

Continuous Stirred-Tank Reactor

**Team Members**

Aarsh Desai

Jessica Martínez Hernández

Jorge Castro

Sarah Lovely Marcelin

**Gas Counter Report**

Professor Himanshu

**Table of Contents**

1. Introduction………………………………………………………………………………..
   1. Purpose………………………………………………………………………
   2. Goals…………………………………………………………………………
2. Hardware Specification…………………………………………………………………...
   1. Arduino……………………………………………………………………...

2.1.1- Specifications………………………………………………………….

2.1.2- Limitations……………………………………………………………..

* 1. Raspberry pi………………………………………………………………….

2.2.1-  Specifications………………………………………………………….

2.2.2-  Capacities……………………………………………………………..

2.3. Gas Counter………………………………………………………………….

2.3.1-  Components…………………………………………………………...

2.3.1.1-  Reed switch…………………………………………………..

2.3.1.2-  Bucket tip mechanism………………………………………..

2.4.  Complete circuitry…………………………………………………………....

2.4.1-  Components…………………………………………………………..

1. Software Specification……………………………………………………………………...

3.1- Algorithm…………………………………………………………………………...

3.1.1-  Communication between Arduino and Python…..…………………………..

3.1.2-  Writing tips simultaneously to 30 csv files…………………………………..

3.1.3-  Matplotlib’s pandas library………………………………………………...

3.2- UML diagram………………………………………………………………………..

3.3- Function blocks description………………………………………………………….

1. Tests and Results…………………………………………………………………………....

4.1- Detecting tips………..………………………………………………………............

4.2-    Reading and Writing to csv files…………………………………………………….

4.2- Data points plots……………………………………………………………………..

1. Discussion…………………………………………………………………………………..

5.1- Observations………………………………………………………………………....

5.2- Difficulties…………………………………………………………………………...

5.3- Future plausible applications………………………………………………………...

1. Conclusion…………………………………………………………………………………
2. References………………………………………………………………………………...

**List of Figures**

1. Figure 1: write name……………………………………………………………………
2. Figure 2: write name……………………………………………………………………
3. Figure 3: write name……………………………………………………………………
4. Figure 4: write name…………………………………………………………………….
5. Figure 5: write name……………………………………………………………………..
6. Figure 6: write name……………………………………………………………………..
7. Figure 7: write name……………………………………………………………………..
8. Figure 8: write name……………………………………………………………………..
9. Figure 9: write name……………………………………………………………………..
10. Figure 10: write name…………………………………………………………………….
11. Etc………………………………………………………………………………………….
12. Introduction
13. Purpose and Goals

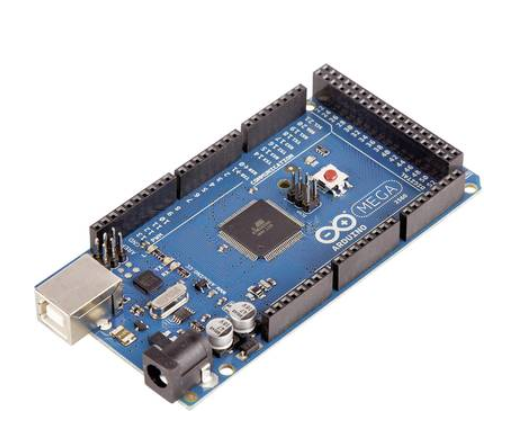
In engineering, data visualization is one of the most used techniques to communicate data information through graph interpretations. These interpretations can be of the forms of points, lines or bar graphs. One of the main purpose of this case study was to be able to visualize the gas flow of the gas counter modules through line graphs; more precisely embedded in a graphical user interface which would allow the user to select the counter of his/her choice and also visualize the flow rate for the last 24 hrs., last 7 days and last 31 days.

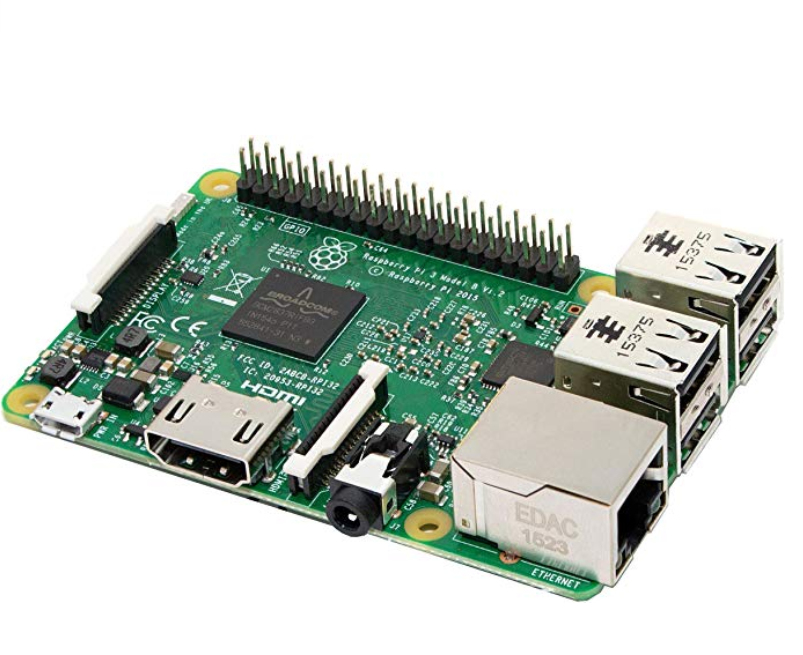
Completing this case study module depends on the achievements of the following goals:

1. Connecting to the Arduino Mega board using python libraries
2. Simultaneously and continuously reading the output values of 30 digital pins
3. Detecting the electrical pulse of the reed switch by implementing the concept of debouncing
4. Once a ‘tip’ is detected, write the current time and volume to the corresponding csv file. A maximum of 30 counters is expected, therefore 30 csv files will be created and written to.
5. The name of the csv files must be written based on a predefined format, YYYYMMDD – GCX; where the timestamp is the time the file was created following with the abbreviation of the corresponding gas counter ( X ranges from 1 to 30).
6. A new file must be created for each counter if the timestamp goes back to 30 days ago and if the file size if greater than 10 MB.
7. The graphical user interface should be easy to manipulate and understand. It should primarily allow the user to select and display the data for any of the 30 counters.
8. Using python’s data analysis tools, the flow rate of each counter should be displayed as chart and should be selectable between month, hours, days.
9. The overall program should be able to run successfully on a raspberry pi’s operating system.

Each of these goals are explained thoroughly in the Software Specification section (Please refer to section 4)

1. Hardware Specifications







3) Software Specifications

3.1- Algorithm

Connecting to the Arduino Mega board using python libraries

We were told that the software needed to be designed for the Raspberry pi

instead of relying solely on the Arduino. Using python's library we were able

to successfully reach this requirement. Python's Nanpy is a library that use

the Arduino as a slave device, controlled by a master device such as a Raspberry Pi.

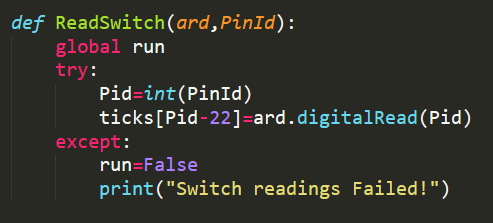
The Following function connects to the Arduino using Nanpy's ArduinoAPI and serial manager. The function’s parameter 'com' is the com port selected by the user for example "/dev/ttyACM1" or "/dev/ttyUSB1". Once connected to the Arduino board, an Arduino object "ard" is passed

to the ArdSetup function which sets the pins mode to output and sets the pins

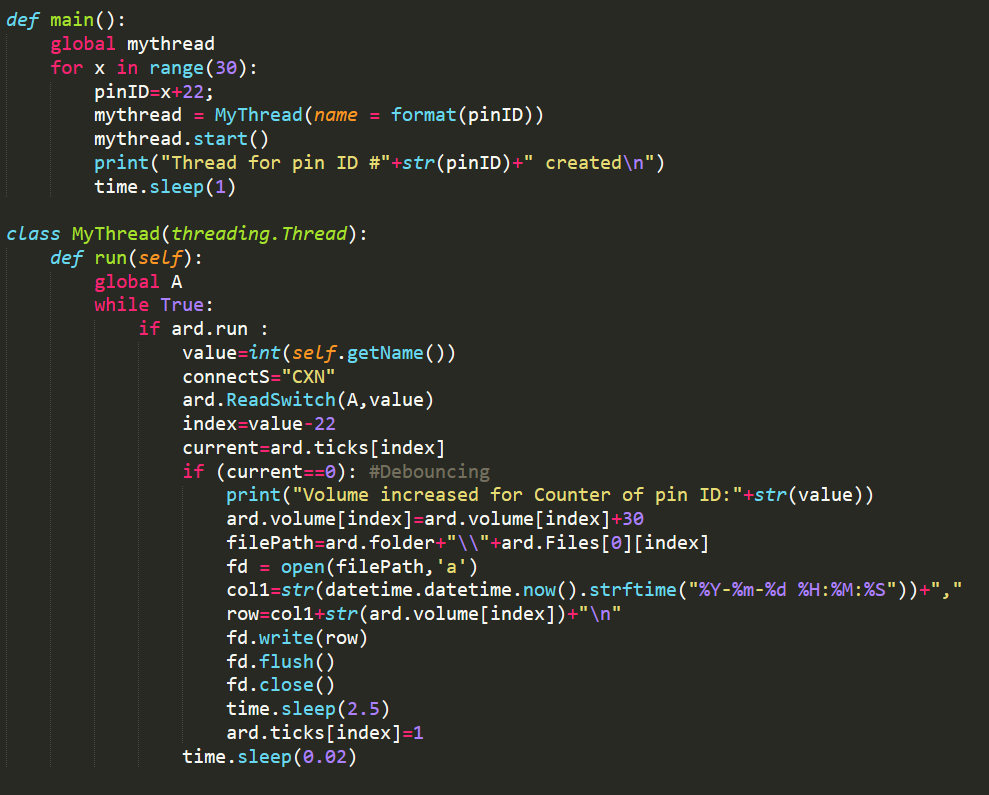
to high or a digital value of 1.



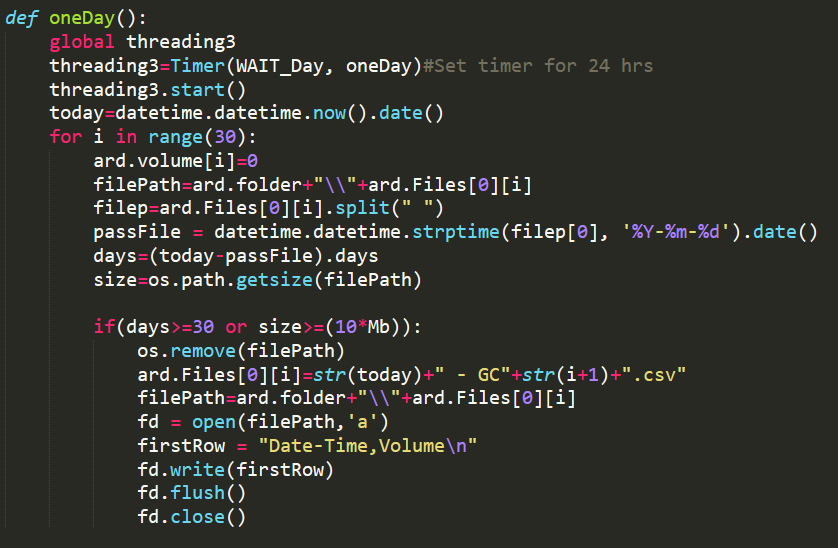
1. Simultaneously and continuously reading the output values of 30 digital pins



1. Detect the electrical pulse of the reed switch by implementing the concept of debouncing. Once a ‘tip’ is detected, write the current time and volume to the corresponding csv file.



1. The name of the csv files must be written based on a predefined format. A new file must be created for each counter if the timestamp goes back to 30 days ago and if the file size if greater than 10 MB.



1. Using python’s data analysis tools, the flow rate of each counter should be displayed as chart and should be selectable between month, hours, days.

