University of Victoria

Department of Electrical and Computer Engineering

ELEC/CENG 499A Design Project

Final Report

Title:	Internet Based Water Saving System		
Date:	July 28th 2006		
From:	Eric Yang	0331697	elyyang@uvic.ca
	Wei Zheng	0423717	weizheng@uvic.ca
	Alan Lin		allin@uvic.ca
To:	Dr. M. Sima	(project supervisor)	

Acknowledgement

We would like to thank the following for making this project possible.

Project supervisor, Dr. Sima for his assistance and advice throughout the project

UVic School of Earth and Ocean Science assistant professor, Dr. Adam Monahan for his assistance in our understanding in weather and climate

UVic Department of Mechanical Engineering assistant professor, Dr. Henning Struchtrup for his assistance in our understanding in Bernoullis' equation for calculating water pressure

UVic Music Information Sound Technology Interdisciplinary Center, PhD student, Ajay Kapur for providing us with the MPLAB ICD2 PIC programmer

Table of contents

Table of figures and equations

Summary

The purpose of this project is to create an irrigation system that offers convenience and energy conservation. The system is fully automated and utilizes water resource efficiently. The system is meant to function by itself or it can adapt to traditional systems with some modifications.

This report describes in detail on the implementation of the proposed project; discussing on how this idea can be achieved. Topics include various aspects of implementation method such as, software solutions, hardware solutions and the Penman's equation.

Apart from the engineering aspect, the report contains a marketing analysis section that focus on the practicality of the product; discussing the systems prospect.

The intended reader is someone with basic knowledge in engineering, who is familiar with basic operation of microcontrollers, circuit theory, basic internet knowledge and elementary weather knowledge.

Introduction

Traditional sprinkler systems only allow the user to setup a timer that provides irrigation at a certain period. This approach has several disadvantages resulting poor water and money saving. Here are several issues regards to traditional system.

- Traditional system operates independently of actual weather status. A classic scenario would be the sprinkler watering the garden during an unexpected summer storm subsequently over hydrating the plants.
- Traditional system requires human intervention. The user is required to determine the duration of the watering process.
- Traditional system requires sensors to improve accuracy. Sensor implementation and maintenance cost extra dollars.

As personal computers and internet usage becomes more and more popular, the inspiration is to make use of readily available resources such as weather data off weather websites and set up the watering operation accordingly.

The exact computation will be based on a model, namely the Penman equation. The model will require several parameters from both the internet and from users input based on a selected city profile.

The weather data will be acquired from weather websites such as http://www.theweathernetwork.com or http://weather.yahoo.com/. Given that most websites updates their information frequently, the system has the advantage of obtaining newest information at all times allowing it to cope with sudden weather changes. The system also has the flexibility of global operation since most websites contains a database of major cities in the world.

This idea has several advantages such as, the ability to determine the amount of water to be used along with the avoidance to setup any sensors. In addition, it is fully automated meaning it requires minimal user operation.

Project challenges

Before the actual implementation of the proposed project, the following challenges are anticipated.

- Establish communication between software and Hardware (PIC).
- Establish a protocol for data transmission between various components.
- Establish an accurate timer inside the PIC.
- Establish an accurate model that determines amount of water to be used.
- Establish the ability to download data off the internet and form a database.
- Establish the ability to work autonomously in the absence of internet.

Overall design description

This section of the report describes the entire design step by step with emphasis on the functionality for each component in a hardware aspect. Please refer to the diagram for a depiction of the entire system.

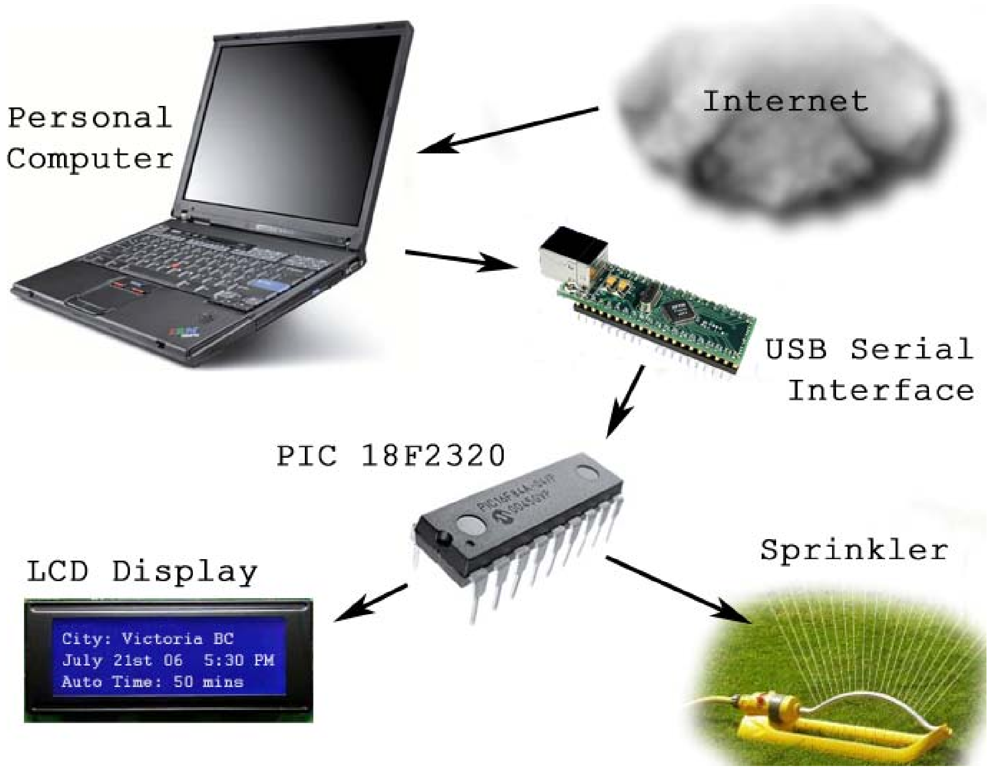


Figure 1 - Overall System Description

Initially the user downloads the weather data via the internet with the provided software program. Combining the data set by the user and the downloaded parameters, the result are being calculated using Penman's equation. The outcome is the duration of how long the sprinkler will operate.

The software program will transmit all the necessary information through the USB serial port interface. The interface converts the data into RS-232 format which is subsequently fed into the microcontroller.

Microchip Technology's PIC is a programmable RISC microcontroller that will control any hardware devices associated with the system. It keeps track of all the data sent from the PC. The idea to have a microprocessor is so that the entire system doesn't have to rely completely on PC's control. If the system were to depend on the PC at all times, it would be utterly useless shall the user decide to shutdown their computer. For simplicity sake, the microcontroller will be referred as PIC from now on.

Based on the computed outcome, the PIC will carry out the irrigation operation accordingly. The amount of water being irrigated simply depends on the duration of the watering process.

To control the water flow, a 2-position valve (also known as a digital valve) will be used. A 2 position valve is normally closed; when there is a presence of a DC voltage input, the valves open fully allowing the water to flow through. This valve can be installed in a continuously flowing water source. For this project, the signal that drives the sprinkler terminates with a LED as indication. Due to budget concerns, a 2 position valve is not implemented in the prototype.

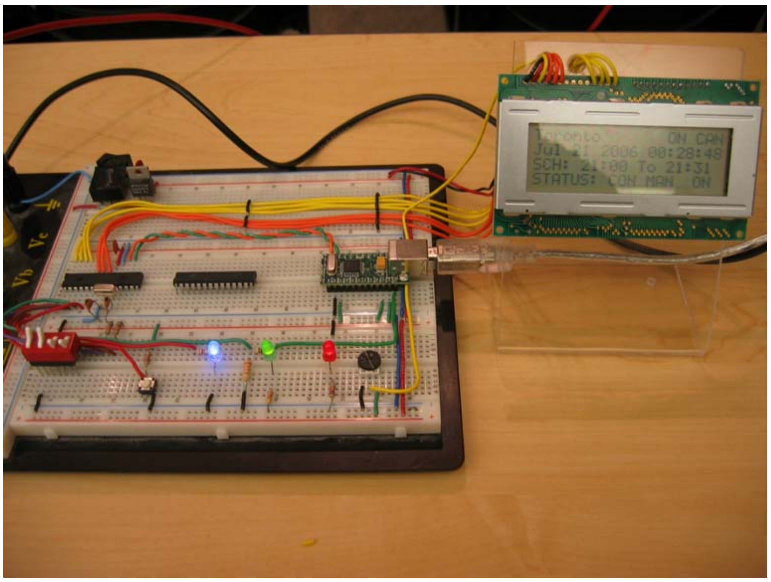


Figure 2a – Complete Prototype

Figure 2b – Complete Prototype with PC

Overall design solution

This section of the report describes the entire system design by discussing the purpose for each component; in other words, why we choose to have these components in the system.

The primary reason for a personal computer to be included in the system is because PC allows a higher level of software program with friendly user interface. It also has the plus side of sufficient memory resources and better capability of handling complicated tasks such as Penman's equation. If the decision is made to implement any complicated tasks in the microcontroller, we risk running out of memory. A second reason is that PC is a readily available gateway to the internet. It is safe to assume most people nowadays have a PC at home with internet access. The initial proposal to have a microcontroller with internet access capability is not desirable due high in cost and limitation to wired access.

The purpose of an USB to serial interface is that USB technology is widely used in most computers. USB have surpassed traditional COM ports by speed and plug-and-play feature. Most new computers do not have the traditional com ports available. Since the PIC is designed to recognize serial data, the interface will serve as a virtual serial port under windows operating system allowing the data to transfer in RS-232 format. Utilizing the FTDI chip that converts legacy peripherals to USB, DLP design's "USB232M-G USB to SERIAL UART" interface module is ideal for this application.

The idea of choosing Microchip's PIC is because its low cost and re-programmable feature. PIC utilizes EEPROM which can be erased electronically with the appropriate apparatus. PIC model 18F2320 is chosen merely because its low power features capable of serial data transfer and I/O operations. It has 8K bytes of memory which is sufficient for our task. In addition, our design team is familiar with this model plus microchip offers free samples.

The motive of a LCD display is for users to monitor the status of the PIC independently of the PC. With small in size and lower power consumption, the LCD display has the flexibility to be setup in different areas. In addition, the LCD display offers the benefit when comes to debugging the PIC during development. HITACHI LM044L LCD display is readily available from one of the team members hence it was used. It has 20 characters by 4 lines with a total of 80 characters; sufficient to display all the data required, plus Its 98 x 60 (mm) dimension is a reasonable size.

Lastly, switches and LED are implemented for monitoring purposes. Moreover, it gives the system the flexibility of selecting different modes of operation. It also has the benefit when it comes to de-bugging purposes.

Overall system description

This section of the report describes the entire design in a system aspect. UML Use Case diagrams and UML Sequence diagram are presented to depict the idea.

The entire system consists of 5 major components that are required to do the following tasks: Configuration, Obtain weather parameters, Compute water usage, display status and Irrigation. The use case diagram shows an overall association of the five major components. Configuration, Obtain weather parameters and compute water usage are achieved using PC; while display status and irrigation are achieved in the PIC.

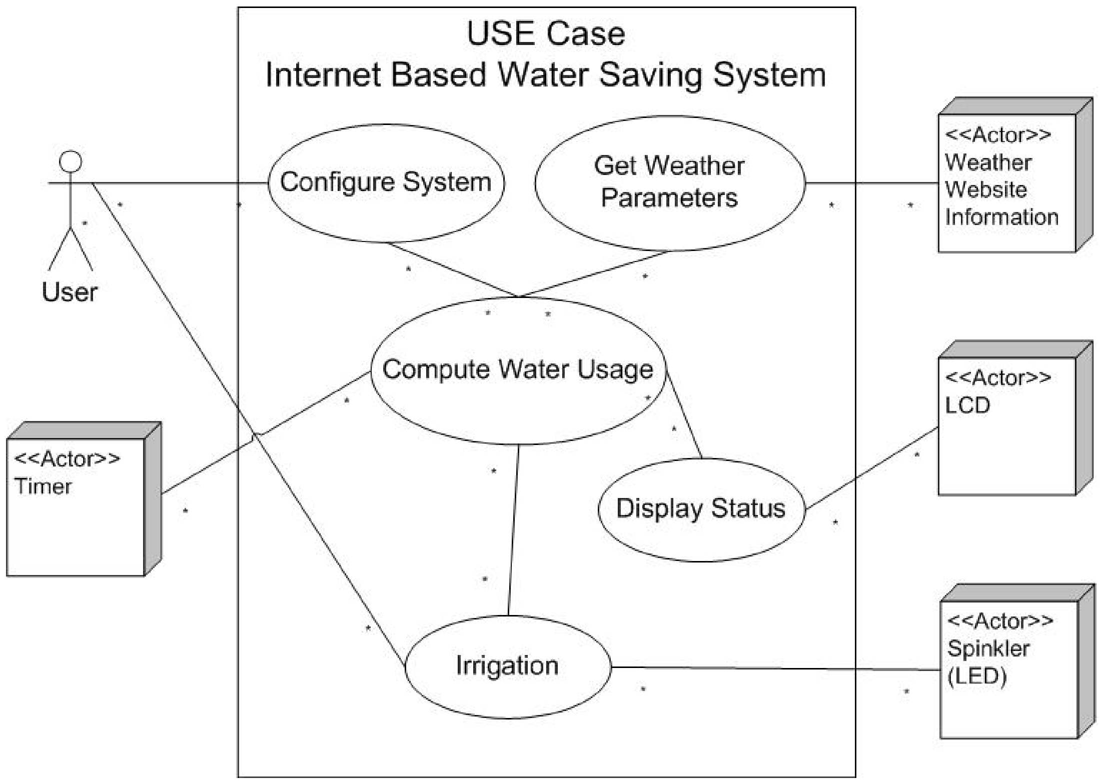


Figure 3 - Use Case

The actors associated are weather website, LCD, sprinkler and the timer.
In this system, the user has access to system configuration and the freedom to choose between manual irrigation and auto irrigation. The system will obtain weather parameters off weather websites. The computation of water usage is based on configuration, the weather parameters and the timer; the system will update a new value every time period. The system will display the current status using the LCD. Using the timer, irrigation process is carried out based on the computed values to turn on the sprinkler in which our system uses a LED as an indication.

Configure System

Configure system shows how the system is setup based on end user's preference. This task is achieved in PC using software created by Visual Basic. Using pull down menus and blank fields, the user has the freedom to setup desired start time and also miscellaneous information such as lawn size and crop type.

Overview: This use case receives user's information, user's location, crop type, pipe size, garden size and display the next schedule time for the sprinkler on the user's configure display menu.

Event: Configure System

Preconditions: None

Post conditions: Sends the user's location, crop type, pipe size, garden size to Computer Water Usage system Display the next schedule time for the sprinkler on the user configure display menu

System: Configure System

Actors: End User

Related Use Case: Compute Water Usage system

Typical Process Description:

User	System
1. End User types in location, crop type, garden size, pipe size.	
2. End User type in next schedule time for the sprinkler	
	3. The configure system sends the user's information (user's location, crop type, garden size, pipe size) to Compute Water Usage system
	4. The Compute Water Usage system sends back the next schedule time for the sprinkler to Configure system.
	5. The next schedule time for the sprinkler are display on the user configure display menu

Variation 1: The user's information are not entered, system will return 'user's information are missing' message.

Exception 1: If user types in next sprinkler schedule time then process 3-4 will not be used.

The sequence diagrams depict the configuration sequence from start to finish in chronological order.

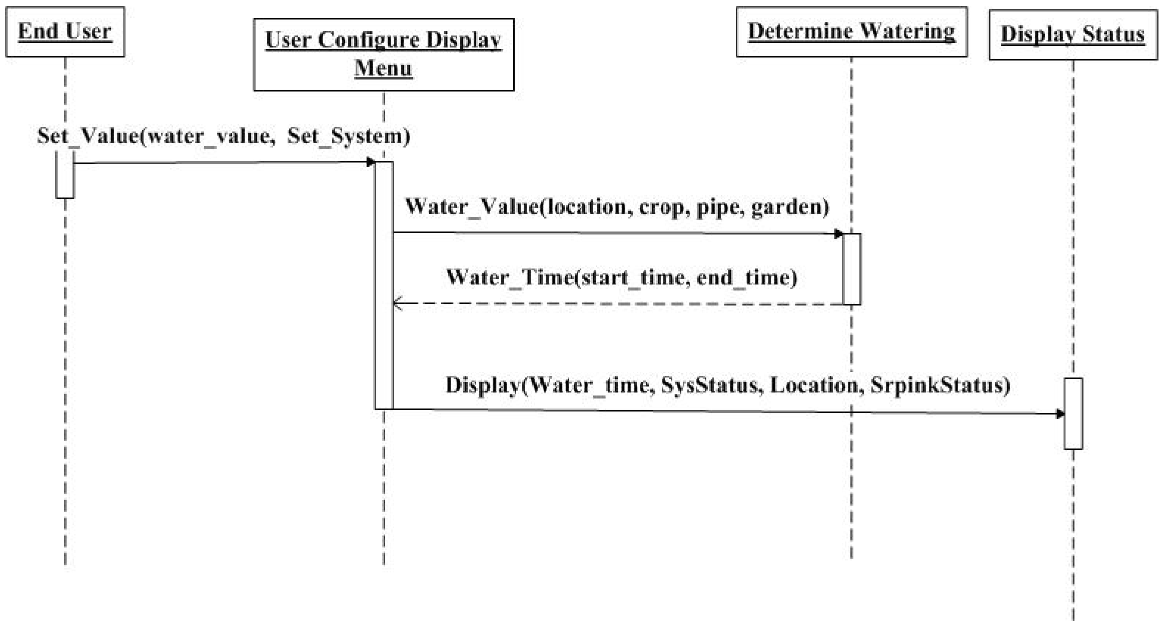


Figure 4 - Configure System Sequence

Compute Water Usage

This process is done in the PC using software. Once all the desired information is present, the software will compute the water usage using Penman's equation. For more detailed information on this process, please refer to PC Software design section. This process is being called upon every period determined by the timer. The user can setup this period through configuring systems. The computed result is displayed on LCD and carried out by irrigation process.

Overview: To determine the next schedule time for the sprinkler

Event: This use case receives a request from the timer and then determines the next schedule time for the sprinkler

Preconditions: End user has already type in all the require data in Configure System

Post conditions: Sends the next sprinkler schedule time to display status system
Sends the next sprinkler schedule time to irrigation system

System: Compute Water Usage

Actors: Timer

Related Use Case: Configure System, Get Weather Parameter, Display Status, Irrigation.

<u>Typical Process Description:</u>

User	System
1. Timer sends a request to Compute Water Usage to determine the next schedule time for the sprinkler	
	2. Compute Water Usage received a request from Timer to determine the next schedule time for the sprinkler.
	3. Bases on the end user's location the Compute Water Usage send a request to Get Weather Parameter.
	4. The Get Weather Parameter sends the weather information to Compute Water Usage.
	5. The Compute Water Usage sends the weather information to the Penman-Monteith equation.
	6. The Penman-Monteith equation sends back the evapotranspiration rate to Compute Water Usage.
	7. The Compute Water Usage checks the evapotranspiration rate and determines the next schedule time for the sprinkler
	8. The compute Water Usage sends the next schedule time to Irrigation system to turn on/off the sprinkler
	9. The compute Water Usage sends the next schedule time to Display Status system.

<u>Variation 1:</u> The weather parameter does not found from the weather website, system will return 'weather information not found' message.

<u>Exception 1:</u> None

The following diagram shows the sequence for computing water usage.

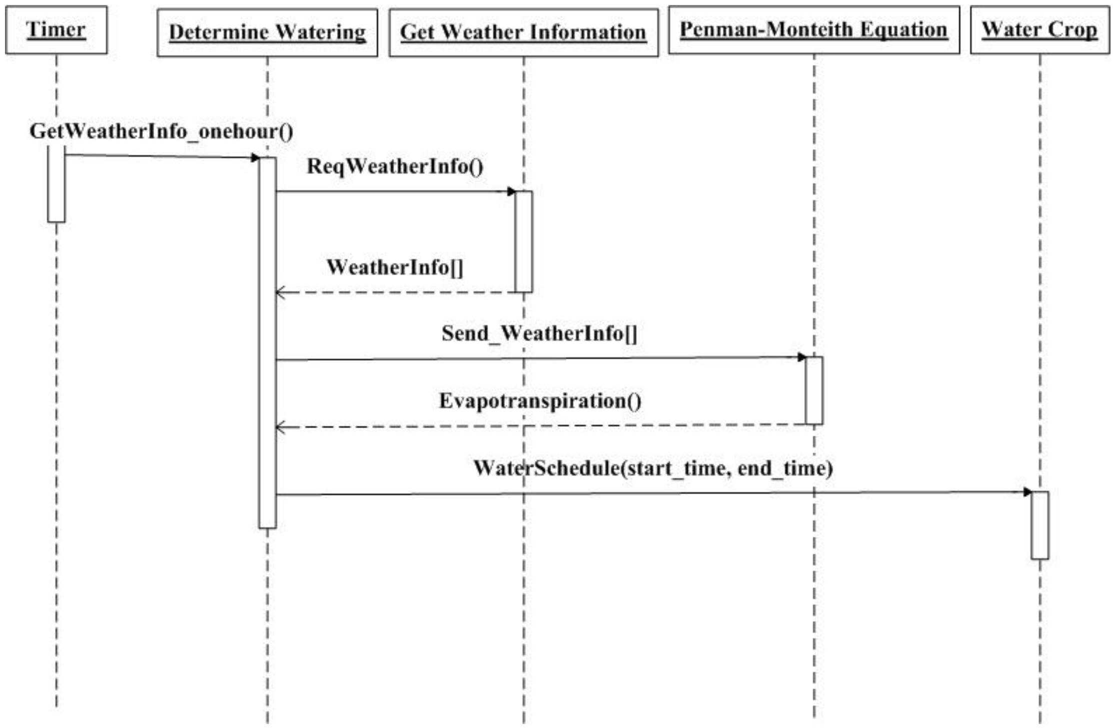


Figure 5 – Compute Water Usage Sequence

Obtaining weather parameters

This process is achieved in PC using Visual Basic.net. It has the ability to download XML data based on a given URL. The below sequence shows how the system downloads data from the weather website once the trigger is activated; the system will obtain the newest information from the weather website. The trigger is set on a timed basis or the user can manually select when to download the data.

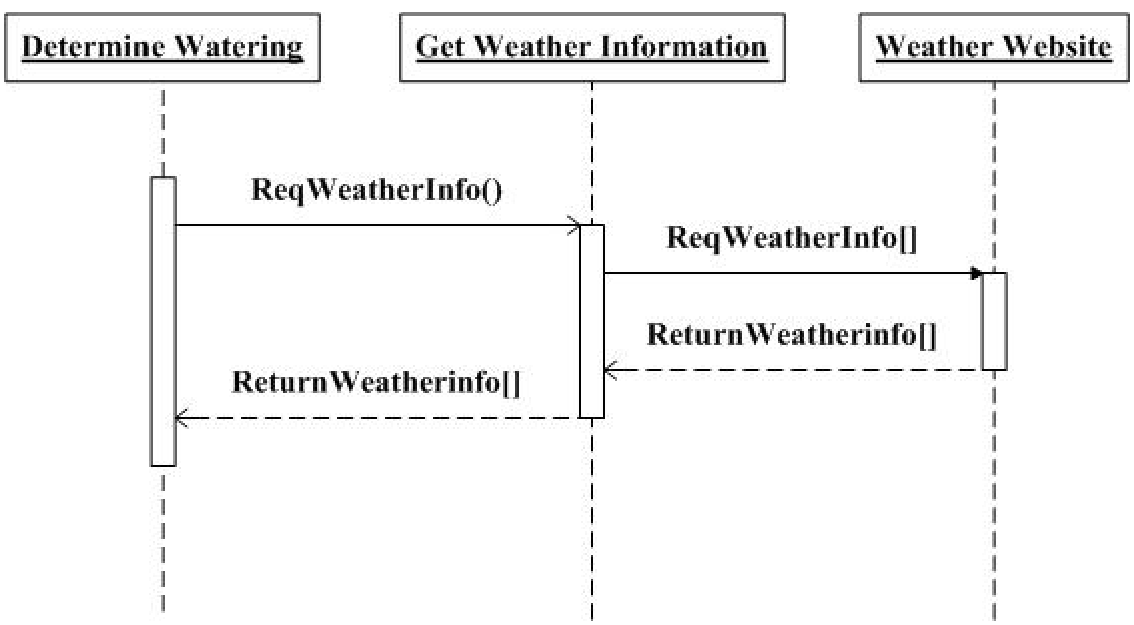


Figure 6 - Obtain Weather Parameters Sequence

Display status

This process is achieved in the PIC. Once the system is setup, the status of start time, end time and other information such as date and location are shown on the LCD. For exact details please refer to PIC software design section.

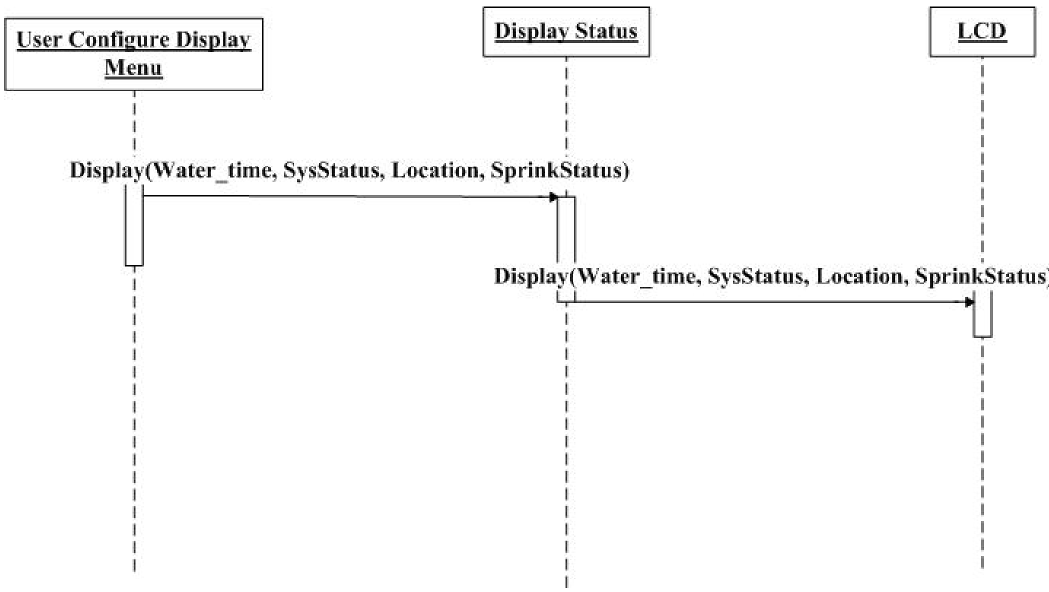


Figure 7 - Display Status Sequence

Irrigation

This process is also achieved in the PIC. The PIC will turn on the sprinkler based on the computed time or by user's manual control. Please refer to PIC software design section.

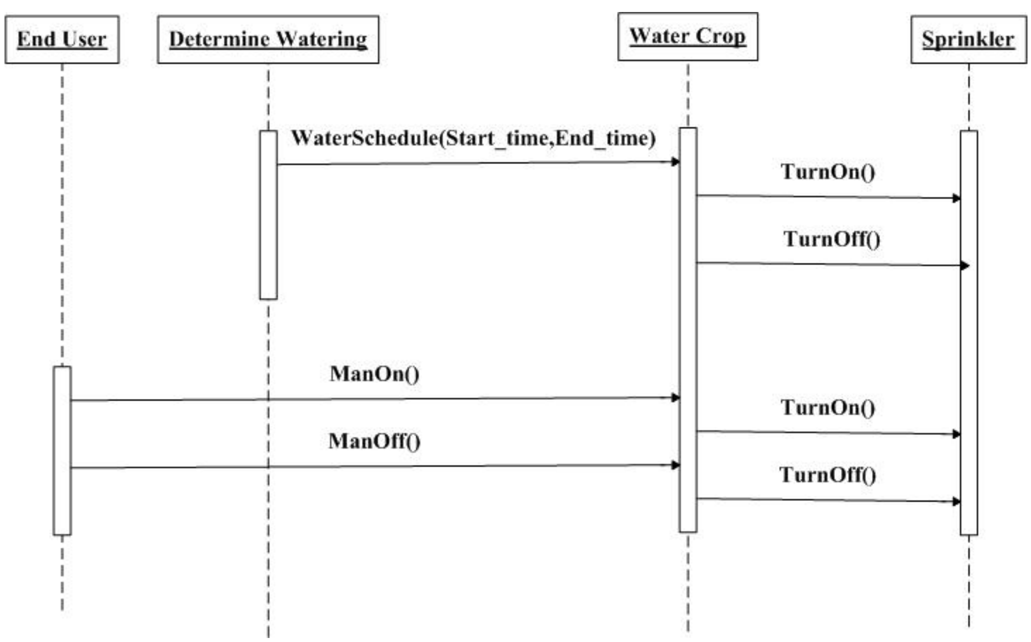


Figure 8 – Irrigation Sequence

PIC Microcontroller software design

This section of the report describes the software design embedded inside the PIC backed by a flow chart.

To program the PIC, a PIC burner is required. In this project, MPLAB's ICD2 with C compiler is used. Traditionally, PIC code is written in Assembly language. C compiler allows us to program the PIC using C language which proves to be a major advantage. Since C is a high level language, it has a rich library of built in functions. This feature makes programming easier since we don't have to worry about initializing any registers.

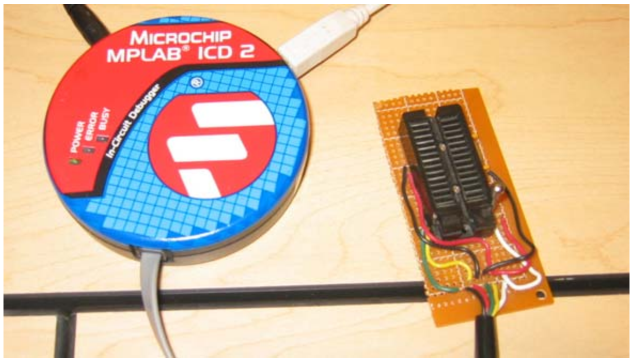


Figure 9 - MPLAB ICD2 PIC Burner

To design the program embedded inside the PIC, the first step is to list all the tasks that the PIC is required to perform. In our project, the PIC is required to attend these major tasks:

- PIC must loops forever.
- PIC will poll the status of the I/O ports, reading inputs for 'AUTO' or 'MANUAL' mode.
- PIC checks the timer at one point. If the current matches with 'start time' the PIC turns on the sprinkler. If the current time matches with 'end time' the PIC turns off the sprinkler.
- PIC checks for received data. If new data is received, it will automatically update its data base with the new value.
- PIC writes the newest value to the LCD display with the newest information.

The polling of I/O to check for switch status is relatively easy. C compiler allows individual pins be checked using input(X) function. Prior to using the I/O, PIC must initialize I/O setup by calling # use fast_io(A) with 'A' being the port name. The direction of the I/O ports can be set by calling set_tris_a().

```
#use fast_io(A)
set_tris_a( 0x07);

if (input(PIN_A1)    {
    lcd_gotoxy(18,4);
    output_high(PIN_A3);
    }
```

Figure 10 - PIC IO Operation

To setup the timer, an interrupt must be generated whenever the 16 bit system clock overflows. Each time the counter overflows at the crystal's frequency, an integer variable should be incremented. To generate 1 second, the calculation must be computed so that number of the integer incremented multiplies the time for increment is equal to 1 second. The value for the 16 bit system clock can be changed using set_timer() command, however, in this project, it was only called once to reset the value when program is being initiated. The setup_timer() function changes the clock's divisible function and determine which clock to use. Once the setup is completed, the system must enable all interrupts by calling enable_interrupts() function.

```
int counter;
#define COUNTS_PER_SECOND 154

#int_rtcc
void clock_isr()    {
    if(--counter == 0)   {
        ++secs;
        counter = COUNTS_PER_SECOND;
        }
    }
set_timer0(0);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
enable_interrupts(INT_TIMER0);
enable_interrupts(GLOBAL);
```

Figure 11 - PIC Timer Operation

To keep in track of important data for this project, the PIC must have the following database listed below. This database is updated whenever the PC sends a stream of new data through the serial port.

- Location: City, Province/State, Country
- Date: Month, Date, Year
- Current time: Hour, Minute, Second
- Start time: Hour, Minute
- End time: Hour, Minute

To check for receive data, PIC calls function if(kbhit). If there are new data in the receive buffer, the PIC will go ahead and process it.

```
#use rs232(baud = 9600, xmit = PIN_C6, rcv = PIN_C7, parity = N, bits = 8)
if(kbhit())
    gets(strRcvd);
```

Figure 12 - RS232 Operation

In terms of the received data, everything is in string format. The reason is because it is much simpler to display the string directly on to LCD. The received string vector is in the following format:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
V	a	n	c	o	u	v	e	r	%20	%20	%20	%20	%20	B	C	%20
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
C	A	N	J	U	L	%20	2	1	%20	2	0	0	6	%20	H1	H1
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
M1	M1	S1	S1	%20	H2	H2	:	M2	M2	%20	H3	H3	:	M3	M3	%20

Figure 13 - PIC Vector Format

Where the number denotes the index of the vector and %20 denotes a space.

Since the entire data is received in string format, some of the data such as current time, scheduled time and end time are to be converted into integer format. The reason is because the timer variables are all in integer format; in order to do any computation or time comparison, we employ the following method to convert ASCII value to integer value.

```
tenSec = secs/10 + 48;
oneSec = secs%10 + 48;
```

Figure 14 - PIC Char to Integer Conversion

The final task for the PIC is to display any necessary information on the LCD. The C Compiler has built in LCD display functions that make life easier.

```
lcd_init();
lcd_putc("\f");
lcd_gotoxy(3,4);
lcd_putc("Initializing...");
```

Figure 15 - LCD Operation

In this project, the data displayed on LCD includes the following:

- Location: City, Province/State, Country
- Date: Month, Date, Year
- Current time: Hour, Minute, Second
- Start time: Hour, Minute
- End time: Hour, Minute
- Mode: Auto or Manual
- USB status: Connected or Disconnected
- Sprinkler status: On or Off

The LCD display is capable of displaying 20 characters by 4 lines. Here is the LCD display format used by the system.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	V	i	c	t	o	r	i	a							B	C		C	A	N
2	J	U	L		2	1		2	0	0	6		1	1	:	2	1	:	3	7
3	S	C	H	:		1	1	:	3	5		T	O		1	3	:	0	0	
4	S	T	A	T	U	S	:		C	O	N		A	U	T	O		O	F	F

Figure 16 - PIC LCD Display Format

Figure 17 - LCD Display

Hardware design

This section of the report describes the hardware design of the system. The hardware system for this project is fairly simple; the circuitry consists of 3 major components, PIC, LCD and the USB interface.

PIC Setup – Apart from the standard Vcc and Vee for powering the PIC, it requires a crystal oscillator that functions as a clock. In our system, a 10 MHz oscillator is used with two 10pf coupling capacitors. In order for the PIC to function, pin 1 MCLR must be connected low. When MCLR is set high, the PIC resets the embedded program. In our system, we implemented a reset feature using a push button connected to MCLR.

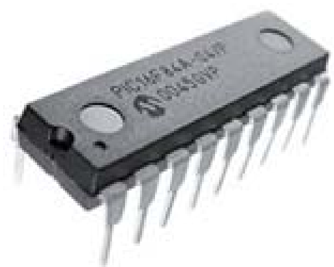


Figure 18 - Microchip PIC Microcontroller

LCD display setup – The display requires only 4 inputs for parallel data transfer into port B of the PIC. The PIC has built in function that utilizes lower nibble to control the LCD display and upper nibble to send in the data. This allow the LCD to be controlled entirely by an 8 bit port as opposed to using entire port for 8 bit parallel data transfer with a second port for controlling signal. Extra features include a contrast adjust by implementing a trim pot connected to pin Vo.

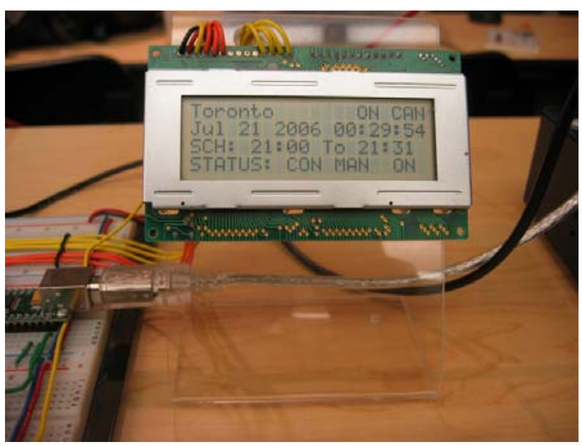


Figure 19 - Hitachi LM044L mounted

USB RS-232 interface setup – The USB interface offers two types of powering method; Internal self-power and external bus-power. In this system, external bus-power is chosen because the PIC needs to function independently of PC connection status. More importantly, if there is a short on the circuit board; power out pin of the USB will not be affected. This will prevent the USB from drawing too much current subsequently damage the PC.

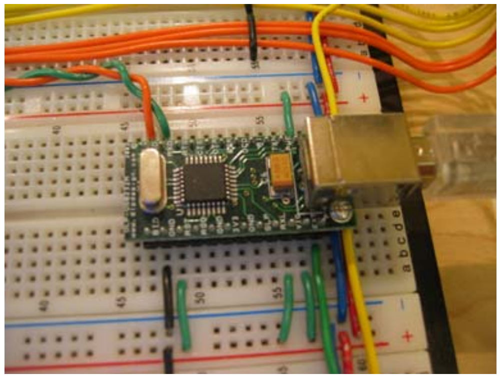


Figure 20 - DLP Design USB RS232 interface

A 5 volt regulator with diode protection is used to provide power for the entire circuit. The initiative is to mimic the product to be powered by a 9 volt battery in the future. If the user somehow mistaken the polarity, the diode will come in play saving the system.

To indicate if USB is connected to the PC, we used a LED on the USB self-power out pin subsequently connected into pin 2 of port B. Initially, if the USB is disconnected, the PIC cannot determine pin 2's logic status, therefore, a resistor must be connected in parallel to the LED.

Lastly, all switches and LED must have a current limiting resistor in series. The purpose is to limit the current that might drain inside the PIC once everything is powered off and also to protect the LED from burning out after drawing too much current.

Please refer to the attached circuit schematic attached in the appendix.

PC Software design

This section of the report describes the PC software design in detail backed by flow charts.

The PC software consists of four major tasks: 1) Obtain weather parameters from internet 2) Calculate irrigate duration 4) Communicate with microcontroller 5) Display weather information.

Obtain weather parameters from the Internet

a) Reasons to use xml to retrieve weather information

XML stands for Extensible Markup Language. It is a makeup language used to describe the data. The advantages of XML are the following:

XML describes the data instead of displaying it. By using the XML, the data can be separated from the text displayed on the web browser. The data is more organized in this fashion.

XML is a cross platform language. It will always work with different types of servers such as, My SQL Server or Microsoft SQL 2005 Server.

b) Reasons to choose Yahoo! Weather.

There are numbers of website available. Some of them do not provide data in XML format but rather in HTML format. A string search has to perform to obtain the weather data in HTML format. The disadvantage of the string search is that it is hard to implement when the data is organized in HTML table format.

Some of the websites have limited number of the cities. Take for example, the proposed www.weather.com. Our project goal is to design a system that is adaptable to in any region of the world.

Yahoo! Weather website provides the data not only in HTML format, but also in XML format, which give us the flexibility to implement our solution.

c) Implementation

Visual Basic .NET includes a name place that allows the program to obtain, parse, and manipulate XML data; it’s under System nameplace.

The XmlDocument class implies the ability to load a XML file. The class implements the DOM, which is an in-memory (cache) tree representation of a XML document, which allows the user to navigate and edit the document.

The XMLNode class represents the fundamental building block of XML documents. It is an implementation of the tree model established by the W3C Document Object Model. XMLNode supplies a series of derived classes, including the XMLDocument class.

In the screenshot, a xmlDocument object is created and the XMLNode is to retrieve the data from the XML document.

```
'define the xmlDocument Object
Dim mDoc As XmlDocument = Nothing
'Create the XML Document
mDoc = New XmlDocument()
'Load the Xml file
mDoc.Load(xmlURL)
```

Figure 21 – Load XML Data from Internet

```
Dim mNode As XmlNode = Nothing
'gets the Atmosphere data
mNode = mDoc.SelectSingleNode("/rss/channel/yweather:atmosphere ", nsmanager)
'Get the location Attribute Value
With objAtmosphere
    .humidity = mNode.Attributes.GetNamedItem("humidity").Value
    .pressure = mNode.Attributes.GetNamedItem("pressure").Value
    .rising = mNode.Attributes.GetNamedItem("rising").Value
    .visibility = mNode.Attributes.GetNamedItem("visibility").Value
End With
```

Figure 22 – Parsing XML Document

Calculate irrigation duration

Water evaporation rate is calculated using Penman equation (see Penman Equation section). The unit of the final result is mm/day and is later converted to conventional volume unit. Once the volume of water evaporate is known, the system takes in account of compensation for evaporation. The number of sprinkler, diameter of the water pipe, and the speed of the water from the sprinkler is applied to calculate the duration of the irrigation time. (see Bernoulli's equation section). The user is given a choice of setting a start time; the end time is simply the calculated duration added on top of start time.

The following formulas are used calculate irrigate duration:

$$Vol = ET_c \bullet A$$

$$Dur = \frac{Vol}{\pi(\frac{D}{2})^2 S}$$

Equation 1 – Irrigation duration

Where:
Vol: Volume of water evaporated
ETc: Evaporation Rate
A: Garden Area (square meters)
S: Water Speed (m/s)
D: Sprinkler Diameter (cm)
Dur: Irrigate Duration (mins)

The user is given the freedom to set the number of sprinkler, diameter of water pipe, and the speed of the water. When the information is saved, the value display on the setting form will be assigned to global variable and saved in registry for the further usage.

Floating point variables are used since most are in floating point and the motive is to maintain a certain degree of accuracy. Helper functions are created to convert units, for example, from Square feet to Square meters. Any constants associate with the computation is initiated globally. Variables such as garden area and sprinkler diameter are linked to the function that acquires configuration data.

Transmit data to microcontroller

From the software's perspective, it needs to communicate with the hardware via serial port. Visual Basic .NET has a SerialPort Class that deals with the serial communication. For this particular application, it is only required to transmit without having to receive; hence a handy custom class is built based on SerialPort Class to handle the serial communications. The class is named serialPIC, it has a methord named openPort(), which is used to initialize connection properties such as baud rate, stop bits, parity bit and data bits. ClosPort() is second method used the terminate the connection. Lastly, the WritePort() method is called to transmit the string data.

The string format follows the same format described in the PIC software design section. Once the "Update Information" button is clicked or timed out, the location along with current time, start time and end time will be formatted into a correct string format and eventually sends through the serial port.

Display the weather information

The weather parameters downloaded from Yahoo! Weather contains the following information: Location, Atmosphere, Astronomy, Condition, Wind, Units and Forecast. Each kind of information comes with a sub-category of detailed data. For example, Atmosphere contains humidity, pressure, visibility and rising. To organize this data structure, a class named 'Atmosphere' with properties of humidity, pressure, visibility and rising is created. Similar classes are generated by the same token. The reason to use class instead of structure is so that the program capacity can grow for the further data processing. For example, the unit of wind speed downloaded is Km/hr; Penman's equation requires m/s. If the Atmosphere is written in class, a method of transforming units can be created.

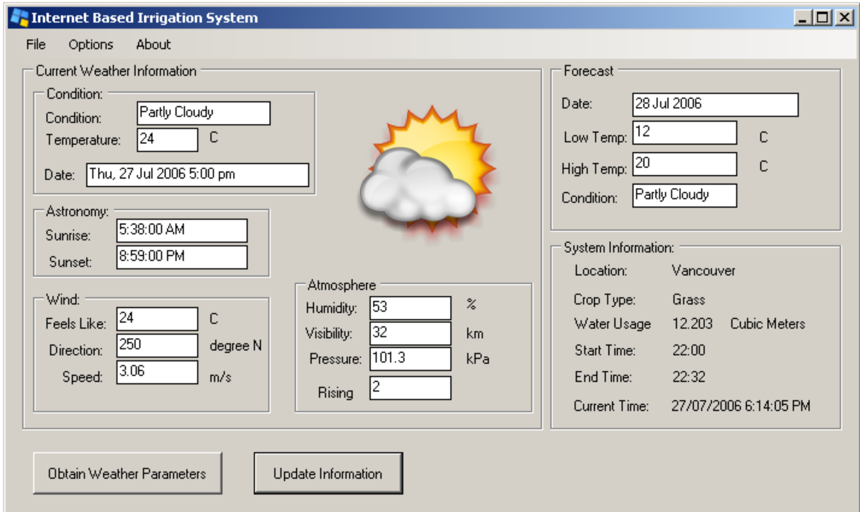


Figure 23 – Screenshots of program main form

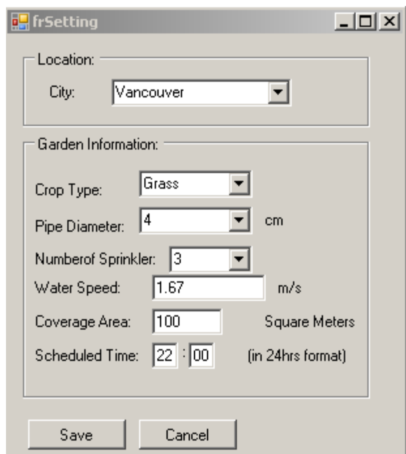


Figure 24 – Screenshots of program setting form

Penman’s equation

One of the most important features of this system is the ability to determine water usage. To achieve this, Penman’s equation is utilized with the help of Dr. Monahan, an assistant professor in school of ocean sciences and weather. This section of the report will discuss Penman’s equation and its implementation.

The Penman’s equation is used to compute the evapotranspoation rate (ET) and subsequently determine the duration time for the sprinkler. Evapotranspiration is combination of two processes, evaporation and transpiration. The entire computation is implemented in software inside the PC.

Evaporation is the process of water transformation from liquid to water vapor and removed from the evaporating surface. The Transpiration is the movement of water within the plant and the lost of water as vapor through stomata in its leaves. Equation one shows the different parameters for calculating the reference evapotranspiration.

$$ET_o = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T+273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)}$$

ET_0 reference evapotranspiration [mm day^{-1}],
R_n net radiation at the crop surface [MJ m^{-2} day^{-1}],
G soil heat flux density [MJ m^{-2} day^{-1}],
T air temperature at 2 m height [°C],
u_2 wind speed at 2 m height [m s^{-1}],
e_s saturation vapour pressure [kPa],
e_a actual vapour pressure [kPa],
e_s - e_a saturation vapour pressure deficit [kPa],
Δ slope vapour pressure curve [kPa °C^{-1}],
γ psychrometric constant [kPa °C^{-1}].

Equation 2 - Penman-Monteith equation

The Evaporation Process:

Energy is required to change the water state from liquid to vapor. The solar radiation and the temperature (Equation 1: T) provide the necessary energy for the water to transform from one state to another. The energy that is required to remove water vapor from evaporating surface is the difference between the water vapor pressure at the evaporating surface(Equation 1: es) and the surrounding atmosphere (Equation 1: ea). As the evaporation process continues the surrounding air slowly become saturated and eventually process stop once the wet air is not transferred to the atmosphere. The wind speed (Equation 1: u2) is one of the key factors to replace the saturated air with the drier air. Therefore, the solar radiation, temperature, humidity and wind speed are the essential parameters for determine the evaporation.

The evaporating surface includes two factors that effects the evaporating process, the soil surface (equation 1: Rn) which will be entered by the user and the amount of water are at the evaporating surface which will be frequent rains and irrigation.

Transpiration Process:

Transpiration is the process of water movement within the plant tissues and the vapor removal to the atmosphere. Crops lost majority of their water through the stomata, the small opening on the crop which allows the gases and water vapor pass through. There are only minority of the water are used within the crop. Although, there are many different factors to determine the transpiration rate, the temperature, radiation, humidity and the wind speed. However, the biggest factor to determine the transpiration rate is the soil water content and the ability of the soil to conduct water to the roots. (Equation 1: G)

Calculation for △ slope vapor pressure [kPa˚C¯¹]]:

The slope of the relationship between saturation vapor pressure and temperature, △ slope vapor pressure [kPa˚C¯¹]] is required for calculating the evapotranspiration

$$\Delta = \frac{4098\left[0.6108\exp\left(\frac{17.27T}{T+237.3}\right)\right]}{(T+237.3)^2}$$

Equation 3 - △ slope vapor pressure curve [kPa]

Where:
△ slope of saturation vapor pressure curve at air temperature T [kPa˚C¯¹]
T air temperature [˚C¯]

Calculation for γ psychrometric constant [kPa˚C¯¹]:

$$\gamma = \frac{c_p P}{\varepsilon \lambda} = 0.665 \times 10^{-3} P$$

Equation 4 - γ psychrometric constant [kPa˚C¯¹]

Where:
γ psychrometric constant [kPa˚C¯¹]
P atmospheric pressure [kpa]
λlatent heat of vaporization, 2.45 [MJ kg¯¹]
Cp specific heat at constant pressure, 1.013 10¯³[MJ kg¯¹ ˚C¯¹]
εRation molecular weight of water vapor/dry air = 0.622

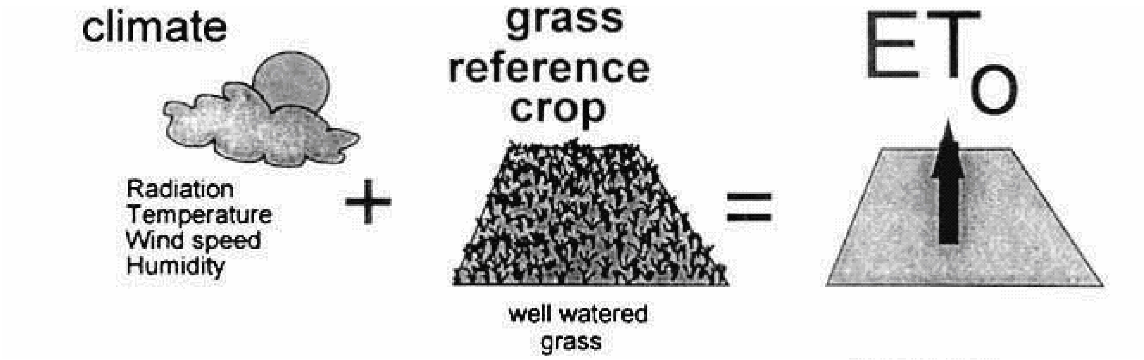


Figure 25 - Evapotranspiration Process

Figure 29 show the transportation process, where the evaporation process takes in account of temperature, humility, solar radiation and wind speed. The transpiration process takes in account of the movement of water within the plant and the lost of water as vapor through stomata in its leaves.

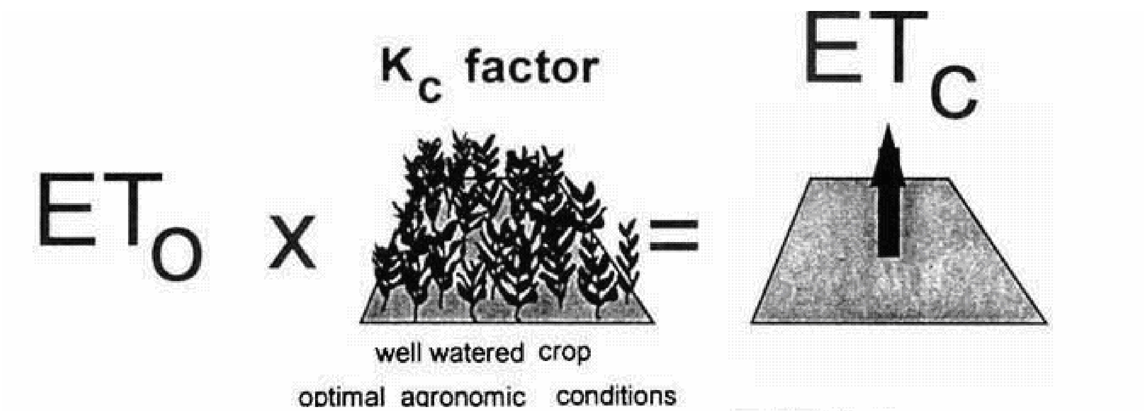


Figure 26 - Evapotranspiration process with co-factor

Figure 30 shows the Evapotranspiration process with co-factor Kc, where Kc stands for different types of the crop.

Implementation:

The calculation for evaporation rate is carried out in the PC with software using floating point variables. Small functions are created to calculate parameters such as $\triangle$ slope vapor pressure and γ psychrometric constant. A main function that represents the entire Penman-Monteith equation is created to calculate the final result. Any constants associate with the computation is initiated globally. Variables such as temperature and pressure are linked to the function that acquires weather data.

Bernoulli’s equation

This section of the report covers briefly on how basic thermodynamics, namely Bernoulli’s equation can help us to calculate water flow. This analysis is crucial when determining how much water should the system use in terms of sprinkler size, nozzle area and water pressure. With the help of Dr. Henning Struchtrup from the Mechanical Engineering department, the following analysis is deduced.

Typical household have water pressure to be about 40 ~ 50psi. It is safe to assume on average is 45 psi. The water pressure depends on the location of the house itself.

The following is Bernoulli’s equation

$$\frac{P_1}{\rho} = \frac{P_2}{\rho} + \frac{1}{2}(V_1^2 - V_2^2) \qquad A_1V_1 = A_2V_2$$

Equation 5 - Bernoulli’s Relation

Where:
P1 and P2 is pressure inside and outside the pipe respectively
ρ is water density (kg/m^{3})
V is the speed of water exiting the pipe (m/s)
A is the area of the nozzle or pipe cross section (cm^{2})

The normal pressure is 1 atm = 14.6956 psi or 1 psi = 0.06805 atm

Please refer to the following diagram for depiction

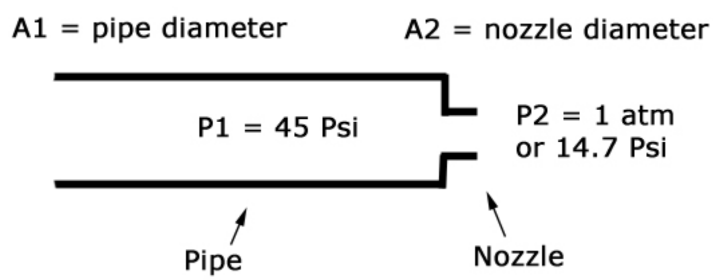


Figure 27 – Basic Thermodynamics

Using the provided equation, the system can calculate water exit speed once the user provides the nozzle area.

Water volume (m^3) = Nozzle area (m^2) x Water exit speed (m/s) x Time duration (s).

The following is a lookup tables that will give a comparison between water usage and related parameters. This look up table is implemented in the system to determine water usage based on the given parameters.

Constants			**Calculated**		
Water Pressure	45	psi	Water speed	8	m/s
Atmospjereic Pressure	1	atm			
Atmospjereic Pressure	14.7	psi			
Water Density	1	kg/m^3			

Nozzle Diameter		**Nozzle Area**	**Water Usage**
cm	m	m^2	m3 / s
1	0.01	7.85398E-05	0.000628319
1.25	0.0125	0.000122718	0.000981748
1.5	0.015	0.000176715	0.001413717
1.75	0.0175	0.000240528	0.001924226
2	0.02	0.000314159	0.002513274
2.25	0.0225	0.000397608	0.003180863
2.5	0.025	0.000490874	0.003926991
2.75	0.0275	0.000593957	0.004751659
3	0.03	0.000706858	0.005654867
3.25	0.0325	0.000829577	0.006636614
3.5	0.035	0.000962113	0.007696902
3.75	0.0375	0.001104466	0.008835729
4	0.04	0.001256637	0.010053096
4.25	0.0425	0.001418625	0.011349003
4.5	0.045	0.001590431	0.01272345
4.75	0.0475	0.001772055	0.014176437
5	0.05	0.001963495	0.015707963

Figure 28 – Water usage look up table

Marketing analysis

The main focus of the proposed project is to design a system that is for practical use. In this section of the report, practicality of the system will be briefly discussed for marketing purposes.

Let's not forget the main focus of the project is to come up with a low cost water efficient solution to replace or adapts to traditional irrigation system. Here are some of the advantages the system offer.

Since we are using weather data that is readily available online for free, the cost is lowered by preventing to install sensors in the home owner's backyard. If the sensor shall malfunction, it takes time to repair as an opposed of our system only requires a software update.

The system is flexible since the software contains a vast database for different types of crops to different regions of the world. The system can be used in most metropolitan area providing it has internet access.

The system itself is not meant to replace existing sprinklers but more of an add-on to enhance the traditional system. This will save potential investor money by keeping their old system.

Conclusion

To summarize, all the proposed solutions were implemented in this project. A functional prototype was developed by the deadline. In theory, this system should provide an energy saving alternative to the traditional sprinkler systems, however, due to time limitation, the accuracy of the result can not be verified.

Although, there isn't sufficient evidence on how the system will perform on a long term basis, for the time being, the system seems to comply with the proposed specification. After numerous test cases, it showed an indication that the system does allow more water to run on a hotter day and completely shuts off on a rainy day.

The marketing analysis stated several advantages this system offers. The future remains bright for the project shall we decide to pursuit the project to a further level.

This project is deemed successful since all the objectives are met.

Discussion and Recommendation

The primary weakness of the system is the timer function inside the PIC generated by interrupts is not accurate. The 24 hr clock will be off by few seconds per day. To compensate this weakness, the value of the timer is constantly updated by the PC as long there is connection. This weakness is not a major issue since ± 1 second won't have much effect on water conservation.

The second weakness of the design is that weather websites does not covers all regions in the world. Weather websites are typically associated with major cities and metropolitan areas thus this limits the coverage area of our system.

Overall, we haven't discovered any major problems that would hinder future development for this system. Improvements that are feasible in the future include; to establishing a wider range of database to allow more flexibility and wireless connections between USB and PC to maximize the operating range and convenience.

References

Microchip, "PIC18F2320 Data Sheet", [online datasheet], 2003 Microchip Technology Inc
Available URL: http://ww1.microchip.com/downloads/en/DeviceDoc/39599c.pdf

Datasheet Archive, "HITACHI LM440L", [online datasheet], 2006 DatasheetArchive.com
Available URL: http://www.datasheetarchive.com

DLP Design, "DLP-USB232M USB Adapter", [online datasheet], 2006 DLP Design
Available URL: http://www.dlpdesign.com/usb/usb232.shtml

Wikipedia, "PIC microcontroller", [online encyclopedia]
Available URL: http://en.wikipedia.org/wiki/PIC_microcontroller

Wikipedia, "Penman equation", [online encyclopedia]
Available URL: http://en.wikipedia.org/wiki/Penman_equation

Carl E. Anderson, "Penman Equation for Potential Evaporation", [online power point presentation] Available URL: http://www.abe.iastate.edu/AE520/Penman.ppt

FAO Corporate Document Repository, "Crop evapotranspiration – guidelines for computing crop water requirements", [online document database], FAO 1998
Available URL: http://www.fao.org/docrep/X0490E/x0490e06.htm

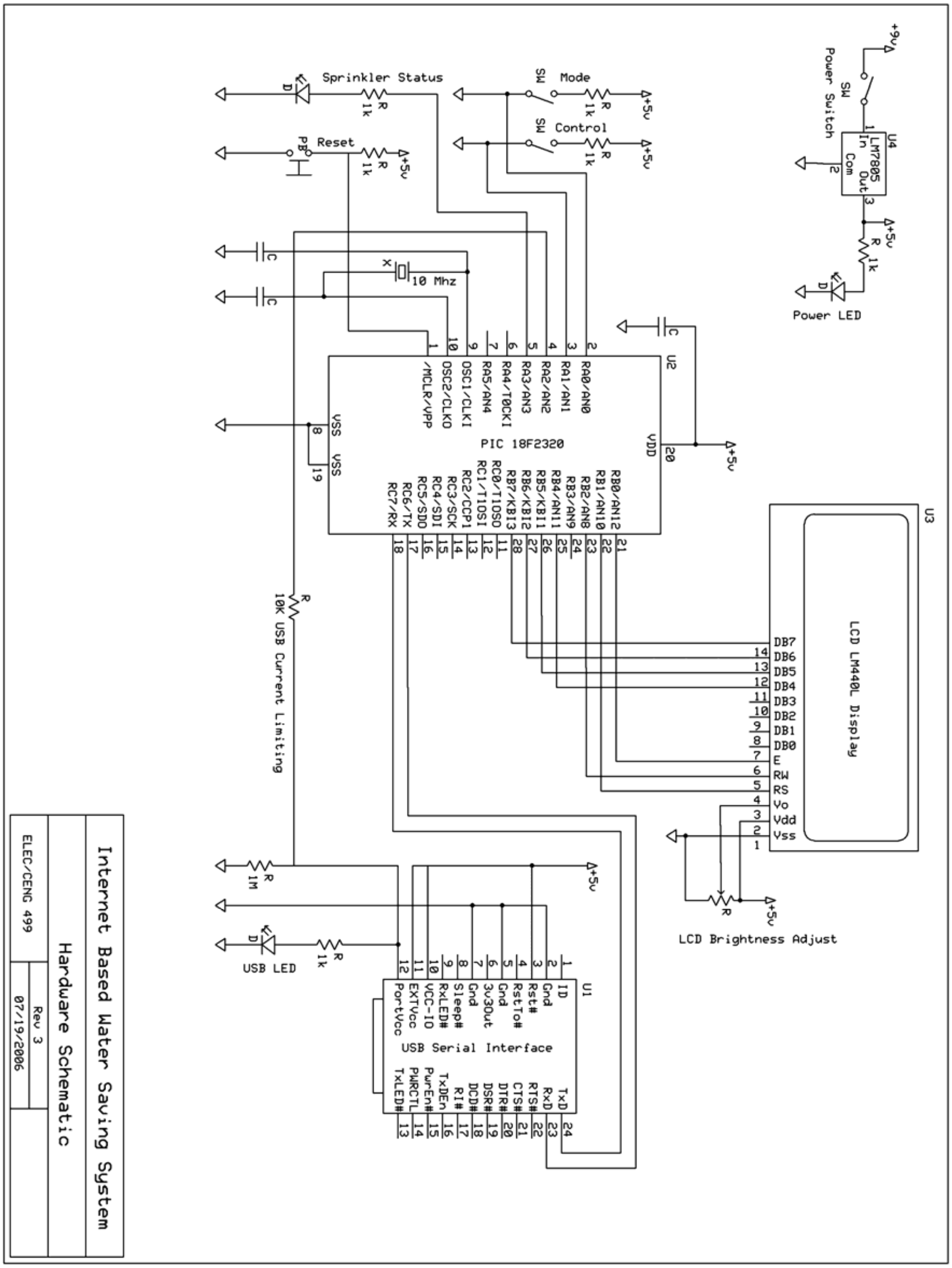
Aerodynamcs of Bicycles, "Bernoulli's Equation", [online document]
Available URL: http://www.princeton.edu/~asmits/Bicycle_web/Bernoulli.html

Dr. Adam Monahan, Assistant professor, University of Victoria School of Earth and Ocean Sciences.
Available URL: http://web.uvic.ca/~monahana/monahan.html

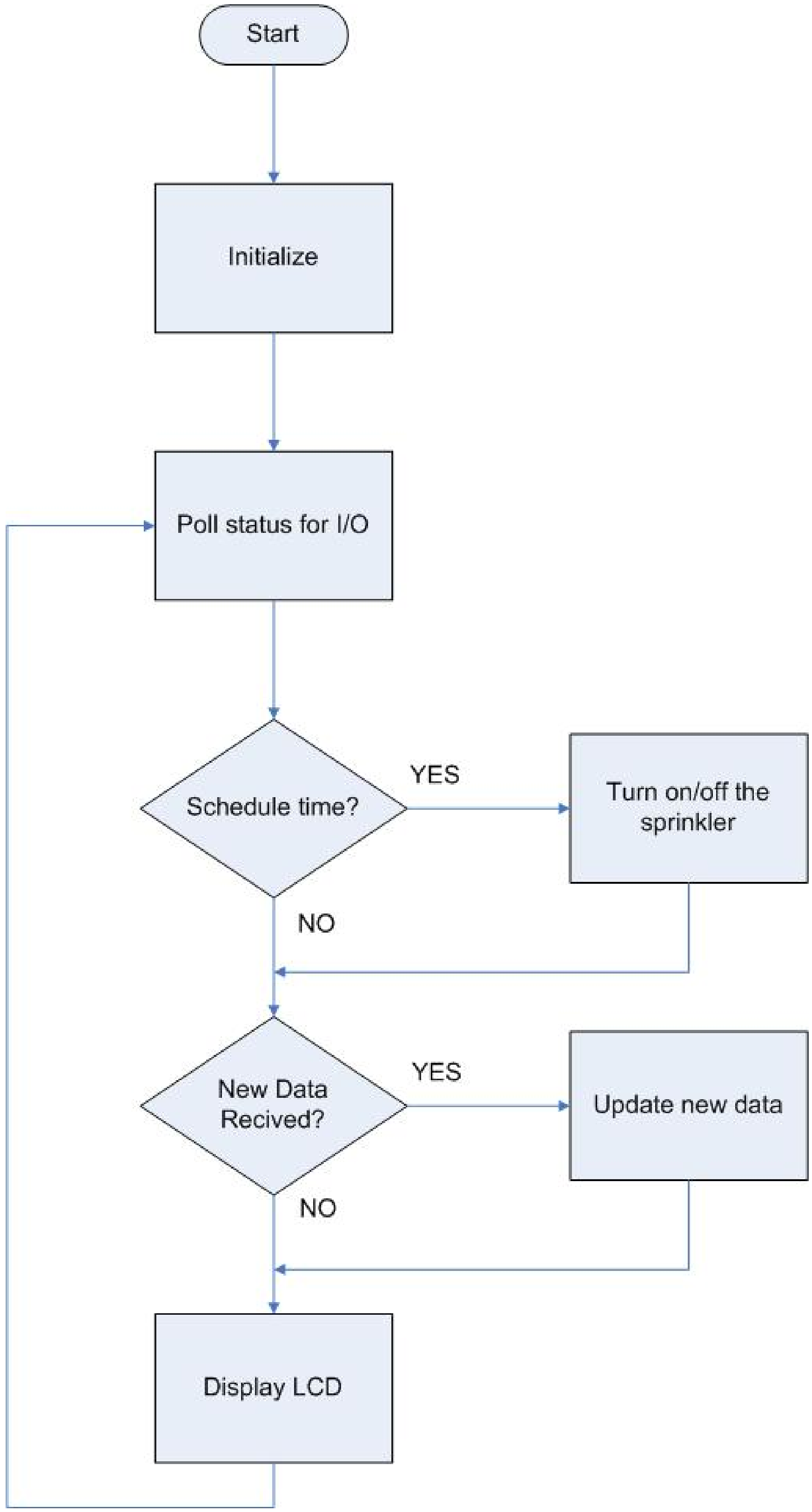
Dr. Henning Struchtrup, Assistant professor, University of Victoria Department of Mechanical Engineering.
Available URL: http://www.me.uvic.ca/Faculty/HSres.html

Appendix

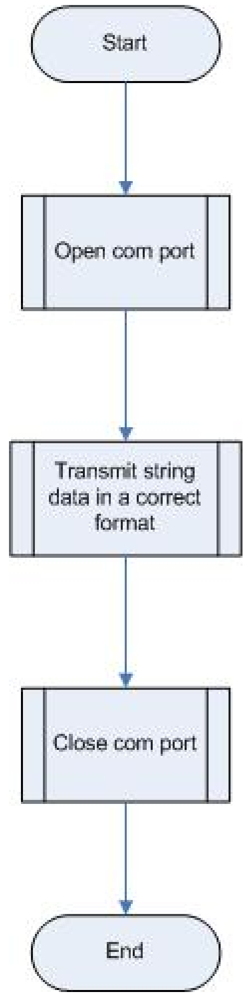
Circuit schematic



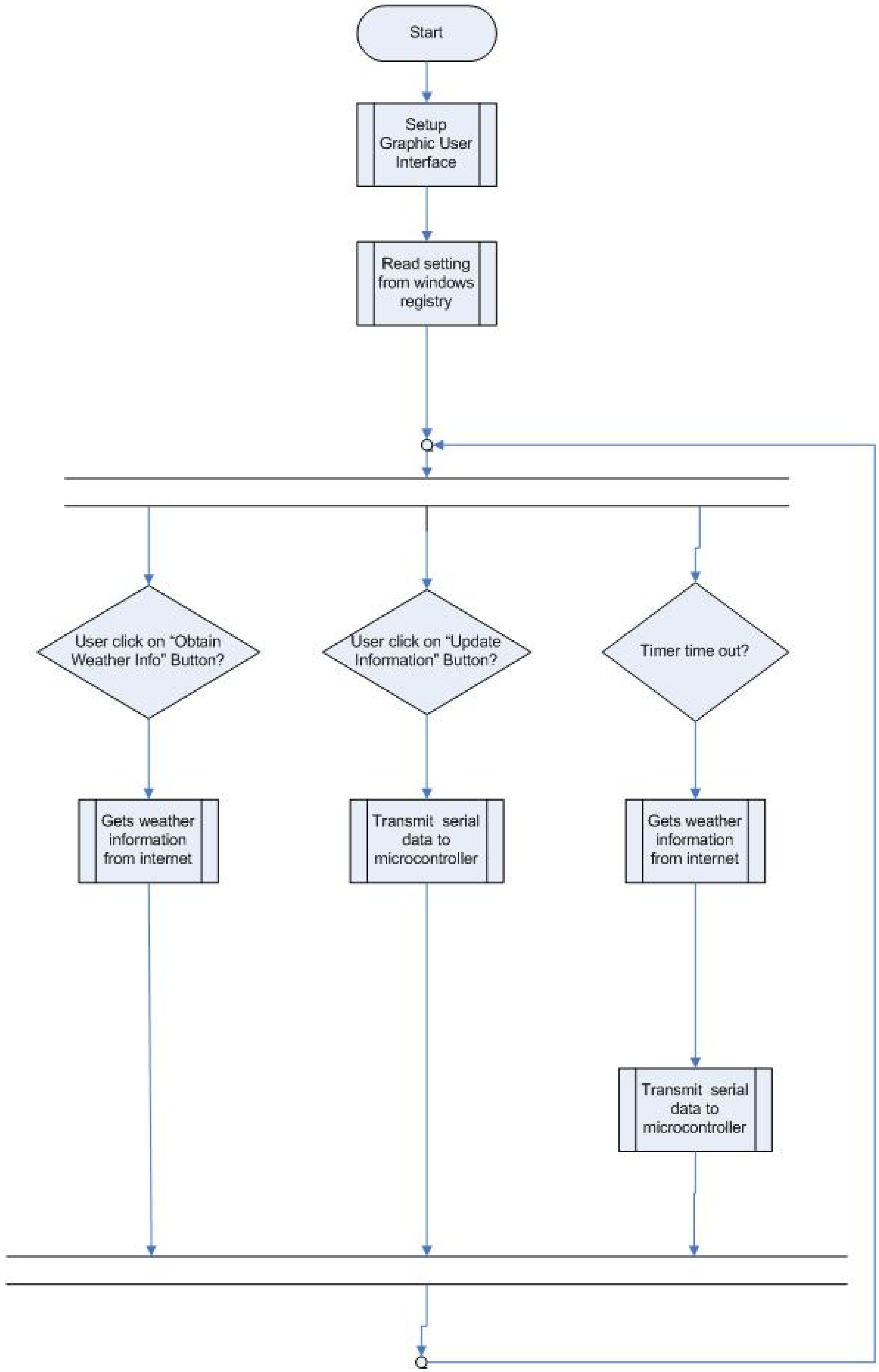
PIC Software Flowchart



PC Software flowchart for serial data transmit



PC Software flowchart for main program



PC Software flowchart for obtain weather parameters