

# Trabajo práctico final

## Robot recolector de residuos

### Protocolo de comunicación

17 de enero de 2010

#### Resumen

En el presente se establece el protocolo de comunicación 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 de *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, bateria.*

## 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. En el cuadro 1 se muestra la estructura interna de un paquete típico.

TIPO_D	ID_D	TIPO_O	ID_O	COMANDO	LARGO	DATO	CRC
--------	------	--------	------	---------	-------	------	-----

Cuadro 1: Formato y header del paquete de datos

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\_D

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

## 2.2. ID\_D

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 está activo, el *ID* se ignora.

## 2.3. TIPO\_O

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 está reservado para hacer broadcast a todas las placas sin importar el grupo y no debe ser utilizado como origen del paquete.

## 2.4. ID\_O

El comando consta de 1 byte. Informa a que número de id debe ser dirigida la respuesta del paquete. 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.

## 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 0x7F.

- *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

Envía 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 cada uno se trata en las secciones como se detalla en el listado. Los comandos específicos para cada grupo deben ser desde 0x40 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 El identificador de grupo es 0x00.

## 6. DC MOTOR

Comandos específicos del controlador de velocidad de motor de continua. El identificador de grupo es 0x01.

### 6.1. SET DIRECTION

Seteo del sentido de giro del motor

#### Comando enviado

- COMANDO: 0x40
- DATO: 0x00 para sentido horario ó 0x01 para sentido anti-horario.

#### **Respuesta al comando**

Sin respuesta

### **6.2. SET DC SPEED**

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

#### **Comando enviado**

- COMANDO: 0x41
- 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 históricas del encoder

#### **Comando enviado**

- COMANDO: 0x42
- 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 históricas del encoder

#### **Comando enviado**

- COMANDO: 0x43
- DATO: vacío

#### **Respuesta al comando**

- COMANDO: 0x43
- DATO: 0x0000 a 0xFFFF. Valores con signo, 0xFFFF es -32768 y 0x7FFF es 32767

## 6.5. RESET ENCODER

Resetear las cuentas históricas a cero

### Comando enviado

- COMANDO: 0x44
- DATO: vacío

### Respuesta al comando

Sin respuesta

## 6.6. SET ENCODER TO STOP

Seteo de cuántas cuentas debe girar hasta detenerse

### Comando enviado

- COMANDO: 0x45
- DATO: 0x0000 a 0xFFFF. Valores con signo, 0xFFFF es -32768 y 0x7FFF es 32767

### Respuesta al comando

Sin respuesta

## 6.7. GET ENCODER TO STOP

Obtener la cantidad de las cuentas restantes que quedan por realizar hasta detenerse.

### Comando enviado

- COMANDO: 0x46
- DATO: vacío

### Respuesta al comando

- COMANDO: 0x46
- DATO: 0x0000 a 0xFFFF. Valores con signo, 0xFFFF es -32768 y 0x7FFF es 32767

## 6.8. DONT STOP

Deshace los comandos 6.6 y 6.7, deshabilita el conteo de cuentas para frenar y sigue en el estado actual.

**Comando enviado**

- COMANDO: 0x47
- DATO: vacío

**Respuesta al comando**

Sin respuesta

**6.9. MOTOR CONSUMPTION**

Consulta sobre el consumo actual del motor

**Comando enviado**

- COMANDO: 0x48
- DATO: vacío

**Respuesta al comando**

- COMANDO: 0x48
- DATO: Valor en punto flotante del consumo promedio del último segundo.  
-VER FORMATO DEL PUNTO FLOTANTE-

**6.10. MOTOR STRESS ALARM**

Indica al controlador principal que hay un consumo extremo en el motor, posiblemente un atasco del motor o de la rueda.

**Comando enviado**

- COMANDO: 0x49
- DATO: Valor en punto flotante del consumo. -VER FORMATO DEL PUNTO FLOTANTE-

**Respuesta al comando**

Sin respuesta

**7. SERVO MOTOR**

Comandos específicos del controlador de servomotores. El identificador de grupo es 0x02.

**7.1. SET POSITION**

Determina la posición en la que debe colocarse el servomotor indicado.

#### **Comando enviado**

- COMANDO: 0x40
- DATO: Valor de 0x00 a 0x05 que determina el id del servo al que se le aplicará la posición. Valor entre 0x0000 y 0x0384 que representa el rango de 0° a 180° con 0.2° de precisión. -VERIFICAR POSIBILIDADES EN LOS SERVOS-

#### **Respuesta al comando**

Sin respuesta

### **7.2. SET ALL POSITIONS**

Determina las posiciones en la que deben colocarse cada uno de los servomotores.

#### **Comando enviado**

- COMANDO: 0x41
- DATO: Consta de 5 valores entre 0x0000 y 0x0384 concatenados, uno para cada uno de los servos conectados al controlador. Cada valor representa el rango de 0° a 180° con 0.2° de precisión. -VERIFICAR POSIBILIDADES EN LOS SERVOS-

#### **Respuesta al comando**

Sin respuesta

### **7.3. GET POSITION**

Obtiene la última posición del servomotor indicado. -INICIALIZARLO EN UN VALOR DETERMINADO CUANDO LLEGA INIT-

#### **Comando enviado**

- COMANDO: 0x42
- DATO: Valor de 0x00 a 0x05 que determina el id del servo del que se requiere la posición.

#### **Respuesta al comando**

- COMANDO: 0x42
- DATO: Valor de 0x00 a 0x05 que determina el id del servo del que se requirió la posición. Valor entre 0x0000 y 0x0384 que representa el rango de 0° a 180° con 0.2° de precisión. -VERIFICAR POSIBILIDADES EN LOS SERVOS-



## 7.4. GET ALL POSITIONS

Obtiene las últimas posiciones de todos los servomotor conectados al controlador. -INICIALIZARLO EN UN VALOR DETERMINADO CUANDO LLEGA INIT-

### Comando enviado

- COMANDO: 0x43
- DATO: vacío

### Respuesta al comando

- COMANDO: 0x43
- DATO: Consta de 5 valores entre 0x0000 y 0x0384 concatenados, uno para cada uno de los servos conectados al controlador. Cada valor representa el rango de 0° a 180° con 0.2° de precisión. -VERIFICAR POSIBILIDADES EN LOS SERVOS-

## 7.5. SET SERVO SPEED

Determina la velocidad a la que el servomotor indicado llegará a la posición.

### Comando enviado

- COMANDO: 0x44
- DATO: Valor de 0x00 a 0x05 que determina el id del servo al que se le aplicará la velocidad. -VERIFICAR POSIBILIDADES EN LOS SERVOS-

### Respuesta al comando

Sin respuesta

## 7.6. SET ALL SPEEDS

Determina las velocidades a la que cada uno de los servomotores llegará a la posición indicada.

### Comando enviado

- COMANDO: 0x45
- DATO: Consta de 5 valores entre -VER POSIBLES VALORES- concatenados, uno para cada uno de los servos conectados al controlador. -VERIFICAR POSIBILIDADES EN LOS SERVOS-

### Respuesta al comando

Sin respuesta

### 7.7. GET SERVO SPEED

Obtiene la velocidad asignada al servomotor indicado. -INICIALIZARLO EN UN VALOR DETERMINADO CUANDO LLEGA INIT-

#### Comando enviado

- COMANDO: 0x46
- DATO: Valor de 0x00 a 0x05 que determina el id del servo del que se requiere la velocidad.

#### Respuesta al comando

- COMANDO: 0x46
- DATO: Valor de 0x00 a 0x05 que determina el id del servo del que se requirió la velocidad. -VERIFICAR POSIBILIDADES EN LOS SERVOS-

### 7.8. GET ALL SPEEDS

Obtiene las velocidades de cada uno de los servomotor conectados al controlador. -INICIALIZARLO EN UN VALOR DETERMINADO CUANDO LLEGA INIT-

#### Comando enviado

- COMANDO: 0x47
- DATO: vacío

#### Respuesta al comando

- COMANDO: 0x47
- DATO: Consta de 5 valores entre 0x0000 y 0x0384 concatenados, uno para cada uno de los servos conectados al controlador. -VERIFICAR POSIBILIDADES EN LOS SERVOS-

### 7.9. FREE SERVO

Deja de aplicar fuerza sobre el servo indicado -VERIFICAR QUE SEA CORRECTO-

#### Comando enviado

- COMANDO: 0x48
- DATO: Valor de 0x00 a 0x05 que determina el id del servo a liberar.

#### Respuesta al comando

Sin respuesta

## 7.10. FREE ALL SERVOS

Deja de aplicar fuerza sobre cada uno de los servomotor conectados al controlador. -VERIFICAR QUE SEA CORRECTO-

### Comando enviado

- COMANDO: 0x49
- DATO: vacío

### Respuesta al comando

Sin respuesta

## 8. DISTANCE SENSOR

Comandos específicos del controlador de sensores de distancia, generalmente telémetros. El identificador de grupo es 0x03.

### 8.1. ENABLE DISTANCE SENSOR

Enciende el sensor de distancia indicado.

#### Comando enviado

- COMANDO: 0x40
- DATO: Valor de 0x00 a 0x05 que representa el id del sensor a encender.

#### Respuesta al comando

Sin respuesta

### 8.2. DISABLE DISTANCE SENSOR

Apaga el sensor de distancia indicado.

#### Comando enviado

- COMANDO: 0x41
- DATO: Valor de 0x00 a 0x05 que representa el id del sensor a apagar.

#### Respuesta al comando

Sin respuesta

### 8.3. SET ALL DISTANCE SENSORS

Enciende o apaga cada uno de los sensores de distancia conectados al controlador.

#### Comando enviado

- COMANDO: 0x42
- DATO: Valor de 0x00 a 0x02F donde cada bit representa el id del sensor a encender o apagar. Si  $2^{ID} = 1$  entonces el sensor  $ID$  está encendido. Si  $2^{ID} = 0$  entonces el sensor  $ID$  está apagado.

#### Respuesta al comando

Sin respuesta

### 8.4. GET DISTANCE

Obtiene la distancia del sensor indicado.

#### Comando enviado

- COMANDO: 0x43
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

#### Respuesta al comando

- COMANDO: 0x43
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0xFFFF que representa la distancia en milímetros al objeto.

### 8.5. GET ALL DISTANCES

Obtiene las distancias de cada uno de los sensores conectados al controlador.

#### Comando enviado

- COMANDO: 0x44
- DATO: vacío

#### Respuesta al comando

- COMANDO: 0x44
- DATO: Consta de 5 valores entre 0x0000 y 0xFFFF concatenados, uno para cada uno de los sensores conectados al controlador.

### 8.6. GET VALUE

Obtiene el valor en voltios del sensor indicado.

#### **Comando enviado**

- COMANDO: 0x45
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

#### **Respuesta al comando**

- COMANDO: 0x45
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0x03FF con el valor en voltios de la lectura que representa la distancia al objeto.

### **8.7. GET ALL VALUES**

Obtiene las distancias de cada uno de los sensores conectados al controlador.

#### **Comando enviado**

- COMANDO: 0x46
- DATO: vacío

#### **Respuesta al comando**

- COMANDO: 0x46
- DATO: Consta de 5 valores entre 0x0000 y 0x03FF concatenados, uno para cada uno de los sensores conectados al controlador. Valor en voltios de la lectura que representa la distancia al objeto.

### **8.8. GET ONE DISTANCE**

Obtiene la distancia del sensor indicado. Igual al comando 8.4 pero si es necesario enciende el sensor, toma la lectura y luego lo apaga para un mayor ahorro de energía.

#### **Comando enviado**

- COMANDO: 0x47
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

#### **Respuesta al comando**

- COMANDO: 0x47
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0xFFFF que representa la distancia en milímetros al objeto.

## 8.9. GET ONE DISTANCE FOR ALL

Obtiene las distancias de cada uno de los sensores conectados al controlador. Igual al comando 8.5 pero si es necesario enciende los sensores, toma las lecturas y luego los apaga para un mayor ahorro de energía.

### Comando enviado

- COMANDO: 0x48
- DATO: vacío

### Respuesta al comando

- COMANDO: 0x48
- DATO: Consta de 5 valores entre 0x0000 y 0xFFFF concatenados, uno para cada uno de los sensores conectados al controlador.

## 8.10. GET ONE VALUE

Obtiene el valor en voltios del sensor indicado. Igual al comando 8.6 pero si es necesario enciende el sensor, toma la lectura y luego lo apaga para un mayor ahorro de energía.

### Comando enviado

- COMANDO: 0x49
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

### Respuesta al comando

- COMANDO: 0x49
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0x03FF con el valor en voltios de la lectura que representa la distancia al objeto.

## 8.11. GET ONE VALUES FOR ALL

Obtiene las distancias de cada uno de los sensores conectados al controlador. Igual al comando 8.7 pero si es necesario enciende los sensores, toma las lecturas y luego los apaga para un mayor ahorro de energía.

### Comando enviado

- COMANDO: 0x4A
- DATO: vacío

#### Respuesta al comando

- COMANDO: 0x4A
- DATO: Consta de 5 valores entre 0x0000 y 0x03FF concatenados, uno para cada uno de los sensores conectados al controlador. Valor en voltios de la lectura que representa la distancia al objeto.

## 9. FLOOR SENSOR

Comandos específicos del controlador de sensores de piso. El identificador de grupo es 0x04.

### 9.1. ENABLE FLOOR SENSOR

Enciende el sensor de piso indicado.

#### Comando enviado

- COMANDO: 0x40
- DATO: Valor de 0x00 a 0x05 que representa el id del sensor a encender.

#### Respuesta al comando

Sin respuesta

### 9.2. DISABLE FLOOR SENSOR

Apaga el sensor de piso indicado.

#### Comando enviado

- COMANDO: 0x41
- DATO: Valor de 0x00 a 0x05 que representa el id del sensor a apagar.

#### Respuesta al comando

Sin respuesta

### 9.3. SET ALL FLOOR SENSORS

Enciende o apaga cada uno de los sensores de piso conectados al controlador.

#### Comando enviado

- COMANDO: 0x42
- DATO: Valor de 0x00 a 0x020 donde cada bit representa el id del sensor a encender o apagar. Si  $2^{ID} = 1$  entonces el sensor  $ID$  está encendido. Si  $2^{ID} = 0$  entonces el sensor  $ID$  está apagado.

#### **Respuesta al comando**

Sin respuesta

### **9.4. GET VALUE**

Obtiene el valor en voltios del sensor indicado.

#### **Comando enviado**

- COMANDO: 0x43
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

#### **Respuesta al comando**

- COMANDO: 0x43
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0x03FF con el valor en voltios de la lectura que representa la distancia al objeto.

### **9.5. GET ALL VALUES**

Obtiene las distancias de cada uno de los sensores conectados al controlador.

#### **Comando enviado**

- COMANDO: 0x44
- DATO: vacío

#### **Respuesta al comando**

- COMANDO: 0x44
- DATO: Consta de 5 valores entre 0x0000 y 0x03FF concatenados, uno para cada uno de los sensores conectados al controlador. Valor en voltios de la lectura que representa la distancia al objeto.

### **9.6. GET ONE VALUE**

Obtiene el valor en voltios del sensor indicado. Igual al comando 9.4 pero si es necesario enciende el sensor, toma la lectura y luego lo apaga para un mayor ahorro de energía.

#### **Comando enviado**

- COMANDO: 0x45
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.



#### **Respuesta al comando**

- COMANDO: 0x45
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0x03FF con el valor en voltios de la lectura que representa la distancia al objeto.

### **9.7. GET ONE VALUE FOR ALL**

Obtiene las distancias de cada uno de los sensores conectados al controlador. Igual al comando 9.5 pero si es necesario enciende los sensores, toma las lecturas y luego los apaga para un mayor ahorro de energía.

#### **Comando enviado**

- COMANDO: 0x46
- DATO: vacío

#### **Respuesta al comando**

- COMANDO: 0x46
- DATO: Consta de 5 valores entre 0x0000 y 0x03FF concatenados, uno para cada uno de los sensores conectados al controlador. Valor en voltios de la lectura que representa la distancia al objeto.

## **10. ULTRASONIC SENSOR**

Comandos específicos del controlador de sensores de ultrasonido. El identificador de grupo es 0x05.

### **10.1. ENABLE ULTRASONIC SENSOR**

Enciende el sensor de ultrasonido indicado.

#### **Comando enviado**

- COMANDO: 0x40
- DATO: Valor de 0x00 a 0x05 que representa el id del sensor a encender.

#### **Respuesta al comando**

Sin respuesta

### **10.2. DISABLE ULTRASONIC SENSOR**

Apaga el sensor de ultrasonido indicado.

**Comando enviado**

- COMANDO: 0x41
- DATO: Valor de 0x00 a 0x05 que representa el id del sensor a apagar.

**Respuesta al comando**

Sin respuesta

**10.3. SET ALL ULTRASONIC SENSOR**

Enciende o apaga cada uno de los sensores de ultrasonido conectados al controlador.

**Comando enviado**

- COMANDO: 0x42
- DATO: Valor de 0x00 a 0x02F donde cada bit representa el id del sensor a encender o apagar. Si  $2^{ID} = 1$  entonces el sensor  $ID$  está encendido. Si  $2^{ID} = 0$  entonces el sensor  $ID$  está apagado.

**Respuesta al comando**

Sin respuesta

**10.4. GET DISTANCE**

Obtiene la distancia del sensor indicado.

**Comando enviado**

- COMANDO: 0x43
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

**Respuesta al comando**

- COMANDO: 0x43
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0xFFFF que representa la distancia en milímetros al objeto.

**10.5. GET ALL DISTANCES**

Obtiene las distancias de cada uno de los sensores conectados al controlador.

**Comando enviado**

- COMANDO: 0x44
- DATO: vacío

#### **Respuesta al comando**

- COMANDO: 0x44
- DATO: Consta de 5 valores entre 0x0000 y 0xFFFF concatenados, uno para cada uno de los sensores conectados al controlador.

### **10.6. GET VALUE**

Obtiene el valor en voltios del sensor indicado.

#### **Comando enviado**

- COMANDO: 0x45
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

#### **Respuesta al comando**

- COMANDO: 0x45
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0x03FF con el valor en voltios de la lectura que representa la distancia al objeto.

### **10.7. GET ALL VALUES**

Obtiene las distancias de cada uno de los sensores conectados al controlador.

#### **Comando enviado**

- COMANDO: 0x46
- DATO: vacío

#### **Respuesta al comando**

- COMANDO: 0x46
- DATO: Consta de 5 valores entre 0x0000 y 0x03FF concatenados, uno para cada uno de los sensores conectados al controlador. Valor en voltios de la lectura que representa la distancia al objeto.

### **10.8. GET ONE DISTANCE**

Obtiene la distancia del sensor indicado. Igual al comando 10.4 pero si es necesario enciende el sensor, toma la lectura y luego lo apaga para un mayor ahorro de energía.

#### **Comando enviado**

- COMANDO: 0x47
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

#### **Respuesta al comando**

- COMANDO: 0x47
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0xFFFF que representa la distancia en milímetros al objeto.

### **10.9. GET ONE DISTANCE FOR ALL**

Obtiene las distancias de cada uno de los sensores conectados al controlador. Igual al comando 10.5 pero si es necesario enciende los sensores, toma las lecturas y luego los apaga para un mayor ahorro de energía.

#### **Comando enviado**

- COMANDO: 0x48
- DATO: vacío

#### **Respuesta al comando**

- COMANDO: 0x48
- DATO: Consta de 5 valores entre 0x0000 y 0xFFFF concatenados, uno para cada uno de los sensores conectados al controlador.

### **10.10. GET ONE VALUE**

Obtiene el valor en voltios del sensor indicado. Igual al comando 10.6 pero si es necesario enciende el sensor, toma la lectura y luego lo apaga para un mayor ahorro de energía.

#### **Comando enviado**

- COMANDO: 0x49
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que se quiere la lectura.

#### **Respuesta al comando**

- COMANDO: 0x49
- DATO: Valor de 0x00 a 0x05 que determina el id del sensor del que proviene el la lectura de distancia. Valor de 0x0000 a 0x03FF con el valor en voltios de la lectura que representa la distancia al objeto.

### 10.11. GET ONE VALUE FOR ALL

Obtiene las distancias de cada uno de los sensores conectados al controlador. Igual al comando 9.5 pero si es necesario enciende los sensores, toma las lecturas y luego los apaga para un mayor ahorro de energía.

#### Comando enviado

- COMANDO: 0x4A
- DATO: vacío

#### Respuesta al comando

- COMANDO: 0x4A
- DATO: Consta de 5 valores entre 0x0000 y 0x03FF concatenados, uno para cada uno de los sensores conectados al controlador. Valor en voltios de la lectura que representa la distancia al objeto.

## 11. BATTERY CONTROLLER

Comandos específicos del controlador de carga y consumo de la batería. El identificador de grupo es 0x06.

### 11.1. GET BATTERY VALUE

Obtiene el valor en voltios de la batería.

#### Comando enviado

- COMANDO: 0x40
- DATO: vacío

#### Respuesta al comando

- COMANDO: 0x40
- DATO: Valor de 0x0000 a 0x03FF que representa la carga en voltios de la batería.

### 11.2. BATTERY FULL ALARM

Mensaje enviado desde el controlador de la batería informando que se completado la carga de la misma.

#### Comando enviado

- COMANDO: 0x41
- DATO: vacío

**Respuesta al comando**

Sin respuesta

**11.3. SET BATTERY EMPTY VALUE**

Establece el valor de la batería para ser tomado como crítico.

**Comando enviado**

- COMANDO: 0x42
- DATO: Valor de 0x0000 a 0x03FF que representa la carga en voltios de la batería.

**Respuesta al comando**

Sin respuesta

**11.4. BATTERY EMPTY ALARM**

Mensaje enviado desde el controlador de la batería informando que de el voltaje llegó a un valor crítico.

**Comando enviado**

- COMANDO: 0x43
- DATO: Valor de 0x0000 a 0x03FF que representa la carga en voltios de la batería.

**Respuesta al comando**

Sin respuesta

**12. TRASH BIN**

Comandos específicos del controlador de carga en el cesto de basura. El identificador de grupo es 0x07.

**12.1. GET TRASH BIN VALUE**

Obtiene el valor que representa que tan lleno está el cesto interno de basura.

**Comando enviado**

- COMANDO: 0x40
- DATO: vacío

**Respuesta al comando**

- COMANDO: 0x40
- DATO: Valor de 0x0000 a 0x03FF que representa que tan lleno está el cesto interno de basura.

**12.2. BIN FULL ALARM**

Mensaje enviado desde el controlador del cesto de basura informando que se completado y debe ser descargado.

**Comando enviado**

- COMANDO: 0x41
- DATO: vacío

**Respuesta al comando**

Sin respuesta

**13. Ejemplos**

NOT YET DEFINED