

## BF-1010 DIO Command Protocol via TCP/IP

### Revised History

1. 2011-05-04: wrong typing of command field length as 1, it should be corrected to 2

In this DIO command protocol is described here to let customer's remote management software to access Digital I/O state through TCP/IP.

### Command Packet Format:

Length (Bytes)	2	2	32	32	2	1
	Start Flag	Command	Data1	Data2	End Flag	CRC

### Command List

Command Code	Description	Note
0x0001	Read Digital I/O state	
0x0003	Trigger Digital I/O	
0x0005	E-mail Alarm Trigger	
0x0007	Reboot	※
0x0009	Reset System Default	※
0x000B	Backup System Configuration	
0x000D	Restore System Configuration	※
0x0011	Set I/O working rule	
0x0013	Get I/O working rule	
0x0015	Get I/O triggering count	

※ **BF1010 will reboot after reply.**

### Return Packet Format:

Length (Bytes)	2	2	32	32	2	1
	Start Flag	Command Status	Data1	Data2	End Flag	CRC

The BF-1010 returns by Return packet. You can get command status to know the result after sending command packet and from Data1 and Data2 to know current I/O state.

**Note:**

Start Flag: 0xF0F0

End Flag: 0xF0F0

**Command Status:** the definition of command code is as following

0x0002 – ACK of Read Digital I/O state

0x0004 – ACK of Trigger Digital I/O

0x0006 – ACK of E-mail Alarm Trigger

0x0008 – ACK of Reboot

0x000A – ACK of Reset to System Default

0x000C – ACK of Backup configuration

0x000E – ACK of Restore configuration

0x0010 – Report current Digital I/O state (If Auto Report I/O Status is enabled, you will receive this report packet from BF-1010)

0x0012 – ACK of Set I/O working rule

0x0014 – ACK of Get I/O working rule

0x0016 – ACK of Get I/O triggering count

0xFFFC – Flag error, incorrect Start Flag or End Flag received in command packet

0xFFFD – Length error, the length of command packet is invalid

0xFFFE – CRC error, incorrect CRC value

0xFFFF – Command error, no such command

**CRC value = 0 – total sum from field of ‘Start Flag’ to ‘End Flag’**

The format of each command code is as following:

**1. Read Digital I/O state**

<b>Length (Bytes)</b>	<b>2</b>	<b>2</b>	<b>32</b>	<b>32</b>	<b>2</b>	<b>1</b>
	<b>0xF0F0</b>	<b>0x0001</b>	<b>Xxx (don't care)</b>	<b>Xxx (don't care)</b>	<b>0xF0F0</b>	<b>CRC</b>

**Return Successful Packet**

<b>Length(Bytes)</b>	<b>2</b>	<b>2</b>	<b>32</b>	<b>32</b>	<b>2</b>	<b>1</b>
----------------------	----------	----------	-----------	-----------	----------	----------

	<b>0xF0F0</b>	<b>0x0002</b>	<b>Data1</b>	<b>Data2</b>	<b>0xF0F0</b>	<b>CRC</b>
--	---------------	---------------	--------------	--------------	---------------	------------

### Data1

Data[0]	Data[1]	Data[2]	Data[3]					Data[30]	Data[31]
IN-1	IN-2	IN-3	IN-4					reserved	reserved

IN-1: state of IN1, 0 for SHORT, 1 for OPEN

IN-2: state of IN2, 0 for SHORT, 1 for OPEN

IN-3: state of IN3, 0 for SHORT, 1 for OPEN

|

IN-10: state of IN10, 0 for SHORT, 1 for OPEN

### Data2

Data[0]	Data[1]	Data[2]	Data[3]					Data[30]	Data[31]
OUT-1	OUT-2	OUT-3	OUT-4					reserved	reserved

OUT-1: state of OUT1, 0 for SHORT, 1 for OPEN

OUT-2: state of OUT2, 0 for SHORT, 1 for OPEN

OUT-3: state of OUT3, 0 for SHORT, 1 for OPEN

|

OUT-10: state of OUT10, 0 for SHORT, 1 for OPEN

## 2. Trigger Digital I/O

<b>Length (Bytes)</b>	<b>2</b>	<b>2</b>	<b>32</b>	<b>32</b>	<b>2</b>	<b>1</b>
	<b>0xF0F0</b>	<b>0x0003</b>	<b>Data1</b>	<b>Data2</b>	<b>0xF0F0</b>	<b>CRC</b>

### Data1

Data[0]	Data[1]	Data[2]	Data[3]					Data[30]	Data[31]
OUT-1	OUT-2	OUT-3	OUT-4					reserved	reserved

OUT-1: the value you want to write into OUT1, 0 for SHORT, 1 for OPEN

OUT-2: the value you want to write into OUT2, 0 for SHORT, 1 for OPEN

OUT-3: the value you want to write into OUT3, 0 for SHORT, 1 for OPEN

OUT-10: the value you want to write into OUT10, 0 for SHORT, 1 for OPEN

### Data2

Data[0]	Data[1]	Data[2]	Data[3]				Data[30]	Data[31]
LAT-1	LAT-2	LAT-3	LAT-4				reserved	reserved

LAT-1: Latch time in seconds for Output1

LAT-2: Latch time in seconds for Output2

LAT-3: Latch time in seconds for Output3

LAT-10: Latch time in seconds for Output10

### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0004	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

### 3. E-mail Alarm Trigger

Length (Bytes)	2	2	64	2	1
	0xF0F0	0x0005	Alarm Message Description	0xF0F0	CRC

**Alarm Message Description:** string of alarm message by customer attach and BF-1010 send this content by e-mail

### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0006	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

### 4. Reboot

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0007	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

#### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0008	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

#### 5. Reset System to Default

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0009	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

#### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x000A	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

#### 6. Backup System Configuration

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x000B	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

#### Return Successful Packet

Command Field	Value	Size(Bytes)
Start Flag	0xF0F0	2
command	0x000c	2
Length of Config Data	N	4

Config Data		N
End Flag	0xF0F0	2
Check Sum	CRC	1

### 7. Restore System Configuration

Command Field	Value	Size(Bytes)
Start Flag	0xF0F0	2
command	0x000D	2
Length of Config Data	N	4
Config Data		N
End Flag	0xF0F0	2
Check Sum	CRC	1

### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x000E	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

### 8. Set I/O Working Rule

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0011	Data1	Data2	0xF0F0	CRC

### Data1

Command Field	description	Size (bytes)
Data[0]	Normal State for Input 1 (0/1: High/Low)	1
Data[1]	Output No. relied on Input 2 for energized (0/1/2../10: not used/Out 1/Out2/.../Out10)	1
Data[2]	Normal State for Input 2 (0/1: High/Low)	1
Data[3]	Output No. relied on Input 2 for energized	1

	(0/1/2../10: not used/Out 1/Out2/.../Out10)	
Data[30]	Reserved	1
Data[31]	Reserved	1

### Data2

Data[0-1]	Data[2-3]	Data[4-5]	Data[6-7]					Data[30]	Data[31]
LAT-1(2 bytes)	LAT-2	LAT-3	LAT-4					reserved	reserved

LAT-1: Latch time in seconds for Input 1.

LAT-2: Latch time in seconds for Input2

LAT-3: Latch time in seconds for Input3

LAT-10: Latch time in seconds for Input10

### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0012	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

### 9. Get I/O Working Rule

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0013	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0014	Data1	Data2	0xF0F0	CRC

### Data1

Command Field	description	Size (bytes)
Data[0]	Normal State for Input 1 (0/1: High/Low)	1
Data[1]	Output No. relied on Input 2 for energized (0/1/2../10: not used/Out 1/Out2/.../Out10)	1
Data[2]	Normal State for Input 2 (0/1: High/Low)	1
Data[3]	Output No. relied on Input 2 for energized (0/1/2../10: not used/Out 1/Out2/.../Out10)	1
Data[30]	Reserved	1
Data[31]	Reserved	1

### Data2

Data[0-1]	Data[2-3]	Data[4-5]	Data[6-7]					Data[30]	Data[31]
LAT-1 (2 bytes)	LAT-2	LAT-3	LAT-4					reserved	reserved

LAT-1: Latch time in seconds for Input 1.

LAT-2: Latch time in seconds for Input 2

LAT-3: Latch time in seconds for Input 3

|

LAT-10: Latch time in seconds for Input 10

### 10. Get I/O Triggering Count

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0015	Xxx (don't care)	Xxx (don't care)	0xF0F0	CRC

### Return Successful Packet

Length (Bytes)	2	2	32	32	2	1
	0xF0F0	0x0014	Data1	Data2	0xF0F0	CRC



**Data1**

Data[0-1]	Data[2-3]	Data[4-5]	Data[6-7]					Data[18-19]	Data[20-31]
Triggering count for input 1	Triggering count for input 2	Triggering count for input 3	Triggering count for input 4					Triggering count for input 10	reserved

2 bytes for each input triggering count

**Data2**

Data[0-1]	Data[2-3]	Data[4-5]	Data[6-7]					Data[18-19]	Data[20-31]
Triggering count for output 1	Triggering count for output 2	Triggering count for output 3	Triggering count for output 4					Triggering count for output 10	reserved

2 bytes for each output triggering count