



# **WPL**

# **Programming Guide**

For Firmware Version 1.0 and Later

Version2.1 2018/01/30

# Update History

<b>Date</b>	<b>Content</b>
<b>2017.06.02</b>	Add sGPI command
<b>2017.06.02</b>	Add sGPO command
<b>2017.06.02</b>	Add sGPOLEV command
<b>2017.06.09</b>	Modify document of sTPHY command
<b>2017.09.04</b>	Add rKEYBOARD and sKEYBOARD command
<b>2017.09.04</b>	Modify rLCD and sLCD command
<b>2017.09.07</b>	Modify sCODEPAGE command
<b>2018.01.03</b>	Add rDISKINFO command
<b>2018.01.03</b>	Add BASIC commands
<b>2018.01.30</b>	Modify w1DB and wTEXT commands
<b>2018.04.30</b>	Add wMLTEXT command

# Contents

Update History .....	2
Contents .....	3
WPL (Wincode Printing Command).....	10
Label Formatting Commands .....	12
w1DB .....	12
w2DB,16K .....	21
w2DB,49 .....	22
w2DB,AZ .....	23
w2DB,DM.....	25
w2DB, GM.....	26
w2DB, MPDF417.....	28
w2DB,MQR .....	30
w2DB,MX.....	32
w2DB,PDF417 .....	34
w2DB,QR.....	35
wBOX .....	37
wDIAGONAL.....	38
wELLIPSE.....	39
wLINE .....	40
wIMAGE .....	42
wPRINT .....	44
wREVERSE.....	45
wTEXT .....	46
wMLTEXT .....	48

File Management Commands .....	50
wDELETE .....	51
wLOAD .....	52
wLOADIMG .....	53
wMD .....	54
wSAVE.....	55
wSAVEB64.....	58
wSAVEEND.....	60
Defined Variable Commands.....	61
vCOUNT .....	62
vDATE .....	64
vTIME.....	66
vVAR .....	68
#SYS .....	70
Retrieve Setting Commands.....	71
rBACKOFFSET.....	72
rBLINE .....	73
rBTNAME .....	74
rBTPINCODE.....	76
rBUZZER .....	77
rCUTACTION .....	78
rCUTMODE .....	79
rCUTNUM .....	80
rDEMAND .....	81
rDENSITY.....	82
rDEVICE.....	83
rDIR.....	84
rDIRECTION.....	85

rDPI .....	86
rDISKINFO .....	87
rFEEDKEY .....	88
rGAP .....	89
rKEYBOARD .....	90
rLABEL .....	91
rLCD .....	92
rMODEL .....	94
rNETIP .....	95
rNETMAC .....	96
rNETNAME .....	97
rNETPORT .....	98
rOFFSET .....	99
rORIGIN .....	100
rRS232 .....	101
rRTCDATE .....	102
rRTCTIME .....	103
rSENSOR .....	104
rSN .....	105
rSPEED .....	106
rTHERMAL .....	107
rTPHY .....	108
rVERB .....	109
rVERK .....	110
Set Commands .....	111
sBACKOFFSET .....	112
sBLINE .....	113
sBTNAME .....	114

sBTPINCODE .....	115
sBUZZER.....	116
sCODEPAGE .....	117
sCUTACTION .....	119
sCUTMODE .....	120
sCUTNUM .....	121
sDEMAND .....	122
sDENSITY.....	123
sDEVICE.....	124
sDIRECTION.....	125
sFEEDKEY .....	126
sGAP .....	127
sGPI.....	128
sGPO .....	130
sGPOLEV .....	133
sHEADOPEN .....	134
sKEYBOARD.....	135
sLABEL.....	136
sLCD .....	137
sMIRROR.....	138
sOFFSET .....	139
sORIGIN .....	140
sRS232 .....	141
sRTCDATE .....	142
sRTCTIME.....	143
sSENSOR .....	144
sSPEED .....	145
sTHERMAL .....	146

sTPHY .....	147
BASIC Commands .....	152
General commands .....	153
ABS.....	153
ASC.....	154
CHR\$ .....	155
GOTO .....	156
INSTR .....	157
INT .....	158
LEFT\$.....	159
LEN.....	160
MID\$ .....	161
REM .....	162
RIGHT\$.....	163
SGN .....	164
STR\$ .....	165
STRING\$.....	166
VAL.....	167
Commands for decision structures .....	168
IF ... THEN ...ENDIF .....	168
Commands for looping structures.....	169
WHILE ... ENDWHILE .....	169
FOR ... NEXT .....	170
Function commands .....	171
ADDZ\$.....	171
ADDZ1\$.....	172
ASC.....	173
COMPUTE \$ .....	174

DATE\$ .....	175
DELCHAR \$ .....	176
DELCRLFTYPE \$ .....	177
DISCARD\$.....	178
GETFIELD \$.....	179
GETLINE \$ .....	180
INSERT\$ .....	181
ONLYNUM\$.....	182
REPLACEDIG\$.....	183
TIME\$.....	184
TRIMSTR\$ .....	185
BEEPON.....	186
LED.....	187
WRITEPORT.....	188
Input commands.....	189
EXINPUT .....	189
EXINPUT2.....	190
EXINPUT_NEW.....	191
INKEY\$ .....	193
INPUT .....	194
INPUT2.....	195
INPUT_LIMIT.....	196
INPUT_COPIES .....	197
File manipulation commands .....	198
OPENIN .....	198
OPENOUT.....	199
OPENUP .....	200
CLOSE.....	201

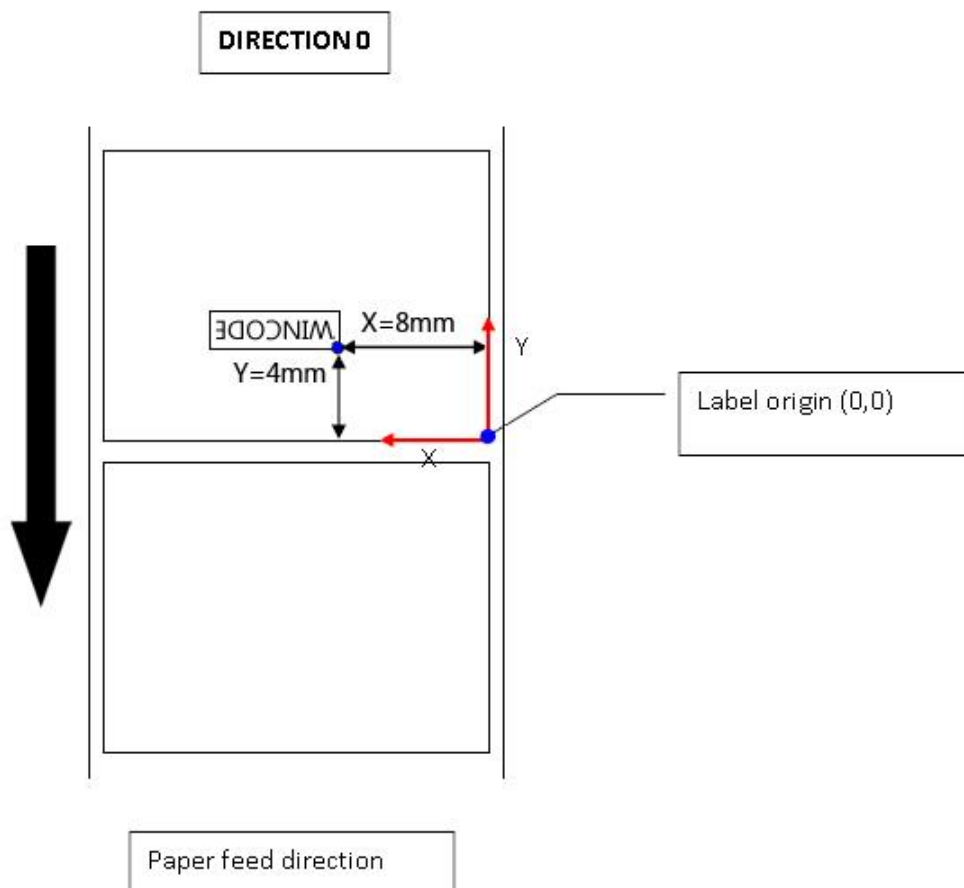


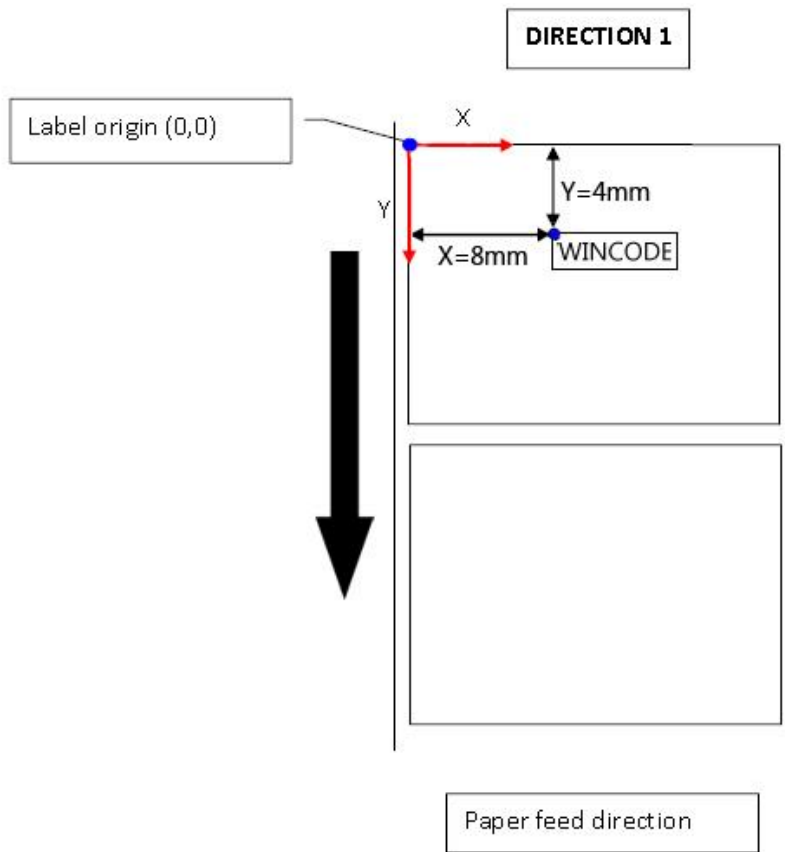
BGETEXT .....	202
BPUT .....	203
GET\$. .....	204
EOF.....	205
PTR.....	206
EXT .....	207
SEARCH_DB\$ .....	208
FETCH_DB \$ .....	209
Other .....	210
CLS .....	210
LOCATE .....	211
PRINT .....	212
Other Commands .....	213
aBACK .....	214
aCALIBRATE .....	215
aCUT .....	216
aDEFAULT .....	217
aFEED.....	219
aFORMAT.....	220
aRESET .....	221
REM .....	222
Appendix.....	223
Appendix - Maximum printing size.....	224
Appendix - Disk.....	224

# 1

## WPL (Wincode Printing Command)

- There are five classifications of WPL command as listed below
  - r: read printer's setting
  - s: adjust printer's setting
  - w: execute actions of drawing and processing
  - #,v: define parameter
  - a: other commands
- Each command uses symbol “ $\text{␣}$ ” as end (ASCII 0x0D, 0x0A).





# 2

## Label Formatting Commands

### w1DB

#### Description

This command defines a 1D Barcode.

#### Format

w1DB,P1,P2,P3,P4,P5,P6,P7,P8,[P9,][P10,][P11,][P12,][P13,][P14,][P15,]  
[P16,]"DATA1"



















Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Barcode Type</b>	See below Table.
<b>P5=Check digit</b>	0=None 1=Enable to show text 2=Enable to hide text
<b>P6= Narrow bar width (dot)</b>	Value: 1 ~ 30
<b>P7= Ratio between narrow and wide bars (dot)</b>	Value: 2 ~ 60









<b>P8= Bar code height (dot)</b>	Value: 1 ~ maximum printing length.
<b>P9=Human readable location (Option)</b>	HA=Above HB=Below HN=None
<b>P10=Text alignment (Option)</b>	AL=Left AC=Center AR=Right AJ= Justified
<b>P11=Font name (Option) or "Font name"</b>	F"0": 8x12 fixed dot font F"1": 10x16 fixed dot font F"2": 12x20 fixed dot font F"3": 14x24 fixed dot font F"4": 16x28 fixed dot font F"5": 20x36 fixed dot font F"6": 24x44 fixed dot font F"...": Content of TTF (True type font) name
<b>P12=Horizontal and vertical multiplier expands to font (Option)</b>	EX, X=1-10 ※TTF can upper than 10
<b>P13=Bearer Type (Option)</b>	BT=Top and bottom line BB=Box (Default) BN=None
<b>P14=Thickness Ratio (Option)</b>	TX, X=1-20
<b>P15=Quiet Zone Ratio (Option)</b>	QX, X=1-20
<b>P16=Pitch between Barcode to Text (Option)</b>	PX
<b>"DATA1"</b>	Represent a fixed data field

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Barcode Type Table (P4):

Value	Type	Outward
C11	Code 11	 12345678
C39	Code 39	 12345678
C93	Code 93	 12345678
C128	Code 128 auto A, B, C modes	 12345678
C128A	Code 128 mode A	 12345678
C128B	Code 128 mode B	 12345678
C128C	Code 128 mode C	 12345678
C128U	Code 128 UCC mode	 (00) 0 0000000 012345678 4
C128UE	Code 128 UCC/EAN mode EAN-128	 (12) 345678
CODA	CODABAR	 A 2 3 3 4 2 4 5 3 D
EAN8	EAN-8	 1234 5670
EAN82	EAN-8 2 digit add-on	 0123 4565 78
EAN85	EAN-8 5 digit add-on	 0000 1236 45678
EAN13	EAN-13	 0 000123 456784
EAN132	EAN-13 2 digit add-on	 0 000001 234565 78
EAN135	EAN-13 5 digit add-on	 0 000000 001236 45678
UPCA	UPC-A	 0 00123 45678 4

UPCA2	UPC-A 2 digit add-on	
UPCA5	UPC-A 5 digit add-on	
UPCE	UPC-E	
UPCE2	UPC-E 2 digit add-on	
UPCE5	UPC-E 5 digit add-on	
25V	Interleaved 2 of 5 Bar Code	
MSI	MSI without Check Digit	
PLESSEY	Plessey Bar Code	
MSIM10	MSI with Check Digit (MOD 10)	
MSIM11	MSI with Check Digit (MOD 11)	
MSIM1010	MSI with Check Digit (MOD 1010)	
MSIM1110	MSI with Check Digit (MOD 1110)	
POSTNET	USPS POSTNET	
PLANET	USPS PLANET	
RM4	Royal Mail 4-SCC	
JPC	Japanese Postal Code	
USPS	USPS Intelligent Mail	
KIX	KIX 4-State Customer Code	

<b>AP4AUTO</b>	Australia Post 4-SCC	
<b>AP4REP</b>	Reply Paid Barcode of AP4	
<b>AP4ROU</b>	Routing Barcode of AP4	
<b>AP4RED</b>	Redirection Barcode of AP4	
<b>KORP</b>	Korea Post	
<b>ITF14</b>	ITF-14 (14 digits of Interleaved 2 of 5) EAN-14	
<b>TELEPEN</b>	ASCII of TELEPEN Barcode	
<b>TELEPENN</b>	Numbers of TELEPEN Barcode	

If below barcode type is selected in the P4 field, the P13, P14, and P15 will be able to function.

Barcode type (P4)	Reference
<b>ITF-14</b>	P13, P14, and P15 are available
<b>25V</b>	P13, P14, and P15 are available

## Data length in each barcode type

Barcode Type	Data Length
<b>C11</b>	Variable
<b>C39</b>	Variable
<b>C93</b>	Variable
<b>C128</b>	Variable
<b>C128A</b>	Variable
<b>C128B</b>	Variable
<b>C128C</b>	Variable
<b>C128U</b>	18
<b>C128UE</b>	Variable
<b>CODA</b>	Variable



<b>EAN8</b>	7
<b>EAN82</b>	9
<b>EAN85</b>	12
<b>EAN13</b>	12
<b>EAN132</b>	14
<b>EAN135</b>	17
<b>UPCA</b>	11
<b>UPCA2</b>	13
<b>UPCA5</b>	16
<b>UPCE</b>	6
<b>UPCE2</b>	8
<b>UPCE5</b>	11
<b>25V</b>	Variable
<b>MSI</b>	Variable
<b>PLESSEY</b>	Variable
<b>MSIM10</b>	18
<b>MSIM11</b>	Variable
<b>MSIM1010</b>	18
<b>MSIM1110</b>	18
<b>POSTNET</b>	Variable
<b>PLANET</b>	Variable
<b>RM4</b>	Variable
<b>JPC</b>	Variable
<b>USPS</b>	Variable
<b>KIX</b>	18
<b>AP4AUTO</b>	Variable
<b>AP4REP</b>	Variable
<b>AP4ROU</b>	Variable
<b>AP4RED</b>	Variable
<b>KORP</b>	6
<b>ITF14</b>	14
<b>TELEPEN</b>	Variable
<b>TELEPENN</b>	Variable

# Example

Codes	Result
<p>sLABEL,812,609<sup>␣</sup></p> <p>sGAP,24,0<sup>␣</sup></p> <p>sDIRECTION,1<sup>␣</sup></p> <p><sup>␣</sup></p> <p>wTEXT,497,47,0,4,2,2,0,"39&amp;39C"<sup>␣</sup></p> <p>w1DB,72,24,0,C39,1,1,3,48,"12345678"<sup>␣</sup></p> <p>w1DB,48,120,0,C39,2,2,3,72,"87654321"<sup>␣</sup></p> <p>w1DB,48,240,0,C39,0,3,3,72,"12345678"<sup>␣</sup></p> <p>w1DB,24,360,0,C39,0,4,3,72,"87654321"<sup>␣</sup></p> <p>w1DB,24,480,0,C39,0,5,2,72,"12345678"<sup>␣</sup></p> <p><sup>␣</sup></p> <p>wBOX,50,24,256,70,3</p> <p>wBOX,28,120,400,100,3</p> <p>wBOX,28,240,560,100,3</p> <p>wBOX,5,360,670,110,3</p> <p>wBOX,5,480,680,110,3</p> <p> </p> <p>wPRINT,1</p>	 <p>12345678</p> <p>39&amp;39C</p>  <p>87654321</p>  <p>12345678</p>  <p>87654321</p>  <p>12345678</p>

Codes	Result
<p>sLABEL,812,609↵</p> <p>sGAP,24,0↵</p> <p>sDIRECTION,1↵</p> <p>↵</p> <p>w1DB,20,10,0,C39,0,1,3,48,HB,AL,F"0",E1,"12345678"↵</p> <p>w1DB,20,120,0,C39,1,1,3,48,HA,AC,F"1",E1,"12345678"↵</p> <p>w1DB,20,230,0,C39,2,1,3,48,HA,AR,F"2",E1,"12345678"↵</p> <p>w1DB,20,340,0,C39,0,1,3,48,HB,AJ,F"3",E1,"12345678"↵</p> <p>w1DB,220,120,0,C39,1,1,3,48,HA,AL,F"4",E1,"12345678"↵</p> <p>w1DB,220,230,0,C39,2,1,3,48,HB,AC,F"5",E1,"12345678"↵</p> <p>w1DB,220,390,0,C39,2,2,5,72,HB,AC,F"6",E3,"12345678"↵</p> <p>w1DB,420,120,0,C39,1,1,3,120,HN,AL,F"4",E1,"12345678"↵</p> <p>↵</p> <p>wPRINT,1↵</p>	
Codes	Result
<p>sLABEL,812,609↵</p> <p>sGAP,24,0↵</p> <p>sDIRECTION,1↵</p> <p>↵</p> <p>w1DB,20,10,0,25V,0,1,3,48,HB,AL,F"0",E1,BT,T2,Q2,"12345678"↵</p> <p>w1DB,20,120,0,25V,1,1,3,48,HA,AC,F"1",E1,BB,T4,Q4,"12345678"↵</p> <p>w1DB,20,230,0,25V,2,1,3,48,HA,AR,F"2",E1,BN,T5,Q5,"12345678"↵</p> <p>w1DB,20,340,0,25V,0,1,3,48,HB,AJ,F"3",E1,BT,T6,Q6,"12345678"↵</p> <p>w1DB,220,120,0,25V,1,1,3,48,HA,AL,F"4",E1,BB,T7,Q7,"12345678"↵</p> <p>w1DB,220,230,0,25V,2,1,3,48,HB,AC,F"5",E1,BN,T5,Q5,"12345678"↵</p> <p>w1DB,220,390,0,25V,2,2,5,72,HB,AC,F"6",E3,BT,T3,Q3,"12345678"↵</p> <p>w1DB,420,120,0,25V,1,1,3,120,HN,AL,F"4",E1,BB,T2,Q2,"12345678"↵</p> <p>↵</p> <p>wPRINT,1↵</p>	

Codes	Result
<p>sLABEL,812,609</p> <p>sGAP,24,0</p> <p>sDENSITY,8</p> <p>sSPEED,3</p> <p>sTHERMAL,1</p> <p>sDEVICE,1</p> <p>sDIRECTION,1</p> <p>w1DB,69,160,0,C39,0,2,2.000,55,HB,AC,F"LATINWD.TTF",E8.00,BN,T 3,Q10,P4,"Test123"</p> <p>w1DB,66,291,0,C128,0,2,2.000,55,HB,AC,F"LATINWD.TTF",E8.00,BN, T3,Q10,P4,"Code39 test"</p> <p>wPRINT,1</p>	<div style="text-align: center;">  <p><b>Test123</b></p>  <p><b>Code39 test</b></p> </div>

# w2DB,16K

## Description

This command defines a 2D Code-16K Barcode.

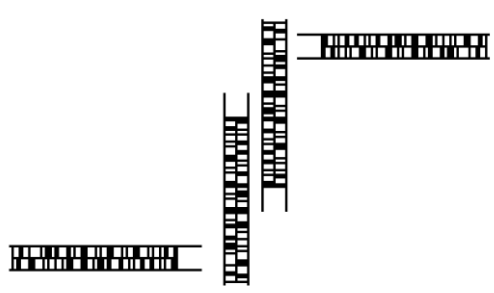
## Format

w2DB, 16K, P1, P2, P3,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Width (dot)</b>	Module width in dot
<b>P5=Height (dot)</b>	Bar height in dot
<b>"DATA"</b>	The content of 2D Code-16K Barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
␣ sLABEL,812,609␣ sGAP,24,0␣ sDIRECTION,1␣ ␣ w2DB,16K,480,100,0, 4,4,"123456"␣ w2DB,16K,400,200,1, 4,4,"123456"␣ w2DB,16K,320,500,2, 4,4,"123456"␣ w2DB,16K,420,400,3, 4,4,"123456"␣ ␣ wPRINT,1␣	

# w2DB,49

## Description

This command defines a 2D Code-49 Barcode.

## Format

w2DB,49,P1,P2,P3,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Width (dot)</b>	Module width in dot
<b>P5=Height (dot)</b>	Bar height in dot
<b>"DATA"</b>	The content of 2D Code-49 Barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
␣ sLABEL,812,609␣ sGAP,24,0␣ sDIRECTION,1 ␣ ␣ w2DB,49,480,250,0,4,4,"00556455555"␣ w2DB,49,380,250,1,4,4,"00556455555"␣ w2DB,49,320,250,2,4,4,"00556455555"␣ w2DB,49,420,350,3,4,4,"00556455555"␣ ␣ wPRINT,1,1␣	

# w2DB,AZ

## Description

This command defines a AZTEC 2D barcode.


## Format

w2DB,AZ,P1,P2,P3,P4,P5,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Symbol Scaling</b>	Value: 1-10
<b>P5=Symbol layer and/or correction levels</b>	0: Default error correction level 01 to 99: minimum error correction percentage 101 to 104: 1 to 4-layer compact symbol 201 to 232: 1 to 32-layer full-range symbol 300: Rune symbol
<b>"DATA"</b>	The content of AZTEC 2D barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

# Example

Codes	Result
sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1 ↵ ↵ w2DB,AZ,562,410,0,3,0,"12345678"↵ w2DB,AZ,200,410,0,4,103,"12345678"↵ w2DB,AZ,511,77,0,7,203,"12345678"↵ w2DB,AZ,207,77,0,7,300,"55"↵ wPRINT,1,1↵	



# w2DB,DM

## Description

This command defines a DataMatrix 2D barcode.


## Format

w2DB,P1,P2,P3,P4,P5,P6,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4= Scaling</b>	Value: 1-10
<b>P5=row</b>	Symbol size of row: 10 to 144
<b>P6=col</b>	Symbol size of col: 10 to 144
<b>"DATA"</b>	The content of 2D DataMatrix barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ w2DB,DM,650,480,0,4,12,12,"12345678"↵ w2DB,DM,543,207,0,9,12,12,"12345678"↵ w2DB,DM,443,411,0,4,32,32,"12345678"↵ w2DB,DM,78,110,0,4,80,80,"12345678"↵ ↵ wPRINT,1↵	

# w2DB, GM

## Description

This command defines a 2D Grid Matrix barcode.

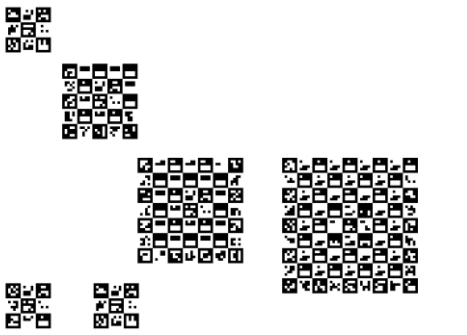
## Format

w2DB,GM,P1,P2,P3,P4,P5,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Data layer.</b>	Value: 0(Auto) -13.
<b>p5=Error correction</b>	0: Auto. 1: 10% 2: 20% 3: 30% 4: 40% 5: 50%
<b>"DATA"</b>	The content of 2D Grid Matrix barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

# Example

Codes	Result
<pre> ↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ w2DB,GM,10,10,0,1,0,"12345678"↵ w2DB,GM,100,100,0,2,1,"12345678"↵ w2DB,GM,220,250,0,3,2,"12345678"↵ w2DB,GM,450,250,0,4,3,"12345678"↵ w2DB,GM,10,450,0,0,4,"12345678"↵ w2DB,GM,150,450,0,1,5,"12345678"↵ ↵ wPRINT,1,1↵ </pre>	

# w2DB, MPDF417

## Description

This command defines a 2D Micro PDF417 barcode.

## Format

w2DB, MPDF417,P1,P2,P3,P4,P5,P6,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation. 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Width</b>	Module width in dot
<b>P5=Height</b>	Module height in dot
<b>P6=Column</b>	0: Auto mode Number of columns: 1-4
<b>"DATA"</b>	The content of 2D Micro PDF417 barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

# Example

Codes	Result
<p>↵</p>	
<p>sLABEL,812,609↵</p>	
<p>sGAP,24,0↵</p>	
<p>sDIRECTION,1↵</p>	
<p>↵</p>	
<p>w2DB,MPDF417,50,50,0,2,5,0,"123456789"↵</p>	
<p>w2DB,MPDF417,50,150,0,3,6,1,"123456789"↵</p>	
<p>w2DB,MPDF417,50,280,0,4,7,2,"123456789"↵</p>	
<p>w2DB,MPDF417,50,400,0,5,12,3,"123456789"↵</p>	
<p>↵</p>	
<p>wPRINT,1,1↵</p>	

# w2DB,MQR

## Description

This command defines a 2D Micro QR barcode.


## Format

w2DB,MQR,P1,P2,P3,P4,P5,P6,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation. 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4= Version (Row * Column)</b>	0: Auto 1: 11 * 11 2: 13 * 13 3: 15 * 15 4: 17 * 17
<b>P5= Scaling</b>	Value: 1-10
<b>P6= Error correction</b>	0: Auto. 1: L 7% 2: M 15% 3: Q 25%
<b>"DATA"</b>	The content of 2D Micro QR barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ w2DB,MQR,400,250,0,4,10,0,"0123456789" ↵ ↵ wPRINT,1,1↵	

# w2DB,MX

## Description

This command defines a 2D MAXICODE barcode on the label format.

## Format

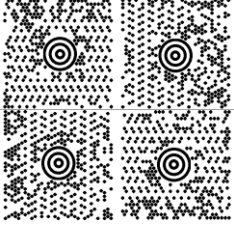
w2DB,MX,P1,P2,P3,P4[,P5][,P6][,P7],”DATA”

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation. 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Mode</b>	-1:Automatic selection mode 2 or 3 0: Automatic selection mode 2,3 or 4 2: US carrier message 3: International carrier message 4: Standard symbol 5: Enhanced symbol 6: Reader program
<b>p5=Class (Option)</b>	Class code (3 digits required).
<b>p6=Country code (Option)</b>	Country code (3 digits required).
<b>p7=Post code (Option)</b>	Mode 2: 5-digit + 4-digit number Mode 3: 6 alphanumeric post code
<b>”DATA”</b>	The content of 2D MAXICODE barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.



# Example

Codes	Result
<p>sLABEL,812,609<sup>␣</sup></p> <p>sGAP,24,0<sup>␣</sup></p> <p>sDIRECTION,1<sup>␣</sup></p> <p><sup>␣</sup></p> <p>w2DB,MX,450,301,0,2,300,840,06810,"DEMO 2 FOR USA MAXICODE"<sup>␣</sup></p> <p>w2DB,MX,430,301,1,3,300,863,107317,"DEMO 3 FOR CANADA MAXICODE"<sup>␣</sup></p> <p>w2DB,MX,430,299,2,4,"DEMO 4 FOR MAXICODE"<sup>␣</sup></p> <p>w2DB,MX,450,299,3,5,"DEMO 5 FOR MAXICODE"<sup>␣</sup></p> <p>wPRINT,1,1<sup>␣</sup></p>	

# w2DB,PDF417

## Description

This command defines a 2D PDF417 barcode on the label format.

## Format

w2DB, PDF417, P1, P2, P3, P4, P5, P6, P7, P8,"DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Correction level</b>	Value: 0-8
<b>P5=Width</b>	Module width in dot
<b>P6=Height</b>	Bar height in dot
<b>P7= Column</b>	0: Auto Maximum number of columns
<b>P8=Truncated flag</b>	0: Not truncated 1: Truncated
<b>"DATA"</b>	The content of 2D PDF417 barcode

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
sLABEL,812,609 <sup>Ⓔ</sup>	
sGAP,24,0 <sup>Ⓔ</sup>	
sDIRECTION,1 <sup>Ⓔ</sup>	
w2DB,PDF417,616,418,0,1,2,9,2,0,"12345678" <sup>Ⓔ</sup>	
w2DB,PDF417,40,265,0,7,2,6,6,0,"12345678" <sup>Ⓔ</sup>	
w2DB,PDF417,142,38,0,1,3,9,1,1,"12345678" <sup>Ⓔ</sup>	
w2DB,PDF417,423,105,0,1,4,12,1,0,"12345678" <sup>Ⓔ</sup>	
wPRINT,1,1 <sup>Ⓔ</sup>	

# w2DB,QR

## Description

This command defines a 2D QR barcode on the label format.

## Format

w2DB, QR, P1, P2, P3, P4, P5, "DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation. 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4= Version</b>	See below Version Table.
<b>P5= Scaling</b>	Value: 1-10
<b>P6= Error correction</b>	0: Auto 1: L 7% 2: M 15% 3: Q 30% 4: H 40%
<b>"DATA"</b>	The content of 2D QR barcode


Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

### Version Table (Row \* Column)

Value	Size	Value	Size
0	Auto	1	21 * 21
2	25 * 25	3	29 * 29
4	33 * 33	5	37 * 37
6	41 * 41	7	45 * 45
8	49 * 49	9	53 * 53
10	57 * 57	11	61 * 61
12	65 * 65	13	69 * 69
14	73 * 73	15	77 * 77
16	81 * 81	17	85 * 85

18	89 * 89	19	93 * 93
20	97 * 97	21	101 * 101
22	105 * 105	23	109 * 109
24	113 * 113	25	117 * 117
26	121 * 121	27	125 * 125
28	129 * 129	29	133 * 133
30	137 * 137	31	141 * 141
32	145 * 145	33	149 * 149
34	153 * 153	35	157 * 157
36	161 * 161	37	165 * 165
38	169 * 169	39	173 * 173
40	177 * 177		

## Example

Codes	Result
sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ w2DB,QR,400,250,0,4,6,4,"0123456789"↵ wPRINT,1,1↵	

# wBOX

## Description

This command defines a box shape on the label format.

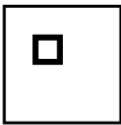
## Format

wBOX, P1, P2, P3, P4, P5, [P6]↵

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Width (dot)</b>	Value: 0 ~ maximum printing width
<b>P4=Height (dot)</b>	Value: 0 ~ maximum printing length
<b>P5=Line thickness (dot)</b>	Value: 0 ~ maximum printing width
<b>P6= round corner.</b>	Value: 0 ~ Width/2 (Optional) Default is 0.

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wBOX,50,50,200,200,5↵ wBOX,100,100,50,50,10↵ ↵ wPRINT,1↵	

# wDIAGONAL

## Description

This command defines a diagonal black line on the label format.


## Format

wDIAGONAL, P1, P2, P3, P4, P5<sup>␣</sup>

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width.
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length.
<b>P3=Width (dot)</b>	Value: 0 ~ maximum printing width.
<b>P4=Height (dot)</b>	Value: 0 ~ maximum printing length.
<b>P5= Line thickness (dot)</b>	Value: 0 ~ maximum printing width.

NOTE: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
<pre>␣ sLABEL,812,609␣ sGAP,24,0␣ sDIRECTION,1␣ ␣ wDIAGONAL,50,50,200,200,5␣ wPRINT,1␣</pre>	

# wELLIPSE

## Description

This command defines an ellipse shape on the label format.

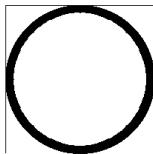
## Format

wELLIPSE, P1, P2, P3, P4, P5<sup>␣</sup>

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width.
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length.
<b>P3=Width (dot)</b>	Value: 0 ~ maximum printing width.
<b>P4=Height (dot)</b>	Value: 0 ~ maximum printing length.
<b>P5=Line thickness (dot)</b>	Value: 0 ~ maximum printing width.

NOTE: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
<pre>␣ sLABEL,812,609␣ sGAP,24,0␣ sDIRECTION,1␣ ␣ wELLIPSE,200,200,200,200,10␣ wLINE,200,200,1,200,1␣ wLINE,200,200,200,1,1␣ ␣ wPRINT,1␣</pre>	

# wLINE

## Description

This command defines a white or a black line on the label format.


## Format

wLINE, P1, P2, P3, P4, P5↵

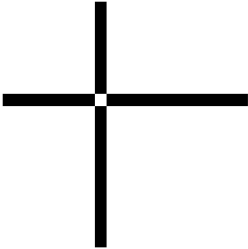
Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Width (dot)</b>	Value: 0 ~ maximum printing width
<b>P4=Height (dot)</b>	Value: 0 ~ maximum printing length
<b>P5= Color</b>	0=White 1=Black 2=Exclusive

NOTE: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wLINE,50,100,400,20,1↵ wLINE,50,200,400,20,1↵ wLINE,50,300,400,20,1↵ wLINE,200,50,20,400,0↵ ↵ wPRINT,1 ↵ ↵	



Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wLINE,50,200,400,20,2↵ wLINE,200,50,20,400,2↵ ↵ wPRINT,1 ↵ ↵	

# wIMAGE

## Description

This command defines an image data in the specified position.

## Format

wIMAGE, P1, P2, P3, P4, P5, P6, P7, DATA<sup>d</sup>

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Width (dot)</b>	Value: 0 ~ maximum printing width
<b>P4=Height (dot)</b>	Value: 0 ~ maximum printing length
<b>P5=Draw Mode</b>	0=OR 1=Overwrite 2=XOR
<b>P6=Data Type</b>	0=Bitmap Data 1=PCX File 2=BMP File 3=PNG File 4=P64 File
<b>P7=Size (BYTES)</b>	Data size in BYTES.
<b>DATA</b>	Image data

NOTE: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
-------	--------

↵  
sLABEL,812,609↵  
sGAP,24,0↵  
sDENSITY,8↵  
sSPEED,3↵  
sTHERMAL,1↵  
sDEVICE,1↵  
sDIRECTION,1↵  
wIMAGE,0,0,812,609,1,1,63336,.....  
wPRINT,1↵



12345678



0 000123 456784

# wPRINT

## Description

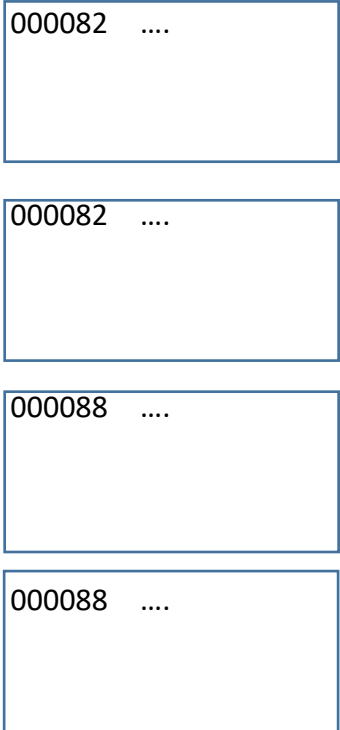
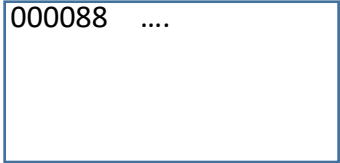
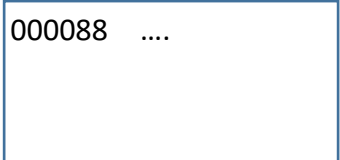
This command prints this label format that is stored in the image buffer.

## Format

wPRINT,P1[,P2]↵

Parameters	Description
<b>P1=Sets of labels</b>	Value: 1 – 999999
<b>P2=Copies of each set label</b>	Value: 1 – 999999

## Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDEVICE,1↵ sDIRECTION,1↵ ↵	
vCOUNT,C1,-6,FC,M10,"Test!!!" ↵ #C1="000088"↵ wTEXT,50,50,0,3,1,1,0,#C1"..."↵ ↵	
wPRINT,2,2↵	

# wREVERSE

## Description

This command reverses a region in the image buffer.


## Format

wREVERSE,P1,P2,P3,P4↵

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width.
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length.
<b>P3=Width (dot)</b>	Value: 0 ~ maximum printing width.
<b>P4=Height (dot)</b>	Value: 0 ~ maximum printing length.

NOTE: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDEVICE,1↵ sDIRECTION,1↵ ↵ wTEXT,50,50,0,1,5,5,0,"LABEL Test!!!" ↵ wREVERSE,0,40,800,100↵ ↵ wPRINT,1↵	

# wTEXT

## Description

This command prints text on label.

## Format

wTEXT, P1, P2, P3, P4, P5, P6, P7, "DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Font Name</b> or <b>"Font name"</b>	0: 8x12 fixed dot font 1: 10x16 fixed dot font 2: 12x20 fixed dot font 3: 14x24 fixed dot font 4: 16x28 fixed dot font 5: 20x36 fixed dot font 6: 24x44 fixed dot font 8: 16x16 BIG-5 9: 16x16 GB-2312 Other: Content of TTF (True type font) name
<b>P5= Horizontal multiplier expands</b>	Value: 1-10 ※TTF can upper than 10
<b>P6=Vertical multiplier expands</b>	Value: 1-10 ※TTF can upper than 10
<b>P7=Reverse Text</b>	0: Normal text 1: Reverse text image
<b>"DATA"</b>	Content of text string

### NOTE:

- 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi, 24 dots = 1 mm in 600 dpi.

- If there is any double quote (") within the text, please change it to \"
- If there is any backslash (\) within the text, please change it to \\

## Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDEVICE,1↵ sDIRECTION,1↵	Single-line Text Single-line Text Single-line Text Single-line Text <b>Single-line Text</b> Single-line Text Single-line Text
↵ wTEXT,190,23,0,0,3,3,0,"Single-line Text"↵ wTEXT,167,78,0,1,3,3,0,"Single-line Text"↵ wTEXT,205,144,0,2,2,2,1,"Single-line Text"↵ wTEXT,176,209,0,3,2,2,0,"Single-line Text"↵ wTEXT,23,281,0,4,3,3,1,"Single-line Text"↵ wTEXT,72,387,0,5,2,2,0,"Single-line Text"↵ wTEXT,40,479,0,"kaiu.ttf",30,30,0,"Single-line Text"↵ ↵ wPRINT,1↵	

# wMLTEXT

## Description

This command prints multi-line text on label.

## Format

wMLTEXT, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, "DATA"

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Rotation</b>	0: No rotation 1: Rotate right 90 degrees 2: Rotate right 180 degrees 3: Rotate right 270 degrees
<b>P4=Width</b>	Width of one line(Dot)
<b>P5=Font name</b>	Content of TTF (True type font) name
<b>P5= Horizontal multiplier expands</b>	Value: 1~
<b>P6=Vertical multiplier expands</b>	Value: 1~
<b>P8= Alignment</b>	0: Left 1: Right 2: Middle
<b>P9= Gap of row</b>	Value: 1~ (Dot)
<b>P10=Reverse Text</b>	0: Normal text 1: Reverse text image
<b>"DATA"</b>	Content of text string

NOTE:

- 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi, 24 dots = 1 mm in 600 dpi.
- If there is any double quote (") within the text, please change it to \"
- If there is any backslash (\) within the text, please change it to \\



## Example

Codes	Result
rWCTK↵ sLABEL,812,609↵ sGAP,24,0↵ sSENSOR,0↵ sTPHY,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDIRECTION,1↵  wMLTEXT,100,10,0,250,"標楷體.ttf",12,12,0,2,0,"\ 標楷體測試\" Testing...123 多行文字測試 563"↵  wPRINT,1↵ rWCTK↵	"標楷體測試" Testing...123 多行文字測試56 3

# 3

## File Management Commands

# wDELETE

## Description

This command deletes a file from the printer memory.

## Format

wDELETE,"FILENAME"↵

Parameters	Description
"FILENAME"	Name with extension name. Support full path. If no any path is specified, file will be deleted from RAM-Disk.

NOTE: Disk information references Appendix-Disk.

## Example

Codes	Result
wDELETE, "c:\test.pcx"↵	; Delete "test.pcx" file in c:\
wDELETE,"c:\WPL_Stuff\*.*"↵	; Delete all files in c:\ WPL_Stuff

# wLOAD

## Description

This command loads a file from the disk via the specified path.


## Format

wLOAD,"FILENAME"↵

Parameters	Description
"FILENAME"	Name with extension name. Support full path. If no any path is specified, file will be loaded from RAM-Disk.

NOTE: Disk information references Appendix-Disk.

## Example

Codes	Result
wSAVE,"Test.prn"↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wTEXT,50,50,0,1,5,5,1,"LABEL Test!!!" ↵ wPRINT,1↵ wSAVEEND↵ ↵ wLOAD,"Test.prn"↵	

# wLOADIMG

## Description

This command loads an image on the label format.

## Format

wLOADIMG, P1, P2, P3, "FILENAME"↵

Parameters	Description
<b>P1=X coordinate (dot)</b>	Value: 0 ~ maximum printing width
<b>P2=Y coordinate (dot)</b>	Value: 0 ~ maximum printing length
<b>P3=Draw Mode</b>	0=OR 1=Overwrite 2=XOR
<b>"FILENAME"</b>	Name with extension name. Support full path. If no any path is specified, file will be loaded from RAM-Disk.  ※Support image format: monochrome PCX/BMP.

NOTE: Disk information references Appendix-Disk.

## Example

Codes	Result
sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ wSAVE,4069,"WLT000",.....↵ wLOADIMG,110,224,0,"WLT000"↵ wPRINT,1↵ ↵	

# wMD

## Description

This command creates a folder in the printer's memory.

## Format

wMD,"FOLDERNAME"↵

Parameters	Description
"FILEPATH"	If no any path is specified, the folder will be created in the RAM-Disk.

NOTE: Disk information references Appendix-Disk.

## Example

Codes	Result
wMD,"e:\WPL_Stuff"↵	;Create the "WPL_Stuff" folder in SD Disk.

# wSAVE

## Description

This command saves current data in a file.

## Format


wSAVE, "FILENAME"↵

wSAVE, P1, "FILENAME" ,DATA↵

Parameters	Description
<b>P1=File Size (Option)</b>	If no file size is mentioned, it needs "wSAVEEND" command to save listed commands.
<b>"FILENAME"</b>	Name with extension name. Support full path. If no any path is specified, the data will be saved in the RAM-Disk.
<b>DATA</b>	

NOTE: Disk information references Appendix-Disk.

## Example 1 - Use wSAVEEND command

Codes	Result
wSAVE,"Test.prn"↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wTEXT,50,50,0,1,5,5,1,"LABEL Test!!!" ↵ wPRINT,1↵ wSAVEEND↵ ↵ wLOAD,"Test.prn"↵	

## Example 2 - File size is mentioned

Codes	Result
sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wSAVE,4069,"WLT000",.....↵ wLOADIMG,110,224,0,"WLT000"↵ wPRINT,1↵ ↵	Save Command Test !!!





# wSAVEB64

## Description

This command saves current data in a file.

## Format


wSAVE, "FILENAME"↵

wSAVE, P1, "FILENAME" ,DATA↵

Parameters	Description
<b>P1=File Size (Option)</b>	If no file size is mentioned, it needs "wSAVEEND" command to save listed commands.
<b>"FILENAME"</b>	Name with extension name. Support full path. If no any path is specified, the data will be saved in the RAM-Disk.
<b>DATA</b>	

NOTE: Disk information references Appendix-Disk.

## Example 1 - Use wSAVEEND command

Codes	Result
wSAVE,"Test.prn"↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wTEXT,50,50,0,1,5,5,1,"LABEL Test!!!" ↵ wPRINT,1↵ wSAVEEND↵ ↵ wLOAD,"Test.prn"↵	

## Example 2 - File size is mentioned

Codes	Result
sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wSAVE,4069,"WLT000",.....↵ wLOADIMG,110,224,0,"WLT000"↵ wPRINT,1↵ ↵	Save Command Test !!!

# wSAVEEND


## Description

This command used to save the above commands.

## Format

wSAVEEND↵

## Example

Codes	Result
<pre>wSAVE,"Test.prn"↵ sLABEL,812,609↵ sGAP,24,0↵ sDIRECTION,1↵ ↵ wTEXT,50,50,0,1,5,5,1,"LABEL Test!!!" ↵ wPRINT,1↵ wSAVEEND↵ ↵ wLOAD,"Test.prn"↵</pre>	

# 4

## Defined Variable Commands

# vCOUNT

## Description

This command defines a counter. This counter can be a variable.

## Format

vCOUNT,Cx,P2[,P3][,P4][,P5][,P6][“PROMPT”]↵

Parameters	Description
<b>Cx=Counter Number</b>	Cx, x=1~10
<b>P2=Step</b>	+ or -, 1~9 digit
<b>P3=Step Method</b> (Option)	S0=Numeric (Default) S1=Numeric + Alphabet
<b>P4=Maximum digits</b> (Option)	Mxx, xx=1~29, Default is 29.
<b>P5=Field Justification</b> (Option)	FL=Left (Default) FR=Right FC=Center
<b>P6=Request once</b> (Option)	R0=Every time (Default) R1=Only once
<b>PROMPT</b> (Option)	Show a text on LCD

## Set Variable

#Cx="VALUE"↵


Parameters	Description
<b>Cx</b>	Counter number x=1~10
<b>VALUE</b>	String of ASCII

## Load Variable

#Cx

Parameters	Description
<b>Cx</b>	Counter number x=1~10

# Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDEVICE,1↵ sDIRECTION,1↵	<div data-bbox="938 344 1273 501" style="border: 1px solid black; padding: 5px;">000076 ... ↵</div> <div data-bbox="938 546 1273 703" style="border: 1px solid black; padding: 5px;">000082 ... ↵</div>
↵ vCOUNT,C1,-6,FC,M10,"Test!!!" ↵ #C1="000088"↵ wTEXT,50,50,0,3,1,1,0,#C1"..."↵	<div data-bbox="938 741 1273 898" style="border: 1px solid black; padding: 5px;">00088 ... ↵</div>
↵ wPRINT,3,1↵	<p data-bbox="1038 972 1166 994" style="text-align: center;">Feed Direction</p> 

# vDATE

## Description

This command defines a variable that sets/loads the date via the Real Time Clock (RTC).

## Format

vDATE,Dx[,P2][,"PROMPT"]↵

Parameters	Description
<b>P1=Date Number</b>	Dx, x=1~10
<b>P2=Initial value</b> (Option)	0=Load value from RTC (Default) 1=User Input for LCD mode
<b>PROMPT</b> (Option)	Show a text on LCD

## Set Variable

#Dx=" P1/P2/P3"↵

Parameters	Description
<b>Dx</b>	Date number x=1~10
<b>P1</b>	4 digits - Year
<b>P2</b>	Month
<b>P3</b>	Day

## Load Variable

#Dx(P1[ | P2 | P3])[+P4]

Parameters	Description
<b>Dx</b>	Date number x=1~10
<b>P1,P2,P3=FORMAT</b> <b>P2,P3 (Option)</b>	P1,P2,P3 can be below: y2= Year displayed as 2 digits (95) y4= Year displayed as 4 digits (1995) me= Month displayed as 3 letters (JAN) mn= Month displayed as 2 digits (01) dd= Day displayed as 2 digits (15)



ju= Julian day (1)  
 j3= Julian day as 3 digits (001)  
 wn= Week of year (1)  
 w2= Week of year as 2 digits (01)

| Separator character. The separator may be any ASCII character value between 032 and 063. The separator character is printed between the results of each supplied parameter.

**P4=Appending Date (Option)** 1~9999

## Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDEVICE,1↵ sDIRECTION,1↵ ↵ vDATE,D1,0,"Date1:" ↵ vDATE,D2,0,"Date2:" ↵ vDATE,D3,0,"Date3:" ↵ ↵ #D1="2015/7/28"↵ ↵ wTEXT,50,50,0,3,1,1,0,"Date1:"#D1(dd/mn/y4)+2" DAY"↵ wTEXT,50,100,0,3,1,1,0,"Date2:"#D2(y2-me-dd)-10" DAY"↵ wTEXT,50,150,0,3,1,1,0,"Date3:"#D3(y4,me,ju)" DAY"↵ wTEXT,50,200,0,3,1,1,0,"Date4:"#D3(y4,me,j3)" DAY"↵ wTEXT,50,250,0,3,1,1,0,"Date5:"#D3(y2,mn,wn)" DAY"↵ wTEXT,50,300,0,3,1,1,0,"Date6:"#D3(y2,mn,w2)" DAY"↵ ↵ wPRINT,1,1↵	<b>Date1:30/07/0015 DAY</b> <b>Date2:15-NOV-01 DAY</b> <b>Date3:2015,NOV,315 DAY</b> <b>Date4:2015,NOV,315 DAY</b> <b>Date5:15,11,46 DAY</b> <b>Date6:15,11,46 DAY</b>

# vTIME

## Description

This command defines a variable that can load and set time in the Real Time Clock (RTC).

## Format

vDATE,Tx[,P2][,"PROMPT"]<sup>Ⓔ</sup>

Parameters	Description
<b>P1=Time number</b>	Tx, x=1~10
<b>P2=Initial value</b> (Option)	0=Load from RTC (Default) 1=User Input for LCD mode
<b>PROMPT</b> (Option)	Show a text on LCD

## Set Variable

#Tx=" P1:P2:P3"<sup>Ⓔ</sup>

Parameters	Description
<b>Tx</b>	Time number x= 1~10
<b>P1</b>	Hour
<b>P2</b>	Minute
<b>P3</b>	Second

## Load Variable

#Tx(P1[|P2|P3])

Parameters	Description
<b>Tx</b>	Time number. x= 1~10
<b>P1,P2,P3=FORMAT</b> <b>P2,P3 (Option)</b>	P1, P2, P3 can be below HH= Hours displayed as 2 digits in 12 hour clock with "AM"/"PM" (02 PM) hh= Hours displayed as 2 digits in 24 hour clock (14) mm= Minutes displayed as 2 digits (01)

ss= Seconds displayed as 2 digits (01)

Separator character. The separator may be any ASCII character value between 032 and 063. The separator character is printed between the results of each supplied parameters.

## Example

Codes	Result
<pre> ↵ sLABEL,812,609↵ sGAP,24,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDEVICE,1↵ sDIRECTION,1↵ ↵ vTIME,T1,0,"Time1:" ↵ vTIME,T2,0,"Time2:" ↵ #T1="06:30:10"↵ #T2="22:56:33"↵ wTEXT,50,50,0,3,1,1,0,"Time:"#T1(HH:mm:ss) ↵ wTEXT,50,100,0,3,1,1,0,"Time:"#T2(hh#mm#ss) ↵ ↵ wPRINT,1↵ </pre>	<pre> <b>Time:06:30:10 AM</b> <b>Time:22#56#33</b> </pre>

# vVAR

## Description

This command defines a variable.

## Format

vVAR,Vx[,P2][,P3][,P4]↵

Parameters	Description
<b>P1=Variable number</b>	x= 1~10
<b>P2=Maximum digits</b> (Option)	Mxx, xx=1~29, Default is 29.
<b>P3=Field Justification</b> (Option)	FL=Left (Default) FR=Right FC=Center
<b>P4=Request once</b> (Option)	R0=Every time (Default) R1=Only once
<b>PROMPT</b> (Option)	Show a text on LCD

## Set Variable

#Vx="VALUE"↵

Parameters	Description
<b>Vx</b>	Variable number. x= 1~10
<b>VALUE</b>	String of ASCII

## Load Variable

#Vx

Parameters	Description
<b>Vx</b>	Variable number x=1~10

# Example

Codes	Result
↵ sLABEL,812,609↵ sGAP,24,0↵ sDENSITY,8↵ sSPEED,3↵ sTHERMAL,1↵ sDEVICE,1↵ sDIRECTION,1↵ ↵ vVAR,V1↵ #V1="Variable Test"↵ wTEXT,50,50,0,3,1,1,0,"VAR:"#V1↵ ↵ wPRINT,1↵	<b>VAR:Variable Test</b>

# #SYS

## Description

This command can get the variable from the settings of printer.

## Load Variable

#SYS (VARNAME)

VARNAME	Description
LABS	Printed labels (pcs)
LENS	Printed length (mm)
CUTS	Cut times

## Example

Codes	Result
␣ sLABEL,812,609␣ sGAP,24,0␣ sDENSITY,8␣ sSPEED,3␣ sTHERMAL,1␣ sDEVICE,1␣ sDIRECTION,1␣ ␣ wTEXT,50,50,0,3,1,1,0,"Printed LABS:""#SYS(LABS)" "s"␣ wTEXT,50,150,0,3,1,1,0,"Printed LENS:""#SYS(LENS)" "mm"␣ wTEXT,50,250,0,3,1,1,0,"Printed CUTS:""#SYS(CUTS)" "times"␣ ␣ wPRINT,1␣	<b>Printed LABS:32 's</b>  <b>Printed LENS:352 mm</b>  <b>Printed CUTS:5 times</b>

# 5

## Retrieve Setting Commands

# rBACKOFFSET

## Description

This command can retrieve the setting of back feed offset.

## Format

rBACKOFFSET,P1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET48:v1

Field	Description
v1	Back offset distance (dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve BACKOFFSET setting via USB channel.

```
rBACKOFFSET,U↵
```

Result:

```
RET48:24
```



# rBLINE

## Description

This command can retrieve black line setting of label.

## Format

rBLINE,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET08:v1,v2

Field	Description
v1	The height of black line(dot)
v2	The extra feeding length of label(dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve black line setting via USB channel.

```
rBLINE,U↵
```

Result:

```
RET08:24,0
```

# rBTNAME

## Description

This command can retrieve the device name..

## Format

rBTNAME,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET08:v1,v2

Field	Description
v1	The height of black line(dot)
v2	The extra feeding length of label(dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve black line setting via USB channel.

```
rBLINE,U↵
```

Result:

```
RET08:24,0
```



# rBTPINCODE

## Description

This command can retrieve black line setting of label.

## Format

rBLINE,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET08:v1,v2

Field	Description
v1	The height of black line(dot)
v2	The extra feeding length of label(dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve black line setting via USB channel.

```
rBLINE,U↵
```

Result:

```
RET08:24,0
```

# rBUZZER

## Description

This command can retrieve the settings of buzzer.

## Format

rBUZZER,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET09:v1

Field	Description
v1	Buzzer level (0~3)

## Example

Retrieve BUZZER setting via USB channel.

```
rBUZZER,U↵
```

Result:

```
RET09:1
```

# rCUTACTION

## Description

This command can retrieve the settings of cutter.

## Format

rCUTACTION,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET49:v1

Field	Description
v1=Cutting settings	0: Every piece of label 1: Depended on the quantity setting to run the cutter once 2: The last piece of label

## Example

Retrieve CUTACTION setting via USB channel.

rCUTACTION,U↵

Result:

RET49:2

# rCUTMODE

## Description

This command can retrieve the current cutting mode.

## Format

rCUTMODE,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET52:v1

Field	Description
v1=Cutting mode	0: Full cut 1: Half cut

## Example

Retrieve CUTMODE setting via USB channel.

rCUTMODE,U↵

Result:

RET52:0

# rCUTNUM

## Description

This command can retrieve the settings of how many printed labels to run the cutter once.

## Format

rCUTNUM,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET12:v1

Field	Description
v1	Set a number of printed labels per cut

## Example

Retrieve CUTNUM setting via USB channel.

```
rCUTNUM,U↵
```

Result:

```
RET12:,2
```



# rDEMAND

## Description

This command can check if the demand function is enabled.

## Format

rDEMAND,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET13:v1

Field	Description
v1	0: Disable this function. 1: Enable this function.

## Example

Retrieve Tap setting via USB channel.

```
rDEMAND,U↵
```

Result:

```
RET13:1
```

# rDENSITY

## Description

This command can retrieve the settings of printing density.

## Format

rDENSITY,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET14:v1

Field	Description
v1	0~15 : Density value

## Example

Retrieve density setting from the printer via USB channel.

```
rDENSITY,U↵
```

Result:

```
RET14:8
```

# rDEVICE

## Description

This command can retrieve the current setting of connected device, there are four modes including Normal, Tear, Peel, and Cut modes.

## Format

rDEVICE,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET15:v1

Field	Description
v1	0:Normal mode 1:Tear mode 2:Peel mode 3:Cut mode

## Example

Retrieve DEVICE setting via USB channel.

```
rDEVICE,U↵
```

Result:

```
RET15:0
```

# rDIR

## Description

This command can check all files from the specified path, and all files can be separated by the "|" symbol.

## Format

rDIR,p1,"FILENAME"

Parameters	Description
p1=Port	R=RS232, U=USB
"PATHNAME"	If there is no any path, the default of path is "c:\"

## Return

RET33:....|....|....

Field	Description
"FILENAME"	The file name from the specified path
	Separator character

## Example

All files can be reviewed by the specified path via USB channel.

rDIR,U,"c:\"

Result:

RET33:Test.prn|Box.prn|Line.prn

# rDIRECTION

## Description

This command can retrieve the setting of printing direction.

## Format

rDIRECTION,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET16:v1

Field	Description
v1	Printing direction 0=Printing from top of image buffer 1=Printing from bottom of image buffer

## Example

Retrieve the setting of printing direction via USB channel.

```
rDIRECTION,U↵
```

Result:

```
RET16:1
```

# rDPI

## Description

This command can retrieve the setting of DPI (Dots per Inch).

## Format

rDPI,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET37:v1

Field	Description
v1	DPI(Dots per inch) of TPH 203=203 DPI 300=300 DPI 600=600 DPI

## Example

Retrieve the printing setting of DPI via USB channel.

rDPI,U↵

Result:

RET37:203

# rDISKINFO

## Description

This command can retrieve the disk information.

## Format

rDISKINFO,p1↵

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET69:v1,v2,v3,v4,v5,v6↵

Field	Description
v1	RAM Disk total size (K bytes)
v2	RAM Disk used size (K bytes)
v3	Flash Disk total size (K bytes)
v4	Flash Disk used size (K bytes)
v5	SD Disk total size (K bytes)
v6	SD Disk used size (K bytes)

## Example

Retrieve the Disk information via USB channel.

```
rDISKINFO,U↵
```

Result:

```
RET69: 1008,8,3048,72,3873344,35236↵
```

# rFEEDKEY

## Description

This command can retrieve the current setting of FEED KEY.

## Format

rFEEDKEY,p1

Parameters	Description
p1=Port	R=RS232, U=USB

## Return

RET17:v1

Field	Description
v1	FEED KEY Type 0=Normal operation(Default) 1=Reprint the last label 2=Ignore the feed button

## Example

Retrieve FEEDKEY setting via USB channel.

```
rFEEDKEY,U↵
```

Result:

```
RET17:0
```



# rGAP

## Description

This command can retrieve the gap setting.

## Format

rGAP,P1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET18:v1,v2

Field	Description
V1	Gap length (dot)
V2	The length after gap (dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve gap setting via USB channel.

```
rGAP,U↵
```

Result:

```
RET18:24,0
```

# rKEYBOARD

## Description

Retrieve Keyboard Country setting.

## Format

rKEYBOARD,P1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET11:v1

Field	Description
v1	0:US keyboard 1:German keyboard

## Example

Retrieve keyboard country setting.

rKEYBOARD,U↵

Result:

**RET11:0**

# rLABEL

## Description

This command can retrieve the setting of label size.

## Format

rLABEL,P1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET19:v1,v2

Field	Description
v1	Label width (dot)
v2	Label height (dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve label setting via USB channel.

```
rLABEL,U↵
```

Result:

```
RET19:808,600
```

# rLCD

## Description

This command can retrieve the setting of LCD Language.

## Format

rLCD,P1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET20:v1

Field	Description
v1	0:English 1:Traditional Chinese 2: Simplified Chinese 3: Spanish 4: Italian 5: Vietnam 6: Korea

## Example

Retrieve LCD Language setting via USB channel.

rLCD,U↵

Result:

**RET20:0**

# rMIRROR

## Description

This command can retrieve the setting of Mirror Image.

## Format

rMIRROR,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET21:v1

Field	Description
V1	0:Normal 1:Mirror

## Example

Retrieve Mirror setting via USB channel.

```
rMIRROR,U↵
```

Result:

```
RET21:0
```

# rMODEL

## Description

This command can retrieve the model name of printer.

## Format

rMODEL,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET22:"MODEL NAME"

Field	Description
"MODEL NAME"	Model name of printer.

## Example

Retrieve Model name of printer via USB channel.

```
rMODEL,U↵
```

Result:

```
RET22: LP423A
```

# rNETIP

## Description

This command can retrieve the setting of Ethernet.

## Format

rNETIP,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET53:V1,V2,V3,V4

Field	Description
V1=IP mode	0:DHCP enable 1:Static IP
V2=IP	IP address
V3=Mask	Subnet mask
V4=Gateway	Default gateway

## Example

Retrieve NETIP setting via USB channel.

```
rNETIP,U↵
```

Result:

```
RET53:1,"192.168.0.1","192.168.25.1","255.255.255.0"
```

# rNETMAC

## Description

This command can retrieve the MAC address of printer.

## Format

rNETMAC,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET56:"MAC Address"

Field	Description
"MAC Address"	MAC address

## Example

Retrieve MAC address of printer via USB channel.

```
rNETMAC,U↵
```

Result:

```
RET56:001B82FF0337
```



# rNETNAME

## Description

This command can retrieve the printer's server name.

## Format

rNETNAME,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET55:"Server name"

Field	Description
"Server name"	The specific name of printer server.

## Example

Retrieve NETNAME setting via USB channel.

```
rNETNAME,U↵
```

Result:

```
RET55:"Label Printer"
```

# rNETPORT

## Description

This command can retrieve the port number of Ethernet. It will be used for printer driver under the RAW data mode.

## Format

rNETPORT,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET54:v1

Field	Description
V1	Port number

## Example

Retrieve NETPORT setting via USB channel.

```
rNETPORT,U↵
```

Result:

```
RET54:9100
```

# rOFFSET

## Description

This command can retrieve the feeding length after printing a label.

## Format

rOFFSET,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET23:v1

Field	Description
V1	Offset Distance (dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve OFFSET setting via USB channel.

```
rOFFSET,U↵
```

Result:

```
RET23:24
```

# rORIGIN

## Description

This command can retrieve the current coordinates of origin (X axis, Y axis).

## Format

rORIGIN,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET24:v1,v2

Field	Description
V1	X axis displacement (Dot)
V2	Y axis displacement (Dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve ORIGIN coordinates via USB channel.

```
rORIGIN,U↵
```

Result:

```
RET24:8,8
```

# rRS232

## Description

This command can retrieve the current setting of serial port.

## Format

rRS232,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET25:v1,v2,v3,v4

Field	Description
V1=Baud rate	Baud rate, available baud rates are listed below: 115200 (Default) 38400 19200 9600 4800 2400 1200
V2= Data bits	7: 7 bits 8: 8 bits
V3= Parity	0: No parity check 1: Even parity check 2: Odd parity check
V4= Stop bits	1: 1 bit 2: 2 bits

## Example

Retrieve RS232 setting via USB channel.

rRS232,U↵

Result:

RET25:115200,8,0,1

# rRTCDATE

## Description

This command can retrieve the Real Time Clock (DATE).

## Format

rRTCDATE,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET26:v1,v2,v3

Field	Description
V1	Year(4 Digits)
V2	Month(2 Digits)
V3	Day(2 Digits)

## Example

Retrieve RTCDATE setting via USB channel.

```
rRTCDATE,U↵
```

Result:

```
RET26:2015,12,02
```

# rRTCTIME

## Description

This command can retrieve Real Time Clock (TIME) in 24 hours format.

## Format

rRTCTIME,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET27:v1,v2,v3

Field	Description
V1	Hour(2 Digits)
V2	Minute(2 Digits)
V3	Second(2 Digits)

## Example

Retrieve RTCTIME setting via USB channel.

```
rRTCTIME,U↵
```

Result:

```
RET27:20,58,30
```

# rSENSOR

## Description

This command can retrieve current sensor type.

## Format

rSENSOR,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET28:v1

Field	Description
v1	Sensor type 0: Reflective gap sensor 1: Transmissive gap sensor

## Example

Retrieve current sensor type via USB channel.

```
rSENSOR,U↵
```

Result:

```
RET28:0
```



# rSN

## Description

This command can retrieve the serial number of connected printer.

## Format

rSN,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET01:SN

Field	Description
SN	Serial number

## Example

Retrieve SN setting via USB channel.

rSN,U↵

Result:

RET01:W1234567890

# rSPEED

## Description

This command can retrieve the setting of printer speed.

## Format

rSPEED,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET30:v1

Field	Description						
V1	IPS (Inch per second) <table border="1"><thead><tr><th>Model name</th><th>Support</th></tr></thead><tbody><tr><td>LP423(203 dpi)</td><td>1~5 IPS</td></tr><tr><td>LP433(300 dpi)</td><td>1~4 IPS</td></tr></tbody></table>	Model name	Support	LP423(203 dpi)	1~5 IPS	LP433(300 dpi)	1~4 IPS
Model name	Support						
LP423(203 dpi)	1~5 IPS						
LP433(300 dpi)	1~4 IPS						

## Example

Retrieve SPEED setting from the printer via USB channel.

rSPEED,U $\leftarrow$

Result:

RET30:3

# rTHERMAL

## Description

This command can retrieve the current printing mode (Direct thermal or thermal transfer).

## Format

rTHERMAL,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET31:v1

Field	Description
V1	Thermal mode 0:Direct thermal 1:Thermal transfer (Default)

## Example

Retrieve THERMAL setting via USB channel.

```
rTHERMAL,U↵
```

Result:

```
RET31:1
```

# rTPHY

## Description

This command can retrieve the offset setting of start line.

## Format

rTPHY,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET29:v1

Field	Description
V1	Start line offsets (dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Retrieve the offset setting of start line

```
rTPHY,U↵
```

Result:

```
RET29:24
```

# rVERB

## Description

This command can retrieve the boot version of connected printer.

## Format

rVERB,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET04:"BOOT VERSION"

Field	Description
"BOOT VERSION"	Boot version

## Example

Retrieve Boot version via USB channel.

```
rVERB,U↵
```

Result:

```
RET04: 1.0-2015/09/05
```

# rVERK

## Description

This command can retrieve Kernel Version of connected printer.

## Format

rVERK,p1

Parameters	Description
P1=Port	R=RS232, U=USB

## Return

RET06:"KERNEL VERSION"

Field	Description
"KERNEL VERSION"	Kernel version

## Example

Retrieve Kernel version via USB channel.

rVERK,U↵

Result:

RET06: BINVER=LP,1.0.13-2015/11/11,

# 6

## Set Commands

# sBACKOFFSET

## Description

This command is used to adjust label start position, and can pull back label to the specified position before each run of printing, suitable for tearing, peel-off, and cutting modes.

## Format

sBACKOFFSET,P1

Parameters	Description
P1	Back offset distance (dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

sBACKOFFSET,24



# sBLINE

## Description

This command can set the height of black line, and extra feeding distance after running each black line.

## Format

sBLINE,P1,P2

Parameters	Description
P1	The height of black line(dot)
P2	The extra feeding length(dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

- Set the height of black line (3 mm) and offset (0 mm) in 203 dpi printer.

```
sBLINE,24,0↵
```

- Set continuous label.

```
sBLINE,0,0↵
```

# sBTNAME

## Description

This command sets the device name.

## Format

sBTNAME,"DevName"

Parameters	Description
DevName	Device name of BT module(0~16 character)

## Example

sBTNAME,"Module\_1"↵

# sBTPINCODE

## Description

This command sets the security pin code.

## Format

sBTPINCODE,"PASSWORD"

Parameters	Description
PASSWORD	Sets the security pin code (6 character)

## Example

sBTNAME,"123456" ↵

# sBUZZER

## Description

This command can adjust the level of buzzer.

## Format

sBUZZER,P1

Parameters	Description
P1	Buzzer level (0~3)

## Example

sBUZZER,2↵

# sCODEPAGE

## Description

Set Character Set Selection (CodePage)

## Format

sCODEPAGE,P1[,P2]

Parameters	Description		
P1=Code page	8 bit data		
	P1	Code page	Description
	0	DOS 437	English - US
	1	DOS 850	Latin 1
	2	DOS 852	Latin 2 (Cyrillic II/Slavic)
	3	DOS 860	Portuguese
	4	DOS 863	French Canadian
	5	DOS 865	Nordic
	6	DOS 857	Turkish
	7	DOS 861	Icelandic
	8	DOS 862	Hebrew
	9	DOS 855	Cyrillic
	10	DOS 866	Cyrillic CIS 1
	11	DOS 737	Greek
	12	DOS 851	Greek 1
	13	DOS 869	Greek 2
	14	Windows 1252	Latin 1
	15	Windows 1250	Latin 2
	16	Windows 1251	Cyrillic
	17	Windows 1253	Greek
	18	Windows 1254	Turkish
19	Windows 1255	Hebrew	
20	VISCII	Vietnam	

7 bit data	
P2	Description
0	USA
1	British
2	German
3	French
4	Danish
5	Italian
6	Spanish
7	Swedish
8	Swiss

**P2= Number of data bits**    7: 7 bit data  
    8: 8 bit data

---

## Example

- Set Code page.  
    sCODEPAGE,0,8

# sCUTACTION

## Description

This command can adjust the setting of cutting mode.

## Format

sCUTACTION,P1

Parameters	Description
<b>P1=Cutting mode</b>	0: Each piece of label 1: Depended on printed label quantity to run the cutter once 2: Cutting after printing the last piece of label

## Example

sCUTACTION,1↵

# sCUTMODE

## Description

This command sets the cutting mode.

## Format

sCUTMODE,P1

Parameters	Description
<b>P1=Cutting mode</b>	0: Full cut 1: Half cut

## Example

sCUTMODE,0↵



# sCUTNUM

## Description

This command sets a number of labels printed out to run the cutter once.

## Format

sCUTNUM,P1

Parameters	Description
P1	Set a number of labels come out per cut

## Example

sCUTNUM,4↵

# sDEMAND

## Description

This command can launch the function which makes user to press the FEED key to continue each label printing.

## Format

sDEMAND,P1

Parameters	Description
<b>P1</b>	0: Disable this function. 1: Enable this function.

## Example

sDEMAND,1↵

# sDENSITY

## Description

This command can set the density for printing effect.

## Format

sDENSITY,P1

Parameters	Description
P1	0~15 : Density level

## Example

sDENSITY,5↵

# sDEVICE

## Description

This command can provide user to select device mode, there are four modes including Normal, Tear, Peel, and Cut modes.

## Format

sDEVICE,P1

Parameters	Description
<b>P1</b>	0:Normal mode 1:Tear mode 2:Peel mode 3:Cut mode

## Example

sDEVICE,3↵

# sDIRECTION

## Description

This command can set the printing direction.

## Format

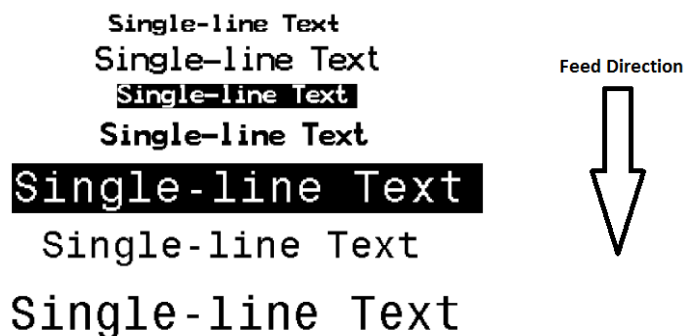
sDIRECTION,P1

Parameters	Description
<b>P1=Print direction</b>	0=Printing from top of image buffer. 1=Printing from bottom of image buffer.

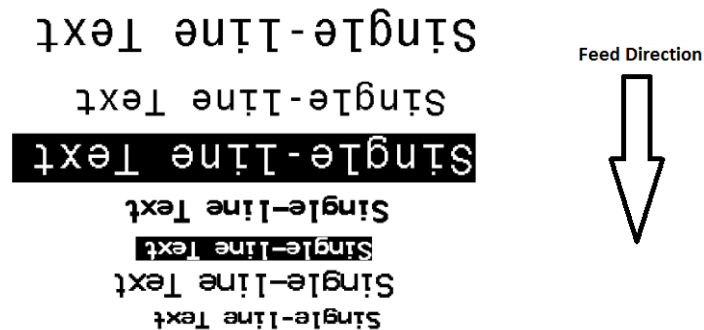
## Example

- Set print direction from top of image buffer.

```
sDIRECTION,1
```



```
sDIRECTION,0
```



# sFEEDKEY

## Description

This command can set the function of the Feed key which is located on the panel of printer.

## Format

sFEEDKEY,P1

Parameters	Description
<b>P1= Feed key function</b>	0=Feeding one label (Default) 1=Reprint the last label 2=no function of Feed key

## Example

sFEEDKEY,0↵

# sGAP

## Description

This command defines gap and offset distance between two labels.

## Format

sGAP,P1,P2

Parameters	Description
<b>P1=Gap Distance (dot)</b>	The gap distance between two labels
<b>P2=Offset Distance (dot)</b>	P2 ≤ each Label Length
P1=0, P2=0 for continuous label	

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

- Set gap distance (3 mm) and offset (0 mm) in 203 dpi printer.

```
sGAP,24,0↵
```

- Set continuous label.

```
sGAP,0,0↵
```

# sGPI

## Description

This command to receive the GPIO signals from external controlling devices.

## Format

sGPI,P1,P2,P3,P4,P5

Parameters	Description
<b>P1=Port number</b>	0~1 Two dedicated inputs are available for the desired control functions.
<b>P2=Signal state</b>	0~3 0=When a low level signal received, will activate. 1=When a high level signal received, will activate. 2= When a negative pulse signal received, will activate. 3= When a positive pulse signal received, will activate.
<b>P3= Pulse</b>	Pulse width. Ignored for level-type signals. Unit: 1 millisecond.
<b>P4= Function</b>	0~4 0= Feeding an empty label. 1= Feed label with the specified length (dot). 2= Print labels. 3= Activate cutter. 4=Pause.
<b>P5= Parameter 1</b>	Parameter for function

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.



## Example

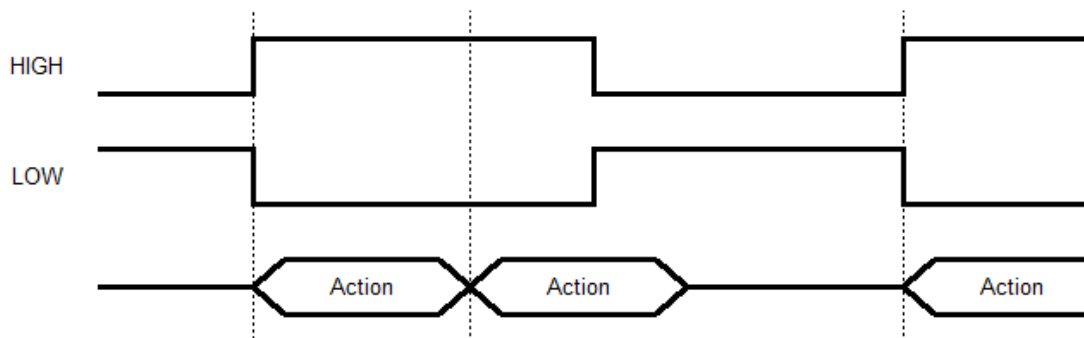
- Set Port0, When a low level signal received, will printing 5 papers.

sGPI,0,0,0,2,5↵

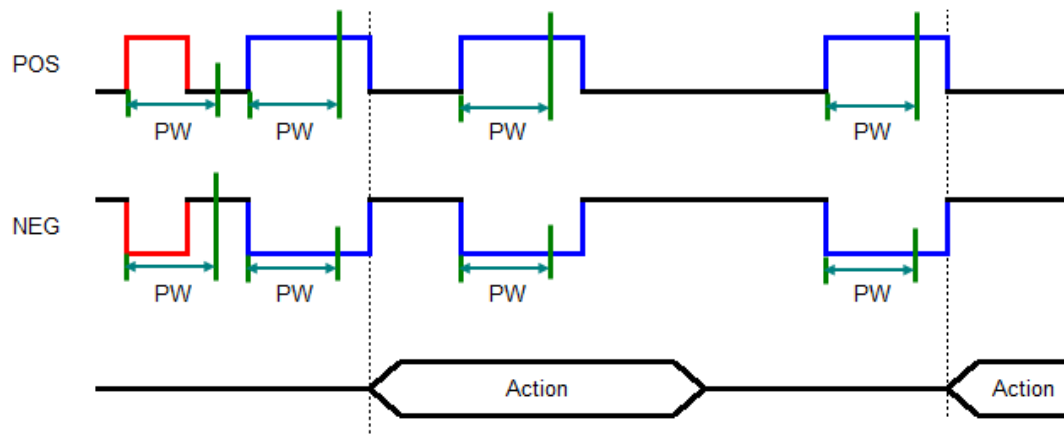
- Set Port1, when a positive pulse signal received (10 ms), will feeding 500 dots.

sGPI,1,3,10,1,500↵

### GPI Level Signal : Continuous action. (Host to printer.)



### GPI Pulse Signal : A pulse is an action. (Host to printer.)



# sGPO

## Description

This command to send out the GPIO signals by the printer.

## Format

sGPO,P1,P2,P3,P4,P5,P6,P7

Parameters	Description
<b>P1=Port number</b>	0~2 Three dedicated outputs are available for the desired function conditions.
<b>P2=Signal state</b>	0~3 0= Goes the low level signal when the following function condition is detected. 1= Goes the high level signal when the following function 2= Goes the negative pulse signal when the following function condition is detected. 3= Goes the positive pulse signal when the following function condition is detected.
<b>P3=Delay 0</b>	After detecting the following function condition, the printer will wait this period of time before sending out the output signal. Unit: 1 millisecond.
<b>P4=Pulse 0</b>	Pulse width corresponding to the function condition becoming. (Ignored for level-type signals.) Unit: 1 millisecond.
<b>P5=Delay 1</b>	After detecting the following function condition is ending, the printer will wait this period of time before sending out the output signal. Unit: 1 millisecond.
<b>P6=Pulse 1</b>	Pulse width corresponding to the function

---

condition is ending.  
(Ignored for level-type signals.)  
Unit: 1 millisecond.

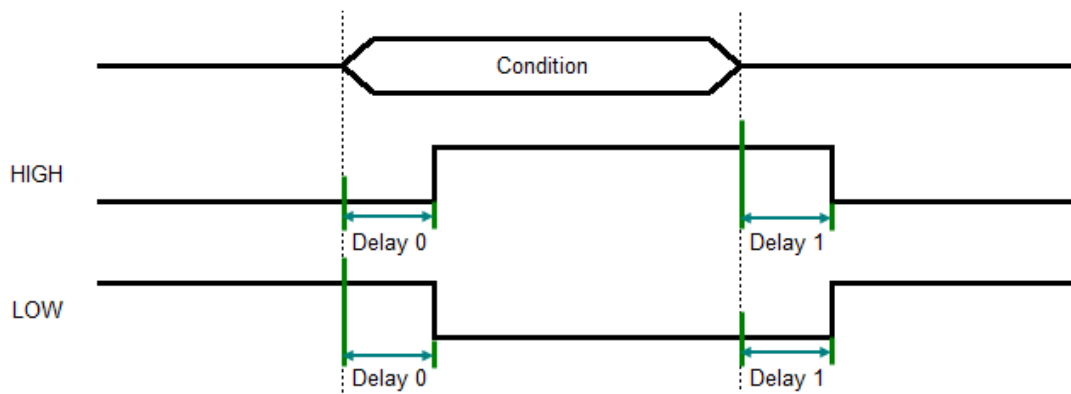
**P7= Function**

0~9  
0:Head open  
1:Ribbon out  
2:Gap out  
3:Label out  
4:Cutter Jam.  
5:No cutter  
6:Ribbon error  
7:Pause  
8:Label error  
9:No peeler

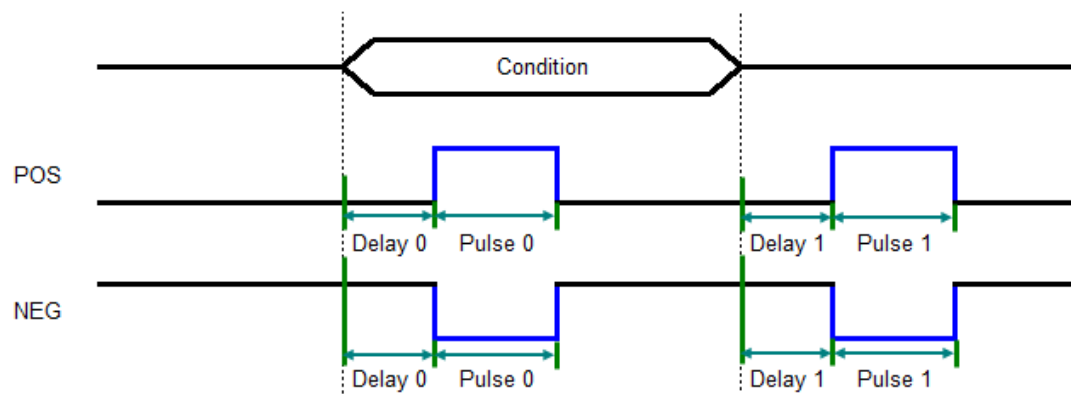
## Example

- Set Port1, Goes the negative pulse signal (20 ms) when the “Head open” condition is detected.  
sGPO,1,2,20,20,20,20,0↵
- Set Port2, Goes the high level signal when the “Gap out” condition is detected.  
sGPO,2,1,20,20,20,20,2↵

**GPO Level Signal : Continuous condition. (Printer to host.)**



**GPO Pulse Signal : A pulse is a condition. (Printer to host.)**



# sGPOLEV

## Description

This command to setting output level.

## Format

sGPOLEV,0,1↵

Parameters	Description
<b>P1=Port number</b>	0~2 Three dedicated outputs are available for the desired function conditions.
<b>P2=Level</b>	0~1 0=Low level 1=Hi level

## Example

- Set Port 0 to Hi level.  
sGPOLEV,0,1↵

# sHEADOPEN

## Description

Disable or enable “Head Open” detect.

## Format

sHEADOPEN,P1<sup>Ⓛ</sup>

Parameters	Description
<b>P1</b>	0:Disable Head Open 1:Enable Head Open

## Example

Enable Head Open

sHEADOPEN,1<sup>Ⓛ</sup>

# sKEYBOARD

## Description

Set Keyboard Country

## Format

sKEYBOARD,P1

Parameters	Description
P1=Keyboard country	0:US Keyboard 1:German Keyboard

## Example

sKEYBOARD,1↵

# sLABEL

## Description

This command defines the size of label.

## Format

sLABEL,P1,P2

Parameters	Description
P1	Label width(dot)
P2	Label length(dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

- Set label width (101 mm) and length (75 mm) in 203 dpi printer.

```
sLABEL,808,600↵
```



# sLCD

## Description

This command can set the language in the LCD screen.

## Format

sLCD,P1

Parameters	Description
P1=LCD Language	0:English 1:Traditional Chinese 2: Simplified Chinese 3: Spanish 4: Italian 5: Vietnam 6: Korea

## Example

sLCD,0

# sMIRROR

## Description

This command can open the function of Mirror image.

## Format

sMIRROR,P1

Parameters	Description
P1=Mirror image	0: Print normal image 1: Print mirror image

## Example

```
sDIRECTION,1↵
```

```
sMIRROR,0↵
```

Single-line Text  
Single-line Text  
~~Single-line Text~~  
Single-line Text  
**Single-line Text**  
Single-line Text  
Single-line Text

Feed Direction



```
sDIRECTION,1↵
```

```
sMIRROR,1↵
```

Single-line Text  
Single-line Text  
~~Single-line Text~~  
Single-line Text  
**Single-line Text**  
Single-line Text  
Single-line Text

Feed Direction



# sOFFSET

## Description

This command defines how many extra feeding length after each label printing, that would be suitable for selection of Tear, Peer, Cut modes.

## Format

sOFFSET,P1

Parameters	Description
P1	Offset Distance (dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

sOFFSET,8↵

# sORIGIN

## Description

This command can set the displacement of origin coordinates including X axis and Y axis.

## Format

sORIGIN,P1,P2

Parameters	Description
<b>P1</b>	The X axis displacement of origin (Dot)
<b>P2</b>	The Y axis displacement of origin (Dot)

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

- Set the X axis displacement of origin (3mm) and the Y axis displacement of origin (2mm) in 203 dpi printer.

```
sORIGIN,24,16↵
```

# sRS232

## Description

This command can adjust the parameters of RS232.

## Format

sRS232,P1,P2,P3,P4[,P5]

Parameters	Description
<b>P1=Baud rate</b>	Baud rate, available baud rates are as listed : 115200 (Default) 38400 19200 9600 4800 2400 1200
<b>P2= Data bits</b>	7: 7 bits 8: 8 bits
<b>P3= Parity</b>	0: No parity check 1: Even parity check 2: Odd parity check
<b>P4= Stop bits</b>	1: 1 bit 2: 2 bits
<b>P5=Flow control</b>	0:None 1:Hardware

## Example

sRS232,115200,8,0,1↵

# sRTCDATE

## Description

This command can set the date in the RTC (Real Time Clock)

## Format

sRTCDATE,P1,P2,P3

Parameters	Description
P1	Year(4 Digits)
P2	Month(1~12)
P3	Day(1~31)

## Example

sRTCDATE,2015,03,03↵

# sRTCTIME

## Description

This command can set the time in 24 hours format of RTC (Real Time Clock).

## Format

sRTCTIME,P1,P2,P3

Parameters	Description
P1	Hour(2 Digits)
P2	Minute(2 Digits)
P3	Second(2 Digits)

## Example

sRTCTIME,10,5,30↵

# sSENSOR

## Description

This command can adjust the sensor type that is used for gap detection.

## Format

sSENSOR,P1

Parameters	Description
P1=Sensor type	0: Reflective gap sensor 1: Transmissive gap sensor

## Example

- Set gap detection sensor to reflective gap sensor.

sSENSOR,0↵



# sSPEED

## Description

This command can adjust the printing speed of printer.

## Format

sSPEED,P1

Parameters	Description						
P1	IPS (Inch per second) <table border="1"><thead><tr><th>Model name</th><th>Printing Speed</th></tr></thead><tbody><tr><td>LP423 (203 dpi)</td><td>1~5 IPS</td></tr><tr><td>LP433 (300 dpi)</td><td>1~4 IPS</td></tr></tbody></table>	Model name	Printing Speed	LP423 (203 dpi)	1~5 IPS	LP433 (300 dpi)	1~4 IPS
Model name	Printing Speed						
LP423 (203 dpi)	1~5 IPS						
LP433 (300 dpi)	1~4 IPS						

## Example

sSPEED,34

# sTHERMAL

## Description

This command can adjust thermal mode of the printer.

## Format

sTHERMAL,P1

Parameters	Description
<b>P1= Thermal mode</b>	0: Direct thermal 1: Thermal transfer

## Example

Set printer in the thermal transfer mode.

sTHERMAL,1↵

# sTPHY

## Description

This command can set the displacement of start printing line.

## Format

sTPHY,P1

Parameters	Description
P1	Start line offset (dot)

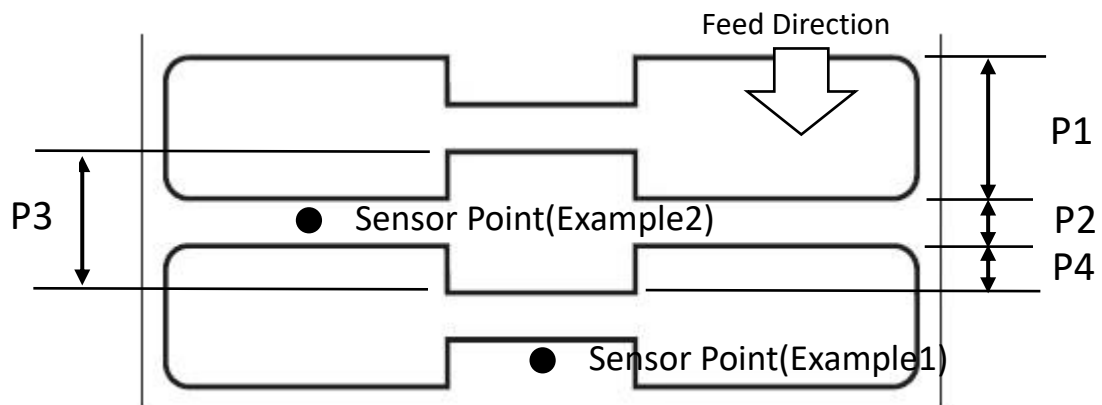
Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Set 3mm displacement of start printing line in 203 dpi printer

sTPHY,24

### 1.1 Butterfly Label



Example1 :

P1 = Label length

P2 = Gap length

TPH Print line offset=0

Example2 :

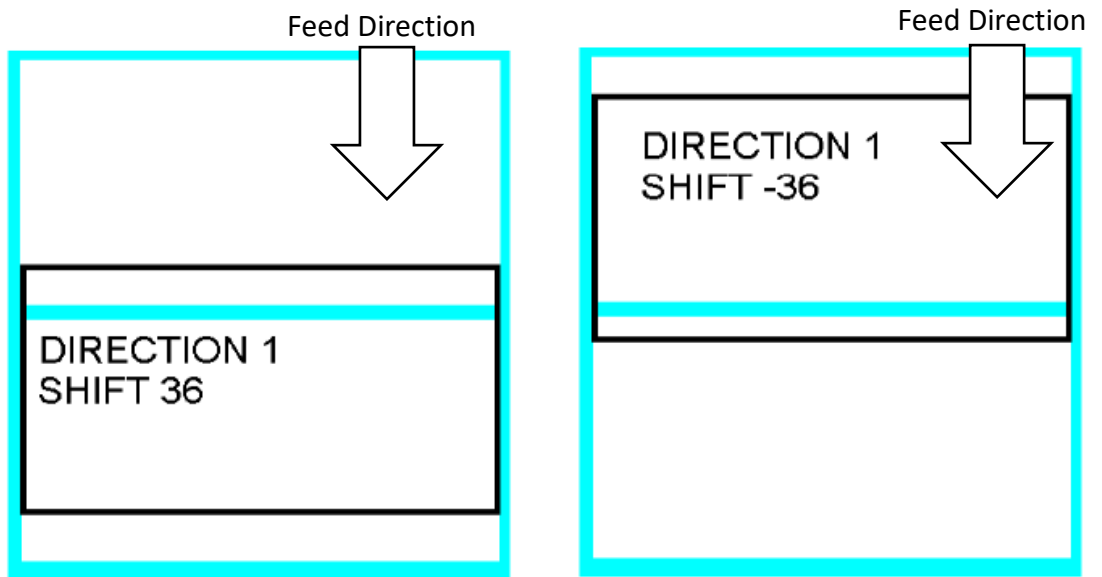
P1 = Label length

P3 = Gap length

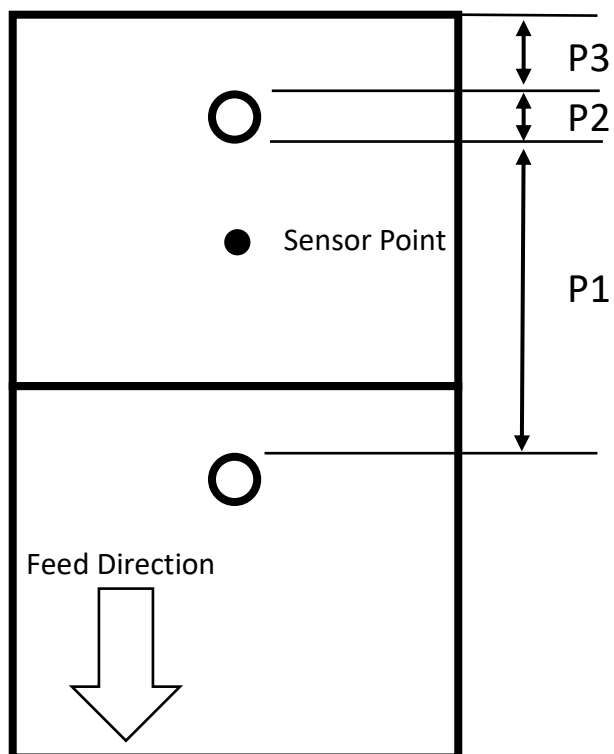
P4 = TPHY offset

TPH Print line offset=-P4

## 1.2 TPHY Offset



## 1.3 Perforation Label Offset



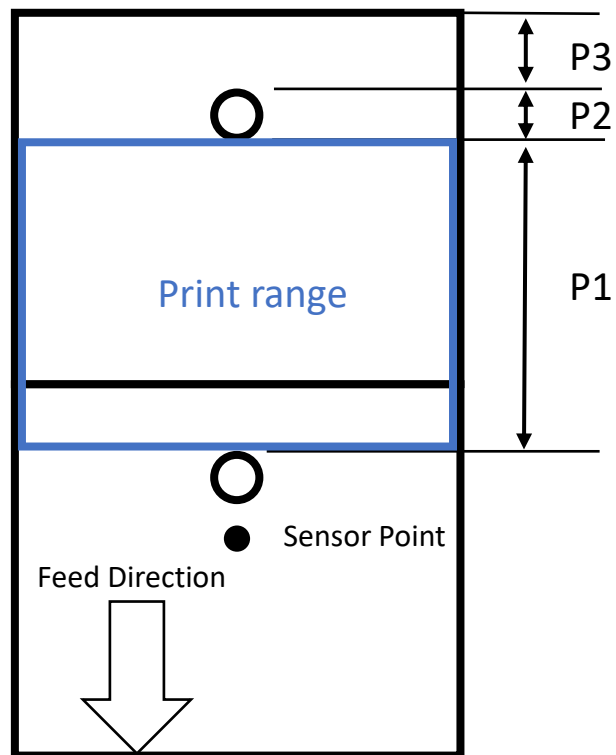
P1 = Label length

P2= Gap length

P3 = TPHY offset

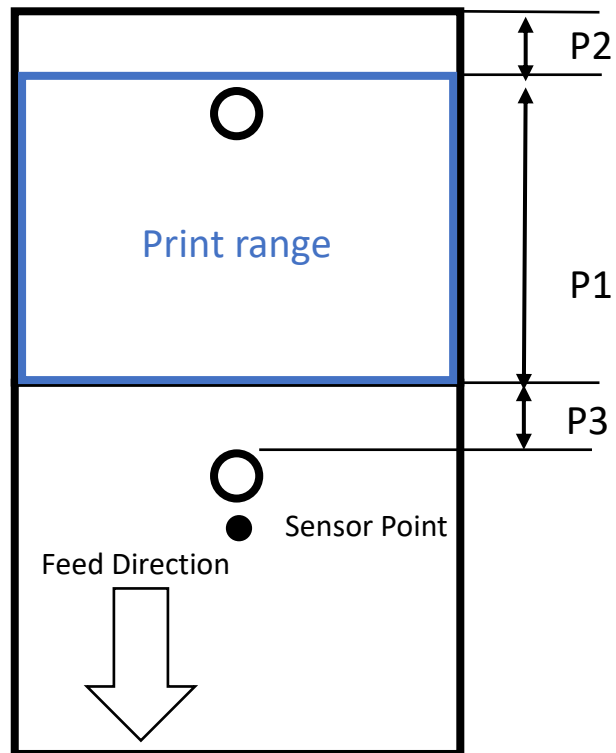
Example1 :

If TPHY offset = 0



Example2 :

If TPHY offset = +P3



# 7

## BASIC Commands



# General commands

## ABS

### Description

To return the absolute value of a numeric expression.

### Format

A% = ABS(N%)

Parameters	Description
A%	Numeric variable to be assigned to the absolute value of a numeric expression.
N%	Numeric expression, it can be an integer or a real number.

### Example

```
wSAVE,"Test.bas"
```

```
Num1% = 2
```

```
Num2% = 9
```

```
Difference% = ABS (Num1% - Num2%)
```

```
wTEXT,50,50,0,1,5,5,0,STR$(Difference%)
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

7

# ASC

## Description

To return the decimal value for the ASCII code for the first character of a given string.

## Format

A% = ASC(Str\$)

Parameters	Description
A%	A% is an integer variable to be assigned to the result.
Str\$	Str\$ is a string variable, consisting of characters.

## Example

84

```
wSAVE,"Test.bas"  
A%=ASC("Test...")  
wTEXT,50,50,0,1,5,5,0,STR$(A%)  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

# CHR\$

## Description

To return the character for a given ASCII value.

## Format

A\$ = CHR\$(N%)

Parameters	Description
A\$	A\$ is a string variable to be assigned to the result.
N%	N% is a numeric expression in the range of 0 to 255.

## Example

```
wSAVE,"Test.bas"
```

```
A$=CHR$(66)
```

```
wTEXT,50,50,0,1,5,5,0,A$
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

**B**

# GOTO

## Description

To branch unconditionally to a specified line number from the normal program sequence.

## Format

GOTO Line Number

### Description

Line Number is the integer number in front of a program line.

## Example

.....

GOTO 100

.....

100

# INSTR

## Description

To search if one string exists inside another one.

## Format

A% = INSTR([N%,] S1\$, S2\$)

Parameters	Description
A%	A% is an integer variable to be assigned to the result.
N%	N% is a numeric expression. Optional offset N% sets the position for starting the search.
S1\$, S2\$	S1\$, S2\$ may be a string variable, string expression, or string constant. If S2\$ is found in S1\$, it returns the position of the first occurrence of S2\$ in S1\$, from the starting point. If N% is larger than the length of S1\$ or if S1\$ is null, or if S2\$ cannot be found, it return 0. If S2\$ is null, it return N% (or 1 if N% is not specified).

## Example

```
wSAVE,"Test.bas"
Str$="ABCGEFGHIJK"
G$="GH"
wTEXT,50,50,0,1,5,5,0,STR$( INSTR(5,Str$, G$))
wTEXT,50,200,0,1,5,5,0,STR$( INSTR(3, Str$, "CGE"))
wPRINT,1
wSAVEEND
wLOAD,"Test.bas"
```

7  
3

# INT

## Description

To return the largest integer that is less than or equal to the given numeric expression.

## Format

A% = INT(N%)

Parameters	Description
A%	A% is an integer variable to be assigned to the result.
N%	N% is a numeric expression, it can be an integer or a real number.

## Example

```
wSAVE,"Test.bas"
```

```
A% = INT(9)
```

```
wTEXT,50,50,0,1,5,5,0,STR$(A%)
```

```
B% = INT(-5.68)
```

```
wTEXT,50,200,0,1,5,5,0,STR$(B%)
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

9  
-6

# LEFT\$

## Description

To retrieve a given number of characters from the left side of the target string.

## Format

A\$ = LEFT\$(Str\$, N%)

Parameters	Description
<b>A\$</b>	A\$ is a string variable to be assigned to the result.
<b>Str\$</b>	Str\$ may be a string variable, string expression, or string constant.
<b>N%</b>	N% is a numeric expression. If N% is larger than the length of Str\$, the Str\$ is returned. If N% is zero, the null string is returned.

## Example

```
wSAVE,"Test.bas"  
Str$ = "ABCDEFGHIJK"  
wTEXT,50,50,0,1,5,5,0, LEFT$(Str$,3)      ABC  
wTEXT,50,200,0,1,5,5,0, LEFT$("168lb",3)  168  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

# LEN

## Description

To return the length of a string.

## Format

A% = LEN(S\$)

Parameters	Description
A%	A% is an integer variable to be assigned to the result.
S\$	S\$ may be a string variable, string expression, or string constant.

## Example

```
wSAVE,"Test.bas"  
Str$="ABCDEFGHJK"  
Str$ = STR$(LEN(Str$))  
wTEXT,50,50,0,1,5,5,0," Len:"+ Str$  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

**Len: 11**



# MID\$

## Description

To retrieve a given number of characters from anywhere of the target string.

## Format

A\$ = MID\$( Str\$, N%[, M%])

Parameters	Description
<b>A\$</b>	A\$ is a string variable to be assigned to the result.
<b>Str\$</b>	Str\$ may be a string variable, string expression, or string constant.
<b>N%,M%</b>	N% and M% are numeric expression. This command returns a string of length M% characters from Str\$ beginning with the N%th character. If M% is equal to zero, or if N% is greater than the length of Str\$, then it returns a null string.

## Example

```
wSAVE,"Test.bas"
```

```
Str$ = "ABCDEFGHJK"
```

```
wTEXT,50,50,0,1,5,5,0, MID$(Str$,5,3)
```

```
wTEXT,50,200,0,1,5,5,0, MID$("123& #168lb",6,5)
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

**EFG**

**#168I**

# REM

## Description

To insert explanatory remarks in a program.

## Format

REM remark

---

### Description

Remark may be any sequence of characters.

BASIC interpreter will ignore whatever follows the  
REM end of the line.

## Example

REM This is function

# RIGHT\$

## Description

To insert explanatory remarks in a program.

## Format

A\$ = RIGHT\$(Str\$, N%)

Parameters	Description
<b>A\$</b>	A\$ is a string variable to be assigned to the result.
<b>Str\$</b>	Str\$ may be a string variable, string expression, or string constant.
<b>N%</b>	N% is a numeric expression. If N% is larger than the length of Str\$, the entire string is returned. If N% is zero, the null string is return.

## Example

```
wSAVE,"Test.bas"
```

```
Str$ = "ABCDEFGHJK"
```

```
wTEXT,50,50,0,1,5,5,0, RIGHT$(Str$,3)
```

```
wTEXT,50,200,0,1,5,5,0, RIGHT$("168lbB",3)
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

**IJK**

**IbB**

# SGN

## Description

To return an indication of the mathematical sign (+ or -) of a given numeric expression.

## Format

A% = SGN(N%)

Parameters	Description								
<b>A%</b>	A% is an integer variable to be assigned to the result. <table border="1"><thead><tr><th>A%</th><th>Meaning</th></tr></thead><tbody><tr><td>1</td><td>N% &gt;0</td></tr><tr><td>0</td><td>N% =0</td></tr><tr><td>-1</td><td>N% &lt;0</td></tr></tbody></table>	A%	Meaning	1	N% >0	0	N% =0	-1	N% <0
A%	Meaning								
1	N% >0								
0	N% =0								
-1	N% <0								
<b>N%</b>	N% is a numeric expression, it can be an integer or a real number.								

## Example

```
wSAVE,"Test.bas"
A% = SGN(9.86)
wTEXT,50,50,0,1,5,5,0, STR$( A%)
A% = SGN(-5.68)
wTEXT,50,200,0,1,5,5,0, STR$( A%)
A% = SGN(0)
wTEXT,50,350,0,1,5,5,0, STR$( A%)
wPRINT,1
wSAVEEND
wLOAD,"Test.bas"
```

**1**  
**-1**  
**0**

# STR\$

## Description

To convert a numeric expression to a string.

## Format

A\$ = STR\$(N%)

Parameters	Description
A\$	A\$ is a string variable to be assigned to the result.
N%	N% is a numeric expression.

## Example

```
wSAVE,"Test.bas"
```

```
wTEXT,50,50,0,1,5,5,0, STR$(256)
```

**256**

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

# STRING\$

## Description

To return a string containing the specified number of the requested character.

## Format

A\$ = STRING\$(N%, J%)

A\$ = STRING\$(N%, X\$)

Parameters	Description
A\$	A\$ is a string variable to be assigned to the result.
N%	N% is numeric expression.
J%	J% is numeric expression in the range of 0 to 255, indicating the ASCII code of a character.
X\$	X\$ may be a string variable or string constant.

## Example

```
wSAVE,"Test.bas"
```

```
wTEXT,50,50,0,1,5,5,0, STRING$(10, 45)
```

```
wTEXT,50,200,0,1,5,5,0, STRING$(3, "89")
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

\_\_\_\_\_

888

# VAL

## Description

To return the numeric value of a string expression in integer form.

## Format

A% = VAL(Str\$)

Parameters	Description
A%	A% is an integer variable to be assigned to the result.
Str\$	Str\$ is a string that includes numeric characters. If the first character is not numeric, this command return 0.

## Example

```
wSAVE,"Test.bas"  
A%= VAL("16898")  
wTEXT,50,50,0,1,5,5,0, STR$(A%)  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

**16898**

# Commands for decision structures

## IF ... THEN ...ENDIF

### Description

To provide a decision structure for multiple-line conditional execution.

### Format

IF condition1 THEN [statements1] ENDIF

#### Description

Condition is a logical expression.

statements can be multiple lines of BASIC statements.

### Example

```
wSAVE,"Test.bas"
```

```
K%=30
```

```
IF K% < 10 THEN
```

```
    wTEXT,50,50,0,1,5,5,0,"Case A"
```

```
ELSE
```

```
    wTEXT,50,50,0,1,5,5,0,"Case B"
```

```
ENDIF
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

**Case B**



# Commands for looping structures

## WHILE ... ENDWHILE

### Description

To provide a decision structure for multiple-line conditional execution.

### Format

```
WHILE condition  
  [Statement Block]  
ENDWHILE
```

---

#### Description

If the condition is true, loop statements are executed until the ENDWHILE statement is encountered. Then the program execution returns to WHILE statement and checks the condition again. If it is still true, the process will be repeated. Otherwise the execution continues with the statement following the ENDWHILE statement.

### Example

```
WHILE SysRP% < 10  
.....  
.....  
ENDWHILE
```

# FOR ... NEXT

## Description

To repeat the execution of a block of statements for a specified number of times.

## Format

```
FOR N% = start value TO end value [STEP step]
```

```
  [Statement Block]
```

```
NEXT N%
```

Parameters	Description
N%	<p>N% is an integer variable to be used as loop counter.</p> <p>Start value is a numeric expression which is the initial value for the loop counter.</p> <p>End value is a numeric expression which is the final value for the loop counter.</p> <p>Step is a numeric expression to be used as an increment/decrement of the loop counter. The step is 1 by default.</p> <p>If the loop counter ever reaches or beyond the end value, the program execution continues to the statement following the NEXT statement. The Statement block will be executed again otherwise.</p>

## Example

```
.....  
FOR I%=1 TO TotalCnt%*5  
    .....  
    .....  
NEXT I%  
.....
```

# Function commands

## ADDZ\$

### Description

Insert a particular character before a string (refers to the ASCII table), the number of inserted characters could be set.

### Format

Result\$ = ADDZ\$(String\$, N%, ASCII%)

Parameters	Description
String\$	To fill up the string
N%	How many bits are being filled
ASCII%	To fill up content, refers to the ASCII table

### Example

```
wSAVE,"Test.bas"  
T0$="12345abcde"  
T0$=ADDZ$(T0$, 15, 33)  
wTEXT,50,50,0,1,5,5,0,T0$  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

!!!!12345abcde

# ADDZ1\$

## Description

Strings are inserted into a particular character (refers to the ASCII table), and the number of inserted characters can be set.

## Format

Result\$ = ADDZ\$(String\$, N%, ASCII%)

Parameters	Description
String\$	To fill up the string
N%	How many bits are being filled
ASCII%	To fill up content, refers to the ASCII table

## Example

```
wSAVE,"Test.bas"  
T0$="12345abcde"  
T0$=ADDZ1$(T0$, 15, 33)  
wTEXT,50,50,0,1,5,5,0,T0$  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

**12345abcde!!!!**

# ASC

## Description

To return the decimal value for the ASCII code for the first character of a given string.

## Format

A% = ASC(Str\$)

Parameters	Description
A%	A% is an integer variable to be assigned to the result.
Str\$	Str\$ is a string variable, consisting of characters.

## Example

```
wSAVE,"Test.bas"  
A%=ASC("Test...")  
wTEXT,50,50,0,1,5,5,0,STR$(A%)  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

**84**

# COMPUTE \$

## Description

The two strings are set to do arithmetic.

## Format

Result \$ = COMPUTE\$(String\$, N1%,N2%,OP%,N3)

Parameters	Description
<b>String\$</b>	To do arithmetic strings
<b>N1%</b>	Retain
<b>N2%</b>	Calculates the number of decimal points
<b>OP%= Operator</b>	0:add 1:subtract 2:multiply 3:divide 4:Reminder
<b>N3=Count2</b>	Arithmetic 2, integer or decimal

## Example

T0\$ = "31.56"

N3 = 1.6555

**33.2**

**29.90**

.....

**52.248**

T1\$ = COMPUTE\$(T0\$, 0, 1, 0, N3)

**19.0637**

.....

**1**

T1\$ = COMPUTE\$(T0\$, 0, 2, 1, N3)

.....

T1\$ = COMPUTE\$(T0\$, 0, 3, 2, N3)

.....

T1\$ = COMPUTE\$(T0\$, 0, 4, 3, N3)

.....

T1\$ = COMPUTE\$(T0\$, 0, 0, 4, 3)

# DATE\$

## Description

The RTC date is returned according to the output format.

## Format

Result \$ = DATE\$(Format\$, Offset%)

Parameters	Description
<b>Format \$ =</b>	y2: 2-digit year
<b>Format of date</b>	y4: 4-digit year
	me: English indicates the month
	mn: The numbers indicate the month
	dd: Date
	TJ0: Julian day
	TJ1: Three digits Julian day
	wn: Weeks
	w2: 2 digit weeks
<b>Offset %</b>	Increment (day)

## Example

.....

S1\$ = DATE\$("TJ1y4",0)                   **05 06 2016**

S2\$ = DATE\$("dd-me-y2",3)               **09-APR-16**

S3\$ = DATE\$("mn dd y4",30)              **0972016**

S4\$ = DATE\$("YTW.mn.dd",20)             **105.04.26**

.....

# DELCHAR \$

## Description

Delete a specific character in the string.

## Format

Result\$ = DELCHAR\$(String\$, DelChar%)

Parameters	Description
String\$	To select string
DelChar %	To delete characters, refers to the ASCII table

## Example

T0\$="0d55d22d"

.....

05522

Result\$ = DELCHAR\$(T0\$, 100)

.....



# DELCRLFTYPE \$

## Description

Remove the line feed symbol in the string.

## Format

Result\$ = DELCRLFTYPE\$( String \$, Type%)

Parameters	Description
String\$	To select string
Type %	Delete type: 0: CR+LF 1: Only CR (ASCII 0xD) 2: Only LF (ASCII 0xA)

## Example

```
.....  
TO$ = DELCRLFTYPE$(TO$, 0)  
.....
```

# DISCARD\$

## Description

The string in the specified direction, delete the specified number of characters.

## Format

Result\$ = DISCARD\$(String\$,Num%,Type%)

Parameters	Description
String\$	To delete a string
Num %	Delete the number
Type% = Delete type	0: Remove from the left 1: Remove from the right

## Example

.....

TO\$ = DISCARD\$(TO\$, 5, 0)

.....

# GETFIELD \$

## Description

In the string, search for the specified string.

## Format

Result\$ = GETFIELD\$(String\$, Search\$)

Parameters	Description
String\$	To search for the string
Search\$	Specify the string
Result\$	Search success: Return string. Search failed: Return Null string

## Example

```
wSAVE,"Test.bas"  
TextStr$="Test:GetField"  
S1$=GETFIELD$(TextStr$,"Test:")  
wTEXT,50,50,0,1,5,5,0,S1$  
wREVERSE,0,40,800,100  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```



# GETLINE \$

## Description

In the string, get the specified number of lines (CR + LF is a newline symbol).

## Format

Result\$ = GETLINE\$(String\$,Num%)

Parameters	Description
String\$	To search for the string
Num%	Obtain the number of rows (1 ~)
Result\$	Result string

## Example

```
wSAVE,"Test.bas"
```

```
TextStr$="A00001"+CHR$(13)+CHR$(10)+"A00002"+CHR$(13)+CHR$(10)+"A00003"+  
CHR$(13)+CHR$(10)
```

```
S1$=GETLINE$(TextStr$,2)
```

```
wTEXT,50,50,0,1,5,5,0,S1$
```

```
wREVERSE,0,40,800,100
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```



A00002

# INSERT\$

## Description

In the string, at the specified location, insert a specific string.

## Format

Result \$ = INSERT\$( String\$,N%,InsertStr\$)

Parameters	Description
String\$	Original string
N%	Insert position (0 ~)
InsertStr\$	A specific string to insert
Result\$	Result string

## Example

```
wSAVE,"Test.bas"
```

```
TO$= "12345"
```

```
TO$ = INSERT$(TO$, 3, "ABC")
```

```
wTEXT,50,50,0,1,5,5,0, TO$
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

**123ABC45**

# ONLYNUM\$

## Description

In the string, only keep the numbers.

## Format

Result \$ = ONLYNUM\$( String\$)

Parameters	Description
String\$	Original string
Result\$	Result string

## Example

```
wSAVE,"Test.bas"
```

```
T0$= "123A45BH999"
```

```
T0$ = ONLYNUM$(T0$)
```

```
wTEXT,50,50,0,1,5,5,0, T0$
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

**12345999**

# REPLACEDIG\$

## Description

Set a group of corresponding string, replace the string contains the number of parts.  
If there is no number of words, no change will be made.

## Format

Result\$ = REPLACEDIG\$(S1\$, ReplaceStr\$)

Parameters	Description
Result\$	Process result
S1\$	Original string
ReplaceStr \$	Set 0 to 9 for the replaced character. Must be 10 characters

## Example

```
wSAVE,"Test.bas"
```

```
TO$ = REPLACEDIG$("A1I5|", "P!OMN@LTAU")
```

A! I@:

```
wTEXT,50,50,0,1,5,5,0, TO$
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

# TIME\$

## Description

The RTC time is returned according to the output format.

## Format

Result\$ = TIME\$( Format\$, Offset%)

Parameters	Description
<b>Format \$ =</b> <b>Format of date</b>	h: Hour m: Minute s: Second +: 12 hours, AM or PM time display
<b>Offset %</b>	Increment (seconds)
<b>Result\$</b>	The result is returned

## Example

```
wSAVE,"Test.bas"
TO$ = TIME$("h:m:s +",0)
T1$ = TIME$("h-m-s",50)
wTEXT,50,10,0,1,2,2,0,TO$
wTEXT,50,60,0,1,2,2,0,T1$
wPRINT,1
wSAVEEND
wLOAD,"Test.bas"
```

05:54:21 PM  
18-44-21



# TRIMSTR\$

## Description

From the left or right, delete the characters specified in the string.

## Format

Result\$ = TRIMSTR\$(S1\$, N1%, Direction%)

Parameters	Description
S1\$	To deal with string
N1%	To delete characters' ASCII code
Direction%	0: From left to right 1: From right to left
Result\$	The result is returned

## Example

```
wSAVE,"Test.bas"
```

```
T0$ = TRIMSTR$("55Test55Q555", 53, 0)      Test55Q555
```

```
T1$ = TRIMSTR$("55Test55Q555", 53, 1)      55Test55Q
```

```
wTEXT,50,10,0,1,2,2,0,T0$
```

```
wTEXT,50,60,0,1,2,2,0,T1$
```

```
wPRINT,1
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

# BEEPON

## Description

Start the buzzer and give a warning sound.

## Format

BEEPON

## Example

```
wSAVE,"Test.bas"  
BEEPON  
wSAVEEND  
wLOAD,"Test.bas"
```

# LED

## Description

Set up the LED

## Format

LED,N1%,N2%

Parameters	Description
<b>N1%</b>	1: Set to blue LED 2: Set to Red LED
<b>N2%</b>	0: Close 1: Open 2: Flash

## Example

```
wSAVE,"Test.bas"
```

```
LED,2,2
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

# WRITEPORT

## Description

To send a string to the host through a specified COM port.

## Format

WRITEPORT(PORTNUM%,SENDSTR\$)

Parameters	Description
PORTNUM%	Port number 1:COM port 2:Bluetooth port
SENDSTR\$	Transfer the string

## Example

```
wSAVE,"Test.bas"
```

```
WRITEPORT(1, "Test write port..." + CHR$(13) + CHR$(10))
```

```
wSAVEEND
```

```
wLOAD,"Test.bas"
```

# Input commands

## EXINPUT

### Description

To retrieve input from the external serial port and store it in a variable.

### Format

Result% = EXINPUT (Prompt\$, Variable\$, Fetch%)

Parameters	Description						
<b>Result %</b>	An integer variable to be assigned to the result. <table border="1"><thead><tr><th>Result%</th><th>Meaning</th></tr></thead><tbody><tr><td>1</td><td>Inputs correctly.</td></tr><tr><td>-1</td><td>Input error</td></tr></tbody></table>	Result%	Meaning	1	Inputs correctly.	-1	Input error
Result%	Meaning						
1	Inputs correctly.						
-1	Input error						
<b>Prompt\$</b>	Prompt of string						
<b>Variable\$</b>	String variable that will receive the input data.						
<b>Fetch%</b>	Fetch duration (millisecond)						

### Example

```
.....  
Result% = EXINPUT("Input", String$, 200)  
.....
```



# EXINPUT2

## Description

To retrieve input from the extern serial port and store it in a variable (Used Image to show the prompt).

## Format

Result% = EXINPUT2 (Width%, Height%, Image\$, Variable\$, Fetch%)

Parameters	Description						
<b>Result %</b>	An integer variable to be assigned to the result. <table border="1"><thead><tr><th>Result%</th><th>Meaning</th></tr></thead><tbody><tr><td>1</td><td>Inputs correctly.</td></tr><tr><td>-1</td><td>Input error</td></tr></tbody></table>	Result%	Meaning	1	Inputs correctly.	-1	Input error
Result%	Meaning						
1	Inputs correctly.						
-1	Input error						
<b>Width%</b>	Width of image						
<b>Height%</b>	Height of image						
<b>Image\$</b>	Image data of prompt						
<b>Variable\$</b>	String variable that will receive the input data.						
<b>Fetch%</b>	Fetch duration (millisecond)						

## Example

```
.....  
Result% = EXINPUT2(48, 13,".....", String$, 100)  
.....
```



# EXINPUT\_NEW

## Description

To retrieve input from the extern port and store it in a variable.

## Format

Result% =EXINPUT\_NEW (Port%, Type%, Width%, Height%, Prompt\$, Variable\$, Fetch%[,Convert%,Req\$[,Rev1\$,Rev%]])

Parameters	Description														
<b>Result %</b>	An integer variable to be assigned to the result. <table border="1"><thead><tr><th>Result%</th><th>Meaning</th></tr></thead><tbody><tr><td>1</td><td>Inputs correctly.</td></tr><tr><td>-1</td><td>Input error</td></tr></tbody></table>	Result%	Meaning	1	Inputs correctly.	-1	Input error								
Result%	Meaning														
1	Inputs correctly.														
-1	Input error														
<b>Port%</b>	<table border="1"><thead><tr><th>Port%</th><th>Meaning</th></tr></thead><tbody><tr><td>0</td><td>Serial port</td></tr><tr><td>1</td><td>USB Host</td></tr><tr><td>2</td><td>Bluetooth</td></tr></tbody></table>	Port%	Meaning	0	Serial port	1	USB Host	2	Bluetooth						
Port%	Meaning														
0	Serial port														
1	USB Host														
2	Bluetooth														
<b>Type%</b>	<table border="1"><thead><tr><th>Type%</th><th>Meaning</th></tr></thead><tbody><tr><td>0</td><td>Normal and prompt is string.</td></tr><tr><td>1</td><td>Normal and prompt is image.</td></tr><tr><td>2</td><td>Used request command and prompt is string.</td></tr><tr><td>3</td><td>Used request command and prompt is image.</td></tr><tr><td>4</td><td>Used continuous mode and prompt is string.</td></tr><tr><td>5</td><td>Used continuous mode and prompt is image.</td></tr></tbody></table>	Type%	Meaning	0	Normal and prompt is string.	1	Normal and prompt is image.	2	Used request command and prompt is string.	3	Used request command and prompt is image.	4	Used continuous mode and prompt is string.	5	Used continuous mode and prompt is image.
Type%	Meaning														
0	Normal and prompt is string.														
1	Normal and prompt is image.														
2	Used request command and prompt is string.														
3	Used request command and prompt is image.														
4	Used continuous mode and prompt is string.														
5	Used continuous mode and prompt is image.														
<b>Width%</b>	Width of image														
<b>Height%</b>	Height of image														
<b>Prompt\$</b>	String data of prompt or Image data of prompt.														
<b>Variable\$</b>	String variable that will receive the input data.														
<b>Fetch%</b>	Fetch duration (millisecond)														
<b>Convert%</b>	0: Normal 1: Store in a variable to convert to character.														

---

<b>Req\$</b>	String is request command. Use "\xx" to input Hex code, and "\\\" replace "\"
<b>Rev1\$</b>	Not used
<b>Rev1%</b>	Not used

---

## Example

.....

```
ck% = EXINPUT_NEW(1, 0, 0, 0, "Source@Exd", S$[1], 100,0, "", "", 1)
```

.....



# INKEY\$

## Description

To read panel key and check the key state.

## Format

Key\$=INKEY\$

Parameters	Description
Key\$	Is Key\$ Null, no any Panel Key is to press.

## Example

.....

```
REM "Wait key is to press"
```

```
10
```

```
    Key$=INKEY$
```

```
    IF Key$="" THEN GOTO 10
```

# INPUT

## Description

To retrieve input from the USB Host port and store it in a variable.

## Format

Result% = INPUT(Prompt\$, Var1\$,Default\$)

Parameters	Description						
<b>Result %</b>	An integer variable to be assigned to the result. <table border="1"><thead><tr><th>Result%</th><th>Meaning</th></tr></thead><tbody><tr><td>1</td><td>Inputs correctly.</td></tr><tr><td>-1</td><td>Input error</td></tr></tbody></table>	Result%	Meaning	1	Inputs correctly.	-1	Input error
Result%	Meaning						
1	Inputs correctly.						
-1	Input error						
<b>Prompt\$</b>	Prompt of string						
<b>Var1\$</b>	String variable that will receive the input data.						
<b>Default\$</b>	Default string						

## Example

```
wSAVE,"Test.bas"  
ck% = INPUT("USB Input", S1$,"####")  
IF ck% = -1 THEN  
    GOTO 1024  
ENDIF  
wTEXT,50,50,0,1,5,5,0, S1$  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```



# INPUT2

## Description

To retrieve input from the USB Host port and store it in a variable (Used Image to show the prompt).

## Format

Result% = INPUT2 (Width%, Height%, Image\$, Var1\$, Default\$)

Parameters	Description						
<b>Result %</b>	An integer variable to be assigned to the result. <table border="1"><thead><tr><th>Result%</th><th>Meaning</th></tr></thead><tbody><tr><td>1</td><td>Inputs correctly.</td></tr><tr><td>-1</td><td>Input error</td></tr></tbody></table>	Result%	Meaning	1	Inputs correctly.	-1	Input error
Result%	Meaning						
1	Inputs correctly.						
-1	Input error						
<b>Width%</b>	Width of image						
<b>Height%</b>	Height of image						
<b>Image\$</b>	Image data of prompt						
<b>Var1\$</b>	String variable that will receive the input data.						
<b>Default\$</b>	Default string						

## Example

```
wSAVE,"Test.bas"  
ck% = INPUT2(83 ,13 , " .....", S1$,"#####")  
IF ck% = -1 THEN  
    GOTO 1024  
ENDIF  
wTEXT,50,50,0,1,5,5,0, S1$  
wPRINT,1  
wSAVEEND  
wLOAD,"Test.bas"
```

A black rectangular box containing the white Japanese text "USB インプット" (USB Input).A black rectangular box containing five white hash symbols "#####".

# INPUT\_LIMIT

## Description

To set input length.

## Format

INPUT\_LIMIT(N%)

Parameters	Description
N%	N% is an integer variable. When using INPUT series command, it can set limit on input length.

## Example

INPUT\_LIMIT(255)

# INPUT\_COPIES

## Description

To input how many copies of each set label

## Format

INPUT\_COPIES (N%)

Parameters	Description
N%	N% is an integer variable. It can set how many copies of each set label.

## Example

```
.....  
INPUT_COPIES(CN%)  
.....  
wPRINT,CN%
```

```
Copies  
( 1 - 99999 )  
00002
```

# File manipulation commands

Access mode string	Meaning
Mode 1	Opens file for reading operation only. Error will be returned if target file does not exist.
Mode 2	Opens existing files for both reading and writing operations. If target file does exist, current contents are destroyed. If target file does not exist then creating.
Mode 3	Opens existing files for both reading and writing operations. If target file does not exist then creating.

## OPENIN

### Description

To open a file (Mode 1) and get the file for further processing.

### Format

F% = OPENIN FileName\$

Parameters	Description
F%	0: Open file fail. Other: Open successfully. It returns the file handle.
FileName\$	A string variable indicating the file path. In case of error, open will return an integer value of 0.

### Example

```
F% = OPENIN "Test.log"
```

# OPENOUT

## Description

To open a file (Mode 2) and get the file for further processing.

## Format

F% = OPENOUT FileName\$

Parameters	Description
F%	0: Open file fail. Other: Open successfully. It returns the file handle.
FileName\$	A string variable indicating the file path. In case of error, open will return an integer value of 0.

## Example

F% = OPENOUT "Test.log"

# OPENUP

## Description

To open a file (Mode 3) and get the file for further processing.

## Format

F% = OPENUP FileName\$

Parameters	Description
F%	0: Open file fail. Other: Open successfully. It returns the file handle.
FileName\$	A string variable indicating the file path. In case of error, open will return an integer value of 0.

## Example

F% = OPENUP "Test.log"



# CLOSE

## Description

To close a file.

## Format

CLOSE #F%

Parameters	Description
F%	An integer indicating the file handle.

## Example

CLOSE #F%

# BGETEXT

## Description

To read a specified number of bytes from a file. The current position is updated after reading.

## Format

Result\$ = BGETEXT(N%) # FILEID%

Parameters	Description
Result \$	Result\$ is a string to be returned to the result.
N%	N% is an integer indicating the number of bytes to be read.
FILEID%	FILEID% is an integer variable indicating the file handle.

## Example

STRING1\$=BGETEXT(5)#FILEID%

# BPUT

## Description

To write data to a file.

## Format

BPUT # FILEID% , <expr 1>, <expr 2>, ... ,<expr n>

Parameters	Description
FILEID%	FILEID% is an integer variable, indicating the file handle.
expr 1 ~ expr n	expr 1 ~ expr n is string expression indicating the string data to write to file.

## Example

```
AAA%=168
```

```
BPUT # FILEID%,STR$(AAA%),"HELLO"
```

# GET\$

## Description

Read a line terminated by a null character “\0” from a file.

## Format

FileData\$ = GET\$ # FILEID%

Parameters	Description
FileData\$	FileData\$ is a string to be returned to the result.
FILEID%	FILEID% is an integer variable, indicating the file handle.

## Example

S\$=GET\$ # FileID%

# EOF

## Description

To check if file pointer of a file reaches end of file.

## Format

E%=EOF # FILEID%

Parameters	Description
E%	E% is an integer to be assigned to the result. 0(False): Not end of file -1(True): End of file
FILEID%	FILEID% is an integer variable, indicating the file handle.

## Example

```
WHILE (EOF#FILEID% <> -1)
    .....
    Result$ = BGETEXT(1) # FILEID%
    .....
ENDWHILE
```

# PTR

## Description

To get or move the file pointer position of a file.

## Format

TELLPTR% = PTR # FILEID%

PTR # FILEID% = NPTR%

Parameters	Description
TELLPTR%	TELLPTR % is an integer variable to be assigned to the result. TELLPTR% = PTR # FILEID%, to get the file pointer position of a file.
NPTR %	NPTR % is an integer variable indicating the offset bytes address been specified.
FILEID%	FILEID% is an integer variable indicating the file handle.

## Example

...

```
TELLPTR%=PTR # FILEID%
```

...

```
PTR # FILEID% = 40
```

# EXT

## Description

To get file length of a file.

## Format

FILESIZE% = EXT # FILEID%

Parameters	Description
FILESIZE %	FILESIZE% is an integer variable to be returned the file length.
FILEID%	FILEID% is an integer variable indicating the file handle.

## Example

FILESIZE%=EXT # FILEID%

# SEARCH\_DB\$

## Description

Open the database file, search for specific fields, and specific Key value and then return search results.

## Format

Record\$ = SEARCH\_DB\$(Field%,Key\$ ,FileName\$)

Parameters	Description
Record\$	Search successfully and return the record in the Record \$ Variable. Null value is returned if search failed or file does not exist.
Field%	To search field (0 ~)
Key\$	To search for Key value
FileName\$	To search the database file name

※Database file field (field) and field need to have "," as separator.

Each record ends with CR(0xD)LF(0xA).

## Example

```
wSAVE,"Test.bas"  
.....  
SDesire$ = SEARCH_DB$(2,STarget$, "Weight.csv")  
IF LEN(SDesire$) = 0 THEN  
    GOTO 1  
ENDIF  
S$[3] = FETCH_DB$(SDesire$, 3)  
S$[4] = FETCH_DB$(SDesire$, 2)  
S$[5] = FETCH_DB$(SDesire$, 0)  
.....  
wSAVEEND  
wLOAD,"Test.bas"
```



# FETCH\_DB \$

## Description

Get a string from a specific field and send the result back.

## Format

FieldString\$ = FETCH\_DB\$( Record\$, FieldNum%)

Parameters	Description
FieldString\$	Get the successful return string as FieldString\$ variables If failed, the Null value will be returned
Record\$	To get the strings
FieldNum%	To get the column

※ Field and field need to have "," as separator.

## Example

```
wSAVE,"Test.bas"  
.....  
SDesire$ = SEARCH_DB$(2,STarget$, "Weight.csv")  
IF LEN(SDesire$) = 0 THEN  
    GOTO 1  
ENDIF  
S$(3) = FETCH_DB$(SDesire$, 3)  
S$(4) = FETCH_DB$(SDesire$, 2)  
S$(5) = FETCH_DB$(SDesire$, 0)  
.....  
wSAVEEND  
wLOAD,"Test.bas"
```

## Other

## CLS

### Description

To clear the LCD display.

### Format

CLS

### Example

CLS

# LOCATE

## Description

To move to a specified location in the LCD display.

## Format

LOCATE X%,Y%

Parameters	Description
X%	X% is an integer variable indicating the new X coordinate position.
Y%	Y% is an integer variable indicating the new Y coordinate position.

## Example

LOCATE 1,1

...

LOCATE 2,3

...

# PRINT

## Description

To display data in the LCD display.

## Format

PRINT expression

---

### Description

expression may be numeric or string expression.

## Example

```
LOCATE 1,1  
PRINT "TEST"  
LOCATE 2,3  
PRINT 555
```

# 8

## Other Commands

# aBACK

## Description

This command can reversely feed label. The feeding length is specified by dot.

## Format

aBACK,P1

Parameters	Description
<b>P1=Reverse feeding length (dot)</b>	Reversely feed the label

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Set reversely feed label in 3mm in 203 dpi printer

aBACK,24 ↵

# aCALIBRATE

## Description

This command can automatically detect the gap length and label length, and also automatically adjust the sensitivity of each sensor.

## Format

aCALIBRATE[,p1]

Parameters	Description
P1=Port	R=RS232, U=USB

## Example

- aCALIBRATE↵

# aCUT

## Description

This command can activate cutter to immediately cut the label, and label will not be reversely fed after cutting action.

## Format

aCUT,P1

Parameters	Description
<b>P1</b>	0: Half cut 1: Full cut

## Example

Activate cutter in full cut mode.

aCUT,1↵



# aDEFAULT

## Description

This command can make printer's settings to be the default configuration.

## Format

aDEFAULT

Parameters	Description
None	N/A

## Example

aDEFAULT↵

# aDUMP

## Description

This command can set printer under the DUMP mode. In DUMP mode, the printer outputs code directly without interpretation.

## Format

aDUMP,P1

Parameters	Description
<b>P1=Dump function</b>	0: Disable function 1: Enable function

PS: Press the FEED Key to exit the Dump mode.

## Example

aDUMP,1↵

# aFEED

## Description

This command can feed label with the specified length (by dot).

## Format

aFEED[,P1]↵

Parameters	Description
<b>P1=Feeding Length (dot)</b>	Feed label with the specified length.

Unit: 8 dots = 1 mm in 203 dpi, 12 dots = 1 mm in 300 dpi.

## Example

Feed 3mm label in 203 dpi printer

aFEED,24 ↵

# aFORMAT

## Description

This command can format any disk from the printer.

## Format

aFORMAT,"P1"↵

Parameters	Description
<b>P1</b>	"C:" or "c:" Format RAM disk. "D:" or "d:" Format FLASH disk. "E:" or "e:" Format SD disk.

## Example

Format FLASH disk in the printer

aFORMAT,"D:"↵

# aRESET

## Description

This command can reset printer and then restart the printer.

## Format

aRESET

Parameters	Description
None	N/A

## Example

aRESET↵

# REM

## Description

This command can put any comment in the coding and that will be ignored by the printer.

## Example

```
REM wBOX,50,50,200,200,5↵
```

# 8

## Appendix

# Appendix- Maximum printing size

## Maximum Printing Width:

Model name	Specification
LP423N/LP423A	108 mm (4.25 inch)
LP433N/LP433A	110 mm (4.32 inch)
LP433E	101.6 mm (4 inch)

## Maximum Printing Length:

Model name	Specification
LP423N/LP423A	4572mm(180 inch)
LP433N/LP433A	2032mm(80 inch)
LP433E	508 mm(20 inch)

# Appendix- Disk

## Disk Information:

Model	Specification	Capacity
LP4 Series	Disk C	SDRAM: 4MB
	Disk D	Flash-ROM: 3MB
	DISK E	SD Card: 64MB~32GB