Trabajo práctico final
Robot recolector de residuos
Protocolo de comunicación

21 de mayo de 2010

Resumen

En el presente se establece el protocolo de comunicación para el envío y recepción de comandos desde y 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

La comunicación entre los distintos componentes del robot es imprescindible para su correcto funcionamiento y por lo tanto, un protocolo acorde es la base fundamental sobre la cual se construye dicha comunicación. Toda interacción con el entorno es transportada por los mensajes enviados desde el controlador principal hacia los periféricos y se debe asegurar que su correctitud.

Para dicha tarea el protocolo está diseñado para ser transmitido por *RS-232* utilizando la configuración de *Daisy-Chain* entre las distintas placas controladoras creando un anillo que conecta a cada nodo con su vecino.

Las placas controladoras se dividen en grupos según su función y periféricos que tenga conectados.

En la sección 2 el formato que deben respetar todos los paquetes enviados bajo el protocolo. En la sección 3 se definen los comandos comunes y en la sección 4, los comandos específicos para cada familia de placas controladoras. En la sección 11 se define el estandar bajo el cual se env'ian números de 16 y 32 bits por el protocolo. Finalmente, en la sección 12 se muestran ejemplos de paquetes para mejor entendimiento.

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.

LARGO	DESTINO	ORIGEN	COMANDO	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. LARGO

Indica el largo en bytes del paquete que viene detras, consta de 1 byte. Necesario ante la existencia del campo *DATO* de longitud variable. En caso que la longitud del campo *DATO* sea cero, es 0x04.

2.2. DESTINO

Identifica al destinatario del paquete, consta de 1 byte. Los 4 bits más significativos indican el grupo y los 4 bits menos significativos, el número de ID de la placa de destino. Los grupos predefinidos se tratan en el apartado 4. Si el ID de placa es F, entonces el paquete es broadcast a todos los IDs del grupo indicado. El valor 0xFF indica que el paquete es broadcast a todas las placas de todos los grupos.

2.3. ORIGEN

Similar al *DESTINO*, consta de 1 byte. Determina el emisor del paquete para la respuesta. Los 4 bits más significativos indican el grupo y los 4 bits menos significativos, el número de ID de la placa de origen. Los grupos predefinidos se tratan en el apartado 4. Los valores permitidos son del 0 al E, ya que F indica broadcast y no es válida una respuesta broadcast.

2.4. 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.

2.5. DATO

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 campo *LARGO* será 0x04.

En el apartado 11 se explican las reglas para la codificacíon de números utilizadas en el protocolo.

2.6. CRC

El comando consta de 1 byte. Cálculo de CRC sobre el paquete enviado. Se realiza un XOR con cada uno de los bytes que conforman el paquete.

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. El bit más significativo indica que el paquete es una respuesta, los bits menos significativos, indicaran a qué comando se está respondiendo.

Todos los comandos generan una respuesta o al menos un ACK con el campo *DATOS* vacío.

- *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.

También utilizado para obtener la confirmación del listado de dispositivos conectados.

Comando enviado

- COMANDO: 0x01
- DATO: vacío

Respuesta al comando

- COMANDO: 0x81
- 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

- COMANDO: 0x82
- DATO: Descripción de la placa en texto plano

3.3. PING

Envia un ping a la placa

Comando enviado

- COMANDO: 0x03
- DATO: vacío

Respuesta al comando

- COMANDO: 0x83
- DATO: vacío

3.4. ERROR

Informa que ha habido un error.

Comando enviado

- COMANDO: 0x04
- DATO: 1 byte con el código de error.

  El valor 0x00 indica un error de CRC en el paquete. En este caso, también se agrega el paquete con el CRC erróneo y luego el CRC esperado.

  El valor 0x01 indica que el comando es desconocido para la placa destinataria.

  Cualquier otro valor, indica un error que depende de la placa y el comando enviado.

Respuesta al comando

Único comando sin respuesta directa. En caso de error de CRC debería retransmitir el paquete o tomar alguna otra acción ante otro tipo de error.

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
- *BATTERY CONTROLLER* - sección 9
- *TRASH BIN* - sección 10

5. MAIN CONTROLLER

Comandos especificos para el controlador principal. El identificador de grupo es 0x0X.

-SIN COMANDOS-

6. DC MOTOR

Comandos específicos del controlador de velocidad de motor de corriente contínua. El identificador de grupo es 0x1X.

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

- COMANDO: 0xC0
- DATO: vacío

6.2. SET DC SPEED

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

Comando enviado

- COMANDO: 0x41
- DATO: 0x00 para sentido horario ó 0x01 para sentido anti-horario. Número entero de 16 bits con signo, que representa la velocidad en cuentas por segundos. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC1
- DATO: vacío

6.3. SET ENCODER

Seteo de la cantidad de cuentas históricas del encoder

Comando enviado

- COMANDO: 0x42
- DATO: Número entero de 32 bits con signo, con el valor para setear en el histórico del encoder. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC2
- DATO: vacío

6.4. GET ENCODER

Obtener la cantidad de cuentas históricas del encoder

Comando enviado

- COMANDO: 0x43
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC3
- DATO: Número entero de 32 bits con signo, que representa el valor histórico del encoder. Ver apartado 11 para mayor información.

6.5. RESET ENCODER

Resetear las cuentas históricas a cero

Comando enviado

- COMANDO: 0x44
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC4
- DATO: vacío

6.6. SET ENCODER TO STOP

Seteo de cuántas cuentas debe girar hasta detenerse

Comando enviado

- COMANDO: 0x45
- DATO: Número entero de 16 bits con signo, que representa la cantidad de cuentas del encoder restantes para que el motor se detenga. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC5
- DATO: vacío

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: 0xC6
- DATO: Número entero de 16 bits con signo, que representa la cantidad de cuentas del encoder restantes para detener el motor. Ver apartado 11 para mayor información.

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

- COMANDO: 0xC7
- DATO: vacío

6.9. MOTOR CONSUMPTION

Consulta sobre el consumo actual del motor

Comando enviado

- COMANDO: 0x48
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC8
- DATO: Número entero positivo de 16 bits en el rango desde 0x0000 hasta 0x03FF, que representa el consumo promedio del último segundo. Ver apartado 11 para mayor información.

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: Número entero positivo de 16 bits en el rango desde 0x0000 hasta 0x03FF, que representa el consumo ante el que sono la alarma. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC9
- DATO: vacío

6.11. MOTOR SHUT DOWN ALARM

Indica al controlador principal que el motor ha sido apagado debido al alto consumo. Enviado luego de sucesivos avisos del comando 6.10.

Comando enviado

- COMANDO: 0x4A
- DATO: Número entero positivo de 16 bits en el rango desde 0x0000 hasta 0x03FF, que representa el consumo ante el que sono la alarma. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xCA
- DATO: vacío

6.12. GET DC SPEED

Obtiene la velocidad del motor en cuentas del encoder por segundo

Comando enviado

- COMANDO: 0x4B
- DATO: vacío

Respuesta al comando

- COMANDO: 0xCB
- DATO: 0x00 para sentido horario ó 0x01 para sentido anti-horario. Número entero de 16 bits con signo, que representa la velocidad en cuentas por segundos. Ver apartado 11 para mayor información.

7. SERVO MOTOR

Comandos específicos del controlador de servomotores. El identificador de grupo es 0x2X.

7.1. SET POSITION

Determina la posición en la que debe colocarse el servo motor indicado.

Comando enviado

- COMANDO: 0x40
- DATO: Valor de 0x00 a 0x04 que determina el ID del servo al que se le aplicará la posición. Valor entre 0x00 y 0xB4 que representa el rango de 0° a 180° con 1° de presición.

Respuesta al comando

- COMANDO: 0xC0
- DATO: vacío

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 0x00 y 0xB4 concatenados, uno para cada uno de los servos conectados al controlador. Cada valor representa el rango de 0° a 180° con 1° de presición.

Respuesta al comando

- COMANDO: 0xC1
- DATO: vacío

7.3. GET POSITION

Obtiene la última posición del servomotor indicado.

Comando enviado

- COMANDO: 0x42
- DATO: Valor de 0x00 a 0x04 que determina el ID del servo del que se requiere la posición.

Respuesta al comando

- COMANDO: 0xC2
- DATO: Valor de 0x00 a 0x04 que determina el ID del servo del que se requirió la posición. Valor entre 0x00 y 0xB4 que representa el rango de 0° a 180° con 1° de presición.

7.4. GET ALL POSITIONS

Obtiene las últimas posiciones de todos los servomotor conectados al controlador.

Comando enviado

- COMANDO: 0x43
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC3
- DATO: Consta de 5 valores entre 0x00 y 0xB4 concatenados, uno para cada uno de los servos conectados al controlador. Cada valor representa el rango de 0° a 180° con 1° de presición.

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 0x04 que determina el ID del servo al que se le aplicará la velocidad. Valor entre 0x00 y 0xB4, velocidad en grados por segundo.

Respuesta al comando

- COMANDO: 0xC4
- DATO: vacío

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 0x00 y 0xB4 concatenados, uno para cada uno de los servos conectados al controlador. Cada valor representa a la velocidad en grados por segundo.

Respuesta al comando

- COMANDO: 0xC5
- DATO: vacío

7.7. GET SERVO SPEED

Obtiene la velocidad asignada al servomotor indicado.

Comando enviado

- COMANDO: 0x46
- DATO: Valor de 0x00 a 0x04 que determina el ID del servo del que se requiere la velocidad.

Respuesta al comando

- COMANDO: 0xC6
- DATO: Valor de 0x00 a 0x04 que determina el ID del servo del que se requirió la velocidad. Valor entre 0x00 y 0xB4, velocidad en grados por segundo.

7.8. GET ALL SPEEDS

Obtiene las velocidades de cada uno de los servomotor conectados al controlador.

Comando enviado

- COMANDO: 0x47
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC7
- DATO: Consta de 5 valores entre 0x00 y 0xB4 concatenados, uno para cada uno de los servos conectados al controlador. Cada valor representa a la velocidad en grados por segundo.

7.9. FREE SERVO

Deja de aplicar fuerza sobre el servo indicado.

Comando enviado

- COMANDO: 0x48
- DATO: Valor de 0x00 a 0x04 que determina el ID del servo a liberar.

Respuesta al comando

- COMANDO: 0xC8
- DATO: vacío

7.10. FREE ALL SERVOS

Deja de aplicar fuerza sobre cada uno de los servomotor conectados al controlador.

Comando enviado

- COMANDO: 0x49
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC9
- DATO: vacío

7.11. GET STATUS

Obtiene el estado de cada uno de los switches conectados al controlador.

Comando enviado

- COMANDO: 0x4A
- DATO: vacío

Respuesta al comando

- COMANDO: 0xCA
- DATO: Valor de 0x00 a 0x7F donde cada bit representa el estado del switch con ese ID. Si $2^{ID} = 1$ entonces el switch con ID está en un estado lógico alto. Si $2^{ID} = 0$ entonces el switch con ID está en un estado lógico bajo.

7.12. ALARM ON STATE

Establece si se desea o no recibir una alarma ante cierto cambio de estado en el switch indicado. Puede ser ante cualquier cambio o sobre un flanco ascendente o descendente.

Comando enviado

- COMANDO: 0x4B
- DATO: Valor de 0x00 a 0x05 con el ID del switch que se está configurando. Valor entre 0x00 y 0x03 con el tipo de cambio ante el cual generar la alarma. Con un 0x00 ignora cualquier cambio en el switch. Se utiliza 0x01 para que sea ante un flanco ascendente y 0x02 para que sea sólo ante un flanco descendente y 0x03 para que cualquier cambio en el switch genere el mensaje (opcional).

Respuesta al comando

- COMANDO: 0xCB
- DATO: vacío

7.13. SWITCH ALARM

Comando enviado desde la placa a la controladora principal informando que fue satisfecha la condición de la alarma.

Comando enviado

- COMANDO: 0x4C
- DATO: Valor de 0x00 a 0x05 con el ID del switch que provocó el comando. Valor entre 0x00 y 0x03 con el tipo de cambio configurado y el estado actual, 0x00 para un estado bajo y 0x01 para un estado alto.

Respuesta al comando

- COMANDO: 0xCC
- DATO: vacío

8. DISTANCE SENSOR

Comandos específicos del controlador de sensores de distancia, telémetros o sensores de piso. También puede estar presente un sensor de ultrasonido o un switch.

El identificador de grupo es 0x3X.

8.1. ON 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. El ID 0x05 hace referencia al led de la placa si esta presente, en caso contrario se ignora.

Respuesta al comando

- COMANDO: 0xC0
- DATO: vacío

8.2. OFF 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. El ID 0x05 hace referencia al sensor de ultrasonido o switch de la placa.

Respuesta al comando

- COMANDO: 0xC1
- DATO: vacío

8.3. ENABLE DISTANCE SENSORS

Habilita o deshabilita cada uno de los sensores de distancia conectados al controlador. Permite identificar los sensores a los que se deberá tener en cuenta para futuras lecturas.

Comando enviado

- COMANDO: 0x42
- DATO: Valor de 0x00 a 0x3F donde cada bit representa el ID del sensor a habilitar o deshabilitar. Si $2^{ID} = 1$ entonces el sensor ID está habilitado. Si $2^{ID} = 0$ entonces el sensor ID está deshabilitado.

Respuesta al comando

- COMANDO: 0xC2
- DATO: vacío

8.4. GET DISTANCE SENSORS STATUS

Obtiene el estado de habilitación de cada uno de los sensores de distancia conectados al controlador.

Comando enviado

- COMANDO: 0x43
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC3
- DATO: Valor de 0x00 a 0x3F donde cada bit representa el ID del sensor a habilitar o deshabilitar. Si $2^{ID} = 1$ entonces el sensor ID está habilitado. Si $2^{ID} = 0$ entonces el sensor ID está deshabilitado.

8.5. GET VALUE

Obtiene el valor promedio de la entrada de los sensores indicados.

Comando enviado

- COMANDO: 0x44
- DATO: Valor de 0x00 a 0x3F donde cada bit representa el ID del sensor del cual obtener la lectura.

Respuesta al comando

- COMANDO: 0xC4
- DATO: Valor de 0x00 a 0x3F donde cada bit representa el ID del sensor del cual proviene el la lectura de distancia. Secuencia de números enteros positivos de 16 bits en el rango desde 0x0000 hasta 0x03FF, con el valor de la lectura que representa la distancia al objeto. En la secuencia de números el orden está dado de izquierda a derecha comenzando por el bit menos significativo.

  En el caso del sensor de ultrasonido el rango es desde 0x0000 hasta 0x7594 que representa la mínima y máxima lectura del sensor.

  En el caso del switch, un estado lógico bajo se lee como 0x0000 y un estado lógico alto se lee como 0xFFFF.

  Ver apartado 11 para mayor información sobre la codificación de números.

8.6. GET ONE VALUE

Obtiene el valor de la entrada del sensor indicado. Igual al comando 8.5 pero sin realizar un promedio de lecturas.

Comando enviado

- COMANDO: 0x45
- DATO: Valor de 0x00 a 0x3F donde cada bit representa el *ID* del sensor del cual obtener la lectura.

Respuesta al comando

- COMANDO: 0xC5
- DATO: Valor de 0x00 a 0x3F donde cada bit representa el *ID* del sensor del cual proviene el la lectura de distancia. Secuencia de números enteros positivos de 16 bits en el rango desde 0x0000 hasta 0x03FF, con el valor de la lectura que representa la distancia al objeto. En la secuencia de números el orden está dado de izquierda a derecha comenzando por el bit menos significativo.

  En el caso del sensor de ultrasonido el rango es desde 0x0000 hasta 0x7594 que representa la mínima y máxima lectura del sensor.

  En el caso del switch, un estado lógico bajo se lee como 0x0000 y un estado lógico alto se lee como 0xFFFF.

  Ver apartado 11 para mayor información sobre la codificación de números.

8.7. ALARM ON STATE

Cuando un switch está presente en el ID: 0x05, establece si se desea o no recibir una alarma ante cierto cambio de estado en el mismo. Puede ser ante cualquier cambio o sobre un flanco ascendente o descendente.

Comando enviado

- COMANDO: 0x46
- DATO: Valor entre 0x00 y 0x03 con el tipo de cambio ante el cual generar la alarma. Con un 0x00 ignora cualquier cambio en el switch. Se utiliza 0x01 para que sea ante un flanco ascendente y 0x02 para que sea sólo ante un flanco descendente y 0x03 para que cualquier cambio en el switch genere el mensaje (opcional).

Respuesta al comando

- COMANDO: 0xC6
- DATO: vacío

8.8. SWITCH ALARM

Comando enviado desde la placa a la controladora principal informando que fue satisfecha la condición de la alarma.

Comando enviado

- COMANDO: 0x47
- DATO: Valor entre 0x00 y 0x03 con el tipo de cambio configurado y el estado actual, 0x00 para un estado bajo y 0x01 para un estado alto.

Respuesta al comando

- COMANDO: 0xC7
- DATO: vacío

9. BATTERY CONTROLLER

Comandos específicos del controlador de carga y consumo de la batería. El identificador de grupo es 0x4X.

9.1. ENABLE

Habilita la alimentación del robot mediante la batería.

Comando enviado

- COMANDO: 0x40
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC0
- DATO: vacío.

9.2. DISABLE

Deshabilita la alimentación del robot mediante la batería.

Comando enviado

- COMANDO: 0x41
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC1
- DATO: vación.

9.3. GET BATTERY VALUE

Obtiene el valor de la entrada de la batería.

Comando enviado

- COMANDO: 0x42
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC2
- DATO: Número entero positivo de 16 bits, en el rango desde 0x0000 hasta 0x03FF, que representa la lectura de los voltios de la batería. Ver apartado 11 para mayor información.

9.4. BATTERY FULL ALARM

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

Comando enviado

- COMANDO: 0x43
- DATO: vacío

Respuesta al comando

- COMANDO: 0xC3
- DATO: vacío

9.5. SET BATTERY EMPTY VALUE

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

Comando enviado

- COMANDO: 0x44
- DATO: Número entero positivo de 16 bits, en el rango desde 0x0000 hasta 0x03FF, que representa la lectura de los voltios de la batería. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC4
- DATO: vacío

9.6. BATTERY EMPTY ALARM

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

Comando enviado

- COMANDO: 0x45
- DATO: Número entero positivo de 16 bits, en el rango desde 0x0000 hasta 0x03FF, que representa la lectura de los voltios de la batería. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC5
- DATO: vacío

9.7. SET FULL BATTERY VALUE

Establece el valor de la batería para ser tomado como carga completa.

Comando enviado

- COMANDO: 0x46
- DATO: Número entero positivo de 16 bits, en el rango desde 0x0000 hasta 0x03FF, que representa la lectura de los voltios a ser tomado como carga completa de la batería. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC6
- DATO: vacío

10. TRASH BIN

Comandos específicos del controlador de carga en el cesto de basura. El identificador de grupo es 0x5X.

10.1. 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: 0xC0
- DATO: Número entero positivo de 16 bits, en el rango desde 0x0000 hasta 0x03FF, que representa que tan lleno está el cesto interno de basura. Ver apartado 11 para mayor información.

10.2. 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

- COMANDO: 0xC1
- DATO: vacío

10.3. SET FULL BIN VALUE

Establece el valor para el cual se envía la alarma 10.2 indicando un nivel alto en el cesto de basura.

Comando enviado

- COMANDO: 0x42
- DATO: Número entero positivo de 16 bits, en el rango desde 0x0000 hasta 0x03FF, que representa la lectura del nivel del cesto de basura. Ver apartado 11 para mayor información.

Respuesta al comando

- COMANDO: 0xC2
- DATO: vacío

11. Codificación de los valores enviados

Para mantener la compatibilidad entre los distintos agentes dentro de la cadena de tranmisión de datos, los números enteros a través del protocolo son codificados bajo el formato *little-endian* con elementos atómicos de 8 bits.

Por ejemplo, si el valor 1023, codificaco en un entero de 16 bits en hexadecimal es 0x03FF, entonces se envía primero el byte 0xFF y luego el 0x03. Si fuera el caso del valor 1193046, que codificado en un entero de 32 bits, cuyo

hexadecimal es 0x00123456, se envía primero el byte 0x56, luego el 0x34, 0x12 y por último, 0x00.

El envío de números con punto flotante no está previsto en el protocolo.

12. Ejemplos

Se detallan paquetes de ejemplo para una mejor comprensión del protocolo.

12.1. Ejemplo 1

El cuadro 2 muestra un paquete enviado desde el controlador principal con *ID* igual a 0, a una placa controladora de motor de corriente contínua con *ID* igual a 1, seteando el sentido de giro anti-horario. El campo *CRC* es 0x55, valor que corresponde con el resultado de realizar la operación *XOR* entre todos los bytes del paquete.

05	11	00	40	01	55

Cuadro 2: Paquete de datos del ejemplo 1

12.2. Ejemplo 2

El cuadro 3 muestra un paquete enviado desde el controlador de baterías con *ID* igual a 2, al controlador principal con *ID* igual a 0, informando que la batería se encuentra con un valor crítico de 0x036B. El campo *CRC* es 0x49, valor que corresponde con el resultado de realizar la operación *XOR* entre todos los bytes del paquete.

06	00	62	45	6B	03	49

Cuadro 3: Paquete de datos del ejemplo 2