2 ¿QUÉ ES LA RADIO FRECUENCIA?

La radiofrecuencia se conoce como el espectro de energía electromagnético que se utiliza en las radiocomunicaciones y que generalmente abarca de los 3 kHz hasta los 300 GHz y sus aplicaciones más relevantes se encuentran en comunicaciones militares, navegación, radares y la radiofonía AM y FM, (Figura 15).

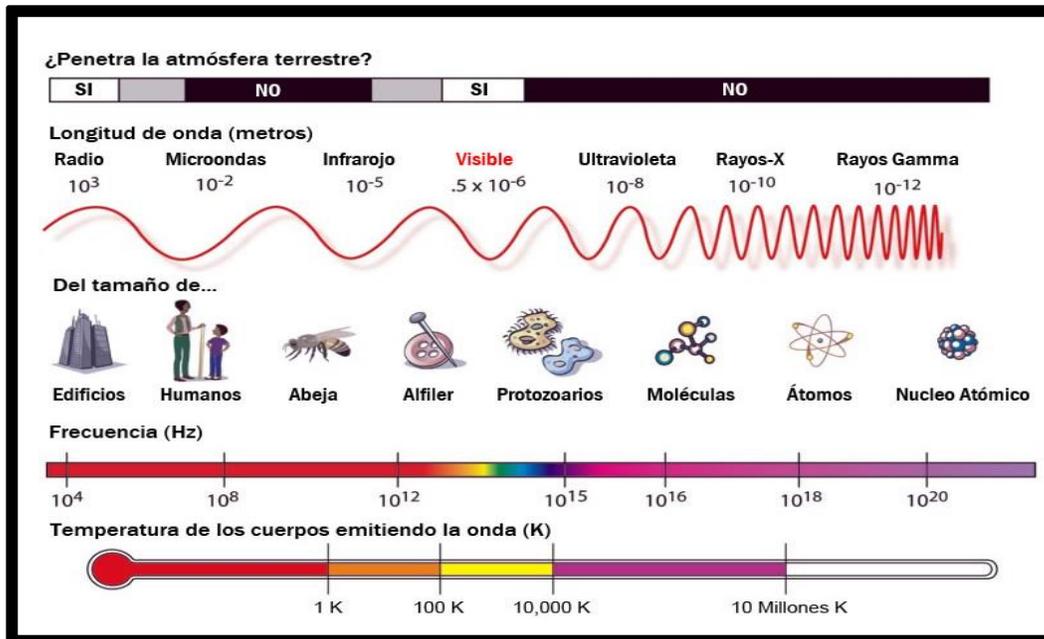


Figura 15. Espectro electromagnético

El espectro electromagnético como se observa en la Figura 15, se refiere a como se distribuye la energía de las ondas electromagnéticas y su longitud de onda, este espectro queda determinado por las radiaciones que se emiten. Las ondas por su parte suponen la propagación de esta radiación acarreado energía.

2.1.3 FRECUENCIAS DE OPERACIÓN

El espectro electromagnético es un conjunto de ondas electromagnéticas, también se le conoce como radiación electromagnética que sirve para identificar una sustancia de manera análoga⁸, estos espectros se pueden observar mediante espectroscopios que permiten realizar medidas como son las longitudes de onda, las frecuencias y la intensidad de la radiación.

El término radiofrecuencia RF, también llamado espectro de radiofrecuencia se aplica a la porción menos energética del espectro electromagnético, el cual se sitúa entre 3 kilo Hertz (3 kHz) y unos 300 Giga Hertz (300GHz), (Tabla 1).

Como se sabe el término Hertz se encuentra asociado a la frecuencia, es la unidad de medida de la frecuencia de las ondas, y corresponde a un ciclo por segundo Figura 16.

$$f = \frac{1}{T_{\text{seg}}}$$
$$\lambda = \frac{c}{\gamma}$$

c = velocidad de la luz en el vacío
λ = longitud de onda
γ = frecuencia

Figura 16. Fórmula para calcular la distancia y velocidad de la frecuencia de onda

⁸ Tipo de señal generada por algún tipo de fenómeno electromagnético representable por una función matemática continua.

Clasificación de la Radiofrecuencia

La radiofrecuencia se encuentra clasificada según su utilización, en la tabla 1 se muestra el rango de operación de cada uno de ellas. Por ejemplo la ultra baja frecuencia opera en frecuencias que oscilan entre 300 y 3000 Hz. y tiene su aplicación para comunicaciones en minas por su capacidad de penetrar fácilmente la superficie terrestre.

| Nombre | Abreviatura inglesa | Banda ITU | Frecuencias | Longitud de onda |
|-----------------------|---------------------|-----------|---------------------------------|------------------------------|
| | | | Inferior a 3 Hz | > 100.000 km |
| Extra baja frecuencia | ELF | 1 | 3-30 Hz | 100.000–10.000 km |
| Super baja frecuencia | SLF | 2 | 30-300 Hz | 10.000–1000 km |
| Ultra baja frecuencia | ULF | 3 | 300–3000 Hz | 1000–100 km |
| Muy baja frecuencia | VLF | 4 | 3–30 kHz | 100–10 km |
| Baja frecuencia | LF | 5 | 30–300 kHz | 10–1 km |
| Media frecuencia | MF | 6 | 300–3000 kHz | 1 km – 100 m |
| Alta frecuencia | HF | 7 | 3–30 MHz | 100–10 m |
| Muy alta frecuencia | VHF | 8 | 30–300 MHz | 10–1 m |
| Ultra alta frecuencia | UHF | 9 | 300–3000 MHz | 1 m – 100 mm |
| Super alta frecuencia | SHF | 10 | 3-30 GHz | 100-10 mm |
| Extra alta frecuencia | EHF | 11 | 30-300 GHz | 10–1 mm |
| | | | Por encima de los 300 GHz | < 1 mm |

Tabla 1. Espectro de frecuencia y Longitudes de onda. [3]

2.1.4 CONTROL MEDIANTE SISTEMAS DE RADIOFRECUENCIA

El concepto de control por radiofrecuencia se puede observar en la figura 17 en ella, se muestra como el transmisor ubicado en la computadora envía la información “señal” al receptor ubicado en la plataforma de robot móvil, para ser manipulado a distancia mediante el envío de órdenes “señales” enviadas desde el transmisor.

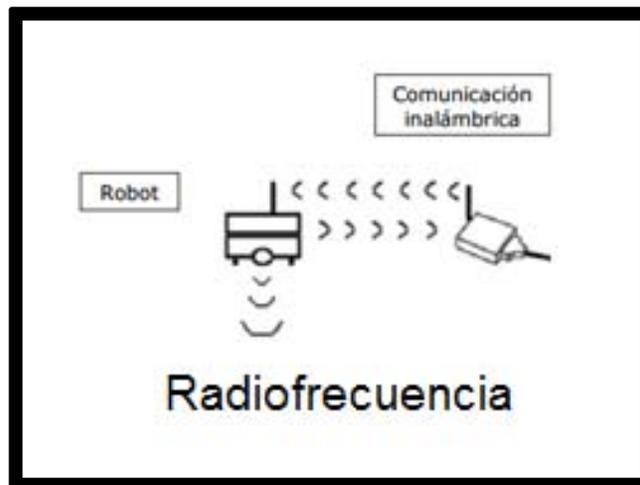
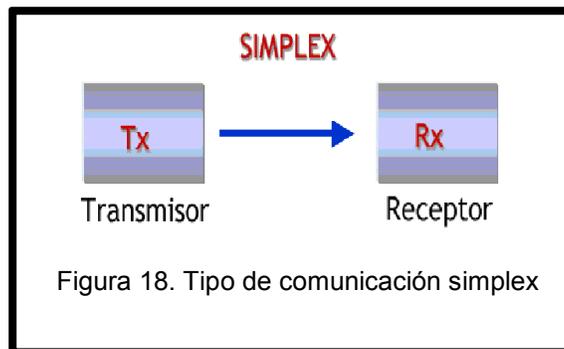


Figura 17. Sistema típico de comunicación por radiofrecuencia.

2.1.5 TIPOS DE COMUNICACIÓN EN SERIE

Simplex: En este caso el transmisor (Tx) y el receptor (Rx) como se puede apreciar en la Figura 18, están perfectamente definidos y la comunicación es unidireccional. Este tipo de comunicaciones se emplean, usualmente, en redes de radiodifusión, donde los receptores no necesitan enviar ningún tipo de dato al transmisor.



Dúplex, halfdúplex o semi-dúplex: En este caso ambos extremos del sistema de comunicación cumplen funciones de transmisor y receptor (Tx/Rx), y los datos se desplazan en ambos sentidos pero no de manera simultánea (Figura 19). Este tipo de comunicación se utiliza habitualmente en la interacción entre terminales y una computadora central.



Full Dúplex: El sistema es similar al dúplex, pero los datos se desplazan en ambos sentidos simultáneamente como se muestra en la Figura 20. Para que sea posible ambos emisores poseen diferentes frecuencias de transmisión o dos caminos de comunicación separados, mientras que la comunicación semi-duplex necesita normalmente uno solo.

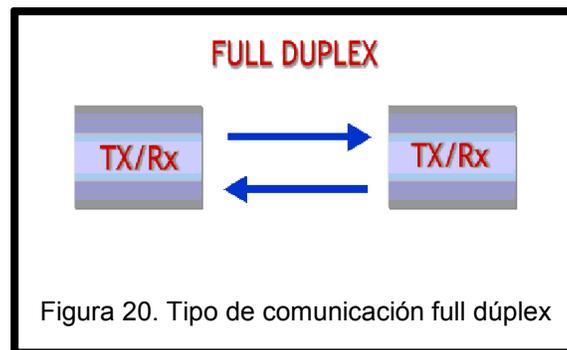


Figura 20. Tipo de comunicación full dúplex

2.2 ROBÓTICA MÓVIL

La tecnología robotizada ha venido sustituyendo aplicaciones en donde la tarea es desgastante o repetitiva para el ser humano. El monitorear o explorar es una de esas actividades que requieren de un desgaste físico muy grande, sobre todo cuando estas tareas de investigación se realizan en zonas de no fácil acceso.

La función de un robot móvil en esencia es desplazarse de un punto a otro y realizar tareas básicas: grabación, transportación, manipulación..etc.

Las aplicaciones de la robótica móvil son infinitas (vea Figura 21), por ejemplo:

- Entretenimiento y ocio
- Autómatas en una fábrica o cadena de producción
- Robots desactivadores de bombas de la policía
- Robots de exploración espacial.



Figura 21. Aplicaciones en la Robótica.

2.2.1 LOCOMOCIÓN

Existe una gran variedad de modos o formas para moverse sobre una superficie sólida las más comunes son mediante la utilización de ruedas, cadenas, patas..etc. (Figura 22). Los cuales permiten una cierta movilidad según sean las necesidades o requerimientos de la tarea.



Figura 22. Locomoción tipo Ruedas.

Algunas características sobre estos tipos son:

Ruedas: Los vehículos de ruedas (Figura 22) son por mucho los más populares por varias razones prácticas, son más sencillos y más fáciles de construir, la carga que pueden transportar es relativamente mayor al peso del robot.

La principal desventaja de las ruedas es su empleo en terreno irregular, en el que se comportan bastante mal. Normalmente un vehículo de ruedas podrá sobrepasar un obstáculo que tenga una altura no superior al radio de sus ruedas, entonces una solución es utilizar ruedas mayores que los posibles obstáculos a superar; sin embargo, esta solución, a veces, puede no ser práctica.

Cadenas: Los robots basados en cadenas (Figura 23) se pueden considerar más pesados, sin embargo es una opción muy buena porque las cadenas permiten al robot superar obstáculos relativamente mayores y son menos susceptibles a uso de ruedas

que pueden sufrir daños por el entorno, como piedras o arena. El principal inconveniente de las cadenas es su ineficacia, puesto que se produce deslizamiento sobre el terreno al avanzar y al girar. Si la navegación se basa en el conocimiento del punto en que se encuentra el robot y el cálculo de posiciones futuras sin error, entonces las cadenas acumulan tal cantidad de error que hace inviable la navegación por este sistema.



Figura 23. Locomoción tipo cadenas o skidsteer

Patas: Potencialmente los robots con patas pueden superar con mayor facilidad los problemas de los terrenos irregulares. A pesar de que hay un gran interés en diseñar este tipo de robots, su construcción plantea numerosos retos. Estos retos se originan principalmente en el gran número de grados de libertad que requieren los sistemas con patas. Cada pata necesita como mínimo un par de motores lo que produce un mayor coste, así como una mayor complejidad y menor fiabilidad y los algoritmos de control se vuelven mucho más complicados por el gran número de movimientos a coordinar, los sistemas de patas son un área de investigación muy activo.

2.2.2 DISEÑO DE RUEDAS

Existen varias plataformas de ruedas para elegir cuando se quiere construir un robot móvil, algunas son diferencial, triciclo, Ackerman. (Figura 24).

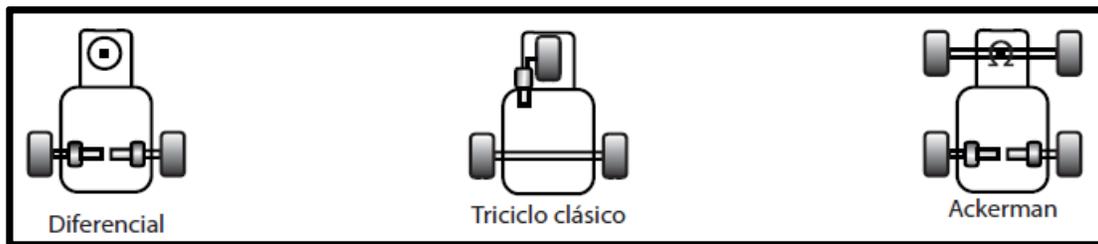


Figura 24. Diseño de ruedas.

2.2.3 DIFERENCIAL

Tanto desde el punto de vista de la programación como de la construcción, el diseño diferencial (figura 25) es uno de los menos complicados sistemas de locomoción. El robot puede ir recto, girar sobre sí mismo y trazar curvas.

Un problema importante es cómo resolver el equilibrio del robot, hay que buscarle un apoyo adicional a las dos ruedas ya existentes, esto se consigue mediante una o dos ruedas de apoyo añadidas en un diseño triangular o romboidal. El diseño triangular puede no ser suficiente dependiendo de la distribución de pesos del robot, y el romboidal puede provocar inadaptación al terreno si éste es irregular lo que puede exigir alguna clase de suspensión.

Otra consideración a hacer en este diseño es cómo conseguir que el robot se mueva recto, para que el robot se mueva en línea recta sus ruedas tienen que girar a la misma velocidad.

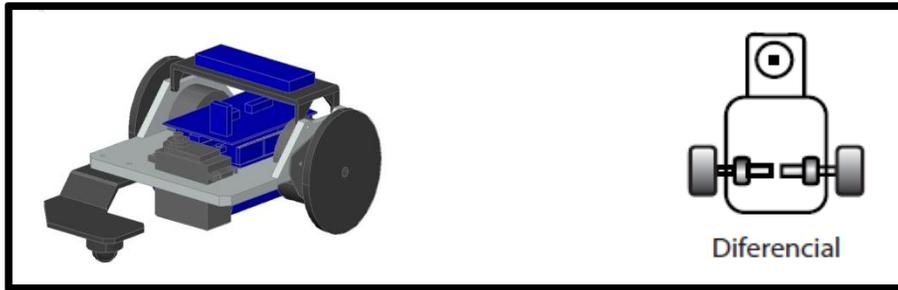


Figura 25. Diseño Diferencial.

Cuando los motores encuentran diferentes resistencias (una rueda sobre moqueta y la otra sobre terrazo) las velocidades de los motores varían y el robot girará incluso aun cuando se le haya ajustado inicialmente para que vaya recto. Esto quiere decir que la velocidad debe ser controlada dinámicamente, o sea, debe existir un medio de monitorizar y cambiar la velocidad del motor mientras el robot avanza. De esta manera la simplicidad del diseño queda minimizada por la complejidad del sistema de control de la velocidad; no obstante la reducción de la complejidad mecánica en detrimento⁹ de la complejidad de la electrónica y del software es frecuentemente una elección más barata y fiable.

2.2.4 DISEÑO SINCRONIZADO

En este diseño todas las ruedas (generalmente tres) son tanto de dirección como motrices, las ruedas están enclavadas de tal forma que siempre apuntan en la misma dirección. Para cambiar de dirección el robot gira simultáneamente todas sus ruedas alrededor de un eje vertical, de modo que la dirección del robot cambia, pero su chasis sigue apuntando en la misma dirección que tenía. Si el robot tiene una parte delantera (es asimétrico) presumiblemente donde se concentran sus sensores, se tendrá que arbitrar un procedimiento para que su cuerpo se oriente en la misma dirección que sus ruedas. El diseño sincronizado supera muchas de las dificultades que plantean el diseño diferencial, en triciclo y de coche, pero a costa de una mayor complejidad mecánica.

⁹ Destrucción leve o parcial de algo.

2.2.5 DISEÑO DE TRICICLO Y COCHE

El diseño de coche con sus cuatro ruedas con suspensión proporciona una buena estabilidad, el diseño en triciclo (figura 26) tiene unas prestaciones similares con la ventaja de ser mecánicamente más simple ya que el coche necesita alguna unión entre las ruedas direccionales. En general en estos dos diseños las ruedas direccionales no son motrices, y no es necesario controlar la velocidad de las ruedas para que el robot se mantenga recto. Esta simplificación tiene su precio como veremos en el próximo apartado.

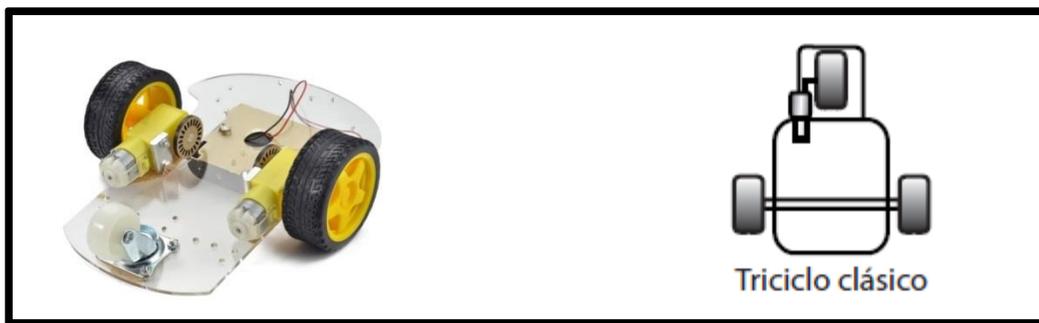


Figura 26. Diseño de Triciclo.

2.2.6 CINEMÁTICA

La cinemática del robot trata de cómo se mueve el mismo, dado que la dirección adopta tal y tal ángulo y que cada rueda gira tantas veces, ¿dónde acabará el robot y qué camino tomará?

Los diseños diferencial y sincronizado tienen una ventaja sutil sobre los otros dos tipos, la diferencia estriba en sus cinemáticas. Considere un triciclo, el cual tiene tres grados de libertad cuando se mueve sobre una superficie plana. Es decir, en relación con un sistema global de coordenadas, el robot puede estar en cualquier posición especificado por dos coordenadas x e y , y apuntando en una dirección especificada por una tercera coordenada, el ángulo q . Estos tres grados de libertad (x,y,q) nos dan la distancia y el ángulo entre el sistema de coordenadas global, y una referencia local en el robot.

Nos gustaría tener la posibilidad de posicionar y orientar nuestro robot en cualquier lugar sobre el plano, es decir, sin considerar de donde arranca, si le damos x, y, θ el robot debe poder moverse a esa posición. Sin embargo, hay un problema, para alcanzar esos tres grados de libertad el robot sólo puede controlar dos parámetros: la dirección, ángulo θ , y la distancia total recorrida, S . Esto quiere decir que la orientación del robot y su posición están ligados, para girar tiene que moverse hacia delante o hacia atrás. El robot no puede ir directamente de una posición y orientación a otra, incluso aun cuando no haya nada en su camino. Para alcanzar una posición y orientación deseadas simultáneamente, el robot tiene que seguir algún camino, posiblemente complejo. Los detalles de ese camino se complican todavía más con la presencia de obstáculos, esta es la razón por la que el aparcamiento en línea es difícil; sin embargo, un robot basado en los diseños diferencial o sincronizado puede, al girar sobre sí mismo, desacoplar efectivamente su posición de su orientación.

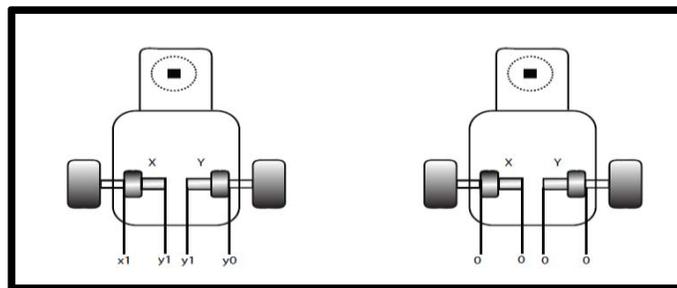


Figura 27. Cinemática de Ángulos del Robot.

2.2.7 FORMA DEL ROBOT

La forma de un robot puede tener un gran impacto en sus prestaciones, un robot no cilíndrico corre mayor riesgo de quedar atrapado por una disposición desfavorable de obstáculos o de fallar en encontrar un camino en un espacio estrecho o intrincado.

Consideremos dos robots del mismo tamaño uno cilíndrico (figura 28) y el otro cuadrado (figura 29), ambos encuentran un paso estrecho según se mueven. Un algoritmo sencillo permitirá al robot cilíndrico pasar, el robot choca, gira y lo intenta de nuevo hasta que pasa. Esto es así de simple porque el robot es capaz de girar estando en contacto con el obstáculo.

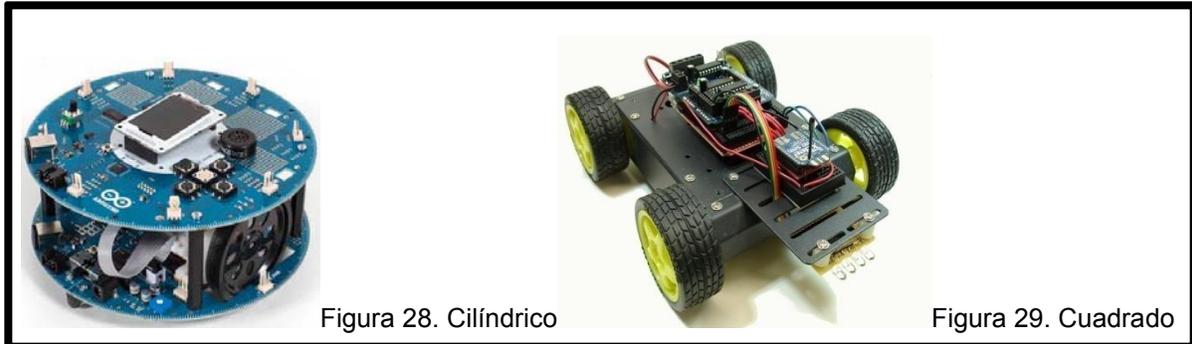


Figura 28. Cilíndrico

Figura 29. Cuadrado

El robot cuadrado, por el contrario, tiene que retroceder y girar si quiere usar la misma táctica. Por tanto, siempre se requiere un algoritmo más complejo para la navegación de un robot cuadrado que para la de uno cilíndrico. Para entender la razón de esto, tenemos que apelar a un concepto avanzado en robótica conocido como *espacio de configuraciones*.

2.2.8 ADAPTACIÓN DE PLATAFORMAS MÓVILES

Hay una gran cantidad de plataformas móviles disponibles para adaptarlas como base de robots móviles: coches de radio-control, vehículos filo dirigidos y otros juguetes de pilas. La mayoría de los sistemas de locomoción de ruedas excepto el sincronizado están bien representados en las tiendas de juguetes.

Es una buena elección escoger el sistema de locomoción y suspensión de un juguete como base de un robot móvil por varias razones. Lo primero es que nos exigirá menos esfuerzo de diseño y construcción ya que la mayor parte de estos problemas los habrá resuelto de antemano el fabricante, y lo segundo que resultará más barato que comprar los componentes por separado.

El diseñador del robot, sin embargo, debe tener presente que existen algunos problemas al hacer esto. Normalmente la plataforma no se adaptará directamente a su uso como robot. Los motores de los juguetes requieren más corriente y tienen poca eficacia, lo que significa que la electrónica será más complicada y el tiempo de funcionamiento será más corto.

Por lo general, los motores y engranajes de los juguetes se diseñan para hacer al juguete más rápido, por tanto, se producen problemas de control cuando el robot debe moverse despacio para responder a los sensores, también es complicado añadirles codificadores de eje. Lo ideal sería encontrar un modelo que tuviera un motor y transmisión independiente a cada rueda.

Los juguetes móviles más baratos tienen un solo motor (figura 30) y maniobran mediante una serie de movimientos adelante, atrás y giros. Cuando el motor gira en una dirección el juguete se mueve adelante, y cuando el motor gira al revés un embrague simple montado en el eje hace que sólo gire una de las ruedas y por tanto el juguete gira. Por tanto, el juguete sólo gira cuando va hacia atrás. Es posible realizar un robot que actúe de esa manera, aunque se puede quedar “pegado” cuando no puede ir hacia atrás. Los juguetes que llevan este sistema de movimiento se reconocen con facilidad porque su control remoto sólo tiene un botón. Cuando el juguete se enciende, anda solamente adelante y cuando se pulsa el botón gira al ir atrás.

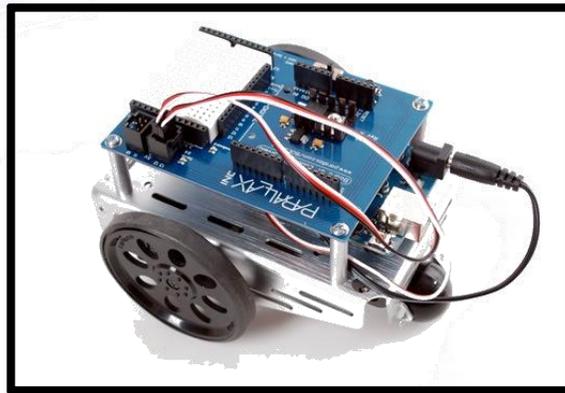


Figura 30. Diseño de Proyecto con un solo motor.

Los juguetes que son más útiles tienen transmisión con diferencial o cadenas o un motor de tracción separado del motor de direccionamiento.

2.3 ¿QUÉ ES UNA CÁMARA IP?

Una Cámara IP (también conocidas como cámaras de Red) son videocámaras especialmente diseñadas para enviar las señales (video, y en algunos casos audio) a través de Internet desde un explorador (por ejemplo el Internet Explorer) o a través de concentrador (un HUB o un SWITCH) en una Red Local (LAN)¹⁰

En las cámaras IP pueden integrarse aplicaciones como detección de presencia (incluso el envío de mail si detectan presencia), grabación de imágenes o secuencias en equipos informáticos (tanto en una red local o en una red externa (WAN), de manera que se pueda comprobar por qué ha saltado la detección de presencia y se graben imágenes de lo sucedido.

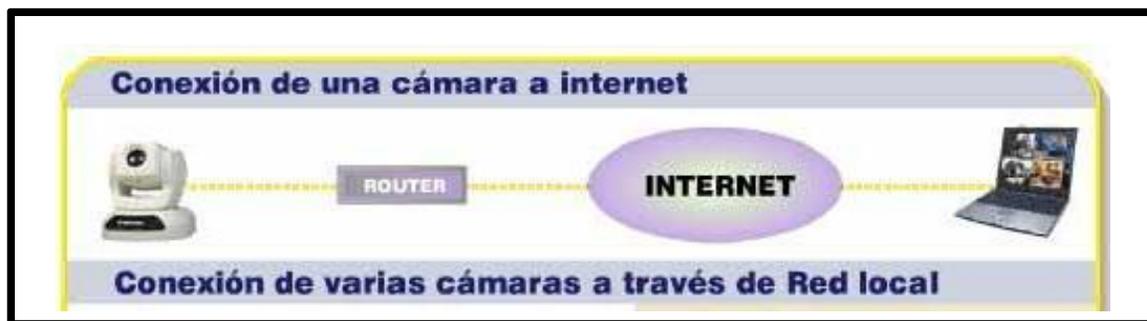


Figura 31. Descripción del funcionamiento de una cámara IP

¹⁰ Cámaras ip descripción y funcionamiento

2.3.1 CARACTERÍSTICAS DE UNA CÁMARA IP

Las cámaras IP (Figura 32) permiten ver en tiempo real qué está pasando en un lugar, aunque esté a miles de kilómetros de distancia. Son cámaras de vídeo de gran calidad que tienen incluido un ordenador a través del que se conectan directamente a Internet.

Una cámara IP (o una cámara de red) es un dispositivo que contiene:

- Una cámara de vídeo de gran calidad, que capta las imágenes
- Un chip de compresión que prepara las imágenes para ser transmitidas por Internet, y
- Un ordenador que se conecta por sí mismo a Internet



Figura 32. Cámara IP

2.4 MICROPROCESADORES

En los años 1970, la electrónica digital no estaba suficientemente desarrollada, pero dentro de la electrónica ya era una especialidad consagrada. En aquel entonces las computadoras se diseñaban para que realizaran algunas operaciones muy simples, y si se quería que estas máquinas pudiesen hacer cosas diferentes, era necesario realizar cambios bastante significativos al hardware.

A principios de la década de 1970, una empresa japonesa le encargó a una joven compañía norteamericana que desarrollara un conjunto de circuitos para producir una calculadora de bajo costo. Intel se dedicó de lleno a la tarea y entre los circuitos encargados desarrolló uno muy especial, algo no creado hasta la fecha: el primer microprocesador integrado.

El Intel 4004 salió al mercado en 1971, es una máquina digital sincrónica compleja, como cualquier otro circuito lógico secuencial sincrónico. Sin embargo, la ventaja de este componente está en que aloja internamente un conjunto de circuitos digitales que pueden hacer operaciones corrientes para el cálculo y procesamiento de datos, pero desde una óptica diferente: sus entradas son una serie de códigos bien definidos, que permiten hacer operaciones de carácter específico cuyo resultado está determinado por el tipo de operación y los operandos involucrados.

Visto así, no hay nada de especial en un microprocesador; la maravilla está en que la combinación adecuada de los códigos de entrada, su ejecución secuencial, el poder saltar hacia atrás o adelante en la secuencia de códigos en base a decisiones lógicas u órdenes

específicas, permite que la máquina realice un montón de operaciones complejas, no contempladas en los simples códigos básicos.

Hoy estamos acostumbrados a los sistemas con microprocesadores, pero en el lejano 1971 esta era una forma de pensar un poco diferente y hasta escandalosa, a tal punto que Busicom, la empresa que encargó los chips a Intel, no se mostró interesada en el invento, por lo que Intel lo comercializó para otros que mostraron interés; el resto es historia: una revolución sin precedentes en el avance tecnológico de la humanidad.

Es lógico pensar que el invento del microprocesador integrado no fue una revelación divina para sus creadores, sino que se sustentó en los avances, existentes hasta el momento, en el campo de la electrónica digital y las teorías sobre computación. Pero sin lugar a dudas fue la gota que colmó la copa de la revolución científico-técnica, porque permitió desarrollar aplicaciones impensadas o acelerar algunas ya encaminadas.

2.4.1 MICROCONTROLADORES

Hemos interactuado con los microcontroladores ya desde hace varios años debido a que estos son parte fundamental en el desarrollo de aplicaciones. Los automóviles, lavadoras, hornos microondas, teléfonos, etc. Son alguno de los ejemplos en donde los microcontroladores han sido relevantes. Gracias a que están diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. En el presente capítulo hablaremos acerca de lo que es un microcontrolador sus características, se dará una explicación sobre las diversas arquitecturas que lo conforman así también como sobre las familias de los microcontroladores Figura 33 y por ultimo una explicación sobre la elección del microcontrolador usado para este proyecto.

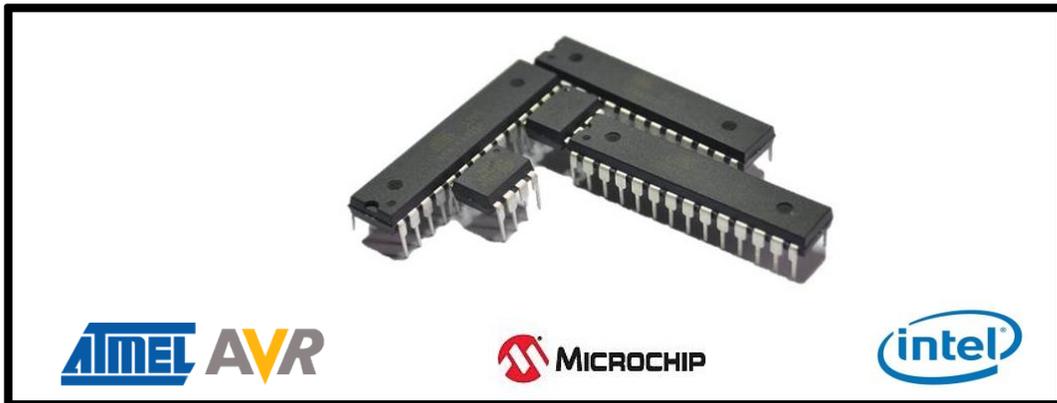


Figura 33. Tipos de microcontroladores más usados

2.4.2 ¿QUÉ ES UN MICROCONTROLADOR?

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia. Tienen la capacidad para mantener la funcionalidad a la espera de un evento como pulsar un botón u otra de interrupción. Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos.

Los microcontroladores son diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación. El control de un electrodoméstico sencillo como una batidora utilizará un procesador muy pequeño (4 u 8 bits) porque sustituirá a un autómata finito. En cambio, un reproductor de música y/o vídeo digital (MP3 o MP4) requerirá de un procesador de 32 bits o de 64 bits y de uno o más códecs¹¹ de señal digital (audio y/o vídeo). El control de un sistema de frenos ABS (Antilock Brake System) se basa normalmente en un microcontrolador de 16 bits, al igual que el sistema de control electrónico del motor en un automóvil.

¹¹ Transformar un archivo con un flujo de datos o una señal, codificar el flujo o señal y descifrarlo.

2.4.3 ARQUITECTURAS DE COMPUTADORA

Básicamente existen dos arquitecturas de computadoras, y por supuesto, están presentes en el mundo de los microcontroladores: Von Neumann y Harvard. Ambas se diferencian en la forma de conexión de la memoria al procesador y en los buses que cada una necesita.

2.4.3.1 ARQUITECTURA VON NEUMANN

La arquitectura Von Neumann (figura 34) utiliza el mismo dispositivo de almacenamiento “memoria principal” tanto para las instrucciones como para los datos, siendo la que se utiliza en un ordenador personal porque permite ahorrar una buena cantidad de líneas de E/S, que son bastante costosas, sobre todo para aquellos sistemas donde el procesador se monta en algún tipo de zócalo¹² alojado en una placa madre. También esta organización les ahorra a los diseñadores de placas madre una buena cantidad de problemas y reduce el costo de este tipo de sistemas.

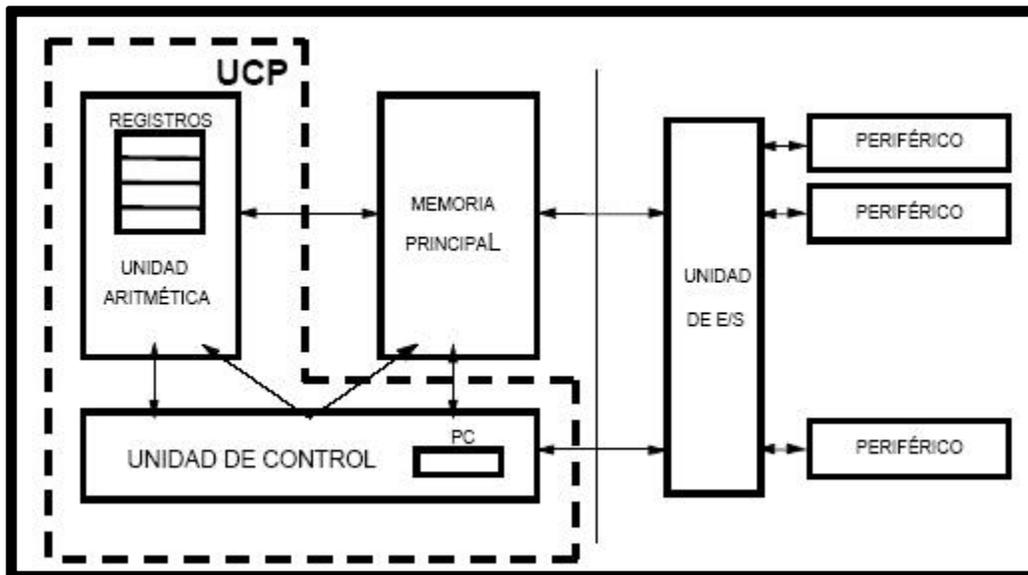


Figura 34. Arquitectura Von Neumann.

¹² Dispositivo para conectar circuitos integrados en circuitos impresos, sin realizar soldadura.

En un ordenador personal, cuando se carga un programa en memoria, a éste se le asigna un espacio de direcciones de la memoria que se divide en segmentos, de los cuales típicamente tendremos los siguientes: código (programa), datos y pila. Es por ello que podemos hablar de la memoria como un todo, aunque existan distintos dispositivos físicos en el sistema (disco duro, memoria RAM, memoria flash, unidad de disco óptico...).

En el caso de los microcontroladores, existen dos tipos de memoria bien definidas: memoria de datos (típicamente algún tipo de SRAM) y memoria de programas (ROM, PROM, EEPROM, flash u de otro tipo no volátil). En este caso la organización es distinta a las del ordenador personal, porque hay circuitos distintos para cada memoria y normalmente no se utilizan los registros de segmentos, sino que la memoria está segregada y el acceso a cada tipo de memoria depende de las instrucciones del procesador.

A pesar de que en los sistemas integrados con arquitectura Von Neumann la memoria esté segregada, y existan diferencias con respecto a la definición tradicional de esta arquitectura; los buses para acceder a ambos tipos de memoria son los mismos, del procesador solamente salen el bus de datos, el de direcciones, y el de control. Como conclusión, la arquitectura no ha sido alterada, porque la forma en que se conecta la memoria al procesador sigue el mismo principio definido en la arquitectura básica.

Algunas familias de microcontroladores como la Intel 8051 (figura 35) y la Z80 implementan este tipo de arquitectura, fundamentalmente porque era la utilizada cuando aparecieron los primeros microcontroladores.

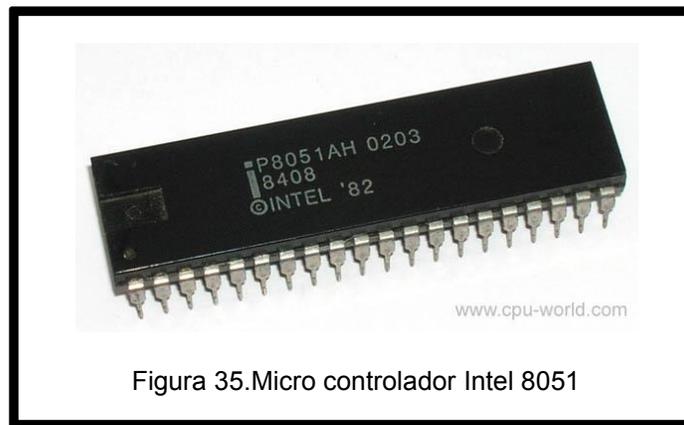


Figura 35. Micro controlador Intel 8051

2.4.3.2 ARQUITECTURA HARVARD

La otra variante es la arquitectura Harvard (figura 36), por excelencia es utilizada en supercomputadoras, en los microcontroladores, y sistemas integrados en general. En este caso, además de la memoria, el procesador tiene los buses segregados, de modo que cada tipo de memoria tiene un bus de datos, uno de direcciones y uno de control.

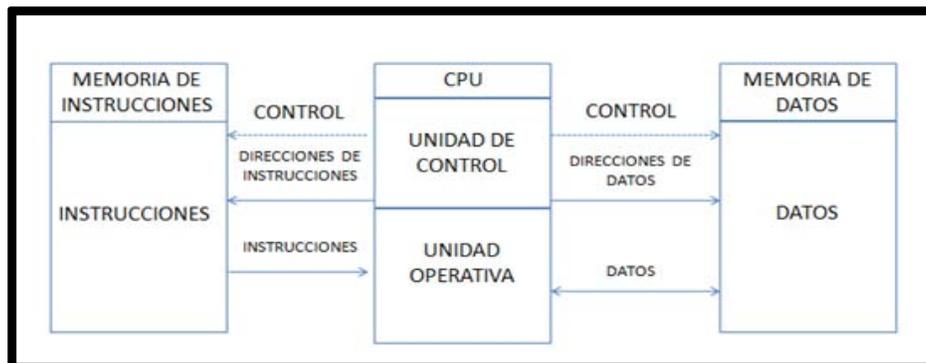


Figura 36. Arquitectura Harvard.

La ventaja fundamental de esta arquitectura es que permite adecuar el tamaño de los buses a las características de cada tipo de memoria; además, el procesador puede acceder a cada una de ellas de forma simultánea, lo que se traduce en un aumento significativo de la velocidad de procesamiento. Típicamente los sistemas con esta arquitectura pueden ser dos veces más rápidos que sistemas similares con arquitectura Von Neumann.

La desventaja está en que consume muchas líneas de E/S del procesador; por lo que en sistemas donde el procesador está ubicado en su propio encapsulado, solo se utiliza en supercomputadoras. Sin embargo, en los microcontroladores y otros sistemas integrados, donde usualmente la memoria de datos y programas comparten el mismo encapsulado que el procesador, este inconveniente deja de ser un problema serio y es por ello que encontramos la arquitectura Harvard en la mayoría de los microcontroladores.

Por eso es importante recordar que un microcontrolador se puede configurar de diferentes maneras, siempre y cuando se respete el tamaño de memoria que este requiera para su correcto funcionamiento.

2.4.3.3 REGISTROS

Los registros sirven para almacenar los resultados de la ejecución de instrucciones, cargar datos desde la memoria interna o externa o almacenarlos en ella.

2.4.3.4 UNIDAD DE CONTROL

Esta unidad es de las más importantes en el procesador, en ella recae la lógica necesaria para la decodificación y ejecución de las instrucciones, el control de los registros, la ALU, los buses y cuanta cosa más se quiera meter en el procesador.

La unidad de control Figura 37 es uno de los elementos fundamentales que determinan las prestaciones del procesador, ya que su tipo y estructura, determina parámetros tales como el tipo de conjunto de instrucciones, velocidad de ejecución, tiempo del ciclo de máquina, tipo de buses que puede tener el sistema, manejo de interrupciones y un buen número de cosas más que en cualquier procesador van a parar a este bloque.

Por supuesto, las unidades de control, son el elemento más complejo de un procesador y normalmente están divididas en unidades más pequeñas trabajando de conjunto. La unidad de control agrupa componentes tales como la unidad de decodificación, unidad de ejecución, controladores de memoria cache, controladores de buses, controlador de interrupciones, pipelines, entre otros elementos, dependiendo siempre del tipo de procesador.

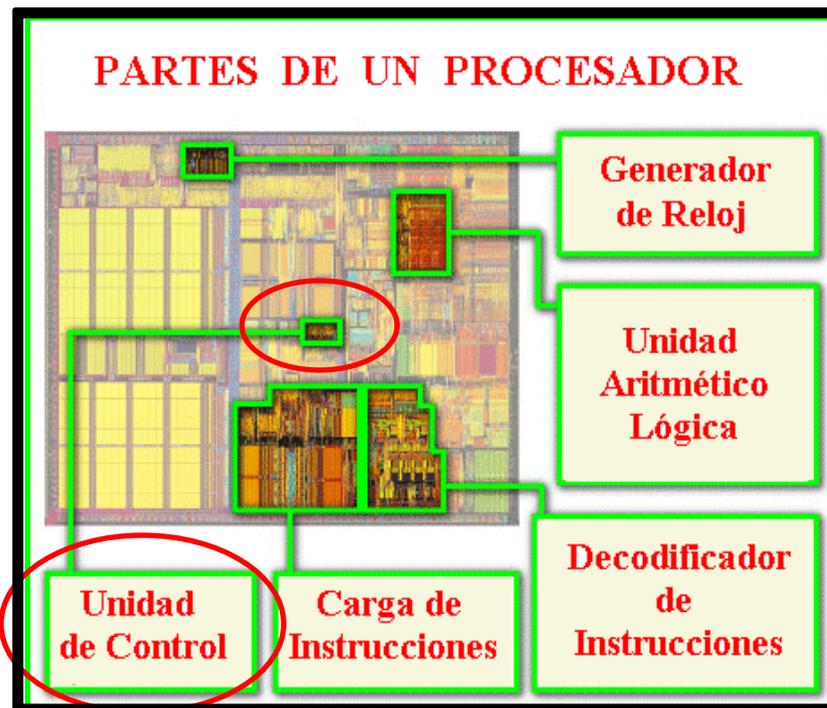


Figura 37. Unidad de control en un procesador

2.4.3.5 UNIDAD ARITMÉTICO-LÓGICA

Como los procesadores son circuitos que hacen básicamente operaciones lógicas y matemáticas, se le dedica a este proceso una unidad completa, con cierta independencia. Aquí es donde se realizan las sumas, restas, y operaciones lógicas típicas del álgebra de Boole.

Actualmente este tipo de unidades ha evolucionado mucho y los procesadores más modernos tienen varias ALU, especializadas en la realización de operaciones complejas como las operaciones en coma flotante. De hecho en muchos casos le han cambiado su nombre por el de “coprocesador matemático”¹³, aunque este es un término que surgió para dar nombre a un tipo especial de procesador que se conecta directamente al procesador más tradicional.

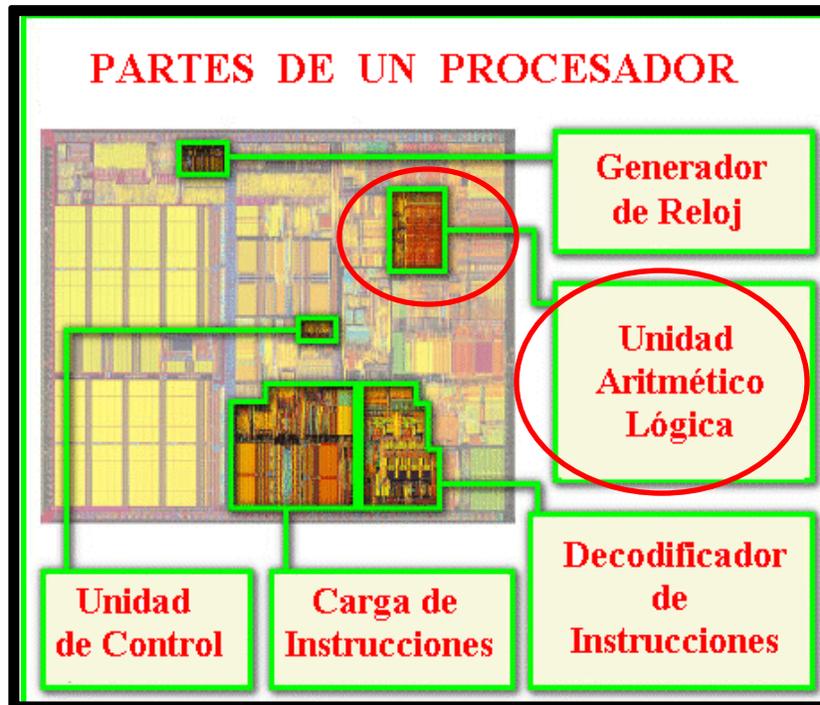


Figura 38. Unidad aritmética lógica

¹³ Utilizado como suplemento de las funciones del procesador principal.

2.4.3.6 BUSES

Son el medio de comunicación que utilizan los diferentes componentes del procesador para intercambiar información entre sí, eventualmente los buses o una parte de ellos estarán reflejados en los pines del encapsulado del procesador.

En el caso de los microcontroladores, no es común que los buses estén reflejados en el encapsulado del circuito, ya que estos se destinan básicamente a las E/S de propósito general y periféricos del sistema.

Existen tres tipos de buses:

- Dirección: Se utiliza para seleccionar al dispositivo con el cual se quiere trabajar o en el caso de las memorias, seleccionar el dato que se desea leer o escribir.
- Datos.
- Control: Se utiliza para gestionar los distintos procesos de escritura lectura y controlar la operación de los dispositivos del sistema.

2.4.3.7 CONJUNTO DE INSTRUCCIONES

Define las operaciones básicas que puede realizar el procesador, que conjugadas y organizadas forman lo que conocemos como software. El conjunto de instrucciones vienen siendo como las letras del alfabeto, el elemento básico del lenguaje, que organizadas adecuadamente permiten escribir palabras, oraciones y cuanto programa se le ocurra.

Existen dos tipos básicos de repertorios de instrucciones, que determinan la arquitectura del procesador: CISC y RISC. [12]

CISC: Del inglés Complex instruction set computing, Computadora de Conjunto de Instrucciones Complejo. Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y que permiten realizar operaciones complejas entre operandos situados en la memoria o en los registros internos. Este tipo de repertorio dificulta el paralelismo entre instrucciones, por lo que en la actualidad, la mayoría de los sistemas CISC de alto rendimiento, convierten las instrucciones complejas en varias instrucciones simples del tipo RISC, llamadas generalmente microinstrucciones.

Dentro de los microcontroladores CISC podemos encontrar a la popular familia Intel 8051 y la Z80, aunque actualmente existen versiones CISC-RISC de estos microcontroladores, que pretenden aprovechar las ventajas de los procesadores RISC a la vez que se mantiene la compatibilidad hacia atrás con las instrucciones de tipo CISC.

RISC: Del inglés Reduced Instruction Set Computer, Computadora con Conjunto de Instrucciones Reducido. Se centra en la obtención de procesadores con las siguientes características fundamentales: [12]

- Instrucciones de tamaño fijo.
- Pocas instrucciones.
- Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.
- Número relativamente elevado de registros de propósito general.

Una de las características más destacables de este tipo de procesadores es que posibilitan el paralelismo en la ejecución, y reducen los accesos a memoria. Es por eso que los procesadores más modernos, tradicionalmente basados en arquitecturas CISC implementan mecanismos de traducción de instrucciones CISC a RISC, para aprovechar las ventajas de este tipo de procesadores.

Los procesadores de los microcontroladores PIC son de tipo RISC.

2.4.4 MEMORIA

Anteriormente habíamos visto que la memoria en los microcontroladores debe estar ubicada dentro del mismo encapsulado, esto es así la mayoría de las veces, porque la idea fundamental es mantener el grueso de los circuitos del sistema dentro de un solo integrado.

En los microcontroladores la memoria no es abundante, aquí no encontrará Gigabytes de memoria como en las computadoras personales. Típicamente la memoria de programas no excederá de 16 K-localizaciones de memoria no volátil (flash o eprom) para contener los programas.

La memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además los registros de trabajo del procesador y los de configuración y trabajo de los distintos periféricos del microcontrolador. Es por ello que en la mayoría de los casos, aunque se tenga un espacio de direcciones de un tamaño determinado, la cantidad de memoria RAM de que dispone el programador para almacenar sus datos es menor que la que puede direccionar el procesador.

El tipo de memoria utilizada en las memorias RAM de los microcontroladores es SRAM, lo que evita tener que implementar sistemas de refrescamiento como en el caso de las computadoras personales, que utilizan gran cantidad de memoria, típicamente alguna tecnología DRAM. A pesar de que la memoria SRAM es más costosa que la DRAM, es el tipo adecuado para los microcontroladores porque éstos poseen pequeñas cantidades de memoria RAM.

En el caso de la memoria de programas se utilizan diferentes tecnologías, y el uso de una u otra depende de las características de la aplicación a desarrollar, a continuación se describen las cinco tecnologías existentes, que mayor utilización tienen o han tenido:

- **MÁSCARA ROM** En este caso no se “graba” el programa en memoria sino que el microcontrolador se fabrica con el programa, es un proceso similar al de producción de los CD comerciales mediante masterización. El costo inicial de producir un circuito de este tipo es alto, porque el diseño y producción de la máscara es un proceso costoso, sin embargo, cuando se necesitan varios miles o incluso cientos de miles de microcontroladores para una aplicación determinada, como por ejemplo, algún electrodoméstico, el costo inicial de producción de la máscara y el de fabricación del circuito se distribuye entre todos los circuitos de la serie y, el costo final de ésta, es bastante menor que el de sus semejantes con otro tipo de memoria. [12]
- **MEMORIA PROM** (Programmable Read-Only Memory) también conocida como OTP (One Time Programmable). Este tipo de memoria, también es conocida como PROM o simplemente ROM. Los microcontroladores con memoria OTP se pueden programar una sola vez, con algún tipo de programador. Se utilizan en sistemas donde el programa no requiera futuras actualizaciones y para series relativamente pequeñas, donde la variante de máscara sea muy costosa, también para sistemas que requieren serialización de datos, almacenados como constantes en la memoria de programas. [12]

- **MEMORIA EPROM** (Erasable Programmable Read Only Memory). Los microcontroladores con este tipo de memoria son muy fáciles de identificar porque su encapsulado es de cerámica y llevan encima una ventanita de vidrio desde la cual puede verse la oblea de silicio del microcontrolador. Se fabrican así porque la memoria EPROM es reprogramable, pero antes debe borrarse, y para ello hay que exponerla a una fuente de luz ultravioleta, el proceso de grabación es similar al empleado para las memorias OTP. Al aparecer tecnologías menos costosas y más flexibles, como las memorias EEPROM y FLASH, este tipo de memoria han caído en desuso, se utilizaban en sistemas que requieren actualizaciones del programa y para los procesos de desarrollo y puesta a punto. [12]
- **EEPROM** (Electrical Erasable Programmable Read Only Memory). Fueron el sustituto natural de las memorias EPROM, la diferencia fundamental es que pueden ser borradas eléctricamente, por lo que la ventanilla de cristal de cuarzo y los encapsulados cerámicos no son necesarios. Al disminuir los costos de los encapsulados, los microcontroladores con este tipo de memoria se hicieron más baratos y cómodos para trabajar que sus equivalentes con memoria EPROM. Otra característica destacable de este tipo de microcontrolador es que fue en ellos donde comenzaron a utilizarse los sistemas de programación en el sistema que evitan tener que sacar el microcontrolador de la tarjeta que lo aloja para hacer actualizaciones al programa. [12]
- **MEMORIA FLASH** En el campo de las memorias reprogramables para microcontroladores, son el último avance tecnológico en uso a gran escala, y han sustituido a los microcontroladores con memoria EEPROM. [12]

A las ventajas de las memorias flash se le adicionan su gran densidad respecto a sus predecesoras lo que permite incrementar la cantidad de memoria de programas a un costo muy bajo. Pueden además ser programadas con las mismas tensiones de alimentación del microcontrolador, el acceso en lectura y la velocidad de programación es superior, disminución de los costos de producción, entre otras.

Lo más habitual es encontrar que la memoria de programas y datos está ubicada toda dentro del microcontrolador, de hecho, actualmente son pocos los microcontroladores que permiten conectar memoria de programas en el exterior del encapsulado. Las razones para estas “limitaciones” están dadas porque el objetivo fundamental es obtener la mayor integración posible y conectar memorias externas consume líneas de E/S que son uno de los recursos más preciados de los microcontroladores.

A pesar de lo anterior existen familias como la Intel 8051 cuyos microcontroladores tienen la capacidad de ser expandidos en una variada gama de configuraciones para el uso de memoria de programas externa. En el caso de los PIC, estas posibilidades están limitadas sólo a algunos microcontroladores de la gama alta, muestra algunas de las configuraciones para memoria de programa que podemos encontrar en los microcontroladores¹⁴.

¹⁴ Diseño de Arquitectura Harvard de acuerdo con los microcontroladores.

Cuando se requiere aumentar la cantidad de memoria de datos, lo más frecuente es colocar dispositivos de memoria externa en forma de periféricos, de esta forma se pueden utilizar memorias RAM, FLASH o incluso discos duros como los de los ordenadores personales, mientras que para los cálculos y demás operaciones que requieran almacenamiento temporal de datos se utiliza la memoria RAM interna del microcontrolador. Esta forma de expandir la memoria de datos está determinada, en la mayoría de los casos, por el tipo de repertorio de instrucciones del procesador y porque permite un elevado número de configuraciones distintas, además del consiguiente ahorro de líneas de E/S que se logra con el uso de memorias con buses de comunicación serie.

2.4.5 INTERRUPTIONES

Las interrupciones son esencialmente llamadas a subrutina generadas por los dispositivos físicos, al contrario de las subrutinas normales de un programa en ejecución. Como el salto de subrutina no es parte del hilo o secuencia de ejecución programada, el controlador guarda el estado del procesador en la pila de memoria y entra a ejecutar un código especial llamado "manejador de interrupciones" que atiende al periférico específico que generó la interrupción. Al terminar la rutina, una instrucción especial le indica al procesador el fin de la atención de la interrupción Figura 38. En ese momento el controlador restablece el estado anterior, y el programa que se estaba ejecutando antes de la interrupción sigue como si nada hubiese pasado.



Figura 38. Rutina de Interrupciones

2.4.6 ENTRADAS Y SALIDAS DE PROPÓSITO GENERAL

También conocidos como puertos de E/S, generalmente agrupadas en puertos de 8 bits de longitud, permiten leer datos del exterior o escribir en ellos desde el interior del microcontrolador, el destino habitual es el trabajo con dispositivos simples como relés, LED, o cualquier otra cosa que se le ocurra al programador.

Algunos puertos de E/S tienen características especiales que le permiten manejar salidas con determinados requerimientos de corriente, o incorporan mecanismos especiales de interrupción para el procesador.

Típicamente cualquier pin de E/S puede ser considerada E/S de propósito general, pero como los microcontroladores no pueden tener infinitos pines, ni siquiera todos los pines que queramos, las E/S de propósito general comparten los pines con otros periféricos. Para usar un pin con cualquiera de las características a él asignadas debemos configurarlo mediante los registros destinados a ellos.

2.4.7 TEMPORIZADORES Y CONTADORES

Son circuitos sincrónicos para el conteo de los pulsos que llegan a su poder para conseguir la entrada de reloj. Si la fuente de un gran conteo es el oscilador interno del microcontrolador es común que no tengan un pin asociado, y en este caso trabajan como temporizadores. Por otra parte, cuando la fuente de conteo es externa, entonces tienen asociado un pin configurado como entrada, este es el modo contador.

Los temporizadores son uno de los periféricos más habituales en los microcontroladores y se utilizan para muchas tareas, como por ejemplo, la medición de frecuencia, implementación de relojes, para el trabajo de conjunto con otros periféricos que requieren una base estable de tiempo entre otras funcionalidades. Es frecuente que un microcontrolador típico incorpore más de un temporizador/contador e incluso algunos tienen arreglos de contadores. Como veremos más adelante este periférico es un elemento casi imprescindible y es habitual que tengan asociada alguna interrupción. Los tamaños típicos de los registros de conteo son 8 y 16 bits, pudiendo encontrar dispositivos que solo tienen temporizadores de un tamaño o con más frecuencia con ambos tipos de registro de conteo.

2.4.8 CONVERSION ANALÓGICO/DIGITAL

Como es muy frecuente el trabajo con señales analógicas, éstas deben ser convertidas a digital y por ello muchos microcontroladores incorporan un conversor analógico-digital Figura 38l, el cual se utiliza para tomar datos de varias entradas diferentes que se seleccionan mediante un multiplexor.

Las resoluciones más frecuentes son 8 y 10 bits, que son suficientes para aplicaciones sencillas. Para aplicaciones en control e instrumentación están disponibles resoluciones de 12bit, 16bit y 24bit¹⁵. También es posible conectar un convertidor externo, en caso de necesidad.

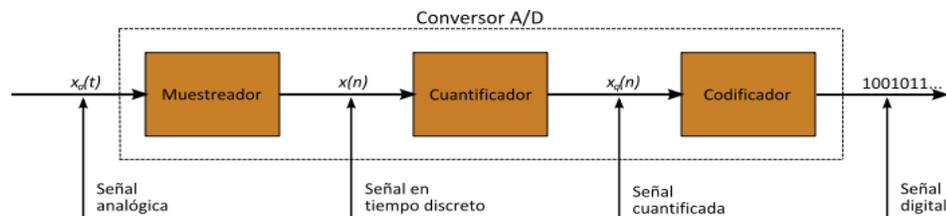


Figura 39. Conversor analógico digital

¹⁵ Muestreo de la señal analógica.

2.4.9 PUERTOS DE COMUNICACIÓN

2.4.9.1 Puerto serie

Este periférico está presente en casi cualquier microcontrolador, normalmente en forma de UART (Universal Asynchronous Receiver Transmitter) o USART (Universal Synchronous Asynchronous Receiver Transmitter) dependiendo de si permiten o no el modo sincrónico de comunicación.

El destino común de este periférico es la comunicación con otro microcontrolador o con una PC y en la mayoría de los casos hay que agregar circuitos externos para completar la interfaz de comunicación. La forma más común de completar el puerto serie es para comunicarlo con una PC mediante la interfaz EIA-232 (más conocida como RS-232), es por ello que muchas personas se refieren a la UART o USART como puerto serie RS-232, pero esto constituye un error, puesto que este periférico se puede utilizar para interconectar dispositivos mediante otros estándares de comunicación. En aplicaciones industriales se utiliza preferiblemente RS-485 por su superior alcance en distancia, velocidad y resistencia al ruido¹⁶.



2.4.9.2 SPI

Figura 39. Puerto serie para comunicaciones de la PC.

Este tipo de periférico se utiliza para comunicar al microcontrolador con otros microcontroladores o con periféricos externos conectados a él, por medio de una interfaz muy sencilla. Hay solo un nodo controlador que permite iniciar cualquier transacción, lo cual es una desventaja en sistemas complejos, pero su sencillez permite el aislamiento galvánico de forma directa por medio de opto acopladores.

¹⁶ RS232 Serial.

2.4.9.3 I2C

Cumple las funciones que el SPI, requiere menos señales de comunicación y cualquier nodo puede iniciar una transacción. Es muy utilizado para conectar las tarjetas gráficas de las computadoras personales con los monitores, para que estos últimos informen de sus prestaciones y permitir la autoconfiguración del sistema de vídeo.

2.4.9.4 USB

Los microcontroladores son los que han permitido la existencia de este sistema de comunicación. Es un sistema que trabaja por polling (monitorización) de un conjunto de periféricos inteligentes por parte de un amo, que es normalmente un computador personal. Cada modo inteligente está gobernado inevitablemente por un microcontrolador¹⁷.



Figura 40. Puerto de comunicación USB

2.4.9.5 Ethernet

Es el sistema más extendido en el mundo para redes de área local cableadas. Los microcontroladores más poderosos de 32 bits se usan para implementar periféricos lo suficientemente poderosos como para que puedan ser accedidos directamente por la red¹⁸. Muchos de los enrutadores caseros de pequeñas empresas están construidos en base a un microcontrolador que hace del cerebro del sistema.



Figura 41. Puerto Jack tipo Ethernet.

¹⁷ Características del puerto USB.

¹⁸ Características del puerto Ethernet.

2.4.9.6 Can

Este protocolo es del tipo CSMA/CD con tolerancia a elevados voltajes de modo común y orientado al tiempo real. Este protocolo es el estándar más importante en la industria automotriz (OBD). También se usa como capa física del "field bus" para el control industrial.

2.4.9.7 Otros puertos de comunicación

Hay una enorme cantidad de otros buses disponibles para la industria automotriz (linbus) o de medios audiovisuales como el i2s, IEEE 1394. Es usuario se los encontrará cuando trabaje en algún área especializada.

2.4.10 COMPARADORES

Son circuitos analógicos basados en amplificadores operacionales que tienen la característica de comparar dos señales analógicas y dar como salida los niveles lógicos „0" o „1" en dependencia del resultado de la comparación. Es un periférico muy útil para detectar cambios en señales de entrada de las que solamente nos interesa conocer cuando está en un rango determinado.

2.4.11 MODULADOR DE ANCHO DE PULSOS

Los PWM (Pulse Width Modulator) son periféricos muy útiles sobre todo para el control de motores, sin embargo hay un grupo de aplicaciones que pueden realizarse con este periférico, dentro de las cuales podemos citar: inversión DC/AC para UPS, conversión digital analógica D/A, control regulado de luz (dimming) entre otras.

2.4.12 MEMORIA DE DATOS NO VOLÁTIL

Muchos microcontroladores han incorporado este tipo de memoria como un periférico más, para el almacenamiento de datos de configuración o de los procesos que se controlan. Esta memoria es independiente de la memoria de datos tipo RAM o la memoria de programas, en la que se almacena el código del programa a ejecutar por el procesador del microcontrolador.

Muchos de los microcontroladores PIC, incluyen este tipo de memoria, típicamente en forma de memoria EEPROM, incluso algunos de ellos permiten utilizar parte de la memoria de programas como memoria de datos no volátil, por lo que el procesador tiene la capacidad de escribir en la memoria de programas como si ésta fuese un periférico más.

2.4.13 FAMILIAS DE MICROCONTROLADORES

Los microcontroladores más comunes en uso se muestran en la siguiente tabla:

| Empresa | 8 bits | 16 bits | 32 bits |
|----------------------------|---|---|---|
| Atmel | AVR (mega y tiny), 89Sxxxx familia similar 8051 | | SAM7 (ARM7TDMI), SAM3 (ARM Cortex-M3), SAM9 (ARM926), AVR32 |
| Freescale (antes Motorola) | 68HC05, 68HC08, 68HC11, HCS08 | 68HC12, 68HCS12, 68HCSX12, 68HC16 | 683xx, PowerPC, ColdFire |
| Holtek | HT8 | | |
| Intel | MCS-48 (familia 8048) MCS51 (familia 8051) 8xC251 | MCS96, MXS296 | x |
| National Semiconductor | COP8 | x | x |
| Microchip | Familia 10f2xx Familia 12Cxx Familia 12Fxx, 16Cxx y 16Fxx 18Cxx y 18Fxx | PIC24F, PIC24H y dsPIC30FXX, dsPIC33F con motor dsp integrado | PIC32 |
| NXP Semiconductors | 80C51 | XA | Cortex-M3, Cortex-M0, ARM7, |

| | | | |
|---|-------------|----------------------------------|--------------------------------------|
| (antes Philips) | | | ARM9 |
| Renesas (antes Hitachi, Mitsubishi y NEC) | 78K, H8 | H8S, 78K0R, R8C, R32C/M3 2C/M16C | RX, V850, SuperH, SH-Mobile, H8SX |
| STMicroelectronics | ST 62, ST 7 | | STM32 (ARM7) |
| Texas Instruments | TMS370 | MSP430 | C2000, Cortex-M3 (ARM), TMS570 (ARM) |
| Zilog | Z8, Z86E02 | | |

Tabla 2. Tipos de micro controladores.

2.5 ELECCIÓN DE UN MICROCONTROLADOR

La elección de un microcontrolador es más un arte que una ciencia. El poder de decisión es tremendamente extenso, se pueden elegir entre decenas de fabricante y dentro de éstos entre microprocesadores de la misma familia, y entre distintas arquitecturas (CISC, RISC, VLIW, súper-escalar).

2.5.1 REGLAS BÁSICAS:

Nivel de rendimiento (performance orthrough put): Se debe determinar el nivel de tiempo de funcionamiento en el peor de los casos. Los parámetros más importantes son el tiempo del ciclo de instrucción y la velocidad del bus, que deben ser evaluados conjuntamente. Un bus de mayor velocidad con el menor número de ciclos por instrucción es el que proporciona mayor rendimiento.

Tamaño del código: La simplificación de código de programa pasa por disponer de eficientes conjuntos de instrucciones y de modos de direccionamiento (óptimas arquitecturas).

Latencia de las interrupciones y rendimiento de los periféricos on-chip: Condicionados por las especificaciones de la aplicación.

Consumo de potencia: En diversas áreas, el consumo de potencia es el factor determinante, incluso por encima del rendimiento (telefonía móvil, en aparatos médicos como audífonos, marcapasos, etc). La relación MIPS/watt es el parámetro utilizado para catalogarlos.

Se produce un balance con respecto a los demás requerimientos del microcontrolador, por ejemplo, puede ser deseable una velocidad del bus más lenta para conseguir menor consumo de potencia. Los micros con tecnología CMOS normalmente tienen un menor consumo que los de tecnología NMOS. Es deseable usar microcos con modos standby o slepp que limitan el consumo de intensidad a varios microamperios cuando están inactivos los circuitos. El consumo de potencia, grosso modo, se suele repartir como un 25% para el procesador, un 50% para las memorias y el 25% restante para los periféricos. Actualmente, existe un amplio abanico de posibilidades de elección de microcontroladores de ultra-bajo consumo.

Adaptación a la aplicación: Una vez considerado los requerimientos del sistema en cuanto a rendimiento (elegir entre varias familias de microcontroladores), el siguiente paso es exigir que cumpla los requerimientos de funcionalidad de los periféricos: tamaño de memoria, tipo de memoria (EPROM, Flash, ...), número de timer, requerimientos de puertos de E/S, etc.

Compatibilidad: Otro factor importante es contemplar la posibilidad de ampliación del diseño, esto es, que sea compatible con la siguiente generación y proporcione mayores prestaciones, como más memoria, mayor frecuencia,.... Esto evitará una nuevo re-diseño del sistema.

Disponibilidad: Se debe asegurar la disponibilidad futura en el mercado, por ello es interesante que existan varios fabricantes del microprocesador, además de que esto proporciona un abaratamiento del coste del micro. Otras consideraciones son asegurarse de que el microcontrolador esté ya en producción y libre de fallos.

Soporte y documentación: Debe existir un buen manual de usuario (User's Guide), hoja de datos software y hardware, la mayor cantidad de documentación de aplicaciones ya resueltas (Application Guide) con códigos fuente completos y artículos técnicos y científicos, así como un servicio de consultas accesible cómodamente y con respuestas rápidas (E-mail, fax, teléfono).

Herramientas de desarrollo: Las herramientas de desarrollo, tanto de hardware como de software, son imprescindibles para el desarrollo de la aplicación. Se debe asegurar que sean de alta calidad, no tengan errores (bugs), y con memorias RAM y memorias programables, entre las que se están consolidando las memorias OTP y las FLASH. También, es importante la conexión con un ordenador vía el estándar serie RS-232 o USB. La mayoría de las herramientas de desarrollo actuales tratan de forma totalmente separada el diseño hardware y el diseño software, aunque se están desarrollando herramientas para el co-diseño hardware/software que permite un proceso integrado y más óptimo de cada una de las partes del diseño completo.

2.6 ARDUINO

Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un micro controlador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa¹⁹.

DATASHEET ARDUINO

| | |
|------------------------------------|--|
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40mA |
| DC Current for 3.3V Pin | 50mA |
| Flash Memory | 32KB (ATmega328) of which 0.5 KB used by bootloader |
| SRAM | 2KB (ATmega328) |
| EEPROM | 1KB (ATmega328) |
| Clock Speed | 16MHz |

Figura 42. Datasheet Arduino (Modelo Atmega 328)

¹⁹ Características principales del microcontrolador Arduino.

2.6.1 ENTRADAS Y SALIDAS

Poniendo de ejemplo al módulo Uno (figura 43), éste consta de 14 entradas digitales configurables como entradas y/o salidas que operan a 5 voltios. Cada contacto puede proporcionar o recibir como máximo 40 mA. Los contactos 3, 5, 6, 8, 10 y 11 pueden proporcionar una salida PWM (Pulse Width Modulation). Si se conecta cualquier cosa a los contactos 0 y 1, eso interferirá con la comunicación USB. Diecimila también tiene 6 entradas analógicas que proporcionan una resolución de 10 bits. Por defecto, aceptan de 0 hasta 5 voltios, aunque es posible cambiar el nivel más alto, utilizando el contacto Aref y algún código de bajo nivel.

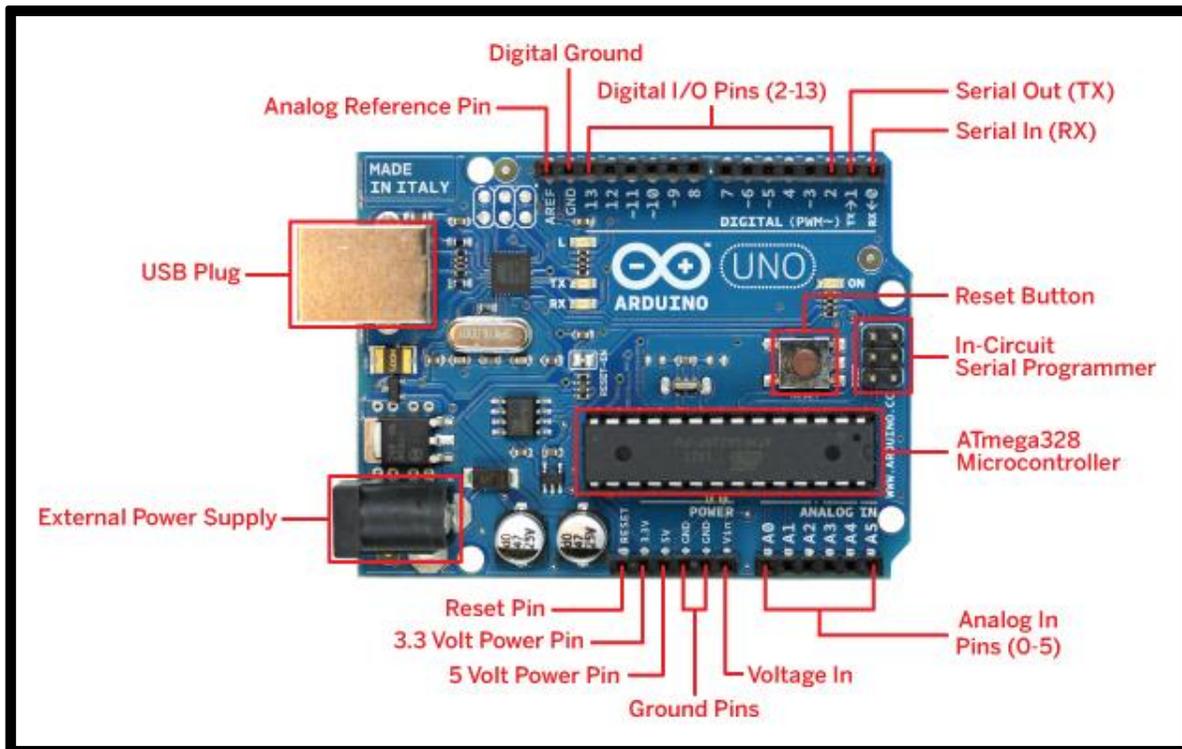


Figura 43. Arduino modelo uno (Chip Atmega 328), Tipos de Funciones.

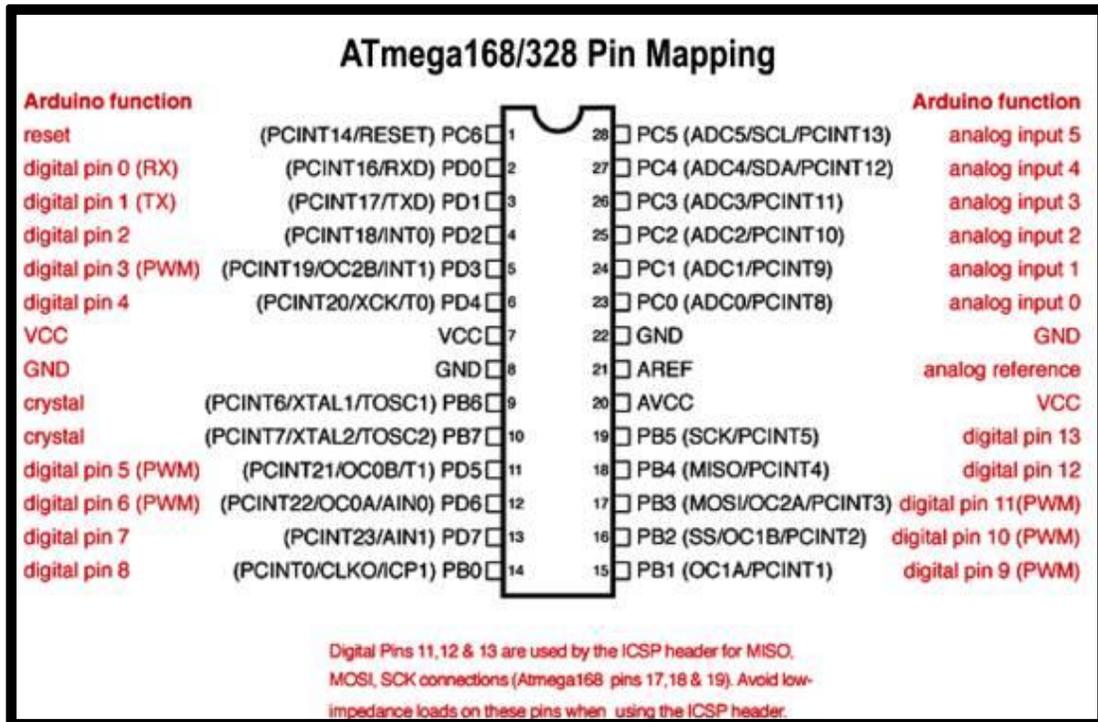


Figura 44. Datasheet de Entradas y salidas Arduino

2.6.2 FUNCIONES BÁSICAS Y OPERADORES

Arduino está basado en C y soporta todas las funciones del estándar C y algunas de C++. A continuación se muestra un resumen con la estructura y sintaxis del lenguaje Arduino:

Sintaxis Básica

- Delimitadores: ;, {}
- Comentarios: //, /* */
- Cabeceras: #define, #include
- Operadores aritméticos: +, -, *, /, %
- Asignación: =
- Operadores de comparación: ==, !=, <, >, <=, >=
- Operadores Booleanos: &&, ||, !
- Operadores de acceso a punteros: *, &
- Operadores de bits: &, |, ^, ~, <<, >>
- Operadores compuestos:
 - Incremento y decremento de variables: ++, --
 - Asignación y operación: +=, -=, *=, /=, &=, |=

Estructuras de control

- Condicionales: if, if...else, switch case
- Bucles: for, while, do... while
- Bifurcaciones y saltos: break, continue, return, goto

Variables

En cuanto al tratamiento de las variables también comparte un gran parecido con el lenguaje C.

Constantes

- HIGH/LOW: representan los niveles alto y bajo de las señales de entrada y salida. Los niveles altos son aquellos de 3 voltios o más.
- INPUT/OUTPUT: entrada o salida.
- false (falso): Señal que representa al cero lógico. A diferencia de las señales HIGH/LOW, su nombre se escribe en letra minúscula.
- true (verdadero): Señal cuya definición es más amplia que la de *false*. Cualquier número entero diferente de cero es "verdadero", según el álgebra de Boole, como en el caso de -200, -1 o 1. Si es cero, es "falso".

Tipos de datos

- void, boolean, char, unsigned char, byte, int, unsigned int, word, long, unsigned long, float, double, string, array.

Conversión entre tipos

Estas funciones reciben como argumento una variable de cualquier tipo y devuelven una variable convertida en el tipo deseado.

- char(), byte(), int(), word(), long(), float()

Cualificadores y ámbito de las variables

- static, volatile, const

Utilidades

- sizeof()

Funciones Básicas

E/S Digital

- pinMode(pin, modo)
- digitalWrite(pin, valor)
- digitalRead(pin)

E/S Analógica

- analogReference(tipo)
- analogRead(pin)
- analogWrite(pin, valor)

E/S Avanzada

- shiftOut(dataPin, clockPin, bitOrder, valor)
- digitalWrite(pin, valor)

Tiempo

- digitalWrite(pin, valor)
- delay(ms)
- delayMicroseconds(microsegundos)

Matemáticas

- min(x, y), max(x, y), abs(x), constrain(x, a, b), map(valor, fromLow, fromHigh, toLow, toHigh), pow(base, exponente), sqrt(x)

Trigonometría

- `sin(rad)`, `cos(rad)`, `tan(rad)`

Números aleatorios

- `randomSeed(semilla)`, `long random(máx)`, `long random(mín, máx)`

Bits y Bytes

- `lowByte()`, `highByte()`, `bitRead()`, `bitWrite()`, `bitSet()`, `bitClear()`, `bit()`

Interrupciones externas

- `attachInterrupt(interruptión, función, modo)`
- `detachInterrupt(interruptión)`

Interrupciones

- `interrupts()`, `noInterrupts()`

Comunicación por puerto serie

Las funciones de manejo del puerto serie deben ir precedidas de la palabra "Serial" aunque no necesitan ninguna declaración en la cabecera del programa. Por esto se consideran funciones base del lenguaje. Estas son las funciones para transmisión serial:

- `begin()`, `available()`, `read()`, `flush()`, `print()`, `println()`, `write()`

Manipulación de puertos

Los registros de puertos permiten la manipulación a más bajo nivel y de forma más rápida de los contactos de entrada/salida del microcontrolador de las placas Arduino. Los contactos eléctricos de las placas Arduino están repartidos entre los registros B(0-7), C (analógicos) y D(8-13). Mediante estas variables se observan y modifican su estado:

- **DDR[B/C/D]:** Data Direction Register (o dirección del registro de datos) del puerto B, C o D. Es una variable de Lectura/Escritura que sirve para especificar cuáles contactos serán usados como entrada y salida.
- **PORT[B/C/D]:** Data Register (o registro de datos) del puerto B, C o D. Es una variable de Lectura/Escritura.
- **PIN[B/C/D]:** Input Pins Register (o registro de pines de entrada) del puerto B, C o D. Variable de sólo lectura.

2.7 CONTROL DE MOTORES

A continuación procederemos a dar una explicación del chip L293B que usaremos para poder suministrar y controlar los motores del robot así como su datasheet y sus funciones que realiza.

2.7.1 DESCRIPCIÓN L293B

El L293B (figura 45) es un driver de 4 canales capaz de proporcionar una corriente de salida hasta 1A por canal. Cada canal es controlado por señales de entrada compatibles TTL y cada pareja de canales dispone de una señal de habilitación que desconecta las salidas de los mismos.

Dispone de una patilla para la alimentación de las cargas que se están controlando, de forma que dicha alimentación es independiente de la lógica de control.

La Figura 45 muestra el encapsulado de 16 pines, la distribución de patillas y la descripción de las mismas.

Pin Nombre Descripción Patillaje

- | | | |
|----|----------------|-----------------------------------|
| 1 | Chip Enable 1 | Habilitación de los canales 1 y 2 |
| 2 | Input 1 | Entrada del Canal 1 |
| 3 | Output 1 | Salida del Canal 1 |
| 4 | GND | Tierra de Alimentación |
| 5 | GND | Tierra de Alimentación |
| 6 | Output 2 | Salida del Canal 2 |
| 7 | Input 2 | Entrada del Canal 1 |
| 8 | V _s | Alimentación de las cargas |
| 9 | Chip Enable 2 | Habilitación de los canales 3 y 4 |
| 10 | Input 3 | Entrada del Canal 3 |
| 11 | Output 3 | Salida del Canal 3 |
| 12 | GND | Tierra de Alimentación |
| 13 | GND | Tierra de Alimentación |
| 14 | Output 4 | Salida del Canal 4 |
| 15 | Input 4 | Entrada del Canal 4 |

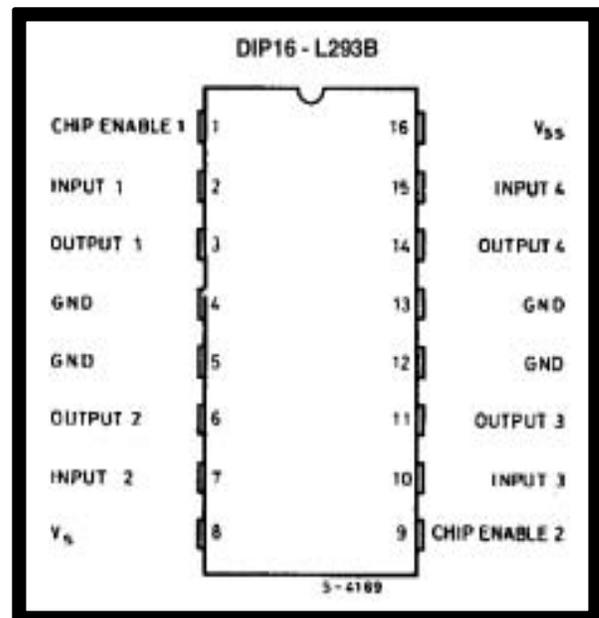
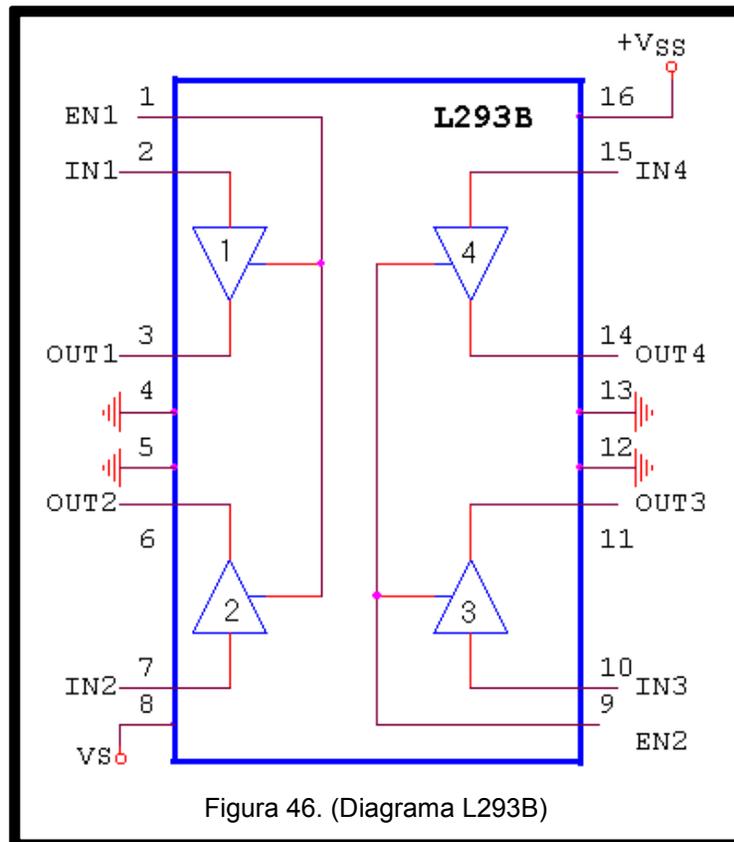


Figura 45.(Chip L293B) Descripción. [14]

2.7.2 DIAGRAMA DE BLOQUES

En la Figura 46, se muestra el diagrama de bloques del L293B. La señal de control **EN1** activa la pareja de canales formada por los drivers 1 y 2. La señal **EN2** activa la pareja de drivers 3 y 4. Las salidas **OUTn** se asocian con las correspondientes **OUTn**. Las señales de salida son amplificadas respecto a las de entrada tanto en tensión (hasta +Vss) como en corriente (máx. 1 A).



2.8 CASO DE ESTUDIO

A continuación se dará una breve explicación de las funciones básicas del robot su control y monitoreo, así también las herramientas a utilizar y su diseño de implementación a seguir. El robot podrá ser controlado y monitoreado de varias formas esto para que el usuario se sienta más a gusto.

La primera será a través de una interfaz gráfica de computadora la cual se desarrolló en visual studio usando como lenguaje de programación C# y el monitoreo a través del navegador Internet Explorer. Este primer sistema es el indicado para usuarios que quieran desempeñar la tarea de exploración ya sea cuevas, casas, pasillos etc. Este sistema cuenta con una interfaz la cual permite controlar el robot y un sistema para controlar la cámara la cual facilita la exploración al usuario.

Para la segunda forma de control se desarrolló y construyo un control remoto montado en un protoboard y el monitoreo a través de un celular, este segundo diseño a comparación del primero es más práctico y portable ya que reduce de tamaño todo el sistema pero con las mismas funciones. Este segundo diseño es ideal para usuarios que les guste la exploración al aire libre al ser muy portable.

Para nuestro último sistema de control consideramos el primer y segundo diseño para que fuera lo más práctico posible y sencillo de usar por lo que se decidió hacer una aplicación para celular esto para controlar y monitorear el robot prácticamente a la hora que queramos y donde queramos. Este último diseño se usa para prácticamente cualquier área exploración, seguridad, monitoreo de áreas o zonas peligrosas, etc.

2.9 SISTEMA MÍNIMO

Para el desarrollo de nuestra aplicación deben contemplarse los siguientes rubros:

- Tarjeta de desarrollo Arduino uno (Cualquier tipo de placa arduino)
- Entorno de desarrollo Microsoft Visual Studio 2010
- Circuito transmisor de radiofrecuencia TWS-433
- Circuito receptor de radiofrecuencia RWS-433
- Fritzing (Programa para realizar diagramas y circuitos en Arduino)
- Windows 7 (cualquier versión)

2.10 DISEÑO DE INTERFAZ DE RADIOFRECUENCIA

Objetivos:

- Diseñar e implementar una interfaz gráfica que le permita al usuario manipular el robot a través de la computadora.
- Diseñar e implementar un módulo de control para manipular al robot con movimientos básicos (adelante, atrás, izquierda y derecha).
- Diseñar e implementar un programa en una tarjeta con un sistema embebido, que permita la comunicación con la computadora y el robot.

En la figura 47 se muestra una figura esquemática de la interfaz de radiofrecuencia.

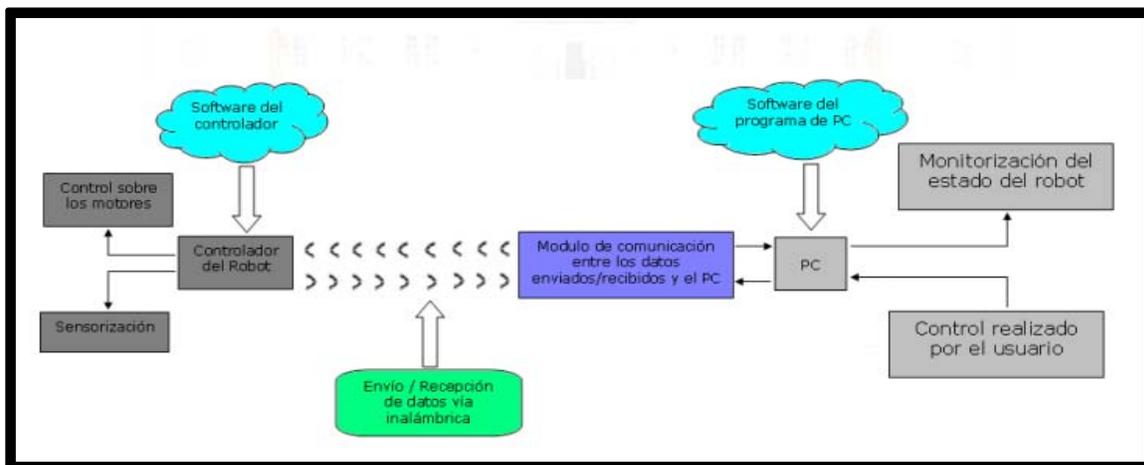
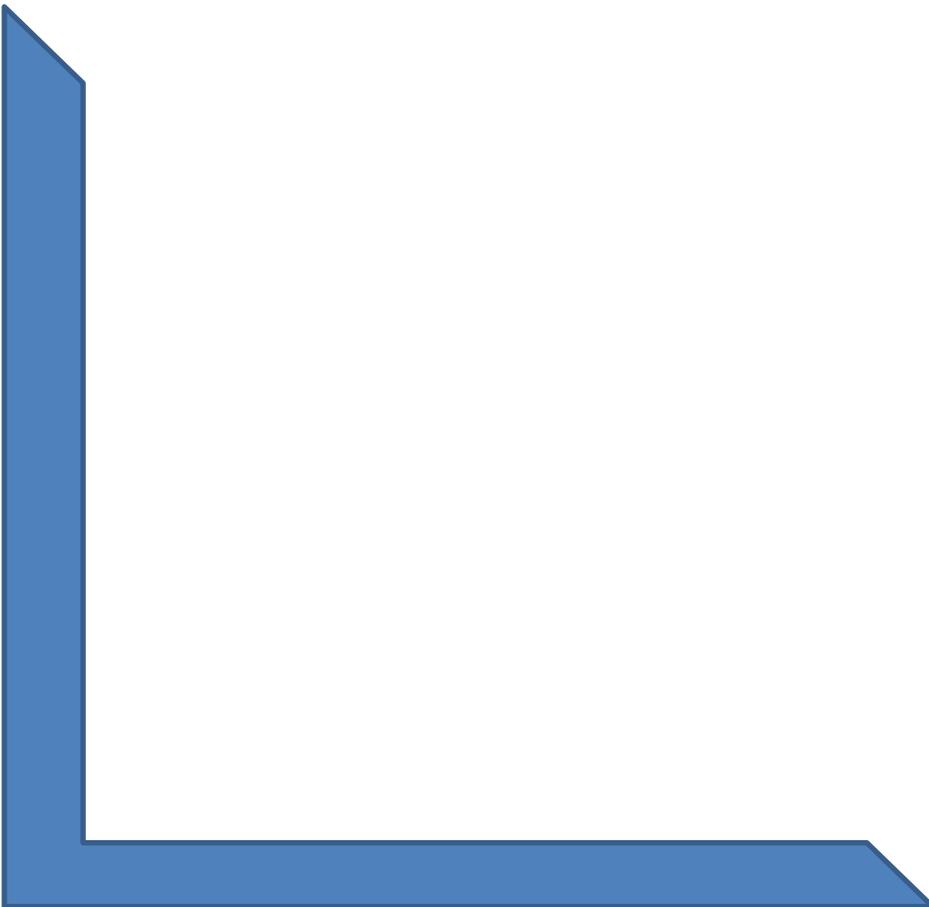


Figura 47. Diseño a Seguir del Modo de Comunicación del Proyecto.

CAPÍTULO 3



Capítulo 3

PROGRAMACIÓN E INTEGRACIÓN DEL SISTEMA DE APLICACIÓN

INTRODUCCIÓN

En este capítulo explicaremos la programación de las diferentes interfaces utilizadas para el control y monitoreo a través de una plataforma móvil considerando que en la integración se utiliza el sistema de desarrollo Arduino y el sistema de desarrollo Visual Studio 2010 Figura 48.



Figura 48. Entornos de desarrollo a usar en el proyecto

3.1 INTEGRACIÓN DE LOS SISTEMAS

En este capítulo explicaremos paso a paso la implementación de todos los sistemas de control a utilizar en el proyecto así también los diferentes lenguajes de programación a usar ya que para cada diseño y sistema se usaron códigos diferentes.

Como primer paso realizaremos el armado de los circuitos transmisor y receptor, para el paso 2 explicaremos el código usado para el transmisor y sus funciones, para el paso 3 mostraremos los procesos de compilación, para el paso 4 explicaremos como seleccionar el puerto de comunicación usado, en el paso 5 les mostraremos como introducir el código al arduino.

PASO 1. Armado circuito transmisor figura 49 para ilustrar de forma virtual se hace uso del programa Fritzing, en la Figura 50 se muestra el armado físico.

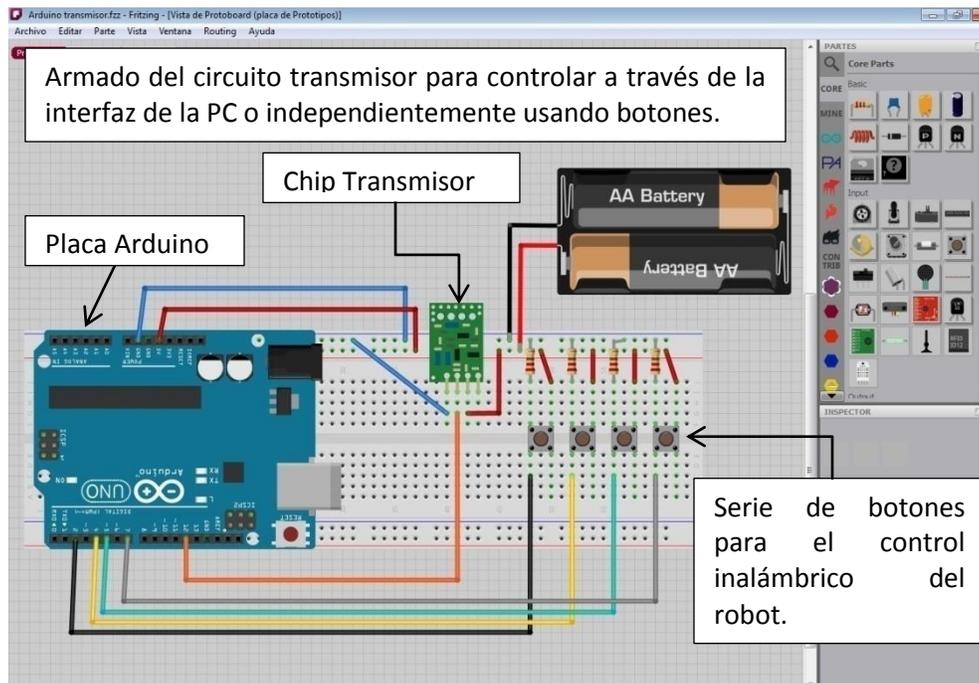


Figura 49. Armado Virtual de Comunicación del Transmisor.

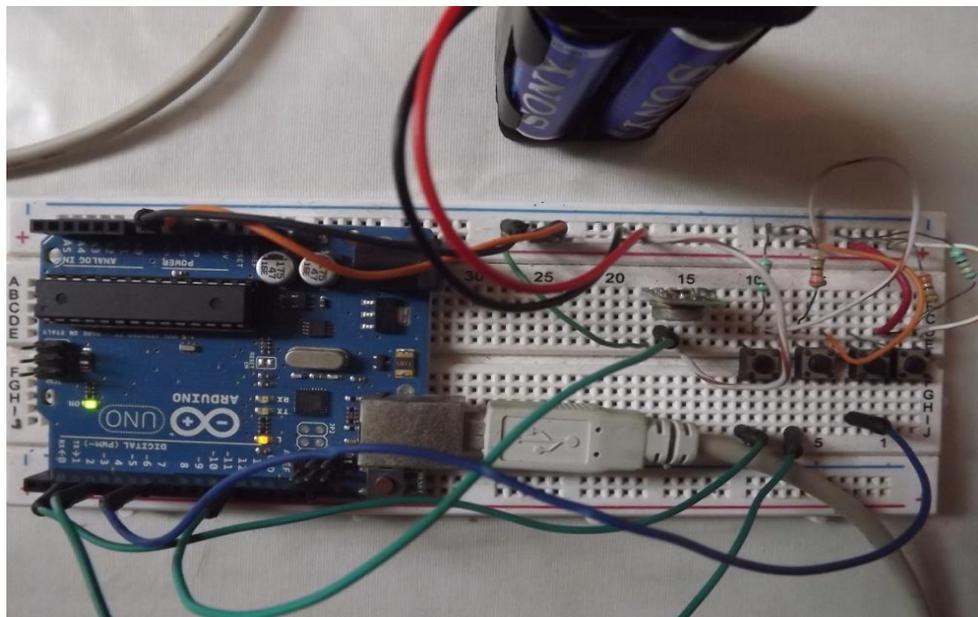


Figura 50. Armado físico del Circuito.

Ahora se mostrara el código de programación de Arduino con una breve explicación que llevara esta placa su correcto funcionamiento como transmisor.

PASO 2. PROGRAMACIÓN DEL CIRCUITO TRANSMISOR

El código de programación y su descripción se muestra a continuación es comunicación entre la computadora y la placa Arduino que va a funcionar como transmisor.

```
#include <VirtualWire.h>

void setup()
{
  Serial.begin(9600);

  Serial.println("setup"); // Inicializa el puerto serie
  vw_setup(2000); // Bits por segundo
}
```

Esta sección del código se encarga de establecer comunicación por el puerto serial (USB), y los bits por segundo que indican el rango de distancia.

El siguiente paso es saber si el puerto se encuentra abierto para eso se utiliza la instrucción (Serial.available), si es así leemos el dato que estamos recibiendo.

```
void loop()
{
  if (Serial.available() > 0) // Si el puerto serie está disponible
  {
    int sendByte = Serial.read(); // Empieza a leer datos del puerto serie
    switch (sendByte) { // Usamos la condición switch para leer el carácter
```

```
  case 'f': // Si se presiona f
  {
    char *msg2 = "f";
    digitalWrite(13, true); // Encendemos un led para mostrar la transmisión
    vw_send((uint8_t *)msg2, strlen(msg2)); // Se envía el byte al receptor
    vw_wait_tx(); // Esperamos a que todo el mensaje sea mandado
```

Una vez que leemos el dato recibido de la CPU, lo siguiente es leer el carácter que recibimos, ya que tenemos que carácter es, ahora será enviar ese carácter a la placa receptora.

```

digitalWrite(13, false); // Apagamos el led una vez enviado el byte
    break;
}
case 'b': // Si se presiona b
    {
char *msg2 = "b";
digitalWrite(13, true); // Encendemos un led para mostrar la transmisión
vw_send((uint8_t *)msg2, strlen(msg2)); // Se envía el byte al receptor
vw_wait_tx(); // Esperamos a que todo el mensaje sea mandado
digitalWrite(13, false); // Apagamos el led una vez enviado el byte
    break;
    }

    case 's': // Si se presiona s
    {
char *msg2 = "s";
digitalWrite(13, true); // Encendemos un led para mostrar la transmisión
vw_send((uint8_t *)msg2, strlen(msg2)); // Se envía el byte al receptor
vw_wait_tx(); // Esperamos a que todo el mensaje sea mandado
digitalWrite(13, false); // Apagamos el led una vez enviado el byte
    break;
    }

    case 'l': // Si se presiona l
    {
char *msg2 = "l";

```

```

digitalWrite(13, true); // Encendemos un led para mostrar la transmisión
vw_send((uint8_t *)msg2, strlen(msg2)); // Se envía el byte al receptor
vw_wait_tx(); // Esperamos a que todo el mensaje sea mandado
digitalWrite(13, false); // Apagamos el led una vez enviado el byte
    break;
}
    case 'r': // Si se presiona r
    {
char *msg2 = "r";
digitalWrite(13, true); // Encendemos un led para mostrar la transmisión
vw_send((uint8_t *)msg2, strlen(msg2)); // Se envía el byte al receptor
vw_wait_tx(); // Esperamos a que todo el mensaje sea mandado
digitalWrite(13, false); // Apagamos el led una vez enviado el byte
    break;
    }
    default:// Si se presiona cualquier otra tecla
    {

Serial.println("wrong entry"); // Imprime en pantalla
    }
    }
    }
}

```

El código que se pondrá a continuación es para controlar el robot mediante nuestro control montado en el protoboard usando radiofrecuencia.

```
#include <VirtualWire.h>

int forward = 8; // Declaración de pines a usar
int backward = 9;
int rightTurn = 10;
int leftTurn = 7;
int stopMotor = 11;
int remotePins[] = {7,8,9,10,11};
```

En esta sección declaramos todos los botones que usaremos para controlar el robot.

```
void setup()
{
  Serial.begin (9600);
  Serial.println ("setup");
  // inicializamos el Puerto serie
  vw_setup (2000); // Bits por segundo
```

En esta sección inicializamos el puerto serie e indicamos los bits por segundo

```
for(int i = 0; i<3 ; i++)
{
  pinMode(remotePins[i], INPUT); // declaramos que los pines a usar sean de salida
}
```

Indicamos que los pines declarados sean de entrada.

```
void loop()
{
  char *msg2; // cargamos el mensaje
```

Dependiendo del botón que se presione se enviara el numero asignado al receptor, encenderemos y apagaremos un led para indicar que se envié correctamente.

```
if(digitalRead(forward) == HIGH) //si el boton forward es presionado
{
  char *msg2 = "1"; //enviamos un 1 al receptor
  digitalWrite(13, true); // encendemos un led para aver que se envie el mensaje
  vw_send((uint8_t *)msg2, strlen(msg2)); //enviamos el byte al receptor
  vw_wait_tx(); // esperamos a que todo el mensaje sea mandado
  digitalWrite(13, false); // Apagamos el led
}
```

```
if(digitalRead(backward) == HIGH) //si el boton backward es presionado
{
  char *msg2 = "2"; //enviamos un 2 al receptor
  digitalWrite(13, true); // encendemos un led para mostrar la transmision
  vw_send((uint8_t *)msg2, strlen(msg2)); //se envia el byte al receptor
  vw_wait_tx(); // esperamos que se envie todo el mensaje
  digitalWrite(13, false); // Apagamos el led
}
```

```
if(digitalRead(leftTurn) == HIGH) //si el boton leftTurn es presionado
{
  char *msg2 = "3"; // enviamos un 3 al receptor
  digitalWrite(13, true); // encendemos un led para mostrar la transmision
```

```

vw_send((uint8_t *)msg2, strlen(msg2)); //se envia el byte al receptor
vw_wait_tx(); // esperamos que se envíe todo el mensaje
digitalWrite(13, false); //Apagamos el led
}

if(digitalRead(rightTurn) == HIGH) //si el boton rightTurn es presionado
{
char *msg2 = "4"; //enviamos un 4 al receptor
digitalWrite(13, true); // encendemos un led para mostrar transmision
vw_send((uint8_t *)msg2, strlen(msg2)); //enviamos el mensaje al receptor
vw_wait_tx(); // esperamos que se envíe todo el mensaje
digitalWrite(13, false); //Apagamos el led
}

if(digitalRead(stopMotor)==HIGH) //si el boton stopMotor es presionado
{
char *msg2 = "5"; //enviamos un 5 al receptor
digitalWrite(13, true); // se enciende un led para mostrar transmision
vw_send((uint8_t *)msg2, strlen(msg2)); //se envia el mensaje al receptor
vw_wait_tx(); // esperamos que se envíe todo el mensaje
digitalWrite(13, false); // Apagamos el led
}
}

```

PASO 3. COMPILACIÓN

Una vez que tenemos el código ahora el siguiente paso es compilarlo para ver que no tenga errores y transferirlo a la placa Arduino.

Para eso necesitamos abrir nuestro compilador Arduino que se muestra en la figura 51.

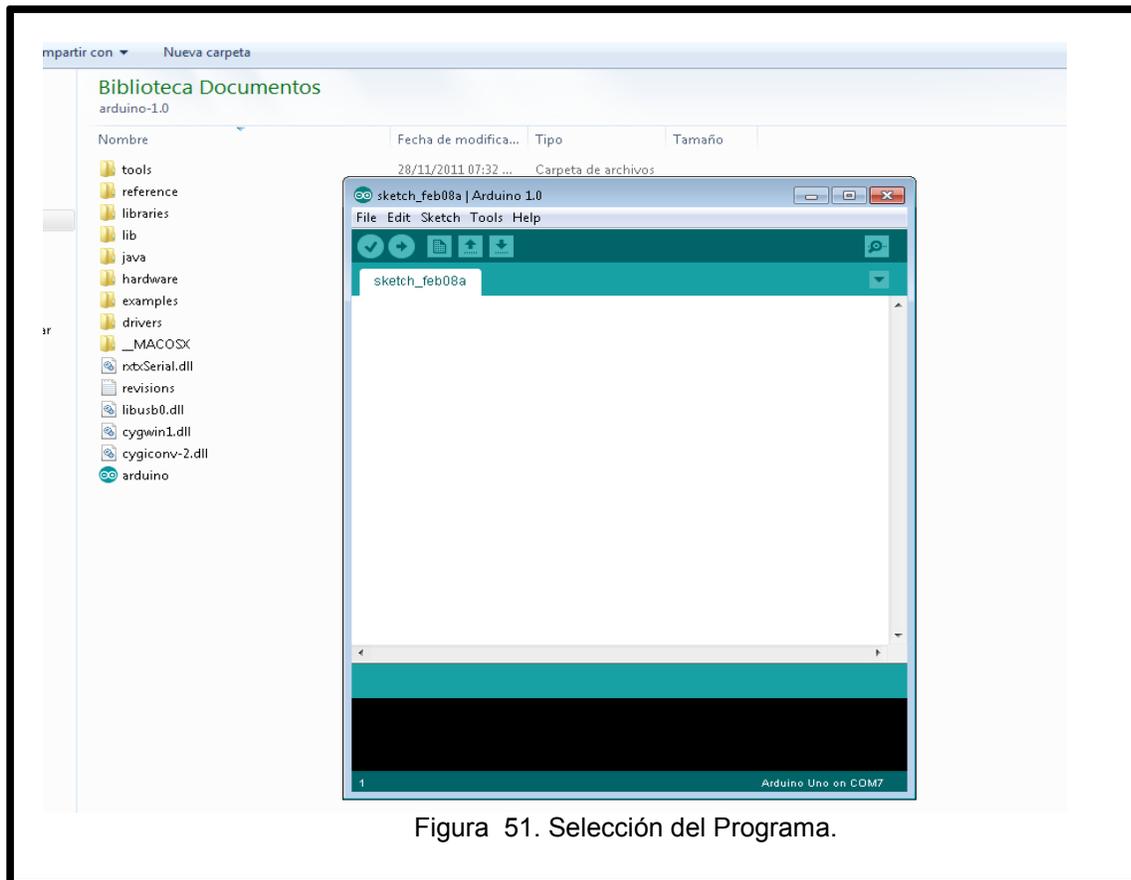


Figura 51. Selección del Programa.

Una vez abierto el programa seleccionaremos la placa arduino que estamos utilizando le damos en tools, board y seleccionamos en este caso estamos utilizando el modelo de placa arduino uno Figura 52.

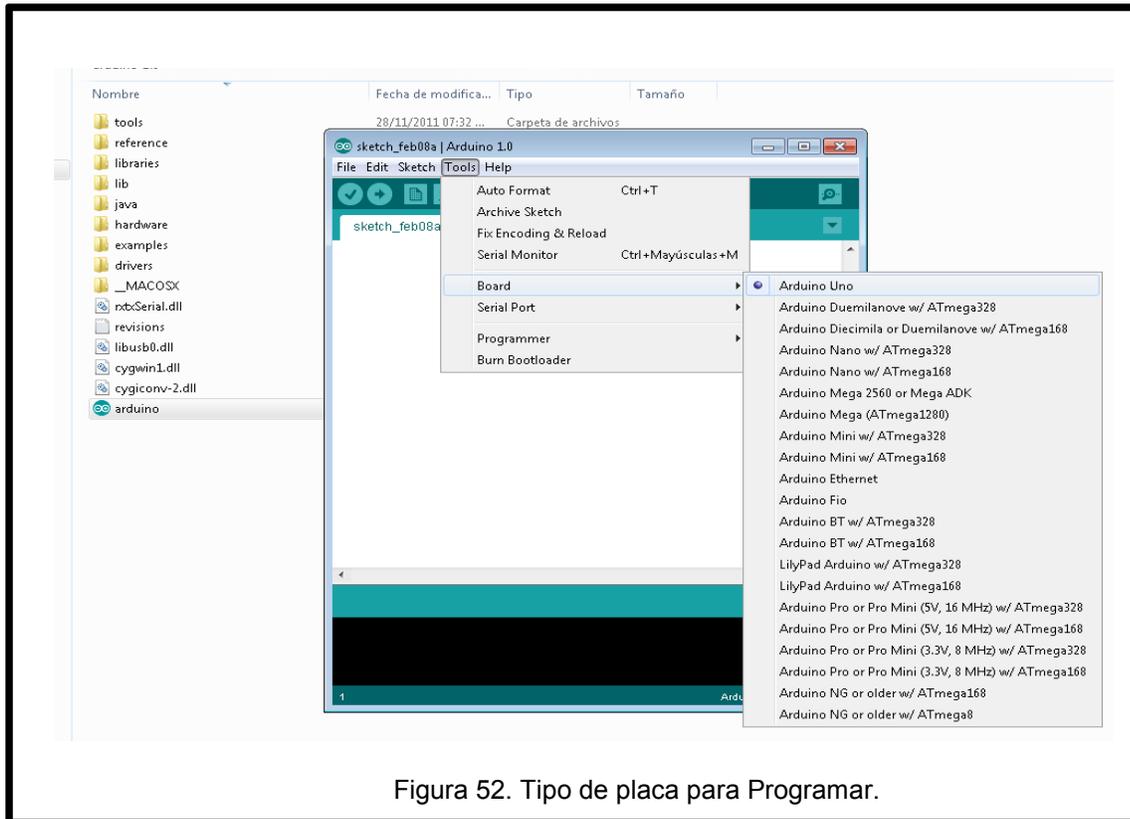


Figura 52. Tipo de placa para Programar.

PASO 4. SELECCIÓN DEL PUERTO DE COMUNICACIÓN

Ahora seleccionaremos el puerto de comunicación que estamos usando, igual nos vamos a tools, serial port y nos muestra las opciones en este caso nos dice que está conectado en el puerto COM7, lo seleccionamos (Figura 53).

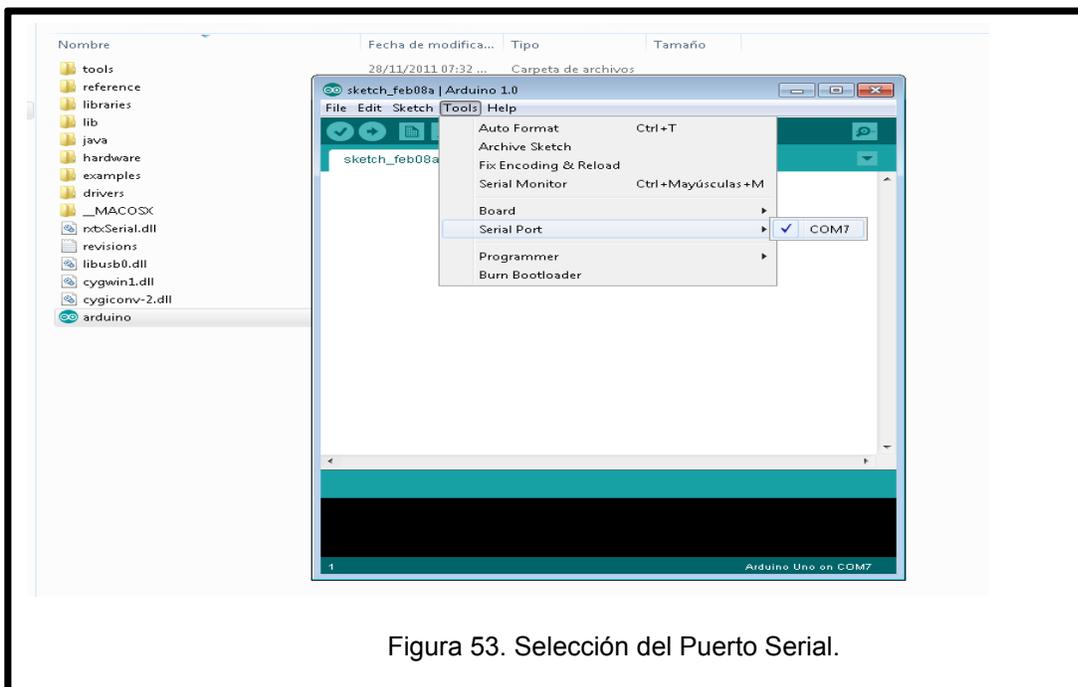


Figura 53. Selección del Puerto Serial.

A continuación procederemos a copiar y pegar el código anteriormente explicado a nuestro compilador como se muestra en la Figura 54.

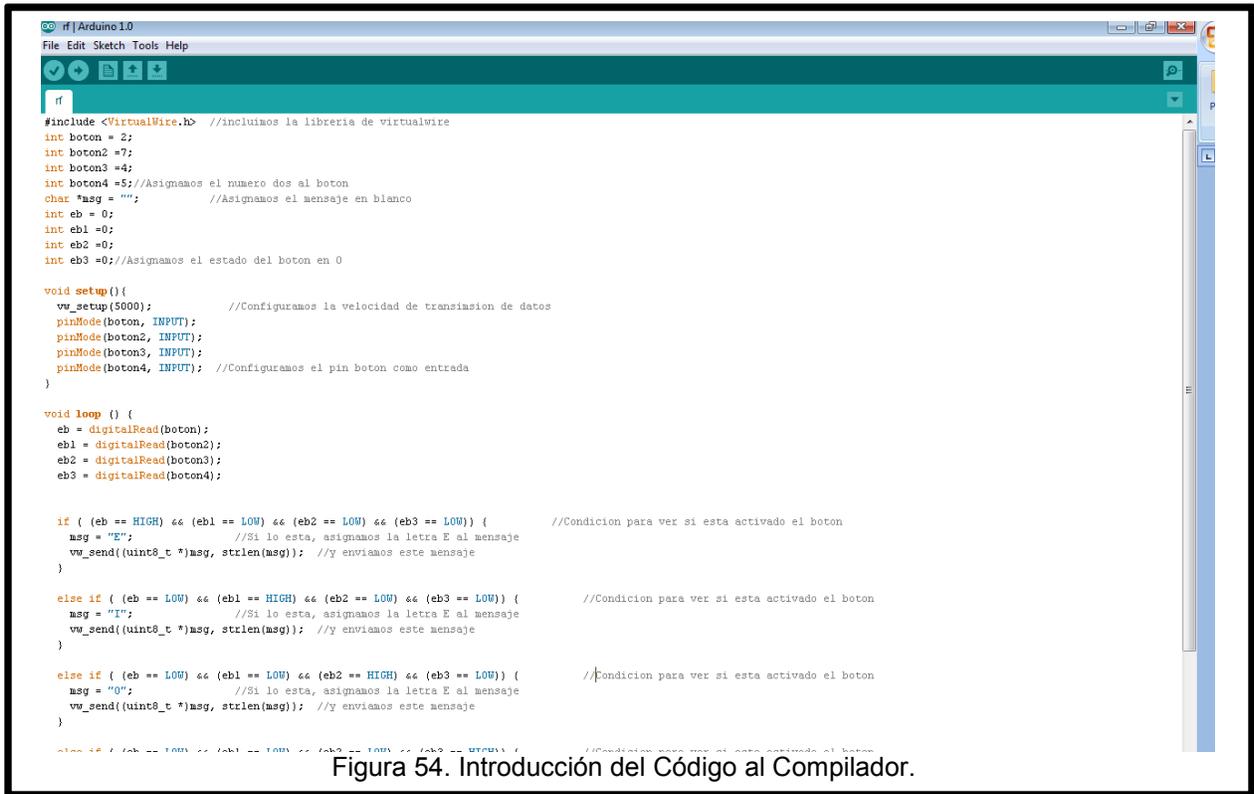


Figura 54. Introducción del Código al Compilador.

Una vez que copiamos el código lo compilaremos para ver que funcione correctamente, daremos en el botón verify. Figura 55.



Figura 55. Compilacion

Esperamos que acabe de compilar. Figura 56.

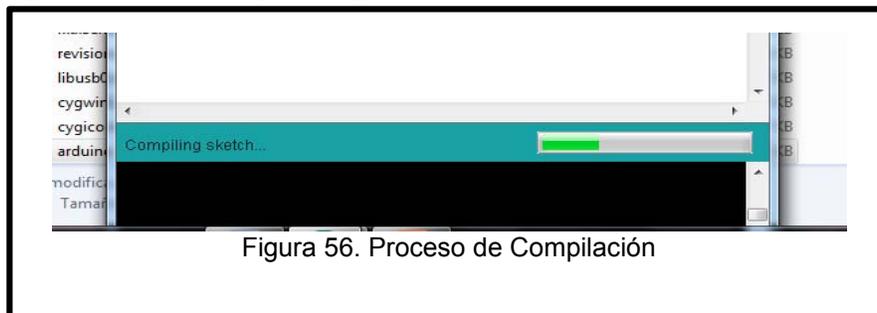


Figura 56. Proceso de Compilación

PASO 5. CARGAR CÓDIGO AL ARDUINO

Una vez compilado ahora el siguiente paso será cargarlo al microcontrolador damos click en el botón upload y esperamos a que se transfiera el código (Figura 57).

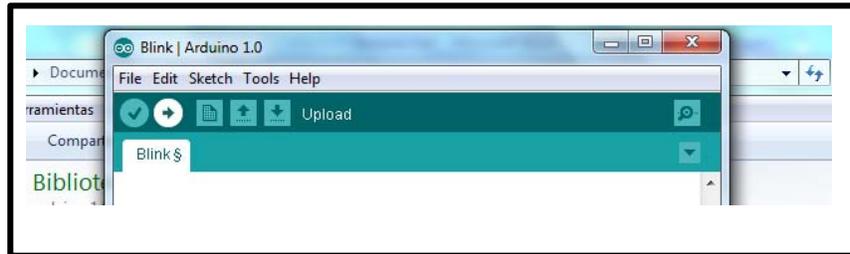


Figura 57. Transferencia del Código a la placa Arduino.

PASO 6. ARMADO CIRCUITO RECEPTOR

Ahora nuestra placa Arduino ya está lista para transmitir datos a una placa receptora nuestro siguiente paso será armar el circuito del diagrama receptor a continuación en la Figura 58 se mostrará el armado integrado de la placa receptora.

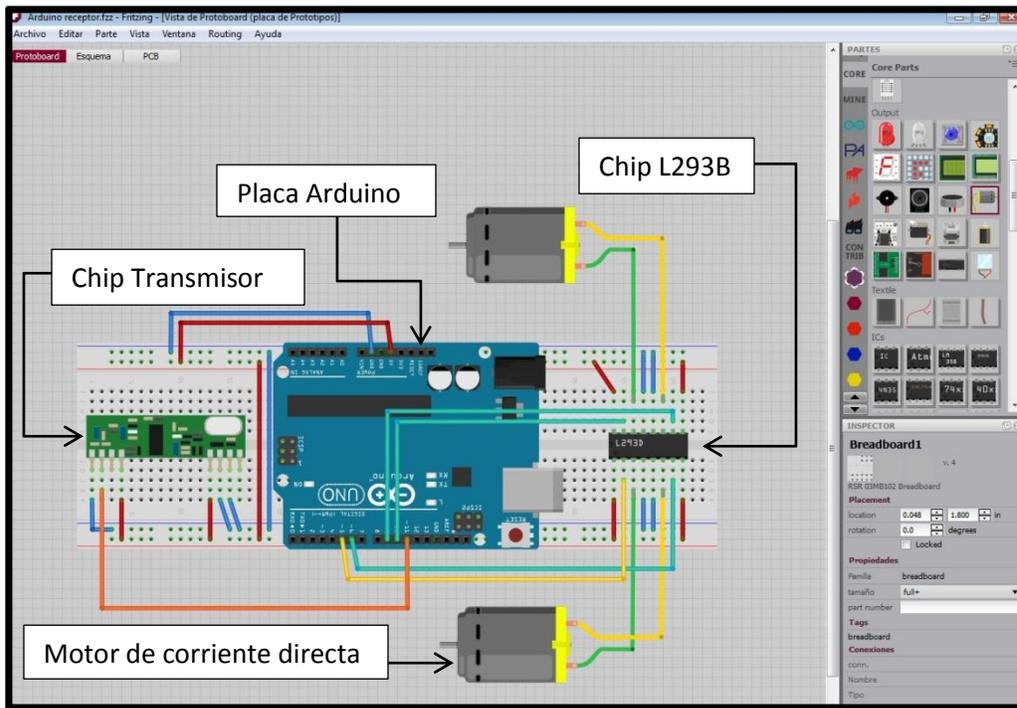


Figura 58. Diseño Virtual de Comunicación del Receptor.

Este es el segundo diseño (Figura 59) que usaremos para lograr la comunicación a través de radiofrecuencia en este ejemplo se muestran los circuitos de la placa receptora el cual recibirá los mensajes.

Una vez armado el circuito el siguiente paso será cargar el código de programación de Arduino que estamos usando como receptor, que se mostrará y explicará a continuación:

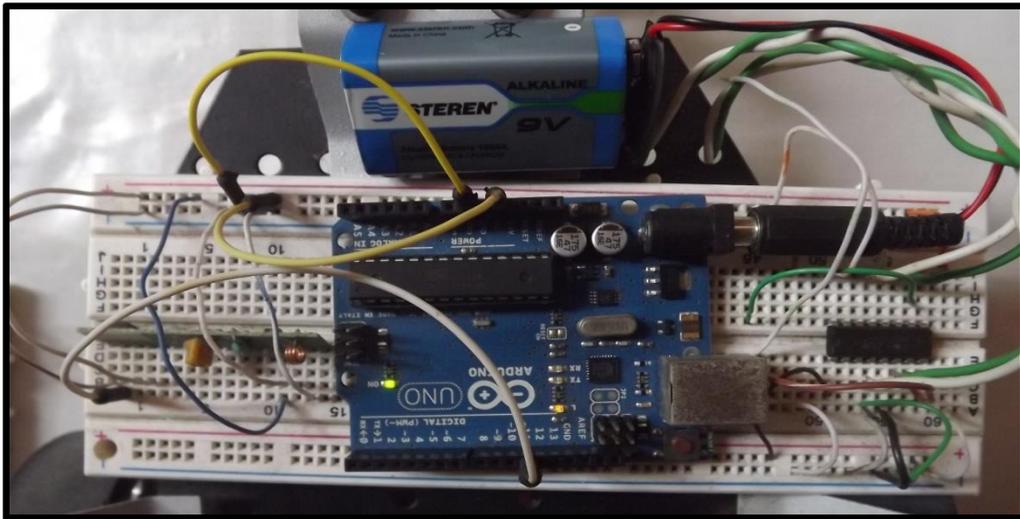


Figura 59. Montado Electrónico del Proyecto de la placa de recepción de datos.

El código que se presentará es para establecer comunicación entre la placa arduino transmisora y la placa arduino receptora usando la interfaz gráfica de la computadora, la función de la placa receptora es leer los mensajes recibidos del transmisor.

PASO 7. PROGRAMACIÓN DEL CIRCUITO RECEPTOR

```
#include<VirtualWire.h>

// Declaramos los pines de salida a utilizar

int en1 = 5;
int en2 = 6;
int in1 = 7;
int in2 = 8;
int in3 = 9;
int in4 = 10;
intmotorPin[] = {5, 6, 7, 8, 9, 10};
```

En este primer paso lo que haremos será declarar los pines a utilizar del Arduino que servirán para activar el chip L293B así también los motores.

```
void setup()
{
  Serial.begin(9600);
  Serial.println("setup"); // Inicializamos el puerto serie
  vw_setup(2000); // Bits por segundo
  vw_rx_start(); // Se avilita el receptor
```

Inicializamos el puerto serie y activamos el receptor para recibir mensajes del transmisor.

```
for (int i = 0; i < 6; i++) // Creamos un loop para declarar todos los pines
{
  pinMode(motorPin[i], OUTPUT); // Indicamos que los pines sean de salida
}
```

En esta sección usaremos una estructura de control usando for, esto para indicar que los pines sean de salida.

```
void loop()
{
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
```

Usaremos esta instrucción para recibir el dato mandado del transmisor.

```
if (vw_get_message(buf, &buflen))
{
    int i;
    digitalWrite(13, true); // Se enciende un led para indicar que se recibido el mensaje
```

Al momento de recibir el mensaje usaremos un LED este nos indicara que el mensaje se recibió correctamente.

```
for (i = 0; i <buflen; i++)// Hacemos un loop para leer el caracter recibido
```

```
{
```

```
    Serial.print(buf[i]);
    if(buf[i] == 'f') // Si el mensaje es f
    {
        forward();// Va hacia adelante
    }
```

Dependiendo del mensaje que recibamos, ejecutaremos la instrucción correspondiente. En este caso el mensaje es "f" lo que significa que el robot ira hacia adelante.

```
if(buf[i] == 'b') // Si el mensaje es b
```

```
{
```

```
backward();// Va hacia atras
```

```
}
```

```
if(buf[i] == 's') // Si el mensaje es s
```

```
{
```

```
stopMotor();// Se detienen los motores
```

```
}
```

```
if(buf[i] == 'l') // Si el mensaje es l
```

```
{
```

```
left();// Va hacia la izquierda
```

```
}
```

```
if(buf[i] == 'r') // Si el mensaje es r
```

```

    {
right();// Va hacia la derecha
    }
}

digitalWrite(13, false); // Apagamos el led una vez recibido el mensaje completo
}
}

```

// Set de instrucciones

```

void forward()
{
digitalWrite(en1,HIGH);// Instrucción que indica que el robot vaya hacia adelante
digitalWrite(en2,HIGH);
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);
}

```

En esta sección pondremos nuestro set de instrucciones que realizara el robot, adelante, reversa, izquierda, derecha y detener.

```

void backward()
{
digitalWrite(en1,HIGH);// Instrucción que indica que el robot vaya de reversa
digitalWrite(en2,HIGH);
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
}

```

```
void left()
{
digitalWrite(en1,HIGH);// Instrucción que indica que el robot vaya hacia la izquierda
digitalWrite(en2,HIGH);
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
}
void right()
{
digitalWrite(en1,HIGH);// Instrucción que indica que el robot vaya hacia la derecha
digitalWrite(en2,HIGH);
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);
}
void stopMotor()
{
digitalWrite(en1,HIGH);// Instrucción que indica que el robot se detenga
digitalWrite(en2,HIGH);
digitalWrite(in1,LOW);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,LOW); }
```

El código que se pondrá a continuación es para comunicar el robot mediante el control remoto independiente que se armó en el protoboard.

```
#include <VirtualWire.h>

//declaramos los pines a utilizar

int en1 = 5;

int en2 = 6;

int in1 = 7;

int in2 = 8;

int in3 = 9;

int in4 = 10;

int motorPin[] = {5, 6, 7, 8, 9, 10};
```

En esta sección declaramos los pines del arduino a utilizar.

```
void setup()

{

  Serial.begin(9600);

  Serial.println("setup");

  // inicializamos el Puerto serie

  vw_setup(2000); // Bits por Segundo

  vw_rx_start(); // activamos el receptor
```

Esta sección es para inicializar el puerto serie y activar el receptor.

```
for (int i = 0; i < 6; i++)

{

  pinMode(motorPin[i], OUTPUT); //declaramos que los pines sean de salida

}
```

Los pines declarados indicamos que sean de salida.

```
void loop()
{
  uint8_t buf[VW_MAX_MESSAGE_LEN];
  uint8_t buflen = VW_MAX_MESSAGE_LEN;
```

Instrucción para recibir el mensaje.

```
if (vw_get_message(buf, &buflen) // si recibimos correctamente el mensaje
{
  int i; // almacenamos el mensaje recibido

  digitalWrite(13, true); // encendemos un led para ver transmision
```

Si recibimos el mensaje se almacena y se enciende un led.

```
  for (i = 0; i < buflen; i++)
  {
    Serial.print(buf[i]);

    if(buf[i] == '1') // si el mensaje recibido es un 1
    {
      forward(); // ir hacia adelante
      Serial.println(" = forward");
    }

    if(buf[i] == '2') // si el mensaje recibido es un 2
    {
      backward(); //ir hacia atras
      Serial.println(" = backward");
    }

    if(buf[i] == '3') //si el mensaje recibido es un 3
```

```

{
  left(); // ir izquierda
  Serial.println(" = left");
}
if(buf[i] == '4') //si el mensaje recibido es un 4
{
  right(); //ir derecha
  Serial.println(" = right");
}
if(buf[i] == '5') //si el mensaje recibido es un 5
{
  stopMotor(); //detener motores
  Serial.println(" = stopped");
}

  Serial.print(" ");
}
  Serial.println("");
  digitalWrite(13, false); // apagamos el led
}
}

```

```
void forward() // instruccion de ir hacia adelante
```

```
{  
  digitalWrite(en1,HIGH);  
  digitalWrite(en2,HIGH);  
  digitalWrite(in1,HIGH);  
  digitalWrite(in2,LOW);  
  digitalWrite(in3,LOW);  
  digitalWrite(in4,HIGH);  
}
```

```
void backward() // instruccion de ir hacia atras
```

```
{  
  digitalWrite(en1,HIGH);  
  digitalWrite(en2,HIGH);  
  digitalWrite(in1,LOW);  
  digitalWrite(in2,HIGH);  
  digitalWrite(in3,HIGH);  
  digitalWrite(in4,LOW);  
}
```

```
void left() // instruccion de ir izquierda
```

```
{  
  digitalWrite(en1,HIGH);  
  digitalWrite(en2,HIGH);  
}
```

```
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);

}

void right() // instruccion de ir derecha
{
digitalWrite(en1,HIGH);
digitalWrite(en2,HIGH);
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
}

void stopMotor() // detener motores
{
digitalWrite(en1,HIGH);
digitalWrite(en2,HIGH);
digitalWrite(in1,LOW);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,LOW);
}
```

Ahora procederemos hacer los mismos pasos que hicimos anteriormente con la placa de transmisión para cargar nuestro código de receptor en otra placa arduino y así establecer la comunicación en nuestras 2 placas arduino.

Una vez que se cargue el código, el proyecto estará finalizado y listo para hacer pruebas de radiofrecuencia.

En la Figura 60 se muestra el robot terminado ahora para que podamos controlarlo crearemos una interfaz gráfica con nuestra computadora usando el entorno de desarrollo Visual Studio 2010.

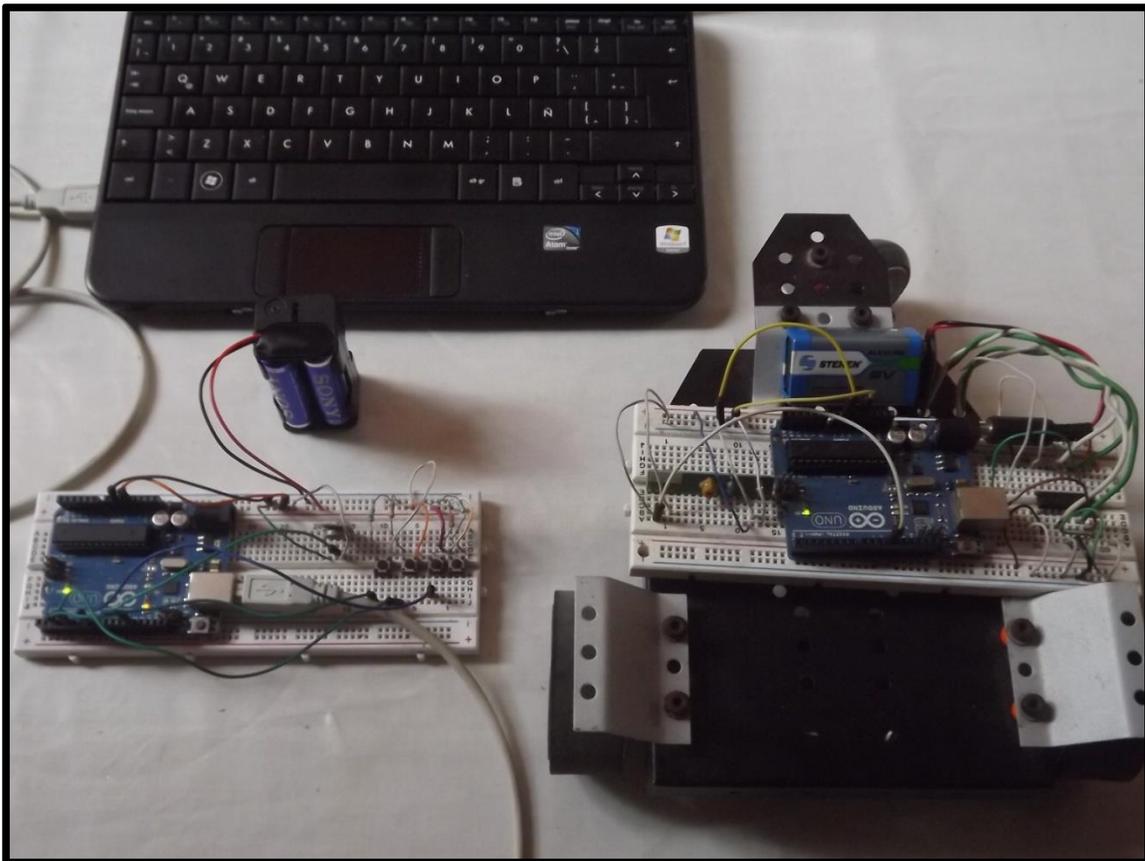


Figura 60. Montaje de placas de Transmisión y Recepción.

3.2 DESARROLLO DE INTERFAZ GRÁFICA

Después de haber programado la placa Arduino como se explicó en el capítulo anterior a continuación desarrollaremos una interfaz utilizando visual studio 2010 en el cual utilizaremos el lenguaje de programación C#.

Empezaremos abriendo nuestro entorno de desarrollo Figura 61 creando un nuevo proyecto.

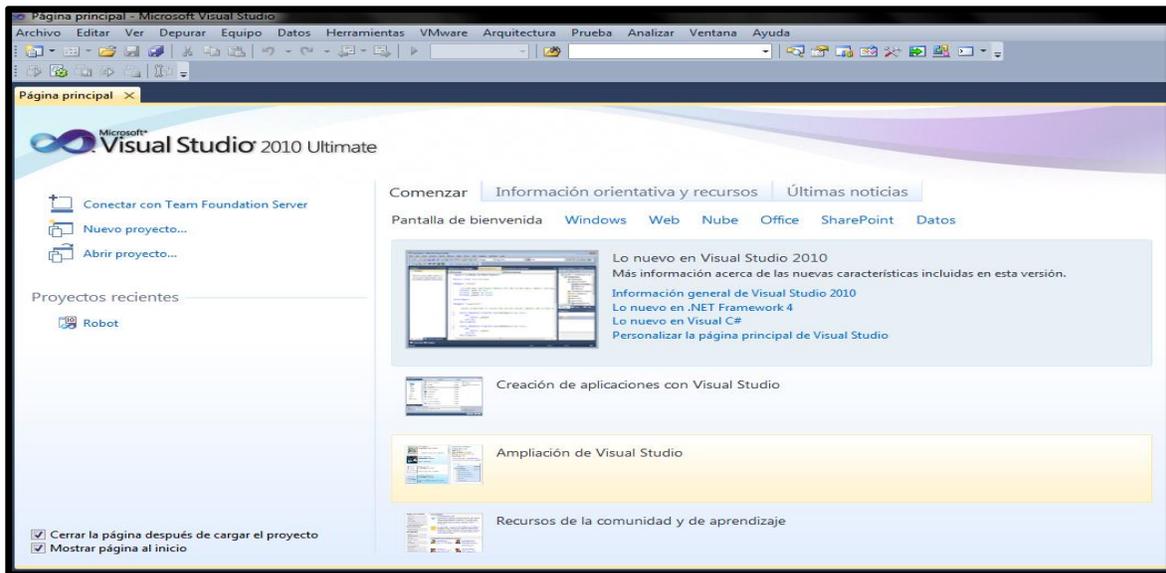


Figura 61. Ventana principal del entorno de desarrollo

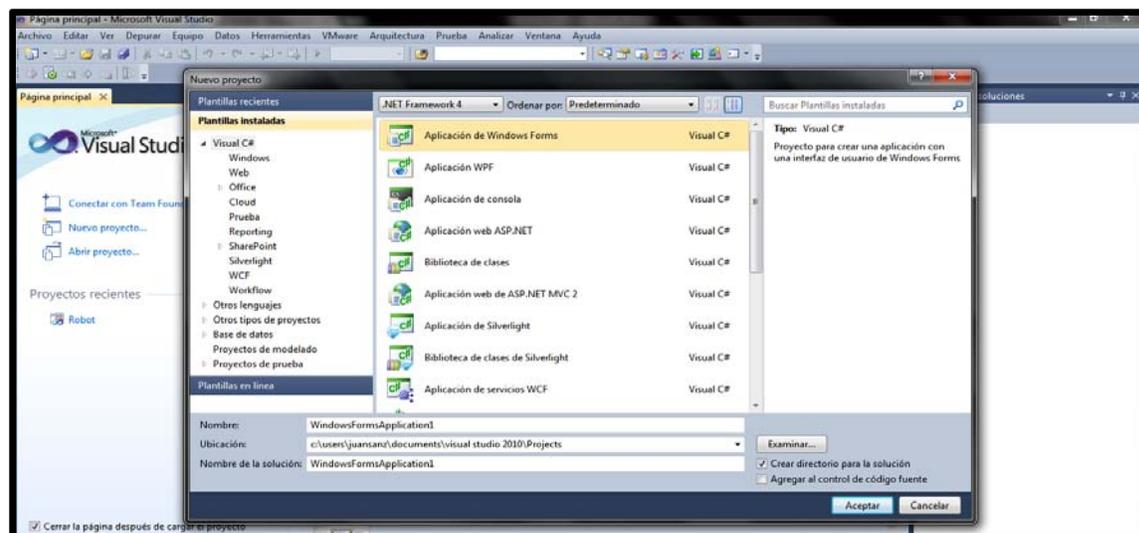


Figura 62. Selección de Aplicación de Windows Forms

Una vez abierto nuestro nuevo proyecto de Windows Forms seleccionaremos en nuestro cuadro de herramientas la opción de serial port (Figura 63) esto para establecer comunicación con nuestra placa arduino mediante el puerto USB.

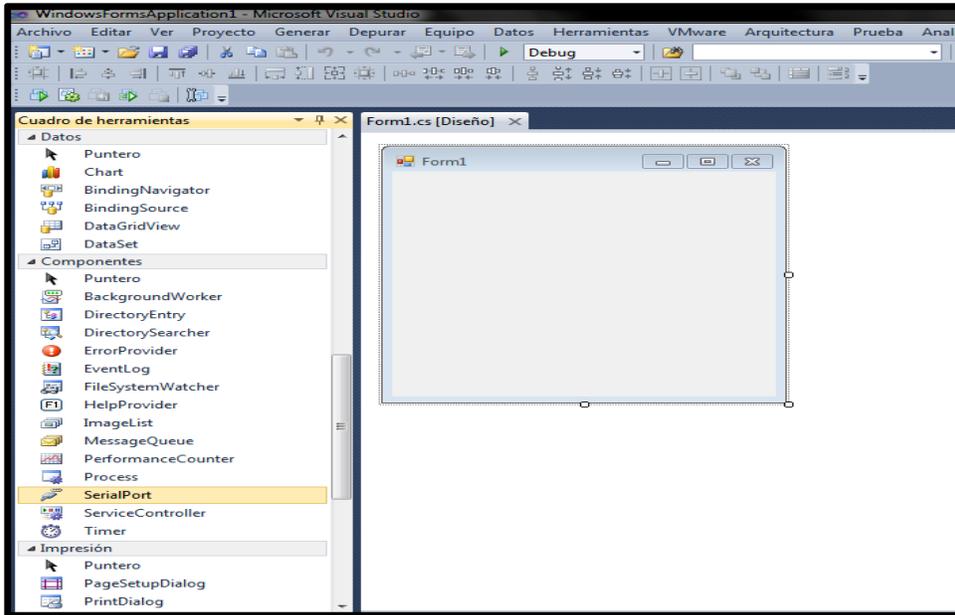


Figura 63. Selección del componente puerto serial

A continuación abriremos nuestro código fuente para implementar las siguientes librerías faltantes para el buen funcionamiento del programa (Figura 64).

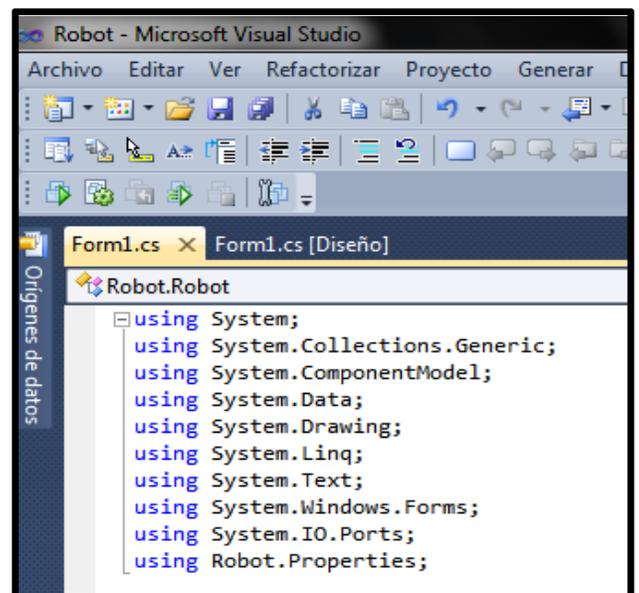
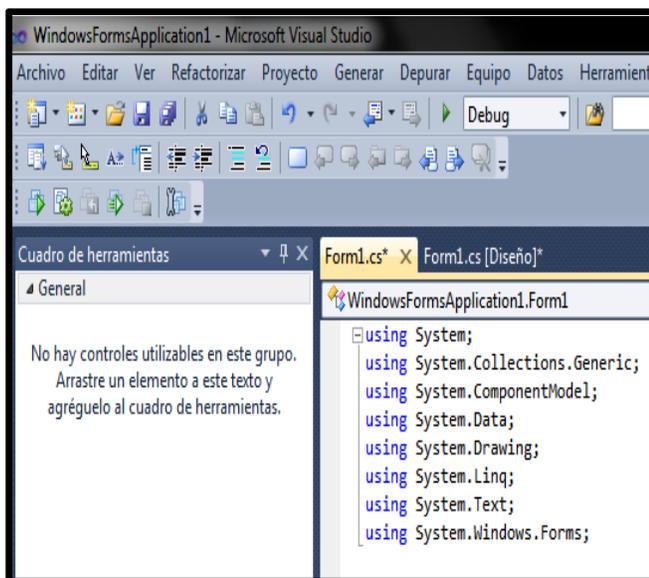


Figura 64. Librerías necesarias

Lo siguiente que haremos será implementar botones en nuestra ventana de aplicaciones.

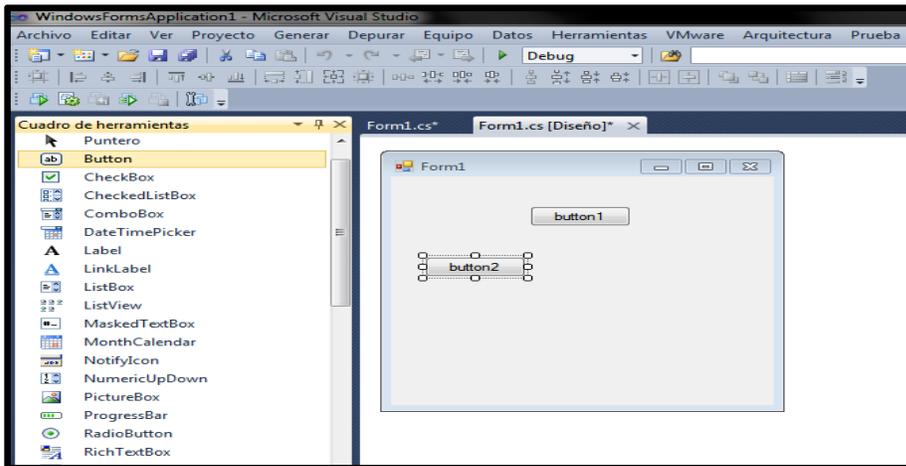


Figura 65. Implementación de botones

PASO 8. PROGRAMACIÓN DEL PUERTO SERIE CON C#

Para poder establecer una comunicación a través de nuestro puerto USB necesitaremos implementar el siguiente código:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports; ← Esta es la librería que necesitamos para
using Robot.Properties;                                     usar las propiedades del puerto serial.

namespace Robot
{
    public partial class Robot : Form
    {
        public Robot()
        {
            InitializeComponent();
            serialPort1.PortName = "COM4"; // aquí pondremos el Puerto que esta
            serialPort1.BaudRate = 9600; //usando la placa arduino para comunicación
            serialPort1.Open(); // se indica que el puerto está abierto
        }
    }
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (serialPort1.IsOpen) serialPort1.Close(); ← Abrimos y cerramos el
}                                                         puerto serie cada vez que lo
                                                         usemos.
```

PASO 9. PROGRAMACIÓN DEL SISTEMA DE CONTROL

En cada botón pondremos un código el cual servirá para mandar datos a la placa transmisora del arduino.

```
Privatevoid Adelante_Click(object sender, EventArgs e)
{
    serialPort1.Write("f"); //este es el dato que se envía a la placa
    Adelante.Enabled = false; // transmisora arduino
    Atras.Enabled = true;
    Izquierda.Enabled = true;
    Derecha.Enabled = true;
    Detener.Enabled = true;
}
```

Le pondremos el nombre de acuerdo a la función que realiza.

Para poder controlar más eficazmente nuestro robot utilizaremos nuestros cursores de flechas del teclado, a continuación explicaremos el código para enlazar el teclado con nuestra interfaz y tener un mejor control de nuestro robot para realizar óptimamente su desempeño.

```
Privatevoid Robot_Load(object sender, EventArgs e)
{
    foreach (Control control in this.Controls)
    {
        control.PreviewKeyDown += new PreviewKeyDownEventHandler(control_PreviewKeyDown);
    }
}
```

```
void control_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
{
    if (e.KeyCode == Keys.Up)
    {
        serialPort1.Write("f");
        Adelante.Enabled = false;
        Atras.Enabled = true;
        Izquierda.Enabled = true;
        Derecha.Enabled = true;
        Detener.Enabled = true;
        e.IsInputKey = true;
    }
}
```

Esta función es para establecer que cursor utilizaremos del teclado.

Al igual se envía el dato que recibirá la placa arduino.

3.3 PROGRAMACIÓN EN ANDROID

En esta sección explicaremos la programación en android para el control de robot por el celular lo siguiente que haremos será abrir nuestro navegador y poner ai2.appinventor.mit.edu nos abrirá una página la cual le daremos en el botón créate y nos mostrara la página de desarrollo como se muestra en la Figura 66. Como se puede apreciar en la imagen en la parte izquierda tiene una serie de herramientas que usaremos para hacer el diseño y comunicación.

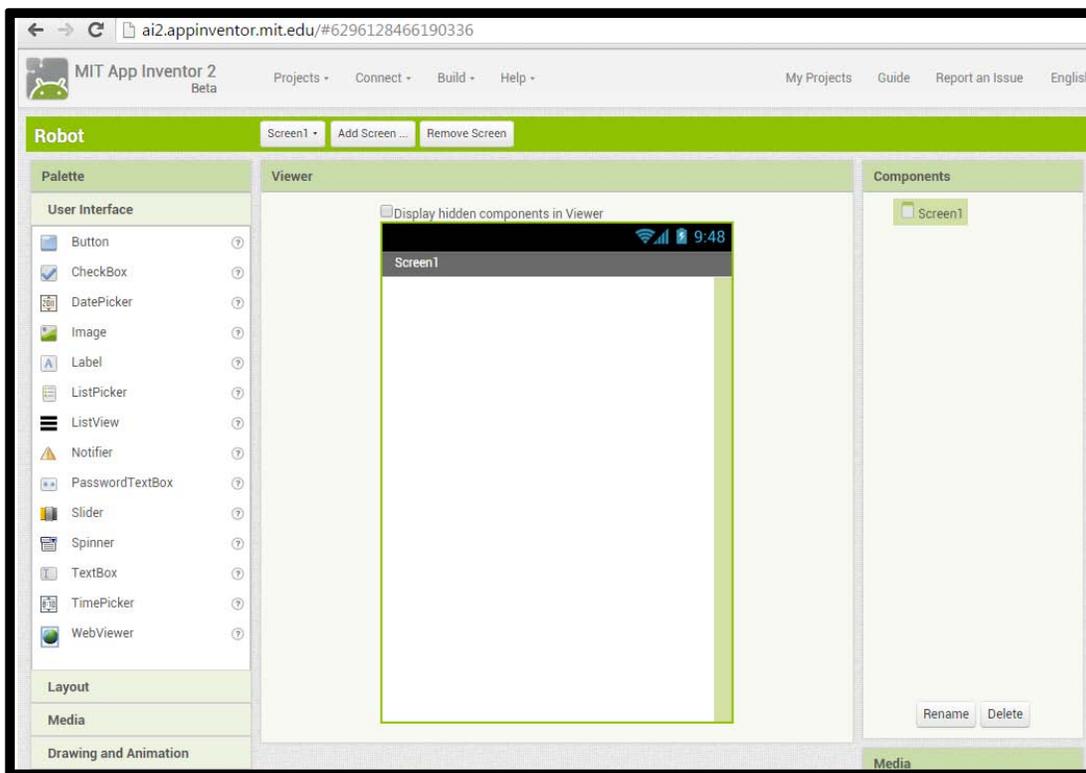


Figura 66. Entorno de desarrollo MIT App Inventor.

PASO 1. Con nuestra sección de herramientas del lado izquierdo procederemos a implementar botones Figura 67 que nos servirán para comandar el robot a través del celular y un botón de conectar para que establezca comunicación con el robot.

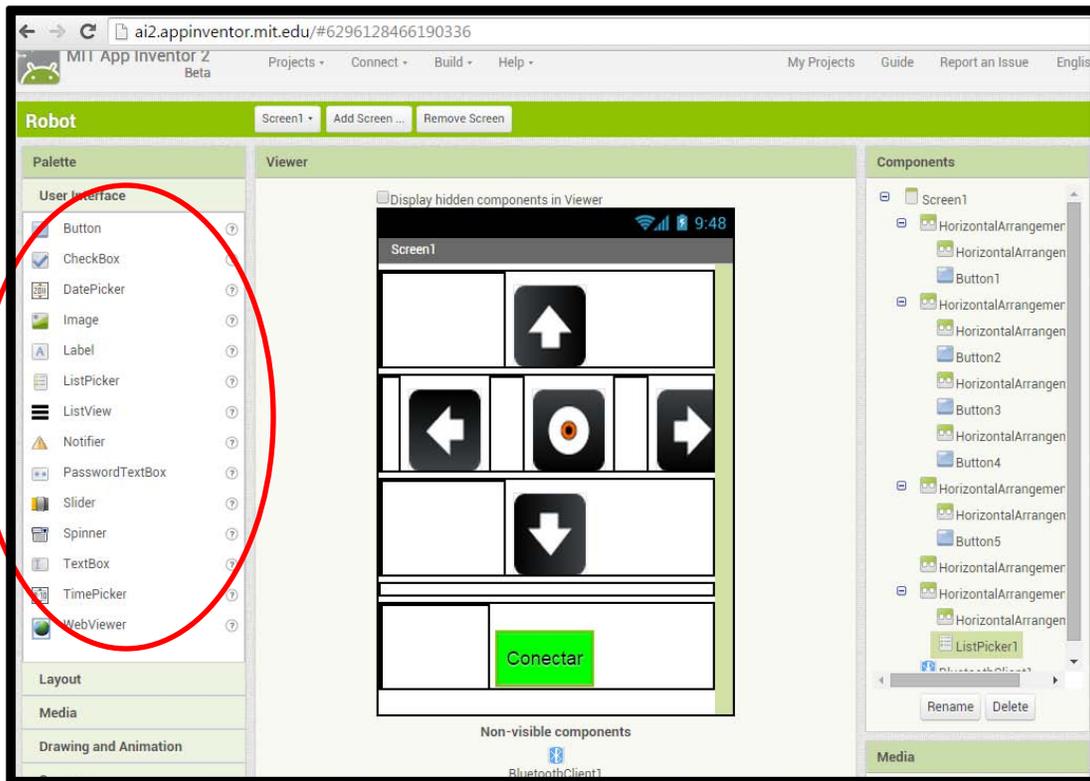


Figura 67. Implementación de botones en la aplicación

PASO 2. Una vez que tenemos el diseño de la aplicación ahora nuestro siguiente paso será programar cada uno de los botones implementados para eso pasaremos a nuestra sección de bloques como se muestra en la Figura 68. Una vez en esta sección seleccionaremos botón por botón para poner el bloque correspondiente a la función que va a realizar.

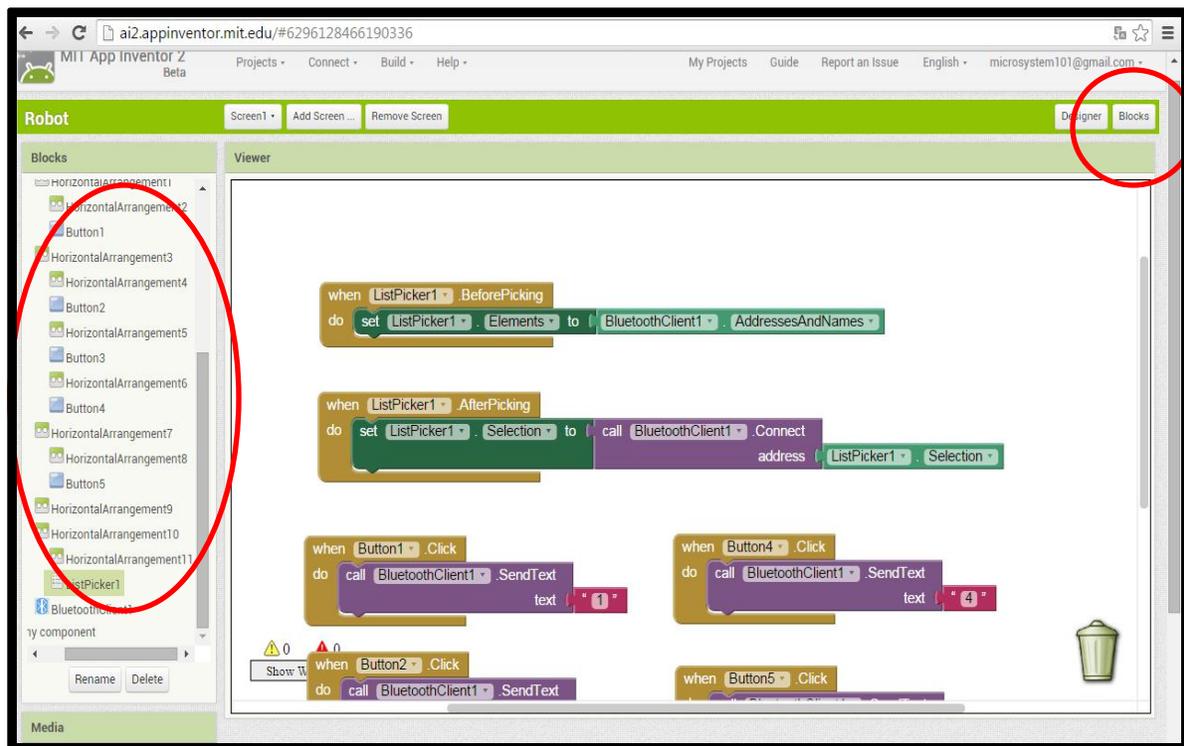


Figura 68. Diseño de bloques para la programación de la aplicación

PASO 3. Ahora le indicaremos a cada botón que es lo que debe de enviar para que el robot sepa el tipo de instrucción que está recibiendo para eso le enviaremos números usando la instrucción sendtext Figura 69, con esto la aplicación podrá enviar mensajes de texto a través de puerto serial y así poder comunicarse con el robot.

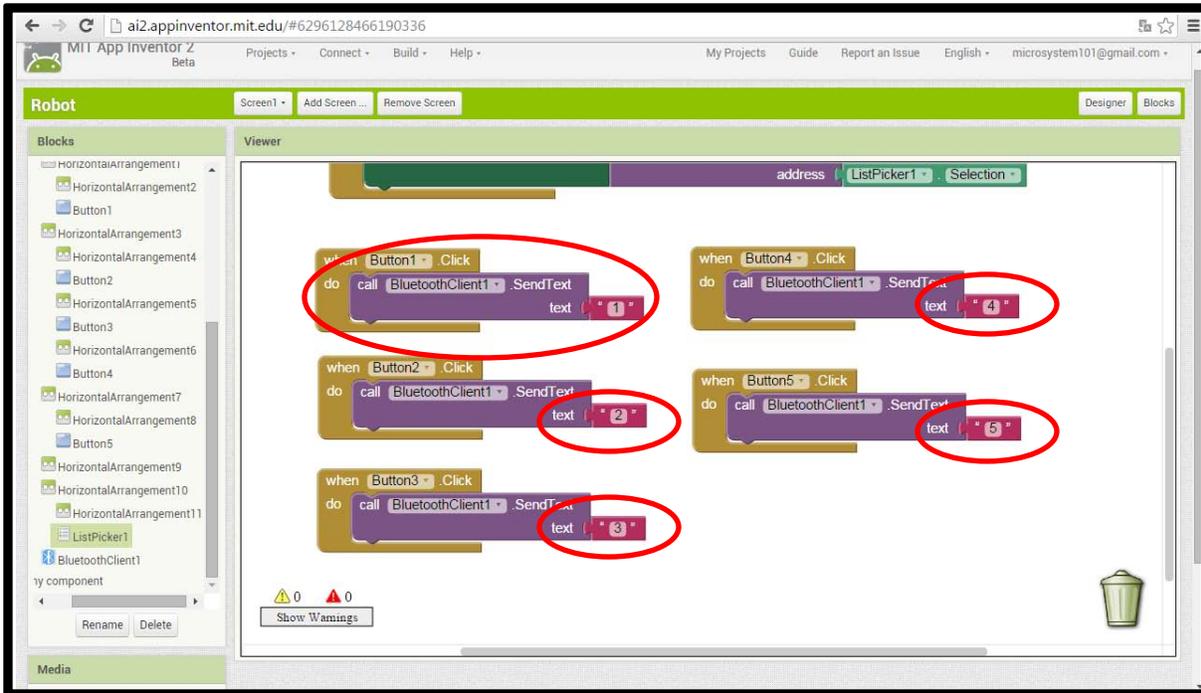


Figura 69. Instrucción para enviar datos por el puerto serial

Una vez que tenemos el diseño y la programación nuestro robot está listo para hacer pruebas de navegación y control.

3.4 CONECTAR CÁMARA IP AL NAVEGADOR

Para que podamos controlar nuestro robot con mayor eficacia a mayor distancia le colocaremos una cámara IP la cual nos ayudara a monitorear el entorno en el que se encuentra el robot Figura 70 y así el usuario sepa la ruta de acceso a cuevas, el riesgo de una casa para el ingreso de policías, descubrimiento de investigaciones arqueológicas, etc.

El procedimiento es el siguiente.

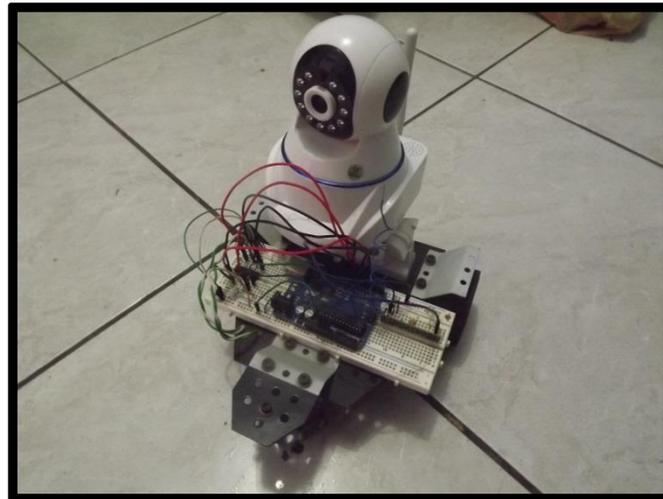


Figura 70. Robot monitoreado con cámara IP.

PASO 1. CONECTAR CÁMARA AL NAVEGADOR

Ahora lo siguiente que haremos será entrar al Internet Explorer y poner la dirección ip de la cámara para que podamos monitorear el robot a distancia (Figura 71).

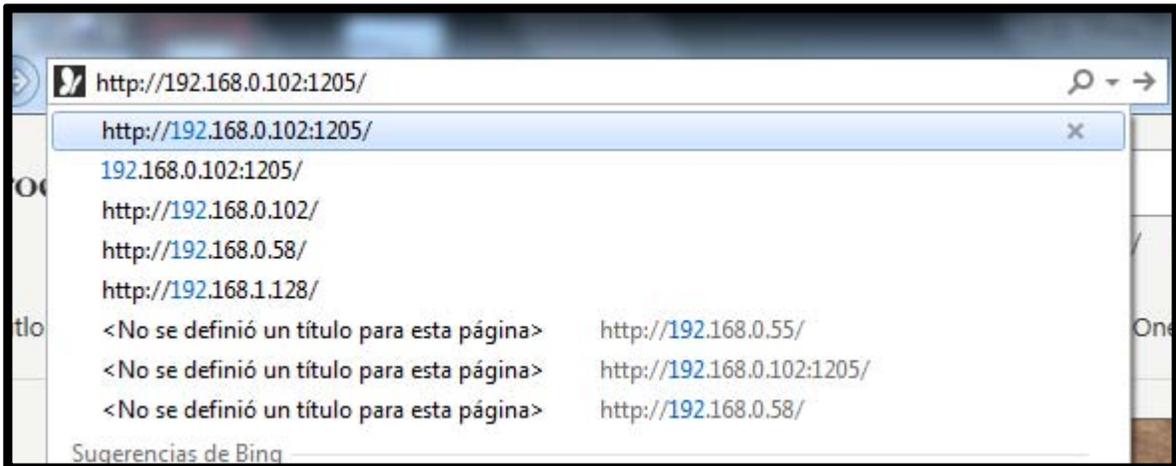


Figura 71. Dirección URL de cámara IP.

Una vez en la página nos pedirá que ingresemos nuestro nombre de usuario y contraseña, ingresamos los datos necesario y nos logueamos en la página (Figura 72).



Figura 72. Ingreso de usuario y contraseña de la cámara IP.

Una vez que nos muestre la cámara la imagen de su entorno (Figura 73) lo que haremos a continuación será ejecutar la interfaz gráfica para controlar el robot y la cámara.

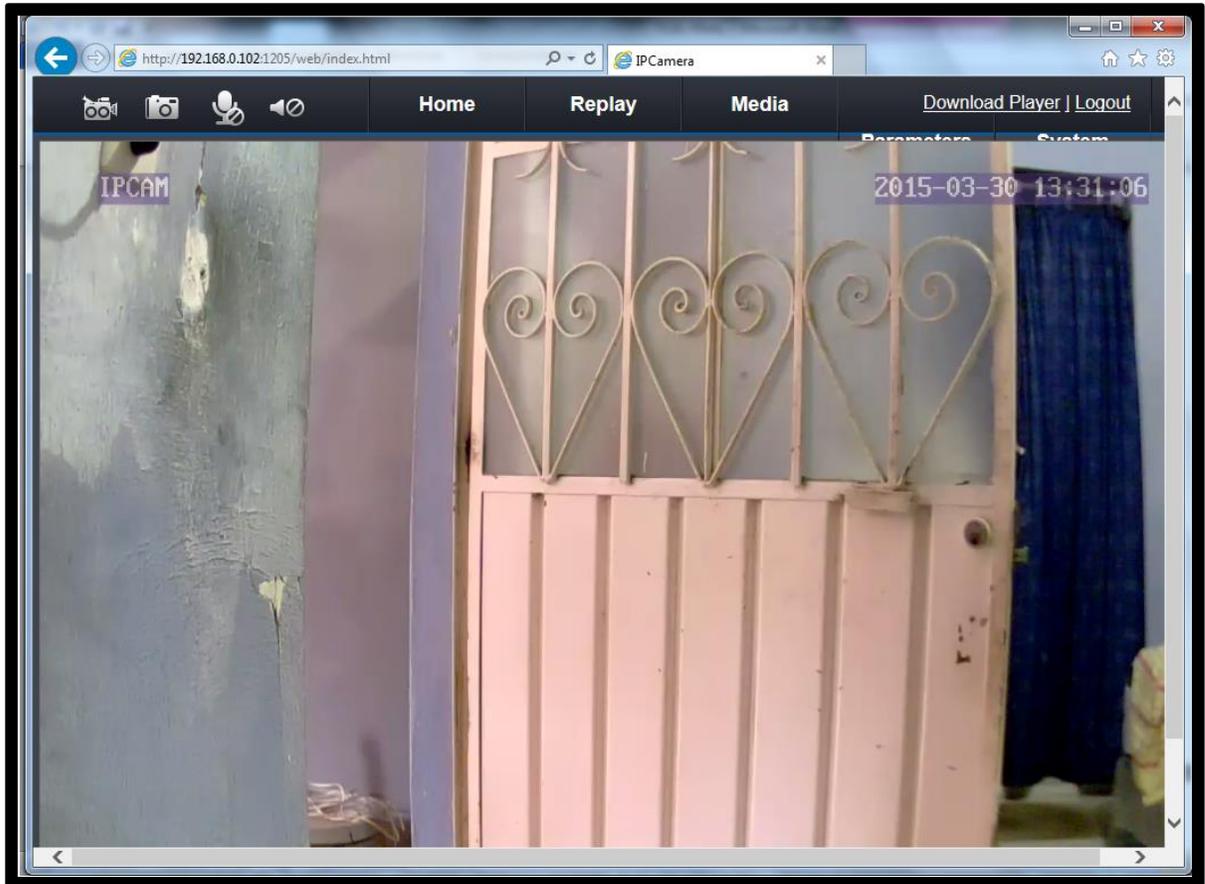


Figura 73. Monitoreo de la cámara IP.

3.5 CONECTAR CÁMARA IP AL CELULAR

Para poder monitorear nuestro robot usando el control remoto inalámbrico necesitaremos transmitir la imagen de la cámara ip a un celular a continuación se explicaran los pasos detallados a seguir para su funcionamiento.

Lo primero que haremos será poner la dirección ip de la cámara en el celular como se muestra en la Figura 74.

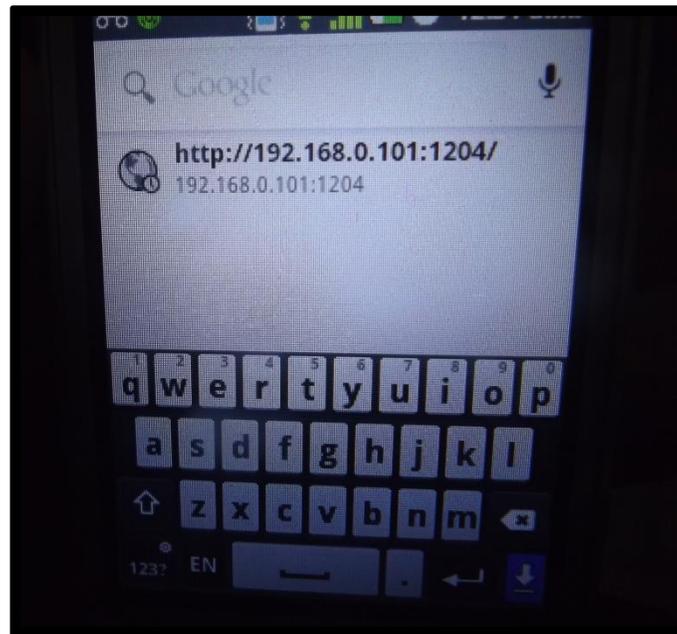


Figura 74. Dirección de cámara ip en el celular

A continuación nos mostrará la página principal en la cual tendremos que introducir el nombre de usuario y contraseña para poder conectarnos con la imagen de la cámara en la Figura 75 se muestra el siguiente paso a realizar.

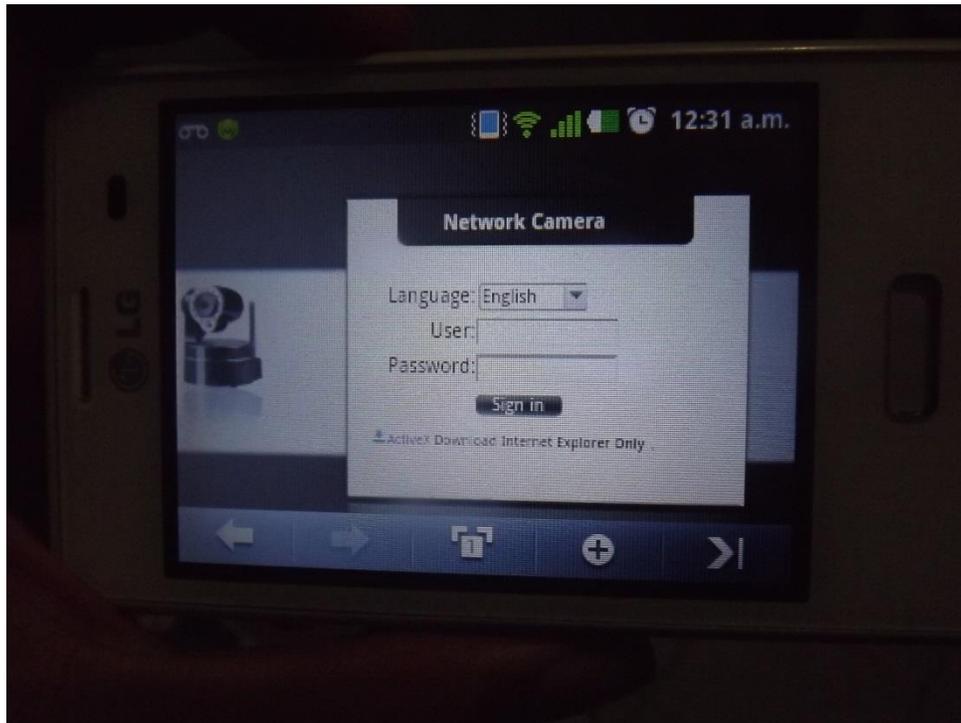


Figura 75. Introducción de usuario y contraseña

Después de introducir el usuario y contraseña nos mandará directamente la imagen en vivo de la cámara una vez que tengamos enlazada la cámara con el celular ahora el robot está listo para ser controlado a distancia por control remoto.

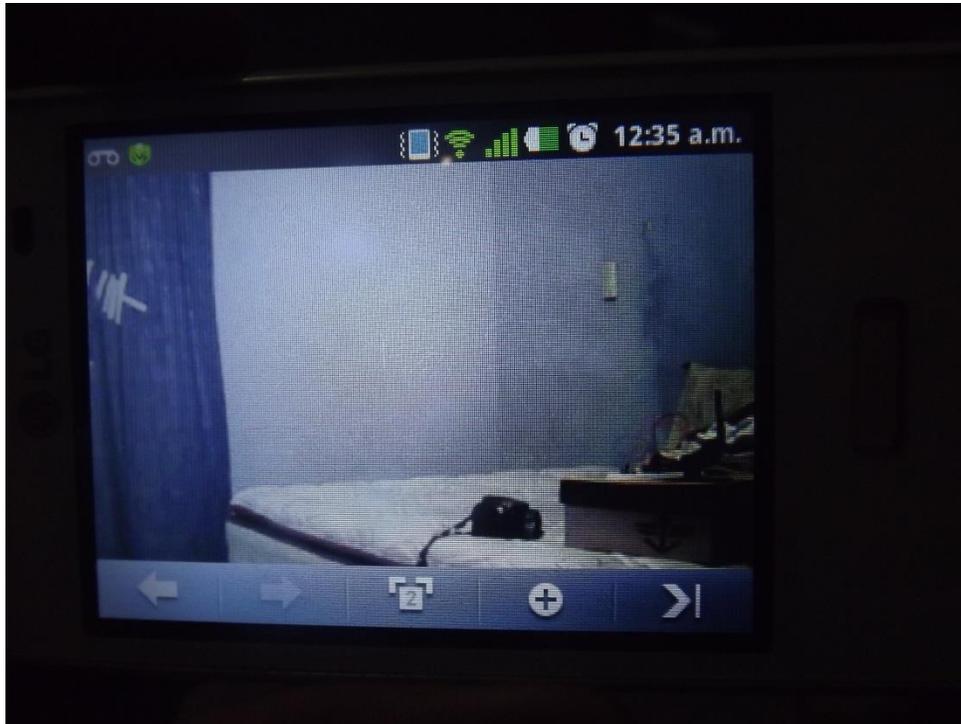


Figura 76. Transmisión en vivo de la cámara ip al celular

CAPÍTULO 4

Capítulo 4

RESULTADOS Y PRUEBAS

INTRODUCCIÓN

En este capítulo se explicara el funcionamiento del sistema de control del robot para probar la eficacia y rapidez de transmisión de datos con el medio de comunicación más usado que es la radiofrecuencia, así también se harán pruebas de monitoreo en vivo con la ayuda de una cámara IP la cual incluiremos en la interfaz para poderla controlar con mayor facilidad.

4.1 PRUEBAS DEL SISTEMA DE CONTROL

La imagen que está a continuación (Figura 77) muestra cómo debe quedar la interfaz una vez que el código este implementado.

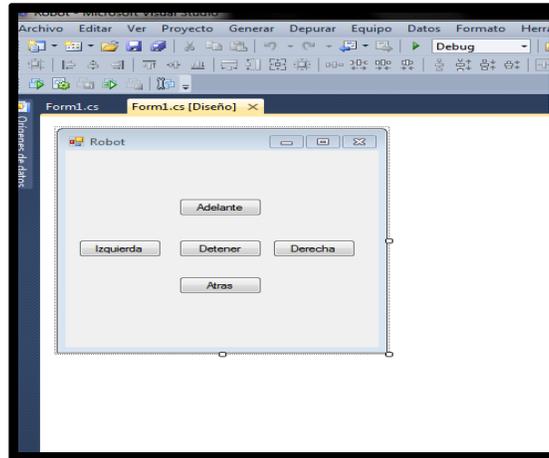


Figura 77. Interfaz de Control

Lo siguiente es hacer pruebas de comunicación y demostrar que la interfaz funcione correctamente con la placa emisora arduino, para eso debemos compilar el programa así como se muestra en la imagen (Figura 78) si compila correctamente quiere decir que estableció comunicación con el transmisor.

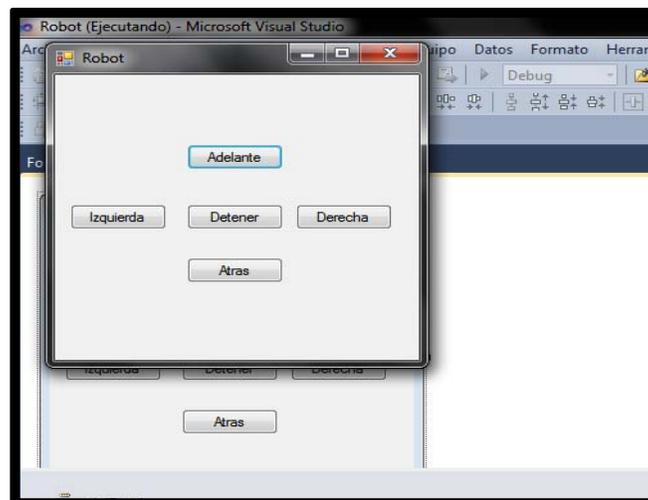


Figura 78. Interfaz establecida con el puerto serial

Una vez compilado y que el programa enlace comunicación a través del puerto serial tendremos la opción de controlar el robot ya sea por medio del cursor (mouse) o teclado (Figura 79).

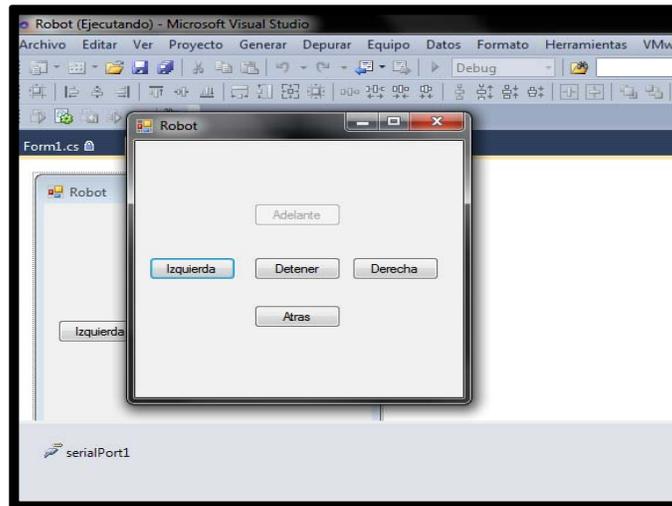


Figura 79. Envío de datos mediante el puerto serial

En la imagen vemos como el botón (adelante) entra en un false demostrando el correcto funcionamiento y envió de datos a la placa transmisora arduino.

Una vez terminada la interfaz y probemos el correcto funcionamiento del puerto serial podremos controlar el robot desde nuestra computadora.

A continuación probaremos nuestra interfaz terminada para establecer comunicación con nuestro robot. En la Figura 80 muestra cómo queda terminado nuestro proyecto y listo para ser ejecutado.

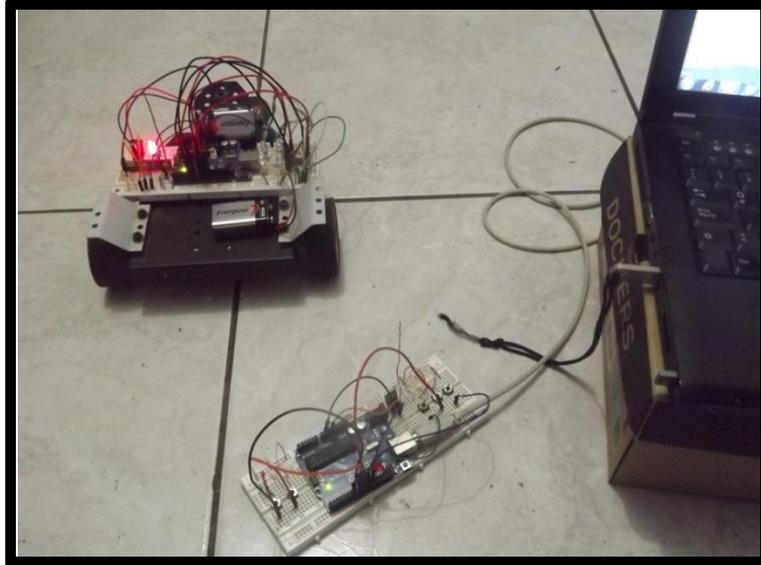


Figura 80. Comunicación por medio de radiofrecuencia.

Ahora procederemos a ejecutar una instrucción la cual será avanzar para asegurarnos de que se envían los datos correctamente para esto hemos puesto unos leds los cuales servirán como medio de indicación que se estableció una buena comunicación.

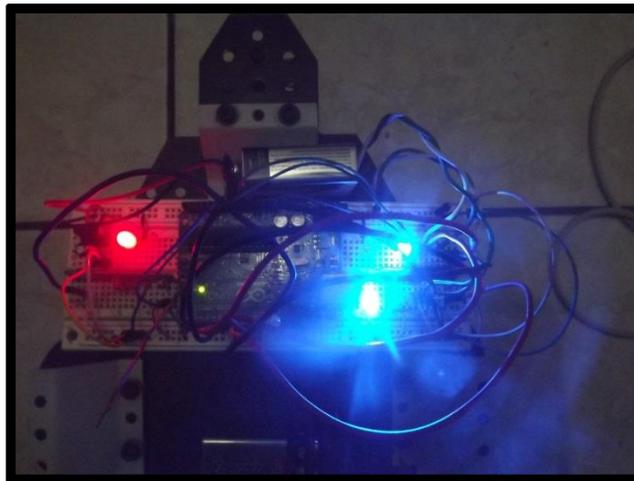


Figura 81. Encendido de leds para verificar transmisión de la interfaz de la CPU.

Como se puede observar en la Figura 81 vemos como se encienden 2 leds quiere decir que el mensaje de la placa arduino receptora llego de forma correcta los leds encendidos son superior derecho e inferior izquierdo el cual significa que está ejecutando la instrucción de ir hacia adelante.

Nuestro siguiente paso será indicarle al robot que vaya hacia la izquierda para eso tiene que prender solamente los leds superior izquierda inferior izquierda (Figura 82), para demostrar su funcionalidad lo presentaremos en la siguiente imagen.

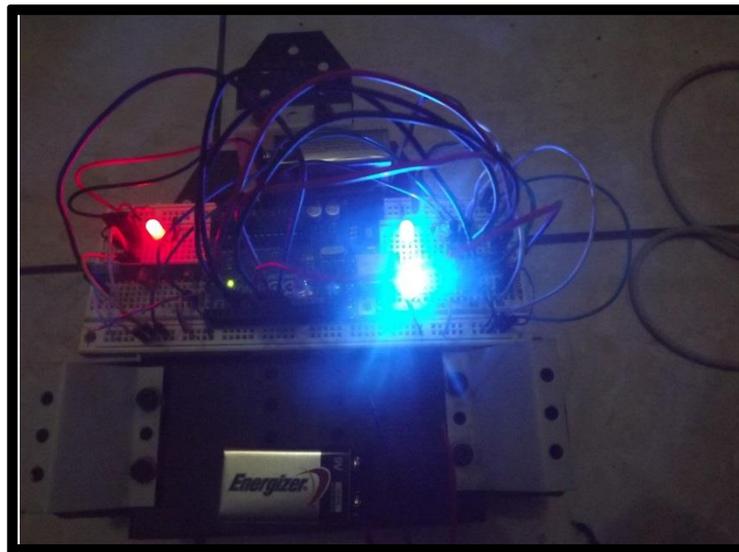


Figura 82. Instrucción de ir hacia la izquierda con indicadores leds.

Ahora como última instrucción indicaremos que se detenga, el cual mostrará todos los leds apagados (Figura 83).

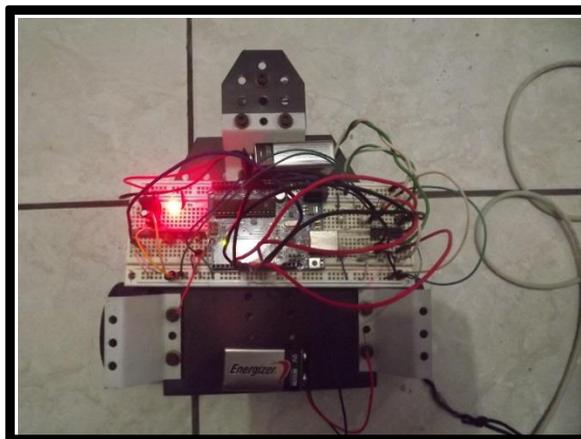


Figura 83. Apagado de leds indica detener motores.

4.2 PRUEBAS DEL SISTEMA DE MONITOREO

Para poder controlar nuestra cámara IP con la interfaz de sistema de monitoreo tendremos que ingresar como se muestra en la Figura 84 la misma dirección IP que usamos para el Internet Explorer, así también el mismo usuario y misma contraseña damos click en el botón que dice conectar y esperamos unos segundos para que se conecte la cámara con la interfaz.

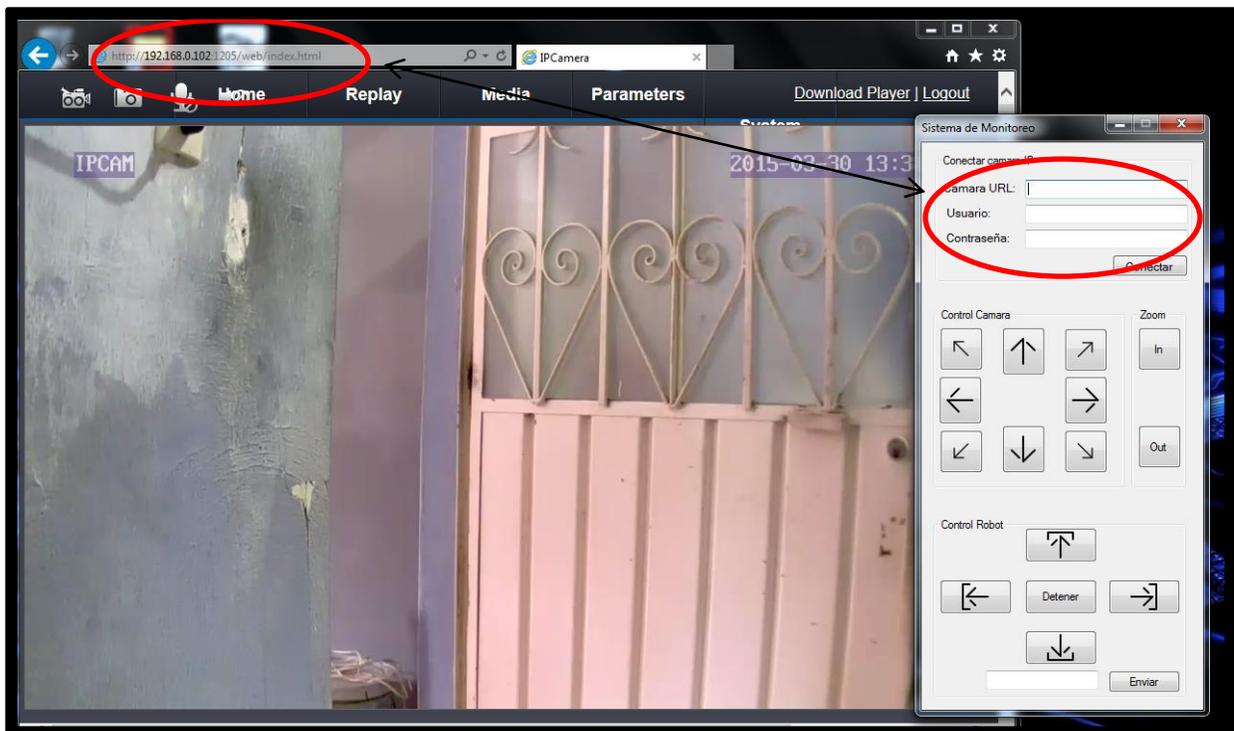
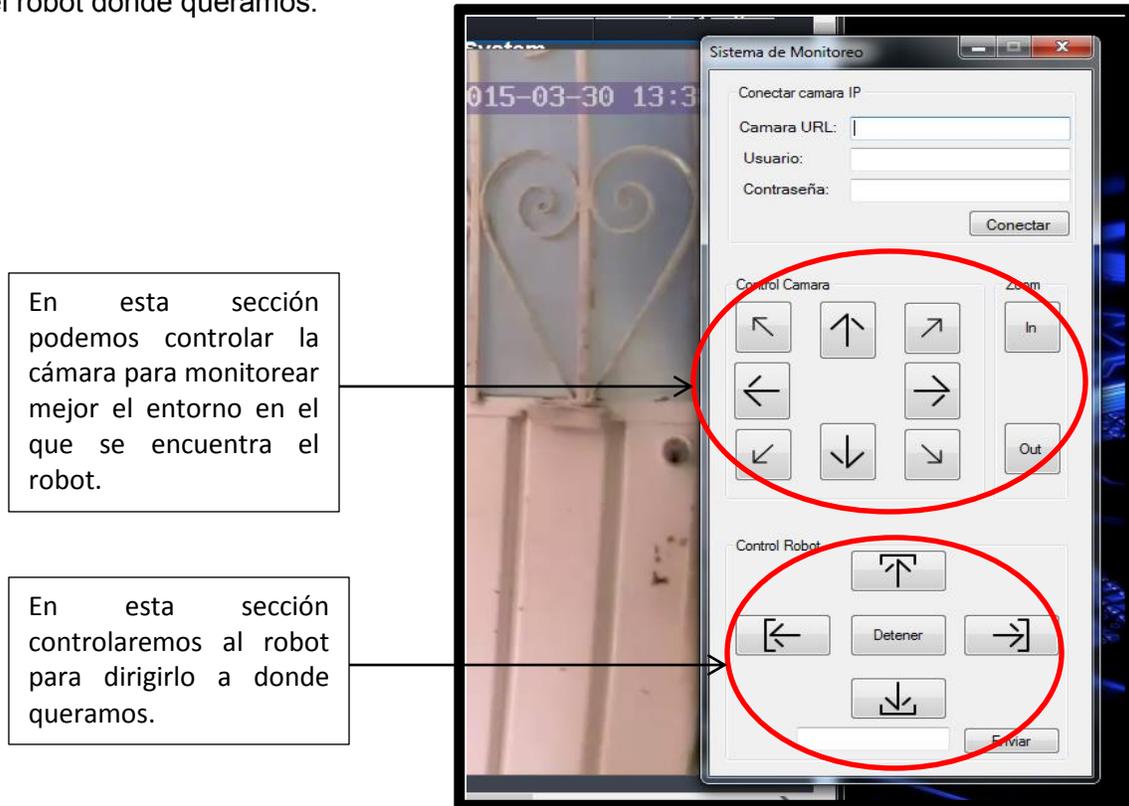


Figura 84. Enlace de cámara IP con interfaz de monitoreo del robot.

Ya con nuestra cámara IP integrada y el robot funcionando tenemos 2 secciones en la interfaz de monitoreo una que dice Control Cámara que es la que se encarga de controlar la cámara a nuestro gusto, y la otra sección que dice Control Robot que se usa para dirigir el robot donde queramos.



En esta sección podemos controlar la cámara para monitorear mejor el entorno en el que se encuentra el robot.

En esta sección controlaremos al robot para dirigirlo a donde queramos.

Figura 85. Sistema del control de la cámara y control del Robot

La cámara IP que estamos utilizando también cuenta con sensor infrarrojo (Figura 86) esto para ayudarnos a ver nuestro entorno en la oscuridad y facilitar la búsqueda en zonas oscuras. Este tipo de cámaras es muy utilizado por investigadores arqueológicos con el fin de mapear rutas en las cuevas que se están explorando.

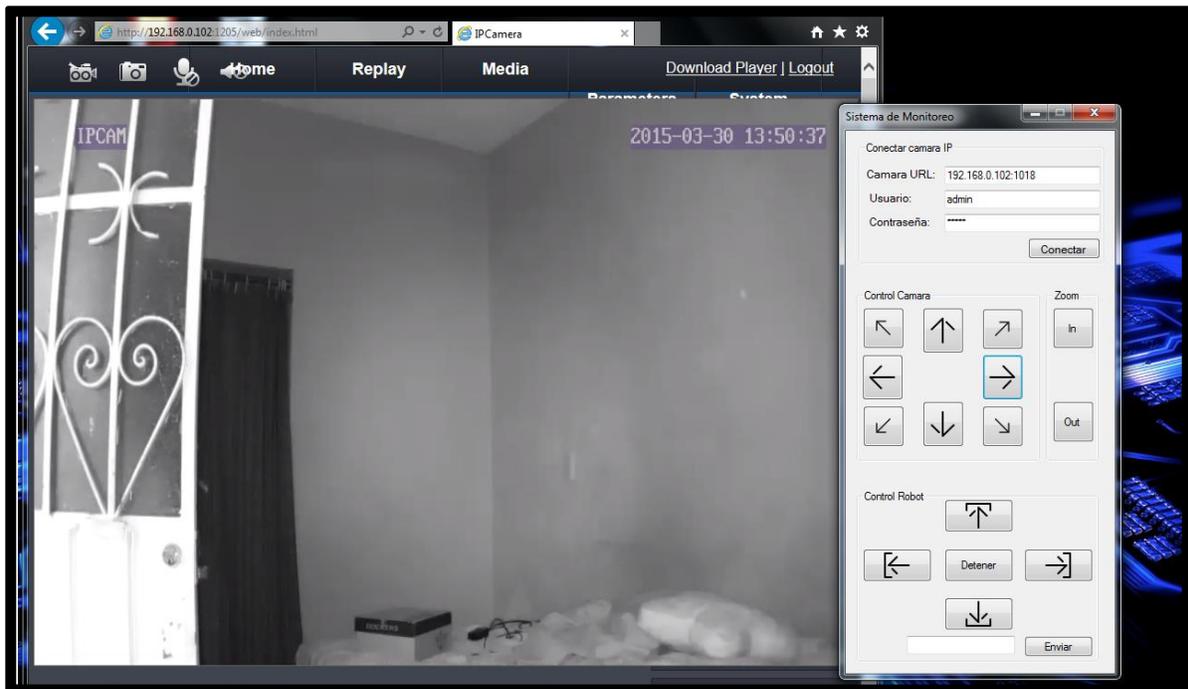


Figura 86. Cámara Infrarroja para acceder a zonas oscuras

En la Figura 86 se muestra el sistema de monitoreo funcionando correctamente, así también establecida la comunicación de la cámara ip y la interfaz, podemos controlar el robot a distancia y usarlo para diferentes áreas de exploración.

4.3 PRUEBAS DEL CONTROL REMOTO

Ahora procederemos hacer pruebas con el control remoto que diseñamos y construimos en la Figura 87 mostramos como queda montado el circuito en el protoboard junto con una antena inalámbrica los botones de la izquierda son para indicarle al robot que vaya hacia adelante o de reversa mientras que los botones de la derecha indican si va hacia la izquierda, derecha o indicar que se detenga.

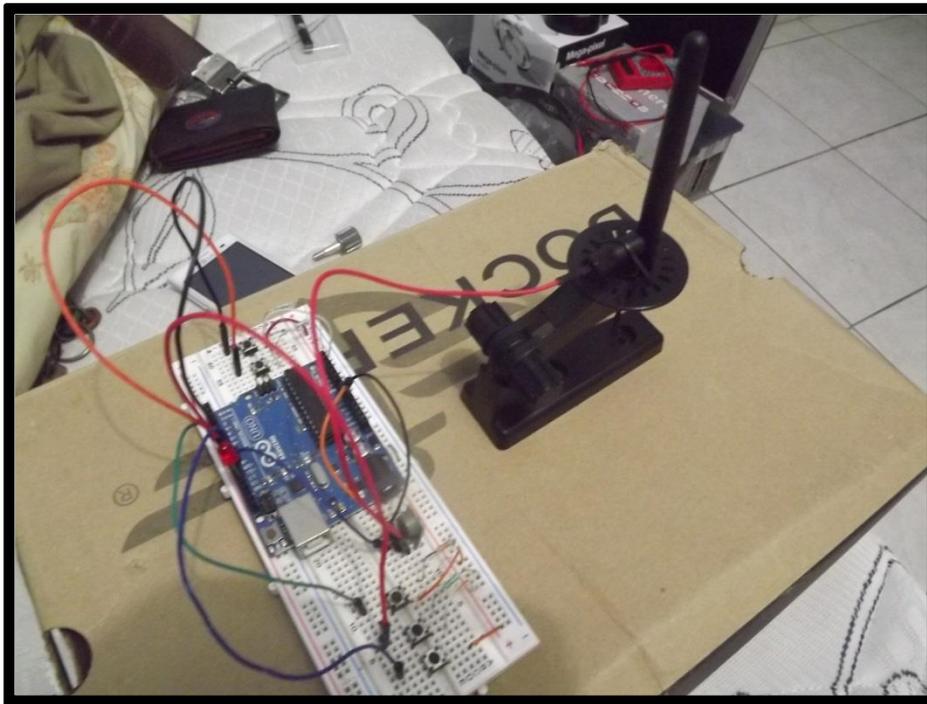


Figura 87. Control de mando a distancia del robot

En el capítulo anterior indicamos como conectar una cámara IP al celular procederemos a probarlo con nuestro control a distancia para indicarle al robot donde queremos que vaya y monitorear el entorno en el que se encuentra, en la Figura 88 le añadimos el celular con la transmisión en vivo de la cámara a nuestro control a distancia esto para poder controlar el robot a distancia de una manera práctica y fácil.

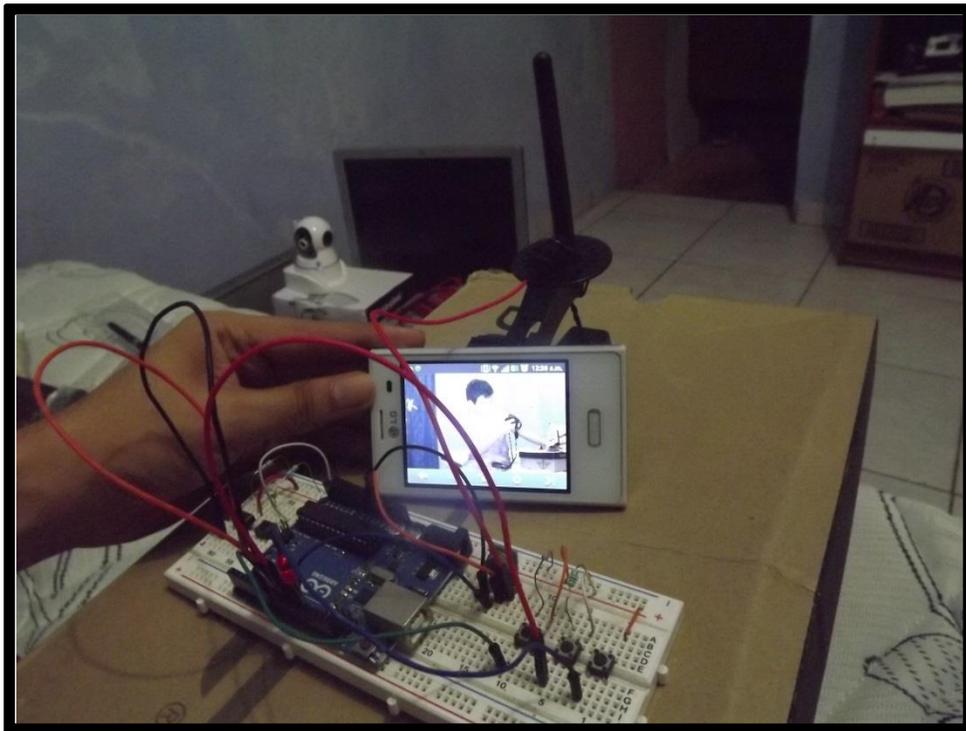


Figura 88. Control del robot y enlace de cámara ip

4.4 PRUEBAS CON LA APLICACIÓN ANDROID

Ahora procederemos hacer pruebas con nuestra aplicación android lo primero que haremos será abrir nuestra aplicación en el celular como se muestra en la Figura 89.



Figura 89. Abrir aplicación en el celular

A continuación procederemos a conectar el robot con nuestra aplicación dándole en el botón conectar esto para establecer comunicación una vez hecho esto podremos controlar nuestro robot con la aplicación Figura 90.

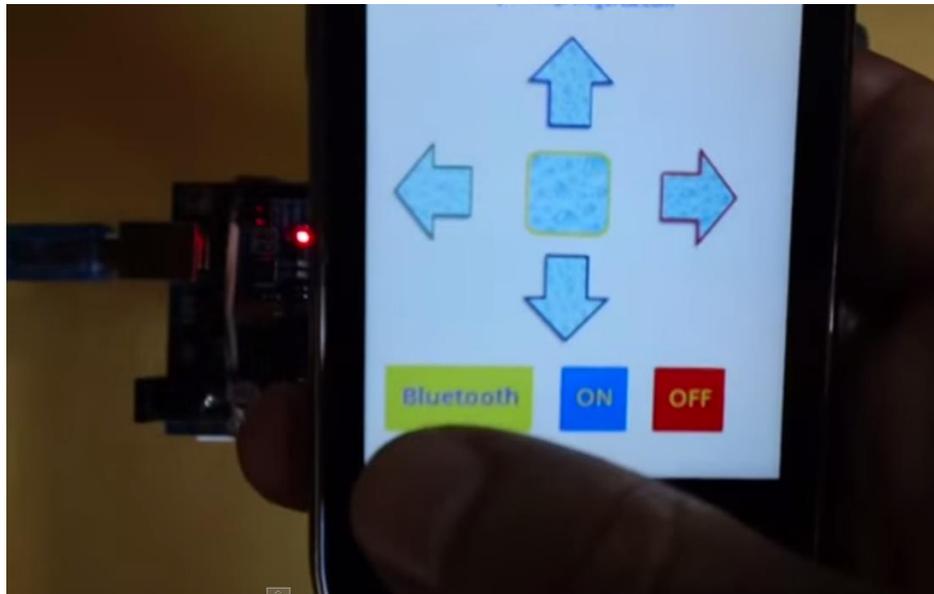


Figura 90. Aplicación android para el control del robot.

CONCLUSIONES Y TRABAJO A FUTURO

En la actualidad el hombre ha hecho uso de la tecnología que ha ido desarrollándose de manera exponencial. Hoy en día, el uso de la robótica ha ido en aumento en sus diversos campos de aplicación, como lo son: en la industria, en la didáctica, en la exploración, en la medicina, etc.

El Proyecto se hizo valido al comprobar y poner a prueba las 2 funciones de monitoreo del robot los cuales mostraron una buena recepción de datos e imágenes, se determinó que al usar la radiofrecuencia como comunicación el robot no mostro ningún tipo de interferencia tanto al aire libre, como en zonas con paredes y obstrucciones.

En el futuro, la mejora en las interfaces de comunicación entre el usuario y el robot, la mejora en la capacidad de captación de información sensorial y en la transmisión de datos por radiofrecuencia, resultara en una extensión del concepto del robot como uno de los más amplios en la realización de actividades muy diversas y de protección de integridad, física siendo esto un gran beneficio para el ser humano.

Determinando así que este proyecto será útil para diversos campos de investigación en México, como por ejemplo en arqueología al utilizar robots exploradores para investigar más a fondo las cámaras ocultas de las pirámides gracias a su eficacia y rapidez de comunicación, en situaciones de rescate al explorar todas las posibilidades de riesgo al lugar que se deberá ingresar, también el monitoreo de vigilancia de una casa o propiedad.

REFERENCIAS

[1] David G. Maxínez, Francisco Javier Sánchez Rangel, Guillermo Castillo Tapia, Petra Baldovinos Noyola, M. Antonieta García Galván, Moisés G Sierra. Robotics, Education and Economy, International Journal of Science and Engineering Investigations vol. 1, issue 8, September 2012

[3] http://cfievalladolid2.net/tecno/ctrl_rob/robotica/intro.htm

[4] A. M. Turing (1950) Computing Machinery and Intelligence. *Mind* 49: 433-460.

[7] OSCILACIONES. Daniel Maggiolo, apuntes de acústica. MARZO 2000

<http://www.eumus.edu.uy/docentes/maggiolo/acuapu/osc.html>

[8] SEÑAL ANALÓGICA. Señal eléctrica Analógica. ENERO 2003

http://www.asifunciona.com/electronica/af_conv_ad/conv_ad_2.htm

[6] COMUNICACIÓN SERIAL. Puerto Serial / COM / RS232. OCTUBRE 2012

http://www.informaticamoderna.com/El_puerto_serial.htm

[3] <http://angrybirdsignals.site90.com/paginas/Adrian.html>

DISEÑO Y CONTROL DE ROBOTS MÓVILES. Aguilera Hernández Martha I., Bautista Miguel A., Iruegas Joaquín Depto. de Ing. Eléctrica y Electrónica. Instituto Tecnológico de Nuevo Laredo. DICIEMBRE 2014

<http://www.mecamex.net/anterior/cong02/papers/art24.pdf>

ROBOTICA. Robots Móviles. 2001

<http://www.xatakaciencia.com/robotica/robots-moviles-i>

MAQUINAS AUTOMATICAS Y ROBOTS. Irene Robótica, Robots Móviles.

<https://sites.google.com/site/irenerobotica/6-robots-moviles>

COMUNICACIÓN Y RADIOFRECUENCIA. Descripción y Funcionamiento.

Eduardo J. Carletti. 2012 http://robots-argentina.com.ar/Comunicacion_RF.htm

MICROCONTROLADORES. El mundo de los microcontroladores, Programación en C con ejemplos. SEPTIEMBRE 2013

<http://www.mikroe.com/chapters/view/79/capitulo-1-el-mundo-de-los-microcontroladores/>

[12]ZOCALO. Electrónica. ABRIL 2000

http://es.wikipedia.org/wiki/Z%C3%B3calo_%28electr%C3%B3nica%29

[13]COPROCESADOR MATEMATICO. Diccionario Informático, Foros y Ayuda.

<http://www.alegsa.com.ar/Diccionario/C/8388.php>

[12]MICROCONTROLADORES PIC. Arquitectura de procesador siguiendo el modelo Harvard. JUNIO 2012.

<http://perso.wanadoo.es/pictob/micropic.htm>

FUNCIONAMIENTO DEL MECANISMO DE INTERRUPCIONES. Procesamiento de una Interrupción. JULIO 2011.

<http://es.wikipedia.org/wiki/Interrupci%C3%B3n>

[15]CONVERSOR ANALOGICO DIGITAL. Muestreo de la Señal Analógica.

http://www.asifunciona.com/electronica/af_conv_ad/conv_ad_5.htm

[10]DESCRIPCION Y FUNCIONAMIENTO DE UNA CAMARA IP.

<http://valetron.eresmas.net/CamarasIP.htm>

[16]RS232 Serial. AGOSTO 2012.

http://www.informaticamoderna.com/El_puerto_serial.htm

[17]PUERTO DE COMUNICACIÓN USB. Características del puerto USB. 2012

http://www.informaticamoderna.com/El_puerto_USB.htm

[18]PUERTO DE COMUNICACIÓN ETHERNET. Características del puerto Ethernet. OCTUBRE 2014

<http://www.mastermagazine.info/termino/4930.php>

[19]MICROCONTROLADOR ARDUINO. Características Principales. MARZO 2010.

<http://arduino.cc/en/Reference/HomePage>

[14]DESCRIPCION L293B. Características y Datasheet. 2010.

http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/3/L293B.shtml

APÉNDICE

CÓDIGO FUENTE PARA INTERFAZ GRAFICA

A continuación se pondrá el código del programa como ejemplo de que se realizó correctamente.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using Robot.Properties;

namespace Robot
{
    public partial class Robot : Form
    {
        public Robot()
        {
            InitializeComponent();
            serialPort1.PortName = "COM3";
            serialPort1.BaudRate = 9600;
            serialPort1.Open();
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (serialPort1.IsOpen) serialPort1.Close();
        }

        private void Adelante_Click(object sender, EventArgs e)
        {
            serialPort1.Write("f");
            Adelante.Enabled = false;
            Atras.Enabled = true;
            Izquierda.Enabled = true;
            Derecha.Enabled = true;
            Detener.Enabled = true;
        }

        private void Atras_Click(object sender, EventArgs e)
        {
            serialPort1.Write("b");
            Atras.Enabled = false;
            Adelante.Enabled = true;
            Izquierda.Enabled = true;
            Derecha.Enabled = true;
            Detener.Enabled = true;
        }

        private void Izquierda_Click(object sender, EventArgs e)
```

```

        {
            serialPort1.Write("l");
            Atras.Enabled = true;
            Adelante.Enabled = true;
            Izquierda.Enabled = false;
            Derecha.Enabled = true;
            Detener.Enabled = true;
        }

privatevoid Derecha_Click(object sender, EventArgs e)
    {
        serialPort1.Write("r");
        Atras.Enabled = true;
        Adelante.Enabled = true;
        Izquierda.Enabled = true;
        Derecha.Enabled = false;
        Detener.Enabled = true;
    }

privatevoid Detener_Click(object sender, EventArgs e)
    {
        serialPort1.Write("s");
        Atras.Enabled = true;
        Adelante.Enabled = true;
        Izquierda.Enabled = true;
        Derecha.Enabled = true;
        Detener.Enabled = false;
    }

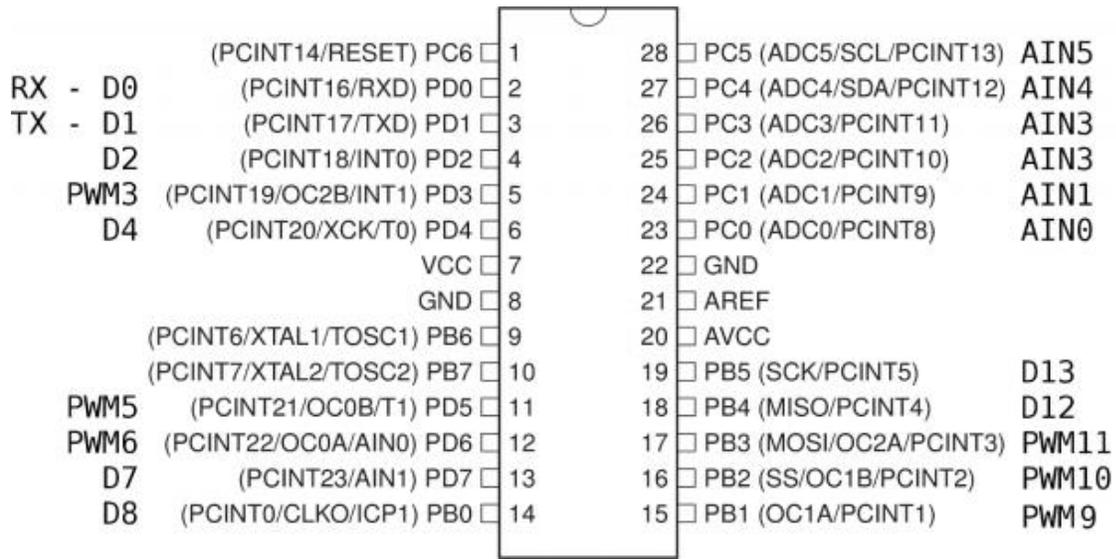
privatevoid Robot_Load(object sender, EventArgs e)
    {
        foreach (Control control in this.Controls)
        {
            control.PreviewKeyDown += new PreviewKeyDownEventHandler(control_PreviewKeyDown);
        }
    }

void control_PreviewKeyDown(object sender, PreviewKeyDownEventArgs e)
    {
        if (e.KeyCode == Keys.Up)
            {
                serialPort1.Write("f");
                Adelante.Enabled = false;
                Atras.Enabled = true;
                Izquierda.Enabled = true;
                Derecha.Enabled = true;
                Detener.Enabled = true;
                e.IsInputKey = true;
            }

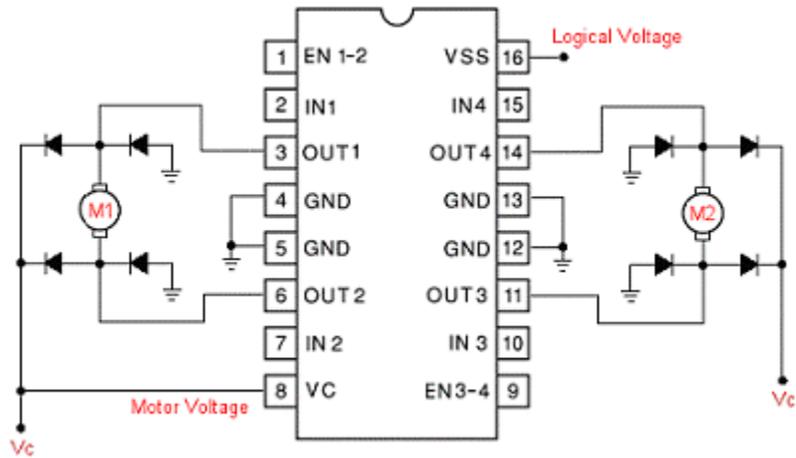
        elseif (e.KeyCode == Keys.Down)
            {
                serialPort1.Write("b");
                Adelante.Enabled = true;
                Atras.Enabled = false;
                Izquierda.Enabled = true;
                Derecha.Enabled = true;
                Detener.Enabled = true;
            }
    }

```

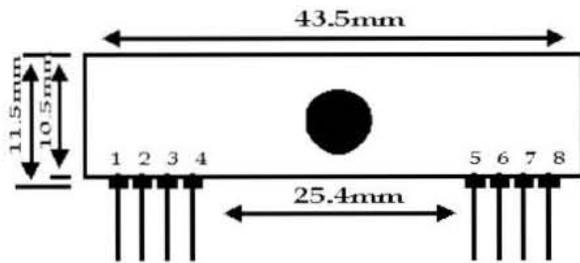

DATASHEET ATMEGA 328P



DATASHEET L293B PARA MOTORES

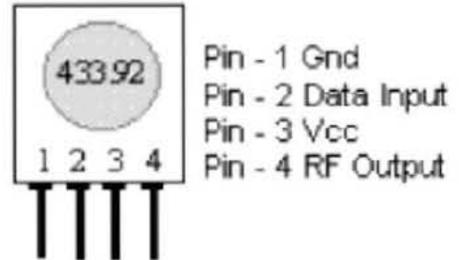


DATASHEET TRANSMISOR



- pin 1 : Gnd
- pin 2 : Digital Output
- pin 3 : Linear Output
- pin 4 : Vcc
- pin 5 : Vcc
- pin 6 : Gnd
- pin 7 : Gnd
- pin 8 : Ant (About 30 - 35 cm)

DATASHEET RECEPTOR



- Pin - 1 Gnd
- Pin - 2 Data Input
- Pin - 3 Vcc
- Pin - 4 RF Output