Brillian CIDRW

Enhancement RFID Reader

MANUAL

(Revision V1.7.3)

Brillian Network & Automation Integrated System Co., Ltd
Add: Add: No.41, Keyi St., Zhunan Township, Miaoli County 350, Taiwan (R.O.C.),
Tel: 886-37-580708
Fax: 886-37-580728
http://www.brillian.com.tw

CATALOG

1 Introduction

1.1 Brief introduction of equipment

The Brillian CIDRW(Carrier ID Reader/Writer) , is BRILLIAN's RFID(Radio Frequency Identification) system.

The way of basic setup:

Put a Transponder(Tag) fixed on the goods which you need to get goods information. The Transponder has a data record for the goods'.

During the Host computer sent a order message to the CIDRW through RS232 , The CIDRW will sent back the goods information which read from the Transponder by radio frequency.

Brillian CIDRW basic form :

Type	AutoID 2000
Normal working voltage	DC15V
Reading/Writing impulse Rod Antenna current	170mA
Working frequency	134.2k Hz
Communication spec.(RS232)	9600,n,8,1
Housing material	ABS-UL94V0

Un-license band: This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Warning: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user authority to operate the equipment.

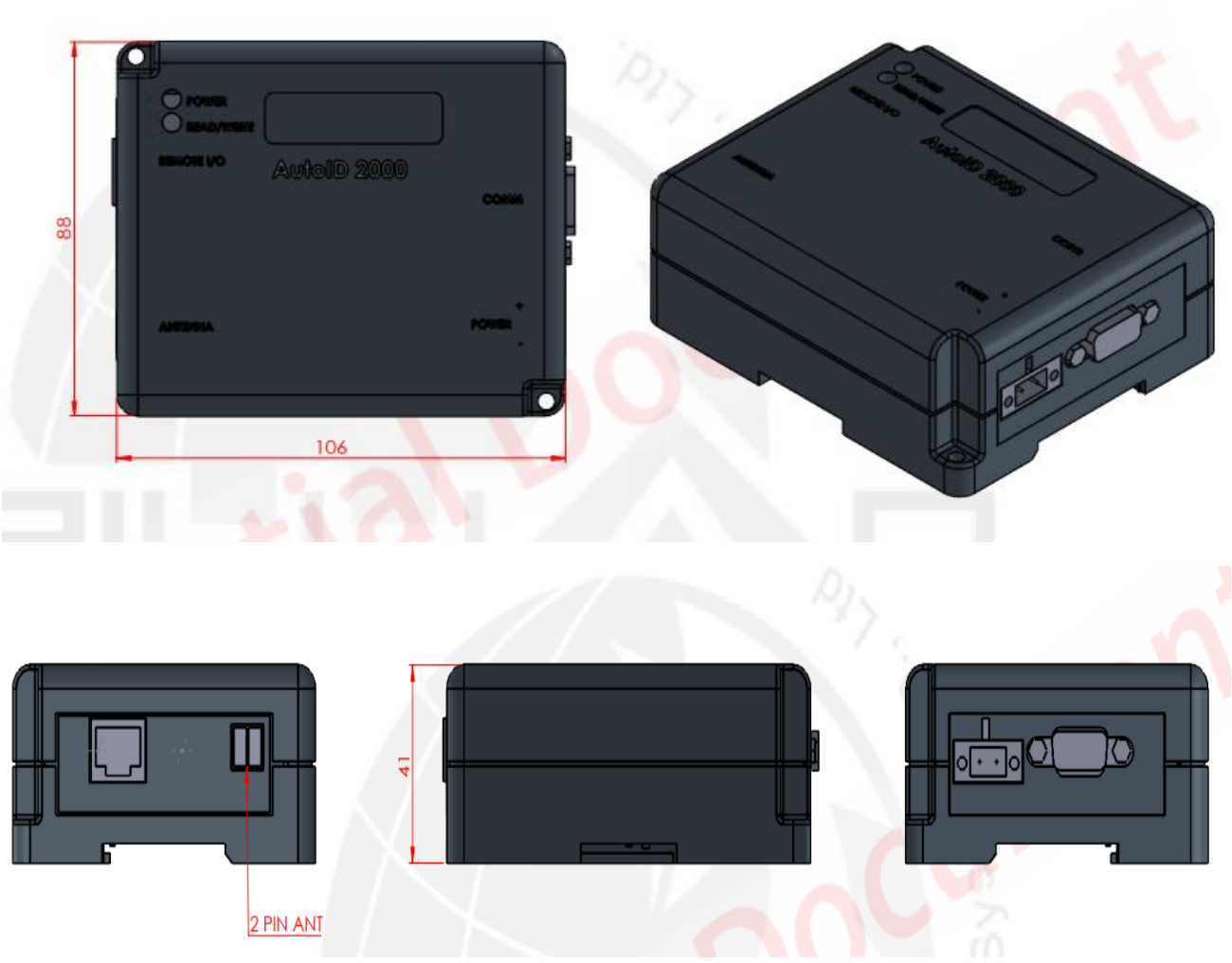
This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications.

However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- ◆ Reorient or relocate the receiving antenna.
- ◆ Increase the separation between the equipment and receiver.
- ◆ Connect the equipment into an outlet on a circuit different from that to which the receiver is needed.
- ◆ Consult the dealer or an experienced radio/TV technician for help.

1.2 Dominations of the equipment



Power：DC power input (15Vdc).
RS232：DB9 female

Antenna：Dipole Antenna
Power-LED：Normal power light.
Read/Write-LED：Read Transponder data normal light。
Remote IO：RJ45
test button – Selftest Button to read Transponder(Tag).

2 Communication protocol

The communication protocol follow the "SEMI Standard E99.

2.1 The way of Communication

Useing RS232 communication protocol , the Host computer can communicate with CIDRW, Reader.

Wiring:

PC	→	**CIDRW (DB9)**	→	**CIDRW (RJ45)**
2(RXD)	→	2(TXD)	→	5(TXD)
3(TXD)	→	3(RXD)	→	6(RXD)
5(GND)	→	5 (GND)	→	4(GND)

2.2 The Communication spec.

Baud Rate : 9600bps.
Data bits: 8
Parity: none
Stop bit: 1

2.3 Handshake code

When the communication starting the **1st** code is ENQ form source equipment to destination equipment for asking destination equipment to receive SECS function string

Then destination equipment will send the **2nd** code EOT for ready status.

3th the source equipment start to send **Message Block(2.4**).

Finally the destination equipment send ACK/Nak to feedback data record good or bad.

Both side use the same way to exchange their function and data.

Name	HEX	Function
ENQ	0x05	Request to Send
EOT	0x04	Ready to Receive
ACK	0x06	Correct Reception
NAK	0x15	Incorrect Reception

2.4 Message Block Form

The maximum length block has 1 byte Length Byte，10 byte Header，and 244 byte Data，and 2 byte Checksum。

name	numerical		mean	seat
Length Byte	10~254		Data lenght	0
Upper Device ID	R	0~127	Device ID	1
Lower Device ID	0~256		TCP ID	2
Upper Message ID	W	0~127	Stream No.	3
Lower Message ID	0~255		Function No.	4
Upper Block No.	E	0~127	Block No.	5
Lower Block No.	1~255		Block No.	6
System Bytes	0~255		System contrast byte 1	7
System Bytes	0~255		System contrast byte 2	8
System Bytes	0~255		System contrast byte 3	9
System Bytes	0~255		System contrast byte 4	10
Data	0~255		data	11~254
Upper Checksum	0~255		SUM check from	255
Lower Checksum	0~255		1byteto 254byte	256

2.4.1 Data record length

One byte，for mean the length of data record ，the value is 0~254 not including the itself and Checksum (2 bytes)，

2.4.2 Data Header

10 bytes for Header including (Device ID), (Message ID)，(Block No.)，and (System Bytes).

Device ID：First byte is R bit： R=0 Host→Equipment
R=1 Equipment→Host。
Message ID：First byte is W bit： W=0 don’t need reply;
W=1 need reply。
Block No.：First byte is E bit： E=0 still have Block E=1
is the last Block。

System byte：Message number, normally increment by transmitter and the receiver reply the same number.

2.4.3 Checksum

Summary the datas，2 bytes length。

2.5 The waiting time and the retry times

2.5.1 Inter Character Timeout

Byte upper limitation for transmit delay time.

2.5.2 Protocol Timeout

Sent the ENQ then waiting the reply EOT timeout.

2.5.3 Reply Timeout

Sent Primary Message，wait respond Secondary Message.

2.5.4 Inter-Block Timeout

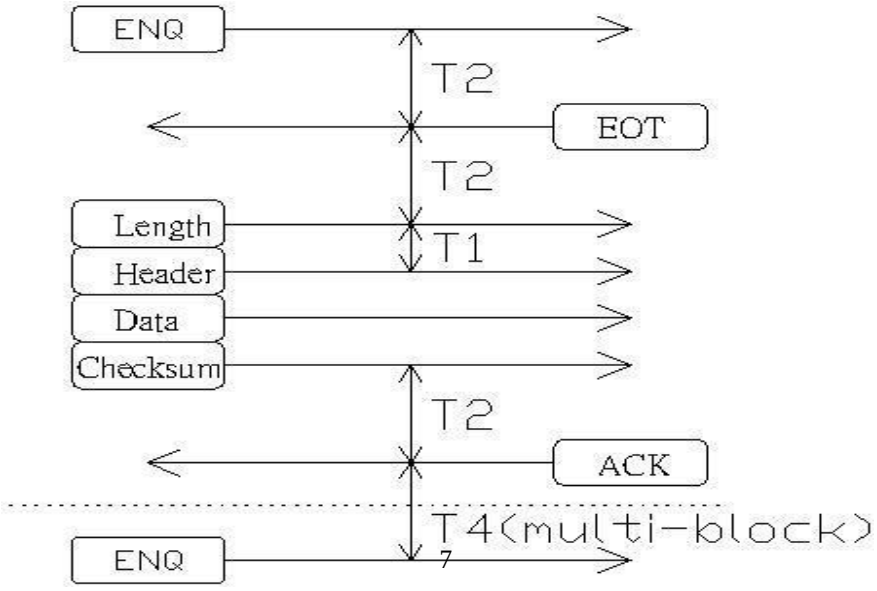
On Multi-Block record，the upper limitation delay for next block.

2.5.5 Retry Limit

When timeout the retry times.

T	Timeout	SEMI	BR2005A
T1	Inter Character	0.5(sec)	0.5(sec)
T2	Protocol Timeout	10(sec)	10(sec)
T3	Reply Timeout	45(sec)	45(sec)
T4	Inter-Block Timeout	45(sec)	45(sec)
RTY	Retry Limit	3(time)	3(time)

Source： Destination：



3 CIDRW read & write code

MID=CarrierID=CID
MID Define on SEMI E5
CID Define on SEMI E99

3.1 MID(CID) code

To read MID (Page1~Page2)，16byte data =>S18F9 To
write MID (Page1~Page2)，16byte data，=>S18F11

* If the length of writing data less than 16 bytes，the rest space will filled “0” automatically by the reader.

3.2 NOTEPAD code

To read or write TI Multi-page data (P1~P17) =>S18F5/ S18F7

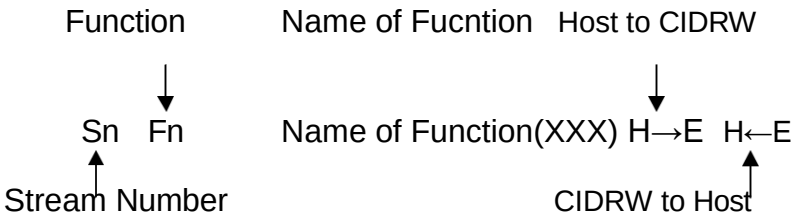
Page mode:

1page，(1~8 bytes)

Offset mode:

120 bytes data could be read/ write at one time (p3-p17).

4 Function explanation



4.1 S1F1 Are you there request(R) H→E

Data structure：
<Header>

explanation：Are you there ?

S1F2 On Line Data(D) H←E

Data structure：
L,2
<MDLN> =>CIDRW model number
<SOFTREV> =>CIDRW software date

explanation：this message is for asking the CIDRW and get the model number and software version

S1F1 example：

16:40:39.307: Send ENQ. =>the Host computer sent ENQ(05)
16:40:39.327: Receive EOT. => the Host computer get EOT(04)
16:40:39.327: Send a Block.(81071296) => the Host computer sent data
0A =>Length data
00 03 81 01 80 01 04 D5 0C C0 =>Header
02 AB =>Checksum
16:40:39.367: Receive ACK. =>the Host computer get data.
currect answer ACK(06) 16:40:39.367:
Send S1F1(81071296). S1F1 W
.

S1F2 example：

16:40:39.367: Receive ENQ. => the Host computer get ENQ
16:40:39.367: Send EOT. => the Host computer sent EOT
16:40:39.408: Received a Block (81071296). => the Host computer get data
1A =>Length

```
81 FF 01 02 80 01 04 D5 0C C0          =>Header
01 02                                  =>List 2 data
      41 06 42 52 39 31 30 30          =>Data “BR9100”
      41 04 56 31 2E 30                =>Data  “V1.0”
06 7B                                  =>Checksum
16:40:39.408: Send ACK.                => the Host computer sent EOT
16:40:39.408: Receive S1F2(81071296).
S1F2
<L[2]
  <A[6] "BR9100">    =>CIDRW model type
  <A[4] "V1.0">      =>CIDRW software data
>
```

.

4.2 S18F1 Read Attribute Request(RAR) H→E

Data structure：

```
L,2
    <TargetID>
    L,n
        <ATTRID1>        => argument 1
        …
        <ATTRIDn>        => argument n
```

explanation： the Host computer ask for **Attribute Data** from CIDRW .

S18F2 Read Attribute Data(RAD) H←E

data structure：

```
L,2
    <TargetID>
    <SSACK> =>CIDRW answer code L,n

        1.  <ATTRVAL1>
        …

    <TargetID>
    L,n
        L,2
            <A      n.  <ATTVALn>
    L,4
        <STATUS1>   =>PM Information
        <STATUS2>   =>Alarm Status
        <STATUS3>   =>Operational  Status
        <STATUS4>   =>Head Status
```

explanation：CIDRW replay the message for the Host computer。

SSACK：2byte reply code.

“NO”-Normal Operation

"CE"-Communication Error
"TE"-Tag Error
"HE"-Hardware Error
"EE"-Execution Error
"EN"-MID write protected
"DI"–MID write protection disable (MID could be write)

PS. When S18F1 argument List is "0" (L,0) , will reply ten argument
Configuration argument value : 01-Number of heads
Alarm Status value : 0-ok;1-alarm
Operation Status "IDLE" Reader
"BUSY" Reader
"NOOP" Reader
Software Revision
Model Number
Head ID Reader ID Preserving value is "00"
Hardware Revision
Manufacturer
Baud rate
MID Length

Attribute Data	Type	Length	Data	Remark
Configuration	A	2	'01'	*
Alarm Status	A	1	'0'	*
Operational Status	A	4	'IDLE'	*
00000000000000000000000000000000000000	A	4	'1.03'	*
Model Number	A	7	'BR2005A'	*
Head ID	A	2	'00'	*
Hardware Revision Level	A	f	'EP-302111 Ver:B'	*
Manufacturer	A	8	'BRILLIAN'	*
BAUDRATE	A	1	'2'	
Carrier ID Length	A	2	'16'	
Carrier ID Offset	A	1	'0'	
CID_MAX_LENGTH	A	2	'16'	
MANTWRITEONLY	A	2	'DI'	

* for BR2005B only

S18F1 example(Read All Attribute) :

```
11:47:25.109: Send ENQ.
11:47:25.140: Receive EOT.
11:47:25.140: Send a Block.(10960740)
12 00 00 92 01 80 01 00 A7 3F    64 01 02 41 02 30 31 01 00 03
06
11:47:25.171: Receive ACK.
11:47:25.171: Send S18F1(10960740).
S18F1 W
<L[2]
   <A[2] "01">
   <L[0]
   >
>
.
```

S18F2 example：

```
11:47:25.171: Receive ENQ.
11:47:25.171: Send EOT.
11:47:25.328: Received a Block (10960740).
78 80 00 12 02  80 01 00 A7 3F     64  01 04 41 02 30 30 41 02 4E
4F 01 0C 41 01 31 41 01 30 41    04 49 44 4C 45  41 04 31 2E 35
36 41 07 42 52 39 32 30 30 77    41 08 42 52 49 4C 4C 49 41 4E
41 04 49 44 4C 45 41 02 30 30    41 0F 45 50 2D  33 30 32 31 31
31 20 56 65 72 3A  42 41 04 39    36 30 30   41 02 31 36 41 01 30
01 04 41 02 4E 45  41 01 30 41    04 49 44 4C 45 41 04 49 44 4C
45 18 4B
11:47:25.328: Send ACK.
11:47:25.328: Receive S18F2(10960740).
S18F2
<L[4]
   <A[2] "00">
   <A[2] "NO">
   <L[12]
      <A[1] "1">
      <A[1] "0">
      <A[4] "IDLE">
      <A[4] "1.56">
      <A[7] "BR9200w">
      <A[8] "BRILLIAN">
      <A[4] "IDLE">
      <A[2] "00">
      <A[15] "EP-302111 Ver:B">
      <A[4] "9600">
      <A[2] "16">
      <A[1] "0">
   >
   <L[4]
      <A[2] "NE">
      <A[1] "0">
      <A[4] "IDLE">
      <A[4] "IDLE">
   >
>
```

.

4.3 S18F3 Write Attribute Request(WAR) H→E

data structure：

L2
  L,2
    <A TTRID1> => argument name 1
    <ATTRVAL1> => argument value 1
  L,2
    <ATTRIDn> => argument name n
    <ATTRVALn> => argument value n

explanation：This message is the Host computer ask the CIDRW to change the argument's value。

S18F4 Write Attribute Acknowledge(WAA) H←E

data structure：

L,3
  <TargetID>
  <SSACK>
  L,s
    <STATUS1>
    …
    <STATUSn>

S18F3 example：

```
08:29:18.816: Send ENQ.
08:29:18.836: Receive EOT.
08:29:18.856: Send a Block.(256518180)
 21 00 03 92 03 80 01 0F 4A 28    24 01 02 41 02 30 33 01 01 01
 02 41 08 42 41 55 44 52 41 54      45 41 01 32 05 71
08:29:18.896: Receive ACK.
08:29:18.896: Send S18F3(256518180).
S18F3 W
<L[2]
    <A[2] "03">
    <L[1]
        <L[2]
        <A[8] "BAUDRATE">
        <A[1] "2">
    >
>
```

.

S18F4 example：

```
08:29:18.896: Receive ENQ.
08:29:18.896: Send EOT.
08:29:18.957: Received a Block (256518180).
19 80 00 12 04 80 01 0F 4A 28    24 01 03 41 02 30 30 41 02 4E
4F 01 01 41 01 32 03 B9
```

```
08:29:18.957: Send ACK.
08:29:18.967: Receive S18F4(256518180).
S18F4
<L[3]
      <A[2] "00">
      <A[2] "NO">
      <L[1]
            <A[1] "2">
      >
>
```

.

4.4 S18F5 Read Data Request(RDR) H→E

data structure：

L,3

<TagetID>
<DATASEG>
<DATALENGTH>

explanation：

DATASEG (P1-P17 for page mode)/(0-119 for offset mode)
DATALENGTH (1-8 for page mode)/(1-120 for offset mode)
* The Host computer sent this message , it could choice to read page1-page17 per page (8 bytes) or 1-120 bytes per record one time data as need.
* Offset Mode only can read tag data from Page 3 to Page 17.

S18F6 Read Data Ack(RDA) H←E

data structure：

L,3

<TargetID>
<SSACK>
<DATA> =>reply value，if read data error then reply an empty alphabetic string。

explanation：Host computer need to choice the page mode or offset mode .

S18F5 example page mode(Read Page 1)：

```
10:55:00.062: Send ENQ.
10:55:00.093: Receive EOT.
10:55:00.093: Send a Block.(8542382)
18 00 00 92 05 80 01 00 82 58    AE 01 03 41 02 30 31 41 03 50
30 31 41 01 38 04 B7
10:55:00.125: Receive ACK.
10:55:00.125: Send S18F5(8542382).
S18F5 W
<L[3]
  <A[2] "01">
  <A[3] "P01">
```

```
   <A[1] "8">
>
.
```

S18F6 example page mode：

```
10:55:00.234: Receive ENQ.
10:55:00.234: Send EOT.
10:55:00.421: Received a Block (8542382).
33 80 00 12 06 80 01 00 82 58     AE 01 04 41 02 30 31 41 02 4E
4F 41 08 4C 4F 54 3A 34 35 36     00 01 04 41 02 4E 45 41 01 30
41 04 49 44 4C 45 41 04 49 44     4C 45 0A 4E
10:55:00.421: Send ACK.
10:55:00.421: Receive S18F6(8542382).
S18F6
<L[4]
   <A[2] "01">
   <A[2] "NO">
   <A[8] "LOT:456">
   <L[4]
      <A[2] "NE">
      <A[1] "0">
      <A[4] "IDLE">
      <A[4] "IDLE">
   >
>
.
```

S18F5 example example offset mode(Offset=0 Length=16)：

```
11:02:06.359: Send ENQ.
11:02:06.390: Receive EOT.
11:02:06.390: Send a Block.(8542383)
17 00 00 92 05 80 01 00 82 58     AF 01 03 41 02 30 31 41 01 30
41 02 31 36 04 65
11:02:06.421: Receive ACK.
11:02:06.421: Send S18F5(8542383).
S18F5 W
<L[3]
   <A[2] "01">
   <A[1] "0">
   <A[2] "16">
>
.
```

S18F6 example example offset mode：

```
11:02:06.625: Receive ENQ.
11:02:06.625: Send EOT.
11:02:06.937: Received a Block (8542383).
3B 80 00 12 06 80 01 00 82 58     AF 01 04 41 02 30 31 41 02 4E
4F 41 10 43 43 43 43 43 43 43     43 44 44 44 44 44 44 44 44 01
04 41 02 4E 45 41 01 30 41 04     49 44 4C 45 41 04 49 44 4C 45
0C C7
11:02:06.937: Send ACK.
11:02:06.937: Receive S18F6(8542383).
S18F6
<L[4]
```

```
  <A[2] "01">
  <A[2] "NO">
  <A[16] "CCCCCCCCDDDDDDDD">
  <L[4]
    <A[2] "NE">
    <A[1] "0">
    <A[4] "IDLE">
    <A[4] "IDLE">
  >
>
```

.

4.5 S18F7 Write Data Request(WDR) H→E

data structure：

L,4

<TargetID>
<DATASEG>
<DATALENGTH>
<DATA>

explanation：

DATASEG (P1-P17 for page mode)/(0-119 for offset mode)
DATALENGTH (1-8 for page mode)/(1-120 for offset mode)
* The Host computer sent this message , it could choice to write page1-page17 per page (8 bytes) or 1-120 bytes per record one time data as need.
* Offset Mode only can write tag data from Page 3 to Page 17.

S18F8 Write Data Acknowledge(WDA) H←E

data structure：

L,3

<TargetID>
<SSACK>
L,4
<STATUS1>
…
<STATUS4>

explanation：this message is reply S18F7，the target page and the target lenght data been wrote ok or fail .

S18F7 example page mode(write Page1 with 8 bytes)：

```
11:55:09.562: Send ENQ.
11:55:09.593: Receive EOT.
11:55:09.593: Send a Block.(10960741)
22 00 00 92 07 80 01 00 A7 3F    65 01 04 41 02 30 31 41 03 50
30 31 41 01 38 41 08 4E 46 42    31 32 33 34 35 06 9B
11:55:09.656: Receive ACK.
11:55:09.656: Send S18F7(10960741).
```

```
S18F7 W
<L[4]
   <A[2] "01">
   <A[3] "P01">
   <A[1] "8">
   <A[8] "NFB12345">
>
.
```

S18F8 example page mode

```
11:55:10.078: Receive ENQ.
11:55:10.078: Send EOT.
11:55:10.140: Received a Block (10960741).
29 80 00 12 08 80 01 00 A7 3F     65 01 03 4102 30 31 41 02 4E
4F 01 04 41 02 4E 45 41 01 30     41 04 49 444C 45 41 04 49 44
4C 45 08 01
11:55:10.140: Send ACK.
11:55:10.140: Receive S18F8(10960741).
S18F8
<L[3]
   <A[2] "01">
   <A[2] "NO">
   <L[4]
      <A[2] "NE">
      <A[1] "0">
      <A[4] "IDLE">
      <A[4] "IDLE">
   >
>
.
```

S18F7 example offset mode(offset=0 data length=12)：

```
12:00:30.796: Send ENQ.
12:00:30.828: Receive EOT.
12:00:30.828: Send a Block.(10960747)
25 00 00 92 07 80 01 00 A7 3F     6B 01 04 41 02 30  31 41 01 30
41 02 31 32 41 0C 32 30 31 31     2F 31 30 2F 31 31  2F 30 06 BD
12:00:30.890: Receive ACK.
12:00:30.890: Send S18F7(10960747).
S18F7 W
<L[4]
   <A[2] "01">
   <A[1] "0">
   <A[2] "12">
   <A[12] "2011/10/11/0">
>
.
```

S18F8 example offset mode

```
12:00:31.859: Receive ENQ.
12:00:31.859: Send EOT.
12:00:31.921: Received a Block (10960747).
29 80 00 12 08 80 01 00 A7 3F     6B 01 03 41 02 30 31 41 02 4E
4F 01 04 41 02 4E 45 41 01 30     41 04 49 44 4C 45 41 04 49 44
```

4C 45 08 07
12:00:31.921: Send ACK.
12:00:31.921: Receive S18F8(10960747).
S18F8
<L[3]
  <A[2] "01">
  <A[2] "NO">
  <L[4]
    <A[2] "NE">
    <A[1] "0">
    <A[4] "IDLE">
    <A[4] "IDLE">
  >
>
.

4.6 S18F9 Read Material ID Request(RMID) H→E

data structure：
<TargetID>

explanation： Host computer request for MID from CIDRW, The MID data is read from pag1 and page2 of Tag，The total length of MID is 16 bytes.

S18F10 Material ID Data(MID) H←E

data structure：
L,4
<TargetID>
<SSACK>
<MID> =>reply the value of read MID data，
when read fail reply empty alphabetic string。
L,4
<STATUS1>
…
<STATUS4>

explanation：CIDRW reply the Host compute S18F9 command and reply the data of MID。

S18F9 example：
13:45:56.328: Send ENQ.
13:45:56.359: Receive EOT.
13:45:56.359: Send a Block.(10960751)
0E 00 00 92 09 80 01 00 A7 3F 6F 41 02 30 31 03 15
13:45:56.390: Receive ACK.
13:45:56.390: Send S18F9(10960751).
S18F9 W

<A[2] "01">
.

S18F10 example：

```
13:45:56.593: Receive ENQ.
13:45:56.593: Send EOT.
13:45:56.687: Received a Block (10960751).
 34 80 00 12 0A 80 01 00 A7 3F     6F 01 04 41 02 30 31 41 02 4E
4F 41 09 4E 46 46   30 30 35 30    33 32 01 04 41 02 4E 45 41 01
 30 41 04 49 44 4C   45 41 04 49    44 4C 45 0A 5C
13:45:56.687: Send ACK.
13:45:56.687: Receive S18F10(10960751).
S18F10
<L[4]
   <A[2] "01">
   <A[2] "NO">
   <A[9] "NFF005032">
   <L[4]
      <A[2] "NE">
      <A[1] "0">
      <A[4] "IDLE">
      <A[4] "IDLE">
   >
>
.
```

4.7 S18F11 Write Material ID Request(WMID) H→E

data structure：

```
L,2
    <TargetID>
    <MID>                =>Exchange MID to this value
```

explanation： Host computer write MID to Tag，The total length of MID is 16 bytes.

S18F12 Write Material ID Acknowledge(WMIDA) H←E

data structure：

```
L,3
    <TargetID>
    <SSACK>
    L,4
        <STATUS1>
        …
        <STATUS4>
```

explanation：

This message is reply S18F11，MID write Tag ok or fail .

S18F11 example：

```
12:04:23.937: Send ENQ.
12:04:23.968: Receive EOT.
12:04:23.968: Send a Block.(10960750)
1B 00 00 92 0B 80 01 00 A7 3F    6E 01 02 41 02 30 31 41 09 4E
46 46 30 30 35 30 33 32 05 67
12:04:24.031: Receive ACK.
12:04:24.031: Send S18F11(10960750).
S18F11 W
<L[2]
  <A[2] "01">
  <A[9] "NFF005032">
>
.
```

S18F12 example：

```
12:04:24.875: Receive ENQ.
12:04:24.890: Send EOT.
12:04:24.953: Received a Block (10960750).
29 80 00 12 0C 80 01 00 A7 3F     6E 01 03 41 02 30 31 41 02 4E
4F 01 04 41 02 4E 45 41 01 30     41 04 49 44 4C 45 41 04 49 44
4C 45 08 0E
12:04:24.953: Send ACK.
12:04:24.953: Receive S18F12(10960750).
S18F12
<L[3]
  <A[2] "01">
  <A[2] "NO">
  <L[4]
    <A[2] "NE">
    <A[1] "0">
    <A[4] "IDLE">
    <A[4] "IDLE">
  >
>
.
```

4.8 S18F71 Event Report Send(ERS) H←E

data structure：

```
L,4
    <TargetID>
    <SSACK>
    <CEID>                    =>message ID
    L,n
        <DVNAME1>             =>message name
        <DVVAL1>              =>message value
        …
        <DVNAMEn>
        <DVVALn>
```

explanation：This message send from CIDRW，during Remote I/O button71

trigged (signal to ground), is active signal and no reply needed.

S18F71 example

```
01:12:26.272: Receive ENQ.
01:12:26.272: Send EOT.
01:12:26.352: Received a Block (438975051).
39 80 03 12 47 80 01 1A 2A 3A     4B 01 04 41 02 30 33 41 02 4E
4F 41 01 31 01 02 41 0C 41 75     74 6F 52 65 61 64 44 61 74 61
41 10 30 30 30 30 30 30 30 4E     46 42 30 30 31 37 39 34 0C AF
01:12:26.352: Send ACK.
01:12:26.352: Receive S18F71(438975051).
S18F71
<L[4]
      <A[2] "03">
      <A[2] "NO">
      <A[1] "1">
      <L[2]
            <A[12] "AutoReadData">
            <A[16] "0000000NFB001794">       => read ok then reply the data
      >
>
```

4.9 S18F75 Event Report Send(ERS) H←E

data structure：

```
L,4
      <TargetID>
      <SSACK>
      <CEID>                    =>message ID
      L,n
            <DVNAME1>           =>message name
            <DVVAL1>          =>message value
            …
            <DVNAMEn>
            <DVVALn>
```

explanation：This message send from CIDRW，during Remote I/O button75 trigged (signal to ground), is active signal and no reply needed.

S18F75 example

```
01:12:26.272: Receive ENQ.
01:12:26.272: Send EOT.
01:12:26.352: Received a Block (438975051).
39 80 03 12 4B 80 01 1A 2A 3A     4B 01 04 41 02 30 33 41 02 4E
4F 41 01 31 01 02 41 0C 41 75     74 6F 52 65 61 64 44 61 74 61
41 10 30 30 30 30 30 30 30 4E     46 42 30 30 31 37 39 34 0C B3
01:12:26.352: Send ACK.
```

```
01:12:26.352: Receive S18F75(438975051).
S18F75
<L[4]
      <A[2] "03">
      <A[2] "NO">
      <A[1] "1">
      <L[2]
            <A[12] "AutoReadData">
            <A[16] "0000000NFB001794">       => read ok reply the data
      >
>
.
```

explanation：When touch the Remote IO，it means when “FOUP”arrive or “FOUP”leave sent this message。

4.10 S9F3 Unrecognized Stream(USN) H←E

data structure：

<MHEAD> =>only sent the error message of Stream Function

explanation：when get a un-definition Stream，sent this message。

S9F3 example：

```
01:49:29.939: Receive ENQ.
01:49:29.939: Send EOT.
01:49:29.979: Received a Block (438975051).
16 80 03 09 03 80 01 1A 2A 3A    4B 21 0A 00 03 93 01 80 01 00
00 00 00 03 1C
01:49:29.979: Send ACK.
01:49:29.979: Receive S9F3(438975051).
S9F3
<B[10] 0x00 0x03 0x93 0x01 0x80 0x01 0x00 0x00 0x00 0x00>
```

4.11 S9F5 Unrecognized Function type(UFN) H←E

data structure：

<MHEAD> =>only sent the error message of Stream Function

explanation：when get a un-definition Function，sent this message。

S9F5 example：

```
01:52:40.313: Receive ENQ.
01:52:40.313: Send EOT.
01:52:40.353: Received a Block (438975051).
16 80 03 09 05 80 01 1A 2A 3A    4B 21 0A 00 03 81 09 80 01 00
00 00 00 03 14
01:52:40.353: Send ACK.
01:52:40.353: Receive S9F5(438975051).
S9F5
<B[10] 0x00 0x03 0x81 0x09 0x80 0x01 0x00 0x00 0x00 0x00>
```

4.12 S9F7 Illegal Data(IDN) H←E

data structure：

<MHEAD> =>only sent the error message of Stream Function

explanation：when get a illegal data，sent this message。

S9F7 example：

01:58:26.551: Receive ENQ.
01:58:26.551: Send EOT.
01:58:26.591: Received a Block (438975051).
16 80 03 09 07 80 01 1A 2A 3A 4B 21 0A 00 03 81 01 80 01 00
02 00 00 03 10
01:58:26.591: Send ACK.
01:58:26.591: Receive S9F7(438975051).
S9F7
<B[10] 0x00 0x03 0x81 0x01 0x80 0x01 0x00 0x02 0x00 0x00>.

4.13 S18F15 Led control Request(WAR) H→E

data structure：

L3
<TargetID>
<SSCMD> =>'04'
L3
<A Function> =>'0'-Off, '1'-On, '2'-Flash
<U1 Seconds> =>'1-FE'-(1-254) seconds, 'FF'-alwaws on
<A Led_number> =>'1'-Led 1 '2'-Led 2

explanation：this message is the Host computer ask to change the led state。

S18F16 Led control Acknowledge(WAA) H←E

data structure：

L,3
<TargetID>
<SSACK>
L,s
<STATUS1>
…
<STATUSn>

S18F15 example(Led 1 Turn Flash 20 Second)：

14:35:19.687: Send ENQ.
14:35:19.718: Receive EOT.
14:35:19.718: Send a Block.(10960764)
1F 00 00 92 0F 80 01 00 A7 3F 7C 01 03 41 02 30 31 41 02 30

34 01 03 41 01 32 A5 01 14 41 01 31 05 78
14:35:19.781: Receive ACK.
14:35:19.781: Send S18F15(10960764).
S18F15 W
<L[3]
  <A[2] "01">
  <A[2] "04">
  <L[3]
    <A[1] "2">
    <U1[1] 20>
    <A[1] "1">
  >
>
.

S18F16 example：

14:35:19.781: Receive ENQ.
14:35:19.781: Send EOT.
14:35:19.812: Received a Block (10960764).
14 80 00 12 10 80 01 00 A7 3F 7C 01 02 41 02 30 31 41 02 4E
4F 04 0C
14:35:19.812: Send ACK.
14:35:19.812: Receive S18F16(10960764).
S18F16
<L[2]
  <A[2] "01">
  <A[2] "NO">
>
.

S18F15 example(Led 2 Turn On)：

14:37:07.468: Send ENQ.
14:37:07.500: Receive EOT.
14:37:07.500: Send a Block.(10960765)
1F 00 00 92 0F 80 01 00 A7 3F 7D 01 03 41 02 30 31 41 02 30
34 01 03 41 01 31 A5 01 FF 41 01 32 06 64
14:37:07.562: Receive ACK.
14:37:07.562: Send S18F15(10960765).
S18F15 W
<L[3]
  <A[2] "01">
  <A[2] "04">
  <L[3]
    <A[1] "1">
    <U1[1] 255>
    <A[1] "2">
  >
>
.

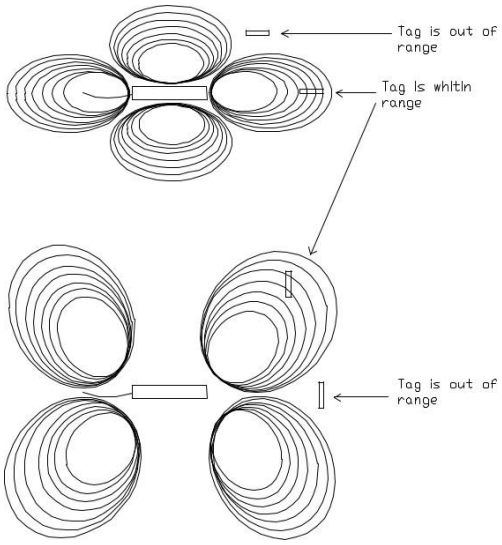
S18F16 example：

14:37:07.562: Receive ENQ.
14:37:07.562: Send EOT.
14:37:07.593: Received a Block (10960765).
14 80 00 12 10 80 01 00 A7 3F 7D 01 02 41 02 30 31 41 02 4E

4F 04 0D
14:37:07.593: Send ACK.
14:37:07.593: Receive S18F16(10960765).
S18F16
<L[2]
  <A[2] "01">
  <A[2] "NO">
>
.

5 Antenna Performance

5.1 Induction range test



5.1.1 read & write range test

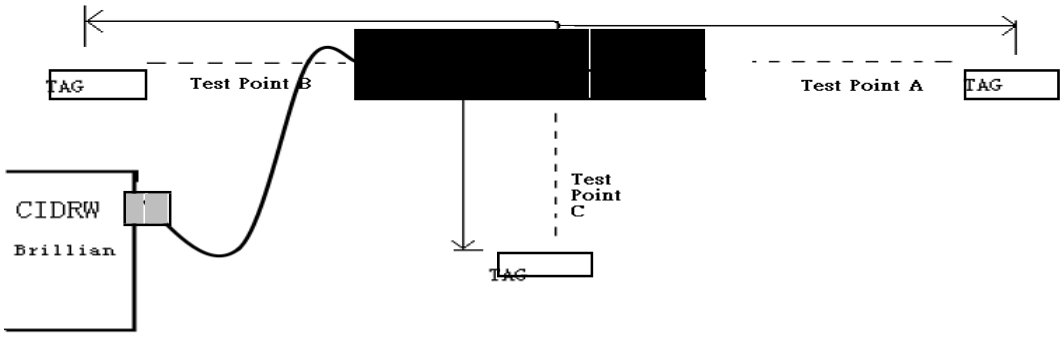
The average value of read/write distance for 100 times.

Average of Read/Write Distance for 100 times (cm)						
	Read			Write		
Tag	A	B	C	A	B	C
1	10.0	10.3	8.5	8.0	8.3	6.5
2	10.0	10.4	8.1	8.0	8.4	6.1
3	9.0	9.5	7.9	7.0	7.5	5.8
4	9.8	10.0	8.2	7.8	8.0	6.2
5	10.0	10.3	8.2	8.0	8.3	6.2
6	9.1	9.1	8.2	7.1	7.1	5.7
7	9.2	9.4	7.8	7.2	7.4	5.6
8	10.0	10.3	8.1	8.5	8.3	6.3

P.S: Above the value is used the same RFID READER for test.

5.2 Read & write continuous test

5.2.1 Put tag on “TAG” (picture1)position
5.2.2 Execute write procedure （S18F11） write data
5.2.3 Execute read p procedure （S18F9）for verification data read
5.2.4 Put Tag on “a” position to write（S18F11）
5.2.5 Execute read p procedure （S18F9）for verification data read
5.2.6 Put Tag on “b” position to write（S18F11）
5.2.7 Execute read p procedure （S18F9）for verification data read
5.2.8 start over from **5.2.4** for 10,000 times without any fault.



【picture 1】

5.3 SAME TAG read & write continuously test

Use the same TAG to read & write and verify continuously

5.2.9 Tag put in（picture1）“TAG” position
5.3.1 Execute write procedure （S18F11）data =”BRILLIAN00000000”
5.3.2 Execute read p procedure （S18F9）for verification data read
5.3.3 Execute write procedure （S18F11）data = ”11111111BRILLIAN”
5.3.4 Execute read p procedure （S18F9）for verification data read
5.3.5 start over from **5.3.3** for 10,000 times read/write without any fault.

6.ASCII Command

6.1 HOST COMMAND FORMAT:

~xx Cmd Data1 Data2 Data3*

Format Decsription

Table 1-1. ASCII host command format

Item	Command	Description
1	~	Start Byte(7EH)
2	xx	Device ID，00 To 08。
3	Cmd	Table 1-2。
4	Data1/Data2/Data3	Parameter string。
5	*	End Byte(2AH)

Host Command List :

Table 1-2. ASCII host command list

Item	Command	Format	Parameter String
1	Are you there	~xx RU*	
2	Read MID	~xx RMID*	
3	Write MID	~xx WMID MID*	MID: max16bytes
4	Read DATA	~xx RDAT DATASEG DATALENGTH*	Offset Mode: DataSEG:000~119 Datalength: 001~120 Page Mode: DataSEG:P01~P17 Datalength:1~8
5	Write DATA	~xx WDAT DATASEG DATALENGTH DATA*	Offset Mode: DataSEG:000~119 Datalength:001~120 Page Mode: DataSEG:P01~P17 DataLength:1~8
6	Write State	~xx WS yy*	yy: 00 operation mode yy: 01 maintenance mode
7	Read State	~xx RS*	
8	Write Event_Report	~xx WEV yyy*	yyy:ASC asc mode yyy:SECS secs mode
9	Read Event Report Mode	~xx REV*	

6.2 READER COMMAND FORMAT:

~xx cmdR rc data*

Format Decsription

Table 1-3. ASCII reader command format

Item	Code	Description
1	~	Start Byte(7EH)
2	xx	Device ID,00 To 08。
3	cmdR	Reader Echo,"R" mean "Reply",Table 1-4
4	rc	SSACK (return code)
5	data	Parameter string。
6	*	End Byte(2AH)

Table 1-4. ASCII reader command list

Item	Command	Format	Parameter String
1	Are you there	~xx RUR softver*	
2	Read MID	~xx RMIDR rc MIDstring*	
3	Write MID	~xx WMIDR rc*	MIDstring: max16bytes
4	Read DATA	~xx RDATR rc datastring*	
5	Write DATA	~xx WDATR rc*	
6	Write State	~xx WSR rc*	
7	Read State	~xx RSR rc*	
8	Wrtie Event Report Mode	~xx WEVR rc*	
9	Read Event Report Mode	~xx REVR ModeType*	ModeType: ASC/SECS
10	Unknow command	~xx UNCMD*	

Table 1-5. Return Code (SSACK)

Item	Name	Description
1	**NO**	Normal Operation
2	**CE**	Communication Error,could not communicate to the Tag
3	**24**	CRC error in Tag data
4	**TE**	Tag Error

5	**SM**	System state Mismatch
6	**DE**	Data Error.
7	**LE**	Invalid Data Length

6.3 ASCII Example

6.3.1 Are you There example:

Host->Reader
~01 RU*
Host<-Reader
~01 RUR Brillian BR920w V1.56*

6.3.2 Write State(set CIDRW to maintenance mode) example:

*CIDRW must be in maintenance mode in order to write
Host->Reader
~01 WS 01*
Host<-Reader
~01 WSR NO*

6.3.3 Read MID Success example:

Host->Reader
~01 RMID*
Host<-Reader
~01 RMIDR NO NFF005032*

6.3.4 Read MID Fail example:

Host->Reader
~01 RMID*
Host<-Reader
~01 RMIDR TE*

6.3.5 Write MID Success example:

Host->Reader
~01 WMID NFB41482*
Host<-Reader
~01 WMIDR NO*

6.3.6 Write MID Fail example:

Host->Reader
~01 WMID NFB41482*
Host<-Reader
~01 WMIDR TE*

Document Title: Brillian CIDRW Enhancement RFID Reader MANUAL

Document No: Revision V1.73

Issuing Department: R&D

Document Revision Details:

Revisions	***Released By***	***Date***	***Description***
1.73	HL.Yeh	2011/07/12	Modified Sample Code.