Ministry of Higher Education and Scientific Research

University M'Hamed Bouguerra
Boumerdes

Faculty of Engineering
Department of Electrical and Electronic Engineering

Report

Presented in Partial Fulfillment of the Requirements of the

MAGISTER DEGREE

In Electronic Systems Engineering

Design and Implementation of an Interrupt–Driven Microprocessor–Based Solar/Electrical Irrigation System

By
BENZEKRI Azzouz

Defended before the Jury:

President: Dr. REFOUFI. Larbi............Professeur..........U.M.B.B

Members: Dr. BENAZZOUZ. Djamel.....M. C..................U.M.B.B
Dr. HAMAMI. Latifa...........M. C..................E.N.P
Dr. KHALFI. Kamel...........C. C..................U.M.B.B

Design and Implementation of an Interrupt-Driven Microprocessor-Based Solar/Electrical Irrigation System

By

Azzouz Benzekri

Maitre Assistant University of Boumerdes

A Report Presented in Partial Fulfillment
Of the Requirements for the
Degree of Magister

UNIVERSITY OF BOUMERDES

TABLE OF CONTENTS

Table of Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

ABSTRACT

Abstract

The theory of soil, water, plants and atmosphere interaction is considered. Different methods of soil-water measurement as well as different irrigation scheduling approaches are studied. The design and implementation of a portable and autonomous Interrupt-Driven Microprocessor-Controlled Irrigation System (field controller) are developed. Full circuit and program codes are implemented to verify system operation. The field controller consists of soil moisture sensors, a hardware input/output (I/O) interface, a microprocessor as the central player and a serial communication means with a host computer.

The firmware enables the system to continuously measure in-situ soil moisture content and climatic parameters, processes them along with other data and commands entered by the user. For ease of operation, all parameter setpoints and commands are selectable through a hexadecimal keyboard. Based on this data processing, the system makes a decision to either initiate irrigation or not. Provisions have been made such that the system's self-diagnostic provides a status indication (visual and buzzing) and an automatic shutdown of the irrigation system for any failure detected in either sensor or solenoid valve.

The system features three different operating modes to meet different agricultural practices. The first is the 'automatic' mode. In this mode, the grower bases the ignition of the irrigation on soil moisture content, climatic parameters and desired set points determined. In the second mode, the 'timer' mode, the irrigation is independent on soil moisture content, the grower has the ability to initiate an irrigation for a predetermined duration, provided that certain climatic parameters are suitable for the irrigation process. The irrigation terminates when programmed time has elapsed. In the third operating mode, the 'manual' mode, the irrigation can be initiated and stopped by the farmer at any time irrelevant of soil and climatic parameters.

The system is serially connected to a Personal Computer by mean of a bi-directional communication link, where field status (soil and climatic parameter) is continuously sent to the host PC via its serial port. The values of soil and climatic parameters are stored and displayed on the PC's screen. The system is designed to allow the user to stop irrigation at any time either in-situ or remotely by a simple click on the mouse.

ملخص

قد تم التطرق في هذه المذكرة إلى الدراسة النظرية لمختلف التفاعلات بين التربة، الماء، النباتات، والمناخ ؛ وكذالك مختلف الطرق قياس نسبة الماء في التربة و طرق السقي. تطوير و تجسيم منظومة سقي محمولة مراقبة من خلال المعالجة الدقيقة؛ ويتم تشغيلها عن طريق دائرة الكترونية تعمل ببرنامج معين. المنظومة تتكون من متحسس رطوبة التربة، مكونات تركيبية مدخل / مخرج مع ترابط منسجم معالجة دقيقة كأساس للمنظومة وكذالك وسيلة اتصالات متسلسلة مع حاسبة رقمية.

يتم تشغيل المنظومة عن طريق المستعمل لأخذ قياسات مستمرة لنسبة الرطوبة و العوامل المناخية ثم معالجتها مع المعلومات الأخرى والتوجيهات المطلوبة من المستعمل وبعد ذالك يتم أخذ قرار البدء في السقي أو عدمه. في حالة ظهور خلل ما، فإن المنظومة تقوم بتوقيف عملية السقي تلقائيا.

المنظومة لها ثلاث صيغ تشغيل مختلفة: الصيغة الأولى هي الصيغة التلقائية، في هذه الحالة رطوبة التربة و العوامل المناخية هي المهمة لبدء عملية السقي. فأما الثانية هي الصيغة الزمنية وفيها تكون عملية السقي مستقلة عن نسبة رطوبة التربة و العوامل الأخرى وتستمر لمدة معينة وبعدها تتوقف. و الصيغة الثالثة تتم عن طريق المستعمل نفسه، في هذه الحالة يتم السقي في أي وقت و كذلك توقفها و لا دخل فيها للعوامل الأخرى.

يمكن استعمال المنظومة و تشغيلها عن بعد عن طريق حاسبة رقمية التي يتم برمجتها لتخزين المعلومات المتوفرة حول رطوبة التربة والعوامل المناخية و كذلك للسماح للمستعمل توقيف عملية السقي في أي وقت يريده.

ACKNOWLEDGMENTS

Acknowledgments

Special gratitude is expressed to my supervisors Dr Djamel BENAZZOUZ and Dr Hadj BOURDOUCEN for their help and encouragement.

I would like to thank my colleague A. AZRAR for his support and encouragement.

Finally, I would like to dedicate this work to those who believe that:

" *Significant advances in engineering often require both the* ***push*** *of technology and the* ***pull*** *of an application that offers some economic and societal benefits* ".

CHAPTER 1

Introduction

1.1 Overview

Irrigating crops is an agricultural practice that goes back thousands of years in human history. Despite significant advances in technology over time, the basic purpose of irrigation is much the same: to supplement water available through rainfall, snow and snowmelt for the purpose of increasing crop yields and crop quality.

Water is a vital element, which affects all aspects of life particularly in agriculture. Throughout the world, irrigation is the most important user of water (except for drinking). Almost 60% of all the world's fresh water withdrawals go towards irrigation uses [78]. This most precious resource is becoming limited in quantity as population growth together with changes in lifestyle has heightened the pressure on water resources.

Water scarcity may cause famines. As human population has grown the demand for resources of all kinds has also grown. In a large number of developing countries, population growth exceeds growth in the production of basic grains and other crops. Supporting more people means producing more food, which in turn requires greater amount of energy, soil nutrients, water and other resources associated with agricultural production. Due to water and energy resources limitations, irrigation - an artificial application of water to the soil - should be planned and managed in such a way to optimize the use of every drop of water to meet the crop's water requirements when natural rainfall is inadequate.

To meet the world food supplies in 2025, the study [78] provides 2 scenarios:

i- no increase in irrigation efficiency

ii- a scenario where irrigation efficiency is dramatically increased.

If scenario (i) is applied, then 60% more water will be required for irrigation to meet the world food supplies in 2025. Even if irrigation efficiency is greatly increased, more than 10% more water will be needed.

The primary source of water for agriculture production for most of the world is rainfall. Three main characteristics of rainfall are its amount, frequency and intensity, the values of which vary from place to place, day to day, month to month and also year to year.

Because of such complexities in the precise knowledge of the rainfall's main characteristics, the irrigation scheduling cannot be planned neither on the minimum values of rainfall nor on maximum. The former may lead to an over-irrigation causing crop diseases and waste of both water and energy, while the latter, to an under-irrigation causing a highly reduction of yields and crops quality.

Algeria, as most countries in North Africa belongs to the hydraulic poor countries. Algeria faces problems of water shortage due to the semi-arid climate as well as an increasing demand created by population and economical growth. Its scarcity and spatial and temporal variability characterize rainfall in Algeria. The average annual rainfall is about 68 mm [19]. Even the more humid areas where the average precipitation can reach 1500mm, dry seasons might last five months. Temperature and solar radiation are rather high, that leads to a high evaporation and lower water resources. While conventional water availability remains relatively constant, the demand is increasing as a result of population growth, industrial development and irrigation expansion. The increased frequency and longer duration of droughts exacerbate water scarcity.

To obtain optimum results from modern irrigation systems water management at the farm level is necessary. To accomplish this, it necessary to help growers decide when to irrigate, how much water to apply and when to apply it, and how to prevent environmental degradation especially with respect to soil and water resources. Therefore, supplementary irrigation should be based on a procedure that takes into account soil, crop and climate factors.

To maximize returns from irrigation development and from efficient water application, there is a need to properly schedule irrigation. Irrigation scheduling methodologies based on sound scientific practice are gaining greater importance, as fresh water is becoming limited [29], [38], [71]. Many methods exist to schedule irrigation. They are based on soil, plant and climatic measurements. Although it is common knowledge that irrigation scheduling is already applied successfully by some farmers throughout the world, it is also a fact that the majority is not scheduling at all. The primary reason why farmers do not schedule is because they do not perceive the net benefit of scheduling to be positive. The challenge to researchers is to develop methodologies and useful systems so that the irrigation community can use them. Otherwise, the research developments and efforts are wasted.

Scientists are more inclined to design their research projects with the view of producing publications, on the other hand, producers want immediate answers to local problems, they are not interested in experimental details [71].

The remaining of this chapter is organized as follows: we first present the motivation for our work in section 1.2. The goal and contribution of the work are summarized in section 1.3. Finally, in section 1.4, the organization of the rest of this report is outlined.

1.2 Motivation

Prior to the introduction of modern irrigation systems, all the agricultural land was irrigated by traditional flood and furrow irrigation methods. Methods that are economical, if water resources are abundant; this was the case in the past when the world's population was small and most of it rural and the industry almost inexistent. Today such irrigation methods are obsolete though they are still used in many parts of the world.

Water is expected to be one of the most critical natural resources in the twenty-first century. The prediction is that by 2025, one quarter of the world's population will face severe watershortage [62]. To avoid social and environmental chaos, there is a clear need for better management of the limited amount of water available. Because agriculture is the main consumer of fresh water, increasing irrigation efficiencies seems to be the practical way to save water.

While agriculture is the main consumer of water, the application efficiency of this vital resource is low. Only 40 to 60% of the applied water is effectively used by the crop, the remainder of the water is lost either through evaporation, through runoff or by deep percolation into the ground water. Poor management of irrigation water is the principle reason for the low water use efficiency in irrigation. Most people tend to overirrigate believing that applying more water will increase yields. Instead, overirrigation can cause more damage to plants and cause environmental problems. The excess of soil moisture often results in plant disease, nutrient leaching, ground water pollution and salinization. In addition water, energy and labor are wasted.

The solution is to automate the irrigation water management process to decide when to irrigate the crop and how much water to apply. The automation involves capabilities such as monitoring soil moisture and climate conditions and applies water at an appropriate time based on an informed and intelligent decision. There are many advantages offered by automation irrigation systems. First of all, automation leads to no human intervention. It helps in getting a warning signal if any of the sensors get choked up. Secondly, in automatic operation, the set points for optimum moisture level can be carefully adjusted to ensure good growth of the plant. Third, this solution is the sole means that will not only allow a better water use efficiency but may also provide necessary information that can be used as a historical database to improve irrigation performance.

Research has made considerable advances over the last few decades and a large number of techniques and methodologies (chapter-2) have been made available for proper management of irrigation water use. The weather-based irrigation management models have been refined over the decades, still, they are not widely adopted in production agriculture [28]. One of the problems associated with these methods is the fact that inherent errors -due to measurements mainly- cumulate and propagate over time. When used with microirrigation for example, these errors can be more costly, since this type of irrigation requires soil water status to be maintained within a narrow window.

An alternate, more desirable solution is to use *in-situ* sensing systems. Techniques based on this approach for the irrigation water management do not care about accurately modeling the " ins " and " outs " of water in the effective root zone. Moreover, they are regardless of the crop development stage or if a crop is even present. All they worry about is how much water is in the effective root zone at any one time. When the soil moisture in the root zone reaches the crops allowable depletion level, which is based on the crop water level threshold, irrigation should be applied. A big advantage of field soil moisture testing is that irrigations are applied when they are needed by the crop, potentially saving a non-essential irrigation events. This precise, soil moisture information driven irrigation scheduling method could result in significant water conservation measures, while preventing plant stress.

With the proliferation of powerful and low cost controllers - microprocessors and microcontrollers - and the availability (in the market) of different type of transducers, the soil status conditions along with on-field weather data can be integrated in a real-time control system to continuously " visualize " what goes in the soil profile and provide the necessary treatment. For these reasons a controller was designed and implemented to control the water content at the root zone of the plants. Of course, a standard commercial computer such a PC could have been used. However, the expense and inconvenience of having a PC in-situ, with the necessary power supply was felt to be excessive. A self-contained unit was considered more attractive and can be fed easily by a solar panel.

1.3 Goal and Contribution

The goal of this work is to design and implement a microprocessor controlled -interrupt driven- irrigation system. It is targeted towards users of automated irrigation systems, in order to improve the irrigation water management and frost protection of crops. To achieve this goal, we developed a procedure integrating the agrohydrological aspect of irrigation with the microprocessing aspects. In other words, we focused our work on three areas:

i- agriculture engineering - the plant root zone as a dynamic system and the growth mechanism of the plants -

ii- real-time microprocessor-based data acquisition and distribution,

iii- serial interfacing and communication between the controller and a personal computer

In order for the system to operate in the automatic mode - no external intervention -, it is necessary to use a closed loop controller mechanism. The closed loop controller requires the acquisition of soil, and weather data parameters, such as soil moisture, temperature, wind speed and humidity. The level of the moisture in the root zone is compared against a desired level and a decision based on this comparison is made whether the irrigation should take place or not. It is known that it takes some time for water to infiltrate through the soil, hence, using a simple 'on - off' controller would result in a waste of water past the root zone. To overcome this time lag due to the water infiltration, we used a technique called 'anticipatory control'. This technique requires at least two soil moisture probes to properly control the water flow front and hence, the irrigation.

The microprocessor-based control system consists of a combination of hardware and firmware that acts as a supervisor used to manage irrigation and other practice such as: frost protection of crops, excessive heat. It may also be used to apply crop nutrients with irrigation water through fertigation. Several programmable peripheral input/output devices are used to interface the different sensors, keyboard and actuators with the microprocessor. To take full advantages of the microprocessor's power and resources, the interrupt-driven mode is used as a means of data transfer.

A serial communication link with a personal computer enables to having an overall picture of the system' s status on the computer screen. Also, the possibility to turn off the irrigation system by a simple click on the mouse.

Figures 1.1-a and 1.1-b depict the general block diagram of the interrupt-driven microprocessor-controlled irrigation system (IDMCIS) for short. The several units are linked together to form an integrated autonomous programmable system, they are:

i- The Microprocessing Unit

It is constructed around an 8-bit Z80 microprocessor running at a speed of 3Mhz provides a computing power which is ample for this application.

ii- The Decoding and Memory Unit

It includes a RAM and an EPROM. The former is used to hold the intermediate results and also as stack, while the latter holds the firmware program of the application itself.

iii- The Keyboard and Data Acquisition Unit

This unit accepts external data from two different sources: a 16 character keypad used to enter data and commands, and an 8-bit multiplexed 8 channels analog to digital converter (ADC) used to interface the different sensors with the microprocessor.

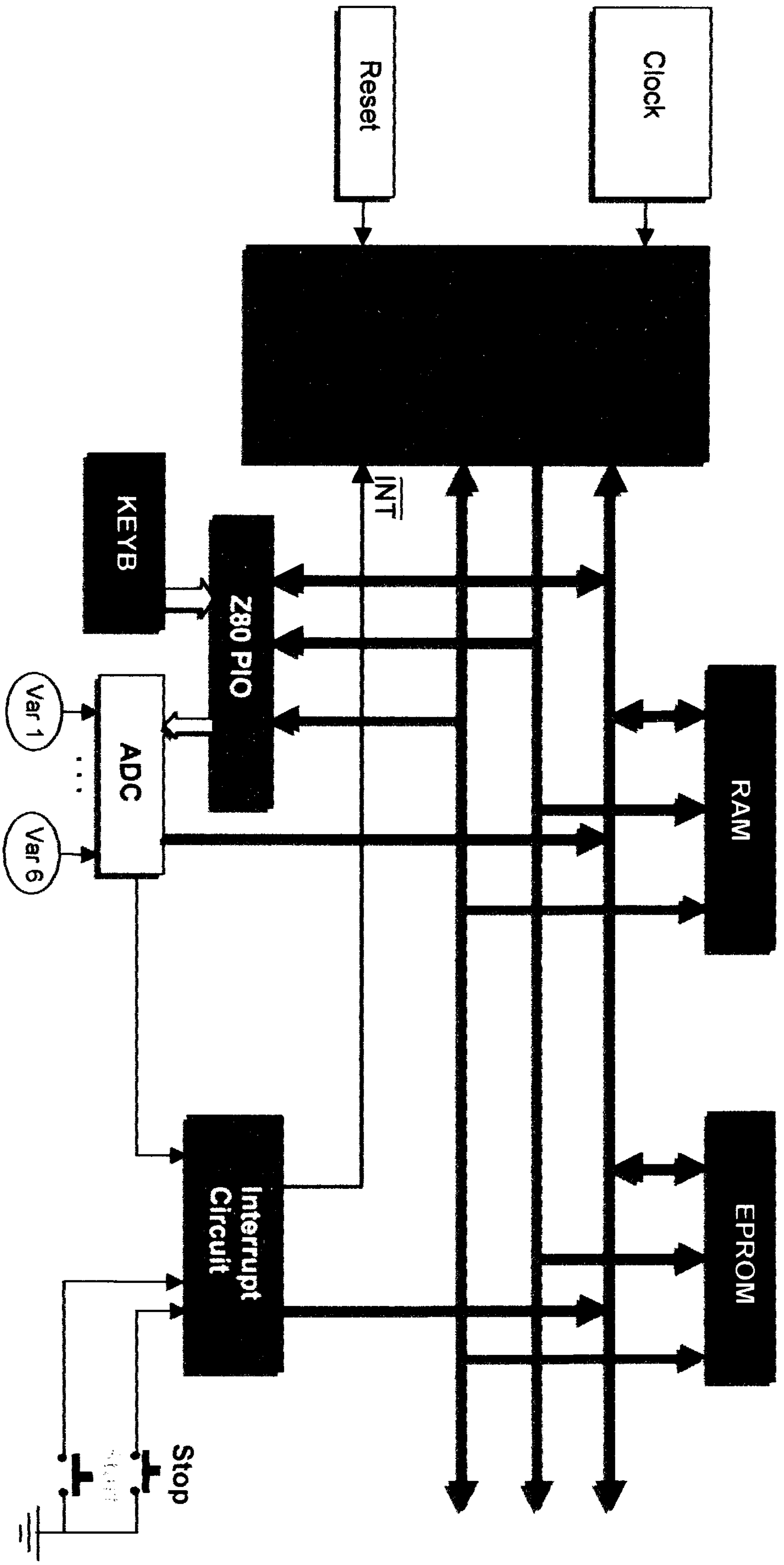


Figure 1.1.a The Interrupt-Driven Microprocessor-Based Irrigation System Block Diagram

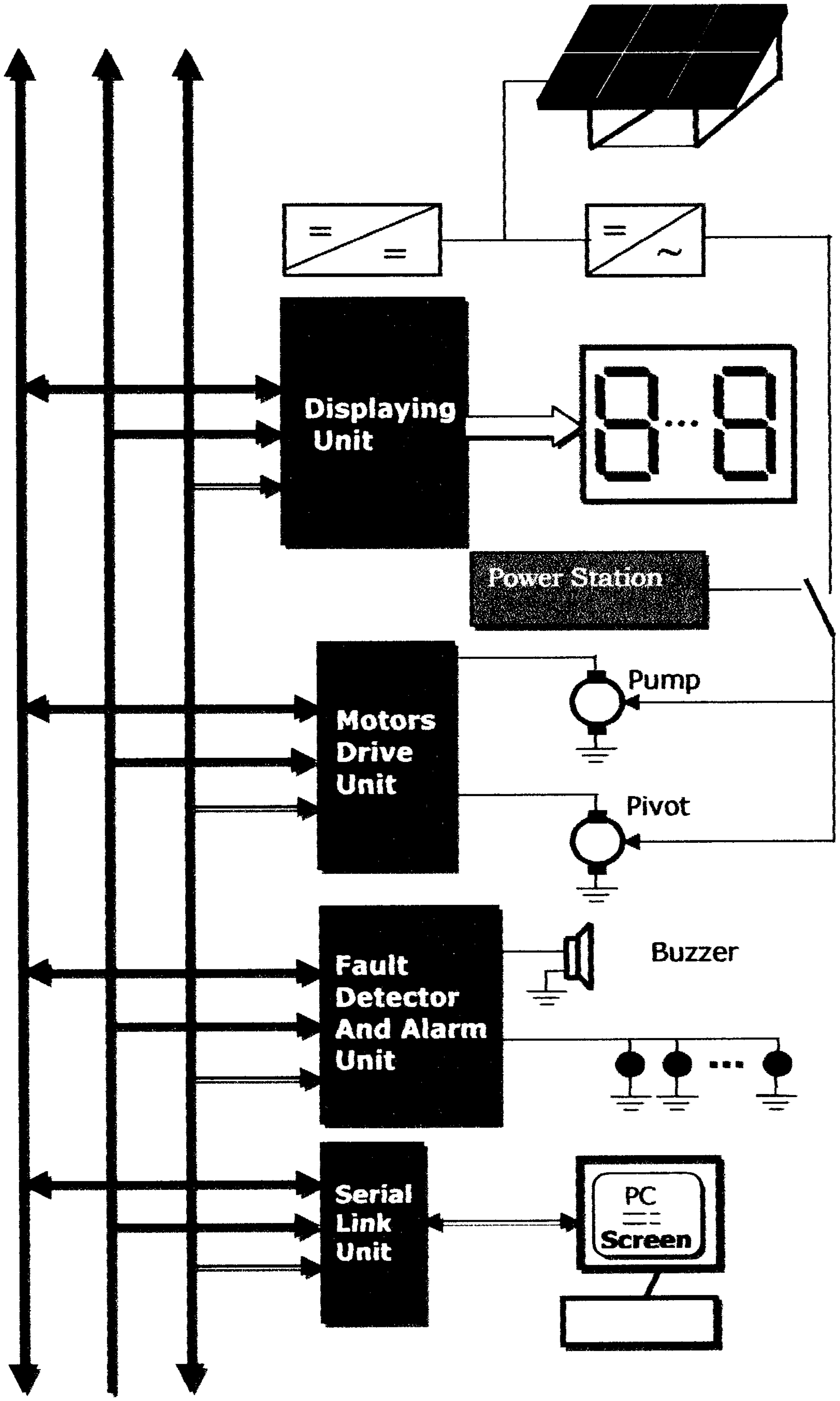


Figure 1.1-b The Interrupt-Driven Microprocessor-Controlled Irrigation System Block Diagram.

iv- The Interrupting Unit

The controller uses extensively the interrupt driven technique. This unit is provided to handle the different sources of interrupts. In manual mode, a simple depresses on START push-button starts the irrigation process regardless of the set points, soil and weather parameters.

v- The Display Unit

This unit is made up of several 7-segment LED displays and its associated decoder driver chips. It provides the digital readout for the system.

vi- The Motors Driving Unit

This unit interfaces the microprocessor and the low power DC motors used to in the prototype to simulate the pump and the pivot motor.

vii- The Fault Detection and Alarm Unit

This unit is used to ensure a minimal amount of safety when using the irrigation system. Upon a failure in a given sensor, a sound alarm is triggered and the corresponding faulty sensor identified.

viii- The Serial Digital Communication Unit

Through the full-duplex serial digital communication, the controller is responsible for sending the data of the sensors and the status of the actuators to the host personal computer (P.C), and receives a command to turn the irrigation system OFF in case of an emergency. The status of the irrigation system is periodically transmitted to the P.C. A graphical user interface (GUI) is developed to allow the user to have an overall picture of the system operation as well as taking some action by a simple click on the mouse.

1.4 Organization of the Report

Having provided the motivation for our work, we proceed to outline the contents of this report. In chapter 2, we discuss the state of knowledge related to irrigation scheduling. In chapter 3, we present a background theory describing soil and its types; the interaction between soil, plant, water and atmospheric parameters. We also discuss the evaluation of surface irrigation and pressurized irrigation systems. In chapter 4, we describe the design approach used for the implementation of our system. We discuss the data transfer technique used, the interrupt mechanism and the why of using more than one probe to monitor soil moisture.

In chapter 5, which constitutes the bulk of the report, each box in the block diagram is turned into actual functions and integrated circuits. The schematics are included and fully described. This chapter ends up with flowcharts summarizing the overall system operation and the window application at run time.

In Chapter-6, we summarize the work presented in this report and discuss several possible directions for continued research into the development of intelligent irrigation scheduling systems for real-time applications.

Finally, the report terminates with an extensive list of references provided for additional information on the subject.

CHAPTER 2

State of Knowledge

This chapter presents previous works in the area that is related to the contribution presented in this report.

2.1 Overview

A reliable automated, real time irrigation scheduling and control system would have obvious advantages that include lower labor costs and lower plant stress levels. It could also include lower water use or higher water use efficiency. The exact time and amount of irrigation are the two deterministic factors for the efficient irrigation management.

Irrigation scheduling is a generic term applied to practice that is intended to aid the farmer in determining when and how much to irrigate. Irrigation scheduling was defined by Jensen [37] as *a planning and decision-making activity that the farm manager or operator of an irrigated farm is involved in before and during most of the growing season for each crop that is grown.* This basic definition remains the typical view of irrigation scheduling today. The level of sophistication for decision making ranges from personal experience, to following neighbors practices to techniques based on microprocessors and expensive computer-aided decision tools that assess crop water needs.

Many methods exist currently to obtaining the information needed to schedule irrigations. Each method is based on an irrigation criterion -- an indicator-- that is used to trigger irrigation, and an irrigation strategy that determines when and how much water to apply. The several approaches are based on soil water measurement, plant stress indicators and soil water balance estimates. The first two are generally site specific, while the atmospheric approach is not crop specific but can access irrigation needs over large areas. Some of the methods will enable "transfer to farmer", while other will be considered as research tool. These methods can range from very simple calendars to intermediate and high technology methods [7], [13], [27]. Current irrigation scheduling methods include:

1- Intuition and Experience
2- Feel and Appearance method
3- Estimates of evapotranspiration and water balance method
4- Plant stress water status indicators and methods
5- Soil water measurement

Intuition and experience method is definitely the least costly and the least accurate. Farmers use their own experience and indicators - wilting characteristics, soil dryness, fixed dates, etc. - to determine when to irrigate.

Feel and appearance is one of the oldest and simplest method. It consists in checking subsurface soil moisture visually and by feel. It is easy, quick and low cost. However, it is not precise and it takes time for farmers to gain the experience to interpret the results.

These two first approaches have proved to be not very accurate. Scientific approaches to schedule irrigation based on the three major approaches: using the soil, the plants and the atmosphere are discussed in the following sections

2.2 The ET Estimations.

Water is absorbed from the soil - effective root zone - via the roots, transported through the plant, then lost to the atmosphere via plant leaves, by transpiration process. Evaporation from the soil surface combined with plant transpiration is referred to as evapotranspiration (ET), also known crop water use. It is a way of estimating how much water has been taken from the soil, and therefore how much irrigation is required. ET is usually expressed in mm per day. Weather and crop development affect daily crop water use throughout the growing season.

Calculating the ET of a crop for an irrigation schedule usually begins with the acquisition of the reference evapotranspiration value noted ETo. ETo represents the rate of evapotranspiration from an extensive surface of a green grass cover, of uniform height of 0.12m, actively growing, completely shading the ground and not short of water [14].

The evapotranspiration of the crop cover being managed noted ETc is related to ETo by a correcting factor Kc, known as the crop coefficient. The mathematical relationship is defined as

$$ETc = ETo \bullet Kc \qquad (1)$$

where

ETc : evapotranspiration of the crop

ETo : evapotranspiration reference

Kc : crop coefficient.

Reference ETo values can be obtained from several sources.

1- In some countries, ETo information is available through radio and newspaper reports

2- It can be measured experimentally using

- Evaporation pans or Evapometers
- ET gauges or Atmometres
- Lysimeters

3- It can theoretically be calculated using one of the numerous methods. Few of them are listed below:

- Penman equation [3].
- Penman-Monteith formula [3], [50].
- Blaney-Criddle method [3].
- Priestly-Taylor [59].
- Hargreaves [24].

ETo can be computed from meteorological data. This gave rise to a large number of more or less empirical methods that have been developed over the last 50 years by numerous scientists and researchers.

The diversity of measurement methods and computations methods revealed the need for formulating a standard method for the computation of ETo. As a result, a consultation of experts and researchers was organized by the Food and Agriculture Organization of the United Nations (FAO) on may 1990 to review the FAO methodologies on crop water requirements. The panel of experts recommended the adoption of the Penman-Monteith method as the sole standard method for the definition and computation of the ETo. The FAO Penman-Monteith method was developed. The FAO Penman-Monteith method requires solar radiation, air temperature, air humidity and wind speed [3]. Other researchers ([3], [25], [31], [33], [68], recommended also the Penman-Monteith method for predicting ETo.

2.2.1 Pan evaporation method

ETo can also be estimated from pan evaporation. Pans provide a measurement of the combined effect of temperature, humidity, wind speed and solar radiation on the reference crop evapotranspiration. The best known pans are class 'A' (a 120.7cm diameter, 25cm depth circular pan) and the Sunken Colorado (a 92cm side, 41cm depth square pan).

When using the evaporation pan to estimate the ETo, in fact it is a comparison that is made between the evaporation from the water surface in the pan and the evapotranspiration of the standard grass. Of course, the water in the pan and the grass do not react in exactly the same way to the climate. A plant is a very complex living organism. It possesses mechanisms for limiting transpiration rates during water stress periods, whereas the pan does not. Therefore, a dimensionless correcting factor is used, it is the pan coefficient KPAN. ETo may be expressed as:

$$ET_0 = K_{PAN} \bullet E_{PAN} \qquad (2)$$

To get an appropriate ETo value, one may use the following 'rule-of-thumb' conversion: multiply the EPAN value by 0.7 in winter or cooler times of the year or by 0.6 during summer or warmer periods. The supplier of the pans usually provides details of the pan coefficient.

Evapometers tend to over estimate ETo leading in over-irrigations. Hence, they are not recommended for anything other than a rough guide to relative rates of ETo.

2.2.2 ET gauges or Atmometers

An atmometer measures evaporation from a wet, porous surface commonly a ceramic disk. Atmometers have been used since 1900's to study plant transpiration [26]. The 'modified atmometer' in which the ceramic plate is covered by green canvas simulates more accurately the transpiration from a plant. It has been compared with Penman ET [4], [8] and a close agreement has been observed.

2.2.3 Lysimeters

In order to determine with great accuracy the different terms in the soil water balance equation, the ET can be measured using a weighing lysimeter. This instrument is used to measure the loss of water from crop plots. Typically, soil is placed in a box that is supported beneath by a weighing scale. Crop is planted on the soil surface, well irrigated and properly maintained. The water loss is directly measured by the change of mass and evapotranspiration can be obtained with a high accuracy.

As lysimeters are too complex and expensive to construct and as their operation and maintenance require special care, their use is limited to research purposes [3].

2.2.4 The water balance equation

One means of determining when irrigation should be supplied is through the use of a soil water balance, or soil water budget method. This involves keeping an 'account' of water going into and out of the root zone. The primary sources of water going into the effective root zone are rainfall and irrigation. Each day however, the crop extracts water from the available water in the root zone. The amount of water taken from the root zone is determined by ETc. The other losses of water are deep percolation past the root zone from excess irrigation or rainfall and runoff.

The total water in the effective root zone on a particular day can be represented by the water balance formula as follows:

$$SWC_t = SWC_{t-1} + IRR + RAIN - ETc - DEEP - Runoff \qquad (3)$$

where

SWC_t = Available soil water content in the root zone today
SWC_{t-1} = Available soil water content in the root zone yesterday
IRR = Irrigation depth since yesterday
RAIN = Rain depth since yesterday
ETc = Crop evapotranspiration
DEEP = Percolation below the root zone
Runoff = Runoff in.

Eq-3 keeps track of water coming into and out of the effective root zone. The determination of the water budget parameters in time and space as accurately as possible is of great importance to reduce potential uncertainties in simulated outputs [1], [2], [15]. Although the water budget is simple in concept, it is difficult to assess quantitatively. Rain is measured using rain gauges or obtained from a local weather station. Irrigation depth can be calculated from the application rate of the irrigation system and the duration application. Estimating the crop evapotranspiration ETc, the most critical parameter, may however be a complex task.

Eq-3 is used to predict the next irrigation. When there is no irrigation and no rainfall, it reduces to:

$$SWC_t = SWC_{t-1} - ETc \qquad (4)$$

The water budget method provides an estimate of the soil moisture at the end of a day. Any irrigation should be applied when the water level at the end of the day, that is, SWC_t is less than or equal to the minimum allowable soil water content or if this level is less than or equal to the refill point. The amount of irrigation that should be applied is usually that amount required to take the soil moisture level back to field capacity. Hence, in order to determine SWC_t, and predict when the next irrigation should take place, one needs to estimate the crop water use ETc.

ETc is affected by both environmental and biological factors. Important environmental factors include solar radiation, temperature, humidity, wind velocity and soil moisture. Biological factors affecting ETc include type of vegetation, and stage of growth.

The values of Kc varies with crop, development stage of the crop and to some extend with wind speed and humidity. For most crops, the Kc value increases from a low value at time of crop emergency to a maximum value during the period when the crop reaches full development. Kc is usually less than one, indicating that the crop usually uses less than the maximum amount of water by ET_0. Crop coefficients are measured empirically through the experimental measurement of evapotranspiration A list of crop coefficients of numerous crops can be found in [49].

A large number of software packages used to help grower schedule their irrigation was developed. These are based mainly on soil water budget equation, in which ET is computed using different models [25], [27], [29], [58], [65].

2.3 Plant Stress Water Criteria and Methods

Plants respond interactively to both soil and environmental conditions. Hence, instead of measuring or estimating, by means of a water balance, the amount of water in the rooted soil, it is possible to get messages from the plant itself. This latter is the link between the soil and the atmosphere, and its water status can give an indication of when to irrigate. Many research works have been carried out based on the bio-information of plant to optimize irrigation management systems. The most widely used plant water stress criteria are:

- Branch and stem diameter change
- Leaf water potential
- Sap flow

- Canopy temperature

Some of the methods using the above criteria are destructive, that is, they require a sample of the plant to be removed for testing. Other methods involve fixing sensors to plants.

It is widely accepted that the microvariation of stem diameter reflects the water status of plant and could be utilized as a criterion for timing irrigation [32], [39], [40], [41], [42], [51]. Lee and Shin [43] developed a prototype system for optimal irrigation control of tomato crop based on stem diameter and transpiration of the plant. The controller collects the data from the stem diameter and weather sensors, calculates the irrigation time and the water quantity and controls the actuators such as pumping motors, solenoid valves, etc.

Sap flow measurements of transpiration remain a promising method for automating irrigation control based on direct physical plant measurement [74]. Plant canopy temperature has been recognized as a sensitive indicator of plant water status, - inadequate water supply increases canopy temperature - and has lead to the development of stress-related indices based on the difference between plant canopy temperature and ambient air temperature [9], [18]. Evett et al [17] tested a system that uses crop canopy temperature to automatically schedule and apply irrigations. The system evaluates crop canopy temperature at every minute of the day, and makes irrigation decisions every night based on the number of minutes in the day that the canopy temperature was above a threshold value. Wanjura et al [76] described a system for using a threshold canopy temperature based on the thermal kinetic window, as an index for automatic control of irrigation on a near real-time basis.

Most of these methods are not available commercially, and tend to be used more for scientific experiments. These methods give the water status of the plants and therefore, give a guide to as when one should irrigate. They present, however, one major limitation that when plant stress is observed some yield reduction may have already occurred.

2.4 Soil Water Status

Automated control of irrigation requires the use of soil, crop, or environmental sensors or their combination to determine the need for irrigation and then either a logic-type controller or a computer to control irrigation sequence [36], [57], [63], [80]. The controller may need to use various control modules to properly manage the irrigation system.

Measuring soil moisture is a well-recognized method of estimating crop water use and is highly recommended. The soil moisture - content and tension or potential – levels can be measured at points in fields. Electronic sensors can be utilized with automated data acquisition methods and integrated into irrigation control to determine how much water to apply, and with some knowledge of stress point for a crop, when to apply that water.

Soil water measurement methods have been widely reviewed [11], [52], [54], [56], [65]. Electrical resistance/capacitance, neutron probes, tensiometers and Granular matrix type sensors dominate the irrigation management. The formers have remained one of the least expensive soil water measurement technologies applicable to producer use [16], [69]. These and other types of soil water sensors have been used successfully in automated control systems.

Testezlaf et al [67] developed an automated irrigation computer control system for management of greenhouse container plants. The system consisted of soil moisture sensors, a hardware I/O interface, a computer with software and actuators. Thompson et al [70] were reported on the use of microprocessor control units for center pivot irrigation. Progue [58] describes how the Watermark soil moisture sensors were used to override landscape irrigation controllers. Leib et al [44] used a microprocessor and pressure switch to monitor the timing and amount of irrigation water applied for individual fields. Phene [53] automated the evaporation pan system, by monitoring it hourly using an electronic water level sensor connected to a data acquisition system. Phene et al [55] used electronic measurement of evaporation pan losses to control a subsurface drip system with a microprocessor. McCann et al [47] documented how microprocessors are being used to move control from the irrigation system level to individual sprinkler level to provide a variable rate water application on continuous move sprinkler systems.

Lieth [45] operated a computer-controlled irrigation system for potted plants. He reported that high-quality, potted crops were produced, and there was a significant decrease in water use.

Soil water measurements are useful for verifying ET models and for triggering or halting irrigation. They are also useful for feedback information on irrigation scheduling based on ET [35]. Since practically all models are unreliable in predicting many water balance components, it is desirable that measurements of soil water be periodically acquired to verify and adjust model output for irrigation application and rainfall infiltration differences [31], [72], [79]. The controller must activate the irrigation based on the current soil water status, so this activation must permit irrigation in time to avoid crop stress.

CHAPTER 3

Soil-Water-Plant Interaction And Irrigation Systems

All field crops need soil, water, light (sunshine) and air to grow. The soil gives stability to the plants; it also stores the water and nutrients that the plants can take up through their roots. Without water, crops cannot grow. Water is essential in the plant environment for a number of reasons:

i- it transports minerals through the soil to the roots
ii- it acts as a solvent for dissolved sugars and minerals
iii- it provides the cooling mechanism that allows plants to maintain the favorable temperature necessary for metabolic processes.

The sunlight provides the energy that is necessary for plant growth. The air allows the plants to "breath".

This chapter focuses on the soil's physical and chemical properties that determine its ability to transport as well as store water and nutrients. It provides information on water movement through different soils and water uptake by plant roots. Although one does not need to know a great deal about soil physics or plant physiology for proper irrigation, however having a general knowledge of these resources interactions can be very helpful when making decisions for their effective management for crop production and helps demystify irrigation scheduling.

3.1 Soil Characteristics

Soil is a habitat for plants. It is a complex system made of solid particles (living and nonliving materials), water, and pore spaces. The solid particles are of two types. One part is originated from the degradation of rocks, they are the mineral particles. The other is made up of decaying plants and animals (rotting leaves, pieces of bones, etc.), they are called organic particles. Water and air are the other ingredients in soil.

The combination of mineral and organic matter is referred to as the solids. The soil particles seem to touch each other, but in reality have space in between. The pore space is the voids between soil particles and is occupied by either air or water. The soil's texture and structure determine the quantity and size of the pore spaces.

When the soil is dry, the pores are mainly filled with air. After irrigation or rainfall, the pores are filled with water. Figure 3.1 illustrates a schematic representation of soil as a dynamic system composed of air, water and solids [64].

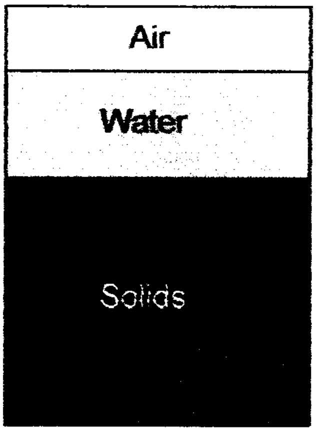


Figure 3.1 Schematic representation of soil as a dynamic system composed of air, water and solids.

3.1.1 Soil Texture

The mineral particles of the soil differ widely in size and can be categorized as in table-3.1 [61].

Name of particles	Size in mm
Sand	0.5 to 1.0
Silt	0.002 to 0.5
Clay	Less than 0.002

Table-3.1 Classification of soil particles

The relative proportion of sand, silt and clay present in the soil determines the soil texture. In coarse textured soils, sand is predominant. These soils are referred to as sandy soils. In medium textured soils, silt is predominant. These soils are called loamy soils. Clayey soils are fine textured soils where clay particles are predominant.

Scientists have established a list of twelve major soil textures [77]. These textures are classified according to the relative proportion of sand, silt and clay found in a given soil.

3.1.2 Soil Structure

Soil structure reflects how the individual soil particles (sand, silt, clay, organic matter and fertilizers) clump together into porous compounds, called aggregates. In other words, soil structure is the shape and arrangement of soil particles into aggregates that are separated by pores. The soil structure is very important since (along with the soil texture) it affects the pore space of the soil. A dense structure will generally reduce the amount of air and water that can move freely through the soil. Also, it will affect the root penetration or the plant' s ability to propagate roots through the soil. These spaces are also where tiny things such as microorganisms and insects are found [77]. Hence, soil's porosity is one of the most important factors that have an effect on the quantity of water and air a soil can hold.

Unlike texture that cannot be changed or destroyed, the structure of a soil can be easily changed. Natural and man-made physical and chemical factors affect soil structure over time. Soil compaction, tillage operations, traffic, etc. reduce porosity by forcing particles closer together. It is possible, however, to maintain good soil structure by practicing beneficial soil management such as: crop rotations, organic matter additions, ridging, etc. These cultivation practices can be applied by a farmer to develop large pores that are essential for rapid movement of water and air through soils.

A sandy, coarse-textured soil drains easily and quickly after a significant rainfall of a heavy irrigation. It presents, however, some disadvantages. It has lower moisture and nutrient holding capacities than a clay soil. A clay soil although its capability of holding high moisture volume, it has some disadvantages. It has poor drainage. After a significant rainfall, the soil remains saturated with water, plant roots will be deprived of oxygen that results in a reduced health and vigor of the plants.

A "good" soil needs to be loose enough that the plant roots can spread easily, absorbent enough that it will collect water and rich enough that it can provide the plant with nutrients.

3.2 Soil Moisture Characteristics

Soil in the plant root zone acts as a reservoir for water. How well it stores this water is of great importance to crop production and the vitality of the land. The soil' s ability to hold water depends on both the soil texture and structure. Water is held in the soil in pores of various sizes. The finer the soil texture (such as clays or loams) with small pores the more water soil can hold.

As the soil goes from wet to dry, the relationship of the water to the soil changes, as well as the type of movement and amount of storage. After a significant rainfall shower or an irrigation application, the soil would have its pore space filled with water. If all soil pores are filled with water, the soil is said to be saturated.

At saturation, no air is present and the plant will suffer. Many crops cannot withstand saturated conditions for long periods [77]. The soil pore spaces would start to drain the water present in the larger pores will move downward, all of the water movement would be gravitational. The water drained from the pores is replaced by air. After a certain time, the gravitational movement and drainage would slow to a very low rate. At this point the soil particles would be holding the water with adhesive forces, thus keeping the pore spaces from draining any further. At this point, the soil is said to be at field capacity or F.C for short. At F.C, plants are easily able to draw water from the soil. This is the ideal situation for crop growth.

As the soil dries out, the rate of evaporation from the surface falls fairly quickly until the soil surface is dry. Plants for their transpiration and growth use the remaining soil moisture, which is the larger part of the available water. Because the plants consume water, then the plant roots take little by little the water stored in the soil. After some time, active growth stops as the plant finds it harder to extract the remaining soil moisture. The soil moisture level at this stage is known as the stress point or refill point. Theoretically, the most efficient irrigation is achieved when soils reach the stress point and enough water is applied to return the soil moisture to F.C or to a level juste below leaving a room for an impredicted rainfall. If the crop is not irrigated or water is not received from other sources, the soil will eventually dry out to the point where it becomes harder for the plants to suck the water from the soil. At this stage, the uptake of water is not sufficient to meet the plant' s needs. The plants loose freshness and wilt, the leaves change color from green to yellow. Finally, the plants die. The soil water content at the stage where the plants die is called permanent wilting point (P.W.P).

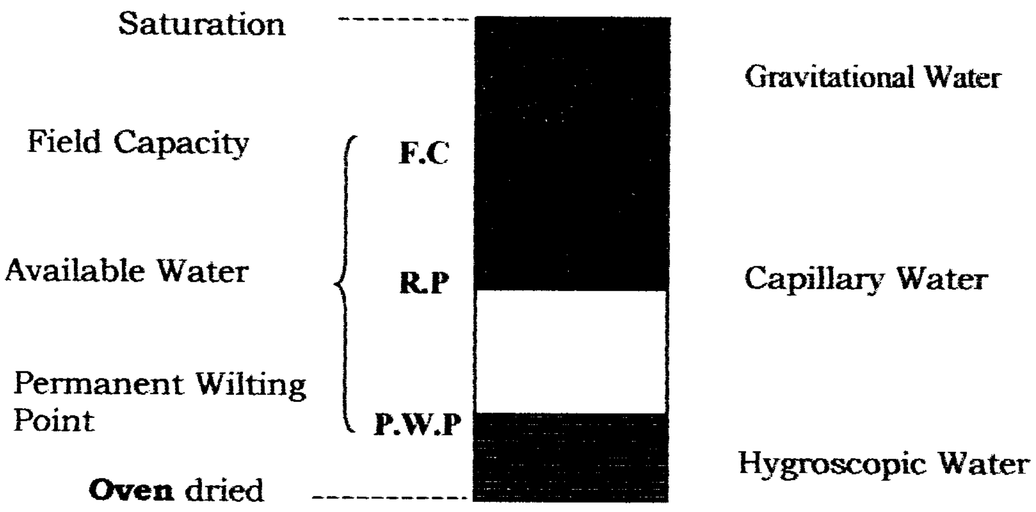


Figure- 3.2 Soil Water Content

At the P.W.P, there is still water in the soil, but it is held so tightly by the soil that the plant cannot use it. This remaining moisture is called hygroscopic water, which can be removed only by applying heat (oven dried) to force drying. Figure 3-2 pictures the possible different soil water levels in the soil. Terms F.C and P.W.P are conceptual, as the actual soil moisture values will vary in each soil due to differences in soil texture and structure.

The moisture between field capacity and permanent wilting point is called available water, (A.W) for short. This amount of moisture can be divided into two portions. The first, the readily available water, (R.A.W) for short, a quantity of water that is relatively easy for a plant to use (it is common to consider about 50% of the available water as readily available water). It is the amount of soil water between F.C and the refill point (R.P). This is the level when soil moisture has declined to a point where the plant becomes stressed. The second part is the little reserved available water. Even though, the plant can use all the water in this portion, the closer the soil is to the wilting point, the harder is for the plant to use the water [48]. In summary, irrigation is about keeping soil moisture between the stress point and F.C, so that growth is not limited.

3.3 Soil Moisture Tension

The degree to which water clings to the soil is the most important soil water characteristic to a growing plant. This concept is often expressed as soil moisture tension. Soil moisture tension is a pressure and commonly expressed in units of bars. Thus, as soil moisture tension increases, the amount of energy exerted by a plant to remove the water from the soil must also increase [48].

A soil that is saturated has a soil moisture tension of about 0.001 bars, or less, which requires little energy for a plant to pull water away from the soil. At field capacity most soils have a soil moisture tension between 0.05 and 0.33 bars. At field capacity it is relatively easy for a plant to remove water from the soil.

The wilting point is reached when the maximum energy exerted by a plant is equal to the tension with which the soil holds the water. For most crops, this soil moisture tension is about 15 bars.

3.4 How Plant Works

In order for plants to grow and be healthy they need six things: light, nutrients, water, the right temperature, space to grow and air. Energy is provided by sunlight and used by the plant to drive the chemical process of photosynthesis, creating plant material by converting carbon dioxide from the air together with chemicals and water into sugar glucose and oxygen. These are the plant's food. Energy is also used by the plant to power the pumping action of lifting water and nutrients in solution from the soil.

Roots, which are an integral part of the plant's body, hold the plant in the ground. They expel carbon dioxide into the soil and absorb oxygen, which means air and porosity are a crucial part of the plant's environment. Most plants will stop growing and eventually die if air is excluded from the roots by waterlogging or compaction. Roots also take minerals and water from the soil and send them through the stem to all parts of the plant [6].

Plants use water as a carrier. Only a fraction of the water pumped by the roots from the soil is actually used within the plant for internal growth. The reservoir of water at root zone is used as a medium to extract nutrients from the soil. Nutrients dissolved in the water are sucked into the plant, the nutrients are extracted and excess water transpired. Hence, most of the water that enters the plant roots does not stay in the plant. Less than 1% of the water withdrawn by the plant is actually used in photosynthesis. The rest of the water moves to the leaf surfaces where it transpires to the atmosphere through tiny openings in their leaves called stomata. It is also through these opening that the plant absorbs the necessary carbon dioxide from the atmosphere.

As water moves from the soil, into the roots, through the stem, into the leaves and through the leaf stomata to the air, it moves from a low water tension to a high water tension. The water tension in the air is related to its relative humidity and is always greater than the water tension in the soil.

If the nutrient solution in the plant root zone is very weak (the soil is too wet or poor) then the plant will cycle large quantities of water for little result, this is very wasteful of water. On the other hand, if the nutrient solution in the soil is too strong (the soil is very dry), the plants cannot extract the water from the soil and will be killed by osmotic pressure sucking water out of the plant.

We have discussed the importance of water to vegetation, and the inference might seem to be that, the better the supply of water, the better the prospects for plant life [28]. This is true, but only up to a certain point. Other factors, intervene, oxygen supply is one such factor. Plants need access to oxygen in the soil for root respiration. If there is no oxygen available, a plant will suffocate. If there is too much water in the soil (saturation), this will limit the oxygen supply, and plant will suffer.

3.5 Irrigation Application Systems

An adequate water supply is important for plant growth. When rainfall is not sufficient, the plants should receive additional water from irrigation. There are many methods that can be used to apply irrigation water to the field. Each method has its advantages and disadvantages.

Probably the simplest method consists of bringing water from the source of supply such as a well, to each plant with a bucket or water can. This is very time consuming method and involves very heavy work. However, it can be used successfully to irrigate very small plots of land, such as vegetable gardens, that are close to the water source. For large irrigation fields, sophisticated methods of water application are to be used. Such irrigation application systems consist of an intake structure or pumping station, a conveyance system, a distribution system, a field application, and a drainage system. The pumping structure, directs water from the source of supply (reservoir, well, river, etc.) into the irrigation system. The conveyance and distribution systems consist of canals, or pipelines transporting the water through the whole irrigation system. The field application system assumes the transport of water within the field. The drainage system is provided to remove the excess water, due to a heavy rainfall or an over-irrigation, from the field.

There are three commonly used methods for water distribution: surface irrigation, pressurized irrigation and drip irrigation. Figure 3.3 illustrates the various field irrigation techniques.

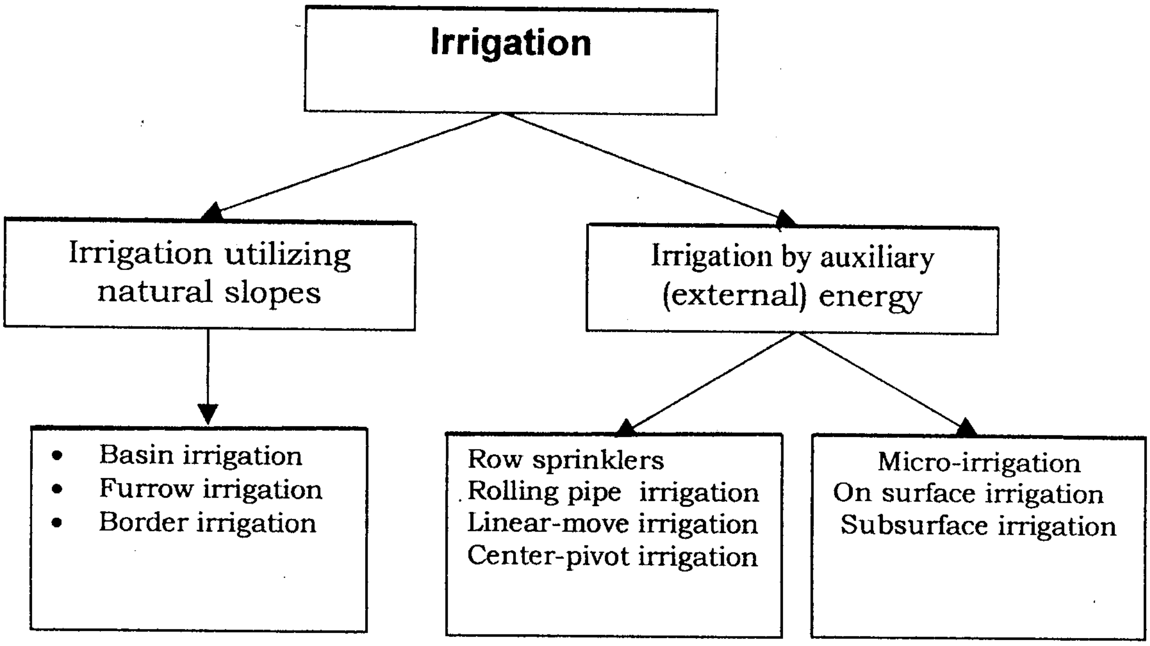


Figure 3.3 Classification of irrigation methods

3.5.1 Surface Irrigation

Surface irrigation also called gravity-flow irrigation method is both the most ancient and the most widespread in the world (about 80% of the irrigated area) [5]. In this method the application of the irrigation water to the field is done at the ground level, that is, the irrigation water is conveyed and distributed at the field level by a free surface, overland flow.

Surface irrigation systems rely on gravity flow to distribute water across the field. Surface systems, while are the least expensive to install, have high labor requirements for operation. They are best suited to medium and fine textured soil with higher moisture-holding capacities. Also, for a better application efficiency of water, field slope should be minimal and fairly uniform to permit controlled advance of water.

There are several surface irrigation types. Either the entire field is flooded (basin irrigation) or the water is fed into small channels (furrow irrigation) or strips of land (border irrigation).

3.5.1.1 Level Basin or (dead level) Irrigation

Basins are horizontal, flat plots of land surrounded by low bunds. The bunds prevent the water from flowing to the adjacent fields [20]. In this method water is led directly from the field channel into the basin through siphon, spiles or bundbreaks. Water is applied over a short period of time to a completely level area enclosed by the bunds. This method is commonly used for rice, and crops that are unaffected by standing in water for long periods.

Basin irrigation is suited to moderate to slow intake soils and deep-rooted crops. To achieve a high uniformity and efficiencies, precision land leveling is very important.

3.5.1.2 Furrow Irrigation

Among the techniques of surface irrigation, furrow irrigation is the most widespread. Furrows are narrow, evenly spaced, shallow channels formed in the soil. They are used to carry water down the land slope between the crop rows. As water runs along the furrows down the slope, it infiltrates into the soil. The crop is usually grown on the ridges or beds between the furrows. This method is suitable for all row crops and for crops that cannot stand in water for long periods. Furrow irrigation can be used on most soil types, however, as with all surface irrigation methods, very coarse sands are not recommended as percolation losses can be high [21]. Its major limitations however, are the inability to apply small amount of water at frequent intervals as needed by shallow rooted vegetable crops.

3.5.1.3 Border Irrigation

In this type of irrigation, the field to be irrigated is divided into strips or borders separated by parallel bunds. Water flow down slope as a sheet guided by the bunds on either side.

This type of irrigation uses by far more water than all other types of irrigations combined.

3.5.2 Pressurized Flow Systems

The decline in gravity-flow acreage has been accomplished by an increase in acreage under pressurized systems. Pressurized systems use pressure to distribute water. Pressurized irrigation or (sprinkler irrigation) is a method of applying irrigation water which is similar to natural rainfall. The irrigation water is led to the field through a system of pipes in which the water is under pressure. It is then sprayed into the air using rotating sprinkler heads or spray nozzles so that it breaks up into small drops that fall to ground [60].

Sprinkler systems are better adapted to operate on moderate sloping and irregular terrains that are unsuited to surface irrigation systems. Since their development in the 1960's, they greatly contributed to the expansion of the acreage suitable for irrigation. They are also suited to coarser soils with higher water infiltration.

There are many types of sprinkler irrigation systems in use throughout the world today [64]. The most common continuous moving systems are center-pivot, linear move, and Low Energy Precision Application (LEPA). In the center-pivot system, sprinklers are placed along a lateral line approximately 400m long, which is pivoted at one end, and move around the field in a circular fashion.

The linear move system is similar, except that it is not pivoted at one end, and moves in a straight line perpendicular to the direction of the lateral pipeline. Both system types are often used to irrigate large areas, typically 50 hectares or more.

The LEPA system uses droptubes extending down from the pipeline to apply water at low pressure below the plant canopy usually a few centimeters above ground. Applying water close to the ground cuts water loss from evaporation and wind and increases application uniformity.

Beside the principal task of moistening the soil, pressurized irrigation, also allow precise delivery of other crop inputs such as fertilizers and pesticides by injection into irrigation water, and to control the peaks of high and low temperature. When plants are coated with an ice cloak during the frost event, sprinkler irrigation can be used to spray crops to warm up the atmosphere resulting in the protection of the crops.

3.5.3 Drip Irrigation

Drip irrigation is the slow and frequent application of small amounts of water. The water is conveyed under pressure through a pipe system to the field, where it drips slowly onto the soil through emitters that are located close to the plant. Only the immediate root zone of each plant is wetted. Therefore, this is by far the most water-efficient method of irrigation, since very little water evaporates or drains off before being used by the plants.

Another variant of drip irrigation is the subsurface drip irrigation. In this method, the tubing and emitters are buried beneath the soil surface rather than laid on the ground. The advantages of this method are less water evaporation, runoff, less labor and less animal damage or human vandalism. Because of the high capital costs of installing a subsurface drip irrigation system, this method may be cost-effective when high value crops are considered.

Compared to conventional irrigation systems, drip irrigation systems need much less water for the same results. This last method does not suffer from the disadvantages of the sprinkler system in which some water is lost by evaporation. Another significant feature of drip irrigation is that the system can be used to deliver agricultural chemicals [10]. Fertilizers and pesticides can be dissolved in water, injected into the irrigation system and distributed directly to the plant's root zone. Moreover, microirrigation systems can be automated and typically require less attention for irrigation purposes. Nonetheless, they may require a significant amount of maintenance to continue operating at maximum uniformity. The emitters in the drip tube can easily clog or plug by particle matter, such as sand or biological organisms.

3.5.4 Surface versus Pressurized Irrigation

While surface irrigation is the most widely used, especially in less developed countries, its average performance is still low. The variability of infiltration is often very important. This induces water runoff and deep percolation beyond the root zone in some parts of the field while others are under-irrigated. Various land treatment and management measures have been developed to reduce water losses under gravity flow systems.

Compared to surface irrigation, pressurized systems, if managed correctly, conserve water and other crop inputs while enhancing crop yields [30]. This is of particular importance in arid regions where precipitation is inadequate to successfully grow most crops, and water is the primary limitation factor.

The reasons for the growing popularity of drip irrigation are several. While it offers improved yield, it requires less water, and reduces the amount of fertilizers and other chemicals to be applied. Drip irrigation can also be used in conditions unsuitable for other irrigation methods (undulating slops, very sandy soils, etc).

3.6 Solar Energy and Agriculture

It is common to use kerosene or diesel to power generators in agricultural operations when farms are far from the grid. While these systems can provide power where needed, there are some significant drawbacks, including:

i. Fuel has to be transported, often over long distances

ii. The noise, fumes and air pollution.

iii. Fuel costs and spills can contaminate the land.

iv. Generators require a significant amount of maintenance, and like all mechanical systems, they break down and need replacement parts.

The harmful effects of air pollution cause acid rain. The burning of fuel contributes to the overall warming of the Earth known as the "Greenhouse Effect".

For many agriculture needs, the alternative is solar energy. Solar energy is unlimited and pollution-free power. Modern ' well-designed, simple-to-maintain solar systems can provide the energy that is needed, where it is needed, and when it is needed. There are thousands of systems that have been tested and proven around the world to be cost effective and reliable. While deserts areas such the Sahara, Australia, Arizona, etc. get more sun than other parts of the world, most areas receive enough sunshine to make solar energy practical.

3.6.1 From Sunshine to Electricity

Photovoltaics (PV) for short, are solar cell that convert sunlight to direct current (DC) electricity. The solar cells in a PV module are made of semiconductor materials. They are constructed by putting a layer of n-type and a layer of p-type semiconductor material together. When the photons in solar radiation strike a solar cell, the kinetic energy of the photons is transferred to the valence level of electrons. The freed electrons and the positive-charged holes attract each other and create positive-negative pairs. The formation of these pairs creates electricity [23]. This electricity can then be used to power a load such as a motor-driven pump, or it can be stored in a battery. A stand-alone PV system electricity generator is depicted in figure 3.4. Solar cells are usually sold in panels that vary in size. Connecting individual cells in series adds the voltages of the individual cells, while connecting cells in parallel adds their current.

The heart of a stand-alone solar system the system is the PV generator. It consists of PV modules, called also an array or (solar panels), interconnected to form a DC power-producing unit. The PV modules produce electricity only when the sun is shining. The array can be fixed or mounted on a tracking structure. This latter follows the sun across the sky. A tracker will add some costs to the power system, but will increase power production by more than 25% as compared to a fixed array. For many applications PV array should be used together with a battery storage system. During peak sunlight hours, the batteries are charged by the solar cells, which produce more power than is required by the load. During the night, the batteries discharge to operate the load.

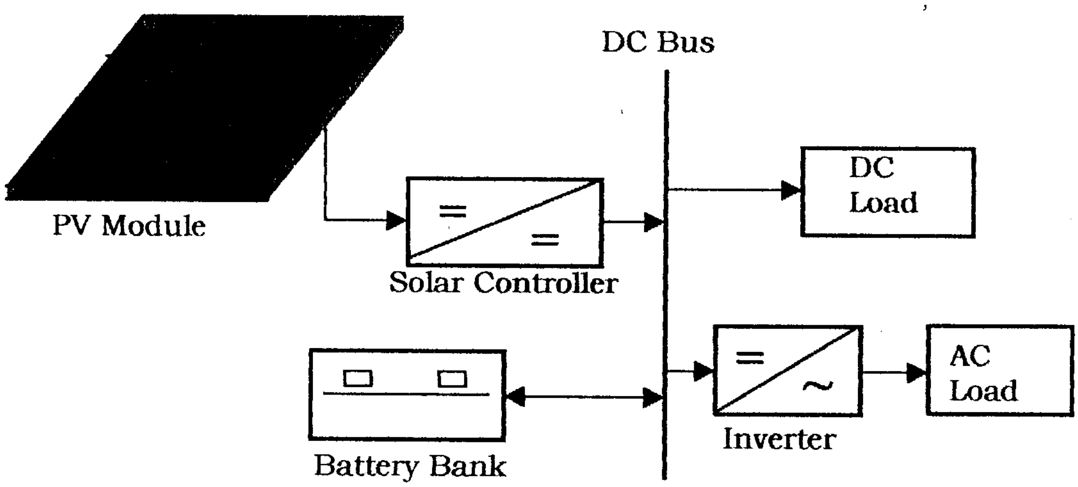


Figure 3.4 A stand-alone PV system generator

PV systems can be very economical in providing electricity at remote locations. They can be much cheaper than installing power lines and step-down transformers in applications such as crop irrigation. Today, PV modules are designed to operate for over 30 years [46]. And, because they require no fuel and have no moving parts, they are more convenient to operate and maintain than diesel or gasoline generators.

3.6.2 Economics and the Future

The cost of PV modules has fallen by 400% in the last 30 years and this trend continues [23]. PV technology also continues to improve the power conversion efficiency of the photovoltaic cell. Increases in P.V cell efficiency decrease the cost of PV power, because fewer modules are required to produce the same amount of power PV systems are very economical in providing electricity at remote locations compared with either extending the electrical grid or using generators.

In the future, when the prices of fossil fuels rise and the economic advantages of mass production reduce the peak watt cost of the PV cell, PV power will become more cost competitive and more common.

CHAPTER 4

Irrigation System Design Approach

When rainfall is insufficient, irrigation water may be supplied to guarantee good harvests. One of the main problems of growers is to know the amount of water that has to be applied to the field and when, that is, how to schedule irrigation. The irrigation scheduling is a process used to maximized the percent of water applied that will be use by the plant, and minimize (or eliminate) any plant water stress by always having sufficient water in the soil.

4.1 The Open/Closed Loop Controllers

A controller is an integral part of an automatic irrigation system. It is an essential tool to control how much water to apply and the right time to apply it to sustain agricultural production and to achieve high level of efficiency in water, energy and chemical uses.

Irrigation controllers have been available for many years and evolved form mechanical and electromechanical irrigation timers to complex computer-based systems. Two general types of controllers are used to control irrigation systems: open loop control and closed loop control systems. In control technology, the term open loop control means a method of control based on a prediction of needs [6]. Sensing and correction is the essence of closed loop control.

In open loop controllers, the basic control parameter is how long is irrigation water is to be applied. Such systems operate on timers. They are inefficient because the water requirements vary dramatically. Open loop controllers invariably apply too much or too little water. While they have the advantage that they are low cost, they do not respond automatically to changing conditions in the environment such as wind, rainfall, and temperature.

A closed control loop or plan-do-monitor-correct procedure is a structured way of continuously improving a process. This structure is illustrated in figure 4.1. The user sets up a general strategy for control. The process creates a cycle by which the consequences of decisions (Do) made and implemented can be checked (monitor) and feature decisions altered (correct) to maintain some set value.

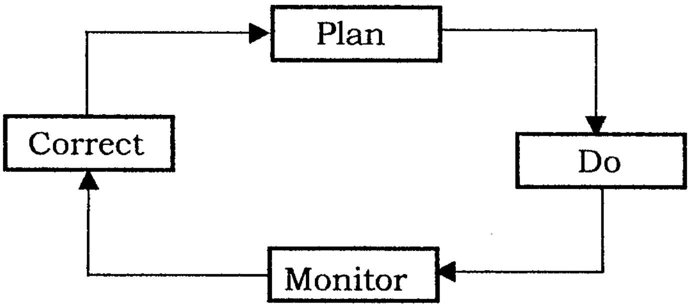


Figure 4.1 A closed control loop process

Closed loop controllers require data acquisitions of environmental parameters such as soil moisture, temperature, wind speed, humidity and rainfall. The irrigation decisions are based on:

i- The direct measurement of the actual soil moisture.

ii- The direct measurement of climatic parameter conditions.

iii- The desired soil moisture.

While the first item allows to “visualize” in real-time the soil moisture content in the root zone, the others determine whether the atmospheric parameter conditions allow a favorable and water efficiency irrigation application. For example, an efficient system should not allow spraying during high wind or during very hot weather. Applying short irrigation cycles with sprinklers in the heat of the day can result in up to 80% of the water being lost by evaporation. Similarly, blowing winds reduce irrigation effectiveness because they increase evaporation and affect sprinkler distribution patterns. The interaction of both wind and temperature affects considerably irrigation.

Closed loop control is vastly superior because if there is any variation between the actual soil moisture content and the set point, the system will adjust to correct the error.

4.2 The Anticipatory Approach Control Mode

In an irrigation system, water losses are due evaporation, runoff and deep percolation beyond the root zone. The runoff can be minimized or completely eliminated by applying water with a rate less than or equal to the soil intake rate. The evaporation losses can be minimized when irrigation is conducted properly interrupting or delaying irrigation on hot windy days should be a common practice.

Minimizing water losses is not enough. A good irrigation scheduling should also maximize irrigation effectiveness. It is known that all irrigation systems have a minimum and maximum irrigation times. The minimum is the time it takes water to start being effective. For example, when the sprinklers start to operate, the water will initially be doing no more than wetting the foliage and the topmost layer of the soil. If the sprinklers were turned off at that point, there would be no effective irrigation because most of the water applied would be lost through evaporation. It is necessary to apply irrigation longer than the minimum time needed for ground to start soaking water. The longer the sprinklers are run the deeper the water will penetrate. Furthermore, even when the sprinklers are shut off, water will continue to soak further into the soil. If the application time is too long, water will percolate beyond the root zone and hence, cannot be used by the plant. For these reasons, a single soil moisture probe in the root zone cannot be used to initiate and stop the irrigation. By the time the water reaches the probe, the irrigation will be stopped, but the soil above the probe being saturated, water will continue to flow past the probe and below the root zone.

A minimum of two probes is required for proper irrigation control. One probe is located in the root bowl of the plant and is set to the desired moisture level. When the moisture level drops below the set point, irrigation will start. A second shallow probe anticipates the water movement by cutting off the irrigation as soon as the flow front reaches the probe and before the water reaches the deeper probe.

The approach used to design the IDMCIS is based on the critical elements listed in section 4.1 and the anticipatory technique described above. The whole process is summarized in the flow graph diagram depicted in figure 4.2.

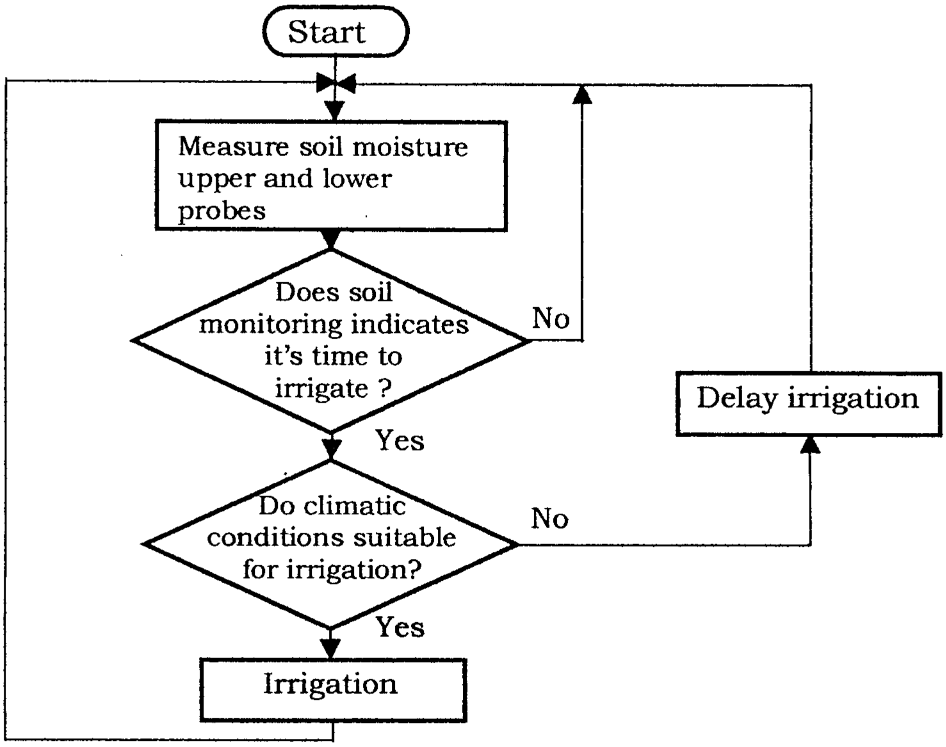


Figure 4.2 Real-time irrigation scheduling flow graph based on soil moisture and climatic

Sections 4.3 and 4.4 describe the theoretical background of data transfer in general and the interrupt technique in particular.

4.3 I/O Mapped I/O

A microprocessor is of no interest by itself. It must be connected to memory and input/output (I/O) devices to carry out computing and application functions. These input and output devices are called either peripherals or I/Os.

The designing of logic circuits (hardware) and the writing of instructions (software) to enable the microprocessor to communicate with the peripheral equipment is called interfacing and the logic circuits are usually labeled as input/output (I/O) ports or interfacing devices [22].

The Z80 microprocessor identifies peripherals either as memory-mapped I/O or I/O-mapped I/O. In the former, a peripheral is identified by a 16-bit address whereas in the latter, by an 8-bit address. In memory-mapped I/O, both I/O and memory devices share the same address space, and any machine instruction that can access memory can be used to transfer data to/from I/O devices. In I/O-mapped I/O, a reduced number of address lines is used. Only 8-bits (A0 .. A_7) are decoded, address lines (A8 .. A_{15}) are irrelevant for I/O instructions. The only disadvantage in mapping I/O devices as I/O-mapped I/O is the restriction to using the simple IN/OUT instructions.

One of the major problems encountered in the transfer of data between the microprocessor and the peripheral device is the large difference in operating speed. The process of data transfer mechanism between the microprocessor and peripherals can be controlled either by the microprocessor or by the peripherals. For those applications where the speed of the CPU is higher than that of peripheral devices, the CPU will have to wait a considerable amount of time before data transfer. If the program is responsible for controlling I/O, the transfer method is called program-controlled I/O. Alternatively, the situation may be that a large amount of high-speed data has to be captured and transferred to memory. In such situation, a faster transfer can be made by transferring directly data into the memory rather than into the CPU and then out to memory. A data transfer of this form is known as direct memory access or DMA.

There exist several approaches to synchronizing the CPU and its external devices. In the microprocessor-controlled mode, the transfer can take place unconditionally, where the I/O device must always be ready for communication, or it can take place conditionally only when the I/O device is ready for communication. In this latter case, transfer requires some form of handshaking between the CPU and the device prior to the actual transfer of data. There are other types of device-initiated I/O transfer, interrupt transfer, using the WAIT signal, and direct memory access. Peripheral-controlled data transfer mode is generally employed when the peripheral is much faster than the microprocessor. It is the only type of data transfer that is not under CPU control but under the control of some special interface circuits called DMA controllers. Figure 4.3 Illustrates the different I/O data transfer approaches.

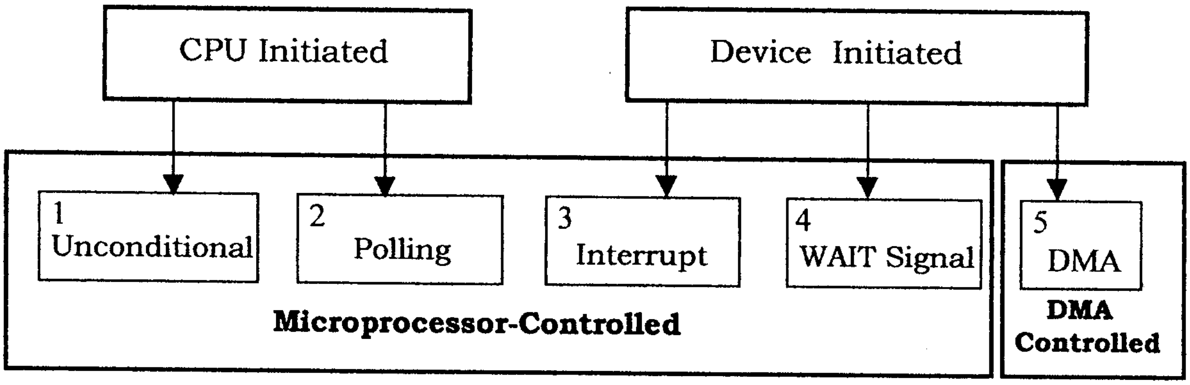


Figure 4.3 I/O data transfer alternatives between the Z80 microprocessor and peripherals

4.3.1 The Unconditional Approach

This type of data transfer is used only in situations where an output device is always ready to accept data from the microprocessor or an input device has data ready for the MPU. In such cases, there is no need of any handshake or exchange of control signals between the MPU and the I/O device.

4.3.2 The Polling Approach

Here the microprocessor continually polls the status of the device to see if it is ready for data transfer. Meanwhile, the CPU keeps executing a loop called a wait loop. This is so called, because the CPU operation stays in this loop while it waits for the I/O device to get ready. An application that uses polling is simple to program. It is appropriate with slower operations.

4.3.3 The Interrupt Approach

Another approach to I/O control that uses a mixture of hardware and software techniques is called interrupt-driven I/O. An interrupt-driven I/O transfer involves the I/O device to send a signal to one of the microprocessor's interrupt lines to inform the CPU that the device requires an immediate service. In general, the interrupt routine requires the microprocessor to respond in an orderly way and since a number of peripheral devices may be interrupting, a priority of servicing has to be set. The advantage of this technique is that the microprocessor is free to perform other tasks rather than waiting in a polling loop. Also, this approach is advised in applications that use high-speed data acquisition. This is the approach that we have chosen in the implementation of our system.

4.3.4 The WAIT States Approach

This approach is preferred when peripheral response time is slower than the execution time of the microprocessor. The WAIT input can be used to add a delay of one or more clock cycles or (T-states) to a processor's instruction execution time to allow it to communicate with slow external devices. This technique provides sufficient time for the peripheral to complete the data transfer and is commonly used in a system with slow memory chips.

4.3.5 The Direct Memory Access (DMA) Approach

DMA is a means of performing I/O data transfer without involving the CPU in the transaction. Bypassing the microprocessor permits direct data transfer from the data acquisition device to memory under the control of special interface circuit called DMA controller. The DMA controller sends a bus request (BUSRQ) signal to the microprocessor, which acknowledges the request by releasing control of the memory and I/O by disconnecting itself from the address, data and control buses. Once the microprocessor is disconnected, another "microprocessor" (a DMA controller) takes over the memory and I/O space.

The major drawback of polled I/O transfer is that, the CPU has to wait for the I/O device. In some cases, especially for slow I/O devices, the microprocessor will waste a lot of its time reading and testing the status of the I/O devices. If there is nothing else for the CPU to do while the I/O device is getting ready, it does not matter. However, in many applications, the CPU can be doing other tasks while it is waiting, such as processing data or communicating with other I/O devices. This can be accomplished by using a more efficient approach, the interrupt mode of I/O transfer.

4.4 The Z80 Interrupts

Unlike subroutines which are predictable events in that they are called up whenever and wherever the program dictates. Real-time situations where the CPU interacts with external physical events are not as simple as that. Often, something happens there, it requires immediate attention of the microprocessor. In this case, a particular mechanism should be provided to handle such external event and prevents the microprocessor. This mechanism is known as interrupt. The interrupt allows an external system to gain the attention of the microprocessor through an interruption. In this case, the CPU drops whatever it is doing and goes to a specific subroutine called the Interrupt handler or interrupt service routine (ISR).

The way the CPU handles interrupts is much like the way ordinary subroutines are handled, although there are some important differences. Normally, subroutines are called with the software, with an interrupt, an external device can demand the attention of the CPU by calling a subroutine through the interrupt structure of the microprocessor. There are pin connections on the microprocessor that, when activated can cause an interrupt to occur.

The main problem with an interrupt-driven I/O is the fact that an interrupt can occur at any instant of time. Due to the randomness of such events, interrupts have acquired something of a reputation of arcane topic. There is an element of truth to this, especially for first time users. It is true that there are certain constraints (case of most asynchronous events) that have to be observed in order for an interrupt routine to work properly.

In the case of an interrupt, a signal on the interrupt input pin of the microprocessor takes the place of the CALL instruction. The main question is how does the CPU find out the address to jump to? There are many different answers to this question. The response of the CPU to an interrupt must be to transfer control to the correct ISR. The major factors to consider are the interrupt type and interrupt mode. The remaining of this section considers the general aspects of interrupts and the principles on which they rest.

Like most microprocessors, the Z80 has two interrupt control input lines which may be asserted by an external device: the Non-Maskable Interrupt ($\overline{\text{NMI}}$) and the maskable INTerrupt ($\overline{\text{INT}}$).

The former interrupt input line, the $\overline{\text{NMI}}$, as its name implies cannot be masked. It is an edge-sensitive input. Whenever the microprocessor senses a negative going edge on its $\overline{\text{NMI}}$ pin and provided that the $\overline{\text{BUSREQ}}$ is inactive, it responses to the interrupt. First it terminates the execution of the current instruction, saves system context (return address and status of IFF1 into IFF2) and disables the maskable interrupt (by clearing IFF1) in order to not be interrupted by the maskable interrupt until the request has been completed. Next, the program counter is automatically loaded with the address 0066h and starts executing the instruction at this address. The $\overline{\text{NMI}}$ interrupt is usually used to handle critical events.

The maskable interrupt is more useful because it can be selectively enabled or disabled by the software. It is a level-sensitive input. When this type of interrupt is enabled (IFF1 and IFF2 set to 1 and $\overline{\text{BUSREQ}}$ is inactive), each time the hardware device generates the interrupt pulse, the interrupt controller prioritizes multiple requests and issues a single interrupt (a negative level to the $\overline{\text{INT}}$ input) to the microprocessor. The CPU responds with an interrupt acknowledge cycle (by asserting both $\overline{\text{M1}}$ and

$\overline{\text{IORQ}}$). The maskable interrupt input can be used in three different modes which can be chosen by the programmer via the software instruction IM i where i := (0, 1, 2). On reset, the CPU always enters mode 0. Table 4.1 depicts the Z80 interrupts.

Type	Mode	Memory Location Program Transfer	Hardware
$\overline{\text{NMI}}$		0066h	Not Required
$\overline{\text{INT}}$	0	0000h, 0008h, 0010h, 0018h, 0020h, 0028h, 0030h, 0038h	Required
	1	0038h	Not Required
	2	Almost Anywhere in the Memory Map	Required.

Table 4.1 The Z80 interrupt types and modes.

i- Mode 0

This mode provides eight different destinations addresses or vectors as shown in table-41, using additional hardware.

ii- Mode 1

When set to mode 1, the program execution is transferred without any additional hardware to memory location 0038h. This mode is advantageous for small systems in that it does not require any external hardware for program transfer.

iii- Mode 2

The third mode of operation, mode 2, is by far the most powerful, in that, it allows multiple interrupt inputs for large systems. It does require however, external hardware and presetting the interrupt register (**I**). The 16-bit vector address is obtained by combining the 8-bits from the external hardware as the low-order byte (must be even), and the high-order byte from the interrupt register **I**. This 16-bit value is not the actual address of the ISR, but rather an address of memory that contain the starting address of the ISR. The indirect method of getting the ISR address is known as a vectored interrupt and the array of addresses are called interrupt vectors.

As in the case of a subroutine, an ISR is a normal program that can be terminated however, with either RET, RETI or RETN. This last instruction is used normally at the end of the nonmaskable interrupt service routine (in addition to transferring the program execution to the interrupted program, it restores the status of the maskable interrupt by copying the state of interrupt flip flop 2 (IFF-2) into IFF-1.

An ISR should be small and efficiently designed, since in some cases it could be invoked hundreds or thousands of times a second [22]. Generally, an interrupt handler performs the minimum amount of work necessary to service the device, and then exits. At that point, the microprocessor returns to running the process that was interrupted as if nothing happened.

CHAPTER 5

System Design, Implementation And Testing

The purpose of this chapter is to discuss in details the design and implementation of the Interrupt Driven Microprocessor Controlled Irrigation System (I.D.M.C.I.S). The I.D.M.C.I.S is based on an 8-bit microprocessor, the Zilog Z80.

The 8-bit microprocessor is preferred because it is cheap (costs less than 1/50 of a 16/32 -bit microprocessor), cheap to interface (16 and 32 bit machine requires 2 or 4 ROM and RAM chips per word, compared to one for an 8-bit microprocessor) and relatively easy to work with (exactly matches the bus width of the majority of memories and peripherals). Moreover, the Z80 as any other 8-bit microprocessor would not be a terrible windows machine, but running at a speed of 3-Mhz, the processing power is ample and more than adequate for this application.

The I.D.M.C.I.S controller contains one Z80 microprocessor, one 6224 RAM, one 27128 EPROM, a memory and I/O decoders, one peripheral input/output the Z80PIO, four 8255s programmable peripheral interface (PPI) devices, one 8-channel ADC, a hexadecimal keypad and seven-segment displays. The microprocessor executes firmware programs to affect irrigation control. The RAM device is used as a scratch memory and the EPROM contains all of the instruction codes for the operation of the system. The keypad provides the user the selection of options and/or data entry. The four 8255 chips and the Z80PIO provide a total of 14 input/output ports to "talk" to the other components in the system. In order to communicate with the host computer, an 8251 universal asynchronous receiver transmitter (UART) is used to link serially the controller to the P.C via the serial port. The several seven-segment displays and L.E.Ds are used to display the numerous parameters values and status.

5.1 System Memory Map

The first step in developing any system memory is to start planning a decoding scheme. This is done through the use of a memory map. A memory map is a graphical representation which help us gain a mental picture that illustrates which segments are to be used for RAM, ROM, and in some cases, where the memory-mapped I/Os reside. The system includes 2 memory chips: one EPROM and one R/W memory. The EPROM is a 4K-bytes while the RAM is an 8K-bytes. We have used a 3-to-8 decoder to divide the 64K-address space into 8 slots of 8K bytes each. While the RAM occupies one complete slot, the EPROM requires only a half resulting in a double memory address or mirror space.

The memory address decoding is largely influenced by the fact that the address must begin at 0000h. This requirement is necessary because the program counter is cleared and the program execution begins at this location when the Z80 microprocessor is reset.

The map in figure 5.1 shows how the I.D.M.C.I.S memory is allocated. The EPROM occupies the range 0000h up to 0FFFh with a mirror ranging 1000h up to 1FFFh. The RAM is mapped in the slot A000h up to BFFFh.

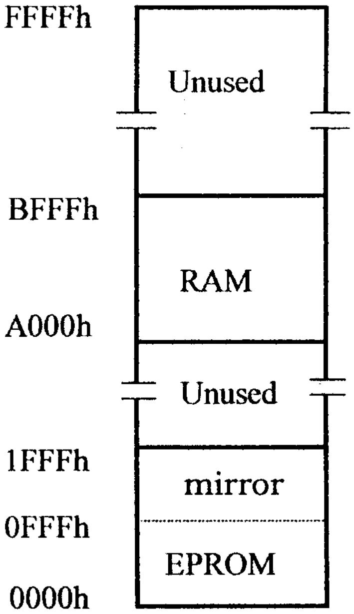


Figure 5.1 The I.D.M.C.I.S memory map.

The memory assignments of the R/W and EPROM chips are given in table 5.1. The 3-to-8 decoder decodes the three most significant address lines. Address line A12 is irrelevant for the EPROM, hence it is marked a don't care.

Device	A15	A14	A13	A12	A11	A10	— — —	A0
EPROM	0	0	0	X	•	•	— — —	•
RAM	1	0	1	•	•	•	— — —	•

0: Logic zero 1: Logic 1 • : Both X: Don't care

Table 5.1 The I.D.M.C.I.S Memory Assignments Table

5.2 System I/O Map

As for the memory organization, the I/O map pictures out how the port numbers (I/O addresses) are allocated. Figure 5.2 shows a map of the ports used in the I.D.M.C.I.S.

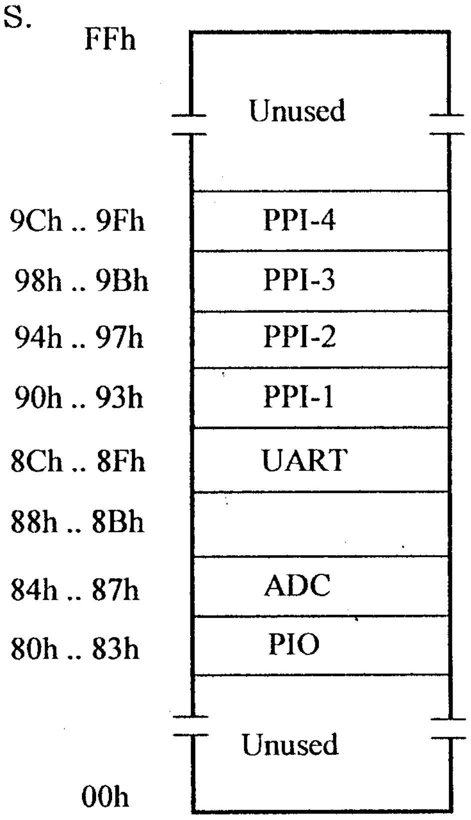


Figure 5.2 The I.D.M.C.I.S Input / Output map

Most of the programmable peripheral adapters used in the system require four addresses, for that reason we managed to break a segment of 32 address locations into 8 blocks of 4 address locations each.

I/O locations 80h .. 9Fh are arbitrarily chosen to be employed for the input/output area. The main reason in choosing such a narrow segment is primary to map most peripherals with absolute addresses hence, minimizing folding spaces and multiple I/O addresses.

Device	A7	A6	A5	A4	A3	A2	A1	A0	Function
PIO	1	0	0	0	0	0	0	0	Data Port-A : Keypad
	1	0	0	0	0	0	0	1	Data Port-B :
	1	0	0	0	0	0	1	0	Control Register A
	1	0	0	0	0	0	1	1	Control Register B
ADC	1	0	0	0	0	1	X	X	ADC address
UART	1	0	0	0	1	1	X	0	Data Register
	1	0	0	0	1	1	X	1	Control Register
PPI-1	1	0	0	1	0	0	0	0	Port-A : Lower Probe
	1	0	0	1	0	0	0	1	Port-B : Shallow
	1	0	0	1	0	0	1	0	Port-C : Sun
	1	0	0	1	0	0	1	1	Control Register
PPI-2	1	0	0	1	0	1	0	0	Port-A : Rainfall
	1	0	0	1	0	1	0	1	Port-B : Wind
	1	0	0	1	0	1	1	0	Port-C : Valves
	1	0	0	1	0	1	1	1	Control Register
PPI-3	1	0	0	1	1	0	0	0	Port-A : Sign.
	1	0	0	1	1	0	0	1	Port-B : Temperature
	1	0	0	1	1	0	1	0	Port-C : Humidity
	1	0	0	1	1	0	1	1	Control Register
PPI-4	1	0	0	1	1	1	0	0	Port-A : Pivot
	1	0	0	1	1	1	0	1	Port-B : Pump
	1	0	0	1	1	1	1	0	Port-C : Alarm
	1	0	0	1	1	1	1	1	Control Register

Table 5.2 The I.D.M.C.I.S Input / Output Port Assignments Table.

The internal I/O port addresses for the different peripherals used are indicated in table 5.2. The programmable input/output the Z80PIO and the programmable peripheral interface the 8255 PPIs are fully decoded whereas the ADC and the UART are partially decoded. The address line A1 is irrelevant for these last two devices and address line A0 is irrelevant for the ADC. The ADC has one "official" address, 84h but it can be accessed with addresses 85h .. 87h.

In order to insure steady progress in the development of the hardware/Firmware/software interfaces, we designed and implemented the controller in a modular manner. The modularity of the design and construction allowed for a systematic method for system testability.

Every time a controller module is added, it was checked for functionality and its interaction with the whole system. The following sections describe the different subsystems or modules that make the I.D.M.C.I.S controller.

5.3 Microprocessing Unit

This unit consists of a Z80 microprocessor chip and unidirectional and bi-directional buffer driver chips. The microprocessor requires additional circuitry to generate appropriate control and clock signals. The clock of the crystal determines the speed of the Z80. The crystal is a 3-Mhz oscillator, which gives a CPU cycle time of approximately 330-nsec, figure 5.3.

The Z80 microprocessor, like most of the large-scale integration (LSI) chips that belong to the NMOS, CMOS or HCMOS logic families, produces very small drive currents at its pins. It has no trouble driving loads of the same families because these loads require very little input current. This is not true, however, when it comes to drive TTL loads. Hence, when configuring a microprocessor-based system, particular attention must be paid to the drive capabilities of the microprocessor and other devices. To do so, buffering drivers are provided as current amplifiers for the address and data buses. The buffering allows these buses to drive the numerous I/O and memory devices that are interfaced with the Z80, and also provides its protection by isolation.

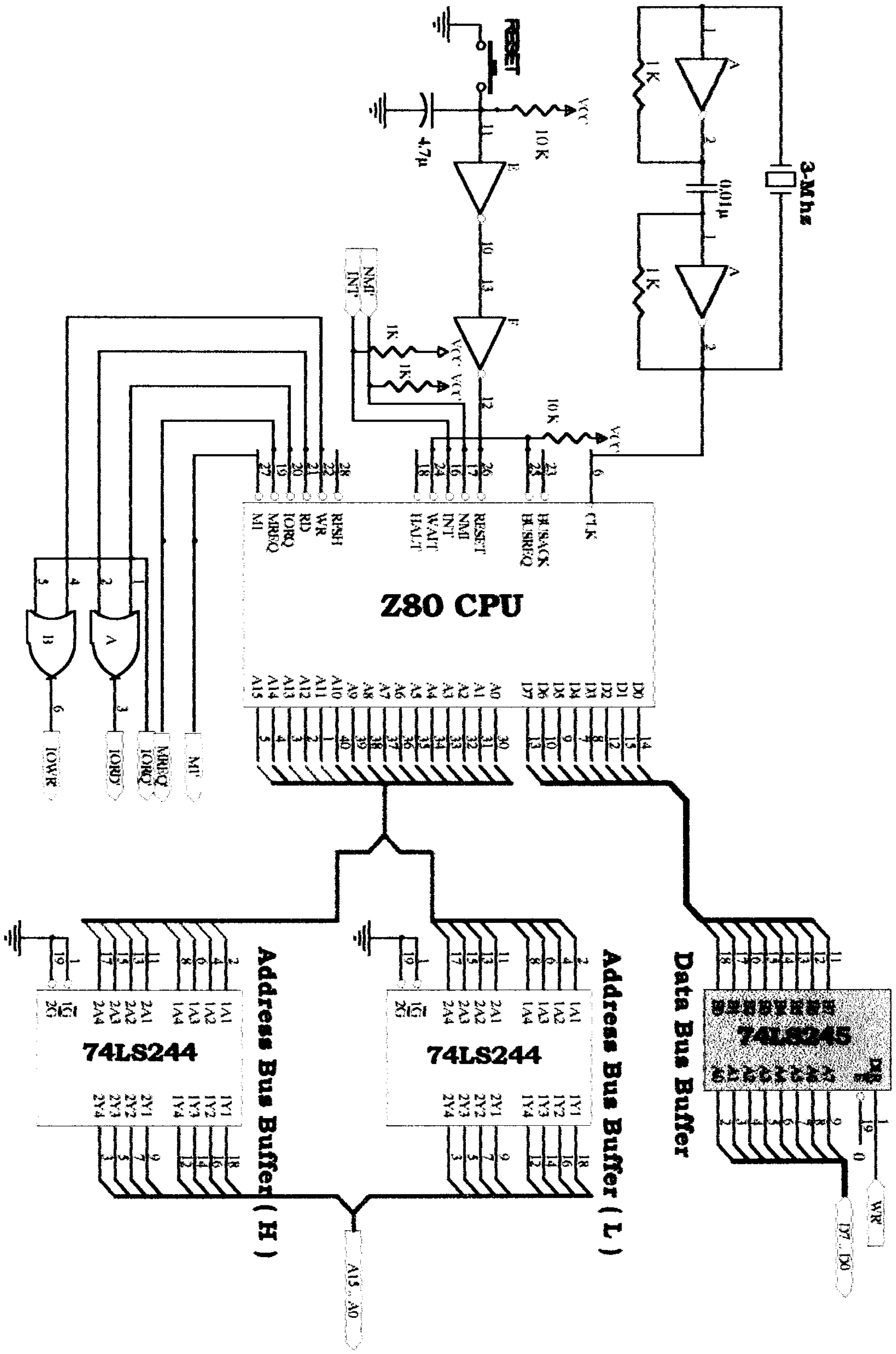


Figure 5.3 Microprocessing and Buffered Busses Unit

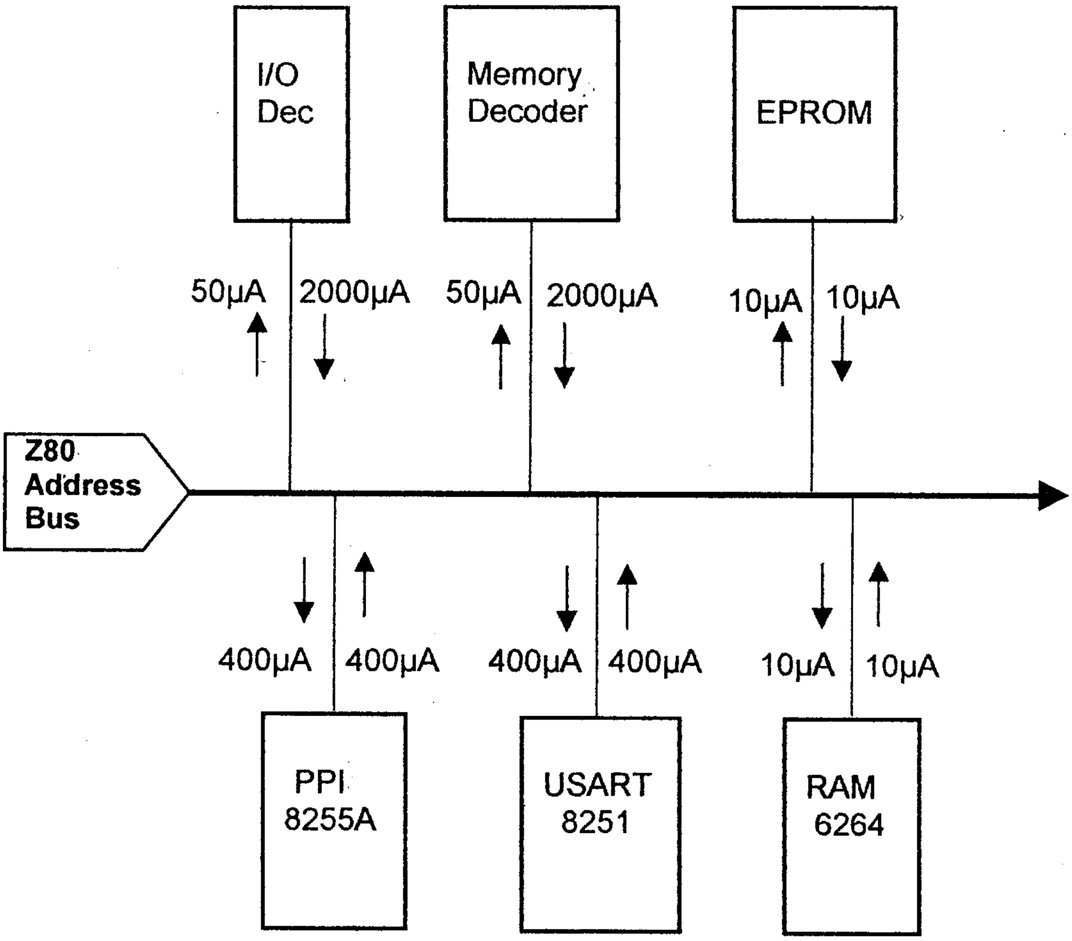


Figure 5.4 The I.D.M.C.I.S Address Bus Loading

The unidirectional Z80's address bus can source up to 250 µA and sink no more than 1.8 mA. The total load on the address bus can be estimated by calculating the total high level input current (I_{IH}) and the total low level input current (I_{IL}) of the different I/O and memory devices driven by the address bus, as illustrated by figure 5.4.

i- Total high level input current (I_{IH})

2 decoders	50µA x 2	=	100µA
EPROM 2732		=	10µA
R/W 6264 memory		=	10µA
5 PPI 8255	400µA x 5	=	2000µA
UART 8251		=	400µA
TOTAL		=	2520µA

ii- Total low level input current (I_{IL})

2 decoders	2000μA x 2	=	4000μA
EPROM 2732		=	10μA
R/W 6264 memory		=	10μA
5 PPI 8255	400μA x 5	=	2000μA
USART 8251		=	400μA
TOTAL		=	6420μA = 6.42 mA

The results of the above calculations conclude that the driving capacity of the address bus must be buffered to withstand the I/O and memory devices current requirements. Two octal buffers (74LS244s) are used as address bus drivers. They are capable of sourcing up to 15mA and sinking up to 24 mA of current.

The data bus has similar driving capacity as the address bus. The loading on the data bus, however, varies considerably, as the bus is bi-directional. To ensure the driving capability of the Z80's data bus, an octal transceiver the 74LS245 is used. This bi-directional octal buffer is capable of sourcing up to 15 mA and sinking up to 24 mA of current. These current capacities are large enough to withstand the system's I/O and memory devices current requirements.

The direction of data flow is controlled by the 74LS245's DIR input signal. By connecting $\overline{WR}$ to this input signal, data byte will flow either from the microprocessor to the peripheral side when $\overline{WR}$ is active or the other way around if it is inactive.

The $\overline{RESET}$ signal is an active low input. When this input is brought low, the system resets: the program counter is forced to 0000h, both I and R registers are cleared, the interrupt request is disabled and set into mode-0. When the reset key is depressed, the $\overline{RESET}$ goes low and slowly rises toward +5V with a time constant of around 50 msec. The two Schmitt trigger inverters are implemented to provide a sharp pulse.

5.4 Decoding and Memory Unit

The hardware for this unit is pictured in the schematic of figure 5.5. The reason of having two 3-to-8 decoders (74138s) is due to the fact that we have preferred to map the I/O devices as I/O mapped I/O. The upper decoder is to generate the required chip selects of the different programmable I/O devices, while the lower is used to generate chip selects for the R/W and EPROM memory devices.

The two memory chips are one EPROM of 4K-bytes and one R/W memory of 8K-bytes. Since we only have 2 memory devices to map in the whole memory address space, it is not truly important of how much mirroring is involved. The memory decoder is driven by A15, A14 and A13 to break the memory map into 8 blocks of 8K bytes each. While the RAM occupies one complete slot meaning absolute decoding, the EPROM, which contains the system firmware program, occupies only a half, leading in a double memory addresses.

The system needs a RAM chip for three different tasks:

i- As a stack to save return addresses whenever a subroutine or an interrupt service routine is called.

ii- To store temporarily the intermediate results obtained from calculation

iii- To store/retrieve the contents of registers pairs when PUSH/POP instructions are executed.

iv- To store set-point variables

The memory address decoding is largely influenced by the fact that the address must begin at 0000h. This requirement is necessary because when the microprocessor is reset, the program counter is cleared and the program execution begins at this location. Another major problem to consider is time requirement. It is necessary to synchronize the execution speed of the microprocessor with the response time of memory. This can be accomplished using the WAIT signal input to the Z80 microprocessor.

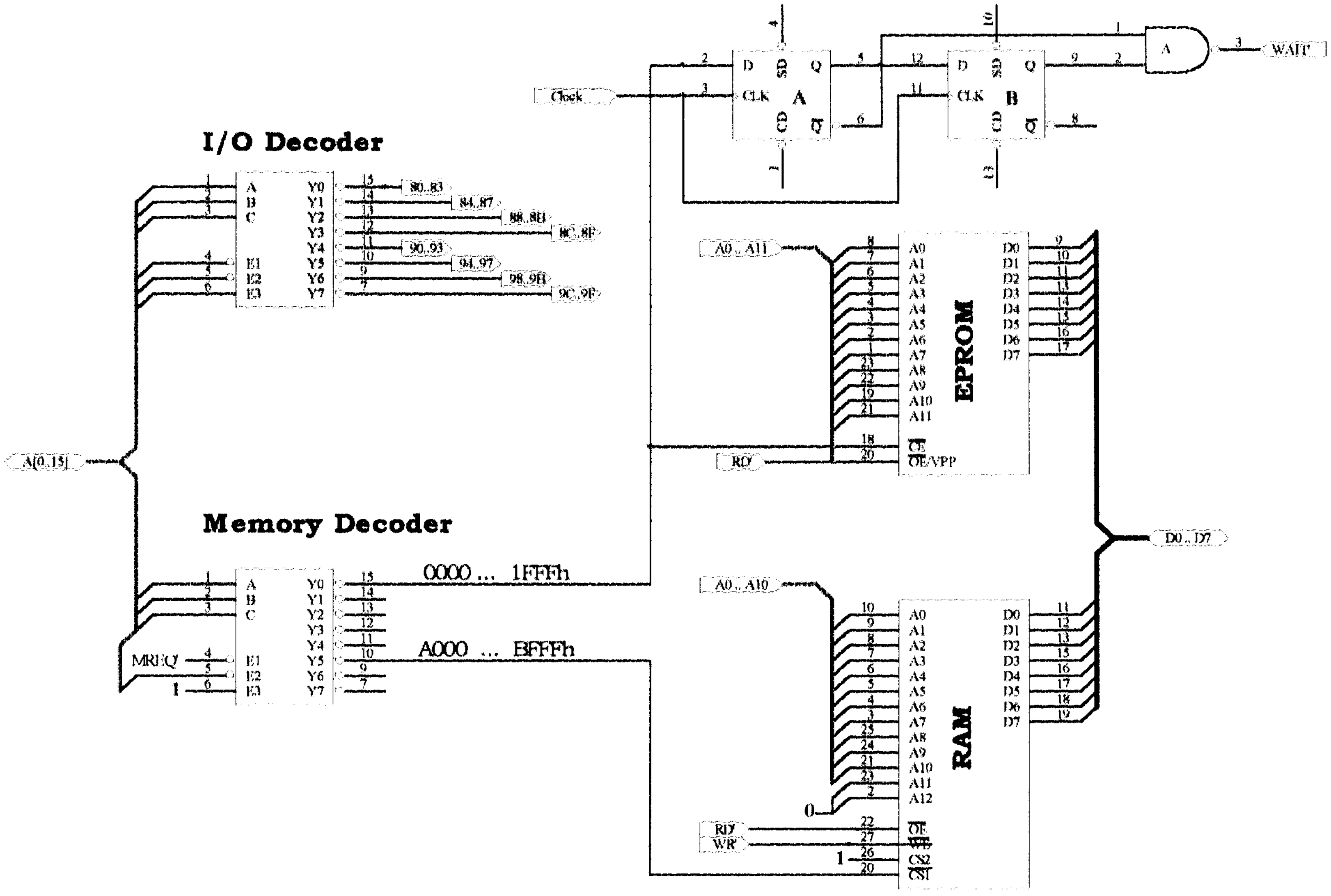


Figure 5.5 The Decoding, System' s Memory and Wait State Generator Unit

To ascertain whether a given memory chip is too slow in comparison with the execution speed of the microprocessor and needs wait states to synchronize the data transfer, we must examine the timing requirements of the microprocessor and the response time of the memory.

Figure 5.6 illustrates the Opcode Fetch machine cycle. It is the most restrictive because the microprocessor begins to read the data byte at the rising edge of T_3. if the memory chip can respond adequately in the Opcode Fetch cycle, we need not be concerned with the memory read cycle.

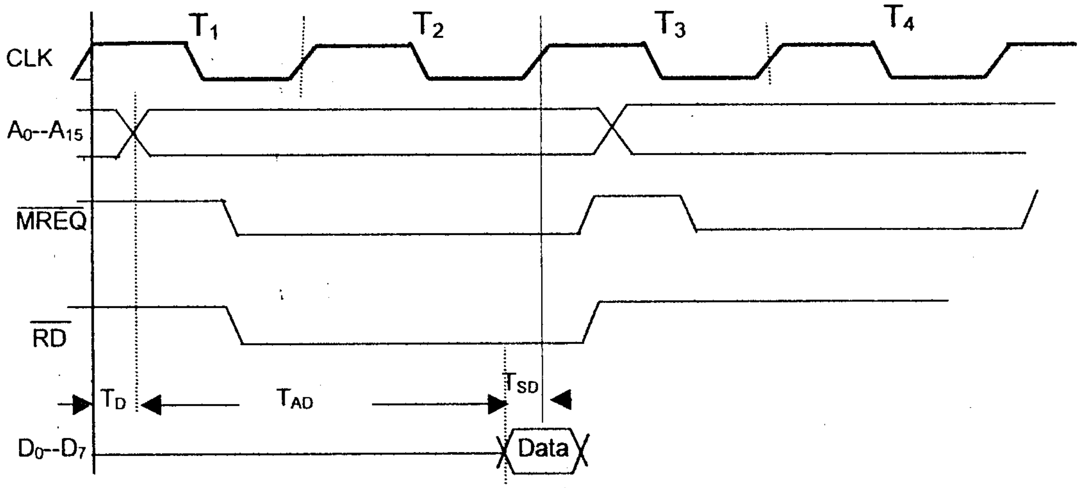


Figure 5.6 Z80 Opcode Fetch machine cycle [22]

Figure 5.6 shows that the time interval T_{AD} between the application of the address and the time it has read the byte. In a 3-Mhz system (T = 330 nsec)

$$T_{AD} = (2 \times T) - T_D^* - T_{SD}^*$$

(* These specifications are obtained from Z80 AC characteristics)

The EPROM access time is 450-nsec, the delay in the decoder is 20-nsec and the system' s clock period is 330-nsec. In the Opcode Fetch cycle, the Z80 CPU reads the data byte at the rising edge of the T_3 cycle. Therefore, the allowable microprocessor read time is:

$$T_{AD} = (2 \times 330) - 145 - 50 = 660 - 145 - 50 = 465 \text{ nsec}$$

The EPROM would require:

Memory read time = memory access time + address decoding delay

= 450 + 20 = 470 nsec.

These calculations show that the memory requires 470-nsec and the microprocessor would begin to read in 465-nsec after placing the address on the address bus. This situation requires the addition of one Wait State. After adding one Wait State, the microprocessor read time is extended to 795-nsec. This leaves 325-nsec as a safety margin for the memory read time.

The only requirement to add Wait states is to keep the WAIT signal low. The Wait state generator generates the WAIT signal. This circuit consists in NANDing the outputs of the 2 edge-triggered D-flip-flops figure-5.3. The flip-flop input D_A is always high unless the microprocessor is accessing the EPROM. We can safely assume that Q_B is initially high. As D_A goes low, after the falling edge of T_1, figure-5.6, this value is delayed to Q_A at the rising edge of T_2 and pulling at the same time $\overline{Q_A}$ to a high state. The NANDing of $\overline{Q_A}$ and Q_B drives the $\overline{\text{WAIT}}$ pin of the Z80. Because $\overline{\text{WAIT}}$ is low at T_2, the Z80 extends the memory machine cycle by one clock period. At the next clock cycle (T_3), Q_B goes low, causing the WAIT signal to go inactive and hence, each time the microprocessor accesses the EPROM, the Wait state generator circuit will add one Wait state to the memory machine cycle.

The 6264 RAM does not require extra Wait states given that it has a relatively short access time. It is of the order of 55 to 70-nesc.

5.5 The Keyboard and Data Acquisition Unit

This unit is used to feed the controller with data and commands. The user enters set-point variables and commands via the keyboard, whereas the controller automatically gathers data from sensors via the analog to digital converter.

5.5.1 The Data Acquisition Part

The data acquisition unit is built around the ADC 0808. It is an 8-bit A/D converter. It works on the principle of the successive approximation conversion technique. The device has 8 multiplexed analog input channels, out of which any one can be selected. The ADC has also an address latch as well as a tri-stated output latch. The former is clocked via the address latch enable (ALE), while the latter can be enabled by asserting the output enable (OE) input.

To acquire data from the outside world, the A/D converter needs interfacing through the microprocessor. This requires both hardware and software. Figure 5.7 shows how the ADC 0808 is connected to the Z80 microprocessor. The data lines D7.. D0 are connected directly to the microprocessor's data bus. This facility is made possible by the fact the ADC 0808 has tri-stated output data lines. Hence, this feature eliminates the need of linking the ADC 0808 to the system' s data bus via an input port.

To initiate a conversion, an input channel of the multiplexer is addressed by a software instruction via address-input lines A, B and C. Also, instead of using an output port to drive these select lines, we preferred to simply connect them directly to the data bus lines D0, D1 and D2 respectively. Information on these lines that has to be placed by the program on the data bus will be selecting the analog input lines. The information is removed when the desired channel has been selected.

The conversion begins on the falling edge of START CONVERSION (SC) pulse. For these reasons both ALE and SC are tied together and driven by the $\overline{\text{WR}}$ signal. By writing a dummy data into the A/D converter the microprocessor asserts both ALE and SC high.

Once the ADC 0808 starts conversion, the end-of-conversion (EOC) output line goes low. Upon completion of the conversion, the EOC goes high. This low-to-high transition is used to instruct the microprocessor that the conversion is complete and that the data are ready to be read.

The EOC output signal of the ADC 0808 is used to drive the input clock of an edge-triggered D flip-flop. The output $\overline{\text{Q}}$ of the flip-flop, labeled $\overline{\text{EOC}}$ is used to issue an interrupt via the priority encoder (interrupt unit). The $\overline{\text{EOC}}$ signal will stay active until the microprocessor acknowledges the interrupt request by asserting both $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ low. These signals are ORed to reset the flip-flop pulling $\overline{\text{Q}}$ high.

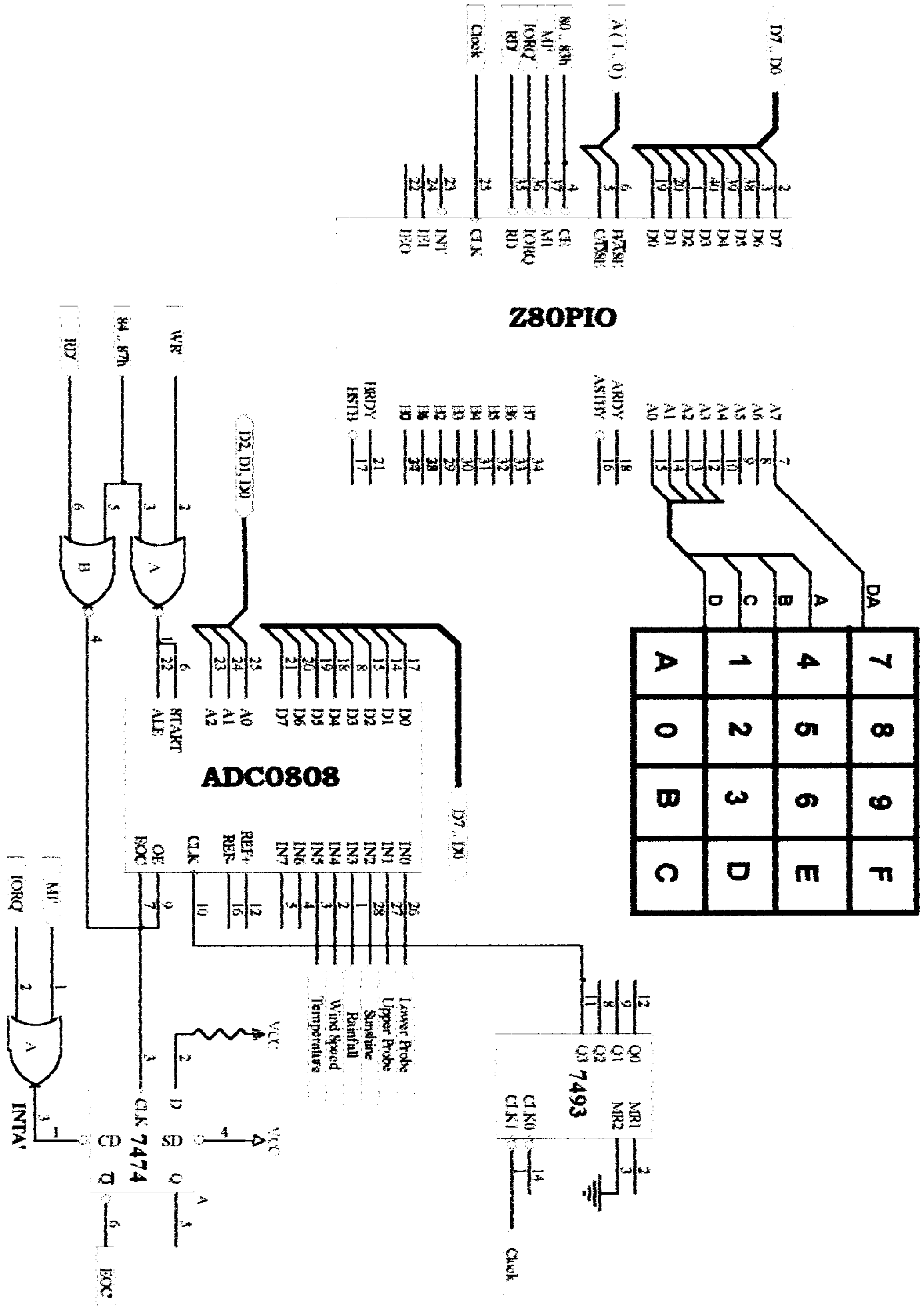


Figure 5.7 The Keypad and Data Acquisition Unit

To get a clock frequency necessary to operate the A/D converter, we used the 7493 counter to divide the 3-Mhz system clock by four, as shown in figure 5.7.

5.5.2 Keyboard unit

The keypad is for the numeric entry of data, such as the time interval, soil moisture threshold, and also use for the selection of various operation types.

The hexadecimal keypad consists of 16 keys arranged in a matrix form. The keypad is organized into two logical groups of keys: the data keys and the function keys. The data keys are the decimal digit keys (0 ·· 9). The function keys are the letter keys (A ·· F).

The keypad is connected to the four least significant bits of the A side of the PIO. The Data Available (DA) is connected to the PIO's A7 input. When a key is depressed, the DA is pooled and causes the microprocessor to call up a routine to read the code of the key that was closed. A sounder is provided to "beep" each time the user presses one of the keys of the keypad. This annunciator tone provides some aural feedback to indicate that the press actually made contact.

The functional description of the keys of the latter group:

Key-A: Automatic Mode

Depressing key-A ignites the automatic mode operation. In this mode, the keypad relinquishes control to the microprocessor (all subsequent key actuations are ignored). The keypad will regain control when the user pushes the RESET key. The irrigation process can be interrupted at any instant of time by depressing the STOP pushbutton.

Key-B: Moisture Data Entry Function

Depressing key-B, instructs the controller that the next two decimal digits entered stand for the soil moisture threshold. This threshold will be displayed on the seven-segment displays mapped at output port (90h).

This threshold value is also converted to binary and stored in memory location A300h.

Key-C: Soil Type Data Entry Function

This key function is used to instruct the controller that the next digit stands for soil type. We categorized soils in 4 categories as follows:

1: Clay
2: Clay loam
3: Sandy loam
4: Sandy

in order to minimize runoff losses, the soil type is used to determine the pumping rate during irrigation application. We designed our system to have up to four different pumping rates. Soil type -1- having the slowest infiltration rate will have the slowest pumping rate, whereas the sandy soils with their high infiltration rates will be sprayed with larger pumping rates.

Key-D: Time Interval Data Entry Function

Depressing the key-D function instructs the controller that the next two decimal digits represent the duration of the irrigation application in the "Timer " mode. The time intervals can durate from 00 up to 99 time units (minutes).

Key-F: Timer Mode

The key-F function is used to initiate the " Timer " or semi-automatic mode irrigation process. The irrigation starts and lasts the number of time units entered by the user provided that the weather conditions (wind speed and temperature) are favorable for an irrigation application. While the irrigation proceeds, the remaining time interval is continuously displayed on the user interface. The irrigation application can be resumed at any instant of time by activating the STOP pushbutton.

5.6 The Interrupting Unit

Mode-2 suggests that almost unlimited possibilities to vector interrupt requests anywhere in memory. Figure 5.8 illustrates the connection of three interrupting devices to the maskable interrupt ($\overline{INT}$) input line of the Z80. Here, we have more than one interrupting source to be connected to the interrupt input. For the processor to distinguish in case of an interrupt which device requests attention, it is necessary to prioritize the interrupts. We have used 8-to-3 priority encoder (74LS148) to deal with the three possible interrupting devices. Theses are in descending order of priority: the STOP pushbutton, the START pushbutton and the end of conversion ($\overline{EOC}$) generated by the ADC and " conditioned " by the delay flip-flop the 7474, figure 5-7. The 8-to-3-priority encoder (74LS148), in addition to determining the priority among the interrupting devices, encodes the interrupting input. In other words, when an interrupting device requests service, one of the 3 input lines (I_7, I_6 or I_5) of the encoder goes low, making the output line $\overline{GS}$ low which in turn interrupts the microprocessor. Once the interrupt is acknowledged, the $\overline{INTA}$ signal is asserted low enabling the latch 74LS573, the code corresponding to the input is placed on data lines D_3, D_2 and D_1 with D_0 being permanently kept at logic 0 (the Z80 expects the vector address to be even).

If for example, the START push button is activated, the output of the encoder will be $(001)_2$. This code is inverted and placed on data lines D_3, D_2 and D_1 owing $D_0 = 0$. D_7, D_6, D_5 and D_4 being permanently high, the byte $(1111\ 1100)_2$ or FCh is thus placed on the data bus. If the interrupt register (**I**) is initialized with the byte 0Fh, then, the Z80 will form the vector address 0FFCh for the START pushbutton. Similarly, it will form the vector address 0FFEh for the STOP pushbutton and the vector address 0FFAh for the EOC the output of the ADC.

The address 0FFCh for example, is not the starting address of the ISR requested by the pushbutton STOP, rather it is the address of a memory location which contains the starting address of the actual service routine.

A depress on START pushbutton initiates the irrigation process unconditionally i.e., regardless of the soil moisture or weather parameters.

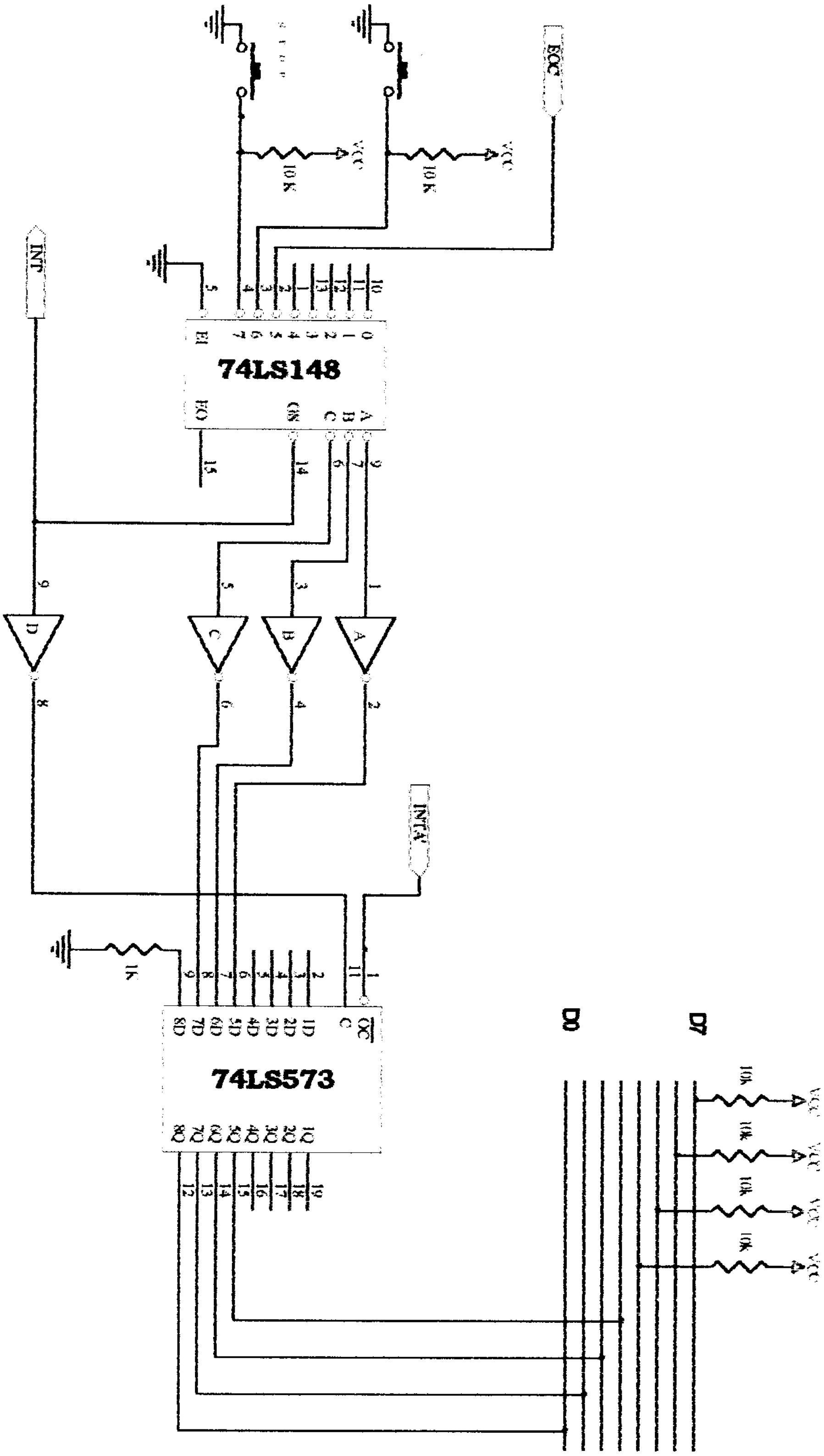


Figure 5.8 The Mode-2 Multiple Interrupt Mechanism

This feature may be of interest to control the peaks of high and low temperature. During the frost event, the irrigation system may spray crops to warm up the atmosphere. Similarly, irrigation may be applied to protect crops from heat stress by cooling up the crops.

5.7 The Display Unit

The display unit consists of fourteen 7-segment displays. Figure 5.9 contains the 7-segment displays for lower and upper soil moisture levels and for the sunshine radiation. Figure 5.10 depicts the numerical displays for ambient temperature and humidity, whereas, figure 5.11 contains the displays for wind speed and rainfall level.

Although the microprocessor processes data in binary form, all the numbers are displayed in decimal to simplify readings.

5.8 The Motors Driving Unit

When the controller makes a decision to apply irrigation water it turns-ON the pump and the pivot motors. Figure 5.12 illustrates how the controller drives these two motors (simulated by two low power DC motors).

We assumed that the speed of the pivot motor should be constant, for this reason the pivot motor is driven by a power Darlington transistor. The pumping rate however is should be variable by varying the speed of the pump motor. Several techniques can be used to control the speed of a DC motor, the most common of which is the Pulse Width Modulation (PWM) technique.

Whichever method is to be used, the speed of the pump motor should strongly be correlated to the soil type. The pumping rate should be in such a way that the runoff losses are minimized. To do so, the application rate should not exceed the soil infiltration rate. Consequently, the finer the soil texture is, the slower the pump motor speed should be. As described previously (section 5.5.2), we subdivided the soil types in four categories (we could have more), and hence we can have up to four different pumping rates.

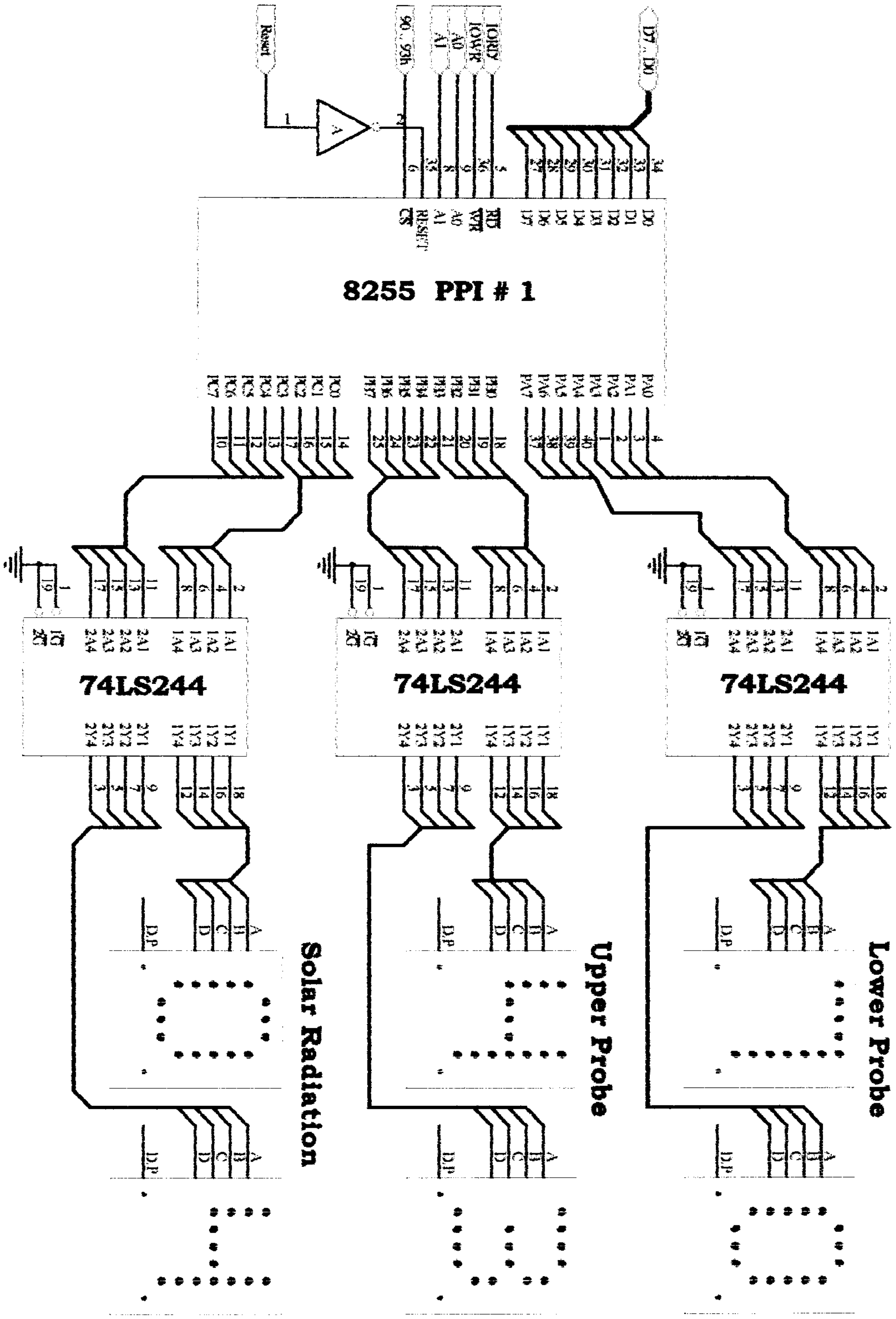


Figure 5.9 Soil Moisture Contents and Solar Radiation Displays

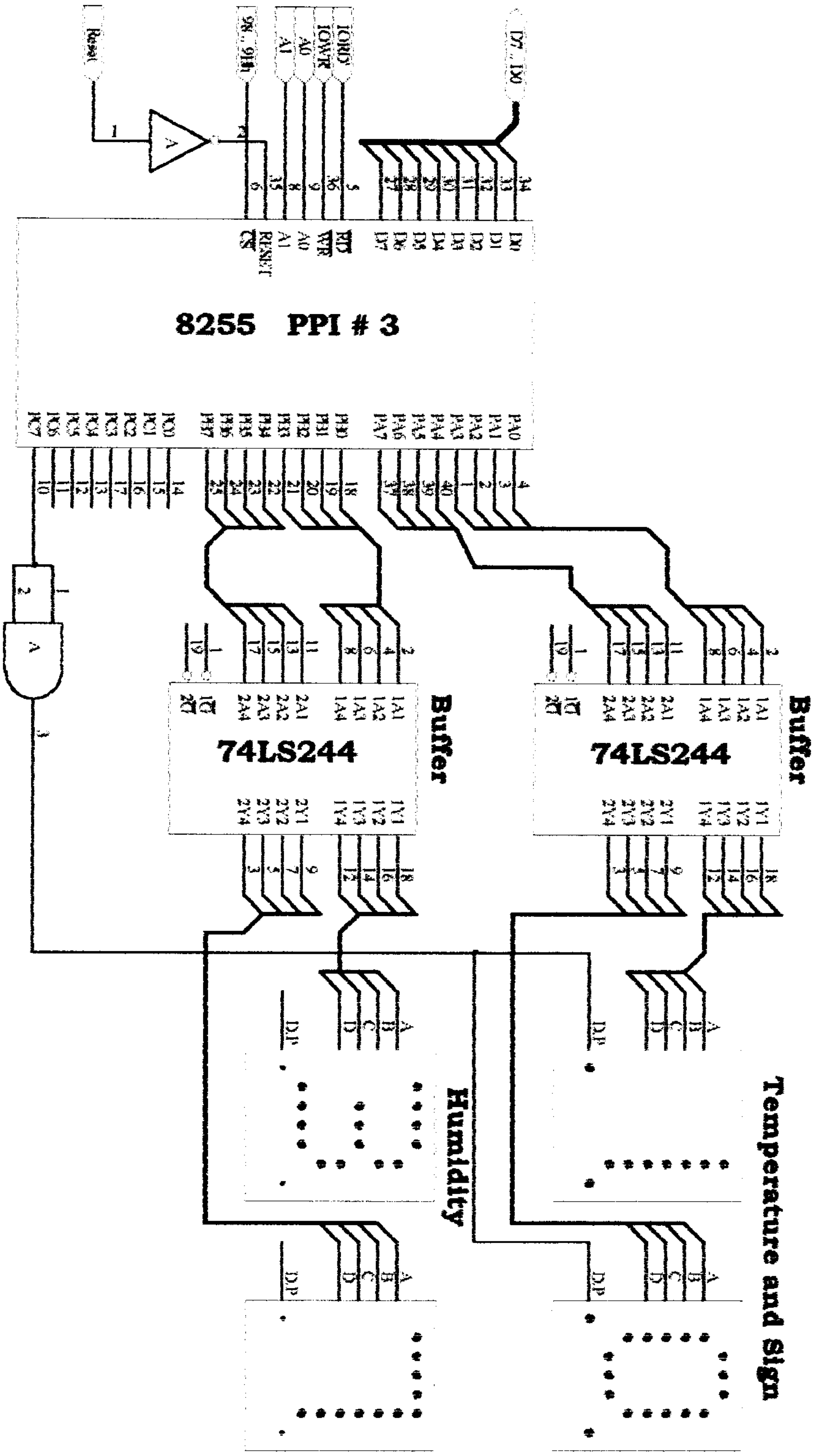


Figure 5.10 Ambient Temperature and Humidity Monitoring

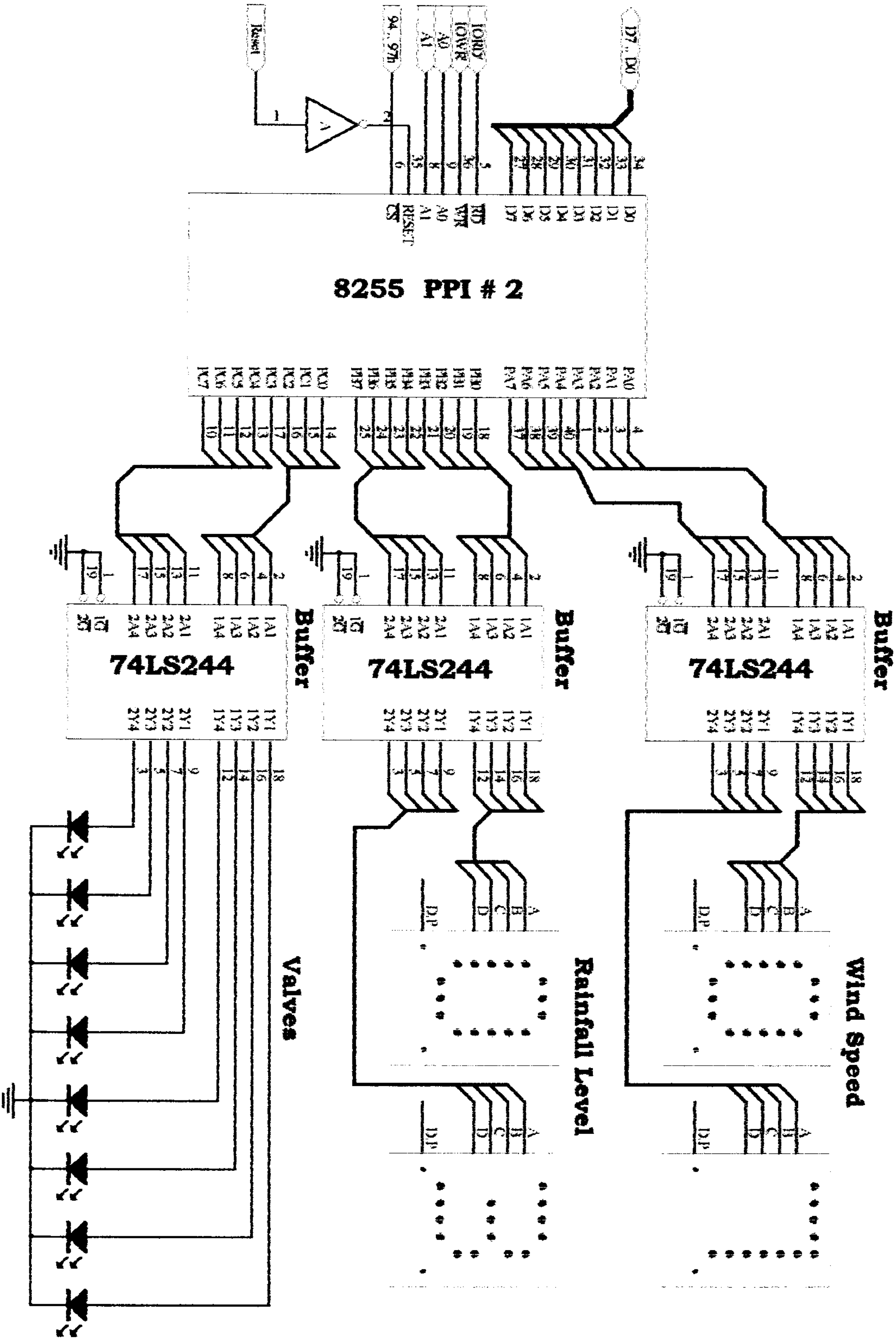


Figure 5.11 Rainfall Level, Wind Speed and Valves Monitoring

Based on the soil type, the controller will output the corresponding digital word to drive the digital to analog converter (DAC) connected to the 8255' s port A. An operational amplifier is connected to the DAC' s output to convert the current output into a voltage. The variable resistor R1 as shown in figure 5.12 is used for gain adjustment, whereas,resistor R2 is used to limit the current going into the base of the transistor.

5.9 The Fault Detection and Alarm Unit

Fault detection and handling as such was not a primary focus of our present work. However, since the work proposed is to control a physical system, it became necessary to implement at least some basic degree of fault detection and handling to ensure at least a minimal amount of predictability and safety when using the system.

Fault detection - a research area on is own - refers to the ability to detect errors within a system. The type of errors, which occur, can generally be classified into several categories. The most common are: software errors (caused by bugs or design oversights in the software), timing errors (they occur when real-time tasks fail to meet their timing requirement), and hardware errors (they involve the failure of hardware components, or the communication interface to the hardware) [66].

We designed our system such that no timing error can occur. First, the WAIT state mechanism is used to synchronize the data transfer between the microprocessor and the memory, eliminating the probability of any timing errors during a fetch or a memory write cycle. Second, the use of the interrupt driven mode for data transfer between the microprocessor and peripherals prevents any timing errors. This is due to the fact that a data is acquired exclusively when it is ready.

It is known that in any acquisition system, the performance of the management and control system is strongly affected by the quality of the acquired data. In our work, we were limited to detect errors that may occur in the different sensors. Some of these errors are relatively easy to detect but others like sensor drifts are extremely hard to detect.

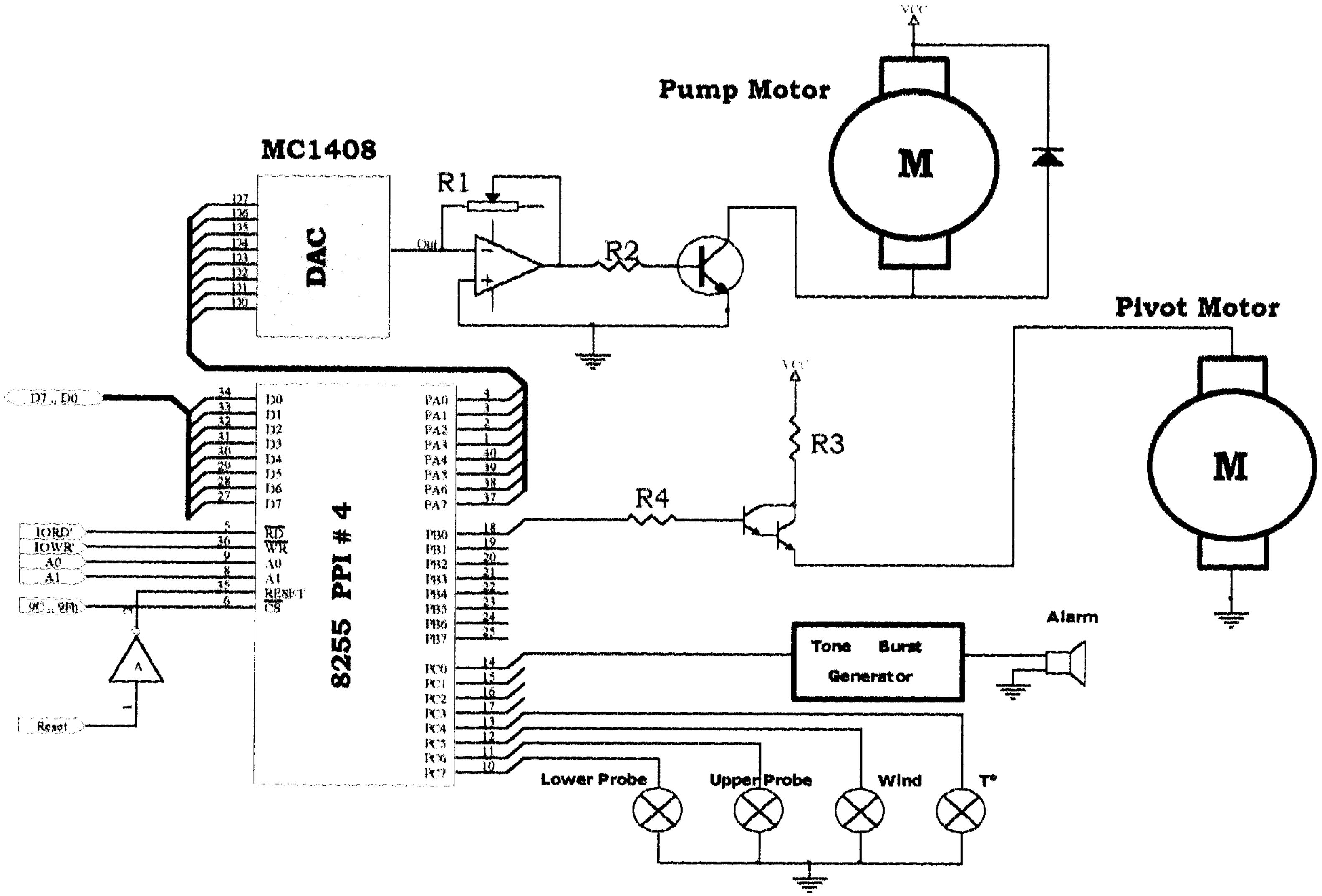


Figure 5.12 The Pump and Pivot Motors Interface and the Fault Detector Circuit

Sometimes, it is easy to detect that there is a fault but to identify what the fault is, can be rather difficult. For this purpose, a fault detection module is implemented to detect sensors malfunctions figure. The module is used to signal any opens or shorts in sensors. Visual indicators and a sound alarm identify these faults as illustrated in figure 5.12.

If a soil moisture sensor indicates a null moisture in the soil or if a temperature sensor indicates an extremely high or an extremely low ambient temperature (situations that never occur in real life), it means that the corresponding sensor is faulty. In order to minimize water and energy wastage, the irrigation process is interrupted as soon as a fault is detected.

Programming in machine language is difficult. Unlike Pascal or even C, the structure of a machine language program is unstructured. Our moderate sized machine language program would have completely overwhelmed the ability of human reader to understand it and become impossible to the designer to troubleshoot it if constructed as one block. The solution was to to clump the program into softawre modules. Then:

- Each module was dedicated to a clearly defined function.
- Control transfer takes place between modules rather than between individual instructions.
- The module themselves are arranged in a heirarchy, in which a master routine calls the subsidiary modules (up to 4 levels of subroutine nesting).
- Each module can be individally tested as a small program.

With this approach, bugs are minimized and systematic procedures are applied to critical error points to eradicate them entirely.

5.10 The Serial Digital Communication Unit

This unit provides a means to the controller to communicate serially with a personal computer (P.C). It is known that the serial port of the P.C uses the RS-232-C (Recommended Standard # 232 release C) data communication standard. For this reason, the microprocessor is interfaced to the P.C through the Intel 8251 universal Asynchronous Receiver Transmitter (UART) chip that handles both transmission and reception of data. It does the conversion parallel/serial during the transmission and the serial/parallel conversion during the reception [73].

Long time ago (before the birth of the P.C), the Electronics Industry Association (EIA) has adopted the RS-232-C serial data communication standard [34]. This oldest and widely used standard defines the logic levels as follows:

Logic-1 = more negative than - 3V
Logic-0 = more positive than + 3V

Any voltage between these two voltages has an undefined logic level.

In conventional microprocessor systems, the standard logic voltage levels conform to the standard TTL voltage levels are 0V and + 5V for a 0 and 1 respectively. The TTL levels are not compatible with the negative true logic of the RS-232-C standard and consequently some type of interface circuit is needed to convert between TTL and RS-232-C voltages.

The RS-232-C buffering and level translation is accomplished through the use of the Motorola MC1488 line drivers and the MC1489 line receivers. Figure 5.13 shows on one side, how the UART is connected to the bus system of the microprocessor along with a toggle flip-flop used to divide the frequency of the clock signal of the system to the clock frequency necessary to synchronize the internal operations the UART. On the other side, how the level conversion chips interface the UART and the DB-9 terminal. Connecting the TxCLK and RxCLK to the CD 4040 BC baud generator does the synchronization of both transmission and reception.

The serial I/O – the 8251 UART – handles the communication with the P.C. it sends data to the windows application and receives commands from the P.C. The communication between the P.C and the controller is done using the interrupt mode. The control signal RxRDY is used to interrupt the microprocessor. It is connected to the $\overline{\text{NMI}}$ input.

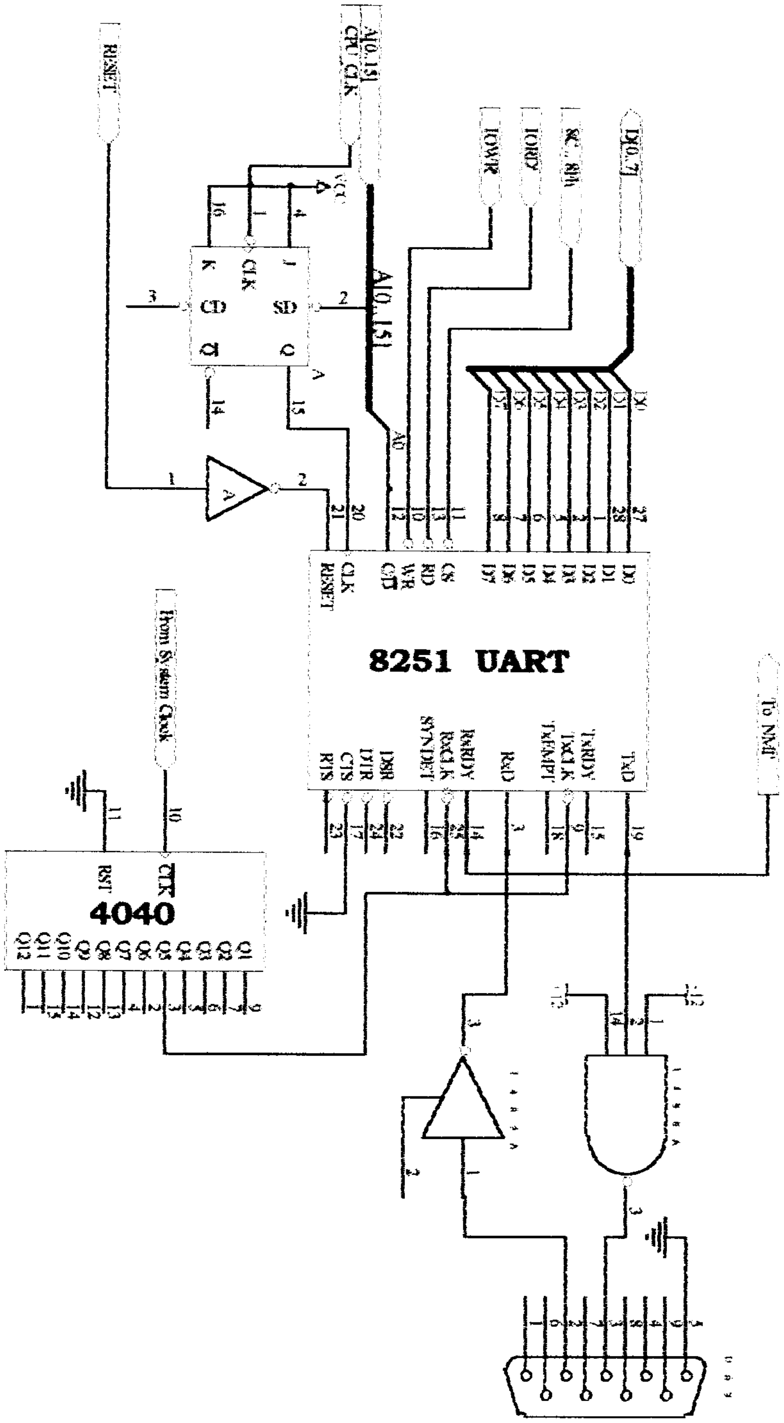


Figure 5.13 Digital Serial I/O Communication with the PC's Serial Port

When the controller wished to transmit a character to the P.C, the following steps occur:

i- The microprocessor sends parallel TTL-level data to the UART

ii- The UART converts these to serial TTL data at its TxD output

iii- The serial TTL data are converted to serial RS-232-C data by the MC1488 chip and transmitted to the PC at a rate determined by the baud rate generator the CD 4040 BC.

iv- The P.C receives the serial data via its serial port

When the P.C is transmitting to the controller, the following steps occur:

i- the computer outputs an RS-232-C serial data signal representing a character

ii- the MC1489 chip converts it to a serial TTL data signal to drive the UART' s RxD input

iii- the UART converts the received serial signal to parallel data that can be read by the microprocessor over the data bus.

The 8251 is interfaced such that the control register is accessed with the port address 8Dh, and the data register is accessed with the port address 8Ch. The UART is configured with as follows: asynchronous mode, two stop bits, no parity and 8-bit character.

Like most Intel programmable devices, the 8251 does not have an $\overline{\text{IORQ}}$ signal; it has $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals. However, the Z80 identifies its I/O opeartion with $\overline{\text{IORQ}}$ signal. Therefore, ANDing $\overline{\text{RD}}$ and $\overline{\text{WR}}$ as shown in figure 5.3 generate the control signals $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ with the $\overline{\text{IORQ}}$ signal.

5.11 The Controller Firmware

Actually, the physical interconnection of components in a microprocessor-based system does not indicate its function. It is the firmware (hard-coded program in EPROM) executed by the microprocessor that primarily determines the system's function.

The firmware manages all the communications and data manipulations between the processor and the peripherals. It handles the transmission of data to the window application GUI. It communicates necessary sensor and actuator conditions from the IDMCIS to the GUI. Each update is an 8-bit packet code that is decoded by the GUI, followed by 8 bits of the current value or state. It is designed to have three independent modes of operation to take into consideration all possible needs.

The first is the " Timer " mode. Here the user selects the interval during which the irrigation will take place. The controller will start the irrigation whenever certain climatic conditions are favorable. The irrigation terminates when the predefined time interval has collapsed.

The second mode of operation is the " Manual " mode. Here the user has the freedom of starting irrigation whenever he feels it is necessary. Depressing the START pushbutton starts the irrigation process regardless of the soil moisture and weather conditions. The STOP pushbutton interrupts and resumes the irrigation process at any moment.

The third mode is the "Automatic" mode. Here the controller manages continuously the irrigation system without any intervention of the user. The irrigation application is controlled by the actual soil moisture and weather parameters. Continuously, data are collected from the sensors and forwards them to the controller. These data, along with the set points entered by the user are processed to determine whether irrigation is to be applied or not. Then, all measured data along with the system status are sent serially to the window application, where they are displayed for the user.

The general philosophy of the software was based on the following premises:

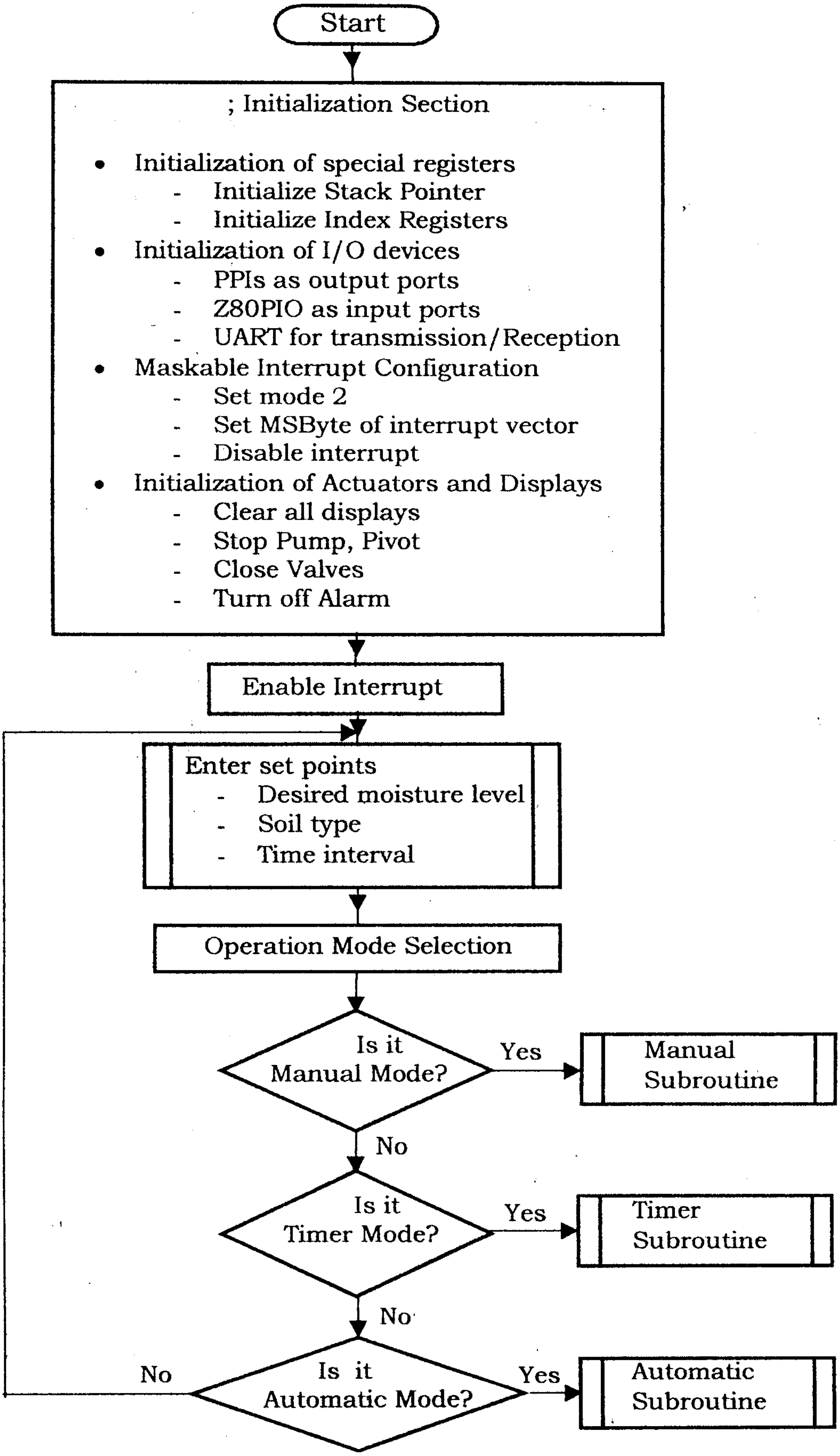


Figure 5.14. Flowchart: the controller's main routine

i- The field controller should have a friendly user interface that is simple to use by irrigators with no particular professional qualifications.

ii- The software should be based on a modular (dedicated subroutines) design basis.

The main routine begins with an initialization section, where the hardware is configured and the variables initialized. The rest consists mainly of calls to subroutines and/or to interrupt service routines (ISRs).

The firmware program was developed using assembly language then manually assembled into machine codes. The flowcharts describing the overall firmware are shown in figure 5.14.

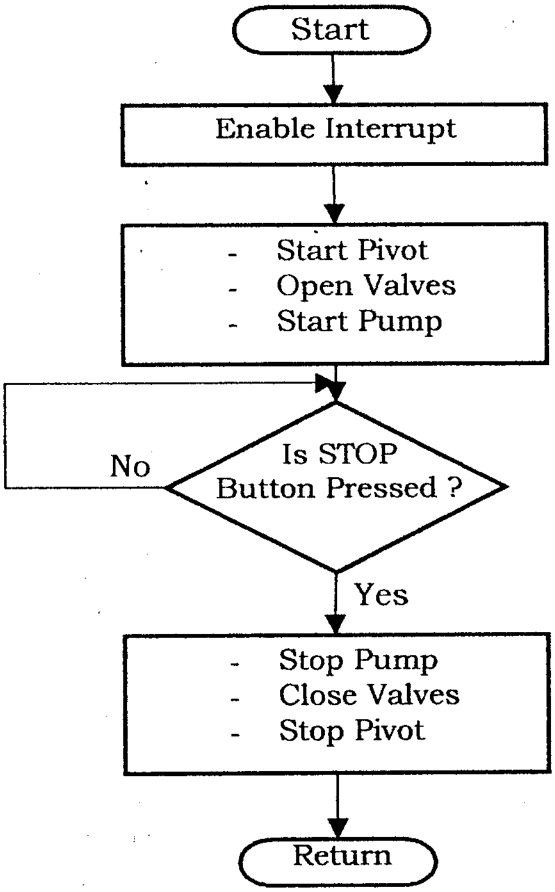


Figure 5.15 Flowchart: the manual mode routine

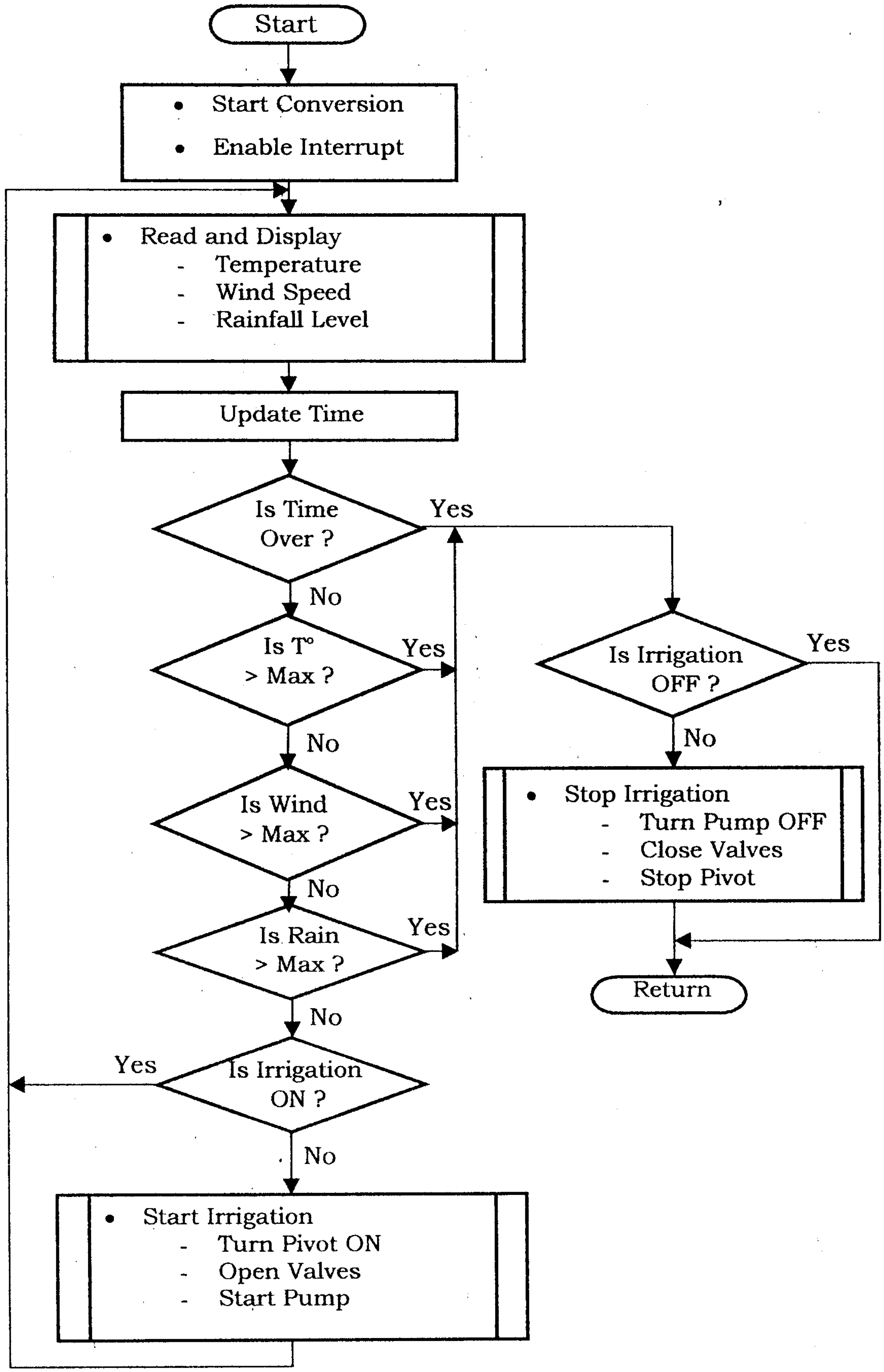


Figure 5.16 Flowchart: the timer mode routine

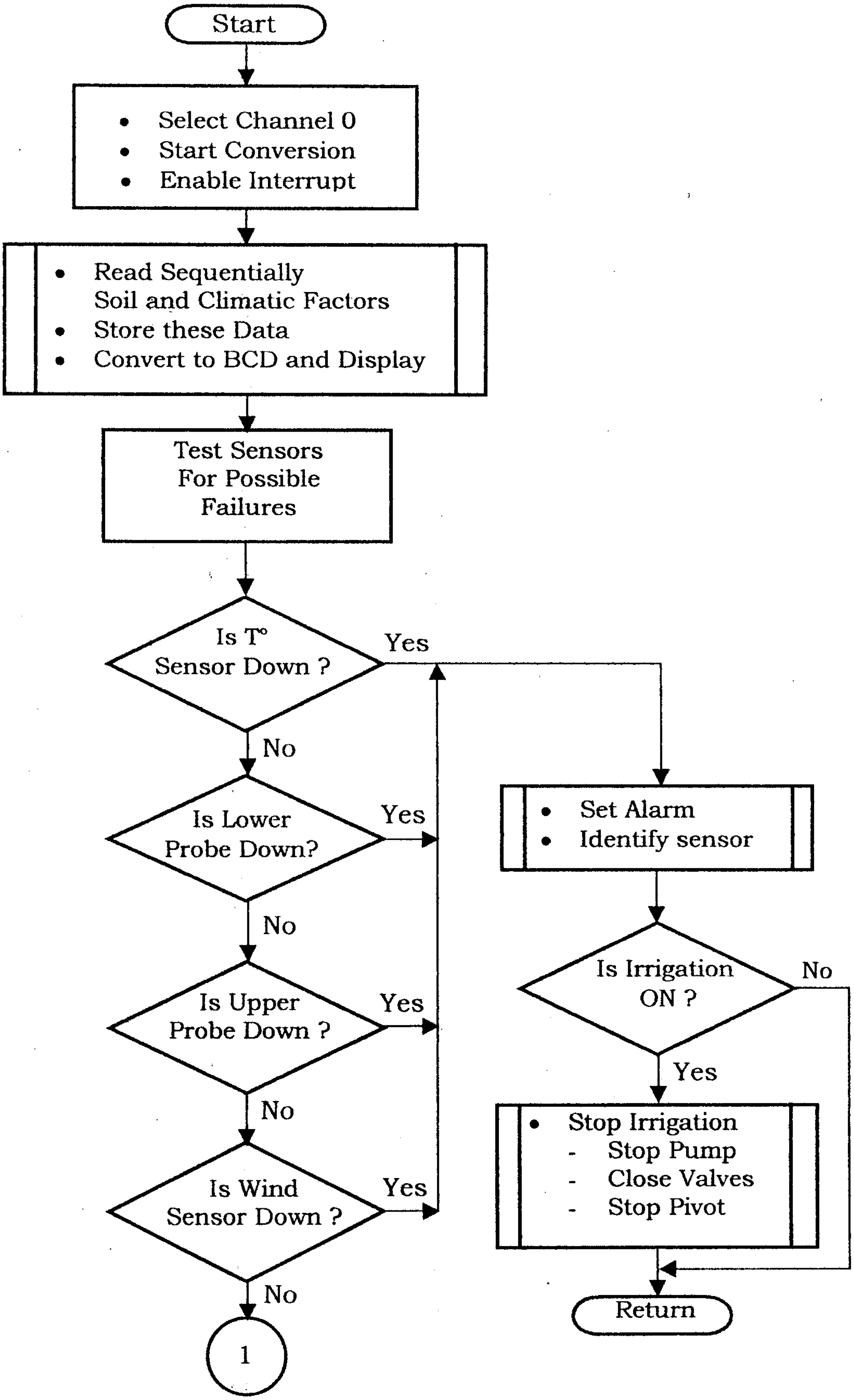


Figure 5.17-a Flowchart: the automatic mode routine

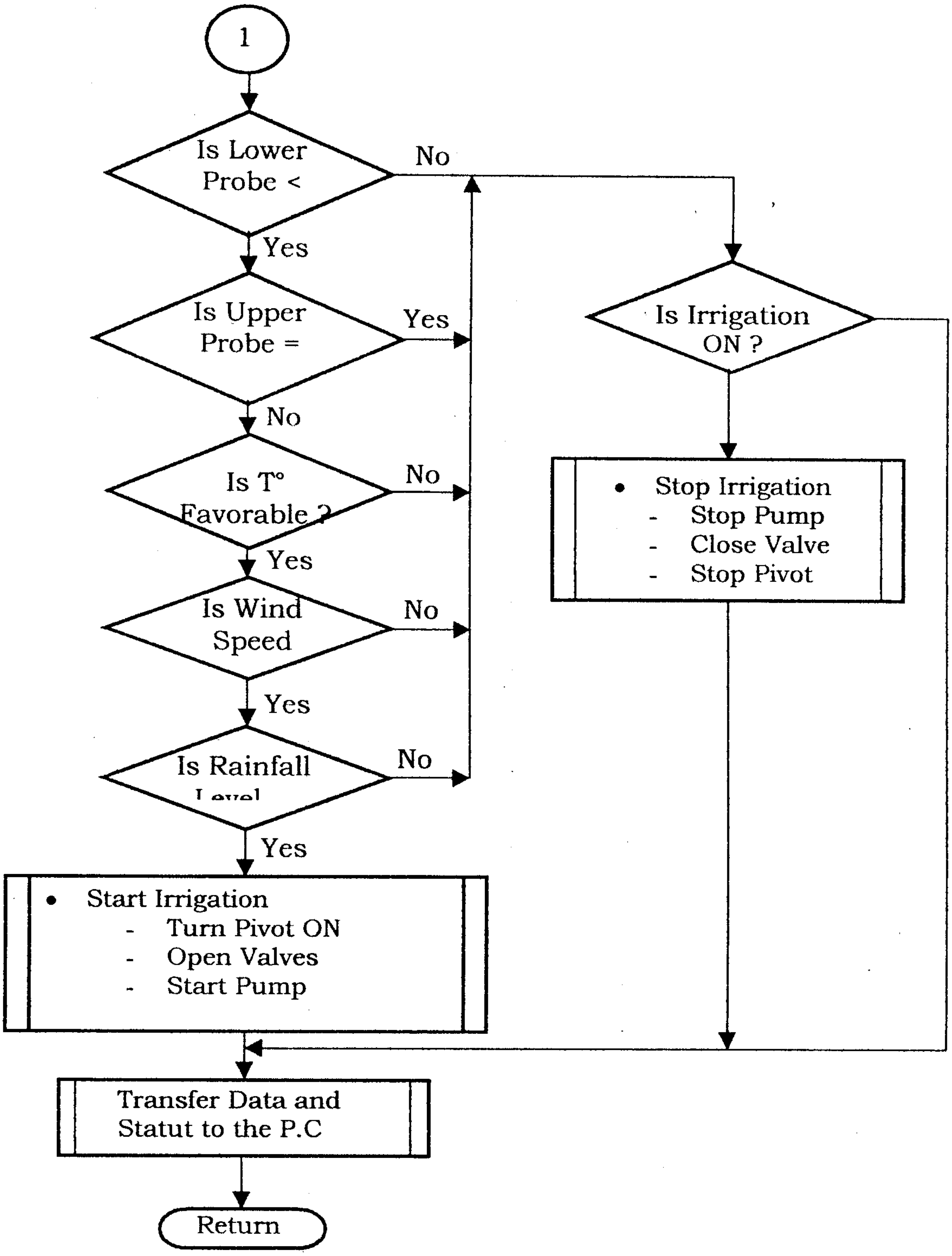


Figure 5.17-b Flowchart: the automatic mode routine

5.12 The Windows Application Program

The Borland Delphi is a sophisticated windows programming environment. It is the preferred integrated development environment (IDE) for many professional programmers [12]. Developing applications for a window 9x environment, Delphi is on of the best choices. It offers a serious object oriented environment and encourages more structured approach to application writing than its rivals do. Its IDE allows for rapid development of a Graphical User Interface (GUI) application interface. These are some of the reasons behind the choice of using Delphi IDE. The window application at design time is illustrated in figure 5.18.

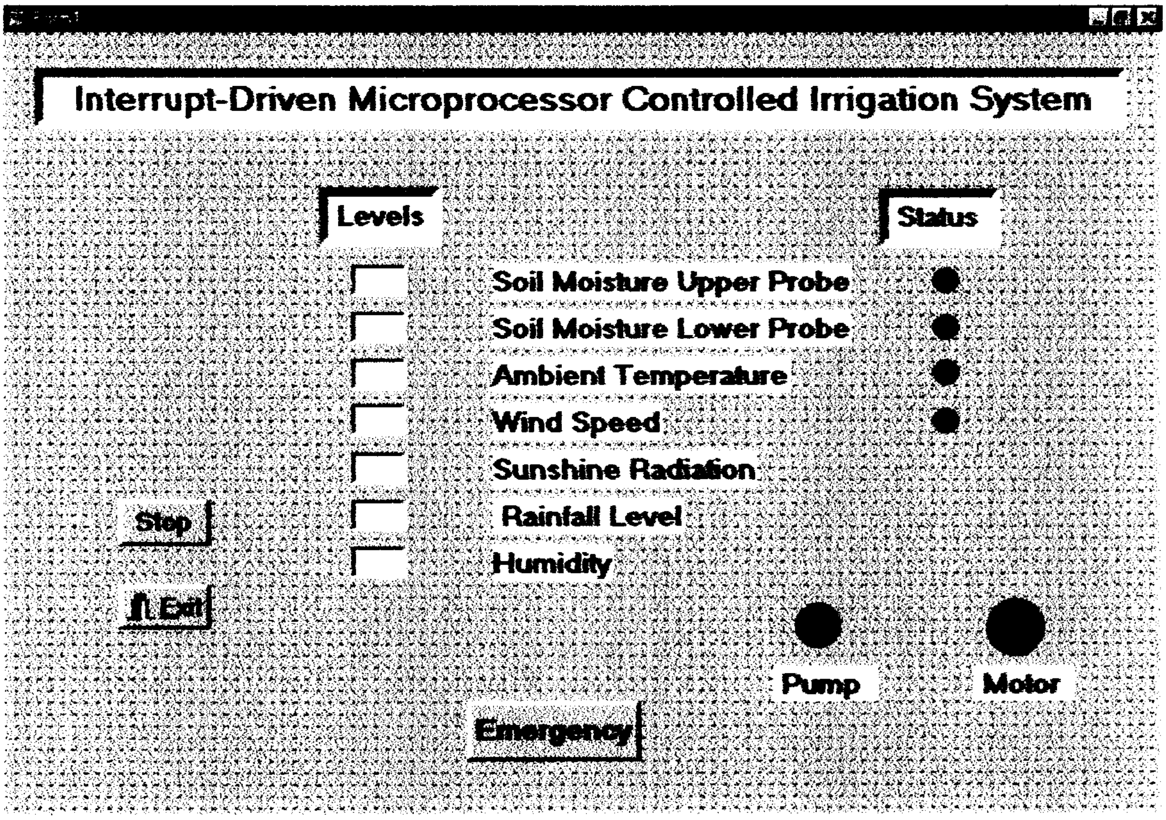


Figure 5.18 The window application at design time

The software program that runs the window application is developed to continuously exchange information with the IDMCIS controller through the serial port. It allows the user to view, in almost real-time, the status of the irrigation system and the possibility to stop the irrigation process by a simple click on the mouse at any instant of time. The flowchart describing the window application program is shown in figure 5.19.

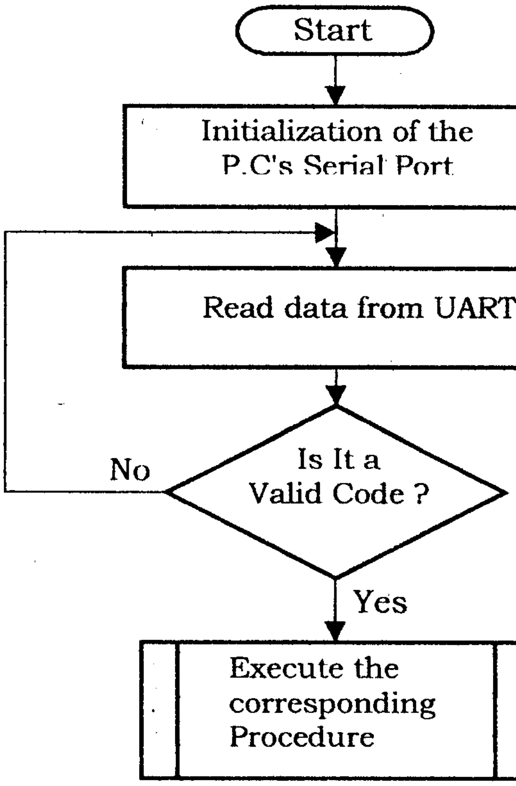


Figure 5.19 Flowchart: the window application

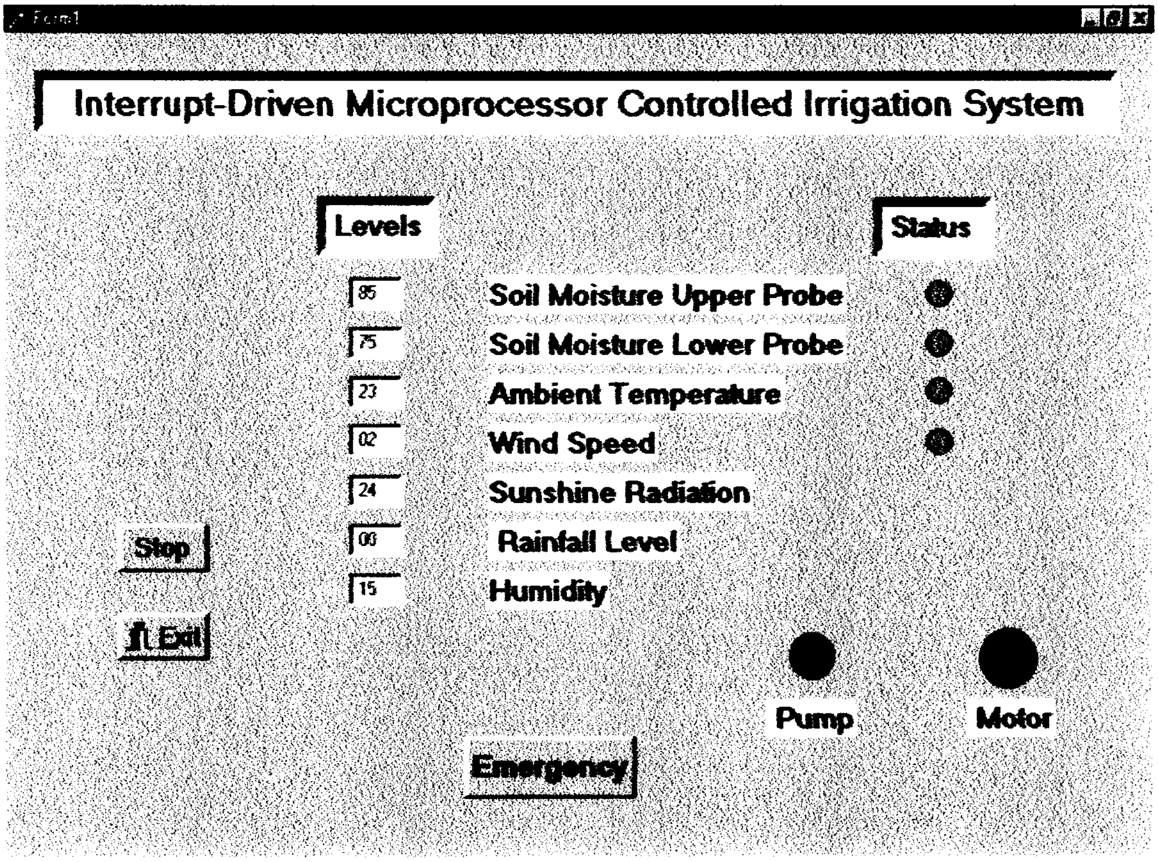
Figure 5.20 to figure 5.24 illustrate the GUI at run time for different irrigation system situations. Data is displayed numerically, however, for system status, a green color means that the corresponding actuator is ON, while the red color means it is OFF. For sensor failure, green color means the corresponding sensor is operating correctly, whereas a red color means that it presents a failure.

Figure 5.20 illustrates the irrigation system at rest, because the soil moisture (lower probe reading) is larger that the moisture threshold. Figure 5.21 depicts the case where the irrigation system is active (both pump and pivot motors are ON). It is the case, because the soil moisture is smaller than the desired moisture and the weather conditions are favorable to the irrigation application.

In figure 5.22, although the soil moisture is lower than the desired moisture, still the irrigation system is interrupted. The controller is designed to have a " rain mode" which simply shuts OFF the irrigation system when a certain minimum level of rain is reached.

A similar situation is depicted in figure 5.23. The irrigation is at rest because the velocity of wind is such that an irrigation application would result in a very poor efficiency. Wind reduces irrigation efficiency because it increases evaporation (particularly on hot windy days) and affects sprinkler distribution patterns.

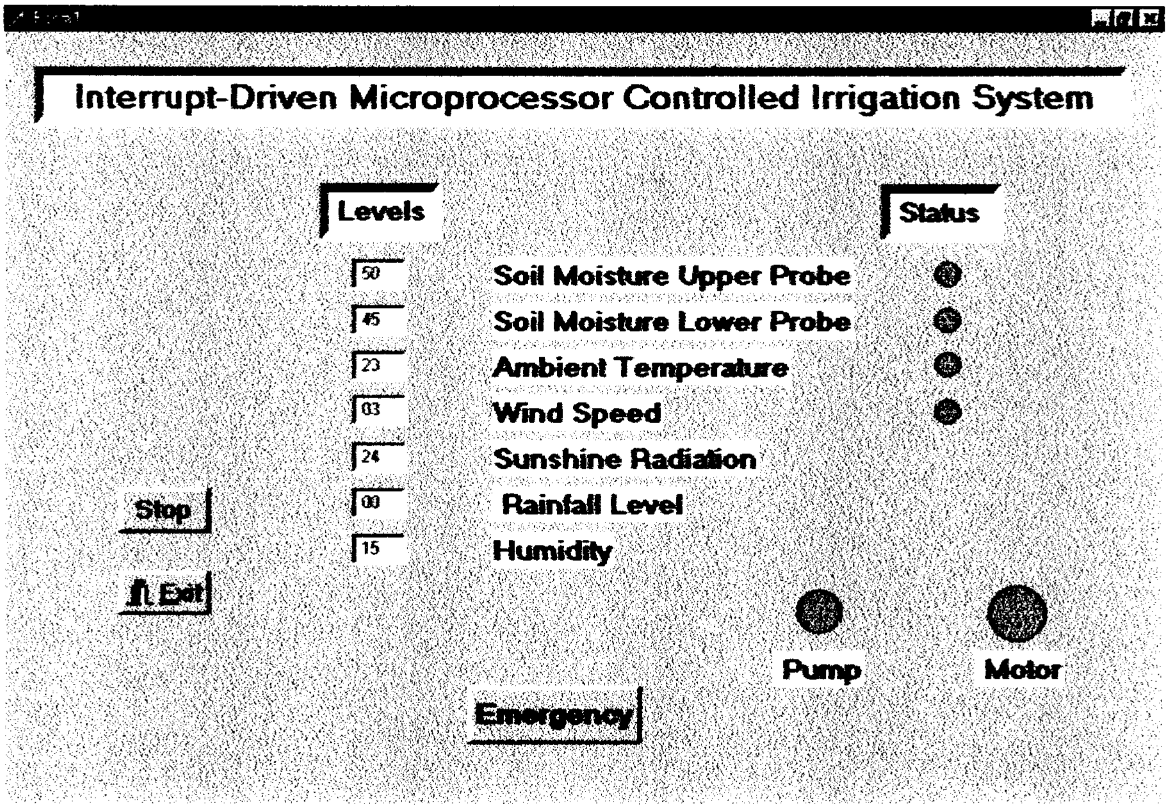
Figure 5.24 illustrates the case where we have a sensor failure in the irrigation system. The soil moisture sensed by the lower probe is null, such a soil moisture content can be obtained by an oven dried process only. Once again, to preserve water and energy resources, the irrigation process is resumed and a sound alarm is triggered figure 5.12.



Interrupt-Driven Microprocessor Controlled Irrigation System

Levels		Status
85	Soil Moisture Upper Probe	●
75	Soil Moisture Lower Probe	●
23	Ambient Temperature	●
02	Wind Speed	●
24	Sunshine Radiation	
00	Rainfall Level	
15	Humidity	

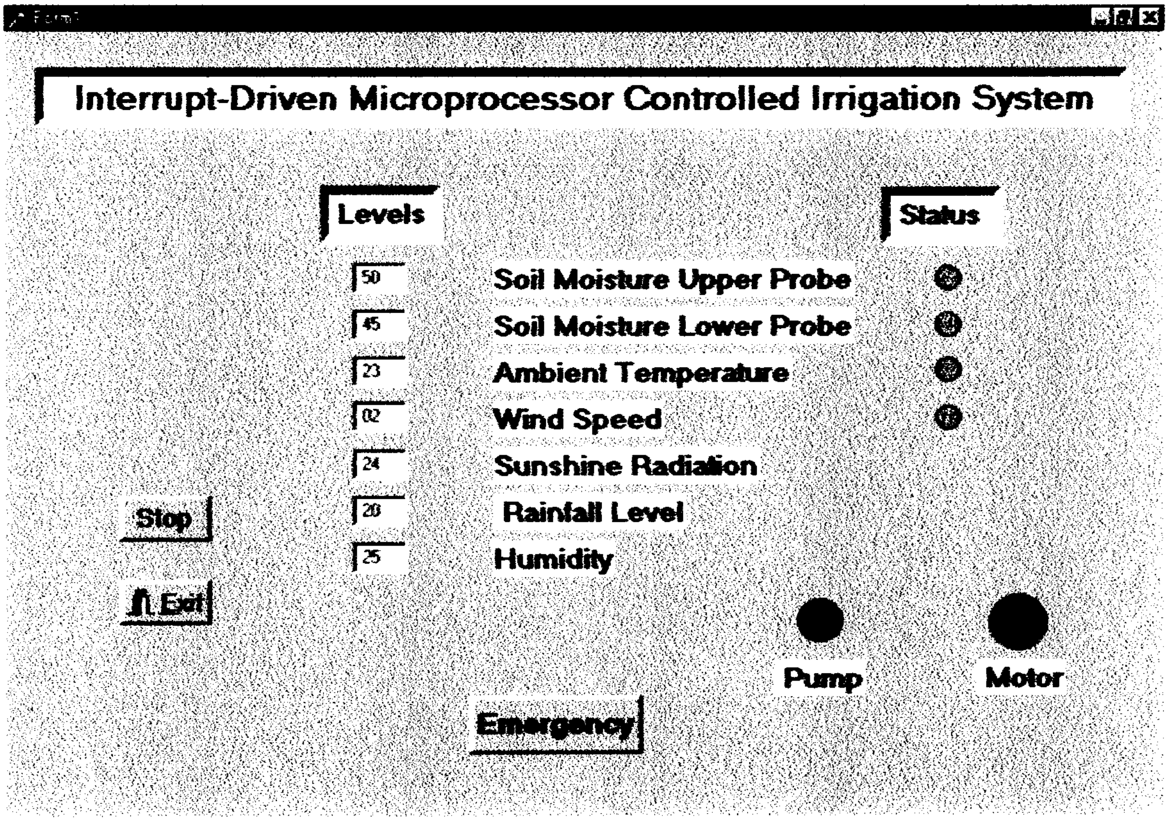
Figure 5.20 The window application at run time.
Case: Irrigation system at rest



Interrupt-Driven Microprocessor Controlled Irrigation System

Levels	
50	Soil Moisture Upper Probe
45	Soil Moisture Lower Probe
23	Ambient Temperature
03	Wind Speed
24	Sunshine Radiation
00	Rainfall Level
15	Humidity

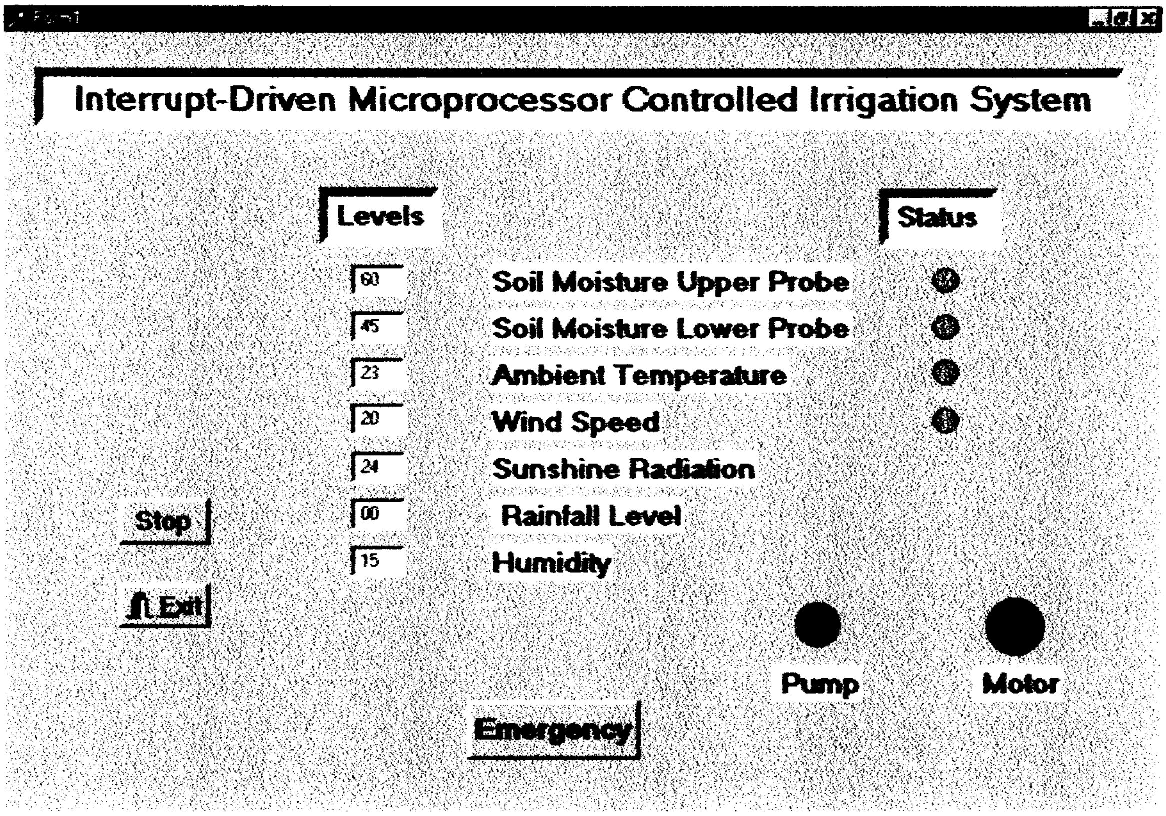
Figure 5.21 The window application at run time.
Case: Irrigation system ON.



Interrupt-Driven Microprocessor Controlled Irrigation System

Levels	
50	Soil Moisture Upper Probe
45	Soil Moisture Lower Probe
23	Ambient Temperature
02	Wind Speed
24	Sunshine Radiation
20	Rainfall Level
25	Humidity

Figure 5.22 The window application at run time. Case:
Irrigation System at rest due to rainfall level.

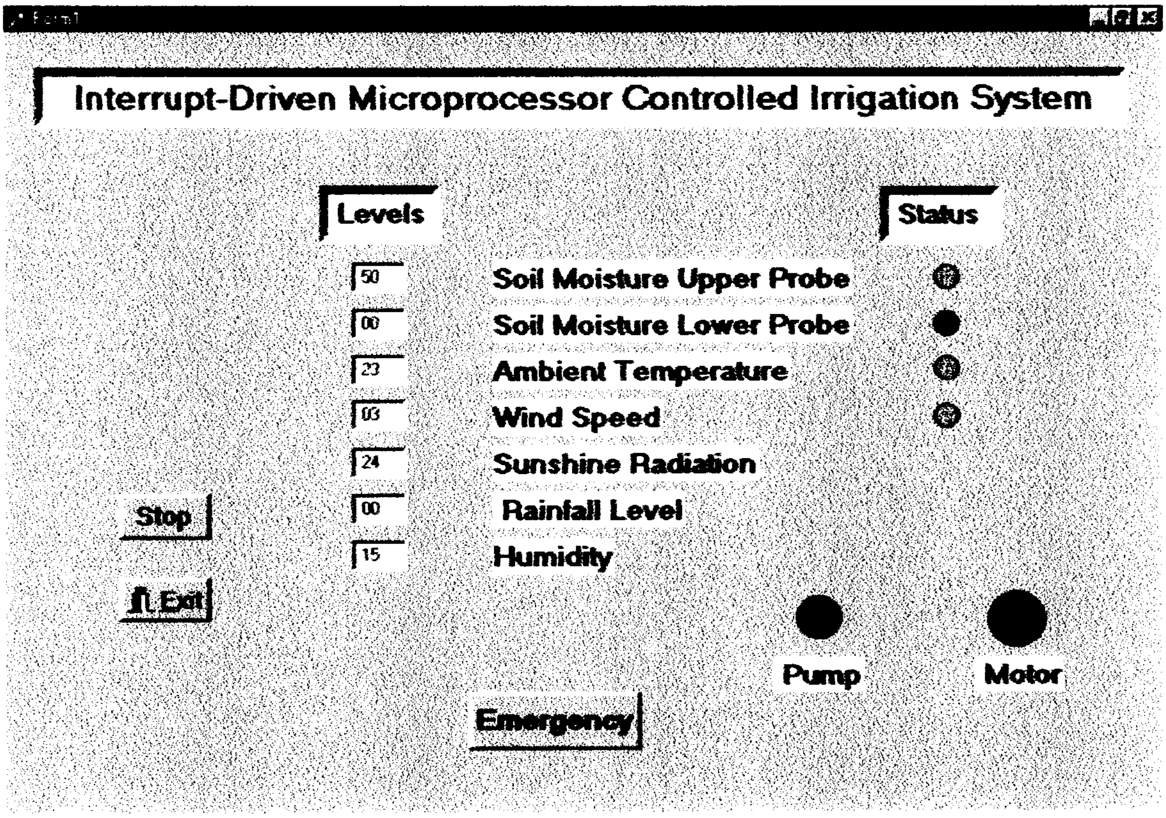


Interrupt-Driven Microprocessor Controlled Irrigation System

Levels

60 Soil Moisture Upper Probe
45 Soil Moisture Lower Probe
23 Ambient Temperature
20 Wind Speed
24 Sunshine Radiation
00 Rainfall Level
15 Humidity

Figure 5.23 The window application at run time. Case: Irrigation system at rest due to wind speed.



Interrupt-Driven Microprocessor Controlled Irrigation System

Levels

50 Soil Moisture Upper Probe
00 Soil Moisture Lower Probe
23 Ambient Temperature
03 Wind Speed
24 Sunshine Radiation
00 Rainfall Level
15 Humidity

Figure 5.24 The window application at run time. Case: Irrigation system at rest due to failure in lower probe.

CHAPTER 6

Conclusion

In this report we presented the design and implementation of an automated real-time microprocessor-controlled irrigation system. The IMDCIS, a field controller electronic package is developed specifically to serve as a tool for irrigation. It takes advantages of electronic technology to analyze complicated interrelationships irrigation parameters related to crop production. It has the capability to control field irrigation in real-time based on soil water content at the root zone of the plant.

To achieve this work, it was necessary to:

i- Choose a suitable hardware
ii- Develop a field controller, friendly user interface.
iii- Develop a friendly graphic user interface.
iv- Implement a decision-making strategy that integrates soil characteristics and weather parameters to decide on when and how much water to apply.

Periodically and continuously, the field controller collects the data from weather sensors and soil moisture probes. Based on these data and the desired soil moisture preset by the grower, the controller computes when and how much water to apply. Also, the controller is designed to auto-detect and identify faulty sensors. In case where a failure is detected, the controller turns off the irrigation system, displays a warning on the user interface and triggers the alarm.

Once the data are collected and processed, the overall status of the system is transmitted serially to the P.C. A window-based software acquires these data via the serial port and displays them on the screen. Also, the field controller can be controlled from the P.C. In case of an emergency, a simple click on the mouse stops the current irrigation regardless of climatic and soil moisture conditions.

6.1 Problems Encountered

It was seldom that things did work on the first try. Many hours of troubleshooting were necessary to overcome the numerous hardware/firmware bugs. The controller was constructed on protoboards. The huge amount of connections using thinly wires caused drop voltages, which in turn caused drops in currents that did not allow the controller to operate correctly.

Although the prototype controller was constructed on protoboards using ordinary wires, the end result is a satisfactory working system that operates as expected to do. We did supply the circuit with more than one source, in order to feed all the integrated circuits with the necessary amount of voltage/current. The problem however problem can be definitely solved by constructing the controller on a printed circuit board.

6.2 Further Work Suggestions

Despite the functionality of our system, but an engineering work is never complete, there is always room for improvement and other issues that may be added. There are several areas where the design could be improved to embellish it. Some of these areas include.

- First, the incoming data from the field controller via the RS 232C and the serial port can be further processed.
  - *i.* Data transmitted to the P.C can be Recorded and stored in files. It may serve as a guide for future management decisions.
  - *ii.* Compute ET and compare it with the actual soil moisture to determine system efficiency.
  - *iii.* Control more than one field by enhancing the IDMCIS and use the P.C as the central manager.
- Second, enhance the window application program by improving the graphic user interface to help the user input the field specific

parameters such as crop type, soil type, growth stage, desired soil moisture ... from the P.C.

- Third, another issue could be to integrate the rain forecast by developing a knowledge base and inference rules based on historical rainfall database.

- Fourth, to enhance the sensor fault detection unit by employing some sophisticated fault detection methods based also on the use of knowledge base and inference rules developed by fuzzy logic

- Fifth, the cost of hardware wiring to link the controller in-situ with the central-computer (P.C) can be prohibitive. Telemetry may solve the problem because it is based on wireless technology.

- Sixth, in designing the IDMCIS, we relied mainly on soil water content in the root zone and on the atmospheric conditions. Irrigation scheduling itself as defined previously in the report is simply determining when and how much water to apply to meet a specified management objective. However, for an operational irrigation plan, many outside factors such as labor, system maintenance, energy, water availability, economics, forecasting ...should be considered for the implementation of an optimal irrigation management system.

6.3 Hardware Enhancement

The hardware can also be improved by:

i. Replace the level translators MC1488 and MC1489 by the MAX- 232, which requires a single +5 V supply.

ii. The system could be further enhanced by the use of devices such as field programmable gate array (FPGA) chips or some type of programmable logic devices (PLD) to guarantee some degree of product secrecy in the hardware implementation as well as reducing the chip count

We can say that we have made microelectronics and computing systems interact with agriculture engineering. Finally, we certify that the Hardware/Firmware/Software design described in this report is ours.

APPENDIX

Listing of the Windows Application Program

```
unit Unit1;

interface

uses

  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  StdCtrls, Buttons, ExtCtrls;

Type

  TForm1 = class(TForm)
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    BitBtn2: TBitBtn;
    Button1: TButton;
    Shape1: TShape;
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;
    Label9: TLabel;
    Label10: TLabel;
    Button2: TButton;
```

```
    Shape5: TShape;
    Shape6: TShape;
    Shape7: TShape;
    Shape8: TShape;
    Shape9: TShape;
    Shape10: TShape;
    Shape11: TShape;
    Shape12: TShape;
    Edit5: TEdit;
    Panel1: TPanel;
    Memo1: TMemo;
    Panel2: TPanel;
    Memo2: TMemo;
    Panel3: TPanel;
    Memo3: TMemo;
    procedure FormCreate(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var

  Form1: TForm1;
  x:boolean;

implementation

{$R *.DFM}
procedure TForm1.FormCreate(Sender: TObject);

begin

asm
    mov dx,03FCh                    //Initialization of the UART.
    mov al,10h
    out dx,al
    mov dx,03FBh
    mov al,80h
    out dx,al
    mov dx,03F8h
    mov al,13h
    out dx,al
```

```
    mov dx,03F9h
    mov al,00h
    out dx,al
    mov dx,03FBh
    mov al,07h
    out dx,al
    mov dx,03FCh
    mov al,00h
    out dx,al

end;

  Shape1.Visible := true;         // LED's
  Shape2.Visible := false;
  Shape3.Visible := true;
  Shape4.Visible := false;
  Shape5.Visible := true;
  Shape6.Visible := false;
  Shape7.Visible := true;
  Shape8.Visible := false;

end;

procedure TForm1.BitBtn2Click(Sender: TObject);

begin
  x:=false;
  close;          // Exit
end;

procedure TForm1.Button1Click(Sender: TObject);

begin
  x:=false;           // Stop
end;

procedure TForm1.Button2Click(Sender: TObject);

begin

  asm
    mov dx,03f8h              // Send interrupt to MPU
     mov al,04h
     out dx,al
  end;

end;
```

```
procedure TForm1.FormActivate(Sender: TObject);

var

 msg:byte;
 n : integer;

begin

 n := 67                  // Delay
 x:=true;
 while x do

   begin
   asm
    mov dx,03f8h      //
    in  al,dx
    mov msg,al
   end;

 case msg of

{********************** ***********First data ***********************************}
 200:begin
 sleep(n);
 asm
   mov dx,03f8h               // read data
   in  al,dx
   mov msg,al
 end;

 Edit2.Text:=inttostr(msg);
 if msg = 0 then

 begin
  Shape2.Visible := true;
  Shape1.Visible := false;
 End

 else

 begin

  Shape1.Visible := true;
  Shape2.Visible := false;
```

```
  end;
  sleep(n);
  end;

{*****************************************Second data ******************************}
  201:begin
  sleep(n);

  asm
    mov dx,03f8h
    in  al,dx
    mov msg,al
  end;

  Edit3.Text:=inttostr(msg);
  if msg = 0 then

  begin
   Shape4.Visible := true;
   Shape3.Visible := false;
  End

  Else

  begin

   Shape3.Visible := true;
   Shape4.Visible := false;

  end;
  sleep(n);
end;

{********************** **********Third data ************************************}
  202:begin                   // Positive temperature.
  sleep(n);

  asm
    mov dx,03f8h
    in  al,dx
    mov msg,al
  end;

  Edit4.Text:=('+') + inttostr(msg);
  if msg >= 75 then

  begin
   Shape6.Visible := true;
   Shape5.Visible := false;
```

```
End
Else

begin

  Shape5.Visible := true;
  Shape6.Visible := false;
end;
sleep(n);

end;

{*********************** **********Fourth data *********************************}
203:begin
sleep(n);

asm
  mov dx,03f8h
  in  al,dx
  mov msg,al
end;

Edit5.Text:=inttostr(msg);
if msg = 0 then

begin

  Shape8.Visible := true;
  Shape7.Visible := false;

End

Else

begin

  Shape7.Visible := true;
  Shape8.Visible := false;

end;
sleep(n);

end;

{********************** *********Fifth data ************************************}
204:begin
sleep(n);

asm
```

```
    mov dx,03f8h
    in  al,dx
    mov msg,al
  end;

  Edit6.Text:=inttostr(msg);
  sleep(n);
end;

{************************************ Sixth data
*********************************}

  205:begin
  sleep(n);

  asm
    mov dx,03f8h
    in  al,dx
    mov msg,al
  end;

  Edit7.Text:=inttostr(msg);
  sleep(n);
 end;


{********************************** Seventh data ******************************}
  206:begin
  sleep(n);

  asm
    mov dx,03f8h
    in  al,dx
    mov msg,al
  end;

  Edit8.Text:=inttostr(msg);
  sleep(n);
 end;


{********************** *********Eigth data **********************************}

  207:begin              // Negative temperature.
  sleep(n);

  asm
    mov dx,03f8h
    in  al,dx
```

```
   mov msg,al
 end;

 Edit4.Text:=('-') + inttostr(msg);
 if msg <= 20 then

 begin

   Shape6.Visible := true;
   Shape5.Visible := false;

 End

 Else

 begin

   Shape5.Visible := true;
   Shape6.Visible := false;

 end;
 sleep(n);

end;


{*********************** ***********Pump status ******************************}

 240:begin
 sleep(2*n);

 asm
   mov dx,03f8h
   in  al,dx
   mov msg,al
 end;

 label9.Caption:=('Pump is ON');
 shape12.Visible:=false;
 shape11.Visible:=true;
 label10.Caption:=('Motor is ON');
 shape10.Visible:=false;
 shape9.Visible:=true;

 sleep(2*n);

end;

 241:begin
```

```
  sleep(2*n);

  asm
    mov dx,03f8h
    in  al,dx
    mov msg,al
  end;

  label9.Caption:=('Pump is OFF');
  shape11.Visible:=false;
  shape12.Visible:=true;
  label10.Caption:=('Motor is OFF');
  shape9.Visible:=false;
  shape10.Visible:=true;

  sleep(2*n);

end;

  end;

  Application.ProcessMessages;

 end;

end;

end.
```

REFERENCES

References

[1] Addiscot, T., Smith, J., and Bradbury, N. (1995). "Critical evaluation of models and their parameters". *Journal Environmental Quality.* 24,

[2] Aggarwal, P.K. (1995). "Uncertainties in crop, soil and weather inputs used in growth models implications for simulated outputs and their applications". *Agricultural Systems.* 48 [3].

[3] Allen, R.G., Pereira, L.S., Raes, D., and Martin, S. (1998). "Crop evapotranspiration: Guidelines for computing crop water requirements". FAO. Rome, Italy. http://www.fao.org/docrep/X0490E/x0490Ee00.htm#Contents

[4] Altenhofen, J. (1992). "Etgauge: A field device for simulating ET". *Irrigation Association Conference.* Nov-1992. New Orleans, Louisiane.

[5] Augier, P., Baudequin, D., and Isberie, C. (1998). "The need to improve the on-farm performance of irrigation systems to apply upgraded irrigation scheduling". http://www.fao.org/docrep/W4367E/w4367e0e.htm

[6] Austin, C., and Whitford, D. (1996). "Intelligent irrigation". http://www.cohort.com.au/aerogatn/booklet/chapter3.html

[7] Boman, B., Smith, S., and Tulos, B. (1994). "Control and automatic in citrus microirrigation systems". http://edis.ifas.ufl.edu/BODY_CH194

[8] Broner, I., and Law, R.A.P. (1991). "Evaluation of atmometer for estimating ET". *Irrigation Science,* 12

[9] Burke, J.J. (1993). "Thermal kinetic windows of plant enzymes". In: *Biotechnology for arid plants.*

[10] Burt, C.M., and Styles, S.W. (1994). "Drip and micro-irrigation for tree, vine, and row crops". *Irrigation Training and research Center.* Cal Poly, San Luis Obispo, CA.

[11] Campbell, G.S., and Mulla, D.J. (1990). "Measurement of soil water content". *Irrigation of Application Crops.* Agron. Mono. n°30, Am. Soc. Argon, Madison, WI.

[12] Cowell, J. (1998). "Essential Delphi 3 Fast". Springer.

[13] De Jager, J.M. and Kennedy, J.A. (1996). "Weather-based irrigation scheduling for various farms". In: *Irrigation Scheduling: From Theory to Practice,* Procs. ICID / FAO Rome. Italy.

[14] Doorenbos, J. and Pruitt, W. O., (1977). "Guidelines forpredicting crop water requirements. (FAO Irrigation and Drainage Paper 24). FAO Rome.

[15] Eitzinger,T., Marinkovic, D., and Hosch, J. (2002). ""Sensitivity of different evapotranspiration calculation methods in different crop-weather-models". http://www.iemss.org/iemss2002/proceedings/pdf/volume%20due/161_eitzinger.pdf

[16] Eldredge, E.P., Shock, C.C., and Stieber, T.D. (1993). "Calibration of granular matrix sensors for irrigation management". *Agron.* J. 85 (6).

[17] Evett, S.R., Howell, T.A., Schneider, A.D., Wanjura, D.F., and Upchurche, D.R. (1996). "Canopy temperature based automatic irrigation control". http://www.cprl.ars.usda.gov/96et_aic.pdf

[18] Evett, S.R., Howell, T.A., Schneider, A.D., Wanjura, D.F., and Upchurche, D.R. (2001)." Water use efficiency regulated by automatic drip irrigation control". http://www.cprl.ars.usda.gov/IA2001_WUE_Auto_DripIrr_Ctrl.pdf

[19] FAO. (1995). Algeria: "Geography and population". http://www.fao.org/docrep/W4356E/W4356e.htm

[20] FAO. (1990). "Basin irrigation". http://www.fao.org/docrep/S8684E/S8684e03.htm

[21] FAO. (1990). "Furrow irrigation". http://www.fao.org/docrep/S8684E/S8684e04.htm

[22] Gaonkar, R.S. (2001). "The Z80 microprocessor: Architecture, Interfacing, Programming, and Design". Prentice Hall.

[23] Garg, H.P. (1987). "Advanced in solar energy technology, Volume 3". Reidel Publishing, Boston, MA.

[24] Hargeaves, G.H., and Samani, Z.A. (1985). "Reference crop evapotranspiration from temperature". *Applied Engr. Agric* 1:

[25] Hess, T.M. (1996). "A microcomputer scheduling program for supplementary irrigation". *Computers and Electronics in Agriculture.* 15 : 233-243.

[26] Hess, T.M. (1996). "Evapotranspiration estimates for water balance scheduling in the UK". Cranfield University. http://www.silsol.cranfield.ac.uk/atmom.pdf

[27] Hill, R.W., and Allen, R.G. (1996). "Simple irrigation calendars: a Foundation for water management". In: *Irrigation Scheduling: From Theory to Practice,* Procs ICID / FAO. Report. Rome. Italy.

[28] Hillel, D.(1990). "Role of irrigation in agriculture systems". In: *Irrigation of Agricultural Crops.* ASA-CSSA-SSSA. Monograph, Madison, Wis.

[29] Hoffman, G.J., Howell, T.A., and Solomon, K.H. (1990). "Management of farm irrigation systems". ASAE Monograph, St. Joseph. MI.

[30] Hoffman, G.J., and Martin, D.L.. (1993). "Engineering systems to enhance irrigation performance". *Irrigation. Sci.* 14: 53-63

[31] Howell, T.A. (1996). "Irrigation scheduling research and its impact on water use". http://www.cprl.ars.usda.gov/96et_keynote.pdf

[32] Huguet, J.G., Pelloux, G and Loreudeau, J.Y. (1992). "Programmateur d'irrigation par les microvariations des diamètres de la plante". In: *Procs of the Workshop on Real-Time Sensing and Control of Automated Irrigation Systems.* ICID, 16th European Regional Conf. Budapest (Hungary). June 1992.

[33] Huygen, J., Van der Brock, B.J. and Kabat, P. (1995). "Hydra model trigger, a soil water balance and crop growth simulation system for irrigation water management purposes". *Irrigation Scheduling: From Theory to Practice,* Procs ICID / FAO. Report. Rome. Italy.

[34] Jeffrey, H.J. (1994). "Build your low cost data acquisition and display devices". Tab Books.

(35] Jensen, M.E. (1969). "Scheduling irrigations with computers". *J. Soil and Water Conserv.* 24(5).

[36] Jensen, M.E., Robb. D.C.N., and Franzoy. L.E (1970). "Scheduling irrigation using climate-crop-soil data". *J. Irrig. And Drain. Div (ASCE) 96(IR1)*

[37] Jensen, M.E., Wright. J.L., and Pratt. B.J. (1981). "Estimating soil moisture depletion from climate, crop, and soil data". *Trans.* ASAE (14(5).

[38] Jensen, M.E. (1983). "Design and operation of farm irrigation systems". ASAE. Monograph, St. Joseph. MI.

[39] Katerji, N., Francois, T, and Olivie, B. (1994). "Behaviour of maize stem diameter during drying cycles: comparison of two methods for detecting water stress". *Crop sci.* 34:

[40] Katerji, N., Itier, B. (1995). "Utilisation d'indicateurs de l'etat hydrique des plantes pour la conduite de l'irrigation". E.E.C. Contract n° 8001-CT 91-0108.

[41] Klepper, B., Browning, V.D., and Taylor, H.M. (1971). "Stem diameter in relation to plant water status". *Plant Physiol* 48:

[42] Lee, B.W. (1997). "Transpiration modeling and verification in greenhouse tomato". *Facilities and envir.* Control. 6 (4).

[43] Lee, B.W., Shin, J.H. (1998). "Optimal irrigation management system of greenhouse tomato based on stem diameter and transpiration model". http://www.jsai.or.jp/afita/afita-conf/1998/P29.pdf

[44] Leib, B.G., Matthews, G., Kroeger, M. (1999). "Monitoring irrigation system "on time" with micro-loggers". http://sis.prosser.wsu.edu/pdf/ASAE-PNW99paper.pdf

[45] Lieth, J.H. (1991). "Tensiometer-based irrigation of potted plants". http://envhort.ucdavis.edu/

[46] Markvari, T. (2000). "Solar energy". John Wiley and Son, LTD.

[47] McCann, I.R., King, B.A. and Stark, J.C. (1997). "Variable rate water and chemical application for continuous move sprinkler irrigation systems". *Applied Engineering in Agriculture.* 13, 5

[48] Ministry of Agriculture and Forestry New Zealand. (1997). "Best management guidelines for sustainable irrigated agriculture". Http://www.maf.gov.nz/mafnet/rural-nz/sustainable-resources-use/irrigation/irrigation-best-management/httoc.htm

[49] Ministry of Agriculture, Food and Fisheries. (2001). "Crop coefficients for use in irrigation scheduling". http://www.agf.gov.bc.ca/resmgmt/publist/500series/577100-5.pdf

[50] Monteith, J.L. (1965). "Evaporation and the environment". In: *The state and Movement of Water in Living Organisms, XIXth Symposium.* Soc. For Exp. Biol., Swansea, Cambridge University Press.

[51] Moriya, H., Iwao, K., and Kageyama, H. (1992). "Studies on non-destructive and continuous measurement of water contents and applications to irrigation in crop culture". Acta Hort. 304:

[52] Oki, L.R., Lieth, J.H., and Tjosvold, S.A. (1995). "Reduction of run-off in greenhouse through automated irrigation based on soil moisture tension". http://envhort.ucdavis.edu/

[53] Phene, C.J (1996). "Real-time irrigation scheduling with automated evaporation pan system". *Proc. Int. I. Conf on Evaporation and irrigation Scheduling,* San Antonio, Texas.

[54] Phene, C.J., Allee, C.J., and Pierro, J., (1989). "Measurement of soil matric potential and real-time irrigation scheduling". *Agri. Water Mgmt.*

[55] Phene, C.J., Detar, W.R, and Clark, D.A. (1992). "Real time irrigation scheduling of cotton with an automated pan evaporation system". *Applied Engineering in Agriculture* 8, 6.

[56] Phene, C.J., Howell, T.A. (1984). "Soil sensor control of high frequency irrigation systems". *Trans.* ASAE 27.

[57] Phene, C.J., Itier, B., B.R. Tanner., and Reginato, R.J. (1990). "Sensing Irrigation Needs". In: *Proc. 3rd National Irrigation Symposium.* ASAE Publication.

[58] Pogue, W.R. (1990). "Electrical resistance measurement of soil water for controlling landscape irrigation". Vision of the future. *Procs of the 3rd National Irrigation Symposium.* ASAE Phoenix, Az

[59] Priestley, C.H.B., and Taylor, R.J. (1972). "On the assessment of surface heat flux and evaporation using large-scale parameters". Mon. Weather Rev. 100:

[60] Przybyla, C. (1996). "Irrigation scheduling in large-scale sprinkler irrigation in the Wielkopolska Region of Poland". In: *Irrigation Scheduling: From Theory to Practice,* Procs ICID/FAO. Report n°8 Rome.

[61] Rogers, D.H., and Sothers, W.M. (1998). "Soil, water and plant relationships. Cooperative Extension Service". Manhattan, Kansas.

[62] Seckler,D., Barker, D.R, and Amarasinghe, U. (1999). "Water scarcity in the 21st century". *International Journal of Water resources Development.* (5(1/2)).

[63] Sharaf Eldin, A and Al-Almoud, A. (1994). "Microcomputer control of automatic irrigation scheduling based on soil water". *Proc. 5th Int. I. Conf. On computers in Agriculture.*

[64] Solomon, K.H. (1990). "Sprinklers irrigation uniformity". CATI, Cal State University, Fresno. Http://cati.csufresno.edu/cit/rese/90/9000803/index.html

[65] Singh, B., Boivin. J., Kirkpatrick, G., and Hum, B. (1995). "Automatic irrigation scheduling system (AISSUM): principles and applications". *J. Irrig. Drain. Engr.* 121(1):43-56.

[66] Stewart, D.B. (1994). "Real-time software design and analysis of reconfigurable multi-sensor based systems". PhD Dissertation, Carnegie Mellon University. Pittsburg, Pennsylvania.

[67] Testezlaf, R., Zazueta, F.S., and Yeager, T.H. (1997). "A real-time control system for greenhouse". *Applied Engineering in Agric.* 13 (3).

[68] Thompson, N., Barrie, I.A., and Ayles, M. (1981). "The meteorological office rainfall and evaporation calculation system: (MORECS)". Hydrological Moorandum n°45. Bracknell, UK.

(69] Thompson, S.J., and Armstrong, C.F. (1987). "Calibration of the Watermark model 200 soil moisture sensor". *Applied Engr. Agric.* 3(2).

[70] Thomson, S.J., Threadgill, E.D. and Stansell, J.R. (1982). "Field test of a microprocessor based center pivot irrigation controller". ASAE n°82-2535.

[71] Tollefsen, L.C., and Wahab, M.N.J. (1997). "Better research-extension-farmer interaction can improve the impact of irrigation scheduling techniques". http://www.fao.org/docrep/

[72] Topp, G.C., Davis, J.L., and Arman, A.P. (1982). "Electromagnetic determination of soil water content using TDR". *Soil. Sci. Soc Am J.46*

[73] Tocci, R.J., and Ambrosio, F.J. (2000). "Microprocessors and Microcomputers: Hardware and Software". Prentice Hall

[74] Valancogne, C. and Nasr.Z. (1989). "Measuring sap flow in the stem of small trees by a heat balance method". *Hortsci* 24 (2).

[75] Varlev, I., Dimitrov, P. and Popova, Z. (1996). "Irrigation scheduling for conjunctive use of rainfall and irrigation based on yield-water relationships". *Irrigation Scheduling: From Theory to Practice,* Procs ICID/ FAO. Report. Rome. Italy.

[76] Wanjura, D.F., Upchurche, D.R., and Mahan, J.R. (1992). "Automated irrigation based on threshold canopy temperature". *Trans.* ASAE 35:153

[77] Williams, S. (1995). "Soil texture: from sand to clay". http://www.ag.usask.ca/cofa/departments/hort/soil.html

[78] Wilson, E. (2003). "Water scarcity in the twentieth century". http://www.futureharvest.org/news

[79] Wraith, J.M. and Baker, J.M. (1991). "High-resolution measurement for root water uptake using automatic TDR". *Soil. Sci. Soc Am J.55*

[80] Zazueta, F.S., Smajtrla, A.G., and Clark, G.A. (1994). "Irrigation system controllers". http://www.edis.ifas.ufl.edu/AE07700.pdf