



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELECTRÓNICA**

**CARRERA AUTOMATIZACIÓN Y CONTROL**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

**AUTOR: PÉREZ LUIS, SÁNCHEZ ERNESTO**

**TEMA: DISEÑO DEL CONTROLADOR Y TELE-OPERACIÓN DEL  
BRAZO ROBÓTICO "ROBOTIC ARM EDGE" POR MEDIO DE UN  
DISPOSITIVO ANDROID Y PC**

**DIRECTOR: ING. EDGAR TIPÁN**

**CODIRECTOR: ING. XAVIER SEGOVIA**

**SANGOLQUÍ, NOVIEMBRE 2013**

**UNIVERSIDAD DE LA FUERZAS ARMADAS – ESPE**  
**INGENIERÍA EN ELECTRÓNICA,**  
**AUTOMATIZACIÓN Y CONTROL**

**CERTIFICADO**

Ing. Edgar Tipán  
Ing. Xavier Segovia

**CERTIFICAN**

Que el trabajo titulado “DISEÑO DEL CONTROLADOR Y TELE-OPERACIÓN DEL BRAZO ROBÓTICO “ROBOTIC ARM EDGE” POR MEDIO DE UN DISPOSITIVO ANDROID Y PC”, realizado por Luis Andrés Pérez Llerena y Ernesto Julián Sánchez Sánchez, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Andrés Pérez Llerena y Ernesto Julián Sánchez Sánchez que lo entreguen al Ingeniero Luis Orozco, en su calidad de Coordinador de la Carrera.

Sangolquí, 27 de Noviembre de 2013



Ing. Edgar Tipán  
DIRECTOR



Ing. Xavier Segovia  
CODIRECTOR

**UNIVERSIDAD DE LA FUERZAS ARMADAS – ESPE**  
**INGENIERÍA EN ELECTRÓNICA AUTOMATIZACIÓN Y CONTROL**

**DECLARACIÓN DE RESPONSABILIDAD**

**LUIS ANDRÉS PÉREZ LLERENA**  
**ERNESTO JULIÁN SÁNCHEZ SÁNCHEZ**

**DECLARAMOS QUE:**

El proyecto de grado denominado “DISEÑO DEL CONTROLADOR Y TELE-OPERACIÓN DEL BRAZO ROBÓTICO “ROBOTIC ARM EDGE” POR MEDIO DE UN DISPOSITIVO ANDROID Y PC”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 27 de Noviembre de 2013



**Luis Andrés Pérez Llerena**



**Ernesto Julián Sánchez Sánchez**

**UNIVERSIDAD DE LA FUERZAS ARMADAS – ESPE**

**INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL**

## **AUTORIZACIÓN**

Nosotros, Luis Andrés Pérez Llerena y Ernesto Julián Sánchez Sánchez

Autorizamos a la universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO DEL CONTROLADOR Y TELE-OPERACIÓN DEL BRAZO ROBÓTICO “ROBOTIC ARM EDGE” POR MEDIO DE UN DISPOSITIVO ANDROID Y PC”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría

Sangolquí, 27 de Noviembre de 2013



Luis Andrés Pérez Llerena



Ernesto Julián Sánchez Sánchez

## DEDICATORIA

*Esta tesis la dedico de manera muy especial a mis padres que me han apoyado incondicionalmente para poder llegar a esta instancia en mis estudios, quienes supieron guiarme por un buen camino, dándome fuerza para seguir siempre adelante y no desmayar ante los problemas.*

*Y a todas aquellas personas que de una u otra manera han contribuido para el logro de este objetivo.*

*Luis Andrés Pérez Llerena.*

## DEDICATORIA

*Dedico el presente proyecto tanto a mi familia como a los docentes de la universidad, quienes gracias a sus enseñanzas y consejos han logrado que crezca como persona, desarrollando nuevas habilidades que en la vida voy a poner en práctica.*

*A mis amigos que han logrado ser parte de mí y han hecho que forme parte de ellos, siempre pasando buenos y malos momentos pero siempre disfrutando de la vida.*

*Ernesto Julián Sánchez Sánchez.*

## AGRADECIMIENTO

*Agradezco a Dios por ser mi amigo fiel, por iluminar mi vida y darme la bendición de haber culminado una etapa muy importante, agradezco a mis padres Luis Oswaldo Pérez y Martha Cecilia Llerena, quienes siempre han sido el ejemplo a seguir, por darme siempre su amor, comprensión y apoyo absoluto en todos los momentos.*

*Un agradecimiento especial para mis abuelitos Papa Luchito y Mama Aurorita por ser parte tan importante de mi vida, que con su amor y ejemplo han logrado formar una familia unida y llena de valores.*

*A mis hermanos Anita y Cristian que con su paciencia, amor y comprensión me han sabido soportar, los quiero. A todas mis amigas y amigos por todo el apoyo y amistad incondicional.*

*Por último a todos los docentes de la Escuela Politécnica del Ejército, ya que son parte esencial de este logro, por su esfuerzo y empeño diario.*

*Por todo esto quiero darles mil gracias.*

*Luis Andrés Pérez Llerena.*



## AGRADECIMIENTO

*Para poder culminar este paso tan importante en mi vida, fue necesario el apoyo de varias personas a las que quiero y aprecio mucho.*

*En primer lugar agradezco a mis padres que con su paciencia, esfuerzo y apoyo constante han logrado guiarme por esta etapa para culminar con éxito.*

*A mis hermanos y amigos que siempre han estado constantemente conmigo en las buenas y malas, ayudándome incondicionalmente.*

*Y finalmente a Dios que el siempre me ha acompañado en toda mi vida.*

*Ernesto Julián Sánchez S.*



## ÍNDICE DE CONTENIDOS

	CAPÍTULO I.....	1
	DIAGNÓSTICO DE ESTUDIO .....	1
1.1.	Título del proyecto .....	1
1.2.	Objetivos .....	1
	Objetivo general .....	1
	Objetivos específicos.....	1
1.3.	Antecedentes del proyecto .....	1
1.4.	Justificación e importancia.....	2
1.5.	Alcance del proyecto .....	3
	CAPÍTULO II.....	4
	INTRODUCCIÓN .....	4
2.1.	Introducción a la robótica industrial.....	4
2.2.	Configuraciones morfológicas de robots industriales .....	6
2.3.	Grados de libertad .....	7
2.4.	Zona y espacio de trabajo.....	8
2.5.	Brazo robótico “robotic arm edge” .....	8
2.6.	Actuadores.....	11
2.6.1.	Motor DC .....	11
2.6.2.	Motores del brazo robótico “Robotic Arm Edge” .....	12
2.6.3.	Control de sentido de giro de motores DC .....	13
2.6.4.	Control de velocidad de motores DC. ....	15
2.7.	Comunicación inalámbrica wireless.....	16
2.7.1.	Comparación entre comunicaciones inalámbricas .....	18
2.8.	Protocolo TCP y UDP .....	19
2.9.	Descripción de conexiones.....	20
2.9.1.	Características de la red.....	21
2.10.	Micro controlador MiniCore RCM5600W .....	22
2.11.	Sensores del dispositivo android.....	24
2.12.	Giroscopio .....	25

2.13.	Acelerómetro.....	25
2.13.1	Aplicaciones comunes.....	25
2.13.2.	Tipos de acelerómetros.....	26
2.13.3.	Comparación entre tipos de acelerómetros .....	29
2.14.	Dispositivo android Sony Experia Miro.....	30
2.15.	NetBeans .....	31
	CAPÍTULO III.....	34
	INTERFACES DE COMUNICACIÓN Y SOFTWARE. ....	34
3.1.	Descripción del HMI desarrollado en Java. ....	36
3.1.1.	Flujograma programa JAVA.....	39
3.2.	Descripción del HMI del dispositivo móvil .....	40
3.2.1.	Diseño interface TouchOSC. ....	40
3.3.	Dynamic C.....	44
3.3.1.	Características Generales. ....	44
3.3.2.	Funciones Dynamic C. ....	46
3.3.3.	Diagrama de flujo del programa desarrollado en Dynamic C.....	52
3.4.	Comunicación CPU y MiniCore RCM5600W .....	52
3.5.	Modo Infraestructura MiniCore RCM5600W.....	54
3.6.	Modo Automático .....	57
3.6.1.	Descripción del proceso .....	57
3.6.2.	Aplicación desarrollada en simulador .....	59
	CAPÍTULO IV.....	61
	IMPLEMENTACIÓN Y RESULTADOS DE LAS PRUEBAS .....	61
4.1.	Toma de datos actuadores .....	61
4.2.	Driver de motor L293D.....	62
4.3.	Regulador 7805 .....	64
4.4.	PWM de la Tarjeta RCM5600 .....	65
4.5.	Diseño circuito de potencia. ....	66
4.6.	Diagramas eléctricos de conexión.....	69
4.7.	Cargar el diseño realizado en el editor TouchOsc hacia el teléfono .....	72
4.8.	Ejecutar y configurar aplicación TouchOSC .....	74
4.9.	Configuración del teléfono Sony Xperia ST23a como Access Point....	77

4.10.	Pruebas del sistema. ....	79
4.11.	Análisis de datos recibidos por la tarjeta RCM 5600W .....	80
4.12.	Medición de tiempos y velocidades de las articulaciones.....	86
4.13.	Consideraciones de operación.....	87
	CAPÍTULO V .....	88
	ANÁLISIS DE COSTOS.....	88
5.1.	Costo de construcción: .....	88
5.2.	Costo de materiales: .....	88
5.3.	Costo de puesta en marcha: .....	89
	CAPÍTULO VI.....	90
	CONCLUSIONES Y RECOMENDACIONES.....	90
6.1.	Conclusiones .....	90
6.2.	Recomendaciones.....	91
6.3.	Trabajo a futuro .....	91
	REFERENCIAS BIBLIOGRÁFICAS .....	93
	ANEXO 1 .....	<b>¡Error! Marcador no definido.</b>
	ANEXO 2 .....	<b>¡Error! Marcador no definido.</b>
	ANEXO 3 .....	<b>¡Error! Marcador no definido.</b>
	ANEXO 4.....	<b>¡Error! Marcador no definido.</b>
	ANEXO 5.....	<b>¡Error! Marcador no definido.</b>
	ANEXO 6.....	<b>¡Error! Marcador no definido.</b>

## ÍNDICE DE FIGURAS

Figura 1.	Cuadro sinóptico aplicaciones de robots industriales.....	5
Figura 2.	Robot Cartesiano .....	6
Figura 3.	Robot cilíndrico .....	6
Figura 4.	Robot Polar.....	6
Figura 5.	Robot antropomórfico .....	7
Figura 6.	Robot SCARA.....	7
Figura 7.	Robotic ARM-EDGE .....	8
Figura 8.	Alcance en grados de la base.....	10
Figura 9.	Alcance en grados del hombro .....	10
Figura 10.	Alcance en grados del codo.....	10
Figura 11.	Alcance en grados de la muñeca .....	10
Figura 12.	Alcance en centímetros del gripper .....	10
Figura 13.	Área de trabajo brazo robótico ARM-EDGE .....	11
Figura 14.	Control de cambio de giro con interruptor simple.....	13
Figura 15.	Control de cambio de giro con interruptor doble. ....	13
Figura 16.	Control de cambio de giro con relés.....	14
Figura 17.	Control de cambio de giro con transistores. ....	14
Figura 18.	Ejemplos del ciclo de trabajo PWM.....	15
Figura 19.	Control de velocidad con el integrado 555.....	16
Figura 20.	Control de velocidad por variación de tensión. ....	16
Figura 21.	Modelo de referencia OSI .....	19
Figura 22.	Esquema de conexiones.....	21
Figura 23.	Tarjeta Mini Core RCM5600W.....	23
Figura 24.	Diagrama de bloques del microprocesador 5000. ....	23
Figura 25.	Distribución de pines puerto J2 del modulo RCM5600W .....	24
Figura 26.	Acelerómetro mecánico.....	26
Figura 27.	Acelerómetro capacitivo.....	26
Figura 28.	Acelerómetro piezo-eléctrico .....	27
Figura 29.	Acelerómetro piezo-resistivo.....	27
Figura 30.	Acelerómetro térmico .....	28

Figura 31.	Acelerómetro capacitivo tipo MEMS.....	28
Figura 32.	Sony Experia Miro .....	30
Figura 33.	Dispositivo móvil actuando como Access point. ....	34
Figura 34.	Esquema general de comunicaciones .....	34
Figura 35.	Diagrama funcional del sistema .....	35
Figura 36.	Diseño del HMI en Java .....	36
Figura 37.	Descripción de cada funcionalidad del HMI Java.....	37
Figura 38.	HMI créditos en Java.....	38
Figura 39.	Diagrama de flujo programación general Java.....	39
Figura 40.	Ícono TouchOSC Editor. ....	40
Figura 41.	TouchOSC Editor. ....	40
Figura 42.	Nuevo, Abrir y Guardar.....	41
Figura 43.	Cortar, Copiar, Pegar y Borrar. ....	41
Figura 44.	Deshacer y Rehacer. ....	41
Figura 45.	Zoom y Cuadrícula. ....	42
Figura 46.	Sincronización y Acerca de. ....	42
Figura 47.	Panel de propiedades. ....	42
Figura 48.	Configuración propiedades HMI.....	43
Figura 49.	HMI del dispositivo móvil android-Ventana 1.....	43
Figura 50.	HMI del dispositivo móvil android-Ventana 2.....	44
Figura 51.	HMI del dispositivo móvil android-Ventana 3.....	44
Figura 52.	Ícono Dynamic C.....	45
Figura 53.	Ventana de configuración Dinamic C. ....	46
Figura 54.	Declaración de entradas y salidas de Dinamic C.....	50
Figura 55.	Diagrama de flujo programación general Dynamic C.....	52
Figura 56.	Comunicación entre sockets UDP .....	53
Figura 57.	Comunicación inalámbrica en modo infraestructura.....	55
Figura 58.	Configuración de la tarjeta RCM5600W modo infraestructura .....	56
Figura 59.	Posición HOME.....	57
Figura 60.	Agarre de material .....	58
Figura 61.	Traslado de material .....	58
Figura 62.	Material depositado en la banda transportadora.....	59

Figura 63.	Robot KUKA.....	59
Figura 64.	Traslado y almacenamiento de piezas metálicas .....	60
Figura 65.	Driver L293D .....	62
Figura 66.	Diagrama de pines L293D.....	63
Figura 67.	Diagrama interno del driver L293D .....	63
Figura 68.	Regulador 7805 .....	64
Figura 69.	Diagrama de pines del regulador 7805 .....	64
Figura 70.	Diagrama de pines puerto J2 de la tarjeta RCM5600W .....	66
Figura 71.	Esquema electrónico tarjeta de potencia. ....	67
Figura 72.	Esquema electrónico diseñado en ARES .....	67
Figura 73.	Diseño implementado.....	68
Figura 74.	Diseño implementado.....	68
Figura 75.	Diseño tarjeta de potencia realizado con relés. ....	69
Figura 76.	Conexión motores al puerto P1_A. ....	70
Figura 77.	Conexión del puerto P2_A hacia los puentes H .....	70
Figura 78.	Conexión sensores, foco y alimentación hacia el puerto P1_B.....	71
Figura 79.	Conexión RCM5600W hacia los puentes H y velocidades.....	71
Figura 80.	Conexión puerto P2_B hacia RCM5600 mas alimentaciones.....	71
Figura 81.	Diseño realizado en TouchOSC .....	72
Figura 82.	Aplicación WiFi FileTransfer.....	72
Figura 83.	Pantalla de presentación de la aplicación .....	73
Figura 84.	Ejecución de la aplicación .....	73
Figura 85.	Transferencia de archivos hacia el dispositivo móvil.....	74
Figura 86.	Ejecutar TouchOSC versión 1.9.1 .....	74
Figura 87.	Ventana de herramientas TouchOSC .....	75
Figura 88.	Parámetros de red TouchOSC .....	75
Figura 89.	Habilitar el uso del acelerómetro.....	76
Figura 90.	Cargar HMI diseñado .....	76
Figura 91.	Configuración Access Point paso 1 .....	77
Figura 92.	Configuración Access Point paso 2.....	77
Figura 93.	Configuración Access Point paso 3.....	78
Figura 94.	Acceso directo celular como Access Point.....	78

Figura 95.	Conectividad PC – RCM5600W .....	79
Figura 96.	Conectividad PC – Dispositivo móvil .....	79
Figura 97.	Ejemplo dato recibido por RCM5600W desde la Pc.....	81
Figura 98.	Ejemplo botón B1 Base Izquierda.....	82
Figura 99.	Ejemplo botón Acelerómetro On.....	82
Figura 100.	Comparación String y decimal datos acelerómetro.....	83
Figura 101.	Estado inicial .....	85
Figura 102.	Inclinación 1 .....	85
Figura 103.	Inclinación 2 .....	85
Figura 104.	Inclinación 3 .....	85
Figura 105.	Inclinación 4 .....	85



## ÍNDICE DE TABLAS

Tabla 1.	Resumen características articulaciones ARM-EDGE .....	10
Tabla 2.	Características de los motores del robot “Robotic Arm Edge” .....	12
Tabla 3.	Comunicaciones inalámbricas .....	18
Tabla 4.	Diferencias entre TCP y UDP .....	20
Tabla 5.	Distribución de direcciones IP en la red.....	22
Tabla 6.	Características de los tipos de acelerómetros .....	29
Tabla 7.	Características generales Sony Xperia Miro .....	30
Tabla 8.	Comparativo entre plataformas de desarrollo Java .....	32
Tabla 9.	Detalle de cada funcionalidad del HMI en Java.....	37
Tabla 10.	Descripción propiedades TouchOSC .....	43
Tabla 11.	Opciones de configuración TCP.....	46
Tabla 12.	Parámetros de red y definiciones usadas en Dinamic C.....	47
Tabla 13.	Canales PWM.....	51
Tabla 14.	Datos de corriente velocidad alta .....	61
Tabla 15.	Datos de corriente velocidad baja.....	61
Tabla 16.	Descripción pines L293D .....	63
Tabla 17.	Características máximas de operación del driver L293D.....	63
Tabla 18.	Descripción pines 7805 .....	64
Tabla 19.	Ciclos de trabajo PWM y velocidad de las articulaciones .....	65
Tabla 20.	Ciclos de trabajo PWM .....	66
Tabla 21.	Distribución de pines de la tarjeta RCM5600 .....	66
Tabla 22.	Descripción de símbolos figura 4.12 .....	70
Tabla 23.	Distribución de direcciones IP en la red.....	79
Tabla 24.	Palabra de control PC .....	80
Tabla 25.	Detalle de datos enviados desde Java- PC.....	80
Tabla 26.	Datos enviados desde el dispositivo androide al RCM5600W .....	81
Tabla 27.	Palabra de control dispositivo Android ventana 2.....	82
Tabla 28.	Palabra de control dispositivo android ventana 3.....	82
Tabla 29.	Desglose datos recibidos del acelerómetro.....	84
Tabla 30.	Análisis de datos acelerómetro.....	85

Tabla 31.	Tiempos por articulación.....	86
Tabla 32.	Velocidades por articulación.....	86
Tabla 33.	Costos construcción y diseño.....	88
Tabla 35.	Costos puesta en marcha.....	89

## RESUMEN

El presente proyecto se enfoca en la descripción de la operación y control de manera remota del brazo robótico “ROBOTIC ARM EDGE”, utilizando la tecnología wireless como medio de comunicación inalámbrica entre el operario y el brazo robótico. Además de la utilización de herramientas que están actualmente al alcance de cualquier persona como es un dispositivo android, teniendo como opción la posibilidad de manipular el movimiento del brazo robótico de manera táctil en el dispositivo móvil por medio de un HMI diseñado con la aplicación TouchOSC, a la vez que también se puede usar los sensores de posición específicamente el acelerómetro del sistema para realizar similar efecto, como segundo terminal de operación se tiene el computador en el cual se diseño un HMI en una plataforma java, dichos dispositivos se comunicaran remotamente con el módulo MINICORE RCM5600W el cual es el controlador del brazo robótico y posee estándar wi-fi.

Palabras clave:

- Robótica
- Wireless
- RCM5600W
- Java
- TouchOSC

## **CAPÍTULO I**

### **DIAGNÓSTICO DE ESTUDIO**

#### **1.1. Título del proyecto**

Diseño del controlador y tele-operación del brazo robótico “robotic arm edge” por medio de un dispositivo android y PC

#### **1.2. Objetivos**

##### **Objetivo general**

Diseñar el controlador del brazo robótico “ROBOTIC ARM EDGE” por medio de un dispositivo móvil compatible con la plataforma Android y PC utilizando comunicación inalámbrica mediante el dispositivo MiniCore RCM5600W con estándar WI-FI.

##### **Objetivos específicos**

- Diseñar la tele-operación y control del brazo robótico “ROBOTIC ARM EDGE”.
- Manipular el movimiento del brazo robótico inalámbricamente mediante los sensores de movimiento del propio sistema del dispositivo móvil.
- Investigar y programar las diferentes plataformas de programación expuestas para el desarrollo del proyecto.

#### **1.3. Antecedentes del proyecto**

El presente proyecto de investigación se orienta a desarrollar sistemas y aplicaciones móviles para realizar el monitoreo y control de forma inalámbrica del manipulador robótico “ROBOTIC ARM EDGE” además de la utilización del dispositivo MiniCore RCM5600W el cual ofrece soluciones de tele-operación para sistemas de control móvil a un bajo costo y consumo de recursos.

En la actualidad el ingeniero busca producir nuevas tecnologías que faciliten la supervisión y manipulación de forma remota de los diferentes procesos industriales, en los que pueden intervenir diferentes manipuladores robóticos y de esta forma tener el menor contacto posible con el manipulador, evitando así posibles accidentes y salvaguardar la integridad de los operarios, además de obtener información de manera remota de la producción y del proceso los cuáles en la actualidad son muy utilizados en procesos industriales a nivel mundial

#### **1.4. Justificación e importancia**

En la actualidad nuestro país se encuentra en vías de desarrollo tecnológico, con lo que hemos visto la necesidad de investigar nuevas tecnologías de control y monitoreo de manera remota; ampliando la visión del uso de la electrónica y nuevas recursos que la informática puede producir.

La principal justificación para el desarrollo de este proyecto radica en aumentar el alcance de la capacidad de manipulación de robots en ambientes remotos, desde que surgen estos sistemas de tele-operación hasta nuestros días se ve que es una tecnología en constante evolución, la cual brinda comodidad y seguridad para el operario y personal que lo está controlando.

Por lo tanto, con la ayuda de un robot tele-operado que interviene en un proceso industrial se obtendrá las siguientes ventajas: reducción de tiempo, ahorro de dinero, un importante aumento de calidad y control continuo desde cualquier lugar.

Además de poder manipular robots, mediante los sensores de movimiento del dispositivo móvil, desarrollando y maximizando el control del manipulador.

Por último, con el desarrollo de este proyecto se puede diseñar y construir controladores para robots funciones o características similares a los existentes en la industria pero de manera remota, estableciéndose de esta manera un avance tecnológico muy notable en nuestra industria electrónica.

### **1.5. Alcance del proyecto**

El presente proyecto contempla efectuar el estudio y diseño de un sistema de control de las articulaciones del robot “ROBOTIC ARM EDGE” de manera inalámbrica mediante el protocolo de comunicación UDP. Para lo cual se detalla varios puntos de los cuales nos ayudan a definir el tema propuesto; introduciendo como primer punto tenemos la interacción con los usuarios utilizando dos HMI, el primero desarrollado en una plataforma Java para una PC y otro diseñado mediante TouchOSC para el dispositivo móvil basado en una tecnología Android, permitiendo al dispositivo móvil ser un teach-pendant inalámbrico del robot, también podremos manipular el brazo robótico mediante los sensores de movimiento del teléfono.

Toda la comunicación y control del robot se lo realizará de manera inalámbrica mediante el módulo MiniCore RCM5600 cuya programación se lo realizará en el software Dynamic C. No se realizará control de posicionamiento del brazo robótico ya que este no cuenta con la instrumentación necesaria.

## **CAPÍTULO II**

### **INTRODUCCIÓN**

La tecnología está en una constante evolución, ya que el ser humano va creando nuevas y diferentes necesidades, es por eso que el departamento de Eléctrica y Electrónica, la carrera de Automatización y Control de la Universidad de las Fuerzas Armadas ESPE está contribuyendo constante en la investigación, aplicación y desarrollo de nuevas herramientas y tecnologías, contribuyendo al progreso del país.

En el presente capítulo se expone una pequeña introducción a la robótica industrial y varios conceptos importantes empleados en este proyecto, con el fin de explicar ciertas ideas para tener una mejor comprensión de este tipo de máquinas y así analizar la importancia a nivel industrial, facilitando las actividades laborales y realizando tareas peligrosas a las que se puede exponer el ser humano, en el tema de producción los robots mejoran los niveles de producción, no solo en tiempo sino en calidad.

#### **2.1. Introducción a la robótica industrial**

La robótica industrial es el estudio, diseño y uso de los robots donde existen actividades mecánicas y repetibles los cuales puede ser automatizables en la producción industrial, el trabajo puede ser monótono y puede causar cansancio y agotamiento en los operarios, introduciendo posibles errores los cuales ponen en peligro la integridad del trabajador a la vez que reduce la productividad, por eso estos puestos pueden estar ocupados por robots aumentando la calidad, precisión y rapidez superando a los de la mano humana y a la vez se que mantienen constantes a lo largo de la jornada laboral.



Las principales aplicaciones de robots que se tienen actualmente son en manufactura y cuyo aumento esperado en productividad justifica la inversión.

A continuación se describe las principales aplicaciones de la robótica industrial.

**Almacenamiento, carga y descarga de objetos.**\_ Esto permite distribuir y clasificar objetos a gran velocidad de materias primas o productos terminados.

**Operaciones industriales de mecanización.**\_ Como son el ensamblaje, soldadura, pintura, corte, taladrado, remachado, fundición, montaje, corte de piezas, etc. las cuales están asociadas en la fabricación del producto final.

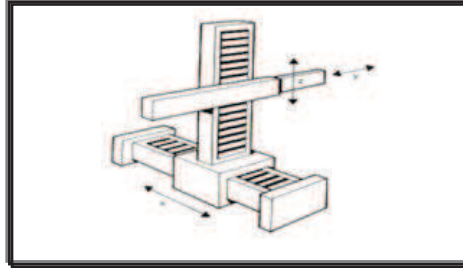
**Control de calidad.**\_ Utilizando diferentes mecanismos y herramientas para detectar posibles errores alcanzando un producto óptimo.



**Figura 1. Cuadro sinóptico aplicaciones de robots industriales.**

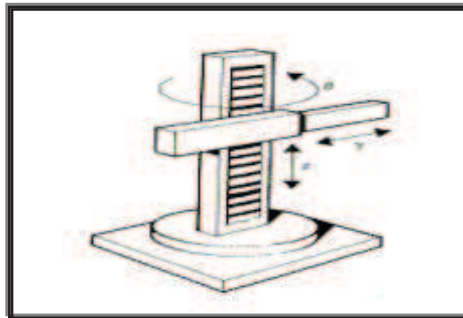
## 2.2. Configuraciones morfológicas de robots industriales

**Cartesiano.** Realiza 3 movimientos mediante articulaciones lineales en los ejes X, Y, Z, posee tres grados de libertad.



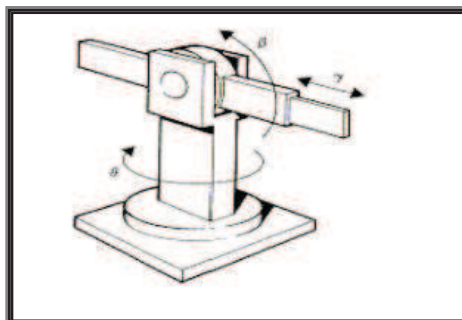
**Figura 2. Robot Cartesiano**

**Cilíndrico.** Realiza un movimiento rotacional sobre una base y dos lineales para el movimiento en altura y en radio goza de tres grados de libertad.



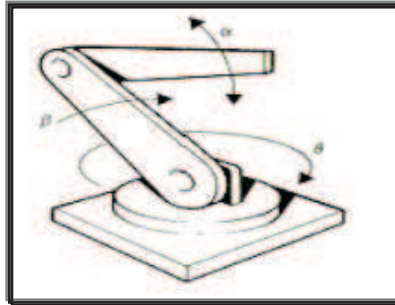
**Figura 3. Robot cilíndrico**

**Polar.** Tiene movimiento rotacional, angular y lineal, ya que cuenta con dos articulaciones rotacionales y una lineal.



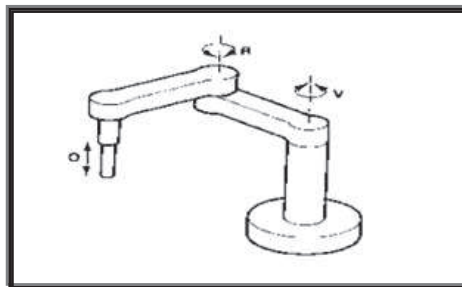
**Figura 4. Robot Polar**

**Antropomórfico.** Presenta con movimiento rotacional y dos angulares, con articulaciones que emulan al brazo humano, sobre la columna giratoria.



**Figura 5. Robot antropomórfico**

**SCARA.** Cuyas siglas en inglés corresponden a “Selective Compliant Articulated Robot Arm”, es un robot que posee dos articulaciones con movimiento rotacional y su tercera articulación realiza un movimiento lineal. (BitLibrary, 2008)



**Figura 6. Robot SCARA**

### 2.3. Grados de libertad

Es una característica fundamental al definir un robot ya que representa el número total de movimientos independientes que las articulaciones del manipulador pueden realizar, los grados de libertad se encuentran relacionados directamente con la anatomía y configuración del brazo robótico en nuestro caso se cuenta con 5 grados de libertad.

#### 2.4. Zona y espacio de trabajo

Ya que el robot físicamente posee una geometría fija y limitada, el espacio o volumen de trabajo se define como el límite de posiciones en el espacio que el robot puede alcanzar sin tomar en cuenta el actuador final, ya que el robot puede tener diferentes y distintos tamaños de grippers. También queda restringida la zona de trabajo por los límites de giro y desplazamiento que existen.

#### 2.5. Brazo robótico “robotic arm edge”

El brazo robótico “Robotic Arm Edge” posee una configuración antropomórfica, el cual es un mecanismo didáctico e interactivo usado para el aprendizaje e investigación de diferentes procesos industriales a escala, el cual puede ser controlado y manipulado por diferentes medios. En el presente proyecto se desarrolla un control de las articulaciones del robot a través de la tarjeta RCM 5600W y una manipulación de forma inalámbrica desde el Pc y el dispositivo android.



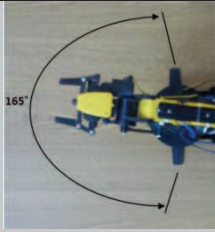
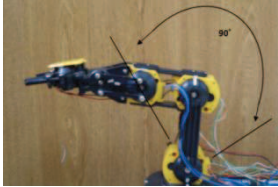
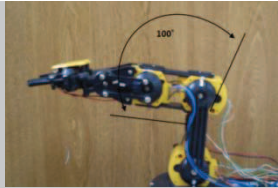
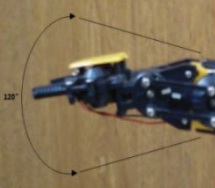
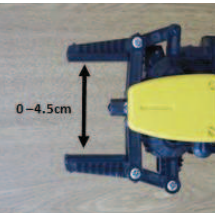
**Figura 7. Robotic ARM-EDGE**

### **Características generales**

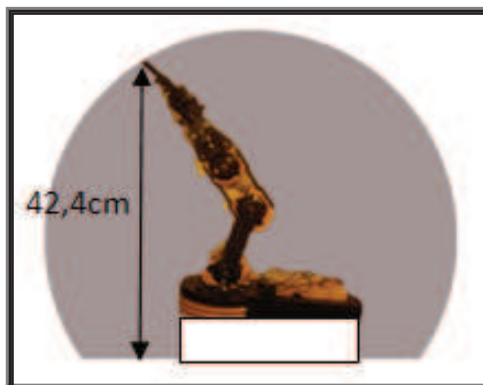
El brazo robótico cuenta con una pinza con capacidad de abrirse y cerrarse según la orden, tiene un movimiento de la muñeca de 120 grados, el codo de 100 grados, la rotación del hombro es de 90 grados, un movimiento desde la base de 165 grados, un alcance vertical de 35 centímetros, un alcance horizontal de 30 centímetros, una capacidad de elevación 100g, finalmente de 0 a 4.5 cm en movimiento de agarre.

Cuando uno de los motores dc encuentre excesiva resistencia al movimiento, la caja reductora hará un ruido que alertara para detener el movimiento del brazo en esa dirección. Un led blanco montado en la “mano” del brazo, iluminara lo que sea que la pinza está sosteniendo, activado este digitalmente. Sus dimensiones son de 22,8cm × 16cm × 38 cm y posee un peso de 1,5 libras. (ARM-EDGE, 2008)

**Tabla 1. Resumen características articulaciones ARM-EDGE**

Articulación	Movimiento	Descripción movimiento articulaciones
Base	0 ° a 165 °	
<b>Figura 8. Alcance en grados de la base</b>		
Hombro	0 ° a 90 °	
<b>Figura 9. Alcance en grados del hombro</b>		
Codo	0 ° a 100 °	
<b>Figura 10. Alcance en grados del codo</b>		
Muñeca	0 ° a 120 °	
<b>Figura 11. Alcance en grados de la muñeca</b>		
Gripper	0 a 4.5 cm	
<b>Figura 12. Alcance en centímetros del gripper</b>		

El área de trabajo del robot ARM EDGE se detalla en la figura 13.



**Figura 13. Área de trabajo brazo robótico ARM-EDGE**

## 2.6. Actuadores

Es un dispositivo el cual transforma un tipo de energía ya sea esta hidráulica, neumática o eléctrica en mecánica, con el fin de proveer fuerza para mover u actuar sobre otros elementos, en este caso todos los actuadores son motores eléctricos de corriente continua.

### 2.6.1. Motor DC

El motor dc es un dispositivo el cual se usa generalmente donde la energía proviene de baterías o fuentes reguladoras, el movimiento del motor, se obtienen a partir de la interacción de un campo magnético y eléctrico. Al pasar intensidad eléctrica por el conductor, este reacciona con el campo magnético produciendo una fuerza de modo que.

$$F = \beta \cdot i \cdot L$$

Donde  $\beta$  es el flujo magnético,  $i$  será la corriente que circula por el conductor y  $L$  la longitud del mismo. Esta fuerza será la responsable de producir el par por tanto el movimiento del motor.



El campo magnético se origina a causa de los polos magnéticos, generados en una bobina arrollada sobre un núcleo ferro magnético. En general los polos de estos motores se ubican en el estator.

Los conductores se ubican en el núcleo ferro magnético móvil para facilitar que al producirse una fuerza sobre ellos pueda girar y provocar el par, a este conjunto se lo conoce como rotor.

### 2.6.2. Motores del brazo robótico “Robotic Arm Edge”

En el presente proyecto, los motores que utiliza el robot “Robotic Arm Edge” en cada una de sus articulaciones poseen las siguientes características:

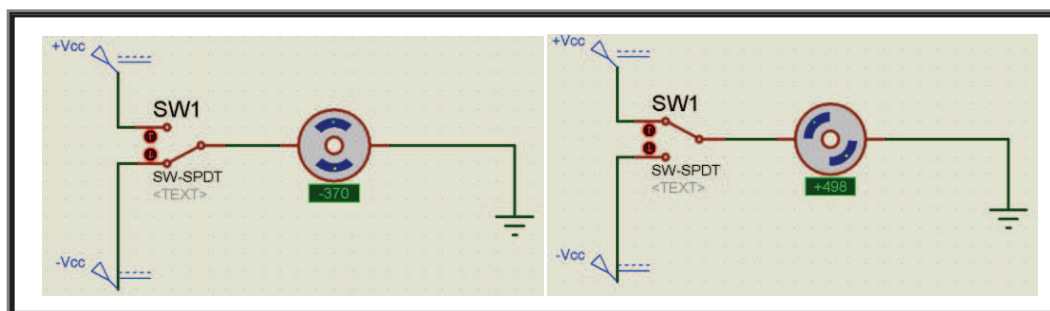
**Tabla 2. Características de los motores del robot “Robotic Arm Edge”**

Detalle	Especificaciones
Voltaje de operación	3V-6V DC
Corriente sin carga	255mA
Velocidad	12340rpm $\pm$ 12%
Torque de bloqueo	60cm*gr
Corriente de bloqueo	3170mA
Largo	35mm
Ancho	25mm
Altura	32.5mm

Para lograr un mejor torque, en cada una de las articulaciones los motores poseen cajas reductoras.

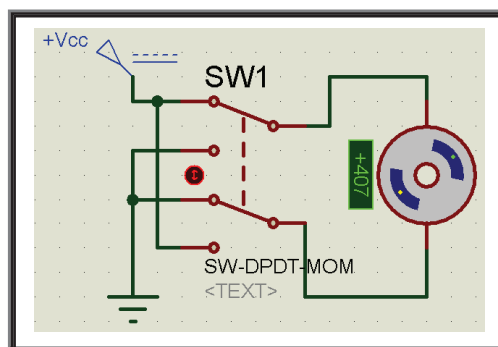
### 2.6.3. Control de sentido de giro de motores DC

Para realizar el control de sentido de giro para un motor DC existe varias formas, una de ellas es utilizar una fuente simétrica o dos fuentes con un interruptor simple de dos contactos, como se puede observar en la figura 14.



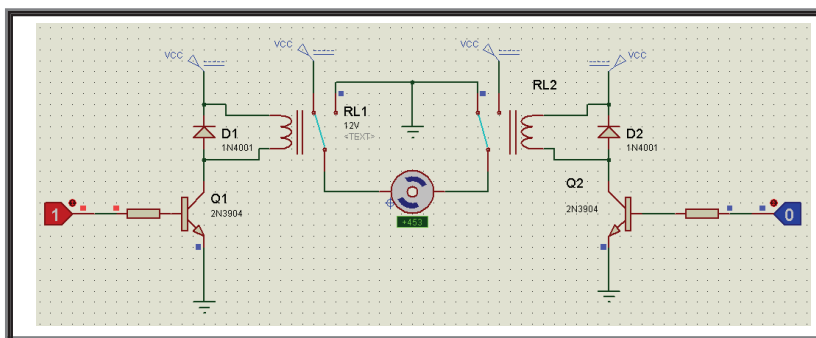
**Figura 14. Control de cambio de giro con interruptor simple.**

Una forma de conseguir cambiar el sentido de giro de un motor DC es utilizar una fuente común con un interruptor de 4 contactos como se puede observar en la figura 15.



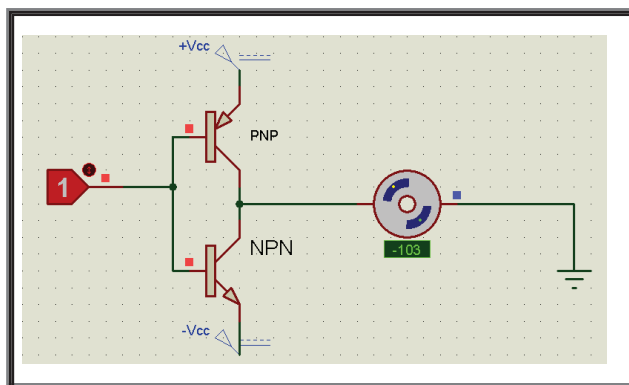
**Figura 15. Control de cambio de giro con interruptor doble.**

Aplicando el mismo principio se puede utilizar relés, mediante un circuito de control de activación y desactivación de dichos relés; el único inconveniente que presenta el relé es que al ser un dispositivo electromecánico con el pasar del tiempo presenta un desgaste en su parte mecánica presentando posteriormente problemas en el control de la carga.



**Figura 16. Control de cambio de giro con relés.**

Otro método de realizar el control de sentido de giro es utilizar transistores, recalcando que uno debe ser tipo PNP y otro NPN dependiendo de una señal de control para la activación o desactivación según el caso, tal como se puede observar en la figura 17.



**Figura 17. Control de cambio de giro con transistores.**

Los drivers también conocidos como puentes H, son integrados los cuales por su lógica permite controlar motores DC en cuanto a la potencia suministrada, teniendo como principales aplicaciones en la robótica, controles de posición y velocidad, controles de motores a paso o como interface para sistemas de corriente.

Existen una gran variedad de drivers para el control de motores DC siendo los más utilizados los integrados L293D y L293B, los cuales proporcionan varias ventajas entre las sobresaliente es la reducción de tamaño del circuito de control del

motor, además que poseen compatibilidad con señales TTL y pueden controlar cuatro motores con un solo sentido de giro o dos motores con doble sentido de giro, haciendo de estos driver los más factibles para el presente proyecto.

#### 2.6.4. Control de velocidad de motores DC.

El primer método de control es a través de la modulación por ancho de pulso (PWM), siendo el más utilizado para el control de velocidad, consiste en el envío de un tren de pulsos variando la duración del tiempo en alto, manteniéndose de esta forma la frecuencia de los pulsos constante, dando como resultado que el ciclo de trabajo de una señal periódica es igual a la relación entre el ancho relativo de su parte positiva con su periodo.

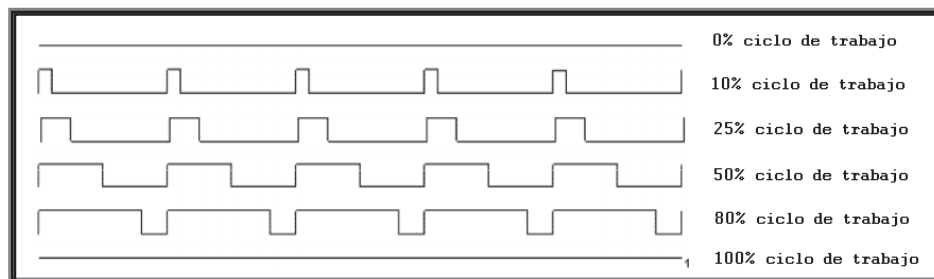
$$D = \frac{\tau}{T}$$

Donde:

D=Ciclo de trabajo

$\tau$ = tiempo de la función positiva (ancho de pulso)

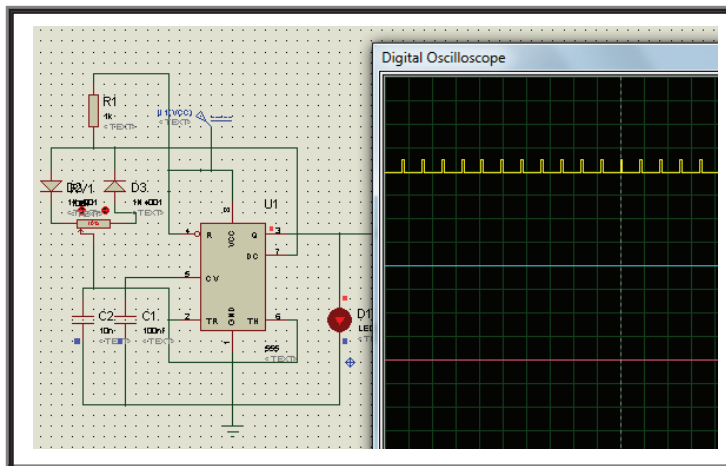
T= periodo de la función.



**Figura 18. Ejemplos del ciclo de trabajo PWM**

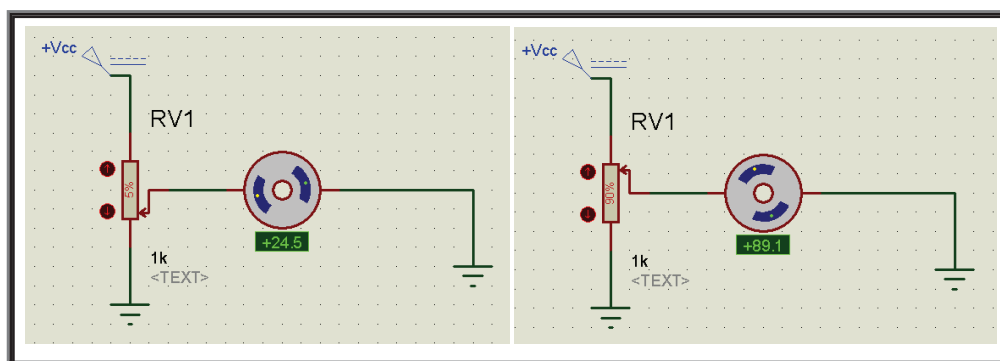
Otro método de control es la modulación por frecuencia de pulsos (PFM), utilizando el integrado 555 utilizado como un circuito astable con un potenciómetro

usado para variar la frecuencia de pulsos como se puede observar en la siguiente figura.



**Figura 19. Control de velocidad con el integrado 555.**

Por último, existe el método de variación de voltaje con la que se está alimentando al motor, siendo el voltaje directamente proporcional a la velocidad, como se observa en la figura 20.



**Figura 20. Control de velocidad por variación de tensión.**

## 2.7. Comunicación inalámbrica wireless

Mecanismo de conexión de dispositivos electrónicos de manera inalámbrica eliminando el uso de cables y ofreciendo gran flexibilidad en las comunicaciones, permitiendo al usuario la movilidad dentro de un perímetro y fácil acceso a la red,

se rige a los estándares IEEE 802.11. La Wifi Alliance es una asociación que asegura la interoperabilidad de los productos que están basados en el estándar 802.11

**IEEE 802.11:** Fue el primer estándar creado en 1997, su máxima conexión es de 2Mbps, transmitidos por señales infrarrojas en la banda de 2.4 GHz.

**IEEE 802.11a:** Creada en 1999, la velocidad máxima de conexión es de 54Mbps, funciona en la frecuencia de 5GHz a una distancia de 10m, la desventaja de este estándar es el uso de equipos únicamente a puntos de línea vista ya que objetos bloquean fácilmente la frecuencia de 5GHz.

**IEEE 802.11b:** También fue creado en 1999, la velocidad de conexión es de 11 Mbps en una frecuencia de 2.4 GHz a una distancias de 50m en ambientes cerrados y 200 en abiertos.

**IEEE 803.11d:** Está ideado para permitir el uso internacional de las redes 802.11 locales. Permite que distintos dispositivos intercambien información cumpliendo con regulaciones locales.

**IEEE 802.11e:** Está destinado a mejorar la calidad del servicio en el nivel de la capa de enlace de datos, para permitir mejores transmisiones de audio y vídeo.

**IEEE 802.11f:** Comunicación entro de una red WLAN mutiproveedor, permitiendo al usuario intercambio de proveedores mientras esta en movimiento.

**IEEE 802.11g:** Creada en el 2003 es un estándar que ofrece un ancho de banda elevado con 54 Mbps, en el rango de frecuencia de 2,4 GHz, tiene un alcance de 27m en ambientes cerrados y 75 en abiertos. Es compatible con el estándar 802.11b

**IEEE 802.11i:** Destinada para el mejoramiento de la seguridad para protocolos de autenticación y de codificación en la transferencia de datos

**IEEE 802.11n:** Lanzado en el 2009, funciona en ambas bandas 2.4 y 5 GHz y una velocidad máxima de hasta 600 Mbps. (kioskea)

**IEEE 802.11ac y ad:** Nuevos estándares que todavía están en desarrollo con el fin de aumentar la velocidad de transmisión de datos. Aspiran a trabajar en la banda de 5 GHz y 60 GHz y obtener velocidades de hasta 7 Gbps.

### 2.7.1. Comparación entre comunicaciones inalámbricas

Existen diferentes tecnologías de comunicación inalámbrica, que por su uso, comunicación, banda ancha, velocidad, personalización de dispositivos, transferencia de datos, uso en redes de sensores y redes multimedia, son razones para explicar la importancia de las tecnologías inalámbricas en el mercado de las telecomunicaciones siendo en la actualidad indispensable en nuestra sociedad de la información. La tabla 3 muestra el resumen de las comunicaciones inalámbricas.

**Tabla 3. Comunicaciones inalámbricas** (Jürgens, 2008)

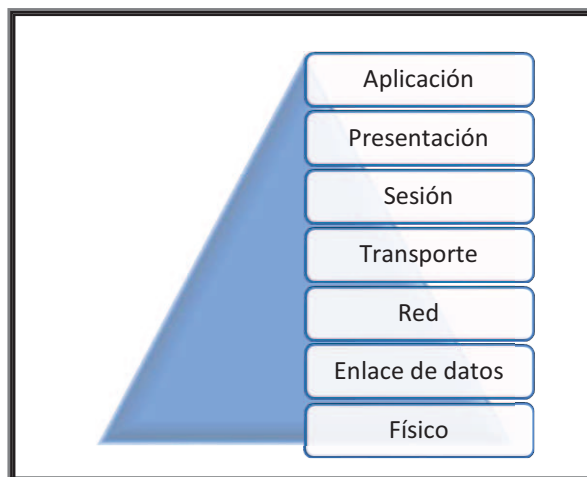
Tipo de red	WWAN (Wireless Wide Area Network)	WMAN (Wireless Metropolitan Area Network)	WLAN (Wireless Local Area Network)	WPAN (Wireless Personal Area Network)
<b>Estándar</b>	GSM/GPRS/UMTS	IEEE 802.16	IEEE 802.11	IEEE 802.15
<b>Denominación / Certificación</b>	2G / 3G	WIMAX	WIFI	BLUETOOTH, ZIGBEE
<b>Velocidad</b>	9.6/170/2000 Kbps	15-134 Mbps	1/2/11/54 Mbps	721 Kbps
<b>Frecuencia</b>	0.9/1.8/2.1 GHz	2/66 GHz	2.4 y 5 GHz	2.4 GHz
<b>Rango</b>	35 km	1 – 50 km	30 – 150 m	10 m
<b>Itinerancia (roaming)</b>	Si	Si	Si	No

## 2.8. Protocolo TCP y UDP

Un protocolo es un sistema que ayuda a determinar un conjunto de reglas de comunicación con el fin de que los distintos dispositivos puedan intercambiar mensajes través de la red.

Los protocolos TCP y UDP se encuentran se encuentran ubicados en el nivel de transporte del modelo de referencia OSI. Figura 21.

Esta capa se encarga del transporte de los datos, de un emisor hacia un receptor independientemente del medio físico que se esté usando.



**Figura 21. Modelo de referencia OSI**

A continuación se presenta un cuadro comparativo entre los protocolos TCP y UDP.



**Tabla 4. Diferencias entre TCP y UDP** (zonesecurity, 2011)

PROCOLO TCP	PROCOLO UDP
Orientado a la conexión	No orientado a la conexión
Transferencia confiable de datos	No garantiza la integridad de los datos
Envío/Recepción de paquetes entre 2	Envío/Recepción de paquetes entre 2 o más
IP: Identifica Host Origen y Host Destino	Identifica Puerto Origen y Puerto Destino
Utiliza ack indicando la recepción de datos	No utiliza acks
Un paquete se relaciona con el otro (primero, segundo,...,último)	No hay relación entre uno y otro, por lo tanto no hay forma de saber cuál es el primero o el último.
Deben llegar organizados	Todos tratan de llegar al destino lo más pronto posible.
Una única conexión	Puede recibir datos de hosts independientes
Lenta al hacer comprobación de datos	Rápida al no hacer seguimiento de parámetros

Con el fin de mantener una uniformidad en la comunicación ya que TouchOSC basa su transmisión en el envío de mensajes UDP y analizar las características del proyecto, se optó por implementar el protocolo UPD tanto en Java como en el Micro RCM 5600W.

## 2.9. Descripción de conexiones

La comunicación se la realiza de manera inalámbrica por medio de ondas electromagnéticas eliminando el uso de cables, los dispositivos inalámbricos a comunicar son los detallados en la figura 22. El módulo MiniCore RCM5600W es el encargado de recibir las ordenes del PC o del dispositivo móvil para controlar el robot según sea la configuración, enviando instrucciones binarias a un puente H, el cual es el encargado de la operación de cada uno de los motores de las articulaciones y el movimiento del brazo robótico.

La red posee un router Wi-Fi como punto central de conexión entre el dispositivo móvil, la tarjeta RCM5600W y la PC, así de esta manera proporciona conectividad entre los tres dispositivos.



**Figura 22. Esquema de conexiones**

### 2.9.1. Características de la red

- Está basado en el estándar 802.11b, proporciona una velocidad máxima de transmisión de 11Mbps, trabaja a una frecuencia de 2.4 GHz y proporciona una velocidad máxima de 5.9 Mbits/s sobre TCP y 7.1 Mbits/s sobre UDP.
- Está basada en una topología tipo estrella, ya que todos los dispositivos están conectados directamente a un punto central, para este caso en el proyecto es un Access Point.
- La red es pequeña por tal motivo se utiliza una red clase C, con máscara  $\backslash 24$  y expandible hasta 254 host.

Se designaron las siguientes IP, puertas de enlace y máscara las cuales son detalladas en la Tabla 5.

**Tabla 5. Distribución de direcciones IP en la red**

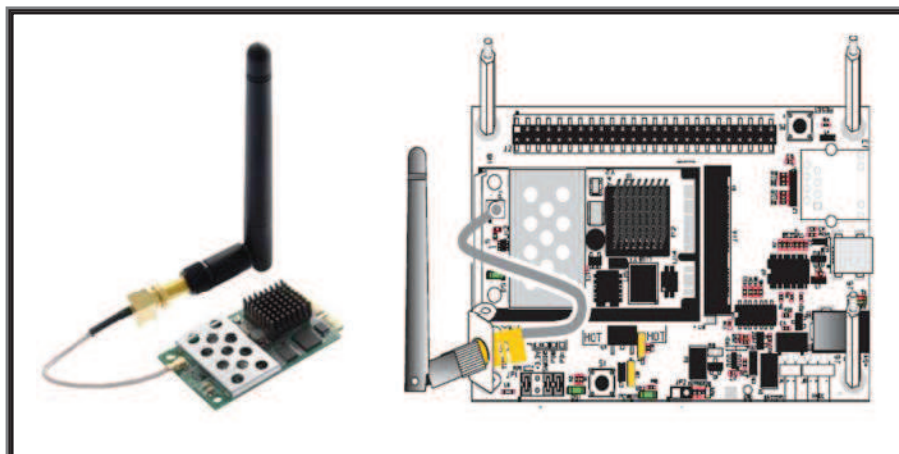
Dispositivo	IP	Mascara de red	Puerta de enlace
PC	192.168.43.242	255.255.255.0	192.168.43.1
RCM 5600W	192.168.43.243	255.255.255.0	192.168.43.1
Dispositivo Android	192.168.43.1	255.255.255.0	

### 2.10. Micro controlador MiniCore RCM5600W

El módulo Wi-Fi RCM5600 es un mini-microprocesador distribuido por Rabbit, con la capacidad de comunicación a través de una red inalámbrica Wi-Fi 802.11 g/b.

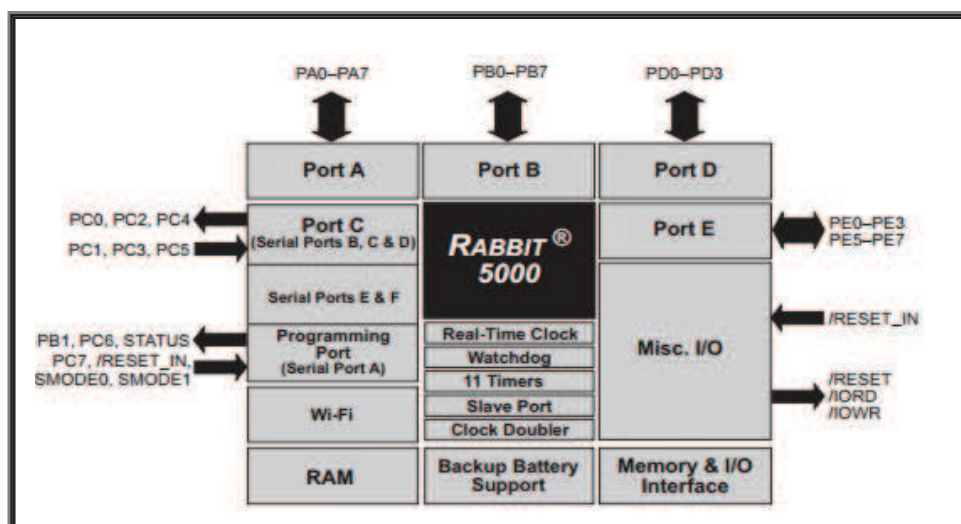
El microprocesador opera a 74MHz, tiene una memoria flash de 1 MB, memoria RAM de datos de 1 MB, 35 líneas I/O de propósito general, una entrada adicional de Reset y otra de Status , 6 puertos CMOS de alta velocidad configurables como asíncrona; 4 SPI y 2 como SDLC/HDL, un modulo Wi-fi 802.11 b/g trabajando a 2,4 GHz con seguridad WEP de 64 bits o 128 bits, posee un reloj en tiempo real, funcionalidades de watchdog, canales PWM, funciona con una alimentación de 3.15- 3.45 Vdc, en la transmisión y recepción de datos consume 625mA de datos y en reposo 85mA, el rango de temperatura en el que trabaja es de -30°C a 55 °C. (Rabbit)

La programación se lo realiza en lenguaje C, utilizando un compilador diseñado por Rabbit llamado Dinamic C el cual se lo detallara posteriormente.



**Figura 23. Tarjeta Mini Core RCM5600W**

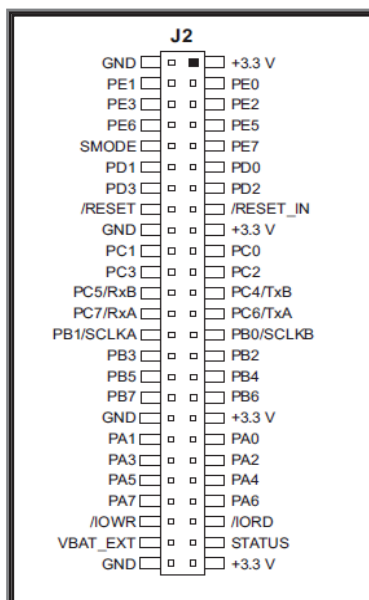
En la figura 24 se detalla el diagrama de bloques de los puertos del microprocesador 5000 que utiliza RCM5600w, estos son configurables por lo que los valores de fabrica predeterminados pueden ser reconfigurados, este micro controlador tiene el papel fundamental en el proyecto ya que servirá como un medio de intercambio de información entre el PC y el brazo robótico tanto para el dispositivo móvil y el robot.



**Figura 24. Diagrama de bloques del microprocesador 5000.**

### Configuración de pines.

Para la interacción con los puertos, la tarjeta RCM5600W dispone de 50 pines de entradas y salidas, distribuidos como se muestra en la figura 25.



**Figura 25. Distribución de pines puerto J2 del modulo RCM5600W**

### 2.11. Sensores del dispositivo android

Los sensores en dispositivos móviles se encuentran en una constante evolución por sus diferentes aplicaciones, permitiendo obtener una mayor información del entorno en el que se encuentra accediendo a variables como la posición, velocidad, orientación, fuerza, etc. en tiempo real, ayudando al desarrollo nuevas aplicaciones móviles que puedan mejorar y optimizar trabajos y recursos en varias aspectos de la vida, para nuestro caso dichos datos serán transmitidos inalámbricamente al modulo RCM600W el cual controla nuestro brazo robótico.

Las nuevas tecnologías de dispositivos móviles incorporan sensores como son el giroscopio y el acelerómetro. Un giroscopio permite cambiar la orientación del móvil haciendo girar su pantalla y el acelerómetro permite obtener la magnitud de la aceleración o vibraciones a la cual se somete el dispositivo.

## **2.12. Giroscopio**

“El giroscopio o giróscopo es un dispositivo formado esencialmente por un cuerpo con simetría de rotación que gira alrededor de su eje de simetría. Cuando se somete el giroscopio a un momento de fuerza que tiende a cambiar la orientación del eje de rotación su comportamiento es aparentemente paradójico ya que el eje de rotación, en lugar de cambiar de dirección como lo haría un cuerpo que no girase, cambia de orientación en una dirección perpendicular a la dirección “intuitiva”.  
(es.wikipedia.org)

Un giroscopio en un dispositivo móvil permite medir y mantener la orientación, monitorizando y controlando la posición del dispositivo.

## **2.13. Acelerómetro**

Es un dispositivo que ayuda a conocer la magnitud de la aceleración producida al efectuarse un movimiento respecto a un eje de referencia, una particularidad del acelerómetro es que puede ser utilizado como sensor de inclinación.

El acelerómetro usado como sensor de inclinación tiene su mayor sensibilidad cuando sus ejes se encuentran perpendiculares a la aceleración de la gravedad.  
(Fernández, 2007)

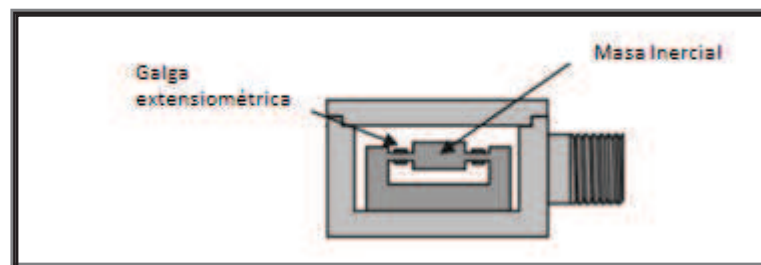
### **2.13.1 Aplicaciones comunes.**

- En los automóviles en la activación del airbag, sistemas de ABS...
- Sistemas electrónicos como teléfonos móviles, camas digitales, reproductores multimedia, consolas de juegos ...
- En el cabezal del discos duros en caso de caída para evitar daños y pérdida de datos.

### 2.13.2. Tipos de acelerómetros.

**Mecánicos.** Utilizan una masa inerte y resortes elásticos. Las variaciones se miden con galgas extensiométricos y posee además de elementos de amortiguación que impiden la oscilación.

Las variaciones al producirse una deformación en las galgas producen una corriente la cual es directamente proporcional a la aceleración.

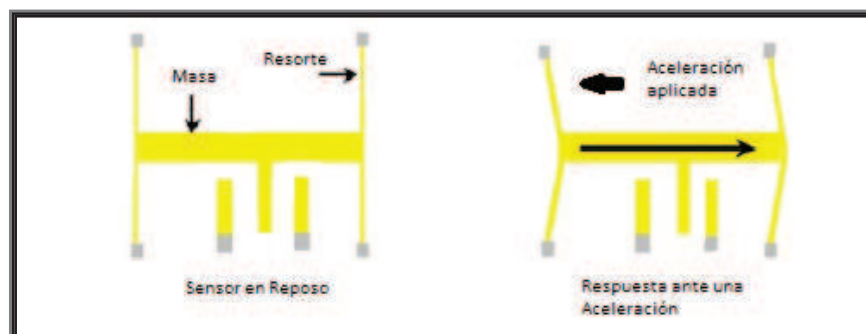


**Figura 26. Acelerómetro mecánico**

### Capacitivos.

Modifican la posición relativa de las placas de un micro-condensador cuando está sometido a la aceleración, lo cual causa variación su capacidad y genera un voltaje el cual es directamente proporcional a la aceleración.

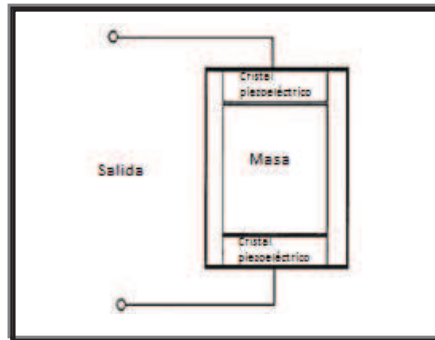
Posee un sensor micro-mecanizado similar a una "H" y un sistema de filamentos delgados con una masa central, los cuales cada uno actúa como una placa de un condensador variable de placas en paralelo.



**Figura 27. Acelerómetro capacitivo**

### Piezo-eléctricos.

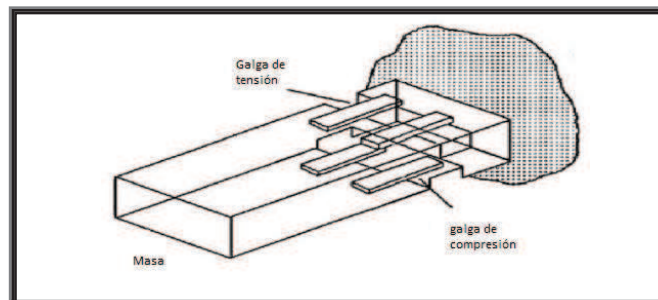
Se basa en cristales piezo-eléctricos los cuales cuando son sometidas alguna fuerza generan una corriente eléctrica, a causa de la variación de su estructura cristalina.



**Figura 28. Acelerómetro piezo-eléctrico**

### Piezo-resistivos.

Posee una masa y un sustrato, cuando existe una aceleración la masa ejerce una fuerza sobre el sustrato variando su resistencia, este es valor de resistencia interactúa en un puente de Whetstone el cual mide la intensidad de la corriente.



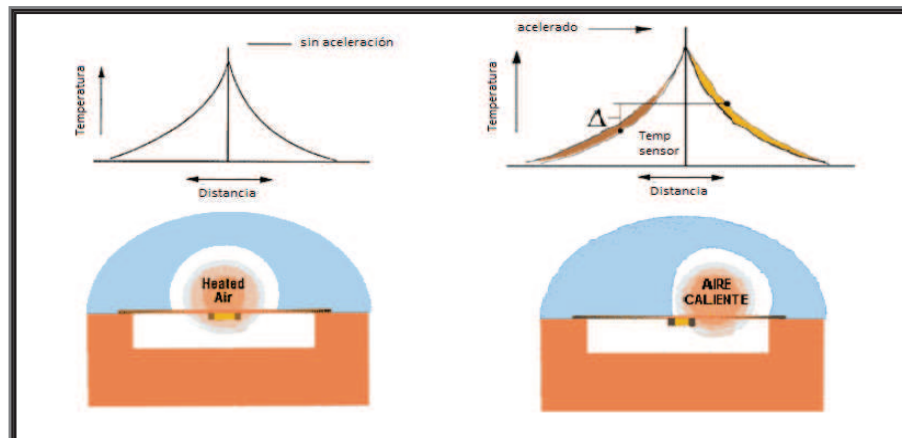
**Figura 29. Acelerómetro piezo-resistivo**

### Térmicos

Basa su funcionamiento en la convección termal, cuando una fuerza externa produce movimiento la burbuja de aire caliente se mueve de una forma análoga al mismo, esta burbuja se calienta por medio de una resistencia y dos termopares son



los encargados de producir un voltaje que es función de la diferencia de temperatura de una burbuja móvil de aire.



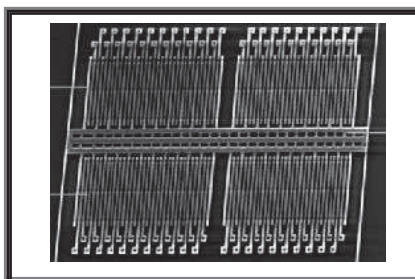
**Figura 30. Acelerómetro térmico**

### Tecnología MEMS

Los sistemas microelectromecánicos son elementos que funcionan en base a dispositivos electrónicos y mecánicos miniaturizados, capaces de trabajar a escala micrométrica son construidas a partir de silicio o vidrio.

Las primeras investigaciones con tecnología MEMS se realizaron en los años 60 pero no fue hasta los años 90 cuando se realizaron los primeros avances a lo que a comercialización y utilización de refiere. (Magni, 2009)

Los acelerómetros basados en tecnología tipo MEMS utilizan las mismas características de funcionamiento de los acelerómetros tipo térmico y tipo capacitivo explicado anteriormente.



**Figura 31. Acelerómetro capacitivo tipo MEMS**

### 2.13.3. Comparación entre tipos de acelerómetros

**Tabla 6. Características de los tipos de acelerómetros**

Tipo	Margen de medida (g)	Ancho de banda (Hz)	Ventajas e inconvenientes	Aplicaciones
Mems	1.5 a 250	0. a 1500	<ul style="list-style-type: none"> <li>Alta sensibilidad</li> <li>Coste medio</li> <li>Uso sencillo</li> <li>Bajas temperaturas</li> </ul>	<ul style="list-style-type: none"> <li>Impacto</li> <li>ABS</li> <li>Airbag</li> <li>Uso de automoción</li> </ul>
Piezoeléctrico	0 a 2000	10 a 20000	<ul style="list-style-type: none"> <li>Sensibilidad media</li> <li>Uso complejo</li> <li>Bajas temperaturas</li> <li>No funcionan en continua</li> </ul>	<ul style="list-style-type: none"> <li>Vibración</li> <li>Impacto</li> <li>Uso industrial</li> </ul>
Piezoresistivos	0 a 2000	0 a 10000	<ul style="list-style-type: none"> <li>Respuesta en continua y alterna</li> <li>Prestaciones medias</li> <li>Bajo coste</li> <li>Tamaño y peso mínimo.</li> <li>Alta sensibilidad</li> </ul>	<ul style="list-style-type: none"> <li>Vibración</li> <li>Impacto</li> <li>Automoción</li> <li>Biodinámica</li> <li>Ensayos de vuelo</li> <li>Test en túnel de viento</li> </ul>
Capacitivos	0 a 1000	0 a 2000	<ul style="list-style-type: none"> <li>Funcionan en continua</li> <li>Bajo ruido</li> <li>Baja potencia</li> <li>Bajo coste</li> </ul>	<ul style="list-style-type: none"> <li>Uso general</li> <li>Uso industrial</li> <li>Sistemas de alarma y seguridad</li> <li>Mediciones sísmicas</li> </ul>
Mecánicos	0 a 200	0 a 1000	<ul style="list-style-type: none"> <li>Alta precisión en continua</li> <li>Lentos</li> <li>Altos coste</li> </ul>	<ul style="list-style-type: none"> <li>Navegación inercial</li> <li>Guía de misiles</li> <li>Herramientas</li> <li>Nivelación</li> </ul>

## 2.14. Dispositivo android Sony Xperia Miro

Para el presente proyecto se utilizo un dispositivo móvil de la marca Sony Xperia, en la tabla 7 se presenta sus principales características. (Sony, 2013)



**Figura 32. Sony Xperia Miro**

**Tabla 7. Características generales Sony Xperia Miro**

<b>Marca</b>	SONY
<b>Modelo</b>	Xperia Miro ST23
<b>Dimensiones (ancho x alto x espesor)</b>	59,4 x 133 x 9,9 milímetros
<b>Peso</b>	100 gramos
<b>Sistema operativo</b>	Android 4.0.3 Ice Cream Sandwich ICS
<b>Procesador</b>	800 MHz
<b>Memoria RAM</b>	512 MB
<b>Resolución de pantalla</b>	320 x 480 Mega pixeles
<b>Wi-Fi</b>	802.11 b/g/n
<b>Bluetooth</b>	Versión 2.1 con A2DP
<b>Java</b>	Compatible
<b>Sensores</b>	Acelerómetro, proximidad y burbuja

Como se indico anteriormente el acelerómetro es utilizado para medir la inclinación, la vibración y el movimiento. Actualmente es posible construir acelerómetros de tres ejes (X,Y,Z) en un sólo chip de silicio, siendo la tecnología tipo MEMS capacitivos los más utilizados en los actuales teléfonos móviles.

### **2.15. NetBeans**

Es un software libre de programación tipo java, es una herramienta libre que permite escribir, compilar, depurar y ejecutar el código realizado en la aplicación que se está desarrollando.



Para mayor facilidad todas las aplicaciones que sean desarrolladas en esta herramienta puedan ser extendidas por otros programadores, ya que permite que las aplicaciones sean desarrolladas a través de paquetes; entendiéndose por paquete al archivo en Java que contiene las clases de Java.

La herramienta NetBeans proporciona las siguientes características:

- Administración de las interfaces de usuario.
- Administración de las configuraciones del usuario.
- Administración del almacenamiento.
- Administración de ventanas.
- Los paquetes que componen la plataforma NetBeans pueden ser descargados individualmente.

En la siguiente tabla se muestra la comparación (FindTheBest, 2013) de las principales características de Netbeans con otros sistemas.

**Tabla 8. Comparativo entre plataformas de desarrollo Java**

			
	IDE	CLASSIC	XCODE
<b>Uso destinado a</b>	<ul style="list-style-type: none"> <li>✓ Desarrollo de aplicaciones</li> <li>✓ Aplicaciones móviles</li> <li>✓ Aplicaciones online</li> </ul>	<ul style="list-style-type: none"> <li>✓ Desarrollo de aplicaciones</li> <li>✓ Aplicaciones móviles</li> <li>✓ Aplicaciones online</li> </ul>	<ul style="list-style-type: none"> <li>✓ Desarrollo de aplicaciones</li> <li>✓ Aplicaciones móviles</li> <li>✓ Aplicaciones online</li> </ul>
<b>Desarrollado para</b>	<ul style="list-style-type: none"> <li>✓ Android</li> <li>✓ iPhone o iOS</li> <li>✓ Linux</li> <li>✓ Mac</li> <li>✓ Windows</li> </ul>	<ul style="list-style-type: none"> <li>✓ Android</li> <li>✓ iPhone o iOS</li> <li>✓ Linux</li> <li>✓ Mac</li> <li>✓ Windows</li> </ul>	<ul style="list-style-type: none"> <li>✗ Android</li> <li>✓ iPhone o iOS</li> <li>✗ Linux</li> <li>✓ Mac</li> <li>✗ Windows</li> </ul>
<b>Licencia</b>	Libre / código abierto	Libre / código abierto	Propietaria
<b>Lenguajes compatibles</b>	<ul style="list-style-type: none"> <li>✓ C y C++</li> <li>✓ Java</li> <li>✗ Objective C</li> <li>✓ PHP</li> </ul>	<ul style="list-style-type: none"> <li>✗ C y C++</li> <li>✓ Java</li> <li>✗ Objective C</li> <li>✗ PHP</li> </ul>	<ul style="list-style-type: none"> <li>✓ C y C++</li> <li>✓ Java</li> <li>✓ Objective C</li> <li>✗ PHP</li> </ul>
<b>Funcionalidades</b>	<ul style="list-style-type: none"> <li>✓ Compilación</li> <li>✓ Editor</li> <li>✓ Control de versión</li> <li>✓ Herramienta de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>✓ Compilación</li> <li>✓ Editor</li> <li>✓ Control de versión</li> <li>✗ Herramienta de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>✓ Compilación</li> <li>✓ Editor</li> <li>✓ Control de versión</li> <li>✓ Herramienta de pruebas</li> </ul>

### Aspectos importantes para elegir NetBeans

Este software proporciona varias ventajas entre las más destacadas tenemos:

- Puede ser usada para cualquier tipo de aplicación.
- Permite la reutilización de paquetes.
- Tanto la instalación como su actualización son fáciles de realizarlas.
- Ejecución de aplicaciones en tiempo real.
- Entre la más sobresaliente es un software libre.

- Permite la programación en otros tipos de lenguaje, como por ejemplo permite la programación en C++.

Por lo expuesto anteriormente para el presente proyecto se escogió NetBeans la versión 7.3 ya que además de ser un software libre, ofrece grandes mejoras que las versiones anteriores ya que tiene la posibilidad de programar interfaces de modo visual mediante Swings.

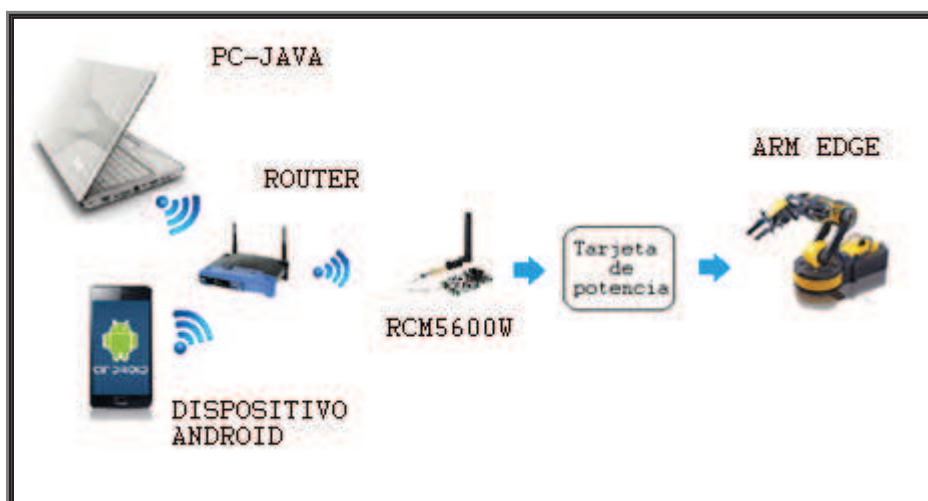
### CAPÍTULO III

#### INTERFACES DE COMUNICACIÓN Y SOFTWARE.

Existe varios tipos de software e interfaces para realizar la comunicación entre los tres dispositivos, cabe destacar que en el presente trabajo se utiliza al dispositivo móvil como router, ya que posee una aplicación propia del teléfono que trabaja en el estándar 802.11b facilitando el uso del dispositivo móvil como punto central de comunicaciones, así de esta manera podemos realizar la comunicación de datos de los tres dispositivos en red.

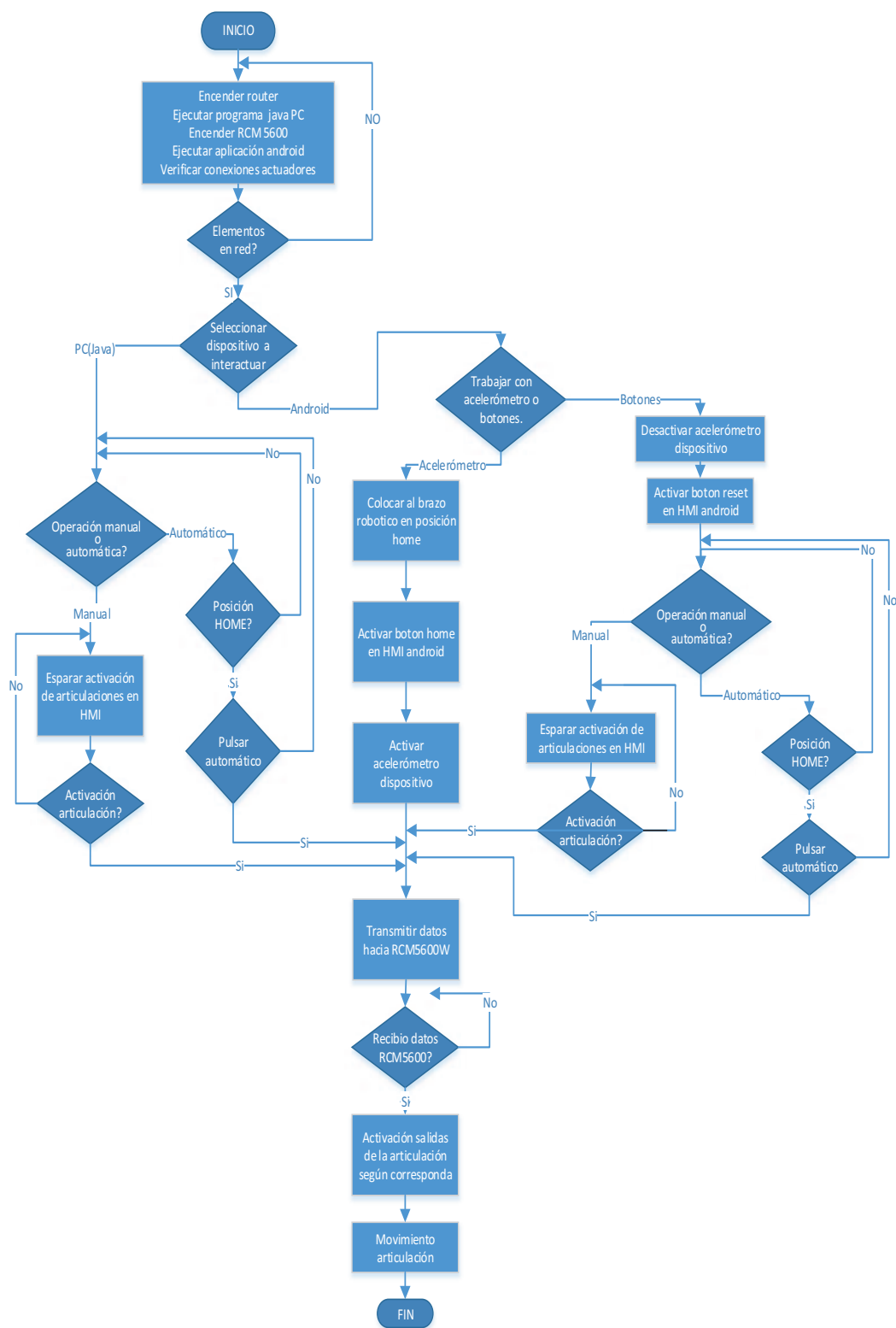


**Figura 33. Dispositivo móvil actuando como Access point.**



**Figura 34. Esquema general de comunicaciones**

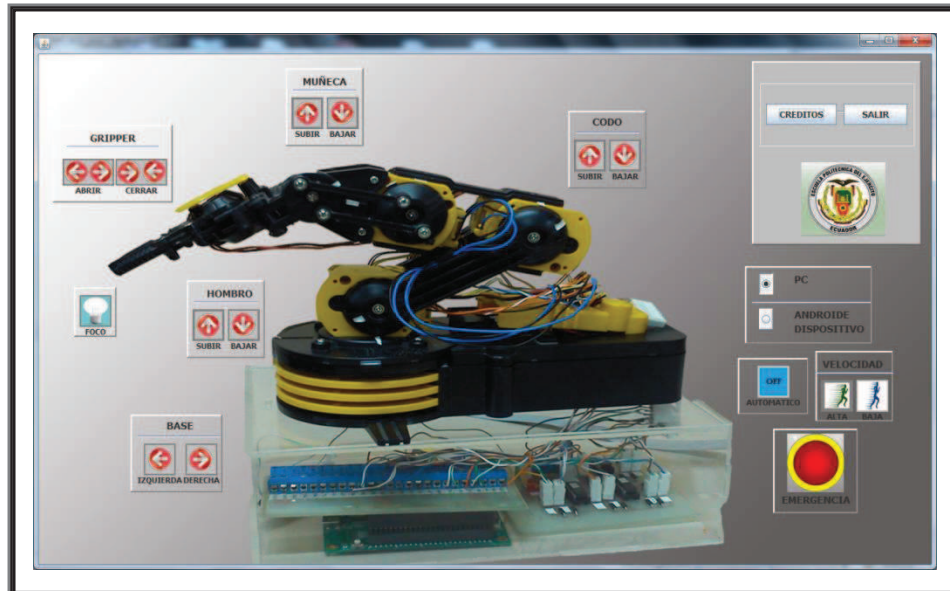
**Diagrama de flujo funcional del sistema.**



**Figura 35. Diagrama funcional del sistema**



### 3.1. Descripción del HMI desarrollado en Java.



**Figura 36. Diseño del HMI en Java**

El HMI es la interacción entre el humano y la maquina, por lo que hay que tomar siempre en cuenta factores como son la habilidades físicas, cognitivas, y comportamiento del usuario ante el HMI, por dichas razones se diseño un HMI interactivo y fácil de usar.

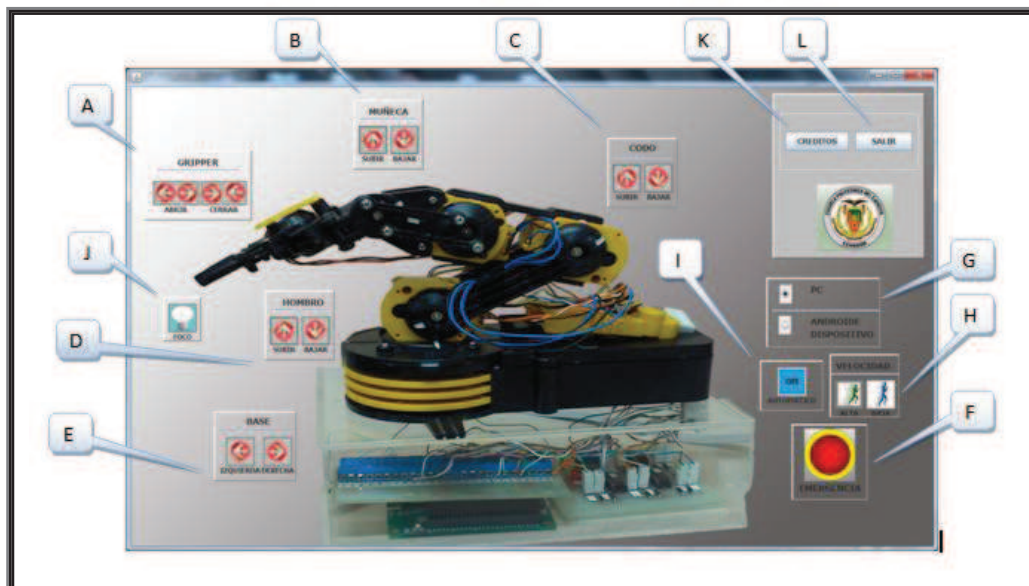
Java es un lenguaje de programación multiplataforma orientada a objetos, los cuales pueden encapsular diferentes funciones o métodos, e implementar aplicaciones cada vez más complejos y basados en red.

Un paquete ayudan a agrupar funciones y clases, en el presente proyecto los paquetes principales a importar son:

**java.net.\*** Permite realizar conexiones e intercambios de información a través de la red.

**java.awt.\*** Contiene clases para usadas crear interfaces gráficas de usuario.

**javax.swing.\*** Crea interfaces gráficas, a diferencia de AWT esta es independiente de la plataforma, en el presente proyecto se usa para insertar gráficos y hacer interactivo el sistema.



**Figura 37. Descripción de cada funcionalidad del HMI Java**

**Tabla 9. Detalle de cada funcionalidad del HMI en Java**

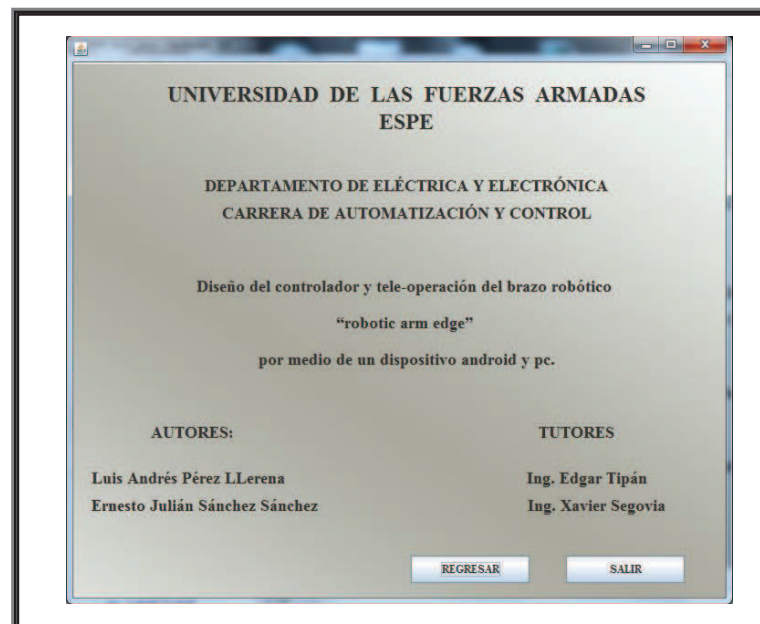
LETRA	DESCRIPCIÓN
A	Abrir o cerrar el gripper.- cada botón posee dos estados On y Off.
B	Subir o bajar la muñeca.- cada botón posee dos estados On y Off.
C	Subir o bajar el codo.- cada botón posee dos estados On y Off.
D	Subir o bajar el hombro.- cada botón posee dos estados On y Off.
E	Derecha o izquierda la base del robot.- cada botón posee dos estados On y Off.
F	Botón de emergencia.- tiene dos estados On y Off, en su estado On todas las demás funciones retornan a tu estado Off y son deshabilitados.
G	Selección del dispositivo con en que se va a interactuar ya sea desde la PC o el dispositivo android.
H	Control de velocidad; alta y baja.
I	Automático
J	Foco del gripper.
K	Créditos.- Carátula y autoría del proyecto.
L	Salida del sistema.

Los botones para la activación de cada articulación son Jtoggle Buttons, botones de dos estados ya que siempre se encuentran en 1 lógico o 0 lógico, alternados con un solo clic.

Para obtener el efecto de pulsación, modificamos la imagen de fondo del botón mediante la función “setIcon”, el cual usa un objeto creado por la función “ImageIcon”, esta última carga una imagen desde una dirección específica.

Para seleccionar que dispositivo va a trabajar sea PC o dispositivo android se usa Radio Button, asociado a un Button Group en cual ayuda a que el usuario no tenga la posibilidad de seleccionar los dos Radio Button y trabajen al mismo tiempo.

En la opción de créditos posee una pequeña pantalla el cual contiene datos generales del proyecto, como se muestra en la figura 38.



**Figura 38. HMI créditos en Java**

### 3.1.1. Flujograma programa JAVA

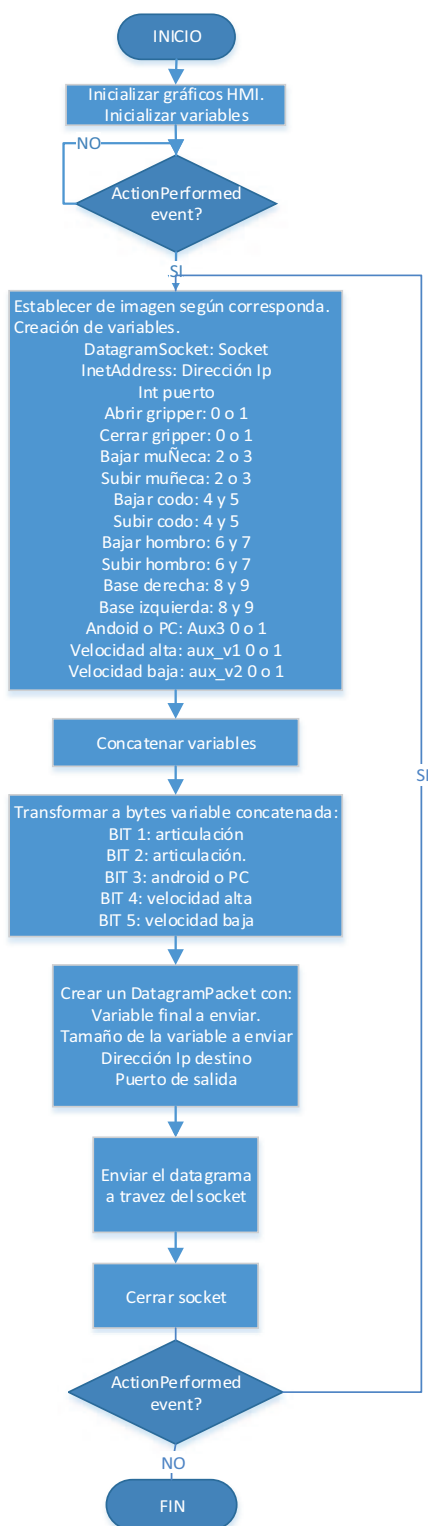


Figura 39. Diagrama de flujo programación general Java

## 3.2. Descripción del HMI del dispositivo móvil

### 3.2.1. Diseño interface TouchOSC.

Para el diseño de la interfaz en el dispositivo móvil android se lo realizo a través del IDE gráfico del TOUCH OSC EDITOR, el cual es proporcionado de forma gratuita por el desarrollador del software hexler.net, cabe recalcar que este software fue creado inicialmente para aplicaciones de control de sonido, como se observa en el presente trabajo se lo puede utilizar para diferentes aplicaciones en android que requieran el envío de datos bajo UDP.

Hay que tener en cuenta que el editor del HMI para el dispositivo móvil android está desarrollado en java que además de su versatilidad es de fácil manejo.

Una vez obtenido la aplicación, al momento de ejecutarlo se obtiene la figura 41:



Figura 40. Ícono TouchOSC Editor.

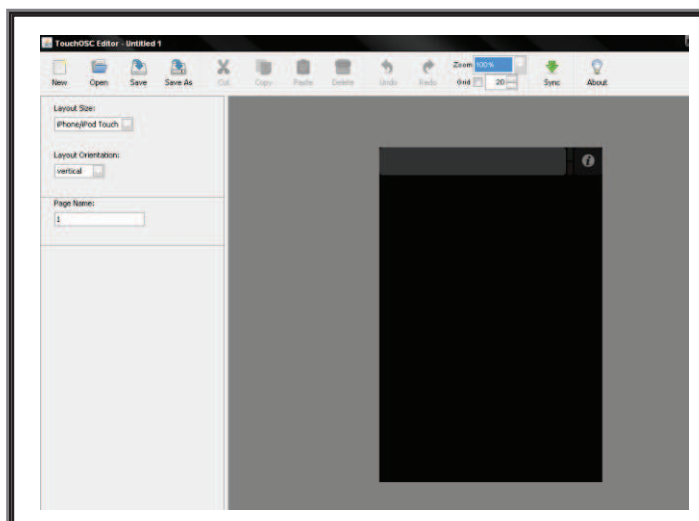
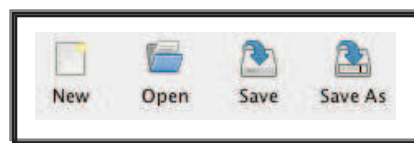


Figura 41. TouchOSC Editor.

El editor del HMI para el dispositivo móvil android está compuesto de tres partes fundamentales, el primero es la barra de herramientas con el cual se puede crear nuevas ventanas y guardarlas, el segundo es el panel de propiedades e información de los objetos que se van a insertar en el diseño del HMI y por último la ventana de edición y vista previa del diseño desarrollado, las cuales van a ser detalladas a continuación:

Primero se inicia con la barra de herramientas, la cual posee iconos básicos de edición de una ventana HMI:

A través de estos iconos se puede crear un nuevo diseño, abrir uno existente y guardar el diseño.



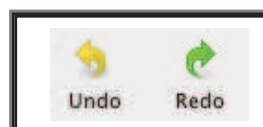
**Figura 42. Nuevo, Abrir y Guardar.**

En la figura 43 se pueden encontrar los iconos de cortar, copiar, pegar y borrar según se va desarrollando el diseño.



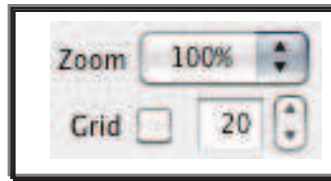
**Figura 43. Cortar, Copiar, Pegar y Borrar.**

En la siguiente figura 44 se tiene los iconos de deshacer y rehacer el cambio realizado en el diseño.



**Figura 44. Deshacer y Rehacer.**

En la siguiente figura se encuentra los iconos para el ajuste de zoom y de ser necesario para un mejor diseño el uso de una cuadrícula.



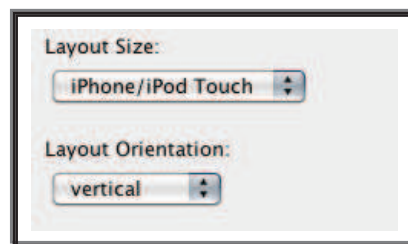
**Figura 45. Zoom y Cuadrícula.**

Y por ultimo en la barra de herramientas se dispone de los iconos de sincronización del dispositivo a cargar el HMI diseñado y el icono de información acerca de la versión.



**Figura 46. Sincronización y Acerca de.**

El siguiente enfoque es en panel de propiedades, en el cual se puede ajustar el tamaño de la ventana según el dispositivo móvil en el cual se vaya a cargar el HMI diseñado, este dispone dos opciones ya sea para cargarlo en un iPhone/ipod o un ipad y también el panel de propiedades dispone del sentido de nuestra ventana en el dispositivo móvil; ya sea vertical u horizontal:



**Figura 47. Panel de propiedades.**

A continuación se detalla la configuración de las propiedades de los botones que se va a utilizar en el HMI.



**Figura 48. Configuración propiedades HMI**

**Tabla 10. Descripción propiedades TouchOSC**

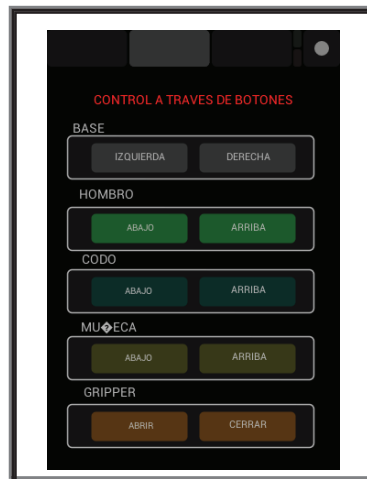
A	Nombre del botón
B	Nombre de la ventana

Teniendo en cuenta todos los iconos y propiedades que dispone este editor, el HMI que se ha realizado para el control del manipulador robótico es el siguiente:



**Figura 49. HMI del dispositivo móvil android-Ventana 1.**





**Figura 50. HMI del dispositivo móvil android-Ventana 2.**



**Figura 51. HMI del dispositivo móvil android-Ventana 3.**

Una vez terminada el diseño del HMI para el control del brazo robótico lo que resta es transferir el archivo al dispositivo móvil.

### **3.3. Dynamic C.**

#### **3.3.1. Características Generales.**

Es el software usado para programar en el Microcore RCM5600W las diferentes órdenes, recibidas inalámbricamente del PC o dispositivo móvil y transformarlas en sentencias binarias que habilitarán a los actuadores del robot, siendo una programación orientada tanto para comunicación como para control.



**Figura 52. Ícono Dynamic C**

Dinamic C es un entorno de desarrollo que utiliza Rabbit para implementar sus productos, soporta multitarea, y posee una gran variedad de bibliotecas las cuales permiten resolver aplicaciones típicas, corre en plataforma Windows.

Se puede programar, compilar y ejecutar tanto en lenguaje C como en assembler, integra las siguientes acciones de desarrollo.

- Edición de la programación.
- Compilación de la programación.
- Enlace con los dispositivos externos
- Carga de los programas en la memoria seleccionada
- Depuración de los programas compilados.

#### **Funciones incorporadas en Dynamic C**

- Funciones estándar de C
- Interface serial 232
- Interface I2C, GPS, SPI
- RTC ( Real Time Clock)

#### **Protocolos complementarios de comunicación.**

- Acceso por socket a nivel UDP y TCP
- Cliente DHCP, POP3, SMTP
- Servidor HTTP
- Cliente/Servidor TFTP y FTP

### 3.3.2. Funciones Dynamic C.

Antes de iniciar la programación se debe configurar los parámetros de comunicación, compilación, depuración del programa, además de definir las características que va a tener la tarjeta RCM5600 en la red. En el entorno de navegación de Dinamic C, ir a Options y Project Options.

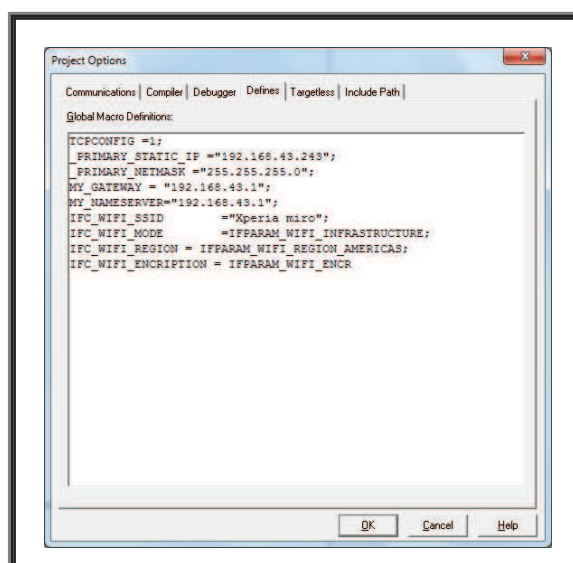


Figura 53. Ventana de configuración Dinamic C.

#### TCPCONFIG:

Tabla 11. Opciones de configuración TCP

1	Usa una configuración de red estática en una red Ethernet.
2	Crea una comunicación serial por el puerto C usando un protocolo punto a punto.
3	Establece una configuración de red dinámica utilizando el protocolo DHCP.
4	Permite usar la configuración 1 y la 2 en el puerto serial C.
5	Utiliza DHCP con la excepción de que el usuario administre los parámetros de conexión al servidor DHCP

Se utiliza la opción 1, ya que permite controlar y administrar las direcciones IP.

**Tabla 12. Parámetros de red y definiciones usadas en Dinamic C**

NOMBRE	DATO	DESCRIPCION
_PRIMARY_STATIC_IP:	192.168.43.243	Dirección IP del módulo RCM5600.
_PRIMARY_NETMASK:	255.255.255.0	Mascara de red
MY_GATEWAY:	192.168.43.1	Es la puerta de enlace utilizada para interconectar redes.
MY_NAMESERVER:	192.168.43.1	Nombre del servidor a conectar.
IFC_WIFI_SSID:	Xperia miro	Nombre de la red inalámbrica a acceder.
IFC_WIFI_MODE:	IFPARAM_WIFI_INFRASTRUCTURE	Modo de funcionamiento wireless; Infraestructura o Adhoc.
IFC_WIFI_REGION:	IFPARAM_WIFI_REGION_AMERICAS	Limita el máximo de potencia de acuerdo a región seleccionada y estable un rango de canales.
IFC_WIFI_ENCRPTION:	IFPARAM_WIFI_ENCR_NONE	Controla el tipo de encriptación o cifrado a utilizar.
<b>Librerías y definiciones</b>		
#define MAX_UDP_SOCKET_BUFFERS:	2	Determina el número máximo sockets UDP.
#define LOCAL_PORT:	5550 Puerto Dispositivo Android 5551 Puerto PC	Puerto de entrada por donde vamos a recibir datos de clientes.
#define REMOTE_UDP_MASK:	255.255.255.0	Mascara de red para la conexión UDP.
#define REMOTE_IP:	192.168.43.1 Dirección Dispositivo Android 192.168.43.242 Dirección PC	Dirección IP remoto del dispositivo a conectar.
#define "rcm 56xxw.lib"		Incluimos las librerías utilizados en el microprocesador 56xx de Rabbit.
#mmap xmem:		Utiliza un espacio de memoria xmem.
#use "dcrtcp.lib":		Incluye las librerías principales de gestión de red de TCP/IP.

Una vez revisado los parámetros y librerías necesarias para el desarrollo del proyecto, a continuación se presentan y explican detalladamente las funciones aplicadas en Dynamic C:

**sock\_init():**

Es un parámetro indispensable ya que inicializa la configuración TCP, ARP, UDP o DNS según corresponda.

Si el proceso de inicialización y conexión con la red fue la correcta, esta función retorna un valor igual a cero caso contrario un valor distinto a cero.

**memset(received\_data, 0, sizeof(received\_data)):**

Se utiliza para inicializar un bloque de memoria en este caso será para inicializar los buffers tanto para los datos de Java como para Android.

Esta función posee 3 argumentos; el primero es el dato que se desea cambiar, el segundo es el valor con el cual se inicializara, y el tercero es el tamaño en bytes que ocupa la variable.

**udp\_open(&my\_socket, LOCAL\_PORT, resolve(REMOTE\_IP), 0, NULL):**

Esta función se utiliza para abrir la conexión UDP.

El primer parámetro para empezar la comunicación es abrir socket asociado a un puerto específico, este puerto se determina en el segundo parámetro.

En el tercer campo es la dirección IP del host remoto, si se especifica la IP este solo aceptará datagramas UDP del host que tenga dicha IP. Si el campo es llenado con 0; el primer host que envíe un datagrama al puerto asociado será en que complete la conexión, y por ultimo si es -1 todos los host remotos pueden enviar datagramas al socket.

El cuarto parámetro es el puerto de recepción de datos del host remoto y el quinto es la función manejador, el cual sirve para a llamar una función cuando se reciben datos. Al colocar NULL este envía los datos al buffer de recepción del socket.

**udp\_recv(&my\_socket, received\_data, sizeof(received\_data)):**

Recibe un datagrama UDP en un socket UDP. Si el búfer no es lo suficientemente grande para el datagrama, el datagrama se trunca y el resto se elimina

Posee tres parámetros, el primero es el socket, el segundo la variable donde se almacenaran los datos de buffer, y el tercero el tamaño máximo que tendrá el buffer.

**BitWrPortI(int io\_port, char \*PORTShadow, int value, int bitcode):**

Esta función permite realizar la escritura de las salidas digitales de los diferentes puertos existentes en la tarjeta ya que la interfaz proporcionada por esta tarjeta es puramente digital, hay que tener en cuenta que todos los registros internos proporcionados por la tarjeta se la realiza mediante macros; las cuales deben definirse al inicio de nuestro programa.

Esta función posee cuatro parámetros, el primero es la dirección del puerto interno al que se desea acceder, el segundo es la dirección del puerto especificado como tipo char, el tercer parámetro es el valor a escribir en el puerto ya sea un cero o uno y el ultimo parámetro es del número del puerto a acceder teniendo en cuenta que para definir todo el puerto como salidas debemos colocar 0xFF.

Hay que tener en cuenta que al momento de declarar el tercer parámetro, para la utilización del puerto como escritura hay que definirlo con el valor de uno.

La figura 54. detalla todas las entradas y salidas aplicadas en el programa de la tarjeta RCM5600W.

```

BitWrPortI(PDDDR, &PDDDRShadow, 1, 0); //gripper abrir
BitWrPortI(PDDDR, &PDDDRShadow, 1, 1); //gripper cerrar
BitWrPortI(PDDDR, &PDDDRShadow, 1, 2); //muneca abajo
BitWrPortI(PDDDR, &PDDDRShadow, 1, 3); //muneca arriba
BitWrPortI(PEDDR, &PEDDRShadow, 1, 1); //codo abajo
BitWrPortI(PEDDR, &PEDDRShadow, 1, 2); //codo arriba
BitWrPortI(PEDDR, &PEDDRShadow, 1, 3); //hombro abajo
BitWrPortI(PEDDR, &PEDDRShadow, 1, 5); //hombro arriba
BitWrPortI(PEDDR, &PEDDRShadow, 1, 6); //base derecha
BitWrPortI(PEDDR, &PEDDRShadow, 1, 7); //base izquierda
BitWrPortI(PCDDR, &PCDDRShadow, 1, 1); //FOCO
BitWrPortI(PBDDR, &PBDDRShadow, 0, 2); //Entrada digital
BitWrPortI(PBDDR, &PBDDRShadow, 0, 3); //Entrada digital
BitWrPortI(PBDDR, &PBDDRShadow, 0, 4); //Entrada digital
BitWrPortI(PBDDR, &PBDDRShadow, 0, 5); //Entrada digital
BitWrPortI(PBDDR, &PBDDRShadow, 0, 6); //Velocidad baja
BitWrPortI(PBDDR, &PBDDRShadow, 0, 7); //Velocidad alta

```

**Figura 54. Declaración de entradas y salidas de Dinamic C.**

#### **BitRdPortI(int io\_port, int bitnumber):**

Esta instrucción permite la lectura del puerto, al igual que la función de escritura está definida por macros y retorna un valor ya sea un uno o cero.

Esta función posee dos parámetros el primero el puerto a leer y el segundo es el número del puerto.

#### **pwm\_set(int canal, int ciclo de trabajo, int opciones).**

Crea un canal PWM para un ciclo de trabajo especificado el cual varía entre 0 y 1024, donde 0 es el estado bajo de la señal pwm 0% y 1024 el 100% del ciclo de trabajo de la señal pwm.

Posee 3 parámetros; el primero es el canal a usar como se detallada en la tabla 13, el segundo el ciclo e trabajo de 0 a 1024 y por último opciones para incluir frecuencias de trabajo.

**Tabla 13. Canales PWM**

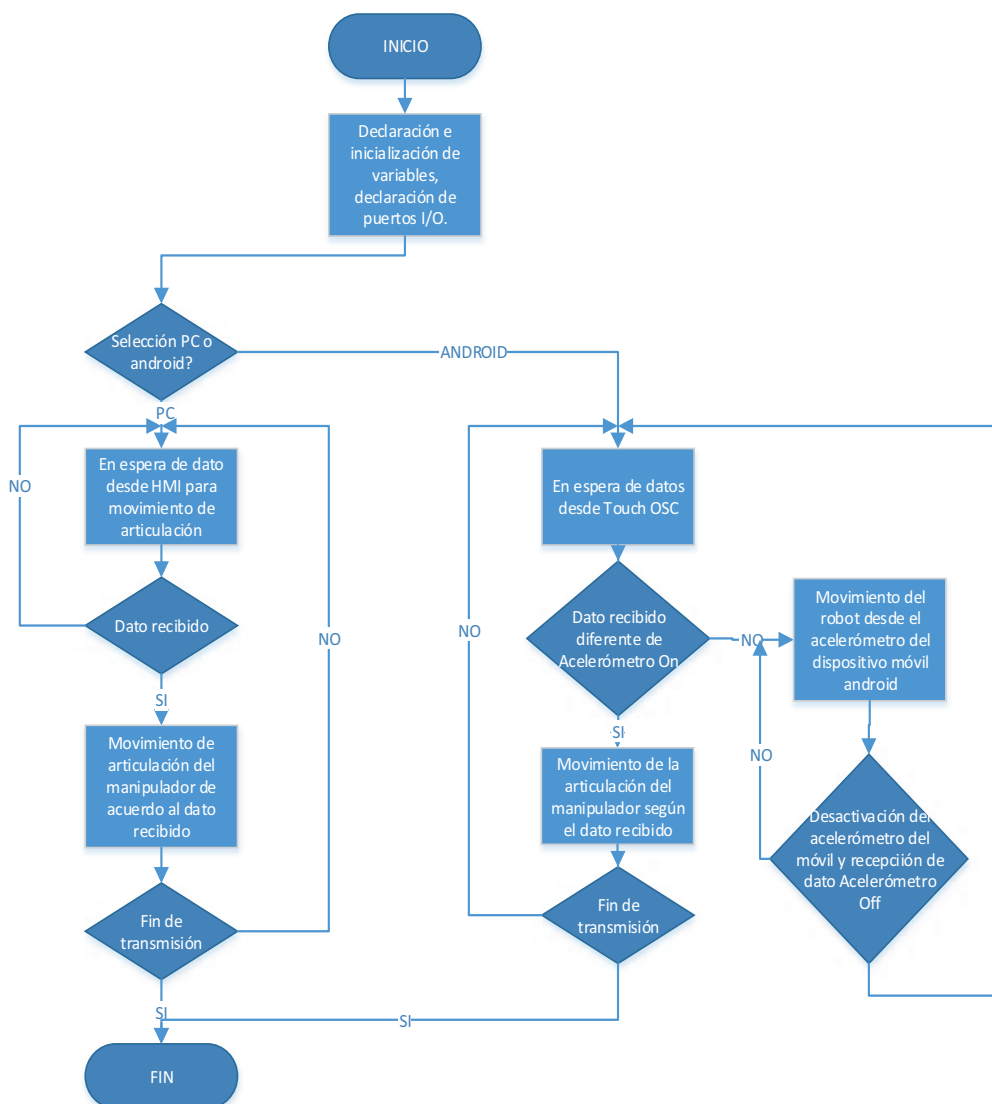
PWM	Pines de salida
Canal 0	PC4, PD4, PE4, PH4
Canal 1	PC5, PD5, PE5, PH5
Canal 2	PC6, PD6, PE6, PH6
Canal 3	PC7, PD7, PE7, PH7

**tcp\_tick ():**

Método utilizado para determinar el estado de la conexión del socket. Monitorizando la comunicación, en el caso de producirse algún error crea un código indicando que la comunicación se perdió.



### 3.3.3. Diagrama de flujo del programa desarrollado en Dynamic C



**Figura 55. Diagrama de flujo programación general Dynamic C**

### 3.4. Comunicación CPU y MiniCore RCM5600W

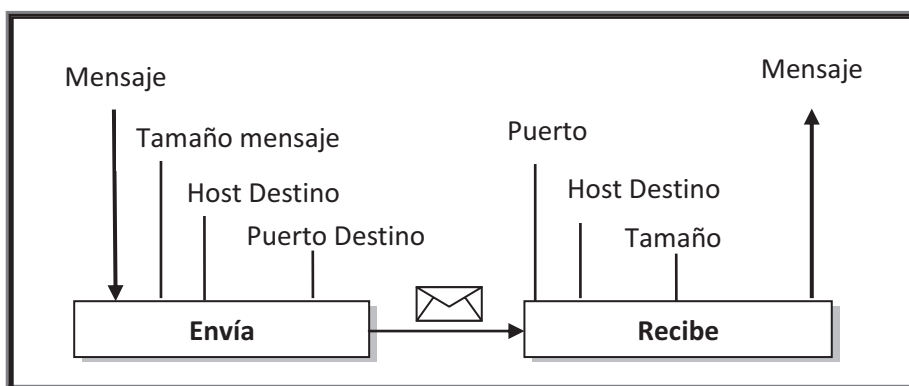
En el presente proyecto en el computador se ha desarrollado un software de soporte diseñado en Java, ya que Java posee un gran conjunto de clases y funciones para un utilización en aplicaciones de red, los cuales permiten abrir sockets, enlazar y aprobar conexiones con clientes y servidores remotos.

La comunicación se lo realiza mediante el protocolo UDP, el cual es un protocolo a nivel transporte no es orientado a la conexión, esto quiere decir que no se garantiza la integridad y llegada de los datos al destino, siendo un protocolo más rápido y ligero que TCP.

### Establecimiento de la comunicación

Java provee dos clases especiales ubicadas en el paquete java.net para el uso de UDP.

- DatagramSocket.- utilizada para la comunicación, es un array de bytes donde se almacena la información que llega.
- DatagramPacket.- contiene los datos y direcciones de red.



**Figura 56. Comunicación entre sockets UDP**

A continuación se detalla los pasos para el establecimiento de la comunicación.

1. El programa cliente crea una instancia de tipo datagramSocket. Ejemplo.

*“DatagramSocket socketCliente=new DatagramSocket()”*

2. El programa cliente crea una instancia de tipo `DatagramPacket`, proporcionándole los datos, tamaño en bytes de los datos a enviar, además de la dirección y el puerto destino. Ejemplo.

```
“DatagramPacket Envio=new DatagramPacket  
(datos,datos.length,DireccionIP,puerto)”
```

3. Java envía el datagrama usando la función `send`; utilizando el socket y el paquete de datos creados. Ejemplo.

```
“socketCliente.send(envio) ”
```

4. En el programa servidor creado en el RCM5600W crea una instancia especificando la IP y puerto asociado mediante función `“udp_open”`.

5. El programa servidor en el RCM5600W invoca un método, que lee continuamente el socket en espera de un dato en el buffer.

6. Utilizando `“udp_recv”` en el RCM5600W guarda los datos del buffer en una variable.

7. El programa en el RCM5600W compara los datos recibidos desde el cliente y activa sus salidas digitales según corresponda, accionando los actuadores del brazo robótico.

### **3.5. Modo Infraestructura MiniCore RCM5600W.**

En este modo cada punto existente en la red se conecta a un punto de acceso a través de un enlace inalámbrico. La tarjeta RCM5600W tiene la posibilidad de trabajar en modo infraestructura, lo cual es posible utilizando un router siendo este el punto de acceso que permite la comunicación entre los diferentes dispositivos existentes en la red, en este caso los elementos existentes en la red son: la tarjeta

RCM5600W, el dispositivo móvil android y la PC con la interfaz gráfica realizada en java. Figura 57.



**Figura 57. Comunicación inalámbrica en modo infraestructura**

Hay que tomar en cuenta varias características que presenta este modo para su implementación, entre estos tenemos:

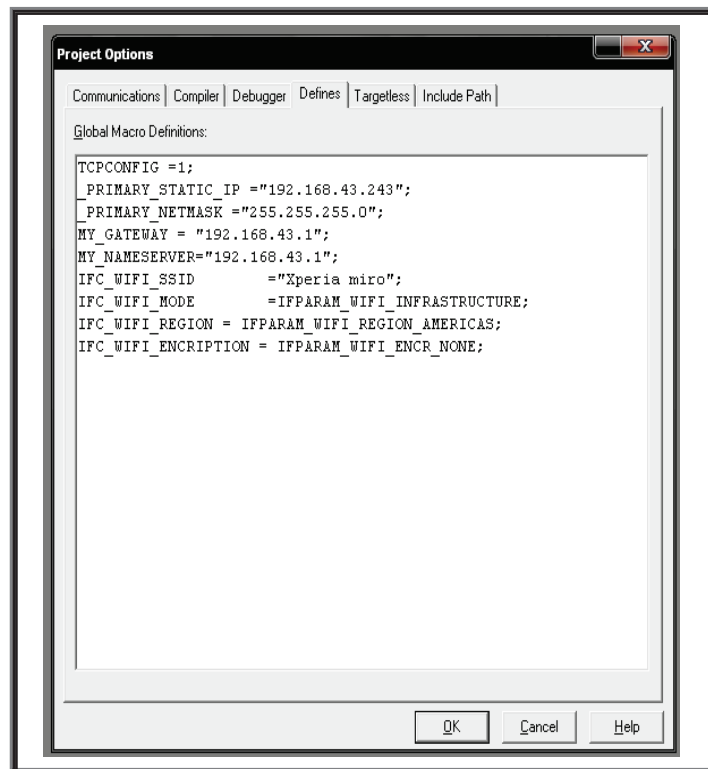
- Está basado en el estándar 802.11b/g/n.
- Posee opciones de configuración IP.
- Se puede incluir varios puntos de acceso dentro de la red.

- Al tener un aumento de dispositivos dentro de la red, el rendimiento de la comunicación disminuye.

En la configuración de la tarjeta RCM5600W hay que verificar tres aspectos importantes para realizar la comunicación de datos en este modo:

- Características de la red a la cual se va a realizar la conexión.
- Las direcciones IP involucradas en la red.
- El canal que se va a utilizar para el intercambio de datos.

A continuación se presenta la configuración que se realizó en la tarjeta RCM5600W para trabajar en este modo, teniendo en cuenta que debe cumplir con los parámetros mencionados anteriormente:



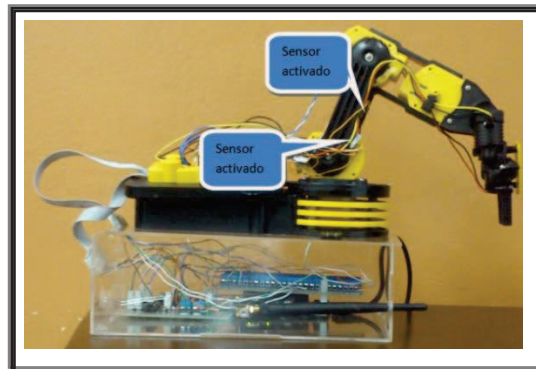
**Figura 58. Configuración tarjeta RCM5600W modo infraestructura.**

### 3.6. Modo Automático

Existe una gran variedad de aplicaciones en las que se puede emplear un brazo robótico, en el presente proyecto se ha optado por realizar una simulación de un proceso industrial en la cual, al manipulador robótico se lo utiliza para el traslado de material desde a una banda transportadora a un lugar específico, teniendo en cuenta que al utilizar un manipulador robótico esta actividad puede ser repetitiva.

Para el inicio del proceso se debe tomar en cuenta dos aspectos importantes expuestos a continuación:

- El robot se debe colocar en la posición inicial (HOME), el gripper debe encontrarse abierto para el agarre del material.
- Los sensores tanto del hombro como del codo debe estar activados para de esta forma asegurarse que el proceso se lo realice sin ningún problema.



**Figura 59. Posición HOME**

#### 3.6.1. Descripción del proceso

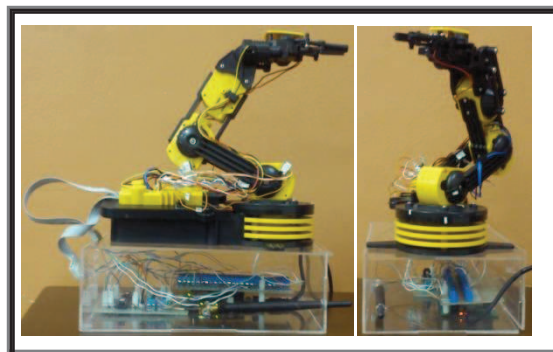
Establecido los parámetros anteriores para el desarrollo del proceso, se explica cada uno de los movimientos que realiza el manipulador para el traslado de material.

Inicia con el agarre del material, como se puede observar en la siguiente imagen donde el gripper se encuentra semi-cerrado.



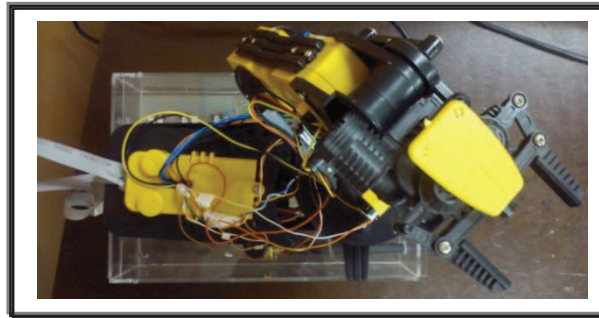
**Figura 60. Agarre de material**

Terminado de recoger el material el manipulador proceder al traslado del material hacia la banda transportadora.



**Figura 61. Traslado de material**

Una vez que el manipulador llega a la banda transportadora el material es depositado, en la siguiente imagen se puede observar que el gripper se encuentra abierto indicando que el material ya ha sido despachado, por lo que el brazo robótico regresa a su posición inicial.

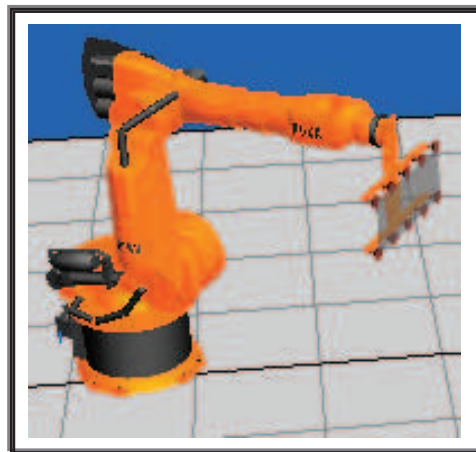


**Figura 62. Material depositado en la banda transportadora**

Este proceso va a ser repetitivo y solo con la orden del operario se lo va a detener.

### **3.6.2. Aplicación desarrollada en simulador**

Como se explico anteriormente el proceso que se está simulando es de traslado de material hacia una banda transportadora, para una mejor apreciación se lo ha realizado en el simulador “KUKA SIM PRO” para de esta forma apreciar que la aplicación desarrollada puede ser implementada en un proceso industrial utilizando un robot KUKA.



**Figura 63. Robot KUKA**

El gripper en este caso usa ventosas, para trasladar una placa metálica a su respectivo almacenamiento.



Como resultado final de la simulación industrial, se puede observar en la siguiente imagen, donde una vez realizado un corte de una placa metálica es dejado en la banda y después el otro manipulador recoge la placa para almacenarla, observando una de tantas aplicaciones que puede realizar un manipulador robótico. Observando la gran importancia que tiene presente proyecto ya que el control de brazo robótico se lo realiza exclusivamente en un medio inalámbrico y puede ser implementado a nivel industrial.



**Figura 64. Traslado y almacenamiento de piezas metálicas**

## CAPÍTULO IV

### IMPLEMENTACIÓN Y RESULTADOS DE LAS PRUEBAS

#### 4.1. Toma de datos actuadores

Para la selección del elemento electrónico a usar para el control de los actuadores, se realizó un estudio previo de la corriente máxima que consumen de los motores dc de cada articulación, efectuando mediciones en los estados en los cuales los motores trabajan en velocidad alta y velocidad baja, teniendo como resultado los siguientes valores.

**Tabla 14. Datos de corriente velocidad alta**

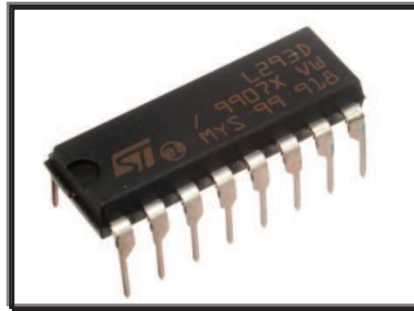
Motor Articulación	Movimiento descendente	Movimiento ascendente
Base	331 mA	330 mA
Codo	165 mA	365 mA
Hombro	240 mA	410 mA
Muñeca	190 mA	230 mA
Gripper	158 mA	161 mA

**Tabla 15. Datos de corriente velocidad baja**

Motor Articulación	Movimiento descendente	Movimiento ascendente
Base	290 mA	290 mA
Codo	175 mA	335 mA
Hombro	195 mA	330 mA
Muñeca	180 mA	215 mA
Gripper	160 mA	160 mA

Al tener una corriente máxima de 410 mA en la articulación del hombro cuando este se encuentra subiendo, establecemos que es la corriente máxima que debe soportar el sistema por lo cual se eligió el circuito integrado L293D que tiene una corriente máxima de 600 mA por canal.

#### 4.2. Driver de motor L293D.



**Figura 65. Driver L293D**

El puente H L293D es el driver que permite contralar los motores DC del brazo robótico en dos direcciones según corresponda, es capaz de proporcionar una salida de corriente de hasta 600 mA por canal atendiendo las necesidades de corriente de cada motor analizadas anteriormente, cada canal es controlado por señales de entrada compatibles con TTL, cada par de canales posee entradas de habilitación (enable), usados para controlar la velocidad de los motores, dichas señales son generadas por la tarjeta RCM5600 a través de salidas PWM con lo cual se controla dos tipos de velocidades alta y baja ya sea mediante el Pc o el dispositivo android.

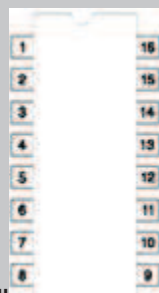
La señal EN1 habilita la pareja de canales 1 y 2, La señal EN2 habilita los canales 3 y 4. En las salidas OUT se conectan los motores dc, estas se activan mediante una combinación binaria en las entradas IN del circuito los cuales dan el sentido de movimiento del motor.

Posee un pin de alimentación de cargas y del circuito integrado.

En el arranque y parada de motores se genera una corriente inversa que pueden destruir un circuito, en el actual proyecto esta corriente es generada por los actuadores por lo que producían sobrecalentamiento en el integrado por tal razón se

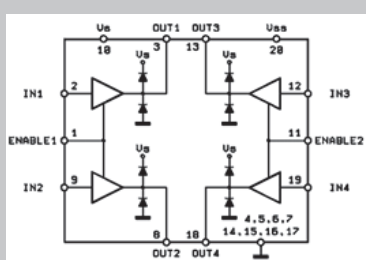
utilizo un circuito de protección constituida por diodos 1N4007 eliminando dicho problema. La tabla 16 detalla la descripción de pines del driver L293D.

**Tabla 16. Descripción pines L293D**

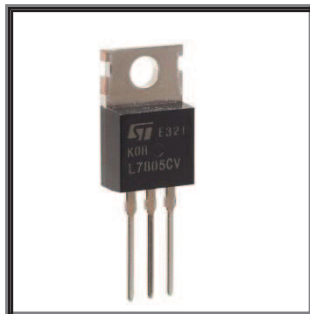
PIN	NOMBRE	DESCIPCION	DIAGRAMA
1	Chip Enable 1	Habilitación de los canales 1 y 2	 <p>Figura 66. Diagrama de pines L293D</p>
2	Input 1	Entrada del canal 1	
3	Output 1	Salida del canal 1	
4	GND	Tierra	
5	GND	Tierra	
6	Output 2	Salida del canal 2	
7	Input 2	Entrada del canal 2	
8	Vs	Alimentación de las cargas	
9	Chip Enable 2	Habilitación de los canales 3 y 4	
10	Input 3	Entrada del canal 3	
11	Output 3	Salida del canal 3	
12	GND	Tierra	
13	GND	Tierra	
14	Output 4	Salida del canal 4	
15	Input 4	Entrada del canal 4	
16	Vss	Alimentación del circuito integrado	

Los valores nominales máximos del driver L293D de detallan a continuación.

**Tabla 17. Características máximas de operación del driver L293D**

PIN	PARAMETRO	VALOR	DIAGRAMA
Vs	Voltaje de alimentación	36 V	 <p>Figura 67. Diagrama interno del driver L293D</p>
Vss	Voltaje de alimentación lógica	36 V	
Vi	Voltaje de entadas	7 V	
Ven	Voltaje de habilitación	7V	
Io	Corriente pico de salida	1.2 A	
Pto	Energía total de disipación T=90 t □ C	4 w	
T	Temperatura de trabajo	-40 – 150 □ C	

### 4.3. Regulador 7805



**Figura 68. Regulador 7805**

El regulador 7805 es un integrado de la familia de reguladores de tensión positiva, este componente tiene la característica de mantener una tensión fija de 5V a una intensidad máxima de 1A, dicha tensión es a la cual trabaja de la tarjeta RCM5600 siendo utilizada exclusivamente para dicho propósito.

Tiene tres terminales; pin de voltaje a regular, la cual recibe la tensión de las baterías que es 6 V, GND y el pin de voltaje de salida regulada que es 5V.

**Tabla 18. Descripción pines 7805**

PIN	NOMBRE	DESCIPCION	DIAGRAMA
1	Input	Entrada de voltaje (6v).	
2	GND	GND	
3	Output	Salida 5v.	

**Figura 69. Diagrama de pines del regulador 7805**

#### 4.4. PWM de la Tarjeta RCM5600

La modulación por ancho de pulso es una técnica la cual modifica el ciclo de trabajo de una señal periódica, como se explicó anteriormente el ciclo de trabajo que produce un canal PWM de la tarjeta RCM5600 varía entre 0 y 1024, donde 1024 es el 100% de la señal y 0 el 0% de la señal.

En el actual proyecto el 100% de la señal PWM, se utiliza para controlar la velocidad alta de todas las articulaciones.

Para la velocidad baja, se tomo en cuenta la inercia que presentaba cada articulación al movimiento, siendo la articulación del codo la más afectada, mediante el método de prueba y error se obtuvo los siguientes valores.

**Tabla 19. Ciclos de trabajo PWM y velocidad de las articulaciones**

ARTICULACIÓN	PWM - VELOCIDAD (rad/s)	PWM - VELOCIDAD (rad/s)	PWM - VELOCIDAD (rad/s)	PWM - VELOCIDAD (rad/s)
BASE	1024-0.224	850-0.3141	510-0.2055	<400-0
HOMBRO	1024-0.293	850-0.2244	510-0.287	<400-0
CODO	1024-0.3575	850-0.1625	510-0	<400-0
MUÑECA	1024-0.276	850-0.1745	510-0.201	<400-0
GRIPPER	1024-0.2425	850-0.1868	510-0.220	<400-0

Como se puede observar en la tabla anterior al tratar de poner una señal pwm de 400(39.1%) y menores, las articulaciones del robot se mantuvieron en inercia por lo que para realizar el control de velocidad en el robot se resumió los valores detallados a continuación en la siguiente tabla:

**Tabla 20. Ciclos de trabajo PWM**

Articulaciones	Velocidad Alta	Velocidad Baja
Codo	1024 (100%)	850 (83%)
Base, hombro, muñeca y gripper	1024 (100%)	510 (49.8%)

**Tabla 21. Distribución de pines de la tarjeta RCM5600**

Puerto	Función
B2	Entrada sensor hombro 1
B3	Entrada sensor hombro 2
B4	Entrada sensor codo 1
B5	Entrada sensor codo 2
B6	Salida led velocidad baja
B7	Salida led velocidad alta
C1	Salida foco del gripper
C4	PWM control velocidades hombro
C5	PWM control velocidades base, codo, muñeca y gripper
D0	Salida abrir gripper
D1	Salida cerrar gripper
D2	Salida muñeca abajo
D3	Salida muñeca arriba
E1	Salida codo abajo
E2	Salida codo arriba
E3	Salida hombro abajo
E5	Salida hombro arriba
E6	Salida base derecha
E7	Salida base izquierda

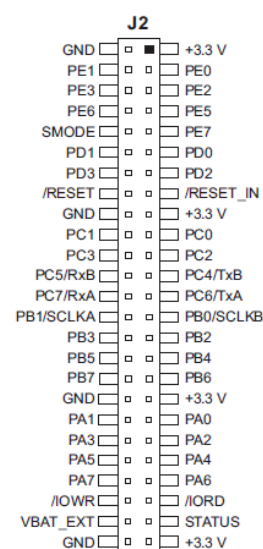
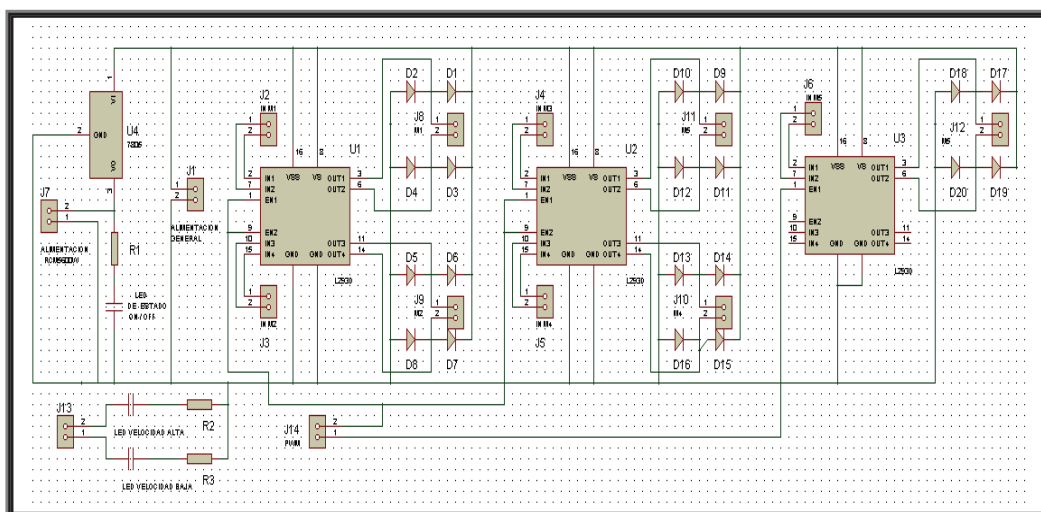


Figura 70. Diagrama de pines puerto J2 de la tarjeta RCM5600W

#### 4.5. Diseño circuito de potencia.

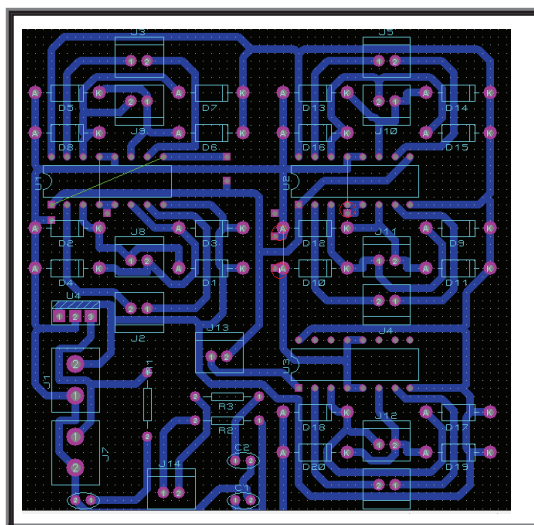
Mediante la herramienta de ISIS de Proteus, se realizó el diseño del circuito de potencia que permite manipular los actuador del brazo robótico, en la figura 70 se detalla la placa con todos los elementos de hardware descritos anteriormente.



**Figura 71. Esquema electrónico tarjeta de potencia.**

El diseño de la tarjeta electrónica para la etapa de potencia de control del brazo robótico se lo ha realizado en el software denominado ARES ya que dispone de la opción de auto-ruteado el cual facilita de mejor manera la conexión de los elementos insertados en el diseño además de manipular las pistas para colocarlos de una forma más compacta.

A continuación se muestra el diseño en ARES de la tarjeta de potencia:

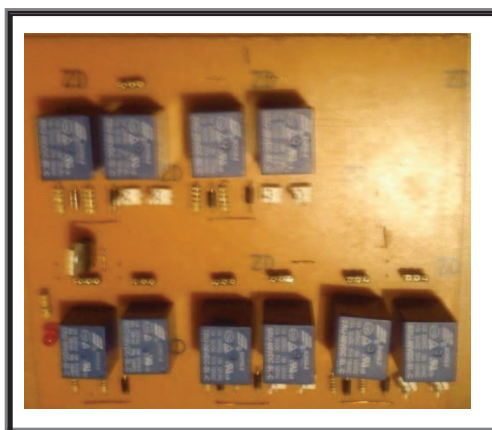


**Figura 72. Esquema electrónico diseñado en ARES**





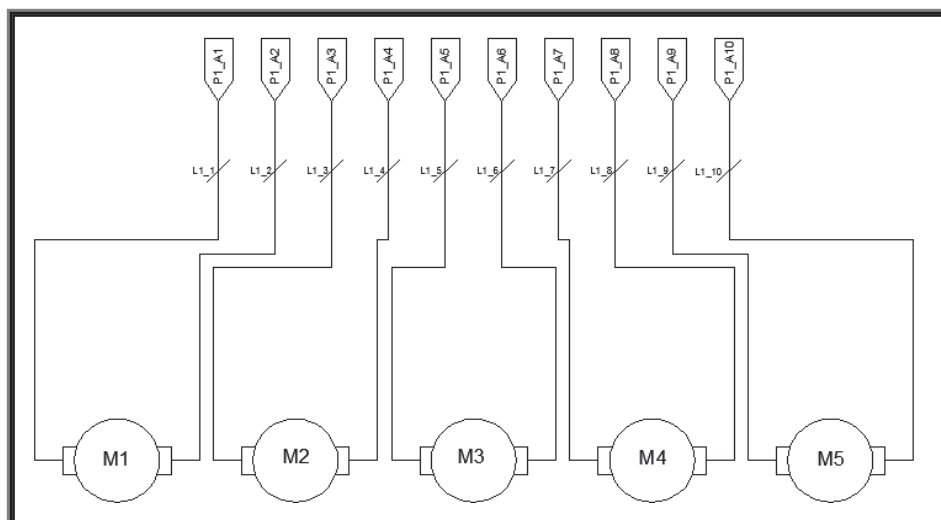
Se realizo anteriormente al circuito de potencia con puentes H un circuito de control el cual incluía relés de 6v como se muestra en la Figura 75, sustituyendo por el actual ya que se observo que los relés al ser dispositivos electromecánicos con el pasar del tiempo presentarían fallas, ya que sus partes mecánicas tenderían a desgastarse. Este diseño no tenía la posibilidad de realizar control de velocidad para el movimiento del robot, generaba ruidos en la activación de los relés, sus elementos al ser grandes en comparación con los puentes H dieron lugar a un circuito con dimensiones demasiado extensas, por tales desventajas electrónicas y con el fin de simplificar y reducir el tamaño del circuito de potencia se optó por el uso del diseño de la figura 73.



**Figura 75. Diseño tarjeta de potencia realizado con relés.**

#### **4.6. Diagramas eléctricos de conexión**

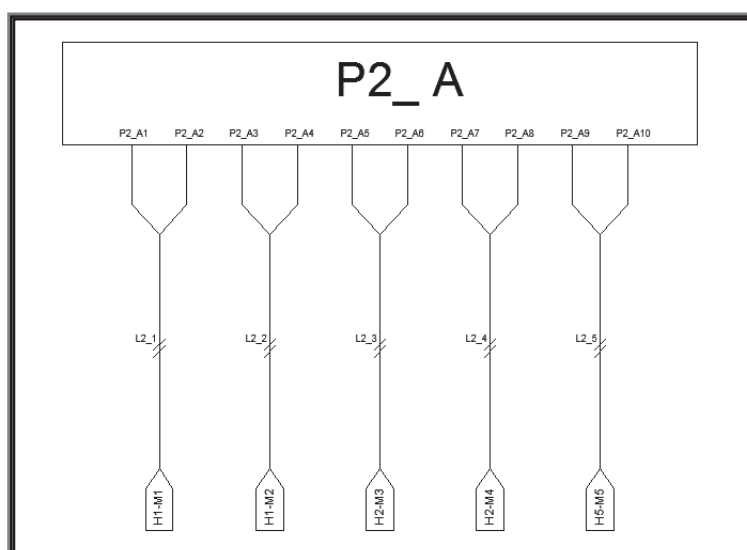
Con el fin de tener planos que indiquen la conexión eléctrica real entre los distintos elementos se diseño los siguientes esquemas.



**Figura 76. Conexión motores al puerto P1\_A.**

**Tabla 22. Descripción de símbolos figura 4.12**

Símbolo	Articulación
M1	Motor de la base
M2	Motor del hombro
M3	Motor del codo
M4	Motor de la muñeca
M5	Motor del Gripper



**Figura 77. Conexión del puerto P2\_A hacia los puentes H**

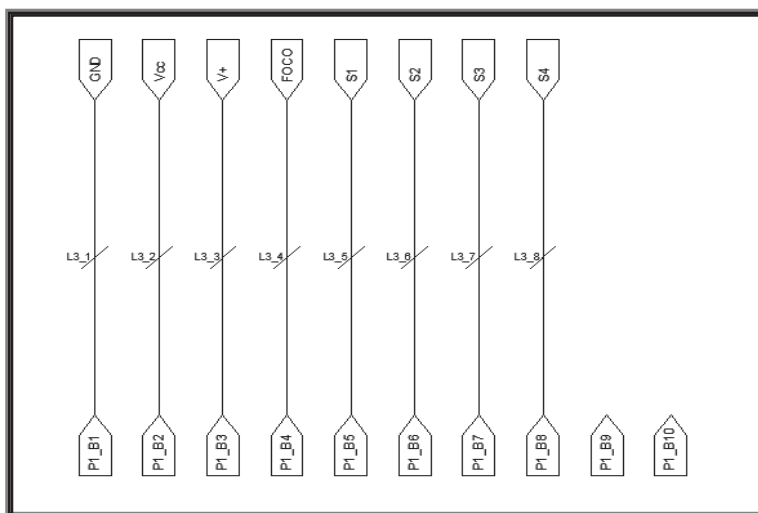


Figura 78. Conexión sensores, foco y alimentación - puerto P1\_B

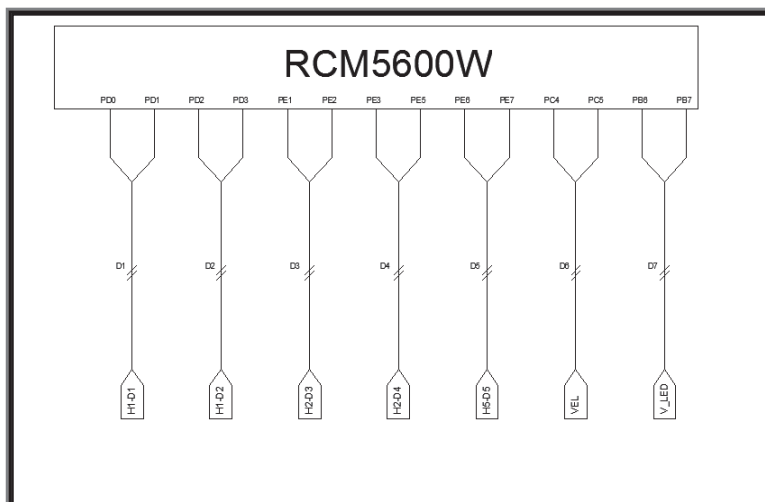


Figura 79. Conexión RCM5600W hacia los puentes H y velocidades

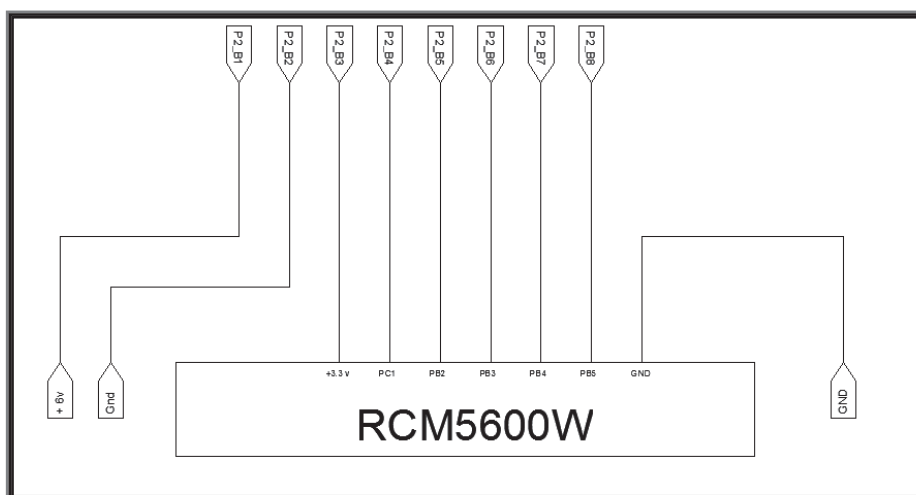
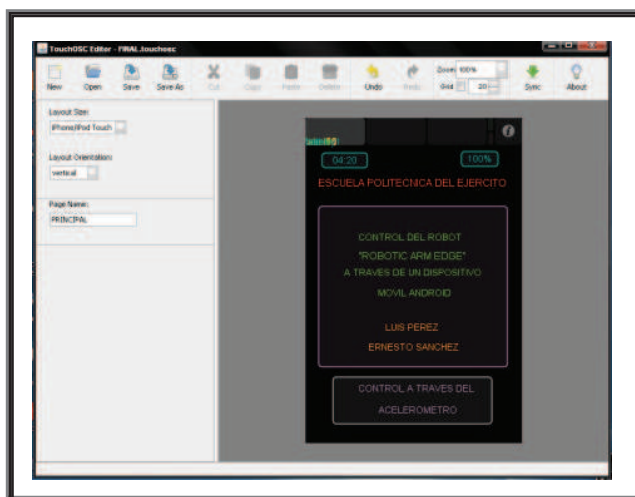


Figura 80. Conexión puerto P2\_B hacia RCM5600 mas alimentaciones.

#### 4.7. Cargar el diseño realizado en el editor TouchOsc hacia el teléfono

Una vez terminada el diseño del HMI en la aplicación TouchOsc como se puede observar en la figura 81 y para poder transferir el archivo vía WiFi se necesita realizar los siguientes pasos:



**Figura 81. Diseño realizado en TouchOSC**

1. Para descargar el HMI diseñado hacia el dispositivo móvil, se debe tener instalado previamente la aplicación “Wifi File Transfer” en el móvil. Descargada de manera gratuita en Google Play (tienda online de Google para aplicaciones android).



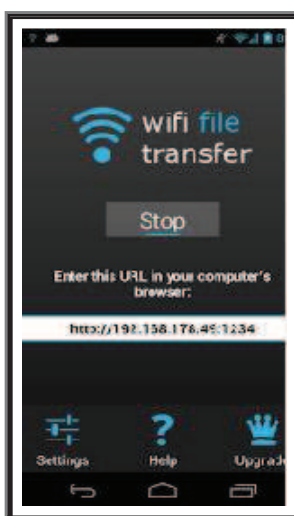
**Figura 82. Aplicación WiFi FileTransfer**

2. Una vez instalada la aplicación y abrirla se obtendrá la siguiente imagen:



**Figura 83. Pantalla de presentación de la aplicación**

3. Para dar inicio a la aplicación se presiona la opción Start, la cual dará una dirección IP.



**Figura 84. Ejecución de la aplicación**

4. La dirección IP que entrega la aplicación se debe colocar en el navegador de internet de la computadora y teniendo como resultado la siguiente imagen, en la cual aparece la opción de búsqueda y descarga del archivo que se desea transferir hacia el dispositivo móvil.



**Figura 85. Transferencia de archivos hacia el dispositivo móvil**

5. Se debe tomar en cuenta que para el uso de esta aplicación tanto el dispositivo móvil como el computador deben estar conectados en la misma red.

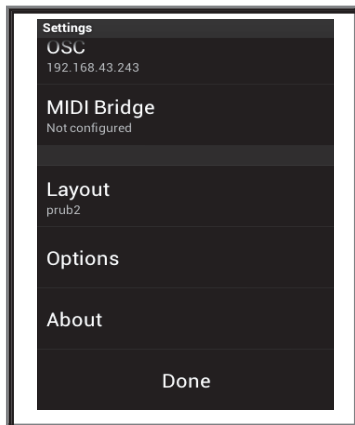
#### **4.8. Ejecutar y configurar aplicación TouchOSC**

1. Descargado y instalado la aplicación de TouchOSC en el dispositivo android lo ejecutamos.



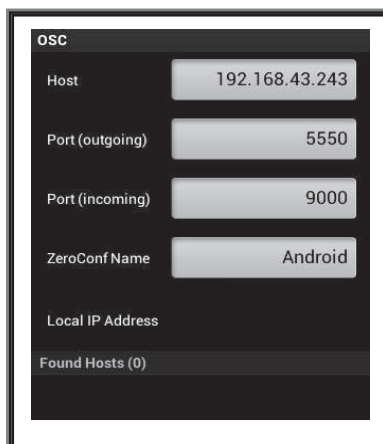
**Figura 86. Ejecutar TouchOSC versión 1.9.1**

2. Ventana de principal del TouchOSC, en la cual se puede configurar los parámetros principales que tendrá la comunicación y el HMI.



**Figura 87. Ventana de herramientas TouchOSC**

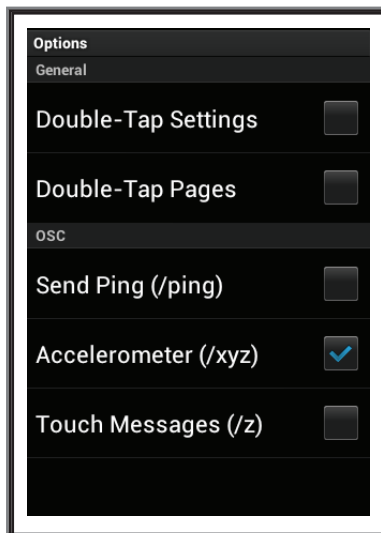
3. Seleccionando la opción OSC accedemos a los parámetros de red.
  - a. Como primer punto se tiene el host, siendo este la dirección IP del destinatario del mensaje.
  - b. Port(outgoing) es el puerto destino en el cual el host destinatario va a recibir el mensaje.
  - c. Port(incoming) puerto de recepción de datos para el presente proyecto no es utilizado.
  - d. Para identificar el nombre del dispositivo en la red.



**Figura 88. Parámetros de red TouchOSC**

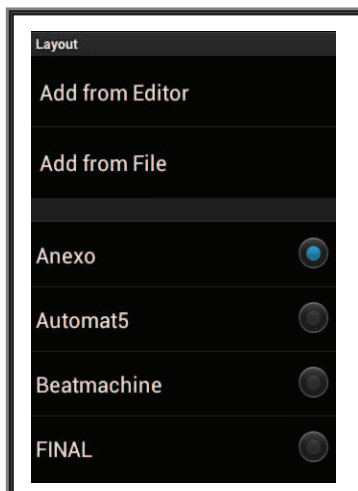


- En el caso de querer usar el acelerómetro, dirigirse a Options y seccionar la opción Accelerometer(/xyz), para el uso de los botones esta opción debe estar desactivada.



**Figura 89. Habilitar el uso del acelerómetro**

- Una vez diseñado el HMI en el Editor y transferido al teléfono, éste es cargado en la tarjeta de memoria del dispositivo móvil, el cual debe ser seleccionado en la aplicación utilizando la opción Add from File, buscar el directorio donde se almacenó el HMI diseñado y seleccionarlo.



**Figura 90. Cargar HMI diseñado**

6. Por último seleccionar DONE para poder correr el HMI en la aplicación y empezar a interactuar con el brazo robótico.

#### 4.9. Configuración del teléfono Sony Xperia ST23a como Access Point

Como se menciona anteriormente, el dispositivo móvil a demás de ser usado como un terminal de operación del brazo robótico funciona como un punto de acceso inalámbrico, el cual interconecta a la red a la tarjeta RCM 5600W, el computador y el teléfono en sí.

A continuación se detalla el proceso de configuración.

1. Ir a la configuración general del sistema del teléfono y seleccionar conexiones inalámbricas y red. Figura 91.



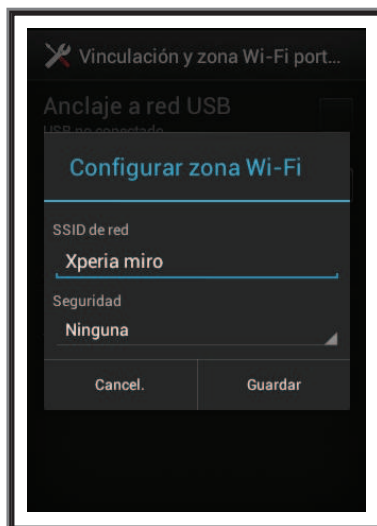
**Figura 91. Configuración Access Point paso 1**

- 2.- Seleccionar vinculación y zona Wi – Fi. Figura 91. En esta opción permite activar el Access Point al elegir Zona Wi-Fi portátil y también configurar características red.



**Figura 92. Configuración Access Point paso 2**

- 3.- Para dar un nombre que identifique a la red y el tipo seguridad Wi-Fi, acceder a configurar zona Wi-Fi. Figura 93.



**Figura 93. Configuración Access Point paso 3**

- 4.- Ya configurada la red, existe una opción de acceso rápido para activar y desactivar el Access Point utilizando el widgets del propio teléfono llamado herramientas. Figura 94.



**Figura 94. Acceso directo celular como Access Point**

#### 4.10. Pruebas del sistema.

El primer paso es verificar conectividad entre los dispositivos mediante la ayuda de las líneas de comando de Windows comprobamos la conexión existente entre todos los dispositivos. Recordando las direcciones IP de la red.

**Tabla 23. Distribución de direcciones IP en la red**

Dispositivo	IP	Mascara de red	Puerta de enlace
PC	192.168.43.242	255.255.255.0	192.168.43.1
RCM 5600W	192.168.43.243	255.255.255.0	192.168.43.1
Dispositivo Android	192.168.43.1	255.255.255.0	

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\landsam>ping 192.168.43.243

Haciendo ping a 192.168.43.243 con 32 bytes de datos:
Respuesta desde 192.168.43.243: bytes=32 tiempo=40ms TTL=64
Respuesta desde 192.168.43.243: bytes=32 tiempo=19ms TTL=64
Respuesta desde 192.168.43.243: bytes=32 tiempo=29ms TTL=64
Respuesta desde 192.168.43.243: bytes=32 tiempo=22ms TTL=64

Estadísticas de ping para 192.168.43.243:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 19ms, Máximo = 40ms, Media = 27ms

C:\Users\landsam>_
  
```

**Figura 95. Conectividad PC – RCM5600W**

```

C:\Windows\system32\cmd.exe

C:\Users\landsam>ping 192.168.43.1

Haciendo ping a 192.168.43.1 con 32 bytes de datos:
Respuesta desde 192.168.43.1: bytes=32 tiempo=25ms TTL=64
Respuesta desde 192.168.43.1: bytes=32 tiempo=21ms TTL=64
Respuesta desde 192.168.43.1: bytes=32 tiempo=24ms TTL=64
Respuesta desde 192.168.43.1: bytes=32 tiempo=17ms TTL=64

Estadísticas de ping para 192.168.43.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 17ms, Máximo = 25ms, Media = 21ms

C:\Users\landsam>_
  
```

**Figura 96. Conectividad PC – Dispositivo móvil**

Al existir conectividad entre el dispositivo móvil y la Pc, la tarjeta RCM 5600W y la Pc se puede concluir que existe comunicación entre la tarjeta y el dispositivo móvil.

#### 4.11. Análisis de datos recibidos por la tarjeta RCM 5600W

Mediante la consola de Dinamic C podemos visualizar y analizar UDP enviadas hacia la tarjeta RCM 5600W.

Al enviar datos desde la PC hacia la tarjeta RCM5600W, se despacha un palabra de control con los siguientes datos compuesta de la siguiente manera. Tabla 24.

**Tabla 24. Palabra de control PC**

Palabra de control						
Caracter 1	Caracter 2	Caracter 3	Caracter 4	Caracter 5	Caracter 6	Caracter 7
Primer movimiento articulación	Segundo movimiento articulación	Selector Pc o Dispositivo Android	Velocidad Alta	Velocidad Baja	Foco Gripper	Automático

**Tabla 25. Detalle de datos enviados desde Java- PC**

Caracter	Función	Dato
<b>Primero</b>	Abrir gripper	0
	Bajar muñeca	2
	Bajar codo	4
	Bajar hombro	6
	Base derecha	8
<b>Segundo</b>	Cerrar gripper	1
	Subir muñeca	3
	Subir codo	5
	Subir hombro	7
	Base izquierda	9
<b>Tercero</b>	Pc	0
	Android	1
<b>Cuarto</b>	Velocidad alta	0
	Velocidad baja	1
<b>Quinto</b>	Foco ON	1
	Foco OFF	0

En la figura 97 es un ejemplo del dato recibido por la tarjeta RCM5600W donde; se abre el gripper, se usa el PC como elemento de control, trabaja en velocidad alta y el foco del gripper se encuentra apagado.

```

BUFFER PC-JAVA:
BUFFER PC-JAVA:
BUFFER PC-JAVA:100100
BUFFER PC-JAVA:
BUFFER PC-JAVA:
BUFFER PC-JAVA:
BUFFER PC-JAVA:

```

**Figura 97. Ejemplo dato recibido por RCM5600W desde la Pc**

En el caso de la recepción de datos del dispositivo móvil android, se obtiene un trama con una cabecera que identifica el evento realizado en el dispositivo seguido de la información que contiene dicho evento.

A continuación se presenta una tabla en la que se señala el botón presionado y la información que se recibe para posteriormente utilizarlo en el programa de la tarjeta RCM5600W:

**Tabla 26. Datos enviados desde el dispositivo androide al RCM5600W**

BOTÓN PRESIONADO	DATO RECIBIDO COMO STRING	CARACTER QUE SE UTILIZA	COMPARACIÓN ASCII DECIMAL
Base: Izquierda	/SECUNDARIA/B1	1	49
Base: Derecha	/SECUNDARIA/B2	2	50
Hombro: Abajo	/SECUNDARIA/B3	3	51
Hombro: Arriba	/SECUNDARIA/B4	4	52
Codo: Abajo	/SECUNDARIA/B5	5	53
Codo: Arriba	/SECUNDARIA/B6	6	54
Muñeca: Abajo	/SECUNDARIA/B7	7	55
Muñeca: Arriba	/SECUNDARIA/B8	8	56
Gripper: Abrir	/SECUNDARIA/B9	9	57
Gripper: Cerrar	/SECUNDARIA/Ba	a	97
Velocidad: Baja	/3/Bb	b	98
Velocidad: Alta	/3/Bc	c	99
Acelerómetro On	/3/Bd	d	100
Acelerómetro Off	/3/Be	e	101
Led: Apagado	/3/Bf	f	102
Led: Encendido	/3/Bg	g	103
Automático	/3/Bh	h	104
Acelerómetro	/accxyz	Varía el dato según la inclinación del dispositivo móvil	65 y 64 192 y 193

Para la segunda ventana del HMI del teléfono los datos son recibidos como se presenta a continuación:

**Tabla 27. Palabra de control dispositivo Android ventana 2**

Caracteres UDP[1]- UDP[12]	Caracteres UDP[13]
Cabecera: recibe el nombre de la ventana 2, seguido del botón seleccionado	Dato: dependiendo del botón presionado llegan valores del 1-9 en código ASCII, el último botón es el caracter "a".

```

BUFFER PC-JAVA:
Dato recibido 0
BUFFER PC-JAVA:
Dato recibido /SECUNDARIA/B1 49
BUFFER PC-JAVA:
Dato recibido 0
BUFFER PC-JAVA:
Dato recibido 0
BUFFER PC-JAVA:
Dato recibido ~
  
```

**Figura 98. Ejemplo botón B1 Base Izquierda**

En la tercera ventana del HMI del teléfono los datos son:

**Tabla 28. Palabra de control dispositivo android ventana 3**

Caracteres UDP[1]- UDP[3]	Caracteres UDP[4]
Cabecera: recibe el nombre de la ventana 3, seguido del botón seleccionado	Dato: dependiendo del botón presionado llegan valores con los caracteres desde la "b hasta la g" en su respectivo código ASCII

```

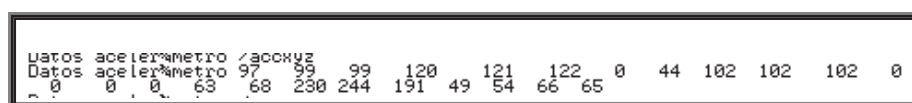
Dato recibido 0
Dato recibido 0
Dato recibido 0
Dato recibido 0
Dato recibido 0
Dato recibido /3/Bd 100
Dato recibido 0
Dato recibido 0
Dato recibido 0
Dato recibido 0
  
```

**Figura 99. Ejemplo botón Acelerómetro On**

Como se puede observar en la tabla 4.13 tanto las cadenas de caracteres “SECUNDARIA” y “3” son los nombres de las ventanas que se dispone en el dispositivo móvil, también se recibe el nombre de los botones que se ha colocado en el diseño del HMI del dispositivo móvil que van desde “B1” hasta “Bg” y para el uso en el código de la tarjeta solo se trata los últimos caracteres que van desde “1” hasta “g” que en este caso se los recibe en código ASCII para el uso en el código según sea necesario.

Los datos del acelerómetro dependen de la inclinación del dispositivo, en la cabecera para todos los movimientos el dispositivo móvil envía solo una información indicando que estamos ocupando esta función, dicha cabecera es el string “/accxyz” y en el segmento de datos los caracteres referenciales son los ubicados en las demás posiciones del datagrama recibido.

A continuación se detalla el desglose de los datos recibidos del acelerómetro, para ello se hizo la impresión del datagrama recibido en la tarjeta y la representación de cada uno de ellos en decimal:



```

Datos acelerometro /accxyz
Datos acelerometro 97 99 99 120 121 122 0 44 102 102 102 0
0 0 0 63 68 230 244 191 49 54 66 65

```

**Figura 100. Comparación String y decimal datos acelerómetro**

El primer mensaje que se observa es el dato recibido como string y después se hizo la impresión en decimal para de este modo comparar los datos recibidos, a continuación se presenta una tabla con las posiciones de cada uno de los datos:








**Tabla 29. Desglose datos recibidos del acelerómetro**

<b>DATO</b>	<b>POSICIÓN</b>	<b>REPRESENTACIÓN ASCII DECIMAL</b>
a	udpBuffer[1]	97
c	udpBuffer[2]	99
c	udpBuffer[3]	99
x	udpBuffer[4]	120
y	udpBuffer[5]	121
z	udpBuffer[6]	122
Datos adicionales por el uso del sensor (no se utilizan)	udpBuffer[7]- udpBuffer[15]	
Datos del sensor eje X	udpBuffer[16]- udpBuffer[19]	
Datos del sensor eje Y	udpBuffer[20]- udpBuffer[23]	
Datos del sensor eje Z	udpBuffer[24]- udpBuffer[27]	

Para la representación de los datos del sensor se utilizo los datos de las posiciones del 16 al 27 teniendo en cuenta que para cada una de sus posiciones se obtiene diferentes datos como se puede apreciar en la siguiente tabla.

Tabla 30. Análisis de datos acelerómetro

POSICIÓN	DESCRIPCIÓN	POSICIÓN DATAGRAMA	RANGO DE DATOS Y ANGULO DE TRABAJO
 <p>Figura 101. Estado inicial</p>	<p><b>Estado inicial</b></p> <p>No se toman en cuenta dichos datos, no existe movimiento del robot</p>	<p><b>udpBuffer[24]-udpBuffer[27]</b></p> <p>Son los valores que corresponden al eje Z</p>	<p>Ángulo de 0 ° y 180 °</p>
 <p>Figura 102. Inclínación 1</p>	<p><b>Inclínación 1</b></p> <p>El movimiento lo realizan las articulaciones del hombro el codo la muñeca y el gripper en sentido ascendente.</p>	<p><b>udpBuffer[20]-udpBuffer[23]</b></p> <p>El carácter utilizado es la que se encuentra en la posición 20 ya que es el dato más significativo</p>	<p>Dato : 64 ( @ ) Angulo: 0-45</p> <p>Dato: 65 ( A ) Angulo: 46-90</p>
 <p>Figura 103. Inclínación 2</p>	<p><b>Inclínación 2</b></p> <p>El movimiento lo realizan las articulaciones del hombro el codo la muñeca y el gripper en sentido descendente.</p>	<p><b>udpBuffer[20]-udpBuffer[23]</b></p> <p>El carácter utilizado es la que se encuentra en la posición 20 ya que es el dato más significativo</p>	<p>Dato: 192 ( L ) Angulo: 90-135</p> <p>Dato: 193 ( ⊥ ) Angulo: 136-180</p>
 <p>Figura 104. Inclínación 3</p>	<p><b>Inclínación 3</b></p> <p>El movimiento lo realizan las articulaciones de la base girando hacia la derecha.</p>	<p><b>udpBuffer[16]-udpBuffer[19]</b></p> <p>El carácter utilizado es la que se encuentra en la posición 16 ya que es el dato más significativo</p>	<p>Dato: 192 ( L ) Angulo: 90-135</p> <p>Dato: 193 ( ⊥ ) Angulo: 136-180</p>
 <p>Figura 105. Inclínación 4</p>	<p><b>Inclínación 4</b></p> <p>El movimiento lo realizan las articulaciones de la base girando hacia la izquierda.</p>	<p><b>udpBuffer[16]-udpBuffer[19]</b></p> <p>El carácter utilizado es la que se encuentra en la posición 16 ya que es el dato más significativo</p>	<p>Dato: 64 ( @ ) Angulo: 0-45</p> <p>Dato: 65 ( A ) Angulo: 46-90</p>

#### 4.12. Medición de tiempos y velocidades de las articulaciones.

Los datos de los tiempos que demoran cada articulación en realizar el movimiento dentro de los límites fijados se detalla en la siguiente tabla.

**Tabla 31. Tiempos por articulación**

ARTICULACIÓN	TIEMPO EN VELOCIDAD BAJA(s)		TIEMPO EN VELOCIDAD ALTA(s)	
	BASE	4.81 Izq.	4.82 Der.	4.04 Izq.
HOMBRO	4.28 Baj.	5.52 Subir	2.56 Baj.	2.8 Subir
CODO	2.83 Baj.	3.04 Subir	2.11 Baj.	2.77 Subir
MUÑECA	3.99 Baj.	4.22 Subir	2.88 Baj.	3.21 Subir
GRIPPER	3.56 Abrir	3.52 Cerrar	2.75 Abrir	2.74 Cerrar

Se realizo las mediciones de la velocidad de cada articulación para diagnosticar como actúan los motores ante las dos velocidades establecidas por el PWM que entrega la tarjeta RCM5600W y así lograr en las de velocidad de las articulaciones del brazo robótico un mejor control y uniformidad en el movimiento.

**Tabla 32. Velocidades por articulación**

ARTICULACIÓN	VELOCIDAD ANGULAR BAJA(rad/s)		VELOCIDAD ANGULAR ALTA(rad/s)	
	BASE	0.206 Izq.	0.205 Der.	0.224 Izq.
HOMBRO	0.183 Baj.	0.142 Subir	0.306 Baj.	0.280 Subir
CODO	0.298 Baj.	0.276 Subir	0.410 Baj.	0.305 Subir
MUÑECA	0.207 Baj.	0.195 Subir	0.292 Baj.	0.260 Subir
GRIPPER	0.220(cm/s) Abrir	0.220(cm/s) Cerrar	0.242(cm/s) Abrir	0.243(cm/s) Cerrar

#### **4.13. Consideraciones de operación.**

- Evitar posibles interferencias generadas por equipos electrónicos.
- Obstáculos que puedan disminuir la calidad de la señal.
- Distancia máxima entre los diferentes dispositivos.
- Utilizar un mismo canal de transmisión en todos los equipos y verificar conectividad.
- Tener presente los posibles retardos que se pueda originarse en la red.
- Tomar en cuenta la velocidad de las articulaciones del brazo robótico, para diferentes aplicaciones que se la pueda dar.
- Considerar el espacio de trabajo del robot, para evitar accidentes y posibles daños al mecanismo al forzar las articulaciones.
- Tomar en cuenta el espacio de trabajo del robot para evitar accidentes tanto para el operario como para el instrumento en sí.
- Verificar constantemente que la alimentación eléctrica del robot se encuentre en rangos de operación aceptables.

## CAPÍTULO V

### ANÁLISIS DE COSTOS.

En esta sección se considerara el costo de implementación del presente trabajo, en el cual se va a realizar un análisis en varios aspectos ya sea en el de construcción, equipo, entre otros.

#### 5.1. Costo de construcción:

Aquí vamos a incluir los costos de obra que más comúnmente se obtendría en el presente proyecto:

**Tabla 33. Costos construcción y diseño**

Descripción	Costo en dólares \$
Construcción y diseño de la placa electrónica	100.00
<b>Total</b>	<b>100.00</b>

#### 5.2. Costo de materiales:

Se detalla el costo de los materiales utilizados para la implementación del presente proyecto:

**Tabla 34. Costos materiales**

Material	Cantidad	Costo unitario \$	Costo total \$
Placa de fibra de vidrio	4	2.50	10.00
Integrado l293d	6	2.80	16.8
Regulador de voltaje 7805	4	0.75	3.00
Resistencia 330 Ohm	30	0.10	3.00
Diodo 1N4007	45	0.15	6.75
Diodo led	10	0.10	1.00
Conectores	60	0.35	21.00
Brazo robótico "robotic arm edge"	1	150.00	150.00
Baterías tipo D	8	2.50	20.00
Tarjeta RCM5600W	1	150.00	150.00
Otros	1	40.00	40.00
		<b>Total</b>	<b>423.55</b>

Se debe tomar en cuenta que el costo de la tarjeta RCM5600W fue cotizado con el precio del extranjero sin tomar en cuenta el costo de envío.

### 5.3. Costo de puesta en marcha:

Aquí se incluye valores de materiales u otros elementos que se pueden necesitar para la puesta en marcha del presente proyecto, además se consideran como costos de imprevistos a elementos que no se han considerado inicialmente en el desarrollo del presente trabajo y tenemos:

**Tabla 35. Costos puesta en marcha**

<b>Descripción</b>	<b>Costo en dólares \$</b>
Costo de puesta en marcha	35.00
Materiales adicionales	60.00
Transporte	30.00
Pruebas	28.00
Repuestos	60.00
Imprevistos	150.00
<b>Total</b>	<b>263.00</b>

Una vez realizado el análisis de los costos que se podría necesitar para el desarrollo de este trabajo se obtiene un total de \$886.55 el cual es un valor aceptable de acuerdo a la propuesta realizada en este trabajo ya que tiene como objetivo la investigación de nuevas tecnologías.

## CAPÍTULO VI

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1. Conclusiones

- Al ser un microprocesador de propósito general, con la capacidad de conexiones tanto físicas como remotas, poseer un alta velocidad de operación, una gran capacidad de memoria RAM y flash y una programación basada en lenguaje C, puede ser enfocado a aplicaciones de control industrial y aplicaciones desarrolladas en red.
- Mediante los resultados conseguidos, se afirma que el sistema electrónico controla óptimamente los motores de las articulaciones obteniendo los resultados esperados.
- Con respecto al movimiento del brazo de las articulaciones se afirma, que la inercia que presentan las articulaciones al movimiento influyen directamente en la fuerza que proporciona por el motor el cual es controlado por el PWM.
- Se incluyo sensores en las articulaciones del hombro y el codo con el fin de conseguir de que el brazo robótico llegue a posiciones críticas evitando daños.
- Se comprobó que el microcontrolador RCM5600W, tiene procesamiento y una gran escalabilidad en una red ya que puede tener varios clientes a la vez.
- La comunicación UDP es eficiente en este tipo de aplicaciones por el motivo que se hace en tiempo real siendo más rápida a comparación que TCP.

- Mediante la tarjeta RCM5600W permite implementar sistemas con comunicación inalámbrica pudiendo hacer un control y monitoreo de dispositivos en lugares remotos.

## **6.2. Recomendaciones**

- Se recomienda que se diseñe e implemente un sistema basado en las conclusiones, recomendaciones y trabajos a futuro con el fin de implementar en un sistema real y poder corregir posibles errores.
- Se recomienda verificar las corrientes de los motores que se estén utilizando en el manipulador para de esta forma dimensionar de una mejor manera los elementos que se van a utilizar para el control del manipulador.
- En el caso de la tarjeta RCM5600W se recomienda verificar su hoja técnica de información para su uso adecuado y evitar daños en el mismo.
- Al momento de realizar el diseño de un circuito se recomienda tratar de utilizar dispositivos netamente electrónicos ya que como se expuso en el presente trabajo, al tratar de hacer uso de dispositivos electromecánicos en el futuro puede presentar fallas en sus partes mecánicas.

## **6.3. Trabajo a futuro**

- Realizar un control de posición, utilizando e implementando una instrumentación adecuada.
- Realizar el control mediante un guante electrónico, emulando el movimiento de las articulaciones del brazo humano.
- Poder grabar posiciones en el espacio del brazo robótico.



- Experimentar con procesos robóticos industriales reales la aplicación android, y el control inalámbrico del brazo robótico.
- Crear una aplicación en la cual se realice control de calidad de diferentes procesos.
- Ampliar la red e interactuar con más terminales de control.
- Analizar en la actual realidad Ecuatoriana la factibilidad del control de procesos robóticos industriales inalámbricos.
- Realizar un control Web utilizando la tarjeta RCM5600W.

## REFERENCIAS BIBLIOGRÁFICAS

ARM-EDGE, R. (2008). *Assembly and Instruction Manual*.

BitLibrary. (2008). *BitLibrary*. Retrieved Abril 14, 2013, from

<http://www.bitlib.net/show.php?id=24505822>

*es.wikipedia.org*. (n.d.). Retrieved from <https://es.wikipedia.org/wiki/Gir%C3%B3scopo>

Fernández, R. (2007). *Sistema de adquisición de posicionamiento geográfico*. Ingeniería Electrónica de la Universidad Politécnica de Cataluña.

FindTheBest. (2013, Octubre 24). *FindTheBest.com*. Retrieved from [http://development-software.findthebest.com/saved\\_compare/Xcode-vs-NetBeans-vs-Eclipse](http://development-software.findthebest.com/saved_compare/Xcode-vs-NetBeans-vs-Eclipse)

Jürgens, B. (2008). *agenciaidea.es*. Retrieved from

[http://www.agenciaidea.es/c/document\\_library/get\\_file?uuid=9e84835b-e749-4b6c-a01d-f60d3399fae9&groupId=10157](http://www.agenciaidea.es/c/document_library/get_file?uuid=9e84835b-e749-4b6c-a01d-f60d3399fae9&groupId=10157)

kioskea. (n.d.). *kioskea.net*. Retrieved from

<http://es.kioskea.net/contents/wifi/wifiintro.php3>

Magni, J. Á. (2009). *MEMS ACELERÓMETROS*. Universidad de Cantabria.

Rabbit. (n.d.). *MiniCore RCM5600W OEM User Manual*. Retrieved from <ftp1.digi.com>.

Sony. (2013, Octubre 22). Retrieved from [http://www-support-](http://www-support-downloads.sonymobile.com/st23/userguide_ES_ST23i_1267-5240.2_Android4.0.pdf)

[downloads.sonymobile.com/st23/userguide\\_ES\\_ST23i\\_1267-5240.2\\_Android4.0.pdf](http://www-support-downloads.sonymobile.com/st23/userguide_ES_ST23i_1267-5240.2_Android4.0.pdf)

zonesecurity. (2011, Abril 03). *zonesecurity2011.blogspot.com*. Retrieved from

<http://zonesecurity2011.blogspot.com/2011/04/diferencias-entre-los-puertos.html>

## FECHA DE ENTREGA

El proyecto fue entregado al Departamento de Eléctrica y Electrónica y reposa en la Universidad de las Fuerzas Armadas- ESPE desde:

Sangolquí: 19 DE DICIEMBRE DE 2013

## ELABORADO POR:



Luis Andrés Pérez Llerena

CI: 171682639-9



Ernesto Julián Sánchez Sánchez

CI: 171058692-4

## AUTORIDADES:



Ing. LUIS OROZCO

DIRECTOR DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL