University of North Carolina at Asheville
Journal of Undergraduate Research
Asheville, North Carolina
May 2020

Research and Development of a Low-Cost Device to Track Natural Gas Consumption of University Facilities

Fredi R. Mino
Mechatronics Engineering Program
The University of North Carolina Asheville
One University Heights
Asheville, North Carolina 28804 USA

Faculty Advisor: Dr. Don Nicholson and Dr. James Perkins

Abstract

Natural gas consumption data can be used by facility managers to implement procedures that make more efficient use of a facility's heating, ventilation, and air conditioning systems (HVAC). Gas consumption data is measured by an external company, and currently, they can only provide monthly consumption data. In order to implement high-performance controllers, campus facilities management requires data to be provided on approximately 15-minute intervals. Additionally, the meters need an upgrade of approximately $88,000 to provide data at that rate. This paper proposes an open-source design for a device that can capture the consumption data at the desired rate for less than $200.

1. Introduction

This project is part of the Undergraduate Research in Climate Action initiative at UNC Asheville. Since time-dependent information on gas and electric usage can provide a detailed picture of a building's HVAC performance when correlated with ambient temperature and building occupation, UNC Asheville's Campus Facilities management would like to obtain these data at small intervals and in real-time. However, currently, only monthly total usage is provided from the gas meters on campus by the company that owns them.

In order to provide data at a rate of 15-minute intervals, the company in charge of the meters determined that an upgrade of approximately $2000 per meter is required. There are approximately 44 gas-meters on campus. Therefore, the cost of upgrading the meters to give this short-term usage is about $2000 per meter, or $88,000 total. The goal of this project is to design and implement a device that can provide gas consumption data at the specified rate while representing only a fraction of the cost proposed by the external company. Offering a cheaper alternative might help in persuading the aforementioned company into providing more competitive prices. Additionally, although the scope of this project is limited to UNC Asheville, it could be used in other campuses and residential areas.

2. Requirements and Constraints

The following list of requirements was proposed by the clients:

- The device must be able to track and store gas consumption data at least once every 15 minutes.
- The device will be located outdoors, and thus it must endure environmental conditions.
- The production costs of the final design should be minimized as much as possible.

The constraints, which arise from the configuration of the environment in which the device must perform and other external factors, are as follows:

- In general, there are no power outlets near the meters. Therefore, the device will have to use batteries and incorporate a that allows recharging them in situ.
- The company that owns the meters and currently provides the data will not allow modifications or attachments of any kind to the gas meters. In other words, the device cannot be in physical contact nor directly interact with the meters.
- The approved prototyping budget for the device is $170.00.

3. Top-Level System Design

Based on the requirements and constraints. The system shown in Figure 1 was proposed, and four main subsystems were identified and are described in Figure 2.

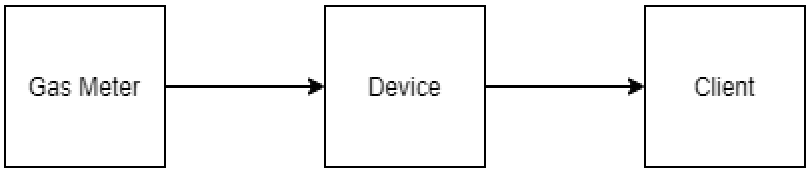

Figure 1. Top-level block diagram of the system

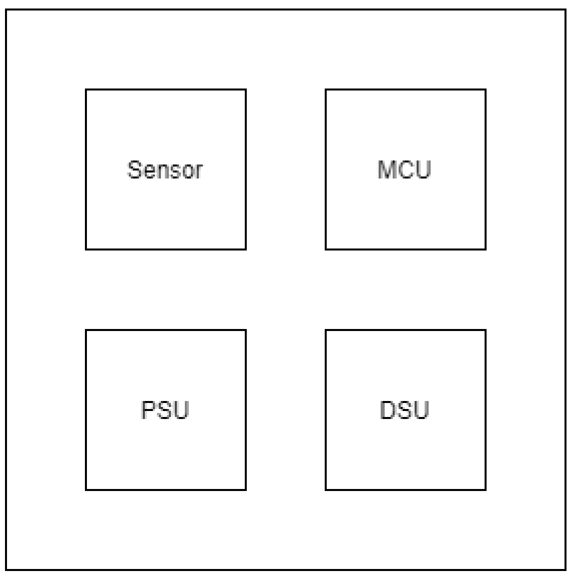

Figure 2. Top-level block diagram of the device.

Figure 2 shows the main subsystems of the device. The sensor or array of sensors extracts gas usage information form the meter without physically interacting with it. The micro-controller unit (MCU) processes the information provided by the sensors and manages its storage and transmission to the client. The power supply unit (PSU) maintains the device constantly powered. The data storage unit (DSU) contains the processed data related to gas consumption until it is acquired by the client.

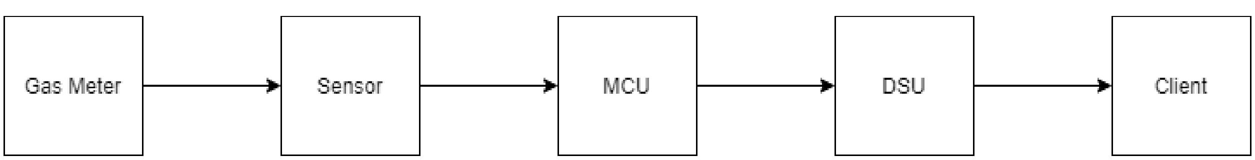

Figure 3. Data-flow diagram of the system

4. Hardware

4.1. Sensor

The device itself did not need to be too accurate, but it needed to be reliable. Since it would be exposed to environmental agents and other disturbance sources, it needed to be robust to support changes in the environment. Computer vision provided this characteristic. It allowed flexibility in the placement of the device. The sensor choice was based on the ease of integration with the microcontroller, and the ability to take pictures of sufficient quality. The Raspberry Pi Camera Module V2 has an 8-megapixel sensor that works with all Raspberry Pi models.

4.2. MCU

Since computer vision was chosen as the data acquisition and computation method, the micro-controller had to be able to handle the processing algorithm. The Raspberry Pi is different from other boards like the Arduino in that it can handle more computational complexity at the expense of power consumption.

For the initial prototype, the Raspberry Pi 4 model B was chosen. It included 4GB of RAM to guarantee that the algorithm could run. The final version of the device would include a board that optimizes resources such as the industrial version of the board (Compute Module 3+) which is cheaper and more power-efficient.

4.3. PSU

Since the power supply was not one of the main requirements of the project, a previously implemented power module would be used as a placeholder until the rest of the design had been completed. In other words, there were no considerations for optimization or constraints. The module in question was a 2-cell 7.4V Lithium-Polymer battery with a capacity of 3000 m-Ah connected to a Battery Eliminator Circuit (BEC) that outputs 5V and up to 3A of current, which meets the power requirements of the microcontroller board. However, LiPo batteries are known for being unstable when not treated with caution, which means they are not suitable for continued outdoor use.

4.4. DSU

The Raspberry Pi uses a Linux distribution that runs the program and manages the storage of information. There are two sources of information: 1) the pictures taken by the camera module, and 2) a log file continuously updated with the output of the image processing algorithm. The information is stored in a 32GB micro-SD card that serves as the hard drive for the Raspberry Pi.

4.5. Client-Device Interface

Currently, the only way to access the data log is to physically remove the micro-SD card from the board. However, the intention is to eventually add an IoT interface to allow data transmission over a wireless connection.

5. Bill of Materials

The total cost of the components for the first iteration of the prototype is itemized in Table 1. It is important to note that these components are the latest versions and therefore are more expensive than older versions that might be able to provide the same functionality. Design optimizations were neglected in favor of speeding up the prototyping process. Eventually, after achieving the proof of concept, components will be chosen based on cost optimization.

Table 1. Bill of Materials of the First Iteration of the Prototype

Item	Quantity	Cost
Raspberry Pi model 4B (4GB)	1	$55.00
Raspberry Pi Camera Module V2	1	$29.95
USB-C Power Supply (5.1V-3.0A)	1	$7.95
Turnigy 3000 mA-h 2S 40C LiPo	1	$17.64
	Total	$110.54

6. Software

6.1. Dial Recognition and Data Acquisition

The computer vision algorithm was implemented using source code provided by an Intel tutorial[1], which was modified in compliance with the MIT License for open software distribution. The software uses OpenCV library functions to detect lines and circles, and then infer measurements based on the angle of the line with respect to the vertical axis and the range of values that can be measured by the meter (0-9).

Figure 4. Residential gas meter located outside Willow Hall.

Figure 4. The meter has a total of 5 black dials that measure natural gas flow in units of cubic feet. Each dial represents a place value ranging from 1,000 to 10,000,000.

Figure 5. Dial from the gas meter in figure 4.

Figure 5 A manually zoomed-in section of the gas meter was used as the input to test the computer vision algorithm. The dial moves clockwise and increases only with the gas flow. The precision of the measurements can be improved by zooming-in on the appropriate dial.

Figure 6. Output image of the gauge detection algorithm.

Figure 6 With figure 5 as the input, the program identifies the largest circle it can find within the image and labels it as the gauge. Then, it proceeds to find its center, which is used as a reference for the line identification function. Note that the reference for the angular displacement starts at 5 (0 degrees) and increases in a clockwise manner.

Figure 7. Filtered image version of figure 5.

Figure 7. In order to improve the performance of the line detection function, the input image is transformed into a greyscale color space. Then, the program looks for pixel values above a certain threshold to make a binary representation of the image that it can use to identify lines.

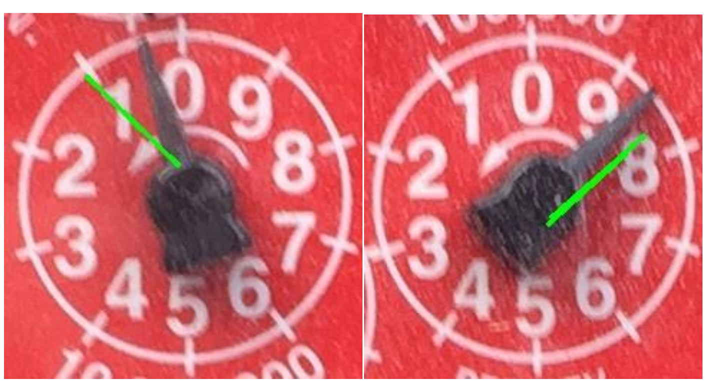

Figure 8. Line detection output for different inputs

Figure 8. The algorithm creates a list of all lines that fall within an acceptable range of the radius of the gauge. Then, it chooses the first acceptable line as the dial. Currently, the algorithm does not correctly align with the dial, but it falls within a range that the client considers acceptable. By running the program every 15 minutes, a log file can be created to save datapoint.

7. Conclusions

At the time of submission of this paper, the design has passed the proof of concept. However, the implementation and testing stages are still underway. Field data is needed to determine the precision and reliability of the data collection processes in addition to stress testing. The data gathered during these tests will be used to optimize power consumption, production costs, and protection from environmental factors.

8. Future Work

There are several ways to improve the functionality of the device, and although they fall outside of the scope for this project, they are worth considering and could be present in future iterations.

8.1. Wireless User Interface

Currently, the client has to remove the micro-SD card in order to be able to access the data logs. A more convenient approach would be to provide a wireless interface using an Internet of Things (IoT) module that can act as a server.

8.2. Solar Powered Batteries

Although the batteries used in this project are rechargeable, they are not suitable for long-term exposure to environmental conditions, and they must be removed in order to be charged. It might be worth looking into using solar cells and more robust energy storage devices that can operate in harsher conditions.

8.3. Power-Efficient Board

The current board was chosen due to its ease of use for prototyping, however, there are cheaper, more professional, and more efficient boards in the market like the MSP430 or the Raspberry Pi Compute Module. These boards should be considered to minimize the costs of production and minimize power consumption.

8.4. Night Vision

The meter used for the proof of concept happens to be illuminated 24/7 thanks to a lamp post nearby. However, if the device is to be of use in a wide range of environments, it will need a way to take clear pictures in the dark. One way to achieve this without adding more components would be to use a camera module without an infrared filter.

9. References

1. B., Joseph. "Analog Gauge Reader Using OpenCV in Python." Intel Software Development Zone. Intel Corporation. Published on November 28, 2017. https://software.intel.com/en-us/articles/analog-gauge-reader-using-opencv