



ESCUELA DE INGENIERIA EN SISTEMAS

Tema:

“DISEÑO DE UN SISTEMA DE CONTROL DE SERVOMOTORES Y SU IMPLEMENTACIÓN PARA UN ROBOT BAILARÍN DE 16 GRADOS DE LIBERTAD”

Disertación de grado previo a la obtención del título de Ingeniero de Sistemas y Computación

Línea de investigación:

Inteligencia Artificial, Robótica, Domótica y Sistemas Expertos

Autor:

Francisco Rubén Coba Yépez

Director:

Mg. Acurio Maldonado Santiago Alejandro

Ambato – Ecuador

Enero 2016

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

“DISEÑO DE UN SISTEMA DE CONTROL DE SERVOMOTORES Y SU IMPLEMENTACIÓN PARA UN ROBOT BAILARÍN DE 16 GRADOS DE LIBERTAD”.

Línea de Investigación:

Inteligencia Artificial, Robótica, Domótica y Sistemas Expertos.

Autor:

COBA YÉPEZ FRANCISCO RUBÉN

Santiago Alejandro Acurio Maldonado, Mg. f. _____

CALIFICADOR

Galo Mauricio López Sevilla, Ing.Mg. f. _____

CALIFICADOR

Enrique Xavier Garcés Freire, Ing. Mg. f. _____

CALIFICADOR

Teresa Milena Freire Aillón, Mg. f. _____

DIRECTORA DE LA ESCUELA DE SISTEMAS

Hugo Rogelio Altamirano Villaroel, Dr. f. _____

SECRETARIO GENERAL PUCESA

Ambato – Ecuador

Enero - 2016

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Francisco Rubén Coba Yépez portador de la cédula de ciudadanía No. 1803988862, declaro que los resultados obtenidos en la investigación que presento como informe final, previo a la obtención del título de Ingeniero de Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la reducción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

.....

Francisco Rubén Coba Yépez

C.I. 1803988862

AGRADECIMIENTO

Quiero agradecer a Dios por permitirme lograr esta meta personal, a mis padres por su ayuda incondicional; al ser que me ha dado la vida mi madre por su dedicación tiempo y compañía en todo este tiempo.

A mi hija Antonella por ser la fuente de inspiración para lograr mis metas y objetivos.

A mis compañeros, docentes y tutores por su amistad y apoyo.

Francisco Rubén Coba Yépez

DEDICATORIA

Este presente trabajo va dedicado a Dios por que sin el nada sería posible, mi familia por su apoyo incondicional y único, a mi princesa mi hija Antonella para ella todas mis dedicatorias de logros y superación profesional.

A mi tutor docente y amigo Ing. Santiago Acurio por su tiempo, dedicación y comprensión en el desarrollo de la tesis de grado.

A mi amor Sole por tu tiempo, esfuerzo y ánimos para que pueda lograr y culminar esta meta.

Francisco Rubén Coba Yépez

RESUMEN

El presente trabajo se enfoca en el desarrollo de un robot de 16 grados de libertad que emula movimientos de ritmos de baile y que además es controlado mediante comunicación infrarroja desde una interface gráfica. Para lograrlo se procede en primera instancia a ensamblar el robot Robonova I y posteriormente realizar la programación de su tarjeta de control con una serie de movimientos que permitan dar la emulación del baile. Además se realiza el diseño y construcción del módulo de comunicación infrarroja que irá conectado mediante comunicación RS232 a una PC cargada con la interface de control. El módulo de comunicación infrarroja reemplaza al control remoto que posee el robot, por lo cual se debe realizar un estricto análisis de los pulsos de datos entregados por el control remoto para posterior a esto proceder a generarlos con el microcontrolador ubicado en la tarjeta diseñada. El presente trabajo muestra como resultados un robot bailarín de 4 ritmos diferentes, manipulado desde su control remoto original o desde una PC mediante el módulo de comunicación infrarroja diseñado y construido por el autor, en el cual se despliegan los códigos fuente de la tarjeta de control del robot de la tarjeta de comunicación infrarroja y el algoritmo de programación de la interfase de control del robot mediante LABVIEW, cuyos códigos podrán ser editados en el caso de que se requiera desarrollar nuevos ritmos de baile o movimientos del robot.

Palabras clave: robonova, labview, robótica, microcontrolador

ABSTRACT

This work is focused on the development of a robot with 16 degrees of independence that emulates dance rhythms movements which is controlled by infrared communication from a graphical interface. At first, the robot Robonova I is assembled to accomplish it, to then program the control board with a series of movements that enable emulation dance. Moreover the design and construction of the infrared communication module will be connected through an RS232 communication to a computer that is loaded in a control interface, the infrared communication module replaces the remote control that the robot has, therefore a strict analysis of data delivered by remote control must be done to later generate data in a microcontroller located in the card board that was designed. This works shows as a product, a dancing robot with 4 different rhythms, handled from the original remote control or from a PC using the infrared communication module designed and developed by the author, in which the source code of the robot control board is displayed from the infrared communication card and the scheduling algorithm of the robot control interface through LABVIEW, whose codes may be edited if it is required to develop new dance rhythms and movements of the robot.

Keywords: robonova, labview, robotics, microcontroller

ÍNDICE DE CONTENIDOS

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD	iii
AGRADECIMIENTO	iv
DEDICATORIA	v
RESUMEN.....	vi
ABSTRACT	vii
ÍNDICE DE CONTENIDOS.....	viii
INDICE DE ILUSTRACIONES	xii
INTRODUCCIÓN.....	1
CAPÍTULO I.....	3
Revisión de la literatura o fundamentos teóricos.....	3
1.1 Antecedentes.....	3
1.2 Problema	4
1.1.1 Descripción del problema.....	4
1.3 Preguntas Básicas	4
1.4 Delimitación del Tema	5
1.1.2 Delimitación Temporal.	5
1.1.3 Delimitación Espacial.....	5
1.1.4 Delimitación Funcional	6
1.5 Objetivos.....	6
1.1.5 General.....	6
1.1.6 Específicos	6
1.6 Marco Teórico.....	7
1.1.7 Clasificación de los robots	8

1.6.1.1	Robots de servicio.....	10
1.6.1.2	Robots teleoperados.....	11
1.1.8	Componentes principales de un robot.....	12
1.6.1.3	Sistema Mecánico.....	12
1.6.1.4	Actuadores.....	12
1.6.1.5	Sensores.....	13
1.7	Robots Humanoides.....	13
1.8	Aplicaciones de los robots.....	14
1.9	Sistema electrónicos de comunicaciones.....	15
1.10	Comunicacion Infrarroja.....	17
1.1.9	Comunicación Infrarroja aplicada a robótica móvil.....	18
1.11	Software Robobasic.....	20
1.12	Software Labview.....	21
1.13	Software Bascom Avr.....	21
1.14	Software Proteus ISIS.....	22
CAPÍTULO II.....		23
Metodología.....		23
2.1	Metodología de Investigación.....	23
2.1.1	Tipo de Investigación.....	23
2.1.2	Fuentes de Información.....	24
2.1.2.1	Primarias.....	24
2.1.2.2	Secundarias.....	24
2.1.3	Técnicas de Investigación.....	24
2.1.3.1	Observación Científica Estructurada.....	24
2.2	Metodología de desarrollo.....	25

CAPÍTULO III.....	27
Resultados.....	27
3.1 Generalidades de Robonova I.....	27
3.2 Fase I: Armado de Robonova I	27
3.2.1 Servos.....	30
3.2.2 Cable de conexión serie.....	31
3.2.3 Microcontrolador	32
3.3 Fase II: Proceso de programación para realizar un movimiento coordinado.....	36
3.4 Fase III: Diseño del módulo infrarrojo.....	41
3.4.1 Metodología de desarrollo de hardware.	41
3.4.2 Generación de Pulsos en comunicación infrarroja.....	41
3.4.3 Desarrollo de interface gráfica en Labview.....	43
3.4.4 Módulo de control.....	44
3.4.5 Generación de tren de pulsos infrarrojos.....	46
3.4.6 Diseño Electrónico	47
3.5 Fase IV: Desarrollo de interface gráfica en Labview.	54
3.5.1 Metodología de desarrollo de software.	54
3.6 Fase V: Captura del tren de pulsos.	61
3.7 Fase VI: Código de programación de microcontrolador.	63
CAPÍTULO IV	68
Discusión, análisis y validación de los resultados	68
4.1 Funcionamiento del Robot, módulo infrarrojo e interface de control de la PC....	68
4.2 Análisis de Resultados:.....	70

4.3	Validación de Resultados:.....	71
	CAPÍTULO V	73
	CONCLUSIONES Y RECOMENDACIONES	73
5.1	CONCLUSIONES	73
5.2	RECOMENDACIONES	75
	BIBLIOGRAFIA.....	76
	ANEXOS.....	79

INDICE DE ILUSTRACIONES

Ilustración 1. 1 Tipos de robots según la IFR.	9
Ilustración 1. 2 Tipos de robots según la AFRI.	9
Ilustración 3. 1 Servomotores del kit robonova I.	28
Ilustración 3. 2 Distribución de señales hacia el servomotor	28
Ilustración 3. 3 Ubicación de los componentes para armado de las piernas del robot.	29
Ilustración 3. 4 Robot armado completamente y conectado con la placa de control.	30
Ilustración 3. 5 Servomotores HSR 8498HB.	31
Ilustración 3. 6 Cable de programación del robot.	32
Ilustración 3. 7 Distribución de los puertos de la placa MCR3024.	34
Ilustración 3. 8 Placa MCR3024.	34
Ilustración 3. 9 Ubicación de la Placa MCR3024 en la espalda del robot.	35
Ilustración 3. 10 localización de elementos principales.	35
Ilustración 3. 11 Ventana de configuración del zeros de los servomotores.	36
Ilustración 3. 12 Ventana ROBONOVA Motor Control.	37
Ilustración 3. 13 Ventana de programación robo BASIC V2.5 y sus componentes principales.	39
Ilustración 3. 14 Comunicación entre control remoto y Robot Robonova.	42
Ilustración 3. 15 Comunicación entre PC y Robot Robonova.	42
Ilustración 3. 16 Conversión de Protocolos de Comunicación.	43
Ilustración 3. 17 Interfaces Desarrolladas en Labview.	44
Ilustración 3. 18 Diagrama de Bloques del módulo de control.	45

Ilustración 3. 19 Distribución de pines del microcontrolador ATMEGA 164p.	46
Ilustración 3. 20 Control Remoto.....	46
Ilustración 3. 21 Visualización de los pulsos.	47
Ilustración 3. 22 Foto del módulo infrarrojo construido.....	48
Ilustración 3. 23 Conexión del circuito integrado MAX232.	48
Ilustración 3. 24 Conexión del microcontrolador.	49
Ilustración 3. 25 Pines de programación del microcontrolador.	50
Ilustración 3. 26 Pantalla LCD para visualización de órdenes.....	50
Ilustración 3. 27 Diagrama de conexión del Microcontrolador con LCD.	51
Ilustración 3. 28 Diagrama de conexión para led infrarrojo.	51
Ilustración 3. 29 Diagrama de conexión de leds.....	52
Ilustración 3. 30 Fuentes de alimentación de 5V DC.....	52
Ilustración 3. 31 PCB de tarjeta electrónica de control.....	53
Ilustración 3. 32 Vista 3D de tarjeta electrónica de control.....	53
Ilustración 3. 33 Software de control realizado en LABVIEW.....	55
Ilustración 3. 34 Ventana de Inicio.....	55
Ilustración 3. 35 Ventana Principal.....	56
Ilustración 3. 36 Ventana Archivos.....	57
Ilustración 3. 37 Inicialización Comunicación RS232.	58
Ilustración 3. 38 Creación de Trama de datos.....	58
Ilustración 3. 39 Selección de ritmo.	59
Ilustración 3. 40 Almacenamiento de ritmo.	59
Ilustración 3. 41 Activación de reproducción de ritmo.	60
Ilustración 3. 42 Ventana de visualización de pulsos generados usando	

osciloscopio.	61
Ilustración 3. 43 Ventana de visualización de pulsos de inicio.	61
Ilustración 3. 44 Pulsos generados por el Microcontrolador.	62
Ilustración 3. 45 Pulsos de comandos generados por el Microcontrolador. ...	62
Ilustración 3. 46 Conjunto total de pulsos generados por el Microcontrolador.	63
Ilustración 3. 47 Ventana de Programación del microcontrolador usando BASCOM AVR.....	64
Ilustración 3. 48 Declaración de variables usando BASCOM AVR.....	64
Ilustración 3. 49 Configuración de puertos usando BASCOM AVR.	65
Ilustración 3. 50 Asignación de puertos usando BASCOM AVR.	65
Ilustración 3. 51 Configuración de comunicación serial usando BASCOM AVR.....	65
Ilustración 3. 52 Interrupción de comunicación serial usando BASCOM AVR.	66
Ilustración 3. 53 Determinación de ritmo usando BASCOM AVR.....	66
Ilustración 3. 54 Selección de ritmo usando BASCOM AVR.	67
Ilustración 3. 55 Portada del módulo de control infrarrojo.	67
Ilustración 4. 1 Hardware armado y construido. Robot-Modulo Infrarrojo- Control Remoto.	68
Ilustración 4. 2 Interface de control del robot.	69
Ilustración 4. 3 Robot Bailando.	72

INTRODUCCIÓN

El presente trabajo de investigación se lo desarrolla buscando la línea de aplicación de la robótica como herramienta para realizar tareas de entretenimiento y aprendizaje, mismo que está estructurado en cuatro capítulos siendo los siguientes:

El CAPÍTULO I, está estructurado con el planteamiento del problema, el marco teórico, el objetivo general y los objetivos específicos de la investigación que se llevará a cabo.

El CAPÍTULO II, lo constituye la justificación de la investigación que permite fundamentar científicamente la propuesta, además de la metodología de desarrollo.

El CAPÍTULO III, lo constituye los resultados con el detalle de todo lo realizado para lograr los objetivos específicos planteados así como el objetivo principal. Se explica el diseño y la elaboración del hardware, además de la implementación de las instrucciones de control para los movimientos del robot.

El CAPÍTULO IV, se analizan y explican los resultados obtenidos para validar de manera adecuada el desempeño del sistema robótico, para ello se realizaron pruebas de los movimientos con distintos ritmos de baile.

El CAPÍTULO V, detalla las conclusiones y recomendaciones del trabajo de investigación desarrollado.

Concluyendo con la bibliografía y anexos del trabajo de investigación.

CAPÍTULO I

Revisión de la literatura o fundamentos teóricos

1.1 Antecedentes

La robótica constituye un indicativo de progreso y desarrollo tecnológico de la humanidad que ha alcanzado metas que hace unos años correspondían al ámbito de la ciencia ficción.

La informática junto al uso de técnicas y metodologías de inteligencia artificial proveerán en pocos años más la aparición de robots dotados de una gran flexibilidad y capacidad de adaptación al entorno para el cual se los diseña ya que se tiene como enfoque global complementar o sustituir las funciones de los humanos en tareas tediosas o peligrosas, alcanzando, en algunos sectores, aplicaciones masivas, llegando a obtener beneficios como: mayor productividad, calidad y seguridad. Además hay que destacar que la robótica ofrece beneficios sociales, resolviendo problemas cotidianos en todos los sectores y edades de la población, mejorando la calidad de vida de los ciudadanos mediante la reducción de las horas de trabajo y de riesgos laborales.

Los sectores a los que actualmente está orientada la robótica son muy amplios desde la industria manufacturera (automóvil y máquina-herramienta) hasta la creación de sistemas de entretenimiento domésticos, así como

también la exploración de ambientes hostiles, tales como entornos submarinos y el espacio exterior. (Comité Español de Automática, 2011)

1.2 Problema

1.1.1 Descripción del problema

Con el fin de apoyo didáctico e incursión en el campo de la robótica, se implementará la construcción de un robot bailarín humanoide con 16 grados de libertad que al poseer una comunicación inalámbrica confiable entre el computador y la tarjeta de control, permitirá diseñar y programar diversas rutinas de movimientos para ser utilizado como material de apoyo para práctica de Robótica e Inteligencia Artificial de la Escuela de Ingeniería en Sistemas de la PUCESA.

1.3 Preguntas Básicas

- **¿En que ayudaría el robot humanoide con 16 grados de libertad a la PUCESA?**

Permitirá desarrollar mayores destrezas en los lenguajes de programación para el control de robots.

- **¿Ayudaría a los estudiantes el estudio de la robótica?**

Si, debido a que la robótica es ampliamente utilizada en los distintos campos de investigación.

- **¿Será material de apoyo didáctico el robot humanoide?**

Si, ya que el robot al poseer varios grados de libertad permitirá a los estudiantes realizar movimientos semejantes a los de una persona.

- **¿Una comunicación inalámbrica adecuada entre la tarjeta de control y el computador permitirá tener una mejor concepción de un robot humanoide controlado libremente?**

Si, debido a que la tendencia actual es que cualquier sistema sea controlado y monitoreado de forma remota, bajo un protocolo de comunicación inalámbrica.

1.4 Delimitación del Tema

1.1.2 Delimitación Temporal.

Este problema será estudiado en el periodo comprendido entre Abril 2014 y Septiembre 2014.

1.1.3 Delimitación Espacial

Provincia:	Tungurahua
Ciudad:	Cantón Ambato
Empresa:	Pontificia Universidad Católica del Ecuador Sede Ambato
Ubicación:	Av. Manualita Sáenz Sector el Tropezón.
Servicio:	Educativo

1.1.4 Delimitación Funcional

El sistema a implementarse es un robot bailarín de 16 grados de libertad comandado a través de una interface en computadora mediante comunicación inalámbrica permitiendo tener un sistema autónomo e independiente.

Con el propósito de lograr lo mencionado, se deberán cumplir las siguientes actividades:

- Ensamblaje de los componentes físicos del robot.
- Programación de la tarjeta de control de posición de servomotores.
- Diseño e implementación de la tarjeta de comunicación inalámbrica.
- Diseño y programación de la Interface visual de comunicación mediante una PC.
- Programación mediante software de movimientos coordinados de baile.

1.5 Objetivos

1.1.5 General

Diseñar un sistema de control de servomotores y su implementación para un robot bailarín de 16 grados de libertad

1.1.6 Específicos

- Recopilar información sobre los diferentes tipos de software y controladores para el robot humanoide.

- Obtener posiciones controladas del movimiento de los servomotores para llegar a tener una semejanza de la movilidad entre el robot y el ser humano.
- Implementar una tarjeta electrónica de comunicación inalámbrica entre el robot y el computador, que permita procesar instrucciones de rutinas de baile y ejecutarlas por control de servomotores.
- Desarrollar una interface gráfica en el computador que permita tener el control del robot humanoide manejado remotamente.

1.6 Marco Teórico

Según Ollero, los robots son máquinas en las que se integran componentes mecánicos, eléctricos, electrónicos y de comunicaciones, y dotadas de un sistema informático para su control en tiempo real, percepción del entorno y programación. La robótica está experimentando un crecimiento explosivo propulsado por los avances en computación, sensores, electrónica y software. (Ollero, A. 2001)

Cada día los medios para gobernar o monitorear un robot están al alcance de desarrolladores e investigadores. En el ámbito de la robótica se dispone de una gran gama de sensores y utilitarios que facilitan dicha actividad. Para el desarrollo de interfaces de control se dispone de herramientas de código abierto, sistemas embebidos de control y adquisición de datos.

La historia de la robótica se destaca por un mundo de fantasía que ha proporcionado la inspiración para convertir la fantasía en realidad. Es una

rica historia con la creatividad cinematográfica, científico y visión empresarial. Sorprendentemente, la definición de un robot es polémica, incluso entre los expertos en robótica. (Kurfess T. s.f.)

La robótica está influyendo en muchas áreas que aunque parezcan incoherentes van incursionando, debido a que colabora de muchas formas, y permiten que determinados trabajos se faciliten considerablemente.

Encaminado en paralelo con la evolución de los laboratorios de investigación, el uso de la robótica en la industria floreció por la necesidad de hacer más cómoda la vida del hombre. La investigación es un fuerte aporte para el avance en proyectos de robótica, Ecuador los últimos año ha aportado con grandes desarrollos que permiten en cierta forma facilitar o ayudar en la comodidad del hogar, tales como limpia vidrios automáticos, aspiradoras automáticas.

1.1.7 Clasificación de los robots

Según RIA (Asociación de Industrias robóticas), un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas. La IFR distingue entre cuatro tipos de robots. (Barrientos A, 2008)

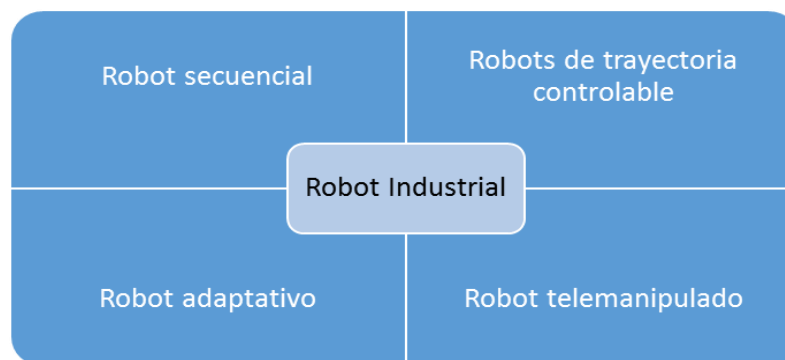


Ilustración 1. 1 Tipos de robots según la IFR.

Fuente: (Barrientos A, 2008)

Esta clasificación coincide en gran medida con la establecida por la Asociación Francesa de Robótica Industrial (AFRI). (Barrientos A, 2008)

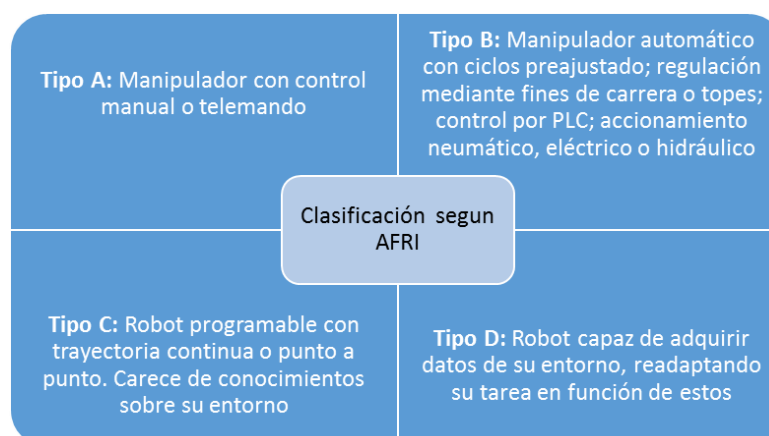


Ilustración 1. 2 Tipos de robots según la AFRI.

Fuente: (Barrientos A, 2008)

La mayoría de estos tipos de robots tienen algo en común, que es la finalidad de uso y el ámbito donde se ponen en marcha, que es el sector industrial. En este tipo de ambientes los robots mantienen una robustez que les hace ser diferentes a los robots comunes.

Adicional a esto los robots tipo industriales puede clasificarse de acuerdo a generaciones

1ra Generación: repite la tarea programada secuencialmente, no toma en cuenta las posibles alteraciones de su entorno

2da Generación: Adquiera información limitada de su entorno y actúa en consecuencia, puede localizar, clasificar (visión), detectar esfuerzos y adaptar sus movimientos en consecuencia

3ra Generación: Su programación se realiza mediante el empleo de un lenguaje natural, posee capacidad para la planificación automática de tareas (Barrientos A, 2008)

Mediante esta subdivisión de robots, permite especializar a un robot en determinada área, logrando mejores resultados y con un número menor de errores.

1.6.1.1 Robots de servicio

Dispositivos electromecánicos móviles o estacionarios, dotados normalmente de uno o varios brazos mecánicos independientes, controlados por un programa de ordenador y que realizan tareas no industriales de servicio, se dividen en:

- Telemanipulador
- Manipulador Secuencial
- Robot servocontrolado estacionario
- Robot sensorizado estacionario
- Robot actuación en exteriores

De una forma similar el presente proyecto sera controlado por medio de ordenes remotas para poder gobernar sus movimientos. (Barrientos A, 2008)

La mayoría de este tipo de robots permiten hacer movimientos donde su base es estacionaria, permitiendo manipular objetos de mayor peso.

1.6.1.2 Robots teleoperados.

Dispositivos robóticos con brazos manipuladores, sensores y cierto grado de movilidad, controlados remotamente por un operador humano de manera directa o a través de un ordenador

- Funcionoides
- Humanoides
- Insectoides
- Robots domésticos
- Robots submarinos

(Barrientos A, 2008)

Dentro de esta distribucion de robots, se encuentra el sistema Rovonoba I, que pertenece a la gama de humanoides. Este tipo de autómatas al ser operados remotamente o mediante un computador poseen una extensa robustez en la parte de software.

1.1.8 Componentes principales de un robot

1.6.1.3 Sistema Mecánico

El sistema mecánico está compuesto por diversas articulaciones. Normalmente se distingue entre el brazo y el órgano terminal o efector final que se puede ser intercambiable, empleando pinzas o dispositivos específicos para distintas tareas.

El aumento del número de articulaciones aporta mayor maniobrabilidad, pero dificulta el problema del control, obteniéndose normalmente menores precisiones por acumulación de errores. (Ollero, A. 2001)

Un aspecto muy esencial en un robot es la estructura o sistema mecánico, debido a los parámetros de estabilidad en los movimientos, además de la robustez que le ayuda a acoplarse a cualquier escenario. A medida que el número de articulaciones o grados de libertad aumentan, de la misma forma aumenta el sistema de control.

1.6.1.4 Actuadores

Los actuadores generan las fuerzas o pares necesarios para animar a la estructura mecánica. Se utilizan tecnologías hidráulicas, para desarrollar potencias importantes, pero en la actualidad se ha extendido el empleo de motores eléctricos y en particular motores de corriente continua servocontrolados, empleándose en algunos casos motores paso a paso y otros actuadores electromecánicos sin escobillas. (Ollero, A. 2001)

Robonoba I que es el sistema robótico empleado posee 16 servomecanismos que le permiten movimientos secuenciales. Este tipo de mecanismos ayudan a tener una mejor resolución en los movimientos graduales o de posicionamiento.

En ambito eléctrico y electrónico existen distintas marcas a nivel mundial que van de la mano dos aspectos torque y resolución versus costo económico.

1.6.1.5 Sensores

Los primeros robots industriales eran programados exclusivamente por guiado manual, almacenando la secuencia de posiciones en memoria digital. La interacción con la tarea se limitaba a la apertura o cierre de una pinza. (Ollero, A. 2001)

Un robot sin sensores equivale a un sistema manual y no autónomo, los sensores permiten a un robot mantener una estabilidad, tener la retroalimentación de que es lo que va haciendo paso a paso, completar tareas con precisión y disminuir errores.

1.7 Robots Humanoides

En la robótica la actitud de los expertos hacia los autómatas humanoides ha oscilado entre el entusiasmo y el escepticismo. Entusiasmo porque un robot humanoide puede tener enormes ventajas para cierta clase de funciones, escepticismo debido a que una máquina robótica sea útil para dichas

funciones, ya se ha demostrado con ejemplos que la forma humana no es necesaria, y a veces es incluso un estorbo (respecto a las capacidades actuales de los androides).

A medida que el tiempo transcurre los sistemas humanoides avanzan de forma acelerada, hoy en día se tiene una variedad de sistemas que permiten ser adquiridos. No existe la limitación de poder obtener uno de estos sistemas. Cabe recalcar que a medida que estos sistemas aumentan su complejidad en movimientos y estabilidad, así como sus funciones, aumenta su costo en el mercado.

1.8 Aplicaciones de los robots

El objetivo principal de algunos investigadores en robótica es construir robots que se parezcan a las personas, tanto en su cuerpo como en su comportamiento. Sin embargo, hasta ahora, los robots más utilizados en investigaciones robóticas han sido los robots manipulados, los móviles y los robots con patas.

Por otro lado, en los últimos años, la industria japonesa ha realizado tantos avances en la construcción de robots humanoides que hemos de plantearnos cuál podría ser el papel de dichos robots en el futuro de la sociedad robótica.

Cabe destacar que las aplicaciones de estos robots no tiene un límite, ya que pueden trabajar directamente en el mismo entorno que los humanos, sin que se deban realizar modificaciones sobre dicho entorno. Los sistemas de

locomoción para aplicaciones de los humanoides se pueden adaptar al entorno en que se encuentren ejemplo:

- Piso Liso: Humanoide con llantas
- Piso no liso, Escaleras: Humanoide bípedo

(Falomir, Z. 2006)

Según (Falomir, Z. 2006) existen una infinidad de aplicaciones que puede ejecutarse con un robot humanoide, algunas de las aplicaciones que pueden se mencionan a continuación:

- Guiar a personas
- Entretener
- Llevar objetos
- Guiar y explicar
- Realizar discursos
- Vigilancia

1.9 Sistema electrónicos de comunicaciones

Un transmisor es un conjunto de dispositivos o circuitos electrónicos que convierte la información de la fuente original en una señal que se presta más a su transmisión a través de determinado medio de transmisión. El medio de transmisión transporta las señales desde el transmisor hasta el receptor, y puede ser tan sencillo como un par de conductores de cobre que propaguen las señales en forma de flujo de corriente eléctrica. También se puede

convertir la información a ondas electromagnéticas luminosas, propagarlas a través de cables de fibra óptica hechas de vidrio o de plástico, o bien se puede usar el espacio libre para transmitir ondas electromagnéticas de radio, a grandes distancias o sobre terreno donde sea difícil o costoso instalar un cable físico. Un receptor es un conjunto de dispositivos y circuitos electrónicos que acepta del medio de transmisión las señales transmitidas y las reconvierte a su forma original.

(Tomasi, W. 2003)

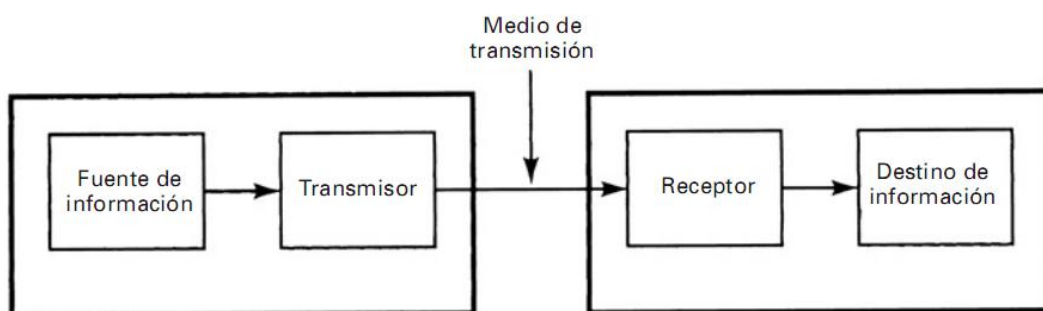


Ilustración 1.1 Diagrama simplificado de bloques de un sistema de comunicaciones electrónicas.

Fuente: (Tomasi, W. 2003)

Según (Tomasi, W. 2003), un sistema de comunicaciones electrónicas está compuesto por un transmisor, un receptor y un medio de transmisión. El presente proyecto emplea el tipo de comunicación infrarroja, en el cual el medio de transmisión corresponde al aire ya que se trata de un tipo de comunicación que emplea ondas electromagnéticas para el envío y recepción de la información. Tanto el transmisor como el receptor están compuestos por circuitos electrónicos conectados a los leds infrarrojos. Se

parte de la línea base que provee el fabricante de Robonova para la comunicación entre el control remoto y el robot.

1.10 Comunicación Infrarroja.

Los sistemas de comunicación infrarroja son aquellos que actualmente se utilizan para la mayoría de mandos de televisores, aparatos de música, reproductores de DVD, etc. Los transmisores utilizan un haz de luz que ilumina en una banda no visible; en particular se utiliza la banda IR de infrarrojos. Al ser comunicaciones donde se utiliza la luz, uno de los mayores inconvenientes de esta tecnología como medio de transmisión es que se necesita iluminar el receptor, lo que implica la necesidad de estar visibles transmisor y receptor o disponer de un camino donde el haz refleje y puede llegar desde el origen hasta el final del destino. No es posible atravesar paredes o cubrir una amplia área de cobertura. Esto supone que las distancias alcanzadas son relativamente pequeñas.

(Vázquez, S. 2015)

La radiación infrarroja es un tipo de radiación mayor que la onda de luz visible y menor que la microondas. La transmisión infrarroja es perfecta para establecer comunicación en enlaces limitados por espacio y obstáculos. Es por eso que este tipo de redes están dirigidas a espacios pequeños como oficinas o escenarios pequeños.

Este tipo de onda son direccionales y no permiten traspasar objetos sólidos, permitiendo tener un direccionamiento y que las ondas no se interfieran

entre sistemas similares en un sitio cercano. Muchas de estas razones hacen que la transmisión infrarroja sean ideales para el uso en sitios reducidos, por ejemplo controles remotos, PDAs, cámaras digitales, celulares, etc.

1.1.9 Comunicación Infrarroja aplicada a robótica móvil

En los medios de comunicación infrarroja, se emplea la emisión infrarroja proveniente de alguna fuente, por ejemplo un diodo emisor infrarrojo, la cual por medios electrónicos es modulada, de una manera tal que permita viajar y ser recibida por un elemento sensible a esta radiación y ser nuevamente convertida en una señal electrónica inteligible.

(López, G. 2003)

El emisor infrarrojo es el encargado de enviar los datos generados por el microcontrolador para ser receptados por el receptor que será el encargado de emitir la información hacia el segundo microcontrolador y cumplir con la tarea asignada.

La modulación de la información se lleva a cabo, haciendo oscilar alguno de los estados lógicos (señales digitales 1 o 0), con una señal cuadrada de frecuencia tal que el filtro pasa banda del receptor conozca la señal que le está enviando el transmisor vía infrarroja.

(López, G. 2003)

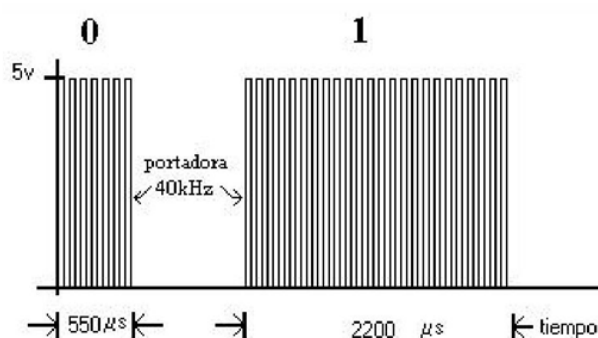


Ilustración 1.2 Modulación de una señal por amplitud

Fuente: (López, G. 2003)

La manipulación de estados lógicos permite agrupar la información que se envía mediante el transmisor infrarrojo el cual deberá ser interpretado de la misma manera por el receptor, por lo cual la modulación debe estar adecuadamente implementada.

Los parámetros a definir para lograr comunicar datos a través de éste método son los siguientes:

- Un protocolo de comunicación, por ejemplo RS-232.
- Seleccionar un método para representar las diferentes señales lógicas, por ejemplo ASK (código de modulación por amplitud).
- Una velocidad de transmisión que permita la respuesta de los componentes del módulo receptor, por ejemplo 1200 bps.
- La frecuencia de la señal portadora, la cual está directamente relacionada con la velocidad de transmisión, por ejemplo 40 kHz.

(López, G. 2003)

Cuando se desea implementar una comunicación infrarroja se deben de tener en cuenta varias condiciones que permitan tener una adecuada transmisión de datos, entre ellas están el elegir el método de representación

de señales, frecuencia de la señal portadora, una adecuada velocidad de transmisión, elección adecuada del dispositivo que genere los estados lógicos y los emisores-receptores infrarrojos. Al emitir una señal de forma directa hacia un foto-dio receptor, se generan voltajes de los cuales se desmonta la información de la portadora, para finalmente realizar una decodificación de la señal emisora. Los dispositivos empleados para la transmisión y recepción de datos se los puede apreciar en la ilustración 5.



Ilustración 1.3 Emisor Infrarrojo (Izquierda) y Receptor Infrarrojo (Derecha).

Fuente Emisor Infrarrojo: (<http://spanish.alibaba.com/product-gs/infrared-remote-control-receiver-module-dqir-38343e8-511243805.html>, 2015)

Fuente Receptor Infrarrojo: (<http://www.steren.com.mx/catalogo/prod.asp?p=2059>, 2015)

1.11 Software Robobasic

Para realizar cada uno de los pasos de baile, se procede a utilizar herramientas del software roboBASIC v2.5 previstas en el kit de Robonova I. (ROBOTICS, H. 2005)

RoboBASIC es un software de programación de tipo secuencial con instrucciones que permiten realizar el movimiento de cada uno de los servomotores presentes en el robot.

1.12 Software Labview

Labview (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) es un entorno de programación gráfica que permite de una forma práctica desarrollar sistemas de control y monitoreo HMI (inteface hombre máquina), debido a que contiene un gran número de bloques de código que permiten su integración. (Arevalo, P. 2009)

Labview a nivel mundial es uno del software de más uso en el ámbito de la investigación y desarrollo, debido a la facilidad de integración entre distintas plataformas robóticas, de tipo industrial, escolar y con la mayoría de hardware que se puede conseguir en el mercado de la electrónica. Además permite la integración por medio de los protocolos de comunicación, tales como: PXI, TCP/IP, USB, Serial RS232 que es el empleado en este proyecto.

El desarrollador en el ambiente Labview dispone de dos tipos de ventana, la principal que es el Panel Frontal y la secundaria que es el Diagrama de Bloques. La primera es la que el usuario final podrá ver y controlar, es el prácticamente el producto final, y la segunda ventana es donde se desarrolla el código fuente, que es bajo un lenguaje gráfico.

1.13 Software Bascom Avr

La gran gama de microcontroladores de Atmel que existen en el mercado pueden ser programados en distintos lenguajes, BASCOM AVR es un compilador destinado para este tipo de microcontroladores, que mediante un

set de instrucciones en lenguaje Basic, pueden ser compilados y cargados al sistema Microcontrolado.

(ATMEL, C. 2015)

Bascom AVR posee sus distintas barras de herramientas, iconos de programación y la parte fundamental que es el lugar para el código fuente. Además posee una gran variedad de librerías que permiten escoger de la gama de microcontroladores el idóneo conforme al proyecto que se desee implementar.

1.14 Software Proteus ISIS

Proteus es un software de simulación y compilación para circuitos electrónicos que permiten al desarrollador crear tarjetas electrónicas en base a diagramas circuitales. (Labcenter, E. 2015)

Proteus contiene dos ventanas principales que son ISIS y ARES. La primera con sus siglas Intelligent Schematic Input System, permiten diseñar el diagrama circuital de toda la tarjeta electrónica. Este software contiene librerías que permiten simular los circuitos y poder determinar si existen fallas antes de su construcción.

CAPÍTULO II

Metodología

2.1 Metodología de Investigación

En el presente proyecto se empleó la siguiente metodología de Investigación.

2.1.1 Tipo de Investigación

Por el lugar: Como medio de obtención de información, se trata de una investigación documental, la cual emplea fuentes bibliográficas científicas para una mejor comprensión de cada una de las fases involucradas en el desarrollo teórico y práctico del proyecto.

Por el propósito: Se trata de una investigación mixta, en la cual se aplicó dos tipos: una investigación básica y una aplicada. Básica, debido a que se emplea un marco teórico conocido y en base a teorías existentes, desarrollar nuevos conocimientos prácticos aplicables a la implementación del proyecto.

Por el alcance: Se trata de una investigación descriptiva, ya que se caracteriza el objetivo y se sistematiza cada uno de los componentes involucrados en la implementación del proyecto.

2.1.2 Fuentes de Información

2.1.2.1 Primarias

La información de fuentes primarias, se encontró en archivos, documentos, y recopilación de información a través de la observación directa en la Pontificia Universidad Católica del Ecuador Sede Ambato.

2.1.2.2 Secundarias

Para fuentes secundarias del presente trabajo de investigación se obtendrá de libros, páginas web, revistas especializados en robótica, diseño e innovación de proyectos nuevos, importancia de la robótica, just in time, optimización de recursos, la eficiencia y la eficacia de la utilización de robot, y demás términos relacionados con el tema de estudio en la presente investigación.

2.1.3 Técnicas de Investigación

La técnica a ser utilizada en el trabajo de investigación será:

2.1.3.1 Observación Científica Estructurada

Esta técnica cuenta con un proceso de investigación que será utilizado para registrar la información resultante de la observación directa, ayudándose de elementos técnicos apropiados para llegar a obtener resultados apropiados en el desarrollo del proyecto.

2.2 Metodología de desarrollo

Como proceso de elaboración o desarrollo se emplearon distintas metodologías que permiten tener el control y funcionamiento del sistema Robonova I con las características de movimientos y giros. Dichas fases son las siguientes:

- Armado de robot Robonova I
- Proceso de programación para realizar un movimiento coordinado.
- Diseño de módulo infrarrojo.
- Desarrollo de interface gráfica en software Labview.
- Captura de tren de pulsos infrarrojos.
- Código de programación de Microcontrolador.

Para el desarrollo de este proyecto por su naturaleza de construcción, elaboración, desarrollo de aplicación de software, que confluyen en un solo proyecto se propone una mezcla de metodologías tanto para la construcción como para el desarrollo de software, la que determine en las siguientes fases:

FASE I Armado de robot Robonova I.- Se aplica la metodología de construcción propuesta por el proveedor del KIT, la que se describe en el ítem 3.2.

FASE II.- Proceso de programación para realizar un movimiento coordinado.- En esta fase se utiliza la metodología, aprobación y funcionamiento del Software provistos por el proveedor acoplado los requerimientos del proyecto, se describe con detalle en el ítem 3.3.

FASE III.- Diseño de módulo infrarrojo.- Se aplica una metodología CAD y un proceso de desarrollo que se describe en el ítem 3.4.

FASE IV.- Desarrollo de interface gráfica en software Labview.- Se aplica un ciclo corto de la metodología de desarrollo en CASCADA, con un alta rigurosidad en el cumplimiento de las fases, por las características del proyecto, se describe en el ítem 3.5.

FASE V.- Captura de tren de pulsos infrarrojos.- Se utiliza una metodología de experimentación de punto que se realiza en sitio con varias pruebas y el ejercicio de programación de los resultados, ítem 3.6

FASE VI.- Código de programación de Microcontrolador

CAPÍTULO III

Resultados

3.1 Generalidades de Robonova I

El kit Robonova I se empleó para la realización del presente trabajo, el cual es un robot de tipo humanoide y tiene la capacidad de poder emular movimientos complejos que permite realizar las rutinas de baile que se plantea.

Este robot puede ser operado remotamente desde su respectivo control remoto o desde una PC gracias a la tarjeta electrónica diseñada y construida que se explicará más adelante.

Además es una plataforma de desarrollo y comprensión del funcionamiento del sistema mecánico y electrónico ya que se cuenta con las herramientas de software para programar cada uno de los movimientos.

3.2 Fase I: Armado de Robonova I

El kit Robonova I consta de un conjunto de componentes que fueron armados en orden tomando en cuenta el sentido de movimiento de servomotores así como también la estructuración del cableado eléctrico hacia la tarjeta de mando y control.

Para realizar el armado se va seleccionando componente a componente, se verifica los acoplamientos de acuerdo al manual de armado del kit robonova.

En este caso se debe tener en cuenta la manera como el cableado ira siendo asegurado a cada una de las partes del robot.

Cada uno de los servomotores deberá ser ubicado entre las articulaciones del robot, tomando en cuenta la longitud de su cable de alimentación y su sentido de polarización.

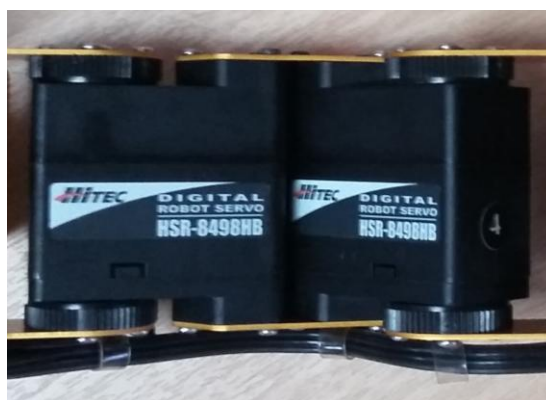


Ilustración 3. 1 Servomotores del kit robonova I.

Fuente: (Autor, 2015)

Se debe tomar en cuenta la distribución de cada uno de los pines de conexión del servomotor y la placa, para ello se verificó de manera adecuada los pines: Gnd (Primer cable negro), +Vcc (Segundo cable negro), y señal de control (Tercer cable color plomo),. Esto se puede apreciar en la siguiente figura:



Ilustración 3. 2 Distribución de señales hacia el servomotor

Fuente: (Autor, 2015)

A continuación se muestra algunas ilustraciones que muestran el armado del robot.

Para empezar se realizó el reconocimiento de cada uno de los componentes del Kit Robonova I, de esta manera se armó desde la parte inferior del robot hasta llegar a su parte superior de acuerdo al orden de numeración de las piezas y utilizando los tornillos adecuados.

Durante todo el proceso de armado se fué verificando la libertad de movimiento de cada pieza así como también que cada uno de los tornillos se encuentre correctamente ajustado.

Para realizar la armazón del robot, se empezó por armar desde los pies a la cabeza.



Ilustración 3. 3 Ubicación de los componentes para armado de las piernas del robot.
Fuente: (Autor, 2015)

La parte de la espalda corresponde al lugar en donde se colocó la tarjeta de control para su posterior conexión a cada uno de los servomotores y batería de alimentación. Una vez finalizada se deberá tener una distribución de componentes unidos como se muestra en la figura.

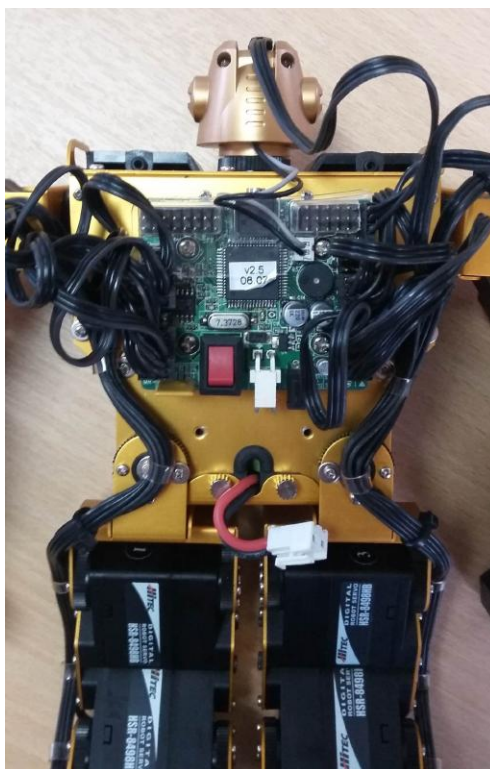


Ilustración 3. 4 Robot armado completamente y conectado con la placa de control.
Fuente: (Autor, 2015)

Todo los servomotores fueron conectados de acuerdo al número del grupo que pertenece. Con esto se podrá guiar cuando se programen los movimientos de cada una de las subrutinas de baile.

3.2.1 Servos

El Robonova I está formado por 16 servos digitales (HSR 8498HB) que cuentan con una característica especial llamada *Motion Feedback*, la cual permite leer externamente la posición real del servo para poder colocar el

robot manualmente en cualquier posición y luego leer y guardar la posición. Esto se hizo con los 16 servomotores desde el mismo controlador.

Las características técnicas del servomotor HSR 8498HB son:

- Velocidad para 60°: 0,2 s a 6V / 0,18 s a 7,4V.
- Fuerza: 7,4 Kg/cm a 6V/ 9 Kg/cm a 7,4V.
- Dimensiones: 40 x 20 x 47 mm. Peso 55 gramos.
- Engranajes: Karbonite.



Ilustración 3. 5 Servomotores HSR 8498HB.
Fuente: (ROBOTICS, H. 2005)

3.2.2 Cable de conexión serie

El cable de conexión serie tiene una longitud de 1,8 metros y se emplea para programar el robot desde el computador. El cable tiene un jack de 3,5 mm estéreo en un extremo y un canon hembra de 9 pines en el otro. Este cable deberá ser conectado a un conversor usb-serial para poder conectar al puerto usb del computador.



Ilustración 3. 6Cable de programación del robot.

Fuente: (Autor, 2015)

3.2.3 Microcontrolador

La tarjeta de control (MR C-3024) del Robonova I está constituida por un microcontrolador Atmel ATmega 128, con 40 puertos de entrada y salida digitales, puerto serie, bus I2C y 8 entradas analógicas. La gran cantidad de puertos permiten la conexión de los servos así como también otros dispositivos como sensores de distancia, giróscopos, displays LCD, sensores de infrarrojos, etc. ROBOTICS, H. (2005).

Esta placa también cuenta con un altavoz para generar tonos de diferentes frecuencias y un conector para un led que se lo puede controlar sin restricciones y cuenta con 64 Kbyte de memoria para los programas, que permiten que una vez que se han descargado, el robot sea completamente autónomo. ROBOTICS, H. (2005).

Según ROBOTICS, H. (2005), sus características principales son:

Alto rendimiento:

- Microcontrolador AVR 8-bit de bajo consumo

Arquitectura:

- 130 Instrucciones
- 32 x 8 Registros de Propósito General

Alta durabilidad de Memoria No Volátil:

- 8K Bytes Flash p
- 512 Bytes EEPROM
- 1K Byte Internal SRAM
- Ciclos de Borrado: 10,000 Flash/100,000 EEPROM

Periféricos

- Dos Temporizador/Contador de 8-bit con pre escala separada y modo de comparación
- Un Temporizador/Contador de 16-bit con pre escala separada y modo de comparación y captura
- Contador de Tiempo real con Oscilador separado
- Tres canales PWM
- 8-canales ADC
- Interfaz serie de dos cables orientado a byte
- USART Programable serie
- Interfaz serie SPI Maestro/Esclavo
- Watchdog programable
- Comparador analógico

Entrada/Salida

- 23 líneas de entrada/salida programables

Voltajes de operación

- 2.7 - 5.5V (ATmega8L)

- 4.5 - 5.5V (ATmega8)

Velocidad de procesador

- 0 - 8 MHz (ATmega8L)

- 0 - 16 MHz (ATmega8)

Consumo de energía a 4 MHz, 3V, 25°C

- Activo: 3.6 mA

- Reposo: 0.5 μ A

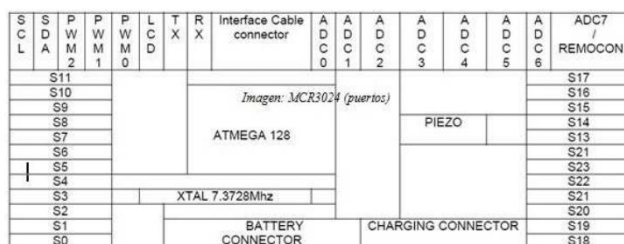


Ilustración 3. 7 Distribución de los puertos de la placa MCR3024.

Fuente: (ROBOTICS,H. 2005)

La tarjeta de control del robot fue diseñada para controlar cada una de las extremidades del robot. Para ello se le ha procedido a distribuir en grupos etiquetados.



Ilustración 3. 8 Placa MCR3024.

Fuente: (ROBOTICS, H. 2005)



Ilustración 3. 9 Ubicación de la Placa MCR3024 en la espalda del robot.
Fuente: (Autor, 2015)

Una vez finalizado el armado se procede a reconocer cada uno de los sockets de conexión ubicados en la parte posterior del robot.

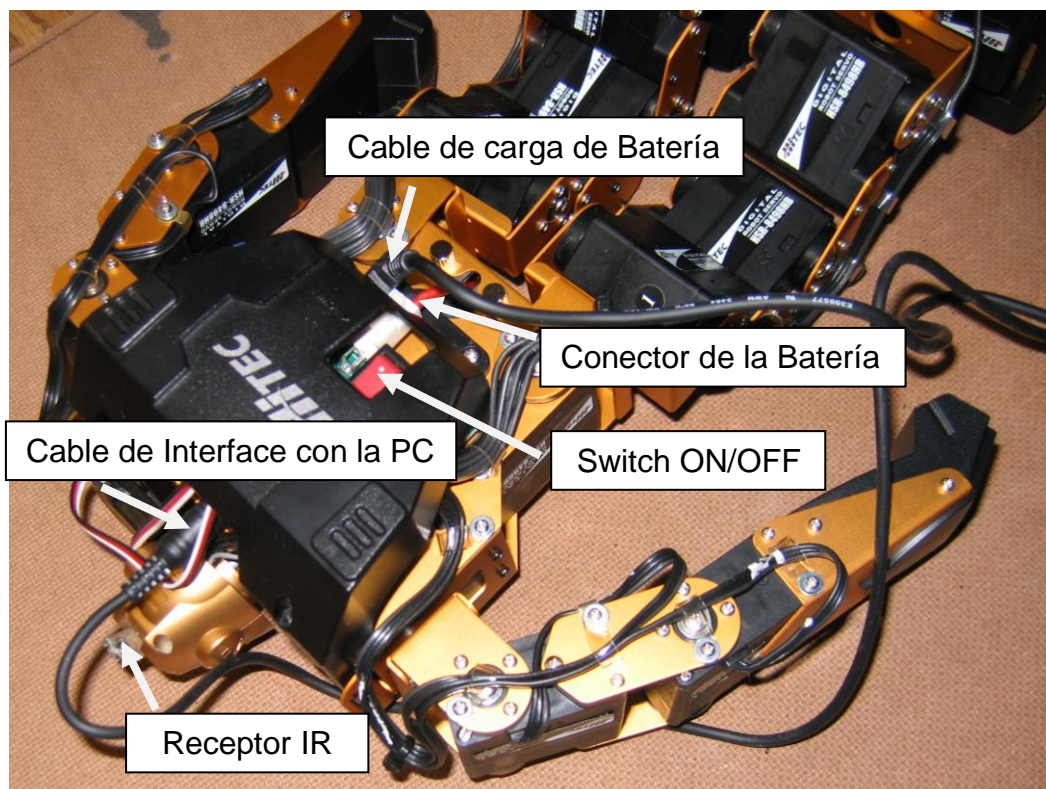


Ilustración 3. 10 localización de elementos principales.
Fuente: (Autor, 2015)

3.3 Fase II: Proceso de programación para realizar un movimiento coordinado.

Para realizar esta tarea se utilizó la metodología de desarrollo en cascada empleando para esto varias herramientas, la primera se llama ROBONOVA Zero Set la cual permite ajustar el cero de cada uno de los servomotores. Con esta herramienta se dio el movimiento inicial del robot cuando fue energizado. En este caso el paso inicial del robot es que se mantenga de pie.

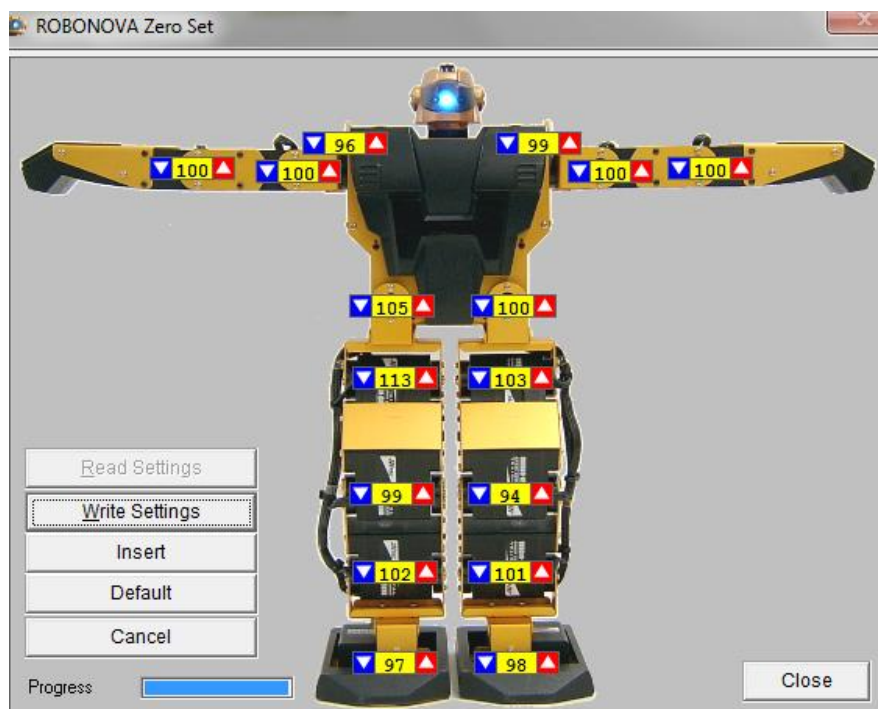



Ilustración 3. 11 Ventana de configuración del zeros de los servomotores.

Fuente: (Autor, 2015)

El zero de los servomotores debe crear una pose del robot de pie y con los brazos abiertos como se muestra en la ilustración. Las flechas encerradas en cuadros azules y rojos permitieron mover hasta encontrar el equilibrio adecuado entre cada uno de los servomotores. Una vez obtenido esta

postura se procede a enviar los datos hacia el robot mediante la opción *Write Settings*. Una vez que la barra de progreso termine, se presiona *Close*.

La segunda herramienta se trata de *ROBONOVA Motor Control* que permite realizar combinaciones secuenciales para ser ingresadas en el roboBASIC v2.5.

Para abrir la ventana *ROBONOVA Motor Control* se presiona el botón  o mediante la pestaña *Controller* y opción *ROBONOVA Motor Control*.

Cuando el Robonova se encuentra conectado y encendido, esta ventana permite interactuar con cada uno de los motores.

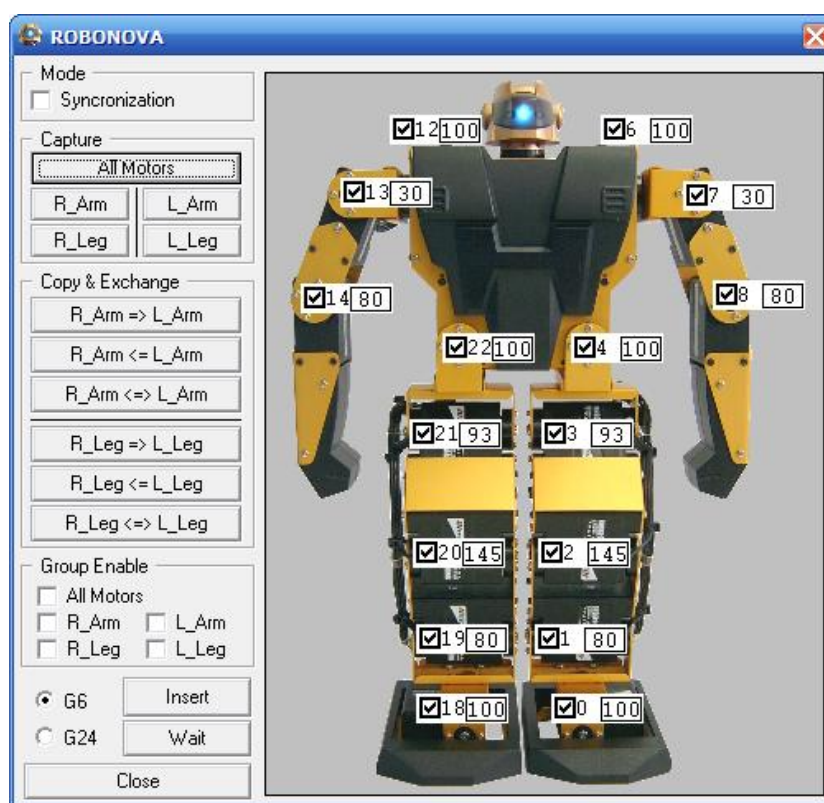


Ilustración 3. 12 Ventana ROBONOVA Motor Control.

Fuente: (Autor, 2015)

Ubicando el mouse sobre cada uno de los servomotores, el botón con un slider de seteo aparece. Con un click se procedió a mover el motor en un

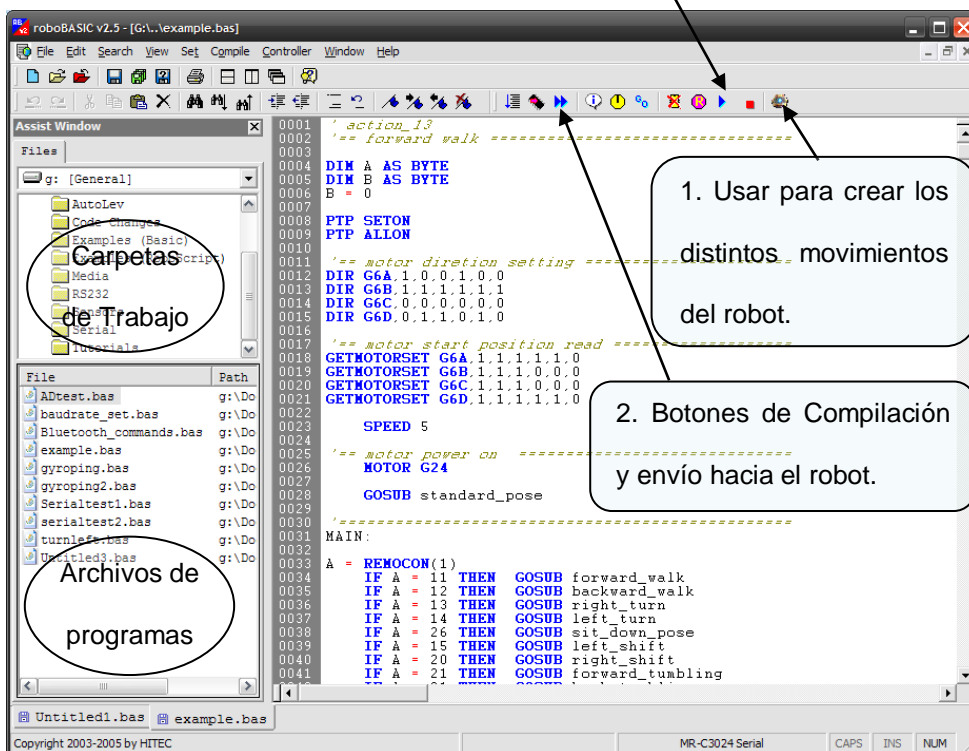
grado y dando un doble click cinco grados. Moviendo directamente el slider se puede mover más rápido la posición del motor. Para capturar e ingresar el movimiento en la ventana del robonova se presionó la opción *INSERT*. Haciendo una combinación de posiciones repetidamente se puede crear una serie de movimientos que el robot realizará.

Para hacer movimientos repetitivos, los botones de la opción *copy & exchange* permite duplicar o intercambiar las posiciones de brazos y piernas con lo cual poniendo únicamente la posición de un lado del cuerpo, se obtendría simetría en el otro lado.

Cada una de estas combinaciones fue ingresada mediante la opción *INSERT* en la ventana de robonova para su posterior estructuración.

Antes de entrar en el proceso de estructuración del programa, se va a explicar cada uno de las partes que comprende la ventana de roboBASIC v2.5:

3. Botones para empezar, detener o resetear el programa y limpiar la memoria del controlador.

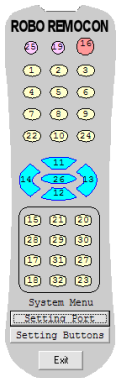


1. Usar para crear los distintos movimientos del robot.

2. Botones de Compilación y envío hacia el robot.

Carpetas de Trabajo

Archivos de programas



4. El RoboRemoCon, o el Control remoto IR permite controlar el Robonova y probar los movimientos programados.

Ilustración 3. 13 Ventana de programación robo BASIC V2.5 y sus componentes principales.

Fuente: (Autor, 2015)

En esta ventana se estructura cada una de las condiciones y movimientos de cada uno de los ritmos de baile, poner velocidad y dirección de cada uno de los movimientos así como realizar el programa secuencial para ser transmitido al robot mediante comunicación serial.

En general la estructura de un programa consiste de 3 partes:

1. Inicialización

- a. Encender los motores con MOTOR G24
- b. Setear la dirección de los motores (típicamente direcciones por defecto) con el comando DIR
- c. Leer posiciones del motor (MOTORSET)
- d. Setear la velocidad inicial con el comando SPEED
- e. Setear PTP ON u OFF (dependiendo de la aplicación)

2. Lazo MAIN (simple)

- a. Leer la fuente de comandos de movimiento que proviene ya sea del control remoto o de la conexión serie.
- b. Analizar la entrada de comandos (ya sea caracteres ASCII del puerto serial, o un número de botón del mando a distancia).
- c. Usar estructuras IF u otro tipo de estructuras para ejecutar los comandos específicos.

3. Subrutinas

- a. Las subrutinas son como un marcador en el código. El comando GOSUB acompañado de un nombre, salta al código en donde se tiene el nombre que acompañaba a la instrucción GOSUB y continúa la ejecución.
- b. Los comandos MOVE típicamente se ejecutan en subrutinas, y básicamente indican las posiciones que deberán adoptar cada uno de los motores. Además van acompañados por el comando SPEED que setea la velocidad de respuesta de los motores.

- c. Siempre se deberá terminar una subrutina con el comando RETURN o END, el cual le dice al programa que regrese al punto donde fue llamada la subrutina, o finaliza el programa.

El programa generado para cumplir con todos los movimientos de baile se lo presenta en el ANEXO 1. (Autor, 2015)

3.4 Fase III: Diseño del módulo infrarrojo.

3.4.1 Metodología de desarrollo de hardware.

La metodología empleada para el diseño del hardware corresponde a la asistida por computadora (Computer Aided Design, CAD) la cual emplea técnicas gráficas para soportar el proceso de diseño. Esta técnica consiste en poder realizar simulaciones para verificar el diseño y funcionamiento del circuito para su posterior implementación, logrando de esta manera minimizar costos por fallas que serán detectadas por el software.

3.4.2 Generación de Pulsos en comunicación infrarroja

Robonova cuenta con una tarjeta electrónica MR C-3024 propia de HITEC, que controla el movimiento de cada uno de los 16 servos, mediante comando o subrutinas previamente programados en su memoria. Para no perder el uso de esta tarjeta electrónica, se mantuvo el esquema de controlar al robot por medio de un control remoto externo.

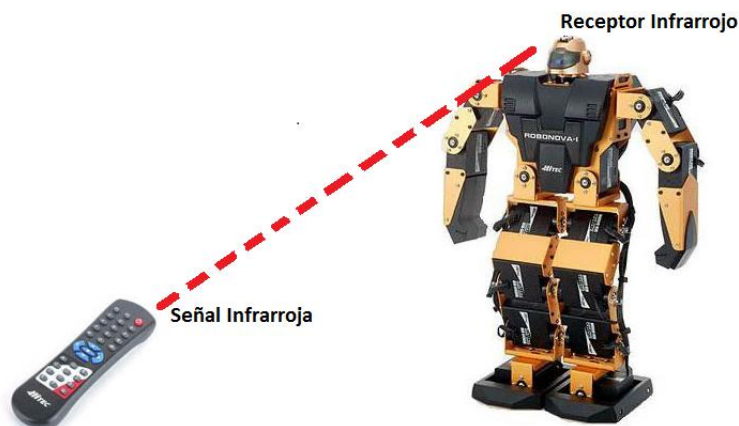


Ilustración 3. 14 Comunicación entre control remoto y Robot Robonova.
Fuente: (ROBOTICS, H. 2005)

Al tener un control remoto propio de la marca HITEC, se decidió suplantar este control con un módulo, que generara las mismas señales de control. Y que la tarjeta MR C-3024 se pueda sincronizar con el mismo.

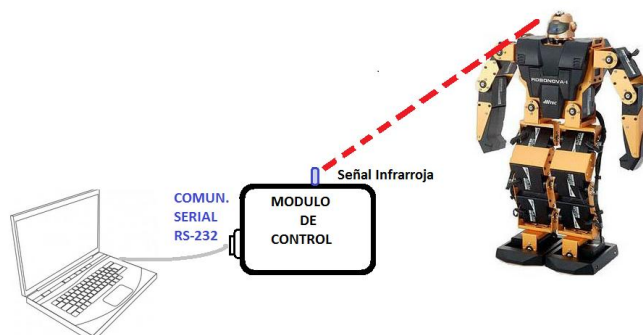


Ilustración 3. 15 Comunicación entre PC y Robot Robonova.
Fuente: (Autor, 2015)

Este nuevo módulo de control, permite al usuario comunicarse entre el computador y el robot, para cual hace una conversión de protocolos de comunicación. Dentro de este proceso de conversión esta como primer paso establecer una comunicación entre el usuario y el computador, para lo cual se empleó una interface gráfica desarrollada en lenguaje “G” mediante la plataforma gráfica Labview, una vez establecida la comunicación hombre-

máquina, se emite las señales al nuevo módulo de control, el cual mediante un sistema microprocesado, recepta las señales RS-232 que provienen del computador y son procesadas para luego transformar a señales infrarrojas.

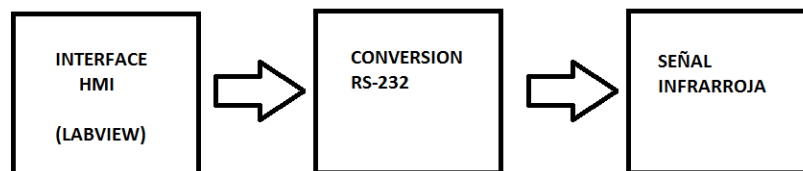


Ilustración 3. 16 Conversión de Protocolos de Comunicación.
Fuente: (Autor, 2015)

3.4.3 Desarrollo de interface gráfica en Labview

Una de las herramientas hoy en día empleadas para desarrollo de aplicaciones de pruebas y medidas es LabView (Laboratory Virtual Instrumentation Engineering Workbench), debido a su fácil uso e incorporación en interfaces gráficas de control.

Labview cuenta con herramientas multimedia para poder manipular controles y reproducir archivos de audio de cualquier extensión, como es el formato .mp3.

El desarrollar en un entorno gráfico de LabView, permite usar íconos gráficos con tan solo un clic y arrastrar los gráficos, además de la facilidad de escribir líneas de texto.

A los programas elaborados en Labview se los conoce como Instrumentos Virtuales (VIs), debido a que se derivan del concepto de control de instrumentos.

Otro de los aspectos en los que ayuda la herramienta LabView es la reducción de tiempo en el desarrollo de las aplicaciones, además permite la integración con cualquier tipo de software y hardware ya sea propios de National Instrument o de otros fabricantes.

Labview permite mediante la interface desarrollada establecer un vínculo entre el usuario y el computador, que para este caso es escoger o seleccionar las canciones para el aplicativo.

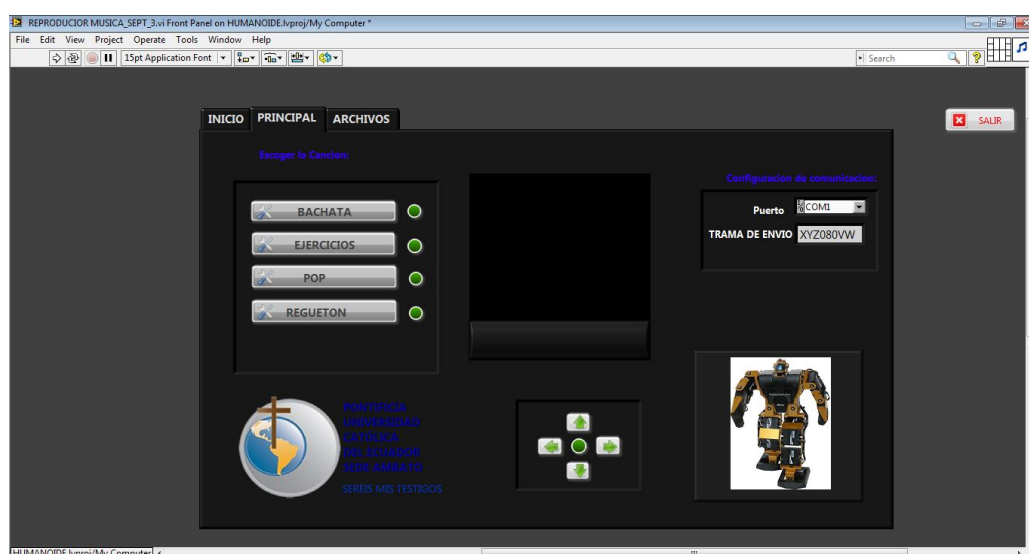


Ilustración 3. 174 Interfaces Desarrolladas en Labview.
Fuente: (Autor, 2015)

La aplicación desarrollada genera hacia el exterior un tramo de control vía comunicación serial RS-232, la cual es adquirida por el módulo de control implementado.

3.4.4 Módulo de control

Este módulo contiene una tarjeta electrónica que recepta las señales provenientes de la interface desarrollada en el computador vía comunicación

serial RS-232. Internamente el módulo contiene la tarjeta electrónica provista de un Microcontrolador encargado de convertir estas órdenes recibidas y transformarlas al protocolo de comunicación infrarrojo.

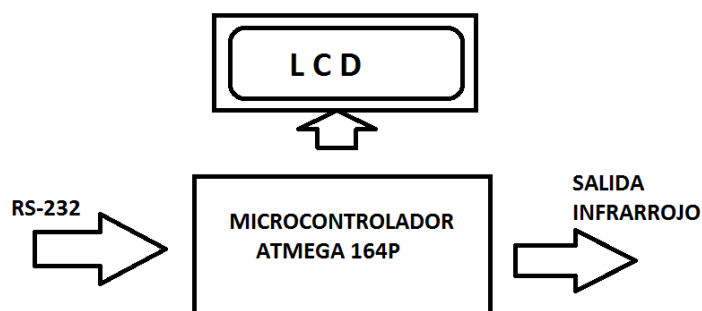


Ilustración 3. 18 Diagrama de Bloques del módulo de control.

Fuente: (Autor, 2015)

Se determinó que una buena opción es un microcontrolador ATMEGA 164p como cerebro para recibir las señales seriales, manejo de LCD de visualización, etc., debido a la disponibilidad de herramientas para desarrollo, además de su bajo costo y la facilidad de adquisición en el mercado local.

El microcontrolador Atmega 164p es un microcontrolador a 8 bits con 16 Kbytes de memoria flash programable, 512 Bytes de memoria EEPROM, 1K byte de memoria RAM, 40 pines tipo PDIP, 8 canales ADC a 10 bits, seis canales de salida PWM, 32 líneas de I/O, 32 registros de propósito general. Posee Watchdog Timer con oscilador interno de 1MHz, 2MHz, 4MHz y 8 MHz, RTC interno con cristal de 32768 Hz, dos puertos de comunicación serial USART, un puerto serial SPI.

(PCINT8/XCK0/T0) PB0	1	40	PA0 (ADC0/PCINT0)
(PCINT9/CLKO/T1) PB1	2	39	PA1 (ADC1/PCINT1)
(PCINT10/INT2/AIN0) PB2	3	38	PA2 (ADC2/PCINT2)
(PCINT11/OC0A/AIN1) PB3	4	37	PA3 (ADC3/PCINT3)
(PCINT12/OC0B/SS) PB4	5	36	PA4 (ADC4/PCINT4)
(PCINT13/MOSI) PB5	6	35	PA5 (ADC5/PCINT5)
(PCINT14/MISO) PB6	7	34	PA6 (ADC6/PCINT6)
(PCINT15/SCK) PB7	8	33	PA7 (ADC7/PCINT7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2/PCINT23)
XTAL1	13	28	PC6 (TOSC1/PCINT22)
(PCINT24/RXD0) PD0	14	27	PC5 (TDI/PCINT21)
(PCINT25/TXD0) PD1	15	26	PC4 (TDO/PCINT20)
(PCINT26/RXD1/INT0) PD2	16	25	PC3 (TMS/PCINT19)
(PCINT27/TXD1/INT1) PD3	17	24	PC2 (TCK/PCINT18)
(PCINT28/XCK1/OC1B) PD4	18	23	PC1 (SDA/PCINT17)
(PCINT29/OC1A) PD5	19	22	PC0 (SCL/PCINT16)
(PCINT30/OC2B/ICP) PD6	20	21	PD7 (OC2A/PCINT31)

Ilustración 3. 19 Distribución de pines del microcontrolador ATMEGA 164p.
Fuente: (ATMEL, 2015)

3.4.5 Generación de tren de pulsos infrarrojos

Robonova es controlado por medio de su propio control remoto, previa compilación de los algoritmos de movimientos compatibles con cada una de las teclas del control remoto. Como por ejemplo si se presiona las teclas de navegación encontradas en la parte media del control, permitirá mover al robot en las direcciones establecidas.



Ilustración 3. 20 Control Remoto.
Fuente: (ROBOTICS, H. 2005)

Al presionar cada una de las teclas del control remoto el diodo infrarrojo ubicado en dicho control genera un tren de pulsos, los cuales fueron extraídos por medio de un osciloscopio para saber la variación de cada uno

de ellos. En los extremos del diodo infrarrojo se ubicó las puntas de prueba del osciloscopio obteniendo las tiempos en alto y bajo de cada una de variaciones del tren de pulsos.

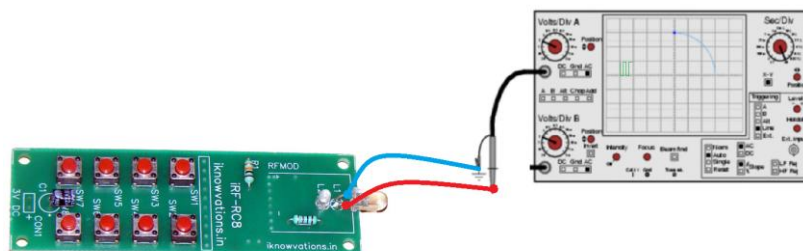


Ilustración 3. 21 Visualización de los pulsos.
Fuente: (Autor, 2015)

3.4.6 Diseño Electrónico

El diseño electrónico agregado principalmente a este trabajo ha sido la construcción de una tarjeta que permita emular el control propio del kit robonova. Para realizar esta tarea, se empleó la metodología CAD explicada en el capítulo II y para esto se tuvo que realizar el análisis del protocolo de comunicación que utiliza el kit Robonova, para posterior a esto simular en software y una vez verificado, construir el hardware que permita enviar los pulsos de información adecuados al robot.

Esta tarjeta además de transmitir los pulsos, permite realizar una comunicación desde una PC para escoger el ritmo a bailar así como también realizar otros movimientos del robot mediante una interface gráfica diseñada mediante la herramienta LABVIEW.



Ilustración 3. 22 Foto del módulo infrarrojo construido.
Fuente: (Autor, 2015)

Para poder comunicarse entre el módulo y el computador se emplea un circuito integrado MAX 232 que es un conversor de niveles TTL (que entrega el Microcontrolador) a niveles RS 232 que entrega el computador, Este chip permite la comunicación entre el PC y la tarjeta electrónica externa.

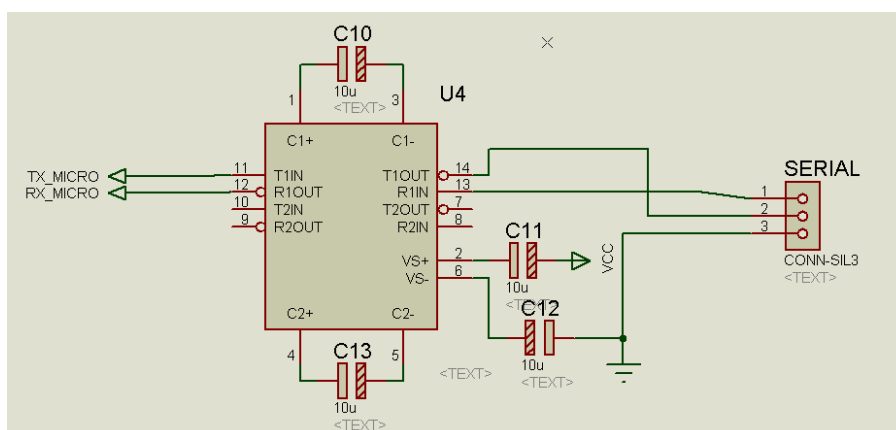


Ilustración 3. 23 Conexión del circuito integrado MAX232.
Fuente: (Autor, 2015)

Para poder generar los niveles RS 232 se colocan 4 capacitores de 10uF, que son recomendados en la hoja técnica de este chip

Una vez que ingresa la señal RS 232 y transformada a niveles TTL (5 vdc) se debe procesar o interpretar esta señal, para realizar esta actividad se emplea el Microcontrolador atmega 164 que mediante sus pines de comunicación (RX_micro y TX_micro) permite la comunicación.

Una vez que el Microcontrolador adquiere la señal proveniente de la computadora, este microcontrolador establece una lógica entre la orden que recibe y el comando que debe generar hacia el led infrarrojo.

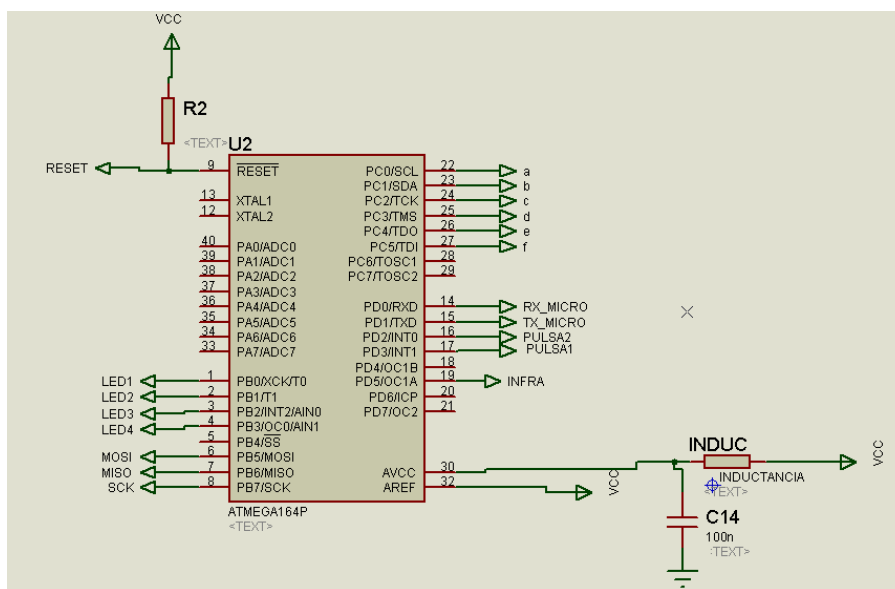


Ilustración 3. 24 Conexión del microcontrolador.

Fuente: (Autor, 2015)

Para poder cargar esta lógica, el Microcontrolador posee 6 pines de programación mediante el canal ISP, Este archivo se genera mediante el programa compilador BASCOM AVR.

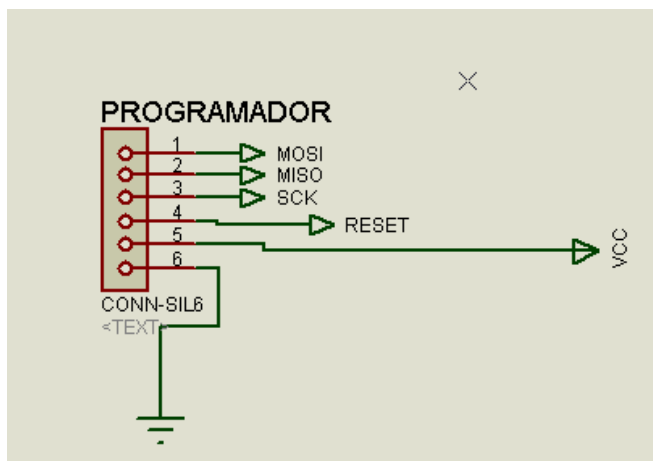


Ilustración 3. 25 Pines de programación del microcontrolador.
Fuente: (Autor, 2015)

Para poder visualizar o tener un control de los datos que llegan al sistema se empleó un LCD de 16 x 2 caracteres.



Ilustración 3. 26 Pantalla LCD para visualización de órdenes.
Fuente: (Autor, 2015)

Para poder visualizar los datos o los caracteres en la pantalla LCD, esta debe ser controlada por el Microcontrolador por lo cual deben estar conectados de la siguiente manera:

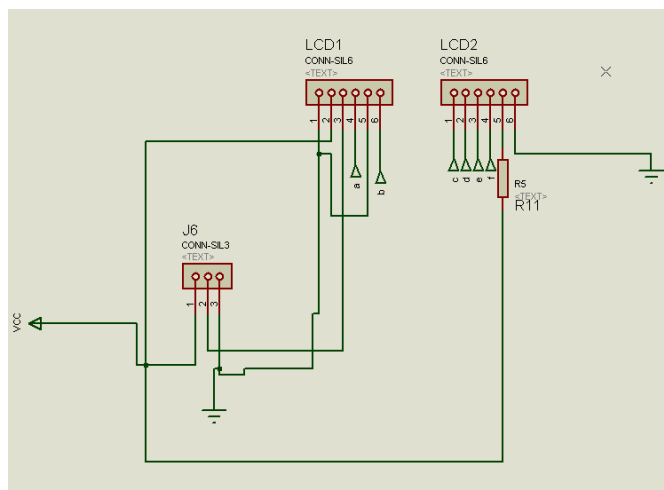


Ilustración 3. 27 Diagrama de conexión del Microcontrolador con LCD.
Fuente: (Autor, 2015)

Al recibir que tipo de canción se está reproduciendo, el módulo genera la señal hacia el led infrarrojo.

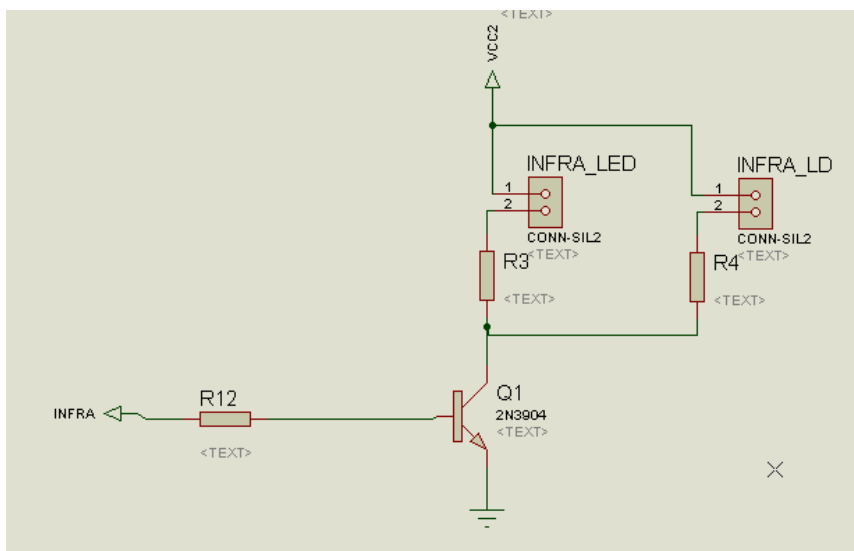


Ilustración 3. 28 Diagrama de conexión para led infrarrojo.
Fuente: (Autor, 2015)

Para poder identificar de una forma visual que ritmo ha sido escogido el módulo posee cuatro leds de alto brillo ubicados en la caja de protección del módulo infrarrojo.

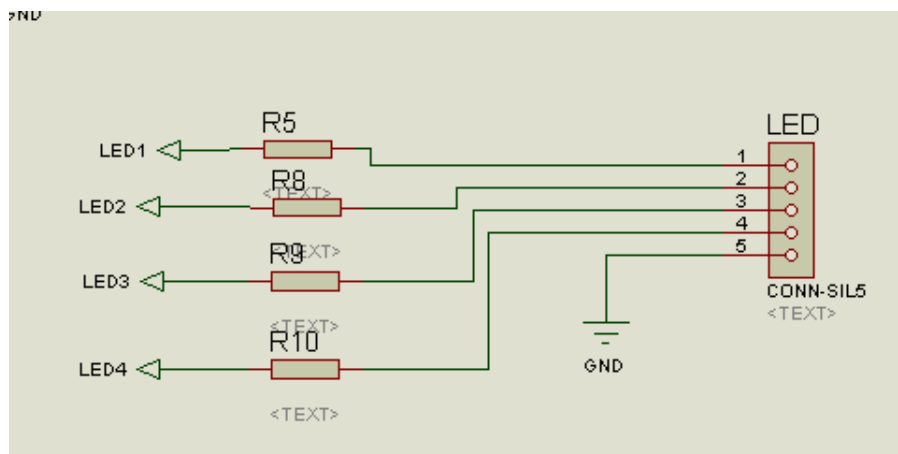


Ilustración 3. 29 Diagrama de conexión de leds.
Fuente: (Autor, 2015)

Para que el sistema pueda funcionar necesita alimentación y sus respectivas conversiones de voltajes. Al sistema le ingresan 12 Vdc por medio de un adaptador que transforma la energía de 120 vac a 12 vdc.

Internamente la tarjeta cuenta con dos reguladores de voltaje 7805, estos reguladores convierten de 12 vdc a 5 vdc, para que funciones los sistemas de LCD, Microcontrolador, leds, led infrarrojos, etc.

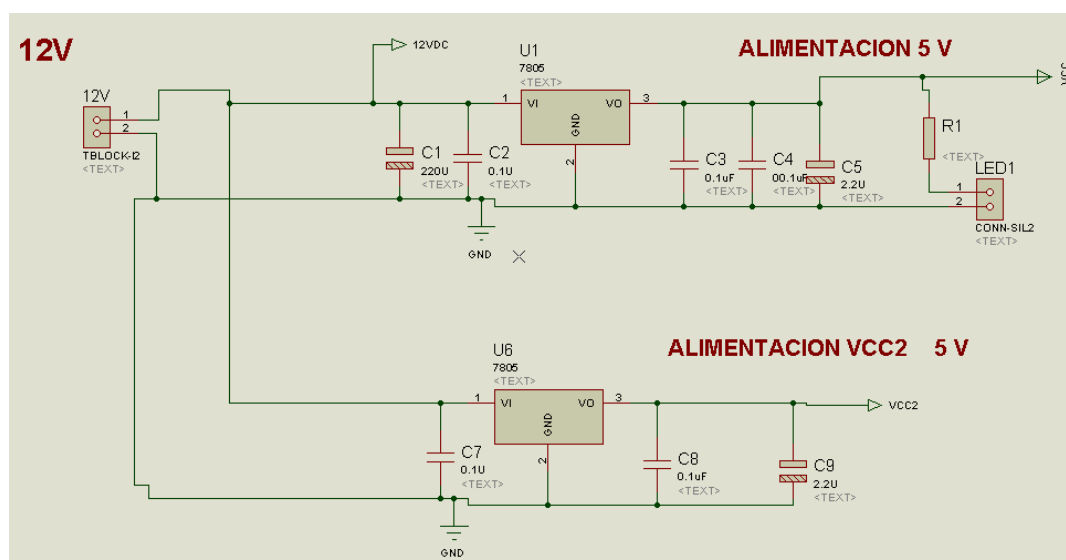


Ilustración 3. 30 Fuentes de alimentación de 5V DC.
Fuente: (Autor, 2015)

Con los diagramas levantados en el software Proteus ISIS, se emplea la herramienta ARES de Proteus para realizar el PCB de la placa electrónica.

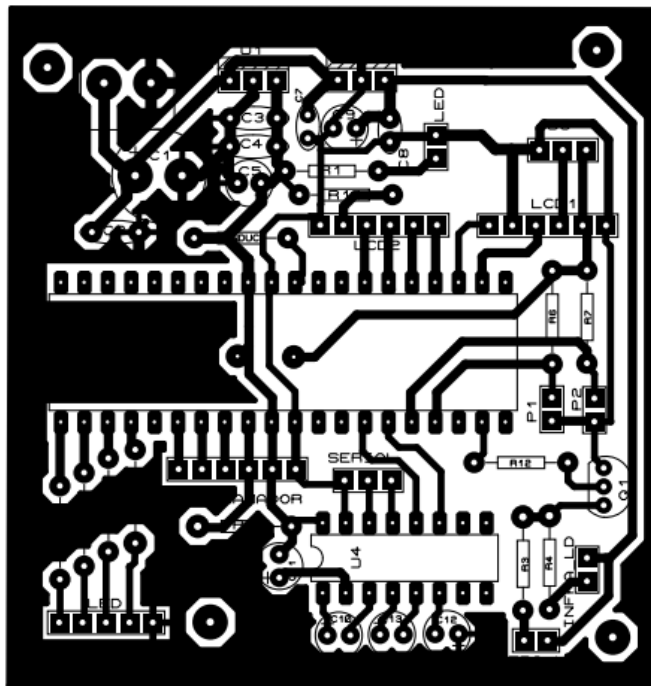


Ilustración 3. 31 PCB de tarjeta electrónica de control.
Fuente: (Autor, 2015)

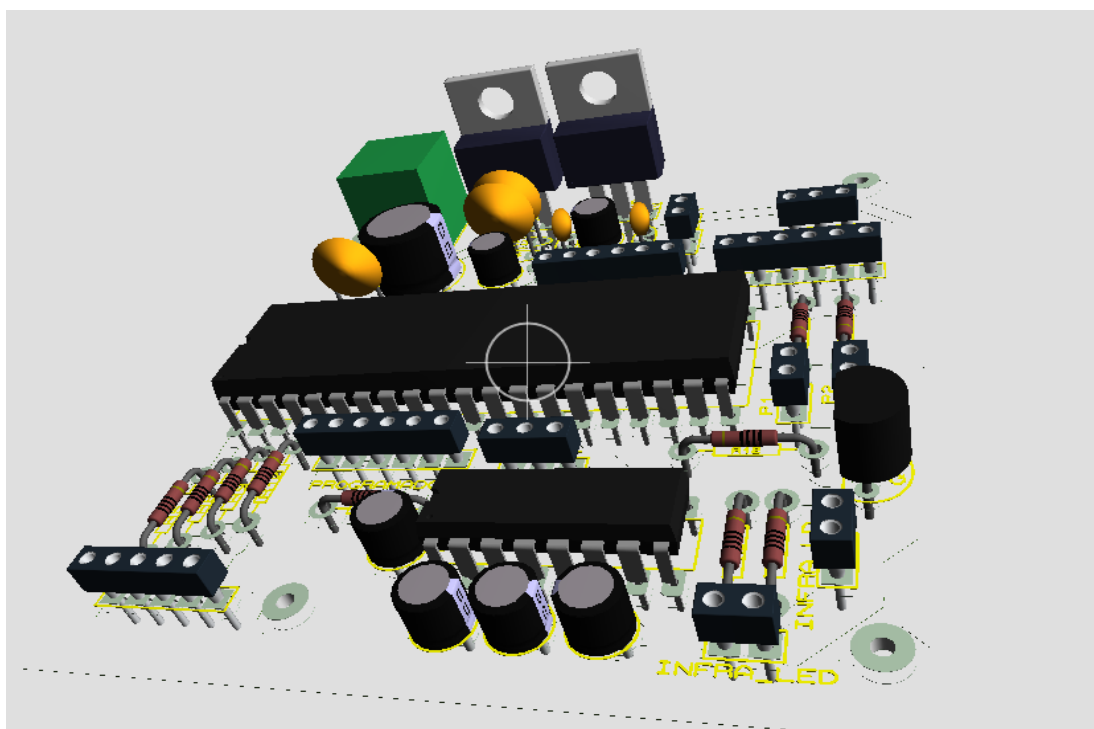


Ilustración 3. 32 Vista 3D de tarjeta electrónica de control.
Fuente: (Autor, 2015)

3.5 Fase IV: Desarrollo de interface gráfica en Labview.

3.5.1 Metodología de desarrollo de software.

Para el desarrollo del software se aplica una metodología tipo cascada en donde las actividades son vistas hacia abajo, es decir secuenciales a través de las fases de análisis de lo que se requiere cumplir, el diseño, implementación en los dispositivos programables y pruebas de los algoritmos elaborados.

Análisis.- Se determina implementar un software de control.

Diseño.- Se diseña la solución del software

Implementación.- Programación en labview

Pruebas.- Se realizan pruebas en sitio con el robot para verificar el funcionamiento del mismo según lo que se solicita en los requerimientos.

Se emplea la herramienta gráfica Labview para el desarrollo de la interface entre el usuario y el robot. Esta interface cuenta con una comunicación Serial RS232, uso de dispositivos multimedia para la reproducción de archivos mp3.

El proyecto consta de un archivo principal denominado REPRODUCCIÓN MÚSICA.

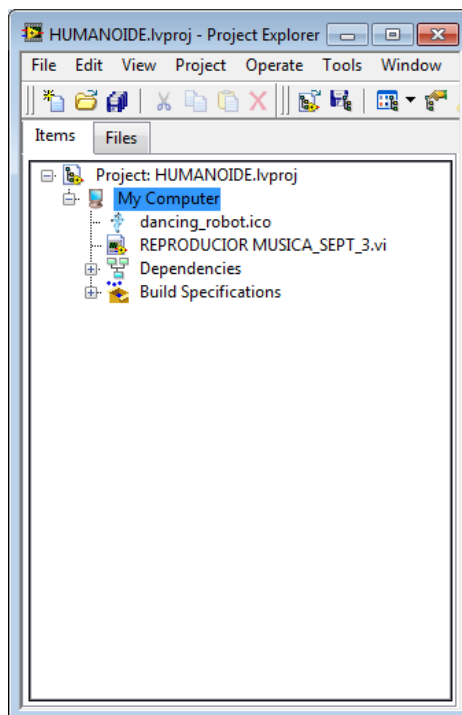


Ilustración 3. 33 Software de control realizado en LABVIEW.

Fuente: (Autor, 2015)

Este archivo consta de 3 ventanas, las cuales permiten al usuario escoger las canciones, escoger los puertos de comunicacion , entre otros.

La ventana de inicio es de bienvenida que contiene información acerca del sistema asi como un introductorio de quien desarrolló el sistema.

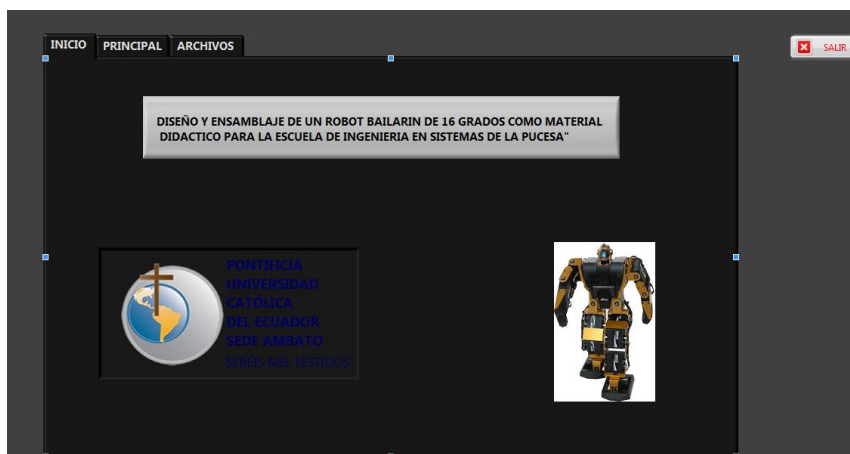


Ilustración 3. 34 Ventana de Inicio.

Fuente: (Autor, 2015)

La segunda ventana contiene cuatro botones de control que son : BACHATA, EJERCICIOS, POP y REGUETÓN, estos cuatro botones permiten al usuario escoger la canción para ser enviada hacia el robot.

A medida que se escoje la canción se envia una orden via RS 232 al módulo de control.

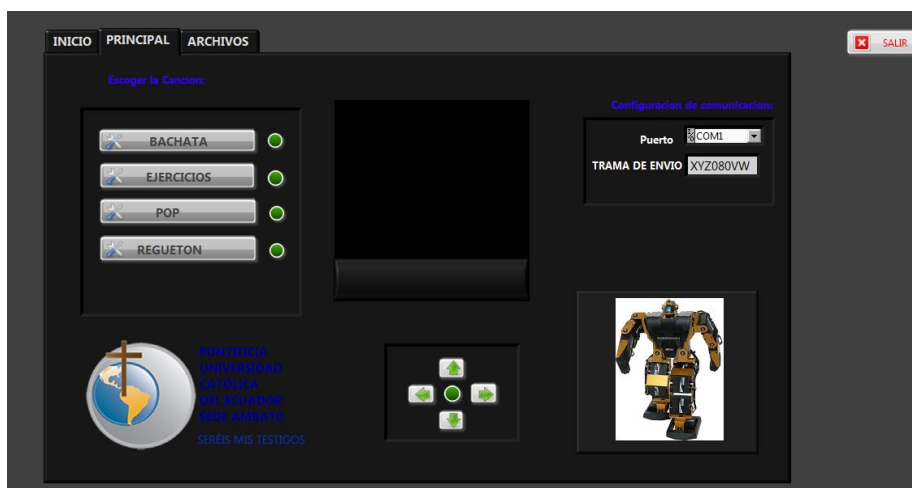


Ilustración 3. 35 Ventana Principal.

Fuente: (Autor, 2015)

De igual forma hay 4 botones que son para el control del movimiento del robot. Todos estos datos son enviados via RS-232 al módulo desarrollado. Este módulo contiene una tarjeta electrónica que recibe esta señal la interpreta y genera los comandos infrarrojos.

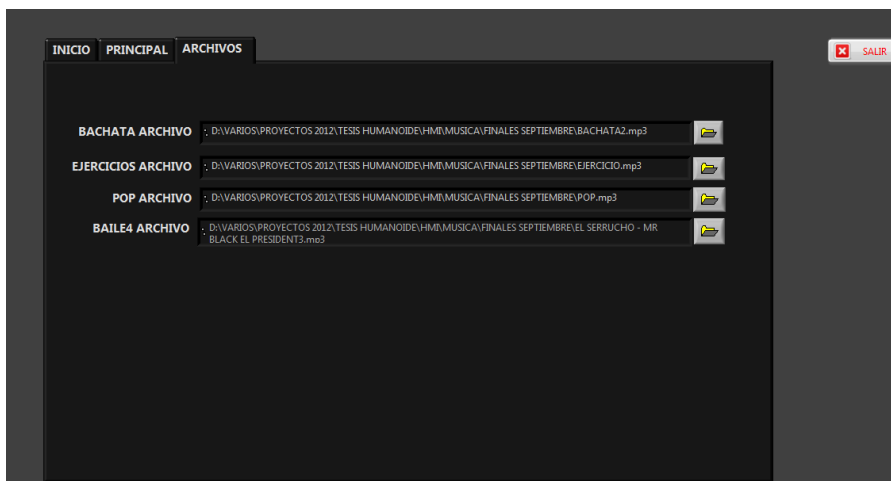


Ilustración 3. 36 Ventana Archivos.

Fuente: (Autor, 2015)

La tercera ventana contiene las direcciones para la lectura de la canciones, esto permite al usuario escoger o redireccionar las canciones cuando se instala la aplicación en un nuevo computador.

Para el desarrollo de la inteface se empleó algunos bloques de programación como los detallados a continuación:

Comunicación RS232.- Esta comunicación contiene bloques que permiten la configuración, escritura y cerrar puertos.



Configuración e inicalizacion de comunicación.



Bloque de escritura RS 232.



Cerrar el puerto.

Para poder comunicarse con el módulo exterior es necesario que se configure el bloque de inicalización, por lo tanto se lo configura de la siguiente forma. (AREVALO, P. 2009)

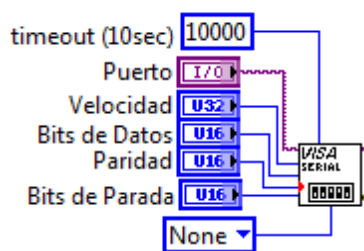


Ilustración 3. 37 Inicialización Comunicación RS232.

Fuente: (Autor, 2015)

Una vez configurado la inicialización es necesario que se cree una trama de datos, la cual es enviada para que el módulo exterior la interprete y tome la acción necesaria.

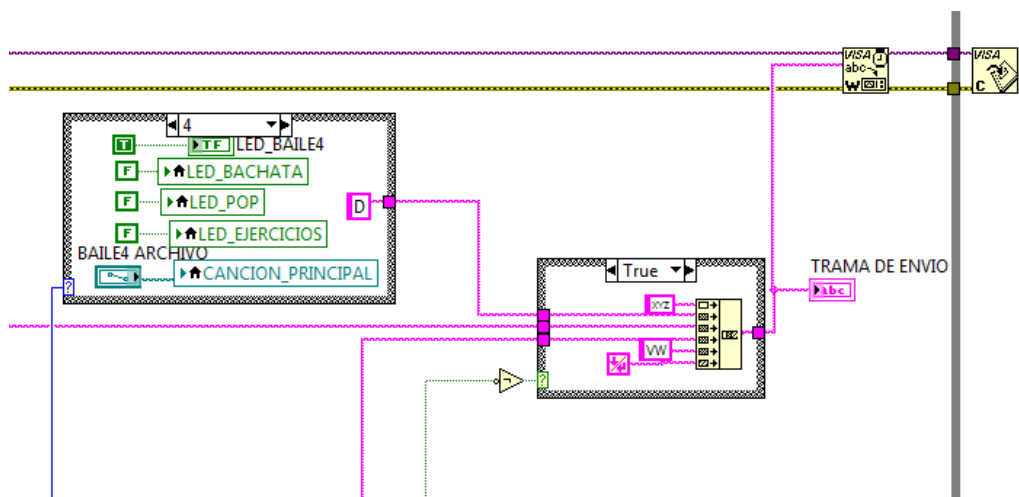


Ilustración 3. 38 Creación de Trama de datos.

Fuente: (Autor, 2015)

A medida que se elige el ritmo, la interface debe escoger automáticamente la canción y reproducirla por lo cual se emplea el lazo Case Structure, el cual elige a medida que el usuario haya presionado el ritmo.

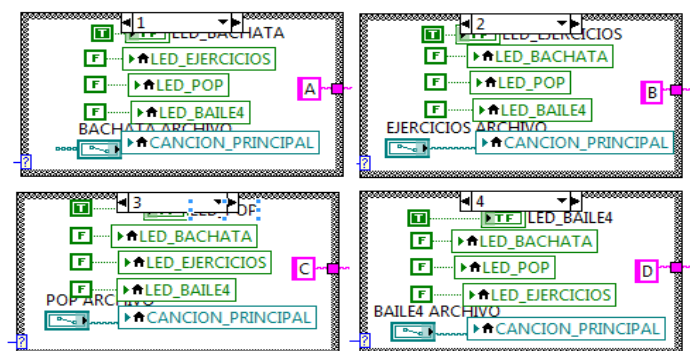


Ilustración 3.39 Selección de ritmo.

Fuente: (Autor, 2015)

Una vez escogida la canción se emplea los bloques de reproducción mp3 que posee Labview.

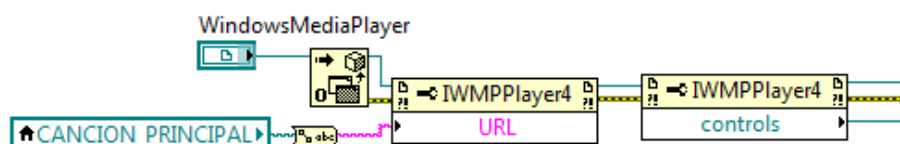


Ilustración 3.38 Selección de canción.

Fuente: (Autor, 2015)

Cada que el usuario presiona el botón del ritmo, se debe hacer un almacenamiento del ritmo escogido, mientras dure la canción escogida, para lo cual se emplea el código siguiente:

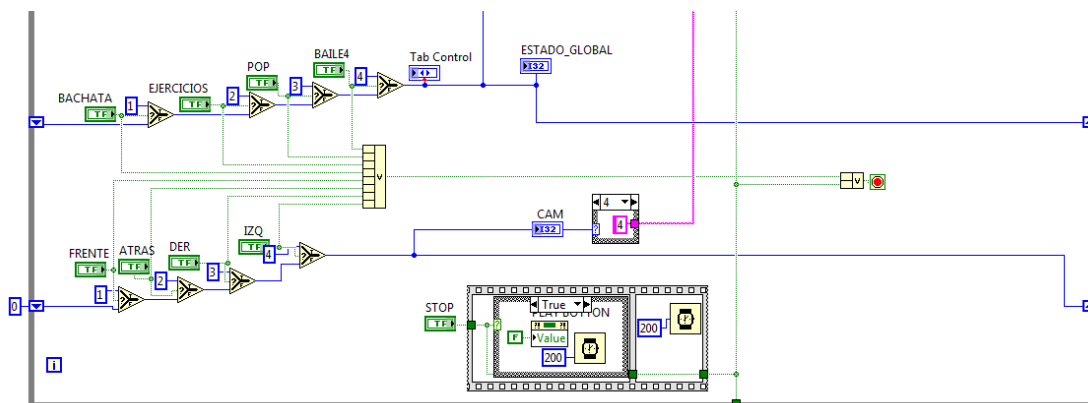


Ilustración 3.40 Almacenamiento de ritmo.

Fuente: (Autor, 2015)

La variable estado Global ayuda al código interno a saber que ritmo se a escogido para poder hacer que a canción dure cierto tiempo y se ejecute el código de PLAY.

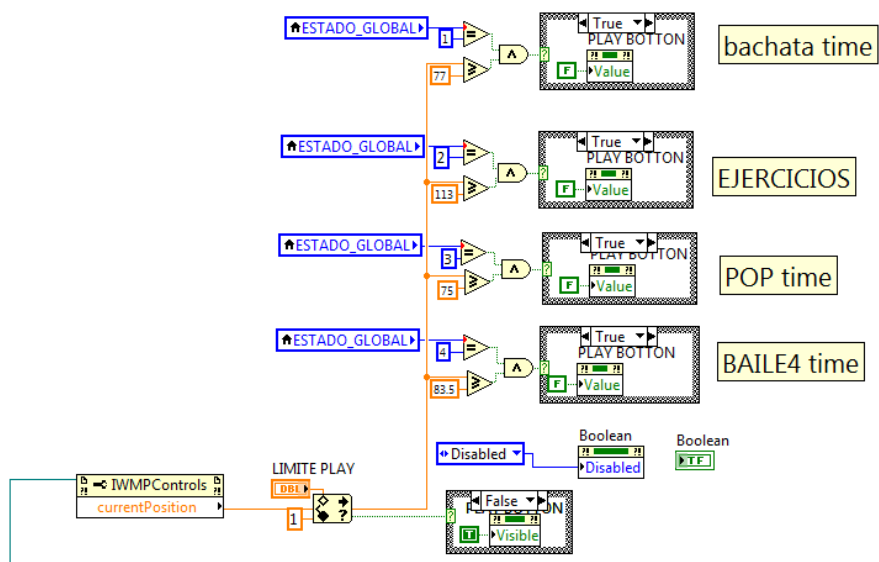


Ilustración 3. 41 Activación de reproducción de ritmo.

Fuente: (Autor, 2015)

3.6 Fase V: Captura del tren de pulsos.

La cadena de pulsos se divide en dos partes principales que son los bits de inicio y bits de comandos.

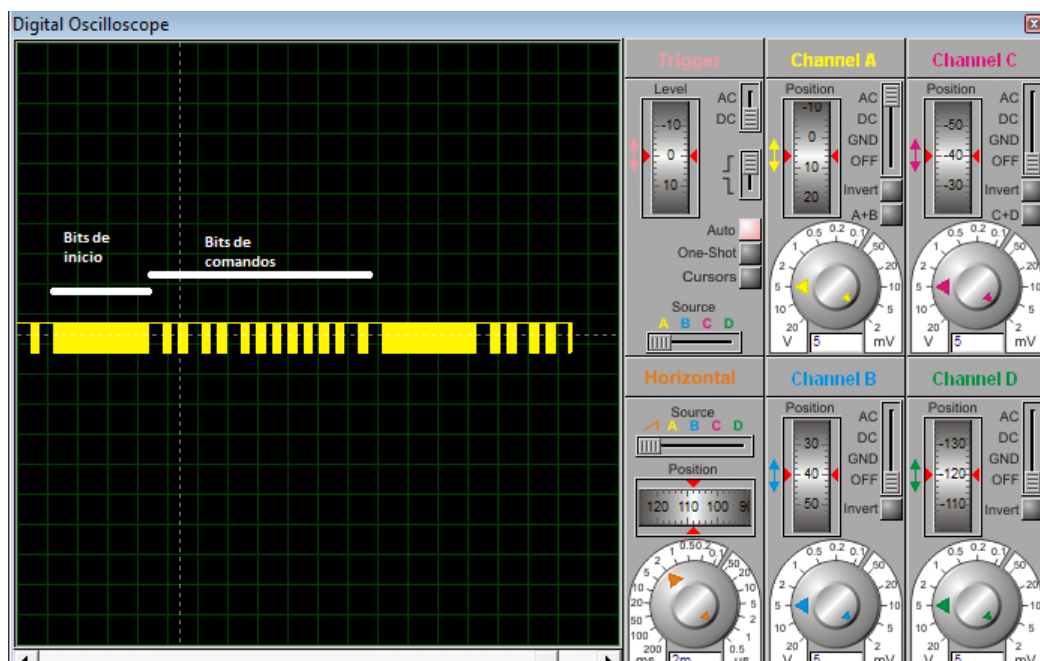


Ilustración 3. 42 Ventana de visualización de pulsos generados usando osciloscopio.
Fuente: (Autor, 2015)

BITS de inicios.-

Estos bits de inicios están conformados por 188 pares de pulsos en alto y bajo.

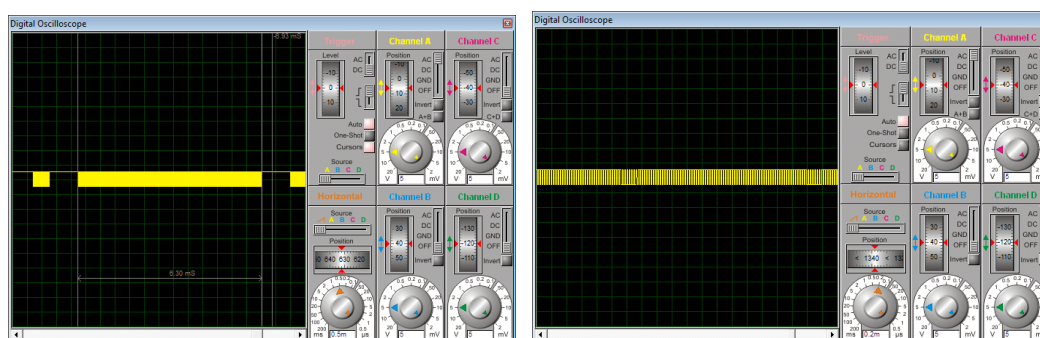


Ilustración 3. 43 Ventana de visualización de pulsos de inicio.
Fuente: (Autor, 2015)

Cada par de pulsos es de 9us en bajo y 13 us en alto, que al ser generados por el microcontrolador se observa una cadena de bits enviados al led infrarrojo.

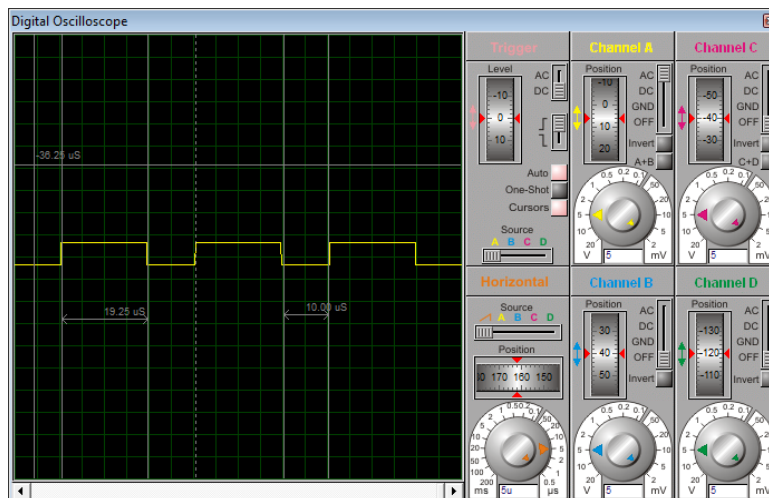


Ilustración 3. 44 Pulsos generados por el Microcontrolador.
Fuente: (Autor, 2015)

BITS de comandos.-

Una vez generado el bit de inicio se genera a continuación los bits de comandos, para simular cada una de las teclas del control remoto Hitec.

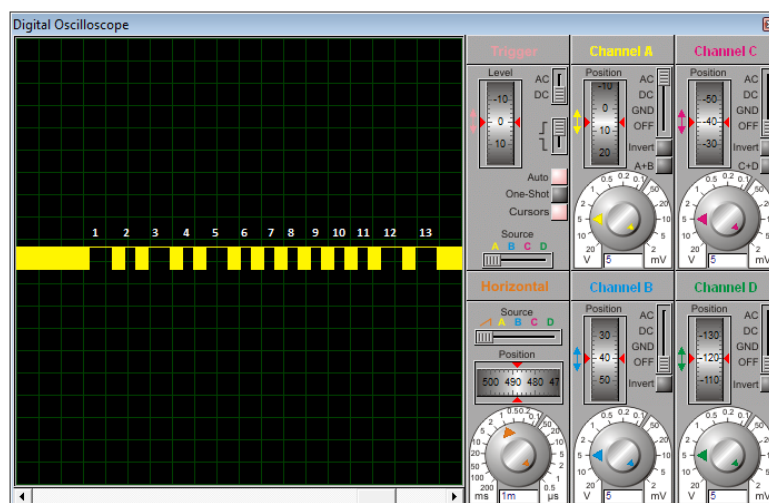


Ilustración 3. 45 Pulsos de comandos generados por el Microcontrolador.
Fuente: (Autor, 2015)

Este tren de pulsos se compone de 13 grupos de pulsos, y cada grupo de pulsos se compone de 18 trenes con un ancho de pulso igual al de bits de inicio (9us en bajo y 13 us en alto)

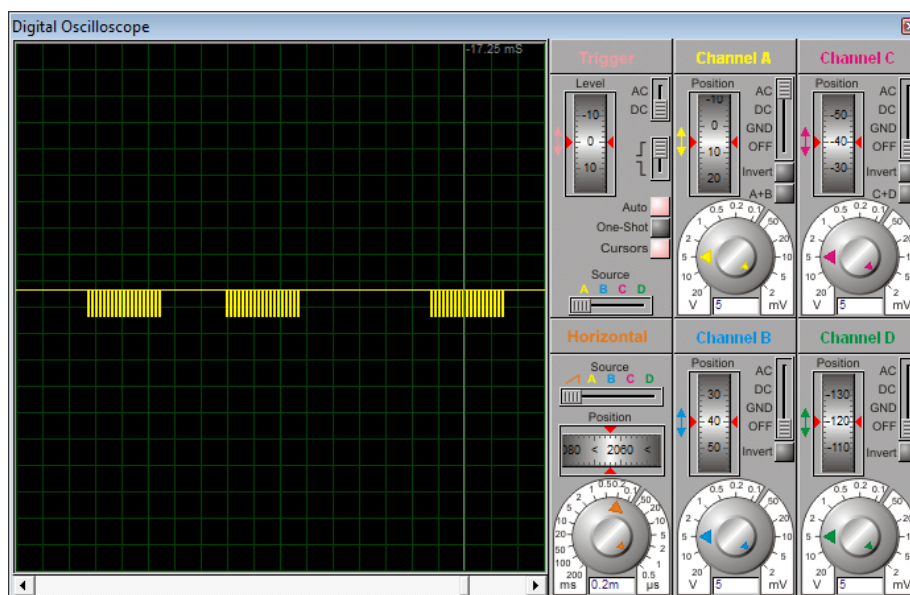


Ilustración 3. 46 Conjunto total de pulsos generados por el Microcontrolador.
Fuente: (Autor, 2015)

3.7 Fase VI: Código de programación de microcontrolador.

Uno de los softwares de compilación para generar archivos .hex o generar lógicas de programación de microcontroladores AVR es la herramienta Bascom Avr, que permite al desarrollador configurar al sistema, obtener datos, etc

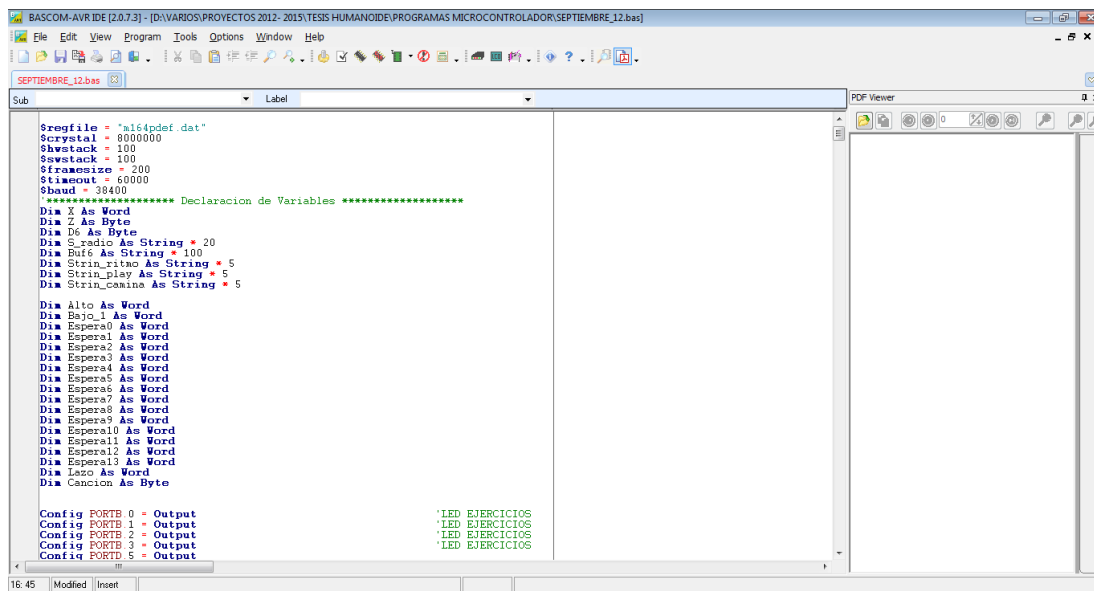


Ilustración 3. 47 Ventana de Programación del microcontrolador usando BASCOM AVR.

Fuente: (Autor, 2015)

Lo principal en el programa es configurar que chip se está empleando que para el caso del sistema es el atmega 164P.

```

$regfile = "m164pdef.dat"
$crystal = 8000000

```

Ilustración 5 Selección del microcontrolador usando BASCOM AVR.

Fuente: (Autor, 2015)

Otro aspecto fundamental es la declaración de variables que se empleó en el programa principal, subrutinas y generación de códigos.

```

***** Declaracion de Variables *****
Dim X As Word
Dim Z As Byte
Dim D6 As Byte
Dim S_radio As String * 20
Dim Buf6 As String * 100
Dim Strin_ritmo As String * 5
Dim Strin_play As String * 5
Dim Strin_camina As String * 5

```

Ilustración 3. 48 Declaración de variables usando BASCOM AVR.

Fuente: (Autor, 2015)

Para poder interpretar una entrada o salida digital que sensa el Microcontrolador, es necesario configurar de acuerdo al uso que se le da.

```
Config PORTB.0 = Output      'LED EJERCICIOS
Config PORTB.1 = Output      'LED EJERCICIOS
Config PORTB.2 = Output      'LED EJERCICIOS
Config PORTB.3 = Output      'LED EJERCICIOS
Config PORTD.5 = Output
Config PORTD.2 = Input       ' PULSADOR 1
Config PORTD.3 = Input       ' PULSADO 2
```

Ilustración 3. 49 Configuración de puertos usando BASCOM AVR.

Fuente: (Autor, 2015)

A cada puerto se le puede colocar un alias para su fácil reconocimiento en toda la lógica del programa, ya sea una entrada o salida.

```
Led_ejercicios Alias PORTB.0
Led_bachata Alias PORTB.1
Led_pop Alias PORTB.2
Led_musica3 Alias PORTB.3
```

Ilustración 3. 50 Asignación de puertos usando BASCOM AVR.

Fuente: (Autor, 2015)

Para establecer la comunicación entre el computador y el sistema se debe configurar la velocidad, paridad, datos bits, y por medio de que puerto se está comunicando.

```
Config Com1 = 19200 , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Open "COM1:" For Binary As #1
Enable Interrupts
On Urxc Serialingreso
Enable Urxc
```

Ilustración 3. 51 Configuración de comunicación serial usando BASCOM AVR.

Fuente: (Autor, 2015)

Al trabajar con interrupción en la comunicación serial, se debe crear una subrutina de ingreso de datos seriales.

```

Serialingreso:
  D6 = Inkey(#1)
  If D6 = 13 Then
    S_radio = Buf6
    Buf6 = ""
  Else
    Buf6 = Buf6 + Chr(d6)
  End If
Return

```

Ilustración 3. 52 Interrupción de comunicación serial usando BASCOM AVR.

Fuente: (Autor, 2015)

Como proceso de visualización de datos en la LCD se emplean las siguientes líneas de instrucción, que permiten al LCD recibir e imprimir en su pantalla los datos necesarios.

```

'***** configuracion LCD *****
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = PORTC.2 , Db5 = PORTC.3 , Db6 = PORTC.4 , Db7 = PORTC.5 , E = PORTC.1 , Rs = PORTC.0
Cursor Off
Cls
Initlcd

```

Ilustración 6 Configuración del LCD usando BASCOM AVR.

Fuente: (Autor, 2015)

Al tener los datos en el sistema Microcontrolado es necesario compararlos y determinar qué tipo de canción es la que se está reproduciendo, para poder generar los códigos infrarrojos.

```

'***** RECEPTE DATOS DE PC *****
For X = 0 To 3
  If Mid(s_radio , X , 3) = "XYZ" Then
    Waitms 30
    Z = X + 3
    Strin_ritmo = Mid(s_radio , Z , 1)
    Z = X + 4
    Strin_play = Mid(s_radio , Z , 1)
    Z = X + 5
    Strin_camina = Mid(s_radio , Z , 1)
  End If
Next X

```

Ilustración 3. 53 Determinación de ritmo usando BASCOM AVR.

Fuente: (Autor, 2015)

Además de comparar que tipo de canción se ha escogido, también es necesario saber si se está reproduciendo la canción

```

If Strin_play = "9" Then
  If Strin_ritmo = "A" Then
    Cancion = 1
    Locate 2 , 1
    Lcd " BACHATA "
    Set Led_ejercicios : Reset Led_bachata : Reset Led_pop : Reset Led_musica3
  End If
  If Strin_ritmo = "B" Then
    Cancion = 2
    Locate 2 , 1
    Lcd " EJERCICIOS "
    Reset Led_ejercicios : Set Led_bachata : Reset Led_pop : Reset Led_musica3
  End If
  If Strin_ritmo = "C" Then
    Cancion = 3
    Locate 2 , 1
    Lcd " POP "
    Reset Led_ejercicios : Reset Led_bachata : Set Led_pop : Reset Led_musica3
  End If
  If Strin_ritmo = "D" Then
    Cancion = 4
    Locate 2 , 1
    Lcd " BAILE 4 "
    Reset Led_ejercicios : Reset Led_bachata : Reset Led_pop : Set Led_musica3
  End If
Else

```

Ilustración 3. 54 Selección de ritmo usando BASCOM AVR.

Fuente: (Autor, 2015)

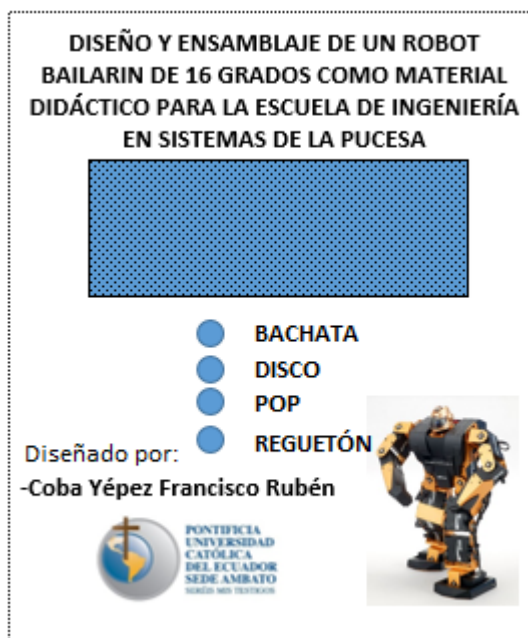


Ilustración 3. 55 Portada del módulo de control infrarrojo.

Fuente: (Autor, 2015)

CAPÍTULO IV

Discusión, análisis y validación de los resultados

4.1 Funcionamiento del Robot, módulo infrarrojo e interface de control de la PC.

El funcionamiento del robot determinará cuan eficiente fue la programación de su tarjeta de control así como del diseño y construcción del módulo infrarrojo que emite las señales de lo seleccionado en la interface de control gráfica. Por ende, el sistema construido consta de:

- Robot Robonova I
- Control remoto
- Módulo infrarrojo construido
- Cable serial, cable serial-usb
- PC con interface de control gráfica.



Ilustración 4. 1 Hardware armado y construido. Robot-Modulo Infrarrojo-Control Remoto.

Fuente: (Autor, 2015)

El funcionamiento del sistema se realiza de la siguiente manera:

Para empezar, lo primero que se debe hacer es conectar la batería y encender el interruptor rojo ubicado en la espalda del robot, con lo cual el robot adoptará la posición inicial de pie.

Se procede a energizar el módulo infrarrojo conectando su enchufe a la red eléctrica de 110V. Los leds infrarrojos deberán estar direccionados apuntando hacia el robot. Se procederá a conectar el cable usb-serial en un puerto de la PC y posterior a esto abrir la interface de control. El control remoto funciona de manera directa con el robot por lo cual es un dispositivo complementario que podrá ser programado para realizar otros movimientos requeridos.

La interface humano máquina creada es intuitiva y consta de la nomenclatura necesaria para poder ser usado por el usuario.

Para ponerla en funcionamiento basta con abrir el directorio del proyecto y ejecutar el comando RUN (Flecha blanca que apunta a la derecha).

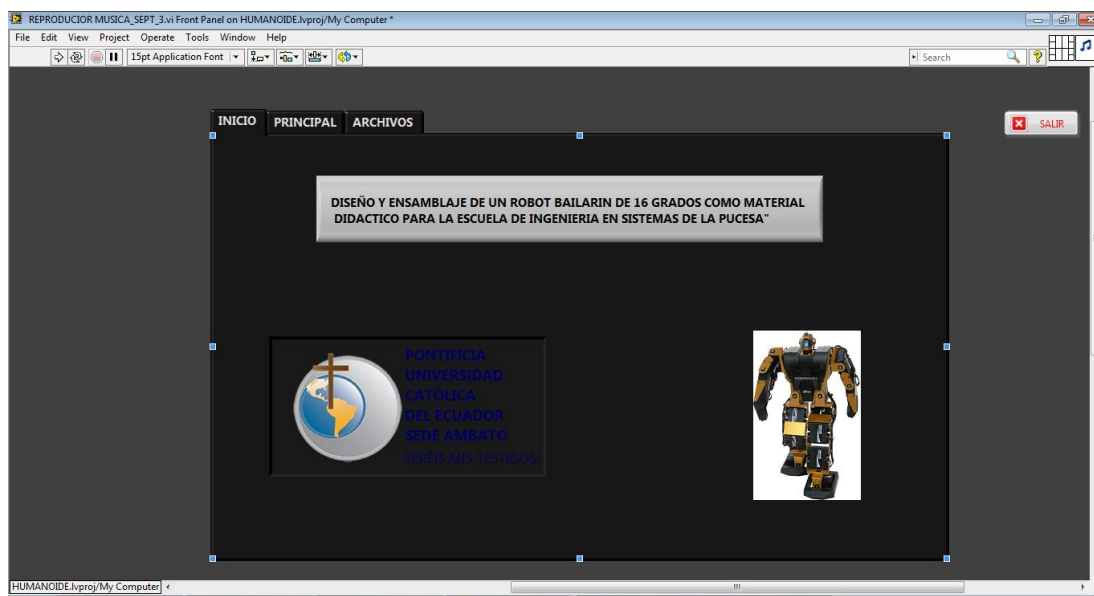


Ilustración 4. 2 Interface de control del robot.
Fuente: (Autor, 2015)

Se procederá a escoger el ritmo de baile y posterior a esto ejecutarlo. El robot recibe el comando y comienza a realizar la rutina de baile siguiendo el ritmo de la canción. Una vez finalizada la rutina, se podrá escoger nuevamente otro ritmo de baile y repetir el proceso.

Cabe recalcar nuevamente que el robot obedece tanto al módulo infrarrojo construido así como a su control remoto original.

4.2 Análisis de Resultados:

Los resultados obtenidos una vez completada la fase de diseño, programación de la tarjeta electrónica adicional como del programa final enviado a la memoria del robot.

Los resultados obtenidos fueron:

- Robot con 4 ritmos de baile distintos.
- Comunicación infrarroja desde la tarjeta diseñada y construida hacia el robot Robonova I.
- Comunicación serial entre la tarjeta diseñada y la PC que contiene la interface gráfica de control del robot.
- Las pruebas realizadas fueron la verificación del robot ante 4 distintos ritmos de baile, los cuales son: bachata, disco, pop y reguetón. Cada uno de estos ritmos tiene una duración aproximada de 80 segundos en los cuales se verifica cada uno de los movimientos programados, así como también la interacción con la PC del usuario.

- Los resultados fueron los deseados, cumpliendo con el objetivo de la tesis, la cual tiene la libertad en un futuro ser modificada para la elaboración de nuevos movimientos y ritmos de baile.
- El código de interface de la PC también puede ser editado en el caso de requerir nuevos comandos que permitan tratar nuevos movimientos y ritmos.

4.3 Validación de Resultados:

El objetivo principal así como los objetivos secundarios fueron validados por parte del Sr. Mg. Acurio Maldonado Santiago Alejandro al realizar una demostración práctica del funcionamiento del robot cumpliendo sin dificultades cada uno de los ritmos de baile planteados. Se procedió también a la verificación de la comunicación infrarroja que emite el módulo construido así como también la visualización del código fuente cargado al microcontrolador del robot, de la placa diseñada y construida así como también el código de programación gráfico de la interface de control.

Esta demostración se la realizó en una de las aulas de la Pontificia Universidad Católica del Ecuador Sede Ambato en el mes de Diciembre del año 2014.



Ilustración 4. 3 Robot Bailando.
Fuente: (Autor, 2015)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El adecuado ensamblaje del robonova I teniendo en cuenta la correcta distribución del peso de sus componentes, fue crucial al momento de encontrar los puntos de equilibrio del robot en cada uno de los pasos de baile, logrando de esta manera estabilidad en cada uno de los 16 grados de libertad para el control del robot.
- La recopilación de información y aplicación de las diferentes herramientas de software, permitió desarrollar cada una de las etapas de programación del robot, microcontrolador del módulo infrarrojo, simulación y control mediante software dejando un excelente punto de partida para la creación de nuevas aplicaciones con el robot.
- Emular el movimiento de una persona que baila es una tarea compleja, pero al realizar el presente trabajo, se encontró una estrategia para obtener posiciones controladas del movimiento de los servomotores para llegar a tener una semejanza de la movilidad entre el robot y el ser humano.

- Un sistema robótico humanoide con más grados de libertad permitiría obtener una mayor similitud entre robot y humano ya que los grados de libertad son los que permitieron realizar movimientos de mayor coordinación.
- Por medio del uso de las herramientas electrónicas como software y hardware, se implementó el control en base a sistemas microprocesados y comunicación inalámbrica, mejorando el tiempo de respuesta y control del robot, además de la movilidad de sus extremidades.
- Uno de los ambientes de desarrollo para la interface gráfica fue la plataforma Labview, la cual permite al usuario realizar modificaciones o cambios con facilidad, debido a su disponibilidad de herramientas de programación.

5.2 RECOMENDACIONES

- Robonova I al conservar la tarjeta electrónica original del fabricante debe mantener el uso del software de programación roboBasic, para mejorar u optimizar la movilidad del mismo.
- Se recomienda continuar con el desarrollo de movimientos de mayor complejidad hasta alcanzar una mayor experticia en la programación del robot Robonova I. Además se recomienda ubicar los servomotores del robot un una posición inicial adecuada con el fin de evitar movimientos bruscos que puedan provocar colisiones entre las articulaciones durante el momento de energizar.
- La comunicación de tipo infrarrojo maneja un alcance corto, por lo cual se recomendaría optar por otro tipo de comunicación como Wireless o Zigbee para alcanzar una mayor distancia.
- Labview es una plataforma de desarrollo que contiene una gran gama de algoritmos de control tanto en posicionamiento y movilidad de plataformas robóticas, por tal razón se recomienda utilizar a futuro dichos sistemas de control para robots humanoides.

BIBLIOGRAFIA

Alterovitz, R. (2008). *Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-Guided Procedures*, Springer Tracts in Advanced Robotics. Springer.

Arevalo, P. (2009). *LABVIEW BÁSICO*. QUITO.

ATMEL. (2015). *ATMEL CORPORATION*. Obtenido de http://www.atmel.com/Images/Atmel-8011-8-bit-AVR-Microcontroller-ATmega164P-324P-644P_datasheet.pdf

Autor. (ABRIL de 2015). AMBATO, ECUADOR.

Barrientos A, P. L. (2008). *Fundamentos de la Robótica*. McGraw-Hill.

Bonell, M. (2011). *DISEÑO Y CONSTRUCCION DE UN ROBOT HUMANOIDE*. BARCELONA.

Comité Español de Automática. (2011). *El Libro Blanco de la Robótica en España Investigación, tecnologías y formación*. CEA - GTRob con subvención del MEC.

Corke, P. (2011). *Robotics, Vision and Control*. Springer.

Cowan, N. J. (2011). "Robotic Needle Steering: Design, Modeling, Planning, and Image Guidance," in *Surgical Robotics: System Applications and Visions*. Springer.

Craig, J. (2006). *Introducción a la Robotica, Mecanismos y Control*. DA.

Dongguan Fosen Electronic Technology Co., Ltd. Norte America (2015). Recuperado de <http://spanish.alibaba.com/product-gs/infrared-remote-control-receiver-module-dqir-38343e8-511243805.html>

Electrónica Steren S.A. de C.V con domicilio en Calzada Camarones No. 112, Col. Obrero Popular, Deleg. Azcapotzalco, México D.F.,C.P.

02840 (2015) Recuperado de

<http://www.steren.com.mx/catalogo/prod.asp?p=2059>.

Falomir, Z. (2006). *Robots Humanoides*. Castellon.

Ferrier, J.-L. (2011). *Informatics in Control, Automation and Robotics*. Springer.

H. Choset, K. M. (2005). *Principles of robot motion: theory, algorithms and implementations*. MIT Press.

Jesus Aramburo, A. R. (2008). *Advances in Robotics, Automation and Control*. InTech.

Kurfess, T. (s.f.). *Robotics and Automation Handbook*.

Labcenter, E. (2015). Obtenido de <http://www.labcenter.com/index.cfm>

López, G. (2003). *Comunicación*. Proyecto de Grado, Universidad de la República Oriental de Uruguay.

Martínez-Duart, J. M. (1988). *Optoelectrónica y comunicación óptica*. Madrid: RAYCAR S.A.

Ollero, A. (2001). *Robótica, Manipuladores y Robots Móviles*. Marcombo Boixareu.

Reed, K. (2011). Robot-Assisted Needle Steering. *IEEE Robotics and Automation Magazine*, 35-46.

ROBOTICS, (2005). *ROBONOVA-I English User Manual*.

Romero, M. (2012). *Robótica: Entra al mundo de la Inteligencia Artificial*. Buenos Aires: Educ.ar S.E.

Secchi, H. (2008). *Introducción a los Robots Móviles*. San Juan.

Tomasi, W. (2003). *Sistemas de comunicaciones electrónicas*. Mexico: Pearson Educación.

Universidad Politécnica de Valencia, Redes De Área Local Inalámbricas

(2015). Recuperado de

http://www.upv.es/antenas/Tema_1/espectro_electromagnetico.htm

Vázquez, S. G. (2015). *Elementos de sistemas de telecomunicaciones*.

Madrid: Paraninfo.

ANEXOS

ANEXO 1: COMANDOS USADOS PARA DECLARACIONES Y DEFINICIONES.

DIM Declara una variable

AS Asigna una variable durante su declaración

CONST Declara una constante

BYTE Se define una variable como tipo byte en su declaración

INTEGER Se define una variable como tipo integer (entero) en su declaración

Comandos para el control de flujo

IF Comienzo de una sentencia condicional

THEN Se ejecuta la siguiente sentencia (o sentencias) cuando la condición es cierta.

ELSE Se ejecuta la siguiente sentencia (o sentencias) cuando la condición es falsa.

ELSEIF Comienzo de otra sentencia condicional.

ENDIF Fin de una sentencia condicional

FOR Comienza un bucle.

TO Asigna el rango de iteraciones de un bucle. **NEXT** Sentencia de cierre de ejecución de bucle.

GOTO Divide el flujo del programa.

GOSUB Invoca a una sub-rutina

RETURN Sale (retorna) de una sub-rutina

END Finaliza la ejecución del programa. **STOP** Detiene la ejecución del programa.

RUN Ejecuta un programa (continua).

WAIT Espera hasta que se complete el programa.

DELAY Hace una pausa en el programa durante un tiempo.

BREAK Detiene la ejecución del programa y pasa a modo de depuración.

Comandos de entrada y salida de señales digitales

IN Lee las señales de un puerto de entrada.

OUT Envía una señal a un puerto de salida

BYTEIN Lee una señal byte de un puerto de entrada

BYTEOUT Envía una señal byte a un puerto de salida.

INKEY Tecla (clave) proveniente de un puerto de entrada.

STATE Estado del puerto de salida

PULSE Envía una señal (de pulso) al puerto de salida.

TOGGLE Invierte el estado del puerto de salida.

KEYIN Toma la entrada del teclado analógico (de entrada).

Comandos de memoria

PEEK Lee datos del controlador de la RAM.

POKE Escribe datos en el controlador de la RAM.

ROMPEEK Lee datos del controlador externo EEPROM RAM.

ROMPOKE Escribe datos en el controlador EEPROM RAM.

Comandos para el LCD

LCDINIT Inicializa el módulo LCD.

CLS Borra todos los caracteres del módulo LCD.

LOCATE Define la posición de caracteres en el módulo LCD.

PRINT Muestra caracteres en el módulo LCD.

FORMAT Define el formato a mostrar en el módulo LCD

CSON Hace que aparezca el cursor en el módulo LCD.

CSOFF Oculta el cursor del módulo LCD.

CONT Define el contraste de los caracteres en el módulo LCD.

DEC Envía un valor decimal al módulo LCD.

HEX Envía un valor hexadecimal al módulo LCD.

BIN Envía un valor binario al módulo LCD

Operadores lógicos

AND Usa la operación lógica AND.

OR Usa la operación lógica OR.

MOD Calcula el módulo de una operación aritmética.

XOR Usa la operación lógica XOR.

NOT Invierte todos los bits

Comandos para el control del motor

ZERO Define el punto 0 (neutro) del servo.

MOTOR Activa el puerto de salida de un servo.

MOTOROFF Desactiva el puerto de salida de un servo.

MOVE Maneja varios servos al mismo tiempo.

SPEED Define la velocidad del servo.

ACCEL Define la aceleración del servo.

DIR Define la dirección (sentido) del motor del servo.

PTP Activa/desactiva varias operaciones del control del motor.

SERVO Controla el servo.

PWM Define la amplitud del pulso de un motor DC.

FASTSERVO Maneja el servo a la velocidad máxima.

HIGHSPEED Activa/desactiva la velocidad máxima del servo.

MOVEPOS Mueve el grupo de servos definidos con POS.

POS Selecciona una posición específica del robot.

FPWM Cambia la amplitud y la frecuencia del pulso

MOVE24 Maneja los 24 servos a la vez.

INIT Define la posición inicial

MOTORIN Lee la posición actual del servo (valor).

SETON Configuración para usar la función "setup"

SETOFF Cancela la función "set up".

ALLON Función de configuración para todos los servos.

ALLOFF Cancela la función de configuración de todos los servos.

GETMOTORSET Lee los valores de un servo y mantiene la posición.

Parámetros para asignación de grupos de motores

G6A Asigna los servos #0~#5 al grupo A.

G6B Asigna los servos #6~#11 al grupo B.

G6C Asigna los servos #12~#17 al grupo C.

G6D Asigna los servos #18~#23 al grupo D.

G6E Asigna los servos #24~#29 al grupo E.

G8B Asigna los servos #8~#15 al grupo B.

G8C Asigna los servos #16~#23 al grupo C.

G8D Asigna los servos #24~#31 al grupo D.

G12 Asigna los servos #0~#11.

G16 Asigna los servos #0~#15.

G24 Asigna los servos #0~#23.

G32 Asigna los servos #0~#31.

Procesado de comandos

ON...GOTO Salto dependiendo del valor de una variable.

Comandos de intención

\$DEVICE Configura el controlador para ser manejado por el programa en ejecución.

\$LIMIT Limita el recorrido de un servo.

ANEXO 2: PROGRAMA DE RUTINAS DE BAILE CREADO EN RoboBASIC v2.5.

```

*****
*****          ***
*****          ***
*****          ***
*****

'== Declaración de variables =====
DIM A AS BYTE
DIM n AS BYTE
DIM i AS BYTE
DIM j AS BYTE
PTP SETON
PTP ALLON

'== Configuración de dirección de motores =====
DIR G6A,1,0,0,1,0,0
DIR G6B,1,1,1,1,1,1
DIR G6C,0,0,0,0,0,0
DIR G6D,0,1,1,0,1,0

'== Posición de arranque de motores =====
GETMOTORSET G6A,1,1,1,1,1,0
GETMOTORSET G6B,1,1,1,0,0,0
GETMOTORSET G6C,1,1,1,0,0,0
GETMOTORSET G6D,1,1,1,1,1,0

        SPEED 5

'== Endendido de motores =====
        MOTOR G24

        GOSUB standard_pose

'===== Programa Principal=====
MAIN:

A = REMOCON(1)

'==RUTINA BACHATA==

        IF A = 2 THEN

                MOVE G24, 100, 76, 145, 93, 100, , 125, 30, 80, , , , 100, 30, 80, , , , 100, 76, 145,
93, 100,

                FOR n=1 TO 3
                GOSUB right_shift
                GOSUB right_turn
                GOSUB left_shift
                GOSUB left_turn
                NEXT

                FOR n=1 TO 2
                GOSUB right_shift
                GOSUB right_turn
                NEXT

                FOR n=1 TO 3
                GOSUB right_shift
                GOSUB right_turn
                GOSUB left_shift
                GOSUB left_turn
                NEXT

```

```

    FOR n=1 TO 2
    GOSUB right_shift
    GOSUB right_turn
    NEXT

    FOR n=1 TO 2
    GOSUB right_shift
    GOSUB right_turn
    GOSUB left_shift
    GOSUB left_turn
    NEXT

    MOVE G24, 100, 76, 145, 93, 100, , 100, 30, 80, , , , 130, 30, 80, , , , 100, 76, 145,
93, 100,

        GOSUB standard_pose
    DELAY 500
    GOSUB bow_pose
    DELAY 500
    GOSUB standard_pose

    ENDIF

'==RUTINA EJERCICIOS

    IF A = 3 THEN
        GOSUB sentado
        DELAY 3000
        FOR n=1 TO 5
            GOSUB flexiones
        NEXT
        DELAY 500
        GOSUB back_stand_up
        SPEED 5
        GOSUB standard_pose
    DELAY 500
        GOSUB sentado
        DELAY 1000
        GOSUB posicion
        DELAY 1000
        FOR n=1 TO 6
            GOSUB flexiones_una_mano
        NEXT
        GOSUB posicion_1
        DELAY 1000
        GOSUB back_stand_up
        SPEED 5
        GOSUB standard_pose
    DELAY 500
    FOR n=1 TO 2
    GOSUB body_move
        GOSUB standard_pose
    NEXT
    DELAY 500
    GOSUB standard_pose
    DELAY 500
    FOR n=1 TO 3
    GOSUB foot_up
    GOSUB standard_pose
    NEXT
    DELAY 500
    SPEED 8
        GOSUB handstanding
        DELAY 500

```

```

        SPEED 6
        GOSUB standard_pose
    DELAY 500
    GOSUB volando
        GOSUB standard_pose
        DELAY 1000
    GOSUB bow_pose
    GOSUB standard_pose
        DELAY 1000
    ENDIF

'==RUTINA TRILLER==

    IF A = 4 THEN

        MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,

        GOSUB rutina_0

        FOR j=1 TO 5
        GOSUB rutina_1
        NEXT j

        GOSUB brazos

        FOR j=1 TO 3 'brazos en escuadra
        GOSUB rutina_2
        NEXT j

        FOR j=1 TO 2 'brazos de lado a lado
        GOSUB rutina_6
        NEXT j

        FOR j=1 TO 3 'brazos en escuadra
        GOSUB rutina_2
        NEXT j

        FOR j=1 TO 2 'brazos de lado a lado
        GOSUB rutina_6
        NEXT j

        GOSUB rutina_00

        FOR j=1 TO 3
        GOSUB rutina_1
        NEXT j

        GOSUB brazos_1 'brazos arriba

        FOR j=1 TO 3
        GOSUB rutina_3
        NEXT j

        FOR j=1 TO 2
        GOSUB rutina_4
        NEXT j

        GOSUB brazos

        FOR j=1 TO 3
        GOSUB rutina_2
        NEXT j

        GOSUB rutina_00

```

```

GOSUB forward_walk
GOSUB rutina_00

FOR j=1 TO 3
GOSUB rutina_1
NEXT j

MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145, 93,
100,

GOSUB standard_pose
DELAY 500

GOSUB bow_pose
DELAY 500
GOSUB standard_pose

ENDIF

'==RUTINA SERRUCHO=

IF A = 5 THEN
FOR j=1 TO 6
GOSUB rutina_1
NEXT j
GOSUB standard_pose
DELAY 100
GOSUB forward_walk
DELAY 100
GOSUB standard_pose
DELAY 100
SPEED 7
MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,

FOR j=1 TO 5
SPEED 6
MOVE G24, 100, 76, 145, 93, 100, , 160, 10, 50, , , , 175, 85, 15, , , , 100, 76, 145,
93, 100,
DELAY 100
SPEED 6
MOVE G24, 100, 76, 145, 93, 100, , 160, 10, 50, , , , 175, 40, 65, , , , 100, 76, 145,
93, 100,
DELAY 100
NEXT j
DELAY 100
MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,
FOR j=1 TO 5
GOSUB rutina_serrucho
NEXT j
DELAY 100
SPEED 8
MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,

DELAY 100
GOSUB standard_pose
DELAY 100
GOSUB forward_walk
DELAY 100
GOSUB standard_pose
DELAY 100

```

```

MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,
'mov rapido de manos
FOR j=1 TO 6
SPEED 9
MOVE G24, 100, 76, 145, 93, 100, , 180, 15, 50, , , , 185, 15, 50, , , , 100, 76, 145,
93, 100,
DELAY 50
MOVE G24, 100, 76, 145, 93, 100, , 160, 45, 50, , , , 160, 45, 50, , , , 100, 76, 145,
93, 100,
DELAY 50
NEXT j
DELAY 50

FOR j=1 TO 6
SPEED 12
MOVE G24, 100, 76, 145, 93, 100, , 180, 15, 50, , , , 185, 15, 50, , , , 100, 76, 145,
93, 100,
DELAY 30
MOVE G24, 100, 76, 145, 93, 100, , 160, 45, 50, , , , 160, 45, 50, , , , 100, 76, 145,
93, 100,
DELAY 30
NEXT j
DELAY 50

FOR j=1 TO 6
SPEED 15
MOVE G24, 100, 76, 145, 93, 100, , 180, 15, 50, , , , 185, 15, 50, , , , 100, 76, 145,
93, 100,
DELAY 10
MOVE G24, 100, 76, 145, 93, 100, , 160, 45, 50, , , , 160, 45, 50, , , , 100, 76, 145,
93, 100,
DELAY 10
NEXT j
DELAY 50

MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,
GOSUB standard_pose

FOR j=1 TO 6
GOSUB rutina_1
NEXT j
GOSUB standard_pose
DELAY 100
GOSUB forward_walk
DELAY 100
GOSUB standard_pose
DELAY 100
SPEED 7
MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,

FOR j=1 TO 5
SPEED 6
MOVE G24, 100, 76, 145, 93, 100, , 160, 10, 50, , , , 175, 85, 15, , , , 100, 76, 145,
93, 100,
DELAY 100
SPEED 6
MOVE G24, 100, 76, 145, 93, 100, , 160, 10, 50, , , , 175, 40, 65, , , , 100, 76, 145,
93, 100,
DELAY 100
NEXT j
DELAY 100

```

```

    MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,
    FOR j=1 TO 5
    GOSUB rutina_serrucho
    NEXT j
    DELAY 100
    SPEED 8
    MOVE G24, 100, 76, 145, 93, 100, , 130, 30, 80, , , , 133, 30, 80, , , , 100, 76, 145,
93, 100,

    DELAY 100

    GOSUB standard_pose
    DELAY 500
    GOSUB bow_pose
    DELAY 500
    GOSUB standard_pose

    ENDIF

'==ADELANTE=

    IF A = 11 THEN
    GOSUB forward_walk
    GOSUB standard_pose
    ENDIF

'==ATRAS=
    IF A = 12 THEN
    GOSUB backward_walk
    GOSUB standard_pose
    ENDIF

'==DERECHA=
    IF A=13 THEN
    GOSUB right_turn_1
    GOSUB standard_pose
    ENDIF

'==IZQUIERDA=
    IF A=14 THEN
    GOSUB left_turn_1
    GOSUB standard_pose
    ENDIF

    GOTO MAIN

handstanding:
    GOSUB fall_forward
    GOSUB standard_pose
    GOSUB foot_up2
    GOSUB foot_up2
    GOSUB foot_up2
    GOSUB standard_pose
    GOSUB back_stand_up
RETURN
'=====

fall_forward:
    SPEED 10
    MOVE G6A, 100, 155, 25, 140, 100, 100
    MOVE G6D, 100, 155, 25, 140, 100, 100
    MOVE G6B, 130, 50, 85, 100, 100, 100
    MOVE G6C, 130, 50, 85, 100, 100, 100

```

```

WAIT
MOVE G6A, 60, 165, 25, 160, 145, 100
MOVE G6D, 60, 165, 25, 160, 145, 100
MOVE G6B, 150, 60, 90, 100, 100, 100
MOVE G6C, 150, 60, 90, 100, 100, 100
WAIT
MOVE G6A, 60, 165, 30, 165, 155, 100
MOVE G6D, 60, 165, 30, 165, 155, 100
MOVE G6B, 170, 10, 100, 100, 100, 100
MOVE G6C, 170, 10, 100, 100, 100, 100
WAIT
SPEED 3
MOVE G6A, 75, 165, 55, 165, 155, 100
MOVE G6D, 75, 165, 55, 165, 155, 100
MOVE G6B, 185, 10, 100, 100, 100, 100
MOVE G6C, 185, 10, 100, 100, 100, 100
WAIT
SPEED 10
MOVE G6A, 80, 155, 85, 150, 150, 100
MOVE G6D, 80, 155, 85, 150, 150, 100
MOVE G6B, 185, 40, 60, 100, 100, 100
MOVE G6C, 185, 40, 60, 100, 100, 100
WAIT
MOVE G6A, 100, 130, 120, 80, 110, 100
MOVE G6D, 100, 130, 120, 80, 110, 100
MOVE G6B, 125, 160, 10, 100, 100, 100
MOVE G6C, 125, 160, 10, 100, 100, 100
WAIT
RETURN

```

```

=====

```

foot_up2:

```

SPEED 6
MOVE G6A, 100, 125, 65, 10, 100, ,
MOVE G6D, 100, 125, 65, 10, 100, ,
MOVE G6B, 110, 30, 80, , , ,
MOVE G6C, 110, 30, 80, , , ,
SPEED 3
MOVE G6A, 100, 125, 65, 10, 100, ,
MOVE G6D, 100, 125, 65, 10, 100, ,
MOVE G6B, 170, 30, 80, , , ,
MOVE G6C, 170, 30, 80, , , ,
WAIT
DELAY 200
SPEED 6
MOVE G6A, 100, 89, 129, 57, 100, ,
MOVE G6D, 100, 89, 129, 57, 100, ,
MOVE G6B, 180, 30, 80, , , ,
MOVE G6C, 180, 30, 80, , , ,
WAIT
MOVE G6A, 100, 64, 179, 57, 100, ,
MOVE G6D, 100, 64, 179, 57, 100, ,
MOVE G6B, 190, 50, 80, , , ,
MOVE G6C, 190, 50, 80, , , ,
WAIT
DELAY 500
MOVE G6A, 100, 64, 179, 57, 100, ,
MOVE G6D, 100, 64, 179, 57, 100, ,
MOVE G6B, 190, 50, 80, , , ,
MOVE G6C, 190, 50, 80, , , ,
WAIT
MOVE G6A, 100, 89, 129, 57, 100, ,
MOVE G6D, 100, 89, 129, 57, 100, ,
MOVE G6B, 180, 30, 80, , , ,
MOVE G6C, 180, 30, 80, , , ,

```



```

WAIT
SPEED 3
MOVE G6A, 100, 125, 65, 10, 100, ,
MOVE G6D, 100, 125, 65, 10, 100, ,
MOVE G6B, 170, 30, 80, , , ,
MOVE G6C, 170, 30, 80, , , ,
WAIT
SPEED 6
MOVE G6A, 100, 125, 65, 10, 100, ,
MOVE G6D, 100, 125, 65, 10, 100, ,
MOVE G6B, 110, 30, 80, , , ,
MOVE G6C, 110, 30, 80, , , ,
WAIT
RETURN

```

```
'=====
```

```

back_stand_up:
SPEED 10
MOVE G6A, 100, 130, 120, 80, 110, 100
MOVE G6D, 100, 130, 120, 80, 110, 100
MOVE G6B, 150, 160, 10, 100, 100, 100
MOVE G6C, 150, 160, 10, 100, 100, 100
WAIT
MOVE G6A, 80, 155, 85, 150, 150, 100
MOVE G6D, 80, 155, 85, 150, 150, 100
MOVE G6B, 185, 40, 60, 100, 100, 100
MOVE G6C, 185, 40, 60, 100, 100, 100
WAIT
MOVE G6A, 75, 165, 55, 165, 155, 100
MOVE G6D, 75, 165, 55, 165, 155, 100
MOVE G6B, 185, 10, 100, 100, 100, 100
MOVE G6C, 185, 10, 100, 100, 100, 100
WAIT
MOVE G6A, 60, 165, 30, 165, 155, 100
MOVE G6D, 60, 165, 30, 165, 155, 100
MOVE G6B, 170, 10, 100, 100, 100, 100
MOVE G6C, 170, 10, 100, 100, 100, 100
WAIT
MOVE G6A, 60, 165, 25, 160, 145, 100
MOVE G6D, 60, 165, 25, 160, 145, 100
MOVE G6B, 150, 60, 90, 100, 100, 100
MOVE G6C, 150, 60, 90, 100, 100, 100
WAIT
MOVE G6A, 100, 155, 25, 140, 100, 100
MOVE G6D, 100, 155, 25, 140, 100, 100
MOVE G6B, 130, 50, 85, 100, 100, 100
MOVE G6C, 130, 50, 85, 100, 100, 100
WAIT
RETURN

```

```
'=====
```

```

standard_pose:

MOVE G6A,100, 76, 145, 93, 100, 100
MOVE G6D,100, 76, 145, 93, 100, 100
MOVE G6B,100, 30, 80, 100, 100, 100
MOVE G6C,100, 30, 80, 100, 100, 100
WAIT

RETURN

```

```
'=====
```

```

left_turn:

SPEED 6

```

```

MOVE G6A, 113, 75, 145, 97, 93, 60
MOVE G6D, 90, 50, 157, 115, 112, 60
MOVE G6B, 165, 90, 75
MOVE G6C, 144, 10, 41
WAIT

```

```

MOVE G6A, 108, 78, 145, 98, 93, 60
MOVE G6D, 95, 43, 169, 110, 112, 60
MOVE G6B, 165, 90, 75
MOVE G6C, 144, 10, 41
WAIT

```

```

RETURN

```

```

=====

```

```

left_turn_1:

```

```

SPEED 6
MOVE G6D, 85, 71, 152, 91, 112, 60
MOVE G6A, 112, 76, 145, 93, 92, 60
MOVE G6C, 100, 40, 80, , , ,
MOVE G6B, 100, 40, 80, , , ,
WAIT

```

```

SPEED 9
MOVE G6A, 113, 75, 145, 97, 93, 60
MOVE G6D, 90, 50, 157, 115, 112, 60
MOVE G6B, 105, 40, 70, , , ,
MOVE G6C, 90, 40, 70, , , ,
WAIT

```

```

MOVE G6A, 108, 78, 145, 98, 93, 60
MOVE G6D, 95, 43, 169, 110, 110, 60
MOVE G6B, 105, 40, 70, , , ,
MOVE G6C, 80, 40, 70, , , ,
WAIT
RETURN

```

```

=====

```

```

izq_giro:

```

```

SPEED 5
MOVE G6D, 85, 71, 152, 91, 112, 60
MOVE G6A, 112, 76, 145, 93, 93, 60
MOVE G6B, 184, 93, 80
MOVE G6C, 150, 28, 21
WAIT

```

```

SPEED 6
MOVE G6A, 113, 75, 145, 97, 93, 60
MOVE G6D, 90, 50, 157, 115, 112, 60
MOVE G6B, 150, 28, 21
MOVE G6C, 150, 28, 21, , , ,
WAIT

```

```

MOVE G6A, 108, 78, 145, 98, 93, 60
MOVE G6D, 95, 43, 169, 110, 108, 60
MOVE G6B, 184, 93, 80
MOVE G6C, 150, 28, 21, , , ,
WAIT

```

```

RETURN

```

```

=====

```

right_turn:

SPEED 6

MOVE G6D, 113, 75, 145, 97, 93, 60
 MOVE G6A, 90, 50, 157, 115, 108, 60
 MOVE G6B, 144, 10, 41, , , , 'izquierda
 MOVE G6C, 165, 90, 75, , , ,

WAIT

MOVE G6D, 108, 78, 145, 98, 93, 60
 MOVE G6A, 95, 43, 169, 110, 108, 60
 MOVE G6B, 144, 10, 41, , , , 'izquierda
 MOVE G6C, 165, 90, 75, , , ,

WAIT

RETURN

'=====

right_turn_1:

SPEED 6

MOVE G6A, 85, 71, 152, 91, 112, 60
 MOVE G6D, 112, 76, 145, 93, 92, 60
 MOVE G6B, 100, 40, 80, , , ,
 MOVE G6C, 100, 40, 80, , , ,

WAIT

SPEED 9

MOVE G6D, 113, 75, 145, 97, 93, 60
 MOVE G6A, 90, 50, 157, 115, 112, 60
 MOVE G6C, 105, 40, 70, , , ,
 MOVE G6B, 90, 40, 70, , , ,

WAIT

MOVE G6D, 108, 78, 145, 98, 93, 60
 MOVE G6A, 95, 43, 169, 110, 110, 60
 MOVE G6C, 105, 40, 70, , , ,
 MOVE G6B, 80, 40, 70, , , ,

WAIT

RETURN

'=====

der_giro:

SPEED 5

MOVE G6A, 85, 71, 152, 91, 110, 60
 MOVE G6D, 112, 76, 145, 93, 92, 60
 MOVE G6B, 150, 28, 21, , , ,
 MOVE G6C, 150, 28, 21, , , ,

WAIT

SPEED 6

MOVE G6D, 113, 75, 145, 97, 93, 60
 MOVE G6A, 90, 50, 157, 115, 110, 60
 MOVE G6B, 150, 28, 21, , , , 'izquierda
 MOVE G6C, 150, 28, 21, , , ,

WAIT

```

MOVE G6D, 108, 78, 145, 98, 93, 60
MOVE G6A, 95, 43, 169, 110, 110, 60
MOVE G6B, 150, 28, 21, , , , 'izquierda
MOVE G6C, 150, 28, 21, , , ,

```

```

WAIT

```

```

RETURN

```

```

=====
left_shift:

```

```

SPEED 6
GOSUB left_shift1

```

```

SPEED 5
GOSUB left_shift2
GOSUB left_shift3
GOSUB left_shift4

```

```

SPEED 6
GOSUB left_shift3
GOSUB left_shift4

```

```

SPEED 6
GOSUB left_shift5
GOSUB left_shift6

```

```

RETURN

```

```

=====
left_shift1:

```

```

MOVE G6A, 85, 71, 152, 91, 112, 60,
MOVE G6D, 112, 76, 145, 93, 92, 60,
MOVE G6B, 165, 90, 75, , , , 'izquierda
MOVE G6C, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====
left_shift2:

```

```

MOVE G6D, 110, 92, 124, 97, 93, 70,
MOVE G6A, 76, 72, 160, 82, 124, 70,
MOVE G6B, 165, 90, 75, , , , 'izquierda
MOVE G6C, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====
left_shift3:

```

```

MOVE G6D, 93, 76, 145, 94, 109, 100,
MOVE G6A, 93, 76, 145, 94, 109, 100,
MOVE G6B, 165, 90, 75, , , , 'izquierda
MOVE G6C, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====

```

```

left_shift4:
    MOVE G6A, 110, 92, 124, 97, 93, 70,
    MOVE G6D, 76, 72, 160, 82, 124, 70,
    MOVE G6B, 165, 90, 75, , , , 'izquierda
    MOVE G6C, 144, 10, 41, , , ,
    WAIT
    RETURN

```

```

=====
left_shift5:
    MOVE G6D, 86, 83, 135, 97, 114, 60,
    MOVE G6A, 113, 78, 145, 93, 93, 60,
    MOVE G6B, 165, 90, 75, , , , 'izquierda
    MOVE G6C, 144, 10, 41, , , ,
    WAIT
    RETURN

```

```

=====
left_shift6:
    MOVE G6D, 85, 71, 152, 91, 112, 60,
    MOVE G6A, 112, 76, 145, 93, 92, 60,
    MOVE G6B, 165, 90, 75, , , , 'izquierda
    MOVE G6C, 144, 10, 41, , , ,
    WAIT
    RETURN

```

```

=====
right_shift:

    SPEED 5 '6
    GOSUB right_shift1

    SPEED 5
    GOSUB right_shift2
    GOSUB right_shift3
    GOSUB right_shift4

    SPEED 5 '6
    GOSUB right_shift3
    GOSUB right_shift4

    SPEED 5 '6
    GOSUB right_shift5
    GOSUB right_shift6

    RETURN

```

```

=====
right_shift1:
    MOVE G6D, 85, 71, 152, 91, 112, 60
    MOVE G6A, 112, 76, 145, 93, 92, 60
    MOVE G6C, 165, 90, 75, , , , 'izquierda
    MOVE G6B, 144, 10, 41, , , ,
    WAIT
    RETURN

```

```

=====
right_shift2:
    MOVE G6A, 110, 92, 124, 97, 93, 70
    MOVE G6D, 76, 72, 160, 82, 126, 70
    MOVE G6C, 165, 90, 75, , , , 'derecha

```

```

MOVE G6B, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====
right_shift3:
MOVE G6A, 93, 76, 145, 94, 109, 100
MOVE G6D, 93, 76, 145, 94, 109, 100
MOVE G6C, 165, 90, 75, , , , 'derecha
MOVE G6B, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====
right_shift4:
MOVE G6D, 110, 92, 124, 97, 93, 70
MOVE G6A, 76, 72, 160, 82, 126, 70
MOVE G6C, 165, 90, 75, , , , 'derecha
MOVE G6B, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====
right_shift5:
MOVE G6A, 86, 83, 135, 97, 114, 60
MOVE G6D, 113, 78, 145, 93, 93, 60
MOVE G6C, 165, 90, 75, , , , 'derecha
MOVE G6B, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====
right_shift6:
MOVE G6A, 85, 71, 152, 91, 112, 60
MOVE G6D, 112, 76, 145, 93, 92, 60
MOVE G6C, 165, 90, 75, , , , 'derecha
MOVE G6B, 144, 10, 41, , , ,
WAIT
RETURN

```

```

=====
bow_pose:

MOVE G6A, 100, 58, 135, 160, 100, 100
MOVE G6D, 100, 58, 135, 160, 100, 100
MOVE G6B, 100, 30, 80, , , ,
MOVE G6C, 100, 30, 80, , , ,
WAIT
RETURN

```

```

=====
sit_down_pose:

SPEED 10
MOVE G6A, 100, 151, 23, 140, 101, 100
MOVE G6D, 100, 151, 23, 140, 101, 100
MOVE G6B, 100, 30, 80, 100, 100, 100
MOVE G6C, 100, 30, 80, 100, 100, 100
WAIT
RETURN

```

```

=====
hans_up:
SPEED 5
MOVE G6A, 100, 76, 145, 93, 120 'PIERNA IZQUIERDA

```

```

MOVE G6D, 100, 76, 145, 93, 120      'PIERNA DERECHA
MOVE G6B, 100, 168, 150      'BRAZO IZQUIERDO
MOVE G6C, 100, 168, 150      'BRAZO DERECHO
WAIT
RETURN

```

```
'=====
```

```
'==RUTINA EJERCICIOS==
```

```
sentado:
```

```

SPEED 10
MOVE G6A,100, 168, 23, 140, 101
MOVE G6D,100, 168, 23, 140, 101
MOVE G6B,100, 30, 80, 100, 100
MOVE G6C,100, 30, 80, 100, 100
WAIT
SPEED 10
MOVE G6A,100, 168, 23, 180, 101
MOVE G6D,100, 168, 23, 180, 101
MOVE G6B,170, 30, 80, 100, 100
MOVE G6C,170, 30, 80, 100, 100
WAIT
SPEED 5
MOVE G6A,100, 168, 53, 180, 101
MOVE G6D,100, 168, 53, 180, 101
MOVE G6B,170, 30, 80, 100, 100
MOVE G6C,170, 30, 80, 100, 100
WAIT
SPEED 10
MOVE G6A,100, 76, 145, 93, 100
MOVE G6D,100, 76, 145, 93, 100
MOVE G6B,170, 30, 80, 100, 100
MOVE G6C,170, 30, 80, 100, 100
WAIT
RETURN

```

```
'=====
```

```
flexiones:
```

```

SPEED 10
MOVE G6A,100, 76, 145, 93, 100
MOVE G6D,100, 76, 145, 93, 100
MOVE G6B,170, 30, 80
MOVE G6C,170, 30, 80
WAIT
SPEED 10
MOVE G6A,100, 76, 145, 93, 100
MOVE G6D,100, 76, 145, 93, 100
MOVE G6B,180, 64, 42
MOVE G6C,180, 64, 42
WAIT
SPEED 10
MOVE G6A,100, 76, 145, 93, 100
MOVE G6D,100, 76, 145, 93, 100
MOVE G6B,170, 30, 80, 100, 100
MOVE G6C,170, 30, 80, 100, 100
WAIT
RETURN

```

```
'=====
```

```
posicion:
```

```

SPEED 4
MOVE G6A,100, 76, 145, 93, 142

```

```

MOVE G6D,100, 76, 145, 93, 110
MOVE G6B,120 , 62, 74
MOVE G6C,180, 15, 70
WAIT

```

```

SPEED 4
MOVE G6A,100, 76, 145, 93, 142
MOVE G6D,100, 76, 145, 93, 110
MOVE G6B,12 , 35, 10
MOVE G6C,180, 15, 70
WAIT
RETURN

```

```

=====
posicion_1:  'levantar de rutina de flexiones una mano

```

```

SPEED 5
MOVE G6A,100, 76, 145, 93, 142
MOVE G6D,100, 76, 145, 93, 110
MOVE G6B,12 , 35, 10
MOVE G6C,180, 15, 70
WAIT

```

```

SPEED 5
MOVE G6A,100, 76, 145, 93, 142
MOVE G6D,100, 76, 145, 93, 110
MOVE G6B,120 , 62, 74
MOVE G6C,180, 15, 70
WAIT

```

```

SPEED 5
MOVE G6A,100, 76, 145, 93, 142
MOVE G6D,100, 76, 145, 93, 110
MOVE G6B,180 , 20, 75
MOVE G6C,180, 20, 75
WAIT

```

```

SPEED 5
MOVE G6A,100, 76, 145, 93, 100
MOVE G6D,100, 76, 145, 93, 100
MOVE G6B,170, 30, 80, 100, 100
MOVE G6C,170, 30, 80, 100, 100
WAIT
RETURN

```

```

=====
flexiones_una_mano:

```

```

SPEED 1
MOVE G6A,100, 76, 145, 93, 142
MOVE G6D,100, 76, 145, 93, 110
MOVE G6B,12 , 35, 10
MOVE G6C,180, 20, 65
WAIT

```

```

SPEED 1
MOVE G6A,100, 76, 145, 93, 142
MOVE G6D,100, 76, 145, 93, 110
MOVE G6B,12 , 35, 10
MOVE G6C,180, 15, 70
WAIT
RETURN

```

```

=====
body_move:

```

```

SPEED 6
GOSUB body_move1
GOSUB body_move2
GOSUB body_move3
MOVE G6A, 93, 76, 145, 94, 109, 100

```



```

MOVE G6D, 93, 76, 145, 94, 109, 100
MOVE G6B, 100, 105, 100, , , ,
MOVE G6C, 100, 105, 100, , , ,
WAIT
MOVE G6A, 104, 112, 92, 116, 107
MOVE G6D, 79, 81, 145, 95, 108
MOVE G6B, 100, 105, 100
MOVE G6C, 100, 105, 100
WAIT
MOVE G6A, 93, 76, 145, 94, 109, 100
MOVE G6D, 93, 76, 145, 94, 109, 100
MOVE G6B, 100, 105, 100, , , ,
MOVE G6C, 100, 105, 100, , , ,
WAIT
MOVE G6D, 104, 112, 92, 116, 107
MOVE G6A, 79, 81, 145, 95, 108
MOVE G6B, 100, 105, 100
MOVE G6C, 100, 105, 100
WAIT
MOVE G6A, 93, 76, 145, 94, 109, 100
MOVE G6D, 93, 76, 145, 94, 109, 100
MOVE G6B, 100, 105, 100, , , ,
MOVE G6C, 100, 105, 100, , , ,
WAIT
GOSUB body_move3
GOSUB body_move2
GOSUB body_move1
RETURN
'=====
body_move3:
MOVE G6A, 93, 76, 145, 94, 109, 100
MOVE G6D, 93, 76, 145, 94, 109, 100
MOVE G6B, 100, 35, 90, , , ,
MOVE G6C, 100, 35, 90, , , ,
WAIT
RETURN
'=====
body_move2:
MOVE G6D, 110, 92, 124, 97, 93, 70
MOVE G6A, 76, 72, 160, 82, 128, 70
MOVE G6B, 100, 35, 90, , , ,
MOVE G6C, 100, 35, 90, , , ,
WAIT
RETURN
'=====
body_move1:
MOVE G6A, 85, 71, 152, 91, 112, 60
MOVE G6D, 112, 76, 145, 93, 92, 60
MOVE G6B, 100, 40, 80, , , ,
MOVE G6C, 100, 40, 80, , , ,
WAIT
RETURN
'=====

forward_standup:

SPEED 10

MOVE G6A, 100, 130, 120, 80, 110, 100
MOVE G6D, 100, 130, 120, 80, 110, 100
MOVE G6B, 150, 160, 10, 100, 100, 100
MOVE G6C, 150, 160, 10, 100, 100, 100
WAIT

MOVE G6A, 80, 155, 85, 150, 150, 100

```

```

MOVE G6D, 80, 155, 85, 150, 150, 100
MOVE G6B,185, 40, 60, 100, 100, 100
MOVE G6C,185, 40, 60, 100, 100, 100
WAIT

```

```

MOVE G6A, 75, 165, 55, 165, 155, 100
MOVE G6D, 75, 165, 55, 165, 155, 100
MOVE G6B,185, 10, 100, 100, 100, 100
MOVE G6C,185, 10, 100, 100, 100, 100
WAIT

```

```

MOVE G6A, 60, 165, 30, 165, 155, 100
MOVE G6D, 60, 165, 30, 165, 155, 100
MOVE G6B,170, 10, 100, 100, 100, 100
MOVE G6C,170, 10, 100, 100, 100, 100
WAIT

```

```

MOVE G6A, 60, 165, 25, 160, 145, 100
MOVE G6D, 60, 165, 25, 160, 145, 100
MOVE G6B,150, 60, 90, 100, 100, 100
MOVE G6C,150, 60, 90, 100, 100, 100
WAIT

```

```

MOVE G6A,100, 155, 25, 140, 100, 100
MOVE G6D,100, 155, 25, 140, 100, 100
MOVE G6B,130, 50, 85, 100, 100, 100
MOVE G6C,130, 50, 85, 100, 100, 100
WAIT

```

```

RETURN

```

```

=====

```

volando:

```

SPEED 5

```

```

MOVE G6A, 85, 71, 152, 91, 112, 60
MOVE G6D,112, 76, 145, 93, 92, 60
MOVE G6B,100, 40, 80, , , ,
MOVE G6C,100, 40, 80, , , ,
WAIT

```

```

MOVE G6A, 90, 98, 105, 115, 115, 60
MOVE G6D,116, 74, 145, 98, 93, 60
MOVE G6B,100, 150, 150, 100, 100, 100
MOVE G6C,100, 150, 150, 100, 100, 100
WAIT

```

```

MOVE G6A, 90, 121, 36, 105, 115, 60
MOVE G6D,116, 60, 146, 138, 93, 60
MOVE G6B,100, 150, 150, 100, 100, 100
MOVE G6C,100, 150, 150, 100, 100, 100
WAIT

```

```

MOVE G6A, 90, 98, 105, 64, 115, 60
MOVE G6D,116, 50, 160, 160, 93, 60
MOVE G6B,145, 110, 110, 100, 100, 100
MOVE G6C,145, 110, 110, 100, 100, 100
WAIT

```

```

FOR i = 10 TO 15
  SPEED i
  MOVE G6B,145, 80, 80, 100, 100, 100
  MOVE G6C,145, 80, 80, 100, 100, 100
  WAIT

```

```

        MOVE G6B,145, 120, 120, 100, 100, 100
        MOVE G6C,145, 120, 120, 100, 100, 100
        WAIT
NEXT i

DELAY 1000
SPEED 6

MOVE G6A, 90, 98, 105, 64, 115, 60
MOVE G6D,116, 50, 160, 160, 93, 60
MOVE G6B,100, 160, 180, 100, 100, 100
MOVE G6C,100, 160, 180, 100, 100, 100
WAIT

MOVE G6A, 90, 121, 36, 105, 115, 60
MOVE G6D,116, 60, 146, 138, 93, 60
MOVE G6B,100, 150, 150, 100, 100, 100
MOVE G6C,100, 150, 150, 100, 100, 100
WAIT
SPEED 4

MOVE G6A, 90, 98, 105, 115, 115, 60
MOVE G6D,116, 74, 145, 98, 93, 60
WAIT

MOVE G6A, 85, 71, 152, 91, 112, 60
MOVE G6D,112, 76, 145, 93, 92, 60
MOVE G6B,100, 40, 80, , ,
MOVE G6C,100, 40, 80, , ,
WAIT
RETURN
'=====

foot_up:
SPEED 5
MOVE G6A, 85, 71, 152, 91, 112, 60,
MOVE G6D, 112, 76, 145, 93, 92, 60,
MOVE G6B, 100, 40, 80, , , ,
MOVE G6C, 100, 40, 80, , , ,
WAIT
MOVE G6A, 90, 98, 105, 115, 115, 60,
MOVE G6D, 116, 74, 145, 98, 93, 60,
MOVE G6B, 100, 95, 100, 100, 100, 100,
MOVE G6C, 100, 105, 100, 100, 100, 100,
WAIT
MOVE G6A, 100, 151, 23, 140, 115, 100,
WAIT
DELAY 500
MOVE G6A, 85, 71, 152, 91, 112, 60,
MOVE G6D, 112, 76, 145, 93, 92, 60,
WAIT
RETURN
'=====

'==RUTINA TRILLER==
rutina_0:
FOR i= 5 TO 15
SPEED i
MOVE G6B, 156, 20, 51 'brazo izq
MOVE G6C, 161, 20, 51 'brazo der
WAIT
MOVE G6B, 156, 20, 61
MOVE G6C, 161, 20, 61

```

```
WAIT  
NEXT i  
RETURN
```

```
'=====
```

```
rutina_00:  
SPEED 10  
MOVE G6B, 156, 20, 51  
MOVE G6C, 161, 20, 51  
WAIT  
MOVE G6B, 156, 20, 55  
MOVE G6C, 161, 20, 55  
WAIT  
RETURN
```

```
'=====
```

```
rutina_5:  
SPEED 10  
MOVE G6A, 85, 76, 145, 93, 120  
MOVE G6D, 110, 76, 145, 93, 90  
MOVE G6B, 125, 45, 65  
MOVE G6C, 100, 125, 180  
OUT 52,1  
WAIT  
DELAY 50  
MOVE G6A, 100, 76, 145, 93, 105  
MOVE G6D, 100, 76, 145, 93, 100  
MOVE G6B, 125, 45, 55  
MOVE G6C, 125, 45, 55  
OUT 52,0  
WAIT  
DELAY 50  
MOVE G6A, 110, 76, 145, 93, 90  
MOVE G6D, 85, 76, 145, 93, 120  
MOVE G6B, 100, 125, 180  
MOVE G6C, 125, 45, 65  
OUT 52,1  
WAIT  
DELAY 50  
MOVE G6A, 100, 76, 145, 93, 105  
MOVE G6D, 100, 76, 145, 93, 100  
MOVE G6B, 125, 45, 55  
MOVE G6C, 125, 45, 55  
OUT 52,0  
WAIT  
DELAY 50  
RETURN
```

```
'=====
```

```
rutina_6:  
SPEED 9  
MOVE G6A, 85, 76, 145, 93, 120  
MOVE G6D, 110, 76, 145, 93, 90  
MOVE G6B, 190, 10, 45  
MOVE G6C, 190, 50, 100  
OUT 52,1  
WAIT  
DELAY 100  
MOVE G6A, 100, 76, 145, 93, 105  
MOVE G6D, 100, 76, 145, 93, 100  
MOVE G6B, 125, 45, 55  
MOVE G6C, 125, 45, 55  
OUT 52,0
```

```

WAIT
DELAY 100
MOVE G6A, 110, 76, 145, 93, 90
MOVE G6D, 85, 76, 145, 93, 120
MOVE G6B, 190, 50, 100
MOVE G6C, 190, 10, 45
OUT 52,1
WAIT
DELAY 100
MOVE G6A, 100, 76, 145, 93, 105
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 125, 45, 55
MOVE G6C, 125, 45, 55
OUT 52,0
WAIT
DELAY 100
RETURN

```

```

=====

```

```

rutina_1:
SPEED 3
MOVE G6A, 101, 76, 145, 93, 95
MOVE G6D, 95, 76, 145, 93, 110
OUT 52,1
WAIT
DELAY 280
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
OUT 52,0
WAIT
DELAY 220
MOVE G6A, 95, 76, 145, 93, 110
MOVE G6D, 101, 76, 145, 93, 95
OUT 52,1
WAIT
DELAY 280
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
OUT 52,0
WAIT
DELAY 220
RETURN

```

```

=====

```

```

rutina_2:
SPEED 3
MOVE G6A, 101, 76, 145, 93, 95
MOVE G6D, 95, 76, 145, 93, 110
MOVE G6B, 100, 90, 20
MOVE G6C, 100, 90, 20
WAIT
DELAY 200
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 100, 90, 10
MOVE G6C, 100, 90, 10
WAIT
DELAY 200
MOVE G6A, 95, 76, 145, 93, 110
MOVE G6D, 101, 76, 145, 93, 95
MOVE G6B, 100, 90, 20
MOVE G6C, 100, 90, 20
WAIT
DELAY 200
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100

```

```
MOVE G6B, 100, 90, 10
MOVE G6C, 100, 90, 10
WAIT
DELAY 200
RETURN
```

```
'=====
brazos:
SPEED 10
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 100, 90, 20
MOVE G6C, 100, 90, 20
```

```
WAIT
RETURN
```

```
'=====
brazos_1:
SPEED 10
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 100, 150, 190
MOVE G6C, 100, 150, 190
```

```
WAIT
RETURN
```

```
'=====
rutina_3:
```

```
SPEED 8
MOVE G6A, 101, 76, 145, 93, 95
MOVE G6D, 95, 76, 145, 93, 110
MOVE G6B, 100, 100, 160
MOVE G6C, 100, 150, 190
WAIT
DELAY 150
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 100, 150, 190
MOVE G6C, 100, 150, 190
WAIT
DELAY 150
MOVE G6A, 95, 76, 145, 93, 110
MOVE G6D, 101, 76, 145, 93, 95
MOVE G6B, 100, 150, 190
MOVE G6C, 100, 100, 160
WAIT
DELAY 150
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 100, 150, 190
MOVE G6C, 100, 150, 190
WAIT
DELAY 150
RETURN
```

```
'=====
rutina_4:
SPEED 8
MOVE G6A, 101, 76, 145, 93, 95
MOVE G6D, 95, 76, 145, 93, 110
```

```

MOVE G6B, 100, 100, 190
MOVE G6C, 100, 100, 190
WAIT
DELAY 150
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 100, 150, 190
MOVE G6C, 100, 150, 190
WAIT
DELAY 150
MOVE G6A, 95, 76, 145, 93, 110
MOVE G6D, 101, 76, 145, 93, 95
MOVE G6B, 100, 100, 190
MOVE G6C, 100, 100, 190
WAIT
DELAY 150
MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
MOVE G6B, 100, 150, 190
MOVE G6C, 100, 150, 190
WAIT
DELAY 150
RETURN

```

```

=====

```

forward_walk:

```

          SPEED 5
MOVE24 85, 71, 152, 91, 112, 60, 100, 40, 80, , , , 100, 40, 80, , , , 112, 76, 145, 93,
92, 60,

```

```

          SPEED 14
'left up
MOVE24 90, 107, 105, 105, 114, 60, 90, 40, 80, , , , 100, 40, 80, , , , 114, 76, 145, 93,
90, 60,
'-----

```

```

'left down
MOVE24 90, 56, 143, 122, 114, 60, 80, 40, 80, , , , 105, 40, 80, , , , 113, 80, 145, 90,
90, 60,
MOVE24 90, 46, 163, 112, 114, 60, 80, 40, 80, , , , 105, 40, 80, , , , 112, 80, 145, 90,
90, 60,

```

```

          SPEED 10
'left center
MOVE24 100, 66, 141, 113, 100, 100, 90, 40, 80, , , , 100, 40, 80, , , , 100, 83, 156,
80, 100, 100,
MOVE24 113, 78, 142, 105, 90, 60, 100, 40, 80, , , , 100, 40, 80, , , , 90, 102, 136, 85,
114, 60,

```

```

          SPEED 14
'right up
MOVE24 113, 76, 145, 93, 90, 60, 100, 40, 80, , , , 90, 40, 80, , , , 90, 107, 105, 105,
114, 60,

```

```

'right down
MOVE24 113, 80, 145, 90, 90, 60, 105, 40, 80, , , , 80, 40, 80, , , , 90, 56, 143, 122,
114, 60,
MOVE24 112, 80, 145, 90, 90, 60, 105, 40, 80, , , , 80, 40, 80, , , , 90, 46, 163, 112,
114, 60,

```

```

          SPEED 10
'right center
MOVE24 100, 83, 156, 80, 100, 100, 100, 40, 80, , , , 90, 40, 80, , , , 100, 66, 141,
113, 100, 100,

```

```
MOVE24 90, 102, 136, 85, 114, 60, 100, 40, 80, , , , 100, 40, 80, , , , 113, 78, 142,
105, 90, 60,
```

```
SPEED 14
```

```
'left up
```

```
MOVE24 90, 107, 105, 105, 114, 60, 90, 40, 80, , , , 100, 40, 80, , , , 113, 76, 145, 93,
90, 60,
```

```
'-----
```

```
SPEED 5
```

```
MOVE24 85, 71, 152, 91, 112, 60, 100, 40, 80, , , , 100, 40, 80, , , , 112, 76, 145, 93,
92, 60,
```

```
RETURN
```

```
'=====
```

```
backward_walk:
```

```
SPEED 5
```

```
GOSUB backward_walk1
```

```
SPEED 13
```

```
GOSUB backward_walk2
```

```
SPEED 7
```

```
GOSUB backward_walk3
```

```
GOSUB backward_walk4
```

```
GOSUB backward_walk5
```

```
SPEED 13
```

```
GOSUB backward_walk6
```

```
SPEED 7
```

```
GOSUB backward_walk7
```

```
GOSUB backward_walk8
```

```
GOSUB backward_walk9
```

```
SPEED 13
```

```
GOSUB backward_walk2
```

```
SPEED 5
```

```
GOSUB backward_walk1
```

```
RETURN
```

```
'=====
```

```
backward_walk1:
```

```
MOVE G6A, 85, 71, 152, 91, 112, 60
```

```
MOVE G6D, 112, 76, 145, 93, 92, 60
```

```
MOVE G6B, 100, 40, 80, , , ,
```

```
MOVE G6C, 100, 40, 80, , , ,
```

```
WAIT
```

```
RETURN
```

```
backward_walk2:
```

```
MOVE G6A, 90, 107, 105, 105, 114, 60
```

```
MOVE G6D, 113, 78, 145, 93, 90, 60
```

```
MOVE G6B, 90, 40, 80, , , ,
```

```
MOVE G6C, 100, 40, 80, , , ,
```

```
WAIT
```

```
RETURN
```

```
backward_walk9:
```

```
MOVE G6A, 90, 56, 143, 122, 114, 60
```



```

MOVE G6D,113, 80, 145, 90, 90, 60
MOVE G6B, 80, 40, 80, , , ,
MOVE G6C,105, 40, 80, , , ,
WAIT
RETURN

```

```

backward_walk8:
MOVE G6A,100, 62, 146, 108, 100, 100
MOVE G6D,100, 88, 140, 86, 100, 100
MOVE G6B, 90, 40, 80, , , ,
MOVE G6C,100, 40, 80, , , ,
WAIT
RETURN

```

```

backward_walk7:
MOVE G6A,113, 76, 142, 105, 90, 60
MOVE G6D, 90, 96, 136, 85, 114, 60
MOVE G6B,100, 40, 80, , , ,
MOVE G6C,100, 40, 80, , , ,
WAIT
RETURN

```

```

backward_walk6:
MOVE G6D, 90, 107, 105, 105, 114, 60
MOVE G6A,113, 78, 145, 93, 90, 60
MOVE G6C,90, 40, 80, , , ,
MOVE G6B,100, 40, 80, , , ,
WAIT
RETURN

```

```

backward_walk5:
MOVE G6D, 90, 56, 143, 122, 114, 60
MOVE G6A,113, 80, 145, 90, 90, 60
MOVE G6C,80, 40, 80, , , ,
MOVE G6B,105, 40, 80, , , ,
WAIT
RETURN

```

```

backward_walk4:
MOVE G6D,100, 62, 146, 108, 100, 100
MOVE G6A,100, 88, 140, 86, 100, 100
MOVE G6C,90, 40, 80, , , ,
MOVE G6B,100, 40, 80, , , ,
WAIT
RETURN

```

```

backward_walk3:
MOVE G6D,113, 76, 142, 105, 90, 60
MOVE G6A, 90, 96, 136, 85, 114, 60
MOVE G6C,100, 40, 80, , , ,
MOVE G6B,100, 40, 80, , , ,
WAIT
RETURN

```

```
'=====
```

```

rutina_serrucho:
SPEED 8

```

```

MOVE G6B, 160, 10, 50 'brazo izq
MOVE G6C, 175, 85, 15 'brazo der

```

```

MOVE G6A, 101, 76, 145, 93, 95
MOVE G6D, 95, 76, 145, 93, 110
OUT 52,1

```

WAIT
DELAY 150

MOVE G6B, 160, 10, 50 'brazo izq
MOVE G6C, 175, 40, 65 'brazo der

MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
OUT 52,0
WAIT
DELAY 150

MOVE G6B, 160, 10, 50 'brazo izq
MOVE G6C, 175, 85, 15 'brazo der

MOVE G6A, 95, 76, 145, 93, 110
MOVE G6D, 101, 76, 145, 93, 95
OUT 52,1
WAIT
DELAY 150

MOVE G6B, 160, 10, 50 'brazo izq
MOVE G6C, 175, 40, 65 'brazo der

MOVE G6A, 100, 76, 145, 93, 100
MOVE G6D, 100, 76, 145, 93, 100
OUT 52,0
WAIT
DELAY 150
RETURN

'=====

ANEXO 3: CÓDIGO DEL MICROCONTROLADOR DEL MÓDULO INFRARROJO.

```

$regfile = "m164pdef.dat"
$crystal = 8000000
$hwstack = 100
$swstack = 100
$framesize = 200
$timeout = 60000
$baud = 38400
***** Declaracion de Variables *****
Dim X As Word
Dim Z As Byte
Dim D6 As Byte
Dim S_radio As String * 20
Dim Buf6 As String * 100
Dim Strin_ritmo As String * 5
Dim Strin_play As String * 5
Dim Strin_camina As String * 5

Dim Alto As Word
Dim Bajo_1 As Word
Dim Espera0 As Word
Dim Espera1 As Word
Dim Espera2 As Word
Dim Espera3 As Word
Dim Espera4 As Word
Dim Espera5 As Word
Dim Espera6 As Word
Dim Espera7 As Word
Dim Espera8 As Word
Dim Espera9 As Word
Dim Espera10 As Word
Dim Espera11 As Word
Dim Espera12 As Word
Dim Espera13 As Word
Dim Lazo As Word
Dim Cancion As Byte

Config Portb.0 = Output           'LED EJERCICIOS
Config Portb.1 = Output           'LED EJERCICIOS
Config Portb.2 = Output           'LED EJERCICIOS
Config Portb.3 = Output           'LED EJERCICIOS
Config Portd.5 = Output
Config Portd.2 = Input            ' PULSADOR 1
Config Portd.3 = Input            ' PULSADO 2

Led_ejercicios Alias Portb.0
Led_bachata Alias Portb.1
Led_pop Alias Portb.2
Led_musica3 Alias Portb.3

Infra Alias Portd.5

Pulsa1 Alias Pind.2
Pulsa2 Alias Pind.3

Config Com1 = 19200 , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Open "COM1:" For Binary As #1
Enable Interrupts
On Urxc Serialingreso
Enable Urxc

***** Configuro LCD *****
Config Lcd = 16 * 2

```

```

Config Lcdpin = Pin , Db4 = Portc.2 , Db5 = Portc.3 , Db6 = Portc.4 , Db7 = Portc.5 , E = Portc.1 , Rs =
Portc.0
Cursor Off
Cls
Initlcd

```

```

For X = 1 To 4
  Set Led_ejercicios : Set Led_bachata : Set Led_pop : Set Led_musica3
  Waitms 50
  Reset Led_ejercicios : Reset Led_bachata : Reset Led_pop : Reset Led_musica3
  Waitms 50
Next X

```

```

Alto = 13
Bajo_1 = 9
Espera0 = 1

```

```

*****PROGRAMA PRINCIPAL *****

```

```

Inicio:
Locate 1 , 1
Lcd "*** HUMANOIDE ** "

```

```

'XYZ080VW
Print #1 , "RAFA" ; S_radio

```

```

***** RECEP TP DATOS DE PC *****

```

```

For X = 0 To 3
  If Mid(s_radio , X , 3) = "XYZ" Then
    Waitms 30
    Z = X + 3
    Strin_ritmo = Mid(s_radio , Z , 1)
    Z = X + 4
    Strin_play = Mid(s_radio , Z , 1)
    Z = X + 5
    Strin_camina = Mid(s_radio , Z , 1)
  End If
Next X

```

```

If Strin_play = "9" Then
  If Strin_ritmo = "A" Then
    Cancion = 1
    Locate 2 , 1
    Lcd " BACHATA "
    Set Led_ejercicios : Reset Led_bachata : Reset Led_pop : Reset Led_musica3
  End If
  If Strin_ritmo = "B" Then
    Cancion = 2
    Locate 2 , 1
    Lcd " EJERCICIOS "
    Reset Led_ejercicios : Set Led_bachata : Reset Led_pop : Reset Led_musica3
  End If
  If Strin_ritmo = "C" Then
    Cancion = 3
    Locate 2 , 1
    Lcd " POP "
    Reset Led_ejercicios : Reset Led_bachata : Set Led_pop : Reset Led_musica3
  End If
  If Strin_ritmo = "D" Then
    Cancion = 4
    Locate 2 , 1
    Lcd " BAILE 4 "
    Reset Led_ejercicios : Reset Led_bachata : Reset Led_pop : Set Led_musica3
  End If
Else
  Cancion = 0
  Locate 2 , 1

```

```

' Lcd " STOP      "
  Reset Led_ejercicios
  Reset Led_bachata
  Reset Led_pop
  Reset Led_musica3
End If

If Strin_camina = "1" Then
  Cancion = 5
End If
If Strin_camina = "2" Then
  Cancion = 6
End If
If Strin_camina = "3" Then
  Cancion = 7
End If
If Strin_camina = "4" Then
  Cancion = 8
End If

'Locate 2 , 15
'Lcd Cancion ; " "
If Cancion = 1 Then Gosub Pulsos2
If Cancion = 2 Then Gosub Pulsos3
If Cancion = 3 Then Gosub Pulsos4
If Cancion = 4 Then Gosub Pulsos5

If Cancion = 5 Then
  Locate 2 , 1
  Lcd " ADELANTE  "
  Gosub Pulsos13
End If

If Cancion = 6 Then
  Locate 2 , 1
  Lcd "  ATRAS   "
  Gosub Pulsos14
End If

If Cancion = 7 Then
  Locate 2 , 1
  Lcd " DERECHA  "
  Gosub Pulsos16
End If

If Cancion = 8 Then
  Locate 2 , 1
  Lcd " IZQUIERDA  "
  Gosub Pulsos16
End If

'If Cancion = 8 Then Gosub Pulsos15

If Cancion = 0 Then
  Locate 2 , 1
  Lcd " STOP      "
End If

Goto Inicio
End

Serialingreso:
D6 = Inkey(#1)
If D6 = 13 Then
  S_radio = Buf6
  Buf6 = ""
'Noecho

```

```

Else
  Buf6 = Buf6 + Chr(d6)
End If
Return

```

```

*****

```

```

' ***** BOTON 2 *****

```

```

Pulsos2:
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 493
Espera9 = 493 : Espera10 = 493
Espera11 = 992 : Espera12 = 992
Espera13 = 493

```

```

'0 (0)
For Lazo = 1 To 188
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitms Espera0

```

```

'1 (1)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera1

```

```

'2 (2)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera2

```

```

'3 (3)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera3

```

```

'4 (4)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera4

```

```

'5(5)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera5

```

```

'6 (6)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera6
'7 (7)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera7
'8 (8)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera8
'9 (9)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera9
'10 (10)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera10
'11 (11)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera11
'12 (12)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera12
'13 (13)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
Return

```

```

Pulsos3:
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 493
Espera9 = 493 : Espera10 = 992
Espera11 = 493 : Espera12 = 992
Espera13 = 493
'0 (0)
For Lazo = 1 To 188
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitms Espera0
'1 (1)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera1
'2 (2)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera2
'3 (3)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera3
'4 (4)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera4
'5(5)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera5
'6 (6)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera6
'7 (7)
For Lazo = 1 To 18

```



```

    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera7
'8 (8)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera8
'9 (9)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera9
'10 (10)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera10
'11 (11)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera11
'12 (12)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera12
'13 (13)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
Return

```

```

| ***** BOTON 4 *****

```

```

Pulsos4:
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 493
Espera9 = 493 : Espera10 = 992
Espera11 = 992 : Espera12 = 493
Espera13 = 493

```

```

'0 (0)

```

```

For Lazo = 1 To 188
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitms Espera0
'1 (1)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera1
'2 (2)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera2
'3 (3)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera3
'4 (4)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera4
'5(5)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera5
'6 (6)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera6
'7 (7)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera7
'8 (8)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1

```

```

    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera8
'9    (9)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera9
'10   (10)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera10
'11   (11)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera11
'12   (12)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera12
'13   (13)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
Return

```

```

' ***** BOTON 5 *****

```

```

Pulsos5:
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 493
Espera9 = 992 : Espera10 = 493
Espera11 = 493 : Espera12 = 992
Espera13 = 493
'0    (0)
For Lazo = 1 To 188
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitms Espera0
'1    (1)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1

```

```

Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera1
'2 (2)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera2
'3 (3)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera3
'4 (4)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera4
'5(5)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera5
'6 (6)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera6
'7 (7)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera7
'8 (8)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera8
'9 (9)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo

```

```

    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera9
'10 (10)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera10
'11 (11)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera11
'12 (12)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera12
'13 (13)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
Return
' ***** BOTON 13 ADELANTE *****
Pulsos13:
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 992
Espera9 = 493 : Espera10 = 992
Espera11 = 493 : Espera12 = 493
Espera13 = 493

'0 (0)
For Lazo = 1 To 188
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitms Espera0
'1 (1)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera1
'2 (2)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo

```

```

Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera2
'3 (3)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera3
'4 (4)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera4
'5(5)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera5
'6 (6)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera6
'7 (7)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera7
'8 (8)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera8
'9 (9)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra
Waitus Espera9
'10 (10)
For Lazo = 1 To 18
Reset Infra : Waitus Bajo_1
Set Infra : Waitus Alto
Next Lazo
Reset Infra : Waitus Bajo_1
Set Infra

```

```

    Waitus Espera10
'11 (11)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera11
'12 (12)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera12
'13 (13)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
Return
| ***** BOTON 14 ATRAS *****
Pulsos14:
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 992
Espera9 = 493 : Espera10 = 992
Espera11 = 992 : Espera12 = 993
Espera13 = 493
'0 (0)
For Lazo = 1 To 188
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitms Espera0
'1 (1)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera1
'2 (2)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera2
'3 (3)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera3

```

```

'4 (4)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera4
'5(5)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera5
'6 (6)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera6
'7 (7)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera7
'8 (8)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera8
'9 (9)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera9
'10 (10)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera10
'11 (11)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera11
'12 (12)
For Lazo = 1 To 18

```



```

    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera12
'13 (13)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
Return
! ***** BOTON 16 DERECHA IZQUIERDA *****

Pulsos16:
If Cancion = 7 Then
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 992
Espera9 = 992 : Espera10 = 493
Espera11 = 493 : Espera12 = 493
Espera13 = 493
End If

If Cancion = 8 Then
Espera1 = 493 : Espera2 = 992
Espera3 = 493 : Espera4 = 992
Espera5 = 493 : Espera6 = 493
Espera7 = 493 : Espera8 = 992
Espera9 = 992 : Espera10 = 493
Espera11 = 992 : Espera12 = 993
Espera13 = 493
End If

'0 (0)
For Lazo = 1 To 188
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitms Espera0
'1 (1)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera1
'2 (2)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1
    Set Infra
    Waitus Espera2
'3 (3)
For Lazo = 1 To 18
    Reset Infra : Waitus Bajo_1
    Set Infra : Waitus Alto
Next Lazo
    Reset Infra : Waitus Bajo_1

```

```

Set Infra
Waitus Espera3
'4 (4)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera4
'5(5)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera5
'6 (6)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera6
'7 (7)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera7
'8 (8)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera8
'9 (9)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera9
'10 (10)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera10
'11 (11)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera11

```

```
'12 (12)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
  Waitus Espera12
'13 (13)
For Lazo = 1 To 18
  Reset Infra : Waitus Bajo_1
  Set Infra : Waitus Alto
Next Lazo
  Reset Infra : Waitus Bajo_1
  Set Infra
Return
```