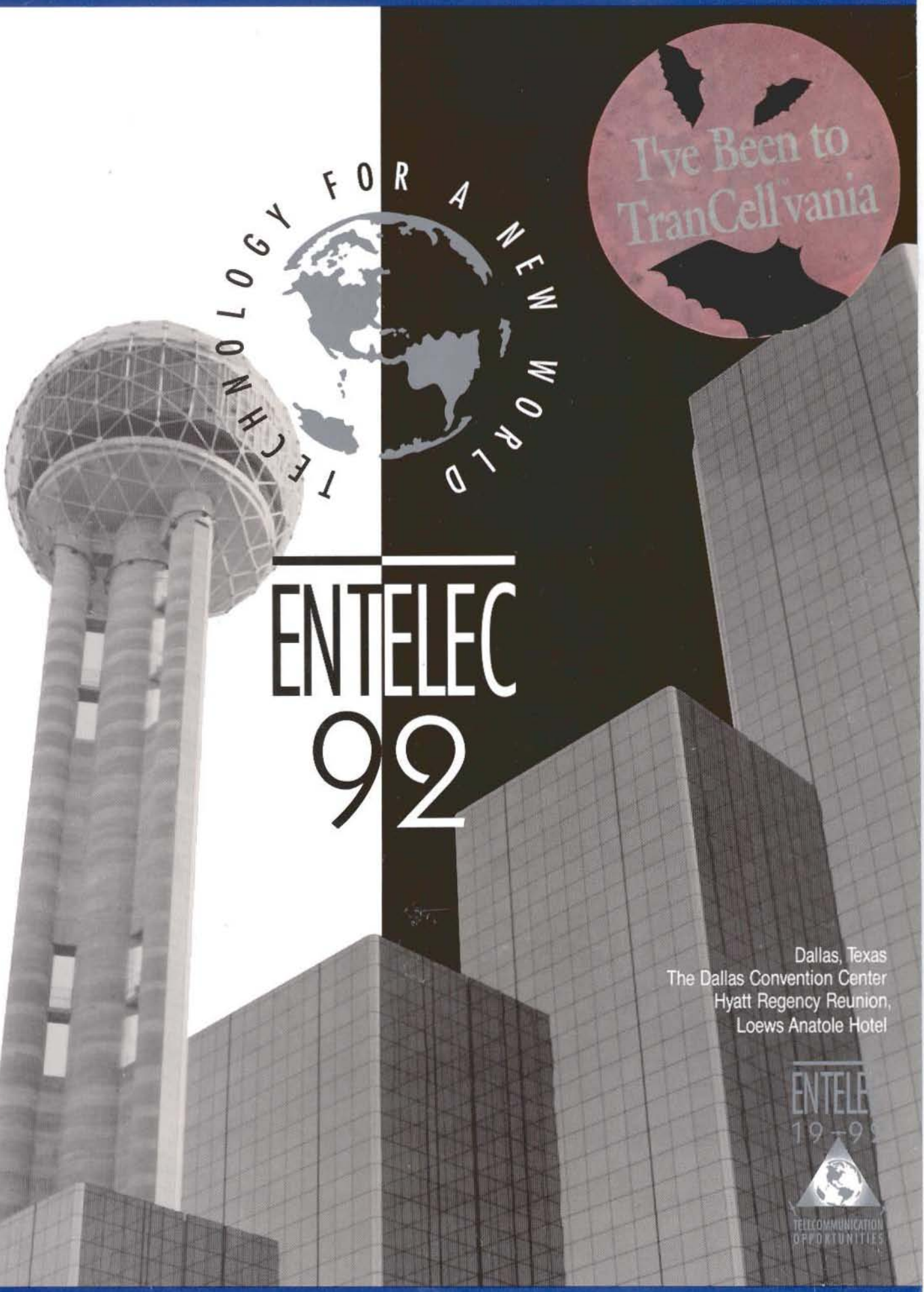
TECHNICAL PAPERS



TECHNOLOGY FOR A NEW WORLD

ENTELEC 92

Dallas, Texas
The Dallas Convention Center
Hyatt Regency Reunion,
Loews Anatole Hotel

ENTELEC 19-92

TELECOMMUNICATION OPPORTUNITIES

MARCH 29 — APRIL 1, 1992

ENTELEC '92 TECHNICAL PROGRAM
Table of Contents

CONTROL SYSTEMS

EMERGING TECHNOLOGIES

GENERAL

SCADA Applications for Packet Radio Controllers

ARTHUR M. ZATARAIN, P.E.
TEST, Inc.

BRIEF

This paper is an introduction to Packet Radio Controller (PRC) technology and its application in remote Supervisory Control and Data Acquisition (SCADA) systems. The discussion will provide basic information on Packet technology, its history, current capability and expected future developments. Specific application experience will also be presented to demonstrate the effective application of the devices to several actual SCADA installations.

This presentation is presented at an introductory level, where extensive electronics or telemetry knowledge is not assumed. The intended audience is SCADA system managers, system developers and applications personnel. Therefore, some awareness of overall telemetry concepts and goals is assumed, as is familiarity with the basics of radio communications systems.

Packet radio technology itself is only discussed at the basic level. For a more detailed explanation of the technology, please refer to the reference list at the end of this document.

INTRODUCTION

Computer Data Communications can be performed with voice (audio) type equipment by use of a device called a Modem. The Modem, short for Modulator-Demodulator, changes the computer's data (sent as electrical 1s and 0s) into audible tones that can be carried easily by traditional communications systems. Many types of modems are available for use in radio, telephone and direct connection links between two or more data devices. This presentation discusses a special modem device called a Packet Radio Controller, or PRC, which provides many functions necessary to use radio equipment for telemetry applications.

Packet technology has been slowly gaining acceptance in a number of fields, most notably amateur radio. Other commercial and industrial applications have also emerged, although no general acceptance of the technique has been formalized by any industrial group. This will surely change as more systems are installed and their benefits are more widely recognized.

PACKET CONTROLLER BENEFITS

When considering any new technology, it is always a good idea to get an idea of the advantages to be gained over existing methods. Packet Technology will be used in place of traditional radio or microwave modems to send and receive digital data, often in some form of remote SCADA system. Some of the benefits the author has identified for Packet Controllers, when compared to existing methods, are as follows:

1. Allows telemetry systems to operate with sub-standard radio systems that would otherwise be unsuitable for SCADA applications.
2. Simplified software generation or modification for control programs because of the inherent intelligence and standardization of the PRC units.
3. Flexible operation that adapts to changes in the communications link during or after installation and startup.
4. Standardized physical and logical design among various manufacturers simplifies equipment specification.
5. Improved radio system performance in shared systems due to collision avoidance built into the PRC control software.
6. Single unit for radio, microwave, and direct connect simplifies program and equipment changes for different systems.
7. Multi-unit relay capability to extend range by allowing a PRC to act as a repeater for a distant unit.
8. Low cost per unit due to software standardization and simplicity of the electronic design.
9. Simplified system troubleshooting using the built-in capabilities of the controller.
10. Potential application of controller as a minimal RTU system with few additional components.

PACKET CONTROLLER HISTORY

Although the PRC is fairly well known in certain radio fields, its use as a general purpose data device is still in the early stages of development. Its original development came in the data communications field as a means of standardizing the exchange of computer data on a shared network. Many systems must deal with numerous users, some of whom rarely use the link. Examples include microwave systems, satellite systems, and general purpose radio systems. Packet technology is aimed at making optimum use of this type of communications arrangement, and Packet Radio is a subset of the standard related to audio type radio systems. This worldwide system was formalized in 1976 by the International Consultative Committee on Telephone and Telegraph (CCITT) as specification X.25. This specification is officially titled "Interface Between Data Terminal Equipment (DTE) and Terminals Operating in the Packet Mode on Public Data Networks." This document was amended in 1980, and is also known as

Bell Specification BX.25 for use in Bell Operating Company systems.

Much of the current interest in Packet Technology has been generated by the Amateur Radio field. As was the case in early personal computers, these "hobbyists" have improved many of the practical aspects that make the technology useful in everyday applications. The Amateur world has developed its own specification based on X.25 called AX.25, which extends the original specification and has features that are valuable in SCADA applications. Specifically, the sender and receiver address schemes have been enhanced to allow radio call sign or station identification codes to be used. The normal X.25 specification uses a less flexible commercial numbering system. The amateur interest has also provided industrial users with low cost, reliable equipment. The packet controllers are often referred to as Terminal Node Controllers, or TNCs, and are designed to comply with the TAPR standards. This is a reference to the Tucson Amateur Packet Radio Corp, which fostered many of the improvements that caused the development of the amateur specification.

In this discussion, no separation between X.25 and AX.25 is made unless an important difference exists.

PACKET CONTROLLER BASICS

The explanation of Packet Technology frequently requires the use of analogies to other communications methods. For example, the standard telephone modem is a well understood concept that has many similarities to packet controller applications. This discussion will follow this path of comparing the new Packet Technology with the more common phone modem device.

A gross generalization of a PRC is to state that it does for data radios what the Hayes Modem has done for telephone-based data communications. The Hayes modem, developed by private industry in the 1970s, is now the worldwide standard for the design and application of dial-up telephone communications. The Hayes modem provides standardized features to both control the data link as well as exchange data while the link is intact. The modem exchanges data with the computer via an RS-232 serial link, and connects to the phone line with an RJ-11 modular jack. Control by the attached computer is done with the "AT" command set, and data is sent using a limited number of Bell and CCITT tone standards.

Similarly, the Packet Radio Controller provides a standard method of control and data exchange over an audio communications channel, which is normally a standard voice radio. In fact, the PRC usually interfaces to a radio's mike and speaker connections. While the PRC is primarily designed for radio applications, other audio based communications links can benefit from it as well. Both device and link control are done with a very simple, text based command set that has been well standardized (and somewhat extended). Data is

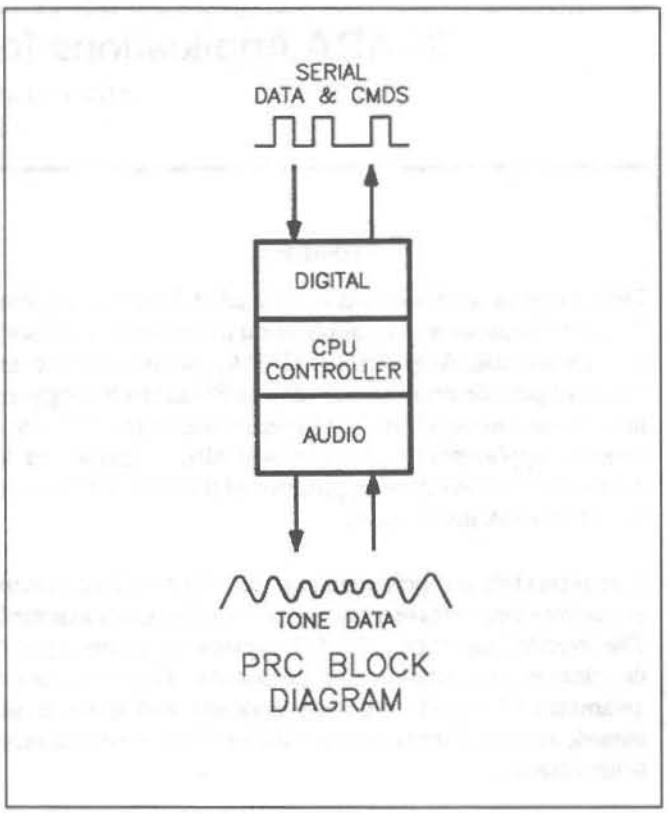


Figure 1

exchanged with a wide range of tones using Bell and CCITT frequencies that can be adjusted to best suit the capability of the radio being used. (Figure 1)

The PRC acts as a physical and logical buffer between each data device and the link. It has the capability to "connect" to another PRC and to indicate to the data device the status of the link. When connected, the two PRCs can exchange data in a manner that is transparent to the data devices. The computers simply send data out and receive data in as if they were directly connected to each other by a simple serial cable. The only deviation from the direct connection is the time delays involved in sending the data, and these delays can be considerable (seconds or minutes) in extreme situations.

Although there are many similarities between the Hayes modem and the PRC, there are many differences as well. Hayes modems are designed for direct dial telephone connections. When in use, the Hayes modem has the exclusive use of the phone line until the call is terminated. The Hayes modem also sends data continuously when connected, with virtually no delay between transmission of data from the sending computer and injection of tones by the modem into the phone line. The modem in this case merely serves to extend the computer's data ports over the phone line. Whatever goes in one end comes out of the other end without any actual processing by the modem itself other than the basic tone generation and detection. (OK, OK, I know that some

modems actually manipulate data for error detection and correction, but let's ignore that for now).

Unlike the Hayes modem, the PRC is designed to use a communication link that is simultaneously shared with other users (though not at every instant). The other users do not have to be any type of compatible equipment, but can be anything that makes noise on the line (including simple voice). The PRC does this by sending data in bursts, called packets, rather than on a continuous basis. The PRC essentially acts as a network controller between connected devices. Normal PRC use is restricted to two devices during any logical connection, although multiple-connect types are allowed by the standards.

Every packet received by a PRC is acknowledged by a short burst back to the sending unit. With this scheme, every single packet that is sent is accounted for by the sender. When a transmission occurs, one of three things can occur.

1. The receiver has no errors and ACKs the packet.
2. The receiver detects errors and NAKs the packet, causing a re-send.
3. The receiver never hears the packet, and the sender tries again.

The packet network design is very flexible because devices can connect and disconnect at any time. No device has to know about another until it needs to connect with it. Also, many simultaneous connections can take place during the same time period because the PRC operates in burst mode and does not "hog" the communications link as is the case in a phone modem and most network systems.

PACKET RADIO PHYSICAL CONNECTION

The data side of the PRC is connected to the computer or other data device through a standard RS-232 serial connection. Both the traditional DB-25 and newer DB-9 "AT" type connectors can be found on various units. While other standards such as RS-422 are possible, the author is not aware of any available units using this method. The computer data enters and exits the PRC as asynchronous, serial characters that are compatible with almost any computer related device. No special codes or protocols are required for the computer side of the connection. (Figure 2)

On the audio side, most PRCs are designed to connect to standard voice type radio systems. They produce a low level audio output, and receive audio input of almost any reasonable level. This allows them to be connected to the mike input and speaker output of the radio, although other internal connections are possible. Also, the PRC must tie into the push-to-talk (PTT) line of the radio, which normally involves shorting the line to ground to key the radio. An optional squelch input to the PRC is normally available that allows a channel "busy" signal to be monitored. When busy, the PRC will not transmit. The PRC also looks at audio noise on the line to avoid collisions, so the squelch input is optional. (Figure 3)

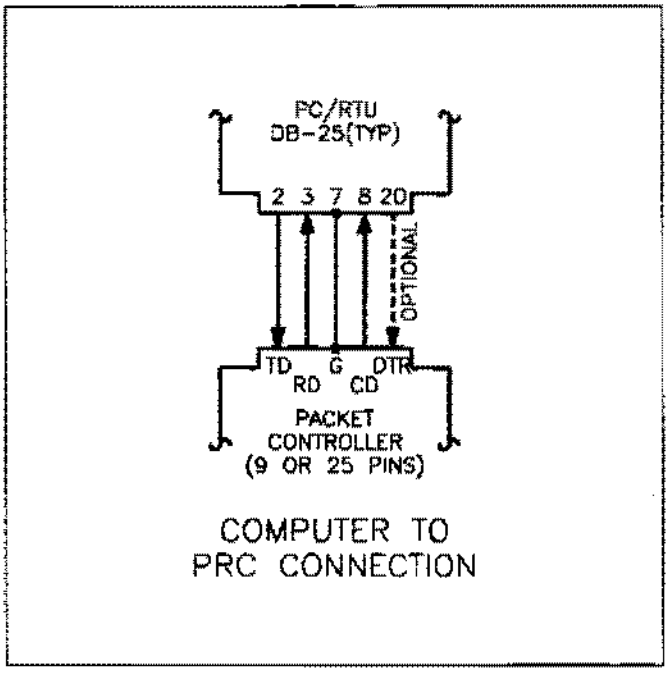


Figure 2

Special PRC circuit cards are available that plug directly into the chassis of the common IBM compatible Personal Computer. These devices rely on the PC for much of the processing, and are not as desirable for SCADA applications where

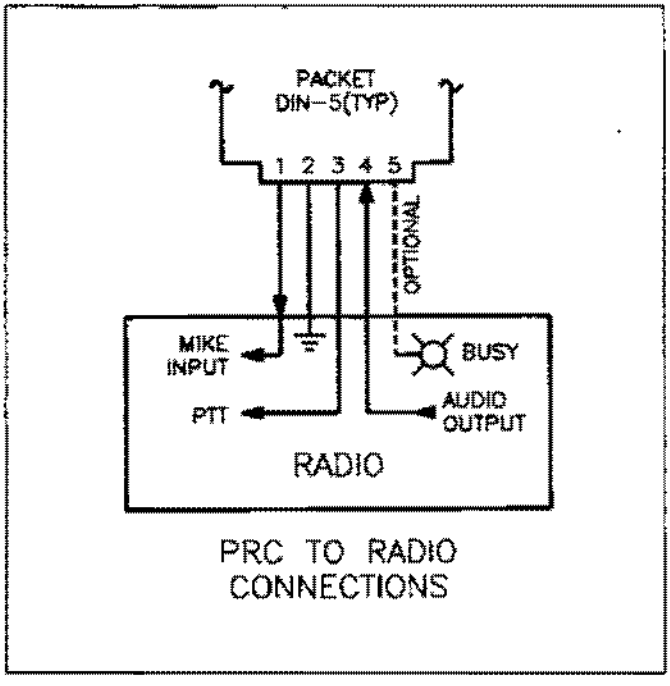


Figure 3

the PC is likely to be very busy already. The self contained PRC, with its own dedicated processor, is much more desirable as a general purpose device.

PRC TRANSMISSION RATES

Computers send serial data at a predetermined rate called the "baud rate," which in the broad sense means "bits per second." This historic term actually relates to the tone changes per second allowed on the communications link, and is generally directly related to the actual data bit rate being transmitted. Modern modems use advanced techniques to pack more bits per baud, resulting in a higher bit rate than the actual tone change rate. For purposes of this discussion, we will avoid the argument over terminology and simply assume that the baud rate is the same as the bit rate.

In the standard phone modem, bits come in from the computer over the RS-232 serial line as electrical 1s or 0s and are instantly changed into a specific tone on the phone line. The receiving modem "hears" this tone, and puts out a corresponding 1 or 0 based on the tone being received. Therefore, there is an almost instantaneous response at the receiving end, and the data bit is intact all along the route as either a 1 or 0. The modem system simply extends the normal electrical connection between the two computer devices.

The PRC is considerably more complex than the phone modem. The computer sends data out of the RS-232 port in the same manner, but the data is accumulated in the PRC until a later transmission time. As the packet is assembled inside the PRC, the sending computer is unaware that there is a hold-up in the transmission. The computer and PRC communicate at a fixed baud rate, called the terminal baud rate or "TBAUD" in PRC terminology.

Data is collected into these larger groups (the actual "packets") which contain the sender's data as well as information regarding routing and error detection. Packets are sent as a complete unit, bit by bit, using the industry standard "HDLC" synchronous bit oriented protocol. This protocol is internal to the PRC and need not be of concern to most users. The transmission time of the packet is fairly short (1-2 seconds), allowing many devices to send bursts on a time shared basis.

When the packet is transmitted, the PRC sends it in a short burst to the receiving unit. The tones, baud and other details of this PRC-to-PRC transmission have absolutely nothing to do with the settings of the terminal-to-PRC connection. The PRC-to-PRC settings are called the "HBAUD" configuration of the units, and they must match exactly on both ends of the communications link. (Figure 4)

So, the computer talks to the PRC with one data rate setup, while the PRCs communicate among themselves with quite another system. While the two PRCs must have a common setup, the computers at each end of the link do not. The only

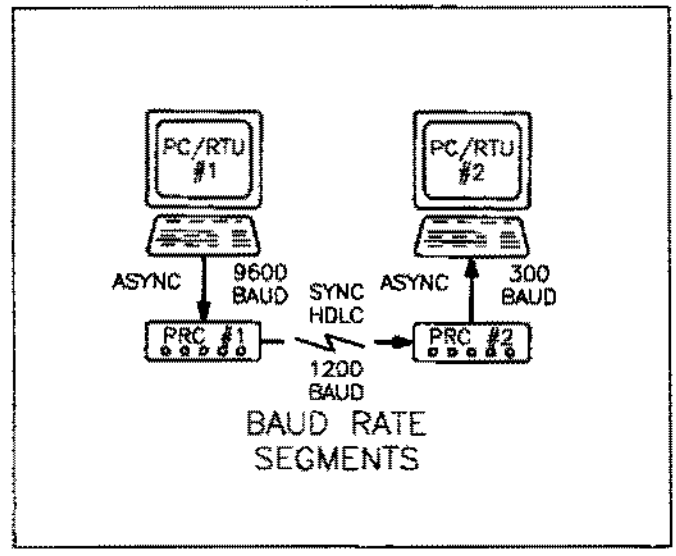


Figure 4

requirement is that each PRC be properly configured to communicate with the attached data device. The setting of the data device at the other end is not important. Thus, it is possible to have the sending computer transmit data into its PRC at 9600 baud, have the data transmitted over the radio at 1200 baud, and then sent from the receiving PRC to its data device at 300 baud. This would be impossible in the Hayes modem setup because there all devices must be at the identical baud rate.

Why would different baud rates be desired in the first place? Well, normally they are not, but the delays associated with the packet assembly and transmission can be reduced by using the highest possible baud rates at each segment. If the radio system can only support 1200 baud, then using 1200 baud throughout results in an effective rate of no more than 400 baud for each packet transmission. The computer-to-PRC at 1200, followed by a PRC-to-PRC transmission at 1200, and then a PRC-to-computer transmission at 1200 takes 3 times longer than a single 1200 baud burst on a phone modem. Add in the key up delays and acknowledgment cycle, and the effective baud rate drops even further. So, the use of a higher baud rate for the PRC-to-computer segment greatly improves the overall performance, and is usually within the capability of the connected equipment.

PACKET CONTROLLER FEATURES

Many features of Packet Technology have been standardized by the AX.25 specification. These features relate to the command set and overall operation of the controller. In addition, many units offer non-standard or semi-standard features to accommodate industry specific requirements. A sampling of the more important features is as follows:

STANDARD FEATURES

1. **Radio Key-Up Delay**

   Most radio systems require some time for the transmitter to power up, and for the receiver to un-squelch. The PRC will automatically assert the Push-To-Talk (PTT) line on the radio and wait before actually sending any data. This time is adjustable for each end of the link in 10 ms increments.

2. **Transparent or Conversational Mode**

   The PRC will assemble a packet and transmit it as a unit. In transparent mode, the transmission takes place after a user settable delay from the time the last byte was received. In Conversational mode, the PRC will only transmit when an end of line character is received. SCADA applications will normally use transparent mode, while manual operation would use conversational mode.

3. **Delay Before Packet Transmission**

   When in transparent mode, this setting determines the number of 10 ms periods to wait before transmission. This allows a slow sending device to have its packet assembled completely and sent as one unit.

4. **Packet Length**

   Packet lengths up to 256 characters are allowed, with the optimum length being determined by the quality of the audio link. Better links can use longer packets due to fewer retries. The actual message blocks can be longer than the packet length, although several packet transmissions may be required to send the user's data segment.

5. **Number of Outstanding Packets**

   The controllers can send multiple packets without waiting for a receipt acknowledgement from the other unit. Because each packet is numbered, the receiver can reassemble the packets in the proper order when they are all received. The max number of outstanding packets (i.e. those not ack'd) can be set to avoid having one end get to far ahead of the other end.

6. **Automatic Station ID (Beacon)**

   Radio licenses requiring periodic station identification can use the PRC to send out a text message at user specified intervals.

7. **Transmit Retry Setting**

   The controller will sense transmit errors and resend packets a user specified number of times.

8. **Radio Hardware Protection Timeouts**

   The controller will only key the radio for a specific time period. This prevents transmitter burnouts due to a stuck Push-To-Talk line previously caused by a crashed computer directly controlling the radio.

OPTIONAL FEATURES

Although the standard specification provides many useful features, some manufacturers offer unique or semi-standard additional features that are helpful in certain applications. Some of the more common ones are listed here:

1. **Packet Repeater Operation**

   Permits packet data to be sent down a series of units to allow connection of units normally out of radio range.

2. **Radio Tone Calibrate Mode**

   Manual activation of the radio link tones allows simplified calibration and verification of the radio system during installation and maintenance.

3. **HAYES Modem Command Emulation**

   A simplified PRC command set based on the Hayes "AT" command set allows use of the PRC with systems actually designed for phone modem use.

4. **Data Encryption**

   A built-in text encryption is available in non-amateur units that will scramble the data block section of the packet. This can be useful if sensitive data is being exchanged, or access to the equipment by an uninvited user is anticipated.

5. **Remote Configuration**

   Many units can be configured from the opposite end of the communications link. This allows for parameter setting to be done without actually connecting to the PRC itself. All that is required is that a remote connection be established over the radio link.

6. **Built-in I/O Connections**

   Some units offer special digital I/O points designed primarily for printer interfaces. However, the potential is there to use these points as a minimal RTU by adding the proper interface devices to the PRC. With the proper Host software, these points could be monitored or controlled to provide a very simple, low cost RTU with considerable communications capability.

TYPICAL PACKET SESSIONS

A typical packet session consists of three separate phases:

1. Initiating and verifying the connection.
2. Data exchanges.
3. Link termination.

Phases 1 and 3 require that the data devices at one or both ends operate the PRC in "command mode," where data sent from the computer is processed locally by the PRC. During Phase 2, data sent from the computers is transmitted and received in a manner similar to normal modem connections.

It is important to understand that the PRC is always in one of two modes: Command or Transfer. When in command mode, the computer connected to the PRC manipulates it directly. When in transfer mode, the PRC acts transparently and exchanges the data with the other PRC. Various methods are available to toggle the PRC between the two modes.

A typical session would start with a computer issuing a connect command to its PRC. Like all PRC commands, a simple plain text message is sent to the unit indicating the name (station ID) of the PRC with which to connect. The PRC takes over at that point, and transmits a request to connect. If a PRC with the proper ID is somewhere on the audio link, it will respond and the connection will be made. At that point, both PRCs have shifted from command to transfer mode. Note that the connection between PRCs is done with short bursts, and the communications link is available for other users during the idle times of the connect period.

While connected, each end of the link can freely send data of any form, including binary data. The PRC does not get involved in the format of the data. It only acts as the carrier of the data. Therefore, any existing data protocol can be sent over the PRC once the connection is made. The two data devices do not have to provide any additional controls or error detection because all of the busywork of maintaining the link is done by the controllers.

GENERAL SCADA APPLICATIONS

Packet technology is best used to allow a common communications system to be shared by many users. The general theme is that although many users exist, none of them use the link a large percentage of the time. In many instances, the link may even be idle for long intervals. Many SCADA applications fall into this category, where periodic updates from an RTU to a Host system take place in brief data transfers that occur at scheduled or random times.

Many remote SCADA systems rely on a voice type communications system for data transmission. When the link will be by radio, the PRC can be used to greatly simplify and enhance the standard two way radio system. It can also be used on other multi-drop links such as microwave channels or simple wire pairs. In all cases, the PRC will provide a means of exchanging data with minimal impact on the programming and other functions of the connected devices.

The basic SCADA application will be to take an existing telemetry or control system, implement some form of simple PRC control scheme, and connect the system to the radios. Once the packet controllers are in place, the data units can exchange information as if they were directly connected. The normal data protocol native to the system can remain largely unchanged, with the only possible trouble area being message timing.

With this setup, systems that normally rely on direct connection (such as Programmable Logic Controllers) can act as remote terminal units over a complicated shared radio network. Without the PRC, extensive modifications to the software in the data devices would be required to implement the numerous functions handled internally by the PRC.

SCADA MISAPPLICATIONS

High speed SCADA systems, or those requiring large data transfers, may not be suitable for Packet Controllers. The instantaneous throughput of packet is somewhat slower than a direct modem approach. This is due to the inherent delays associated with the assembly, transmission, and acknowledgment of the packets.

Radio systems with restricted data licenses may also have problems implementing packet. The typical restriction is the time limit (seconds per minute) that the SCADA system can use the channel for "secondary signaling." This usually refers to some sort of tone signaling device, although SCADA often comes under this umbrella. However, the increasing acceptance of packet technology by the FCC will relax these restrictions because it is recognized that packet makes better overall use of the available channels.

LINK CONTROL SOFTWARE MODIFICATIONS

The implementation of Packet Controllers in SCADA will almost always involve software modification to the data device that controls it. However, the modifications required are far simpler than would be required if the software had to perform all of the functions handled by the PRC itself. The changes (or enhancements) required involve the sending of commands to the PRC only. In many cases, no modifications to the normal data transfer procedures will be required. If the software is already capable of operating a telephone line modem, then changes to accommodate radio communications via PRC are almost trivial.

The modifications can be kept to a minimum by only providing the software with the ability to make and break the connection. This would require only two commands, CONNECT and DISCONNECT. Although commands would be required to configure the PRC initially, they can be done "offline" with a personal computer or simple ASCII terminal. Many settings are maintained by the PRC even after power is removed, so a one-time setup may be all that is required. Once configured, the PRC need only be commanded to connect and then optionally to disconnect when the transfer is complete.

A more complete command capability would give the computer the ability to send a variety of messages to control and configure the PRC. With this capability, the computer could dynamically change the action of the PRC to suit different links and different operational conditions.

One important software feature that must be added is the ability to switch the PRC to and from command mode. When in command mode, the PRC responds to messages rather than transmitting them. If the PRC is in transfer mode, and the computer wants to tell it to disconnect, then the PRC must first be switched to command mode. Common methods of doing this involve a special code sequence in a specific period of time. Many units use a sequence of three control-C characters that are sent rapidly within one second after a one second wait. This may sound complicated, but it is well within the capability of even the most rudimentary modern RTU.

All PRC commands are done with simple text phrases followed by one or more optional parameters. Examples are CONNECT, TXDELAY, RETRY and TBAUD. The commands can be shortened to less than the full length as long as all letters are correct, such that the command DISCO is the same as DISCONNECT. The PRC command set is quite extensive and fairly standardized, so software prepared for one manufacturer's unit can be used with others that follow the AX.25 standard.

COMMUNICATIONS PROTOCOL CONSIDERATIONS

The main consideration of the sending and receiving software is that time delays will be present when data is exchanged. The protocol must operate in a "relaxed" mode, where some time is allowed for a response from the other unit. In a clean communications system, the response time will be a few seconds at most. However, in a noisy system, several retrys may be required by the PRC that extend the delay to 10, 20 or more seconds. Radio repeater applications (discussed later) can further extend the delays to over a minute. The data system is unaware of the retries or other communications problems being handled by the PRC, and simply waits for the reply if one is expected.

While this may sound quite limiting, it is not normally a problem in many modern SCADA systems. Typical applications involve a Host system that periodically "polls," or interrogates one or more Remote Terminal Units (RTUs) to determine the status of the equipment at that location. Additionally, the RTU may constantly accumulate information that is sent to the host during the brief poll session. Even on a large system, the quantity of data involved is fairly small. Therefore, the transmission time of several seconds rather than a fraction of a second is not a factor in the overall operation.

Some simple SCADA systems operate in simplex mode where they only transmit data to a host computer. The RTU end of the link transmits information and does not process a reply from the host end at all. These systems will realize little impact from the PRC delay because each phase of the transmission will not have to wait for a reply from the other end. There will be some delay in delivering the message to the host, but the messages can come in at a steady rate. The advantage is that the simple RTU will not have to get involved in all the work required to manage the communications line at all.

However, more advanced SCADA systems requiring duplex communications will require some alteration to allow for the slower pacing of the message transfers. If the sending end must wait for a reply from the receiver before proceeding to the next step, then the time involved to receive the response must be considered. The sophistication of the user's protocol will determine the impact of these delays, and advanced techniques such as the "Kermit sliding windows" protocol will minimize the effect on overall throughput. This protocol, developed at Columbia University and used in many file based systems, was specifically designed to accommodate packet type transmissions. Other equipment, such as Programmable Logic Controllers (PLCs), may also support a numbered message system where overlapping blocks can be handled to allow implementation of a packet system.

If the user's system has a built-in retry mechanism, then this scheme must be coordinated with the one in the PRC. One way to do this is to set the PRC retry count at 0, so that it will not attempt a retransmission of a failed burst. However, this does not take advantage of the unit's capability. It is better to let the PRC do the retrys, and set the sender's software to limit retry attempts.

If the sender's software has a retry scheme based on a reply timeout, then it is possible that the receiver will get the same message more than once. This is because the PRC may have had difficulty getting the message through, and the sender sent it more than once because it did not receive a reply in time. The multiple messages will be delivered in sequence, and the receiver may have to be aware of this if multiple processing of the same message would pose a problem. The message numbering system inherent in many SCADA and PLC protocols should eliminate any problems with retry overlaps, but it should always be considered when designing a system.

CASE STUDIES

The following examples of PRC application are based on actual field installations done by the author. Minor changes to the actual requirements have been made in these examples to simplify the descriptions, but the detailed technical components are accurately represented. The intent is to present actual experiences without making any commercial representations of any kind in this document.

Case Study #1 — Simple RTU

The first PRC installation done by the author occurred in 1988 for a single RTU to HOST computer link. The installation was to be on an offshore production facility reporting back to a nearby manned facility which was also located

offshore. The RTU had been developed by the author as a new product based on Personal Computer (PC) technology. This unit was the first in what has since become a standard product. Hardware and software development was complete except for the communications control software. This had been delayed until the end of development as it was assumed that the control functions would depend on the actual communications equipment provided for each location. (Figure 5)

When it was decided that the RTU would communicate over the standard voice radio used by the facility operators, the many details required to perform this task became evident. This was compounded by the requirement to operate with radios in the 50 Mhz range, whose audio quality is normally considered too poor to support reliable data transmission. The various timing, error detection, routing, and FCC requirements presented a sizable software development effort. The task was made even less desirable when it was realized that the efforts would be mostly unique to the installation at hand, and future installations would surely require considerable alterations.

Assuming that the RTU software could be developed in the time required, installation could not be completed until suitable low power modems capable of connecting to the standard voice radios could to be located and tested. The search for the simple modem accidentally lead to the PRC, which was initially selected to provide an interim solution until the software could be further developed.

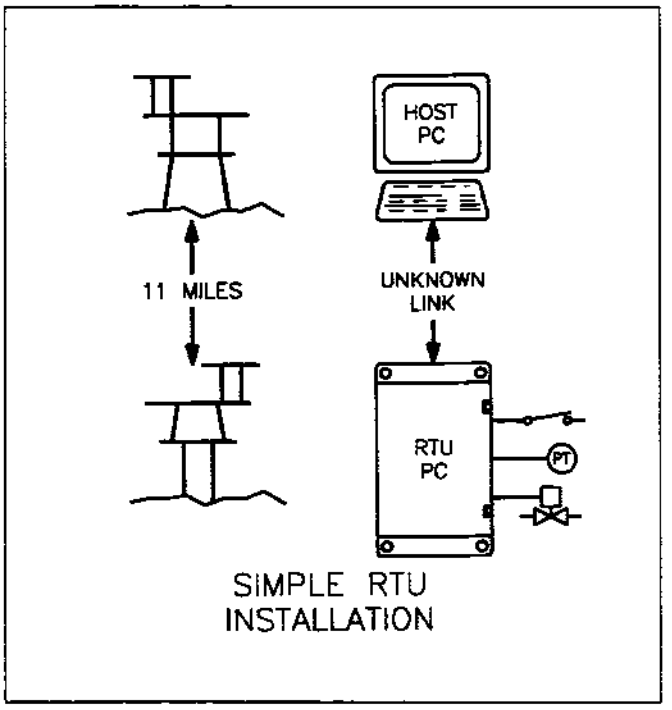


Figure 5

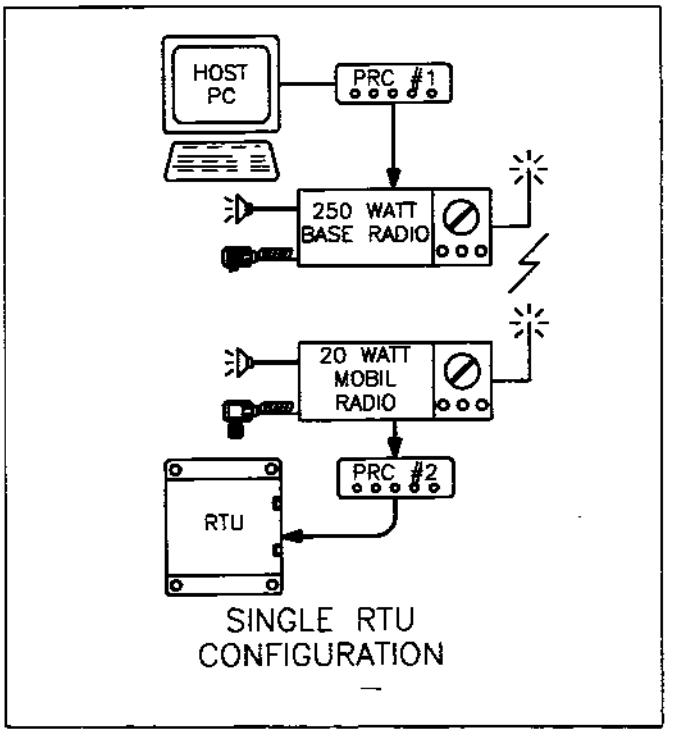


Figure 6

However, the PRC worked so well that thoughts of emulating its performance within the RTU software were quickly abandoned. The only RTU software modification required was the addition of a single command process that sends messages to the PRC. With this single addition, all of the functions of the PRC became available to the main RTU software. The performance of the PRC, both as a routine data link and as a diagnostic tool, far exceeded the initial expectations. When it was realized that the PRC was itself a fairly standard product available from numerous sources, its permanent addition to the PC based RTU became obvious. The PRC has since become an integral component in all radio and microwave based systems designed or installed by the author.

The final configuration consisted of a Packet Radio Controller connected to RS-232 serial ports on the RTU and HOST computer. The PRCs were connected to the mike input, speaker output, and Push-to-talk (PTT) line of the voice radio. Once properly setup, the PRC took care of all overhead associated with the transmission of data over the radio system. This not only included the normal functions of data transmission, but also included the details of sharing the busy radio channel with frequent voice traffic. (Figure 6)

A continuous poll of the RTU was not possible in this system due to electrical power considerations at the RTU as well as the normal voice use of the radio channel. In this system, the RTU was set up to initiate a call to the Host whenever an abnormal condition was detected. This was expected to occur only three or four times per day. The PRC took care of all

communications control functions, including not "stepping" on a voice transmission in progress. After the channel is clear, the PRC exchanges the data packet with the other unit and receives an ACK indicating that the block was received properly. If a voice transmission occurs while the data transmission is in progress, the PRC will detect the error and retry as required. All this takes place without any effort on the part of the computer devices.

Case Study #2 — Complex RTU System

A later installation of similar equipment required that four RTUs connect to a single Host computer. In addition, it was desired to have one of the RTU locations access information from the other units such that it acted as a sub-host. The PRC provided the perfect solution without the need for any additional hardware.

This system was set up on its own radio channel in the 150Mhz range, and interference from nearby traffic was not anticipated. In normal use, the Host unit connects to each RTU in a sequence initiated automatically every 30 minutes or on demand by the operator. The only limitation here was the battery power available at the RTU locations, which were solar powered. Additionally, the RTUs were programmed to initiate a call to the Host on an abnormal condition. With this scheme, the battery power was conserved while still providing the operator with timely alarm information. (Figure 7)

The ability of the PRC to connect to any other PRC on the communications channel allowed the secondary Host to be easily implemented at one of the RTU locations. On demand by an operator keypress, the secondary Host would connect to the other RTU locations and download data in a manner similar to the normal host. This allowed operators located at this RTU to see data for all of the other units without having to travel back to the primary host. All this was accomplished using off-the-shelf PRC units, including the eventual use of units from two different manufacturers.

Case Study #3 — Long Distance Application

The ability of the PRC to act as a relay station was put to use in an application involving a host and two RTUs, one of which was 65 miles from the Host location. Fortunately, the physical layout was such that one of the RTUs was between the farthest one and the host location. Communications were adequate between the close RTU and the Host, and between the two RTUs themselves. However, the Host could not connect to the farther RTU.

The solution was to have the Host connect to the farther RTU via the closer one. All that is required to do this is the inclusion of the routing information in the station ID. This function, call "Digipeating," is standard on many of the commercially available PRC devices. Instead of simply

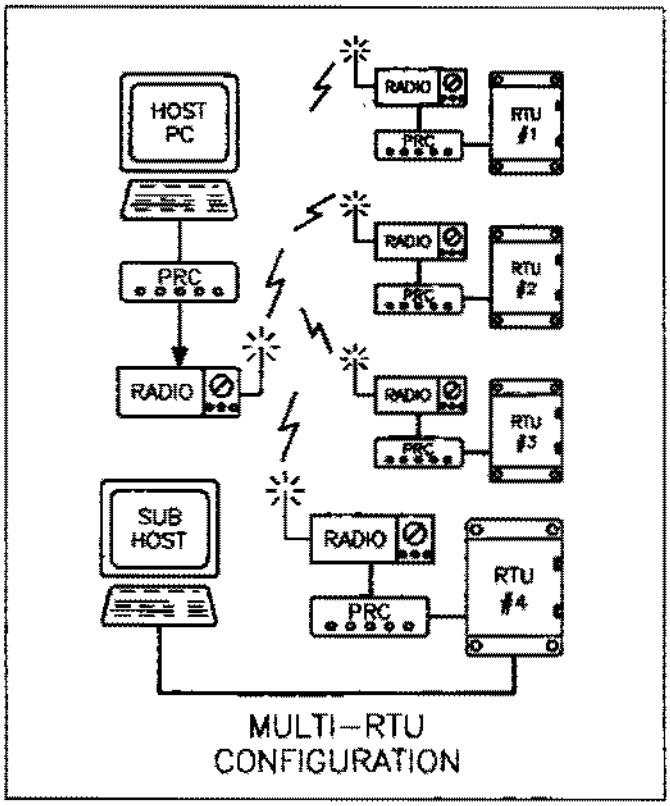


Figure 7

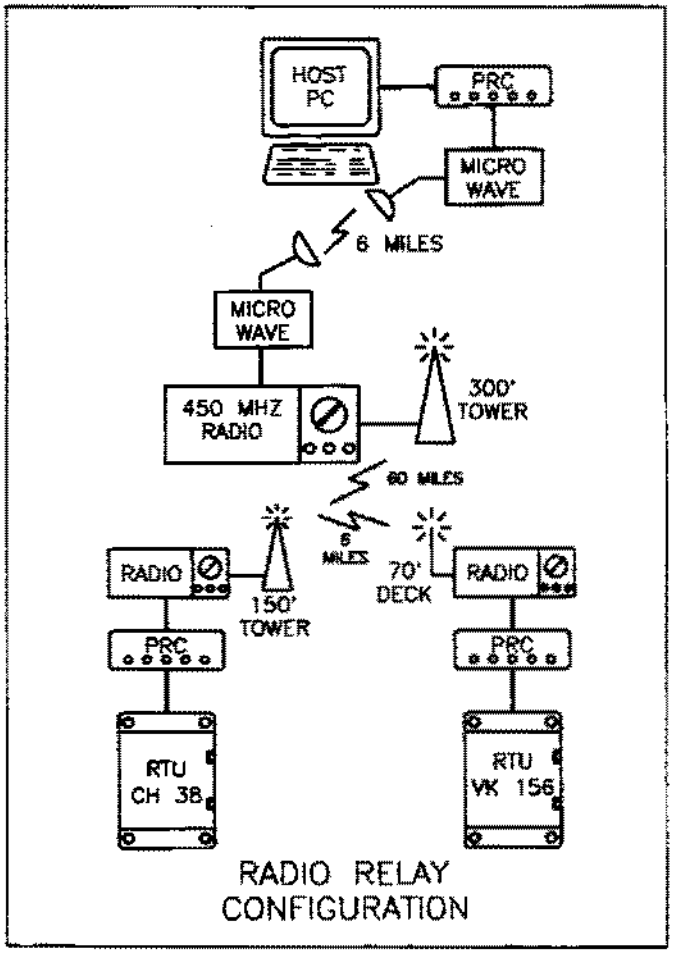


Figure 8

specifying the ID of the end unit, a "path" is provided that mentions the name of the intermediate PRC. No additional programming of the RTU or PRC is necessary. (Figure 8)

The two RTU ID tags are in this system are CH38 and VK156, which are the names of the offshore platforms where the RTUs are located. Normally, the host would send the command "CONNECT CH38" or "CONNECT VK156" to initiate the connection directly to the desired RTU. Because the host could not connect properly with VK156, the relay system was implemented through the use of the command "CONNECT VK156 VIA CH38." In this case, messages are routed from the host all the way to VK156 by passing through the PRC at CH38. The computer equipment at CH38 is not involved in this transaction at all. The PRC at CH38 receives the packets from the host, and retransmits them automatically to VK156. The reverse occurs when VK156 sends a response back to the Host.

The only drawback to this situation is the additional time delays required to perform the relay. The time required to send and receive packets is essentially doubled over the non-relay case. However, this scheme eliminates the requirement for a repeater radio system that would have been considerably more expensive to install and maintain. It may also be possible to implement the digipeating scheme only during periods of poor communications, and use the normal direct connect at other times.

Note that several relays can be in place so that a number of PRCs can be involved in the process. The practical limitations of this are related to the extended time required for the packets to pass through each of the units in the chain.

Case Study # 4 — Low Power RTU Application

A SCADA system was installed that was designed to operate at a very low electrical power level. The system consisted of two RTUs and a single Host, with the addition of other RTUs anticipated in the near future. The design is based on a RTU that powers down when not in use, removing all of the power required for the processor as well as the signal transmitters and end devices.

A design based on standard components was selected as the RTU. One key requirement was that the RTU be signaled when it must power up and transmit data. This should occur when the RTU is being contacted by the Host system. Because the PRC has a connect signal available at the normal Carrier Detect position on the RS-232 connector, it was fairly simple to interface the PRC to the power-up circuitry of the RTU. (Figure 9)

In normal operation, the radio receiver and the PRC are the only devices that are continuously powered. The units used had a combined standby power requirement of 40 ma at 12 VDC, making battery and solar powered operation quite feasible. The PRC awaits a connection, and asserts its CD line

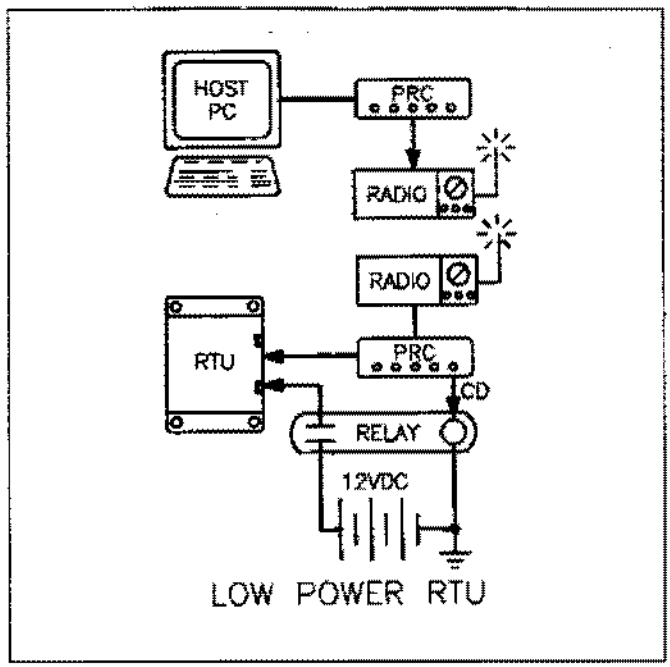


Figure 9

when the Host issues a connect request. The PRC thus acts as the unit controller as well as the communications controller.

After the data transfer is complete, the RTU removes power from itself and effectively goes to sleep. The PRC remains powered to wait for the next session. In this particular case, the RTU is not even aware that it is connected to the PRC at all. It simply sees commands come in from the serial line, and sends data out the same way. All of the details of PRC operation are preprogrammed into the unit, and the Host location controls all sessions.

THE PRC AS A DIAGNOSTIC TOOL

One powerful feature of the PRC is its ability to assist in the initial setup and eventual troubleshooting of the communications link. One of the most common problems in the operation of any SCADA system is isolating communications problems, and the PRC can often be used as a diagnostic tool for this purpose. Because the PRCs use the link independently of the data devices, the viability of the link can be tested and verified without assistance from the computers themselves. In other words, the quality of the communications channel can be verified without the actual use of the telemetry equipment, and this can reduce "finger pointing" during discussions over which party is responsible for repairing the system.

If a PRC can connect to another PRC, then the communications channel is good. The PRC sends and receives its handshake data in the same manner that will be used to send the user's data, so a simple connect test is all that is needed to verify the communications. However, if the PRCs fail to link properly, then no amount of adjustment in the computer side

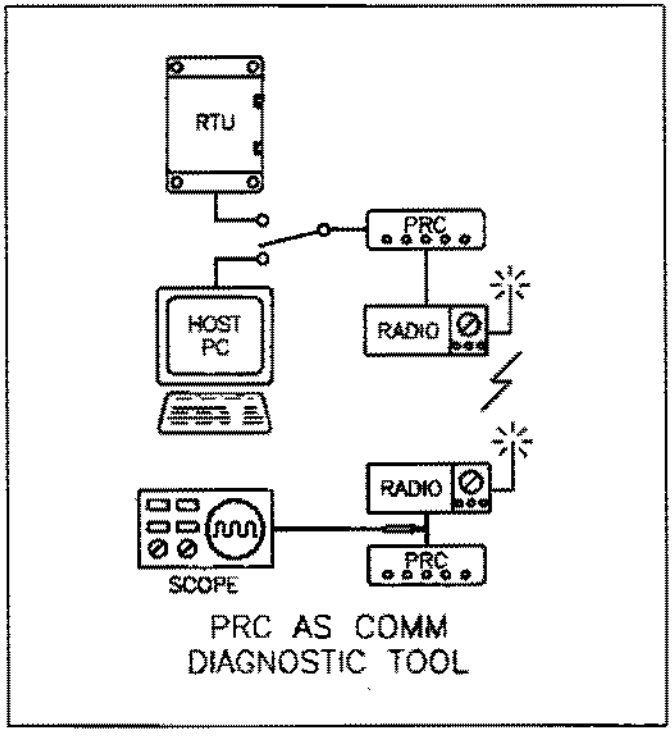


Figure 10

of the system will fix the problem. In this manner, the PRC acts as a great arbitrator in the battle between the computer people and the communications people. (Figure 10)

The PRC can usually be manually operated (via simple commands) to send test tones into the communications channel. These tones are needed for radio calibration, and it greatly simplifies examination of the tones at various points in the link. Without this simple capability, additional equipment would be needed to inject tones into the system. The PRC provides this function at no additional cost.

ALTERNATE APPLICATIONS

Although the PRC is designed for radio system use, the basic technology lends itself well to any shared communications medium. One obvious use is to apply the PRC to a microwave audio circuit that has numerous connection points. A PRC at each point serves to connect the devices connected to the channel in a manner very similar to that of the radio. However, in the microwave case the PRC will likely be connected to a 4 wire low impedance audio terminal rather than to the speaker and mike of the voice radio. Also, the push-to-talk function will not be required. If the microwave is full-duplex, the PRC can be easily configured to take advantage of this to send and receive at the same time.

Another application is in units operating in a direct connect mode. For short distances and small unit counts, the PRCs can be directly connected together with the transmit and receive leads all shorted together. In this case, the common

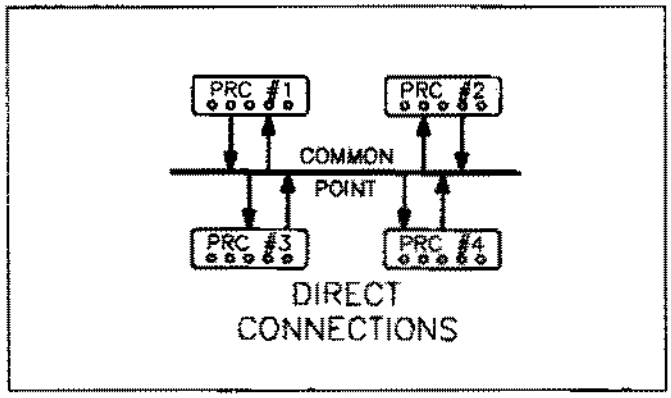


Figure 11

connection point forms a network, although at a very simple level. This mode can also be used to test a system without the actual use of radio equipment prior to installation. This simulation allows for all parameters to be tested except for the actual radio operation. (Figure 11)

Longer distances or higher unit counts may require the use of amplifiers or interposing relays to reduce the load on each PRC's transmit circuit. With this scheme, each PRC transmits onto the common connection and receives input directly from the connection.

SUGGESTED PRC IMPROVEMENTS

While the units currently available on the market provide acceptable service, there are several areas in which improvement can be made to increase acceptance by the industrial market. These items are suggestions by the author based on personal experience, and are somewhat subjective in nature. They are listed in no particular order.

1. Improve reliability of the unit with the installation of a failsafe or watchdog timer in the PRC itself. With this feature, the inevitable lockups common to this type of device can be corrected automatically without the use of external reset devices.
2. Reduce electrical power consumption through the use of CMOS devices and more efficient power regulators wherever possible. The SCADA market demands low power in many instances, and many of the available units would be perfectly acceptable if their power consumption were drastically reduced.
3. Designs should provide full diagnostic LEDs or other indicators. The PRC acts as an excellent diagnostic tool, and the use of indicators allows untrained operators to report the status more easily during telephone troubleshooting sessions.
4. Use hardware settings for the most basic setups such as terminal baud rate. This enhances reliability in unattended installations where an inadvertent change in the terminal baud rate will effectively kill the unit. The use

of autobaud type settings should be avoided or made strictly optional.

5. Provide the ability for the user to easily store common settings in a permanent ROM. Reliance on battery backed RAMS or even NOVRAMS is not acceptable to many industries, and many current PRC designs use this technique.
6. Provide a simple hardware reset feature accessible from the RS-232 line. The author has made simple modifications to several commercially available units that cause a hardware reset when the RS-232 DTR line (pin 20) is toggled. This is similar to the technique used by many Hayes modems as a hardware "hangup," and is desirable because it is easy to implement and does not depend on timing or band rates.

FUTURE APPLICATIONS

The most likely future applications of PRC technology will come from the inclusion of PRC control capability in currently available equipment. Just as communications programs universally support the Hayes command set for phone modems, future programs may incorporate PRC command structures. The standardized nature of the A.25 specification and the features of currently available PRC units makes this a very realistic concept.

It will be necessary for various industry groups to sponsor this development. The actual details of the PRC AX.25 specification have already been developed. All that remains is the adoption of application specific details unique to each industry so that a common design base can be established. From that point, PRC unit can be applied as a generic component without concern of the unit specific features in most installations.

CLOSING

It should be apparent that the author is very enthusiastic about the use of Packet technology in everyday remote SCADA applications. Field experience has proven that the devices are invaluable in the rapid application of off-the-shelf components configured as a modern SCADA system. Continued development is likely as industry becomes aware of the benefits that can be gained, and the expected improvement in available equipment will further advance the technology.

Packet Radio References

The following references offer excellent introductory and advanced information on Packet Radio Technology.

AEA PK-90 Manual
Advanced Electronics Applications (AEA), Inc.
P.O. Box C2160, Bldg O & P
Lynnwood, WA. 98036-0918

The 1988 ARRL Handbook, 1986
American Radio Relay League

Operating Systems Network communications Protocol Spec BX.26,
Issue 2, Publication 54001, 1979
American Telephone and Telegraph Company

Pocket Packet Model HK-21 Equipment Manual
Heath Company
Benton Harbor, MI. 49022

Recommendations of the X series, Geneva, 1980
International Consultative Committee on
Telephone and Telegraph (CCITT)

Packet Radio
Rouleau, Robert, TAB Books, 1981