

PROJECT LEAD THE WAY

PLTW

Igniting imagination and innovation through learning.

Introducción a la Programación en ROBOT-C Lego Mindstorm NXT



Developed in collaboration with





Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT





Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

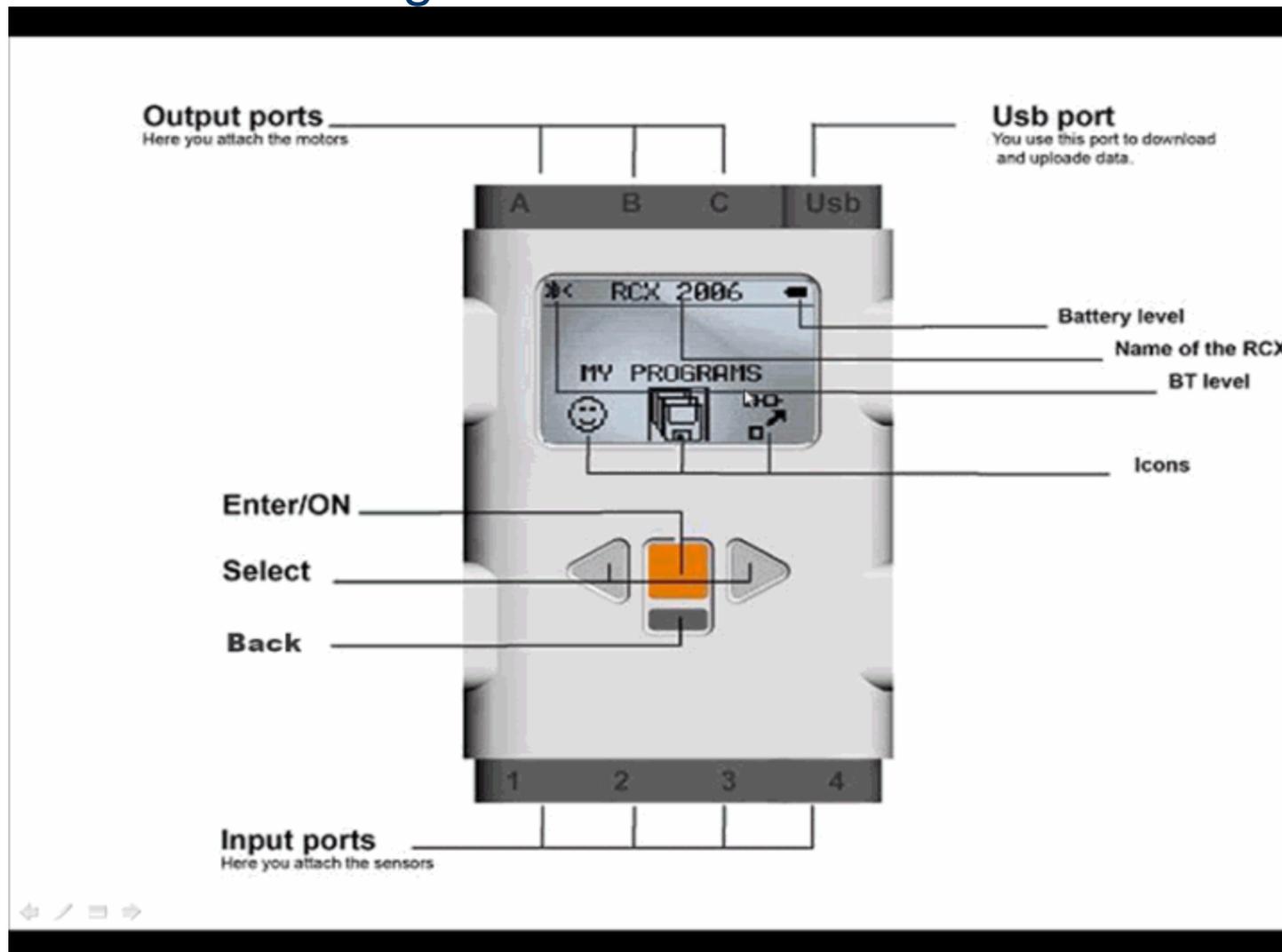
La NXT



- ARM7 de 32 bits
- 256 Kb de memoria Flash
- 64 Kb de RAM
- 4 entradas para los sensores.
- 3 Salidas de energía => Motores, Luces y Partes Móviles.



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT





Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT

SENSOR DE LUZ



- Es sin duda uno de los mas útiles de todo el kit de lego mindstorms NXT. Este sensor le permite al robot distinguir entre luz y oscuridad, midiendo la intensidad de la luz la cual le permite ver en blanco y negro.



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT

Sensor de sonido



- El sensor de sonido es capaz, mediante el uso del ladrillo inteligente LEGO MINDSTORMS NXT, de medir niveles de sonido en dBs (decibelios) y dBAs (frecuencias entre 3 y 6 kHz). También es capaz de reconocer patrones de sonido e identificar diferencias de tono. Se necesita un cable de conexión, que se incluye en la caja Base 9797.



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT

SENSOR ULTRASONICO



- El sensor ultrasónico genera ondas de sonido de alta frecuencia y lee los retardos de sus ecos para detectar y medir la distancia de objetos. Los estudiantes pueden diseñar un sistema de motorización de tráfico



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT

SENSOR TACTIL



Conocido como sensor de choque.

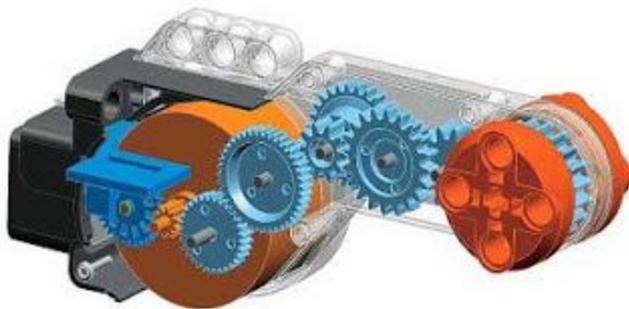
Devuelve un 0 cuando no está presionado y un 1 mientras esté presionado.

A veces, se usa como botón de arranque, haciendo que el programa comience o pare al pulsar sobre el.



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT SERVO-MOTORES (motores paso a paso)





Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT OTROS SENSORES





Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Partes del Robot NXT CABLES CONEXIÓN





Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Tarea / procedimiento / función

función NXT motor

Operador de asignación de valor

```
1 task main()  
2 {  
3  
4     motor[motorC] = 100;  
5     wait1Msec(3000);  
6  
7 }
```

CODIGO
Robot C

Indicadores de
comienzo y fin de
estructura

procedimiento

Indicador de final de
sentencia



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

```
1 task main()  
2 {  
3  
4     motor[motorC] = 100;  
5     wait1Msec(3000);  
6  
7 }
```

ESTILO de
PROGRAMACIÓN
CORRECTO

```
1 task main() {motor[motorC  
2 ]=0;wait1Msec(3000);}
```

ESTILO de PROGRAMACIÓN INCORRECTO



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

```
task main()  
{  
    motor[motorC]=100;  
    motor[motorB]=100;  
    wait1Msec(5000);  
  
    //motor[motorC]=100;  
    //motor[motorB]=-100;  
    //wait1Msec(1500);  
}
```



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Variables

```
task main()  
{  
    int speed;  
  
    speed = 75;  
  
    motor[motorC] = speed;  
    motor[motorB] = speed;  
    wait1Msec(2000);  
}
```

Definición previa
especificando su TIPO

Asignación de un
valor inicial

Uso de ese valor a
través del nombre
dado a la variable



Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Variables

Tipo de dato	Descripción	Ejemplo	Código
Entero	Positivos y negativos incluyendo el 0	-35, -1, 0, 33, 345,14567	int
Decimal	Números con decimales, aunque sea cero	-.123, 0.56, 3.0, 1000,46	float
Texto	Cadena de caracteres, con espacios en blanco y símbolos	“Se encontró línea negra” “STOP !”	string
Lógico	Verdad o falso, 1 o 0	true, false	bool



Introducción a la Programación en ROBOT-C

Lego Mindstorm NXT

Variables

Reglas de escritura de nombre de variables

- ✓ No puede tener espacios en blanco
- ✓ No puede tener símbolos
- ✓ No puede comenzar por número
- ✓ No se puede usar el nombre de una palabra reservada

CORRECTO	INCORRECTO
cuentalneas	cuenta lineas
hueco	hueco!
distancia3	3distancia
vmotor	motor



Introducción a la Programación en ROBOT-C

Leao Mindstorm NXT

Ejemplos de asignaciones correctas

Sentencia	Descripción
<code>VelocidadMotor = 50;</code>	Asignamos un valor entero a una variable (int)
<code>CantidadDeLuz = SensorValue(lightSensor);</code>	La función <code>SensorValue</code> pregunta al sensor de tipo luz cuanto está midiendo actualmente y ese valor se guarda en la variable <code>CantidadDeLuz</code> (int)
<code>Suma = variable1 + variable2;</code>	Se suma el contenido de <code>variable1</code> con el de <code>variable 2</code> y el resultado se guarda en la variable <code>Suma</code> (int)
<code>Media = Suma / 2;</code>	Se divide <code>Suma</code> entre dos para calcular una media y se guarda en la variable <code>Media</code> (float)
<code>contador = contador + 1;</code>	Incrementamos una variable en una unidad y guardamos el valor actualizado dentro de la misma variable
<code>int cero = 0;</code>	Definimos una variable de tipo entero y de paso, la inicializamos con un valor.

Introducción a la Programación en ROBOT-C Lego Mindstorm NXT

Estructura repetitiva: WHILE

```
while (condition)  
{  
    // repeated-commands  
}
```

CONDICIÓN

CONJUNTO DE
INSTRUCCIONES
QUE QUEREMOS
SE REPITAN

```
task main()  
{  
    while (nMotorEncoder[motorC] < 360)  
    {  
        motor[motorC] = 100;  
        motor[motorB] = 100;  
    }  
}
```

CONDICIÓN

CONJUNTO DE
INSTRUCCIONES
QUE QUEREMOS
SE REPITAN

Introducción a la Programación en ROBOT-C

Lego Mindstorm NXT

REPEAT con WHILE

Usaremos un contador para establecer la **CONDICIÓN** de repetir un conjunto de instrucciones

```
int vueltas=0;  
while(vueltas<=2)  
{  
    .....  
    .....  
    .....  
    vueltas=vueltas+1;  
}
```

Introducción a la Programación en ROBOT-C

Lego Mindstorm NXT

REPEAT con WHILE

Usaremos un contador para establecer la **CONDICIÓN** de repetir un conjunto de instrucciones, por ejemplo, dos veces.

```
int vueltas=1;  
while(vueltas<=2)  
{  
    .....  
    .....  
    .....  
    vueltas=vueltas+1;  
}
```

La condición

Todo lo que esté entre las llaves del “while”, se ejecutará mientras que la variable “vueltas” tenga como valor un número menor o igual a 2.

“vueltas” vale 1 cuando se entra por primera vez en el “while”, se ejecutan las órdenes y se incrementa la variable en 1. como entonces “vueltas” vale 2, y sigue cumpliendo la condición, se repetirá todo.
La segunda vez, se incrementa la variable en 1 y ahora valdrá 3, con lo que al no cumplir la condición, se saldrá sin ejecutar las órdenes.