



*“Responsabilidad con pensamiento positivo”*

# **UNIVERSIDAD TECNOLÓGICA ISRAEL**

## **TRABAJO DE TITULACIÓN**

**CARRERA:** ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

**TEMA:** Diseño de un prototipo de proceso industrial para la clasificación de objetos con la utilización de un kit de robótica Lego Mindstorms NXT y herramientas de software como Matlab y Simulink.

**AUTOR:** Freddy Julián Ortiz López

**TUTOR:** Ing. José Robles Salazar Mg.

Año 2014

## **AUTORÍA DE TRABAJO DE TITULACIÓN**

El abajo firmante, en calidad de estudiante de la carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este Trabajo de Titulación requisito previo a la obtención del Grado de Ingeniería en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M., Octubre de 2014

-----

Freddy Julián Ortiz López

CI: 1804025060

## **APROBACIÓN DEL TUTOR**

En mi calidad del Tutor del Proyecto Integrador de Carrera certifico:

Que el Trabajo de Titulación “DISEÑO DE UN PROTOTIPO DE PROCESO INDUSTRIAL PARA LA CLASIFICACIÓN DE OBJETOS CON LA UTILIZACIÓN DE UN KIT DE ROBÓTICA LEGO MINDSTORMS NXT Y HERRAMIENTAS DE SOFTWARE COMO MATLAB Y SIMULINK”, presentada por el señor Freddy Julián Ortiz López, estudiante de la carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M., Octubre 2014

TUTOR

-----

Ing. José Robles Salazar Mg.

## APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado, aprueban el Trabajo de Titulación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D.M., Octubre de 2014

Para constancia firman:

TRIBUNAL DE GRADO

-----  
PRESIDENTE

-----  
MIEMBRO 1

-----  
MIEMBRO 2

## **AGRADECIMIENTO**

Primeramente a Dios que es el que incondicionalmente me da fuerzas y esperanza para lograr alcanzar las metas propuestas.

A todas las personas que despertaron en mi las cualidades y defectos con los que cuento hoy, entre ellos mis amigos (as) que he compartido muchas cosas a lo largo de mi carrera estudiantil, a mis hermanos (as) los cuales siempre me han estado apoyando para seguir adelante y una mención especial a mis padres los cuales con su sacrificio me han ayudado en los momentos más difíciles, ellos son fuente de mi inspiración, por los cuales he llegado a ser la persona que soy, ellos me enseñaron el verdadero valor que hay que darle uno a la vida.

A todos ellos con los cuales aprendí que no importa que obstáculo se ponga en el camino hacia el éxito, nunca darme por vencido.

Freddy Julián Ortiz López

## **DEDICATORIA**

En el mundo encontrarás que el éxito empieza con la voluntad del hombre. Si quieres conseguir algo, jamás te des por vencido, recuerda la constancia de una gota de agua puede más que la firmeza de una montaña.

Dedicado a la perseverancia, la constancia, la rectitud, la fuerza de voluntad, el carácter, el entusiasmo, la lealtad, la iniciativa, la comprensión, el compañerismo, el amor, y a todos aquellos atributos que la vida brinda al ser humano y que gracias a ellos el hombre puede llegar alcanzar el éxito.

Dedicado para ustedes mami Violeta y papi Segundo mis padres.

Freddy Julián Ortiz López

## ÍNDICE DE CONTENIDOS

AUTORÍA DE TRABAJO DE TITULACIÓN .....	II
APROBACIÓN DEL TUTOR.....	III
APROBACIÓN DEL TRIBUNAL DE GRADO .....	IV
AGRADECIMIENTO .....	V
DEDICATORIA .....	VI
ÍNDICE DE CONTENIDOS.....	VII
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS .....	XI
INTRODUCCIÓN.....	1
Objetivos.....	1
Objetivo general.....	1
Objetivos específicos .....	2
CAPÍTULO I.....	3
1. FUNDAMENTACIÓN TEÓRICA .....	3
1.1. Introducción.....	3
1.2. Lego Mindstorms NXT.....	3
1.2.1. Componentes del kit.....	4
1.2.1.1. Bloque programable NXT .....	5
1.2.1.2. Sensor de Luz .....	8
1.2.1.3. Sensor Ultrasónico .....	9
1.2.1.4. Sensor de Tacto .....	11
1.2.1.5. Sensor de Sonido .....	11
1.2.1.6. Servomotores .....	12
1.2.2. Herramientas de Software.....	14

1.2.2.1.	Programación de Lego Mindstorms NXT .....	14
1.2.2.2.	Matlab y Simulink .....	15
CAPITULO II.....		18
2.	DIAGNÓSTICO Y BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO .....	18
2.1.	Introducción.....	18
2.2.	Fundamentación de la construcción.....	18
2.3.	Análisis de la estructura .....	19
2.4.	Análisis del Software .....	21
2.4.1.	Software para la banda transportadora .....	21
2.4.2.	Software para el brazo robótico.....	21
2.4.3.	RWTH - Mindstorms NXT Toolbox .....	22
2.5.	Comunicación entre PC vs NXT.....	26
2.5.1.	Comunicación vía USB.....	27
2.5.2.	Comunicación vía Bluetooth.....	27
CAPITULO III.....		29
3.	PRESENTACIÓN DE RESULTADOS.....	29
3.1.	Introducción.....	29
3.2.	Diseño del Hardware del proyecto .....	29
3.2.1.	Diseño de la banda transportadora .....	30
3.2.2.	Diseño del brazo robótico.....	31
3.3.	Programación del proyecto .....	31
3.3.1.	Programación de la banda transportadora con el software NXT-G.....	31
3.3.1.1.	Diagrama de bloques del funcionamiento.....	32
3.3.1.2.	Programación del bloque NXT-1 de la banda transportadora .....	33
3.3.2.	Programación en Matlab .....	34
3.3.2.1.	Código del archivo *.m.....	36
3.4.	Funcionamiento del sistema.....	37

3.5.	Presentación de resultados .....	40
3.5.1.	Calibración de sensores NXT .....	41
3.5.2.	Valores medidos.....	42
3.6.	Evaluación técnica .....	47
3.7.	Costos del proyecto.....	47
4.	CONCLUSIONES Y RECOMENDACIONES .....	49
	CONCLUSIONES .....	49
	RECOMENDACIONES.....	50
	BIBLIOGRAFÍA.....	51
	ANEXOS.....	53

## ÍNDICE DE FIGURAS

Figura 1.1. Presentación del kit Lego Mindstorms NXT .....	4
Figura 1.2. Características del bloque Mindstorms NXT .....	6
Figura 1.3. Diagrama de bloques del Mindstorms NXT .....	7
Figura 1.4. Sensor de luz.....	9
Figura 1.5. Funcionamiento del sensor de ultrasonido.....	10
Figura 1.6. Sensor Ultrasónico.....	11
Figura 1.7. Sensor de Tacto .....	11
Figura 1.8. Sensor de Sonido .....	12
Figura 1.9. Servomotor .....	14
Figura 1.10. Entorno software NXT-G.....	15
Figura 1.11. Entorno software Matlab .....	17
Figura 2.1. Diseño de la banda transportadora.....	20
Figura 2.2. Diseño del brazo robótico .....	21
Figura 2.3. Interfaz Matlab, instalación del soporte de Lego para Matlab. ....	25
Figura 2.4. Librerías Lego NXT instaladas en el programa Matlab/Simulink.....	26
Figura 2.5. Diagrama de bloques de la comunicación PC vs NXT .....	27
Figura 3.1. Diseño mecánico de la banda transportadora y de la ubicación de los sensores .....	30
Figura 3.2. Diseño mecánico del brazo robótico .....	31
Figura 3.3. Diagrama de bloques de la programación de la banda transportadora.....	33
Figura 3.4. Programación de la banda transportadora.....	34
Figura 3.5. Diagrama de bloques de la programación del brazo robótico .....	35
Figura 3.6. Etapa de inicio del sistema. ....	38
Figura 3.7. Etapa de inicio de sensado de nivel del producto. ....	39
Figura 3.8. Etapa de inicio del brazo robótico. ....	39
Figura 3.9. Etapa de traslado y clasificación de envases.....	40
Figura 3.10. Rango de valores del sensor de luz. ....	41
Figura 3.11. Rango de Valores, Sensor Ultrasónico en nivel de producto Vacío. ....	42
Figura 3.12. Rango de Valores, Sensor Ultrasónico en nivel de producto Medio.....	44
Figura 3.13. Rango de Valores, Sensor Ultrasónico en nivel de producto Lleno.....	45

## ÍNDICE DE TABLAS

Tabla 3.1. Tabla de elementos y funciones.....	30
Tabla 3.2. Valores medidos del sensor ultrasónico.....	42
Tabla 3.3. Resultados de la evaluación técnica .....	47
Tabla 3.4. Costos del proyecto. ....	48

## **Introducción**

En estos tiempos en la que la tecnología está sustituyendo la mano de obra humana mediante robots, los cuales realizan los mismos trabajos del hombre en las industrias, incrementando la producción en poco tiempo y evitando a la vez cierto porcentaje de gastos económicos que ello implica; debido a estos avances tecnológicos cada vez las empresas han optado por implementar sistemas autómatas que aseguren al cliente un producto con la mejor calidad.

El presente proyecto tiene como objetivo principal el diseño de un prototipo de proceso industrial para la clasificación de objetos con la utilización de un kit de robótica Lego Mindstorms NXT empleando librerías del software Matlab y Simulink, cuyo proceso se trata de un diseño y construcción de una banda transportadora y un brazo robótico, en el cual el cerebro del proceso es un bloque de programación Lego, el mismo que permite el control de sensores y servomotores conectados a los puertos del bloque y programados con Matlab. Para el control de la banda transportadora se utiliza el software NXT-G que es el programa que viene por default en el Kit ya mencionado. El Kit Lego Mindstorms consta de 577 piezas que permiten armar cualquier tipo de robot y programarlo de acuerdo a la necesidad, desde un entorno de programación básico hasta un entorno de ingeniería.

El proyecto se basa fundamentalmente en desarrollar un prototipo de brazo robótico con sus respectivos sensores que permita clasificar objetos con características diferentes, en este caso se realizará la clasificación de envases dependiendo el nivel del producto que contenga.

### **Objetivos**

#### **Objetivo general**

- Diseñar un prototipo de proceso industrial para la clasificación de envases según el nivel de producto con la utilización de un kit de robótica Lego Mindstorms NXT y herramientas de programación como Matlab y Simulink.

## **Objetivos específicos**

- Investigar sistemas que realicen procesos similares que permitan tener como base para la creación de la estructura mecánica del sistema.
- Distribuir de forma estratégica los distintos sensores a utilizar en el sistema para que puedan dar una buena lectura al procesador central.
- Encontrar un lenguaje de programación alternativo al original, el cual permita cubrir con las necesidades del sistema de clasificación de objetos.
- Realizar la programación necesaria del bloque Mindstorms NXT con la utilización de Matlab y Simulink mismo que servirá para el control de calidad y clasificación de objetos en determinado proceso.

Para cumplir los objetivos planteados se utilizará métodos científicos como:

En la primera etapa de la investigación de otros sistemas similares se utilizará el método Inductivo que permitirá recoger datos actuales de una infraestructura a partir de las instalaciones y equipos existentes.

En la segunda etapa la de distribución de los sensores se utilizará el método sistemático el cual permitirá modelar la estructura mecánica montada mediante la utilización de sus componentes activos y pasivos.

En la tercera etapa la de experimentación de lenguajes de programación se utilizará el método Deductivo que permitirá la utilización de leyes, teorías, reglas, conceptos y principios para llegar a determinar los componentes idóneos para el buen funcionamiento del bloque y el sistema en su totalidad.

En la cuarta etapa la de programación del sistema se utilizará el método Heurístico para generar una nueva metodología a partir de las experiencias con el software Matlab y Simulink a utilizarse, con el fin de detectar objetos y poder clasificarlos sin fallas de esta manera se podrá acercarse a la solución del problema de control de calidad.

# CAPÍTULO I

## FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

La tecnología ha sido un proceso acumulativo clave en la experiencia humana. Es así que desde hace mucho tiempo atrás el hombre ha venido experimentando avances tecnológicos significativos, gracias a la manipulación de dispositivos y elementos con los cuales puede armar proyectos antes de lanzar al mercado para su consumo.

En el mercado existe varios kits de robótica para entrenamiento de personas aficionadas a la rama, uno de esos kits es el Lego Mindstorms NXT el cual con sus 577 piezas que dispone se puede armar y programar cualquier prototipo de robot, solo hace falta la imaginación.

En este capítulo se describe las características de las piezas y dispositivos que consta el kit en forma general, sus aplicaciones y funcionamiento, el lenguaje de programación con el que se va a desarrollar el proyecto y los dispositivos adicionales para la estructura mecánica del mismo.

### 1.2. Lego Mindstorms NXT

El kit Lego Mindstorms es una herramienta de entrenamiento para todas las edades el cual contiene elementos básicos de las teorías de la robótica. Este kit surge de la fructífera colaboración entre Lego y el MIT (Instituto Tecnológico de Massachusetts por sus siglas en inglés) la primera creación de esta unión salió al mercado en septiembre de 1998.

Este kit de robótica programable NXT surge de la evolución de la versión del RCX (kit creado en el año de 1998) anterior a este, el cual permite construir y programar soluciones robóticas de la vida real, donde el objetivo no sólo es crear estructuras robóticas, sino programarlas y darles vida para que realicen distintas acciones. Hoy en día Lego Mindstorms está entrando en los diferentes colegios, universidades e institutos para

sus clases de robótica; a la vez que posee partes mecánicas, sensores y actuadores que permiten al estudiante desarrollar los conocimientos en el área de robótica.

En la figura 1.2.1. se puede observar el cartón que contiene todas las piezas y elementos que consta el kit de robótica Lego Mindstorms NXT.



Figura 1.1. Presentación del kit Lego Mindstorms NXT  
Fuente: (Parker, 2009)

### 1.2.1. Componentes del kit.

Específicamente es el producto Lego número 8527 conocido como el Lego Mindstorms NXT.

Este incluye:

- 1 Bloque programable (brick NXT).
- 1 Sensor de luz.
- 1 Sensor Ultrasónico.
- 1 Sensor Táctil.
- 1 Sensor de Sonido.
- 3 Servomotores.
- 577 Piezas Lego Technic.
- 1 CD con el software de programación e instrucciones para construir 4 robots (compatible con Windows y Mac).

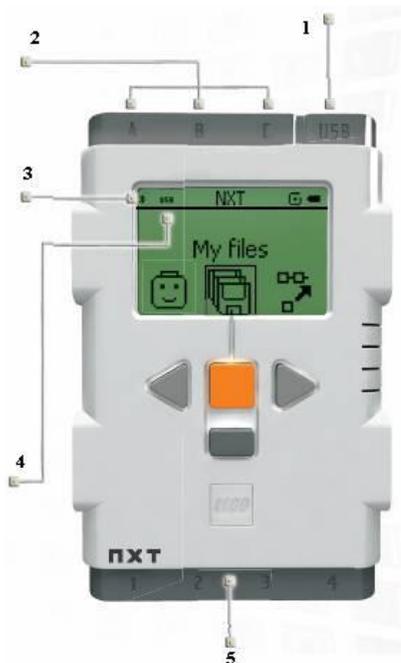
### 1.2.1.1. Bloque programable NXT

El bloque NXT es el cerebro de un robot MINDSTORMS el cual puede ser controlado por computadora, el ladrillo de Lego permite que el robot cobre vida y realice diferentes operaciones.

“En el bloque NXT se encuentra un total de 8 conectores muy parecidos (pero no iguales) a los que se usan en los teléfonos o conexiones de red de datos. En este caso los conectores tienen seis conexiones cada uno y nos permiten conectar 3 servomotores, 4 sensores y un enlace por USB para controlar y descargar programas. Adicionalmente esta función se puede realizar mediante una conexión inalámbrica de tipo bluetooth”. (EducaMadrid, 2011).

Actualmente la evolución de este kit ha llegado al mercado hasta la versión 3 (Lego Mindstorms EV3).

En la figura 1.2. se puede observar el bloque programable NXT con sus respectivos puertos y componentes.



1. Puerto USB
2. Puertos de salida
3. Icono de Bluetooth
4. Icono de USB
5. Conexión al adaptador de corriente



1. Icono de funcionamiento
2. Nivel de batería
3. Altavoz
4. Botones del NXT
5. Puertos de entrada

Figura 1.2. Características del bloque Mindstorms NXT  
Fuente: (Marcos, 2010)

### Características del microcontrolador:

- Ladrillo inteligente NXT:
  - Microcontrolador ARM7 de 32-bit.
  - 256 Kbytes FLASH, 64 Kbytes RAM.
  - Microcontrolador AVR de 8-bit.
  - 4 Kbytes FLASH, 512 Byte RAM.
  - Comunicación inalámbrica Bluetooth que permite comunicarlo con ordenadores, teléfonos móviles o PDAs (alcance de al menos 10 metros).
  - Puerto USB 2.0.
  - 4 puertos de entrada.
  - 3 puertos de salida.
  - Display de 60 x 100 pixels.
  - Altavoz, calidad de sonido de 8 KHz.
  - Alimentación, 6 pilas AA.

En la figura 1.3. se puede observar el diagrama de bloques de los componentes internos del bloque NXT con sus respectivos periféricos.

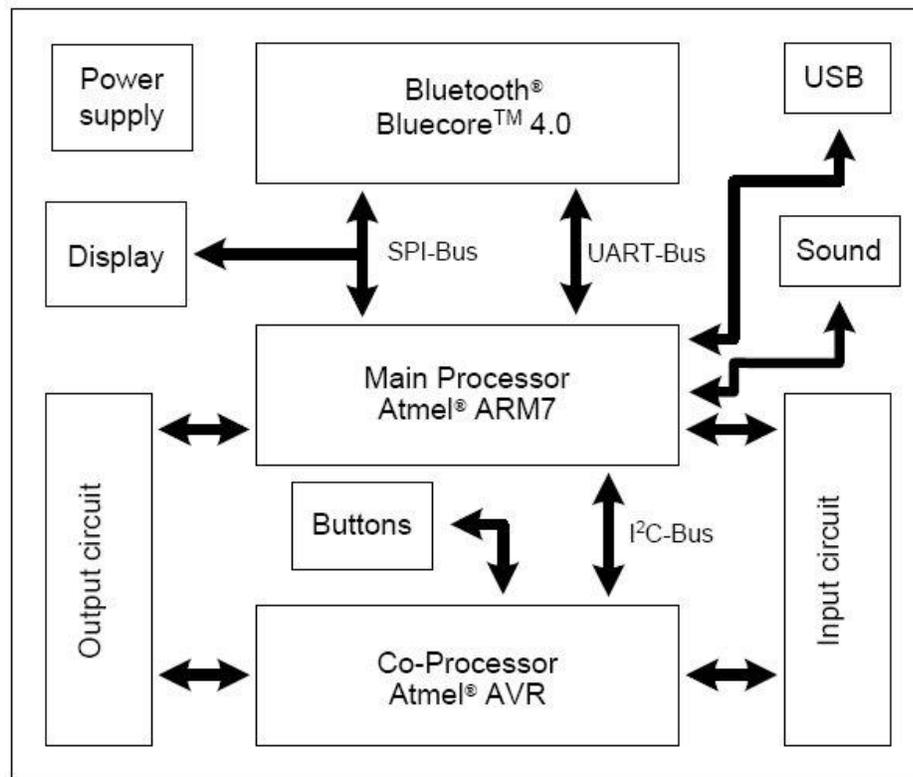


Figura 1.3. Diagrama de bloques del Mindstorms NXT  
Fuente: (EducaMadrid, 2011)

## Comunicaciones

“La comunicación del bloque NXT con el computador es mediante la interfaz de USB que posee, la cual ya viene en la versión 2.0. Además, para comunicarse con otros robots en las cercanías posee una interfaz Bluetooth que es compatible con el Clase II v 2.0. Esta conectividad con Bluetooth no tan sólo permite conectarse con otros bloques, sino también con computadores, palms, teléfonos móviles, y otros aparatos con esta interfaz de comunicación.” (School, 2009).

Dentro de las posibilidades de conexión se encuentran:

- Se pueden conectar hasta tres dispositivos NXT distintos.
- Realizar búsquedas y conectarse a otros dispositivos que tengan Bluetooth.

- Guardar dispositivos con los cuales emparejado anteriormente para conectarse más rápidamente.
- Configurar el bloque NXT para que este en estado visible o invisible con el resto de los dispositivos que tengan Bluetooth.

#### **1.2.1.2. Sensor de Luz**

Cualquier sensor de luz detecta la iluminancia en objetos o ambiente, estos miden la cantidad de luz que receipta una célula foto-eléctrica (básicamente una resistencia). La resistencia con presencia de luz es baja y alta con oscuridad. Se usan para medir intensidad, orientarse, protegerse, enfocarse. Su colocación influye mucho debido a que el sensor debe estar ubicado a una distancia prudente, así como también debe ser calibrado dependiendo la luz exterior en la que esté trabajando para que pueda tener una buena lectura. En conclusión un sensor fotoeléctrico (luz) es un dispositivo electrónico que responde a la variación de la intensidad de la luz. Los sensores necesitan de un dispositivo emisor que genera la luz, y otro dispositivo receptor que “ve” la luz generada por el emisor. Los diferentes modos o formas de sensado se basan en este principio de funcionamiento. Estos sensores están diseñados especialmente para la detección, clasificación y posicionamiento de objetos; la detección de colores, formas y diferencias de superficie, incluso su trabajo se realiza bajo condiciones ambientales extremas.

“El sensor de luz Lego Mindstorms permite tomar una muestra de luz de un objeto; en un lado trae un conductor eléctrico y por el otro lado una cámara oscura que capta las luces. Este receptor es capaz de captar intensidad de luz entre los rangos de 0,6 a 760 lux. Este valor es considerado como un porcentaje de iluminación, el cual es procesado por el bloque lógico programable, obteniendo un porcentaje aproximado de luminosidad.” (EducaMadrid, 2011).

En la figura 1.4. se puede observar el sensor de luz y las escalas en las cuales puede trabajar el mismo.



Figura 1.4. Sensor de luz  
Fuente: (EducaMadrid, 2011)

Este sensor capta grados de luminosidad, por lo que no es capaz de diferenciar colores, sólo capta la existencia del blanco y negro y los tonos de grises que están entre estos dos colores los mismos corresponden a los distintos porcentajes de luz presentes en el medio.

### 1.2.1.3. Sensor Ultrasónico

“Antes de la explicación de este sensor se debe tener en cuenta que los ultrasonidos son sonidos, exactamente igual que los que se escucha normalmente, salvo que tiene una frecuencia mayor que la máxima que puede ser captada por el oído humano. Esta comienza desde unos 16 Hz hasta un límite superior de aproximadamente 20 KHz, mientras que los ultrasonidos utilizan una frecuencia de 40 KHz que están presentes en los sensores ultrasónicos.” (Perez, 2012)

El funcionamiento básico de los sensores ultrasónicos como medidores de distancia se puede observar de una manera muy clara en la figura 1.5, en donde hay un receptor el cual emite un pulso de ultrasonido el mismo rebota sobre un determinado objeto que se encuentre en el rango del sensor y la reflexión de ese pulso es detectada por un receptor de ultrasonidos.

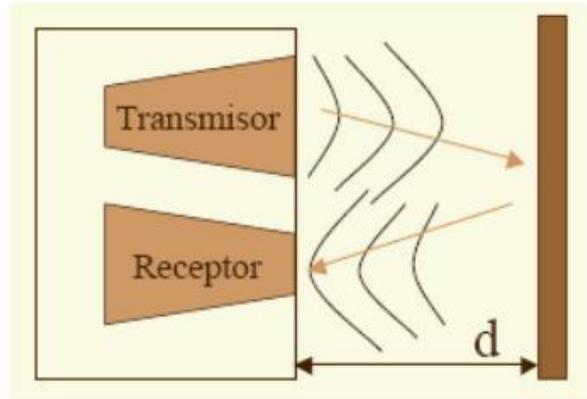


Figura 1.5. Funcionamiento del sensor de ultrasonido  
Fuente: (Perez, 2012)

La mayoría de los sensores ultrasónicos se basa en la emisión de un pulso de ultrasonido cuyo lóbulo, o campo de acción, es de forma cónica. Con la medición del tiempo que transcurre entre la emisión del sonido y la percepción del eco, se puede determinar la distancia a la que se encuentra el objeto que ha producido la reflexión de la onda sonora, mediante la fórmula:

$$d = \frac{1}{2} V \cdot t$$

En donde  $V$  es la velocidad del sonido que se propaga en el aire (343 m/s), y  $t$  viene a ser el tiempo transcurrido entre la emisión y recepción del pulso.

El sensor ultrasónico Lego Mindstorms puede detectar objetos que estén de 0 a 255 centímetros de distancia con una precisión de +/-3 cm, significa que si existe un objeto a 40 cm exactamente el sensor puede dar una lectura desde 37 hasta 43cm, este es su margen de error.

Este sensor funciona de la misma forma como lo hace un radar o un sonar, enviando ondas de sonido a una frecuencia muy alta y midiendo el tiempo que el "eco" (el rebote de las ondas) de sonido demora en regresar al sensor. Esto es muy idéntico a como un murciélago "ve".

También se puede utilizar para hacer que el robot evite obstáculos, al sentido y medida a distancia, y detecte el movimiento.



Figura 1.6. Sensor Ultrasonico  
Fuente: (EducaMadrid, 2011)

#### 1.2.1.4. Sensor de Tacto

El sensor de tacto permite detectar si el mismo ha colisionado o no con algún obstáculo que se halle en su trayectoria inmediata. Al tocar una superficie un pequeño pulsador externo se contrae, permitiendo que un switch dentro del bloque cierre un circuito eléctrico y comience a circular energía, provocando una variación de voltaje de 0 a 5 Vdc.

En el caso anterior, si la presión supera una medida estándar de 450, la cual se puede observar en la pantalla de LCD, se considera que el sensor está presionado (activado), de otro modo, se considera que está sin presión (desactivado).

En la figura 1.6 se muestra el sensor de tacto del Lego Mindstorms.



Figura 1.7. Sensor de Tacto  
Fuente: (EducaMadrid, 2011)

#### 1.2.1.5. Sensor de Sonido

Este sensor lee el sonido ambiental y entrega una medida de 0 a 100%. Se puede configurar para que el sensor lea Decibeles o Decibeles Ajustados. En términos muy concretos los decibeles ajustados solo son sonidos que el oído humano puede escuchar, a diferencia de los decibeles normales que incluye frecuencias que no se puede escuchar pero que el sensor de sonido capta.

Configurar el sensor de sonido en modo dBA es más complicado de lo que debería hacer. Lo que se tiene que hacer es conectar una variable lógica puesta en TRUE (verdadero) al conector dBA del sensor de sonido.

“Este sensor puede medir la presión del sonido por encima de los 90 dB. Las lecturas se despliegan en el Lego Mindstorms en forma de porcentaje: 4-5 % es el silencio de un dormitorio, 5 – 10 % es el sonido de una persona hablando cierta distancia. De 10 a 30 % es el nivel de una conversación cercana normal o de música a un volumen normal y del 30 al 100 % representa volúmenes altos de conversación o de música. Estos rangos se asumen a una distancia de 1 metro entre la fuente de sonido y el sensor” (Marcos, 2010)

En la figura 1.7. se puede observar el sensor de sonido incluido en el kit de robótica Lego Mindstorms NXT.



Figura 1.8. Sensor de Sonido  
Fuente: (EducaMadrid, 2011)

#### 1.2.1.6. Servomotores

Un Servo es un dispositivo pequeño que tiene un eje de rendimiento controlado. Este puede ser llevado a posiciones angulares específicas al enviar una señal codificada. Con tal de que una señal codificada exista en la línea de entrada, el servo mantendrá la posición angular del engranaje. Cuando la señal codificada cambia, la posición angular de los piñones cambia. En la práctica, se usan servos para posicionar superficies de control como el movimiento de palancas, pequeños ascensores y timones. Ellos también se usan en radio control, títeres, y por supuesto, en robots.

El motor del servo tiene algunos circuitos de control y un potenciómetro (una resistencia variable) esta es conectada al eje central del servo motor. En la figura se puede observar al lado derecho del circuito. Este potenciómetro permite a la circuitería de control,

supervisar el ángulo actual del servo motor. Si el eje está en el ángulo correcto, entonces el motor está apagado. Si el circuito chequea que el ángulo no es el correcto, el motor girará en la dirección adecuada hasta llegar al ángulo correcto. El eje del servo es capaz de llegar alrededor de los 180 grados. Normalmente, en algunos llega a los 210 grados, pero varía según el fabricante. Un servo normal se usa para controlar un movimiento angular de entre 0 y 180.

Los motores desmontables son alimentados mediante cables que poseen conductores eléctricos que transmiten la energía a los inductores. Como son motores paso a paso, el sentido de conexión no entrega la misma dirección de movimiento.

El modelo NXT usa servomotores, los cuales permiten la detección de giros de la rueda, indicando los giros completos o medios giros, que es controlado por el software.

Los servomotores del kit de robótica LEGO® MINDTORMS® son motores DC que incorporan un sensor de posición con un grado de resolución y una velocidad de rotación del eje de 170 rpm.

Estos motores se conectan al bloque programable a través de los puertos A, B y C. Los tres motores pueden estar conectados al bloque programable y usarse al mismo tiempo.

Los servos además de incluir un motor eléctrico convencional también incluyen un sensor de posición. Este sensor permite saber a qué velocidad se está moviendo el motor, y corregirla si es necesario. Además se puede saber exactamente cuántos grados han girado el motor en todo momento. Con esto se tiene un control muy preciso del movimiento del robot.

El sensor de posición además de servir para controlar la velocidad y avance del robot también es útil en sí mismo y permite usar los motores del NXT como sensores de movimiento.

En la figura 1.8. se puede observar el servomotor que incluye en el kit de robótica Lego Mindstorms NXT, en total son 3 servomotores.



Figura 1.9. Servomotor  
Fuente: (EducaMadrid, 2011)

### **1.2.2. Herramientas de Software**

El proyecto consta de dos bloques NXT, el primero se utiliza para armar la banda transportadora y utilizar el software de programación de Lego Mindstorms (NXT-G), mientras que el segundo bloque se utiliza para el brazo robótico el mismo usa las herramientas de Matlab para la programación y adquisición de datos en tiempo real. En las dos etapas se usa los elementos incluidos en el kit.

#### **1.2.2.1. Programación de Lego Mindstorms NXT**

La programación del Lego Mindstorms se realiza a través del software que se adjunta en el empaque original, este trae el firmware del robot V1.0 y un programa que emula un árbol de decisiones, para los cuales, el usuario debe programar las acciones para que ejecute el robot. En el software se encuentran modelos pre establecidos para cada modelo de robot que puede ser construido y que viene recomendado en el empaque.

Una de las principales características de este software de programación, es su entorno visual (similar a LabVIEW), el cual mediante el posicionamiento y configuración de bloques da la posibilidad a cualquier usuario aprendiz acostumbrarse rápidamente a la programación en el NXT-G.

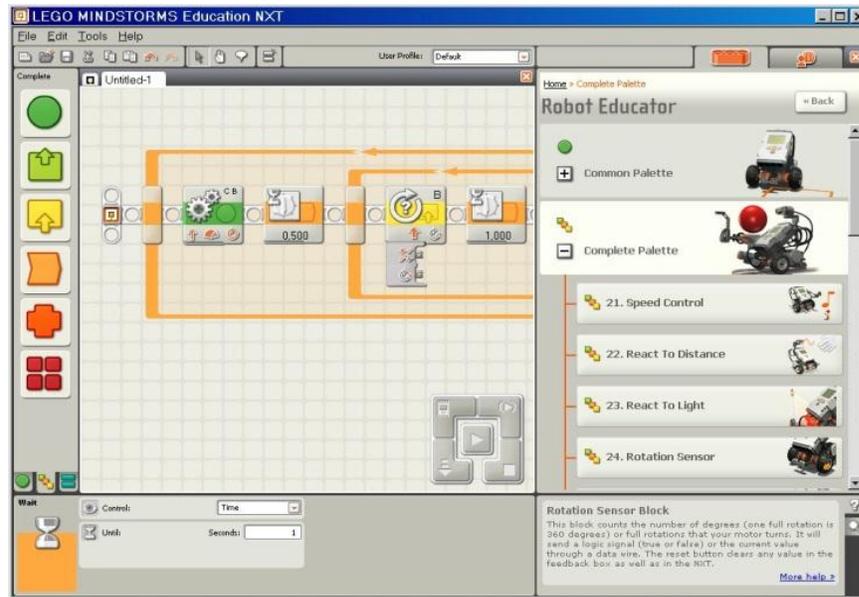


Figura 1.10. Entorno software NXT-G  
Fuente: (EducaMadrid, 2011)

Este lenguaje permite las instrucciones secuenciales, instrucciones de ciclos e instrucciones de decisiones, éstas últimas, basadas en los datos reportados por los sensores que se puede añadir al robot.

### 1.2.2.2. Matlab y Simulink

Es un ambiente de cómputo de alta ejecución numérica y de visualización. MATLAB integra el análisis numérico, cálculo de matrices, procesamiento de señales, diseño de sistemas de potencia, mapeo y tratamiento de imágenes, instrumentación y adquisición de datos, identificación de sistemas, traficación, entre otras aplicaciones en un ambiente sencillo de utilizar, donde los problemas y sus soluciones son expresadas justamente como están escritas; a diferencia de la programación tradicional. Permite resolver problemas en una fracción de tiempo.

Integra:

- Análisis numérico
- Cálculo matricial
- Procesamiento de señales

- Gráficos

MATLAB, también cuenta con algunas familias de soluciones para aplicaciones específicas llamadas TOOLBOX, que son colecciones de bloques y funciones utilizadas para resolver algún problema en particular como:

- Procesamiento de señales
- Diseño de sistemas de control
- Simulación de sistemas dinámicos
- Identificación de sistemas
- Redes neuronales y otros.

Probablemente la característica más importante de MATLAB es su capacidad de crecimiento. Esto permite convertir al usuario en un autor contribuyente, creando sus propias aplicaciones. En resumen, las prestaciones más importantes de MATLAB son:

- Escritura del programa en lenguaje matemático.
- Implementación de las matrices como elemento básico del lenguaje, lo que permite una gran reducción del código, al no necesitar implementar el cálculo matricial.
- Implementación de aritmética compleja.
- Un gran contenido de órdenes específicas, agrupadas en TOOLBOX.
- Posibilidad de ampliar y adaptar el lenguaje, mediante ficheros de script y funciones .m.

SIMULINK es un software que funciona bajo la plataforma de Matlab y es una herramienta muy útil para modelar, simular y analizar sistemas, tanto lineales como no lineales. Permite al usuario realizar sus estudios tanto en el dominio del tiempo como el de Laplace, expresar las funciones de transferencia en las diferentes formas incluyendo la del espacio de los estados y otras opciones. En una interfaz gráfica (GUI) como la que se observa en la Figura 1.11, el usuario construye un diagrama de bloques que desarrollan procedimientos que realizan las operaciones matemáticas requeridas para la solución de un modelo.

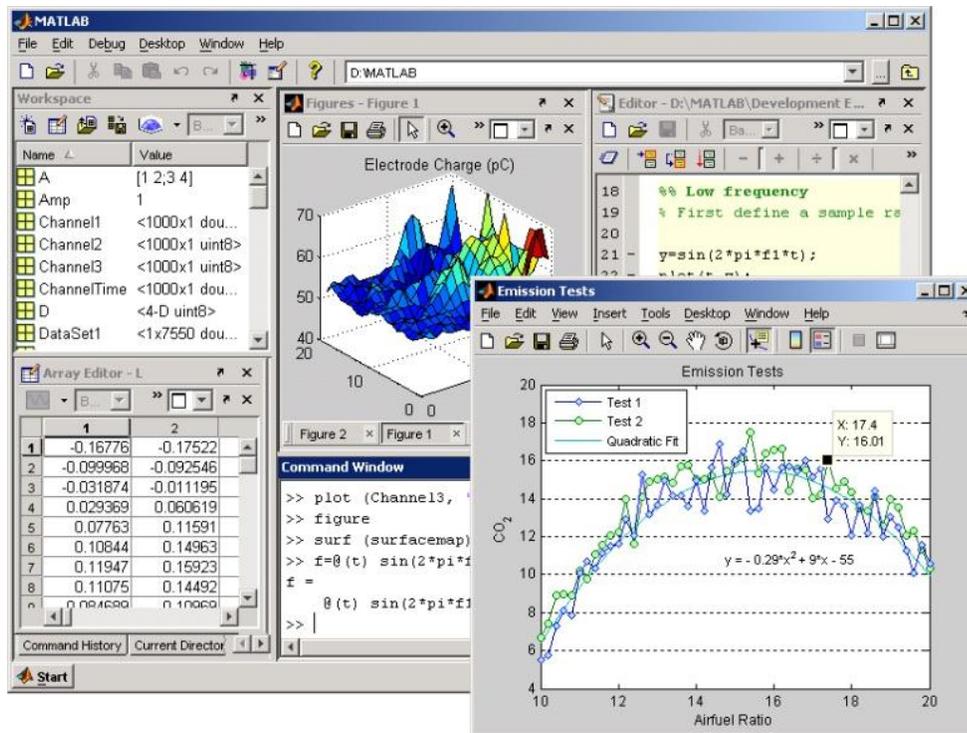


Figura 1.11. Entorno software Matlab  
Fuente: (Cohen, 2014)

## CAPITULO II

### DIAGNÓSTICO Y BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO

#### 2.1. Introducción

En este capítulo, se realiza un análisis del problema, describe las funciones específicas de los sensores de luz y ultrasonido que son los que van a realizar la parte de sensado del sistema, además se describe la posición real de todos los elementos que forman parte del proyecto.

También se describe la función del brazo robótico Lego Mindstorms y los sensores que actúan conjuntamente con el mismo.

Utilizando el principio del llenado de botellas, se diseña una banda transportadora y un brazo robótico capaz de seleccionar envases completamente llenos de otros que no contengan el nivel exacto de líquido, esto puede ser utilizado en el control de calidad para un proceso de llenado de envases. La programación se realiza con las herramientas de Lego Mindstorms, el cual funcionan bajo la plataforma de Labview y Matlab.

#### 2.2. Fundamentación de la construcción

Una vez decidido el proyecto práctico a realizarse y haber escogido desarrollar un prototipo de brazo robótico con sensores que permita clasificar objetos con características diferentes más una banda transportadora que traslade los mismos, es necesario fundamentar y justificar su estructura y características con la que cuenta el sistema tanto para poder desplazar los envases como para la ubicación de los mismos. De igual manera se cita las ventajas de utilizar el Kit Lego Mindstorms para el proyecto.

¿Por qué utilizar Lego Mindstorms para realizar el proyecto?

- Fácil de Armar y Desarmar, no es necesario usar soldadura, ni tornillo. Todo lo que se arma se puede desarmar rápidamente.

- Permite usar las piezas en múltiples diseños distintos.
- Muy extendido por todo el mundo, lo que permite encontrar gran cantidad de información e ideas por Internet, diseños y soluciones.
- No es un pack cerrado, es decir, se puede comprar más ampliaciones de lego, adquirir piezas deterioradas o perdidas, o añadir piezas realizadas manualmente, como por ejemplo, sensores o motores, e incluso circuitos neumáticos.
- Múltiples posibilidades y lenguajes de programación, desde el nivel más básico e intuitivo, uso de lenguajes conocidos como C, Java, LabVIEW o Matlab, utilización de Linux, además podemos programar en lenguaje gráfico.
- Indicado para entornos educativos, desde colegios a universidades, pues se puede aprender de forma fácil tanto mecánica como electrónica.

A continuación se describe el análisis de hardware y software utilizado para el funcionamiento del sistema.

### **2.3. Análisis de la estructura**

El proyecto consiste en la construcción de una banda transportadora con un brazo robótico, los cuales permitan trasladar un recipiente a través de la banda y colocar los mismos en el lugar indicado dependiendo del nivel de llenado de los mismos.

El movimiento de la banda está a cargo del servomotor Lego NXT, por la banda circulan recipientes llenados en diferentes niveles del producto, para que la banda se mueva debe detectar un pulso del sensor de tacto, una vez que detecta este pulso la banda se mueve hacia la derecha transportando el envase, cuando el recipiente es detectado por el sensor de luz (2) la banda se detiene, el sensor de luz (1) le comunica al sensor ultrasónico (1) que tiene que proceder con el sensado de nivel del producto, una vez que obtiene la lectura, el recipiente avanza hasta que el sensor ultrasónico (2) detecta el objeto y detiene el movimiento de la banda hasta que el brazo robótico actúe.

En la figura 2.1 se aprecia el modelo mecánico de la banda transportadora y la ubicación de los sensores en la misma.

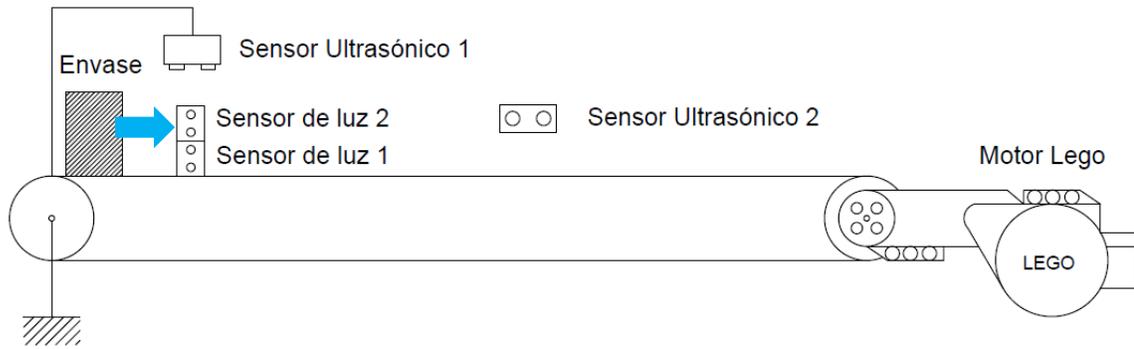


Figura 2.1. Diseño de la banda transportadora  
Fuente: Investigador (ACAD 2012)

La banda transportadora es controlada por un bloque Lego Mindstorms el mismo tiene como dispositivos de entrada tres sensores (luz 2, ultrasónico 2, y tacto), y un dispositivo de salida que es el servomotor. Todos los elementos mencionados anteriormente funcionan con el software Lego NXT-G.

Una vez realizadas estas funciones de los sensores, el brazo actúa para coger el recipiente.

El brazo robótico clasifica los recipientes según el nivel medido, el mismo tiene 2 grados de libertad; el brazo utiliza 2 servomotores Lego Mindstorms: Motor A para mover la base de brazo es decir en forma horizontal, Motor B para el movimiento vertical del brazo, este sirve para acercar el brazo a la altura del envase, y el Motor C que controla el gipper o tenaza.

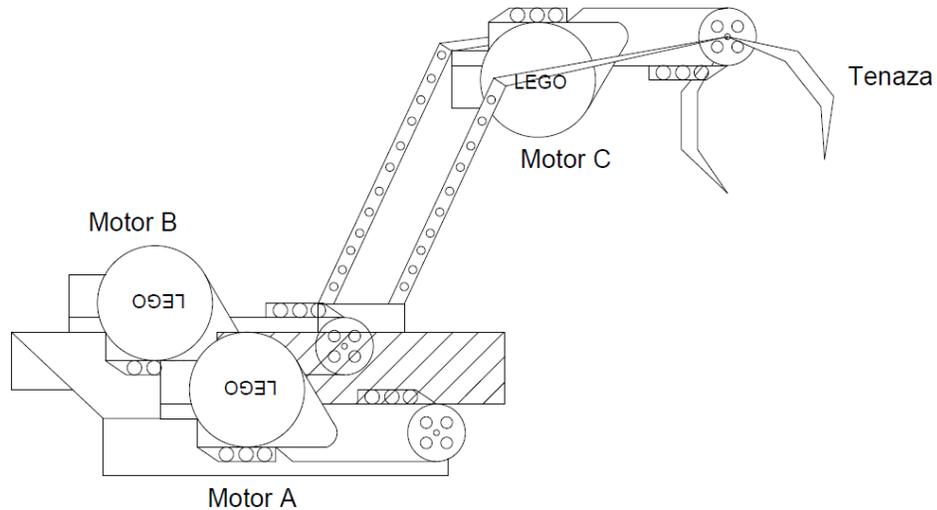


Figura 2.2. Diseño del brazo robótico  
Fuente: Investigador (ACAD 2012)

La programación para que el brazo robótico Lego Mindstorms realice la secuencia y adquisición de datos se desarrolló con el software MATLAB R2013b.

## 2.4. Análisis del Software

### 2.4.1. Software para la banda transportadora

La programación para la banda transportadora está realizada en el software original de Lego Mindstorms NXT, el NXT-G el cual puede trabajar en un ambiente gráfico similar al de LabVIEW, el mismo permite controlar el motor Lego de la banda transportadora, el sensor de luz (2), el sensor ultrasónico (2) y un sensor de tacto descritos en la figura 2.1.

La programación en este lenguaje está descrita en el capítulo III del mismo documento.

### 2.4.2. Software para el brazo robótico.

La programación para el brazo robótico está realizada en MATLAB R2013b que es un entorno de cálculo técnico de altas prestaciones, para cálculo numérico y visualización.

“Los robots LEGO MINDSTORMS NXT son una serie de robots adaptables y programables que son controlados por una unidad de microprocesador, referida como el

ladrillo NXT. A través de la programación del ladrillo NXT, se puede hacer que el robot realice una serie de acciones diferentes, como moverse hacia adelante, hacia atrás o recoger objetos. Habitualmente, un robot NXT está programado manualmente, a través de la pantalla de ladrillo NXT o mediante la aplicación LEGO Mindstorms NXT. Sin embargo, la aplicación MathWorks de MATLAB, tiene una extensión integrada diseñada para la programación del NXT llamada RWTH Aachen MINDSTORMS NXT Toolbox.” (AACHEN.RWTH, 2009).

Para poder trabajar en el entorno de MATLAB y SIMULINK hay que proceder a instalar algunas herramientas de software, entre ellas esta los siguientes programas:

### **2.4.3. RWTH - Mindstorms NXT Toolbox**

La RWTH - Mindstorms NXT Toolbox para MATLAB es una caja de herramientas que ha sido desarrollado para el control de robots LEGO ® Mindstorms NXT con MATLAB a través de una interfaz USB o una conexión Bluetooth inalámbrica y proporciona funciones de MATLAB para interactuar con un robot directamente. Los comandos y sus características se pueden encontrar en la ayuda del MATLAB.

Este concepto de comunicación entre PC y el bloque NXT permite combinar aplicaciones de los robots con las operaciones matemáticas complejas y visualizaciones en MATLAB.

¿Por qué utilizar Matlab y RWTH - Mindstorms NXT Toolbox para realizar la programación?

Las ventajas del uso de la RWTH - Mindstorms NXT Toolbox en comparación con los lenguajes de programación que operen directamente en el ladrillo NXT) son:

- Enorme aumento de la potencia de la CPU y la memoria disponible.
- Tamaño del programa prácticamente ilimitada (en comparación con los programas de NXT clásicos).

- Control de múltiples robots desde dentro de un solo programa (sólo limitado por el número de adaptadores Bluetooth instalados al mismo tiempo).
- Ejecutar programas de MATLAB remotos para observar sensores, etc, mientras que los programas clásicos de NXT están ejecutando al mismo tiempo.
- El uso de hardware adicional común, por ejemplo, webcams, joysticks, etc.
- Las funciones avanzadas de depuración son totalmente compatibles, es decir, puntos de interrupción, ejecuciones paso a paso, inspección de variables en marcha.
- Disfrutar de las ventajas de las redes e Internet en las aplicaciones de los robots.
- Las capacidades 2D y 3D de visualización impresionante y bien documentados para la representación de datos.
- Programas, cajas de herramientas o bibliotecas de uso existentes, por ejemplo, redes neuronales, procesamiento de imágenes, bases de datos.

Esta caja de herramientas abre posibilidades ilimitadas para proporcionar la inteligencia artificial de los robots Lego y otras mejoras al usar las características múltiples de MATLAB y los cálculos para el proceso de señal numérica.

Para poder trabajar en el entorno de Matlab y Lego Mindstorms se necesita de ciertos requisitos del Sistema Operativo que estemos utilizando.

Vista general de los requisitos del sistema de la caja de herramientas:

- Sistema operativo: Windows, Linux o Mac OS.
- MATLAB versión 7.7 (R2008b) o superior.
- Kit de construcción de LEGO ® Mindstorms NXT (por ejemplo, Kit Educación).
- LEGO ® Mindstorms NXT v1.26 firmware o compatible (mínimo)  
LEGO ® Mindstorms NXT v1.28 firmware o superior (se recomienda).
- Bluetooth 2.0 Adaptador modelo recomendado por LEGO ® (por ejemplo, AVM BlueFRITZ! USB) que soporta el perfil de puerto serie (SPP).
- Para Windows 64 bits que necesita Matlab2011b o superior con los controladores de la caja de herramientas V3.0 o superior.
- Cable USB y libusb\_win32 (<http://www.libusb.org/wiki/libusb-win32>) para Windows (32 bits y 64 bits) o LEGO Mindstorms NXT Fantom controlador USB para

Windows (32 bits) y Mac OS (<http://mindstorms.lego.com/en-us/support/files/Driver.aspx>), o en la biblioteca libusb de Linux.

Para poder realizar programaciones existen dos formas de trabajar con Lego Mindstorms y Matlab; la primera es descargando del internet e instalando los programas: Cygwin, GNU ARM, NXTOsek, NeXTTool.exe, ECRobot (Embedded Coder Robot NXT) todos estos individualmente, la segunda es descargar e instalar directamente desde el propio Matlab de la siguiente forma:

- Introducir en el Command Window: “>> supportPackageInstaller” este comando le enviara automáticamente a una ventana para descargar e instalación del paquete de soporte de Lego para Matlab. También se puede lograr lo mismo desde el Add-Ons como se puede observar en la figura 2.3. Para ello debe estar conectado al internet.
- Para que pueda reconocer Matlab al ladrillo NXT se debe actualizar el firmware introduciendo en el Command Window “>> targetupdater”, esto le envía a una ventana para actualizar al NXT.
- Es necesario igualmente el programa Nextool.exe o el BricxCC para ingresar a los archivos del bloque y poder copiar el soporte para control de motores Lego con Matlab (MotorControl22.rxe) a nuestro ladrillo, este archivo se encuentra en la carpeta RWTHMindstormsNXT/Tools/ MotorControl que hay que descargar de internet para tener acceso a todas las librerías y demos para probar nuestro Lego.
- Por ultimo para tener acceso a los archivos del RWTHMindstormsNXT/Tools hay que agregarle esta carpeta en la ubicación que está instalado el Matlab.

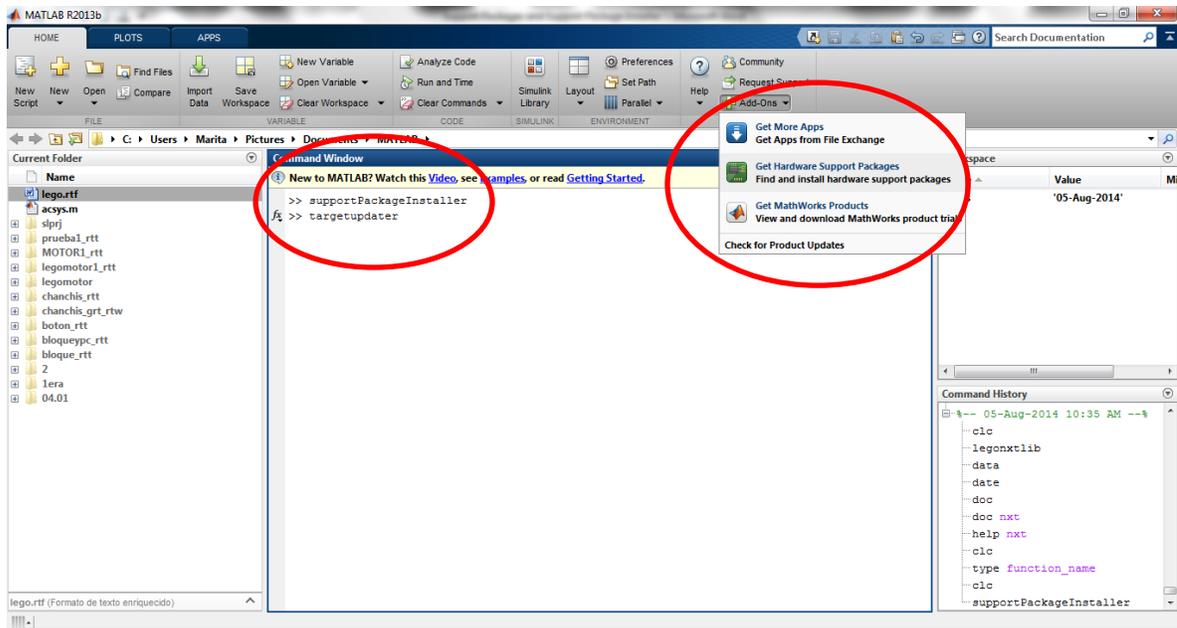


Figura 2.3. Interfaz Matlab, instalación del soporte de Lego para Matlab.  
Fuente: Investigador (Matlab R2013b).

En los dos casos hay que instalar el controlador USB (libusb\_win32), porque caso contrario se produce un error al correr programas desde Matlab hacia el Lego.

De esta forma se logra instalar la librería “RWTHMindstormsNXT” y los ejemplos “RWTH-MindstormsNXTExamples” en el toolbox de Matlab los cuales se descarga de la página oficial del programa. Esta librería adiciona bloques de Lego a Simulink. De esta forma la programación del Lego NXT se realiza desde Simulink, tomando en cuenta que Matlab utiliza indirectamente a las herramientas adicionales GNU ARM, Gygwin y Nexttool para compilar, cargar y ejecutar las instrucciones creadas desde el software.

En la figura 2.4 se observa el Simulink Support Package for LEGO MINDSTORMS NXT Hardware que es la librería cargada para Simulink para poder trabajar con el programa MATLAB y el Bloque NXT

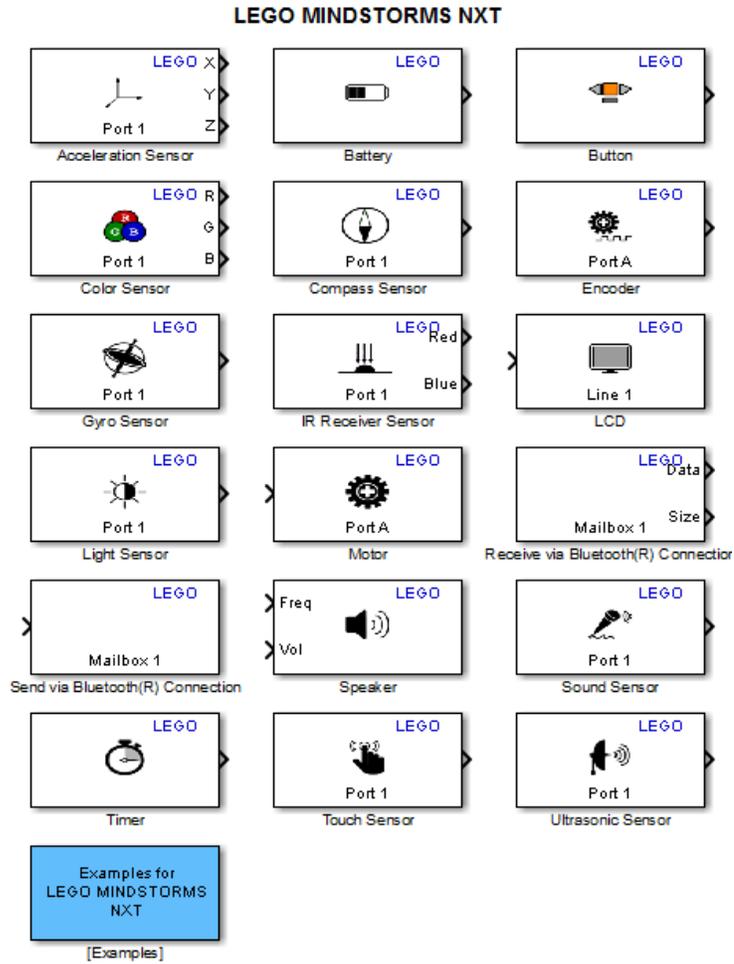


Figura 2.4. Librerías Lego NXT instaladas en el programa Matlab/Simulink  
Fuente: Investigador (Simulink Library Browser)

## 2.5. Comunicación entre PC vs NXT

El protocolo utilizado entre la PC y el NXT es USB de tercer tipo; esta interface es usada para la transferencia de datos con una alta velocidad (2.0), con una tasa de transferencia de hasta 480 Mbps (60 MB/s). También se puede conectar con la PC y otros bloques mediante Bluetooth v2.0.

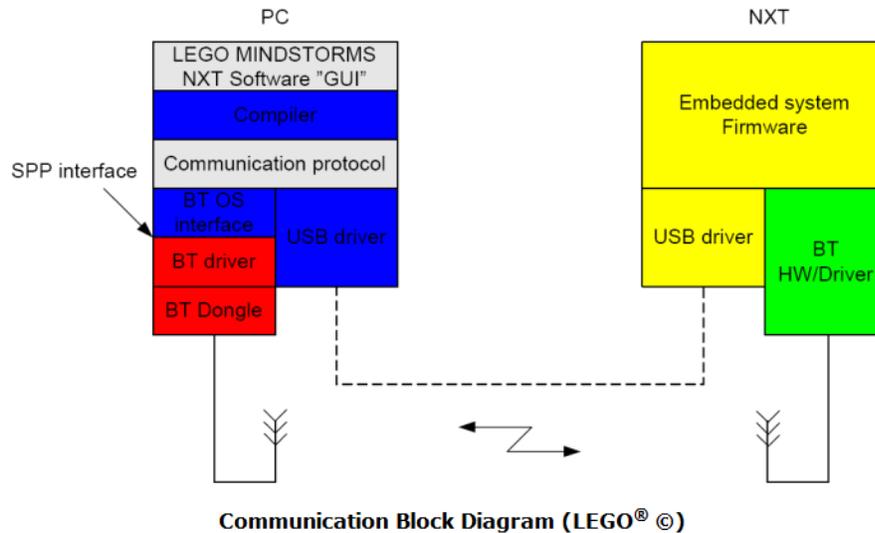


Figura 2.5. Diagrama de bloques de la comunicación PC vs NXT  
Fuente: Help RWTH - Mindstorms NXT Toolbox (Matlab R2013b)

### 2.5.1. Comunicación vía USB

El ladrillo inteligente NXT también se comunica a través de la interfaz USB y también puede ser programado para usar el protocolo de comunicación de LEGO® Mindstorms NXT. Las posibilidades de comunicación a través del dispositivo USB debe ser la misma a través del dispositivo Bluetooth. La comunicación USB V2.0 es compatible.

En el presente proyecto se utiliza este tipo de comunicación debido a que mediante bluetooth existe un leve retardo de presentación de resultados entre la PC y el programa que se está ejecutando en el ladrillo del Lego Mindstorms.

### 2.5.2. Comunicación vía Bluetooth

El ladrillo inteligente NXT soporta la comunicación inalámbrica mediante comunicación Bluetooth, v2.0 con EDR al incluir un chip CSR BlueCore 4 Versión 2. Es compatible con el perfil de puerto serie (SPP), que puede ser considerado como un puerto serie inalámbrico. El ladrillo NXT puede comunicarse con dispositivos Bluetooth que se programan para comunicarse con el LEGO Mindstorms NXT, y con los comandos del protocolo de comunicación que admitan el perfil de puerto serie (SPP). Es posible enviar y recibir información al ladrillo durante la ejecución del programa con ciertos retrasos

(30ms). Para reducir el consumo de energía utilizado por Bluetooth, la tecnología ha sido implementada por un dispositivo Bluetooth Clase II, lo que significa que puede comunicarse a una distancia de aproximadamente 10 metros.

## CAPITULO III

### PRESENTACIÓN DE RESULTADOS

#### 3.1. Introducción

En el presente capítulo se encuentra el diseño final del hardware, los diagramas de flujo de los programas y los algoritmos necesarios para el funcionamiento del proyecto. Todos los valores medidos por los sensores de luz y de ultrasonido son representados en un GUI (Interfaz Gráfica de Usuario) del software Matlab, de esta manera se puede observar el nivel del producto que contiene cada envase.

Además se encuentra descrito el código de programación para la lectura de los distintos sensores que se ocupan en el proyecto (anexos), así también el código para la adquisición de datos tomados por los sensores y representados en una pantalla en el software simulink.

#### 3.2. Diseño del Hardware del proyecto

En la tabla Nº 3.1 se observa un resumen de los elementos utilizados en el proyecto con sus respectivas funciones y entorno de programación con los cuales están trabajando.

Sistema	Elemento	Función	Entorno de Programación
Brazo mecánico	Sensor de luz 1 (L1)	Detección de objetos, inicio de sensado.	Matlab
Banda transportadora	Sensor de luz 2 (L2)	Detección de objetos, paro de banda.	NXT-G
Brazo mecánico	Sensor ultrasónico 1 (US1)	Sensado de nivel de producto.	Matlab
Banda transportadora	Sensor ultrasónico 2 (US2)	Detección de objetos, paro de banda.	NXT-G
Banda transportadora	Motor Lego 1	Movimiento de la banda.	NXT-G

Brazo mecánico	Motor A	Desplazamiento horizontal	Matlab
Brazo mecánico	Motor B	Desplazamiento vertical	Matlab
Brazo mecánico	Motor C	Movimiento garra	Matlab

Tabla 3.1. Tabla de elementos y funciones  
Fuente: Investigador

### 3.2.1. Diseño de la banda transportadora

El sensor de ultrasonido 1 (US1) se colocó en la parte superior de la banda y a la altura de los sensores de luz, de esta forma se logra medir la cantidad de producto contenida dentro de cada recipiente, así se aprovecha la función del sensor utilizándolo como sensor de nivel, mientras que el sensor de luz 1 (L1) sólo detecta la presencia de objetos, estos dos sensores están controlados por el Bloque NXT programado con Matlab. El sensor de luz 2 (L2) es el encargado de detener la banda transportadora al momento que llega a la área de sensado, el sensor ultrasónico 2 (US2) es encargado de detectar al envase y detenerlo una vez que haya sido sensado y transportado a lo largo de la banda, estos sensores son las entradas del NXT que controla la banda y programado con el lenguaje NXT-G; en la segunda etapa se detendrá la banda para hacer la mediciones de nivel, mientras que en la tercera etapa la banda se detendrá para que actué el brazo robótico para la clasificación de los recipientes. En la figura 3.1 se encuentra el diseño final de la banda.

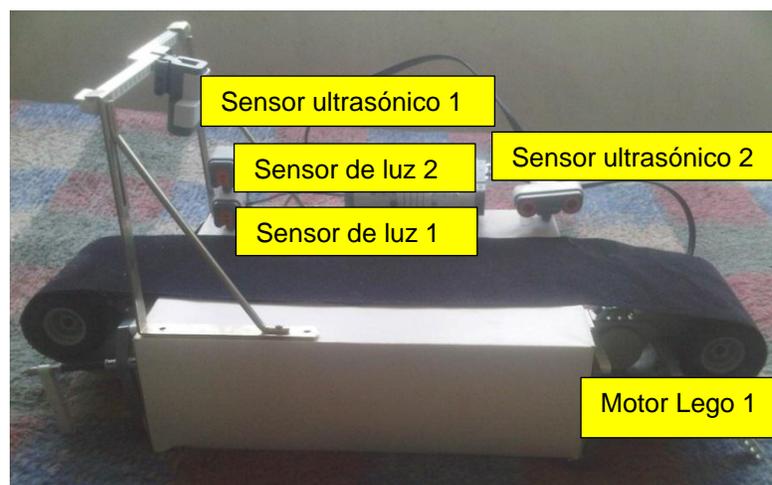


Figura 3.1. Diseño mecánico de la banda transportadora y de la ubicación de los sensores  
Fuente: Investigador

Con un solo motor Lego (1) se logra movilizar la banda transportadora para el transporte de los envases.

### 3.2.2. Diseño del brazo robótico

El diseño del brazo robótico está constituido con 227 piezas del kit Lego Mindstorms NXT. El inventario del número de piezas se encuentra en el Anexo N° 5.

Se utiliza 3 servomotores; un servomotor (A) para el desplazamiento horizontal del brazo, un servomotor (B) para el desplazamiento vertical del brazo y otro motor (C) para la apertura y cierre de la garra.

El brazo coloca los recipientes con nivel inferior de producto en la zona de rechazo, y los recipientes con el nivel correcto los depositará en la zona de producto aceptado.

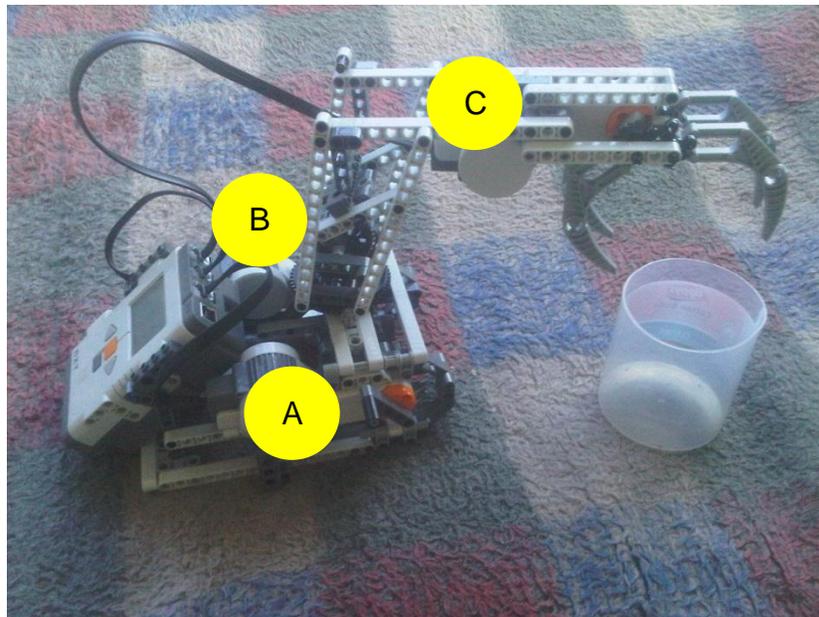


Figura 3.2. Diseño mecánico del brazo robótico  
Fuente: Investigador

El manual de ensamblaje del brazo robótico se encuentra en el Anexo N° 5.

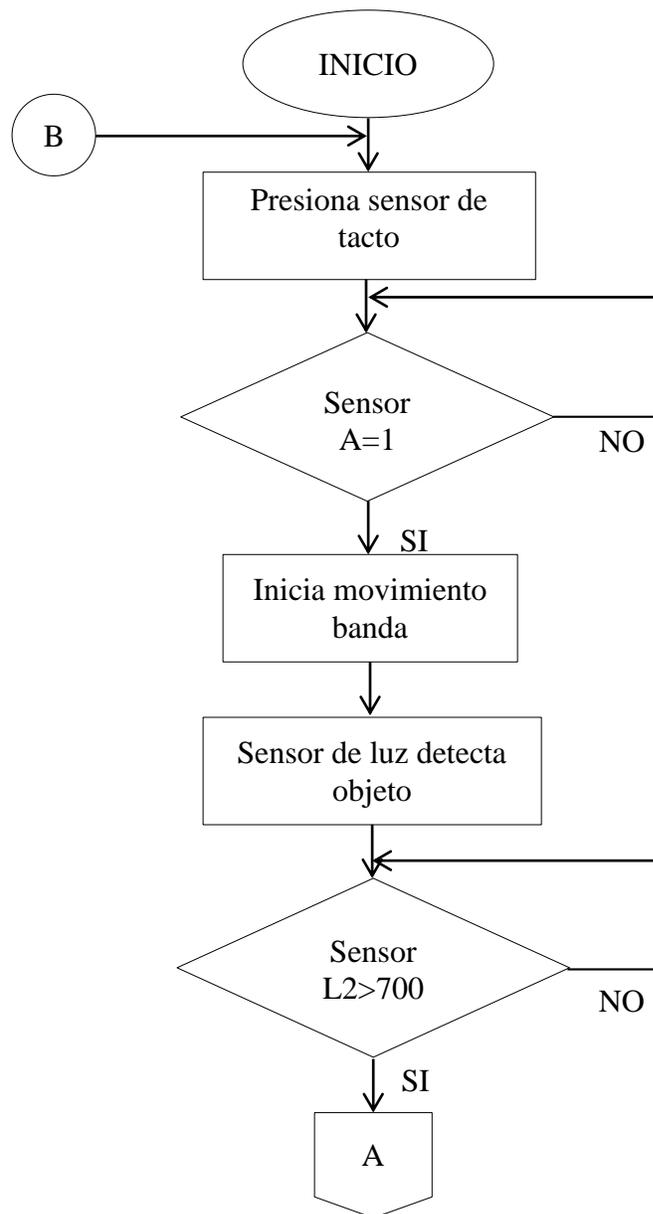
### 3.3. Programación del proyecto

#### 3.3.1. Programación de la banda transportadora con el software NXT-G

La banda transportadora está controlada con otro bloque NXT cuyos componentes están independientes del bloque que controla al brazo robótico.

### 3.3.1.1. Diagrama de bloques del funcionamiento

A continuación se observa el diagrama de bloques de la programación del bloque NXT para la banda transportadora.



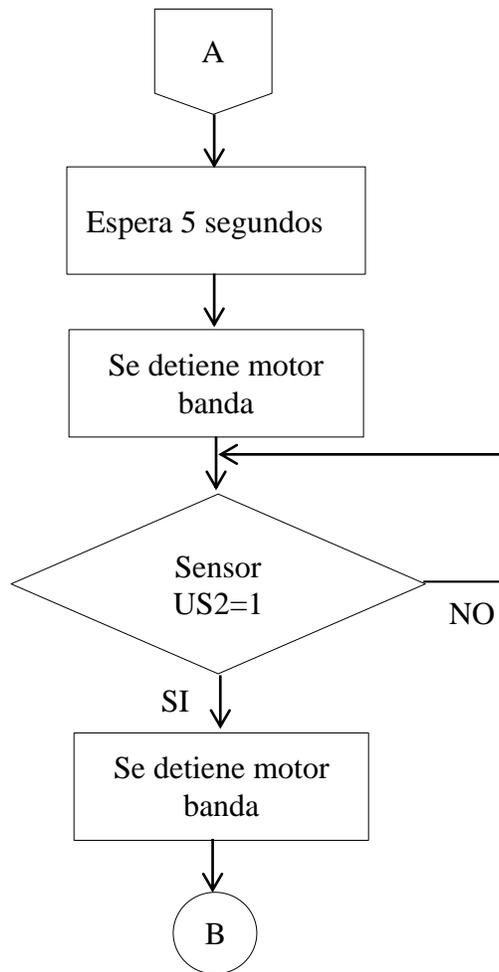


Figura 3.3. Diagrama de bloques de la programación de la banda transportadora  
Fuente: Investigador

### 3.3.1.2. Programación del bloque NXT-1 de la banda transportadora

El código del programa de la banda transportadora está basado en el entorno gráfico del software NXT-G, el cual se basa en bloques los cuales permite configurarlos dependiendo la acción que ejecute y el ambiente de trabajo en el cual se desenvuelva.

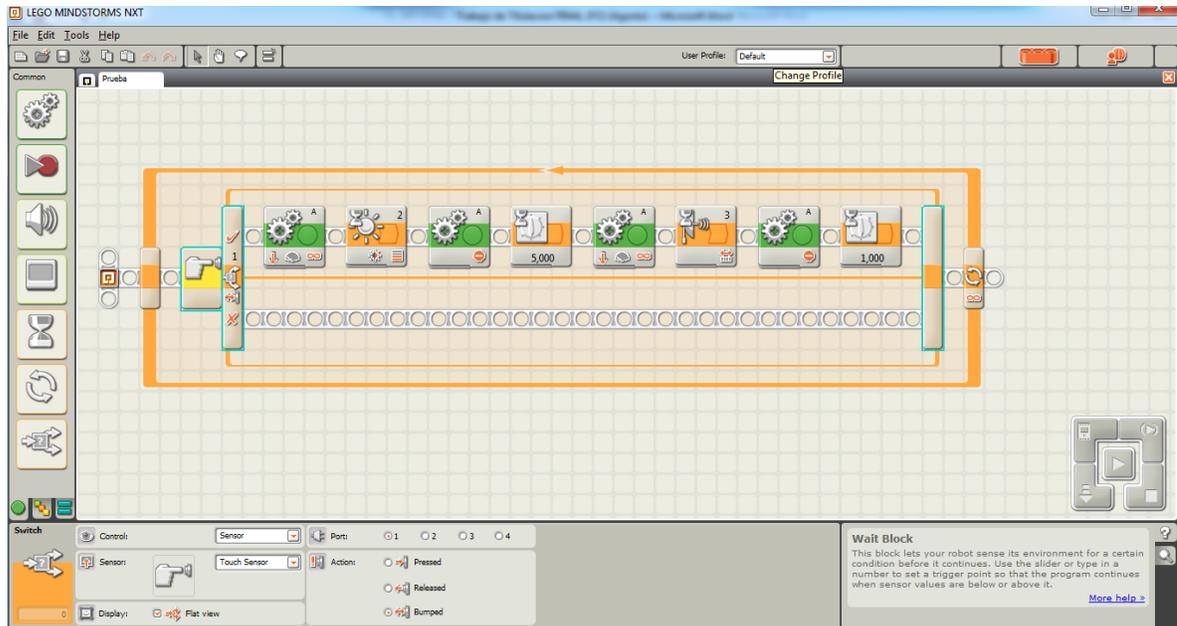


Figura 3.4. Programación de la banda transportadora  
Fuente: Investigador (NXT-G)

### 3.3.2. Programación en Matlab

A continuación se detalla el diagrama de flujo del funcionamiento del proyecto.

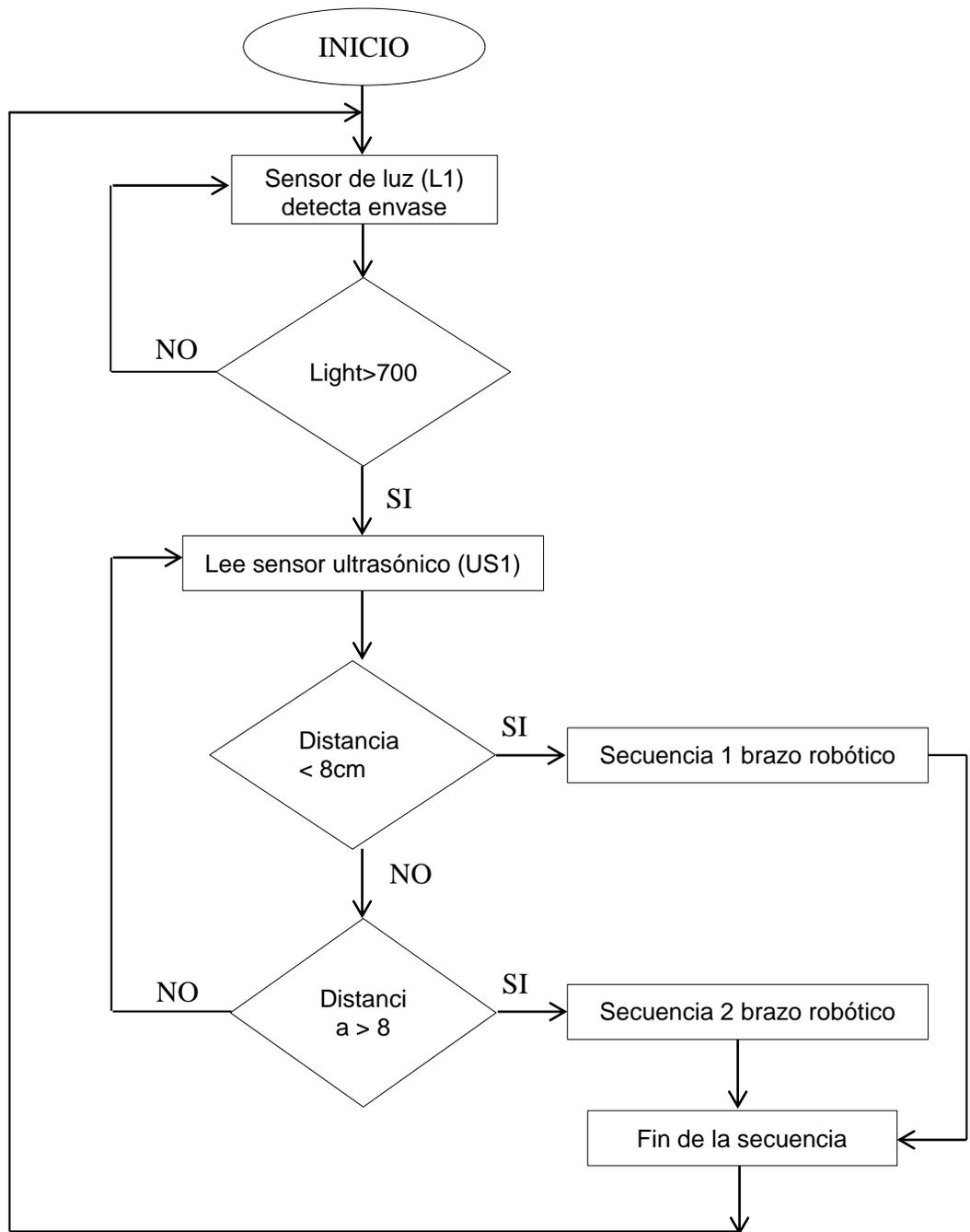


Figura 3.5. Diagrama de bloques de la programación del brazo robótico  
Fuente: Investigador

### 3.3.2.1. Código del archivo \*.m

Los rangos, configuración de entradas, configuración de sensores y comandos utilizados en el software de MATLAB necesarios para el funcionamiento del proyecto se detallan a continuación.

#### RANGO DE CALIBRACIONES DE LOS SENSORES

- Rango Sensor ultrasónico: 0 –255
- Rango Sensor de luz: 0 -1024
- Estado Sensor Tacto: 0 (off) –1(on)

#### ENTRADAS NXT:

- Sensor Tacto = SENSOR\_1
- Sensor Luz = SENSOR\_3
- Sensor Ultrasonido = SENSOR\_4

#### CONFIGURACIÓN DE SENSORES:

- **Sensor Luz:**

Light > 700 » Presencia de Recipiente

- **Sensor Ultrasonido:**

US < 8 » Nivel Recipiente Correcto

US > 8 » Nivel Recipiente Incorrecto

- **Sensor Tacto:**

SW = 1 » Se detiene Secuencia Brazo

#### COMANDOS UTILIZADOS

- h = COM\_OpenNXT()
- OpenLight()

- GetLight()
- OpenUltrasonic()
- GetUltrasonic()
- USMakeSnapshot()
- USGetSnapshotResults()
- OpenSwitch()
- GetSwitch()
- SetMotor()
- SetPower()
- SetAngleLimit ()
- NXT\_PlayTone(Hz,time)
- CloseSensor();
- COM\_CloseNXT();
- xlswrite („dta.xls“,name)

El programa con el cual realiza la rutina el brazo robótico se encuentra detallado en el Anexo N° 4.

La librería con los respectivos comandos para la programación en Matlab se detallan en el Anexo N° 5.

### **3.4. Funcionamiento del sistema**

El proyecto final consta de 5 sensores distribuidos de forma estratégica en diferentes posiciones, los del Lego NXT presentan señales digitales: el sensor de ultrasonido de 8 bits de resolución, el sensor de tacto de 2 bits y el sensor de Luz de 10 bits, en este capítulo se presentan los valores configurados y medidos para la toma de decisiones en el circuito controlador.

El brazo robótico está construido con las piezas de Lego Mindstorms, utilizando 2 servomotores con dos grados de libertad (90° y 180°) para su movimiento y traslado de envases (Puerto A y B), el tercer motor (Puerto C) se usa para el cierre y la apertura de la

garra, un sensor de luz L1 (Puerto 3) para identificar la presencia del envase y un sensor de ultrasonido US1 (Puerto 4) que mide el nivel de producto dentro del envase.

El funcionamiento del sistema se divide en 4 etapas marcadas, a continuación se describe cada una de ellas:

## 1. ETAPA DE INICIO DEL SISTEMA

En la figura 3.6 se observa el inicio del sistema en el cual se coloca un envase con cualquier nivel de llenado de producto (vacío, medio, lleno), para que el sistema se inicie se procede a presionar el sensor de tacto este envía una señal al bloque de programación NXT-1 el mismo controla el movimiento del motor de la banda para su inicio.

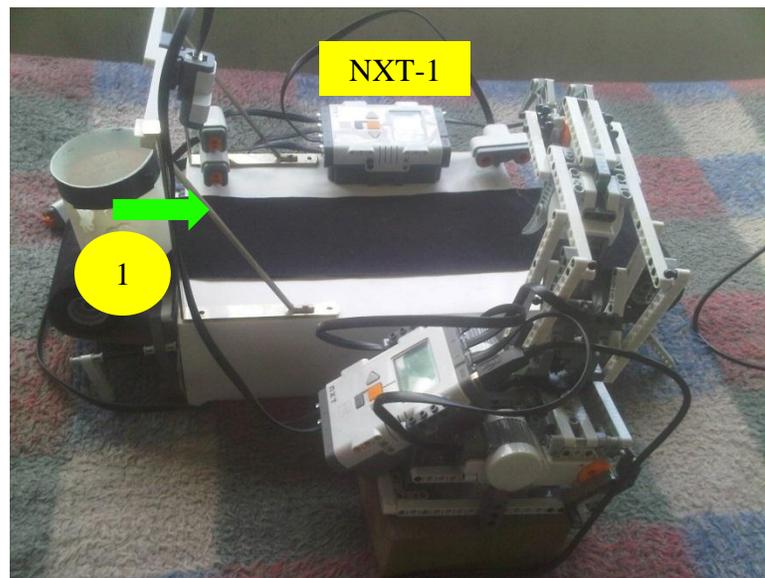


Figura 3.6. Etapa de inicio del sistema.  
Fuente: Investigador

## 2. ETAPA DE SENSADO DE NIVEL DEL PRODUCTO

En la figura 3.7 se observa que después del movimiento del envase los sensores de luz L1 que detecta el envase, detienen la banda transportadora y el L2 envía una señal para que el sensor ultrasónico US1 mida el nivel de llenado del producto, el cual es representado en una pantalla en la PC en el software de Matlab los resultados de la medición se observan en el ítem 3.6.1.

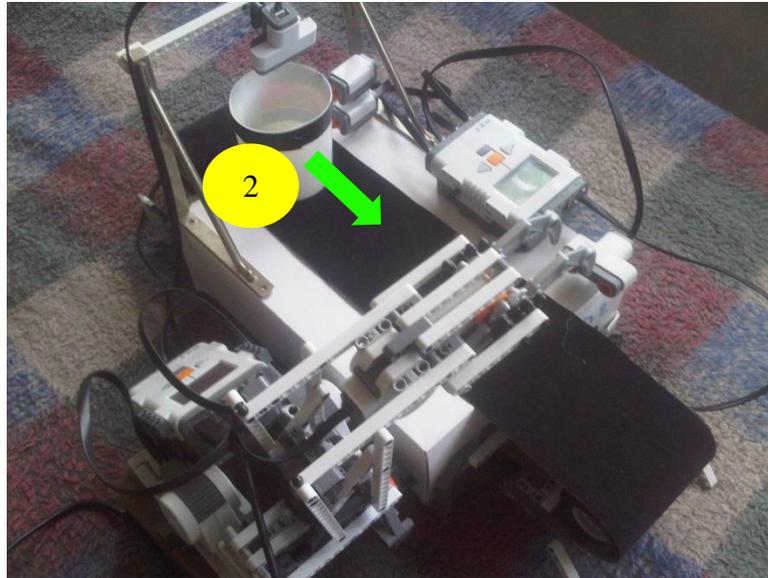


Figura 3.7. Etapa de inicio de sensado de nivel del producto.  
Fuente: Investigador

### 3. ETAPA DE INICIO DEL BRAZO ROBÓTICO

En la figura 3.8 se observa la etapa en la que el envase es detenido por efecto de la detección del sensor ultrasónico US2, para que el brazo robótico proceda a recoger el recipiente y llevarlo a su lugar dependiendo el nivel del producto.

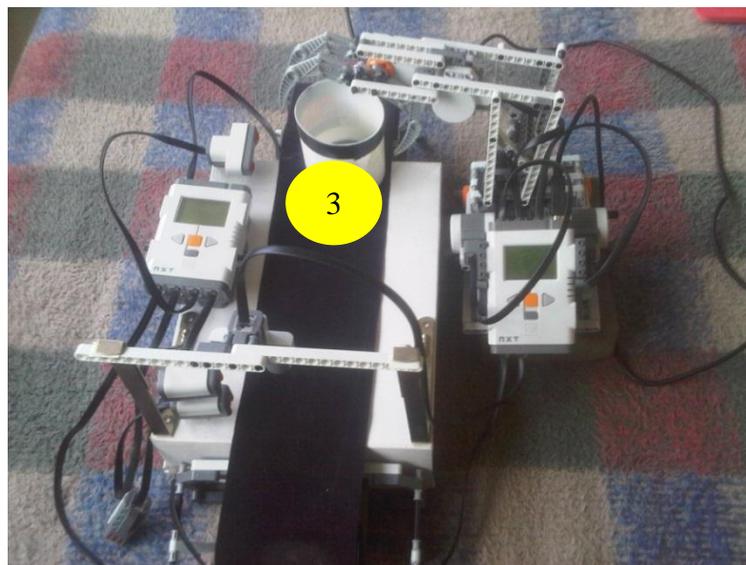


Figura 3.8. Etapa de inicio del brazo robótico.  
Fuente: Investigador

#### 4. ETAPA DE TRASLADO DE ENVASES

Esta es la última etapa del sistema en la cual una vez que el bloque NXT-2 ha recibido el dato del nivel del producto, el brazo robótico procede a clasificar los envases dependiendo su nivel. Si los recipientes están incompletos el brazo realiza un giro de 180° en su eje hacia la derecha hasta colocar el objeto al terminar el giro, si los recipientes están completos el brazo gira 90° hacia la derecha hasta llegar a la posición en la que debe depositar el objeto.

De esta manera se completa la secuencia del programa, el brazo regresa a su posición original hasta que nuevamente se repita el lazo del programa.

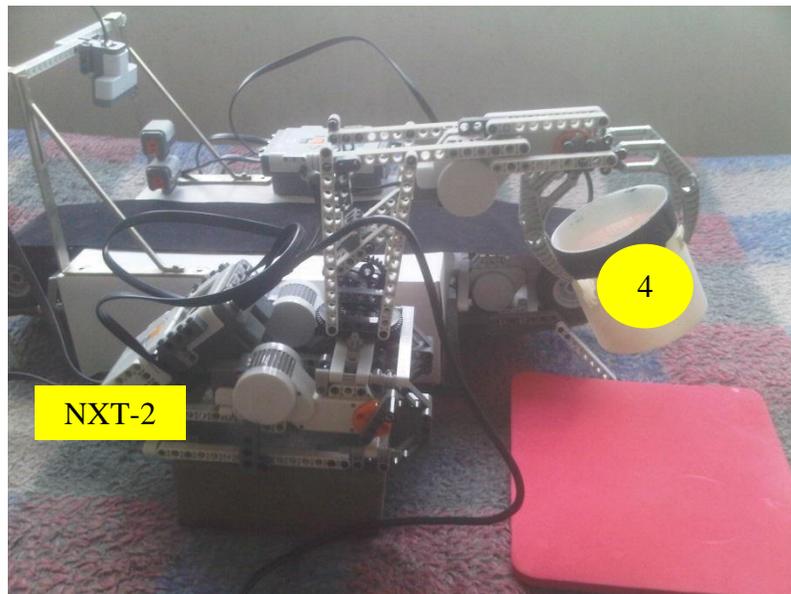


Figura 3.9. Etapa de traslado y clasificación de envases.  
Fuente: Investigador

#### 3.5. Presentación de resultados

Para la calibración de los sensores de luz y ultrasónico se usa la herramienta Watch Sensor GUI que está dentro del paquete de instalación de RWTH Aachen MINDSTORMS NXT Toolbox, instalado y explicado en el capítulo 2, mismo que permite fijar los parámetros o valores para la actuación del brazo robótico.

### 3.5.1. Calibración de sensores NXT

Dentro del toolbox de Matlab RWTH - Mindstorms NXT se encuentran herramientas GUI que permiten una visualización gráfica de los valores analógicos medidos por los sensores Lego. El GUI se encuentra en la librería de ejemplos NXT del Matlab.

El proyecto contiene 2 sensores, un ultrasónico y uno de luz. El sensor de luz se lo usa para detectar la presencia de un envase u objeto en la banda, en este punto el sensor ultrasónico se encarga de medir el nivel de llenado del producto existente en los envases.

El sensor de luz que está conectado al puerto 3 del bloque tiene una resolución de 10 bits, es decir muestra valores de 0 a 1124. En la figura 3.10 se observa valores inferiores a 350 bits, esto significa que no existe la presencia de ningún objeto claro cerca del sensor y los valores que superan el rango de 420 bits es porque existe la presencia de un objeto claro cerca del sensor. Este GUI puede realizar la lectura y calibración de todos los sensores y motores existentes en el kit.

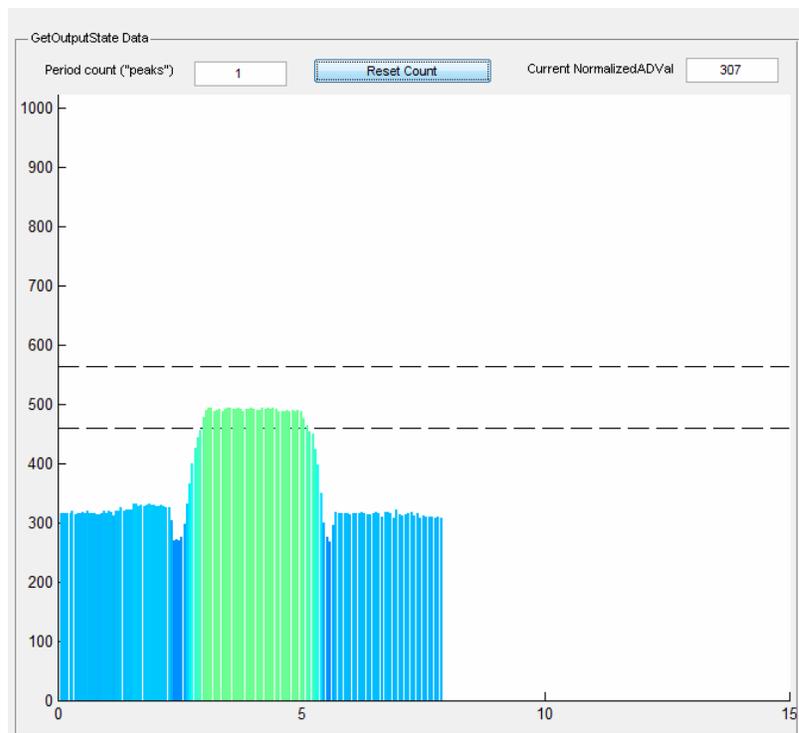


Figura 3.10. Rango de valores del sensor de luz.  
Fuente: Investigador (Matlab R2013b)

El sensor Ultrasónico consta de 8 bits de resolución, es decir que representa valores entre 0 a 256, para el proyecto se utiliza envases con tres niveles de producto, vacío, medio y lleno respectivamente. El sensor ultrasónico está colocado a una altura de 10cm por encima de la banda transportadora.

### 3.5.2. Valores medidos

Niveles	Valor medido (bits)
Vacío	16
Medio	11
Lleno	7

Tabla 3.2. Valores medidos del sensor ultrasónico  
Fuente: Investigador (Matlab R2013b)

La superficie de color blanco en las gráficas mostradas representan los valores medidos del sensor ultrasónico (US1) en este caso no detecta nada de producto en un recipiente.

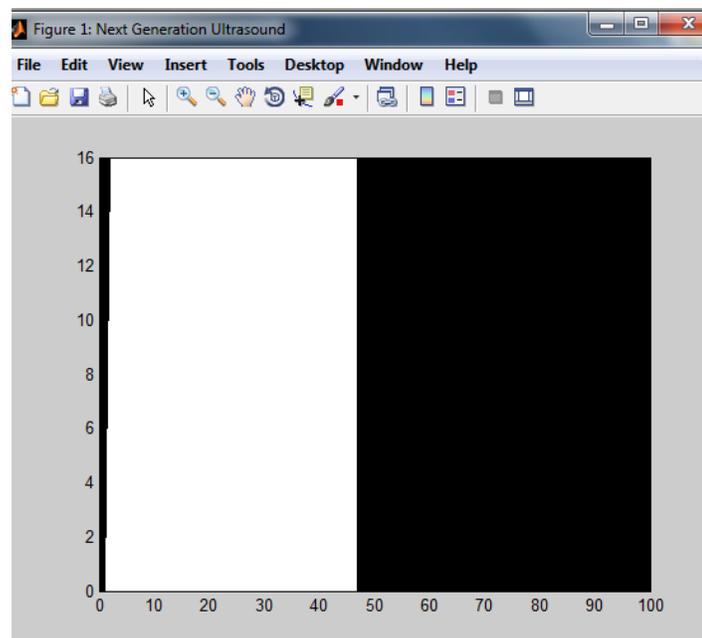


Figura 3.11. Rango de Valores, Sensor Ultrasónico en nivel de producto Vacío.  
Fuente: Investigador (Matlab R2013b)

El siguiente texto es el resultado de las configuraciones establecidas en la programación realizada en Matlab y mostradas en el Command Window, se resalta con amarillo el nivel INCOMPLETO obtenido por el sensor ultrasónico (US1).

>> PBtesis

NXTMotor object properties:

Port(s): 0	(A)
Power: -100	(reverse)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 0	(A)
Power: 100	(forward)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 1	(B)
Power: -100	(reverse)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 1	(B)
Power: 100	(forward)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 2	(C)
------------	-----

Power: -20 (reverse)  
SpeedRegulation: 1 (on)  
SmoothStart: 0 (off)  
TachoLimit: 0 (no limit)  
ActionAtTachoLimit: 'Brake' (brake, turn off when stopped)

NXTMotor object properties:

Port(s): 2 (C)  
Power: 20 (forward)  
SpeedRegulation: 1 (on)  
SmoothStart: 0 (off)  
TachoLimit: 0 (no limit)  
ActionAtTachoLimit: 'Brake' (brake, turn off when stopped)

Connectando.....

**Nivel de Recipiente INCOMPLETO**

En la figura 3.12 se muestra un nivel medio de producto que contiene un recipiente.

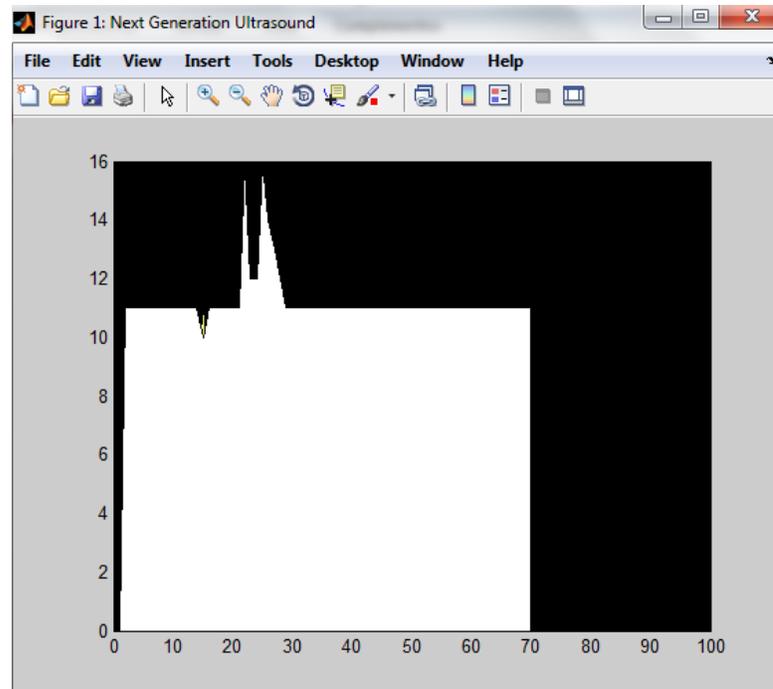


Figura 3.12. Rango de Valores, Sensor Ultrasónico en nivel de producto Medio.  
Fuente: Investigador (Matlab R2013b)

La superficie de color amarillo en las gráficas mostradas representan los valores medidos del sensor ultrasónico que indica el nivel de producto en el recipiente.

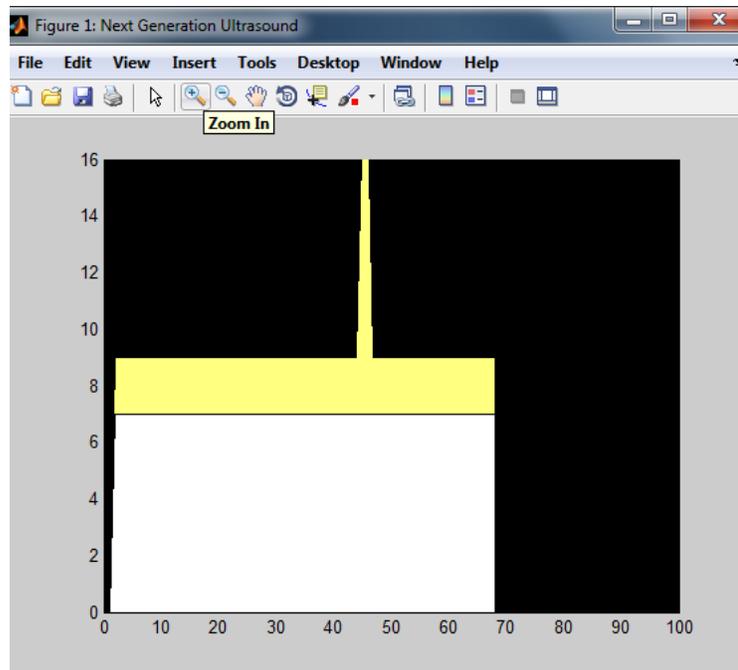


Figura 3.13. Rango de Valores, Sensor Ultrasónico en nivel de producto Lleno.  
Fuente: Investigador (Matlab R2013b)

El siguiente texto es el resultado de las configuraciones establecidas en la programación realizada en Matlab y mostradas en el Command Window, se resalta con amarillo el nivel OK obtenido por el sensor ultrasónico (US1).

>> PBtesis

NXTMotor object properties:

Port(s): 0	(A)
Power: -100	(reverse)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 0	(A)
Power: 100	(forward)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 1	(B)
Power: -100	(reverse)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 1	(B)
Power: 100	(forward)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 2	(C)
Power: -20	(reverse)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

NXTMotor object properties:

Port(s): 2	(C)
Power: 20	(forward)
SpeedRegulation: 1	(on)
SmoothStart: 0	(off)
TachoLimit: 0	(no limit)
ActionAtTachoLimit: 'Brake'	(brake, turn off when stopped)

Connectando.....

**Nivel de Recipiente OK**

### 3.6. Evaluación técnica

Antes de operar cualquier sistema es necesario realizar una evaluación cuyos resultados garanticen el buen funcionamiento del proyecto. En la siguiente tabla se puede observar los resultados obtenidos del funcionamiento del proyecto.

PROCEDIMIENTO	FUNCIONAMIENTO	
	CORRECTO	INCORRECTO
Encendido del bloque NXT-1	✓	
Encendido del bloque NXT-2	✓	
Activación del sensor de táctil controlador de la banda	✓	
Activación del motor controlador de la banda transportadora	✓	
Activación del sensor de luz L1	✓	
Activación del sensor de luz L2	✓	
Sensado de nivel del producto por el sensor ultrasónico US1	✓	
Presentación de resultados en la pantalla de la PC	✓	
Activación del motor controlador de la banda transportadora	✓	
Activación del sensor ultrasónico US2	✓	
Activación del motor movimiento vertical B	✓	
Activación del motor movimiento tenaza C	✓	
Activación del motor movimiento horizontal A	✓	
Regreso al estado inicial del sistema de todos los motores	✓	

Tabla 3.3. Resultados de la evaluación técnica  
Fuente: Investigador

### 3.7. Costos del proyecto

En la tabla 3.4 se muestra el costo total del proyecto con sus respectivos rubros.

Item	Rubro	Unidad	Cantidad	V. Unitario	V. Total
01	Kit Lego Mindstorms NXT	u	1	500.00	500.00
02	Baterías AA recargables	u	6	4.50	27.00
03	Cargador de baterías 4U	u	1	12.00	12.00
04	Caja de soporte	u	1	5.00	5.00
05	Bases soporte sensores	u	2	5.00	10.00
06	Papel gamuza	m	2	0.50	1.00
07	Envases	u	2	0.60	1.20
08	Impresiones	glb	1	50.00	50.00
09	Mano de obra	glb	1	500.00	500.00
10	Internet	glb	1	100.00	100.00
09	Transporte	glb	1	5.00	5.00
10	Material menudo	glb	1	10.00	10.00
				<b>TOTAL</b>	<b>1,221.20</b>

Tabla 3.4. Costos del proyecto.  
Fuente: Investigador

## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

- La amplia gama de piezas Lego con el que consta el kit permite realizar un sinnúmero de proyectos, al mismo tiempo se puede basar en proyectos ya realizados para tener una idea del resultado final a lograr. Tanto el brazo robótico como la banda transportadora están realizados con piezas lego con la ayuda de ciertos materiales que complementan su estructura mecánica.
- Los sensores permiten al sistema reaccionar de manera autónoma ante la presencia de fenómenos físicos, es así que los dos sensores de luz y un ultrasónico en este proyecto se utiliza para la detección de los envases y el otro sensor ultrasónico para la medición del nivel de producto dentro del recipiente, estos están colocados a una distancia prudente para que estos elementos ofrezcan una buena lectura para lo cual se los asigno.
- El bloque programable NXT permite realizar programaciones con varios lenguajes como por ejemplo C++, LabVIEW, Visual Basic, pero para el presente se utiliza Matlab como lenguaje alternativo para la movilización del brazo robótico, para adquisición y representación de los datos adquiridos por los sensores. Para complementar el proyecto se utiliza el lenguaje que viene por default en el kit el NXT-G para el control de la banda transportadora, es así que el proyecto se desenvuelve con dos entornos de programación.
- Lo interesante de la utilización de Lego Mindstorms NXT con Matlab es que mediante varios comandos que incluyen en la librería se puede realizar un mejor control de posición y de velocidad de los servomotores utilizados para el brazo robótico, además se puede calibrar los sensores para obtener mejor resultado en la adquisición de datos. El proyecto simula un entorno industrial para clasificación de envases que llevan cualquier tipo de producto a diferente nivel, además como valor agregado el proyecto ofrece la representación en una pantalla de la pc para que pueda ser monitoreado desde la estación de trabajo mediante conexión USB.

## RECOMENDACIONES

- Realizar montaje de diseños similares hasta lograr la estructura deseada para no tener problemas con la estructura mecánica y mantener una buena combinación hardware/software.
- Calibrar los sensores antes de poner en funcionamiento a los mismos, debido a que en diferentes ambientes los sensores pueden comportarse de diferente forma y entregar datos diferentes a los que se requiere.
- Investigar lenguajes alternativos los cuales sean compatibles con el bloque NXT, instalar las librerías necesarias para el funcionamiento del programa, también es recomendable instalar programas que permitan trabajar en el entorno de Matlab, de vital importancia para el funcionamiento es necesario los siguientes programas Cygwin, GNU ARM, NXT0sek, NeXTTool.exe, ECRobot (Embedded Coder Robot NXT) y si el bloque no se reconoce con el Matlab actualizar los drivers de los puertos USB.
- Utilizar los comandos con los cuales se puede manipular con mejor exactitud a los servomotores del kit, de la misma forma actualizar si es necesario el firmware del bloque dependiendo que versión se esté utilizando, de esta forma se evita que el bloque rechace los programas que se ejecutan.

## Bibliografía

- AACHEN.RWTH. (14 de Noviembre de 2009). *RWTH - Mindstorms NXT Toolbox for MATLAB*. Recuperado el 02 de Junio de 2014, de <http://www.mindstorms.rwth-aachen.de/>
- Cohen, J. (26 de Agosto de 2014). *High-End Computing Program*. Recuperado el 28 de Agosto de 2014, de <http://www.hec.nasa.gov/>
- EducaMadrid. (21 de Noviembre de 2011). *Complubot*. Recuperado el 30 de Julio de 2012, de [http://complubot.educa.madrid.org/pruebas/lego\\_nxt\\_version\\_educativa/lego\\_nxt\\_version\\_educativa\\_index.php](http://complubot.educa.madrid.org/pruebas/lego_nxt_version_educativa/lego_nxt_version_educativa_index.php)
- Marcos. (08 de Abril de 2010). *Roble.pntic*. Recuperado el 11 de Junio de 2014, de [http://roble.pntic.mec.es/amoc0048/nxt/dosmarcos/el\\_ladrillonxt.htm](http://roble.pntic.mec.es/amoc0048/nxt/dosmarcos/el_ladrillonxt.htm)
- Mindstorms, L. (6 de Julio de 2014). *Robotica*. Obtenido de Robotica: <http://robotica.com/es/tienda/LEGO-Education/LEGO-Mindstorms-Education-NXT/>
- Parker, D. (07 de Julio de 2009). *nxtprograms*. Recuperado el 16 de Mayo de 2014, de <http://163.19.6.25/~yuat/Lego/Lego-pics/www.nxtprograms.com/index.html>
- Perez, D. (04 de Enero de 2012). *Alcabot*. Recuperado el 8 de Julio de 2014, de <http://www.alcabot.com/alcabot/seminario2006/Trabajos/DiegoPerezDeDiego.pdf>
- School, A. B. (3 de Diciembre de 2009). *Robotica ABS*. Recuperado el 11 de Junio de 2014, de <http://gruporobotica-abs.zxq.net/robots.html>
- Asti Vera, A. (2013). Metodología de la investigación. Editorial Kapeluz, 5ta. Edición. Argentina.
- Hernández Sampieri, R; Fernandez Collado, C; Baptista Lucio, P. (2006). Metodología de la investigación. Editorial Mc Graw Hill. México.
- Paul, K. (1980). Introducción a la investigación científica, Ed. Trillas, México.
- Baena, G. (1980). Instrumentos de la Investigación, Editores Mexicanos Unidos, S.A. México D.F.

Marshall, G. (1985). Lenguaje de Programación para Microcontroladores. Editorial Paraninfo. Madrid-España.

Moncayo, F. (1990). Dinámica de la Manipulación de Robots y Control de Trayectoria. Quito, EPN.

Moore, H. (2010). Matlab para ingenieros. Editorial Prentice Hall. EEUU.

Rodriguez, M. (2003). Introducción rápida a Matlab y Simulink, Ediciones Díaz Santos, Madrid.

## **ANEXOS**

## **ANEXO 1**

**Entrevista estudiantes de la Universidad Israel carrera de Electrónica y Telecomunicaciones.**

**CONTROL DE ROBOTS LEGO CON MATLAB**

Fuente: Investigador

**Nombre:**

**Nivel:**

**1. ¿Tiene conocimiento de la existencia del kit de entrenamiento Lego Mindstorms?**

**si**

**no**

**2. ¿A tenido la oportunidad de realizar prácticas con el Lego NXT?**

**si**

**no**

**3. ¿A experimentado alguna vez la programación con Matlab?**

**si**

**no**

**4. ¿A podido observar algún proceso de automatización que involucre el control de nivel de un producto en envases?**

**si**

**no**

**5. ¿Con que tipo de lenguaje de programación ha tenido más experiencia en los laboratorios de la Universidad Israel? Cite el más utilizado.**

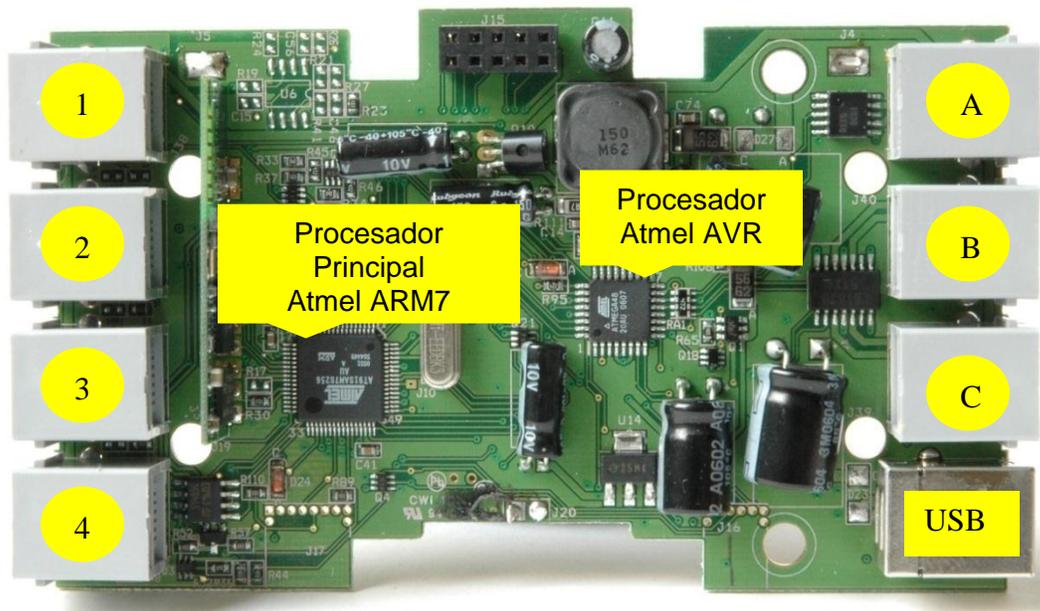
.....

## **ANEXO 2**

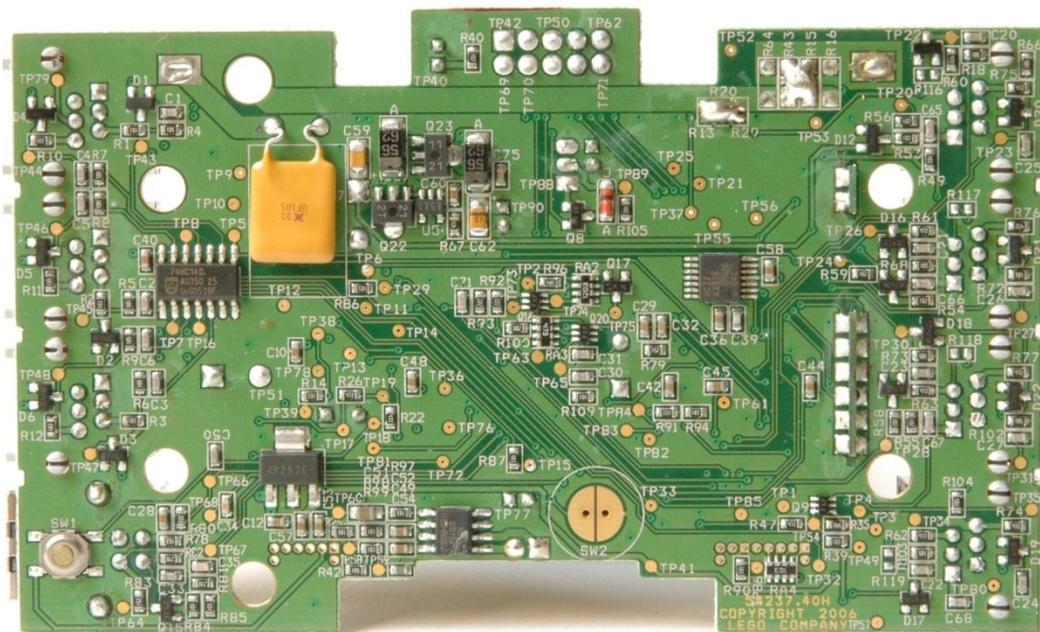
**Placa interna de los elementos y dispositivos de E/S entre otros**

Fuente: (Mindstorms, 2014)

# Placa, procesadores y periféricos de E/S



FUENTE: (Mindstorms, 2014)



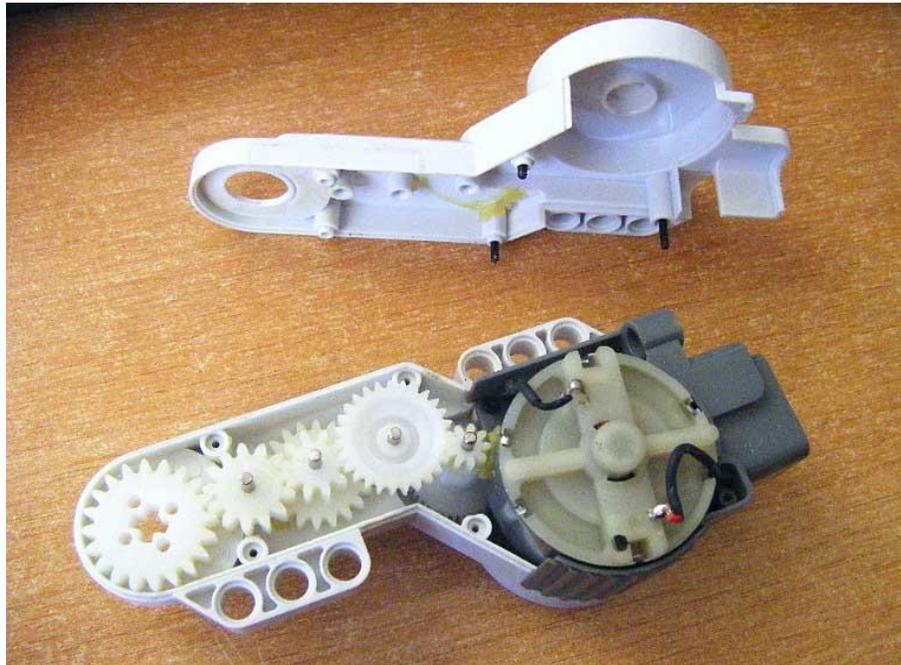
FUENTE: (Mindstorms, 2014)

## **ANEXO 3**

**Fotos internas de los Servomotores del Mindstorms NXT**

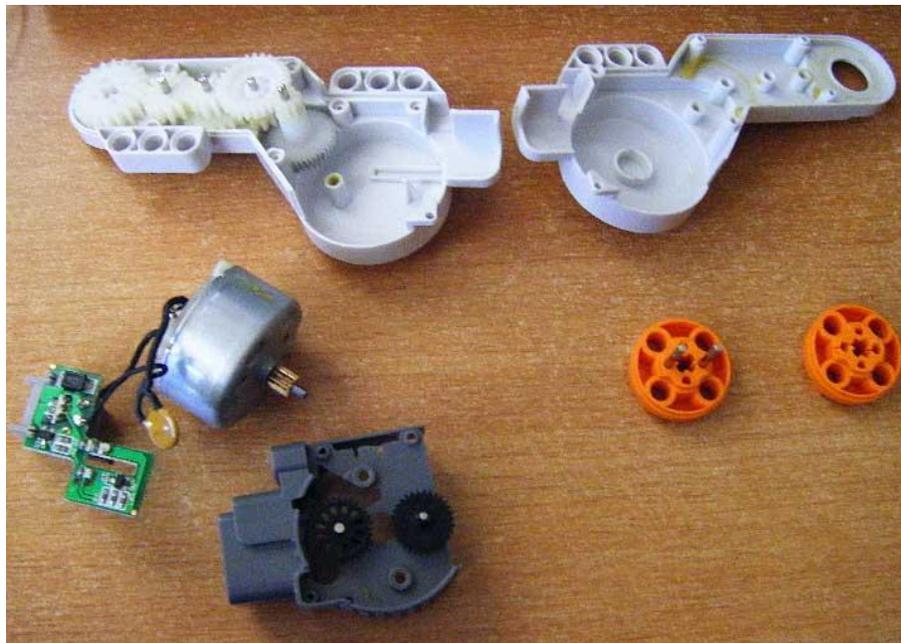
Fuente: (Mindstorms, 2014)

## Sistema de engranajes



Fuente: (Mindstorms, 2014)

## Servomotor y placas de control



Fuente: (Mindstorms, 2014)

## **ANEXO 4**

**Código del programa para la secuencia del brazo robótico y sensores**

Fuente: Investigador

```

%UNIVERSIDAD TECNOLÓGICA ISRAEL
%PROYECTO INTEGRADOR DE CARRERA
%ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES
%PROGRAMA PARA SECUENCIA DE BRAZO ROBÓTICO Y SENSORES
%FREDDY JULIÁN ORTIZ LÓPEZ
%QUITO - AGOSTO 2014

%Inicio de Programa
COM_CloseNXT all
clear all
close all
format compact

%Definición de puertos y variables
puertoSW = SENSOR_1;
puertoLH = SENSOR_3;
puertoUS = SENSOR_4;
mdA = NXTMotor('A', 'Power', -100)
muA = NXTMotor('A', 'Power', 100)
mdB = NXTMotor('B', 'Power', -100)
muB = NXTMotor('B', 'Power', 100)
mdC = NXTMotor('C', 'Power', -20)
muC = NXTMotor('C', 'Power', 20)

%Comunicación entre Matlab y NXT
h = COM_OpenNXT();
COM_SetDefaultNXT(h);

%Formato para gráfica del sensor ultrasónico
figure('name', 'Next Generation Ultrasound')
set(gca, 'Color', 'black');
hold on

%Mensaje en pantalla de Matlab de inicio de secuencia del brazo
robótico
disp ('Connectando.....')

%Inicio de lazo cerrado que entra con condición verdadera
while true

%Inicialización de variables para graficar el sensor de
ultrasonido
    n = 8;
    count = 100;
    plotcols = 8;
    outOfRange = 16;
    colors = flipud(hot(8));
    data = zeros(1, n);
    allX = (1:count+1)';

%Lectura de Sensor de Ultrasonido

```

```

    OpenUltrasonic(puertoUS);

for i = 1 : count
    USMakeSnapshot(puertoUS);
    pause(0.05);
    echos = USGetSnapshotResults(puertoUS);
    echos(echos == 255) = outOfRange;
    echos = [echos(1); diff(echos)];
    data = vertcat(data, echos');
    x = allX(1:i+1);
    clf
    hold on
    set(gca, 'Color', 'black');
    axis([0 count 0 outOfRange])

for j = plotcols : -1 : 1
    area(x, data(:, j) , 'FaceColor', colors(j, :))
end

%Lectura de sensor de luz
    OpenLight(puertoLH, 'ACTIVE');
    light = GetLight (puertoLH);
if light > 700
    OpenUltrasonic(puertoUS);
    pause (1);

%Captura de lectura de sensor de ultrasonido
    distancia = GetUltrasonic(puertoUS);
    pause (1);

%Condición de lectura de ultrasonido menor que 8
if distancia < 8

    %Mensaje en pantalla que recipiente tiene el nivel ok
    disp ('Nivel de Recipiente OK')
    pause (2);

%Servo B se mueve 6000 grados para bajar brazo a nivel de la banda
    mdB.SpeedRegulation      = false;
    mdB.TachoLimit           = 6000;
    mdB.ActionAtTachoLimit   = 'Brake';
    mdB.SmoothStart          = true;
    mdB.SendToNXT();
    mdB.WaitFor();
    NXT_PlayTone(400, 500);

% Servo C se mueve 40 grados para agarrar recipiente
    muC.SpeedRegulation      = false;
    muC.TachoLimit           = 40;
    muC.ActionAtTachoLimit   = 'Brake';
    muC.SmoothStart          = true;

```

```

    muC.SendToNXT();
    muC.WaitFor();
    NXT_PlayTone(400,500);

%Servo B se mueve 6000 para regresar a estado inicial
muB.SpeedRegulation    = false;
muB.TachoLimit        = 6000;
muB.ActionAtTachoLimit = 'Brake';
muB.SmoothStart       = true;
    muB.SendToNXT();
    muB.WaitFor();
    NXT_PlayTone(400,500);

%Servo A se mueve 4900 grados para mover base de servo y dejar
recipiente con nivel correcto
mdA.SpeedRegulation    = false;
mdA.TachoLimit        = 4900;
mdA.ActionAtTachoLimit = 'Brake';
mdA.SmoothStart       = true;
    mdA.SendToNXT();
    mdA.WaitFor();
    NXT_PlayTone(400,500);

%Servo C se mueve 40 grados para dejar recipiente
mdC.SpeedRegulation    = false;
mdC.TachoLimit        = 40;
mdC.ActionAtTachoLimit = 'Brake';
mdC.SmoothStart       = true;
    mdC.SendToNXT();
    mdC.WaitFor();
    NXT_PlayTone(400,500);

%Servo A se mueve 4900 grados para regresar a estado inicial
muA.SpeedRegulation    = false;
muA.TachoLimit        = 4900;
muA.ActionAtTachoLimit = 'Brake';
muA.SmoothStart       = true;
    muA.SendToNXT();
    muA.WaitFor();
    NXT_PlayTone(400,500);

%Sonido para indicar que el brazo está listo para coger otro
recipiente
    NXT_PlayTone(262,350);
    pause(0.2);
    NXT_PlayTone(440,50);
    pause(0.2);
    NXT_PlayTone(330,50);
    pause(0.2);

else

```

```

%Mensaje en pantalla que recipiente tiene el nivel incorrecto
    disp ('Nivel de Recipiente INCOMPLETO')

%Servo B se mueve 6000 grados para bajar brazo a nivel de la banda
    mdB.SpeedRegulation    = false;
    mdB.TachoLimit        = 6000;
    mdB.ActionAtTachoLimit = 'Brake';
    mdB.SmoothStart       = true;
    mdB.SendToNXT();
    mdB.WaitFor();
    NXT_PlayTone(400,500);

% Servo C se mueve 40 grados para agarrar recipiente
    muC.SpeedRegulation    = false;
    muC.TachoLimit        = 40;
    muC.ActionAtTachoLimit = 'Brake';
    muC.SmoothStart       = true;
    muC.SendToNXT();
    muC.WaitFor();
    NXT_PlayTone(400,500);

%Servo B se mueve 6000 para regresar a estado inicial
    muB.SpeedRegulation    = false;
    muB.TachoLimit        = 6000;
    muB.ActionAtTachoLimit = 'Brake';
    muB.SmoothStart       = true;
    muB.SendToNXT();
    muB.WaitFor();
    NXT_PlayTone(400,500);

%Servo A se mueve 9600 grados para mover base de servo y dejar
recipiente incorrecto
    mdA.SpeedRegulation    = false;
    mdA.TachoLimit        = 9600;
    mdA.ActionAtTachoLimit = 'Brake';
    mdA.SmoothStart       = true;
    mdA.SendToNXT();
    mdA.WaitFor();
    NXT_PlayTone(400,500);

%Servo C se mueve 40 grados para dejar recipiente
    mdC.SpeedRegulation    = false;
    mdC.TachoLimit        = 40;
    mdC.ActionAtTachoLimit = 'Brake';
    mdC.SmoothStart       = true;
    mdC.SendToNXT();
    mdC.WaitFor();
    NXT_PlayTone(400,500);

%Servo A se mueve 96000 grados para regresar a estado inicial

```

```

    muA.SpeedRegulation      = false;
    muA.TachoLimit          = 9600;
    muA.ActionAtTachoLimit  = 'Brake';
    muA.SmoothStart         = true;
    muA.SendToNXT();
    muA.WaitFor();
    NXT_PlayTone(400,500);

%Sonido para indicar que el brazo está listo para coger otro
recipiente
    NXT_PlayTone(262,350);
    pause (0.2);
    NXT_PlayTone(440,50);
    pause (0.2);
    NXT_PlayTone(330,50);
    pause (0.2);
end
else
end
end

%Lectura de sensor de tacto para terminar secuencia del brazo
    OpenSwitch(puertoSW);
    GetSwitch(puertoSW);
    switchState = GetSwitch(puertoSW);
if switchState == 1
    NXT_PlayTone(800,450);
    disp ('Finalizado.....')
break
end
end

%Exportación de datos Sensor_US a excel
xlswrite (DATOS_US,data,Sheet1,A2);

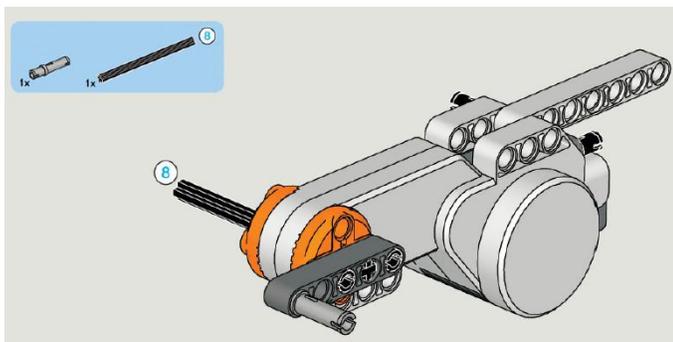
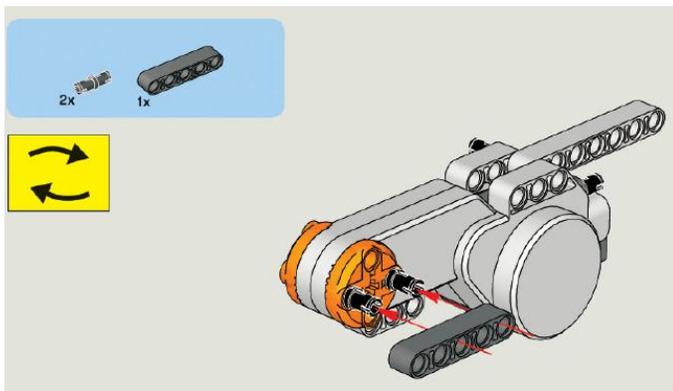
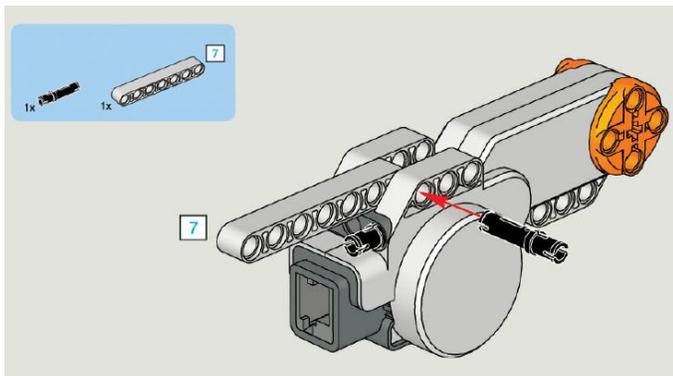
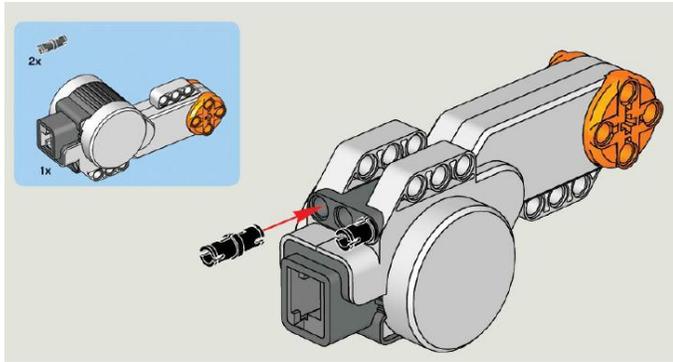
%Fin de comunicación de puertos
CloseSensor(puertoSW);
CloseSensor(puertoLH);
CloseSensor(puertoUS);
COM_CloseNXT(h);

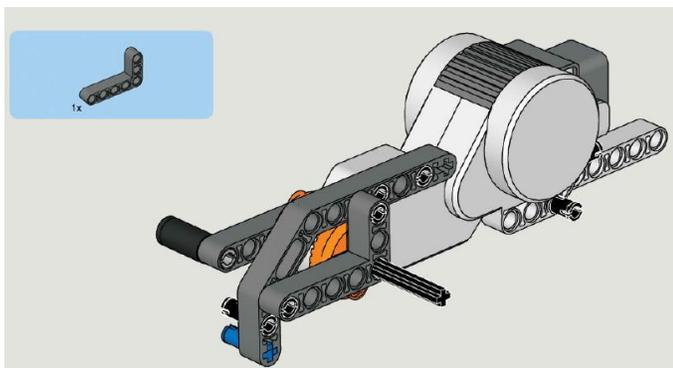
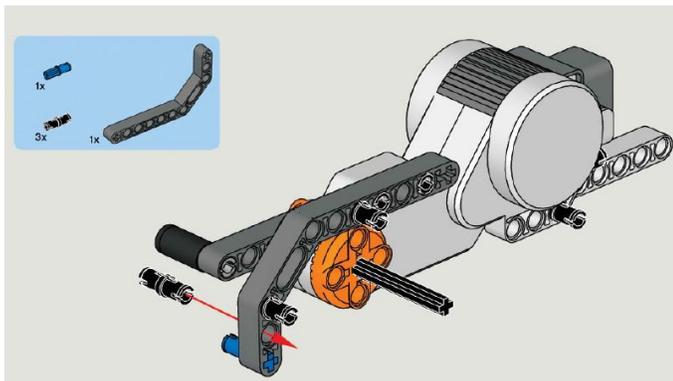
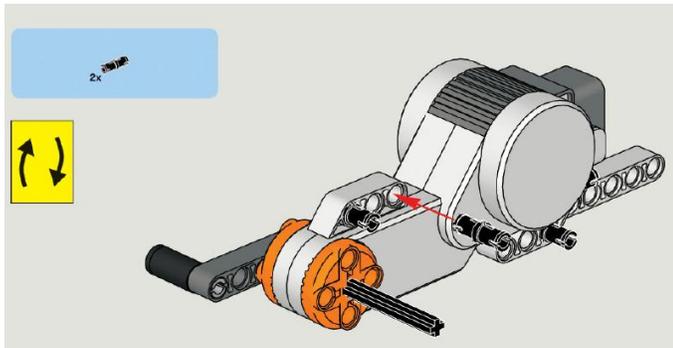
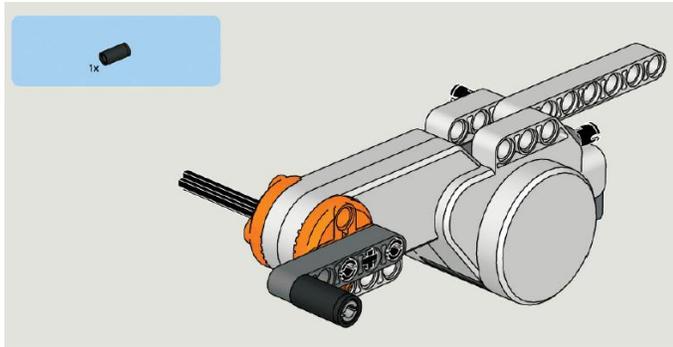
```

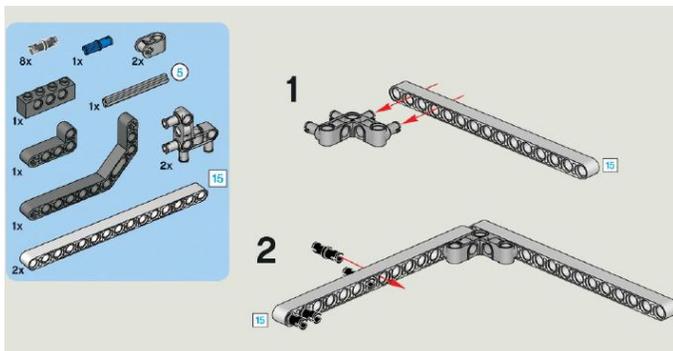
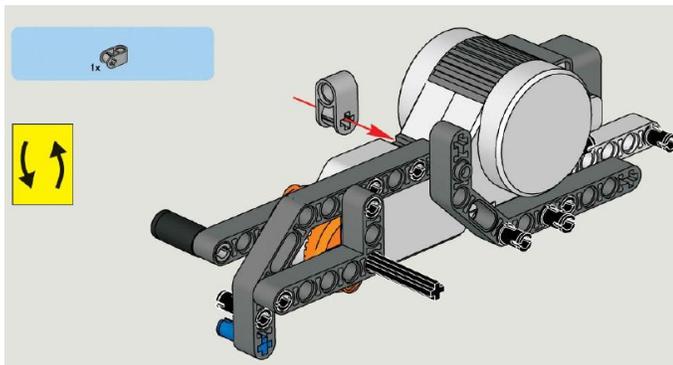
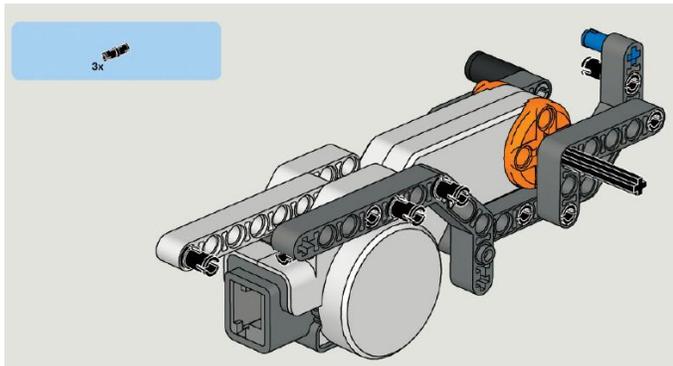
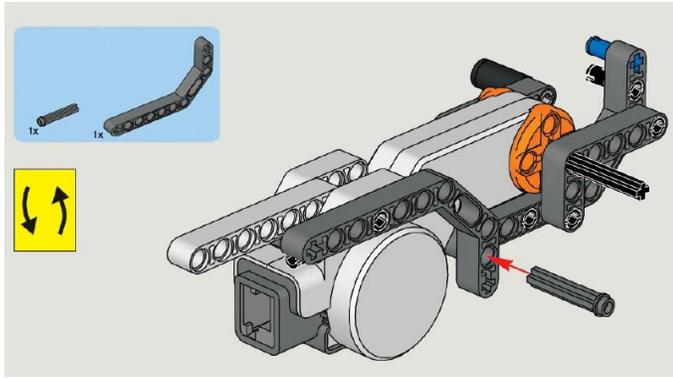
## **ANEXO 5**

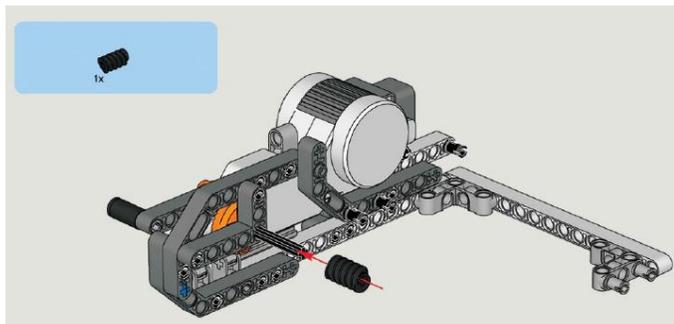
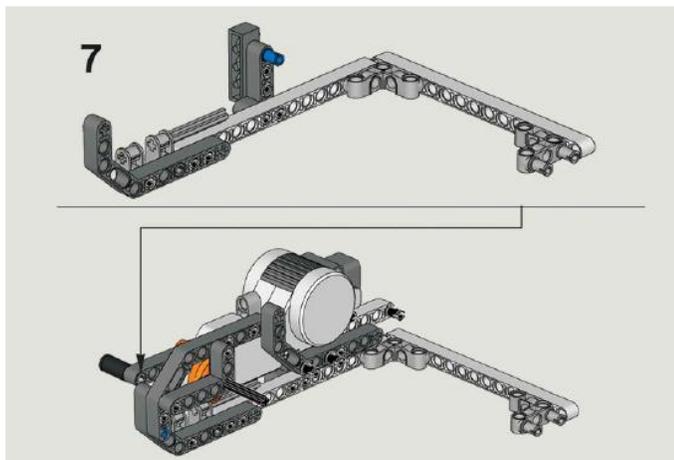
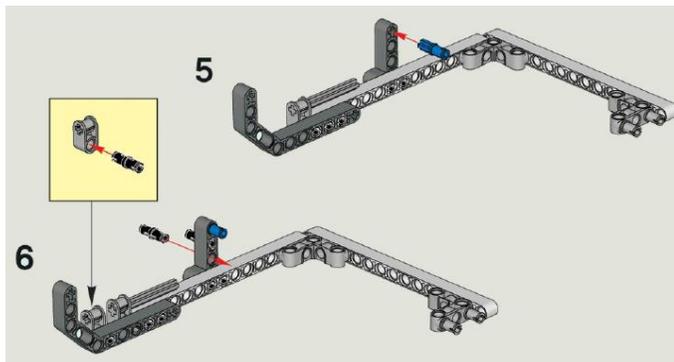
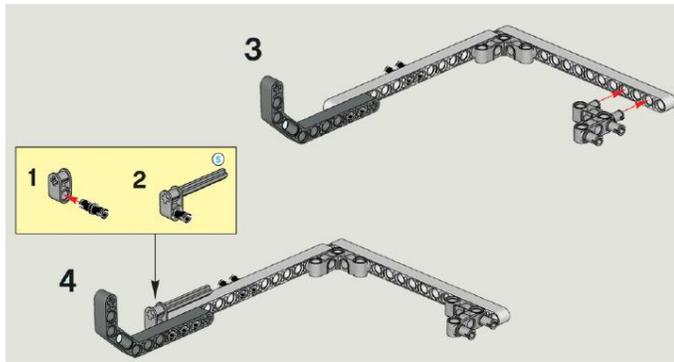
### **Manual de montaje del Brazo Robótico.**

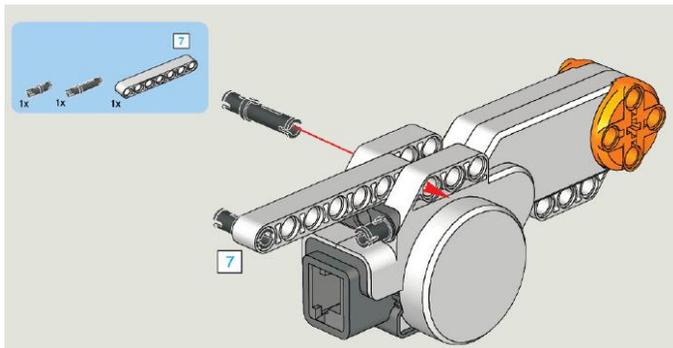
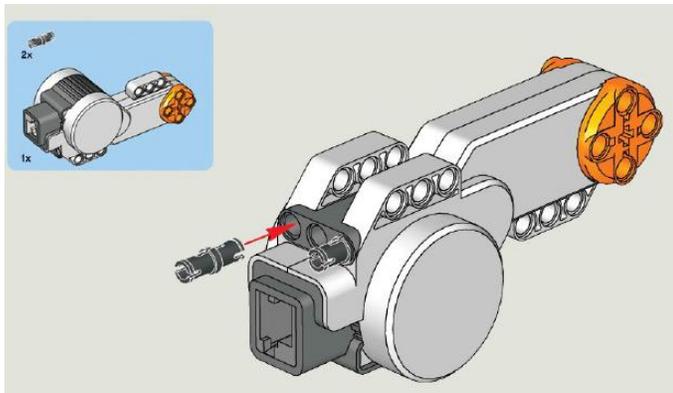
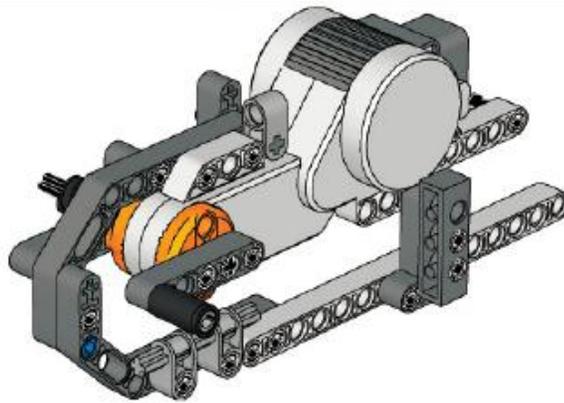
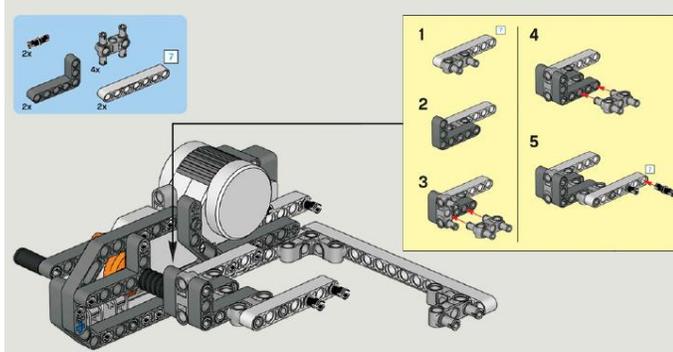
Fuente: Investigador (LEGO Digital Designer 4.2)

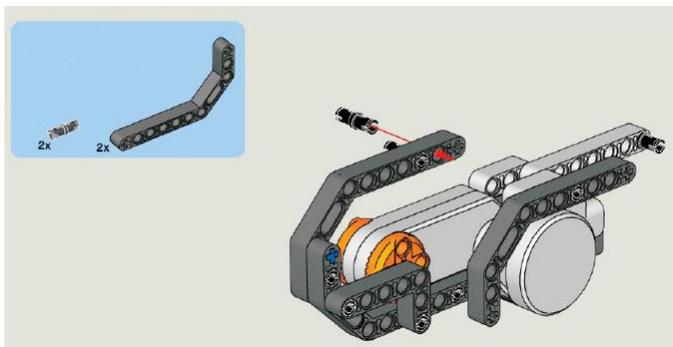
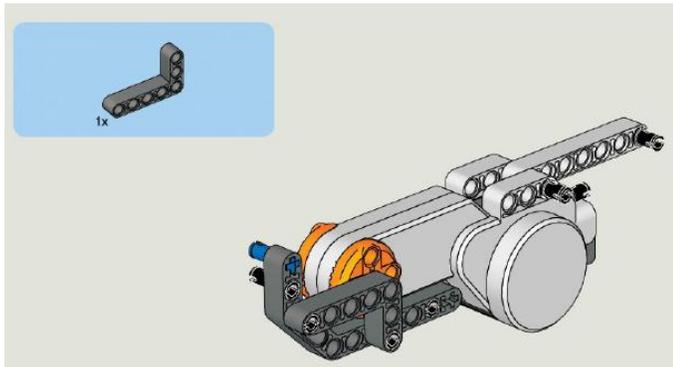
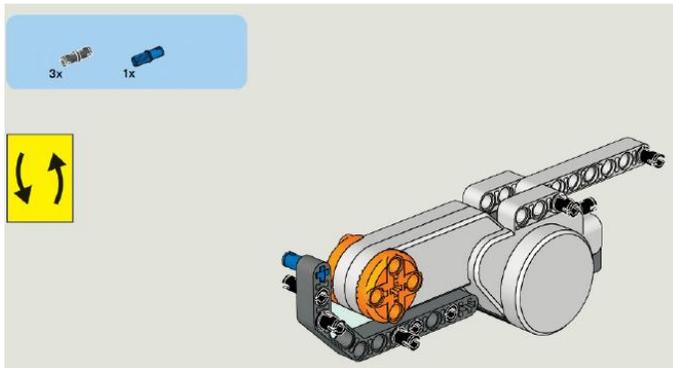
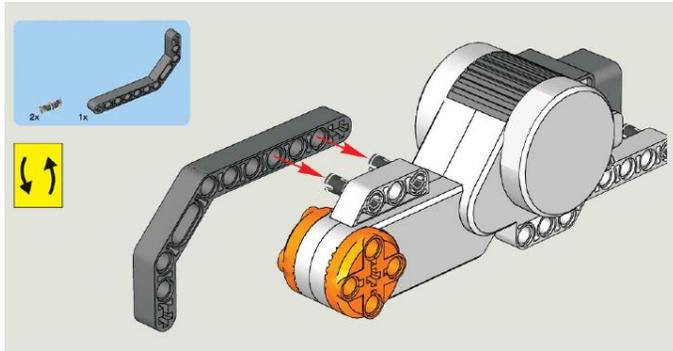


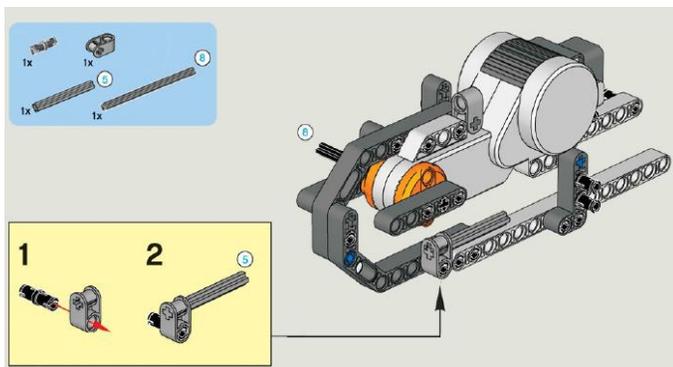
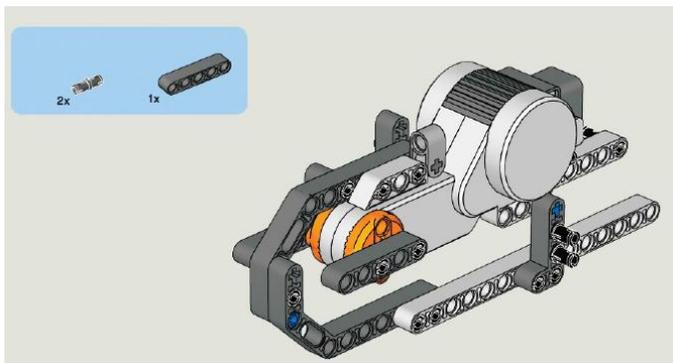
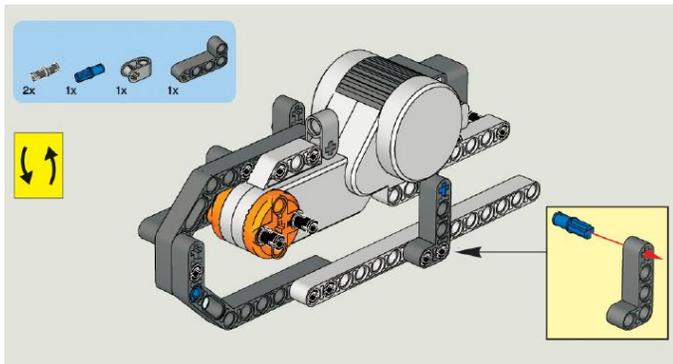
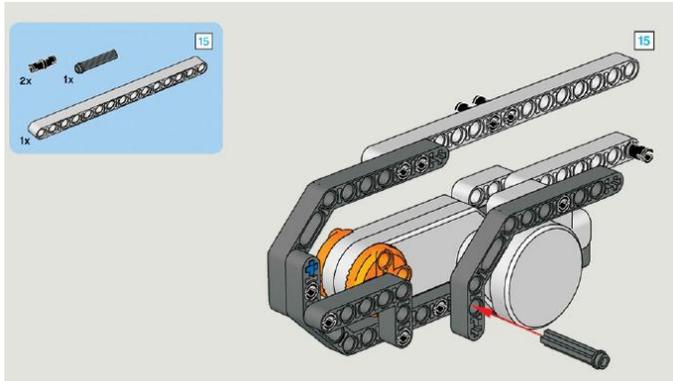


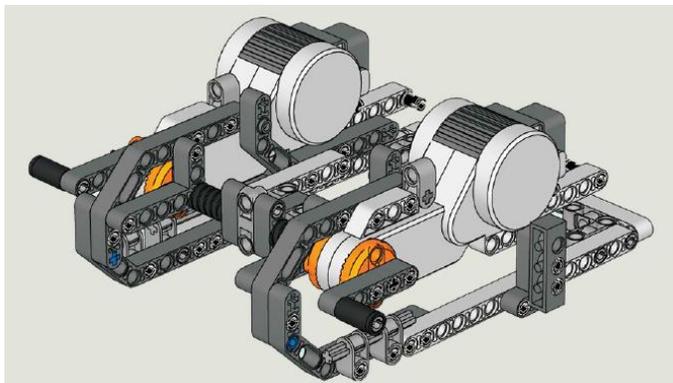
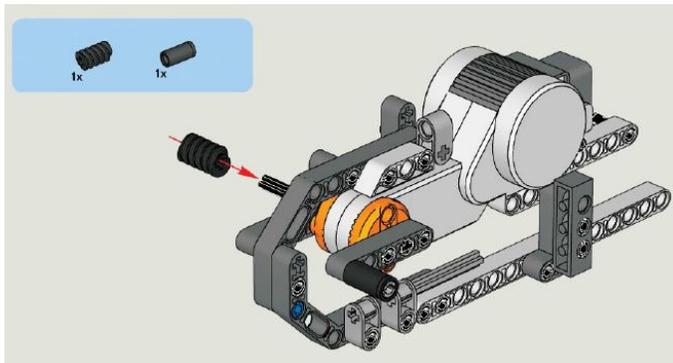
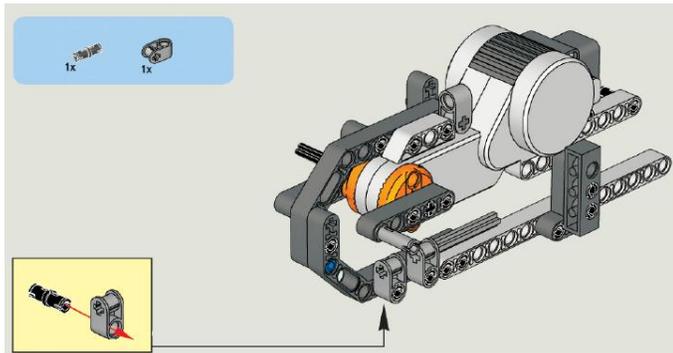
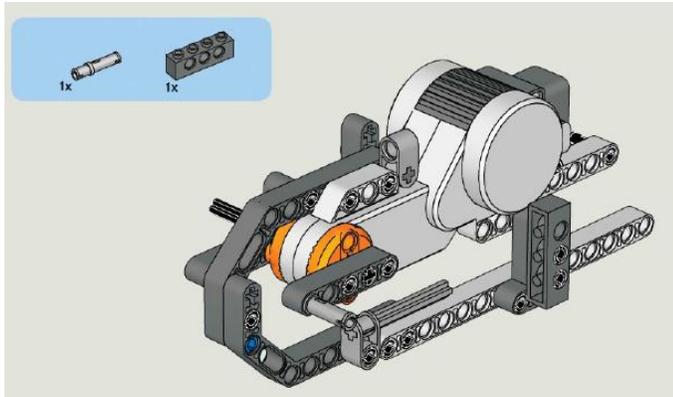


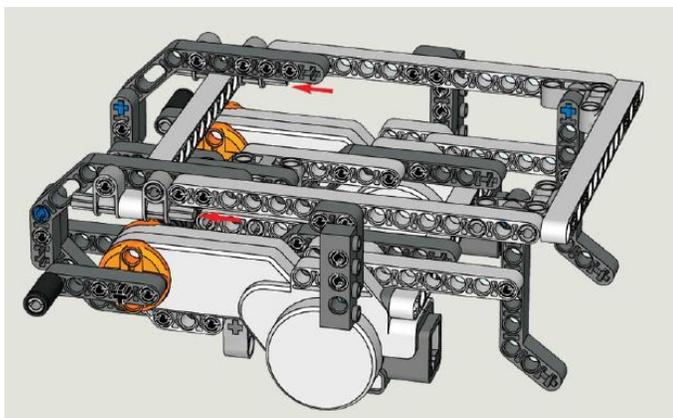
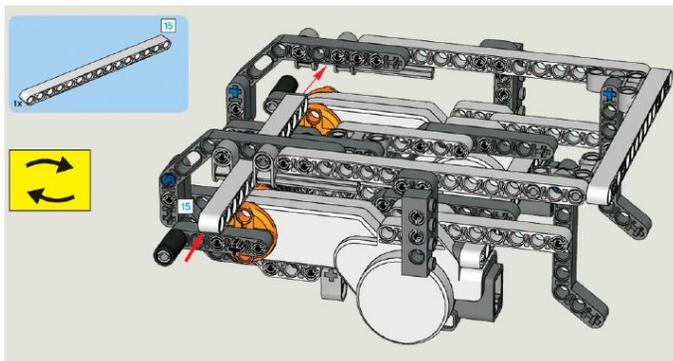
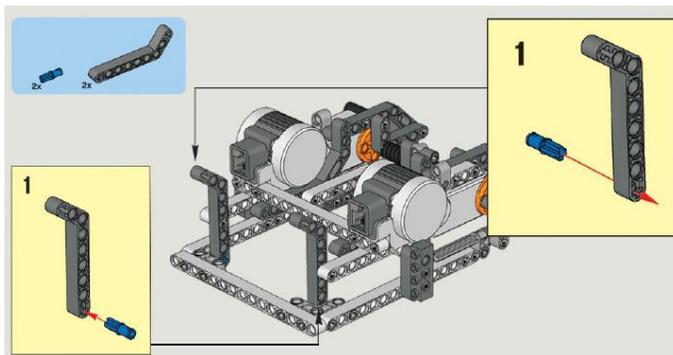
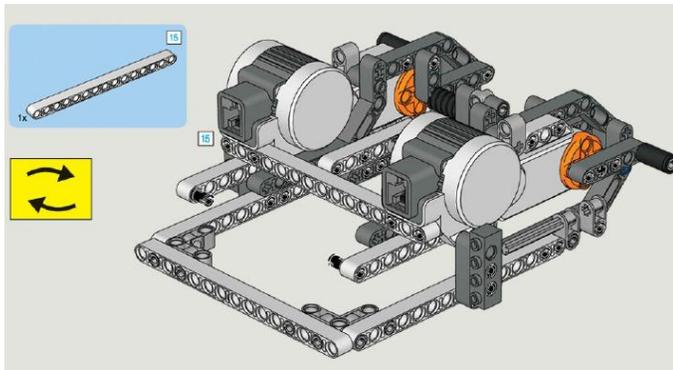


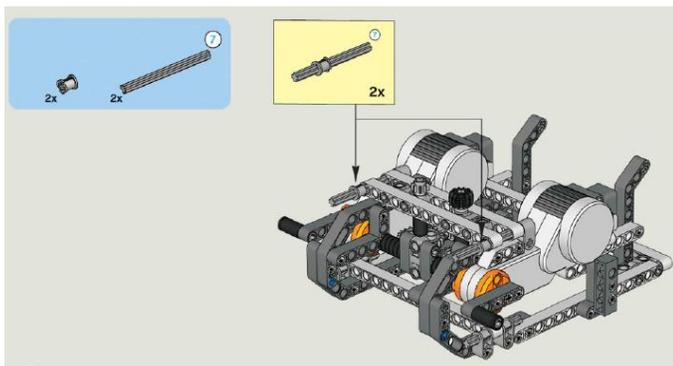
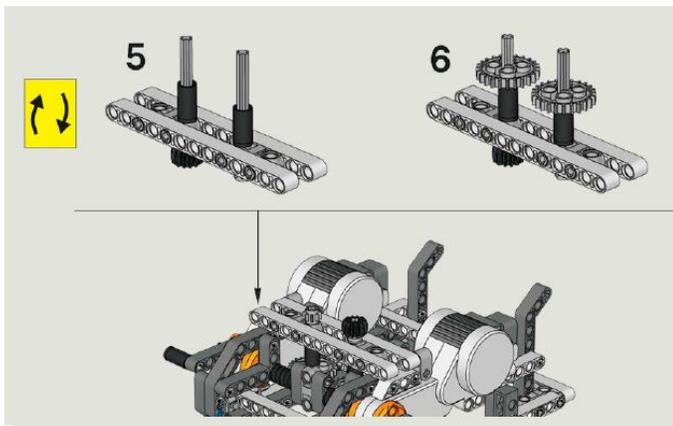
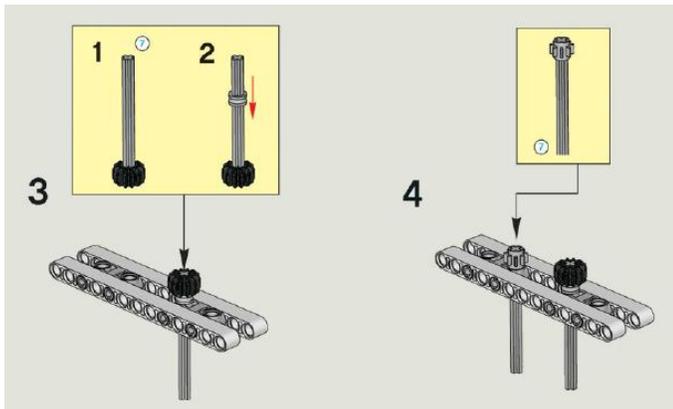
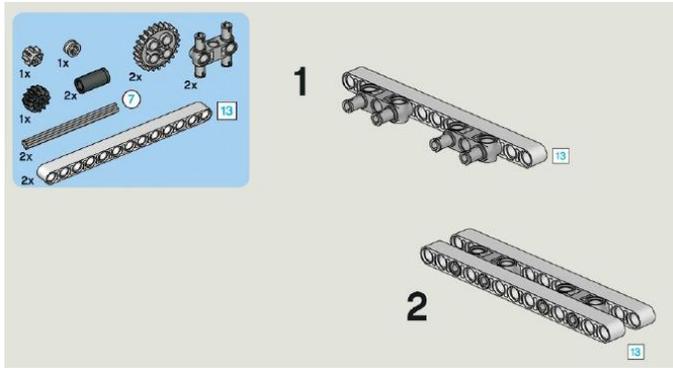


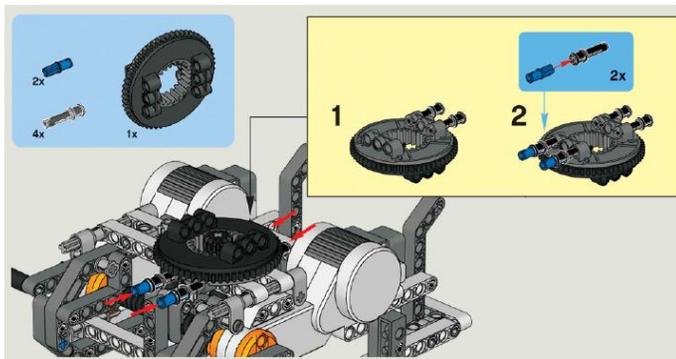
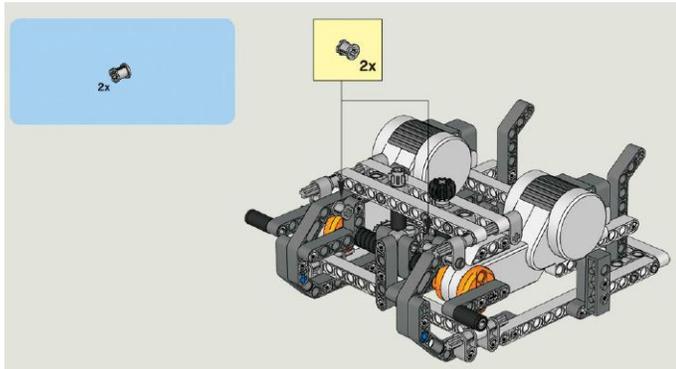
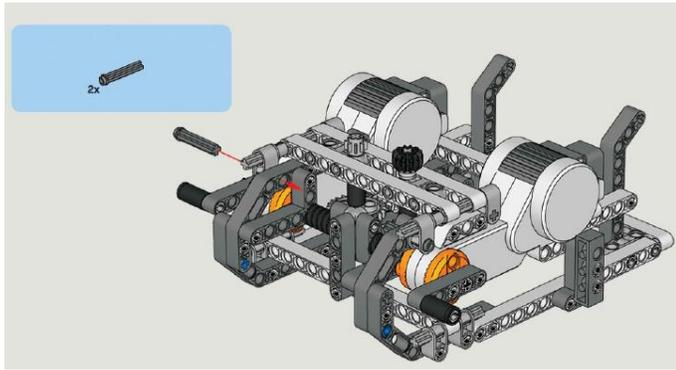
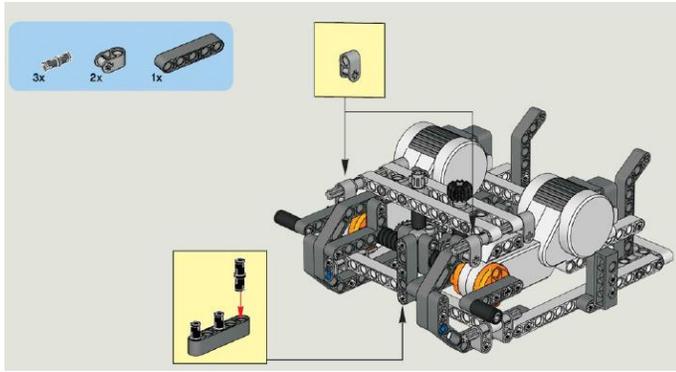


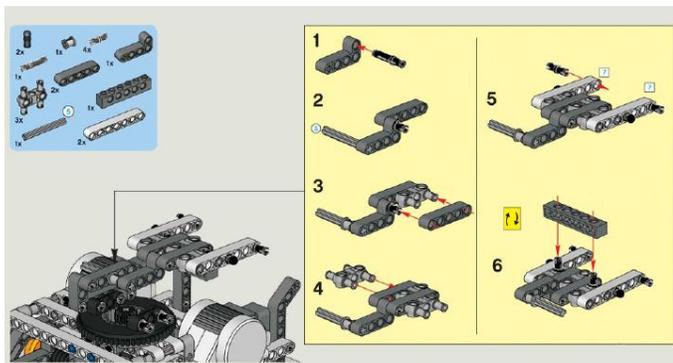
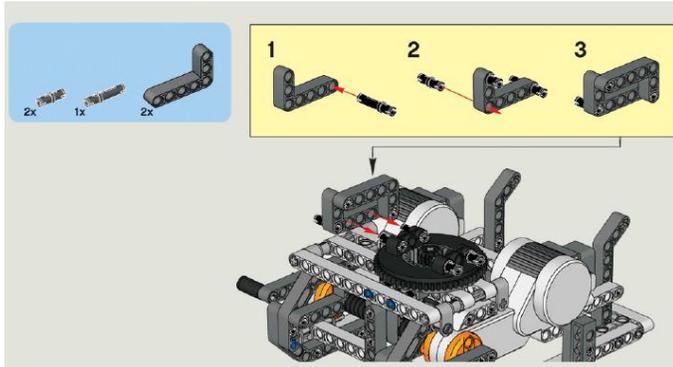
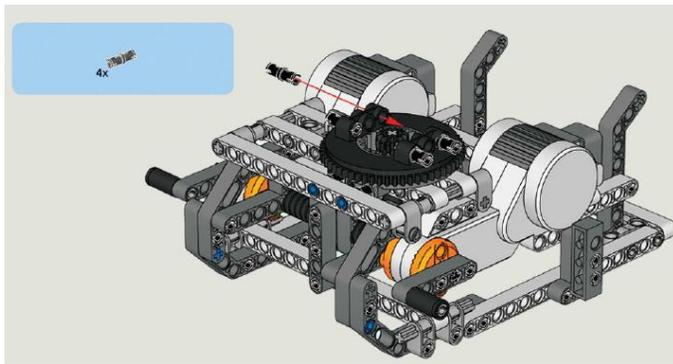
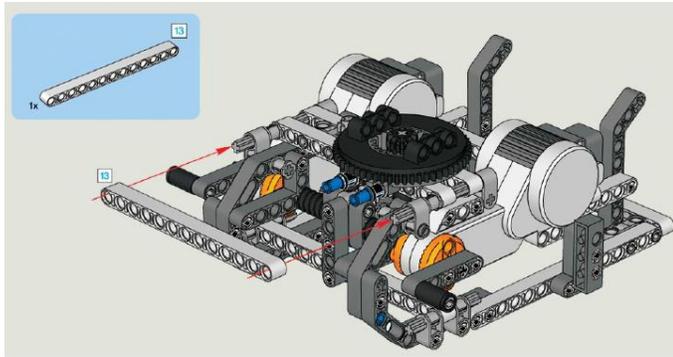


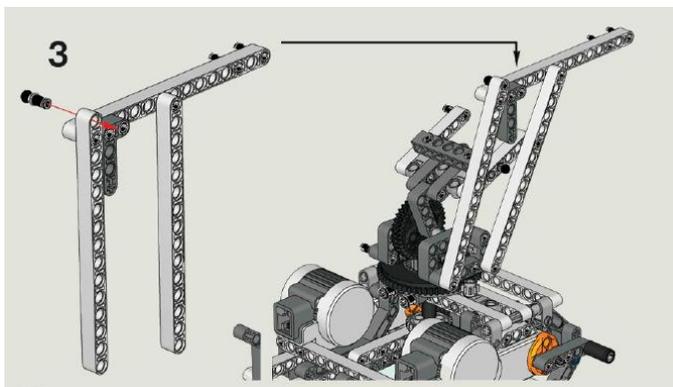
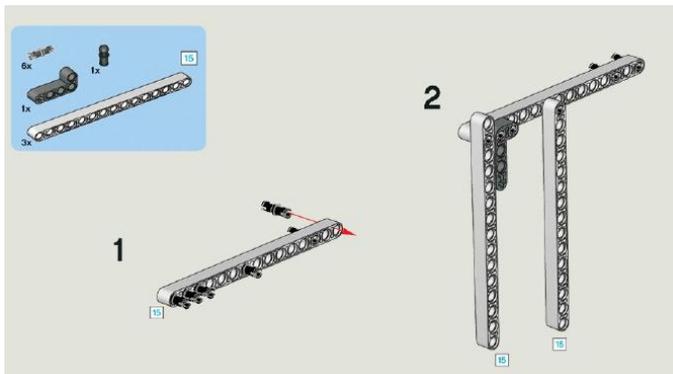
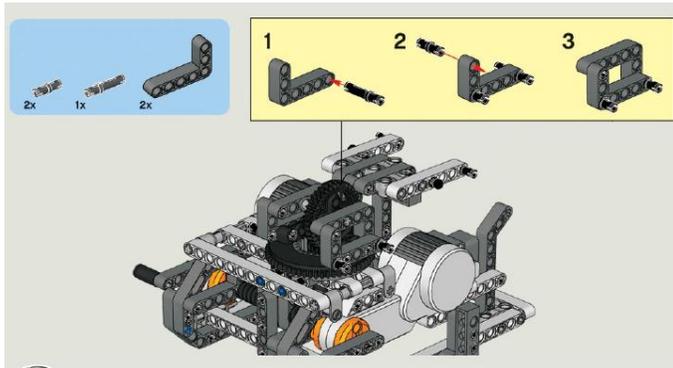
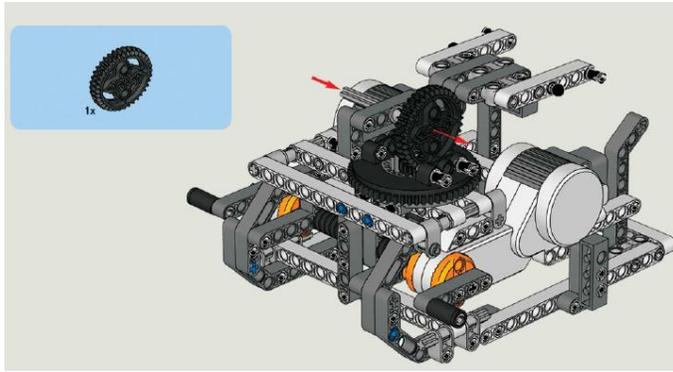




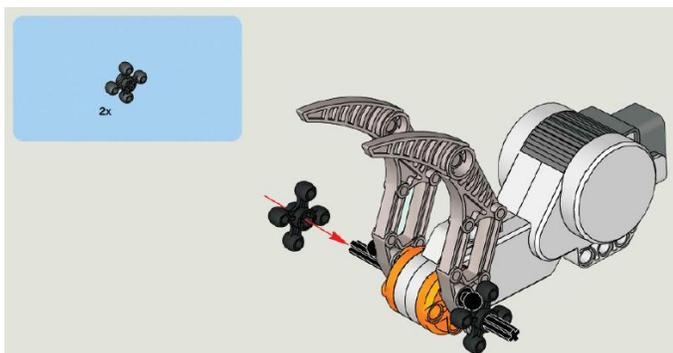
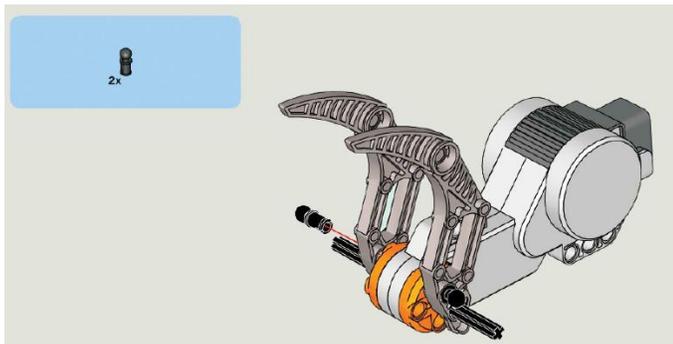
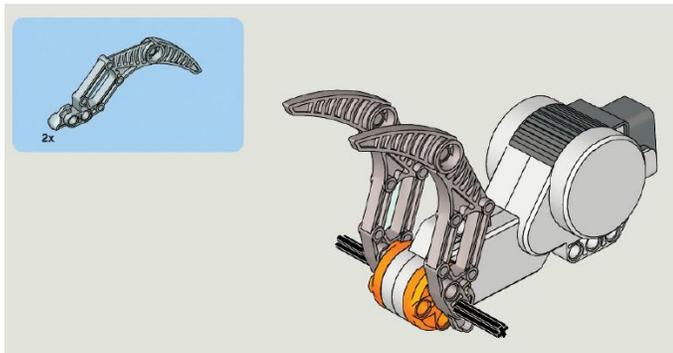
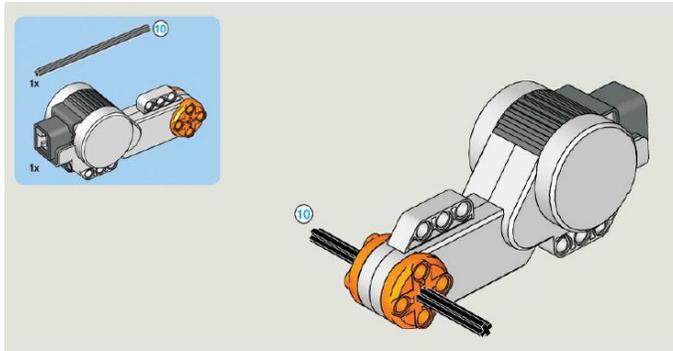




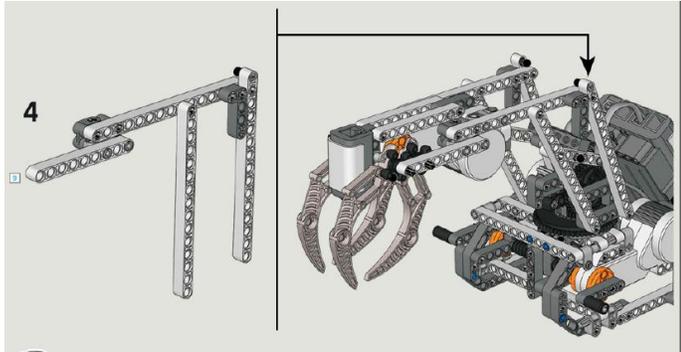
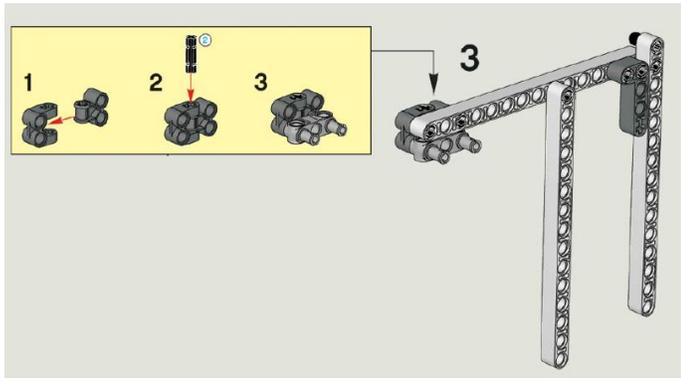
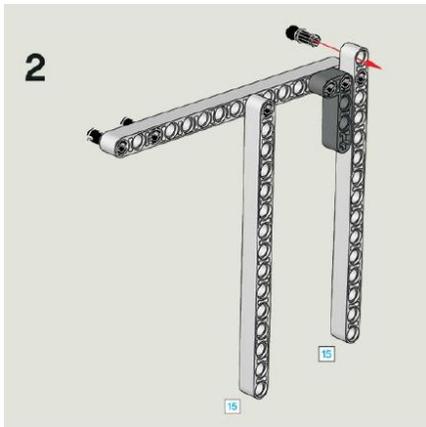




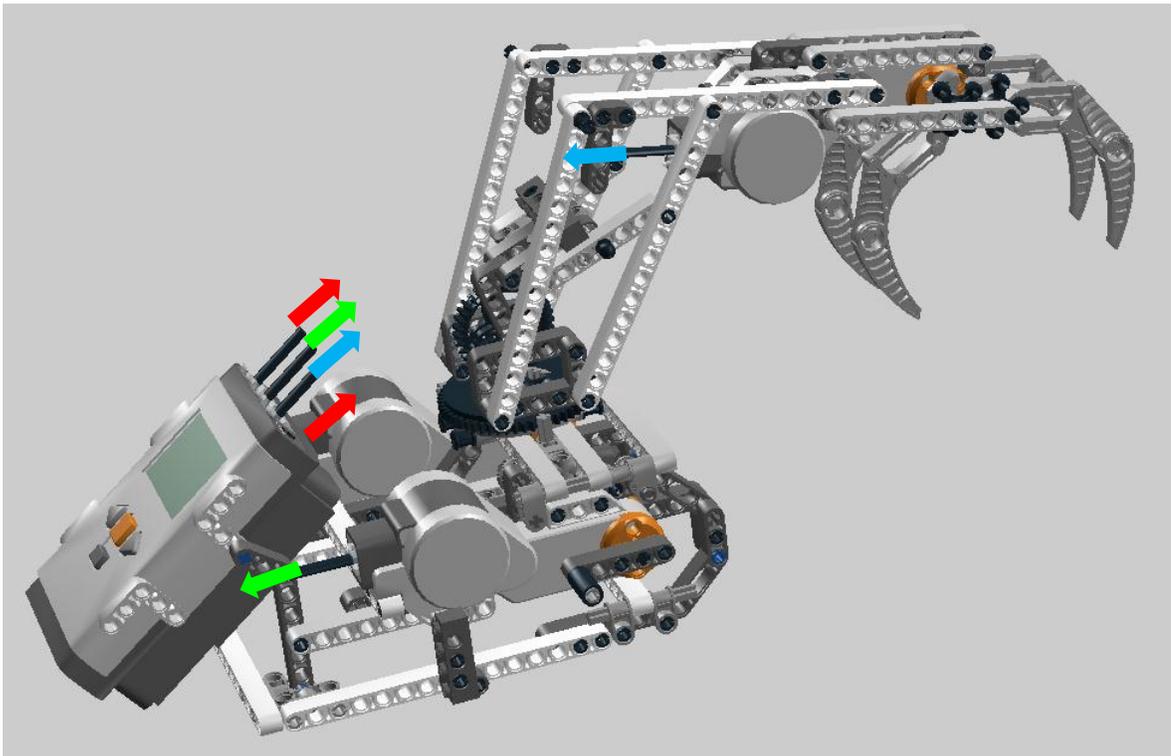








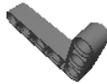
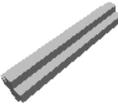
## BRAZO ROBOTICO FINALIZADO



Fuente: Investigador (LEGO Digital Designer 4.2)

### Inventario de las piezas utilizadas para el montaje del Brazo Robótico.

Bloque	Nombre	Imagen	# de Parte	Cantidad
4213607	TECHNIC BRICK 1X4, Ø4,9		3701	2
4210917	TECHNIC BRICK 1X6, Ø4,9		3894	1
6024583	CABLE 350 MM		11146	3
4297008	Tacho Motor		53787	3
6034375	NXT		53788	1
4210751	TECHNIC 3M BEAM		32523	1
4210686	TECHNIC 5M BEAM		32316	5
4297199	TECHNIC 7M BEAM		32524	6
4210667	TECHNIC ANG. BEAM 4X2 90 DEG		32140	6

4297202	TECHNIC 9M BEAM		40490	4
4297203	TECHNIC 13M BEAM		41239	3
4542578	TECHNIC 15M BEAM		32278	11
4210753	TECHNIC ANG. BEAM 3X5 90 DEG.		32526	8
4210673	TECHNIC ANGULAR BEAM 3X7		32271	2
4210668	DOUBLE ANGULAR BEAM 3X7 45°		32009	6
4109810	2M CROSS AXLE W. GROOVE		32062	2
4121715	CONNECTOR PEG W. FRICTION		2780	66
4211815	CROSS AXLE 3M		4519	1

4211622	BUSH FOR CROSS AXLE		6590	6
4184169	BALL WITH FRICTION SNAP		6628	4
4206482	CONN.BUSH W.FRIC./CROSSALE		43093	10
655826	CONNECTOR PEG W. FRICTION 3M		6558	9
4211709	3M CONNECTOR PEG		32556	2
4211086	CROSSAXLE 3M WITH KNOB		6587	4
4211639	CROSS AXLE 5M		32073	3
4211775	CROSS BLOCK 90°		6536	8
4526982	TUBE Ø7,84 2M		75535	4

4107742	2M FRIC. SNAP W/CROSS HOLE		32054	4
4211805	CROSS AXLE 7M		44294	4
370726	CROSS AXLE 8M		3707	2
373726	CROSS AXLE 10M		3737	2
4210655	TECHNIC CROSS BLOCK 2X1		32291	2
4234599	TECHNIC CROSS BLOCK/FORK 2X2		41678	2
4225033	BEAM 3 M. W/4 SNAPS		48989	13
4296059	Angular beam 90degr. w.4 snaps		55615	2
4177431	DOUBLE CONICAL WHEEL Z12 1M		32270	1

4211635	DOUBLE CONICAL WHEEL Z20 1M		32269	2
4248204	TECHNIC ANGULAR WHEEL		32072	4
4255563	DOUBLE CONICAL WHEEL Z36		32498	1
4624645	TURNTABLE 4.85		50163	1
4495192	CORRUGATED PIPE 16MM		72504	2
4252701	Tool 4x12 nr. 4		50914	4
<b>Total:</b>				<b>227</b>

Fuente: Investigador (LEGO Digital Designer 4.2)