Energy Harvesting Water Flow Meter

Nick Yu
Delft University of Technology

***Abstract*—Water is one of the most valuable resources. Naturally, as the human population increases, so does water demand. Statistics have shown that water scarcity is an increasing issue, also in more developed countries. Meanwhile, domestic households waste considerable amounts of water due to the unnecessary use of appliances and fixtures. In an effort to lower household water consumption, it would be handy to be able to monitor the water usage of each appliance, faucet, shower, etc. — and thus know who the biggest consumers are, and where to reduce usage. This paper presents the design of a "plug-and-play" water flow meter, which at its core is a small turbine that is installed between a water source and a fixture. The flow meter itself is powered by the turbine, but also measures the amount of power the turbine generates — from which it determines the amount of water that has passed through. The water usage is then communicated to the user using wireless capabilities on the flow meter. Since the device is self-powered and does not require any cables, minimal modifications needs to be made to the environment it is placed in. The idea is that each water source could be equipped with such a flow meter, enabling smart analytics around the total water usage.**

I. Introduction

Water scarcity is a problem that is growing due to factors such as climate change; according to the WWF, by 2025, two-thirds of the world's population may face water shortages [2]. The average household uses water in many appliances throughout the day, but many people do not consider how much water is used. In the US, approximately 90 gallons of water is wasted by a single household, per day [4]. One way to reduce consumption is by monitoring just how much water is used by each part of a household's water network. In addition to improving conservation efforts, this also allows the detection of leaks, lowers water utility expenses, and can be used to improve a household's plumbing system.

Commercial systems such as Flume[1] exist, which track the total usage using the existing mains water meter. But for water usage monitoring to be effective, you would ideally like to track the individual usage of each appliance and water fixture in addition to the total consumption. To measure the water flow through a pipe with a sensor, the sensor must sit between the water source and the sink. This poses a challenge, since, for most household cases this connection is in hard-to-reach places (e.g. behind appliances, in the wall), and usually has no power source nearby. Moreover, the sensor data needs to be read out and somehow transferred from these hard-to-reach places to the user.

In this paper, we try to answer the question: *can we design an inexpensive energy harvesting water flow meter?* We address this question by presenting the design of a battery-free, self-powered, wireless flow sensor. The sensor is powered by an inexpensive turbine, also used as the input to the sensor. The turbine is placed between the source and sink, and the generated power is used as a metric for estimating water flow.

[1]https://flumewater.com/product/

This idea of using a turbine as a water meter comes with a set of challenges. Namely, how do you accurately estimate the water flow from generated power? A second challenge is creating a system that does this in a power-efficient way, as the power generated by a small turbine is generally low.

To summarize, this paper makes the following contributions:

- We investigate methods to estimate water flow using the power generated by a turbine as a metric.
- We propose a simple and inexpensive design of a self-powered water flow sensor.
- We build a prototype and perform experiments to validate the discussed method for estimating water flow.

Next to the prototype, we also designed a PCB version of the design with the entire electrical system, which is shown in subsection IV-D. This paper is organized as follows: section II will go over the method used to estimate flow rate, section III evaluates the technique using experimental data. In section IV we present the details of the final implemented system, section V will go over the results from this research and provide steps to follow up on this paper as future work. Lastly, some concluding remarks are given in section VI.

II. Designing a self-powered flow meter

An easy way to harness energy from a fluid is by converting the fluid's energy to electrical energy by utilizing a turbine. To calculate total water consumption over some time period, you need to know the water flow rate at each timestep. The flow rate can be measured using an additional sensor (e.g. differential pressure, velocity, positive displacement) — doing it with a separate sensor would add additional cost to the system. We aim to instead use the measured generated power by the turbine to estimate the flow rate.

Assuming we have a small turbine situated in a pipe, we can model the total available hydropower generated by the turbine using the pump equation:

$$P = \eta \rho g Q \Delta H \tag{1}$$

Where ρ is the fluid density, g is the gravitational constant, H is the total energy head, and η is the efficiency ratio. For this example we consider a horizontal Pelton-style turbine with equal-sized pipes at both ends, Figure 1 provides a diagram of the setup used. Due to the conservation of energy, we know that the energy at point 1 should equal the energy at point 2, minus some losses due to, e.g. friction, and minus the energy used by the turbine. The total flow in and total flow out of the system is equal, assuming that the pipe diameter at the intake

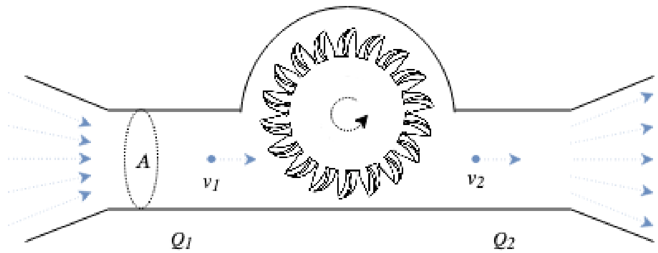

Fig. 1. Side view of Pelton-style turbine with equal sized input and output pipes.

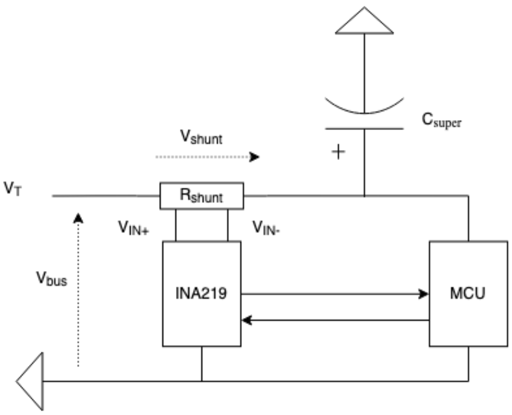

Fig. 2. Overview of the flow meter circuit, the power monitor chip takes two voltage signals, V_{in+} and V_{in-} and calculates the voltage drop over the shunt resistor and the voltage V_{bus} across the entire load. These two values are then used to internally calculate the current. The INA219 sends the measurements to the MCU, this is two-way communication since the power monitor is sent calibration values on startup.

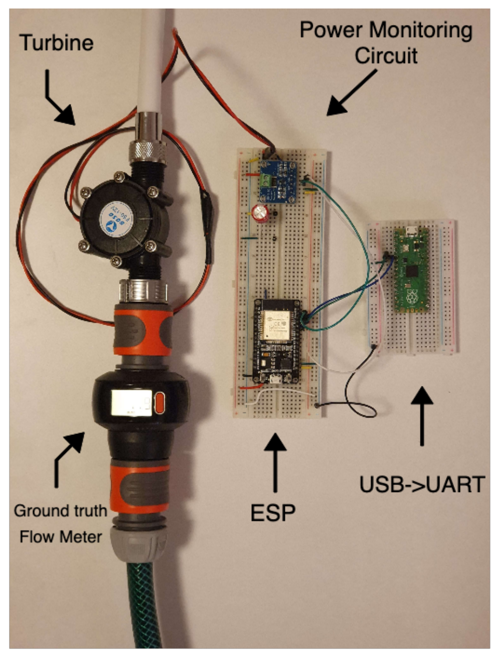

Fig. 3. Overview of the breadboard version of the flow meter used for testing.

and outlet are equal, so we also know that $v_1 = v_2$. Then by using Bernoulli's principle:

$$\begin{gathered} E_1 = E_2 \\ \frac{1}{2}v_1^2 + gz_1 + \frac{p_1}{\rho} - \Delta H - E_{loss} = \frac{1}{2}v_1^2 + gz_1 + \frac{p_2}{\rho} \end{gathered} \tag{2}$$

We can cancel out the velocity head elevation head at both sides since in this case, we assume that there is no difference in elevation of the in and outlet (this does not hold if the pipe is not in a horizontal setting). Then we are left with:

$$\begin{aligned} \frac{p_1}{\rho} - \Delta H - E_{loss} &= \frac{p_2}{\rho} \\ \Delta H &= \frac{p_1 - p_2}{\rho} - E_{loss} \end{aligned} \tag{3}$$

i.e. ΔH is equal to the pressure drop over the turbine. In a household water network, the incoming pressure would depend on mains water pressure, and in most cases p_2 would be atmospheric pressure — for example, when water exits through a faucet. The incoming pressure is dependent on the velocity of the incoming water, and thus also dependent on the flow rate.

Using Equation 1, we can see that there is relation between the flow rate and the power generated by the turbine. Note that the power produced by the turbine is used by the electrical system, and it could be that the load of the system is not constant. Including this and other factors, the relationship between P_H and Q is assumed to be non-linear. However, in practice (shown in section III), the relationship is approximately linear.

A. *Measuring incoming power*

To measure the incoming power, we need to be able to measure the induced voltage of the turbine, and the current passing through the system. In our implementation, we use a turbine which outputs DC power, so we consider methods for measuring DC power.

The power generated by the turbine is not always stable, so to help with this, a large capacitor is placed on the power line. Next to acting as a short-term battery, the capacitor also compensates for sudden voltage drops if the sensor load changes suddenly.

1) Measuring current: There is a multitude of ways to measure current [6], of which the two most used DC methods are: measuring current by using a Hall-effect-based current sensor, and measuring the voltage drop across a shunt resistor to calculate current. Hall-effect-based current sensors are generally more suitable for high current use-cases and generally cost — whereas shunt-based current sensors are more accurate for low current applications and are much simpler in use.

The flow meter sensor uses the shunt resistor based method since small turbines do not produce much power, so the current is also in the low range. A nice bonus of this type of sensor is that it already measures voltage, so an additional circuit for measuring voltage is not necessary. The exact sensor used is the INA219 [5], which is a power monitor that measures two voltage points using an accurate analog-to-digital converter (accuracy of max 0.5% over) and calculates the power, current, and voltage internally. The INA219 feeds the voltage signals to a programmable gain amplifier, which allows calibrating the sensor to different voltage and current ranges.

2) Measuring voltage: What we are interested in is the voltage across the turbine output and ground (i.e. the supply voltage), this can be read out using a common Analog-to-Digital Converter (ADC) on a microcontroller, but these ADCs are not always accurate and are limited in the range of the microcontroller's internal voltage (e.g. 0-3.3V, 0-5V).

But if the shunt resistor is placed on the input line, then the INA219 already measures the supply voltage. Furthermore, the INA219 can sense a larger voltage range (0 to 26V), which is more than sufficient for small power generators. Hence we use the INA219 for measuring both bus voltage (supply voltage) and current. **??** provides an overview of the flow meter setup.

B. Estimating flow rate from power

We now demonstrate the feasibility of estimating the flow rate using the discussed power monitor sensor. To start, we conducted experiments to acquire data on the relationship between the sensor measurements and ground truth flow rate. To know the ground truth we used the Gardena Aquacount [2], which can track the average flow rate and cumulative flow rate.

For the water turbine, we used the Goso F50 micro hydro generator, which is a Pelton-style 3-phase generator.

We measured 10-20 seconds of power data at different steady-state flow rates. Figure 4 shows a sample of the measurements captured at a flow rate of 7.5 liters per minute (lpm). The bus voltage, current, and power data are calculated and stored on registers on the INA219 chip, which are read out at 100 Hz by the MCU [3].

1) Signal Noise: Figure 5 shows a close-up of the same measurements. Right away, it can be seen that there is slight measurement noise in the signal. To smooth the signal, we compared different methods: A Gaussian filter; Wiener filter; and a low-pass filter. To validate which method would be most effective, we took measurements where the flow meter was powered by a lab bench power supply at a constant supply voltage.

In Figure 6, we can see how the different filters affect the distribution of the power measurements. In this case, we know that the incoming power is relatively constant, so we should choose a method that reduces the variance as much as possible. Looking at the distributions, the Wiener filter and low-pass filter produced the best results, but ultimately a low-pass filter with a cutoff frequency of 7 Hz was chosen since it is easier to implement in software.

2) Estimation: To estimate the flow rate we evaluated using the following 3 metrics: just the bus voltage, the current, or both (power).

Of household water fixtures, appliances and showers use the most water. The US energy bill states that shower heads have

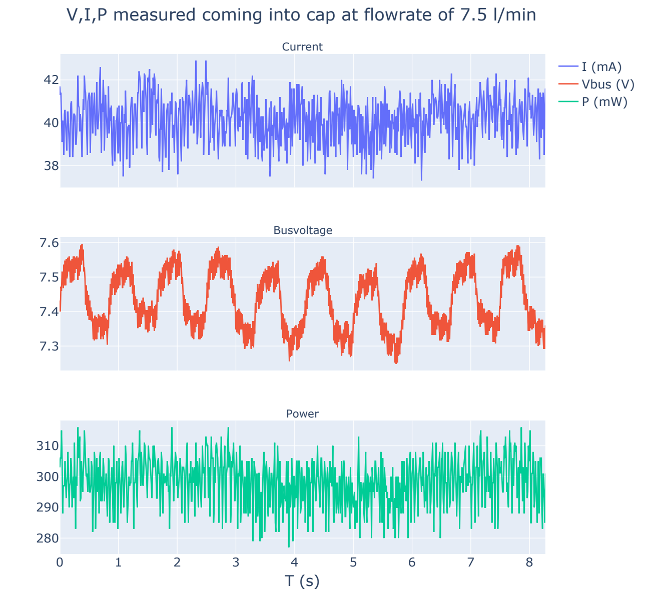

Fig. 4. Measurements from the INA219 of the generated power at a flow rate of 7.5 lpm.

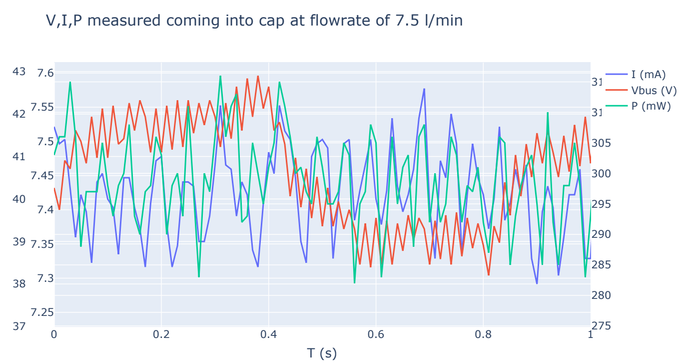

Fig. 5. 1s period of all power measurements. Zoomed in view of Figure 4.

a max flow rate of 2.5 gallons per minute (abbr. gpm, 2.5 gpm = 9.46 lpm) [3]. While in the EU, it is recommended to have a max of 9.0 lpm for showerheads [1]. Appliances generally use more, reaching upwards of 3 gpm (11.36 lpm). For the scope of this paper, we measured power data at flow rates between 7.5 and 9.7 lpm.

Figure 7 shows the average measured power value at flow rates in increments of 0.5 lpm. The mean power increases approximately linearly with the flow rate. The flow rate for household water fixtures can thus be roughly estimated using simple linear regression. The parameters for the regression line were calculated using non-linear least squares.

III. Experimental Results

We performed offline and online experiments using a linear estimator for the flow rate using the available power metrics.

1) Offline: A dataset consisting of samples of the power metrics (bus voltage, shunt voltage, current, and power) at steady-state flow rates are used. The dataset is split into train

[2]https://www.gardena.com/int/products/watering/aquacount/

[3]The reader may notice the constant fluctuation in the bus voltage at an approximate frequency of 1 Hz. This was caused by a blinking LED on the MCU.

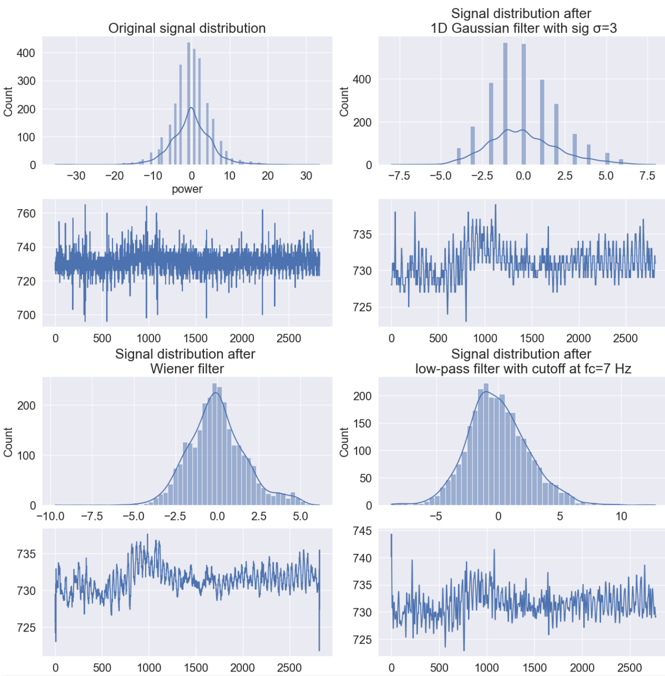

Fig. 6. This figure shows the 0-mean distribution and accompanying plot of the P measurements when using a lab bench as the power supply. From left to right, top-to-bottom we have: The original P signal, the P signal after a gaussian filter, P signal after a Wiener filter, and the P signal after a low-pass filter.

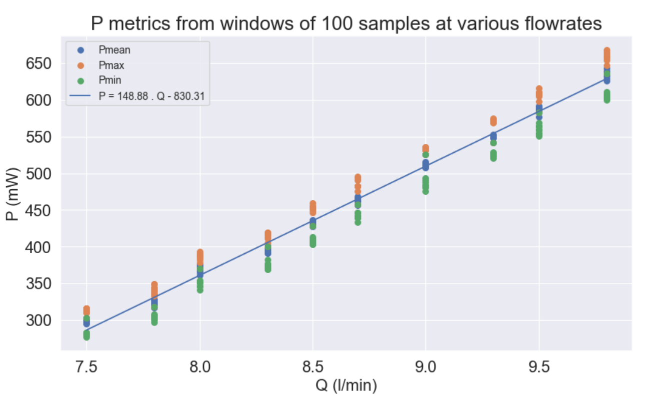

Fig. 7. This figure shows a linear regression line that calculates the generated power (P) from the flow rate (Q). The dots in the figure are the min, max and mean P values that are calculated from windows of 100 samples.

and test groups with an 8:2 ratio; each metric is then used to fit a regression line and validated using the Mean Squared Error (MSE).

Table I summarizes the error of the flow rate estimators compared to the ground truth for different window sizes.

Window Size	MSE($\hat{Q}_{Vshunt}$)	MSE($\hat{Q}_{Vbus}$)	MSE($\hat{Q}_I$)	MSE($\hat{Q}_P$)
10	0.006658	0.008867	0.007870	0.003822
20	0.006436	0.007195	0.004882	0.004081
50	0.003235	0.004738	0.003979	0.002394
100	0.001560	0.002821	0.001522	0.002465

TABLE I
MEAN SQUARED FLOW RATE ERROR OF DIFFERENT ESTIMATORS, USING THE TEST SAMPLES.

Based on the results, estimating the flow rate from the measured current is the most accurate.

2) Online: Next we performed tests using the simple flow rate estimation algorithm with the following setup:

- Estimation algorithm running on STMF4 board, powered by the F50 generator
- 1000uF capacitor for C_{super} (Figure 2)
- 0.1 Ω shunt resistor
- Sample rate of 100Hz
- INA219 calibrated to the 16V 400mA range

We are interested in the cumulative flow, which the Aquacount flow meter also measures. The cumulative flow was calculated on the STM32F4 and read out over serial.

Two types of tests were done, one where the flow rate was kept constant, and one where the flow rate was arbitrarily changed. For a period of 15 seconds, for the test with a constant flow rate, the measured water consumption was 0.02 liter less than the ground truth on average; whereas, for the test with a varying flow rate, the measured water consumption was over by 0.11 liter on average.

This shows that it is feasible to measure water consumption using a simple power monitor.

IV. IMPLEMENTATION

In this section we show the implementation of a prototype self-powered flow meter and the design decisions that were made. In section II the data was read out using a separate computer with UART, but as discussed in section I, a much more practical implementation would be to send data wirelessly. subsection IV-A discusses the considered methods for wireless data transfer. Additionally, the overall software architecture; how data is received; and the designed PCB are discussed.

A. Data transfer

To get an idea of which wireless protocol to use, first, some requirements are constraints are defined:

- The flow meter runs for very short bursts at a time, it should be able to quickly send the measured data within the timeframe of water running.
- Where the flow meter is placed is not always easily accessible.
- It can not be guaranteed that a person is always nearby.

All common protocols in IoT were considered. Ultimately, the choices were narrowed down to connectionless protocols, since it is not critical that all packets are received at the endpoint — and not using a connection reduces the time between boot and initially sending data.

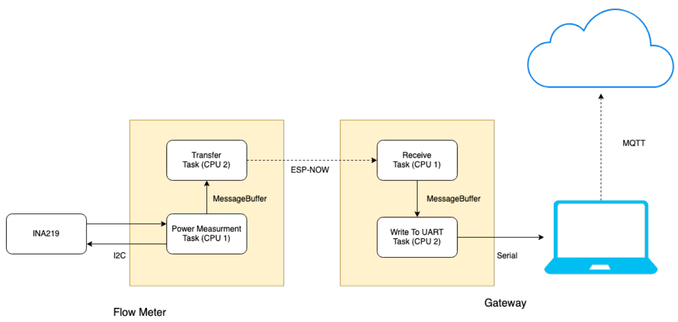

Fig. 8. Overview of each component of the software system. Each arrow represents sending data, dotted arrows show wireless communication.

Using something such as Wi-Fi would be impractical, since e.g. scanning and connecting takes some time, especially if the device must be assigned a dynamic IP, it could be that the user turns off the water source even before a connection is made. Ideally, we want a sort of "fire and forget" type of protocol.

The protocol chosen for the flow meter is ESP-NOW [4], a connectionless protocol on the low-power 2.4 GHz band. This protocol is tied to Espressif's ESP microcontrollers, so an ESP MCU is also used in the final flow meter.

B. Software Architecture

The firmware for the flowmeter is built around FreeRTOS [5] and Espressif's IoT Development Framework [6].

The overview of the architecture is given is Figure 8. The chip used on the flow meter is the ESP32, which is dual-core. The work can thus be divided; the first core is dedicated to the RTOS scheduler and also handles reading and processing data from the INA219 chip, transmitting the data is handled by the second core. The final flow data is pushed to the transmit core using a ring buffer.

The data is then received on a second ESP32, which is the gateway.

C. Acquisition Gateway

The idea behind the gateway is that it is a central receiver for all flow meters within a household. In the prototype, this is a second ESP32 that passes the received ESP-NOW packets to a different computer. This computer then transfers this data to a server using MQTT [7].

The data can then be viewed in an analytics dashboard. For the prototype, a simple web dashboard was used (Figure 9).

A much simpler solution would be that the ESP32 that receives the data also does the work of the external computer (sending data to the server). This was the original design, but there are some issues with this configuration. That is, the channel used for ESP-NOW should be the same as the channel

[4]https://www.espressif.com/en/products/software/esp-now/overview
[5]https://www.freertos.org/
[6]https://github.com/espressif/esp-idf
[7]https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

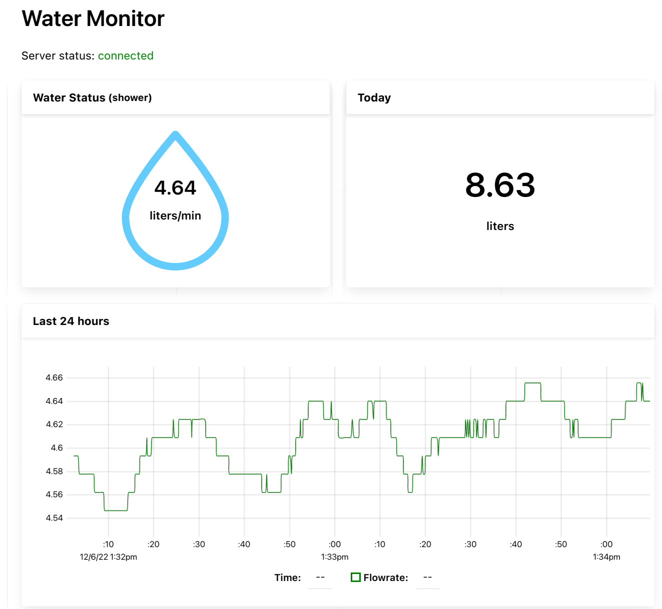

Fig. 9. Analytics dashboard that shows data received from the flow meter. Top left is the current flow rate of 1 flow meter, top right shows the total water used in the day, and the bottom graph shows the flow rates from the last 24 hours.

of the Wi-Fi Access Point (AP) that the gateway is connected to.

And it could be that the AP decides to change channels, which both the flow meter and gateway must account for ESP-NOW. In tests, ESP-NOW packet loss was common when the gateway ESP has to simultaneously receive the flow meter packets and transmit them over Wi-Fi due to channel switching. A way to circumvent this would be for the flow meter to sense which channel is used by the AP the gateway is connected to, but this defeats the "fire and forget" idea. So for simplicity, two devices were used for the gateway, which was also much more robust in testing.

D. PCB

The final flow meter prototype (see Figure 10) works by connecting the DC output from the turbine on the side via a JST connector. This is then fed through the power monitoring circuit and then to a voltage regulator, which powers the onboard ESP32.

V. Discussion

Some issues were encountered with the ESP-based prototype, which is discussed below. Future work and improvements on the design are also discussed.

A. Brownout Detection on the ESP32

The ESP32 boards have a built-in hardware brownout detector, which is triggered when the supply voltage drops below a certain threshold. The chip then resets after. This should not be an issue, however, if the supply voltage is below the brownout

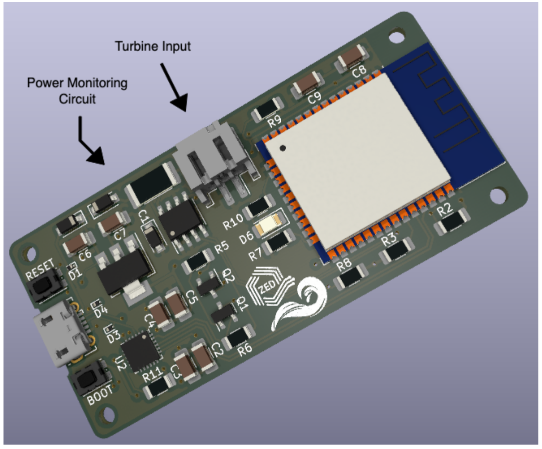

Fig. 10. 3D render of the designed PCB.

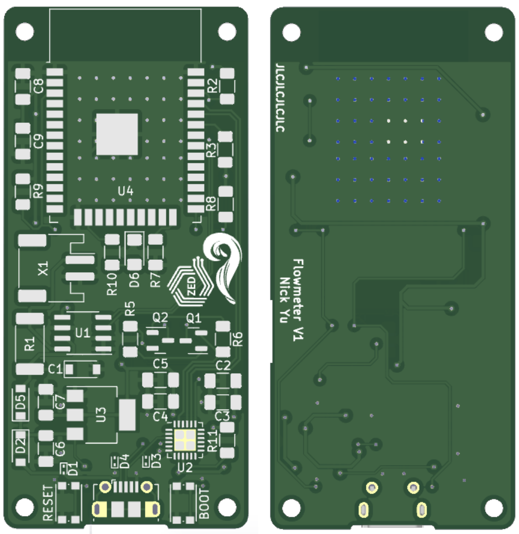

Fig. 11. Front and back view of the bare PCB.

threshold on initial boot, the ESP32 hangs indefinitely and must be physically reset to function again. i.e. when powered via the turbine, the voltage ramps up from 0 when turning on the water source, causing the ESP32 to hang on every boot. Whereas, the STM32F4 did not exhibit this behavior and would just restart continuously until a suitable voltage level is reached.

The brownout detector can be switched off in software, and the device would not brownout when already booted up, but in testing, this change did not fix the hang on the initial boot. A hardware mitigation would be to only supply power to the ESP above a certain threshold (e.g. via a MOSFET and voltage divider). This was not looked into, but can be considered for the future.

B. Power Consumption

Another issue with the ESP32 is power consumption, the ESP32 draws a base current of 40mA at 7.5V ($\approx$0.3W). Which the Goso F50 turbine could provide with no problem. But when the Wi-Fi PHY is used, the ESP draws $\approx$0.75W of power, at flow rates above 8 lpm, the turbine generates enough power, but below that the ESP32 would switch off.

Even though the datasheet of the turbine states that it should be able to generate 10W at max, in reality, the turbine would generate less than 1W at the max flow rate.

Some measures were attempted to lower the power consumption or fix this issue, such as:

- Decreasing the Wi-Fi antenna TX power.
- Using a different protocol, such as BLE.
- Only starting the Wi-Fi PHY after a delay; The idea is that the large capacitor in the circuit has time to charge up and provide power for the PHY chip.
- Disabling RF calibration, a step done at the initialization of the PHY, which draws a lot of power.

An idea not tried could be to have the ESP board run without the PHY turned on, and only turn it on for a short time to transmit a single packet.

All in all, the ESP32 board was not entirely suitable for this low-power use case.

C. Future Work

The flow rate estimation algorithm used is naive and just assumes that the relationship is linear and must be exactly calibrated for the turbine used. Work could be done to measure power generation at a larger range of flow rates and validate the relationship better. Other ways to measure flow rate could also be looked into, such as using the RPM of the turbine, and checking how this method compares to using the supply power as a metric. Another improvement is to design a more power-efficient implementation than the one presented in this paper.

VI. Conclusion

We have shown the design and implementation of a device that is powered by a water turbine and simultaneously uses the input power to estimate the amount of water flowing through the turbine. This design is plug-and-play and only requires installing the flow meter between the water fixture, and its existing water source. After which, the flow meter automatically sends data on how much water is used to a base station. To see if it is feasible to determine water consumption using power measurements, we have also performed experiments and validated different metrics to estimate water flow rate and water consumption. Finally, we designed a PCB version of the breadboard prototype.

VII. Acknowledgements

This paper was written as part of the Advanced Practical I.o.T and Seminar course given at the Delft University of Technology. The authors would like to thank the course staff for the wonderful course. The authors would also like to thank Niels Hokke for his guidance and support throughout this project.

REFERENCES

[1] EU GPP Criteria for Sanitary Tapware, December 2022. [Online; accessed 2. Dec. 2022].

[2] Water Scarcity | Threats | WWF, December 2022. [Online; accessed 2. Dec. 2022].

[3] 102nd US Congress. Energy policy act of 1992, October 1992. [Online; accessed 2. Dec. 2022].

[4] Washington State department and Health. Stop Water Waste, 2020.

[5] Texas Instruments. *Zerø-Drift, Bidirectional Current/Power Monitor With I2C Interface*, 12 2015. Rev. G.

[6] Chucheng Xiao, Lingyin Zhao, T. Asada, W. G. Odendaal, and J. D. van Wyk. An overview of integratable current sensor technologies. In *38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003.*, volume 2, pages 1251–1258vol.2. IEEE, October 2003.