DEVELOPMENT OF SPIROMETER

Contents

1 Introduction **2**
1.1 Spirometer Fundamental 2
1.1.1 Spirometry 2
1.1.2 Common spirometry parameters 3
1.1.3 Flows and volumes measurement 5

2 Hardware Description **7**
2.1 Spirometer block diagram 7
2.2 Respiration set 7
2.3 Pressure sensor 9
2.4 Microcontroller 9

3 Software Description **13**
3.1 Converting between digital and analog signals 13
3.2 Converting from a voltage to pressure 13
3.3 Calculation of the volumetric flow 16

4 Sampling times and process **17**

5 Calculation of the parameters **19**

6 Future Project **27**
6.0.1 Texas Instruments BLE software development platform 27
6.0.2 Bluetooth Technology 27
6.0.3 Programming CC2541 28
6.0.4 ADC of CC2541 28

Chapter 1

Introduction

The contents of this application note show how it is possible to use the Arduino families of microcontrollers along with the Differential pressure sensors to implement a device capable to quantify human respiration capacities, by measuring volumes and flow rates.
The results of the test are sent afterwards to a host computer installed with a graphical user interface (GUI) that shows a graph of the respiration process and the measured results. The data is sent using the USB stack.

1.1 Spirometer Fundamental

This section provides a general explanation of the spirometry parameters, and the advantages of being able to measure such parameters. It also gives a general explanation how the spirometer takes flow and volume measurements.

1.1.1 Spirometry

Spirometry refers to a series of simple tests of a persons respiratory capacities. Spirometry takes measurements of the quantity of air inhaled and exhaled by the lungs during a certain period of time to determinate the pulmonary capacity. The device used with this purpose is called a spirometer. These measurements are useful when it comes to checking pulmonary function, since results are valuable in diagnosing diseases such as pulmonary fibrosis, asthma, cystic fibrosis and COPD (chronic obstructive pulmonary disease). When the spirometer is a pneumotachograph type, the process to perform a spirometry test usually involves the patient breathing through a hose or tube

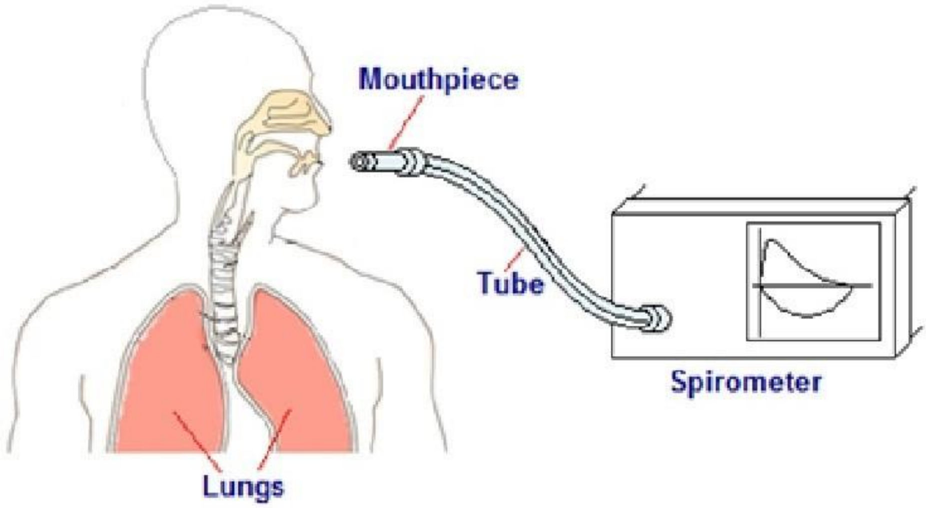

Figure 1.1: Basic Spirometer System

in which one of the ends contains a sensor that quantifies the air flow. The basic components of a spirometer are the mouthpiece, a hose or tube, and an electronic device to measure flows and calculate spirometry parameters. The process to perform a test involves the patient to breathe using the mouthpiece to generate an air flow through the tube, which allows its conversion, by a sensor to an electrical signal.

1.1.2 Common spirometry parameters

The most usual and important spirometry parameters are mentioned and described below:

1. Tidal Volume (TV) - The amount of air imhaled or exhaled during a single breath without forced conditions.

2. Inspiratory Reserve Volume (IRV) - The maximum additional air that can be inhaled at the end of a normal inspiration.

3. Expiratory Reserve Volume (ERV) - Refers to the maximum volume of air that can be exhaled at the end of a normal expiration.

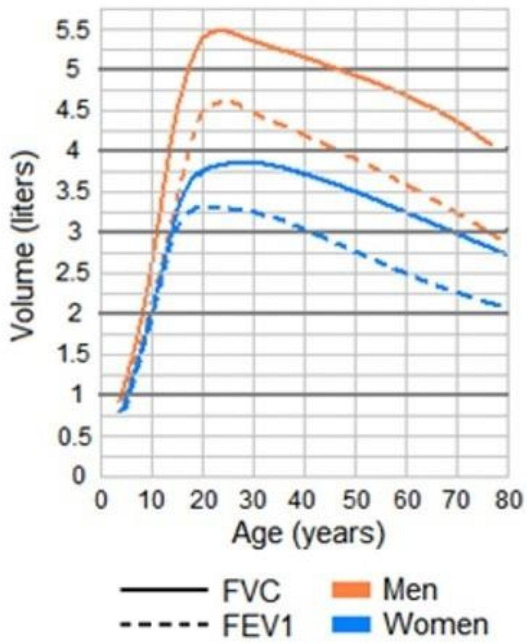

Figure 2. Average values for FVC and FEV1

4. Vital Capacity (VC) - The maximum amount of air that can be expelled from a persons lungs after a maximum inspiration. The vital capacity is equal to the sum of IRV, ERV, and TV.

5. Forced Vital Capacity (FVC) - The volume of air that can be blown out by a person at a maximal speed and effort after a full inspiration.

6. Forced Expiratory Volume in 1 second (FEV1) - Represents the maximum volume of air that can be exhaled in a forced way in the first second, after taking a deep breath.

7. Forced Inspiratory Vital Capacity (FIVC) - The maximum air volume that can be inhaled.

8. Peak Inspiratory Flow (PIF) - The forced maximum flow that can be achieved during inhalation.

9. Peak Expiratory Flow (PEF) - The maximum air flow that can be forced during exhalation.

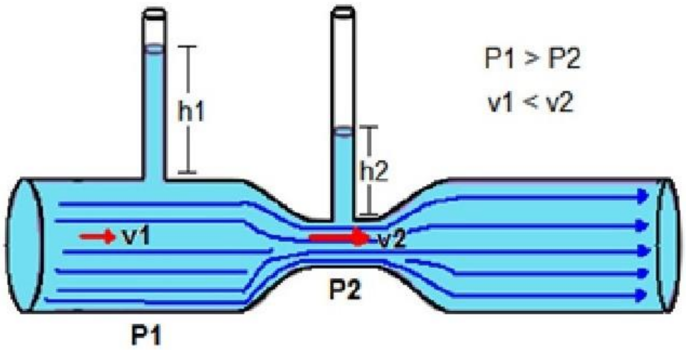

Figure 1.2: Venturi Tube

1.1.3 Flows and volumes measurement

The method to measure flow in this application is based on the fluid dynamics laws. One of these laws is the Venturi effect according to which when a fluid passes from a wider to a narrower section of a pipe, the pressure of the fluid reduces while the velocity increases.

The velocity and pressure of the fluid change to satisfy mass conservation is regulated by the next equation, called the Venturi effect equation:

$$P_1 + P_2 = \frac{d}{2}(v_2^2 - v_1^2)$$

The flow rate in the sensor can be determined if the differential pressure between the sections of the different diameters is known.The flow sensor consists of ports A and B which measures total or static pressure, depending on the direction of the flow. The difference between the two of them represents the dynamic pressure, which is proportional to the velocity of the fluid passing through the tube.

$$dP = \left(\frac{W^2}{2\rho}\right) \times \left(\frac{1}{A_2^2} - \frac{1}{A_1^2}\right)$$

The mass of the air so obtained is

$$massFlow = 1000\sqrt{\frac{2\rho(pressure)}{\frac{1}{area_2^2} - \frac{1}{area_1^2}}}$$

The volume of air that flows through the cross-section in 1 sec

$$volFlow = \frac{massFlow}{\rho}$$

Chapter 2

Hardware Description

This chapter presents an explanation about the hardware needed to implement a spirometer.

2.1 Spirometer block diagram

The blocks shown in Figure, describes the basic layout of the spirometer.

2.2 Respiration set

The first step to implement a spirometer is the acquisition of the signal, in this case coming from the breathing process, so there is a way to detect the air volumes and flow rates involved in the process. This can be achieved using a special mouthpiece that consists of a respiration tube and a hose with two air ways to transmit the air pressure to a pressure sensor that handles the conversion of the signal. During a test, the patient breathes using the tube, which contains a flow restriction mechanism that enables the air to flow just through one of the air ways in the hose, depending on whether the person is inhaling or exhaling. The mouthpiece can be observed in Figure.

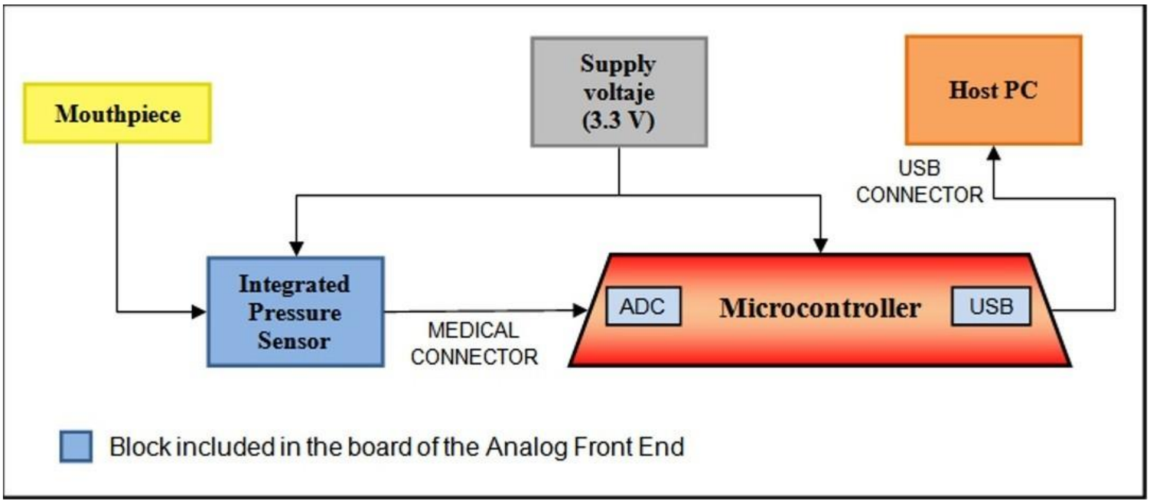

Figure 2.1: Spirometer System Block Diagram

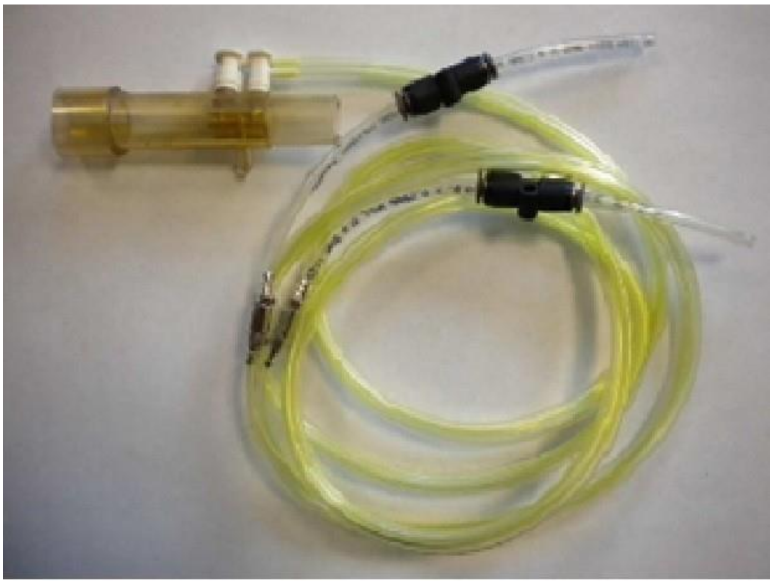

Figure 2.2: Mouthpiece

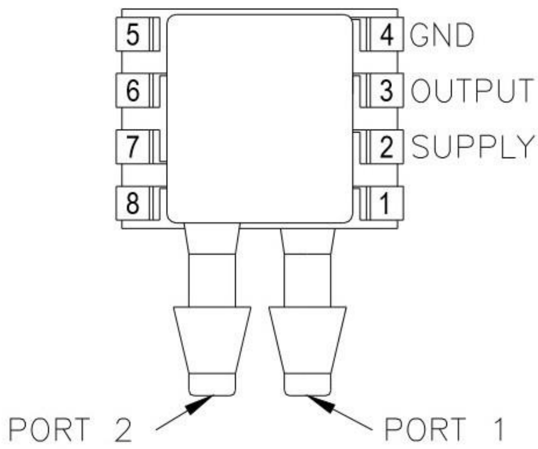

Figure 2.3: MS4525 Differential Pressure Sensor

2.3 Pressure sensor

A key part when implementing a spirometer is a transducer to convert the air flow into an electrical signal, which can be then processed by electronic means. To achieve this, the system uses an integrated silicon pressure sensor, the MS4525.

It is a differential pressure sensor that delivers an analog output voltage proportional to the applied differential pressure on the sensor.
The key features of this pressure sensor are listed below:

- Pressure range: -34 to 34 kPa (-5 to 5 psi).
- Output: 0.0 to 3.3 V.
- Ideally suited for microprocessor or microcontroller systems.
- Temperature compensated over -40 to +125C.
- 3.3 or 5.0 V dc Supply Voltage.

2.4 Microcontroller

The microcontroller is a fundamental piece of the design. It performs the principal functions of the system, from sampling and converting the signal delivered by the pressure sensor to calculating the spirometry parameters

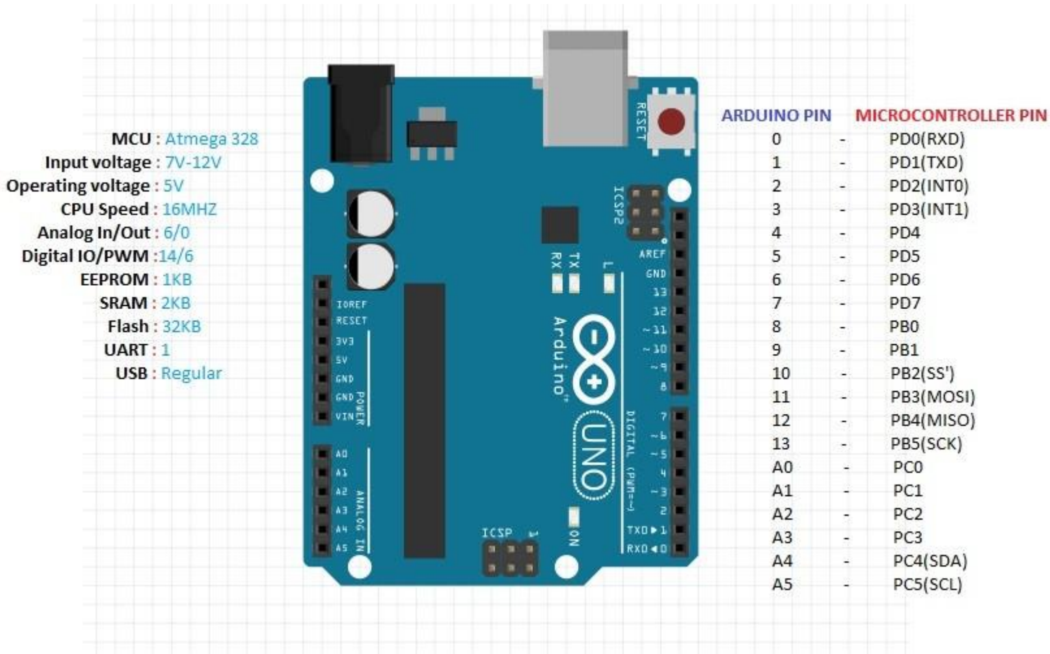

Figure 2.4: Arduino Uno board

and sending data and results to the GUI.
Here Arduino Uno is being used as the microcontroller and has the following features and peripherals in its integrated measurement engine.

- Ultra low-power consumption
- 10-bit analog-to-digital converter(ADC)
- 14 digital pins out of which 6 could be used as PWM pins.
- Inter-integrated circuit (I2C)
- Universal Serial Bus connectivity (USB)

The activities assigned to the microcontroller module are the following:

1. Take samples and convert the analog voltages from the pressure sensor into digital values using the internal ADC.
2. Receive requests from the Host PC via USB communication to start and stop measurements; it also sends both confirmation and data.

3. Quantify and detect the current state of the breathing process, to check if the patient is inhaling or exhaling and the limits in which the measurement must be paused or stopped.
4. Perform spirometry calculations and send volume and flow data to the GUI so it can plot them on the screen of the PC.
5. Send the final results to be shown on the GUI.

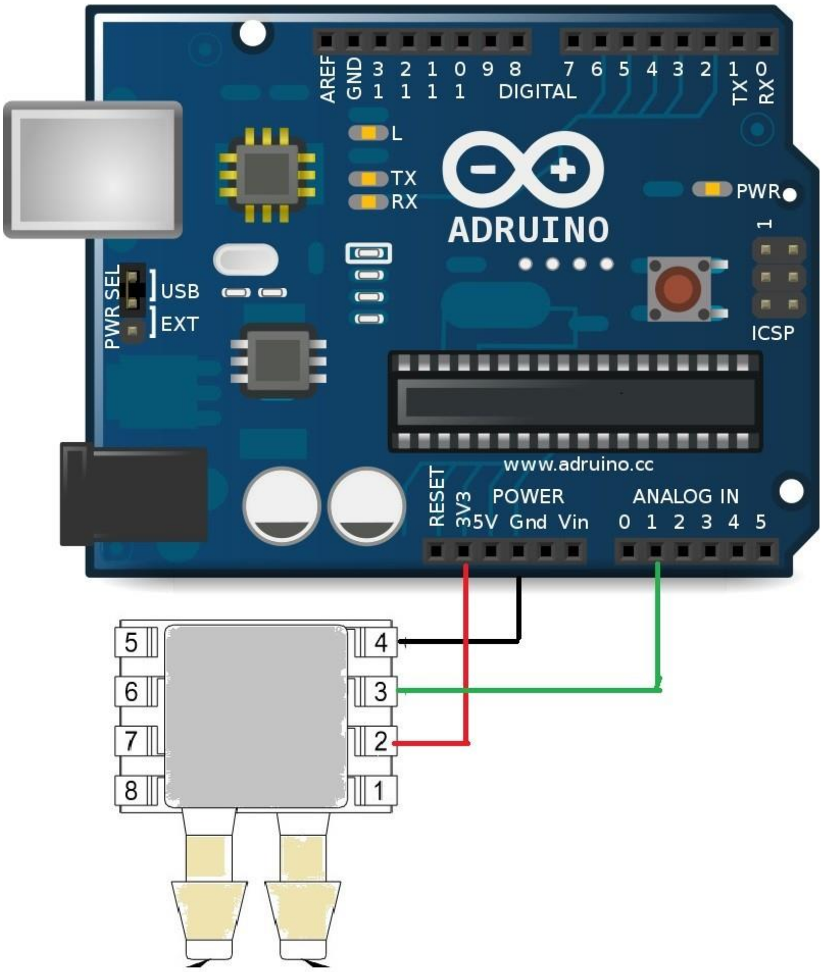

Figure 2.5: Connections

Chapter 3

Software Description

3.1 Converting between digital and analog signals

The two ports of the differential pressure sensor takes in the absolute pressure and then derives the differential pressure. When an analog signal (like the voltage produced by our pressure sensor) is processed by the Arduino, it is converted to a digital value between 0 and 1023(since the ADC of Arduino is 10-bit).

inputVolt = analogRead(analogInPin);

$$volt = inputVolt \times \frac{3.3}{1023.0};$$

3.2 Converting from a voltage to pressure

The purpose of the pressure transducer is to turn a pressure difference into a voltage, but now we want to determine the pressure difference based on a given voltage .Hence the pressure could be computed from voltage again.

$$pressurepsi = \frac{15}{2} \times (volt - 1.65);$$

The pressure obtained is in pounds per sqaure inch(psi).

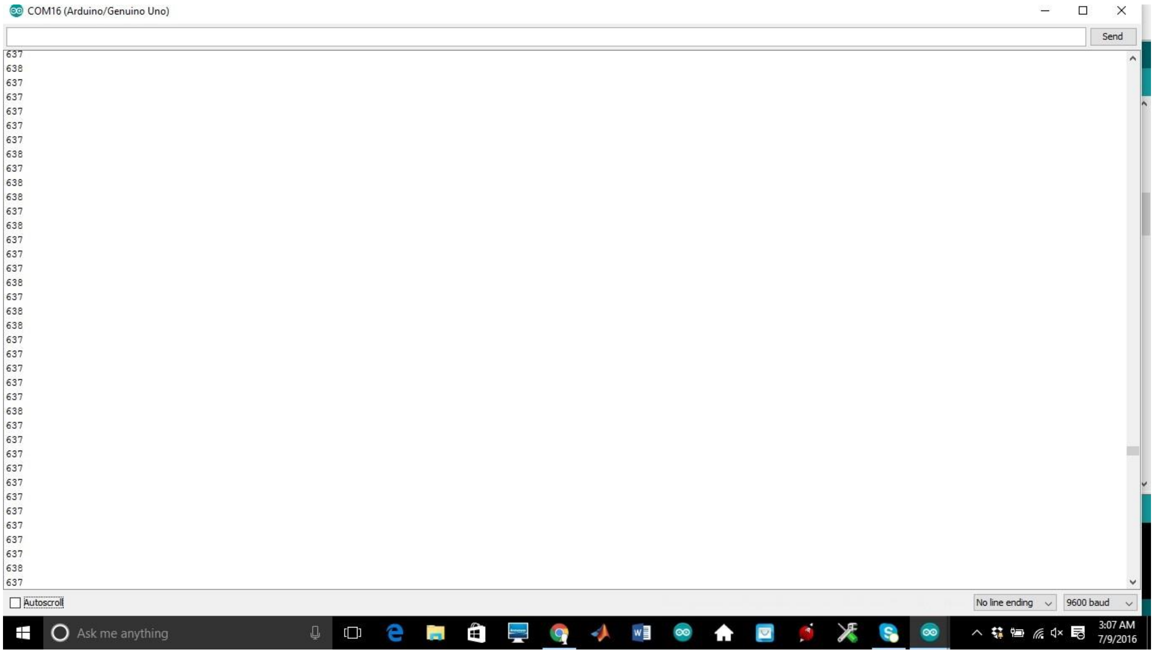

Figure 3.1: Offset values obtained at equillibrium

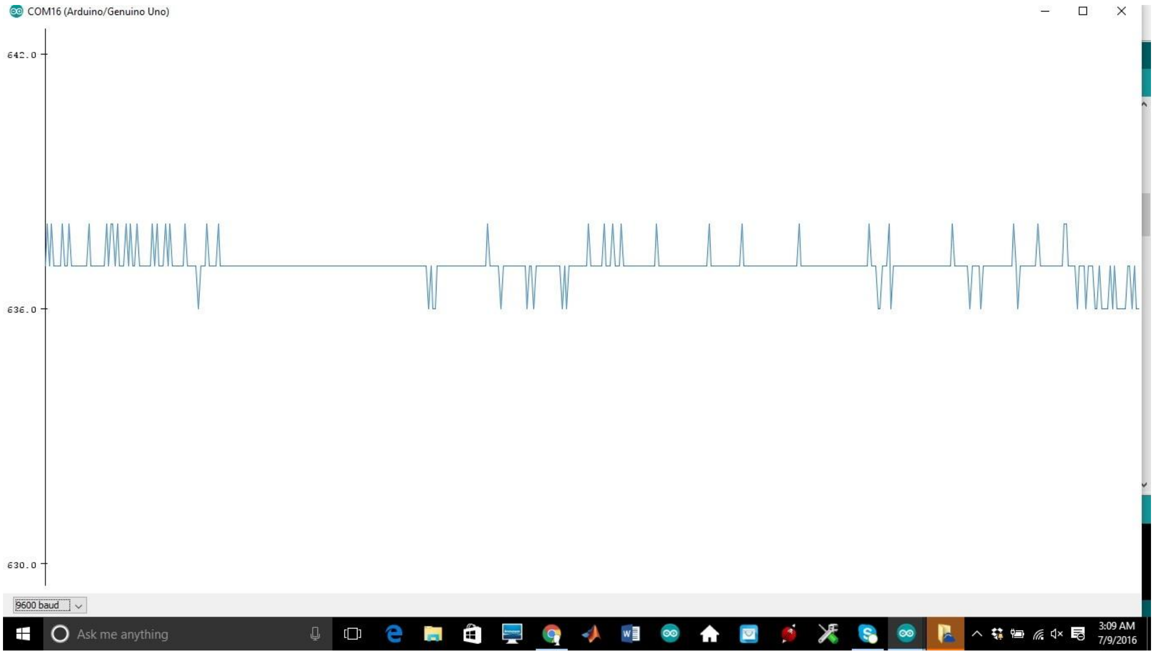

Figure 3.2: Plot of offset when no air is blown through Spirometer

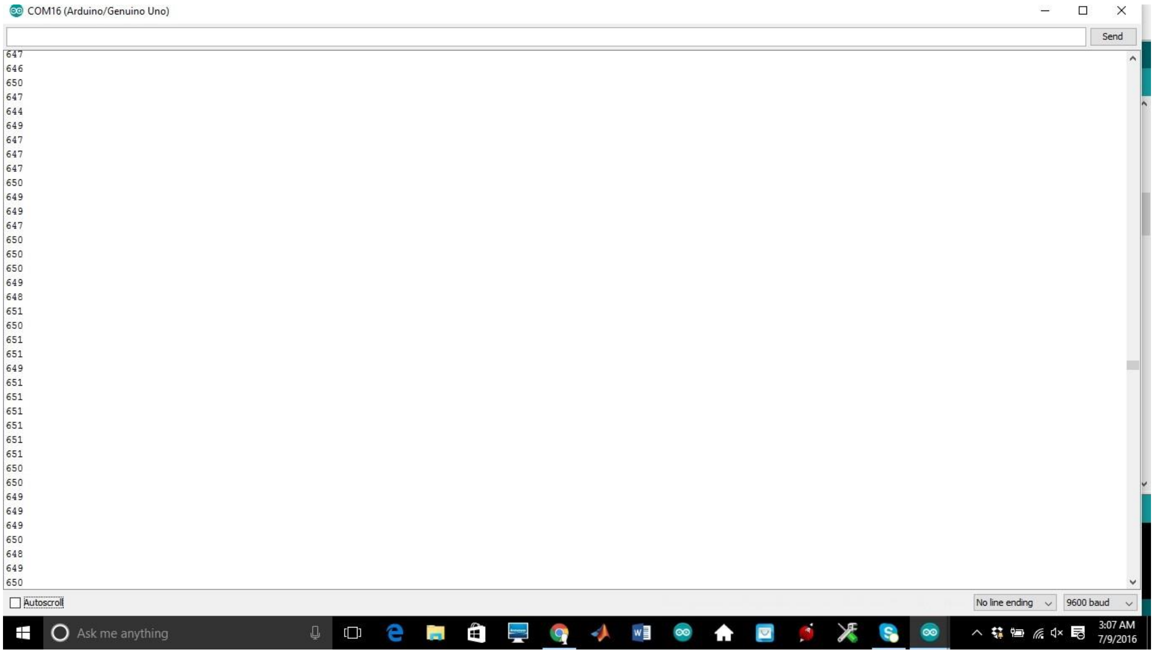

Figure 3.3: Offset obtained during exhalation

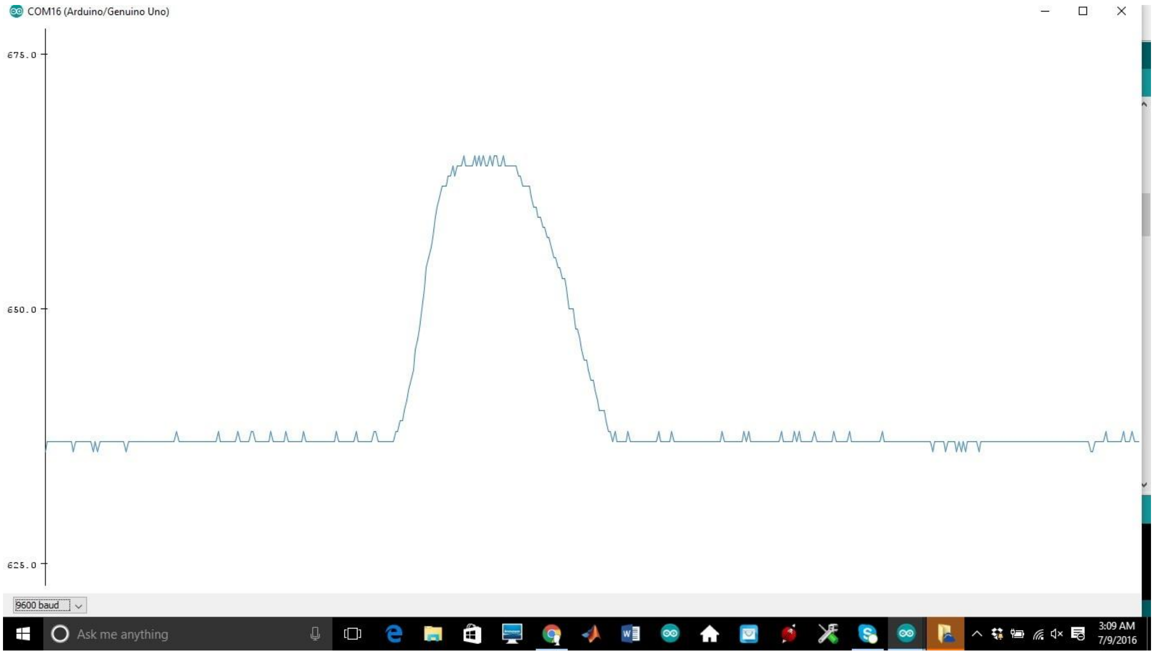

Figure 3.4: Plot of Offset obtained during exhalation

3.3 Calculation of the volumetric flow

Using the equations of fluid dynamics, the volumetric flow can be computed.

$$dP = (\frac{W^2}{2\rho}) \times (\frac{1}{A_2^2} - \frac{1}{A_1^2})$$

The mass of air that flows:

$$massFlow = 1000 \sqrt{\frac{2\rho(pressure_p a)}{\frac{1}{area_2^2} - \frac{1}{area_1^2}}};$$

Volumetric flow of air:

$$volFlow = \frac{massFlow}{\rho};$$

This gives the value of the volume flow of air through a cross-section, per second.

Chapter 4

Sampling times and process

For each data to be recorded, the samples are made to be sufficient enough. The first state to be performed is initialization. It acquires an average of the offset value, meaning a baseline. The signal is stabilized by the moving average process which is used as a filter for the signal AD conversions.
Afterwards, 100 samples are used to calculate the average offset. This state is run each time when turning on the system.
After receiving a request from the host PC to start a new spirometry measurement, the device has to change its state. In this state, the microcontroller is continuously checking the value of the sigal and comparing it with a threshold value.
After the result of the ADC is less than the threshold, it pulls the value to zero. With each new sample, the microcontroller calculates and updates the spirometry parameters, including flow and volume, which are sent to the host PC via USB communication.
When the signal value is equal to the baseline; the state changes to the sending results state. Flow chart for this state is shown in Figure.
The refined values are then used to calculate the various parameters. Once the end of exhalation is detected, the next state is the sending results state. It consists of filling the results buffer with the final results obtained with the calculation routines, and generates an event to send that information to the computer.

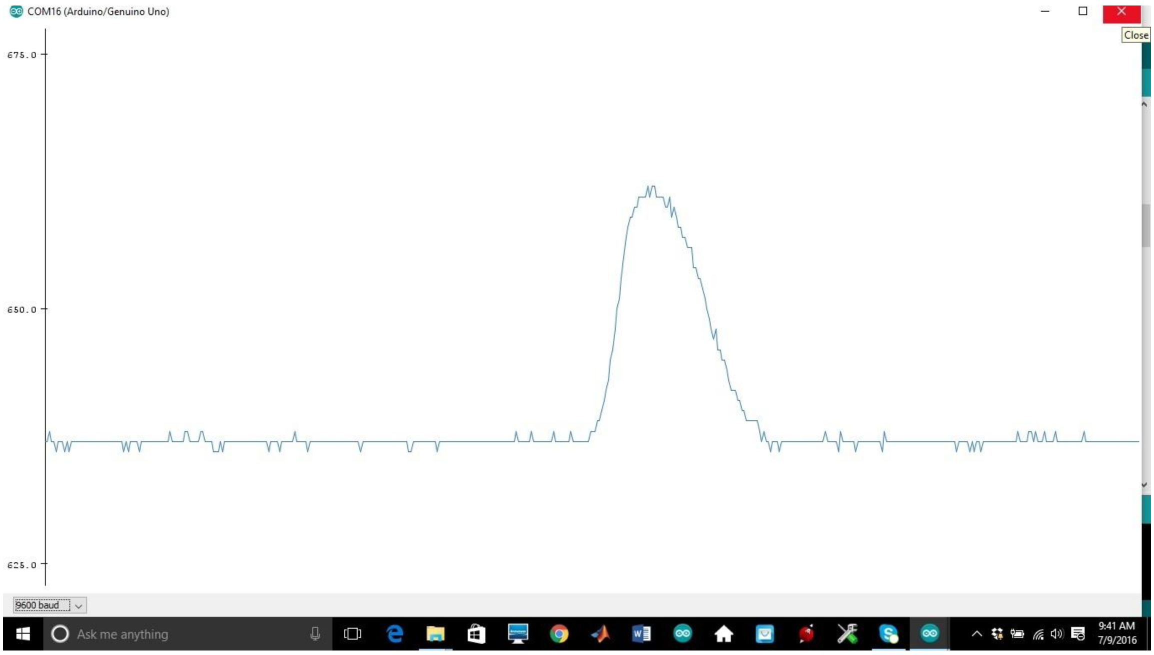

Figure 4.1: Exhalation plot after moving average filteration

Chapter 5

Calculation of the parameters

The formulas used to calculate the spirometry parameters are embedded in the MCU software.
With each new sample taken by the ADC, the next calculations are performed:

If $ADCValue > Signaloffset$:

$$Laverage = (ADCValue - Signaloffset) \times Conversionfactor$$

Where Laverage is the final refined value of input pressure.

If $(Laverage < 200)$, then $Laverage = 0$.
If $(Laverage > 200)$, then Laverage could be used to find the various parameters for spirometer like

1. Forced Vital Capacity(FVC)
2. Forced Vital Capacity in 1 second (FVC1)
3. Peak Expiratory Flow(PEF)

Since the Arduino does not have the capability to perform integrals, we have to manually add up our volumetric flow rate over time. Since volumetric flow rate is simply volume over time, volume is sum of the volumetric flow rate over time.
$FVC = volFlow \times dt + FVC$;
$dt = 0.001$;
Forced Vital Capacity in 1 second is the amount of air exhaled in initial first

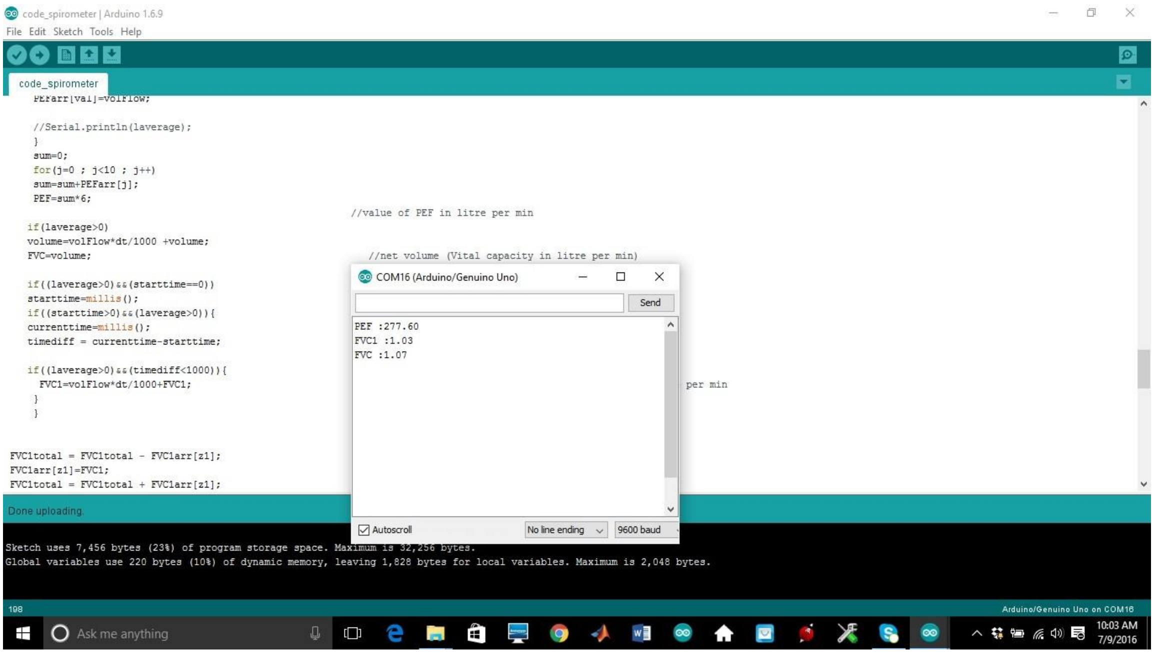

Figure 5.1: The values of FVC, FVC1, PEF on exhalation

second of the exhalation. Peak Expiratory Flow (PEF) is the maximum instant flow detected during exhalation. It could be executed with the logic as shown.

```
minimum =PEFarr [ 0 ] ;
for  ( i =1 ; i <10 ; i ++)   {
if  ( minimum > PEFarr [ i ] ) {
minimum=PEFarr [ i ] ;
val=i ;
}
}
if ( volFlow >PEFarr [ val ] ){
PEF[ val ]= volFlow ;
}
```

PEF, FVC, and FVC1 thus recorded are then sent, and displayed.