REALTIME WEATHER BASED SMART SPRINKLER SYSTEM FOR GOLF COURSE

INTRODUCTION

The importance of building an automation system for an office home or field is increasing day-by-day. Automation makes an efficient use of the electricity, water and reduces much of the wastage. Smart water sprinkler irrigation system makes an efficient use of water for the growth of plants. Heart of the system is Raspberry Pi 3 mini computer, shown in figure 1. Raspberry Pi model 3 has dedicated general purpose input outputs (GPIO) pins. These all GPIO pins can be accessed for controlling hardware such as LEDs, sensors, and relays, which are examples of outputs. Need of automatic Irrigation  Simpler and easy to install and configure.  Saving energy and resources, so that it can be utilized in appropriate way.

LITREATURE REVIEW

After extensive research in the agricultural field, many researchers found that the agriculture area and its productivity are decreasing by the day. With the Use of different technology in the field of agriculture we can increase the production as well as reduce manual efforts. This paper shows the sprinkler enabling techniques, protocols and architecture for sprinkler which is widely used for agricultural, gardening, home and office purpose. Chandankumar Sahu et al. proposed a system on “A Low Cost Smart Irrigation Control System”. It includes a number of wireless sensors which are placed in different directions of the farm field. Each sensor is integrated with a wireless networking device and the data received by the “ATMEGA318” microcontroller which is on the “ARDUINO-UNO” development board. The Raspberry pi is used to send various types of data like text messages and images through internet communication to the microcontroller process

THEORETICAL ANALYSIS

### 1. Sensors

#### (a) Soil moisture sensor

The soil moisture sensor is used to measure the volumetric water content of soil. It is used to monitor soil moisture content to control irrigation in greenhouses. A moisture sensor is used to sense the level of moisture content present in irrigation field. It has a level detection module in which we can set a reference value.

#### (b) Temperature and Humidity Sensor

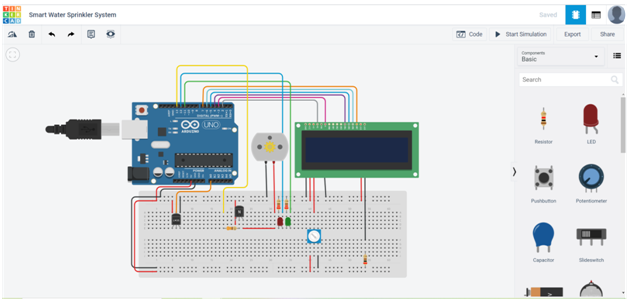
The temperature and humidity sensor is necessary to reduce the watering frequency. That is when the weather gets cooler, less water is needed whereas vice versa in the other case.

### 2. Relay Module

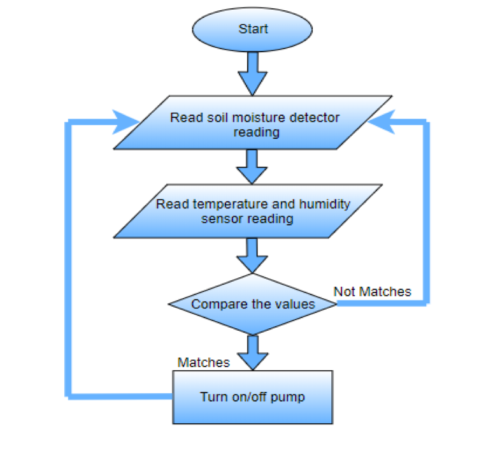
The relay module is an electrically operated switch that allows you to turn ON or OFF a circuit using voltage and/or current much higher than a Microcontroller could handle. There is no connection between the low voltage circuit operated by the Microcontroller and the high power circuit. The relay protects each circuit from the other. Each channel in the module has three connections named NC, COM, and NO. Depending on the input signal trigger mode, the jumper cap can be placed at high level effective mode which ‘closes’ the normally open (NO) switch at high level input and at low level effective mode which operates the same but at low level input.

### 3. Peristaltic Pump

A peristaltic pump is a type of positive displacement pump used for pumping a variety of fluids. The fluid is contained within a flexible tube fitted inside a circular pump casing. It is reputed to pump water from a depth of about 31 feet.

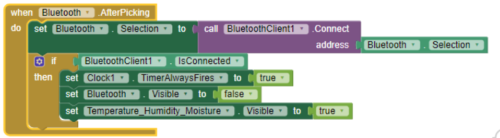


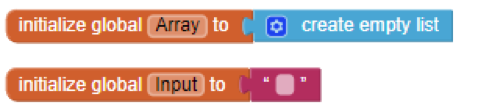
FLOW CHART



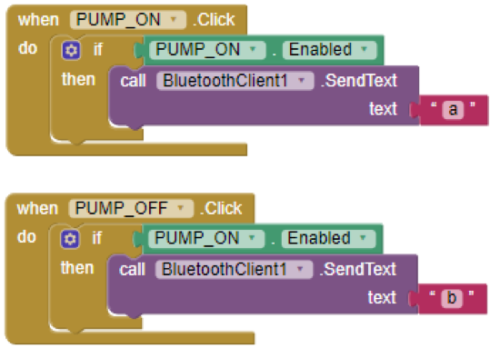
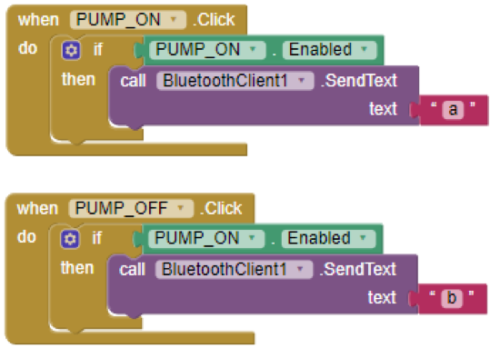
## App Blocks: (MIT App Inventor)











Code :

mport time  
import sys  
import ibmiotf.application  
import ibmiotf.device  
import random  
#Provide your IBM Watson Device Credentials  
organization = "sj98sz"  
deviceType = "raspberrypi"  
deviceId = "123456"  
authMethod = "token"  
authToken = "12345678"  
  
  
def myCommandCallback(cmd):  
 print("Command received: %s" % cmd.data)#Commands  
   
  
try:  
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}  
 deviceCli = ibmiotf.device.Client(deviceOptions)  
 #..............................................  
   
except Exception as e:  
 print("Caught exception connecting device: %s" % str(e))  
 sys.exit()  
  
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times  
deviceCli.connect()  
  
while True:  
   
 hum=random.randint(10, 40)  
 #print(hum)  
 temp =random.randint(30, 80)  
 #Send Temperature & Humidity to IBM Watson  
 data = { 'Temperature' : temp, 'Humidity': hum }  
 #print (data)  
 def myOnPublishCallback():  
 print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % hum, "to IBM Watson")  
  
 success = deviceCli.publishEvent("Weather", "json", data, qos=0, on\_publish=myOnPublishCallback)  
 if not success:  
 print("Not connected to IoTF")  
 time.sleep(2)  
   
 deviceCli.commandCallback = myCommandCallback  
  
# Disconnect the device and application from the cloud  
deviceCli.disconnect()

Advantages:

* Reduces Damage. Sprinkler systems reduce property damage. ...
* Low **Costs**. Sprinkler systems are affordable to install and maintain. ...
* Low **Maintenance**. Annual inspections are about the only **maintenance** that sprinkler systems require. ...
* Save Lives. ...
* Fire Prevention and Protection Services from Advanced Fire Protection Systems.

Disadvantages:

* High initial cost.
* The water must be clean and free of sand, debris and dissolve salts.
* Cannot be used in windy climate.
* Fruits/crops can be damaged due to excessive water.
* Requires high and continuous power supply.

Sources

Google