Introduction to IoT and Its Industrial Applications (CS698T)
Indian Institute of Technology Kanpur
Assignment 2

QUESTION 1

Member Names: Ankit Raja, Anuj Shrivastava
Member Emails: ankitr21@iitk.ac.in, anujs21@iitk.ac.in
Member Roll Numbers: 21111012, 21111013
Date: November 3, 2021

Abstract

As we know that around 70% of people in India depends on agriculture and a major percentage of our GDP comes from agriculture. But the methods of farming used in India are traditional methods which are operated manually. The basic problem with traditional methods is that two scare and valuable resources that is water and energy are not utilised properly. And In today's world, due to many environmental reasons, the farmers are dealing with shortage of rain and inadequacy of water. Hence, we require a modern solution for this problem, so that we can strengthen our agriculture sector.

1 Introduction

India is country where around 70% of the people depend on agriculture, out of which only few farmers uses advance agricultural methods while all other uses the same old traditional methods only. Hence methods of farming needs to improved. We want to improve the yields from the farmland of farmers so we have built an "Intelligent irrigation system" that waters the farmland automatically based on present humidity and temperature levels in the farm.

So, Our Problem Statement which is "Aravind is an ambitious farmer who wants to improve the yields from his farmland. While looking for ways to improve the yields, he came across tech buzzwords like the Internet of Things and Machine Learning. He wants to try them out to build an intelligent irrigation system for his field. The irrigation system he wants to build contains servo motors that control the water supply to the farm, based on the present humidity, temperature levels in the farm. Using sensor data, Aravind wants to build a machine learning model that predicts how much water (in percentage) should be supplied to his farm. Aravind also does not want to water the plants during nighttime."

Therefore, as per Aravind's requirement irrigation system that we built contains 4 sensors in such a way that each sensor can sense humidity and temperature level. These values are then provided to a ML model on the IOT board to predict the water flow in percentage. After this, the water flow percentage will be provided as input to the servo motor which will then map these values with the servo motor rotation (0 degree to 180 degree). The proposed system makes use of microcontroller Arduino Mega and IOT which enables Aravind to monitor moisture and temperature levels. One another thing is that Aravind also doesn't want to water the plants during the night time. So, We'll be using LDR sensor to detect day/night and at the night time all the servo motors will automatically come at 0 degree which in turn means the water flow at night time will be 0% .

2 Intelligent irrigation system

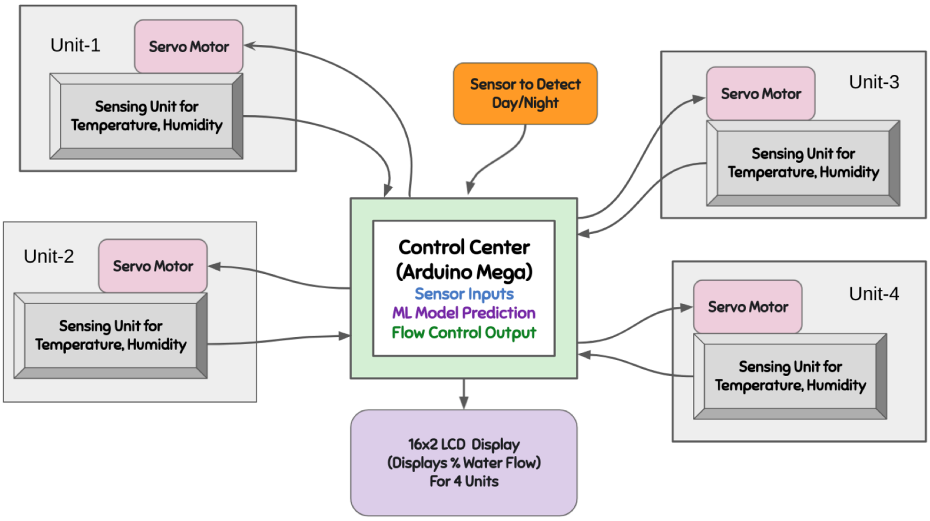

Figure 1: Block diagram of proposed Irrigation System

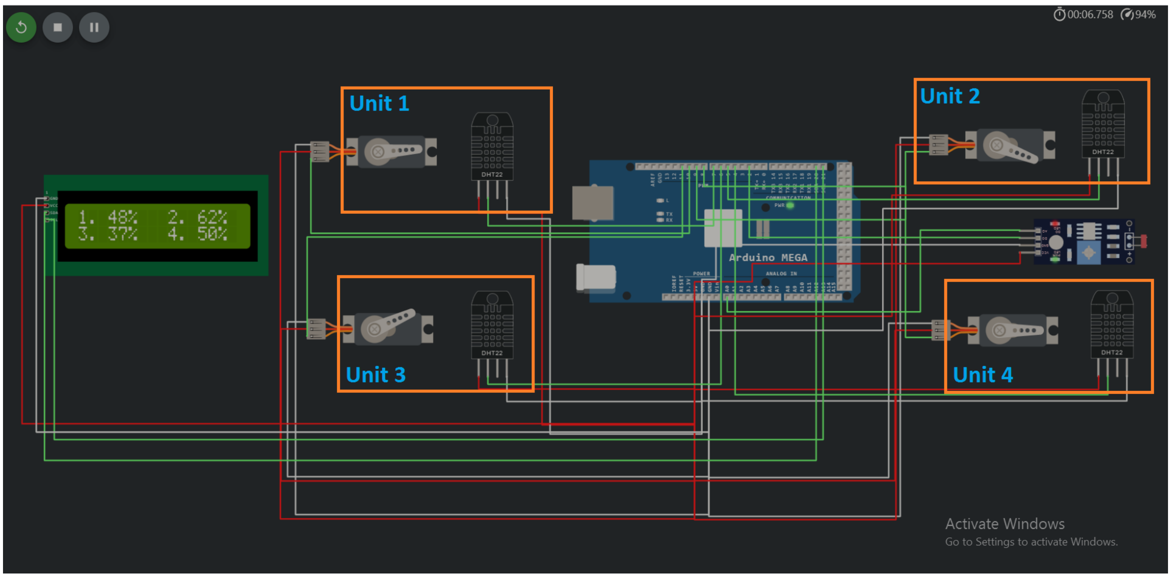

Figure 2: Our Irrigation System circuit

3 Hardware Components used

3.1 Arduino Mega

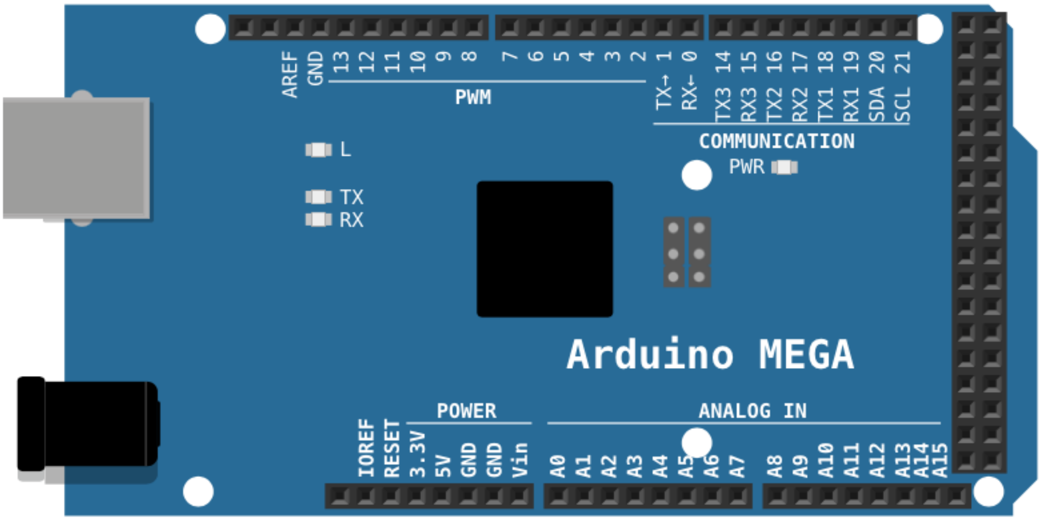

Figure 3: Arduino Mega

Powered by the ATmega2560 chip, Arduino Mega 2560 comes with a 256 KB of Flash program memory, 4KB of EEPROM and 8KB of SRAM. This board has 54 digital pins, 16 analog input pins, & 4 serial ports. It runs at a frequency of 16Mega HZ.

3.2 DHT22 Sensor

The DHT22 is a basic, low-cost digital temperature and humidity sensor. This Sensor is used to get the soil moisture levels and temperature. Soil moisture sensors are used for measuring the water content of the soil. They basically use capacitance to measure dielectric permittivity of the surrounding medium in soil, dielectric permittivity is a function of the water content. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

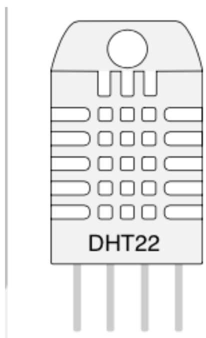

Figure 4: DHT22 Sensor

3.3 LDR sensor

The LDR (light-dependant resistor) sensor, also known as photoresistor, is bascically a photocell that works on the principle of photoconductivity. The passive component of this sensor is actually a resistor whose resistance value decreases when the intensity of light decreases.

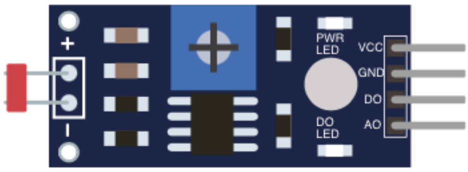

Figure 5: LDR sensor

3.4 Servo Motor

Servo motors are electronic devices and rotary or linear actuators that rotate and push parts of a machine with precision.

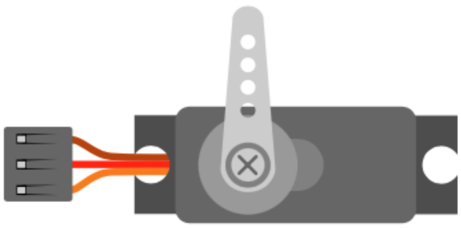

Figure 6: Servo Motor

3.5 LCD display

The LCD which we are using is LCD1602 which uses the Hitachi HD44780 LCD Controller chip. The chip comes with a built-in font, as well as, it has the ability to define up to 8 custom characters. It will display the % water flow for the the 4 units.

Figure 7: 16x2 LCD

4 How the hardware is connected

There are 4 units in the system we built, each unit contains a servo motor and a DHT22 sensor. For 1^{st} unit servo motor is connected at 10^{th} GPIO pin and DHT22 sensor is connected 7^{th} GPIO pin of Arduino Mega board. Similarly for 2^{nd} unit they are connected at 5^{th} and 9^{th} GPIO pins, for 3^{rd} unit they are connected at 6^{th} and 11^{th} GPIO pins and for 4^{th} unit they are connected at 4^{th} and 8^{th} GPIO pins respectivley.

For detecting day/night we are using a LDR sensor which is connected at "A0" analogue pin of Arduino board. Since LDR sensor will give analogue output that's why we connected it at analogue pin of arduino board.

For displaying water percent we are using a LCD display whose SDA and SLC pins are connected at respective SDA and SLC pins of ardiono board.

For giving power to sensors their VCC and GND pins are connected at arduino's VCC and GND pins respectively. So this is how the connection is done.

5 How we built this irrigation system

After looking at the connection part now comes the actual working that how this whole system is integrated and how this system as a whole work together.

So we kept code in a loop where every time sensors values at all the four units are read and what water flow percent should be there at each unit is predicted using the ML model we built. After getting the water percentage value, this value is fed to a function named "change water flow" along with the unit number. So with the help of water percentage value, "change water flow" function moves the servo motor with that much percentage value for the respective unit number.
Also we are showing the latest updated water percentage value at each unit on the LCD screen we are using.

For integrating the ML model along with our arduino circuit we made a header file named

"Predict Water Percentage.h" in which we coded the ML model in C++ language with the help of weights and bias we got after the model training. So we made a function inside this header file for predicting water flow percentage and we are calling this function from our sketch.ino file. So this is how the whole system as a whole work together.

For training the ML model that can predict the water percentage we have used a dataset with 200 data inputs. We split the dataset into training and testing data in 80:20 ratio. Then we made a sequential model with two hidden layers to train our model using training data. Once model is trained we predicted the water percentage value for the testing data and calculated RMSE and R2 Score for testing data.
Inorder to integrate our ML model with the circuit board we implemented a forward pass using weights and biases from the model. In forward pass it takes current temperature and humidity as input and use model weights and biases to predict optimal water percentage. Then this predicted water percentage is displayed on the LCD screen.

6 ML model Architecture

We have used sequential model to build this irrigation system because a sequential model is appropriate for:

- a plain stack of layers where each layer has exactly one input tensor and one output tensor.
- model that doesn't have multiple outputs.

In this sequential model we have a input layer, two hidden layers and a output layer. In input layer we have two inputs as humidity and temperature value. In hidden layer1 we have 8 nodes each with ReLU activation function and for hidden layer2 again we have 8 nodes each with ReLU activation function. In our output layer we have sigmoid activation function. Here is our sequential model with total 105 parameters to be learned.

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 8)                 24
_________________________________________________________________
dense_19 (Dense)             (None, 8)                 72
_________________________________________________________________
dense_20 (Dense)             (None, 1)                 9
=================================================================
Total params: 105
Trainable params: 105
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 8: Our sequential model

ReLU and Sigmoid activation function:

We have used ReLU activation function at both hidden layers and Sigmoid activation function at output layers. The rectified linear activation function(ReLU) is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

Here is ReLU activation function:

$$f(x) = max(0, z) = \begin{cases} 0 & \text{if } z \leq 0 \\ z & \text{otherwise} \end{cases}$$

The sigmoid function is a special form of the logistic function and is usually denoted by $\sigma(x)$ or sig(x). We have use activation function for output layer as a sigmoid function because it guarantees that the output of this unit will always be between 0 and 1.

Here is Sigmoid activation function:

$$\sigma(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$

Model details:

- Train-Test split ratio: 80:20
- Optimizer used: Adam
- Loss: Mean Squared Error
- No. of Epochs: 10,000
- Batch Size: 32
- Performance metric: Root Mean Squared Error (RMSE), R2 Score
  RMSE of test dataset is 19.259767
  R2 Score of test dataset is 74.35790

7 Instructions to run the ML model

1. You must have following dependencies on your machine to run this model: Jupyter Notebook, Python3, Tensorflow, Keras, Numpy, Matplot, Sklearn
2. Open IOT_Assign2.ipynb file in jupyter notebook.
3. Inorder to reproduce the same results use the following seed number at the top of ML model file(already added): $seed(1)$
   $random.set\_seed(2)$
4. Run the code, training the model will take few minutes.
5. RMSE, Weights and biases of each layer gets printed.

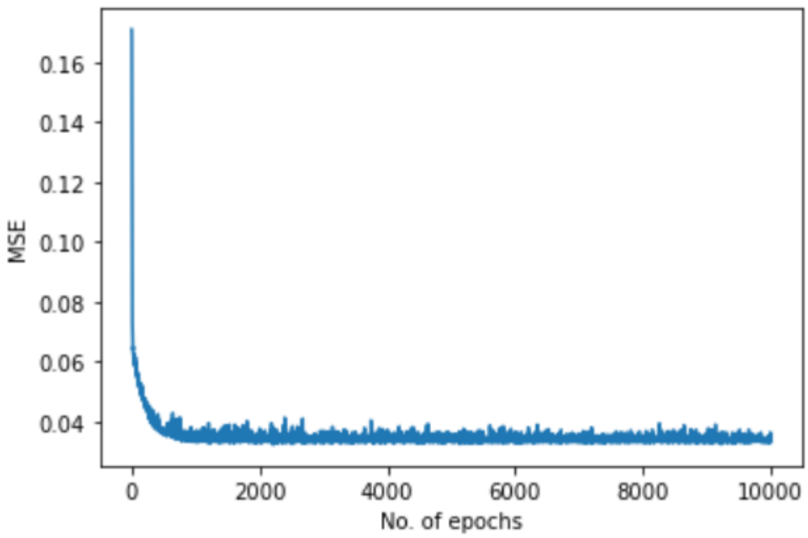

Figure 9: MSE value for each Epoch

8 Advantages

This modern solution of the traditional irrigation problem is very efficient. As the soil moisture level is measured, we can provide water as per requirement, which in result prevents water clogging of soil.
Also, as valves are under our system's control, we therefore doesn't require labour.
Another major advantage of this model is that user can handle , check the moisture levels even if he is at opposite point of the globe through mobile.
This system proves to be a cost effective solution for our problem and is proficient in conserving water and reducing its wastage.

9 Summary

So summarising in brief we say that this system reads temperature and humidity values and from those values it predicts appropriate water flow percentage which in turn is used for controlling servo motors according to the water flow percentage value which helps in feeding appropriate amount of water according to land requirements.

This proposed system can control the water flow depending on the soil moisture and temperature, thereby automating the whole process of irrigation. This IOT system uses information collected from soil moisture sensor and humidity sensor to irrigate soil which prevents clogging of soil, which in result prevents crop damage. From this IOT system, we can conclude that it is a considerable development in farming which is great for building an Intelligent Irrigation system.

Below we have showed output for two testcases. First output is for day time and optimal water percentage is displayed on the LCD screen. Second output is for night time when all servo motor water percentage is set to 0 and same is displayed on the LCD screen. For detailed information you can visit below project link.

Link for wokwi project:
https://wokwi.com/arduino/projects/313999672039440960

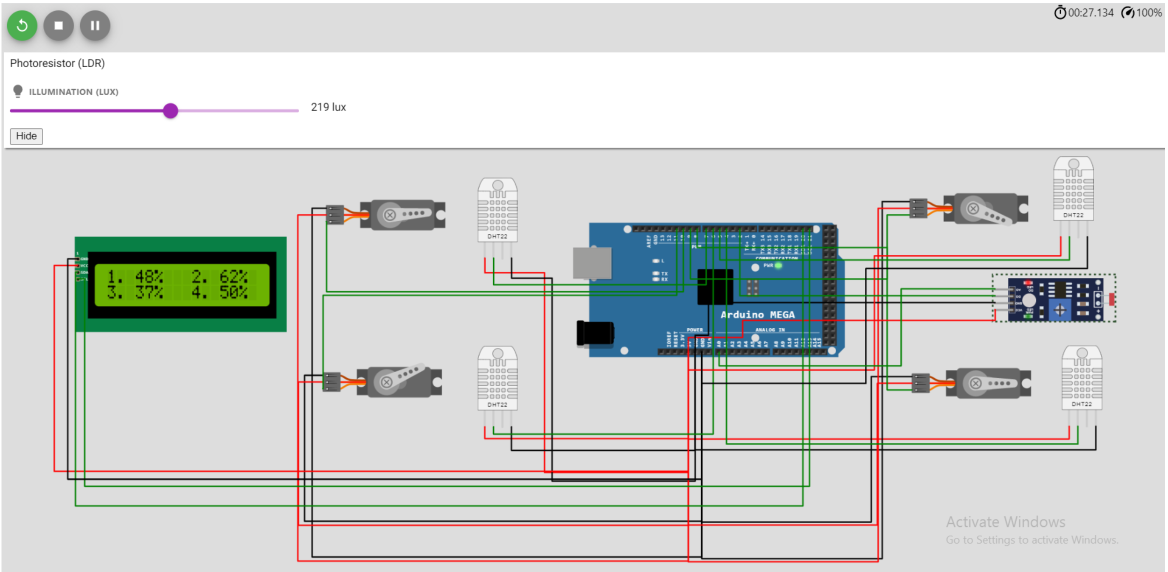

Figure 10: Our Irrigation System during Day time

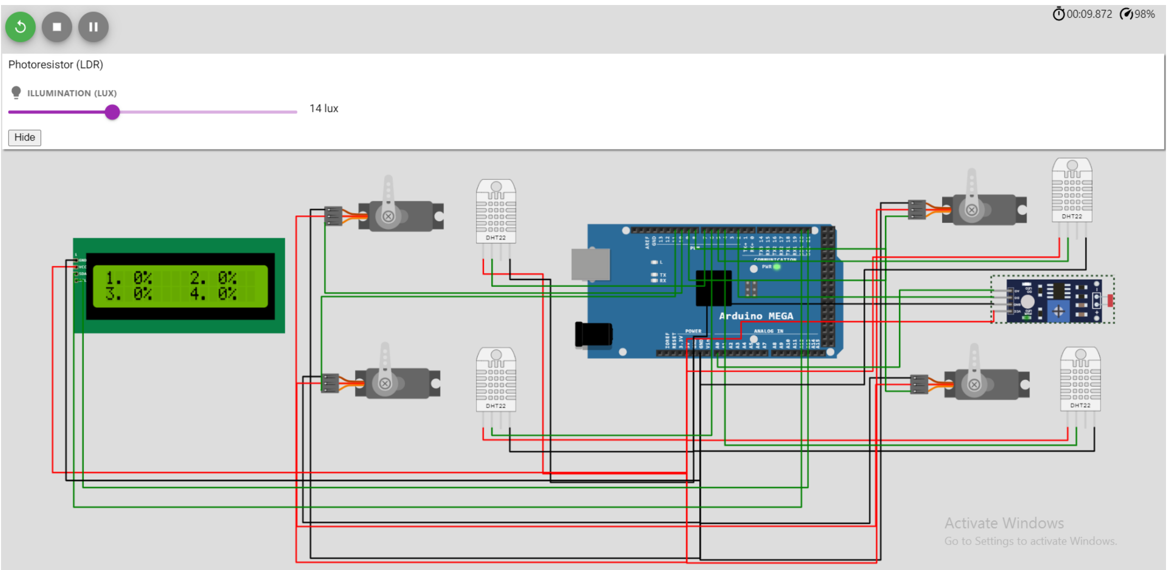

Figure 11: Our Irrigation System during Night time