CL5000J Communication Protocol

2007/10/05
Rev 1.0

CAS

1. Basic Protocol Structure

* All protocol number data's byte order is little endian
* Scale ID : default is 1
* INT32U = integer, 32 bit, unsigned
* INT16U = integer, 16 bit, unsigned
* INT8U = interger, 8 bit, unsigned
* INT32S = integer, 32 bit, signed
* CR = 0x0D

Header								Body							Tail	
Opcode (2 Bytes)		Address (4 Bytes)				,	Data length (2 Bytes)	:	Data (Max. 512 Bytes)					:	Checksum (1 Byte)	CR (1 Byte)
Opcode[0]	Opcode[1]															
		Checksum range (Address[0] + ··· + ':')														
		78	56	34	12	2C	04	00	3A	41	42	43	44		3A	88	
Checksum Ex.) Address = 0x12345678, Data length = 4, Data = "ABCD" checksum = (0x78+0x56+0x34+0x12+0x2C+0x04+0x00+0x3A+0x41+0x42+0x43+0x44+0x3A) % 0x100																

Opcode[0]	Description
W	Write
R	Read
G	Good
N	No good

Opcode[1]	Description
A	Label
B	Barcode
G	Sale message
J	Bitmap (Picture)
K	Speed key
L	PLU
M	Indirect message
N	Status
O	Origin
S	Shop
T	Tare (TLU)
U	Report (Account)
V	Version
Y	PCS (Quantity symbol)

Error No.	Description
99	Data end
98	Data isn't exist
97	Data struct fail
95	Unknown data type
89	memory full
88	direct message full
84	header, tail or CK fail
82	number is over

2. PLU Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	L	0							

* DeptPLU Number : Department Number * 1000000 + PLU number
  Ex) Dept 10, PLU 999999 -> 10999999
* Scale ID : default is 1

1. Download PLU without direct message

* Error number 89 : PLU memory full

PC	W	L	0	,	147	:	PLU struct(147)	:	CK	CR
Scale	G	L	Scale ID	,	4	:	DeptPLU number(4)	:	CK	CR
	N	L	Scale ID	,	5	:	DeptPLU number(4) + Error number(1)	:	CK	CR

2. Download PLU with direct message

* If direct Message's last char must be null.
* Error number 89 : PLU memory full
* Error number 88 : Direct message memory full

PC	W	L	0	,	147 + Direct message	:	PLU struct(147) + Direct message(1~300)	:	CK	CR
Scale	G	L	Scale ID	,	4	:	DeptPLU number(4)	:	CK	CR
	N	L	Scale ID	,	5	:	DeptPLU number(4) + Error number(1)	:	CK	CR

3. Upload PLU with room number

* If direct Message's last char must be null.
* Error number 99 : PLU data end

PC	R	L	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	L	Scale ID	,	151	:	Room number(4) + PLU struct(147)	:	CK	CR
	W	L	Scale ID	,	451	:	Room number(4) + PLU struct(147) + Direct message(300)	:	CK	CR
	N	L	Scale ID	,	5	:	Room number(4) + Error number(1)	:	CK	CR

4. Upload PLU with DeptPLU number

* If direct Message's last char must be null.
* Error number 98 : PLU isn't exist

PC	R	L	0	,	4	:	DeptPLU number(4)	:	CK	CR
Scale	W	L	Scale ID	,	151	:	DeptPLU number(4) + PLU struct(147)	:	CK	CR
	W	L	Scale ID	,	451	:	DeptPLU number(4) + PLU struct(147) + Direct message(300)	:	CK	CR
	N	L	Scale ID	,	5	:	DeptPLU number(4) + Error number(1)	:	CK	CR

5. Erase PLU

* If Dept. number is 0 and PLU number is 0, Erase All PLUs
* If you want to erase many PLUs,
  Erase All PLUs and Download PLU because it's faster than Erasing individual PLUs.
* Error number 98 : PLU isn't exist

PC	W	L	0	,	6	:	Dept. number(2) + PLU number(4)	:	CK	CR

Scale	G	L	Scale ID	,	4	:	DeptPLU number(4)	:	CK	CR
	N	L	Scale ID	,	5	:	DeptPLU number(4) + Error number(1)	:	CK	CR

<< PLU Data Struct >>

* All number data's byte order is little endian
* PLU size (Bytes) : 147(Basic) + 0~300(Direct ingredient)

No	Offset	Type	Description
1	0	INT16U	Department number
2	2	INT32U	PLU number
3	6	INT8U	PLU type
4	7	char[40]	PLU name 1
5	47	char[40]	PLU name 2
6	87	char[5]	PLU name 3
7	92	INT16U	Group number
8	94	INT16U	Label number
9	96	INT16U	Aux. label number (for future use, set 0)
10	98	INT16U	Origin number
11	100	INT8U	Unit weight number (for future use, set 0)
12	101	INT32U	Fixed weight
13	105	INT32U	Item Code
14	109	INT16U	PCS (quantity)
15	111	INT8U	PCS (quantity) symbol number
16	112	INT8U	Use fixed price type
17	113	INT32U	Unit price
18	117	INT32U	Special price
19	121	INT32U	Tare weight
20	125	INT8U	Tare number
21	126	INT16U	Barcode number
22	128	INT16U	Aux. barcode number (for future use, set 0)
23	130	INT16U	Produced date
24	132	INT16U	Packed date
25	134	INT8U	Packed time
26	135	INT32U	Sell By date
27	139	INT8U	Sell By time
28	140	INT16U	Message number
29	142	INT16U	reserved, set 0
30	144	INT16U	reserved, set 0
31	146	INT8U	Sale message number
32	147	char[300]	Direct message (Option)

EX) Download PLU without direct message

PC

```
57 4C 00 00 00 00 2C 93 00 3A 01 00 01 00 00 00 01 50 4C 55 30 30 30 31 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6E 61 6D 65 32
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 6E 61 6D 65 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 11 27 00 00 00 00 00 00 11
27 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 3A FF 0D
WL....,? .:.....PLU0001.....name2.....name3.....
...'.....'.....:y.
```

Download PLU 1 of dept 1 (147 bytes no message)

Scale

```
47 4C 01 00 00 00 2C 04 00 3A 41 42 0F 00 3A 37 0D
GL....,..:AB..:7.
```

Downloading PLU 1 of dept 1 is complete (deptPLU number = 1000001 = 0x0F4241)

EX) Download PLU with direct message

PC

```
57 4C 00 00 00 00 2C 9C 00 3A 01 00 01 00 00 00 01 50 4C 55 30 30 30 31 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6E 61 6D 65 32
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 6E 61 6D 65 33 00 00 00 00 00 00 00 00 00 00 00 00 00 11 27 00 00 00 00 00 11
27 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 6D 65 73 73 61 67 65 31 00 3A 1E 0D
WL....,? .:.......PLU0001.....................................name2.....................................name3.....
...'.....'.....................................message1.:..
```

Download PLU 1 of dept 1 (156 bytes with message : message1)

Scale

```
47 4C 01 00 00 00 2C 04 00 3A 41 42 0F 00 3A 37 0D
GL....,..:AB..:7.
```

Downloading PLU 1 of dept 1 is complete (deptPLU number = 1000001 = 0x0F4241)

EX) Upload PLU with room number

PC

```
52 4C 00 00 00 00 2C 04 00 3A 01 00 00 00 3A A5 0D
RL....,..:....:¥
```

Upload PLU in room 1

Scale

```
57 4C 01 00 00 00 2C C3 01 3A 01 00 00 00 01 00 01 00 00 00 01 50 4C 55 30 30 30 31 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6E
61 6D 65 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 6E 61 6D 65 33 00 00 00 00 00 00 00 00 00 00 00 00 11 27 00 00 00
00 00 00 11 27 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 6D 65 73 73 61 67 65 31 00 31 00 65 60 29 65 6E 74 20 20 20 00 20 20 20 20
...
20 20 20 20 20 20 20 20 20 20 20 20 20 A0 20 20 20 20 20 20 20 20 20 20 20 28 20 3A C8 0D
WL....,
A.:.....PLU0001.....................................name2.....................................name3.....'.
.....'.....................................message1...  ( :E.
```

Uploading PLU 1 of dept 1 in room 1 (PLU include direct message 300 bytes)

EX) Erase PLU

PC

```
57 4C 00 00 00 00 2C 06 00 3A 01 00 01 00 00 00 3A A8 0D
WL....,..:.....:¨.
```

Erase PLU 1 of Dept. 1

Scale

```
47 4C 01 00 00 00 2C 04 00 3A 41 42 0F 00 3A 37 0D
GL....,..:AB..:7.
```

Erasing PLU 1 of dept 1 in room 1 is complete (deptPLU number = 1000001 = 0x0F4241)

EX) Erase All PLU

PC

```
57 4C 00 00 00 00 2C 06 00 3A 00 00 00 00 00 00 3A A6 0D
WL....,..:.....:¦
```

Erase All PLU (PLU 0 of Dept. 0 means all PLU)

Scale

```
47 4C 01 00 00 00 2C 04 00 3A 00 00 00 00 3A A5 0D
GL....,..:....:¥
```

Erasing All PLUs is complete (deptPLU number = 0)

3. Indirect Message Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	M	**0**							

* If Indirect Message's last char must be null(0) like a C string type data.

1. Download Indirect message

PC	W	M	0	,	404	:	Message number(4) + Message(400)	:	CK	CR
Scale	G	M	Scale ID	,	4	:	Message number(4)	:	CK	CR
	N	M	Scale ID	,	5	:	Message number(4) + Error number(1)	:	CK	CR

2. Upload Indirect message

* Error number 99 : Indirect message data end

PC	R	M	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	M	Scale ID	,	404	:	Message number(4) + Message(400)	:	CK	CR
	N	M	Scale ID	,	5	:	Message number(4) + Error number(1)	:	CK	CR

3. Erase Indirect message

PC	W	M	0	,	4	:	Message number(4)	:	CK	CR
Scale	G	M	Scale ID	,	4	:	Message number(4)	:	CK	CR
	N	M	Scale ID	,	5	:	Message number(4) + Error number(1)	:	CK	CR

EX) Download indirect message

PC

```
57 4D 00 00 00 00 2C 94 01 3A 1D 00 00 00 49 6E 67 72 65 64 69 65 6E 74 73 00 00 00 00 00 00
... 00 00 00 3A CE 0D
WM....,? .:....Ingredients.............:I.
```

Download indirect message 29 (400 bytes : Ingredients)

Scale

```
47 4D 01 00 00 00 2C 04 00 3A 1D 00 00 00 3A C2 0D
GM....,..:....:A.
```

Downloading indirect message 29 is complete

4. Barcode Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	B	**0**							

* If Barcode's last char must be null(0) like a C string type data.

1. Download Barcode

PC	W	B	0	,	36	:	Barcode number(4) + Barcode(32)	:	CK	CR
Scale	G	B	Scale ID	,	4	:	Barcode number(4)	:	CK	CR
	N	B	Scale ID	,	5	:	Barcode number(4) + Error number(1)	:	CK	CR

2. Upload Barcode

* Error number 99 : Barcode data end

PC	R	B	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	B	Scale ID	,	36	:	Barcode number(4) + Barcode(32)	:	CK	CR
	N	B	Scale ID	,	5	:	Barcode number(4) + Error number(1)	:	CK	CR

3. Erase Barcode

PC	W	B	0	,	4	:	Barcode number(4)	:	CK	CR
Scale	G	B	Scale ID	,	4	:	Barcode number(4)	:	CK	CR
	N	B	Scale ID	,	5	:	Barcode number(4) + Error number(1)	:	CK	CR

<< Barcode Data Struct >>

No	Offset	Type	Description
1	0	INT8U	Barcode type
2	1	char[31]	Barcode text

EX) Upload barcode data

PC

52 42 00 00 00 00 2C 04 00 3A 01 00 00 00 3A A5 0D
RB.....,..:....:¥

Request barcode data in room 1

Scale

57 42 01 00 00 00 2C 24 00 3A 01 00 00 00 01 44 44 49 49 49 49 49 50 50 50 50 50 43 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3A 8F 0D
WB....,$.:.....DDIIIIIPPPPPC.....:? .

Response barcode data 1 in room 1, (type = EAN13, text = "DDIIIIIPPPPPC")

5. Shop(Store) Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	S	**0**							

* If Shop name, tel, addr size is under memory,
  last char must be null like a C string type data, but sending data size is 126.

1. Download Shop

PC	W	S	0	,	130	:	Shop number(4) + Shop(126)	:	CK	CR
Scale	G	S	Scale ID	,	4	:	Shop number(4)	:	CK	CR
	N	S	Scale ID	,	5	:	Shop number(4) + Error number(1)	:	CK	CR

2. Upload Shop

* Error number 99 : Shop Data End

PC	R	S	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	S	Scale ID	,	130	:	Shop number(4) + Shop(126)	:	CK	CR
	N	S	Scale ID	,	5	:	Shop number(4) + Error number(1)	:	CK	CR

3. Erase Shop

PC	W	S	0	,	4	:	Shop number(4)	:	CK	CR
Scale	G	S	Scale ID	,	4	:	Shop number(4)	:	CK	CR
	N	S	Scale ID	,	5	:	Shop number(4) + Error number(1)	:	CK	CR

<< Shop Data Struct >>

* Shop size (Bytes) : 126

No	Offset	Type	Description
1	0	char[26]	Name
2	26	char[20]	Tel. number
3	46	char[80]	Address

EX) Download shop data

PC

```
57 53 00 00 00 00 2C 82 00 3A 01 00 00 00 53 74 6F 72 65 20 4E 61 6D 65 20 31 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 54 45 4C 20 31 32 33 2D 34 35 36 37 00 00 00 00 00 00 00 00 61 64
64 72 65 73 73 20 31 32 33 34 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3A C5 0D
WS....,? .:....Store Name 1.............TEL 123-4567.....address
```

Download shop data 1

Scale

```
47 53 01 00 00 00 2C 04 00 3A 01 00 00 00 3A A6 0D
GS....,.:....:¦
```

Downloading shop data 1 is complete

6. Sales Message Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	G	**0**							

* If Sales Message's last char must be null(0) like a C string type data

1. Download Sales Message

PC	W	G	0	,	34	:	Message number(4) + Message(30)	:	CK	CR
Scale	G	G	Scale ID	,	4	:	Message number(4)	:	CK	CR
	N	G	Scale ID	,	5	:	Message number(4) + Error number(1)	:	CK	CR

2. Upload Sales Message

* Error number 99 : Sales message data end

PC	R	G	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	G	Scale ID	,	34	:	Message number(4) + Message(30)	:	CK	CR
	N	G	Scale ID	,	5	:	Message number(4) + Error number(1)	:	CK	CR

3. Erase Sales Message

PC	W	G	0	,	4	:	Message number(4)	:	CK	CR
Scale	G	G	Scale ID	,	4	:	Message number(4)	:	CK	CR
	N	G	Scale ID	,	5	:	Message number(4) + Error number(1)	:	CK	CR

EX) Download Sales message

PC

```
57 47 00 00 00 00 2C 22 00 3A 05 00 00 00 54 68 61 6E 6B 20 79 6F 75 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 3A 3A 0D
WG....,".:....Thank you.....::.
```

Download sales message number 5 data (Thank you)

Scale

```
47 47 01 00 00 00 2C 04 00 3A 05 00 00 00 3A AA 0D
GG....,..:....:ª.
```

Downloading sales message number 5 is complete

7. Origin Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	O	0							

* If Origin's last char must be null(0) like a C string type data

1. Download Origin

PC	W	O	0	,	36	:	Origin number(4) + Origin(32)	:	CK	CR
Scale	G	O	Scale ID	,	4	:	Origin number(4)	:	CK	CR
	N	O	Scale ID	,	5	:	Origin number(4) + Error number(1)	:	CK	CR

2. Upload Origin

* Error number 99 : Origin sata end

PC	R	O	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	O	Scale ID	,	36	:	Origin number(4) + Origin(32)	:	CK	CR
	N	O	Scale ID	,	5	:	Origin number(4) + Error number(1)	:	CK	CR

3. Erase Origin

PC	W	O	0	,	4	:	Origin number(4)	:	CK	CR
Scale	G	O	Scale ID	,	4	:	Origin number(4)	:	CK	CR
	N	O	Scale ID	,	5	:	Origin number(4) + Error number(1)	:	CK	CR

EX) Upload Origin

PC

```
52 4F 00 00 00 00 2C 04 00 3A 01 00 00 00 3A A5 0D
RO....,..:....:¥
```

Request origin in room number 1

Scale

```
57 4F 01 00 00 00 2C 24 00 3A 02 00 00 00 4B 6F 72 65 61 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3A B9 0D
WO....,$.:....Korea.....................:¹.
```

Response origin number 2 data(Korea) in room 1

PC

```
52 4F 00 00 00 00 2C 04 00 3A 02 00 00 00 3A A6 0D
RO....,..:....:¦
```

Request origin in room number 2

Scale

```
4E 4F 01 00 00 00 2C 05 00 3A 02 00 00 00 63 3A 0B 0D
NO....,..:....c:..
```

No more data is exist (Error number : 99(0x63))

8. TLU(Tare) Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	T	0							

* TLU(Tare) : tare look up weight (INT32U)

1. Download TLU

PC	W	T	0	,	8	:	TLU number(4) + TLU(4)	:	CK	CR
Scale	G	T	Scale ID	,	4	:	TLU number(4)	:	CK	CR
	N	T	Scale ID	,	5	:	TLU number(4) + Error number(1)	:	CK	CR

2. Upload TLU

* Error number 99 : TLU data end

PC	R	T	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	T	Scale ID	,	8	:	TLU number(4) + TLU(4)	:	CK	CR
	N	T	Scale ID	,	5	:	TLU number(4) + Error number(1)	:	CK	CR

3. Erase TLU

PC	W	T	0	,	4	:	TLU number(4)	:	CK	CR
Scale	G	T	Scale ID	,	4	:	TLU number(4)	:	CK	CR
	N	T	Scale ID	,	5	:	TLU number(4) + Error number(1)	:	CK	CR

EX) Upload TLU

PC

```
52 54 00 00 00 00 2C 04 00 3A 01 00 00 00 3A A5 0D
RT.....,..:....:¥
```

Request tare in room number 1

Scale

```
57 54 01 00 00 00 2C 08 00 3A 01 00 00 00 E8 03 00 00 3A 95 0D
WT.....,..:....e...:? .
```

Response tare number 1 data(1000) in room 1

PC

```
52 54 00 00 00 00 2C 04 00 3A 02 00 00 00 3A A6 0D
RT.....,..:....:¦
```

Request tare in room number 2

Scale

```
4E 54 01 00 00 00 2C 05 00 3A 02 00 00 00 63 3A 0B 0D
NT.....,..:....c:..
```

No more data is exist (Error number : 99(0x63))

9. PCS Symbol Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	Y	**0**							

* If Symbol's last char must be null(0) like a C string type data.

1. Download PCS symbol

PC	W	Y	0	,	14	:	Symbol number(4) + Symbol(10)	:	CK	CR
Scale	G	Y	Scale ID	,	4	:	Symbol number(4)	:	CK	CR
	N	Y	Scale ID	,	5	:	Symbol number(4) + Error number(1)	:	CK	CR

2. Upload PCS symbol

* Error number 99 : TLU data end

PC	R	Y	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	Y	Scale ID	,	14	:	Symbol number(4) + Symbol(10)	:	CK	CR
	N	Y	Scale ID	,	5	:	Symbol number(4) + Error number(1)	:	CK	CR

3. Erase PCS symbol

PC	W	Y	0	,	4	:	Symbol number(4)	:	CK	CR
Scale	G	Y	Scale ID	,	4	:	Symbol number(4)	:	CK	CR
	N	Y	Scale ID	,	5	:	Symbol number(4) + Error number(1)	:	CK	CR

EX) Upload PCS symbol

PC

```
52 59 00 00 00 00 2C 04 00 3A 01 00 00 00 3A A5 0D
RY.....:....:¥
```

Request Symbol in room number 1

Scale

```
57 59 01 00 00 00 2C 0E 00 3A 01 00 00 00 70 63 73 00 00 00 00 00 00 00 3A F6 0D
WY.....:....pcs.....:÷.
```

Response symbol number 1 data(pcs) in room 1

PC

```
52 59 00 00 00 00 2C 04 00 3A 02 00 00 00 3A A6 0D
RY.....:....:↓
```

Request Symbol in room number 2

Scale

```
4E 59 01 00 00 00 2C 05 00 3A 02 00 00 00 63 3A 0B 0D
NY.....:....c:..
```

No more data is exist (Error number : 99(0x63))

10. Speed Key Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 580 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	K	**0**							

1. Download Speed key

PC	W	K	0	,	580	:	Key number(2) + Key type(2) + Key Table(576)	:	CK	CR
Scale	G	K	Scale ID	,	4	:	Key number(2) + Key type(2)	:	CK	CR
	N	K	Scale ID	,	5	:	Key number(2) + Key type(2) + Error number(1)	:	CK	CR

2. Upload Speed key

PC	R	K	0	,	4	:	Key number(2) + Key type(2)	:	CK	CR
Scale	W	K	Scale ID	,	580	:	Key number(2) + Key type(2) + Key Table(576)	:	CK	CR
	N	K	Scale ID	,	5	:	Key number(2) + Key type(2) + Error number(1)	:	CK	CR

<< Speed Key Data Struct >>

* Speed key table size (Bytes) : 576 = 72(key) * 2(shift) * 4(PLU number)

No	Offset	Type	Description
1	0	INT32U	Speed key 1 PLU number
2	4	INT32U	Speed key 2 PLU number
…	…	…	…
143	568	INT32U	Speed key 143 PLU number
144	572	INT32U	Speed key 144 PLU number

EX) Upload speed key table

PC

```
52 4B 00 00 00 00 2C 04 00 3A 01 00 00 00 3A A5 0D
RK.....,..:....:¥
```

Request speed key table 1 data

Scale

```
57 4B 01 00 00 00 2C 44 02 3A 01 00 02 00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 05
... 00 3A B2 0D
WK....,D.:.........:²
```

Response speed key 1 data (580 bytes : keytype = pole, speed key 1 = PLU 1, …)

11. Report Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	U	**0**							

* DeptPLU Number : Department Number * 1000000 + PLU number
  Ex) Dept 10, PLU 999999 -> 10999999
* Scale ID : default is 1

1. Upload PLU report

* Error number 99 : Report data end

PC	R	U	0	,	4	:	Room number(4)	:	CK	CR
Scale	W	U	Scale ID	,	24	:	Room number(4) + DeptPLU number(4) + Report struct(16)	:	CK	CR
	N	U	Scale ID	,	5	:	Room number(4) + Error number(1)	:	CK	CR

2. Upload other report

* Report number
  100000000=grand total
  100000001=all PLU
  100000002=NON-PLU
  200000000 ~ 200000099=dept 0 ~ 99 (0 : Don't use)
  300000000 ~ 300000099=group 0 ~ 99
  400000000 ~ 400000023=hourly 0 ~ 23

PC	R	U	0	,	4	:	Report number(4)	:	CK	CR
Scale	W	U	Scale ID	,	20	:	Report number(4) + Report struct(16)	:	CK	CR
	N	U	Scale ID	,	5	:	Report number(4) + Error number(1)	:	CK	CR

3. Upload report start time

* Start time number : 100000010

PC	W	U	0	,	4	:	Start time	:	CK	CR
Scale	W	U	Scale ID	,	10	:	Start time number(4)	:	CK	CR
	N	U	Scale ID	,	5	:	Start time number(4)	:	CK	CR

4. Clear all report

* Clear report number : 100000000
* Set start time to current time

PC	W	U	0	,	4	:	Clear report number(4)	:	CK	CR
Scale	G	U	Scale ID	,	4	:	Clear report number(4)	:	CK	CR
	N	U	Scale ID	,	5	:	Clear report number(4) + Error number(1)	:	CK	CR

<< Report Data Struct >>

* Report struct size (Bytes) : 16

No	Offset	Type	Description
1	0	INT32U	Sale count
2	4	INT32U	Weight
3	8	INT32U	PCS (Quantity)
4	12	INT32U	Price

<< Report Data Struct >>

* Start time struct size (Bytes) : 6

No	Offset	Type	Description
1	0	INT8U	Year (20XX)
2	1	INT8U	Month
3	2	INT8U	Date
4	3	INT8U	Hour
5	4	INT8U	Minute
6	5	INT8U	Second

EX) Upload PLU report

PC

52 55 00 00 00 00 2C 04 00 3A 0A E1 F5 05 3A 89 0D

RU....,..:.ao.:? .

Upload start time (Start time number : 100000010 = 0x05F5E10A)

Scale

57 55 01 00 00 00 2C 0A 00 3A 0A E1 F5 05 07 09 0A 0F 16 10 3A DF 0D

WU....,..:.ao.....:ß.

Response start time (Year : 7, Month : 9, Date : 10, Hour 15, Min : 22, Sec : 16)

EX) Upload PLU report

PC

52 55 00 00 00 00 2C 04 00 3A 01 00 00 00 3A A5 0D

RU....,..:....:¥

Upload PLU report in room 1

Scale

57 55 01 00 00 00 2C 18 00 3A 01 00 00 00 41 42 0F 00 38 01 00 00 61 7A 02 00 00 00 00 00 5C B2 18 00 3A 88 0D

WU....,..:....AB..8...az.....₩².:? .

Uploading PLU report 1 of dept 1 in room 1 is complete (deptPLU number = 1000001 = 0x0F4241)

Sale count : 312, Weight : 162.401 kg, PCS : 0, Price : $16185.24

EX) Upload Depatment 1 report

PC

52 55 00 00 00 00 2C 04 00 3A 01 C2 EB 0B 3A 5D 0D

RU....,..:.Ae.:].

Upload Department 1 report (report number = 200000001)

Scale

57 55 01 00 00 00 2C 14 00 3A 01 C2 EB 0B 99 01 00 00 1E 8F 03 00 26 27 00 00 3C 7D 40 00 3A FE 0D

WU....,..:.Ae.? ....? ..&'..<}@.:þ.

Uploading Department 1 report is complete

Sale count : 409, Weight : 233.246 kg, PCS : 10022, Price : $42263.64

EX) Clear All report

PC

57 55 00 00 00 00 2C 04 00 3A 00 E1 F5 05 3A 7F 0D

WU....,..:.Ae.:].

Clear All report (report number = 100000000 = 0x05F5E100)

Scale

47 55 01 00 00 00 2C 04 00 3A 00 E1 F5 05 3A 80 0D

GU....,..:.Ae.? ....? ..&'..<}@.:þ.

Clearing All report is complete

12. Label Protocol Structure

Header						Body	Tail		
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	A	**0**							

1. Download Label

PC	W	A	0	,	4 + data size(1~256)	:	Label number(2) + Data offset(2) + Data(1~256)	:	CK	CR
Scale	G	A	Scale ID	,	4	:	Label number(2) + Data offset(2)	:	CK	CR
	N	A	Scale ID	,	5	:	Label number(2) + Data offset(2) + Error number(1)	:	CK	CR

2. Upload Label

PC	R	A	0	,	4	:	Label number(2) + Data offset(2)	:	CK	CR
Scale	W	A	Scale ID	,	260	:	Label number(2) + Data offset(2) + Data(256)	:	CK	CR
	N	A	Scale ID	,	5	:	Label number(2) + Data offset(2) + Error number(1)	:	CK	CR

<< Label Data Struct >>

* Label Data Size (Bytes) : 3072
* CL5000J label form is different to CL5000 label form

13. Bitmap Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	J	**0**							

1. Download Bitmap

PC	W	J	0	,	4 + data size(1~256)	:	Bitmap number(2) + Data offset(2) + Data(1~256)	:	CK	CR
Scale	G	J	Scale ID	,	4	:	Bitmap number(2) + Data offset(2)	:	CK	CR
	N	J	Scale ID	,	5	:	Bitmap number(2) + Data offset(2) + Error number(1)	:	CK	CR

2. Upload Bitmap

PC	R	J	0	,	4	:	Bitmap number(2) + Data offset(2)	:	CK	CR
Scale	W	J	Scale ID	,	260	:	Bitmap number(2) + Data offset(2) + Data(256)	:	CK	CR
	N	J	Scale ID	,	5	:	Bitmap number(2) + Data offset(2) + Error number(1)	:	CK	CR

<< Bitmap Data Struct >>

* Label data size (Bytes) : 8192 (Max)

No	Offset	Type	Description
1	0	INT8U	Bitmap width (dots)
2	2	INT8U	Bitmap height (dots)
3	4	INT8U[]	Bitmap data (0: white, 1:Black)

14. Status Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	N	**0**							

1. Upload Version

PC	R	N	0	,	0	:	none	:	CK	CR
Scale	W	N	Scale ID	,	30	:	Status(30)	:	CK	CR

<< Status Struct >>

* Status struct size : 30 Bytes

No	Offset	Type	Description
1	0	INT8U	Load flag (0:Zero, 1:Non zero, 2:Overload)
2	1	INT8U	Stable flag (0:Unstable, 1:Stable)
3	2	INT8U	Tare flag (0:No tare, 1:tare)
4	3	INT8U	Dual range (2: dual range)
5	4	INT8U	Weight unit (0:kg)
6	5	INT8U	Weight decimal point
7	6	INT8U	Price decimal point
8	7	INT8U	reserved
9	8	INT32U	Tare
10	12	INT32S	Weight
11	16	INT32U	Unit price
12	20	INT32U	Total price
13	24	INT32U	PLU number
14	28	INT16U	Department number

15. Version Protocol Structure

Header						Body		Tail	
Opcode (2 Bytes)		Address (4 Bytes)	,	Data Length (2 Bytes)	:	Data (Max. 512 Bytes)	:	Checksum (1 Byte)	CR (1 Byte)
Opcode0	Opcode1								
	V	**0**							

1. Upload Version

PC	R	V	0	,	0	:	none	:	CK	CR
Scale	W	V	Scale ID	,	16	:	Version(16)	:	CK	CR

<< Version Struct >>

* Version struct size : 16 Bytes

No	Offset	Type	Description
1	0	INT8U	Class (default : 0)
2	1	INT16U	Model number (default : 5010)
3	3	INT8U	Type (1-Bench, 2-Pole, 3-Hanging, 4-Self)
4	4	INT16U	Scale ID
5	6	INT16U	Main version
6	8	INT16U	Sub version
7	10	INT16U	Data structure version
8	12	INT16U	Country number
9	14	char[2]	reserved

EX) Upload version

PC	52 56 00 00 00 00 2C 00 00 3A 3A A0 0D RV.....,..::?.
Scale	57 56 01 00 00 00 2C 10 00 3A 00 92 13 02 00 00 01 00 03 00 01 00 00 00 00 00 3A 5D 0D WV.....,..:.?.....:].

Medel : CL5000J, Main version : 1, Sub vertion : 3, Data structure version : 1...