Braille Printer [Dot] Protocol V1.6

- Control command

all commands start from PC to MBED and always have reply.

each conmand command consists of 3 parts: frame header + data + frame tail, as specified below

	description	data lenth	remark
frame command	Header byte	1	frame header recognition, 0x02, <STX>
	Command byte	1	CMD
	Lenth byte	1	LEN
frame data	Data	0~21	description data lenth 0~21
	Data check byte	1	data check sum
frame command	Tail byte	1	frame tail recognition, 0x03, <ETX>

- Command Byte

command	byte
start print	0x01
whoami	0x03

- Start Print command

idx	description	data	remark
0	start sequence byte	0x02	<STX>
1	command byte	0x01	CMD
2	lenth byte	0x15	21
3	dot data1 high byte	=*(uint8_t *)(&dot data1)	
4	dot data1	=*((uint8_t *)(&dot data1)+1)	
5	dot data1	=*((uint8_t *)(&dot data1)+2)	
6	dot data1	=*((uint8_t *)(&dot data1)+3)	
7	dot data1	=*((uint8_t *)(&dot data1)+4)	
8	dot data1	=*((uint8_t *)(&dot data1)+5)	
9	dot data1	=*((uint8_t *)(&dot data1)+6)	
10	dot data1 low byte	=*((uint8_t *)(&dot data1)+7)	
11	dot data2 high byte	=*(uint8_t *)(&dot data2)	
12	dot data2	=*((uint8_t *)(&dot data2)+1)	
13	dot data2	=*((uint8_t *)(&dot data2)+2)	
14	dot data2	=*((uint8_t *)(&dot data2)+3)	
15	dot data2	=*((uint8_t *)(&dot data2)+4	
16	dot data2	=*((uint8_t *)(&dot data2)+5)	
17	dot data2	=*((uint8_t *)(&dot data2)+6)	
18	dot data2 low byte	=*((uint8_t *)(&dot data2)+7)	
19	dot data3 high byte	=*(uint8_t *)(&dot data3)	
20	dot data3	=*((uint8_t *)(&dot data3)+1)	
21	dot data3	=*((uint8_t *)(&dot data3)+2)	
22	dot data3	=*((uint8_t *)(&dot data3)+3)	
23	dot data3	=*((uint8_t *)(&dot data3)+4)	
24	dot data3	=*((uint8_t *)(&dot data3)+5)	
25	dot data3	=*((uint8_t *)(&dot data3)+6)	
26	dot data3 low byte	=*((uint8_t *)(&dot data3)+7)	
27	data check byte	3~26 byte check sum	check sum
28	end sequnce byte	0x03	<ETX>

MBED reply

[case ①]

idx	description	data	remark
0	communication succeed	0x06	<ACK>

[case ②]

idx	description	data	remark
0	communication failed	0x15	<NAK>

[case ③]

idx	description	data	remark
0	line complete	0x19	<EM>

- whoami command

idx	description	data	remark
0	start sequence byte	0x02	<STX>
1	command byte	0x03	CMD
2	lenth byte	0x00	0
3	data check byte	0xFF	255
4	end sequnce byte	0x03	<ETX>

MBED reply

idx	description	data	remark
0	communication succeed	0x06	<ACK>

- Check Sum Algorithm

통신 데이터 예

```
:01 0E 214601360121470136007EFE09D2 40:
:01 0E 2146017EB7C20001FF5F16002148 88:
:01 0E 194E79234623965778239EDA3F01 A7:
:01 0E 3F0156702B5E712B722B73214601 C7:
:02 00:
```

🟨 Start code
🟩 Command
🟪 Length
🟦 Data
🟥 Checksum
🟧 End code

```
:01 01 214601360121470136007EFE09D2 40:
```

여기서 데이터 바이트들만 더하면

```
    21+46+01+36+01+21+47+01+36+00+7E+FE+09+D2 = 917 = 0x395
```

여기서 1의 보수를 취하면

```
    ~917 = 0xFFFFFC6A
```

여기서 1 Byte만 취하면

```
    0xFFFFFC6A & 0XFF = 0x6A <- Check Sum byte
```