WebChip™ Series Material

PS2000A Network Communication Controller Datasheet

Document Number: TECINHD001

Document Date: February 2002

Copyright

Trademarks

WebChip™, MCUnet™, MCUap™ and WebChipServer™

All other trademarks are the property of their respective owners.

Patents

The protocol and/or products of P&S DataCom Corp. include inventions that are the subject of patents pending.

Table of Contents

1 Introduction

P&S DataCom Corporation has developed an internet-enabling technology, called WebChip™, for any microcontroller-based application. This technology includes a silicon chip (the WebChip™) and a gateway. The WebChip™ unifies the microcontroller application data with the gateway networking capability. Object-oriented techniques are used to map any microcontroller application data in a structured, generic fashion into a format usable by gateway intelligence. This object mapping and communication is achieved using two open protocols. The MCUnet™ protocol handles data communication between the microcontroller and the WebChip™, while the MCUap™ protocol handles data communication between the WebChip™ and the gateway.

Real-time microcontroller data, transferred through the WebChip™ to the gateway, can be made available anywhere on the Internet with supervision and management control. The gateway maintains the Internet connectivity capability for numerous WebChip™-enabled devices. This affords users a single point of contact for monitoring and controlling multiple devices, with less administrative overhead than that required by a user-to-device direct solution. Users can potentially access any and all devices via the Internet. However, device owners and users require some control and limitation of access to their devices. The gateway provides this device management, security management, and network management for Internet-connected devices. These supervisory functions extend basic Internet connectivity with real-world requirements. Practical device use over the Internet necessitates inclusion of these essential management features.

The PS2000A is one of the members of the WebChip^{TM} family. It is a network communication controller designed to connect any MCU (“Microcontroller”) application system to a gateway. It therefore can serve as a bridge for an intelligent device (any electronic device that contains an MCU) to easily connect to the Internet. The communication between the MCU application system and the PS2000A conforms to the MCUnet^{TM}1.0 Protocol, and the communication between the network server and the PS2000A conforms to the MCUap^{TM}1.0 protocol. A typical application of the PS2000A is shown in Figure 1.

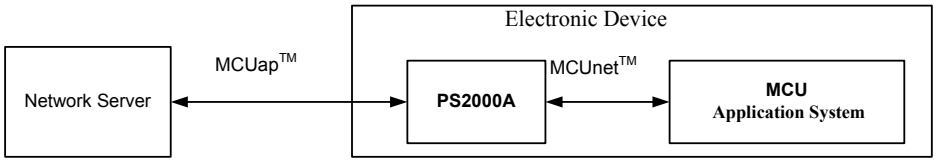


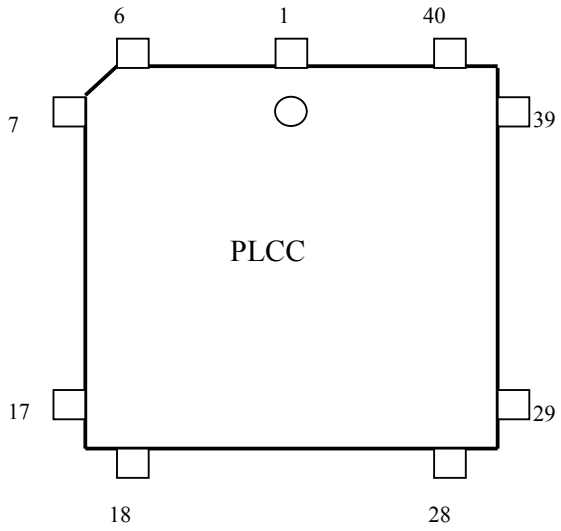
Figure 1 A typical PS2000A Application

2 Features

- Network Communication Controller in a single chip
- Unique 64 bit internal digital ID for device identification
- High Speed, meets most MCU system communication control needs
- Supports 16 bit addressing
- 2K bytes of FLASH memory for MCU device property table storage
- Supports both local and remote MCU device property table download
- Supports both SPI and I²C for MCU-WebChip communication
- Supports any physical interface for WebChip-Gateway communications. Current stock includes RS232, RS485 and Modem interface capability.
- Low power “Sleep” mode with auto-wakeup
- Low EMI
- Single 5V power supply

3 Pin Configuration and Ordering Information

The PS2000A has two packages: PLCC and LQFP, as shown in Figure 2 and Figure 3

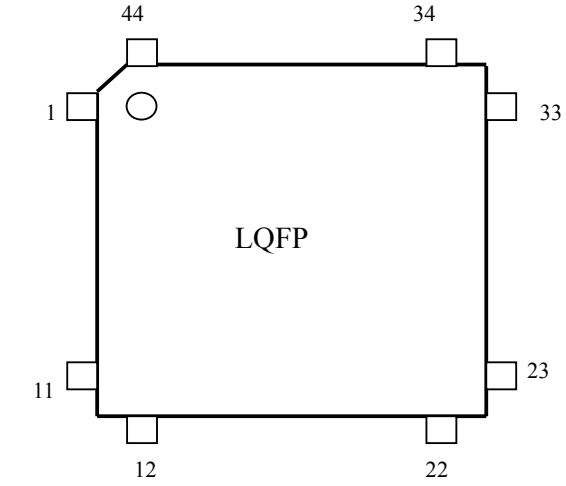


Pin	Function	Pin	Function
1	NIC*	17	/DEBUG
2	/STATE	18	CTS
3	S/I	19	DE/RE
4	DCD	20	XTAL2
5	DTR	21	XTAL1
6	/INT	22	Vss
7	DO	23	NIC*
8	CLK (SCL)	24~31	ND**
9	DI (SDA)	32	/ PROG
10	RST	33	ND**
11	RXD	34	NIC*
12	NIC*	35	PEN
13	TXD	36~43	ND**
14	/CS	44	Vcc
15	ND**		
16	RTS		

* No internal connection

** Not defined (reserved)

Figure 2 PLCC Package



Pin	Function	Pin	Function
1	DO	17	NIC*
2	CLK (SCL)	18~25	ND**
3	DI (SDA)	26	/ PROG
4	RST	27	ND**
5	RXD	28	NIC*
6	NIC*	29	PEN
7	TXD	30~37	ND**
8	/CS	38	Vcc
9	ND**	39	NIC*
10	RTS	40	/STATE
11	/DEBUG	41	S/I
12	CTS	42	DCD
13	DE/RE	43	DTR
14	XTAL2	44	/INT
15	XTAL1		
16	Vss		

* No internal connection

** Not defined (reserved)

Figure 3 LQFP Package

NOTE: LEAVE THE UNDEFINED PIN FLOATING.

TABLE 1

Part Number	TEMPERATURE RANGE (℃)	VOLTAGE RANGE	PACKAGE
PS2000ACP	0 to +70	4.50-5.50 V	PLCC
PS2000AIP	-40 to +85	4.75-5.25 V	PLCC
PS2000ACQ	0 to +70	4.50-5.50 V	LQFP

4 Block Diagram

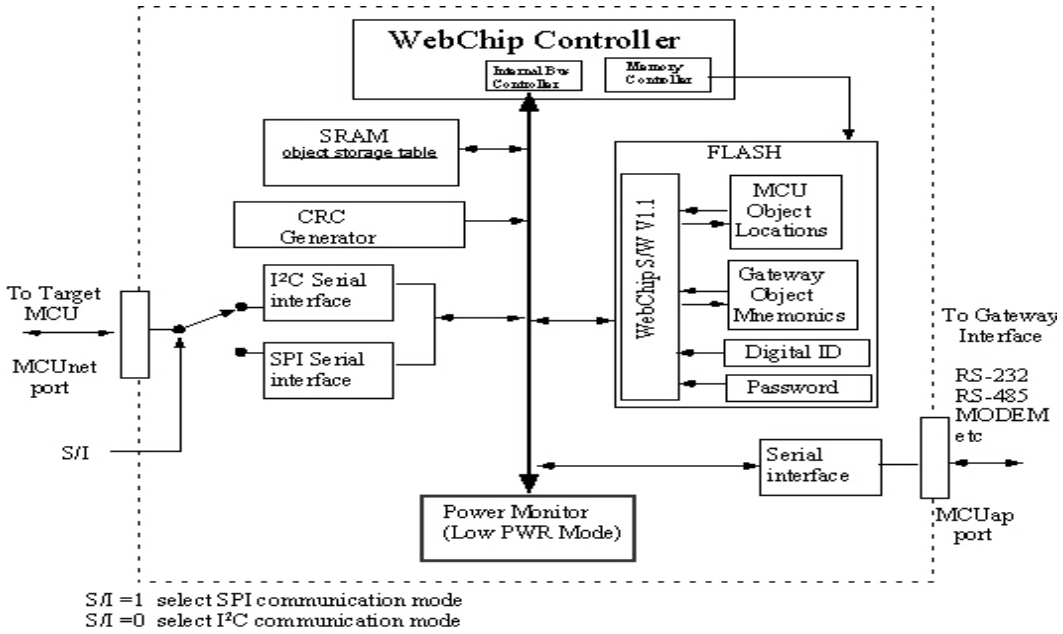


Figure 4 Block Diagram

5 Pin Configuration and Description

Table 2

Name	Pin Number		Type	Description
	PLCC	LQFP		
Vss	22	16	I	Ground
Vcc	44	38	I	Power Supply
/CS	14	8	I	Chip Select: MCU select PS2000A, active Low
/INT	6	44	O	Interrupt: PS2000A requests MCU to receive and transmit data, active Low
DO	7	1	O	SPI Data Signal Output
CLK	8	2	I	SPI Clock Signal Input
DI	9	3	I	SPI Data Signal Input
SCL	8	2	I	I^2C Clock Signal Input
SDA	9	3	I/O	I^2C Data Signal
RXD	11	5	I	Serial Input
TXD	13	7	O	Serial Output
/STATE	2	40	O	Operation State Signal Output, active Low
DE/$\overline{RE}$	19	13	I	RS-485 Transceiver Control
S/$\overline{I}$	3	41	I	SPI/I^2C Communication Mode Select
DTR	5	43	O	Data Terminal Ready Signal Output
DCD	4	42	I	Data Carrier Detect Signal Input
CTS	18	12	I	Clear to Send Signal for Standard Modem
RTS	16	10	O	Request to Send Signal for Standard Modem
RST	10	4	I	Reset Signal Input, active High
PEN	35	29	I	Operation Control Port , active High
XTAL1	21	15	I	Input to the Inverting Oscillator Amplifier
XTAL2	20	14	O	Output from the Inverting Oscillator Amplifier
/DEBUG	17	11	I	Debugging Mode Control Port, active Low
/PROG	32	26	I	Programming Mode Control Port, active Low. For manufacturer use only. Should be float normally.

6 Parameters

6.1 Absolute Maximum Ratings

Table 3

Parameter	MIN TYP MAX	Units
Operating Temperature Range	0 ~ 70	℃
Storage Temperature Range	-65 ~ 150	℃
Supply Voltage V+ to GND	-0.5 ~ 6.5	V
Maximum I_{OL} per I/O Pin	15	mA
Power Consumption	1.5	W

6.2 Operational Characteristics

Table 4

Symbol	Parameter	Value			Units
		MIN	TYP	MAX	
Vcc	Operating Voltage	4.5	5.0	5.5	V
f_{osc}	Operating Frequency		11.0592		MHz
I_{state}	STATE Port Output Current		8	12	mA
Sserver	RS232/RS-485 Mode Communication Rate	1200		115200	Bps
Smodem	Standard Modem Hayes Compatible Modem Communication Rate	1200		115200	Bps
S_{SPI}	SPI Mode Communication Rate			24	Kbyte/s
S_{I^2C}	I^2C Mode Communication Rate			400	Kbyte/s
Trst	Time Interval of Reset Signal (High Voltage)	1	200		ms

6.3 DC Characteristics

T_A = 0°C ~ +70°C, 5V±10%

Table 5

Symbol	Parameter	Test Conditions	Value			Units
			MIN	TYP	MAX	
V_{IL}	Input low voltage	4.5<Vcc<5.5	-0.5		0.2Vcc-0.1	V
V_{IL1}	Input low voltage to SCL, SDA		-0.5		0.3Vcc	V
V_{IH}	Input high voltage		0.2Vcc+0.9		Vcc+0.5	V
V_{IH1}	Input high voltage to SCL, SDA		0.7Vcc		6.0	V
V_{OL}	Output low voltage	Vcc=4.5V I_{OL}=1.6mA			0.4	V
V_{OL1}	Output low voltage to SCL, SDA	I_{OL}=3.0mA			0.4	V
V_{OH}	Output high voltage	Vcc=4.5V I_{OH}=-30μA	Vcc-0.7			V
I_{IL}	Logic 0 input current	V_{IN}=0.4V	-1.0		-75.0	μA
I_{L1}	Input leakage current to SCL, SDA	0<V_I<6V 0<V_{CC}<5.5V			10	μA
Icc	Operating Current	f_{osc}=11.0592MHz		26.7	39	mA
Icc1	Sleep Mode Current			2	14.3	mA

6.4 Digital ID

The PS2000A has a unique 64 bit digital ID (“DID”) for device identification.

It has 8 bytes:

[DID0，DID1，DID2，DID3，DID4，DID5，DID6，DID7]

In the PS2000A:

DID0 is always preset to 99h;

DID1=80h, DID2=00h;

DID3, DID4, DID5, DID6 ranges between 00000000h and 0FFFFFFFh.

DID7 is the CRC verify value.

Therefore, in the PS2000A, the DID is:

[99h, 80h, 00h, DID3, DID4, DID5, DID6, CRC]

The equivalent polynomial function of this CRC is: $X^8+X^5+X^4+1$, where the array for verification is:

DID7=CRC {DID0, DID1, DID2, DID3, DID4, DID5, DID6}

Here is the algorithm for the CRC value calculation in C language:

```
unsigned char DID[7] = {DID0, DID1, DID2, DID3, DID4, DID5, DID6};
unsigned char getCrc()
{
        // X^8+X^5+X^4+1        =110001
        unsigned char crc, oldCrc;
        unsigned char i, j;

        crc = 0;
        for(i=0;i<7;i++ )
        {
          crc ^= DID[ i ] ;
          for( int j = 0; j < 8; j ++ )
          {
                oldCrc = crc;
                crc = crc << 1;
                if( oldCrc & 0x80 )
                crc ^= 0x31;
          }
        }
        return crc;
}
```

7 Typical Circuits

7.1 Power Supply

To improve the anti-noise ability of the PS2000A, you should connect a Tantalum capacitor of 0.1μF at the power input terminal. The typical circuit is shown in Figure 5.

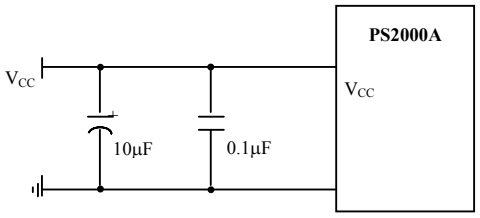


Figure 5 PS2000A Power Supply

The PS2000A may be powered from the power supply of the MCU application system or from a separate power supply. The PS2000A should be connected to the common ground of the MCU application system if the PS2000A uses a separate power supply.

7.2 Reset Signal

The reset signal of the PS2000A is active-high. It should be noted here that the PS2020A and the PS2050A are active low Reset. The reset signal may be provided by the MCU application system or produced by some other separate circuit. The high level of the reset signal needs to stay active long enough (200ms Typ), so as to guarantee the power on reset.

Users may implement the simple circuit (Figure 6) to produce the power on reset signal, or use the special reset IC (e.g. IMP810L) if it is needed.

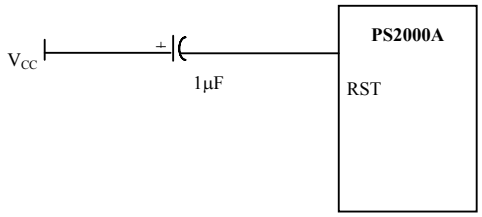


Figure 6 A Simple Power on Reset Circuit

7.3 Oscillator Circuit

Pin XTAL1 is the input terminal of the internal reverse amplifier, and pin XTAL2 is the output terminal of it. We can use this reverse amplifier to configure an oscillator by connecting an 11.0592MHz Crystal as shown in Figure 7.

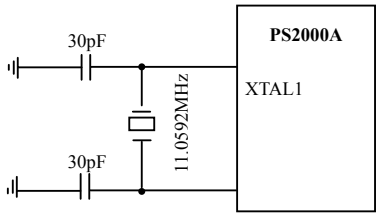


Figure 7 PS2000A Oscillator Circuit

If you use the external clock signal, this clock signal (must be 11.0592MHz) can input from pin XTAL1 of the PS2000A but leave pin XTAL2 floating, as shown in Figure 8.

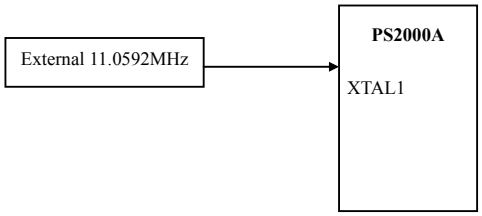


Figure 8 PS2000A Use the External Clock Signal

7.4 STATE Signal

Pin STATE outputs the signal to indicate the current running state of the PS2000A which is active-low. Users may use this signal to drive an LED directly as shown in Figure 9.

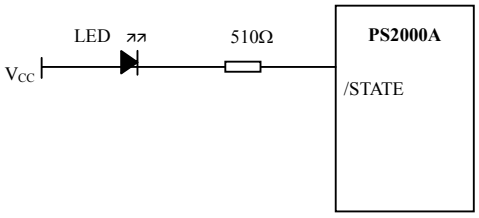


Figure 9 The Usage of the /STATE signal

Pin STATE will generate the active signal (Low), when the PS2000A is running in the following processes:

1. After reset, PS2000A is in the initial configuration mode;

2. PS2000A receives valid data from the network server;

3. PS2000A sends data to the network server;

4. PS2000A starts up the Modem to establish the connection process; and

5. Success in the PS2000A download of the device property table.

If the STATE signal has been amplified, you may use it to drive the high power indicator as shown in Figure 10.

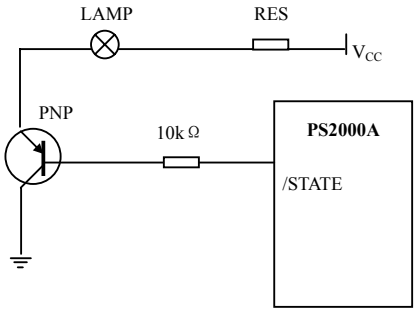


Figure 10 The Amplifier Circuit of /STATE Signal

8 Control Registers

There are three 8 bit control registers in the PS2000A. They are CHIP_CTL, PHY_SEL, and MODE_SEL. CHIP_CTL is used to control the operational state. PHY_SEL and MODE_SEL are used to set the operation mode of the PS2000A communication channel.

8.1 CHIP_CTL

The bit definition of the CHIP_CTL control register is shown in Table 6.

Table 6

D7	D6	D5	D4	D3	D2	D1	D0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Sleep	Reset

D0 (Reset): Software reset for PS2000A, the same function as a hardware reset. In normal operation, Reset = 0. When the MCU sets bit Reset to 1, PS2000A will be reset. The MCU may modify the bit Reset in operation.

D1 (sleep): Low power sleep control bit. When Sleep = 0, PS2000A is in operation state. When Sleep = 1, PS2000A is in low power sleep state. After the MCU or network server wakes up the PS2000A, the Sleep bit will be cleared to 0 automatically.

D2~D7 (Reserved): Bits reserved for system expansion.

8.2 PHY_SEL

The PHY_SEL control register is used mainly for selecting the communication mode and controlling the connection status between the PS2000A and the network server.

The bit definition of PHY_SEL is shown in Table 7.

Table 7

D7	D6	D5	D4	D3	D2	D1	D0
Link_CTL	Reserved	Reserved	CA/AN	PHY3	PHY2	PHY1	PHY0

D3~D0 (PHYn): Define the communication mode between the PS2000A and the network server. The communication mode defined is shown in Table 8.

Table 8

PHY3	PHY2	PHY1	PHY0	Mode
0	0	0	0	RS-232
0	0	0	1	RS-485
0	0	1	0	Modem
0	0	1	1	Reserved
…				
1	1	1	1	Reserved

D4 (CA/AN): In Modem mode, if D4=1, the PS2000A connects to the network server with Call dial mode; if D4=0, the PS2000A is waiting for a connection to the network server in the Answer dial mode.

D5, D6 (Reserved): Bits reserved for system expansion.

D7 (Link_CTL): Controls the connection between the PS2000A and the network server.

The MCU controls the logical or physical connection between the PS2000A and the network server through Link_CTL. When Link_CTL = 0, the PS2000A is disconnected from the network server; when Link_CTL = 1, the PS2000A is connected to the network server.

8.3 MODE_SEL

MODE_SEL control register is primarily used for selecting the communication rate between the PS2000A and the network server. It also sets the security function for the PS2000A. When the PS2000A communicates with the network server in Modem mode, it determines the Modem type. The bit definition of MODE_SEL is shown in Table 9.

Table 9

D7	D6	D5	D4	D3	D2	D1	D0
M2	M1	M0	Security	Reserved	BAUD2	BAUD1	BAUD0

D2~D0 (BAUDn): Sets the communication rate between the PS2000A and the network server, as shown in Table 10.

Table 10

BAUD2	BAUD1	BAUD0	BAUD RATE
0	0	0	1200bps
0	0	1	2400bps
0	1	0	4800bps
0	1	1	9600bps
1	0	0	19200bps
1	0	1	28800bps
1	1	0	57600bps
1	1	1	115200bps

D3 (Reserved): Bit reserved.

D4 (Security): Used for setting the security function. When Security = 1, the security function is enabled and the password is needed for the network server to access the PS2000A. When Security = 0, the security function is disabled and the password is not needed for the network server to access the PS2000A.

D7~D5 (Mn): Undefined for Non-Modem mode communications.

When the PS2000A communicates with the network server in Modem mode, M0, M1, M2 define the Modem type. The PS2000A supports two types of standard Modem, as shown in Table 11.

Table 11

M2	M1	M0	Description
0	0	0	Standard Modem, Compatible with the Hayes AT Command Set
0	0	1	TDK 73M2901(Embedded)
0	1	0	Reserved
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

If the MCU doesn't respond to the configuration request or the PS2000A doesn't connect with the MCU application system, the PS2000A will operate with the default settings as follows:

The default setting of CHIP_CTL register is 00h;

The default setting of PHY_SEL register is 80h;

The default setting of MODE_SEL register is 03h; and

The default PS2000A time constant value is 10 (refer to W_CNTL command in section 7.2 for more details).

9 Communication between the PS2000A and the MCU

The communication between the PS2000A and the MCU is based on the MCUnet^{TM}1.0 protocol. The communication network which is established by the MCUnet^{TM}1.0 protocol is a master-slave mode. In the process of communication, the PS2000A is in master mode and the MCU is in slave mode.

An integrated process of communication between the PS2000A and the MCU should include two parts: the PS2000A sends commands and data to the MCU application system and the MCU responds to the PS2000A.

9.1 Data Packet Structure

Based on section 2.3 of the MCUnet^{TM}1.0 protocol, the structure of the communication data packet between the PS2000A and the MCU is shown as Table 12.

Table 12

Length of Data Packet	Command (Cmd)	Command Control Signal	Main Parameter (Param)	Data (Data)
(One byte)	(One byte)	(One byte)	(Two bytes)	(n bytes)

It is defined by the PS2000A that the transceiver buffer must be not less than 16 bytes or greater than 69 bytes. The PS2000A adjusts the length of the data packet according to the size of the MCU transceiver buffer. Because of the structure of the data packet, the first four items (length of data packet, command, command control signal and main parameter) are fixed in their byte count; the PS2000A only can adjust the length of the data packet by its data area.

The data packet which is sent from the PS2000A to the MCU application system has the same structure as the packet that the MCU sends to the PS2000A.

After the PS2000A sends out the signal of requesting communication, the MCU application system should accomplish the whole data communication with the PS2000A within 500ms. Otherwise the current communication will fault with an overtime error.

9.2 Command

The command bytes between the PS2000A and the MCU include two parts: bits D7~D5 are the command state and bits D4~D0 define the commands as shown in Table 13.

Table 13

Command State			Command				
D7	D6	D5	D4	D3	D2	D1	D0
More	Reserved	Online	W_INIT (0Eh)				
			W_CNTL (0Fh)				
			W_READ (00h)				
			W_RDEV (06h)				
			W_WRIT (01h)				

D7 (More) Data state bit. When the PS2000A accesses the MCU (W_READ and W_WRIT), this bit is valid.

When D7=1, some data hasn't been sent.

When D7=0, the data has been sent already.

When the PS2000A reads some data of an object from the MCU application system, this bit is set by the MCU to indicate that some data hasn't been sent and the PS2000A should issue the read command to read the remaining data continuously. This bit is mainly used for the MCU application system to control some operation or to start the function call. If the data of some object hasn't been received completely, then the user may not start some other operation or function call.

D6 (Reserved) Future use bit.

D5 (Online) Connection state bit. It is used for indicating the connection between the PS2000A and the server. This bit is valid only when the PS2000A sends a command to the MCU and it is set by the PS2000A.

D5=0, PS2000A and network server are in a connecting state.

D5=1, PS2000A and network server are in state of disconnecting.

The command functions are shown in Table 14.

Table 14

Mnemonic designation	Code D4~D0	Function
W_INIT	0Eh	Initial connection command from the PS2000A to the MCU
W_CNTL	0Fh	the PS2000A requests the MCU into initialization. The main parameters are used to configure the related registers in the PS2000A
W_READ	00h	the PS2000A reads the object data information in the MCU
W_RDEV	06h	the PS2000A queries the events in the MCU
W_WRIT	01h	the PS2000A writes data into the objects in the MCU application system

9.3 Physical Connection between PS2000A and MCU

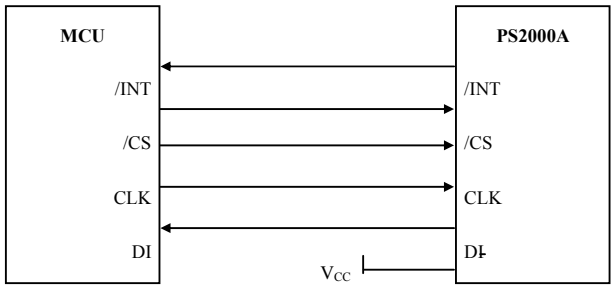
The PS2000A can select the communication mode via pin S/I. The PS2000A supports SPI or I^2C mode to communicate with MCU, as described below:

S/I =1 select SPI communication mode;

S/I =0 select I^2C communication mode.

9.3.1 Connecting PS2000A and MCU in SPI Mode

When pin S/I is connected to Vcc, the PS2000A is in SPI mode as shown in Figure 11.



NOTE: WHEN PIN S/I IS CONNECTED TO VCC OR IT IS LEFT FLOATING, THE PS2000A IS IN SPI MODE.

Figure 11 The Physical Connection between PS2000A and MCU (SPI mode)

When using the SPI interface mode between the MCU and the PS2000A, the PS2000A acts as a peripheral device of the MCU. The MCU operates in master transmission and receiving state and the PS2000A is in slave state.

PS2000A signals transfer is recognized via 3 lines, they are:

CLK: Clock signal line, it's from the MCU to the PS2000A;

DI: Data signal input line, it's from the MCU to the PS2000A;

DO: Data signal output line, it's from the PS2000A to the MCU.

There are two transfer control lines:

/INT: The PS2000A requests the MCU to receive and send data;

/CS: The MCU uses it to strobe the PS2000A.

9.3.2 Connecting PS2000A and MCU in I^2C Mode

When the pin S/I is connected to GND, the PS2000A is in I^2C mode as shown in Figure 12.

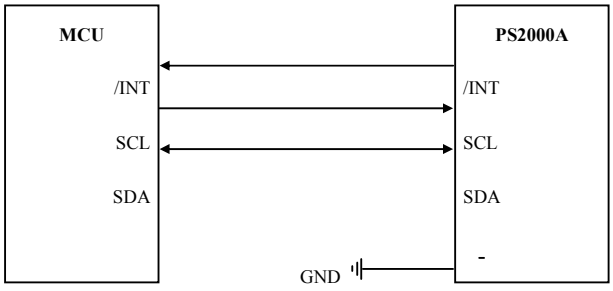


Figure 12 The Physical Connection between PS2000A and MCU (I^2C mode)

When using the I^2C interface mode between the MCU and the PS2000A, the PS2000A acts as a peripheral device of the MCU. The MCU operates in master transmission and receiving state and PS2000A in slave state.

PS2000A signals transfer is recognized via 2 lines, they are:

SCL: Clock signal line, it's from the MCU to the PS2000A;

SDA: Data signal line, it's dual direction;

There is a transfer control line:

/INT: The PS2000A requests the MCU to receive and send data.

9.4 Timing

Whether the SPI or I^2C communication mode is used, the communication between the PS2000A and the MCU is comprised of two processes:

1. The PS2000A sends data to the MCU;

2. The PS2000A receives data from the MCU.

The process is shown as in Figure 13.

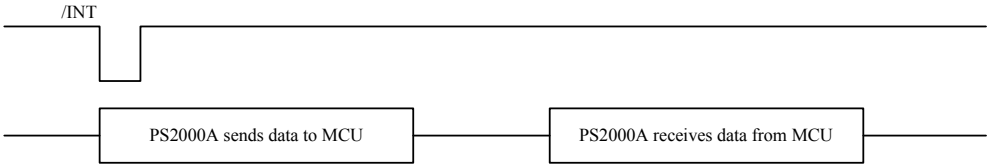


Figure 13 One Communication Process between PS2000A and MCU

9.4.1 SPI Mode Timing

In SPI mode, the communication between PS2000A and the MCU conforms to the following rules:

1. First, the PS2000A sends an /INT signal to request a data exchange with the MCU, the following communication transaction will be controlled by the MCU, that is, the MCU sends chip select signal /CS and clock signal CLK.

2. Data will be transferred in byte form. The MSB is transferred first until the LSB is received.

3. Data is sampled on the falling edge of CLK and shifted on the rising edge.

4. In idle state, CLK is in high level.

The timing of the PS2000A sending data to the MCU is shown in Figure 14.

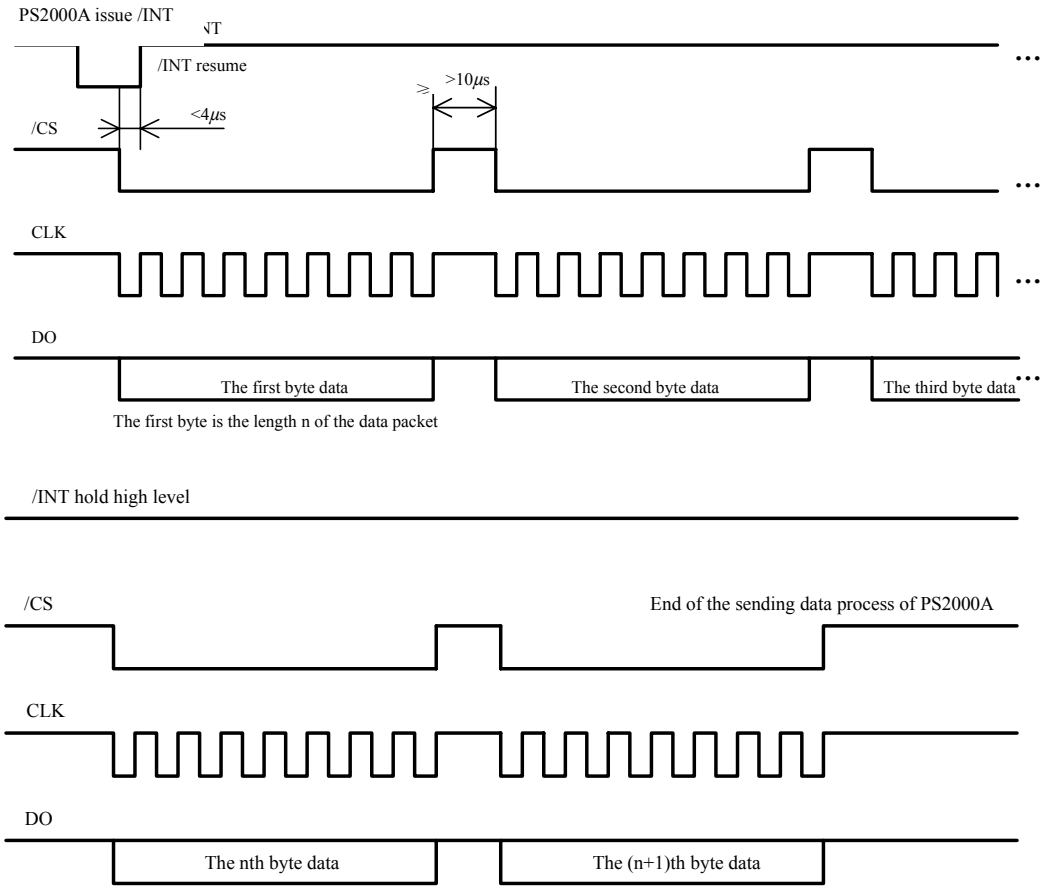


Figure 14 Timing of PS2000A Sending Data to MCU

The detailed CLK and DO timing is shown in Figure 15.

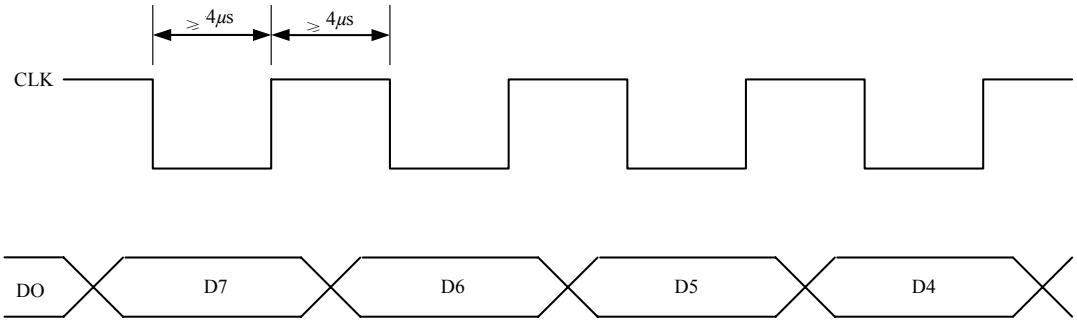


Figure 15 Detailed CLK and DO Timing

The timing of PS2000A receiving data from the MCU is shown in Figure 16.

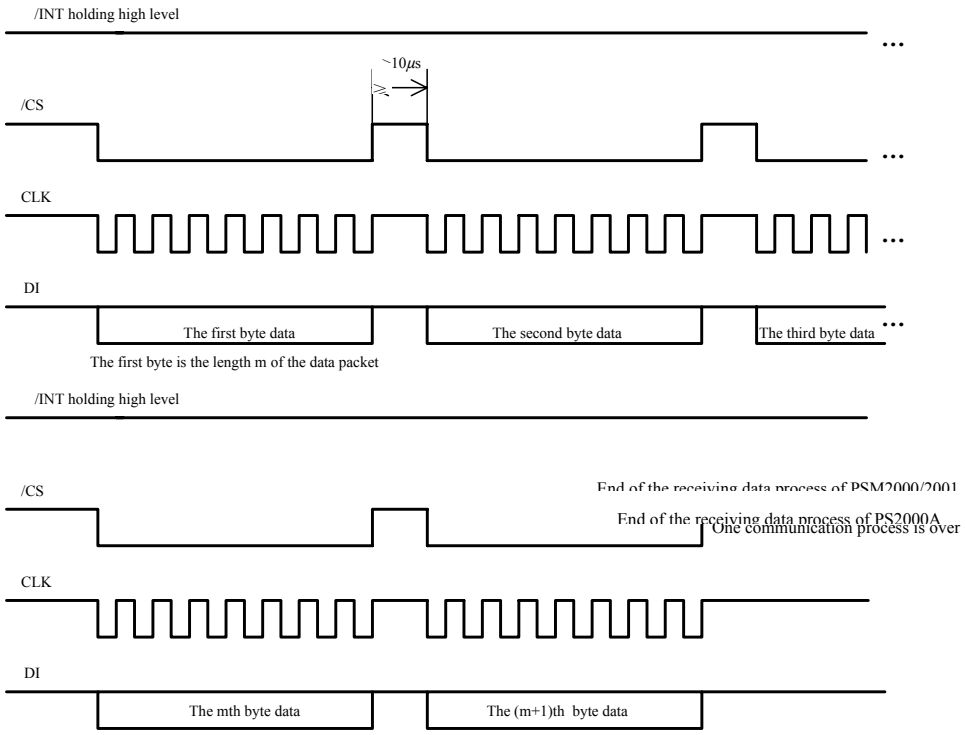


Figure 16 Time sequence of PS2000A receiving data from MCU

The expanded CLK and DI timing is shown in Figure 17.

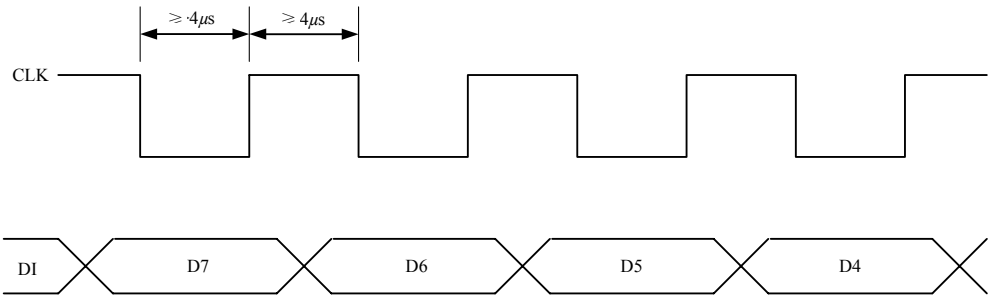


Figure 17 Expanded CLK and DI Timing

9.4.2 I²C Mode Timing

The I²C interface of the PS2000A is compatible with the I²C-Bus Specification Version 2.0-1992. It supports both low-speed mode and fast-speed mode; and the bit speed rate is from 0~400kbit/s. The PS2000A adopts 7 bits fixed address code and the address value is 01111111 R/W b. The PS2000A doesn't identify the broadcast address (00h); e.g. the signal which is sent with the broadcast address can not be received by the PS2000A. Any slave device can be connected to the I²C-bus of the PS2000A, if the address is different.

The timing of the PS2000A sending data to the MCU is shown in Figure 18.

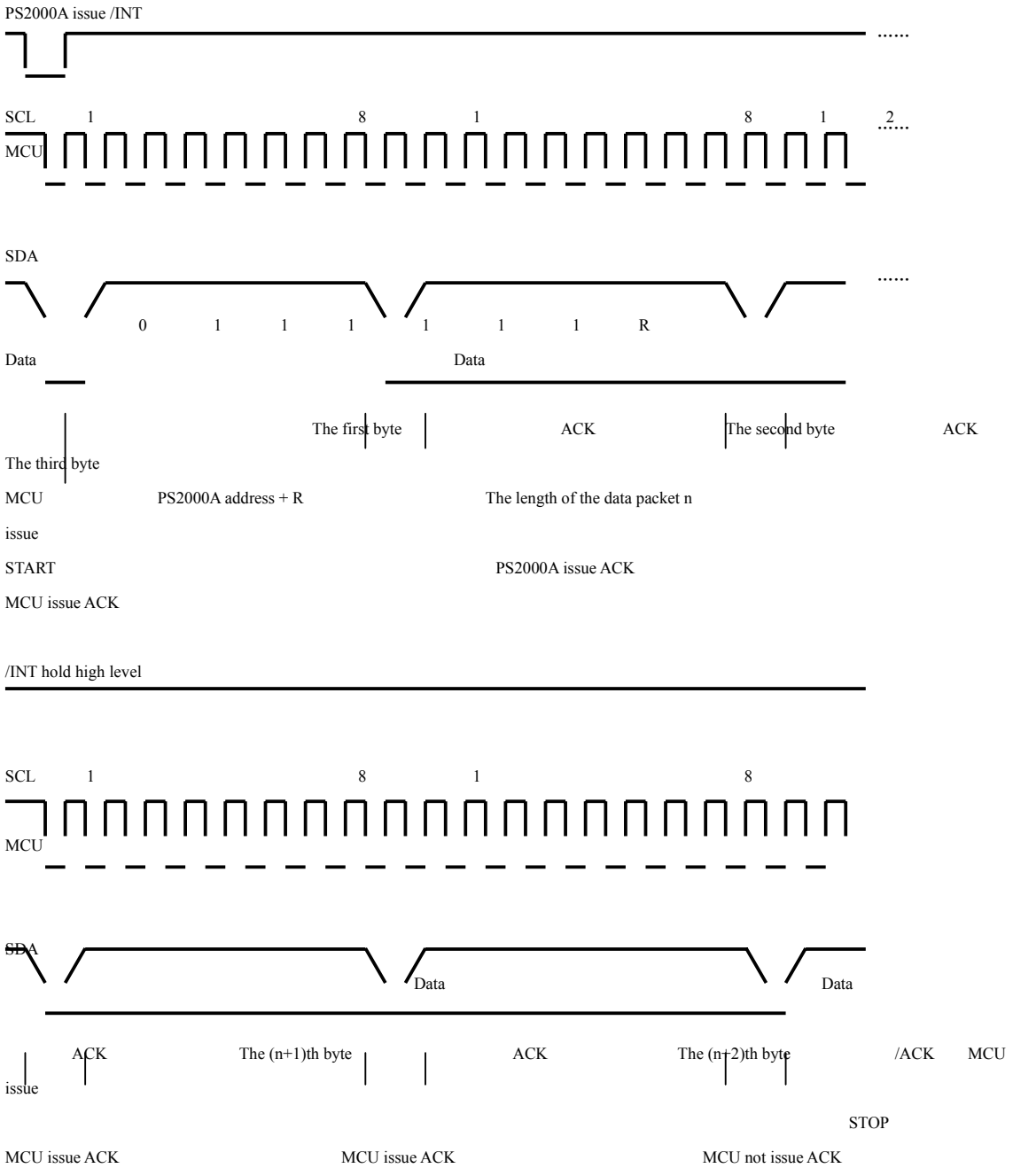


Figure 18 Timing of PS2000A sending data to MCU

The timing of the PS2000A receiving data from the MCU is shown in Figure 19.

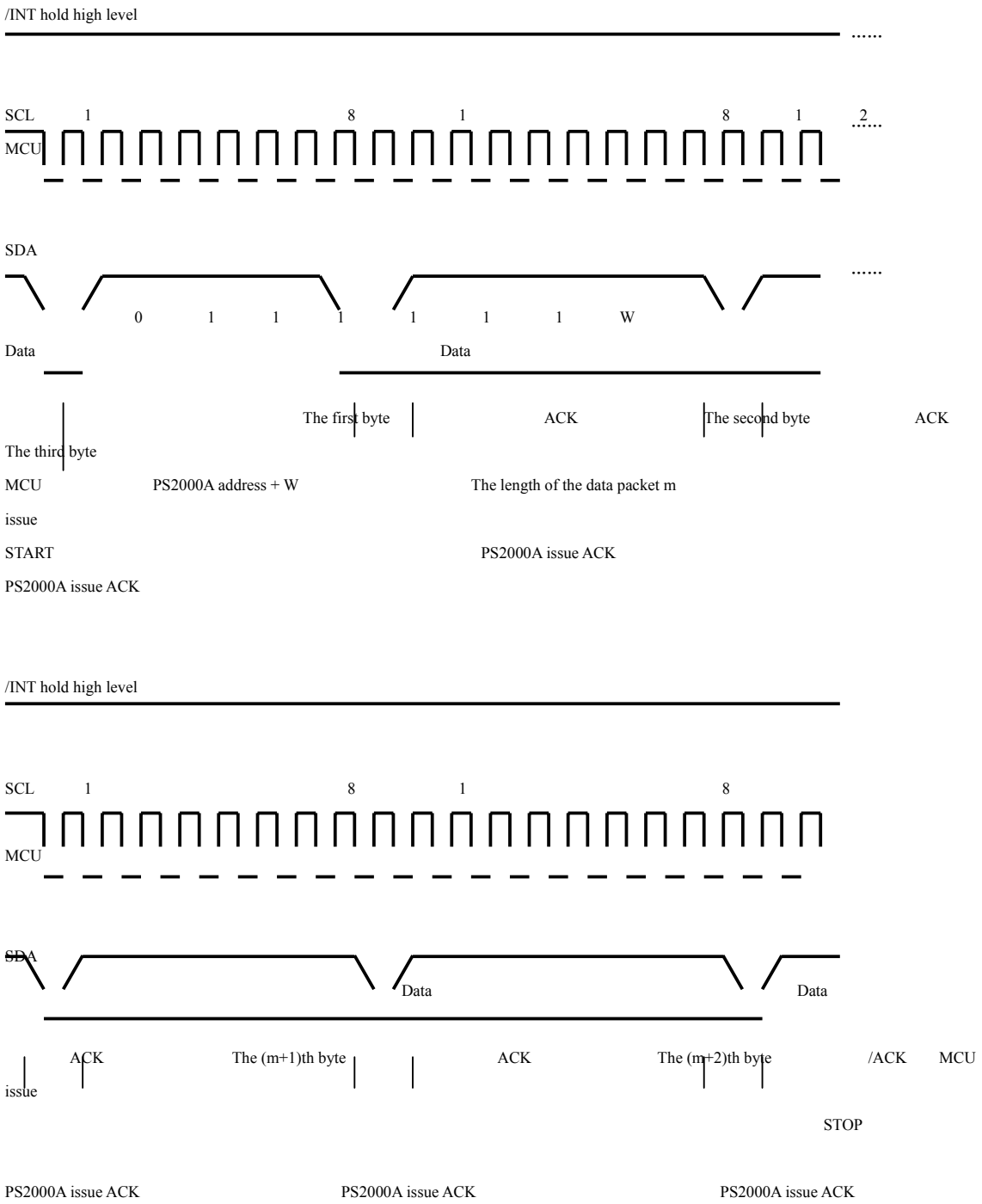


Figure 19 Timing of PS2000A receiving data from MCU

9.5 Troubleshooting

During the communication process between the MCU and the PS2000A, several problems may occur.

Communication Timeout

The time from the initial request signal (/INT) from the PS2000A to the end of the communication process should not exceed 500ms. Exceeding this time limit will cause the communication to fault out.

In SPI mode, to indicate the starting of the response from the MCU to the communication request of the PS2000A, the /CS issued by the MCU changes from high to low.

In I^2C mode, to indicate the starting of the response from the MCU to the communication request of the PS2000A, the MCU issues the START timing signal.

MCU Communication Clock Frequency Incorrect

During the communication process, the MCU provides the clock signal. If the clock frequency is too high, this may cause data transfer errors; if the clock frequency is too low, this may cause a communication timeout. The clock frequency must be set correctly.

Data Packet Size Error

The data packet transfer is controlled by the packet size. If the size is set wrong, it may cause a communication failure, or even cause MCU register conflicts. Therefore, it is important to set the correct size of the data packets.

Timing Error

Sometimes, an external interference may cause the MCU to detect that the communication timing is wrong. In this case, the MCU may withdraw from the communication process. After the communication timeout, the MCU then will begin to prepare for the next communication.

10 Communication between PS2000A and Gateway

The communication between the PS2000A and network server is based upon the MCUap™1.0 protocol. The physical connection between the PS2000A and the server may be RS-232, RS-485 or Modem.

10.1 Data Packet Structure

When the communication between the PS2000A and the network server adopts RS-232, RS-485 or Modem mode, the format of the data packet is as shown in table 20.

Table 20 Communication Data Packet Used for PS2000A and Network Server

Character	Byte	Description	
ESC	1	Escape (1Bh)	Data Packet Header
SOH	1	Start of Heading (01h)	
ESC	1	Escape (1Bh)	
SOH	1	Start of Heading (01h)	
Server_ID	4	Network Server Identification Code	Data Packet of MCUap™ Protocol Application Layer
Device_ID	8	Device Identification Code	
State_Info	1	State Information Byte	
Cmd	1	Command Byte	
OBJECT_NAME	≤13	Object Name String	
Data0	1	Data 0	
Data1	1	Data 1	
Data2	1	Data 2	
…		…	
Data n-1	1	Data n-1	
ESC	1	Escape (1Bh)	Data Packet End
ETB	1	End of Transmission Block (17h)	
CRC	2	CRC Check Value	Check Word

NOTE: ***IF THERE IS A SYMBOL ESC(0X1B) IN THE DATA PACKET OF MCUAP™ PROTOCOL APPLICATION LAYER, A SECOND ESC(0X1B) SHOULD BE INSERTED. BUT CRC CHECKOUT DOES NOT COUNT THE INSERTED ESC.***

The PS2000A allows the maximum length of the data packet to be 99 bytes at once, transmitting or receiving. To establish the buffer size of the communication transceiver used for the network server, the PS2000A should be set up according to the buffer size of the MCU application system.

Only when the device identification code (Device_ID) is matching with the programmed Device_ID in the PS2000A, can the PS2000A receive the data packet. Otherwise, the PS2000A will not respond to the data from the network server.

The PS2000A supports the broadcast access of the network server that uses the super Device_ID code (0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh, 0FFh). But the PS2000A will not respond to the command sent by the network server with the super Device_ID.

The object name supported by the PS2000A is an ASCII character string ended with NULL. The string size of the object name is less than 13.

The following characters are allowed in an object name string:

1. English letters: a~z, A~Z;

2. Number: 0123456789;

3. Other character: Dollar ($), underscore (_), #.

The first character of an object name string must be an English letter or '$'.

The object name supported by the PS2000A is not case sensitive.

The check value of CRC check used in the PS2000A is calculated by the MCUap^{TM}1.0 protocol application layer data packet (from the first byte of Server_ID to Data (n-1)). The sending order of the CRC word is MSB first and LSB last.

The CRC generating multinomial is $X^{16} + X^{12} + X^5 + 1$. The C language algorithm is as shown below:

```
unsigned short ComputeCrc(unsigned short crc, char* bufptr, short len)
{
    int     i;
    crc = 0;
    while (len--) {                 /* calculate CRC from end to start */
        crc ^= (unsigned short) (*bufptr++)<<8;
        for ( i=0 ; i<8 ; ++i ) {
            if (crc & 0x8000)
                crc = (crc << 1) ^ 0x1021;
            else
                crc <<= 1;
        }
    }
    return(crc);
}
```

The bit definition of the state information (State_Info) is shown in table 21.

Table 21

Bit	Token	Description
D7	Event	Event indication bit, which is set when the PS2000A reports. Event=0: the PS2000A has no event to report. Event=1: PS2000A has some event to report, server should continuously use W_RDEV command to read.
D6	Busy	Device is busy and cannot respond to the server's request. The bit is set by the PS2000A according the state of MCU application system.
D5	Up	Up=0: the network server sends data package to PS2000A. Up=1: the PS2000A sends data package to the network server.
D4	DPT_Error	Set by the PS2000A when reporting. DPT_Error=0: the device property table of PS2000A is correct. DPT_Error=1: the device property table of PS2000A is error or not loaded.
D3	Reserved	Reserved.
D2 ~ D0	Sequence	Data packet sequence number. The sequence number is '1' or '0' when the network server sends connecting command (CONT). After connection, every time server sends a data packet, the sequence number increases by '1' (MOD 8). The sequence number of returned data packet from the PS2000A is same as the received one. Re-sent data packet does not change the sequence number.

10.2 COMMAND

10.2.1 Command Structure and Function

The command byte in the data package, with which the PS2000A communicates with the Network Server, consists of two parts, the command state and the command, as shown in table 22.

Table 22

Command State			Command				
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
More	Error	NO_ACK	CONT				
			DICN				
			READ				
			RDAL				
			RDPR				
			RDEV				
			WRIT				
			WRPR				
			ENEV				
			DSEV				
			DISV				
			RESET				

10.2.1.1 Command State

D7 (More): Data state indication bit, which is set by the PS2000A and valid when the PS2000A responds to a READ command from the Network Server.

D7=1: The object has not finished transmitting the data and the Network Server should continue to transmit the READ command to read the data.

D7=0: The data transmission of the object has been finished.

D6 (Error): Flag bit indicating command error which is set by the PS2000A and valid in all return commands.

D6=0: Indicating that the command sent from the Network Server has been successfully executed by the PS2000A.

D6=1: Indicating that the command sent from the Network Server cannot be executed by the PS2000A.

D5 (NO_ACK): ACK control bit. The Network Server sets this bit.

D5=1: Indicating that the PS2000A should execute the command of the Network Server, without reporting the execution results back to the server.

D5=0: Indicating that the PS2000A should execute the command of the Network Server and report the execution results to the server.

10.2.2.2 The Function of Command

The commands and the function of the commands that the PS2000A can respond to from the Network Server are shown in Table 23.

Table 23

Mnemonic Symbol	Code D4~D0	Function
CONT	1Fh	Network Server requests connection to the PS2000A.
DICN	1Eh	Network Server requests disconnection from the PS2000A.
READ	00h	Network Server reads data information from the device object.
RDAL	06h	Network Server reads all variables and data associated with events from the device.
RDPR	02h	Network Server reads the property table of the device.
RDEV	04h	Network Server inquires about the event of the device.
WRIT	01h	Network Server writes data information to the device object.
WRPR	05h	Network Server updates the device property table of the PS2000A.
ENEV	03h	Enable the device to report a certain event to the Network Server.
DSEV	13h	Disable the device to report a certain event to the Network Server.
DISV	10h	Network Server inquires and registers new devices.
RESET	18h	Network server hardware reset the PS2000A

10.3 Physical Connection between PS2000A and Gateway

The PS2000A can be physically connected to the network server in RS-232, RS-485 or Modem mode.

10.3.1 RS-232 Mode

Because there is not an RS-232 driver/receiver inside the PS2000A, its input and output signal can connect to the RS-232 port of the network server only after the level is converted, as shown in Figure 20.

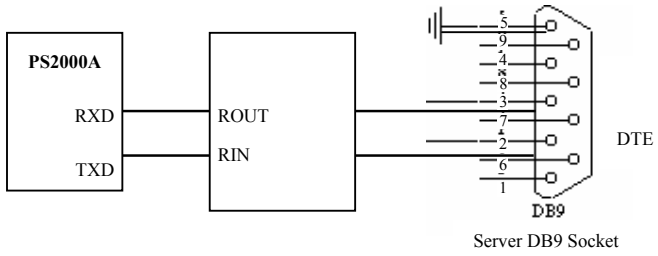


Figure 20 The simplified diagram of PS2000A connecting to the external RS-232 driver/receiver

10.3.2 RS-485 Mode

Because there is no RS-485 driver/receiver inside the PS2000A, its input and output signal is needed to be connected to the external RS-485 driver/receiver to connect to the RS-485 port of the network server, as shown in Figure 21.

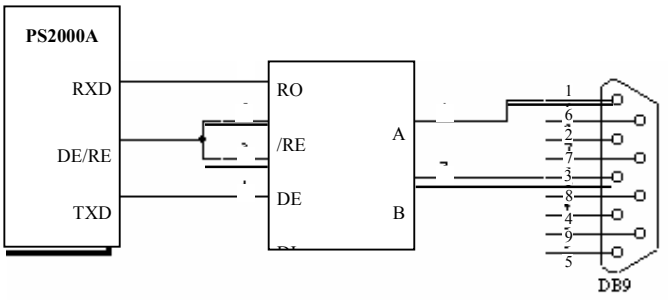
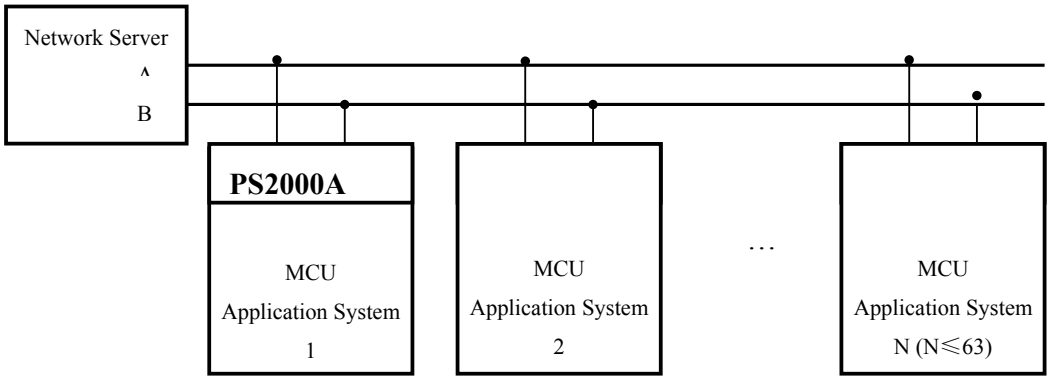


Figure 21 PS2000A is connected to the external RS-485 driver/receiver

Multiple PS2000A's may be connected to the RS-485 bus, and together with the network server to form an RS-485 net. Within this net, the maximum amount of PS2000A's is 63, as shown in Figure 22.



Note: The RS-485 interface circuit is not shown in the Figure

Figure 22 Multiple PS2000As and a Network Server form an RS-485 net

10.3.3 Modem Mode

The PS2000A provides a communication interface for the standard Modem, therefore the PS2000A may use the standard Modem to communicate with the Modem of the network server via Public Switch Telephone Network (PSTN), as shown in Figure 23.

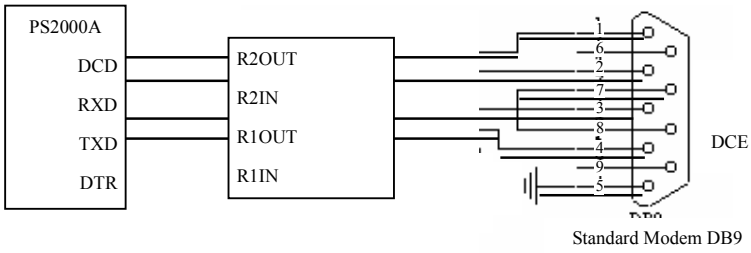


Figure 23 The simplified diagram of interface circuit between PS2000A and standard Modem with the external RS-232 driver/receiver

The PS2000A provides the RTS and CTS signal lines of a standard Modem, but may not use them actually. These two lines should be shorted.

Modem Set in Call Mode

The MCU application system may set the Modem of the PS2000A in call mode, while the MCU should send the telephone number to the PS2000A. The PS2000A will connect to the network server by dialing this telephone number.

The PS2000A dials to call the Modem of the network server. If the Modem does not answer within 30 seconds, this action fails. If the Modem of the network server answers but the PS2000A has not connected to the network server successfully within 30 seconds, then the current action also fails.

After the PS2000A connects to the network server successfully, if the PS2000A has not received the valid data transmitted by the network server within 1 minute, the PS2000A will command the Modem to hang-up to finish the communication connection.

Modem Set in Answer Mode

The MCU application system may set the Modem of the PS2000A in answer mode, then the PS2000A enters into the status waiting for the network server's Modem dialing.

Once the network server's Modem dials and calls the PS2000A, the PS2000A should answer and respond after the first ringing has finished. If the PS2000A has answered and responded but not connected to the network server successfully within 30 seconds, then the current Modem connection fails.

After the PS2000A connects to the network server successfully, if the PS2000A hasn't received the valid data transmitted by the network server within 1 minute, then the PS2000A will command the Modem to hang up to finish the communication connection.

Once the PS2000A enters the waiting call status, it can not respond to other requests from the MCU application system. If the MCU wants to make the PS2000A exit the waiting call status, it may input a low-level signal at /CS pin as shown in Figure 24.

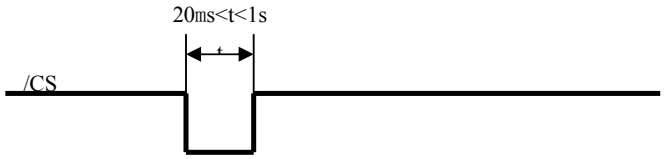


Figure 24 The signal on CS to exit PS2000A from the waiting call status

10.4 Gateway Discovery Process

The network server can only access the PS2000A after it is registered. The network server uses DISV to inquire if there are non-registered PS2000A's connected to the network system. If this is the case, these new PS2000A's will be registered automatically.

The format of the command sent from the network server is shown in Table 25. In the data package of the DISV command the network server sends to the PS2000A, the ID of the device should be 0, or use the super device ID (0FFH). Otherwise, the PS2000A will not respond to the command.

Table 25

Byte	Package content	Description
4	ESC (0x1b)	Package header
	SOH (0x01)	
	ESC (0x1b)	
	SOH (0x01)	
4	Server ID Code	Server identification code
8	Device ID code	Device identification code
1	00h	State information byte
1	DISV (10h)	Discover command
1	00h	Null
1	00h	Null
4	ESC (0x1b)	Packet trailer
	ETB (0x17)	
	CRC	CRC value, 2 bytes (MSB first)

NOTE: TO COMMUNICATE WITH THE PS2000A, THE NETWORK SERVER AND THE DPTGT CAN USE THE DISV COMMAND TO REGISTER THE DEVICE AND GET THE DEVICE ID.

When the PS2000A receives the command, it should be estimated whether the network server accesses it. If not, the PS2000A will be considered as a non-registered device, it will respond to the command.

The PS2000A responds to the DISV command different than other commands. The PS2000A needs to have a delay time, which ranges form 0 to 500ms.

In the delay time, the PS2000A listens whether other communication controllers of the network respond to the command or not. If not, the delay is over and the PS2000A sends the data package to the network server. If so, the PS2000A will not respond to the command, unless the network server sends the DISV command again.

If multiple non-registered PS2000As connects to the network at the same time, the data collision can be minimized with the help of the function.

After the PS2000A responds to the DISV command of the network server successfully, it is not considered that the PS2000A has been registered in the network server, unless the network server has accessed the PS2000A by the use of the correct device ID code. If the

network server doesn't access the PS2000A, the PS2000A will respond to the DISV command when the network server sends it next time.

11 Generating and Loading the Device Property Table

11.1 Generating the Device Property Table

For the PS2000A, a key step is that the device property table is generated correctly and written in the PS2000A successfully. With the help of P&S DataCom's device property table generating and downloading tool, the device property table can be generated through the MCU application system, and the table can be loaded into the PS2000A remotely or locally. For the detailed use of P&S DataCom's device property table generating and loading tool, refer to the WebChip™ Device Property Table Generating Tool User's Manual.

11.1.1 Device Property Table

Any device object based on a MCU can be defined with the following five types of objects: Constant, Variable, Event, Function, and File. For detailed descriptions of these five types, refer to the MCUnet™ 1.0 Protocol.

The Device Property Table (DEVICE_PROP_TABLE) sets up the object-oriented controlling relationship between the MCU and the network server and is a complete description of a device. The network server accesses the device through its DEVICE_PROP_TABLE.

For the detailed description of the Device Property Table, please refer to the MCUnet™ 1.0 Protocol.

In the PS2000A, the bit definition of the Device Property Description word (DEVICE_PROP_DESP) is shown in Table 26.

Table 26

D15	D14	D13	D12~D10	D9	D8	D7	D6~D0
1	0	0	Reserved (always 0)	0	0	0	Number of device objects (Obj_Num)

In the PS2000A, the object name character string supports ASCII code only. Address length is 16 bits. The expression of word (or double word) is from high to low in sequence.

The number of objects in the device supported by the PS2000A should be less then 128.

In the MCUnet™ 1.0 protocol, there are 12 user-defined memory types in the item "Storage Type" of the Device Property Table. The user-defined memory may be any memory type (Internal RAM, External RAM, NVS and PROM of MCU). It is available for the user to easily distinguish the different memory types when being accessed.

For instance, in the MCS-51 application system shown in Figure 25, the microcontroller connects to two X25C02s using its I/O ports.

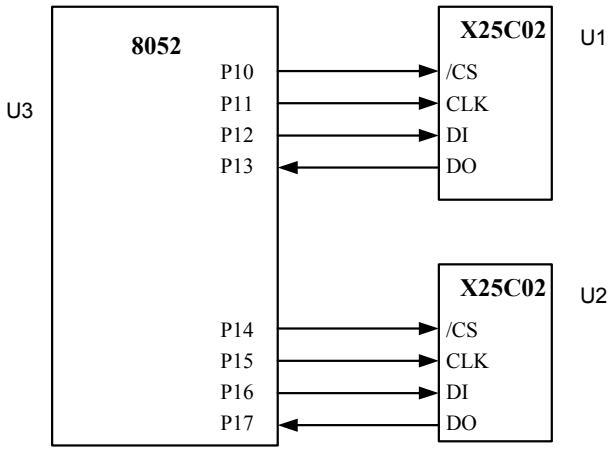


Figure 25 An MCS-51 Application System

In Figure 25, U1 and U2 are NVMs (Non-Volatile Memory). They both connect to the MCU through an SPI interface and are able to Read/Write with the same access address of 0~0FFh. If both U1 and U2 are defined as NVM of MCU, they can't be distinguished.

In this case, these two memories can be defined as "user-defined memory", for example, U1 is defined as 0C4h and U2 as 0C5h, so that the network server can easily distinguish them.

The device password in the PS2000A should be up to 13 characters of ASCII string and end with "NULL". Otherwise, the password is invalid.

The super password in the PS2000A should be up to 7 characters of ASCII string and end with "NULL". Otherwise, the password is invalid.

The PS2000A supports up to 17 digits (or other valid characters) of ASCII string for a telephone number and it should end with "NULL".

11.2 Loading the Device Property Table

The loading of the Device Property Table in the PS2000A can be accomplished by the network server or the Device Property Table Generating Tool (DPTGT).

The Device Property Table is the most important information in the PS2000A. It should not be modified without authority. The Device Property Table of the PS2000A is protected through a super password, which can be set and modified by the network server or the DPTGT.

The super password of the PS2000A is specified as an ASCII string of 7 characters ended with "NULL". The PS2000A will refuse to receive a super password which is less or greater than 7 characters.

The name of the super password of the PS2000A should only be "$Super_PW+NULL" of ASCII code.

The super password of the PS2000A is a write-only variable. The network server or the DPTGT cannot read the super password of the PS2000A.

The PS2000A can connect to the network server or the DPTGT to download the Device Property Table without connecting to the MCU application system. But at this case, the PS2000A can communicate with the network server or the DPTGT in RS-232 mode, the baud rate is fixed as 9600bps. The length of receiving and transmitting buffer section is fixed as 56 bytes. The PS2000A needs the minimal physical connection as shown in Figure 27.

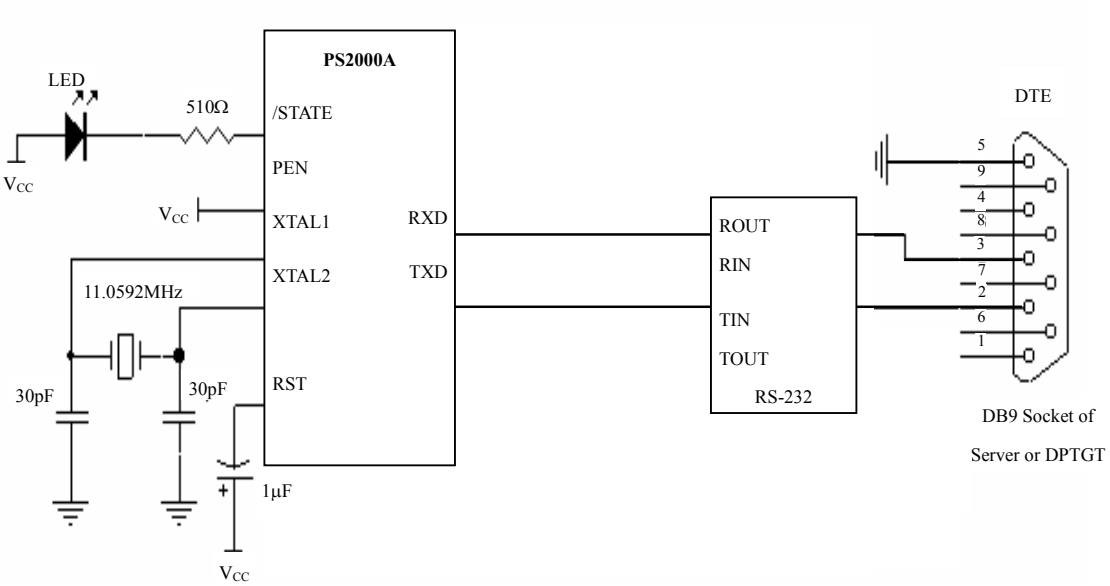


Figure 27 The simplified diagram of minimal connection for PS2000A downloading device property table

In the Figure 27, the output signal STATE drives the LED to indicate the state of downloading the device property table and determine whether the device property table is loaded in the PS2000A successfully.

The PS2000A may download the device property table after connecting to the MCU application system. The MCU may select the PS2000A communicate with the network server or the DPTGT in another mode and at another speed. But in the case, the size of receiving and transmitting buffer section of the MCU application system cannot affect the buffer section of the PS2000A which is fixed as 56 bytes.

The network server or the DPTGT uses two methods to load the device property table in the PS2000A:

1. local connection via RS-232 or RS-485

2. remote connection via Modem

Here, we discuss the network server or the DPTGT loading of the device property table locally in the PS2000A.

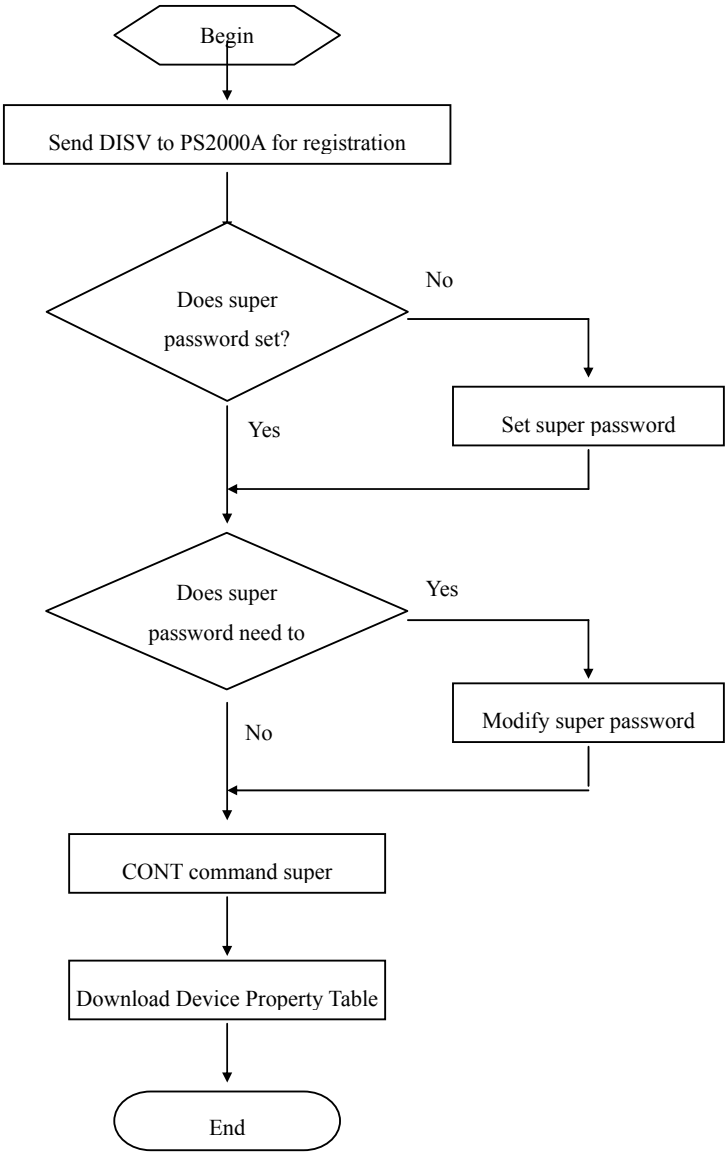


Figure 28 The process of loading device property table locally

11.2.1 Setting or Modifying the Super Password

The PS2000A does not have a super password when leaving the factory. A super password must be set before a Device Property Table can be downloaded.

The network server and the DPTGT must use the WRIT command to set the super password, the format of the data package is shown in Table 45. In the Table 45, because the PS2000A has not set the super password at first, it does not care for the old password and uses the new one directly. The network server or the DPTGT may fill in 7 characters in the old password, or may not send it.

If the PS2000A has a super password, the network server and the DPTGT must send out the new password together with the old one when modifying a super password. Only when the old super password matches the original one in the PS2000A will the new super password be valid. The network server and the DPTGT use the WRIT command to modify a super password as shown in Table 29.

Table 29

Byte	Package content	Description
4	ESC (0x1b)	Package header
	SOH (0x01)	
	ESC (0x1b)	
	SOH (0x01)	
4	Server ID Code	Server identification code
8	Device ID code	Device identification code
1	State information byte	Refer Table 21
1	WRIT (01h)	Write command
10	$Super_PW+NULL	Name of super password
7	Character string of old super password	Character string of old super password
7	Character string of new super password	Character string of new super password
4	ESC (0x1b)	Packet trailer
	ETB (0x17)	
	CRC	CRC value, 2 bytes (MSB first)

When the PS2000A sets the super password, the network server and the DPTGT continue the following operation only after receiving the returned data from the PS2000A. If they don't receive any information from the PS2000A within 15 minutes, the current set-up fails. The PS2000A is powered off and then powered on again.

When setting the new super password, the PS2000A will check if the Device Property Table storage is empty first. If not, it will take about 10 seconds for the PS2000A to clear the storage. Then, the PS2000A will return a successful message "setting super password to the network server or the DPTGT." If the Device Property Table storage is empty, the PS2000A will return a successful message "setting the super password to the network server or the DPTGT."

When modifying the super password, if the password is modified successfully, the PS2000A will return a successful message of modifying the super password to the

network server or the DPTGT.

The network server or the DPTGT cannot use the WRIT command to clear the super password.

The format of the data package returned from the PS2000A is shown in Table 30.

Table 30

Byte	Package content	Description
4	ESC (0x1b)	Packet header
	SOH (0x01)	
	ESC (0x1b)	
	SOH (0x01)	
4	Server ID Code	Server identification code
8	Device ID code	Device identification code
1	State information byte	See Table 21
1	WRIT (01h)	Write command
10	$Super_PW+NULL	Super password name
2	00h, 07h	Received super password characters of 7 bytes
4	ESC (0x1b)	Packet trailer
	ETB (0x17)	
	CRC	CRC value, 2 bytes (most significant byte first)

If the super password is not set up or modified successfully, the data returned from the PS2000A is shown in Table 31 or the PS2000A may not return any data.

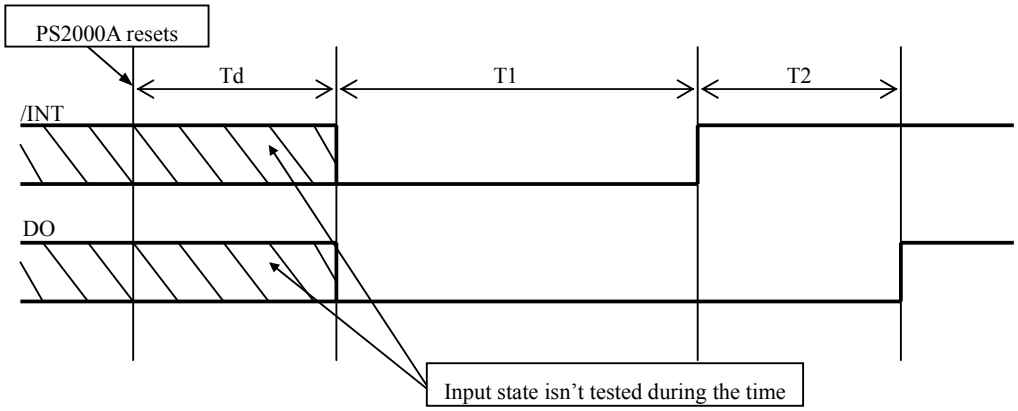
Table 31

Byte	Package content	Description
4	ESC (0x1b)	Packet header
	SOH (0x01)	
	ESC (0x1b)	
	SOH (0x01)	
4	Server ID Code	Server identification code
8	Device ID code	Device identification code
1	State information byte	Refer Table 21
1	41h	Write command WRIT + ERROR symbol
10	$Super_PW+NULL	Super password name
4	ESC (0x1b)	Packet trailer
	ETB (0x17)	
	CRC	CRC value, 2 bytes (most significant byte first)

11.2.2 Clearing the Super Password

The super password is so important that users should use it carefully.

Although the PS2000A cannot use the WRIT command to clear the super password, it provides a method of loading a timing signal to clear the password. After the PS2000A is powered on reset, the internal super password will be cleared by inputting a timing signal on the DO and INT pin of the PS2000A. The timing of clearing the password of the PS2000A is shown in Figure 29.



Td=300ms±10ms; T1≥500ms; 100ms≤T2≤300ms

Figure 29 The timing diagram of clearing the super password of the PS2000A

After the super password is cleared, the network server and the DPTGT must set the super password again. In this case, the PS2000A is allowed to write a new Device Property Table.

In the Figure 29, the timing signal can be input on the pins of the PS2000A correctly only through the MCU application system. It should be noted that the DO and /INT pin of the PS2000A are inputs during this time. And, they are outputs in the normal condition. The designers of the MCU application system should determine if they will use this method to clear the super password in the PS2000A or not.

To improve the security of the system, the PS2000A doesn't provide the method of clearing the super password remotely.

11.2.3 Loading the Device Property Table

The network server or the DPTGT must use CONT to connect to the PS2000A with the super password and get its receiving and transmitting buffer section's parameters before loading the device property table in the PS2000A. If the network server or the DPTGT don't connect to the PS2000A, the PS2000A will refuse to receive the downloaded device property table.

When the network server connects to the PS2000A with the super password, the receiving and transmitting buffer section of the PS2000A is fixed as 56 bytes.

The DPTGT uses WRPR to load the device property table data into the device, the data

package of loading the table is shown in Table 32.

Table 32

Byte	Package content	Description
4	ESC (0x1b)	Package header
	SOH (0x01)	
	ESC (0x1b)	
	SOH (0x01)	
4	Server ID code	Server identification code
8	Device ID code	Device identification code
1	State information byte	See Table 21
1	WRPR (85h or 05h)	Command to write device property table (note)
1	NULL	Always NULL
7	Super password characters	Super password characters
2	Offset	2 bytes (MSB first)
n	Device property table data	Device property table data
4	ESC (0x1b)	Packet trailer
	ETB (0x17)	
	CRC	CRC value, 2 bytes (MSB first)

NOTE: IF THE DPTGT CANNOT SEND ALL DATA OF THE DEVICE PROPERTY TABLE IN A SINGLE PACKAGE, IT CAN SEND THEM IN MULTIPLE PACKAGES.

WHEN THERE IS MORE DATA TO BE SENT, WRPR=85H;

WHEN THERE IS NO MORE DATA LEFT, WRPR=05H.

ONLY WHEN THE DPTGT SETS WRPR VALUE CORRECTLY, CAN THE PS2000A RECEIVE DATA OF THE DEVICE PROPERTY TABLE RELIABLY.

If the super password in the WRPR data package matches the one in the PS2000A, it will receive the data of the device property table. The data package returned from the PS2000A is shown in Table 33.

After the PS2000A receives the first data package sent from the network server or the DPTGT, it should clear the original device property table first, which needs about 10 seconds. The network server and the DPTGT continue the following operation only after receiving the returned data from the PS2000A. Otherwise, the data will be lost. If they don’t receive any data from the PS2000A within 15 minutes, the current operation fails. The PS2000A is powered off and then powered on again.

Once the device property table is loaded successfully according to the connection as shown in Figure 27, the indicator will blink 10 times continuously at a frequency of 1Hz. The PS2000A will restart 5 seconds later after the indicator turns off.

Table 33

Byte	Packet content	Description
4	ESC (0x1b)	Package header
	SOH (0x01)	
	ESC (0x1b)	
	SOH (0x01)	
4	Server ID Code	server identification code
8	Device ID code	Device identification code
1	State information byte	Refer Table 21
1	WRPR (05h)	Command to write device property table
1	NULL	Always NULL
2	Byte number of the received data	Including super password characters, offset and the data of device property table, 2 bytes (MSB first)
4	ESC (0x1b)	Packet trailer
	ETB (0x17)	
	CRC	CRC value, 2 bytes (MSB first)

If the super password in the data package does not match the one in the PS2000A, the data of the device property table will be refused by the PS2000A. The package returned from the PS2000A is shown in Table 34

Table 34

Byte	Packet content	Description
4	ESC (0x1b)	Package header
	SOH (0x01)	
	ESC (0x1b)	
	SOH (0x01)	
4	Server ID Code	server identification code
8	Device ID code	Device identification code
1	State Information Byte	Refer Table 21
1	WRPR (05h)	Command to write device property table
1	NULL	Always NULL
2	00h, 00h	Byte received is 0
4	ESC (1x1b)	Packet trailer
	ETB (0x17)	
	CRC	CRC value, 2 bytes (MSB first)

In the process of loading the device property table, if the super password ($Super_PW) in the WRPR data package is wrong, the PS2000A will terminate the loading process immediately, and restart the operation. But the new device property table may be overlaid by the old device property table.

In the process of loading the device property table, if the data package of another command is received, the PS2000A will terminate the loading process immediately, and restart the operation. But the new device property table may be overlaid by the old device property table.

When either of the above situations occurs, both the network server and the DPTGT must reload the device property table.

11.2.4 Loading the Device Property Table Remotely

The network server and the DPTGT load the device property table remotely as shown in Figure 30.

In the Figure, the remote loading process is the same as the local one except for adding the connection with the remote Modem of the PS2000A.

The network server and the DPTGT must dial in Call Mode to connect to the Modem of the PS2000A.

Whether the PS2000A has loaded the device property table or not, the MCU application system sets it into Modem Answer Mode. If the PS2000A has not loaded the device

property table, whatever mode the MCU application system sets it, the PS2000A will enter into Modem Answer Mode automatically.

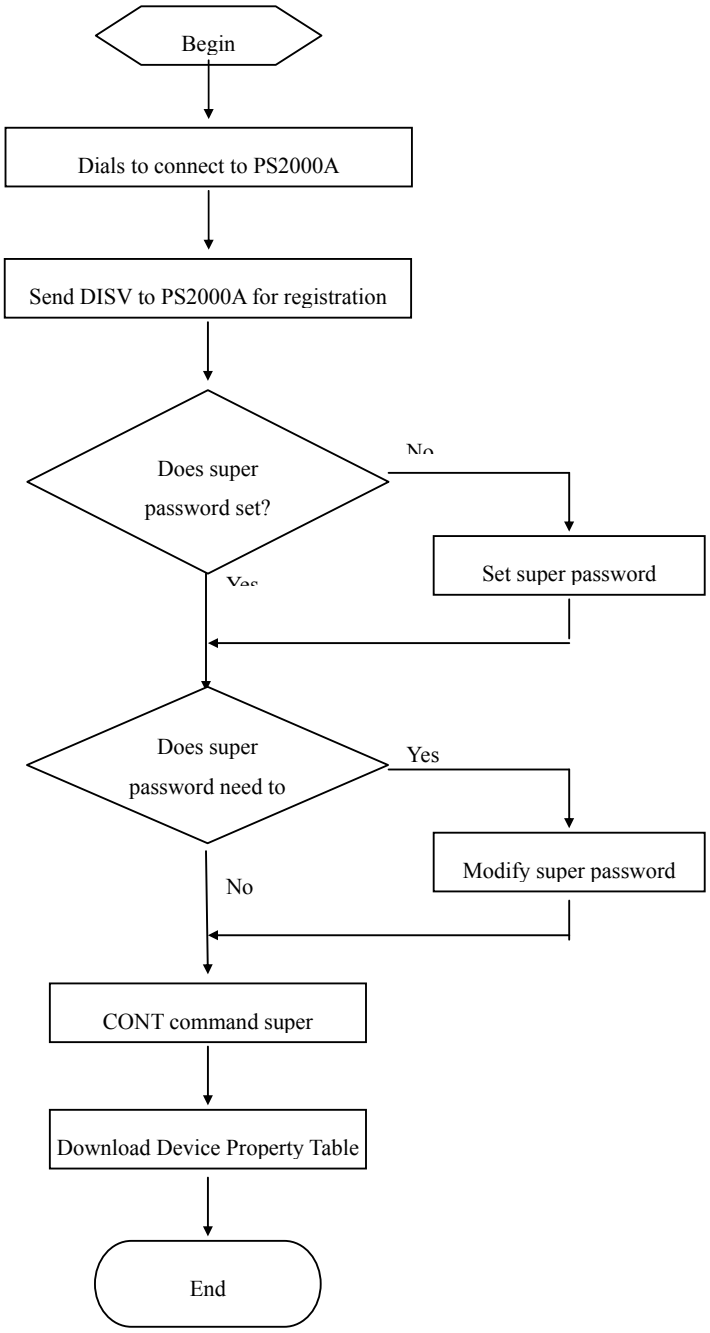


Figure 30 The process of loading device property table remotely

After the loading of the device property table is finished, the PS2000A will disconnect with the network server or the DPTGT, and then restarts.

12 Reference Documents

WebChip™ ASDK User's Guide

Interfacing the User MCU to the WebChip™

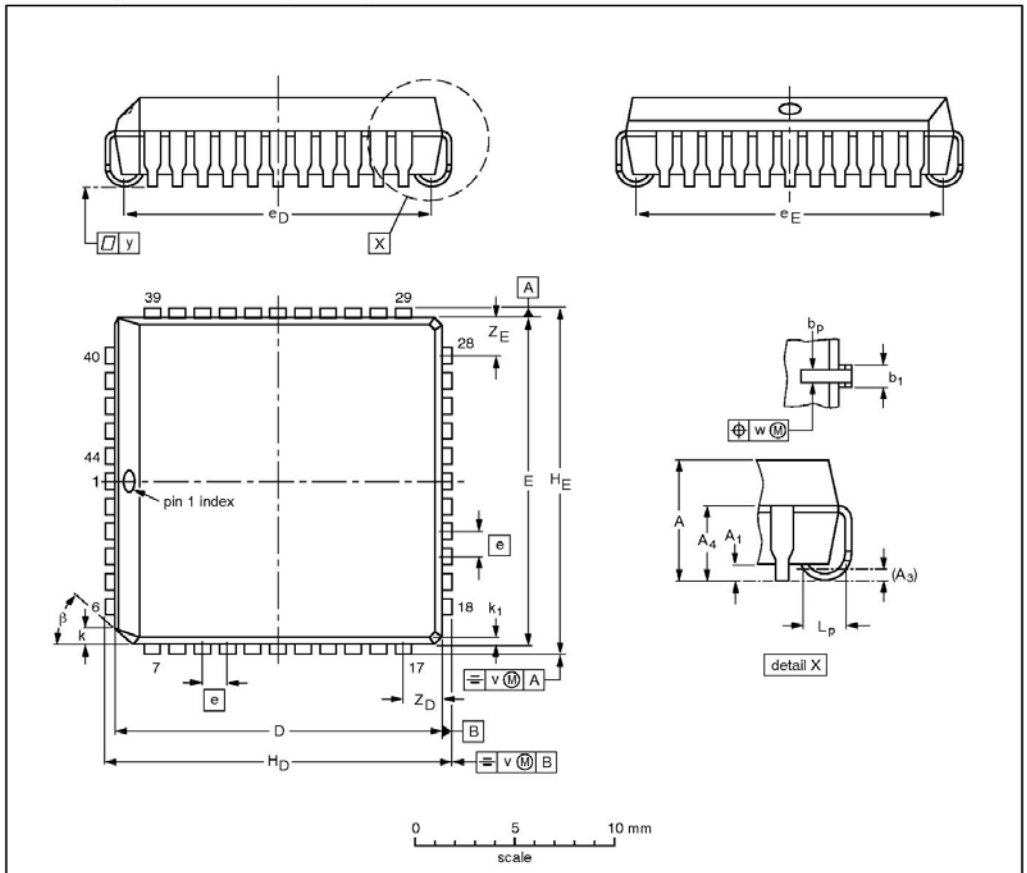
MCUnet™ 1.0 Protocol

MCUap™ 1.0 Protocol

13 Package Information

Package drawing with mechanical dimensions.

PLCC44: plastic leaded chip carrier; 44 leads



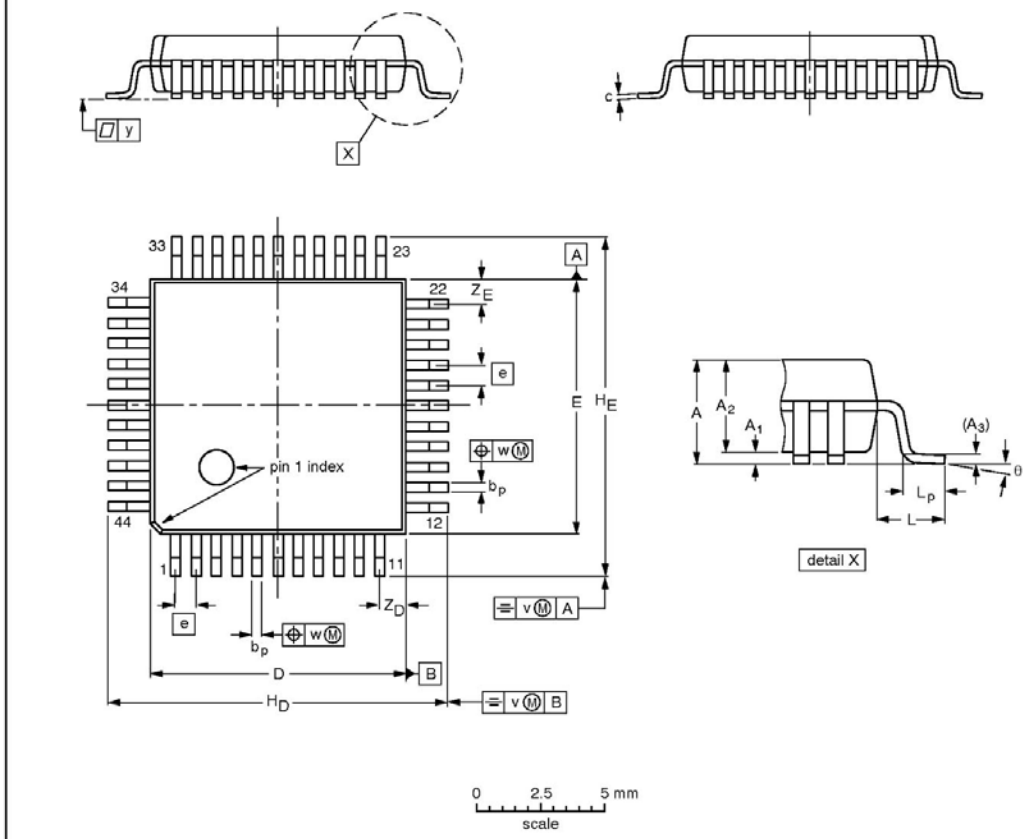
DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	k	k_1 max.	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.66 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	0.51	1.44 1.02	0.18	0.18	0.10	2.16	2.16	45°
inches	0.180 0.165	0.020	0.01	0.12	0.021 0.013	0.032 0.026	0.656 0.650	0.656 0.650	0.05	0.630 0.590	0.630 0.590	0.695 0.685	0.695 0.685	0.048 0.042	0.020	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

LQFP44: plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.60	0.15 0.05	1.45 1.35	0.25	0.45 0.30	0.20 0.12	10.10 9.90	10.10 9.90	0.80	12.15 11.85	12.15 11.85	1.0	0.75 0.45	0.20	0.20	0.10	1.14 0.85	1.14 0.85	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

Notice

The contents of this document are provided in connection with P&S DataCom Corporation products. Information furnished by P&S DataCom is believed to be accurate and reliable. However, P&S DataCom makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication. P&S DataCom assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

P&S DataCom reserves the right to make changes to the referenced products, these specifications, and the product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” P&S DataCom reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

MCUnet™ 1.0 Protocol and MCUap™ 1.0 Protocol described in this documentation may only be used in accordance with the terms of related license agreement.

Software is provided “as is”. In no event shall P&S DataCom be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage. P&S DataCom assumes no responsibility for any third-party software. Use, representations, warranties, specifications, and license of any and all third-party software shall be governed and controlled by documents and agreements maintained by the third-party software vendor.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in P&S DataCom’s Terms and Conditions of Sale for such products, P&S DataCom assumes no liability whatsoever, and P&S DataCom disclaims any express or implied warranty, relating to sale and/or use of P&S DataCom products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

P&S DataCom Corporation’s products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of P&S DataCom’s product could create a situation where personal injury, death, or severe property or environmental damage may occur.