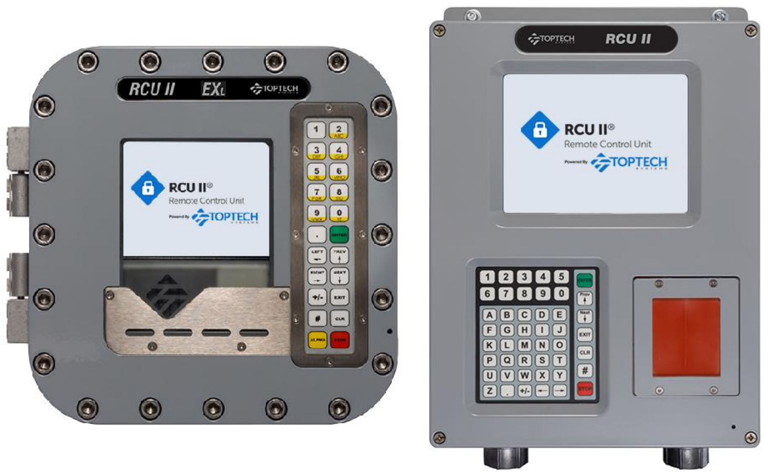
RCU II/II+ Open Protocol Communications Guide



MultiLoad/RCU II Hardware
(with RCU II Firmware)

MultiLoad/RCU III Hardware
(with RCU II+ Firmware)

Firmware Version 10.36.01 & 11.36.01

September 8, 2023

Copyright Notice

Disclaimer

Toptech Systems assumes no responsibility for damages resulting from installation or use of its products. Toptech Systems will not be liable for any claims of damage, lost data, or lost time as a result of using its products.

TOPTECH SYSTEMS

MULTILOAD II Integrated Flow Controller

MULTILOAD II SCS Skid Control System

MULTILOAD II SMP Single Meter Preset

RCU II Remote Control Unit

TMS6 Terminal Management System

UAP Unified Automation Platform

Table of Contents

CHAPTER 1 PROTOCOL SPECIFICATION

The RCU auto-detects three low level computer protocols for RS-232 or RS-485 multi-dropped installations:

- Smith Protocol
- Brooks Protocol
- Daniels/Modbus-RTU Protocol

Note: All unknown or not allowed signal or code sequences are rejected and have no impact on the software or measurement data.

1.1 Definitions

The non-printing characters that form the skeleton of the Smith and Brooks protocols are standard ASCII (American Standard Code Information Interchange):

ASCII CHARACTER	DECIMAL	HEX	BINARY
NUL	0	0	00000000
STX	2	2	00000010
ETX	3	3	00000011
SOH	1	1	00000001
PAD	127	7F	01111111

CHARACTER	DESCRIPTION
BCC	Block Check Characters. The ASCII hex representation of the binary sum of all the data in the message from the SOH through the ETX character.
LRC	Longitudinal Redundancy Check. The LRC is an ASCII character computed as the exclusive or (XOR) sum of all characters following the STX and including the ETX.
CRC	Cyclic Redundancy Check.
A1..A3	A 3-character ASCII unit address of the RCU. Please refer to the RCU User Guide for configuring the unit address.
D1..Dn	Data field characters.
Fn	Function field code.
Adr	A single character binary unit address of the RCU.

1.1.1 Smith Protocol

The Smith protocol is compatible with devices from Smith Meter, such as the Accuload I and Accuload II mini computer modes of operation. Using this protocol, RCU accepts data in the format:

NUL	STX	A1	A2	data	ETX	LRC	PAD

A1 and A2 are the last two digits of the unit address in the RCU configuration.

1.1.2 Brooks Protocol

The Brooks protocol is compatible with devices from Brooks Instruments such as their Petrocount RAU™ and IMS Control™ units running in computer mode. Using this protocol, RCU accepts data in the format:

SOH	DESTINATION			SOURCE			STX	data	ETX	BCC 1	BCC 2
	A1	A2	A3	S1	S2	S3					

1.1.3 Daniels/Modbus-RTU Protocol

s Flow Products. Using this protocol, RCU accepts data in the format:

Quiet Time	ADR	FN	LEN (2 – 252)	D1...DN	CRC1	CRC2	Quiet Time

Note: Daniels/Modbus-RTU protocol messages are framed by a quiet time of three and one-half characters.

ADR is binary character of the address of the RCU. Typically 0x01.

D1…Dn is string data containing the commands listed in this manual.

Note: Fn is expected to be 0x41/0x42 alternating on each command. Responses will have Fn as 0x41/0x42 for normal responses and 0xc1/0xc2 for exception responses. Fn codes are defined by the device and not the protocol.

Note: Modbus extention to larger packet sizes: On messages with data packet sizes from 2 to 252 characters the Fn values of 0x41/0x42 and 0xC1/0xc2 values will be used. With messages outside this range Fn will be the MSB value of the data size and the Len will be LSB of the data size.

For Example:

Fn = 0x41, Len = 0x80, when data packet size = 0x0080,

Fn = 0x42, Len = 0x80, when data packet size = 0x0080,

Fn = 0x00, Len = 0xFF, when data packet size = 0x00FF (255),

Fn = 0x01, Len = 0x00, when data packet size = 0x0100 (256),

Fn = 0x01, Len = 0x01, when data packet size = 0x0101 (257),

Fn = 0x02, Len = 0x00, when data packet size = 0x0200 (512),

Fn = 0x04, Len = 0x00, when data packet size = 0x0400 (1024),

Fn = 0x08, Len = 0x00, when data packet size = 0x0800 (2048).

CHAPTER 2 RCU MESSAGE SET

Regardless of protocol, the RCU commands must be transmitted to the RCU in the data portion of the protocol message.

All valid command replies start is an "A" (Ack) in the first character.

All invalid command replies start is an "N" (Nak) in the first character.

Returned parameters are returned starting at the second character.

Failure to receive a command after 60 seconds will cause the RCU to display a “System Unavailable Message”.

Valid commands are:

Definition	Command	Reply
Query Card #	QC	<card data, if card present>
Clear Display	CD	A
Display Text	DT#<text>	A (# is line number in hex 0x0-0xF)
Key Input	KI	A (one term key press, no cursor)
String Input	SI	A (max 20 characters with cursor display)
Query Input	QI	<term key>,<string> where: <term key> of NEXT='a', PREV='b', ABORT/EXIT='c', ENTER='d', CLR='e', STOP='s', Card Pull=’F’
Abort Input	AI	A (forces end of input with EXIT term key)
Reset Display	RD	A (Send display back to idle message)
Output 1 On	OUT1ON	A Turns On: RCU II+ digital output fcm 0 port 5 RCU II digital DC output port 10 RCU I digital out 1
Output 1 Off	OUT1OFF	A Turns Off: RCU II+ digital output fcm 0 port 5 RCU II digital DC output port 10 RCU I digital out 1
Relay 1 On	RLY1ON	A Turns On: RCU II+ digital output fcm 0 port 0 RCU II digital AC output port 0 RCU I relay 1
Relay 1 Off	RLY1OFF	A Turns On: RCU II+ digital output fcm 0 port 0 RCU II digital AC output port 0 RCU I relay 1
Output Port 0 On	PORT0ON	A Turns On: RCU II+ digital output fcm 0 port 0 RCU II digital AC output port 0
Output Port 0 Off	PORT0OFF	A Turns Off: RCU II+ digital output fcm 0 port 0 RCU II digital AC output port 0

Output Port 1 On	PORT1ON	A Turns On: RCU II+ digital output fcm 0 port 1 RCU II digital AC output port 1
Output Port 1 Off	PORT1OFF	A Turns Off: RCU II+ digital output fcm 0 port 1 RCU II digital AC output port 1
Output Port 2 On	PORT2ON	A Turns On: RCU II+ digital output fcm 0 port 2 RCU II digital AC output port 2
Output Port 2 Off	PORT2OFF	A Turns Off: RCU II+ digital output fcm 0 port 2 RCU II digital AC output port 2
Output Port 3 On	PORT3ON	A Turns On: RCU II+ digital output fcm 0 port 3 RCU II digital AC output port 3
Output Port 3 Off	PORT3OFF	A Turns Off: RCU II+ digital output fcm 0 port 3 RCU II digital AC output port 3
Output Port 8 On	PORT8ON	A Turns On: RCU II+ digital output fcm 0 port 4 RCU II digital AC output port 8
Output Port 8 Off	PORT8OFF	A Turns Off: RCU II+ digital output fcm 0 port 4 RCU II digital AC output port 8
Output Port 10 On	PORT10ON	A Tuns On: RCU II+ digital output fcm 0 port 5 RCU II digital DC output port 10
Output Port 10 Off	PORT10OFF	A Tuns Off: RCU II+ digital output fcm 0 port 5 RCU II digital DC output port 10
Output Port 11 On	PORT11ON	A Turns On: RCU II+ digital output fcm 0 port 6 RCU II digital DC output port 11
Output Port 11 Off	PORT11OFF	A Turns Off: RCU II+ digital output fcm 0 port 6 RCU II digital DC output port 11
Output Port 12 On	PORT12ON	A Turns On: RCU II+ digital output fcm 0 port 7 RCU II digital DC output port 12
Output Port 12 Off	PORT12OFF	A Turns Off: RCU II+ digital output fcm 0 port 7 RCU II digital DC output port 12

Output FCM and Port On	PORT_fpON	A Turns On: RCU II+ output fcm f port p where f is fcm number 0-3 and p is port 0-7 to set
Output FCM and Port Off	PORT_fpOFF	A Turns OFF: RCU II+ output fcm f port p where f is fcm number 0-3 and p is port 0-7 to set
Draw Display Graphics	DDG…	Writes a graphic bitmap directly to the display.
Read Graphics Cache	RGC…	Reads data from the graphic cache.
Write Graphics Cache	WGC…	Writes data to the graphic cache.
Draw Graphic Cache	DGC…	Displays data from the graphic cache.
Draw Progress Box	DPB…	Displays a progress box outline,
Update Progress Box	UPB…	Updates the progress in a progress box.
Display Attributed Text	DAT…	Positions the cursor and writes text with the specified attributes.

2.1 Card Data Format (QC)

QC => Asss…

Where:

sss… = Driver card data string.

For HID 26-bit Prox Cards,

0=yy=0000000=xxxxxxxx=01 – Card number that was read from the card.

Where:

yy is the facility number (00-99)

xxxxxxxx is the card number (00000000-00065535).

For HID 26-bit Corporate 1000 Formatted Prox Cards,

0=yyyy=00000=xxxxxxxx=01 – Card number that was read from the card.

Where:

yyyy is the company number (0000-4095)

xxxxxxxx is the card number (00000000-01048575).

For HID 37-bit H10320 formatted Prox cards,

0=yy=0000000=xxxxxxxx=01 – Card number that was read from the card.

Where:

yy is the facility number (Always 00).

xxxxxxxx is the card number (00000000-99999999).

For TWIC cards,

1111,2222,333333,4,5,6666666666788889 – TWIC Card FASC-N that was read from the card.

Where,

1111 = Agency Code

2222 = System Code

333333 = Credential Number

4 = Credential Series

5 = Individual Credential Issue

6666666666 = Person Identifier (Used as Card Number in Access ID Database)

7 = Organizational Category

8888 = Organizational Identifier

9 = Person/Organization Association

2.2 Draw Display Graphics (DDG…)

DDGrcwd => A

Where:

r = (character) 0x20 + Top Text Row Position (0 = top row, 15 = bottom row)

c = (character) 0x20 + Left Text Column Position (0 = leftmost column, 39 = rightmost column)

w = (character) 0x20 + Width in Text Columns

d = (character) 0x20 + pixel color palette index values from left to right, top to bottom,

Note: Each text column = 16 pixels wide, each text row = 30 pixels high

Pixel color palette index values defined as:

Black	= 0x00
Blue	= 0x01
Red	= 0x02
Magenta	= 0x03
Green	= 0x04
Cyan	= 0x05
Yellow	= 0x06
White	= 0x07
Custom 0	= 0x08 (Not available)
Custom 1	= 0x09 (Not available)
Custom 2	= 0x0a (Not available)
Custom 3	= 0x0b (Not available)
Custom 4	= 0x0c (Not available)
Custom 5	= 0x0d (Not available)
Flash 1	= 0x0e
Flash 2	= 0x0f

Example: To position a small graphic starting at row 5, column 20, column width 2, code the string as follows:

```
DDG%4”    ’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’
          ’’’’’$$$’’’’’’’’’’’’’’’’$$$’’’’’
          ’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’
          ’’’’’’’’’’’’’’$$$’’’’’’’’’’’’’’’
          ’’’’’$$’’’’’’’’’’’’’’’’’’$$’’’’’
          ’’’’’’’$$$$’’’’’’’’’’’$$$$’’’’’’’
          ’’’’’’’’’’’$$$$$$$$$$’’’’’’’’’’’

Note: formatted for illustration only, there are no spaces or CRs in the above
command.
```

'%' = 0x20 + 5 '4' = 0x20 + 20 “ = 0x20 + 2

‘ = 0x20 + 0x07 (White) $ = 0x20 + 0x04 (Green)

2.3 Read Graphics Cache (RGC…)

RGCaaaaaaaa => Ad…

Where:

aaaaaaaa = 8 character hexadecimal cache offset start value.

Cache size = 16MB, Cache offset values = 0x00000000 - 0x00ffffff

Two pixels per cache byte.

Note: Graphics Cache is in volatile memory. On power cycle, entire Graphics Cache will be populated with 0's (spaces) for pixel data. Use RGC command to verify cache has not been cleared.

Note: Graphics Cache memory is also used for temporary storage of uploaded firmware images before writing into flash memory. Do not use the RGC, WGC or DGC commands while uploading or flashing new firmware.

d… = 64 (characters) 0x20 + pixel color palette index values,

See DDG command for more details on pixel data.

2.4 Write Graphics Cache (WGC…)

WGCaaaaaaaad… => A

Where:

aaaaaaaa = 8 character hexadecimal cache offset start value.

See RGC command for more details on cache offset.

d… = even number of (characters) 0x20 + pixel color palette index values,

See DDG command for more details on pixel data.

2.5 Display Graphics Cache (DGC…)

DGCrcwhaaaaaaaa => A

Where:

r = (character) 0x20 + Top Text Row Position (0 = top row, 15 = bottom row)

c = (character) 0x20 + Left Text Column Position (0 = leftmost column, 39 = rightmost column)

w = (character) 0x20 + Width in Text Columns

h = (character) 0x20 + Height in Text Columns

aaaaaaaa = 8 character hexadecimal cache offset start value.

See RGC command for more details on cache offset.

2.6 Display Progress Box (DPB…)

DPBrcwh => A

Where:

r = (character) 0x20 + Top Text Row Position (0 = top row, 15 = bottom row)

c = (character) 0x20 + Left Text Column Position (0 = leftmost column, 39 = rightmost column)

w = (character) 0x20 + Width in Text Columns

h = (character) 0x20 + Height in Text Rows Down

2.7 Update Progress Box (UPB…)

UPBrcwhp => A

Where:

r = (character) 0x20 + Top Text Row Position (0 = top row, 15 = bottom row)

c = (character) 0x20 + Left Text Column Position (0 = leftmost column, 39 = rightmost column)

w = (character) 0x20 + Width in Text Columns

h = (character) 0x20 + Height in Text Rows Down

p = (character) 0x20 + Percent Value to Display (0 = Nothing, 100 = Complete)

2.8 Display Attributed Text (DAT…)

DATrcsfbt… => A

Where:

r = (character) 0x20 + Start Text Row Position (0 = top row, 15 = bottom row)

c = (character) 0x20 + Start Text Column Position (0 = leftmost column, 39 = rightmost column)

s = (character) 0x20 + Font Size (0 = normal, 1 = double sized font)

f = (character) 0x20 + Foreground Color Palette Index

b = (character) 0x20 + Background Color Palette Index

t = ASCII or Unicode (BMP UTF-8 encoding) text to display.

Pixel color palette index values defined as:

Black	= 0x00
Blue	= 0x01
Red	= 0x02
Magenta	= 0x03
Green	= 0x04
Cyan	= 0x05
Yellow	= 0x06
White	= 0x07
Custom 0	= 0x08 (Not available)
Custom 1	= 0x09 (Not available)
Custom 2	= 0x0a (Not available)
Custom 3	= 0x0b (Not available)
Custom 4	= 0x0c (Not available)
Custom 5	= 0x0d (Not available)
Flash 1	= 0x0e
Flash 2	= 0x0f

Example: To position the large font green word 'Hello' at row 5, column 20, code the string as follows:

```
DAT%4!!$’Hello
```

'%' = 0x20 + 5 '4' = 0x20 + 20 ! = 0x20 + 1

$ = 0x20 + 0x04 (Green) ‘ = 0x20 + 0x07 (White)

2.9 Typical Message Exchange

2.9.1 Wait for card in

"QC" (loop while reply is null)

2.9.2 Display product selection screen.

"CD"

"DT0--------------Product List--------------"

"DT1 1) Unlead"

"DT2 2) Premium"

"DT3 3) High Sulpher Diesel (with dye)"

"DT4 4) Low Sulpher Diesel"

"DT5 4) Low Sulpher Diesel (with dye)"

"DT6 5) Kero (with dye)"

"DT7 6) Jet"

"DT8   Enter Selection: "

"SI"

"QI" (loop while termination key is " ")

2.9.3 Display volume entry screen

"CD"

"DT0-------------Delivery Volume------------"

"DT1   Enter Volume: "

"SI"

"QI" (loop while termination key is " ")

2.9.4 Display summary screen

"CD"

"DT0------------Delivery Summary------------"

"DT1 Product: High Sulpher Diesel (with dye)"

"DT2 Volume: 2500"

"DT4 Press Enter to Accept, Exit to Reject."

"KI"

"QI" (loop while termination key is " ")

2.9.5 Display card out message.

"CD"

"DT2   Thanks For Loading At"

"DT3   Joe's Gas"

"DT5   Please Remove Your Card"

"QC" (loop while not null)

"RD" (when done)

CHAPTER 3 DATA COMMUNICATION NOTES AND RECOMMENDATIONS

This topic provides some basic communications concepts, as well as Toptech's recommendations for achieving optimum performance.

3.1 Protocol Selection: RS-232 vs. RS-485

3.1.1 RS-232

RS-232 communications protocol was designed for point-to-point (i.e., computer to a single device) communications for short distances. The actual specification for RS-232 distance is limited to 50 feet. In practice, however, RS-232 communications can be successful at distances over 1000 feet. We generally try to limit distances to 500-600 feet. RS-232 requires a minimum of 3 wires: Transmit, Receive and Signal ground wires. More wires are required for hardware handshaking. RS-232 can work on straight, non twisted-pair wiring. If existing wiring that is not twisted-pair must be used, then RS-232 protocol should be used.

3.1.2 RS-485

RS-485 communications protocol was designed for multi-point (i.e., computer to multiple devices, also called multi-dropped) communications and can support distances over 5000 feet. RS-485 requires 4 wires (2 twisted-pair) for normal full-duplex communications. With special hardware, 2-wire half-duplex RS-485 can be accomplished. RS-485 utilizes a transmit pair of wires (TDA and TDB) and a receive pair of wires (RDA and RDB). A ground wire is not recommended normally, although some devices may have a terminal block for an RS-485 ground. Most multi-dropped devices require RS-485 communications. When installing new wiring, RS-485 twisted-pair wiring is preferred.

3.1.3 Cable Selection

The speed and distance of communications is mostly affected by the capacitance and resistance of the wiring. Copper wiring is generally low resistance, so this is not normally the limiting factor. Capacitance, however, can vary greatly from one type of wire to another. In general, the larger the wire, the higher the capacitance. Wiring size of 20 to 24 gauge is typically best for data communications. Capacitance of 16 pf or less is preferred. With 16pf cable, 9600 baud communications can operate over 1000 feet for RS-232, and over 5000 feet for RS-485. Doubling the capacitance will generally halve the baud rate or the distance.

Data communications cable should always be shielded. Individual wires or pairs do not need shielding, but an overall shield should always be used. Whenever possible, always run data communications cable in conduit and protected from moisture. Moisture can invade most cable coatings and disrupt data communications integrity. For exposed routing or direct burial, Teflon coatings are recommended. Armored cable is also available for running over-head without conduit.

RS-232 communications will work over almost any type of wiring, although speed and distance may be limited by some wiring. Straight (individual) wires or twisted-pair wiring can be used. RS-485 communications requires twisted-pair wiring with an impedance of 100 ohms.

For new installations, if you select wiring suitable for RS-485 wiring, it will always work for RS-232. This will provide the ability to change from one communications protocol to another without replacing the wiring.

3.2 Line Termination in Multi-dropped Communications

Some vendors, particularly Smith Meters and Toptech Systems, have specialized hardware that will support multi-dropping of Acculoads and RCUs on a single RS-232 communication line. This is accomplished by tri-stating or disconnecting the transmitter of a device when it is not transmitting on the communications line. This is similar to the RS-485 multi-dropping method. Due to the tri-stating of the transmitters, the transmit circuit is 'floating' when no device is transmitting. This can sometimes cause problems for the computer's receiver circuit, usually framing errors or break conditions.

To keep these problems under control, multi-dropped lines need line termination resistors installed. This can be installed on the receiver of the line driver for the line. For RS-232 communications, a 500 to 1000 ohm resistor is placed between the receive (RD) and signal ground (SG) wires of the line driver for the line. For RS-485 communications, a 220 to 270 ohm resistor is placed between the receive pair (RDA and RDB) wires of the line driver for the line. Termination resistors actually weaken the driver's ability to transmit; therefore, when longer lines are used or many devices are multi-dropped, the higher resistor values must be used. Unfortunately, the proper value must sometimes be determined by experimentation. For RS-232, the resistor value must be large enough to allow the transmitter to drive the voltage levels to at least +3VDC and -3VDC. For RS-485, the resistor value must be large enough to allow the transmitter to produce a .25 VDC difference in the transmit pair, but not large enough to disrupt the balance of the line. Too low of a resistor value will typically cause a short on the data communications line and inhibit communications.

Please note that in RS-485 communications specifications, you will find that 100 ohm termination resistors are recommended. This is a different type of line termination than what we are discussing here. The 100 ohm termination is specifically for line balancing and is typically not required for baud rates less then 38,000 baud. This is well above most PC data communications capabilities.

3.3 Optical Isolation

Due to the major power fluctuations that may occur in an industrial environment and the effect of lightning storms in an area with lots of piping, we recommend that any data communications line connecting devices powered from different AC circuits be protected with optical isolation devices. The RS-232 to RS-485 converters that Toptech supplies provide this isolation. We also provide an RS-232 optical isolation device for RS-232 lines. Similar devices are available from Black Box Corporation and Burr-Brown. One isolation device must be installed on each data communications line. This is different from line drivers or short-haul modems that require a device on each end.

Please note that any line termination used must be installed on the field side of the optical isolator. Also, optical isolators are directional, which means that one side is for the computer (DCE) and the other side is for the field equipment (DTE). They will not transmit data if installed backwards!

3.4 Shield Grounding

Due to all of the electrical noise generated by an industrial environment, all data communications cable must be properly shielded. If not properly shielded, communications may work for a while, but you will most probably experience intermittent communications errors and outages. Most data communications cable comes with good shielding built in, but if not properly installed, the shielding will be ineffective. Proper installation requires that the shield be attached to earth ground on ONE END ONLY, typically at the junction box on the computer end, and MUST BE CONTINUOUS through all junction boxes out to the field equipment. In addition, the shield should be taped back at each field device and should NEVER be connected to a field device. Unfortunately, several of the field device manufacturers have terminal blocks labeled for attachment of the shield, and many of their installation drawings indicate that the shield should be attached to the field devices. **Under no circumstances should the shield ever be attached to a field device!**

When connecting through a junction box, care must be taken that the shields be treated just like any other wire. The shield must remain continuous across the junction box. Don't tie multiple shields together. Route each shield across the junction box, making sure that it does not short to the box or any other point. Inside of a junction box, wires are typically unshielded. If a junction box is installed in an electrically noisy area, near motors, etc., the junction box must be able to provide shielding. Also, AC wiring should not be run into a junction box that has unshielded data communications wiring. Especially, AC circuits with loads that are switching off and on or that have a high current flow must be avoided. These will introduce noise into the data communications lines inside of the junction box.