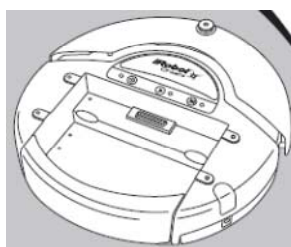


PRÁCTICA 2 (Parte I)

Conexión Raspberry Pi 2 – PC y programación mediante comandos Create

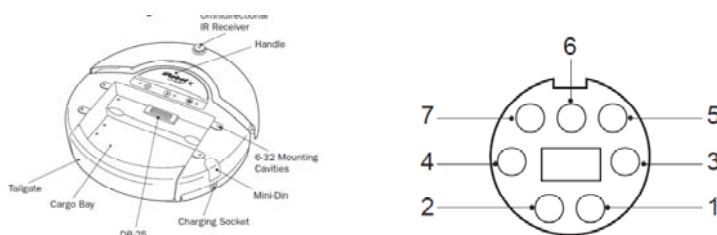


Introducción

- **OBJETIVOS**
 - Conocer la arquitectura y la programación a bajo nivel del iRobot Create.
 - Entender la conexión de la Raspberry Pi2 con el iRobot (alimentación y línea serie).
 - Programar el iRobot mediante comandos numéricos y scripts.
 - Enviar comandos al iRobot desde la Raspberry Pi a través de NetBeans.
- **MATERIAL NECESARIO**
 - **Hardware**
 - iRobot Create
 - Cable serie PC-iRobot
 - Cables de alimentación y conexión serie Raspberry Pi2 <---> iRobot
 - **Software**
 - Realterm
 - Free Hex Editor Neo
 - **Manuales (en eGela)**
 - Guía del usuario: iRobot Create Owner's Guide
 - Manual del lenguaje de comandos: iRobot Create Open Interface.

Conexión del iRobot al PC mediante el cable serie, en Windows 7

- Conectar el **cable serie iRobot** al conector Mini-DIN del puerto serie del iRobot Create y por el otro lado al cable Serie-USB.
 - Conectar este último a una entrada USB del PC.
 - Encender iRobot Create.
- El cable serie iRobot transforma el voltaje lógico de RXD y TXD (0v-5v) al voltaje requerido por el puerto serie RS232C del PC.



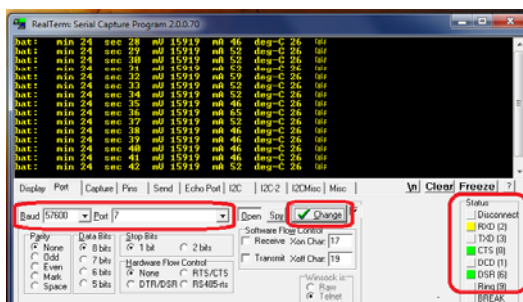
Conexión del iRobot al PC mediante el cable serie, en Windows 7

- Desde Windows: Inicio → “dispositivos e impresoras”:
- Aparecerá un dispositivo de nombre: “Prolific USB-to_Serial Comm Port” y el nombre de puerto entre paréntesis.
- Leer el puerto asignado



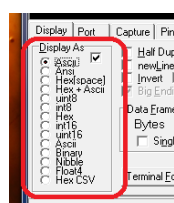
Configuración y uso de Realterm

- Realterm es un terminal virtual diseñado para capturar, controlar y depurar flujos de datos binarios y depurar comunicaciones.
- Abrir Real Term como administrador
 - Configurar para comunicarse por la línea serie con el iRobot usando la pestaña **PORT**.
 - Seleccionar el puerto COM en el que está enlazado el iRobot en Windows y cambiar la velocidad a 57600 Baudios.
 - Pulsar en el botón “Change” para aplicar los cambios.



Visualización de los datos de iRobot desde RealTerm

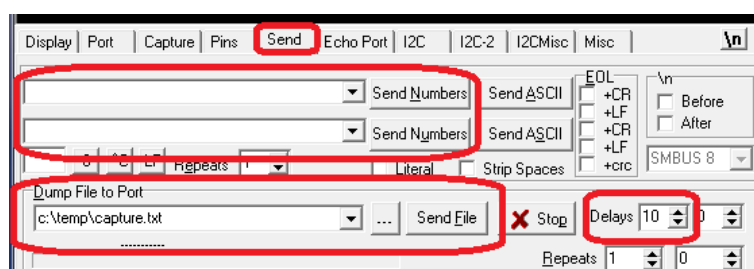
- En la pestaña **DISPLAY** están los modos de visualización de los datos que se reciben por línea serie.
- Para verlos elegir el modo “Binary”, “int16” o “uint16” en función de la codificación del sensor **NO ASCII**.



Sensor	Codificación
Bumps & WheelDrops	Máscara de bits
Wall, Cliffs x 4, Virtual Wall	1 bit value (El menos significativo)
Infrared	1 Byte [0,255]
Distance, Angle, Requested Radius	Signed 16 bit value [-32768 - 32768]
Wall Signal, Cliffs x 4,	Unsigned 16 bit [0 - 4095]
Requested Velocity x3	Signed 16 bit value [-500, 500]

Envío de Comandos

- Usar el botón “Send Numbers” (en la pestaña **SEND**) para mandar los códigos de los comandos del robot.
- Podemos cargar un Script en el cuadro de opciones “Dump File to Port” y enviarlo
- Previamente configurar un “Delay” de 10 ms entre código y código.
- Los archivos que se envían mediante la opción “Dump File to Port” de RealTerm solo deben contener los códigos OI de iRobot y estar editados desde un editor hexadecimal.



Ejemplo de comandos

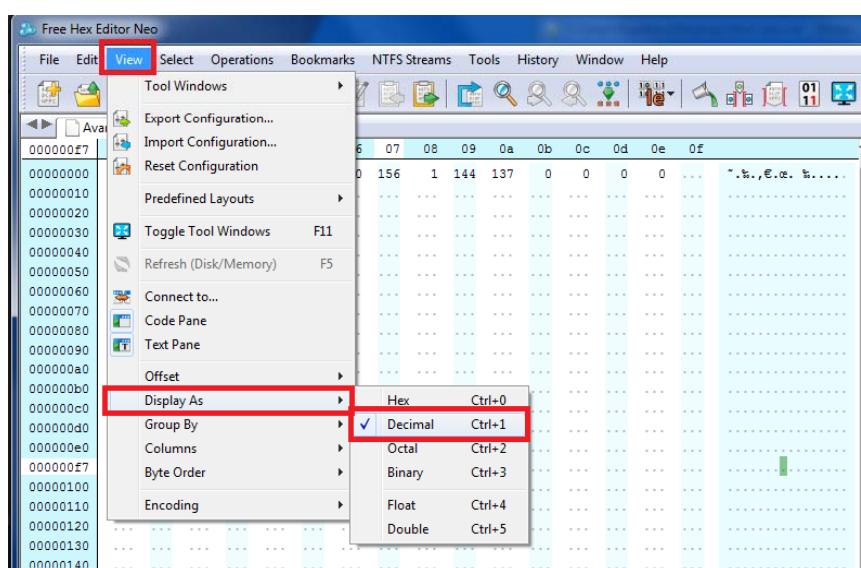
- Probad los siguientes comandos:
 - Activar el LED de PLAY: 128 132 139 2 0 0
 - Leer el estado del sensor de desnivel de la izquierda: 128 142 9 (observar cómo cambia la respuesta cuando se levanta el robot)
 - Para hacer música con iRobot: enviar estos conjuntos de comandos por separado:
 - 128 132 (Pone el robot en el modo completo (full))
 - 140 0 4 62 12 66 12 69 12 74 36 (define la canción)
 - 141 0 (toca la canción)
- Poned el robot sobre la caja de plástico para que no se mueva al poner en marcha los motores y enviar los siguientes comandos:
 - Para hacer avanzar hacia adelante al iRobot: 128 132 137 0 100 128 0
 - Para detener al iRobot: 128 132 137 0 0 0 0
- Probar los comandos del documento iRobot Create Open Interface.
 - En especial los que activan los motores: Drive y Drivedirect.

Programación con scripts. Hex Editor Neo

Un script es una serie de comandos que se ejecutan secuencialmente. Se pueden hacer programas complejos agrupando los comandos en scripts.

- Un script consta de comandos OI y puede tener hasta 100 bytes de longitud.
- El primer código debe ser **153** y el segundo, el **número de bytes** que siguen dentro del script (entre 0 y 100).
 - Se usa la longitud de 0 para borrar la secuencia de comandos actual.
 - El resto de bytes son comandos del Open Interface y bytes de datos.
- La secuencia debe ser: [152] [Longitud de secuencia de comandos] [Opcode 1] [Opcode 2] [Opcode 3] etc.
- El comando 152 sólo graba el script en memoria, pero no lo ejecuta.
- Para que se ejecute hay que enviar un 153 después del script. Si se mete el 153 en el propio script, este ciclará infinitamente. Pero necesitará otro 153 fuera del script para empezar a ejecutarse.
- Observad que el primer script se ejecuta una vez y el segundo no (salvo que se añada otro 153 y entonces cicla):
 - 152 **12** 145 0 80 0 80 155 20 145 0 00 0 00 153
 - 152 **13** 145 0 80 0 80 155 20 145 0 00 0 00 153
- Los scripts contenidos en ficheros de texto se envían al iRobot mediante "Dump File to Port".
 - Es necesario que los scripts sean de texto puro, sin caracteres de control.
 - Como la mayoría de editores (ej. NotePad) incluyen caracteres de control, usamos el editor hexadecimal "**Free Hex Editor Neo**" que dispone de la posibilidad de codificar los caracteres mediante un valor entero.

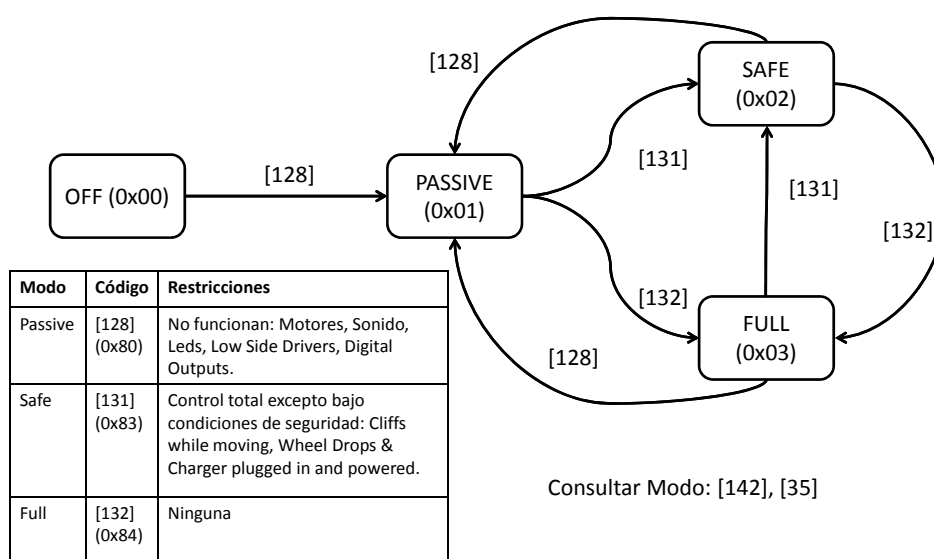
Programación con scripts. Hex Editor Neo



Actividades

- Editar con Free Hex Editor Neo los siguiente scripts que aparecen en la página 15 del manual iRobot Create Open Interface y ejecutarlos:
 - Drive 40cm and stop
 - Toggle led and bump
 - Drive in a square
- **Diseñar un script propio y ejecutarlo.**
 - **Puntúa la variedad de comandos usados y la creatividad.**
 - **Cuando funcione, enseñádselo al profesor**
 - **Añadid el listado en el informe de la práctica, con comentarios.**
- Parte voluntaria:
 - Diseñar un script que permita leer los códigos IR enviados por la base de carga y el mando a distancia.
 - Cuando funcione, enseñádselo al profesor
 - Añadid el listado en el informe de la práctica, con comentarios.

Modos de funcionamiento (OI Modes)



Actuadores: Drive vs. Drive-Direct

- Existen dos formas distintas de mover el robot:
 - Drive mueve el robot a una velocidad y con un radio de giro.
 - Drive-Direct controla cada rueda del robot de forma independiente.

Modo	Comandos	Rango
Drive	[137] Velocity 16bits Radius 16 bits	[-500 – 500 mm/s] [-2000 – 2000 mm]
Drive Direct	[145] V. Rueda Der. 16 bits V. Rueda Izq. 16 bits	[-500 – 500 mm/s] [-500 – 500 mm/s]

- Los parámetros de velocidad y Radio están codificados en “Complemento a 2”:
- Ejemplo “Velocidad máxima”:
 - +500 : 0x01 0xF4 || [1] [244] || 0000 0001 1111 0100
 - 500 : 0xFE 0x0B || [254] [11] || 1111 1110 0000 1011
- Valores por debajo de 30/-30 mm/s pueden no ser suficientes para mover el robot

Scripting + Eventos

- Scripts [152 – 153 - 154]:
 - Specify script [152] + bytes_script + commands*
 - Play Script [153]
 - Show Script [154]
- Eventos - Espera [155 – 156 – 157 – 158]
 - Durante la espera, el robot no cambia de estado, ni responde a los inputs, o línea serie. **Usar solo para Scripts**
 - Por Tiempo [155]: 1 Byte: 0 – 255 decenas de segundo
 - Por Distancia [156]: 2 Bytes: -32767 – 32767 mm
 - Por Ángulo [157]: 2 Bytes: -32767 – 32767 grados
 - Por Evento [158]: Código de evento

Sensores

- Comando para leer sensores [142] [packet id]
 - (página 13, 17-25 del manual Open Interface)
 - Ejemplo: [142] [7] para compobar Bumps and WheelDrops (registro de 8 bits)

Bumps and Wheel Drops

Bit	7	6	5	4	3	2	1	0
Sensor	n/a	n/a	n/a	Wheel-drop Caster	Wheel-drop Left	Wheel-drop Right	Bump Left	Bump Right

- Streaming de sensores [148]

Codificaciones Sensores iCreate

Sensor	Codificación
Bumps & WheelDrops	Máscara de bits
Wall, Cliffs x 4, Virtual Wall	1 bit value (El menos significativo)
Infrared	1 Byte [0,255]
Distance, Angle, Requested Radius	Signed 16 bit value [-32768 - 32768]
Wall Signal, Cliffs x 4,	Unsigned 16 bit [0 – 4095]
Requested Velocity x3	Signed 16 bit value [-500, 500]