

# Trabajo práctico final

## Robot recolector de residuos

### Protocolo de comunicación

10 de enero de 2010

#### Resumen

En el presente se establece el protocolo de comunicación entre para el envío y recepción de comandos hacia los controladores de motores, servos y sensores que proveen información del entorno al robot recolector de residuos. El protocolo está diseñado para ser transmitido a través *RS-232* utilizando la configuración de *Daisy-Chain* entre las distintas placas controladoras.

**Palabras Clave:** *Robot, residuos, protocolo, serial, rs-232, daisy chain, motor, servo, sensor, telémetro, ultrasonido, distancia, batería.*

## 1. Introducción

TODO

## 2. Formato del paquete

El paquete consta de un header común con datos que identifican el emisor y receptor del paquete, el comando a enviar y posibles datos extras que sean requeridos. Tanto los paquetes de envío de datos como los de respuesta tienen el mismo formato y comparten el valor en el campo de comando.

### 2.1. TIPO

El comando consta de 1 byte. Informa el grupo del destinatario del paquete. Grupos predefinidos en el listado 4. Bit 7: Broadcast a todos los IDs

TIPO	ID	TIPO	ID	COMANDO	LARGO	DATO	CRC
------	----	------	----	---------	-------	------	-----

Cuadro 1: Formato y header del paquete de datos

de un mismo grupo. Bits 6-0: Grupo al que va dirigido el paquete. El grupo 0x7F esta reservado para hacer broadcast a todas las placas sin importar el grupo (Grupo 0xFF).

## **2.2. ID**

El comando consta de 1 byte. Marca el número de id de la placa de destino Si el bit 7 del campo *TIPO* de destino esta activo, el *ID* se ignora.

## **2.3. TIPO**

El comando consta de 1 byte. Determina el grupo del emisor del paquete para la respuesta del paquete. Grupos predefinidos en el listado 4. Bit 7: El 0 es mandatorio, la respuesta debe tener un único destinatario. Bit 6-0: Grupo de origen del paquete. El grupo 0x7F esta reservado para hacer broadcast a todas las placas sin importar el grupo y no debe ser utilizado como origen del paquete.

## **2.4. ID**

El comando consta de 1 byte. Es mandatorio, debe estar informado para una correcta respuesta. En caso contrario, se toma como valor por defecto el del controlador principal, en este caso, la PC. Informa a que número de id de la placa debe ser dirigida la respuesta del paquete.

## **2.5. COMANDO**

El comando consta de 1 byte. Comando enviado al destino, que puede o no tener datos en el campo *DATO*. Definidos en la sección 3.

## **2.6. LARGO**

El comando consta de 1 byte. Largo en bytes del campo *DATO*. En caso que el largo del campo *DATO* sea cero, igual debe ser informado con 0x00.

## **2.7. DATO**

El comando consta de *LARGO* bytes. Contiene los parámetros o datos extras que puedan ser necesarios para el comando enviado. En el caso que el comando no los requiera, el campo debe ser nulo y el largo será 0x00.

## **2.8. CRC**

El comando consta de 1 byte. Cálculo de CRC sobre el paquete enviado.  
-Algoritmo a ser determinado-

### 3. Posibles comandos

El campo *COMANDO* determina la acción que debe realizarse en el destinatario o la respuesta al comando recibido. El rango para los comandos comunes a todos los grupos de tarjetas son desde 0x00 hasta 0x3F. Los comandos específicos para cada grupo deben ser desde 0x40 hasta 0xFF.

- *INITIALIZE*
- *RESET CARD*
- *PING*
- *ERROR*

#### 3.1. INIT

Sincroniza el inicio de todas las placas en la cadena. Debe ser recibido por la placa para inicializarse y poder informar al controlador principal de su existencia.

##### Comando enviado

- COMANDO: 0x01
- DATO: vacío

##### Respuesta al comando

- COMANDO: 0x01
- DATO: Descripción de la placa en texto plano

#### 3.2. RESET

Pide el reset de la tarjeta

##### Comando enviado

- COMANDO: 0x02
- DATO: vacío

##### Respuesta al comando

Sin respuesta

### 3.3. PING

Envia un ping a la placa

#### Comando enviado

- COMANDO: 0x03
- DATO: vacío

#### Respuesta al comando

- COMANDO: 0x03
- DATO: vacío

### 3.4. ERROR

Informa que ha habido un error.

#### Comando enviado

- COMANDO: 0x04
- DATO: 1 byte con el código de error y la descripción del error en texto plano.

#### Respuesta al comando

Sin respuesta.

## 4. Comandos específicos

Cada grupo de placas tiene comandos propios y específicos dependiendo de la función que deban desempeñar en el sistema. Existen grupos con comandos predefinidos como se aprecia en el listado 4. El rango para nuevos grupos se extiende hasta el valor 0x7E.

- *MAIN CONTROLLER* - sección 5
- *DC MOTOR* - sección 6
- *SERVO MOTOR* - sección 7
- *DISTANCE SENSOR* - sección 8
- *FLOOR SENSOR* - sección 9
- *ULTRASONIC SENSOR* - sección 10

- *BATTERY CONTROLLER* - sección 11
- *TRASH BIN* - sección 12

## 5. MAIN CONTROLLER

NOT YET DEFINED

## 6. DC MOTOR

### 6.1. DIRECTION

Seteo del sentido de giro del motor

#### Comando enviado

- COMANDO: 0x00
- DATO: 0x00 para sentido horario ó 0x01 para sentido anti-horario.

#### Respuesta al comando

Sin respuesta

### 6.2. SET SPEED

Seteo de la velocidad del motor en cuentas del encoder por segundo

#### Comando enviado

- COMANDO: 0x01
- DATO: consta de 6 bytes. SENTIDO = 0x00 para sentido horario ó 0x01 para sentido anti-horario. VELOCIDAD = 0x0000 a 0xFFFF. Valores positivos, en cuentas por segundos.

#### Respuesta al comando

Sin respuesta

### 6.3. SET ENCODER

Seteo de la cantidad de cuentas historicas del encoder

**Comando enviado**

- COMANDO: 0x02
- DATO: 0x0000 a 0xFFFF. Valores con signo, 0xFFFF es -32768 y 0x7FFF es 32767

**Respuesta al comando**

Sin respuesta

**6.4. GET ENCODER**

Obtener la cantidad de cuentas historicas del encoder

**Comando enviado**

- COMANDO: 0x03
- DATO: vacio

**Respuesta al comando**

- COMANDO: 0x03
- DATO: 0x0000 a 0xFFFF. Valores con signo, 0xFFFF es -32768 y 0x7FFF es 32767

**6.5. RESET ENCODER**

Resetear las cuentas historicas a cero

**Comando enviado**

- COMANDO: 0x04
- DATO: vacio

**Respuesta al comando**

Sin respuesta

**6.6. SET ENCODER TO STOP**

Seteo de cuantas cuentas a girar hasta detenerse

**Comando enviado**

0x0000 a 0xFFFF. Valores con signo, 0xFFFF es -32768 y 0x7FFF es 32767

- COMANDO: 0x05
- DATO: vacio

**Respuesta al comando**

Sin respuesta

**6.7. GET ENCODER TO STOP**

Obtener la cantidad de cuantas cuentas restantes hasta detenerse

**Comando enviado**

- COMANDO: 0x06
- DATO: vacio

**Respuesta al comando**

- COMANDO: 0x06
- DATO: 0x0000 a 0xFFFF. Valores con signo, 0xFFFF es -32768 y 0x7FFF es 32767

**6.8. DO NOT STOP**

Borrar la limitacion de cuentas para frenar

**Comando enviado**

- COMANDO: 0x06
- DATO: vacio

**Respuesta al comando**

Sin respuesta

7. SERVO MOTOR
8. DISTANCE SENSOR
9. FLOOR SENSOR
10. ULTRASONIC SENSOR
11. BATTERY CONTROLLER
12. TRASH BIN
13. Ejemplos
14. Conclusión