INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 4)
Available online at: www.ijariit.com

Capacitive sensor level measurement and Bluetooth control with Arduino

Daniel Asemota
danielasemota904@gmail.com
Vellore Institute of Technology, Vellore, Tamil Nadu

Harsha Jayana
harshass.jayana@gmail.com
Vellore Institute of Technology, Vellore, Tamil Nadu

ABSTRACT

This project deals with the measurement of water level in a tank using capacitance. The level of water in the tank will be displayed on our cell phone by using a Bluetooth module in an Arduino circuit. When the water level exceeds the highest level, a buzzer starts ringing to alert us of rising water levels. This prevents overflowing in tanks in industries. Accurate control of the level of material in a tank, reactor, or other vessel is important in many process applications especially, in the bulk material handling industry. One of the main purposes of the level measurement system is to measure inventory. In order to achieve good control, accurate measurement is a must.

Keywords— *Capacitor, Sensor, Arduino*

1. OBJECTIVE

Our major aim is to measure the level of a liquid in the container by the means of capacitive level sensor. Capacitive liquid level sensors work by measuring capacitance of the probe which is immersed into the liquid or even placed around the tank (which in that case must be non-metallic). Depending on amount of liquid between probe electrodes, the resulting capacitance will be different which can be detected by electronics and used to calculate level or volume of liquid inside a tank. Measurement of liquid level inside a container with various methods has been developed occasionally. Liquid level measurement can be utilized from the characteristic of the liquid itself; such as permittivity, permeability, conductivity, et cetera. One type of sensor which developed for liquid level measurement is capacitive sensor. Capacitive sensor can be categorized as reactive sensor. Hence, it is influenced by its input frequencies. Generally, capacitive sensor has non- contact characteristic.

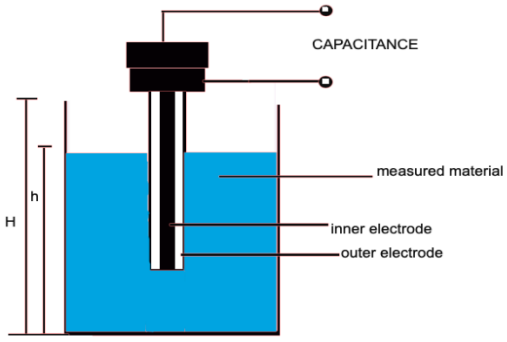

Fig. 1: Experimental setup

2. COMPONENTS

- 2 parallel plate capacitors
- A plastic container
- Wooden base
- An Arduino board
- Bluetooth Module
- Buzzer
- LED Indicator

- Android cell phone
- Breadboard circuit
- Electric wires
- Laptop with Arduino Software
- Electric drill
- Tape
- Wire cutters
- Glue gun

3. METHODOLOGY

The two plates are connected to an Arduino which is then connected to a breadboard circuit. This circuit has an LED indicator and a buzzer connected to it. It starts buzzing when the liquid level reaches the highest level and the indicator starts glowing. The signal conditioning circuit is encrypted on the software using the Arduino application. The software takes the capacitive input, converts the capacitance into voltage, and then gives the required level. Usually, the capacitance is in the range of picoFarads.

A capacitor has 2 conducting plates (Electrodes). When a charge is applied tothese plates, the space in between the plates will also get a charge. The charge that can be between these plates depends on the material in between the plates (the Dielectric). The ability of a material to be charged is called relative permittivity.

The capacitive level sensor has the 2 conducting plates in the form of 2 electrically isolated aluminum, placed in parallel. The space between the tubes is the dielectric. When the tube is empty, the space is occupied by air. When the tube starts to fill, more and more of the space will be occupied by water. Water holds more charge than air and thus the capacitance will rise (mostly) linearly with the water level.

By using parallel plated probes, distance'd' is fixed. Also, gain of the instrument is both linearized and increased, where gain is the capacitance change per inch change in level. We then connected one of the electrodes to system ground. The Arduino's way of measuring capacitance involves charging it and measuring the time it takes for the voltage to rise on the other electrode – at a certain threshold it compares the current time to when it began charging. Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth. It has 6 pins,

3.1 Key/EN

It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode. HC-05 module has two modes:

- **Data mode:** Exchange of data between devices.
- **Command mode:** It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.

3.2 VCC

Connect 5 V or 3.3 V to this Pin.

3.3 GND

Ground Pin of module.

3.4 TXD

Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)

3.5 RXD

Receive data serially (received data will be transmitted wirelessly by Bluetooth module).

3.6 State

It tells whether module is connected or not.

Every message the Arduino wants to send, is first given to the Bluetooth-Module, which sends the message wirelessly. To avoid problems with the UART, Arduino and Bluetooth-Module have to use the same baud-rate (in default 9600). Before using the app the Bluetooth-Module (HC-05/HC-06) has to be coupled to the Android in the system-preferences. In the special case of the HC-05 the default PinCode for initiating the Coupling-Process is "1234" or "0000".

4. CODE

```
#include <SoftwareSerial.h> #include <LiquidCrystal.h>
//LiquidCrystal lcd(12, 11, 5, 4, 3, 2); int led = 13;
int led2 = 12;
int contrast=120;
const int OUT_PIN = A2; const int IN_PIN = A0;
const float IN_STRAY_CAP_TO_GND = 24.48;
const float IN_CAP_TO_GND = IN_STRAY_CAP_TO_GND; const float R_PULLUP = 34.8;
```



```
const int MAX_ADC_VALUE = 1023; const int buzzer = 11;

void setup()
{
pinMode(OUT_PIN, OUTPUT);
analogWrite(6,contrast);
//lcd.begin(16, 2); pinMode(IN_PIN, OUTPUT); pinMode(buzzer, OUTPUT); pinMode(led,OUTPUT); Serial.begin(9600);
}

void loop()
{
pinMode(IN_PIN, INPUT); digitalWrite(OUT_PIN, HIGH); int val = analogRead(IN_PIN); digitalWrite(OUT_PIN, LOW);

if (val < 1000)
{
pinMode(IN_PIN, OUTPUT);

float capacitance = (float)val * IN_CAP_TO_GND / (float)(MAX_ADC_VALUE - val);

Serial.print(F("Capacitance Value = ")); Serial.print(capacitance, 3);

Serial.print(F(" pF (")); Serial.print(val); Serial.println(F(")"));

delay(100);
if (capacitance > 58)
{
Serial.print("LEVEL = 0 CM"); Serial.print("\n"); Serial.print("TANK EMPTY"); Serial.print("\n"); digitalWrite(led2,HIGH); de-
lay(700);
//digitalWrite(led2,LOW);
}
}
else
{
pinMode(IN_PIN, OUTPUT);//discharge the capacitor (from low capacitance test) delay(1);
pinMode(OUT_PIN, INPUT_PULLUP);//Start charging the capacitor with the internal pullup unsigned long u1 = micros();
unsigned long t; int digVal;
do
{
digVal = digitalRead(OUT_PIN); unsigned long u2 = micros();
t = u2 > u1 ? u2 - u1 : u1 - u2;
} while ((digVal < 1) && (t < 400000L));

pinMode(OUT_PIN, INPUT); val = analogRead(OUT_PIN); digitalWrite(IN_PIN, HIGH);
int dischargeTime = (int)(t / 1000L) * 5; delay(dischargeTime); pinMode(OUT_PIN, OUTPUT);
digitalWrite(OUT_PIN, LOW); digitalWrite(IN_PIN, LOW);
//has reached ss for apprx 1000
float capacitance = -(float)t / R_PULLUP
/ log(1.0 - (float)val / (float)MAX_ADC_VALUE);

Serial.print(F("Capacitance Value = ")); if (capacitance > 1000.0)
{
Serial.print(capacitance / 1000.0, 2); Serial.print(F(" uF"));
delay(100);
}
else
{
Serial.print(capacitance, 2); Serial.print(F(" nF"));

delay(100);
}
//if (capacitance = 2322 / 1000, 2) if (capacitance > 51000)
{
Serial.print("TANK FULL"); Serial.print("\n"); Serial.print("LEVEL = 15CM"); digitalWrite(buzzer, LOW); digital-
Write(led,HIGH); delay(1000);
// detachInterrupt(digitalPinToInterrupt(11));
//digitalWrite(led,LOW); digitalWrite(led2,LOW);
```



```
if (capacitance >46000)
{
Serial.print("LEVEL = 12CM"); Serial.print("\n");
}

if (capacitance > 38000)
{
Serial.print("LEVEL = 9CM"); Serial.print("\n");
}

if (capacitance >32000)
{

Serial.print("LEVEL = 6CM"); Serial.print("\n");

}
if (capacitance > 27000)
{

Serial.print("LEVEL = 3CM"); Serial.print("\n");

}
if (capacitance /;> 21000)
{

Serial.print("LEVEL = 2CM"); Serial.print("\n");
}

}
else if (capacitance <53000)
{
digitalWrite(buzzer,HIGH);
}
if(Serial.available())
{
String value = Serial.readStringUntil('\n'); Serial.println(value);
if(value == "0")
{
digitalWrite(buzzer, LOW); digitalWrite(led,LOW); delay(1000000000);
// detachInterrupt(digitalPinToInterrupt(11)); Serial.print("500 ml/s");
Serial.print(",");
}
else if(value != "0")
{
digitalWrite(led,LOW); digitalWrite(buzzer, LOW);
}
}

Serial.print(F(" ("));
Serial.print(digVal == 1 ? F("Normal") : F("HighVal")); Serial.print(F(", t= "));
Serial.print(t); Serial.print(F(" us, ADC= ")); Serial.print(val); Serial.println(F(")"));
}while (millis() % 1000 != 0);}
```

5. RESULT

The capacitance was successfully measured and converted into level. The Bluetooth module enabled us to measure the level of the tank and warned us when the tank was empty or full. Hence, the experiment was successfully implemented.

6. REFERENCES

[1] http://iopscience.iop.org/article/10.1088/1742-6596/776/1/012118/pdf
[2] https://www.instructables.com/id/Capacitive-Fluid-Level-Sensor/
[3] https://www.best-microcontroller-projects.com/capacitive-liquid-level- sensor.html
[4] http://www.circuitbasics.com/how-to-make-an-arduino-capacitance- meter/

