

**BUILD MOBILE APP:**

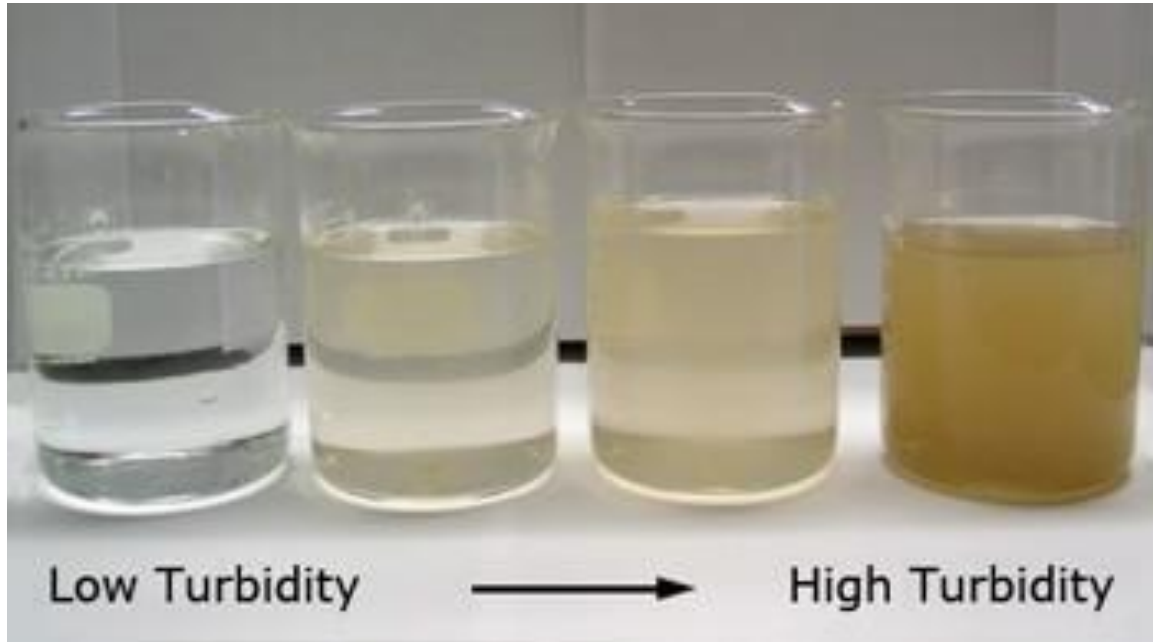
**DESIGN YOUR UI TO DISPLAY THE WATER TURBIDITY PH  
VALUES:**

**TEAM ID: PNT2022TMID14081**

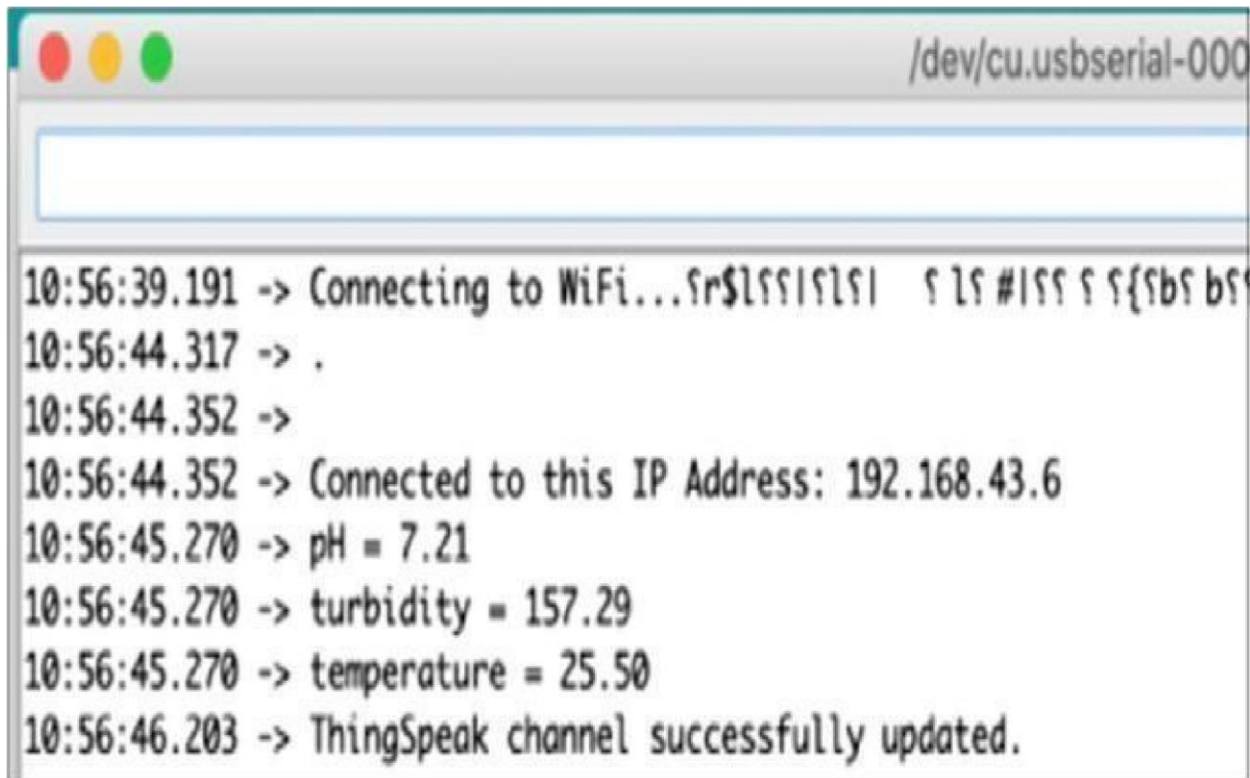
## **Materials and system design.**

The proposed domestic water temperature, pH and turbidity monitoring system consisted of a network of 3 sensors for collecting data on temperature, pH and turbidity of water. These sensors were connected to an Arduino microcontroller which processes the data before relaying it to a cloud platform through a Wi-Fi module.

the cloud platform, the information is pulled and displayed on a website. The authorities in charge are then able to monitor the data for the different water quality parameters as well as analyze the data in form of graphs. Should the data collected vary from the set standards, then the authorities are alerted instantly. Figure [1](#) shows the block diagram of the proposed system.



[Turbidity](#) is the **cloudiness** or **haziness** of a fluid caused by large numbers of individual particles that are generally invisible to the naked eye, similar to smoke in the air. The **measurement of turbidity** is a key test of **water quality**

A screenshot of a terminal window with a title bar containing three colored circles (red, yellow, green) and the text "/dev/cu.usbserial-000". The terminal displays a series of log messages with timestamps and status updates. The messages indicate the device is connecting to WiFi, becoming connected to the IP address 192.168.43.6, and then reporting sensor data: pH = 7.21, turbidity = 157.29, and temperature = 25.50. Finally, it reports that the ThingSpeak channel was successfully updated.

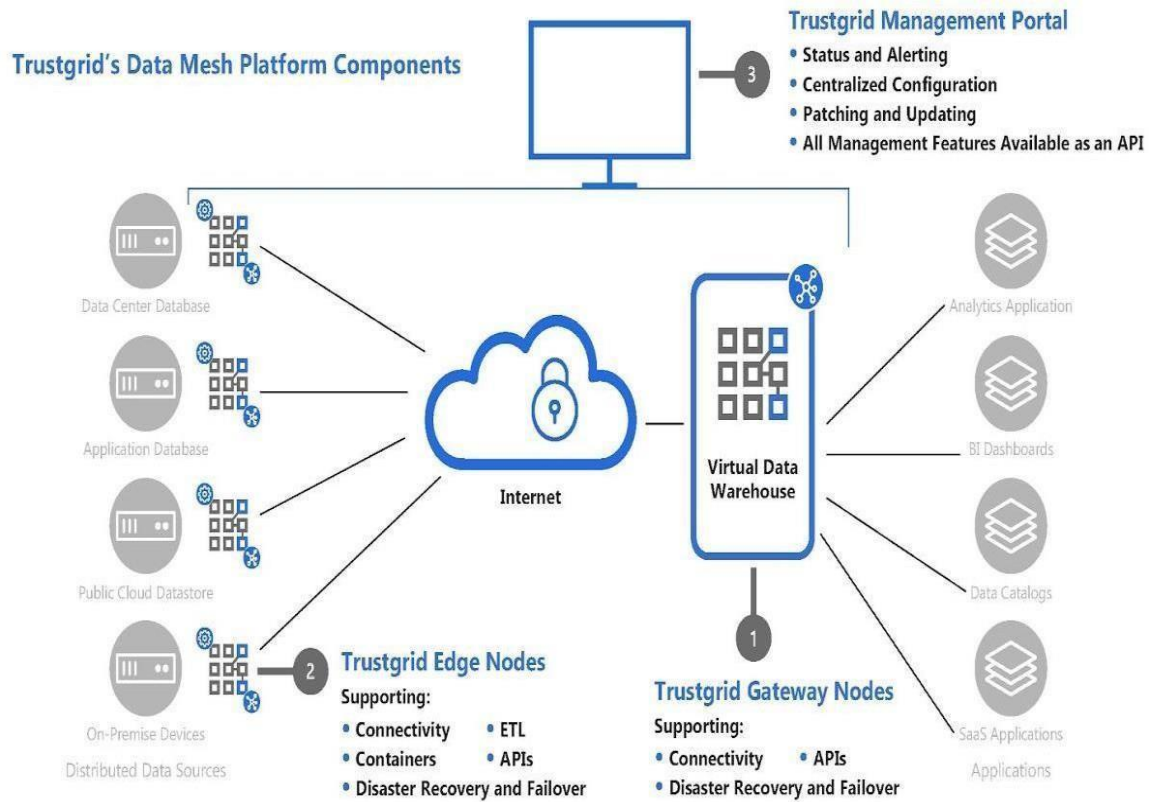
```
10:56:39.191 -> Connecting to WiFi...  
10:56:44.317 -> .  
10:56:44.352 ->  
10:56:44.352 -> Connected to this IP Address: 192.168.43.6  
10:56:45.270 -> pH = 7.21  
10:56:45.270 -> turbidity = 157.29  
10:56:45.270 -> temperature = 25.50  
10:56:46.203 -> ThingSpeak channel successfully updated.
```

When it was placed in clear tap water, the system indicated that the pH was 7.54 and turbidity was 0.00 which is in line with the NWSC standards for pumped and treated water. When placed in water with settled dirt, the system displayed that the turbidity was 2969 NTUs which makes the water unpalatable since it lies out of range of the acceptable standards (greater than 5 NTUs). When placed in muddy water, the system displayed that the turbidity was 3000 NTUs which means the water was unpalatable since its turbidity was beyond the acceptable limit. All the data were captured and displayed on the website as the most recent parameter status reading. The graphs and tabular format as presented in Additional file

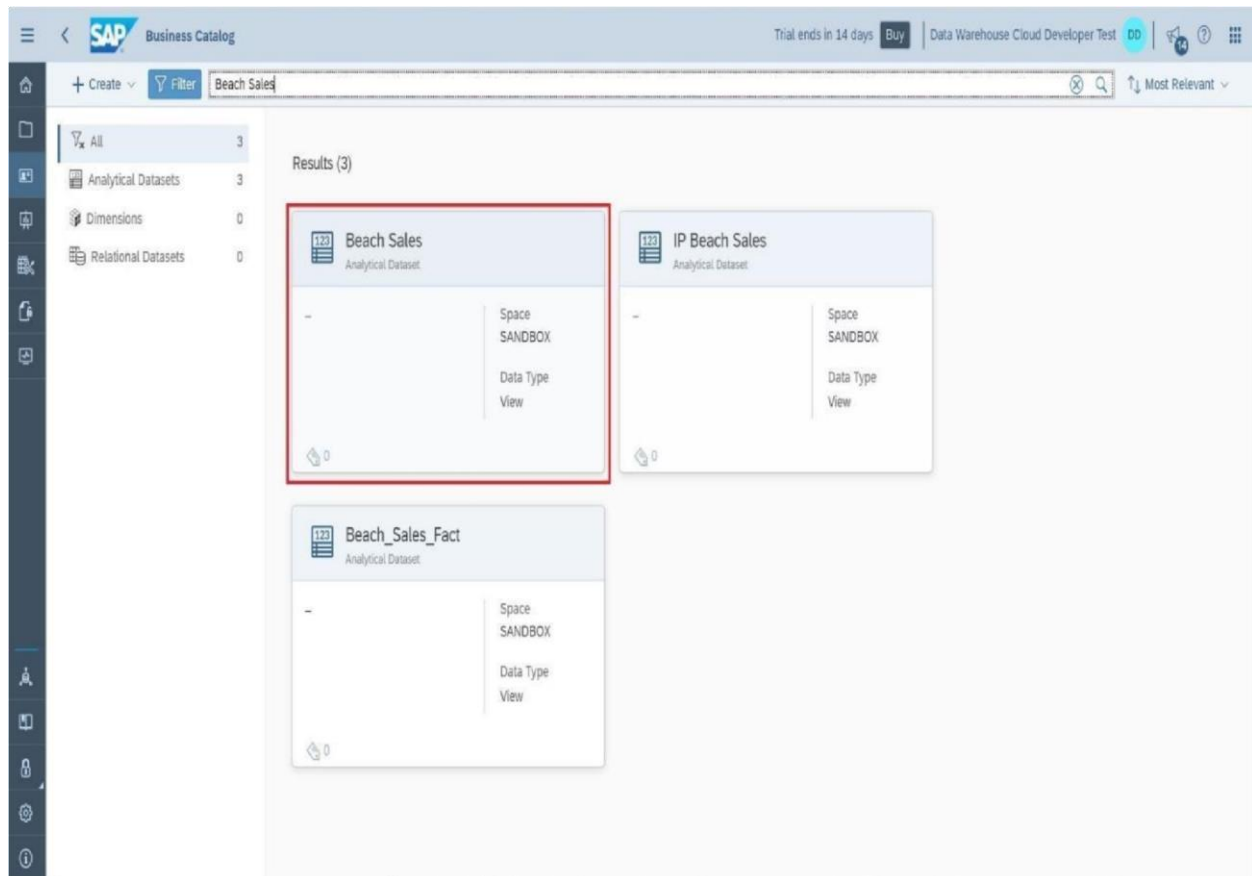
## **CONFIGURE THE APPLICATION TO RECEIVE THE DATA FROM CLOUD:**

This is created through the use of gateway nodes to create a **Virtual Data Warehouse**. This Virtual Data Warehouse allows application developers to map access to remote data points.

This software-defined gateway is run adjacent to the application it serves and can be deployed within a cloud environment or in a data center.



This Virtual Data Warehouse allows for the virtual aggregation of data so that an application (or many applications) can easily consume it. Once a data source is added to the Virtual Data Warehouse an application has secure, real-time, persistent access to that data set.



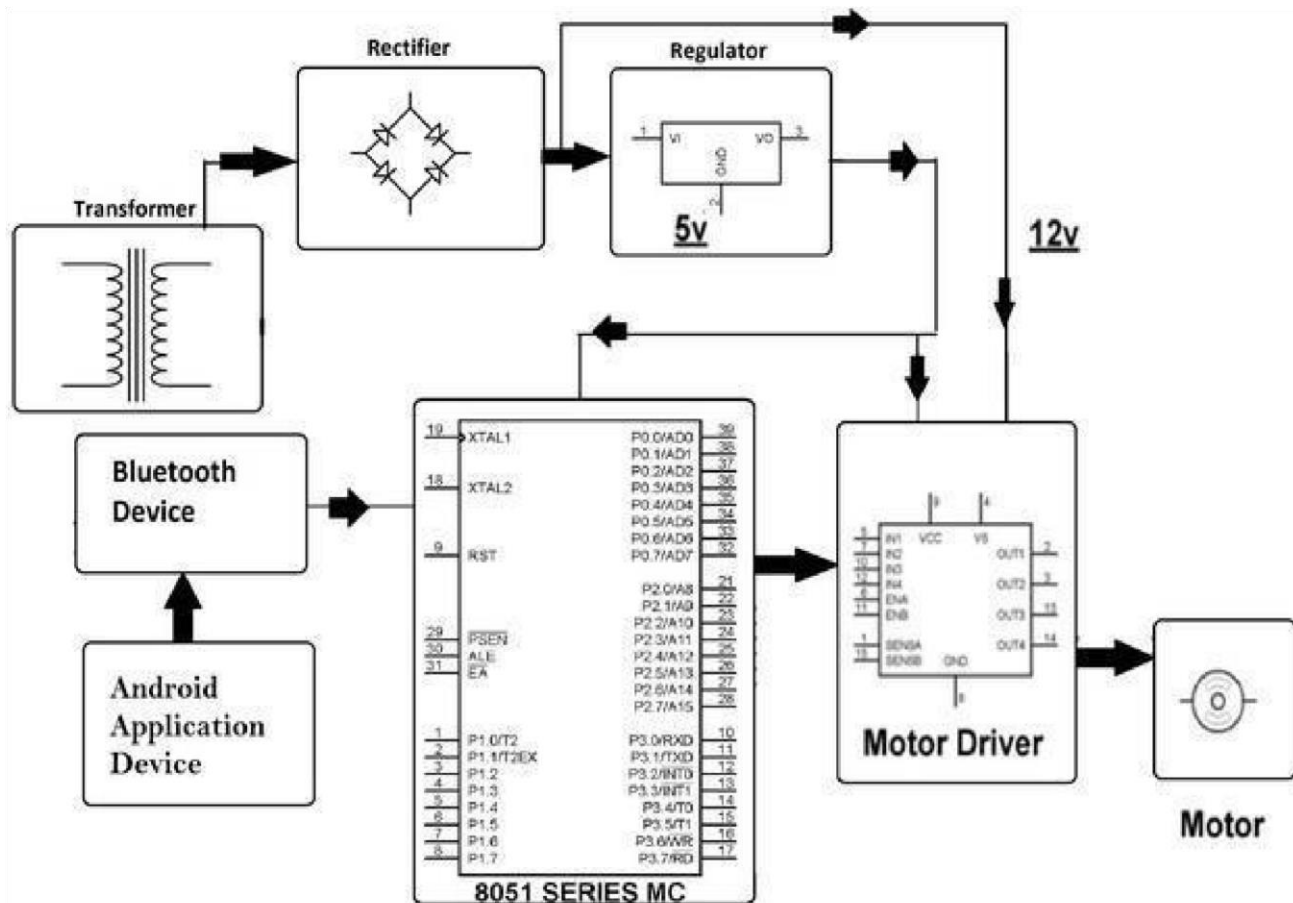
Hardware device – The hardware device is one of the easiest methods of deployment because Trust grid handles all of the software imaging, logistics and deployment support for the end-user. A hardware appliance is ideal for environments with limited onsite support.

Virtual Appliance – For those that prefer not to deal with the logistical or cost considerations of a physical hardware appliance, Trust grid's Data Mesh Platform can create a node using only software. This virtual appliance works in any environment supporting VMWare vSphere. And is ideal for environments with DevOps management and support.

Cloud End Points – For cloud-native data stores the platform supports AWS, Azure, Google Cloud and Oracle Cloud to easily form endpoint connections to public cloud environments.

## **CONFIGURE THE MOBILE APP FOR CONTROLLING MOTOR USING BUTTONS:**

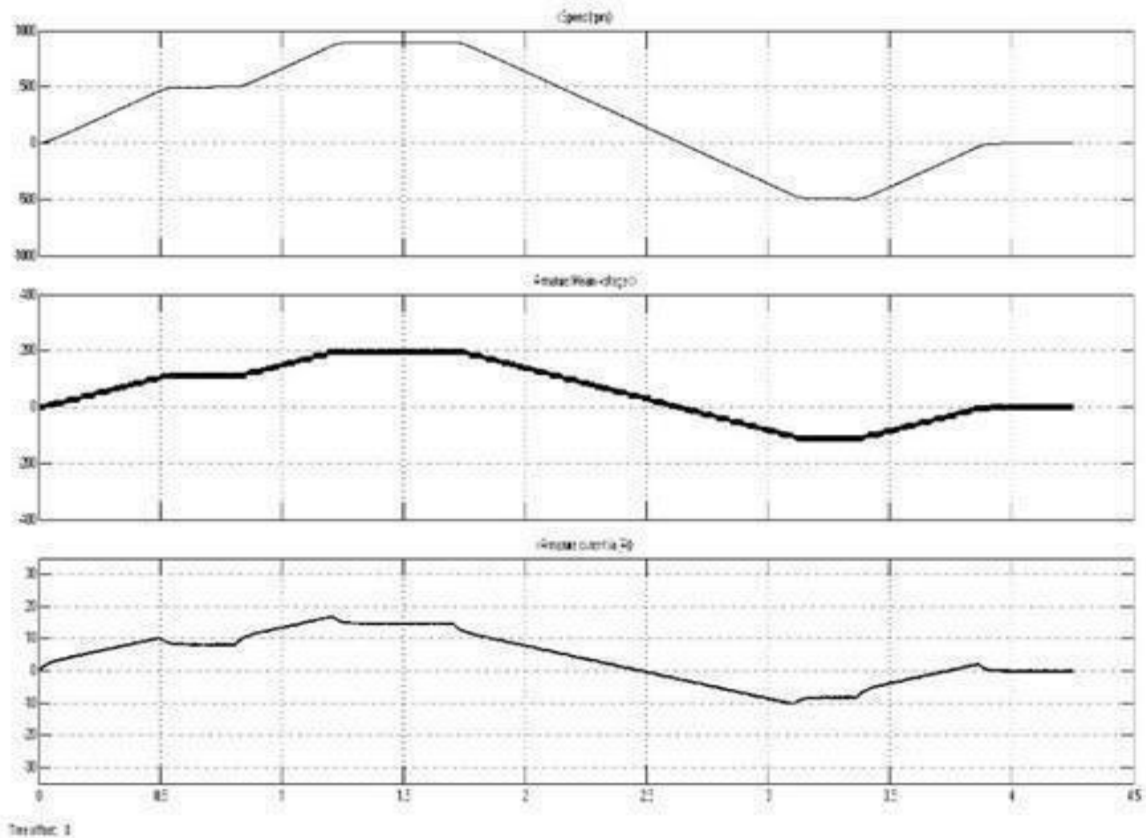
This system DC motor Controller by Android is developed to control the speed of the DC motor in both clockwise and anticlockwise direction. For this DC motor is interfaced to the 8051 microcontroller. A Bluetooth modem is used to receive direction commands and PWM commands. When an Android device sends commands, it is received by the Bluetooth modem which then sends the commands to the microcontroller. The microcontroller then controls the DC motor through motor driver. The entire system is powered by 12V transformer. LCD display is used to show the status and the speed of the DC motor. The android application is used to control the entire system. The start button is first clicked to start the motor and then the motor can run in both clockwise and anticlockwise direction. Simultaneously the status of the system is displayed on the LCD screen and also the speed of the DC motor is displayed on the screen. Thus the speed of the motor can be increased or decreased in clockwise or anticlockwise direction with the help of this android application.



Speed control of Four Quadrant Operation of DC motor

In this method there are two cascade PI control, comparing speed ( $\omega$ ) and current ( $I_a$ ) as speed controlling and current controlling. First the actual speed ( $\omega$ ) of the motor is compared with a set speed ( $\omega^*$ ). Second the current ( $I_a^*$ ) which comes from speed controller is compared with actual armature current ( $I_a$ ). Fig.1 is the schematic of Simulink diagram for the DC motor drive. The regulation switch produces the current reference, either as provided by the speed controller during speed regulation or computed from direct from the torque reference provided by the user during torque control.





illustrates the developed “Mat lab-GUI” for the designed universal motor drive. The GUI application comprised of 6 windows as main window and other five windows relevant to each motor controlling parts. Fig.3(a) shows the main screen which requires the users to select the motor to be operated. If user selects AC three phase motor, Fig.3(b) window is displayed on the screen. Then user can set the desired speed and push start button to start the relevant operation. If the user selects DC motor, user can select any controlling method from main window and can get relevant window such as Figure 3(c). These windows display the actual speed of the motor, voltage of the motor and current of the motor. When the stop button is selected, all the operations are stopped.