A NOVEL MICROCONTROLLER-BASED PRESSURIZED FLUSHING SYSTEM FOR WATER CONSERVATION IN SMART HOMES AND PUBLIC SPACES

BY

WISAM MOHAMED WAHID

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

May 2016

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **WISAM MOHAMED WAHID** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING**

Dr. Yahya E. Osais
(Advisor)

Dr. Jihad H. AlSadah
(Co-Advisor)

Dr. Tarek Sheltami
(Member)

Dr. Mohammed Mahmoud
(Member)

Dr. Esmail M. A. Mokheimer
(Member)

For:

Dr. Ahmad Almulhem
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

9/6/16
Date

To my beloved parents, sisters and brother-in-law

ACKNOWLEDGMENTS

All praise to Allah, the Cherisher and the Sustainer of the worlds, none is worthy of worship but Him. Peace and blessings be upon the beloved Prophet Muhammad (PBUH). My thanks to King Fahd University of Petroleum and Minerals for providing a great environment for education and research. I extend my deepest gratefulness to my thesis advisor Dr. Yahya E. Osais for his continuous backing, patience and inspiration. He stood by me in all the difficult times and was ever ready to provide support at any time during my tenure in the university. I would also like to thank my thesis co-advisor Dr. Jihad H. AlSadah for his support and encouragement. I would also like to thank my thesis committee members Dr. Tarek Sheltami, Dr. Mohammed Mahmoud and Dr. Esmail Mokheimer for their time and valuable comments. I would also like to thank the Deanship for Scientific Research (DSR) at KFUPM for financial support through research group project **IN131054**. I would like to thank Mr. Qassim for helping me whenever I needed very promptly. My deepest gratitude and warmest affection to my parents, Mohamed Wahid and Sadiqua Wahid. I could never have done this without their support, encouragement and unconditional love. Their prayers and blessings made me sail through every difficulties, shaped me into proper human and made me what I am. They are the source of inspiration and their efforts towards me are beyond words. I would also like to mention my dear sisters Wafeeqa and Wania for their phenomenal display of support and love, thank you for understanding and loving me. I would like to thank my brother-in-law, Naveed Ahamed for his support and encouragement. I would also like to thank all my friends especially Mohammed Faizuddin, Mustafa Qadri, and friends from KFUPM whose presence made my stay at KFUPM enjoyable and informative.

TABLE OF CONTENTS

ACKNOWLEDGMENTS V

TABLE OF CONTENTS VII

LIST OF TABLES XI

LIST OF FIGURES XII

LIST OF ABBREVIATIONS XV

ABSTRACT XVI

ملخص الرسالة XVIII

CHAPTER 1 INTRODUCTION 1

1.1 Water Scarcity 1

1.2 Cyber Physical Systems (CPS) 2

1.3 Water Conservation 3

1.4 Problem Statement 5

1.5 Need for a Microcontroller-based Approach 6

1.6 Thesis Objectives 8

1.7 Thesis Overview 9

CHAPTER 2 LITERATURE REVIEW 10

2.1 Water Sustainability 10

2.2 Water Cyber-Physical Systems 11

2.3 Applications of CPS for Water Sustainability 13

2.4 Sensing and Instrumentation 15

2.5 Wireless Heterogeneous Networks 16

2.6 Inferences 17

2.7 Historic Developments of the Flushing System 18

2.8 Flushmate™ 21

2.8.1 Operating Principle of Flushmate™ 21

2.8.2 Flushmate™ Curve 23

2.9 Water Sense Label 23

2.10 Dual Flushing Technique 24

CHAPTER 3 PHYSICAL SYSTEM DESIGN 28

3.1 Design 29

3.2 Arduino 32

3.3 Arduino Uno 33

3.4 Relay Board 34

3.5 Level Sensor 35

3.6 Pressure Sensor 37

3.7 Air Compressor 37

3.8 Water Pump 38

3.9 Electrical Feed Through 39

3.10 Solenoid Valve 40

CHAPTER 4 PHYSICAL SYSTEM INTERFACE 43

4.1 Interfacing of the Level Sensor 43

4.2 Interfacing of the Pressure Sensor 46

4.3 Need for a Microcontroller 47

4.4 Characterizing the Ability of the Physical System 47
4.4.1 Schematics 50
4.4.2 Pseudo code 53
4.4.3 Algorithm 54
4.5 Designing an 8 bar flushing system 58
4.5.1 Schematics 60
4.5.2 Pseudo code 62
4.5.3 Algorithm 62
Appendix 4.1 Description of the Behavior of the Scalability Test 66
CHAPTER 5 SIMULATION 70
5.1 Simulation Overview 71
5.2 System Design 72
5.3 Main Packages of the Simulated System 73
5.3.1 User Interaction Package 75
5.3.2 Toilet Package 76
5.4 Self-Pressurized Tank (First Module) 82
5.4.1 Hydraulic Power Package for the Self-Pressurized Tank 83
5.5 The Pressurized Tank and Its Components 84
5.5.1 The Pressurized Tank Generation 85
5.5.2 Compressor and Pump Package 95
5.6 Two User System (Second Module) 98
5.7 Twelve User System (Third Module) 101
CHAPTER 6 PHYSICAL SYSTEM RESULTS 104
6.1 Scalability of the system 104

6.1.1 Comparison of flows at various pressures 106

6.2 Flushing system at 8 bar pressure 108

CHAPTER 7 SIMULATION RESULTS 110

7.1 Simulation Result for the Self-pressurized tank 110

7.2 Simulation Result for the Pressurized tank with 2 User 112

7.3 Simulation Result for the Pressurized tank with 12 Users 119

7.4 Comparison with the Physical System Results 125

7.5 Comparison with Flushmate™ 126

CHAPTER 8 CONCLUSION AND FUTURE WORK 127

APPENDICES 129

Appendix A: Images of the physical prototype 129

Appendix B: Arduino Code for characterizing the stability of the system 133

Appendix C: Arduino Code for the Toilet flushing system with 8 bar pressure 139

Appendix D: State Transition Table 145

Appendix E: Code for generating the Pressurized Tank in Simulink 148

Appendix F: Matlab Code for analyzing the pressure, volume and interval of discharge 153

Appendix G: Matlab Code for analyzing the flush 154

REFERENCES 156

VITAE 161

LIST OF TABLES

Table 3- 1 Specification of Arduino Uno [42] 34
Table 4- 1 Signals used in the design 54
Table 4- 2 Inputs to the Arduino for the Scalability Tests 55
Table 4- 3 State Description for the Scalability Test 55
Table 4- 4 Inputs to the Arduino for the Flushing System 63
Table 4- 5 State Description for the Flushing System 63
Table 4- 6 Inputs to the System for the Scalability Tests 66
Table 4- 7 Inputs to the System and their values 66
Table 4- 8 Signals used as the Output of the System 67
Table 4- 9 State Name and Description for the Scalability Test 67
Table 5- 1 Self-pressurized tank specifications 77
Table 5- 2 Module components of the New Pressurized tank 84
Table 6- 1 Volume of Water flowing out at 4 bar pressure for fixed time interval 105
Table 6- 2 Volume of Water flowing out at 6 bar pressure for fixed time interval 105
Table 6- 3 Volume of Water flowing out at 8 bar pressure for fixed time interval 106
Table 6- 4 Comparison of Volume of Water flowing out at different pressures for varying time intervals 106
Table 6- 5 Average Flow Rate at Different Pressures 107
Table 6- 6 Steady State Flow Rate at different pressures 108
Table 6- 7 Volume of water with in the tank at the fixed time 108
Table 7- 1 Toilet Average Result for 2 Users connected to Pressurized Tank 119
Table 7- 2 Discharge Result for different scenarios in the 12 user system 123
Table A- 1 State Transition Table 145

LIST OF FIGURES

Figure 2- 1 Architecture of a typical water CPS [10]. 12
Figure 2- 2 A Water Distribution System [14]. 13
Figure 2- 3 Operational stages of Flushmate™ [8]. 22
Figure 2- 4 Flushmate™ Drain Carry performance [8]. 22
Figure 2- 5 Flushmate™ Bowl Design. 22
Figure 2- 6 Flushmate™ Curve [8]. 23
Figure 3- 1 Model of the proposed system. 28
Figure 3- 2 General User Interface for Light or Heavy Flushing 29
Figure 3- 3 The Built Prototype 30
Figure 3- 4 Arduino Uno Board [42]. 33
Figure 3- 5 8-Channel Relay Board [43]. 35
Figure 3- 6 Liquid Level Sensor [45]. 36
Figure 3- 7 Pressure Sensor [47]. 37
Figure 3- 8 Air Compressor [48]. 38
Figure 3- 9 Water Pump [49]. 38
Figure 3- 10 Electrical Feed Through. 39
Figure 3- 11 Solenoid Valve at different states. 40
Figure 3- 12 Stainless Steel Solenoid Valve [52]. 41
Figure 3- 13 Plastic Solenoid Valve [53]. 41
Figure 4- 1 Voltage Divider Circuit. 43
Figure 4- 2 Typical eTape Sensor Output [46]. 45
Figure 4- 3 Physical Setup to characterize the ability of the system. 49
Figure 4- 4 Schematics of the Arduino to characterize ability of the system. 51
Figure 4- 5 Sensor Wiring. 51
Figure 4- 6 Relay connections from the Arduino to the various components. 52
Figure 4- 7 Connections for the flush and to release the pressure. 53
Figure 4- 8 Flowchart to characterize the ability of the system. 57
Figure 4- 9 Physical system designed for 8 bar flushing system. 59
Figure 4- 10 Schematic of the Arduino for the 8 bar flushing system. 61
Figure 4- 11 Flowchart for the 8 bar flushing system. 65
Figure 4- 12 State Transition Diagram 69
Figure 5- 1 Main Packages. 74
Figure 5- 2 User interaction package. 75
Figure 5- 3 Parameters of the user interaction signal. 75
Figure 5- 4 Toilet Subsystem. 77
Figure 5- 5 Hydraulic Subsystem. 78
Figure 5- 6 Elbow and Pipeline parameters. 80
Figure 5- 7 Control System. 80

Figure 5- 8 Dataset Subsystem. 82
Figure 5- 9 Self Pressurized Tank Design. 82
Figure 5- 10 Hydraulic Power System of Self-Pressurized Tank. 83
Figure 5- 11 Tank Configuration of the Self-Pressurized Tank. 84
Figure 5- 12 Hydraulic-Pneumatic Pressurized Tank. 85
Figure 5- 13 Hydraulic Power System. 91
Figure 5- 14 Outlet Control Group for 2 user system. 92
Figure 5- 15 Outlet Control Group for 12 user system. 92
Figure 5- 16 Module Tank Configuration. 94
Figure 5- 17 Hydraulic Power System. 96
Figure 5- 18 Pneumatic Power System. 96
Figure 5- 19 Level Control System. 97
Figure 5- 20 Main Packages of the tank connected to the 2 toilets. 99
Figure 5- 21 User Interaction system for Multiple Users. 100
Figure 5- 22 Main packages of the tank connected to 12 toilets. 102
Figure 5- 23 The 12 toilets which are connected to the Pressurized Tank. 103
Figure 6- 1 Volume of Water flowing out Vs. Time at 4 Bar Pressure. 105
Figure 6- 2 Volume of Water flowing out Vs. Time at 6 Bar Pressure. 105
Figure 6- 3 Volume of Water flowing out Vs. Time at 8 Bar Pressure. 106
Figure 6- 4 Graphical Comparison of Volume of Water flowing out Vs. Time at Different Pressures. 107
Figure 6- 5 The Degradation of the volume of the water within the tank when the flush is on. 109
Figure 7- 1 The interval, volume and pressure of water discharged for a Toilet which requests for a 3 L flush. 111
Figure 7- 2 The interval, volume and pressure of water discharged for a Toilet which requests for a 1 L flush. 111
Figure 7- 3 The interval, volume and pressure of water discharged for Toilet 1, when 2 users request for a 3 L flush at the same time. 113
Figure 7- 4 The interval, volume and pressure of water discharged for Toilet 2, when 2 users request for a 3 L flush at the same time. 113
Figure 7- 5 Decrease in Hydraulic Volume in the tank due to the flushes. 114
Figure 7- 6 Pressure in the pipeline for both the toilets. 114
Figure 7- 7 Flow rate (L/sec) at each toilet. 115
Figure 7- 8 Hydraulic Volume, Gas Volume and their Ratio within the tank during the simulation. 115
Figure 7- 9 Flowrate for a single toilet (3L flush). 116
Figure 7- 10 The interval, volume and pressure of water discharged for Toilet 1, when 2 users request for a 1 L flush at the same time. 116

Figure 7- 11 The interval, volume and pressure of water discharged for Toilet 2, when 2 users request for a 1 L flush at the same time. 117
Figure 7- 12 Flowrate for a single toilet (1L flush). 117
Figure 7- 13 Pressure in the pipeline when the users request for a 3L and a 1L flush at different time. 118
Figure 7- 14 Flow when the users request for a 3L and a 1L flush at different time across the different toilets. 118
Figure 7- 15 The interval, volume and pressure of water discharged for a Toilet which requests for a 3 L flush. 120
Figure 7- 16 The interval, volume and pressure of water discharged for a Toilet which requests for a 1 L flush. 121
Figure 7- 17 Hydraulic Volume, Gas Volume and their Ratio within the tank during the simulation. 121
Figure 7- 18 Pressure in the pipeline when the users request for a 3L and a 1L flush at different times. 122
Figure 7- 19 Comparison of Results obtained from physical system and simulation. ... 125
Figure 7- 20 Comparison of the flow rate of our system with flushmate. 126
Figure A- 1 Side View of the Pressurized Tank. 129
Figure A- 2 Top view of the pressurized tank. 130
Figure A- 3 Top view of the prototype. 130
Figure A- 4 The air compressor, and the flushing container. 131
Figure A- 5 Arduino-Relay interface along with various switches. 131
Figure A- 6 Different power supplies for solenoids and the air compressor. 132
Figure A- 7 The Main Solenoid flushing valve. 132

LIST OF ABBREVIATIONS

CPS	:	Cyber Physical System
Water Distribution Systems	:	WDNs

ABSTRACT

Full Name : WISAM MOHAMED WAHID

Thesis Title : A Novel Microcontroller-based Pressurized Flushing System for Water Conservation in Smart Homes and Public Spaces

Major Field : Computer Engineering

Date of Degree : May 2016

Water scarcity is a serious problem that affects the entire world and in particular countries in dry regions such as Saudi Arabia. Appropriate measures have to be taken to minimize the usage of water. Toilet flushing consumes a significant percentage of water in domestic use. Water should be conserved without compromising the hygienic function of this process. In this thesis, we propose a microcontroller-based pressurized flushing system that significantly reduces water consumption when compared to the conventional gravity-based flushing system. The newly proposed system enables a dual flushing option, where the quantity of water flushed depends on the type of waste. The proposed system is a Cyber Physical System (CPS) that contains a microcontroller, various sensors and pumps to perform the physical process, which is controlling the flushing of the pressurized water. The microcontroller can process various analog and digital signals, along with many other functionalities which helps to build the proposed system.

The new system, basically uses compressed air to pressurize the water which is used for flushing. Water that comes under high pressure is more effective in clearing the contents of the bowl. Thus, less amount of water is needed. The process is too fast for manual operation; hence the need for a microcontroller based approach. Typical flushing time is

300-800 milliseconds for a pressure range of 4-8 atmospheres. The gravity-based system takes approximately 5 seconds for a pressure range of 0.05-0.2 atmospheres.

The process requires timed steps where both sensor inputs and actuation actions are monitored and controlled by a microcontroller. This system is designed to use lesser amount of water than the present day scenario, while maintaining the efficacy of the flushing system. The system is also simulated to check the efficiency of the system in a multi-user environment as in public spaces. The proposed system provides significant water savings with adequate quality of service. It saves around 62 liters per day per person out of 74 liters used in conventional systems i.e. 84% of water is saved using this CPS when compared to the conventional technique.

ملخص الرسالة

الاسم الكامل: وسام محمد وحيد

عنوان الرسالة: نظام جديد لإستخدام الماء المضغوط بمساعدة الميكروكنترولور من أجل الشطف لترشيد استعمال المياه في البيوت الذكية والأماكن العامة

التخصص: هندسة الحاسب الآلي

تاريخ الدرجة العلمية: مايو ٢٠١٦

تعتبر ندرة المياه من المشاكل الخطيرة التي تؤثر على العالم بأسره وبالذات في المناطق الجافة مثل المملكة العربية السعودية. وبالتالي لابد من اتخاذ العديد من التدابير للحد من استهلاك المياه. تستهلك عملية شطف المرحاض من اجل التخلص من الأوساخ كمية كبيرة من الماء لأعتمادها على الضغط الجوي. ولكن يمكن تقليل كمية المياه اللازمة لإنجاز نفس المهمة اذا تم ضغط المياه ضغط عالي. لا يمكن لنظام الشطف التقليدي التعامل مع هذا الضغط العالي وبالتالي لابد من استخدام الميكروكنترولور من اجل توقيت العملية كلها وحتى يمكن استخدام صمامات الكترونية ذات سرعة مناسبة. يقوم النظام المقترح بتوفير كمية كبيرة من المياه مع الحفاظ على مستوى الخدمة. أنه يوفر النظام المقترح حوالي ٦٢ لتراً يوميا للشخص الواحد مقارنةً ب 74 لتراً تستخدم حالياً في الأنظمة التقليدية.

CHAPTER 1

INTRODUCTION

Water is a natural resource on which all lifeforms depend on. The ever increasing demand has decreased the supply of available fresh water. Water scarcity has become a prominent issue across the world. Hence, water conservation should be the prerequisite for any intelligent water control and monitoring system. Water is typically wasted in homes and public spaces by its excessive use for flushing. This thesis aims at conserving water used for flushing with the help of a Cyber Physical System using techniques of Computer Engineering.

1.1 Water Scarcity

Water is the need of every living organism on the earth. It is an essential natural resource on which the world's economic and social functions depend. By 2050, the world population is expected to grow to 9.1 billion [1]. This unprecedented growth in population has led to an ever-increasing demand for fresh water.

Saudi Arabia is one of the most water scarce countries in the world. The main sources of useable water in Saudi Arabia are desalinated seawater and underground water. As for the underground water, it is receding at an alarming rate and most of it has already been depleted. This is due to the population in the Arab region which has increased by 43% in past two decades [2]. Water consumption due to household in Saudi Arabia is more than eight million cubic meter per day and this demand is increasing by 7.5% annually [3]. The receding underground water meets the 40% of water demand and the rest is met by desalinating seawater. The cost of desalinating seawater is 1-2$/m^3 which is likely to increase as the price of the fossil energy increases.

Water savings has two advantages:

- More water can be made available for future increase in demand.
- Less waste water is generated which lowers disposal cost.

An efficient water saving measures using Communication and Information Technology (CIT) should be seriously considered.

1.2 Cyber Physical Systems (CPS)

The term CPS was first used by Helen Gill to refer to the integration of physical system with computation system at the National Science Foundation in 2006 [4]. A CPS consists of sensors, network embedded controllers and actuators which make any physical system that adapts itself according to the system's state and it's interaction with humans. An actuator is a device, which is used to alter a physical quantity and is controlled by a voltage input. The sensors monitor the physical layer and periodically collect information about the physical layer. This information is also periodically sent to the software/computation layer where it is analyzed and a decision is made about the state of the physical layer. The actuators/controllers then change the state of the physical layer based on the input from the software layer. Thus, the two layers form a feedback loop where the computations affect the physical state and vice versa.

Designing a CPS requires one to have understanding of the physical process, sensor and networks; and the dynamics of joining the two layers. It is the intersection of these layers not the union that sets CPS apart from other systems. Recently CPS are often packaged with network interfaces which allows them to be a part of the internet. It is this trend of connecting the physical world with the information world using CPS and internet that has given rise to IoT.

This thesis provides a new approach for flushing systems using a CPS. Wherein, the physical layer interacts with the cyber layer with the help of sensors and actuators.

1.3 Water Conservation

Since the dawn of history, sanitation in fresh water is far in concern with the human. Due to the improper sewage system, diseases were common during earlier days. Then came the advent of toilets which effectively enabled formation of large cities without the public health concerns of the earlier generations.

In modern residential buildings, toilets are the major source of water consumption. Studies show that toilets represent almost 30 % of the indoor water utilization. This can be attributed partly to the old inefficient toilets.

Modern flush toilets utilize approximately 90 liters of water per capita per day [2]. Modern low flush toilets are designed in such a way as to use less water per flush i.e., approximately 6.1 - 4.5 liters of water [2]. Reducing the amount of water requires that the sewage treatment systems to accommodate more concentrated waste.

There are three general methods of human waste disposal:

1. Water free: This occurs in places with no sewage or drainage access, such as in deserts. This will result in people living far away from each other and not advisable for people living on high stories. This system is completely unsanitary and can promote the spread of diseases.
2. Vacuum flushing: This system uses very little water but requires high energy. The room should be air tight and there should be no gaps. In case of any gap, this flushing will not be reliable and it would take a long time to realize the source of the gap. This technique was tried, but was not effective in the residential buildings due to its inefficiency [5].

   The vacuum flushing has a high investment cost and is highly dependent of electric power supply. It may not be efficient as coarse materials may cause blockage of the collections system. This technique also requires a house-based or community-based vacuum station.
3. Pressurized Water flushing: This is one of the latest technologies and is also very reliable. Pressurized water flushing takes hygiene into consideration as it is a very important factor in the waste disposal system.

Pressurized Water Flushing can be utilized in two ways:

1. Gravity: This can be done by increasing the level of water such that the water can take away the waste with higher pressure at the time of the flush. However, the

siphon cannot be higher than 3 meters, as there are limitations on the height of the ceiling. In such toilets, the pull string is used when the water needs to be flushed. This method does not help with reducing the amount of water used.

2. Pressure: Another technique is the use of high pressure chambers. High pressure chambers helps in saving water, while maintaining the efficiency. Flushmate™ technology uses a high pressure chamber. This thesis describes an innovative technique which can be used as a high pressure chamber.

1.4 Problem Statement

Water is an essential resource for all lives. The modern style of living in large cities is enabled by a public hygienic practice. In particular, the toilet-flushing system is a key technology that enables people to live in proximity to each other without a negative public health impact. However, modern flushing technologies consumes as much as 30% of domestic water as per the United States Environmental Protection Agency [7]. This large amount of water represents a pressure on water production, water transportation, sewage lines, and sewage treatment facilities. In addition it has a significant environmental impact. The large amount of water used in flushing is because of the use of the simple gravitational flow of water in siphon effect. A recent technology is to use pressure assisted flushing by pressure tank utilizing water means as the source of pressure. Pressure-assisted toilets are water-efficient that can substitute the standard gravity toilets. Even though they appear to be similar to gravity toilets, they operate quite differently. When the tank is filled with water, an air-filled diaphragm inside the tank shrinks. When the toilet is flushed, the compressed air functions like a piston, pushing the water into the bowl with a strong force greater than the force created by gravity [8]. However, in this thesis, we propose to use pressurized water with the help of compressed air to increase the pressure ceiling beyond the aforementioned limitation.

We believe that this approach is more efficient and it will help in saving a huge amount of water. In order to achieve, the above objective, a new system that employs a microcontroller, sensors, actuators, and pumps need to be constructed. These elements form the cyber layer in the CPS. This system will also contain a mechanical subsystem that forms the physical layer in the CPS. This new approach will enable a flushing system to be operated in new ways that are not possible using only mechanical elements. The result will be significant saving of water. This system is also simulated to be used in a public environment.

1.5 Need for a Microcontroller-based Approach

The proposed system aims at controlling the flow of water at a very high pressure and serves this water to multiple users. This action cannot be achieved with mechanical valves, as the valves need to be switched on for a very short duration, in the order of a few hundreds of milliseconds. Hence, the system uses electronic valves which are controlled by a microcontroller.

The pressure in the tank is also monitored and controlled with the help of various sensors, which are placed in the tank and are interfaced with a microcontroller. This microcontroller also helps in maintaining the level of the water and the pressure within the tank. If there is a leak in the pressure in the tank, this leak can be detected and the pressure in the tank can be brought back to the initial pressure by the actuator connected to the microcontroller. The volume of water in the tank can also be calculated with the help of the sensors. If the volume of water within the tank is less than the permissible limit, the water is pumped into the tank.

In the simulation, the system uses a centralized water tank in a multi user environment, which can serve multiple users at the same time. Hence, there is a need for arbitration among users to offer different levels of quality of service. The system acts as an arbiter, wherein the pressurized tank is shared by multiple users at the same time. The system incorporates multiple requests at the same time and also schedules the user service while maintaining the desired quality of service. If more than one user requests a flush, the system is capable of flushing both users at the same time as well as providing a delay in the servicing of the flush to each user. This delay is in the order of milliseconds and is very negligible from the user's perspective. However, this delay helps in maintaining the desired quality of service.

The amount of water flushed out is highly dependent on the pressure. That is, if the pressure varies, then the time for which the valve is opened also needs to be varied with the help of the microcontroller. The mechanical valves can be opened and closed for a fixed duration only. However, in this system a smart technique is implemented wherein the valves are opened for a varying time duration in a multi user environment. The valve is opened until the user receives the requested amount of water. This valve opening is also controlled for multiple users with a centralized control system.

1.6 Thesis Objectives

This thesis proposes a pressurized flushing technique to significantly conserve water, while maintaining quality of service. Compressed air is used to pressurize the water in the tank, before the water is used for flushing. This high-pressure water can be used to flush away the waste while using lesser volume of water. The control system for this pressurized tank can flush the water depending on the type of the waste to be flushed. For heavy flushing, three liters of water is flushed out. On the other hand, for light flushing one liter of water is flushed out.

The proposed systems integrates the mechanical system with the electronic system to form a CPS. A prototype of the proposed system is built for a single user with dual flushing capabilities. The characteristics of this prototype are tested and the results are analyzed. The electronic layer contains sensors and actuators to provide control. It also contains special electronic valves appropriate for our application.

The proposed system has been simulated using the physical modeling tools provided by MATLAB [9] in a multi-user setting. Different scenarios, in which 12 users request different levels of flush at varying times have been tried and tested. The pressure with which the water is flushed and the time taken for the flush have been recorded with the help of various modeled sensors.

The contributions can be summarized as follows,

1. Design and develop the electronic component (i.e. cyber layer) of the newly proposed CPS,
2. Designing and building a prototype with a dual flushing feature for a single-user toilet and use it to collect measurements,

3. Developing a simulation model of the proposed flushing system to test multi-user scenarios, and
4. Studying the performance of the new system and compare it with that of the state of the art system.

1.7 Thesis Overview

This thesis presents a novel system in the area of pressurized flushing systems that helps to conserve water at homes and in public spaces. The rest of the thesis is organized as follows. Firstly, a detailed literature review is provided in Chapter 2. Chapter 3 provides information about the system design. Chapter 4 gives details of the hardware components used in building the prototype. Chapter 5 describes the design of the electronic layer in the CPS. Chapter 6 describes the simulation model. Chapter 7 discusses the performance of the physical prototype. Chapter 8 discusses the simulation results. Conclusion and future work are given in Chapter 9.

CHAPTER 2

LITERATURE REVIEW

In the past decade, cyber physical systems have found widespread acceptance and they are already being used in water monitoring and conservation applications. In this section, some of the recent works in this area are reviewed. Further, their advantages and limitations are highlighted. A review of the current commercial systems available in the market is also done.

2.1 Water Sustainability

Water is the life force of the planet. According to Zhaohui Wang, Houbing Song and David W. Watkins [10], for the sustainable development of the present day world water sustainability holds key importance as it is closely linked with a number of worldwide challenges such as industrialization and population growth. Attaining water sustainability, in line, requires attaining worldwide accessibility to clean drinking water, hygiene and sanitation; enhancing the water use efficiency for commercial determinations; strengthening accountable, participatory and equitable water governance; enhancing the control and treatment discarded water and safeguarding the quality of water; and decreasing the risks of human-induced and natural tragedies related to water.

Even though water is a resource which can be renewed, the availability and quality of water may not be desirable to serve its purpose and may affect the systematic functioning of the ecosystem. Water scarcity may occur according to the seasons or according to the geographic locations which depends on the availability of the community water systems or the regional river basins or the presence of groundwater aquifers.

With the presence of these critical challenges of the water resources, Millett and Estrin [11], promoted the "green" applications of information and communications technology (ICT) in order to achieve the sustainability of water. Green ICT has many advantages, which may include improving the quality of water, providing a more effective emergency response, and increasing the public awareness of the various environmental challenges. The authors predict various technologies such as communications and networking, sensor technology, hydrodynamic modeling, data analysis, and human-centered decision support systems in order to facilitate an intelligently networked water cyber-physical systems (CPS).

2.2 Water Cyber-Physical Systems

According to Lee [12], Cyber-physical systems (CPS) are networked systems with intelligently embedded with actuators processors and sensors, that are intended to interact and sense with the outer world (counting human consumers), and upkeep real-time, assured efficiency in critical-safety applications, as described in a Cyber-physical systems Vision Proclamation printed in 2012 by the Central Networking and IT Research and Development Program's Cyber-physical systems High-ranking Directing Group. The interaction between the "physical" and "cyber" components amongst the Cyber-physical systems is complex: computing, networking, sensing, and regulation should be intensely incorporated in each constituent of Cyber-physical systems, and the Cyber-physical systems constituents should be interoperable with a concentrated strategy.

Wang et al. [10], presented a typical water CPS architecture which is depicted in Figure 2-1. According to authors, the water CPS monitors water quality conditions in real-time, and detects water pollution events quickly.

The authors mentioned that in order To develop a water CPS, the followings are needed:

- Communications, sensing, and networking technologies to enable reliable, flexible, and high efficient dispersed networking within a water CPS, Computing technologies such as computational modeling, data management, machine learning, and other tools to understand, address, and communicate water sustainability challenges.
- Predictive and adaptive classified hybrid\ control technologies are critical to attain closely synchronized and coordinated interactions and actions in a water CPS that is essentially distributed, noisy and synchronous.

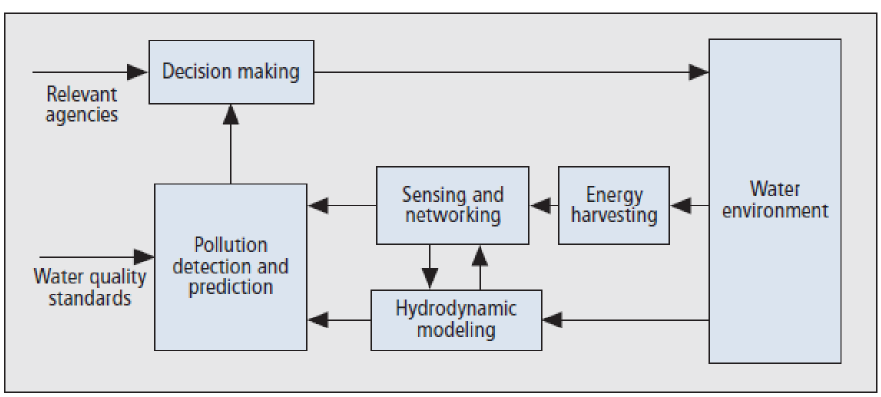

Figure 2- 1 Architecture of a typical water CPS [10].

2.3 Applications of CPS for Water Sustainability

Quaglia [13] mentioned in her paper that the applications of water CPS can set the path towards water sustainability by increasing the public confidence in terms of reliability, efficiency and security in various categories of water systems. There have been many advancements in the field of real-time water quality monitoring, networking and computing, which have played a tremendous role in environmental monitoring and interaction.

Water Distribution Systems (WDNs)

Lin et al. [14] discussed Water distribution system monitoring and regarded it as an important application of CPS in the water sector. According to authors, WDNs are evolving Cyber-physical systems fields. Physical constituents, e.g., reservoirs, pipes, and valves, are joined with the software and hardware that provides smart allocation of water. An example is showed in Figure 2-2.

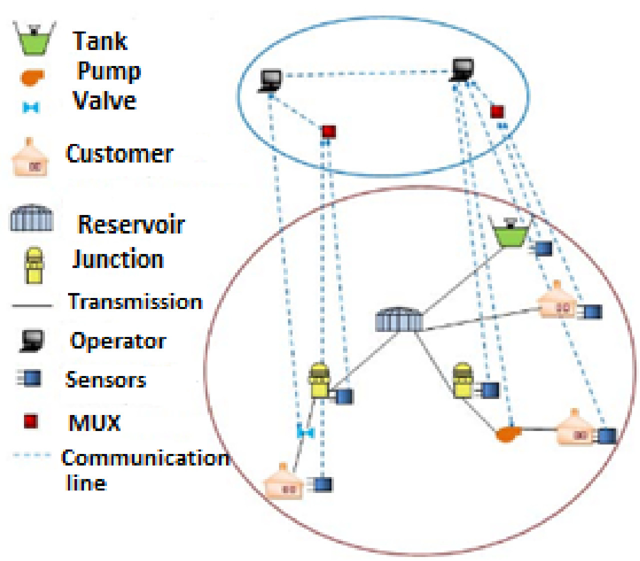

Figure 2- 2 A Water Distribution System [14].

According to Suresh [15], the main goal of Water Distribution Systems is to offer a reliable cause of drinkable water to the community. Information for instance, water quantity (pressure head and flow), demand patterns, and quality of water (minerals and contaminants) is vital in attaining this aim, and valuable in regulating conservation determinations and classifying susceptible regions necessitating monitoring and fortification. Sensors disseminated in the physical framework gather this information, which is provided to systems (frequently dispersed) operating on the cyber framework. These procedures offer assessment provision to hardware controllers that are utilized to organize chemical structure (quality) and the distribution (quantity) of the water. Mohamed and Moniri [16], are of the view that as Water Distribution Systems become more complex and larger, their consistency comes into question. The growing usage of Supervisory Control and Data Acquisition systems correspondingly increases apprehensions about the weakness of Water Distribution Systems.

Lin et al. [14], discussed in their paper a process for an intelligent Water Distribution System simulation, for example physical features of the Cyber-physical systems and cyber (computing) are precisely and accurately exemplified. EPANET and Matlab, correspondingly, are utilized to simulate the physical infrastructures and computing of smart WDNs.

Nasir [17], presented a human-centric CPN architecture of in-pipe water feedback and monitoring system. This scheme includes the physical infrastructure for water delivery, in cooperation with the software and hardware maintained smart agents for allotment of water, detection of leakage and pollution spread control. An agent centered method for linking the physical and cyber layers is designed, where the agent gets info from sensors observing the physical constituents and offers this info to the CPN.

2.4 Sensing and Instrumentation

With the advancements made in the field of computer and electrical engineering, there has been a significant reduction in the cost, size and power requirements of various digital electronics. The use of devices which cost less and consume less power but have storage and capabilities to process data, has led to a significant increase in wireless sensing in a variety of applications.

Kaur and JayPrakash [21] discussed in their paper that in order to achieve water sustainability, there should be monitoring of multiple parameters continuously such as flowrate, conductivity, turbidity, dissolved oxygen and the presence of various pollutants. The sensors for a monitoring system which is autonomous need to be accurate as well as stable for longer durations of time. If these sensors are exposed to harsh temperature changes or weather, the durability of these sensors may be affected. However, these sensors can be properly packed with the addition of high-performance electronic components for its long term use in spite of the harsh environments. One more issue, which can affect the longevity of the many under water sensors is the biofouling effect. The biofouling effect is when the aquatic flora and fauna, grow and spread around the sensors. The sensors effected are prevented from operating normally.

Mathur [22] discussed that many techniques which have been used to decrease or limit the biofouling effects. Manual cleaning of the sensors by scraping or wiping the sensors from time too time is efficient in cleaning but requires a lot of labor. Applying biocides around the sensors is a common technique which does not need much labor and can be used for long term deployment of the sensors. These biocides are generally added into the surfaces of the sensor housings or as coating layers around the sensors to protect the sensor housings to be slowly leached out. Electro-mechanical techniques have also been used to reduce biofouling. According to Mathur, in order to remove the biofouling effect the mechanical vibrations can also be used. Direct electrification of organisms has also been tested to remove fouling organisms from the sensor surface.

2.5 Wireless Heterogeneous Networks

Wang et al. [24], discussed the possibility of using the wireless communication technology for monitoring water quality and various other parameters. This is due to the fact that access to the aquatic environment directly is very difficult. Various sensors and actuation modules can be enabled with wireless communication technologies which can be remotely controlled for executing tasks as well as collecting the monitoring information and sending them to the centralized computer for the analysis of data. The aquatic environment is very complex and requires intelligent networks for the real time monitoring of the aquatic conditions. These intelligent networks can also be a collection of several types of sub-networks.

In the aquatic environment sound is used as the carrier of information to communicate between the under-water nodes, hence there is a presence of acoustic modem. Sound is used as the carrier of information as there is a large attenuation of electromagnetic waves in water. The data samples or other relevant information can be collected by the surface buoys which are equipped with various sensors. These buoys are equipped with radio-frequency modems to communicate with the control center as well as other buoys which are present above the water surface. These buoys also have various acoustic modems to communicate with nodes below the surface of the water [24].

2.6 Inferences

It can be concluded from the literature review conducted, there are serious issues in the maintenance of water sustainability by the world. However, there is tremendous advancements in the field of communications and information technologies, which can help in analyzing and capturing the data at a large scale. Cyber-Physical systems can help in various decision making processes with respect to threats on water security and many other challenges and issues related to water security. These systems can also help in predicting the climate change and its impact, forecasting of the floods and also helps in determining the presence and utilization of ground and surface water.

In order to face the challenges from water sustainability effectively there needs to be an intensive multi-disciplinary research. Given the complexity of these issues researchers must communicate with water sector professionals for better decision making. The integration of water sector professionals and the best scientific technologies can help in making better tools for effective water management.

2.7 Historic Developments of the Flushing System

From the beginning of history sanitation is the main concern with the humans. Urbanization resulted in possible infectious ills due to improper managing of human waste. Examples include cholera epidemic in London 1854. The management of human waste was started first during Mesopotamian civilization leading to the formation of toilets. Following this, many improvements have been made. The development have reached till the point of utilizing pressurized water for flushing which is a remarkable development.

In the following, historic development of the flushing systems will be presented followed by detailed account of pressurized flushing systems.

John Harington is the actual inventor of modern flush toilet. Starting with Ajax by John Harington it reached up to Flushometer by William Sloan. Following is a brief history of development of the flushing system.

'The Metamorphosis of Ajax' was published in 1594, by John Harington in which he described about a flush toilet installed in his house [27]. In that design, it uses a flush valve to push the water through the outlet of the tank. Another design is a wash-down design for emptying the bowl. Unfortunately, this Ajax was not adopted in England, but it was adopted in France.

With the advent of Industrial Revolution, the flushing system developed into its contemporary form. In 1775 Scottish watchmaker and mechanic Alexander Cumming [28], made a remarkable invention in the field of plumbing by which a pattern for the flush toilet was acquired with all important invention with S-bend in the drainage pipe. Standing water trapped in the S-bend helped in preventing the characteristic foul smell of the drainage. This model consists of a sliding valve above the trap, in the outlet of the bowl. This was a great step forward. Alexander Cummings was thus awarded the first patent for the flushing toilet. But unfortunately feces tend to get trapped in the sliding valve thereby clogging occurs.

In 1775, Joseph Bramah found that the current model could freeze the waste during peak winters. This was corrected by substituting the slide valve by a self- cleaning hinged flap which is watertight at the lower end in the design. This model needs to be connected to water source to maintain sanitation conditions to seal off the water. For the flush tank a float valve system was also developed by him. This then became the first practical flush toilet design. However, it was noisy and the flushing action failed often. In addition, it was found that the seal would dry up whether the toilet was not used often used.

The popularity of the flush toilets then drastically increased. The main reason for the popularity of flush toilet increases is due to the substantial growth in the sewage system. Also, the popularity of flush toilets increases due to the health and sanitation reasons.

The single piece of ceramic flush toilet was proposed and invented by Thomas William Twyford. For sanitary industry, 1870s are considered as the peak and remarkable period. He released his first sanitary ware catalogue in 1879. His first water closets (the National, the Crown and the Alliance) were not free-standing. About 1884 he introduced the first freestanding closet, the Unitas, which were more hygienic than the existing ones. In 1884 he achieved patent for a ceramic baffle near the flush inlet to assist in the distribution of water around the basin. In 1886 a second type of freestanding closet with exterior styling called Florentine. About 1887 he introduced earthenware hand basins made with improved holes for the taps, integrated overflow chambers, and facias designed to hide the iron brackets supporting the basin. He also introduced a pedestal bidet with hot and cold taps, standing waste, overflow, and flushing rim.

Then in 1888, Thomas William Twyford applied for the patent protection for his invention. In 1885 Frederick Humpherson of England demonstrated the modern pedestal flush-down toilet.

In order to empty the tank, Thomas Crapper commercialized the siphon system in 1885. The floating valve was replaced by him as it caused a lot of leakage. He applied 9 patents, three of which were regarding the enhancements of water closets such as floating ballcock. Actually Albert Giblin who received a patent for siphon discharge system designed the flush toilets. The flushing system was invented in Britain, however this technology went around very soon.

William Elvis Sloan was the inventor of Flushometer in 1906.The Flushometer uses the pressurized water which is taken from the water line directly. Even today many public restrooms are equipped with these kinds of Flushometer.

In 1907 Thomas MacAvity Stewart develops a self-cleaning vortex-flushing toilet bowl. Philip Haas invented the flush rim in 1911. The flush rim has many outlets of water in the shape of a ring which resembles the modern day toilet.

2.8 Flushmate™

Flushmate™ pressure assist technology is a flushing system that offers high performance and lower consumption of water. This system is capable to harness the pressure from the water supply line to provide energy needed for complete flush. Flushmate™ pressure assist technology cannot be retrofitted into a standard gravity fed toilet.

2.8.1 Operating Principle of Flushmate™

When Flushmate™ is empty, it contains air at atmospheric pressure. As Flushmate™ vessel gets filled with water, it uses the supply line pressure of the water to compress the air trapped inside the vessel. This compressed air increases the potential energy of the entire system. When toilet is flushed, this energy pushes the water into a specifically designed toilet bowl. Thus instead of the siphon action of a gravity toilet, the pressure-assist toilet helps in pushing waste out of the bowl. This vigorous flushing action helps in cleaning the bowl much better than gravity units. This vessel is also called the Sloan™ (trade mark) pressure vessel [29]. Figure 2-3 below illustrates the working of Flushmate™ equipped toilet flushing:

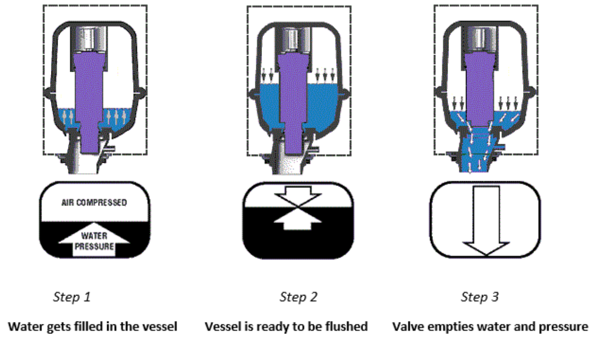

Figure 2- 3 Operational stages of Flushmate™ [8].

The graph shown in Figure 2-4 shows as the volume of the water decreases, drain line carry capability decreases for those toilets unequipped with Flushometer tanks [29].

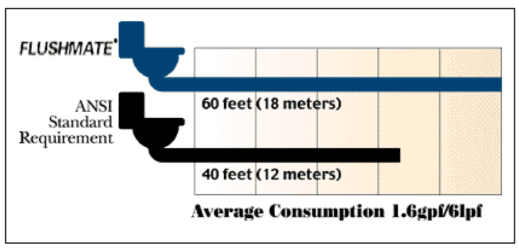

Figure 2- 4 Flushmate™ Drain Carry performance [8].

Flushmate™ equipped bowls are designed hydraulically as in Figure 2-5 to push the contents out of the trapway instead of gravity siphoning. The need for double flushing is eliminated in Flushmate™ equipped water closets because of the trapway design that has fewer bends.

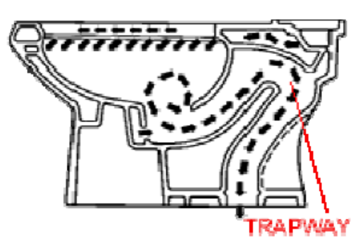

Figure 2- 5 Flushmate™ Bowl Design.

2.8.2 Flushmate™ Curve

Flushmate™ curve in Figure 2-6 below shows a comparison between Flushmate™ equipped toilets and gravity siphoning toilets [29].

There are many benefits while using Sloan pressure vessel in performance such as:

- Water delivery is maximized at the start of flush cycle in order to produce maximum powerful pushing action.
- Rest of the water forms large water surface area towards the end of cycle.
- Entire cycle is about only 5 seconds while in the other for about 12 seconds.

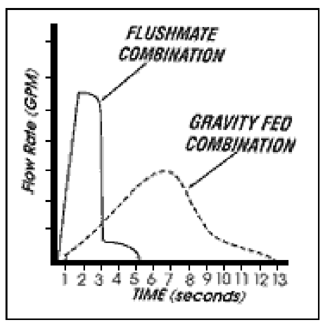

Figure 2- 6 Flushmate™ Curve [8].

2.9 Water Sense Label

Water Sense is a partnership program by U.S Environmental Protection Agency. The program aims to reduce the water wastage by using water efficient products. Water Sense helps to promote the value of water efficiency and helps to reduce water use, thereby reducing the strain on water resources. Only those products and services that are certified to be at least 20% more water efficient without compromising the performance and quality will earn Water Sense Label. With the advancements in the design, modern toilets use 1.28 gallons (4.84 liters) per flush or less with superior performance and quality. This is about 20% less water than the standard limit. By replacing it with water sense labeled toilets an average family can reduce the water usage by 20 to 60 percent [30].

2.10 Dual Flushing Technique

One of the primary challenges extracted from the issue of climate change and affected the entire world is of availability and sustainability of fresh water. Due to the climate change the most affected areas in term of availability of fresh water would be the arid regions which are now facing shortage of water. Similarly the issue of water inequality is also on rise leading to the conflict between people living in upstream and downstream region [31]. Such issues lead to the conflicts between the neighboring countries. In the wake of these evolving challenges, excessive and unsustainable usage of water at the expense of basic needs of water is a problem which needs to be addressed on priority basis [32].

According to Gleick, [33] a bare minimum requirement of water for an individual is 13.2 gallons per day and in contracts to that every American uses about 98 gallon of water per day. Inskeep, [34] estimated that if the number of flushes per house would be decreased, it could save 7% of indoor water use. There had been many articles on flushing behaviors and identified barriers to conserve water via reduced flushing but most of them revealed similar results that flushing behavior has been a natural habit regardless of purpose of using the toilet [35].

The other perspective regarding flush is the quantity of water used at every time flush button is pressed. The standard flush is of two types and that heavy flush and low flush. The standard quantities of water for heavy flushes are between 3.5 to 4.5 liters along with pressure $1.0 \pm .5$ bars [36]. As higher the pressure lower would be the usage of water the standard was set to make a water tank above the toilet seat to maintain the pressure and provide enough water on pressing the flush button that the cleansing has been done satisfactorily. Similarly, in the case of light flush, the standard set is 1.5 liter of water. The light flush uses the pressure produced by the gravity pulling the water out helping to get rid of the waste. However, the flush system standards had been revised in consultation with global authorities interested in reducing the usage of water [37].

In this case the standards of water in flush have been reduced to 3 liters per flush keeping in view the variation in pressure of water in the tank. In our case the pressure of water has been raised to 8 bars which are 700% higher than the previous standard. This has been done to reduce the quantity of water being used in each flush. This has been known through the survey that people who are in the habit of flush are more concerned about hygiene and clean toilet seat [31]. They are not concerned with the quantity of water being used per flush. The quantity of water and the time duration the flush operates is not of any concern for the user instead he intends to see the waste gone and the seat is clean. It is also derived from this discussion that the tank size and pressure required for the flush are inversely proportional.

$$Q = 1/P$$

As the pressure increases the flow of water from tank to the seat also increases thus there could be incidents that the waste in the seat could not get cleaned through single flush. This pushes an individual to reuse the flash to have satisfactorily results in terms of cleanliness. However, if the pressure of water to be flown through the tank is increased in form of placing the tank at height, there are fair possibilities that the seat would be washed properly. In this regard, keeping in view the long term sustainability of water, the standards of toilets had been revised thus option for the flush has also been considered. The saving of 0.8 liter of water per flush reduces some of the pressure on climate and contribution from America [38]. As we say that every American household uses 13 gallon of water in flush every day, if we calculate the quantity of water being saved through reduction of standard from 3.8 to 3 is given below:

13 gallons = 13*4.546875 = 59.109375 liters

59.109375 / 3.8 = 15.55 flushes per day per household

15.55*.8= 12.444 liters per day per household are saved

12.444*365 = 4,542 liters per year per household are saved

This shows that there is a need to reduce the quantity of water being used in each flush; however it is also essential that the cleanliness is not compromised [37]. Therefore, when the pressure of flow of water is taken to the level of 8 bars there would be very few chances that it would leave any waste behind in the toilet seat.

Another technique used in flush is to make the tanks of 1.5 liter which would flow at extremely low pressure allowing the water with enough time to clean the toilet seat. However, maintaining the pressure of 8 bars in light flush it is suggested to use the quantity of water upto 1 liter making the less water clean the seat with the high pressure water. This system uses the combination of air and water both to increase force within the water and clean the waste properly [39].

According to UN's sustainability development goals targeted to be achieved by 2020 have one component to conserve the usage of water by changing the habits regarding usage of water. The water management is need of the time but it is required to be done without compromising on hygiene. Therefore, pressurized flushing system would allow the households to not only keep the toilet seats clean but also streamline the usage of water through flush [40]. The increased pressure would disseminate the water in speed enabling it to create force on the waste on the toilet seat. Almost all developed nations are working over it to make the usage of water through flush optimize and bring it upto the essential levels.

CHAPTER 3

PHYSICAL SYSTEM DESIGN

In this section, the design of the dual pressurized flushing system is presented. The hardware components used in building the prototype are also described.

The thesis proposes an idea of using an air compressor to pressurize the water and a high pressure pump, to pump water into the tank. This idea is implemented using a physical model for a single user toilet and simulating the same design to implement it in a public place with 12 users.

The Figure 3-1, shows that the air compressor and water pump are connected to the pressurized tank as inlets, which in turn is connected to the toilet at the outlet.

Figure 3- 1 Model of the proposed system.

The air to water ratio in the pressurized tank is maintained at 4:1 and the compressed air is pressurized to 8 bar pressure within the tank. The user will also have an option of selecting the type of flush, depending on light or heavy flushing as shown in Figure 3-2. This is essential as there is a considerable amount of water wasted when the light and heavy waste consume same amount of water for flushing. The light flushing flushes 1 liter of water and the heavy flushing flushes around 3 liter of water. As most users require the light flush more than the heavy flush, this dual flushing system saves considerable amount of water.

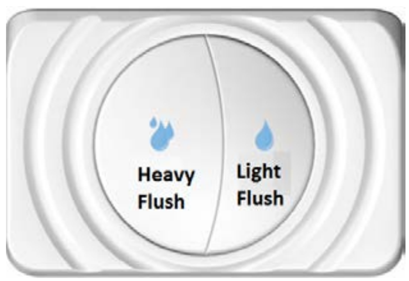

Figure 3- 2 General User Interface for Light or Heavy Flushing

3.1 Design

In order to build the physical system, a pressurized tank is used. The volume of the physical tank is 20 liters. If the ratio of air to water needs to be maintained at 4:1, then the water in the tank has to be 4 liters, the rest of the tank has to be filled with compressed air. The volume of water in the tank is measured with the help of a level sensor. A level sensor is used as the water flows out of the tank with a non-steady flow and a very high speed, and using a flowmeter will not be ideal. The pressure in the tank is measured with the help of a pressure sensor. Different solenoid valves are used to connect the air compressor and the water pump to the pressurized tank. The valve used to flush the water from the tank is also a solenoid valve. The Arduino controls the opening and closing of the valves, working of the air compressor and the water pump with the help of the readings obtained from the level sensor and the pressure sensor. Figure 3-3 shows the picture of the built prototype.

Figure 3- 3 The Built Prototype

The images of the built prototype, the air compressor, the water pump and the electronic circuit can be found in Appendix A.

While building the physical prototype, the characteristic ability of the system is also tested. This is done by using different pressures with in the tank and flushing the water at varying time intervals. This experiment helps in generating the flows at different pressures and for different time intervals. The pressures used in this experiment are 4 bar, 6 bar and 8 bar.

In building the prototype of the flushing system, the air compressor is switched on to pressurize the tank just once. Once there is a flush of water from the tank, the pressure of the air and the volume of water within the tank reduces. However, the water pump, pumps in water into the tank, thereby increasing the volume of water in the tank and increasing the pressure in the tank as the air which had expanded, compresses again. The purpose of the air compressor is to pressurize the tank just once. After which each time there is a flush the water pump, pumps water into the tank and increases the pressure within the tank to approximately 8 bar.

This light and heavy flushing feature is also a part of this prototype. The user can select the amount of water which is required to be flushed. If the user requests for a 3 liter flush then the flushing valve is open for a certain time and if the user requests for a 1 liter flush then the flushing valve is open for a shorter duration of time. This time interval is derived from the characteristic ability of the system.

Even if there is a 1 liter flush the high pressure pumps in water into the tank, to achieve the initial 8 bar pressure. At any given time there has to be water in the tank, to prevent the pressurized air from escaping through the flush valve. However, in case the pressure in tank needs to be reduced there is a valve connected to the tank. When the valve is turned on the pressurized air within the tank escapes, this reduces the pressure in the tank and can be monitored with the help of the pressure sensor.

3.2 Arduino

Arduino is an electronics prototyping platform that is open source, simple to learn and use. Arduino is a tool, which allows the computers to sense and control the physical world [41]. It is also the name of the company the designs and manufactures the microcontroller board and kits, which can be used to create interactive environments. Compared to the existing microcontroller platforms in the market; Arduino is inexpensive, operating system independent, open source, extensible, simple and easy to program.

The platform consists of both software and hardware. The hardware consists of a simple microcontroller board and the software is an integrated development environment that can be used to program the microcontroller. A typical Arduino board contains an ATMEL microcontroller and other components that enable programming and addition of more circuits. The microcontroller is based on Atmel's ATMEGA168 and ATMEGA8 microcontrollers. It is preprogrammed with a boot loader that makes it easy to upload program directly onto the on-chip flash memory. These Shields extend the functionality of the board such as wireless connectivity, GPS, LCD or motor control. This board can be connected to various analog and digital sensors depending on the category of the Arduino used.

The Arduino software consists of an integrated development environment (IDE) and core libraries. The IDE is derived from the Processing language development environment and is written in JAVA. But the core libraries are written in C/C++

language.

Appendix B and C contain the codes used to program the Arduino microcontrollers. The code in Appendix B is used to test the scalability of the system and the code in Appendix C is used to build the 8 bar flushing system

3.3 Arduino Uno

Arduino Uno Board: Arduino Uno board is a microcontroller board based on Atmega328. It has 14 digital input/output pins of which 6 are pwm pins, 6 analog channels, 16MHz crystal oscillator, USB connection, power jack, an ICSP header and a reset button. Atmega328 has 32KB of flash memory for storing code. The analog pins are used for the analog sensors used in the experiment. It also has 2KB of SRAM and 1KB of EEPROM [42]. Figure 3-4 shows an Arduino Uno board.

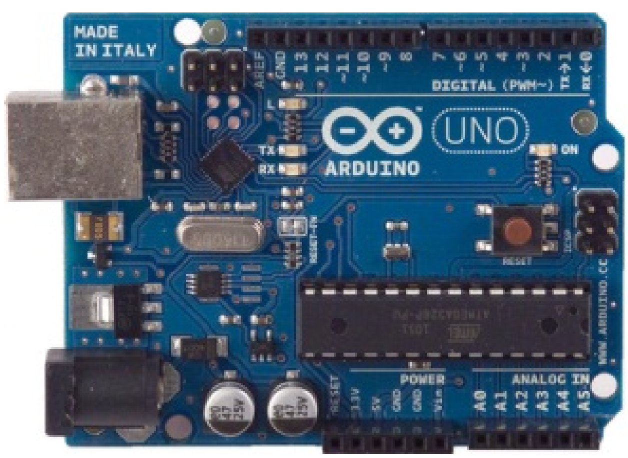

Figure 3- 4 Arduino Uno Board [42].

The specifications of the Arduino Uno Microcontroller are as shown in the table 3-1 given below.

Table 3- 1 Specification of Arduino Uno [42]

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P)
	of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz

3.4 Relay Board

The solenoid valves are operated using electrically operated switch called Relay. They are used in many applications that require control switching. They are known for their reliability, long life and simplicity. An electromagnet is used in the relay to mechanically operate the switch. Whenever the applied voltage or current reaches a certain value, the switch activated either to close or open the contacts.

The 8-channel relay board manufactured by Sainsmart shown in Figure 3-5 is used in the prototype. A 15-20mA driver current and 5V is needed to operate the relay. The relay can handle voltages up to 250V AC and 30V DC at 10A. The board has opto-isolated inputs by which it can be safely controlled by a microcontroller.

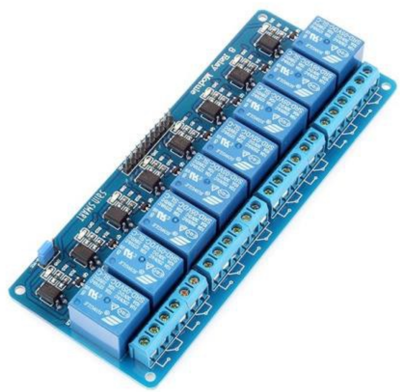

Figure 3- 5 8-Channel Relay Board [43].

3.5 Level Sensor

The volume in the tank cannot be determined by a flowmeter as the flow is at a very high speed and for only a few milliseconds. The flow is a non-steady flow and using a flowmeter to measure this flow would be unrealistic. Hence, the system uses a level sensor placed in the tank to calculate the volume of the water in the tank.

A Liquid Level Sensor from Milone Technologies is a great way to measure the level of a liquid [44]. This level sensor can be used in a high pressure environment and can hence be placed in the pressurized tank. The eTape Liquid Level Sensor is a solid-state sensor with a resistive output that varies with the level of the fluid. It does away with clunky mechanical floats, and easily interfaces with electronic control systems. The eTape sensor's envelope is compressed by the hydrostatic pressure of the fluid in which it is immersed. This results in a change in resistance that corresponds to the distance from the top of the sensor to the surface of the fluid. The sensor's resistive output is inversely proportional to the height of the liquid: the lower the liquid level, the higher the output resistance; the higher the liquid level, the lower the output resistance. Figure 3-6 shows the liquid level sensor used in the thesis.

Figure 3- 6 Liquid Level Sensor [45].

The technical details [46] of the level sensor are as follows:

Sensor Length: 14.1" (358 mm)

Thickness: 0.015" (0.208 mm)

Resistance Gradient: 150Ω / inch (59Ω / cm), ± 10%

Active Sensor Length: 12.6" (320.7 mm)

Sensor Output: 2250Ω empty, 400Ω full, ± 10%

Resolution: 0.01 inch (0.25 mm)

Temperature Range: 15°F - 140°F (-9°C - 60°C)

3.6 Pressure Sensor

To check the pressure within the tank we use a pressure sensor designed by Pace Scientific Data Loggers and Sensors [47]. This pressure sensor can be used for rugged industrial applications and has IP67 Waterproof and Weatherproof housing. It has various ranges from 0-30 psig to 0-20000 psig. It has a stainless steel pressure port and can be used to calculate the pressure in different environments. The model we use is P1600-Vacuum – 150 psig. Figure 3-7 shows the pressure sensor for models in the range of P1600.

Figure 3- 7 Pressure Sensor [47].

3.7 Air Compressor

In order to send the compressed air in the tank and to pressurize the tank, an air compressor is used. The air compressor used is shown in the following Figure 3-8. This air compressor has an oil lubricated piston compressor, along with a full cast iron cylinder. It has large wheels which can heighten performance. It has Pressure regulator, plastic shroud covers hot and rotating parts. It also has 2 pressure gauges. This air compressor works with 230 Volts/50 Hz supply. The maximum working pressure is 8 bar. It can displace air at 200lt/min. This air compressor also works with automatic start-stop system.

Figure 3- 8 Air Compressor [48].

3.8 Water Pump

A high pressure water pump is used to pump water into the tank. Since the pressure in the tank is around 8 bar the water has to be pumped into the tank at a pressure greater than 8 bar. An aquatec 220 psi Triplex Diaphragm pressure switched Bypass Pump is used in this thesis [49]. This pump, pumps in water at a pressure of 15 bar. It uses 115 Volts and uses a diaphragm style of pumping. The Figure 3-9 shows the pump used in the thesis.

Figure 3- 9 Water Pump [49].

3.9 Electrical Feed Through

The level sensor is placed inside the pressurized tank. The wires from the level sensor needs to be connected to the Arduino, for it to receive the sensor values. Since, the wires need to be connected from within the pressurized tank to the Arduino, without creating any leaks from the pressurized tank, an electrical feed through is used. The wires pass through the electrical feed through, which is fixed to the top of the pressurized tank. The electrical feed through is used to send the sensor values from within the tank to the Arduino without creating any leaks from the tank. A feedthrough acts as a conductor which helps in carrying a signal through an enclosure. There are two types of feedthroughs, which are Power feedthrough and Instrumentation feedthrough. The power feedthrough is used to carry high current and high voltage. The instrumentation feedthroughs carry low current or low voltage and are basically used to carry electrical signals. An electrical feedthrough connection can withstand high pressure across its length. The feedthrough used is built by PAVE technologies [50]. The PAVE description code for the feedthrough used is PT8-SS-200-9-TZ22-24-24. The Figure 3-10 shows the electrical feedthrough.

Figure 3- 10 Electrical Feed Through.

3.10 Solenoid Valve

Solenoid valve is electromechanically operated relay. Valve is controlled by an electric current through the solenoid. Figure 3-11 below shows the basic working principle of a solenoid valve.

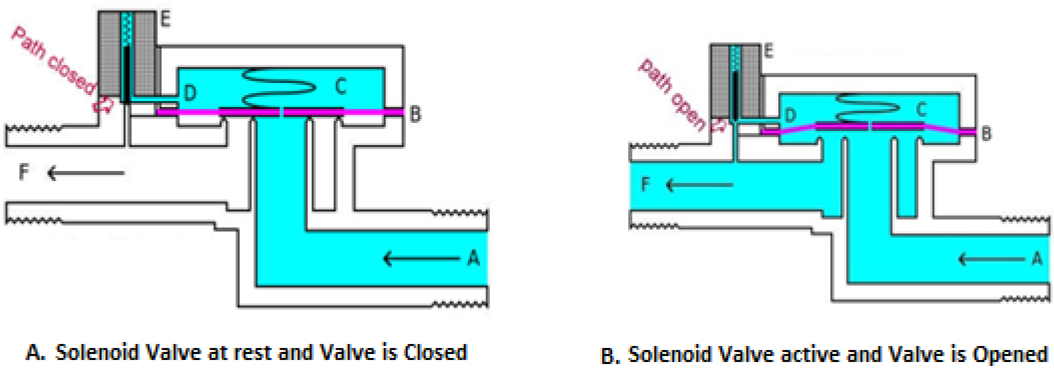

Figure 3- 11 Solenoid Valve at different states.

A solenoid valve is the combination of a solenoid and a mechanical valve. The two main parts of solenoid valve are electrical solenoid and mechanical valve. Solenoid is a coil of wire used as an electromagnet that converts electrical energy to mechanical energy. When electric current flows through the coil a magnetic field is generated which creates the linear motion. When it comes to solenoid valve, this mechanical energy is used to operate a mechanical valve that is to open, close or to adjust in a position. Here we could use a solenoid valve that is operated at 230V [51]. The valve's body can be made up of various materials such as plastic, stainless steel or brass.

For the purpose of this thesis, both stainless steel and plastic solenoid valves are used shown in Figures 3-12 and 3-13 respectively. But the plastic solenoid valve requires a minimum pressure of 3 PSI (0.02 Mpa) to activate. Whereas the stainless steel solenoid valve does not require any minimum pressure as it does not have a gasket unlike the plastic solenoid valve.

Figure 3- 12 Stainless Steel Solenoid Valve [52].

Figure 3- 13 Plastic Solenoid Valve [53].

The solenoid valves are used to connect the water pump, the air compressor to the pressurized tank. Another solenoid valve is used to release the excess pressure within the tank, when requested by opening up. Also the solenoid valve is used to flush the water out of the pressurized tank. When the water is requested to be flushed the water is flushed out through the solenoid valve.

The stainless steel solenoid valve is a normally closed electric solenoid valve. It has 1 inch ports, a stainless steel body, a viton seal and is operated with a supply of 12 volt DC. The electric solenoid valve is constructed with a high-quality stainless steel body, two-way inlet and outlet ports with one inch (1") female threaded (NPT) connections, and heat and oil resistant Viton gasket. This solenoid valve works in the pressure range of 0 to 10 bar pressure and is best suited for the purpose of this thesis.

The plastic solenoid valve, works with a power supply of 9V or 12V. This solenoid valve has half inch female threaded connections (NPT) and has a working pressure of 0.02 MPa – 0.8 MPa. It is a one directional solenoid valve and is in the normally closed state.

CHAPTER 4

PHYSICAL SYSTEM INTERFACE

4.1 Interfacing of the Level Sensor

The level sensor is fixed to a rod which is fixed to the base of the tank from inside the tank. Only one side of the level sensor is glued to the rod. The rest of the level sensor is freely hanging from the rod. The level sensor is placed such that the lower tip is just above the base of the tank. The Level sensor is connected to the A0 pin of the Arduino board. The level sensor is used in a voltage divider circuit where the A0 pin of the Arduino is considered as the output of the voltage divider circuit. The second and the third pin of the level sensor is used to make the connection. In the below voltage divider circuit, the level sensor is used as R2. R1 is taken as 2.2Kohm in the setup. Vin is connected to +5Volt of the Arduino and Vout is connected to the A0 pin of the Arduino. Figure 4-1 shows the voltage divider circuit which is used in the interfacing of the level sensor.

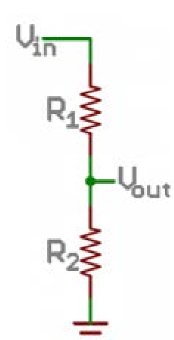

Figure 4- 1 Voltage Divider Circuit.

At A0 pin of the Arduino, we get a sensor value which has to be converted into the respective output voltage. The following equation is used to convert the Sensor Value to the output voltage.

$$Voltage = ((Sensor\ Value)/1023) * 5 \tag{5.1}$$

This Voltage calculated is considered as the Vout in the voltage divider circuit. Given below is the equation for a voltage divider circuit. The above variables such as Vout, Vin, R1 are replaced in the following equation and hence the Resistance (R2) of the level sensor is computed.

$$Vout = Vin * \ \ R2/(R1 + R2) \tag{5.2}$$

The derivation of the equation used in the Arduino program is as follows:

$$((Sensor\ Value)/1023) * 5 = \ Vin * \ \ R2/(R1 + R2)$$

Since, input voltage, Vin =5V,

$$R2 = \ \ R1/((1023/(Sensor\ Value)) - 1) \tag{5.3}$$

This equation 5.3 is used in the Arduino code to convert the sensor value into Resistance. Once we get the resistance of the level sensor we use the graph shown in Figure 4-2, available in the datasheet of the level sensor to convert that resistance into the respective height in inches.

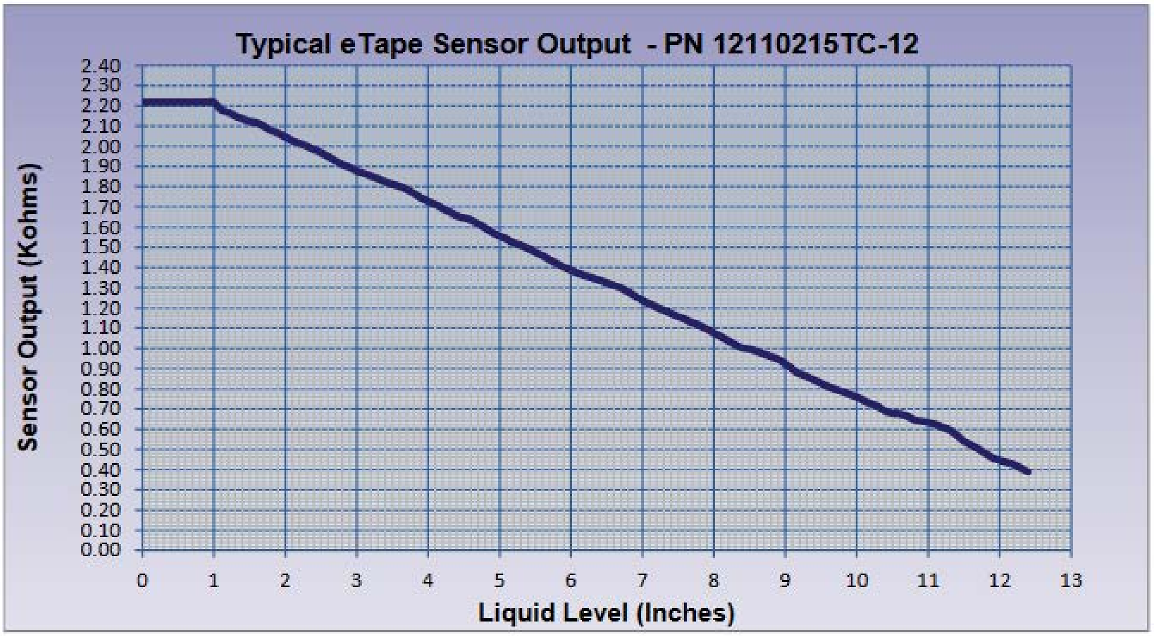

Figure 4- 2 Typical eTape Sensor Output [46].

In the level sensor used in the setup there is a linear shift of 470 ohms in the output resistance of the sensor corresponding to the height. So using the linear nature of the level sensor we calculate the slope and the equation of the line. Such that the line can help us determine the height of the water in the tank when given the resistance of the level sensor pins.

The equation of this line is calculated to be

$$y = -160x + 2810 \tag{5.4}$$

Where, x is height in inches and y is the resistance.

This equation is used in the Arduino program to convert the Resistance into height. Once the height is determined, it is converted into centimeters. If the height is less than 3 centimeters then the Serial Monitor of the Arduino displays “Less than minimum amount of water available", as in the above graph if the height of the water is less than one inch the data is not reliable. If the height of the water is greater than 3 centimeters then the program calculates the volume of the water in the cylindrical tank with the help of the following equation.

Volume of the water in the tank = $\pi r^2 h$

The r here is constant which is 12.75 cm, the radius if the cylindrical tank. And h is the height of the water in the tank.

4.2 Interfacing of the Pressure Sensor

The Pressure Sensor used in the setup is P1600-Vac –150 from PACE Scientific Data Loggers and Sensors. The pressure sensor is fixed to the top of the tank. The sensor needs a 5.0 Volt DC power supply which is supplied from the Arduino. One pin of the pressure sensor is also connected the ground of the Arduino. The output wire is connected to pin A1 of the Arduino. The reading from this pin is firstly converted into voltage using the following equation.

$$Voltage = ((Sensor\ Value)/1023) * 5$$

The voltage signal is scaled to pressure (bar) by multiplying the output voltage of the sensor by 2.84 and then adding -2.43 to the result.

The P1600-vac-150 has an error band of 1% of span, and its span from vac (-14.7) to 150 is 164.7 [54]. Hence when the pressure is less than 1.3 bar on the sensor, the serial monitor displays “Normal Pressure”. When the pressure is slowly increasing in the tank the pressure is continuously monitored and displayed on the serial monitor.

4.3 Need for a Microcontroller

An Arduino is used as the microcontroller to control the working of the system. Arduino has analog pins which help in converting the analog data to the digital data. This microcontroller is available with a built-in analog to digital converter. Depending on the continuously changing level of water and the pressure in the tank, the Arduino controls the working of the physical system. The Arduino also serves the user with the desired level of water to be flushed, which is determined by the user. The system requires a timed process, which is achieved using the software which runs on the microcontroller. If the number of users who are using the pressurized tank is increased, the system can be modified to accommodate more number of users. The scheduling of these users (to determine which user to serve) can also be done with the help of the microcontroller.

4.4 Characterizing the Ability of the Physical System

In order to characterize the ability of the system, the experiment is run several times, with different pressures within the tank and with different time intervals for which the water is flushed out of the system. Firstly, water is filled into the tank with the help of a water pump and a solenoid valve. As the water is filled into the tank, the volume of the water in the tank is monitored and displayed on the serial monitor. Once the water in the tank is 4 L the water pump stops pumping and the solenoid valve connecting it to the tank closes. Then the solenoid valve to the air compressor opens and compressed air is filled in to the tank, which pressurizes the water in the tank. The pressure within the tank is monitored and displayed on the serial monitor with the help of a pressure sensor. Once the pressure within the tank reaches 4, 6 or 8 bar (Depending on which pressure the experiment needs to be run for) the solenoid valve connecting the air compressor and the tank is turned off. Now the Arduino code waits for the flushing button to be pressed. Once pressed, the main solenoid valve which flushes the water out of the system is turned on for different time intervals (in hundreds of milliseconds), during which water from the tank is flushed out of the system. The level sensor now measures the level of the water within the tank and hence can help determine the quantity of the water flushed out. The second button is then pressed, to release the pressure within the tank with the help of a solenoid till the pressure inside the tank becomes normal.

In short, to characterize the ability of the system, the following steps are taken:

- Building the Physical Setup.
- Setting the pressure within the tank to 4 bar pressure and changing the flush duration from 100ms to 700ms in steps of 100 milliseconds and analyzing the amount of water flushed.
- Setting the pressure within the tank to 6 bar pressure and changing the flush duration from 100ms to 700ms in steps of 100 milliseconds and analyzing the amount of water flushed.

- Setting the pressure within the tank to 8 bar pressure and changing the flush duration from 100ms to 700ms in steps of 100 milliseconds and analyzing the amount of water flushed.

The following Figure 4-3 describes the experimental setup of the system.

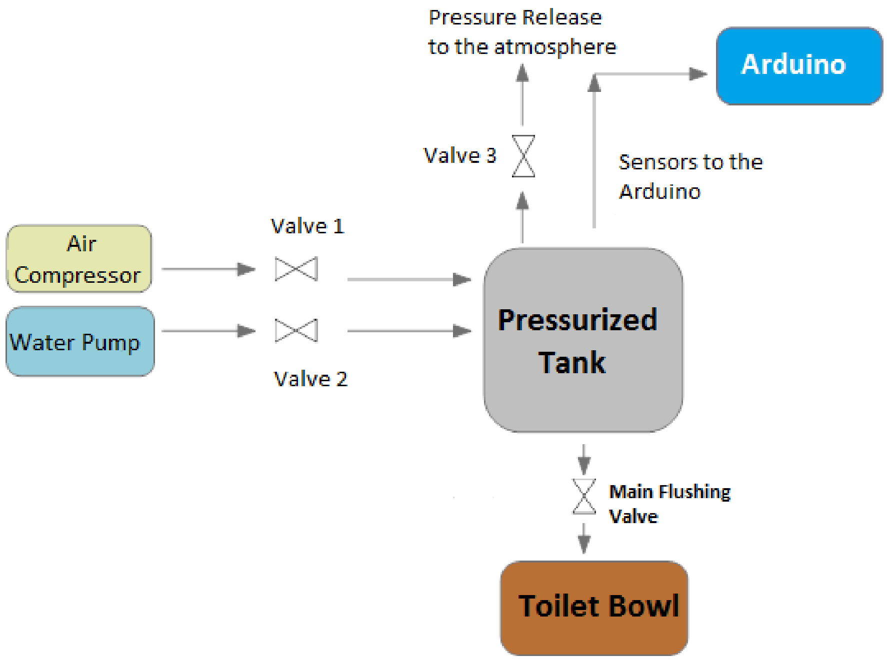

Figure 4- 3 Physical Setup to characterize the ability of the system.

All the solenoid valves are connected to power supply with the help of a relay and an arduino. When the arduino signals the relay, the solenoid valves can be turned on and off. In Figure 4-3,

Solenoid 1 is used to pump in air from the air compressor to the pressurized tank.

Solenoid 2 is used to pump in water from the water pump to the pressurized tank.

Solenoid 3 is used to release the built in pressure to the atmosphere.

The working of the level sensor in the experiment is as follows:

- Approximately 4 liters of water is filled in the tank, at this time the level sensor is dipped to around 8.6 cm, so from the datasheet the resistance of the sensor will be 2298 ohm.
- When water is at its minimum level, the code is designed to display “Less than minimum water available”.
- When resistance of liquid sensor varies from 2670 to 2100 ohms, the voltage at the pin A0 of the Arduino varies from 3V to 2V. This change in signal is used to perform the necessary action.

The working of the pressure sensor in the experiment is as follows:

- When pressure of pressure sensor varies from 1 to 8 bar, it produces the output voltage from 0.83 V to 3.6 V and this signal is applied at the pin A1 pin of the Arduino to perform the necessary action.
- The output voltage is used to calculate the pressure of the sensor and displays the pressure on the serial monitor.

4.4.1 Schematics

In the Arduino Uno Pin 2, 3, 6 and 8 are connected to the solenoid valve of the water pump, solenoid valve of air compressor, the main flushing solenoid valve and to the valve which releases extra pressure when requested. Analog pins A0 is connected to the level sensor and A1 is connected to the pressure sensor. Pin 12 is connected to the button, which is pressed when the main solenoid valve needs to flush the water in the tank. Pin 13 is pressed when the excess pressure needs to be released from within the tank. Figure 4-4 shows the schematics of the Arduino used in this setup.

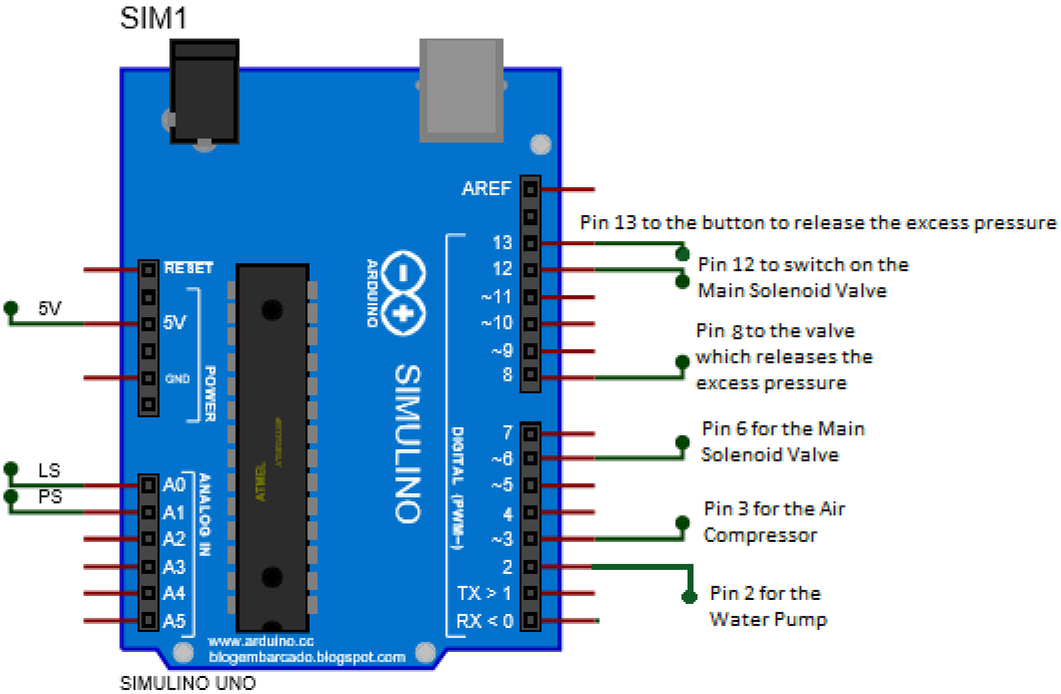

Figure 4- 4 Schematics of the Arduino to characterize ability of the system.

The connections of the level sensor and the pressure sensor are as shown in Figure 4-5, each one has a data pin which is connected to the Arduino.

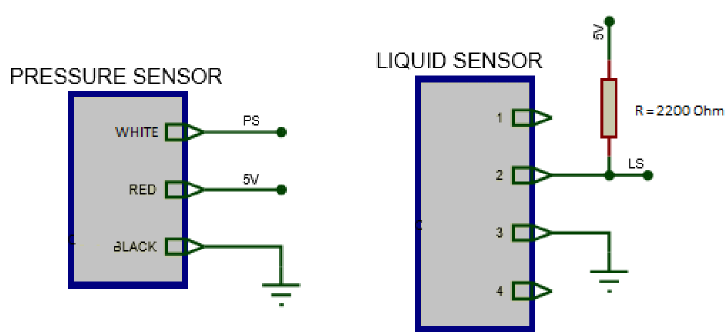

Figure 4- 5 Sensor Wiring.

This setup requires a relay circuit for turning on and off the solenoid valves for flushing, for releasing compressed air into the tank, for releasing excess pressure within the tank and to operate the water pump. The Relay connections to the various elements are shown in Figure 4-6.

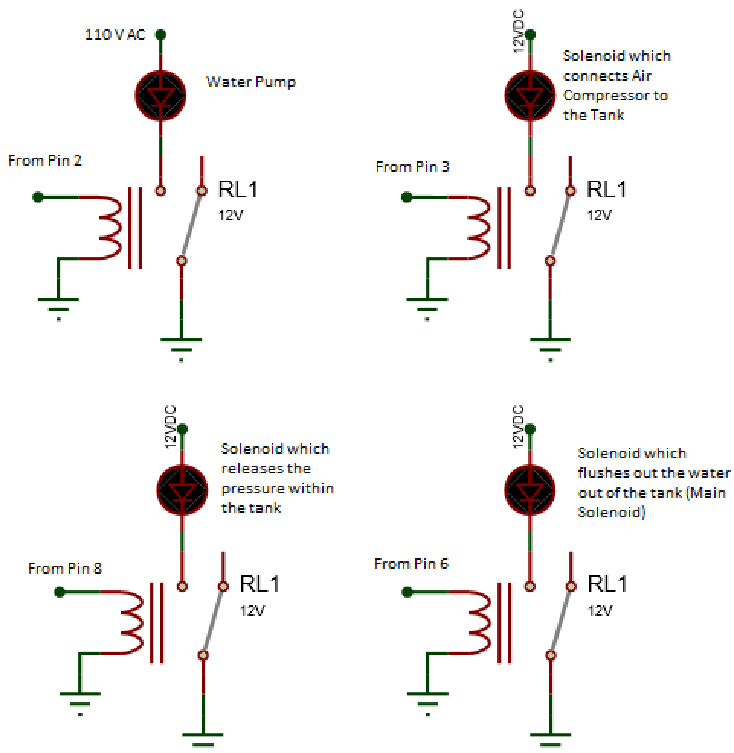

Figure 4- 6 Relay connections from the Arduino to the various components.

There are two buttons used in the experiment. One button is used to flush the water and the other is used to release the excess pressure within the tank. This release in the excess pressure valve is because we run the scalability test by varying the pressure of the air within the tank. The button connected to Pin 12 helps in opening the Main solenoid valve for flushing and the button connected to Pin 13 helps to release the excess pressure from within the tank. The connections of the two buttons are as shown in Figure 4-7.

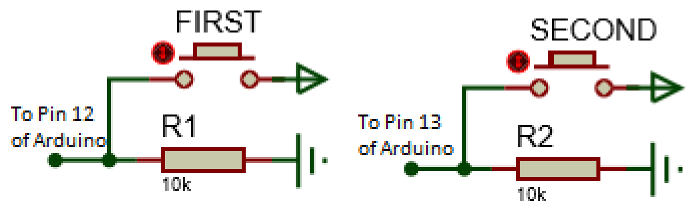

Figure 4- 7 Connections for the flush and to release the pressure.

4.4.2 Pseudo code

The Pseudo code to characterize the ability of the system is as shown below:

Step 1: Fill 4 liters of water in the tank with the help of the pump.

Step 2: Let the air compressor be switched on and let it pressurize the tank up to the desired pressure (4bar, 6bar or 8bar).

Step 3: Wait for the user to request for a Flush.

Step 4: Once the flush is requested, turn on the flush solenoid valve for varying time intervals.

Step 5: Wait for the user to press a switch to release the excess pressure within the tank.

Step 6: Release the excess pressure within the tank.

Step 7: Observe the water level in the tank.

The solenoid valve responsible for flushing, is turned on for varying time intervals in order to determine, the volume of the water which flows out of the tank at varying pressure for varying time. This approach determines the scalability of the system. The pressure within the tank is released as this system is run for varying pressure.

4.4.3 Algorithm

The code is designed to have 6 States (S0, S1, S2, S3, S4 and S5) and the execution of each state occurs when a certain condition is fulfilled. The signals such as S_WATER, S_AIR, S_EP and S_MAIN are used in the designing of the code. The purpose of the signals is shown in the Table 4-1.

Table 4- 1 Signals used in the design

No.	Signal Name	Purpose
1.	S_WATER	Switches On/Off the valve to the water pump
2.	S_AIR	Switches On/Off the valve to the air compressor
3.	S_EP	Switches On/Off the valve to release the pressure within the tank
4.	S_MAIN	Switches On/Off the valve to flush water into the bowl

These signals are used as variables in the Arduino code and its values are sent to the respective pins as shown in the schematic in the Figure 4-4. When these signals are LOW, it indicates that the respective valve or pump is ON and when the signals are high, it indicates that the respective valve or pump are OFF.

The input to the Arduino are as shown in Table 4-2. There are 4 inputs which are sent the Arduino. The inputs from the level sensor and the pressure sensor are feedback inputs which send signals to the Arduino depending on the dynamically changing parameters within the tank. The input to flush the water and release the pressure from within the tank are user inputs and are sent to the Arduino with the help of switches.

Table 4- 2 Inputs to the Arduino for the Scalability Tests

No.	Signal Name	Purpose
1.	LS_IN	Signal from the Level Sensor to the Arduino
2.	PS_IN	Signal from the Pressure Sensor to the Arduino
3.	SW_1	Switch which requests for a Flush
4.	SW_2	Switch which requests for the Pressure in the tank to be released

Table 4-3 gives the description of the various states and their purpose.

Table 4- 3 State Description for the Scalability Test

No.	State	Purpose
1.	S0	Fills Water in the Tank
2.	S1	Pressurizes the Tank
3.	S2	Waits for the user to Request for a Flush
4.	S3	Flushes the Water from the Tank
5.	S4	Waits for the user to Release the Pressure within the tank.
6.	S5	Releases the Pressure from within the Tank

When the code is uploaded in the Arduino, the level sensor checks the level of the water within the tank. If the water in the tank is less than 4 liters then the State S0 is entered wherein the water pump, pumps water into the tank. As the water in the tank reaches the desired level the pump is switched off. The pressure sensor then determines the pressure within the tank. When the pressure within the tank is less than the desired pressure the air compressor valve is turned on which is State S1. The experiment is conducted for 4 bar, 6 bar and 8 bar pressure, this pressure threshold is changed when the pressure for which the experiment is run is changed.

- When the experiment is run for 4 bar pressure the air compressor is turned on till the pressure sensor value reads 463.
- When the experiment is run for 6 bar pressure the air compressor is turned on till the pressure sensor value reads 607.
- When the experiment is run for 8 bar pressure the air compressor is turned on till the pressure sensor value reads 730.

Once the pressure reaches the desired level the code exits State S1. The Arduino then waits for the Button (SW_1) for flushing to be pressed and it enters State S2 where the valve for the air compressor is turned OFF. When the button is pressed the code enters State S3. In this state, the main flushing solenoid valve is turned ON for a fixed interval ranging from 100 milliseconds to 800 milliseconds in steps of 100 milliseconds each time the code is run. This interval is selected because it takes approximately 800 milliseconds for 3 liters of water to be flushed at 8 bar pressure. Once the main solenoid valve is turned on and then turned off the code exits State S3. The code then enters State S4 which waits for the second button (SW_2) to be pressed to release the pressure inside the tank. Once the button is pressed the code enters State S5 and remains in this state till the pressure in the tank reaches normal pressure.

The complete algorithm for the whole process is shown in the flow chart in Figure 4-8.

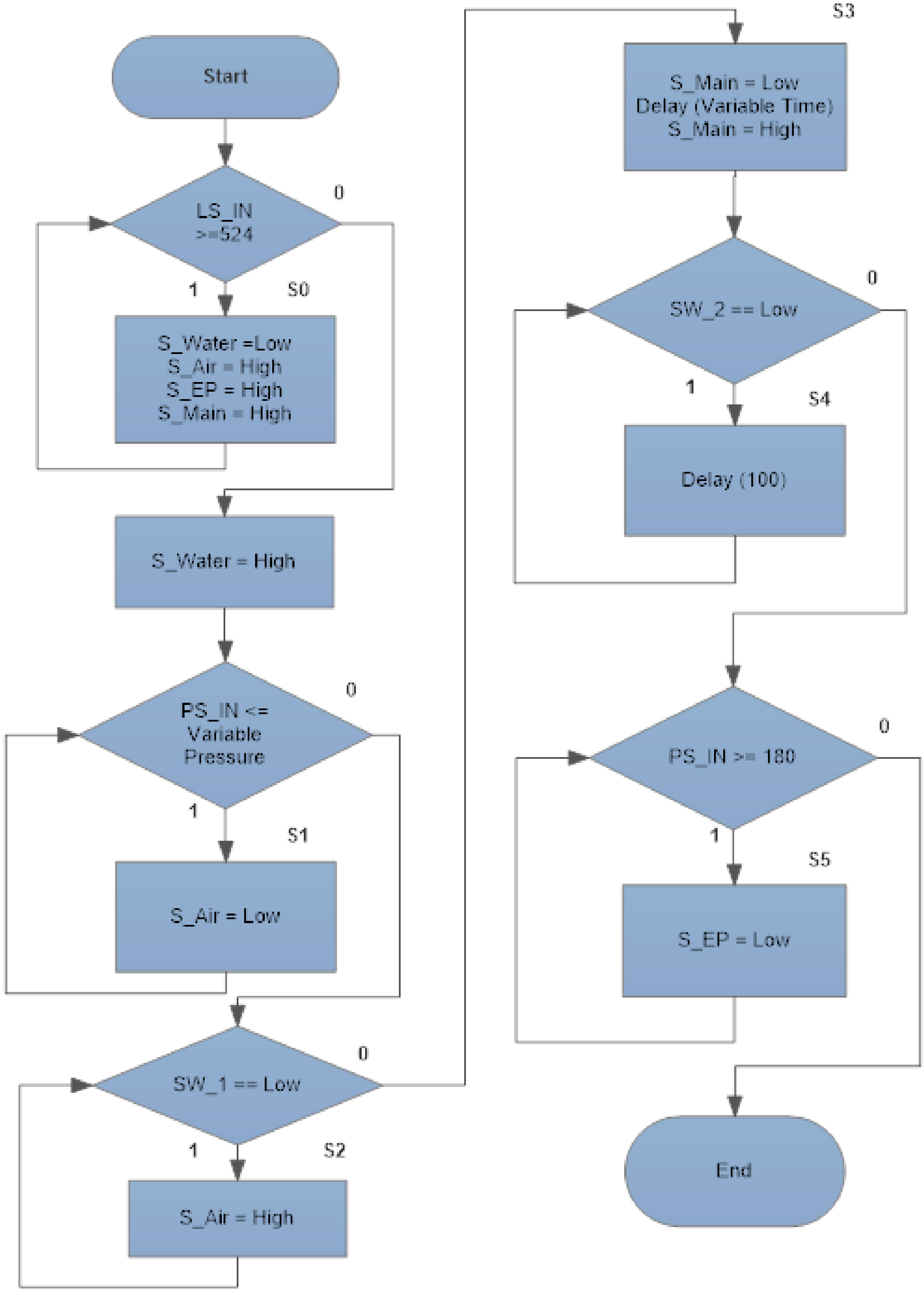

Figure 4- 8 Flowchart to characterize the ability of the system.

4.5 Designing an 8 bar flushing system

In order to build an 8 bar flushing system, we use a pump which can pump in water at a pressure greater than 8 bar. The pump used in this experiment pumps in water at 15 bar pressure. The pump initially pumps water into the tank, till the volume of the water within the tank reaches 4 liters. Then the air compressor pumps in compressed air into the tank through a solenoid valve with 8 bar pressure. After the pressure inside the tank reaches 8 bar, the solenoid valve is turned off, thereby turning off the air compressor. The arduino then waits for the user to press the flush. There are 2 types of flushes available for the user to press, one is heavy flushing and the other is light flushing. The heavy flushing, flushes 3 liter of water and the light flushing flushes 1 liter of water. When the user presses the 3 liter flush the main solenoid valve which flushes the water opens up for 700 milliseconds, which is enough to flush out 3 liters of water. When the user presses the 1 liter flush the main solenoid valve opens for 200 milliseconds, which is enough to flush out 1 liter of water. Once the water is flushed out of the tank, the pressure within the tank reduces. However, after every flush, the water pump again fills water into the tank against the pressure in the tank, till the water within the tank has a volume of 4 liter. Also, when the water within the tank increases the pressure within the tank also becomes almost 8 bar if there is no leakage within the tank. The arduino then waits for the user to press the flush. Each time a flush is requested, the main solenoid valve opens for a certain time interval and the discharged water is pumped in back to the water tank with the help of the high pressure water pump.

The following Figure 4-9 describes the experimental setup of the system.

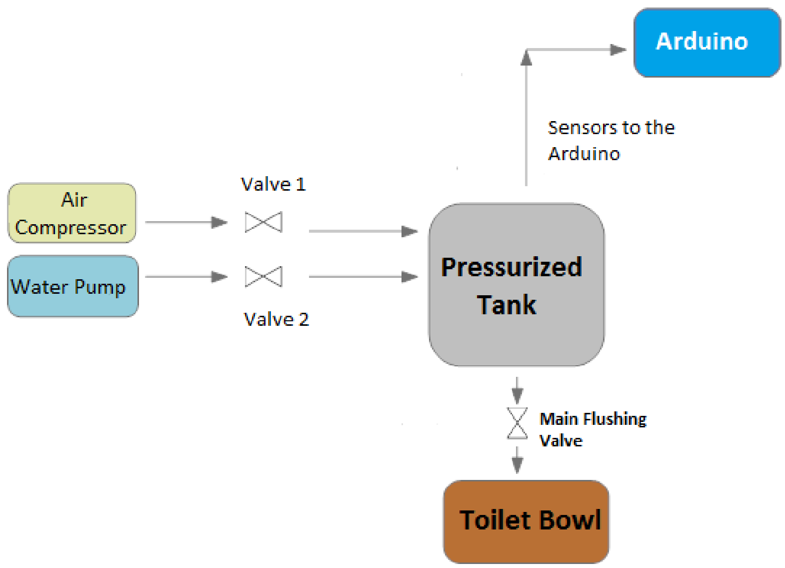

Figure 4- 9 Physical system designed for 8 bar flushing system.

All the solenoid valves are connected to power supply with the help of a relay and an arduino. When the arduino signals the relay, the solenoid valves can be turned on and off.

Solenoid 1 is used to pump in air from the air compressor to the pressurized tank.

Solenoid 2 is connected to the water pump.

The Level sensor and the Pressure Sensor are connected from inside the tank to the Arduino board.

The working of the level sensor in the experiment is as follows:

- Approximately 4 liters of water is filled in the tank, at this time the level sensor is dipped to around 8.6 cm, so from the datasheet the resistance of the sensor will be 2298 ohm.
- When water is at its minimum level, the code is designed to display "Less than minimum water available".
- When resistance of liquid sensor varies from 2670 to 2100 ohms, the voltage at the pin A0 of the Arduino varies from 3V to 2V. This change in sensor value is used to perform the necessary action.

The working of the pressure sensor in the experiment is as follows:

- When pressure of pressure sensor varies from 1 to 8 bar, it produces the output voltage from 0.83 V to 3.6 V and this signal is applied at the pin A1 pin of the Arduino to perform the necessary action.
- The output voltage is used to calculate the pressure of the sensor and displays the pressure on the serial monitor.

4.5.1 Schematics

The Arduino Uno pin 2, pin 3 and pin 6 are connected to the relay of the water pump, relay of the solenoid valve of air compressor and the relay of the main flushing solenoid valve. Analog pins A0 is connected to the level sensor and A1 is connected to the pressure sensor. Pin 12 is connected to the button, which is pressed when the main solenoid valve needs to flush 3 liter of water. Pin 7 is connected to the button, which is pressed when the main solenoid valve needs to flush 1 liter of water. Figure 4-10 shows the schematics of the Arduino for this setup.

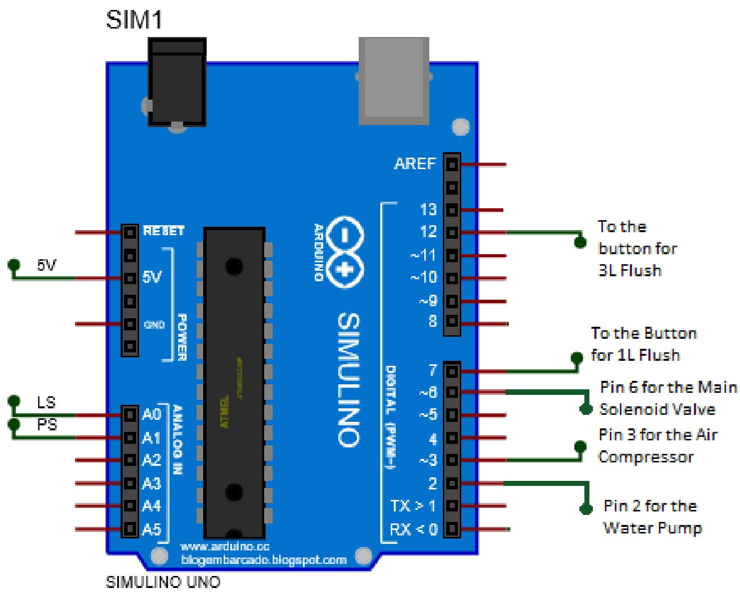

Figure 4- 10 Schematic of the Arduino for the 8 bar flushing system.

The connections of the level sensor and the pressure sensor are as shown in the Figure 4-5 each one has a data pin which is connected to the Arduino. The relay circuit for the water pump, the solenoid valve for the air compressor and the main solenoid valve for flushing is shown in Figure 4-6.

There are two buttons used in the experiment. One button is used to flush 3 liter of water and the other button is used to flush 1 liter of water. The button connected to Pin 12 of the arduino board helps to flush 3 liter of water by opening the Main solenoid valve for flushing for 700ms. The button connected to Pin 7 of the arduino board helps to flush 1 liter of water by opening the Main solenoid valve for flushing for 200ms. The connections of the two buttons are as shown in the Figure 4-7.

4.5.2 Pseudo code

Pseudo code for the flushing system is as follows:

Step 1: Fill 4 Liters of water in the tank with the help of the pump.

Step 2: If this is the first run then, let the air compressor be switched on and let it pressurize the tank to 8 Bar Pressure. Else, Skip this step.

Step 3: Wait for the user to request for 3 liter or 1 liter Flush.

Step 4: Once the flush is requested turn on the flush solenoid valve for a fixed duration such that the water is flushed out of the system depending on the user requirements.

Return to Step 1

The volume of the water and the pressure within the tank is continuously monitored with the help of the Arduino interfaced with the level and the pressure sensors.

4.5.3 Algorithm

The code is designed to have 5 States (S0, S1, S2, S3A and S3B) and the execution of each state occurs when a certain condition is fulfilled. The water pump, the solenoid valve for the air compressor and the solenoid valve for the main flushing are all connected to the relay and are controlled by the signals S_WATER, S_AIR and S_MAIN respectively. These signals are used as variables in the Arduino code and its values are sent to the respective pins as shown in the schematic in the Figure 4-10. When these signals are LOW, it indicates that the respective valve or pump is ON and when the signals are high, it indicates that the respective valve or pump are OFF.

The input to the Arduino are as shown in Table 4-4. There are 4 inputs which are sent the Arduino. The inputs from the level sensor and the pressure sensor are feedback inputs which send signals to the Arduino depending on the dynamically changing parameters within the tank. The input to flush the water and release the pressure from within the tank are user inputs and are sent to the Arduino with the help of switches.

Table 4- 4 Inputs to the Arduino for the Flushing System

No.	Signal Name	Purpose
1.	LS_IN	Signal from the Level Sensor to the Arduino
2.	PS_IN	Signal from the Pressure Sensor to the Arduino
3.	SW_1	Switch which requests for a Heavy Flush
4.	SW_3	Switch which requests for a Light Flush

Table 4-5 gives the function of the 5 states used in the generation of the code.

Table 4- 5 State Description for the Flushing System

No.	State	Purpose
1.	S0	Fills Water in the Tank
2.	S1	Pressurizes the Tank
3.	S2	Waits for the user to Request for a Flush
4.	S3A	Serves the user with a Heavy Flush
5.	S3B	Serves the user with a Light Flush.

When the code is uploaded in the Arduino, the level sensor checks the level of the water within the tank. If the water in the tank is less than 4 liters then the State S0 is entered wherein the water pump, pumps water into the tank. As the water in the tank reaches the desired level the pump is switched off and the state S0 is exited.

The “If” statement then checks if this is the first run of the program. If this is the first run of the program and the pressure within the tank is less than 8 bar pressure then the code enters State S1, where the solenoid valve lets the compressor pressurize the tank up to 8 bar pressure. Once the tank is pressurized to 8 bar, the count of the run of the program is incremented.

If this is not the first run of the program, then the State S1 is skipped and the air compressor is not switched on.

The code then waits for either of the switches to be pressed and enters State S2 wherein it turns off the solenoid valve of the air compressor and waits for a user to press the switch. Once any of the switches are pressed the code exits State S2.

Once any of the switches are pressed the code determines which switch is pressed. If SW_1 (at pin 12 of the arduino) is pressed then the State S3A is executed wherein the main solenoid valve is opened for 700ms. If SW_3 (at pin 7 of the arduino) is pressed then the State S3B is executed wherein the main solenoid valve is opened for 200ms.

After executing either of the states there is a 5 second delay, after which the execution of the above code is repeated.

The complete algorithm is shown in the Figure 4-11 as a flow chart.

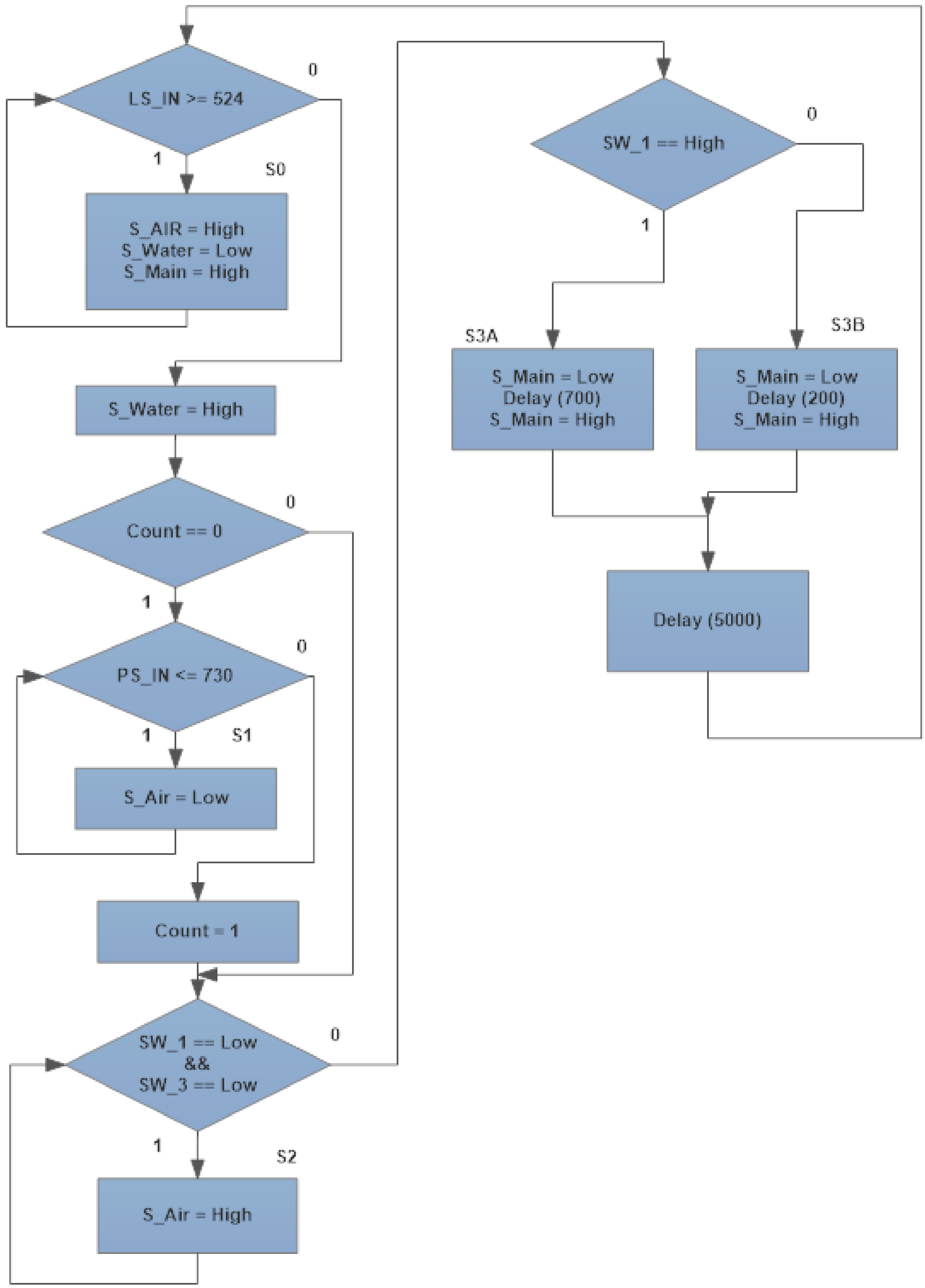

Figure 4- 11 Flowchart for the 8 bar flushing system.

Appendix 4.1 Description of the Behavior of the Scalability Test

This section briefly describes the design of the system which is used for testing the Scalability of the physical system using the state transition table and the state transition diagram. There are four inputs namely, L, P, S1 and S2. Their description is shown in the following table 4-6.

Table 4- 6 Inputs to the System for the Scalability Tests

No.	Signal Name	Purpose
1.	L	Signal from the Level Sensor
2.	P	Signal from the Pressure Sensor
3.	S1	Switch which requests for a Flush
4.	S2	Switch which requests for the Pressure in the tank to be released

For the purpose of designing the system, the inputs are assigned the values ‘0’ or ‘1’ based on the conditions in the following table 4-7.

Table 4- 7 Inputs to the System and their values

No.	Signal	Value ‘0’	Value ‘1’
1.	L	There is no water in the Tank	There is Water in the Tank
2.	P	The Tank is not Pressurized	The Tank is Pressurized
3.	S1	The User doesn’t Request for a Flush	The User Requests for a Flush
4.	S2	The User does not request for the Excess Pressure to be Released	The User requests for the Excess Pressure to be Released

The signals such as S_WATER, S_AIR, S_EP and S_MAIN are used in the design and are considered as the output of the system. Their description as shown in the following Table 4-8.

Table 4- 8 Signals used as the Output of the System

No.	Signal Name	Purpose
1.	S_WATER	Switches On/Off the valve to the water pump
2.	S_AIR	Switches On/Off the valve to the air compressor
3.	S_EP	Switches On/Off the valve to release the pressure within the tank
4.	S_MAIN	Switches On/Off the valve to flush water into the bowl

If the output signal is HIGH, it indicates that the valve is ON and water/air is passing through the valve. If the value of the above signals is LOW, it indicates that the valve is blocked. There are 6 states used in the design of the system. They are S0, S1, S2, S3, S4 and S5. The purpose of these States is as shown in the following Table 4-9.

Table 4- 9 State Name and Description for the Scalability Test

No.	State	Physical Meaning
1.	S0	Water is filling into the Tank.
2.	S1	The Air Compressor is pressurizing the Tank
3.	S2	The Pressurized Tank waits for the User to Press the Flush
4.	S3	Flush Occurs and Water is emptied from the Tank.
5.	S4	The empty Tank waits for the Excess Pressure to be Released
6.	S5	Tank is depressurized.

The system begins with State S0. At this state, the water is filled in the tank. The Tank is not pressurized at this State. The Level Sensor indicates when the water is not filled in the tank until a certain level is reached. Once the Level sensor, senses enough water in the tank, the sensor becomes high and the water pump is turned Off. The system then transitions to State S1. The system stays in this state till the pressure in the tank is less than the desired pressure. The air compressor is turned on in this state. Once the tank is pressurized, the pressure sensor then gives a high input and the air compressor is turned Off and the system transitions into State S2. At this State the system waits for the user to request for the flush. Once the user requests the flush the system transitions to State S3. The system remains in this State till the level sensor indicates that the water has been flushed out. Once the water has been flushed out the System transitions to State S4. In this State the system waits for the User to press the button which requests for the excess pressure to be released. Once the button to release the pressure is pressed the system transitions to State S5. At this state the excess pressure within the tank is released and the once the pressure is below the threshold, the system goes to State S0.

The following design rules are made:

1. The system will default (reboot itself) to the initial state (S0) when it enters an undefined state.
2. The system will also default (reboot itself) to the initial state (S0) whenever there is a mismatch between the input combination and the Current State. For example, When the system is in State S4 and the level sensor input indicates that there is water in the system (L=1).

3. There are three scenarios in each state depending on the current state and the input combination:

   - The system moves to the Next Physically Correct State.
   - The system stays in the same state.
   - The system moves to the Default (initial) state.

Table A- 1 in Appendix D gives the State Transition Table. The 4 inputs are described with various combinations. For each of the combinations the Current State, the Next State and the values of the output signals are described.

Figure 4-12 shows the State Transition Diagram.

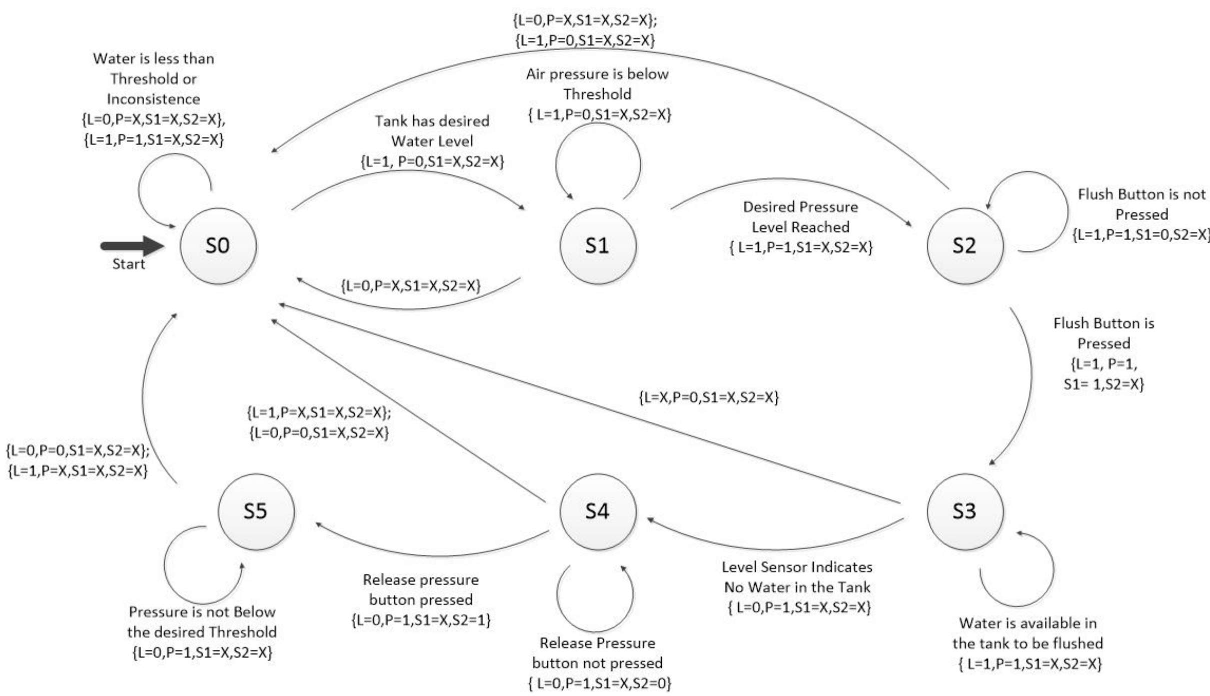

Figure 4- 12 State Transition Diagram

CHAPTER 5

SIMULATION

The simulation of the proposed toilet and its implementation in a public space is done using Simulink and physical modeling tool SimHydraulics with new features introduced in 2014 provided by MATLAB [9]. Simulink is an environment that can be used to design simulation models of multiple domains. Simulink supports modeling and simulation of dynamic systems of multiple domains, automatic code generation, system-level design and continuous test and verification of embedded systems.

It has a user friendly graphical editor, solvers for modeling and simulating dynamic systems and customizable block libraries. It is combined with MATLAB, enabling the users to integrate MATLAB algorithms into models and export the results obtained from simulation to MATLAB for detailed analysis. It contains libraries that help in modeling continuous-time and discrete-time systems. The simulation results can be viewed using scopes and data displays.

SimHydraulics is physical modeling software that gives us different ways to simulate and evaluate hydraulic power and control systems in the Simulink environment. It includes models of hydraulic components, such as pumps, valves, pipelines, actuators, and hydraulic resistances. These components can be used to model the toilet and the tank. The models developed in SimHydraulics can be used to develop control systems and test system-level performance. The models can be parameterized using MATLAB variables and expressions.

5.1 Simulation Overview

This simulation will help in better understanding the behavior of the new approach of toilet flushing system. The simulation will also simulate the water pump and the air compressor. The work on simulation will be done by splitting it into different modules. Each module will have its own results and requirements. Firstly, a model of a simple self-pressurized water tank is built and connected to a user. This will be followed by a building a water tank connected to a water pump and an air compressor. This pressurized tank will then be connected to multiple users, wherein more than one user can flush at the same time. Simulation will also allow different levels of discharge (1L/3L) which may result in variable values of output pressure, bowl volume and total time to discharge the required volume of water. With each flush, the variable values of pressure, interval and volume of water discharged will be discovered using Simulink® software and MATLAB.

The following modules will be built for this simulation:

- The first module will be a model of a simple self-pressurized water tank using Simulink components. This will be applied to a single toilet, allowing two different levels of water discharge (1L and 3L), and answering required variable values of pressure, interval and volume of discharge.
- The second module will consist of a pressurized water tank which will be self-controlled and constrained in a two-phase mixture of air and water which will be proportional in the 4:1 ratio. Two toilets will be connected to this tank and each toilet will allow two different levels of water discharge (1L and 3L). In this model

also, with each flush the variable values of pressure, interval and volume of discharge will be determined.

- The third module will consist of a pressurized water tank which will be self-controlled and constrained in a two-phase mixture of air and water which will be proportional in the 4:1 ratio. Twelve toilets will be connected to this tank and each toilet will allow two different levels of water discharge (1L and 3L). In this model also, with each flush the variable values of pressure, interval and volume of discharge will be determined. Tests of random user behaviors are conducted in this module.

5.2 System Design

The self-pressurized tank, which is provided by SimHydraulics is used in a single-user scenario to help build the system for multiple toilets and validate the results. This self-pressurized tank can provide unrestrained water at a constant pressure which can be pre-defined by the user. The pressure in the tank does not dynamically vary and hence, cannot replicate a real-time scenario. The pressure in the tank needs to vary depending on the volume of water and the air pressure in the tank. Simulink does not have elements to represent an air and water pressurized tank. Each hydraulic system in Simulink can contain only one fluid and not a mixture of fluids. To accomplish an air and water pressurized tank, the SimScape Language is used and a code is designed. The custom element represents the laws of physics and applies them in this situation (Section 5.5). The code for generating this air and water pressurized tank is available in Appendix E.

The simulation model is designed and built to accommodate multiple users in the system. In the 2 user system, both the users are in the system from the beginning of the simulation run. Both the users request for a flush at regular intervals of the simulation time. The volume of the water requested to be flushed (1 liter or 3 liter) by the user is pre-defined. The users can request for a flush at the same time or different times depending on pre-defined inputs to the simulation model which is described in Section 5.6. The simulation model is also programmed to provide a slight delay to the second user in case both the users request for a flush at the same time. This delay can occur by changing the flushing parameter of the simulation model before running the simulation. The 12 user module has the same features as that of the 2 user module in terms of user characteristics.

The events are generated by each flushing request made by a user in the system and when the water in the pressurized tank reaches its maximum and minimum level. When a user requests for a flush, the water from the tank flows into various elbows and pipes before reaching the user. When this water is flushed the volume of water within the tank reduces. As soon as the volume of water in the tank is below the pre-defined minimum level the pump starts pumping in the water, till the water reaches the maximum level.

5.3 Main Packages of the Simulated System

The simulation model is described in this section. In order to build the simulation model as described in Section 5.2, ideal subsystems are created with Simulink components to represent the hydraulic design. One of Simulink´s features is to use packages to encapsulate system parts, improving system understanding and maintenance. The main system will consist of three packages, named User Interaction package, Hydraulic Power package and Toilet package, which can be seen in Figure 5-1. The main packages are common to all the simulation modules built.

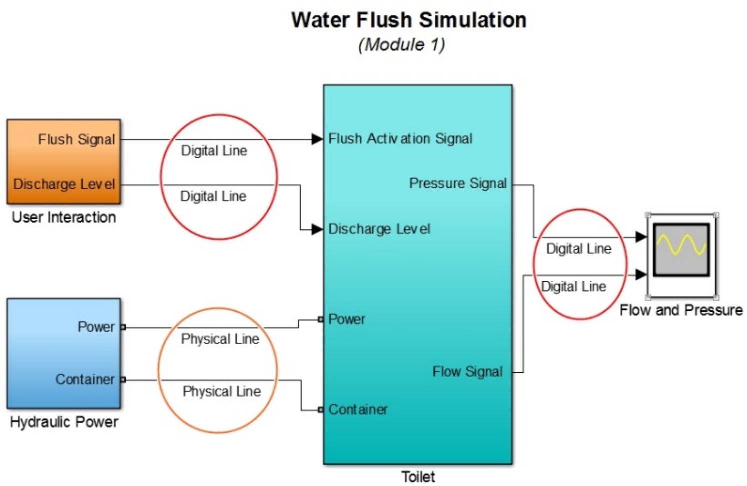

Figure 5- 1 Main Packages.

Figure 5-1 also shows two different lines connecting the different packages. The one circumscribed by red circles represents digital lines, which carry common Simulink signals (also called S). Whereas, the orange circle represent physical lines that carry Physical Signals (also called PS), for instance water passing inside of a tube is a PS. A flow and pressure graph component, responsible to plot the flow and pressure signal variations in time domain is also shown in this module. Each one of the shown packages has its own sub-packages as well. The User Interaction package and the Toilet Package are described in Section 5.3.1 and 5.3.2. The Hydraulic power package for the Self-Pressurized tank is described in Section 5.4.1. The Hydraulic power package for the Air Pressurized tank is described in Section 5.5.

5.3.1 User Interaction Package

The user interaction package makes different types of user decisions, such as the level of water to be discharged (1 liter or 3 liters) and the time the flush is requested. This package is illustrated in Figure 5-2. Red arrow shows the selector which helps in selecting the volume to be discharged.

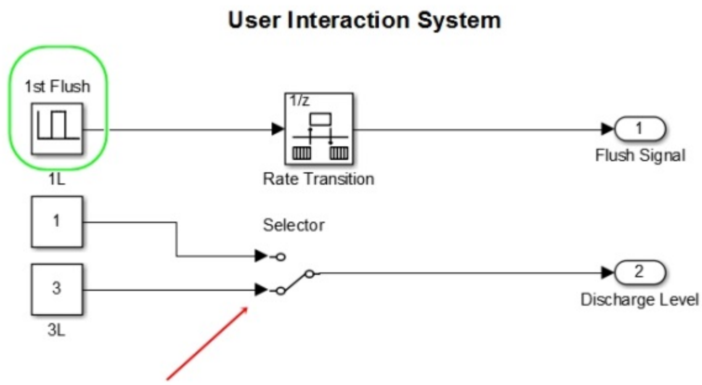

Figure 5- 2 User interaction package.

A green circle focuses on the stair signal. Through this component, the user will decide when to request for the flush. This signal will periodically repeat during the entire simulation time. The flushing signal input is modified for a multi-user environment and is described in Section 5.6. The following Figure 5-3 shows the parameters of this component, which can be adjustable.

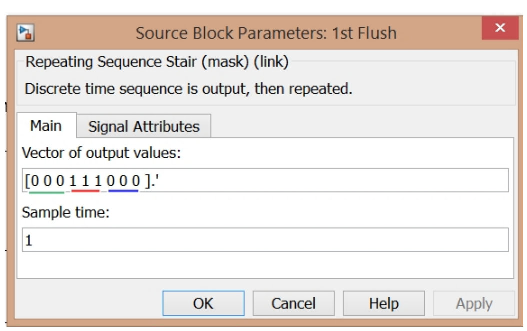

Figure 5- 3 Parameters of the user interaction signal.

In the Figure 5-3, values under-marked with colorful traces are values of a digital vector allowing two possible choices: 1 which implies On, and 0 which implies Off. Green trace shows that in the first three instants (in this simulation, time is measured in seconds, so each instant is equal to 1s of real time), flush valve is not pressed. The next the three instants, marked with the red trace, shows that a valve is in the pressed state and a request for flush is processed. Finally, the blue trace shows, the next three instants are in a relief state.

5.3.2 Toilet Package

The simulation model also consist of a toilet package, where major simulation components may be found. This package contains, three other packages, the control system, hydraulic system and the dataset system (Figure 5-4). The hydraulic system is responsible to model the pipeline structure, using elbows, discharge valve and the pipes. The control system, regulates the discharge level, which implies that as soon as the volume of water requested to be discharged by the user is discharged, the discharge valve is turned off by the control system. Thus, maintaining the specified volume of water flushed. The dataset system will create a set of useful data, which helps in the analysis after the simulation.

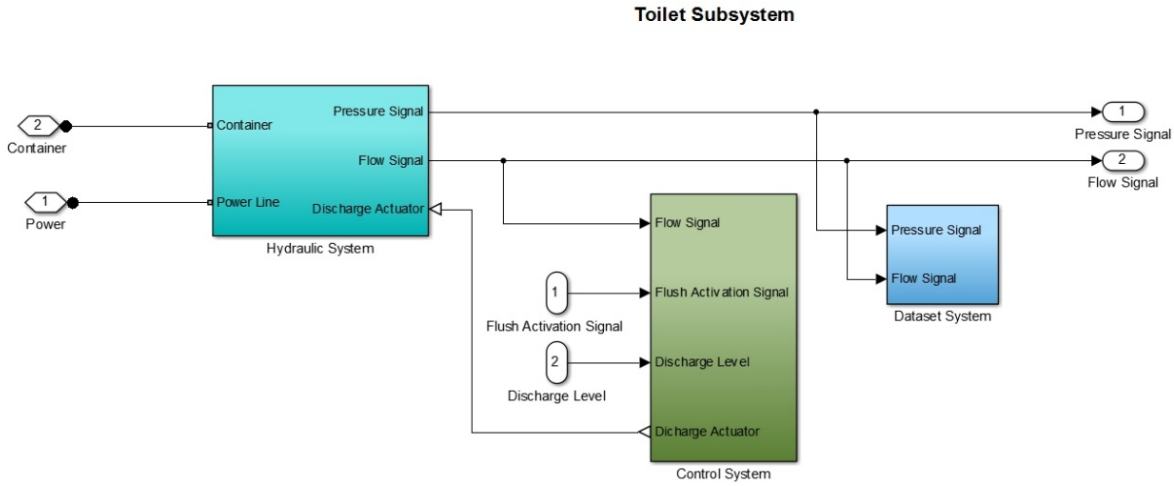

Figure 5- 4 Toilet Subsystem.

The hydraulic system was designed with the values of Table 5-1, (for example, the length of the pipe connecting the 2 elbows is 3 meter or the diameter of the elbow is 1 inch). Hence, there is no cross sectional area variation, across the pipelines, which implies there is no loss of pressure due to the variation in the diameters of the elbows or the pipes. The pressure varies with the variation of the cross sectional area of the restricting elements. This package can be seen in Figure 5-5. A toilet valve is created to regulate the volume of the water flushed, and it is controlled by the logical combination between the user interaction signal and the level of discharge, implemented at the Control Package.

Table 5- 1 Self-pressurized tank specifications

Components	Diameter (in)	Height(m)
Top Elbow	1	—
Bottom Elbow	1	
Pipe	1	3

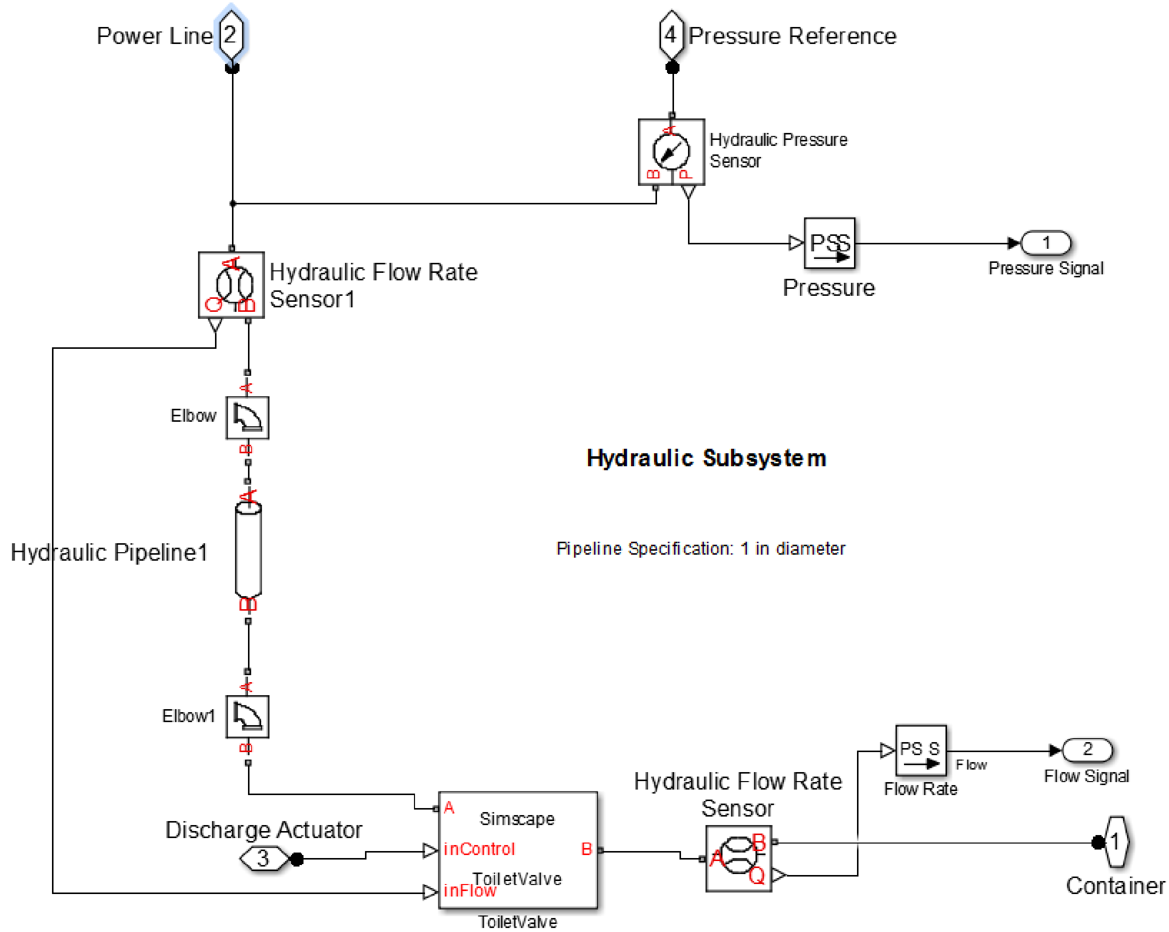

Figure 5- 5 Hydraulic Subsystem.

The Power Line connector allows the water to flow from the pressurized tank to the Hydraulic Subsystem, after the water flows through the circuit elements and is flushed, it will reach the Container collector. Before the water reaches the container it is flushed in the direction of the toilet bowl.

Along this path, a Toilet Valve element was created to help to regulate the flow of water being flushed, this block uses the signal from the discharge actuator. The signal from the discharge actuator, is a signal emitted from the control subsystem and it helps to block the flow of water, as soon as the total volume of water flushed reaches the desired value specified by the user (flush of 1 liter or 3 liter specified in the discharge level of the User Interaction System).

Also, an inflow gate is attached to the Toilet Valve, to be certain that the right amount of water is flushed out from the power line. In the valve, when the discharge actuator is activated, water flowing through the pipeline and the valve is recorded and verified.

The module also uses two other sensors which are, the pressure sensor and the hydraulic flow sensor. These two sensors are placed in the design, to record various data for our analysis and also to generate the pressure and flow signal graphs per toilet. Both the sensors transmit the values through Pressure Signal and Flow Signal gates. The elbow and pipeline parameters are as shown in the Figure 5-6.

The Control package, in Figure 5-7 monitors the flow of the water until the discharge level is reached, after which a signal is sent to unable the discharge actuator. This signal will help the toilet valve to turn off and hence putting an end to the flushing of water. The component, Simulink-PS converter, helps in exchanging signals between the different physical lines and Simulink lines.

The relational operator helps to compare the discharge level requested and the result of the integrator. The integrator, is a component responsible to accumulate the volume of water discharged with the help of the flow rate signal. Once the volume of water discharged is greater than or equal to the volume of water requested, the discharge valve is turned off and the integrator is reset.

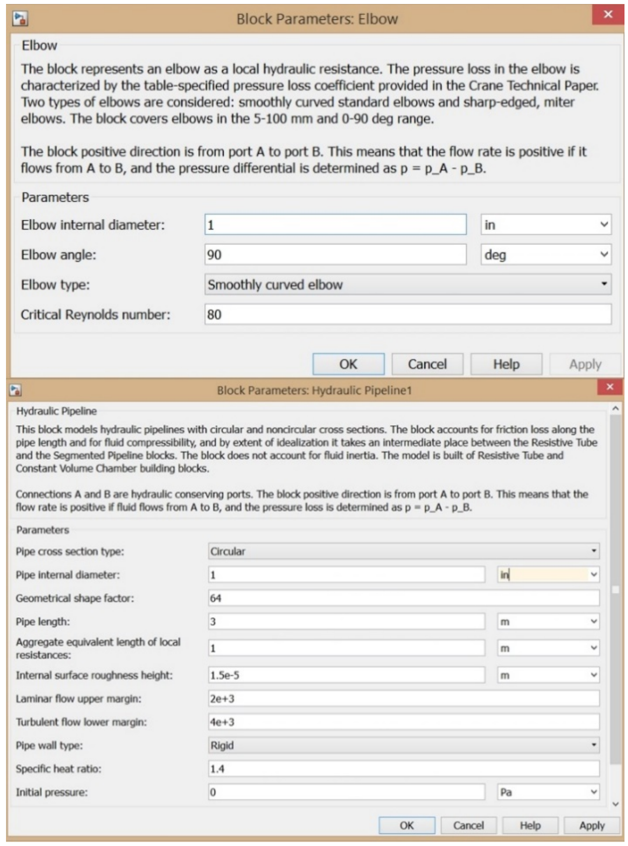

Figure 5- 6 Elbow and Pipeline parameters.

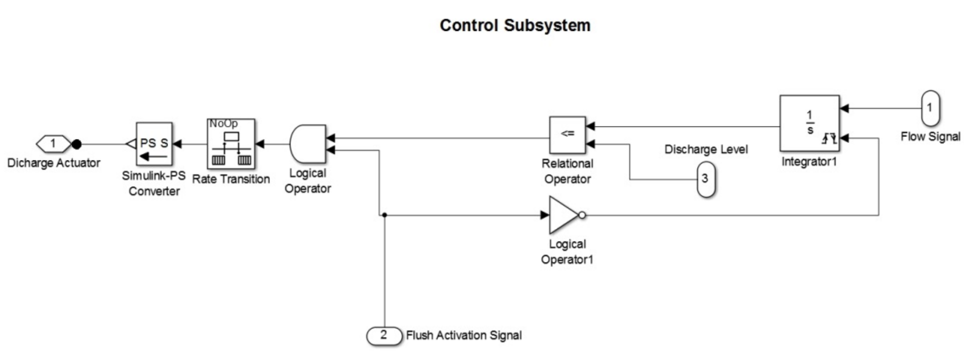

Figure 5- 7 Control System.

The flush activation signal is associated with logical operators (inverter and an "and" gate), it is used to activate the flush when requested. When there is a low on the flush activation signal, there is no flush requested and when there is a high on this signal there is a flush requested. The inverter, inverts the values of the flush activation signal before sending it to the integrator. The inverter allows a positive value to reset the integrator as soon as a flush finishes. When the user requests for a flush, there is a change of signal in the user interaction package from a low to a high. This high value will be inverted by the inverter and a low pules is sent to the reset of the integrator, which means the integrator starts summing up the flow. The relational operator passes the flow as the value from the integrator is less than the discharge level requested. As soon as the value of the integrator is greater than the discharge lee the relational operator stops the flow. The AND gate is activated when there is a flush requested (when the flush activation signal is high) and when the volume of the water discharges is less than the discharge level. When the user stops requesting for the flush, the inverter resets the integrator by converting a low pulse to a high pulse and then sending it to the reset of the integrator.

The components in the Dataset system are as shown in the Figure 5-8. These components will log the simulation data from the flow and the pressure meters. These components help in achieving the right result and help in effective result analysis for each toilet in the system. The, FlowMeter1 corresponds to the flow meter in the 1st Toilet. In the other modules, the second toilet shall have FlowMeter2, and so on. This naming convention applies to the pressure meters as well.

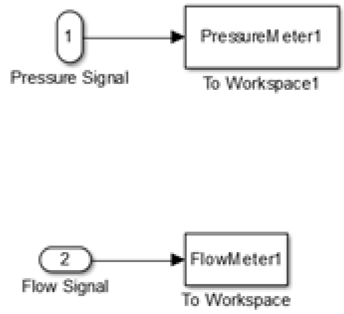

Figure 5- 8 Dataset Subsystem.

5.4 Self-Pressurized Tank (First Module)

The first module consists of a self-pressurized tank which is placed at a 3.5 m height from the ground. Figure 5-9 shows the specification required in the tank and hydraulic pipeline designs. There are 2 elbows which help the pipeline in taking the water from the tank to the flushing valve. The User interaction package and the Toilet package are the same as the ones described earlier. The description of the Hydraulic power package is described in Section 5.4.1.

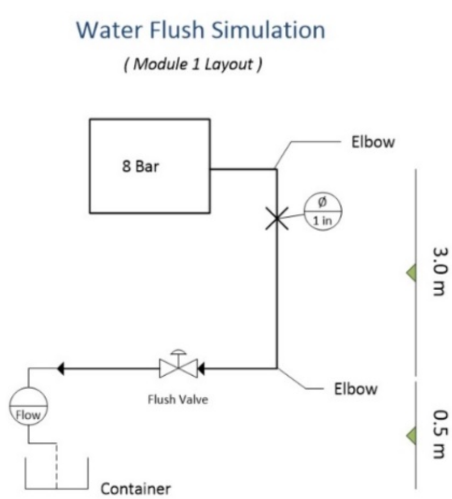

Figure 5- 9 Self Pressurized Tank Design.

5.4.1 Hydraulic Power Package for the Self-Pressurized Tank

The Hydraulic Power Package for the self-pressurized tank is described in this section. Here, the hydraulic sources and sinks such as the tank and the container (also called hydraulic reference) are available. The hydraulic liquid used is set to water for the purpose of this simulation. The tank used (constant head tank component) is self-pressurized at 8 bar (Figure 5-10) which contains one gate, that can be used as an inlet or an outlet, depending on the pressure dynamic existing in the circuit. Another component is a solver, which is responsible for numerical algorithms applied in any physical system in Simulink. Figure 5-11 shows the parameters of the constant head tank where the pressurization is set to 8 bar.

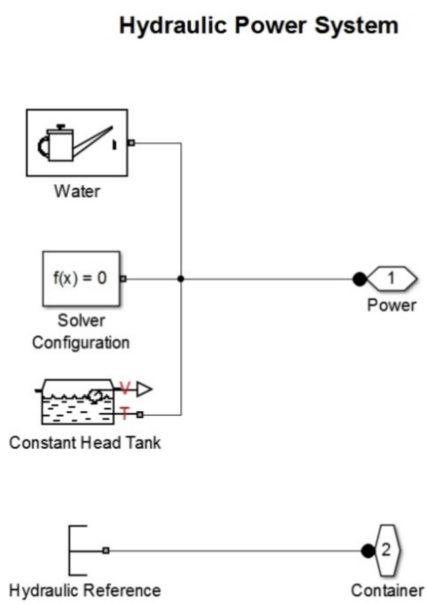

Figure 5- 10 Hydraulic Power System of Self-Pressurized Tank.

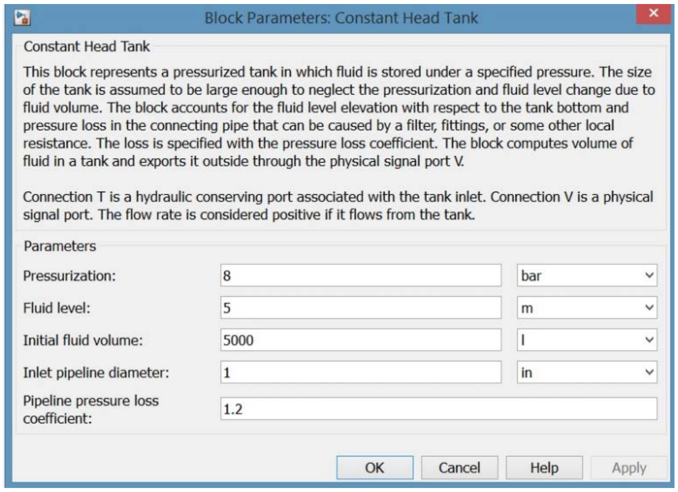

Figure 5- 11 Tank Configuration of the Self-Pressurized Tank.

5.5 The Pressurized Tank and Its Components

A new tank is designed which will be connected to an air compressor and a water pump. In the tank, the air and water will be available in an initial ratio of 4:1. The tank design and specifications can be found in the following Table 5-2. Packages in this module are essentially identical to module 1, except Hydraulic Power Subsystem and the addition of toilet sets connected to the tank. The hydraulic power subsystem package will now include an air compressor and a water pump.

Table 5- 2 Module components of the New Pressurized tank

Components	Diameter (in)	Distance Gate A (cm)	Volume (l)
Tank	50 cm	—	1000
Outlet gate (A)	1	—	—
Inlet water gate (B)	0.5	40	—
Inlet air gate (C)	0.5	50	—

The following Figure 5-12 shows the Hydraulic-Pneumatic pressurized tank design.

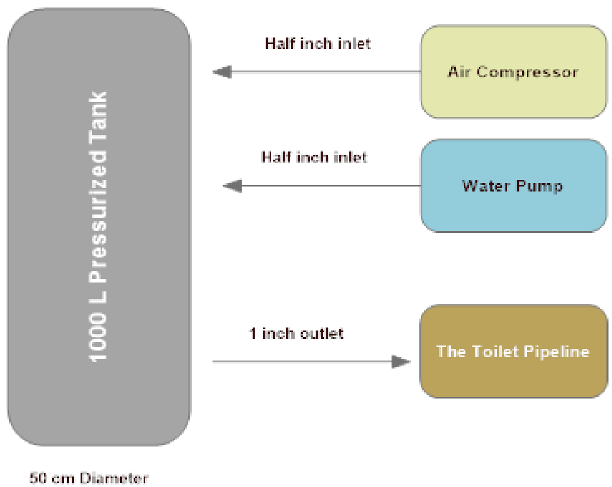

Figure 5- 12 Hydraulic-Pneumatic Pressurized Tank.

This pressurized tank has a capacity of 1000 liters. The maximum volume of water in the tank is 200 liters and the maximum volume of air in the tank is 800 liters. The tank is also connected to an air compressor and water pump. The water pump, pumps water into the tank, when the volume of the water within the tank is below 100 liters and it continues pumping until there is 200 liters of water in the tank. As the volume of water within the tank reduces with different flushes, the air in the tank expands and the pressure of the water reduces. When the pump is operating, the air in the tank is compressed and the pressure slightly increases.

5.5.1 The Pressurized Tank Generation

Simulink does not have elements to represent an air and water pressurized tank. To accomplish an air and water pressurized tank, the SimScape Core Language is used and a code is designed. The custom element represents the laws of physics and applies them in this situation. The code for generating this tank is available in Appendix E.

The tank is modelled using the Ideal Gas Law, Water Column Pressure equation, and the pressure loss equation.

- The Ideal Gas Law [55], is responsible to model the air within the tank as well as the change in pressure with the change of volume. This law is represented in the form of an equation shown in equation 6.2. The temperature is treated as a constant in this implementation.
- The Water Column Pressure [56] is responsible to model the pressure created by the increase or decrease of water level inside the tank.
- The Pressure Loss equation [56] is used to model the decay of pressure with the increasing flow and also due to the flow resistances due to the contact with the walls of the elements.

In this way, the tank´s variables will change dynamically along the simulation time. These variables mainly depend on the demand of water by the toilets, the activation of the water pump and the air compressor. These equations are used in the implementation of the tank in Simulink.

At the tank code implementation, each of the above equation will be treated as an independent component, and it will receive a set of variables that will implement the behavior of the tank. After which, each component will be summed up to get the final total pressure with which the water is flushed out.

The code contains the main pressure equation, which is modelled by the following,

$$P = P_h + P_o + P_g - P_{loss} \tag{6.1}$$

Where, P_h, represents the hydraulic pressure created by water column influence,

P_o, represents the initial pressurization,

P_g, represents the Gas Pressure that will implement the Ideal Gas Law, and

P_{loss} represents the component which will decrease the pressure based on the Pressure Loss Equations.

Water column pressure and loss equations,

$$p = p_{elev} - p_{loss} + p_{pr} \tag{6.2}$$

$$p_{elev} = \rho * g * H \tag{6.3}$$

$$p\_loss = K\, \rho/(2A\_p\hat{}2\ )\ q|q| \tag{6.4}$$

$$A\_p = (\pi * d\hat{}2)/4 \tag{6.5}$$

Where,

p Pressure at the tank inlet

p_{elev} Pressure due to fluid level

p_{loss} Pressure loss in the connecting pipe

p_{pr} Pressurization

ρ Fluid density

g Acceleration of gravity

H Fluid level with respect to the bottom of the tank

K Pressure loss coefficient

A_p Connecting pipe area

d Connecting pipe diameter

q Flow rate

In the code, all the parameters of the design of the tank (specified in Table 5-2) will be treated as constants. Also, parameters such as gravity (9.8 m/s^2) and pressure loss coefficient (0.001) will be treated as constants. The pressure loss coefficient should be experimentally found, which depends on the material used, in other words, for commercial elements the material and design used by the manufacturer will define this value. In this thesis, this value is set by default. The parameters which represent the air or water features, such as their density [57] [58], are also treated as constants in the code.

In the Equation 6.5, it is observed that the connecting pipe area is also calculated by taking into account the gate diameter.

Ideal Gas Law, $$PV = nRT \tag{6.6}$$

It is essential, to observe the influence of the pressurized air in the system. The ideal gas law states that the pressure inside an enclosed tank whose volume is constant, is also constant unless there is a chemical reaction or the variation of temperature in the system. The temperature for the purpose of this simulation is kept constant. However, gas also has its own participation in the pressure phenomena, otherwise a “heavy atmosphere” would have the same effect as that of a “light atmosphere” in some specified space. This heavy effect is the pressure´s effect, force over area, and it will be shaped by relative density of air.

Natural dried air density is approximately 1.2 Kg/m^3, and this value changes if to a fixed volume/space there is a permanent injection of extra gas mass. If there is a greater amount of gas inside the same volume, the air density is heavier. This is possible due to the special compressibility feature of the gases. Such effect is modeled in an equation (derived from Ideal Gas Law) shown in Equation 6.12.

The equation 6.11, can be derived from the Ideal Gas Law in the following steps:

$$P\_g = (n * R * T)/V\_g \tag{6.7}$$

Where, $n = (Gas\ Mass)/(Molar\ Gas\ Mass)$ (6.8)

Applying, (6.8) in (6.7),

$$P\_g = ((Gas\ Mass)/V\_g\ ) * ((R * T)/(Molar\ Gas\ Mass)) \tag{6.9}$$

Also, $Gas\ Relative\ Density = (Gas\ Mass)/V$ (6.10)

And, $K\_g = (R * T)/(Molar\ Gas\ Mass)$ (6.11)

Applying 6.10 and 6.11 in 6.9,

$$P_g = Gas\ Relative\ Density * K_g \tag{6.12}$$

Where, P_g is Gas Pressure,

V_g is Gas Volume,

K_g is Gas Constant,

R is the Universal Gas Constant,

T is Temperature

N is the amount of Substance (number of moles)

Gas Relative Density is the change in density as the air mass is injected into the system (when the air compressor is activated) when the volume of air is constant (considering that the volume of air will change only when the volume of water changes).

$$\text{Gas Relative Density} = (Current\ Gas\ Mass)/(Volume\ of\ Gas\ in\ tank) \qquad (6.13)$$

Also, Kg, is a constant value with respect to the gas features (temperature, gas molar mass and universal gas constant), and Current Gas Mass increases with Gas Mass injections from the air compressor.

With the help of these equations, it is observed that the gas pressure (P_g) is increased when the air compressor is on and the gas mass within the tank is increasing. Also, when the water is flushed out of the toilet, the air with in the tank expands, thereby increasing its volume and reducing its pressure. When the water fills up in the tank, the air volume is compressed and the pressure of the air within the tank increases.

Once the tank dynamics are modelled, the new element can be applied in this module, as shown in the Figure 5-13. The AirWater Pressurized Tank is the new element. This new element is connected to a Compressor and Pumper package. The tank is being driven by an outlet control which is derived from the user interaction signal. This user interaction signal indicates to the new element when a toilet requests for a flush, thus generating flow rates.

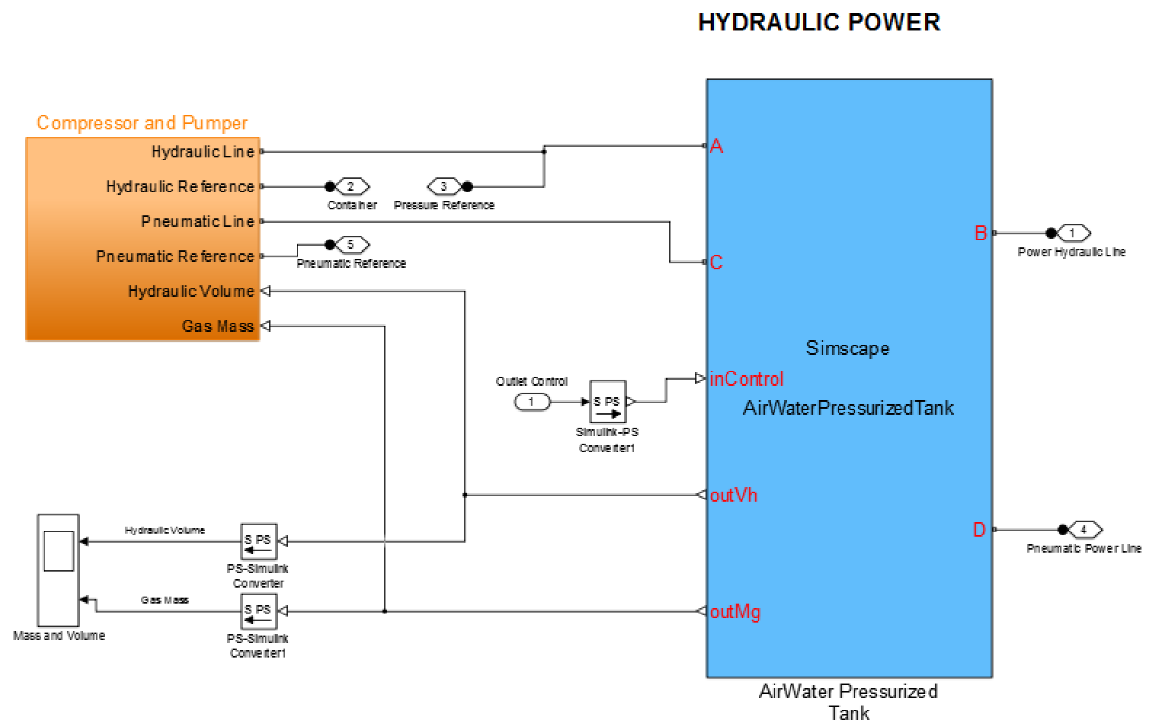

Figure 5- 13 Hydraulic Power System.

In the Figure 5-13, the newly created tank has various input/output connections, in the form of Gates A, B, C and D. The gates A and B correspond to the hydraulic inlet and outlet respectively, while the Gates C and D correspond to the pneumatic inlet and outlet line respectively. The outVh is the output of the Hydraulic Volume which can be represented on the Volume graph. The input gate inControl, is the signal from Outlet Control Group (Figure 5-14), this signal will indicate to the tank when a user presses the flush in order to generate the water flow at gate B.

The Outlet Control Group, is basically an "or" logical gate connected with the user interaction signal sources. The purpose of this package it is to show when the tank should allow flow rate to occur, in other words, if at least one of the toilets require a service at any time, the tank should create flow. The outlet control group of module 2 can be seen in Figure 5-14. The outlet control group of module 3 has 12 user interaction signal sources connected from 12 different toilets to a single “or” gate as shown in Figure 5-15. Logical operator outputs a Boolean signal, data type conversion is there to assure a proper conversion to double values read by outlet control.

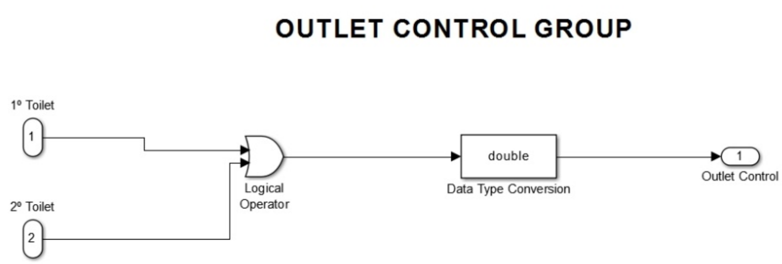

Figure 5- 14 Outlet Control Group for 2 user system.

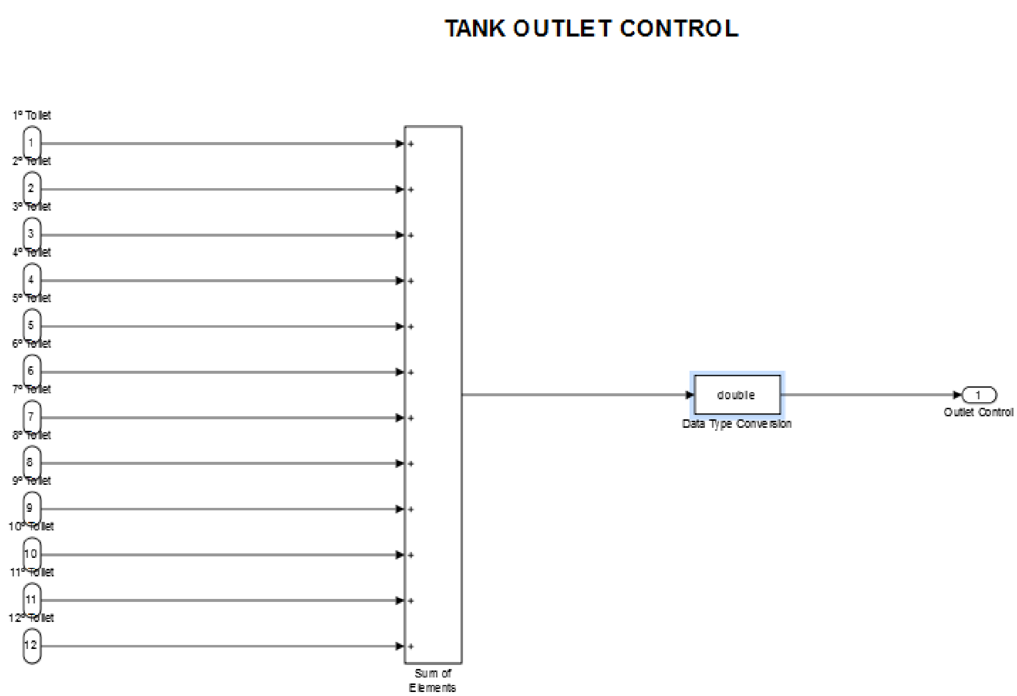

Figure 5- 15 Outlet Control Group for 12 user system.

In the process of creation of the tank, the specifications mentioned earlier were considered. The total hydraulic volume and air will depend on the ratio entered in the parameters dialog box for the AirWater Pressurized Tank shown in Figure 5-16. This ratio will be applied to determine the maximum volume of air and water which can be filled in the tank based on the volume of the tank. The tank designed is a 1000 L tank and the ratio specified is 4:1 where there is 4 parts of air and 1 part of water within the tank. This indicates that the maximum water volume with in the tank is 1/5th the volume of the tank (0.2 m^3 or 200l, considering an 1000Kg/m^3 water density) and the maximum gas volume within the tank is 4/5th the volume of the tank (800 liters). In order to make an air pressure of 8 bar within the tank, the gas mass within the tank should be 7.6 kg. This 7.6 kg of air should be accommodated in 800 liters of the tank. The weight of 7.6kg is calculated using Ideal Gas Law as described in the equation 6.6. The tank is designed to have 7.6 kg in the tank at the beginning of the simulation run. Also, the minimum volume of water and gas mass with in the tank is set as 10% of the total tank volume. This implies the volume of water within the tank should be at least 100 liters. If the water volume goes below the specified level, the water pump will start operating.

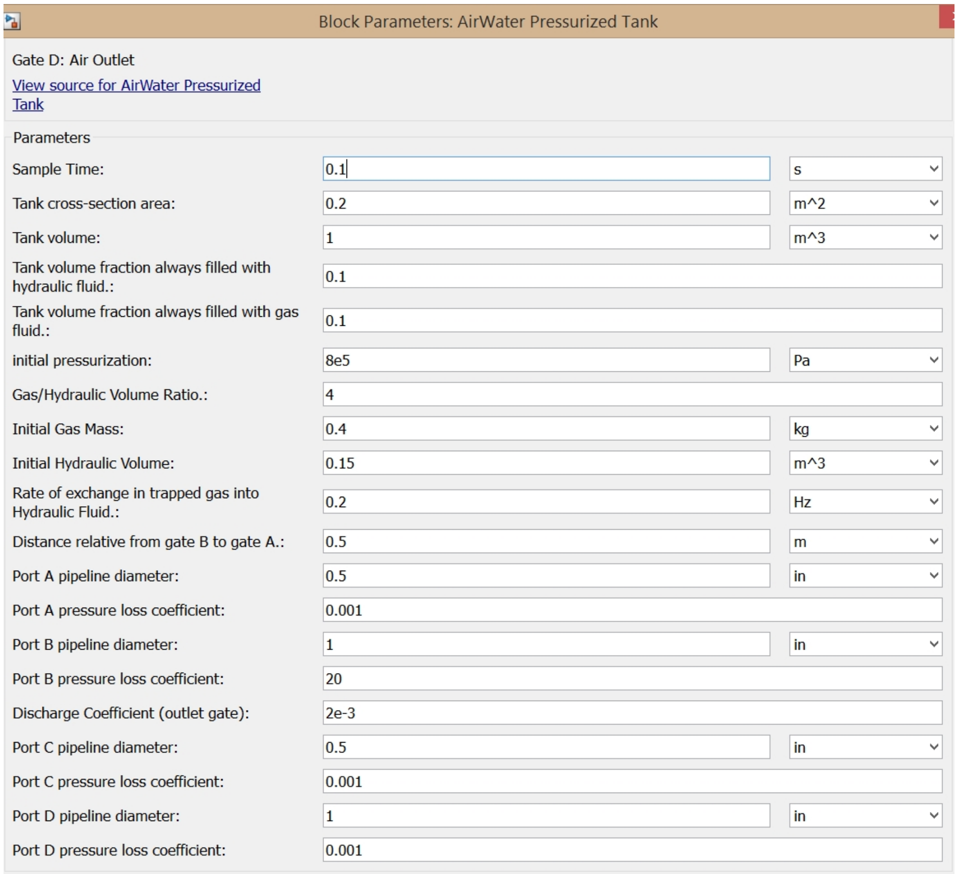

Figure 5- 16 Module Tank Configuration.

Figure 5-16 also shows the parameters which influence the pressure loss coefficients, such as the tank cross sectional area, which is the area of the tank calculated considering the tank diameter. The discharge coefficient, depends on the material specifications and the design of the internal gate in the tank, this value will be defined by manufacturers but in this context it was obtained based on Simulink values for discharge coefficients and adjusted experimentally. The pressure loss equations, take the values of the discharge coefficient, as well as the cross sectional area of each gate (obtained by the circular area formula, applied in each gate diameter), and the flow in that gate. Thus, in the gates A and B, the water flow rate that will be considered. Whereas, through the gates

C and D the gas flow rate will be considered. The increase of flow leads to an increase of pressure loss.

5.5.2 Compressor and Pump Package

The air compressor and the water pump will be implemented in the compressor and pumper package. The compressor and pumper package contains two power lines, one for hydraulic fluids (shown in Figure 5-17) and the other for pneumatic fluids (shown in Figure 5-18). Hydraulic flow rate is created by action of an ideal pump (hydraulic constant flow rate Simulink element), whilst pneumatic is created by an ideal air compressor (pneumatic constant flow rate Simulink element). In this module, one of specific objectives, is to control the level of the water, and amount of gas within the tank. Such that, as soon as the water or gas in the tank reaches a minimum level, it will activate the respective actuator (pumper or compressor). When the water or gas in the tank reaches the maximum level, it will de-activate the respective actuator (pump or compressor).

The hydraulic constant flow is set to a value of 0.003 m^3/s (3l/s) whilst the pneumatic constant flow is set to 0.4 Kg/m^3. These values can be changed, in order to simulate different environments and to observe the action of the air compressor and the water pump. The results can also show the tank dynamics.

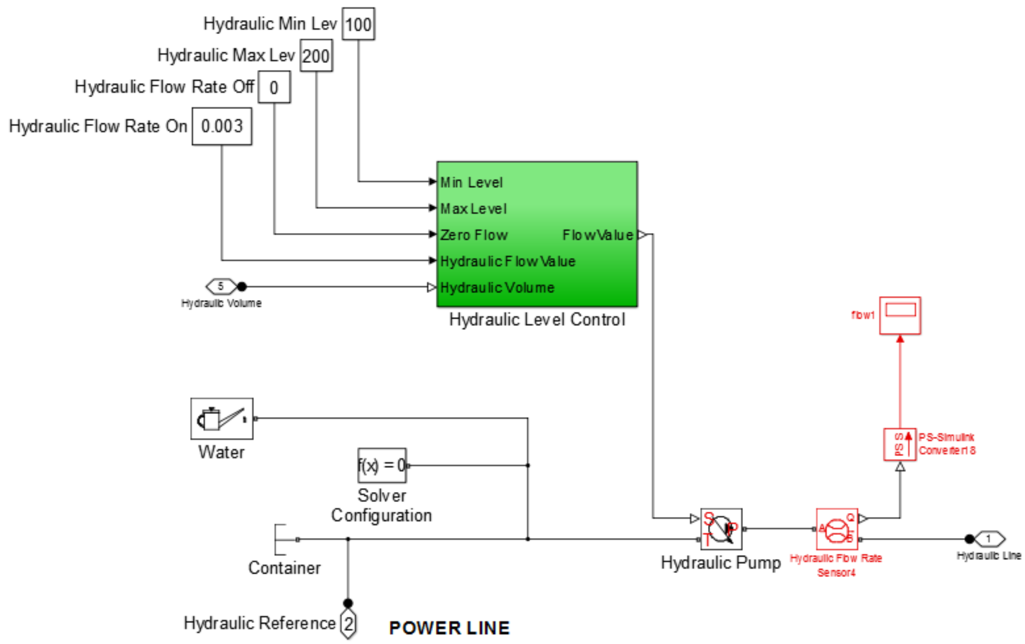

Figure 5- 17 Hydraulic Power System.

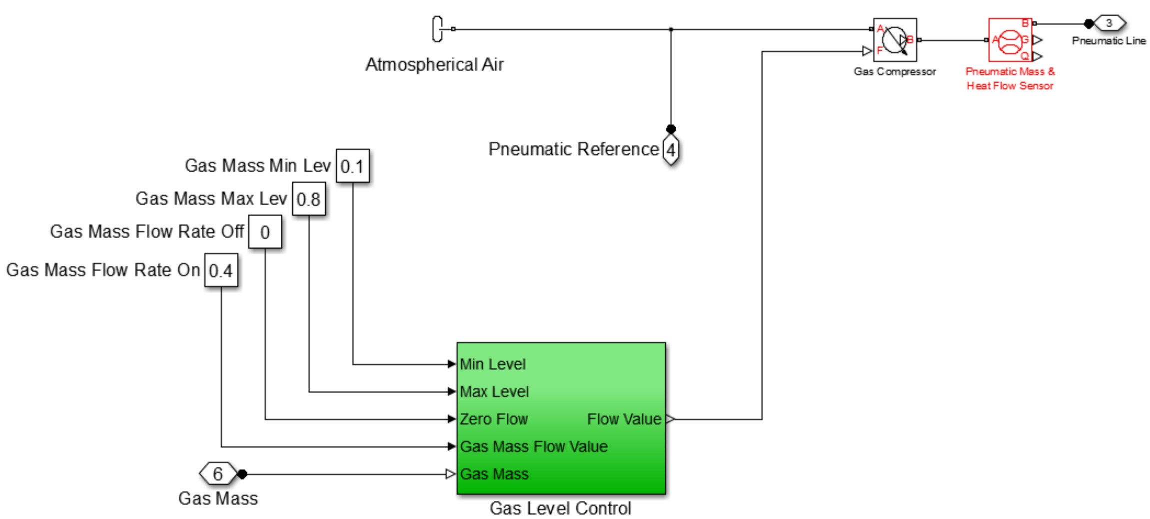

Figure 5- 18 Pneumatic Power System.

To control the dynamics of the tank, another package was designed, which measures the current flow of the air and the water. Each of the power line has its own control level package. Each of the control level package of the air and the water obey the same logic. This package can be observed in Figure 5-19. The hydraulic volume and the gas volume will be compared by the relational operators with the maximum and minimum levels of air and water in the tank. The relational operator drives a switch which will allow or block, the air or the water from flowing in the tank. The value of the switch is specified as zero flow (turned off state) or a constant flow value (turned on state) which is given by the package in Figure 5-17 and 5-18.

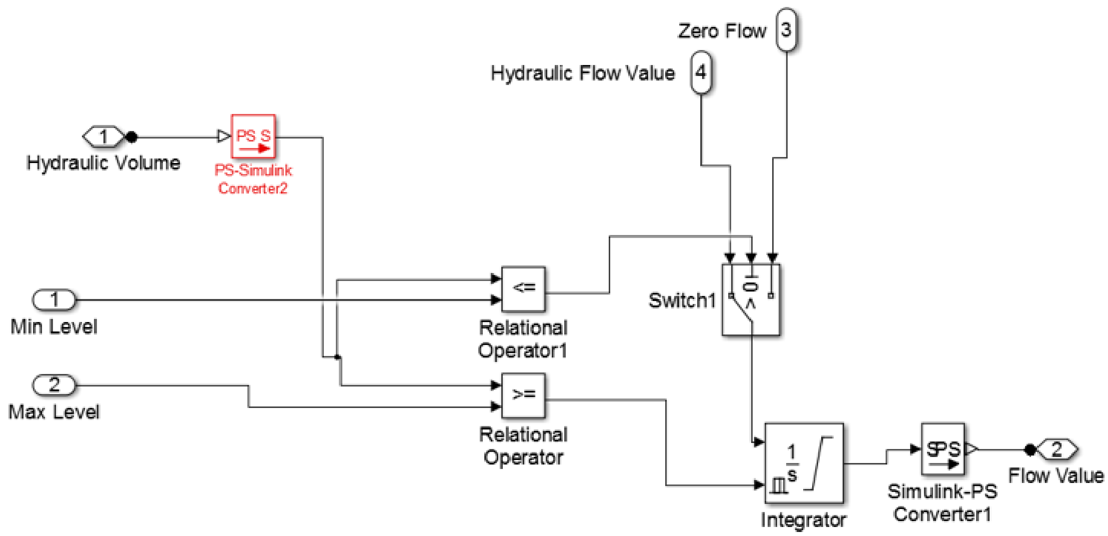

Figure 5- 19 Level Control System.

The switch serves as a commutator of states. Each time a certain requirement is reached, the state is interchanged. In this package, zero is specified as the low signal or the element control and the condition greater than zero is considered as the high signal. Therefore, each time there is a high signal, from the relational operator between the minimum level and the volume of water in the tank, at the input of the switch, the state will be commutate and the flow is allowed to pass.

The current volume of the water in the tank is continuously monitored and is compared with the minimum and maximum value connectors specified at the power line level. Each time the current volume is less than the minimum value, a high signal is generated, forcing the switch to allow the flow to pass. When the current volume goes above the maximum level, then a high signal inputs a zero to the integrator, eliminating the value accumulated and applying zero flowrate.

The switch is useful in changing the zero flow rate to a constant flow rate. However, the switch is a memoryless element and it does not keep values of the previous states. Which implies, that as soon as the volume is greater than the minimum level, the switch will start passing the zero flow value and deactivates the actuator. However, this behavior of the switch is undesirable as the flow into the tank has to be made until it reaches the maximum volume.

In order to keep the past values an integrator is added into the level control package. The integrator component, will accumulate values along the time. If the switch passes any specific constant value, this value will be stored in the integrator component. The integrator has a reset property, which will be activated using the result of the relational operator between volume value and the maximum level (or amount). When the integrator is reset, it will switch back to the zero flow which the switch sends it, representing the actuators in a turned off state.

5.6 Two User System (Second Module)

This module uses the pressurized tank and tests it with the help of two working toilets, allowing the possibility of services at the same time or at different times. If both toilets request for a flush at the same time then there can be two possible outcomes.

Either both can be flushed at the same time or each flush can happen at one time with a small delay between each of the flushes.

The main packages of Module 2 can be seen in the Figure 5-20. There will be just one Hydraulic Power package which will supply water to all the working toilets. Each toilet has a user interaction package of its own.

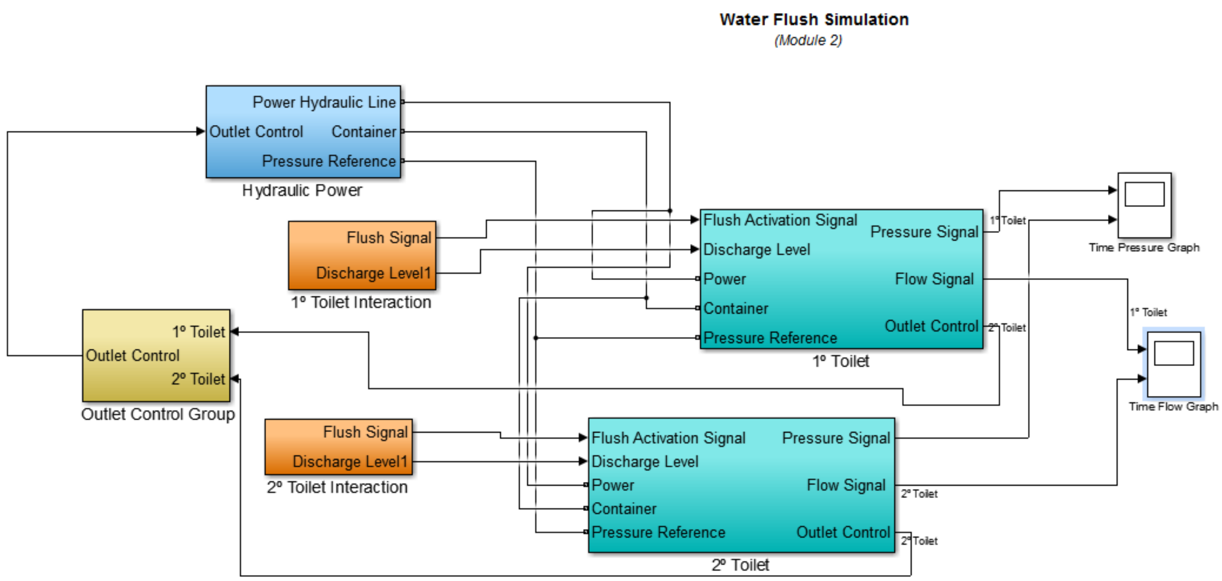

Figure 5- 20 Main Packages of the tank connected to the 2 toilets.

The Figure 5-21 shows the user interaction system for a single toilet for this module. There are two selectors which help determine the time of the flush with respect to the other flushes. The first selector shown in the Figure 5-21, when in the On position, allows each user to flush at a different time. When this selector is in the Off position, each user selects to flush at the same time. When the second selector is in the On position, each user flush is serviced at the same time, that is all the toilet with this selector in the On position will be serviced at the exact same time. However, when it is in the off position each user flush is delayed by a very small time instant.

In the second selector branch, there is a delay element. Each toilet in the module will have a different delay associated with it. That is there is a delay between the time the flush is requested and the flush is serviced. So the first toilet will have a delay of 2 units (z^{-2}), as shown in the Figure 5-21 (which is 200 milliseconds), the second toilet can have a delay of 5 units (z^{-5}) (500 milliseconds) and so on. This design helps in implementing the service of one flush at a time even though more than one flush is requested at a time.

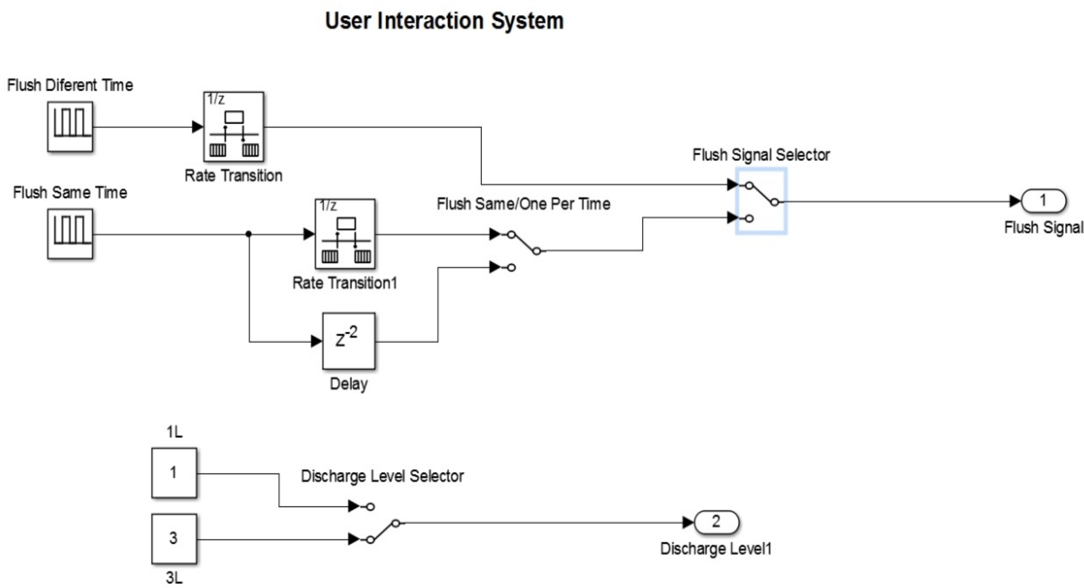

Figure 5- 21 User Interaction system for Multiple Users.

Another element shown in each of the branches is a signal generator and the Rate transition as explained earlier. The signal generator simulates the user arrivals. Simulink allows elements to work in different sample time, and still creates analyzable results. For example, some elements such as the tank, or the hydraulic pipes, are continuous in time, while other elements such as the combinational logic gates work with discrete time.

The flush signal will affect the behavior of the toilet valve. Here the flush signal and the toilet valve work at different configurations. The signal generator, is discrete and the toilet valve is continuous. Hence, the rate transition element is used in each of the branches. The branch with the delay element does not make use of a rate transition because the delay element, takes the rate transition into consideration.

Hence, the Module 2 is designed and used for gathering and analyzing the results.

5.7 Twelve User System (Third Module)

This module is used to check the working of 12 different toilets connected to one pressurized tank, allowing the possibility of services at the same time or at different times for each of the toilets. If more than one toilet requests for a flush at the same time then there can be two possible outcomes. Either all can be flushed at the same time or each flush can happen at one time with a small delay between each of the flushes.

The main packages of Module 3 can be seen in the following Figure 5-22. There will be just one Hydraulic Power package which will supply water to all the working toilets. Each toilet has a user interaction package of its own.

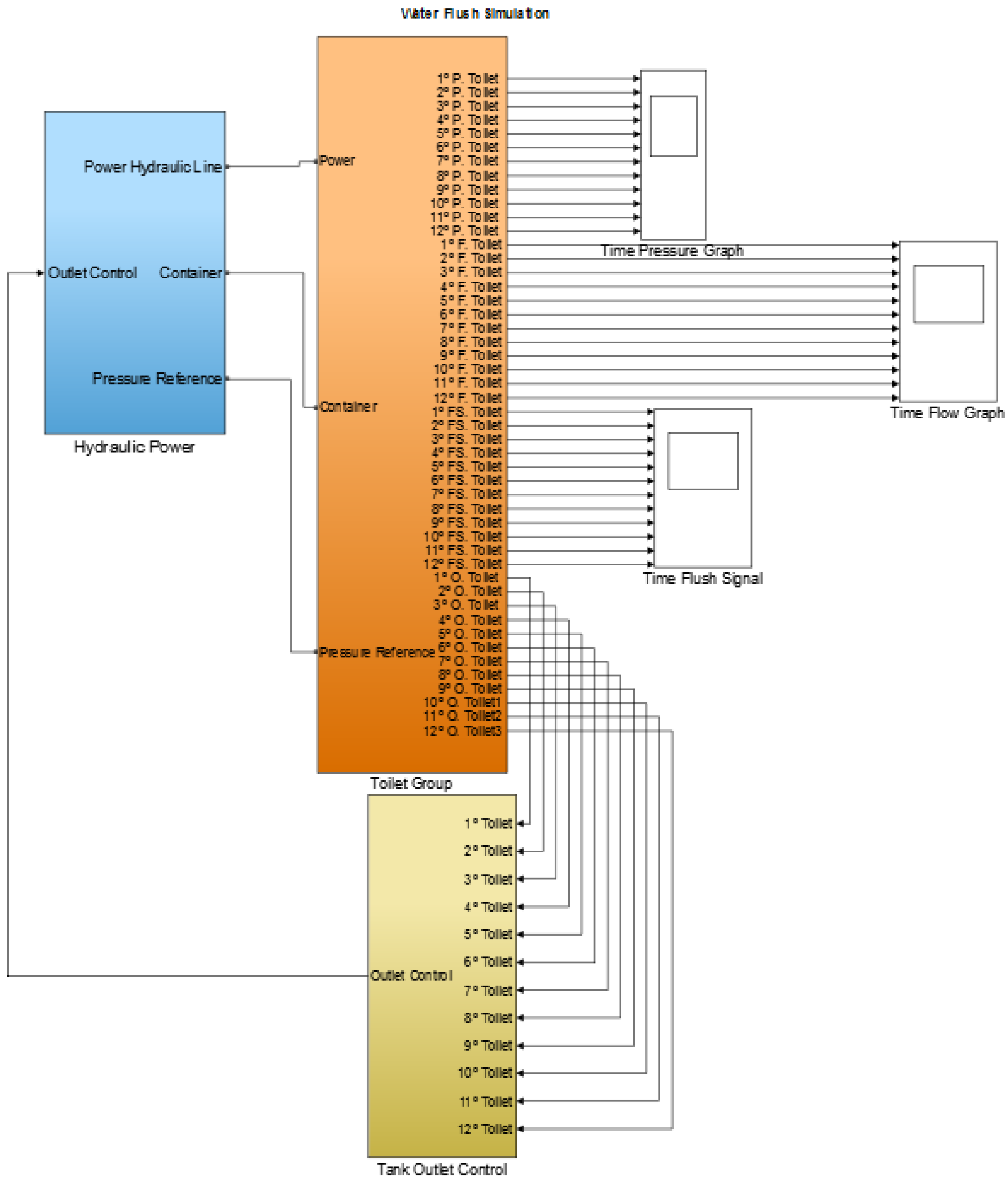

Figure 5- 22 Main packages of the tank connected to 12 toilets.

Due to the increase of toilets, and to improve the design, a new package called the Toilet Group was created. This Toilet Group package consists of the user interaction package and the toilet package for each of the user, which means it has 12 identical set of packages meant for each of the 12 toilets as shown in the following Figure 5-23.

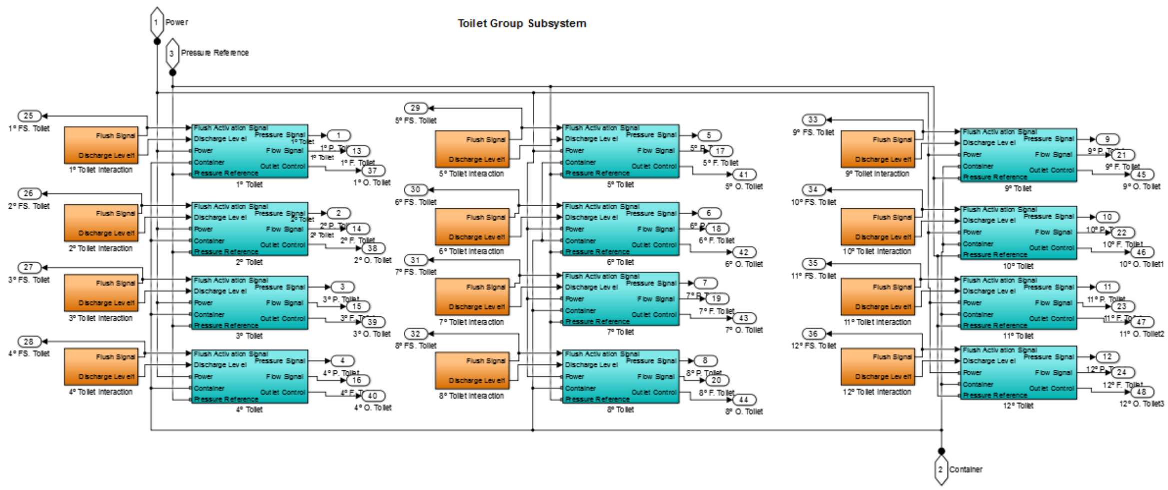

Figure 5- 23 The 12 toilets which are connected to the Pressurized Tank.

The user interaction system for each user is similar to the one described in Module 2. Hence, all the 12 toilets can flush at the same time or different time depending on the configuration of the user interaction package for each user.

Hence, the Module 3 is designed and used for gathering and analyzing the results.

CHAPTER 6

PHYSICAL SYSTEM RESULTS

The physical system was built and many experiments were run on it. To calculate and confirm the results various data was logged during the process. The data was collected from the level and pressure sensors attached to the prototype. The Serial monitor displays the volume of the water within the tank and the pressure within the tank. This data is recorded and used for analysis.

6.1 Scalability of the system

The experiment to characterize the ability of the system determines the amount of water which flows out of the pressurized tank at varying pressures and varying time intervals. Given below are the results of the series of experiments run with different pressures. The initial volume of water within the tank for each run is the same which is 4 liters of water.

1. When the pressure within the tank is 4 bar

The following Table 6-1 and the plot in Figure 6-1 indicates the volume of water which flows out of the tank when the main solenoid valve is open for the corresponding time interval in milliseconds.

Table 6- 1 Volume of Water flowing out at 4 bar pressure for fixed time interval

4 Bar Pressure	
Time (ms)	Quantity (L)
100	0.2
200	0.675
300	1
400	1.4
500	1.6
600	1.9
700	2.1

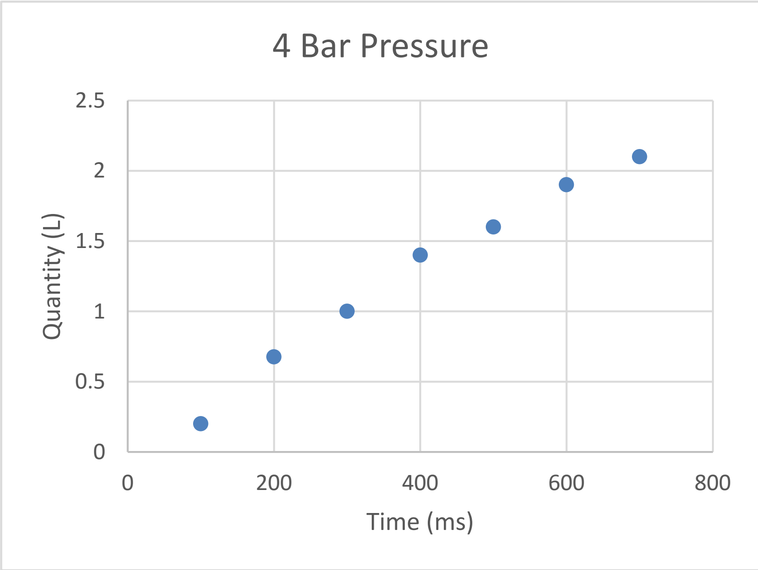

Figure 6- 1 Volume of Water flowing out Vs. Time at 4 Bar Pressure.

2. When the pressure within the tank is 6 bar

The following Table 6-2 and the plot in Figure 6-2 indicates the volume of water which flows out of the tank when the main solenoid valve is open for the corresponding time interval in milliseconds.

Table 6- 2 Volume of Water flowing out at 6 bar pressure for fixed time interval

6 Bar Pressure	
Time (ms)	Quantity (L)
100	0.24
200	0.76
300	1.25
400	1.65
500	2.15
600	2.45
700	2.8

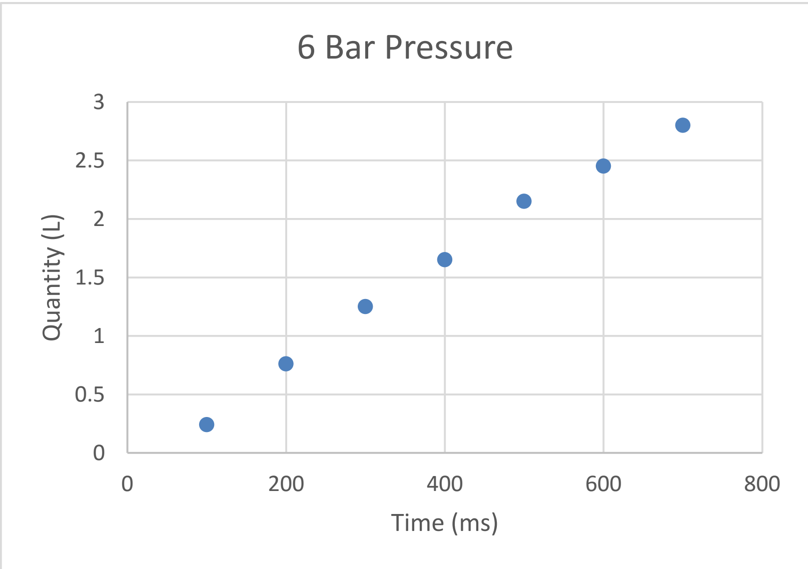

Figure 6- 2 Volume of Water flowing out Vs. Time at 6 Bar Pressure.

3. When the pressure within the tank is 8 bar.

The following Table 6-3 and plot in Figure 6-3 indicates the volume of water which flows out of the tank when the main solenoid valve is open for the corresponding time interval in milliseconds.

Table 6- 3 Volume of Water flowing out at 8 bar pressure for fixed time interval

8 Bar Pressure	
Time (ms)	Quantity (L)
100	0.3
200	0.9
300	1.45
400	1.9
500	2.5
600	2.7
700	3.1

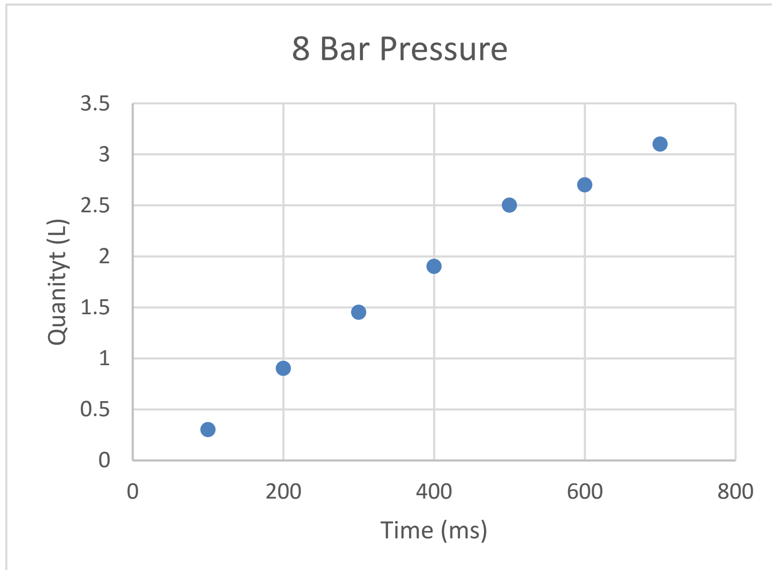

Figure 6- 3 Volume of Water flowing out Vs. Time at 8 Bar Pressure.

6.1.1 Comparison of flows at various pressures

Given below is a comparison of the various flows at different pressures in Table 6-4.

Table 6- 4 Comparison of Volume of Water flowing out at different pressures for varying time intervals

	4 Bar	6 Bar	8 Bar
Time (ms)	Quantity (L)	Quantity (L)	Quantity (L)
100	0.2	0.24	0.3
200	0.675	0.76	0.9
300	1	1.25	1.45
400	1.4	1.65	1.9
500	1.6	2.15	2.5
600	1.9	2.45	2.7
700	2.1	2.8	3.1

Figure 6-4, compares the volume of water flowing out of the tank at different pressures in varying time intervals.

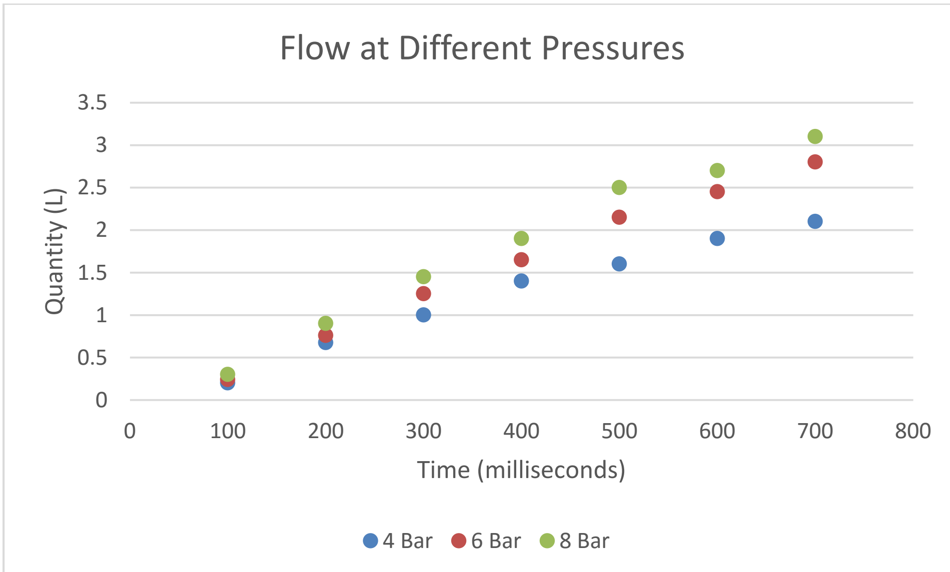

Figure 6- 4 Graphical Comparison of Volume of Water flowing out Vs. Time at Different Pressures.

From the above results, the computation of the average flow rate and the average flow rate is given in the following Table 6-5.

Table 6- 5 Average Flow Rate at Different Pressures

	Average Flow Rate (Liter/min)		
Time (ms)	8 Bar	6 Bar	4 Bar
100	180	144	120
200	270	228	202.5
300	290	250	200
400	285	247.5	210
500	300	258	192
600	270	245	190
700	265.7143	240	180
Average (l/min)	280.119	244.75	195.75
Average (l/sec)	4.668651	4.079167	3.2625

From the flows at different pressures, if we ignore the flow at the 100ms interval considering it to have a greater ON/OFF time, we can get the following instantaneous flow or steady state. For example, if the solenoid valve is on for a longer period of time at 8 bar pressure the water flows out at a rate of 330 l/min for a greater period of time. The steady state flow rate at varying pressures is given in Table 6-6.

Table 6- 6 Steady State Flow Rate at different pressures

	Steady State Flow Rate		
	8 Bar	6 Bar	4 Bar
Steady State Flow Rate (l/min)	330	286.5	210
Steady State Flow Rate (l/sec)	5.5	4.775	3.5

6.2 Flushing system at 8 bar pressure

When the solenoid valve is open, the decrease in volume of the water within the tank is as displayed in the Table 6-7 and plot in Figure 6-5 shown below. The time corresponds to the time during the experiment e.g. at the 345th millisecond of the experiment the water in the tank is 4.086 liters.

Table 6- 7 Volume of water with in the tank at the fixed time

Time (ms)	Volume (Liters)
345	4.086940412
454	3.786940412
566	3.238454525
675	3.088104294
784	2.575482525
896	2.128415155
1002	1.768955548

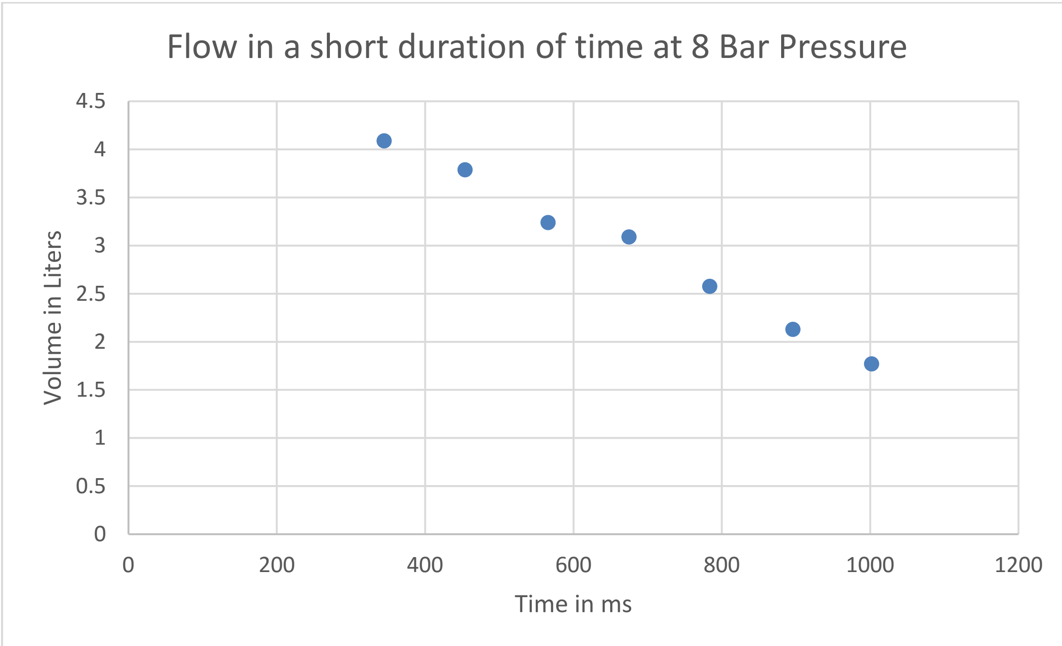

Figure 6- 5 The Degradation of the volume of the water within the tank when the flush is on.

CHAPTER 7

SIMULATION RESULTS

In this chapter, we analyze the results obtained from simulation and compare them with the results of the physical system and flushmate.

7.1 Simulation Result for the Self-pressurized tank

Simulation effectively happens when the run button is hit, and the time of simulation is specified. Once the simulation is run for the time specified, data analysis can be done by calling the function WaterFlushSimulation which is available in Appendix F in the Matlab console. The code for analyzing the flushing is available in Appendix G.

The following Figures show the result from the simulation. Figure 7-1 shows the simulation result for a 60s simulation with three flushes each of 3L discharge. Figure 7-2 shows the simulation result for a 60s simulation with three flushes each of 1L discharge.

The result shows four graphs, the first graph is a comparison between pressure, interval and volume of discharge, whereas the other three graphs show each variable individually. Each bar corresponds to an instant in which the flush is requested by the user interaction signal.

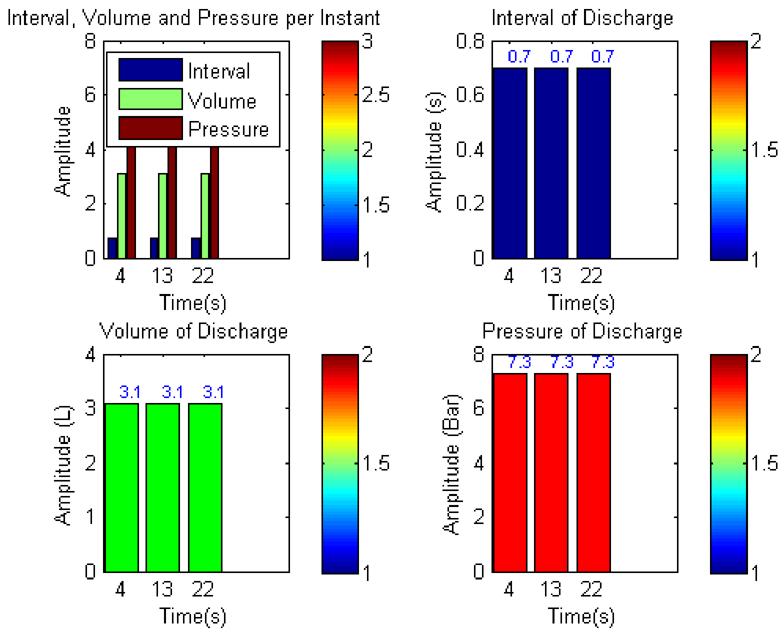

Figure 7- 1 The interval, volume and pressure of water discharged for a Toilet which requests for a 3 L flush.

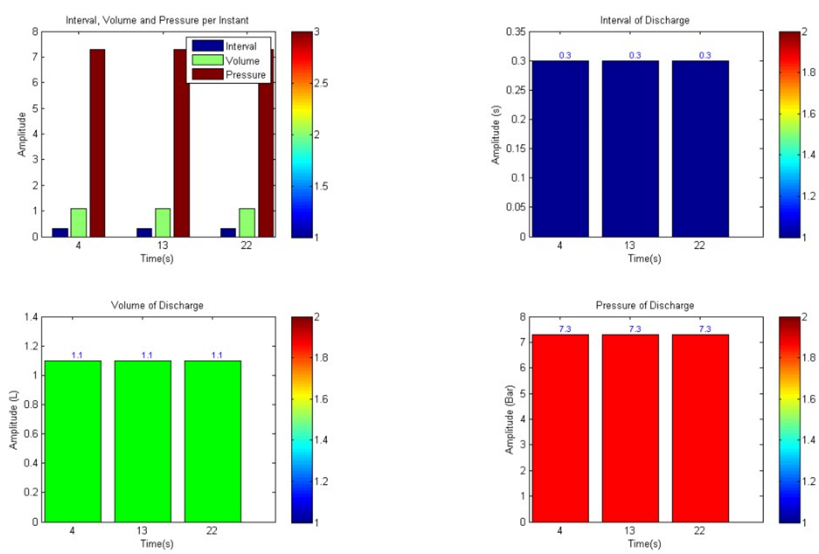

Figure 7- 2 The interval, volume and pressure of water discharged for a Toilet which requests for a 1 L flush.

Figure 7-1 shows that the flush happens with a pressure of 7.3 bar and it takes 0.7 seconds to discharge 3.1L of water. The valve may or may not close at exactly at 3liter discharge as there is some delay between the flowmeter signal and the signal to the discharge actuator.

Figure 7-2 shows that the flush happens with a pressure of 7.3 bar and it takes 0.3 seconds to discharge 1.1L of water.

7.2 Simulation Result for the Pressurized tank with 2 User

This module uses the pressurized tank with the air compressor and water pump connected to it. This tank is also connected to two toilets, these toilets can work in various combinations of time they request for flush and the volume of water requested for flush.

The following Figures 7-3 through Figure 7-8 are for one simulation run. This simulation is run for 60 seconds, in which there are two toilets. Each of the toilet requests for a 3L flush at the same time, 6 times.

From Figures 7-3 and 7-4, it is observed that 3.2L of water is discharged in 0.8 seconds at a pressure of 7.7 bar for both the toilets. Also, as the water in the tank empties and the volume of air in the tank expands there is a slight pressure drop from 7.8 bar to 7.5 bar. This drop increases with increase in volume of air within the tank.

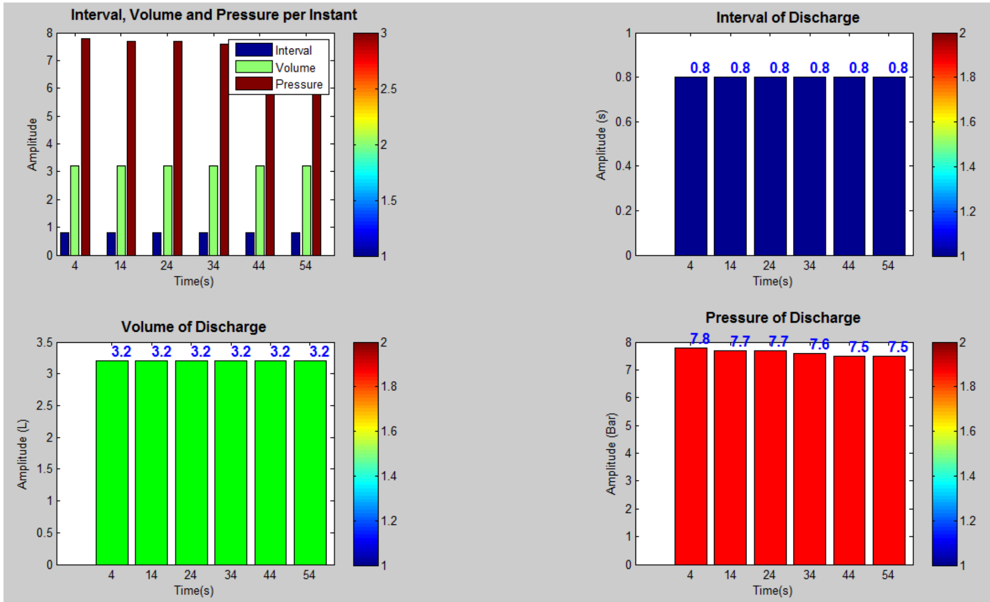

Figure 7- 3 The interval, volume and pressure of water discharged for Toilet 1, when 2 users request for a 3 L flush at the same time.

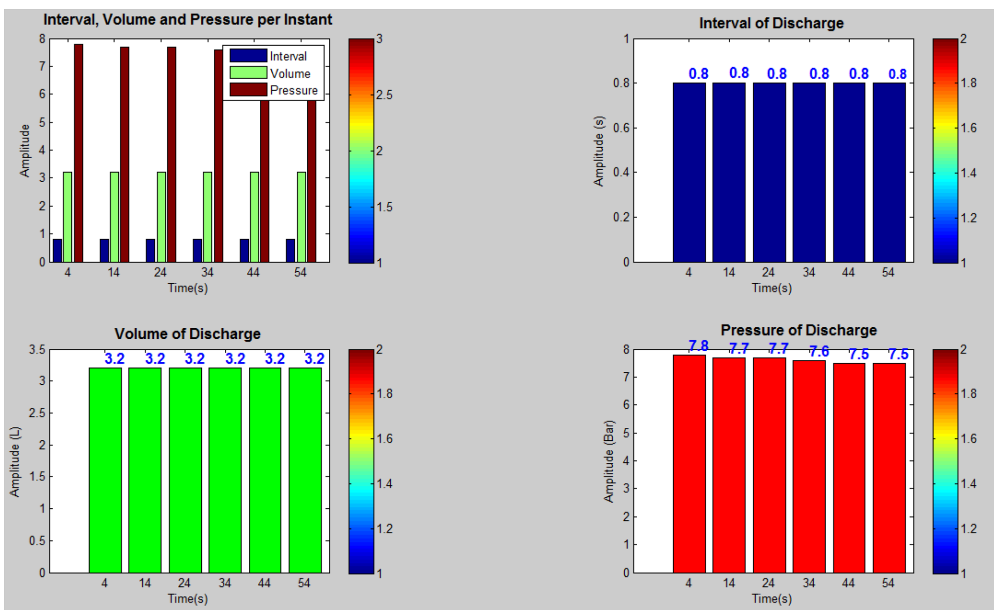

Figure 7- 4 The interval, volume and pressure of water discharged for Toilet 2, when 2 users request for a 3 L flush at the same time.

Figure 7-5 shows the volume of water within the tank gradually decreasing, as the number of flushes increase. The water in the tank is filled up to 200L initially. In the simulation, there are two flushes happening every 10 seconds. Towards the end of the simulation run there is approximately 164 liters of water in the tank.

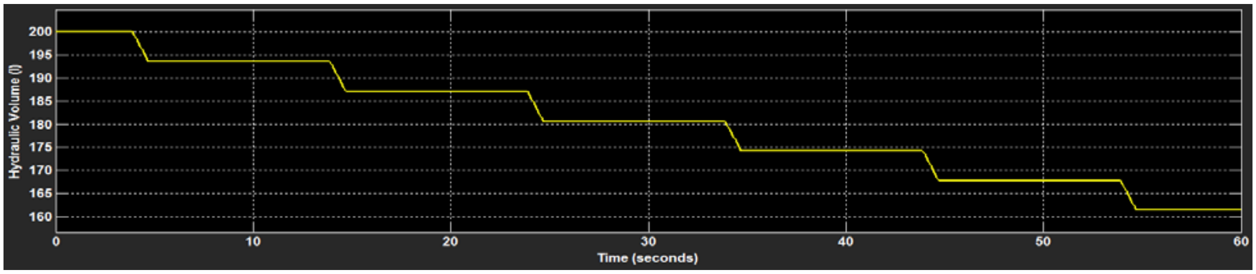

Figure 7- 5 Decrease in Hydraulic Volume in the tank due to the flushes.

Figure 7-6 shows the pressure drop at every toilet when a flush is requested. When there is no flush the pressure of the water in the pipeline is constant. However, when there is a flush there is a pressure drop across every toilet in the pipeline. Figure 7-6 shows the pressure drop across both the toilets. This behavior is a characteristic of Pascal's Law or the principle of transmission of fluid pressure. Here, the pressure exerted on an incompressible fluid in a confined space, is equally transmitted in all the directions of the fluid such that the variation in pressure remains the same.

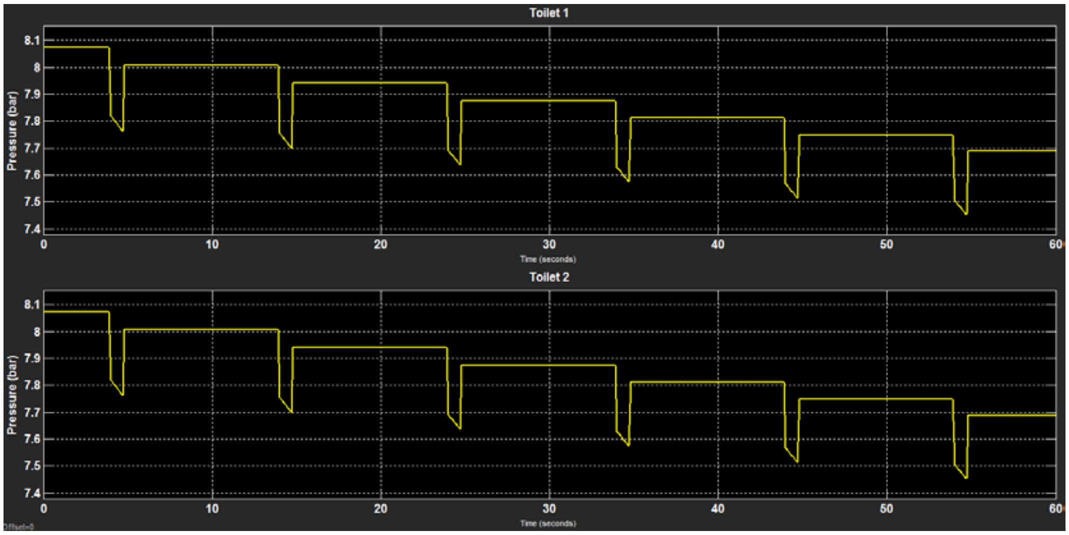

Figure 7- 6 Pressure in the pipeline for both the toilets.

Figure 7-7 shows the flow rate across the different toilets when the flush is requested. The flow rate across the toilet seems to reach a little over 4l/s. That is the speed of the eater flowing out can reach 4L in a second. The time for which the flow rate is on, shows the time interval for which the toilet discharge valve is open.

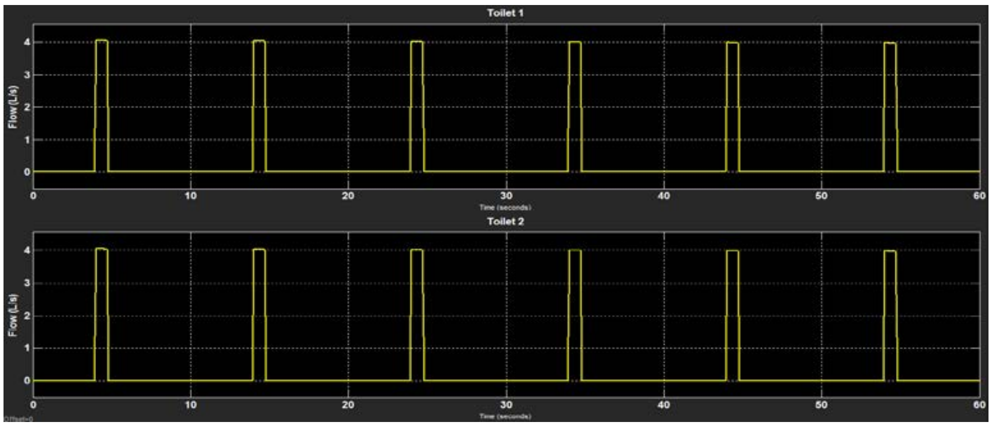

Figure 7- 7 Flow rate (L/sec) at each toilet.

Due to the Ideal Gas behavior and considering the loss of water with in the tank, the initial ratio of 4:1 between gas and water vary along time. As there is a decrease of water, the volume of gas will linearly expand to occupy the empty space left. This is possible due to the compressibility nature of the gas. Figure 7-8 illustrates this situation.

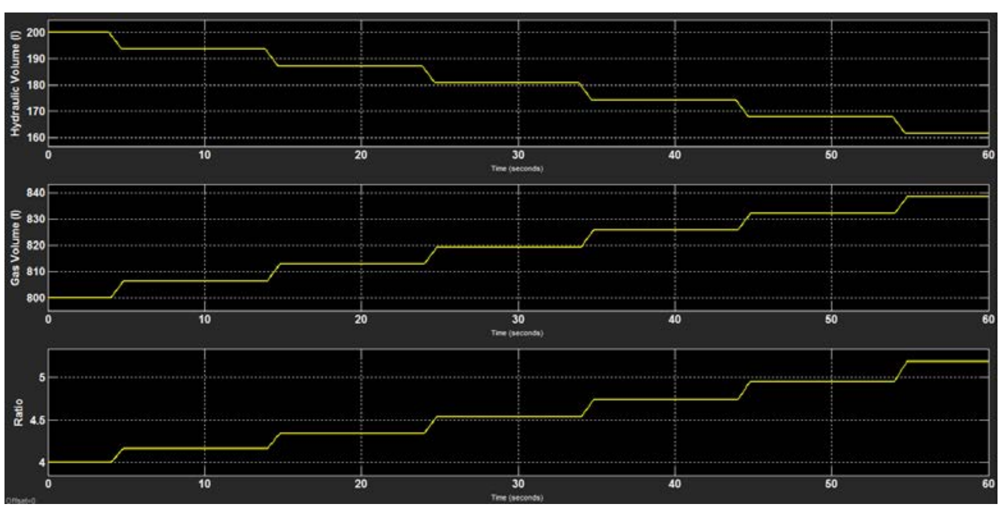

Figure 7- 8 Hydraulic Volume, Gas Volume and their Ratio within the tank during the simulation.

Figure 7-9 shows the flow rate for one toilet for a single flush. It is observed that around 3 liters of water can flushed in 800 milliseconds.

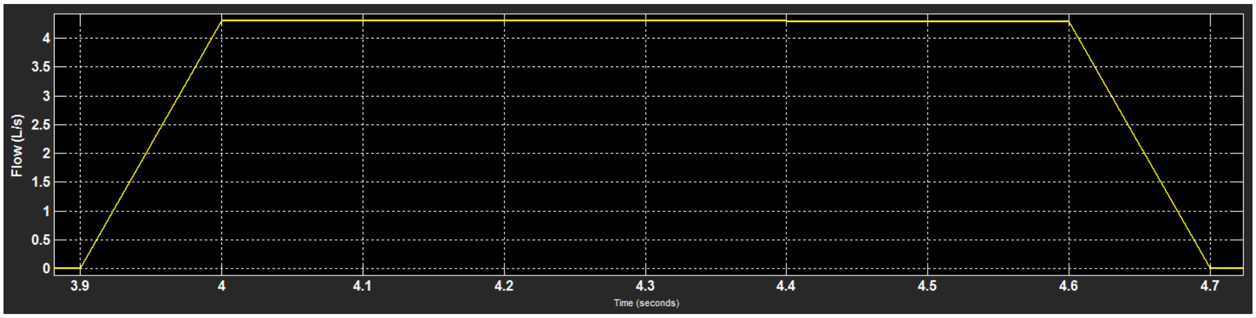

Figure 7- 9 Flowrate for a single toilet (3L flush).

Figure 7-10 and 7-11 represent the pressure, interval of discharge and volume of discharge when both the toilets request for a 1L flush at the same time.

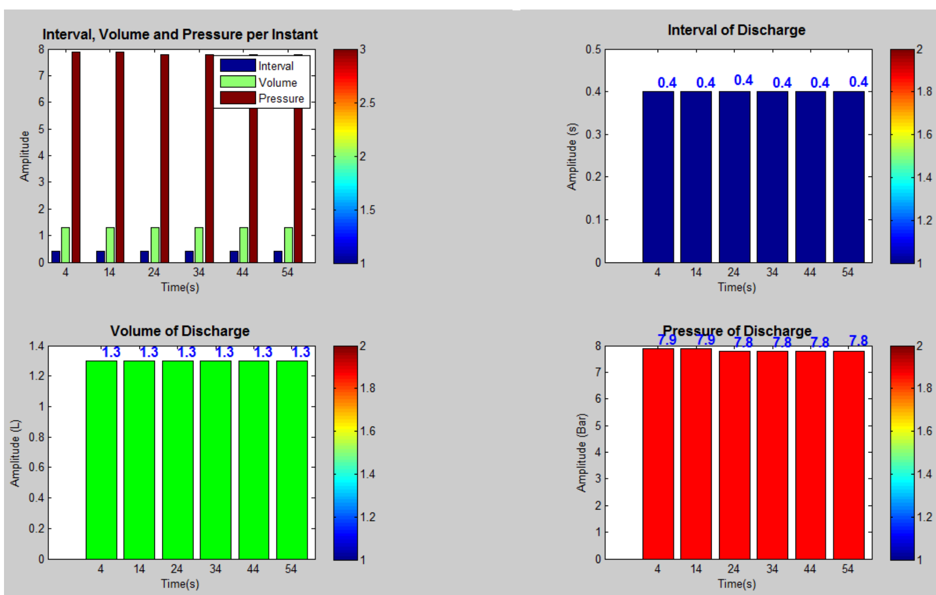

Figure 7- 10 The interval, volume and pressure of water discharged for Toilet 1, when 2 users request for a 1 L flush at the same time.

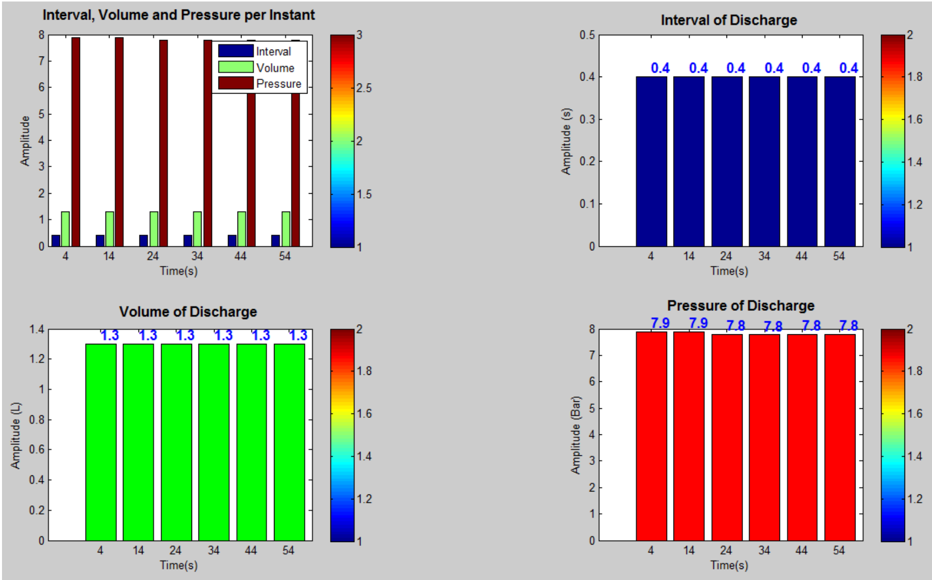

Figure 7- 11 The interval, volume and pressure of water discharged for Toilet 2, when 2 users request for a 1 L flush at the same time.

Figure 7-12 shows the flow rate for one toilet for a single flush. It is observed that around 1liter of water can flushed in 400 milliseconds.

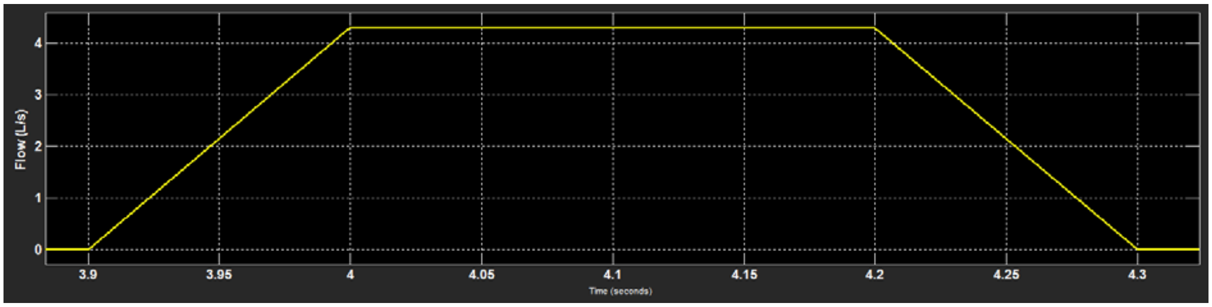

Figure 7- 12 Flowrate for a single toilet (1L flush).

Figure 7-13 shows the pressure drop when the users request for a 3 L flush and a 1L flush at different time. It is observed that the pressure drop for a greater period of time is for a 3 L flush and the drop for lesser period of time is for a 1L flush.

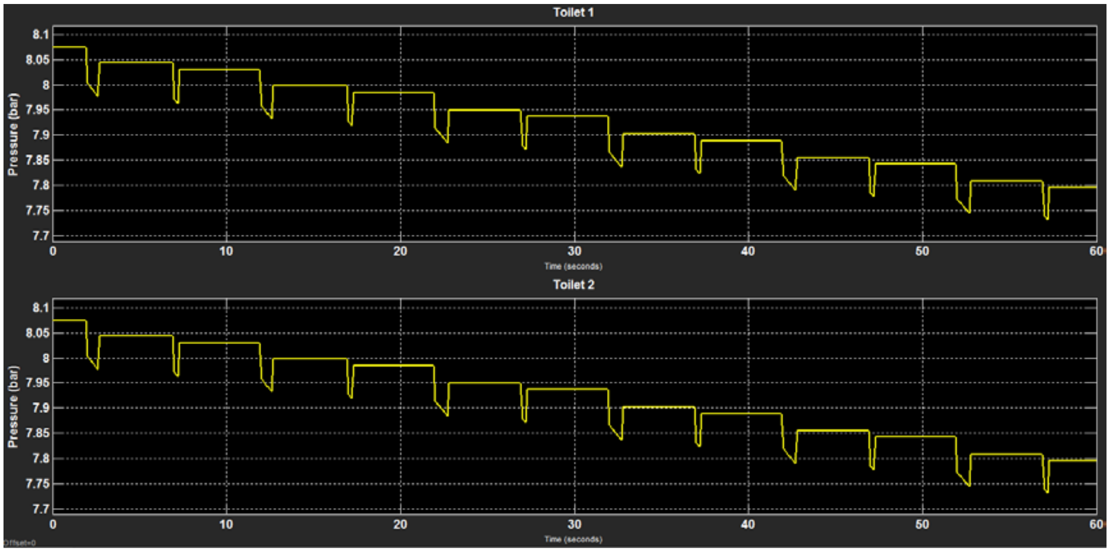

Figure 7- 13 Pressure in the pipeline when the users request for a 3L and a 1L flush at different time.

Figure 7-14 shows the difference in flow rate when the users request for a 3 L flush and a 1L flush at different time. It is observed that the flow period for a greater period of time is for a 3 L flush and for lesser period of time is for a 1L flush.

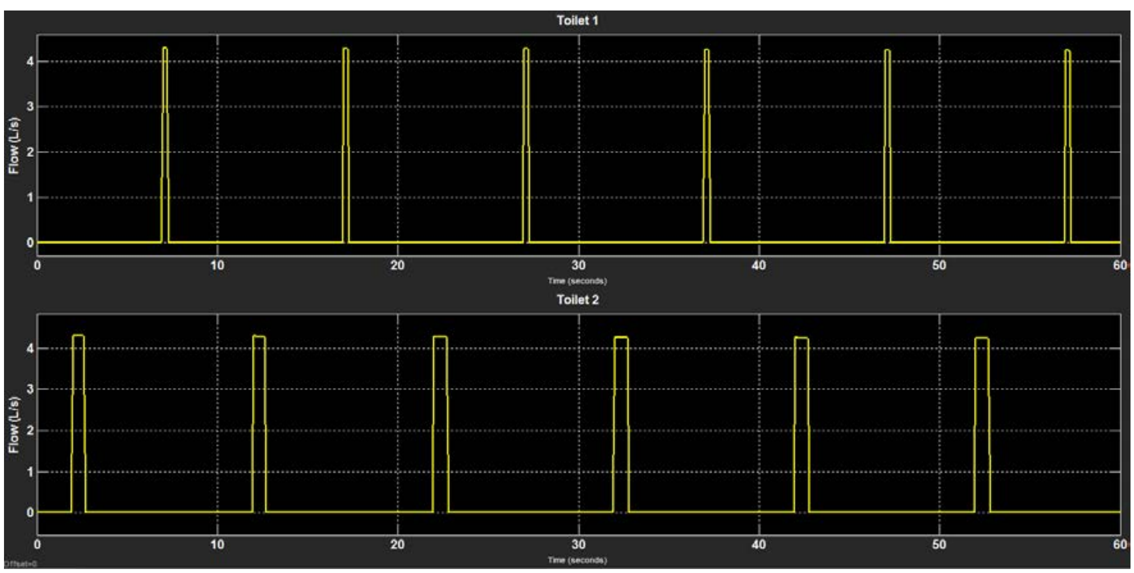

Figure 7- 14 Flow when the users request for a 3L and a 1L flush at different time across the different toilets.

This module was experimented with different combinations. Such as,

- When both the users request for the different flushes at a time.
- Different users request for different flushes at different times.
- Different users request for different flushes at the same time but are serviced with delay between the respective flushes and so on.

However, it is observed that the maximum pressure drop occurs when there is a request for a 3 liter flush by both the users at the same time. The pressure with which this flush occurs is 8.4 bar in 700 milliseconds. The average of the results is shown in the below Table 7-1.

Table 7- 1 Toilet Average Result for 2 Users connected to Pressurized Tank

Toilets Average	Interval (s)	Volume (l)	Pressure(bar)
1L of Discharge	0.4	1.3	7.9
3L of Discharge	0.7	3	7.9

7.3 Simulation Result for the Pressurized tank with 12 Users

This module uses the pressurized tank with the air compressor and water pump connected to it. This tank is also connected to twelve toilets, these toilets can work in various combinations of time they request for flush and the volume of water requested for the flush.

The following Figures 7-15 through Figure 7-18 are for one simulation run. This simulation is run for 60 seconds, in which all the 12 toilets request for a flush at the same time, but six toilets request for a 3 liter flush and the remaining 6 toilets request for a 1l flush. However, the flushes are serviced with a 300 millisecond delay for each toilet.

Figure 7-15 shows the discharge result of any one random toilet which requests for a 3 liter flush.

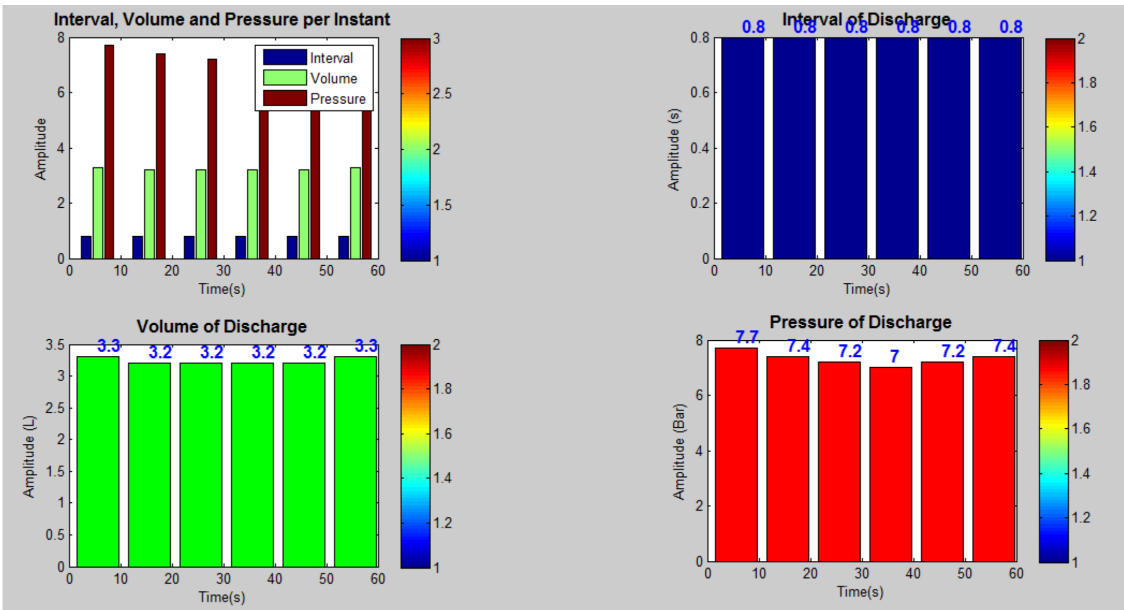

Figure 7- 15 The interval, volume and pressure of water discharged for a Toilet which requests for a 3 L flush.

Figure 7-16 shows the discharge result of any one random toilet which requests for a 1L flush.

Figures 7-15 and 7-16, show that the pressure for the 3L discharge is greater than the 1L discharge this can also be verified from Figure 7-18. Since 3L discharge has a greater period of time for which the valve is On, the pressure drop is more for this discharge.

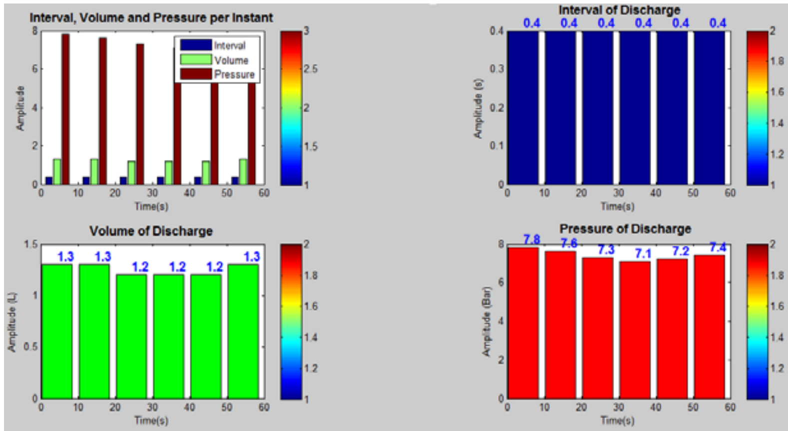

Figure 7- 16 The interval, volume and pressure of water discharged for a Toilet which requests for a 1 L flush.

Figure 7-17 shows the volume of the water decreasing till it goes below 100l in the tank. Once the volume of the water in the tank goes below 100liter, the water pump starts pumping water in the tank. The water pump starts operating at the 36th second of the simulation. The rate at which the water pump, pumps in water is 5l/s for the simulation purpose (which is set in the compressor and pumper package). The water pump, pumps in water into the tank till the volume of the water in the tank reaches 200l. The volume of air to the volume of water ratio with in the tank dynamically changes throughout the simulation.

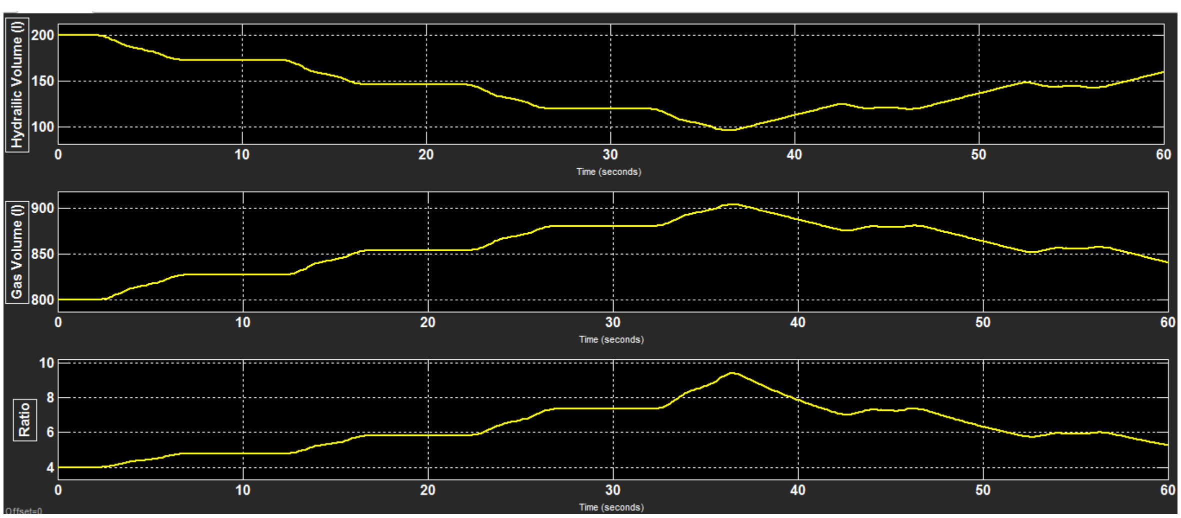

Figure 7- 17 Hydraulic Volume, Gas Volume and their Ratio within the tank during the simulation.

Figure 7-18, shows the pressure in the pipeline. Whenever there is a pressure drop in the hydraulic line, it indicates that there is a flush to any of the toilet in the pipeline. It is observed that the pressure drop is greater when a 3L flush is requested and the pressure drop is lesser when a 1L discharge is requested. When the pressure is almost constant (almost like a straight line) it indicates that there is no flush requested. When the pressure is increasing slowly (after the 36th second), it indicates the pressure in the pipeline is increasing, as the water in the tank is filling up.

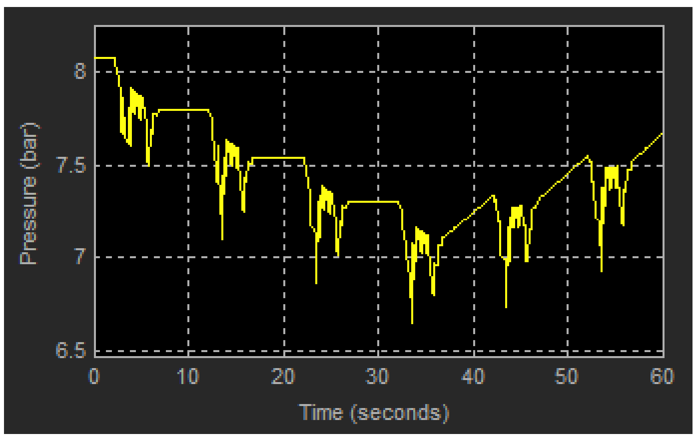

Figure 7- 18 Pressure in the pipeline when the users request for a 3L and a 1L flush at different times.

This module can be run for many scenarios, however for the purpose of this report, a few of the scenarios are tried and tested and tabulated in the Table 7-2. Table 7-2 shows the various scenarios as well as the interval, pressure and volume of discharge of any 2 of the twelve toilets are displayed. Below is brief description for each of the scenario:

1. When all 12 users flush at the same time and request for a 3L flush. This scenario is very very unlikely to happen as all the users will have to request for the flush at the exact same millisecond which is impossible. In this case the pressure drops to

2.1 bar and it takes 1.2 seconds to discharge 3L of water. Hence this can be the worst case scenario.

2. When all 12 users flush at the same time and request for a 1L flush. This scenario is very very unlikely to happen as all the users will have to request for the flush at the exact same millisecond which is impossible. In this case the pressure drops to 2.1 bar and it takes 0.5 seconds to discharge 1L of water.

Table 7- 2 Discharge Result for different scenarios in the 12 user system

No.	Scenario	Toilet 1			Toilet 2		
		Interval (seconds)	Pressure (Bar)	Volume (Liters)	Interval (seconds)	Pressure (Bar)	Volume (Liters)
1	3L/Same/12 user	1.2	2.1	3.1	1.2	2.1	3.1
2	1L/Same/12 user	0.5	2.1	1.3	0.5	2.1	1.3
3	3L/Diff/12 user	0.8	7.5	3.1	0.8	7.5	3.2
4	1L/Diff/12 user	0.4	7.8	1.3	0.4	7.7	1.3
5	3L/Delay/12 user	0.8	7.2	3.2	0.8	7.3	3
6	1L/Delay/12 user	0.4	7.4	1.3	0.4	7.5	1.3
7	Flush Same time 6user/3L, 6user/1L	0.8	6.2	3.1	0.5	2.1	1.4
8	Flush With Delay 6user/3L, 6user/1L	0.8	7.3	3.2	0.4	7.4	1.2

3. When the 12 users request for a 3L flush at different times. The pressure with which the water is discharged is 7.5 bar on an average.

4. When the 12 users request for a 1L flush at different times. The pressure with which the water is discharged is 7.8 bar on an average. This is the best case scenario, as the pressure doesn't drop much because the valves are opened for a short duration of time comparatively.
5. When the 12 users request for a 3L flush at the same time, but they are serviced with a 300 millisecond delay between each flush. The pressure drop in this scenario is not much and can be at a 7.2 bar pressure, as the flushes overlap a little. Since a single flush happens for 800 milliseconds and the delay is only for 300 milliseconds. So at any given time there can be 2 flushes happening at one time only.
6. When the 12 users request for a 1l flush at the same time, but they are serviced with a 300 millisecond delay between each flush. In this case at any given time there is only one flush and hence the pressure is 7.4 bar. This is due to the fact that the flush takes only 400 milliseconds to occur.
7. Here all the 12 users request for a flush at the same time, 6 of which request for a 3L flush and the remaining 6 request for a 1L discharge. In this case the, the pressure for the 3 liter flush drops to 6.2 bar and for the 1L flush drops to 2.1 bar. This is because when the 1L flush is requested, all the flushes are requested at the same time so the pressure for the 1l flush drops to 2.1 bar. However, once the one liter flush is complete the pressure for the 3 liter flush is increased to 6.2 bar, as there will be only 6 toilets working at the same time.
8. Here all the 12 users request for a flush at the same time, 6 of which request for a 3L flush and the remaining 6 request for a 1L discharge. However, each of the

flushes will experience a delay of 300ms. The toilets with 3L discharge request flush at 7.3 bar pressure and the ones with 1L request flush at 7.4 bar pressure.

7.4 Comparison with the Physical System Results

This section gives a brief comparison between the results obtained from the physical setup and the simulation. The pressure with which the experimental setup is run is 8 bar. However, in the simulation, the pressure drops down between 7 bar ad 8 bar. The Figure 7-19 shows the volume of water flushed in the specified time interval.

It is observed, that the simulation model needs 400 milliseconds to flush approximately 1.3 liters of water, whereas the physical system needs 300 milliseconds to flush 1.45 liters of water. It is also observed, that the simulation model needs 800 milliseconds to flush approximately 3.2 liters of water, whereas the physical system needs 700 milliseconds to flush 3.1 liters of water.

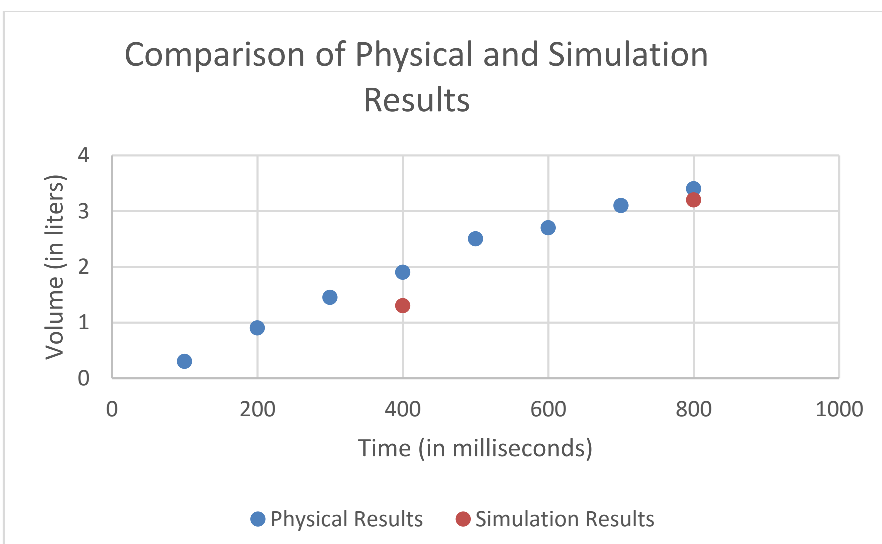

Figure 7- 19 Comparison of Results obtained from physical system and simulation.

7.5 Comparison with Flushmate™

The Figure 7-20 shows the comparison between the flow rate of flushmate and the flow rate of the proposed system obtained by simulation. It is observed that the proposed system flushes approximately 3.2 liters of water in 800 milliseconds and flushmate flushes approximately 6 liters of water in 5 seconds.

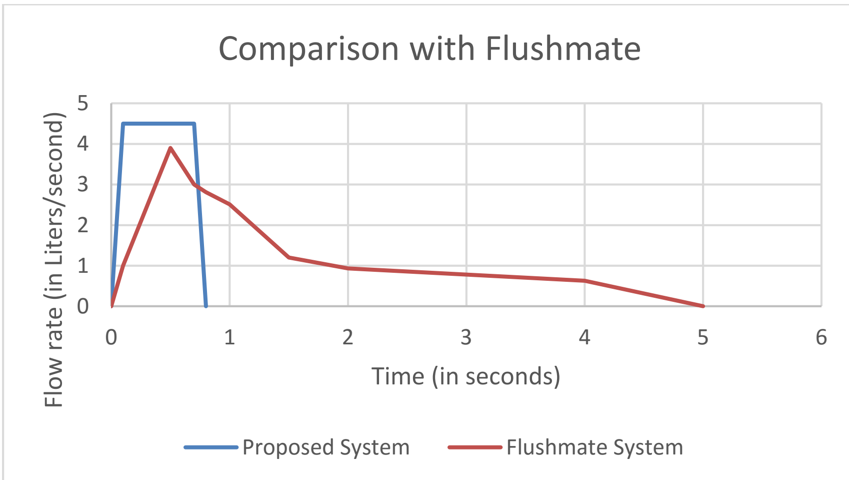

Figure 7- 20 Comparison of the flow rate of our system with flushmate.

CHAPTER 8

CONCLUSION AND FUTURE WORK

Old gravity siphon flushing systems consume about 4 to 6 liters of water per flush. When scarcity of water is taken into account this amount of water is considerably high and can reach 27% of domestic water usage [59]. Thus modern flushing toilets were modified to overcome this shortcoming, Flushmate improved the water amount to 3.8 liters per flush. This improvement is made possible by using confined air pressure to flush water rather than the gravity as in the older flush toilets. However, flushmate has only a single mode of operation, which implies irrespective of the waste to be flushed it always consumes 3.8 liters of water.

This thesis, investigates a new approach of toilet flushing system and can be applied in commercial Ultra Low Flush toilets solutions of Dual Flush type. This approach can use 1 liter and 3 liter of water for discharge for heavy and light flushing. The pressure of the water, volume and interval of discharge can be computer controlled to a high accuracy. The exact amount needed in flushing should be studied later as it impacts also the bowl design and flow hydrodynamics.

If a toilet flushes 13 liters per flush, one person can consume as much as 74 liters per day, or 27,010 liters per year. With flushmate, one person can consume as much as 20 liters per day, or 7,300 liters per year [60]. With the help of this approach, a person can consume as little as 12 liters of water per day or 4380 liters of water per year. This analysis shows that the proposed system saved 84% of water when compared to the old flushing systems and 36% of water when compared to flushmate.

This system can handle multiple flushing requests simultaneously as the there is one pressurized tank used for multiple toilets. This system is compact and can be placed even on the ground level as the water is already pressurized and can drain away the waste easily.

In the future, the toilet system can be installed at homes or a public space and real time data can be collected. The volume of the water flushed can also be varied further as well as the pressure with which the water should be flushed. Data such as the amount of water saved and the pressure of the water with which the flush occurs can be recorded. The user interaction and behavior can also be used to better analyze and confirm the simulation results.

APPENDICES

Appendix A: Images of the physical prototype

This appendix contains the images of the prototype which was built using the hardware mentioned in the Chapter 4.

Figure A- 1 Side View of the Pressurized Tank.

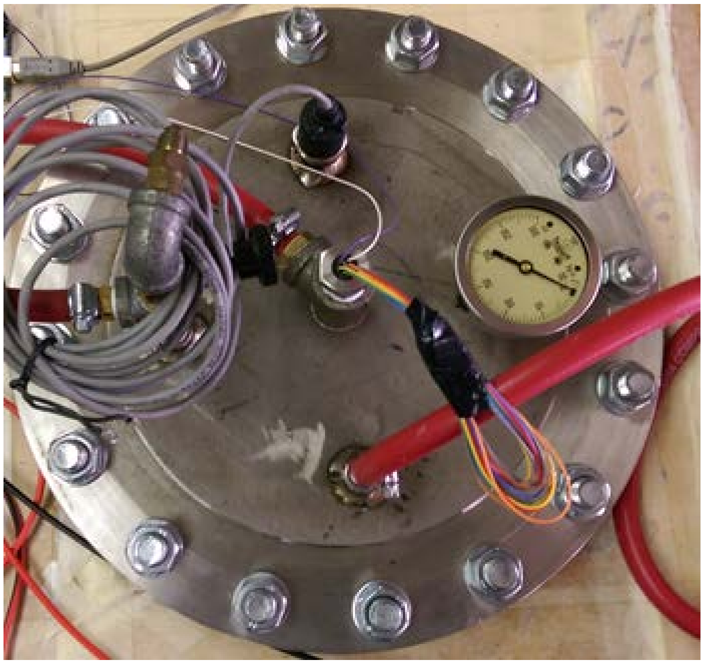

Figure A- 2 Top view of the pressurized tank.

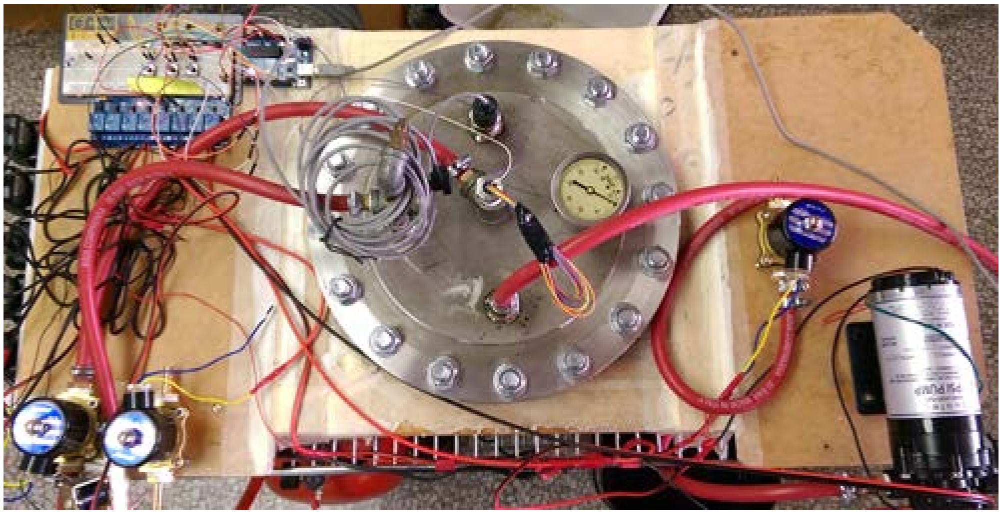

Figure A- 3 Top view of the prototype.

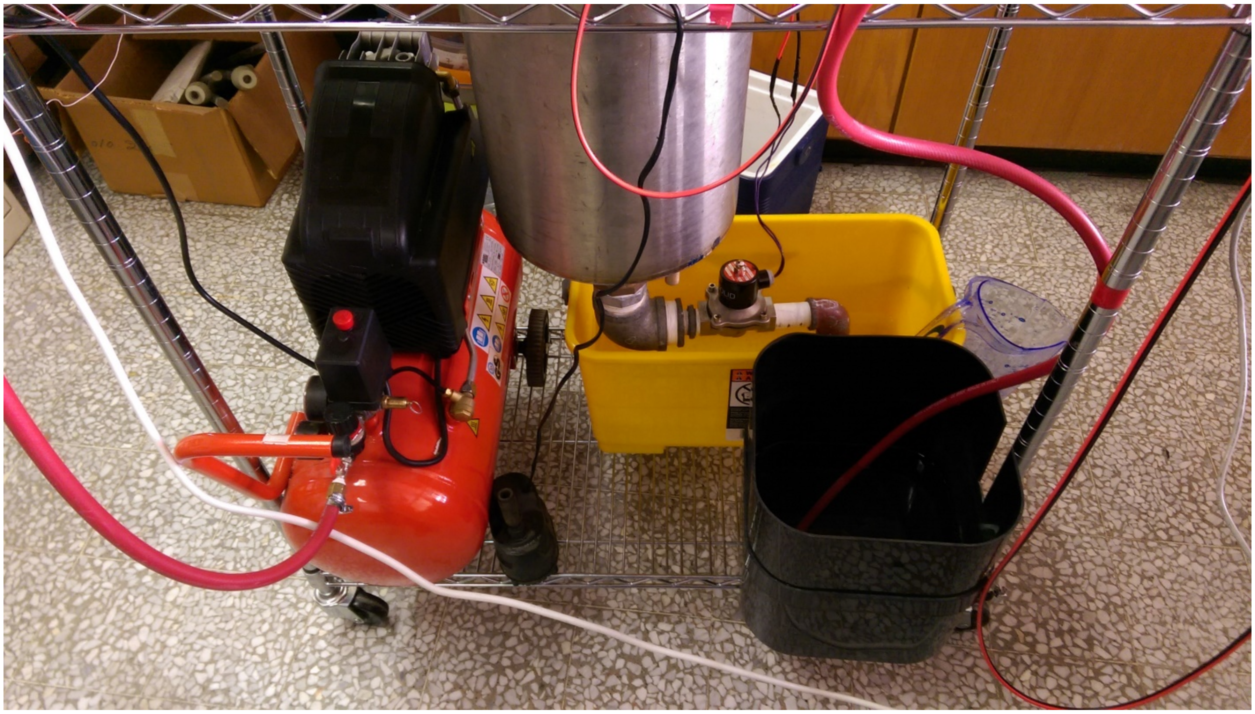

Figure A- 4 The air compressor, and the flushing container.

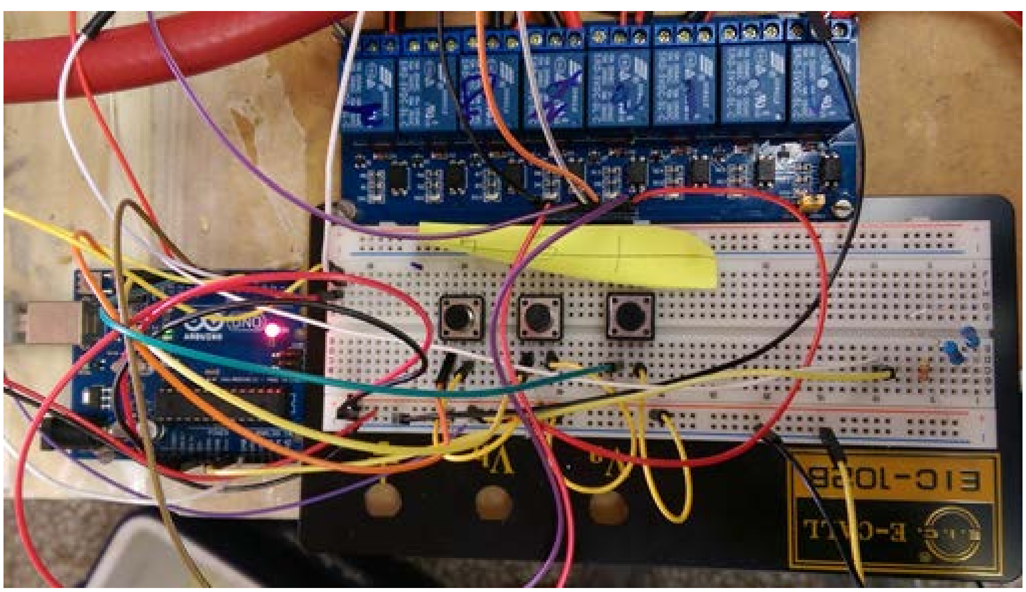

Figure A- 5 Arduino-Relay interface along with various switches.

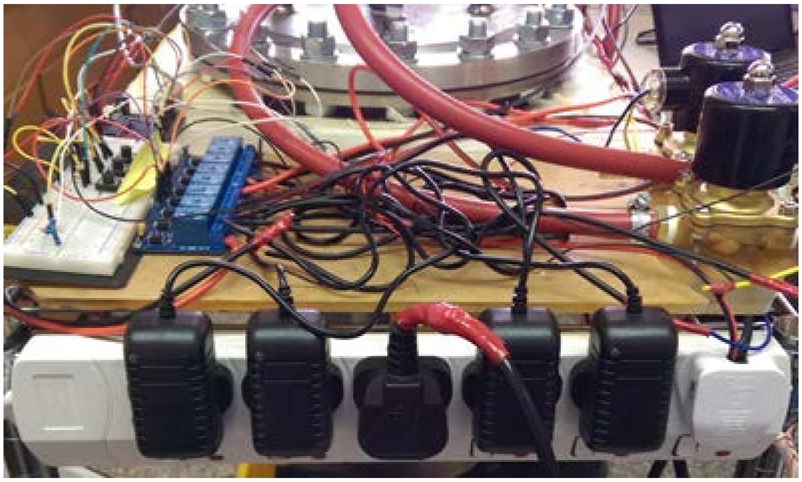

Figure A- 6 Different power supplies for solenoids and the air compressor.

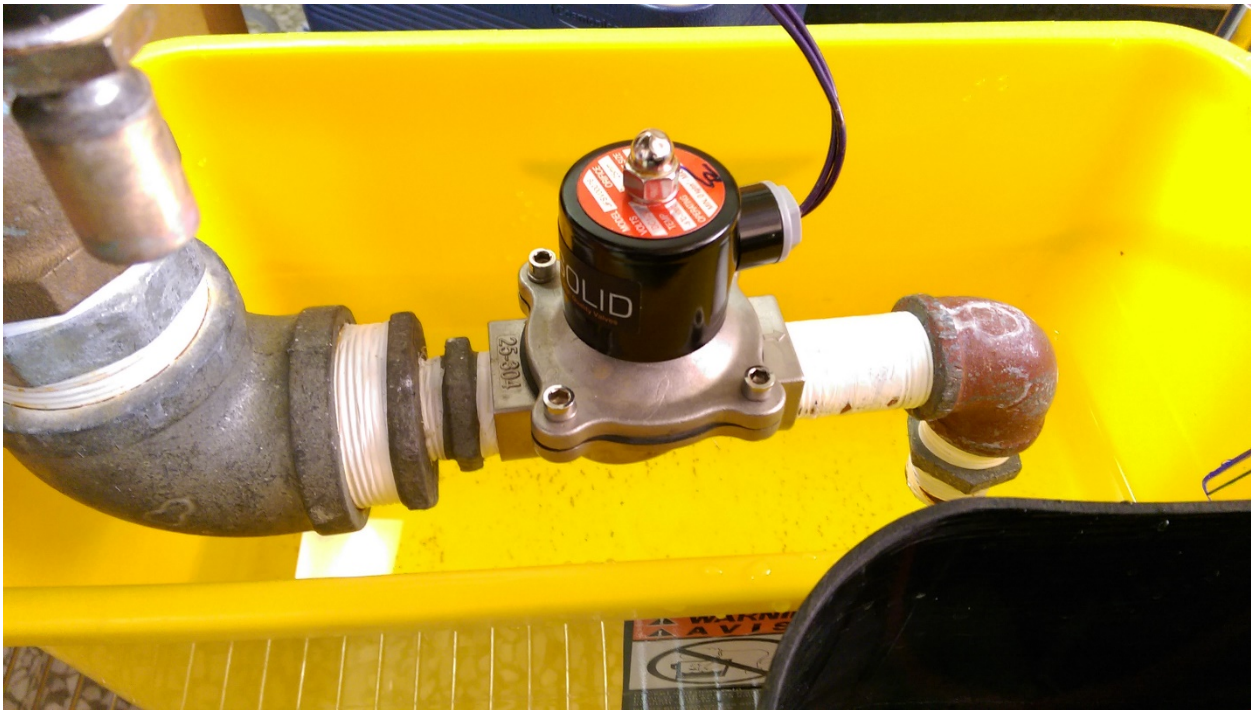

Figure A- 7 The Main Solenoid flushing valve.

Appendix B: Arduino Code for characterizing the stability of the system

```
// the value of the 'other' resistor
#define SERIESRESISTOR 2200
 // What pin to connect the sensor to
#define SENSORPIN A0
#define SENSORPRE A1
int S_AIR = 3;     // Solenoid 1
int S_WATER = 2;   // Solenoid 2
int S_EP = 8;      // Solenoid 3
int S_MAIN = 6;    // Main Solenoid
int SW_1 = 12;      // Button for Main Solenoid Valve
int SW_2 = 13;      // Button to release the pressure
int LS = A0;        // Liquid Sensor
int PS = A1;        // Pressure Sensor
// VARIABLES
float LS_ADC = 0, LS_IN = 0, PS_ADC = 0, PS_IN = 0;
void setup() {          // Defining Inputs and Outpus
pinMode(S_AIR, OUTPUT);
pinMode(S_WATER, OUTPUT);
pinMode(S_EP, OUTPUT);
pinMode(S_MAIN, OUTPUT);
pinMode(SW_1, INPUT);
pinMode(SW_2, INPUT);
pinMode(LS, INPUT);
pinMode(PS, INPUT);
```

```
Serial.begin(9600);
}
void loop()
{
LS_IN = LIQUID_SENSOR();        // Taking input from the liquid sensor
while (LS_IN >= 524)
{
  LS_IN = LIQUID_SENSOR();
  STATE_S0();
  }
digitalWrite(S_WATER,HIGH);
PS_IN = PRESSURE_SENSOR();    // Taking input from the pressure sensor
delay(100);
while (PS_IN <= 730)          // if pressure is less than 8 bar, pressurized the tank
{
  PS_IN = PRESSURE_SENSOR();
  STATE_S1();
  }
while (digitalRead(SW_1) == LOW)
{
  STATE_S2();
  }
STATE_S3();           //turn on the Main Solenoid for 3 second
while (digitalRead(SW_2) == LOW)
{
  STATE_S4();
  }
```

```
PS_IN = PRESSURE_SENSOR();

while (PS_IN >= 180)            // release extra pressure, until pressure becomes less than
Normal pressure which is normal pressure

{

  PS_IN = PRESSURE_SENSOR();

  STATE_S5();

  }

} // end loop

float LIQUID_SENSOR()

{

float LS_ADC = analogRead(LS);

delay(100);

return LS_ADC;

}

float PRESSURE_SENSOR()

{

float PS_ADC = analogRead(PS);

delay(100);

return PS_ADC;

}

void STATE_S0()

{

  DISPLAYVALUE ();

  digitalWrite(S_AIR,HIGH);

  digitalWrite(S_WATER,LOW);      // Turn on the Water Pump

  digitalWrite(S_EP,HIGH);

  digitalWrite(S_MAIN,HIGH);
```

```
  delay(100);
}
void STATE_S1()
{
  DISPLAYVALUE ();
  digitalWrite(S_AIR,LOW);
  delay(100);
}
void STATE_S2()
{
  DISPLAYVALUE ();
    digitalWrite(S_AIR,HIGH);
  delay(100);
}
void STATE_S3()
{
   DISPLAYVALUE ();
    digitalWrite(S_MAIN,LOW);
  delay(800);
    digitalWrite(S_MAIN,HIGH);
}
void STATE_S4()
{
  DISPLAYVALUE ();
  delay(100);
}
void STATE_S5()
```

```
{
  DISPLAYVALUE ();
  digitalWrite(S_EP,LOW);
  delay(100);
}
void DISPLAYVALUE ()
{
  /* Serial.print(analogRead(LS));    // display the voltage level on serial monitor
  Serial.print(',');
  Serial.println(analogRead(PS));    // display the pressure on serial monitor */
  float reading;
  float heightiinch;
  float heightcm;
  float volume;
   reading = analogRead(SENSORPIN);
   Serial.print(" Level Analog reading =  ");
  Serial.print(reading);
   // convert the value to resistance
  reading = (1023 / reading)  - 1;
  reading = SERIESRESISTOR / reading;
  Serial.print(" , Sensor resistance = ");
  Serial.print(reading);
  if ( reading > 2650 )
  {    Serial.print(" , Less than minimum Water Available ");   }
  else
  {
  heightiinch = (2810 - reading) / 160;
```

```
  heightcm = heightiinch * 2.54 ;
  volume = 3.14 * heightcm * 12.75 * 12.75;
  volume = volume / 1000 ;
  Serial.print(" , Height in cm = ");
  Serial.print(heightcm);
  Serial.print(" , Volume of Water = ");
  Serial.print(volume);
 // Serial.println("");
  }
   float value;
    value =  analogRead(SENSORPRE);
  Serial.print(" , Pressure Analog reading ");
  Serial.print(value);

if (value < 247.10)
 {   Serial.println(" , Normal Pressure");}
else
{
  value = (value * 5) / 1023 ;
  value = ( value * 2.84 ) - 2.32 ;
  Serial.print(" , Pressure in bar = ");
  Serial.println(value);
}
  delay(500);
}
```

Appendix C: Arduino Code for the Toilet flushing system with 8 bar pressure

```
// the value of the 'other' resistor
#define SERIESRESISTOR 2200
 // What pin to connect the sensor to
#define SENSORPIN A0
#define SENSORPRE A1
int S_AIR = 3;     // Solenoid 1
int S_WATER = 2;   // Solenoid 2
int S_MAIN = 6;    // Main Solenoid
int SW_1 = 12;      // Button for Main Solenoid Valve 3L Flush
int SW_3 =  7;        // Button for Main Solenoid Valve 1L Flush
int LS = A0;        // Liquid Sensor
int PS = A1;        // Pressure Sensor
int count = 0;
// VARIABLES
float LS_ADC = 0, LS_IN = 0, PS_ADC = 0, PS_IN = 0;
void setup() {          // Defining Inputs and Outpus
pinMode(S_AIR, OUTPUT);
pinMode(S_WATER, OUTPUT);
pinMode(S_MAIN, OUTPUT);
pinMode(SW_1, INPUT);
pinMode(LS, INPUT);
pinMode(PS, INPUT);
Serial.begin(9600);
}
```

```
void loop()
{
LS_IN = LIQUID_SENSOR();        // Taking input from the liquid sensor
while (LS_IN >= 524)
{  LS_IN = LIQUID_SENSOR();
  STATE_S0();  }
digitalWrite(S_WATER,HIGH);
if (count == 0)
{  PS_IN = PRESSURE_SENSOR();    // Taking input from the pressure sensor
delay(100);
while (PS_IN <= 730)
{  PS_IN = PRESSURE_SENSOR();
  STATE_S1();  }
  count = 1;
}
while ((digitalRead(SW_1) == LOW) && (digitalRead(SW_3) == LOW))
{  STATE_S2();  }
  if ( (digitalRead(SW_1) == HIGH))
 {
   STATE_S3A();   //turn on the Main Solenoid for 3L fush
 }
else
 {
   STATE_S3B();   //turn on the Main Solenoid for 1L flush
 }
delay(5000); //Added for delay
} // end loop
```

```
float LIQUID_SENSOR()
{
float LS_ADC = analogRead(LS);
delay(100);
return LS_ADC;
}
float PRESSURE_SENSOR()
{
float PS_ADC = analogRead(PS);
delay(100);
return PS_ADC;
}
void STATE_S0()
{
  DISPLAYVALUE ();
    digitalWrite(S_AIR,HIGH);
  digitalWrite(S_WATER,LOW);      // Turn on the Water Pump
  digitalWrite(S_MAIN,HIGH);
  delay(100);
}
void STATE_S1()
{
  DISPLAYVALUE ();
  digitalWrite(S_AIR,LOW);
  delay(100);
}
```

```
void STATE_S2()
{
 DISPLAYVALUE ();
 digitalWrite(S_AIR,HIGH);
 delay(100);
}
void STATE_S3A()
{
  DISPLAYVALUE ();
   digitalWrite(S_MAIN,LOW);
 delay(700); // -----For 3L flush
   digitalWrite(S_MAIN,HIGH);
}
void STATE_S3B()
{
  DISPLAYVALUE ();
   digitalWrite(S_MAIN,LOW);
 delay(200); // --------For 1L Flush
   digitalWrite(S_MAIN,HIGH);
}
void DISPLAYVALUE ()
{
 /* Serial.print(analogRead(LS));    // display the voltage level on serial monitor
 Serial.print(',');
 Serial.println(analogRead(PS));    // display the pressure on serial monitor */
 float reading;
 float heightiinch;
```

```
float heightcm;
float volume;
 reading = analogRead(SENSORPIN);
 Serial.print(" Level Analog reading =  ");
Serial.print(reading);
 // convert the value to resistance
reading = (1023 / reading)  - 1;
reading = SERIESRESISTOR / reading;
Serial.print(" , Sensor resistance = ");
Serial.print(reading);
if ( reading > 2650 )
{
  Serial.print(" , Less than minimum Water Available ");
}
else
{
heightiinch = (2810 - reading) / 160;
heightcm = heightiinch * 2.54 ;
volume = 3.14 * heightcm * 12.5 * 12.5 ;
volume = volume / 1000 ;
Serial.print(" , Height in cm = ");
Serial.print(heightcm);
Serial.print(" , Volume of Water = ");
Serial.print(volume);
// Serial.println("");
}
 float value;
```

```
    value =  analogRead(SENSORPRE);
  Serial.print(" , Pressure Analog reading ");
  Serial.print(value);

if (value < 247.10)
 {
   Serial.println(" , Normal Pressure");
}
else
{
  value = (value * 5) / 1023 ;
  value = ( value * 2.84 ) - 2.32 ; //2.43 datasheet , 2.22 to make 8 bar
  Serial.print(" , Pressure in bar = ");
  Serial.println(value);
}
  delay(1000);
}
```

Appendix D: State Transition Table

The following Table gives the State Transitions in the Design made in Chapter 4, Appendix 4.1.

Table A- 1 State Transition Table

S.N		Inputs				Output Signals					Comment
	C.S.	L	P	S1	S2	S_Water	S_Air	S_Main	S_EP	N.S.	
1	S0	0	0	0	0	1	0	0	0	S0	The system is in State S0, while monitoring the level sensor input (L) as 0.
2	S0	0	0	0	1	1	0	0	0	S0	
3	S0	0	0	1	0	1	0	0	0	S0	
4	S0	0	0	1	1	1	0	0	0	S0	
5	S0	0	1	0	0	1	0	0	0	S0	
6	S0	0	1	0	1	1	0	0	0	S0	
7	S0	0	1	1	0	1	0	0	0	S0	
8	S0	0	1	1	1	1	0	0	0	S0	
9	S0	1	0	0	0	0	0	0	0	S1	The system moves from S0 to S1 as soon as the Level sensor indicates a High.
10	S0	1	0	0	1	0	0	0	0	S1	
11	S0	1	0	1	0	0	0	0	0	S1	
12	S0	1	0	1	1	0	0	0	0	S1	
13	S0	1	1	0	0	0	0	0	0	S0	The system stays in State S0 as there is an inconsistency between P and the State.
14	S0	1	1	0	1	0	0	0	0	S0	
15	S0	1	1	1	0	0	0	0	0	S0	
16	S0	1	1	1	1	0	0	0	0	S0	
17	S4	0	0	0	0	0	0	0	0	S0	P indicates that there is no pressure which is inconsistent with the State
18	S4	0	0	0	1	0	0	0	0	S0	
19	S4	0	0	1	0	0	0	0	0	S0	
20	S4	0	0	1	1	0	0	0	0	S0	
21	S4	0	1	0	0	0	0	0	0	S4	S2 is not pressed
22	S4	0	1	0	1	0	0	0	0	S5	S2 is pressed
23	S4	0	1	1	0	0	0	0	0	S4	S2 is not pressed
24	S4	0	1	1	1	0	0	0	0	S5	S2 is pressed
25	S4	1	0	0	0	0	0	0	0	S0	L indicates there is water in the tank which is inconsistent with the state.
26	S4	1	0	0	1	0	0	0	0	S0	
27	S4	1	0	1	0	0	0	0	0	S0	
28	S4	1	0	1	1	0	0	0	0	S0	
29	S4	1	1	0	0	0	0	0	0	S0	

30	S4	1	1	0	1	0	0	0	0	S0	
31	S4	1	1	1	0	0	0	0	0	S0	
32	S4	1	1	1	1	0	0	0	0	S0	
33	S2	0	0	0	0	0	0	0	0	S0	Indicates no water and no pressure in the system, which is inconsistent.
34	S2	0	0	0	1	0	0	0	0	S0	
35	S2	0	0	1	0	0	0	0	0	S0	
36	S2	0	0	1	1	0	0	0	0	S0	
37	S2	0	1	0	0	0	0	0	0	S0	
38	S2	0	1	0	1	0	0	0	0	S0	
39	S2	0	1	1	0	0	0	0	0	S0	
40	S2	0	1	1	1	0	0	0	0	S0	
41	S2	1	0	0	0	0	0	0	0	S0	
42	S2	1	0	0	1	0	0	0	0	S0	
43	S2	1	0	1	0	0	0	0	0	S0	
44	S2	1	0	1	1	0	0	0	0	S0	
45	S2	1	1	0	0	0	0	0	0	S2	No Flush Requested (S1 not Pressed)
46	S2	1	1	0	1	0	0	0	0	S2	
47	S2	1	1	1	0	0	0	0	0	S3	Indicates Flush Requested (S1 Pressed)
48	S2	1	1	1	1	0	0	0	0	S3	
49	S1	0	0	0	0	0	0	0	0	S0	L Indicates no water in the tank, which is inconsistent with the State.
50	S1	0	0	0	1	0	0	0	0	S0	
51	S1	0	0	1	0	0	0	0	0	S0	
52	S1	0	0	1	1	0	0	0	0	S0	
53	S1	0	1	0	0	0	0	0	0	S0	
54	S1	0	1	0	1	0	0	0	0	S0	
55	S1	0	1	1	0	0	0	0	0	S0	
56	S1	0	1	1	1	0	0	0	0	S0	
57	S1	1	0	0	0	0	1	0	0	S1	P indicates that there is no threshold Pressure and the Air Compressor is On.
58	S1	1	0	0	1	0	1	0	0	S1	
59	S1	1	0	1	0	0	1	0	0	S1	
60	S1	1	0	1	1	0	1	0	0	S1	
61	S1	1	1	0	0	0	0	0	0	S2	P Indicates the Tank has Water and Is Pressurized.
62	S1	1	1	0	1	0	0	0	0	S2	
63	S1	1	1	1	0	0	0	0	0	S2	
64	S1	1	1	1	1	0	0	0	0	S2	
65	S5	0	0	0	0	0	0	0	0	S0	P and L indicate there is No Water and No Pressure in the Tank.
66	S5	0	0	0	1	0	0	0	0	S0	
67	S5	0	0	1	0	0	0	0	0	S0	
68	S5	0	0	1	1	0	0	0	0	S0	
69	S5	0	1	0	0	0	0	0	1	S5	P indicates there is Pressure in the Tank which has to be
70	S5	0	1	0	1	0	0	0	1	S5	

71	S5	0	1	1	0	0	0	0	1	S5	Released.
72	S5	0	1	1	1	0	0	0	1	S5	
73	S5	1	0	0	0	0	0	0	0	S0	L indicates there is water in the tank which is inconsistent with the state.
74	S5	1	0	0	1	0	0	0	0	S0	
75	S5	1	0	1	0	0	0	0	0	S0	
76	S5	1	0	1	1	0	0	0	0	S0	
77	S5	1	1	0	0	0	0	0	0	S0	
78	S5	1	1	0	1	0	0	0	0	S0	
79	S5	1	1	1	0	0	0	0	0	S0	
80	S5	1	1	1	1	0	0	0	0	S0	
81	S3	0	0	0	0	0	0	0	0	S0	P Indicates the Tank is not Pressurized which is inconsistent with the System.
82	S3	0	0	0	1	0	0	0	0	S0	
83	S3	0	0	1	0	0	0	0	0	S0	
84	S3	0	0	1	1	0	0	0	0	S0	
85	S3	0	1	0	0	0	0	0	0	S4	L Indicates Water in the Tank is completely Flushed.
86	S3	0	1	0	1	0	0	0	0	S4	
87	S3	0	1	1	0	0	0	0	0	S4	
88	S3	0	1	1	1	0	0	0	0	S4	
89	S3	1	0	0	0	0	0	0	0	S0	P Indicates the Tank is not Pressurized which is inconsistent with the System.
90	S3	1	0	0	1	0	0	0	0	S0	
91	S3	1	0	1	0	0	0	0	0	S0	
92	S3	1	0	1	1	0	0	0	0	S0	
93	S3	1	1	0	0	0	0	1	0	S3	L Indicates Water in the Tank is not completely Flushed.
94	S3	1	1	0	1	0	0	1	0	S3	
95	S3	1	1	1	0	0	0	1	0	S3	
96	S3	1	1	1	1	0	0	1	0	S3	

Appendix E: Code for generating the Pressurized Tank in Simulink

```
component LinearAirWaterTank
% Linear AirWater Pressurized Tank
% Gate A: Water Inlet
% Gate B: Water Outlet
% Gate C: Air Inlet
% Gate D: Air Outlet
 nodes
    A = foundation.hydraulic.hydraulic; % A:left
    B = foundation.hydraulic.hydraulic; % B:right
    C = foundation.pneumatic.pneumatic; % C:left
    D = foundation.pneumatic.pneumatic; % D:right
end
 variables
   q = { 1, 'm^3/s' };
   q2 = { 0, 'm^3/s' };
   G = { 1, 'kg/s'};
   G2 = { 0, 'kg/s'};

   p = { 0, 'Pa' };
   pGas = { 0, 'Pa' };
   pHydraulic = { 0, 'Pa' };

   gMass = {0, 'kg'};
   hVolume = {0, 'm^3'};
end
 inputs
    inControl = { 0, '1'};
 end
 outputs
    outVh = {0, 'm^3'}; %outVh: left
   outMg = {0, 'kg'}; %outMg: left
   outVg = {0, 'm^3'}; %outVg: left
   outRt = {0, '1'}; %outRt: left
    outPGas = { 0, 'Pa' };
    outPV = { 0, 'N*m' };
 end
 parameters
   % gas/hydraulic tank parameters
   SampleTime = {0.1, 's'}; %Sample Time
   AreaTank = {0.2, 'm^2'}; %Tank cross-section area
   VolumeTank = {1 ,'m^3'}; %Tank volume
   hCriticalLevel = {0.105, '1'} % Critical Hydraulic Level
   hMinimumLevel = {0.1, '1'} %Tank volume fraction always filled with hydraulic fluid.
   gMinimumLevel = {0.1, '1'} %Tank volume fraction always filled with gas fluid.
```

```
  Po = {8e5, 'Pa'}; %initial pressurization
  Rt = {4, '1'}; %Gas/Hydraulic Volume Ratio.
  gMo = {7.63, 'kg'}; %Initial Gas Mass
  hVo = {0.15, 'm^3'}; %Initial Hydraulic Volume
  hmExchangeRate = {0.2, 'Hz'} %Rate of exchange in trapped gas into Hydraulic Fluid.
  Hba = {0.5,'m'}; %Distance relative from gate B to gate A.
  Da = { 0.5, 'in'}; % Port A pipeline diameter
  KlossGa = { 0.001, '1' }; % Port A pressure loss coefficient
  Db = { 1, 'in'}; % Port B pipeline diameter
  KlossGb = { 20, '1' }; % Port B pressure loss coefficient
  cDischargeB = { 0.6, '1'}; % Outlet gate Discharge
  Dc = { 0.5, 'in'}; % Port C pipeline diameter
  KlossGc = { 0.001, '1' }; % Port C pressure loss coefficient
  Dd = { 1, 'in'}; % Port D pipeline diameter
  KlossGd = { 0.001, '1' }; % Port D pressure loss coefficient
  AllowanceFactor = {5, '1'} % Allowance Factor
end

parameters(Hidden=true)
   % Internal Variables
   Mg = { 2.8964e-2, 'kg/mol'}; %Dried Air Molecular Mass
   gravity = { 9.8, 'm/s^2' };
   R = { 8.3144598, 'm^3*Pa/mol/K'};
   T = {293.15, 'K'}
   % Converters Variables
   GasConverter = {0.4, 'm^3/kg'};
   TimeConverter = {1, 's'};
   AreaConverter = {6.4516e-4, 'm^2/in^2'};
   GasFlowConverter = { 1, 'kg/m^3'};
   HydraulicPressureConverter = { 1e5, 'Pa/bar'}
   ReferencePressure = { 0, 'Pa'};
   AirCompressorConverter = { 1, 'N*m/Pa'};
   FlowRateConverter = {1, 's/m'};
   HydraulicZeroFlow = {1, 'm^3/s'};
   ZeroPressure = {0, 'Pa'};
   ValueOne = {1, '1'};
end
function setup
   through( q, A.q, [] );
   through( q2, [], B.q );
   through( G, C.G, [] );
   through( G2, [], D.G );
   across( pHydraulic, A.p, B.p);
   across( pGas, C.p, D.p);

   if Rt < 0
```

```
        error ( 'Ratio of gas/hydraulic must be greater than 0');
    end
end
  equations

    let
        Aa = ((pi*Da^2)/4)*AreaConverter; % Area Gate A
        Ab = ((pi*Db^2)/4)*AreaConverter; % Area Gate B
        Ac = ((pi*Dc^2)/4)*AreaConverter; % Area Gate C
        Ad = ((pi*Dd^2)/4)*AreaConverter; % Area Gate D
        HydraulicVolume = if time > SampleTime, delay(hVolume,SampleTime) else hVo
end;
        gVolume = (VolumeTank - HydraulicVolume);
    in
         let
            H = HydraulicVolume / AreaTank;
            gRelativeDensity = gMass / gVolume;
            N = gMass / Mg;
            Kg = N * R * T;
        in
            let
                Pg = Kg / gVolume;
            in
                let
                    %PressureToFlowRate = if time > 0, Po + delay(Pg,SampleTime) else (Po +
Pg) end;
                    PressureToFlowRate = Po + Pg;
                    hMinLevel = hMinimumLevel*VolumeTank;
                    hCriticalVol = hCriticalLevel*VolumeTank;
                    hMaxLevel = if Rt < 1, VolumeTank*(Rt/(Rt+1)) else VolumeTank/(1 + Rt)
end;
                    hFlowAllowance = if HydraulicVolume < hCriticalVol,
ValueOne/AllowanceFactor else ValueOne end;
                in
                    outRt == gVolume / HydraulicVolume;
                    outVg == gVolume;
                 if inControl > 0
                         q2 == hFlowAllowance  * inControl * cDischargeB* AreaTank *
sqrt((2*(gravity*H + PressureToFlowRate/B.density)));
                         G2 == hmExchangeRate * inControl * gMass;
                      else
                         q2 == 0;
                         G2 == 0;
                     end
                      % Defining pressure.
                    let
```

```
                GasFlowIn = G * GasConverter * SampleTime / TimeConverter;
                GasFlowOut = G2 * GasConverter * SampleTime / TimeConverter;
                HydraulicFlowIn = q * SampleTime / TimeConverter;
                HydraulicFlowOut = q2 * SampleTime / TimeConverter;
             in
                let
                   PlossGa = KlossGa*(A.density * HydraulicFlowIn *
abs(HydraulicFlowIn)) / (2*Aa^2);
                   PlossGb = KlossGb*(A.density * HydraulicFlowOut *
abs(HydraulicFlowOut)) / (2*Ab^2);
                   PlossGc = KlossGc*(gRelativeDensity * GasFlowIn * abs(GasFlowIn))
/ (2*Ac^2);
                   PlossGd = KlossGd*(gRelativeDensity * GasFlowOut *
abs(GasFlowOut)) / (2*Ad^2);
                in
                   let
                      PlossGate = PlossGa + PlossGb + PlossGc + PlossGd;
                      PlossDifLevel = A.density*gravity*Hba;
                   in
                      let
                         Ph = A.density*gravity*H;
                         Ploss = PlossGate + PlossDifLevel;
                      in
                         p == Ph + Po + Pg - Ploss;
                         pHydraulic == p;
                         pGas == Po + Pg;
                         outPGas == pGas;
                         outPV == Kg;
                      end
                   end
                end
             end
          end
       end
    end
  end
  let
    hMinLevel = hMinimumLevel*VolumeTank;
    hMaxLevel = if Rt < 1, VolumeTank*(Rt/(Rt+1)) else VolumeTank/(1 + Rt) end;
    gMinMass = 7.63;
    gMaxMass = 7.63;
    hPreviousVolume = if time > SampleTime, delay(hVolume,SampleTime) else hVo
end;
    gPreviousMass = 7.63;

  in
```

```
      let
      %    hCurrentVolume = if time > 0, if inControl > 0, (q - delay(q2,SampleTime)) *
SampleTime else q * SampleTime end else q * SampleTime end;
         hCurrentVolume = if time > 0, if inControl > 0, (q - q2) * SampleTime else q *
SampleTime end else q * SampleTime end;
         gCurrentMass = 7.63;
      in

         if (hCurrentVolume + hPreviousVolume) > hMaxLevel
            hVolume == hMaxLevel;
         else
            if (hCurrentVolume + hPreviousVolume) < hMinLevel
               hVolume == hMinLevel;
            else
               hVolume == hCurrentVolume + hPreviousVolume;
            end
         end;

         if (gCurrentMass + gPreviousMass) > gMaxMass
            gMass == gMo;
         else
            if (gCurrentMass + gPreviousMass) < gMinMass
               gMass == gMo;
            else
               gMass == gMo;
            end
         end;

         outMg == gMass;
         outVh == hVolume;

      end
   end
end
 end
```

Appendix F: Matlab Code for analyzing the pressure, volume and interval of discharge

```
% ------------------------ STATEMENT --------------------------- %
% Purpose Statement:
%   Analyse water flush simulation and show its results.
% ------------------------- CODE --------------------------- %


close all;
addpath('C:\Users\User\Documents\MATLAB');
if (exist('FlowMeter1') && exist('PressureMeter1'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter1, PressureMeter1), '1'); end;
if (exist('FlowMeter2') && exist('PressureMeter2'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter2, PressureMeter2), '2'); end;
if (exist('FlowMeter3') && exist('PressureMeter3'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter3, PressureMeter3), '3'); end;
if (exist('FlowMeter4') && exist('PressureMeter4'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter4, PressureMeter4), '4'); end;
if (exist('FlowMeter5') && exist('PressureMeter5'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter5, PressureMeter5), '5'); end;
if (exist('FlowMeter6') && exist('PressureMeter6'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter6, PressureMeter6), '6'); end;
if (exist('FlowMeter7') && exist('PressureMeter7'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter7, PressureMeter7), '7'); end;
if (exist('FlowMeter8') && exist('PressureMeter8'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter8, PressureMeter8), '8'); end;
if (exist('FlowMeter9') && exist('PressureMeter9'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter9, PressureMeter9), '9'); end;
if (exist('FlowMeter10') && exist('PressureMeter10'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter10, PressureMeter10), '10'); end;
if (exist('FlowMeter11') && exist('PressureMeter11'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter11, PressureMeter11), '11'); end;
if (exist('FlowMeter12') && exist('PressureMeter12'))
PlotFlushAnalysis(FlushAnalysis2(FlowMeter12, PressureMeter12), '12'); end;
```

Appendix G: Matlab Code for analyzing the flush

```
% ------------------------- STATEMENT ---------------------------- %
% Purpose Statement:
%   Create information, based upon previous data simulation, to answer
% required questions about water flush system, e.g. volume of water
% discharged, time of discharge and pressure of discharge. This function
% will return information only in the instants that flow rate surplus
% threshold (flush detection).
% ------------------------- CODE ---------------------------- %

function result = FlushAnalysis(Flow, Pressure)

   FLOW_THRESHOLD = 0.01;

   % Variables Inicializing Block
   intCountAcumulator = 0;
   intCountResult = 0;
   dblAcumulator = 0.0;
   dblFlushTime = 0.0;
   dblFlushStart = 0.0;
   dblFlowS = 0.0;
   dblFlowE = 0.0;
   dblMedianPressure = 0.0;
   dblMedianFlowRate = 0.0;
   FlowRateAcumulator = [];
   PressureAcumulator = [];

   % Getting the total size of the simulation
   if size(Flow.time) <= size(Pressure.time)
      intSampleSize = size(Flow.time);
   else
      intSampleSize = size(Pressure.time);
   end;

   for intCount = 1:intSampleSize

      if Flow.signals.values(intCount) > FLOW_THRESHOLD

         %Seeking to create an acumulator to measure up
         %the median FlowRate and Pressure during water dicharge.

         if dblFlushStart == 0.0
            dblFlushStart = Flow.time(intCount);
```

```
        end

        intCountAcumulator = intCountAcumulator + 1;
        FlowRateAcumulator(intCountAcumulator) = Flow.signals.values(intCount);
        PressureAcumulator(intCountAcumulator) = Pressure.signals.values(intCount);
      else

        if intCountAcumulator > 0

          % Calculating time of discharge
          dblFlowS = dblFlushStart;
          dblFlowE = Flow.time(intCount);

          % Calculating pressure of discharge
          dblMedianPressure = str2num(num2str(median(PressureAcumulator),2));


          % Calculating water volume of discharge
          dblMedianFlowRate = median(FlowRateAcumulator);

          %dblVolume = fix(str2num(num2str((dblMedianFlowRate * dblFlush),2)));
          dblVolume = str2num(num2str((dblMedianFlowRate * (dblFlowE -
dblFlowS)),2));
          dblFlushTime = dblFlowE - dblFlowS;



          % Creating the information result vector  to each detectable
          % flush
          intCountResult = intCountResult + 1;
          result(intCountResult) =
struct('Time',dblFlushStart,'Interval',dblFlushTime,'Volume',dblVolume,'Pressure',
dblMedianPressure);
          dblFlushStart = 0;
          intCountAcumulator = 0;

        end
      end
    end
end
```

References

[1] Bremner J., Frost A. , Haub C. , Mather M. , Ringheim K. , Zuehlke E., " World Population Highlights: Key Findings From PRB’s 2010 World Population Data Sheet," Population Reference Bureau, 2010.

[2] "T. H. E. United and N. World, Managing Water under Uncertainty and Risk" [Online]. Available: http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/SC/pdf/WWDR4%20Volume%201-Managing%20Water%20under%20Uncertainty%20and%20Risk.pdf. [Accessed December 2015].

[3] A. A. ALUWAISHEG, "Runaway water consumption alarming for Saudi Arabia," Arab News, 2013. [Online]. Available: http://www.arabnews.com/news/460158.

[4] Seshia, E. A. Lee, "Introduction to Embedded Systems, A Cyber-Physical Systems Approach," in *National Science Foundation*, 2011.

[5] A. A. Al-Ibrahim, "Water Use in Saudi Arabia: Problems and Policy Implications," *Journal of Water Resources Planning and Management,* 1990.

[6] Erdmann, Wolfgang, Torsten Gienke, and Hans-Juergen Heinrich, "Waterless vacuum toilet system for aircraft". U.S. Patent 6,977,005, December 2005.

[7] "United States Environmental Protection Agency," [Online]. Available: http://www.epa.gov/watersense/products/toilets.html. [Accessed December 2015].

[8] FLUSHMATE™, "FLUSHMATE™ 1.6 and 1.0 Gallon Pressure Assist Toilet Operating System," [Online]. Available: http://www.flushmate.com(. [Accessed May 2015].

[9] MathWorks, "Malab," [Online]. Available: http://www.mathworks.com/.

[10] Zhaohui Wang, Houbing Song, David W. Watkins, "Cyber-Physical Systems for Water Sustainability:Challenges and Opportunities," 2015.

[11] Estrin, L. I. Millett, "Computing Research for Sustainability," Washington DC, 2012.

[12] E. A. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?," Austin, TX, 2011.

[13] D. Quaglia, "Cyber-physical systems: Modeling, simulation, design and validation," 2013.

[14] Jing Lin, Sahra Sedigh, and Ann Mille, "Towards Integrated Simulation of Cyber-Physical Systems: A Case Study onIntelligent Water Distribution," 2012.

[15] M. Suresh, "A cyber-physical system for continuous monitoring of Water Distribution Systems," 2014.

[16] M. Mohamed, W. Wu and M. Moniri, "Power harvesting for smart sensor networks in monitoring water distribution system," 2011.

[17] A. Nasir, " Framework of WSN based human centric cyber physical in-pipe water monitoring system," 2012.

[18] Yongsong Wei, Shaoyuan Li, "Water supply networks as cyber-physical systems and controllability analysis," 2015.

[19] L. Xifeng, N. Shujie and C. Shiyue, "The Technology Research of Water Quality Monitoring System Using Remote Sensing in Lakes," 2011.

[20] R. Evans and W. Iversen, "Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network," 2012.

[21] K. Kaur and L. JayPrakash, "Modeling and simulation of Cyber Physical Systems: A case of an adaptive water management system," 2014.

[22] A. Mathur, "Designing safe and secure cyber physical systems for water distribution," 2014.

[23] A. Jang et al, "State-of-the-Art Lab Chip Sensors for Environmental Water Monitoring," 2011.

[24] J. Wang, X.-l. Ren, Y.-l. Shen and Shuang-Yin Liu, "A Remote Wireless Sensor Networks for Water Quality Monitoring," 2010.

[25] X. Li and C. Jiao, "Water conservancy monitoring system based on wireless sensor network," 2011.

[26] S. Nandurkar, V. Thool and R. Thool, "Design and development of precision agriculture system using wireless sensor network," 2014.

[27] Palmer, Marilyn, and Ian West, "Comfort and Convenience at Audley End House," *English Heritage Historical Review 8.1,* pp. 82-103, 2013.

[28] Kravetz, Robert E., ""The Flush Toilet.," *The American journal of gastroenterology,* 2009.

[29] C. Eden, "Flushmate Vessel Welding Process Improvements".

[30] United States Environmment Protection Agency. [Online]. Available: http://www.epa.gov/watersense/products/toilets.html.

[31] Lute, Michelle L., "Dont rush to flush," *Journal of Environmental Psychology,* pp. 105-

111, 2015.

[32] Salman, Mohd Syafiq, "INVESTIGATING FLUSH SYSTEM METHOD FOR MINIMUM WATER USAGE OF TOILET SYSTEM," *Universiti Tun Hussein Onn Malaysia,* 2014.

[33] P. H. Gleick, "Basic water requirements for human activities," *Water International,* 1996.

[34] B. Inskeep, "The water short list: the most effective actions U.S. households can take to curb water use," *Environment: Science and Policy for Sustainable Development,* 2013.

[35] S. D. Tsao, "Behavioral avoidance tests and disgust in contamination fears," *Behavior Research and Therapy,* 2004.

[36] P. Singapore, "Public Utilities Board Singapore," 02 06 2014. [Online]. Available: https://app.pub.gov.sg/WaterFittingsUsers2/WaterWsFitting_WCFVReqts.aspx?fd=Requirements%20for%20WC%20flush%20valves&l1=2&l2=8&l3=4-2&l4=5. [Accessed 10 04 2016].

[37] S. Z. Attari, "Perceptions of water use," *Proceedings of the National Academy of Science of the United States of America,* pp. 5129-5134, 2014.

[38] S. Rachman, "Fear of contamination," *Behaviour Research and Therapy,* 2004.

[39] WHO, "First adapt then act! A booklet to promote safer food in diverse setting," *WHO,* 2004.

[40] J. Adams, "Water, Sanitation and Hygiene Standards," *WHO,* 2009.

[41] Arduino. [Online]. Available: https://www.arduino.cc/.

[42] "Arduino UNO," [Online]. Available: http://arduino.cc/en/main/arduinoBoardUno.

[43] Sainsmart. [Online]. Available: http://www.sainsmart.com/8-channel-dc-5v-relay-module-for-arduino-pic-arm-dsp-avr-msp430-ttl-logic.html. [Accessed December 2015].

[44] "Milone Technologies," [Online]. Available: http://milonetech.com/products/standard-etape. [Accessed November 2015].

[45] Milone Technologies, "Liquid level sensor," [Online]. Available: http://milonetech.com/products/standard-etape. [Accessed November 2015].

[46] Milone Technologies, "Datasheet," [Online]. Available: http://www.original.milonetech.com/uploads/Standard_eTape_Datasheet.pdf. [Accessed November 2015].

[47] Pace Secietific Data Loggers and Sensor, "P1600 Pressure Sensor," [Online]. Available: http://www.pace-sci.com/pressure-1600.htm. [Accessed 2015 December].

[48] Black and Decker, "Air Compressor," [Online]. Available: http://www.blackanddecker.co.uk/powertools/productdetails/catno/CP2525/partner/2/. [Accessed December 2015].

[49] Streambite, "Triplex diaphragm pressure pump," [Online]. Available: http://www.steambrite.com/aquatec-triplex-diaphragm-pressure-switched-bypass-pump-volts-mytee-c322d-p-10801.html. [Accessed December 2015].

[50] "PAVE Technologies, Electrical Feed Through," [Online]. Available: http://www.pavetechnologyco.com/design/pdf/1588.html. [Accessed 2016 January].

[51] Ying-bao Zhao, Xue-ling Song, Chao-ying Liu, Zhe-ying Song, " Solenoid Valve Respond Time Test System Basing on Single Chip Micro-computer," in *International Conference on Intelligent Human-Machine Systems and Cybernetics. IHMSC*, 2009.

[52] Amazon, "Stainless Solenoid Valve," [Online]. Available: http://www.amazon.com/Stainless-Steel-Electric-Solenoid-Valve/dp/B00DQ28R6S. [Accessed 2015 September].

[53] Amazon, "Plastic Solenoid Valve," [Online]. Available: http://www.amazon.com/12V-Plastic-Water-Solenoid-Valve/dp/B00NRH9VQW.

[54] PACE, "Pressure sensor datasheet," [Online]. Available: www.pace-sci.com/XR5-SE.pdf.

[55] I. G. Law, *Ideal Gas Law.*

[56] Matlab, "Mathwork Documentation," [Online]. Available: http://www.mathworks.com/help/physmod/hydro/ref/variableheadtank.html?searchHighlight=headed%20tank. [Accessed December 2015].

[57] Mathwork, "Pneumatic Domain Documentation," [Online]. Available: http://www.mathworks.com/help/physmod/simscape/lang/pneumatic-domain.html?searchHighlight=pneumatic%20domain. [Accessed 2015].

[58] Mathwork, "Hydraulic Domain Documentation," [Online]. Available: http://www.mathworks.com/help/physmod/simscape/lang/hydraulic-domain.html?searchHighlight=hydraulic%20domain.

[59] "Water Sense, An EPA partnership," United States Environmental Protection Agency, [Online]. Available: https://www3.epa.gov/watersense/pubs/indoor.html. [Accessed April 2016].

[60] Flushmate, "Document Library," [Online]. Available:

https://www.flushmate.com/documentlibrary/pdf/FLU6072-EN-Introducing-Flushmate-IV-For-High-Efficiency-Toilets.pdf.

Vitae

Name	: Wisam Mohamed Wahid
Nationality	: Indian
Date of Birth	:6/4/1990
Email	: wisamwahid@hotmail.com, g201310410@kfupm.edu.sa
Address	: Little Flower, Balmatta Road, Mangalore -575001, India
Academic Background	: Bachelors of Engineering in Electronics and Communication (June, 2012) from VT University, Mangalore, India.
Publication	: Wisam Mohamed Wahid, Jihad H. Alsadah, Esmail M. A. Mokheimer and Yahya E. Osais, "A Novel Microcontroller-based Pressurized Flushing System for Water Conservation in Smart Homes and Public Spaces", submitted to ACM Transactions on Cyber-Physical Systems (May 2016).