# AUTOMATIC WATER TANK FILLER

Overview:

The automated multiple water tanks control system is designed and implemented for the usefulness of household, industries and manufacturing processes. Also, for the control and monitoring of water overflow or chemicals in the overhead tanks, as result of common wastage of water runoff from the overhead tanks. The system failure rate, and some relevant constraints are the challenges. It includes memory capacity, power consumption, wireless facilities and many others.

Abstract:

This project is to provide customers who are not aware of weather their water motor is turned off or not in procedural amount of time. Most of the time users switch on their water motor after mean while they forget to turn it off ,which will ultimately result in power and water wastage. So to control this we introduce this model which will tell users the water level and when its full will automatically turn the motor to low.

Phase 1:

In this phase, the primary goal is to build and develop the working prototype that is to be implemented on the tank which will sense and produce the required control signal. After producing the control signal this signal is sent to android application which will indicate the user about the water level that is present in the water tank.

Phase 2:

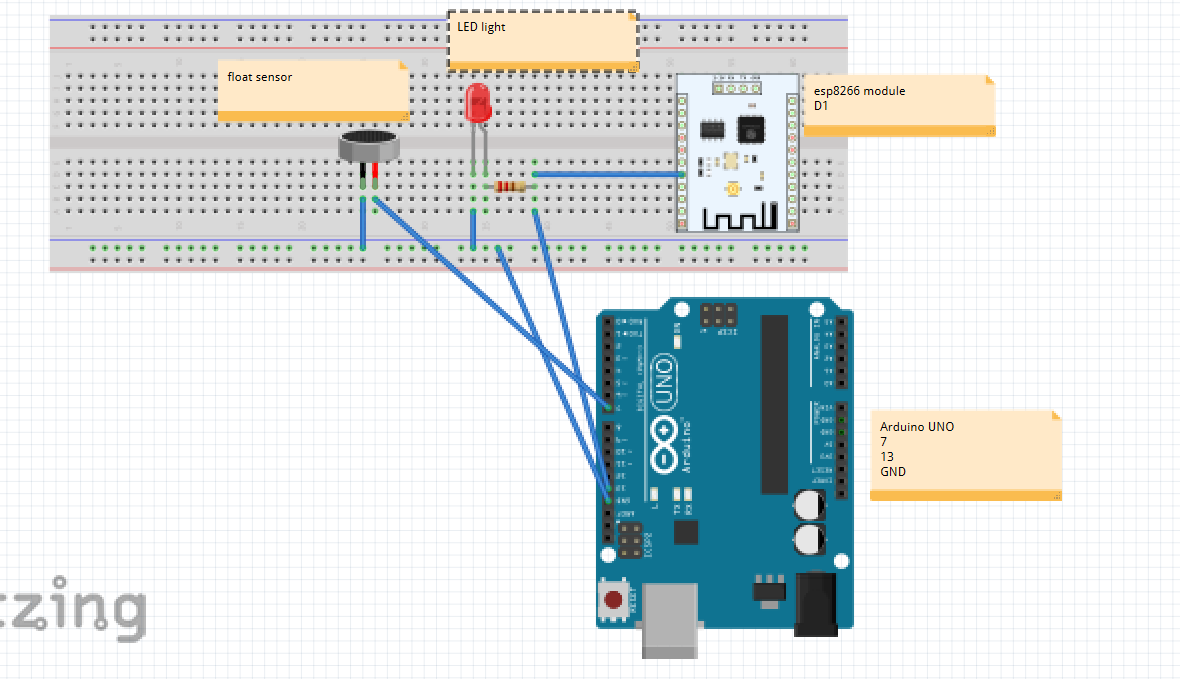
In this phase, the goal is to build, integrate and develop the modules that were made in the previous phase with the control action that is to be take on the motor side. To make it clear the control signal that is produced by the phase 1 will be sent to another module which will help the motor module to decide weather to turn on or turn off the working motor.

Phase 3:

In this final phase, we will be combining all the modules that were developed in the previous phases and see weather there are any drawbacks that were found in the final real time working model.

## Phase 1:

Schematic Diagram:



Code:

ARDUINO UNO:

int floatswitch = 7;

int button = 8;

void setup() {

// put your setup code here, to run once:

pinMode(floatswitch,INPUT\_PULLUP);

pinMode(LED\_BUILTIN, OUTPUT);

pinMode(button, INPUT);

Serial.begin(115200);

}

void loop() {

// put your main code here, to run repeatedly:

int x = digitalRead(floatswitch);

Serial.println(x);

if(x == 0)

digitalWrite(LED\_BUILTIN, LOW);

else

digitalWrite(LED\_BUILTIN, HIGH);

delay(100);

}

ESP8266 Code:

#include <ESP8266WiFi.h>

#include <Adafruit\_MQTT\_Client.h>

#define heartbeat A0

#define wifi "chandan"

#define password "9666459542"

#define server "io.adafruit.com"

#define port 1883

#define username "rohinivsenthil"

#define key "386d4c1e3b9645c985c680355889c4a4"

WiFiClient esp;

Adafruit\_MQTT\_Client mqtt(&esp,server,port,username,key);

Adafruit\_MQTT\_Publish feedHBR = Adafruit\_MQTT\_Publish(&mqtt,username"/feeds/hello");

void setup(void)

{

Serial.begin(115200);

pinMode(5, INPUT);

Serial.println("Adafruit MQTT demo");

Serial.print("Connecting to ");

Serial.println(wifi);

WiFi.begin(wifi,password);

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

{

delay(500);

Serial.print(".");

}

Serial.println("WiFi connected");

Serial.println("IP Address: ");

Serial.println(WiFi.localIP());

Serial.println("Connecting to MQTT...");

while(mqtt.connect())

{

Serial.print(".");

}

}

void loop() {

int bpm = analogRead(5);

Serial.print("Heartbeat found. bpm = ");

Serial.println(bpm);

if(mqtt.connected())

{

Serial.print("Sending heart rate data. ");

if(feedHBR.publish(bpm))

{

Serial.println("Success");

}

else

{

Serial.println("Fail upload!");

}

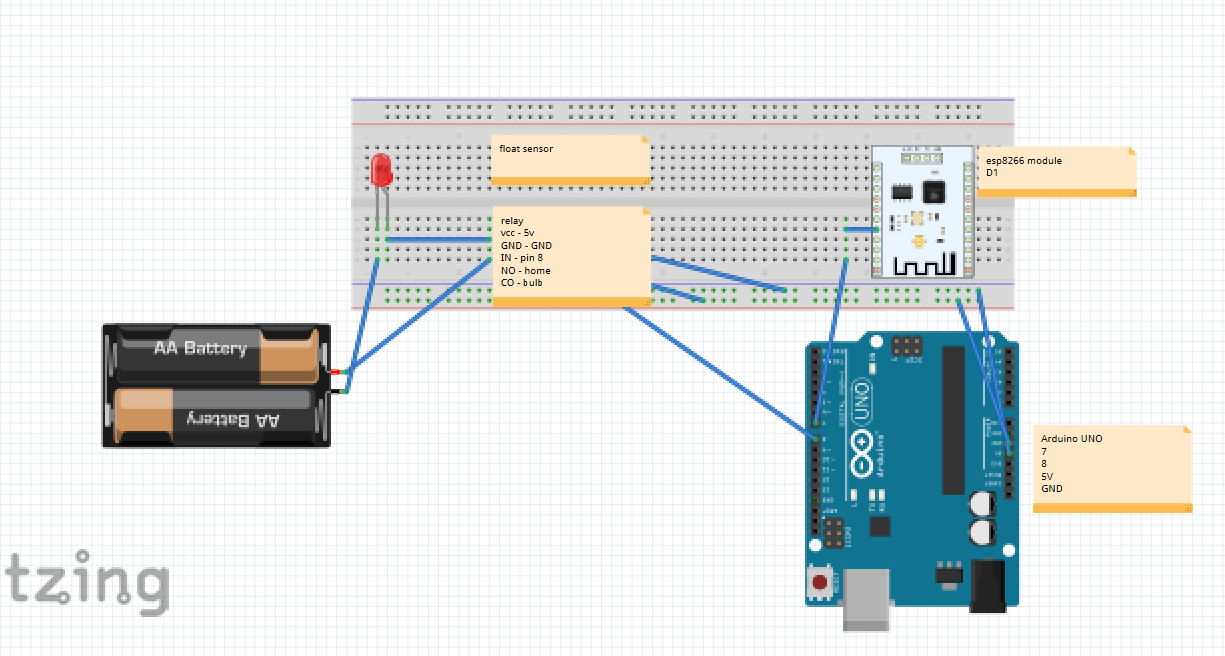
delay(2000);

}

}

## Phase 2:

Schematic Diagram:



Code:

ARDUINO UN0:

int relayPin = 8;// define output pin for relay

#define inputpin 7

int x = 0;

int t = 0;

void setup() {

Serial.begin(115200);

pinMode(relayPin, OUTPUT);// define pin 8 as output

pinMode(inputpin, INPUT);

}

void loop() {

// digitalWrite(relayPin, LOW); // turn the relay ON (low is ON if relay is LOW trigger. change it to HIGH if you have got HIGH trigger relay)

//

// delay(1000); // wait for 500 millisecond

// digitalWrite(relayPin, HIGH);// // turn the relay OFF (HIGH is OFF if relay is LOW trigger. change it to LOW if you have got HIGH trigger relay)

// delay(1000);// wait for 500 millisecond

int x = digitalRead(inputpin);

Serial.println(x);

if(x == LOW){

t = 0;

}

else{

t = 1;

}

if(t == 0){

digitalWrite(relayPin,LOW);

delay(1000);

}

else{

digitalWrite(relayPin,HIGH);

delay(1000);

}

}

ESP8266 MODULE:

#include <ESP8266WiFi.h>

#include <Adafruit\_MQTT.h>

#include <Adafruit\_MQTT\_Client.h>

// Change YOUR Wi-Fi SSID and Password below

const char\* ssid = "chandan";

const char\* password = "9666459542";

// Set MQTT Server, User name and password details below

#define MQTT\_SERVER "io.adafruit.com"

#define MQTT\_USERNAME "rohinivsenthil"

#define MQTT\_PASSWORD "386d4c1e3b9645c985c680355889c4a4"

#define MQTT\_SERVERPORT 1883

#define SUBSCRIBE\_TOPIC "rohinivsenthil/feeds/hello"

#define RELAY\_PIN 5

// Create an ESP8266 WiFiClient class to connect to the MQTT server.

WiFiClient client;

// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.

Adafruit\_MQTT\_Client mqtt(&client, MQTT\_SERVER, MQTT\_SERVERPORT, MQTT\_USERNAME, MQTT\_PASSWORD);

// Subscribe

Adafruit\_MQTT\_Subscribe onoffbutton = Adafruit\_MQTT\_Subscribe(&mqtt, SUBSCRIBE\_TOPIC);

void MQTT\_connect();

/\*

\* Set Up

\*/

void setup() {

// Connect to WiFi Network

connect2WiFi();

mqtt.subscribe(&onoffbutton);

}

/\*

\* Connects to WiFi Network

\*/

void connect2WiFi()

{

// set the baud rate

Serial.begin(115200);

// Set the Pin mode to "Output" - This is required!

pinMode(RELAY\_PIN, OUTPUT);

// wait for 10 milli seconds for it to initialize

delay(10);

Serial.println();

Serial.print("Connecting to WiFi Network - ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi Network is connected with the following.");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

}

void loop() {

// Connect to MQTT

MQTT\_connect();

Adafruit\_MQTT\_Subscribe \*subscription;

while ((subscription = mqtt.readSubscription(5000)))

{

// Check if its the onoff button feed

if (subscription == &onoffbutton)

{

// Serial.print(F("Relay Status: ")); Serial.println((char \*)onoffbutton.lastread);

if ( strcmp((char \*)onoffbutton.lastread, "1023") == 0)

{

digitalWrite(RELAY\_PIN,HIGH);

Serial.println("HIGH");

}

if ( strcmp((char \*)onoffbutton.lastread, "0") == 0)

{

digitalWrite(RELAY\_PIN,LOW);

Serial.println("LOW");

}

}

}

// ping the server to keep the mqtt connection alive

if(! mqtt.ping()) {

mqtt.disconnect();

}

}

void MQTT\_connect()

{

int8\_t ret;

if (mqtt.connected())

{

return;

}

Serial.print("Connecting to MQTT... ");

uint8\_t retries = 3;

while ((ret = mqtt.connect()) != 0)

{

// connect will return 0 for connected

Serial.println(mqtt.connectErrorString(ret));

Serial.println("Retrying MQTT connection in 5 seconds...");

mqtt.disconnect();

delay(5000); // wait 5 seconds

retries--;

if (retries == 0)

{

while (1);

}

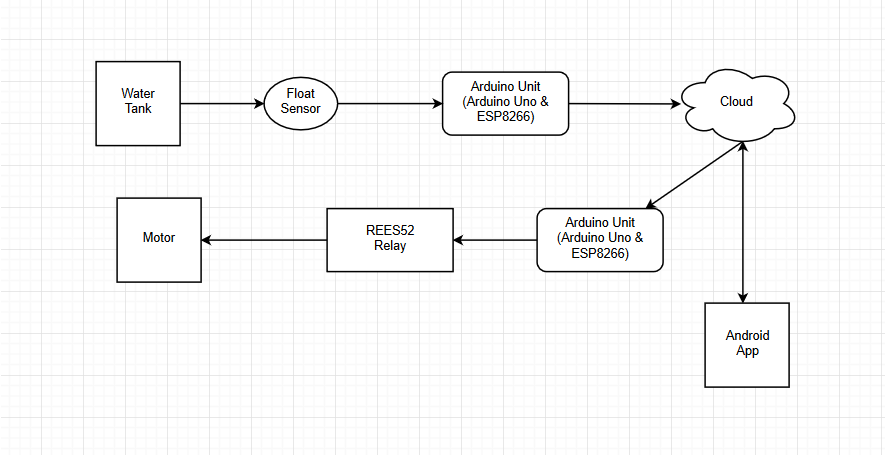
}

Serial.println("MQTT Connected!");

}

## Phase 3:

Flow Diagram:



Explanation: