PD165	Communication Data Method		
	Set Range：	Unit：	Factory Setting：0

0：8N1 For ASCII　1：8E1 For ASCII　2：8O1 For ASCII

3：8N1 For RTU　4：8E1 For RTU　5：8O1 For RTU

MODBUS Communication Protocol

When using the RS485 communication interface, each of the inverters must be set for its own address so that the computer can use this individual address to carry out the control.

1：The communication protocol has two kinds of control mode：

⑴ RTU (Remote Terminal Unit) mode

⑵ ASCII（American Standard Code for information interchange）mode

Information of codes:

RTU mode: Each of 8-bit data is composed of two 4-bit (hexadecimal), for example: 64H

ASCII mode: Each of 8-bit data is composed of two ASCⅡ byte, for example:

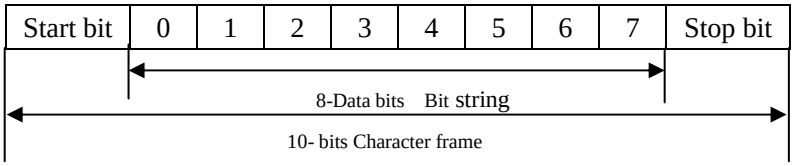
One 1-bit data 64H (hexadecimal) is composed of ASCⅡ byte“64”, included“6”(36H) and“4”(34H).

Byte	0	1	2	3	4	5	6	7
ASCII code	30H	31H	32H	33H	34H	35H	36H	37H

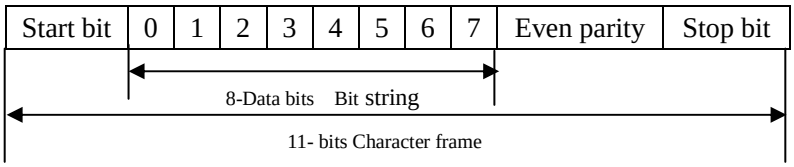
Byte	8	9	A	B	C	D	E	F
ASCII Code	38H	39H	41H	42H	43H	44H	45H	46H

2：Communication Data Method

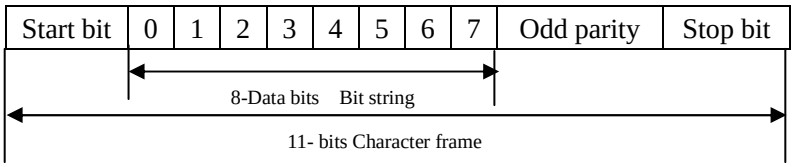
⑴ 8N1 For ASCII PD165=0



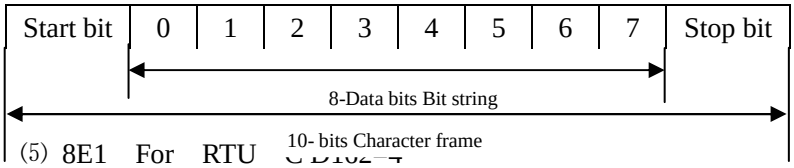
⑵ 8E1 For ASCII PD165=1



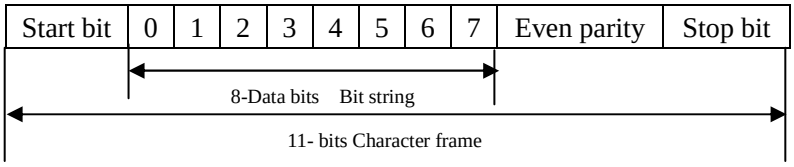
⑶ 8O1 For ASCII PD165=2



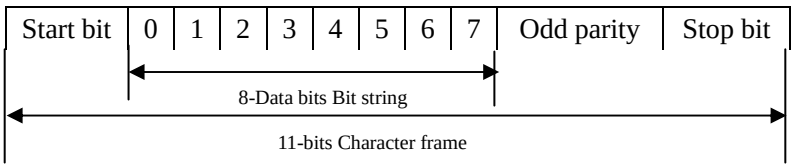
(4) 8N1 For RTU P D165=3



(5) 8E1 For RTU C D165=4



(6) 8O1 For RTU P D165=5



3：Communication Document Formats

3.1 ASCII Mode

Communication Document Forms

STX “：”（3AH）	ADDR	FUNC	LEN	DATA （n-1） ... $DATA_0$	CRC	END CR（0DH） LF（0AH）

(1) STX：Starting unit“：”（3AH）

(2) ADDR communication address,8-bit data is composed of two ASCⅡ byte.

00：Broadcast mode is MODBUS

01~250：Address of the corresponding inverters.

(3) FUNC：Function code 8-bit data is composed of two ASC Ⅱ byte.

01：FUNC READ, Read the data of function code

02：FUNC WRIT, write the data of function code

03：Write control data

04：Read control status data

05：Write inverter frequency data

06：Reserved

07：Reserved

08：Loop test

a：Read function code data

Format:

ADDR 01 LEN FUNC Data

ADDR=0 means no answer

ADDR≠0 means a reply from inverter of this address

When inverter reply normal, the format as follows：

ADDR 01 LEN FUNC Data

If DATA is one word, the LEN=3, If DATA is one byte, the LEN=2 .

When inverter has no this function code or reply no effect, the format as follows：

ADDR 81H 01 FUNC

b：Write function code data

Format：

ADDR 02 LEN FUNC Data

ADDR=0 for broadcast, it write to all inverter, but no reply.

ADDR≠0, set data and reply from inverter of this address.

When the setting is incorrect or the inverter does not have this function, the format returned is as follows：

ADDR 81H 01 FUNC

c：Control commands

Format：

ADDR 03 LEN CNTR

ADDR=0 for broadcast, it write to all inverter, but no reply

ADDR≠0, reply and return.

CNTR

7	6	5	4	3	2	1	0
jogr	jogf	jog	r/f	stop	rev	for	run

When the setting is correct it will return to present control status.

Format：ADDR 03 LEN CNST

CNST

7	6	5	4	3	2	1	0
Track start	Braking	r/f	jogging	running	r/f	jog	run

When the check is not correct,

ADDR 83H 01 CNST

d：Read status value

Format：

ADDR 04 01 CFG

ADDR=0, no reply

ADDR≠0, reply.

CFG=0~7, reply single data

0：Set F 1：Out F 2：Out A 3：RoTT

4：DCV 5：ACV 6：Cont 7：Tmp

For example: read agreed frequency

Send：01 04 03 00 CRC

Return：01 04 03 13 88 CRC

In which, 13 88 are data

13 for high order, while 88 for low order.

⑷ LEN：data length, It means the length of $D_{(n-1)}\cdots D_0$, Length set: when one word, LEN=3，when one byte or ＜1byte, LEN=2。

⑸ DATA：<Data characters> data content. 2n ASCII compose n bytes, it have fifty ASCⅡ at most.

⑹ LRC：longitudinal redundancy check

ASCII mode: Get LRC methods is that add ADDR to the last data, if the result is more than 256,then the result subtract 256 until the result is less then 256 (if the result is 128H, take 28H), then 100H subtract the result get LRC.

⑺ For example: write 30.00Hz to inverter of 01（write to PD003）

STX	ADDR	FUNC	LEN	DATA	LRC	END
":"	"0" "1"	"0" "2"	"0" "3"	"0" "0" "0" "B" "B" "8"	"3" "7"	"CR" "LF"
3AH	30H 31H	30H 32H	30H 33H	30H 30H 30H 42H 42H 38H	33H 37H	0DH 0AH

Calculate LRC: 01H+02H+03H+00H+0BH+B8H=C9H

C9H subtracted from 100H: 37H

So the sent data is following：3AH 30H 31H 30H 32H 30H 33H 30H 30H 30H

42H 42H 38H 33H 37H 0DH 0AH

3.2 RTU Mode

Quiet	ADDR	FUNC	LEN	$D_{(n-1)}$ ~$D_{(0)}$	CRC	Quiet
＞50ms						＞50ms

⑴ Quiet: the time of no data is more than 50 ms

⑵ ADDR：Communication address, 8-bit data

⑶ FUNC：Function code, 8-bit data, refer to 3.1-3

⑷ LEN：Data length, the length of $D_{(n-1)}$ ~D_0

⑸ DATA：data content, n*8-bit

⑹ LRC：Longitudinal Redundancy Check

RTU mode: get CRC（cyclical Redundancy Check）.

The CRC calculation method is following:

⑴ make a 16-bit register and set value 0FFFFH(call CRC register)

⑵ done first byte of data Exclusive OR with low byte of 16-bit CRC register and save the result to CRC register

⑶ done 1 bit right shift with CRC register and fill zero to left bit, then check low bit of CRC register.

⑷ if the low bit is zero, then do repeat setp3, else CRC register do Exclusive OR with 0A001H.

⑸ done repeat step 3 and 4,until CRC register done right shift 8 times, then the byte is fully done.

⑹done repeat step 2 to 5 for the next byte of data, until process completely all data. The last data of CRC register is CRC value. When send CRC value in command data, low bytes must change the sequence with high bytes, i.e. low bytes will be sent first.

⑺ Example 1: Write 30.00Hz to inverter of 01

Command data

ADDR	FUNC	LEN	DATA	CRC
01H	02H	03H	00H 0BH B8H	7FH 0CH

Sent data：01H 02H 03H 00H 0BH B8H 7FH 0CH

⑻ Example 2:

The following is that get CRC value with C language. The function has two parameters:

```
Unsigned char data ←———the point of data buffer
Unsigned char length ←———  number of data buffer
This function will send back the CRC value with unsigned integer format.
Unsigned int crc_chk (unsigned char data, unsigned char length)
{
int j;
unsigned int reg_crc=0xffff;
while (length--){
reg_crc^=*data=++;
for(j=0;j<8;j++={
      if(reg_crc&0×01){/*LSB(b0)=1*/
         reg_crc=(reg_crc>>1)^0xa001;
         }else{
         reg_crc=reg_crc>>1;
      }
         }
 }
  return reg_crc;
 }
```

PD170	Display Items		
	Set Range： 0—5	Unit：1	Factory Setting：0

This parameter is only valid when Bit 2 is set to 1 in PD171. For the details refer to PD171.

0：Inverter Temperature

1：Counter Value

2：PID Target Value

3：PID Feedback Value

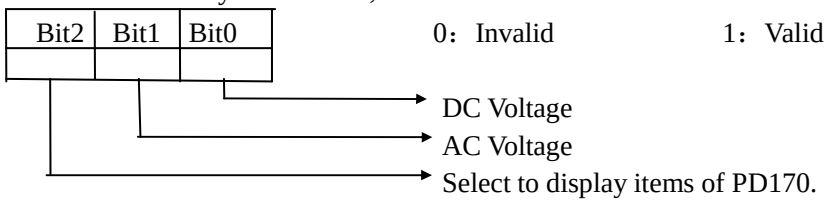
4：Present running time of power up (Unit: Hour)

5：Total running time of power up (Unit: Hour)

PD171	Display Items Open		
	Set Range：0—7	Unit：1	Factory Setting：0

This parameter is set for selection of displaying of DC voltage, AC voltage and other items so that the customer can monitor and view them in sequence through the switch key.

It can be is set first in the binary 3 bits mode, and then converted to a decimal value.



In the contents displayed the factory setting is to show output frequency, set frequency, output current and output revolution through the switch key. If it is necessary to view and monitor other items they can be set through PD170 and PD171.

PD172 Fault Clear		**
Set Range：00—10	Unit：1	Factory Setting：00

01 is for fault clear. Others have no function.

PD174 Rated Current of Inverter	Unit：1A	Factory Setting：*

It is depending on the model and can't be changed.

PD175 Inverter Model		
Set Range：0—1	Unit：1	Factory Setting：0

0：Constant torque 1：For kinds of fan. It can be observed, but not changed.

PD176 Inverter Frequency Standard	Unit：1	Factory Setting：*

0：50Hz 1：60Hz It is factory setting. It can be observed, but not set.

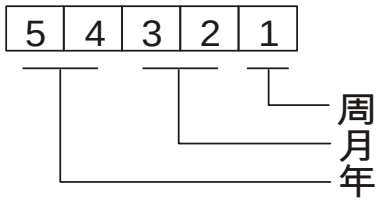
PD177 Fault Record 1	Factory Setting：————
PD178 Fault Record 2	Factory Setting：————
PD179 Fault Record 3	Factory Setting：————
PD180 Fault Record 4	Factory Setting：————

When it has no fault record it shows ——. After access to this parameter the fault display can be checked.

PD181 Software Version	Factory Setting：*

It can be observed, but not set.

PD182 Manufacture date	Factory Setting：*



It is factory setting. It can be observed, but not set.

Week
Month
Year

PD183 Serial No.	Factory Setting：*

It is factory setting. It can be observed, but not set.

Note:

* means the said parameter has a variety of set values or should be set specifically according to concrete conditions.

** means the said parameter can be set during the operation.