8-2003

A Distributed Control System for Priority-Based Site-Specific Irrigation

Fabio Rodrigues de Miranda
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Fabio Rodrigues de Miranda entitled "A Distributed Control System for Priority-Based Site-Specific Irrigation." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Biosystems Engineering.

Ronald E. Yoder, Major Professor

We have read this dissertation and recommend its acceptance:

John B. Wilkerson, Daniel C. Yoder, John R. Buchanan, Richard A. Straw

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Fabio Rodrigues de Miranda entitled "A Distributed Control System for Priority-Based Site-Specific Irrigation." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Biosystems Engineering.

Ronald E. Yoder _
Major Professor

We have read this dissertation
and recommend its acceptance:

__John B. Wilkerson_______

__Daniel C. Yoder_________

__John R. Buchanan________

__Richard A. Straw_________

Accepted for the Council:

__Anne Mayhew_______
Vice Provost and
Dean of Graduate Studies

(Original signatures are on file with official student records.)

A DISTRIBUTED CONTROL SYSTEM FOR PRIORITY-BASED SITE-SPECIFIC IRRIGATION

A Dissertation
Presented for the
Doctor of Philosophy Degree
The University of Tennessee, Knoxville

Fabio Rodrigues de Miranda
August 2003

DEDICATION

This dissertation is dedicated to my wife Selma, to my daughter Gabriela, to my parents, and my brothers for always believing in me and encouraging me to reach higher to achieve my goals.

ACKNOWLEDGMENTS

I would like to thank the Brazilian Agricultural Research Corporation (Embrapa) for the financial support during the course and all colleagues at the Embrapa Tropical Agro-industry Research Center for conceding me the opportunity for this training. I also would like to thank the Brazilian National Council for Scientific and Technological Development (CNPq) for the financial support for school and maintenance.

I would like to express my gratitude to my fellow students, the staff, and the faculty members of the Department of Biosystems Engineering and Environmental Science of The University of Tennessee for the collaboration over these four years. Special thanks are expressed to Dr. Ronald E. Yoder and Dr. John Wilkerson for their friendship, support, and guidance in all phases of the study. I also would like to express my appreciation to Dr. Daniel Yoder, Dr. John Buchanan, and Dr. Allen Straw, members of my doctoral committee, for their review, comments, and assistance.

I am also especially grateful to David Smith for his help in the development of the electronic circuit involved in this study, and to Wesley Wright for his support during the field part of the research.

Finally, I would like to thank my wife Selma and our daughter Gabriela for their love, patience, and support during the course, and all the friends that we met in Knoxville for their friendship and support.

ABSTRACT

Site-specific irrigation enables maximizing yields and water use efficiency for fields with variation in soil water availability. Distributed control for fixed irrigated systems, with controllers close to the sensors and actuators in the field, is easier to install and maintain, and less susceptible to damage by lightning strikes compared to centralized control, but is only economically viable with affordable, low-power controllers. A low cost, solar-powered, feedback irrigation controller for distributed control of fixed irrigation systems was developed and tested. The system used soil water potential measurements to control the amount of water applied to each specific zone of a field. Priority scheduling and hydraulic pressure measurements were used to allocate water resources among irrigation controllers. Each irrigation controller was autonomously powered, minimizing maintenance and eliminating hard-wire connections among control units. The study methodology involved system design (hardware and software), experimental implementation, performance evaluation, and power supply optimization. The irrigation controller proved to be effective in maintaining the soil water potential in the root zone close to a predetermined set point. Performance of the priority scheduling approach for water allocation among the irrigation controllers was satisfactory, with irrigation of management zones always occurring according to the priority rank, and under adequate operating pressure. Advantages of the control system compared to centralized control systems include significant reduction in wiring costs, lower risk of system shut down, and higher flexibility.

TABLE OF CONTENTS

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

FIGURE **PAGE**

CHAPTER 1

INTRODUCTION

Statement of the Problem

Agriculture is the primary user of water in the world, with most of that going to irrigation. Irrigation covers about 20 percent of the cropland in the world, and contributes 40 percent of total food production. Irrigated agriculture is responsible for approximately 70 percent of all the freshwater withdrawn in the world, and more water will be used for irrigation in the future, as world food production continues to increase to meet demand (FAO, 2002). The projected increase in irrigated agriculture will require significant improvements in irrigation management to be sustainable.

Irrigation scheduling is the process of determining when and how much to irrigate, and is an important element in improving water use efficiency. Scheduling maximizes irrigation efficiencies by determining the exact amount of water needed to replenish the soil moisture to a desired level. Over-irrigation wastes water, energy, and labor, leaches nutrients below the root zone, reduces soil aeration, and ultimately reduces crop yield and product quality. Under-irrigation stresses the plant and also reduces yield and product quality.

Variations in water availability across a field due to different soil characteristics or crop needs may require site-specific irrigation management to achieve optimum yields and maximize water use efficiency. Although many irrigation scheduling methods have been developed over the years, acceptance by farmers has been limited due to cost, time,

information, and the quality of decisions involved. A solution to this problem is possible through the total automation of irrigation using feedback control systems.

Most irrigation control systems use centralized control, with soil moisture sensors and actuators in the field and the controller in a central location, requiring separate wires connecting the sensors, controllers, and actuators. This approach is expensive and difficult to maintain in an environment where mechanical damage and lightning are concerns. This is especially a concern for site-specific irrigation, which requires a large network of sensors and actuators.

Distributed control, with each zone of the field having one controller interconnected to sensors and valves, would make the control system easier to install and maintain, as well as less susceptible to lightning damage. However, since additional control units are required, using distributed control for site-specific irrigation is only viable with low-cost controllers and sensing/actuating devices that have low power requirements. Wireless communication among the controllers is also required to optimize the hydraulic operation of the irrigation system.

Research Objectives

The objectives of this research were:

1. To develop and evaluate a low cost, solar-powered, feedback irrigation controller that maintains a desired soil water potential level within the root zone.
2. To develop a distributed control system for site-specific irrigation that eliminates hard-wire connections between control units, and allocates water resources among

zones of the field according to a priority rank, always giving water to higher priority zones over lower priority zones.

CHAPTER 2

LITERATURE REVIEW

Irrigation Scheduling

Irrigation scheduling is commonly defined as determining when to irrigate (frequency), and how much water to apply (quantity). According to Stegman et al. (1981), irrigation can be scheduled to meet objectives such as maximizing yield per unit area, maximizing yield per unit of water applied, maximizing net profit, and minimizing energy requirements.

The success of any irrigation method depends largely on utilizing irrigation scheduling principles to develop a management plan, and on efficiently implementing the plan. Excessive irrigation leaches salts from the root zone, which is beneficial for salinity control. However, excessive irrigation may also leach nutrients important to the crop; the leached nutrients can become pollutants in groundwater and streams. Under-irrigation may limit yields, especially if it occurs during flowering and fruit development stages.

Research on irrigation scheduling began more than fifty years ago, and since then several irrigation-scheduling methods have been developed (Jensen et al., 1970; Campbell and Campbell, 1982; Shearer and Vomocil, 1982; Stegman et al., 1976). Quantitative irrigation scheduling methods are based on two approaches: (a) soil and/or crop monitoring, and b) soil water balance computations. For monitoring methods, the soil water content or matric potential is generally measured at several locations in a field. Methods based on plant measurements generally involve monitoring leaf water potential and/or canopy temperature. Soil water balance computations require estimates of soil

water storage capacity, rooting depth, allowable depletion, and crop evapotranspiration to develop an irrigation schedule (Martin et al., 1990).

Since soil and crop parameters involved in the soil water balance approach are difficult to estimate with precision, field measurements of soil and plant conditions are necessary parts of the irrigation scheduling procedure (Howell et al., 1986). Using monitoring methods for irrigation scheduling requires an understanding of potential limitations, such as: (a) the threshold value of the monitored quantity may change depending on the measurement location, climate, crop stage, etc; (b) spatial variability may require that many sites be monitored to represent the average field condition; and (c) monitoring crop conditions (leaf water potential and/or canopy temperature) can be used to estimate irrigation timing, but does not provide any information on the amount of water to apply (Martin et al., 1990).

For irrigation systems capable of applying water on a high-frequency basis (microirrigation, solid-set), the need for irrigation scheduling techniques based on management allowed depletion or plant water stress thresholds becomes less important (Martin et al., 1990). Under high-frequency irrigation scheduling, the soil water potential should be maintained nearly constant, with irrigation being applied at a rate equivalent to the evapotranspiration. Hence, there is a strong need for “real-time” irrigation scheduling (Phene et al., 1992).

According to Stegman et al. (1981), despite the availability of several irrigation scheduling techniques, farmers have not adopted any particular method. Many factors contribute to the lack of adoption, the most significant being: (a) the cost of irrigation

water is often low relative to the costs of practices that would improve water management; (b) yield reductions caused by delayed irrigations, improper fertilization, and excessive irrigations are not easily recognized or quantified; (c) the necessary data are often not available to those making water management decisions on a day-to-day basis; (d) irrigation management decisions are generally made by busy people with limited technical background and training in the management of a complex crop-soil-climate system; and (e) traditional scheduling methods have tended to require that every farmer/manager become a specialist in irrigation water management.

Full automation of irrigation systems through feedback control represents a solution for both the farmer's lack of interest in conventional irrigation scheduling techniques and the need for "real-time" irrigation scheduling. Advances in electronics and the decreasing costs of computers, microcontrollers, sensors, and actuators in the last decade have made the development of irrigation control systems that integrate sensing, decision-making, and controlling of irrigation variables economically viable.

Irrigation Automated Control

A well-controlled irrigation system is one that optimizes the spatial and temporal distribution of water. Optimization does not necessarily produce the highest yield or use the least water, but maximizes the benefit–to-cost ratio (Hillel, 1980).

The implementation of control for an irrigation system depends largely on the irrigation method used. For surface irrigation, limitations imposed by the water distribution system and the advance time in the furrows or basins restrict the irrigation frequency. For such systems, automated control has been applied mainly with the

objectives of reducing the labor in water distribution operations and increasing the irrigation efficiency by reducing the advance time and the runoff losses after the water reaches the end of the field.

For pressurized irrigation systems water delivery is more easily controlled than for surface irrigation systems. Confined within a pipeline, water responds very quickly to pressure or flow rate changes. Regardless of the irrigation system, whether drip or sprinkler, uniform water application depends on maintaining control of the flow rate through each outlet. Poor performance of pressurized systems often results from changes in water pressure, either from one place to another or with time (Duke et al., 1990).

According to Phene (1986), control systems are usually divided into open-loop systems or closed-loop systems. An open-loop control system is defined as one in which the results of the operation are independent of the input and an operator is needed to make decisions. In a closed-loop control system the input is directly dependent on the output through a feedback mechanism from the output to the input. The feedback allows for comparison of the output to some reference input signal, thus achieving precise control.

Full automation of irrigation systems using feedback (closed-loop) systems involves four distinct functions:

- Sensing and measuring the parameters to be controlled;
- Collecting and correlating the measurements, and the decision for the course of action;
- Activating the irrigation system by turning on or off a valve, a pump, etc.;

- Establishing a feedback circuit to check whether to continue or stop the irrigation.

Phene (1986) described four basic closed-loop feedback approaches that can be used in irrigation: (1) soil water, (2) plant water, (3) evapotranspiration, and (4) combinations of 1, 2, and 3. Irrigation control using plant water status is based on the concept that the plant integrates the effect of soil and atmospheric conditions. Several methods are available to estimate plant water status, such as the plant canopy temperature method, the stem diameter method, the leaf water potential method, the sap flow method, and others. The most frequently used methods for automatic irrigation control are the plant canopy temperature method and the sap flow method (Phene, 1986).

Wanjura et al. (1992) developed a feedback automated drip irrigation system for cotton, controlled solely by continuously measuring plant canopy temperature. Accumulated daily time when canopy temperature was above a biologically determined optimum temperature was used to start an irrigation. However, time thresholds and threshold temperatures are required for each crop and region, and these are not yet available for most crops.

Van Bavel (1995) presented a closed-loop control for microirrigation systems that continuously monitored the transpiration of individual plants using sap flow sensors, and delivered a matching amount of water using a pump controller. However, the system still required some soil moisture monitoring to prevent soil saturation, and measurement of the soil moisture level at the beginning of the planting period for the crop.

Feedback control systems based on evaporation measurements were described by Vermeiren and Jobling (1980) and Phene et al. (1992). In both systems the control was

accomplished by measuring the evaporation in a class "A" pan, and triggering the irrigation when the water level dropped to a preset limit. The pan was refilled proportionally to the amount of water being applied through irrigation, and a feedback system was installed so that the system switched off when the pan was refilled.

Phene (1986) described a feedback irrigation controller that used the direct measurement of crop evapotranspiration by a modified crop lysimeter. A water tank was attached to the lysimeter so that the weight of the daily irrigation was included in the weight of the lysimeter. After one millimeter of evapotranspiration occurred, the lysimeter was automatically irrigated by a subsurface drip irrigation system to maintain steady state soil water potential. The lysimeter tank was automatically refilled daily at night to a constant tank level. Therefore, the accumulated daily change of lysimeter weight represented the crop growth and total weight.

The method for soil sensor control of high-frequency irrigation systems is based on four assumptions: (a) irrigation water is distributed uniformly throughout the field; (b) plant population is uniform in size and distribution; (c) water is used uniformly from the soil, mostly by evapotranspiration; and (d) the soil water sensor is installed in the root zone of an area of the soil-plant-atmosphere system typical of average field conditions, or several sensors are installed to obtain an average soil matric potential (Phene and Howell, 1981).

High-frequency irrigation can be controlled accurately by an electronic feedback soil moisture sensor installed in the crop root zone (Phene and Howell, 1984). Automatic irrigation using feedback from soil sensors makes it possible to maintain almost constant

soil water potential in the root zone. This produces the desired plant responses, and hence high yields, while using the exact volume of water required to maintain the crop.

Monitoring soil water potential and controlling an irrigation system automatically requires equipment to automatically sample several sensors sequentially, compare each sensor output to the soil matric potential level at which irrigation is to start, and produce computer outputs capable of controlling the irrigation system. The first soil-sensor-based feedback control systems were developed in the 1960s. Electro-tensiometers equipped with mercury-filled manometers were used in Israel, with electric switching units as controllers. The mercury level was used to make or break the contact between two electrodes and to transmit a signal to the controller, opening or closing solenoid valves (Arlosoroff, 1971).

A non-electronic feedback control was presented by Peterson et al. (1993) who developed an irrigation control valve that mechanically linked the soil water potential, expressed through negative pressure developed in a tensiometer, to the position of a piston that controlled the flow of water through the valve. The control valve required no external power for opening or closing. The flow rate increased as tensiometer pressure decreased due to soil dryness, and allowed the adjustment of the threshold soil matric potential to initiate and end the irrigation. However, the control valve was only tested for irrigation of a single container, and it was not proved that it could control the irrigation of an entire irrigation set. In addition to that, tensiometers require periodic servicing, which is not desirable for feedback irrigation control systems.

Within either open-loop or closed-loop control systems, three major control modes are available: (1) on-off control, (2) stepwise control, and (3) continuous control (Phene, 1986). The ideal irrigation system would supply water directly to the root zone at exactly the rate that the plant is using the water. This would require a continuously varying flow rate throughout the day. In practice, however, most irrigation systems can only supply water at a fixed rate, in an on-off mode, and matching the irrigation supply to the crop needs is achieved by varying the irrigation time.

Instrumentation for Irrigation Control

Automated open-loop control of solid-set sprinkler systems and microirrigation systems is typically done using timers that turn the pump on or off and control the operation time of valves in the field, allowing the system to run for a predetermined period of time or to deliver a predetermined volume of water to each irrigation set. Electronic timers have been widely used in the last decade because they are relatively inexpensive, permit great flexibility in the sequencing of irrigation sets, and provide very precise timing.

According to Phene (1986), the instrumentation needed to control pressurized irrigation systems can be divided into the following categories: (1) controllers, (2) sensors, and (3) valves. For closed-loop systems, the controller must be capable of: a) receiving feedback information about the soil moisture status from sensors in the field, weather data, plant water stress, line pressure, flow rate, etc.; b) comparing that information with desired limits, and modifying the irrigation cycle accordingly; and c) sending commands for operating the actuators (valves, pumps, etc.). Several closed-loop

irrigation control systems have been developed using microcomputers and data loggers as controllers (Phene and Howell, 1984; Stone et al., 1985; Zazueta and Smajstrla, 1992; Wessels et al., 1995; Testezlaf et al., 1997; Torre-Neto et al., 2000; Meron et al., 1995; Shock et al., 1998).

Various types of soil moisture sensors, weather instrumentation, and plant water stress or crop canopy temperature sensors are available that can be used in feedback mode for irrigation control. Pressure sensors are often used to provide information about the irrigation system operation, such as pump operation, pipeline leakage, and plugging of emitters and filters.

Whenever more than one irrigation set is supplied from the primary water source, some type of valving is usually required to allow selection of the portion of the field to be irrigated. Automated valves are used to switch water on or off, flush filters, mains and laterals, sequence water from one field to another, or regulate pressure in mains, submains, or laterals. The controller issues commands for valve operation and may receive feedback information to verify correct operation.

Electrically operated valves that use solenoid switches to control diaphragms or pistons are the most widely used valves for automatic irrigation systems. Solenoid valves usually operate using 24-V AC power. Electric surges in electrical storms may represent a problem for such systems, especially when using centralized control. Latching solenoid valves, which use 9-V or 12-V DC power, facilitate the automation of fields where AC power is not available.

Communication between the irrigation controller and control point has traditionally been through low voltage buried wires or small diameter plastic tubing for hydraulic systems. Simple electronic controllers require a separate wire to each control point, with a second neutral conductor, which is often shared by several control points. As control requirements become more sophisticated, the cost of a separate wire to each control point may become prohibitive. Where digital electronic controls are utilized, often in conjunction with microprocessors or personal computers, it may be feasible for a single wire pair or triplet to provide power and control signals to all control points (Duke et al., 1990).

Torre-Neto et al. (2000) developed an automated irrigation system that used a single cable, a multipoint serial data communication bus, to power and control a network of soil matric sensors and solenoid valves. The controller was configured using a personal computer and each sensor or valve had an embedded circuit that carried out the digital conversion close to the measurement/actuation points.

Wireless communication using radio telemetry has also been used for irrigation control. Modern radio transceivers and digital signal processing equipment allow very sophisticated checking of data to ensure accurate transmission.

More recently, spread spectrum technology has been studied for irrigation control communication. Spread spectrum wireless technology enables spreading the normally narrowband information signal over a relatively wide band of frequencies. This allows the communications to be more immune to noise and interference from RF sources such as pagers, cellular phones, and multipath. The system does not require an FCC license to

operate, and can be used for transmission distances of one to 16 km or more using inexpensive antennae, with low power consumption.

Infrared transmission may be used for in-field communication where line-of-sight communication is possible. These devices utilize an infrared light source of a few milliwatts of power to transmit digitally coded pulses of light to a remote detector. Infrared transmission units are relatively low cost and require no licensing, but are limited to distances of 1 to 2 km (Duke et al., 1990).

Soil Moisture Sensors for Irrigation Control

The soil moisture sensor is a critical component of closed-loop irrigation control systems and needs to be reliable, low-priced, require minimum maintenance, and be easily adaptable for automation. For a sensor to provide the input needed for controlling high-frequency irrigation, it must respond rapidly to changes in soil water. Extensive literature reviews on soil moisture sensors are presented by Yoder et al. (1998), Phene et al. (1990), Gardner (1986), Schmugge et al. (1980), and others.

Most soil sensor based irrigation control systems developed in the past have not been widely adopted because of the absence of affordable and reliable soil sensors that can be simply connected to irrigation controllers in sufficient numbers (Meron et al., 1995). Tensiometers equipped with pressure transducers and electrical resistance sensors, meet the cost and reliability requirements and have been used in many automatic systems.

Tensiometers, however, require considerable time for preparation, installation, periodic servicing, and removal from the field. Interpretations of the sensed matric potential may be confounded by poor soil contact, leaks, and a limited range of soil water

contents that can be measured by tensiometers (Martin et al., 1990). Maintenance free operation for extended periods is critical if a sensor is to be used in automatic irrigation control (Thomson and Armstrong, 1987).

Electrical resistance soil moisture sensors consist of two electrodes embedded in a porous matrix or block that is buried in the soil. Once in hydraulic contact with the soil, the block absorbs or releases water in response to soil matric potential gradients until equilibrium is reached. Electrically, the sensor consists of a relatively conductive liquid (soil solution) interspersed within virtually non-conductive solid and gaseous phases.

Wires are run from the electrodes to the surface so that the electrical resistance between the electrodes can be read externally. A fixed AC or DC voltage is passed through the block and a bridge is used to make the measurement. The measured voltage output is a function of the electrical resistance of the material between the electrodes, which in turn is a function of the soil water potential with which the block is at equilibrium. The electrical resistance of the block increases as the soil water potential decreases (becomes more negative).

The use of DC voltages to excite electrical resistance devices may result in "polarization" of the sensor over time by causing the migration of cations or anions to the electrodes. Polarization effects create distorted results and probe deterioration. To prevent polarization, resistance blocks should be excited using high frequency AC voltages, or DC excitation should be used only for very short duration measurements.

The voltage to the electrodes can be induced and measured manually with the aid of a special ohmmeter provided by the manufacturer, or by using a data logger. This

characteristic, and the fact that unlike tensiometers electrical resistance sensors do not require periodic maintenance, makes them well suited for the automation of irrigation control systems. Another advantage of electrical resistance sensors over tensiometers is that in freezing climates tensiometers can be damaged if exposed to freezing temperatures, while low temperatures do not damage electrical resistance sensors.

Since electrical resistance is also affected by temperature, a correction factor is usually required for greater accuracy. The salinity of the soil solution may also affect the conductivity within the block. The use of gypsum in the porous matrix buffers the effect of the salinity of the solution on the electrical resistance.

Although resistance type devices can be constructed of gypsum, nylon, fiberglass, and other materials, the most common electrical resistance soil moisture sensors used are the gypsum block and the granular matrix sensor (GMS). Gypsum blocks have the advantage of being inexpensive, allowing many replicates, but have two primary limitations for irrigation scheduling: (1) non-linear sensitivity limits the sensing mostly to the moderate to drier end of the soil water range, and (2) the measurement is not stable with time because of the gradual disintegration of the gypsum under irrigated conditions (Phene et al., 1990).

Gypsum has physical characteristics much like very heavy clay, in which small pores do not begin to lose water until a soil water potential of –30 kPa is observed. For sandy soils, more than half of the water available to the crop has already been depleted at this tension. Thus, gypsum blocks are not the instruments of choice for coarse textured soils.

The relationship between sensor resistance and soil matric potential in gypsum blocks varies not only from block to block, but also for each block over time, requiring individual block calibration. The block eventually dissolves completely into the soil solution. The time required for this to take place may be a year or more, depending on the soil moisture conditions.

Gypsum blocks also exhibit hysteresis, with more resistance to wetting or drying (or vice versa) at a set soil water potential. The sensitivity in the dry range is usually very flat (a large change in dryness is reflected by small changes in measured resistance). Upon drying, the block may become uncoupled from the soil solution.

The granular matrix sensor (GMS) marketed as the Watermark® sensor (Irrometer Co., Riverside, CA), consists of two concentric electrodes embedded in a particle matrix. The particle matrix has a consistency similar to fine sand and is held in place by a synthetic porous membrane, protected by a stainless steel exterior. A reservoir of gypsum is included in the matrix to minimize the effect of salinity (Thomson and Armstrong, 1987).

Granular matrix sensor technology reduces the problems inherent in gypsum blocks by use of a mostly insoluble porous matrix. Pore sizes in the sensor matrix are larger than those of the gypsum block allowing more sensitivity in the wet range of soil moisture. Therefore, this sensor may be used in coarse textured soils.

Granular matrix sensors such as the Watermark® have high potential use in automatic irrigation control systems because they are low cost, do not require periodic maintenance during the growing season, operate in the 0 to -200 kPa range, and can be

easily interfaced with electronic data gathering devices (Pogue and Kline, 1995). Data acquisition with Watermark® sensors can be taken from remote sites by using electrical wires. Therefore, the plants and soil at the measurement site remain undisturbed (Eldredge et al., 1993). Yoder et al. (1998) compared the performance of eight different soil water sensors and reported that the GMS was one of the four sensors that performed best when accuracy, reliability, durability, and installation factors were considered.

Thomson and Armstrong (1987) reported the following calibration equation for the Watermark® sensor:

$$SWP = \frac{-R}{0.01306*[1.062*(34.21 - T + 0.1060*T^2) - R]}$$

Where SWP is the soil water potential in kPa, R is the resistance of the Watermark® in kOhms, and T is the average soil temperature in degrees Celsius.

A different calibration equation was used by Shock et al. (1996) to control 42 drip irrigation systems using over 1000 Watermark® model 200SS sensors:

$$SWP = \frac{-(2.678 + 0.003892*R)}{(1 - 0.01201*T)}$$

Where the resistance of the Watermark® is given in Ohms, and SWP and T units are given in kPa and degrees Celsius, respectively.

Bausch and Bernard (1996) compared soil water potential calculated from measured Watermark® sensor resistance and soil temperature using both equations, with soil water potential measured by tensiometers. They determined that at the lower soil water potential

limit the equation developed by Shock et al. (1996) performed better than the equation reported by Thomson and Armstrong (1987).

Since soil water sensors used in irrigation control systems make point measurements, averaging of the readings of several sensors is needed to overcome sensor variability, spatial variability of soil characteristics and plant population, and diversions in water front advance because of cracks and macropores. According to Levin et al. (1985), the success of a feedback control system based on soil water measurements depends upon identifying the optimal soil water potential, and the distance and the depth of the sensors relative to the water source.

The soil sensor should be placed midway in the root zone, such that the majority of the root zone is never allowed to dry beyond the soil matric potential threshold before the sensor detects the drying trend and triggers the irrigation. When using a single sensor per location the wetted volume is dictated by the position of the sensor. For a given soil matric potential threshold, location of the sensor (and hence size of wetted volume) is the primary factor determining the irrigation frequency. Two sensors may also be used, with the first sensor placed in the root zone and actuating the start of the water application. The second sensor, located at the lower limit of the root zone, triggers the closing of the water flow (Meron, 1992).

Site-Specific Irrigation Control

Precision agriculture is defined as a management strategy that uses information technologies to bring data from multiple sources to bear on decisions associated with crop production. The precision agriculture concept implies a spatially variable application

of inputs in order to maximize the profits and improve the environmental sustainability of agriculture. Precision agriculture has basically three components: capture of data at an appropriate scale and frequency, interpretation and analysis of those data, and implementation of a management response at an appropriate scale and time.

According to Sadler et al. (2000), variations in water availability across a field because of different soil characteristics may cause farmers to: 1) ensure that areas with the smallest water holding capacity receive adequate water, 2) manage the whole field based on average soil water conditions, or 3) limit water application to avoid overirrigating the wettest areas. All of these scenarios will cause overirrigation or underirrigation of other areas due to the inability of current irrigation systems to differentially apply irrigation water based on soil and plant factors within a single irrigated field. Chemical leaching below the root zone, surface runoff, or potential yield decreases in particular areas can occur under each management strategy.

Heterogeneity also occurs when multiple crops are planted in the same field, requiring the crop areas to be treated as separate fields, and a unique irrigation schedule to be maintained for each crop. It is necessary to resolve conflicts that arise if both crops need irrigation at the same time and the irrigation manager must determine the appropriate irrigation sequence (Martin et al., 1990).

Site-specific irrigation management will most likely be economical for crops where yield and quality are highly water sensitive and the crop price structure is heavily dependent upon crop quality. A field study carried out by King et al. (2002) showed that site-specific irrigation management on potatoes increased marketable yield and gross

income relative to conventional uniform irrigation management. A gross income increase of $165/ha (about 4.5% increase) was observed when site-specific irrigation was applied. However, according to the same authors, attaining greater net return will depend upon the cost and useful life of the equipment required for site-specific irrigation as well as the operational costs.

With respect to irrigation, the precision agriculture objectives of increasing productivity while decreasing production costs and minimizing environmental impacts by applying site-specific input amounts can be accomplished using spatially-variable irrigation systems. Most spatially variable irrigation systems previously developed have used self-propelled irrigation systems, such as center-pivots or linear moves, as the platform on which sensing and control takes place.

Infrared plant canopy temperature sensors or remote sensing of soil water can be installed in the system laterals, providing feedback control over a large area of the field for fixed application rate systems, or for spatially variable irrigation systems. Global Positioning System technology can be used to apply variable irrigations depths, according to recorded data of soil properties or crop yields (McCann et al., 1997; Camp and Sadler, 1998).

Fraisse et al. (1995a and 1995b) developed a system that used the concept of pulse irrigation, in which solenoid valves were used to control the flow through sprinkler heads. Different application rates were obtained by pulsing the flow and varying the pulse cycle.

Another alternative was presented by Omary et al. (1997), who developed an automated system for center pivots that enabled variable application depths within 9-m long segments at a given speed. The system used three manifolds per segment that could be operated individually or in various combinations to provide eight different application rates at any given tower velocity.

McCann et al. (1997) developed a control system for center pivots and linear move systems that enabled spatially varied water application along the lateral in a stepwise manner, using electric solenoid valves and control modules to operate multiple sprinklers with different nozzle sizes. Signals to the control modules were transmitted along a single cable by a microprocessor according to the position of the irrigation system relative to a target application map.

Buchleiter et al. (1995) developed a spatially variable application system for a linear move system that used computerized control to vary the travel speed, and hence application depth, in the direction of travel. Variable water application in each half-span along the lateral was achieved by pulsing the flow to individual manifolds with an auxiliary controller interfaced with the primary control panel.

Several other research studies have been carried out in the past decade using real-time spatially variable irrigation for self-propelled systems (Sadler et al., 1996; Wall et al., 1996; King et al., 1999, Sadler et al., 2000). Application of these systems to commercially available equipment is still limited due to the costs involved, but may be feasible for high value crops in the near future.

Spatially variable control of fixed irrigation systems such as solid set sprinkler and microirrigation requires a network capable of controlling a large number of sensors and valves, in order to control irrigation of small areas in the field. This can be achieved by using centralized control or distributed control. Centralized control, with sensors and actuators in the field and the controller in a central building, requiring separate wires to connect them, may be expensive and difficult to maintain in a lightning environment, especially for site-specific irrigation control in large irrigated fields.

Distributed control systems, on the other hand, have valves and controllers close to the crop. The advantages of distributed control are low wiring and piping costs, and easier installation and maintenance. However, distributed control requires more control units in the field, which is only affordable with low cost networking and sensing/actuating devices (Torre-Neto et al., 2000).

Cromer et al. (1989) presented a distributed control and monitoring system for subsurface irrigation and drainage that used low-cost microcontrollers connected to a central computer. The central microcomputer controlled the irrigation and drainage system using inputs from a crop growth/water management simulation model. Remote field controllers (slave controllers) measured soil water levels and controlled pumps and valves to maintain the desired soil water level.

Torre-Neto et al. (2000) developed a closed-loop site-specific irrigation control system for microirrigation, based upon a distributed network of sensors and control valves. The system had one embedded circuit for each sensor/actuator element that carried out the digital conversion at the point of sensing or actuation, and enabled the

communication with the controller using a single cable, a multipoint serial data communication bus based on the RS-485 standard.

The hydraulic control of variable water application systems is another issue that may require a benefit-to-cost analysis compared to conventional systems. Supplying water at constant pressure to a system that may change water demand several times a day may require variable-rate pumps, multiple pumps linked in parallel, or a constant-rate pump with a recirculating by-pass. All of these solutions increase initial and operating costs compared to conventional systems.

Designing a hydraulic system capable of irrigating all specific zones of the field at the same time increases pumping and pipeline costs significantly. There are usually restrictions on available pumping capacity, so not all zones can be irrigated when needed. In this case the irrigation control system should be able to sequence irrigation applications according to a priority rank among the zones.

The irrigator should set the priority of each zone based on economic factors, crop sensitivity to water stress, etc.. When the flow rate required by the zones that need to be irrigated exceeds the pump capacity, the zones with the highest priorities are irrigated first. The zones with lower priorities are irrigated after the irrigation requirements are met for zones with higher priorities.

It may also happen that water resources are not sufficient to meet the evapotranspiration demand of all zones on a daily basis. Priority scheduling guarantees that in the zones with highest profitability (highest priority) irrigation demand is met. The remaining water resources are applied to the zones with lower profitability.

CHAPTER 3

SYSTEM DEVELOPMENT

The primary objective of this research was to develop a distributed irrigation control system for site-specific irrigation of fixed irrigation systems. For distributed control, each irrigation zone of a field should be managed autonomously by one stand-alone irrigation controller. Each controller should maintain the soil water potential in the crop root zone between field capacity and a threshold level set by the user.

Since multiple distributed controllers are required to accomplish the site-specific irrigation of fixed irrigation systems, the hardware needed to build each individual controller must be low cost and low maintenance. The controllers should also work independently from external power sources, without hard-wire connections among control units. Therefore, each controller must have a low power requirement with an autonomous power source.

To optimize the hydraulic design of the irrigation system, a distributed control system should be designed to irrigate only part of the field at a given time. If the flow rate for multiple zones being irrigated simultaneously is greater than the capacity of the irrigation system, the control system must sequence irrigation among zones according to a priority rank. Since there are no hard-wire connections among distributed control units, wireless communication among the controllers is also necessary.

A distributed control system reduces wiring costs significantly compared to centralized control systems. If the irrigation controllers are low cost, a distributed control system should be more affordable than a centralized control system.

Besides the lower cost, a distributed irrigation control system presents other advantages over a centralized control system. A distributed control system is simpler, requiring less time and less technical expertise for installation and maintenance. Compared to centralized control, the risk of failure caused by mechanical damage or lightning strikes is distributed over a field; such events would affect the irrigation of just part of the field, instead of the whole system.

A distributed control system is also more flexible than centralized control systems. Adding a new irrigated zone to the system is easy and does not require changing the control system program, adding new hardware to a central control, or burying additional cable, all of which are required for modifying centralized control.

Development of a Distributed Control System

A distributed control system was developed to provide autonomous irrigation control in multiple site-specific zones. Each individual controller operates with a microprocessor that receives data from soil water potential sensors. The controller is programmed to sequence irrigation of individual zones according to a preset priority ranking.

Site-Specific Irrigation Control

For a site-specific irrigation system a field will be divided into management zones prior to the installation of the irrigation system and the control system. The size and shape of each management zone should be based on soil water holding capacity, soil fertility, crop, etc.. A distributed control system will perform the site-specific irrigation by varying the amount of water and the time when irrigation is applied according to the

needs of the crop in each management zone. One stand-alone irrigation controller will manage the irrigation in each management zone.

To maintain the soil water potential (SWP) in the root zone close to a preset level, each irrigation controller will need feedback control from SWP measurements. The system will operate with a simple water balance concept; water removed from the controlled portion of the root zone will cause the soil water potential to decrease slightly below the chosen threshold level, triggering an irrigation event, which will last until the soil water potential increases above a set level. The SWP threshold will be set by the user according to the crop, growth stage, and soil characteristics of each zone of the field.

Soil Water Sensors

The irrigation controller developed in this project is programmed to measure the soil water potential, and make a decision every 15 min about starting or stopping irrigation. This sampling interval was chosen because the system that is being controlled (the plant root/soil system) does not change quickly, and a sampling interval of 15 min is adequate for detecting soil water potential changes, even for high frequency irrigation (Wessels et al., 1995).

Each irrigation controller monitors three soil water potential sensors. That number of sensors was chosen to keep the cost of the system low while providing enough data for reliable irrigation decisions. When two of the sensors indicate that the soil water potential in the root zone is more negative than the threshold set by the user, the irrigation controller opens a solenoid valve, starting the irrigation of that zone. Irrigation continues

until two of the soil sensors indicate that the soil water potential exceeds the threshold level.

After sensor measurements and the irrigation decision are made, the controller stores date, time, soil temperature, soil water potential, hydraulic pressure, and valve status data. Data are stored in non-volatile memory to prevent loss if a power failure occurs. The user can download data to check for proper sensor or controller operation.

If soil water potential measurements are out of a predetermined range, the controller activates an alarm for a few seconds every minute. The sound of the alarm warns any person near the controller about improper sensor operation, and that downloading controller data is recommended to identify which sensor has a problem.

Priority Scheduling

Irrigation of site-specific management zones should occur according to a priority rank when more than one zone requires irrigation at the same time and the water demand is greater than the water supply pump capacity. Priority scheduling controls the access to water for each management zone based on a priority rank set by the user and by hydraulic pressure measurements. After the control system is installed, the user enters the priority of each controller/zone, which is decided in advance. Factors such as the expected benefit-to-cost ratio of irrigation for each zone, or crop sensitivity to water stress (for multiple crop fields) may be used as the criteria for determining the zone priority.

Most irrigation systems are designed to irrigate only part of a field at any time because of water supply or system hydraulic limitations. When using constant speed pumps, which is the most common situation, increasing the discharge causes the

hydraulic pressure to decrease. In this case only a certain number of zones, which will vary according to the irrigation system, can be irrigated at the same time.

The clock of all individual irrigation controllers in a field are synchronized, measuring the soil water potential and opening or closing irrigation valves at constant intervals and at the same time. If the water demand is high, i.e., multiple controllers require water, the irrigation system pressure may drop below the operating pressure required by the emitters. In this case zones with lower priority will not be irrigated, allowing the zones with higher priority to be irrigated first.

The irrigation controllers perform the priority scheduling by measuring the hydraulic pressure in the irrigation system after the irrigation event starts. If the pressure is lower than a certain value set by the user, the controller closes the solenoid valve, interrupting the irrigation until the next soil water status measurement. The higher the priority set by the user for a particular controller, the longer the elapsed time before pressure is checked. This allows the controllers with highest priority to continue irrigating, since the pressure in the pipeline will be adequate when it is checked.

The irrigation controller program was written such that the lower the priority number assigned to a controller, the lower the priority for the set irrigated by that controller. For example, a controller with priority one corresponds to the lowest priority possible. When setting controller priorities it is important to ensure that the irrigation system can irrigate all zones with the same priority at the proper operating pressure.

The priority scheduling approach guarantees that the management zones with higher profitability are irrigated first and get sufficient water. The remaining water

resources are allocated to zones with lower profitability. The capacity of the irrigation system to meet the evapotranspiration demand of the irrigated field will determine if certain zones with low priority will suffer water stress. If water resources or the irrigation system design do not meet the crop water requirements on a daily basis, some zones of the field may be underirrigated and suffer water stress.

Hydraulic Pressure

The distributed control system uses the hydraulic pressure in the irrigation system as the communication bus to transmit information about which controller(s) can irrigate at a given time. Some variables that affect the irrigation system hydraulics such as the pipeline length, materials used, and topography need to be considered when transmitting information by the hydraulic pressure.

The pipeline length affects the volume of water stored in the irrigation system and the time that it may take for the system pressure to become stable after irrigation of a management zone stops or starts. Large irrigation systems may require longer delay times between valve opening and pressure measurement by the irrigation controllers.

The material used in the pipeline also affects the pressure response time to flow rate variations in the system. For pipelines made of non-flexible materials such as galvanized steel, the pressure shows quicker response to flow rate variations than for flexible materials such as polyethylene. Control of water hammer using proper valves is also recommended to guarantee that pressure measurements represent the actual system pressure.

Variation in the field topography should also be taken into account to ensure that the pressure measured by each controller reflects the real situation in the system; otherwise the zones at lower elevations in the field will always show higher pressure values. When installing the controller the user will enter the difference in elevation between the point where the pipeline pressure is measured by that particular controller, and the point where the controller at the highest elevation measures the pressure. The difference in elevation is subtracted from the measured pressure before the controller makes comparisons with the pressure threshold.

The difference in elevation can also be measured using the irrigation controllers. In this case the irrigation pipeline is filled with the water until it reaches the pressure sensor of the controller at the highest elevation, maintaining zero pressure on it. The pressure head measurement made by each individual controller is then entered as the difference in elevation for that particular controller.

To operate as designed, the irrigation system must be continuously pressurized. A pressure switch may be used to maintain the pressure in the irrigation system. The pressure switch turns on the pump when the pressure in the pipeline drops below a preset value, and turns the pump off when the pressure rises above a set point.

Controller Synchronization

Since all controllers in the field should measure soil water potential and make the initial decision about starting or stopping the irrigation at the same time, the controller clocks must remain synchronized. Differences in the frequency of the clock crystals may cause time drift among the controllers.

The system hydraulic pressure is used to synchronize the controller clocks. An irrigation timer at the main control is programmed to activate a 3-way valve periodically, releasing water from the pipeline, and dropping the hydraulic pressure to zero. The irrigation controllers are programmed to reset the clocks whenever the pressure in the pipeline drops below a preset value. Topographic differences in the field must be taken into account to guarantee that all controller clocks are synchronized when the pressure in the pipeline is reestablished; the difference in elevation above the irrigation controller is used to make that compensation.

Power Consumption

Since there are no hard-wire connections between control units, each controller must be autonomously powered. Solar power is the primary power source to operate the irrigation controller. A battery is used to balance the differences between available power and demand on an hour-to-hour basis.

To optimize the power supply components, all hardware used to build the controller, including sensors and actuators, are designed to use the least amount of power possible. Sensors are powered only for a short time, when measurements are needed. The controller uses a latching solenoid valve to control the water flow. This valve uses DC power and only requires a 100 ms current pulse to be opened or closed, thus saving a significant amount of power compared to the 24-VAC solenoid valve normally used for irrigation control.

System Hardware

Overview

The hardware used in the irrigation controller was designed considering adequacy for the tasks involved and cost. Since the sampling interval could be as long as several minutes, the speed of the microcontroller was not a concern. Low power consumption and low cost were the most important features for the electronic devices, sensors, and actuators.

Each irrigation controller consists of a microcontroller, clock and data storage devices, an A/D converter, and associated discrete components (transistors, capacitors, resistors, etc.). The controller is powered by a solar panel and a battery, with a battery charge regulator incorporated into the controller circuit.

A block diagram of the irrigation controller hardware is shown in Figure 1. The irrigation controller circuit and electronic components are shown in Figure 2. The irrigation controller monitors three soil water potential sensors, one soil temperature sensor, and one pressure transducer. The controller uses a latching solenoid valve and an alarm as actuators. A complete description of the irrigation controller parts, costs, connections, and sensor calibrations are shown in Appendix A.

The irrigation controller uses a BASIC Stamp 2 microcontroller (Parallax, Inc., Rocklin, CA) to control the irrigation of each specific zone in the field. This microcontroller was selected because it is a low-cost device that meets the speed and power requirements of the irrigation controller, and facilitates the integration into a custom-built circuit. The microcontroller is the master device that is programmed to keep

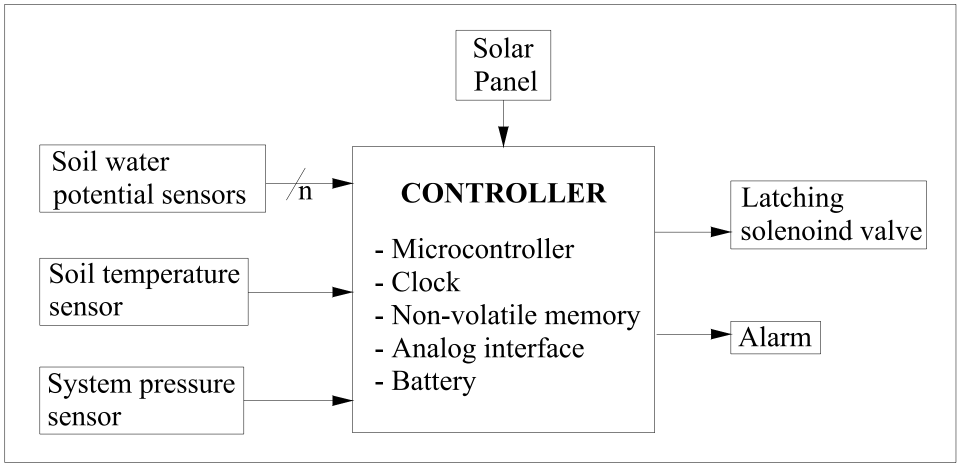


Figure 1. Block diagram of irrigation controller developed in this study.

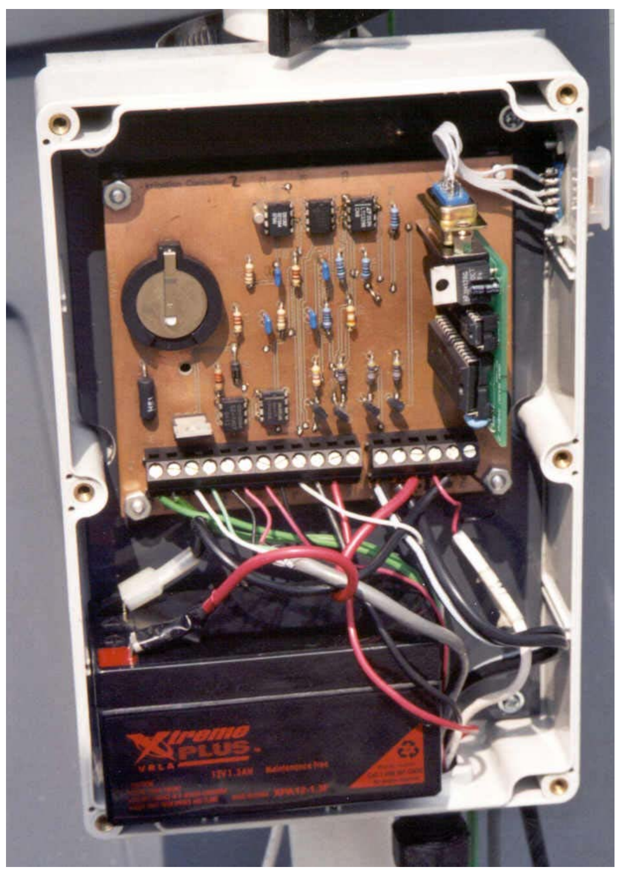


Figure 2. Irrigation controller developed in this study.

time, communicate with data storage devices, control the battery-charging modes, read sensors, and control the actuators.

The sensor used to monitor the soil water potential in the root zone was chosen to meet important requirements of an automated irrigation control system such as accuracy, reliability, durability, low maintenance, ease of interfacing with data acquisition systems, and low cost. Although no existing soil sensor scores high in all those requirements, the Watermark® sensor Model 200SS (Irrometer Co., Riverside, CA) was selected as the best option for being easy to install and interface with data acquisition systems. It is also relatively maintenance free, and presents the best combination of price, accuracy, and reliability among soil sensors on the market.

Electrical Power Requirements

The irrigation control system was designed to use autonomous, distributed irrigation controllers that operate independently of AC power and without wires installed between individual irrigation controllers. Therefore, one inherent system requirement is the use of an autonomous DC power source to run the controllers, such as batteries and solar panels. The controller was designed to operate with a 12-VDC power source, as required by the latching solenoid valve.

In order to determine power requirements for the irrigation controller, the entire circuit (Basic Stamp 2 microcontroller interfaced with sensors, clock, A/D converter, EEPROM, and actuators) power consumption was measured. The current drawn by the irrigation controller when running (sensor readings, data storage, etc.) and in stand-by mode was 14 mA and 10 mA, respectively.

A 5-W solar panel and a 1.2-Ah, 12-V sealed lead acid battery were used to power the irrigation controller. Details of the power supply system design are presented in Appendix A.

System Software

Overview

The irrigation controller program (*icontroller.bs2*) was written using BASIC Stamp Editor software version 1.1. The program has 480 lines of code and uses the whole memory capacity of the microcontroller. The program flowchart is shown in Figure 3, and the program code is presented in Appendix B.

The program contains a main loop and several subroutines that enable the microcontroller to perform the following tasks:

- Measure the soil temperature, the soil water potential, and the hydraulic pressure every 15 min;
- Compare soil water potential and pipeline pressure values to threshold values programmed by the user and stored in the EEPROM, and make a decision about opening or closing the solenoid valve;
- Store date, time, soil temperature, soil water potentials, pressure, and valve status in the controller EEPROM;
- Warn the user about sensor out-of-range readings by activating an alarm;
- Reset the controller clock if the pressure drops below a certain threshold level;
- When queried by the user, transfer data stored in the EEPROM to a computer;
- Allow the user to change the controller irrigation priority, the time when irrigation

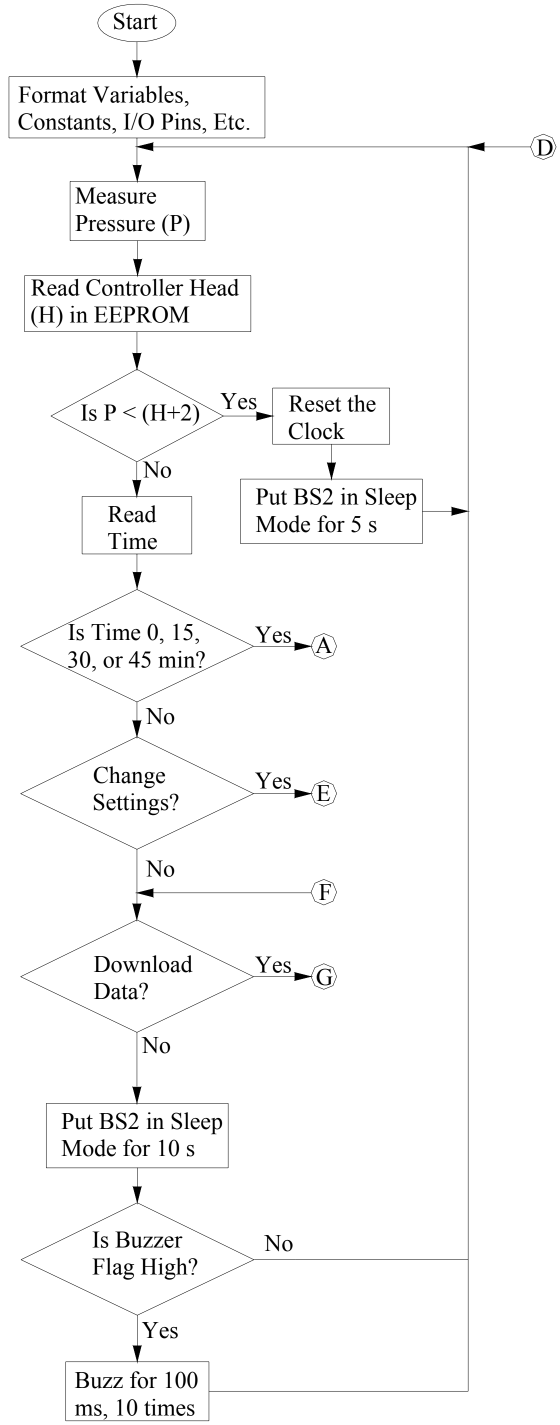


Figure 3. Flow chart for irrigation controller program *icontroller.bs2*.

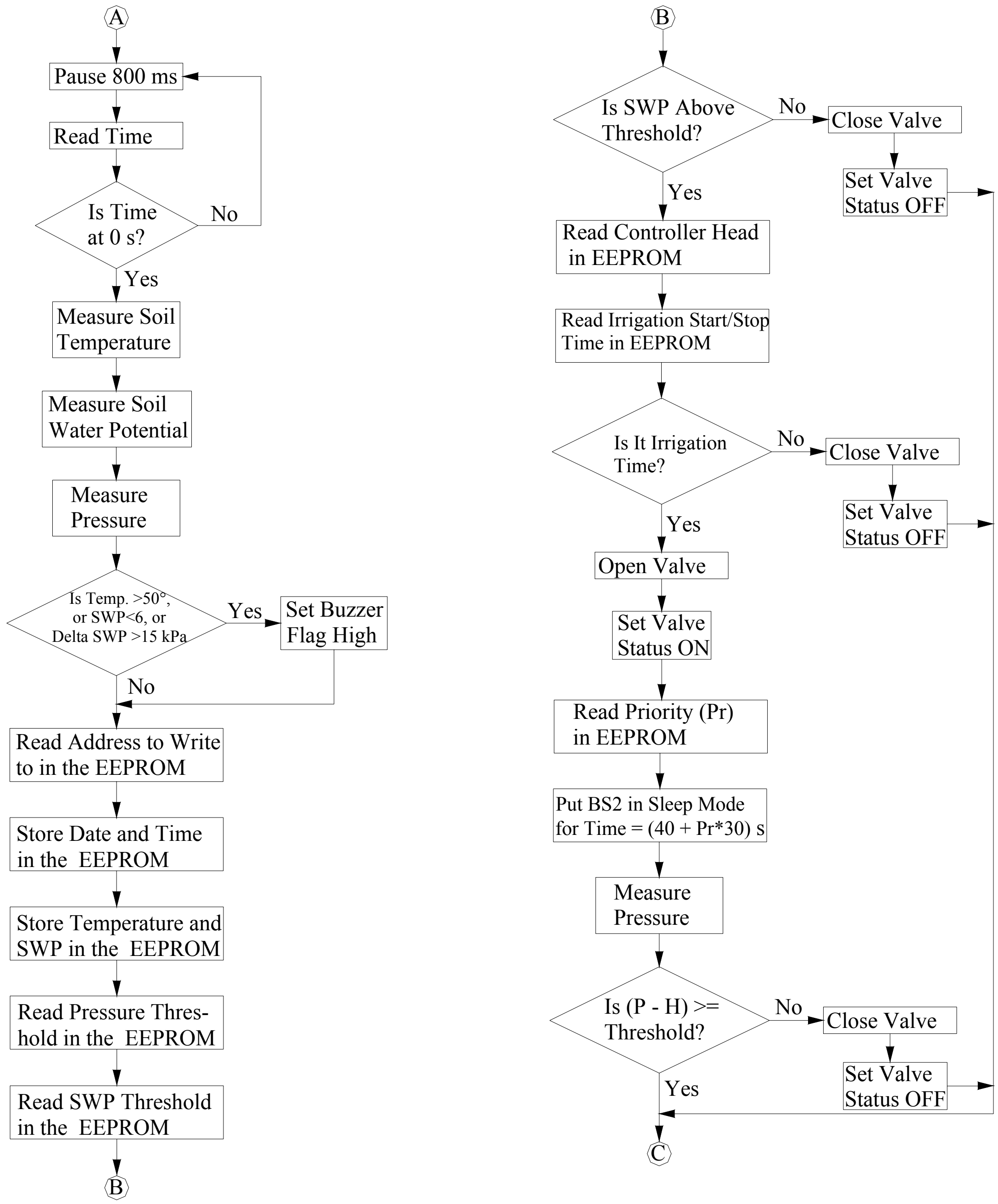


Figure 3. Continued.

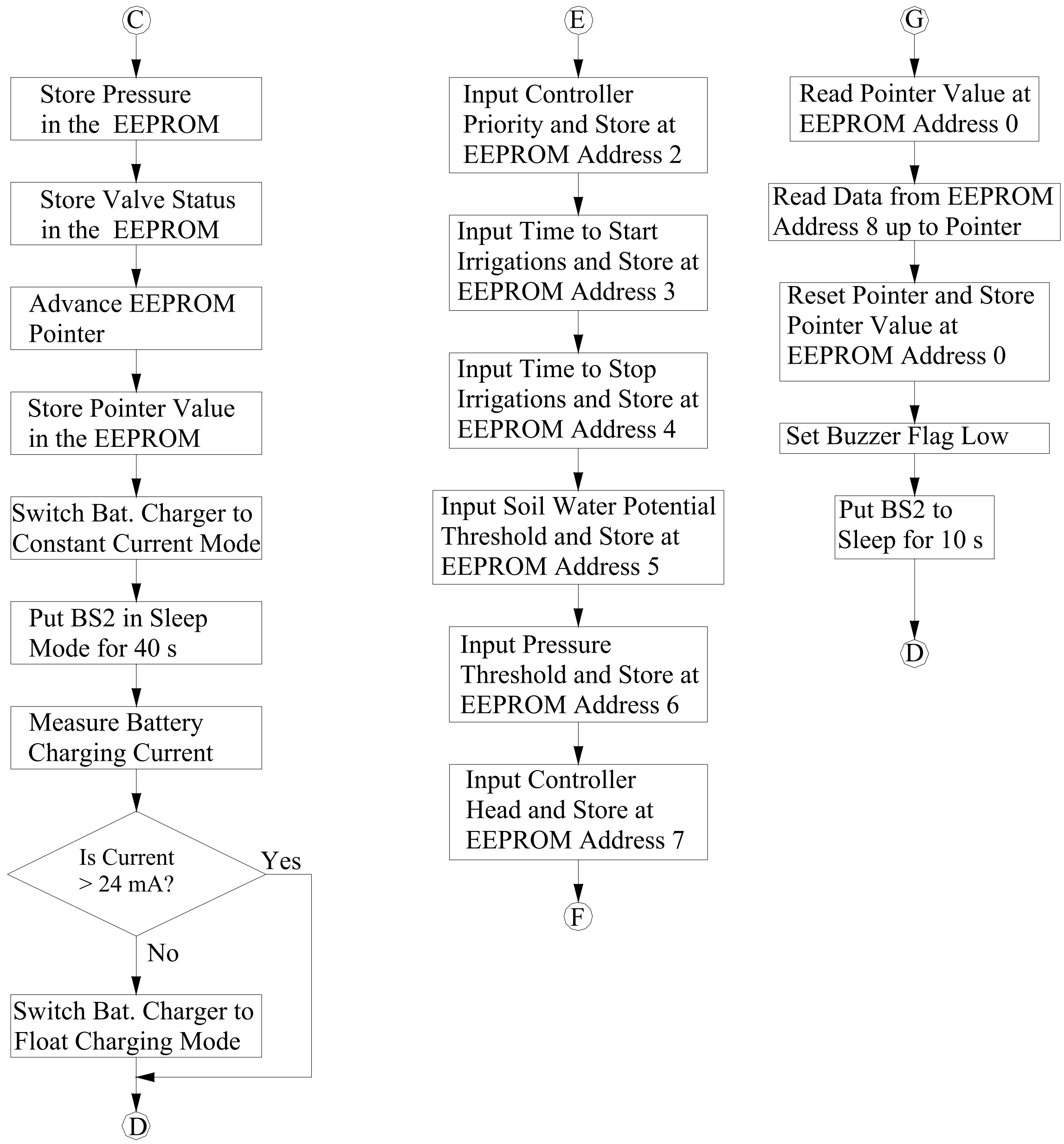


Figure 3. Continued.

is allowed to occur, the soil water potential threshold, and the pressure threshold;

- Manage the battery charging process.

Main Loop

Before the main loop begins, the program configures the variables, constants, and I/O pin assignments. The main loop is repeated two times per minute when no soil moisture readings are needed. It begins with the microcontroller measuring the hydraulic pressure in the pipeline, reading the water column head above the irrigation controller (stored in the EEPROM), and comparing both values.

If the difference between the measured pressure minus the head is smaller than a pre-determined value, the microcontroller resets the clock, goes to stand-by mode for 5 s, and then repeats the procedure. This procedure is used to reset the irrigation controller clock when the actual pressure in the pipeline goes below the pre-determined value, allowing all irrigation controllers to synchronize the clocks, with a maximum time difference of 5 s. The threshold pressure of 13.8 kPa is used for the evaluation system to guarantee that the controller clocks will be reset whenever the irrigation line is drained, even if small deviations in the pressure sensors calibration occur.

If the actual pressure in the pipeline is above the threshold, the program reads the current time from the clock. For example, if the sampling interval is set to 15 min, the program goes to a subroutine called “Stage1” when time is at 0, 15, 30, or 45 min. In the “Stage1” subroutine sensor readings and the irrigation decision takes place.

If it is not time to read the sensors, the microcontroller prompts the user for changes in the irrigation variables such as controller priority, time when irrigations are

allowed to start, time when irrigations should stop, soil water potential threshold, pressure threshold, and water column head above the controller. All these variables are set using the subroutine “Set, ” and are set the first time the controller is installed, or anytime the user needs to change them.

The microcontroller prompts the user to determine if downloading the irrigation data stored in the EEPROM is desired, which is executed by subroutine “Download.” To download data the microcontroller reads the last address written to the EEPROM (variable “Pointer”), which is stored in addresses 0 and 1 of the EEPROM. The microcontroller reads data from EEPROM address 8 to the last address written, and transfers the data to the computer, where it can be saved as a comma delimited text file using the software HyperTerminal. The microcontroller then resets the variable “Pointer”, stores the new pointer value in the EEPROM, and sets the alarm flag low.

The main loop proceeds by putting the microcontroller in power save mode for 10 s. After that, if the alarm flag is high, the microcontroller warns the user of possible sensor failures by activating the alarm. The program returns to the beginning of the main loop.

Stage1 Subroutine

This subroutine begins by pausing the microcontroller for 800 ms and reading the time from the clock. If time is not at 0 s the microcontroller executes a loop until that condition is met. This is used to synchronize sensor readings and irrigation decisions with other irrigation controllers.

The microcontroller reads soil temperature, soil water potential, and pressure sensors ("Measures" subroutine), and stores temperature and soil water potential in the EEPROM ("Store1" subroutine). In the subroutine "Measures" the alarm flag is set high if the temperature reading is higher than 50°C, or the SWP is less negative than -6 kPa, or the difference among SWP sensor readings is higher than 15 kPa.

The microcontroller reads the soil water potential threshold stored in the EEPROM and compares it to the soil moisture sensor readings. If two of the Watermark® sensors indicate that the SWP is above the threshold, the program goes to the "Stage2" subroutine. Otherwise the microcontroller closes the solenoid valve ("Closev" subroutine), stores pressure and valve status ("Store2" subroutine), and goes back to the main loop.

Stage2 Subroutine

The microcontroller reads the times when the irrigation is allowed to start and to stop; the times are stored in the EEPROM, and compared to current time. If the current time is within the allowed irrigation time, the microcontroller opens the solenoid valve. The microcontroller reads the priority stored in the EEPROM and stays in stand-by mode for a time equal to the priority number multiplied by 30 s, plus 40 s. After that time the microcontroller checks the pressure again and compares it to the threshold plus the water column head.

If the pressure is higher than the threshold value, the microcontroller stores the pressure and valve status in the EEPROM. If the pressure is lower than the threshold value, it closes the valve before storing the pressure and valve status.

Before returning to the main loop the microcontroller checks the battery charging current ("Bat" subroutine). This is done by switching to constant current/constant voltage mode and waiting 40 s for the3 current to stabilize, after which the microcontroller reads the charging current through the current sensing amplifier and the A/D converter. If the charging current is higher than 24 mA the microcontroller goes back to the main loop, otherwise it switches back to float voltage charging and then goes back to the loop.

CHAPTER 4

SYSTEM EVALUATION

The irrigation control system performance was evaluated during the summer of 2002 at the University of Tennessee Department of Biosystems Engineering and Environmental Sciences. Objectives of the evaluation were: (a) to verify the effectiveness of the irrigation controllers for maintaining the soil water potential in the root zone within a preset range; and (b) to verify the effectiveness of the priority rank approach for the simultaneous operation of several irrigation controllers under limited water conditions.

Irrigation System

The designed irrigation control system that was developed was used to schedule irrigation of 12 soil containers cultivated with Bermuda grass (*Cynodon dactylon*). Each container was filled with gravel to a depth of 0.1 m, and was filled with soil from that level to the top. The soil was taken from a Sequatchie Silt Loam Ap horizon. Each container had the soil packed manually, was irrigated to field capacity, and was fertilized with 35 g of 24-5-11 N, P_2O_5 , and K_2O. Bermuda grass sod was planted in the containers on June 21, 2002 (calendar day 172), and from that date forward irrigation was performed using drip irrigation. A hole was drilled at the bottom of the containers to allow the collection of drainage water if overirrigation occurred.

A drip irrigation system was installed to simulate the site-specific irrigation of four different zones of a field. Each irrigation zone was represented by one lateral

irrigating three containers; each line had a pressure regulator, flow meter, and flow control valve (Figure 4).

Each container was irrigated by one on-line non-compensated dripper (Netafim Button Dripper) with a nominal flow rate of 2.0 L h^{-1}, at an operating pressure of 100 kPa. Laterals were made of ¾ -in. (19 mm) diameter polyethylene tubes, and 1 ¼ -in. (32 mm) PVC pipes were used for the mainline. At the beginning of the mainline a ¾ -in. (19 mm), 155-mesh, plastic screen filter was installed.

The system was continuously pressurized by a 1/3-hp (0.25 kW) centrifugal pump, model 9K860A, from Berkeley Pumps. A pressure switch was used to turn the pump on when the pressure in the mainline dropped to 207 kPa, and to turn the pump off when the pressure reached 345 kPa. A diaphragm pressure tank (WaterAce, model RPT20H) was used to avoid frequent pump start/stop cycles. The tank air pressure was set to 193 kPa, and the drawdown capacity was 23.5 L.

A 3-way valve model Motortrol (Erie Controls, Milwaukee, WI), installed downstream of the pressure tank was used to occasionally close the water flow from the tank to the mainline and release water from the mainline back to the reservoir. This allowed the pressure in the mainline to drop to zero for some time and controller clocks to be resynchronized.

An irrigation timer (RainBird ISA408) was used to activate the valve. The timer was programmed to switch the valve every three days, for three minutes. Preliminary observation of time drift among controller clocks showed that this procedure was sufficient to keep the clocks synchronized.

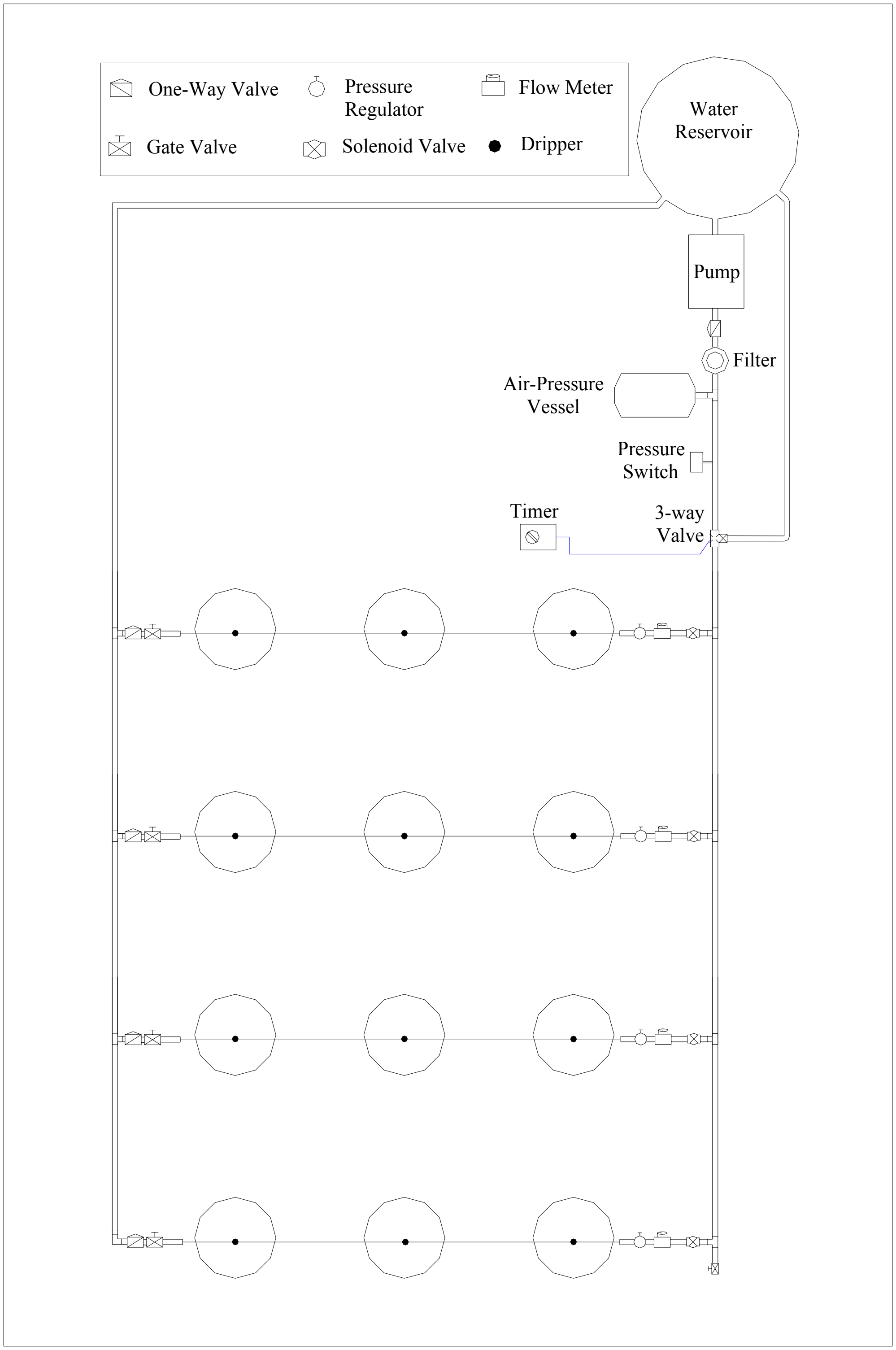


Figure 4. Schematic of the irrigation system test setup used for the distributed control system evaluation.

A pressure regulator was installed at the beginning of each lateral, ensuring a maximum line pressure of 100 kPa. A gate valve and a manometer, installed at the end of each lateral, were used to control the lateral hydraulic pressure and flow rate which was measured by a flow meter. Lateral flow rates were adjusted during operation using a flow meter to match the pump capacity, such that the pump could irrigate only one lateral at a time with proper operating pressure.

The adjusted lateral flow rate was 1.42 $m^3 h^{-1}$, for a mainline pressure (before the pressure regulator) of 200 kPa. When irrigating more than one lateral the pressure in the mainline would drop to less than 100 kPa. After the lateral flow rate was adjusted, the flow rate of each dripper was measured. The dripper average flow rate was 2.01 L h^{-1}, with a uniformity coefficient of 90%. Excess water at the end of the laterals was collected by a 1 ¼-in. (32 mm) PVC pipe and returned to the reservoir.

Irrigation Control

Each irrigation controller was designed to monitor the soil water potential of three containers using one Watermark® sensor per container (Figure 5). Before installation, the Watermark sensors were soaked in water for 24 h, according to the manufacturer's recommendation. The sensors were attached to ½-in. (12 mm) diameter, 0.3-m long PVC pipes and installed at a depth of 0.20 m, and at a lateral distance of 0.15 m from the drippers (Figure 6).

The soil temperature was monitored by the controller using a thermistor installed in the first container of each row, at the same depth as the Watermark® sensor. The hydraulic pressure sensor was installed in the mainline, immediately upstream of the flow

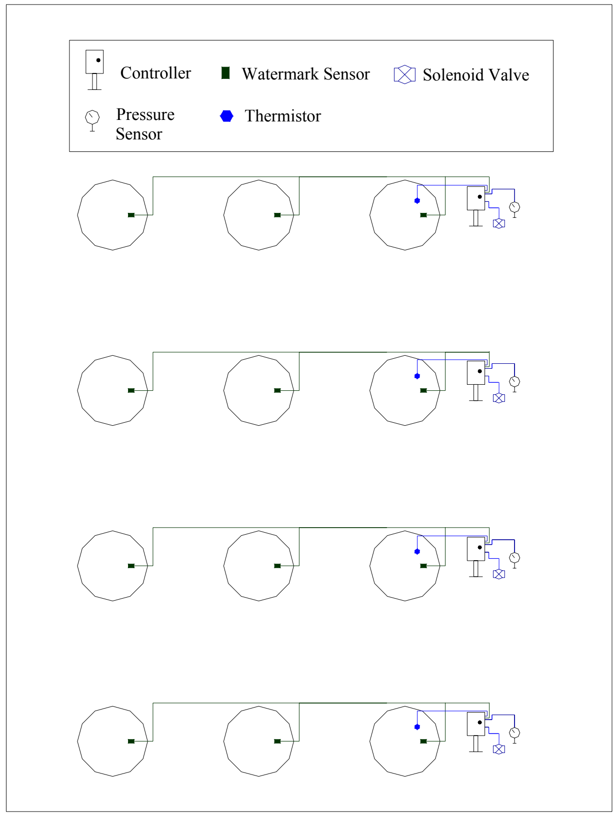


Figure 5. Irrigation control system test setup used for the distributed control system evaluation.

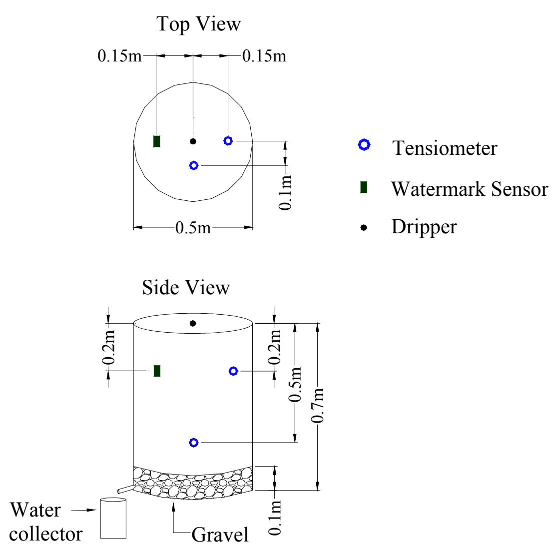


Figure 6. Schematic of the soil container and soil water sensor positions used for the system evaluation.

control valve in each lateral. Details of the irrigation control system test setup are shown in Figures 7, 8, and 9.

Using the software Stampw.exe, the *setclock.bs2* program was downloaded to the controllers and the clocks were set to the same time. The *icontroller.bs2* program was downloaded to the controllers and the irrigation variables of each controller were entered.

The irrigation controller was programmed to check the soil water potential status every 15 min. Irrigation priorities 1, 2, 3, and 4 were assigned to controllers of the first, second, third, and fourth laterals respectively. For priority 1 the controller was programmed to check the pressure 40 s after the irrigation began. For priorities 2, 3, and 4 the controller was programmed to check the pressure at 70, 100, and 130 s after the irrigation began.

The delay times were chosen based on preliminary tests performed with the irrigation system. The tests indicated that it took at least 28 s for the pressure in the mainline to decrease and to stabilize when more than one controller started irrigation simultaneously. It took at least 20 s for the pressure in the mainline to increase and stabilize when one of the controllers irrigating at the same time closed a valve.

The soil water potential threshold was set to -15 kPa. This threshold value was chosen because Watermark® sensors typically do not produce reliable measurements for soil water potentials less negative than –10 kPa. However, the SWP threshold chosen still could be considered adequate for most crops with high sensitivity to water stress, for the soil type used in the system evaluation.

Figure 7. Irrigation control system test setup.

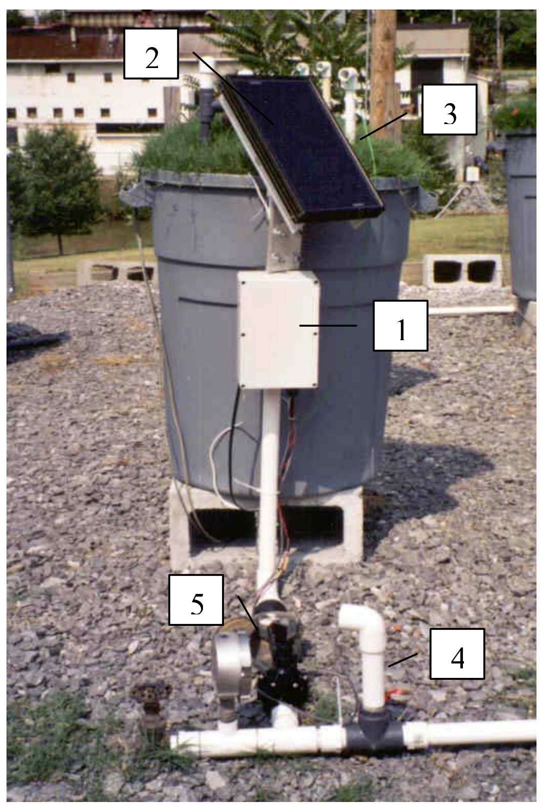


Figure 8. Detail of the irrigation control system showing: 1) irrigation controller; 2) solar panel; 3) soil water potential sensor; 4) pressure sensor; and 5) solenoid valve.

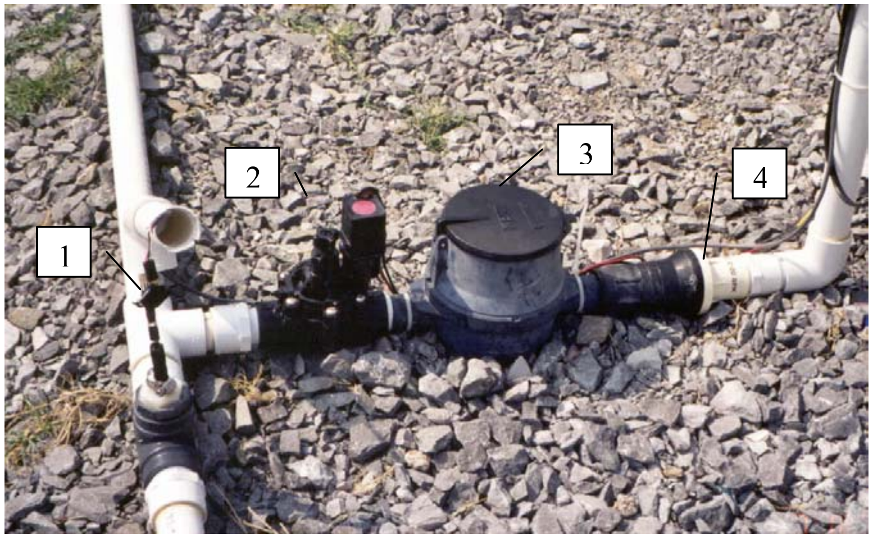


Figure 9. Detail of lateral head showing: 1) pressure sensor, 2) latching solenoid valve, 3) flow meter, and 4) pressure regulator.

The pressure threshold was set to 100 kPa due to the emitter operating pressure requirement. That pressure threshold was also adequate for the priority scheduling evaluation, since the pressure in the irrigation system would drop to less than 100 kPa when more than one zone was irrigated at the same time. Since all controllers were at the same topographic level, the water pressure head above the controllers was set to 0 m.

The containers were protected from rainfall using a transparent plastic cover every time a rainfall event occurred, except for August 25 (calendar day 237), when a 27 mm precipitation event occurred. Until August 5 (calendar day 217), the time when irrigation was allowed to start was set to 18:00 h, and the stop time was set to 24:00 h for all controllers. This was done to increase the probability of having more than one zone irrigating at the same time, thus allowing evaluation of the priority scheduling performance. From day 217 forward the allowed irrigation time was set from 7:00 h until and 24:00 h.

Soil Water Potential Monitoring System

The soil water potential in the containers was also monitored using tensiometers to verify the proper functioning of the irrigation controllers. Two tensiometers were installed in each container, at depths of 0.2 m and 0.5 m as shown in Figure 6.

The tensiometers were equipped with pressure transducers (Motorola, Inc. MPX5700DP). The pressure transducers were calibrated in the lab using the hanging water column method (Haines apparatus) (Klute, 1986). Two pressure transducers were used in the calibration procedure, with the water column ranging from 0 to 2.75 m (0 to 30 kPa).

Calibration results are shown in Figure 10. The results showed that although some variation was observed among pressure transducer measurements, a linear model represented a good fit to the data, explaining 98.5% of the variation. Variation in the experimental setup was the most probable cause of the differences observed between the measurements for the two sensors.

A Campbell Scientific 21X data logger and an AM-416 relay multiplexer were used to excite and measure output voltages from the pressure transducers. The data logger code used to read the tensiometer pressure transducers (*tensiom.dld*) is presented in Appendix B. Figure 11 shows a view of the soil container with the two tensiometers and the Watermark® sensor installed.

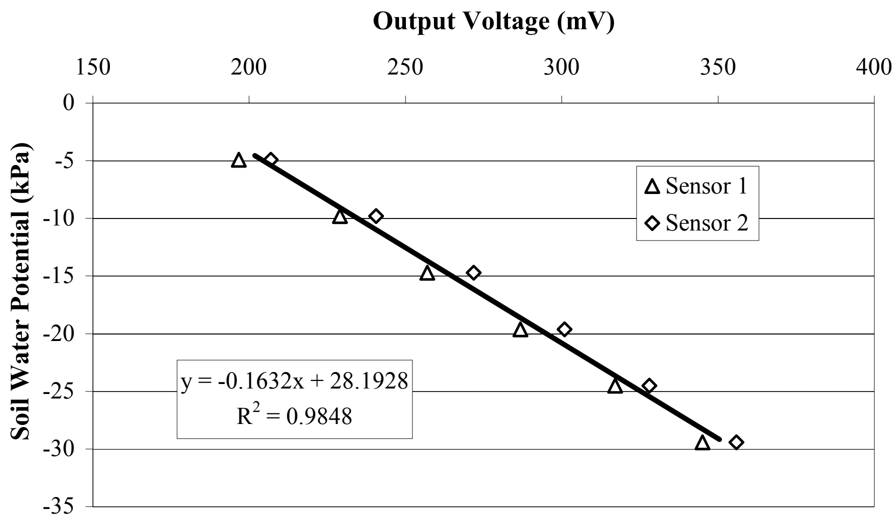


Figure 10. Calibration results obtained for the Motorola MPX5700DP pressure transducer.

Figure 11. Detail of tensiometers with pressure transducers and Watermark® sensor used for the control system evaluation.

CHAPTER 5

RESULTS AND ANALYSIS

The performance of the system was analyzed in terms of: (a) the adequate operation of the system hardware and software; (b) the control of the soil water potential in the root zone on a real-time basis; and (c) water allocation among the specific zones/controllers according to the priority rank.

The irrigation control system performance was considered satisfactory if the soil water potential in the root zone was maintained less negative than the SWP threshold plus 20% tolerance (-18 kPa) for at least 90% of the time and overirrigation or excess water did not account for more than 5% of the irrigation depth applied. The priority scheduling performance was considered effective if the zones with highest priorities were irrigated first at least 90% of the occasions when more than one zone required irrigation at the same time.

Soil Water Potential Control

The irrigation controllers proved to be effective for controlling the soil water potential in the root zone of the Bermuda grass. Figure 12 shows soil water potential changes at the 0.2-m depth, measured during calendar day 202, 2002 with three Watermark® sensors connected to irrigation controller 4. This day represented the most common situation observed during the evaluation period, with soil water extraction during the day triggering an irrigation event in the evening.

Soil water potential readings were almost constant during the night and in the morning. The Watermark® sensors responded well to soil water extraction during the

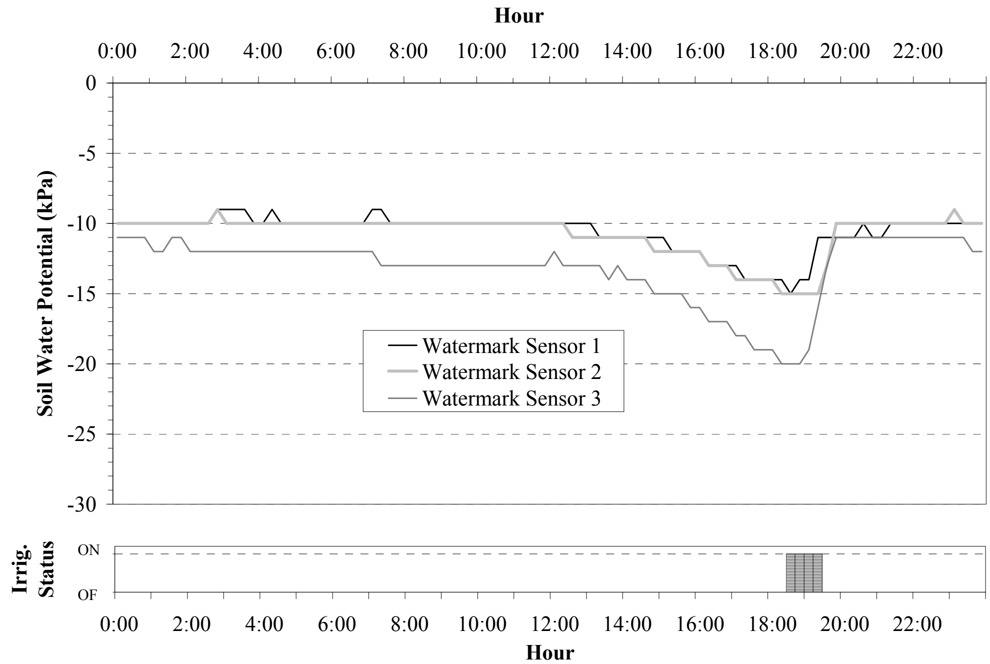


Figure 12. Soil water potential measured with irrigation controller 4 at the 0.2-m depth, on calendar day 202, 2002.

period of peak water use (12:00-18:00 h). Irrigation started at 18:30 h when Watermark® sensors 2 and 3 measured soil water potentials of –15 kPa and –20 kPa respectively. The water application stopped at 17:30 h, when sensors 1 and 2 indicated SWP values of –11 kPa and –13 kPa respectively.

The Watermark® sensors also showed quick response to the wetting front, preventing overirrigation. The results confirmed preliminary tests performed in the laboratory, which showed that when a wetting front reaches the sensor it takes only from 5 to 20 min for the SWP reading to increase from an initial value of –25 kPa to –10 kPa.

Since the irrigation controllers were programmed to switch the irrigation on or off when 2 out of 3 Watermark® sensors indicated that the soil water potential was above or

below the threshold level, the median sensor reading was the one that defined when irrigation would start or stop. For the day illustrated in Figure 12, Watermark® sensor 2 was the one that determined the beginning and the end of the irrigation event.

The changes in soil water potential in soil containers with irrigation controlled by controllers 1 to 4, for 47 days of 2002, are shown in Figures 13 to 16. Watermark® measurements are presented as median readings, while tensiometer measurements are presented as the average of the three sensor readings.

Watermark® readings at the 0.2-m depth showed that the irrigation controllers worked as expected over the entire testing period. The soil water potential in the root zone was maintained within an adequate range for plant development (between –11 kPa and –20 kPa) according to the Watermark® sensors.

From day 196 to day 217 the controllers were programmed to irrigate only between 18:00 h and 24:00 h, which may explain why Watermark® readings were more negative than the preset value during some hours of the day. After day 217, irrigation was allowed from 7:00 h to 24:00 h, and only a few Watermark® readings more negative than the SWP threshold level occurred, being caused primarily by the irrigation priority schedule. Although the daily time that irrigation was allowed never impeded any zone from being irrigated on any day, the priority schedule delayed the irrigation of the zones with low priority.

Tensiometer readings at the 0.2-m depth confirmed adequate control of the soil water potential by the irrigation controllers most of the time. Tensiometers readings often began to decrease sooner than the Watermark® sensors in response to soil water

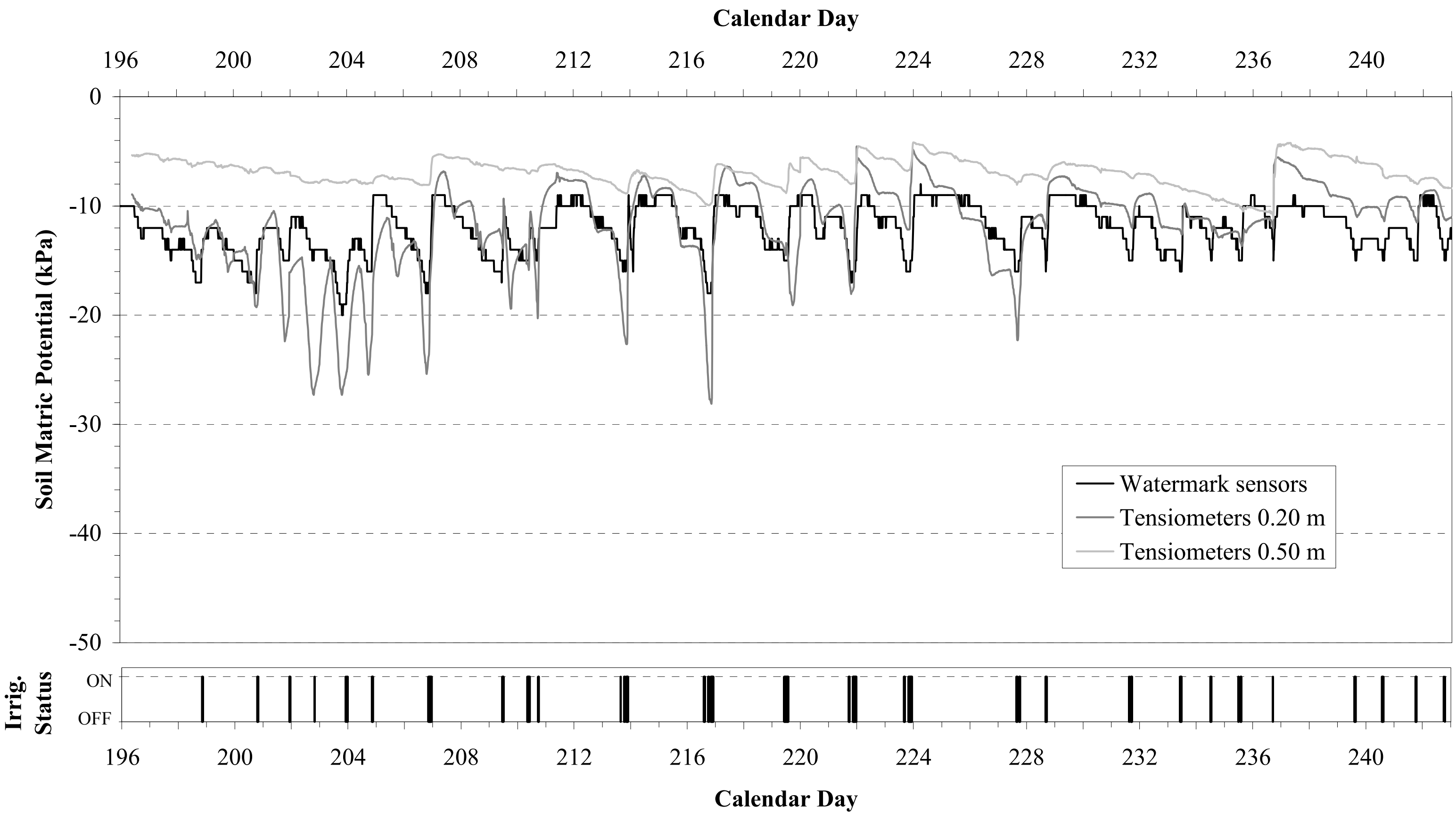


Figure 13. Soil water potential measured in containers irrigated by irrigation controller 1.

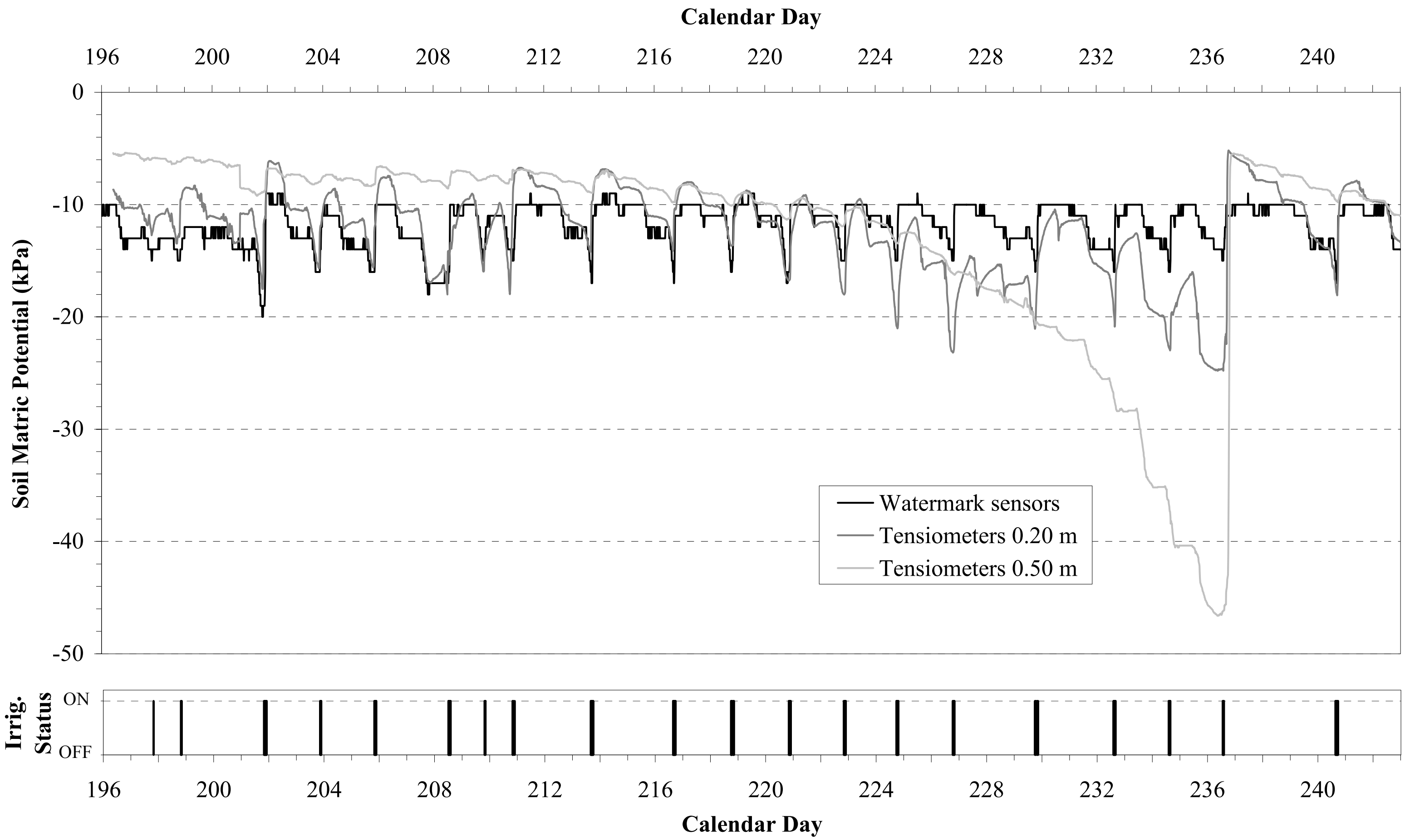


Figure 14. Soil water potential measured in containers irrigated by irrigation controller 2.

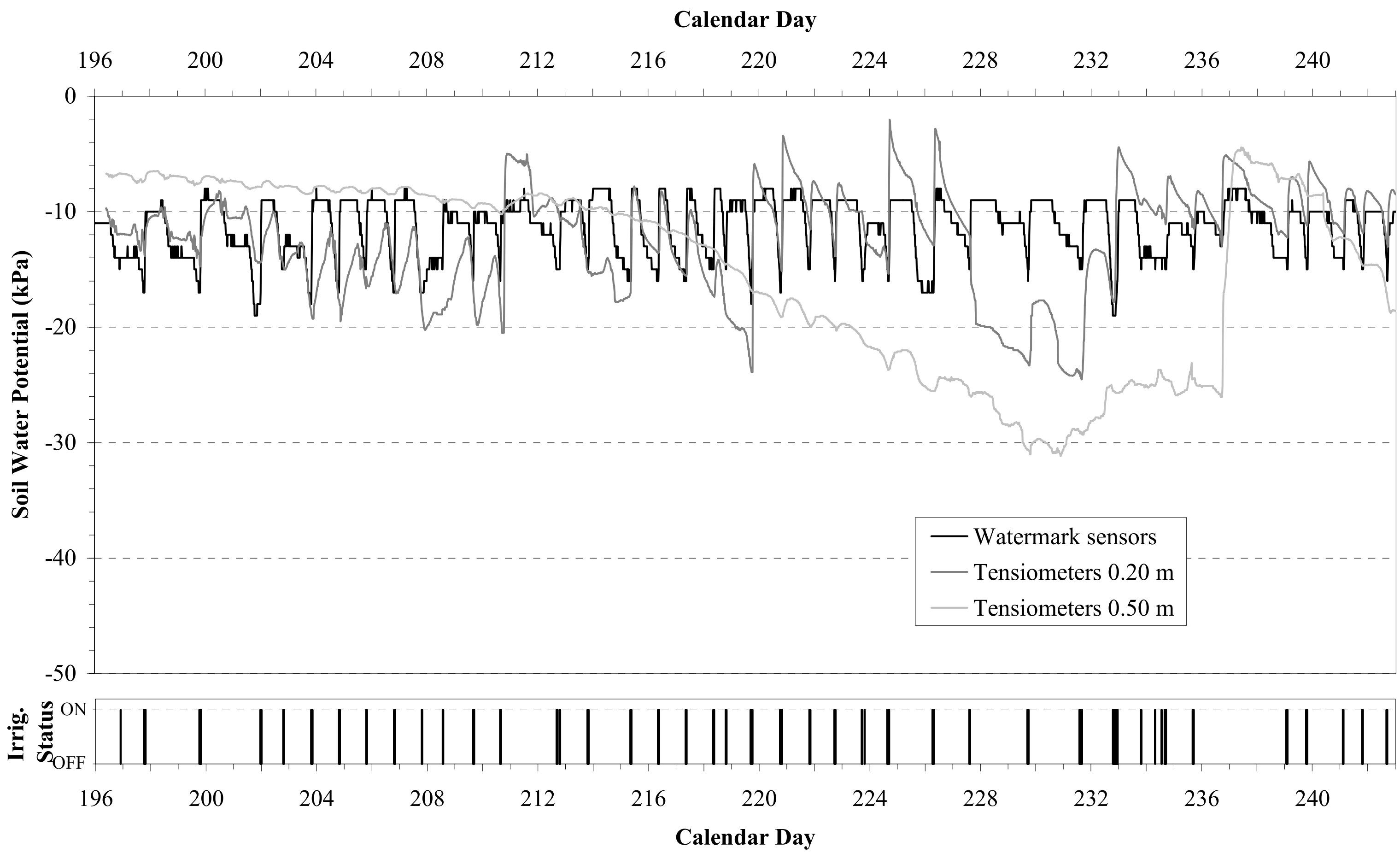


Figure 15. Soil water potential measured in containers irrigated by irrigation controller 3.

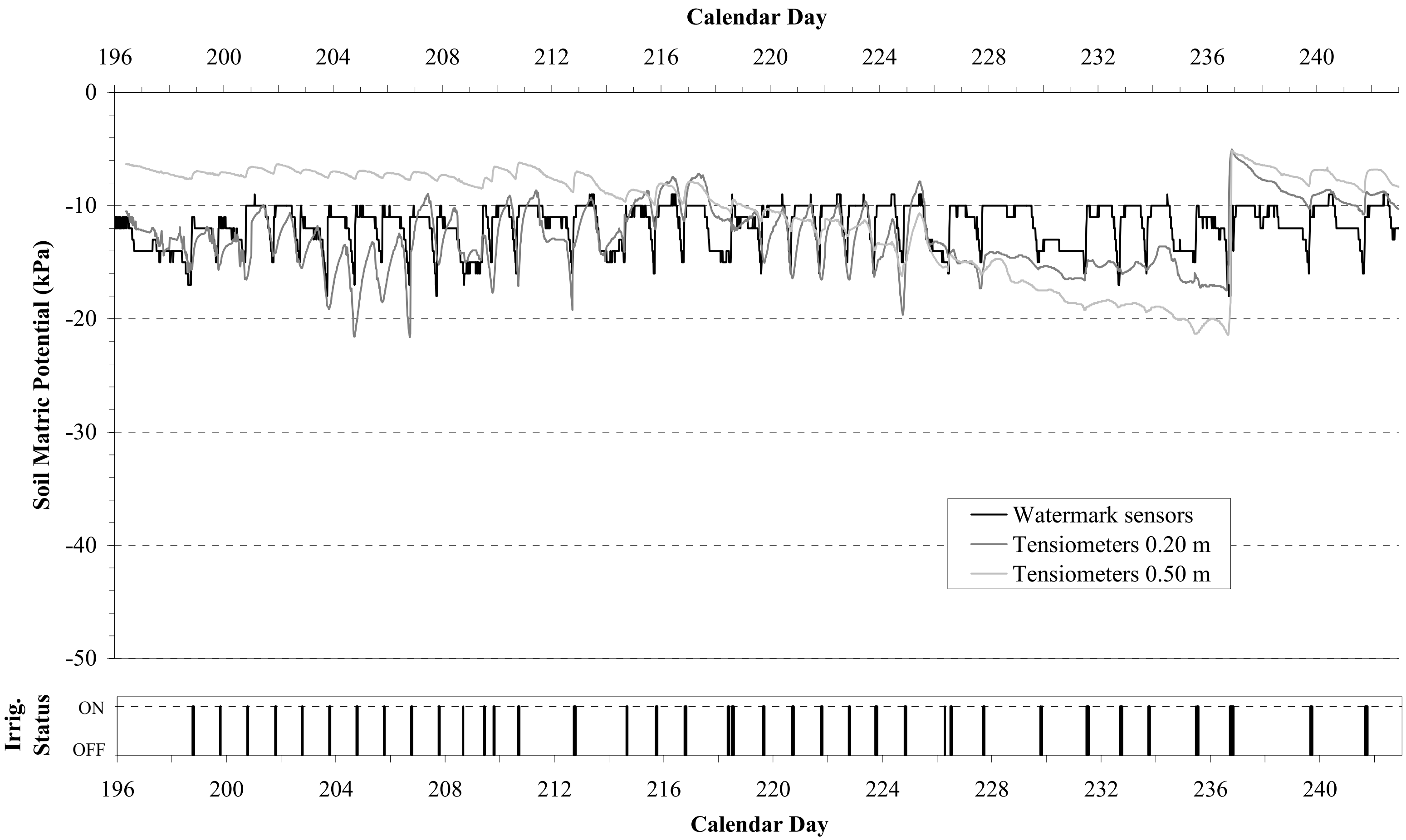


Figure 16. Soil water potential measured in containers irrigated by irrigation controller 4.

extraction, and reached more negative SWP values before irrigation started. Watermark® sensors on the other hand showed very good response to wetting. Similar results were reported by Meron et al. (1996), who used tensiometers to monitor Watermark®-based irrigation control systems in a field study with cotton.

Watermark® readings seemed to plateau at maximum soil water potentials of –7 kPa to –10 kPa. That fact confirmed that the soil water potential threshold level should not be much higher than –15 kPa when using the Watermark® sensors with the calibration equation developed by Shock et al. (1996). Average soil water potentials at the 0.2-m depth measured with tensiometers and Watermark® sensors were very similar (Table 1), although the tensiometers showed higher maximum SWP readings.

During testing, some variability among the readings of tensiometers and Watermark® sensors at the 0.2-m depth was observed. Besides intrinsic characteristics of each sensor, such as response to drying and to wetting, and sensor calibration, that variability can be explained by heterogeneity of the media, by uneven growth of the roots in the containers, and by uneven wetting patterns.

Table 1. Average and maximum soil water potential values measured for controllers 1 to 4 for 2002.

Irrigation	Tensiometer Readings (kPa) 0.2-m depth		Tensiometer Readings (kPa) 0.5-m Depth		Watermark Readings (kPa) 0.2-m depth	
Controller	Average	Maximum	Average	Maximum	Average	Maximum
1	-11.7	-28.1	-6.9	-11.1	-12.0	-20.0
2	-12.2	-24.8	-12.5	-46.6	-11.8	-20.0
3	-12.4	-24.5	-14.8	-31.2	-11.2	-19.0
4	-12.3	-21.6	-12.3	-21.4	-11.5	-18.0

Except for controller 1, tensiometer readings at the 0.5-m depth showed decreasing soil water potentials after the first week of August (day 215). This could be explained by the root growth beyond the depth where the Watermark® sensors were placed, and could indicate that the irrigations were not exceeding the water holding capacity in the root zone. On day 236 a rainfall of 27 mm occurred while the containers were not shielded, making the soil water potential go above field capacity both at the 0.2-m and at the 0.5-m depth.

As shown in Figures 13 to 16, controlling the soil water potential with sensors installed 0.2 m deep and emitters located at the soil surface inevitably led to cycling in the soil water potential, mostly above the threshold level. Similar results were observed in previous studies using soil sensor-based irrigation control systems (Phene and Howell, 1984; Ribeiro, 1998; and Meron et al., 1996).

This cycling can be attributed to: (a) the time that it takes for the wetting front to move from the emitter through the soil medium to the place where the sensor is installed, and (b) after the water reaches the sensors, the SWP quickly increases above the threshold level due to saturated conditions in the soil medium above the wetting front.

However, the crop can still develop its full potential if the soil water potential in the root zone is maintained within an optimum range most of the time. The effectiveness of the irrigation controllers in maintaining the soil water potential in the root zone within a satisfactory range was evaluated in terms of the percentage of time that the SWP remained less negative than the threshold plus 20% tolerance (-18 kPa), which is shown in Table 2. The controllers showed very good performance, maintaining the SWP at the

Table 2. Percentage of time that soil water potential at the 0.2-m depth remained above -18 kPa.

Irrigation Controller	Soil Water Potential > -18 kPa (% of time)	
	Tensiometers	Watermarks
1	93	98
2	94	100
3	88	98
4	98	100

0.2-m depth within the optimum range for 93% of the time on average according to the tensiometers, and 99% of the time according to the Watermark® sensors.

Water Allocation Among Irrigation Controllers

The irrigation control system was able to allocate the water resources among the controllers or zones according to the priority rank established, allowing the irrigation system to operate continuously under adequate pressure. An illustrative example is presented in Tables 3 to 5, showing data downloaded from irrigation controllers 1, 2, and 4 on calendar day 219.

On that day the irrigation controllers were programmed to allow irrigations only from 18:00 h to 24:00 h. At 17:45 h Watermark sensors of controllers 1, 2, and 4 were already showing soil water potential values below the threshold level (-15 kPa), but the control valves were still closed (irrigation status equal 0).

At 18:00 h all three controllers began an irrigation, which caused the hydraulic pressure in the mainline to drop below the programmed threshold (100 kPa). Forty seconds after the irrigation was initiated, controller 1 measured the hydraulic pressure in the mainline as 76 kPa and closed the solenoid valve. Controller 2 measured the

Table 3. Temperature, soil water potential, hydraulic pressure, and irrigation data recorded by irrigation controller 1, day 219, 2002.

Month	Day	Hour	Temperature (°C)	WM 1 (kPa)	WM 2 (kPa)	WM 3 (kPa)	Pressure (kPa)	Irrigation Status
7	18	17:45	35	-14	-18	-17	290	0
7	18	18:00	35	-15	-18	-17	76	0
7	18	18:15	35	-15	-18	-17	62	0
7	18	18:30	35	-15	-18	-17	62	0
7	18	18:45	35	-15	-18	-17	62	0
7	18	19:00	35	-15	-18	-17	62	0
7	18	19:15	35	-15	-18	-17	76	0
7	18	19:30	35	-15	-18	-17	76	0
7	18	19:45	34	-14	-18	-17	69	0
7	18	20:00	34	-14	-18	-17	228	1
7	18	20:15	34	-14	-18	-17	221	1
7	18	20:30	34	-14	-15	-17	221	1
7	18	20:45	34	-14	-11	-17	221	0
7	18	21:00	33	-14	-11	-17	290	0

Table 4. Temperature, soil water potential, hydraulic pressure, and irrigation data recorded by irrigation controller 2, day 219, 2002.

Month	Day	Hour	Temperature (°C)	WM 1 (kPa)	WM 2 (kPa)	WM 3 (kPa)	Pressure (kPa)	Irrigation Status
7	18	17:45	37	-14	-17	-15	290	0
7	18	18:00	37	-14	-17	-15	83	0
7	18	18:15	37	-14	-17	-15	83	0
7	18	18:30	37	-14	-17	-15	83	0
7	18	18:45	36	-14	-17	-15	83	0
7	18	19:00	36	-14	-17	-15	83	0
7	18	19:15	37	-14	-17	-15	200	1
7	18	19:30	37	-14	-17	-15	200	1
7	18	19:45	37	-14	-17	-15	200	1
7	18	20:00	36	-14	-17	-13	221	0
7	18	20:15	36	-14	-16	-12	221	0
7	18	20:30	36	-14	-16	-12	221	0
7	18	20:45	36	-14	-16	-12	241	0
7	18	21:00	36	-14	-15	-12	290	0

Table 5. Temperature, soil water potential, hydraulic pressure, and irrigation data recorded by irrigation controller 4, day 219, 2002.

Month	Day	Hour	Temperature (°C)	WM 1 (kPa)	WM 2 (kPa)	WM 3 (kPa)	Pressure (psi)	Irrigation Status
7	18	17:45	37	-17	-17	-19	290	0
7	18	18:00	37	-17	-17	-20	200	1
7	18	18:15	37	-17	-17	-20	200	1
7	18	18:30	37	-17	-17	-20	200	1
7	18	18:45	37	-17	-17	-16	200	1
7	18	19:00	37	-14	-17	-15	200	1
7	18	19:15	37	-11	-17	-14	193	0
7	18	19:30	37	-11	-15	-12	214	0
7	18	19:45	36	-11	-14	-11	214	0
7	18	20:00	36	-11	-13	-11	221	0
7	18	20:15	36	-11	-13	-11	221	0
7	18	20:30	36	-11	-12	-11	221	0
7	18	20:45	36	-11	-12	-11	241	0
7	18	21:00	36	-11	-12	-11	290	0

hydraulic pressure as 83 kPa 30 s later and also interrupted the irrigation.

At the time controller 4 checked the pressure (60 s after controller 2) the pressure in the mainline was already above the threshold level (200 kPa) and the irrigation continued. That sequence was repeated every 15 min until 19:15 h, when soil water potential readings of two Watermark® sensors of controller 4 were higher than –15 kPa, determining the end of the irrigation for that zone. Access to water was then transferred to irrigation controller 2 until 20:00 h, when controller 1 finally began an irrigation.

The approach of using the hydraulic pressure in the mainline and the time when the pressure measurement is taken to control the zones access to the water proved to be effective for the tested irrigation system. The irrigation sequence showed that when more than one zone required irrigation at the same time the zone with the highest priority was irrigated first.

A summary of the number of events when irrigation was required by each zone and the number of times when irrigation was applied are shown in Table 6. Each event corresponds to a 15-min cycle. As expected, the percentage of events when irrigation was applied compared to the number of times it was required increased as the controller priority increased, and the zone with the highest priority (priority 4) was irrigated every time it required irrigation, confirming the effectiveness of the priority scheduling.

The volume of water applied to each container during normal irrigation events, the volume of water applied during denied irrigation requests before the valve was closed, and the estimated evapotranspiration are shown in Table 7. The volume of water applied to each container by irrigations was calculated by multiplying the emitter flow rate (2.01 L h^{-1}) by the number of events when irrigation was applied (from Table 6). The volume of water applied on denied irrigation requests was calculated using the same procedure and considering the delay time for each controller (40 s, 70 s, and 100 s for controllers 1, 2, and 3).

Table 6. Number of times irrigation was requested by the controllers and number of times irrigation was applied.

Controller Priority	Number of irrigation requests	Number of times that irrigation was applied*	%
1	367	180	49
2	200	136	68
3	220	160	73
4	193	193	100

* Each irrigation event corresponds to a 15-min period.

Table 7. Irrigation data compiled for controllers 1 to 4.

Controller Priority	Number of Irrigations	Volume irrigated (L)	Volume applied on denied requests (L)	Drainage (L)	Estimated ET (mm)
1	24	90	4	3	236
2	20	68	3	0	184
3	35	80	3	0	216
4	35	97	0	0	252

* Calculated ET (FAO Penaman-Monteith) for period: 212 mm

The variation in the volume irrigated among the controllers can be attributed to uneven development of the grass in the containers, since the irrigation time did not restrict the daily irrigation of any controller priority, and tensiometer readings showed that the soil water potential control in the root zone was effective. In two containers irrigated by controller 2, the grass did not develop as well as it did in the other containers. This could be caused by variation in the sod used or variation in soil fertility among the containers. The containers that received less irrigation (controller 2) also showed higher soil water potential at the 0.5 m depth, indicating that less water was stored below the controlled root zone.

The volume of water applied before the valve was closed each time an irrigation request was denied represented less than 5% of the volume irrigated per container. In a real situation on an irrigated field this percentage probably would be smaller, since the pressure in the system during the delay time would be lower than the recommended operating pressure for the emitters, and thus the flow rate of non-compensated emitters would be smaller.

During the period evaluated no drainage water was collected at the bottom of the containers irrigated by controllers 2, 3, and 4 as a result of irrigations, confirming that the system did not overirrigate during the testing period. Controller 1 had 0.7 L and 2.1 L of drainage water collected on days 208 and 225, respectively. However, that amount represented only 3% of the total irrigation depth applied during the evaluation. For both days the irrigation events the previous day lasted longer than usual, probably due to changes in the wetting front pattern. The Watermark® sensors only detected changes in soil water potential caused by the wetting front after 2.5 h and 3 h of irrigation, respectively, which was above the average irrigation time observed (1 h).

The total evapotranspiration was calculated by dividing the net volume of water applied to each container by the estimated grass surface area. Based on visual observations, an estimated surface area of 0.385 m^2 (0.7 m diameter) was used in the calculations, since the lateral growth of the Bermuda grass was not controlled and part of the grass canopy extended beyond the outside edge of the containers.

The total reference evapotranspiration (ETo) for the evaluation period, calculated according to the FAO Penman-Monteith method was 212 mm. The total evapotranspiration observed for containers irrigated by controllers 1, 2, and 3 was higher than the estimated ETo. One probable reason for higher ETo was the height of grass in the containers (about 0.3 m), which was higher than the standard grass height used in ETo calculations (0.12 m).

For the priority scheduling to work well over the entire testing period it was important to keep the clocks of the irrigation controllers synchronized. This was

successfully achieved using the timer and the 3-way valve to drop the pressure in the mainline to 0 kPa every 3 days. When the pressure drop occurred the clocks of the irrigation controllers were reset simultaneously, keeping the time difference among them within 5 s.

It should be noticed that the priority schedule approach used in this test is only needed if the irrigation system has limited capacity to compensate for flow rate and pressure fluctuations in the mainline caused by the irrigation of more than one zone at the same time. Other possible solutions to the problem could be the use of variable-speed pumps, the use of more than one pump in parallel activated according to the system flow-rate requirement, or a constant-rate pump with a recirculating by-pass. These solutions however, imply an increase in pumping costs.

System Hardware and Software Performance

The adequate operation of the hardware was verified by measuring the system down time. The irrigation control system hardware and software worked as planned performing all tasks as designed. Data downloaded from the controllers showed that all irrigation controllers were able to continually measure soil temperature, soil water potential, and hydraulic pressure, and to open or close the solenoid valve when needed without failure. User interaction with the controllers using a notebook computer was also successful.

The solar panel voltage, the battery voltage, and charging current measured during February of 2003 for one of the irrigation controllers are shown in Figure 17. The charger control circuit worked as designed, regulating the solar panel voltage and keeping

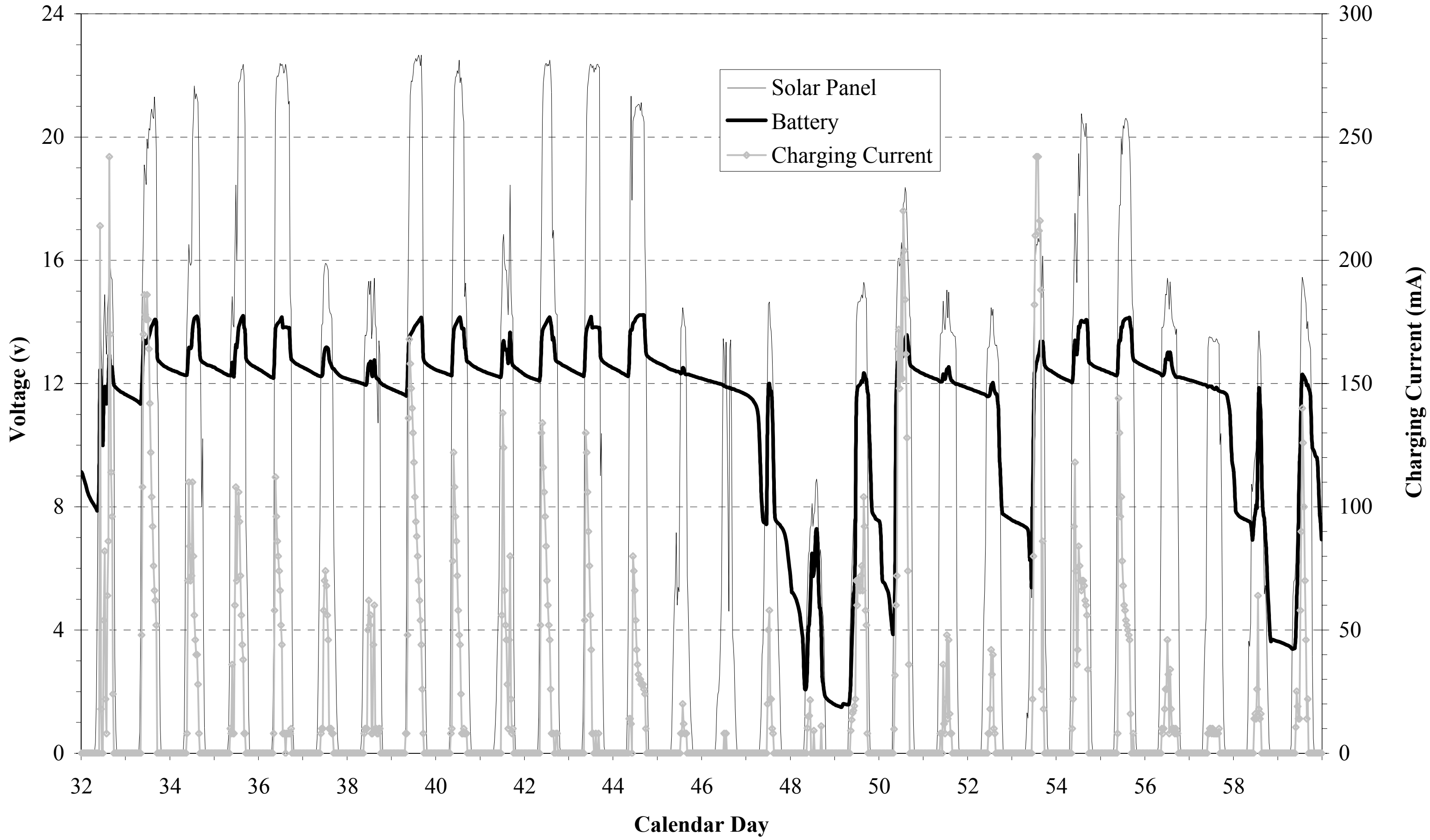


Figure 17. Battery voltage, solar panel voltage, and charging current measured during February 2003.

the battery charged without overcharging. As designed, the maximum charging current applied to the battery was 242 mA, and the system switched to float charging when the charging current decreased to 12 mA as seen on days 42 and 43.

The power supply subsystem kept the irrigation controller working during 80% of the time for that month, even though the system was designed considering average annual solar radiation, and the average solar radiation observed during the period shown (2.3 kWh m^{-2} d^{-1}) was about half of the annual average for the region (4.5 kWh m^{-2} d^{-1}).

As observed on days 45 to 48, after being fully charged the battery lasted 62 h (about 2.5 days) without being recharged, confirming design calculations. A decrease in the battery capacity was expected, since the average temperature during days 45 to 48 was only 5°C and the battery capacity is rated at 20°C. Even with the battery completely discharged, the irrigation controllers started working again as soon as the solar radiation increased and the solar panel voltage reached 7 V.

Occasional interruptions observed in the controller operation were not of concern because the system was not designed to operate during winter; with lower solar radiation, less evapotranspiration, and more rainfall, very little or no irrigation would be required. However, deep discharge at low rates significantly reduces the battery life and should be avoided.

Daily solar radiation values measured by a weather station approximately 30 m from the experiment site and the battery voltage measured daily after it had been charged (at 18:00 h), during February of 2003 are shown in Figure 18. The graph shows that daily solar radiation values of 2.8 kWh m^{-2} d^{-1} were sufficient to maintain the battery

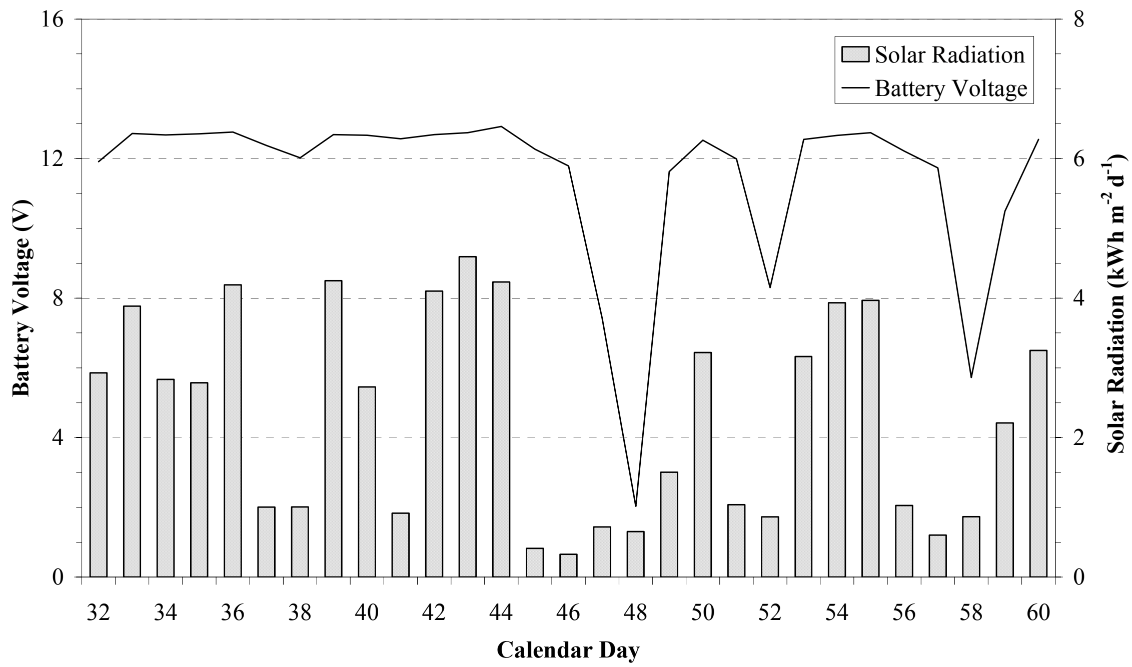


Figure 18. Battery voltage and daily solar radiation observed in Knoxville, TN, 2003.

charge (days 34, 35, and 40). Considering that the average solar radiation for the region is 4.5 kWh m^{-2} d^{-1}, it is clear that a smaller solar panel would be sufficient to power the irrigation controller during the normal irrigation season.

The battery was completely discharged after more than 2 d when the solar radiation was lower than 1.4 kWh m^{-2} d^{-1} (days 45 to 48 and 56 to 58). This problem was due to an under-designed battery capacity (1.2 Ah, designed to last only 2.5 days without recharge). For regions where more than 2 d of low-solar radiation is expected, the use of a battery of higher storage capacity should solve the problem, enabling the controller to operate year around without interruption.

The results indicated that the performance of the controller power supply system could be improved by using a battery of larger capacity, and that the solar panel power

rating could be optimized. Optimization of the controller power supply system, considering its power requirements and the climate characteristics of the region where it will be used is discussed in Appendix C.

The optimization results showed that a 2.5-W solar panel operating with a 2.2-Ah battery represented a better option for Knoxville, TN (latitude 36° N), compared to the 5-W solar panel, 1.2-Ah battery combination used. For a region with higher solar radiation, such as Northeast Brazil (latitude 3° S), a 1.2-Ah battery used with a 1.5-W solar panel would be the best option to power the irrigation controller.

Irrigation Controller Cost

As of January of 2003, the hardware needed to construct one irrigation controller unit cost approximately $190, including the solar panel and the battery. With sensors and the latching solenoid valve, the total cost was $310. For a production scale of 1000 units, the estimated unit cost would be $120 for the irrigation controller, and $210 including sensors and valve. These costs do not include user interface, labor, profits, and the development of the controller. A list with the cost of each component of the irrigation controller hardware is shown in Appendix A.

The estimated costs for the distributed irrigation control system developed in this study and for a centralized control system using a computer as the controller are shown in Table 8. The estimate is for an irrigated field with dimensions of 400 m by 250 m, and the site-specific irrigation of eight zones in the field.

The cost of the eight irrigation controllers required by the distributed control system does not include development, labor, and profits involved in the product

Table 8. Estimated cost of centralized and distributed control systems for a 10-ha irrigated field with eight irrigated zones.

Item	Centralized Control		Distributed Control	
	Quantity	Cost	Quantity	Cost
Desktop computer and monitor	1	$800.00	-	-
A/D card with 64 channels	1	$1,395.00	-	-
Distributed controllers	-	-	8	$1,520.00
Instrumentation cable for sensors (cable #18 AWG - 6 pairs, shielded)*	2000 m	$4,320.00	-	-
Instrumentation cable for sensors (cable #22 AWG - 3 pairs, shielded)*	-	-	400 m	$427.00
Cable #14 AWG -1 pair (for valves)*	1750 m	$2,327.00	10 m	$13.00
Soil moisture sensor	24	$504.00	24	$504.00
Pressure Sensor	8	$145.60	8	$145.60
Temperature sensor	8	$46.40	8	$46.40
Latching solenoid valve	8	$320.00	8	$320.00
Total Cost		**$9,858.00**		**$2,976.00**

* From the 2003 Digikey Catalog.

manufacturing. Therefore, it cannot be compared to the cost of the computer and the A/D card required by the centralized control system. However, the costs of the cables required by both systems differ considerably. Since the distributed controllers are close to the sensors and valves of each irrigated zone, shorter and smaller gage cables are required compared to the centralized control system. As a result the wiring costs for the distributed control system represented only 7% of the wiring costs for the centralized control system, which represented a saving of $6200.

CHAPTER 6

SUMMARY AND CONCLUSIONS

The development and testing of a low-cost, low-power, closed-loop, distributed irrigation control system for site-specific irrigation has been discussed in this dissertation. The system used soil water potential measurements to control the amount of water applied to each specific zone, maintaining the soil water potential in the root zone within a preset range.

Each irrigation controller was autonomously powered by a solar panel and a battery, minimizing maintenance and inter-connection wiring requirements. Hydraulic pressure measurements and priority scheduling were used to allocate water resources among irrigation controllers, in order to irrigate specific zones according to a priority rank.

The study methodology involved system design (hardware and software), experimental implementation, performance evaluation, and power supply optimization. The results obtained in this study allow the following conclusions:

- The irrigation controllers were effective in maintaining the soil water potential in the root zone close to a preset value, and within the optimum range for crop development during 99% of the period evaluated according to Watermark® sensor measurements, and 93% of the time according to tensiometer measurements.
- The performance of the priority scheduling approach for water allocation among the irrigation controllers was excellent, with irrigation of specific zones always occurring

according to the preset priority rank, and under adequate operating pressure.

The distributed control system developed is simpler, less expensive, and more flexible than centralized control systems. The risk of system failure caused by mechanical damage or lightning strikes is distributed over the field, affecting only part of the field, rather than the whole field.

Topics for Future Research

The irrigation controller has been satisfactorily tested under ideal conditions of controlled irrigation of soil containers using a small-scale irrigation system. Field studies using larger irrigation systems, and fields with different topographic, soil, or crop characteristics are recommended to validate the distributed control approach using priority scheduling.

Although the irrigation controller worked adequately with the current hardware configuration, advances and decreasing costs of microprocessors and other electronic components in the future will allow improving the irrigation controller performance without increasing the cost significantly. Such improvements could be:

- Using a microprocessor with larger memory capacity would allow increasing controller capabilities related to detecting faulty sensors, data filtering, and assuring that zones with low priority will receive at least a minimum amount of water even when water resources are very limited.
- Investigating techniques by which data could be centralized for the purpose of better visualization of system performance, checking sensor performance, etc., and

techniques by which additional information could be transmitted through the hydraulic system.

REFERENCES

REFERENCES

Arlosoroff, S. 1971. Automation of Irrigation in Israel. *In* Automated Irrigation. FAO Irrigation and Drainage Paper No.5, Rome, pp.15-44.

Bausch, W.C. and T.M. Bernard. 1996. Validity of the Watermark sensor as a soil moisture measuring device. Proceedings of the International Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, ASAE, St. Joseph, MI, pp. 933-938.

Buchleiter, G.W., D.F. Heerman, and H.R. Duke. 1995. Automation of variable irrigation water and chemical applications. Clean Water, Clean Environment, 21st Century Team Agriculture, Working to Protect Water Resources Conf. Proceedings, March 5-8, Kansas City, Missouri, St. Joseph, MI:ASAE, v.3 p.49-52.

Buresch, M.,1983. Photovoltaic energy systems: design and installation. McGraw-Hill Book Company, New York, NY. 335 pp.

Camp, C.R. and E.J. Sadler. 1998. Modified center pivot system for precision management of water and nutrients. Applied Engineering in Agriculture 14(1):23-31.

Campbell, G.S. and M.D. Campbell. 1982. Irrigation scheduling using soil moisture measurements: Theory and practice. *In* Advances in Irrigation, ed. D. Hillel, 1:25-42. New York: Academic Press.

Cromer, W.A., D.L. Thomas, and M.C. Smith. 1989. A distributed control and monitoring system for a subsurface irrigation and drainage research site. ASAE Paper No.89-6073, ASAE, St. Joseph, MI 49085.

Duke, H.R., L.E. Stetson, and N.C. Ciancaglini. 1990. Irrigation System Controls. *In* Hoffman, G.J., T.A. Howell, and K.H. Solomon. Management of Farm Irrigation Systems. ASAE, St. Joseph, MI, pp. 265-312.

Eldredge, E.P., C.C. Shock, and T.D. Stieber. 1993. Calibration of granular matrix sensors for irrigation management. Agronomy Journal 85(6):1228-1232.

FAO. 2002. Water resources, Development and Management Service. Rome, Italy: Food and Agriculture Organization. Available at: www.fao.org/ag/AGL/aglw. Accessed 03 February 2003.

Fraisse, C.W., H.R. Duke, and D.F. Heermann. 1995a. Laboratory evaluation of variable water application with pulse irrigation. Transactions of the ASAE, 38(5):1363-1369.

Fraisse, C.W., D.F. Heermann, and H.R. Duke. 1995b. Simulation of variable water application with linear-move systems. Transactions of the ASAE, 38(5):1371-1376.

Gardner, W.H. 1986. Water Content. *In* Methods of Soil Analysis. Part I. Physical and Mineralogical Methods (Klute, A., ed.). Agronomy Series No 9. 2nd ed, pp.493-544. ASA Madison WI.

Hillel, D., 1980. Applications of Soil Physics. Academic Press, New York, NY. 385 pp.

Howell, T.A., D.A. Bucks, D.A. Goldhamer, and J.M. Lima. 1986. Management Principles: Irrigation Scheduling. *In* Trickle Irrigation for Crop Production: Design, Operation, and Management, ed. F.S. Nakayama and D.A. Bucks, Elsevier, Tokyo, pp. 241-279.

Jensen, M.E., D.C.N. Robb, and G.E. Franzoy. 1970. Scheduling irrigations using climate-crop-soil data. Journal of Irrigation and Drainage Division. ASCE 96(IRI):25-38.

King, B.A., I.R. McCann, C.V. Eberlein, and J.C. Stark. 1999. Computer control system for spatially varied water and chemical application studies with continuous-move irrigation systems. Computer Electronics in Agriculture 24(3)177-194.

King, B.A., R.E. Reeder, R.W. Wall, and J.C. Stark. 2002. Comparison of site-specific and conventional uniform irrigation management for potatoes. ASAE Paper No.02-2175, ASAE, St. Joseph, MI 49085.

Klute A. (ed.) 1986. Methods of Soil Analysis, Part 1, Physical and Mineralogical Methods, 2nd ed., Agronomy Monograph No. 9, ASA/SSSA, Madison, WI.

Laws, R.J., 1983. Solar cells: what you always wanted to know. Enslow Publishers, Hillside, NJ. 127 pp.

Levin, I., S. Sarig, and M. Meron. 1985. Tensiometers location in controlled automated drip irrigation of cotton. *In* Drip / Trickle Irrigation in Action. Proc. of the 3rd Inter. Drip / Trickle Irrig. Cong., ASAE, Nov. 18-21, Center Plaza Holiday Inn, Fresno, California, USA. II:782-785.

Martin, D.L., E.C. Stegman, and E. Fereres. 1990. Irrigation Scheduling Principles. *In* Hoffman, G.J., T.A. Howell, and K.H. Solomon. Management of Farm Irrigation Systems. ASAE, St. Joseph, MI, pp. 155-203.

McCann, I.R., B.A. King, and J.C. Stark. 1997. Variable water and chemical application for continuous-move sprinkler irrigation systems. Applied Engineering in Agriculture 13(59):609-615.

Meron, M. 1992. Automatic irrigation actuated by soil sensors. Water & Irrigation Review, 12(3):12-16.

Meron, M., R. Assaf, B. Bravdo, R. Wallach, R. Hallel, A. Levin, and I. Dahan. 1995. Soil sensor actuated microirrigation of apples. Microirrigation for a changing world:

conserving resources-preserving the environment. Proceedings of the Fifth International Microirrigation Congress, Orlando, Florida, USA, 2-6 April, 1995, 486-491.

Meron, M.R., R. Hallel, G. Shay, and R. Feuer. 1996. Soil-sensor actuated automatic drip irrigation of cotton. Proceedings of the Int. Conference on Evapotranspiration and Irrigation Scheduling, San Antonio, Texas, November, 1996, 886-892.

Omary, M., C.R. Camp, and E.J. Sadler. 1997. Center pivot irrigation system modification to provide variable water application depths. Applied Engineering in Agriculture 13(2):235-239.

Peterson, D.L., D.M. Glenn, and S.D. Wolford. 1993. Tensiometer-irrigation control valve. Applied Engineering in Agriculture 9(3):293-297.

Phene, C.J. 1986. Operation Principles: Automation. *In* Trickle Irrigation for Crop Production: Design, Operation, and Management, ed. F.S. Nakayama and D.A. Bucks, Elsevier, Tokyo, pp. 188-215.

Phene, C.J., and T.A. Howell. 1981. Control of high-frequency irrigation system. ASAE Paper No.81-2013, ASAE, St. Joseph, MI 49085.

Phene, C.J., and T.A. Howell. 1984. Soil sensor control of high-frequency irrigation systems. Transactions of the ASAE 27(2):392-396.

Phene, C.J., R.J. Reginato, B. Itier, and B.R. Tanner. 1990. Sensing Irrigation Needs. *In* Hoffman, G.J., T.A. Howell, and K.H. Solomon. Management of Farm Irrigation Systems. ASAE, St. Joseph, MI, pp. 207-261.

Phene, C.J., W.R. DeTar, and D.A. Clark. 1992. Real-Time irrigation scheduling of cotton with an automated pan evaporation system. Applied Engineering in Agriculture. 8(6):787-793.

Pogue, W.R. and J.L. Kline. 1995. Watermark moisture sensors use with ET based scheduling models. Microirrigation for a changing world: conserving resources-preserving the environment. Proceedings of the Fifth International Microirrigation Congress, Orlando, Florida, USA, 2-6 April, 1995, 969-974.

Ribeiro, R.S.F. 1998. Fuzzy Logic Based Automated Irrigation Control System Optimized Via Neural Networks. Ph.D. dissertation, Department of Agricultural & Biosystems Engineering, The University of Tennessee, Knoxville, TN.

Sadler, E.J., C.R. Camp, D.E. Evans, and L.J. Usrey. 1996. Irrigation System for Coastal Plain Soils. In Precision Agriculture: Proceeding of the Third International Conference, Minneapolis, MN, 23-26 Mar. 1996. ASA/CSSA/SSSA, Madison, WI. (1)827-834.

Sadler, E.J., R.G. Evans, G.W. Buchleiter, B.A. King, and C.R. Camp. 2000. Design considerations for site specific irrigation. Proceedings of the ASAE 4th National Irrigation Symposium, Phoenix, Arizona, 14-16 November, 2000, 304-315.

Schmugge, T.J., T.J. Jackson, and H.L. McKim. 1980. Survey of methods for soil moisture determination. Water Resources Research 160:961-979.

Shearer, M.N. and J. Vomocil. 1982. Twenty-five years of modern irrigation scheduling promotional efforts. In Proceedings of the Irrigation Scheduling Conference, 208-212. St. Joseph, MI:ASAE.

Shock, C.C., E.B.G. Feibert, and M. Saunders. 1996. Malheur Experiment Station Annual Report. Special Report 964, Oregon State Univ., Ontario, Oreg.

Shock, C.C., E.B.G. Feibert, M. Saunders. 1998. Irrigation Management fro drip-irrigated onions. Oregon State University Agricultural Experiment Station, Special Report 988: 42-48.

Stegman, E.C., L.H. Schiele, and A. Bauer. 1976. Plant water stress criteria for irrigation scheduling. Transactions of the ASAE 19(5):850-855.

Stegman, E.C., J.T. Musick, and J.I. Stewart. 1981. Irrigation Water Management. *In* Jensen, M.E. Design and Operation of Farm Irrigation Systems. ASAE, St. Joseph, MI, pp. 763-816.

Stone, K.C., A.G. Smajstrla, and F.S. Zazueta. 1985. Microcomputer-based data acquisition system for continuous soil water potential measurements. Soil and Crop Sci. Soc. of Fla. Proc. 44:49-53.

Testezlaf, R., F.S. Zazueta, and T.H. Yeager. 1997. A real-time irrigation control system for greenhouses. Applied Engineering in Agriculture. Vol. 13, No. 3, pp. 329-332.

Thomson, S.J., and C.F. Armstrong. 1987. Calibration of the Watermark Model 200 soil moisture sensor. Applied Engineering in Agriculture. Vol. 3, No. 2, pp. 186-189.

Torre-Neto, A., J.K. Schuller, and D.Z. Haman. 2000. Networked sensing and valve actuation for spatially-variable microsprinkler irrigation. ASAE Paper No. 001158.

Van Bavel, M.G. 1995. Advances in microirrigation control by sap-flow monitoring systems. Proceedings of the Fifth International Microirrigation Congress, Orlando, Florida, pp. 235-238.

Vermeiren, I. and G.A. Jobling. 1980. Localized Irrigation – Design, Installation, Operation, Evaluation. FAO Irrigation and Drainage Paper No.36, Rome, 203p.

Wall, R.W., B.A. King, and I.R.McCann. 1996. Center-Pivot irrigation system control and data communications network for real-time variable water application. *In* Precision Agriculture: Proceedings of the Third International Conference on Precision Agriculture, Mineapolis, Minesota, ASA/CSSA/SSSA, pp. 757-766.

Wanjura, D.F., D.R. Upchurch, and J.R. Mahan. 1992. Automated Irrigation Based on Threshold Canopy Temperature. Transactions of the ASAE 35(1): 153-159.

Wessels, W.P.J., W.H. Steyn, and J.H. Moolman. 1995. Automatic microirrigation and salt injection system for research and commercial applications. Proceedings of the Fifth International Microirrigation Congress, Orlando, Florida, pp. 116-122.

Yoder, R.E., D.L. Johnson, J.B. Wilkerson, and D.C. Yoder. 1998. Soil water sensors performance. Applied Engineering in Agriculture 14(2):121-133.

Zazueta, F.S. and A.G. Smajstrla. 1992. Microcomputer-based control of irrigation systems. Applied Engineering in Agriculture 8(5):593-596.

APPENDICES

APPENDIX A

IRRIGATION CONTROLLER HARDWARE

Microcontroller

The BASIC Stamp 2 has a 20-MHz PIC processor (Microchip Tech., Inc. PIC16C57) with 32 bytes of internal RAM (6 I/O, 26 variable). The microcontroller is a 24-pin DIP package with 16 programmable I/O pins (TTL-level, 0-5 V), and two additional pins dedicated to serial I/O.

The microcontroller is capable of running 4,000 instructions per second, and is programmed with a customized form of the BASIC programming language developed by Parallax, Inc., called PBASIC2. The microcontroller is incorporated into an OEM module (Figure A-1), which is designed as a low-cost solution, aimed to facilitate the integration of the Basic Stamp 2 circuit directly into the custom-built circuit. Each OEM module has an on-board voltage regulator, a serial EEPROM, and a DB-9 RS-232 serial port connection. The EEPROM has 2048 bytes of program storage; enough for 500 to 600 lines of program code.

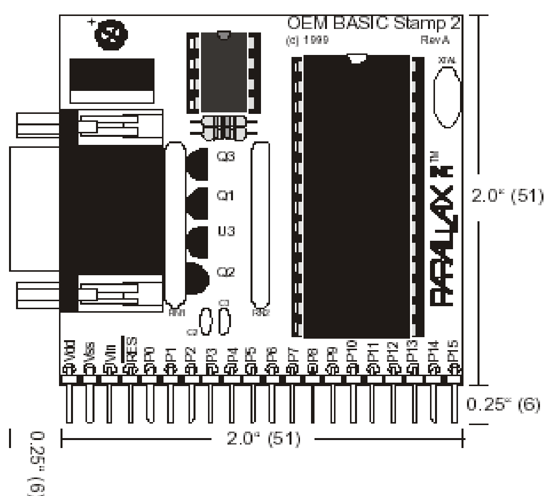


Figure A-1. OEM Basic Stamp 2 module used in the irrigation controller.

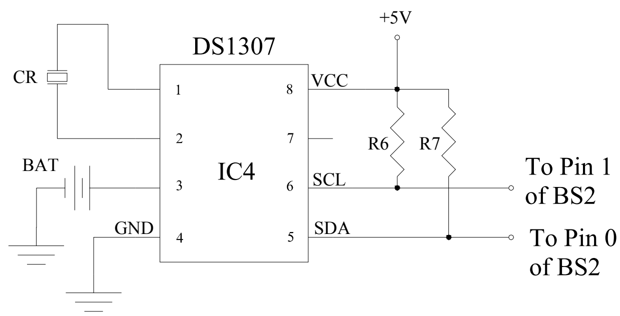
The 5-V regulator converts an input of 6 to 15 VDC (on the VIN pin) to the 5 V required by the BASIC Stamp 2 components. The BASIC Stamp 2 consumes 8 mA in running mode and 100 µA in sleep mode, not including current on the I/O pins. Each I/O pin can source up to 20 mA, and can sink up to 25 mA, limited to 40 mA and 50 mA per eight I/O pins for sourcing and sinking.

Clock

The irrigation controller uses a serial real-time clock (Dallas Semiconductor, DS1307) for timekeeping. The clock requires 2.5 to 5.5 V for full operation, and uses 200 nA of current in stand-by mode, and 1.5 mA when active. It has a built-in power sense circuit, which detects power failures and automatically switches to a 3-V back-up battery supply. The clock accuracy is dependent upon the accuracy of a 32.768-kHz crystal, and temperature shifts can cause the crystal frequency to drift.

The controller communicates with the clock via a 2-wire serial interface, shown in Figure A-2. The clock operates as a slave device on the serial bus. Access is obtained by implementing a “start” condition and providing a device identification code followed by a register address.

Subsequent registers can be accessed sequentially until a “stop” condition is executed. A change in the state of the data line (SDA) from high to low, while the clock line (SCL) is high, defines a “start” condition for data transmission. A change in SDA from low to high, while SCL is high, defines a “stop” condition. This 2-wire handshaking enables the controller writing and reading to and from the clock 8-bits at time. The program used to set the clock is shown in Appendix B.

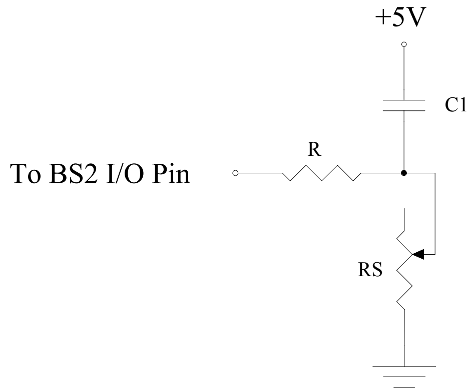


Label	Device	Specifications
IC4	Timer	Dallas Semic. DS1307
R6, R7	Resistor	1 kOhm, 0.25 W, 5%
CR	Crystal	Epson Elec. C-001R 32.768 kHz
BAT	Battery	Panasonic BR 2020, 3 V

Figure A-2. Schematic of the serial real-time clock interfacing with the Basic Stamp 2 microcontroller.

Soil Water Potential Sensor

The irrigation controller monitors the soil water potential in the root zone using three Watermark® sensors. The Basic Stamp 2 microcontroller measures Watermark® sensor resistance by using the RC circuit shown in Figure A-3. Each Watermark sensor is connected to an I/O pin on the microcontroller. To measure the sensor resistance, the capacitor C1 is first discharged until both sides of the capacitor measure 5 V with respect to the ground. This is done by setting the I/O pin high for 1 ms. Through the command "RCtime", the I/O pin is set as an input and the microcontroller measures the time it takes to change states from 1 to 0. In other words, the microcontroller measures the time it takes for the voltage seen by the I/O pin to drop from 5 V to 1.5 V (the RC charge time).



Label	Device	Specifications
C1	Capacitor	Panasonic P/N ECU-S2A 104 KBA, 0.1 μF
R	Resistor	220 Ohm, 0.25 W, 5%
RS	Resistance Sensor	Watermark, Thermistor

Figure A-3. RC circuit used by the microcontroller to read resistance type sensors.

The 220-Ohm resistor protects the I/O pin from a short circuit in case the sensor resistance drops to zero. Watermark® sensors were connected to the controller using #18 AWG-UF wires.

Sensor resistance, in Ohms, can be calculated according to the equation below, that is valid for a 0.1-µF capacitor. The Watermark calibration equation presented by Shock et al. (1996), which was introduced in the previous chapter, is used to convert from resistance to soil water potential, in kPa.

$$R = \frac{RCTime * 2\mu s}{\left(\ln\left(5.0V / 1.5V\right) * 0.1\mu F\right)}$$

Or:

$$R(\Omega) = RCTime * 16.61$$

Soil Temperature Sensor

The irrigation controller uses a thermistor to measure the soil temperature in the root zone. The soil temperature is required to calculate the soil water potential when using an electrical resistance type sensor. The thermistor (Thermometrics, Inc. C100F103G) was chosen because it is a low-cost solid-state temperature sensor, with high sensitivity in typical range of soil temperatures (-10°C to 50°C).

Temperature measurements using the thermistor are performed by the microcontroller through the same RC circuit used for the Watermark® readings, as shown in Figure A-3. The thermistors were coated in epoxy before installation and connected to the controller using #18 AWG-UF wires.

Calibration of the thermistor was performed by comparing the thermistor resistance measured by the microcontroller, with the actual temperature measured with a mercury thermometer. An environmental chamber was used to control the temperature ranging from 5°C to 40°C, with measurements taken at 5°C interval. Measurements of three thermistors were used in the regression analysis.

Figure A-4 shows the thermistor calibration results. Very little variation among thermistors measurements for the same temperature was observed. The regression analysis showed that a quadratic model was the one that best fit the data, with 99.6% of the variation in resistance explained by the calibration equation.

Pressure Sensor

The irrigation controller uses a piezoresistive pressure transducer (Motorola, Inc. MPX5700GP) to monitor the hydraulic pressure in the pipeline. This device is a low-cost

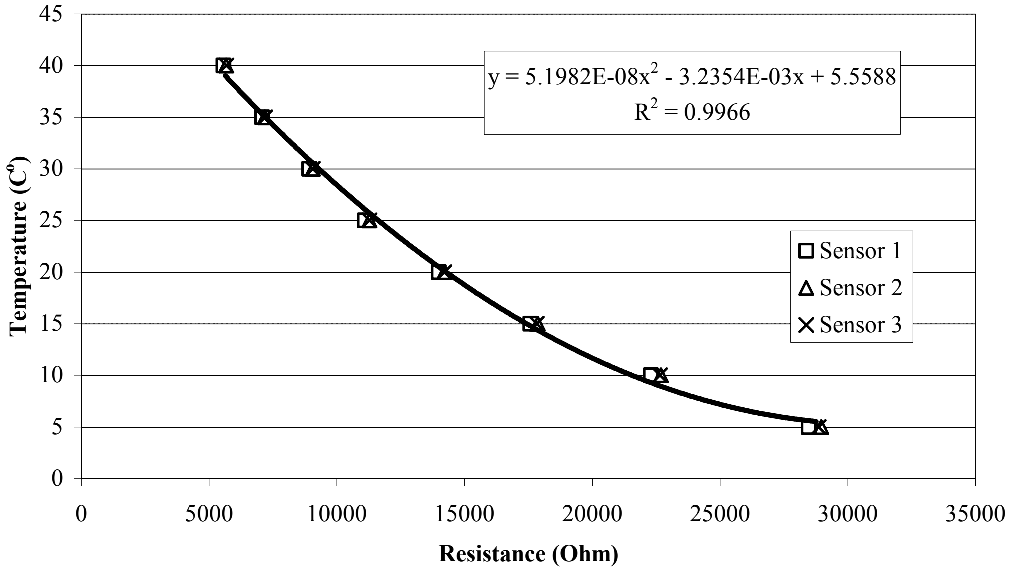
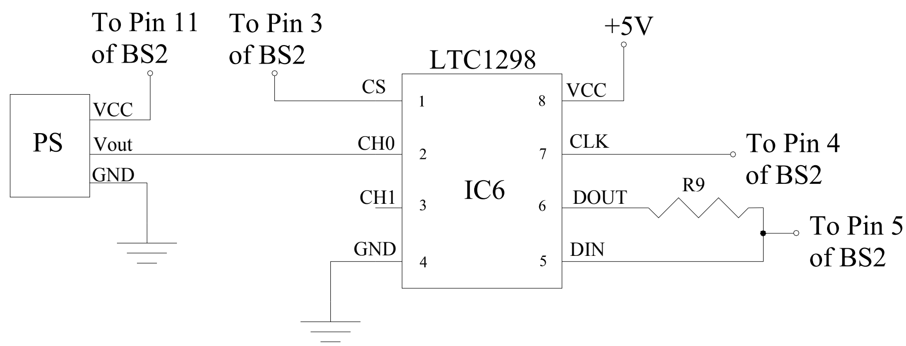


Figure A-4. Calibration data obtained for the Thermometrics C100F103G thermistor.

silicon pressure sensor suited for interfacing with microcontroller-based systems through analog-to-digital converters. It is capable of measuring gauge pressures up to 700 kPa, with an accuracy of ±2.5%, over a temperature range of 0° to 85°C.

The pressure sensor requires a supply voltage of 4.75 to 5.25 VDC, and a supply current of 7-10 mA. The sensor output ranges from 0.2 to 4.7 VDC, and the full-scale response time is 1 ms.

Figure A-5 shows the circuit used to interface the pressure sensor with the A/D converter and the microcontroller. To save power, the pressure sensor is powered for 1 s through an I/O pin of the microcontroller only when a pressure measurement is needed. The pressure transducer signal output is connected to input channel 0 of the analog-to-digital converter (Linear Technology, Inc. LTC1298).



Label	Device	Specifications
IC6	A/D Converter	Linear Tech. LTC1298
R9	Resistor	1 kOhm, 0.25 W, 5%
PS	Pressure Sensor	Motorola MPX5700GP

Figure A-5. Schematic of the pressure sensor and A/D converter circuit used in the irrigation controller.

The A/D is a 12-bit, two-channel converter, with a 1.22 mV resolution over a full-scale voltage input of 0-5 VDC. It has an internal sample-and-hold feature that prevents errors when it is used to measure rapidly changing signals. The channel voltage is measured relative to the ground and returns a value between 0 and 4095 (2^{12} - 1). Supply current to the A/D is typically 250 µA when operating, and 1 nA when not in use.

The A/D interfaces with the microcontroller through four wires: chip selected (CS), clock (CLK), data in (D_{IN}), and data out (D_{OUT}). Data in (D_{IN}) and D_{OUT} are tied together with a 1-kΩ resistor. In order to read the voltage at the A/D input channels, the microcontroller activates CS by taking it low, sends (shifts out) configuration bits to the LTC1298, reads (shift in) the 12-bit measurement from the LTC1298, and deactivates CS by taking it high.

The pressure sensor interfaced with the A/D and the microcontroller, as shown in Figure A-5, was calibrated by reading at the same time the number of bits out of the A/D, and the line pressure shown by Bourdon type pressure gauges. The calibration pressure ranged from 0 to 485 kPa (0 to 70 psi), in 68.96 kPa (10 psi) pre-set pressure intervals. For each pre-set pressure, four pressure sensor readings and the average reading of three Bourdon type manometers were used to calculate the regression equation.

Figure A-6 shows the calibration results. The pressure sensor calibration presented excellent results. Practically no variation was observed among the pressure transducer measurements for the same pressure. The linear model explained almost 100% of the variation presented in the data.

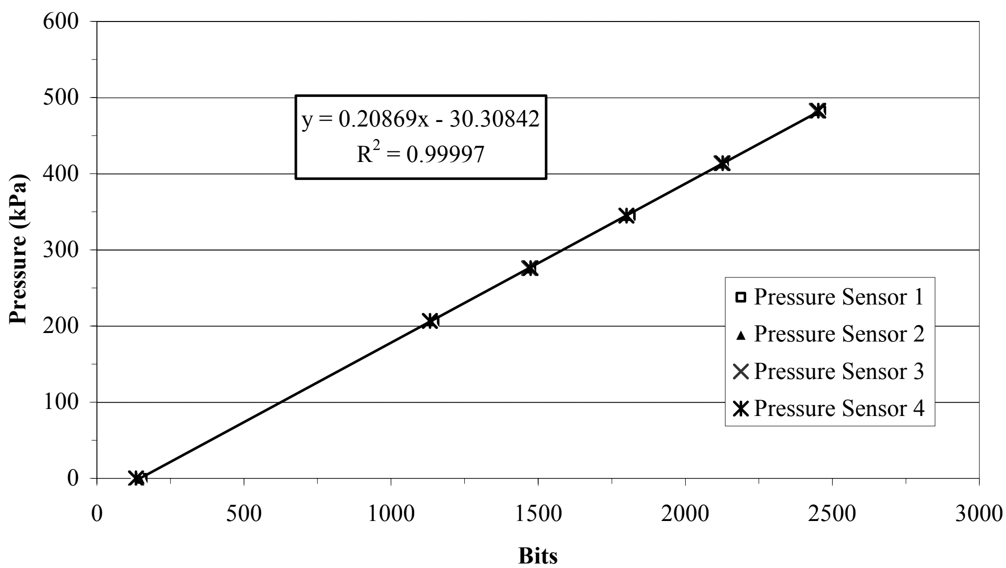


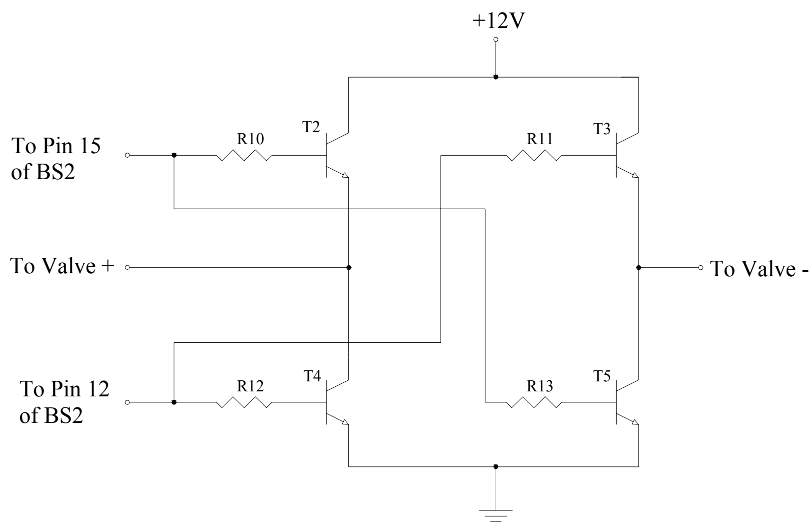
Figure A-6. Calibration data obtained for the Motorola MPX4700GP pressure transducer.

Actuator

A solenoid valve (Rain Bird DV-100-SS) was used as the irrigation actuator, allowing control of the water flow to the specific field zone. The original 24-VAC solenoid coil was replaced with a potted latching solenoid (Rain Bird TBOSPSOL). The valve can be used for flow rates ranging from 0.05 to 9.08 m^3h^{-1}, at operating pressures from 100 to 1000 kPa, but the potted latching solenoid is compatible with several other valve models from the same manufacturer.

The latching solenoid valve is suited for low-power, battery-operated irrigation controllers, and can be powered by 9-VDC or 12-VDC sources. A short DC electrical current pulse (100 ms) flows from the positive to the negative pole of the valve to open the latching solenoid valve. An electrical pulse with the same duration in the opposite direction closes the valve. Preliminary tests showed that the supply current needed to turn on/off the latching solenoid valve was approximately 1 A.

A uni-polar H-bridge circuit (Figure A-7) was developed to activate the latching solenoid valve using the microcontroller and NPN Darlington transistors (Zetex, Inc. ZTX605). To open the valve, an I/O pin of the microcontroller is set high for 100 ms turning on Darlington transistors T2 and T5. This enables current from the battery (+12 VDC) to flow from the positive to the negative terminal of the latching valve. Setting another I/O pin of the microcontroller high for 100 ms, turns on Darlington transistors T3 and T4, making the current flow through the negative to the positive terminal of the latching coil, which closes the valve.



Label	Device	Specifications
T2, T3, T4, T5	Darlington Transistor	Zetex ZTX605
R10, R11, R12, R13	Resistor	4.7 kOhm, 0.25 W, 5%

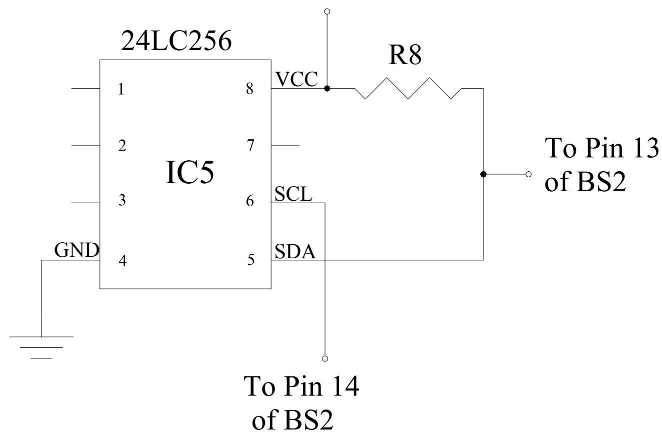
Figure A-7. Uni-polar H-bridge circuit used to activate the latching solenoid valve.

Data Storage

Data collected by the microcontroller, such as date, time, soil water potential, soil temperature, hydraulic pressure in the pipeline, valve status, and some program variables are stored in a 32-kB serial EEPROM (Microchip Tech., Inc. 24LC256), and can be retrieved by the user when needed.

An EEPROM retains the contents of memory, with or without power, until it is overwritten. Since 10 bytes of data are written to the 32-kB EEPROM every 15 min, it can store up to 33 days of data from the irrigation controller.

Communication between the microcontroller and the external EEPROM is done using an I^2C compatible 2-wire serial interface bus (Figure A-8). A serial data bi-directional line (SDA) is used to transfer addresses and data with the device. A serial



Label	Device	Specifications
IC5	32 kB EEPROM	Microchip 24LC256
R8	Resistor	10 kOhm, 0.25 W, 5%

Figure A-8. Schematic of EEPROM interfacing with the Basic Stamp 2 microcontroller .

clock line (SDL) is used to synchronize data transfer between two devices. For normal data transfer, SDA is allowed to change only when SCL is low. Changes while SCL is high are reserved for indicating the start and stop conditions.

The microcontroller is the master device that controls the SCL, controls the bus access, and generates the start and stop conditions, while the EEPROM works as the slave device. Both the master and slave can operate as a transmitter or receiver, but the master device determines which mode is activated.

Alarm

The irrigation controller uses an audible alarm (Cui, Inc. CEP-2242) to warn the user of possible temperature and soil moisture sensor out-of-range readings. The alarm resonant frequency is 4.1 kHz, with an operating voltage ranging from 3 to 16 VDC, and

maximum current consumption of 7 mA at 12 VDC. The controller activates the alarm through I/O pin 2 of the microcontroller.

Battery Selection

A 12-volt sealed lead acid (SLA) battery was selected to store the energy and to power the irrigation controller when sunlight was not available. This type of battery is characterized by low-cost, long operational life, and no memory problems.

Battery capacity was determined using data found in Table A-1. The calculations took into account the possibility that the solar radiation could be very low and not enough to charge the battery for 2.5 consecutive days (two days and one night), and assumed that 70% of the battery capacity was available to use (Buresch, 1983).

A 1.2 Ah battery (Power-Sonic PS-1212) was selected. Considering that 70% of its capacity could be used before the battery voltage dropped below the voltage requirements of the controller, the battery should last 69 hours, or 2.9 days.

Solar Panel Selection

The solar panel was conservatively designed to fully recharge the battery in a single day, after 70% of the battery capacity had been depleted, and considered an average of 4.5 peak sun hours per day for Knoxville, TN (Laws, 1983). Table A-2 shows the calculations used to determine the solar panel capacity.

The calculated solar panel rated power output was 3 W. After searching for solar panel modules available in the market, the Siemens ST-5 solar panel was selected as the best option, meeting system power requirements, with lower cost, and higher quality than smaller solar panels.

Table A-1. Battery capacity calculations used in the irrigation controller design.

Equipment	Mode	Current Draw	Time Active per Hour	Average Current
Irrigation controller circuit	Running	14 mA	½	7 mA
Irrigation controller circuit	Stand-by	10 mA	½	5 mA
Latching valve	Activated	1000 mA	1/9000	0.1 mA
Total Average Current				12.1 mA
Daily load: (0.0121 A * 24 h)				0.29 Ah d^{-1}
Number of days that the battery should run the controller without receiving a charge				2.5 d
Calculated Battery Capacity = (0.29 Ah d^{-1} * 2.5 d) / 70%				**1.0 Ah**

Table A-2. Solar panel capacity calculations used in the irrigation controller design.

Battery capacity = 1.2 Ah x 70%	0.84 Ah
Peak sun hours per day for Knoxville, TN (yearly average)	4.5 hours
Calculated charge current: battery capacity / charge time	187 mA
Average current draw by the irrigation controller	12.1 mA
Total current required	199.1 mA
Calculated panel size: total current x 15 V	3.0 W

* Voltage required to charge the 12-volt battery.

The Siemens ST-5 solar panel dimensions are 0.329 m x 0.206 m x 0.034 m. It presents a peak power rating of 5 W (for an irradiance of 1000 W m^{-2}), with a rated current of 0.32 A, and rated voltage of 15.6 V. Since the power rating of the solar panel was larger than required, it should take fewer than 5 peak sun hours to recharge the battery.

Charge Controller Circuit

A charge controller system is necessary to prevent the battery from overcharging and significantly reducing battery life. The charge controller is used to regulate the charging voltage and current by sensing when the battery is fully charged, then stopping or decreasing the amount of current flowing to the battery.

When the charge voltage is too high, excessive current will flow into the battery after reaching full charge, causing decomposition of water in the electrolyte and premature aging. At high rates of overcharge the battery will heat up. As it gets hotter, it will accept more current, heating up even further. This so-called “thermal runaway” can destroy a battery in a few hours.

The best way to charge sealed lead acid batteries is using the constant current/constant voltage method. This method consists of: (a) applying a constant voltage of 2.45 V/cell to the battery (14.7 V for a 12-volt battery) and limiting initial charging current to 0.2 times the battery capacity (240 mA for an 1.2 Ah battery); (b) charging until the current accepted by the battery drops to less then 0.01 times the battery capacity (12 mA for the 1.2 Ah battery); (c) then switching to a float voltage holding the

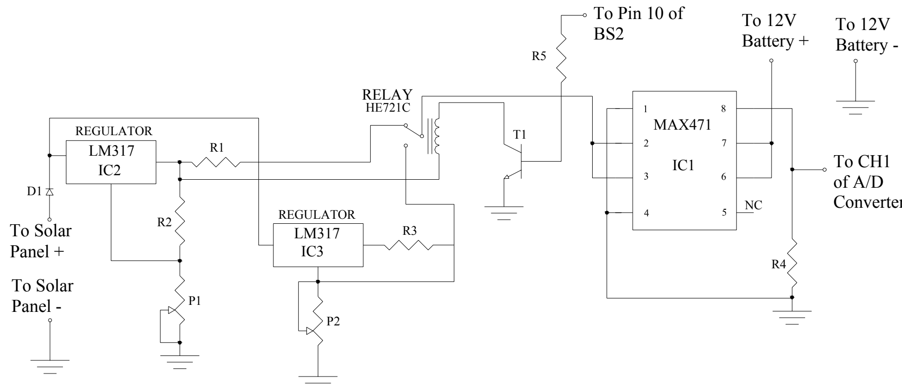
battery across a constant voltage source of 2.25 to 2.30 V/cell (13.50 to 13.80 V for a 12-V battery) continuously.

The charge controller circuit designed to charge the 12-V, 1.2 Ah SLA battery is shown in Figure A-9. A blocking diode (D1) was installed to prevent the solar panel from draining power from the battery when sunlight was not available. Current and voltage regulation were achieved using 3-terminal adjustable regulators (National Semiconductors LM317). The first regulator (IC2) and associated resistors were used to limit the charging voltage to 14.7 V and the charging current to 240 mA. The second regulator (IC3) was used to adjust the floating voltage to 13.8 V.

A precision high-side current-sense amplifier (IC1) (Dallas Semiconductors MAX471) was used to sense the charging current. The amplifier output voltage (0-1.5 V) was connected to channel 1 of the A/D converter and then read by the microcontroller. The microcontroller was programmed to switch from constant current/constant voltage mode to float voltage when the charging current dropped below 24 mA (12 mA for the battery plus 12 mA for the microcontroller).

To switch between the two charge modes, a single-pole, double-throw relay (Hamlin HE721C1210) was used. Switching from current regulation (normal operation) to float voltage is done by setting high one I/O pin of the microcontroller. The 29-mA current draw of the relay coil is powered by the 14.7-V regulated source via an NPN transistor.

In this arrangement, the relay is activated only when there is excess power available, i.e., only when the solar panel is generating power and the battery is fully



Label	Device	Specifications
IC1	Current-Sense Amplifier	Maxim MAX471
IC2, IC3	3-Terminal Adjustable Regulator	National Semic. LM317
RL	Relay	Hamlin HE721C1210
T1	Transistor	Zetex ZTX458
R1	Resistor	5 Ohm, 1 W, 5%
R2, R3	Resistor	820 Ohm, 0.25 W, 5%
R4	Resistor	38.8 kOhm, 0.25 W, 5%
R5	Resistor	4.7 kOhm, 0.25 W, 5%
P1, P2	Potentiometer	Bourns 3006P/103, 10 kOhm, 0.75 W
D1	Diode	1N4004

Figure A-9. Schematic of the charge controller circuit used in the irrigation controller.

charged. The normally closed operation of the relay allows the battery to start charging and to turn on the microcontroller anytime the sun is shining, even if the battery is completely discharged.

<u>I/O Interface</u>

Interface between the irrigation controller and an external microcomputer was accomplished by using the Basic Stamp 2 RS-232 serial port connection. The software Stampw.exe, developed by Parallax, Inc., was used to write and send PBASIC programs to the microcontroller, and to communicate with it, through the "debug" function. That allowed irrigation variables used by the irrigation controller program, such as priority, soil water potential threshold, pressure threshold, and period of time when irrigation is

allowed, to be set. The software HyperTerminal was used to download and save data from the irrigation controller, and also allowed the setting of irrigation controller variables.

Enclosure

The irrigation controller was housed in a weather proof polycarbonate enclosure that meets or exceeds those requirements of the NEMA 4X (PN-1329, Bud Industries). The outer box dimensions measured 75 x 146 x 222 mm. Holes were drilled in the bottom of the box for electrical cables, and in the top right side for the DB-9 serial cable, allowing communication with the controller without having to open the enclosure. Figure A-10 shows data being downloaded from the irrigation controller to a laptop computer. The schematic of the irrigation controller circuit is presented in Figure A-11. The complete parts list and cost of each component of the irrigation controller is shown in Table A-3.

Figure A-10. Data download from the irrigation controller using a laptop computer.

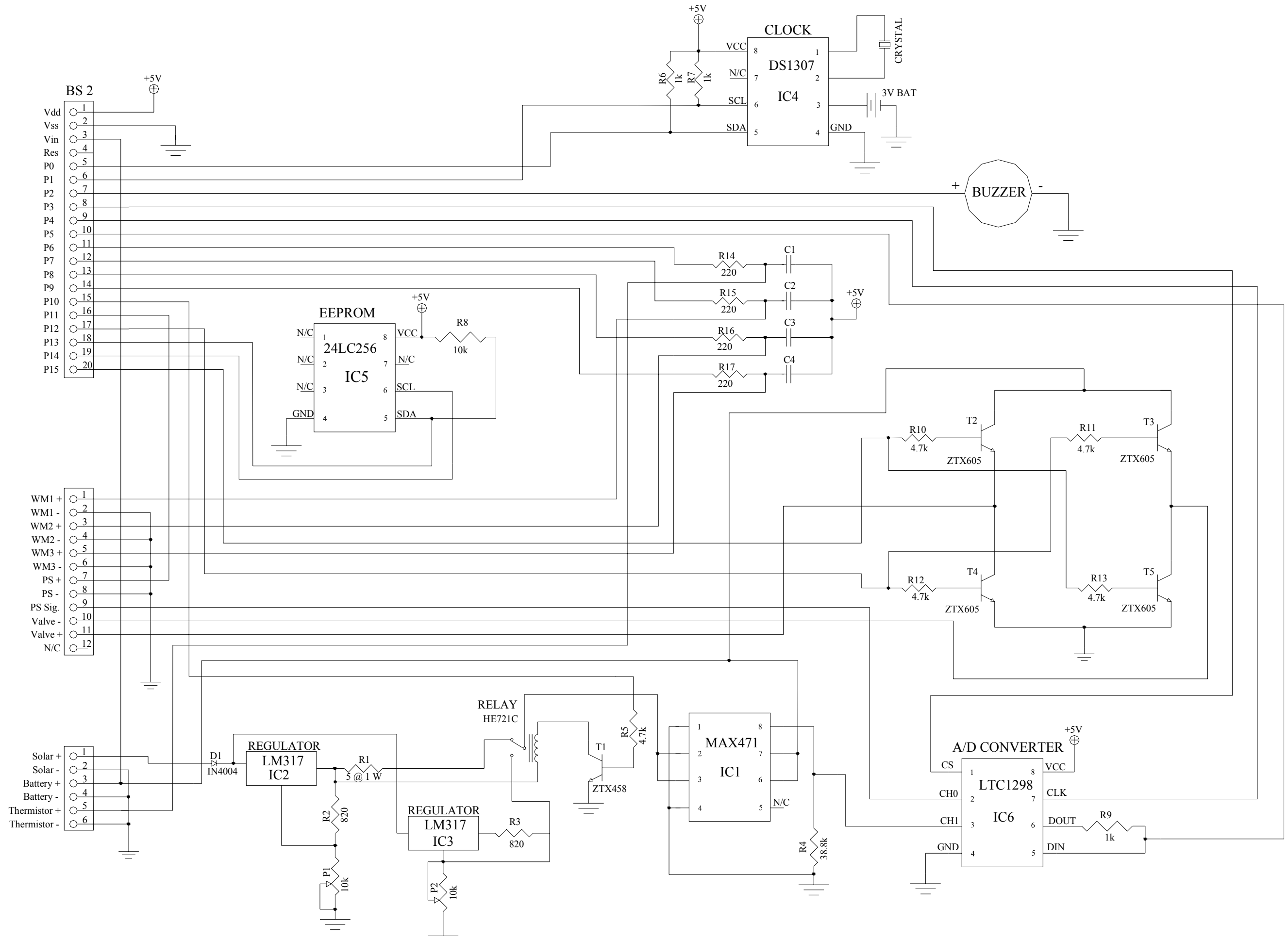


Figure A-11. Schematic of electric circuit used in the irrigation controller.

Table A-3. List of parts used in the irrigation controller.

PART	Units	Unit Cost*	Total Cost
Controller Circuit:			
Enclosure	1	16.40	16.40
Printed Circuit Board	1	3.00	3.00
Parallax OEM Basic Stamp 2	1	45.00	45.00
Real Time Clock Dallas DS1307	1	4.16	4.16
Crystal 32768 KHz C-001R	1	0.42	0.42
Battery 3V	1	2.50	2.50
Battery holder (3V battery)	1	1.00	1.00
A/D Converter LTC1298 - Linear Technology	1	9.88	9.88
EEPROM Microchip 24LC256	1	2.28	2.28
Capacitor 0.1 microF	4	0.30	1.20
Resistor 220 Ohm (RRBr)	4	0.05	0.20
Resistor 1 kOhm (BrBlR)	3	0.05	0.15
Resistor 10 kOhm (BrBlOr)	1	0.05	0.05
Resistor 4.7 kOhm (YBrR)	4	0.05	0.20
Darlington Transistor ZTX605	4	0.78	3.12
PCB Terminal Block 12 pin	2	3.74	7.48
Alarm CEP-2242	1	2.72	2.72
Battery B&B BP1.2-12	1	12.00	12.00
Solar Panel Siemens ST5	1	57.00	57.00
Diode 1N4004 (charger ckt)	1	0.43	0.43
Adjustable Regulator LM317 Nation. Semic.	2	1.76	3.52
Current-sense amplifier MAX471	1	4.76	4.76
Relay HE721C	1	6.11	6.11
Transistor ZTX458	1	0.54	0.54
Resistor 5 Ohm, 1W	1	1.00	1.00
Potentiometer 3006 P 5K	2	1.73	3.46
Resistor 820 Ohm	2	0.05	0.10
Resistor 38.8 kOhm	1	0.05	0.05
Resistor 4.7 kOhm	1	0.05	0.05
		Subtotal =	**188.78**
Sensors and actuator:			
Soil Moisture Sensor Watermark 200SS	3	21.00	63.00
Pressure Sensor Motorola MPX5700GP	1	18.19	18.19
Thermistor NTC C100F103G	1	5.80	5.80
Valve Rain Bird 100-DV-SS	1	18.75	18.75
Latching Solenoid RainBird	1	18.75	18.75
		Subtotal =	**124.49**
		Total* =	**313.27**

* According to the 2003 Digikey catalog, for unit price. The total cost for 1000 units would be $210.71.

APPENDIX B

SOFTWARE CODE

BASIC Stamp 2 Program Code for *icontroller.bs2*

```
'{$STAMP BS2}
'***************************************************************
' This program controls the irrigation of a field specific zone.
'**** *********************************************************
'*****ASSIGN VARIABLES, CONSTANTS, AND I/O PINS *****
'*****Variables for the CLOCK:
Tadd     var    word          'Address to read from or write to in DS1307
Tdata    var    byte          'Data read from or written to DS1307
Tack     var    bit
P        var    byte          'Miscellaneous counter
timeA    var    byte(7)
Tclk     con    1             'Time clock line
Tdat     con    0             'Time data line
'*****Variables for the EEprom:
EEadd    var    word          'Address to read from or write to in EEprom
EEdata   var    byte          'Data read from or written to EEprom
EEack    var    bit
EEclk    con    14            'EEprom clock line
EEdat    con    13            'EEprom data line
Pointer  var    word
'*****Variables for AD Converter (battery charging current  and pressure measurements):
CS1      con    3             'Chip select; 0=active
CLK1     con    4             'Clock to ADC; out on rising, in on falling edge
DIO_n    con    5             'Data I/O pin number
Batt            con  2        'Data I/O pin number (battery)
config   var    nib           'Configuration bits for ADC, Bat. Voltage.
configP  var    nib           'Configuration bits for ADC, Pressure.
startB   var    config.bit0   'Start bit for comm with ADC
sglDif   var    config.bit1   'Single-ended or differential mode
oddSign  var    config.bit2   'Channel selection
msbf     var    config.bit3   'Output 0s after data xfer complete
'*****Variables for Soil Moisture, soil temperature, and pressure measurements:
T        var    word          ' Miscellaneous counter
R        var    word          'Variable to hold 12-bit AD result for pressure, watermarks resistance,
                              ' and the Address to read from, or write to in DS1307
Temp     var    byte          'Variable to hold the soil temperature (C)
SWP1     var    byte          'Variable to hold the soil water potential at location 1 (kPa)
SWP2     var    byte          'Variable to hold the soil water potential at location 2 (kPa)
SWP3     var    byte          'Variable to hold the soil water potential at location 3 (kPa)
Valve    var    bit           'Variable to hold the solenoid valve status
Buzz     var    bit           'Variable to hold the alarm status

OUTA = %0011                  'Initialize: Tclk = High
DIRA = %0001                  'Initialize: Tdat = Input
OUTD = %0011                  'Initialize: EEclk = High
DIRD = %0001                  'Initialize: EEdat = Input
Low 12                        'Open "valve switches"
Low 15

LOOP1:                        '*****BEGIN MAIN LOOP*****
```

```
    GOSUB PRESSURE              'Check the pressure
    EEadd=7
    GOSUB ReadEE                'Read the controller Head in EEprom address7
    Temp= (EEdata*/$016C)+2     'Convert Head from m to psi and add factor
    IF P<Temp THEN CLOCKRESET               'Reset the clock if the pressure is <2psi
    GOSUB ReadTime              'Read time
    PAUSE 1100
    '*****If time is at 0, 15, 30 or 45 min read sensors
     IF (timeA.lownib(3)*10 + timeA.lownib(2)+1)//15=0 THEN STAGE1
     DEBUG tab, "Ch. settings?",cr,10               'Ask if user want to change settings
     SERIN 16,16468,ASK1,5000,ASK1,[str Temp\1]
     IF Temp = "y" OR Temp = "Y" THEN SET       'If "yes" go to SET subroutine
    ASK1:
    DEBUG tab, "Download?",CR,10         'Ask if user want to download data from controller
    SERIN 16,16468,5000,REST,[str Temp\1]
    IF Temp = "y" OR Temp = "Y" THEN DOWNLOAD      'If "yes" go to DOWNLOAD subroutine
    REST:                       'Subroutine to save power during intervals
    SLEEP 10                    'Put the Basic Stamp to sleep for10 s
    IF Buzz=1 THEN FREAK        'If alarm flag is high then send warning signal
GOTO LOOP1                      'Go back to loop
'**********************************************************************
FREAK:                          'Subroutine to send warning signal
   FOR P=1 to 10
        FREQOUT 2,100,4100      'Buzz 10 times, for 100 ms, at 4100Hz
        PAUSE 500
   NEXT
GOTO LOOP1                                      'Go back to loop
'**********************************************************************
CLOCKRESET:                     'Subroutine to reset the clock
    GOSUB ReadTime    'Read time
    timeA.lownib(5)=0     'Set hour to 1, and min and seconds to zero
    timeA.lownib(4)=1
    timeA.lownib(3)=0
    timeA.lownib(2)=0
    timeA.lownib(1)=0
    timeA.lownib(0)=0
    TAdd = 0
    GOSUB SendAddr
    GOSUB INICLK
    GOSUB SendStop
    SLEEP 5                             'Put the Basic Stamp to sleep for 5 s
GOTO LOOP1                                      'Go back to loop
'**********************************************************************
DOWNLOAD:                       'Subroutine to download data stored in the EEprom
    EEadd=0                     'Read the value of the pointer.LOWBYTE in EEprom address 0
    GOSUB SendAddrEE
    GOSUB InitReadEE
    GOSUB SeqReadEE
    Pointer.LOWBYTE=EEdata
    GOSUB LastReadEE            'Read the value of the pointer.HIGHBYTE in EEprom address 0
    Pointer.HIGHBYTE=EEdata
    Pointer=(Pointer.HIGHBYTE*256)+ Pointer.LOWBYTE 'Calculate the pointer value
    EEadd=8                     'Initial EEPROM address to read from
```

```
    Pointer=ABS(Pointer-1)        'Final pointer to read
    FOR T = 0 to Pointer
      DEBUG cr,10
      FOR P = 1 to 5
        GOSUB SendAddrEE
        GOSUB InitReadEE
        GOSUB SeqReadEE
        DEBUG DEC2 EEData, ", "
        GOSUB LastReadEE
        DEBUG DEC2 EEData, ", "
        EEadd=EEadd+2
      NEXT
    NEXT
    Pointer=0                     'Reset the Pointer
    EEadd=0                       'Address to write the pointer
    EEdata=Pointer.LOWBYTE
    GOSUB STORE                   'Store the new value of the pointer low-byte in address 0
    EEdata=Pointer.HIGHBYTE
    GOSUB STORE                   'Store the new value of the pointer high-byte in address 1
    Buzz=0                        'Reset the alarm
    SLEEP 10                      'Put the BS2 in low power for 10 s
    GOTO LOOP1                    'Go back to the beginning
'*************************************************************************
SET:                              'Subroutine to change controller settings
    DEBUG tab, "Priority",CR,10             'Input the controller PRIORITY (1, 2, 3, …, n)
    SERIN 16,16468,20000, ASK1,[dec Temp]
    EEadd=2
    EEdata=Temp
    GOSUB STORE                   'Store the priority in EEprom address 2
    DEBUG tab, "Start Irr",CR,10            'Input the TIME TO START IRRIGATIONS (0-24 Hours)
    SERIN 16,16468,20000,ASK1,[dec Temp]
    EEadd=3
    EEdata=Temp
    GOSUB STORE                   'Store irrigation start time in EEprom address 3
    DEBUG tab, "Stop Irr",CR,10             'Input the TIME TO END IRRIGATIONS (0-24 Hours)
    SERIN 16,16468,20000,ASK1,[dec Temp]
    EEadd=4
    EEdata=Temp
    GOSUB STORE                   'Store irrigation stop time in EEprom address 4
    DEBUG tab, "SWP",CR,10        'Input the SOIL WATER POTENTIAL TRESHOLD (kPa)
    SERIN 16,16468,20000,ASK1,[dec Temp]
    EEadd=5
    EEdata=Temp
    GOSUB STORE                   'Store SWP threshold in EEprom address 5
    DEBUG tab, "P",CR,10          'Input the PRESSURE TRESHOLD (PSI)
    SERIN 16,16468,20000,ASK1,[dec Temp]
    EEadd=6
    EEdata=Temp
    GOSUB STORE                   'Store minimum hydr. pressure in EEprom address 6
    DEBUG tab, "H",CR,10          'Input the controller Head (m)
    SERIN 16,16468,20000, ASK1,[dec Temp]
    EEadd=7
    EEdata=Temp
```

```
    GOSUB STORE                 'Store the Head in EEprom address 7
    GOTO ASK1                   'Return to main loop
'*****************************************************************************
STAGE1:                         'Subroutine to read sensors and make irrigation decision
    PAUSE 800
    GOSUB ReadTime              'Read time in the DS1307
    '*****Loop: Wait until second=0 (sincronize with other controllers)
    IF timeA.lownib(1)*10 + timeA.lownib(0)>0 THEN STAGE1
    GOSUB MEASURES              'Read soil temp., soil water potential and pressure
    GOSUB STORE1                'Store time and soil water potential in the EEprom
    EEadd=6
    GOSUB ReadEE                'Read pressure threshold in EEprom address 6
    Temp=EEdata                 'Temp receives the pressure threshold
    EEadd=5
    GOSUB ReadEE                'Read SWP threshold in EEprom address 5
    '*****Irrigate if SWP of 2 Watermark sensors are above the threshold
    IF SWP1>=EEdata AND SWP2>=EEdata THEN STAGE2
    IF SWP1>=EEdata AND SWP3>=EEdata THEN STAGE2
    IF SWP2>=EEdata AND SWP3>=EEdata THEN STAGE2
GOTO CLOSEV             'Close irrigation if SWP of 2 Watermark sensors are below the threshold
'*****************************************************************************
STAGE2:                         'Subroutine for when irrigation is needed
    EEadd=7
    GOSUB ReadEE                'Read the controller head at Eeprom address 7
    SWP3= (EEdata*/$016C)+Temp          'Convert head from m to psi and add pressure threshold
    EEadd=3
    GOSUB ReadEE                'Read the irrigation start time at EEprom address 3
    Temp=EEdata                 'Tranfer value to variable Temp
    EEadd=4
    GOSUB ReadEE                'Read the irrigation stop time at EEprom address 4
    '*****Check if it is time to irrigate, if NOT then close valve and store data in EEprom
    IF (timeA.lownib(5)*10+timeA.lownib(4))<Temp OR
(timeA.lownib(5)*10+timeA.lownib(4))>=EEdata THEN CLOSEV
    GOSUB OPENV                 'If it is time to irrigate then open the valve
    EEadd=2
    GOSUB ReadEE                'Read the controller priority in EEprom address 2
    R=EEdata*30                 'Calculate the time to check the pressure
    SLEEP 40                    'Wait to the pressure to stabilize
    SLEEP R                     'Put the BS into low power until the pressure checking
    GOSUB PRESSURE              'Check the  pressure
     IF P<SWP3 THEN CLOSEV  'Close the valve if the pressure dropped below the threshold
GOTO STORE2                     'If pressure is OK store pressure and valve status and go back to loop
'*****************************************************************************
MEASURES:                       'Subroutine to read temperature, SWP and pressure
    GOSUB SOILTEMP              'Measure the soil temperature
    GOSUB SOILMOISTURE          'Measure the soil water potential
    GOSUB PRESSURE              'Measure the hydraulic pressure
    IF Temp>50 THEN BUZZ1       'Set alarm flag high if any reading is out of range
    IF SWP1<6 OR ABS(SWP1-SWP2)>15 THEN BUZZ1
    IF SWP2<6 OR ABS(SWP2-SWP3)>15 THEN BUZZ1
    IF SWP3<6 OR ABS(SWP3-SWP1)>15 THEN BUZZ1
    Cont1:
    RETURN                      'Return to Stage1
```

```
    Buzz1:
    Buzz=1                    'Set alarm flag high
GOTO Cont1                    'Return
'*************************************************************************
OPENV:                        'Subroutine to OPEN the latching valve
    Valve=1                   'Set the valve variable to ON
    HIGH 15                   'Switch on transistors T2 and T5
    PAUSE 100                 'Latching time
    LOW 15                    'Switch off transistors T2 and T5
RETURN
'*************************************************************************
CLOSEV:                       'Subroutine to CLOSE the latching valve
    Valve=0                   'Set the valve variable to OFF
    HIGH 12                   'Switch on transistors T3 and T4
    PAUSE 100                 'Latching time
    LOW 12                    'Switch off  transistors T3 and T4
GOTO STORE2                   'Store pressure and valve status
'*************************************************************************
SOILTEMP:                     'Subroutine to measure the soil temperature
    HIGH 6                    'Discharge the capacitor
    PAUSE 1                   'for 1 ms
    RCTIME 6,1,T              'Measure RC time charge
    'Converting Rctime to degrees Celcius:    Temp= 55.589 - 0.05375(Rctime)+ 1.435E-05(Rctime)^2
    Temp= 56+(((T/10)*(T/10)*/$016F)/1000)-((T*/$0560)/100)
RETURN
'*************************************************************************
SOILMOISTURE:                 'Subroutine to measure the soil water potential with Watermark sensors
    '*****Reading Watermark soil moisture sensor 1:
    HIGH 7                    'Discharge the capacitor
    PAUSE 1                   'for 1 ms
    RCTIME 7,1,T              'Measure RC time charge
    '*****Converting from RCtime to resistance:     R = (RCtime * 2 micros)/( ln(1.5V/5.0V)*0.1 microF)
    'R (ohm) = RCtime * 16.6
    R = (T*15)+(T*/$0199)
    '*****Converting from Resistance to kPa:   SWP (kPa) = {[(2.678+0.003892*R)/[1- 0.01201*Temp]}
     SWP1 = (2678 + (R*/$03E4))/(1000-(Temp*12))
     '*****Reading Watermark soil moisture sensor 2:
     HIGH 8                   'Discharge the capacitor
     PAUSE 1                  'for 1 ms
     RCTIME 8,1,T             'Measure RC time charge
     R= (T*15)+(T*/$0199)     'Convert RCtime to resistance (Ohm)
     SWP2 = (2678 + (R*/$03E4))/(1000-(Temp*12))       'Convert resistance to soil water potential (kPa)
    '*****Reading Watermark soil moisture sensor 3:
     HIGH 9                   'Discharge the capacitor
     PAUSE 1                  'for 1 ms
     RCTIME 9,1,T             'Measure RC time charge
     R= (T*15)+(T*/$0199)     'Convert RCtime to resistance (Ohm)
     SWP3 = (2678 + (R*/$03E4))/(1000-(Temp*12))       'Convert resistance to soil water potential (kPa)
RETURN
'*************************************************************************
PRESSURE:                     'Subroutine to measure the hydraulic pressure
    HIGH 11                   'Switch on power to pressure sensor
    PAUSE 200
```

```
    HIGH CS1                    'Deactivate ADC to begin
    PAUSE 500
    HIGH DIO_n                  'Set data pin for first start bit
    configP=configP |%1011      'Set all bits except oddSign.
    LOW CS1                     'Activate the ADC
    SHIFTOUT DIO_n,CLK1,lsbfirst,[configP\4]     'Send config bits
    SHIFTIN DIO_n, CLK1, msbpost,[R\12]          'Get data bits
    HIGH CS1                    'Deactivate the ADC.
    R=R MIN 170                 'Set the minimum R to 170 (pressure = 0)
    P= ((R*/$0306)/100)-3-(1*/$0165)        'Pressure (psi) = 0.03026*R - 4.3947
    LOW 11                      'Switch off power to pressure sensor
RETURN
'*****************************************************************
STORE1:                         'Subroutine to store the time and soil water potential in EEprom
    EEadd=0                     'Read the value of the pointer.LOWBYTE in EEprom address 0
    GOSUB SendAddrEE
    GOSUB InitReadEE
    GOSUB SeqReadEE
    Pointer.LOWBYTE=EEdata
    GOSUB LastReadEE            'Read the value of the pointer.HIGHBYTE in EEprom address 0
    Pointer.HIGHBYTE=EEdata
    Pointer=(Pointer.HIGHBYTE*256+ Pointer.LOWBYTE)          'Calculate the pointer value
    EEadd=Pointer*10+8          'Calculate the address to write to
    EEdata=timeA.lownib(11)*10+timeA.lownib(10)
    GOSUB STORE                 'Store the month in the EEprom
    EEdata=timeA.lownib(9)*10+timeA.lownib(8)
    GOSUB STORE                 'Store the day in the EEprom
    EEdata=timeA.lownib(5)*10+timeA.lownib(4)
    GOSUB STORE                 'Store the hour in the EEprom
    EEdata=timeA.lownib(3)*10+timeA.lownib(2)
    GOSUB STORE                 'Store the min in the EEprom
    EEdata=Temp
    gosub STORE                 'Store soil temperature in the EEprom
    EEdata=SWP1
    GOSUB STORE                 'Store the SWP 1 in the EEprom
    EEdata=SWP2
    GOSUB STORE                 'Store the SWP 2 in the EEprom
    EEdata=SWP3
    GOSUB STORE                 'Store the SWP 3 in the EEprom
    T=EEadd                     'Store the EEprom address
RETURN
'*****************************************************************
STORE2:                         'Subroutine to store the pressure and valve status in EEprom
    EEadd=T                     'Return the EEadd
    EEdata=P
    GOSUB STORE                 'Store the pressure in the EEprom
    EEdata=Valve
    GOSUB STORE                 'Store the valve status in the EEprom
    Pointer=Pointer+1           'Advance the pointer
    IF Pointer=3270 THEN RESET1          'Reset the pointer if EEPROM is full
    Cont2:
    EEdata=Pointer.LOWBYTE
    EEadd=0                     'Store pointer.LOWBYTE  value in EEprom address 0
```

```
    GOSUB STORE
    EEdata=Pointer.HIGHBYTE
    GOSUB STORE                'Store pointer.HIGHBYTE  value in EEprom address 1
    GOSUB BAT                  'Check the battery charging current
GOTO LOOP1                     'Return to the loop
RESET1:
Pointer=0                      'Reset the pointer
GOTO Cont2
'**********************************************************************
BAT:                           'Subroutine to check battery charging current
    LOW 10                     'Switch charger to constant voltage/constant current mode
    SLEEP 40                   'Wait until current stabilize
    HIGH CS1                   'Deactivate ADC to begin
    PAUSE 500
    HIGH DIO_n                 'Set data pin for first start bit
    config=config |%1111       'Set all bits except oddSign, read CH.1.
    LOW CS1                    'Activate the ADC
    SHIFTOUT DIO_n,CLK1,lsbfirst,[config\4]      'Send config bits.
    SHIFTIN DIO_n, CLK1, msbpost,[R\12]          'Get data bits.
    HIGH CS1                   'Deactivate the ADC.
    '*****If current is higher than 24mA continue in const. current/const. Voltage mode
    IF R>585 THEN LOOP1
    HIGH 10                    'If current is <24mA switch to float voltage mode
RETURN
'**********************************************************************
STORE:                         'Subroutine to store data in the EEprom
    GOSUB SendAddrEE
    GOSUB SendDataEE
    GOSUB SendStopEE
    EEadd=EEadd+1              'Advance the address to write
RETURN
'**********************************************************************
ReadTime:                      'Subroutine to read the time in the DS1307
    GOSUB SendAddr
    GOSUB SendStop
    GOSUB InitRead
    GOSUB ReadClk
    GOSUB LastRead
RETURN
'**********************************************************************
SendAddr:
    HIGH Tclk                  'Send Start Condition
    LOW Tdat                   'Send 4-bit device code, 3-bits of
    LOW Tclk                   'null address, and 1 "write" opcode
    SHIFTOUT Tdat,Tclk,MSBFIRST,[%11010000]
    INPUT Tdat                 'Then wait for the acknowledgment
    HIGH Tclk                  '(DS1307 pulls dat line low)
    TAck = IN0(Tdat)           'If DS1307 not ready, try again
    LOW Tclk
    IF TAck = 1 THEN SendAddr  'Shift out 16 bit address
    SHIFTOUT Tdat,Tclk,MSBFIRST,[Tadd.LOWBYTE,%0\1]
    INPUT Tdat
RETURN
```

```
'*************************************************************************
InitRead:
    HIGH Tclk                   'Send start bit
    LOW Tdat                    'Send 4-bit device code, 3-bits of
    LOW Tclk                    'null address, and 1 "read" opcode
    SHIFTOUT Tdat,Tclk,MSBFIRST,[%11010001]
    INPUT Tdat                  'Then wait for the acknowledgment
    HIGH Tclk                   ' (DS1307 pulls dat line low)
    TAck = IN0(Tdat)
    LOW Tclk
    IF TAck = 1 THEN InitRead   'Shift out 16 bit address
RETURN
'*************************************************************************
SendData:
    SHIFTOUT Tdat,Tclk,MSBFIRST,[TData]
    INPUT Tdat                  'Then wait for the acknowledgment
    HIGH Tclk                   '(DS1307 pulls dat line low)
    TAck = IN0(Tdat)            'If DS1307 not ready, try again
    LOW Tclk
RETURN
'*************************************************************************
SeqRead:
    SHIFTIN Tdat,Tclk,MSBPRE,[TData]
    SHIFTOUT Tdat,Tclk,MSBFIRST,[%0\1]
RETURN
'*************************************************************************
LastRead:                       'Special Acknowledge for last byte
    SHIFTIN Tdat,Tclk,MSBPRE,[TData]
    PULSOUT Tclk,1
    GOSUB SendStop
RETURN
'*************************************************************************
SendStop:                       'Send Stop Condition
    LOW Tdat
    HIGH Tclk
    HIGH Tdat
RETURN
'*************************************************************************
INICLK:
    FOR P = 0 to 12 step 2
       Tdata.lownib = timeA.lownib(P)
       Tdata.highnib = timeA.lownib(P+1)
       DEBUG dec ? Tdata
       GOSUB SENDData
    NEXT
    Tdata = 0
    FOR P = 7 to 63
       GOSUB SendData
    NEXT
RETURN
'*************************************************************************
READCLK:
    FOR P = 0 to 6
```

```
        GOSUB SeqRead
        timeA(P) = TData
    NEXT
    DEBUG cls,tab,dec timeA.lowbit(20),dec timeA.lownib(4),":",dec timeA.lownib(3),dec
timeA.lownib(2),":",dec timeA.lownib(1),dec timeA.lownib(0)
    IF timeA.lowbit(22) = 0 THEN DATE
    LOOKUP timeA.lowbit(21),["AP"],SWP1
    DEBUG       tab,str SWP1,"M",cr
    DATE:
    DEBUG tab,dec timeA.lownib(11),dec timeA.lownib(10),"/",dec timeA.lownib(9),dec
timeA.lownib(8),"/",dec       timeA.lownib(13),dec timeA.lownib(12),cr,10
RETURN
'**********************************************************************
ReadEE:                         'Subroutine to read data from a predefined EEprom address
    GOSUB SendAddrEE
    GOSUB InitReadEE
    GOSUB LastReadEE
RETURN
'**********************************************************************
SendAddrEE:
    HIGH EEclk                  'Send Start Condition
    LOW EEdat                   'Send 4-bit device code, 3-bits of
    LOW EEclk                   'null address, and 1 "write" opcode
    SHIFTOUT EEdat,EEclk,MSBFIRST,[%10100000]
    INPUT EEdat                 'Then wait for the acknowledgment
    HIGH EEclk                  ' (24C65 pulls data line low)
    EEAck = IN0(EEdat)          'If EEPROM not ready, try again
    LOW EEclk
    IF EEAck = 1 THEN SendAddrEE        'Shift out 16 bit address
    SHIFTOUT EEdat,EEclk,MSBFIRST,[EEadd.HIGHBYTE,%0\1,EEadd.LOWBYTE,%0\1]
    INPUT EEdat
RETURN
'**********************************************************************
SendDataEE:
    SHIFTOUT EEdat,EEclk,MSBFIRST,[EEData,%0\1]
RETURN
'**********************************************************************
SendStopEE:                     'Send Stop Condition
    LOW EEdat
    HIGH EEclk
    HIGH EEdat
RETURN
'**********************************************************************
InitReadEE:
    HIGH EEclk                  'Send start bit
    LOW EEdat                   'Send 4-bit device code, 3-bits of
    LOW EEclk                   'null address, and 1 "read" opcode
    SHIFTOUT EEdat,EEclk,MSBFIRST,[%10100001]
    INPUT EEdat                 'Then wait for the acknowledgment
    HIGH EEclk                  '(24C65 pulls dat line low)
    EEAck = IN0(EEdat)
    LOW EEclk
RETURN
```

```
'*************************************************************************
SeqReadEE:
    SHIFTIN EEdat,EEclk,MSBPRE,[EEData]
    SHIFTOUT EEdat,EEclk,MSBFIRST,[%0\1]
RETURN
'*************************************************************************
LastReadEE:                         'Special Acknowledg for last byte
    SHIFTIN EEdat,EEclk,MSBPRE,[EEData]
    PULSOUT EEclk,1
    GOSUB SendStopEE
RETURN
'*************************************************************************
```

BASIC Stamp 2 Program Code for *Setclock.bs2*

```
'{ $STAMP BS2 }
'*****************************************************************
' This program set the time of a DS1307 timer from Dallas Semiconductor. *
'*****************************************************************
'*****ASSIGN VARIABLES, CONSTANTS, AND I/O PINS *****
Tadd    VAR   WORD        'Address to read from or write to in DS1307
Tdata   VAR   BYTE        'Data read from or written to DS1307
Tack    VAR   BIT
i       VAR   BYTE        'Miscellaneous counter
j       VAR   BYTE        'Miscellaneous counter
timeA    var  byte(7)
PM      var   byte
Tclk    CON   1           'Time clock line
Tdat    CON   0           'Time data line
OUTA = %0011              'Initialize: Tclk = High
DIRA = %0001              'Initialize: Tdat = Input
'*****************************************************************
DEBUG "Set Clock ? Y/N",CR    'Set CLOCK?
PAUSE 500
SERIN 16,16468,6000,CLOCK,[str j\1]
IF j = "N" OR j = "n" THEN CLOCK
TIMEIN:Debug cls,"Enter time as HH:MM:SS",cr
SERIN 16,16468,[dec1 timeA.lownib(5),dec1 timeA.lownib(4),skip 1,dec1 timeA.lownib(3),dec1
timeA.lownib(2),skip 1,dec1 timeA.lownib(1),dec1 timeA.lownib(0)]
DEBUG cr,"You entered : "
DEBUG dec timeA.lowbit(20),dec timeA.lownib(4),":",dec timeA.lownib(3),dec timeA.lownib(2),":",dec
timeA.lownib(1),dec timeA.lownib(0)
DEBUG cr,"Correct Y/N ",cr
SERIN 16,16468,[str j\1]
IF j = "N" or j = "n" THEN TIMEIN
TIMEH:Debug cls,"Enter 12/24 hr time",cr
SERIN 16,16468,[dec i]
DEBUG cr,"You entered : ",dec i,cr
DEBUG "Correct Y/N ",cr
SERIN 16,16468,[str j\1]
IF j = "N" or j = "n" THEN TIMEH
IF i=24 THEN setH
timeA.lowbit(22) = 1
TIMEP:Debug cls,"Enter AM = A or PM = P ",cr
SERIN 16,16468,[str PM\1]
DEBUG cr,"You entered : ",str PM,cr
DEBUG "Correct Y/N ",cr
SERIN 16,16468,[str j\1]
IF j = "N" or j = "n" THEN TIMEP
LOOKDOWN PM,["AP"],PM
LOOKDOWN PM,["ap"],PM
DEBUG dec ? PM
timeA.lowbit(21)= PM
GOTO setD
setH: timeA.lowbit(22) = 0
DEBUG dec ? timeA.lowbit(22)
```

```
setD:
DEBUG cls,"Enter Weekday as # :Sun=1/Mon=2/Tues=3/Wednes=4/Thur=5/Fri=6/Satur=7",cr
SERIN 16,16468,[dec1 timeA.lownib(6)]
DEBUG cr,"You entered : ",dec timeA.lownib(6),cr
DEBUG "Correct Y/N ",cr
SERIN 16,16468,[str j\1]
IF j = "N" or j = "n" THEN setD
T_date:Debug cls,"Enter Date as MM/DD/YY",cr
SERIN 16,16468,[dec1 timeA.lownib(11),dec1 timeA.lownib(10),skip 1,dec1 timeA.lownib(9),dec1
timeA.lownib(8),skip 1,dec1 timeA.lownib(13),dec1 timeA.lownib(12)]
DEBUG cr,"You entered : ",cr
DEBUG tab,dec timeA.lownib(11),dec timeA.lownib(10),"/",dec timeA.lownib(9),dec
timeA.lownib(8),"/",dec timeA.lownib(13),dec timeA.lownib(12),cr
DEBUG "Correct Y/N ",cr
SERIN 16,16468,[str j\1]
IF i = "N" or i = "n" THEN T_date
TAdd = 0
GOSUB SendAddr
GOSUB INICLK
GOSUB SendStop
PAUSE 5000
CLOCK:
TAdd = 0
GOSUB SendAddr
GOSUB SendStop
GOSUB InitRead
GOSUB ReadClk
GOSUB LastRead
PAUSE 5000
SLEEP 1563
GOTO CLOCK
END
'**********************************************************************
SendAddr:
HIGH Tclk                          'Send Start Condition
LOW Tdat                           'Send 4-bit device code, 3-bits of
LOW Tclk                           'null address, and 1 "write" opcode
SHIFTOUT Tdat,Tclk,MSBFIRST,[%11010000]
INPUT Tdat                         'Then wait for the acknowledgment
HIGH Tclk                          '(DS1307 pulls dat line low)
TAck = IN0(Tdat)                   'If DS1307 not ready, try again
DEBUG ? TAck
LOW Tclk
IF TAck = 1 THEN SendAddr          ' Shift out 16 bit address
SHIFTOUT Tdat,Tclk,MSBFIRST,[Tadd.LOWBYTE,%0\1]
INPUT Tdat
RETURN
'**********************************************************************
InitRead:
HIGH Tclk                          'Send start bit
LOW Tdat                           'Send 4-bit device code, 3-bits of
LOW Tclk                           'null address, and 1 "read" Opcode
SHIFTOUT Tdat,Tclk,MSBFIRST,[%11010001]
```

```
INPUT Tdat                    'Then wait for the acknowledgment
HIGH Tclk                     ' (DS1307 pulls data line low)
TAck = IN0(Tdat)
LOW Tclk
IF TAck = 1 THEN InitRead     ' Shift out 16 bit address
RETURN
'**************************************************************************
SendData:
SHIFTOUT Tdat,Tclk,MSBFIRST,[TData]
INPUT Tdat                    'Then wait for the acknowledgment
HIGH Tclk                     '(DS1307 pulls data line low)
TAck = IN0(Tdat)              'If DS1307 not ready, try again
LOW Tclk
RETURN
'**************************************************************************
SeqRead:
SHIFTIN Tdat,Tclk,MSBPRE,[TData]
SHIFTOUT Tdat,Tclk,MSBFIRST,[%0\1]
RETURN
'**************************************************************************
LastRead                      'Special Acknowledg for last byte
SHIFTIN Tdat,Tclk,MSBPRE,[TData]
PULSOUT Tclk,1
GOSUB SendStop
RETURN
'**************************************************************************
SendStop
LOW Tdat                      'Send Stop Condition
HIGH Tclk
HIGH Tdat
RETURN
'**************************************************************************
INICLK:
FOR i = 0 TO 12 step 2
  Tdata.lownib = timeA.lownib(i)
  Tdata.highnib = timeA.lownib(i+1)
  DEBUG dec ? Tdata
  GOSUB SENDData
NEXT
Tdata = 0
FOR i = 7 TO 63
  GOSUB SendData
NEXT
RETURN
'**************************************************************************
READCLK:
FOR i = 0 to 6
  GOSUB SeqRead
  timeA(i) = TData
NEXT
DEBUG cls,tab,dec timeA.lowbit(20),dec timeA.lownib(4),":",dec timeA.lownib(3),dec
timeA.lownib(2),":",dec timeA.lownib(1),dec timeA.lownib(0)
IF timeA.lowbit(22) = 0 THEN DATE
```

```
LOOKUP timeA.lowbit(21),["AP"],PM
DEBUG tab,str PM,"M",cr
DATE:  debug    tab,dec timeA.lownib(11),dec timeA.lownib(10),"/",dec timeA.lownib(9),dec
timeA.lownib(8),"/",dec timeA.lownib(13),dec timeA.lownib(12),cr
        branch timeA.lownib(6)-1,[SUN,MON,TUE,WED,THU,FRI,SAT]
EREADCLK:     return
'*********************************************************************
'Day write below
SUN:
DEBUG tab,"SUNDAY",cr
GOTO EREADCLK
MON:
DEBUG tab,"MONDAY",cr
GOTO EREADCLK
TUE:
DEBUG tab,"TUESDAY",cr
GOTO EREADCLK
WED:
DEBUG tab,"WEDNESDAY",cr
GOTO EREADCLK
THU:
DEBUG tab,"THURSDAY",cr
GOTO EREADCLK
FRI:
DEBUG tab,"FRIDAY",cr
GOTO EREADCLK
SAT:
DEBUG tab,"SATURDAY",cr
GOTO EREADCLK
```

Campbell Scientific 21X Data Logger Code for *Tensiom.dld*

```
};21X
MODE 1
SCAN RATE 30
1:P92
1:0
2:15
3:10

2:P10
1:1

3:P17
1:2

4:P86
1:41

5:P87
1:0
2:12

6:P86
1:72

7:P4
1:1
2:5
3:1
4:1
5:5
6:5000
7:3--
8:1
9:0

8:P95

9:P86
1:51

10:P86
1:41

11:P87
1:0
2:12

12:P86
1:72

13:P4
```

```
1:1
2:5
3:2
4:2
5:5
6:5000
7:15--
8:1
9:0

14:P95

15:P86
1:51

16:P11
1:1
2:3
3:3
4:27
5:1
6:0

17:P77
1:1110

18:P71
1:27
2:1
```

APPENDIX C

IRRIGATION CONTROLLER POWER SUPPLY OPTIMIZATION

Battery

The battery capacity to be used by the controller should be chosen considering the expected number of consecutive days in the region with no significant solar radiation. The daily load consumed by the irrigation controller is 0.29 Ah d^{-1}, and a battery efficiency of 70% may be used to determine the recommended battery capacity.

Commercially available SLA battery capacities and the expected number of days that the irrigation controller could operate without external charging are shown in Table C-1. For example, a minimum battery capacity of 2.2 Ah would be required to operate the controller without interruption during winter in Knoxville, TN (latitude 36° 88' N).

Solar Panel

According to Laws (1983) the solar panel rating (in peak Watt) can be calculated by dividing the daily power requirement of the load by the average peak sun hours per day for the region, and adding a 20% safety factor. Considering that the irrigation controller requires 3.5 Wh d^{-1} of power and that the annual average peak sun hours for Knoxville, TN is 4.5 h d^{-1}, the calculated solar panel rating should be 1 W.

Table C-1. Available battery capacities and expected irrigation controller operation without battery recharge.

Battery Capacity (Ah)	Number of days of operation without recharge*
1.2	2.9
2.2	5.3
2.8	6.8
3.4	8.2

* Number of days = (battery capacity * efficiency) / daily load

However, these calculations consider annual averages and depending on the region, solar radiation can vary significantly over the year. For a location like Knoxville, TN, the average solar radiation during winter is 22% lower than the annual average, and four or more days of very low solar radiation can occur. As a result, designing the solar panel considering the average annual solar radiation would require a battery of large capacity to compensate for seasonal and occasional variations in solar radiation.

The calculations also assume an average of 4.5 h per day of full sun (1000 W m^{-2}), when in fact it is reasonable to assume that most sites will actually average about 85% of full sun. The conversion efficiency of most solar panels averages about 11% to 15%. Dust, water vapor, air pollution, seasonal variations, altitude, and temperature all affect how much power the solar panel actually receives.

In order to design the optimum solar panel for the irrigation controller using a more realistic approach, a simulation model for the state of battery charge was developed to be applied to actual solar radiation data. The model was developed as a spreadsheet and was based on the conversion efficiency of the Siemens ST-5 solar panel, operating with the charge controller circuit developed for the irrigation controller.

The conversion efficiency of the controller's power supply system was calculated comparing hourly solar radiation data (Wh m^{-2}) from February 2003, corrected for the surface area of the solar panel Siemens ST-5, with the power going into the battery (W h), calculated as the measured charging current multiplied by the battery voltage. The conversion efficiency varied from 1% to 16% depending on the demand for power. When the battery was fully charged the demand was low, resulting in a lower efficiency. The

average conversion efficiency of the solar panel and the charge controller circuit observed during the battery charging process was 5%.

The battery state of charge (% of full capacity) was calculated hourly by dividing the battery power available by the battery full capacity. The battery power available was calculated according to the following equation:

$$BP_i = BP_{i-1} + (SR \times A \times Ef) - PL$$

Where,

BP_i = Battery power available, Wh;

BP_{i-1} = Battery power the hour before, Wh;

SR = Average solar radiation during the last hour, Wh m^{-2};

A = Solar panel surface area, m^2;

Ef = Conversion efficiency of the solar panel and the charge controller circuit, decimal;

PL = Power used by the load (12 V x 0.0121 Ah), Wh;

The simulation model was tested using hourly solar radiation data from Knoxville, TN, and battery voltage data measured simultaneously during February 2003. According to Buresch (1983), the state of charge of a battery is closely related to its voltage.

Figure C-1 shows simulated battery capacity and measured battery voltage. The graph indicated that the model performed well in tracking battery voltage variations. It was observed that low values of battery voltage (below the voltage required to operate the irrigation controller) occurred when the simulated battery charge was about 50% of full capacity.

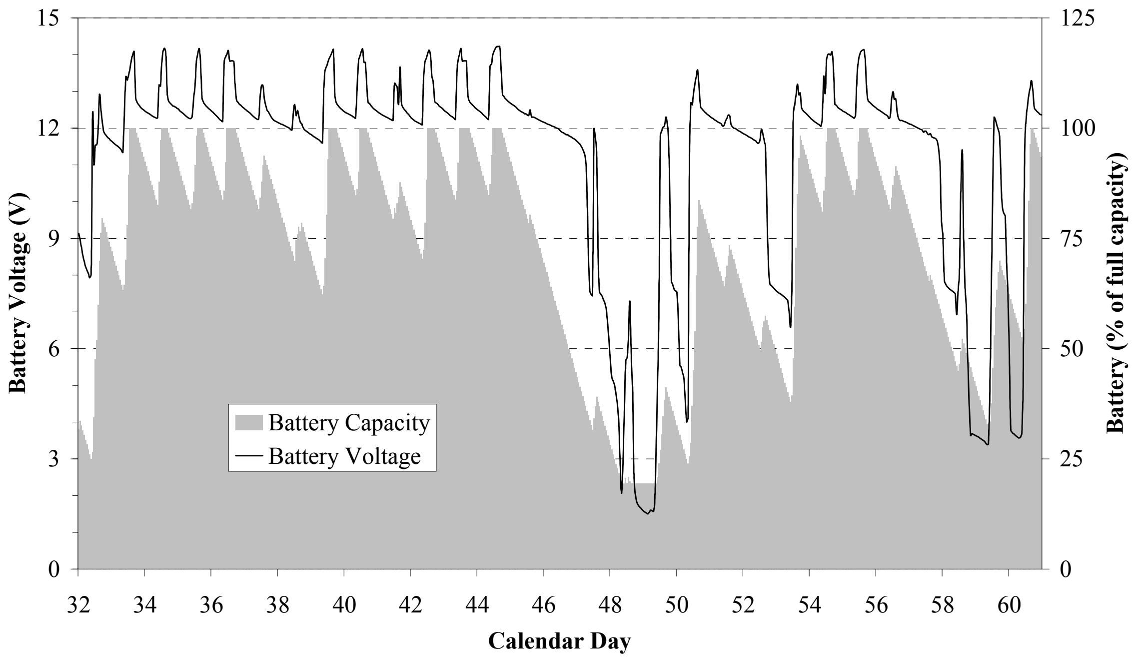


Figure C-1. Measured battery voltage and simulated percentage of battery capacity for Knoxville, TN, 2003.

The simulation model was then applied to solar radiation data from two locations with different climates: Knoxville, TN (latitude 36° 88' N) and Ceara, Brazil (latitude 3° 43' S). For Knoxville solar radiation data from March to September of 2002 were used, period when crops may require irrigation. For Ceara, Brazil, data from January to June of 1998 were used in the simulation. For that region irrigation is required almost for the entire year, but a higher probability of cloudy days is expected in the first six months of the year.

The performances of four solar panel power ratings (1.5 W, 2.0 W, 2.5 W, and 5 W), combined with two battery capacities (1.2 Ah and 2.2 Ah) were simulated using the model. The simulations assumed that all solar panels had the same conversion efficiency as the Siemens ST-5 solar panel, and that differences among solar panel power ratings

were due to their surface areas. Each simulation began with the battery at full capacity on the first day.

For each location, the simulation results were analyzed in terms of the percentage of battery capacity used and cost of the components. A maximum use of 50% of the battery capacity, anytime during the period evaluated was considered the threshold to accept or reject the solar panel-battery combination. The solar panel-battery combination that met that design criteria and presented minimum cost was selected as the optimum combination for each region.

Table C-2 shows a summary of the simulation results for Knoxville, TN. According to the simulation, when using the 5-W solar panel and the 1.2-Ah battery in Knoxville, TN, the battery state of charge reduced to less than 50% on two occasions during the period evaluated (Figure C-2). This could cause the operation of the irrigation controller to be interrupted during some days of the spring season, and could reduce the battery life. These results confirmed that the 1.2-Ah battery was not the best option for the region.

Table C-2. Battery state of charge simulation results for Knoxville, TN.

Battery Capacity (Ah)	Solar Panel Rating (W)	Average Battery Charge (%)	Minimum Battery Charge (%)	Days with Charge < 50%
1.2	1.5	65	0	75
	2.0	85	0	15
	2.5	90	9	6
	5.0	93	44	2
2.2	1.5	65	0	71
	2.0	91	34	4
	2.5	94	51	0
	5.0	96	69	0

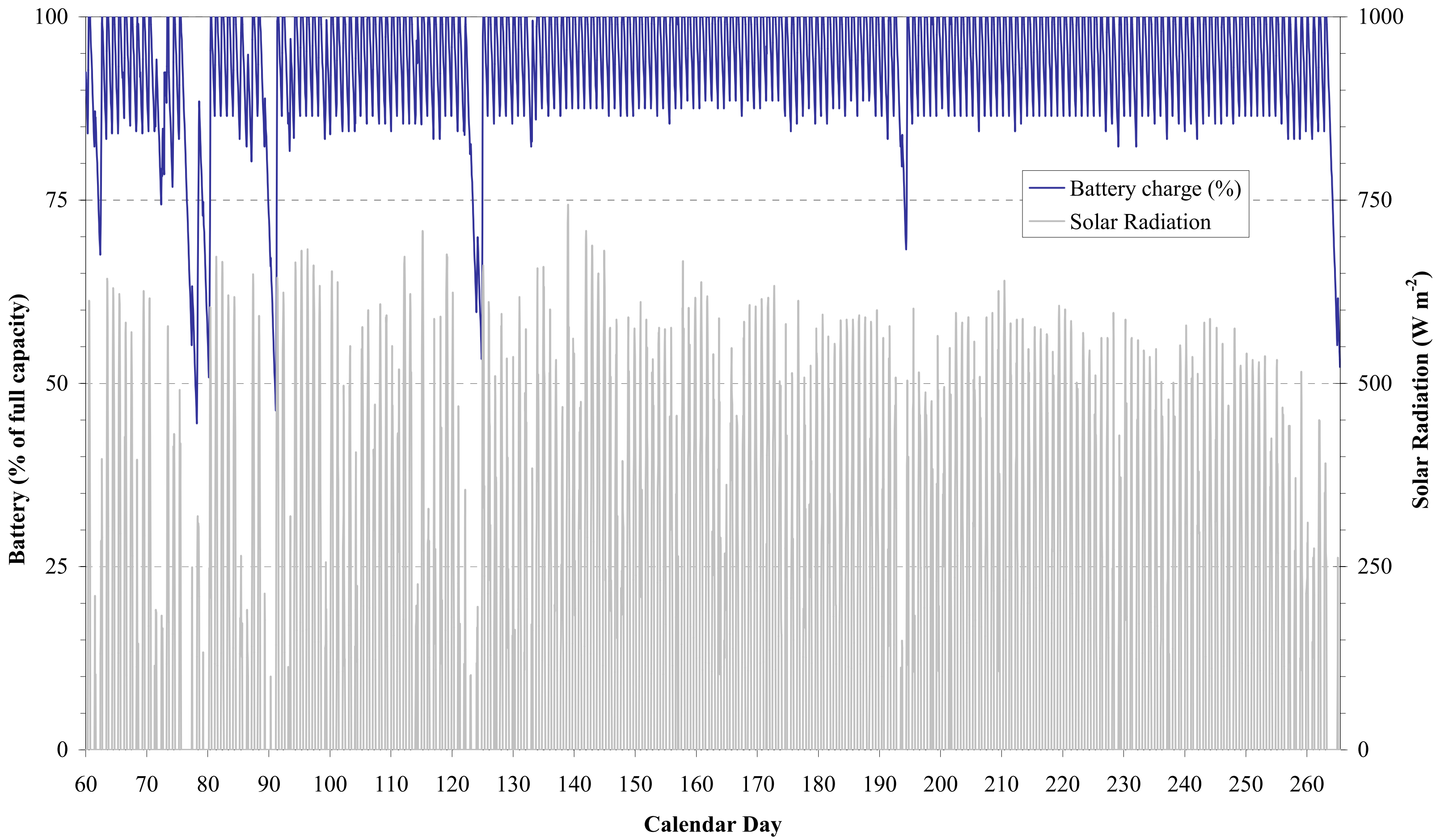


Figure C-2. Hourly solar radiation and simulated battery state of charge for a 1.2-Ah battery and a 5-W solar panel, for Knoxville, TN, 2002.

The optimum solar panel-battery combination for Knoxville was the 2.5-W solar panel operating with the 2.2-Ah battery. The simulation results showed that although the battery charging was slower than for the 5-W solar panel, the 2.2-Ah battery provided enough storage capacity to compensate for solar radiation variations. Even using a much smaller solar panel, the battery state of charge was always maintained above 50%, and most of the time above 75% (Figure C-3).

The 2.5-W solar panel operating with the 2.2-Ah battery also represented a more economical option compared to the combination currently being used. In January of 2003 the price difference between the 1.2-Ah battery and the 2.2-Ah battery was only $3, while the cost of solar panels in the 2-W to 5-W range varied from $10 to $12 per watt at peak output. A system with a 2.5-W solar panel and a 2.2-Ah battery would cost about $45, which represented 65% of the cost of the 5-W solar panel, 1.2-Ah battery combination ($69).

The results for Ceara, Brazil, indicated that for that climate a 1.2-Ah battery used with a 1.5-W solar panel would be sufficient to power the irrigation controller (Table C-3). For that region the average solar radiation (5.8 kWh m^{-2} d^{-1}) is about 30% higher than that observed in Knoxville, requiring a smaller solar panel. Even during the rainy season, more than two consecutive days without significant solar radiation did not occur during 1998, and the 1.2-Ah battery would provide enough storage capacity to keep the controller working (Figure C-4). The estimated cost of this solar panel-battery combination would be approximately $35.

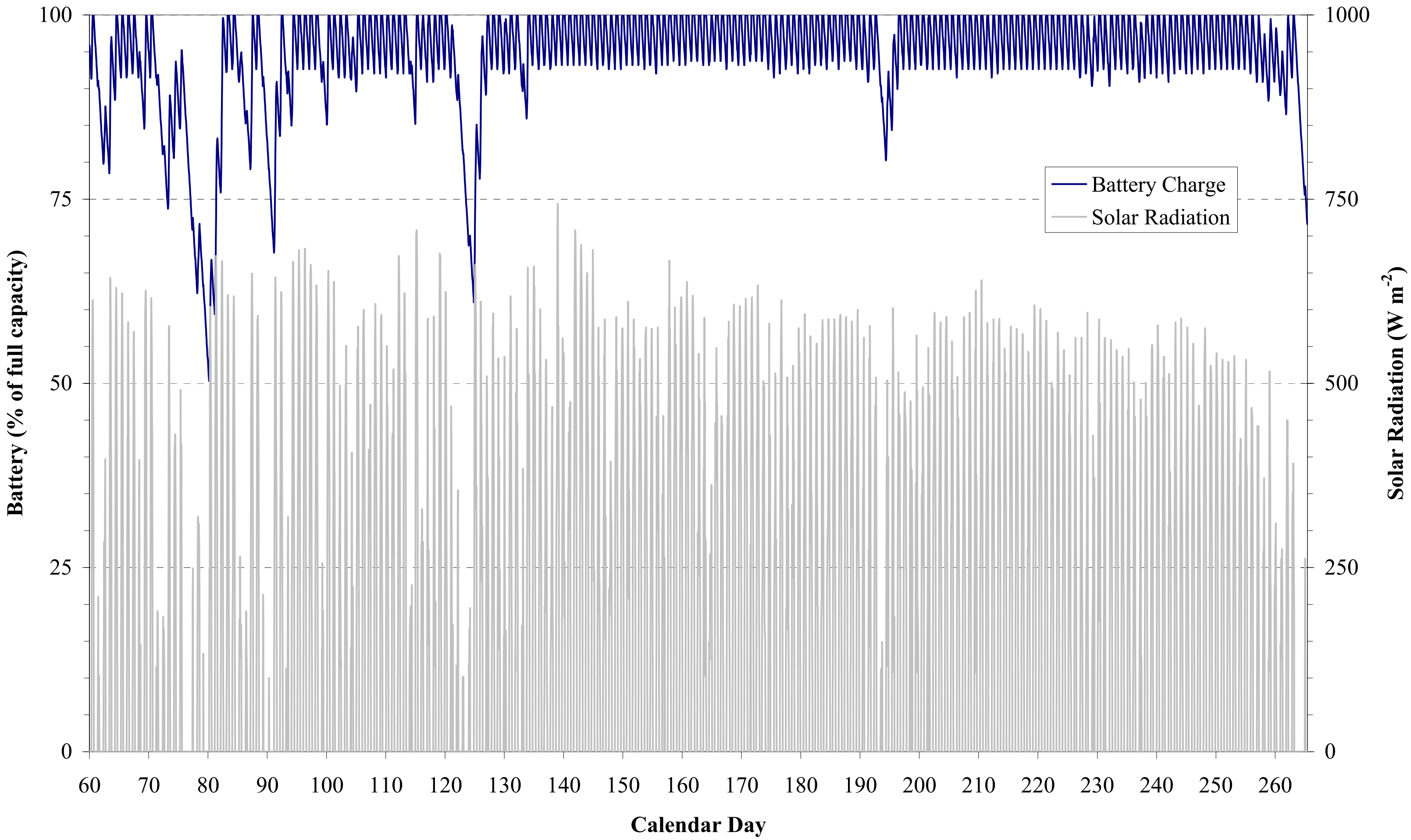


Figure C-3. Hourly solar radiation and simulated battery state of charge for a 2.2-Ah battery and a 2.5-W solar panel, for Knoxville, TN, 2002.

Table C-3. Battery state of charge simulation results for Ceara, Brazil.

Battery Capacity (Ah)	Solar Panel Rating (W)	Average Battery Charge (%)	Minimum Battery Charge (%)	Days with Charge < 50%
1.2	1.5	91	51	0
	2.0	93	66	0
	2.5	94	70	0
	5.0	95	78	0
2.2	1.5	95	73	0
	2.0	96	82	0
	2.5	97	83	0
	5.0	97	88	0

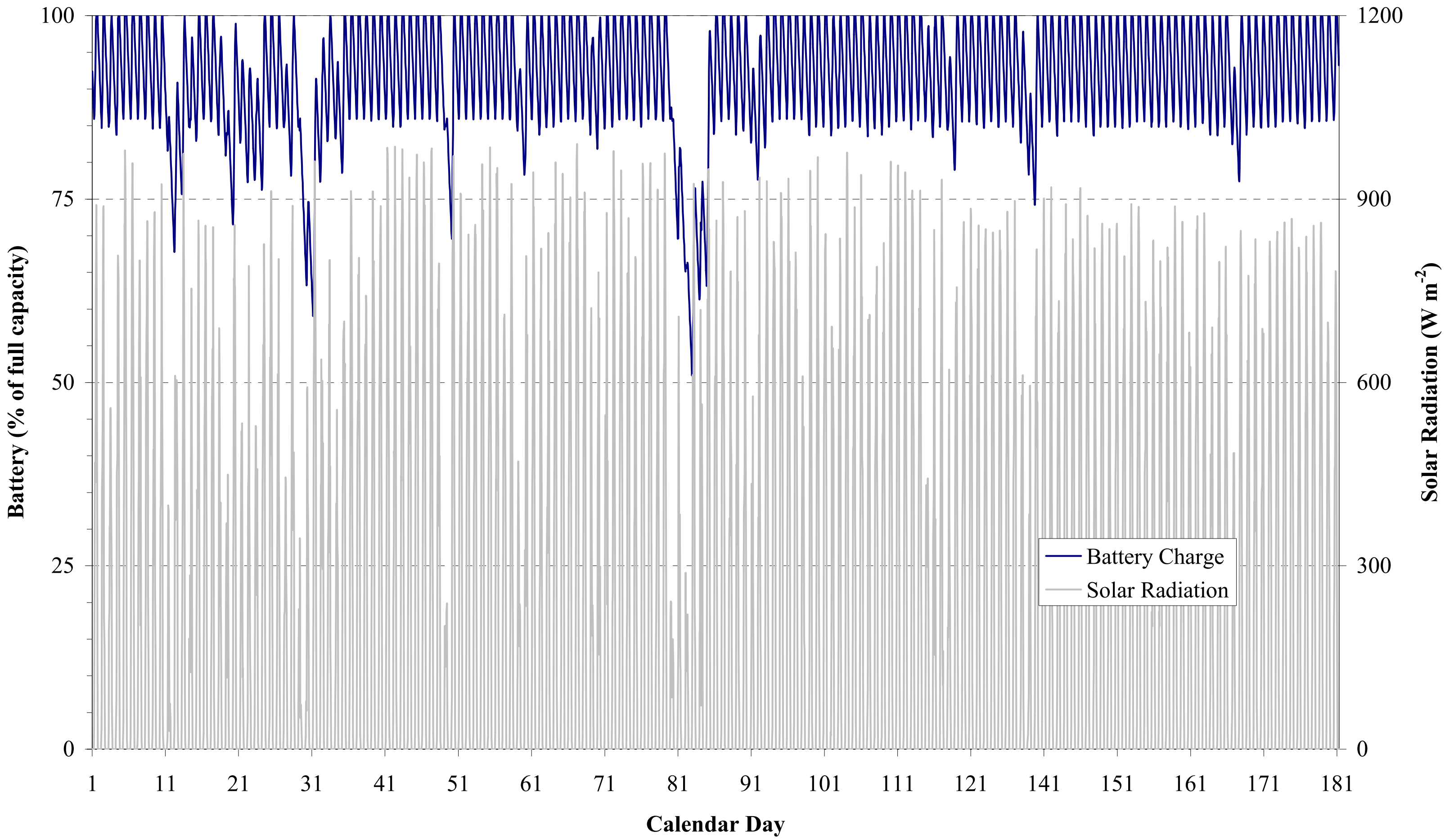


Figure C-4. Hourly solar radiation and simulated battery state of charge for a 1.2-Ah battery and a 1.5-W solar panel, for Ceara, Brazil, 1998.

APPENDIX D

CLIMATIC DATA

Table D-1. Climatic data recorded for Knoxville, TN, 2002.

Calendar Day	Air Temp. (°C)	Relative Humidity (%)	Solar Radiation (kWh m^{-2})	Wind Speed (m zs^{-1})	Rainfall (mm)	Evapotrans-piration* (mm)
196	24.8	86.5	4.5	0.57	0.0	3.6
197	26.3	77.7	4.8	0.56	0.0	4.7
198	26.5	80.9	3.7	0.54	1.0	4.0
199	26.5	79.2	3.9	0.69	0.0	4.3
200	25.3	80.9	4.2	0.91	2.3	4.5
201	24.2	90.0	3.3	0.48	14.7	3.5
202	26.9	77.3	5.1	0.60	0.0	5.3
203	27.2	76.5	4.3	0.61	0.0	4.6
204	26.2	77.4	4.7	0.83	17.8	5.0
205	25.6	81.5	3.9	0.50	0.0	4.1
206	26.3	82.7	3.7	0.65	0.0	3.9
207	27.1	78.5	4.3	0.76	0.0	4.7
208	28.0	77.3	4.6	0.80	0.0	5.0
209	27.0	83.3	4.0	0.60	5.8	4.2
210	28.4	76.3	5.1	0.90	0.0	5.5
211	26.1	81.0	4.2	0.94	15.2	4.7
212	25.4	83.1	4.0	0.62	0.0	4.1
213	27.1	75.4	4.7	0.75	0.0	5.0
214	28.2	68.5	4.5	0.90	5.8	5.1
215	27.3	72.8	4.8	0.87	0.3	5.3
216	28.0	66.6	4.8	0.84	0.0	5.3
217	27.8	70.7	4.7	0.65	0.0	5.1
218	27.4	63.5	4.4	1.75	0.0	5.5
219	25.2	47.6	5.2	1.48	0.0	5.9
220	25.2	47.1	5.1	1.03	0.0	5.3
221	25.3	60.0	4.8	0.71	0.0	4.9
222	25.8	67.3	4.1	0.83	0.0	4.5
223	26.7	65.8	4.5	0.77	0.0	4.8
224	27.1	64.7	4.3	0.63	0.0	4.6
225	26.7	65.1	3.8	0.58	0.0	4.1
226	26.2	69.3	3.7	0.80	0.0	4.2
227	26.1	77.9	3.7	0.76	0.3	4.1
228	24.8	85.3	3.0	0.64	0.8	3.3
229	24.9	86.7	2.8	0.82	6.1	3.1
230	26.0	80.5	4.0	0.76	0.0	4.2
231	23.7	90.1	2.7	0.58	4.6	3.0
232	26.1	75.8	4.2	0.84	0.0	4.5
233	27.8	65.0	4.4	0.73	0.0	4.8

Table D-1. Continued.

234	28.5	69.1	4.2	0.76	0.0	4.7
235	28.2	73.5	3.9	0.82	0.5	4.4
236	27.7	76.0	3.5	0.79	0.0	3.9
237	24.5	87.2	2.4	0.54	27.9	2.7
238	23.9	80.1	3.7	1.04	1.0	3.9
239	23.1	79.5	3.8	0.84	0.0	3.9
240	24.5	70.4	4.4	1.50	0.0	4.7
241	24.8	71.3	3.7	0.92	0.0	3.9
242	25.7	71.2	3.4	0.85	0.0	3.6
243	25.6	70.6	4.3	0.83	0.0	4.3

* Reference evapotranspiration (ETo) calculated according to the FAO-Penman-Monteith equation, and using the REF-ET Reference Evapotranspiration Software, from R.G. Allen, University of Idaho.

VITA

Fabio Rodrigues de Miranda was born in Sao Pedro dos Ferros, State of Minas Gerais, Brazil on April 19, 1967. He was raised in Rio de Janeiro, Brazil where he attended elementary, middle, and high school. He entered the Federal University of Viçosa, State of Minas Gerais, in 1985 and graduated in March of 1990 with a Bachelor of Science in Agronomic Engineering. He continued his studies at the Federal University of Viçosa and in September of 1992 he received a Master of Science degree in Agricultural Engineering. In September 1992, he took a position with the Minas Gerais Agricultural Research Corporation, as a researcher. In November 1994, he obtained a position with the Brazilian Agriculture Research Corporation, working as a researcher at the Tropical Agro-industry Research Center in Fortaleza, Ceará, Brazil. In August 1999, he was granted a scholarship from the Brazilian National Council for Scientific and Technological Development – CNPq, to pursue his doctoral degree at The University of Tennessee, Knoxville. The doctoral degree was awarded in August of 2003.