

RAYON's IP serial device server product line

Traditionally we will use serial port to connect PC with equipment. When the distance between PC and equipment is within 50 feet, we may use RS232 interface to connect. When the distance between PC and equipment is within 4000 feet, we may use RS422 interface for full duplex communication or RS485 interface for half duplex communication. When the distance between PC and equipment is over 4000 feet, we may use other converter and other method for such environment. Now network connection is very popular in our application environment. If we could change serial port data to IP packet, then we can send serial port data to any network reachable location. What we need is IP serial device server to exchange serial port data and IP packet.

Model	RS232	isolated RS232	isolated RS422/485	Total
GPORT104	4	×	×	4
GPORT108	8	×	×	8
GPORT116	16	×	×	16
APORT200	2	×	×	2
APORT101	1	×	1	2
APORT020	×	2	×	2
APORT400	4	×	×	4
APORT040	×	4	×	4
APORT004	x	x	4	4
APORT202	2	×	2	4
APORT022	×	2	2	4
APORT800	8	×	×	8
APORT080	x	8	x	8
APORT008	x	x	8	8
APORT620	6	2	×	8
APORT602	6	×	2	8

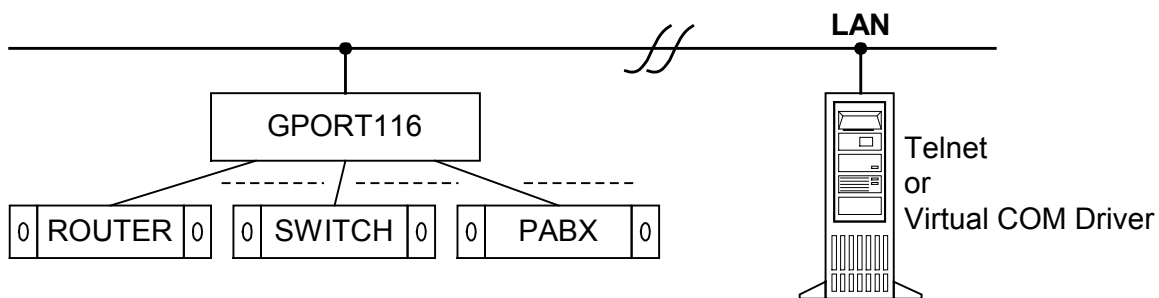
RAYON's IP serial device server product line can be GPORT series, APORT series and IPORT series. In GPORT series product line we can support standard serial port application mode (RTTY virtual COM port mode, TCP server mode, TCP client mode and UDP send/receive mode) and special Linux SDK mode for user to develop their proprietary application software. Because GPORT series product line is one box with Linux based hardware. So every standard serial port application mode is one

application program in Linux system. Generally our standard serial port application mode can meet user's serial port application requirement. But some people may not just use raw serial port data to be converted and transmitted in network. They may need to process serial port data and send in network. In this condition our standard serial port application mode may not meet their requirement and it is not easy for us to provide such proprietary software. Fortunately, Linux system is open source environment. All the software developed in PC Linux system can be ported to other hardware platform. So user can develop their proprietary application software in PC Linux system. The final execution file can be ported to our GPORT Linux box. Then our GPORT Linux box can be the only one product for user's proprietary application environment. User can protect their software value and system security. Right now we have GPORT104/108/116 to support 4/8/16 RS232 serial ports. User can choose the suitable one for their application. But user needs to keep in mind that our standard serial port application mode will be not available for Linux SDK mode.

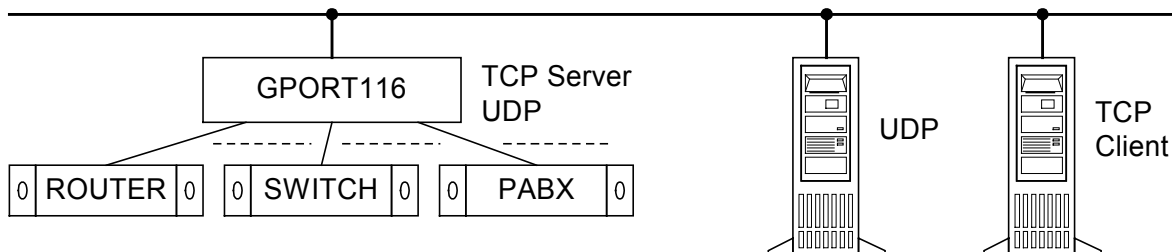
APORT series product line will be focused in different serial port interface type to meet different application environment. For cost consideration we can have RS232 interface to be used. For security consideration we can have GROUND isolated RS232 for local connection and GROUND isolated RS422/RS485 for remote connection. RS232 is common mode signal connection. We will connect signal GROUND in both device. When we just have one RS232 serial port in one APORT box to connect, we may have little problem in this condition. When we have multiple serial port in APORT box, we may need to concern this condition. Because each RS232's equipment will be connected with APORT box and the signal GROUND will connect together. In rigid application environment each RS232's equipment may have big power consumption to generate GROUND bounce. Such signal ground potential difference between two RS232 equipments may have big problem in data transmission error or IC damaged. Then we need to have GROUND isolated in each RS232 equipment. In this condition we may need GROUND isolated RS232 serial port for APORT box. APORT200/400/800 series product line will support 2/4/8 serial port with different interface type combination to meet cost and security consideration.

In Windows system APORT box can support two virtual COM port driver type for user. As we know that UDP method can promise IP packet to be sent as fast as possible and TCP method can promise IP packet to be sent error free. When we set one serial port to be RTTY mode, we will use UDP type virtual COM port driver in Windows. In this condition we can send data between PC and APORT box with low latency time. It means that the data received in APORT will be sent to PC as fast as possible. So we can

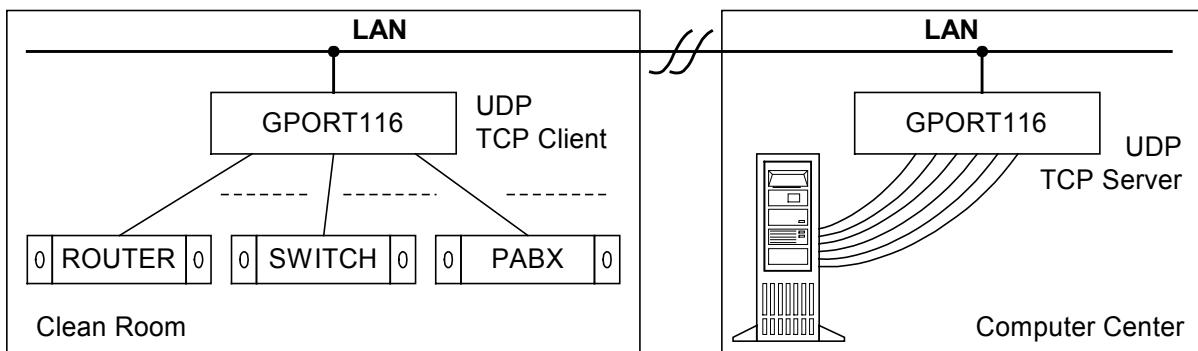
respond such data quickly. But UDP packet may be discarded or masked in network environment. In such condition (UDP packet can be discarded upon network bandwidth is not enough) it is not suitable for us to use RTTY virtual COM driver in Windows system. We can have the other TPORT virtual COM driver in Windows system. We will set one serial port in TCP server mode. Then we can send TCP packet between PC and APORT box. We just need to specify target TCP port number in network to be transmitted everywhere (firewall or routing table acceptable). Then we can have serial port data to be duplicated between PC and APORT. But we can not promise the data latency time.



Virtual COM driver in Windows system can let your serial port in IP serial device server box as your COM port in PC. It is easy way for current COM port application software in Windows system to use serial port data over network.



WinSocket function is easy way to send raw serial port data in PC.



TCP client to TCP server mode is easy way to duplicate raw serial port data over

network. UDP send/receive mode is fastest way to duplicate raw serial port data over local area network (no use Gateway condition).

Model	RS232	RS422/485	isolated RS422/485	Total
IPOINT111	1	1	×	1
IPOINT112	2	2	×	2
IPOINT114	4	4	×	4
IPOINT118	8	8	×	8
APOINT131	X	1	X	1
APOINT132	X	2	X	2
IPOINT134	×	4	×	4
IPOINT138	X	8	×	8
IPOINT104P	4	X	×	4
IPOINT108P	8	x	X	8
IPOINT134I	X	×	4	4
IPOINT138I	×	X	8	8

IPOINT series product line can support RS232/RS422/RS485 interface type settable in each serial port. We can use DIP switch to set target interface type for all serial ports. Or we can use WEB and console setup to set each serial port interface type independently.

User may have other dedicated application environment, then we can talk in Email to check such condition. For example we may have serial port to be set in RS422 or RS485 interface type in APOINT131 and APOINT132 model. But user may need to use RS485 interface type only (some operator may have wrong setting for RS422 mode by mistake with problem in application environment). Then we may have APOINT1311 and APOINT1321 model to be special model of APOINT131 and APOINT132 model to support RS485 interface type only.