SMART WATER MANAGEMENT USING IBM WATSON PLATFORM

## ABSTRACT:

The Internet, invention of the century, has completely revolutionized the world and brought people closer to each other than ever before. The advancement in technologies of computing, communication brings the next generation of Internet, Internet of Things. As the population and urbanization increases, the cities have to transform to Smart Cities which can be achieved with the help of Internet of Things. Water is one of the vital resource for existence of human life and so Smart water management system has a key role in smart city. The paper reviewed different technologies and platforms that are required for a smart environment. An architecture design for Smart water management is proposed and an implementation detail of Smart water monitoring system is discussed.

INTRODUCTION:

This paper presents an IOT device which help to manage and plan the usage of water. This system can be easily installed in residential societies. Sensors placed in the tank which continuously informs the water level and flow rate at the current time. This information will be updated on the cloud and using an android application, user can visualize the water level and flow rate on a Smartphone anywhere that is connected to Internet and can also be intimated before realease of water. According to the level of water in the tank the motor functioning will be automatically controlled, at low level of water motor will automatically turn on and when tank is about to fill up it will cut off.

Problem statement:

Water is the most widespread and important natural resource which support the life on Earth. Oceans hold 97% of Earth’s surface as saline water and the remaining 3% only occurs as fresh water, of this 2.4% is frozen in glaciers and polar ice caps and the rest 0.6% is in liquid fresh water forms and available in rivers, lakes, and ground water etc. (Figure 1.1). About 22% of liquid fresh water exists as ground water, which constitutes about 97% of all liquid fresh water available for human use which represents the availability of ground water is meager in Earth’s total global water content (Foster, 1998). Hence in most of the countries, the ground water is the major source of potable water. It is also widely used for agriculture and industrial purposes in several nations. The availability of ground water has great influence on human life as well as other forms of life. Ground water is an important renewable natural resource of socio-ecological significance. Particularly, ground water is emerging as a formidable poverty-alleviation tool, which can be delivered direct to poor community more cheaply, quickly and easily than canal water (IWMI, 2001).

Project working process:

Smart water management has many advantages.Here few sensors like ultrasonic sensor and flow sensors are used to sense the values of level of the water in a tank and flow sensor is used to sense the rate of flow of water. Even additional sensors can also be interfaced and measure the respective parameters like turbidity and purity of water etc.

In this project we used aurdino and nodemcu boards got interfaced with the sensors. The sensors like ultrasonic and flow sensor have been interfaced with arduino and the values from sensors be recorded or stored onto Arduino using ultrasonic, flowsensors codes respesctively.

As the Arduino doesnot have any wifi module to push the values of waterdepth and flowrate we will use nodemcu which has wifi module along with arduino .

The Arduino and nodemcu communicate serially through rxd and txd pins.the aurdino sends the values of water depth and flowrate to the nodemcu.The nodemcu with the written code using wifi module sends values to ibm cloud.

In ibm cloud we use watsonplatform and nodered applications.In Watson platform through device credintials ,we login and the gauges display the values visually.In nodered through flow we create use interface and use that link in MIT app inventor and use the application at user end to display the values of waterdepth and flowrate and get a intimation that water has started or not.

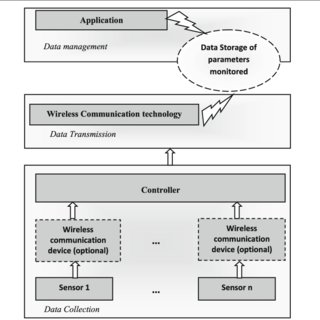
DESIGN OF SMART WATER MANAGEMENT:

Smart Water Management (SWM) uses Information and Communication Technology (ICT) and real-time data and responses as an integral part of the solution for water management challenges. SWM is becoming an area of increasing interest as governments from around the world integrate smart principles into their urban, regional and national strategies. The potential application of smart systems in water management is wide and includes solutions for water quality, water quantity, efficient irrigation, leaks, pressure and flow, floods, droughts and much more.

By applying SWM infrastructure such as sensors, smart meters, monitors, GIS and satellite mapping, and other data sharing tools to water management, real-time solutions can be implemented and broader networks can work together to reduce current water management challenges.

**Objectives of the SWM Project**

• Promote the use of SWM for current water challenges  
• Showcase and provide insights from exemplary cases of SWM from around the world  
• Support future SWM projects by highlighting the enablers and barriers for SWM to decision makers  
• Identify the contribution SWM can offer in achieving the SDGs



Components used:

HARDWARE COMPONENTS:

1.ultrasonic sensor.

2.flow sensor.

3.arduino.

4.Nodemcu(ESP8266).

5.Jumpers(as required).

6.USB cables for Arduino and nodemcu.

SOFTWARE :

1.Arduino IDE.

2.IBM cloud Watson platform.

Images of the components we have used:





ULTRASONIC SENSOR:

This has 4 pins namely:

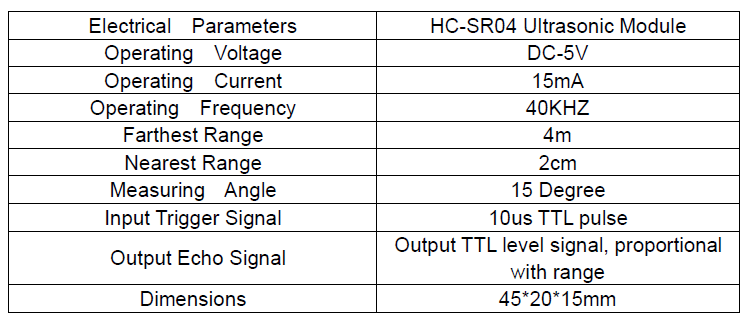
1.vcc(5v).

2.gnd.

3.echo.(10th pin for Arduino)

4.trigger.(9th pin for Arduino)

It generally works on principle of wave reflection where from a source triggered emits ultrasonic waves this waves get reflected when it reaches or touches an obstacle.based on this reflection velocity of wave the time in which arrived to sensor it indirectly calculates distance between an obstacle.

Datasheet of an ultrasonic sensor:

FLOWSENSOR:

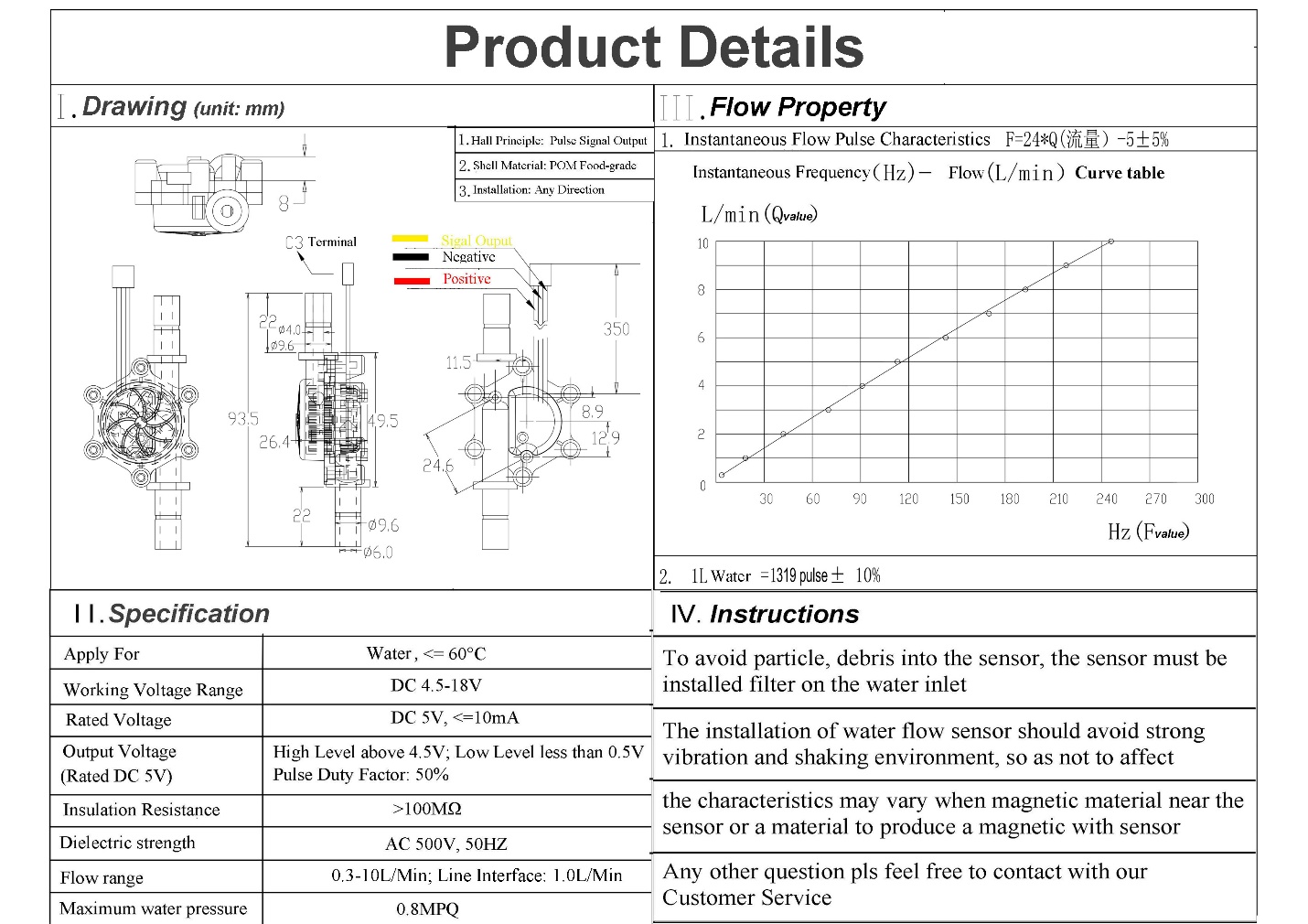
Flow sensor has only 3 pins namely:

1.vcc(5v).

2.ground(0v)

3.input(2nd pin for arduino)

This generally has a mini turbine or a fan like structure which is actually enclosed in a magnetic field when a water or a blow comes into an sensor fan rotates and cuts the magnetic field which induces voltage according to the faradays law of electromagnetic induction.this voltage induced is proportional to the quantity of water entered into sensor.



Code for SWM:

For Arduino to sense flowrate and waterdepth:

#include <SoftwareSerial.h>

SoftwareSerial myarduino(5,6); // rx-5, tx-6

const int trigPin = 9;

const int echoPin = 10;

// defines variables

long duration;

int distance;

volatile int flow\_frequency; // Measures flow meter pulses

unsigned int l\_hour; // Calculated litres/hour

unsigned char flowmeter = 3; // Flow Meter Pin number

unsigned long currentTime;

unsigned long cloopTime;

int s;

void flow () // Interruot function

{

flow\_frequency++;

}

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

Serial.begin(9600); // Starts the serial communication

myarduino.begin(9600);

pinMode(flowmeter, INPUT);

digitalWrite(flowmeter, HIGH);

Serial.begin(9600);

attachInterrupt(0, flow, RISING); // Setup Interrupt

// see http://arduino.cc/en/Reference/attachInterrupt

sei(); // Enable interrupts

currentTime = millis();

cloopTime = currentTime;

}

void loop() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance= duration\*0.034/2;

// Prints the distance on the Serial Monitor

Serial.print("Distance: ");

Serial.println(distance);

currentTime = millis();

// Every second, calculate and print litres/hour

if(currentTime >= (cloopTime + 1000))

{

cloopTime = currentTime; // Updates cloopTime

// Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min. (Results in +/- 3% range)

l\_hour = (flow\_frequency \* 60 / 7.5); // (Pulse frequency x 60 min) / 7.5Q = flow rate in L/hour

flow\_frequency = 0; // Reset Counter

Serial.print(l\_hour, DEC); // Print litres/hour

Serial.println(" L/hour");

}

if(myarduino.available())

{

String data = "#" + String(distance) + "#" + String(l\_hour)+ "~" ;

myarduino.println(data);

}

}

Code for nodemcu to get interfaced with Arduino and get data:

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <SoftwareSerial.h>

SoftwareSerial mynodemcu(D5,D6); // rx-5, tx-6

String data;

int waterdepth;

int flowrate;

String getValue(String data,char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++)

{

if (data.charAt(i) == separator || i == maxIndex){

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

//-------- Customise these values -----------

const char\* ssid = "manu.akhi";

const char\* password = "manasa123";

#define ORG "u98tr2"

#define DEVICE\_TYPE "NODEMCU"

#define DEVICE\_ID "7410"

#define TOKEN "9638527410"

//-------- Customise the above values --------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char topic[] = "iot-2/evt/sensordata/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883,wifiClient);

void setup() {

Serial.begin(9600);

Serial.println();

Serial.print("Connecting to ");

Serial.print(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.print("WiFi connected, IP address: ");

Serial.println(WiFi.localIP());

// Starts the serial communication

mynodemcu.begin(9600);

}

void loop() {

if(mynodemcu.available())

{

data = mynodemcu.readStringUntil('~');

}

Serial.println(data);

String a = getValue(data,'#',0);

String b = getValue(data,'#',1);

String c = getValue(data,'#',2);

Serial.print("waterdepth: ");

Serial.println(b);

Serial.print("flowrate: ");

Serial.println(c);

waterdepth = b.toInt();

flowrate = c.toInt();

PublishData(waterdepth,flowrate);

delay(1000);

}

void PublishData(int waterdepth, int flowrate){

if (!!!client.connected()) {

Serial.print("Reconnecting client to ");

Serial.println(server);

while (!!!client.connect(clientId, authMethod, token)) {

Serial.print(".");

delay(500);

}

Serial.println();

}

String payload = "{\"d\":{\"waterdepth\":";

payload += waterdepth;

payload+="," "\"flowrate\":";

payload += flowrate;

payload += "}}";

Serial.print("Sending payload: ");

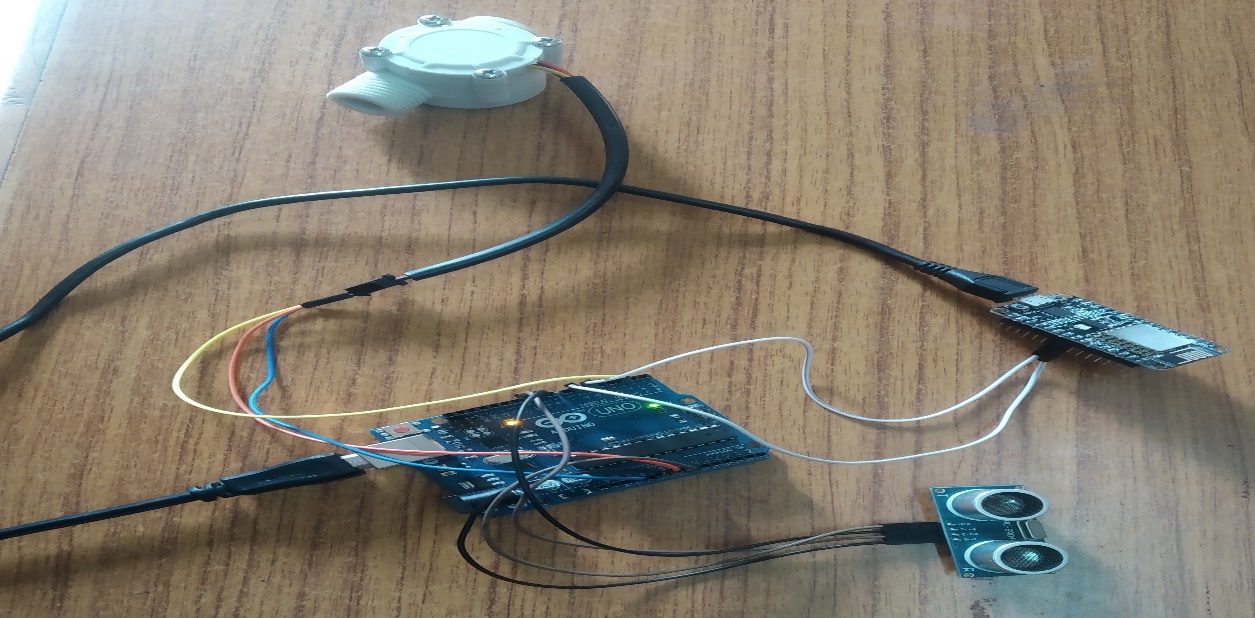
Serial.println(payload);

if (client.publish(topic, (char\*) payload.c\_str())) {

Serial.println("Publish ok");

} else {

Serial.println("Publish failed");

}

}