

# Materia de Graduación

## Microcontroladores Avanzados

### Legó Mindstorms NXT

### Robot Explorador



Integrantes: Martha Aguirre  
Jonathan Cagua  
Oswaldo Criollo



- **Descripción general del sistema.**
  - Sistema de **monitoreo** de una **variable** del tipo **analógico**: ruido - presión - temperatura.
  - Consiste en un **robot explorador** que se **moverá en diferentes direcciones** evadiendo **obstáculos**, y cuando el usuario lo requiera se realice la **adquisición de datos**.



- **Antecedentes**

- Aplicaciones previas: **NXTWay-G** - **Seguidor de Línea**
- **Robot Spirit**, envía fotografías de la superficie de marte.





## ● Alcance del proyecto

- El robot explorador no tiene una aplicación específica sino un estándar el cual es el monitoreo de parámetros utilizando la innovación de la robótica.
- Con cambiar un sensor o hacer pequeñas innovaciones en el diseño se lo puede aplicar para otro campo de la industria.
- Si se reemplaza el sensor de sonido con sensores que mida el nivel de oxígeno en el aire o gases nocivos se puede aplicar en la industria minera.



## ● **Fundamento Teórico : Componentes del Lego NXT**

### ➤ **El ladrillo o cerebro del NXT posee:**

- **Un UP Atmel** de 32 bits ARM, de 256 KB de Flash, 64 KB de RAM y un reloj de 48 MHz.
- **Un UC Atmel** de 8 bits AVR de 256 Kb de Flash, 64 Kb de RAM y un reloj de 48 MHz.
- **Mayores capacidades** de ejecución de programas.
- Comunicación **vía Bluetooth**, de la clase II V2.0
- Comunicación **mediante USB** con una velocidad de 12 Mbit/s
- **4 puertos de entrada** y **3 puertos de salida**, mediante cable de 6 líneas
- **Un altavoz de 8 KHz**, con una resolución de 8-bit y una resolución de 2-16 KHz de muestreo.
- **Alimentación** es de 6 pilas AA





## ● **Fundamento Teórico : Componentes del Lego NXT**

### ➤ **Sensores posee:**

- **Servo Motores Interactivos:** permiten la detección de giros de la rueda, indicando los giros completos o medios giros, que es controlado por el software
- **Sensor Ultrasónico:** Son los "ojos" del robot, y miden la distancia, los movimientos y detectan los objetos
- **Sensor de Sonido:** Son los "oídos" del robot y permiten a los robots reaccionar antes comando de sonido y tonos
- **Sensor de Tacto:** Son los "dedos" del robot, reaccionado ante la presión o relajación de este sensor
- **Sensor Óptico:** Detecta los diferentes colores y la intensidad de luz





## ● **Fundamento Teórico**

### ➤ **Comunicación del NXT**

- Mediante la **interfaz de USB**, la cual ya viene en la versión 2.0.
- Posee una **interfaz Bluetooth** que es compatible con la Clase II v 2.0.
- La conectividad Bluetooth no tan sólo permite conectarse con otros bloques, sino también con **computadoras, palms y teléfonos móviles**, y otros aparatos con esta interfaz de comunicación.

### ➤ **Lenguajes de programación**

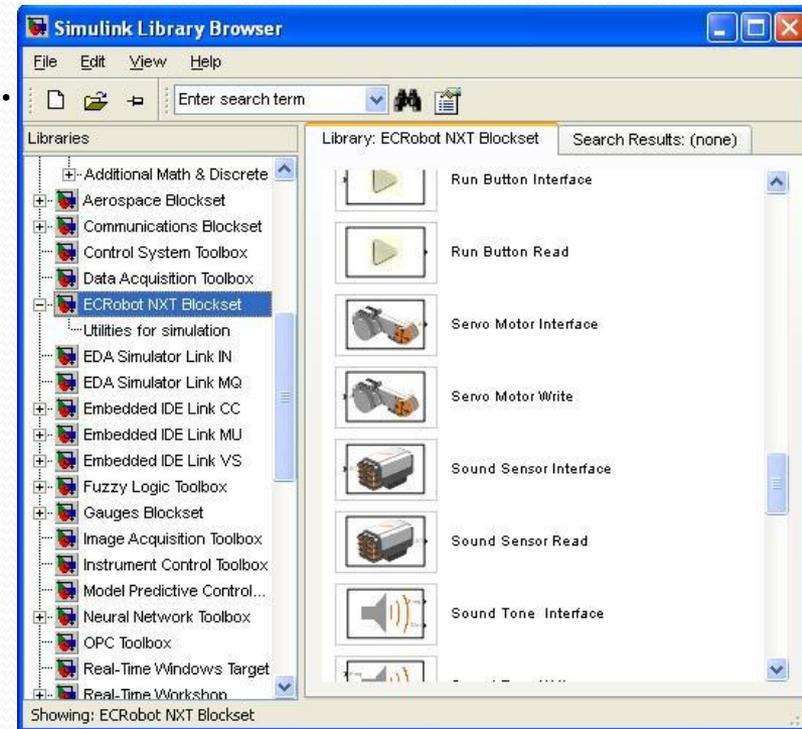
- **Posee un software propio** de programación, él se **basa en la construcción por bloques**.
- Existen varios **lenguajes** para programar los NXT, como lo son:
  - **NXC**: Lenguaje parecido a C.
  - **Leejos NXJ**: Lenguaje para java.
  - **Labview**: Se pueden desarrollar nuevos bloques.
  - **Matlab y Simulink**: Para controlar el robot vía bluetooth.
  - **NBC**: Lenguaje ensamblador.



## • Fundamento Teórico

### ➤ Simulink con Lego NXT

- **Real Time Workshop** de MATLAB: permite **generar** el código C autónomo para desarrollar y probar algoritmos modelados en **Simulink**.
- **Real-Time Workshop**: facilita la **conversión** del programa desde Simulink a un archivo de **ejecución** que luego se **carga** en el **Legó** mediante comunicación **USB**.
- **ECRobot** ofrece un **Toolbox** para la programación en bloques en **Simulink** para el control del **NXT**, hace uso del **Real-Time Workshop** y **NXTOSEK** se envía el programa al **Robot NXT**.

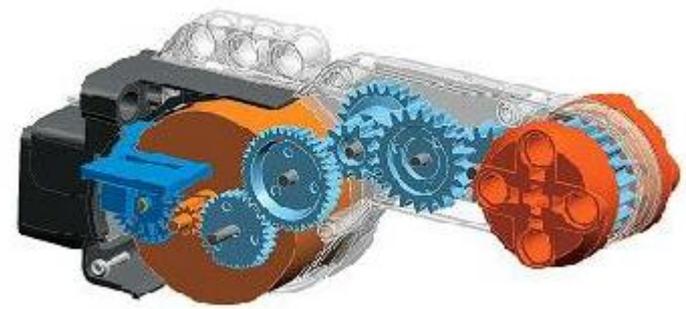




## ● **Fundamento Teórico : Componentes del Sistema**

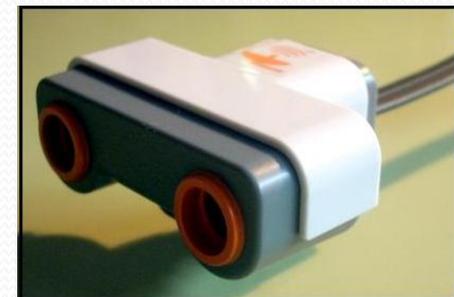
### ➤ **Servos Motores:**

- Tres **motores** de corriente **continua**.
- Funcionamiento **PWM**.
- En su **interior** un **sistema de reducción** por **tren de engranajes** y un sensor de rotación de **tipo tacométrico**.



### ➤ **Sensor Ultrasonico:**

- **Detecta** las **distancias** de un objeto que se **interponen** en el camino del robot.
- Usa el **principio** de la **detección ultrasónica**.
- **Detecta** objetos que se encuentran desde **0 a 255 cm**, con una **precisión** relativa de **+/- 3 cm**.





## ● **Fundamento Teórico : Componentes del Sistema**

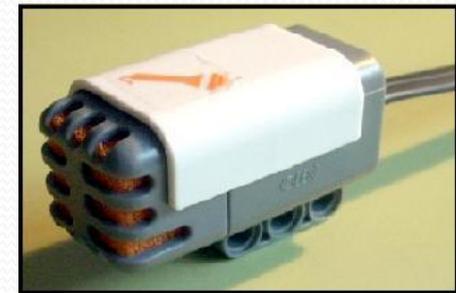
### ➤ **Sensor de Tacto:**

- Detecta si ha **colisionado** o no con algún objeto que se encuentre en su **trayectoria**.
- Al **tocar** una superficie, una pequeña cabeza externa se **contrae** y dentro del bloque cierra un **circuito eléctrico**, provocando una variación de energía de **0 a 5 V**.



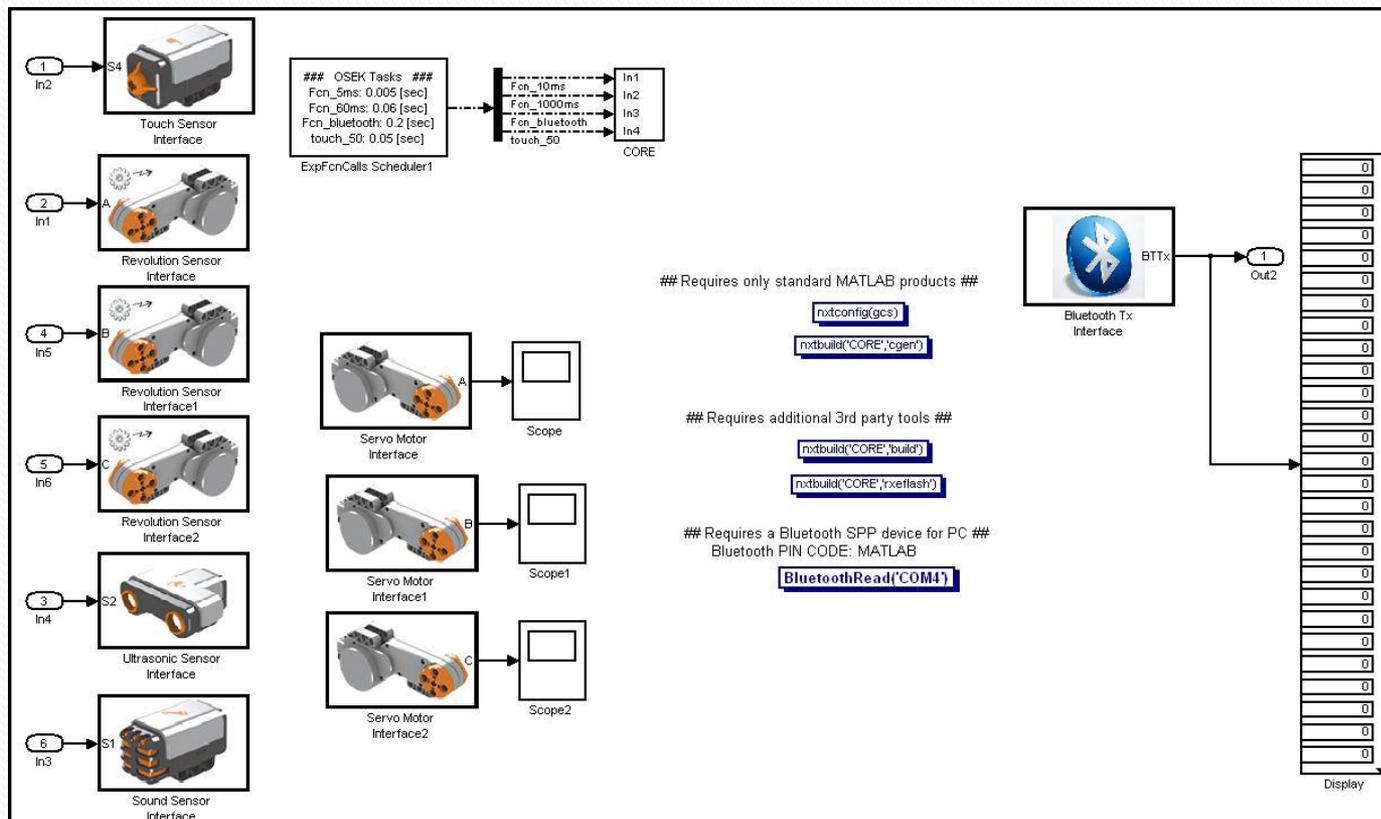
### ➤ **Sensor de Sonido:**

- **Sensa** las **variaciones** de ruido en **decibelios**.
- La **sensibilidad** máxima se encuentra en los **90 dB**.
- La **respuesta** a intensidades en decibelios es aproximadamente **exponencial**.





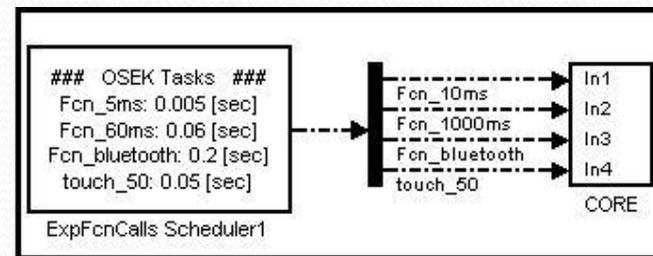
## • Descripción del programa para la movilidad del Robot en Simulink



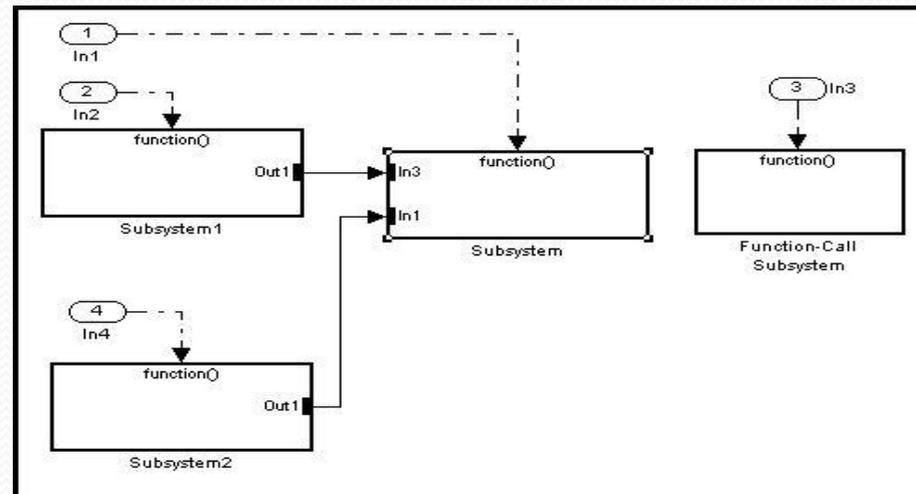


- **Descripción del programa para la movilidad del Robot en Simulink**

- **Núcleo del programa**



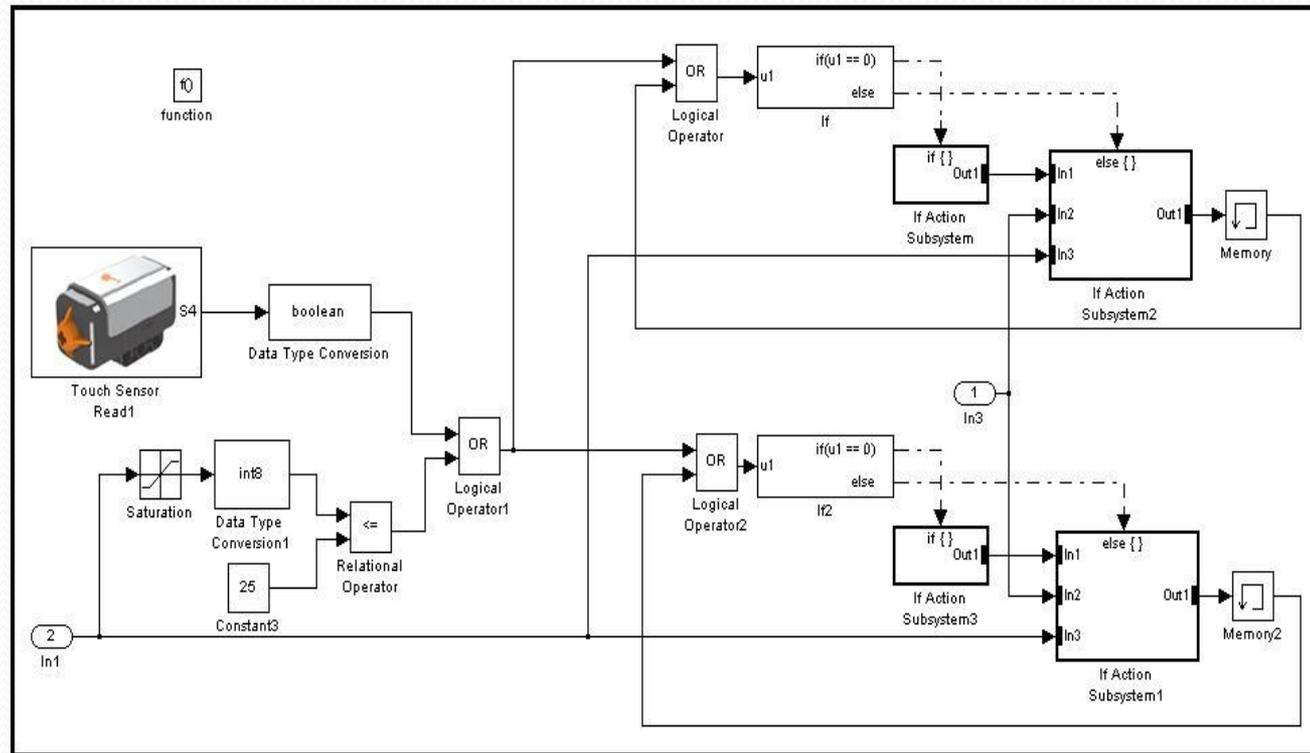
- **Bloques Internos**





- **Descripción del programa para la movilidad del Robot en Simulink**

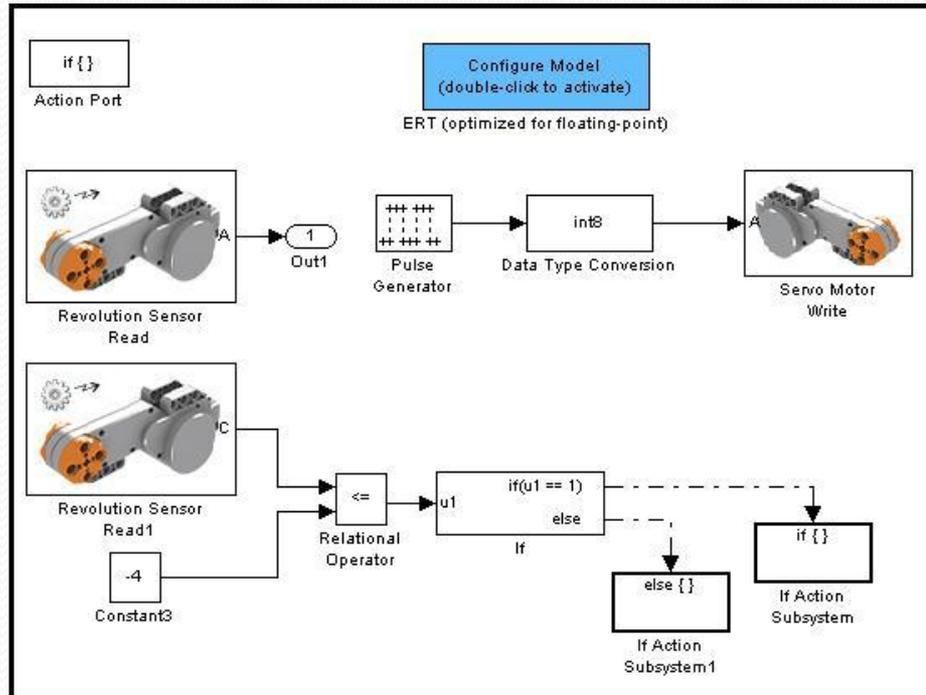
- **Detalle del programa principal de la Movilidad**



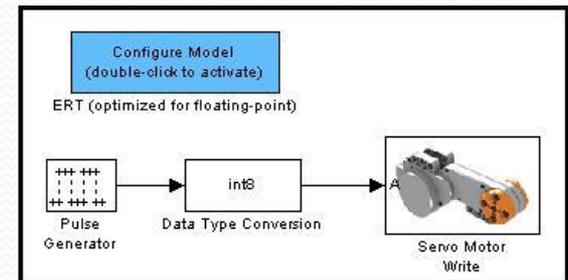


## • Descripción del programa para la movilidad del Robot en Simulink

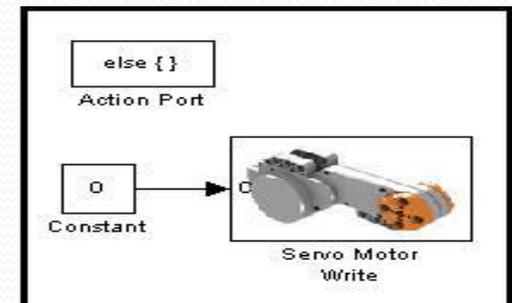
### \* Detalle de los bloques



### \* Generador de PWM



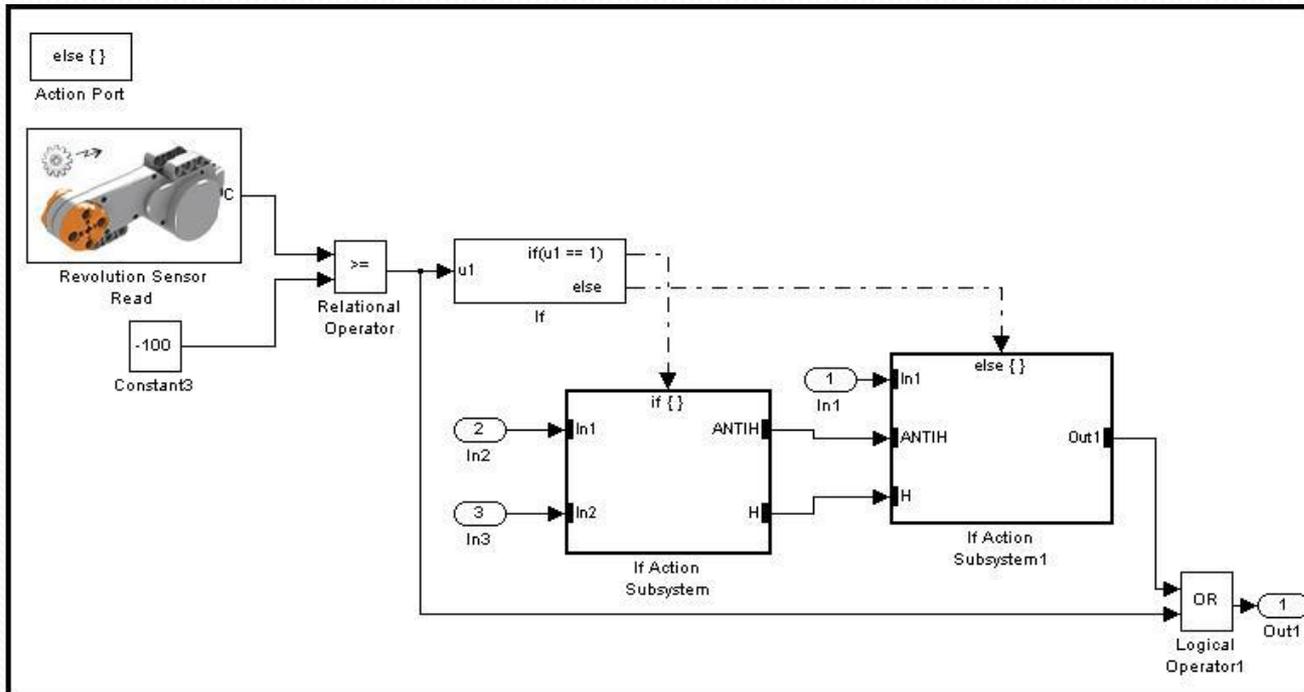
### \* Frenado de Motores





- **Descripción del programa para la movilidad del Robot en Simulink**

- \* Detalle de los bloques

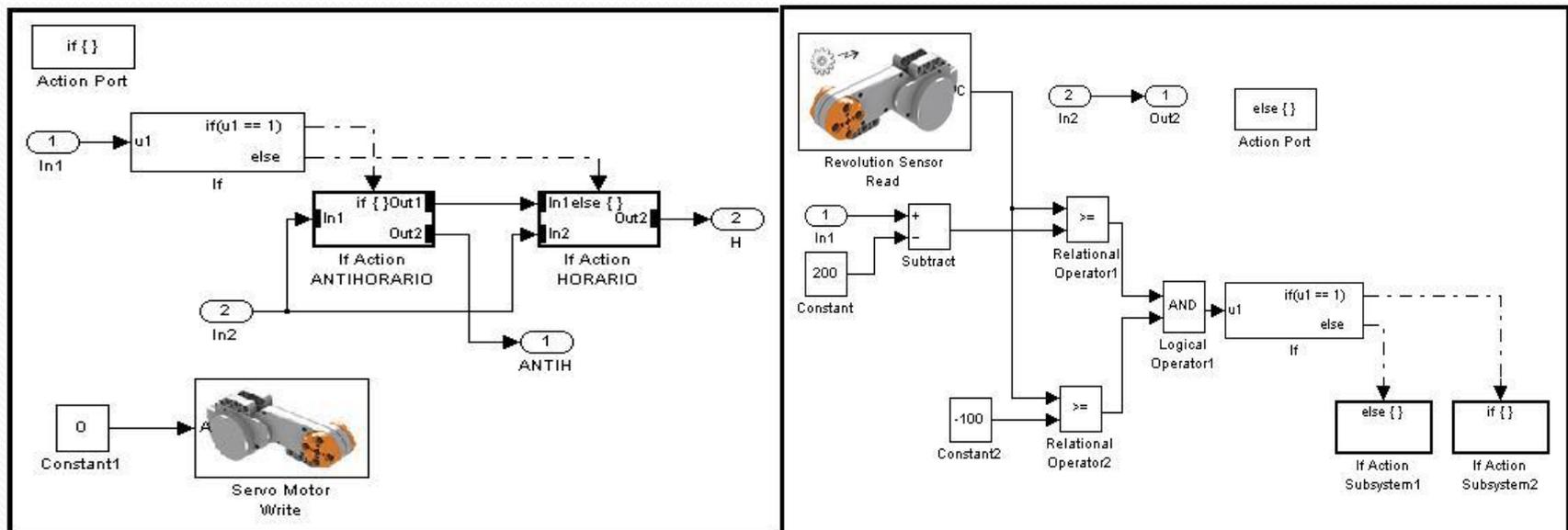




## • Descripción del programa para la movilidad del Robot en Simulink

### \* Detalle de los bloques:

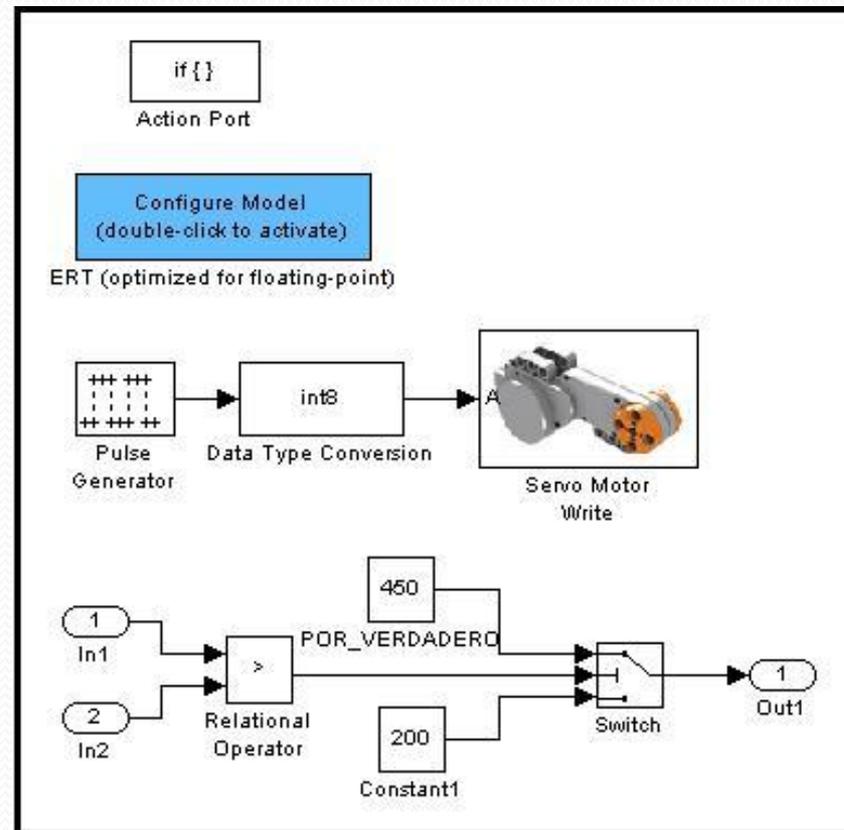
Bloque antihorario: Bloque con un generador de PWM y un lector del motor





- Descripción del programa para la movilidad del Robot en Simulink

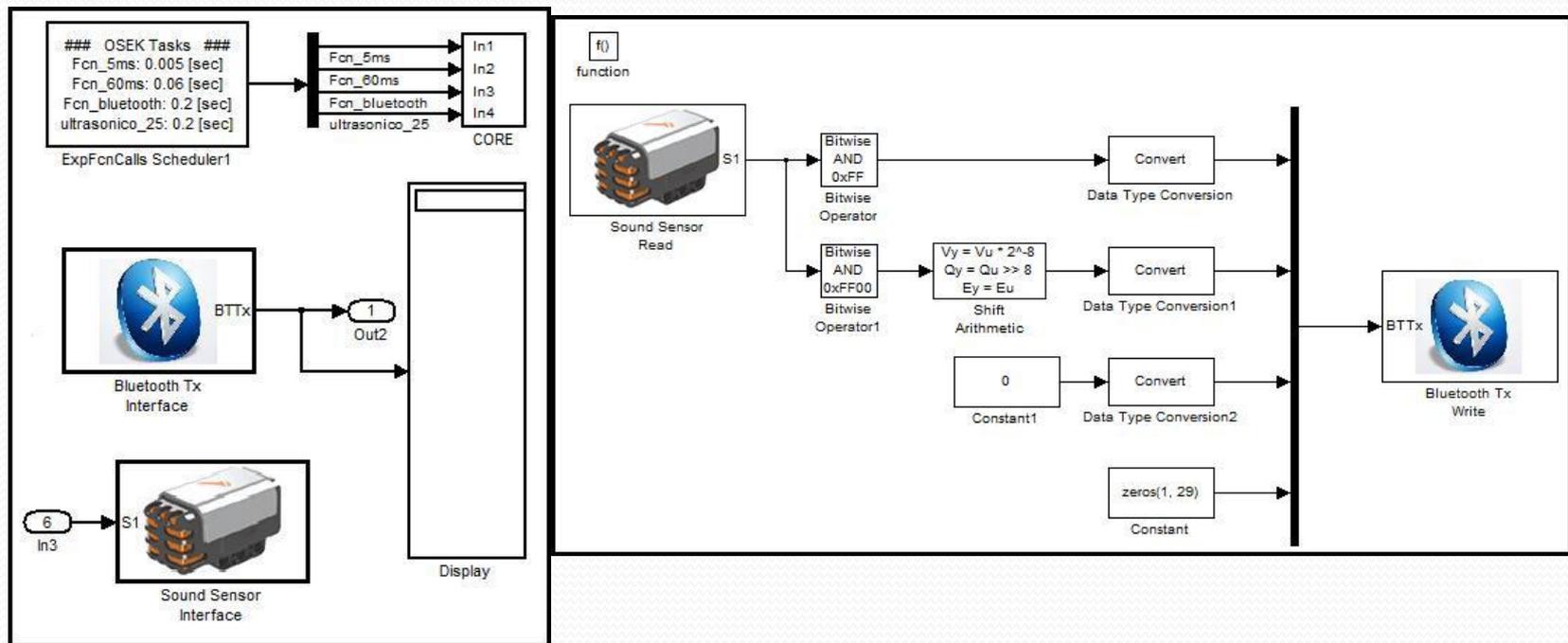
\* Detalle de los bloques:





- **Descripción del programa para la adquisición de datos.**

\* **Simulink:** trama de 32 bits y se adiciona un bit de inicio y uno bit de parada.





- **Descripción del programa para la adquisición de datos.**

\* GUI - Matlab:

Número de ciclos para la adquisición

```
disp('Ready for DAQ via Bluetooth. Run your NXT!');
%El for es usado para adquirir un numero de muestras
for i = 1:NUM_OF_CYCLE
    packet = fread(s, PACKET_SIZE, 'uint8');
    %Comparamos la longitud del tamaño de trama q adquirimos
    if length(packet) == PACKET_SIZE
        SoundSensorDAQ(i) = 1023-( packet(4)*256+packet(3));
        %UltrasonicSensorDAQ(i) = packet(5);
    end
end
%Con plot mostramos la grafica en GUI
plot(PERIOD:PERIOD:NUM_OF_CYCLE*PERIOD, SoundSensorDAQ(1:NUM_OF_CYCLE));
title('Sound Sensor A/D');
```

Lee el packet enviado vía bluetooth por el NXT

Se guarda en la variable para luego ser graficada en el AXES

Mostramos la gráfica con los datos adquiridos y guardados en SoundSensorDAQ



- Descripción del programa para la adquisición de datos.

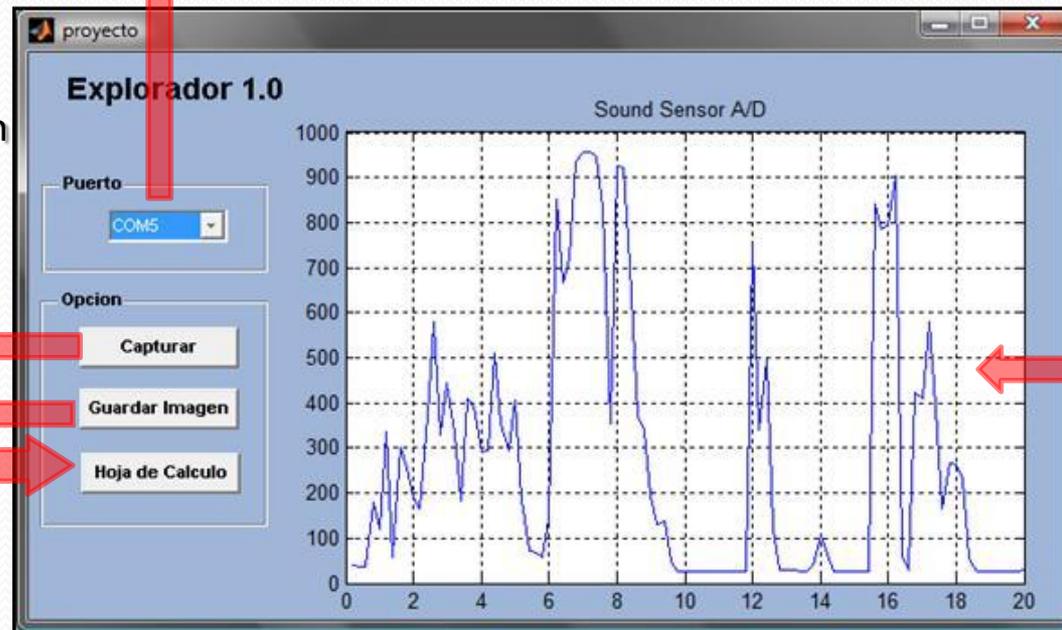
## \* Interfaz GUI:

Elige el COM de comunicación para habilitar la adquisición

Permite la adquisición durante 20 seg.

Permite guardar el resultado en un archivo .bmp

Permite guardar los datos en una hoja de Excel



Área donde se presentan los datos



## ● Conclusiones:

- La elección de **Simulink** para la programación del **Legó NXT** fue correcta, ya que el **robot explorador no tuvo una dependencia** con un **computador** para cumplir su propósito de **movilidad**.
- La **adquisición** que se realizó por las herramientas de **Matlab** y **Simulink** permitió **lectura eficiente** del **sensor del robot**, para luego con **GUI** presentar dichos **datos** en una **interfaz** que permita al **usuario** manejar las **opciones** que la **interfaz**.
- El **sistema** que se desarrolló es **básico** con un **sensor de sonido** como el **receptor** de los **datos**, pero este sensor se podría **cambiar** a otros ya sea de **presión** o **temperatura** para la **adquisición** de una variable, y así emplearlo en **otros niveles** de aplicación.

## ● Recomendaciones:

- Al **hacer** la lectura del **ultrasónico** se lo debe realizar en un **bloque diferente** ya que este **no trabajaba** eficientemente a la **misma frecuencia de muestreo** de los demás bloques, debido a que necesita un determinado **tiempo** para realizar una **lectura precisa**.
- Al **sistema** se le podría dar un **rango de exploración más elevado**, con agregarle módulos **zigbee** que alcanzan hasta **300 metros de comunicación**, mediante implementación **I2C** entre los módulos y el **Legó XT**.