

A LOW COST WATER CONSUMPTION METER SYSTEM BASED ON GSM TECHNOLOGY

MWANGI PETER NGUGI

I56/24955/2013

A thesis submitted in partial fulfillment of the requirements for the award of the Degree of Master of Science (Electronics & Instrumentation) in the School of Pure and Applied Sciences of Kenyatta University.

OCTOBER 2016

DECLARATION

This thesis is my original work and has not been presented for the award of a degree or any other award in any University.

Mwangi Peter Ngugi

Signature Date

We confirm that the candidate carried out the work reported in this thesis under our supervision

Prof. Elijah Mwangi

School of Engineering

University of Nairobi/ Kenyatta University

Signature Date

Dr. Patrick Karimi

Institute of Energy and Research (IESR)

Kenya Power

Signature Date

DEDICATION

This work is dedicated to my loving wife, Purity, to my dear children, Precious and Peniel, who have been a source of encouragement in the pursuit of my dreams.

ACKNOWLEDGEMENTS

It gives me great pleasure to express my sincere gratitude to my supervisors Prof. Elijah Mwangi and Dr. Patrick Karimi for their invaluable guidance, motivation, constant inspiration and above all their ever co-operating attitude that enabled me come up with this thesis in the present form. My sincere thanks are also extended to all lecturers and laboratory technical staff in the Physics department for their support.

The most important part of my life is my family. Thanks to my lovely children Precious and Peniel who are a source of inspiration in the pursuit of my dreams. I would never have been able to accomplish any of my goals without the support my lovely wife Purity. I wish to thank her for love, encouragement and sacrifice throughout my study. Most importantly, I also wish to thank the almighty God for everything, for I believe that his great blessing has enabled me to make progression in my career.

TABLE OF CONTENTS

DECLARATION.....	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF FIGURES	ix
LIST OF TABLES	x
ABBREVIATIONS AND ACRONYMS	xii
ABSTRACT	xiv
CHAPTER ONE	1
INTRODUCTION.....	1
1.1 Background to the study.....	1
1.1.1 Displacement water meters.....	2
1.1.2 Velocity water meter.....	3
1.2 Meter Reading	4
1.3 Statement of the research problem.....	5
1.4 Objectives	6
1.4.1 General objective.....	6
1.4.2 Specific objectives	6
1.5 Rationale for the research.....	6
1.6 Scope of the study.....	7
1.7 Thesis Outline.....	7
1.8 Note on publication.....	8
CHAPTER TWO	9
LITERATURE REVIEW	9
2.1 Introduction	9
2.2 Automatic meter reading	9
2.3 Related works.....	9
2.4 Proposed investigation.....	12
CHAPTER THREE	13
THE PIC18F4550 MICROCONTROLLER	13
3.1 Introduction	13
3.2 The PIC18F4550 Microcontroller chip.....	13

3.2.1 PIC18F4550 Architecture	14
3.2.2 Data Memory	15
3.2.3 Programme Memory	16
3.2.4 Programme stack memory	18
3.3 Registers	19
3.3.1 EECON1 AND EECON2 Registers	19
3.3.2 Table Latch (TABLAT) Register	20
3.3.3 Table Pointer (TBLPTR) Register	20
3.4 I/O PORTS	21
3.4.1 PORTA	22
3.4.2 PORTB	22
3.4.3 PORTC	23
3.4.4 PORTD	23
3.4.5 PORTE	24
3.5 Interrupts	24
3.6 Timer module	26
3.6.1 Timer0 module	26
3.6.2 Timer1 module	28
3.6.3 Timer2 module	30
3.6.4 Timer3 Module	31
3.6.5 Watchdog Timer	33
3.7 Oscillators	33
3.7.1 Oscillator control	34
3.7.2 Oscillator types	34
3.8 EUSART	35
3.8.1 Registers in the EUSART	35
3.8.2 EUSART asynchronous mode	37
3.8.3 EUSART synchronous master mode	37
3.8.4 EUSART Synchronous Slave Mode	37
CHAPTER FOUR	38
TECHNICAL CHOICE OF THE SYSTEM SOFTWARE AND HARDWARE	38
4.1 Introduction	38
4.2 Mikroc Integrated Development Environment	38
4.3 PICkit2 programmer	39

4.4 Power supply	40
4.4.1 Transformer	41
4.4.2 Rectification	42
4.4.3 Capacitor filter.....	42
4.4.4 Regulation.....	43
4.5 Liquid Crystal Displays (LCDs)	43
4.6 Global System for Mobile Communications (GSM)	46
4.6.1 GSM Network Architecture	46
4.6.2 Mobile Station (MS)	47
4.6.2.1 Mobile Equipment (ME)	47
4.6.2.2 Base Transceiver Station (BTS)	47
4.6.3 Base Station Controller (BSC)	48
4.6.4 Mobile Switching Center (MSC)	48
4.7 Flow sensors and measurements	49
4.7.1 Flow measurement	49
4.7.2 Turbine Flow meter.....	50
4.7.3 Flow sensor	49
4.8 Solenoid valve	51
CHAPTER FIVE	53
RESEARCH METHODOLOGY	53
5.1 Introduction	53
5.2 Hardware components	53
5.2.1 Flow sensor.....	54
5.2.2 Liquid Crystal Display to PIC 18F4550 microcontroller interface.....	55
5.2.3 GSM interface to PIC18F4550 microcontroller	57
5.2.4 Solenoid control valve	58
5.3 Software design	60
5.3.1 Flowchart for the system software	60
5.3.2 Programme code development	62
CHAPTER SIX	66
RESULTS AND DISCUSSIONS	66
6.1 Introduction	66
6.2 Programme Test	66

6.3 System response to water flow at different frequencies	67
6.4 Flow rate calculation.....	67
6.5 The GSM Module	70
6.6 Control valve	71
6.7 Regulated power supply	72
6.8 Testing the entire system and comparing to other systems	73
CHAPTER SEVEN	75
CONCLUSION AND RECOMMENDATIONS.....	75
7.1 Conclusion.....	75
7.2 Recommendations for future work.	75
REFERENCES.....	77
APPENDICES.....	80
Appendix A: Architecture of PIC18F4550.....	80
Appendix B: Materials employed in the research	81
Appendix C :Schematics of the water meter system.....	82
Appendix D: Data sheet	83
Appendix E: Published paper	84
Appendix F: The programme code.....	90

LIST OF FIGURES

Figure 1.1: PIC18F4550 pin diagram configuration	2
Figure 1.2: PIC18F4550 with harvard architecture.....	4
Figure 3.1: PIC18F4550 pin diagram configuration	14
Figure 3.2: PIC18F4550 with harvard architecture.....	14
Figure 3.3: Internal architecture of the microcontroller.....	15
Figure 3.4: PIC18F4550 Data Memory Map.....	16
Figure 3.5: Program memory map.....	17
Figure 3.6: Table read operation.....	20
Figure 3.7: Table write operation.....	21
Figure 4.1: A picture of the PICkit™2 Starter Kit.....	39
Figure 4.2: A schematic showing the block diagram of a regulated power supply	41
Figure 4.3: Smoothed varying direct current.....	42
Figure 4.4: Schematic diagram of an LCD.....	45
Figure 4.5: GSM architecture.....	48
Figure 4.6: Internal structure of $G_{1/2}$ flow sensor.....	51
Figure 4.7: A solenoid valve.....	51
Figure 5.1: Block diagram of the proposed intelligent water meter system.....	53
Figure 5.2: Pin configuration of the $G_{1/2}$ flow sensor.....	54
Figure 5.3: Flow sensor interface to PIC18F455.....	55
Figure 5.4: Liquid Crystal Display to PIC 18F4550 microcontroller interface.....	57

Figure 5.5: GSM interface to PIC18F4550 microcontroller.....	58
Figure 5.6: Interface of Solenoid valve to the PIC18F4550.....	59
Figure 5.7: Flowchart for the system software design.....	61
Figure 5.8: A screenshot of Mikroc software illustrating creation of new project.....	62
Figure 5.9: A Screenshot illustrating selection of project name and project folder.....	63
Figure 5.10: A Screenshot illustrating the process of addition or removal of files.....	63
Figure 5.11: A Mikroc screen shot illustrating library selection.....	64
Figure 5.12: A screenshot showing successful creation of a new project.....	64
Figure 6.1: A picture of the proposed designed system.....	66
Figure 6.2(a): Pulses at lower frequency.....	67
Figure 6.2(b): Pulses higher frequency.....	67
Figure 6.3: A picture of the valve used to control flow rate.....	68
Figure 6.4: A graph of Frequency against Flow rate.....	69
Figure 6.5: A picture of a cell phone showing SMS received.....	71
Figure 6.6: Regulated power supply.....	72

LIST OF TABLES

Table 3.1: Registers associated with program memory.....	19
Table 3.2: Ports of PIC 18F4550.....	21
Table 3.3: Timer0 Control Register (T0CON).....	27
Table 3.4: Timer1 Control Register (T1CON).....	28
Table 3.5: Timer2 Control Register (T2CON).....	30
Table 3.6: Timer3 Control Register (T3CON).....	31
Table 3.7: Transmit status and control register.....	36
Table 3.8: Receive status and control register.....	36
Table 3.9: Baud rate control register.....	36
Table 4.1: The PIC kit™ 2 programmer pin functions.....	40
Table 5.1: Liquid Crystal Display pins configuration.....	56
Table 6.1: Time(s) taken to fill 1 L at different frequencies.....	68
Table 6.2: Flow rate at different frequencies.....	69
Table 6.3: LCD and GSM readings.....	70
Table 6.4: Table showing the response of the control valve to SMS.....	71
Table 6.5: Output Voltage against loading of power supply	72
Table 6.6: Measured volume of the proposed design versus the actual volume.....	73
Table 6.7: Mechanical meters versus the designed water meter system.....	74

ABBREVIATIONS AND ACRONYMS

ADC	Analogue to Digital Converter
CPU	Central Processing Unit
DAC	Digital to Analogue Converter
EEPROM	Electrically Erasable Read Only Memory
EUSART	Enhanced Universal Synchronous Asynchronous Receiver Transmitter
ETSI	European Telecommunications Standards Institute
FOSC	Frequency of the Oscillator
GPRS	General Packet Radio Service
GPR	General Purpose Register
GSM	Global System for Mobile Communications
IDE	Integrated Development Environment
I/O	Input/output
LCD	Liquid Crystal Display
MCU	Microcontroller Unit
MMS	Multimedia Messaging Service
MSB	Most Significant Bit

Ofwat	Office of water services (England and Wales)
PIC	Peripheral Interface Controller
PLL	Phase Locked Loop
RAM	Random Access Memory
ROM	Read Only Memory
SFR	Special Function Register
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter
WAP	Wireless Application Protocol
WDT	Watchdog Timer
WDTEN	Watchdog Timer Enable

ABSTRACT

In the water distribution industry, meter reading is an important activity as source of revenue to the water utility company and at the same time it is meant to ensure accurate recording of consumption. Manual water meter reading is still being done in many countries in the world where employees of water companies take the readings from residential and business premises. Water bills are then prepared based on this data. This process is inefficient and costly in terms of man-hours, especially with the increase in residential houses in most of urban areas. The employees of water companies also face some difficulties in accessing some premises especially in the slum areas and hence the meter readings in such areas are estimated. The process of manual meter reading is both cumbersome and inaccurate. Different researchers have used different technologies in order to automate meter reading but very little has been done using the GSM technology. In addition to this, the earlier designs were expensive. In this research a prototype of an automatic water meter reading system has been designed, fabricated and tested. It consists of $G_{1/2}$ flow sensor, PIC18F4550 microcontroller and its interfacing board, SIM 900 GSM module, solenoid control valve. $G_{1/2}$ flow sensor has been used for water flow detection, the solenoid control valve has been used to connect or disconnect the water supply and the GSM module has been used for sending short messages on water volume consumption while the entire system is controlled by PIC18F4550 microcontroller. In this design as water flows through flow sensor, the rotor blades rotate generating pulses. The pulses are counted and converted to a volumetric flow using the PIC18F550 microcontroller. The volume of water consumption is displayed on an LCD. The water meter reading is sent automatically to the Water Company as a short message for billing purposes after a specified period. The designed system has been found to be about 98% accurate and is cost effective. It has the potential to automate water meter reading and hence increase efficiency.

CHAPTER ONE

INTRODUCTION

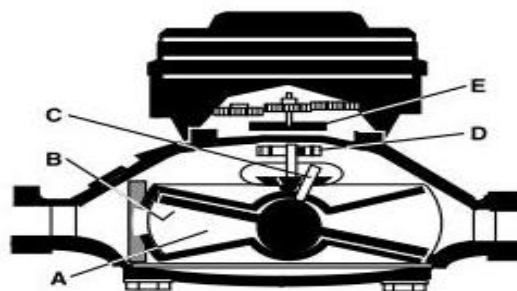
1.1 Background to the study

In many parts of the world analogue water meters have been installed by water companies to measure the consumer's water consumption. Water meters are read on a monthly basis by authorized water meter reader hired by the water companies. The consumer's bill is computed based on the approved water rates and according to the amount of water consumed. If the meter reading is not possible for instance where there is no physical access to the meter, then the bill is prepared based on average consumption. This method is expensive and is also error prone. This traditional way of collecting information has shown noticeable disadvantages to the water companies and to their clients due to discrepancies in the information collected manually.

Water meters generally fall into three categories: simple mechanical water meters, mechanical water meters with an electronic communication device, and fully electronic water meters. Mechanical water meters are used to measure the volume of water used by residential and commercial buildings that are supplied by a public water supply system. In most parts of the world, water meters measure flow in m^3 over a period of time. However, for some considerable time, there has been a constant debate, as to the validity and accuracy characteristics of mechanical meters (Ofwat, 2013). This has led to the search for fully electronic meters. There are two main types of water meters for volumetric measurement; these are the displacement and velocity types.

1.1.1 Displacement water meters

This type of water meter is most often used in residential and small commercial applications. Displacement meters are commonly referred to as positive displacement meters. The two common types are oscillating piston meters and nutating disk meters. Either method relies on the water to physically displace the moving measuring element in direct proportion to the amount of water that passes through the meter. The piston or disk moves a magnet that drives the register. Positive displacement meters are generally very accurate at the low-to-moderate flow rates typical of residential and small commercial users. Because displacement meters require that all water flows through the meter to "push" the measuring element, they are generally not practical in large commercial applications requiring high flow rates or low pressure loss. Positive displacement meters normally have a built-in strainer to protect the measuring element from rocks or other debris that could stop or break the measuring element (Ofwat, 2013). The positive displacement meter is shown in figure 1.1



Liquid flowing through the meter chamber (A) causes a disc (B) to nutate or wobble. This motion, in turn, results in the rotation of a spindle (C) and drive magnet (D). Rotation is transmitted through the wall of the meter to a second magnet (E) which operates the transmitter.

Figure 1.1: Positive displacement meter (Copyright Badger Meter, Inc., Milwaukee, WI)

1.1.2 Velocity water meter

A velocity-type meter measures the velocity of flow through a meter of a known internal capacity. The speed of the flow can then be converted into volume of flow to determine the consumption. The velocity- based meters include jet meters (single-jet and multi-jet), turbine meters and electromagnetic flow meters. Most of these meters have an adjustment vane for calibrating the meter to the required accuracy. Multi-jet meters use multiple ports surrounding an internal chamber to create multiple jets of water against an impeller, whose rotation speed depends on the velocity of water flow. Multi-jets are very accurate at low flow rates.

Turbine meters use the mechanical energy of the fluid to rotate the rotor in the flow stream. The speed of water flow is proportional to the rotor speed. Blade movement is detected magnetically. These meters are less accurate than displacement and jet meters at low flow rates, but the measuring element does not occupy or severely restrict the entire path of flow. The flow direction is generally straight through the meter, allowing for higher flow rates and less pressure loss than displacement-type meters. They are the meter of choice for large commercial users, fire protection and as master meters for the water distribution system. They are accurate in normal working conditions but are greatly affected by the flow profile and fluid conditions (Mulley, 2004).

Electromagnetic flow meter is a velocity-type water meter, except that they use electromagnetic properties to determine the water flow velocity, rather than the mechanical means used by jet and turbine meters. Electromagnetic flow meters apply Faraday's law of induction. They need alternating current or direct current electricity from a power line or battery to generate the magnetic field. Since these flow meters have no mechanical measuring element, they normally have the advantage of being

able to measure flow in both directions, and use electronic circuit for measuring and quantifying the flow. Magnetic flow meters can also be useful for measuring raw water and waste-water, since there is no mechanical measuring element to get clogged or damaged by debris flowing through the meter. Since stray electrical energy flowing through the flow tube can cause inaccurate readings, most of these meters are installed with either grounding rings or grounding electrodes to divert stray electricity away from the electrodes used to measure the flow inside the flow tube (Mulley, 2004). A velocity type meter is shown in figure 1.2.

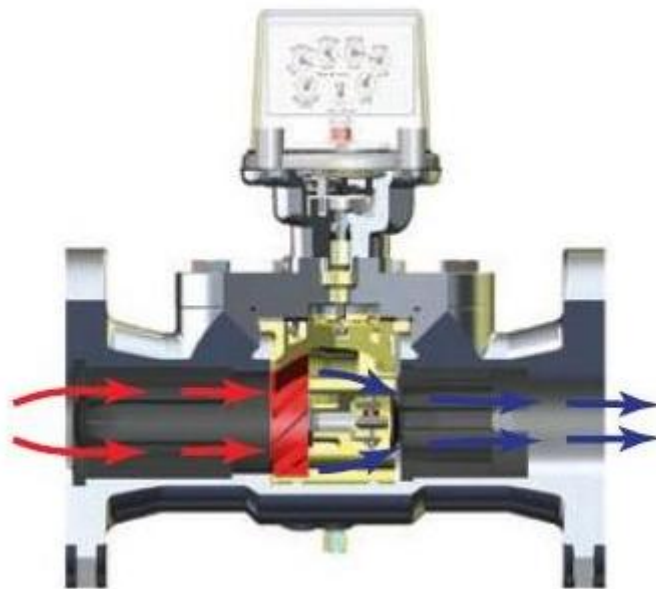


Figure 1.2: Velocity type meter (Copyright IMAC Systems, Inc., Tully town, PA)

1.2 Meter Reading

In water meters the cumulative record of the number of units consumed are stored in a register. The meters can be manufactured to read cubic meters. There are several ways in which meter reading can be accomplished.

a) Direct reading

This is where an individual directly reads the meter register.

b) Remote reading

This is where reading is done through an electronic signal, usually with a wire to a separate station where the individual could either direct read or touch read.

c) Touch read or plug-in reader

This is where a handheld device is used. The device picks the electronic signal from plugging-in or touching the remote station and translates it into the number of cubic meters used. The remote station can even be on top of the meter pit lid. The readings are stored on the hand-held device until the meter reader gets back to the office to be downloaded to a computer.

d) Automatic meter reading

This is a newer development which enables the meter reader to drive down the street and obtain the readings from the meters usually through radio transmission.

1.3 Statement of the research problem

In many parts of the world the reading of water meters is a manual task. Such a process is monotonous, error prone and expensive in manpower utilization. In addition, some consumers may not be physically accessible or live in a hostile social environment that is difficult to pay regular visits hence making it difficult to take water meter readings. The available water meter systems in such countries are not provided with the means of automating meter reading. With the growth of mobile communication technology and the availability of low cost sensors and microcontrollers, the automation of water meter reading can be envisaged.

1.4 Objectives

1.4.1 General objective

The general objective of this research work is to design a cost effective system that will automate data collection of water consumption to allow utility companies to provide efficient services to consumers.

1.4.2 Specific objectives

The specific objectives of this research work are;

- i. To design a sensor system to detect presence of water in a pipe and the flow rate.
- ii. To develop and implement a C language-based program for flow rate and volumetric flow determination using a PIC18F4550 microcontroller and display the same on the LCD.
- iii. To develop and implement a C language-based program to enhance communication between the PICF4550 microcontroller and GSM module, sending and receiving of short messages through the GSM Network.
- iv. To develop and implement a C language-based program to enhance communication amongst the PICF4550 microcontroller, solenoid control valve and GSM module, opening and closing the valve when a command is issued.

1.5 Rationale for the research

Currently, manual meter reading is still widely used in water meter system, and has been noted to be inefficient and costly in terms of man-hours. With the growth in population, the built environment can no longer be served by the traditional manual meter-reading method. The proposed low cost GSM based water meter system can provide a low cost solution by improving the efficiency water consumption billing.

Microcontrollers can be used and are designed for embedded applications, have peripherals integrated on the same chip, can interface directly to sensors. A microcontroller can employ sensors to detect flow rate and display the same on an LCD. It can serve as an efficient water billing system if interfaced to a GSM Modem that sends data to the utility office.

1.6 Scope of the study

This research concentrated on developing an automatic water meter reading system since manual water meter reading is still being widely used in many parts of the world. The system is GSM based. The billing process has been excluded but the display of the volume of water consumed in a given duration is included.

1.7 Thesis Outline

In this section an outline of the thesis is given.

Chapter 1 introduces the thesis, gives the objectives, rationale and the scope of the study. In chapter 2 a brief highlight on some water meter system designs that have been developed in the effort of automating water meter reading is given.

Chapter 3 gives a detailed discussion of the PIC18F4550 microcontroller. The areas discussed include the microcontroller features, architecture, registers, ports, interrupts, timer modules, oscillators and EUSART.

In chapter 4 an explanation of MikroC Integrated Development (IDE), PICKit2 programmer, Liquid crystal display, GSM module, Control valve and flow measurement is given.

Chapter 5 gives a description of the steps taken in the design of a GSM based water meter system.

The results and discussions are presented in chapter 6.

In chapter 7 a conclusion and recommendations for future work are given.

1.8 Note on publication

The contribution made in the course of the investigations reported in this thesis has resulted in the following publication: Peter Mwangi, Elijah Mwangi and Patrick Karimi, “A Low Cost Water Meter System based on the Global System for Mobile Communications,” *International Journal of Computer Applications*, vol 142, no.12, pp.7-12, May 2016.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter gives a brief highlight on some of the water meter system designs that have been developed in the effort of automating water meter reading.

2.2 Automatic meter reading

Automatic Meter Reading System (AMR) is the remote collection of consumption data from customers' utility like energy meters, water meters among others using radio frequency, telephony, power-line or satellite communications technologies and process the data to generate the bill (Moghavvemi and Wong, 2005). With automatic meter reading researchers have developed pulse or encoder registers to produce electronic output for radio transmitters, reading storage devices, and data logging devices. Pulse meters send a digital or analogue electronic pulse to a recording device. The encoder registers have an electronic means permitting an external device to interrogate the register to obtain the stored reading.

2.3 Related works

Nelson (2014) designed a non-contact arrow pointer sensor design to reduce power consumption of low-cost and high accuracy hybrid water meter. This system involved embedding an electrical circuit into the body of a conventional mechanical water meter, to be compatible with the currently used arrow meter. An electrical circuit was mounted on Printed Circuit Board (PCB) and only one low-cost PIC16F877A was used in the meter body for reading the arrow scale. At a given frequency through the arrow sensor to a detection circuit, a signal generator (SG) outputs a square

waveform. The detected result is then binary encoded and transmitted to a server. The data is transmitted through GSM network.

Yogendra and Tadwalkar (2014) implemented a GSM based water meter as a step towards automation in billing system. A paddle wheel flow sensor JT121 was used to measure the water flow accurately with the help of rotating paddles. The system was able to send water usage within fraction of seconds in the form of text message by using existing GSM network. It gave better results than any other metering systems such as mechanical, ultrasonic, electro-magnetic systems. By using this system, water consumption can be observed in real time. Such a system can also incorporate the control of water consumption.

Ria *et al.* (2013) designed and developed an automatic flow meter for irrigation systems which supplies only the required amount of water to crops. This saves water as well as energy. They employed a $G_{1/2}$ Hall Effect water flow sensor as a sensing unit with a turbine rotor inside it whose speed of rotation changes with the different rate of flow of water. The Hall Effect sensor gives an output of the corresponding pulse train for frequency input to the microcontroller. The whole system comprised of an AT89S52 microcontroller, $G_{1/2}$ Hall Effect water flow sensor, relay, optocoupler, a water pump, 5V supply, LCD, keypad and some passive components.

Rainer *et al.* (2013) developed remote automatic water meter reading system using global system for mobile communications. The study aimed at developing an automatic water meter reading system that would enhance the analogue water meter data collection by utilizing wireless technology using GSM module system for data transmission to increase the range of data transmission. It consisted of the hardware

and the software design. The hardware involved the analogue meter, the GSM module, the differential pressure sensor, and a desktop or a laptop. While the software was developed in order to receive and to store data transmitted from the GSM module into the database and output graphically. A T-test was used to conduct and to verify if the readings from the design were comparable to the actual volume measured from the analogue meter.

Santhosh and Roy (2012) proposed an intelligent flow measurement technique using an ultrasonic flow meter with an optimized neural network. The objective of their work was to extend the linearity range of measurement to 100% of the input range, to make the measurement system adaptive to variations in pipe diameter, liquid density, and liquid temperature, and to achieve the above two objectives by an optimal Artificial Neural Network.

Han and Myaing (2011) developed microcontroller-based water flow control system. In this system, the sensing unit consisted of a photo interrupter and slotted disk that produced pulse train for frequency input of the microcontroller. This signal is converted into flow rate by software program in PIC. This flow rate is compared to the set point value. The microcontroller can control the water valve by using a DC motor to vary the water flow rate based on this comparison.

Lee *et al.* (2008) developed a wireless digital water meter with low power consumption for automatic meter reading in which they used magnetic hole sensors to determine the amount of water consumption. They used ZigBee wireless protocol to transfer amount of water consumption to the gateway.

2.4 Proposed investigation

In the reported works, little or no attention has been paid to aspect of using GSM network to update the utility company on the consumption of water automatically after certain duration of time. The proposed system seeks to pay more attention on use of GSM network to update the water utility company on water consumption. The GSM network has been used to send short messages on volume of water consumed after a specified period of time.

CHAPTER THREE

THE PIC18F4550 MICROCONTROLLER

3.1 Introduction

The chapter gives a detailed discussion on the PIC18F4550 microcontroller. The areas discussed include the microcontroller features, architecture, registers, ports, interrupts, timer modules, oscillators and EUSART.

3.2 The PIC18F4550 Microcontroller chip

The PIC18F4550 is one of the advanced 8-bit microcontrollers developed by Microchip Technology Inc. It belongs to PIC18F family of microcontrollers and has the following features.

- i. It has five I/O ports of different sizes.
- ii. It has a facility for generating the clock internally by operating in internal oscillator mode and therefore there is no need of an external crystal.
- iii. It is a Nano watt technology, which can decrease the power consumption during the operation.
- iv. It works with the operating frequency of up to 48 MHz.
- v. It has a program memory addressing up to 2MB and data memory addressing 4kB.
- vi. It has a fast 10-bit analogue to digital converter.
- vii. It has four general timers and a watchdog timer.

The PIC18F4550 microcontroller has 40 pins. Its pin configuration is shown in figure 3.1.

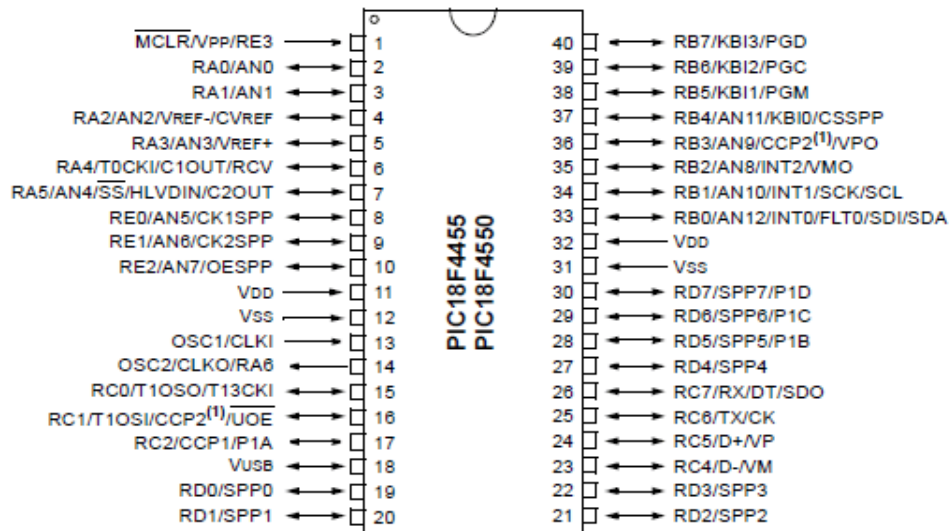


Figure 3.1: PIC18F4550 pin diagram configuration (www.microchip.com).

3.2.1 PIC18F4550 Architecture

The PIC18F4550 microcontroller has a Harvard architecture which include; Microprocessor unit, Program memory for instructions, Data memory for data, I/O ports and support devices like timers. The advantage of this architecture is that instructions and data can be fetched simultaneously since they use two separate buses. The PIC18F4550 Harvard architecture is illustrated in figure 3.2.

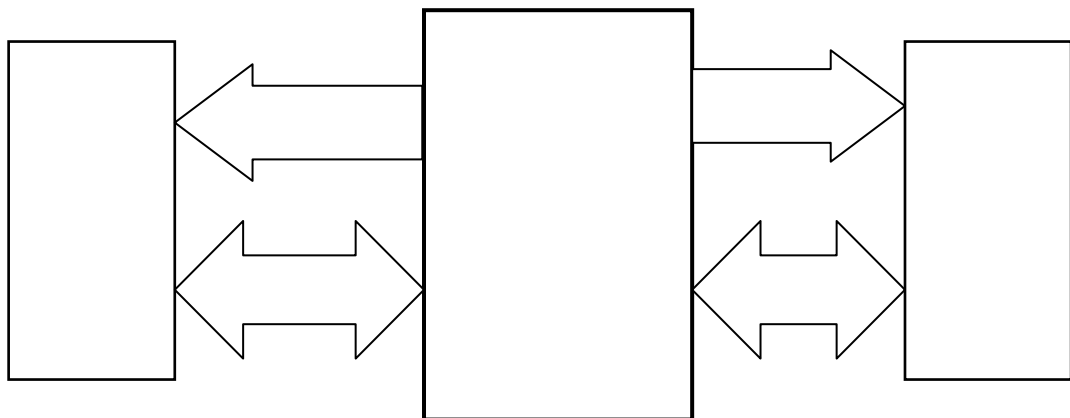


Figure 3.2: PIC18F4550 Harvard architecture.

The internal architecture of PIC18F4550 microcontroller is illustrated in figure 3.3 while a detailed architecture of the microcontroller is given in appendix A.

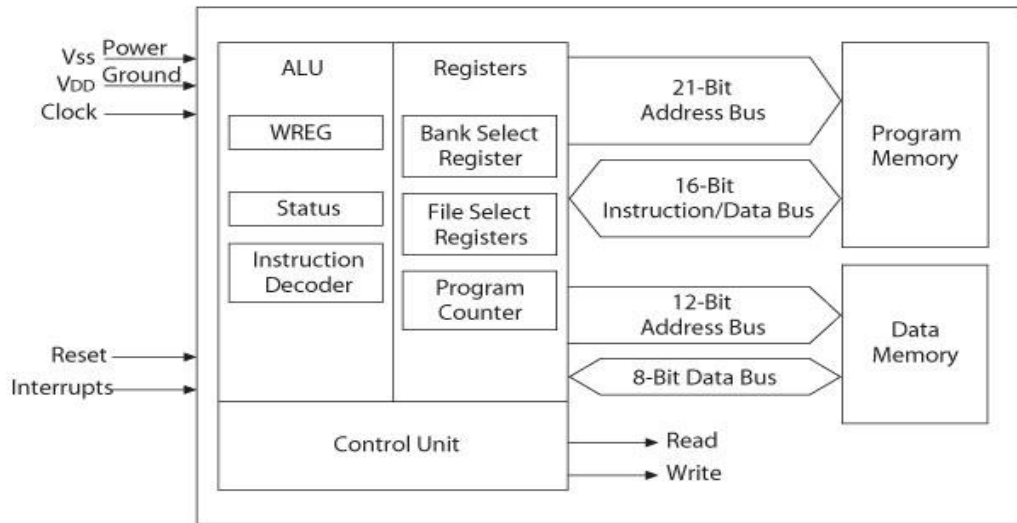


Figure 3.3: Internal architecture of the microcontroller (www.microchip.com)

3.2.2 Data Memory

This memory is used for transitory data when the programme is being executed. It can either be a SRAM or an EEPROM or both. An EEPROM is non-volatile whereas a SRAM is volatile. The registers in the data memory have 12-bit address each which allow up to 4096 bytes of data memory. The data memory is divided into General purpose Registers (GPR) and Special Function Registers (SFR).

The GPRs are used for data storage and scratchpad operations in the user's application while SFRs are used for control and status of the controller and peripheral functions. The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by Direct, Indirect or Indexed addressing modes (Trivedi, 2013).

The bank selection is achieved by using bank select register (BSR). The PIC18F4550 microcontroller implements access bank to ensure SFRs and GPRs registers are retrieved in a single cycle. The access bank is a 256-byte memory space that provides fast access to SFRs and the lower portion of GPR Bank 0 without using the bank select register. Figure 3.4 illustrates the data memory map.

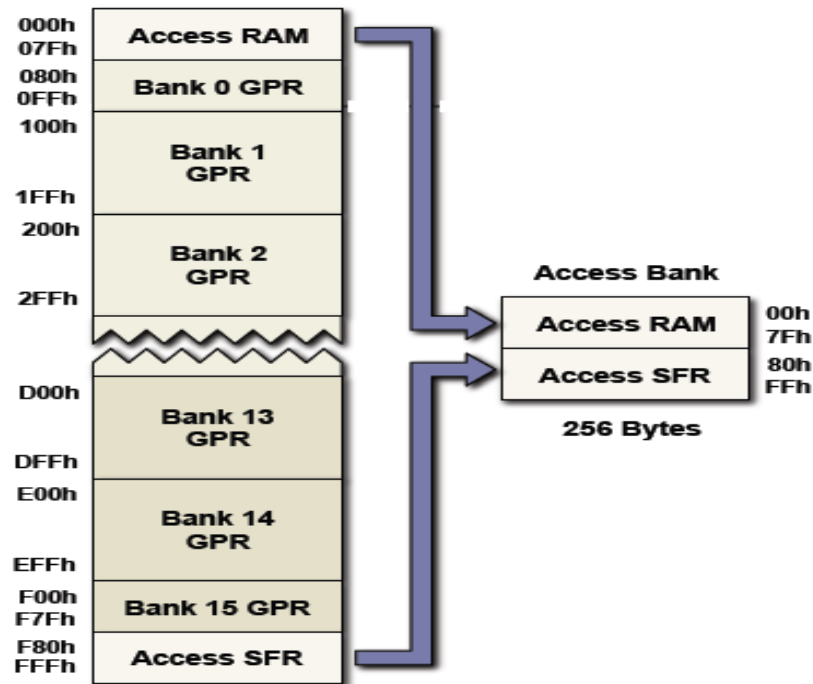


Figure 3.4: PIC 18F4550 Data Memory Map (www.microchip.com)

The memory map is divided into 16 banks each of 256B. The two access banks termed as Access RAM and Access SFR are partitioned from bank 0 and bank 15. The Access RAM is at the top of the memory map with address 000h to 07Fh. The Access SFR is at the bottom of the memory map with address F80h to FFFh.

3.2.3 Programme Memory

The PIC18F4550 microcontroller has 32kB of flash memory and can store up to 16384 single word instructions. The memory is readable, writable and erasable during normal operation. A read from the program memory is executed on one byte at a time. A write to program memory is executed on blocks of 32B at a time. The programme

memory is erased in blocks of 64 bytes at a time. Instruction fetching stops during programme memory write or erase until the operation is complete. The programme memory write or erase is terminated by an internal programming timer. A value written to program memory does not need to be a valid instruction. Figure 3.5 shows the program memory map

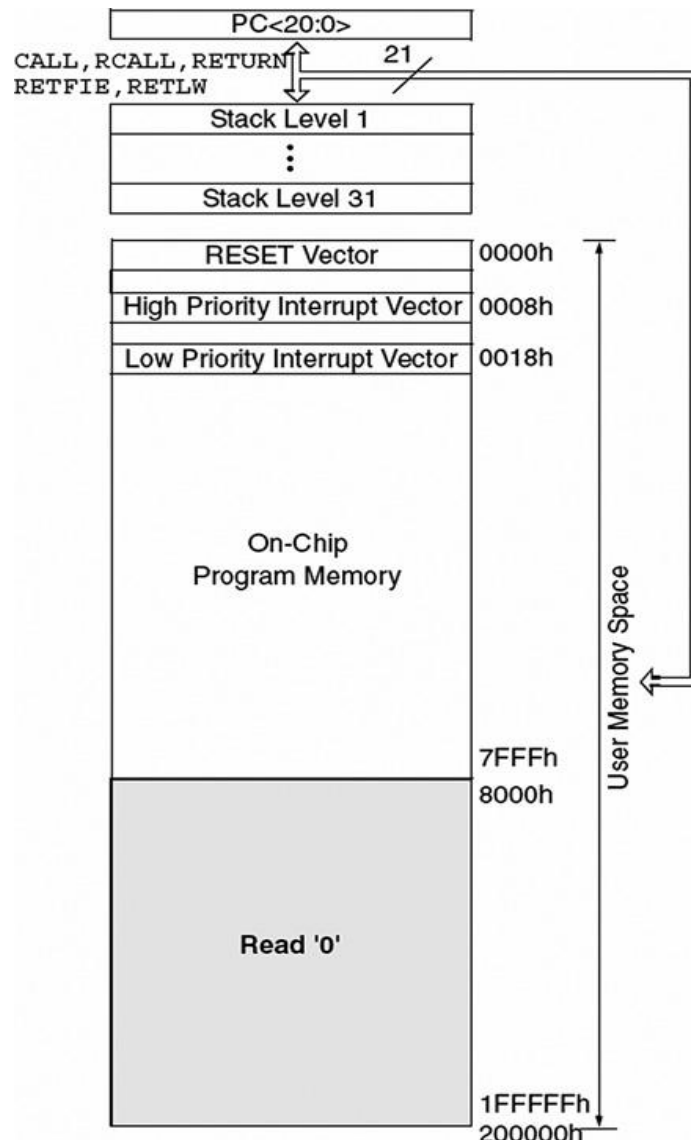


Figure 3.5: Program memory map (www.microchip.com)

The programme memory map comprises of programme counter, stack levels from 1 to 31, reset vector with address 0000h, interrupt vectors with addresses 0008h and 0018h, on-chip program memory which goes up to address 7FFFh and an unused

portion from address 8000h to 1FFFFFFh. In order to read from or write into programme memory, table read (TBLRD) and table Write (TBLWT) operations allow the processor to move bytes between the program memory space and the data RAM.

The programme memory space is 16 bits wide, while the data RAM space is 8 bits wide. The tables read and write operations move data between these two memory spaces through an 8-bit register (TABLAT). The tables read operations retrieve data from programme memory and place it into the data RAM space while table write operations store data from the data memory space into holding registers in programme memory. The table operations work with byte entities. A table block containing data, rather than programme instructions, is not required to be word-aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into programme memory, programme instructions will need to be word-aligned. The programme memory can therefore access at most 2MB instructions. Almost all instructions of PIC18 are 16-bit and therefore the instruction bus is 16-bit long. The program memory addresses are 21-bit address starting at location 000000h.

3.2.4 Programme stack memory

The PIC18F4550 microcontroller contains a programme stack that stores up to 31 return addresses from functions. It is used to hold the return addresses for a subroutine call and interrupt processing. During a subroutine call the stack pointer is first incremented, the memory location it points to is written with the contents of the program counter. During the return from the subroutine call, the memory location the stack pointer has pointed is decremented. When the return occurs within the function, the return address is retrieved from the stack and placed into the program counter.

3.3 Registers

There are a number of control registers that are used in conjunction with the TBLRD and TBLWT instructions. They include:

- a) EECON1 register
- b) EECON2 register
- c) TABLAT register
- d) TBLPTR registers

The registers are illustrated in table 3.1.

Table 3.1: Registers associated with program memory (www.microchip.com)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TBLPTRU	—	—	bit 21 ⁽¹⁾	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					53
TBLPTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								53
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								53
TABLAT	Program Memory Table Latch								53
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	53
EECON2	EEPROM Control Register 2 (not a physical register)								55
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	55
IPR2	OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP	56
PIR2	OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF	56
PIE2	OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE	56

3.3.1 EECON1 AND EECON2 Registers

The EECON1 register is the control register for memory accesses. The EECON2 register is not a physical register although it is used exclusively in the memory write and memory erases sequences. The MSB in the EECON1 register is term EEPGD and is a control bit that determines if the access will be a program or data EEPROM memory access. When clear, any subsequent operations will operate on the data EEPROM memory. When set, any subsequent operation will operate on the program memory. The CFGS control bit of EECON1 register determines if the access will be

to the configuration/calibration registers or to program memory/data EEPROM memory.

3.3.2 Table Latch (TABLAT) Register

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between the programme memory and the data RAM.

3.3.3 Table Pointer (TBLPTR) Register

The Table Pointer (TBLPTR) register addresses a byte within the programme memory. It is comprised of three SFR registers which are Table Pointer Upper Byte (TBLPTRU) which is 6-bit wide, Table Pointer High Byte (TBLPTRH) which is 8-bit wide and Table Pointer Low Byte (TBLPTRL) which is 8-bit wide. The three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2MB of program memory space. The 22nd bit allows access to the device ID, the user ID and the Configuration bits. The Table Pointer, TBLPTR, is used by the TBLRD and TBLWT instructions. The table read operation with the program memory and the data RAM is illustrated in figure 3.6

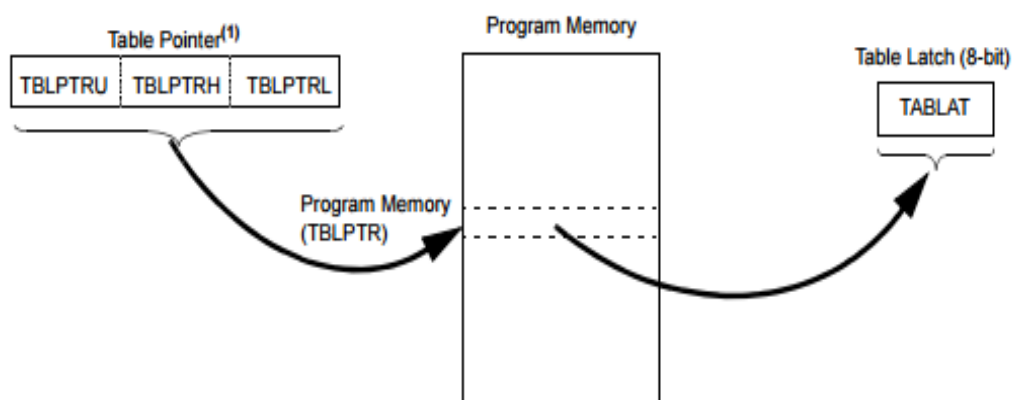


Figure 3.6: Table read operation (www.microchip.com)

The table write operation with the program memory and the data RAM is illustrated in figure 3.7

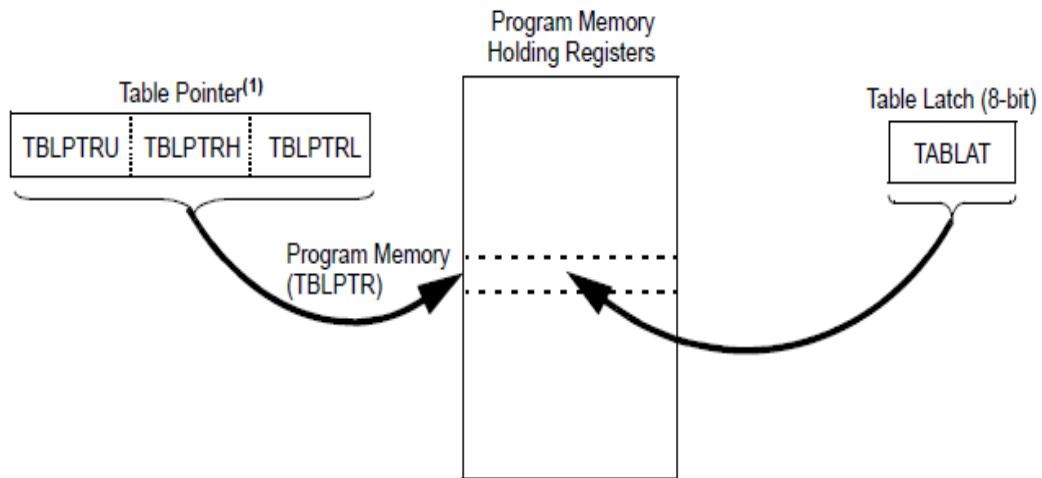


Figure 3.7: Table write operation (www.microchip.com)

3.4 I/O PORTS

The PIC18F4550 microcontroller has five ports and each has three special function registers for its operation. Table 3.2 shows the PORT description of the PIC18F4550

Table 3.2: Ports of PIC 18F4550

PORTS	BUS SIZE	BUS NAME
A	7	RA ₀ -RA ₆
B	8	RB ₀ -RB ₇
C	7	RC ₀ -RC ₂ , RC ₄ -RC ₇
D	8	RD ₀ -RD ₇
E	4	RE ₀ -RE ₃

The ports have the following registers:

- i) The TRIS register which provides direction functionality.
- ii) The PORT register which reads logic levels on the input lines.
- iii) The LAT register which is an output latch.

3.4.1 PORTA

PORTA is 7-bit wide and is a bidirectional port. Its corresponding data direction register is TRISA. When TRISA bit is set, it makes the corresponding PORTA pin an input. When TRISA bit is cleared, it makes the corresponding PORTA pin an output. The status of the pins is known by reading the PORTA while writing to it will write to the port latch. The Data Latch register (LATA) is memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA. The RA4 and RA6 pins are multiplexed with Timer 0 and the main oscillator respectively. All other PORTA pins have TTL input levels and full CMOS output drivers. The TRISA register controls the direction of the RA pins, even when they are being used as analogue inputs. The bits in the TRISA register must be maintained set when using them as analogue inputs.

3.4.2 PORTB

This is an 8-bit wide port and is bidirectional. Its corresponding data direction register is TRISB. When TRISB bit is set, the corresponding PORTB pin will be an input. Clearing a TRISB bit will make the corresponding PORTB pin an output. The data latch register (LATB) is also memory mapped. Read-modify-write operation on the LATB register will read and write the latched output value for PORTB. Each of the PORTB pins has a weak internal pull-up. The RB7, RB6, RB5 and RB4 pins of PORTB have interrupt- on-change feature. Any of the mentioned pin configured as an output is excluded from the interrupt- on-change comparison. Only pins configured as inputs can cause this interrupt to occur.

3.4.3 PORTC

PORTC is a 7-bit wide, bidirectional port. Its corresponding data direction register is TRISC. When a TRISC bit is set, the corresponding PORTC pin will be an input. When a TRISC bit is cleared, the corresponding PORTC pin will be an output. The RC3 pin is not implemented in the PIC18F4550. The Data Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register will read and write the latched output value for PORTC. In addition the port is primarily multiplexed with serial communication modules, including the EUSART, MSSP module and the USB module. Pins RC4 and RC5 are multiplexed with the USB module. Both RC4 and RC5 have TTL input buffers instead of the Schmitt trigger buffers on the other pins. As digital ports, they can only function as digital inputs.

3.4.4 PORTD

PORTD is an 8-bit wide, bidirectional port. Its corresponding data direction register is TRISD. Setting a TRISD bit makes the corresponding PORTD pin an input while clearing a TRISD bit makes the corresponding PORTD pin an output. The data latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register will read and write the latched output value for PORTD. All pins on PORTD are implemented with Schmitt trigger input buffers. Each pin is individually configurable as an input or output. Each of the PORTD pins has a weak internal pull-up. The weak pull-up is automatically turned off when the port pin is configured as a digital output or as one of the other multiplexed peripherals. The pull-ups are disabled on a Power-on Reset. Three of the PORTD pins are multiplexed with outputs, P1B, P1C and P1D, of the Enhanced CCP module.

3.4.5 PORTE

PORTE is a 4-bit wide port. Its corresponding data direction register is TRISE. Three pins RE0, RE1 and RE2 can be individually configured as inputs or outputs. These pins have Schmitt trigger input buffers. Setting a TRISE bit will make the corresponding PORTE pin an input while clearing a TRISE bit will make the corresponding PORTE pin an output. When selected as an analogue input, these pins will read as '0's. In addition to port data, the PORTE register, TRISE controls the direction of the RE pins, even when they are being used as analogue inputs. The pins must be maintained as inputs when using them as analogue inputs. The data latch register (LATE) is also memory mapped. Read-modify-write operations on the LATE register will read and write the latched output value for PORTE.

The RE3 pin can only be configured as an input. Its operation is controlled by the MCLRE Configuration bit. When selected as a port pin, it functions as a digital input only pin; as such, it does not have TRIS or LAT bits associated with its operation. Otherwise, it functions as the device's Master Clear input. In either configuration, RE3 also functions as the programming voltage input during programming.

3.5 Interrupts

An interrupt is an event that requires the processor to stop normal operation after which a program code related to the event causing the interrupt is executed. They can be internally or externally generated. The PIC18F4550 has multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will

interrupt any low-priority interrupts that may be in progress. The high priority interrupt and low priority interrupts are illustrated in figure 3.5.

There are ten registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1 and PIR2
- PIE1 and PIE2
- IPR1 and IPR2

Each interrupt source has three bits to control its operation. The functions of these bits are:

- (i). Flag bit to indicate that an interrupt event occurred.
- (ii). Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set.
- (iii). Priority bit to select high priority or low priority.

The interrupt priority feature is enabled by setting the IPEN bit of the RCON register. When interrupt priority is enabled, there are two bits which enable interrupts globally. When the interrupt flag enable bit and appropriate global interrupt enable bit are set, the address 0008h or 0018h will be selected, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

For an interrupt to be accepted by the CPU the following conditions must be satisfied:

- i) The global interrupt enable/disable bit GIE of INTCON must be set to 1.
- ii) The interrupt enable bit of the interrupt source must be enabled. For instance, if the interrupt source is external interrupt pin INT0, then the bit INTOIE of register INTCON must be set to 1.
- iii) The interrupt flag of the interrupt source must be cleared. For instance, if the interrupt source is external interrupt pin INT0, then the bit INTOIF of register INTCON must be cleared to 0.
- iv) The peripheral interrupt enable/disable bit PEIE of INTCON must be set to 1 if the interrupt source is a peripheral.

3.6 Timer module

The PIC18F4550 has four general timers and a watchdog timer. The timers can operate as counters and also in various modes.

3.6.1 Timer0 module

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit and 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt on overflow

The T0CON register controls all aspects of the module's operation, including the prescale selection. The functions are illustrated in table 3.3.

Table 3.3: Timer0 Control Register (T0CON)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
Bit7						Bit 0	

Bit 7 **TMR0ON**: Timer0 On/Off Control bit; 1 = Enables Timer0, 0 = Stops Timer0

Bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit; 1 = Timer0 is configured as an 8-bit timer/counter, 0 = Timer0 is configured as a 16-bit timer/counter

Bit 5 **T0CS**: Timer0 Clock Source Select bit; 1 = Transition on T0CKI pin, 0 = Internal instruction cycle clock (CLKO).

Bit 4 **T0SE**: Timer0 Source Edge Select bit; 1 = Increment on high-to-low transition on T0CKI pin, 0 = Increment on low-to-high transition on T0CKI pin

Bit 3 **PSA**: Timer0 Prescaler Assignment bit; 1 = Timer0 prescaler is NOT assigned, Timer0 clock input bypasses prescaler, 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

Bit 2-0 **T0PS2:T0PS0**: Timer0 Prescaler Select bits

111 = 1:256 Prescale value 110 = 1:128 Prescale value 101 = 1:64 Prescale value

100 = 1:32 Prescale value 011 = 1:16 Prescale value, 010 = 1:8 Prescale value

001 = 1:4 Prescale value 000 = 1:2 Prescale value

Timer0 operates either as a timer or a counter. The operation mode is selected by clearing the T0CS bit. In timer mode, the module increments on every clock by default unless a different prescaler value is selected. If the TMR0 register is written into, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register. The Counter mode is selected by setting the T0CS bit (= 1). In this mode Timer0 increments either on every rising or falling edge of pin RA4/T0CKI/C1OUT/RCV. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE.

3.6.2 Timer1 module

The Timer1 timer/counter module incorporates the following features:

- Software selectable operation as a 16-bit timer or counter.
- Readable and writable 8-bit registers (TMR1H and TMR1L).
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options.
- Interrupt on overflow.
- Module Reset on CCP special event trigger.
- Device clock status flag (T1RUN).

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register as shown in Table 3.4. It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON.

Table 3.4: Timer1 Control Register (T1CON)

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
Bit 7							Bit 0

Bit 7 **RD16**: 16-Bit Read/Write Mode Enable bit

1 = Enables register read/write of Timer1 in one 16-bit operation

0 = Enables register read/write of Timer1 in two 8-bit operations

Bit 6 **T1RUN**: Timer1 System Clock Status bit

1 = Device clock is derived from Timer1 oscillator

0 = Device clock is derived from another source

Bit 5-4 **T1CKPS1:T1CKPS0**: Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale value; 10 = 1:4 Prescale value; 01 = 1:2 Prescale value and 00 = 1:1 Prescale value

Bit 3 **T1OSCEN**: Timer1 Oscillator Enable bit

1 = Timer1 oscillator is enabled

0 = Timer1 oscillator is shut off

The oscillator inverter and feedback resistor are turned off to eliminate power drain.

Bit 2 **T1SYNC**: Timer1 External Clock Input Synchronization Select bit

When TMR1CS = 1:

1 = Do not synchronize external clock input

0 = Synchronize external clock input

When TMR1CS = 0:

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

Bit 1 **TMR1CS**: Timer1 Clock Source Select bit

1 = External clock from RC0/T1OSO/T13CKI pin (on the rising edge)

0 = Internal clock (FOSC/4)

Bit 0 **TMR1ON**: Timer1 On bit

1 = Enables Timer1

0 = Stops Timer1

The operating mode is determined by the clock select bit, TMR1CS. When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction cycle (FOSC/4).

When the bit is set (= 1), Timer1 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled. When Timer1 is enabled, the RC1/T1OSI/UOE and RC0/T1OSO/T13CKI pins become inputs.

3.6.3 Timer2 module

The Timer2 module timer incorporates the following features:

- An 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP module

The module is controlled through the T2CON register as illustrated in table 3.5. The register enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON, to minimize power consumption.

Table 3.5: Timer2 Control Register (T2CON)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
__	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
Bit 7							Bit 0

Bit 7 __ It is unimplemented Read as '0'

Bit 6-3 T2OUTPS3 to T2OUTPS0: Timer2 Output Post scale Select bits

From 0000 to 1111 the postscale will be from 1:1 to 1:16 in the increasing order.

Bit 2 TMR2ON: Timer2 on bit

1 = Timer2 is on 0 = Timer2 is off

Bit 1-0 T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits

00 = Prescaler is 1 01 = Prescaler is 4 1x = Prescaler is 16

In the normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A 2-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by- 16 prescale options. These are selected by the prescaler control bits,

T2CKPS1:T2CKPS0. The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler.

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (Power-on Reset, MCLR Reset, Watchdog Timer Reset or Brown-out Reset) TMR2 is not cleared when T2CON is written.

3.6.4 Timer3 Module

The Timer3 module timer/counter incorporates the following features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Module Reset on CCP Special Event Trigger

The Timer3 module is controlled through the T3CON register. It also selects the clock source options for the CCP modules. These functions are illustrated in table 3.6

Table 3.6: Timer3 Control Register (T3CON)

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON
Bit 7							Bit 0

Bit 7 **RD16**: 16-Bit Read/Write Mode Enable bit

1 = Enables register read/write of Timer3 in one 16-bit operation

0 = Enables register read/write of Timer3 in two 8-bit operations

Bit 6, 3 **T3CCP2:T3CCP1**: Timer3 and Timer1 to CCPx Enable bits

1x = Timer3 is the capture/compare clock source for both CCP modules

01 = Timer3 is the capture/compare clock source for CCP2;

Timer1 is the capture/compare clock source for CCP1

00 = Timer1 is the capture/compare clock source for both CCP modules

Bit 5-4 **T3CKPS1:T3CKPS0**: Timer3 Input Clock Prescale Select bits

11 = 1:8 Prescale value 10 = 1:4 Prescale value 01 = 1:2 Prescale value

00 = 1:1 Prescale value

Bit 2 **T3SYNC**: Timer3 External Clock Input Synchronization Control bit

(Not usable if the device clock comes from Timer1/Timer3.)

When TMR3CS = 1:

1 = Do not synchronize external clock input

0 = Synchronize external clock input

When TMR3CS = 0:

This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

Bit 1 **TMR3CS**: Timer3 Clock Source Select bit

1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)

0 = Internal clock (FOSC/4)

Bit 0 **TMR3ON**: Timer3 On bit

1 = Enables Timer3

0 = Stops Timer3

The operating mode is determined by the clock select bit, TMR3CS. When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction cycle. When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled. As with Timer1, the RC1 and RC0 pins become inputs when the Timer1 oscillator is enabled.

3.6.5 Watchdog Timer

In the PIC18F4550 microcontroller the watchdog timer (WDT) is a free running on-chip RC-based oscillator and does not require any external components. When the WDT times out, a device RESET is generated. If the device is in SLEEP mode, the WDT time-out will wake it up and continue with normal operation. The watchdog is enabled or disabled by bit SWDTEN of register WDTCON. Setting SWDTEN = 1 enables the WDT, and clearing this bit turns off the WDT.

3.7 Oscillators

The PIC18F4550 microcontroller incorporates a different oscillator and microcontroller clock system. The microcontroller has a USB module which has a unique requirement for a stable clock source. It makes it necessary to provide a separate clock source that is compliant with both USB low-speed and full-speed specifications. The PIC18F4550 microcontroller device includes a new clock branch to accommodate these requirements, to provide a 48 MHz clock for full-speed USB operation. Since it is driven from the primary clock source, an additional system of prescalers and postscalers has been added to accommodate a wide range of oscillator frequencies.

3.7.1 Oscillator control

The operation of the oscillator in PIC18F4550 devices is controlled through two Configuration registers and two control registers. Configuration registers, CONFIG1L and CONFIG1H, select the oscillator mode and USB prescaler or postscaler options. As Configuration bits, these are set when the device is programmed and left in that configuration until the device is reprogrammed.

The OSCCON register selects the active clock mode and is primarily used in controlling clock switching in power-managed modes.

The OSCTUNE register is used to trim the INTRC frequency source, as well as select the low-frequency clock source that drives several special features.

3.7.2 Oscillator types

The PIC18F4550 microcontroller can be operated in twelve different oscillator modes. The FOSC3, FOSC2, FOSC1 and FOSC0 configuration bits can be programmed to select one of these modes:

- i. XT Crystal/Resonator
- ii. High-Speed Crystal/Resonator
- iii. High-Speed Crystal/Resonator with PLL Enabled
- iv. External Clock with FOSC/4 Output
- v. External Clock with I/O on RA6
- vi. External Clock with PLL Enabled and FOSC/4 Output on RA6
- vii. External Clock with PLL Enabled, I/O on RA6
- viii. Internal Oscillator used as Microcontroller Clock Source, HS Oscillator used as USB Clock Source
- ix. Internal Oscillator used as Microcontroller Clock Source, EC Oscillator used as USB Clock Source, Digital I/O on RA6

- x. Internal Oscillator used as Microcontroller Clock Source, EC Oscillator used as USB Clock Source, FOSC/4 Output on RA6.

3.8 EUSART

A serial communication interface can either be USART or UART. A USART supports both synchronous and asynchronous modes of operation. The PIC18F4550 microcontroller device supports either one or two identical USARTs. The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of the two serial I/O modules. The EUSART can operate in asynchronous mode, synchronous master mode and synchronous slave mode.

It can be configured as full duplex asynchronous system to communicate with peripheral devices like personal computers

It can be configured as half duplex synchronous system to communicate with peripheral devices like analogue digital converters and digital-analogue converters. It also implements additional features like automatic baud rate detection and calibration. The pins of the Enhanced USART in PIC18F4550 microcontroller are multiplexed with PORTC. In order to configure RC6/TX/CK and RC7/RX/DT/SDO as an enhanced USART the following bits must be set; SPEN and TRISC [6 and 7] must be set.

3.8.1 Registers in the EUSART

The operation of the Enhanced USART module is controlled by three registers namely:

- (i). Transmit Status and Control (TXSTA)
- (ii). Receive Status and Control (RCSTA)

(iii). Baud Rate Control (BAUDCON)

Table 3.7 Transmit status and control register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
Bit 7							Bit 0

Bit 7 CSRC: Clock Source Select bit Bit 6 TX9:9 bit transmit enable bit

Bit 5 TXEN: Transmit Enable bit Bit 4 SYNC: EUSART Mode Select bit

Bit 3 SENDB: Send Break Character bit Asynchronous mode

Bit 2 BRGH: High Baud Rate Select bit Bit 1 TRMT: Transmit Shift Register Status bit.

Bit 0 TX9D: 9th bit of Transmit Data.

Table 3.8 Receive status and control register

R/W	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	RX
SPEN	RX9	SREN	CREN	ADDEN	FERR	DERR	RX9D
Bit 7							Bit 0

Bit 7 SPEN: Serial port enable bit Bit 6 RX9 : Bit receive Enable bit

Bit 5 SREN: Serial port enable bit Bit 4 CREN: Continuous receive enable bit

Bit 3 ADDEN: Address detect enable bit Bit 2 FERR: Framing error bit

Bit 1 OERR: Overrun error bit Bit 0 RX9D: 9th bit of receive data.

Table 3.9 Baud rate control register

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	-	WUE	ABDEN
Bit 7							Bit 0

Bit 7 ABDOVF: Auto-Baud Acquisition Rollover Status

Bit 6 RCIDL: Receive Operation Idle Status

Bit 5 RXDTP: Received Data Polarity Select bit Asynchronous mode

Bit 4 TXCKP: Clock and Data Polarity Select bit Asynchronous mode

Bit 3 BRG16: 16-Bit Baud Rate Register

Bit 2 Unimplemented

Bit 1 WUE: Wake-up Enable bit Asynchronous mode

Bit 0 ABDEN: Auto-Baud Detect Enable bit Asynchronous mode

3.8.2 EUSART asynchronous mode

It transmits and receives the least significant bit first. The transmitter and receiver function independently but use the same data format and baud rate.

3.8.3 EUSART synchronous master mode

In this mode data is transmitted in a half-duplex manner. When transmitting data, reception is inhibited and when receiving data, transmission is inhibited.

3.8.4 EUSART Synchronous Slave Mode

To activate this mode, CSRC bit is cleared. The shift clock in this mode is supplied externally at the CK pin instead of being supplied internally in Master mode. This allows the device to transfer or receive data while in any power-managed mode.

CHAPTER FOUR

TECHNICAL CHOICE OF THE SYSTEM SOFTWARE AND HARDWARE

4.1 Introduction

This Chapter gives a detailed discussion of Mikroc Integrated Development Environment (IDE), PICkit2 programmer, Liquid crystal display, GSM module, Control valve and flow measurement.

4.2 Mikroc Integrated Development Environment

The Mikroc PRO for PIC is a powerful development tool for PIC microcontrollers. It has numerous features. It provides the programmer with the easiest possible solution for developing applications for embedded systems. It allows the following applications;

- i) The C source code is written using the built-in Code Editor.
- ii) The code development is speeded up by availability of PIC libraries in Mikroc PRO.
- iii) The program structure, variables, and functions are monitored in the Code Explorer.
- iv) A human-readable assembly language and a standard HEX compatible with all programmers are generated.
- v) The integrated Mikroc In-Circuit Debugger real-time debugging tool can be used to monitor program execution on the hardware level.
- vi) The program flow and debug executable logic with the integrated Software Simulator can be inspected.

vii) Mikroc PRO for PIC provides plenty of examples to expand, develop, and use as building segments in the projects being developed.

viii) The applications can be organized into projects, consisting of a single project file (extension .mcppi).

ix) Multiple projects can be managed using PRO for PIC IDE.

In order to create code that is executable by the target microcontroller, source files need to be put into a project file. The project file created contains information such as project name and optional description, target device, device flags (configuration word), device clock, list of the project source files with paths, header files (*.h), binary files (*.mcl), image files among others.

4.3 PICkit2 programmer

The PICkit2 programmer is a low cost in-circuit programming module that is capable of programming most of Microchip`s PIC microcontrollers. The PICkit2 starter kit includes the low pin count board incorporating the PIC18F4550 MCU and some minimal circuitry as shown on figure 4.1.



Figure 4.1: A picture of the PICkit™2 Starter Kit

The programmer can be connected to the USB port of the host PC running Mikroc, with a six-pin in-line output plugged into a six-pin male connector on the target board.

The board has four light-emitting diodes (LEDs) that are used to output/display the programmed output sequences, a push button connected to !MCLR and a small pot providing an analogue test input. The LPC board can be powered from the USB port via pins 2 and 3. When programming the microcontroller, +12 V is applied to pin 1, but after programming is complete, it reverts to the reset (! MCLR) input function. On detaching from the programmer, the push button can be configured as a reset input or as a digital input. The Pins 4 and 5 are the ICSPDAT/PGD and the ICSPCLK/PGC lines which are used to carry the programme code to be burned in the PIC memory and the system clock respectively. The function of each of the six-pin connections is summarized in table 4.1.

Table 4.1: The PICkit2 programmer pin functions

Pin	Label	Function
1	V _{PP} /!MCLR	Programming voltage or reset input
2	V _{DD} Target	Power supply positive voltage (+5 V)
3	V _{SS} (ground)	Power supply reference voltage (+0 V)
4	ICSPDAT/PGD	Programming code: bidirectional serial signal
5	ICSPCLK/PGC	Programming clock: unidirectional clock signal
6	Auxiliary	Connected to T1G/CLKOUT

4.4 Power supply

A regulated dc power supply is an electrical circuit that is designed to convert the a.c commercial power supply to a suitable low voltage supply for electronic circuits and other devices. A regulated power supply can be broken down into a series of blocks, each of which performs a particular function. A block diagram of a regulated power supply system which converts a 240V AC mains supply into a regulated 5 V DC supply is illustrated in figure 4.2.

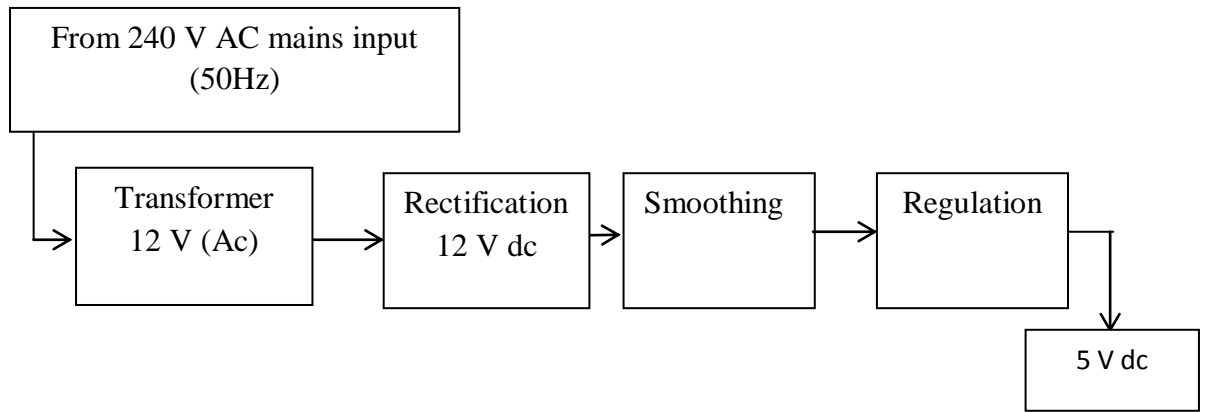


Figure 4.2: A schematic showing the block diagram of a regulated power supply

4.4.1 Transformer

A transformer uses the principle of mutual induction to change values of alternating voltages and currents. According to Faraday's laws of electromagnetic induction, an emf is induced in the secondary which is proportional to the number of turns in the secondary and the flux linking the secondary. The ratio of the number of turns in the secondary and the flux linking the secondary. The ratio of the number of turns in the primary coil to the number of turns in the secondary coil determines the ratio of the voltages in the two coils. A step-down transformer has a large number of turns on its primary coil which is connected to the high voltage mains supply, and a small number of turns on its secondary coil to give a low output voltage.

$$\text{Turns ratio} = \frac{V_p}{V_s} = \frac{N_p}{N_s} \quad (4.1)$$

Where; V_p = primary (input), V_s = Secondary Output Voltage, N_p = number of turns on primary coil while N_s is the number of turns on the secondary coil.

4.4.2 Rectification

A rectifier converts an AC waveform into a DC waveform. There are two types of rectifier namely; half wave and full wave rectifiers. Both use diodes to convert AC into DC with the diode only allowing current to flow through it in one direction. In this case the diode is said to be forward biased and the only effect on the signal is that there will be a voltage drop of around 0.7 V. In the opposite direction the diode is said to be reverse biased and no current will flow through it. A rectifier is referred to as a full-wave rectifier because it uses the entire AC wave (both positive and negative sections).

4.4.3 Capacitor filter

In most circuits, smoothing is performed by a large value electrolytic capacitor connected across the dc supply to act as a reservoir such that it supplies current to the output when the varying dc voltage from the rectifier is falling. An unsmoothed varying dc (dotted line) and a smoothed DC (solid line) are illustrated in figure 4.3

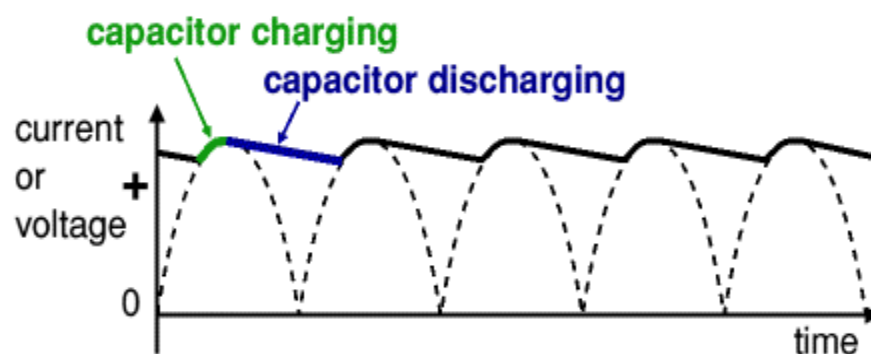


Figure 4.3: Smoothed varying direct current (www.kpsec.freeuk.com)

Smoothing significantly increases the average DC voltage to almost the peak value (1.4 multiplied by the root mean square value). It is not perfect because the capacitor voltage falls a little as it discharges, giving a small ripple voltage. A large capacitor

gives fewer ripples and must be doubled when smoothing half-wave DC. To obtain the value of the smoothing capacitor, the equation below is used;

$$C = \frac{5I_0}{(V_s F)} \quad (4.2)$$

Where: C = smoothing capacitance in Farads (F), I_0 = output current from the supply in amperes (A), V_s = supply voltage in volts (V) and F= frequency of the AC supply in Hz.

4.4.4 Regulation

While a number of circuits can tolerate a smoothed power supply, others must have a completely regulated supply with no ripple voltage. The LM78XX series of three terminal regulators has several fixed output voltages making them useful in a wide range of applications. Although they are primarily designed as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents. The LM78XX is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is also included to limit the peak output current to a safe value.

4.5 Liquid Crystal Displays (LCDs)

A liquid crystal display (LCD) is a low cost module that can be interfaced to a microcontroller to show register contents as characters and numerals. Although seven segments are easy to use they are bulky and quite limited in the set of characters that they can display. When many letters and digits are to be displayed, seven segment displays become inadequate. The LCDs are preferred to seven segments and other multi-segment LEDs because:

- i) They are economical in that they exhibit low power consumption

- ii) Have small foot print and the ability to display both characters and graphics in contrast to LEDs, which are limited to numbers and a few characters
- iii) They possess an internal refreshing controller. This relieves the CPU from continuously sending data to the display.

The most common type of LCDs allow light to pass through when activated. A segment is activated when a low frequency bipolar signal in the range of 30 to 1000Hz is applied to it. The polarity of this voltage must alternate or the Liquid Crystal Display will be unable to change rapidly. On applying a voltage across the segment, an electrostatic field which aligns the crystals in the liquid is set up. This alignment allows light to pass through the segment. If no voltage is applied across a segment the crystals appear to be opaque because they are randomly aligned. Random alignment is assured by the AC excitation voltage applied to each segment.

Most character liquid crystal displays in common use are compatible with the Hitachi 44780 controller, making a wide range of microcontrollers to incorporate libraries that make displaying messages as easy as a single line of code. A 2x16 LCD implies that it can display sixteen characters per line and there are two such lines. In this LCD, each character is displayed in a 5x8 pixel matrix with the LCD having two registers; command and data registers. Most liquid crystal display modules available in the market are compatible with the Hitachi 44780 controller. These LCDs have 14- or 16-pin connection shown on figure 4.4. This connection may be a single or double row header at one edge of the LCDs' printed circuit.

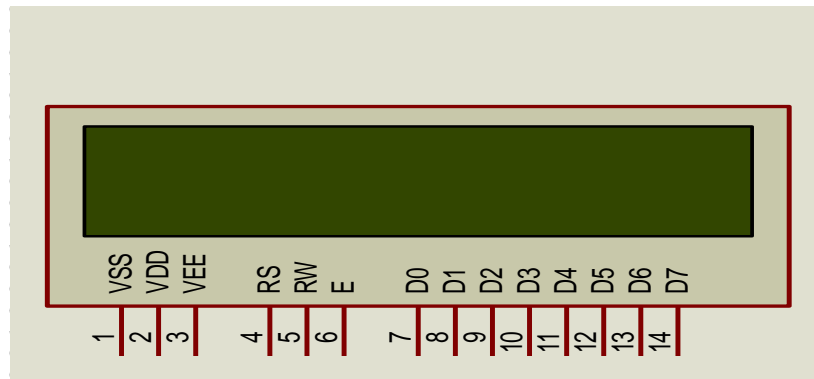


Figure 4.4: Schematic diagram of an LCD.

The command register stores the command instructions given to the LCD to do a predefined task such as initializing it, clearing its screen, setting the cursor position and controlling the display. On the other hand, data register stores the data to be displayed on the LCD. This data is the American Standard Code for Information Interchange (ASCII) value of the character to be displayed (Lovine, 2011). The LCD requires 3 control lines from the microcontroller namely:

- (i). Enable (E)
- (ii). Read/ $\overline{\text{Write}}$ (R/ $\overline{\text{W}}$)
- (iii). Register select (RS)

The Enable line allows access to the display through R/ $\overline{\text{W}}$ and RS lines. When this line is low, the LCD is disabled and ignores signals from R/ $\overline{\text{W}}$ and RS. The LCD checks the state of the two control lines and responds accordingly in the event the E line is high. The R/ $\overline{\text{W}}$ line on the other hand determine the direction of data between the LCD and the microcontroller. When this line is low, data is written to the LCD and when high data is read from the LCD. The RS is used to interpret the type of data on data lines. For instance when it is low, an instruction is being written to the LCD but when high a character is being written to the LCD.

4.6 Global System for Mobile Communications (GSM)

The acronym GSM stands for Global System for Mobile Communications. The original French acronym stands for Groupe Spécial Mobile. It was developed in 1984 as a standard for a digital mobile telephone system that could be used across Europe. However it is now an international standard for mobile telephone service. It has the advantage of high mobility as subscribers can easily roam worldwide access any GSM network. The GSM offers a number of services including voice communications, Short Message Service (SMS), fax, voice mail, and other supplemental services such as call forwarding and caller ID. Currently there are several bands in use in GSM. The frequency bands at 450 MHz, 850 MHz, 900 MHz, 1800 MHz, and 1900 MHz are the most common ones. Some bands also have Extended GSM (EGSM) bands added to them, increasing the amount of spectrum available for each band. GSM uses Time Division Multiple Access (TDMA) as its access scheme. This is how the MS interfaces with the network. TDMA is the protocol used on the Air (Um) Link. GSM uses Gaussian Minimum-Shift Keying (GMSK) as its modulation methods. Time Division means that the frequency is divided up into blocks of time and only certain logical channels are transmitted at certain times. The time divisions in TDMA are known as time Slots.

4.6.1 GSM Network Architecture

A GSM network is made up of multiple components and interfaces that facilitate sending and receiving of signaling and traffic messages. It is a collection of transceivers, controllers, switches, routers, and registers. A Public Land Mobile Network (PLMN) is a network that is owned and operated by one GSM service provider or administration.

4.6.2 Mobile Station (MS)

The Mobile Station (MS) is made up of two components:

4.6.2.1 Mobile Equipment (ME)

This is the physical phone itself. The phone must be able to operate on a GSM network. Older phones operated on a single band only. Newer phones are dual-band, triple-band, and even quad-band capable. A quad-band phone has the technical capability to operate on any GSM network worldwide.

A Subscriber Identity Module (SIM) card is inserted into a phone or a GSM modem. It carries information specific to the subscriber. It can be removed from one phone and inserted into another GSM capable phone and the subscriber will get the same service as always (Kahabka, 2012).

4.6.2.2 Base Transceiver Station (BTS)

This is the Mobile Station's access point to the network. It is responsible for carrying out radio communications between the network and the mobile station. It handles speech encoding, encryption, multiplexing (TDMA), and modulation/demodulation of the radio signals. It is also capable of frequency hopping. A BTS will have between 1 and 16 transceivers (TRX), depending on the geography and user demand of an area. One BTS usually covers a single 120 degree sector of an area. Usually a tower with 3 BTSs will accommodate all 360 degrees around the tower. However, depending on geography and user demand of an area, a cell may be divided up into one or two sectors, or a cell may be serviced by several BTSs with redundant sector coverage. A BTS is assigned a Cell Identity. The cell identity is a 16-bit number that identifies that cell in a particular Location Area. The cell identity is part of the Cell Global

Identification (CGI), which is discussed in the section on the Visitor Location Register (VLR).

4.6.3 Base Station Controller (BSC)

It controls multiple Base Transmission Stations. It handles allocation of radio channels, frequency administration, power and signal measurements from the MS, and handovers from one BTS to another (if both BTSs are controlled by the same BSC). It reduces the number of connections to the Mobile Switching Center (MSC) and allows for higher capacity connections to the MSC. A BSC may be co-located with a BTS or it may be geographically separate. It may even be co-located with the Mobile Switching Center (MSC).

4.6.4 Mobile Switching Center (MSC)

This is the heart of the GSM network. It handles call routing, call setup, and basic switching functions. An MSC handles multiple BSCs and also interfaces with other MSC's and registers. It also handles inner-BSC handoffs as well as coordinates with other MSC's for inter-MSC handoffs. There is another important type of MSC, called a Gateway Mobile Switching Centre (GMSC). It functions as a gateway between two networks. The GSM architecture is illustrated in figure 4.5

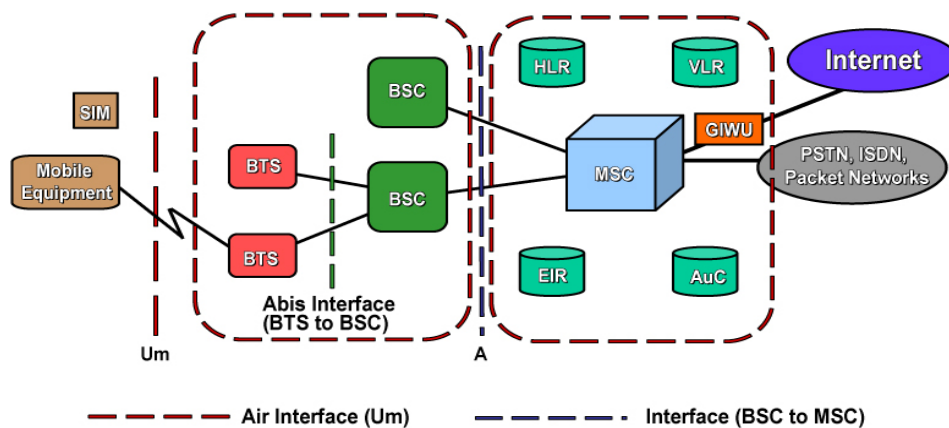


Figure 4.5 GSM architecture after (Kahabka, 2012).

4.7 Flow sensors and measurements

4.7.1 Flow measurement

There are two basis of measuring flow, these are either volumetric or the weight basis.

Liquids are measured either in volume rate or in weight rate.

The basic relationship for determining the liquid's flow rate is given in equation 4.1:

$$Q = VA \quad (4.1)$$

where Q = Liquid flow through the pipe

V = Average velocity of the flow

A = Cross-sectional area of the pipe

The other factors that affect liquid flow rate include viscosity, density and the friction of the liquid in contact with the pipe. Direct measurements of liquid flow can be made with positive-displacement flow meters. The total flow is an accumulation of the measured increments, which can be counted by mechanical or electronic techniques. The performance of flow meters is also influenced by a dimensionless unit called the Reynold's Number. It is defined as the ratio of the inertial force to drag force.

The equation is:

$$R = \frac{3160QG_t}{Dh} \quad (4.2)$$

where: R is the Reynold's number; Q is the flow rate; G_t is specific gravity; D is the inside pipe diameter and h is the viscosity. There are many types of flow meters of which turbine flow meters, variable area flow meters and target flow meters are the most common.

4.7.2 Turbine Flow Meters

These belong to the category of inferential flow meters. The fluid to be measured enters the flow meter, then passes through a rotor. It impinges on the blades of turbine

(rotor), imparting a force to the blade surface which causes the rotation of the rotor. At a steady rotational speed, the speed of the rotor is directly proportional to the fluid velocity, and hence to volumetric flow rate. The speed of rotation is monitored in most of the meters by a magnetic pick-up coil, which is fitted to the outside of the meter housing. The magnetic pick-up coil consists of a permanent magnet with coil windings which is mounted in close proximity to the rotor but external to the fluid channel. As each rotor blade passes the magnetic pick-up coil, it generates a voltage pulse which is a measure of the flow rate, and the total number of pulses gives a measure of the total flow. Using digital techniques, the electrical voltage pulses can be processed so that a zero error characteristic of digital handling is provided from the electrical pulse generator to the fluid readout. The number of pulses generated is given as:

$$n_p = \frac{T_p f}{Q} \quad (4.3)$$

where n_p are the pulses per volume unit; T_p is the time constant; Q is the volumetric flow rate and f is the frequency in Hz.

4.7.3 Flow sensor

A flow sensor detects and measures water flowing through pipes. In this system in order to calculate the flow rate a Hall Effect sensor is attached to $G_{1/2}$ water flow sensor. The rotor blades rotate as the water flows through the flow sensor. A magnetic field is produced and therefore an alternating current pulse is generated, which is then converted into the digital output with the help of Hall Effect sensor placed just after the turbine. The number of pulses generated per litre can be determined by software programming. The pulses produce an output frequency that is directly proportional to the volumetric flow rate. A $G_{1/2}$ flow sensor is shown in figure 4.6.

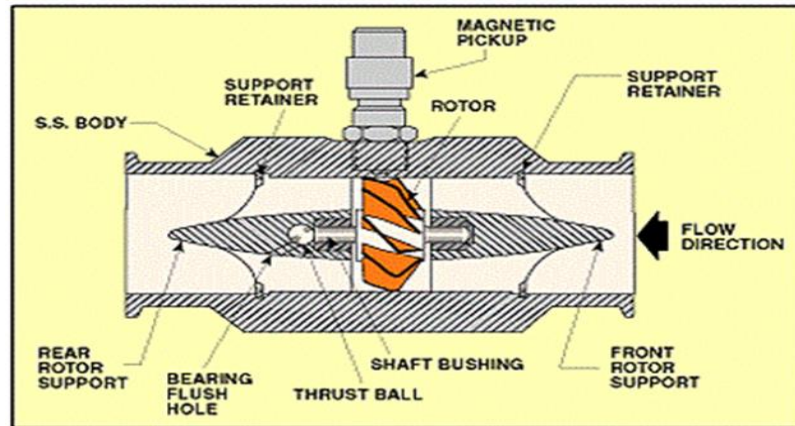


Figure 4.6: Internal structure of $G_{1/2}$ water flow sensor (www.enggcyclopedia.com)

4.8 Solenoid valve

Most solenoid valves operate on a digital principle. They have two distinct states. These are the coil activation by an electrical current and the valve in a resting position without any current flow. The valve functions are defined from the resting position. A solenoid valve is shown in figure 4.7.

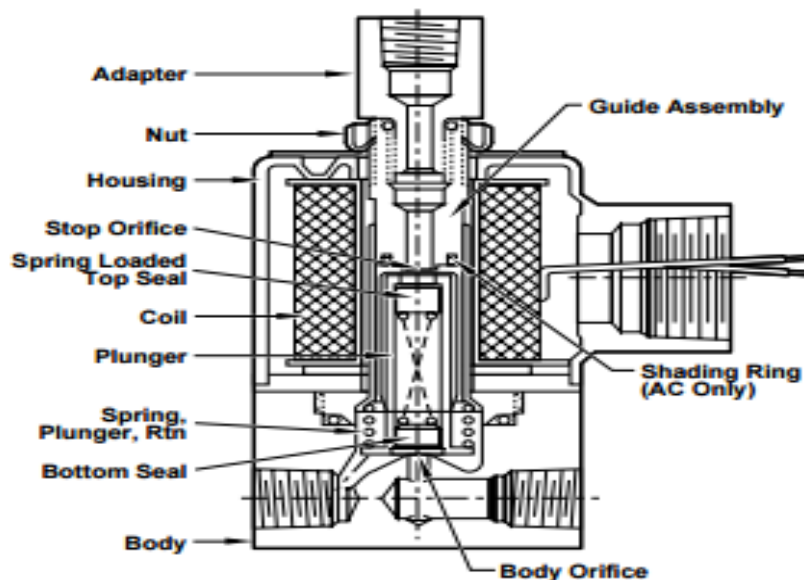


Figure 4.7: A solenoid valve (Parker, 2011).

The direct acting or pilot operated solenoid valves may have two functions. They are:

i) Normally closed (NC)

A solenoid valve is normally closed if there is no flow across the valve in its resting position.

ii) Normally open (NO)

A solenoid valve is said to be normally open when it enables fluid to pass in its resting position.

CHAPTER FIVE

RESEARCH METHODOLOGY

5.1 Introduction

This chapter describes the methodologies used for the design of the hardware and software parts. The main hardware parts are the flow sensor and how it is interfaced to the microcontroller and the GSM interface. Others are the valve control mechanism and the output display. In addition, a regulated power supply is also implemented. The software part is mainly Mikroc Integrated Development Environment.

5.2 Hardware components

The block diagram of the main parts of the fabricated meter system is shown in figure 5.1.

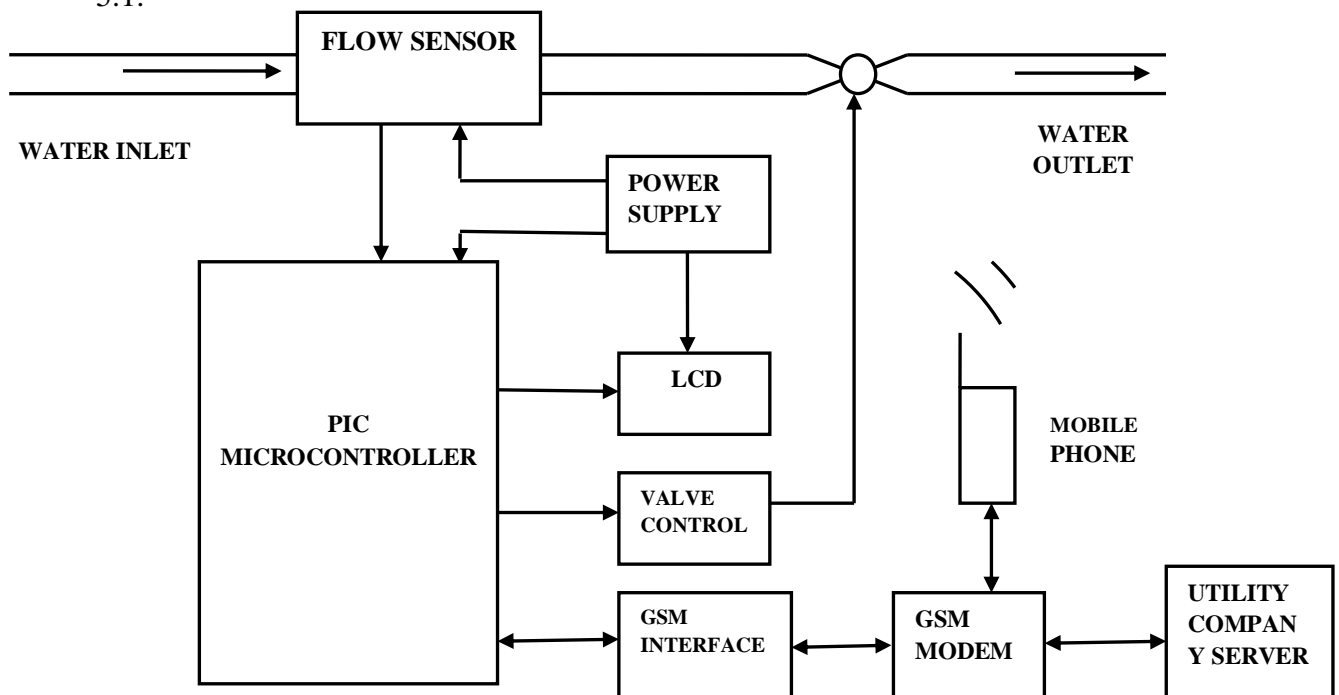


Figure 5.1: Block diagram of the intelligent water meter system

The hardware part consists of the flow sensor, water sensor, PIC18F4550 Microcontroller, LCD, valve control, power supply and a GSM modem. The water

meter system has a control system that accepts input signals from sensors and takes a decision that regulates a valve mechanism.

5.2.1 Flow sensor

The $G_{1/2}$ flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. The Hall-effect is an ideal sensing technology. When subjected to a magnetic field, it responds with an output voltage proportional to the magnetic field strength. When water flows through the sensor its rotor rolls. The speed changes with different rate of flow. The hall-effect sensor gives a corresponding pulse signal. The pulse signal is then sent to the PIC 18F4550 microcontroller. The pin configuration of the sensor shown in figure 5.2 and has different pins as follows.

- i) Pin 1 Yellow-Signal
- ii) Pin 2 Black- Ground
- iii) Pin 3 Red - V_{CC}

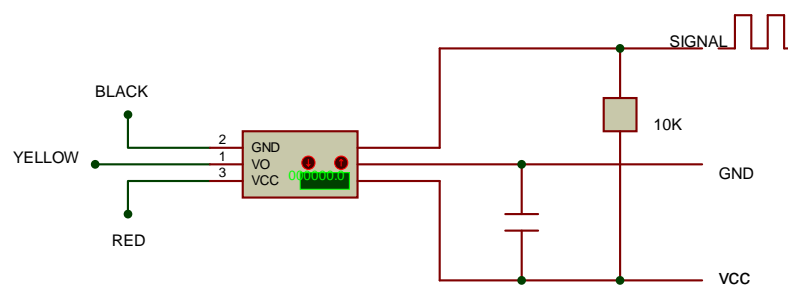


Figure 5.2: Pin configuration of the $G_{1/2}$ flow sensor

The yellow wire is connected to the Pin 33 of the microcontroller to introduce the signal to the PIC18F4550 microcontroller. Figure 5.3 shows the interface between the flow sensor and the PIC18F4550.

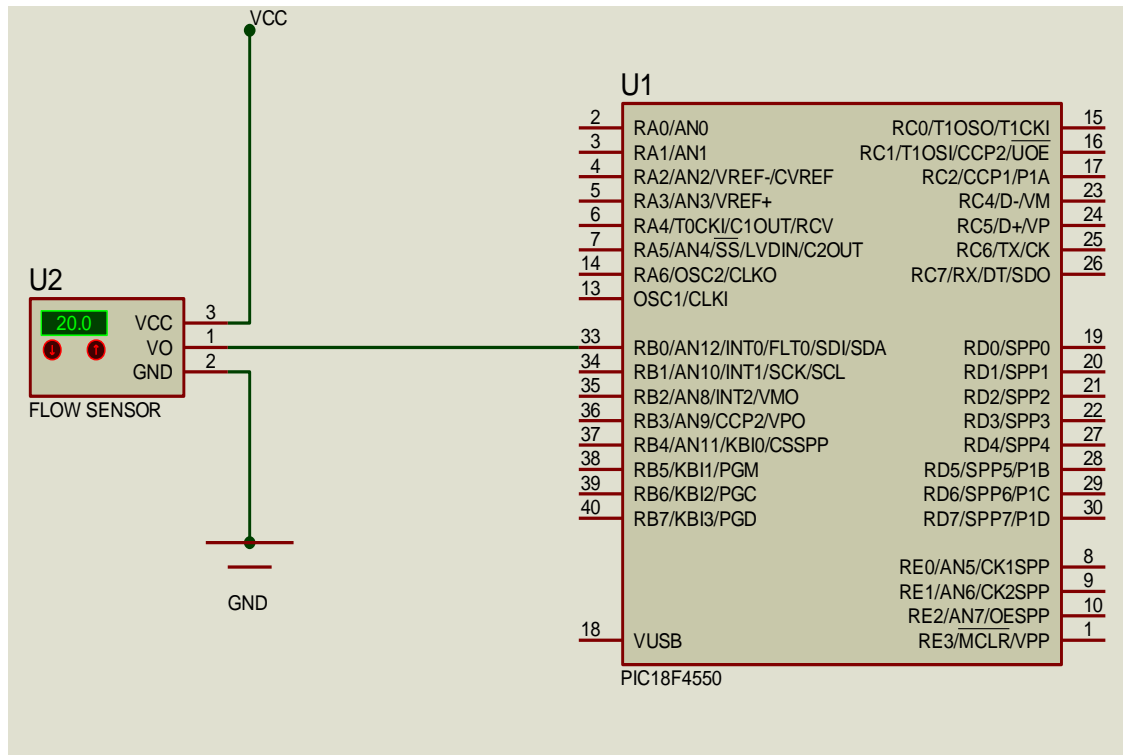


Figure 5.3: Flow sensor interface to PIC18F4550

5.2.2 Liquid Crystal Display to PIC 18F4550 microcontroller interface

The 2x16 character LCD has two rows of 16 characters each. One popular 2x16 LCD is the 1602ZFA of the Shenzhen Eone Electronics limited which is pin compatible with the Hitachi HD44780 and is equipped with a backlight function. The LCD pins and functions are given in the table 5.1. It has 16 pins with register select, read/write and enable control lines. In addition it has a display contrast voltage line. The 4-bit interface was employed because it saves on port pins which could otherwise be used for other additional functions. Pins 15 and 16 were employed to enable visibility in the dark and for characters to be viewed with ease from a distance. The third pin (V_o)

is connected to a variable resistor to control the LCD contrast, while the R/W (Read/Write) pin is connected to ground indicating that the LCD is receiving data only. The display receives ASCII codes for each character at D4 to D7 which is connected directly to the 3 PORTB pins and 1 PORTD pin of the PIC18F4550 microcontroller. To initialize the LCD, the register select line RS must be set to logic 0.

Table 5.1: Liquid Crystal Display pins configuration

Pin no.	Symbol	Function	Pin no.	Symbol	Function
1	V _{SS}	Ground	9	D2	GND
2	V _{DD}	Supply 2.7V to 5.5V	10	D3	GND
3	V _O	Display contrast	11	D4	RB3
4	RS	Register Select	12	D5	RB2
5	R/W	Read/Write	13	D6	RB1
6	E	Enable	14	D7	RD7
7	D0	GND	15	N/A	Backlight (+5V)
8	D1	GND	16	N/A	Backlight (GND)

The 8-bit code for each ASCII character is sent in two halves; high nibble first, low nibble second. Although this makes the software only slightly more complex, it saves on I/O pins and allows the LCD to be driven using only six lines as shown in figure 5.4.

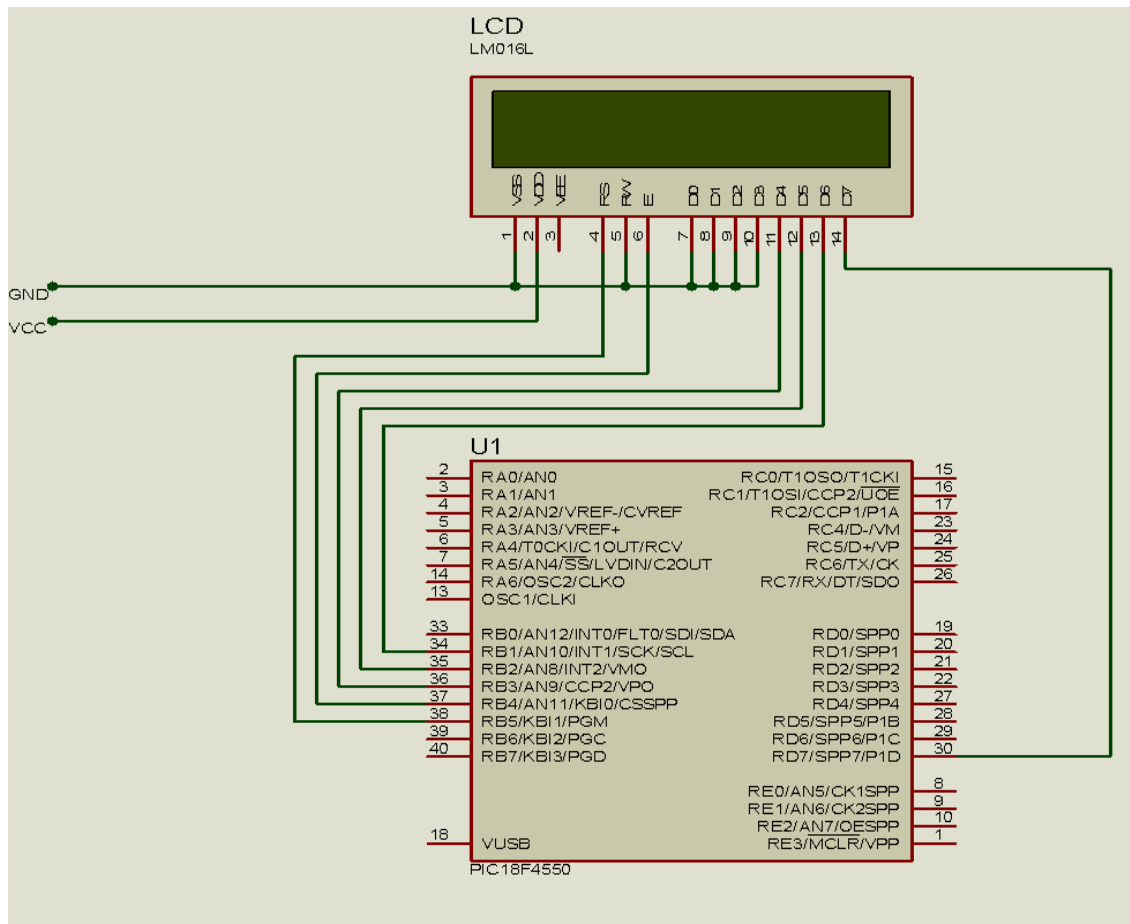


Figure 5.4: Liquid Crystal Display to PIC 18F4550 microcontroller interface

5.2.3 GSM interface to PIC18F4550 microcontroller

A GSM modem accepts a subscriber identity Module (SIM) card. It is based on commands that always start with AT (Attention) and finish with a character (CR). The GSM Module communicates with the microcontroller and mobile phone through the UART port. The GSM module SIM 900 has been employed in the proposed system. The SIM900 is a Quad-band GSM/GPRS. It works on the frequencies; 850 MHz, 900 MHz, 1800 MHz and 1900 MHz. The GSM Module SIM 900 can be directly interfaced with PIC Microcontrollers, through the three lines; TX, RX, GND. The transmit signal of the serial port of the microcontroller is connected with the transmit signal (TxD) of the serial interface of GSM Modem while receive signal of the

microcontroller serial port is connected with receive signal (RxD) of serial interface of GSM modem. Figure 5.5 shows how the GSM is interfaced with PIC18F4550 microcontroller.

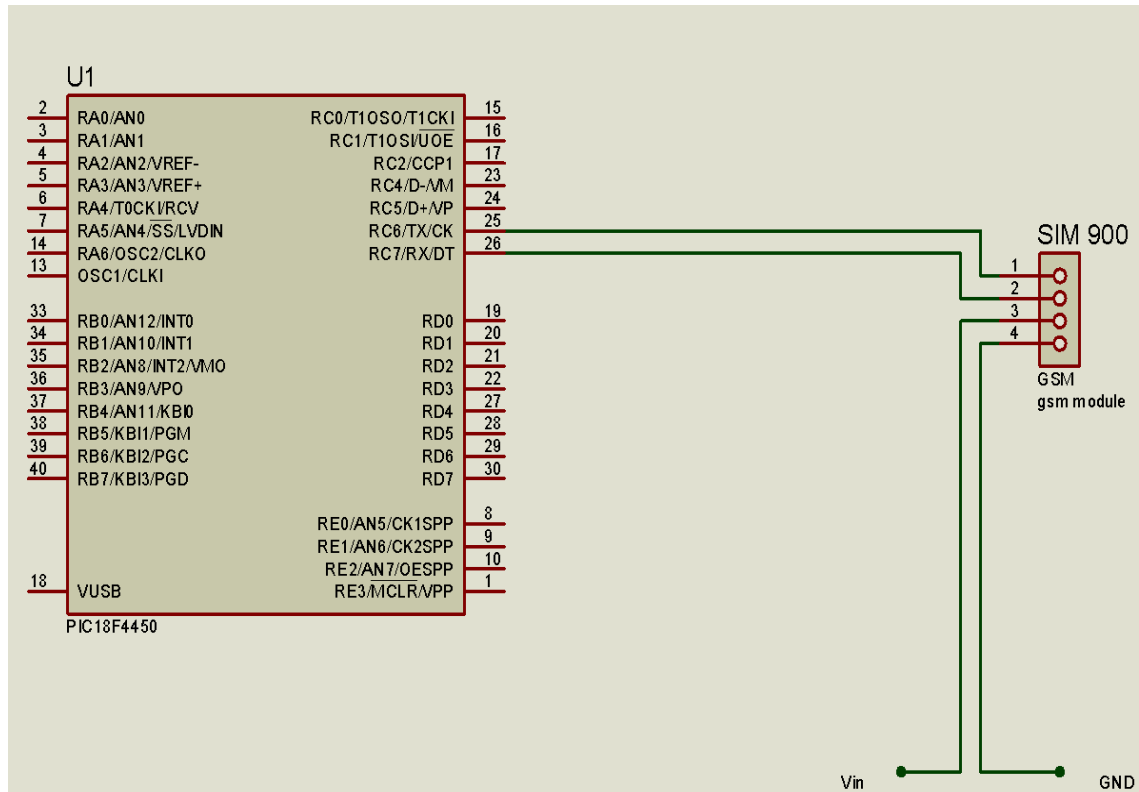


Figure 5.5: GSM interface to PIC18F4550 microcontroller

5.2.4 Solenoid control valve

A flow control valve regulates the flow or pressure of a fluid. In this investigation a solenoid valve has been used. It is an electromechanically controlled valve has a moveable ferromagnetic core at the centre usually referred to as a plunger. An electric current through the coil creates a magnetic field. The magnetic field exerts a force on the plunger which pulls it toward the centre of the coil thereby opening the orifice. When interfacing the solenoid valve to the PIC18F4550 a relay is employed. This is because a relay can be used to switch higher power devices like solenoids. The relay is powered by a separate 12V supply source to enhance the solenoid valve to be

interfaced with the microcontroller. A transistor is used which is wired as a switch which carries the current required for the operation of the relay. A back emf suppression diode is used across the relay contacts to prevent damage of the transistor when the relay switches off. When the connections are effected, the relay which is connected to the solenoid is then interfaced with the PIC18F4550 microcontroller. Figure 5.6 shows the interface of the microcontroller and the relay to the solenoid valve.

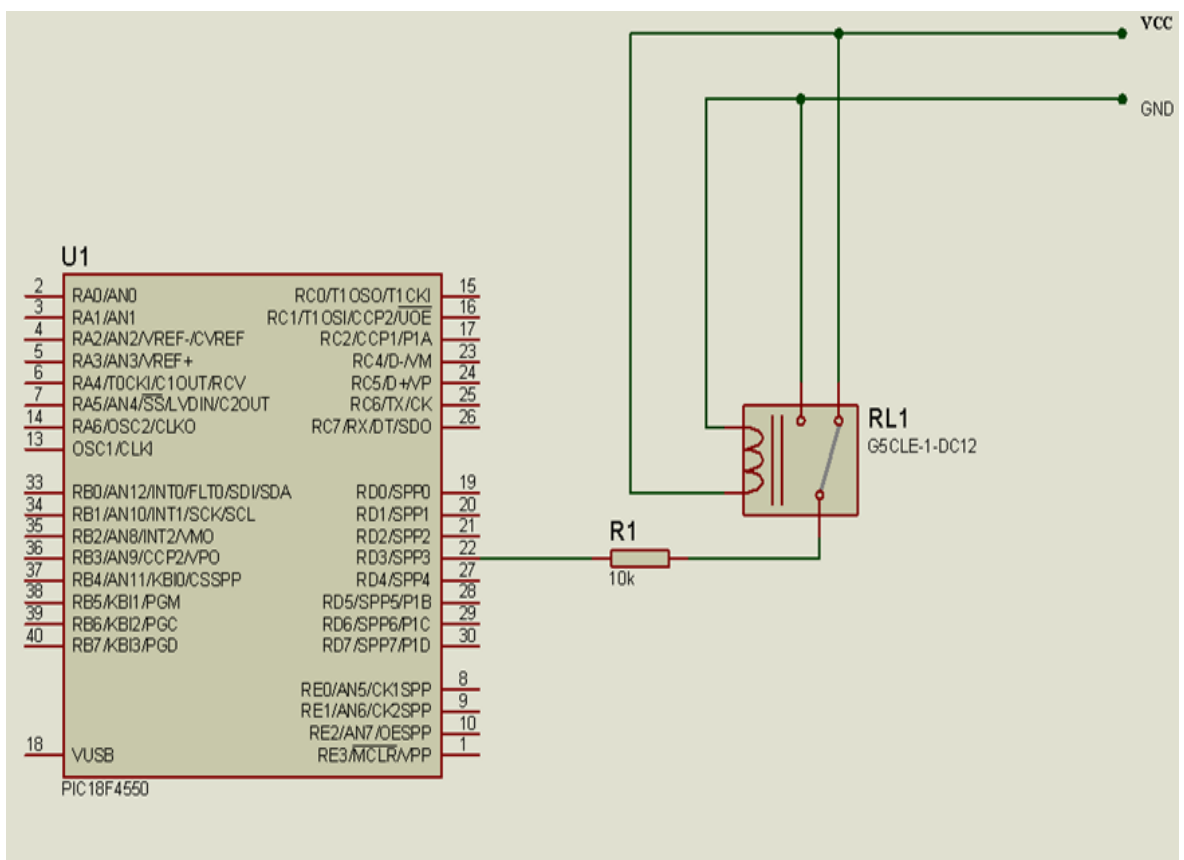


Figure 5.6: Interface of solenoid valve to the PIC18F4550

5.3 Software design

5.3.1 Flowchart for the system software

In order to write the programme code to control the microcontroller operations, the following algorithm has been proposed.

- i. Start: In this step, the microcontroller is reset and the first instruction to be executed shall be at address 0001h
- ii. Initialize the LCD: The LCD display is cleared the cursor moves to the start of first line
- iii. Initialize GSM module. It searches for network. A short message “GSM Test” is received. This indicates that the module is ready.
- iv. The LCD displays the Message “FLOW RATE.....” and “LITRES”.....
- v. Check if rotor of the $G_{1/2}$ flow meter has rolled.
- vi. If YES then display Volume change in Litres.
- vii. If NO display no change in volume on the LCD and go to step v
- viii. An SMS of the number of litres is received automatically after every minute via GSM network.
- ix. If an SMS is not received via the GSM network go to step iii
- x. Send an SMS to close the control valve if a customer has not paid the bills. This closes the control valve preventing water from flowing.
- xi. If the valve does not close go to step iii.
- xii. If the customer has paid the bills send an SMS to open the control valve.
- xiii. End

The above steps are illustrated in the flow chart given in fig 5.7.

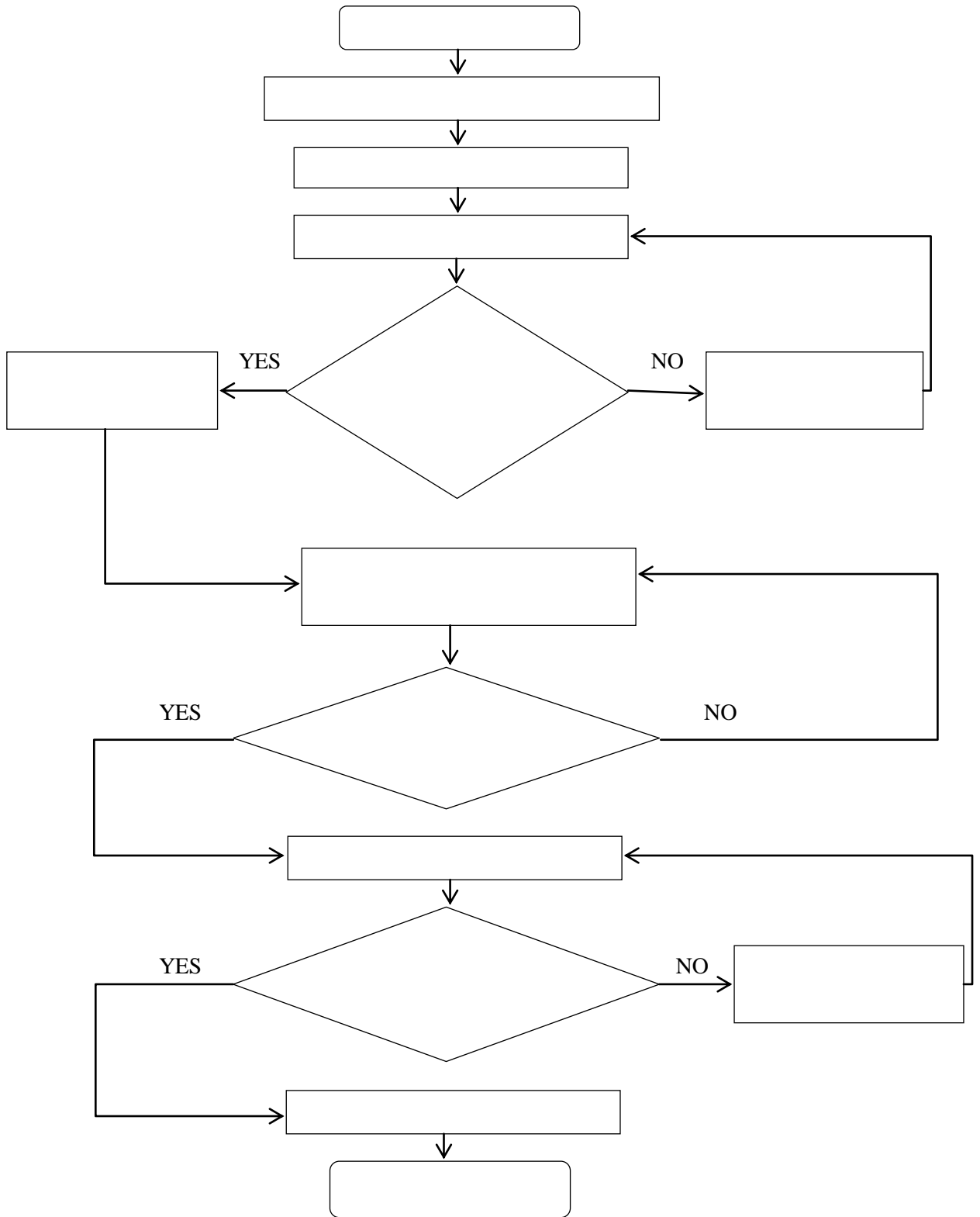


Figure 5.7: Flowchart for the system software design

5.3.2 Programme code development

The programme code was written as a text file and converted to a machine code using the MikroC Integrated Development Environment (IDE). The MikroC IDE is a powerful, feature rich development tool for PIC microcontrollers. The process involves the following steps.

- i. Creating a new file.

The file icon is selected, then New file Icon is clicked, then after clicking new project a window appears that allows creation of a new project. The window details are illustrated in figure 5.8.



Figure 5.8: A screenshot of MikroC software illustrating creation of new project

- ii. After clicking next, a window appears that allows the selection of the device name and frequency of the device clock. A new project file is then created and a project directory selected where the necessary files needed for the project are stored as shown in the window illustrated in figure 5.9.

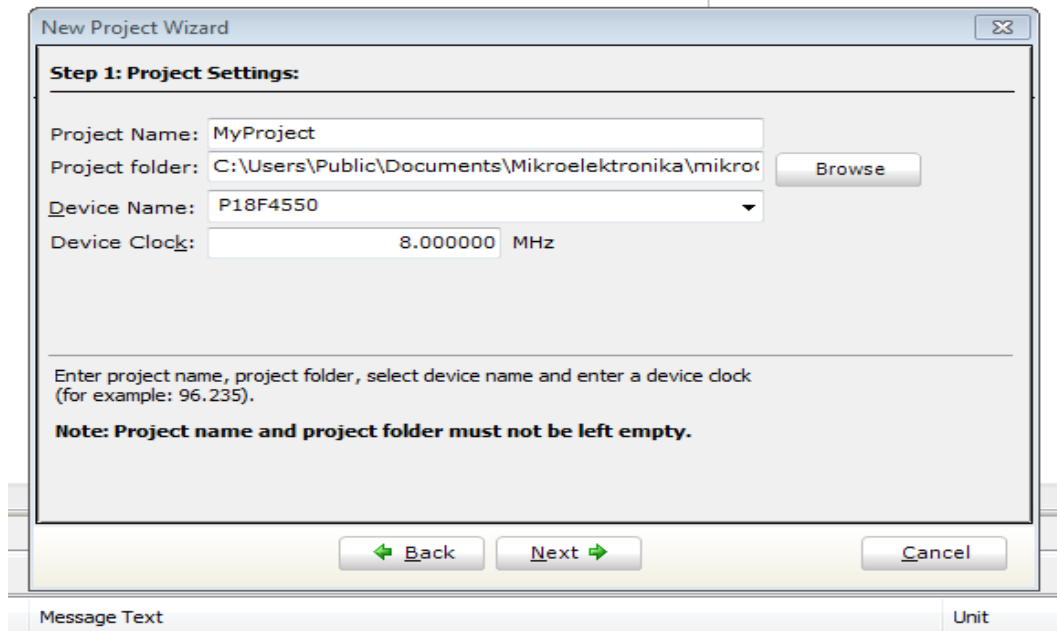


Figure 5.9: A screenshot illustrating selection of project name and project folder

- iii. The next step is adding or removing any files where need be. The window that is used for this purpose is shown in figure 5.10.

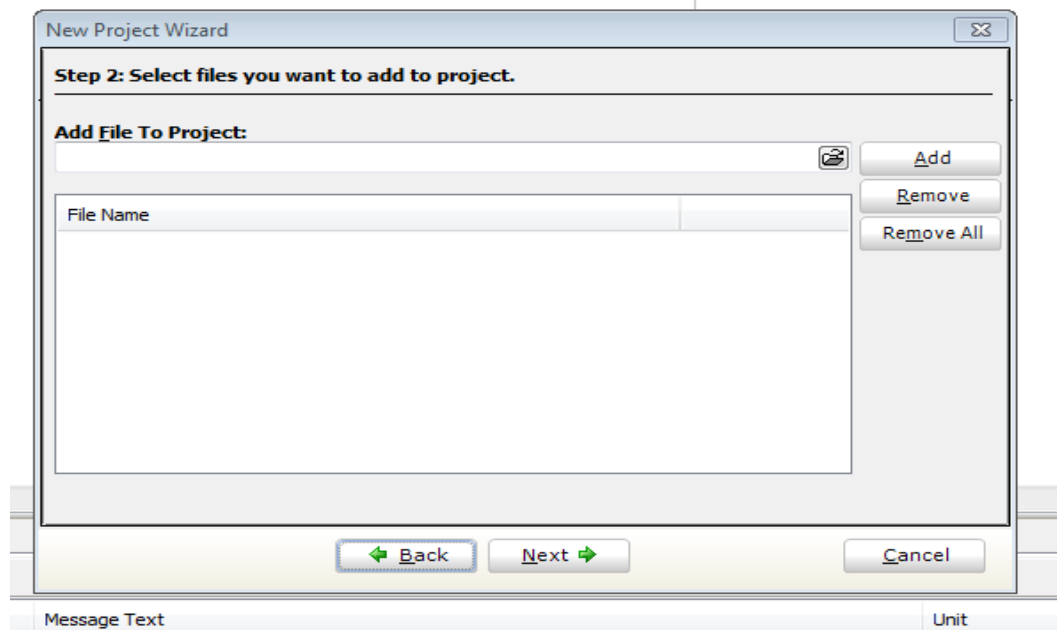


Figure 5.10: A screenshot of MikroC with a provision for addition or removal of files

- iv. The next step is selecting the initial state for library manager. The MikroC IDE provides built in and library routines in addition to ready-to-use code

examples. This will enhance faster and easier application. The window for this purpose is illustrated in figure 5.11.

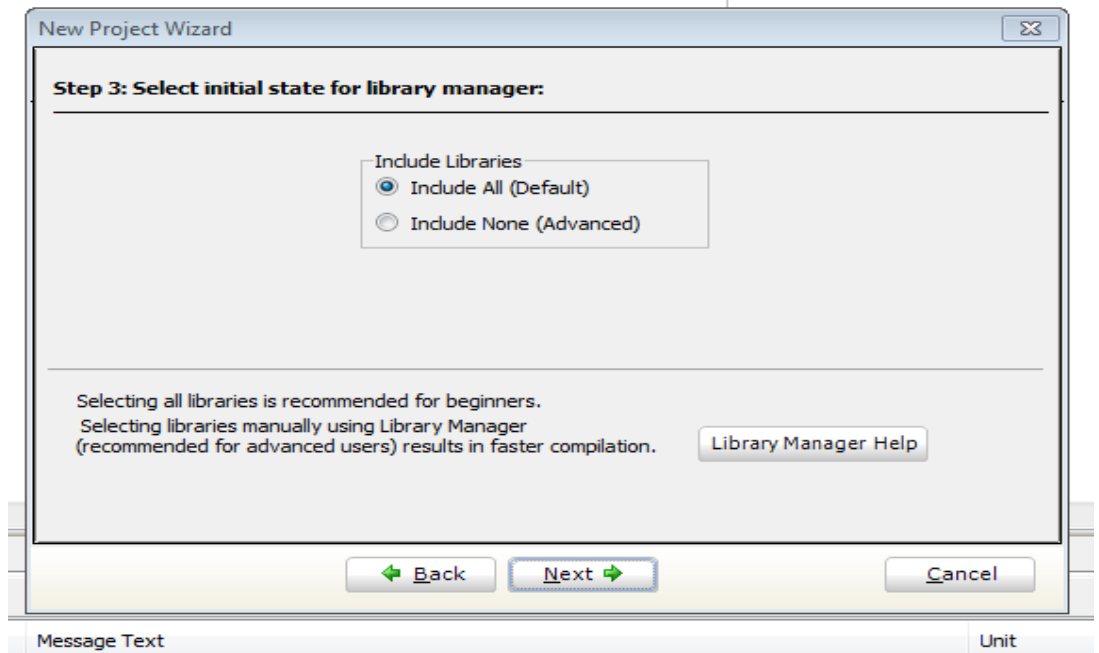


Figure 5:11: A Mikroc screen shot illustrating library selection

- v. Once libraries are included the finish button is activated to finalise the process. A window illustrating a successfully created new project is shown in figure 5.12.

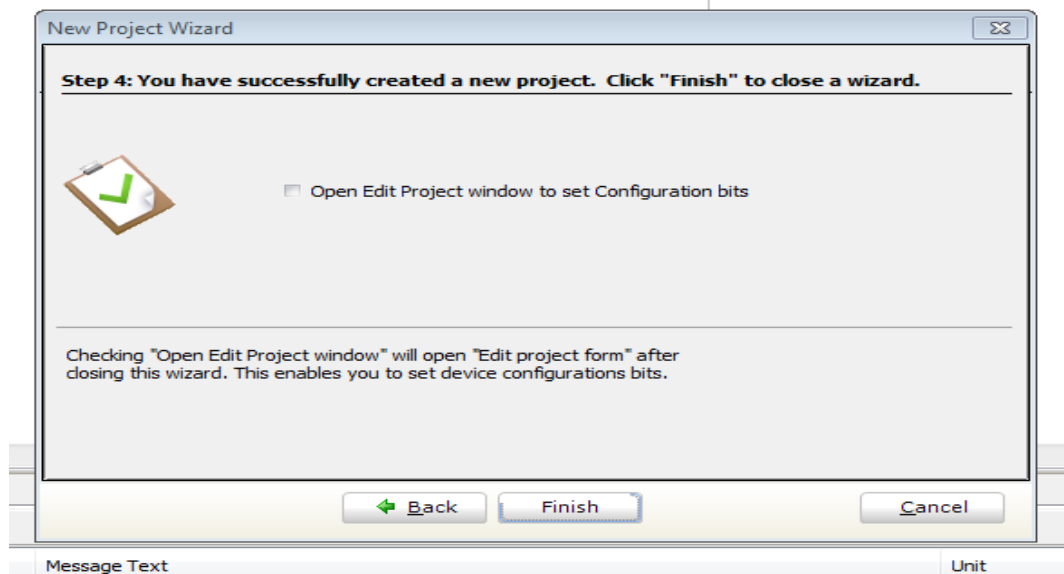


Figure 5:12: A screenshot showing successful creation of a new project

- vi. A new file is opened from the file menu where the programme code is then written. The next step is compilation. This is achieved by selecting the Build icon. The progress bar appears displaying the status of compilation. The compiler generates a hex file that is burned to the PIC18F4550 memory using the PICkit2 programmer.

CHAPTER SIX

RESULTS AND DISCUSSIONS

6.1 Introduction

This chapter presents the results obtained on testing the designed system. The results were obtained in the following parts of the system; the flow sensor, GSM module and the control valve. Figure 6.1 shows a picture of the designed system while testing.

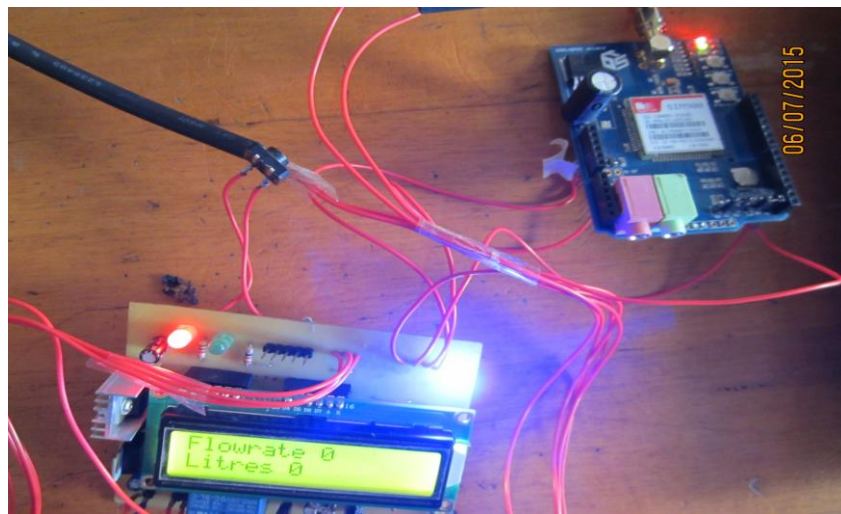


Figure 6.1: A picture of the designed system

6.2 Programme Test

The programme was first tested in simulation mode before downloading it to the target board. Once edited and assembled in Mikroc, the programme is run with the special function registers and stopwatch displayed. The program was written, assembled and simulation runs with the source code. The programme responded well. The simulation test allowed the syntax and logic to be checked before downloading the programme. Any syntax errors were detected by the assembler, with the line number and the error being indicated on the output window. On confirmation of

successful build on the project, the resulting hex file was downloaded into the PIC18F4550 programme memory using the PIC kit2.

6.3 System response to water flow at different frequencies

In order to determine and transmit the rate of flow and volume of water, a system comprising of G1/2 flow sensor, LCD, GSM module has been used. When the control valve was opened the water passed through the rotor, causing it to rotate at the speed equivalent to velocity of flow. The rotor speed changes with different rate of flow of water. As each blade passed through the magnet, a magnetic field is created at the base of the Hall sensor and thus pulses are generated. These pulses produce an output frequency proportional to the volumetric flow through the sensor.

To test the system, water is allowed to flow at different rates. Figure 6.2(a) displays pulses generated at a lower frequency while figure 6.2(b) displays pulses generated at a higher frequency.

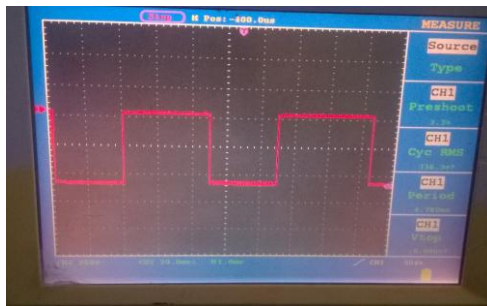


Figure 6.2(a) Pulses at lower frequency

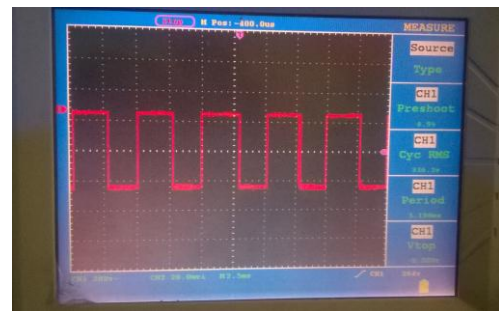


Figure 6.2(b) Pulses higher frequency

6.4 Flow rate calculation.

To read flow rate at various frequencies, a C program was written. A valve was used to control the amount of water flowing through the sensor as shown in figure 6.4.



Figure 6.3: A picture of the valve used to control flow rate

As water flow at different frequencies, the time taken to fill a 1 litre container was recorded. A water valve was used during this test to control the amount of water flowing. It was opened at different intervals but randomly. When opened, the frequency when the lowest amount of water flowing was found to be 6 Hz. The opening of the valve was increased and various frequencies were measured. The maximum frequency when the valve was completely opened was found to be 64Hz. Table 6.1 shows the relationship between Frequency in Hz and Time (in sec).

Table 6.1: Time(s) taken to fill 1 L at different frequencies

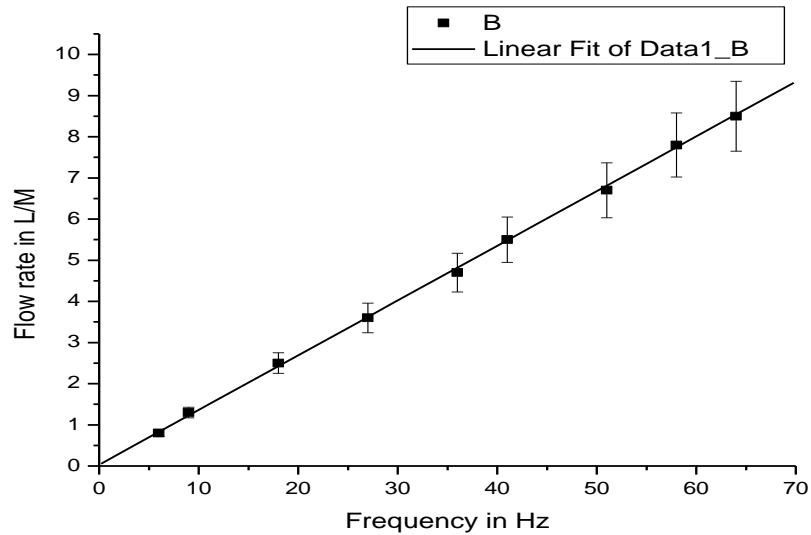
Frequency (Hz)	Time (Sec)
6	75
9	50
18	24
27	16
36	12
41	11
51	9
58	8
64	7

From Table 6.1 the flow rate for each of the frequencies can be computed. The flow rate is in litres/minute and is determined to an accuracy of $\pm 5\%$. Table 6.2 shows the relationship between the frequency and the flow rate.

Table 6.2: Flow rate at different frequencies

Frequency (Hz)	Flow rate (litres/min) ($\pm 5\%$)
06	0.8
09	1.2
18	2.5
27	3.6
36	4.7
41	5.5
51	6.7
58	7.5
64	8.5

From table 5.1 a graph of flow rate in litres/min against Frequency (Hz) is plotted.

**Figure 6.4: Frequency variation with flow rate**

From the graph, the gradient has been determined to be 0.1333. A factor was obtained

from the reciprocal of the gradient as follows $\frac{1}{0.1333} = 7.5$

Using the factor obtained, any flow rate in litres/min can be obtained. It is given by;

$$\text{Flow rate in litres/ min} = \text{Frequency in Hz}/7.5$$

A subroutine in C has been written to compute the volume after every second as follows.

$$\text{Volume in litres} = \text{Flow rate in litres/sec} \times 1\text{sec}$$

The program is such that it can store the cumulative volume over a period of time.

To obtain the amount of water flowing, the average flow rate is multiplied by the time taken.

To determine cumulative quantity of water over a unit period, a programme code was written and burned in the PIC18F4550.

6.5 The GSM Module

For this research, the GSM module SIM900 was employed. It is a quad-band GSM/GPRS. It works on the frequencies: 850 MHz, 900 MHz, 1800 MHz and 1900 MHz. The GSM module SIM 900 has been interfaced directly with PIC18F4550 microcontroller. The baud rate is configurable from 9600-115200 bps through AT (Attention) commands. An experimental investigation was carried out on the GSM transmission and the readings were recorded as shown on table 6.3. Figure 6.6 shows some of the short messages received through a cell phone.

Table 6.3: LCD and GSM readings

Trials (Minutes)	LCD Reading in (L)	Reading via GSM in (L)
1	0	0
2	2	2
3	4	4
4	6	6
5	8	8
6	8	8
7	8	8
8	10	10
9	11	11
10	11	11

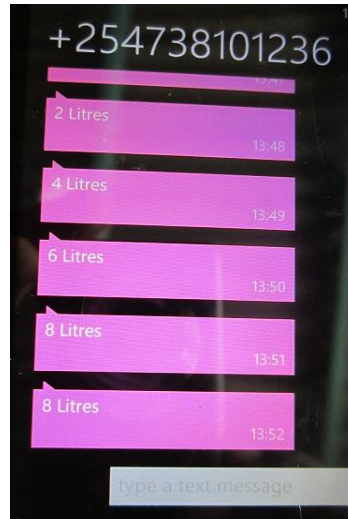


Figure 6.5: A cell phone showing SMS received

6.6 Control valve

This is a valve that opens or closes to either allow water to pass through or prevent water from flowing. A solenoid valve gravity feed was used in this research. The opening and closure of the valve is affected by the execution of a program code. A short message to open or close the valve is sent through the GSM network to the microcontroller that is interfaced to the valve. The control part of the proposed system was noted to function as expected. Whenever a message was sent to close or open the valve functioned as expected. The results obtained from the control valve once an SMS is sent to open or close it are shown in table 6.4.

Table 6.4: Table showing the response of the control valve to SMS

Purpose of the message	Response of the control valve	Remarks
Close the valve	Valve closed	OK
Open the valve	Valve did not open	Network problem
Open the valve	Valve opened	GSM module was reset
Close the valve	Valve closed	Ok
Open the valve	Valve opened	Ok
Close the valve	Valve closed	OK
Open the valve	Valve opened	Ok
Close the valve	Valve closed	Ok
Open the valve	Valve opened	Ok
Close the valve	Valve closed	Ok

6.7 Regulated power supply

A power supply for 5V and 12V has been designed. The 240V from the a.c source is stepped down by the transformer to 12V ac. The 12V ac is rectified to 12V dc and then smoothed by the capacitor. The 12V d.c is fed to the 7805 and 7812 regulators. The connections of the various components is shown in figure 6.6. To ascertain the stability of the power supply, resistors were connected in a cascading manner across the outputs of each of the two regulators. The output voltage was then measured for each of the regulator and the results are shown in table 6.5.

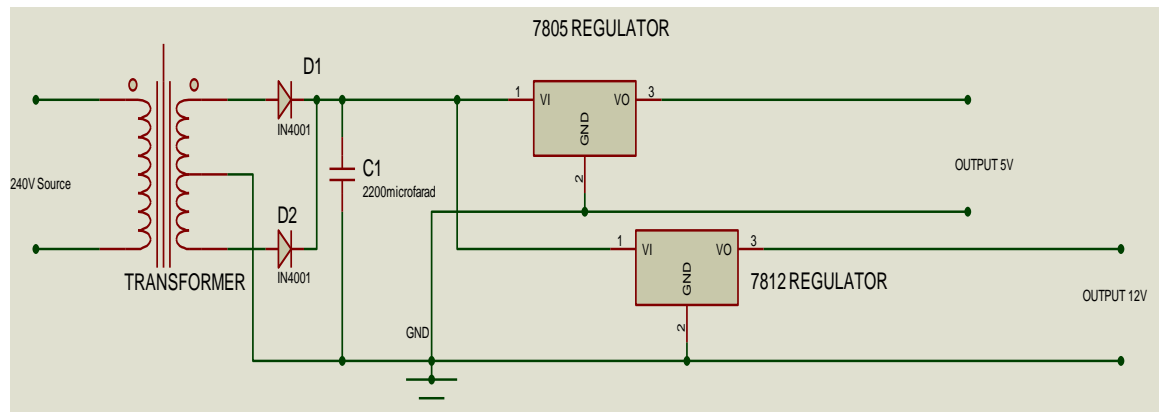


Figure 6.6: Regulated power supply

Table 6.5: Output Voltage against loading of power supply

5 v power supply			12v power supply		
Load(Ω)	V_{out} (V)	Error	Load(Ω)	V_{out} (V)	Error
1K	4.99	0.005	1K	11.9	0.005
2K	4.99	0.005	2K	11.9	0.005
3K	4.99	0.005	3K	11.9	0.005
4K	4.99	0.005	4K	11.9	0.005
5K	4.99	0.005	5K	11.9	0.005
6K	4.99	0.005	6K	11.9	0.005
7K	4.99	0.005	7K	11.9	0.005
8K	4.99	0.005	8K	11.9	0.005
9K	4.99	0.005	9K	11.9	0.005
10K	4.99	0.005	10K	11.9	0.005
∞	4.99	0.005	∞	11.9	0.005

The designed power supply produces 11.9 V and 4.99 V simultaneously at no load, at full load and at all points in between. In this design the percent of regulation is zero % which is the ideal situation hence the output is constant under all load conditions.

6.8 Testing the designed meter system and comparing to other systems

The entire system was tested against a measuring cylinder. The following results were obtained after 15 trials. Table 6.6 shows the volume obtained from the proposed design compared to the actual volume.

Table 6.6: Measured volume of the proposed design versus the actual volume

Trials	Flow rate in L/h	Actual Volume in L	Volume in L displayed by designed meter	Percentage error (%)
1	510	1.00	1.01	1
2	300	1.00	1.00	0
3	60	1.00	0.98	2
4	480	1.00	1.00	0
5	500	1.00	1.02	2
6	100	1.00	1.01	1
7	90	1.00	0.99	1
8	450	1.00	1.00	0
9	400	1.00	1.02	2
10	350	1.00	0.99	1
11	300	1.00	0.99	1
12	120	1.00	0.99	1
13	300	1.00	1.00	0
14	200	1.00	1.01	1
15	350	1.00	1.01	1

From table 6.6 the average error was computed and found to be $0.93 \% \pm 0.7$. The measurements were done at different flow rates between 60 l/h and 510 l/h. For the designed water meter, the accuracy of the volume measured is not related to the flow rate as shown on the table.

From a study conducted on domestic mechanical water meters on accuracy, it was found that their average error of most Class C mechanical meters was about $2\% \pm 1\%$ (Arregui *et al*, 2006). Class C mechanical water meters are those that have been used for a period not more than 10 years. It was also found that their accuracy depended on flow rates. The study was done for the frequency range of between 40 l/h and 500 l/h which is similar to the range of frequency under which the designed meter was tested.

From these results, the designed water meter in addition to automating meter reading, is more accurate than the existing mechanical water meters over the same frequency range.

The designed system can comfortably compete with the existing water meter systems in that majority of these systems require a human being to take the readings for billing purposes. The designed low cost water consumption meter system based on GSM transmits the meter readings automatically. A comparison of mechanical meters and the designed meter is given in table 6.7.

Table 6.7 Mechanical water meters versus the designed water meter system

Capability	Mechanical water meter systems	Low cost consumption water meter based on GSM(This system)
Meter reading	It is manual	It is automated
Cost of meter reading	High since it requires an employee of water company to take readings manually	No cost since meter reading are transmitted automatically
Accuracy	Depends on meter reader. It is error prone ($2\% \pm 1\%$)	It is more accurate ($0.93\% \pm 0.7\%$)
Billing	When meter is not accessible for reading, bills are based on estimated readings	Readings are transmitted automatically hence bills are accurate

CHAPTER SEVEN

RECOMMENDATIONS AND CONCLUSIONS

7.1 Conclusion

A low cost GSM based water meter system has been assembled and implemented. The system integrates a $G_{1/2}$ flow sensor, a liquid crystal display, a PIC18F4550 microcontroller and a solenoid valve. MikroC Integrated Development Environment has been used to write, develop and debug codes for PIC18F4550. The assembled system managed to read flow rate and based on the code stored in the microcontroller's program memory, calculated the volume in litres and displayed both flow rate and volume on the LCD. The same volume reading was also transmitted to a mobile phone through GSM module. The system is effective and hence can be used in water meter industry to automate water meter reading hence saving on cost and man hours.

7.2 Recommendations for future work.

Though the designed system was able to automate water meter readings, some few developments can be made to make the system more robust. These include;

- (i). Extending the system to enhance monitoring of water usage by the utility company. This would require individual consumers be provided with the proposed system. Another system is to monitor whether the total volumes of each of the individual meter system matches the total volume supplied by the utility company would be implemented.
- (ii). Introduction of some sensors which could be used to enable the system to distinguish between water and air since there times when water flows

combined with some air bubbles. This would minimize errors in readings brought about by air under pressure

- (iii). Microcontrollers or any other controllers with greater memories can be used to enhance continuous data storage data since the designed system may not store massive data. Utility companies may require data for a certain period of time thus introduction of systems with higher memories may be a solution.

REFERENCES

- Beil, J.** (1996). *Choosing a microcontroller Electronic product design*: Newness. Oxford.
- Enggcyclopedia**, "Turbine flow meters"<http://www.enggcyclopedia.com/turbineflow-meters/> / accessed on December 2014.
- Kahabka, M.** (2012). *Pocket Guide for fundamentals and GSM Testing*: Wandel & Goltermann GmbH&Co. Eningen u. A, Germany.
- Lee et al.** (2008). Wireless digital water meter with low power consumption for automatic meter reading. *International Conference on Hybrid Information Technology* **5**: 639-645.
- Lovine, J.** (2011). *PIC Projects for non-programmers* (1st Ed) London: Newness.
- Han, T.M. and Myaing, M.O.** (2011). Design and construction of Microcontroller-based water flow control system. *International Conference on Circuit System and Simulation* **7**: 304-309.
- Hsia, S.C. Sheu, M.H. and Chang, Y.G.** (2013). Arrow Pointer Sensor Design for Low Cost Water Meter. *Proceedings of IEEE Sensors Journal* **13**: 1281-1287.
- Ofwat**, (2013). <http://www.ofwat.gov.uk/future/customers/metering> accessed on June 2014.
- Meiklejohn, D.** (2008). *Introduction to PIC Programming*. <http://www.gooligum.com>.
- Moghavvemi, S.Y. and Wong, S. K.** (2005). A reliable and economically feasible automatic meter reading system using power line distribution network. *International Journal of Engineering -Materials and Energy Research Center* **18**: 301-318.

Mulley, R (2004). *Flow of Industrial Fluids: Theory and Equations*. USA: CRC Press.

Nelson, S. (2014). A non-contact arrow pointer sensor design to reduce power consumption of low cost and high accuracy hybrid water meter. *Proceedings of IEEE Sensors Journal* **3**: 558-565.

Trivedi, P. (2013). PIC 18 Peripheral Interface Controllers. *International Journal of Innovative Research in Science, Engineering and Technology* **2**:1804-1809.

Quan, X.L. and Gang, L. (2010). Design of remote automatic meter reading system based on ZigBee and GPRS. *Proceedings of the Third International Symposium, Jiaozuo, China* **10**: 186-189.

Rob, K. and Gabor, M. (2010). *A Perspective on Radio-Frequency Exposure Associated With Residential Automatic Meter Reading Technology*. California, USA: Electrical Power Research Institute.

Mouly, M. and Pautet, M.B. (2002). *The GSM System for Mobile Communications*. Leland Scott Bloebaum, USA: Telecom Publishing.

Bates M. (2011). *PIC microcontrollers; An Introduction to Microelectronics*: Newness. London.

Mohamed F., Mounir B. and Bedda M. (2008). Microcontroller based heart rate monitor. *The International Arab Journal of Information Technology* **5** (4): 1-5.

Reinhardt, S. (2000). Integrating hardware and software concepts in a microprocessor based system lab. *IEEE Computers* **26**: 49-55.

Parker, H. (2011). *Hydraulic Cartridge Systems*: Catalog HY15-3502/US.

Rainer et al. (2013) Remote automatic water meter reading system using global system for mobile communications. Mapua Institute of Technology.

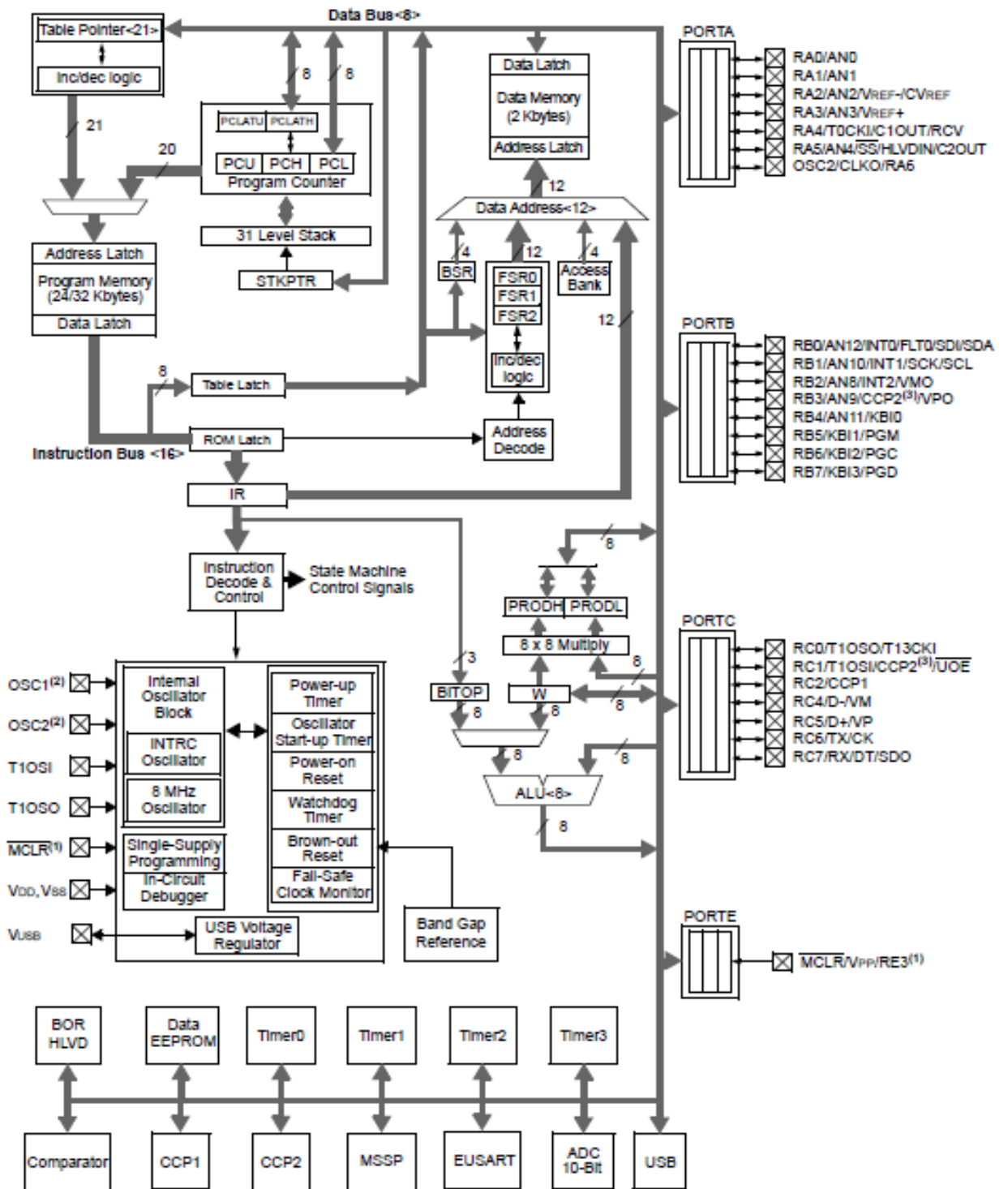
Ria, S et al. (2013). Design and development of automatic water flow meter. *International Journal of Computer Science, Engineering and Applications* **3**(3):49-59.

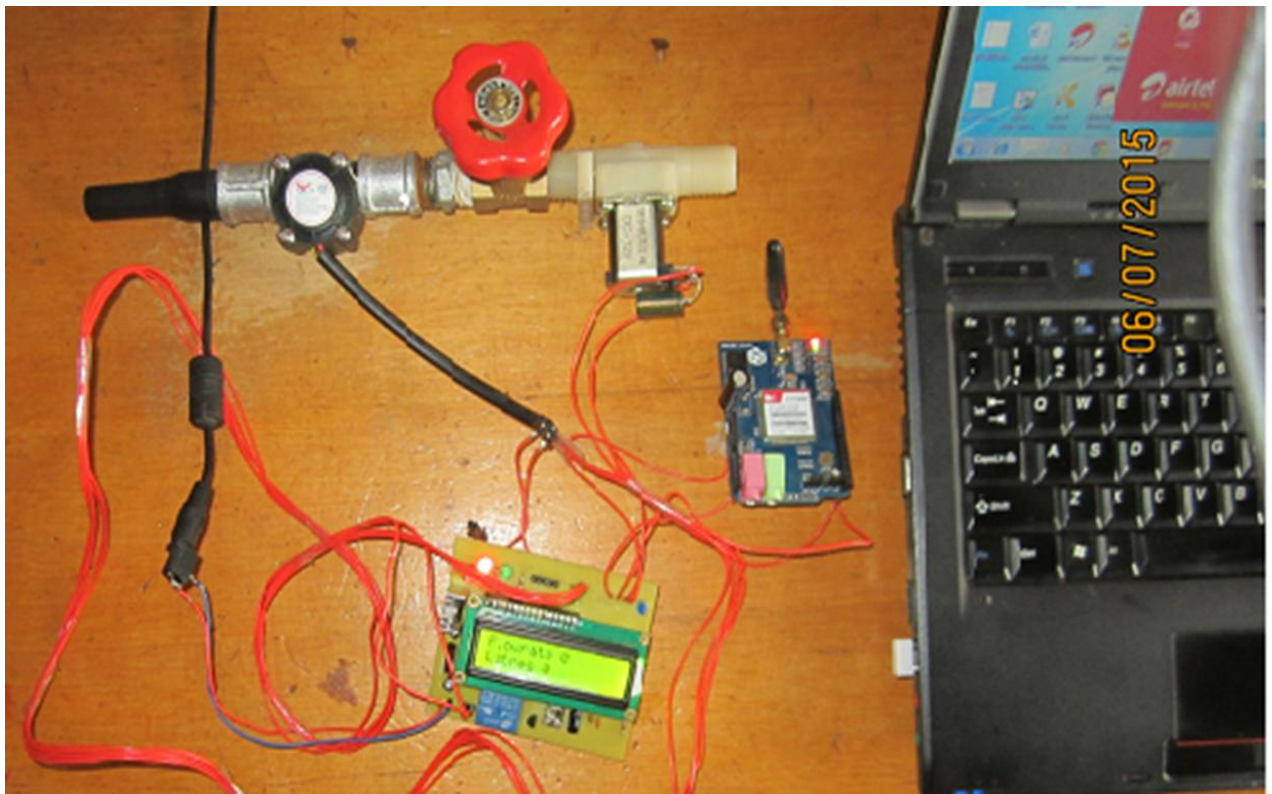
Santhosh, K.V. and Roy, B.K. (2012). An intelligent flow measurement technique using an ultrasonic flow meter with an optimized neural network. *International Journal of Control and Automation* **5**(4): 185-196.

Yogendra, P. J. and Tadwalkar, M. B. (2014). Implementation of GSM Based Water Meter a Step towards Automation in Billing System. *IOSR Journal of Electronics and Communication Engineering* **9**(4):1-4.

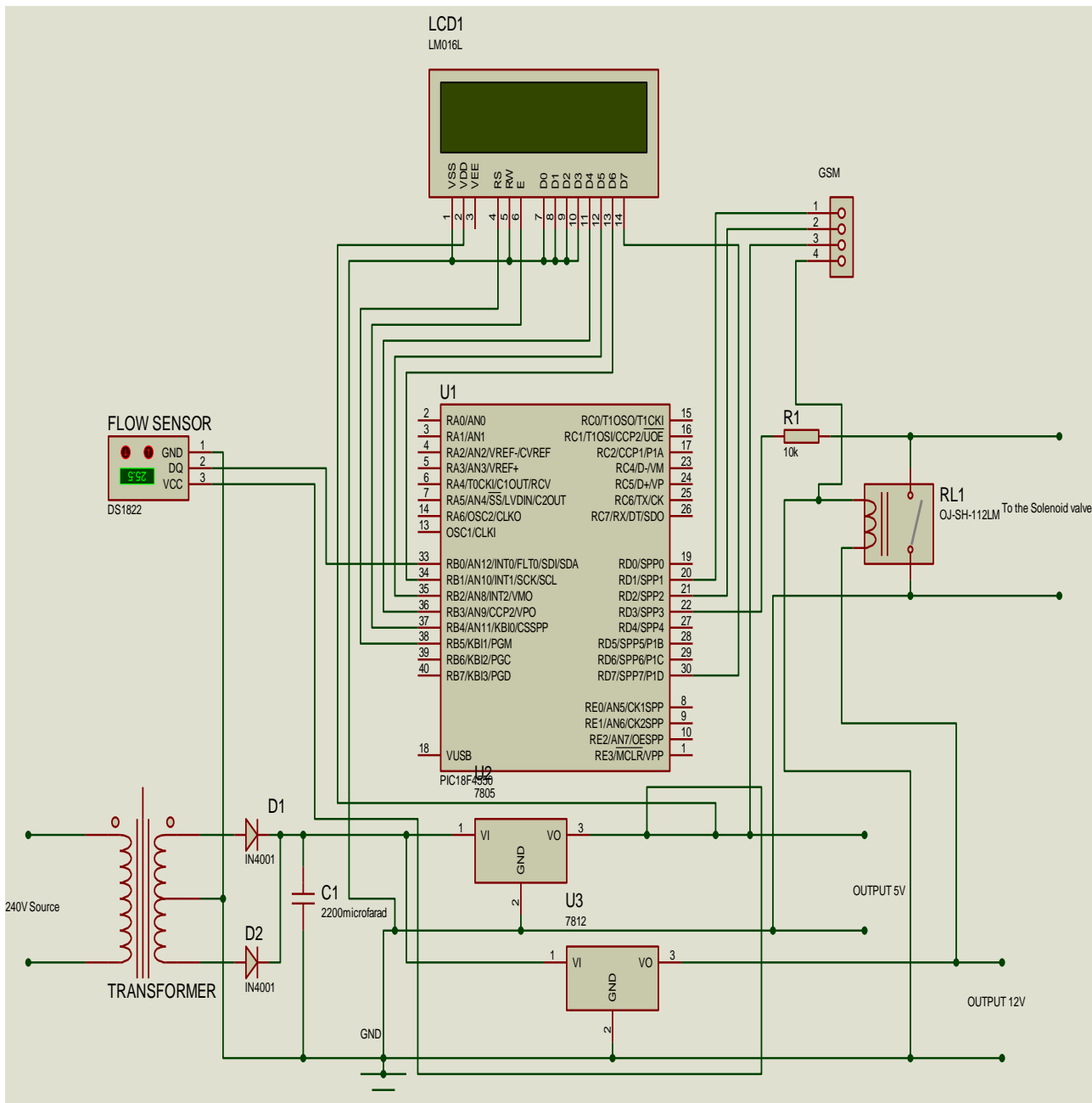
APPENDICES

Appendix A: Architecture of PIC18F4550




Appendix B: Materials employed in the research

Appendix C :Schematics of the water meter system



Appendix D: Data sheet


MICROCHIP PIC18F2455/2550/4455/4550
**28/40/44-Pin, High-Performance, Enhanced Flash,
USB Microcontrollers with nanoWatt Technology**
Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1 Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, Peripherals on
- Idle: CPU off, Peripherals on
- Sleep: CPU off, Peripherals off
- Idle mode Currents Down to 5.8 μ A Typical
- Sleep mode Currents Down to 0.1 μ A Typical
- Timer1 Oscillator: 1.1 μ A Typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A Typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High-Precision PLL for USB
- Two External Clock modes, Up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator Options allow Microcontroller and USB module to Run at Different Clock Speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns ($T_{OV}/16$)
 - Compare is 16-bit, max. resolution 83.3 ns (T_{OV})
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module Supporting 3-Wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

- C Compiler Optimized Architecture with Optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory Typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory Typical
- Flash/Data EEPROM Retention: > 40 Years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Optional Dedicated ICD/ICSP Port (44-pin, TQFP package only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		USART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I ² C™			
PIC18F2455	34K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	34K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

Appendix F: The programme code

```

/*-----*/
/* Program Name: A LOW COST WATER METER SYSTEM BASED ON GSM */
/* Name of Author: PETER MWANGI*/
/* PHYSICS DEPARTMENT */
/* KENYATTA UNIVERSITY */
/* JUNE 2015 */
/*-----*/

/* the code displays volume of water after a specific time. The flow rate and volume are
displayed on LCD and a short Message on the same is sent through GSM network to a
mobile phone*/
/*-----*/

// LCD module connections
sbit LCD_RS at LATB.B5;
sbit LCD_EN at LATB.B4;
sbit LCD_D4 at LATB.B3;
sbit LCD_D5 at LATB.B2;
sbit LCD_D6 at LATB.B1;
sbit LCD_D7 at LATD.B7;

sbit LCD_RS_Direction at TRISB5_bit;
sbit LCD_EN_Direction at TRISB4_bit;
sbit LCD_D4_Direction at TRISB3_bit;
sbit LCD_D5_Direction at TRISB2_bit;
sbit LCD_D6_Direction at TRISB1_bit;
sbit LCD_D7_Direction at TRISD7_bit;

int rx_ptr = 0;
int pn_ptr = 0;
int Save_No_flag = 0;
char rxBuffer[50]={0x00};
char Data_In;
int rx_Flag = 0;
char PhoneNumber[15]= "";

```

```

char* dat;
char *ptr;
char *ptr1;
int Minute = 0;
int delay_time = 0;
unsigned long ms;
int seconds;
int PulseCount =0;
float litres = 0;
int total_lits = 0;
char Flowrate[7];
char litres_ASCII[15];
char *res;
char *res1;
int Cut_Off_Flag = 0;
int Open_Flag = 0;
int No_Water = 0;

long prevSeconds;
float Flowrate_L_per_Sec; // flowrate in litres per second
float Flowrate_L_per_Min; // flowrate in litres perMin

void clear_gsm_buf(){
    int ni;
    for(ni=0;ni<=50;ni++){
        rxBuffer[ni] = 0x00;
    }
}
void clear_phone_buf(){
    int ii;
    for(ii=0;ii<=15;ii++){
        PhoneNumber[ii] = 0x00;
    }
}

void Send_text_array(char *k, int size ){
    int _n = 0;
    for(_n; _n < size; _n++){
        UART1_Write(*k);
        k++;
    }
}
/*
void Get_Phone_Number(){
    int kl,j,position;

```

```

position = 0;
for(kl=0; kl<40; kl++){
    if((rxBuffer[kl]==34)&&(rxBuffer[kl+1]=='+')&&(rxBuffer[kl+2]=='2')){position = kl+1;
    }
}
if(position){
    dat = &PhoneNumber[0];
    for(j=position; j<(position+13); j++){
        *dat = rxBuffer[j];
        dat++;
    }
}
}
*/
void Send_Message()
{
    UART1_Write_Text("AT+CMGS=");
    Delay_ms(200);
    UART1_Write_Text("\"); // ASCII equivalent for ( " )
    Delay_ms(200);
    //Send_text_array(Phone Number, 13);
    UART1_Write_Text ("0721717032");
    Delay_ms (200);
    UART1_Write_Text("\r\n"); // ASCII equivalent for ( " )
    Delay_ms(200);
    UART1_Write_Text (litres_ASCII);

    Delay_ms(200);
    UART1_Write(' ');
    Delay_ms(20);
    UART1_Write('L');
    Delay_ms(20);
    UART1_Write('i');
    Delay_ms(20);
    UART1_Write('t');
    Delay_ms(20);
    UART1_Write('r');
    Delay_ms(20);
    UART1_Write('e');
    Delay_ms(20);
    UART1_Write('s');
    Delay_ms(20);

    UART1_Write(0x1A); //ctrl Z
    Delay_ms(2000);

```

```

}

void timer2_init() {
    // Setting Timer2
    T2CON.TMR2ON = 1;
    T2CON.T2CKPS0 = 1;
    T2CON.T2CKPS1 = 1; // prescaler 1:16
    T2CON.TOUTPS0 = 1;
    T2CON.TOUTPS1 = 0;
    T2CON.TOUTPS2 = 0;
    T2CON.TOUTPS3 = 1; // postscaler 1:10
    TMR2 = 0; // Timer2 initial value
    PIE1.TMR2IE = 1; // Timer 2 interrupt enable
    //INTCON = 0xC0; // Set GIE, PEIE
    PR2 = 250; // PR2 = 250
}

void Send_Message_test()
{
    UART1_Write_Text("AT+CMGS=");
    UART1_Write_Text("\n"); // ASCII equivalent for ( "
    Delay_ms(900);
    UART1_Write_Text("0721717032");
    //Send_text_array(PhoneNumber,13);
    UART1_Write_Text("\r\n"); // ASCII equivalent for ( "

    Delay_ms(900);
    UART1_Write_Text("GSM Test");
    UART1_Write(0x1A); //CTRL+Z
    Delay_ms(500);
}
/*
void MakeCall()
{
    UART1_Write_Text("ATD");
    UART1_Write_Text("0721717032");
    UART1_Write_Text(";");
    UART1_Write_Text("\r\n"); // CR
}
*/

void interrupt(){
    if (PIR1.TMR2IF) {
        ms++; // Interrupt causes cnt to be incremented by 1
        PIR1.TMR2IF = 0; // reset timer 2 interrupt flag
        TMR2 = 0; // Timer TMR2 is returned its initial value
    }
}

```

```

if (ms > 1009) { // Assumption every 50 count is one 1 sec
    //ms = 0;
    //seconds ++; //
}
}

if(INTCON.INT0IF == 1)
{
    PulseCount++;
    INTCON.INT0IF = 0;
}

if(PIR1.RCIF){
    Data_In = UART1_Read();

    switch(Data_In){
    case('*'):
        rx_Flag = 1;

        break;

    case('#'):
        Cut_Off_Flag = 1;

        break;

    case('@'):
        Open_Flag = 1;
        break;

    }

    rxBuffer[rx_ptr] = Data_In;
    rx_ptr +=1;
    PIR1.RCIF = 0;

}
}

void main() {
    UART1_Init(115200);
    ADCON1= 0X0f; //Setting bits as digital

    //RCON.IPEN = 1; //Enable priority interrupts
    TRISB = 0x00; // PORTC O/P
}

```

```

TRISD7_bit = 0;
//TRISC4_bit = 0; //unused pin as output
//RC4_bit = 0;
LATC.B4 = 0;

Lcd_Init();
Lcd_Cmd(_LCD_CURSOR_OFF);

INTCON.PEIE = 1; //Enable peripheral interrupt
PIR1.RCIF = 0; // UART GSM disable receive interrupt flag
INTCON.GIE = 1; //Global interrupt enable

INTCON.INT0IE = 1; //Enable RB0/INT external Interrupt
INTCON2.INTEDG0 = 0; //Interrupt on rising edge
INTCON.INT0IF = 0;

TRISD2_bit = 0; //led pin as output
TRISD3_bit = 0; //led pin as output
TRISC7_bit = 1; //rx pin as input
TRISC.B4 = 0; //unused pin as output
TRISB0_bit = 1; // set flow sensor to input

//-----Timer 2 setting-----

T2CON.TMR2ON = 1; //turn timer on
T2CON.T2CKPS0 = 1;
T2CON.T2CKPS1 = 0; // prescaler 1:4
T2CON.TOUTPS0 = 1;
T2CON.TOUTPS1 = 0;
T2CON.TOUTPS2 = 0;
T2CON.TOUTPS3 = 1; // postscaler 1:10

TMR2 = 0; // Timer2 initial value
PIE1.TMR2IE = 1; // Timer 2 interrupt enable
PR2 = 125; // PR2 = 250

// GSM Initialisation
// Sets the GSM into SMS Mode and Deletes all previous messages
UART1_Write_Text("ATE1\r\n"); // Disables Echo Mode. ie AT commands are not returned
after sending them to the GSM Module
Delay_ms(500);
UART1_Write_Text("AT+CMGF=1\r\n"); // Setting SMS Mode
Delay_ms(500);

```

```
UART1_Write_Text("AT+CNMI=2,2\r\n"); // Configuration for receiving the whole message
instead of the message location
```

```
Delay_ms(500);
```

```
UART1_Write_Text("AT+CMGD=4\r\n"); // Deleting messages
```

```
Delay_ms(500);
```

```
PIE1.RCIE = 1; // UART GSM Receive interrupt enable
```

```
PIR1.RCIF = 0;
```

```
//RD2_bit = 0;
```

```
LATD.B2 = 0;
```

```
rx_ptr = 0;
```

```
ms = 0;
```

```
PulseCount=0;
```

```
// MakeCall();
```

```
Send_Message_test();
```

```
while(1){
```

```
Delay_ms(1000);
```

```
Lcd_Out(1,1,"Flowrate");
```

```
Lcd_Out(2,1,"Litres");
```

```
if(delay_time < 5000){
```

```
delay_time ++;
```

```
}
```

```
If (seconds > 60){
```

```
RC4_bit = 1;
```

```
LATC.B4 = 1;
```

```
Delay_ms (1000);
```

```
RD2_bit = 0;
```

```
LATD.B2 = 0;
```

```
RC4_bit = 0;
```

```
LATC.B4 = 0;
```

```
PIE1.RCIE = 0;
```

```
Lcd_out(2,1,PhoneNumber);
```

```
Lcd_Cmd(_LCD_CLEAR);
```

```
Send_Message();
```

```
rx_ptr = 0;
```

```
rx_Flag = 0;
```

```
clear_gsm_buf();
```

```
clear_phone_buf();
```

```
PIE1.RCIE = 1; //Enable interrupt
```

```
PIR1.RCIF = 0;
```

```

    seconds = 0;
}

if(Cut_Off_Flag == 1){
    PIE1.RCIE = 0;
    //RD3_bit = 1;
    LATD.B3 = 1; //Switch off Solenoid Valve
    rx_ptr = 0;
    Cut_Off_Flag = 0;
    clear_gsm_buf();
    clear_phone_buf();
    PIE1.RCIE = 1; //Enable interrupt
    PIR1.RCIF = 0;
    No_Water = 1;
}

if(Open_Flag == 1){
    PIE1.RCIE = 0;
    LATD.B3 = 0; //Switch on Solenoid Valve
    rx_ptr = 0;
    Open_Flag = 0;
    clear_gsm_buf();
    clear_phone_buf();
    PIE1.RCIE = 1; //Enable interrupt
    PIR1.RCIF = 0;
    No_Water = 0;
}

if (ms>1000){
    //RD2_bit ^= 1;
    LATD.B2 ^=1; // Blue Led
    Flowrate_L_per_Min = (int)(((ms/1000)*PulseCount) / 7.5); // Flowrate in litres per min
    ms = 0;
    PulseCount=0;
    seconds++;
}

if (delay_time > 3){
    IntToStr(Flowrate_L_per_Min,Flowrate);
    litres += ((Flowrate_L_per_Min/60));
    IntToStr(litres,litres_ASCII);
    //Lcd_Out(1,1,"          ");

    res1 = Ltrim(Flowrate);
}

```

```
Lcd_Out(1,10,res1);

res = Ltrim(litres_ASCII);
Lcd_Out (2,8,res);

if (litres>100){
  //RD3_bit = 1;
  LATD.B3 = 1; // Switch off Solenoid Valve
}
else if(No_Water == 0){
  //RD3_bit = 0;
  LATD.B3 = 0; //Switch on Solenoid Valve
}
delay time = 5000;
}
}
}
```