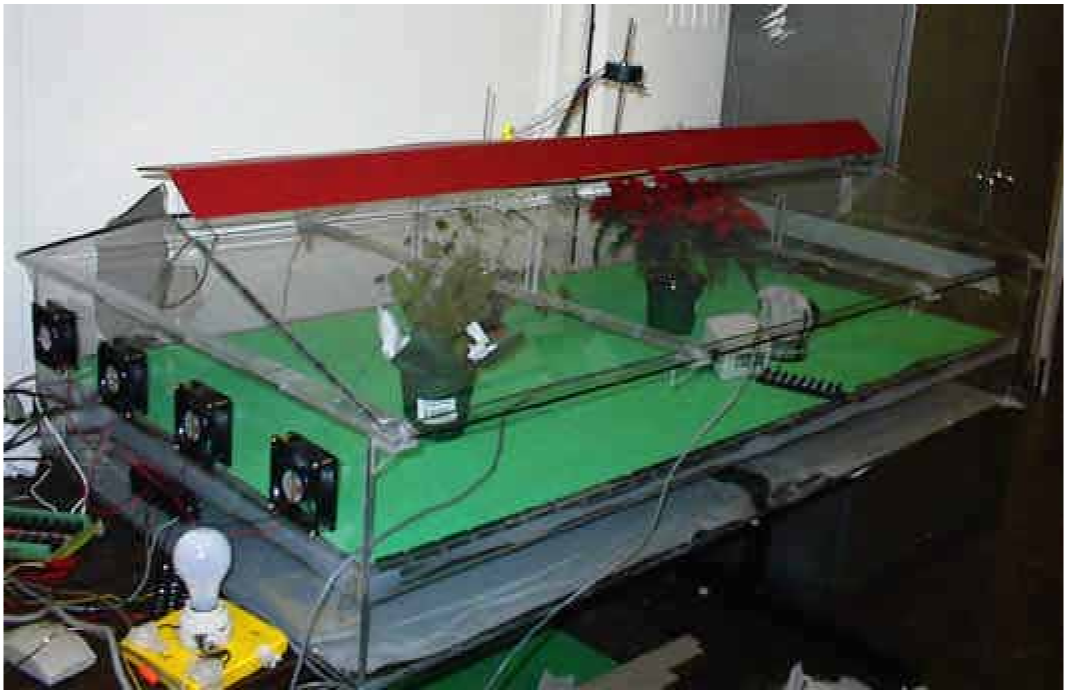
Model for Testing Greenhouse Ventilation Logic



Submitted by:
Amber Hasche

Iowa State University
B.S. AE, May 2003

Dr. Steven J. Hoff
Dr. Carl J. Bern
Advisors

Contestant

Student Branch Faculty Advisor

Date

ORIGIN OF PAPER

The following paper is a final report of a design project completed in the fall semester of 2002 for AE 504, Instrumentation for Agricultural and Biosystems, at Iowa State University. This project was completed by Amber Hasche, an Agricultural Engineering undergraduate student at Iowa State University under the supervision of Dr. Steven Hoff, Associate Professor in Agricultural and Biosystems Engineering. The subject of this paper was suggested by Dr. Hoff to facilitate the development of systems for testing greenhouse ventilation.

ABSTRACT

Proper ventilation in a greenhouse is essential for economical production of crops. Improper ventilation systems can cost businesses using greenhouses through loss of product, increased need for expensive sprays, and wasted energy. For this reason, testing ventilation logic with a model is more efficient as it prevents expensive losses incurred by ventilation mistakes made on the greenhouse site. In this project, a model greenhouse building was constructed and sensors and controllers installed to simulate a common gable greenhouse. A sample logic program was written to sense environmental variables and control the ventilation system to maintain a desirable internal environment. The model has been found to be bery useful in testing different ventilation logic scenarios.

INTRODUCTION

The idea of producing fruit and flowers out of season in a glass building was put into practice in the Netherlands in the 17th century. The glass covering of the building allowed light to enter and provided insulation to create a climate within the structure that favored the growth, bloom, and fruiting of plants regardless of the temperature outside. The structure was very expensive due to the amount of glass used in the construction, so the lilacs grown were sold mostly to the wealthy nobility of Europe (Nelson 2003). In the 18th century, the first greenhouse in the United States, also covered in glass, was built near New York City to provide produce to the wealthy residents of that area (Nelson 2003). The purpose of these greenhouses was commercial production only for the rich. At this time, very few people owned private greenhouses, even the very wealthy. Since then, due to scientific advances made during the Second World War, such as the development of plastic film greenhouse covering and inexpensive plastic pots, the greenhouse industry has seen a significant boom. Greenhouses became simpler and less expensive to construct. Plastic pots are much easier to manufacture and work with than the traditional clay pots they replaced. Now greenhouse goods and greenhouse buildings formerly available only to the rich are available to all people.

Greenhouses can be found in all different climates on every continent, even Antarctica! Obviously, greenhouse operation will vary from climate to climate, but the main design and intent remain the same. In the United States, the definition of a greenhouse is a structure sufficiently tall to permit a person to stand within, covered with a transparent material to admit natural light for plant growth, that has supplemental heating in cooler climates (Nelson 2003). The sole purpose of a greenhouse is to "provide and maintain the environment that will result in optimum crop production or maximum profit" (Aldrich and Bartok 1994).

Greenhouse crops

Greenhouse crops also differ from one region to another in the world. Complexes of greenhouses in Colombia are used to grow carnations for world export (Nelson 2003) while greenhouses in Antarctica are used to grow fruit and vegetables for the researchers working in the area (Fosdick 2003). Many crops are grown within greenhouses such as cut flowers, bedding plants, foliage plants, greenery for cut flower arrangements, and ornamentals while some greenhouses are used as propagation buildings or conservatories. The most commonly grown greenhouse crop is garden bedding plants totaling 50% of the United States floriculture production while the second most commonly produced is flowering potted plants, such as poinsettias and mums, which totals 19% (USDA 2002). These various crops require different environments ranging from tropical to desert to temperate.

The United States is the leading producer of floriculture products in the world with an annual wholesale total of $4.78 Billion. The greenhouses of this industry cover 86 million square meters throughout the country (USDA 2002). The majority of this production is in California and Florida because of their suitable climate. In Iowa the floriculture industry sells $47,546,000 per year in wholesale products and has a total of 650,000 square meters of greenhouse space in production (USDA 2002). Although the floriculture business is really increasing worldwide, Iowa is still considered a deficit producer for all of the floriculture crops tracked by the USDA. This means that more products are purchased within the state than are produced, so they are brought in from other states. With increased education and the realization of the potential of this market, soon many new greenhouse businesses will be started throughout the state. This will lead to the construction of many new greenhouse facilities. In order to compete with the greenhouses located in milder climates, new greenhouses in Iowa will need to be more energy efficient during the extreme climate conditions.

Greenhouse environment management

The main way to create an environment within a greenhouse is through ventilation. Ventilation is replacing inside air with outside air (Buffington *et al.* 1987). Three main purposes for ventilating a greenhouse exist. The first is to reduce high temperatures in the summer due to trapping of solar radiation by the greenhouse glazing. The second purpose is to reduce the amount of humidity in a greenhouse especially in the winter. Molds, rots, and many other plant diseases thrive in relative humidities above 90%. Since reducing the amount of plant matter lost to disease has a large impact on increasing profits, a good ventilation system can save money. The third purpose for ventilation is to provide uniform airflow through the greenhouse. This aids in gas mixing and reduces plant stress. A fresh air supply is beneficial to plants because it assures an adequate level of carbon dioxide for photosynthesis and reduces the amount of condensation on plants which also reduces plant disease frequency.

Ventilation systems in greenhouses can involve a complicated combination of natural air currents and mechanical systems. See Figure 1 for pictures of modern greenhouse ventilation system components. Natural ventilation uses a combination of roof vents on the ridgeline and side vents (Figure 1a, b, c). Warm air rises due to convective currents and wind currents and escapes through roof vents while cool air enters through side vents (West Virginia University Extension 1996). Mechanical ventilation uses exhaust fans mounted on one end wall of the greenhouse to pull air into the greenhouse through side vents at the other end (Figure 1d). The side vents can be motorized so that the inlet area is variable. Correspondingly, the fan ventilation rate should be variable also. This may be accomplished by using either variable speed fans or a bank of single speed fans that are controlled individually. Components of natural and mechanical ventilation systems can be combined, but can oppose each other instead of combining advantages if improperly used together (West Virginia University Extension 1996).

Greenhouse design has changed little in recent history. Most greenhouses built today come as kits and are not individually designed. Therefore, even though an engineer designed a greenhouse, it will not be intended for a particular site and crop. In addition to this, the technology of control of the greenhouse ventilation has recently seen many advances. In the beginning, the ventilation was controlled manually, requiring someone to be present in the greenhouse at all times. After that, people used thermostats to turn on appliances such as heaters and fans. The problem with thermostats is that they need to be constantly calibrated to make sure that the heating and cooling system thermostats are synchronized. If they are not, the heating and cooling system could possibly be on at the same time wasting energy. Within the past 10 years, step controllers, micro controllers, and computer control systems have started to replace thermostats. These systems offer a greater potential for controlling the environment at optimum conditions because of the large number and types of ventilation components that can be controlled (Nelson 2003). With this new technology, testing must be done to determine the correct ventilation logic programs for different crops and areas that will provide the optimum conditions in the most efficient manner. An individual greenhouse manager cannot afford to try to find the best ventilation logic for their production greenhouse because any mistakes become very costly. The advantages of the recent technological advances will go unrealized unless someone can test ventilation logic programs in an inexpensive and effective way.

OBJECTIVE

The purpose of this project was to create a model greenhouse that could be used to scientifically develop correct greenhouse ventilation. The model provides a way to test various ventilation logic systems for energy efficiency and for ability to appropriately respond to changes in the external environment. With this tool, a ventilation logic program can be refined before being applied to a full size greenhouse, thus avoiding expensive losses due to inadequacies of the ventilation system.

MATERIALS AND METHODS

Greenhouse Model

Building a physical model for this project instead of using a software simulation program was important for two reasons. The first reason was that even with the most complex combination of formulas, a simulation could never take into account all of the factors affecting environmental changes and predict the reactions of the system. Second, operation of the model building allows the use of small linear actuators and fans that are controlled in the same way as their full-size counterparts implemented in a greenhouse building.

The greenhouse model building was constructed of Plexiglas (Figure 2). The model is approximately 150 cm long, 76 cm wide, and 29 cm from the false floor to the peak (Figure 3). This is approximately a 1:20 model of a full-size greenhouse. The model can represent either a hog confinement building or a greenhouse with a few minor adjustments. The adjustments required for greenhouse modeling include a false floor made from tag board laid over the slotted floor of the confinement and a ridge vent in the roof of the building. A red piece of plastic was heated in the middle and bent to match the roof angle to cover the vent opening. The fans and end wall vent are utilized both when the building is simulating a greenhouse and when it is simulating hog confinement building. In this project, only the greenhouse control is considered.

Model Instrumentation

The most common environmental conditions that are controlled in a greenhouse include air temperature, air relative humidity, and carbon dioxide concentration. After the variables are sensed, decisions are made by the ventilation logic to operate fans, vents, heaters, humidifiers, and air coolers accordingly. In this model, air temperature and relative humidity are sensed and fans, vents, humidifier, and heater are controlled.

Data Acquisition System

In this system, a Model PCL 711B PC-MultiLab Card from Advantech Co., Ltd. was used for data acquisition and control (Figure 4). It features 12 bit resolution for A/D conversion, accepts 8 single-ended analog inputs, has onboard 16-bit digital input and digital output, and can be programmed in QUICK BASIC (Advantech Co., Ltd 1995). The analog sensor temperature and relative humidity data were inputted through a breakout box connected to the CN1 location on the card (Figure 5). The PCLD-785 relay output board (Figure 6) connected to the card at the CN3 location activated all of the controlled components of the ventilation system. The two sides of the relay board have different addresses on the card, so the program can call them at different times. In this application, all of the devices that ran continuously once activated were attached to the left hand side of the board and devices that were only activated for shorter specified periods of time were attached to the right hand side. This convention proved to be useful because it was not necessary to interrupt the action of the continuously-running components to control another component for a short period of time.

Sensors

The two environmental variables sensed in the greenhouse are temperature and relative humidity. Both variables are measured using an Omega HX93V temperature and relative humidity transmitter (Omega Engineering Inc. 2003, Figure 7). The HX93V uses a thin-film polymer capacitor to sense relative humidity and a platinum 100-ohm RTD to sense temperature (Omega Engineering Inc. 2003). A linear voltage signal proportional to the measured humidity or temperature is sent to the computer. Within the software, the voltage signal is converted to relative humidity using a calibration curve supplied by Omega. Having a temperature and relative humidity sensor in the same transmitter is very convenient because the relative humidity measurements must be temperature compensated (Omega Engineering Inc. 2003).

This project also uses two AD592 temperature sensors from Analog Devices, Inc. These sensors output a variable current corresponding to temperature. The signal is conditioned using a resistor to convert the current into a voltage reading. This analog voltage signal is converted using an A/D converter on the data acquisition card. The software calculates the temperature reading from each sensor and averages them. In a greenhouse, it is important to take temperature measurements at various points to obtain the average temperature because the building environment is not homogeneous throughout.

Controls

After sensing temperature and relative humidity within the greenhouse, logic within a computer program controls the ventilation system so as to maintain the internal environment at a stable desired condition. The four controllable parts are the ridge vent, end wall vent, humidifier, heater, and fan bank. Figure 8 shows the complete control system wiring diagram. The ridge vent and end wall vent are moved only immediately after the sensor information acquisition. They maintain their position until the logic calls for them to alter their settings. They are all wired to the right side of the relay board. The heater, humidifier, and fans remain running once activated until the environment changes to the point where they are no longer necessary and the logic turns them off. These are all wired to the left side of the relay board.

Both types of vents are opened and closed using Hurst Sl 4013-001 synchronous linear actuators (Figure 9). These linear actuators are bi-directional (forward or reverse) to provide for the up and down motion of the vents. The linear actuator rotor turns an internal nut, which moves a threaded rod up or down at a rate of 6.1 mm/s (Beckwith Electronics, 2001). The threaded rod must be restrained from rotating along with the nut by being anchored to the piece it is lifting. On the ridge vent (Figure 2), the threaded rod is directly attached to the vent that is restrained from rotating by guides attached to the rafters of the building. The end wall vent (Figure 2d, e) utilizes a slide (Figure 2f) to keep the rod from rotating since the rod could not be directly attached to the vent. Two relays are needed to control the up and down motion as is shown in the wiring schematic

(Figure 8). The linear actuators require 120V ac applied to either the forward or reverse leg to become active. A 0.62-µF capacitor between the forward and reverse legs charges when switching from one direction to the other to provide a time lag which prevents overloading of the linear actuator. The vent is set to different positions or stages by running the linear actuator for a short specified period of time. The ridge vent is set to run for 3 s between stages and the end wall vent for 2 s.

The fans are 12V dc computer fans purchased from Radio Shack. Each fan in the fan bank is wired into one of four relays and each one can be controlled independently of the other three (Figure 8). For the example logic program written for the demonstration, increasing the fan stage is equivalent to turning on an additional fan. The fans are run from a 12 Vdc power source and can only exhaust air from the greenhouse.

The evaporative humidifier consists of a water basin, a slotted wet cloth to wick up the water, and a fan (Figure 2b). The fan blows air through the wet cloth causing the water to evaporate into the air and increase the relative humidity. The control of the humidifier consists of turning its fan on or off. This fan also requires a 12 Vdc source and is activated using a relay.

The heater in this system is a 40 W incandescent lamp. It requires 120 V 60 Hz alternating current and is turned on using a relay (Figure 8).

Software User Interface

The control program for this system was written using QUICK BASIC software (V4.1). This software is run from the MS DOS prompt. A software user interface was developed for this model. The first screen (Figure 10a) asks for input on the optimal greenhouse temperature, and the minimum and maximum bounds for relative humidity.

This user input is important during testing because different crops require different environments and a logic program shown to be the best for one crop may not be universally the best. Once the appropriate information has been entered, the status of the greenhouse system is described in various ways on the operational screen (Figure 10b). The basic format of this screen, which stays constant while the program is running, is divided into 4 sections: time information, sensor data, control state, and optimal set point information. Date, time, and Julian day are all shown in the top section on the screen. This information is important in sophisticated ventilation logic programs which take into account the season and length of day. The sensor data section displays the values that the sensors are reading and some interim answers calculated by the software using psychrometric equations. The control state shows vent stages, fan stages, and operation of the heater and humidifier. The bottom part of the screen displays the optimum conditions entered by the user and the average temperature of the temperature sensors that are very important values when debugging logic code.

Logic Program

An example logic program written in QUICK BASIC for this system is included in Appendix A. The main routine, PLANTS.BAS, is the backbone of the program. It calls all of the subroutines at the appropriate time. It prompts the user for temperature and relative humidity set-point information in the beginning and has an escape function to stop the program when the user desires. It then closes the vents and turns off the fans and heater. Some of the subroutines, such as the one that calculates Julian day and psychrometric properties, were written by previous AE 504 students or Dr. Hoff and were integrated into this project so the focus of the project could be on controlling the model. The subroutines “initial” and “display” control how the user screen looks and what values are printed to the screen. The subroutine "logic" contains all of the ventilation decisions that are made based on temperature differences. Within each temperature gradient, another subroutine is called to make ventilation decisions based relative humidity. Another set of conditions could easily be added to this subroutine to control CO_2 levels. The vents are opened and closed by calling the EV subroutines for the endwall vent or RV subroutines for the ridge vent.

RESULTS and DISCUSSION

Two types of tests were run using this model in order to verify its function. In all tests, a heat source was placed inside the model building and the cooling system was tested. In the first type of test, the cooling system and heater were turned on simultaneously to show the building response to a slow build up of heat similar to the increase in solar energy following sunrise. In the second type of test, the heater was allowed to heat up before the cooling system was turned on, thus simulating the building response to a sudden source of heat, such as a malfunctioning heater.

For the first type of test, two different logic programs were simulated. In the first trial, a logic program was used which turned on stages of the cooling system at even 1.1°C increments above the set-point temperature at 1.1, 2.2, 3.3, and 4.4°C (2, 4, 6, and 8°F, Figure 11, trial 1). In the second trial, the stages were at 0.28, 0.56, 1.1, and 2.2°C (0.5, 1, 2, and 4°F, Figure 11, trial 2) above the optimum temperature. In both cases, the optimum temperature was reached but the first logic program was slower in response than the second logic program. This is an example of how to use the model for comparing different logic programs.

For the second type of test, only the second logic program was simulated (Figure 12). It demonstrates the effectiveness of the logic program in cooling the building to the optimum temperature.

The overall design of the model is a success. After reading in and analyzing data from the sensors, the logic makes adjustments in the ventilation controls of the greenhouse. By watching the sensor data on the operational screen, it is easy to see that when the controls are activated, the internal environment responds accordingly. To test a logic system for efficiency, the model greenhouse could be placed in a controlled chamber or wind tunnel where weather patterns could be simulated. This way, several ventilation logic programs could be evaluated using the same created weather conditions within the chamber. A comparative test of this nature would be impossible in the field because the weather is not that predictable.

This basic model has fulfilled all of the requirements of a basic greenhouse model used to test ventilation. With a few simple additions, it can be used to simulate almost any greenhouse system. For example, most greenhouses have an internal air circulation system to mix air and prevent the development of a temperature gradient between the floor and the ceiling. The two most common systems are horizontal airflow fans (HAF) and convection tubes (Boyette *ca* 1996). Depending on what system the greenhouse model was simulating, these systems could be easily added. In addition to this, cooling systems could be evaluated. Three types of cooling systems include evaporative pad and fan cooling, mist cooling, and swamp coolers. These would be a little more complicated to implement, but would make the system more versatile. The possibilities of what this model could be used for is limitless as shade cloth, heat screens, and even solar energy storage could be included to model a wide variety of systems.

CONCLUSIONS

The model greenhouse meets all of the objectives set out at the beginning of the project. It reads parameter values from the sensors, the logic evaluates these values, and the ventilation system is controlled accordingly to alter the greenhouse environment. Testing verified that this model is effective in comparing responses of different ventilation logic programs to changes in the environment and in evaluating the efficiency of single ventilation logic programs. With a few simple or complicated additions, this model will be able to simulate endless combinations of ventilation systems.

REFERENCES

Advantech Co., Ltd. 1995. PC-Labcard DLL Driver User's Manual. Advantech Co., Ltd. Taiwan.

Aldrich, Robert A., John W. Bartok. 1994. Greenhouse Engineering. Natural Resource, Agriculture, and Engineering Service (NRAES). Cooperative Extension. Ithaca, NY.

Beckwith Electronics. 2001. Hurst Manufacturing. Fenton, Missouri. http://www.beckwithelectronics.com/HURST/slb_sls.htm. Consulted February 4, 2003.

Boyette M. D. *ca* 1996. Producing Tobacco Transplants in a Greenhouse: Heating and Ventilation Systems AG-488-4. North Carolina Cooperative Extension Services. http://www.ces.ncsu.edu/resources/crops/tobacco/transplants/AG488-4/#vent-air. Consulted February 4, 2003.

Buffington, D. E., R. A. Bucklin, R. W. Henley and D. B. McConnell. 1987. Greenhouse Ventilation. University of Florida Cooperative Extension Services. http://edis.ifas.ufl.edu/AE030. Consulted February 4, 2003.

Evoc Intelligent Technology Company, Inc. 1998. PCL-711B PC-MultiLab Card. http://www.evoc.com/I-english/isa/pcl-711b.htm Consulted February 22, 2003.

Fosdick, Dean. 2003. Gardener Supplies Veggies in Antarctica. Associated Press. http://www.sanluisobispo.com/mld/sanluisobispo/news/nation/4901017.htm. Consulted February 11, 2003.

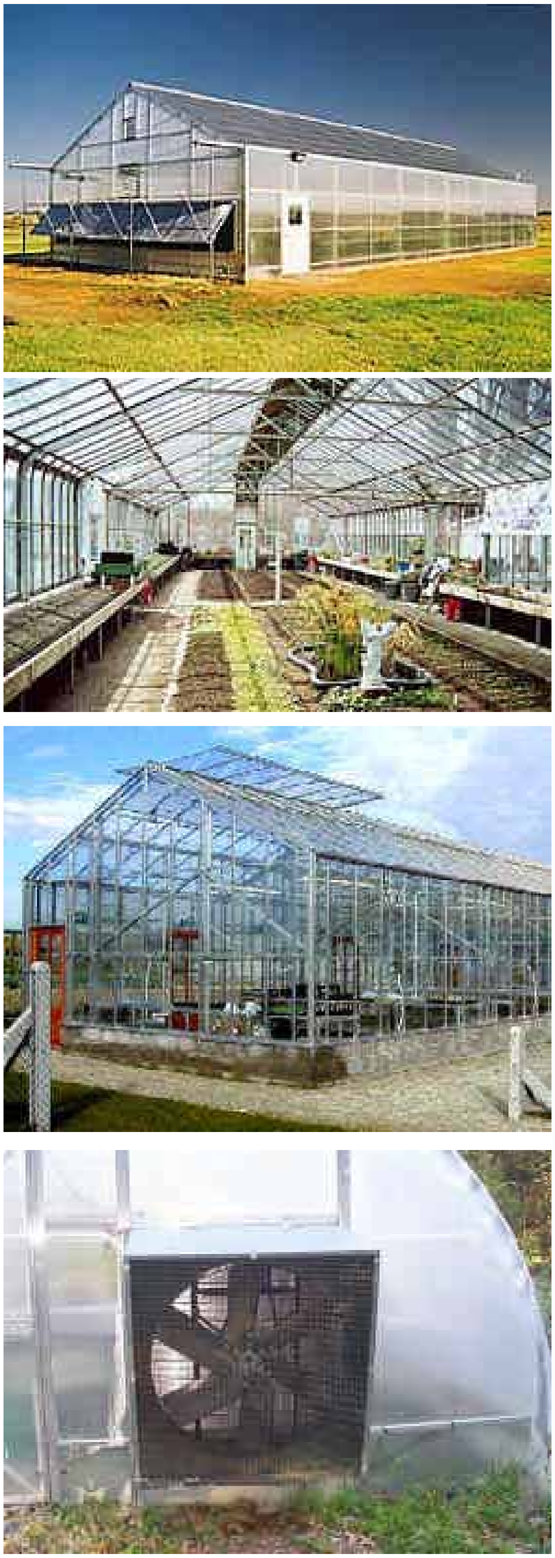
Jovanka, Ciric and Milan Milicevic. 1994. User's Guide for PC-Lab PCL-711B Card. http://titan.etf.bg.ac.yu/rti/ri5rvl/dodatak/pc_lab.htm. Consulted February 22, 2003.

Nelson, Paul V. 2003. Greenhouse Operation and Management. Prentice Hall. Upper Saddle River, New Jersey.

Omega Engineering Inc. 2003. Stamford, Connecticut. http://www.omega.com/ppt/pptsc.asp?ref=HX93&Nav=hean01. Consulted February 4, 2003.

United States Department of Agriculture National Agricultural Statistics Service. 2002. Floriculture Crops 2001 Summary http://usda.mannlib.cornell.edu/reports/nassr/other/zfc-bb/floran02.txt Consulted February 21, 2003.

West Virginia University Extension Services. Adapted from Fact Sheet 645 - University of Maryland Cooperative Extension Service, David S. Ross. 1996 Planning and Building a Greenhouse. http://www.wvu.edu/~agexten/hortcult/greenhou/building.htm#Ventilation Consulted February 21, 2003



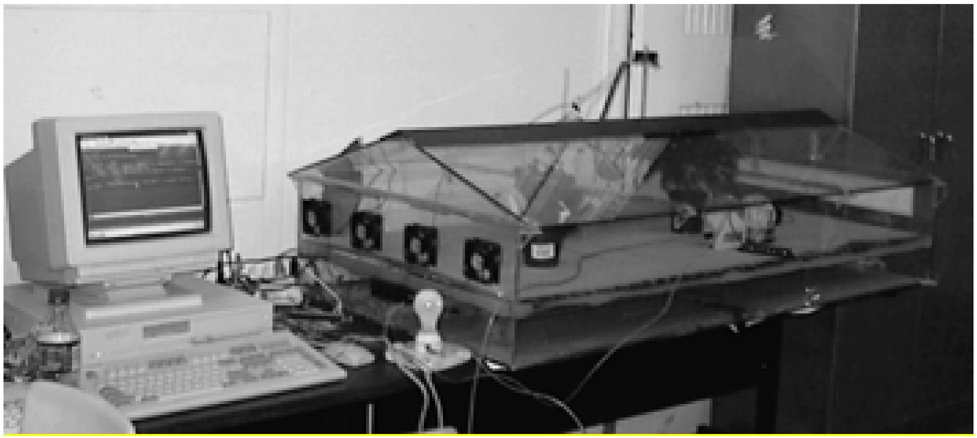
a. An example of a side vent on the endwall of the greenhouse. The inlet opening is variably controlled using linear actuators.

b. An example of a modern production greenhouse. Notice the ridge vent on the top.

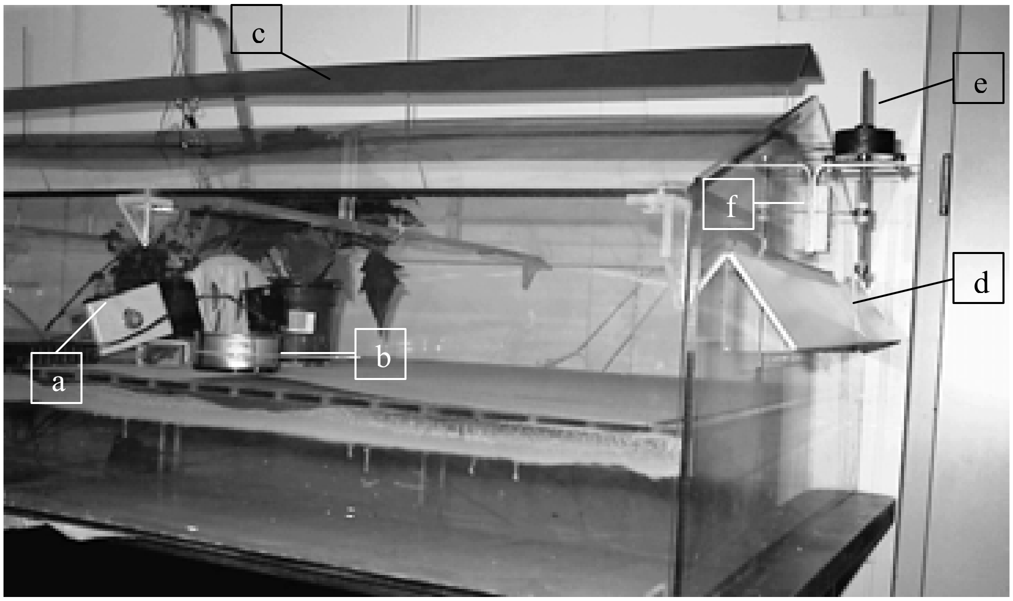
c. An example of a ridge vent that hinges along the ridge pole for natural ventilation.

d. An example of a common type of greenhouse ventilation fan.

Figure 1. Modern production greenhouse ventilation system components (Google image search)



Computer and plexiglass model building.



Plexiglas model showing the relative humidity sensor (a), humidifier (b), ridge vent in the open position (c), endwall vent in the open position (d), Hurst Sl 4013-001 synchronous linear actuator (e), and the slide (f) used to restrain the linear actuator from rotating.

Figure 2. Model greenhouse

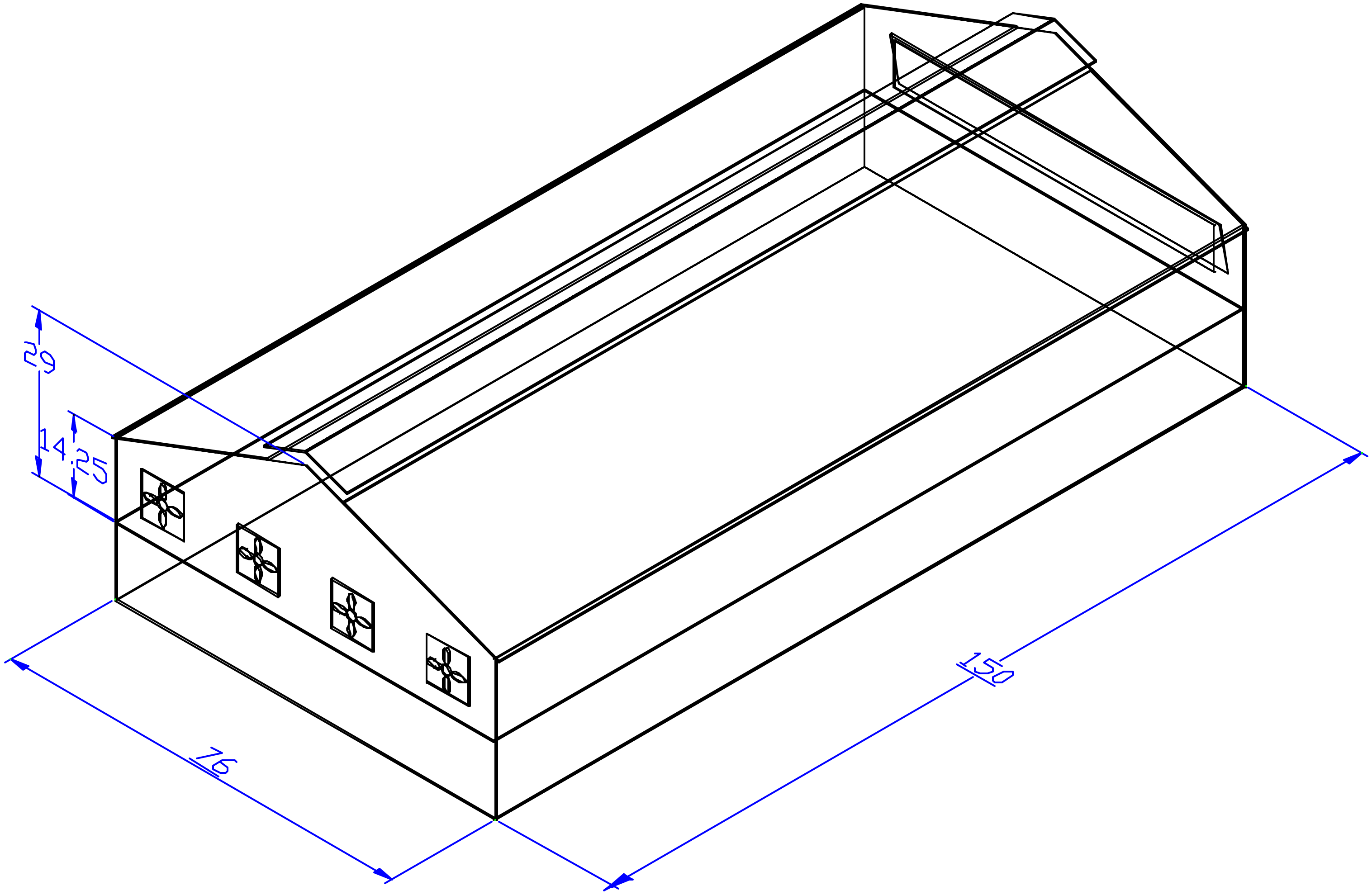


Figure 3. Model greenhouse AutoCAD drawing (Dimensions are in cm)

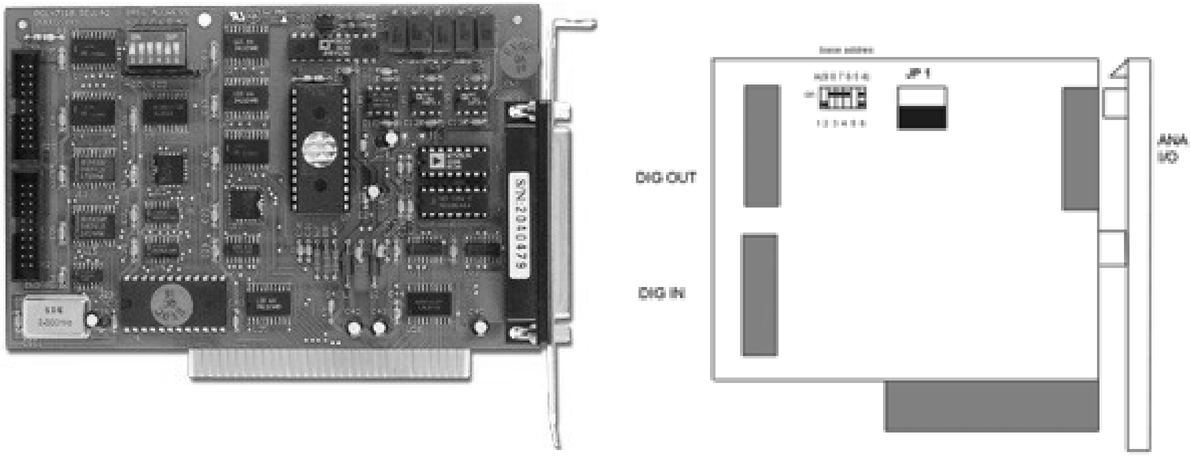
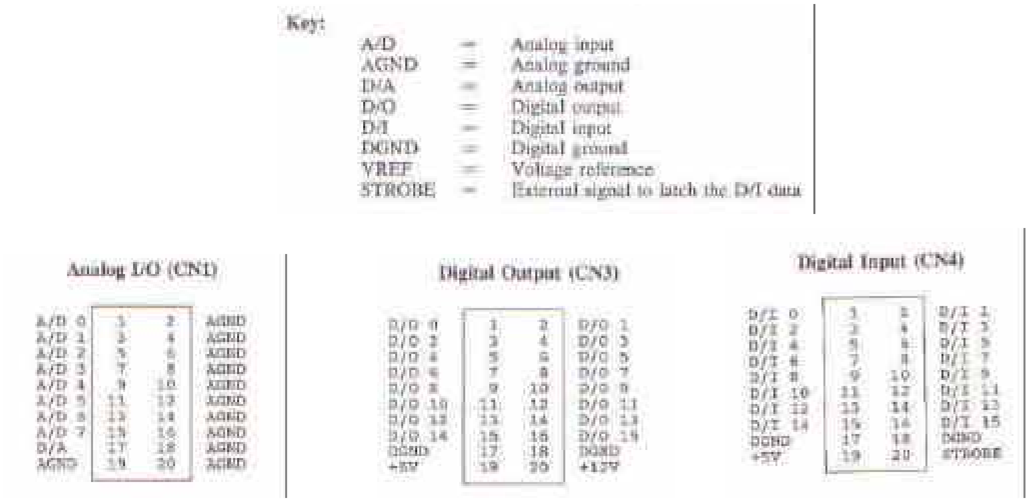


Figure 4. PCL-711B Picture and Drawing (EVOC Intelligent Technology Co. Inc 1998, Jovanka and Milicevic 1994)



Key:

A/D	=	Analog input
AGND	=	Analog ground
D/A	=	Analog output
D/O	=	Digital output
D/I	=	Digital input
DGND	=	Digital ground
VREF	=	Voltage reference
STROBE	=	External signal to latch the D/I data

Analog I/O (CN1)

A/D 0	1	2	AGND
A/D 1	3	4	AGND
A/D 2	5	6	AGND
A/D 3	7	8	AGND
A/D 4	9	10	AGND
A/D 5	11	12	AGND
A/D 6	13	14	AGND
A/D 7	15	16	AGND
D/A	17	18	AGND
AGND	19	20	AGND

Digital Output (CN3)

D/O 0	1	2	D/O 1
D/O 2	3	4	D/O 3
D/O 4	5	6	D/O 5
D/O 6	7	8	D/O 7
D/O 8	9	10	D/O 9
D/O 10	11	12	D/O 11
D/O 12	13	14	D/O 13
D/O 14	15	16	D/O 15
DGND	17	18	DGND
+5V	19	20	+12V

Digital Input (CN4)

D/I 0	1	2	D/I 1
D/I 2	3	4	D/I 3
D/I 4	5	6	D/I 5
D/I 6	7	8	D/I 7
D/I 8	9	10	D/I 9
D/I 10	11	12	D/I 11
D/I 12	13	14	D/I 13
D/I 14	15	16	D/I 15
DGND	17	18	DGND
+5V	19	20	STROBE

Figure 5. PCL-711B Illustration of connectors (Advantech Co., Ltd 1995)

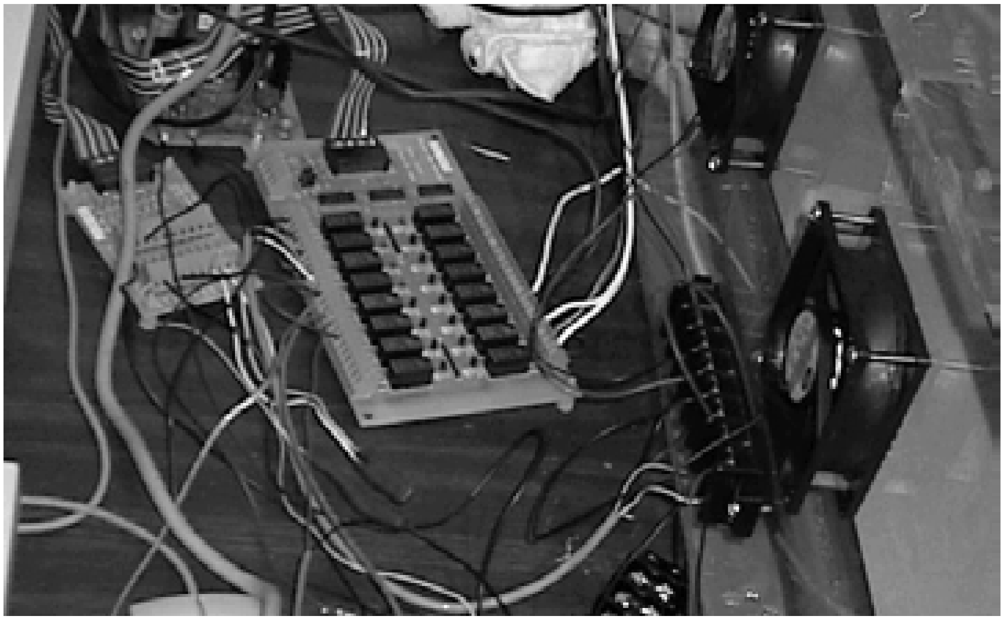


Figure 6. PCLD-785B 24/16 Channel relay output board from Advantech Co., Ltd

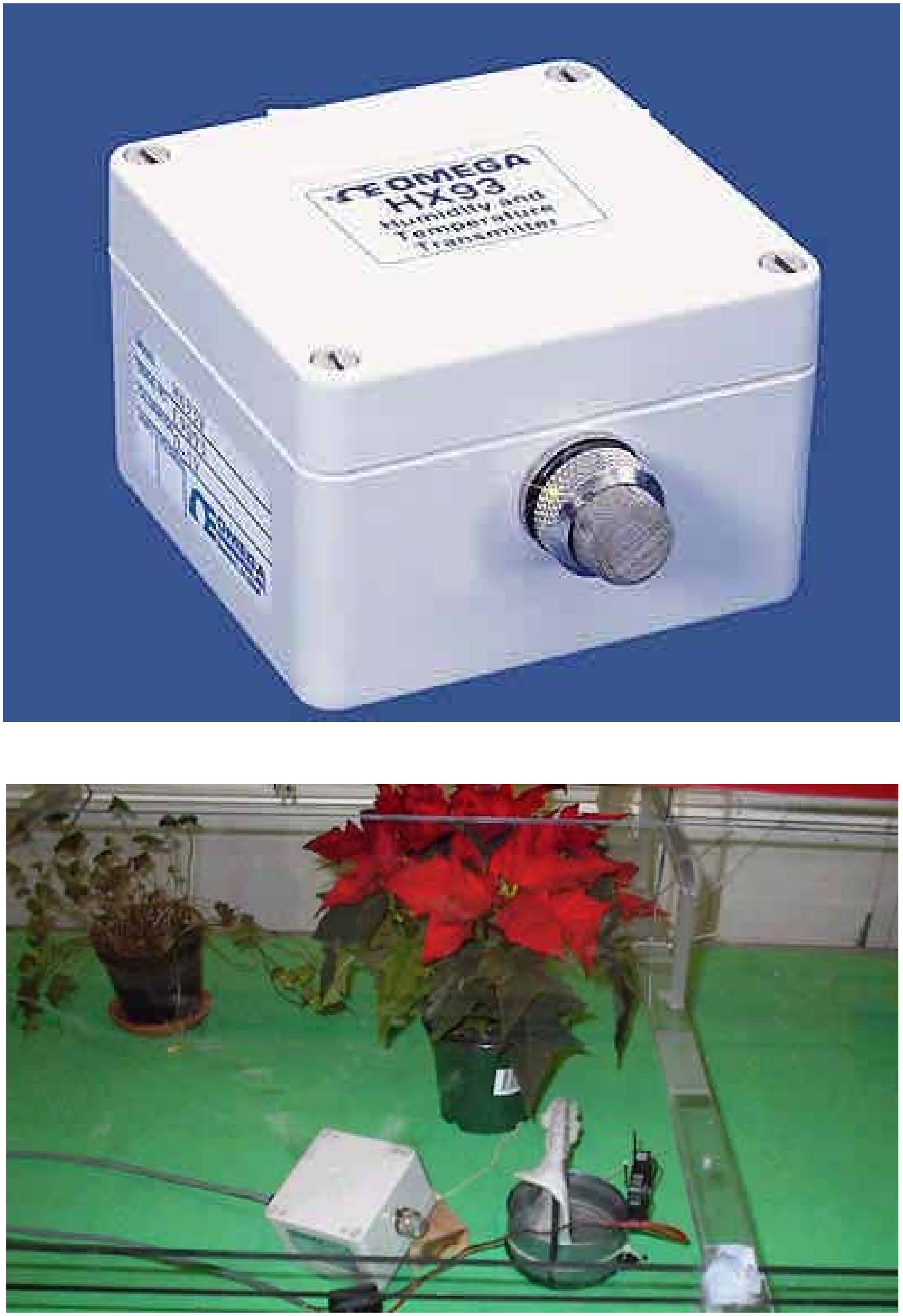
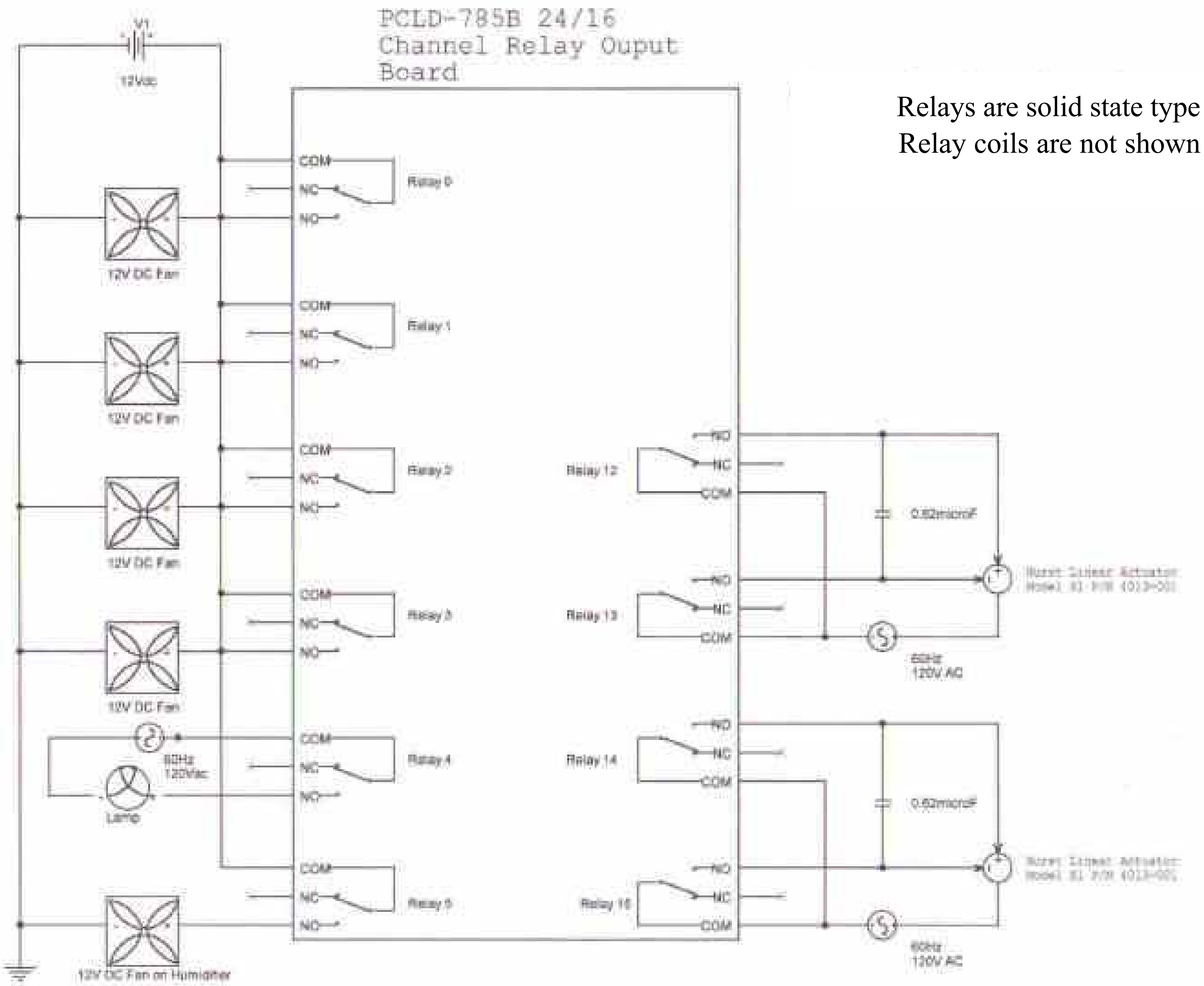
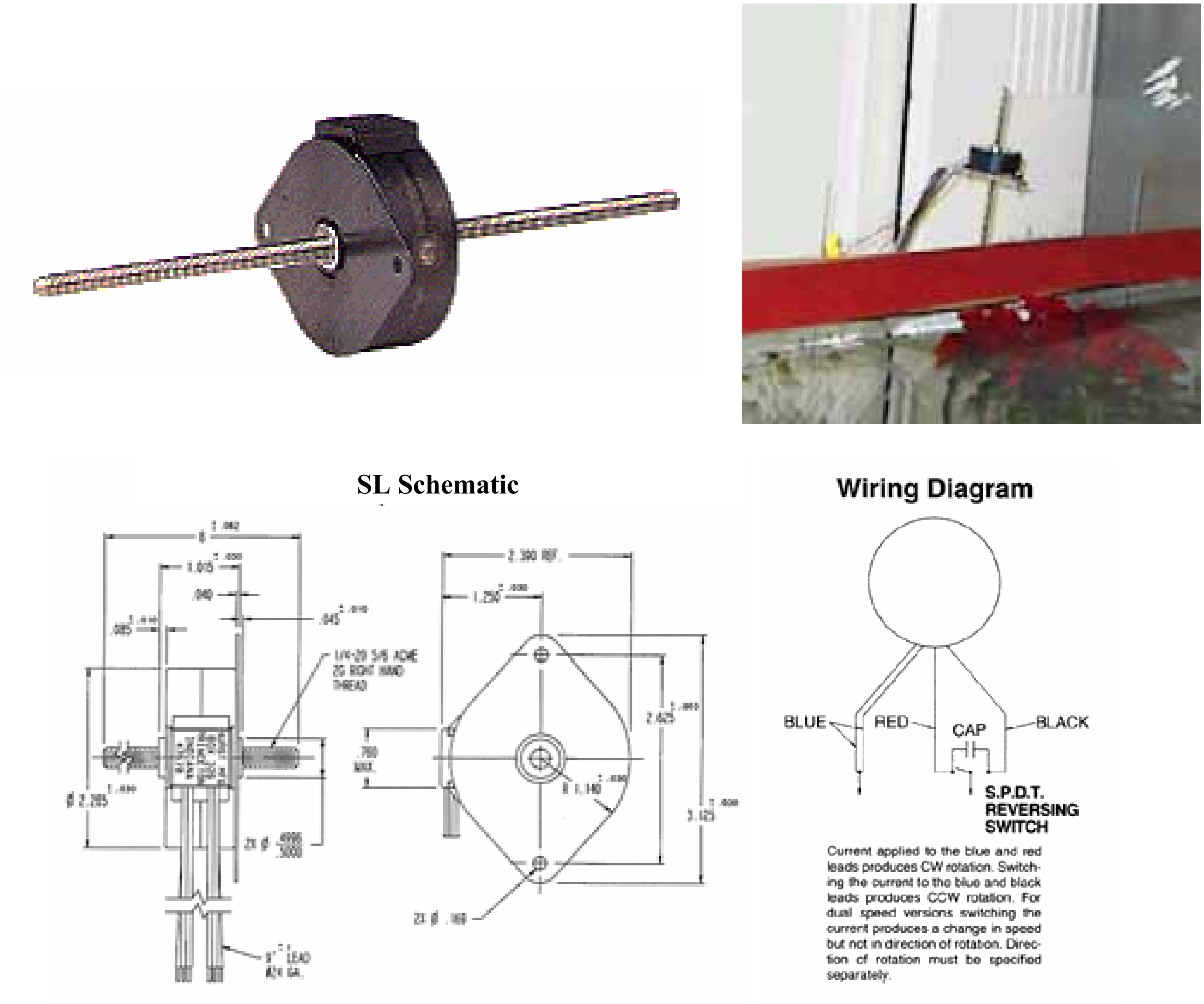


Figure 7. Omega HX93 Relative humidity and temperature sensor (Omega Engineering Inc 2003)



Relays are solid state type
Relay coils are not shown

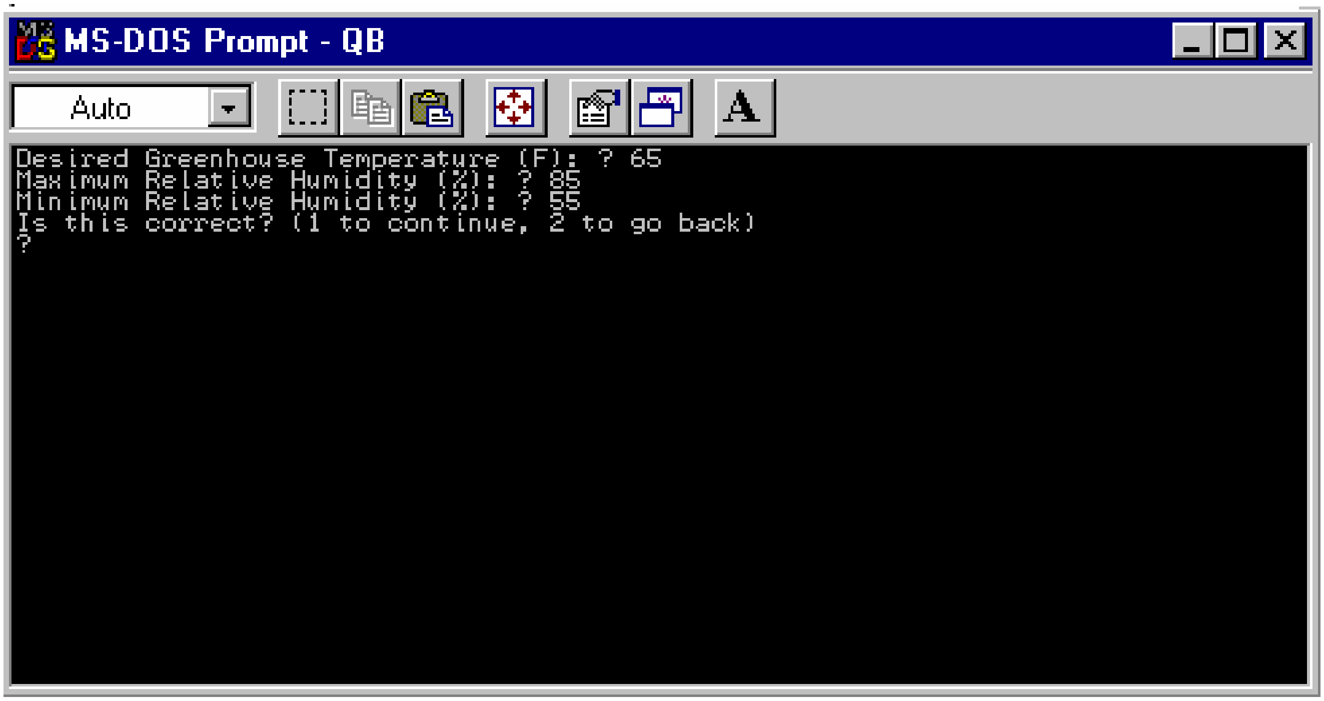
Figure 8. Wiring schematic



Current applied to the blue and red leads produces CW rotation. Switching the current to the blue and black leads produces CCW rotation. For dual speed versions switching the current produces a change in speed but not in direction of rotation. Direction of rotation must be specified separately.

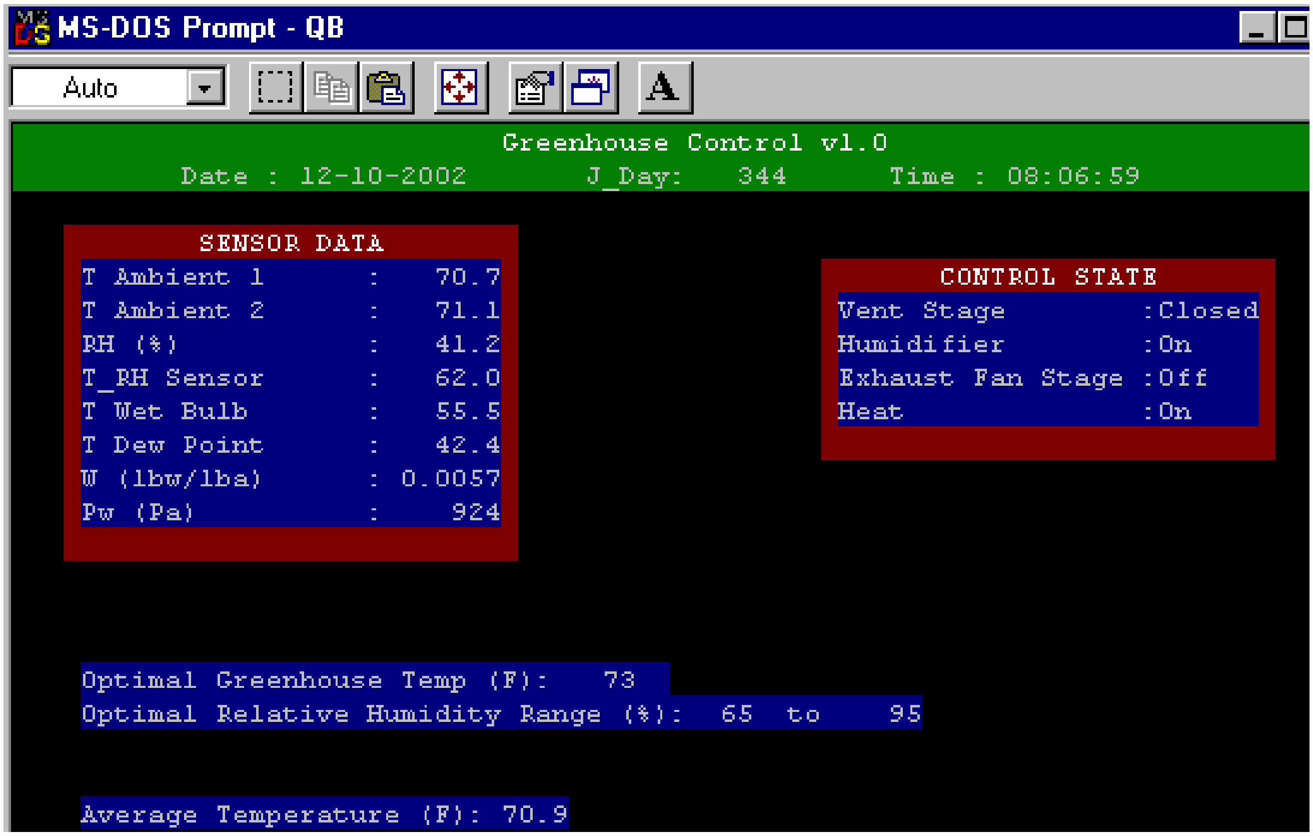
SYNCHRONOUS LINEAR ACTUATOR Model	Part Number	Stock Item	In/Sec	Maximum Load (lbs)	Input Power (watts)	Normal Voltage	Capacitor Value (mfd)	Full Load Temp. Rise (oC)	Weight (with screw) (oz)
SL	4013-001	X	.24	15	6.5	115V.60Hz	.62	37	10.5

Figure 9. Hurst Sl 4013-001 synchronous linear actuator (Beckwith Electronics 2001)



```
Desired Greenhouse Temperature (F): ? 65
Maximum Relative Humidity (%): ? 85
Minimum Relative Humidity (%): ? 55
Is this correct? (1 to continue, 2 to go back)
?
```

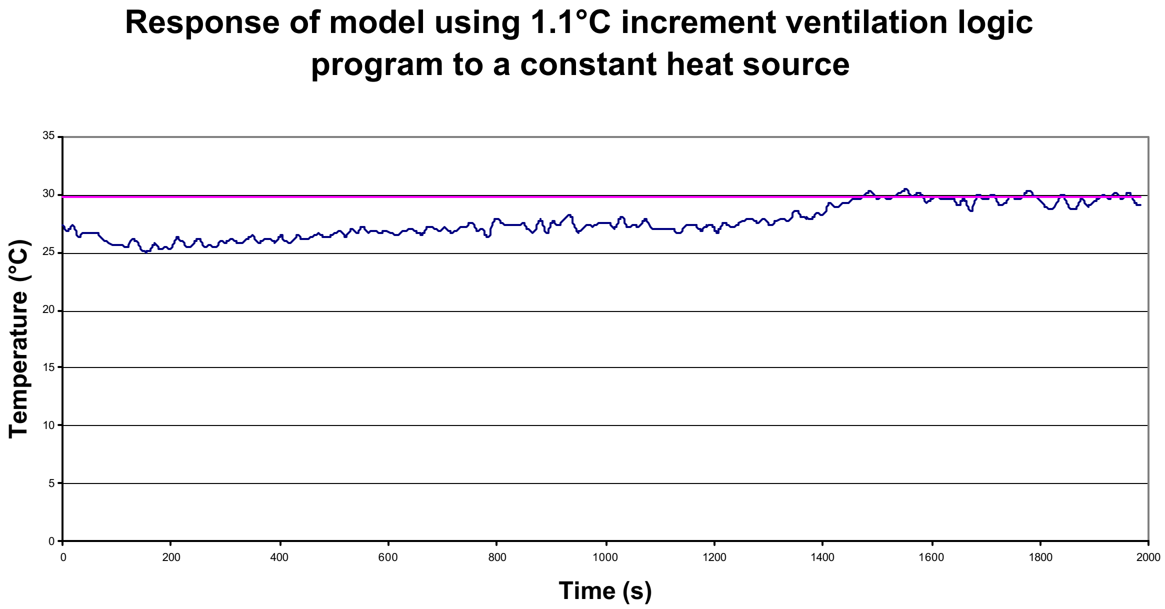
a. User input screen



```
                    Greenhouse Control v1.0
       Date : 12-10-2002        J_Day:    344        Time : 08:06:59

          SENSOR DATA
 T Ambient 1          :   70.7
 T Ambient 2          :   71.1                      CONTROL STATE
 RH (%)               :   41.2             Vent Stage          :Closed
 T_RH Sensor          :   62.0             Humidifier          :On
 T Wet Bulb           :   55.5             Exhaust Fan Stage   :Off
 T Dew Point          :   42.4             Heat                :On
 W (lbw/lba)          : 0.0057
 Pw (Pa)              :    924


 Optimal Greenhouse Temp (F):    73
 Optimal Relative Humidity Range (%):  65  to    95


 Average Temperature (F): 70.9
```

b. Operational Screen

Figure 10. User interface screens


Response of model using 1.1°C increment ventilation logic program to a constant heat source

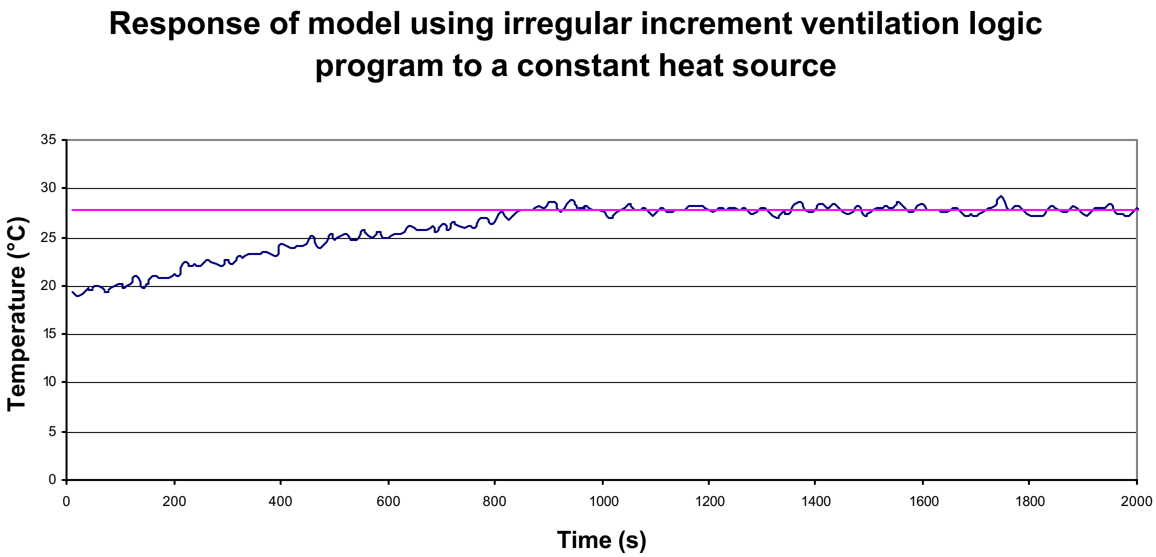

Response of model using irregular increment ventilation logic program to a constant heat source

Figure 11. Results of test type 1, trials 1 and 2

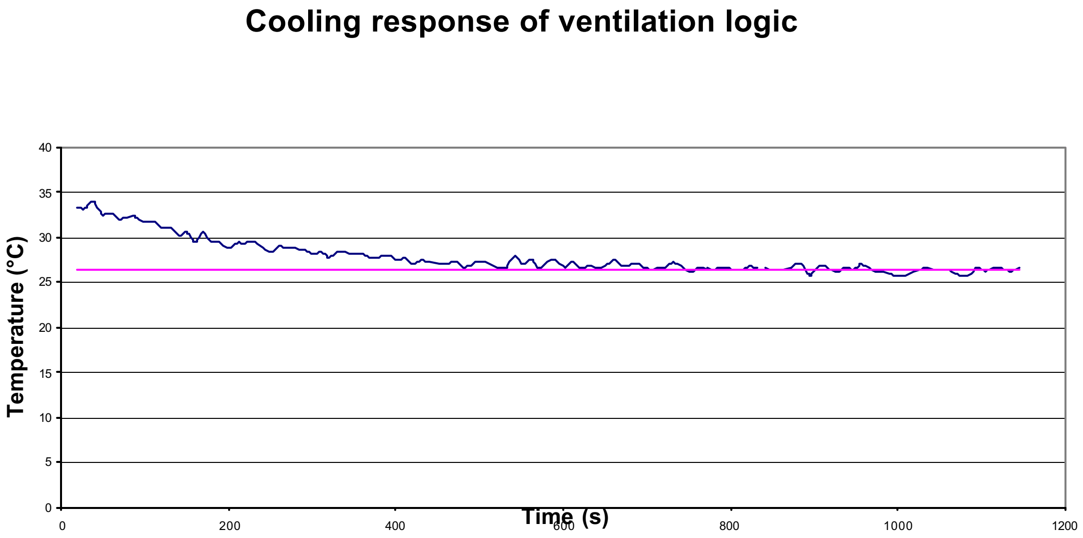


Figure 12. Results of test type 2

Appendix B: Computer Logic

PLANTS.BAS

```
DECLARE SUB wetbulb ()
DECLARE SUB Logic ()
DECLARE SUB julianday (ijd%)
DECLARE SUB Initial ()
DECLARE SUB display ()
DECLARE SUB sensors ()
DECLARE SUB DtoA ()
DECLARE SUB devices ()
DECLARE SUB devices1 ()
DECLARE SUB devices2 ()
DECLARE SUB logic1 ()
DECLARE SUB logic2 ()
DECLARE SUB logic3 ()
DECLARE SUB RVdown ()
DECLARE SUB RVup ()
DECLARE SUB EVup ()
DECLARE SUB EVdown ()

COMMON SHARED BaseAdd, Temp6, Temp7, RH, DryBulb, TWetBulb, Julian
COMMON SHARED DewPoint, AbsHum, VaporP
COMMON SHARED RelayStatus, CurrentStatus$
COMMON SHARED SetPoint, relay2, relay1, RHsetL, RHsetH, RHdemo, yn, TAve
COMMON SHARED fanstage, fancontrol, HeatCont, MistCont, ventstage, ventstageprev,

    BaseAdd = 768
1
    CLS : 'clear the screen
    PRINT "Desired Greenhouse Temperature (F): "; : INPUT SetPoint
5
    PRINT "Maximum Relative Humidity (%): "; : INPUT RHsetH
    PRINT "Minimum Relative Humidity (%): "; : INPUT RHsetL
   IF (RHsetL > 100 OR RHsetH > 100) THEN PRINT "Relative Humidity cannot be larger than 100%": GOTO 5
    IF (RHsetL > RHsetH) THEN PRINT "Minimum value cannot be larger than maximum value.": GOTO 5
6
    PRINT "Is this correct? (1 to continue, 2 to go back) ": INPUT yn
    IF yn = 1 THEN GOTO 7
    IF yn = 2 THEN GOTO 1
   IF (yn < 1 OR yn > 2) THEN PRINT "This is not a valid entry": GOTO 6
7
    CLS
    ventstage = 0!
'Inilialize the screen output statements
    CALL Initial
    CALL julianday(ijd%)
    CALL DtoA
10
    CALL sensors
    CALL wetbulb
    CALL Logic
    CALL display

    IF INKEY$ = "q" THEN GOTO 20
    GOTO 10

20
'Turn off fans, heater, humidifier
  OUT BaseAdd + 13, 0
```

```
'Close ridge vent
j = 0!
WHILE j < ventstage: CALL RVdown: j = j + 1: WEND

j = 0!
WHILE j < ventstage: CALL EVdown: j = j + 1: WEND

'Turn off all relays as a precaution
   OUT BaseAdd + 13, 0: OUT BaseAdd + 14, 0
END SUB
```

```
SUB devices

'This is where we will actually control devices via relays

'Relay Driver Control (PCLD-785 via the PCLD-711S)
   OUT BaseAdd + 13, relay1
   OUT BaseAdd + 14, relay2

END SUB
```

```
SUB devices1
   OUT BaseAdd + 13, relay1
END SUB
```

```
SUB devices2
   OUT BaseAdd + 14, relay2
END SUB
```

```
SUB display

LOCATE 5, 23: COLOR 7, 1: PRINT "       "
LOCATE 6, 23: PRINT "      "
LOCATE 7, 23: PRINT "      "
LOCATE 8, 23: PRINT "      "
LOCATE 9, 23: PRINT "      "
LOCATE 10, 23: PRINT "      "
LOCATE 11, 23: PRINT "      "
LOCATE 12, 23: PRINT "      "

LOCATE 5, 24: PRINT USING "####.#"; Temp6
LOCATE 6, 24: PRINT USING "####.#"; Temp7
LOCATE 7, 24: PRINT USING "####.#"; RHdemo
LOCATE 8, 24: PRINT USING "####.#"; DryBulb
LOCATE 9, 24: PRINT USING "####.#"; TWetBulb
LOCATE 10, 24: PRINT USING "####.#"; DewPoint
LOCATE 11, 24: PRINT USING "#.####"; AbsHum
LOCATE 12, 24: PRINT USING "######"; VaporP

LOCATE 2, 18: COLOR 7, 2: PRINT DATE$
LOCATE 2, 60: PRINT TIME$

LOCATE 6, 69: COLOR 7, 1
    IF (ventstage = 0) THEN PRINT "Closed"
    IF (ventstage = 1) THEN PRINT " 1    "
    IF (ventstage = 2) THEN PRINT " 2    "
    IF (ventstage = 3) THEN PRINT " 3    "
    IF (ventstage = 4) THEN PRINT " 4    "
LOCATE 7, 69
    IF (MistCont = 32) THEN PRINT "On "
    IF (MistCont = 0) THEN PRINT "Off"
```

```
LOCATE 8, 69
        IF (fanstage = 0) THEN PRINT "Off"
        IF (fanstage = 1) THEN PRINT " 1  "
        IF (fanstage = 2) THEN PRINT " 2  "
        IF (fanstage = 3) THEN PRINT " 3  "
        IF (fanstage = 4) THEN PRINT " 4  "

LOCATE 9, 69
        IF (HeatCont = 0) THEN PRINT "Off"
        IF (HeatCont = 16) THEN PRINT "On "

LOCATE 17, 35: PRINT USING "###"; SetPoint
LOCATE 18, 43: PRINT USING "##"; RHsetL
LOCATE 18, 53: PRINT USING "##"; RHsetH

LOCATE 21, 29: PRINT USING "###.#"; TAve

END SUB
```

```
SUB DtoA

'D/A Output Control (via the PCLD-711S)
' For now this is used to power AD592's
     OUT BaseAdd + 4, 255
     OUT BaseAdd + 5, 15

END SUB
```

```
SUB EVdown
    relay2 = 64
    CALL devices2
    SLEEP 2
    relay2 = 0
    CALL devices2

END SUB
```

```
SUB EVup
    relay2 = 128
    CALL devices2
    SLEEP 2
    relay2 = 0
    CALL devices2

END SUB
```

```
SUB Initial

CLS

'sensor border
LOCATE 4, 4: COLOR 9, 4: PRINT "                          "
LOCATE 5, 4: COLOR 9, 4: PRINT " "
LOCATE 6, 4: COLOR 9, 4: PRINT " "
LOCATE 7, 4: COLOR 9, 4: PRINT " "
LOCATE 8, 4: COLOR 9, 4: PRINT " "
LOCATE 9, 4: COLOR 9, 4: PRINT " "
LOCATE 10, 4: COLOR 9, 4: PRINT " "
LOCATE 11, 4: COLOR 9, 4: PRINT " "
LOCATE 12, 4: COLOR 9, 4: PRINT " "
LOCATE 13, 4: COLOR 9, 4: PRINT "                          "
LOCATE 5, 30: COLOR 9, 4: PRINT " "
LOCATE 6, 30: COLOR 9, 4: PRINT " "
LOCATE 7, 30: COLOR 9, 4: PRINT " "
LOCATE 8, 30: COLOR 9, 4: PRINT " "
LOCATE 9, 30: COLOR 9, 4: PRINT " "
LOCATE 10, 30: COLOR 9, 4: PRINT " "
LOCATE 11, 30: COLOR 9, 4: PRINT " "
LOCATE 12, 30: COLOR 9, 4: PRINT " "

LOCATE 4, 12: COLOR 15, 4: PRINT "SENSOR DATA"
LOCATE 5, 5: COLOR 7, 1: PRINT "T Ambient 1      :       "
LOCATE 6, 5: COLOR 7: PRINT "T Ambient 2      :       "
LOCATE 7, 5: PRINT "RH (%)           :       "
LOCATE 8, 5: PRINT "T_RH Sensor      :       "
LOCATE 9, 5: PRINT "T Wet Bulb       :       "
LOCATE 10, 5: PRINT "T Dew Point      :       "
LOCATE 11, 5: PRINT "W (lbw/lba)      :       "
LOCATE 12, 5: PRINT "Pw (Pa)          :       "

'Top banner
LOCATE 1, 1: COLOR 15, 2: PRINT "
"
LOCATE 1, 30: COLOR 15, 2: PRINT "Greenhouse Control v1.0 "; ""
LOCATE 2, 11: COLOR 7: PRINT "Date :  "
LOCATE 2, 53: PRINT "Time : "

'control border
LOCATE 5, 49: COLOR 9, 4: PRINT "                          "
LOCATE 6, 49: COLOR 9, 4: PRINT " "
LOCATE 7, 49: COLOR 9, 4: PRINT " "
LOCATE 8, 49: COLOR 9, 4: PRINT " "
LOCATE 9, 49: COLOR 9, 4: PRINT " "
LOCATE 10, 49: COLOR 9, 4: PRINT "                          "
LOCATE 6, 75: COLOR 9, 4: PRINT " "
LOCATE 7, 75: COLOR 9, 4: PRINT " "
LOCATE 8, 75: COLOR 9, 4: PRINT " "
LOCATE 9, 75: COLOR 9, 4: PRINT " "

LOCATE 5, 56: COLOR 15, 4: PRINT "CONTROL STATE"
LOCATE 6, 50: COLOR 7, 1: PRINT "Vent Stage        :      "
LOCATE 7, 50: PRINT "Humidifier        :      "
LOCATE 8, 50: PRINT "Exhaust Fan Stage :      "
LOCATE 9, 50: PRINT "Heat              :      "

'Set Points
LOCATE 17, 5: PRINT "Optimal Greenhouse Temp (F):       "
LOCATE 18, 5: PRINT "Optimal Relative Humidity Range (%):      to      "
LOCATE 21, 5: PRINT "Average Temperature (F): "
END SUB

SUB julianday (ijd%)

   b$ = DATE$

'   first get the julian date based on the month

   a1$ = MID$(b$, 1, 2)

   IF a1$ > "06" THEN
      IF a1$ = "12" THEN mes% = 334: ' december
      IF a1$ = "11" THEN mes% = 304: ' november
      IF a1$ = "10" THEN mes% = 273: ' october
      IF a1$ = "09" THEN mes% = 243: ' september
      IF a1$ = "08" THEN mes% = 212: ' august
      IF a1$ = "07" THEN mes% = 181: ' july
   ELSE
      IF a1$ = "06" THEN mes% = 151: ' june
      IF a1$ = "05" THEN mes% = 120: ' may
      IF a1$ = "04" THEN mes% = 90: ' april
      IF a1$ = "03" THEN mes% = 59: ' march
      IF a1$ = "02" THEN mes% = 31: ' february
      IF a1$ = "01" THEN mes% = 0: ' january
   END IF

'            then account for the day of the month

   a1$ = MID$(b$, 4, 1)

   IF a1$ = "0" THEN day% = 0
   IF a1$ = "1" THEN day% = 10
   IF a1$ = "2" THEN day% = 20
   IF a1$ = "3" THEN day% = 30

   a1$ = MID$(b$, 5, 1)

   IF a1$ > "5" THEN
      IF a1$ = "9" THEN day% = day% + 9
      IF a1$ = "8" THEN day% = day% + 8
      IF a1$ = "7" THEN day% = day% + 7
      IF a1$ = "6" THEN day% = day% + 6
   ELSE
      IF a1$ = "5" THEN day% = day% + 5
      IF a1$ = "4" THEN day% = day% + 4
      IF a1$ = "3" THEN day% = day% + 3
      IF a1$ = "2" THEN day% = day% + 2
      IF a1$ = "1" THEN day% = day% + 1
   END IF

'  account for leap year

   FOR x = 9 TO 10

    a1$ = MID$(b$, x, 1)

   IF a1$ > "4" THEN
      IF a1$ = "9" AND x = 9 THEN ano% = 90
      IF a1$ = "9" AND x = 10 THEN ano% = ano% + 9
      IF a1$ = "8" AND x = 9 THEN ano% = 80
      IF a1$ = "8" AND x = 10 THEN ano% = ano% + 8
      IF a1$ = "7" AND x = 9 THEN ano% = 70
      IF a1$ = "7" AND x = 10 THEN ano% = ano% + 7
      IF a1$ = "6" AND x = 9 THEN ano% = 60
      IF a1$ = "6" AND x = 10 THEN ano% = ano% + 6
      IF a1$ = "5" AND x = 9 THEN ano% = 50
      IF a1$ = "5" AND x = 10 THEN ano% = ano% + 5
   ELSE
      IF a1$ = "4" AND x = 9 THEN ano% = 40
      IF a1$ = "4" AND x = 10 THEN ano% = ano% + 4
      IF a1$ = "3" AND x = 9 THEN ano% = 30
      IF a1$ = "3" AND x = 10 THEN ano% = ano% + 3
      IF a1$ = "2" AND x = 9 THEN ano% = 20
      IF a1$ = "2" AND x = 10 THEN ano% = ano% + 2
      IF a1$ = "1" AND x = 9 THEN ano% = 10
      IF a1$ = "1" AND x = 10 THEN ano% = ano% + 1
      IF a1$ = "0" AND x = 9 THEN ano% = 0
   END IF

   NEXT x

   leap% = 0

   IF (ano% / 4) = INT(ano% / 4) THEN leap% = 1

'   calculate julian day

   ijd% = mes% + day% + leap%

   LOCATE 2, 35: COLOR 7, 2: PRINT "J_Day:  "; ijd%
   Julian = ijd%

END SUB
```

```
SUB Logic

ventstageprev = ventstage
'This is where code will go to control greenhouse operation

IF RH > 20 THEN RHdemo = (RH - 20) ^ 2 / 12 + 20: IF RHdemo > 100 THEN RHdemo = 100
IF (RH < 20 OR RH = 20) THEN RHdemo = RH

TAve = .5 * (Temp6 + Temp7)
'Set Humidifier system control to 0
MistCont = 0
'Turn Heater off for any temperature above optimum
IF (TAve > (SetPoint)) THEN HeatCont = 0

IF (TAve < SetPoint) THEN HeatCont = 16: CALL logic1
IF (TAve >= SetPoint AND TAve < (SetPoint + 2)) THEN CALL logic1
IF (TAve >= (SetPoint + 2) AND TAve < (SetPoint + 4)) THEN ventstage = 1: fanstage = 1: CALL logic2
IF (TAve >= (SetPoint + 4) AND TAve < (SetPoint + 6)) THEN ventstage = 2: fanstage = 2: CALL logic2
IF (TAve >= (SetPoint + 6) AND TAve < (SetPoint + 8)) THEN ventstage = 3: fanstage = 3: CALL logic2
IF (TAve >= (SetPoint + 8)) THEN ventstage = 4: fanstage = 4

vstagediff = ventstage - ventstageprev

i = 0

IF vstagediff > 0 THEN
      WHILE i < vstagediff: CALL RVup: i = i + 1: WEND
      i = 0
      WHILE i < vstagediff: CALL EVup: i = i + 1
      WEND
END IF

IF vstagediff < 0 THEN
      WHILE i < ABS(vstagediff): CALL RVdown: i = i + 1: WEND
      i = 0
      WHILE i < ABS(vstagediff): CALL EVdown: i = i + 1
      WEND

END IF

IF fanstage = 0 THEN fancontrol = 0
IF fanstage = 1 THEN fancontrol = 2
IF fanstage = 2 THEN fancontrol = 10
IF fanstage = 3 THEN fancontrol = 14
IF fanstage = 4 THEN fancontrol = 15
```

```
relay1 = fancontrol + HeatCont + MistCont

CALL devices1

END SUB
```

```
SUB logic1
'Turn Exhaust fans off
fanstage = 0

'Set ventilation to 0 to begin the test for minimum ventilation need
ventstage = 0

'Check if need to mist
IF (RHdemo < (RHsetL + (RHsetH - RHsetL) / 4)) THEN MistCont = 32'Mist on
IF (RHdemo > RHsetL) THEN MistCont = 0 'Mist off

‘Check if minimum ventilation is needed to control RH
IF (RHdemo > RHsetH) THEN ventstage = 1
END SUB
```

```
SUB logic2
IF fanstage < 2 THEN
IF (RHdemo < (RHsetL + (RHsetH - RHsetL) / 6)) THEN MistCont = 32: fanstage = 0: ventstage = 1'Mist on
  IF RHdemo >= RHsetL THEN MistCont = 0
'   IF RHdemo < RHsetL THEN MistCont = 32: fanstage = 0: ventstage = 0
END IF

IF RHdemo > RHsetH THEN ventstage = ventstage + 1: fanstage = fanstage + 1
END SUB
```

```
SUB RVdown
    relay2 = 16
    CALL devices2
    SLEEP 3
    relay2 = 0
    CALL devices2

END SUB
```

```
SUB RVup
   relay2 = 32
   CALL devices2
   SLEEP 3
   relay2 = 0
   CALL devices2
END SUB
```

```
SUB sensors

'A/D Control and read (via the PCLD-711S)

'Note Range Codes for A/D:
'    0=+/-5.0000
'    1=+/-2.5000
'    2=+/-1.2500
'    3=+/-0.6250
'    4=+/-0.3125

'AD592 Sensors First.......
     OUT BaseAdd + 9, 4:   'Range set for +/- 0.3125 vdc
```

```
    OUT BaseAdd + 10, 7:  'Scan SE channel 7
    OUT BaseAdd + 12, 10: 'Software trigger A/D conversion
    SLEEP 1
    Hi = INP(BaseAdd + 5) * 256
    Lo = INP(BaseAdd + 4)
    RawBinary = Lo + Hi
    vdc = (RawBinary / 4096) * .625 - .3125
    Temp7 = ((vdc / .000698) - 273.2) * 1.8 + 32

    OUT BaseAdd + 10, 6:  'Scan SE channel 6
    OUT BaseAdd + 12, 10: 'Software trigger A/D conversion
    SLEEP 1
    Hi = INP(BaseAdd + 5) * 256
    Lo = INP(BaseAdd + 4)
    RawBinary = Lo + Hi
    vdc = (RawBinary / 4096) * .625 - .3125
    Temp6 = ((vdc / .000698) - 273.2) * 1.8 + 32

'RH and Temperature Transmitter Last.......
    OUT BaseAdd + 9, 2:   'Range set for +/- 1.2500 vdc
    OUT BaseAdd + 10, 0:  'Scan SE channel 0
    OUT BaseAdd + 12, 10: 'Software trigger A/D conversion
    SLEEP 1
    Hi = INP(BaseAdd + 5) * 256
    Lo = INP(BaseAdd + 4)
    RawBinary = Lo + Hi
    vdc = (RawBinary / 4096) * 2.5 - 1.25
    RH = vdc * 100

    OUT BaseAdd + 10, 1:  'Scan SE channel 1
    OUT BaseAdd + 12, 10: 'Software trigger A/D conversion
    SLEEP 1
    Hi = INP(BaseAdd + 5) * 256
    Lo = INP(BaseAdd + 4)
    RawBinary = Lo + Hi
    vdc = (RawBinary / 4096) * 2.5 - 1.25
'NOTE: Need to call Omega Regarding this calibration curve
'     It does appear to be correct though
    DryBulb = (90 * vdc - 20) * 1.8 + 32

END SUB
```

```
SUB wetbulb

    P = 101325!
    T = (Temp6 - 32) / 1.8
    RH = RH / 100!

    T = T + 273.2

'''' calculate the saturation vapor pressure

    IF ((T - 273.2) < 0!) THEN
      a1 = -5674.5359#
      a2 = 6.3925247#
      a3 = -9.677843E-03
      a4 = .000000622157#
      a5 = .0000000020747825#
      a6 = -9.484024000000001D-13
      a7 = 4.1635019#
    ELSE
      a1 = -5800.2206#
    a2 = 1.3914993#
    a3 = -.04864023#
    a4 = .000041764768#
    a5 = -.000000014452093#
    a6 = 0!
    a7 = 6.5459673#
   END IF

   t1 = T
   b1 = a1 / t1 + a2 + a3 * t1 + a4 * t1 ^ 2 + a5 * t1 ^ 3 + a6 * t1 ^ 4 + a7 * LOG(t1)
   pws = EXP(b1)
   pw = pws * RH

'''' calculate the dew point temperature

   td = -35.957 - 1.87261 * LOG(pw) + 1.1689 * (LOG(pw)) ^ 2

'''' calculate the humidity ratios

   w = .62198 * pw / (P - pw)
   hum = w
   ws = .62198 * pws / (P - pws)
   T = T - 273.2

'''' theory from the adiabatic saturator''''''''''''''''

  FOR tw1 = T + 273.2 TO T + 273.2 - 30 STEP -.01

   IF ((tw1 - 273.2) < 0!) THEN
    a1 = -5674.5359#
    a2 = 6.3925247#
    a3 = -9.677843E-03
    a4 = .000000622157#
    a5 = .0000000020747825#
    a6 = -9.484024000000001D-13
    a7 = 4.1635019#
   ELSE
    a1 = -5800.2206#
    a2 = 1.3914993#
    a3 = -.04864023#
    a4 = .000041764768#
    a5 = -.000000014452093#
    a6 = 0!
    a7 = 6.5459673#
   END IF

   b1 = a1 / tw1 + a2 + a3 * tw1 + a4 * tw1 ^ 2 + a5 * tw1 ^ 3 + a6 * tw1 ^ 4 + a7 * LOG(tw1)
   pwstw = EXP(b1)
   wstw = .62198 * pwstw / (P - pwstw)

   h = 1.006 * T + w * (2501 + 1.805 * T)
   hs = 1.006 * (tw1 - 273.2) + wstw * (2501 + 1.805 * (tw1 - 273.2))
   hw = 4.186 * (tw1 - 273.2)

   right = hs
   left = h + hw * (wstw - w)
   delta = right - left

   IF (delta <= .5) THEN
    TWetBulb = (tw1 - 273.2)
    GOTO 100
   END IF
  NEXT tw1

100

  TWetBulb = TWetBulb * 1.8 + 32
  RH = RH * 100
  DewPoint = td * 1.8 + 32
  AbsHum = w
  VaporP = pw

END SUB
```