Feasibility Report: Digital Water Metering System Design (DWMS)

Sarwan Shah

February 8, 2025

Abstract

This report aims to conduct a feasibility study on the design of a Digital Water Metering System (DMWS) that is cost-effective, self-powered and IoT enabled, with the goal of helping Karachi Water and Sewerage Board (KWSB) deploy a network of such metering systems across the city of Karachi to efficiently bill customers.

Introduction

The Karachi Water and Sewage Board currently bill customers according to their plot size. This billing process is inefficient and inadequate as any customer can consume as much water as they want without any increment in the bill. This encourages wasteful consumption practices. This lack of revenue generation means that KWSB is not a viable organization as of now and has to depend heavily on government subsidies and funds to remain operational.

The idea of developing a solution or device that can measure the flow rate of water in a pipe with high accuracy to get accurate estimates of water volume consumption is not entirely novel. There exist several solutions globally that cater to the need for metering water connections or pipes. However, one challenge around these existing solutions is that they are expensive and availability within Pakistan is scarce. This makes them unfeasible for large scale implementation locally, as KWSB already lacks the necessary funding. Another major shortcoming with most of these solutions is that they are not IoT (Internet of Things) enabled. They are unable to transmit their readings over a network and require taking manual readings. This adds to manual meter reading labor costs. Furthermore, existing setups cannot be self-powered, adding to their carbon footprint and maintenance costs.

These factors have motivated us to develop an integrated solution that fulfills our need requirements while maintaining cost-effectiveness and longevity. These requirements can be summarized as follows:

- A device that can measure the flow rate of water through a water pipe with a high degree of accuracy.
- An electronic system that can continuously receive, store, process, and transmit this information wirelessly.
- Our system should not rely on any external source of power.

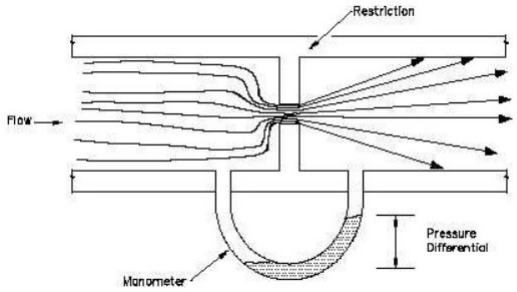
The next section provides a detailed literature review of the two critical components and constraints that go into consideration in our system: flow rate measurement and self-powering abilities. This is to justify the practical design choices we make when modeling our system. In the section after that, we will layout the overall modeled systems design, explaining our rationale for choosing the devices that are to be integrated into our system and potential pathways that could be taken to achieve the desired results. Lastly, we go through the power and cost feasibility of such a setup.

Reviewing Existing Designs & Approaches

Available Types & Principles of Fluid Flow Measurement

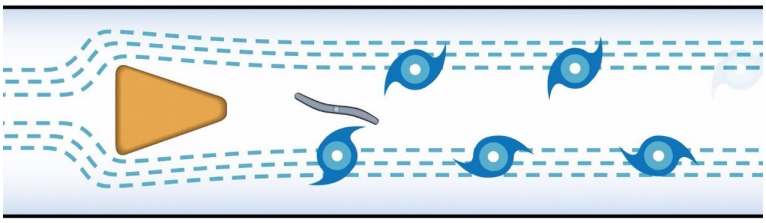
Since Fluid Flow Measurement in the form of water will be a key feature of the system, let us explore through the wide variety of principles and their congruent flow-meters that are utilized for the volumetric flow rate measurement.

The most traditionally used are the differential head type flow-meters, with a heavy utilization in industries due to high accuracy, precision, and robustness for a wide variate of industrial applications. These are based on the principle of differential pressure at two points in a fluid due to a difference in fluid velocity. However, they are not very suited for installation in an urban water network due to their complex installation requirements, conditions, and maintenance. Head Type Flow Meters:



A similar reason also follows for Variable Area type Rotameters Meters, in addition to their inaccuracies [2][3], or Mass Displacement type Inertial Mass Flow Meter which in addition require complex conditioning and robustness [3], or Ultrasonic & Electromagnetic type meters which are subject to variables like fluid bubbles, conductivity, and power-consumption [2].

The two more recently developed electro-mechanical fluid flow meters are Vortex and Coriolis flowmeters. The former positive-displacement type flow meter has been promising due to its robustness due to no moving parts and high accuracy [2]. It utilizes the von Karman effect which involves the generation of a stream of alternating parallel fluid vortexes by an obstruction shedder, leading to fluid vibrations detected as frequency changes by a transducer. On the other hand, the latter Mass-displacement type, also offers high accuracy and detection of additional variables such as density and viscosity [3], utilizing the conservation of angular momentum of fluid on a free-suspended mass of fluid, detecting it's net phase change to small impulses by an actuator. Vortex Flow Meter working principle:

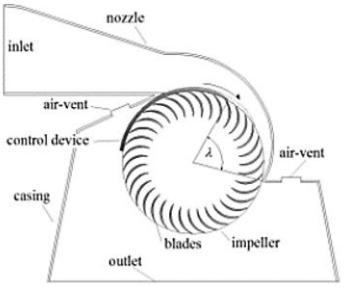


However, neither has been utilized for a system that involves Self-Powered Fluid Flow Measurement to the best of our knowledge yet. The flow meters utilized so far in Self Powered Fluid Flowmeter designs have been Turbine based flow meters [4][5][7][8][9]. This is primarily due to their easy integration with pipes and water flow, and use as both: a power-generation source and sensor. This makes them an ideal choice for our Digital Water Metering System (DWMS) model.

Design Considerations

Turbine Design

An important consideration to maximize power generation efficiency was towards the design of the turbine or impeller that rotated the generator. One design used a 7-wheel Pelton wheel steel roller bearing coupled design [4] based on the experimental findings of [7], which concluded 7-wheel semi-spherical blades with a radian curvature of 0.5 for pipes less than 50 mm in diameter, higher attack angles (0 - 36 degrees) for pipes in the size range 50 - 300 mm, and a double propeller (generator front-back mounted) shaped design for pipes greater than 300 mm, with a perpendicular installation. Furthermore, a Miroslav Cink's cross-flow radial turbine design was also presented by [10] to improve efficiency by increasing the fluid's contact time with Turbine and preventing reverse and overflow issues. The design is shown below:



Generation Design and Power Generation

Generator construction features were critical to maximizing generation and efficiency. All proposed systems involved 3-Phase Generators for maximizing output [4][5][8][9]. [7] suggested surrounding the turbine magnets with a metal ring to reduce flux leakage, reporting an improvement of 300% over for turbines to be installed in large pipes (¿ 300m).

A roughly linear trend was observed between flow rate and power generation in [4][5][8]. The 12-winding generator design in [4] achieved an efficiency of 50%, generating 5.3 V, 25 mA, and 0.13 W of power for a flow rate of 10 L/min and a maximum of 21 V, 115 mA, and 2.39 W at 45 L/min. [5] used a star-circuit 2-pole ring-magnet induction generator sourced to a 83% efficiency energy harvesting circuitry, allowing a power transfer of 0.15 W at 10 L/min, maxing out at 0.3 W at 19 L/min. [8] designed an 8-Pole 6-Slot coreless cogg-resistance-free axial-flux permanent magnet (AFPM) generator rated for 1 W at 1200 RPM, and using one only one phase it was able to power its setup that involved no wireless transmission for 720s given an energy harvesting period of 18s at 10 RPS.

Furthermore, [8][9] also demonstrated the longer terms feasibility of using an existing commercial use 12V-10W micro-turbine generator after little modification to generate sufficient power for an SPFFM system with strong optimizations and a compromise of 1 meter reading per hour with the wireless transmission in the case of [9]. Strong optimizations and analysis also conducted in [4].

Energy Harvesting & Regulation Mechanism

Given that the turbine generators proposed in the reviewed studies involved 3-Phase generators, an efficient mechanism to rectify, use, and store this energy was essential. [4] used a full-bridge to rectify the AC phase coupled via a high-freq transformer to charge AAA Na-Cd batteries and isolate the circuit-load from the turbine generator while the electronic circuitry performed other operations, while supply was regulated via an ultra-low dropout regulator to minimize losses. On the other hand [5] presented a more mature design using Lithium polymer batteries and features involving over-voltage protection, deep-discharge protection, battery charging, and monitoring, voltage conversion, and safe-start, active and sleep modes, with the aid of LTC2935 power management IC. Unexpectedly, [8] proposed a battery-free solution that utilized a simple half-wave rectification circuit only utilizing a single-phase under the rationale that complete 3-Phase 6 Pulse diode-based or IC based rectification significantly increased the generator load and consequently torque making it unsuited for operation at lower flowrates, thus bringing into light a unique concern. A 3-Phase 6 Pulse rectifier was also utilized in [9] [10], however no such concerns as those presented in [8] were raised, while [10] did not provide any power analysis results at all.

Signal Conditioning

In our review we came across two distinct types of SPFFM systems: one type extracted the power and signal from a 3-Phase turbine flow-meter setup [7][8], while the other type used a small turbine for signal generation, and a relatively larger 3-Phase turbine for power generation [4][5]. The reason for the latter can be speculated for around the challenges with AC signal conditioning for accurate detection as seen in [9], whereas DC rectification of AC signal lead to a voltage drop in the signal and directly detecting the AC signal involved complex filtering techniques for accuracy. In addition, it offered good accuracy only for a flow rate range of 250 - 650 L/hour. Using a separate turbine allowed for simpler low-signal for easier signal processing [4].

Calibration

Calibration was not generally a focus point for the scope of most reviewed papers. This is perhaps rooted in the idea that WDN's do not require industrial levels of accuracy. [9] presents only a piece-wise mathematical

function between signal voltage and flow rate derived from the experimental data and the accuracy of which is questioned by the paper itself. [9] initially uses a line of best-fit based on experimental data of the signal voltage vs tachometer readings, after which the line of best fit was further optimized accuracy by comparing the flow rate and volume of consumption for a fixed volume of water over a time interval.

Other Considerations

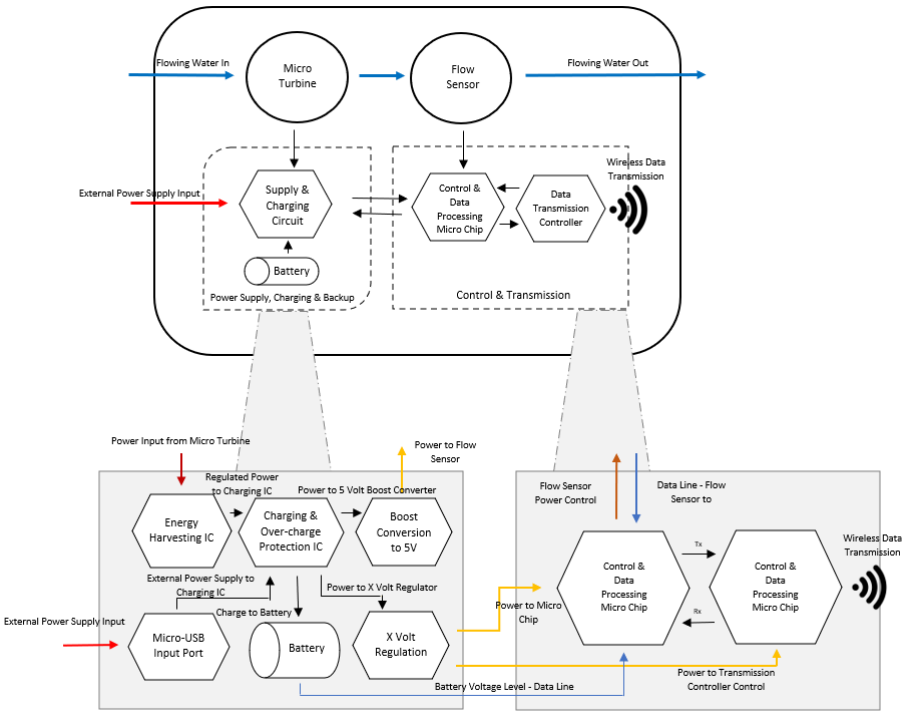
Another important variable that was found to impact the flow rate accuracy and the flow of water adversely was the pressure loss due to the obstruction that a flow meter or turbine generator device posed by extracting kinetic energy from it. [4] had interesting findings in this regard, and concluding that such a loss was insignificant to have any impact on the flow rate itself or an in-direct inaccuracy.

Transmission Method

The last observed consideration for such a system was found the be the medium of Wireless Transmission. This is an important consideration as wireless transmission can be a major source of power consumption as demonstrated by [4], which utilized the ZigBee technology achieving an outdoor transmission range of 80 meters and indoor 30 meters. [5] as a European Commission funded project, it utilized the Wireless M-Bus Standard doing long-range transmission over the 169 MHz RF band. [8][9] both proposed new technology Bluetooth solutions due to their low power consumption.

System Design Overview

Having studied the necessary considerations and existing work done in the domain. This is the final system's layout we concluded with:



Choice of Turbine

The Turbine will responsible for turning the linear flow velocity of water into angular velocity for rotating the AC Generator of choice. Based on our research, as presented above, we have decided to use a turbine design that will have 7 semi-spherical blades with radian curvature of 0.5, as this design was experimentally proven to have generated the best power generation efficiency [7]. This design will serve as a starting point, which could be improved with practical testing alongside the generator. The design could be made on SolidWorks and then 3D printed for the proof of concept prototype. A sample design is shown below:

Choice of Generator

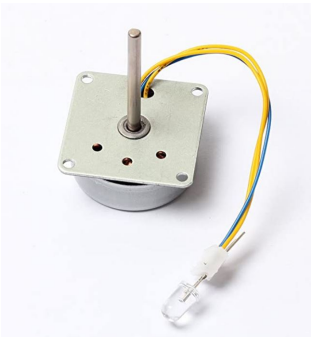
To translate the rotational energy provided by the turbine, we will utilize a CrocSee mini 9-Pole 3-Phase AC Brus Generator. This will act as the main source of power for our system. The rationale for choosing this motor is its cost-effectiveness for our purpose and our limited scope of resources for this. This was the only small commercially available decent AC Generator we could find, as is available on both: Aliexpress and Amazon. The specifications are tabulated below:

Parameter	Rating
Cost (PKR)	330
Min. Voltage Output	3 V
Max. Voltage Output	24 V
Rated Speed Range (RPM)	300 – 6000
Current Output	50 mA – 1000 mA
Rated Power	0.15 - 12 W

The Pakistan Institute of Specifications is at large similar to the British Standard of Specifications for household water supply. This standard defines the house water pressure supply to be 40 PSI, which equates to a flow-rate of about 27 liters/min. This flow rate should be able to easily provided between 500 - 1000 RPM for our turbine, and consequently generator. This should allow the generator to easily provide a power of up to:

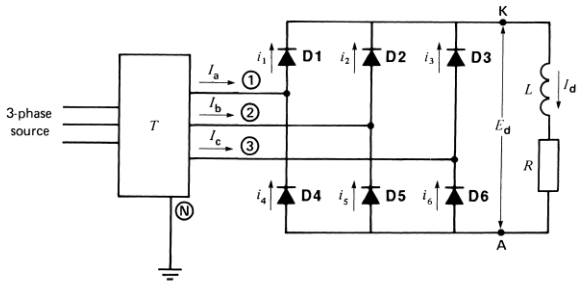
RPM Range	Power Produced (W)
500 - 1000	0.15 - 3 W

While this generator serves as proof of concept for the bare minimum possible that could be achieved with a commercial generator, more efficient and better generators could be designed for large scale production for such meters. An image of CrocSee mini 3-Phase 9-Pole Brushless AC Motor is shown below:



Choice of Rectification Circuit

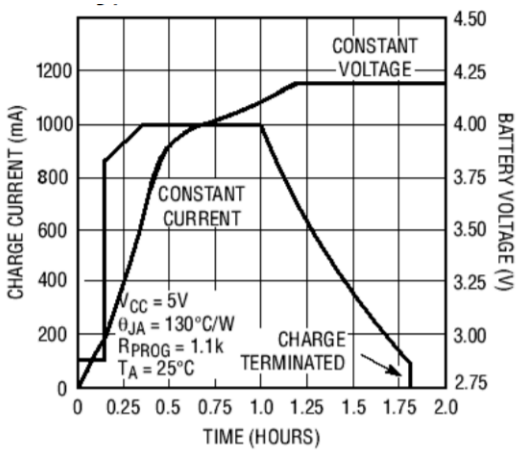
This will be done using 3-Phase 6 Pulse Full Bridge Rectifier to utilize all 3 AC Phases, with low-voltage dropout diodes to minimize loading effects on the generator and maximize output efficiency. One potential candidate for usage in this application would be the MAX40200 Ultra-Tiny Micropower, 1A Ideal Diode with an Ultra-Low Voltage Dropout of only 43 mV, this chip is often utilized in USB-Powered and Toy application, as such showing its suitability for low-powered applications, such as ours. The following bridge circuit will help achieve this task:



The circuit is a better choice, if we can minimize its voltage drop, as it manages to rectify 6 Pulse (3 positive half cycles & and 3 negative half-cycles), maximizing efficiency, in comparison to only 2 Pulse and 3 Pulses in the full-bridge and 3-Phase 3 Pulse rectifier. This is what makes it a desirable choice. The output from this rectification will feed a high-efficiency 5V regulator such as MIC5129, which only has a quiescent current consumption of 30 uA, which will then pass it on to a charging circuity.

Choice of Battery

The battery we will use for our setup will be a 3.7 V 6000 mAH Lithium Polymer battery. This is because these are easily available, cost-effective, and widely used in electronic setups because of their relatively flat discharge curves, high energy density, a high number of discharge cycles, and quick-charging. However, they do require a specialized 3 stage charging process: Pre-charge, Constant Current Charge, and Constant Voltage Charge. It is summarized by the image below:



This will be handled by the TP4056 1A Lithium Ion Charging IC, which works at the specification of the above-mentioned charging process and has overcharge protection, in-fact the figure was extracted from its datasheet. This IC will act as an uninterruptible power supply to provide power to the rest of the electronic circuitry i.e. the MCU, Wireless Transmission Unit, and Flow Meter, from either the battery or the AC generator, given it is generating sufficient current and voltage.

Choice of Flow Meter

We will utilize G1-FS400A Hall-Effect type flow meter, due to its relatively low power consumption, wide availability, low cost, fairly high accuracy (97%), easy calibration, and easy integration with MCU's. The flow sensor utilizes a hall-effect sensor which based on the speed of rotation detects and varies the width of the square pulse sent along a data-line, which are then detected by an MCU and translated to its flow-rate.

Parameter	Rating
Size	1"
Min. Working Voltage	4.5 V
Max. Working Current	15 mA
Working Voltage Range	5 – 24 V
Flow Rate Range	1 – 60 L/min
Pulse Frequency to Flow Rate Constant	4.8

Furthermore, the 1-inch size is ideal as most household pipes lie close to this range of diameter, and hence can be easily coupled without losing accuracy. The calibration for this flow-meter could be done easily by timing how much time it takes for a fixed volume of water to discharge vs the volume discharged based on the meter's output flow rate.

Choice of Microcontroller

For this we will use the Attiny85 as our setup does not require a lot of processing power and using the Attiny85 will save on a lot of power and has a small form factor. Other options such as the NodeMCU or Arduino though offer more I/O pins and processing power but have a strong disadvantage of high power consumption.

The Attiny85 will be responsible for receiving data pulses from the flow meter and processing them into the flow rate. It will use pulse excitation to trigger the flow sensor for only a few milliseconds every second to retrieve this data to save on the overall systems power. In addition, it will be responsible for storing this data for a fixed time interval, before triggering the Wireless Transmission device to transmit it all to KWSB servers in one go, in bulk form. Lastly, the Attiny85 will also be responsible for keeping a check on the batteries' voltage levels using its in-built ADC. This allows the Attiny85 to issue a warning, backup data, and minimize all other operations to save power in the extreme event that the system is out of power. Hence, allowing the battery to either charge sufficiently or warning relevant stakeholders about the issues apriori.

Choice of Transmission Controller

For the wireless transmission we will utilize the ESP8266 WiFi chip. This is ideal because of its low power consumption compared to its other counterparts such as the NodeMCU. Furthermore, it is easy to integrate with such setups given its wide availability, being a popular choice for IoT based projects. The ESP8266 will mostly stay in deep sleep mode, saving power, and will only be triggered by the Attiny85 for transmission of data at fixed intervals of time. The WiFi will use the lightweight MQTT Protocol to transmit this data to KWSB servers over the web, where it can be processed and added to the customer's bill. The MMQT Protocol works on a subscribe and publish methodology, in which devices can either subscribe to a topic on the server if they wish to receive data from it, or they can publish data to a topic. We explored this MQTT Protocol in further detail and tested it on an ESP8266 device that was available with us. The snapshots of the designed application and the respective codes are attached in the appendix.

Feasibility Analysis

In this section, we will look at two of the critical components that will ensure whether that this system is practically realizable for the specifications we require or not, primarily, the need to be self-powered and its cost-effectiveness.

Power Feasibility

Using Pulse Excitation and interval based data transmission and collection, below are the estimated power consumption statistics of our system based on values extracted from data sheets of the mentioned components. All values were extracted in mA and converted to mW (P = VI) using knowledge of the voltage consumed by these devices. We are also assuming the generator is generating power only 6 hours a day to stay on the safe side.

Component	Nominal Power Draw (mW)	Avg. Power Per Second (mW)
Rectification Circuit	15	15
TP4056 IC	0.007	0.007
MIC5219 IC	0.02	0.02
ESP8266 - WiFi Operation (1/600 s)	264	5
ESP8266 - Deep Sleep Mode	0.07	0.07
Attiny85 - Normal operation	7	0.4
Attiny85 - Sleep Mode (950 ms)	0.07	0.07
G1-FS400 - Normal operation (10)	75 W	0.7
Total	361	21.3
Component	Nominal Power Generated (mW)	Avg. Power Generated Second (mW)
AC Generator	300	75

We can see that even with the assumption of only being able to generate power for 6 hours a day, the power output by the generator provides more than ample power to run the system (75 mW generated vs 21.3 mW consumed), even if we consider these estimates to be optimistic, it does demonstrate the potential of further refining such a setup for the possibility of more realistic, but similar results, that can allow the setup to be power independent.
Furthermore, the 6000 mAH battery can provide a backup of 22,200 MWh. This means that even without any power from the generator it could run for roughly (22200/21.3) or roughly 1000 hours, which equates to about 41 days. This show's how reliable such a setup could be.

Cost Feasibility

Component	Cost (PKR)
G1-FS400A	1200
CrocSee 3-Phase AC Generator	400
ESP8266	300
Attiny85	250
TP4056	50
MIC5219	50
Miscellaneous	250
Total	2500

Hence, we can also see the strong cost-effectiveness of our system, costing only PKR 2500 for a setup that could potentially be self-powered and IoT enabled.

Conclusion

To conclude, through this report we have demonstrated the feasibility of the proposed system design that can act as a self-powered Digital Water Meter System, which can be utilized by KWSB to bill its customers in an efficient manner, at a very reasonable cost, with a wide array of desirable features. This too at a multitudes less of the cost compared to globally available commercial solutions which could cost the institution at-least PKR 20000 per unit, if not more.

Appendix

A snapshot of the application and the Python and ESP8266 script codes are attached below:

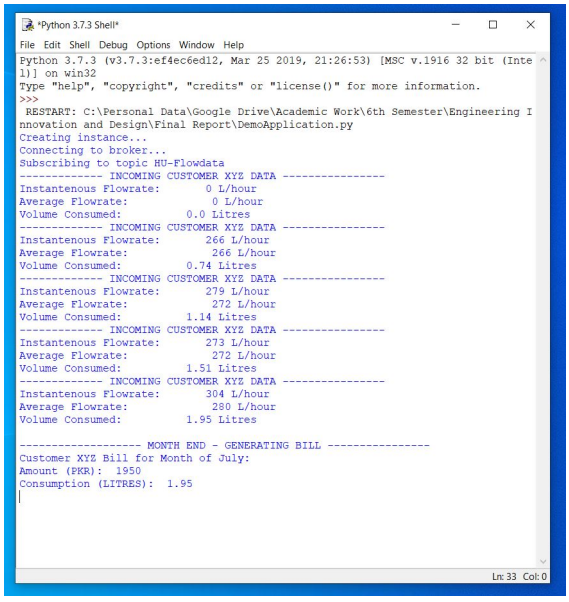


Figure 1: This is a demo application of the working of the MQTT Protocol over WiFi. This application is receiving data simulated and sent from ESP8266 over the WiFi utilizing the MQTT Protocol, utilizing a demo server available for public testing.

```
#----------------------- Python Demo Application Code -----------------------
import time
import paho.mqtt.client as mqtt
global incomingData

def on_message(client, userdata, message):
      if int(str(message.payload.decode("utf-8"))) > 10:
          dataSource.setData(int(str(message.payload.decode("utf-8"))))

class MMQTProtocol:

    def __init__(self, broker, topic):
        print("Creating instance...")
        client = mqtt.Client("AHSAN-PC")
        print("Connecting to broker...")
        client.connect(broker)
        client.on_message = on_message
        client.loop_start()
        print("Subscribing to topic", topic)
        client.subscribe(topic)
        self.data = 0


    def getData(self):
        return self.data

    def setData(self, incomingData):
        self.data = incomingData


def average(flowrate, timeData):
        total = sum(flowrate[-60:])
        n = len(flowrate[-60:])-1
        if n == 0:
            return total
        else:
            return total/n

def volume(flowrate, timeData):
        return (average(flowrate, timeData)/3600)*timeData[-1]


dataSource = MMQTProtocol("broker.mqtt-dashboard.com", "HU-Flowdata")

flowrate = []
timeData = [1]
avg = []
vol  = []
rate = 1000

count = 1

while True:
      count += 1
      flowrate.append(dataSource.getData())
      a = int(average(flowrate, timeData))
      v = round(volume(flowrate, timeData)*5,2)
      print('------------- INCOMING CUSTOMER XYZ DATA ----------------')
      print('Instantenous Flowrate:   ', flowrate[-1], 'L/hour')
      print('Average Flowrate:         ', a, 'L/hour')
      print('Volume Consumed: ','    ', v, 'Litres')
      timeData.append(count)
      avg.append(a)
      vol.append(v)
      time.sleep(1)
```

```
    if count == 6 or count == 20 or count == 30:
        print(' ')
        print('------------------- MONTH END - GENERATING BILL ----------------')
        print('Customer XYZ Bill for Month of July:')
        print('Amount (PKR): ', int(vol[-1]*rate) )
        print('Consumption (LITRES): ', vol[-1])
        avg = []
        vol = []
        time.sleep(1000)
        print('------------------ RESTARTING RECORDING CYCLE----------------')
```

```
#---------------------------- ESP8266 Publish Code -----------------------------
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
const char* ssid = "ZONG MBB-E5573-6A5C";
const char* password = "94310071";
const char* mqtt_server = "broker.mqtt-dashboard.com";
char strCalc[5];

WiFiClient espClient;
PubSubClient client(espClient);

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
      Serial.print("Attempting MQTT connection...");
      // Create a random client ID
      String clientId = "ESP8266Client-";
      clientId += String(random(0xffff), HEX);

    // Attempt to connect
    if (client.connect(clientId.c_str())) {
        Serial.println("connected");
        // Once connected, publish an announcement...
        }

      else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
```

```
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
       delay(2000);}
  }
}

void setup() {
  //pinMode(BUILTIN_LED, OUTPUT);  // Initialize the BUILTIN_LED pin as an output
  Serial.begin(115200);
  setup_wifi();
  WiFi.mode(WIFI_STA);
  wifi_set_sleep_type(LIGHT_SLEEP_T);
  client.setServer(mqtt_server, 1883);
}

void loop() {

  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  delay(990);
  int calc = random(250, 350);
  snprintf (strCalc, 5, "%ld", calc);
  //Serial.print("Publish message: ");
  //Serial.println(strCalc);
  client.publish("HU-Flowdata", strCalc);
  delay(10);
  }
```

References

[1] A. T. J. Hayward, Flowmeters: A Basic Guide and Source-Book for Users-Macmillan Education. THE MACMILLAN PRESS LTD, 1979.

[2] M. Pereira, "Flow meters: Part 1," 2009.

[3] "Fluid flow measuring devices used in an inertial measurement unit based on different flow measurement sensors."

[4] S. Wang and R. Garcia, "Development of a self-rechargeable digital water flow meter," Journal of Hydroinformatics Vol 15.3, 2013.

[5] R. G. D. H. A. W. Y. M. P Becker, B Volkmer, "Energy autonomous wireless water meter with an integrated turbine-driven energy harvester," 2013.

[6] R. C. Baker, Flow Measurement Handbook: Industrial Designs, Operating Principles, Performance, andApplications. Cambridge University Press, 2016.

[7] R. G. Song-Hao Wanga, Ronald Jos e Doblado Perez, and J. Chend, "Development of pipe flow generators, Advanced Materials Research Vols 233-235, 2011.

[8] R. G. Wang Song Hao, "Development of a digital and battery-free smart flowmeter," Energies, 2014.

[9] P. H. J. C. Xue Jun Li, "Design and implementation of a self-powered smart water meter," Sensors,2019.

[10] V. Ramanathan, "Self-powered flow rate sensor," 2010.