# ENHANCEMENT OF NORTH CAROLINA AGRICULTURAL AUTOMATED WEATHER NETWORK AND DEVELOPMENT OF ADVANCED COMMUNICATION, DATA ACQUISITION, AND DISSEMINATION SYSTEMS

By

**VINAYAK K. PARAMESHWARA**

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the requirements for the degree of
Master of Science

**MARINE, EARTH AND ATMOSPHERIC SCIENCES**
**WITH CO-MAJOR IN**
**COMPUTER ENGINEERING**

Raleigh,
May, 2000

APPROVED BY:

Dr. Paul D. Franzon,
Co-chair of Advisory Committee

Dr. Katharine B. Perry

Mr. P. Allan Sadowski

Dr. Robert J. Wayland

Dr. Sethu Raman
Chair of Advisory Committee

# ABSTRACT

PARAMESHWARA, VINAYAK KOTA. Enhancement of North Carolina Agricultural Automated Weather Network and Development of Advanced Communication, Data Acquisition, and Dissemination Systems (Under the direction of Dr. Sethu Raman)

AgNET is a weather network, which collects surface and subsurface meteorological data in North Carolina. Each AgNet weather station measures a variety of weather parameters such as air temperature, relative humidity, wind speed and direction, barometric pressure, solar radiation, photosynthetically active radiation, soil temperature, soil moisture and precipitation. The station summarizes the above parameters for every hour and stores them in its local memory. Currently the data collected over the entire day are then transferred to a central server through phone lines. The data collected at the central server are published on the Internet and disseminated by other means as well. The purpose of this thesis is to describe the modernization of the AgNet weather network by making the network conform to standards, implement a uniform sensor configuration and improve the methods of data acquisition, dissemination and display of weather data.

Since phone-based communication is not economical for real time data transfers other techniques such as Radio Frequency (RF) communication and satellite based communication were evaluated. RF communication was the most promising communication technique. Since RF is limited by distance, a combination of RF and Internet was designed. In this type of communication, data are transmitted from the weather station and received by a base station that is on the Internet. The received data are then forwarded to a central server at the State Climate Office where they are archived and disseminated. A successful test was conducted as proof of concept between Lake Wheeler road Field laboratory site, Raleigh and Varsity Laboratory, NC State University.

Since every system has inherent noise, data collected at each station have to go through different quality control algorithms to insure data quality. Different Quality Control (QC) algorithms were implemented which are discussed in the thesis.

# BIOGRAPHY

Vinayak Parameshwara was born on 29$^{th}$ November 1973 in Karnataka, India. He obtained elementary and middle school education in Sri Vidya Mandir, Bangalore, India. He later obtained his high school education in Seshadripuram Pre University College, Bangalore, India. He did his undergraduate degree with a major in Instrumentation Technology in R. V. College of Engineering, Bangalore in 1995.

After completing his degree, he worked in the Indian Institute of Science, Bangalore where he designed and developed instruments and dataloggers for measuring air pollution. He also developed the supporting hardware and software for measuring turbulence on a wing blade.

Since August 1997 he has been working as a Graduate Research Assistant at North Carolina State University. He has been actively involved in designing and developing a statewide weather network, which can provide data in real-time. Also he is involved in developing software for publishing the collected data on to the web.

# DEDICATION

*To my parents Appaih and Amma*

ACKNOWLEDGEMENTS

I would like to thank Dr. Sethu Raman, my advisor and chair of my graduate committee for his guidance, support, kindness, patience and encouragement throughout my graduate study. I would like to thank Dr. Paul Franzon, co-chair of my graduate committee for serving on my thesis committee. I would like to thank Dr. Katharine Perry, for her guidance and support during my graduate study and also for serving on my thesis committee. I would like to thank Allan Sadowski and Dr. Robert Wayland for their support and serving on my thesis committee.

I thank Devdutta Niyogi for his suggestions, patience and support throughout my graduate study and especially during the installation of AgNet stations. I thank Ryan Boyles for his support and feedback during the development of AgNet. I thank Syed Ameenulla for providing me with valuable information during modernization of AgNet.

Funding for the enhancements and operations of the AgNet was provided by the North Carolina Agricultural Research Services (NCARS), headed by Dr. Johnny Wynne, Associate Dean, College of Agriculture and Life Sciences.

I am thankful to all State Climate Office and AirSea Lab colleagues: Bic Fort, Aaron Sims, Jenifer Kehoe, Robert Gilliam, Wendy Seller, Hasnah Hassan, Orbita Roswintiarti and Kettyah Chhak for their support during the entire graduate study.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF EQUATIONS

# CHAPTER 1

# INTRODUCTION

Weather data help us in developing a better understanding of the intricate yet comprehensible climate. In order to understand the weather systems, different parameters such as air temperature, relative humidity, wind speed, wind direction, barometric pressure, radiation, precipitation, soil temperature and soil moisture must be known. These parameters collectively play a major role in determining the weather conditions. In order to measure these parameters one has to convert them into a form that can be processed and measured. The device that converts the parameter from one form to another is known as a sensor. The output of these sensors is digitized using an analog-to-digital converter on a datalogging device and is stored in the datalogger. These sensors and devices are collectively known as an Automated Weather Station (AWS). The digitized data on the datalogger is then transferred to a personal computer (PC) host, where it is stored and archived for further processing. Collections of these automated weather stations are known as a weather network and AgNet is one such network.

## 1.1 What is AgNet?

AgNet is a weather network that is also known as the North Carolina Agricultural Weather Network. Currently AgNet is a group of seventeen automated weather stations for collecting weather data throughout North Carolina. These automated weather stations are located at the outlying agricultural research stations and field laboratories of the College of Agriculture and Life Sciences that are spread throughout the state.

## 1.2 History of AgNet

The North Carolina Agricultural and Research Service (NCARS) started the AgNet in 1978. Since its inception, components of this network were gradually installed between 1978 and 1987. Stations were located at a field laboratory and at thirteen of the fifteen research stations of North Carolina State University and the North Carolina Department of Agriculture and Consumer Services. Since there were no suppliers of automatic weather stations at that time, original stations were designed and built by NC State University's Department of Biological and Agricultural Engineering under the leadership of Dr. James H. Young.

In April 1991, the supervision of the AgNet was transferred to the NC Agricultural Weather Program, administered through the Department of Horticultural Science in the College of Agriculture and Life Sciences (CALS). Between May 1992 and August 1993, stations were upgraded with more advanced data loggers and sensors.

In July 1997 the network was transferred to the State Climate Office of North Carolina under a cooperative arrangement between the State Climate Office and the North Carolina Agriculture Research Service. Since the integration of the network into the State Climate Office operations, the stations have been upgraded to include 10m towers and the addition of soil moisture sensors.

## 1.3 Locations of AgNet sites

The NCARS weather network was initially intended to have a weather station at each of the fifteen research stations of NCARS. There are currently 18 automated AgNet stations throughout the state. By 1997, 15 of these research stations had automated weather networks installed, while two stations such as Laurel Springs and Reidsville needed new ones. US Department of Agriculture (USDA) Tide Water research station already has an Automated weather station. The geographical locations of these stations are as shown below in Figure 1.1.

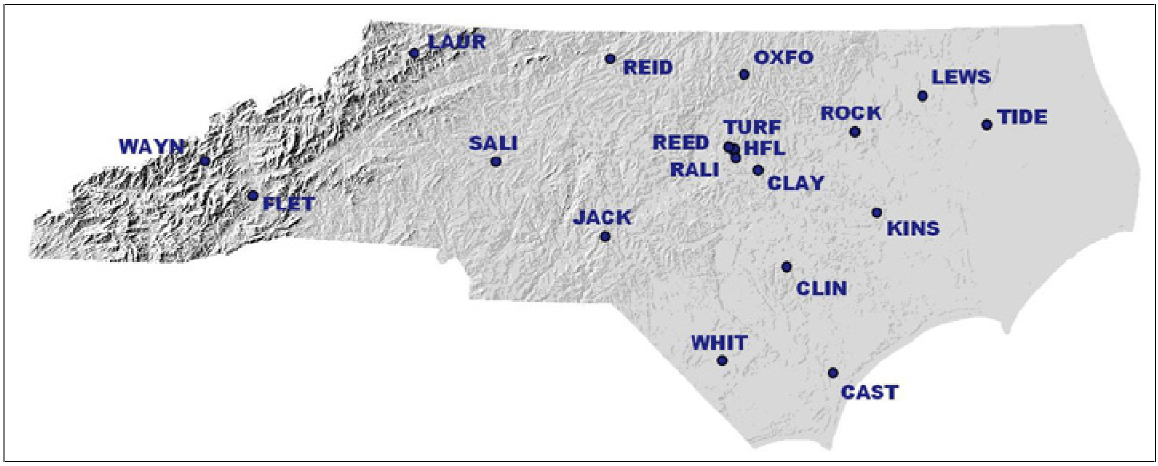



**Figure 1.1:** Location of the AgNet Weather Stations

The list of AgNet Stations and the station ID is as shown below in Table 1-1.

**Table 1-1 :** List of AgNet Station Names and Station IDs

| **Station Name, City** | **Station ID** |
|---|---|
| Border Belt Tobacco Research Station, Whiteville | WHIT |
| Central Crops Research Station, Clayton | CLAY |
| Cunningham Research Station, Kinston | KINS |
| Horticultural Crops Research Station, Castle Hayne | CAST |
| Horticultural Crops Research Station, Clinton | CLIN |
| Horticulture Field Laboratory, Raleigh | HFL |
| Lake Wheeler Road Field Laboratory, Raleigh | LAKE |
| Mountain Horticultural Crops Research Station, Fletcher | FLET |
| Mountain Research Station, Waynesville | WAYN |
| Oxford Tobacco Research Station, Oxford | OXFO |
| Peanut Belt Research Station, Lewiston | LEWS |
| Piedmont Research Station, Salisbury | SALI |
| Reedy Creek Field Laboratory, Raleigh | REED |
| Sandhills Research Station, Jackson Springs | JACK |
| Turfgrass Field Laboratory, Raleigh | RALI |
| Upper Coastal Plain Research Station, Rocky Mt. | ROCK |

| | |
|---|---|
| Upper Mountain Research Station, Laurel Springs | LAUR |
| Upper Piedmont Research Station, Reidsville | REID |

## 1.4 Parameters monitored by AgNet sites

AgNet measures a variety of weather parameters that include:

- Air Temperature
- Relative Humidity
- Wind Speed
- Wind Direction
- Solar Radiation
- Photosynthetically Active Radiation
- Barometric Pressure
- Precipitation
- Soil Temperature
- Soil Moisture

Depending upon the parameters, the data are further processed in the datalogger to output instantaneous, averaged, maximum/minimum or standard deviation values. Additional parameters such as soil moisture and new pressure sensors have been incorporated into the network since the integration of AgNet into the State Climate Office operations.

## 1.5 Current standards for Weather Monitoring

Since AgNet measures weather parameters, the parameter measured must conform to international standards in order for the data to be ingested into weather models for further analysis. Measuring parameters under different standards such as varying heights for wind measurement complicates use of the data. Also measuring parameters in less than ideal conditions can make the measurement invalid and thus useless data. There are many organizations which put forth measurement standards, the most prominent of these are:

- World Meteorological Organization (WMO)
- Environmental Protect Agency (EPA)
- Association of American State Climatologists (AASC)

These organizations each have different measurement standards and if conformed to these standards, make the data easier to compare and use. These standards contain a collection of guidelines, which indicate the following

- Instrument siting procedures;
- Instrument maintenance procedures such as calibration procedures; and,
- Sampling and averaging intervals.

A brief list of standards for instrument siting set forth by WMO [44] is listed below in Table 1-2.

**Table 1-2:** World Meteorological Organization (WMO) standards for weather monitoring.

| Parameter | Sensor Height/Depth | Siting Details |
|---|---|---|
| Air Temperature and Relative Humidity | 1.25m to 2m | Housed in a ventilated radiation shield to protect measurement from being corrupted by thermal radiation |
| Wind Speed and Wind Direction | 10m | The instrument should be away atleast ten times the height of the nearest obstruction. |
| Radiation | | Unrestricted view of sky and no obstruction above the 10° horizontal plane. |
| Soil Temperature and Soil Moisture | 0.05m, 0.1m, 0.2m, 0.5m and 1m | In area with atleast $1m^2$ of soil of same kind. The ground should level for 10m radius. |
| Precipitation | Minimum of 0.3m | The instrument should be away atleast four times the height of the nearest obstruction. Orifice of the gauge should be exact horizontal plane |

A brief list of standards for instrument siting set forth by EPA [15] is listed below in Table 1-3.

**Table 1-3:** Environmental Protection Agency (EPA) standards for weather monitoring.

| Parameter | Sensor Height/Depth | Siting Details |
|---|---|---|
| Air Temperature and Relative Humidity | Single Level: 2.0m<br>Two Level: 2m & 10m | Housed in a ventilated radiation shield to protect measurement from being corrupted by thermal radiation. Also the sensor should be away atleast four times the height of the nearest obstruction and should be away atleast by 30m from paved roads and parking lots. |
| Wind Speed and Wind Direction | 10m | The instrument should be away atleast ten times the height of the nearest obstruction. |
| Radiation | | Unrestricted view of sky and no obstruction above the 10° horizontal plane. |
| Soil Temperature and Soil Moisture | 0.05m, 0.1m, 0.2m, 0.5m and 1m | In area with atleast $1m^2$ of soil of same kind. |
| Precipitation | Minimum of 0.3m | The instrument should be away atleast four times the height of the nearest obstruction. Orifice of the gauge should be exact horizontal plane. Also the sensor should be away atleast four times the height of the nearest obstruction. |

A brief list of standards for instrument siting set forth by AASC [41] is listed below in Table 1-4.

**Table 1-4:** American Association of State Climatologists (AASC) standards for weather monitoring.

| Parameter | Sensor Height/Depth | Siting Details |
|---|---|---|
| Air Temperature and Relative Humidity | 1.5m±1.0m | Housed in a ventilated radiation shield to protect measurement from being corrupted by thermal radiation |
| Wind Speed and Wind Direction | Single Level: 3m±0.1m Two Level: 2m±0.1m,10m± 0.5m | The instrument should be away atleast ten times the height of the nearest obstruction. |
| Radiation | | Unrestricted view of sky and no obstruction above the 10° horizontal plane. |
| Soil Temperature and Soil Moisture | 0.1m±0.01m | In area with atleast $1m^2$ of soil of same kind. The ground should level for 10m radius. |
| Precipitation | 1.0m±0.2m | The instrument should be away atleast four times the height of the nearest obstruction. Orifice of the gauge should be exact horizontal plane. Also the sensor should be away atleast four times the height of the nearest obstruction |

## 1.6 Use of AgNet

Weather data are playing an increasing role in making major decisions. Since AgNet collects weather data, it has a wide variety of applications varying from individual users to North Carolina Department of Emergency Management and Rescue during crisis situations.

### 1.6.1 Agricultural Applications

Since AgNet was originally conceived to provide weather data to agricultural scientists and farmers, there are a variety of applications and uses for AgNet weather data. Some of the derived products from AgNet data that can used for agricultural applications are growing degree-days, evaporation rate, rainfall and soil moisture.

During days of insecticide and pesticide spraying, the farmer needs wind data to effectively apply pesticides. Using the AgNet real-time network it is possible for a farmer to get near real-time winds to perform this operation. During days of irrigation, the farmer needs soil moisture, precipitation or growing degree-day data. Using the AgNet it is possible to get the data and make correct decisions. Certain agricultural models require growing degree-days as one of the inputs.

### 1.6.2 North Carolina Emergency Management and Rescue (NCEMR)

Hurricane Floyd made landfall on the coast of North Carolina on September 14, 1999 with wind gust exceeding 110mph. After landfall, the North Carolina Emergency Management and Rescue Division needed to take necessary steps to rescue people and livestock that were adversely affected by Hurricane Floyd. One of the urgent steps to be taken was to drop food and other necessary rescue equipment in the affected area. In order to perform this operation, the affected area should be weather friendly for either helicopters or planes to land. AgNet stations could provide this information in real time as they did during Hurricane Floyd (1999) for Tarboro rescue operations.

AgNet towers are reasonably robust and strong. They could provide weather data in real time during the landfall of Hurricane Bonnie (1998), Dennis (1999), and Floyd (1999). This data were disseminated over the State Climate Office web site every hour. The North Carolina State Management and Rescue division accessed to these data on the web and was able to use them for their operations.

### 1.6.3 Pesticide Control and Air Quality Monitoring

After the passage of Hurricane Floyd, resulting heavy rains created areas of stagnant water. These areas of stagnant water and swollen lakes created a perfect conditions for breeding of infectious pests such as mosquitoes and houseflies. Immediate steps had to be taken by the state Pesticide control officials to avoid a major catastrophe of infectious diseases. In order to control these pests, pesticides had to be sprayed over a large area in a short duration of time.

Spraying pesticides over a large area of land using aircraft can create problems if winds are not favorable. Due to unfavorable winds, the pesticides sprayed on an area to eliminate pests can either land in an area inhabited by humans or livestock. Hence in order to do efficient and safe spraying, it is necessary to have real-time winds to avert such problems. Using the AgNet real-time data network it was possible for the Pesticide Control and Division of Air Quality to monitor winds during this activity.

During the summer of every year, the rate of surface Ozone ($O_3$) generation increases rapidly due to photochemical oxidation. North Carolina Division of Air Quality does a regular forecast of ozone during this time. In order to do this forecast, the Division of Air Quality needs hourly surface winds and temperature. Using the AgNet real-time data network, Division of Air Quality has easy access to hourly surface winds and air temperature throughout the state.

## 1.7 Motivation

The AgNet data disseminated in real-time has a wide variety of applications as discussed above. But, AgNet data collected prior to 1997 had some shortcomings such as

- Incorrect exposure of instruments;
- Data dissemination interval was monthly;
- Non-uniform sensor configuration in the network; and,
- Absence of soil moisture data

Most of the AgNet stations were located near an office building for electrical and telephone line access. A tripod was installed with a fenced area of about 10'X10' and the ground was covered with granite stones. On the tripod, air temperature, relative humidity and radiation sensors were installed and soil temperature was installed at a depth of approximately 0.1m. The rain gauge was installed over a separate galvanized iron (GI) pipe at a height of 1m. The wind sensors were generally mounted on the roof of an adjacent building at a height between 20’ and 60’.

The land heating effect due to granite stones on the ground increases the surface temperature and thus affects air and soil temperature readings. Since wind sensors were over the roof of a office building, the wake created by the building generally corrupted the wind measurements. Thus the data collected by these sites did not comply with the standards set forth by the organizations mentioned earlier.

The sensors used in the AgNet stations were purchased from Campbell Scientific Incorporated, Logan, Utah and based on availability a variety of sensors were used. To measure air temperature and relative humidity, two kinds of sensors were used: CS 500 or HMP 35C air temperature/relative humidity probe. To measure barometric pressure either Setra Model 270 or Yellow Springs Model 2014 was used. By using different sensors to measure the same parameter, the calibration procedures and data validations do get more complicated.

Data dissemination was done manually and hence the data were disseminated on the web only monthly through the College of Agriculture and Life Sciences web page. The data disseminated were the daily summaries of various parameters collected by the AgNet.

The soil moisture parameter is required by many applications in the fields of Agriculture and Meteorology. Many weather models require soil moisture for setting up the initial conditions for the model. Estimation of evapotranspiration also requires soil moisture data.

Since there is a great need for weather data, improving and enhancing AgNet would make it an ideal network for obtaining real-time data for the applications mentioned earlier. In order to improve and enhance AgNet certain procedures were adopted and implemented stage by stage. The procedures adopted were:

- Modernization of AgNet with uniform data;
- Addition of soil moisture sensors;
- Real-time communication; and,

- Real-time data dissemination.

In order to overcome incorrect exposure of sensors, it was decided to upgrade all stations to meet the WMO standards [44]. Since WMO recommends winds be measured at 10m height, it was decided to install a weather tower of 10m height at each location. It was decided that the sensors installed on all these sites would be uniform, so that calibration and redundancy would not be a major issue. Also by using uniform sensors, it was possible to get uniform data format.

Since there is a great need for soil moisture data, it was decided to add a soil moisture instrument at each remote site. The soil moisture could be indirectly measured using either a water content reflectometer or soil potential. Water content reflectometer was preferred over soil potential. Soil potential is measured using a gypsum block instrument, which is not an accurate measurement and varies relative to soil types.

Since the uses of AgNet data (as mentioned earlier) need real-time acquisition of data and dissemination, better communication and dissemination techniques had to be implemented or developed. Different communication techniques were evaluated on the basis of current requirements. In order to have better dissemination techniques, software was developed that converted data into Hyper Text Markup Language (HTML) pages and was executed at regular intervals of arrival of new data.

## 1.8 Organization of the thesis

In chapter 2 will be a discussion about first steps in the direction of modernization of AgNet. We will be discussing about different components and procedures involved in modernizing AgNet. In Chapter 3 we will be discussing the additional enhancements to the AgNet to achieve real-time data transfer and dissemination. In Chapter 4 we will be discussing different quality control procedures that are available and the implementation of these procedures. Typical data sets from AgNet are presented in Chapter 5. A summary of this research and future work in enhancing AgNet are given in Chapter 6. Sample source code, documented configuration file, sample web pages and sample quality control constants are provided in Appendices.

# CHAPTER 2

# COMPONENTS OF THE ENHANCED AUTOMATED WEATHER STATION

One of the goals for enhancing the automated weather station is to have good quality of data set. Having a uniform dataset helps achieve this objective. Uniform sensor configuration or having the same kind of instruments throughout the network is thus important. In this chapter, the techniques for achieving uniform dataset and a description of the complete network from sensor selection to dissemination techniques are presented.

## 2.1 Types of Instruments and Datalogger

As mentioned in the previous chapter AgNet measures a variety of parameters. In order to measure each parameter a separate instrument or sensor is required. A sensor can be widely thought of as a device, which converts energy from one form to another. A brief description of each instrument and its specifications follow.

### 2.1.1 Air Temperature

Air temperature or ambient temperature is measured using a Campbell Scientific HMP 45C. This is a combined air temperature and relative humidity instrument as shown in Figure 2.1. The instrument is based on variation of resistance of the platinum wire with temperature. The resistance of the platinum resistance detector varies linearly with the temperature. Thus by either applying a current source or connecting the detector across a potential divider, it is possible to measure temperature as a variation of voltage. The platinum resistance detector is preferred over other kinds of sensors such as thermocouples or thermistors due to their inherent linearity. Also using a sensor like a thermistor or a thermocouple that has an insulating bead to protect the sensor from corrosion will increase the response time of the instrument. The on-board electronics of the instrument converts the variations of the temperature into voltage variation and is available as an output from the sensor. Using equation 2.1 it is possible to convert the output voltage of the sensor to measured air temperature in degree celsius. Typical specifications of the sensors are shown below in Table 2-1. A detailed description of the instrument characteristics and specification can be found in Campbell Scientific HMP 45C Instruction Manual [22].

Relationship between output voltage and measured temperature is given by:

$$T = 0.1V_{out} - 40 \tag{2.1}$$

where: T is the temperature and

$V_{out}$ is the output voltage of sensor in millivolts

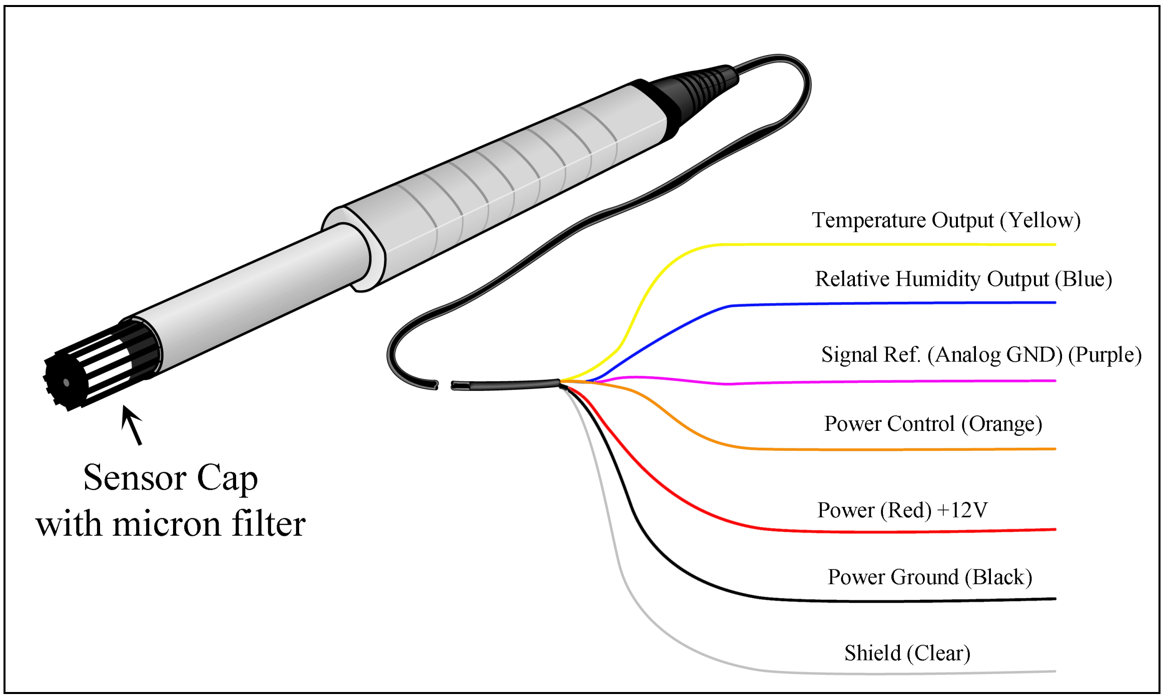



**Figure 2.1:** Campbell Scientific HMP 45C Air Temperature and Relative Humidity sensor.

**Table 2-1:** Typical specifications of Campbell Scientific HMP 45C Air Temperature

| **Parameter** | **Value** |
|---|---|
| Sensor Type | 1000Ω Platinum Resistance wire |
| Measurement Range | -40°C to 60°C |
| Output Voltage Range | 0.008V to 1.0V |
| Accuracy | ±1% |
| Output Settling Time | 150ms |

### 2.1.2 Relative Humidity

Relative humidity is measured using the Campbell Scientific HMP 45C instrument. This is a combined air temperature and relative humidity sensor shown in Figure 2.1. The sensor is manufactured by Vaisala Incorporated and is known as Vaisala HumiCAP. The HumiCAP is a capacitor whose capacitance varies with the humidity. Variations in relative humidity cause variations in the dielectric between the capacitance plates which in turn varies the capacitance value of the capacitor. Either by applying an AC voltage through a resistive potential divider or by connecting the capacitance to an oscillator circuit one can measure the capacitance value of the capacitor. Knowing the capacitance value it is possible to determine relative humidity. Using equation 2.2 it is possible to get relative humidity in terms of output voltage of the sensor. Typical specifications of the relative humidity sensor are shown in Table 2-2. A detailed description of the sensor specification can be found in Campbell Scientific HMP 45C Instruction Manual [22].

Relationship between output voltage of the sensor and relative humidity is given by:

$$RH = 0.1V_{out} \tag{2.2}$$

where: RH is the relative humidity and

$V_{out}$ is the output voltage of the sensor in millivolts.

**Table 2-2:** Typical specifications of Campbell Scientific HMP 45C relative humidity sensor

| **Parameter** | **Value** |
|---|---|
| Sensor Type | Vaisala HumiCAP |
| Measurement Range | 0 to 100% |
| Output Voltage Range | 0.008V to 1.0V |
| Accuracy | ±2% (0-90% RH)<br>±3% (90%-100% RH) |

### 2.1.3 Solar Radiation

Solar radiation is measured using a Campbell Scientific LI 200X shown in Figure 2.2. The instrument is manufactured by Licor Incorporated and is based on the photovoltaic current generation technique. The instrument measures the incoming solar radiation only. The sensor is constructed out of a silicon photovoltaic detector with a cosine-corrected head. The sensor is factory calibrated for a daylight frequency spectrum of 400nm to 1100nm. It generates current based on the intensity of the solar radiation incident on it. The current generated is then converted to voltage by connecting across a precision resistor to get voltage variation. Thus, one can measure total solar radiation in terms of voltage. During night, due to RF noise the sensor generates negative current, which can be safely ignored and is not a measure of outgoing radiation. Using equation 2.3 it is possible to convert the output voltage of the sensor into solar radiation in Watts per square meter ($W.m^{-2}$). Typical specifications of the photovoltaic detector and the LI 200X instrument are shown in Table 2-3. A detailed description of the sensor specification and mounting can be found in Campbell Scientific LI 200X Pyranometer Instruction Manual [25].

Relationship between differential output voltage and solar radiation is given by:

$$SR = 200V_{diff} \qquad (2.3)$$

where: SR is the solar radiation

$V_{diff}$ is the differential output voltage of the sensor.

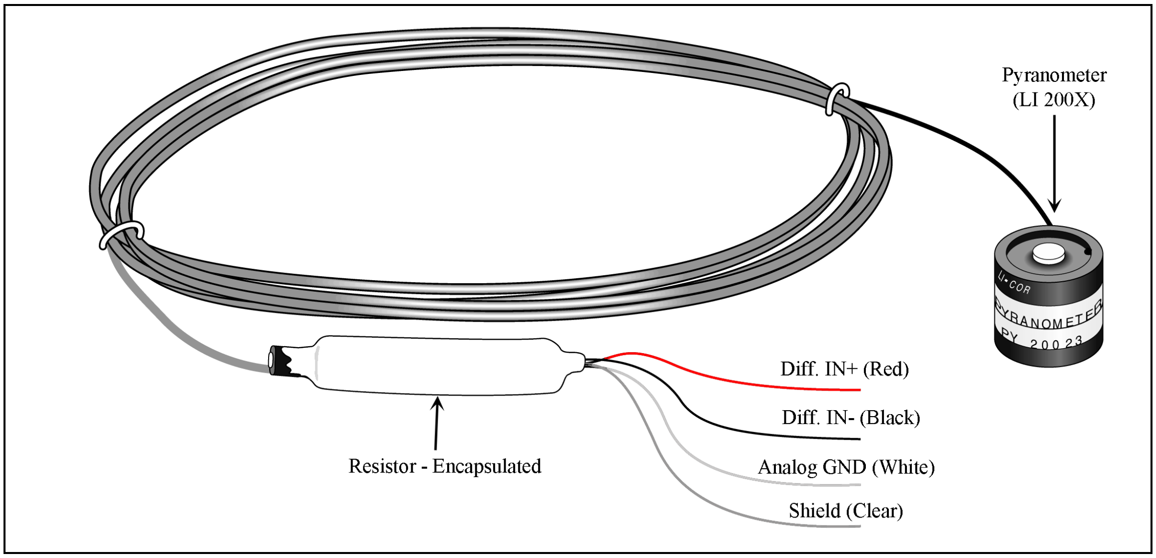



**Figure 2.2:** Campbell Scientific LI 200X Pyranometer Sensor.

**Table 2-3 :** Typical specification of Campbell Scientific LI200X Pyranometer

| **Parameter** | **Value** |
|---|---|
| Sensor Type | Silicon photovoltaic detector |
| Measurement Range | 0 to 3000W.m$^{-2}$ |
| Sensitivity | 0.2 kW.m$^{-2}$.mV$^{-1}$ |
| Linearity | ±1% up to 3000W.m$^{-2}$ |
| Accuracy | ±3% typical<br>±5% maximum |

### 2.1.4 Photosynthetically Active Radiation

Photosynthetically active radiation is measured using a Campbell Scientific LI 190SB shown in Figure 2.3. The instrument is manufactured by Licor Incorporated and is based on the photovoltaic detector technique. The sensor measures Photosynthetic Photon Flux Density (PPFD). The sensor is constructed out of a silicon photovoltaic detector (silicon diode) with a cosine-corrected head. A filter is placed in front of the detector which cutoffs the frequency spectrum greater than 700nm. The sensor is factory calibrated to measure daylight and artificial light in the spectrum range of 400nm to 700nm. The sensor generates a current, which varies with the incident PPFD. The generated current is then converted to voltage by inserting a precision resistor in the current loop and measuring the voltage across the resistor. Thus by measuring the voltage across the resistor one can estimate the PPFD. Using equation 2.4, one can convert the measured output voltage of the sensor to PPFD in mole.m$^{-2}$.day$^{-1}$. Typical specifications of the photovoltaic detector and the LI 190SB instrument are shown in Table 2-4. A detailed description of the sensor specifications and mounting can be found in Campbell Scientific LI 190SB Quantum Sensor Instruction Manual [24].

Relation between the output voltage and the PPFD (PAR) is given by:

$$PAR = kV_{out} \tag{2.4}$$

where: PAR is the PPFD in moles.m$^{-2}$.day$^{-1}$

k is a constant based on the factory calibration of the precision resistor and

$V_{out}$ is the output voltage of the sensor.

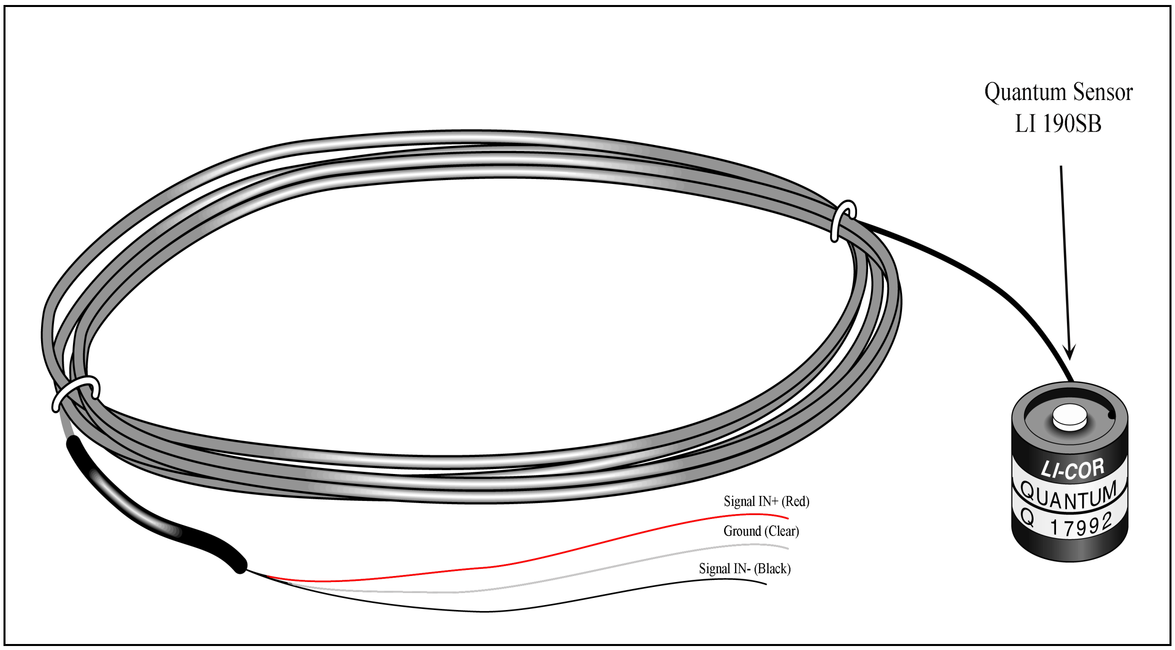



**Figure 2.3:** Campbell Scientific LI 190SB Quantum Sensor.

**Table 2-4:** Typical Specifications of Campbell Scientific LI 190SB Quantum Sensor

| **Parameter** | **Value** |
|---|---|
| Sensor Type | Silicon photovoltaic detector |
| Measurement Range | 0 to 10,000 μmoles.$s^{-1}$.$m^{-2}$ |
| Sensitivity | 0.005 μA per μmoles.$s^{-1}$.$m^{-2}$ |
| Linearity | ±1% up to 10,000μmoles.$s^{-1}$.$m^{-2}$ |
| Accuracy | ±5% typical |

### 2.1.5 Wind Speed

Wind speed is measured using the Campbell Scientific 05103 Wind Monitor. The instrument is manufactured by RM Young and is shown in Figure 2.4. The instrument is a combined wind speed and wind direction sensor. The wind speed sensor is based on a four-blade helicoid propeller. The propeller is attached to the armature of an Alternating Current (AC) generator. The angular speed of the propeller is proportional to the wind speed. Thus when wind rotates the propeller, the AC generator generates an AC voltage. The frequency of the generated AC voltage is proportional to the angular speed of the armature. Thus by measuring the frequency of the AC voltage one can evaluate the wind speed. Using equation 2.5 one can convert the frequency of the AC signal measured at the output of the wind speed sensor to wind speed in meters per second (m.$s^{-1}$). The typical specifications of the helicoid propeller and wind monitor are shown in Table 2-5. A detailed description of the sensor and its mounting can be found in Campbell Scientific 05103 and 05305 R. M. Young Wind Monitor [1]

Relation between the output frequency of the AC signal and wind speed is given by:

$$WS = 0.098F_{ws} \tag{2.5}$$

where: WS is the wind speed and

Fws is the frequency of the AC Signal at the output of the Wind speed sensor

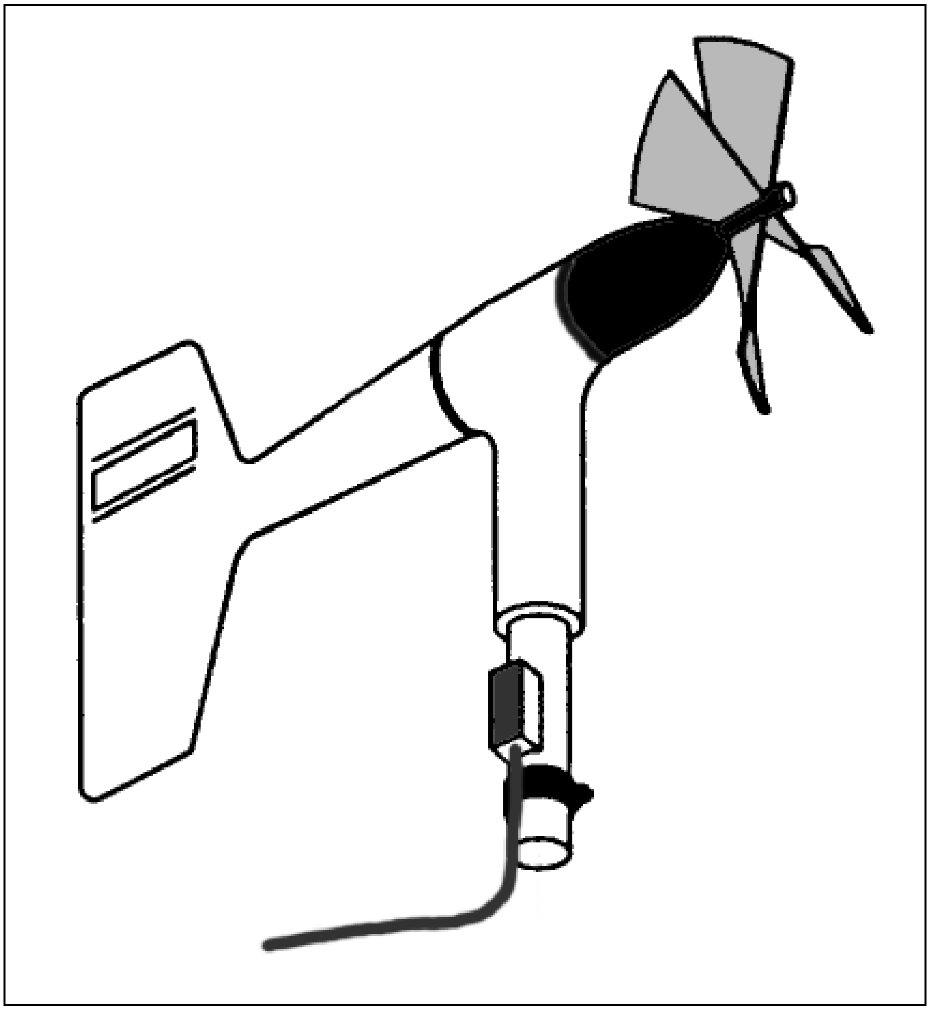


**Figure 2.4:** Campbell Scientific 05103 Wind Monitor.

**Table 2-5:** Typical specifications of Campbell Scientific 05103 Wind monitor

| **Parameter** | **Value** |
|---|---|
| Sensor Type | 4 blade helicod propeller on AC generator |
| Measurement Range | 0 to 60 $m.s^{-1}$ (sustained)<br>0 to 130 $m.s^{-1}$ (gust) |
| Threshold Sensitivity | 0.9 $m.s^{-1}$ |
| Distance Constant | 2.7m for 63% recovery |
| Output Frequency | 3 cycles per revolution<br>0.098 m.s-1 per hertz |

### 2.1.6 Wind Direction

Wind direction is measured using the Campbell Scientific 05103-wind monitor shown in Figure 2.4. The instrument is based on a balanced vane with a 360° single turn 10KΩ potentiometer. The balanced vane is mounted on bearings and is directly connected to the slider of the potentiometer. Thus by applying a reference voltage across the two ends of the potentiometer and measuring the voltage out of the center tap (slider), one can measure the wind direction. Using equation 2.6 it is possible to convert the output of the wind direction sensor in voltage into wind direction in degrees. Typical specifications of the 360° potentiometer and the wind direction instrument are shown in the Table 2-6. A detailed description of the wind monitor can found in Campbell Scientific 05103 Instruction Manual [1].

Relation between output voltage of wind direction sensor, excitation voltage and the wind direction is given by:

$$WD = \left(\frac{360}{V_{ex}}\right) V_{out} \tag{2.6}$$

where: WD is the wind direction in degrees

$V_{ex}$ is the Excitation Voltage set using the I/O instruction

$V_{out}$ is the output voltage of the wind direction sensor

**Table 2-6 :** Typical specifications of Campbell Scientific 05103 wind monitor

| **Parameter** | **Value** |
|---|---|
| Sensor Type | Balanced Vane on a 10kΩ potentiometer |
| Measurement Range | 0 – 360° (0-355° electrical) |
| Threshold Sensitivity | 1.0 $m.s^{-1}$ for 10° displacement<br>1.5 $m.s^{-1}$ for 5° displacement |
| Delay Distance | 1.3m for 63% recovery |
| Damping Ratio | 0.25 |

### 2.1.7 Barometric Pressure

Barometric pressure is measured using the Campbell Scientific CS105. The instrument is manufactured by Vaisala Incorporated and is shown in Figure 2.5. The instrument uses Vaisala's BaroCAP that is based on capacitive measurement techniques. Capacitive based pressure measurements are constructed by exposing one plate of the capacitor to atmosphere and other held firm with space between the two plates filled with air of known pressure. The distance between the two plates of the capacitor varies with the pressure on the other side of exposed plate. With varying distance, the capacitance of the capacitor varies. By either applying an AC voltage across the plates or connecting the capacitor to an oscillator tank, we can measure the capacitance of the capacitor. By knowing capacitance it is possible to evaluate barometric pressure. Using equation 2.7 it is possible to convert the output voltage of the sensor in volts to barometric pressure in millibars. The typical specifications of the BaroCAP and barometric pressure instrument are shown in the Table 2-7. A detailed description of the barometric pressure instrument can be found in Campbell Scientific CS105 barometric pressure sensor Instruction Manual [10].

Relation between the output voltage and the barometric pressure is given by:

$$BP = 0.184V_{out} + 600 \tag{2.7}$$

where: BP is the barometric pressure in millibars

$V_{out}$ is the output voltage of the barometric pressure sensor

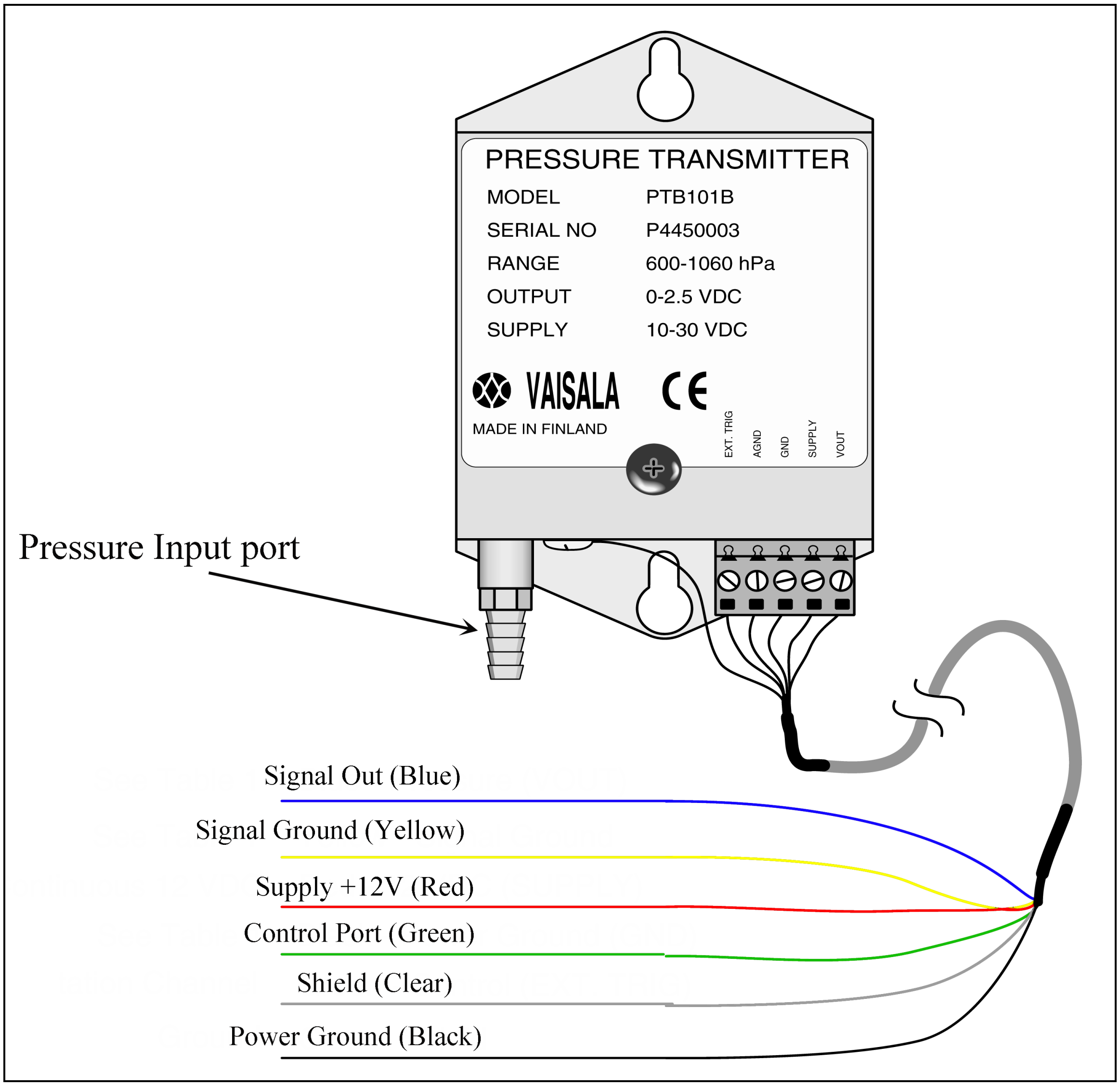



**Figure 2.5:** Campbell Scientific CS 105 Barometric pressure gauge sensor.

**Table 2-7:** Typical specifications of Campbell Scientific CS 105 barometric pressure gauge.

| **Parameter** | **Value** |
|---|---|
| Sensor Type | Vaisala's BaroCAP – Capacitance |
| Measurement Range | 600mb to 1060mb |
| Linearity | ±0.45mb at 20°C |
| Accuracy | ±0.5mb at 20°C<br>±2mb at 0°C to 40°C |

| | |
|---|---|
| | ±4mb at –20°C to 45°C<br>±6mb at –40°C to 65°C |
| Output Voltage Range | 0-2.5V |

### 2.1.8 Precipitation

Precipitation is measured using the Campbell Scientific TE 525 instrument. The instrument is manufactured by Texas Engineering and is shown in Figure 2.6. The instrument is based on National Weather Service's tipping bucket mechanism. The instrument has a funnel that collects the precipitation. The collected precipitation is directed towards a tipping mechanism that tips a definite amount of precipitation. Thus by counting the number of tips, one can evaluate the amount of precipitation for the interval. Using equation 2.8, it is possible to convert the pulse count per interval counted by the datalogger into precipitation in inches per interval. The typical specifications of the tipping mechanism and precipitation gauge are shown in Table 2-8. A detailed description of the precipitation gauge can be found in Campbell Scientific TE 525 Precipitation Gauge Instruction Manual [40].

Relationship between pulse count per interval and the precipitation per interval is given by:

$$\mathrm{Pr}\,ecip = 0.01P_{count} \tag{2.8}$$

where: Precip is the precipitation in inches per interval

$P_{count}$ is the count of the pulse per interval measured by the datalogger

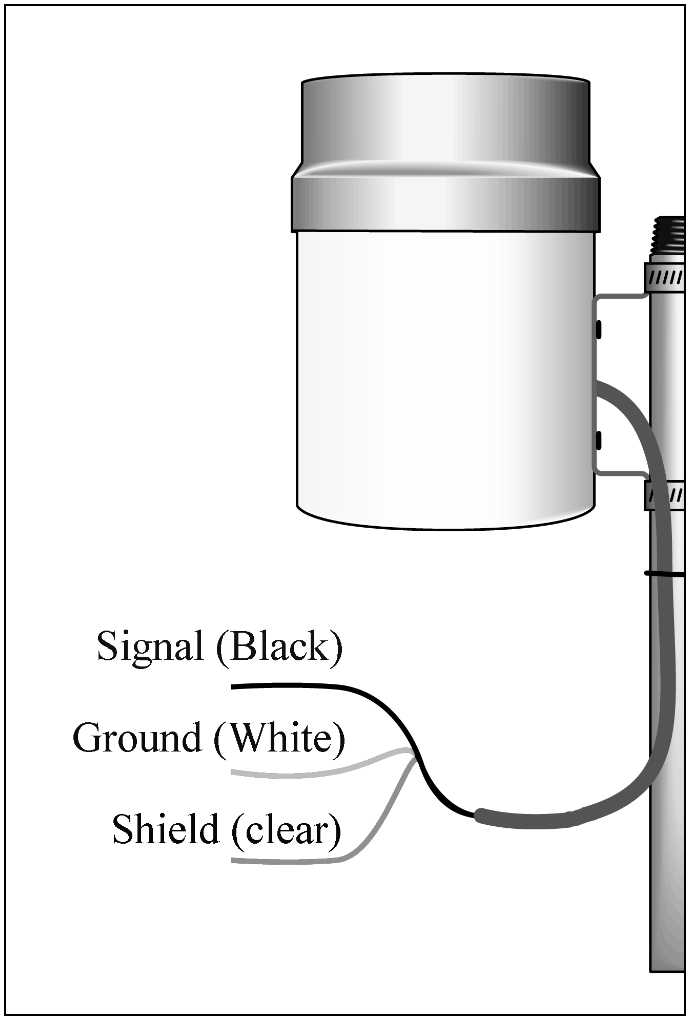



**Figure 2.6:** Campbell Scientific TE 525 precipitation gauge.

**Table 2-8 :** Typical specifications of Campbell Scientific TE 525 precipitation gauge.

| **Parameter** | **Value** |
|---|---|
| Sensor Type | National Weather Service Tipping Mechanism |
| Measurement Resolution | 1 tip |
| Rainfall per tip | 0.01in |
| Accuracy | ±1% for 0-1 in/hr<br>±3% for 1-2 in/hr<br>±5% for 2-3 in/hr |
| Funnel Size | 6 in collector |

### 2.1.9 Soil Temperature

Soil temperature is measured using the Campbell Scientific CS 107 instrument. The instrument is based on a thermistor sensor and the schematic diagram is shown in Figure 2.7. The thermistor probe is placed at the tip of the sensor and is encapsulated in a weatherproof covering. Weatherproof covering is crucial as any moisture in contact with the sensor can corrode the sensor rendering the instrument unusable. The resistance of the sensor varies with temperature. Thus, either by applying a current source across the sensors or by connecting the sensor across a potential divider, one can measure the resistance of the thermistor as a voltage. Knowing the resistance of the thermistor, it is possible to evaluate the temperature of the soil. Using equation 2.9, the output voltage of the sensor can be converted to measure soil temperature. The typical specifications of the thermistor and the soil temperature instrument are shown in Table 2-9. A detailed description of the soil temperature instrument can be found in Campbell Scientific Model 107 Temperature Probe Instruction Manual [28].

Relation between output voltage of the sensor and measured soil temperature is given by:

$$ST = -53.4601 + 90.807V_{out} - 83.257V^{2}_{out} + 52.283V^{3}_{out} - 16.723V^{4}_{out} + 2.211V^{5}_{out} \tag{2.9}$$

where: ST is the soil temperature

$V_{out}$ is the output voltage of the CS 107 sensor

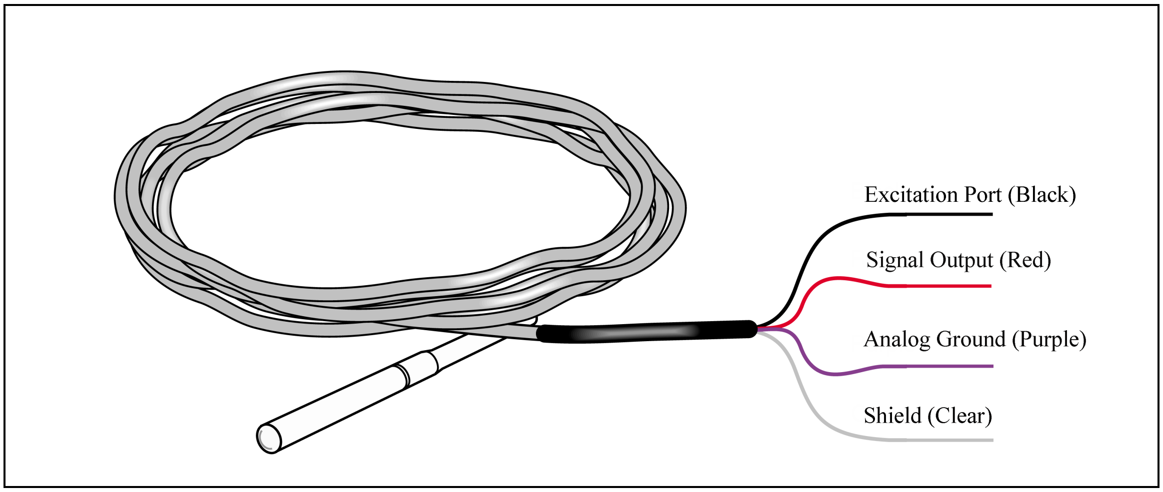



**Figure 2.7:** Campbell Scientific CS 107 Soil Temperature sensor.

**Table 2-9:** Typical specifications of Campbell Scientific CS 107 Soil Temperature probe.

| **Parameter** | **Value** |
|---|---|
| Sensor Type | Thermistor based measurement |
| Measurement Range | -35°C to 50°C |
| Linearity Error | Less than ±0.5°C over entire range |
| Accuracy | ±0.4°C for –24°C to 48°C<br>±0.9°C for –38°C to 53°C |
| Time Constant | 200 seconds ± 10 seconds |

### 2.1.10 Soil Moisture

Soil moisture is measured using the Campbell Scientific CS 615 Water Content Reflectometer shown in Figure 2.8. The instrument is based on time domain reflectometry (TDR) techniques. The dielectric constant of the soil varies with the water content information of the soil. Thus by measuring the dielectric constant one can measure the water content of the soil. In order to measure the water content, one of the techniques that can be used is time domain reflectometry. In this technique, a conduction rod made of stainless steel is inserted in the soil and a high frequency square wave is injected in the probes at one end. The signal travels through the rod and when it hits the other end the signal is completely reflected back. The time taken by the signal to traverse through the steel rod depends upon the dielectric material surrounding the rod. Thus by using timing circuits and measuring the delay, we can evaluate the soil water content. Using equation 2.10, it is possible to convert the average period measured by the CS 615 water content reflectometer to soil moisture in $m^3$ per $m^3$. The typical specifications of the soil moisture sensor are shown in the Table 2-10. A detailed description about the soil moisture instrument can be found in Campbell Scientific CS615 Water Content Reflectometer Instruction Manual [9].

Relationship between the average period and the soil water content (soil moisture) is given by:

$$SM = -0.187 + 0.037 P_{avg} - 0.335 P^2_{avg} \tag{2.10}$$

where: SM is the measured soil moisture in m3 per m3

$P_{avg}$ is the average period of the output of CS 615 Sensor in milliseconds

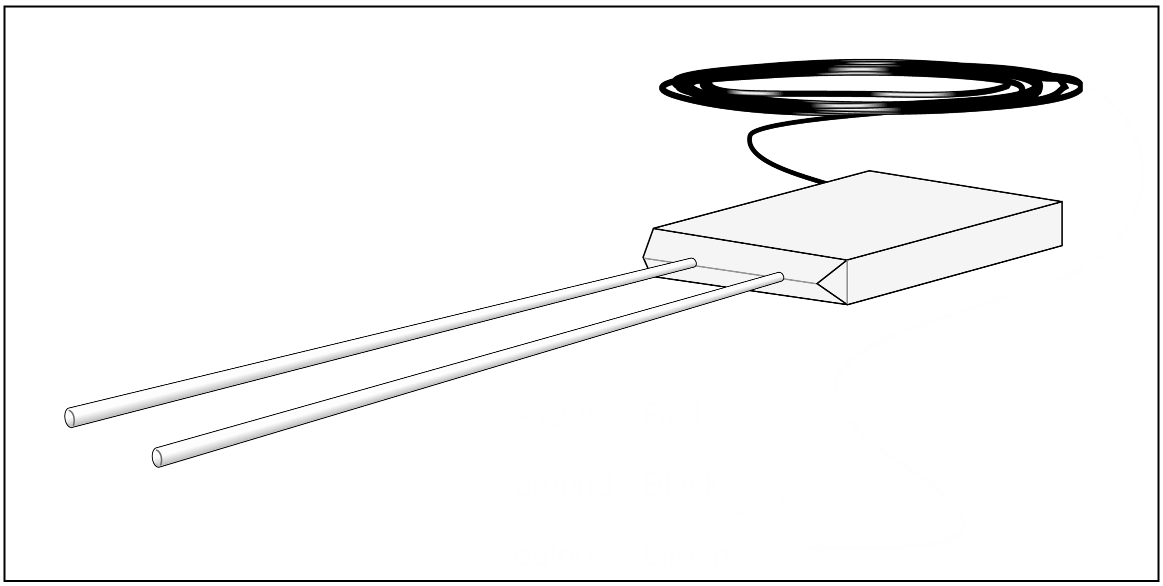


**Figure 2.8:** Campbell Scientific CS 615 Water content reflectometer.

**Table 2-10 :** Typical specification of Campbell Scientific CS 615 Water Content Reflectometer

| Parameter | Value |
|---|---|
| Sensor Type | Time Domain Reflectometry based period measurement |
| Measurement Resolution | $10^{-6}$ $m^3.m^{-3}$ |
| Accuracy | ±2% typical |

### 2.1.11 Solar Panels

Since the entire system is situated in a remote site, installing power cables to supply power to the system was expensive and not practical. Hence it was decided to install solar panels along with a regulator and a battery. Solar panels convert the solar energy to electrical energy. The generated power was regulated before supplying the power to dataloggers and sensors. Since solar panels fail during night, batteries were used to power the datalogger. The batteries were charged using the solar panels. Campbell Scientific supplied the solar panels, the +12V regulator and the battery. Typical specifications of the solar panels used for AgNet are shown in Table 2-11. A detailed description of the solar panels and mounting can be found in Campbell Scientific MSX10, MSX10R and MSX20R Solar Panels Instructions Manual [30].

**Table 2-11 :** Typical specification of Campbell Scientific MSX10/20R Solar panels

| Parameter | MSX10/MSX10R | MSX20R |
|---|---|---|
| Peak power | 10 W | 20 W |
| Voltage at peak power | 17.5 V | 17.1 V |
| Current at peak power | 0.57A | 1.17A |
| Average power | 9 W | 18 W |

### 2.1.12 Datalogger

The dataloggers used for AgNet are Campbell Scientific CR10/CR10X. This model of datalogger is programmable and can be programmed through PC software. The datalogger can accept twelve single ended (six differential ended) analog inputs and two pulse (digital) inputs. The datalogger can be connected to a computer/terminal through the communication port. The datalogger also has several control ports, which can be used to control devices such as the CS 615 water content reflectometer, CS 105 barometric pressure gauge and HMP 45C air temperature probe. The excitation ports provide a reference voltage for sensors that require a reference such as wind direction and soil temperature. The +12V sensor power port provides DC power for the sensors that require DC power, e.g. the air temperature, barometric pressure and soil moisture.

The analog and digital outputs of the sensors are connected to the datalogger through the wiring panel on the datalogger. The layout of the wiring panel is shown in Figure 2.9. The analog signals are digitized using a 13-bit successive approximation register type analog-to-digital converter (ADC). The pulse inputs are counted using either 8 bit or 16 bit digital counters. The digitized and pulse data are stored in the input storage location before it is processed. The data at the input storage is processed based on the program and is stored in the on-board memory available on the datalogger known as Final Storage Area (FSA). The Final Storage Area can be partitioned into two areas, which are known as FSA1 and FSA2. The datalogger also has 64 internal registers to store intermediate values.

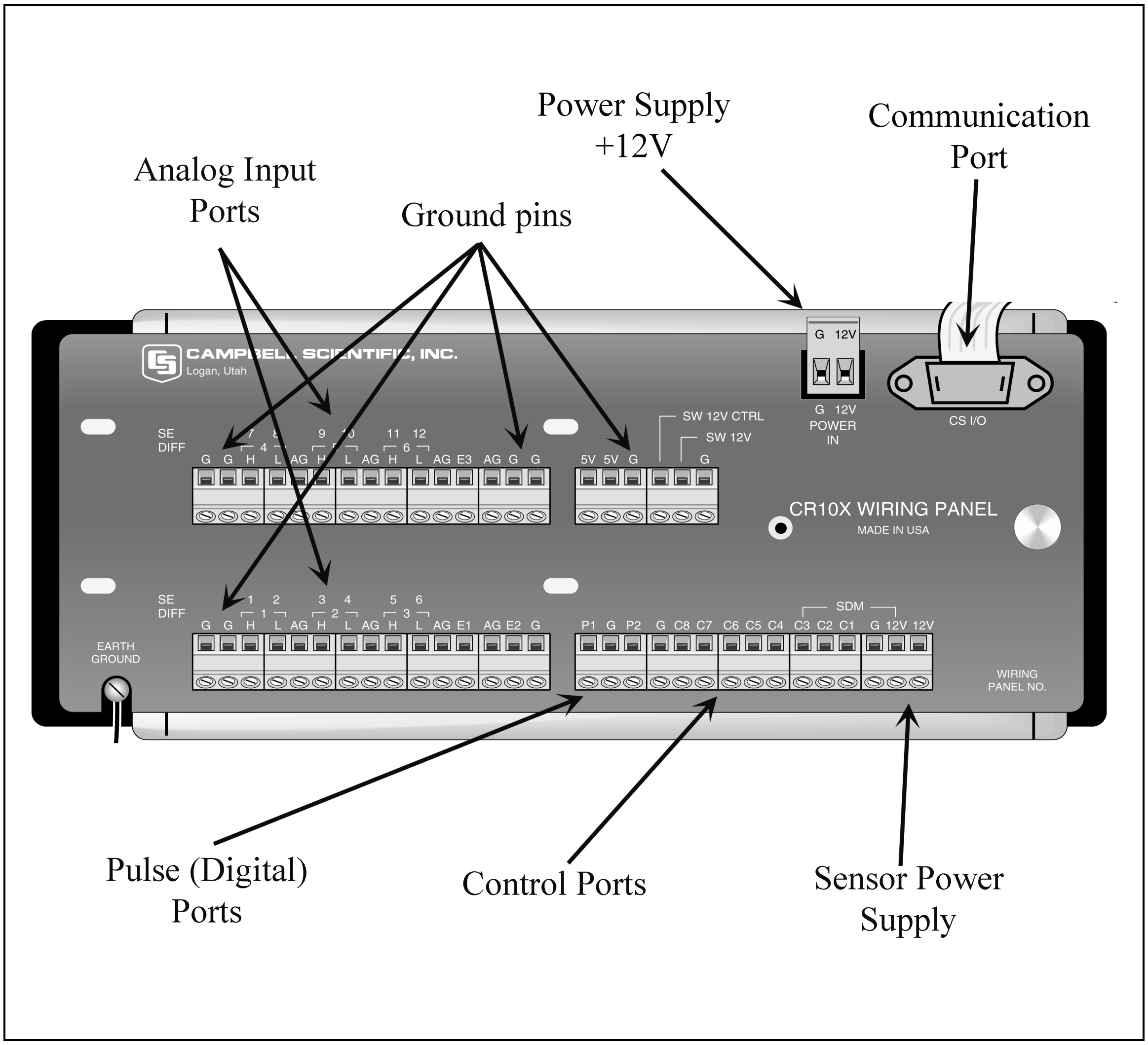



**Figure 2.9:** Wiring Panel of the Campbell Scientific CR 10X

The datalogger can be programmed either by connecting the keyboard or through a computer/terminal. The datalogger has 98 instructions, which can be broadly classified as

- Input/output instructions;
- Processing instructions
- Output processing instructions; and,
- Program control instructions

The Input/output instructions are used to gather either digitized analog data or counter output. Also these instructions can be used to influence the control and excitation ports. Processing instructions are used to process the collected data. These instructions include addition, subtraction, multiplication, division, square root, logarithm and exponential, statistical and trigonometric functions among others. The output processing instructions are used to output the processed data and the instructions include averaging, sampling, maximum/minimum, histogram, standard deviation and others. The program control instructions are used to control the flow of the program and these instructions include conditional statements, loop statements and others.

Using the instructions mentioned earlier, it is possible to program the datalogger to do a specific job. For the purpose of AgNet data acquisition, the datalogger was programmed to collect data and based on the parameter and output instructions, the datalogger outputs instantaneous, averaged, maximum or minimum or standard deviation. Final Storage Area 1 (FSA1) was used to store hourly and daily data while Final Storage Area 2 was used for five and fifteen minute data. Please refer to Appendix B1 for a detailed listing of the datalogger program used in the AgNet.

## 2.2 Site Selection

From previous chapters one can clearly see that site selection is important for good data collection. Since AgNet sites were all situated at research stations and field laboratories, which had barns, buildings or tall trees, choosing a good site for the weather station posed a challenge. Areas that were clear from obstructions were ideal for wind measurement but were not ideal for precipitation measurements, as measurements would be corrupted by wind effect. These two conditions were mutually exclusive to each other. A way around this problem was to have the rain gauge away from the tower site and close to bushes. Since this was not possible, as laying a sensor cable across fields was hazardous, rain gauges were mounted near the tower.

The data collected at the site had to represent a larger area around it, hence sites that were near obstructions such as buildings, tall trees and corns fields were avoided. Since areas with either a water pool or paved road within the vicinity of the tower corrupted either air temperature or the relative humidity measurement, these sites were also avoided. Also sites that were in either low-lying areas or had a history of flooding which would affect measurements of soil moisture and soil temperature were also avoided. Only sites that had a long fetch for the wind in the predominant wind direction were chosen.

At research stations such as Reidsville and Reedy Creek, the site was chosen inside a pasture as it satisfied the conditions of wind measurement and was protected by a fence.

## 2.3 Tower Installation

Once the site was chosen, a hole of 3'x3'x4' was dug to fit the base of the tower. The tower was made of aluminum and had three independent sections of 10' each. Aluminum towers were preferred over cast (galvanized) iron towers, as they last longer, are easier to handle and require less maintenance. Also since these towers were made of light aluminum, they needed no guys wires for support. Towers could be easily brought down and they could be tilted over at the base through a hinge for maintenance of wind sensors.

The base of the tower along with the lower section was lowered into the foundation. The tower was aligned using levels and care was taken to align the tower in a direction that had a clear area of around 10 meters (33 feet). Concrete was poured into the hole and was allowed to set. In order to support the lower section during concrete hardening, guy wires were attached. The whole setting was allowed to set for approximately one week. All three sections of the telescopic tower were attached and raised. With all the sections attached the tower height is approximately 35 feet as shown in Figure 2.10. On the top of the tower a lightning-arrester rod was installed and connected to the ground using a thick copper cable. The lightning arrester protected the tower from lightning by shorting the energy of the lightning to ground.

**Figure 2.10:** Typical view of the tower at Reedy Creek

## 2.4 Instrumenting the tower

Once the concrete had hardened the tower was ready to be instrumented. Before instrumenting the tower, the datalogger was installed. The datalogger was housed in an enclosure that protected the datalogger, battery, voltage regulator and pressure gauge from environmental corrosion. The enclosure was mounted approximately 1.5 m above ground to have easy access to the wiring. Instrumentation of the tower was done in stages. The order of instrumentation was:

- Wind speed and wind direction;
- Air temperature and relative humidity;
- Solar radiation and photosynthetically active radiation;
- Precipitation
- Soil moisture and soil temperature; and,
- Barometric pressure

After the installation of the sensors, solar panel was mounted facing south to get maximum power of the panel. After installing all the sensors and devices, electrical wiring was done. The wiring diagram of the tower is shown in Figure 2.19.

### 2.4.1 Installing Wind Speed and Wind Direction sensors

As per WMO Standards [44], wind speed and wind direction was mounted on the top of the tower at 10m (33 feet). Using a magnetic compass, magnetic North-South was determined. The crossarm sensor mount shown in Figure 2.11 was installed first and using levels the crossarm was made perpendicular to the ground. The sensor was then mounted on the crossarm using the base clamp. The sensor was then temporarily connected to either a datalogger or ohmmeter. The sensor was then aligned towards the magnetic North-South visually and then using the ohmmeter or datalogger, the base of the clamp was tightened at a position when the output of the wind direction sensor was zero. This assured that the wind direction sensor was aligned towards magnetic North-South.

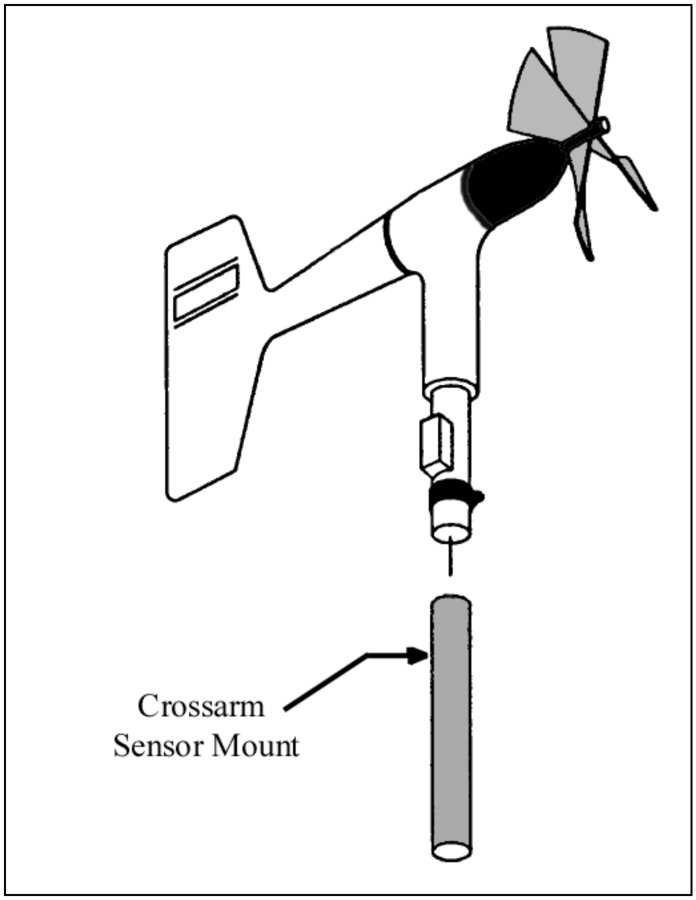



**Figure 2.11:** Installation of Campbell Scientific 05103 Wind Monitor.

### 2.4.2 Installing Air Temperature and Relative Humidity sensors

As per WMO Standards [44], air temperature and relative humidity should be mounted at 2 meters. To avoid effects of radiation on the air temperature sensor, it was mounted in a radiation shield. The radiation shield was made out of plastic and had 12 plates in it. Since the Humidity sensor is sensitive to environmental corrosion, a fine filter covered the probes inside the sensor. With the radiation shield and filter the air circulation around the sensor is reduced and also the response time. Hence these sensors are well suited only for slow measurements. In order to avoid heat induced by the tower, the radiation shield is mounted on a crossarm bar shown in Figure 2.12. The crossarm bar is then mounted on the tower at a height of 2m.

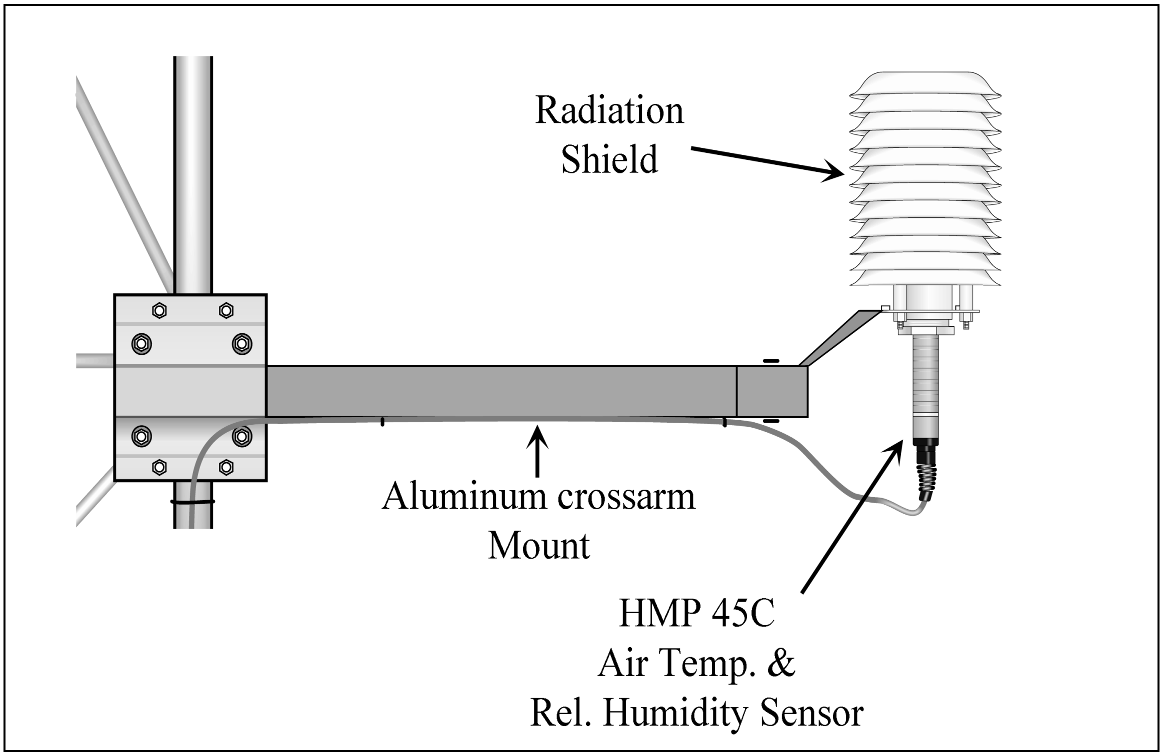



**Figure 2.12:** Installation of Campbell Scientific HMP 45C air temperature and relative humidity instrument.

### 2.4.3 Installing Solar Radiation and Photosynthetically Active Radiation Sensors

As per WMO Standards [44], solar radiation and photosynthetically active radiation should be mounted at a height between 1.5 and 2 m. The pyranometer (radiation sensor) for measuring incoming radiation should be in unrestricted view of the sky in all directions, and no obstructions above 10° of the horizontal plane. Locations where the shadow of the tower fell were avoided to avoid corruption of data. Mounting of the pyranometer to be level is critical, since the radiation incidence should be exactly horizontal. In order to have horizontal mounting, the sensor was installed on a leveling base as shown in Figure 2.13. Using the levels on the base, the sensor was mounted exactly horizontal with respect to the ground plane.

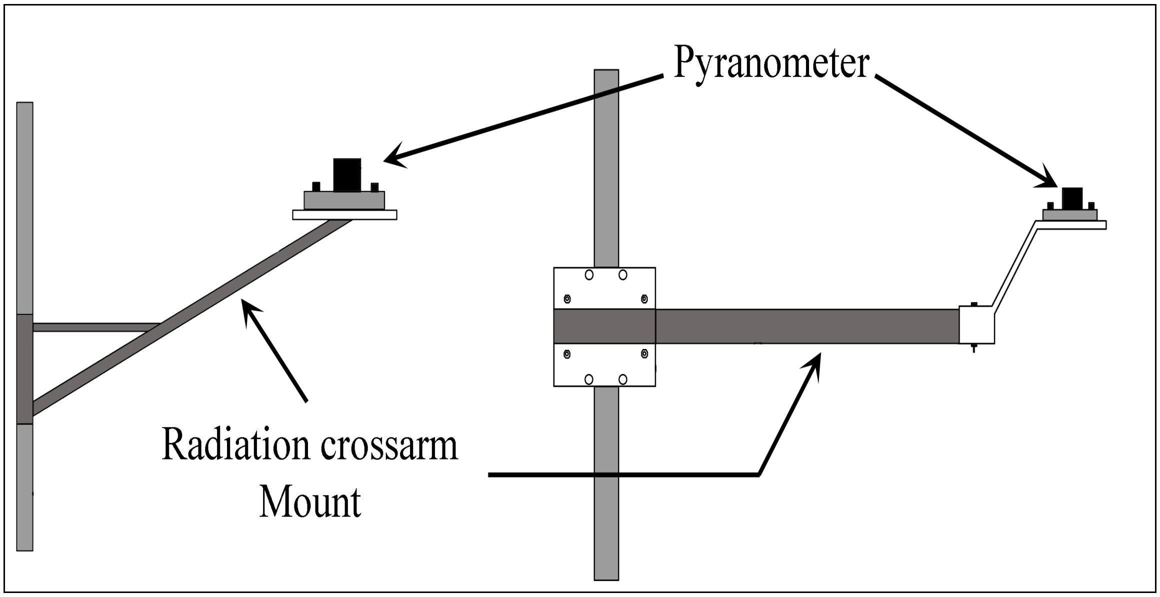



**Figure 2.13:** Installation of Campbell Scientific Pyranometer LI200X and Quantum sensor LI190SB.

### 2.4.4 Installing Precipitation Gauge

A separate galvanized iron (GI) pipe was installed to mount the precipitation gauge. Since the orifice of the precipitation gauge has to be exactly horizontal to avoid wind effects, mounting of the precipitation gauge is very crucial. In order to mount the gauge horizontal, the GI pipe was first installed exactly horizontal using a level. Later using mounting clamps the precipitation gauge was mounted as shown in Figure 2.14. The gauge was mounted at an approximate height of 1m for easy maintenance and also to avoid splashing and snow accumulation effects. As per WMO standards [44] the minimum height for precipitation measurement is 0.3m. The signal wire was buried under the soil to avoid any damage to the wire due to weathering or grass mowing.

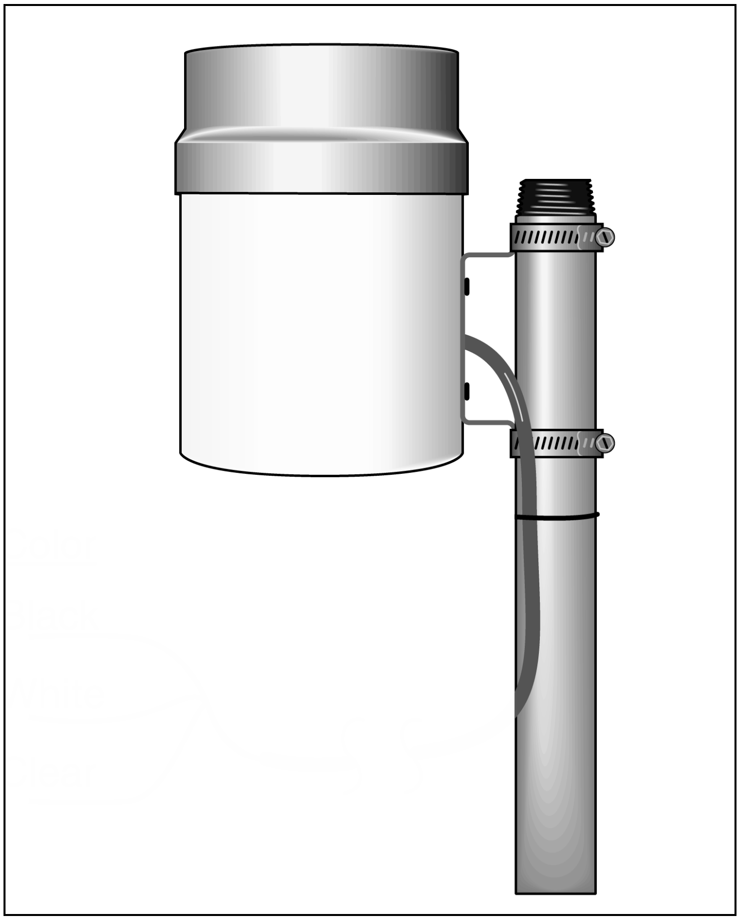


**Figure 2.14:** Installation of Campbell Scientific TE 525 Precipitation Gauge.

### 2.4.5 Installing Barometric Pressure Sensor

The barometric pressure instrument was mounted inside the enclosure to avoid any corrosion, as a condensing environment was not favorable for the instrument. The sensor was mounted on a perforated plate, which is held by the enclosure. For easy access and routing of other sensor wires the instrument was installed either at the top right or bottom right corner. A typical layout of the datalogger and pressure gauge inside the enclosure is shown in Figure 2.15.

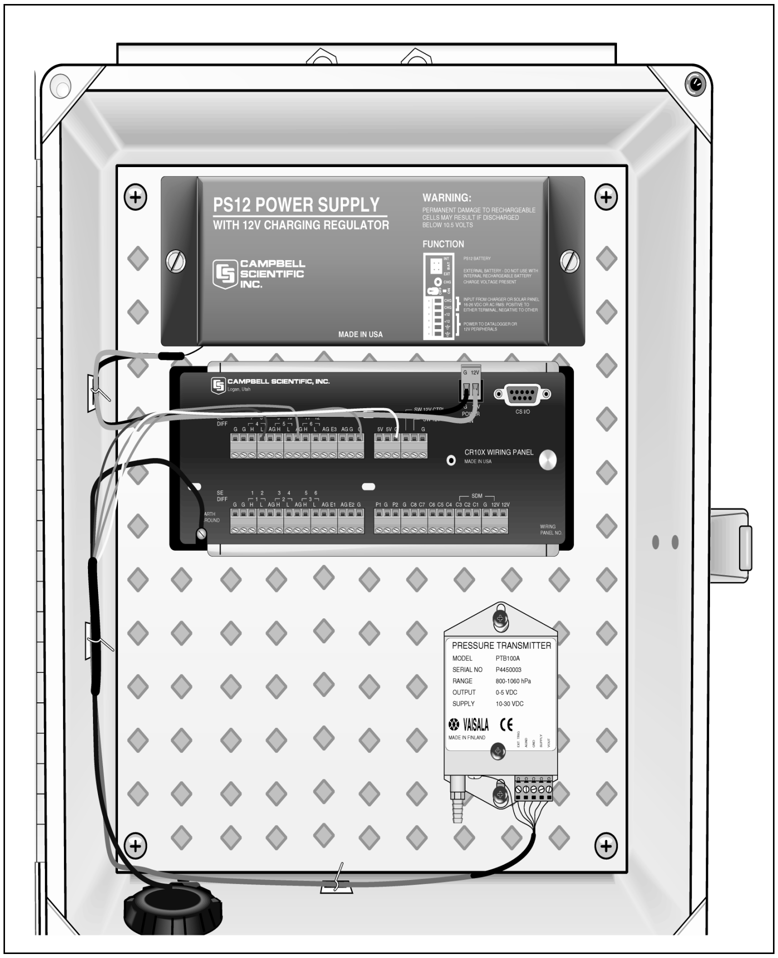



**Figure 2.15:** Installation of Campbell Scientific CS 105 Pressure gauge and typical layout of datalogger and Pressure gauge inside enclosure.

### 2.4.6 Installing Soil Temperature Sensor

According to WMO Standards [44], soil sensors have to be measured at 5 levels: 0.05m, 0.1m, 0.2m, 0.5m and 1.0m. Due to inadequate availability of analog channels on the datalogger, only a single measurement at 0.1m (approximately 4”) has been implemented. To install soil temperature, a hole of diameter 0.3” was created using an iron rod to a depth of 0.1m (approximately 4”). The sensor was installed in this hole such that tip of the sensor is in contact with the undisturbed soil at the end of the hole. The lead wire was buried under soil to avoid any heating effects and also damage to the wire due to weathering and grass mowing. The ground was leveled after installing the instrument to avoid water logging. We plan to modify the installation by placing the sensor horizontally in undisturbed soil as in Figure 2.16 so that multiple measurements can be made at different depths in future.

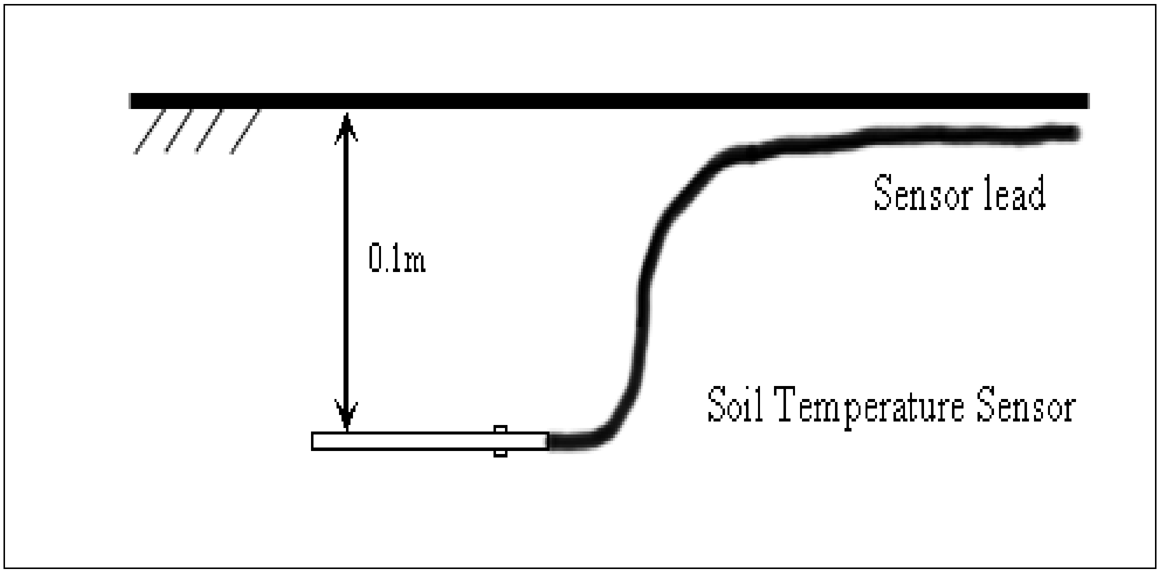



**Figure 2.16:** Installation of Campbell Scientific CS 107 Soil Temperature instrument

### 2.4.7 Installing Soil Moisture Sensor

Soil moisture sensor was installed in close proximity with the soil temperature sensor CS 107. The sensor was installed at an angle such that the tip of the sensor was at a depth of 0.1m (approximately 4”) as shown in Figure 2.17. By placing the sensor at an angle, the average volume water content was measured rather than an area measurement. Soil moisture was installed using the soil moisture installation toolkit, which included a CS615G probe insertion guide tool and CS615P pilot tool. Using the insertion guide tool and pilot tool, guides for the two leads of the soil moisture probe were made. The guides prevented the leads of the soil moisture from bending during installation. After installation the ground was leveled to avoid any water logging.

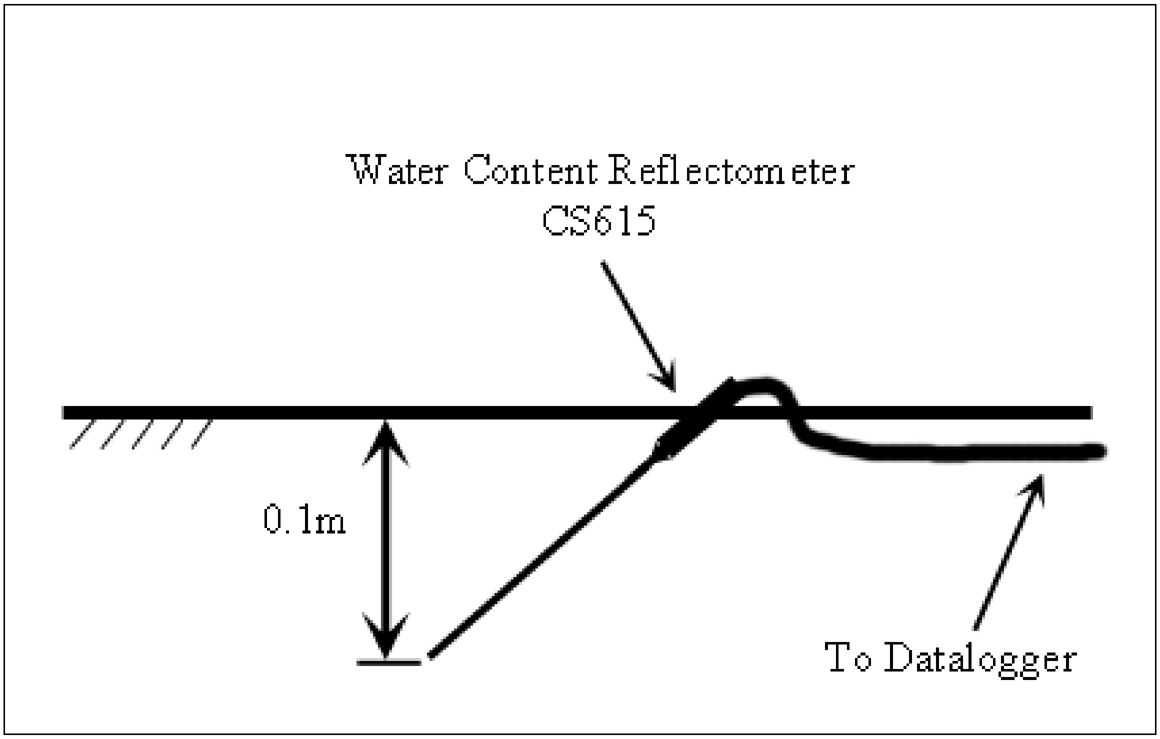



**Figure 2.17 :** Installation of Campbell Scientific CS 615 Soil Moisture Instrument.

### 2.4.8 Installing Solar Panels

A 10W/20W Solar panel along with a battery and voltage regulator were installed to supply DC Power for the datalogger and its accessories. High power solar panels (20W) were installed in mountain areas to accommodate for lower radiation and power supply during long winters. Lower power solar panels (10W) were installed in others regions. The solar panels were installed on the tower as shown in Figure 2.18 facing south to maximize the output of the panel. The output voltage of the solar panel varies from 9V to 35V depending upon the incident solar radiation. The voltage regulator provides a regulated voltage of +12V, which is connected to the datalogger and battery. During nighttime, when solar power is off, the battery provides the DC power required by the datalogger. To maximize the space and to ease routing of the wires, the regulator and battery were mounted in the lower mid-section of the enclosure.

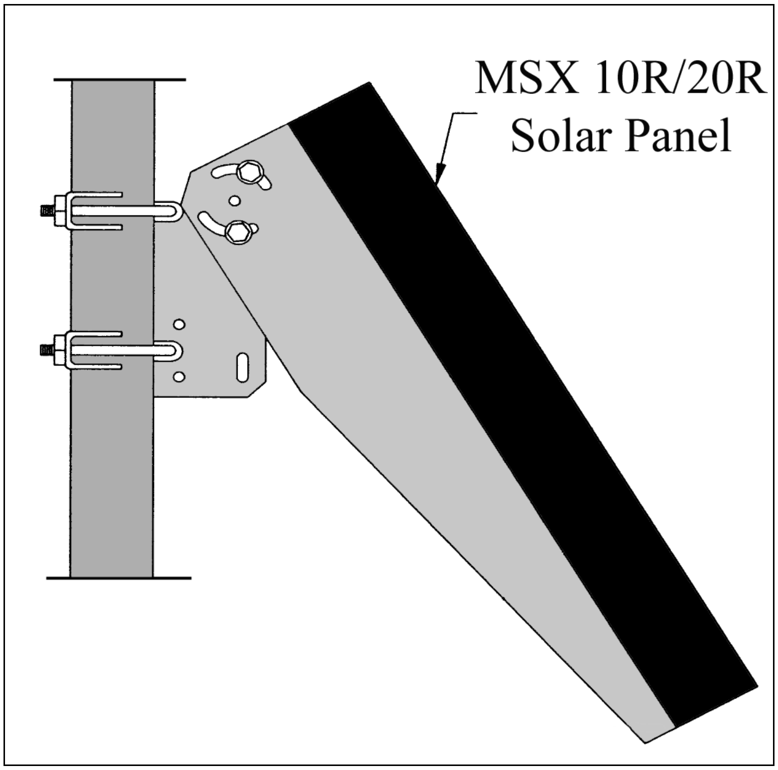



**Figure 2.18:** Installation of Campbell Scientific MSX 10R/20R Solar Panel.

### 2.4.9 Installing Datalogger

The datalogger was installed inside the enclosure to avoid corrosion to the contacts and wiring panel. To facilitate easy routing of the sensor wire and connections to accessories, and access to other devices, the datalogger was mounted on the top left corner. The custom communication port known as CSIO is used to connect communication devices such as phone modem, RF modem to the datalogger. The power to the datalogger is provided through the voltage regulator output. The wiring diagram of the sensor connections is as shown in Figure 2.19.

The overall connection diagram of the AgNet weather station using phone-based communication is shown in Figure 2.20.

## 2.5 Establishing communication

The data collected at the tower is stored in the datalogger's memory. Since the datalogger acquires data continuously, the data collected will be overwritten if it is not copied to another location outside the datalogger such as a PC Host. These data have to be transferred to a computer where they are archived and further processed. The data is transferred using either Campbell Scientific COM200 [7] or DL 112 phone modem on the datalogger side. A regular PC modem was used on the PC host side to establish communication. The modems on either side act as interfaces between the PC host and datalogger. The data collection is performed using the Campbell Scientific PC-208W software suite. The software is a collection of utilities that support data collection and archiving. The software has features to collect from multiple sites sequentially and also can be automated to download data at regular intervals.

The host computer establishes contact with the remote weather station by communicating through telephone lines daily at 5:00AM. Once the connection is established the host computer downloads only the latest data and terminates the connection. The collected data is formatted into comma delimited values and is appended to the archived file in the hard disk of the host computer. The procedure is repeated for all AgNet stations and data are appended to each unique station file. The overall block diagram of the phone-based communication system used by AgNet is shown in Figure 2.21. The program also has the feature to reprogram the datalogger, update the time on the datalogger and change the sampling frequency of the datalogger.

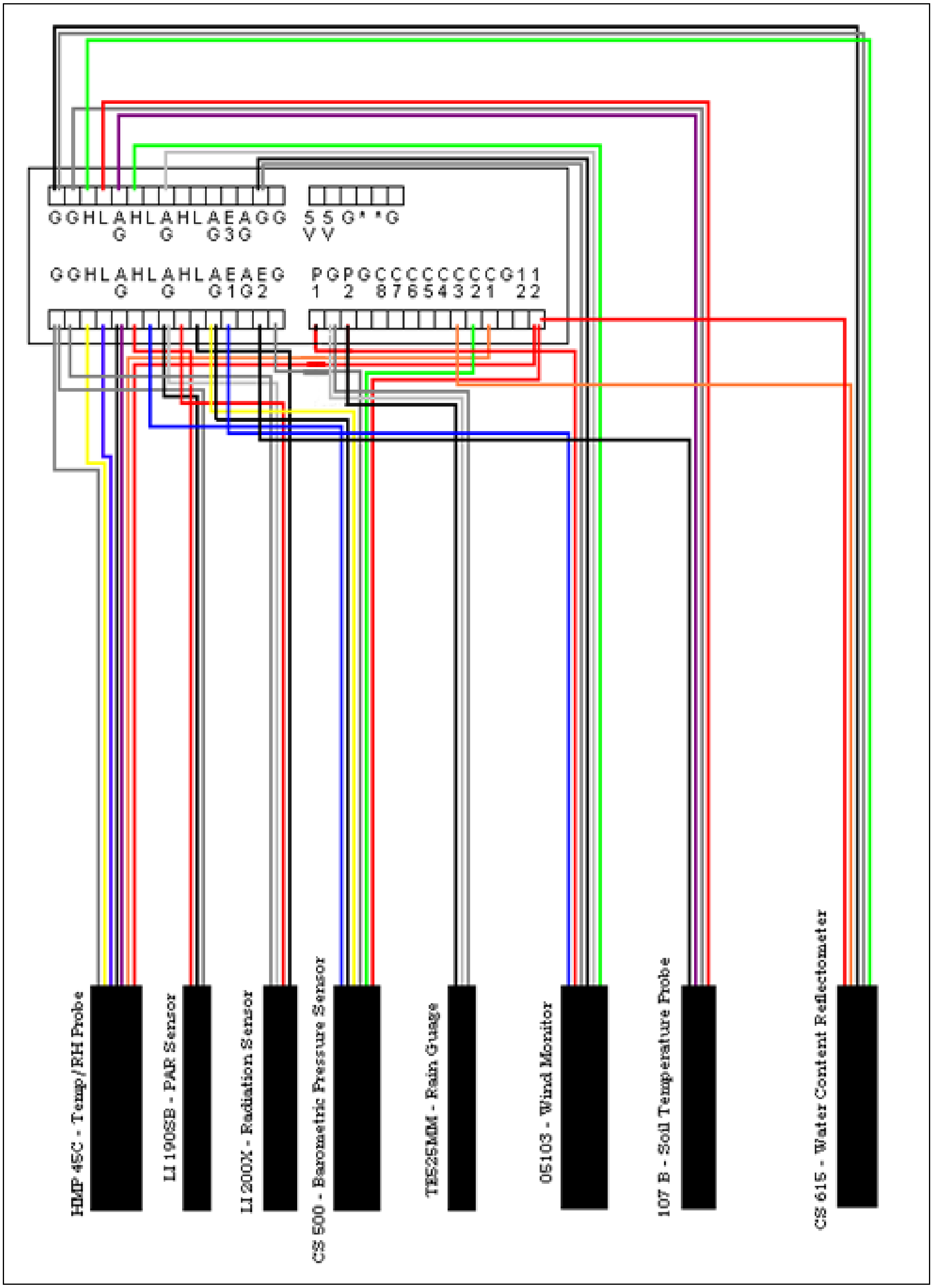



**Figure 2.19:** Wiring diagram of the AgNet Station.

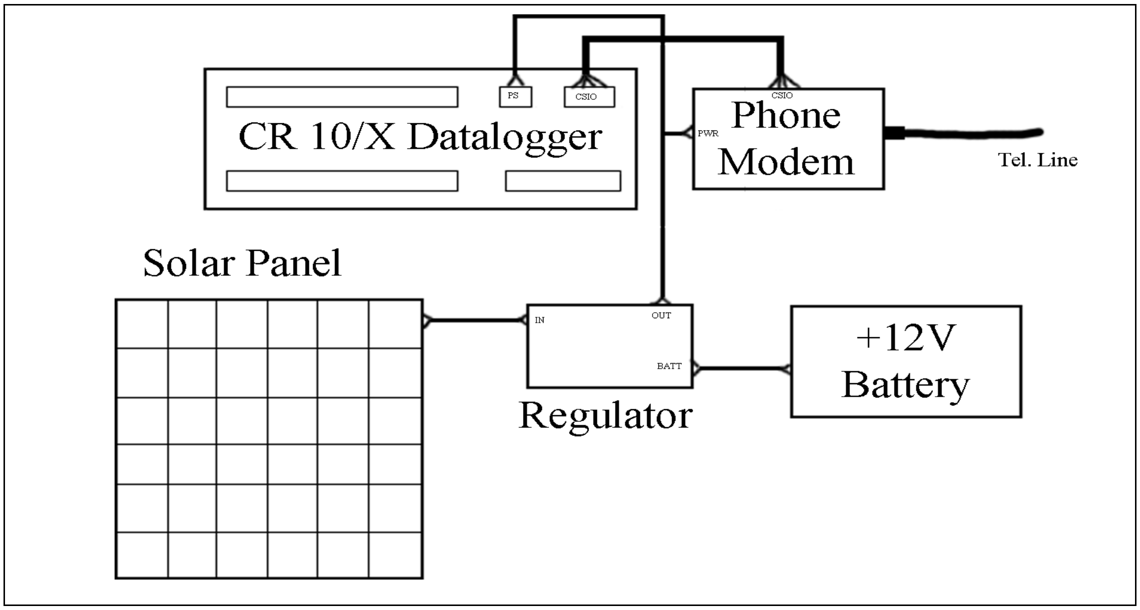



**Figure 2.20 :** Overall wiring diagram of the AgNet Station.

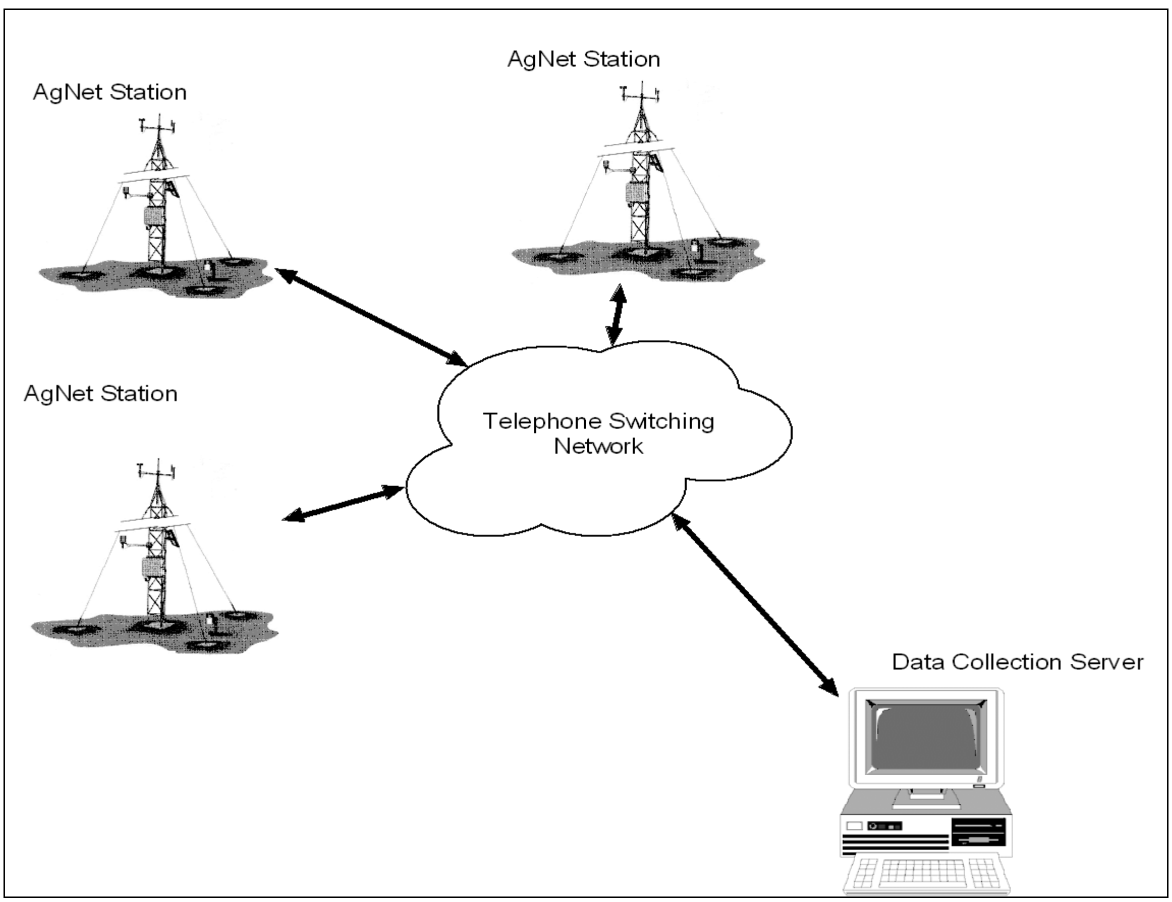



**Figure 2.21:** Overall Block diagram of phone based communication.

## 2.6 Data Collection

The data are collected by the datalogger by performing an analog-to-digital conversion of the analog signal generated by the sensors. This process of sampling is repeated at a regular interval to get a time history of the parameter. The sampling frequency is, however, based on the Nyquist sampling frequency. Hence in order to measure a variation of a parameter in the atmosphere, one should sample atleast at twice the frequency of the variation.

As recommend by EPA [16], estimates of the mean should be based on at least 60 samples to obtain good accuracy. Table 2-12 and Table 2-13 shows the various sampling and averaging frequencies of the different parameters at present for hourly and daily values.

**Table 2-12:** Sampling and averaging intervals of Hourly data.

| Parameter | Sampling Interval | Averaging Interval |
|---|---|---|
| Air Temperature | 1 hour | - |
| Relative Humidity | 1 hour | - |
| Wind Speed | 1 min | 60 min |
| Wind Direction | 1 min | 60 min |
| Standard Deviation for Wind Direction | 1 min | 60 samples |
| Soil Moisture | 1 hour | - |
| Soil Temperature | 1 hour | - |
| Precipitation | 1 min | Cumulated over an hour |
| Pressure | 1 hour | - |
| Solar Radiation | 1 min | 60 min |
| Photosynthetically Active Radiation | 1 min | 60 min |

**Table 2-13:** Sampling and averaging intervals of Daily data

| Parameter | Sampling Interval | Averaging Interval | Maximum / Minimum Sampling Interval |
|---|---|---|---|
| Air Temperature | 1 min | 1440 min | 1 min |
| Relative Humidity | 1 min | 1440 min | 1 min |
| Wind Speed | 1 min | 1440 min | 1 min |
| Wind Direction | 1 min | 1440 min | - |
| Soil Moisture | 1 min | 1440 min | 1 min |
| Soil Temperature | 1 min | 1440 min | 1 min |
| Precipitation | 1 min | Summed over 1440 min | - |
| Pressure | 1 min | 1440 min | 1 min |
| Solar Radiation | 1 min | 1440 min | 1 min |
| Photosynthetically Active Radiation | 1 min | 1440 min | 1 min |

## 2.7 Data Dissemination

One goal is to disseminate data in an efficient and cost-effective way. In order to do so, web based data dissemination was found to be ideal. In this technique the data collected at the data collection server had to be transferred to a web server. The data has to be converted into Hyper Text Markup Language (HTML) pages, which are then placed on web page publishing folders. The data was disseminated in a hierarchical format, classification on the basis of year, month and day.

The data collected at the data collection server is downloaded to the web server at 6:00 AM. The data are then processed using in-house software known as “CONVERT”. The basic function of the convert program is to read the data and create HTML pages, based on the configuration for each site. The software also can be configured to user needs based on command line switches or configuration declaration.

### 2.7.1 Convert Program

The convert program is written in ‘C’ language on a Sun machine for creation of HTML pages. The software was designed to operate in two modes, namely

- Automated mode; and,
- Manual mode

In Automated mode, the program executes based on the command line switches and does not require any input from the user during runtime. Based on the command line switches, configuration files and other support files, the program could convert the data into HTML pages. However, the manual mode was designed to be operate with a user interface. In manual mode, the software was conceived to have dynamic “on the fly” declaration of data file, data format, QC parameters, etc. A variety of command line switches can be declared to make the program very flexible to meet the user needs.

The “CONVERT” program when used for publishing daily and hourly data creates five different kinds of pages. The hierarchies of the HTML pages created are as shown in Figure 2.22. The different pages can be listed as

- Main page;
- Yearly page;
- Monthly page;
- Hourly summary page; and,
- Daily summary page.

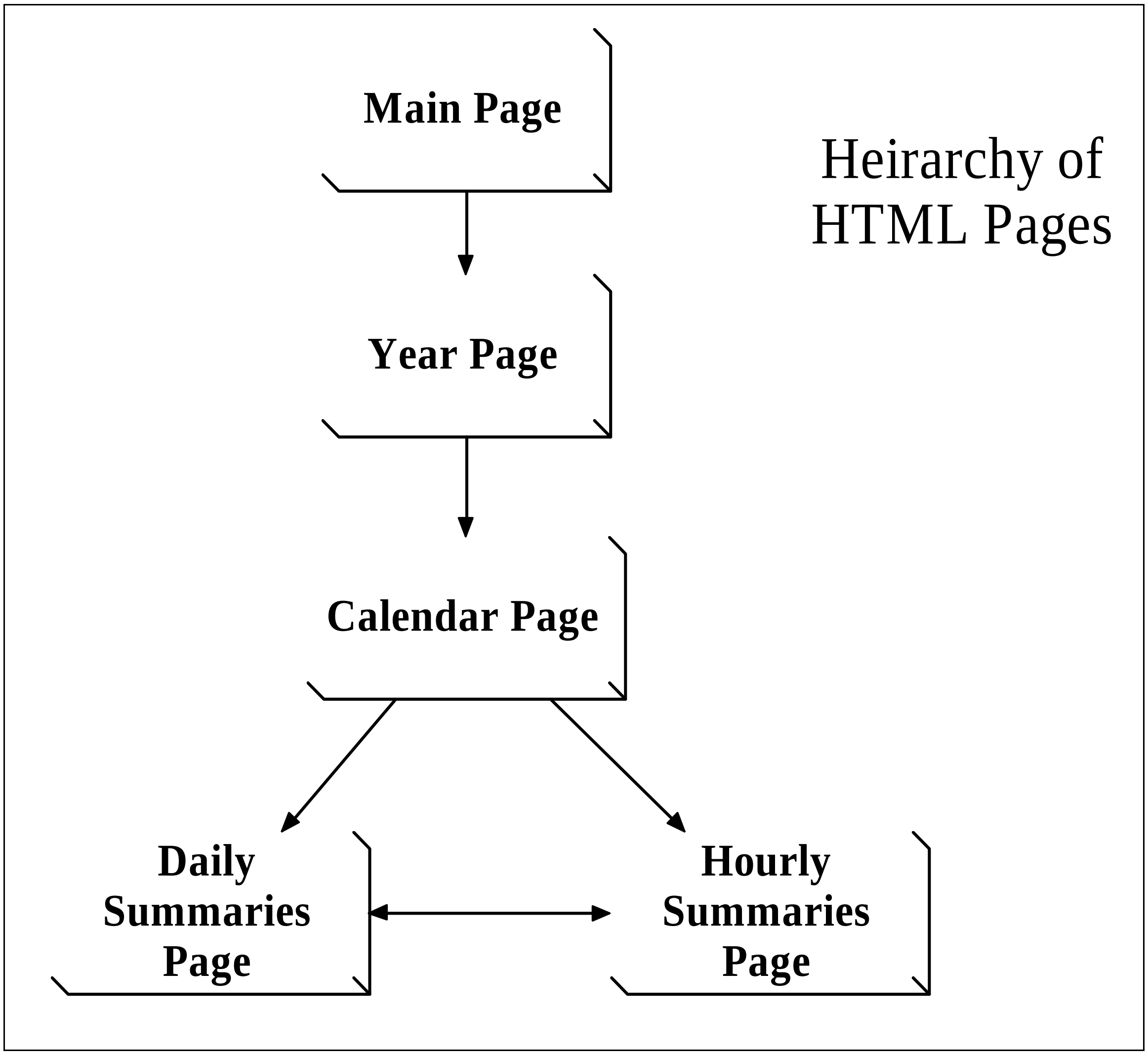



**Figure 2.22** : Heirarchy of the generated HTML pages in auto mode

Main page forms the first page in the hierarchy with links to yearly data. Main page contains the information about the station like the station name, station address, NOAA ID, station latitude, station longitude, station elevation, station map and details of instrumentation on the tower and the site characteristics. The yearly page forms the second page in the hierarchy with links to the calendar page. Based on either existing calendar pages or available data, the program enables the link to the calendar page. The calendar page forms the third page in the hierarchy with links to daily and hourly summaries page. The calendar page is laid in the form of a calendar with links to the daily summaries of the previous, current and next month. The links on the monthly page are enabled on the basis of either data or existing files. The daily summary page forms the fourth in the hierarchy with the daily summary data laid out in tables format. A variety of parameters are shown in this page based on the configuration declaration. This page has links to the hourly summaries for every day in the month. The hourly summary page also forms the fourth in the hierarchy with hourly summary data laid out in the form of a table. The page shows a variety of parameters that are based on the configuration declaration.

The flow diagram of the “CONVERT” program is as shown in Figure 2.23. The different stages in the flow can be listed as:

- Initialize variables;
- Load command line switches;
- Read station configuration;
- Create a table of existing files;
- Data file scanning – Checks for missing data, repeating, undeclared format;
- Creating the Main page;
- Creating the Yearly page;
- Creating the Monthly page;
- Creating the Daily summaries page;
- Creating the Hourly summaries page; and,
- Release allocated variables.

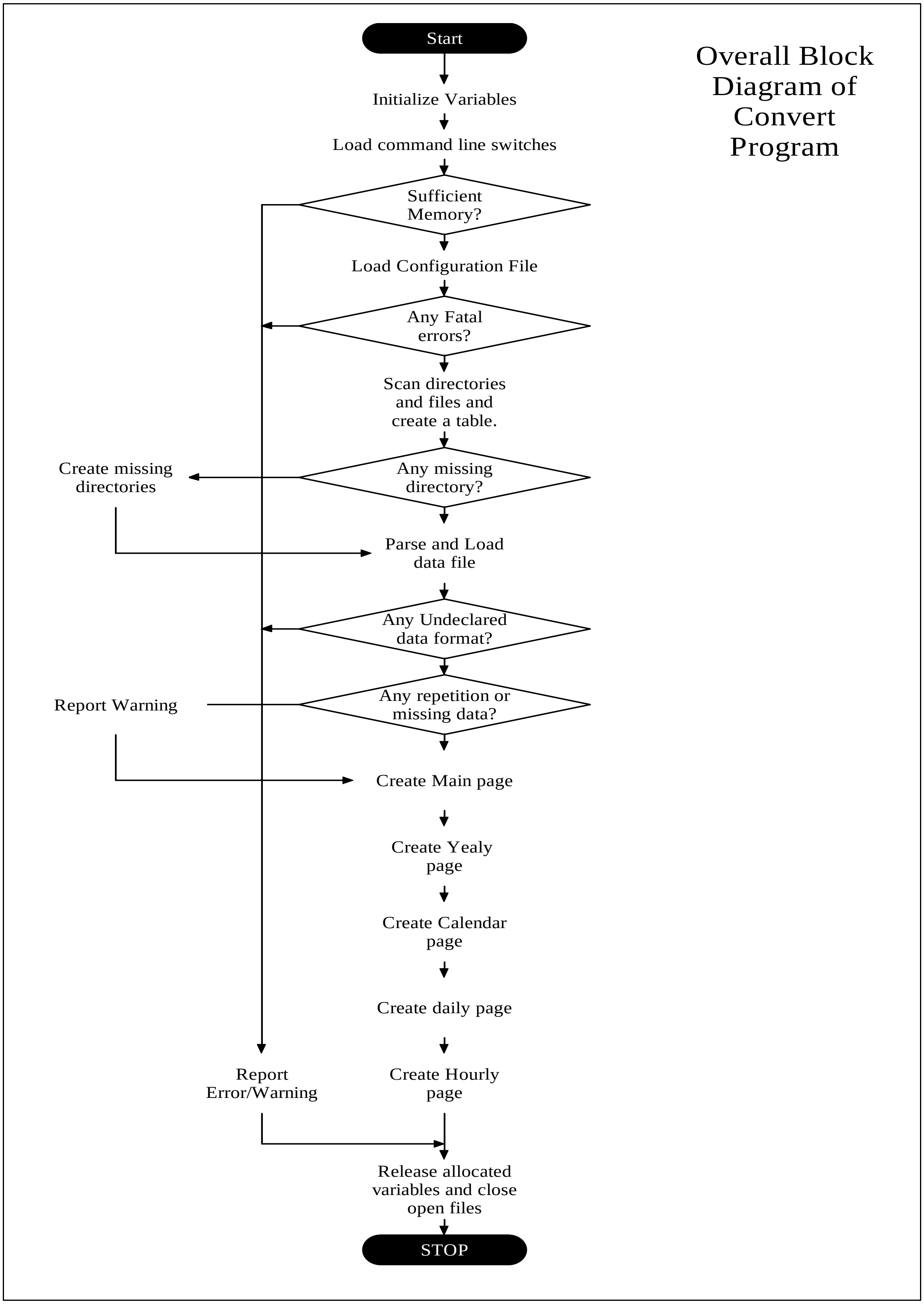



**Figure 2.23:** Overall block diagram of the Convert program in AUTO mode.

In the “Initialize Variable” section, the program allocates memory for the huge static arrays required for storing the data. Arrays are organized on the basis of the data class. The different classes of data are

- Daily summary data;
- Hourly summary data;
- Half hourly summary data;
- Quarter hourly summary data (15 min); and,
- 5 minute summary data

The program currently can handle daily summary data and hourly summary data. Once the memory is allocated, it is initialized to zero. Also another array is allocated which holds information about the file list.

Once the arrays are initialized, command line variables are read into the program. There are a variety of command line options that can be declared. List of the few command line switches is shown in Table 2-14

**Table 2-14 :** List of command line switches and its description

| Command line switch | Description |
|---|---|
| config | Location of the configuration file.<br>e.g. –config /export/home/www/agnet/cast_twr/conf/cast_twr.config |
| data | Location of the data file to be used. Overides configuration file<br>e.g. –data /export/home/www/agnet/data/cast1999.dat |
| debug | Debugging options, options include files,missing,qc,config or all |
| overwrite | File creation mode. This option overwrites the existing files. Options include mainpage, yearly, calendar, daily, monthly or all |
| mainpage | Create the Main page (Station Information page) |
| qc_report | Generate a report of QC Failures for the data |
| no_qc | Do not perform QC procedures |
| format | Format declaration – parameters are declared with comma seperation |

| | |
|---|---|
| | e.g. –format AirTempC,SoilTempC,WindSpd,WindDir |
| type | Data format |
| delimit | Delimiters used for the data<br>e.g. –delimit colon (For ':' delimiter) |
| allmonths | Create HTML pages for all months in the year |
| year | Create HTML pages for the year |
| log | Create log file for errors and warnings generated during execution |

Once the command line switches are loaded, the program loads configuration specific to the station through the "-config" switch. A variety of configuration declarations can be made in the configuration file. Please refer to Appendix A2 for the listing of a typical configuration file.

After loading the configuration file the program is now ready to begin reading the data file. During the process of loading, the program checks for any undeclared data formats or for any repetition in the data. Undeclared data formats are those data points in the data file that have an array ID that is not declared in the configuration file using the configuration declaration "HOURLY_DATA_FORMAT" or "DAILY_DATA_ FORMAT". The "CONVERT" program creates a table based on year, Julian day and the number of data points in the data file. Any discontinuity in the table or the number of data points for the interval less than expected is an indication of missing data. Any field in the table overwritten or number of data points for the interval greater than expected, is an indication of repetition in data. Depending on either command line switch "-debug" or configuration declarations "LOG_DEBUG", the warning during loading are reported on the console. Once the data is loaded, the program checks for any missing data. A log is generated and appended to the log file.

Once data are checked for errors, the program does a QC check if the check is enabled through either command line switches or a configuration declaration. QC procedures are described in detail in Chapter 4.

The data are now ready to be published over the web. The program first generates the main page by copying the header section and during the process all embedded variables that are present in the header are evaluated. The program creates links to the yearly pages if data are present or files are present for the year. The footer section is copied from another file and during the process any embedded functions are evaluated. The block diagram of the Main Page module is shown in Figure 2.24.

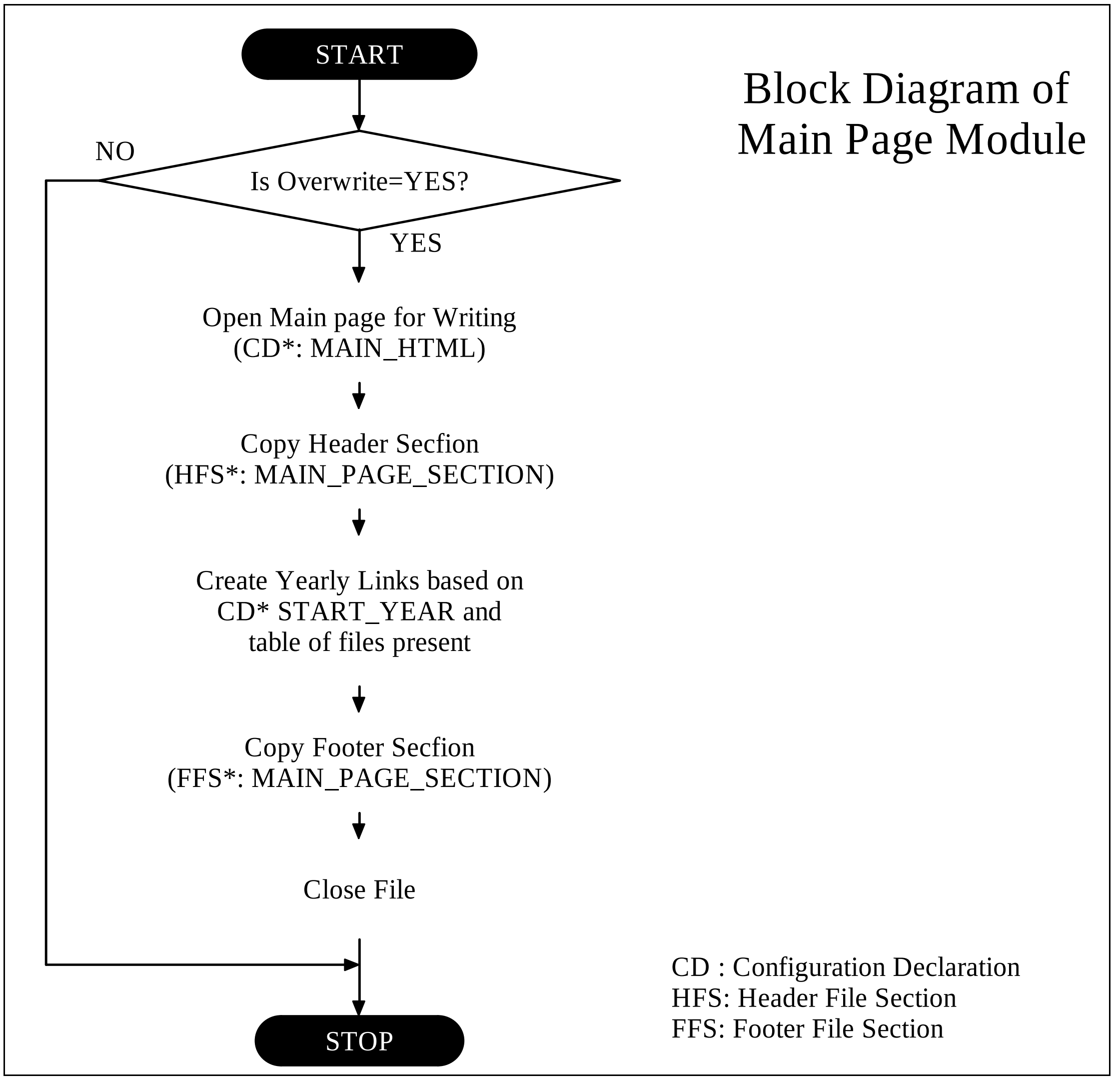



**Figure 2.24:** Block diagram of the Main page module.

The yearly page is created and the header information is first copied into the file. All the embedded functions are evaluated before writing the file. Once the header is written, links to the calendar page are either created or enabled based on the data and existing files. The footer section is copied from another file and all embedded functions are evaluated during the process. The block diagram of the Year page module is as shown in Figure 2.25.

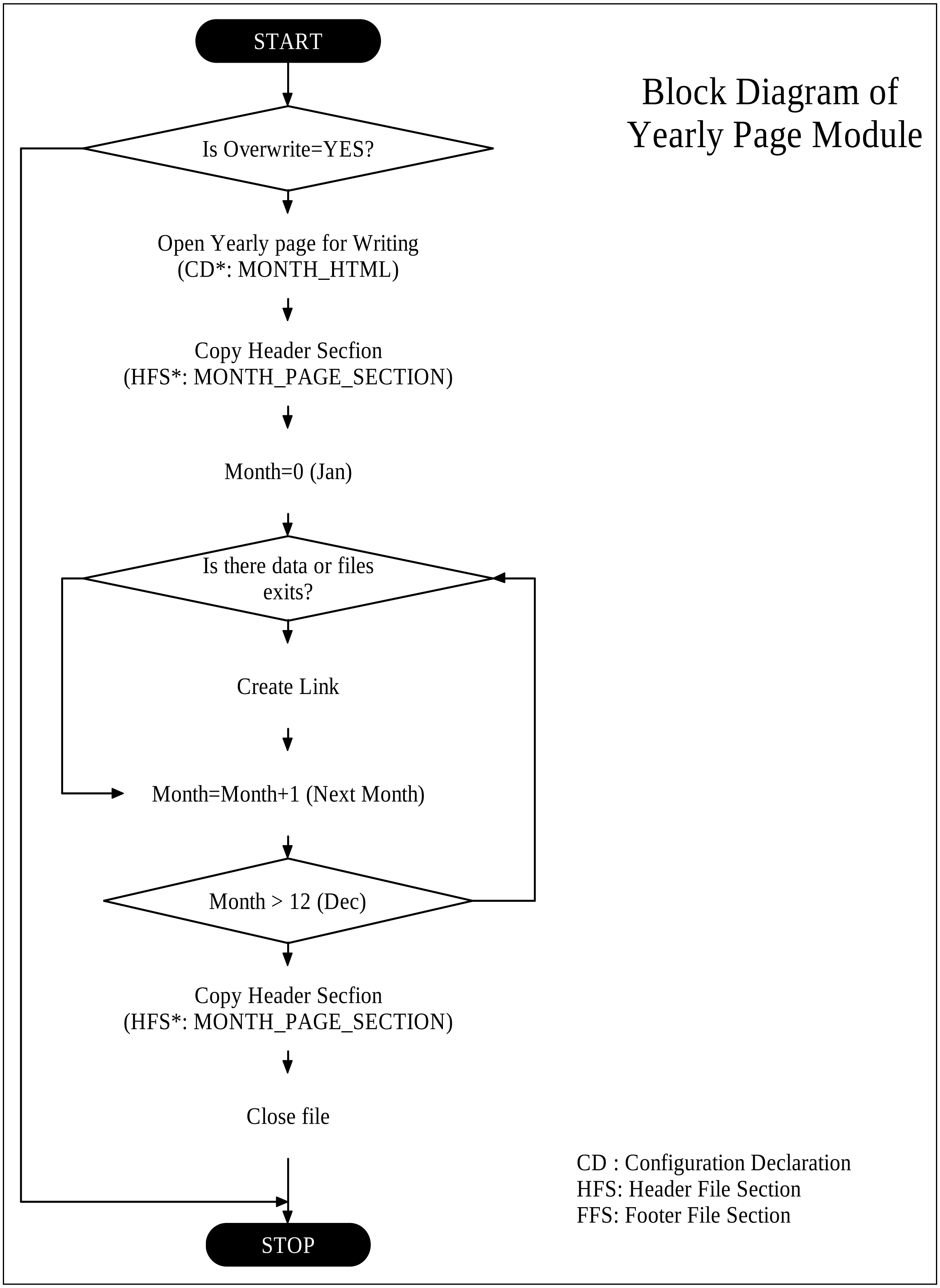



**Figure 2.25:** Block diagram of the Yearly Page module

The calendar page is created and links to the daily and hourly summaries are created on the basis of data or existing files. The block diagram of the Calendar page module is as shown in Figure 2.26.

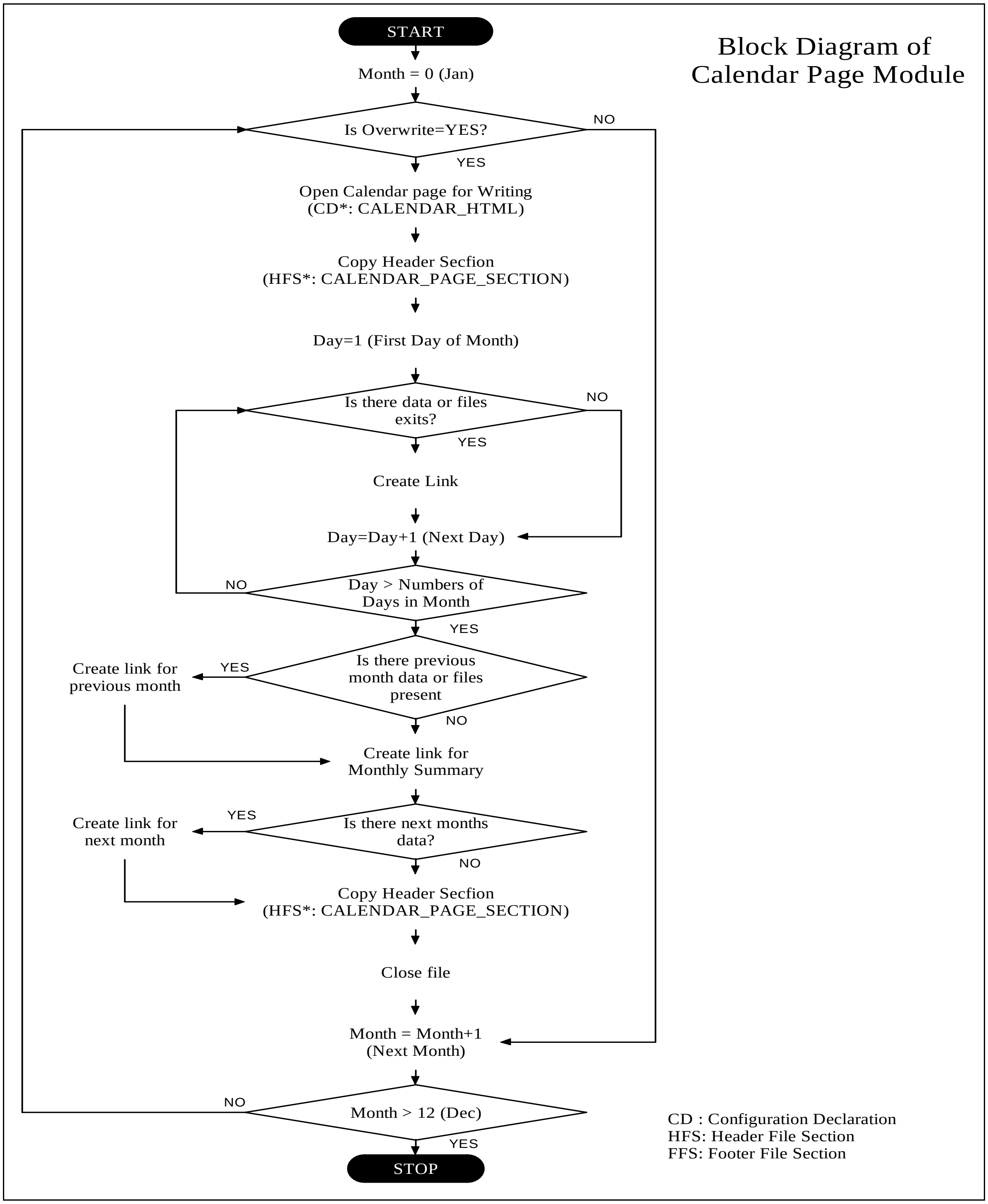



**Figure 2.26:** Block diagram of the Calendar Page module.

The daily summary page is created on the basis of data. Initially the header is copied into the file and HTML tags for the daily data is generated on the basis of output format declared in the configuration file. Finally footer information is copied into the page with all embedded variables evaluated. The block diagram of the Daily page module is as shown in Figure 2.27.

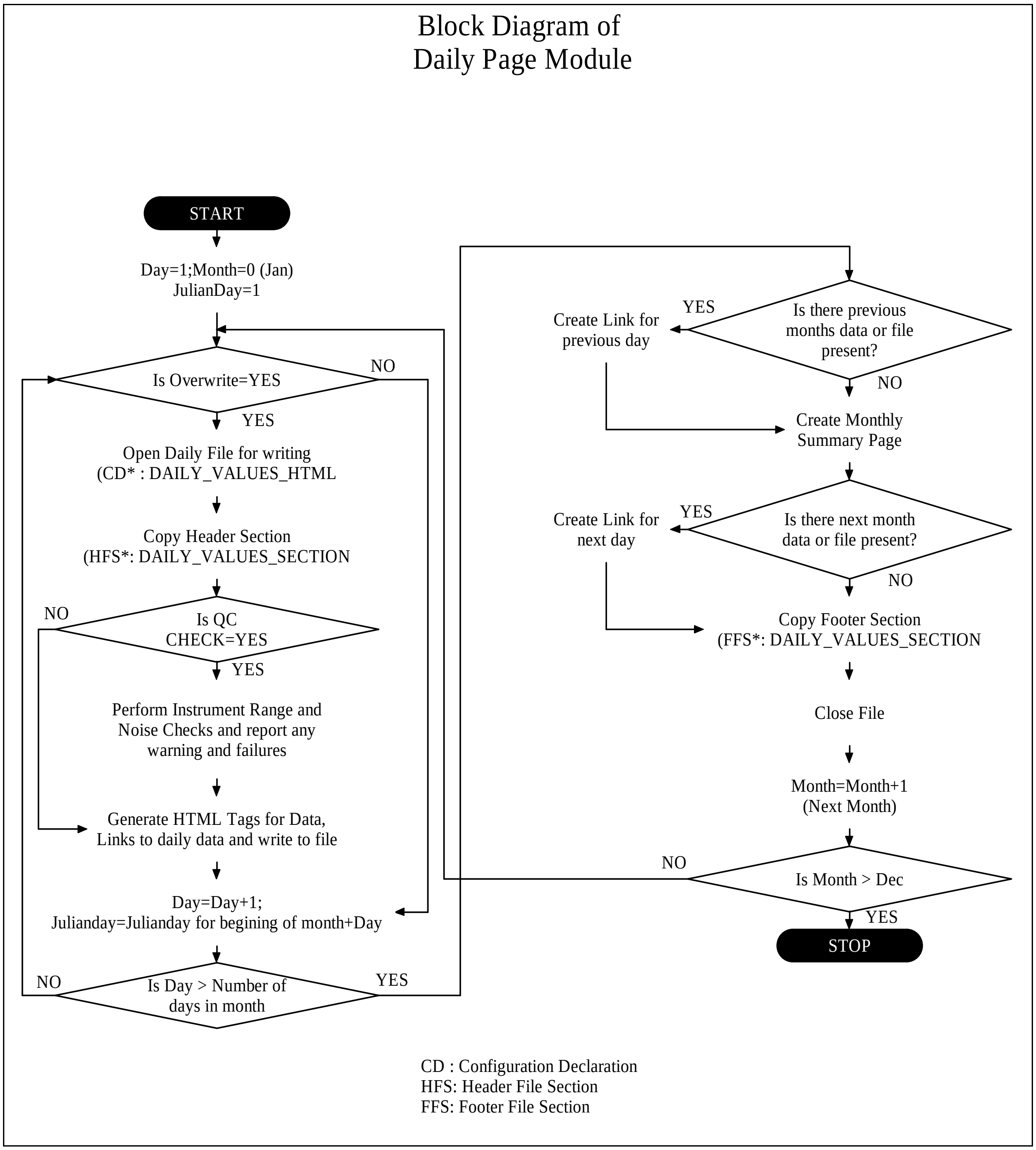



**Figure 2.27:** Block diagram of the daily page module

The hourly summary page is created on the basis of data. The header is copied into the file and HTML tags for the hourly data is generated on the basis of output format declared in the configuration file. The footer information is copied into the page with all embedded variables evaluated. The block diagram of the Hourly page module is as shown in Figure 2.28.

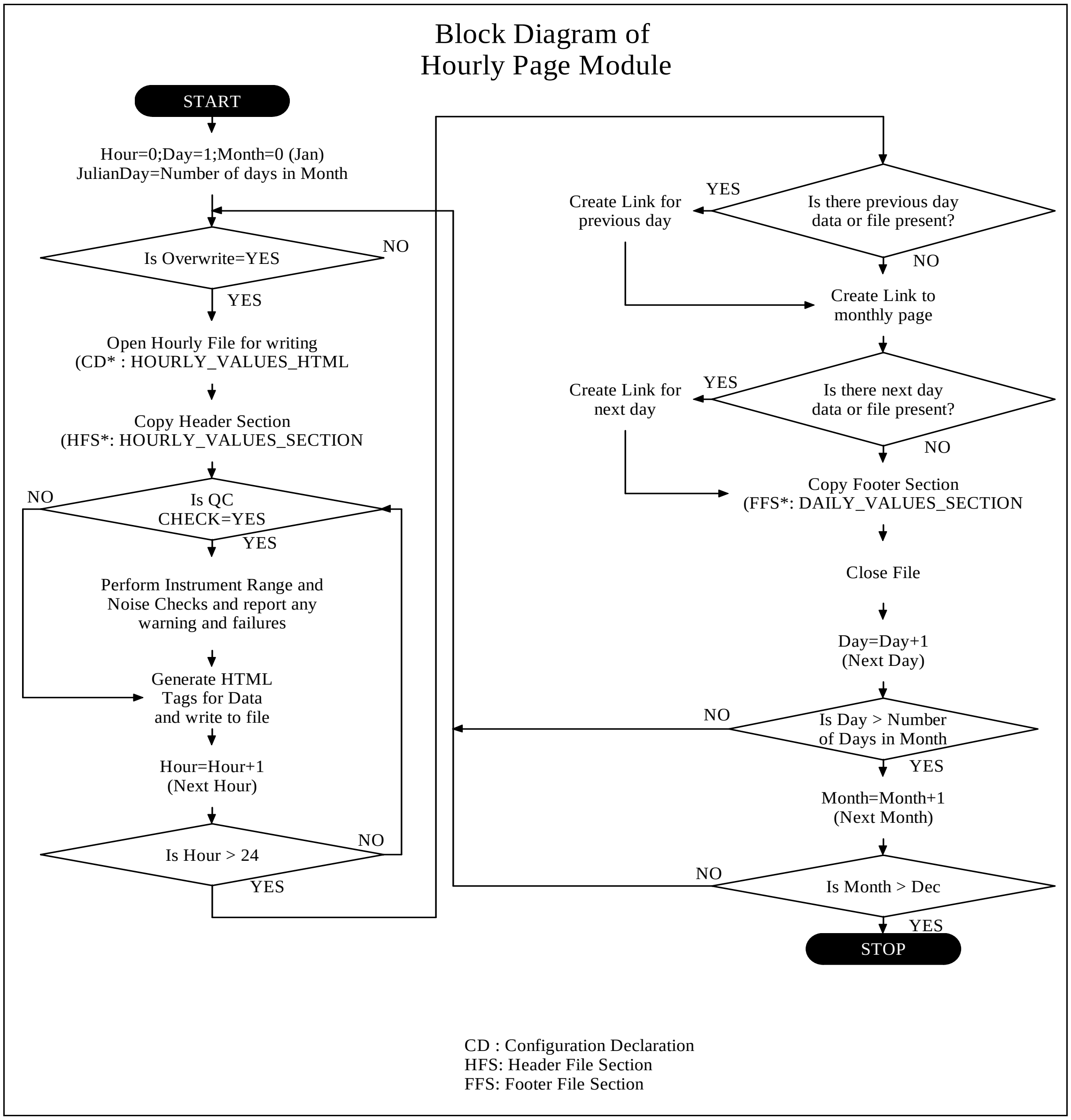



**Figure 2.28:** Block diagram of the Hourly page module.

Sample web pages generated by the “Convert” program can be found in Appendix A3.

# CHAPTER 3

# ENHANCEMENT IN COMMUNICATION AND DISSEMINATION

Availability of real time data has tremendous applications and use such as Crop and Pest management and in severe weather. Since this kind of communication involves maintenance and constant usage of the communication media throughout the year, we had to come up with an efficient, reliable and cost-effective mode of communication and dissemination. In this chapter we will discuss different techniques that were considered to achieve this goal.

## 3.1 Comparison of different communication techniques

In order achieve real-time data communication we started with gathering information about all the available options for communication. We evaluated several options on the basis of the advantages and disadvantages in the mode of communication. Since all our sites are remote and are operating on solar panels and batteries, first concern is power. Communication mode that is chosen should consume the least power. Also, because the data are to be transferred in real-time, the station must be connected regularly via the communication media. Since there is a charge for using the communication media, there is a recurring expense to have continuous data in real-time. With these two important factors in mind we evaluated the following communication modes:

- Phone based communication;
- Cellular phone based communication;
- Satellite based communication;
- Short haul modem based communication; and,
- RF based communication.

Table 3-1 lists the advantages and disadvantages of these different kinds of communications.

**Table 3-1:** Comparison of different communication techniques

| **Communication Technique** | **Advantages** | **Disadvantages** |
|---|---|---|
| Phone | Simple installation<br>Low initial cost<br>Low power consumption | Telephone line required<br>High recurring cost as each call is expensive<br>Communication depends upon condition of the telephone line |
| Cellular phone | Simple installation<br>Mobile<br>Low initial cost | Requires Cell service<br>High recurring cost as each call is expensive<br>Power consumption is high |
| Satellite | Reliable<br>Remote | One-way communication<br>High Initial Cost<br>Power consumption is high |
| RF (Radio Frequency) | Remote<br>Reliable<br>Low recurring cost<br>Power consumption is reasonable | Pre-allocated frequencies<br>Limited by distance<br>High Initial Cost |
| Short Haul Modem | Low initial cost<br>Low power consumption | Complex Installation<br>Limited by distance<br>Not Reliable |

Phone-based communication was easy to implement and was also the lowest power-consuming device, but was not well suited as long distance calling was too expensive for real-time data transfer. Also the local telephone company had to install the telephone line before the data could be transferred. In case of a bad line during emergency conditions, repairing the line was not a priority for the phone company.

Cellular phone-based communication was also easy to implement. It avoided the problem of waiting for the phone company to install a telephone line. For very remote site cellular phone-based communication has the advantage over phone-based communication as long as they are in the vicinity of a cellular phone base station. But disadvantages include expensive cellular service for real-time data transfer, high initial cost and high power consumption.

Satellite-based communication was ideal for very remote sites. Since it had no connection with any ground equipment for data transfer, it can be used for any remote site. Disadvantages include one-way communication, a separate dish had to be mounted to enable satellite communication, high initial cost and power consumption. One way communication was one of the major disadvantages as it limited datalogger reprogramming based on existing conditions.

Short-haul communication was based on connecting the datalogger to a computer/terminal through short-haul modems that were connected by a twisted wire. This kind of communication is limited by distance. Also laying a cable to connect the modems over fields is not a recommended practice.

Radio Frequency (RF)-based communications is based on a RF link from the datalogger to the computer/terminal. Since the communication media is pre-allocated by the Federal Communication Commission (FCC) for use, there are no charges for using this communication media. Hence, it is least expensive for real-time data transfer. The power consumption of both the RF Radio and RF modem was within reasonable limits. Disadvantages include the need for pre-allocated frequencies, finite distance as RF is attenuated by the atmosphere and the high initial cost.

## 3.2 RF Telemetry

From the comparison above, RF based communication was very promising for our needs. Hence we decided to try this mode of communication between one of the AgNet sites and our facility at Varsity Lab. We chose the Lake Wheeler site for the test of this mode of communication as it was easily accessible and the site was within the range of the RF link. Varsity Lab facility at NC State University was used as a base station for the experiment.

Since a pre-allocated frequency had to be used with RF telemetry, we applied for frequency allocation through Federal Communication Commission (FCC) and Association of Public-safety Communications Officials (APCO). We were allocated five different frequencies, which are

- 460.08750 MHz;
- 460.26250 MHz;
- 460.33750 MHz;
- 460.48750 MHz; and,
- 460.51250 MHz.

There were certain restrictions put forth by FCC on using these frequencies. Frequencies such as 460.33750 MHz could not be used within 110km radius of the South Carolina state border and 460.48750 MHz could not be used within 110km radius of the Georgia state border.

It was decided to install a set of instruments and a separate datalogger for this experiment. The power for the datalogger was derived using a DC adapter from 110V AC mains supply. The tower was instrumented with air temperature, relative humidity, soil temperature, wind speed, wind direction, barometric pressure and solar radiation. A separate enclosure was installed on which we mounted the Datalogger, RF Modem and RF Radio. A directional Yagi antenna (manufactured by Celwave Incorporated) was mounted on the top of the tower to have better reliability in communication. Since the base station can access 254 different remote stations, each station is uniquely identified on the network by a station ID. The jumper switches of the RF Modem can be used to set the station ID of a remote station. The station ID of the RF Modem at the base station is factory set to 255. For this experiment we set the station ID of the remote station at Lake Wheeler Field laboratory as 1.

At the base station in Varsity Lab, we installed an omni directional antenna manufactured by Celwave Incorporated. The antenna was installed on the roof of the Varsity lab at an approximate height of 30 feet. A base station consisting of a RF Modem and a RF radio was used to establish communication. The antenna was connected to the base station using a RG 8A/U cable. The base station was connected to a computer through the serial port RS232.

The block diagram of the RF Telemetry is shown in Figure 3.1. Data collected at the remote site is transmitted through the RF Modem to the RF Radio. The base station computer requests the RF Modem to link to the datalogger by specifying the station ID. The RF Modem communicates with the RF Radio, which modulates the data with a carrier and is transmitted by the antenna. The antenna at the remote station converts the radio waves into electrical signals and is fed to the RF Radio. The RF Radio demodulates and decodes the data. Based on the station ID selected, the RF Modem acknowledges the communication and also establishes a communicating link between the base station computer and the datalogger. Once the link is established, the base station issues the command for downloading data and once the downloading is finished the communication is terminated.

The different components of the RF based communication are, namely

- RF Modem;
- RF Radio;
- RF Antenna;
- Datalogger; and,
- Base station computer

Brief descriptions of each of these components are as below.

### 3.2.1 RF Modem

The RF modem used for this experiment was RF 95A, manufactured by Campbell Scientific Incorporated and is the interface between the datalogger at the field station and the radio. The RF Modem is connected to the datalogger through the serial IO port and the RF Radio through the transceiver IO port.

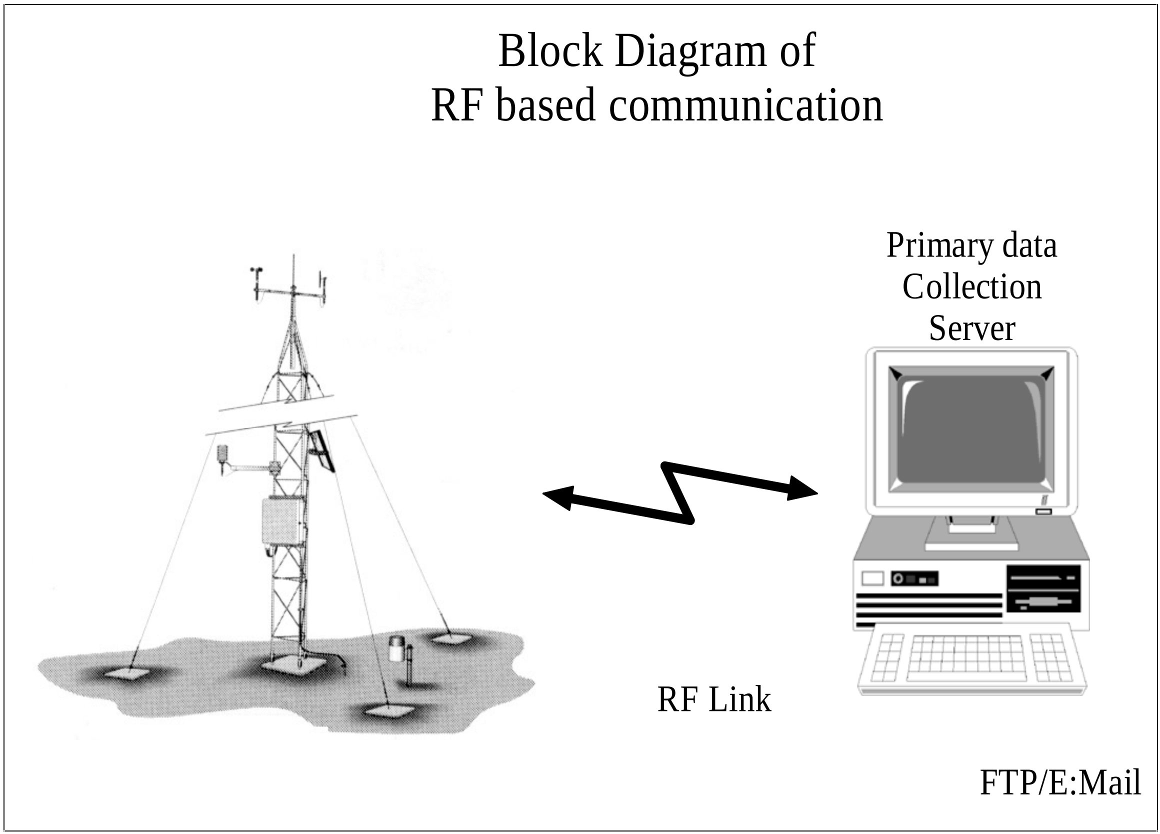



**Figure 3.1:** Block diagram of RF based communication.

The basic function of the RF Modem is to establish the communication link between the base station and the datalogger based on the station ID. The station ID for a station can be set using the jumper switches on the RF modem printed circuit board (PCB). There are eight jumper switches on the PCB that can be used to set the station ID from 1 to 254. The RF modem enables the communication request from the base station if and only if the station ID matches.

Since the mode of communication is simplex on half-duplex technique, packet based communication is implemented on the RF modem. In this technique, a continuous stream of data is broken down into packets and transmitted over the radio waves. The maximum packet size the RF modem uses for communication was 238 bytes. Once the RF modem receives 238 bytes through the communication port CSIO, the RF modem closes the packet, adds the header information and transfers the data to the RF radio through the "Data In" pin of the RF radio.

### 3.2.2 RF Radio

The RF radio used for this experiment was RF 302, manufactured by Johnson Data Incorporated (Model DL-3410). The RF 302 Radio converts the digital data into radio waves and vice versa, which can then be transmitted or received using a RF Antenna. RF 302 is designed to operate for frequencies in the range of 403 MHz to 512 MHz. The RF 302 radio uses the frequency modulation technique to modulate the digital data. For this experiment we used the frequency 460.26250 MHz for communication. The typical specifications of the radio are shown in Table 3-2.

**Table 3-2 :** Typical specifications of the RF 302 Radio

| | |
|---|---|
| Frequency Range | 400-520 MHz |
| Operating Voltage | 12V |
| Operating Temperature | -30C to 60C |
| Data Interface | Audio Subcarrier |
| RF Output Power at 12/13.8V | 3W/5W |
| RF Output Impedance | 50 Ω |
| Attack Time | Less than 3mS |
| Frequency Stability | 5 ppm |
| RF Input Impedance | 50 Ω |
| Carrier Detect Attack Time | Less than 10mS |

### 3.2.3 RF Antenna

Celwave Incorporated manufactured the antenna used for the RF communication. Two different kinds of antenna were used for this purpose. At the remote datalogger site we used a directional Yagi antenna and at the base station site we used the Omni directional antenna. The Yagi directional antenna was used at the remote station to enhance the signal strength at the base station end. Since the base station can access many remote data loggers, using a directional antenna was not advantageous. Hence, we used an Omni directional antenna for this purpose. The antenna was connected to the radio using a RG 8A/U cable that had attenuation of 165db per km at 400MHz.

Since the height at which the antenna is mounted is very critical for operation of the radios and communication, we mounted the antenna at the top of the tower at a height of approximately 10m. Also since the base station had an omni-directional antenna mounted vertically with horizontal polarization, we had to mount the yagi antenna also in the horizontal polarization for efficient communication.

### 3.2.4 Datalogger

As mentioned earlier we used a separate datalogger and a separate set of instruments for this experiment. The datalogger used was Campbell Scientific's CR10 datalogger. The datalogger was mounted on a separate enclosure and was mounted on the same tower. The datalogger was connected to the RF radio through a RF modem.

### 3.2.5 Base station computer

To collect the data from the datalogger and archive it we decided to use a dedicated system. The Operating system platform used for this communication was Linux as it offers remote access. Since the communication software needed to download the data from the remote site was either on a DOS or a Windows platform, we had to use DOS emulators on Linux (DOSemu) for this experiment. DOSemu provided the necessary DOS platform on linux.

With DOSemu we were able to execute the DOS communications package provided by Campbell Scientific Incorporated. The communication program was configured to download the recent (new) data from the remote site and exit after downloading. To execute the program at regular intervals we used the Linux built-in program "crontab" which is a scheduler program. A shell script was written which essentially executed the DOS communication program and processed the collected data.

## 3.3 RF/Internet-based communication

Even though RF offered features that proved to be efficient and cost-effective, RF links are limited by distance. Hence, in order to overcome this problem, we decided to use combined RF and Internet based communication. The block diagram of the flow of data is shown in Figure 3.2. In this technique, within the range of the RF link data was downloaded on to a computer that is on the Internet. The data is then forwarded to the server through the Internet where the data was further processed.

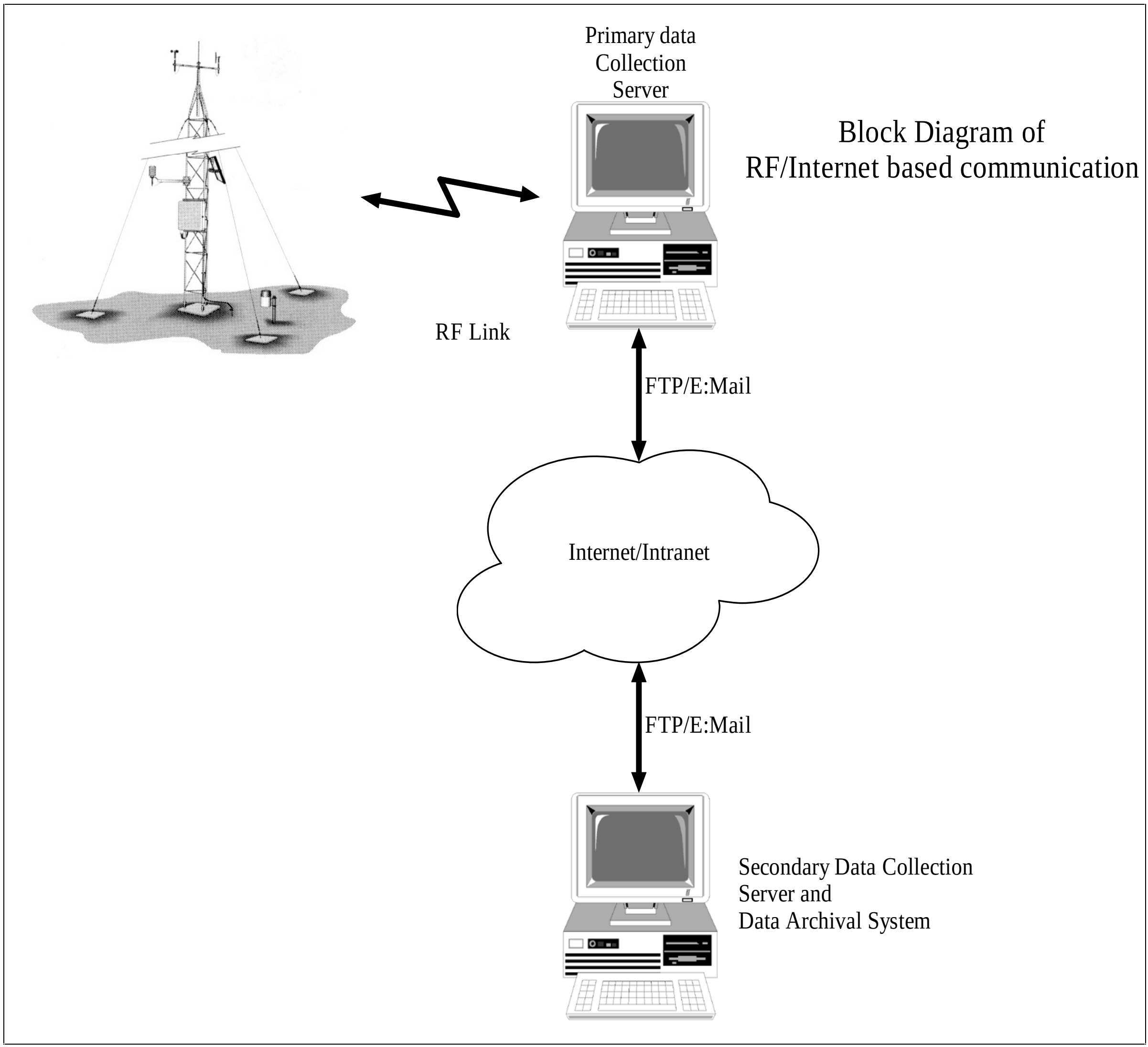



**Figure 3.2:** Block diagram of the RF/Internet based communication.

Since the remote access was required for the system, it was decided to use the Linux Operating System. Since PC 208W was designed to operate on a Windows platform, we decided to use the DOS version of PC 208. Using the DOS emulators on Linux, we were able to use DOS version of PC 208.

In order to test this technique, it was decided to experiment using the Varsity Lab and the Lake Wheeler site along with data published in real-time on the State Climate Office Web server. The data was collected on the base known as the primary data collection server computer at the Varsity lab using the RF Telemetry explained before. The shell script used for RF telemetry was modified to process the collected data. Please refer to Appendix C3 for the shell script written for the primary data collection server. The newer data was sent electronically via the mail system (e-mail) to another computer known as the secondary data collection server. E-mail mode of data transfer was preferred over the FTP mode of data transfer for security reasons. A separate shell script was written on the secondary data collection server, which collected the data from e-mail, sorted it on the basis of station and concatenated the data file. Please refer to Appendix C4 for the shell script written for the secondary data collection server.

## 3.4 North Carolina State Highway Patrol network

The North Carolina State Highway Patrol (NCSHP) is a State Law Enforcement Agency with full police powers and statewide jurisdiction. The NCSHP is a Division within the North Carolina Department of Crime Control and Public Safety (CC&PS). Though the most important assets of the NCSHP are the troopers on the road, the NCSHP also has a number of technology-based support tools. Premier amongst these technological tools are the NCSHP communications infrastructures and the associated radios and computers. For NCSHP communications, there exist three main infrastructures: Voice Radio Communications, Mobile Data Terminal (MDT) communications, and the NCSHP Intranet.

As the premier statewide traffic law enforcement organization, NCSHP is responsible for data connectivity for the Criminal Justice Information Network (CJIN) Mobile Data Terminal Network (MDT Network), the NCSHP IntraNet (connecting Communications centers and Highway Patrol District Offices), and the communications circuits connecting the numerous voice radio sites. Because of the statewide responsibility, NCSHP uses both organic organizational microwave circuits as well as terrestrial data communications. These interconnecting NCSHP sites are in almost every county in the state of North Carolina.

By using the infrastructure of the NCSHP Intranet and extending the concept of RF/Internet-based communication it would be possible to have real-time data communication between each AgNet station and the secondary data collection server. In order to do so, a primary data collection server has to be installed for each remote station. Since RF communication can communicate to multiple stations using the same frequency, it would also be possible to have a primary data collection server and many remote sites. The data collected from these sites can be transferred to the secondary data collection server by email using the NCSHP Intranet.

## 3.5 Real-time Data Dissemination

The data collected at the secondary data collection server has to be published to achieve our goal of real-time data dissemination. In order to achieve this we have to update the page with newer data at regular intervals. For testing purposes, it was decided that the page should be updated with newer data every 5 minutes. The datalogger program had to be modified to give five-minute and fifteen-minute samples. The shell script on the primary data collection server for collecting data was also modified to download data from the datalogger through the RF Link at regular intervals of 5 minutes. The newer data collected from the datalogger is e-mailed to the secondary data collection server. E-mail was preferred over FTP mode of transfer for security and scalability reasons. Also in case of NCSHP network, it was easier to transfer data through e-mail than FTP as there was a network firewall in between, which blocked many protocols including FTP.

The data collected at the secondary data collection server is transferred to the Web server over the local Internet. The data is published on the web server on a separate folder. The page contains 5-minute samples and a history of 2 hours of 5-minute samples. The sample page is as shown in Appendix C2. The page is generated using a separate program. A shell script was written which downloads the data from the secondary data server, processes the data file and executes the program that processed data file.

The complete block diagram of the real-time data dissemination is shown in Figure 3.3.

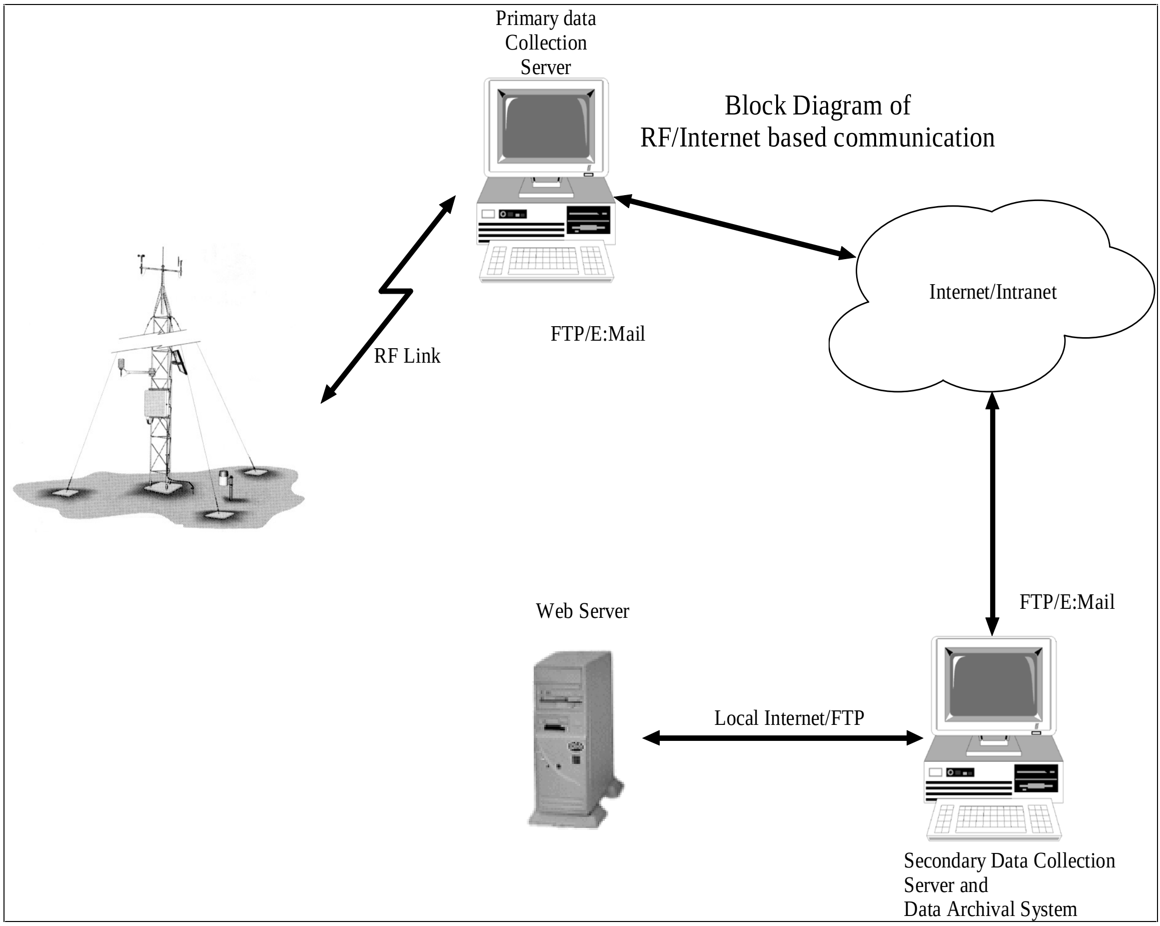



**Figure 3.3:** Complete block diagram of the real-time data dissemination system

From the figure we can make out the different stages of the system

- Remote site;
- RF Telemetry;
- Primary data collection server;
- Secondary data collection server; and,
- Web server.

The remote site used for this experiment was the Lake Wheeler Field Laboratory. The remote site had sensors for air temperature, relative humidity, wind speed, wind direction, solar radiation and soil temperature and a datalogger CR10. The datalogger program was incorporated with 5-minutes and 15-minutes instantaneous sample outputs. The datalogger program is shown in Appendix C1.

RF telemetry consisted on RF modem, RF radio and antenna at the remote site and a RF base station and computer at the base station end. The primary data collection server was housed at the NC State University Varsity Field Laboratory. The computer was connected to the RF base station through the RS 232 serial port. Linux was the operating system and the communication package for downloading the data was executed through the DOS emulation on Linux. The newer data was downloaded at regular intervals, and e-mailed to a user 'agnet' on the secondary data collection server. A copy of the data was archived on the primary data collection server as a backup.

The secondary data server was housed at the State Climate Office facility. The data arriving through the e-mail was processed and concatenated into the data file on the basis of station name. The data on the mailbox is deleted to conserve disk space. The Web server then downloads data through FTP from the secondary data server and publishes the data on the web folder.

# CHAPTER 4

# QUALITY CONTROL PROCEDURES

It is a known fact that noise is present in all system. During measurement there are a variety of sources, which inject noise into the system. For example, drift in the electronics due to varying operating temperature varies the output signal voltage, long wires used to connect the sensor to the datalogger act as antennas and Electro-magnetic Induction (EMI) induced voltage is injected into the output signal voltage, etc. Since noise sources cannot be completely eliminated, it is necessary to come up with techniques that can blank out these noise sources. These techniques are known as Quality Control (QC) procedures. Quality Control procedures are used to increase the confidence in the data. In this chapter we will be discussing about the different Quality Control procedures that have been implemented to have a better confidence in data.

## 4.1 Types of Quality Control procedures

### 4.1.1 Instrument Range Check

The static range of the sensor limits the output of the instrument. For example, HMP 45C air temperature instrument is designed to measure temperature in the range of –40°C to +50°C and relative humidity measure relative humidity in the range of 0-100%. Hence any measurement which out of these ranges is definitely an invalid data point. The instrument range check is implemented to handle this problem. In this check, the measurement of each instrument is compared against the limits of the sensor. Any data point that is out of range of the instrument is blanked out. The blanking procedure is done replacing the data point with a user-specified value. The current implementation of blanking procedure replaces the data point with a value of 99999.

### 4.1.2 Instrument Noise Check

Instruments like relative humidity, solar radiation, photosynthetically active radiation and wind direction have bounded outputs. For example, wind direction can vary only between 0° and 360°, while solar radiation measurements are bounded between 0 to 1380Wm$^{-2}$. Hence any measurements which are above the bounds is a definite indication of instrument noise. This noise can be due to either a drift in the electronics of the instrument/amplifiers or the quantization error of the analog-to-digital converter (ADC). In order to solve this problem, instrument noise check was implemented. Any data point that was outside the limits of the boundary is replaced with the value nearest to the limits of the bound. For example, a value of 102% in relative humidity is replaced with 100%.

The solar radiation sensor, due to ambient RF Noise, generates a negative current during nighttime. This is reflected in the datalogger as a negative voltage and hence as a negative solar radiation values in the data. Since the radiation sensor is designed only to measure incoming radiation, any negative value in the radiation is a sure sign of an invalid data point. Using the instrument noise check these negative data points can be replaced with lower limit of the boundary, that is zero.

### 4.1.3 Relation Check

Since different parameters in the weather data are strongly correlated, we can use the correlation to check the confidence in the data. For example, there is a strong correlation between the air temperature and soil temperature with a time lag between the two parameters. Also there is a strong correlation between the precipitation and soil moisture. Hence performing a correlation between the parameters and determining the correlation coefficient, we can decide about the quality of data being sampled from the instrument.

### 4.1.4 Variability Check

This check is used to check the working of the sensors. Generally when either the sensor output or the sensor input is short or open, the output of the sensor stabilizes to a constant value. Depending upon the situation, the output voltage can settle within the output voltage range of the sensor. By using the variability check it is possible to determine whether the instrument has failed. In this check, the standard deviation and the mean values of the data are computed. If standard deviation of the data is lower than the lower limits, it is indication of instrument failure. This check however, fails for parameters that do not have a diurnal variation such as precipitation.

### 4.1.5 Buddy Check

In a network that has automated weather monitoring stations close to each other data collected at a station can be verified with a neighboring station. By comparing similar parameters under certain conditions, such as radiation on a clear day, it is possible to know whether the instrument is malfunctioning or has a bias in its output. This kind of check assumes that there exists a positive correlation for measured parameters between the two stations, as the two stations are geographically near. For stations to be geographically near, density of these stations should be of fine resolution. Hence for a weather network which has a coarse station density, this check would not be of significant importance.

## 4.2 Implementation of Quality Control Procedures

The quality control (QC) procedures were implemented by modifying the convert program, which was used for publishing the web pages. A separate quality control module was integrated into the program that performed the quality control procedures. The program was modified such that the normal operation of the publishing was unhindered and quality control checks can be done in either stand alone or integrated mode.

In the stand alone mode, by specifying the command line switch "–qc_only", the program executed only the quality control module after loading the quality control constants for the quality control file. During execution, the program scans the data and any quality control failure is reported either on the console or to a file.

In the integrated mode, the quality control module loaded the quality control constants from the file and "on the fly" scanned the data from the data file and was later processed for publishing on the web server. For normal operation, integrated mode scanning was implemented. The quality control could be disabled either through command line switch of "-no_qc" or by assigning "QC_CHECKS" variable to "NO" in the configuration declaration.

Two types of checks were implemented in the Quality Control module, namely

- Instrument Range Check; and,
- Instrument Noise Check.

The QC module loads the file QC file, which is declared in the configuration file as configuration variable "QC_FILE". Depending upon either the configuration declaration "QC_CHECK" or command line switch "-qc_only" or "-no_qc", the QC module is executed. Please refer to Appendix D1 for a typical QC file.

The bounds for the range check can be declared in the QC file in the following format

**"range: {lower limit} < {parameter name} < {upper limit} {month=[list of months]} {replace=[replace value]}"** where

**range –** Indicates that the declaration is Instrument's measurement range

{**lower limit**} **–** Indicates the lower bound limit of the instrument. For example for Relative humidity lower bound is zero

{**upper bound**} **–** Indicates the upper bound limit of the instrument. For example, upper bound for Relative Humidity is 100

{**parameter name**} – Name of the parameter whose range is being declared. Different parameter names that can be declared are AirTempC for air temperature in celsius, RelHum for relative humidity, SoilTempC for soil temperature in celsius, Pressure for barometric pressure, Precip for Precipitation, SoilMoist for soil moisture, SolarRad for solar radiation and PAR for photosynthetically active radiation.

{**month=[list of months]**} – Indicates the valid months when the range of the instrument is valid. Weather parameters like relative humidity, wind direction have upper and lower bounds fixed through out the year, while parameters like Air Temperature can have a varying range through out the year even though the measurement range is fixed. The months for the range are declared as comma separated month numbers.

{**replace=[replace value]**} – Replacing value for the data point in case the condition for the range is not satisfied.

Typical example for Air Temperature during months of January, Febuary and March is

“range: -30<AirTempC<30 month=1,2,3 replace=99999”

The instrument noise check constants are declared using the “clean” variable. The format to declare instrument noise check is as shown below

“**clean**: {**parameter name**} [< or >] {**parameter bound value**} {**replace**=[**replace value**]}” where

**clean –** Indicate the declaration is instrument noise check declaration

{**parameter name**} – Is the name of the parameter whose noise check is declared. The parameter name declaration is similar to the declaration of the range check.

[< or >] – Condition for value declaration

{**parameter bound value**} – Is the value within which the parameter has to be bounded. Any value out of this bound indicates instrument noise.

{**replace=[replace value]**} – Is the replacing value in case of failure.

Typical example for relative humidity is

“clean: Relhum < 0 replace=0”

“clean: RelHum > 100 replace=100”

Once when the QC file is parsed and loaded, the module is now ready to accept data and check the data for QC failures. Any QC failures are reported either to the console or the QC log file based on the configuration declaration “LOG_QC”.

Since there are only 17 stations located throughout the state, the network is very coarse. Implementing “Buddy Check” on such a network will not yield good results, as the heterogeneous environment between the two neighboring stations can no longer be considered to be gradient.

# CHAPTER 5

# EXAMPLES OF ENHANCED AGNET DATA

Some examples of products that can be derived from the AgNet data are presented in this chapter. AgNet data were very significant during Hurricane Floyd and Hurricane Dennis. Using MATLAB it was possible to get the polar plots of the winds, which are very useful in visualizing the winds in and around the affected areas. These plots generated in real-time will be very useful for many applications.

## 5.1 AgNet products for Hurricane Dennis

Hurricane Dennis made landfall at Cape Lookout National Seashore, North Carolina on September 4$^{th}$, 1999 at 17:00 hrs after re-intensification. At the time of landfall, the wind was just below hurricane-force winds. Figure 5.1, generated using Kinston AgNet data, indicates the direction of the winds and variation of pressure during the event. Figure 5.2, generated using Reedy Creek AgNet data, indicates the direction of the winds and variation of pressure during the event. From the two figures, one can notice the pressure trends and winds direction and estimate the movement of the eye of the hurricane. From the values of pressure and wind speed, one can estimate the wind gusts and also decide about the safety for rescue operations. These plots generated in real-time can be a useful tool for North Carolina Emergency Management and Rescue.

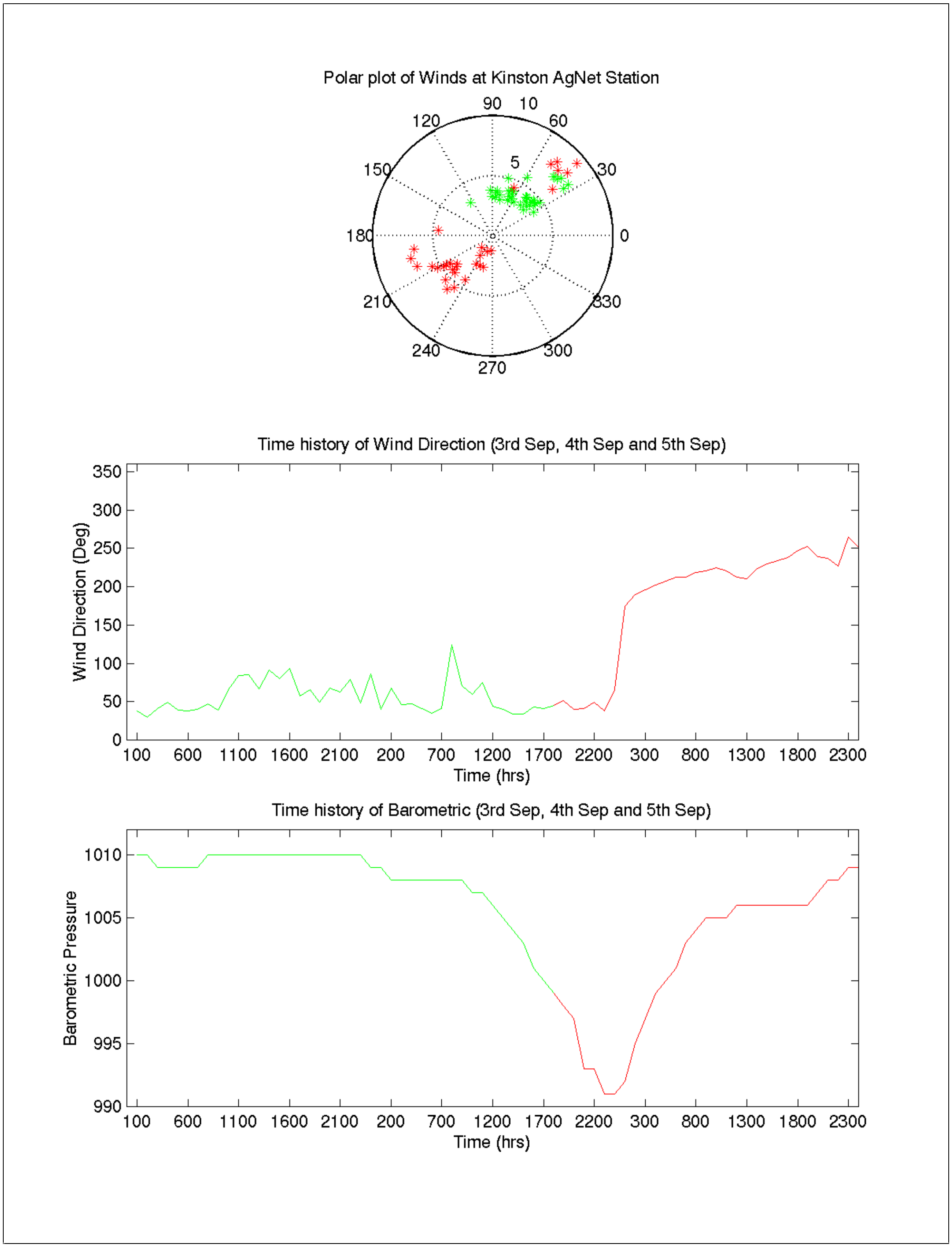



**Figure 5.1:** Plots of wind and time history of pressure during Hurricane Dennis at Kinston AgNet Station

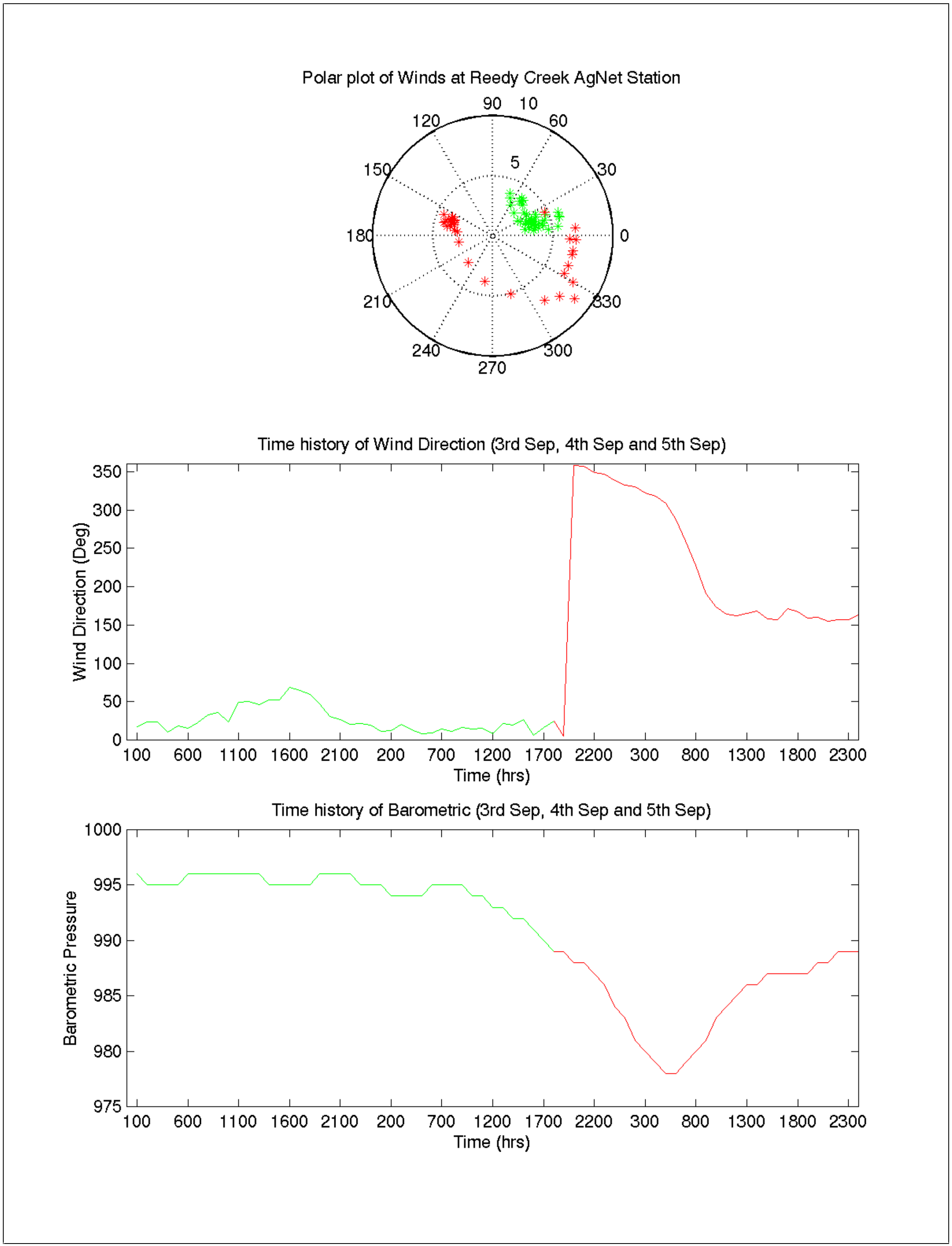



**Figure 5.2:** Plots of wind and time history of pressure during Hurricane Dennis at Reedy Creek AgNet Station

## 5.2 AgNet products for Hurricane Floyd

As mention earlier, after Hurricane Floyd made landfall on the coast of North Carolina on September 14, 1999 at 0230 hrs, the North Carolina Emergency Management and Rescue Division had to begin the rescue operation for those affected by hurricane. Using the polar plots of winds at Castle Hayne, Kinston and Lewiston-Woodville (Figure 5.3, Figure 5.4 & Figure 5.5) at intervals of 6-hours for 12-hours before landfall and 24-hours after landfall, one can estimate the strength of surface winds and also the direction. We notice from the plots that within 0-6 hours of the landfall winds at Castle Hayne rose from 5-10 $m.s^{-1}$ to 10-20 $m.s^{-1}$. While at Kinston, winds increased from 5-10 $m.s^{-1}$ to 10-20 $m.s^{-1}$ at around 3-6 hours of landfall and around 6-12 hours of landfall at Lewiston-Woodville. The North Carolina Emergency Management and Rescue Division used this data during the Hurricane Floyd to take necessary steps for rescue operations.

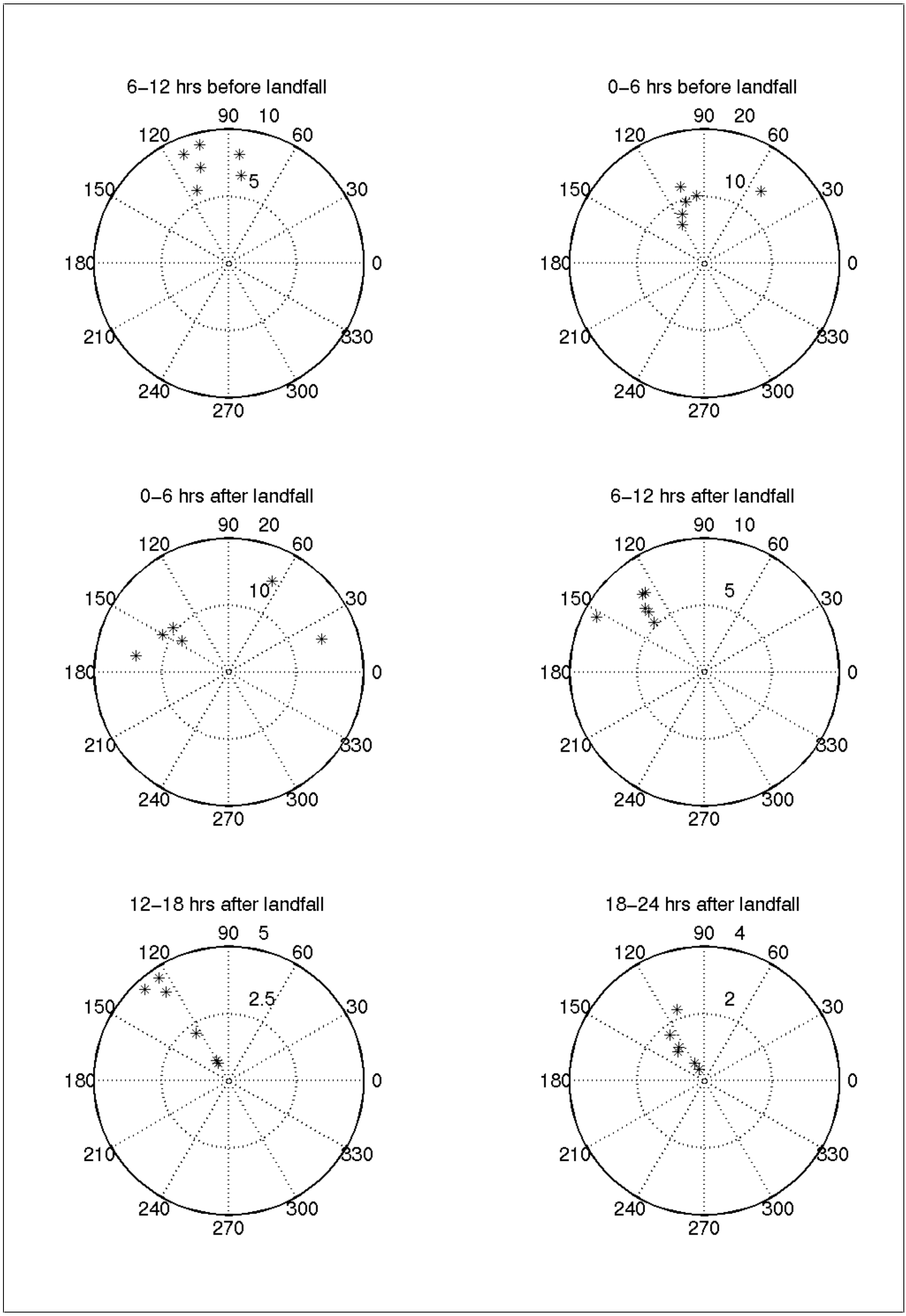



**Figure 5.3:** Polar plots of winds at Castle Hayne for 12 hours before and 24 hours after landfall of Hurricane Floyd

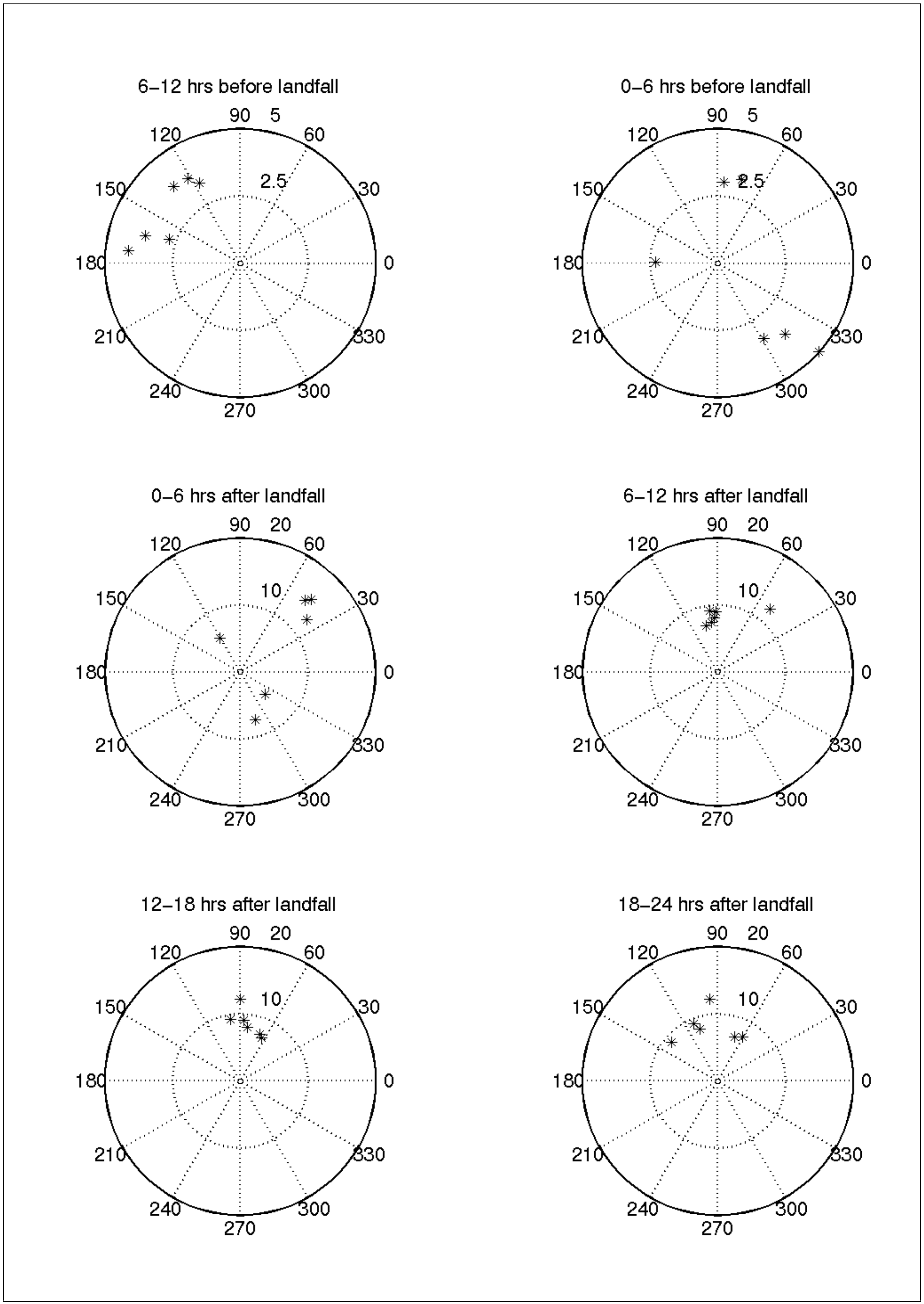



**Figure 5.4:** Polar plots of winds at Kinston for 12 hours before and 24 hours after landfall of Hurricane Floyd

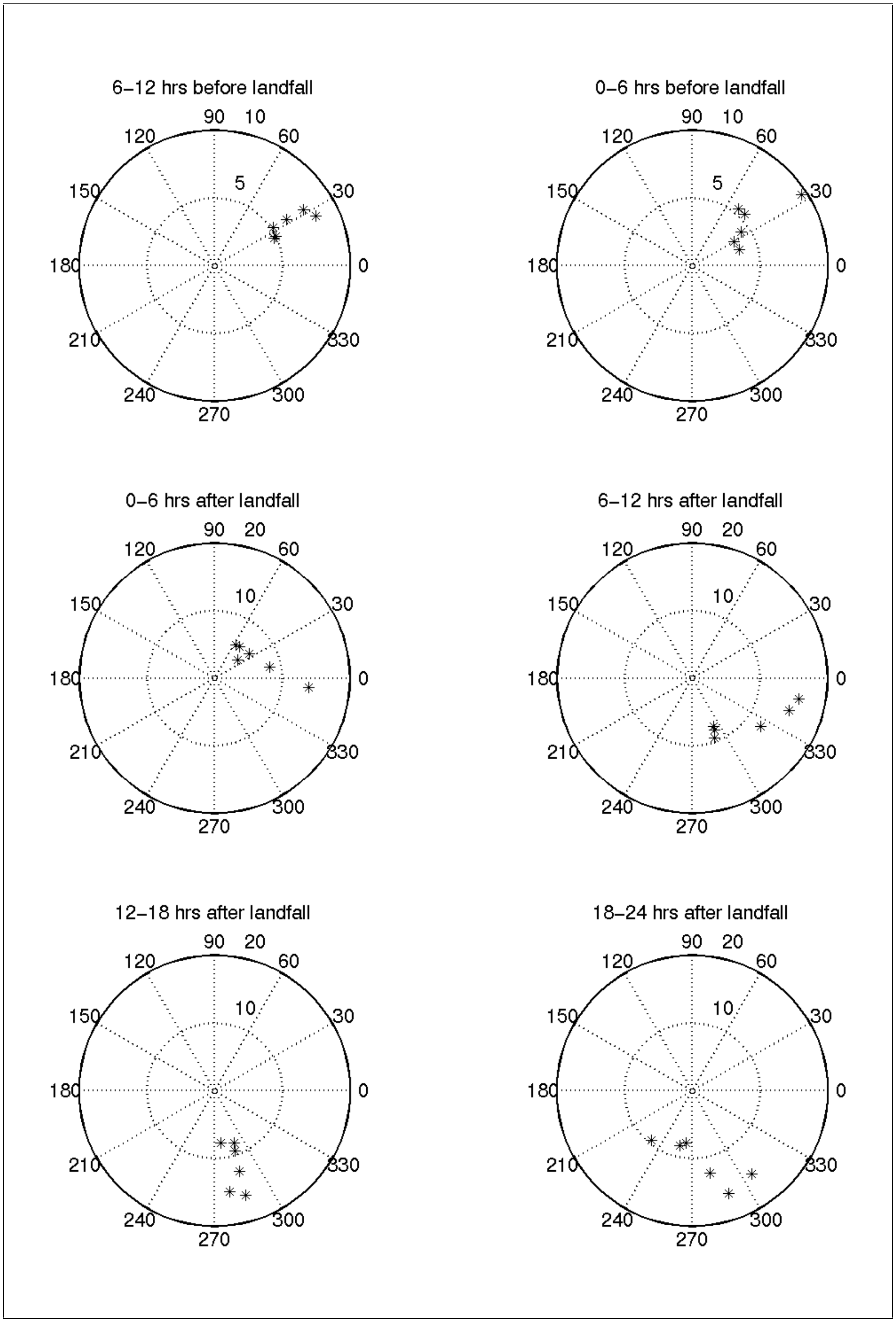



**Figure 5.5:** Polar plots of winds at Lewiston-Woodville for 12 hours before and 24 hours after landfall of Hurricane Floyd

# CHAPTER 6

# SUMMARY AND FUTURE WORK

## 6.1 Summary

This thesis describes steps taken and designs involved in the modernization of the North Carolina AgNet weather network and enhancements that were made to achieve real-time data communication and data dissemination.

With the modernization, the upgraded AgNet weather networks with wind sensor at 10m height, temperature sensor at 2m height, precipitation sensor at 1m height and soil temperature sensor at 0.1m depth is in compliance with the WMO Standards. The exposure of the instruments are also in conformation with the WMO and EPA standards with unrestricted view for radiation and wind sensors, radiation shield for air temperature, uniform soil for soil temperature and soil moisture.

The entire network configuration is now homogenous and uniform with similar instruments through out the network. This provides a uniform data set for all the weather stations in AgNet. The soil moisture sensor has been added to all the stations in the network and hence a continuous soil moisture dataset is now available at regular intervals of one hour.

With the development of the “CONVERT” program, daily and hourly data are updated everyday at 6:00 AM on the State Climate Office web page. The new data web pages are created in a hierarchical order with links to the pages in lower hierarchy.

With the implementation of RF and Internet based communication, the data are transferred in real-time. By modifying the datalogger program to collect five-minute samples and using the RF/Internet communication technique, these samples are transferred to the secondary data collection server in real-time. With five minute updates arriving at the secondary data collection server in real-time, the web page is updated at the same interval.

With preliminary QC implemented effects of certain kinds of noises such as RF noise in solar radiation and saturation in relative humidity can be reduced, thereby increasing the confidence in data.

## 6.2 Future Work

The goal was to modernize and enhance the AgNet Weather Network. Even though AgNet was modernized, there are still areas in which AgNet needs improvements. These can be listed as

- Data Resolution of the Network;
- Installation of RF Communication throughout the network;
- Structured Query Language (SQL) Database server at the collection server;
- Data dissemination based on Java with SQL interface for data; and,
- Implementation of other QC Algorithms.

Since AgNet currently has only seventeen stations through out the state, the resolution of the data is coarse. In order to have an effective data set, the data resolution must be increased which means installation of more AgNet stations.

The RF/Internet communication between datalogger and secondary data collection server was tested as proof of concept between Lake Wheeler Field Lab and the NC State University Varsity Lab. The communication techniques were found to be reliable and cost-effective. Hence implementing the communication over the complete network using the NC-SHP network would be very useful.

Since the "CONVERT" program generates static HTML pages, there is a continuous usage of hard disk space over the web server. By using Java and Perl languages, it is possible to generate the HTML pages dynamically. The performance of the web server with dynamically generated HTML pages is far better than static HTML pages. But in order to implement a dynamic HTML page server, a back-end SQL database server is needed. Hence implementing an SQL database server as a back end application with Java based front end graphical and textual interface would be better for data dissemination.

As the resolution of the network increases, other QC algorithms such as buddy checks can be implemented to improve data quality.

# REFERENCES


1. 05103 and 05305 R. M. Young Wind Monitor Instruction Manual, Campbell Scientific, Inc., Rev 9/96, Campbell Scientific., Inc., Logan, Utah 84321.
2. Barnett, T.P., J. Ritchie, J. Foat and G. Stokes. 1998. On the Space-Time Scales of the Surface Solar Radiation Field. Journal of Climate 11(1):88-96.
3. Baynton, H. W., 1976: Errors in Wind Run estimates from Rotational anemometers, Bul. Am. Met. Soc., Vol 57, No. 9, 1127-1130.
4. Brock, F. V., Crawford, K. C., Elliott, R. L., Cuperus, G. W. Johnson, H. L., Eilts, M. D., The Oklahoma Mesonet: A technical overview, Journal of Atm. and Oceanic Tech., Vol 12, 5-19
5. Brotzge, J.A. and K.C. Crawford. 1999. Estimating Sensible Heat Flux from the Oklahoma Mesonet. Journal of Applied Meteorology in press for Dec 1999.
6. Carlson, J.D. 1998. The Oklahoma Alfalfa Weevil Model: What It Is and How to Use It. Oklahoma Alfalfa Hay and Seed Association Newsletter: pp.1.
7. COM200 Telephone Modem Instruction Manual, Campbell Scientific, Inc. Rev 7/98, Campbell Scientific., Inc., Logan, Utah 84321.
8. Crawford, K. and M. Morrissey. 1998. Using the Oklahoma Mesonetwork to Develop and Test a Practical Sampling Error Statistic for Meteorological Time Series. J. Geophys. Res. Vol. 103 , No. D8 , p. 8979
9. CS 615 Water Content Reflectometer Instruction Manual, Campbell Scientific, Inc., Rev 10/96, Campbell Scientific, Inc., Logan, Utah 84321.
10. CS105 Barometric Pressure Sensor Instruction Manual, Campbell Scientific, Inc., Rev 3/97, Campbell Scientific, Logan, Utah 84321.
11. Dalton, F. N., W. N. Herkelrath, D. S. Rawlins, & J. D. Rhoades. 1984. Time domain reflectometry: Simultaneous measurement of soil-water content and electrical conductivity with a single probe. *Science* (Washington) 224:989-990.
12. Damicone, J.P., K.E. Jackson, J.R. Sholar and M.S. Gregory. 1994. Evaluation of a Weather-Based Spray Advisory for Management of Early Leaf Spot of Peanut in Oklahoma. Peanut Science 21:115-121.

13. Davis, J. L. & A. P. Annan. 1977. Electromagnetic detection of soil moisture. Progress report 1. *Can. J. Remote Sensing 3*. 1:76-86.
14. Elliott, R.L., F.V. Brock, M.L. Stone and S.L. Harp. 1994. Configuration Decisions for an Automated Weather Station Network. Applied Engineering in Agriculture 10(1):45-51.
15. EPA (1987). On-Site Meteorological Program Guidance for Regulatory Modeling Applications, EPA-450/4-87-013. Office of Air Quality Planning and Standards, Research Triangle Parks, North Carolina 27711.
16. EPA (1989). Quality Assurance Handbook for Air Pollution Measurement Systems, EPA Office of Research and Development, Research Triangle Park, North Carolina 27711.
17. Fellner-Feldegg, H. R. 1969. The measurement of dielectrics in the time domain. *J. Phys. Chem*. 73:616-623.
18. Gill, G. C., 1973: The Helicoid Anemometer Atmosphere, II, 145-155
19. Goff, J. A. and S. Gratch, 1946: Low pressure properties of water from -160°F to 212°F, Trans. Amer. Soc. Heat Vent. Eng., 51, 125-164
20. Greene, J.S. and M.L. Morrissey. 1998. Evaluation and Validation of Simulated and Observed Climate Data. Climate Prediction for Agricultural and Resource Management.
21. Heimovaara, T. J. 1993. Time domain reflectometry in soil science: Theoretical Backgrounds. Measurements and Models. Ph.D. dissertation. Universiteit van Amsterdam, 169 pp
22. HMP45C Temperature and Relative Humidity Probe, Campbell Scientific, Inc., Rev4/99, Campbell Scientific, Inc., Logan, Utah 84321.
23. Jacobsen, O. H. & P. Schjonning. 1994. Comparison of TDR Calibration Functions for Soil Water Determination. Applications in Soil Science, Research Centre Foulon, Ministry of Agriculture and Fisheries, Danish Institute of Plant and Soil Science.
24. LI190SB Quantum Sensor Instruction Manual, Campbell Scientific, Inc., Rev 6/97, Campbell Scientific, Inc., Logan, Utah 84321.
25. LI200X Pyranometer Instruction Manual, Campbell Scientific, Inc., Rev 2/97, Campbell Scientific, Inc., Logan, Utah 84321.

26. Lowe, P. R. 1977: An approximating polynomial for the computation of saturation vapor pressure, J. Appl. Meteor., 16, 100-103
27. McPherson, R.A. and K.C. Crawford. 1996. The EARTHSTORM Project: Encouraging the Use of Real-Time Data from the Oklahoma Mesonet in K-12 Classrooms. Bulletin of the American Meteorological Society 77(4):749-761.
28. Model 107 Temperature Probe Instruction Manual, Campbell Scientific, Inc., Rev 9/97, Campbell Scientific, Inc., Logan, Utah 84321.
29. Morrissey, M.L. and J.S. Greene. 1998. Using the Oklahoma Mesonet to Develop and Test a Sampling Error Statistic for Meteorological Time Series. Journal of geophysical Research 103 (D8):8979-8984.
30. MSX10, MSX10R and MSX20R Solar Panels Instruction Manual, Campbell Scientific, Inc., Rev 12/94, Campbell Scientific., Logan, Utah 84321
31. Pereira Fo., A.J. and K.C. Crawford. 1999. Mesoscale Precipitation Fields. Part I: Statistical Analysis and Hydrologic Response. Journal of Applied Meteorology 38(1): 82-101.
32. Pereira Fo., A.J., K.C. Crawford and C.L. Hartzell. 1998. Improving WSR-88D Hourly Rainfall Estimates. Weather and Forecasting 13: 1016-1028.
33. Pereira Fo., A.J., K.C. Crawford and D.J. Stensrud. 1999. Mesoscale Precipitation Fields. Part II: Hydrometeorologic Modeling. Journal of Applied Meteorology 38(1): 102-125.
34. Petersen, L. W. 1994. Sampling volume of TDR probes used for water content monitoring: Practical investigation. Proceedings of the Symposium on Time Domain Reflectometry in Environmental, Infrastructure, and Mining Applications, Evanston, Illinois, Sept 7-9, U.S. Bureau of Mines, Special Publication SP 19-94, NTIS PB95-105789, pp. 93-104.
35. Radiotelemetry Network Introduction packet, Campbell Scientific, Inc., Rev 4/95, Campbell Scientific., Inc., Logan, Utah 84321.
36. Radiotelemetry Network, Campbell Scientific, Inc., Rev 5/99, Campbell Scientific., Logan, Utah 84321

37. Richardson, S.J. 1995. Automated Temperature and Relative Humidity Calibrations for the Oklahoma Mesonetwork. Journal of Atmospheric and Oceanic Technology 12(4):951-959.
38. Shafer, M. A., T. Hughes, and J. D. Carlson, 1993: The Oklahoma mesonet: Site Selection and Layout. Preprints, Eight Symp. On Meteorological Observations and Instrumentation, Anaheim, CA, Amer. Meteor. Soc., 231-236.
39. Tanner, Bertrand D. (1990). Automated Weather Stations, Remote Sensing Reviews, 1990, Vol. 5 (1), p.73-98.
40. TE 525 Tipping Bucket Rain Gauge Instruction Manual, Campbell Scientific, Inc., Rev 6/98, Campbell Scientific, Inc., Logan, Utah 84321.
41. The State Climatologist (1985) Publication of the American Association of State Climatologists: Heights and Exposure Standards for Sensors on Automated Weather Stations, v. 9, No. 4 October 1985.
42. UT30 Weather Station Installation Manual, Campbell Scientific, Rev 4/98, Campbell Scientific, Inc., Logan, Utah 84321.
43. Weiss, A. 1977: Algorithms for the calculation of moist air properties on a hand calculator, Amer. Soc. Ag. Eng., 20, 1133-1136.
44. WMO (1983). Guide to Meteorological Instruments and Methods of Observation. World Meteorological Organization No. 8, 5th edition, Geneva Switzerland.

# APPENDIX A1

## SAMPLE SOURCE CODE OF "CONVERT"

File "global.h" is header file that contains global variable declarations. The file "convert.c" is the main 'C' file of the convert program.

———————————————File: global.h———————————————

```
// Header file for "CONVERT" program
// Include common definitions from another external include file
#include "defs.h"

// Structure for storing data
struct Data_Format *Data;
struct Data_Struct *data_struct;
struct Data_Struct *html_struct;
struct Data_Present *Data_table;
struct ArrayID *ArrayID;

// Structures for storing Quality control constants
struct clean   qc_clean[NUM_OF_QC_CLEAN];
struct range   qc_range[NUM_OF_QC_RANGE];

// Definition of global variables. These variables are visible globally
// throughout the program
unsigned char   config_file[PATHSIZE];
// Variable contain name of the Configuration File
unsigned char   data_file[PATHSIZE];
// Variable contains name of the data file
unsigned char   header_file[PATHSIZE];
// Variable contains the name of the header file
unsigned char   footer_file[PATHSIZE];
// Variable contains the name of the footer file
unsigned char   qc_file[PATHSIZE];
```

```
// Variable contains the name of the quality control values file
unsigned char   mail_file[PATHSIZE];
// Variable contains the name of the Mail log file
unsigned char   work_file[PATHSIZE];
// Variable contains the present working file
unsigned char   parameter_value[SIZE_OF_PARAMETER];
// Variable used for transfer working parameter converted into ascii.
//unsigned char   log_error = 0;
// Variable for logging condition

//unsigned char   log_mail = YES;
// Condition to create a mail log file
unsigned char   error_log_config = ALL;
// Variable for error logging condition
unsigned char   mail_log_config = ALL;
// Variable for mail logging condition
unsigned char   sort_data = YES;
// Condition for sorting out data from the RAW Data
unsigned char   log_file[PATHSIZE];
// Variable contains the name of the Error Log file
unsigned int    mode = AUTO;
// Variable for holding mode definition
unsigned char   debug=0;
// Variable to hold the debug condition
unsigned char   main_page=NO;
// Variable to compulsory update of main page.
unsigned char   quiet=NO;
// Variable to hold the Output Condition

unsigned char   mainpage=NO;
unsigned int    input_month;
unsigned int    input_year;
unsigned int    table_size;

// # of days for which data exists
unsigned char   qc_warn_color[COLOR_DEF_LENGTH];
// Pointer for storing the warning color
unsigned char   qc_error_color[COLOR_DEF_LENGTH];
```

```
// Pointer for storing the error color
unsigned int    start_year;
// Variable to store the start year of the data
unsigned int    start_month;
// Variable to store the start month of the data
unsigned int    noaa_id;
// Variable to store NOAA ID
unsigned char   station_city[MAX_STRING_LENGTH];
// Variable to save the station Location
unsigned char   station_address1[MAX_STRING_LENGTH];
// Variable to save the station Location

unsigned char   station_address2[MAX_STRING_LENGTH];
// Variable to save the station Location
unsigned char   station_zipcode[5+1+4];
// Variable to hold the zip code (Zipcode + Hyphen + PO Box)
unsigned char   station_name[MAX_STRING_LENGTH];
// Variable to save the station name
unsigned char   station_id[MAX_ID_LENGTH];
// Variable to save the Station ID
unsigned char   station_county[MAX_STRING_LENGTH];
// Variable to save the Station County
unsigned char   station_lat[MAX_STRING_LENGTH];
// Variable to save the Station Latitude
unsigned char   station_long[MAX_STRING_LENGTH];
// Variable to save the Station Longitude
unsigned char   station_elevation[MAX_STRING_LENGTH];
// Variable to save the Station Elevation
unsigned char   station_das[MAX_STRING_LENGTH];
// Variable to save the Station Data Acquisition System
unsigned char
sensor_airtemp_l1[MAX_STRING_LENGTH],location_airtemp_l1[MAX_STRING_LEN
GTH];
// Variable to store type and location of Air Temp. Sensor at level 1
unsigned char
sensor_relhum_l1[MAX_STRING_LENGTH],location_relhum_l1[MAX_STRING_LENGT
H];
```

```
// Variable to store type and location of Relative Humidity Sensor at
level 1
unsigned char
sensor_airtemp_l2[MAX_STRING_LENGTH],location_airtemp_l2[MAX_STRING_LEN
GTH];
// Variable to store type and location of Air Temp. Sensor at level 2
unsigned char
sensor_relhum_l2[MAX_STRING_LENGTH],location_relhum_l2[MAX_STRING_LENGT
H];
// Variable to store type and location of Relative Humidity Sensor at
level 2
unsigned char
sensor_precip[MAX_STRING_LENGTH],location_precip[MAX_STRING_LENGTH];
// Variable to store type and location of Precipitation Sensor
unsigned char
sensor_windspd[MAX_STRING_LENGTH],location_windspd[MAX_STRING_LENGTH];
// Variable to store type and location of Wind Speed Sensor
unsigned char
sensor_winddir[MAX_STRING_LENGTH],location_winddir[MAX_STRING_LENGTH];
// Variable to store type and location of Wind Direction Sensor
unsigned char
sensor_pressure[MAX_STRING_LENGTH],location_pressure[MAX_STRING_LENGTH]
;
// Variable to store type and location of Pressure Sensor
unsigned char
sensor_soiltemp[MAX_STRING_LENGTH],location_soiltemp[MAX_STRING_LENGTH]
;
// Variable to store type and location of Soil Temperature Sensor
unsigned char
sensor_soilmoist[MAX_STRING_LENGTH],location_soilmoist[MAX_STRING_LENGT
H];
// Variable to store type and location of Soil Moisture Sensor
unsigned char
sensor_solarrad[MAX_STRING_LENGTH],location_solarrad[MAX_STRING_LENGTH]
;
// Variable to store type and location of Solar Radiation Sensor
unsigned char
sensor_par[MAX_STRING_LENGTH],location_par[MAX_STRING_LENGTH];
```

```
// Variable to store type and location of PAR Sensor
unsigned char
sensor_others[MAX_STRING_LENGTH],location_others[MAX_STRING_LENGTH];
// Variable to store type and location of Other Sensor
unsigned char   base_dir[PATHSIZE];
unsigned char   data_dir[PATHSIZE];
unsigned char   log_dir[PATHSIZE];
unsigned char   html_dir[PATHSIZE];
unsigned char   daily_val_dir[PATHSIZE];
unsigned char   hourly_val_dir[PATHSIZE];

unsigned char   daily_data_generic[PATHSIZE];
// Variable to save the template for daily data file
unsigned char   hourly_data_generic[PATHSIZE];
// Variable to save the template for hourly file
unsigned char   stat_html_generic[PATHSIZE];
// Variable to save the template for Statistics HTML Page
unsigned char   hourly_val_html_generic[PATHSIZE];
// Variable to save the template for Daily HTML Page
unsigned char   daily_val_html_generic[PATHSIZE];
// Variable to save the template for Hourly HTML Page
unsigned char   calendar_html_generic[PATHSIZE];
// Variable to save the template for Calendar HTML Page
unsigned char   main_html[PATHSIZE];
// Variable to save the template for Main HTML Page
unsigned char   month_html_generic[PATHSIZE];
// Variable to save the template for Month HTML Page
unsigned char   main_html_format[MAX_DATA_LENGTH];
// Variable to hold the html tags before and after the Year Desc. in
Main HTML file
unsigned char   month_html_format[MAX_DATA_LENGTH];
// Variable to hold the html tags before and after the link Desc. in
Month HTML file
unsigned char   tripod_link[PATHSIZE];
// Variable to hold the location of the link.
unsigned char   tower_link[PATHSIZE];
// Variable to hold the location of the tower
unsigned char   station_log[PATHSIZE];
```

```
// Variable to hold the location of the Station Log File
unsigned char   calendar_page_html_format[MAX_DATA_LENGTH];
unsigned char   dummy_month_html_format[MAX_DATA_LENGTH];
unsigned char   dummy_calendar_html_format[MAX_DATA_LENGTH];
unsigned char   space_calendar_html_format[MAX_DATA_LENGTH];
unsigned char   hourly_values_html_format[MAX_DATA_LENGTH];
unsigned char   daily_values_html_format[MAX_DATA_LENGTH];
unsigned char*  pass_error;
// Variable to pass the error string for logging
unsigned char*  pass_mail;
// Variable to pass the mail string for logging
const char*     name_type[] = { "Generic"
                                "Hourly",
                                "Daily",
                                "Half Hourly",
                                "Quarter Hourly",
                                "Five Minutes" };

unsigned int       julday[] = {
0,31,59,90,120,151,181,212,243,273,304,334,365 };
unsigned char   num_days[] = { 31,28,31, 30, 31, 30, 31, 31, 30, 31,
30, 31};
const int       max_fields[] = { 48, 24, 31, 48, 96, 288 };

FILE*           config_stream;
// FILE Stream to handle Configuration File.
FILE*           header_stream;
// FILE handle for Header File
FILE*           footer_stream;
// FILE handle for Footer File
FILE*           data_stream;
// FILE handle for Data File
FILE*           qc_stream;
// FILE handle for Quality Control Values File
FILE*           log_stream;
// FILE handle for Error Logging
FILE*           mail_stream;
// File handle for Mail Log stream
```

```
FILE*           work_stream;
// File handle for all Outputs to file

unsigned char   qc_check = 0xff;
// Variable for Quality Control condition
unsigned char   overwrite = 0;
// Variable for Quality Control condition
unsigned char   hour_start;
// Variable to handle difference in Hour format (0-2300 & 100-2400)
unsigned char   data_type = UNINITIALIZED;
// Variable to hold data type
unsigned char   data_delim = UNINITIALIZED;
// Variable to hold delimiter for data
unsigned char   max_records[NUM_OF_DATA_TYPES];
// Variables to hold maximum # of records
unsigned char   num_records[NUM_OF_FORMATS];
// Variable to hold the record size of each format.
unsigned char   data_present[NUM_OF_DATA_TYPES];
// Variables to check whether data type is declared
unsigned char   detected_format;
// Variable to hold the # of detected formats
unsigned int    last_day_data;
// Index variable to hold the last day for which data is found
unsigned char   file_present[NUM_OF_DAYS_IN_YEAR + 1];
// Variable to hold presence of files

unsigned int    work_quarterhour;
unsigned int    work_halfhour;
unsigned int    work_hour;
unsigned int    work_day;
unsigned int    work_month;
unsigned int    work_year;
unsigned int    work_julday;
unsigned int    work_type;
unsigned int    work_idx_data;
unsigned int    work_idx_fmt;
unsigned int    work_yr_idx;
unsigned int    max_data_pts;
```

```
unsigned char   num_hours;
unsigned char   num_day;
unsigned char   num_qc_options;
unsigned char   report_footer;
unsigned char   report_header;

unsigned int    now_hour;
unsigned int    now_day;
unsigned int    now_month;
unsigned int    now_year;
unsigned int    now_min;
// Group of Variables which are used for work.

const    char*  month_name[] = { "January",
                                 "Feburary",
                                 "March",
                                 "April",
                                 "May",
                                 "June",
                                 "July",
                                 "August",
                                 "September",
                                 "October",
                                 "November",
                                 "December" };

const    char*  short_month_name[] = { "JAN",
                                       "FEB",
                                       "MAR",
                                       "APR",
                                       "MAY",
                                       "JUN",
                                       "JUL",
                                       "AUG",
                                       "SEP",
                                       "OCT",
                                       "NOV",
```

```
                                       "DEC" };

unsigned char   pos_hour[NUM_OF_FORMATS];
unsigned char   pos_day[NUM_OF_FORMATS];
unsigned char   pos_month[NUM_OF_FORMATS];
unsigned char   pos_year[NUM_OF_FORMATS];
unsigned char   pos_julday[NUM_OF_FORMATS];
```

File: convert.c

```
// Main 'C' program of "CONVERT" program

// Standard header declarations
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <assert.h>

// Custom header declarations
#include "struct.h"
#include "errors.h"             // File contains Values for Exit Errors
#include "convert.h"              // File contains declaration of local
functions
#include "global.h"             // File contains Global Declarations
#include "common.h"               // File containes declares for Common
Functions

// Version declaration
const char *VERSION="3.21B";

int main ( int argc, char *argv[] )
{
   initialize_variables ( );  // Function to Initialize Variable
   put_software_info ( );     // Function to display software info.
   if ( argc < 2 )            // Check for insufficient arguments
   {
      printf ("\nInsufficient Arguments for execution.. Use -h option
for help\n" );
      printf ("Aborting....\n");
      exit ( INSUFFICIENT_ARGUMENTS );
   }

   load_environment ( argc, argv ); // Load Environment Variable
                                    // Command line switches
```

```
   load_configuration ( );          // Load Configuration file

   check_dirs ( );            // Check for existance of directories

   scan_files ( );            // Check for existance of files

   scan_data ( ); // Scan data for missing and undeclared data formats
   if ( ( quiet == NO ) && ( debug & REP_MISSING_MASK ) )
      printf ( "Scanning of Data was successful.\n" );

   if ( ( mode == AUTO ) || ( mainpage == YES ) )
   // Main Page is updated only in Auto Mode as Main Page Updation is a
part Daily Update
   {
      main_page_update ( );
   }

   if ( ( mode == AUTO ) || ( mode == ALLMONTHS ) || ( mode == SPECIFIC
) || ( mode == EXTERNAL ) )
   {
      month_page_update ( );        // Yearly page update
      calendar_page_update ( );     // Calendar page update
      create_monthly_html ( );// Daily summaries page update/create
      create_daily_html ( );  // Hourly summaries page update/create
   }

   if ( ( sort_data == YES ) && ( mode == AUTO ) )
      seperate_data ( );

   free_variables ( );        // Release allocated variables

   exit ( NORMAL );

}

void put_software_info ( void )
{
```

```
   printf ("Convert V%s - Data to HTML Converter Program\n", VERSION);
   printf ("***********************************************\n");
}

void put_version_info ( void )
{
   printf ("Convert V%s - Data to HTML Converter Program\n", VERSION);
   printf ("Data Sorter: V1.00A\n");
   printf ("QA & QC    : V1.00A\n");
   printf ("HTML COder : V1.00A\n");
   printf ("***********************************************\n");

}

// Function to put help information of the console
void put_help_info ( void )
{
   printf ("Convert V%s - Data to HTML Converter Program\n", VERSION);
   printf ("***********************************************\n");
   printf ("Command Syntax :\n");
   printf ("  convert [options] <file> [options] <file>....\n");
   printf ("     where:\n");
   printf ("        [options] are\n");
   printf ("        -config (-c) <file> : Configuration file\n");
   printf ("        -data (-d) <file>    : Data file (overrides
configuration)\n");
   printf ("        -auto               : Auto mode ( default )\n");
   printf ("        -manual             : Manual mode ( interactive
mode )\n" );
   printf ("        -help               : Print this help\n" );
   printf ("        -qc_report          : Perform QC Checks on data
only\n" );
   printf ("        -version            : Print Version information\n"
);
   printf ("        -month <value>      : Convert data for only <value>
month\n" );
   printf ("        -year <value>       : Convert all months data for
year <value>\n" );
```

```
   printf ("        -allmonths           : Convert data for all months
of the year\n");
   printf ("        -all                 : Convert data for all data in
the file into HTML\n");
   printf ("        -credits             : Credits information\n" );
   printf ("        -debug <options>     : Debugging Information\n");
   printf ("                                     <options> are all, qc,
config, missing and files\n" );
   printf ("        -no_qc               : By-pass Quality Control
Algorithmns\n");
   printf ("        -no_error            : No Error Logging\n" );
   printf ("        -no_mail             : Do not create Mail Log
file\n" );
   printf ("        -datadir <dir>       : Default Data Directory\n" );
   printf ("        -delim <#>           : Ascii # of the Delimiter in
Data\n" );
   printf ("        -format              : Format of Data (See README
for Details)\n" );
   printf ("        -type                : Type of Data ( See README for
Details)\n" );
   printf ("        -main_page           : Update Main page also\n" );
   printf ("        -overwrite <options>: Overwrite files that are
already created\n" );
   printf ("                              <options> are hourly, daily,
monthly, calendar and mainpage\n" );
   printf ("************************************************\n");
}

void initialize_variables ( void )
{
// Please global.h for more information about the variable functioning
   int            i;
   int            j;


   pass_error = ( char * ) calloc ( MAX_DATA_LENGTH, sizeof ( char ) );
   pass_mail = ( char * ) calloc ( MAX_DATA_LENGTH, sizeof ( char ) );
```

```
   data_struct = ( struct Data_Struct * ) malloc ( sizeof ( struct
Data_Struct ) * NUM_OF_FORMATS );
   html_struct = ( struct Data_Struct * ) malloc ( sizeof ( struct
Data_Struct ) * NUM_OF_DATA_TYPES );

   Data_table = ( struct Data_Present * ) calloc ( NUM_OF_DAYS_IN_YEAR
* NUM_OF_YEARS, sizeof ( struct Data_Present ) );

   ArrayID = ( struct ArrayID * ) malloc ( sizeof ( struct ArrayID ) *
NUM_OF_FORMATS );

   for ( i = 0 ; i < NUM_OF_FORMATS ; i++ )
   {
      data_struct[i].type = UNINITIALIZED;
      for ( j = 0 ; j < NUM_OF_PARAMETERS ; j++ )
      {
         data_struct[i].Parameter_name[j] = ( char * ) calloc (
SIZE_OF_PARAMETER, sizeof ( char ) );
         if ( data_struct[i].Parameter_name[j] == NULL )
         {
            printf ( "Could not allocate memory for Parameter Name
Arrays.\nAborting...\n" );
            exit ( FAILED_ALLOCATE_MEMORY );
         }
      }
   }
   for ( i = 0 ; i < NUM_OF_DATA_TYPES ; i++ )
   {
      for ( j = 0 ; j < NUM_OF_PARAMETERS ; j++ )
      {
         html_struct[i].Parameter_name[j] = ( char * ) calloc (
SIZE_OF_PARAMETER, sizeof ( char ) );
         if ( html_struct[i].Parameter_name[j] == NULL )
         {
            printf ( "Could not allocate memory for Parameter Name
Arrays.\nAborting...\n" );
            exit ( FAILED_ALLOCATE_MEMORY );
         }
```

```
      }
   }

}

void initialize_data_arrays ( void )
{
//  Please  refer  global.h  for  more  information  about  variable
functioning
   unsigned int   i = 0;
   unsigned int   j = 0;
   unsigned int   k = 0;
   time_t         xtime;
   struct tm*     time_construct;

   xtime = time ( NULL );
   time_construct = localtime ( &xtime );

   now_day = time_construct->tm_mday;
   now_month = time_construct->tm_mon;
   now_year = time_construct->tm_year;
   now_hour = time_construct->tm_hour;
   now_min = time_construct->tm_min;

   Data = ( struct Data_Format * ) malloc ( sizeof ( struct Data_Format
) * NUM_OF_DATA_TYPES );
   for ( i = 0 ; i < NUM_OF_DATA_TYPES ; i++ )
   {
      assert ( Data[i].record = ( struct rec_data * ) malloc ( sizeof (
struct rec_data ) * max_fields[i] ) );
      for ( j = 0 ; j <= max_fields[i] ; j++ )
      {
         assert  (  Data[i].record[j].parameter  =  (  float  *  )  malloc  (
sizeof ( float ) * max_records[i] ) );
         assert  (  Data[i].record[j].qc_status  =  (  char  *  )  malloc  (
sizeof ( char ) * max_records[i] ) );
/*     assert  (  memset  (  Data[i].record[j].parameter,  0xff,  sizeof  (
float ) * max_records[i] ) );*/
```

```
         for ( k = 0 ; k < max_records[i] ; k++ )
         {
            Data[i].record[j].parameter[k] = (float) 0;
            Data[i].record[j].qc_status[k] = QC_OKAY;
         }
      }

   }
}

void free_variables ( void )
{

   free ( data_struct );
   free ( html_struct );
   free ( pass_error );
   free ( pass_mail );
   free ( Data_table );
   free ( Data );
   free ( ArrayID );

}


void scan_files ( void )
{
   unsigned int   i = 0;
   unsigned int   k = 0;
   unsigned char  mask = 0;

   scan_stream ( work_file, month_html_generic );
   if ( ( access (work_file, F_OK) == NULL ) )
   {
      for ( k = 1 ; k < NUM_OF_DAYS_IN_YEAR ; k++ )
         file_present[k] |= FILE_MONTHLY_MASK;
   }
   for ( i = 0 ; i <= 11 ; i++ )
   {
```

```
      mask = 0;
      work_month = i;
      scan_stream ( work_file, calendar_html_generic );
      if ( ( access ( work_file, F_OK ) == NULL ) )
         mask |= FILE_CALENDAR_MASK;

      scan_stream ( work_file, daily_val_html_generic );
      if ( access ( work_file, F_OK ) == NULL )
         mask |= FILE_DAILY_MASK;

      for ( k = julday[i] + 1 ; k <= julday[i+1] ; k++ )
      {
         work_julday = k;
         work_day = k - julday[i];
         scan_stream ( work_file, hourly_val_html_generic );
         if ( access ( work_file, F_OK ) == NULL )
            file_present[k] |= mask | FILE_HOURLY_MASK;
         else
            file_present[k] |= mask;
      }
   }
}

void check_dirs ( void )
{
   unsigned int   i = 0;
   unsigned int   j = 0;
   unsigned char* bufferstr;
   time_t         xtime;
   struct tm*     time_construct;
   unsigned int   save_year;

   bufferstr = ( char * ) malloc ( MAX_DATA_LENGTH );
   save_year = work_year;

   xtime = time ( NULL );
   time_construct = localtime ( &xtime );
```

```
   for ( i = start_year ; i <= time_construct->tm_year + 1900 ; i++ )
   {
      work_year = i;
      scan_stream ( bufferstr, html_dir );
      if ( chdir ( bufferstr ) != NULL )
      {
         if ( ! quiet ) printf ( "Creating Directory %s\n", bufferstr
);
         if ( mkdir ( bufferstr, 0755 ) == -1 )
         {
            printf("Cannot either change directory or create directory
%s\nAborting...\n", bufferstr );
            exit ( FAILED_CHDIR );
         }
      }
      for ( j = 0 ; j <= 11 ; j++ )
      {
         work_month = j;
         scan_stream ( bufferstr, daily_val_dir );
         if ( chdir ( bufferstr ) != NULL )
         {
            if ( ! quiet ) printf ( "Creating Directory %s\n",
bufferstr );
            if ( mkdir ( bufferstr, 0775 ) == -1 )
            {
               printf("Cannot either change directory or create
directory %s\nAborting...\n", bufferstr );
               exit ( FAILED_CHDIR );
            }
         }
         scan_stream ( bufferstr, hourly_val_dir );
         if ( chdir ( bufferstr ) != NULL )
         {
            if ( ! quiet ) printf ( "Creating Directory %s\n",
bufferstr );
            if ( mkdir ( bufferstr, 0775 ) == -1 )
            {
```

```
                printf("Cannot   either   change   directory   or   create
directory %s\nAborting...\n", bufferstr );
                exit ( FAILED_CHDIR );
             }
          }



      }
   }
   work_year = save_year;
   free ( bufferstr );
}
```

# APPENDIX A2

## TYPICAL CONFIGURATION FILE

The typical configuration file used by the “CONVERT” program used to process data and generates HTML pages.

---

# Typical Configuration File

# Lines starting with either # or ; or any other non ascii characters are considered as a comment

# Note on Embedded variable:

# All embedded variables are declared by encapsulating the variable with ‘@@’

# The embedded variable STATION_ID is replaced with the value of the configuration variable STATION_ID declared in this file.

# The embedded variable YEAR is replaced with the value of the working year as determined by the “convert” program.

# The embedded variable MONTHNAME is replaced with the value of the name of the working month as determined by the “convert” program.

# The embedded variable SHORTMONTHNAME is replaced with the value of the first 3 characters of the month’s name as determined by the “convert” program. Eg. Jan, Feb

# The embedded variable MMDDYY is replaced with the value of the working month, working day and working year as determined by the “convert” program. Eg. 20$^{th}$ Jan 2000 would be replace by 01202000

# The embedded variable MMYY is replaced with the value of the working month and working year as determined by the “convert” program. Eg. Jan 2000 would be replace with a value 012000

**HTML_HEADER_FILE**=agnet/@@STATION_ID@@/conf/header.html.txt

# HTML_HEADER_FILE declaration is used for declaring the location of the file which contains the header information for all the HTML pages. The path declared is relative to configuration declaration “BASE_DIR”

**HTML_FOOTER_FILE**=agnet/@@STATION_ID@@/conf/footer.html.txt

# HTML_FOOTER_FILE declaration is used for declaring the location of the file which contains the footer information for all the HTML pages. The path declared is relative to configuration declaration “BASE_DIR”

**DATA_FILE**=/export/home/www/work/CAST_TWR.DAT

# DATA_FILE declaration is used the declare the location of the data file to be used by the convert program. The path declared is relative to configuration declaration “DATA_DIR”

**QC_FILE**=agnet/@@STATION_ID@@/conf/@@STATION_ID@@.qc

# QC_FILE declaration is used for declaring the location of the QC File. The path declared is relative to configuration declaration “BASE_DIR”

**ERROR_LOG_FILE**=agnet/@@STATION_ID@@/logs/error_@@STATION_ID@@.log

# ERROR_LOG_FILE is used to declare the location of the Error Log file. The errors and warnings reported on the console during execution are appended to this file. The logs are generated on the basis of command line switch “-log” or configuration declaration “ERROR_LOG”. The path declared is relative to configuration declaration “BASE_DIR”

**MAIL_LOG_FILE**=agnet/@@STATION_ID@@/logs/mail_@@STATION_ID@@.log

# MAIL_LOG_FILE is used to declare the location of the Mail Log file. The errors and warnings reported on the console during execution are copied to this file. A new file is created for each execution. The path declared is relative to configuration declaration “BASE_DIR”

**ERROR_LOG**=ALL

# ERROR_LOG declaration is used to control the parameters that have to be logged to the error file. Different parameters that can be logged are files,qc,missing,config

# “files” option logs all activities related to file such as file creation

# “qc” option logs all QC related activities such as QC failures and warnings

# “missing” option logs all missing data in data file

# “config” option logs all the declarations and activities of the configuration declaration

**MAIL_LOG**=ALL

# MAIL_LOG declaration is used to control the parameters that have to be logged to the mail log file. Different parameters that can be logged are files,qc,missing,config

**MAIN_HTML**=agnet/@@STATION_ID@@/index.html

# MAIN_HTML declaration is used to declare the location of the Main HTML page. The path declared is relative to configuration declaration “BASE_DIR”

**MONTH_HTML**=agnet/@@STATION_ID@@/@@YEAR@@/@@STATION_ID@@_@@YEAR@@.html

# MONTH_HTML declaration is used to declare the location of the Yearly HTML pages. The path declared is relative the configuration declaration ”BASE_DIR”

**CALENDAR_HTML**=agnet/@@STATION_ID@@/@@YEAR@@/@@MONTHNAME@@/@@STATION_ID@@_@@SHORTMONTHNAME@@.html

# CALENDAR_HTML declaration is used to declare the location of the Calendar HTML Page. The path declared is relative to the configuration declaration “BASE_DIR”

**HOURLY_VALUES_HTML**=agnet/@@STATION_ID@@/@@YEAR@@/@@MONTHNAME@@/@@STATION_ID@@_@@MMDDYY@@.html

# HOURLY_VALUES_HTML declaration is used to declare the location of the Hourly Values (Summaries) HTML Page. The path declared is relative to the configuration declaration “BASE_DIR”

**DAILY_VALUES_HTML**=agnet/@@STATION_ID@@/@@YEAR@@/@@MONTHNAME@@/@@STATION_ID@@_@@MMYY@@.html

# DAILY_VALUES_HTML declaration is used to declare the location of the Daily Values (Summaries) HTML Page. The path declared is relative to the configuration declaration “BASE_DIR”

**STATISTICS_HTML**=agnet/@@STATION_ID@@/@@YEAR@@/@@MONTHNAME@@/@@STATION_ID@@_S_@@MMYY@@.html

# STATISTICS_HTML declaration is used to declare the location of the Statistics HTML Page. The path declared is relative to the configuration declaration “BASE_DIR”. This function is not yet implemented

**SORT_DATA**=YES

# SORT_DATA declaration is used to sort the data before creating HTML Pages. This configuration declaration is used when the data is not chronological order. This function is not yet implemented.

**DAILY_DATA_FILE**=@@STATION_ID@@@@MMYY@@.dat

# DAILY_DATA_FILE declaration is used to declare the name of the data file to be used for processing. The path to the file is determined by the configuration declaration “DATA_DIR”. This declaration can be overridden with command line switch “-data”

**HOURLY_DATA_FILE**=@@STATION_ID@@@@MMYY@@.dat

# HOURLY_DATA_FILE declaration is used to declare the name of the data file to be used for processing. The path to the file is determined by the configuration declaration “DATA_DIR”. This declaration can be overridder with command line switch “-data”

**QC_CHECKS**=YES

# QC_CHECKS declaration is used to control if QC Checks have to be performed on the data. If declared as "YES", then QC_FILE should also be declared.

**TRIPOD_LINK**=/agnet/cast

# TRIPOD_LINK declaration is used to declare the location of the Old tripod link. This declaration is used only for upgraded tower sites that had a tripod.

**DATA_DIR**=/export/home/www/agnet/data

# DATA_DIR is used to declare the path of the data file

**BASE_DIR**=/export/home/www

# BASE_DIR is used to declare the relative path where either file can found or created.

**HTML_DIR**=/export/home/www/agnet/@@STATION_ID@@/@@YEAR@@

# HTML_DIR is used to declare the relative path where the HTML Files like Yearly page and monthly directories can either be found or created.

**DAILY_VAL_DIR**=/export/home/www/agnet/@@STATION_ID@@/@@YEAR@@/@@MONTHNAME@@

# DAILY_VAL_DIR is used to declare the location of the Daily Values (Summaries) page can either be found or created.

**HOURLY_VAL_DIR**=/export/home/www/agnet/@@STATION_ID@@/@@YEAR@@/@@MONTHNAME@@

# HOURLY_VAL_DIR is used to declare the location of the Hourly Values (Summaries) page can either be found or created.

**STATION_NAME**=Horticultural Crops Research Station

**STATION_CITY**=Castle Hayne

**STATION_ADDRESS1**=3800 Castle Hayne Road

**STATION_ZIPCODE**=28429-6519

**STATION_COUNTY**=New Hanover

**STATION_LAT**=34.32 Degrees North

**STATION_LONG**=77.92 Degrees West

**STATION_ELEVATION**=40 feet

**STATION_DAS**=Campbell CSI CR-10X

# Configuration declarations STATION_NAME, STATION_CITY, STATION_ADDRESS1, STATION_ADDRESS2 and STATION_ZIPCODE are used to declare the address of the station. Configuration declarations such as STATION_COUNTY, STATION_LAT (Station Latitude), STATION_LONG (Station Longitude), STATION_ELEVATION (Station Elevation) and STATION_DAT (Data Acq. System) are additional information. This information is published in the Main HTML page along side station information.

**STATION_ID**=cast_twr

# STATION_ID configuration declaration is a unique id for each station. The declaration format for STATION_ID followed at State Climate Office is the first four letters of the STATION_ID are first four letters of the STATION_CITY. The last four letters are "_TWR" for tower installation else blank for other installations.

# Following declarations indicate the variety of sensors present on the tower.

**SENSOR_AIRTEMP_L1**=CSI HMP45C Vaisala

# Level 1 Air Temperature

**SENSOR_AIRTEMP_L2**=No second Level

# Level 2 Air Temperature

**SENSOR_RELHUM_L1**=CSI HMP45C Vaisala

# Level 1 Relative Humidity

**SENSOR_RELHUM_L2**=No second level

# Level 2 Relative Humidity

**SENSOR_PRECIP**=Texas Electronics TE525WS - Tipping Bucket Rain Guage

# Precipitation

**SENSOR_WINDSPD**=RM Young 05103 Wind Monitor - Gill Propeller

# Wind Speed

**SENSOR_WINDDIR**=RM Young 05103 Wind Monitor – Balanced Vane

# Wind Direction

**SENSOR_PRESSURE**=CSI CS-105 Pressure Transducer

# Barometric Pressure

**SENSOR_SOILTEMP**=CSI 107B Water/Soil Probe – Thermocouple

# Soil Temperature

**SENSOR_SOLARRAD**=Li-Cor LI200X Silicon Pyranometer

# Solar Radiation

**SENSOR_PAR**=Li-Cor LI190SB Quantum Sensor

# Photosynthetically Active Radiation

**SENSOR_SOILMOIST**=CSI CS-615 Water Content Reflectometer

# Soil Moisture

**SENSOR_OTHERS**=None

# Other possible sensors

# Following declarations are used to declare the siting of the instrument

**LOCATION_AIRTEMP_L1**=2 meters, Covered by gill shield

**LOCATION_RELHUM_L1**=2 meters, Covered by gill shield

**LOCATION_AIRTEMP_L2**=No second Level

**LOCATION_RELHUM_L2**=No second level

**LOCATION_PRECIP**=4 feet

**LOCATION_WINDSPD**=10 meters; On the top of the tower

**LOCATION_WINDDIR**=10 meters; On the top of the tower

**LOCATION_PRESSURE**=Inside a weatherproof cabinet at 1.5 meters.

**LOCATION_SOILTEMP**=8 Inches in ground

**LOCATION_SOILMOIST**=8 Inches in ground

**LOCATION_SOLARRAD**=2 meters; on a pyranometer arm

**LOCATION_PAR**=2 meters; on a pyranometer arm

**LOCATION_OTHERS**=None installed

**START_MONTH**=8

**START_YEAR**=1999

# START_MONTH and START_YEAR declarations are used to declare the month and year from which the data collection was started

**NOAA_ID**=9467

# NOAA_ID is used the declare the Unique NOAA ID for each site.

**QC_WARN_COLOR**=FF0000h

**QC_ERROR_COLOR**=CC00CCh

# QC_WARN_COLOR and QC_ERROR_COLOR are used to flag QC Warning and Errors in the page. This function is not implemented.

**QUARTER_HOURLY_DATA**=csv_w_id,1,Year,JulianDay,Hour,SolarRadTotal,PARTotal,RelHumInst,PressureInst,PrecipTotal,WindSpeed15Min,WindDir15Min,StdDevWS15Min,AirTempCInst,SoilTempCInst,SoilMoistInst

**HOURLY_DATA_FORMAT**=csv_w_id,2,Year,JulianDay,Hour,PressureInst,PrecipTotal,AirTempCInst,RelHumInst,PARAvg,SolarRadAvg,SoilTempCInst,SoilMoistInst,WindSpeed15Min,WindDir15Min,StdDevWS15Min

**DAILY_DATA_FORMAT**=csv_w_id,3,Year,JulianDay,Hour,BatteryMin,PressureMax,PressureMin,PrecipTotal,WindSpeedMax,AirTempCMax,AirTempCMin,AirTempCAvg,RelHumMax,RelHumMin,PARMax,PARMin,PARAvg,SoilTempCMax,SoilTempCMin,SoilTempCAvg,SoilMoistMax,SoilMoistMin,SoilMoistAvg,BatteryAvg

# QUARTER_HOURLY_DATA, HOURLY_DATA_FORMAT and DAILY_DATA_FORMAT are used to declare the data format for the hourly and daily data. The first field indicates the data type. Description of each field are as follows:

# First field indicates data type. Different data options are csv_w_id (Comma Seperated with Array ID), csv_wo_id (Comma seperated without Array ID), space ( Space

seperated) and other ( other data types ). When “other” data types are declare, the delimiter should also be declared.

# If the data type has an array id, the second field should be used to declare the array ID.

# After declaring the data type, each parameter should be declared based on the data format with comma seperation. The last 3 to 5 characters of each parameter indicate whether data is sampled (inst), averaged (avg), Cumulated (total), Maximum summary (max) or Minimum summary (min). If temperature measurement is done in centigrade and average over the hour, the parameter should be declared as “AirTempCavg”, for Fahrenheit measurement the parameter should be declated as “AirTempFavg”.

**DAILY_HTML_FORMAT**=@JulianDay,@Month,@link_hourly_values,AirTempFMax,AirTempFMin,AirTempFAvg,SoilTempFMax,SoilTempFMin,SoilTempFAvg,SoilMoistAvg,PrecipTotal,PressureMax,PressureMin,RelHumMax,RelHumMin,WindSpeedMax,PARMax,PARAvg

**HOURLY_HTML_FORMAT**=@JulianDay,@Month,@Day,@Year,@Hour,AirTempFInst,SoilTempFInst,RelHumInst,PrecipTotal,PressureInst,WindSpeed15Min,WindDir15Min,StdDevWS15Min,SolarRadAvg,PARAvg,SoilMoistInst

# HOURLY_HTML_FORMAT and DAILY_HTML_FORMAT are used to declare the output (HTML) format for the hourly and daily data. Embedded variable can be used output working parameters like Julianday, Month, Day, Year and Hour.

# Each parameter should be declared based on the data format with comma seperation. The last 3 to 5 characters of each parameter indicate whether data is sampled (inst), averaged (avg), Cumulated (total), Maximum summary (max) or Minimum summary (min). If temperature measurement is done in centigrade and average over the hour, the parameter should be declared as “AirTempCavg”, for Fahrenheit measurement the parameter should be declated as “AirTempFavg”.

# Following configuration variable are used to create HTML tags.

# Embedded variables LINK_MONTH_FILE_OF_YEAR is replaced with the link to the Yearly pages. This HTML tag is used in the Main page.

# Embedded variable LINK_CALENDAR_FILE_OF_MONTH is replaced with the link to the Calendar page. This HTML tag is used in Yearly page.

# Embedded variable LINK_HOURLY_VALUES is replaced with the link to the Hourly summary page and is found in Calendar Page.

# Embedded variable PARAMETER_VALUE is replace with the value of the weather parameter and is used in Hourly and Daily summaries page.

**MAIN_PAGE_HTML_FORMAT**=<tr align=center><td>
@@LINK_MONTHLY_FILE_OF_YEAR@@</td></tr>

**MONTH_PAGE_HTML_FORMAT**=<tr align=center><td>
@@LINK_CALENDAR_FILE_OF_MONTH@@</td></tr>

**CALENDAR_PAGE_HTML_FORMAT**=<td>@@LINK_HOURLY_VALUES@@
</td>

**DUMMY_MONTH_HTML_FORMAT**=<tr align=center><td>
@@MONTHNAME@@,@@YEAR@@</td></tr>

**DUMMY_CALENDAR_HTML_FORMAT**=<td>@@DAY_OF_MONTH@@</td>

**SPACE_CALENDAR_HTML_FORMAT**=<td> </td>

**HOURLY_VALUES_HTML_FORMAT**=<td>@@PARAMETER_VALUE@@</td>

**DAILY_VALUES_HTML_FORMAT**=<td>@@PARAMETER_VALUE@@</td>

# APPENDIX A3

## SAMPLE HTML PAGES

This section contains typical HTML files generated using the “convert” program. The pages generated are in hierarchical order which is as shown below

- Main Page
- Yearly Page
- Calendar Page
- Hourly Summaries Page
- Daily Summaries Page

---

Horticultural
Crops Research
Station
2450 Faison Highway
Clinton, NC 28328-9501

# NCARS Weather & Climate Network

Station Data
Station Description
Station Geography
Sensor Descriptions
Sensor Locations
Station Location

## Station Data

**Click on the Year for which you want data**

| Year |
|---|
| 2000 |

Old data (Tripod) is available here

## Station Description

| | |
|---|---|
| **County** | Sampson |
| **NCDC/NOAA ID number** | 1881 |
| **Latitude** | 35.02 Degrees North |
| **Longitude** | 78.28 Degrees West |
| **Elevation** | 158 feet |
| **Data Acquisition System** | Campbell CSI CR-10X |

## Station Geography

No images are available at this point of time. Please check back later.

## Sensor Description

| Sensors | Manufacturer |
|---|---|
| Air temperature at Level 1 | CSI HMP45C Vaisala |
| Air temperature at Level 2 | No second Level |
| Relative Humidity at Level 1 | CSI HMP45C Vaisala |
| Relative Humidity at Level 2 | No second level |
| Precipitation | Texas Electronics TE525WS - Tipping Bucket Rain Guage |
| Wind Speed | RM Young 05103 Wind Monitor - Gill Propeller |
| Wind Direction | RM Young 05103 Wind Monitor |
| Barometric Pressure | CSI CS-105 Pressure Transducer |
| Soil temperature | CSI 107B Water/Soil Probe - Thermocouple |
| Solar radiation | Li-Cor LI200X Silicon Pyranometer |
| Photosynthetically active radiation | Li-Cor LI190SB Quantum Sensor |
| Other Sensors | None |

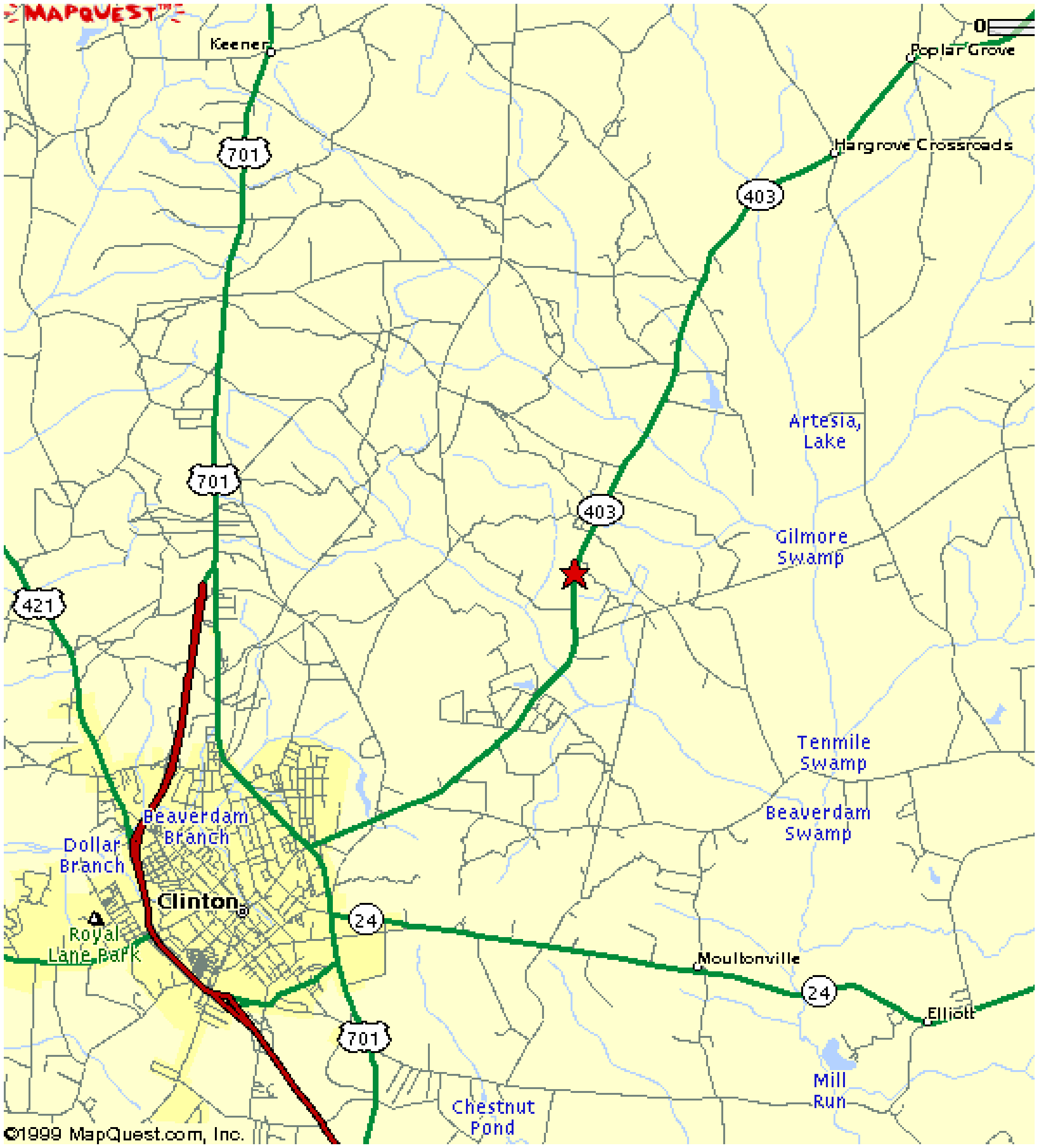
CSI stands for Campbell Scientific Inc.

## Sensor Locations

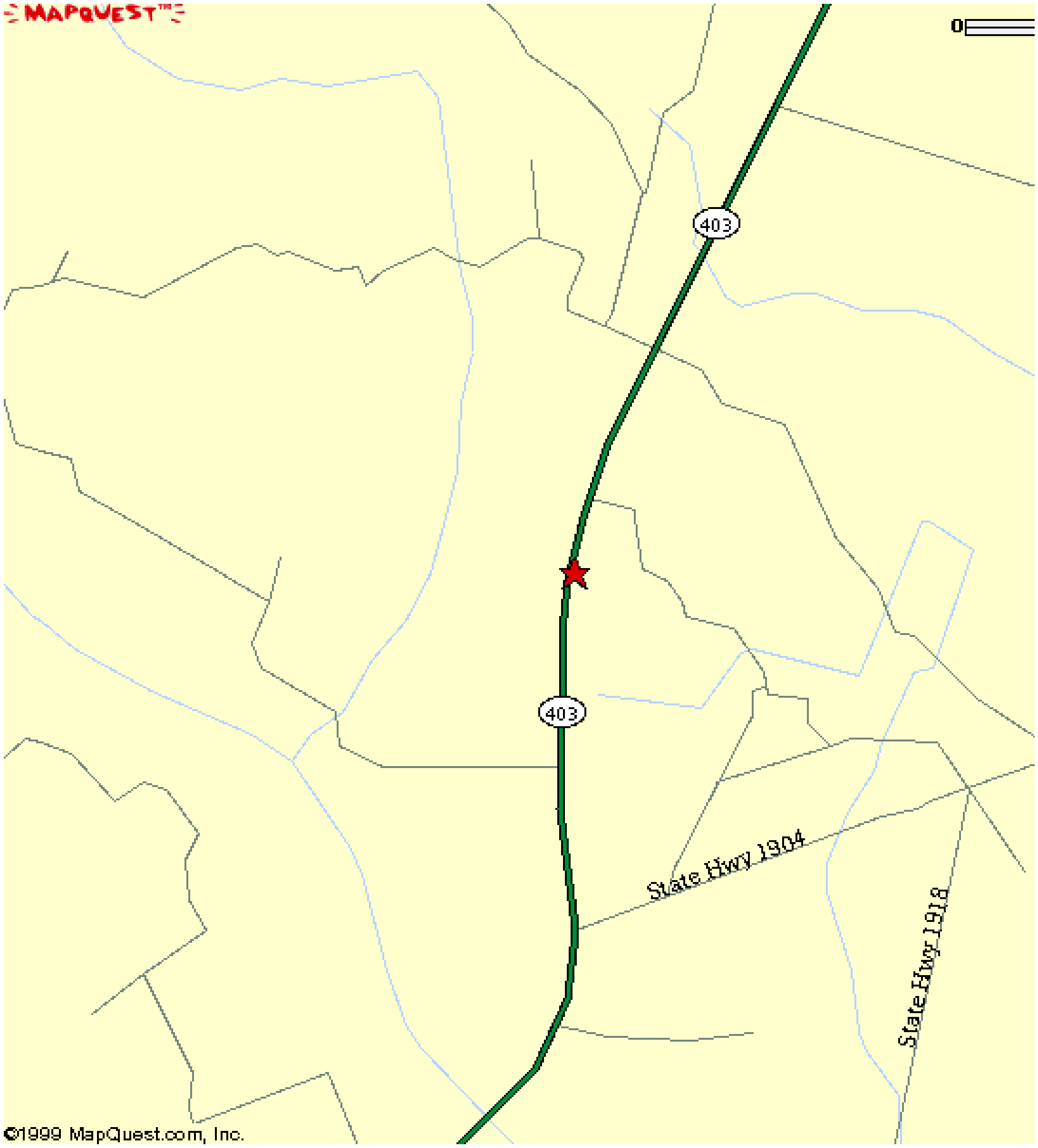
| **Sensor** | **Physical Location** |
| --- | --- |
| Air temperature at Level 1 | 2 meters, Covered by gill shield |
| Air temperature at Level 2 | No second Level |
| Relative Humidity at Level 1 | 2 meters, Covered by gill shield |
| Relative Humidity at Level 2 | No second level |
| Precipitation | 4 feet |
| Wind Speed | 10 meters; On the top of the tower |
| Wind Direction | 10 meters; On the top of the tower |
| Barometric Pressure | Inside a weatherproof cabinet at 1.5 meters. |
| Soil temperature | 8 Inches in ground |
| Solar radiation | 2 meters; on a pyranometer arm |
| Photosynthetically active radiation | 2 meters; on a pyranometer arm |
| Other Sensors | None installed |

## Station Location

Map below indicates the location of the station.




A Closer view of the station.




These maps were generated by Mapquest Inc..

This page is maintained and updated by staff of State Climate Office. If you have any questions or comments regarding our Web Site or any page in this web site, please send a mail to sco@cumulus.meas.ncsu.edu.

This page was updated on 8,May 2000 at 6:31 hrs

**Horticultural Crops Research Station**
**2450 Faison Highway**
**Clinton, NC 28328-9501**

**NCARS Weather & Climate Network**

**2000**

Click on the Month below for which you want data.

| |
|---|
| January,2000 |
| Feburary,2000 |
| March,2000 |
| April,2000 |
| May,2000 |
| June,2000 |
| July,2000 |
| August,2000 |
| September,2000 |
| October,2000 |
| November,2000 |
| December,2000 |

This page is maintained and updated by staff of State Climate Office. If you have any questions or comments regarding our Web Site or any page in this web site, please send a mail to sco@cumulus.meas.ncsu.edu.

This page was updated on 7,May 2000 at 6:31 hrs

**Horticultural Crops Research Station**
**2450 Faison Highway**
**Clinton, NC 28328-9501**

NCARS Weather & Climate Network

**May 2000**

**Select the date for which you want hourly values.**

May,2000

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | | | |

[April,2000] [Monthly Summary of May] [June,2000]

This page was updated on 7,May 2000 at 6:31 hrs

**Horticultural Crops Research Station**
**2450 Faison Highway**
**Clinton, NC 28328-9501**

**NCARS Weather & Climate Network**

April 2000

**Daily values for April, 2000 Click on the day for hourly values for that day**

| Julian Day | Month | Day | Maximum Air Temp. | Minimum Air Temp. | Avg. Air Temp. | Maximum Soil Temp. | Minimum Soil Temp. | Avg. Soil Temp. | Avg. Soil Moisture | Total Precipitation | Maximum Pressure | Minimum Pressure | Maximum RH | Minimum RH | Maximum Wind Speed | Maximum PAR | Avg. PAR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (°F) | (°F) | (°F) | (°F) | (°F) | (°F) | $m^3/m^3$ | (in) | (mb) | ($m.s^{-1}$) | (%) | (%) | ($m.s^{-1}$) | ($mole.m^{-2}.day^{-1}$) | ($mole.m^{-2}.day^{-1}$) |
| 92 | 04 | 01 | 88.6 | 31.0 | 61.4 | 76.1 | 48.6 | 60.9 | 0.24 | 0.00 | 1022.0 | 1017.0 | 93 | 19 | 7.0 | 124.2 | 37.5 |
| 93 | 04 | 02 | 90.7 | 37.5 | 65.9 | 73.5 | 54.4 | 63.3 | 0.23 | 0.00 | 1019.0 | 1013.0 | 90 | 35 | 9.3 | 139.7 | 27.1 |
| 94 | 04 | 03 | 97.3 | 49.7 | 75.1 | 78.9 | 62.4 | 68.8 | 0.23 | 0.00 | 1014.0 | 1006.0 | 80 | 43 | 10.1 | 137.7 | 27.6 |
| 95 | 04 | 04 | 83.3 | 39.7 | 69.5 | 67.0 | 57.3 | 64.9 | 0.23 | 0.04 | 1007.0 | 999.0 | 88 | 22 | 10.6 | 63.5 | 11.2 |
| 96 | 04 | 05 | 75.8 | 28.8 | 57.7 | 71.5 | 48.8 | 58.9 | 0.23 | 0.00 | 1014.0 | 1007.0 | 56 | 17 | 9.5 | 122.0 | 38.8 |
| 97 | 04 | 06 | 95.0 | 33.9 | 73.1 | 75.7 | 50.4 | 62.1 | 0.23 | 0.00 | 1012.0 | 1004.0 | 69 | 22 | 11.0 | 122.0 | 38.7 |
| 98 | 04 | 07 | 98.8 | 39.7 | 78.0 | 80.9 | 56.5 | 67.1 | 0.22 | 0.00 | 1012.0 | 1008.0 | 76 | 28 | 9.1 | 122.9 | 39.0 |
| 99 | 04 | 08 | 95.1 | 34.4 | 65.4 | 79.2 | 54.4 | 65.4 | 0.22 | 0.68 | 1009.0 | 994.0 | 90 | 29 | 13.6 | 131.0 | 35.0 |
| 100 | 04 | 09 | 74.0 | 30.2 | 55.8 | 66.1 | 45.5 | 54.3 | 0.24 | 0.00 | 1016.0 | 1005.0 | 60 | 19 | 10.5 | 124.8 | 40.3 |
| 101 | 04 | 10 | 90.2 | 26.9 | 67.1 | 76.3 | 46.0 | 59.4 | 0.23 | 0.00 | 1020.0 | 1016.0 | 90 | 15 | 5.9 | 122.7 | 39.6 |
| 102 | 04 | 11 | 92.0 | 39.3 | 70.6 | 77.7 | 55.9 | 64.8 | 0.23 | 0.00 | 1022.0 | 1016.0 | 84 | 29 | 9.4 | 135.0 | 33.8 |
| 103 | 04 | 12 | 94.9 | 45.5 | 74.5 | 79.2 | 60.3 | 68.1 | 0.23 | 0.12 | 1021.0 | 1015.0 | 88 | 37 | 9.1 | 159.8 | 25.6 |
| 104 | 04 | 13 | 64.5 | 33.0 | 42.2 | 62.3 | 49.8 | 53.4 | 0.24 | 0.39 | 1025.0 | 1021.0 | 94 | 67 | 8.6 | 31.2 | 4.8 |
| 105 | 04 | 14 | 75.4 | 33.9 | 53.5 | 59.5 | 49.7 | 54.6 | 0.24 | 0.02 | 1021.0 | 1014.0 | 96 | 74 | 7.4 | 65.6 | 10.7 |
| 106 | 04 | 15 | 86.6 | 48.0 | 70.5 | 71.4 | 58.0 | 63.1 | 0.24 | 0.39 | 1016.0 | 1008.0 | 95 | 68 | 4.4 | 119.7 | 14.1 |
| 107 | 04 | 16 | 94.9 | 50.1 | 75.8 | 75.9 | 62.2 | 67.2 | 0.24 | 0.03 | 1013.0 | 1009.0 | 94 | 45 | 5.2 | 168.5 | 26.3 |
| 108 | 04 | 17 | 97.0 | 47.3 | 76.0 | 81.2 | 59.5 | 68.4 | 0.24 | 0.00 | 1010.0 | 1001.0 | 95 | 35 | 7.8 | 146.3 | 33.1 |
| 109 | 04 | 18 | 79.4 | 41.4 | 51.5 | 65.8 | 53.8 | 59.6 | 0.25 | 0.77 | 1006.0 | 998.0 | 96 | 65 | 10.6 | 56.5 | 9.1 |
| 110 | 04 | 19 | 84.0 | 39.6 | 56.1 | 68.0 | 52.0 | 58.2 | 0.25 | 0.00 | 1012.0 | 1006.0 | 91 | 48 | 8.5 | 175.2 | 29.4 |
| 111 | 04 | 20 | 92.0 | 36.0 | 68.9 | 79.4 | 52.6 | 64.7 | 0.25 | 0.00 | 1015.0 | 1006.0 | 96 | 42 | 7.8 | 128.0 | 39.6 |
| 112 | 04 | 21 | 95.8 | 50.2 | 79.2 | 80.2 | 60.5 | 68.6 | 0.24 | 0.00 | 1006.0 | 994.0 | 89 | 34 | 9.5 | 160.4 | 33.0 |
| 113 | 04 | 22 | 78.1 | 37.7 | 61.4 | 74.4 | 57.5 | 65.4 | 0.24 | 0.00 | 1003.0 | 996.0 | 72 | 35 | 7.3 | 176.6 | 29.2 |
| 114 | 04 | 23 | 85.4 | 38.2 | 62.1 | 79.6 | 55.2 | 66.2 | 0.24 | 0.00 | 1005.0 | 1002.0 | 83 | 32 | 7.0 | 130.7 | 38.9 |
| 115 | 04 | 24 | 86.0 | 38.4 | 58.8 | 75.8 | 58.7 | 65.1 | 0.23 | 0.21 | 1008.0 | 998.0 | 92 | 43 | 6.5 | 149.6 | 22.7 |
| 116 | 04 | 25 | 71.3 | 42.1 | 48.3 | 61.5 | 54.8 | 59.3 | 0.24 | 0.62 | 1007.0 | 996.0 | 96 | 74 | 8.9 | 68.5 | 7.0 |
| 117 | 04 | 26 | 79.8 | 34.4 | 56.6 | 73.8 | 52.2 | 61.0 | 0.25 | 0.00 | 1011.0 | 1007.0 | 91 | 39 | 5.0 | 140.1 | 36.8 |
| 118 | 04 | 27 | 83.0 | 35.7 | 55.3 | 75.5 | 55.0 | 61.7 | 0.24 | 0.00 | 1011.0 | 1005.0 | 93 | 44 | 5.5 | 149.6 | 27.7 |
| 119 | 04 | 28 | 74.8 | 37.4 | 52.8 | 64.9 | 55.2 | 58.4 | 0.24 | 0.27 | 1007.0 | 1000.0 | 96 | 78 | 6.6 | 156.0 | 7.5 |
| 120 | 04 | 29 | 86.1 | 36.8 | 59.9 | 81.3 | 52.4 | 63.5 | 0.24 | 0.27 | 1012.0 | 1005.0 | 93 | 40 | 12.2 | 136.6 | 36.8 |
| 121 | 04 | 30 | 91.9 | 36.6 | 66.7 | 83.2 | 51.6 | 65.0 | 0.24 | 0.01 | 1014.0 | 1011.0 | 97 | 27 | 6.5 | 126.9 | 41.5 |

[March,2000] [May,2000]

These data are preliminary.
Values such as 99999.0 or 99999 indicates that these data values have failed either instrument range or noise check.
Please consult the staff at State Climate Office for further details.

Data used should be cited in the following way:
"Information was provided by State Climate Office of North Carolina at NC State University"

This page was updated on 7,May 2000 at 6:31 hrs

**Horticultural Crops Research Station**
**2450 Faison Highway**
**Clinton, NC 28328-9501**

NCARS Weather & Climate Network

30 April 2000

Hourly values for 30, April 2000

| Julian Day | Month | Day | Year | Hours | Air Temp. | Soil Temp. | Relative Humidity | Precipitation | Barometric Pressure | Wind Speed | Wind Direction | Std. Dev. of Wind Dir. | Total Radiation | Photosynthetically Active Radiation | Soil Moisture |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MM | DD | YY | HHMM | (°F) | (°F) | (%) | (in) | (mb) | ($m.s^{-1}$) | (deg) | | ($W.m^{-2}$) | ($mole.m^{-2}.hour^{-1}$) | $m^3/m^3$ |
| 121 | 04 | 30 | 2000 | 100 | 62.1 | 55.3 | 95 | 0.00 | 1012.0 | 0.5 | 264 | 15.8 | 1.09 | 0.4 | 0.25 |
| 121 | 04 | 30 | 2000 | 200 | 60.2 | 54.3 | 96 | 0.00 | 1011.0 | 0.8 | 294 | 45.0 | 0.77 | 0.4 | 0.25 |
| 121 | 04 | 30 | 2000 | 300 | 47.6 | 53.7 | 89 | 0.01 | 1012.0 | 1.3 | 279 | 15.8 | 0.66 | 0.4 | 0.25 |
| 121 | 04 | 30 | 2000 | 400 | 45.2 | 52.7 | 89 | 0.00 | 1012.0 | 0.3 | 286 | 8.9 | 0.47 | 0.4 | 0.25 |
| 121 | 04 | 30 | 2000 | 500 | 45.2 | 52.0 | 91 | 0.00 | 1012.0 | 0.8 | 274 | 21.6 | 0.41 | 0.5 | 0.25 |
| 121 | 04 | 30 | 2000 | 600 | 48.0 | 52.1 | 89 | 0.00 | 1013.0 | 1.4 | 289 | 10.2 | 10.45 | 2.4 | 0.25 |
| 121 | 04 | 30 | 2000 | 700 | 43.7 | 56.5 | 85 | 0.00 | 1014.0 | 1.0 | 298 | 16.2 | 96.90 | 19.0 | 0.25 |
| 121 | 04 | 30 | 2000 | 800 | 46.3 | 63.2 | 74 | 0.00 | 1014.0 | 1.4 | 297 | 12.0 | 224.30 | 45.4 | 0.25 |
| 121 | 04 | 30 | 2000 | 900 | 79.0 | 70.8 | 60 | 0.00 | 1014.0 | 1.2 | 278 | 16.3 | 354.10 | 75.3 | 0.25 |
| 121 | 04 | 30 | 2000 | 1000 | 82.1 | 75.6 | 51 | 0.00 | 1014.0 | 2.2 | 286 | 20.7 | 464.10 | 100.1 | 0.24 |
| 121 | 04 | 30 | 2000 | 1100 | 84.1 | 78.9 | 46 | 0.00 | 1014.0 | 2.9 | 274 | 28.3 | 541.60 | 115.3 | 0.24 |
| 121 | 04 | 30 | 2000 | 1200 | 87.1 | 81.6 | 42 | 0.00 | 1013.0 | 2.7 | 291 | 31.0 | 591.90 | 124.2 | 0.24 |
| 121 | 04 | 30 | 2000 | 1300 | 89.7 | 82.6 | 37 | 0.00 | 1013.0 | 2.6 | 302 | 38.4 | 602.10 | 125.1 | 0.24 |
| 121 | 04 | 30 | 2000 | 1400 | 91.3 | 81.9 | 30 | 0.00 | 1012.0 | 3.2 | 321 | 24.2 | 572.10 | 117.1 | 0.24 |
| 121 | 04 | 30 | 2000 | 1500 | 90.6 | 74.7 | 33 | 0.00 | 1012.0 | 4.3 | 319 | 25.5 | 504.30 | 102.1 | 0.24 |
| 121 | 04 | 30 | 2000 | 1600 | 90.8 | 74.0 | 29 | 0.00 | 1011.0 | 3.9 | 336 | 21.9 | 394.10 | 78.2 | 0.24 |
| 121 | 04 | 30 | 2000 | 1700 | 91.0 | 71.9 | 30 | 0.00 | 1011.0 | 3.2 | 318 | 23.3 | 280.10 | 54.6 | 0.24 |
| 121 | 04 | 30 | 2000 | 1800 | 89.4 | 68.7 | 31 | 0.00 | 1012.0 | 1.9 | 328 | 16.1 | 145.30 | 27.2 | 0.24 |
| 121 | 04 | 30 | 2000 | 1900 | 77.1 | 65.1 | 69 | 0.00 | 1012.0 | 1.4 | 337 | 8.1 | 34.53 | 6.4 | 0.24 |
| 121 | 04 | 30 | 2000 | 2000 | 51.8 | 62.4 | 74 | 0.00 | 1013.0 | 0.2 | 24 | 34.9 | 0.22 | 0.5 | 0.24 |
| 121 | 04 | 30 | 2000 | 2100 | 46.2 | 60.4 | 81 | 0.00 | 1014.0 | 0.1 | 212 | 8.8 | 0.00 | 0.4 | 0.24 |
| 121 | 04 | 30 | 2000 | 2200 | 42.1 | 58.9 | 82 | 0.00 | 1014.0 | 0.0 | 0 | 0.0 | 0.14 | 0.5 | 0.24 |
| 121 | 04 | 30 | 2000 | 2300 | 48.6 | 57.8 | 86 | 0.00 | 1014.0 | 0.2 | 67 | 6.4 | 0.83 | 0.4 | 0.24 |
| 121 | 04 | 30 | 2000 | 2400 | 46.7 | 56.8 | 86 | 0.00 | 1014.0 | 0.5 | 167 | 53.9 | 1.07 | 0.4 | 0.24 |

[Previous Day (29 April)] [Monthly Summary of April] [Next Day(1 May)]

Back to Station Map

These data are preliminary.
Values such as 99999.0 or 99999 indicates that these data values have failed either instrument range or noise check.
Please consult the staff at State Climate Office for further details.
Data used should be cited in the following way:
"Information was provided by State Climate Office of North Carolina at NC State University"
This page was updated on 2,May 2000 at 6:31 hrs

# APPENDIX B1

## SAMPLE DATALOGGER PROGRAM

This section contains the sample datalogger program used in the AgNet stations. The program was written using Campbell Scientific's EDLOG software which is a part of the PC 208W V2.00 suite.

——————————————File: lews_twr.csi——————————————

```
;{CR10}
*Table 1 Program
  01: 60.0000   Execution Interval (seconds)



; Enable Pressure sensor
1:  Do (P86)
 1: 42       Set Port 2 High
; Enable Temperature Sensor
2:  Do (P86)
 1: 41       Set Port 1 High
; Enable Soil Moisture sensor
3:  Do (P86)
 1: 43       Set Port 3 High
; Measure Air Temperature
4:  Volts (SE) (P1)
 1: 1        Reps
 2: 5        2500 mV Slow Range
 3: 1        SE Channel
 4: 1        Loc [ AirTemp   ]
```

```
 5: .1         Mult
 6: -40        Offset
; Measure Relative Humidity
5:  Volts (SE) (P1)
 1: 1          Reps
 2: 5          2500 mV Slow Range
 3: 2          SE Channel
 4: 2          Loc [ RelHum    ]
 5: .1         Mult
 6: 0          Offset
; Check for value greater that 100%
6:  IF (X<=>F) (P89)
 1: 2          X Loc [ RelHum    ]
 2: 3          >=
 3: 100        F
 4: 30         Then Do
; Replace with 100% if greater
7:  Z=F (P30)
 1: 100        F
 2: 0          Exponent of 10
 3: 2          Z Loc [ RelHum    ]
8:  End (P95)
; Disable Air Temperature Sensor
9:  Do (P86)
 1: 51         Set Port 1 Low
; Measure PAR
10:  Volts (SE) (P1)
 1: 1          Reps
 2: 23         25 mV 60 Hz Rejection Range
 3: 3          SE Channel
 4: 3          Loc [ PAR       ]
 5: 233.19     Mult
```

```
 6: 0        Offset

; Check for less than zero and replace with zero if less
11:  IF (X<=>F) (P89)
 1: 3        X Loc [ PAR       ]
 2: 4        <
 3: 0        F
 4: 30       Then Do
12:  Z=F (P30)
 1: 0        F
 2: 0        Exponent of 10
 3: 3        Z Loc [ PAR       ]
13:  End (P95)
; Measure Solar Radiation
14:  Volt (Diff) (P2)
 1: 1        Reps
 2: 22       7.5 mV 60 Hz Rejection Range
 3: 3        DIFF Channel
 4: 4        Loc [ SolarRad  ]
 5: 120.92   Mult
 6: 0        Offset
; Check for less than zero and replace with zero if less
15:  IF (X<=>F) (P89)
 1: 4        X Loc [ SolarRad  ]
 2: 4        <
 3: 0        F
 4: 30       Then Do

16:  Z=F (P30)
 1: 0        F
 2: 0        Exponent of 10
 3: 4        Z Loc [ SolarRad  ]
```

```
17:  End (P95)
; Measure Wind Direction with a delay of 200ms
18:  Excite-Delay (SE) (P4)
 1: 1        Reps
 2: 5        2500 mV Slow Range
 3: 9        SE Channel
 4: 1        Excite all reps w/Exchan 1
 5: 20       Delay (units 0.01 sec)
 6: 1800     mV Excitation
 7: 12       Loc [ WindDir   ]
 8: .2       Mult
 9: 0        Offset
; Measure wind speed
19:  Pulse (P3)
 1: 1        Reps
 2: 1        Pulse Input Channel
 3: 21       Low Level AC, Output Hz
 4: 11       Loc [ WindSpeed ]
 5: .098     Mult
 6: 0        Offset
; Measure soil period for soil moisture measurement
20:  Period Average (SE) (P27)
 1: 1        Reps
 2: 4        2 V Peak to Peak/150 kHz Max. Freq.
 3: 7        SE Channel
 4: 10       No. of Cycles
 5: 5        Timeout (units = 0.01 seconds)
 6: 8        Loc [ SoilPrd   ]
 7: .001     Mult
 8: 0        Offset
; Disable soil moisture sensor
```

```
21:  Do (P86)
 1: 53       Set Port 3 Low

; Compute Soil moisture from period measurement
22:  Polynomial (P55)
 1: 1        Reps
 2: 8        X Loc [ SoilPrd   ]
 3: 9        F(X) Loc [ SoilMoist ]
 4: -.187    C0
 5: .037     C1
 6: .335     C2
 7: 0        C3
 8: 0        C4
 9: 0        C5
; Measure soil temperature
23:  Temp (107) (P11)
 1: 1        Reps
 2: 8        SE Channel
 3: 2        Excite all reps w/Exchan 2
 4: 7        Loc [ SoilTemp  ]
 5: 1        Mult
 6: 0        Offset
; Measure battery voltage
24:  Batt Voltage (P10)
 1: 10       Loc [ Battery   ]
; Measure precipitation
25:  Pulse (P3)
 1: 1        Reps
 2: 2        Pulse Input Channel
 3: 2        Switch Closure, All Counts
 4: 6        Loc [ Precip    ]
 5: .01      Mult
```

```
 6: 0        Offset

; Measure barometric pressure with 1 sec delay after “ON”
26:  Excitation with Delay (P22)
 1: 1        Ex Channel
 2: 0        Delay W/Ex (units = 0.01 sec)
 3: 65       Delay After Ex (units = 0.01 sec)
 4: 0        mV Excitation
27:  Volts (SE) (P1)
 1: 1        Reps
 2: 25       2500 mV 60 Hz Rejection Range
 3: 4        SE Channel
 4: 5        Loc [ Pressure  ]
 5: .184     Mult
 6: 600      Offset
; Disable pressure sensor
28:  Do (P86)
 1: 52       Set Port 2 Low
; Output variables
; Check if hour has elapsed
29:  If time is (P92)
 1: 0        Minutes (Seconds --) into a
 2: 60       Interval (same units as above)
 3: 10       Set Output Flag High
; Set array ID as 4
30:  Set Active Storage Area (P80)
 1: 1        Final Storage Area 1
 2: 4        Array ID
; Store Time stamping information
31:  Real Time (P77)
 1: 1220     Year,Day,Hour/Minute (midnight = 2400)
; Store instantaneous value of pressure
```

```
32:  Sample (P70)
 1: 1        Reps
 2: 5        Loc [ Pressure  ]
; Accumulate precip readings over the hour
33:  Totalize (P72)
 1: 1        Reps
 2: 6        Loc [ Precip    ]
; Store instantaneous value of air temperature
34:  Sample (P70)
 1: 1        Reps
 2: 1        Loc [ AirTemp   ]
; Store instantaneous value of relative humidity
35:  Sample (P70)
 1: 1        Reps
 2: 2        Loc [ RelHum    ]
; Average PAR over the hour
36:  Average (P71)
 1: 1        Reps
 2: 3        Loc [ PAR       ]
; Average Solar Radiation over the hour
37:  Average (P71)
 1: 1        Reps
 2: 4        Loc [ SolarRad  ]
; Store instantaneous value of soil temperature
38:  Sample (P70)
 1: 1        Reps
 2: 7        Loc [ SoilTemp  ]
; Store instantaneous value of soil moisture
39:  Sample (P70)
 1: 1        Reps
 2: 9        Loc [ SoilMoist ]
; Compute wind vector
```

```
40:  Wind Vector (P69)
 1: 1        Reps
 2: 60       Samples per Sub-Interval
 3: 0        S, é1, & å(é1) Polar
 4: 11       Wind Speed/East Loc [ WindSpeed ]
 5: 12       Wind Direction/North Loc [ WindDir   ]
; Check if day has elapsed
41:  If time is (P92)
 1: 0        Minutes (Seconds --) into a
 2: 1440     Interval (same units as above)
 3: 10       Set Output Flag High
; Set array ID as 5
42:  Set Active Storage Area (P80)
 1: 1        Final Storage Area 1
 2: 5        Array ID
; Time stamp data
43:  Real Time (P77)
 1: 1220     Year,Day,Hour/Minute (midnight = 2400)

44:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 10       Loc [ Battery   ]
45:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 5        Loc [ Pressure  ]
46:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 5        Loc [ Pressure  ]
47:  Totalize (P72)
```

```
 1: 1        Reps
 2: 6        Loc [ Precip    ]
48:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 11       Loc [ WindSpeed ]
49:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 1        Loc [ AirTemp   ]
50:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 1        Loc [ AirTemp   ]
51:  Average (P71)
 1: 1        Reps
 2: 1        Loc [ AirTemp   ]
52:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 2        Loc [ RelHum    ]
53:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 2        Loc [ RelHum    ]
54:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 3        Loc [ PAR       ]
55:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
```

```
 3: 3         Loc [ PAR       ]
56:  Average (P71)
 1: 1         Reps
 2: 3         Loc [ PAR       ]
57:  Maximize (P73)
 1: 1         Reps
 2: 0         Value Only
 3: 7         Loc [ SoilTemp  ]
58:  Minimize (P74)
 1: 1         Reps
 2: 0         Value Only
 3: 7         Loc [ SoilTemp  ]
59:  Average (P71)
 1: 1         Reps
 2: 7         Loc [ SoilTemp  ]
60:  Maximize (P73)
 1: 1         Reps
 2: 0         Value Only
 3: 9         Loc [ SoilMoist ]
61:  Minimize (P74)
 1: 1         Reps
 2: 0         Value Only
 3: 9         Loc [ SoilMoist ]
62:  Average (P71)
 1: 1         Reps
 2: 9         Loc [ SoilMoist ]
63:  Average (P71)
 1: 1         Reps
 2: 10        Loc [ Battery   ]
64:  Maximize (P73)
 1: 1         Reps
 2: 0         Value Only
```

```
 3: 4        Loc [ SolarRad  ]

65:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 4        Loc [ SolarRad  ]
66:  Average (P71)
 1: 1        Reps
 2: 4        Loc [ SolarRad  ]

*Table 2 Program
  01: 0.0000    Execution Interval (seconds)

*Table 3 Subroutines

End Program
```

# APPENDIX C1

## SAMPLE DATALOGGER PROGRAM FOR RF/INTERNET COMMUNICATION

This section contains the sample datalogger program used in the RF/Internet based communication test. The program was written using Campbell Scientific's EDLOG software which is a part of the PC 208W V2.00 suite.

——————————————File: lake_rf.csi——————————————

```
;{CR10}
*Table 1 Program
  01: 60.0000   Execution Interval (seconds)

; Enable pressure sensor
1:  Do (P86)
 1: 42       Set Port 2 High
; Enable Temperature Sensor
2:  Do (P86)
 1: 41       Set Port 1 High
; Delay for the temperature sensor. Voltage settling
3:  Excitation with Delay (P22)
 1: 1        Ex Channel
 2: 0        Delay W/Ex (units = 0.01 sec)
 3: 15       Delay After Ex (units = 0.01 sec)
 4: 0        mV Excitation
; Measure temperature
4:  Volts (SE) (P1)
 1: 1        Reps
 2: 5        2500 mV Slow Range
 3: 1        SE Channel
 4: 1        Loc [ AirTemp   ]
 5: .1       Mult
```

```
 6: -40      Offset
; Measure Relative Humidity
5:  Volts (SE) (P1)
 1: 1        Reps
 2: 5        2500 mV Slow Range
 3: 2        SE Channel
 4: 2        Loc [ RelHum    ]
 5: .1       Mult
 6: 0        Offset
; Disable temperature sensor
6:  Do (P86)
 1: 51       Set Port 1 Low
; Measure PAR
7:  Volts (SE) (P1)
 1: 1        Reps
 2: 23       25 mV 60 Hz Rejection Range
 3: 3        SE Channel
 4: 3        Loc [ PAR       ]
 5: 139.3    Mult
 6: 0        Offset
; Measure Radiation
8:  Volt (Diff) (P2)
 1: 1        Reps
 2: 22       7.5 mV 60 Hz Rejection Range
 3: 3        DIFF Channel
 4: 4        Loc [ SolarRad  ]
 5: .012     Mult
 6: 0        Offset
; QC for Radiation
9:  IF (X<=>F) (P89)
 1: 4        X Loc [ SolarRad  ]
 2: 4        <
 3: 0        F
 4: 30       Then Do

10:  Z=F (P30)
 1: 0        F
 2: 0        Exponent of 10
```

```
 3: 4        Z Loc [ SolarRad  ]

11:  End (P95)

12:  Excitation with Delay (P22)
 1: 1        Ex Channel
 2: 0        Delay W/Ex (units = 0.01 sec)
 3: 85       Delay After Ex (units = 0.01 sec)
 4: 0        mV Excitation
; Measure pressure
13:  Volts (SE) (P1)
 1: 1        Reps
 2: 25       2500 mV 60 Hz Rejection Range
 3: 4        SE Channel
 4: 5        Loc [ Pressure  ]
 5: .184     Mult
 6: 600      Offset

14:  Do (P86)
 1: 52       Set Port 2 Low
; Measure precip
15:  Pulse (P3)
 1: 1        Reps
 2: 2        Pulse Input Channel
 3: 2        Switch Closure, All Counts
 4: 6        Loc [ Precip    ]
 5: .01      Mult
 6: 0        Offset
; Diable pressure sensor
16:  Do (P86)
 1: 43       Set Port 3 High
; Measure Soil Moisture period
17:  Period Average (SE) (P27)
 1: 1        Reps
 2: 4        2 V Peak to Peak/150 kHz Max. Freq.
 3: 7        SE Channel
 4: 10       No. of Cycles
 5: 5        Timeout (units = 0.01 seconds)
```

```
 6: 8        Loc [ SoilPrd   ]
 7: .001     Mult
 8: 0        Offset

18:  Do (P86)
 1: 53       Set Port 3 Low
; Compute soil moisture
19:  Polynomial (P55)
 1: 1        Reps
 2: 8        X Loc [ SoilPrd   ]
 3: 9        F(X) Loc [ SoilMoist ]
 4: -.187    C0
 5: .037     C1
 6: .335     C2
 7: 0        C3
 8: 0        C4
 9: 0        C5
; Measure soil temperature
20:  Temp (107) (P11)
 1: 1        Reps
 2: 8        SE Channel
 3: 2        Excite all reps w/Exchan 2
 4: 7        Loc [ SoilTemp  ]
 5: 1        Mult
 6: 0        Offset
; Measure Wind Direction
21:  Excite-Delay (SE) (P4)
 1: 1        Reps
 2: 5        2500 mV Slow Range
 3: 9        SE Channel
 4: 1        Excite all reps w/Exchan 1
 5: 20       Delay (units 0.01 sec)
 6: 1800     mV Excitation
 7: 12       Loc [ WindDir   ]
 8: .2       Mult
 9: 0        Offset
; Measure Wind Speed
22:  Pulse (P3)
```

```
 1: 1        Reps
 2: 1        Pulse Input Channel
 3: 21       Low Level AC, Output Hz
 4: 11       Loc [ WindSpeed ]
 5: .098     Mult
 6: 0        Offset
; Measure Battery Voltage
23:  Batt Voltage (P10)
 1: 10       Loc [ Battery   ]
; Check if 5 min has elapsed
24:  If time is (P92)
 1: 0        Minutes (Seconds --) into a
 2: 5        Interval (same units as above)
 3: 10       Set Output Flag High
; Set array ID to 100
25:  Set Active Storage Area (P80)
 1: 2        Final Storage Area 2
 2: 100      Array ID
; Dump Time Stamp
26:  Real Time (P77)
 1: 1220     Year,Day,Hour/Minute (midnight = 2400)
;  Sample  Air  Temperature,  Relative  Humidity,  PAR,  Pressure,  Solar
Radiation, Soil Moisture, Soil Temperature
; Average Wind Speed and Wind Direction. Accumulate precip
27:  Sample (P70)
 1: 1        Reps
 2: 1        Loc [ AirTemp   ]
28:  Sample (P70)
 1: 1        Reps
 2: 2        Loc [ RelHum    ]
29:  Sample (P70)
 1: 1        Reps
 2: 3        Loc [ PAR       ]
30:  Sample (P70)
 1: 1        Reps
 2: 5        Loc [ Pressure  ]
31:  Sample (P70)
 1: 1        Reps
```

```
 2: 4        Loc [ SolarRad  ]
32:  Sample (P70)
 1: 1        Reps
 2: 7        Loc [ SoilTemp  ]
33:  Sample (P70)
 1: 1        Reps
 2: 9        Loc [ SoilMoist ]
34:  Average (P71)
 1: 1        Reps
 2: 11       Loc [ WindSpeed ]
35:  Average (P71)
 1: 1        Reps
 2: 12       Loc [ WindDir   ]
36:  Totalize (P72)
 1: 1        Reps
 2: 6        Loc [ Precip    ]

; Check if 15 min has elapsed
37:  If time is (P92)
 1: 0        Minutes (Seconds --) into a
 2: 15       Interval (same units as above)
 3: 10       Set Output Flag High
; Set array ID to 1
38:  Set Active Storage Area (P80)
 1: 2        Final Storage Area 2
 2: 1        Array ID
; Dump Time Stamping information
39:  Real Time (P77)
 1: 1220     Year,Day,Hour/Minute (midnight = 2400)
; Sample Air Temperature, Relative Humidity, PAR, Pressure, Solar
Radiation, Soil Moisture, Soil Temperature
; Average Wind Speed and Wind Direction. Accumulate precip
40:  Sample (P70)
 1: 1        Reps
 2: 1        Loc [ AirTemp   ]
41:  Sample (P70)
 1: 1        Reps
 2: 2        Loc [ RelHum    ]
```

```
42:  Sample (P70)
 1: 1        Reps
 2: 3        Loc [ PAR       ]

43:  Sample (P70)
 1: 1        Reps
 2: 5        Loc [ Pressure  ]
44:  Sample (P70)
 1: 1        Reps
 2: 4        Loc [ SolarRad  ]
45:  Sample (P70)
 1: 1        Reps
 2: 7        Loc [ SoilTemp  ]
46:  Sample (P70)
 1: 1        Reps
 2: 9        Loc [ SoilMoist ]
47:  Average (P71)
 1: 1        Reps
 2: 11       Loc [ WindSpeed ]
48:  Average (P71)
 1: 1        Reps
 2: 12       Loc [ WindDir   ]
49:  Totalize (P72)
 1: 1        Reps
 2: 6        Loc [ Precip    ]

; Check if an hour has elapsed
50:  If time is (P92)
 1: 0        Minutes (Seconds --) into a
 2: 60       Interval (same units as above)
 3: 10       Set Output Flag High
; Set FSA to 1 and array to array ID to 2
51:  Set Active Storage Area (P80)
 1: 1        Final Storage Area 1
 2: 2        Array ID
; Dump Time Stamp infomation
52:  Real Time (P77)
 1: 1220     Year,Day,Hour/Minute (midnight = 2400)
```

```
53:  Sample (P70)
 1: 1        Reps
 2: 5        Loc [ Pressure  ]

54:  Totalize (P72)
 1: 1        Reps
 2: 6        Loc [ Precip    ]

55:  Sample (P70)
 1: 1        Reps
 2: 1        Loc [ AirTemp   ]

56:  Sample (P70)
 1: 1        Reps
 2: 2        Loc [ RelHum    ]

57:  Average (P71)
 1: 1        Reps
 2: 3        Loc [ PAR       ]

58:  Average (P71)
 1: 1        Reps
 2: 4        Loc [ SolarRad  ]

59:  Sample (P70)
 1: 1        Reps
 2: 7        Loc [ SoilTemp  ]

60:  Sample (P70)
 1: 1        Reps
 2: 9        Loc [ SoilMoist ]

61:  Wind Vector (P69)
 1: 1        Reps
 2: 15       Samples per Sub-Interval
 3: 0        S, é1, & å(é1) Polar
 4: 11       Wind Speed/East Loc [ WindSpeed ]
```

```
 5: 12       Wind Direction/North Loc [ WindDir   ]

62:  If time is (P92)
 1: 0        Minutes (Seconds --) into a
 2: 1440     Interval (same units as above)
 3: 10       Set Output Flag High

63:  Set Active Storage Area (P80)
 1: 1        Final Storage Area 1
 2: 3        Array ID

64:  Real Time (P77)
 1: 1220     Year,Day,Hour/Minute (midnight = 2400)

65:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 10       Loc [ Battery   ]

66:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 5        Loc [ Pressure  ]

67:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 5        Loc [ Pressure  ]

68:  Totalize (P72)
 1: 1        Reps
 2: 6        Loc [ Precip    ]

69:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 11       Loc [ WindSpeed ]
```

```
70:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 1        Loc [ AirTemp   ]

71:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 1        Loc [ AirTemp   ]

72:  Average (P71)
 1: 1        Reps
 2: 1        Loc [ AirTemp   ]

73:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 2        Loc [ RelHum    ]

74:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 2        Loc [ RelHum    ]

75:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 3        Loc [ PAR       ]

76:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 3        Loc [ PAR       ]

77:  Average (P71)
 1: 1        Reps
 2: 3        Loc [ PAR       ]
```

```
78:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 7        Loc [ SoilTemp  ]

79:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 7        Loc [ SoilTemp  ]

80:  Average (P71)
 1: 1        Reps
 2: 7        Loc [ SoilTemp  ]

81:  Maximize (P73)
 1: 1        Reps
 2: 0        Value Only
 3: 9        Loc [ SoilMoist ]

82:  Minimize (P74)
 1: 1        Reps
 2: 0        Value Only
 3: 9        Loc [ SoilMoist ]

83:  Average (P71)
 1: 1        Reps
 2: 9        Loc [ SoilMoist ]

84:  Average (P71)
 1: 1        Reps
 2: 10       Loc [ Battery   ]

*Table 2 Program
  01: 0.0000    Execution Interval (seconds)

*Table 3 Subroutines

End Program
```

# APPENDIX C2

## SAMPLE HTML PAGES OF RF/INTERNET COMMUNICATION

This section contains typical HTML page for RF/Internet based communication. The page can be accessed via web site at http://www.nc-climate.ncsu.edu/rf/lake

Lake Wheeler Road
Field Laboratory
3720 Lake Wheeler
Road
Raleigh, NC 27603

NCARS Weather & Climate Network

**Following is the Weather Data**
**Time : 11:30, 8 May 2000**

| Weather Parameter | Value |
|---|---|
| Air temperature (deg. C) | 27.2 |
| Relative Humidity (%) | 49 |
| Barometric Pressure (mb) | 1004.0 |
| Solar radiation ($W.m^{-2}$) | NP |
| Soil temperature (deg. C) | 21.6 |
| Wind Speed ($m.s^{-1}$) | 2.2 |
| Wind Direction (deg) | 50 |

**History of the Weather Data for past 2 Hours**

| Julian Day | Year | Time | Air Temp. | Soil Temp. | Relative Humidity | Barometric Pressure | Wind Speed | Wind Direction | Total Radiation |
|---|---|---|---|---|---|---|---|---|---|
| 129 | 2000 | 930 | 23.3 | 20.3 | 59 | 1004.0 | 1.9 | 22 | NP |
| 129 | 2000 | 935 | 23.4 | 20.3 | 60 | 1004.0 | 2.0 | 24 | NP |
| 129 | 2000 | 940 | 23.7 | 20.4 | 61 | 1004.0 | 2.1 | 26 | NP |
| 129 | 2000 | 945 | 23.5 | 20.4 | 58 | 1004.0 | 2.4 | 28 | NP |
| 129 | 2000 | 950 | 23.8 | 20.5 | 58 | 1004.0 | 2.8 | 29 | NP |
| 129 | 2000 | 955 | 23.9 | 20.5 | 59 | 1004.0 | 2.4 | 31 | NP |
| 129 | 2000 | 1000 | 24.0 | 20.5 | 60 | 1004.0 | 2.4 | 32 | NP |
| 129 | 2000 | 1005 | 24.2 | 20.6 | 58 | 1004.0 | 2.7 | 33 | NP |
| 129 | 2000 | 1010 | 24.4 | 20.6 | 57 | 1004.0 | 2.6 | 34 | NP |
| 129 | 2000 | 1015 | 24.4 | 20.6 | 56 | 1004.0 | 3.0 | 35 | NP |
| 129 | 2000 | 1020 | 24.6 | 20.7 | 55 | 1004.0 | 3.3 | 36 | NP |
| 129 | 2000 | 1025 | 24.9 | 20.8 | 57 | 1004.0 | 3.0 | 37 | NP |
| 129 | 2000 | 1030 | 25.1 | 20.8 | 55 | 1004.0 | 2.7 | 38 | NP |
| 129 | 2000 | 1035 | 25.6 | 20.9 | 54 | 1004.0 | 2.6 | 39 | NP |
| 129 | 2000 | 1040 | 25.6 | 20.9 | 52 | 1004.0 | 2.7 | 40 | NP |
| 129 | 2000 | 1045 | 25.6 | 21.0 | 51 | 1004.0 | 2.8 | 41 | NP |
| 129 | 2000 | 1050 | 25.8 | 21.0 | 53 | 1004.0 | 2.6 | 42 | NP |
| 129 | 2000 | 1055 | 26.3 | 21.1 | 56 | 1004.0 | 2.0 | 43 | NP |
| 129 | 2000 | 1100 | 26.5 | 21.2 | 55 | 1004.0 | 2.2 | 44 | NP |
| 129 | 2000 | 1105 | 26.4 | 21.2 | 51 | 1004.0 | 2.5 | 45 | NP |
| 129 | 2000 | 1110 | 26.6 | 21.3 | 50 | 1004.0 | 2.3 | 46 | NP |
| 129 | 2000 | 1115 | 26.8 | 21.3 | 50 | 1004.0 | 2.6 | 47 | NP |
| 129 | 2000 | 1120 | 26.9 | 21.4 | 49 | 1004.0 | 2.1 | 48 | NP |
| 129 | 2000 | 1125 | 26.9 | 21.5 | 51 | 1004.0 | 2.1 | 49 | NP |

**This page is updated every 5 minutes ( 300 seconds )**
**This page was last-updated on 8, May, 2000 at 11:32 hrs**

# APPENDIX C3

## SHELL SCRIPT FOR PRIMARY DATA COLLECTION SERVER

This section of the thesis contains the shell script at the primary data collection server. The script is executed at regular intervals of 5 minutes by a "cron" job.

File: download.sh

```
#! /bin/bash

# DESCRIPTION
# The shell script is written to download the data from the remote
agnet site
# which is on RF. The downloading is done using telcom software from
Campbell
# Scientific. The downloaded data is backed up and mail to user "agnet"
at
# shp.meas.ncsu.edu. After mailing the recent data, the data file is
moved
# to a another location with time stamped file name. Since the cron job
is run
# by root, the file permissions are change back to agnet. lastly the
file
# make a log in the log file.

# Please dosemu manual before making any changes to the below
instruction.
# This instruction executes a series of dos commands and exits the dos
emulator
/usr/bin/dos -D-a -I ’keystroke "D:\rcd \\home\\agnet\\campbell\rtelcom
lake-rf /C\rtelcom lake-dly /C\rexitemu\r"’ 2> /dev/null

# Copy the rec# Create a logent downloaded file to main backup file
```

```
/bin/cat /home/agnet/campbell/lake-rf.dat >>
/home/agnet/campbell/RF_LAKE.DAT

# Also copy the daily data file to its mail backup file
/bin/cp /home/agnet/campbell/lake-dly.dat
/home/agnet/campbell/DLY_LAKE.DAT

# Check if new data has arrived and only then mail the newer data.
if [ -s /home/agnet/campbell/lake-rf.dat ]; then
      mail -s "LAKE-RF" agnet@shp.meas.ncsu.edu <
/home/agnet/campbell/lake-rf.dat
fi

# Move the recent data file with unique filename (Time Stamped)
/bin/mv /home/agnet/campbell/lake-rf.dat
/home/agnet/campbell/backup/lake-rf.‘date +%H%M-%m%d%Y‘

# Create a file with name lake-rf.dat of null size
/bin/touch /home/agnet/campbell/lake-rf.dat

# Reset the permissions on the file
/bin/chown agnet:users /home/agnet/campbell/*.*

# Create a log
/bin/date >> /home/agnet/campbell/execution.log
```

# APPENDIX C4

## SHELL SCRIPT FOR SECONDARY DATA COLLECTION SERVER

This section of the thesis contains the shell script at the secondary data collection server. The script is executed at regular intervals of 5 minutes by a “cron” job.

File: processmail.sh

```
# !/bin/bash

# DESCRIPTION:
# The Shell script is written to process the mail on the mailbox of
agnet
# The script cuts the last mail message about Lake Wheeler RF Data and
# pastes it to the temporary file. This then extracts the data from the
# temporary file and puts it into Data File.
# Author: Vinayak K. Parameshwara
# Created: 03/26/2000

# Delete all temporary Files
if [ -f /tmp/tmp1 -o -f /tmp/tmp2 ]; then
      rm /tmp/tmp*
fi
if [ -f /tmp/mailbox.new ]; then
      rm /tmp/mailbox.new
fi
# Cut the mail from the Mailbox @ /var/spool/mail/agnet
# Create a dummy file /tmp/tmp file
# Get the Last Mail Message - General one
sed -n -e ‘grep -n "From " /var/spool/mail/agnet | awk -F: ’{print $1}’
| tail -1‘,‘wc -l /var/spool/mail/agnet | awk ’{print $1-1}’‘p
/var/spool/mail/agnet > /tmp/tmp1
while [ -s /tmp/tmp1 ]; do
      # Dump text from the message which has "Subject: LAKE-RF"
```

```
      sed -n -e '/Subject: LAKE-RF/,//p' /tmp/tmp1 > /tmp/tmp2
      # Check if the last message has the "Subject: LAKE-RF"
      if [ -s /tmp/tmp2 ]; then
            # Process the message to get the data out
            sed -n -e '/^[0-9]/p' /tmp/tmp2 >> /home/agnet/RF_LAKE.DAT
            # Cut the message from the Mailbox
            sed -n -e 1,`grep -n "From " /var/spool/mail/agnet | awk -
F: '{print $1-1}' | tail -1`p /var/spool/mail/agnet > /tmp/mailbox.new
            # Copy back the mailbox
            cp /tmp/mailbox.new /var/spool/mail/agnet
            # Delete the created mailbox
            rm /tmp/mailbox.new
      else
            # Create a new file in first run; doesn't matter later runs
            touch /tmp/tmp4
            # Extract the unwanted mail message from mailbox
            sed -n -e `grep -n "From " /var/spool/mail/agnet | awk -F:
'{print $1}' | tail -1`,`wc -l /var/spool/mail/agnet | awk '{print
$1}'`p /var/spool/mail/agnet > /tmp/tmp3
            # Put the old messages over the new messages
            cat /tmp/tmp4 >> /tmp/tmp3
            # Delete Old messages
            rm -f /tmp/tmp4
            # Make the new message as old
            mv /tmp/tmp3 /tmp/tmp4
            # Cut the message from mailbox - Temporaily
            sed -e `grep -n "From " /var/spool/mail/agnet | awk -F:
'{print $1}' | tail -1`,`wc -l /var/spool/mail/agnet | awk '{print
$1}'`d /var/spool/mail/agnet > /tmp/mailbox.new
            # Copy back the mailbox
            cp /tmp/mailbox.new /var/spool/mail/agnet
            # Delete newly created mailbox
            rm /tmp/mailbox.new
      fi
      # Get the Last Mail Message - General one
      rm /tmp/tmp1
      # Get last
      if [ -s /var/spool/mail/agnet ]; then
```

```
            sed -n -e ‘grep -n "From " /var/spool/mail/agnet | awk -F:
’{print $1}’ | tail -1‘,‘wc -l /var/spool/mail/agnet | awk ’{print $1-
1}’‘p /var/spool/mail/agnet > /tmp/tmp1
      fi
done
cp /tmp/tmp4 /var/spool/mail/agnet
```

# APPENDIX D1

## SAMPLE QC FILE

This section of the thesis contains the sample QC file that acts as one of the inputs file to the “convert” program during execution.

————————————————File: clin_twr.qc————————————————

```
# Clean section cleans the data before loading the data into the
arrays. The data point which
# satisfies the condition is replaced by the <replace> value. The
<month> indicates
# the number of months the condition is applicable (0 thru’ 11 - 0
stands for Jan and 11 for Dec)
# While declaring the month option either the conditions should be
declared for all months else
# a default values should be declared which holds good for all months.

clean: RelHum > 100 replace=100
clean: RelHum < 0   replace=0
clean: WindDir > 360 replace=360
clean: WindDir < 0   replace=0
clean: SolarRad < 0  replace=0
clean: PAR < 0 replace=0

range: AirTempC > 50 replace=99999
range: AirTempC > 50 replace=99999 month=0,1,2,3
range: AirTempC < -30 replace=99999
range: AirTempC < -30   replace=99999 month=0,1,2,3
range: SoilTempC > 50 replace=99999
range: SoilTempC > 50 replace=99999 month=0,1,2,3
range: SoilTempC < -30 replace=99999
range: SoilTempC < -30  replace=99999 month=0,1,2,3
range: Pressure < 850 replace=99999
```

```
range: Pressure > 1100 replace=99999
range: WindSpeed < 0 replace=99999
range: WindSpeed > 50 replace=99999
range: PAR < 0 replace=0
range: PAR > 4000 replace=99999
range: SolarRad < 0 replace=0
range: SolarRad > 1500 replace=99999
```