1985

A Comparison of Analog and Direct Digital Control (DDC) of a Physical Process

Clifton R. Hargrove
University of Central Florida

A Comparison of Analog and Direct Digital Control (DDC) of a Physical Process

by

Clifton Rufus Hargrove
B.E.T., University of Central Florida, 1983

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1985

ABSTRACT

The following report presents the development of a computer-based system designed to examine the response of a physical process while under different control systems. The physical process is a process trainer, PT326, from Feedback Corporation.

There are two main categories of control available to the user: (1) analog control and (2) digital control. Within each of these categories the following types of control systems are available : (1) Proportional control, (2) Proportional-Integral control, (3) Proportional-Integral-Derivative control and (4) On-Off control. The user can examine both regulator and servo response for any of the given control systems. All parameters associated with the selected control system are variable and user-input. The software is designed to be user-friendly.

The user examines the response of the process to different control systems through the use of full-screen, high-resolution, color graphs. These graphs are linear and labeled. There are two main categories of graphs available to the user: (1) real-time monitoring and (2) graphs with labeled time base. Within each category there are four different graphs available. These graphs are available under any given control system. The user may also obtain hardcopy of these graphs.

ACKNOWLEDGEMENTS

I would like to take this time to thank those who have been instrumental in my education. I wish to thank my parents for their help. I also want to acknowledge Dr. Harold I. Klee and Theodore Wesson. Dr. Klee served as my advisor, teacher and as a member of my thesis committee. Theodore Wesson supplied answers where they were needed.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

FIGURE

INTRODUCTION

This study concerns the features, limitations, and possible uses of a computer control system that has been developed. The system was designed to allow examination of process trainer PT326 from Feedback Corporation. A description of the physical process is given in Chapter 1.

The system allows examination of the process under the following types of control:

(1) Analog Proportional control

(2) Analog Propoortional-Integral control

(3) Analog Proportional-Integral-Derivative control

(4) Analog On-Off control

(5) Digital Proportional control

(6) Digital Proportional-Integral control

(7) Digital Proportional-Integral-Derivative control

(8) Digital On-Off control

The system can examine regulator or servo response under any of the above-mentioned control systems. For the sake of brevity, the following report assumes that the reader is familiar with these types of control systems.

The user examines the process by looking at a graph. These graphs are drawn in high-resolution, color graphics on an Apple IIe microcomputer. These graphs are full screen, labeled and calibrated. The user has four different graphs available to aid in examining the response of the process. These graphs are available in two different forms: (1) real-time monitoring and (2) graphs with labeled time base. All graphs are available under any type of control system.

CHAPTER ONE - PHYSICAL PROCESS

SECTION 1 - Description of Process

The process trainer, PT326, from Feedback Corp. is shown in Figure 1. A diagram of the process and control board is shown in Figure 2. The process consists of a blower forcing air through a heater mounted inside the tube. The blower opening is variable from 10 to 170 degrees and is calibrated (see Fig. 1). Once the process is started the blower motor runs at a constant speed. The amount of power output to the process is variable and is used to control the temperature of the process.

A thermistor based detector , with three possible positions (see Fig. 1), monitors the actual temperature of the process. If the the user places the detector anywhere except as close as possible to the heater then there will be an additional distance delay. The detector yields a voltage level corresponding to the temperature detected. This voltage can be monitored at point Y (see Fig. 2). This voltage is used to power a calibrated gauge on the control board. The user selects a desired set point by reading a calibrated gauge of the same type and adjusting the set value knob on the control board (see Fig. 2).

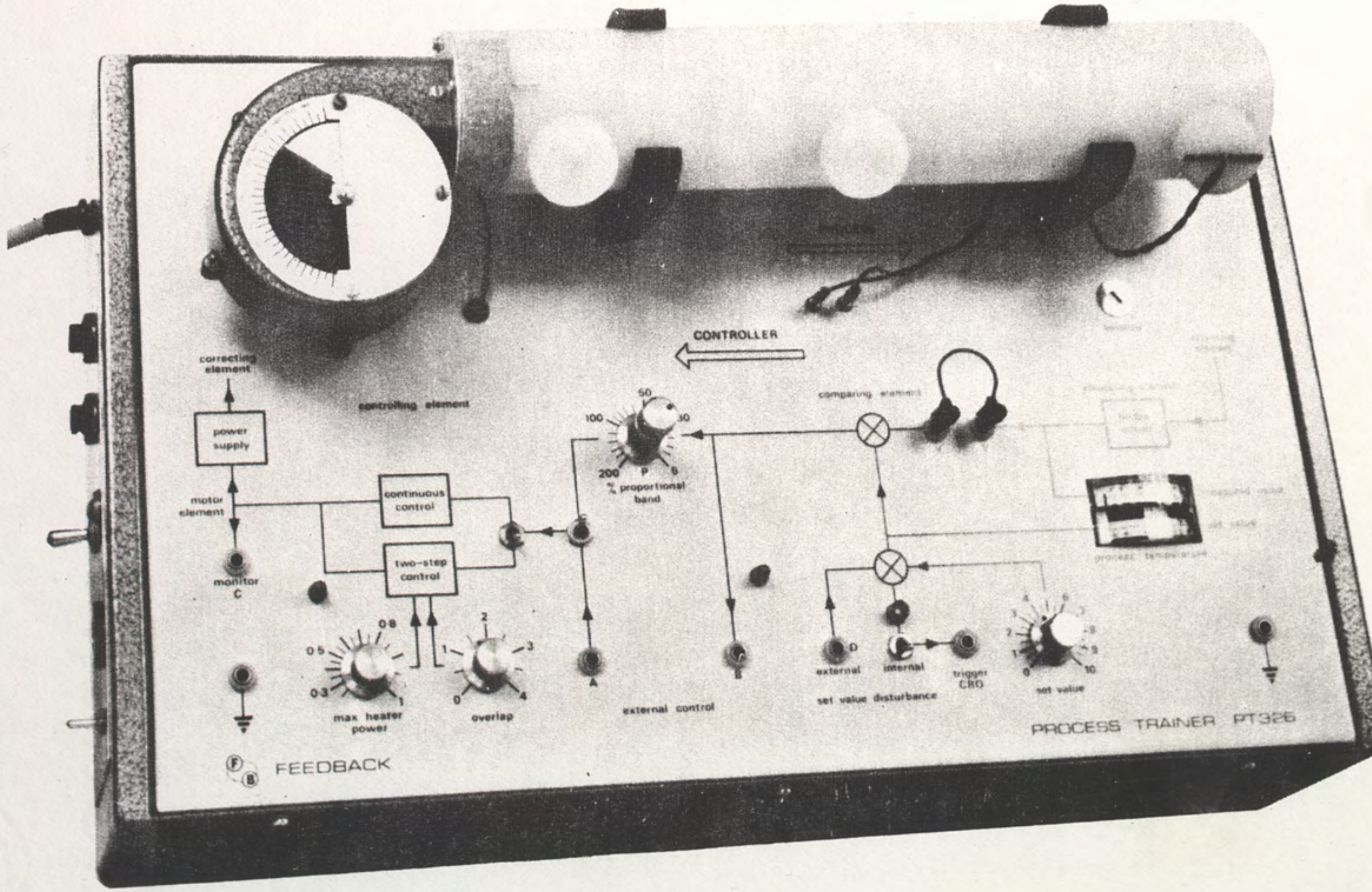


FIGURE 1 - Illustration of Process and Control Board

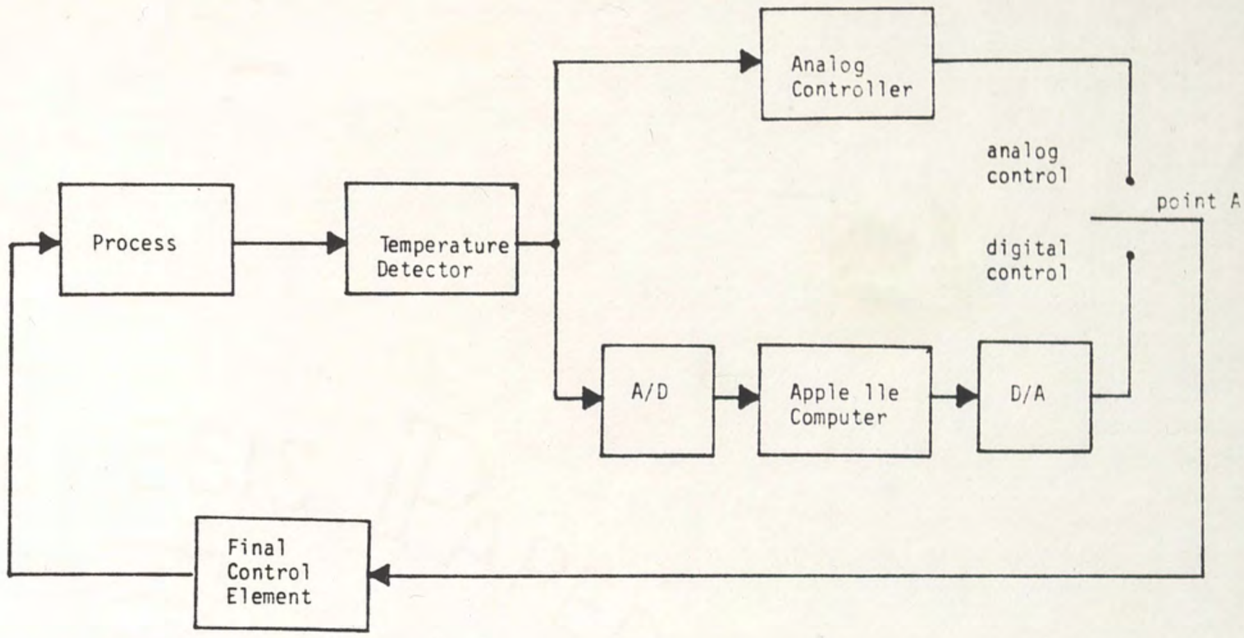


Figure 2a - System Overview Block Diagram

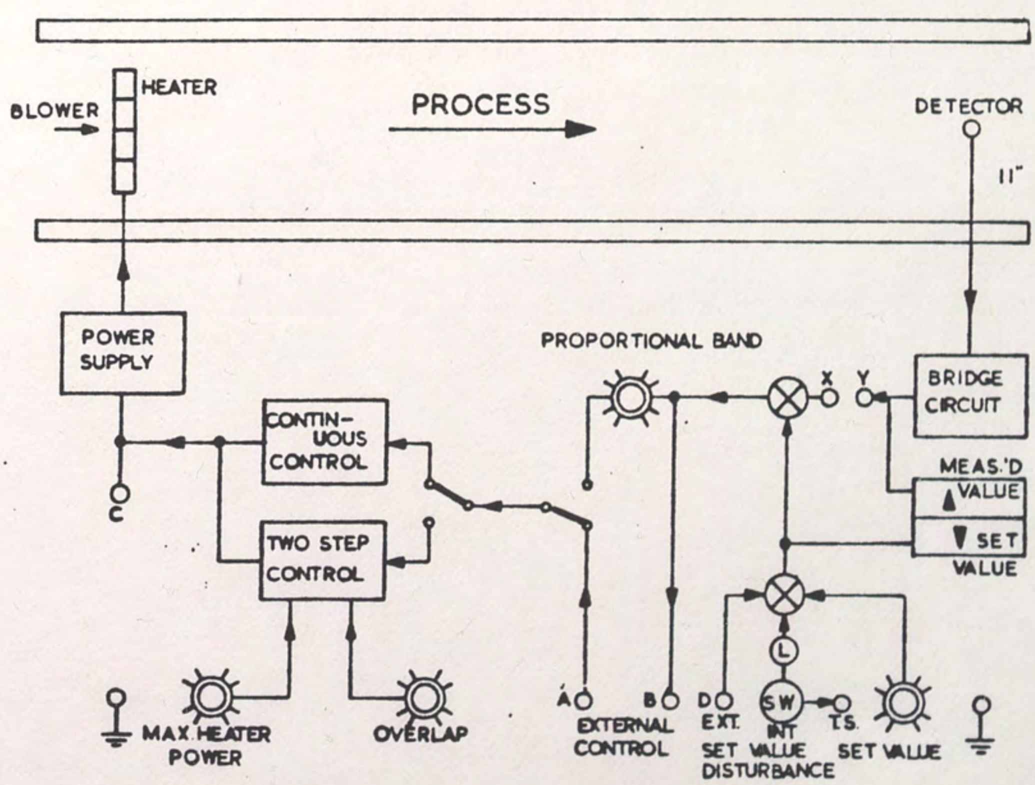


Figure 2b - Diagram of Process and Control Board

FIGURE 2 - Diagram of Process Control Board and System Overview

The power supply shown is a voltage controlled power supply. The amount of voltage input to it determines the amount of power that is output to the process. The amount of voltage input to the power supply can be monitored at point C (see Fig. 2).

The final result of everything mentioned thus far is that we now have the ability to "monitor" all of the process variables. All that remains is to translate these voltage levels into useful information for the computer to use.

SECTION 2 - Determination of Physical Requirements

It can be shown that all of the process variables have a unique voltage corresponding to a given condition. Figure 3A shows the relationship of the power supply output versus input voltage. Figure 3B shows the relationship between the temperature transmitter output voltage versus process temperature. The data points to support these graphs are given in Appendix A.

Upon examining the temperature versus voltage relationship, it is obvious that the temperature swing must be limited. One reason is that the interface device can only look at a limited range of voltages. Another consideration is that below 30 C and above 60 C the temperature versus voltage relationship becomes nonlinear. As the reader might suspect the temperature swing is limited from 30 C to 60 C. The voltage of the temperature transmitter at these two endpoints is:

TEMPERATURE	VOLTAGE
3ØC	2.65V
6ØC	11.5V

The interface device selected will only accept analog input/output voltages in the range of -5 to +5 volts. The problem is not the voltage swing itself since it is less than 1Ø volts (8.85 volts). The problem is that the voltage region in which the swing takes place is too high. One solution is to bring down the region of the voltage swing i.e. add -7.65 volts to the transmitter output voltage. This voltage is added to the transmitter output voltage through the use of an operational amplifier summer circuit. Figure 4 shows a schematic of the circuitry. The third circuit shown is for digital control and will be discussed later.

As a result of this voltage being added, the following relationship now applies to the temperature transmitter.

TEMPERATURE	VOLTAGE
3ØC	-5.ØV
6ØC	+3.85V

SECTION 3 - Additional Equipment Required

This section discusses equipment utilized by the system in addition to the circuitry just discussed. The following is a comprehensive list of the equipment utilized:

(1) APPLE IIe microcomputer - 64 K RAM, 3.3 Disk Operating System

(2) TAXAN Color Monitor

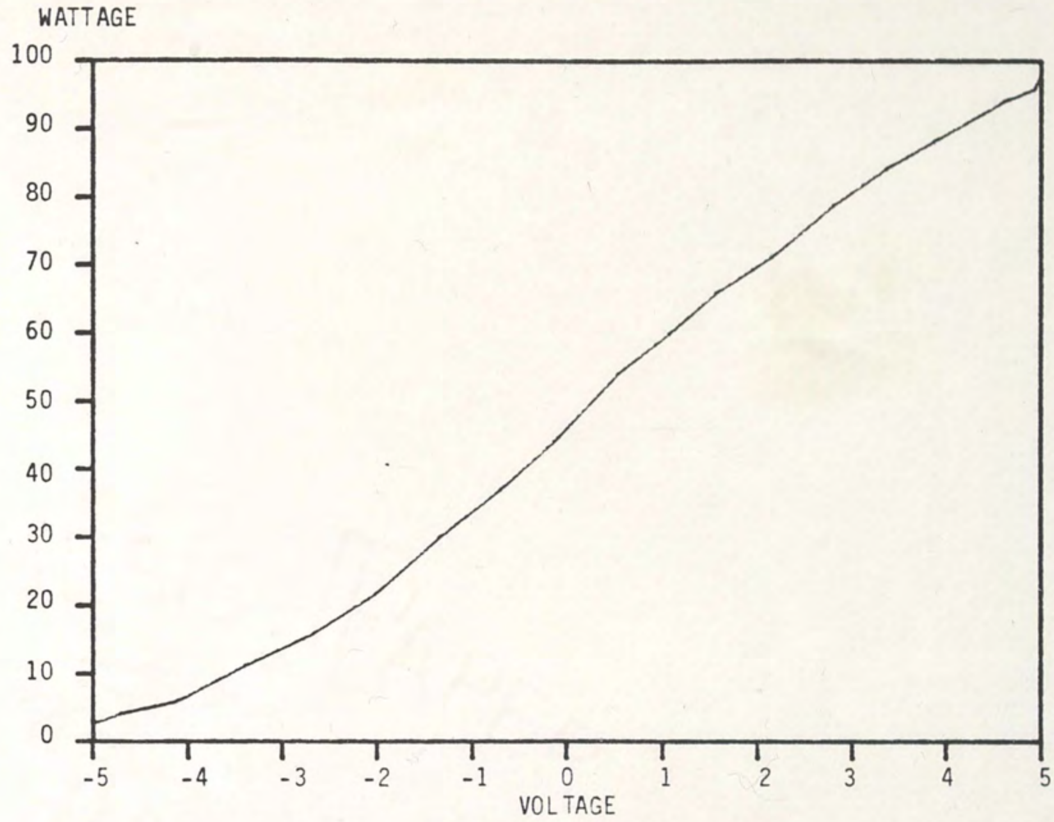


Figure 3a - Power Supply Output versus Input Voltage

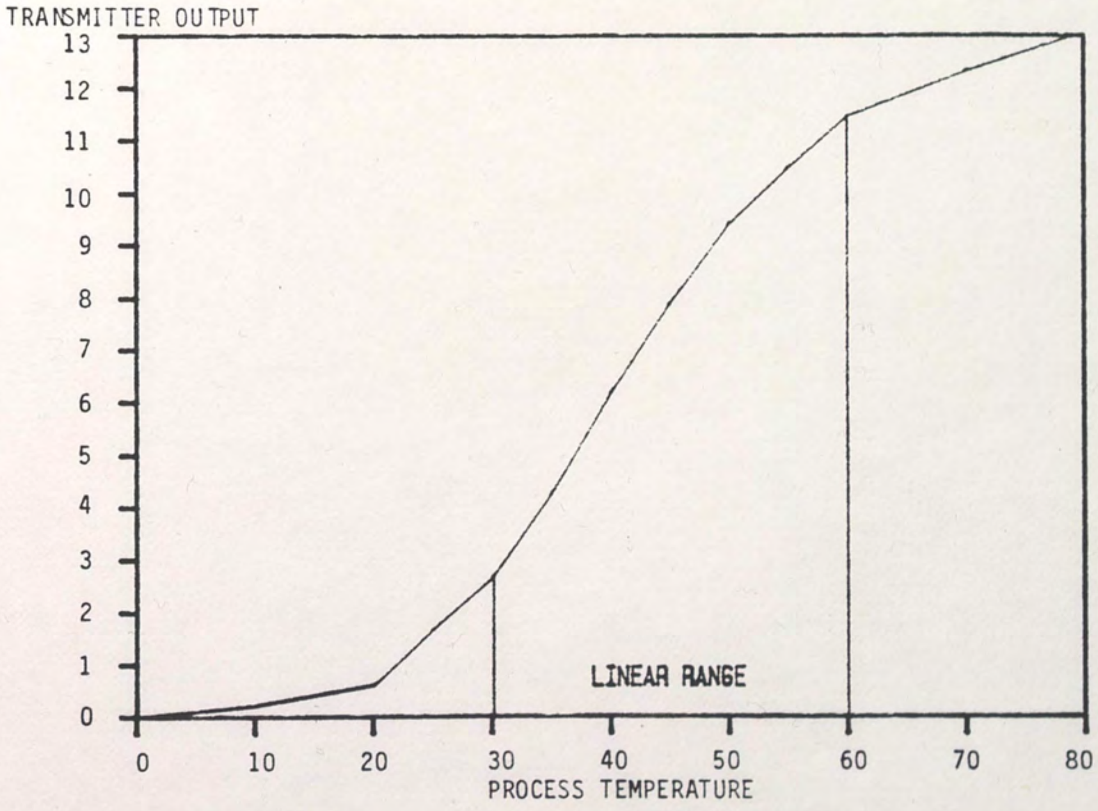


Figure 3b - Transmitter Output Voltage versus Temperature
FIGURE 3 - Manipulated/Controlled Variables versus Voltage

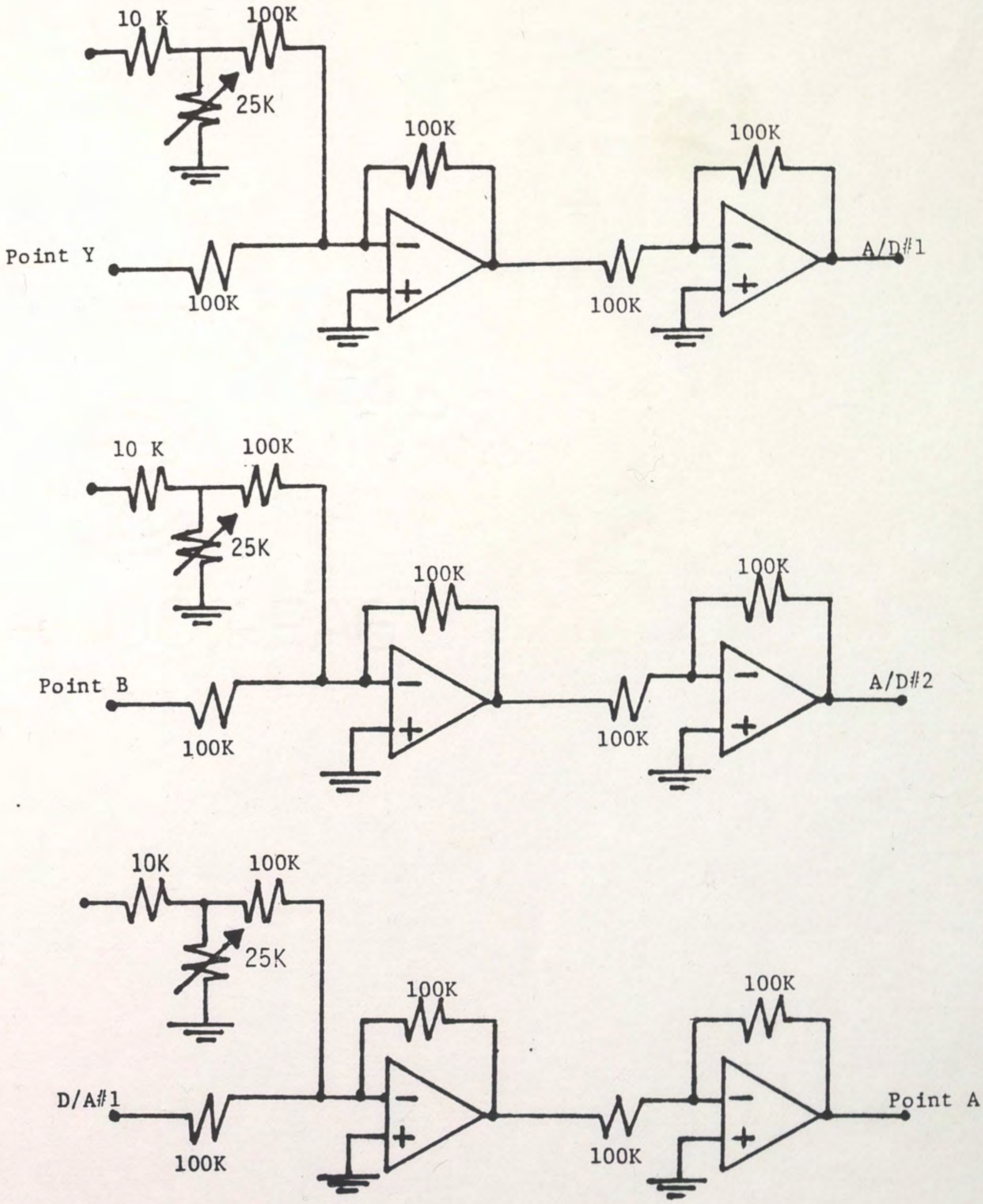


FIGURE 4 - Schematic of Interface Circuits

(3) OKIDATA 92 microline printer

(4) ISAAC (Integrated System for Automated Acquisition and Control) interface device from CYBORG INC.

(5) PCS327 Analog Control Simulator from FEEDBACK CORP.

(6) Dual Power Supply from HEWLITT-PACKARD INC.

(7) GRAPPLER printer interface buffer

(8) 80 COLUMN CARD with 64k extended memory

The additional power supply is necessary because of the additional circuitry shown in Figure 4. All of the remaining equipment listed is discussed later where pertinent to features, limitations and potential uses of the system. Figure 5 shows all of the connections required for normal use.

Different connections are required if the user wishes to examine the process while under On-Off control. The connections required for On-Off control are given in Appendix B. Referring to Fig. 5, note that the connection at point A is dependent on whether the process is under analog or digital control. If the process is under analog control, the output from the analog control simulator should be connected to point A. If the process is under digital control then the output from the third circuit shown in Fig. 4 should be connected to point A. Exclusive of that connection, the remaining connections should not be changed unless the process is to be examined under On-Off control.

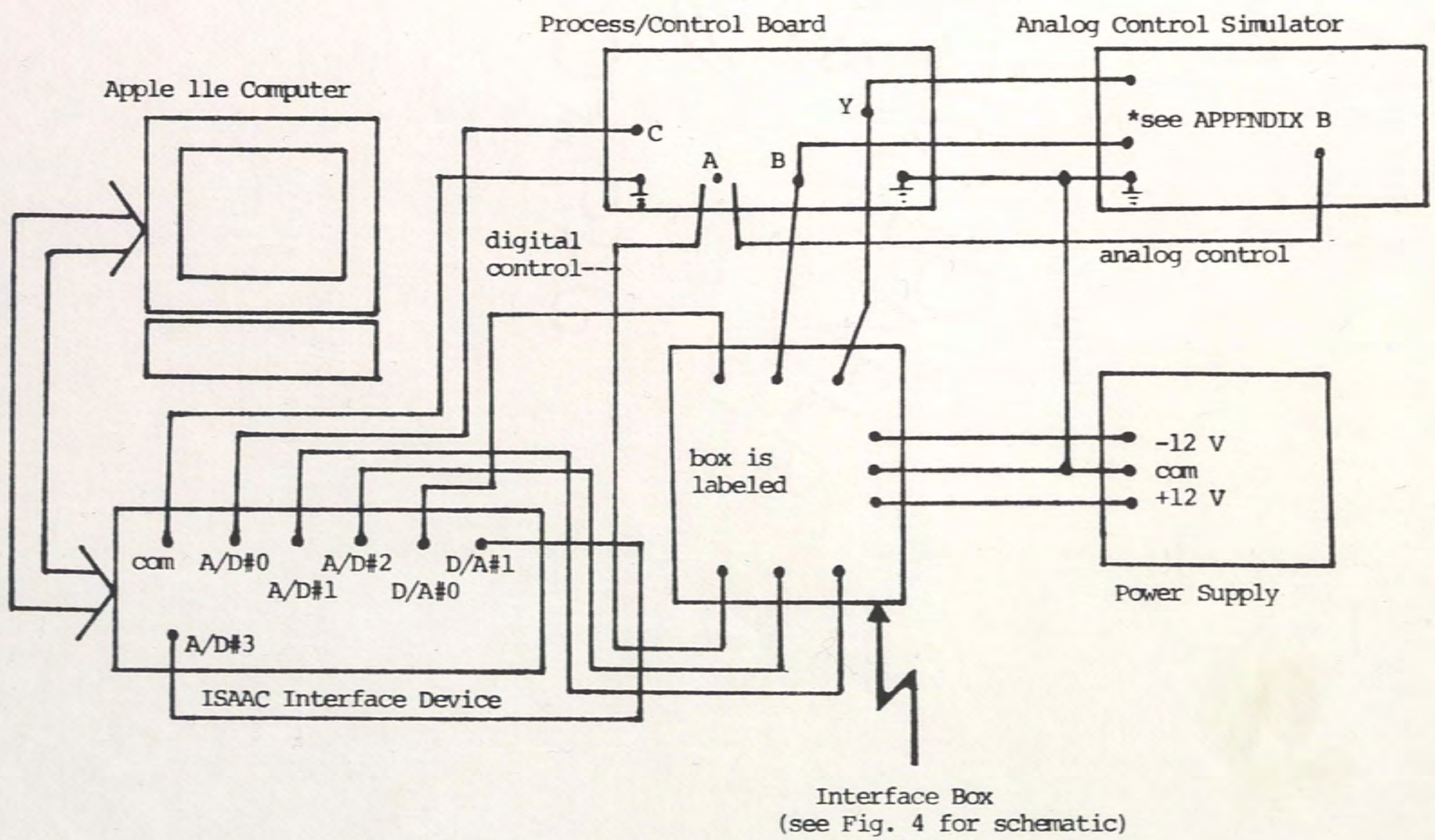


FIGURE 5 - Required Connections for Operation

CHAPTER 2 - RELATIONSHIPS BETWEEN PHYSICAL PROCESS AND CLASSICAL CONTROL THEORY

SECTION 1 - Process Variables

Figure 6 shows the process in block diagram form. For this particular process there is one load variable (blower opening), one manipulated variable (wattage output from voltage controlled power supply), and one controlled variable (process temperature).

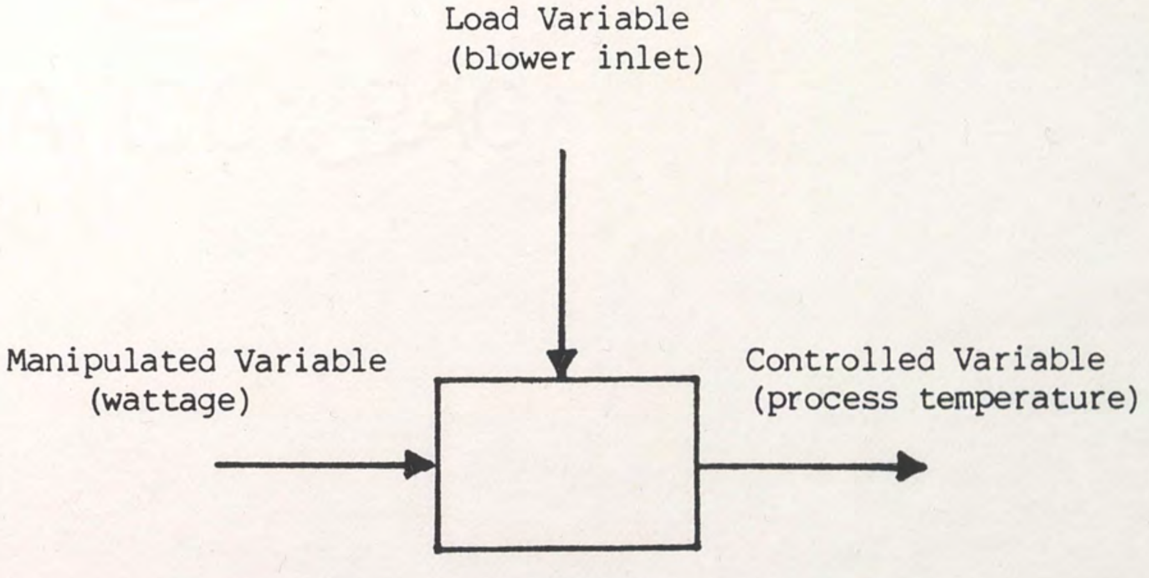
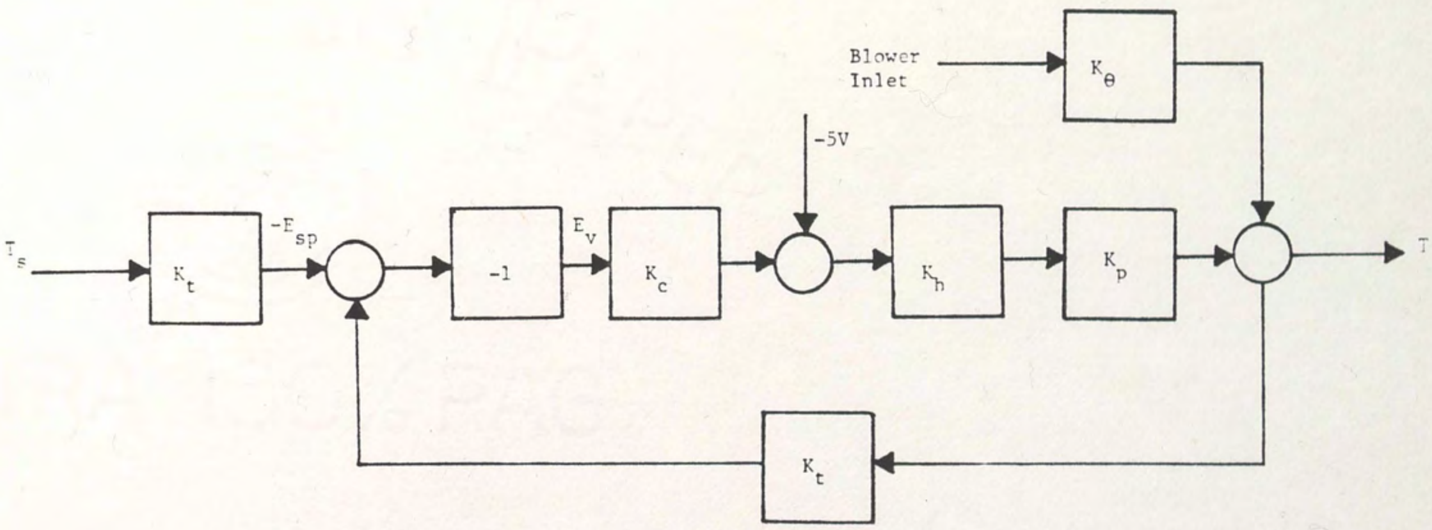


FIGURE 6 - Block Diagram of Process

Figure 7 shows the process and control system in block diagram form. This illustration depicts the control system as it is on the board. Utilizing simplification techniques, this block diagram can be redrawn as shown in Figure 8. The system being discussed in this report allows the user to "replace" the type of controller being used and then examine the process variables while under these different controllers.

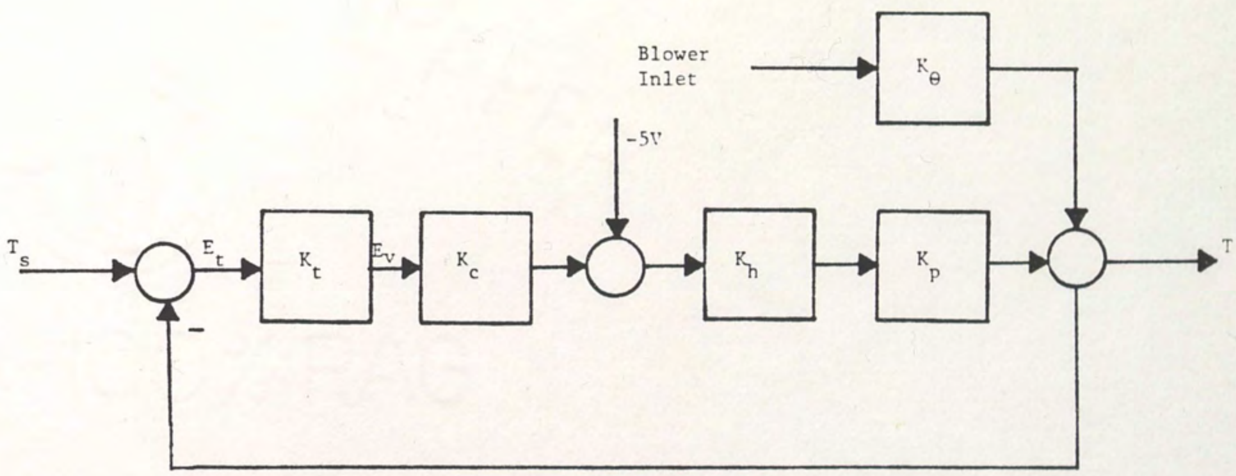
SECTION 2 - Design Conditions

Design conditions are considered, as the name implies, to be the conditions at which the process was designed to operate, i.e., maintain the temperature with zero error. There are two main problems with the present configuration. They are (1) design conditions are to simply blow air through the tube and (2) nonlinear control. Design conditions being to blow air through the tube is considered a problem because the normal operating region is higher than room temperature. The second main problem is related to design conditions being room temperature. Anytime the process temperature exceeds the set point, then the error goes negative (error = set point temperature - process temperature). Referencing Fig. 8, note that -5 volts is being added in after the controller (Kc). This means that whenever the process temperature exceeds the set point temperature the voltage into the final control element is less than - 5 volts. Referencing Fig. 3a, note that this corresponds to zero watts output to the process. The final result of this discussion is that anytime the process temperature exceeds the set point temperature then the output of the final control element (power supply) goes to zero. Note that the controller is not controlling the process, it is simply shutting the power supply off.



LEGEND: Ts - Set point temperature
T - Process temperature
Ev - Error voltage
Esp - Set point voltage
Et - Process voltage
Kc - Controller gain
Kh - Heater gain
Kp - Process gain
Kt - Transmitter gain
Kθ - Blower inlet gain

FIGURE 7 - Block Diagram of Process/Controller



LEGEND:
T - Process temperature
Ts - Set point temperature
Et - Error in temperature
Ev - Error voltage
Kt - Transmitter gain
Kc - Controller gain
Kh - Heater gain
Kp - Process gain
Ko - Blower inlet gain

FIGURE 8 - Simplified Block Diagram of Process/Controller

The root of both of these problems is the - 5 volts being added in before the final control element.

If + 5 volts is added in to counter the - 5 volts added by the process (this is internal to the process control board and cannot be accessed) then the controller output versus deviation changes from Fig. 9 (present configuration) to Fig. 1Ø (proposed configuration). Note that under this configuration if the process temperature exceeds the set point temperature then the controller output is not automatically less than -5 volts, i.e., the final control element does not simply shut off and we are still controlling the process.

The + 5 volts being added in also impacts design conditions. Note that under digital control the interface device can output analog voltages from - 5 to + 5 volts. Referencing Fig. 3a, note that - 5 volts corresponds to Ø watts and + 5 volts corresponds to 1ØØ watts. This means that the computer is capable of controlling the final control element over its entire operating region. This + 5 volts is added in through the use of the third operational amplifier circuit shown in Fig 4. The problem remains of how to compute the design value to be output to the process (D/A) to maintain the desired set point temperature and blower inlet. This is discussed in Chapter 5.

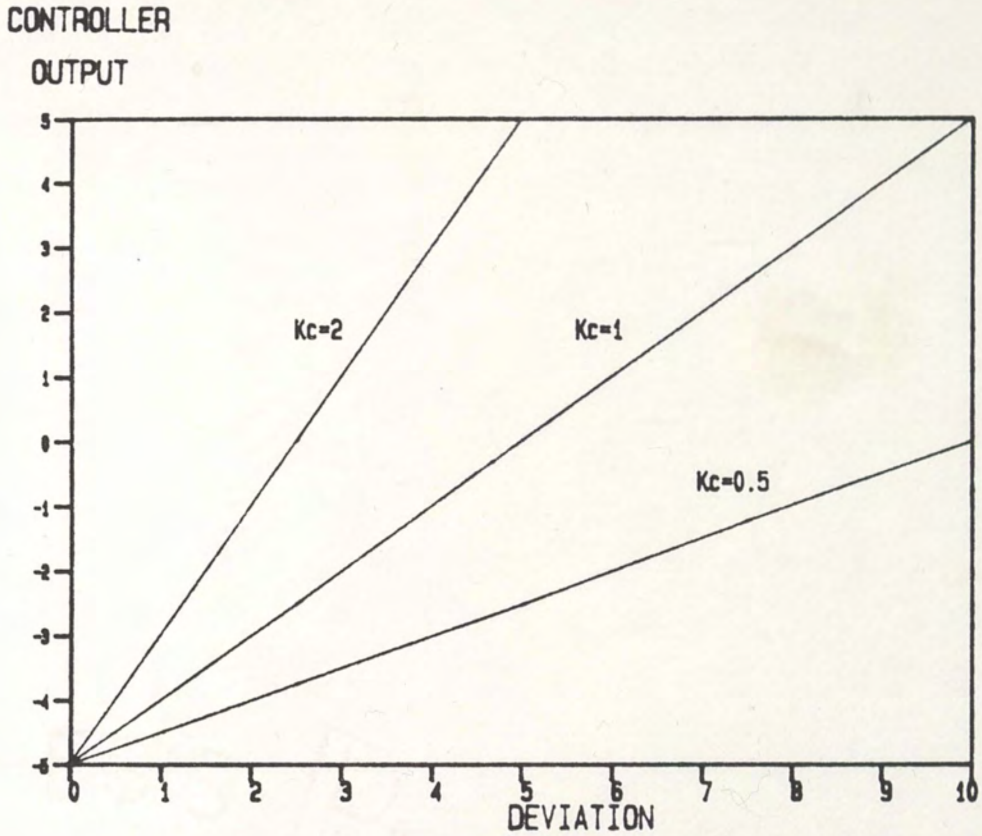


FIGURE 9 - Controller Output versus Deviation

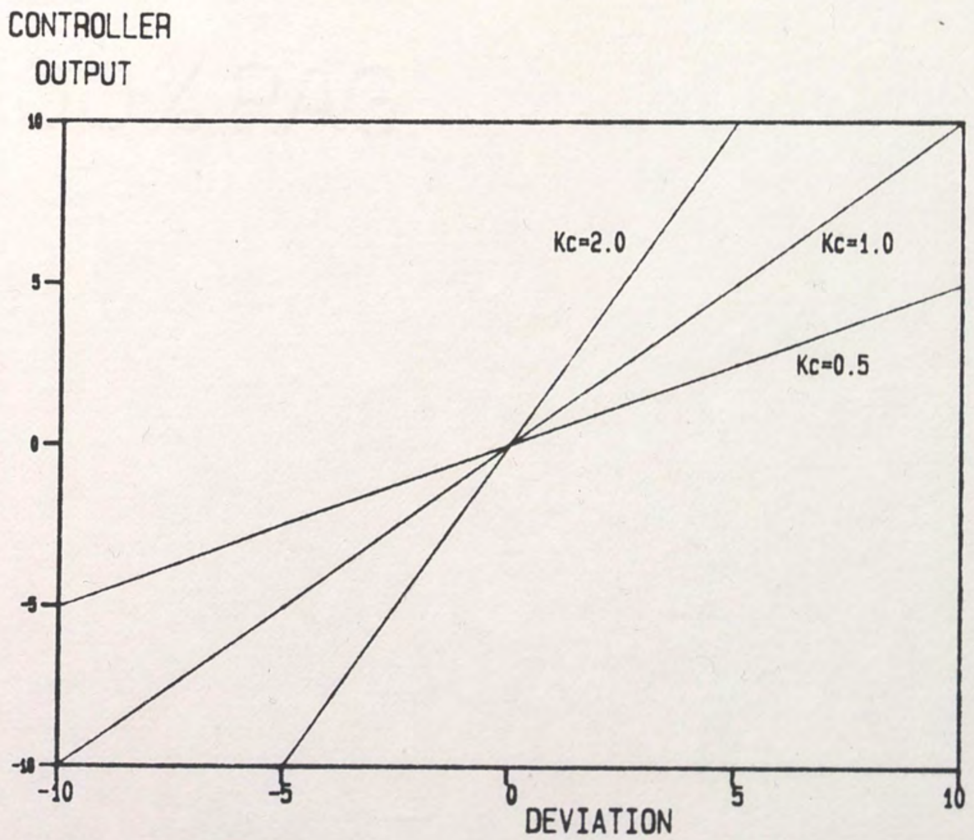


FIGURE 1Ø - DDC Controller Output versus Deviation

CHAPTER 3 - DESCRIPTION OF SOFTWARE

SECTION 1 - Features of Graphs

The user examines the response of the process under different control systems by examining a series of four graphs that display:

(1) Manipulated variable (wattage) vs. time

(2) Controlled variable (process temperature) vs. time

(3) Set point temperature vs. time

(4) Controlled variable and set point temperature vs. time

All graphs are outlined and labeled with the following:

(1) Graph title

(2) Y-axis title

(3) X-axis title

(4) Labeled minimum/maximum values on Y-axis

(5) linear tic marks, both axis

(6) Labeled X-axis *

* applies only to the category graphs with labeled time base

The graphs are also linear. This linearity is a direct result of limiting the temperature swing to the linear region of the temperature transmitters output, 30 to 60 degrees celcius (see Fig. 3a). One limitation that this imposes is that the user cannot examine the process at room temperature (design conditions under analog control).

There are two main categories of graphs (1) real time monitoring and (2) graphs with labeled time base. Real time monitoring is analogous to the electrocardiogram used by doctors, i.e., any change in the process is immediately reflected onto the screen. The graph starts on the left hand side and plots toward the right. When the graph reaches the right-hand margin, the entire graph is shifted to the left to make room i.e. the graph starts to scroll. This scrolling allows for a continuous graph immediately reflecting any response by the process.

Graphs with labeled time base are different in that they "look" at the process for a specified period of time and then plot the system response. With certain restrictions, these graphs are as if the user were examining the response under real time monitoring for a certain period of time and then labeling the time base to reflect how long. The time base is labeled in three places to reflect this period.

The main difference between the two types concerns how they are plotted. Real time monitoring plots a new point on the graph every time a data acquisition is executed. The program flow can be considered a infinite loop of data acquisition/plot. This infinite loop is exited by the user pressing any key on the keyboard. In contrast, graphs with labeled time base are plotted from data stored in a data array. When the system starts to "look" at the process, all of the data are stored in an array. The screen is actually blank while the system is "looking" at the process. The data is stored in an array and plotted later for several reasons. One reason is that we want to be able to construct four graphs altogether. If the data is not stored in an array, the system would only be able to create one graph (the one it is plotting). Another reason why the data is stored in an array is that the number of data acquisitions per graph remains the same for each graph. This means that in order to control how long the system "looks" at the process, the system merely needs to compute the time/sample and then use that value to control how often the system executes a data acquisition.

SECTION 2 - General Features

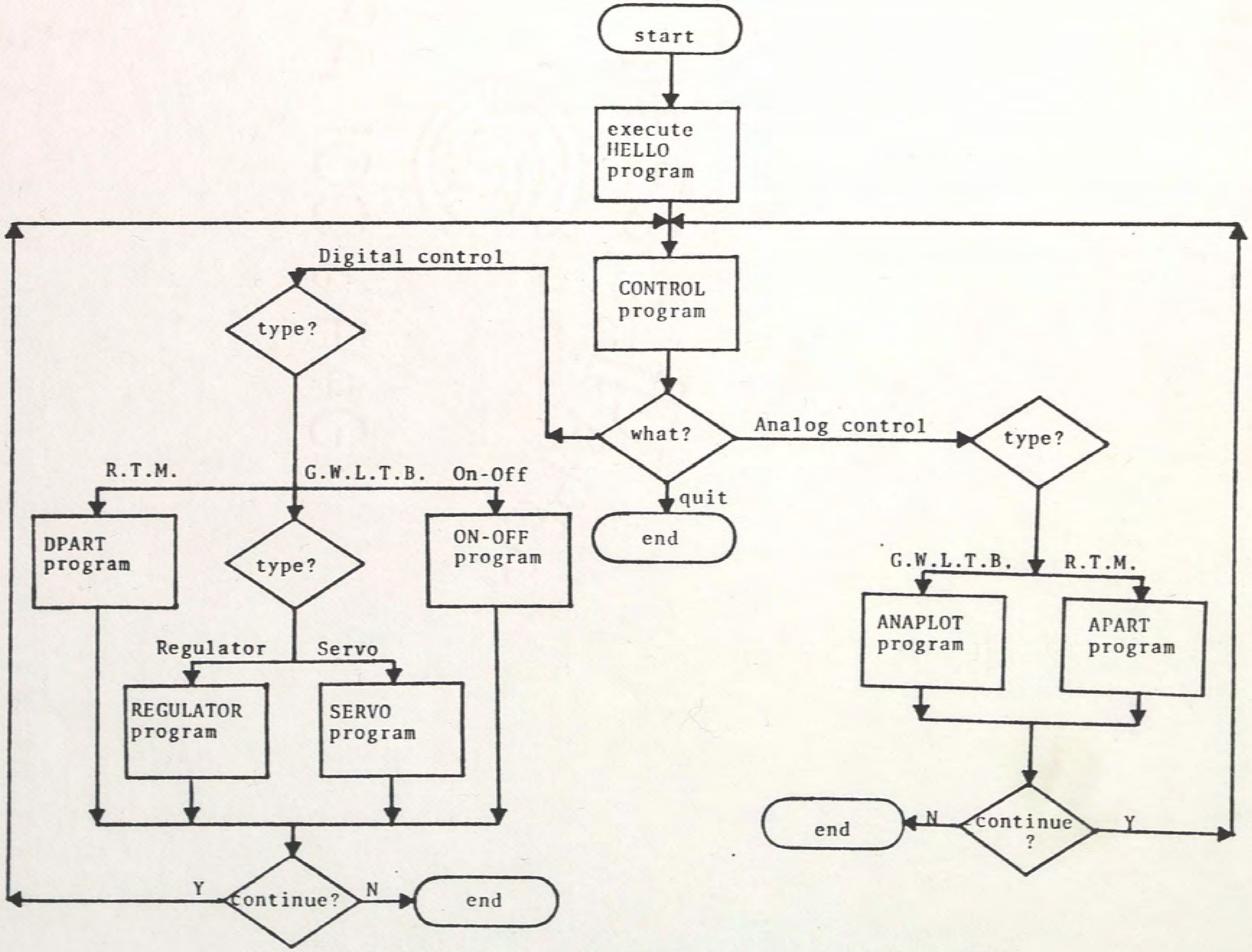
The software developed for this system was designed to be user friendly. This also applies to the start-up of the system. All the user has to do to start the computer is to turn it on. From this point on the programs run based on user inputs. This is possible because the computer always runs the same program to boot up the system (HELLO program). All necessary housekeeping, loading of files, shape tables, etc. is done in the HELLO program and then control is transferred to another program.

All user inputs are checked for potential error. If an error is flagged the system will allow the user to reenter the input. The programs will not continue on until a valid entry is made. This is done to insure that the system cannot be "crashed" due to illegal entries. What is a correct input is dependent upon where the user is in the system. If the input is of the "SELECT ONE OF THE FOLLOWING" category then the user must input one of the options given. The programs will not continue until one of the options shown is picked by the user. On the other hand, there are inputs such as control system parameters which are designed to be variable. These inputs are checked for limits ,i.e., must be between two values. As before, the system will not continue on until an allowable value is entered. As a general rule, inputs of this type are not being checked for stability. This means that the control system parameters themselves are virtually unrestricted. This flexibility allows the user to examine the response of the process to both stable and unstable control systems.

The user also has unlimited time to read system generated messages. All system-generated messages that do not require user input are followed by "STRIKE ANY KEY TO CONTINUE." This is accomplished by checking a certain memory location for a value greater than 127 decimal (greater than 127 means the keyboard has been pressed). This gives the user time to read, interpet, and act upon the message shown.

SECTION 3 - Program Structure

The following section describes the overall structure of the software developed. Figure 11 shows a flowchart of the main sections of software and possible program flow. Figure 11 also shows program name for each associated section. The BASIC code for each section is given in Appendix C.



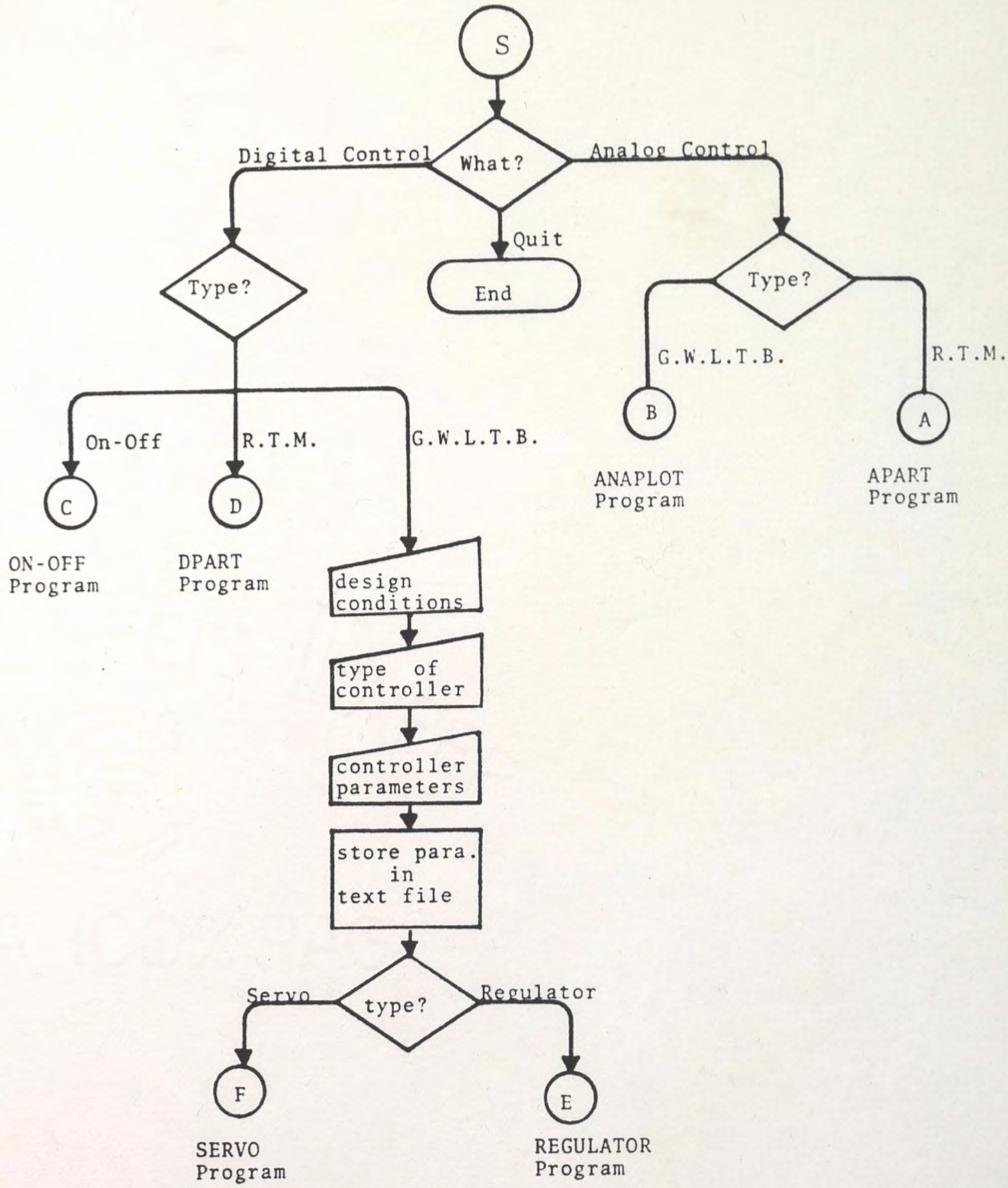
G.W.L.T.B. - graphs with labeled time base
R.T.M. - real time monitoring

FIGURE 11 - Overall Structure of Software

Each main section is a program within itself. The main reason for this is that due to memory considerations, no one program could exceed 16K in length. As a result, software was forced to be "modularized." The only drawback to this approach is that there are now additional time delays associated with disk accesses.

Excluding the HELLO and CONTROL PROGRAM, all of the remaining programs fall within two main categories (1) Analog Control and (2) Digital Control (see Figure 11). The CONTROL PROGRAM is designed to be the main decision-making program. It is within this program that all decisions regarding what program will be run take place. Figure 12 illustrates the flowchart to CONTROL PROGRAM.

There are two main sections within each of the two categories (1) real time monitoring and (2) graphs with labeled time base. In In real time monitoring, the user can examine any process variable as long as desired. The examination is terminated by pressing any key on the keyboard. In addition, the user can easily repeat the examination and change parameters every time. Due to the methodology used in constructing the graphs, the user can examine different process variables quickly. Changing from one graph to another can be accomplished in a matter of seconds.



G.W.L.T.B. - graphs with labeled time base
R.T.M. - real time monitoring

FIGURE 12 - Flowchart to CONTROL Program

Graphs with labeled time base are slightly different in that programs in this category always present to the user a sequence of four graphs. The graphs are always in the following order: (1) Manipulated variable (wattage), (2) Controlled variable (process temperature), (3) Set point temperature and (4) Temperature (process and set point). The user can vary the time base and the associated control system parameters. The user can also obtain hardcopy of the graphs. These programs can be repeated as the user desires, however, it is much more difficult than real time monitoring. Repeating these programs requires two additional disk accesses whereas real time monitoring requires none. The basic idea for having two different sections like this is to use real time monitoring to quickly find a response of interest. Once the user finds a case of interest, graphs with labeled time base can be used to print out a sequence of graphs. These graphs contain all the information needed to examine the response of the process.

CHAPTER 4 - ANALOG CONTROL

SECTION 1 - Introduction

There are two different types of graphs available to the user, real time monitoring and graphs with labeled time base. This corresponds to APART and ANAPLOT, respectively (see Fig. 11). Both programs have one thing in common, that is neither of them actually control the process. While under analog control, the process is controlled by PCS 327 Analog Control Simulator (see Fig. 5). As a result these programs only monitor the process. This makes the analog programs inherently more simple than digital control programs which control the process as well as monitor it.

SECTION 2 - Real Time Monitoring

Once the user enters this section there remains only one question, which process variable to examine? Another way to look at it is which of the four graphs available do you wish to see? The user can examine any of the graphs as many times as desired. The flowchart to APART is shown in Figure 13.

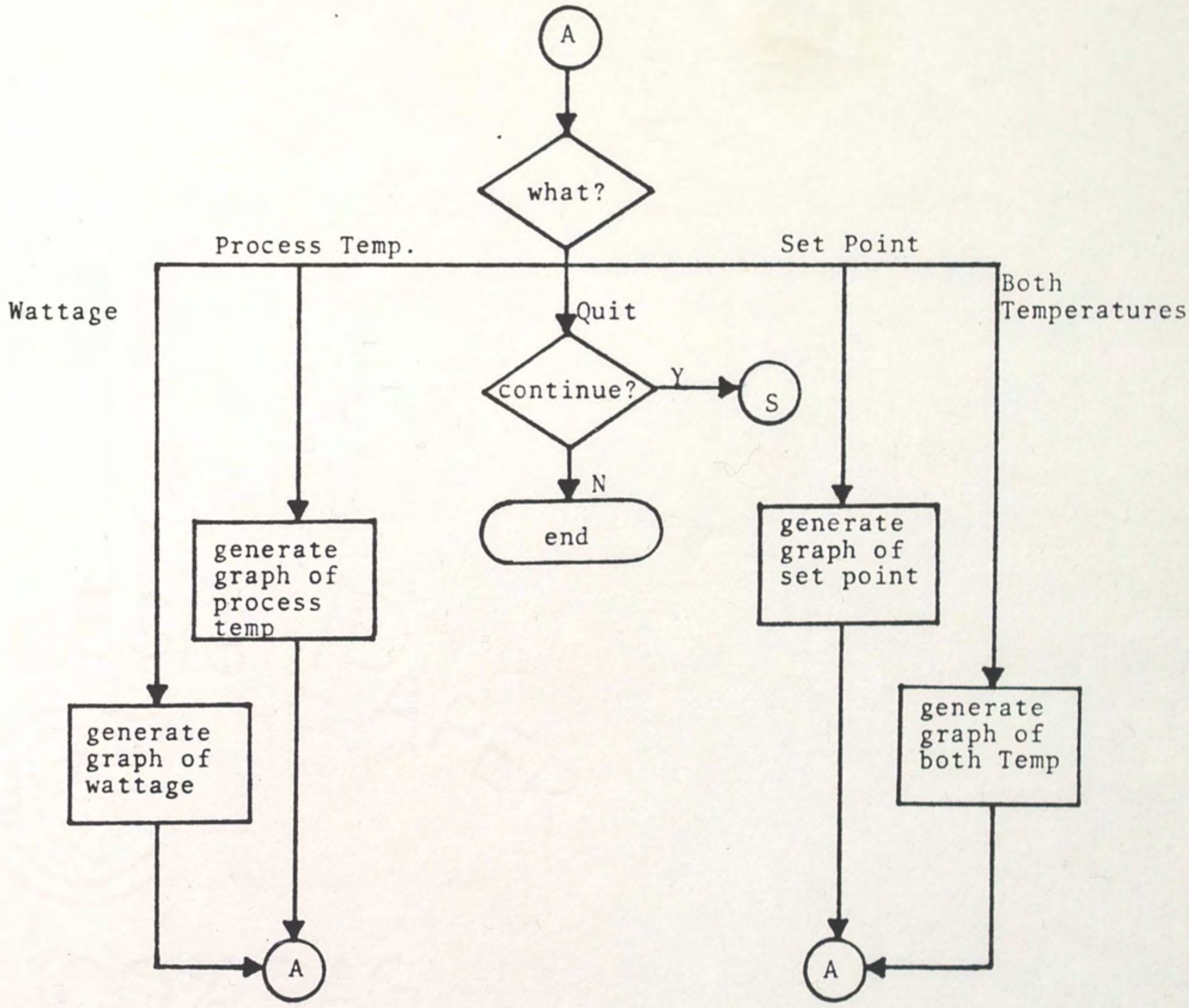


FIGURE 13 - Flowchart to APART program

Typical system-generated messages to the user while going to real time monitoring under analog control are:

```
THE FOLLOWING PROGRAMS ALLOW EXAMINATION OF PROCESS TRAINER
PT326 FROM FEEDBACK CORP.

THE FOLLOWING OPTIONS ARE AVAILABLE AT THIS TIME:

1. ANALOG CONTROL
2. DIGITAL CONTROL
3. QUIT
ENTER CHOICE=        (here the user would enter 1)

WHICH OF THE FOLLOWING DO YOU WISH TO DO?

1. REAL TIME MONITORING
2. GRAPHS WITH LABELED TIME BASE

ENTER CHOICE=        (here the user would enter 1)

THE FOLLOWING OPTIONS ARE AVAILABLE AT THIS TIME:

1. REAL TIME DISPLAY OF MANIPULATED VARIABLE
2. REAL TIME DISPLAY OF CONTROLLED VARIABLE
3. REAL TIME DISPLAY OF TEMP SET POINT
4. REAL TIME DISPLAY OF SET POINT AND PROCESS TEMP
5. QUIT

ENTER CHOICE=
```

Plotting begins once the user selects which graph followed by a message at the bottom of the graph "PRESS ANY KEY TO RETURN TO MENU." This will clear the screen and return the user to the menu shown at the bottom of the above example. Termination of real time monitoring is accomplished by selecting 5 on the example menu. The system will then ask "DO YOU WISH TO CONTINUE ON (Y OR N)?". If the user wants to continue on a Y must be entered. If a Y is entered then the CONTROL PROGRAM is loaded and executed. If a N is entered, the programs terminate. This section allows the user to go from one process variable to another one, repeat examination, etc. very easily.

SECTION 3 - Graphs With Labeled Time Base

As discussed in Chapter 3, this section will output a series of four graphs to the screen every time the program is run. Figure 14 shows the flowchart to ANAPLOT (graphs with labeled time base under analog control).

The time base of these graphs is variable and user input. The system can look at the process anywhere from 1 second to 115 seconds. This is accomplished by controlling the interval time between data acquisitions. This interval time is fairly easy to compute because the number of data points per graph remains the same (260). Once the user inputs the time base, the system then computes how often to do a data acquistion (time per sample). The interface device also has a real time clock that is used to clock events such as time per sample. This clock is accurate to 1 millisecond. As a result, the time base shown on the graphs is very accurate. The system-generated messages while going to graphs with labeled time base under analog control is shown below:

```
THE FOLLOWING OPTIONS ARE AVAILABLE AT THIS TIME:

1. ANALOG CONTROL
2. DIGITAL CONTROL
3. QUIT

ENTER CHOICE=          (the user would enter 1)

WHICH OF THE FOLLOWING DO YOU WISH TO DO?

1. REAL TIME MONITORING
2. GRAPHS WITH LABELED TIME BASE

ENTER CHOICE=         (the user would enter 2)

DO YOU WANT HARDCOPY OF THE GRAPHS (Y or N)?
     (the user must enter Y or N)

NOTE: MIN LENGTH OF SAMPLE IS 1 SECOND
      MAX LENGTH OF SMAPLE IS 115 SECONDS

HOW LONG OF A SAMPLE DO YOU WANT (SECONDS)=
   (must be between 1 and 115)

AT THE BEEP DATA ACQUISITION BEGINS
YOU HAVE (input) SECONDS OF ACQUISITION TIME.
```

At this point the system is generating the four graphs for the user to see. Figure 15 shows typical output from ANAPLOT. Note that these graphs will not be output to the printer unless the user enters a Y(es) when asked if a hardcopy is wanted (see above example of system messages).

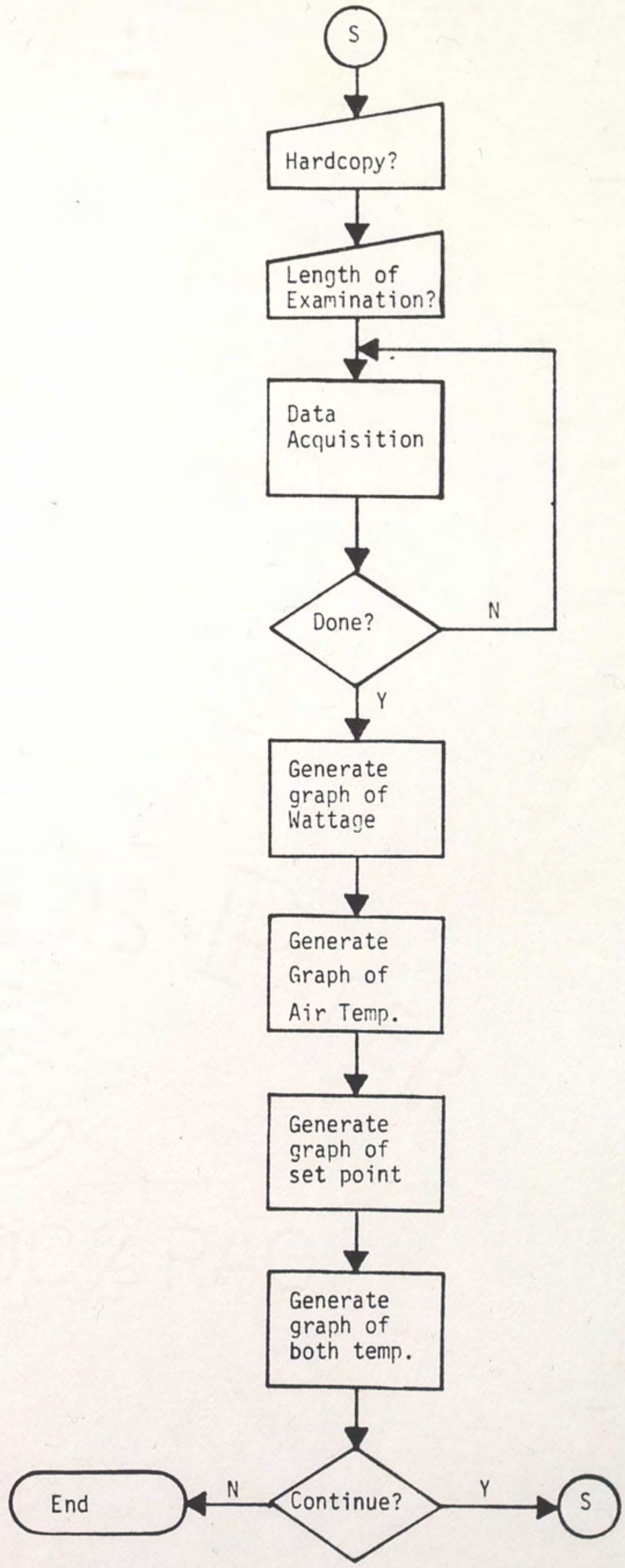


FIGURE 14 - Flowchart to ANAPLOT Program

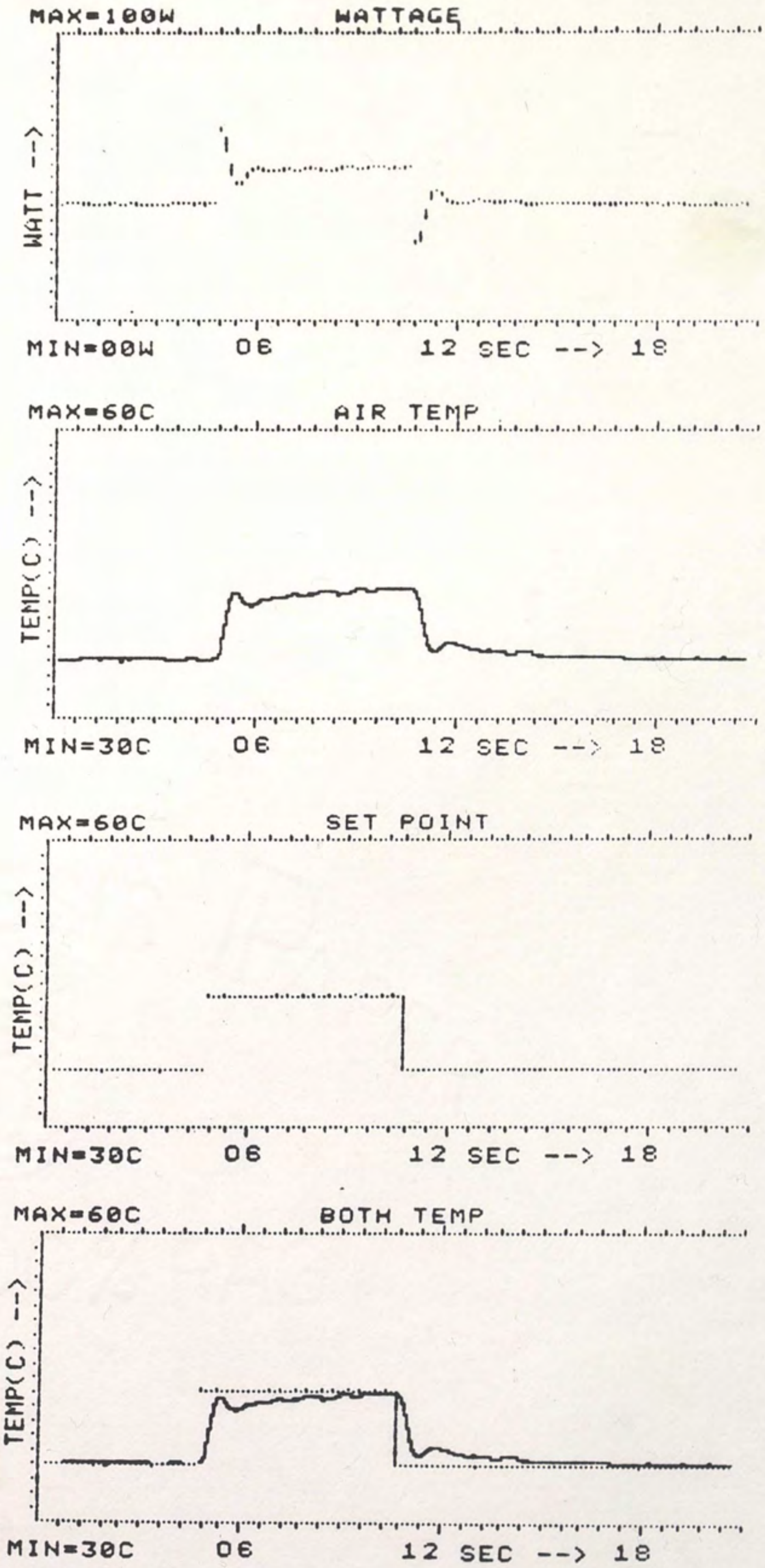


FIGURE 15 - Typical output from ANAPLOT Program

CHAPTER 5 - DIGITAL CONTROL

SECTION 1 - Introduction

This chapter discusses the system under digital control. This category of software is basically an identical counterpart to the category of analog control, subject to limitations. As before, there are two different types of graphs available to the user: (1) real time monitoring and (2) graphs with labeled time base. The name of the program that does real time monitoring is DPART. There are two different programs within the section of graphs with labeled time base. They are (1) SERVO and (2) REGULATOR (see Fig. 10). As before, control is transferred to these programs by the CONTROL PROGRAM.

There is one major difference between digital control and its counterpart, analog control. Digital control requires the software to control the process as well as maintain it. Analog control software only has to monitor the process. The task of exerting control over the process requires additional code. Unfortunately, there is not enough memory to ask the questions necessary for control, the code to exert control, and storage for a data array (2x260). To overcome this problem the code to ask the questions necessary for control has been placed in the CONTROL PROGRAM. The control parameters are then transferred, through a text file, along with control to the program. Note that this applies only to graphs with labeled time base under digital control. Real time monitoring does not require a data array, therefore there is no memory conflict.

As before, the user still has the ability to examine the process while under different types of control. The user can still examine all four of the following types of control systems (1) Proportional control, (2) Proportional-Integral control, (3) Proportional-Integral-Derivative control and (4) On-Off control. This includes both regulator and servo responses. Since the process is under the computer's control, a set point change (servo response) has to be entered through the keyboard. To implement a set point change while under digital (the computer) control requires different code than for regulator response (no set point change). This is the main reason for the two different programs under graphs with labeled time base, SERVO and REGULATOR (see Fig. 11).

A comparison of digital control versus analog control is discussed in section 8. One of the main benefits derived from digital control is the ability to have variable design conditions (recall that the computer is capable of controlling the final control element over its entire operating region). Design conditions are discussed in the next section.

SECTION 2 - Design Conditions

While under digital control, the user can choose design conditions within limitations. This is possible because the computer is capable of controlling the amount of wattage output to the process.

Recall that the circuitry shown in Figure 4 adds in +5 volts to cancel out the -5 volts added in on the control board. This causes the relationship depicted in Figure 1Ø to apply. Under this configuration the computer can control the wattage output to the process. The problem remains of how to compute the design wattage to be output to the process for a given desired set point temperature and blower inlet (load variable). This is accomplished using a regression equation. A series of data points (see Appendix D) was taken and used as input to a software package. This software package then computed the coefficients for a nine-term polynomial that is a function of two independent variables and one dependent variable. When the user enters desired set point temperature (first independent variable) and then the blower inlet (second independent variable), the design wattage is computed, i.e., Wattage=Function of (Ts, Theta). The regression equation and its associated coefficients are shown below.

```
Wattage = A1+(A2*X)+(A3*X*X)+(A4*Y)+(A5*Y*Y)+(A6*X*Y)+(A7*X*X*Y)
          +(A8*X*Y*Y)+(A9*X*X*Y*Y)

      WHERE:      X = Set point temperature
                  Y = Blower inlet (degrees)
                 A1 =-65.176394
                 A2 = 2.46541807
                 A3 =-0.0167582955
                 A4 =-1.80406
                 A5 = 0.0199352325
                 A6 = 0.110026458
                 A7 =-9.88198962 E-04
                 A8 =-1.18368066 E-03
                 A9 = 1.5608222 E-05
```

Once the system has the design wattage to be output to the process, it need only be output once (D/A). The interface device will maintain the analog voltage at that value.

The system incorporates error checking here in the form of determining whether or not it is possible to maintain the process at a given set point temperature and blower inlet. This is done immediately after computing the design wattage to be output to the process. The maximum heater output is 1ØØ watts. If the design wattage computed by the regression equation is over 96 watts, then the user has entered, for all practical purposes, unmaintainable conditions. If this happens, the system will display the following message:

> "YOU HAVE CHOSEN DESIGN CONDITIONS WHICH ARE NOT PHYSICALLY REALIZABLE BECAUSE THE MAXIMUM HEATER OUTPUT IS 1ØØ WATTS."

The system then allows for the user to reenter design conditions. The system will not continue on until a set of maintainable conditions are entered by the user. Figure 16 shows an illustration of the wattage required to maintain a given set point temperature, with a fixed blower inlet. The blower inlet was stepped from 1Ø to 9Ø degrees, in increments of 1Ø degrees.

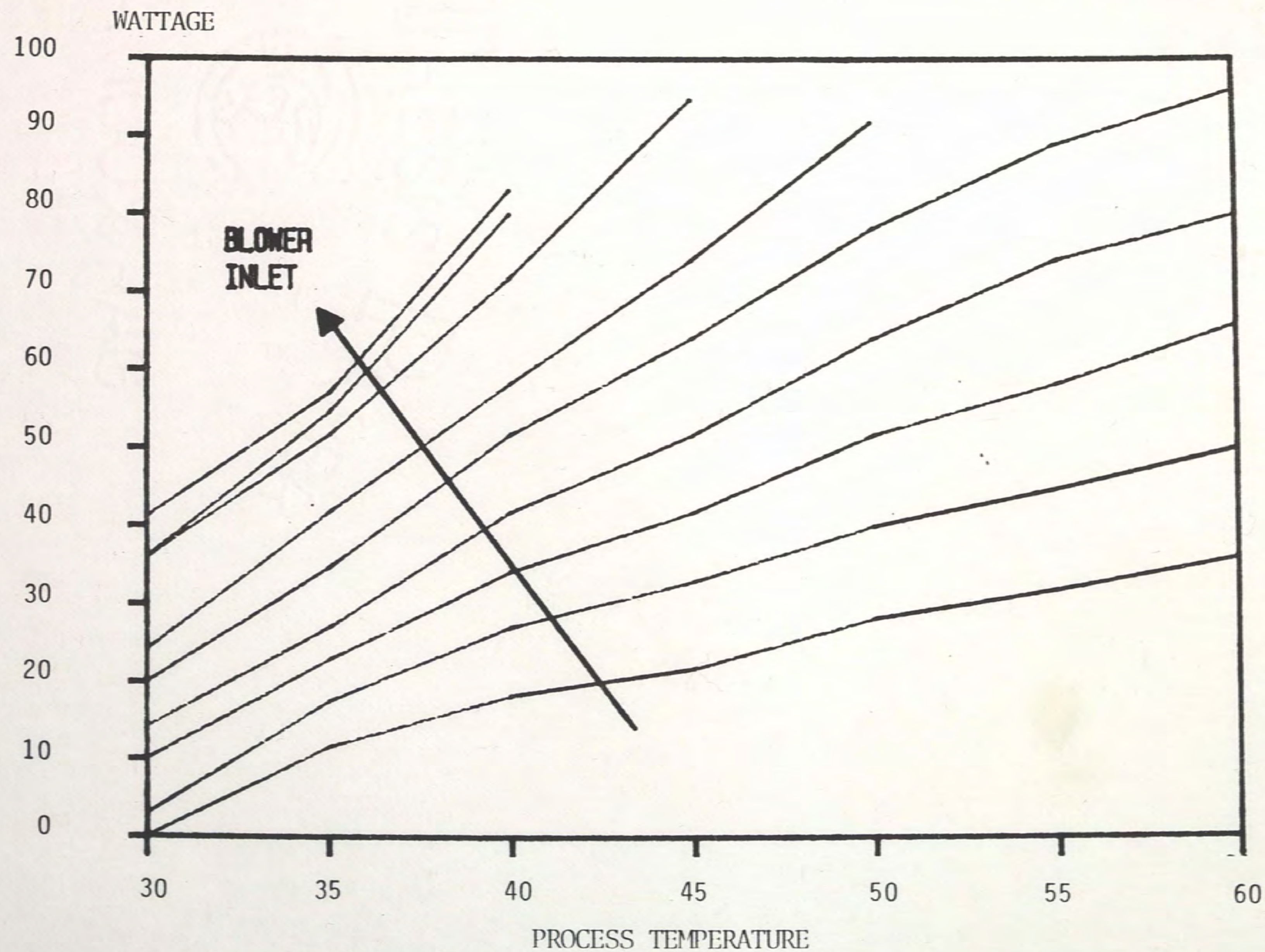


FIGURE 16 - Wattage Requirements (Fixed Blower Inlet)

SECTION 3 - Unmaintainable Conditions (Regulator Action)

Unmaintainable conditions occur whenever the process requires over 100 watts to maintain the set point temperature. This can occur if the process requires a large amount of wattage to maintain the temperature and then there is a large deviation in the load variable, i.e., the user opens the blower inlet. Whenever this occurs, the discrete value representing the voltage exceeds allowable limits. These limits are determined by the fact that the interface device performs conversions (D/A, A/D) using 12 bits. This means that the largest value that can be read by the interface is 4096. Whenever the computer outputs a value representing over 100 watts (greater than 4096), the interface will only accept the lower 12 bits. As a result, the most significant bit (MSB) is lost. This causes the actual voltage output to the power supply to "rollover." The decision was made not to correct for this because the controlling of the process is a critically timed loop and the test added too much time to the loop. Recall that the programs are in interpeter BASIC. Note that this problem can be avoided entirely by lowering the set point temperature. Figure 17 is an example of "rollover" occurring.

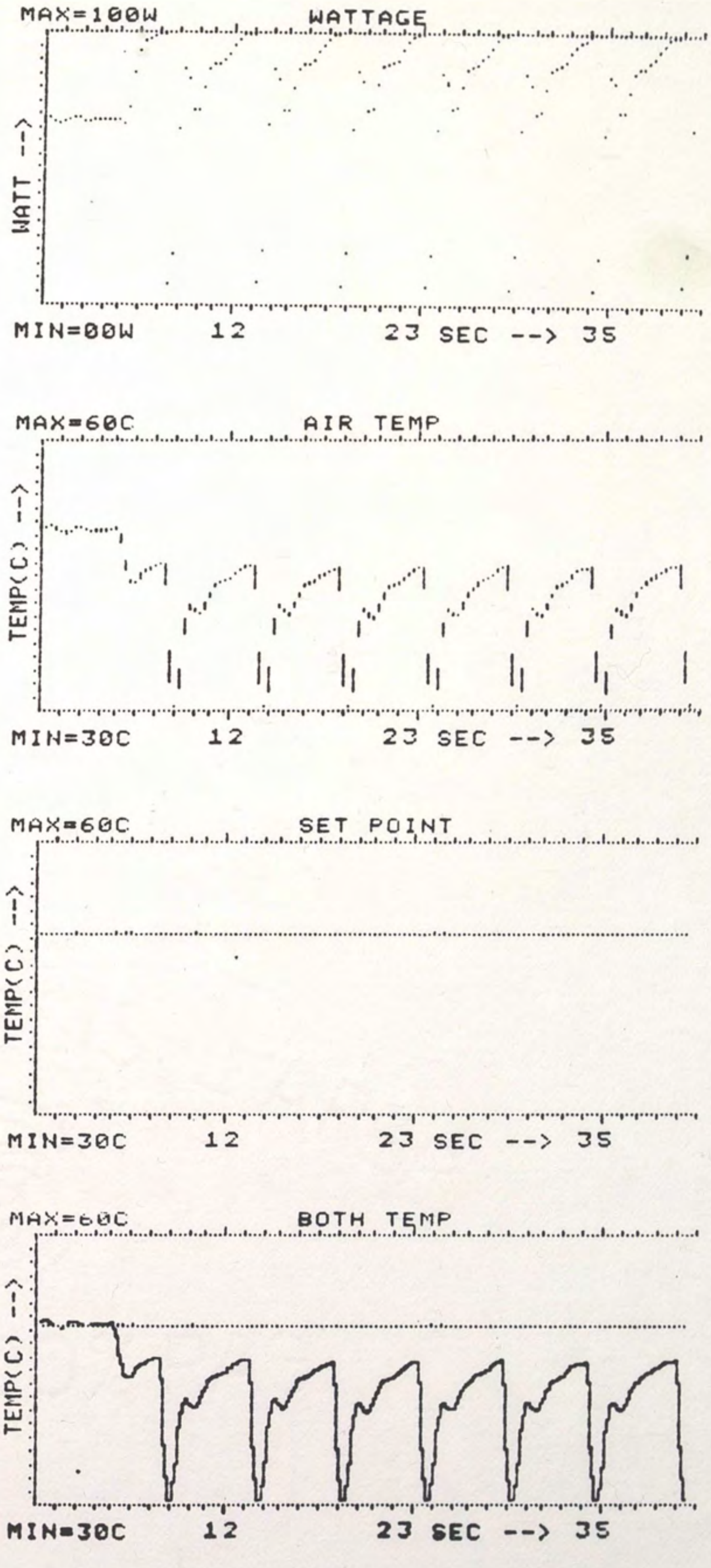


Figure 17 - Example of "Rollover"

SECTION 4 - Implementation of Integral Action

Once the A/D conversion has been completed, the computer has an integer value representing the state of a process variable. The computer is now free to manipulate this number in any way necessary to control the process. Each different control algorithm is implemented through a series of equations. There are two different algorithms to digitally implement (write code for) the integral term in P-I and P-I-D control. They are the position algorithm and the velocity algorithm. The position algorithm computes the actual controller output for the sampling instant. It computes the controller output based on the current position. The velocity algorithm computes an incremental term to be added to the last controller output. This sum is the actual controller output for the sampling instant. Both algorithms are equivalent in terms of what they accomplish, i.e., both implement the integral term in P-I and P-I-D control. The velocity algorithm, however, also provides protection against reset windup (error caused by starting and stopping the system). The integral term is implemented using the velocity algorithm.

SECTION 5 - Real Time Monitoring

Real time monitoring under digital control allows the user to examine the response of the process under the following control schemes: (1) Proportional control, (2) Proportional-Integral control, and (3) Proportional-Integral-Drivative control. On-Off control is considered to be a special case and is a separate program in itself. On-Off control is discussed further in section 7. As mentioned, real time monitoring under digital control is accomplished by the DPART program. Figure 21 shows the flowchart to DPART.

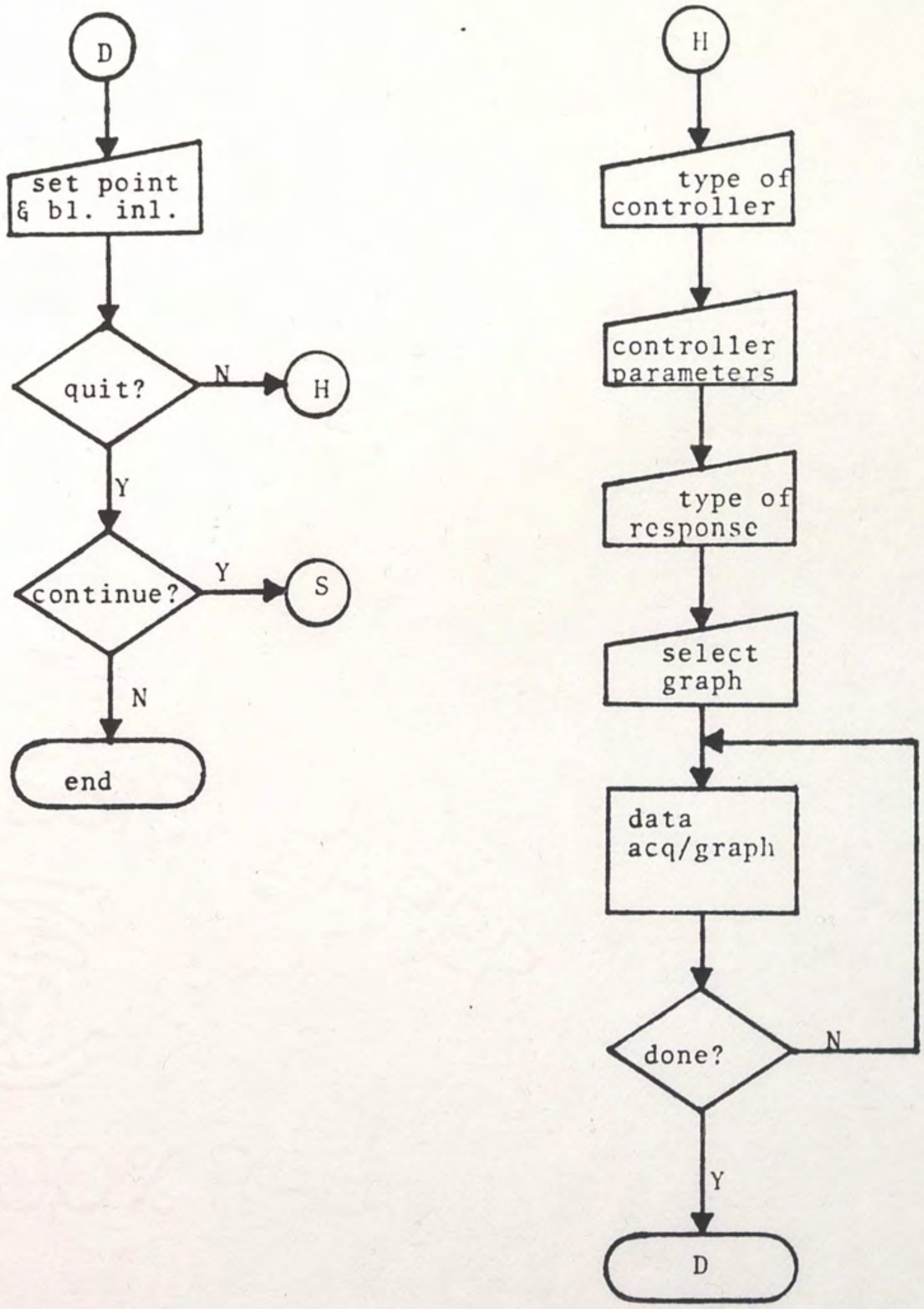


FIGURE 18 - Flowchart to DPART program

The software is designed to be user friendly. Shown below is typical system generated messages to DPART.

```
THE FOLLOWING OPTIONS ARE AVAILABLE AT THIS TIME:

1. ANALOG CONTROL
2. DIGITAL CONTROL
3. QUIT

ENTER CHOICE=          (here user would enter 2)

WHICH OF THE FOLLOWING DO YOU WISH TO EXAMINE?

1. REAL TIME MONITORING
2. GRAPHS WITH LABELED TIM BASE
3. ON-OFF CONTROL

ENTER CHOICE=          (here the user would enter 1)

YOU HAVE THE FOLLOWING OPTIONS CONCERNING CONTROL ALGORITHMS
AT THIS POINT:

1. PROPORTIONAL CONTROL
2. PROPORTIONAL-INTEGRAL CONTROL
3. PROPORTIONAL-INTEGRAL-DERIVATIVE CONTROL
4. QUIT

ENTER YOUR CHOICE NOW=       (the user must select one)

AT THIS POINT THE SYSTEM WILL QUERY YOU ON RELATED CONTROL
SYSTEM PRAMETERS. WHAT USER WILL BE ASKED IS DEPENDENT ON
CONTROL SYSTEM SELECTED.

AT THIS TIME YOU HAVE THE FOLLOWING OPTIONS:

1. REGULATOR CONTROL (RESPONSE TO LOAD VARIABLE CHANGE)
2. SERVO CONTROL (RESPONSE TO SET POINT CHANGE)

ENTER CHOICE=            (user must select one)
```

If the user had asked for servo control the system will prompt "ENTER SET POINT TEMP="

```
YOU HAVE THE FOLLOWING OPTIONS AT THIS POINT:

1. REAL TIME DISPLAY OF WATTAGE
2. REAL TIME DISPLAY OF AIR TEMPERATURE
3. REAL TIME DISPLAY OF SET POINT TEMPERATURE
4. REAL TIME DISPLAY OF BOTH TEMPERATURES

NOTE: YOU ARE NOW RUNNING THE CONTROL ALGORITHM
      THAT YOU JUST SELECTED
```

At this point the system will begin plotting the response of the process. The following control parameters are selected by the user: (1) gain, (2) reset time, (3) derivative time and (4) sample time. Sample time is the time between samples and the user can change it everytime he selects a control algorithm. The system will only ask you for necessary parameters, i.e., reset time cannot be entered under proportional control.

The system does not incorporate stability checking on the control system parameters. As a result, the user can examine the response of the process for both stable and unstable control systems. The sample time is checked to make sure that it is greater than 1ØØ milliseconds (all control algorithms take at least this long to execute). Note that sample time corresponds to the time between the beginning of the control algorithm. It does not correspond to how long the algorithm takes, that is a function of complexity, i.e., P-I-D control takes more time to execute than Proportional control. As a result, if the user enters a sample time that is greater than 1ØØ milliseconds, but less than the time for the control algorithm to execute then the system is "free running," i.e., simply going as fast as possible.

SECTION 6 - Graphs with Labeled Time Base

Under digital control, graphs with labeled time base outputs a sequence of four graphs similar to those previously discussed. There is one major difference between this category and its analog counterpart, i.e., the user cannot specify the time base. Under analog control the user could enter the desired time base in seconds. Under digital control the user can enter a sample rate. The sample rate influences the actual time base of the graph. Once the user has entered the control parameters, the system starts doing data acquisition and/or controlling the process. This loop is executed until the data array is full (260 data acquisitions). The time base is then computed using a counter to determine how many times the control portion was executed. This value is then multiplied by the time required to execute the loop and added to the time required to execute 260 data acquisitions. This is the total time the system has been looking at the process and this value is used to label the time base.

Under digital control the user can still examine the following: (1) Proportional control, (2) Proprotional-Integral control and (3) Proportional-Integral-Derivative control. The user cannot examine On-Off control in this category. As before, On-Off control is considered a special case. The user can examine both regulator and servo response under any of the mentioned control systems. From Fig. 11, note that regulator and servo response correspond to REGULATOR and SERVO programs, respectively. Figure 19 shows the flowchart to the REGULATOR program. Figure 20 shows the flowchart to the SERVO program. Note that the main difference between the two programs is that SERVO is looking at the keyboard, checking to see if the user has pressed it signalling to change the set point temperature. Recall from previous discussion that both of these programs have all control system parameters passed to them through a text file. This was necessary because of memory requirements (see Chapter 3, Section 3).

The system-generated messages to run these programs are almost exclusively within the CONTROL program. All control system parameters are gathered there and passed to the appropriate program.

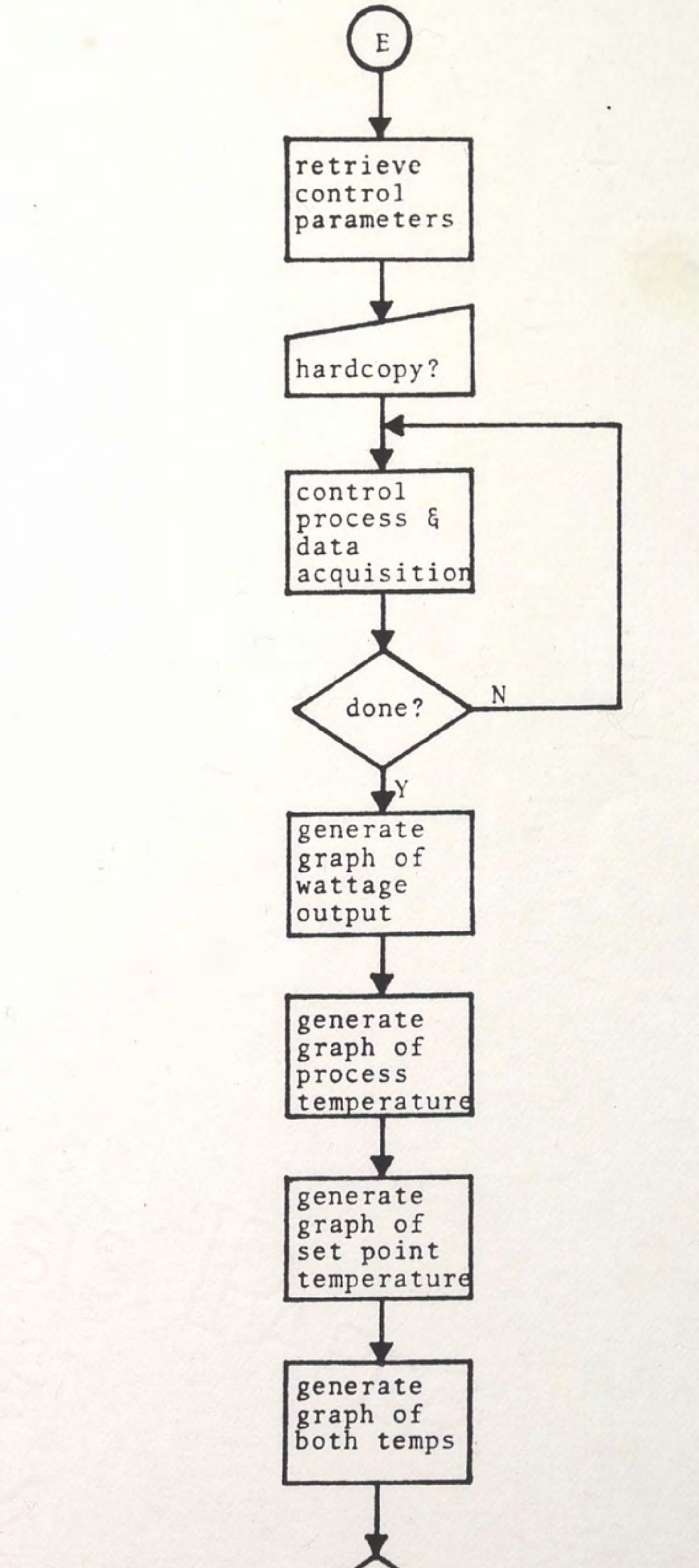


FIGURE 19 - Flowchart to REGULATOR Program

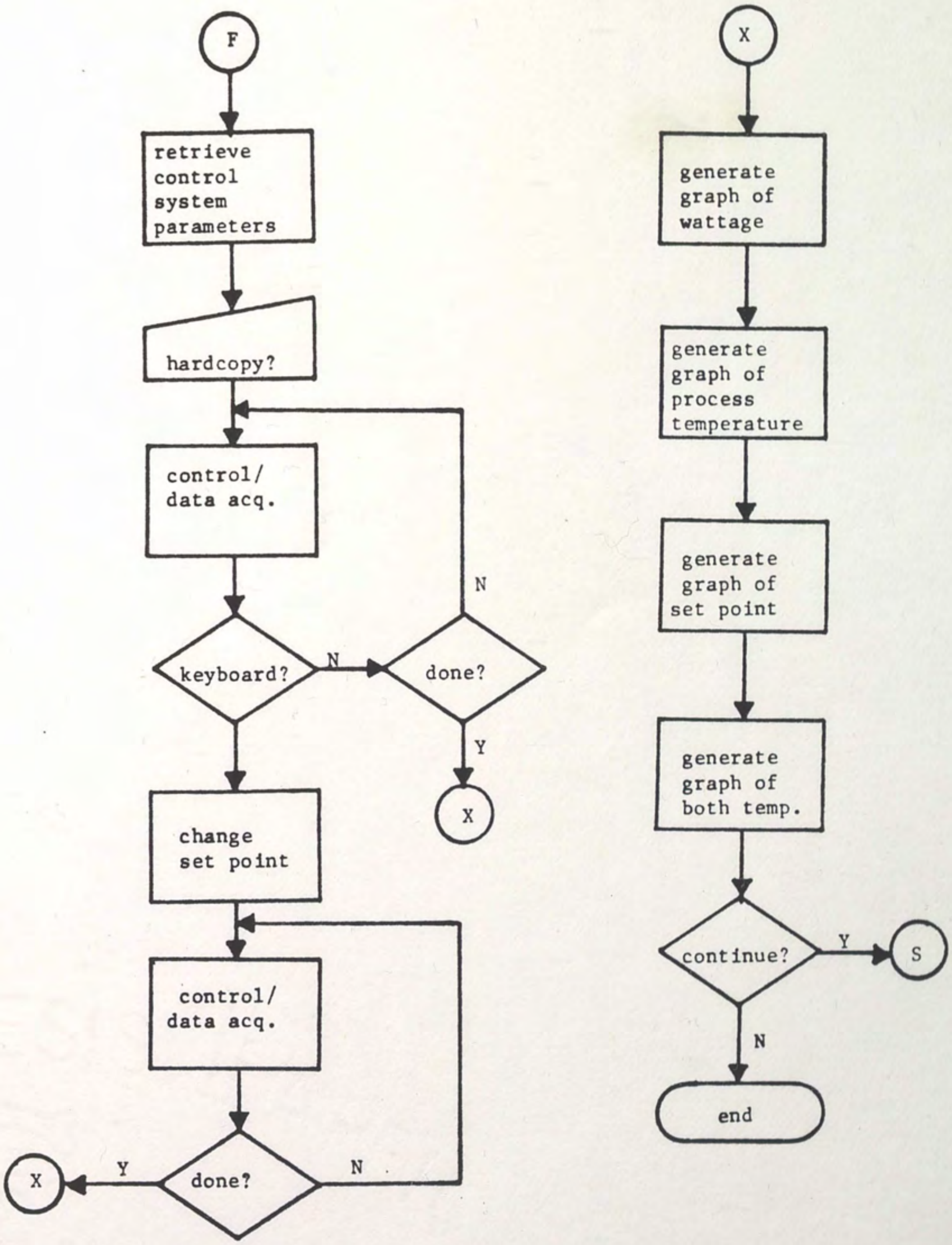


FIGURE 2Ø - Flowchart to SERVO program

SECTION 7 - On-Off Control

This type of control is considered separately. This is the only nonlinear control system under digital control. This category of control is subject to a serious restriction, namely the execution speed of the computer combined with the fact that the programs are written in interpeter BASIC. On-Off control requires fast execution time to be effective. As presently configured, the system has no control over the "dead range" because of this restriction. Figure 21 shows typical On-Off control under the present configuration. Since the computer has negligble control over the "dead range," this type of control is not available under graphs with labeled time base. As Figure 21 shows, this program does effectively demonstrate the basic principle of On-Off control.

SECTION 8 - Comparison to Analog Control

The following section is a comparison of using digital versus analog means to control a physical process. There are three main implications that arise from the implementation of digital control. The first is the fact that the control system is now linearized, allowing examination by classical control theory. The second implication is that under digital control design conditions are variable, within restrictions. The final implication is that the sampling interval is variable under digital control.

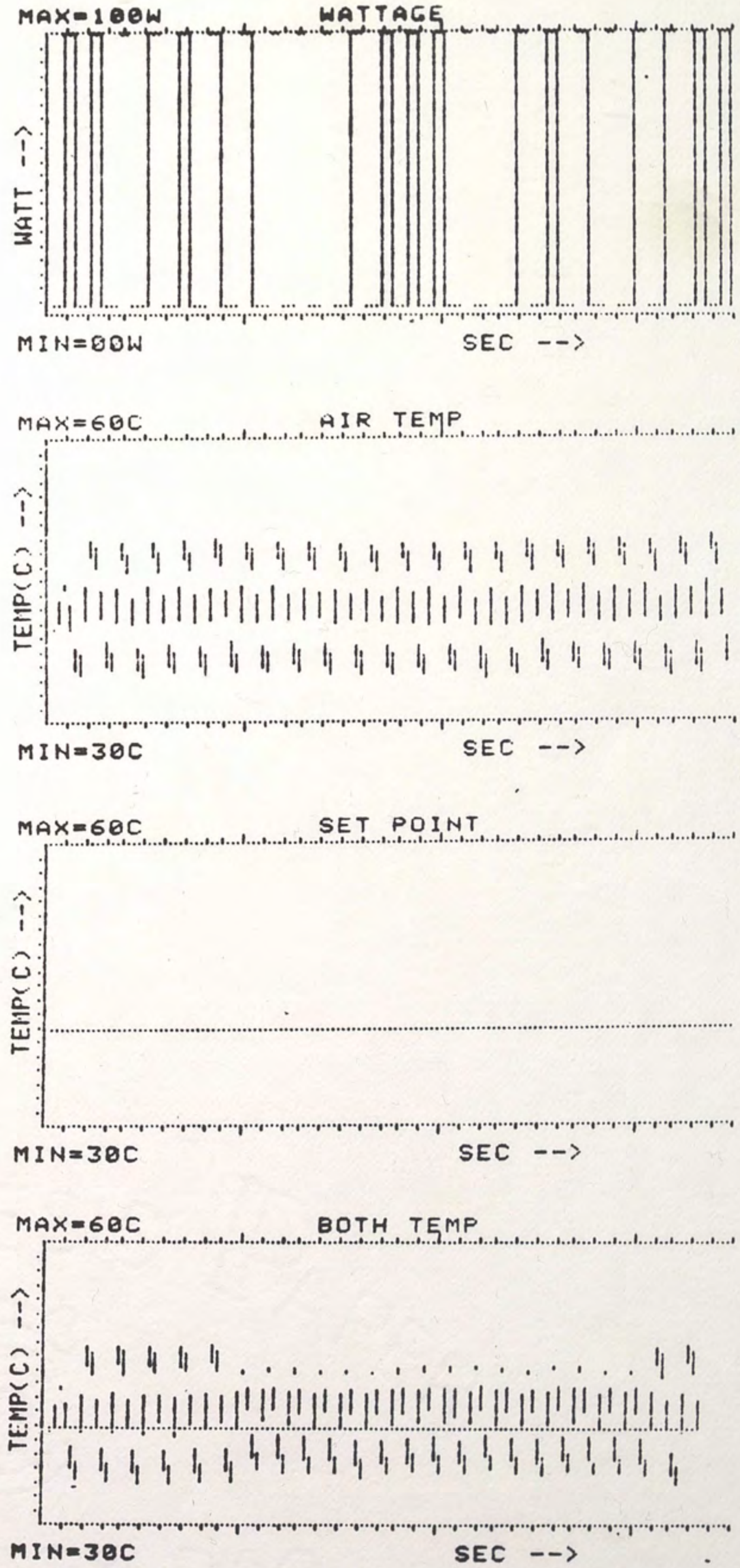


FIGURE 21 - Example of Typical On-Off Control

The fact that the system is now a linearized system that classical control theory can now be applied. At first glance this may not seem important, however, this greatly simpifies analysis of the control system.

The fact that design conditions are now variable gives rise to superior control in some cases. As an example suppose the user wants to maintain the process at 5ØC under proportional control. This would be initially input as design conditions, thus when the system initially began to control the process there would be no error. Uner analog control this would not be the case. Design conditions are fixed at room temperature, thus when the system begins controlling the process there is an immediate error (5ØC - room temperature). Note that under proportional control with a stable gain this would result in a large offset ,i.e., poor control of the process.

There are several points to made about the fact that the sampling interval can be variable under digital control. The first point is that this implies that the control system is not looking at the process at all times. The process could be considered to be under "open-loop" control while the controller is not looking at it. This could be a serious disadvantage if the process is fast responding and there is a large sampling interval. This also causes certain types of control to be inherently worse under digital control than it would otherwise be for analog control. This is mainly in reference to P-I and P-I-D type of control systems. The integral term introduces a component of the controller output based on time. Under digital control the process is not being looked at all of the time. This reduces the effectiveness of the integral term.

As mentioned, under digital control the controller is not looking at the process at all times. This means that a single computer could be made to control several different physical processes at the same time. This could be done under analog control as well, however, not nearly as easily or as quickly. This is possible because of the flexibilty afforded by using a computer as a controller. This flexibilty also extends to type of control algorithms that could be implemented. In this system the different control systems correspond to different mathematical equations. There are virtually no restrictions on the execution of mathematical equations in a computer, i.e., no restrictions on types of control systems that could be implemented.

CHAPTER 6 - SUMMARY AND CONCLUSIONS

SECTION 1 - Summary

The system that has been the topic of this thesis has demonstated the versatility afforded through the introduction of digital control to the physical process. The system also serves as an excellent teaching aid in the demonstration of linear control theory. In an indirect way the system also serves as an example of the deficiencies of the original design (analog control).

Further suggestions for work include going inside of the analog control board and physically rewiring it such that it would be a linear control system. Once this has been accomplished the user would have the opportunity to do an analysis and comparison of the two different systems.

APPENDIX A - CALIBRATION DATA

SET POINT CALIBRATION

VALUE ON KNOB	VOLTAGE AT POINT B (X-Y OPEN)	SET POINT TEMPERATURE
0.0	0.07 (volts)	0.0 (Celsius)
0.5	0.58	15
1.0	1.21	22
1.5	1.96	26
2.0	2.61	29
2.5	3.26	32
3.0	4.0	34
3.5	4.56	35
4.0	5.25	37
4.5	5.82	38
5.0	6.28	40
5.5	6.87	41
6.0	7.44	43
6.5	8.00	45
7.0	8.72	47
7.5	9.29	49
8.0	10.00	52
8.5	10.72	55
9.0	11.48	59
9.5	12.15	65
10.0	13.10	75

TEMPERATURE TRANSMITTER CALIBRATION

VALUE ON KNOB	VOLTAGE AT POINT Y (X-Y OPEN)	PROCESS TEMPERATURE
0.0	0.0 (volts)	0.0 (celsius)
1.1	0.25	10
1.5	0.65	20
2.1	1.65	25
2.7	2.65	30
3.6	4.25	35
4.6	6.20	40
5.7	7.85	45
7.0	9.40	50
5.0	10.50	55

APPENDIX A - CONTINUED

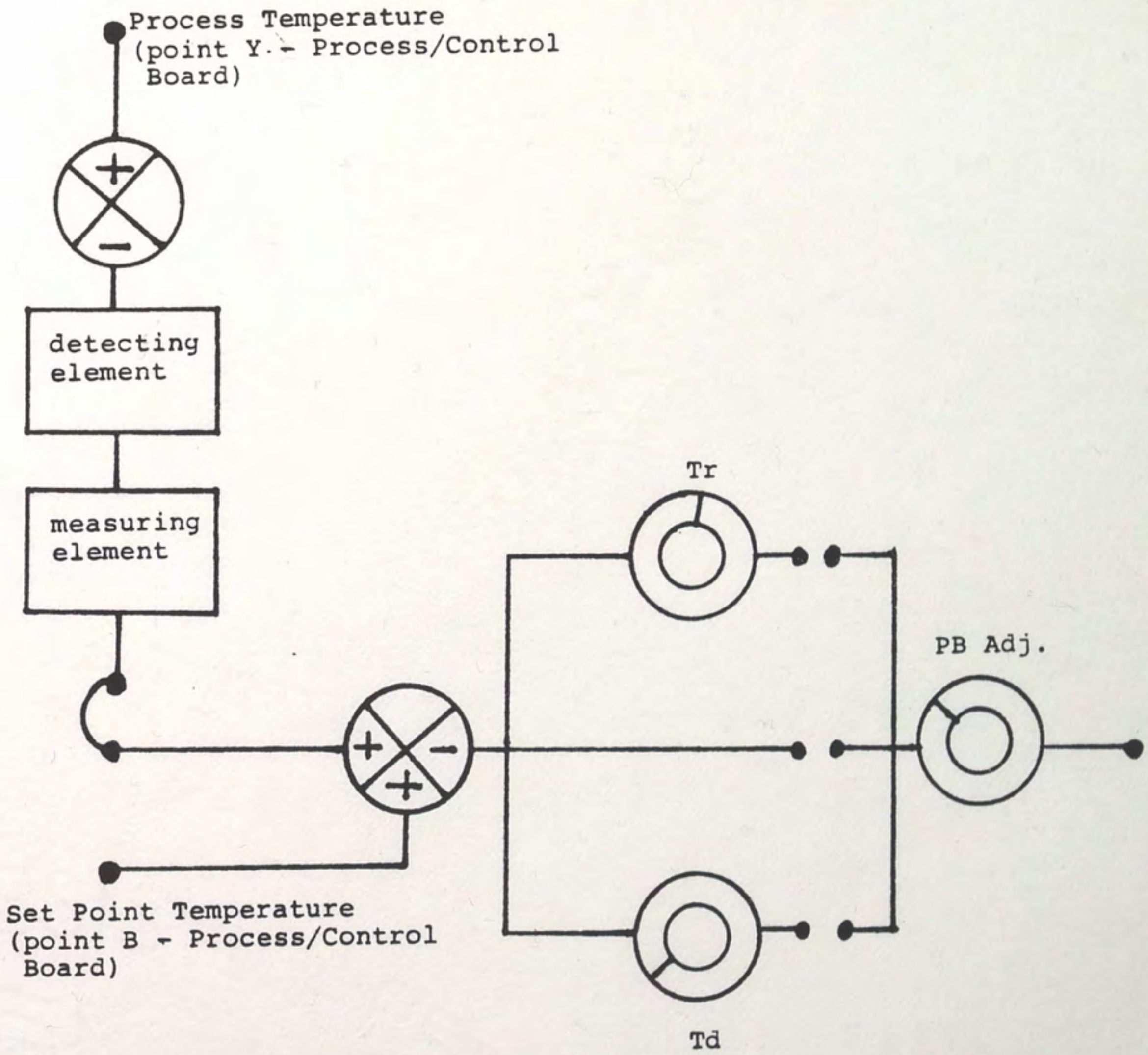
VALUE ON KNOB	VOLTAGE AT POINT Y (X-Y OPEN)	PROCESS TEMPERATURE
5.5	11.50	60
6.0	12.30	70
6.5	13.10	80

FINAL CONTROL ELEMENT CALIBRATION

VALUE ON KNOB	VOLTAGE TO POWER SUPPLY (VOLTS)	WATTAGE OUTPUT TO PROCESS (WATT)
0.0	-4.96	2.75
0.5	-4.71	4.0
1.0	-4.14	6.0
1.5	-3.39	11.0
2.0	-2.69	16.0
2.5	-2.02	22.0
3.0	-1.34	30.0
3.5	-0.71	37.0
4.0	-0.09	45.0
4.5	0.53	54.0
5.0	1.07	60.0
5.5	1.56	66.0
6.0	2.15	76.0
6.5	2.80	79.0
7.0	3.36	84.0
7.5	3.86	88.0
8.0	4.61	94.0
8.5	4.94	96.0
9.0	5.01	99.0
9.5	5.05	100.0

APPENDIX B - ANALOG ON-OFF CONNECTIONS

To examine the response of the process under On-Off control, the user should do the following: (1) change from continuous to two step control (see Fig. 2) (2) set the controller gain to 1 (proportional control only, see below). Shown below is an expanded view of the control portion of the analog control simulator. For control purposes, connections are made as needed, i.e., connect the Tr Adj if the integral term is desired.



APPENDIX C - DOCUMENTED SOFTWARE

```
10  REM
11  REM     THIS PROGRAM IS TITLED "HELLO" PROGRAM. THIS PROGRAM
.12 REM     IS EXECUTED EVERY TIME THE SYSTEM IS BOOTED UP.
13  REM
14  REM     THIS SECTION OF CODE LOADS DOS 3.3 INTO MEMORY AND SETS
15  REM     THE HIGH MEM POINTER JUST BELOW IT.
25  REM
30  TEXT : PRINT
40  HOME
50  PRINT "DOS VERSION 3.3              08/25/80"
60  PRINT : PRINT "APPLE II PLUS OR ROMCARD   SYSTEM MASTER"
70  REM
80  REM  --POKE LANGUAGE CARD FINDER
90  POKE 768,0: POKE 769,173: POKE 770,0: POKE 771,224: POKE 772,72: POKE
     773,173: POKE 774,129: POKE 775,192: POKE 776,104:
100  POKE 777,72: POKE 778,205: POKE 779,0: POKE 780,224: POKE 781,208: POKE
     782,35: POKE 783,173: POKE 784,131: POKE 785,192:
110  POKE 786,173: POKE 787,131: POKE 788,192: POKE 789,169: POKE 790,165
     : POKE 791,141: POKE 792,0: POKE 793,208: POKE 794,205:
120  POKE 795,0: POKE 796,208: POKE 797,208: POKE 798,19: POKE 799,74: POKE
     800,141: POKE 801,0: POKE 802,208: POKE 803,205:
130  POKE 804,0: POKE 805,208: POKE 806,208: POKE 807,10: POKE 808,173: POKE
     809,129: POKE 810,192: POKE 811,173: POKE 812,129:
140  POKE 813,192: POKE 814,169: POKE 815,1: POKE 816,208: POKE 817,2: POKE
     818,169: POKE 819,0: POKE 820,141: POKE 821,0:
150  POKE 822,3: POKE 823,104: POKE 824,205: POKE 825,0: POKE 826,224: POKE
     827,240: POKE 828,3: POKE 829,173: POKE 830,128:
160  POKE 831,192: POKE 832,96:
170  CALL 769
180  IF  PEEK (768) < > 1 THEN 270
190  REM
200  REM --IF THERE, LOAD INTG AND PA#1
210  REM
220  VTAB 10: PRINT "(LOADING INTEGER INTO LANGUAGE CARD)"
230  PRINT "BLOAD INTBASIC,A$D000"
240  REM  --WRITE PROTECT THE CARD
250 A =  PEEK ( - 16254): REM   $C082
320  POKE 768,0: POKE 769,173: POKE 770,0: POKE 771,224: POKE 772,72: POKE
     773,173: POKE 774,129: POKE 775,192: POKE 776,104
330  POKE 777,72: POKE 778,205: POKE 779,0: POKE 780,224: POKE 781,208: POKE
     782,35: POKE 783,173: POKE 784,131: POKE 785,192
340  POKE 786,173: POKE 787,131: POKE 788,192: POKE 789,169: POKE 790,165
     : POKE 791,141: POKE 792,0: POKE 793,208: POKE 794,205
350  POKE 795,0: POKE 796,208: POKE 797,208: POKE 798,19: POKE 799,74: POKE
     800,141: POKE 801,0: POKE 802,208: POKE 803,205
360  POKE 804,0: POKE 805,208: POKE 806,208: POKE 807,10: POKE 808,173: POKE
     809,129: POKE 810,192: POKE 811,173: POKE 812,129:
370  POKE 813,192: POKE 814,169: POKE 815,1: POKE 816,208: POKE 817,2: POKE
     818,169: POKE 819,0: POKE 820,141: POKE 821,0:
380  POKE 822,3: POKE 823,104: POKE 824,205: POKE 825,0: POKE 826,224: POKE
     827,240: POKE 828,3: POKE 829,173: POKE 830,128:
390  POKE 831,192: POKE 832,96:
391  REM
392  REM     THE FOLLOWING SECTION OF CODE LOADS THE OPERATING SYSTEM
393  REM     FOR THE INTERFACE DEVICE (ISAAC). THE OPERATING SYSTEM
394  REM     IS LOADED BELOW DOS IN HIGH MEMORY. THE HIGHEST FREE
395  REM     MEMORY IS 38400 (DECIMAL).
396  REM
400  CALL 769
```

Due to memory requirements the code shown will not execute unless all REM statements are removed. Included only for illustration purposes.

```
410  IF  PEEK (768) < > 1 THEN 510
420  REM
430 X =  PEEK ( - 16247)
440 X =  PEEK ( - 16247)
450  PRINT  CHR$ (4)"BLOAD LABSOFT.RAM.OBJ,A$D000"
460  PRINT  CHR$ (4)"BRUN LABSTART.OBJ,A$300"
470 X =  PEEK ( - 16247)
480 X =  PEEK ( - 16247)
490  PRINT  CHR$ (4)"BLOAD LABEL.SET,A" PEEK (970) +  PEEK (971) * 256
500  GOTO 530
510  PRINT  CHR$ (4)"BRUN LABSOFT.ROM.OBJ,A$7000"
520  PRINT  CHR$ (4)"BLOAD LABEL.SET,A" PEEK (970) +  PEEK (971) * 256
530  TEXT
540  HOME
550  PRINT
560  PRINT  CHR$ (4)"BLOAD USECNT.DAT,A6"
570  IF  PEEK (6) = 0 THEN 670
580  POKE 6, PEEK (6) - 1
590  PRINT  CHR$ (4)"BSAVE USECNT.DAT,A6,L1"
600  PRINT  CHR$ (4)"BLOAD ISAAC.PIC,A$4000"
610  & HIRES2
620  GET X$
630  PRINT
640  TEXT
650  HOME
660  PRINT
670  PRINT "LABSOFT MASTER DISK          VERSION ";
680  FOR N = 2 TO 4
690  PRINT  CHR$ ( PEEK ( PEEK (968) +  PEEK (969) * 256 + N));
700  NEXT N
710  PRINT
720  PRINT "RELEASE ";
730  FOR N = 7 TO 14
740  PRINT  CHR$ ( PEEK ( PEEK (968) +  PEEK (969) * 256 + N));
750  NEXT N
760  IF  PEEK (968) +  PEEK (969) * 256 < 1024 THEN  HTAB 25: PRINT "IN L
     ANGUAGE CARD"
770  IF  PEEK (968) +  PEEK (969) * 256 > 1024 THEN  HTAB 24: PRINT "IN P
     ROGRAM MEMORY"
771  REM
772  REM       THE FOLLOWING SECTION OF CODE LOADS FOUR BINARY FILES
773  REM       FROM DRIVE 1 AND TRANSFERS THEM TO MEMORY ON THE 80-COL
774  REM       EXTENDED MEMORY EXPANSION CARD. THE FOLLOWING SECTION ALSO
775  REM       LOADS INTO LOW MEMORY A MACHINE LANGUAGE SUBROUTINE TO BE
776  REM       USED TO TRANSFER THE BINARY FILES IN EXTENDED MEMORY TO
777  REM       HIGH-RESOLUTION PAGE 1.
778  REM
780  PRINT
790  PRINT
800  PRINT  CHR$ (4) + "PR#3"
850  TEXT : HOME
860  PRINT  CHR$ (17)
870 PARM = 768:PUTPAGE = 769:BRINGPAGE = 800
880  FOR I = 0 TO 64
890  READ BYTE
900  POKE PUTPAGE + I,BYTE
910  NEXT I
920  PRINT  CHR$ (4) + "BLOAD FIRST"
930  POKE PARM,32: CALL PUTPAGE
940  PRINT  CHR$ (4) + "BLOAD FIRST"
950  POKE PARM,64: CALL PUTPAGE
960  PRINT  CHR$ (4) + "BLOAD SECOND"
970  POKE PARM,96: CALL PUTPAGE
980  DATA  169,0,133,60,169,32,133,61,169,248,133,62,169,63,133
990  DATA  63,169,0,133,66,173,0,3,133,67,56,32,17,195,96,0
1000  DATA  169,0,133,66,169,32,133,67,169,0,133,60,173,0,3,133
```

```
1010  DATA  61,169,248,133,62,24,173,0,3,105,31,133,63,24,32,17,195,96
1020  PRINT  CHR$ (4) + "BLOAD THIRD"
1030  POKE PARM,128: CALL PUTPAGE
1040  PRINT  CHR$ (4) + "BLOAD FOURTH"
1050  POKE PARM,160: CALL PUTPAGE
1051  REM
1052  REM        THIS SECTION OF CODE LOADS INTO MEMORY A SHAPE TABLE
1053  REM        TO BE USED TO DRAW THE NUMBERS ON GRAPHS WITH A LABELED
1054  REM         TIME BASE. IT IS STORED STARTING AT ADDRESS 6000.
1055  REM
1060  PRINT  CHR$ (4) + "BLOAD TIME LABELS,A6000"
1070  POKE 232, PEEK (43634): POKE 233, PEEK (43635)
1080  PRINT  CHR$ (18)
1090  LET U$ = "CONTROL PROGRAM"
2000  PRINT  CHR$ (4);"RUN "U$
```

```
200  REM
201  REM      THE FOLLOWING CODE IS TITLED CONTROL PROGRAM. THIS SECTION
202  REM      OF CODE IS USED TO DECIDE WHAT PROGRAM THE SYSTEM WILL BE
203  REM      RUNNING WHEN THE USER IS VEIWING A GRAPH. THE FIRST
204  REM      QUESTION ASKED OF THE USER IS ANALOG/DIGITAL CONTROL?
205  REM
230  HIMEM: 38395
240  LOMEM: 16384
250  & SLOT# = 2
260  HOME : PRINT "THIS FLOPPY DISK CONTAINS PROGRAMS WHICH"
270  PRINT "ARE AN INTEGRAL PART TO THE THESIS OF CLIFTON R. HARGROVE"
280  PRINT "ANY UNAUTHORIZED TAMPERING WITH SOFTWARE HEREIN IS "
290  PRINT "STRICTLY PROHIBITED."
300  VTAB 7: PRINT "THE FOLLOWING PROGRAMS ALLOW EXAMINATION OF"
310  PRINT "PROCESS TRAINER PT326 FROM FEEDBACK."
320  VTAB 12: PRINT "THE FOLLOWING OPTIONS ARE AVAILABLE AT THIS TIME:"
330  PRINT : PRINT "  1.  ANALOG CONTROL"
340  PRINT "  2.  DIGITAL CONTROL"
350  PRINT "  3.  QUIT "
360  PRINT : INPUT "ENTER CHOICE=";Q
365  PRINT  CHR$ (18)
370  IF Q = 1 THEN  GOTO 1100
380  IF Q = 2 THEN  GOTO 430
390  IF Q = 3 THEN  END
400  GOTO 320
410  REM
411  REM      THIS PATH IS TAKEN IF USER SELECTS DIGITAL CONTROL.
412  REM .    THE NEXT QUESTION ASKED OF THE USER IS WHICH CATEGORY
413  REM      OF GRAPHS. IF GRAPHS WITH A LABELED TIME BASE THEN THE
414  REM      CONTROL PROGRAM WILL ASK ALL QUESTIONS PERTAINING TO
415  REM      SELECTION.
416  REM
430  HOME : PRINT "WHICH OF THE FOLLOWING DO YOU WANT TO EXAMINE?"
440  PRINT : PRINT "  1. REAL-TIME MONITORING"
450  PRINT "  2. GRAPHS WITH LABELED TIME BASE"
460  PRINT "  3. BANG-BANG CONTROL (ON-OFF CONTROL)"
470  PRINT : INPUT "ENTER CHOICE NOW=";A
480  IF A = 1 THEN U$ = "DPART"
490  IF A = 2 THEN  GOTO 600
500  IF A = 3 THEN U$ = "BANGBANG"
505  IF A <  > 1 AND A <  > 2 AND A <  > 3 THEN  GOTO 430
510  PRINT  CHR$ (4);"RUN "U$
515  REM
516  REM       IF THE USER SELECTED REAL-TIME MONITORING THEN CONTROL
517  REM       IS TRANSFERRED TO "DPART" PROGRAM. IF BANG-BANG CONTROL
518  REM       WAS SELECTED THEN TRANSFERRED TO "BANG-BANG" PROGRAM.
519  REM       THIS PATH IS TAKEN IF USER SELECTED GRAPHS WITH LABELED
520  REM        TIME BASE. THE FIRST STEP TAKEN IS TO DETERMINE AND
521  REM        OUTPUT DESIGN WATTAGE TO PROCESS.
522  REM
600  GOSUB 1000:OLD = OUT:COMPAR = 136 * (X - 30):BLO = Y
610  & AOUT, (DV) = OUT, (C#) = 0
620  & AOUT, (DV) = COMPAR, (C#) = 1
621  REM
622  REM       THE FOLLOWING SECTION ASKED THE USER WHICH OF THE
623  REM       AVAILABLE CONTROL SYSTEMS DOES HE WANT?
624  REM
630  HOME : PRINT "YOU HAVE FOLLOWING CHOICE CONCERNING CONTROL"
```

```
640  PRINT "ALGORITHMS AT THIS POINT:"
650  PRINT : PRINT "  1. PROPORTIONAL CONTROL"
660  PRINT "  2. PROPORTIONAL-INTEGRAL CONTROL"
670  PRINT "  3. PROPORTIONAL-INTEGRAL-DERIVATIVE CONTROL"
680  PRINT : INPUT "ENTER YOUR CHOICE NOW=";Y
690  IF Y = 1 THEN  GOTO 750
700  IF Y = 2 THEN  GOTO 760
710  IF Y = 3 THEN  GOTO 790
720  GOTO 630
721  REM
722  REM        FOR WHICHEVER TYPE OF CONTROL SYSTEM SELECTED, THE
723  REM        FOLLOWING CODE ASK FOR ITS ASSOCIATED PARAMETERS.
724  REM
750  PRINT : INPUT "ENTER AMOUNT OF GAIN (Kc)=";GAIN:K = 1: GOTO 830
760  PRINT : INPUT "ENTER AMOUNT OF GAIN (Kc)=";GAIN
770  PRINT : INPUT "ENTER RESET TIME (SECONDS)=";TR
775  PRINT : PRINT "NOTE: SMALLEST SAMPLE TIME=100 (MSEC)"
776  INPUT "ENTER DESIRED SAMPLE TIME (SECONDS)=";TIME
777 TR = TR * 1000:TIME = TIME * 1000:K = 2: GOTO 830
790  PRINT : INPUT "ENTER AMOUNT OF GAIN (Kc)=";GAIN
800  PRINT : INPUT "ENTER RESET TIME (SECONDS)=";TR
810  PRINT : INPUT "ENTER DERIVATIVE TIME (SECONDS)=";TD
815  PRINT : PRINT "NOTE: SMALLEST SAMPLE TIME=100 (MSEC)"
820  INPUT "ENTER DESIRED SAMPLE TIME (SECONDS)=";TIME
825 TR = TR * 1000:TD = TD * 1000:TIME = TIME * 1000:K = 3
826  REM
827  REM       THE FINAL QUESTION IS REGULATOR OR SERVO RESPONSE?
828  REM
830  HOME : PRINT "AT THIS TIME YOU HAVE FOLLOWING OPTIONS:"
840  PRINT : PRINT "  1. REGULATOR CONTROL (RESPONSE TO LOAD VARIABLE CHA
     NGE)"
850  PRINT "  2. SERVO CONTROL (RESPONSE TO SET POINT CHANGE)"
880  PRINT : INPUT "ENTER CHOICE NOW=";Z
885  IF Z <  > 1 AND Z <  > 2 THEN  GOTO 830
890  IF Z = 1 THEN  GOTO 910
900  GOSUB 1006:OWE = 136 * (X - 30)
901  REM
902  REM       THE FOLLOWING SECTION OF CODE OUTPUTS TO A TEXT FILE
903  REM       ALL PARAMETERS JUST GATHERED. THIS TEXT FILE WILL BE
904  REM       OPENED AND READ BY NEW PROGRAM. CONTROL IS THEN TRANS-
905  REM       FERRED TO APPROPRIATE PROGRAM
906  REM
910  PRINT  CHR$ (4);"OPEN VARPASS"
920  PRINT  CHR$ (4);"WRITE VARPASS"
930  PRINT OLD: PRINT COMPAR: PRINT GAIN: PRINT TR: PRINT TD
940  PRINT K: PRINT OWE: PRINT TIME
950  PRINT  CHR$ (4) + "CLOSE" + U$
960  IF Z = 1 THEN U$ = "REGULATOR"
970  IF Z = 2 THEN U$ = "SERVO"
995  PRINT  CHR$ (4);"RUN "U$
996  REM
997  REM       THE FOLLOWING SUBROUTINE COMPUTES DESIGN WATTAGE TO BE
998  REM       OUTPUT TO PROCESS. ALSO CHECKS FOR ALLOWABLE CONDITIONS.
999  REM
1000  HOME : PRINT "AT THIS TIME YOU NEED TO ENTER DESIGN TEMPERATURE"
1001  PRINT "AND CURRENT BLOWER INLET."
1002  PRINT : PRINT "NOTE: DESIGN TEMP MUST BE BETWEEN 30 AND 60 (C)"
1003  INPUT "ENTER DESIGN SET POINT (C)=";X: IF X < 30 OR X > 60 THEN  GOTO
     1000
1004  PRINT : PRINT "NOTE: BLOWER INLET MUST BE BETWEEN 10 AND 170 (DEG)"

1005  INPUT "PLEASE ENTER BLOWER INLET=";Y: GOTO 1014
1006  PRINT : PRINT : INPUT "ENTER NEW SET POINT TEMP=";X
1007  IF X < 30 OR X > 60 THEN  GOTO 1006:Y = BLO
1014  LET Q = 0:Q =  - 65.176394 + (2.46541807 * X)
1016  LET Q = Q + ( - 0.0167582955 * X ^ 2) + ( - 1.80406 * Y)
```

```
1018  LET Q = Q + (0.0199352325 * Y ^ 2) + (.110026458 * X * Y)
1020  LET Q = Q + ( - 9.88198962E - 04 * X ^ 2 * Y) + ( - 1.18368066E - 0
     3 * X * Y ^ 2)
1022  LET Q = Q + (1.5608222E - 05 * X ^ 2 * Y ^ 2)
1024  IF Q < 4 OR Q > 96 THEN  GOTO 1036
1026  LET V = (0.1 * Q) - 5:OUT = 409.5 * (V + 5)
1034  GOTO 1048
1036  PRINT : PRINT "YOU HAVE CHOSEN DESIGN CONDITIONS WHICH"
1038  PRINT "ARE NOT PHYSICALLY REALIZABLE BECAUSE THE MAXIMUM"
1040  PRINT "HEATER OUTPUT IS 100 WATTS."
1042  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE"
1044  IF  PEEK ( - 16384) < 127 THEN  GOTO 1042
1046  GET T$: GOTO 1000
1048  RETURN
1060  REM
1061  REM      THIS PATH IS TAKEN IF USER SELECTS ANALOG CONTROL.
1062  REM      THE ONLY REMAINING QUESTION IS REAL-TIME OR GRAPHS
1063  REM      WITH LABELED TIME BASE?
1064  REM
1100  HOME : PRINT "WHICH OF THE FOLLOWING DO YOU WISH TO DO?"
1110  PRINT : PRINT "  1. REAL-TIME MONITORING"
1115  PRINT "  2. GRAPHS WITH LABELED TIME BASE"
1120  PRINT : INPUT "ENTER CHOICE=";A
1122  IF A <  > 1 AND A <  > 2 THEN  GOTO 1100
1125  IF A = 1 THEN U$ = "APART"
1130  IF A = 2 THEN U$ = "ANAPLOT"
1135  PRINT  CHR$ (4);"RUN "U$
```

```
10  HIMEM: 38395
12  LOMEM: 16384
15  & SLOT# = 2
16  REM
17  REM      THIS PROGRAM IS TITLED "APART" AND IT ALLOWS REAL
18  REM      TIME MONITORING UNDER ANALOG CONTROL.
20  REM
21  REM     ONCE PROGRAM FLOW REACHES THIS POINT THE ONLY REMAINING
22  REM     QUESTION IS "WHICH OF THE AVAILABLE GRAPHS DOES THE USER
23  REM     WISH TO SEE?"
24  REM
170 PARM = 768:PUTPAGE = 769:BRINGPAGE = 800
180  HOME : PRINT "THE FOLLOWING OPTIONS ARE AVAILABLE AT THIS TIME:"
190  PRINT : PRINT "  1.  REAL-TIME DISPLAY OF MANIPULATED VARIABLE"
200  PRINT "  2.  REAL-TIME DISPLAY OF CONTROLLED VARIABLE"
210  PRINT "  3.  REAL-TIME DISPLAY OF TEMP SET POINT"
220  PRINT "  4.  REAL-TIME DISPLAY OF SET POINT AND AIR TEMP"
240  PRINT "  5.  QUIT"
250  PRINT : INPUT "ENTER CHOICE:";Q
260  IF Q = 1 THEN  GOTO 330
270  IF Q = 2 THEN  GOTO 390
280  IF Q = 3 THEN  GOTO 460
290  IF Q = 4 THEN  GOTO 530
300  IF Q = 5 THEN  GOTO 1870
310  REM
311  REM      ONCE THE USER SELECTS CHOICE, THE PROGRAM BRANCHES
312  REM      TO THE APPROPRIATE ROUTINE TO HANDLE IT
313  REM
320  GOTO 180
321  REM
322  REM      THE FOLLOWING SECTION ALLOWS EXAMINATION OF WATTAGE
323  REM      (MANIPULATED VARIABLE) UNDER ANALOG CONTROL
324  REM
330  GOSUB 1770: POKE PARM,64: CALL BRINGPAGE: GOSUB 1820
340  & PLTFMT = 2
350  & AIN,(TV) = N,(C#) = 0,(GA)
370  LET N = 0: IF  PEEK ( - 16384) < 127 THEN  GOTO 350
380  GET T$: TEXT : GOTO 180
381  REM
382  REM     THE FOLLOWING ROUTINE HANDLES EXAMINATION OF CONTROLLED
383  REM      VARIABLE, PROCESS TEMPERATURE, UNDER ANALOG CONTROL
384  REM
390  GOSUB 1770: POKE PARM,96: CALL BRINGPAGE: GOSUB 1820
400  & PLTFMT = 3
410  & AIN,(TV) = N,(C#) = 1,(GA)
440  LET N = 0: IF  PEEK ( - 16384) < 127 THEN  GOTO 410
450  GET T$: TEXT : GOTO 180
451  REM
452  REM      THE FOLLOWING SECTION ALLOWS EXAMINATION OF SET POINT
453  REM      TEMPERATURE UNDER ANALOG CONTROL
454  REM
460  GOSUB 1770: POKE PARM,128: CALL BRINGPAGE: GOSUB 1820
470  & PLTFMT = 6
480  & AIN,(TV) = N,(C#) = 2,(GA)
510  IF  PEEK ( - 16384) < 127 THEN  GOTO 480
520  GET T$: TEXT : GOTO 180
521  REM
522  REM     THE FOLLOWING SECTION EXAMINES BOTH TEMPERATURES. SET POINT
```

```
523  REM    AND PROCESS TEMPERATURE, UNDER ANALOG CONTROL
524  REM
530  GOSUB 1770: POKE PARM,160: CALL BRINGPAGE: GOSUB 1820
550  & PLTFMT = 2,5
560  & AIN, (TV) = N, (C#) = 1, (GA)
570  & AIN, (TV) = Y, (C#) = 2, (GA)
580  IF  PEEK ( - 16384) < 127 THEN  GOTO 560
585  & FMTDFLT: GET T$: TEXT : GOTO 180
1770  HGR : HCOLOR= 1: ROT= 0: SCALE= 1
1780  & SCROLLSET
1790  RETURN
1800  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE"
1810  RETURN
1820  VTAB 22: PRINT "PRESS ANY KEY TO RETURN TO MENU"
1830  RETURN
1860  END
1870  PRINT  CHR$ (4) + "PR#0": PRINT  CHR$ (4) + "PR#3"
1871  REM
1872  REM    THIS SECTION DETERMINES WHERE TO GO FROM HERE. THE
1873  REM    USER CAN QUIT OR CONTINUE ON
1874  REM
1875  HOME : INPUT "DO YOU WISH TO CONTINUE (Y OR N)";A$
1880  IF A$ = "Y" THEN U$ = "CONTROL PROGRAM"
1890  IF A$ = "N" THEN  END
1900  IF A$ <  > "N" AND A$ <  > "Y" THEN  GOTO 1875
1910  PRINT  CHR$ (4);"RUN "U$
```

```
10  REM
11  REM   THIS PROGRAM IS TITLED "ANAPLOT". THE FIRST SECTION DOES
12  REM   FOLLOWING: 1. SET MEMORY POINTERS 2. DIMENSION AN ARRAY
13  REM   3. LOADS A SHAPE TABLE 4. ACTIVATE THE INTERFACE 5. DECLARE
14  REM   VARIABLES
15  REM
20  HIMEM: 38395
30  LOMEM: 16384
40  DIM Y(2,260)
45  PRINT  CHR$ (4);"BLOAD TIME LABELS,A6000"
47  POKE 232, PEEK (43634): POKE 233, PEEK (43635)
50  & SLOT# = 2
60 PARM = 768:PUTPAGE = 769:BRINGPAGE = 800
61  REM
62  REM    THE FOLLOWING SECTION BEGINS GATHERING PARAMETERS  i.e.
63  REM    WHAT EXAMINATION DOES THE WANT?
64  REM
70  HOME : INPUT "DO YOU WANT A HARDCOPY OF GRAPHS (Y OR N)";C$
80  IF C$ <  > "N" AND C$ <  > "Y" THEN  GOTO 70
90  VTAB 7: PRINT "NOTE:MIN LENGTH OF SAMPLE IS 1 SECOND"
100  PRINT "NOTE: MAX LENGTH OF SAMPLE IS 115 SECONDS"
110  INPUT "HOW LONG OF A SAMPLE DO YOU WANT (SECONDS)";Q
120  IF Q < 1 OR Q > 115 THEN  GOTO 90
180  VTAB 15: PRINT "AT THE BEEP DATA ACQUISITION BEGINS."
190  PRINT "YOU HAVE ";: PRINT Q;: PRINT " SECONDS OF ACQUISITION TIME."
200  & PAUSE = 5:SEC = Q
201  REM
202  REM   THE FOLLOWING SECTION TURNS ON THE BEEPER, DOES 260 DATA
203  REM   ACQUISIITIONS, TURNS THE BEEPER BACK OFF.
204  REM
230  LET TIME =  INT ((Q / .78) + .5)
240  & BEEP ON
250  & ANAFMT = 0,1,2
260  & @AIN,(AV) = Y,(RT) = TIME
270  & BEEP STOP
271  REM
272  REM   THIS SECTION GENERATES A GRAPH OF WATTAGE OUTPUT TO THE
273  REM   PROCESS DURING EXAMINATION TIME.
274  REM
280  GOSUB 1140: POKE PARM,64: CALL BRINGPAGE: GOSUB 510
285  & PLTFMT = 2
290  FOR W = 0 TO 260
300  & NXTPLT = Y(0,W) / 32
310  NEXT W
320  GOSUB 1400
324  IF  PEEK ( - 16384) > 127 THEN  GOTO 328
326  GOTO 324
328  IF C$ = "Y" THEN  GOSUB 1450
330  REM
331  REM   THE FOLLOWING SECTION GENERATES A GRAPH OF THE PROCESS
332  REM   TEMPERATURE RESPONSE DURING THE EXAMINATION.
333  REM
340  GET T$: GOSUB 1140: POKE PARM,96: CALL BRINGPAGE: GOSUB 510
345  & PLTFMT = 7
350  FOR W = 0 TO 260
360  & NXTPLT = Y(1,W) / 32
370  NEXT W
380  GOSUB 1400
```

```
384  IF  PEEK ( - 16384) ) 127 THEN  GOTO 388
386  GOTO 384
388  IF C$ = "Y" THEN  GOSUB 1450
390  REM
391  REM   THIS SECTION GENERATES A GRAPH OF THE SET POINT TEMPERATURE
392  REM   RESPONSE DURING THE EXAMINATION
393  REM
400  GET T$: GOSUB 1140: POKE PARM,128: CALL BRINGPAGE: GOSUB 510
405  & PLTFMT = 5
410  FOR W = 0 TO 260
420  & NXTPLT = Y(2,W) / 32
430  NEXT W
440  GOSUB 1400
444  IF  PEEK ( - 16384) ) 127 THEN  GOTO 448
446  GOTO 444
448  IF C$ = "Y" THEN  GOSUB 1450
450  GET T$: GOSUB 1140: POKE PARM,160: CALL BRINGPAGE: GOSUB 510
452  & PLTFMT = 3,5
454  FOR W = 0 TO 260
456  & NXTPLT = Y(1,W) / 32
458  & NXTPLT = Y(2,W) / 32
460  NEXT W
461  & FMTDFLT: GOSUB 1400
462  IF  PEEK ( - 16384) ) 127 THEN  GOTO 464
463  GOTO 462
464  IF C$ = "Y" THEN  GOSUB 1450
465  TEXT : HOME
466  REM
467  REM   THE FOLLOWING SECTION DETERMINES PROGRAM FLOW FROM THIS
468  REM   POINT. CONTINUE OR TERMINATE?
469  REM
470  INPUT "DO YOU WISH TO CONTINUE ON (Y OR N)";A$
480  IF A$ = "N" THEN  END
490  IF A$ = "Y" THEN  GOTO 1220
500  GOTO 470
501  REM
502  REM   THE FOLLOWING CODE LABELS THE GRAPH WITH A TIME BASE.
503  REM   THIS SECTION IS A SUBROUTINE CALLED BY THE MAIN PROGRAM.
504  REM
510  LET N = (SEC / 7) * 2
520  IF N ( )  INT (N) THEN  GOTO 540
530  LET N = N + .001
540  IF N ) 3.3 THEN  GOTO 850
550  LET X = 80:Y = 156: GOSUB 650
570  LET N = N * 2:X = 150: GOSUB 650
600  LET N = N + (N / 2):X = 230: GOSUB 650
630  HPLOT 88,156: HPLOT 158,156: HPLOT 238,156
640  RETURN
650  LET D = N + .05:D$ =  STR$ (D)
670  FOR J = 1 TO 15
680  IF  MID$ (D$,J,1) = "." THEN  GOTO 700
690  NEXT J
700  LET J = J - 1
710  IF J ( ) 0 THEN  GOTO 740
720  LET B = 10: GOSUB 820: GOTO 750
740  GOSUB 780
750  LET J = J + 2
760  GOSUB 780: RETURN
780  LET B$ =  MID$ (D$,J,1)
790  LET B =  VAL (B$)
800  IF B ( ) 0 THEN  GOTO 820
810  LET B = 10
820  DRAW B AT X,Y
830  LET X = X + 10
840  RETURN
850  LET X = 80:Y = 156: GOSUB 940
```

```
870  LET N = N * 2:X = 150: GOSUB 940
900  LET N = N + (N / 2):X = 230: GOSUB 940
930  RETURN
940  LET D = N + .5
950  LET D$ =  STR$ (D)
960  FOR J = 1 TO 15
970  IF  MID$ (D$,J,1) = "." THEN  GOTO 990
980  NEXT J
990  LET J = J - 2
1000  IF J <  > 0 THEN  GOTO 1030
1010  LET B = 10: GOSUB 1110
1020  GOTO 1040
1030  GOSUB 1070
1040  LET J = J + 1
1050  GOSUB 1070
1060  RETURN
1070  LET B$ =  MID$ (D$,J,1)
1080  LET B =  VAL (B$)
1090  IF B <  > 0 THEN  GOTO 1110
1100  LET B = 10
1110  DRAW B AT X,Y:X = X + 8: RETURN
1120  REM
1121  REM   THE FOLLOWING SECTION IS A SUBROUTINE CALLED BY THE
1122  REM   MAIN PROGRAM TO CLEAR THE SCREEN AND READY FOR DRAWING
1123  REM   ANOTHER GRAPH.
1124  REM
1140  HGR : HCOLOR= 3: ROT= 0: SCALE= 1
1150  & SCROLLSET
1160  RETURN
1170  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE"
1180  RETURN
1190  VTAB 22: PRINT "PRESS ANY KEY TO RETURN TO MENU"
1200  RETURN
1220  LET U$ = "CONTROL PROGRAM": PRINT  CHR$ (9);"e"
1230  PRINT  CHR$ (4);"RUN "U$
1400  HOME : VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE"
1410  RETURN
1450  PRINT  CHR$ (4);"PR#1": PRINT  CHR$ (9);"G": PRINT  CHR$ (9);"e"
1455  RETURN
```

```
10  HIMEM: 38395
20  LOMEM: 16384
30  & SLOT# = 2
31  REM
32  REM     THIS PROGRAM IS TITLED "DPART". THIS PROGRAM ALLOWS REAL
33  REM     TIME MONITORING UNDER DIGITAL CONTROL.
34  REM
40  REM
50  REM    THIS SECTION OF CODE SETS MEMORY POINTERS, ACTIVATES THE
60  REM     INTERFACE DEVICE, AND DECLARES VARIABLES, RESPECTIVELY.
70  REM
80 PARM = 768:PUTPAGE = 769:BRINGPAGE = 800
90  REM
100  REM   THE FOLLOWING SECTION ASKS THE USER FOR DESIRED
110  REM   SET POINT TEMPERATURE AND BLOWER INLET. THEN PROGRAM
120  REM   OUTPUTS TO THE PROCESS DESIGN WATTAGE TO MAINTAIN
130  REM   DESIRED SET POINT
140  REM
150  GOSUB 1400:OLD = OUT:COMPAR = 136 * (X - 30):BLO = Y
160  & AOUT, (DV) = OUT, (C#) = 0
170  & AOUT, (DV) = COMPAR, (C#) = 1
180  REM
190  REM   AT THIS POINT PROGRAM ASKS USER DESIRED TYPE OF CONTROL
200  REM   SYSTEM WHILE UNDER DIGITAL CONTROL
210  REM
220  HOME : PRINT "YOU HAVE THE FOLLOWING OPTIONS CONCERNING"
230  PRINT "CONTROL ALGORITHMS AT THIS POINT:"
240  PRINT : PRINT "  1. PROPORTIONAL CONTROL"
250  PRINT "  2. PROPORTIONAL-INTEGRAL CONTROL"
260  PRINT "  3. PROPORTIONAL-INTEGRAL-DERIVATIVE CONTROL"
270  PRINT "  4. QUIT"
280  PRINT : INPUT "ENTER YOUR CHOICE NOW=";Y
290  IF Y = 1 THEN  GOTO 430
300  IF Y = 2 THEN  GOTO 460
310  IF Y = 3 THEN  GOTO 500
320  IF Y = 4 THEN  GOTO 380
330  GOTO 220
340  REM
350  REM   THIS SECTION DEALS WITH PROGRAM FLOW AFTER
360  REM    USER INPUTS 4 (QUIT)
370  REM
380  HOME : INPUT "DO YOU WISH TO CONTINUE (Y OR N)";A$
390  IF A$ = "N" THEN  END
400  IF A$ = "Y" THEN  GOTO 420
410  GOTO 380
420  LET U$ = "CONTROL PROGRAM": PRINT  CHR$ (4);"RUN "U$
421  REM
422  REM   THE FOLLOWING SECTION OF CODE IS BRANCHED TO DEPENDING ON THE
423  REM   TYPE OF CONTROL SYSTEM SELECTED EARLIER. THIS SECTION OF CODE
424  REM   ASKS THE USER FOR CONTROL PARAMETERS.
425  REM
430  PRINT : INPUT "ENTER AMOUNT OF GAIN (Kc)=";GAIN
440  PRINT : INPUT "ENTER DESIRED SAMPLE TIME (SECONDS)=";TIME
450  LET TIME = TIME * 1000:K = 1: GOTO 550
460  PRINT : INPUT "ENTER AMOUNT OF GAIN (Kc)=";GAIN
470  PRINT : INPUT "ENTER RESET TIME (SECONDS)=";TR
480  PRINT : INPUT "ENTER DESIRED SAMPLE TIME (SECONDS)=";TIME
490  LET TR = TR * 1000:TIME = TIME * 1000:K = 2: GOTO 550
```

```
500  PRINT : INPUT "ENTER AMOUNT OF GAIN (Kc)=";GAIN
510  PRINT : INPUT "ENTER RESET TIME (SECONDS)=";TR:TR = TR * 1000
520  PRINT : INPUT "ENTER DERIVATIVE TIME (SECONDS)=";TD
530  PRINT : INPUT "ENTER DESIRED SAMPLE TIME (SECØNDS)=";TIME
540  LET TD = TD * 1000:TIME = TIME * 1000:K = 3
541  REM
542  REM   THIS SECTION OF CODE QUERIES THE USER ON THE TYPE OF
543  REM   RESPONSE (REGULATOR OR SERVO)?
544  REM
550  HOME : PRINT "AT THIS TIME YOU HAVE FOLLOWING OPTIONS:"
560  PRINT
570  PRINT "  1. REGULATOR CONTROL (RESPONSE TO LOAD VARIABLE CHANGE)"
580  PRINT "  2. SERVO CONTROL (RESPONSE TO SET POINT CHANGE)"
590  PRINT : INPUT "ENTER CHOICE=";Z
600  IF Z <  > 1 AND Z <  > 2 THEN  GOTO 550
610  IF Z = 1 THEN  GOTO 630
620  GOSUB 1470:DWE = 136 * (X - 30)
621  REM
622  REM   THE FOLLOWING SECTION ASKS THE USER WHICH OF THE FOUR GRAPHS
623  REM   AVAILABLE DOES HE WANT TO SEE?
624  REM
630  HOME : PRINT "YOU HAVE FOLLOWING OPTIONS AT THIS POINT:"
640  PRINT : PRINT "  1.  REAL-TIME DISPLAY OF WATTAGE"
650  PRINT "  2.  REAL-TIME DISPLAY OF AIR TEMPERATURE"
660  PRINT "  3.  REAL-TIME DISPLAY OF SET POINT TEMPERATURE"
670  PRINT "  4.  REAL-TIME DISPLAY OF BOTH TEMPERATURES"
680  PRINT : PRINT "NOTE: YOU ARE NOW RUNNING CONTROL ALGORITHM"
690  PRINT "THAT YOU JUST SELECTED"
700  PRINT : INPUT "ENTER CHOICE=";Y
710  IF Z = 1 AND Y = 1 THEN  GOTO 820
720  IF Z = 2 AND Y = 1 THEN  GOTO 830
730  IF Z = 1 AND Y = 2 THEN  GOTO 840
740  IF Z = 2 AND Y = 2 THEN  GOTO 850
750  IF Z = 1 AND Y = 3 THEN  GOTO 860
760  IF Z = 2 AND Y = 3 THEN  GOTO 870
770  IF Z = 1 AND Y = 4 THEN  GOTO 880
780  IF Z = 2 AND Y = 4 THEN  GOTO 970
810  GOTO 630
820  LET Z = 0:T = 2: GOSUB 1140: GOTO 220
830  LET Z = 0:T = 2: GOSUB 1230: GOTO 150
840  LET Z = 1:T = 3: GOSUB 1140: GOTO 220
850  LET Z = 1:T = 3: GOSUB 1230: GOTO 150
860  LET Z = 3:T = 4: GOSUB 1140: GOTO 220
870  LET Z = 3:T = 4: GOSUB 1230: GOTO 150
871  REM
872  REM   THE FOLLOWING IS A SERIES OF SUBROUTINES. EACH ONE PROVIDES
873  REM   THE ABILITY TO SEE ONE PARTICULAR CASE e.g. A GRAPH OF
874  REM   THE WATTAGE OUTPUT WHILE UNDER A PREDEFINED CONTROL
875  REM   SYSTEM
876  REM
877  REM  THE FIRST ROUTINE SHOWS BOTH TEMPERATURES , REGULATOR RESPONSE
878  REM
880  GOSUB 1640: GOSUB 1670: POKE PARM,160: CALL BRINGPAGE
890  & PLTFMT = 3,5:LAST = 0:PREV = 0:MIST = 0
900  & CLRTIMER
910  & AIN,(TV) = IN,(C#) = 1,(GA)
920  & AIN,(TV) = R,(C#) = 3,(GA)
930  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 910
940  ON K GOSUB 1710,1720,1760: & AOUT,(DV) = OUT,(C#) = 0
950  IF  PEEK ( - 16384) < 127 THEN  GOTO 900
960  & FMTDFLT: GET T$: TEXT : GOTO 220
961  REM
962  REM  THIS ROUTINE ALLOWS A GRAPH OF BOTH TEMPERATURES
963  REM  SERVO RESPONSE (SET POINT CHANGE)
964  REM
970  GOSUB 1640: GOSUB 1690: POKE PARM,160: CALL BRINGPAGE
```

```
980  & PLTFMT = 3,6:LAST = 0:PREV = 0:MIST = 0
990  & CLRTIMER
1000  & AIN,(TV) = IN,(C#) = 1,(GA)
1010  & AIN,(TV) = N,(C#) = 3,(GA)
1020  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 1000
1030  ON K GOSUB 1710,1720,1760: & AOUT,(DV) = OUT,(C#) = 0
1040  IF  PEEK ( - 16384) < 127 THEN  GOTO 990
1050  GET T$:COMPAR = OWE: GOSUB 1670
1060  & AOUT,(DV) = COMPAR,(C#) = 1
1070  & CLRTIMER
1080  & AIN,(TV) = IN,(C#) = 1,(GA)
1090  & AIN,(TV) = N,(C#) = 3,(GA)
1100  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 1080
1110  ON K GOSUB 1710,1720,1760: & AOUT,(DV) = OUT,(C#) = 0
1120  IF  PEEK ( - 16384) < 127 THEN  GOTO 1070
1130  & FMTDFLT: TEXT : GOTO 150
1131  REM
1132  REM   THE FOLLOWING SUBROUTINE GENERATES GRAPH OF EITHER
1133  REM   WATTAGE, PROCESS OR SET POINT TEMP, REGULATOR RESPONSE
1134  REM
1140  GOSUB 1640: GOSUB 1670: POKE PARM,T * 32: CALL BRINGPAGE
1150  & PLTFMT = 1:LAST = 0:PREV = 0:MIST = 0
1160  & CLRTIMER
1170  & AIN,(TV) = N,(C#) = Z,(GA)
1180  & AIN,(TV) = IN,(C#) = 1
1190  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 1170
1200  ON K GOSUB 1710,1720,1760: & AOUT,(DV) = OUT,(C#) = 0
1210  IF  PEEK ( - 16384) < 127 THEN  GOTO 1160
1220  GET T$: TEXT : RETURN
1221  REM
1222  REM  THIS ROUTINE EXAMINES SERVO RESPONSE OF EITHER WATTAGE
1223  REM   PROCESS OR SET POINT TEMPERATURE
1224  REM
1230  GOSUB 1640: GOSUB 1690: POKE PARM,T * 32: CALL BRINGPAGE
1240  & PLTFMT = 6:LAST = 0:PREV = 0:MIST = 0
1250  & CLRTIMER
1260  & AIN,(TV) = N,(C#) = Z,(GA)
1270  & AIN,(TV) = IN,(C#) = 1
1280  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 1260
1290  ON K GOSUB 1710,1720,1760: & AOUT,(DV) = OUT,(C#) = 0
1300  IF  PEEK ( - 16384) < 127 THEN  GOTO 1250
1310  GET T$:COMPAR = OWE: GOSUB 1670
1320  & AOUT,(DV) = COMPAR,(C#) = 1
1330  & CLRTIMER
1340  & AIN,(TV) = N,(C#) = Z,(GA)
1350  & AIN,(TV) = IN,(C#) = 1
1360  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 1340
1370  ON K GOSUB 1710,1720,1760: & AOUT,(DV) = OUT,(C#) = 0
1380  IF  PEEK ( - 16384) < 127 THEN  GOTO 1330
1390  GET T$: TEXT : RETURN
1391  REM
1392  REM   THIS SUBROUTINE QUERIES USER ABOUT DESIRED SET POINT
1393  REM   TEMP AND BLOWER INLET AND THEN SOLVES A 9 TERM POLYNOMIAL
1394  REM   EQUATION TO COMPUTE DESIGN WATTAGE NEEDED. IF ILLEGAL
1395  REM   ENTRY WAS MADE, ANOTHER ONE MUST BE MADE.
1396  REM
1400  HOME : PRINT "AT THIS TIME YOU NEED TO ENTER DESIGN TEMP"
1410  PRINT "AND DESIGN BLOWER INLET"
1420  VTAB 4: PRINT "NOTE: TEMP MUST BE BETWEEN 30 AND 60 (C)"
1430  INPUT "ENTER DESIGN TEMP=";X: IF X < 30 OR X > 60 THEN  GOTO 1420
1440  VTAB 8: PRINT "NOTE: BLOWER INLET MUST BE BETWEEN 10 AND 170 (DEG)"

1450  INPUT "ENTER BLOWER INLET SETTING=";Y: IF Y < 10 OR Y > 170 THEN  GOTO
    1440
1460  GOTO 1500
1470  PRINT : PRINT : INPUT "ENTER NEW SET POINT TEMP=";X
```

```
1480  IF X < 30 OR X > 60 THEN  GOTO 1470
1490 Y = BLO
1500  LET Q = 0:Q =  - 65.176394 + (2.46541807 * X)
1510  LET Q = Q + ( - 0.0167582955 * X ^ 2) + ( - 1.80406 * Y)
1520  LET Q = Q + (0.0199352325 * Y ^ 2) + (.110026458 * X * Y)
1530  LET Q = Q + ( - 9.8819896E - 04 * X ^ 2 * Y) + ( - 1.18368066E - 03
      * X * Y ^ 2)
1540  LET Q = Q + (1.5608222E - 05 * X ^ 2 * Y ^ 2)
1550  IF Q < 4 OR Q > 96 THEN  GOTO 1570
1560  LET V = (0.1 * Q) - 5:OUT = 409.5 * (V + 5): GOTO 1630
1570  PRINT : PRINT "YOU HAVE CHOSEN DESIGN CONDITIONS WHICH"
1580  PRINT "ARE NOT PHYSICALLY REALIZABLE BECAUSE THE"
1590  PRINT "MAXIMUM HEATER OUTPUT IS 100 WATTS."
1600  GOSUB 1650
1610  IF  PEEK ( - 16384) < 127 THEN  GOTO 1600
1620  GET T$: GOTO 1400
1621  REM
1622  REM   THE FOLLOWING SUBROUTINES CLEAR/READY SCREEN FOR GRAPHS
1623  REM   AND DISPLAY MESSAGES TO HELP THE USER
1624  REM
1630  RETURN
1640  HGR : ROT= 0: SCALE= 1: & SCROLLSET: RETURN
1650  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE                          "
1660  RETURN
1670  VTAB 22: PRINT "PRESS ANY KEY TO RETURN TO MENU                    "
1680  RETURN
1690  VTAB 22: PRINT "PRESS ANY KEY TO CHANGE SET POINT TEMP.  "
1700  RETURN
1701  REM
1702  REM   THE FOLLOWING SUBROUTINES PROVIDE DIGITAL CONTROL.
1703  REM   THEY ARE PROPORTIONAL, P-I, P-I-D, RESPECTIVELY.
1704  REM
1710  LET MIST = (COMPAR - IN) * GAIN:OUT = OLD + MIST: RETURN
1720 KEVIN = COMPAR - IN
1730  LET MIST = GAIN * ((KEVIN - LAST) + ((TIME / TR) * KEVIN))
1740 PREV = LAST:LAST = KEVIN:OUT = OLD + MIST
1750  LET OLD = OUT: RETURN
1760 KEVIN = COMPAR - IN
1770  LET MIST = GAIN * ((KEVIN - LAST) + ((TIME / TR) * KEVIN))
1780 MIST = MIST + GAIN * ((TD / TIME) * (KEVIN - (2 * LAST) + PREV))
1790 PREV = LAST:LAST = KEVIN:OUT = OLD + MIST
1800  LET OLD = OUT: RETURN
```

```
10  HIMEM: 38395
20  LOMEM: 16384
30  DIM Y(2,260)
40  & SLOT# = 2
41  REM
42  REM   THIS PROGRAM IS TITLED "SERVO". IT ALLOWS EXAMINATION OF
43  REM   PROCESS UNDER DIGITAL CONTROL, SERVO RESPONSE. THE FIRST
44  REM   SECTION SETS MEMORY POINTERS, DIMENSIONS AN ARRAY, THEN
45  REM   ACTIVATES THE INTERFACE DEVICE.
46  REM
47  REM   THE NEXT SECTION OPENS A TEXT FILE TO RETREIVE THE CONTROL
48  REM   SYSTEM PARAMETERS AND LOAD A SHAPE TABLE NEEDED LATER.
49  REM
50  PRINT  CHR$ (4);"OPEN VARPASS"
60  PRINT  CHR$ (4);"READ VARPASS"
70  INPUT OLD: INPUT COMPAR: INPUT GAIN: INPUT TR: INPUT TD: INPUT K
80  INPUT OWE: INPUT TIME: PRINT  CHR$ (4);"CLOSE VARPASS"
90  PRINT  CHR$ (4);"BLOAD TIME LABELS,A6000"
100  POKE 232, PEEK (43634): POKE 233, PEEK (43635)
101  REM
102  REM   THIS NEXT SECTION ASKS THE USER IF HARDCOPY DESIRED
103  REM
110  HOME : INPUT "DO YOU WANT HARDCOPY (Y OR N)=";A$
120  IF A$ <  > "Y" AND A$ <  > "N" THEN  GOTO 110
130 PARM = 768:PUTPAGE = 769:BRINGPAGE = 800
140  VTAB 15: PRINT "AT THE BEEP DATA ACQUISITION BEGINS"
141  REM
142  REM   THE FOLLOWING SECTION BEGINS CONTROLLING AND EXAMINING
143  REM   THE PROCESS. IT IS ALSO LOOKING AT THE KEYBOARD FOR INPUT
144  REM   TO SIGNAL THE CHANGE IN SET POINT
145  REM
150  LET J = 0:P = 0:LAST = 0:PREV = 0:MIST = 0: GOSUB 1070: & BEEP ON
160  & CLRTIMER
170  & AIN,(TV) = W,(C#) = 0: & AIN,(TV) = IN,(C#) = 1
180  & AIN,(TV) = Y,(C#) = 3:Y(0,J) = W:Y(1,J) = IN:Y(2,J) = Y
190  LET J = J + 1: IF J = 261 THEN  GOTO 320
200  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 170
210  ON K GOSUB 1090,1100,1130: & AOUT,(DV) = OUT,(C#) = 0
220  IF  PEEK ( - 16384) > 127 THEN  GOTO 240
230  GOTO 160
240  HOME : GET T$:COMPAR = OWE: & AOUT,(DV) = COMPAR,(C#) = 1
250  & CLRTIMER
260  & AIN,(TV) = W,(C#) = 0: & AIN,(TV) = IN,(C#) = 1
270  & AIN,(TV) = Y,(C#) = 3:Y(0,J) = W:Y(1,J) = IN:Y(2,J) = Y
280  LET J = J + 1: IF J = 261 THEN  GOTO 320
290  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 260
300  ON K GOSUB 1090,1100,1130: & AOUT,(DV) = OUT,(C#) = 0
310  GOTO 250
320  & BEEP STOP :Q = (J * 103 / 1000) + 3 + (P * 120 / 1000)
321  REM
322  REM   THE FOLLOWING IS A SERIES OF ROUTINES. EACH ONE GENERATES
323  REM   A GRAPH TO THE USER. THE FIRST ROUTINE GENERATES A GRAPH
324  REM   OF THE WATTAGE OUTPUT TO THE PROCESS
325  REM
330  GOSUB 1000: POKE PARM,64: CALL BRINGPAGE: GOSUB 750
340  & PLTFMT = 2
350  FOR W = 0 TO 260
360  & NXTPLT = Y(0,W) / 32
```

```
370  NEXT W
380  GOSUB 1030
390  IF  PEEK ( - 16384) ) 127 THEN  GOTO 410
400  GOTO 390
410  IF A$ = "Y" THEN  GOSUB 1170
411  REM
412  REM   THE ROUTINE GENERATES A GRAPH OF PROCESS TEMPERATURE
413  REM   RESPONSE TO SET POINT CHANGE
414  REM
420  GET T$: GOSUB 1000: POKE PARM,96: CALL BRINGPAGE: GOSUB 750
430  FOR W = 0 TO 260
440  & NXTPLT = Y(1,W) / 32
450  NEXT W
460  GOSUB 1030
470  IF  PEEK ( - 16384) ) 127 THEN  GOTO 490
480  GOTO 470
490  IF A$ = "Y" THEN  GOSUB 1170
491  REM
492  REM   THIS ROUTINE CREATES A GRAPH OF THE SET POINT
493  REM
500  GET T$: GOSUB 1000: POKE PARM,128: CALL BRINGPAGE: GOSUB 750
510  FOR W = 0 TO 260
520  & NXTPLT = Y(2,W) / 32
530  NEXT W
540  GOSUB 1030
550  IF  PEEK ( - 16384) ) 127 THEN  GOTO 570
560  GOTO 550
570  IF A$ = "Y" THEN  GOSUB 1170
571  REM
572  REM   THIS ROUTINE GENERATES A GRAPH OF BOTH PROCESS AND SET
573  REM   POINT TEMPERATURES SIMULTANEOUSLY
574  REM
580  GET T$: GOSUB 1000: POKE PARM,160: CALL BRINGPAGE: GOSUB 750
590  & PLTFMT = 3,5
600  FOR W = 0 TO 260
610  & NXTPLT = Y(1,W) / 32
620  & NXTPLT = Y(2,W) / 32
630  NEXT W
640  & FMTDFLT
650  GOSUB 1030
660  IF  PEEK ( - 16384) ) 127 THEN  GOTO 680
670  GOTO 660
680  IF A$ = "Y" THEN  GOSUB 1170
690  TEXT
691  REM
692  REM   AFTER THE GRAPHS HAVE BEEN OUTPUT TO THE USER THE SYSTEM
693  REM   ASKS IF THE USER WANTS TO CONTINUE ON?
694  REM
700  HOME : INPUT "DO YOU WISH TO CONTINUE (Y OR N)";A$
710  IF A$ = "N" THEN  END
720  IF A$ = "Y" THEN  GOTO 740
730  GOTO 700
740 U$ = "CONTROL PROGRAM": PRINT  CHR$ (4);"RUN "U$
741  REM
742  REM   THE FOLLOWING IS A MAIN SUBROUTINE CALLED TO LABEL
743  REM   THE GRAPH WITH A TIME BASE USING A SHAPE TABLE
744  REM
750  LET N = (Q / 7) * 2
760  IF N (  )  INT (N) THEN  GOTO 780
770  LET N = N + .001
780  LET X = 80:Y = 156: GOSUB 820
790  LET N = N * 2:X = 150: GOSUB 820
800  LET N = N + (N / 2):X = 230: GOSUB 820
810  RETURN
820  LET D = N + .5
830  LET D$ =  STR$ (D)
```

```
840  FOR J = 1 TO 15
850  IF  MID$ (D$,J,1) = "." THEN  GOTO 870
860  NEXT J
870  LET J = J - 2
880  IF J < > 0 THEN  GOTO 910
890  LET B = 10: GOSUB 990
900  GOTO 920
910  GOSUB 950
920  LET J = J + 1
930  GOSUB 950
940  RETURN
950  LET B$ =  MID$ (D$,J,1)
960  LET B =  VAL (B$)
970  IF B < > 0 THEN  GOTO 990
980  LET B = 10
990  DRAW B AT X,Y:X = X + 8: RETURN
991  REM
992  REM   THE FOLLOWING SUBROUTINES READY THE SCREEN FOR A NEW
993  REM   GRAPH, AND DISPLAY MESSAGES TO THE USER
994  REM
1000  HGR : HCOLOR= 3: ROT= 0: SCALE= 1
1010  & SCROLLSET
1020  RETURN
1030  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE                    "
1040  RETURN
1050  VTAB 22: PRINT "PRESS ANY KEY TO RETURN TO MENU"
1060  RETURN
1070  VTAB 22: PRINT "PRESS ANY KEY TO CHANGE SET POINT TEMP"
1080  RETURN
1081  REM
1082  REM   THE FOLLOWING SUBROUTINES PROVIDE DIGITAL CONTROL
1083  REM   THEY ARE PROPORTIONAL, P-I, P-I-D, RESPECTIVELY.
1084  REM
1090  LET MIST = (COMPAR - IN) * GAIN:OUT = OLD + MIST: RETURN
1100 KEVIN = COMPAR - IN:P = P + 1
1110  LET MIST = GAIN * ((KEVIN - LAST) + ((TIME / TR) * KEVIN))
1120 PREV = LAST:LAST = KEVIN:OUT = OLD + MIST:OLD = OUT: RETURN
1130 KEVIN = COMPAR - IN:P = P + 1
1140  LET MIST = GAIN * ((KEVIN - LAST) + ((TIME / TR) * KEVIN))
1150 MIST = MIST + GAIN * ((TD / TIME) * (KEVIN - (2 * LAST) + PREV))
1160 PREV = LAST:LAST = KEVIN:OUT = OLD + MIST:OLD = OUT: RETURN
1170  HOME : PRINT  CHR$ (4);"PR#1": PRINT  CHR$ (9);"G"
1180  PRINT  CHR$ (9);"e": RETURN
```

```
10  HIMEM: 38395
20  LOMEM: 16384
30  DIM Y(2,260)
40  & SLOT# = 2
41  REM
42  REM   THIS PROGRAM IS TITLED "REGULATOR". IT ALLOWS EXAMINATION
43  REM   OF GRAPHS WITH LABELED TIME BASE, REGULATOR RESPONSE.
44  REM   THE FIRST SECTION SETS MEMORY POINTERS, DIMENSIONS AN ARRAY,
45  REM   AND ACTIVATES INTERFACE DEVICE, RESPECTIVELY.
46  REM
47  REM   THE NEXT SECTION OPENS A TEXT FILE TO RETREIVE THE CONTROL
48  REM   PARAMETERS AND THEN LOADS A SHAPE TABLE TO LABEL THE GRAPH.
49  REM
50  PRINT  CHR$ (4);"OPEN VARPASS"
60  PRINT  CHR$ (4);"READ VARPASS"
70  INPUT OLD: INPUT COMPAR: INPUT GAIN: INPUT TR: INPUT TD: INPUT K
80  INPUT OWE: INPUT TIME: PRINT  CHR$ (4);"CLOSE VARPASS"
90  PRINT  CHR$ (4);"BLOAD TIME LABELS,A6000"
100  POKE 232, PEEK (43634): POKE 233, PEEK (43635)
101  REM
102  REM   THIS SECTION ASKS USER IF HARDCOPY DESIRED
103  REM
110  HOME : INPUT "DO YOU WANT HARDCOPY (Y OR N)=";A$
120  IF A$ <  > "Y" AND A$ <  > "N" THEN  GOTO 110
130 PARM = 768:PUTPAGE = 769:BRINGPAGE = 800
140  VTAB 15: PRINT "AT THE BEEP DATA ACQUISITION BEGINS"
141  REM
142  REM   THIS SECTION BEGINS CONTROLLING AND EXAMINING THE
143  REM   PROCESS. THEN IT COMPUTES THE TOTAL TIME THE PROCESS
144  REM   WAS LOOKED AT.
145  REM
150  LET J = 0:P = 0:LAST = 0:MIST = 0:PREV = 0: & BEEP ON
160  & CLRTIMER
170  & AIN,(TV) = W,(C#) = 0: & AIN,(TV) = IN,(C#) = 1
180  & AIN,(TV) = Y,(C#) = 3:Y(0,J) = W:Y(1,J) = IN:Y(2,J) = Y
190  LET J = J + 1: IF J = 261 THEN  GOTO 230
200  & TIMERIN,(TV) = SEC: IF SEC < TIME THEN  GOTO 170
210  ON K GOSUB 980,990,1020: & AOUT,(DV) = OUT,(C#) = 0
220  GOTO 160
230  & BEEP STOP :Q = (J * 103 / 1000) + 3 + (P * 120 / 1000)
231  REM
232  REM   THE FOLLOWING IS A SERIES OF ROUTINES. EACH ROUTINE
233  REM   GENERATES A GRAPH. THE FIRST ROUTINE GENERATES A
234  REM   GRAPH OF THE WATTAGE OUTPUT TO THE PROCESS.
235  REM
240  GOSUB 910: POKE PARM,64: CALL BRINGPAGE: GOSUB 660
250  & PLTFMT = 2
260  FOR W = 0 TO 260
270  & NXTPLT = Y(0,W) / 32
280  NEXT W
290  GOSUB 940
300  IF  PEEK ( - 16384) > 127 THEN  GOTO 320
310  GOTO 300
320  IF A$ = "Y" THEN  GOSUB 1060
321  REM
322  REM   THE NEXT ROUTINE GENERATES A GRAPH OF THE PROCESS
323  REM   TEMPERATURE
324  REM
330  GET T$: GOSUB 910: POKE PARM.96: CALL BRINGPAGE: GOSUB 660
```

```
340  FOR W = 0 TO 260
350  & NXTPLT = Y(1,W) / 32
360  NEXT W
370  GOSUB 940
380  IF  PEEK ( - 16384) > 127 THEN  GOTO 400
390  GOTO 380
400  IF A$ = "Y" THEN  GOSUB 1060
401  REM
402  REM   THIS ROUTINE GENERATES A GRAPH OF THE SET POINT TEMPERATURE
403  REM
410  GET T$: GOSUB 910: POKE PARM,128: CALL BRINGPAGE: GOSUB 660
420  FOR W = 0 TO 260
430  & NXTPLT = Y(2,W) / 32
440  NEXT W
450  GOSUB 940
460  IF  PEEK ( - 16384) > 127 THEN  GOTO 480
470  GOTO 460
480  IF A$ = "Y" THEN  GOSUB 1060
481  REM
482  REM   THIS ROUTINE GENERATES A GRAPH OF BOTH THE PROCESS AND
483  REM   SET POINT TEMPERATURES SIMULTANEOUSLY.
484  REM
490  GET T$: GOSUB 910: POKE PARM,160: CALL BRINGPAGE: GOSUB 660
500  & PLTFMT = 3,5
510  FOR W = 0 TO 260
520  & NXTPLT = Y(1,W) / 32
530  & NXTPLT = Y(2,W) / 32
540  NEXT W
550  & FMTDFLT
560  GOSUB 940
570  IF  PEEK ( - 16384) > 127 THEN  GOTO 590
580  GOTO 570
590  IF A$ = "Y" THEN  GOSUB 1060
600  TEXT
601  REM
602  REM   AFTER THE GRAPHS HAVE BEEN OUTPUT TO THE USER THE SYSTEM
603  REM   ASKS THE USER IF HE WANTS TO DO ANOTHER EXAMINATION
604  REM
610  HOME : INPUT "DO YOU WISH TO CONTINUE (Y OR N)";A$
620  IF A$ = "N" THEN  END
630  IF A$ = "Y" THEN  GOTO 650
640  GOTO 610
650 U$ = "CONTROL PROGRAM": PRINT  CHR$ (4);"RUN "U$
651  REM
652  REM   THE FOLLOWING IS A SUBROUTINE THAT LABELS THE GRAPHS
653  REM   WITH A TIME BASE USING THE SHAPE TABLE LOADED AT THE
654  REM   BEGINNING OF PROGRAM
655  REM
660  LET N = (Q / 7) * 2
670  IF N <  >  INT (N) THEN  GOTO 690
680  LET N = N + .001
690  LET X = 80:Y = 156: GOSUB 730
700  LET N = N * 2:X = 150: GOSUB 730
710  LET N = N + (N / 2):X = 230: GOSUB 730
720  RETURN
730  LET D = N + .5
740  LET D$ =  STR$ (D)
750  FOR J = 1 TO 15
760  IF  MID$ (D$,J,1) = "." THEN  GOTO 780
770  NEXT J
780  LET J = J - 2
790  IF J <  > 0 THEN  GOTO 820
800  LET B = 10: GOSUB 900
810  GOTO 830
820  GOSUB 860
830  LET J = J + 1
```

```
840  GOSUB 860
850  RETURN
860  LET B$ =  MID$ (D$,J,1)
870  LET B =  VAL (B$)
880  IF B <  > 0 THEN  GOTO 900
890  LET B = 10
900  DRAW B AT X,Y:X = X + 8: RETURN
901  REM
902  REM   THE FOLLOWING ARE SUBROUTINES TO READY SCREEN FOR NEW
903  REM   GRAPH AND DISPLAY MESSAGES TO USER.
904  REM
910  HGR : HCOLOR= 3: ROT= 0: SCALE= 1
920  & SCROLLSET
930  RETURN
940  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE"
950  RETURN
960  VTAB 22: PRINT "PRESS ANY KEY TO RETURN TO MENU"
970  RETURN
971  REM
972  REM   THE FOLLOWING SUBROUTINES PROVIDE DIGITAL CONTROL.
973  REM   THEY ARE PROPORTIONAL, P-I, P-I-D CONTROL, RESPECTIVELY.
974  REM
980  LET MIST = (COMPAR - IN) * GAIN:OUT = OLD + MIST: RETURN
990 KEVIN = COMPAR - IN:P = P + 1
1000  LET MIST = GAIN * ((KEVIN - LAST) + ((TIME / TR) * KEVIN))
1010 PREV = LAST:LAST = KEVIN:OUT = OLD + MIST:OLD = OUT: RETURN
1020 KEVIN = COMPAR - IN:P = P + 1
1030  LET MIST = GAIN * ((KEVIN - LAST) + ((TIME / TR) * KEVIN))
1040 MIST = MIST + GAIN * ((TD / TIME) * (KEVIN - (2 * LAST) + PREV))
1050 PREV = LAST:LAST = KEVIN:OUT = OLD + MIST:OLD = OUT: RETURN
1060  HOME : PRINT  CHR$ (4);"PR#1": PRINT  CHR$ (9);"G"
1070  PRINT  CHR$ (9);"e": RETURN
```

```
10  & SLOT# = 2
20  HIMEM: 38395
30  LOMEM: 16384
31  REM
32  REM   THIS PROGRAM IS TITLED "BANGBANG". THIS PROGRAM ALLOWS
33  REM   EXAMINATION OF THE PROCESS UNDER ON-OFF CONTROL, SERVO
34  REM   OR REGULATOR RESPONSE. REAL-TIME MONITORING ONLY.
35  REM
36  REM  THE FIRST SECTION SETS MEMORY POINTERS AND ACTIVATES THE
37  REM  INTERFACE DEVICE. THEN THE PROGRAM ASKS THE USER FOR DESIRED
38  REM  SET POINT AND BLOWER INLET
39  REM
40 PARM = 768:PUTPAGE = 769:BRINGPAGE = 800
50  GOSUB 940:OLD = OUT:COMPAR = 136 * (X - 30):BLO = Y
60  & AOUT, (DV) = OUT, (C#) = 0
70  & AOUT, (DV) = COMPAR, (C#) = 1
71  REM
72  REM   THE FOLLOWING SECTION ASKS FOR DESIRED DEAD RANGE AND TYPE
73  REM   OF RESPONSE (SERVO OR REGULATOR)
74  REM
80  HOME : INPUT "ENTER DEAD RANGE (C)=";DR:DR = 136 * DR
90  HOME : PRINT "AT THIS POINT YOU HAVE FOLLOWING CHOICES:"
100  PRINT : PRINT "  1. REGULATOR CONTROL (RESPONSE TO LOAD VARIABLE CHA
    NGE)"
110  PRINT "  2. SERVO CONTROL (RESPONSE TO SET POINT CHANGE)"
120  PRINT "  3. QUIT"
130  PRINT : INPUT "ENTER CHOICE NOW=";Z
140  IF Z = 1 THEN  GOTO 190
150  IF Z = 2 THEN  GOTO 180
160  IF Z = 3 THEN  GOTO 1270
170  GOTO 90
180  GOSUB 1010:OWE = 136 * (X - 30)
181  REM
182  REM   THIS SECTION ASKS THE USER WHICH OF THE FOUR AVAILABLE
183  REM   GRAPHS DOES HE WISH TO SEE?
184  REM
190  HOME : PRINT "YOU HAVE FOLLOWING OPTIONS AT THIS POINT:"
200  PRINT : PRINT "  1. REAL-TIME DISPLAY OF WATTAGE"
210  PRINT "  2. REAL-TIME DISPLAY OF AIR TEMPERATURE"
220  PRINT "  3. REAL-TIME DISPLAY OF SET POINT TEMPERATURE"
230  PRINT "  4. REAL-TIME DISPLAY OF BOTH TEMPERATURES"
240  PRINT : INPUT "ENTER CHOICE NOW=";Y
250  IF Y < 1 OR Y > 4 THEN  GOTO 190
260  IF Z = 1 AND Y = 1 THEN  GOTO 340
270  IF Z = 2 AND Y = 1 THEN  GOTO 350
280  IF Z = 1 AND Y = 2 THEN  GOTO 360
290  IF Z = 2 AND Y = 2 THEN  GOTO 370
300  IF Z = 1 AND Y = 3 THEN  GOTO 380
310  IF Z = 2 AND Y = 3 THEN  GOTO 390
320  IF Z = 1 AND Y = 4 THEN  GOSUB 670: GOTO 50
330  IF Z = 2 AND Y = 4 THEN  GOSUB 770: GOTO 50
340  LET Z = 0:T = 2: GOSUB 400: GOTO 50
350  LET Z = 0:T = 2: GOSUB 500: GOTO 50
360  LET Z = 1:T = 3: GOSUB 400: GOTO 50
370  LET Z = 1:T = 3: GOSUB 500: GOTO 50
380  LET Z = 3:T = 4: GOSUB 400: GOTO 50
390  LET Z = 3:T = 4: GOSUB 500: GOTO 50
391  REM
```

```
392  REM   THE FOLLOWING IS A SERIES OF SUBROUTINES ALLOWING
393  REM   EXAMINATION OF THE PROCESS. THE FIRST ONE EXAMINES
394  REM   WATTAGE, PROCESS OR SET POINT TEMPERATURE, REGULATOR
395  REM   RESPONSE
396  REM
400  GOSUB 1200: GOSUB 1230: POKE PARM,T * 32: CALL BRINGPAGE
410  & PLTFMT = 5
420  & AIN, (TV) = IN, (C#) = 1
430  & AIN, (TV) = W, (C#) = Z, (GA)
440  IF IN < COMPAR - DR THEN OUT = 4095
450  IF IN > COMPAR + DR THEN OUT = 0
460  & AOUT, (DV) = OUT, (C#) = 0
470  IF  PEEK ( - 16384) < 127 THEN  GOTO 420
480  & AOUT, (DV) = OLD, (C#) = 0
490  GET T$: TEXT : & FMTDFLT: RETURN
491  REM
492  REM   THIS SUBROUTINE EXAMINES WATTAGE, PROCESS OR SET POINT
493  REM   TEMPERATURE , SERVO RESPONSE.
494  REM
500  GOSUB 1250: GOSUB 1200: POKE PARM,T * 32: CALL BRINGPAGE
510  & PLTFMT = 2
520  & AIN, (TV) = W, (C#) = Z, (GA)
530  & AIN, (TV) = IN, (C#) = 1
540  IF IN < COMPAR - DR THEN OUT = 4095
550  IF IN > COMPAR + DR THEN OUT = 0
560  & AOUT, (DV) = OUT, (C#) = 0
570  IF  PEEK ( - 16384) < 127 THEN  GOTO 520
580  GET T$: GOSUB 1230: & AOUT, (DV) = OWE, (C#) = 1
590  & AIN, (TV) = W, (C#) = Z, (GA)
600  & AIN, (TV) = IN, (C#) = 1
610  IF IN < OWE - DR THEN OUT = 4095
620  IF IN > OWE + DR THEN OUT = 0
630  & AOUT, (DV) = OUT, (C#) = 0
640  IF  PEEK ( - 16384) < 127 THEN  GOTO 590
650  & AOUT, (DV) = OLD, (C#) = 0
660  GET T$: TEXT : & FMTDFLT: RETURN
661  REM
662  REM   THIS ROUTINE EXAMINES BOTH PROCESS AND SET POINT
663  REM   TEMPERATURE, REGULATOR RESPONSE.
664  REM
670  GOSUB 1200: GOSUB 1230: POKE PARM,160: CALL BRINGPAGE
680  & PLTFMT = 1,5
690  & AIN, (TV) = IN, (C#) = 1, (GA)
700  & AIN, (TV) = W, (C#) = 3, (GA)
710  IF IN < COMPAR - DR THEN OUT = 4095
720  IF IN > COMPAR + DR THEN OUT = 0
730  & AOUT, (DV) = OUT, (C#) = 0
740  IF  PEEK ( - 16384) < 127 THEN  GOTO 690
750  GET T$: TEXT : & FMTDFLT: & AOUT, (DV) = OLD, (C#) = 0
760  RETURN
761  REM
762  REM   THE FOLLOWING ROUTINE GENERATES A GRAPH OF BOTH PROCESS
763  REM   AND SET POINT TEMPERATURES, SERVO RESPONSE
764  REM
770  GOSUB 1250: GOSUB 1200: POKE PARM,160: CALL BRINGPAGE
780  & PLTFMT = 3,5
790  & AIN, (TV) = IN, (C#) = 1, (GA)
800  & AIN, (TV) = W, (C#) = 3, (GA)
810  IF IN < COMPAR - DR THEN OUT = 4095
820  IF IN > COMPAR + DR THEN OUT = 0
830  & AOUT, (DV) = OUT, (C#) = 0
840  IF  PEEK ( - 16384) < 127 THEN  GOTO 790
850  GET T$: GOSUB 1230: & AOUT, (DV) = OWE, (C#) = 1
860  & AIN, (TV) = IN, (C#) = 1, (GA)
870  & AIN, (TV) = W, (C#) = 3, (GA)
880  IF IN < OWE - DR THEN OUT = 4095
```

```
890  IF IN > OWE + DR THEN OUT = 0
900  & AOUT,(DV) = OUT,(C#) = 0
910  IF  PEEK ( - 16384) < 127 THEN  GOTO 860
920  GET T$: TEXT : & FMTDFLT: & AOUT,(DV) = OLD,(C#) = 0
930  RETURN
931  REM
932  REM   THE FOLLOWING SUBROUTINE INPUTS DESIRED SET POINT AND
933  REM   DESIRED BLOWER INLET AND THEN SOLVES A 9 TERM REGRESSION
934  REM   EQUATION TO SOLVE FOR DESIGN WATTAGE. IF UNMAINTAINABLE
935  REM   CONDITIONS ENTERED, MUST BE REENTERED.
936  REM
940  HOME : PRINT "AT THIS POINT YOU NEED TO ENTER DESIGN TEMP"
950  PRINT "AND DESIGN BLOWER INLET."
960  VTAB 4: PRINT "NOTE: TEMP MUST BE BETWEEN 30 AND 60 (C)"
970  INPUT "ENTER DESIGN TEMP=";X: IF X < 30 OR X > 60 THEN  GOTO 960
980  VTAB 8: PRINT "NOTE: BLOWER INLET MUST BE BETWEEN 10 AND 170 (DEG)"
990  INPUT "ENTER BLOWER INLET=";Y: IF Y < 10 OR Y > 170 THEN  GOTO 980
1000  GOTO 1040
1010  HOME : INPUT "ENTER NEW SET POINT TEMPERATURE=";X
1020  IF X < 30 OR X > 60 THEN  GOTO 1010
1030 Y = BLO
1040  LET Q = 0:Q =  - 65.176394 + (2.46541807 * X)
1050  LET Q = Q + ( - 0.0167582955 * X ^ 2) + ( - 1.80406 * Y)
1060  LET Q = Q + (0.0199352325 * Y ^ 2) + (.110026458 * X * Y)
1070  LET Q = Q + ( - 9.88198962E - 04 * X ^ 2 * Y) + ( - 1.18368066E - 0
     3 * X * Y ^ 2)
1080  LET Q = Q + (1.56082222E - 05 * X ^ 2 * Y ^ 2)
1090  IF Q < 4 OR Q > 96 THEN  GOTO 1130
1100  LET V = (0.1 * Q) - 5
1110  LET OUT = 409.5 * (V + 5)
1120  GOTO 1190
1130  PRINT : PRINT "YOU HAVE CHOSEN DESIGN CONDITIONS WHICH"
1140  PRINT "ARE NOT PHYSICALLY REALIZABLE BECAUSE THE"
1150  PRINT "MAXIMUM HEATER OUTPUT IS 100 WATTS."
1160  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE"
1170  IF  PEEK ( - 16384) < 127 THEN  GOTO 1160
1180  GET T$: GOTO 940
1190  RETURN
1191  REM
1192  REM   THE FOLLOWING SUBROUTINE READIES THE SCREEN FOR A NEW
1193  REM   GRAPH TO BE CREATED.
1194  REM
1200  HGR : ROT= 0: SCALE= 1: & SCROLLSET: RETURN
1201  REM
1202  REM   THE FOLLOWING SUBROUTINES OUTPUT MESSAGES TO THE USER
1203  REM
1210  VTAB 22: PRINT "PRESS ANY KEY TO CONTINUE                    "
1220  RETURN
1230  VTAB 22: PRINT "PRESS ANY KEY TO RETURN TO MENU              "
1240  RETURN
1250  VTAB 22: PRINT "PRESS ANY KEY TO CHANGE SET POINT TEMP       "
1260  RETURN
1261  REM
1262  REM   THE FOLLOWING SECTION DETERMINES WHETHER TO TERMINATE
1263  REM   OR CONTINUE ON TO ANOTHER PROGRAM
1264  REM
1270  HOME : INPUT "DO YOU WISH TO CONTINUE (Y OR N)=";A$
1280  IF A$ = "Y" THEN  GOTO 1310
1290  IF A$ = "N" THEN  END
1300  GOTO 1270
1310  LET U$ = "CONTROL PROGRAM"
1320  PRINT  CHR$ (4);"RUN "U$
```

APPENDIX D - DATA FOR REGRESSION EQUATION

PROCESS TEMPERATURE (CELSIUS)

BLOWER INLET (DEGREES)	30	35	40	45	50	55	60
10	0	12	18.5	22	28	32	36
20	3	17.5	27	33	40	45	50
30	10	23	34	42	52	58	66
40	14	27	42	52	64	74	80
50	20	35	52	64	78	89	96
60	24	42	58	74	92	**	**
70	29	48	67	85	**	**	**
80	36	52	72	95	**	**	**
90	36	55	80	**	**	**	**
100	41	57	83	**	**	**	**
110	41	61	89	**	**	**	**
120	44	66	94	**	**	**	**
130	48	70	**	**	**	**	**
140	51	72	**	**	**	**	**
150	51	74	**	**	**	**	**
160	51	74	**	**	**	**	**
170	51	74	**	**	**	**	**

WATTAGE REQUIRED TO MAINTAIN GIVEN CONDITIONS

**-represents conditions requiring over 100 watts to maintain
i.e. unmaintainable conditions