*Open*Link™ Wire and Wireless Open Data Communication Protocol

***Open*Link™** is a reliable, routing, multi-drop, peer-to-peer, and peer-to-peer-multi-peer data communications protocol developed by Dexter Fortson Associates, Inc. and used by the DFA Remote Terminal Unit (RTU). The use of this communication protocol is not legally restricted and is available to all.

The table below indicates the data structure and the type of information contained in each byte of the data packet.

BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7-x	BYTE x-y	BYTE y+2
Group Address	RTU Address	Byte 1	Byte 2	Byte 3	Packet Length	Data	SCC-RTU	CRC16
Next Step		Control			Entire Packet	Optional	Route	Entire Packet

BYTE 1 Group Address — Next Step Group, 1 – 254, 0 and 255 illegal.

BYTE 2 RTU Address — Next Step RTU, 1 – 254, 0 and 255 illegal.

BYTE 3 Control Byte 1

```
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |    Eight bits or bits 0 – 7 equal one byte.
  |   |
  |   + - - - - - - - - - - - - -    Bits 0 – 6 must be 0.
  |
  + - - - - - - - - - - - - - - -    Bit 7 is always 1.
```

BYTE 4 Control Byte 2

```
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |    Eight bits or bits 0 – 7 equal one byte.
  |   |   |   |   |
  |   |   |   |   + - - - - - - -    0 - F or 0 - 3; Designates Starting Bank;
  |   |   |   |                      There are 8 bytes per Bank.
  |   |   |   + - - - - - - - - -    Request = 0; Acknowledge = 1
  |   |   |
  |   |   + - - - - - - - - - - -    AC Power Loss = 0, AC Power OK = 1;
  |   |                              Bit 5 is ignored by the RTU.
  |   + - - - - - - - - - - - - -    Write or Data = 0; Request or No Data = 1
  |
  + - - - - - - - - - - - - - - -    Save Route = 0; Don't Save Route = 1
                                     Bit 7 is ignored by SCC.
```

BYTE 5 Control Byte 3

```
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |    Eight bits or bits 0 – 7 equal one byte.
  |   |   |   |   |
  |   |   |   |   + - - - - - - -    0 – F or 0 - 3; Designates the Number of Banks,
  |   |   |   |                      (0 = 1 Bank)
  |   |   |   + - - - - - - - - -    Incoming = 0; Outgoing Message = 1;
  |   |   |                          (Routing Direction)
  |   |   + - - - - - - - - - - -    Must be 0 = 0
  |   |
  |   + - - - - - - - - - - - - -    Nothing = 0; Routing by RTU-RTU Lookup = 1;
  |
  + - - - - - - - - - - - - - - -    Nothing = 0; Routing by Group-RTU Lookup = 1;
```

BYTE 6 Packet Length — Includes CRC16 or error detection checksum. Maximum Length = 255.

BYTE 7-x Optional Data — Number of Banks in packet. There are up to 8 banks with 8 bytes per bank.

BYTE x+1-y Route Steps — Group-RTU or RTU-RTU Addressing; Minimum of two steps required. Example: Group-RTU, Group-RTU, etc. or RTU, RTU, RTU, etc.

BYTE y+1-z	CRC16	Standard 16-bit CRC zero seed. Cyclic Redundancy Check (CRC) is a function used to produce a checksum for error detection.

Standard Bank Structure for Protocol (8 bytes per bank)

Bank 0 = Digital Inputs or Outputs 1 through 64.

Bank 1 = Analog Inputs or Outputs 1 through 8.

Bank 2 = Analog Inputs or Outputs 9 through 16.

Bank 3 = Counter Inputs.

Bank 4 = Upper bank used for software purposes.

Bank 5 = Upper bank used for software purposes.

Bank 6 = Upper bank used for software purposes.

Bank 7 = Upper bank used for software purposes.

Structure of One Byte of Data

When used as an Analog there is 1 Analog per Byte.
When used as Digitals there are 8 Digitals per Byte.
Value = 0 to 255 decimal or 00 to FF Hex.
Decimal Equivalent = Added values of bits **ON** or = 1.

```
     Bits of One Byte
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
  |   |   |   |   |   |   |   |
  |   |   |   |   |   |   |   |
  |   |   |   |   |   |   |   |
  |   |   |   |   |   |   |   + - - -                  Digital 1   Decimal Equivalent = 1; Hex Equivalent = 1
  |   |   |   |   |   |   + - - - - - -                Digital 2   Decimal Equivalent = 2; Hex Equivalent = 2
  |   |   |   |   |   + - - - - - - - -                Digital 3   Decimal Equivalent = 4; Hex Equivalent = 4
  |   |   |   |   + - - - - - - - - - - -              Digital 4   Decimal Equivalent = 8; Hex Equivalent = 8
  |   |   |   + - - - - - - - - - - - - - -            Digital 5   Decimal Equivalent = 16; Hex Equivalent = 1
  |   |   + - - - - - - - - - - - - - - - - -          Digital 6   Decimal Equivalent = 32; Hex Equivalent = 2
  |   + - - - - - - - - - - - - - - - - - - - -        Digital 7   Decimal Equivalent = 64; Hex Equivalent = 4
  + - - - - - - - - - - - - - - - - - - - - - - -      Digital 8   Decimal Equivalent = 128; Hex Equivalent = 8
```

Hexadecimal Explanation

Hexadecimal values break a byte into two nibbles, the upper 4 bits of a byte and the lower 4 bits of a byte. The decimal equivalent of a nibble will vary from 0 to 15. Since only 1 character can be occupied in the data stream to represent the value being displayed, the letters **A** through **F** are utilized. A decimal 10 is displayed in Hex as an **A**, a decimal 11 is displayed in Hex as an **B**, a decimal 12 is displayed in Hex as an **C**, a decimal 13 is displayed in Hex as a **D**, a decimal 14 is displayed in Hex as a **E**, and a decimal 15 is displayed in Hex as a **F**.

For the following examples, each table represents the organization and information contained in a data packet. The top row of each table designates the byte in the information packet in sequential order, the second row is the actual information contained in the specified byte of the table column and the third and fourth rows are information about the structure or values of the information of the specified byte of the table column.

<u>Example</u>: Short Frame Request and Response using Direct Routing

The following is an example of a short frame request of Bank 0 data or digital data. Saves a direct Group-RTU route. Outgoing message is from Group 3, RTU 9 to Group 3, RTU 1.

1	2	3	4	5	6	7	8	9	10	11	12
03	01	80	60	90	0C	03	09	03	01	00	00
Next Step		1	2	3	Packet Length	SCC		RTU		CRC16	
		Control Bytes				Route					

The following is an example of the response from Group 3, RTU 1 of the Bank 0 data short frame request. Note that this is an incoming message from Group 3, RTU 1 to Group 3, RTU 9. Digital input DI_1 is **ON**. Digital inputs DI_2 through DI_64 are **OFF**. Digital inputs DI_9 through DI_63 are **ON**.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
03	09	80	30	80	14	01	FF	FF	FF	FF	FF	FF	7F	03	09	03	01	00	00
Next Step		1	2	3	Packet Length	1	2	3	4	5	6	7	8	SCC		RTU		CRC16	
		Control Byte				Data								Route					

<u>Example</u>: Data Transfer Request and Response using Direct Routing

The following is an example of a standard data transfer request. Writing Bank 0 data or digital data. Uses a Group-Station route. Sets digital output DO_1 and DO_3.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
03	01	80	20	90	14	05	00	00	00	00	00	00	00	03	09	03	01	00	00
Next Step		1	2	3	Packet Length	1	2	3	4	5	6	7	8	SCC		RTU		CRC16	
		Control Byte				Data								Route					

The following is an example of the response from Group 3, RTU 1 from standard data transfer request.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
03	09	80	30	80	14	01	FF	FF	FF	FF	FF	FF	7F	03	09	03	01	00	00
Next Step		1	2	3	Packet Length	1	2	3	4	5	6	7	8	SCC		RTU		CRC16	
		Control Byte				Data								Route					

Example: **Short Frame Request and Response using One Repeater Routing**

The following is an example of a short frame request of Bank 0 data. Saves Group-RTU route. Outgoing message is from Group 3, RTU 9 to Group 3, RTU 1 repeating through Group 4, RTU 7.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
03	01	80	60	90	0E	03	09	04	07	03	01	00	00
Next Step		1	2	3	Packet Length	SCC		Step 1		RTU		CRC16	
		Control Bytes				Route							

The following is an example of the response from Group 3 RTU 1 of the Bank 0 data short frame request. Note that this is an incoming message from Group 3, RTU 1 to Group 3, RTU 9 repeating through Group 4, RTU 7. Digital input DI_1 is **ON**. Digital input DI_2 through DI_8, and DI_64 are **OFF**. Digital inputs DI_9 through DI_63 are **ON**.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
03	09	80	30	80	16	01	FF	FF	FF	FF	FF	FF	7F	03	09	04	07	03	01	00	00
Next Step		1	2	3	Packet Length	1	2	3	4	5	6	7	8	SCC		Step 1		RTU		CRC16	
		Control Bytes				Data								Route							

Example: **Data Transfer Request and Response Using Two Repeater Routing**

The following is an example of a standard data transfer request. Writing Bank 0 data. Saves Group-RTU route. This is an outgoing message from Group 3, RTU 9 to Group 3, RTU 1 repeating through Group 4, RTU 7 and Group 5, RTU 2. Sets digital output DO_1 and DO_3.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
03	01	80	20	90	18	05	00	00	00	00	00	00	00	03	09	04	07	05	02	03	01	00	00
Next Step		1	2	3	Packet Length	1	2	3	4	5	6	7	8	SCC		Step 1		Step 2		RTU		CRC16	
		Control Bytes				Data								Route									

The following is an example of the response from Group 3, RTU 1 from standard data transfer request. This is an incoming message from Group 3, RTU 1 to Group 3, RTU 9 repeating through Group 5, RTU 2 and Group 4, RTU 7.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
03	09	80	30	90	18	01	FF	FF	FF	FF	FF	FF	7F	03	09	04	07	05	02	03	01	00	00
Next Step		1	2	3	Packet Length	1	2	3	4	5	6	7	8	SCC		Step 1		Step 2		RTU		CRC16	
		Control Bytes				Data								Route									

Example: **Group-RTU Routing Segment**

This example illustrates the Group-RTU routing segment of an data packet. This example uses four steps, from Group 3, RTU 9 through Group 5, RTU 2 and Group 4, RTU 7 to Group 3, RTU 1. Note that in Group-RTU routing, the RTUs do not have to be in the same group. Routing is relative to the SCC. Therefore, when sending an outgoing message, the route is examined from SCC to RTU. Conversely, when sending an incoming message, the route is examined from RTU to SCC.

1	2	3	4	5	6	7	8
03	09	05	02	04	07	03	01
SCC		Step 1		Step 2		SCC	
Route							

Example: **RTU-RTU Routing Segment**

This example illustrates the RTU-RTU routing segment of an data packet. This example uses four steps, from Group 3, RTU 9 through Group 3, RTU 2 and Group 3, RTU 5 to Group 3, RTU 3. Note that in RTU-RTU routing, all RTUs must be in the same group. Routing is relative to the SCC. Therefore when sending an outgoing message, the route is examined from SCC to RTU. Conversely, when sending an incoming message, the route is examined from RTU to SCC.

1	2	3	4
09	02	05	03
SCC	Step 1	Step 2	RTU
Route			