REMOTE CONTROL - PROBLEMS AND OPPORTUNITIES

JAMES D. SULLIVAN

MIDWEST CABLE & SATELLITE, INC.

ABSTRACT

The problems and opportunities of remote control are well documented by anyone who has wandered into the labyrinth of options and costs involved in even the most simple applications. Attempting to monitor, compile and display information from multiple monitoring points increases the complexities and costs exponentially. But, as active electronics increase, the costs of maintenance, potential failure and lost revenue also increase. As a function of limited personnel, and unsatisfied customers, we attempted to design a remote control and monitoring system. We have used the most available computers today, the IBM PC and PC Jr., and our data acquisition by "off the shelf" manufacturers. The software was mostly custom written using compiled basica and assembler as building blocks.

INTRODUCTION

As state of the art, a term whose meaning changes daily, is employed to everyday tasks there are natural concerns to the limited use cycle of the equipment. In remote monitoring very little has changed over the last several decades in basic design. While we expand our reach through use of impulse pay, a form of remote control, microwave interconnects, additional channels and deeper truck cascades, our methods of seeking the source of problems has mostly been limited to customer service complaints. Traditionally, remote control has been limited to relays and lightbulbs leaving the operators to interpret potential problems and design corrective action.

We have taken remote control into a second phase by putting to use the "intelligence" of a user programmed microprocessor.

The microprocessor is the brains of the remote monitor, with traditional relays being the nervous system collecting information for interpretation by the computer.

In the following pages, I will discuss the problems and opportunities we encountered in trying to assemble the brain and nervous system of a remote control and monitor. The complications we encountered were multiplied by the needs we had to access information from many geographically separated locations. Our application required 24 hour surveillance of electronic receivers and transmitters spread throughout the Minneapolis/St. Paul metropolitan area

In addition we had to be aware of the ongoing operating overhead of any system employed. With the apparent need to move information between the control center and the remote sites we were concerned about running up exorbitant phone line costs. In our system we did not have duplex communications available, as you might over coaxial cable. Solutions had to be found to this problem or project feasibility would be in jeopardy.

Lastly, the use of "off the shelf" hardware was of paramount concern. We had limited component production capability, as well as little spare technician time to spearhead a totally customized monitoring system. Unfortunately, the obvious choice of buying a prefabricated remote control proved too costly for the desired features, and totally inflexible to our unique needs. The result is a highbred of IBM microprocessors, Electromation Inc., interface hardware and joint custom software by ourselves and CAT Systems Inc., of New York.

The resultant system has proven to be a flexible microprocessor based remote control driven by user modified software. In the succeeding pages I will describe the techniques of operation as well as the important technical specifications used in completion of the final product.

NEEDS ASSESSMENT

Our need for remote control is brought about by the use of microwave to relay television signals from our operations center to cable television headends within a 30 mile radius. Point to point microwave communications was chosen because of its cost efficiency in sending television signals over extended distances at a minimal cost, and relatively high reliability. As we added additional paths to our microwave relay system the problem of geographic diversity became more apparent. If a tech was dispatched to a southern site for routine maintenance, it could take as long as two hours to drive to a northern site for an emergency repair. As complete failures are fairly far apart, it is impossible to convince management to add additional technicians for the occasional failures.

We found that failures were usually prefaced by deterioration of a certain component. Left unmonitored, this deterioration would not be found out until a failure had occurred. After a failure, locating the source of the problem usually ended in a response of "there's no picture". This type of assistance is worth little, except to cause panic to the service technician.

Another concern, was the cable operators who receive local commercial advertising via our microwave system. Not all operators have chosen to take our complete network feed, opting instead to switch from their local T.V.R.O. to the microwave feed only during local commercial avails. This added dimension creates a whole new set of problems relating to the inherently unreliable cue tones decoding from the satellite.

It seemed fairly straight forward that we needed an automatic method by which we could measure the performace of the interconnect. First we took a careful look at what was available from traditional remote control vendors. We found little bits and pieces that applied to our problems, but no clear solutions jumped out at us. It looked like we would just keep going along with our one tech running about putting out fires.

MICROPROCESSOR TECHNOLOGY

With the falling cost of microcomputers and the availability of prepackaged software a solution to our monitoring seemed near at hand.

First we set a listing of functions that were deemed essential to the project.

°Cost effective
°User-friendly
°Minimize use of phone lines
°Display information with easy to read graphics.
°Provide hard copy log of out of spec operations.
°Save data for reliability and outage forecasting.
°Verify the switch positions at each headend.
°Verify all voltages and alarms on receiver at each headend.
°Verify all voltages and alarms on transmitters.
°Monitor security, fire/smoke alarms and air temperature at transmitter building.

With this basic outline several of us sat down to decide which areas would get developed internally and which by outside vendors. Our initial plan called for a Commodore VIC 20 micro. Using the game part as I/O inputs we felt that a modest system could be built. After buying one unit and playing with the available software it was decided to go to an IBM PC Jr. micro for increased performance and additional flexibility. The most important of which was the floppy disk drive which made blowing PROMS unnecessary.

Once we started trying available software i.e. PC TALK, LOTUS, and PROKEY we found that it was going to take a more custom software package to give us the features we had identified. We called Joe Soll at CAT Systems, Inc. in New York, who has had considerable experience with custom software applications in remote control. We got together and came up with the following ideas for the software.

°Compiled BASICA should be used for speed in programming and versatility in making changes.
°Assembly language would be used for all I/O functions, i.e. communication, screen drawing, and light pen control.

As we went through possible configurations, it became clear that the software could be broken down into blocks. This method allowed us to see more clearly the relations between different functions that we wanted to perform. The peripheral or auxiliary programs to the main program were fairly self explanatory. Their relation to each other is best described by the following block diagram.

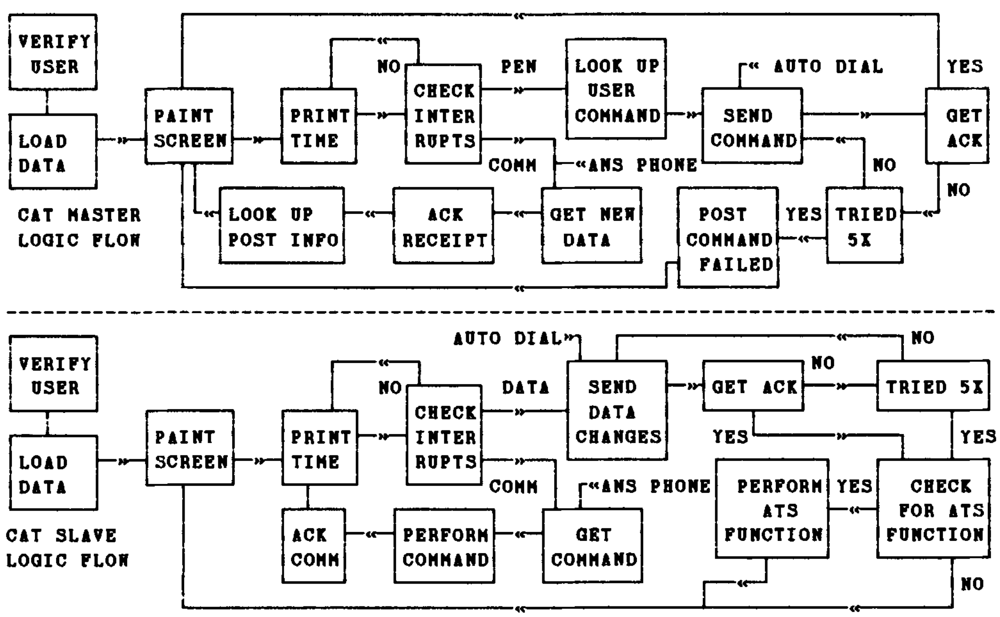
Communications Protocol

Communications Protocol was identified as one of our most important programs. The error free movement of information between remote site and control site is essential to basic reliability. We were unable to find "off the shelf" programs that combined the speed and accuracy features we needed. Joe worked up a protocol for us that gave us the features we required. The communication program runs in assembly language. This allows the program to run in the background. A baud rate of 1200 was selected as the optimal compromise between speed, modem cost and phone line compatibility. As we established duplex capabilities on the interconnect, a second modem will be added at the remote site operating at 9600 baud with the 1200 as a standby in case of path fade or radio failure. Naturally, data errors were a big concern as phone line quality varies from connection to connection. Joe was able to devise an error checking scheme that does not appear to slow the system down. Errors appearing at the control center have been nonexistant to date.

Another major concern for any phone line access computer is security. With the explosion of "hackers", (we've all heard the horror stories) security questions had to be resolved. Our solution came by way of using a binary protocol running on asynchronous modems. By using binary codes instead of ASKI words, there is never a prompt on a screen. In fact, when the computer picks up an incoming call it waits for a 50 digit number that it must receive before any transmission is sent. This method appears to be hack proof. Additionally, the system goes off line after 3 unsuccessful attempts to enter the code sequence.

site it corresponded to. Within each box a summary of status could be viewed. For instance, the 60V power source used on the AML would be shown as a green 60. But if the AC voltage went outside its predetermined alarm points the new voltage, say 50, would be in red. Also, in each box is the time of the last communication. The second section of the screen was reserved for commands and system status. This area contains "buttons", illuminated squares that when "depressed" by light pen, command the software to carry out preset functions. For instance, the dial button, when depressed, sets the control site computer to call a remote site. After depressing dial, the light pen is then put against the square that has the desired site name. Once this is done, the computer autodials the remote and retrieves the current status information for display at the control site.

The third screen section will display this information in its expanded state. All voltages are brought back in their entirety and can be displayed in this area. Additionally, the site can be left on line

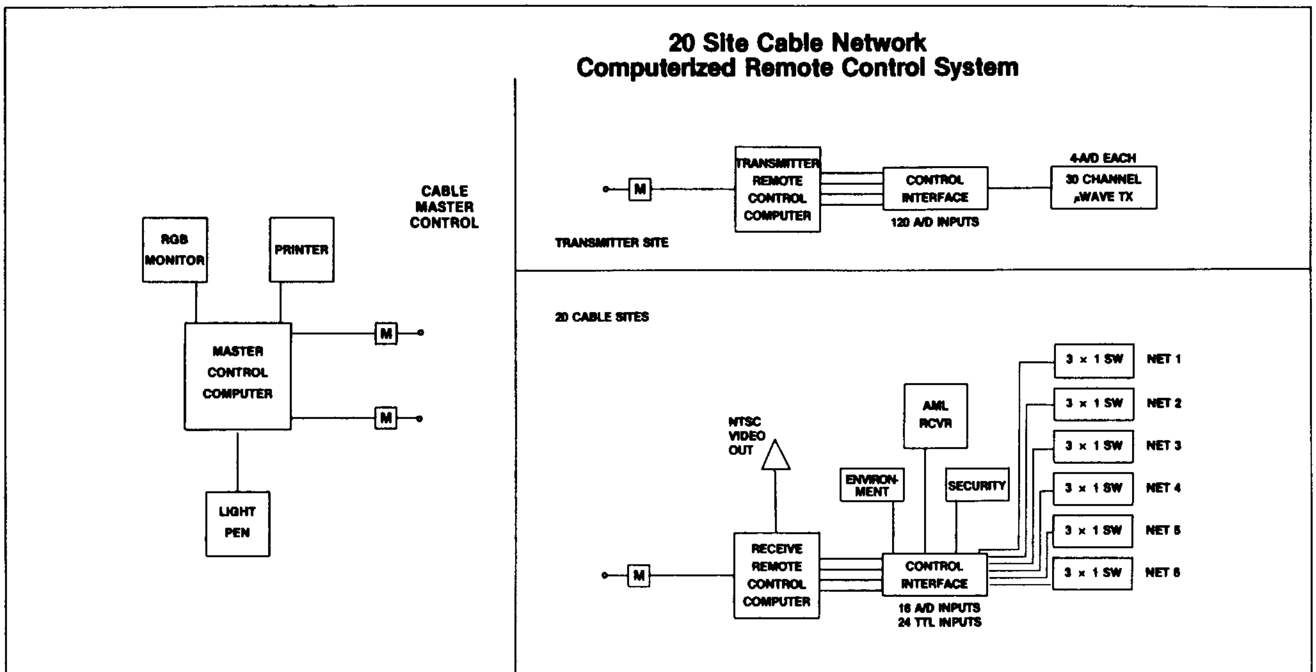


Display

Display of information in a concise, easy to understand format was also an important goal in our needs assessment. We did not want to use the traditional L.E.D. and alarm type display. Instead we chose an R.G.B. monitor to display our reporting information. With the help of multiple colors it was fairly easy to code most information for easy retrieval by the most casual glance. The control center screen was broken into 3 sections. The first being 20 small squares each labeled with the remote

for active viewing of a particular site's parameters.

At the remote sites, the PC Jr. is capable of an NTSC output. We took advantage of this and put a display routine in the remote site hardware. At the remote, all system parameters are displayed on one screen in logical order. The same color coding is used - red for alarm, blue, green or yellow denotes normal operation.



SPECIAL FUNCTIONS

DAILY PRINTOUT OF LIST OF NETWORK SWITCHING INFORMATION FOR EACH SITE.

DAILY PRINTOUT OF LIST OF NETWORK SWITCHING DISCREPANCIES.

DAILY PRINTOUT OF LIST OF ALL FAULT OCCURRENCE INFORMATION.

STORAGE AND ANALYSIS OF FAULT OCCURRENCE INFORMATION.

TTL INPUTS

HI LOC
LO LOC
DC SUPPLY
18 NET SW POS.

A/D INPUTS

AC LINE
24 VDC PSU
20 VDC PSU
PHASE LOC
SOURCE LOC
AGC
TEMPERATURE
SOURCE ALARM

SPECIAL FUNCTIONS

STORAGE OF ALL NETWORK SWITCHES AND TIME OF SWITCH

DAILY CALL TO MASTER TO DOWNLOAD SWITCHING INFORMATION

STORAGE OF ALL FAULTS, TIME OF FAULT AND DURATION

CALLS MASTER AT TIME OF FAULT AND TIME FAULT CLEARS

DAILY CALL TO MASTER TO DOWNLOAD FAULT INFORMATION

Light Pen

Light Pen software is also run in machine assembly language. This routine is straight forward and available from several vendors. The importance of the light pen for our application is the reduction in computer literacy required for operation. One problem we found with most computer based systems is the need for extensive operator training. We strove for simplicity in operation as a key point in software design. Once the program is up and running all commands are activated by light pen control.

I/O

I/O, or inputs and outputs, are also on the computer buss using machine language. When looking for an interface vendor it was important to determine what language they used for communication to the main program. Many of the interface vendors supply support software for interpretation of their data. We chose a vendor that would give us the most basic level of interface. By using machine language, directly between the main program and the I/O, we are able to save processing time. We found that by speeding up the simple communication tasks, the software was more user friendly and all around less frustrating for the operator.

Printer

Printer tasks are accomplished with a small EPSON dot matrix printer. Whenever there is out of specification data, the screen changes and the information is dumped to the printer for hard copy record. Both the alarm and the time of day are recorded. In the event that an alarm is cleared, that also is posted to the printer.

The Central Program

The Central Program ties each of these auxiliary routines together into a cohesive remote control and monitoring system. When the program is loaded from disk into memory, a RAM drive is created. This saves the computer from running the disk drive each time new screen information is needed. Software response time is also greatly enhanced by the increased access speed of a RAM drive.

The main program contains the lookup tables and multipliers for each of the I/O ports and light pen. Therefore, when a voltage test point is outside the capture range of our A/D (analog to data) converter the software can compensate for any external voltage dividers used. This is an important feature as it allows us the flexibility of changing input values without worry of calibration problems.

The main program is quite simple in its routine. Its basic responsibility is to keep track of time. Between each clock cycle the program looks to see if anything has changed with the other routines. For instance, a light pen interrupt would be present if someone entered a command. The program gets the screen coordinates for the pen interrupt and goes to a look up table for directions. In the look up table it might say dial. If so, the main program would then tell the communication program to ready itself for autodial and so on. Each time a status is changed, the program simply looks up what it is supposed to do in the table and follows out that command.

The importance of this design is that Joe has left us the capability to change the software by simply adding points of reference to the look up table. When we went to add more features, it was a simple matter of changing the look up table and the auxiliary program associated with the function desired.

For our application, phone line costs are of paramount concern. In an effort to conserve costs we are using autodial modems. Fortunately, we have been able to keep all of our sites within the calling zone of our local Bell operating company. I can see other applications where this may not be possible making the flexibility in calling routines a nice feature. By having a microcomputer at each monitoring site we are able to save data and make decisions on a communications emergency. When a parameter goes to an alarm status the computer is forced into a dial routine, sending only that fault data to the central computer. Hopefully, though, we will not be receiving alarms from every site daily. Because of this, we built in a "handshake" routine that tells the remote site to call every three hours. This time span was an arbitrary selection and can be changed depending on the cost of each phone call. The handshake takes less than 5 seconds, this keeps the control site computer available to receive calls from other sites. If a site does not report within its three hour time frame, the central computer initiates a call to the remote. Failing a connection, the central computer puts up a "no communications" alarm.

Nightly the central computer calls out to each remote site for a data "dump". This is done between 1:00AM and 6:00AM. During the dump, the remote site sends it's logs of switching changes from local TVRO to the microwave interconnect. This information is then compared against local switches for a discrepency log. During a typical day, we save approximately 16K of switch data in RAM. Using 1200 baud modems, the system takes about 5 minutes to send and verify all of its information.

If during a data dump or any online period another remote station calls with an emergency, that remote site will get a busy signal. As we only have one input communication port at this time, that remote site simply redials until it gets through.

The next step in our system development will be saving all data to a hard disk drive. We are now working out the details of a trend analysis program. It seems obvious that, as we have all of this raw data on the micro, a natural extension of the system is trend analysis. By capturing all of the data in a file labeled with the site name and date we will be able to go back and retrieve any number of statistical data runs. Using an external program and graphing the information it should be easy to interpolate the information into a forecast. For instance, an STX-141 transmitter tube's current and voltage could be plotted on an X,Y graph showing us the rate of decay in the klystron tube. Hopefully we will be able to predict the tube failure beforehand and thereby saving expensive down time and express freight from the Hughes Microwave plant in California. We hope to finish up this part of the system by year's end.

HARDWARE

When it became apparent that we had software that would answer most of our problems we began hunting down the parts necessary to interface the IBM PC and PC Jr. to the outside world. At first glance we considered building in house the necessary interface boards to acquisition the data. The following list was made identifying the parameters of an interface module.

Rack Frame

- °A rack mountable unit to support both the interface and the unrackable PC Jr.
- °Mother board configuration to allow expansion and exchange and back up of parts.
- °User addressable memory location by dip switches on mother board.

°Ability to stack several units for increased monitoring capabilities.
°Easy to wire back plane, split cylinder type punch down connectors.
°A common numerator for expansion. (48 was chosen)

Analog to Data

The Analog to Data conversion A/D is one of the most important inputs in our application. On the Hughes AML microwave equipment, all alarms and test points are analog voltages. Because the computer can not read an analog voltage directly, the interface must convert the common voltage to a digital number for computation. As the software was being developed, it was decided that all voltages should be read directly, leaving conversion to the software running in the micro. This seemed most flexible and has proven to be quite useful. The following list outlines the important parameters to solve our application problem.

A/D

°A minimum of 16 A/D inputs.
°Differential inputs, to avoid ground loops between equipment, and to allow for more accurate metering.
°12 bit resolution giving us a 4096 scale divider.
°Jumper selectable input ranges from .5 Vdc to 20Vdc.
°A processing rate of at least 10 channels per second. While this rate is not considered fast, money can be saved by making fewer tests per second.
°Use a precision multicurrent temperature compensated zener voltage reference for most accurate test reference.
°Measurement accuracy should be within 2%.
°DC inputs only, all AC externally rectified to reduce risks to technicians working around backplane.
°Peripheral interface converts all measurements directly to machine language for connection to PC boss.
°Leave all offsets, linear or logarhithmic slopes to be figured by computer software program.

Status Inputs

Status Inputs are another important feature for monitoring the positions of the audio/video switchers at certain sites. We can also use TTL inputs for checking a variety of alarms and status conditions. The following list defines what we considered essential in our application.

°A minimum of 16 inputs per card, a multiple of our 48 possible points in a rack frame.
°High level inputs, jumper selectable up to 24 volts DC either positive or negative in polarity.
°TTL level inputs, an industry standard used by most manufacturers.
°Contact closure input, where relays are closed for status indication.
°Opto coupled inputs to protect the computer from external voltage spike, technician error, and reducing ground loop problems between equipment.
°Use of an onboard peripheral interface and adapter to convert input status to machine language for interpolation by computer program.

Control Outputs

Control outputs are a future addition that we want flexibility in adding. While we have no current need to control equipment at the remote sites, there may come a time when we want to perform simple remote controls i.e. changing an audio/video switch, switching to backup receivers, or controlling test equipment. For our applications, we could foresee relays providing the most flexible point of control. A few of the desired features were:

°Single pole double throw relays.
°Have both poles and wiper on back plane for normally open or normally closed wiring.
°Loose power and all relays normalize.
°Better the 500V isolation between contact and coil for surge protection.
°Operating power limited to 24V AC or DC 1 amp current, all controls needing higher voltage or current would be done with off board relays.
°Peripheral interface adapter to convert relay information to machine language.

With this basic design criteria set out, we looked through "Byte Magazine" for potential suppliers. We selected a local firm that was willing to "customize" their product to our needs. Electromation Inc., in Circle Pines, Minnesota was able to put together what we needed at a modest cost. They also were able to deliver a few bells and whistles that we had not really contemplated. The most important was their Failsafe module.

Failsafe

Failsafe is an important feature in a control system. The failsafe module was built for broadcasters that need to meet certain FCC rules for remote control. We found that its 'reboot' feature would be a lifesaver for us.

When there is spurious electricity, or a brown out, a term that means partial electricity, a computer can lock up the software, halting any continued operation. As a means to combat this problem there are several vendors that supply U.P.S., Uninterruptable Power Supply, for brown outs and surge protection for spurious electrical damage. While the failsafe card will not protect the computer from electrical damage, it will cause the computer to go into a reboot sequence. If communications stop between the interface unit and micro, the failsafe card will put the relays into a preset mode, user selected by internal dip switch. This feature will be more valuable to sites that use more extended control features. At this time, we plan to use only the reboot section of the card.

SUMMARY

The remote control and monitoring system outlined in these pages has become an integral part of our overall test and maintenance routines. On the expense side, our monthly cost, including operating overhead and depreciating capital, yields a monthly expense of $92.08 per remote site. The dollar savings minimized in outage and response time has paid for the system many times over. By incorporating existing microcomputer technology and common sense software principles, we have been able to insure optimum signal performance to our cable viewers.

I am certain that our needs and applications will change in the future. The system design principles we have implemented will insure flexibility to meet our needs well beyond the 5 year depreciation of the equipment.

ACKNOWLEDGEMENTS

It is important to mention that this monitoring system was the fruits of many individuals labor. The entire engineering staff at Midwest Cable Satellite, Inc. pitched in to make the project a reality.

Also the unique and personal attention we received from Joseph M. Soll, President, CAT Systems, Inc., New York, and David C. Lunder, President, Electromation, Inc., Minnesota, were instrumental in putting the project past the blackboard stage and into the cable headends.