ME5643: Mechatronics

Final Project Report

Autonomous Sprinkler System with Object Avoidance

Group Members

Jared Frank

Brian Tai

Yuri Shnirman

Prof. Vikram Kapila
Polytechnic Institute of NYU

Brooklyn, New York

May 1st, 2009 (05/01/09)

Table of Contents

Abstract

Most if not all home sprinkler systems are open-loop, mechanical-based systems that are adjusted manually to output constant water flow that does not take into consideration the current state of the system's surroundings. Without checking whether the lawn is moist already from rain, if the weather conditions are appropriate for watering, or if an object is passing the water jet or not, the sprinkler remains active. With the usual sprinkler parts employed by these products as the base and an attached hose to supply the water, the proposed sprinkler system makes use of valves to control the flow rate in order to avoid objects. When an obstruction approaches the sprinkler within its sensing range, the sprinkler will hinder the water flow in the direction of the object. The proposed product is an autonomous, weather-controlled sprinkler system utilizing an obstruction sensing and avoidance method.

A PIC16F57 microcontroller embedded in a Parallax BASIC Stamp chip will be the CPU of the product. Sensors capable of detecting weather conditions such as a soil moisture, temperature, and light will be interfaced in order for the system to decide when it is appropriate to initiate sprinkling. Along with a timer integrated circuit, this system will autonomously employ both weather and time control. A series of pushbuttons and an LCD will also be exploited with a remote controller to allow for human interaction and manual control. Upon completion, the autonomous sprinkler system with object detection and avoidance will prevent what has irritated so many of us for so long in the form of a practical, easy to afford mechatronics solution. Due to slight errors in circuitry and programming, the release of this product is delayed.

Introduction

2.1 - Sprinklers

A sprinkler is a watering device that is used to to deliver the necessary water to the soil evenly, amply, and easily or in a timely manner. It is quite obvious that it would be impractical for a human being to water a 10000 square foot area grass field every day and multiple times throughout the day. For the requirements of this particular project, an irrigation sprinkler was used. This type of sprinkler can be used for farms, golf courses, residential lawns, or any other patch of land that requires water. Other types of sprinklers include but are not limited to Fire sprinklers, Sprays, Impact sprinklers, Floppy sprinklers, Spray Pop-upd,Pulsating sprinklers, Rainguns, and Underground sprinklers. The most basic sprinkler can be just a hose connected to a water flow separator which properly separates the thick single stream of water from the hose to many thin streams that could distribute water evenly. The next step up, is a sprinkler with a rotating head. It is just a sprinkler which changes the direction of it's water flow. This can be done through a variety of methods including rotation due to water impact, or an electrical motor. The most common and practical type is rotation due to water impact. This method uses the pressure or forces due to water to hit a certain component of the sprinkler to cause it to rotate. Using an electrical motor would prove to be more accurate however the complication, cost, and power consumption that arise outweigh the pros. The sprinkler to be designed will use the features of a mechanically rotated sprinkler with variable flow rate through the use of a DC motor connected to a valve.

2.2 - Watering Factors

The water content in the soil is a very important factor when growing plants or fruits of any kind. In order for the soil, the basis for all of the resources, to do it's job it must be properly maintained with the correct amount of water. There are many factors involved when it comes to watering the soil and the plants. They include climate/region, type of plant, time, and amount/frequency. On a hot sunny day in the middle of summer, the average lawn should receive approximately 125 gallons of water per 1000 square feet. On a cloudy day with similar conditions, as little as 10 gallons per 1000 square feet can be used. This is due to the fact the heat from the sun evaporates most of the water that has been supplied. Other climate conditions that affect the watering process include humidity and strong winds. Low air humidity means there is a higher chance for evaporation, leaving less water for the soil. For example areas such as Hong Kong, where humidity is high year round, less watering is required. However on the other hand, in areas such as the American mid-west, where it is very dry year round, abundant and consistent watering is required. Strong winds can also contribute to watering factors by blowing away water that has not been fully absorbed into the soil or even the soil itself. It is not as large of a factor as heat from the sun and humidity however. If it just so happened that there was a heavy rain storm in the middle of the day, then there is no need to water the plants for that week. Doing so will over flow the soil, causing run off and possibly water related diseases.

The type of plant also affects the amount that needs to be watered. Like humans, not all plants are created equal and require a different amount of nutrients. The common lawn grass, Gramineae, require the most water since they are the most susceptible to water deficiency while a desert cactus would require less water on a day to day basis.

The type of soil will play a significant role as well. Soils are classified by their grain size which can range from sandy (large, coarse rock particles) to clay like (small particles). The general consensus is that the larger the grain size shorter and more frequent the watering must be. This is due to the fact that larger particles correspond to larger air gaps making the soil fill up and drain out water faster than usual. For smaller grain soils, the opposite must be done than that of the larger grain soils.

Another critical factor is the time and frequency of watering. The water should thoroughly soak through to about 6 to 12 inches worth of soil. During the watering process, it should not be watered all in one shot. It should be intermittent, taking the course of several hours which would lead to deeper penetration. Watering the soil all at once would also cause run off and lead to uneven water distribution since this method does not allow sufficient time for the soil to even absorb the water. Most plants require only a good drenching once or twice a week. This method of watering is preferred rather than to water the plants lightly on a daily basis. Excessive or infrequent watering can lead to lawn diseases, insects, mold, and drowned root damage just to name a few. The average lawn requires only approximately 1 to 2 inches of water per week depending on the weather conditions. As mentioned above, if there was a large rain storm then little to no watering is required for that week.

As for the time of the day, the best time to water is early in the morning. This is due to the fact that the low amount of heat and sun light will allow the soil to absorb a sufficient amount of water but also not have excess water lying around leading to what was mentioned above; that is lawn disease, insects, mold, and drowned root damage.

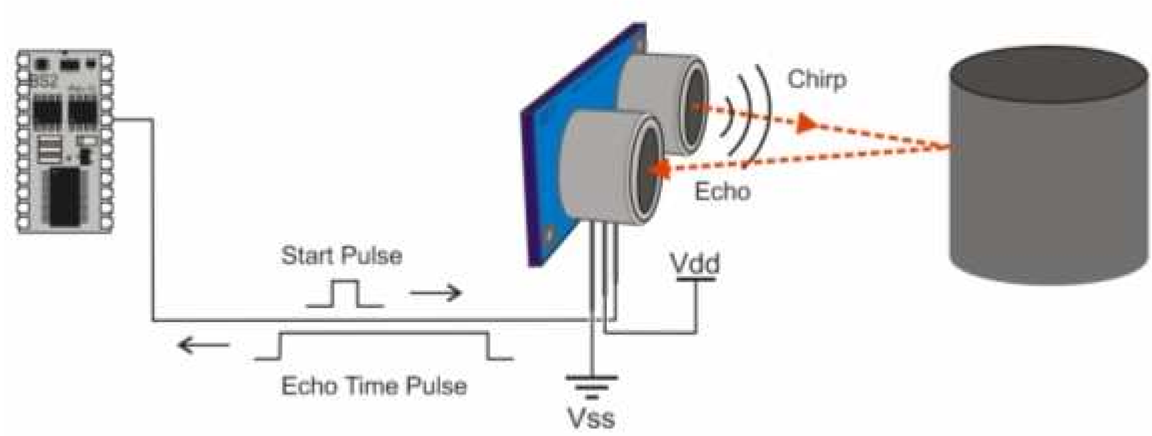
The next best time of the day to water is late afternoon as it allows enough water to be absorbed and evaporated just as the early morning process does. The worst times of the day to water are mid-afternoon and late night. During mid-afternoon the shear temperature of the water would burn the plants and also it would require a significantly larger portion of water since it would evaporate very quickly due to the high heat. Late at night, the water will take a much longer time to evaporate and absorb into the water, thus causing lawn disease, insects, mold, and drowned root damage. It is also important to allow the soil to dry after each watering session.

While watering, it is important to note that every inch of water per 1000 square feet is approximately 623 gallons of water. This will soak 6 to 8 inches of soil and if there is any run off, it is a good indication to stop immediately. Some time must be allowed to pass in order for the soil to dry off before the next watering phase. The amount watered so far depends on the flow rate of the hose. A simple way to tell how much water has covered a certain area of the lawn is to place a cup in that area and then turn on that sprinkler. The time it takes to gather 1 inch of water needs to be determined.

Another method is to multiply the square footage of the lawn by 0.62 gallons (1 inch of water per square foot). This determines exactly how much water is needed for 1 week. With this, the watering schedule can be created.

2.3 - Technical Background

There are a variety of methods in order to perform distance detection. One common way is time-of-flight ranging. In this method, a radio frequency device, an optical reflective pair, or an ultrasonic speaker and microphone pair may be utilized to measure the round-trip time for a pulse to travel to an object and return back to the sensor. The designed product makes use of the PING))) Ultrasound sensor, which exploits the travel of acoustic waves and the calculation for the speed of sound at room temperature to determine the distance of an object from it. The autonomous sprinkler system will use the device in order to detect whether an intruder is approaching the lawn sprinkler's water jet. If either one of two range finders detects an abrupt change in distance measurement, the BASIC Stamp 2 module sends out a PWM signal to a custom-built motor-valve assembly that controls the amount of water flow within the hose. Therefore it is the ultrasound sensor that is the foundation of the proposed product; a smart sprinkler (with a BASIC Stamp 2 module as its CPU) which monitors range inputs and reduces the sprinkler flow rate to avoid objects based on these range inputs. The figure on the next page displays the operation of an ultrasound sensor interfaced with BS2.

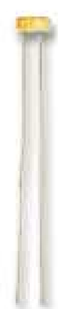


Ping))) Distance Sensor
(Source: www.Parallax.com)

Component Selection and Testing

In order for our product to respond to the weather and its environment, the selection of appropriate and affordable sensors was critical.

In order to sense brightness, a photo resistor was chosen because of its simple, variable resistance behavior, its extremely low cost, and its small, unobtrusive size. Testing this sensor by placing a flashlight at varying distances from the sensor, as well as turning the lights of a room on and off, gave the designers an idea of what the light sensor will output at various times of a typical day. This allowed for the selection of an appropriate capacitor value to use in the sensor's RC circuit.



Photoresistor

In order to sense temperature, a temperature probe, thermocouple, or thermistor would have done the job fine. However, most of these options are too expensive and have rapid sampling rates that are not necessary for this application. The affordable, practical alternative in this case was the DS1620 temperature sensor IC because of its small, easy-to-interface DIP architecture, its low cost, and its moderate sampling time.

DS1620 Temperature Sensor
(Source: www.Parallax.com)

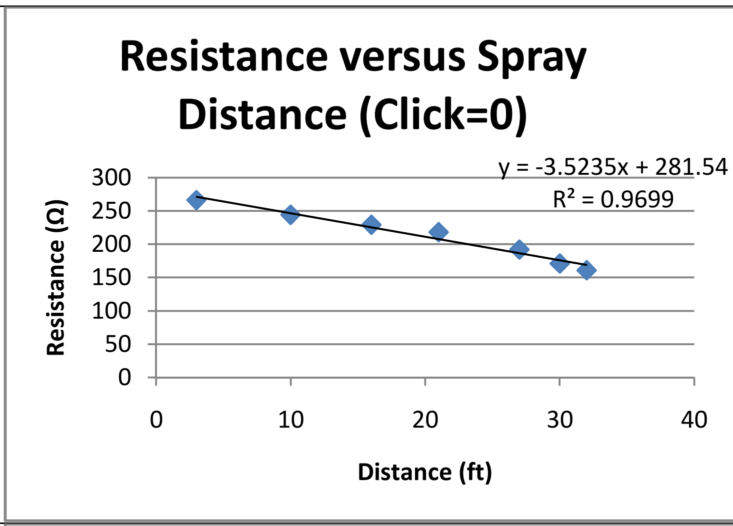
Since no sensors for detecting moisture are available from Parallax, and since there are several easy methods to construct homemade moisture sensors, the soil moisture sensor used for this product was built according to the suggestions presented by a website (see appendix). This sensor functions as a soil moisture sensor and was extremely easy to construct using two paper clips, a drinking straw, and plaster of Paris. Testing this sensor using separate

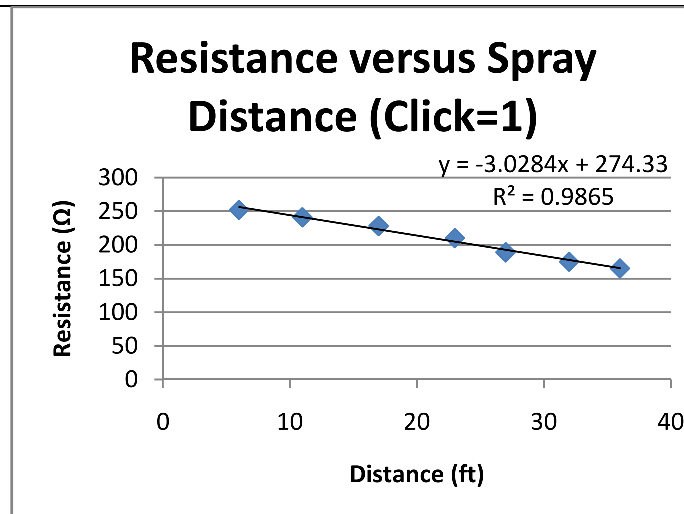
Moisture sensor

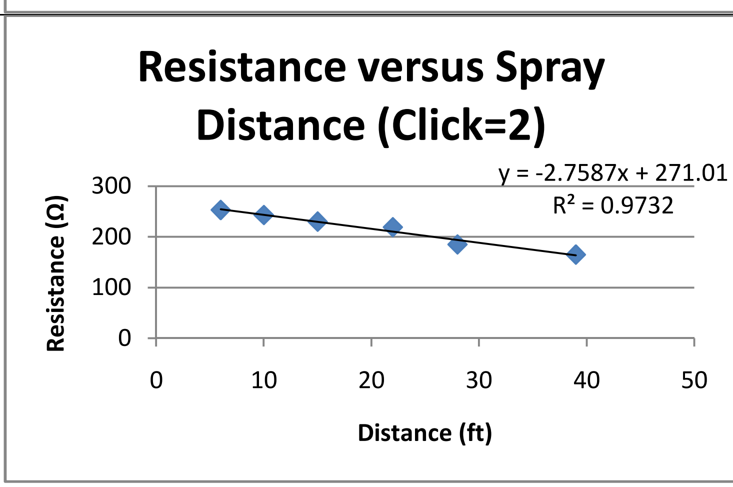
In order to sense pressure, no method could have been simpler than this product's variable resistor method. Though the (MAKE/MODEL/ETC) was chosen because of its affordable price and its ease of use, the most promising characteristic of this selection is the device's linear behavior which ultimately simplifies the feedback control of the DC motor. Testing of this sensor was done by varying the input pressure to the hose while keeping the amount of clicks of the sprinkler head constant and then varying the amount of clicks of the sprinkler head while keeping the input pressure constant. The results of this testing are shown in the Figure. The results of these tests provided positive results: under the vast majority of operating conditions of the sprinkler system, the pressure sensor output a linear relationship between pressure and resistance. The results of the experiment allowed for the selection of an appropriate capacitor value to use in this sensor's RC circuit.

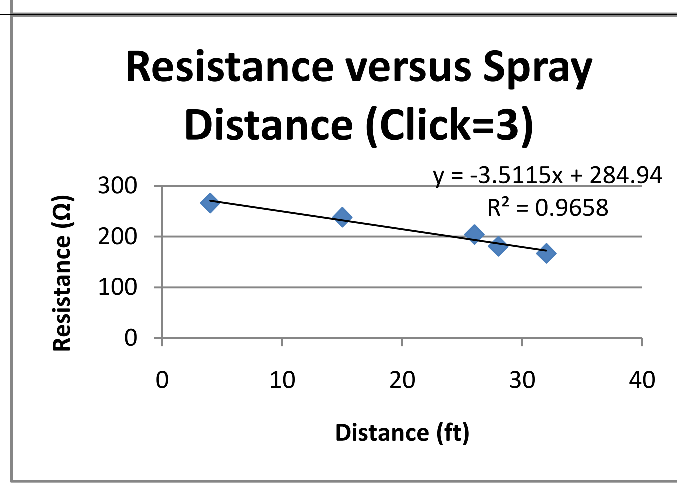
Pressure Sensor

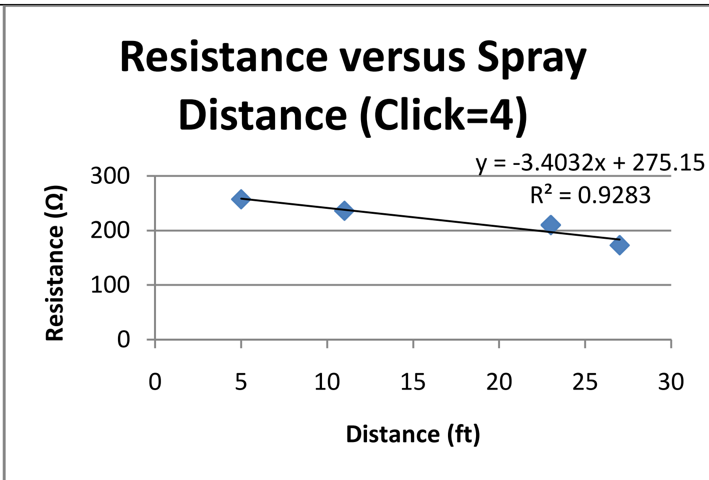
Below is experimentally obtained data, it shows the relationship between resistance (Ω) and spray distance (ft). The number of "Clicks" is the position of the spray interrupter which changes the range of water spray. The general relationship is that the resistance decreases almost linearly as the flow rate increases (higher range).

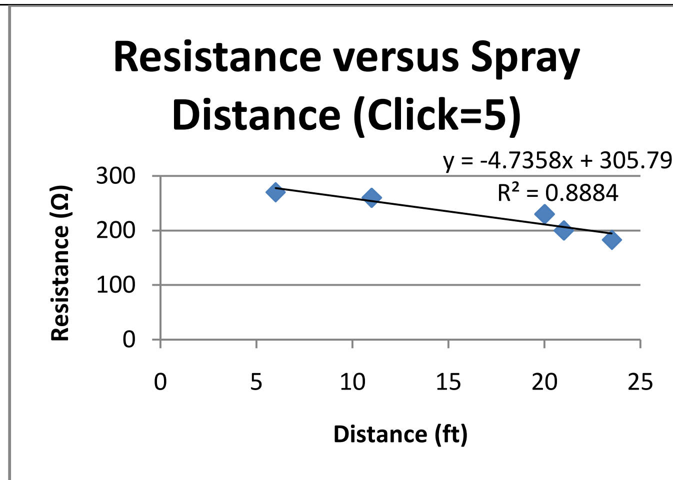


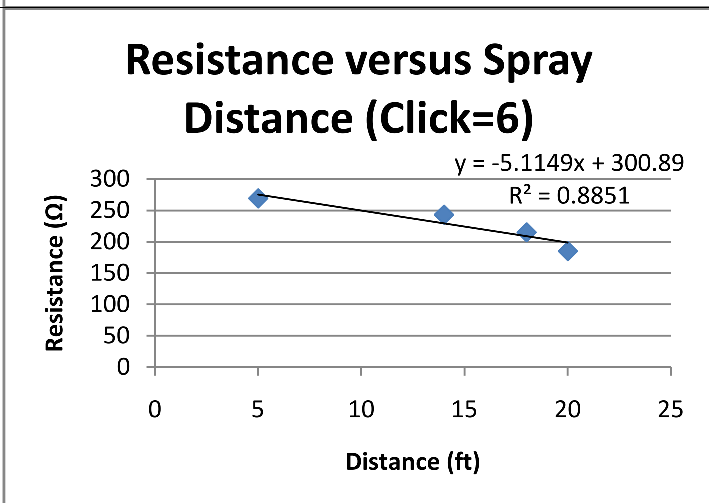


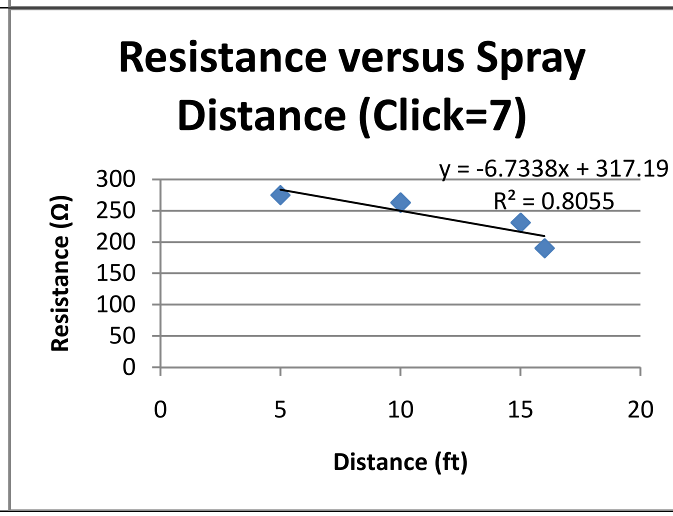


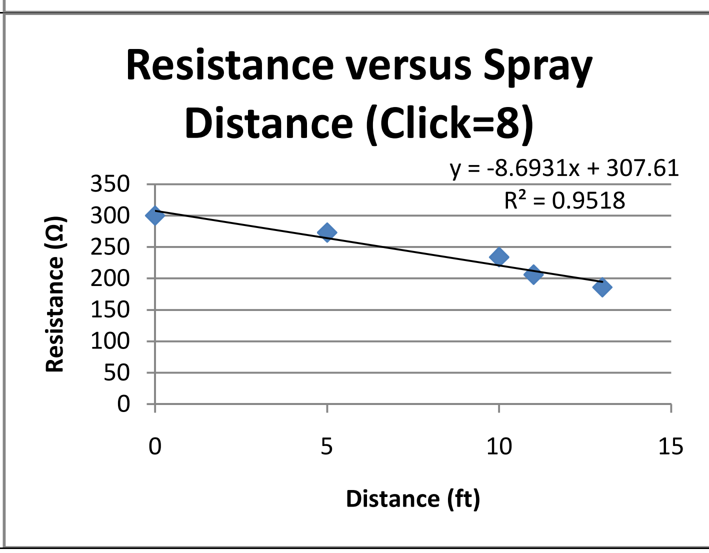



Resistance versus Spray Distance (Click=4)
y = -3.4032x + 275.15
R² = 0.9283
Resistance (Ω)
Distance (ft)


Resistance versus Spray Distance (Click=5)
y = -4.7358x + 305.79
R² = 0.8884
Resistance (Ω)
Distance (ft)


Resistance versus Spray Distance (Click=6)
y = -5.1149x + 300.89
R² = 0.8851
Resistance (Ω)
Distance (ft)

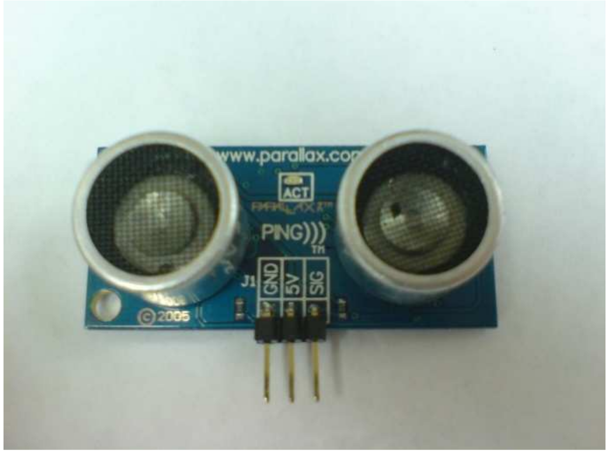

Resistance versus Spray Distance (Click=7)
y = -6.7338x + 317.19
R² = 0.8055
Resistance (Ω)
Distance (ft)


Resistance versus Spray Distance (Click=8)
y = -8.6931x + 307.61
R² = 0.9518
Resistance (Ω)
Distance (ft)

Clicks	Distance (ft)	Resistance (Ω)
8	13	186
	11	206
	10	234
	5	273
	0	300
7	16	190
	15	231
	10	263
	5	275
6	20	185
	18	215
	14	243
	5	269
5	23.5	183
	21	200
	20	230
	11	260
	6	270

Clicks	Distance (ft)	Resistance (Ω)
4	27	173
	23	210
	11	236
	5	257
3	32	167
	28	181
	26	204
	15	238
	4	266
2	39	165
	28	185
	22	219
	15	230
	10	243
	6	253
1	36	165
	32	175
	27	189
	23	210
	17	228
	11	241
	6	252
0	32	161
	30	171
	27	192
	21	218
	16	229
	10	244
	3	266

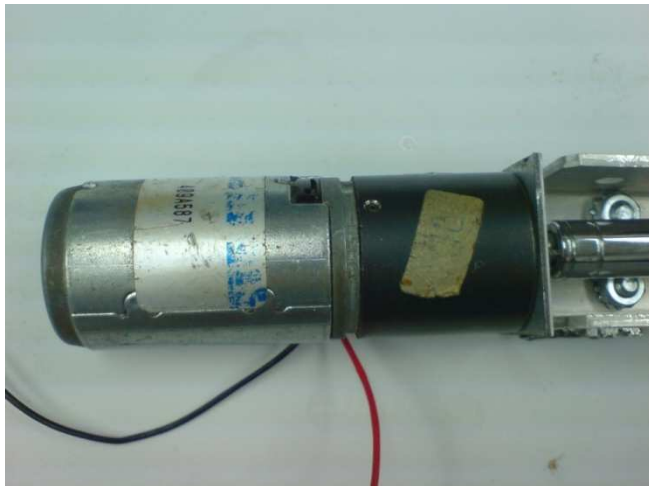
Choosing the method of distance sensing was not an easy task. Since every other available method of ranging proved to be either too sophicated and expensive or less sophisticated and less expensive, the best choice was to choose a ranging method that would provide medium range and affordable cost.



PING Distance sensor

In order to change the flow rate through the system, a valve of the appropriate diameter was chosen. This diameter is 5/8”. The valve is _____. In order to drive the valve, a motor with the proper speed-torque ratio was required. The DC motor chosen for this product needed to be able to control the valve quick enough in order to avoid obstacles, yet provide enough torque to drive the valve screw.


DC Motor-Sprinkler Valve Apparatus



DC Motor

DC Motor-Sprinkler Valve Connection

Sprinkler Valve

There are a variety of ways in which timing could have been implemented to the system. This timing was necessary in order to monitor how long the sprinkler has been activated and when it would need to activate after a certain length of idling time. The simplest and most affordable method was selected: a 555 timer operating in astable mode outputting a steady frequency of pulses (0.5 Hz).

An appropriate housing mechanism was desired to place and waterproof all the circuitry as well as position the range finders at a fixed, proper orientation from one another. Not only were RadioShack's project enclosures inexpensive, they provide an increased amount of robustness and ruggedization to the product as well as an aesthetically pleasure appearance.

Materials & Cost

Part #	Item	Item Cost ($)	Quantity	Total Cost ($)
01	Basic Stamp 2 Module	49.00	1	49.00
02	Sprinkler Head	4.99	1	4.99
03	Hose	11.00	1	11.00
04	Project Box (big)	4.99	1	4.99
05	Project Box (small)	2.99	2	5.98
06	LCD	29.99	1	29.99
07	Pushbuttons	0.99	4	3.96
08	Pressure Sensor	19.99	1	19.99
09	Photoresistor	1.99	1	1.99
10	Moisture Sensor	0.99	1	0.99
11	Temperature Sensor	6.99	1	6.99
12	Range Finder	29.99	2	59.98
13	Killswitch	0.99	1	0.99
14	555 Timer	1.99	1	1.99
15	Resistors	0.15	10	1.50
16	Capacitors	0.20	5	1.00
17	Jumper Wire	0.05	20	1.00
18	PCB	2.00	1	2.00
19	LED	0.50	2	1.00
20	Additional Accessories			0.00
	Total Cost			**207.33**

If this product was mass produced, the cost to manufacture each unit would be significantly reduced. An essential way of reducing the price is if the Basic Stamp module were replaced by a low to midrange PIC microcontroller with the appropriate amount of memory, computing speed, and digital I/O pins. The temperature sensor could have been replaced by an inexpensive thermistor and controlled using an RC circuit. The pressure sensor could have been replaced by a less expensive position tracking method such as a potentiometer attached to a rubber wheel. Making these adjustments and purchasing materials in bulk, as well as obtaining a more efficient and affordable ranging device, the estimated manufacturing cost of the mass produced product would easily become less than half the price calculated above.

Mechanical Design

The project was constructed as two major subsystems. The first consisted of the large project box as its foundation, containing the circuit board, the attached motor-valve assembly, and the LCD and pushbuttons for human interfacing. A slot and four buttons were machined in the large project box in order to fit the LCD screen and pushbuttons. Additional holes were machined to accept the wires running from the smaller project box. The selected DC motor was fitted to the water valve using an aluminum C-bar and bolts. A dual power supply is used for the directional control of the motor; 5V is used in order to close the valve and 9V is used to open the valve. This is because additional power is required in order to overcome the tightening of the valve as well as the static friction developed between the parts.

The second subsystem of the product consisted of a sprinkler head with a small project box mounted on it. The sprinkler head's rotation is governed by the flow rate of water through it – it also mechanically reverses its direction once its predefined extremes of rotation are reached. Therefore, there is no necessity for a motor to rotate the sprinkler. A range finder is mounted to the assembly facing slightly ahead of the sprinkler head in order to apprehend an obstacle before the water hits it. A problem occurs: how will the system detect an object when the sprinkler head rotates in the opposite direction? If two motors were used – one to control the water valve and the other to rotate the range finder back and forth between the appropriate orientations – the product would not have been able to sense an object approaching from behind it. Thus a person walking on a sidewalk toward the water would have been unlucky. Therefore, the decision was made to use two range finders which would be housed in the same small project box as opposed to two boxes in order to simplify machining time, orientation angle calculations, and reduce the cost of purchasing an additional project box. The smaller project box which contained the range finders was mounted atop the sprinkler head in order to rotate with it. This was done using vex aluminum members. Holes were also machined into the box in order to run wires through it from external sensors, as well as run wires from the entire subsystem to the large project box, which was designed to hold the product's main circuit board.

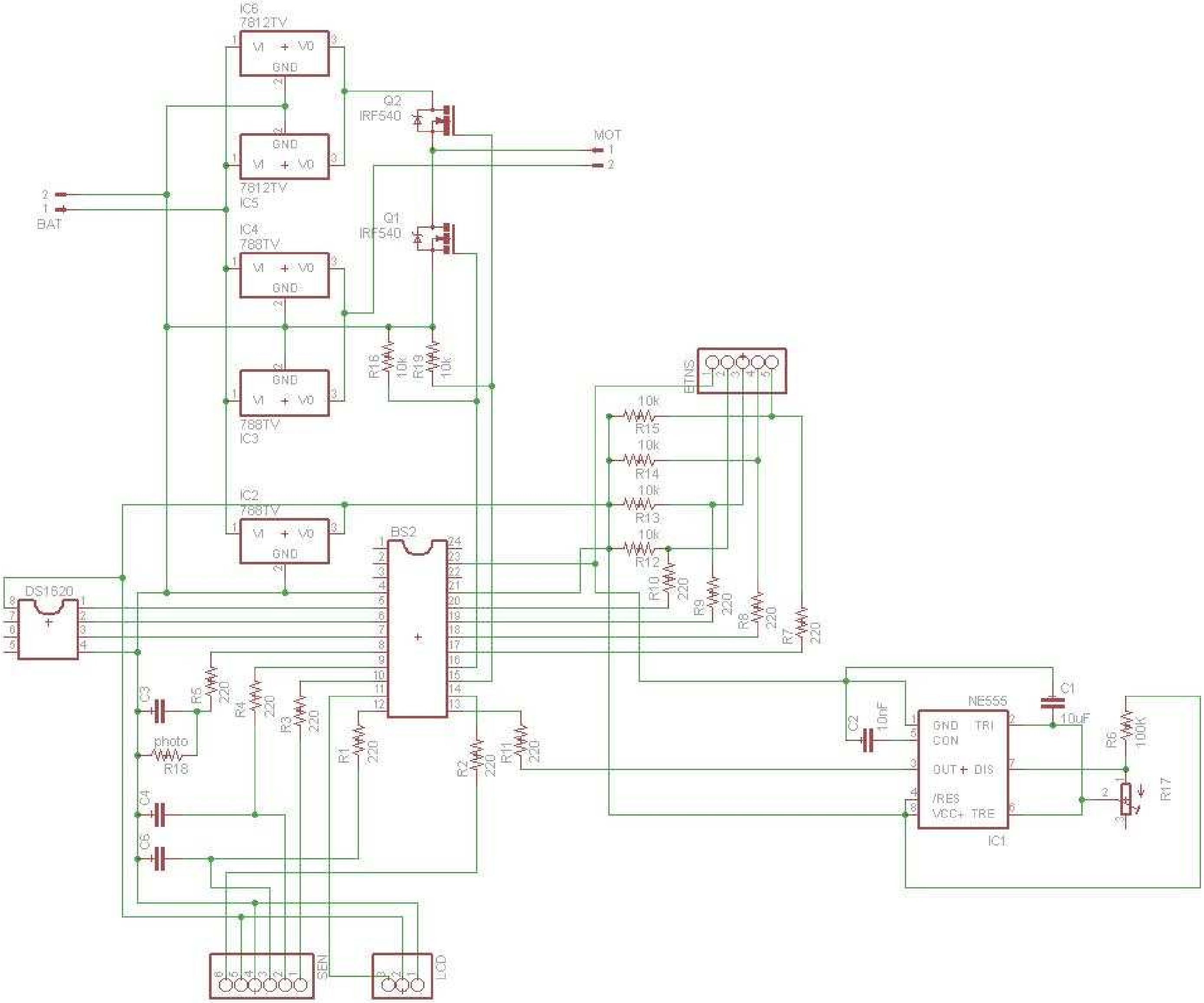
Circuit

The circuit was designed using Cadsoft Eagle v5.5.0 and etched by first printing the circuit diagram onto photo paper and then transferring the ink from the photo paper to a printed circuit board. After this was done, the components were soldered to the board and the external sensors and devices were attached using jumper wires. The circuit makes use of parallel RC circuits to monitor the varying resistance sensors – the pressure sensor, photo resistor, and soil moisture sensor.

The circuit for the temperature sensor was borrowed from Parallax's App Kit for the DS1620 temperature sensor IC. An astable multivibrator 555 timer circuit was chosen to function as the product's timing mechanism. The resistors and capacitors of this circuit were chosen appropriately to output a frequency of 0.5Hz; each high pulse sensed by BS2 represents the passing of two seconds. The second resistor of the 555 timer circuit, R2, was replaced with a trimmer potentiometer to allow for fine tuning in order to compensate for time delays in the program as well as instruction speed.

In order to perform direction control of the DC motor, a half bridge constructed with discrete n-channel enhancement MOSFETs and a dual power supply was utilized. Voltage regulators were used in order to provide an externally regulated 5 Volts to the BS2 and low power circuitry as well as provide regulated 5V and 9V to the motor.

Four pushbutton circuits and a Parallax 2x16 non-backlit LCD were used for human interfacing. The pushbuttons were chosen to be normally open active low devices with pull-up resistors since these buttons are the preferred type in avoiding noise-induced errors. Also, external pins were placed for the connections with the external sensors and the range finders, which are located in and near the small project box.


IC6
7812TV
IC5
IC4
788TV
IC3
IC2
Q2
IRF540
Q1
MOT
BAT
R16
R19
10k
ETNS
R15
R14
R13
R12
BS2
DS1620
R10
R9
R8
R7
220
NE555
C1
10uF
C2
10nF
R6
100K
R17
IC1
C3
R5
photo
R18
R4
R3
R1
R2
R11
C4
C6
SEN
LCD

Program

The PBASIC program used for this product is contained in the appendix.

Advantages & Disadvantages

The disadvantages of this product are its cost and its limited range. The limited range of the product was a result of the fact that there simply is not an available market base for affordable distance sensors with a range much longer than 15 feet. The distance sensors that would have given this product an appropriate range would have utilized laser technology. This would have drastically increased the price to an impractical amount. Another disadvantage is that the motor-valve is only programmed to operate between an open state and a slightly closed state. The closed state is still open enough to cover a small radius and move slowly, however it would be more desirable to continuously spray at a radius just below the distance of the intruding object. This is a promising possibility for future developments.

Home owners have a great deal of problems to worry about concerning what needs to be done to maintain their home. One of the advantages of this project is that it provides a few conveniences that virtually no sprinkler products provide at this time. Of course, like other sprinkler systems, it allows the user to no longer have to water their lawns themselves. However, unlike most sprinklers, it provides the convenience of users no longer having to wonder how much water they should provide or having to calculate by themselves how long they must water their lawns. The product provides the convenience of the user and of passer-bys no longer having to worry about walking in front of the sprinkler, of driving their car by the sprinkler, of passing the sprinkler while delivering the mail, while ___. The product assures the user that their lawns will no longer be watered when it is not the appropriate time to water them. When the climate is very hot and it very bright outside, the sprinkler will supply the appropriate amount of water. When the weather changes and becomes cold and/or dark outside, the sprinkler's behavior will adjust to these changes and water the lawn appropriately. When the soil under the lawn is too moist or too dry, the sprinkler knows that it should or should not activate. This saves the customer a great deal of time, money, and stress in the long run when one considers the attempts home owners usually must make in order to understand and successfully manage their lawns. Therefore, the product relieves preoccupied customers of what is usually an aggravating priority.

Conclusion

The final result of the project was that in the amount of time given to complete the full product, the mechanical and circuit aspects were completely tackled. However, due to changes required in the motor control section of the circuit, the product is capable of successfully reading all sensor output but not able to control the motor. If given another week, the discrete MOSFETs would have been replaced with relays or a MOSFET H-Bridge IC to perform directional motor control. In the end, some slight adjustments in the program and its flow as well with the motor control part of the circuit would result in a product that could take the readings the project can already obtain and use them towards a successful object avoidance system.

Future Possibilities

Some future possibilities include increasing the distance sensors' range, and having a more sophisticated weather control. Currently the PING Distance Sensor's max range is 3 meters, which would limit this design to small lawns only. With a larger sensing range, the automatic mode of the sprinkler could be used in larger lawns. Improving the weather sensing devices would lend to better timing and watering precision.

	Requirement	**Our Project**
a	Must be controlled by BS2	Our project uses BS2
b	You must incorporate and document hardware and software features to prevent damage to the BS2 IC and other components on your device. In addition, you must provide guidelines for safe operation of your device. Include a provision for instantaneous shutdown of your device in case of incorrect/unsafe operation	Project boxes are used to keep circuitry protected from surroundings. Safety resistors and transistors in use. Killswitch for emergency termination. High current circuitry electrically isolated from BS2 and low current circuitry
c	Your project must include some form of a user interface so that a human user can monitor and control your device (excluding 2b above)	Menu provides either Auto or Manual mode. Manual mode includes full motor control
d	Your project must utilize at least one actuator (excluding 2b, 2c above)	Our project uses a DC motor for valve control and an LCD for human interfacing
e	Your project must utilize at least one analog and one digital sensor (excluding 2b, 2c above)	Our project uses a handful of both analog and digital sensors
f	Your actuator must be controlled using sensory feedback. You can use any primitive to advance control design methodology for this purpose	The DC motor is controlled by the pressure sensor readings which dictate the flow condition

References

http://www.parallax.com/
http://www.cheapvegetablegardener.com/2009/03/how-to-make-cheap-soil-moisture-sensor.html
http://www.gardenguides.com/
http://www.backyardstyle.com/watering-guide.php
http://www.sunset.com/garden/earth-friendly/summer-watering-guide-00400000011810/
http://gardeneryardener.blogspot.com/2006/06/why-soil-temperature-is-important.html

Appendix A: PBASIC Code

```
'This is the program for Group 4's Autonomous Sprinkler System.
'Temperature, Light, Range, Soil Moisture, and Pressure in the
'sprinkler head are being measured. Motor is being controlled to
'activate flow AND control flow rate.
'
' {$STAMP BS2}          ' Target device = BASIC Stamp 2
' {$PBASIC 2.5}         ' Language = PBASIC 2.5
'
'************** PINS, VARIABLES, AND CONSTANTS ******************  <<<<<<<<<
'
' ==================== PIN ASSIGNMENTS ========================
'
BUTTONA PIN 15          'Push Buttons
BUTTONB PIN 14
BUTTONC PIN 13
BUTTOND PIN 12
MotorF PIN 11          ' Forward Motor Pin (Drive High while MotorR is Low)
MotorR PIN 10          ' Reverse Motor Pin (Drive High while MotorF is Low)'
RangePinRight PIN 9     ' Pin 9 <=> 2nd PING ))) Range Finder
ClockPin PIN 8         ' Pin 8 <=> Timer
PressurePin PIN 7       ' Pin 7 <=> Pressure Sensor
LcdPin PIN 6           ' Pin 6 <=> LCD Pin
RangePinLeft PIN 5      ' Pin 5 <=> PING ))) Range Finder
MoistPin PIN 4         ' Pin 4 <=> Moisture Sensor
LightPin PIN 3         ' Pin 3 <=> Light Sensor
DQ PIN 2               ' Pin 2 <=> Temp DQ.
CLK PIN 1              ' Pin 1 <=> Temp CLK.
RST PIN 0              ' Pin 0 <=> Temp RST (high = active).
'
Mode VAR Bit           '0 => Auto , 1=> Manual
'
' ==================== Temperature Variables ==========
Temp VAR Word          ' Word variable to hold 9-bit data.
TempL CON 10           'Lower Temperature Threshold
TempH CON 30           'Higher Temperature Threshold
Sign VAR Temp.BIT8      ' Sign bit of raw temperature data.
T_sign VAR Bit         ' Saved sign bit for converted temperature.
```

```
' ==================== Temperature Constants ==========
' >>> Constants for configuring the DS1620
Rconfig CON $AC         ' Protocol for 'Read Configuration.'
Wconfig CON $0C         ' Protocol for 'Write Configuration.'
CPU CON %10             ' Config bit: serial thermometer mode.
NoCPU CON %00 '         Config bit: standalone thermostat mode.
OneShot CON %01         ' Config bit: one conversion per start request.
Cont CON %00            ' Config bit: continuous conversions after start.
' >>> Constants for serial thermometer applications.
StartC CON $EE          ' Protocol for 'Start Conversion.'
StopC CON $22           ' Protocol for 'Stop Conversion.'
Rtemp CON $AA           ' Protocol for 'Read Temperature.'
' >>> Constants for programming thermostat functions.
RhiT CON $A1            ' Protocol for 'Read High-Temperature Setting.'
WhiT CON $01            ' Protocol for 'Write High-Temperature Setting.'
RloT CON $A2            ' Protocol for 'Read Low-Temperature Setting.'
WloT CON $02            ' Protocol for 'Write Low-Temperature Setting.'
'
' ==================== Light Variables =================
Light VAR Word          'Word variable to hold 9-bit data.
' ==================== Light Constants ==================
LightL CON 40           'Too bright condition
LightH CON 20000        'Too dark condition
' ==================== Soil Moisture Variables =========
Moisture VAR Byte
' ==================== Soil Moisture Constants =========
MoistureH CON 100       'Theshold moisture to be compared against
' ==================== Range Variables =================
DistanceLeft VAR Word
DistanceRight VAR Word
timeLeft VAR Word
timeRight VAR Word
' ==================== Range Constants =================
' Conversion constant for room temperature measurements.
CmConstant CON 2260
'
' ==================== Pressure Variables ==============
Pressure VAR Byte
' ==================== Pressure Constants ==============
PressureOpen CON
PressureClosed VAR 66
' ==================== LCD Variables ===================
' ==================== LCD Constants ===================
Baud9600 CON 84
HOMEX CON 128
CLSX CON 12
CRX CON 13
'
' ==================== Clock Variables =================
second VAR Word
'
'********************** INITIALIZATION ************************ <<<<<<<<<
'
' ====================Temp Initialization ====================
LOW RST                 ' Deactivate '1620 for now.
HIGH CLK                ' Put clock in starting state.
```

```
PAUSE 100                ' Let things settle down a moment.
HIGH RST                 ' Activate the '1620 and set it for continuous..
SHIFTOUT DQ,CLK,LSBFIRST,[Wconfig,CPU+Cont] ' ..temp conversions.
LOW RST                  ' Done--deactivate.
PAUSE 50                 ' Wait for the EEPROM to self-program.
HIGH RST                 ' Now activate it again and send the..
SHIFTOUT DQ,CLK,LSBFIRST,[StartC] ' Send start-conversion protocol.
LOW RST                  ' Done--deactivate.
' ======================LCD Initialization ===================
SEROUT LcdPin, Baud9600, [22, 12] ' Initialize LCD
'
'
'********************** PROGRAM ****************************** <<<<<<<<<

'Introduction at start-up
SEROUT LcdPin, Baud9600, ["Welcome!"]
PAUSE 1000
SEROUT LcdPin, Baud9600, [CLSX]
PAUSE 500
SEROUT LcdPin, Baud9600, [HOMEX, "Press any button", CRX, "to begin"]

DO
LOOP UNTIL (BUTTONA = 0) OR (BUTTONB = 0) OR (BUTTONC = 0) OR (BUTTOND = 0)     ' wait for
button
DO
LOOP UNTIL (BUTTONA = 1) AND (BUTTONB = 1) AND (BUTTONC = 1) AND (BUTTOND = 1)   'wait
for button to unpress

Menu:
SEROUT LcdPin, Baud9600, ["Select Mode:", CRX, "A=Auto  B=Manual" ]

DO
LOOP UNTIL (BUTTONA = 0) OR (BUTTONB = 0)                    'wait for button

IF (BUTTONA=0) THEN
  DO
  LOOP UNTIL (BUTTONA=1)                                     'wait for button to unpress
WRITE 0, (0)
GOSUB Auto                                                 'initiate Auto mode
ENDIF
IF (BUTTONB=0) THEN
  DO
  LOOP UNTIL (BUTTONA=1)                                     'wait for button to unpress
WRITE 0, (1)
GOSUB Manual                                               'initiate Manual mode
ENDIF


Auto:
SEROUT LcdPin, Baud9600, [CLSX, HOMEX, "Auto Mode"]
DO

GOSUB ReadTemp                                             'get Temperature reading
GOSUB ReadLight                                            'Light reading
GOSUB ReadMoisture                                         'Moisture reading
IF (Temp > TempL) AND (Temp < TempH) THEN
```

```
 IF (Light > LightL) AND (Light < LightH) THEN                              'if weather condition is fine
  IF (Moisture < MoistureH)THEN
  'initiate the timer
  second = 0
   GOSUB Activate                                                  'activate sprinkler
   ELSE                                                       'otherwise
   SEROUT LcdPin, Baud9600, [SDEC Temp, "F L=", DEC Light, CRX, "M=",DEC Moisture]    'display
sensor results
   PAUSE 1000                                                   'for one second
   ENDIF
   ENDIF
   ENDIF
   SEROUT LcdPin, Baud9600, [CLSX, HOMEX, "A=Moisture,B=Temp", CRX, "C=Menu"]         'give
choice to view sensors
   PAUSE 1000                                                   'or return to menu


IF (BUTTONA=0) THEN
 DO                                                          'Wait for unpress
 LOOP UNTIL (BUTTONA=1)
 SEROUT LcdPin, Baud9600, [SDEC Moisture, CRX]                              'Show moisture reading
ENDIF
IF (BUTTONB=0) THEN
 DO                                                          'Wait for unpress
 LOOP UNTIL (BUTTONB=1)
 SEROUT LcdPin, Baud9600, [SDEC Temp," degF", CRX]                             'Show signed
temperature in F.
 GOTO Menu
ENDIF
IF (BUTTONC=0) THEN
 DO                                                          'Wait for unpress
 LOOP UNTIL (BUTTOND=1)
 GOTO Menu                                                       'Goto menu
ENDIF


LOOP
RETURN



Manual:
SEROUT LcdPin, Baud9600, [CLSX, HOMEX, "A=Open B=Close", CRX, "C=Menu"]

DO
LOOP UNTIL (BUTTONA=0) OR (BUTTONB=0) OR (BUTTONC=0)
 IF (BUTTONA=0) THEN
  DO
  LOOP UNTIL (BUTTONA=1)
 IF (Pressure > PressureOpen - 5) AND (Pressure < PressureOpen + 5) THEN 'If motor is within some
reasonable range of being open already
 SEROUT LcdPin, Baud9600, ["Already Open"]                         'Do not Open --> Alert the user that its
already open
 GOTO Manual
 ELSE
 HIGH MotorF
```

```
 LOW MotorR
 PAUSE 1000
 ENDIF
 ENDIF

 IF (BUTTONB=0) THEN
 DO
 LOOP UNTIL (BUTTONB=1)
 IF (Pressure > PressureClosed - 5) AND (Pressure < PressureClosed + 5) THEN 'If motor is within some reasonable range of being close already
 SEROUT LcdPin, Baud9600, ["Already Closed"]                    'Do not close --> Alert the user that its already closed
 GOTO Manual
 ELSE

 ENDIF
 ENDIF

  IF (BUTTONC=0) THEN
 DO
 LOOP UNTIL (BUTTONC=1)
 GOTO Menu
 ENDIF

RETURN


Activate:
DO
GOSUB Timer
GOSUB ReadRange
IF (DistanceLeft < 310) OR (DistanceRight < 310) THEN              'If distance is less than maximum, object is detected
   IF (Pressure<(65-DistanceLeft/16) OR (Pressure<(65-DistanceRight/16)) THEN   ' check that water does not shoot that far
     GOSUB motorlow
     ELSE
       IF (Pressure>(70-DistanceLeft/16) OR (Pressure>(70-DistanceRight/16)) THEN   ' check that water does not shoot that far
     GOSUB motorhigh
     ENDIF
    ENDIF                                               'close the valve
  ELSE
    IF (Pressure<47) THEN                                    'if nothing detected check the flow and open up to full
      GOSUB motorhigh
    ENDIF

PAUSE 150                                              'alow for flow adjustment
' try again
ELSE
ENDIF
LOOP UNTIL (second = 600)                                   'Stop until time is approximately 10 minutes
RETURN

motorlow:                                             'close valve
```

```
HIGH MotorF
LOW MotorR
PAUSE 200
LOW MotorF
LOW MotorR
RETURN

motorhigh:                               'open valve
HIGH MotorR
LOW MotorF
PAUSE 200
LOW MotorF
LOW MotorR
RETURN


ReadTemp:
HIGH RST                                 ' Activate the '1620.
SHIFTOUT DQ,CLK,LSBFIRST,[Rtemp]                  ' Request to read temperature.
SHIFTIN DQ,CLK,LSBPRE,[Temp\9]                   ' Get the temperature reading.
LOW RST
T_sign = Sign                            ' Save the sign bit of the reading.
Temp = Temp/2                              ' Scale reading to whole degrees C.
IF T_sign = 0 THEN no_neg1
Temp = Temp | $FF00                          ' Extend sign bits for negative temps.
no_neg1:
'convert to Farenheit
Temp = (Temp */ $01CC)                        ' Multiply by 1.8.
IF T_sign = 0 THEN no_neg2                     ' If negative, extend sign bits.
Temp = Temp | $FF00
no_neg2:
Temp = Temp + 32                            ' Complete conversion.
RETURN

ReadLight:
HIGH LightPin                              'charge capacitor
PAUSE 1                                   'allow capacitor to fully charge
RCTIME LightPin, 1, Light                      'collect discharge time
RETURN

ReadMoisture:
HIGH MoistPin                               'charge capacitor
PAUSE 1                                    'allow capacitor to fully charge
RCTIME MoistPin, 1, Moisture                     'collect discharge time
RETURN

ReadPressure:
HIGH PressurePin                              'charge capacitor
PAUSE 1                                      'allow capacitor to fully charge
RCTIME PressurePin, 1, Pressure                    'collect discharge time
RETURN

ReadRange:
PULSOUT RangePinLeft, 5                         'obtain range of left sensor
PULSIN RangePinLeft, 1, timeLeft
DistanceLeft = cmConstant ** timeLeft
```

```
PAUSE 100

PULSOUT RangePinRight, 5                              'obtain range of right sensor
PULSIN RangePinRight, 1, timeLeft
DistanceRight = cmConstant ** timeRight
PAUSE 100
RETURN


Timer:
IF (ClockPin=(second//2)) THEN                        'Detect if high pulse has occured
second=second+1
ENDIF
RETURN
```