Automatic Urea Fertilizer Application Machine And Irrigation System

Nahian Sourav*, Dabarjun Mazumdar†, Fardin Hossain‡ and Shagufta Sajid§
Dept. of Comuter Scienece and Engineering
United International University, Dhaka, Bangladesh
011201013*, 011201016†, 011201017‡ and 011201460§

***Abstract*—Our project can be a best invention for a farmer. Although we have a lot of machines for Urea fertilization and irrigation system but it is an automatic machine which can do Urea fertilization and irrigation at the same time. With the help of this machine, farmers can know the moisture of the soil and the machine will supply water in soil by its own when it is needed. we know that too much water in the soil is harmful for the plants and our machine can detect if there is too much water in the soil or in plants root. Then it can also extract extra water from plants root or soil and restore them. By this way farmers can save their plants and save water too. And it is totally an automated process which is not expensive at all to implement. So it can save our farmers time, save the plants and save water through irrigation system.**

***Index Terms*—IOT, Arduino, Sonar, GPS**

I. Project Overview

We use 2 separate Arduino to complete our project. Arduino-1 is for the mechanism of fertilization and Arduino-2 is for the mechanism water supply, soil moisture detection and extra water restoring from soil. So here is the overview of our full project.

- Flowchart Of Fertilizer Giving Process Mechanism:

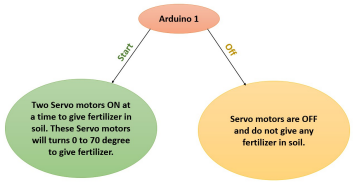

Fig. 1. Mechanism-1

- Flowchart Of Soil Moisture Detecting and Water Giving And Storing Process:

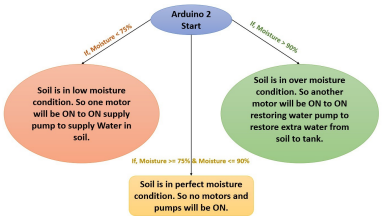

Fig. 2. Mechanism-2

II. Component List

1) Arduino
2) Motor/Pump
3) Servo motor
4) Soil moisture sensor
5) I2C driver
6) Battery
7) 16x2 LCD display
8) Wheels
9) L293D motor driver
10) Jumper wires
11) Breadboard

A. Arduino

Arduino UNO is the ideal board for getting started with electronics, through fun and engaging hands-on projects. As the board can be easily connected to the other computer system via USB port. We used to supply the power supply to the board and can act as a serial device to connect the board to a computer system. The major components of Arduino UNO board are USB connector, power port, Microcontroller, Analog input pins, Digital pins, Reset switch, Crystal oscillator, USB interface chip. In our Project we use 2 Arduino UNO.

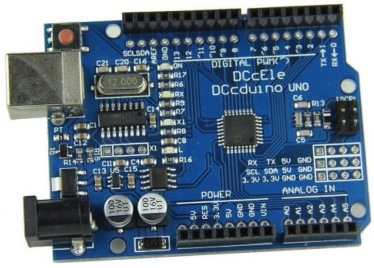

Fig. 3. Arduino

B. Motor/Pump

An electric motor is an electrical machine that converts electrical energy into mechanical energy. Most electric motors operate through the interaction between the motor's magnetic

field and electric current in a wire winding to generate force in the form of torque applied on the motor's shaft. Electric motors can be powered by direct current (DC) sources, such as from batteries, or rectifiers, or by alternating current (AC) sources, such as a power grid, inverters or electrical generators. An electric generator is mechanically identical to an electric motor, but operates with a reversed flow of power, converting mechanical energy into electrical energy.In our project we use 2 motors to supply water in our plants or soil and restoring extra water from soil.

Fig. 4. Motor

C. Servo motor

A servomotor (or servo motor) is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. We use servo motor in our project to complete our fertilization process. We rotate our servo motors 0 to 70 degree to complete this process.

Fig. 5. Servo Motor

D. Soil moisture sensor

Soil moisture sensor measure the volumetric water content in soil. Since the direct gravimetric measurement of free-soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content. In our project we use this sensor to detect the actual moisture of soil.

Fig. 6. Soil Moisture Sensor

E. I2C driver

This an interface. To show data on LCD display we need to connect this interface. Without this we cannot see out data on LCD display. The I2C driver's pin will be connect without Arduino's ground pin and voltage supply pin.

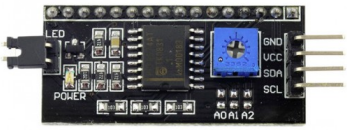

Fig. 7. I2C Driver

F. Battery

This is one-directional flow of electric charge. An electrochemical cell is a prime example of DC power. Direct current may flow through a conductor such as a wire, but can also flow through semiconductors, insulators, or even through a vacuum as in electron or ion beams. In our project the battery will actually supply us full power. The power coming from the battery will run our whole project and we will have current supply in the whole project.

G. 16x2 LCD display

This is a liquid crystal display. There are liquid crystal materials sandwiched between two sheets of glass. It works by using liquid crystals to produce an image. In our project it will shows us the moisture of soil. That's why use an LCD display in our project.

Fig. 8. Battery

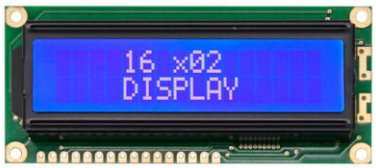

Fig. 9. LCD Display(16x2)

H. Wheels

Wheels are an important part of our project. We will be implementing our project on a hardboard and a lot of equipment will be used there. We will need to take our project from one place to another. For that reason we will need wheels. Wheels are the only thing which we can use to take our project from one place to another. So that's how we will be using wheels in our project

Fig. 10. Wheel

I. L293D motor driver

The L293D is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors or single stepper motor. It can drive 4 bi-directional DC motors with 8-bit speed selection(0-255), 2 stepper motors (uni polar or bipolar) with single coil, double coil, interleaved or micro-stepping and 2 servo motors. In our project we use it to connect our supply motor and restoring motor with Arduino.

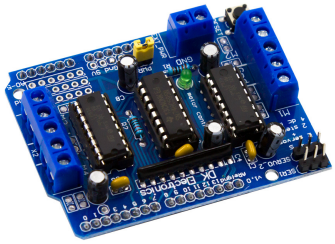

Fig. 11. L293D Motor Driver

J. Jumper wires

Jumper wires are used for making connections between items on my breadboard and Arduino's header pins. Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper acts as a switch by closing or opening an electrical circuit. Jumpers can be added or removed to change the function or performance of a PC component.

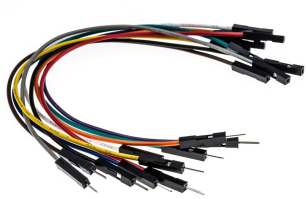

Fig. 12. Jumper Wires

K. Breadboard

A breadboard is used to build and test circuits quickly before finalizing any circuit design. A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode). In our project breadboard will be connect LCD display and I2C driver together. To show the mechanisms of every part of the experiment we need to connect all of the parts together. Breadboard will be used here as that equipment which will connect all other equipment together so that we can work on our project effortlessly.

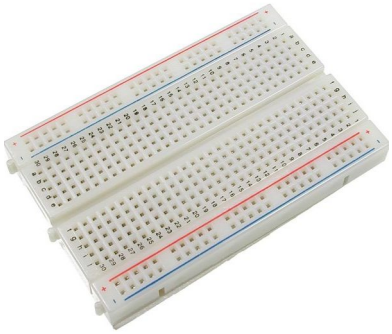

Fig. 13. Breadboard

III. Implementation

- Fertilization Process:

Connection between Servo motor and Arduino-1

Upper Servo motor pins	Arduino-1 pins	Lower Servo motor pins	Arduino-1 pins
Upper pin	Voltage source/ supply pin	Upper pin	Voltage source/ supply pin
Middle pin	9	Middle pin	10
Lower pin	Ground	Lower pin	Ground

Fig. 14. Connections

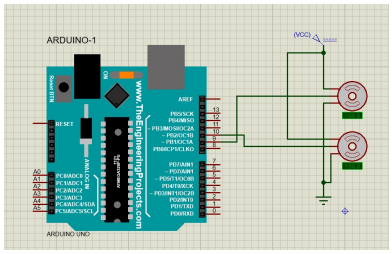

Fig. 15. Giving Fertilizer System

```
//Fertilizing system for our project using servo motors
#include <Servo.h>

Servo myservo;  // create servo object to control a servo no 1
Servo mys;      // create servo object to control a servo no 2

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo no 1 on pin 9 to the servo object
  mys.attach(10);     // attaches the servo no 2 on pin 10 to the servo object
}
void loop() {
  for (pos = 0; pos <= 70; pos += 1) { // goes from 0 degrees to 70 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo no 1 to go to position in variable 'pos'
    delay(5);                        // waits 5 ms for the servo no 1 to reach the position
    mys.write(pos);                 // tell servo no 2 to go to position in variable 'pos'
    delay(5);                        // waits 5 ms for the servo no 2 to reach the position
  }
  for (pos = 70; pos >= 0; pos -= 1) { // goes from 70 degrees to 0 degrees
    myservo.write(pos);              // tell servo no 1 to go to position in variable 'pos'
    delay(5);                        // waits 5 ms for the servo no 1 to reach the position
    mys.write(pos);                 // tell servo no 2 to go to position in variable 'pos'
    delay(5);                        // waits 5 ms for the servo no 2 to reach the position
  }
}
```

Fig. 16. Code Of Giving Fertilizer System

- Soil Moisture Detection,Water Supply And Restoring Process:

Connection between Soil moisture sensor and Arduino-2

Sensor pins	Arduino-2 pins
A0	A0
GND	Ground
VCC	Voltage source/ supply pin

Connection between LCD display and Arduino-2

LCD Display Pins	Arduino-2 pins
VDD	Voltage source/ supply pin
SCL	A5
SDA	A4
VSS	Ground

Connection between Motors and L293D

Supply motor pins	L293D pins	Restoring motor pins	L293D pins
Upper pin	3	Upper pin	11
Lower pin	6	Lower pin	14

Connection between L293D and Arduino-2

L293D pins	Arduino-2 pins
2	5
7	6
1	11
9	12
10	9
15	10
VSS (16) & VS (8)	Voltage source/ supply pin
GND & GND	Ground

Fig. 17. Connections

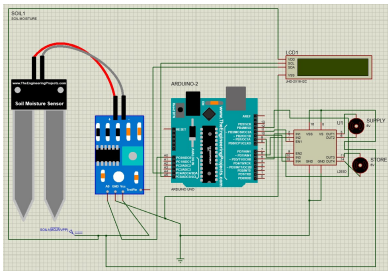

Fig. 18. Soil Moisture Detection,Giving Water And Restoring Water System

```
#include<Wire.h>
#include "DFRobot_LCD.h"

int soil;
int m;
int EN1=11;
int EN2=12;
DFRobot_LCD lcd(16,2);
void setup() {
  lcd.init(); // LCD display initially

  pinMode(EN1,OUTPUT);    // for motor 1 suuply water
  pinMode(EN2,OUTPUT);    // for motor 2 to store extra water from soil

  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
}
void loop() {

  soil=analogRead(A0);      //analog output from soil moisture sensor
  m= map(soil,0,1023,100,0);  // convert analog output in between (0 to 100%) moisture
 //m=80;
 // m=95;
 //m=50;
  lcd.print("MOISTURE = ");
  lcd.print(m);    // showing moisture value of soil in between (0 to 100%) in LCD display
  lcd.print("%");
  if(m<75)              // if moisture is less then 75%
  {
      lcd.setCursor(0,1);
      lcd.print("Supplying Water  ");
      digitalWrite(EN1,HIGH);     // start water supply motor to start water supply pump
      for1();
  }
 else if(m>90)        //   if moisture is greater then 90%
 {
   lcd.setCursor(0,1);
   lcd.print("Restoring Water  ");
   digitalWrite(EN2,HIGH);   //the motor start which will help another pump to store extra water from soil
   for2();
 }
 else     // moisture is in between 75 to 90
 {
   lcd.setCursor(0,1);
   lcd.print("Perfect Moisture   ");
   digitalWrite(EN1,LOW);   // water supply motor off
   digitalWrite(EN2,LOW);   // water restore motor off
 }
 delay(500);
}
    void for1() // supply motor forwarding
    {
      analogWrite(5,100);
      analogWrite(6,0);
    }
    void for2()  // restoring motor Backwarding
    {
      analogWrite(9,0);
      analogWrite(10,100);
    }
```

Fig. 19. Code Of Detecting Soil Moisture, Giving Water And Restoring Water System