

COMMUNICATION PROTOCOL

1. Communication interface: RS232c, baud rate 9600, 8 digits, no parity, 1 start bit, 1 stop bit

2. Unit price setting, PLU downloading, PLU uploading, unit price & total price reading,

communication process is as follows: a. PC sends 0X44.

b. PC sends start package after receiving scale response 0x02

c. PC sends command package after receiving scale response 0x02.

d. After PC receiving scale response 0x02 & command response package, if needs to send command, switch to c.

e. PC sends end package.

f. PC terminates after receiving scale response 0x02.

Start package, fixed 6 bytes: 11 00 00 00 00 EF5

End package, fixed 6 bytes: 33 00 00 00 00 CD

Command package command response package, format is as follows:

Cmd type adr1 adr0 datlen [dat] checksum

cmd : Only 55 read command or 77 write command.

type : Operating data type is as follows:

f9 : unit price or PLU

f4 : total price

adr1,adr0 : Form 2 bytes integer $adr = adr1 * 100 + adr0$

When adr is 0, stands for operating current unit price or total price

When adr is not 0, stands for operating memory area PLU, at this time adr is PLU's position.

$$adr1 = (DC + pluN * 4) / 100$$

$$adr0 = (DC + pluN * 4) \% 100$$

datlen : data length of reading or writing. Data length of PLU is 4

[dat] : the data of writing command, length is decided by datlen

checksum : checksum, checksum =

$$100 - (cmd + type + dar1 + dar0 + datlen + [dat]) \% 100$$

If set current unit price 111.00:

DO 44
DI 02
DO 11 00 00 00 00 ef
DI 02
DO 77 f9 00 00 04 00 00 2b 5c 01
DI 02
DO 33 00 00 00 00 cd
DI 02

If set unit price of PLU1 111.00:

DO 44
DI 02
DO 11 00 00 00 00 ef
DI 02
DO 77 f9 00 e0 04 00 00 2b 5c 21
DI 02
DO 33 00 00 00 00 cd
DI 02

If read PLU1: DO 44

DI 02
DO 11 00 00 00 00 ef
DI 02
DO 55 f9 00 e0 04 ce
DI 02
DI 55 fd 00 e0 04 00 00 2b 5c 43 ;;
DO 33 00 00 00 00 cd
DI 02

If read current unit price & total price: DO 44

DI 02
DO 11 00 00 00 00 ef
DI 02

DO 55 f4 00 00 09 ae
 DI 02
 DI 55 f4 00 00 04
 DI 00 00 00 00 de ;; total price 2.22
 DI 00 00 2b 5c ;; unit price 111.00
 DI 4e
 DO 33 00 00 00 00 cd
 DI 02

3 Read current weight

PC sends 0X05, after receiving scale response 0x06, senfs 0x11, scale will response weight package.

Weight Package formats: SOH STX STA SIGN WEIGHT_ASCII WEIGHT_UNIT BCC ETX
 EOT

SOH: start to sending symbol, one byte, fixed at 01H.

STX: start symbol of data item, one byte, fixed at 02H.

STA: words to describe weight status, one byte, only possible to be 53H or 55H46H.

53H –weight is stable, 55H – weight is unstable, 46H –weight is abnormal.

SIGN :sign of weight, one byte, only possible to be 2DH or 20H. 2DH-weight is negative, 20H-weight is positive.

WEIGHT_ASCII: character string of weight. 5~6 byte, only possible to be 30H~39H, decimal (2EH), space (20H)

WEIGHT_UNIT : character string of weight unit. 1~2 byte, only possible to be : ‘TJ’: catty of Taiwan, ‘TL’: tael of Taiwan, ‘SJ’: ‘SJ’: jin, ‘LB’: pound, ‘KG’: kilogram, ‘G’: gram.

BCC: character of data verify. One byte. Set STA to D1, SIGN to D2, byte before BCC to DN, so $BCC=D1 \wedge D2 \wedge \wedge DN \wedge \wedge$ is a logic XOR operator.

ETX:symbol of ending data item, one byte, fixed at 03H.

EOT: symbol of ending sending, one byte, fixed at 04H.