Remote IO Control Protocol

Document Version: R3-2020

RIOC messages can be transferred between RIOC units over CAN-Bus, UART (serial), UDP, TCP or WebSocket.

Remote IO Control over CAN-Bus

For each data frame transferred between RIOC units, both the address and message are packaged in an extended CAN datagram. The source address and destination address occupy 16 bits in CAN EID. The rest 2 bits in CAN EID and 11 bits in CAN SID are reserved in RIOC protocol presently. And, the RIOC message is filled in an 8-byte payload of the CAN datagram. In other words, the length of any RIOC message must be 8 bytes (or less than 8 bytes).

<EID15~EID8>	source unit address
<EID7~EID0>	destination unit address (master unit: 0, slave unit: 1~254, broadcast: 255)
<DATA0>	RIOC class
<DATA1>	command (byte7 marks transmission direction)
<DATA2>	channel
<DATA3>	parameter
<DATA4>	parameter
<DATA5>	parameter
<DATA6>	parameter
<DATA7>	parameter

Remote IO Control over UART

14 bytes in each serial data frame

0x5A	leading character
0x00	data frame version
<ADD_S >	source unit address
<ADD_D >	destination unit address
<DATA0>	RIOC class
<DATA1>	command
<DATA2>	channel
<DATA3>	parameter
<DATA4>	parameter
<DATA5>	parameter
<DATA6>	parameter
<DATA7>	parameter
<CRC>	checksum
0xA5	ending character

Remote IO Control over UDP

11 bytes in each UDP datagram

0x00	data frame version
<ADD_S >	source unit address
<ADD_D >	destination unit address
<DATA0>	RIOC class
<DATA1>	command
<DATA2>	channel
<DATA3>	parameter
<DATA4>	parameter
<DATA5>	parameter
<DATA6>	parameter
<DATA7>	parameter

Remote IO Control over TCP

13 bytes in each TCP data frame

0x5A	leading character
0x00	data frame version
<ADD_S >	source unit address
<ADD_D >	destination unit address
<DATA0>	RIOC class
<DATA1>	command
<DATA2>	channel
<DATA3>	parameter
<DATA4>	parameter
<DATA5>	parameter
<DATA6>	parameter
<DATA7>	parameter
0xA5	ending character

Remote IO Control over WebSocket (Text Mode)

22 ASCII characters in each web socket text message

(each field below is a double-digit hex number with two characters)

“00”	data frame version
<ADD_S >	source unit address
<ADD_D >	destination unit address
<DATA0>	RIOC class
<DATA1>	command
<DATA2>	channel
<DATA3>	parameter
<DATA4>	parameter
<DATA5>	parameter
<DATA6>	parameter
<DATA7>	parameter

Protocol for RIOC Objects

With the RIOC messaging protocol, a set of remote-controllable objects is defined. The hardware developers can follow RIOC logical object definitions and make RIOC-compatible devices.

General IO

- Digital In

SETUP	01 00 <pin> <mode> <filter> <sample_interval_H[1]> <sample_interval_L[1]>
(rsp)	01 80 <pin> <ok>
READ	01 01 <pin>
(rsp)	01 81 <pin> <value>
SET_NOTIFICATION	01 02 <pin> <enable_notification>
(rsp)	01 82 <pin> <enable_notification >
(notify)	01 83 <pin> <value>

[1] microseconds

- Digital Out

SETUP	02 00 <pin> <mode>
(rsp)	02 80 <pin> <ok>
WRITE	02 01 <pin> <value>
(rsp*)	02 81 <pin> <value>
~~SET_PWM~~	~~02 02 <pin> <pwm_period>~~
~~(rsp)~~	~~02 82 <pin> <pwm_period>~~
WRITE_PWM	02 03 <pin> <pwm_value>
(rsp*)	02 83 <pin> <pwm_value>
PULSE	02 04 <pin> <value> <pw_H[1]> <pw_M[1]> <pw_L[1]>
(rsp*)	02 84 <pin> <value> <pw_H[1]> <pw_M[1]> <pw_L[1]>
READ	02 05 <pin>
(rsp)	02 85 <pin> <value>
READ_PWM	02 06 <pin>
(rsp*)	02 86 <pin> <pwm_value>

[1] microseconds

- Analog In

SETUP	03 00 <pin> <mode> <filter> <sample_interval_H[1]> <sample_interval_L[1]>
(rsp)	03 80 <pin> <ok>
READ	03 01 <pin>
(rsp)	03 81 <pin> <value_H> <value_L>

SET_NOTIFICATION

03 02 <pin> <enable_notification> <interval_H[2]> <interval_L[2]> <significant_bits[3]>

(rsp) 03 82 <pin> <enable_notification> <interval_H[2]> <interval_L[2]> <significant_bits[3]>

(notify) 03 83 <pin> <value_H> <value_L>

[1] microseconds
[2] milliseconds
[3] 0 or 1~16bits

- Analog Out

SETUP 04 00 <pin> <mode>

(rsp) 04 80 <pin> <ok>

WRITE 04 01 <pin> <value_H> <value_L>

(rsp*) 04 81 <pin> <value_H> <value_L>

READ 04 02 <pin>

(rsp) 04 82 <pin> <value_H> <value_L>

- UART Serial

SETUP 05 00 <port> <baud_H> <baud_M> <baud_L> <config[1]>

(rsp) 05 80 <port> <ok>

SEND 05 01 <port> <length> <byte1> <byte2> <byte3> <byte4>

(receive) 05 82 <port> <length> <byte1> <byte2> <byte3> <byte4>

[1]
5N1=0x00, 6N1=0x02, 7N1=0x04, 8N1=0x06, 5N2=0x08, 6N2=0x0A, 7N2=0x0C, 8N2=0x0E,
5E1=0x20, 6E1=0x22, 7E1=0x24, 8E1=0x26, 5E2=0x28, 6E2=0x2A, 7E2=0x2C, 8E2=0x2E,
5O1=0x30, 6O1=0x32, 7O1=0x34, 8O1=0x36, 5O2=0x38, 6O2=0x3A, 7O2=0x3C, 8O2=0x3E

- Multiple Digital In

SETUP 06 00 <pin> <number[1]> <mode>

(rsp) 06 80 <pin> <ok>

READ 06 01 <pin>

(rsp) 06 81 <pin> <value_bits1> <value_bits2> <value_bits3> <value_bits4>

SET_ NOTIFICATION

06 02 <pin> <enable_notification> <interval_H[2]> <interval_L[2]>

(rsp) 06 82 <pin> <enable_notification> <interval_H[2]> <interval_L[2]>

(notify) 06 83 <pin> <value_bits1> <value_bits2> <value_bits3> <value_bits4>

[1] 1~32 pins for digital in
[2] milliseconds

- Multiple Digital Out

SETUP 07 00 <pin> <number[1]> <mode>

(rsp) 07 80 <pin> <ok>

WRITE 07 01 <pin> <value_bits1> <value_bits2> <value_bits3> <value_bits4>

(rsp*) 07 81 <pin> <value_bits1> <value_bits2> <value_bits3> <value_bits4>

READ 07 02 <pin>

(rsp) 07 82 <pin> <value_bits1> <value_bits2> <value_bits3> <value_bits4>

[1] 1~32 pins for digital out

Motion

- DC Motor (2 Lines)

SETUP	11 00 <pin1[1]> <pin2[1]> <mode[1]>
(rsp)	11 80 <pin1> <ok>
RUN	11 01 <pin1> <dir> <power>
(rsp*)	11 81 <pin1> <dir> <power>
READ	11 02 <pin1>
(rsp)	11 82 <pin1> <dir> <power>

[1] pin1, pin2 are connected to motor v+/v- for mode 0; pin1, pin 2 are connected to motor PWM and DIR for mode 1.

- Stepper (4 Lines or PUL+DIR)

SETUP	12 00 <pin1[1]> <pin2[1]> <pin3> <pin4> <mode[1]>
(rsp)	12 80 <pin1> <ok>
STEP	12 01 <pin1> <dir> <steps_H> <steps_L>
(rsp*)	12 81 <pin1> <dir> <steps_H> <steps_L>
GOTO	12 02 <pin1> <pos_sign> <pos_H> <pos_M> <pos_L>
(rsp*)	12 82 <pin1> <pos_sign> <pos_H> <pos_M> <pos_L>
STOP	12 03 <pin1>
(rsp*)	12 83 <pin1>
SET_SPEED	12 04 <pin1> <speed_H[2]> <speed_L[2]>
(rsp*)	12 84 <pin1> <speed_H[2]> <speed_L[2]>
GET_SPEED	12 05 <pin1>
(rsp)	12 85 <pin1> <speed_H[1]> <speed_L[1]>
SET_POSITION	12 06 <pin1> <pos_sign> <pos_H> <pos_M> <pos_L>
(rsp*)	12 86 <pin1> < pos_sign> < pos_H> < pos_M> < pos_L>
GET_POSITION	12 07 <pin1>
(rsp)	12 87 <pin1> <pos_sign> <pos_H> <pos_M> <pos_L>

[1] pin1, pin2, pin3, pin4 are connected to stepper A+, A-, B+, B- for mode 0; pin1, pin 2 are connected to stepper PUL and DIR for mode 1.
[2] steps per second

- Servo (Rudder)

SETUP	13 00 <pin> <mode>
(rsp)	13 80 <pin> <ok>
SET_ANGLE	13 01 <pin> <angle[1]>
(rsp*)	13 81 <pin> <angle[1]>

GET_ANGLE 13 02 <pin>
(rsp) 13 82 <pin> <angle[1]>

SET_ENABLE 13 03 <pin> <enable>
(rsp*) 13 83 <pin> <enable>

GET_ENABLE 13 04 <pin>
(rsp) 13 84 <pin> <enable>

[1] 0 ~ 180 degrees

Sensor

- Encoder (A/B signals)

SETUP 21 00 <pin1> <pin2> <mode> <sample_interval_H[1]> <sample_interval_L[1]>
(rsp) 21 80 <pin1> <ok>

READ 21 01 <pin1>
(rsp) 21 81 <pin1> <value_sign> <value_H> <value_M> <value_L>

SET_ NOTIFICATION
21 02 <pin1> <enable_notification> <interval_H[2]> <interval_L[2]> <significant_bits[3]>
(rsp) 21 82 <pin1> <enable_notification> <interval_H[2]> <interval_L[2]> <significant_bits[3]>
(notify) 21 83 <pin1> <value_sign> <value_H> <value_M> <value_L>

WRITE 21 04 <pin1> <value_sign> <value_H> <value_M> <value_L>
(rsp*) 21 84 <pin1> <value_sign> <value_H> <value_M> <value_L>

[1] microseconds
[2] milliseconds
[3] 0 or 1~24bits

- Ultrasonic Ranger

SETUP 22 00 <pin1> <pin2> <mode>
(rsp) 22 80 <pin1> <ok>

RANGE 22 01 <pin1>
(rsp) 22 81 <pin1> <value_H1[1]> <value_L[1]>

[1] cm

- Thermometer

SETUP 23 00 <pin1> <mode>
(rsp) 23 80 <pin1> <ok>

MEASURE 23 01 <pin1>
(rsp) 23 81 <pin1> <temp_H[1]> < temp_L[1]> <humidity_H[2]> < humidity_L[2]>

[1] temperature / kelvins x 10
[2] humidity / 0 ~ 1000 mapped to 0 ~ 100.0%

Sound

- Tone (Frequency)

SETUP 31 00 <pin> <mode>
(rsp) 31 80 <pin> <ok>

PLAY 31 01 <pin> <frequency_H> <frequency_L> <duration_H[1]*> <duration_L[1]*>
(rsp*) 31 81 <pin> <frequency_H> <frequency_L> <duration_H[1]*> <duration_L[1]*>

STOP 31 02 <pin>
(rsp*) 31 82 <pin>

[1] milliseconds

Light

- RGB LED Strip (WS2812)

SETUP 41 00 <pin> <mode> <led_count_H> <led_count_L>
(rsp) 41 80 <pin> <ok>

SHOW_RGB 41 01 <pin> <led_id_H> <led_id_L> <red> <green> <blue>
(rsp*) 41 81 <pin> <led_id_H> <led_id_L> <red> <green> <blue>

SET_RGB 41 02 <pin> <led_id_H> <led_id_L> <red> <green> <blue>
(rsp*) 41 82 <pin> <led_id_H> <led_id_L> <red> <green> <blue>

SHOW 41 03 <pin>
(rsp*) 41 83 <pin>

GET_RGB 41 04 <pin>
(rsp) 41 84 <pin> <led_id_H> <led_id_L> <red> <green> <blue>

Communication

- IR Transmitter

SETUP 51 00 <pin> <mode>
(rsp) 51 80 <pin> <ok>

SEND 51 01 <pin> <format> <byte1> <byte2> <byte3> <byte4>
(rsp*) 51 81 <pin> <format> <byte1> <byte2> <byte3> <byte4>

- IR Receiver

SETUP 52 00 <pin> <mode>
(rsp) 52 80 <pin> <ok>

(receive) 52 81 <pin> <format> <byte1> <byte2> <byte3> <byte4>

Universal Commands for All Objects

- Silence Mode (no response for * marked items)

SET_SILENCE	<object> 70 <pin/port> <silent>
(rsp)	<object> F0 <pin/port> <silent>

System Reserved Control for Units

- Unit Control

(start)	00 8f <ver_1> <ver_2> <desc_1> < desc_2> < desc_3> < desc_4>
RESET	00 01
(rsp!)	0 81 <ver_1> <ver_2> <desc_1> < desc_2> < desc_3> < desc_4>
VERSION	00 02
(rsp)	00 82 <ver_1> <ver_2> <desc_1> < desc_2> < desc_3> < desc_4>
SET_ID	00 03 <unit_id>
(rsp)	00 83 <unit_id>
SYNC_BEGIN	00 04
(rsp)	00 84
SYNC_END	00 05
(rsp)	00 85
SLEEP	00 06 <duration_H[1]> <duration_L[1]> <rspMode [2]>
(rsp)	00 86 <duration_H[1]> <duration_L[1]>
SET_SILENCE	00 07 <silent>
(rsp)	00 87 <silent>

[1] milliseconds

[2] mode 0 = no response, mode 1 = response before sleep, mode 2 = response after sleep, mode 3 = response for both

System Reserved Control for App

- App Control

SIGN_IN	00 00 <app_sign>
(rsp)	00 80 <connection_count>