Design and Implementation of Automatic Irrigation Control System

Oborkhale, Lawrence I (Ph.D.)¹, Abioye, A. E.², Egonwa B. O.³, Olalekan T. A.⁴  
¹Electrical/Electronic Engineering Department, Michael Okpara University of Agriculture, Umudike, Abia State, Nigeria.  
²Electrical/Electronic Engineering Department, Akanu Ibiam Federal Polytechnic, Unwana, Ebonyi State, Nigeria.  
³Electrical/Electronic Engineering Department, University of Ilorin, Ilorin, Kwara State, Nigeria.

Abstract: The objective of irrigation is to keep measure on food security and the aim of automatic irrigation control system is to minimize the intervention of the human operator (gardener) in irrigation activities. The automatic irrigation control system is used to achieve this aim. This control system is built around ATMEGA32 microcontroller programmed using embedded C language. Inputs are the signals from four sensors namely soil moisture sensor using hygrometer module, water level sensor using the LM 324 Op-amp was configured here as comparator, light sensor with the aid of Light dependent resistor and temperature sensor using LM 35. The microcontroller processes the input signals by using the control software embedded in its internal ROM to generate three output signals, using one of the output signals to control a water pump that irrigates the garden, the second output signals to control a water pump that draws water from the river to the reservoir or storage tank while the other to switch a buzzer that alerts the gardener when there is shortage of water in a tank that supplies the garden. The project can be applied in agricultural area of any type where water readily available for irrigation. It can also be applied in agricultural research institutes such as the Michael Okpara University of Agriculture, (MOUAU) Umudike.

Keywords: ATMEGA32, Automatic, Control, Embedded C language, Irrigation, Microcontroller, Sensor.

Introduction

Agriculture is the key to food security for every nation. Food security is a situation in which all people eat all the time and have a good economic access to sufficient, safe and nutritious food to meet their diet and nutritional needs and food preferences for a healthy and active life. Chief among high crop yield factors is irrigation. Irrigation is a process of causing water to flow over land to nourish plants or the watering of land by artificial means to foster plant growth [1].

After independence, successive Nigerian governments have adopted irrigation agriculture as a policy instrument for achieving the set objective of self-reliance and self-sufficiency in food production. The strategy of irrigation projects is primarily designed to mitigate the effects of drought and desertification on crop yield especially in the northern parts of Nigeria and increase crop production to meet the greater demands for food by the ever-increasing population of the country.

Irrigation can be described as the application of water to the soil to make available essential moisture for plant growth. It also serves as insurance against drought and to provide a cooling effect on the soil environment for plant growth and development. Irrigation is also aimed at improving and raising the productivity of soil resources. The principle, according to [2] is that the environment is characterized by fair to good soils but poor and unreliable low precipitation as it is the case in dry and semi-dry lands.

The manual method of irrigation is used predominantly by rural farmers in most developing countries especially in areas where the season of rainfall is very short. Earlier human used to control manually but this has always involved errors. Some of those controllers had to be automated.

Irrigation is carried out mainly through the use of surface or flood irrigation and the drip irrigation type. In the surface irrigation, water is applied and distributed over the soil surface by gravity. The drip irrigation allows water to drip slowly to the root system of plants so it is easier to control the amount of water on the root zone through a network of valves, pipes, and tubes.

The drip irrigation has many advantages over basin flooding and localized method of irrigation; it eliminates the possibility of soil erosion and can be used for the application of liquid fertilizers [3]. In the modern world today, automation is encompassing nearly every walk of life. Automation solutions are more accurate, reliable and flexible and also have replaced human efforts right from agriculture to space technology. Many automatic processes record parameters, analyze trend and control the desired parameter. The disadvantage of automation is the necessity of the manufacturing industries to survive in the globally competitive market. For any process to be automated, we need most essentially a real-time automatic controller that is a microcontroller based [4].

DOI: 10.9790/0661-1742991111 www.iosrjournals.org
With the advent of digital electronics and the invention of microprocessors and microcontrollers, came the concept of automation. The controllers developed, could be implemented in real-time with the help of the microprocessors or microcontrollers. Hence, we could control irrigation processes and the level of liquid data. Redefinesthepointof different parameters with the help of proper controller using an embedded device like a microprocessor or microcontroller to implement the control algorithm.

According to [5] in everyday life, there must be some physical elements that need to be controlled in order for them to perform their expected behaviours. A controller system therefore can be defined as a device or set of devices that manages, commands, directs, regulates the behaviour of other devices(s) or systems(s). Consequently, automatic control involves designing a controller system that functions with minimal or human intervention. Intelligent systems are being used in widerange of fields including from medical sciences to financial sciences, education, law, and soon. Several of them are embedded in the design of everyday devices.

An automated irrigation system has important advantages over other manual methods used by the local farmers: it ensures a more precise application and conservation of water, high crop yield, as well as the removal of human errors. The current trend in irrigation is to shift from manually operated type of irrigation to automated types.

Hence, the parameters that influence irrigation control are temperature of the environment, soil moisture content, light, and source of water. Temperature is the degree of coldness and hotness of a body and is commonly expressed in degree Celsius. This climatic factor influences all plant growth processes such as photosynthesis, transpiration rate, seed germination, protein synthesis and translocation. In general, plants survive within a temperature range of 0 to 50°C. Light is a climatic factor that is essential in the production of chlorophyll and in photosynthesis. Soil moisture content is the amount of water in the soil. All this influence plant growth and development.

I. Related Works

An automatic irrigation system based on embedded and Global System for Mobile Communication (GSM) technology was developed using ATMEGA (AT89S52). The developed system incorporates: Sensing devices which sense the dry condition of the defined field or farmland and pass the state to the sensing logic of the automation system, A Control algorithm for water flow regulation. The deficiencies in this irrigation system were: The system lacked the ability to detect soil temperature; the system had no timer mechanism for irrigation scheduling [6].

[7] Designed and developed a mobile irrigation lab for water conservation. The developed system comprises of Sprinkler irrigation management device and Water management and irrigation scheduling software program. The software program has essential functionalities to perform device calibration, fuel cost evaluation, pumping cost evaluation and also scheduling of pumping. The novelty in their design is the integration of both the hardware and software components. The system lacks detailed notifications based on sensed parameters and actions taken by the system when in action.

[8] Used an advanced microcontroller LM3S5T36 which is 32-bit ARM CortexM3 with features of flash, single, flash memory, 12kb RAM, three 32-bit timers and 10-bit analog digital converter. Developed an automatic drip irrigation system. However, the inability of this architecture to determine the exact temperature at which irrigation should commence due to the absence of temperature sensor makes it ineffective as irrigation can commence at any time even when it is sunny (high) which might affect the plant.

II. Materials and Methods

The hardware components used in this design include: ATEGA 32 Microcontroller, sensors, relay and ULN 2803 driver, Electric motor, Liquid crystal display, Transistors, Capacitors, Resistors and Buzzer.

The LM35 Temperature sensor has an integrated circuit sensor that can be used to measure temperature. The LM35 temperature sensor is an integrated circuit that can be used to measure temperature. The LM35 Temperature sensor is a resistive thermal sensor that can be used to measure temperature. The LM35 Temperature sensor is a resistive thermal sensor that can be used to measure temperature. The LM35 Temperature sensor is a resistive thermal sensor that can be used to measure temperature. The LM35 Temperature sensor is a resistive thermal sensor that can be used to measure temperature.

Water level sensor part is built using op-amp IC LM324. Op-amp was configured here as comparator for each water level. Soil Moisture Sensor is a Soil Hygrometer Detection Module used to detect moisture present in the soil and present an output in the form of analog voltage.

ULN2803 relay driver is a high-voltage, high-current Darlington transistor array. Each driver consists of seven NPN Darlington pairs that feature high voltage output with common-cathode clamp diodes for switching inductive loads. The collector—current rating of a single Darlington pair is 500 mA. The Darlington pairs can be paralleled for higher current capability with output to the feedersystem.
The Liquidcrystaldisplay(LCD)isathin,flat electronic visual displaythatusethelight modulatingpropertiesof liquidcrystals(LCs). This project used LCD16x2 as a display system. The LCD 16x2 has two rows, where each row displays 16 characters.

The Atmega 32 has an on-chip oscillator but requires an external clock to run it. A 16MHz quartz crystal oscillator is connected to inputs XTAL1 (pin 13) and XTAL2 (pin 12) for clocking of the microcontroller [9].

The simulated software algorithm began with flow-chart (Fig. 1) and finally an embedded C language program developed, which is converted to machine code (HEX file) and written to the microcontroller’s internal ROM for the appropriate controlling of the device [10].

The specific software was deployed for the virtual design and implementation of the device in the Proteus simulation software [11]. The software has two environments: the ISIS and the ARES environments. We used the ISIS environment for the circuit design and instead of implementing the printing of the circuit board (PCB) in the ARES, we used a Veroboard for our hardware implementation.

The diagram below describes theflow operation in the system as well as the inter-operability (Fig.1).

![Diagram of Automatic Irrigation Control System](image)

**Fig.1: Block diagram of Automatic Irrigation control System.**

This system consists of four inputs (temperature, Water level Sensor, Soil moisture and LDR sensor) and four output (buzzer, Sprinkler/drip water pump, Reservoir water pump and display unit). The Atmega 32 Microcontroller act the main brain for this system because it controls the overall irrigation process. It also has inbuilt ADC for conversion of analog signal from the sensors to digital form.

Temperature sensor used to detect the temperature in the farm environment. When the temperature sensor detects a high temperature, it gives an output voltage linearly proportional to the sensed temperature value. The Microcontroller will send the output signal to the relay base on the pre-set temperature value contained in the embedded program stored in the microcontroller to either ON or OFF the sprinkler pump.

The microcontroller receives input signals from light and moisture sensors. Depending on the input received, it takes decisions to let water out to the sprinkler system. Then, when water level sensor detects water, the microcontroller receives as input signal. Depending on the input received, it takes decision to close or pump water from the reservoir tank.

Hence, the value of the four monitored parameters are displayed on a 16 * 2 LCD Display. While the buzzer sounds an alarm when the water level in the reservoir is below 10%.

### III. System Design and Implementation

This work is divided into hardware and software sections. The hardware part consists of three sub-systems which include: the input sub-system, controls-sub-system and output sub-system.

DOI: 10.9790/0661-174299111
4.0 Hardware Sub System Design

4.1: Interfacing of Soil Moisture Sensor

A soil moisture sensor is excited by giving suitable power supply of 5V and is disconnected to one bit of Port A (PA3/ADC3). Port A pins are internally connected to ADC on chip which is of 10 bit resolution. The sensor senses the amount of moisture present in the soil and present an output in the form of an analog voltage ranging between 0V (fully wet condition) to 5V (completely dried condition) respectively as illustrated in the figure below using a potentiometer which is essentially an adjustable voltage divider used for measuring electric potential. As the moisture content increases, the resistance decreases and vice versa leading to change in analog voltage at the input pin (PA3/ADC3). The relation between the measured resistance and soil moisture is then calibrated in software to depict percentage of water present in soil.

![Fig. 3: Interfacing circuit of soil moisture sensor to Atmega 32](image)

The sensor values which are in analog form are converted to digital values in real time and are stored in the ADC data registers ADCL and ADCH respectively, and used to decide the condition of the relay which controls the valve and thus watering the field for predefined amount of time. This system uses the moisture range in between 1% to 50%. It can be changed manually whenever wanted.

4.2: Interfacing of Light Sensor

A light dependent resistor (LDR) sensor is excited by giving suitable power supply of 5V and is disconnected to one bit of Port A (PA2/ADC2). Port A pins are internally connected to ADC on chip which is of 10 bit resolution. The sensor senses the amount of light intensity present and present an output in the form of an analog voltage ranging between 0V (low light intensity) to 5V (high light intensity) respectively as illustrated in the figure below using an LDR and a resistor which is essentially a voltage divider adjustable and used for varying electric potential to Port A (PA2/ADC2). As the light intensity on the LDR increases, the resistance decreases and vice versa leading to change in analog voltage at the input pin (PA2/ADC2). As shown in Figure 4.
The sensor values which are in analog form are converted to digital values in real time and are stored in the ADC data registers ADC1 and ADC2 respectively. The average of all the sensor values is computed and used to decide the condition of the relay which controls the pump and thus watering the field for predefined amount of time.

4.3: Interfacing of Temperature Sensor

The LM35 temperature sensor is excited by giving suitable power supply of 5V and is connected to Port A (PA1/ADC1) while the third pin is grounded. Port A pins are internally connected to AD Conchip which has 10-bit resolution. The sensor senses the ambient temperature of the environment and gives an output analog voltage linearly proportional to the Centigrade temperature ranging between 0V (low temperature) to 5V (high temperature) respectively.

4.4: Interfacing of Water Level Sensor

The water level sensor used in this project is a comparator: LM324 (Low Power Quad Operational Amplifiers). The voltage output from the LM324 to the controller is 0V when the non-inverting input of the LM324 is not in contact with water. If the water sensor probe (non-inverting input) detects water, it conducts and if the voltage sensed is greater than the reference voltage, it gives a high output. Conversely, the voltage from the LM324 to the controller is about 4.7V (High).

It measures the level or height of water in a container using the conductive property of water. It’s made up of electrodes positioned at different levels in the container housing the liquid. As water rises in the container the electrodes make new connection with a varying analog voltage at different levels.
4.5: Output Interfacing Circuit

4.5.1: Interfacing of Relay and AC Motor with ATMEGA 32

The output port B of the ATMEGA 32 is interfaced with the river pump via a ULN 2803 Darlington Array driver and a 5V relay.

Relays are devices which allow low power circuits to switch a relatively high Current/Voltage ON/OFF. For a relay to operate a suitable pull-in & holding current should be passed through its coil. Generally relay coils are designed to operate from a particular voltage often its 5V or 12V.

If you want to connect more relays to microcontroller then you can use ULN 2003 for connecting seven relays or ULN 2803 for connecting eight relays.

Figure 7: Shows how to connect a relay to microcontroller using ULN 2003/ULN 2803. These IC’s are high voltage, high current darlington transistor arrays with open collector outputs and free-wheeling clamping diodes hence there is no need of a diode across the relay. Also there is no need of the series base resistor as the IC has an internal resistor of 2.7KΩ.
When the water level in the reservoir is low, the output of the output of the microcontroller port B (PB6/MOSI) which is same as the input of the ULN 2803 (PIN 5B and 6B) is high while the output at PIN 5C and 6C connected to the armature coil of the relay will be low. Hence the relay will be energized and the switch which is normally open will close, turning ON the river AC water pump.

When the water level in the reservoir is High (Filled), the output of the microcontroller port B (PB6/MOSI) which is same as the input of the ULN 2803 (PIN 5B and 6B) is low while the output at PIN 5C and 6C connected to the armature coil of the relay will be high. Hence the relay will be deenergized and the switch which is normally open will open, turning OFF the river AC water pump.

When the Preset value the sensors such Temperature, soil moisture and light is ok for irrigation is attained, the output of the microcontroller port B (PB7/SCK) which is same as the input of the ULN 2803 (PIN 7B and 8B) is high while the output at PIN 7C and 8C connected to the armature coil of the relay will be low.

Hence the relay will be energized and the switch which is normally open will close, turning ON the Drip AC water pump.

When the Preset value the sensors such Temperature, soil moisture and light is not ok for irrigation, the output of the output of the microcontroller port B (PB6/SCK) which is same as the input of the ULN 2803 (PIN 7B and 8B) is low while the output at PIN 7C and 8C connected to the armature coil of the relay will be high.

Hence the relay will be deenergized and the switch which is normally open will open, turning OFF the drip AC water pump.

### 4.5.2: Interfacing Circuit for LCD and Virtual Terminal

![Interfacing Circuit for LCD and Virtual Terminal](image)

Like many microcontrollers AVR also has a dedicated hardware for serial communication this part is called the USART – Universal Synchronous Asynchronous Receiver Transmitter. This special hardware makes life easier for programmer. You just have to supply the data you need to transmit and it will do the rest.

In this project, Serial communication occurs at standard speeds of 9600 bps and this speeds are slow compared to the AVR CPUs speed.

The advantage of hardware USART is that you just need to write the data to one of the registers of USART and you’re done, you are free to do other things while USART is transmitting the byte. Also the USART automatically senses the start of transmission of RX line and then inputs the whole byte and when it has the byte it informs you (CPU) to read that data from one of its registers.

### 4.6: Software Sub System Design

The software subsystem design involves programming of the microcontroller using embedded C programming language. A lot of registers of the microcontroller were configured in the control software burned into the microcontroller.

#### 4.6.1: USART of AVR Microcontrollers

The USART of the AVR is connected to the CPU by the following six registers.

- **UDR** – USART Data Register: Actually this is not one but two register but when you read it you will get the data stored in receive buffer and when you write data to it goes into the transmitters’ buffer.
- **UCSRA** – USART Control and status Register A: As the name suggests it is used to configure the USART and it also stores some status about the USART. There are two more of these kinds the **UCSRB** and **UCSRC**.
UBRRH and UBRRH: This is the USART Baud rate register, it is 16 bit wide so UBRRH is the High Byte and UBRRL is Low byte. But as we are using C language it is directly available as UBRR and compiler manages the 16 bit access.

4.6.2: UCSRA - USART Control and Status Register A

Table 4.1: USART Control and Status Register A

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXC</td>
<td>TXC</td>
<td>UDRE</td>
<td>FE</td>
<td>DOR</td>
<td>PE</td>
<td>U2X</td>
<td>MPCM</td>
</tr>
</tbody>
</table>

The binary value of the register is 00000000 while the hexadecimal equivalent is 00 hence UCSRA = 0X00

4.6.3: UCSRB - USART Control and Status Register B

Table 4.2: USART Control and Status Register B

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXCIE</td>
<td>TXCIE</td>
<td>UDRIE</td>
<td>RXEN</td>
<td>TXEN</td>
<td>UCSZ2</td>
<td>RXB8</td>
<td>TXB8</td>
</tr>
</tbody>
</table>

The binary value of the register is 10011000 while the hexadecimal equivalent is 98 hence UCSRB = 0X98

4.6.4: UCSRC - USART Control and Status Register C

Table 4.3: USART Control and Status Register C

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>URSEL</td>
<td>UMSEL</td>
<td>UPM1</td>
<td>UPM0</td>
<td>USB</td>
<td>UCSZ1</td>
<td>UCSZ0</td>
<td>LCPOL</td>
</tr>
</tbody>
</table>

The binary value of the register is 10000110 while the hexadecimal equivalent is 86 hence UCSRC = 0X86

4.7: Setting the Baud Rate for Serial Communication with the Virtual Terminal

USART Baud rate register is 16 bit wide, so UBRRH is the High Byte and UBRRL is Low byte. But as we are using C language it is directly available as UBRR and compiler manages the 16 bit access.

This register is used by the USART to generate the data transmission at specified speed (say 9600Bps). The USART needs a clock signal that determines the baud rate. It is generated in the chip by dividing the CPU clock frequency by the UBRR register value. It must be 16 x higher than the desired baud rate.

The x16 factor is used by the USART to sub-sample the received serial data, it improves noise immunity by calculating the received bit value from the average of 16 samples. UBRR value is calculated according to following formula.

\[
UBRR = \frac{fosc}{16 \times \text{Baud Rate}} - 1
\]

So if the desired baud rate is 9600 baud and the CPU clock is 16 MHz then UBRR is (16000000 / (16 x 9600)) - 1 = 103.167.

Round that to the closest integer = 103

Converting 103 from decimal to binary gives 01100111.

Hence 01100111 converting to hexadecimal give 67.

Which makes UBRRL = 0x67, UBRRH = 0x00.

4.8: Setting the Clock Frequency for ADC

ADC Prescaler Selects Bit: These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

ADPS2-ADPS0 Bit –

These bits selects the Prescaler for ADC. We set the ADC frequency to 1000 KHz i.e using Frequency of crystal Oscillator/Prescaler factor

16MHZ/16 = 1000 KHz
A Prescaler of 16 is chosen for this project

### Table 4.4: ADC Prescaler Selects Bit

<table>
<thead>
<tr>
<th>ADPS2</th>
<th>ADPS1</th>
<th>ADPS0</th>
<th>DIVISION FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>128</td>
</tr>
</tbody>
</table>

### 4.9: Setting ADC Status and Control Register

#### Table 4.5: ADC Status and Control Register A

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADEN</td>
<td>ADSC</td>
<td>ADATE</td>
<td>ADIF</td>
<td>ADIE</td>
<td>ADPS2</td>
<td>ADPS1</td>
<td>ADPS0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Bit7–ADEN:ADCEnable** – Set this to 1 to enable ADC
- **ADPS2-ADPS0** – These selects the Prescaler for ADC. We picked a prescaler of 16

ADC Start Conversion

ADSCR= 0X40 i.e 01000000

#### Table 4.6: ADC Status and Control Register B

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADEN</td>
<td>ADSC</td>
<td>ADATE</td>
<td>ADIF</td>
<td>ADIE</td>
<td>ADPS2</td>
<td>ADPS1</td>
<td>ADPS0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Bit6–ADSC:ADCStartConversion** - We need to set this to one whenever we need ADC to do a conversion.

#### Table 4.6: ADC Status and Control Register C

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADEN</td>
<td>ADSC</td>
<td>ADATE</td>
<td>ADIF</td>
<td>ADIE</td>
<td>ADPS2</td>
<td>ADPS1</td>
<td>ADPS0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Bit4ADIF:ADCInterruptFlag** – This is the interrupt bit this is set to 1 by the hardware when conversion is complete. So we can wait till conversion is complete by polling this bit like while ((ADCSRA & 0X10 ==0). The loop does nothing while ADIF is set to 0, it exits as soon as ADIF is set to one, and i.e. conversion is complete. ADSCR= 0X10 i.e 00010000
Fig 9: Circuit Design of Automatic Irrigation Control System
Fig. 10: Process Control Flowchart

1. **Start**
2. **Initialize**
3. **Is reservoir tank Empty?**
   - **No**
     - **Display Parameters on LCD**
     - **Switch on River pump**
   - **Yes**
     - **Display Parameters on LCD**
     - **Switch on River pump**
4. **Water Pumping**
   - **Is water level 50L?**
     - **No**
       - **Display Parameters on LCD**
       - **Read Temperature, Light, Water level and Soil moisture**
     - **Yes**
       - **Buzzer On**
5. **Is water level 20L?**
   - **No**
     - **River pump off**
   - **Yes**
     - **Buzzer Off**
6. **Is water level 15L?**
   - **No**
     - **Display Parameters on LCD**
     - **Sprinkler pump On**
   - **Yes**
     - **Sprinkler pump Off**
7. **Stop**
IV. System Testing And Result

Test Plan

Testing was done to ascertain the performance of the sub-circuits and then whole system degree of accuracy as well as the reliability. The testing of this automated irrigation control system was carried out in sequential manner starting from the individual component to the sub-circuits and finally the whole system as mentioned above.

This was done to know whether the system is performing well or not, and if the design specifications conform with the systems operations.

Result of Simulation Using Proteus Virtual Simulation Module

![Fig. 11: Simulation Result on Proteus VSM via Virtual Terminal](image1)

![Fig. 12: Simulation Result on Proteus VSM Virtual Terminal showing the logged Data.](image2)

![Fig. 13: Simulation Result on Proteus VSM via LCD](image3)

The actual implemented hardware result is shown in figure 13 and the Proteus simulated result talleys. For the first time it is possible to draw a complete circuit for a micro-controller based system and then test it interactively, all from within the same piece of software.

V. Conclusion

The process control design meets all of the objectives set forth while satisfying the constraints. The step by step processes in the design of a microcontroller-based irrigation control has been presented in this project.

Atmega32 microcontroller is programmed to automate irrigation process and simulated using Proteus VSM and the result are satisfactorily executed and verified with physical prototype.

It is also important to mention that the entire system was implemented using readily available components and no formal training is necessary to operate the system for past users of manual irrigation. This project particularly is significant in view of the fact that our nation Nigeria is at the moment of commercializing agricultural activities which automated irrigation is a key to its success.
References


[7]. Clark, Gary; Rogers, Danny; Alum, Mahbub; Fjell, Dale and Briggeman, Steven (2013); “A Mobile Irrigation Lab For Water Conservation: Physical and Electronic Tools”, Available online at www.kser.com


