Development of Internet of Things (IoT) based Data Acquisition system using Arduino Uno

Shrushti B Walvekar

Department of Electronics and Communication, KLS Gogte Institute of Technology, Belgaum-590008

---***---

ABSTRACT :

The main objective of our project is focused around the design and development of cost effective IoT based data acquisition system (DAS), both hardwares and softwares. which are used for measuring slowly varying physical variables. Physical parameters such as temperature, humidity, flow rate etc., which are generally considered as slowly varying signals, are measured by the designed DAS. Design consideration of the PC based data acquisition system has been made, based on commonly available ports of the personal computers (PCs) viz. serial port. The Internet of Things based Data Acquisition System will be equipped with different possible ways of data transmission. Implementation of this automated data acquisition will help in continuous monitoring of data from any part of the world. It can also be extended to acquire different types of sensors to monitor different parameters. The implementation is economical and the system will be a generalized system to be used in various fields such as for medical, industrial, robotics etc. with all connectivity options and with different range options.

Keywords: DAS, IoT, PCs, Arduino Uno

1. INTRODUCTION

The Data acquisition system is simply defined as the system that either monitors or controls the parameters in the outside world. In today's time each and every field cannot be thought of without considering these data acquisition systems. For medical instruments, industrial equipment, appliances for home etc. has a necessary need for these systems. Hence the need of such data collection and processing the collected data for further use led to the development of Internet of Things (IoT) based Data Acquisition system using Arduino Uno. The system will be designed to collect (acquire) data from the float, flow and temperature of water. The basic need was to develop our system was portability of the system, no range limitation and also more than one connectivity options for transfer of data. The system will also be capable of displaying the acquired data over 20*4 alphanumeric display. Further, it is supposed to be designed in such a way that it transfers these data to various ranges. Thus it will be useful for data acquisition and transfer from short distances to longer.

It could be regarded as a general purpose system with multiple transfer options. The system will be designed to be equipped with all possible features which enhance the working capability of the system. The system will be consisted of Arduino Uno which is one of the advanced processor and has advantages like low price and compact in size. Our system will consist of an inbuilt Analog to Digital conversion module which will be used to convert the analog output from the sensors to digital values for further processing. The system will be using a number of protocols for achieving the concept of Internet of Things for our application. The system will be able to communicate with all possible wired or wireless ways thus have multiple way of communication. Wi-Fi will be used for medium range while GSM will be used long range transmission. Thus, making is an automated data acquisition system.

The motive of the current project is the physical parameters such as temperature, humidity, float and flow rate are sensed using appropriate sensors. The sensed data is then collected, analysed and acted accordingly to control critical parameters in an industry. These sensed parameters are then sent over the cloud (IoT) through ESP8266 WIFI module interfaced via Arduino. Posting data onto to the IoT one can access from any part of the world easily.

2. METHODOLGOY

Firstly, the instantaneous data like temperature, humidity, flow and float values are sensed by the appropriate sensors. The code written checks the condition for tank-2 indicating if it is empty (i.e., the float switch is at low level) and provided that temperature is below set threshold. Once the condition is satisfied the relay turns on and in turn causes the submersible pump to get turned on which results in the flow of water from tank-1 to tank-2. Elsewhere if the condition fails the relay does not get turned on and the entire process stops. However, a question may arise that what if the relay goes off damaged? For this very reason it is made sure that the data is sent to the IoT intimating a message to prescribed person only when it is working else data is not sent to IoT. Secondly, these acquired parameters are posted on to the IoT channels (Thingspeak) where the code is programmed to receive these for every 14 secs. If the relay is off no data is received only a pulse of level 10 (indicating pump status as off) is sent to respective field.

2.1 Software requirement

2.1.1 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) -contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. We employed the same data types, operators, and statements as we use in C programming language. The open-source Arduino Software (IDE-Integrated development Environment) makes it easy to write code and upload it to the board. • IDE runs on Windows, Mac OS X, and Linux. The environment is written in Java. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and is also capable of compiling and uploading programs to the board with a single click. The code for the Arduino is written in C/C++. The Arduino IDE comes with a C/C++ library, which makes many common input/output operations much easier.

2.2 Hardware requirement

2.2.1 Power supply

The circuit needs two different voltages, +5V & +12V, to work. These dual voltages are supplied by this specially designed power supply. The main object of this ‘power supply’ is, as the name itself implies, to deliver the required amount of stabilized and pure power to the circuit.

Every typical power supply contains the following sections:

1. Step-down transformer: The conventional supply, which is generally available to the user, is 230V AC. It is necessary to step down the mains supply to the desired level. This is achieved by using suitably rated step-down transformer. The reason for this is, for proper working of the regulator IC (say KIA 7805) it needs at least 2.5V more than the expected output voltage.

2. Rectifier stage: Then the step-downed Alternating Current is converted into Direct Current. This rectification is achieved by using passive components such as diodes. If the power supply is designed for low voltage/current drawing loads/circuits (say +5V), it is sufficient to employ full- wave rectifier with centre-tap transformer as a power source. While choosing the diodes the PIV rating is taken into consideration.

3. Filter stage: But this rectified output contains some percentage of superimposed a.c. ripples. So to filter these a.c components filter stage is built around the rectifier stage. The cheap, reliable, simple and effective filtering for low current drawing loads (say upto 50 mA) is done by using shunt capacitors. This electrolytic capacitor has polarities, take care while connecting the circuit.

4. Voltage regulation: The filtered d.c. output is not stable. It varies in accordance with the fluctuations in mains supply or varying load current. This variation of load current is observed due to voltage drop in transformer windings, rectifier and filter circuit. These variations in d.c. output voltage may cause inaccurate or erratic operation or even malfunctioning of many electronic circuits. For example, the circuit boards which are implanted by CMOS or TTL ICs.

5. Circuit description: A d.c. power supply which maintains the output voltage constant irrespective of a.c. mains fluctuations or load variations is known as **regulated d.c. power supply.** It is also referred as full-wave regulated power supply as it uses four diodes in bridge fashion with the transformer. This laboratory power supply offers excellent line and load regulation and output voltages of +5V & +12 V at output currents up to 1 Amp.

2.2.2 Arduino Uno

The Arduino UNO is a widely used open-source microcontroller board based on the ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board features 14 Digital pins and 6 Analog pins. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. The Arduino UNO is generally considered the most user-friendly and popular board, with boards being sold worldwide for less than 5$.

Technical specifications

- Microcontroller: ATmega328P
- Operating Voltage: 5v
- Input Voltage: 7-20v
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm, Width: 53.4 mm, Weight: 25 g

2.2.2 Arduino Technology

A typical example of the Arduino board is Arduino Uno. It includes an ATmega328 microcontroller and it has 28-pins.

Arduino Pin-out specification

The pin configuration of the Arduino Uno board is consists of 14-digital i/o pins. Wherein 6 pins are used as pulse width modulation outputs and 6 analog inputs, a USB connection, a power jack, a 16MHz crystal oscillator, a reset button, and an ICSP header. Arduino board can be powered either from the personal computer through a USB or external source like a battery or an adaptor. This board can operate with an external supply of 7-12V by giving voltage reference through the I/O Ref pin or through the pin Vin.

Digital Inputs: It comprises of 14-digital I/O pins, each pin take up and provides 40mA current. Some of the pins have special functions like pins 0 & 1, which acts as a transmitter and receiver respectively. For serial communication, pins-2 & 3 are external interrupts, 3,5,6,9,11 pins delivers PWM o/p and pin-13 is used to connect LED.

Analog inputs: It has 6-analog I/O pins, each pin provide a 10 bits resolution.

Aref: This pin gives a reference to the analog inputs.

Reset: When the pin is low, then it resets the microcontroller.

2.2.3 Temperature and humidity sensor

In this Arduino Tutorial we will learn how to use the DHT11 sensor for measuring temperature and humidity with the Arduino board. DHT11 temperature range is from 0 to 50 degrees Celsius with +-2 degrees accuracy.DHT11 humidity range is from 20 to 80% with 5% accuracy.

DHT11	SPECIFICATIONS
0 - 50˚C /±2˚C	Temperature Range
20 - 80% / ±5%	Humidity Range
1Hz (one reading every second)	Sampling Rate
15.5mm x 12mm x 5.5mm	Body Size
3 - 5V	Operating Voltage
2.5mA	Max Current During Measuring

Table 1 DHT11 Specification

Working

They consist of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC on the back side of the sensor. For measuring humidity they use the humidity sensing component which has two electrodes with moisture holding substrate between them. So as the humidity changes, the conductivity of the substrate will change or the resistance between these electrodes will change. This change in resistance is measured and processed by the IC which makes it ready to be read by a microcontroller. On the other hand, for measuring temperature these sensors use a NTC temperature sensor or a thermistor. A thermistor is actually a variable resistor that changes its resistance with change of the temperature. These sensors are made by sintering of semi conductive materials such as ceramics or polymers in order to provide larger changes in the resistance with just small changes in temperature. The term "NTC" means "Negative Temperature Coefficient", which means that the resistance decreases with increase of the temperature.

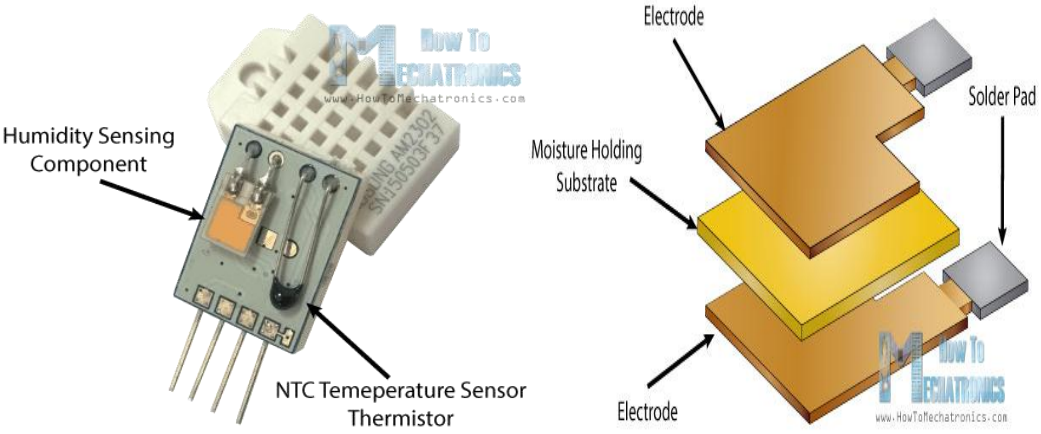

Humidity Sensing Component
NTC Temeperature Sensor Thermistor
How To Mechatronics
www.HowToMechatronics.com
Electrode
Solder Pad
Moisture Holding Substrate
Electrode
How To Mechatronics
www.HowToMechatronics.com

Fig.1 DHT11 Working

2.2.4 Arduino Interfacing to DHT11 Sensor

The DHT11 sensors have four or three pins, VCC, GND, data pin and a not connected pin which has no usage. A pull-up resistor from 5K to 10K Ohms is required to keep the data line high and in order to enable the communication between the sensor and the Arduino Board. There are some versions of these sensors that come with a breakout boards with built-in pull-up resistor and they have just 3 pins. The DHT11 sensors have their own single wire protocol used for transferring the data. This protocol requires precise timing and the timing diagrams for getting the data from the sensors can be found from the datasheets of the sensors.

Float sensor

Float level switches are general liquid level switches, and are used to detect liquids. These level switches have been used in various locations where liquid levels must be detected. We recommend our float type level switch for low cost liquid measurement. The principle of float type level meter is that a float moves up and down due to buoyancy. A reed switch in a stem is actuated by a magnet in the float, and outputs detection signal (see the figure below).However, the level meter may not measure liquids with some viscosities and gravities. The purpose of a float switch is to open or close a circuit as the level of a liquid rises or falls. Most float switches are "normally closed," meaning the two wires coming from the top of the switch complete a circuit when the float is at its low point, resting on its bottom clip or stop (for example, when a tank is dry).Connect the Ground pin to Gnd then connect Signal pin to Digital pin D9, this sensor has control circuit hence there is no need for pull up resistor.

Flow sensor

A flow sensor is a device used to measure the instant flow rate or quantity of a gas or liquid passing through a pipeline. Flow meters are also known to by other names, such as flow gauge, flow indicator, liquid meter, etc. Depending on the particular industry, yet the function to measure flow remains the same in all the meters. This water flow sensor has only three wires and it can be easily interfaced between any microcontroller and Arduino board. It requires only +5V Vcc and gives pulse output, the sensor needs to be tightly fitted between water pipeline. Connect the +5V wire to Arduino power pin 5V and Ground pin to Gnd then connect Signal pin to Analog pin A0, this sensor has control circuit hence there is no need for pull up resistor, some sensor requires pull up resistors.

2.2.5 SIM 800A

GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a SIM (Subscriber Identity Module) card just like mobile phones to activate communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.
2. Read, add, search phonebook entries of the SIM.
3. Make, Receive, or reject a voice call.

SIM800A Modem is built with Dual Band GSM/GPRS based SIM900A modem from SIMCOM. It works on frequencies 900/ 1800 MHz. SIM800A can search these two bands automatically. The frequency bands can also be set by AT Commands. The baud rate is configurable from 1200-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. SIM800A is an ultra-compact and reliable wireless module. This is a complete GSM/GPRS module in a SMT type and designed with a very powerful single-chip processor integrating AMR926EJ-S core, allowing you to benefit from small dimensions and cost-effective solutions.

General features

• Quad-band 850/900/1800/1900MHz
• GPRS multi-slot class 12/10
• GPRS mobile station class B
• Compliant to GSM phase 2/2+
– Class 4 (2 W @ 850/900MHz)
– Class 1 (1 W @ 1800/1900MHz)
• Bluetooth: compliant with 3.0+EDR
• Dimensions: 24*24*3mm
• Weight: 3.14g
• Control via AT commands

(3GPP TS 27.007,27.005 and SIMCOM enhanced AT Commands)
- Supply voltage range 3.4 ~ 4.4V
- Low power consumption
- Operation temperature:-40℃ ~85℃

Interface GSM Module to Arduino-Send SMS

To interface **GSM Module to Arduino**. There are different kinds of GSM modules available in market. We are using the most popular module based on **Simcom SIM800** and **Arduino Uno**. We only need to make 3 connections between the gsm module and arduino.

Booting of SIM800A

- Insert your SIM card to GSM module and lock it.
- Power up your gsm by connecting it to Arduino's 5V and GND.
- Connect the Antenna.
- Now wait for some time (say 1 minute) and see the blinking rate of 'status LED' or 'network LED' (GSM module will take some time to establish connection with mobile network)
- Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. You may try making a call to the mobile number of the sim card inside GSM module. If you hear a ring back, the gsm module has successfully established network connection

Connecting GSM Module to Arduino

There are two ways of connecting GSM module to Arduino. In any case, the communication between Arduino and GSM module is serial. So we are supposed to use serial pins of Arduino (Rx and Tx). So if you are going with this method, you may connect the Tx pin of GSM module to Rx pin of Arduino and Rx pin of GSM module to Tx pin of Arduino. **GSM Tx –> Arduino Rx** and **GSM Rx –> Arduino Tx**. Now connect the ground pin of Arduino to ground pin of gsm module. So given below is the circuit diagram to connect gsm module to Arduino – and hence use the circuit to send sms using Arduino and gsm modem.

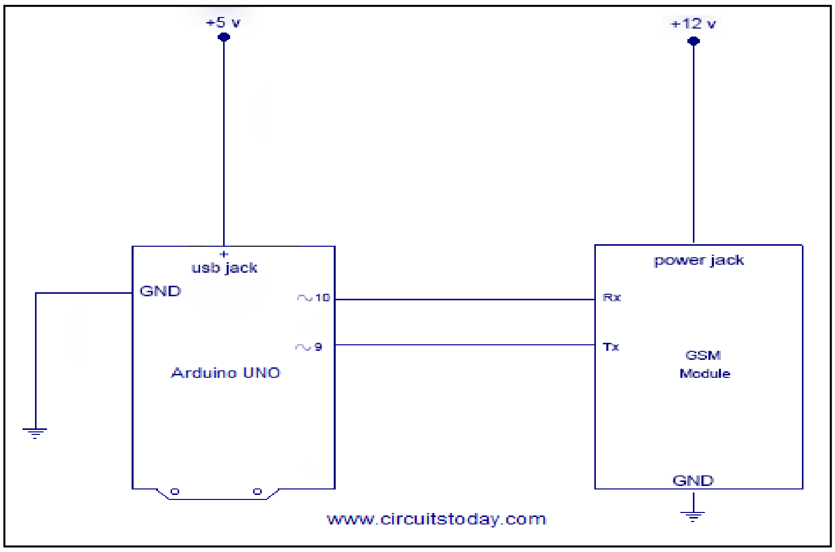


Fig 2.Interfacing of SIM800A

Make the connections as Shown. Send SMS using Arduino and GSM Module – to a specified mobile number inside the program.

Lets get the functions **Send Message ()**

SendMessage () – is the function we created in our Arduino sketch to send an SMS. To send an SMS, we should set our GSM module to Text mode first. This is achieved by sending an AT Command "AT+CMGF=1" We send this command by writing this to Software Serial port. To achieve this we use the mySerial.println() function. mySerial.println writes data to software serial port (the Tx pin of our Software Serial – that is pin 10) and this will be captured by GSM module (through its Rx pin). After setting the GSM module to Text mode, we should the the mobile number to which we shall send the SMS. This is achieved with AT command **"AT+CMGS=\"+91xxxxxxxxxx\"\r"** – where you may replace all x with the mobile number.

In next step, we should send the actual content of SMS. The end of SMS content is identified with CTRL+Z symbol. The **ASCII** value of this **CTRL+Z** is **26**. So we send a char(26) to GSM module using the line mySerial.println((char)26); Each and every AT command may be followed by 1 second delay. We must give some time for GSM module to respond properly. Once these commands are send to GSM module, you shall receive an SMS in the set mobile number.

AT Commands to Send SMS using Arduino and GSM Module

- AT+CMGF=1 /7/ Set the GSM module in text mode.
- AT+CMGS=\"+YYxxxxxxxxxx\"\r // Input the mobile number| YY is country code "the message" with stopping character (char)26 // ASCII of ctrl+z

20*4 Alphanumeric LCD

A liquid-crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly. Here, in this we are going to use a monochromatic 20x4 alphanumeric LCD. 20x4 means that 20 characters can be displayed in each of the 4 rows of the 20x4 LCD, thus a total of 80 characters can be displayed at any instance of time.

Features

- **Type:** Character
- **Display format:** 20 x 4 characters
- **Built-in controller:** ST 7066 (or equivalent)
- **Duty cycle:** 1/16 • 5 x 8 dots includes cursor
- + 5 V power supply (also available for + 3 V)
- LED can be driven by pin 1, pin 2, pin 15, pin 16 or A and K
- N.V. optional for + 3 V power supply

Interfacing LCD with Ardiuno

We are going to use a 20x4 blue character LCD. The Arduino talks to the LCDs via the four data lines. We use the digital pins on the Arduino to talk to the LCD and display what we want on it. Apart from these lines, there is also an enable pin, RS pin and an RW pin. The backlight on the LCD is activated when you give 5V from the Arduino to pin 15 on the LCD and by grounding pin 16. Apart from this, you also need a 10K potentiometer to adjust the contrast of the LCD and the connection is carried out as shown in fig 4.

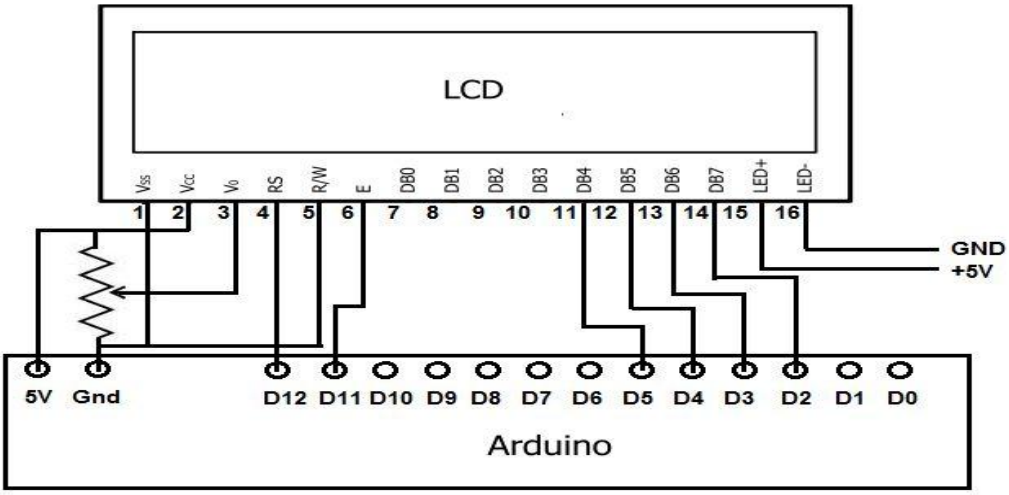


Fig 3. 20*4 LCD Display Interfacing

2.3 IMPLEMENTATION

As stated earlier, this project heavily relies on simple data acquired from the sensors. This is facilitated by a ESP8266 and several software libraries that support data acquisition.

BLOCK DIAGRAM

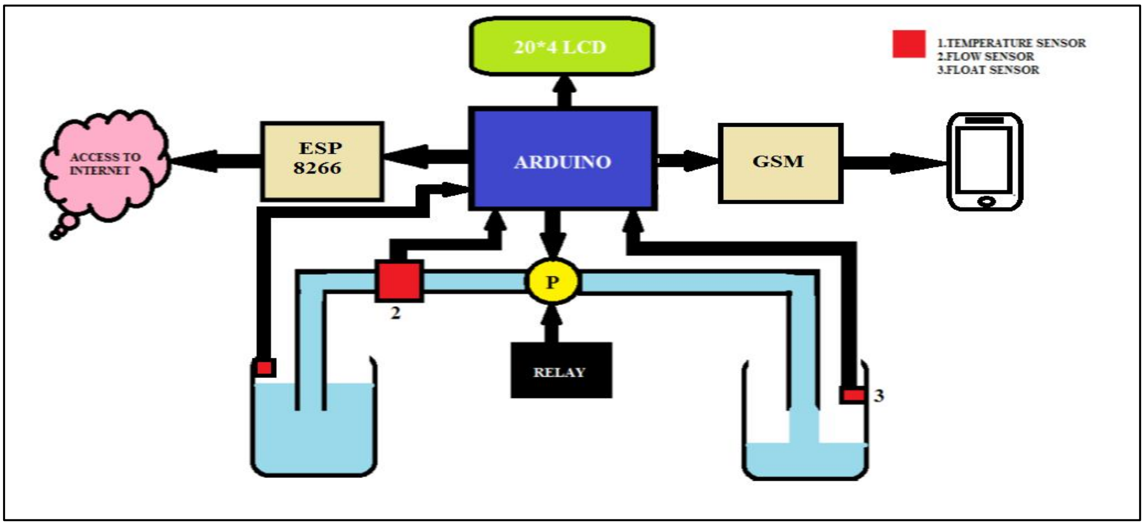


Fig 4. Block Diagram

Instantaneous data from the sensors, such as temperature, humidity, flow and float values are acquired. The logic written is to check the condition for temperature (dht11 sensor) is greater than set threshold value and the float(float sensor) is at low level. Only if this condition is satisfied the water flows from tank-1 to tank-2, determining the flow rate of the liquid that flows through it. This action is done by the flow sensor. Sensed data is sent to IoT (thingspeak channel fields) using ESP8266 (Wi-fi module).If the above condition fails then entire procedure stops and no data is sent to the cloud through Wi-fi module. A message is sent to particular person using GSM module to indicate whether the relay is ON and that the tank is empty.

The data sent to the cloud can be viewed on the thingspeak channel fields for every 14 seconds. Simple cost effective methods and techniques have been proposed for the design and development of PC based DAS for measuring some physical parameters. They are

1. Design and development of serial port (9600) DAS. It includes the hardware design i.e. PCB design and fabrication for circuits like power supply.
2. For serial port (9600) based DAS, it uses Arduino-Uno controller. It has built-in ADC. For proper functioning of the device, firmwares have been developed using 'C'.
3. Application programs for communication with the designed H/W have been developed using 'C'. Observations were taken using the temperature and humidity sensors.

3. RESULTS

The following results were obtained in the project.

- **Temperature:** Based on the conditions that temperature should be less than specified threshold (here 40˚c) we have obtained the results as 29˚c this sensed value along with the pump status determines the overall operation as depicted in figure below.

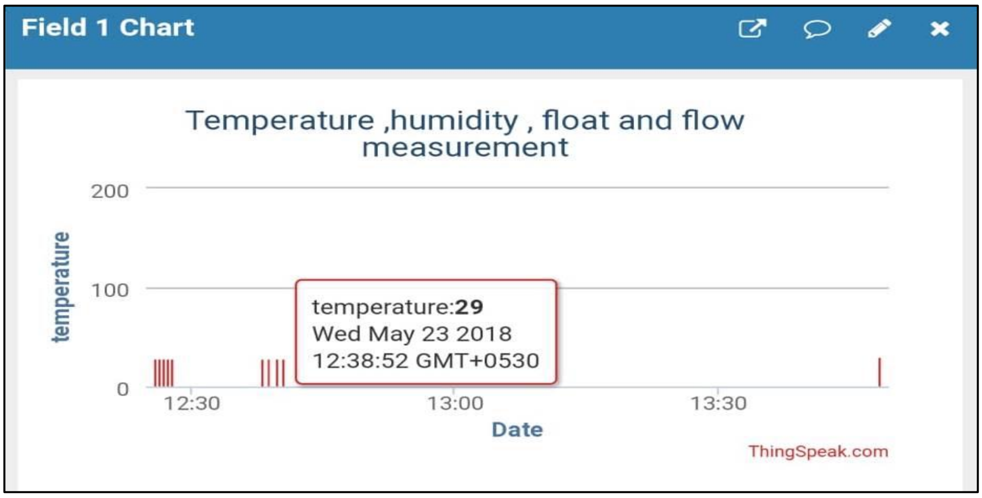


Fig 5 (a) Temperature

- **Humidity:** Humidity changes as the moisture content in the surrounding air changes the below results are obtained within a specific unit or container.

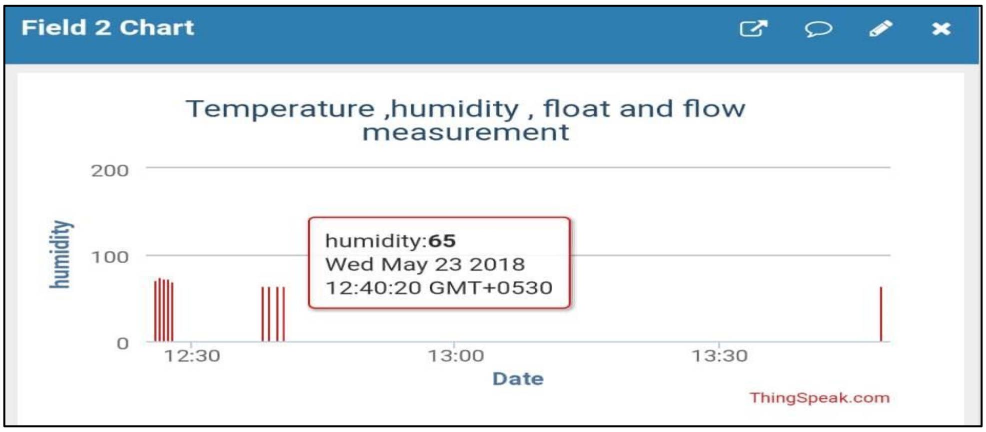
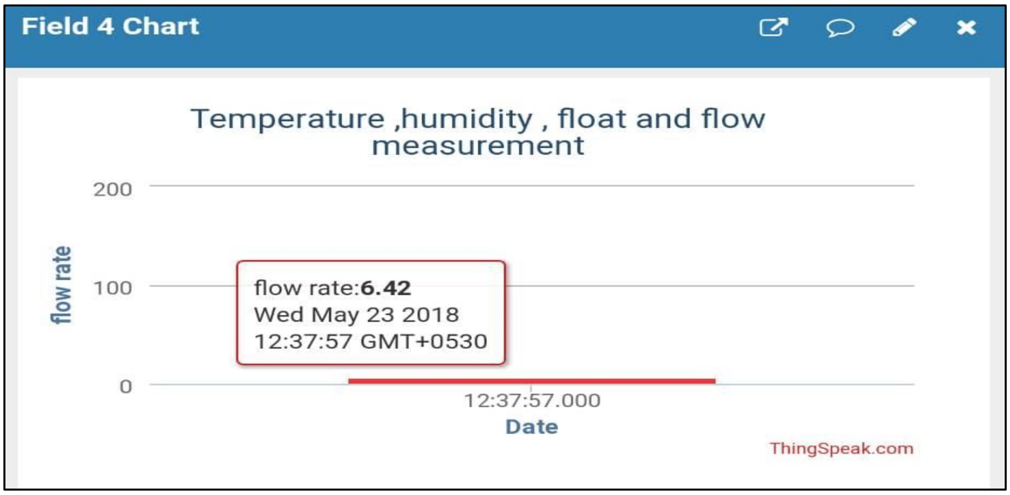


Fig 5 (b) Humidity

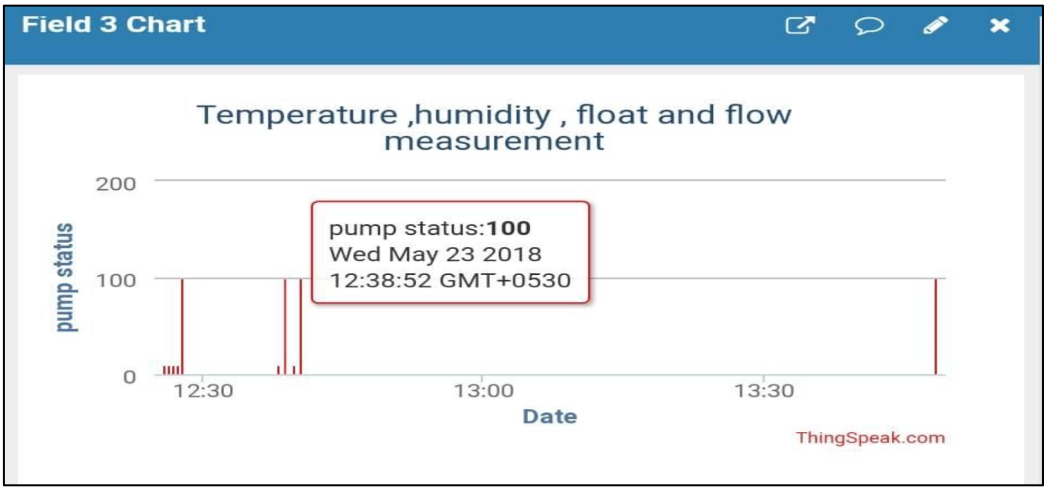
- **Flow Rate:** Every rotation means 2.25mL of fluid has passed through the sensor, since water flown is 6.42mL, the sensor produces of about 2.85 Rpm.



flow rate:**6.42**
Wed May 23 2018
12:37:57 GMT+0530

Fig 5 (c) Flow Rate

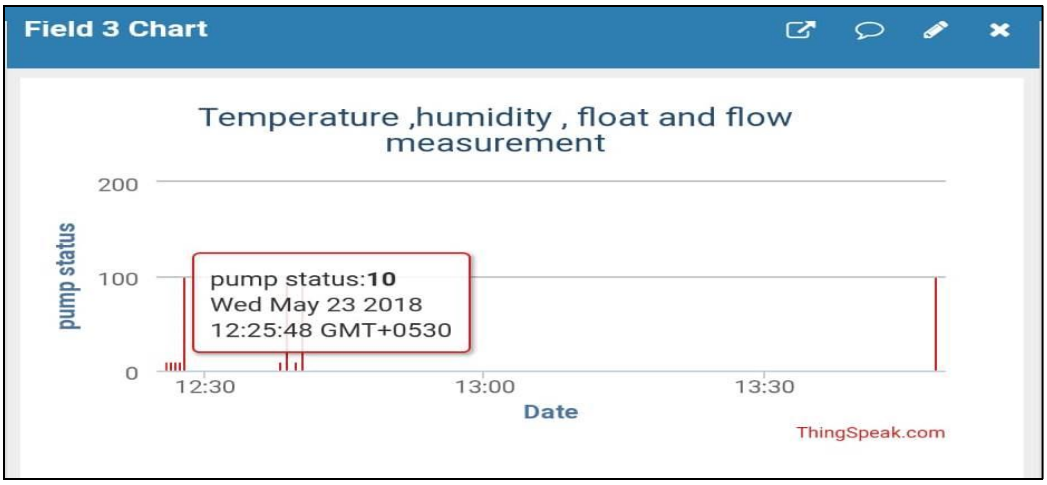
- **Pump Status (Float level):** The pump status 100 indicates that the tank is full and that it will no more allowed the water to flow through flow sensor or turns off the motor.



pump status:**100**
Wed May 23 2018
12:38:52 GMT+0530

Fig 5 (d) Pump Full Status

The pump status 10 indicates that the tank is empty and that it allow the water to flow through flow sensor or turns ON the motor.



pump status:**10**
Wed May 23 2018
12:25:48 GMT+0530

Fig 5 (e) Pump Empty Status

4. DISCUSSION AND CONCLUSION

The present work has been concentrated on the design and development of both hardwares and softwares for cost effective data acquisition system (DAS), which are used for measuring slowly varying physical variables. Physical parameters such as temperature, humidity, flow rate etc., which are generally considered as slowly varying signals, are measured by the designed DAS. Design consideration of the PC based data acquisition system has been made, based on commonly available ports of the personal computers (PCs) viz. serial port.

- The Internet of Things based Data Acquisition System will be equipped with different possible ways of data transmission.
- Implementation of this automated data acquisition will help in continuous monitoring of data from any part of the world. It can also be extended to acquire different types of sensors to monitor different parameters. The implementation is economical.
- The system will be a generalized system to be used in various fields such as for medical, industrial, robotics etc. with all connectivity options and with different range options. The system is small in size and also portable.

The present work is focused around the design and development of cost effective IoT based data acquisition system (DAS), both hardwares and softwares. The designed systems are having 10-bit resolution only. Resolution can be further increased by using high resolution external ADCs. To make the designed system more flexible with better performance, the following modifications can be made

- **ADC :** ADC's resolution can be increased by selecting high resolution ADCs.
- **Controller :** Selection of advanced version of controllers for higher speed, resolution and better performance such as Raspberry pi, etc.
- **Sensor :** Selection of high quality sensors, integrated or smart sensors to get faster response speed with higher resolution and better performance.
- **Hardware and software :** Hardware and software modifications to enable long distance acquisition, remote access (wired and wireless) etc.
- **Miniaturization :** Miniaturization of the designed system by using surface mount devices (SMD components).

Then, the scope and applicability of the designed system will be increased. It can be used for the applications in various fields in physics, chemistry, life sciences, engineering, medical, geological applications etc.

REFERENCES

1. ESP8266EX DatasheetVersion 4.3 by Espressif Systems IoT team june 2015.
2. AT Command Manual - Designing Electronics in Spain www.jechavarria.com/wp-content/.../05/SIM900-AT-Commands-Manual_V1.07.pdf**.**
3. Using the ESP8266 WiFi Module with Arduino Uno publishing to ThingSpeak https://medium.com/@cgrant/using-the-esp8266-wifi-module-with-arduino-uno-publishing-to-thingspeak-99fc77122e82.
4. Arduino - Wikipedia https://en.wikipedia.org/wiki/Arduino
5. **LCD-020N004L 20 x 4 Character LCD - Vishay** https://www.vishay.com/docs/37314/lcd020n004l.pdf.