Biotelemetry X
July 1 to August 5, 1988
Fayetteville, AR, USA

79

USE OF SMALL COMPUTERS AS TELEMETRY DATA COLLECTORS

Valerian B. Kuechle
James M. Haynes
Richard A. Reichle

ABSTRACT

Personal computers and single board computers were used as telemetry receiver controllers and data collectors. Apple II series computers are used where main power is available. Receivers may be placed up to 1000 meters from the computer with a pair of wires to pass data. Battery powered single board computers are used where power is not available. Interfacing the computer to the receiver and measurement algorithms are described. Criteria used to detect low quality data are discussed. Programming is in basic language with calls to assemble language routines where speed is critical. The programs can be readily modified by the user in the field. Methods of data handling to facilitate interfacing to spread sheets and data bases are also given. Data in the Apple system is stored on floppy disk and in RAM on the single board systems. Data in the single board systems is off-loaded using an RS–232 interface. The systems have been used to collect presence/absence data, temperature data, and heart rate.

INTRODUCTION

Most data in the wildlife field are collected and recorded using pencil and paper. With the introduction of temperature, pressure, and other parameter measurement transmitters continuous data recording is desired. In addition, many studies desire presence/absence data. Some of these data have been recorded using Esterline Angus or other strip chart recorders, (Siniff et al., 1971; Gilmer & Kuechle, 1971) however, these methods require the tedious task of taking the data from the strip chart and transcribing it to a computer for analysis.

It is possible to design a data collection system using a dedicated microprocessor, but design costs for these systems are usually high, and they require experienced programmers. Secondly, they are difficult to modify if the experiment requires a change. We have designed and used two different but similar systems that overcome most of these problems.

THE SYSTEM

The general requirement of a wildlife radio tracking data collecting system is interfacing a radio tracking receiver to a computer for control of the receiver and to access the data from the receiver. Once the computer has received the data, it is desirable to check the data for errors and store the data in a format amendable for later analysis. In addition, because requirements often change in the field, we desire an easy method of changing the program (software) to satisfy the required modifications.

The first of these systems used an Apple II series computer to monitor the temperature selection of fish as they traversed the thermal environment of the warm water outfall near a coal field power plant. This system used three receivers—one at the computer location, one 400 meters from the computer, and the third 750 meters from the computer. Main power was available for the computer; automobile storage batteries were used to power the two outlying stations. The receivers used were built by the University of Minnesota bioelectronics lab and had a digital interface for frequency control and data output.

The computer was modified by the addition of an Aptech Controller Board (Systems Manufacturing Technology, Inc., San Marcos, California) into one of the vacant slots on the Apple II. This board has a real time clock and circuitry to control and read data from remote slave stations. The controller used pulse width modulation for data transmissions and was based on the MM 54240 Asynchronous Receiver/Transmitter Remote Controller from National Semiconductor (Santa Clara, California). The MM54240 allows transmitting and receiving data from remote sites using a single pair of wires. The remote sites have a similar circuit board that acts as a slave station. The controller in the Apple computer activates a particular slave station which is controlled by sending an address specific to that site. Addresses of the slave sites are set by switches using a binary code. The 16 input/output lines are also programmed as either input or output using on-board switches. Communication between the devices is done in two parts. The first 7 bits transmitted are address bits, while bit 8 indicates read or write. If an address match is found, the slave either receives the data from the computer or sends the computer its data. This is determined by the status of bit 8. Data are transmitted as 8 bits with a parity bit. We used RG-58 coaxial cable to communicate between the controller and slave stations.

Programming was done in the basic language with calls to assembly language sub-routines. Only speed-dependent and controller parts of the program were written in assembly language. Using this technique the program can be easily modified in the field because the assembly portions are not changed.

The second version of the data collector we used is based on a single board computer "Tattletale" supplied by Onset Computer, North Falmouth, Mass. This single board computer uses a Hitachi 6303 microprocessor and is supplied with a resident basic interpreter.

We use the Model II which has 256K Dynamic Random Access Memory (RAM) for data storage. The Model II has 14 digital input/output lines. Since we require a minimum of 13 lines for frequency control, shift registers are used to output receiver controls. Then a sufficient number of data lines remain for data input and other control purposes. Communication between the computer and the user is with an external terminal using an RS-232 interface at 9600 baud and full duplex. Data offloading is also achieved via the RS-232 interface. Power to the computer is supplied by external battery. The average drain is approximately 2 ma with peaks of 70 ma during memory refresh and memory write.

PROGRAMMING

Programming for both devices is similar. Programs are written in basic with calls to assembly language routines if necessary for speed. The basic portions and assembly language portions of the program are written in modular fashion to allow easy modification and debugging. Menus are used to enable operation in the field without referring to an operator manual or operating codes.

In the Apple computer, the controller is designated and data sent via the memory I/O space as assigned by Apple. With the "Tattletale," the basic program allows specifying individual pins as input or output and as high or low. All data transfers for the "Tattletale" use these I/O lines.

The usual mode of operation is to set up a frequency table in memory. Scan times can be designated to be as fast as possible with a fixed time on each channel or may be grouped to start at a pre-assigned time, scan through the frequency list, and then wait until the next start time.

Data are usually collected as pulse rate information, either number of pulses in the case of location transmitters or as a variable pulse rate for temperature, pressure, or heart rate. For the Apple computer, the data is sampled every 12 ms to determine whether data is a 0 or 1. The principle speed limitation is the time it takes the controller to communicate with the slave site. Averaging is used to reduce the error to less than 5 ms. Data acquisition for the "Tattletale" is easier because the basic interpreter includes period and count functions.

If this method is not accurate enough, we use clock interrupts and an assembly language routine to determine the period. This technique is used to measure heartrate and body temperature on a pulse-by-pulse basis. The period and count functions can be used directly to measure period and frequency to approximately 1.3 microseconds accuracy.

Once the data points have been sampled, they are averaged and the pulse rate data applied to a regression fit for temperature or pressure calibration. The regression equation coefficients are stored in the frequency table.

Good data is separated from bad or marginal data by performing an analysis of variance on the pulse rates. If the variance is low, the data is good; if high, the data has most likely been corrupted by noise. Our experience indicates that choice of variance to decide good or bad data is not critical. For the Apple system, a value of 15 was used. The "Tattletale" system's basic interpreter only allows for integer numbers; to overcome this problem, we scaled all numbers to maintain accuracy.

A second method we used to enhance data was to look for patterns of pulses when noise may have inserted extra pulses. The software starts with the first pulse and looks for another in a zone near where another pulse is expected; if none is found, it looks to the next pulse and attempts to find a match. This process is repeated until the specified number of successive matchup is found. The width of the acceptable match zone and the number of successive matches required, determine the speed of the process. If a valid pulse is found; successive matches are quickly found.

Pulse width can also be used to discriminate pulses. Since we use a tone decoder to detect pulses, pulse width is already highly variable dependent on signal strength and thus not usuable in our system.

DATA STORAGE AND RETRIEVAL

In the Apple system, data was stored on disks using sequential files. Data can then be analyzed by employing a user-written program or imported to another data analysis program. We also use a short basic program that filters data from the disk for a quick look at a particular frequency while in the field.

With the "Tattletale" system, data can be off-loaded in ASCII form using a basic program or may be off-loaded using x-modem protocol. Data is stored in binary format to maximize memory storage capacity. X-modem off-loads block for block from the memory and thus data remains in binary format. It must then be converted to ASCII in the host computer before data can be imported to analysis programs. The primary advantage of using x-modem protocol is speed. Using x-modem, 224K bytes of memory can be off-loaded in about 10 minutes; whereas using ASCII and an ordinary terminal program could take several hours depending on the terminal used.

ADVANTAGES

The primary advantage of these systems is that they are software based and can be easily modified. Also they are assembled from widely used computer boards where possible design problems have already been detected. Using basic as the primary language makes errors easier to find because they are usually identified. Additionally most basic interpreters have "ONERR" constructs which allow programming for unattended error recovery. Correcting and detecting errors in assembly language requires special tools because no indication of error is given.

PROBLEMS AND IMPROVEMENTS

In the Apple system, we had to add extra line drives on the coaxial cable to reduce the errors. We do not believe this would have been necessary had we used a twisted pair cable instead of the RG-58. Twisted pair cable has much lower capacitance per unit length than RG-58 cable. Another problem we have encountered is that, even though basic and modular programs are employed, users are reluctant to modify programs in the field. Extensive user manuals are required for unassisted field operation. Further improvements in menus may also be helpful. Computers are noise generators and measures must be taken to reduce this emission or sensitivity will be reduced. We enclosed all circuitry with metal and used low pass on all interfacing cables.

ACKNOWLEDGMENTS

Development of the Apple Computer System was funded under contract from New York State Electric and Gas Corp., 4500 Vestal Parkway, Binghamton, NY 13903.

REFERENCES

Gilmer, D. S. & V. B. Kuechle, 1971, A device for monitoring radio marked animals, Journal of Wildlife Management, 35(4): 829–832.

Onset Computer Corporation, May 1988, Tattletale data logger operations manual.

Siniff, D. B., J. R. Tester & V. B. Kuechle, 1971, Some observations on the activity patterns of Weddell Seals as recorded by telemetry. Antarctic Pinnipedia, Antarctic Research Series 18: 173–180.