بسم الله الرحمن الرحيم

Sudan University of Science and Technology

College of Graduate Studies

School of ElectronicEngineering

MSC Computer and Network Engineering

Design Control and Monitoring System Of Water Distribution Networks

تصميم نظام تحكم ومراقبة في شبكات توزيع المياه

A Research Submitted in Partial fulfillment for the Requirements of M.Sc. in Electronics Engineering (Computer and Network Engineering)

Prepared by: Rania Mohamed Hassan Osman

Supervisor: Dr. Ala Eldin Awouda

July-2017

الآية

بسم الله الرحمن الرحيم

قال تعالى :

أَوَلَم يَرَ الَّذينَ كَفَروا أَنَّ السَّماواتِ وَالأَرضَ كانَتا رَتقًا فَفَتَقناهُما وَجَعَلنا مِنَ الماءِ كُلَّ شَيءٍ حَيٍّ أَفَلا يُؤمِنونَ ﴿٣٠﴾

صدق الله العظيم

(سورة الأنبياء الآية 30)

Dedication

To whom who has scarified and struggled for the sake of our well being and prosperity, dear father.

To the source of patience and optimism and hope, dear mother.

To those who has favored me than themselves, to those who taught me the meaning of life dear brothers.

To those with whom I tasted the most beautiful moments, my friends

ACKNOWLEDGMENT

First of all we thank Allah for all those blessings I have day by day.

My beloved supervisor s DR. Ala Eldin Awouda for their kindness and precious help.

I thank my lovely families for the support I have got they have been there for me in each moment saying thank you won't be enough either. I also thank my teachers. Ialso thank My Teachers in School of Electronics Engineering Sudan University of Science and Technology last of all I thank My friends those who make this journey an incredible one, each moment was irreplaceable, god bless them all.

ABSTRACT

Nowadays, water distribution network automation is becoming more and more popular day by day due to its numerous advantages, an internet based water distribution network system focuses on monitoring and controlling water distribution network instance**.** Water distribution network face wastage of water due to improper water supply management, non-monitoring in real time,this caused scarcity of water, scarcity of water consider mainly problem in cities. water distribution network automation systems face four main problems; high cost of ownership, inflexibility, poor manageability, and difficulty in achieving security. The main objectives of this study is to overcome this problems and design and implement a water distribution network monitoring and controlling system using Internet of Things (IoT) that can provides remote controlling and monitoring most of the network instance through an easy manageable web interface.The system consist of three nodes(main node, two branches node), communication between branches node and main node through xbee, all node to calculate consumption water quantity and bursts, leakage detection by using two sensors(pressure, flow) which connected to Arduino which connected to solenoid valve, this measured values send from branches nodes to main node, from main node to database then to webpage via Ethernet, by inverse from webpage can send frame to main node to controlling ON/OFF in specific solenoid valve, then Short Message Service(SMS) message special of water distribution network instance display on monitoring screen. The result of this study shows that the measured values received from all node stored in database and display on webpage ,also can easily controlling from webpage in solenoid valve, also this system is able to reduces water scarcity problems.

المستخلص

في الوقت الحاضر، أصبح التحكم الآلي في شبكات توزيع المياه يزداد شعبية يوم بعد يوم نظرا" لمزاياه العديدة. تركز نظم المراقبه الالية في شبكات توزيع المياه القائمة على الانترنت، على رصد ومراقبة حالات شبكات توزيع المياه. تواجه شبكات توزيع المياه خسائر في المياه بسبب الادارة الغير صحيحة وعدم المراقبة في الزمن الحقيقي للشبكة وهذه الاسباب تودي الي ندرة المياه , تُعد ندرة المياه مشكلة رئيسية في المدن.تواجه أنظمة التحكم الآلي للشبكات توزيع المياه أربع مشاكل رئيسية، ارتفاع تكلفة الملكية وعدم المرونة، وسوء الإدارة، وصعوبة تحقيق الأمن. الأهداف الرئيسية لهذه الدراسة التغلب على المشاكل اعلاه وتصميم نظام تحكم ومراقبة في شبكات المياه باستخدام إنترنت الأشياء (IoT) التي يمكن من التحكم عن بعد ومراقبة الشبكة من خلال واجهة ويب سهلة التحكم. منهجية الدراسه تقوم على إنترنت الأشياء لتصميم نظام مراقبة وتحكم عن بعد يتكون النظام المصمم من ثلاثة خطوط (خط رئيسي و خطين فرعيين)، تتصل الخطوط الفرعية مع الخط الرئيسي عن طريق (xbee), الخطوط لحساب كمية المياه المستهلكة ولاكتشاف التسرب والدمج باستخدامحساسين(الضغط ، التدفق) تتصل بي (Arduino)التي تتصل مع (بلف) يتم إرسال القيم المقاسة من الخطوط الفرعية الي الخط الرئيسي ,كما يتم ارسالها من الخط الرئيسي إلى قاعدة بيانات ثم إلى صفحة ويب عن طريق الانترنت، ويمكن كذلك التحكم في البلف بتشغيله وايقافه من صفحة ويب عن طريق إرسال اطارمحدد إلى الخط الرئيسي, كما يتم عرض بعض الرسائل الخاصة بي حالات شبكات توزيع المياه علي شاشة المراقبة. اظهرت نتائج هذا البحث أن القيم المقاسة من وحده القياس تم تخزينها في قاعدة البيانات ثم تم عرضها على صفحة ويب، كمااظهرت النتائج انه يمكن التحكم بسهولة من صفحة ويب في البلف وان النظام قادر للحد من مشاكل ندرة المياه.

List of Contents

List of Figures

ListOf Tables

ListOf Abbreviations

ATM: Atmosphere.

CSMA/CD: Carrier Sense Multiple Access with CollisionDetection.

DMA: District Meter Areas

DTMF: Dual Tone Multi-Frequency.

FPS: Feet per second.

FS: Full scale.

FTDI: Future Technology Devices International.

GigE: Gigabit Ethernet.

GSM: Global system for mobile communication.

GUI: Graphical user interface.

ICSP: In-Circuit Serial Programming.

IDE: Integrated development environment.

KPA: Kilopascal.

L/D: Liter per day.

MAC: Media Access Control.

mA: milliamp.

MII: Media Independent Interface.

Ml/m: milliliter per minute.

MPA: megapascal.

Ml/s: milliliter per second.

N/mm2: Newtons per millimeter squared.

Pa: Pascal.

PLC: Programmable Logic Controller.

PSI: pounds per square inch.

PSIA: pounds per square inch absolute.

RFID: Radio frequency identification.

RTUs: Remote terminal units.

SCADA: Supervisory Control and Data Acquisition.

SI: International System of Units.

SPI: Serial peripheral interface bus.

TTL: Transistor Transistor logic.

VDC: Virtual Device Contexts.

CHAPTER ONE
INTRODUCTION

1.1 Overview

An internet based water distribution network control and monitoring system focuses on controlling network instances whether you are inside or outside your control room. Automatic control gives an individual the ability to remotely or automatically control things around the network. A network appliance is a device or instrument designed to perform aspecific function, especially an network instances, such as a leakage detection.Automation is today's fact, where things are being controlled automatically, usually the basic tasks of turning on/off of solenoid valve and beyond, either remotely or in close proximity. Automation lowers the owners judgment to the lowest degree possible but does not completely eliminate it. The concept of remote management of network instances over the internet from anywhere, any time in the world today can be a reality. Assume a system where from the office desk, the user could view the status of the network and decides to take control by closes any line, opens any line detection leakage, detection bursts, decrease pressure, open new sources.

According to recent survey, water has become a big issue because of less rain fall, increase in population, many cities are facing this problem people have to suffer from this problem they don't have sufficient amount for their daily need. Due to lack of monitoring water can't be supplied properly, there is a need of continuous monitoring, water supply scheduling and proper distribution another problems are excessive consumption, overflow of tanks, leakage in pipeline. Water is a basic need of every human being everyone has to save the water many time, with lack of monitoring, overflow of these overhead tanks can occur because of this lots of water get wasted, another thing because of overflow in the pipeline with more pressure there is possibility of pipeline damage, leakage detection is one more problem all these are because of lack of monitoring, manual work, less man power, before implementing this project I have taken a survey of city and field survey to understand water supply distribution and related problems with the system, after tacking a survey I observe that all the work in manual and need a better technology to make proper distribution by focusing in problems is traditional methods our system design and develop a low cost embedded system device for real time monitoring of water distribution system in (IOT) platform . IOT is a world where billions of objects can sense, communicate and

share information, all interconnected over public or private internet protocol (IP) networks. These interconnected objects have data regularly collected analyzed and used to initiate action, providing a wealth of intelligence for planning, management and decision making.

1.2 Problem Statement

The problem of this project contain chronically decrease of water Quantity this prevent economic development and mayhem to environment & health. Large of decrease in water quantity due to high speed growth of population, not adequate of water distribution network, arising of leakage & bursts rate in water network. Lagging water developments works, lack of awareness sanitation at all level, poor planning & shortage of funding in water network, lack of clear policy used in water netwok. That is not expectations of consumption of water quantity in (Khartoum, Bahry, Om Dorman) this is cause real problem in water network & water distribution system, because the pumps is operate in same quality in all time. This leading problem like (Leak,Bursts) & detection of Leak , Bursts localization this take amount of time.

1.3 Proposed Solution

The proposed solution is to design a control and monitoring system which will be able to fairly distribute the water and monitor the network for errors.

1.4 Aim and Objectives

The main aim is to design a control system for water distribution network which will provides enhanced automated process control and also can fully process data in real time to yield the meaningful information that can be put to work to save water. To achieve this objective:-

1- To leak & bursts detection – apply one or more method best suited for the water distribution networks.
2- To propose a control and monitor circuit for water distribution network.
3- To select a mechanism to transform this massive amount of data into meaningful information and transfer it quickly and accurately throughout the utility to all function.
4- To use good policy to make the water network management is easily.
5- To create database that can provide complete reports & build query in database the user can easy to retrieval the information in any time.
6- Instance monitoring and controlling with lees delay possible.

7- To design system by used low cost , high security, flexibility, high manageability.

8- To propose system to solve wasted water by high rate.

1.5 Methodology

The overall system functionality outcome from interaction between the system components: nodes, web pages, database. data collect in project based on nodes, also project contain three nodes, two branch node exampled branches lines in water distribution network , main Node exampled main line in network. all nodes We used sensors technology to continuously measuring network weathers , branch nodes send this value by (XBEE) to the main node that are connected to sensors, actuators and database and web page via internet using Ethernet, main node consider Data Collector that send the value and values of branches nodes , this values stores in database, database contain two table , line table used to store (name , number of line) , sensor table used to store (id, line number, flow , pressure, date &time), and display on webpage, then the webpage should take action on specific actuator according to pre assigned conditions, turn specific actuators on/off or select specific position of the actuators, finally display SMS represented other network instance on webpage

1.6 Research Layout

This thesis is a written documentary that contain records such as the idea generated, concepts applied, and final activities done. It consists of five chapters. Following is a chapter-by-chapter description of information in this thesis.

In chapter one, discussion on what the project is really all about, such as the Introduction of the project, the project objectives, problem statement, proposed solution.

Chapter two looks into the literature review that has been done especially on theoretical concepts on the various method .this chapter discusses the background study.

Chapter three is regarding the project methodology, also contain the project theoretical that involves the necessary task and activities to be undertaken to complete the project such as hardware descriptions, development and software development the major bulk of the projet.

Chapter four discusses about the results from the research and discussion.

Finally, Chapter five concludes the thesis and gives some recommendation for future researchers the summary of the final project. The conclusion, suggestions or recommendatins for improvement can be implemented in future are discussed as well.

CHAPTER TWO

BACKGROUND AND LITERATURE REVIEW

2.1 Background

This section will introduce basic information about Arduino, Ethernet, Xbee Module Actuators, Sensors, Data base and Web Application.

2.2 Arduino Microcontroller

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world. This project is based on microcontroller board designs, manufactured by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on the processing project, which includes support for the C and C++programming languages. The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors (Kainka, 2013).

2.2.1 Arduino Hardware

An Arduino board historically consists of an Atmel 8-, 16- or 32-bit AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as shields. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus, so many shields can be stacked and used in parallel. Prior to 2015 official Arduino had used the Atmel mega AVR series of chips, specifically(the ATmega8, ATmega168, ATmega328, ATmega1280).
and ATmega2560 and in 2015 units by other manufacturers were added. A handful of other processors have also been used by Arduino compatible devices. Most boards

include a 5 V regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the lily pad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer. Currently, opt boot loader is the default boot loader installed on Arduino UNO (Kainka, 2013).

2.2.2 Arduino Software

Arduino programs may be written in any programming language with a compiler that produces binary machine code. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio. The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in Java. It originated from the IDE for the Processing programming language project and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism for compiling and loading programs to an Arduino board. A program written with the IDE for Arduino is called a "sketch". The Arduino IDE supports the C and C++ programming languages using special rules of code organization. The Arduino IDE supplies a software library called "Wiring" from the Wiring project, which provides many common input and output procedures (Kainka, 2013).There are a many type of Arduino boards as shown in Figure (2.1).

(a)Arduino Yun(b)Arduino Leonardo (c)Arduino UNO

(d) Arduino MEGA 2560

Figure 2. 1: Examples of Arduino Hardware

2.2.3 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. The circuit works on a frequency

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. (Devika, 2014)

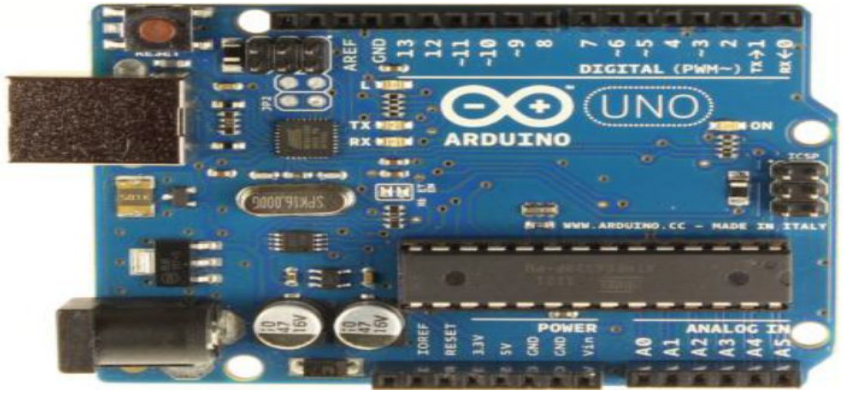


Figure 2.2: Arduino Uno

Features

Table 2.1 illustrated the features and specification of each feature:

Table 2.1 Arduino Specification

Feature	Specification
Microcontroller	Atmega328
Operating voltage	5v
Input volt (recommended)	7-12v
Input voltage (limits)	6-20v
Digital I/O pins	14(of which 6 provide PWM output)
Analog input pins	6
DC current per I/O pin	40 Ma
DC current for 3.3v pin	50 Ma
Flash memory	32kB(Atmega328) of which .5KB used by boot loader
SRAM	2KB(Atmega328)
EEPROM	1KB(Atmega328)
Clock speed	16MHZ

2.3 EthernetModule

Ethernet shield enables the Arduino to send and receive data from anywhere in the world with an internet connection.It allows you to easily connect the Arduino to the internet**.**Ethernet has been a relatively inexpensive, reasonably fast, and very popularLANtechnology for several decades. Two individuals at Xerox PARC -- Bob Metcalfe andD.R. Boggs-- developed Ethernet beginning in 1972 and specifications based on this workappeared in IEEE802.3 in 1980. Ethernet has since become the most popular and most widely deployed network technology in the world. Many of the issues involved with Ethernet are common to many network technologies, and understanding how Ethernet addressed these issues can provide a foundation that will improve your understand of networking in general. The Ethernet standard has grown to encompass new technologies as computer networking has matured. Specified in a standard, IEEE 802.3, an Ethernet LAN typically uses coaxial cable or special grades of twisted pair wires. Ethernet is also used in wireless LANs.

Ethernet uses the CSMA/CD access method to handle simultaneous demands. The most commonly installed Ethernet systems are called 10BASE-T and provide transmission speeds up to 10 Mbps. Devices are connected to the cable and compete for access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. Fast Ethernet or 100BASE-T provides transmission speeds up to 100 megabits per second and is typically used for LAN backbone systems, supporting workstations with 10BASE-T cards. Gigabit Ethernet provides an even higher level of backbone support at 1000 megabits per second (1 gigabit or 1 billion bits per second). 10-Gigabit Ethernet provides up to 10 billion bits per second.

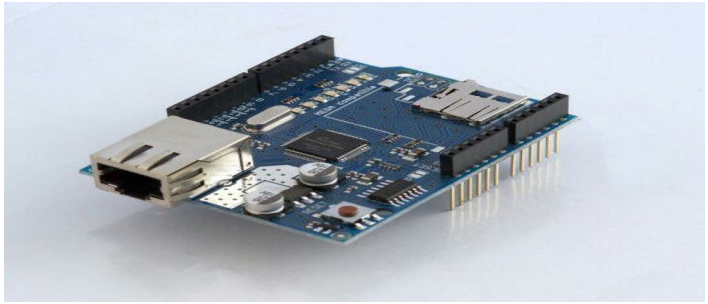


Figure 2.3: Ethernet shield

2.3.1 Ethernet Controller Features

ENC28J60-H board use ENC28J60 stand-alone Ethernet controller with these features:

- IEEE 802.3.Compatible Ethernet controller.
- Fully Compatible with 10/100/1000Base- T Networks.
- Integrated MAC and 10Base- T PHY.
- Supports one 10Base- T Port with Automatic Polarity Detection and Correction.
- Supports Full and Half-Duplex modes.
- Programmable Automatic Retransmit on Collision.
- Programmable Padding and CRC Generation.
- programmable Automatic Rejection of Erroneous packets.
- SPI interface with clock speeds up to 20 MHZ.

2.3.2 Transmission Speed of Ethernet

1-10BASE-T:

The most commonly installed Ethernet systems are called 10BASE-T and provide transmission speeds up to 10 Megabits per second (Mbps). Devices are connected to the cable and compete for access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol.

2.100BASE-T Fast Ethernet or 100BASE-T provides transmission speeds up to 100 Megabits per second(Mbps) and is typically used for LAN backbone systems, supporting workst- ations with 10BASE-T cards. The 100BASE-T standard consists of five different co- mponent specifications. These include the Media Access Control (MAC) layer, the Media Independent Interface (MII), and the three physical layers, (100 BASE-TX,100

BASET4, and 100BASE-FX)

3- Gigabit Ethernet:

Gigabit Ethernet provides an even higher level of backbone support at 1000 megabits per second (1 gigabit or 1 billion bits per second). Used mostly for backbones, the first IEEE standard (802.3z) for Gigabit Ethernet (GigE) was defined in 1997 for use over multimode optical fiber. 802.3z provides full-duplex operation from switch to end station or to another switch and half-duplex using CSMA/CD in a shared environment

4- 10Gigabit Ethernet

10 Gigabit Ethernet is an upcoming Ethernet technology that transmits at 10 Gbps 10 Gigabit Ethernet enables a familiar network technology to be used in LAN, MAN a-nd WAN architectures. However the CSMA/CD method for gaining access to the physical medium is not employed, and half duplex operation is not supported. 10 Gigabit Ethernet uses multimode optical fiber up to 300 meters an single mode fiber up to 40 kilometers.

2.4 Xbee Module

Xbee series1 modules which are based on the IEEE 802.15.4 standards. These modules allow very reliable and simple communication between microcontrollers, computers, systems, really anything with a serial port, point to point and multi-point networks are supported. For easy interfacing with xbee series1 module, Arduino board was used. The Arduino boards come with a library for interfacing with xbee series1 module and for dealing with analog or digital inputs and output. xbee received data via wired connection of RX and TX port at the Arduino board, it will transmit that data wirelessly to the GUI which is configured as a coordinator. Small size, low power, low cost, long battery life, don't need to be configured, and used for high-throughput applications requiring low latency and predictable communication timing are the reasons of using xbee series1 module. Operate within 2.4 GHz frequency band and are pin-for-pin compatible with each other (Industries, 2016)

Figure2.4: Xbee module

2.4.1 The Features of the Xbee module:

- 3.3V @ 50mA

- 250kbps Max data rate

2.5 Sensor:

Sensor or transducers is defining as a device that receives energy from one system and transmit it to another .like a physical into signal variable. Broadly define ,the sensor is a device which capable of being actuated by energizing input from or more transmission media and turn generating a related signal to one or more transition system .it provides a useable output in response to specific input measured , which may be physical or mathematical quantities ,property, or conditions. The energy transmitted by these systems may be electrical, mathematical or acoustical .the natural electrical output from the transducers depends on the basic principle involved in the design. The output may be analog, digital or frequency modulated. The term sensor should be distinguished from transducer. The latter is a converter of any one type of energy into another, whereas the former converts any type of energy into electrical energy as shown below in Figure 2.5.

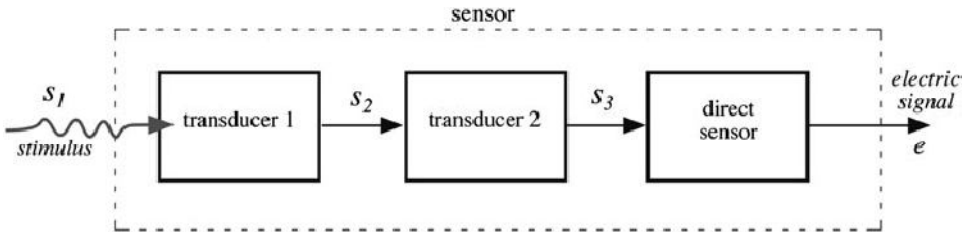


Figure 2.5: Converting measured energy to electrical signal

A sensor may incorporate several transducers. s1, s2, and so on are various types of energy, note that the last part is a direct sensor producing electrical output (e).The sensor has to be physical compatibly with it is intended application .there have eights specification that should be consider.

2.5.1 while selecting

1- operating range: chosen to maintain range requirement and good resolution

2- sensitivity: chosen to allow sufficient output

3- frequency response resonant frequency flat over desired range.

4- environment compatibility, temperature range conserve fluid, pressure, interaction size and mounting restriction.

5- minimum sensitivity: to expected errors as well as errors expected due to sensitivity to other stimulus.

6- usages and ruggedness, ruggedness both of mechanical and electrical intensities versus size and weight.
7- electrical parameters: length and type of cable required signal to noise ratio when combined with amplifiers and frequency response stimulus other then measure.
8- accuracy: repeatability and calibration Limitation(Anon 2016).

For this project pressure and flow sensor have been chosen.

2.5.2 Sensor Classification

Sensor classification schemes range from very simple to the complex.There are different ways to classify sensors

1. Passive and Active Sensor

passive sensor does not need any additional energy source and directly generates an electric signal. The active sensors require external power for their operation, which is called an excitation signal.

2. Absolute and Relative Sensor

Absolute sensor detects a stimulus in reference to an absolute physical scale that is independent on the measurement conditions. e.g. a thermostat: a temperature-sensitive resistor, a pressure sensor. Relative sensor produces a signal that relates to some special case. e.g. thermocouple is a relative sensor. It produces an electric voltage that is function of a temperature gradient across the thermocouple wires. Thus, a thermocouple output signal cannot be related to any particular temperature without referencing to a known baseline. Another way to look at a sensor is to consider all of its properties, what it measures (stimulus), what its specifications are, what physical phenomenon it is sensitive to, what conversion mechanism.

2.5.3 Flow Sensors

Flow sensors the devices that detect and measure water flowing through pipes, are becoming necessary components of efficient irrigation systems and mainly acts as a sensory organ for the brain in the irrigation controller, giving it information to make operating decisions. Flow meter basically works with the output of the flow sensor, this is Known as G1/2 water flow sensor. As the water flows the rotor, As the turbine rotates magnetic field is produced and accordingly an Ac 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 liter can be counted by the software programming. Thus pulses produce an output frequency which is directly proportional to the volumetric flow rate/total flow rate through the meter. Also

measuring flow rate through rotating rotor provides high accuracy, excellent repeatability, simple structure and low pressure loss. The sensor used for measuring the flow of the water coming from the rivers is a price type analog meter from Gurley precision instruments. This device use buckets wheel, conical buckets that rotate with the flow of water. the rotation of the bucket wheel rotates a shaft inside the device, which makes contact with the upper contact wire once a revolutions, and the lower contact wire once every 5 revolutions, completing a circuit from the binding post. Measuring along the chassis allows the user to detect when the circuit is completed, and therefore, when a rotation of the bucket wheel has been made. The price Meter is rated to read velocities ranging from(0.2 – 25 fps) feet per with 2% accuracy. With the analog meter, this is typically done with a headphones attachment. The headphones have an internal 1.5 volt battery that sends voltage down to the binding post of the meter. The second lead is attached somewhere along the chassis to then detect when the circuit is completed. An audible clicking sound is heard in the headphones, as the user counts how many clicks over a sample period(Ria Sood ,2013).

Figure 2.6: flow sensor

2.5.3.1 Selection criteria of flow rate sensor:

- Should be able to measure flow rate of transparent fluid accurately.
- Should be able to measure flow rate in the range of .05 mL/s to .5 mL/s.
- Should have an output compatible with 8051 microcontroller.

2.5.3.2 Specification of flow rate sensor :

- Model– sea YF-S201.
- Sensor type– Hall effect.

- Working voltage– 4.5 to 18 VDC.
- Maximum draw current- 5 mA at 5V.
- Output type- 5 TTL.
- Working flow rate- 1-30 mL/m.
- Working temperature rang- (-25 - +85 degree Celsius).
- Accuracy- (+-10%).
- Maximum water pressure- 2MPA.
- Durability- Minimum 300,000 cycle.
- Pulses per liter-(450).

2.5.3.3 Flow SensorsTechnology

Selecting a flow meter always impacts maintenance requirements; however, this is seldom used as a selection criterion. Maintenance contain in flow meter technology

1. Non- Invasive Flow Meters

These meters are characterized by being able to measure the process flow without coming into contact with it. The following table(2-1) summarize the information of non –invasive flow meter type.

Table 2.2: Non- Invasive Flow Meters comparison

	Transit time	Doppler	Array- based	Hybrid ultrasonic
Accuracy	±1% of reading in some cases, ±1% of full scale in most	±1% of full scale	±1% of reading	Depends on measurement method
Repeatability	±0.5% of reading	±1% of range	±0.3% of reading	"
Clean liquids	YES	NO	YES	YES
Dirty liquids, slurries, gas bubbles	NO	YES	YES	YES
Minimum flow rate	No minimum flow rate	0.03 ft/s	3 ft/s	Depends on measurement method
Maximum flow rate	40 ft/s	18 ft/s	No maximum flow rate	"
Bidirectional flow	YES	NO	NO	"

2.Semi- Invasive Flow Meter

These meters come in contact with the process flow through a small orifice drilled into the pipe, which does not usually exceed 2 inches in diameter. The following table (2-2)summarize the information of semi – invasive flow meter type.(MAPLA, 2013).

Table 2.3: Semi - Invasive Flow Meters comparison

	Wetted transit time	Insertion magnetic
Accuracy	±1% of reading	±2% of reading
Repeatability	±0.5% of reading	±0.5% of reading
Clean liquids	YES	YES
Dirty liquids, slurries, gas Bubbles	NO	YES
Minimum flow rate	1.5 ft/s	3.3 ft/s
Maximum flow rate	60 ft/s	16 ft/s
Bidirectional flow	YES	NO

2.5.4 pressure sensors

a pressure sensor is a device which senses pressure and converts it into an analog electrical signal whose magnitude depend upon the pressure applied . since they convert pressure into an electrical signal, they are also termed as pressure transducers , a pressure sensor measures the pressure, typically of gases or liquids. Pressure is an expression of the force required to stop a gas or fluid from expanding , and is usually stated in term of force per unit area. A pressure sensor generates a signal related to the pressure imposed . typically, of gases or liquids. Pressure is an expression of the force required to stop a gas or fluid from expanding, and is usually stated in term of force per unit area. A pressure sensor generates a signal is related to the pressure imposed. Typically , such a signal is electrical , but it might also include additional means, such as optic signals, visual signals and/or auditory signals. Pressure is the force per unit area exerted by a fluid or gas. The recognized International System of Units (SI) for pressure measurement is the Pascal (Pa); however, pounds per square inch (psi), inches of water (in-H2O), Newtons per millimeter squared (N/mm2) and Bar are also common. The most critical mechanical component in any pressure transducer is generally the pressure sensing structure (spring element). The pressure of the fluid or gas is a force on the pressure sensing structure. The function of the structure is to serve as the reaction for this applied force; and, in doing so, to focus the effect of the force into an isolated uniform strain field where strain gages can be placed for pressure measurement.

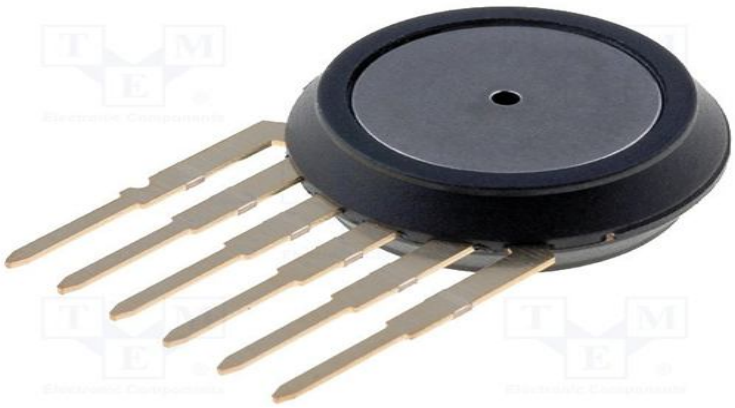


Figure 2.7: pressure sensor

2.5.4.1 pressure sensors Classification

pressure sensor classification contain three type gauge pressure sensor, differential pressure sensor , absolute pressure sensor.

1. Gauge Pressure Sensors

This sensor is used in different application because it can be calibrated to measure the pressure relative to a given atmospheric pressure at a given location. An example of gauge pressure whould be a tire pressure gauge. When the tire pressure gauge read 0 (PSI) , there is really (14.7 PSI atmospheric pressure) in the tire.

2. Differential Pressure Sensors

This sensor measures the difference between two or more pressure intrdused as input to the sensing unit. For example, measuring the pressure drop across an oil filter, Differential Pressure is also used to measure flow or level in pressurized.

3. Absolute Pressure Sensors

This sensor measures the pressure relative to perfect vacuum pressure (0 PSI or now pressure). Atmospheric pressure is about (100 KBA 14.7 PSI) at sea level. Atmospheric pressure is an on Absolute Pressure.An absolute or non-vented pressure sensor measures all pressure forces detected by the strain gauge, including atmospheric pressure (PATM). The unit of measure is PSIA (pounds per square inch absolute), measured with respect to zero pressure. The back of an absolute pressure sensor is sealed from the atmosphere. Therefore, the front of the absolute pressure sensor responds to both atmosphere pressure and the pressure head of water above the sensor. This is show by the following equation and drawing.(Jie Yang,2013).

PMEAS = PW + PATM

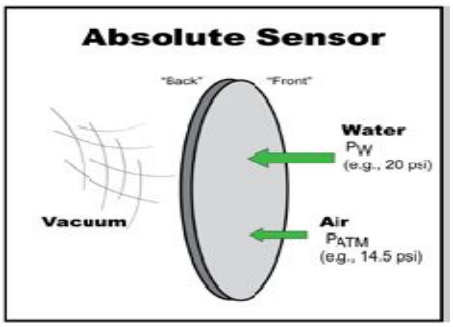


Figure2.8: Absolute pressure sensor

2.5.4.2 Accuracy Specifications

- Level TROLL and TROLL 9500 Instruments have an accuracy specification of ±0.1% full scale (FS) from -5° to 50° C for both gauged and absolute sensors.
- Aqua TROLL 100/200 Instruments have an accuracy specification of ±0.1% FS from 0° to 50° C for both gauged and absolute sensors.
- Full scale (FS) is defined as the PSI range for the sensor installed on the transducer.

In-Situ Inc.'s Level TROLL and Aqua TROLL Instruments have an accuracy specification of ±0.05% at 15° C because this is the average temperature of groundwater and creates stability for the pressure readings.

2.5.4.3 MPX4115

The MPX4115 series is designed to sense absolute air pressure in an altimeter or barometer applications. Motorola's BAP sensor integrates on–chip, bipolar op amp circuitry and thin film resistor networks to provide a high level analog output signal and temperature compensation. The small form factor and high reliability of on–chip integration makes the Motorola BAP sensor a logical and economical choice for application designers (Alldatasheet.com, 2016)

The formula of Atmosphere Pressure:

Atmosphere Pressure = ((Vout/Vin)+0.095)/0.009

2.5.4.4 Methods of measurement

A number of quite different principles are utilised in pressure measuring instruments. Some of these are fundamental in character such as measuring the height of a liquid column of known density. One such example listed below:

1. **Direct resonant pressure sensors**
2. **Piezo-electric devices**

Certain crystal materials when subjected to stress via external pressure develop a voltage across their surfaces. This piezo-electric effect can be used to measure the pressure although this voltage decays quite rapidly and some means of capture by use of a high impedance charge amplifier is needed. This is a self generating sensor requiring no external power supply. The response is very fast, making these sensors suited to dynamic pressure/peak pressure measurement. They are not suited to the measurement of steady pressure values. Quartz is the main material employed, although certain ceramics also exhibit the piezo-electric effect. The major use of this type of sensor is in the measurement of very high frequency pressure variations such

as in measuring pressures in combustion chambers of engines. They are also capable of withstanding high over-pressures.(as opposed to the longitudinal piezo-electric effect). In the transverse case, a load in the y direction results in a charge across the x direction.

2.5.4.5 pressure sensing technologies

While there are various types of pressure sensing technologies, two will be discussed Piezoresistive-Type Pressure and Foil-Based Pressure listedbelow

1. **Foil-Based Pressure.**
2. **Piezoresistive-Type Pressure**

In piezoresistive-type pressure sensors, the transduction elements which convert the stress from the diaphragm deflection into an electrical signal are called piezoresistors. Piezoresistance equals changing electrical resistance due to mechanical stress. The pressure sensing element is a diaphragm which is made from silicon. This silicon diaphragm is attached to a glass substructure (i.e., that acts as a constraint/mounting structure for the silicon). This silicon diaphragm structure performs in a predictable and repeatable manner as the pressure is applied (i.e., a very slight deflection in the structure). This pressure is translated into a signal voltage by the resistance change of the strain gages which are doped (i.e., implanted) onto the silicon diaphragm surface, then organized in an electrical circuit. The silicon diaphragm, with the exposeddoped Wheatstone Bridge, in test and measurement pressure sensors, is isolated from the pressure media being measured (i.e., media isolated pressure sensors). This is achieved by creating a cavity between the media being measured and the silicon diaphragm, then filling it with oil that does not attack the silicon or electrical circuit.**2.6 Solenoid Valve**

Is an electromechanically operated valve. The valve is controlled by an electric current through a solenoid: in the case of a two-port valve the flow is switched on or off; in the case of a three-port valve, the outflow is switched between the two outlet ports. Multiple solenoid valves can be placed together on a manifold. Solenoid valves are the most frequently used control elements in fluidics. Their tasks are to shut off, release, dose, distribute or mix fluids. They are found in many application areas. Solenoids offer fast and safe switching, high reliability, long service life, good medium compatibility of the materials used, low control power and compact design. Besides the plunger-type actuator which is used most frequently, pivoted-armature actuators and rocker actuators are also used.A solenoid valve is an electro

magnetically controlled mechanical valve used for the ON/OFF (open/closed) or diverting control of liquid or gas

2.6.1 Types of Solenoid Valves

according to their application of solenoid valve, the following types solenoid valve are available:

2-Way Valves

Two-way valves have one inlet and one outlet pipe connection. They are used to allow or shut off fluid flow, and are available in either: Normally Closed – closed when de-energized and open when energized. Normally Open – open when de-energized and closed when energized

3-Way Valves

Three-way valves have three pipe connections and two orifices (when one is open, the other is closed, and vice versa). They are commonly used to alternately apply pressure to and exhaust pressure from the diaphragm operator of a control valve, single -acting cylinder, or rotary actuator.

2.6.2 Solenoid Valves work principle

A solenoid valve, otherwise known as an electrically-operated valve is an automatic valve which serves the purpose of removing the need for an engineer to operate a valve manually. Solenoids operate using an electromagnetic solenoid coil to change the state of a valve from open to closed, or vice-versa. If the solenoid valve is 'normally closed', when the coil is energized, the valve gets lifted open by the electromagnetic force produced by the coil.

Figure 2.9: Solenoid valve

2.7Related Works

Water distribution network control and monitoring system can be described as introduction of technology within the network environment to provide convenience, comfort, and energy efficiency to its occupants. Adding intelligence to network environment can provide increased quality of life for the elderly and disabled people who might otherwise require caregivers or institutional care. There has been a significant increase in water network control in recent years due to higher affordability and advancement in Smart phones and tablets which allows vast connectivity. With the introduction of the Internet of Things, the research and implementation of water network automation are getting more popular. Much of the research attention has been given in academia. Various wireless technologies that can support some form of remote data transfer, sensing and control such as Bluetooth, Wi-Fi, RFID, and cellular networks have been utilized to embed various levels of intelligence in the water network.

In their study, they presented system using ARM-7, Firstly we are initializing all of them I/O port of microcontroller used for interfacing the device .after that we initialize the LCD and GSM modem. it show activated system with initial condition of (flow rate =0 & water =0) on the LCD screen. the microcontroller will check the output of flow sensor.it will check whether the flow rate increases above 500 pulse count by flow sensor it will off the solenoid valve for 5 sec and will display the message "HIGH FLOW RATE" on the LCD followed by sending a message through GSM to concerned authority. Again it will check for the same condition till the flow rate is not found to be less than predefind value. after that it will check whether the water limit or not. And when water limit reached it will off the solenoid valve for 5 sec and will send the message through the GSM and also display a message "LIMIT REACHED" on LCD (here we have used a limit of 1liter) (N.B.Bhawarkar2016).

In their study, they presentedA feed for each parameter is created on Adafruit. First it checks turbidity off water here mapping has been done for turbidity if turbidity of water is less than five motor in ground tank will start automatically otherwise motor will remain off. As motor get started it will fill water in both overhead tanks according to its level of water in tank water is supplied. this valves operate

automatically flow sensors gives flow rate in ml/sec. If we want to cut supply of any line we can control it from adafruit by making relay ON/OFF so, controlling is possible from a remote location. If there is no water in any line GSM will trigger a message also if there is excessive consumption in any line it will trigger a message that abnormality in line. On adafruit server we can see previous record also data continuously pushed on cloud so that we can monitor and control it in real time. 16*2 LCD is used to observe data locally connected to raspberry pi. All the sensors are connected to arduino. It takes data from all the sensor. Relays and LCD are connected to raspberry pi connector. Solenoid valves and motors operated through relay. GSM module has USB through which it is connected to raspberry pi. Arduino is connected raspberry pi through micro USB. Raspberry pi takes data and continuously push it on cloud(Pranita Vijaykumar Kulkarni 2016).

In their study, they presentedThe proposed scale aims to simulate a DMA-based smart water network and consists of three individual layers, Supply (upper layer): The water network backbone which simulates a reservoir, a pumping station, and three neighbouring DMA inlets that are monitored by pressure and flow sensors, and controlled by analog valves, DMAs (middle layer): DMA water volume state or DMA tank water levels with varying area sizes, Demand (lower layer): DMA water demand level which has one valve per DMA to mimic the variation of demand in time. After the demand layer, the water is collected in a large tank and is recycled back to the upper-layer reservoir through a underwater pump. This closed loop feature supports longterm experiments without the need for water refilling. DMA system based any area contain multiple Districts represented by one DMA, used device named logger , this device connected to pipe ,because registered all information specified leak and burst in real time (Sokratis Kartakis 2015).

This paper using PLC and SCADA. It consist of the following module such as PLC controller ,SCADA software ,GSM module , DTMF module ,Sensors such as Flow sensor and water level sensor ,actuators such as Solenoid valve and Pump. The term supervisory station refers to the servers and software responsible for communicating with the field equipment (RTUs, PLCs, SENSORS etc.), station represent an apartment which is consisting of 8 indvidual houses with 2 overhead tanks and one sump. The water which is present in the sump is moved to the overhead tank by the

Pump, if the water in the overhead tank is reduced below the predefined level. Hence the PLC controller monitors the status of the water level sensor to operate the pump. In this section we are setting a limit for per day usage depending on the number of members in each house, Even then if they need more water using DTMF we are giving an option for requesting more water if,The water entering into the houses are measured through Flow sensors in order to monitor the water that is consumed.we are implementing GSM through which each customer will be given a sms alert when they exceed their monthly limit(Harish K M 2016).

In their study, they presentedsystem use layout of distribution system like grid iron distribution system .this system contain central control room , Atmega328, tank, two Solenoid valve ,switch, GSM.When the switch of the central control room is activated, microcontroller triggers the relay circuit and it acts as a switch for solenoid valve. The solenoid valve gets open and water starts flowing from valve 1,2 to thecustomer tank1&2 for a pre-determined time interval and after the calculated time interval is reached water automatically stops and a message is delivered to the authority that ―both tanks are full‖. If any one of the customer in the apartment requires access amount of water other than regular interval allotted to the whole apartment then it can be delivered using the selector switch which is set for that particular house/room and hence after the same procedure carried above, when the calculated time interval is reached water automatically stops and message is delivered to the authority that ―customer tank 1/2 ...is full‖ (AdityaRaj Hemrajan 2016)

2.7.1 Summary

All previous studies mentioned above talked aboutDesign of Water Networks Control and Monitoring system but everyone had proposed different mechanism and tools, we proposed a system gets benefits from all above system and reduced all possible drawbacks, to obtain our objective that we mentioned above.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

Mainly the system is an embedded system designed for remotely monitoring and controlling water distributionnetwork, the system consists ofthree nodes, two branches nodes exampled branches lines in water distributionnetwork , main Node exampled main line in network, main node consistsof two sensors nodes (pressure,flow),Arduino, solenoid valve,Ethernet and XBEE moduleasshown in Figure(3.1), any branch nodeconsists of two sensors nodes (pressure, flow), Arduino, solenoid valve and XBEEmoduleas shown in Figure (3.2). in our system we have three scenario between branch node,main Node in water distributionnetwork and webpage. First scenario in thebranch node the Arduino board reads the sensor's values , valve instance and send all this data by xbee to main node, also in main node the Arduino board reads the sensor's values , valve instance, and sending this data and data sender by xbee to server by Ethernet,in same time stores this data into data base and display the readings onwebpagethen the web should take action on specific actuator according to the data come from and pre assigned conditions_ as we will see in system steps _ finallydisplay SMS represented other network instances on webpage.

3.2System Block Digram

The block diagram of main node Arduino based pressure, flow, Arduino, solenoid valve ,Ethernet and xbee module as shown in Figure 3.1.

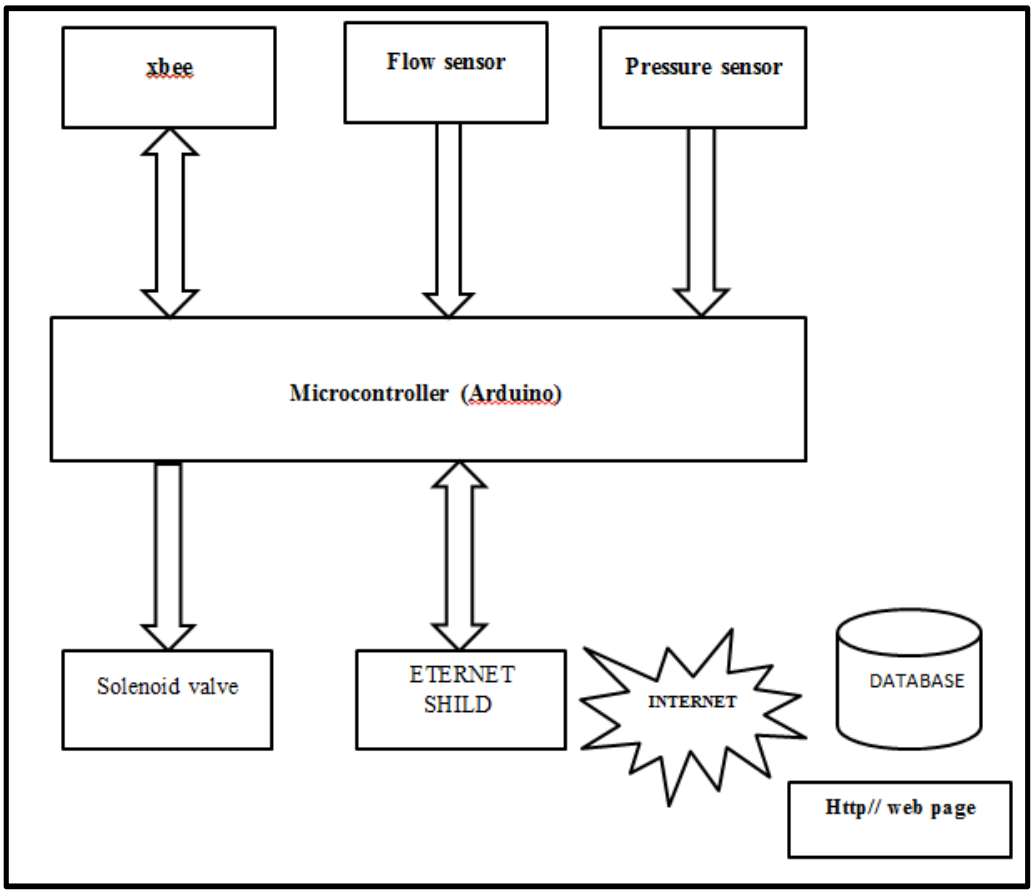


Figure 3.1: Main Node Block Diagram

The block diagram of branch node Arduino based pressure ,flow ,Arduino, solenoid valve , and xbee module as shown in Figure(3.2).

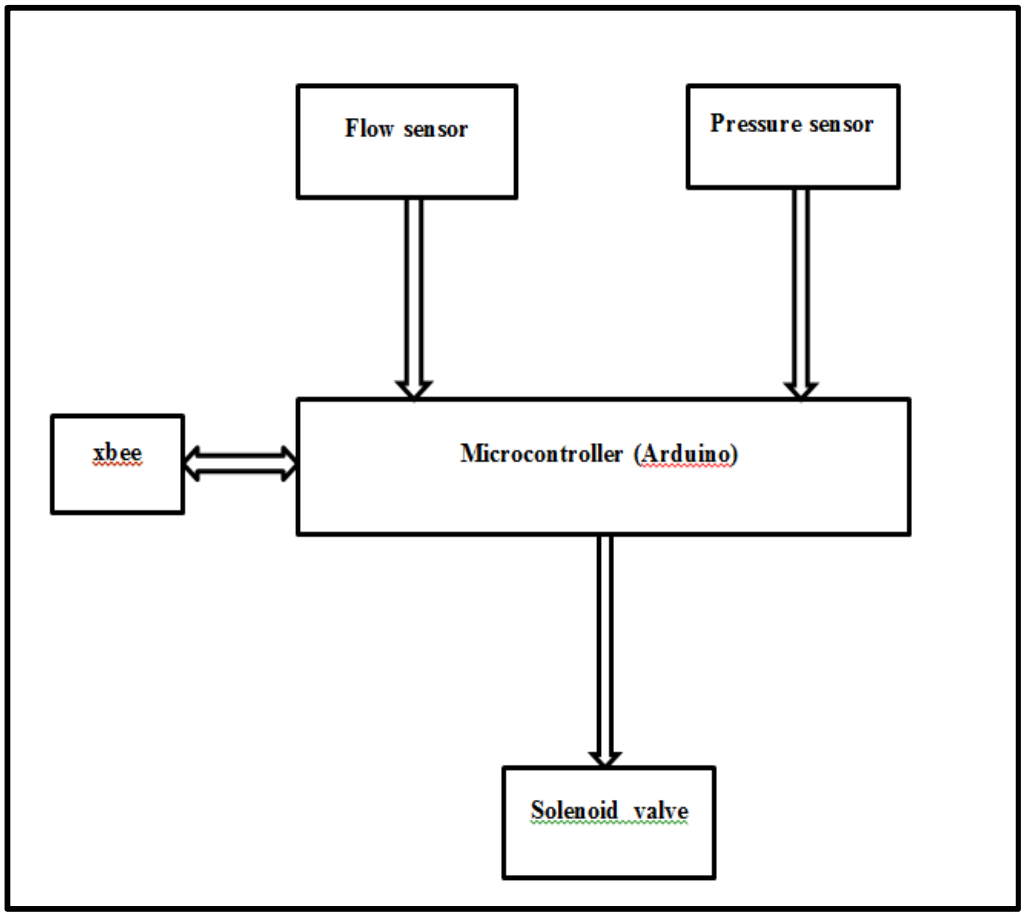


Figure 3.2: Branch Node Block Diagram

3.3 System Design Steps:

The system have two branch node, main node for any node one they have scenario based different step and format each one will illustrate in details :

3.3.1 Branch Node:

Automatic scenario:

The system followed many steps, these steps are listed below and the flow chart is shown in Figure (3.6):

1-All Xbee s1 modules scan their channel to connected with each other.

2-Arduino send to all sensors to start measuring**.**

3-The sensors start measuring**.**

i. Pressure (in kba).

ii. Flow (in L/D).

4-Arduino take the valve instance.

5-The sensors send the measured value to the Arduino board.

6-The Arduino board reads the sensors values and valve instance.

7-The Arduino board sends the frame through the xbee connection to main node .

Figure (3.3) illustrate the sent frame from the Arduino board to the main node.

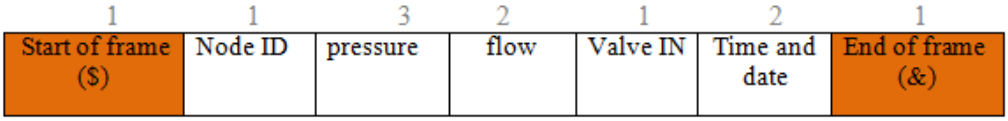


Figure 3.2: Frame Structure from branch node to main node(14bits)

3.3.2 Main Node:

Automatic scenario:

The system followed many steps, these steps are listed below and the flow chart is shown in Figure (3.7,3.8):

1- Arduino send to all sensors to start measuring**.**

2- The sensors start measuring**.**

i. Pressure (in kba).

ii. Flow (in L/D).

3- Arduino take the valve instance

4- The sensors send the measured value to the Arduino board.

5- The Arduino board reads the sensors values and valve instance

6- The Arduino read xbee

7- The Arduino board sends the frame through the Ethernet connection to server .

Figure (3.4) illustrate the sent frame from the main node to the serve

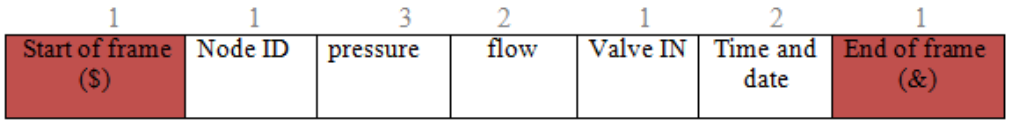


Figure 3.4: Frame Structure from main node to server(14 bits)

3.3.3 Server (Web Site):

The system followed many steps, these steps are listed below and the flow chart is shown in Figure (3.7,3.8):

Database

Create database named wncam this database consists of two table, line table consists of (line id, line name, number of house, number of person), data read table consists of (data read id, line id, pressure, flow, valve instance, time and date).

Web Site

Design Four page, Main page, inputting page named line this consists of (line name, number of house, number of person), View and control page named sensor this consists of (line id, pressure, flow, valve instance, time and date),report page.

The web page indicates to main node to send data,through frame, this frame send to all node. intended node response and send data,the main node indicates to branch node to send data,branch node send data to main node. Main node send data read by Arduino board and data sender by branch node to web page. Web page reads the sensors values and valve instance then if some conditions happens it takes specific actions listed below, Arduino receive frame from webpage and set the control to the actuators depend on received frame and other frame send by webpage latter until it is received other frame declared stop manual mode . Figure (3.5) illustrate the sent frame from web site to the main node.

Binary digits if 1 this means turn ON if 0 then turns OFF

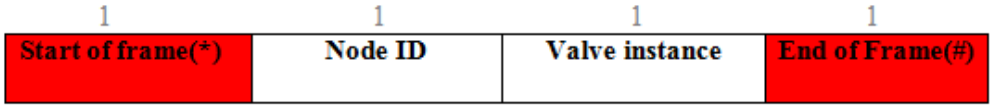


Figure 3.5: Frame Structure from server to the main node

First: Consumptions

1. Optimal Consumptions to any line is calculated by equation
   q = (number of house * number of person * 115* 1.45) % 84600
2. Save the Optimal Consumptions on web
3. Receive sender frame from main node in variable
4. Save frame in database
5. CalculatedConsumptions to any line by equation Q = V * A
6. If Q < q then display a message decrease the pump by 170 on web page.

Second: Water limit

If value of pressure sensor < 16 & value of flow sensor = 0 & valve instance = 1 of any line then display a message limit reached to this line on web page

Third: Leakage

In this case compare between flow rate of main node and flow rate of branches nodes(flow rate of main node equal total flow rate of branches nodes and keep it below flow rate in main node =X,then flow rate of any branch node = (X/2) when flow rate of any branch nodedecreesthan(X/2), the web page automatically send frame to controller to make solenoid valve offof this branch node.

Fourth: No Consumptions

If value of pressure sensor > 16 & value of flow sensor = 0 & valve instance =1 of any line then display a message decrease the pump by 270 on web page

Fifth: Stablest Instance

If value of pressure sensor in main node equal total value of pressure sensor in branch node then display a messagestablestinstance on web page.

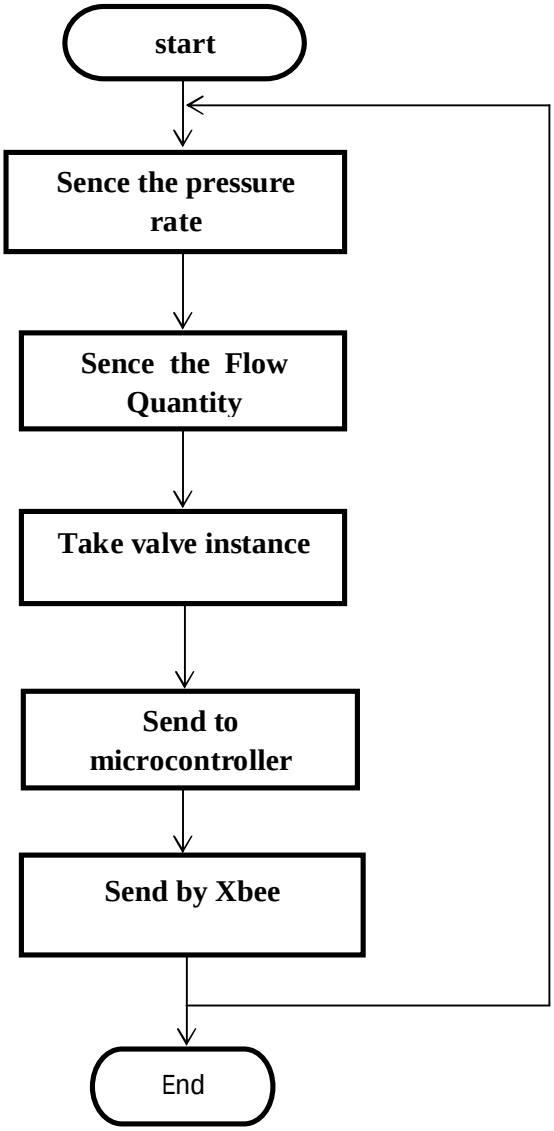
Sixth: burst

If pressure sensor of main node >50 , pressure sensor of branch node = 0 & value of flow sensor > 0) of any line when (flow sensor and pressure sensor) arrive this value the controller automatically send frame to web page, and web page send frame to main node turnSoleind valve OFF of this line

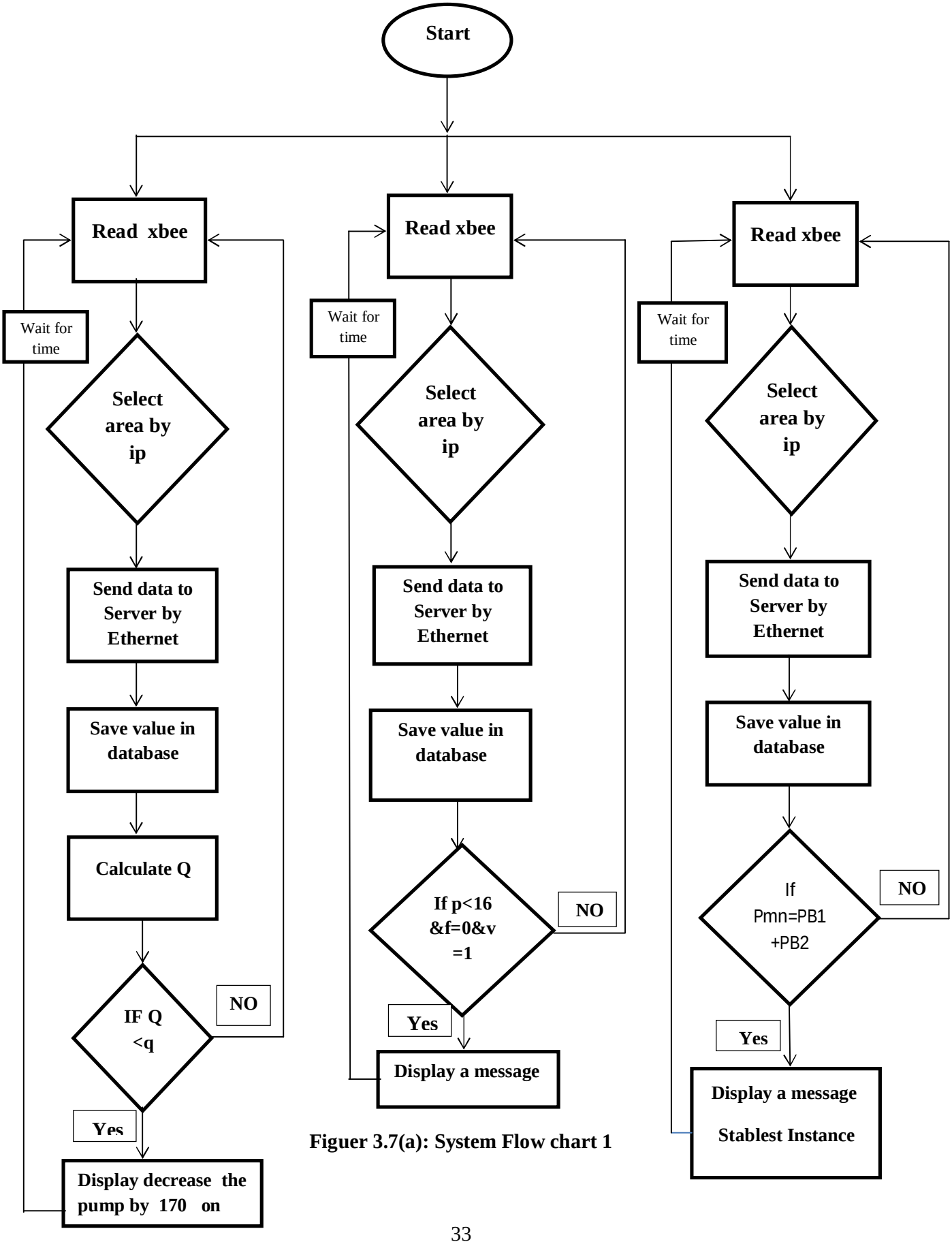
Also this system can ability to took report to all the network as real time.

3.4Flow Chart of Project

Figure 3.6 and Figure 3.7 show the flowchart of the system that illustrate the steps of the system execution:



Figuer 3.6: Branch Node Flow Chart



Figuer 3.7(a): System Flow chart 1

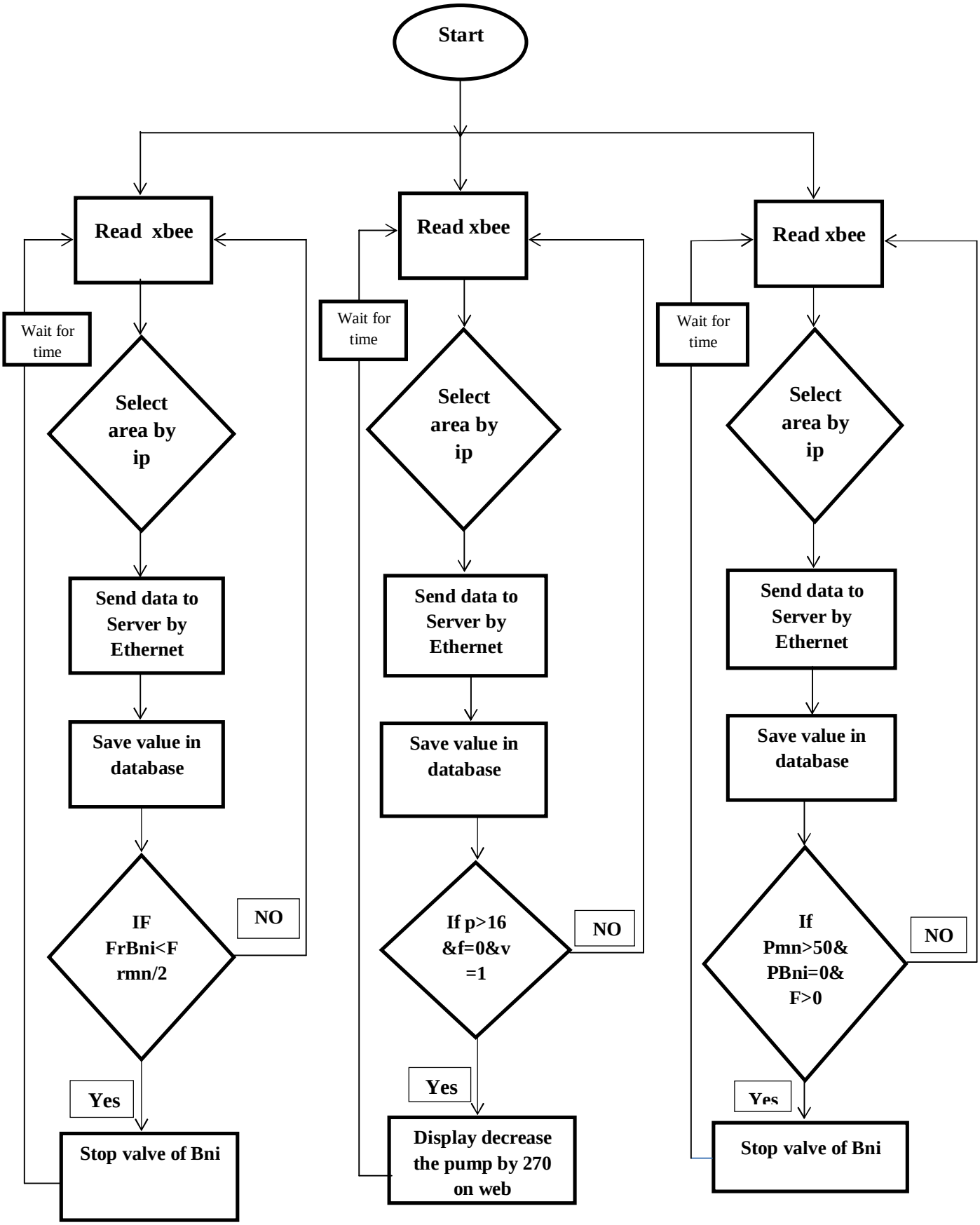


Figure 3.7(b): System Flow chart 2

3.5 The System Implementation

The system was implemented by designing circuit for each branch node and main node which is controlled andaccessed by web site. The collected data from the branches nodes stored in main node , the collected data from the main node stored in the database tables and displayed in the monitoring screen.

3.5.1 Simulation System Components:

The circuit of the system designed using proteus7 software. Figure (3.9) illustrate the main components that needed in main node, Figure (3.10) illustrate the main components that needed in branch node in this system. The description of that components is illustrated in Table (3.1).

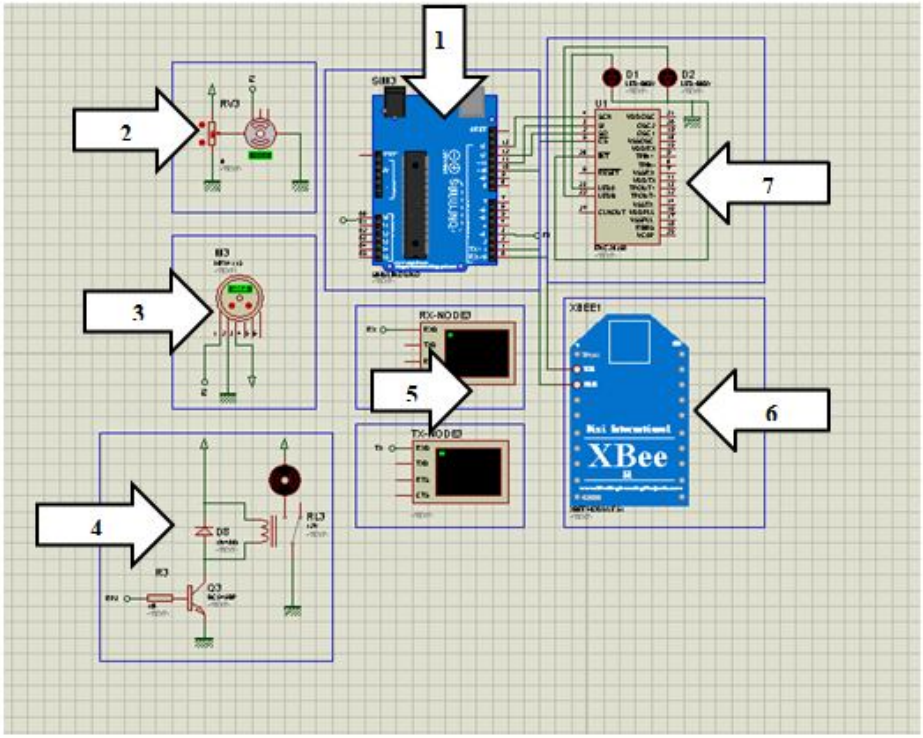


Figure 3.8: Main NodeHardware Circuit

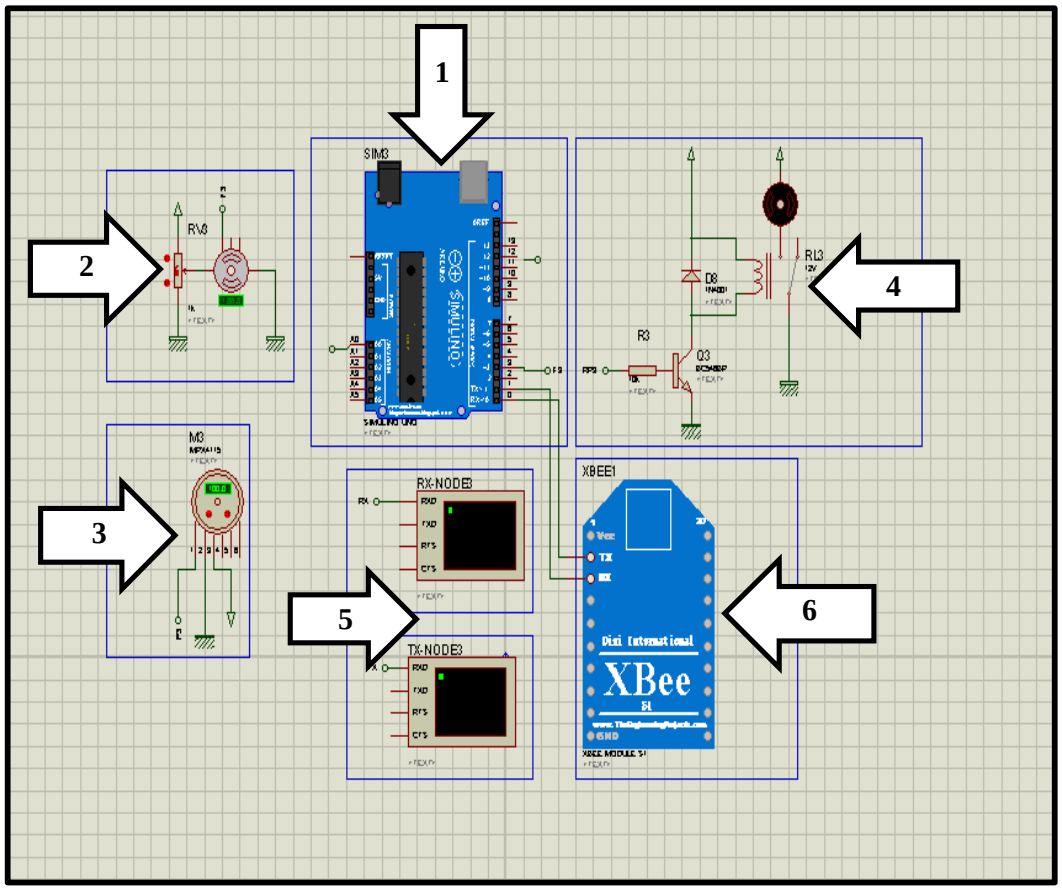


Figure 3.9: Branch Node Hardware Circuit

Table3.1: System Components

No	Component	Label	Value	Objective
1	Arduino UNO	-	-	Receive the frame from the central station, read the sensors values, and send the frame to the central station hosted by PC
2	Flow Sensor	YF-S	201	To measuring the flow rate in pipelines
3	Pressure Sensor	MPX	4115	To measuring the pressure in pipelines
4	Solenoid Valve	-	-	Used for the ON/OFF or diverting control of liquid
5	Virtual terminal	-	-	To display what is send and receive data from and to Arduino and Web Site
6	Xbee	Xbee-s1	-	To send the frame between the branch node and main node
7	Ethernet module	-	ECN260	To enable connecting to internet, to enable connecting between main node and web page

3.5.2 The Hardware Requirements

The main components used to design this systemare listed below:

Arduino, xbee module, Ethernet, pressure sensor, flow sensor, solenoid valve

3.5.3 Software Requirements

Proteus ISIS Professional

This software is being used to simulate the whole project and test the code and principle of the work before applying them actually using hardware components. Fig (3.11).

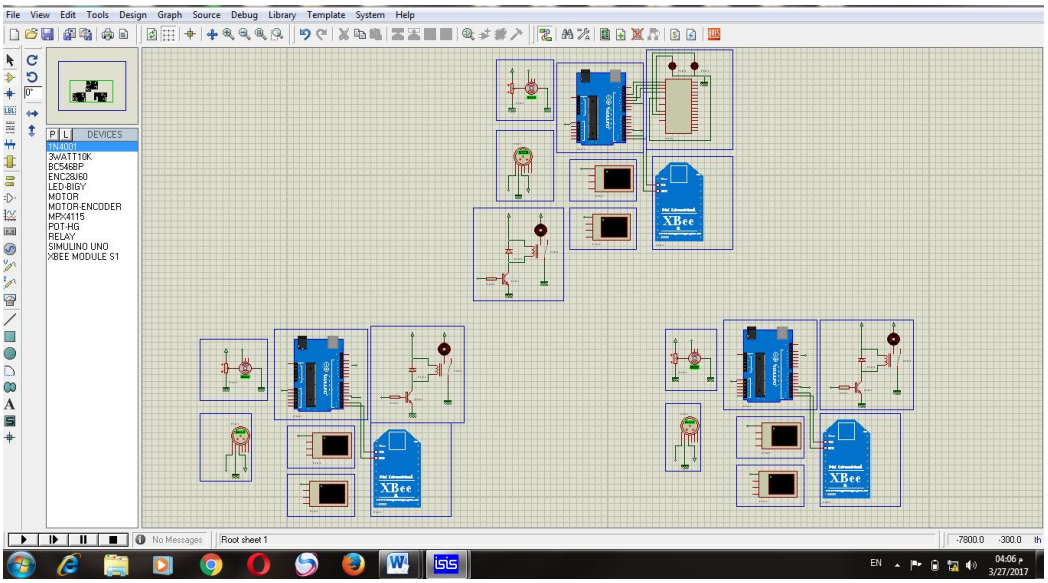


Figure 3.10: Proteus Software

Arduino

The Arduino application installation is not a typical "Run the installer and follow the prompts…" process. You simply download the ZIP package, unzip it to any drive/directory you choose, and run the "arduino.exe" application in the folder. A shortcut to the application on your desktop or taskbar is handy since there is no install process to make one.

The other potential installation requirement is drivers for the microcontroller programming hardware, which is usually some kind of USB to serial translation IC or emulation, commonly known as a VCP, or "Virtual COM Port".

Typically, official Arduino boards, and a lot of compatible designs, have used the common FTDI USB-serial driver IC (as do some Cal-Eng boards). There are other custom devices used on the Arduino UNO, Leonardo, Mega256 and others.

The drivers for all the official Arduino boards are located in the "\arduino-1.0\drivers\" install directory. If and when you are prompted for a driver installafter plugging in an Arduino-compatible board, point the driver installer to this directory.

Typically, Mac and Linux do not need a driver install, but if they do, the process is similar.

A couple of Cal-Eng boards use unique VCP driver IC's, and may need separate drivers, but that process will be explained in the individual product page.

If you've unzipped Arduino and loaded the VCP drivers, lets do a quick test. Launch Arduino and look for your board (or compatible equivalent). This example shows the common Arduino "Duemilanove" being selected:

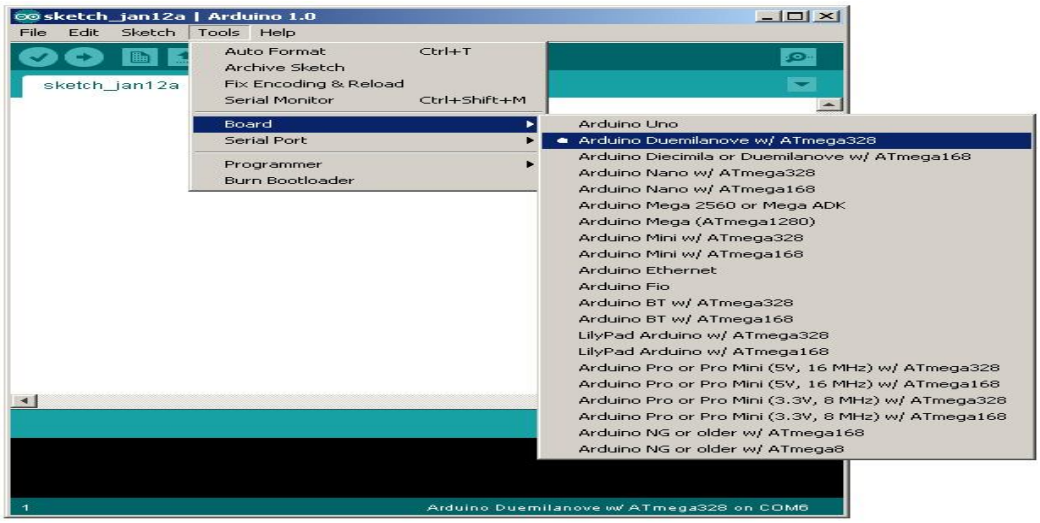


Figure 3.11: Examples of Arduino (Kainka, 2013)

Next, we need to pick which COM port your USBVCP is listed as. The actual COM port number (COM1, etc) will vary with each system, and with most modern PC's and laptops there will probably only be one. The easiest way to tell the COM port number is to remove the USB device, check the list, plug it back in and re-check to see what was added. Here's an example of a COM port list:

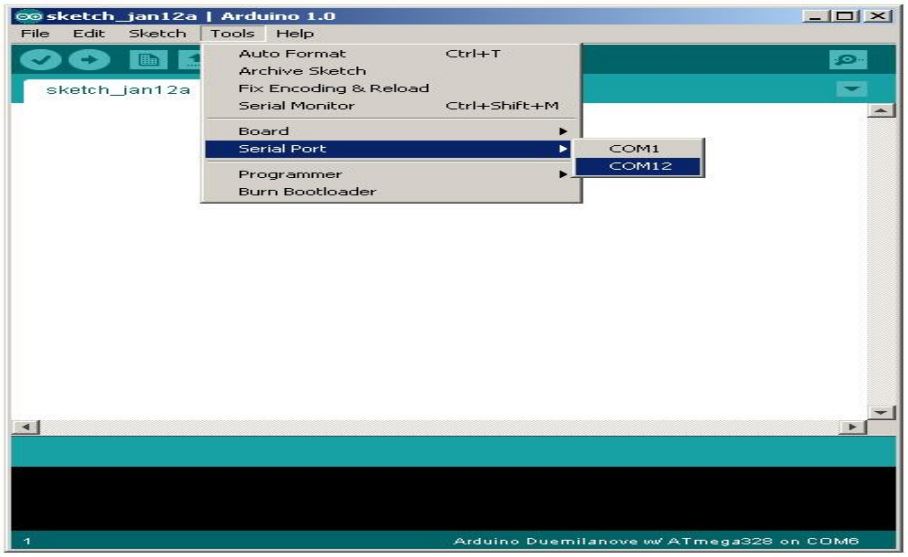


Figure 3.12: Examples of COM port of Arduino (Kainka, 2013)

Now we can load the classic “Blink” program, which is the absolute simplest test to see if your board is working. This program does nothing but blink the D13 LED once a second (forever).

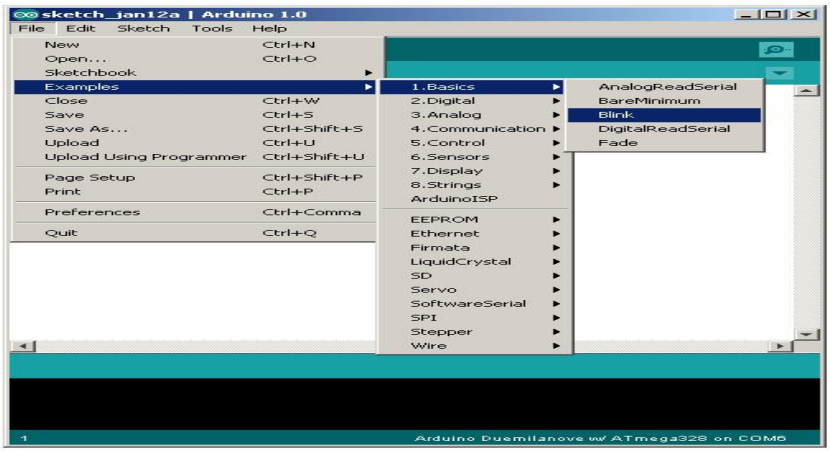


Figure 3.13: Blink program of Arduino (Kainka, 2013)

This is the moment of truth. Double-checking that your board is properly connected, press the second button (with the right arrow) labeled “Upload”. You should see some red text and other messages in and above black window at the bottom, and after a few seconds your board should be happily blinking away.

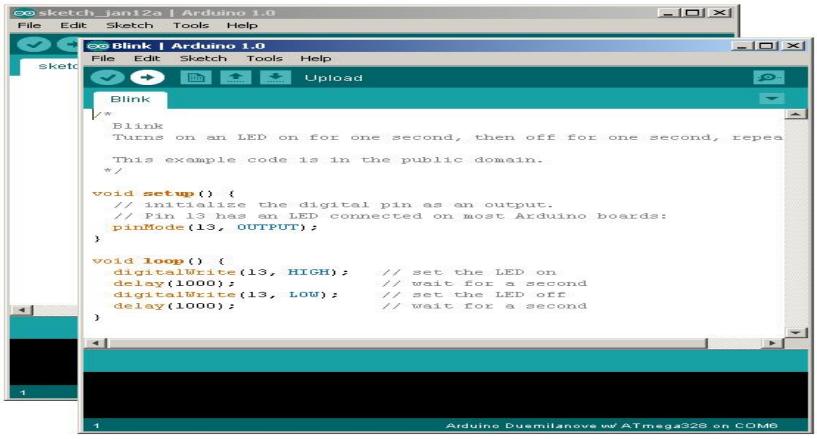


Figure 3.14: Arduino IDE.

AP Server

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images .In this research the database named WNCAM was created using MySQL, it consists of a two table, first table named line table contains 4 columns, first column is line id, second column is

line name, third column is number of house , fourth column is number of person, second table named data read contains 7 columns,first column is data read id,second column is line id, third column is pressure store the value of pressure sensor,fourth column is flow store the value of flow sensor, fifth column is valve stores the valve instance, sixth column is date stores the time and date as shown in Figure (3.26).

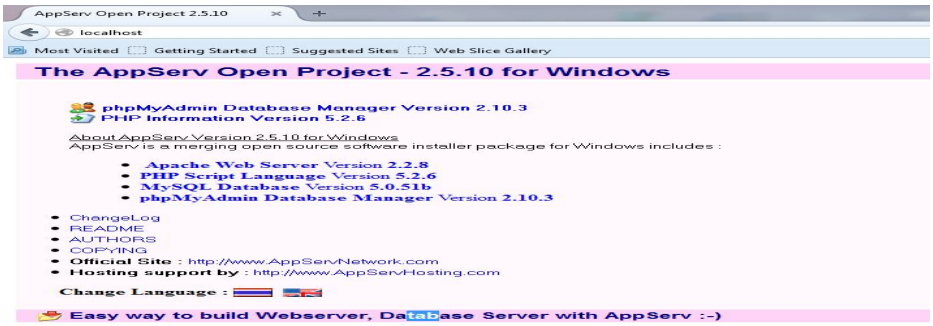


Figure 3.15: AP server program

Adobe Dreamweaver

Adobe Dreamweaver is a proprietary web development tool developed by Adobe system. Dreamweaver was created by Macromedia in 1997 and was maintained by them until Macromedia was acquired by Adobe systems in 2005.Adobe Dreamweaver is available for windows. Following Adobe's acquisition of the Macromedia product suite, releases of Dreamweaver subsequent to version 8.0 have been more complaint with W3C standards. Recent versions have improved support for Web technologies such as CSS, Java script and various server-side scripting language and frameworks including ASP, ColdFusion, Scriptlet and PHP.

Figure 3.16: Dreamweaver program

HTML

is short for HyperText Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.HTML code ensures the proper formatting of text and images so that your Internetbrowser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elementsor load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheetsare overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction:

This chapter presents the simulation results implemented using proteus7 software and Dreamweaver 8 Software and MySQL.

4.2 Result

All Xbee s1 modules scan their channel to connected with each other. The web site send frame contain of (ID, Valve instance) of each node to the Main node. The Main node sends the frame contain of (ID,Valve instance) of each node to the Xbee s1 module connected to the branch node, the coordinator sends broadcast to all the nodes wirelessly because all nodes operates in the same channel. All nodes receive the frame, and compare between ID Node in received frame and its own ID, only the node has the same ID start measuring the values (All sensors (pressure, flow) continuously measuring values)) . This node become response by sending the frame as missioned in chapter 3 in Figure 3.3 to the xbee s1 module connect to the specific's node and then sends the frame wirelessly to the coordinator, the main node receive the frame from the coordinator, then the main node get IP address and gateway by Ethernet. After that, the main node sends all frames to the server and storing in the database, finally display the readings on the monitoring screen.

4.2.1 Execute System Steps:

Sensors measurement and Automatic Control in factories :

First:**Consumptions:**

The system should continuously measure the flow value

1.Optimal Consumptions to any line is calculated by equation

qj = (number of house * number of person * 115* 1.45) % 86400

2.Save the Optimal Consumptions on web site

7. Receive sender frame from main node in variable

4.Save frame in database

5.CalculatedConsumptions to any line by equation Qi = Vi * A

6.If Qi < qj then display a message decrease the pump by 170 on web page.

In our system simulation in Proteus software Consumptions value calculation as show in figure (4.1).

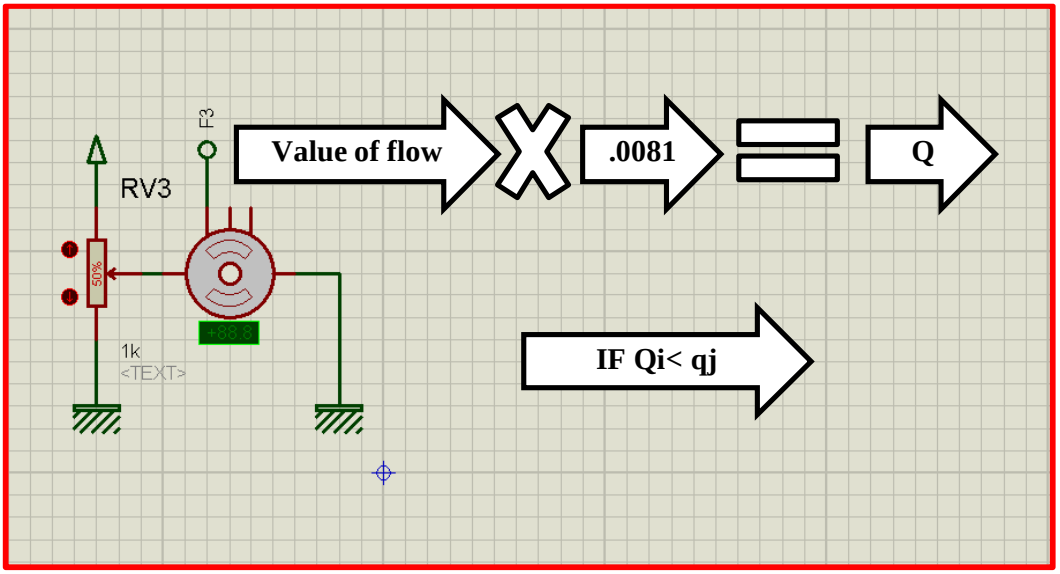


Figure (4.1): simulation of Consumptions value calculation

IF Q1< q1 display a message line1 decrease the pump by 170 on webpageas show infigure (4.2).

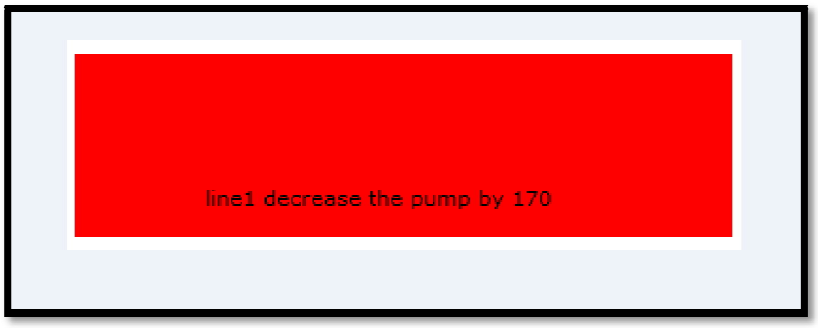


Figure (4.2): webpage display line1 amessage decrease the pump by 170

IF Q2< q2 display a message line2 decrease the pump by 170 on webpageas show infigure (4.3).

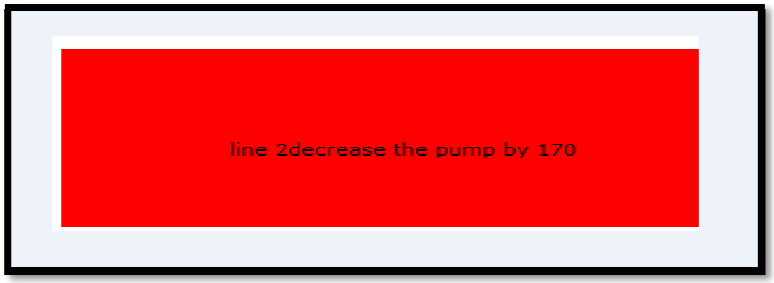


Figure (4.3): webpage display a message line2 decrease the pump by 170

Also store flow value of two branch node in database and directly present in webpage as show in figure (4.4).

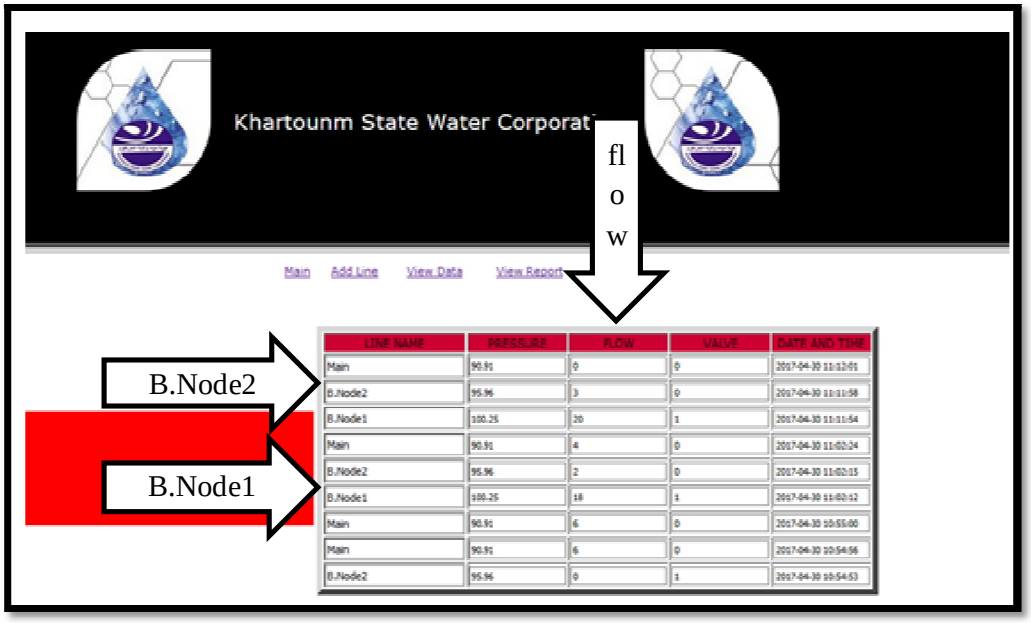


Figure (4.4): webpage present flow value of two branches nodes

Second **: Water limit:**

The system should continuously measure the (flow sensor value and pressure sensor value) also take the valve instance and keep it below(pressure sensor < 16 & value of flow sensor = 0 & valve instance = 1) of any line when (flow sensor and pressure sensor) arrive this value and valve instance take this value the controller automatically send frame to web page, and web page display a message limit reached to this line on webpage. In our system simulation in Proteus software(pressure sensor < 16 value of flow sensor = 0&valve instance = 1) as show in figure (4.5)

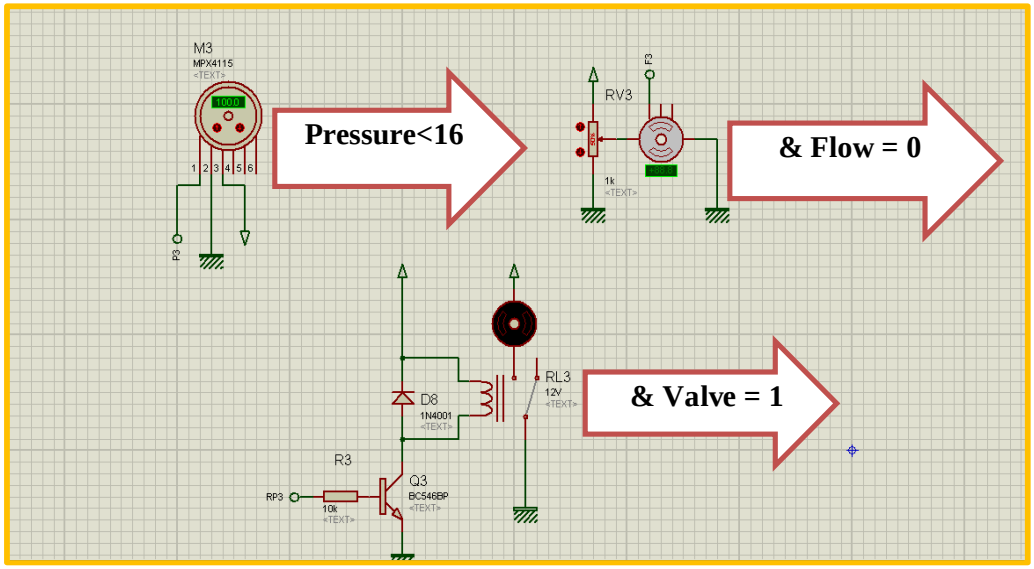


Figure (4.5): simulation of Water limit

display a message limit reached to this line onwebpage .

if (Pressure1<16 , Flow1 =0, Valve1 =1) then display a message limit reached on line1on webpage as show in figure (4.6).

Figure (4.6): webpage display a message limit reached on line1

if (Pressure2<16 , Flow2 =0, Valve2 =1) then display a message limit reached on line2 on webpage as show in figure (4.7).

Figure (4.7): webpage display a message limit reached on line2

Also store(pressure,flow) value& valve instance of two branch node in database and directly present in webpage as show in figure (4.8).

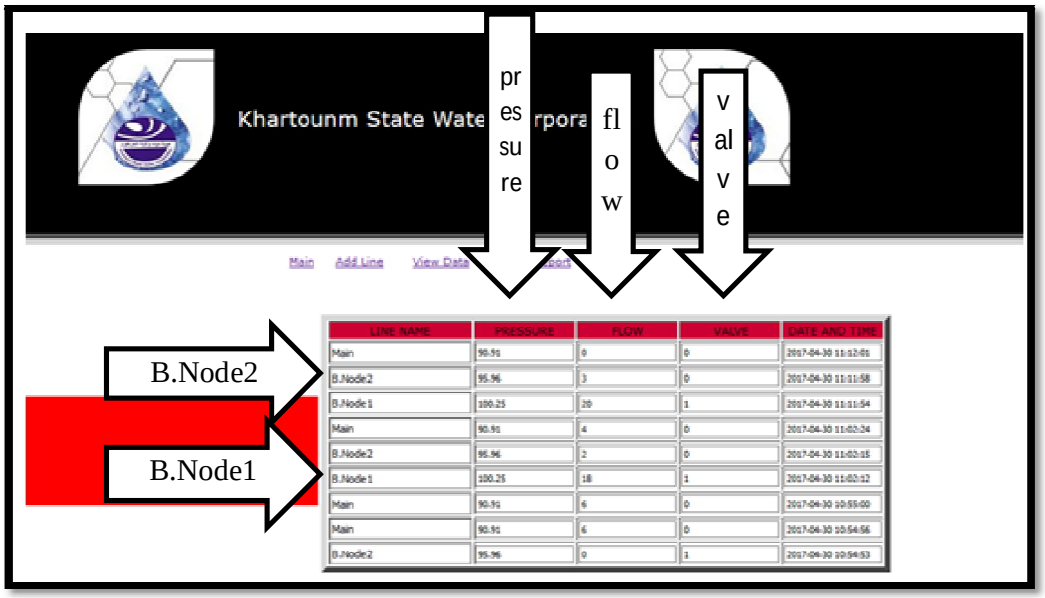


Figure (4.8): webpage present(pressure, flow) value& valve instance of B.Nodes

Third**: Leakage:**

Case1:The system should continuously measure the flow value and counted flow rate of each flow sensor and compare between flow rate of main node and flow rate of branches nodes(flow rate of main node equal total flow rate of branches nodes and keep it below flow rate in main node =X,then flow rate of any branch node = (X/2) when flow rate of any branch nodedecreesthan(X/2), the web page automatically send frame to controller to make solenoid valve off**.**

In our system simulation in Proteus software (flow rate of main node equal total flow rate of branches nodes) as show infigure (4.9).

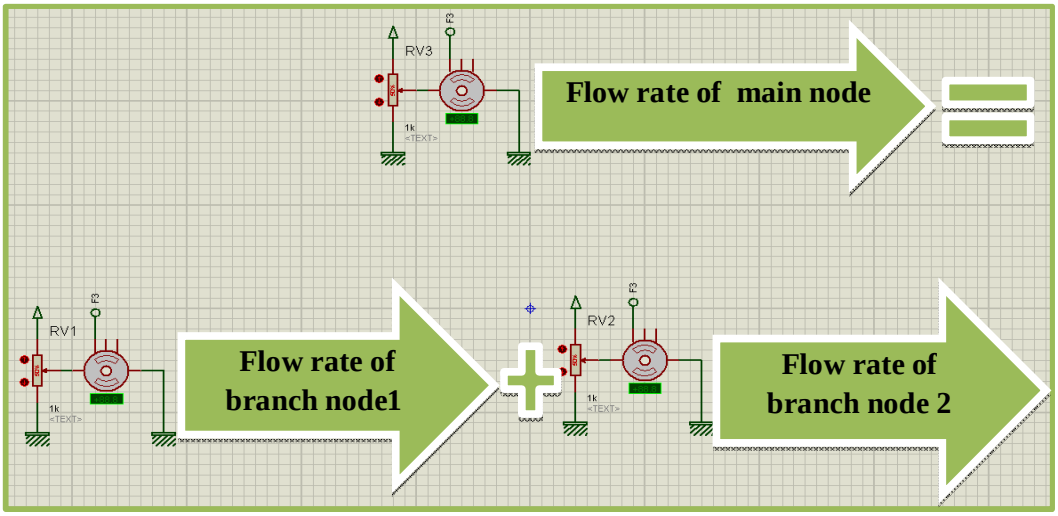


Figure (4.9): simulation of leakage case1

Case2: IF Flow rate of branch node1 (< X/2), then turn solenoid valve 1 off.

In our system simulation in Proteus software flow rate of branch node 1 decrease than (X/2) as show in figure (4.10).

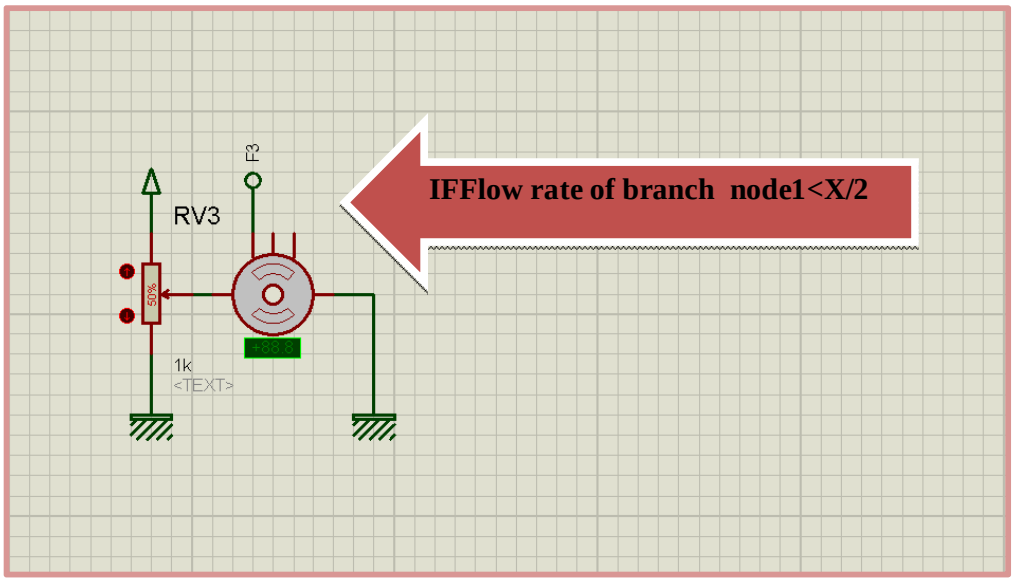


Figure (4.10): simulation Flow rate of branch node1

Turn Solenoid Valve 1 OFF automatically as show in figure (4.11).

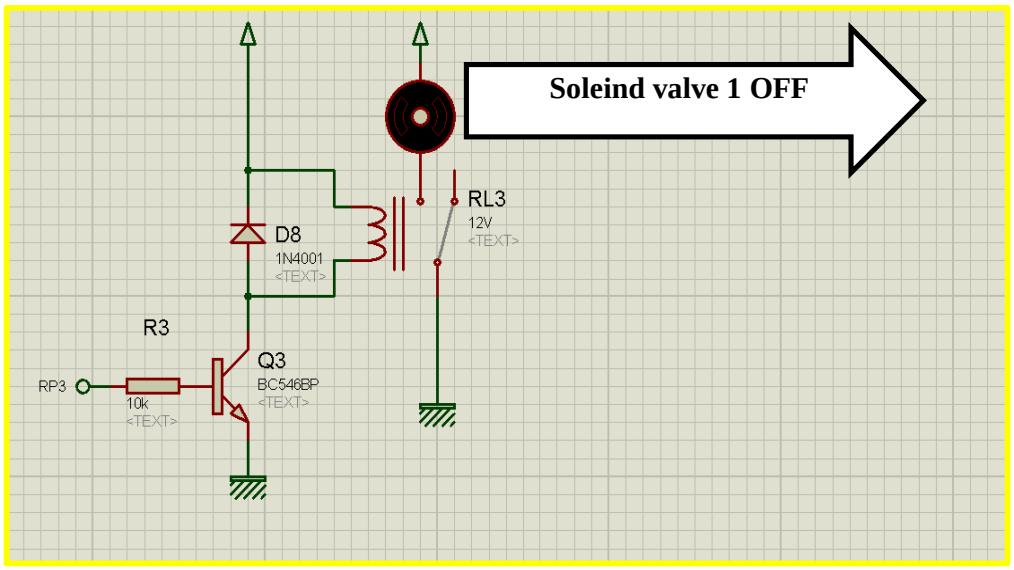


Figure (4.11): simulation of Soleind Valve 1 OFF

Case3: IF Flow rate of branch node2 (< X/2), then turn solenoid valve 2 off.

In our system simulation in Proteus software flow rate of branch node 2 decrease than (X/2) as show in figure (4.12).

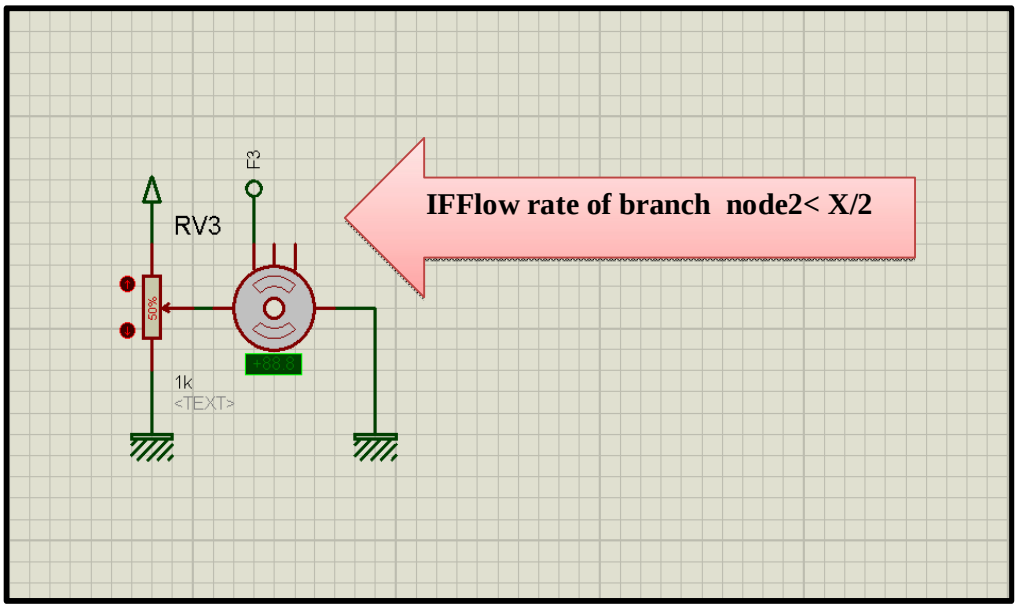


Figure (4.12): simulation Flow rate of branch node2

Turn Solenoid Valve 2OFF automatically as show in figure (4.13).

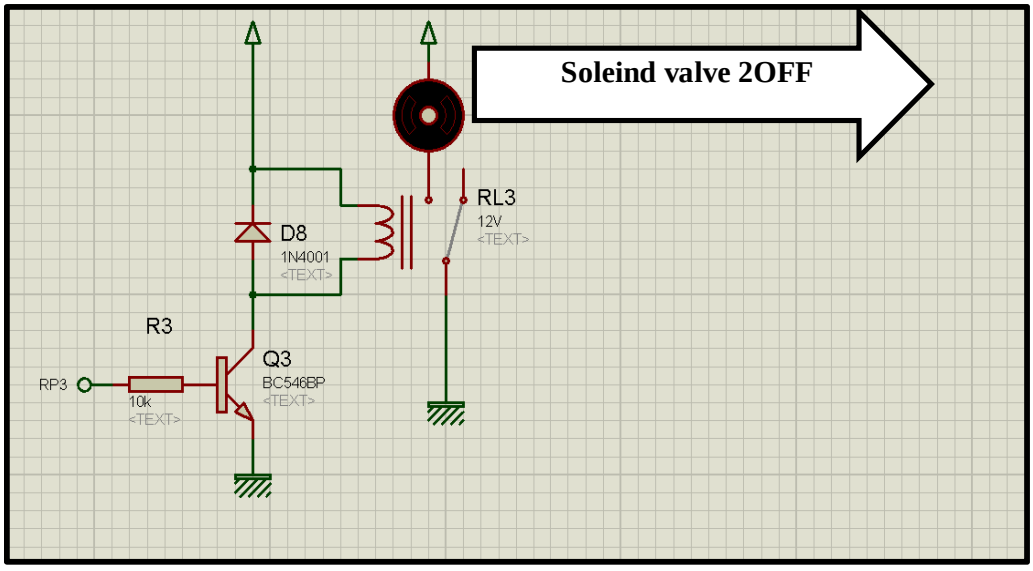


Figure (4.13): simulation of Soleind Valve 2 OFF

Also store flow value of main node and two branch node in database and directly present in webpage as show in figure (4.14).

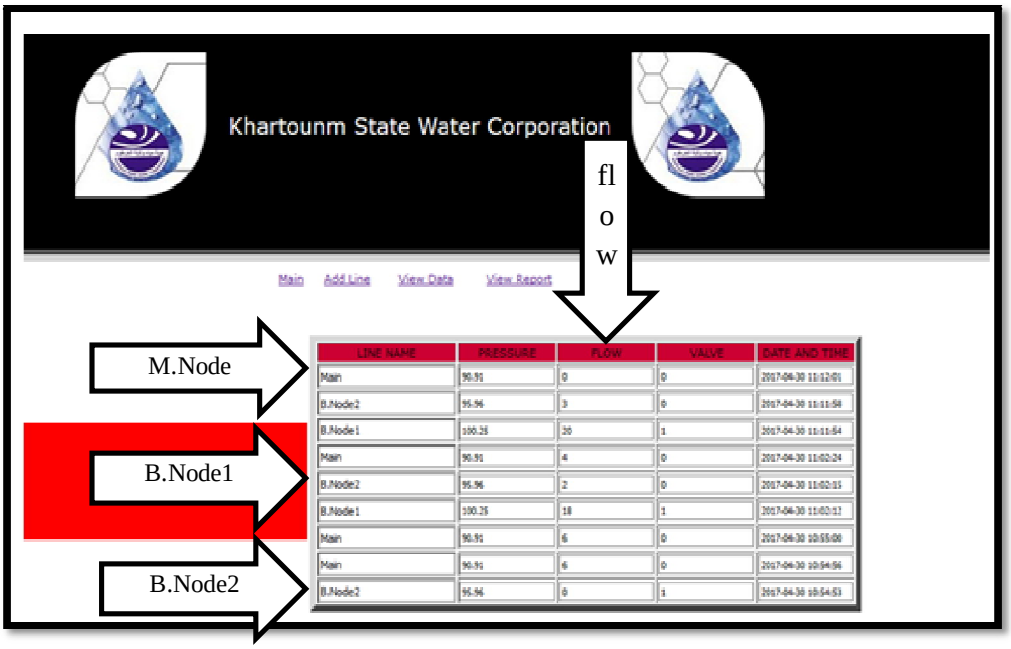


Figure **(**4.14**)**: webpage presentflow value of M.Node &B.Nodes

Fourth: **No Consumptions**

The system should continuously measure the (flow sensor value and pressure sensor value) also take the valve instance and keep it below(pressure sensor > 90 & value of flow sensor = 0 & valve instance = 1) of any line when (flow sensor and pressure sensor) arrive this value and valve instance take this value the controller automatically send frame to web page, and webpage display a decrease the pump by 270 on webpage.

In our system simulation in Proteus software(pressure sensor > 16 & value of flow sensor = 0 & valve instance = 1) as show in figure (4.15).

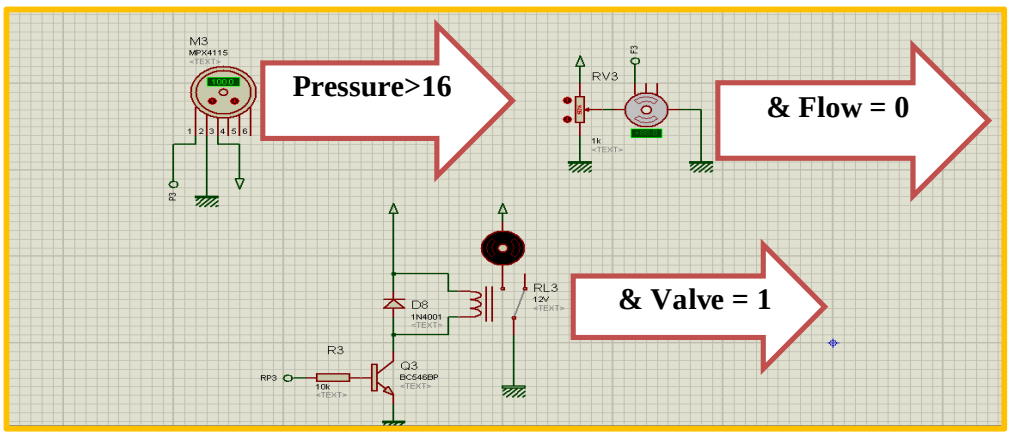


Figure (4.15): simulation of No Consumptions

display a message decrease the pump by 270 onwebpageas show infigure (4.14).

if (Pressure1>16 , Flow1 =0, Valve1=1) then display a message No consume in line1 decrease the pump by 270 on webpage as show in figure (4.16).

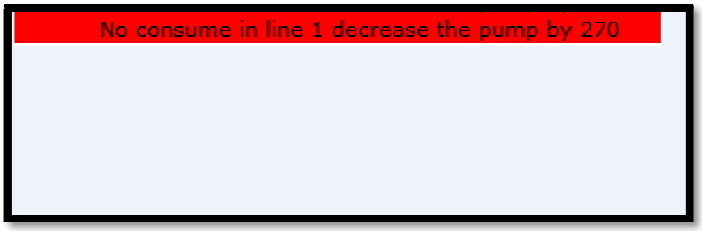


Figure (4.16): webpage display a message No consume in line1 decrease the pump by 270

if (Pressure2>16 , Flow2 =0, Valve2 =1) then display a message No consume in line2 decrease the pump by 270 on webpage as show in figure (4.17).

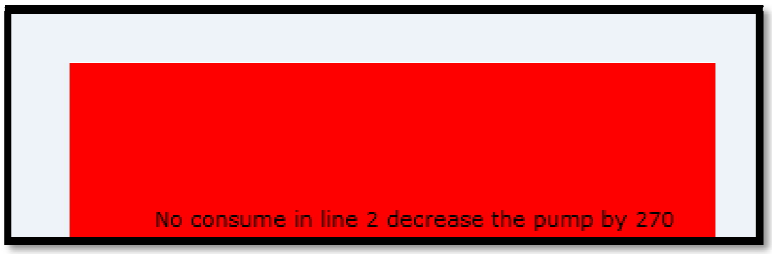


Figure (4.17): webpage display a message No consume in line2 decrease the pump by 270

Also store(pressure,flow) value& valve instance of two branch node in database and directly present in webpage as show in figure (4.18).

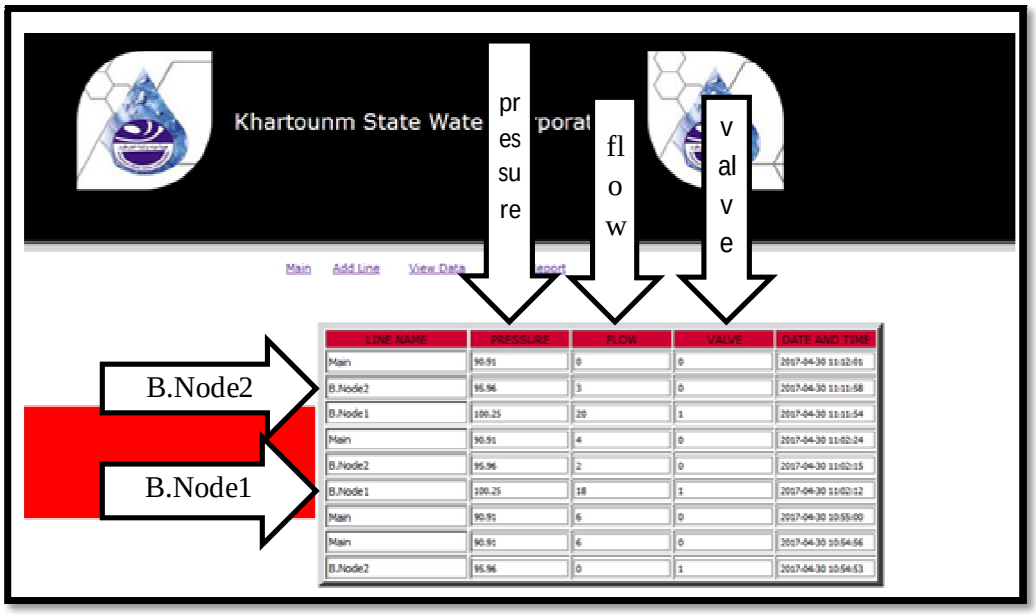


Figure **(**4.18**)**: webpage present(pressure, flow) value& valve instance of B.Nodes

Fifth:Stablest Instance

If value of pressure sensor in main node equal total value of pressure sensor in branches nodes then display a messagestablestinstance on webpage

In our system simulation in Proteus software(value of pressure sensor in main node equal total value of pressure sensor in branches nodes) as show in figure (4.19).

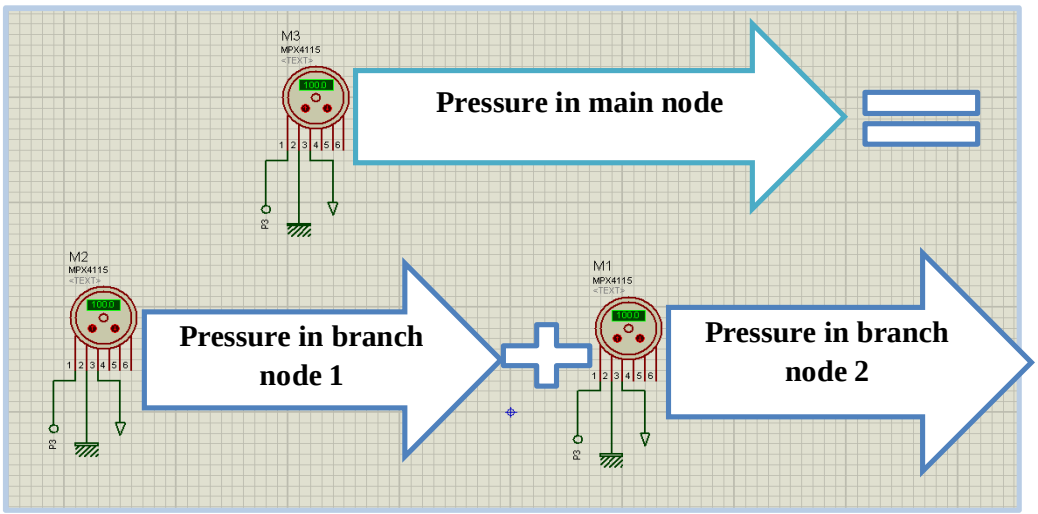
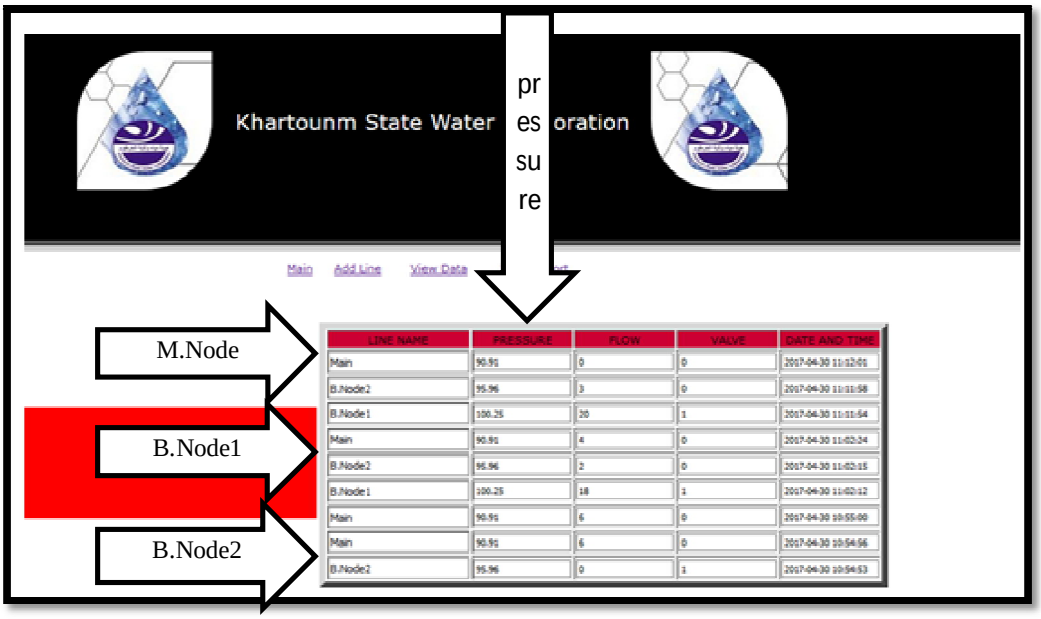


Figure (4.19): simulation of Stablest Instance

display a message stablest instance onwebpageas show infigure (4.20).

Figure (4.20): webpage display a message stablest instance

Also store pressure value of main node and two branch node in database and directly present in webpage as show in figure (4.21).



Figure(4.21): webpage present pressure value of M.node &B.Nodes

Sixth: burst

The system should continuously measure the (flow sensor value and pressure sensor value of main node and branch node) and keep it below(pressure sensor of main node >50 , pressure sensor of branch node = 0 & value of flow sensor > 0) of any line when (flow sensor and pressure sensor) arrive this value the controller automatically send frame to web page, and web page send frame to main node turnSoleind valve OFF of this line.

In our system simulation in Proteus software(value of pressure sensor in main node , pressure sensor in branches nodes & flow rate) as show in figure (4.22).

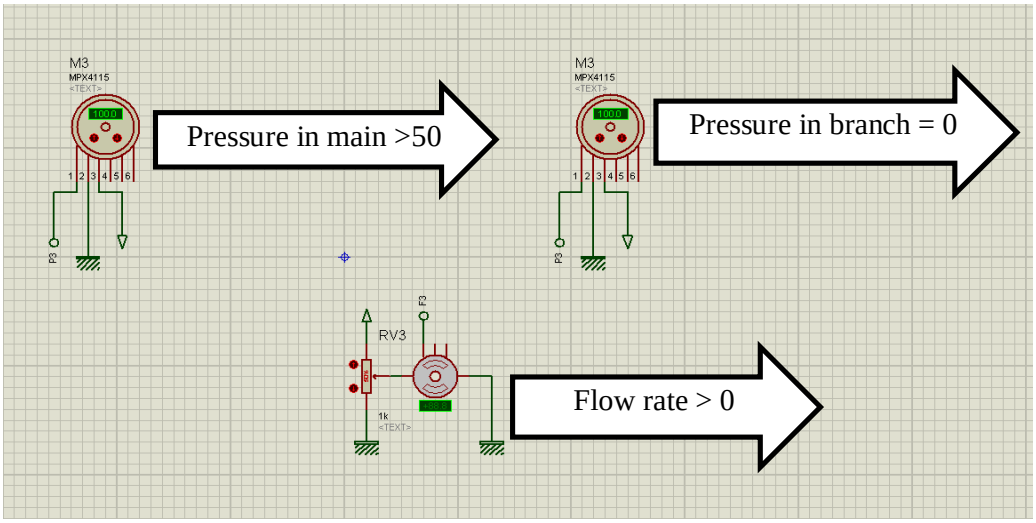


Figure (4.22): simulation of burst

if (Pressure in main node >50 , pressure1 = 0 & Flow1 > 0,) then turn Soleind valve1 OFFas show in figure (4.23).

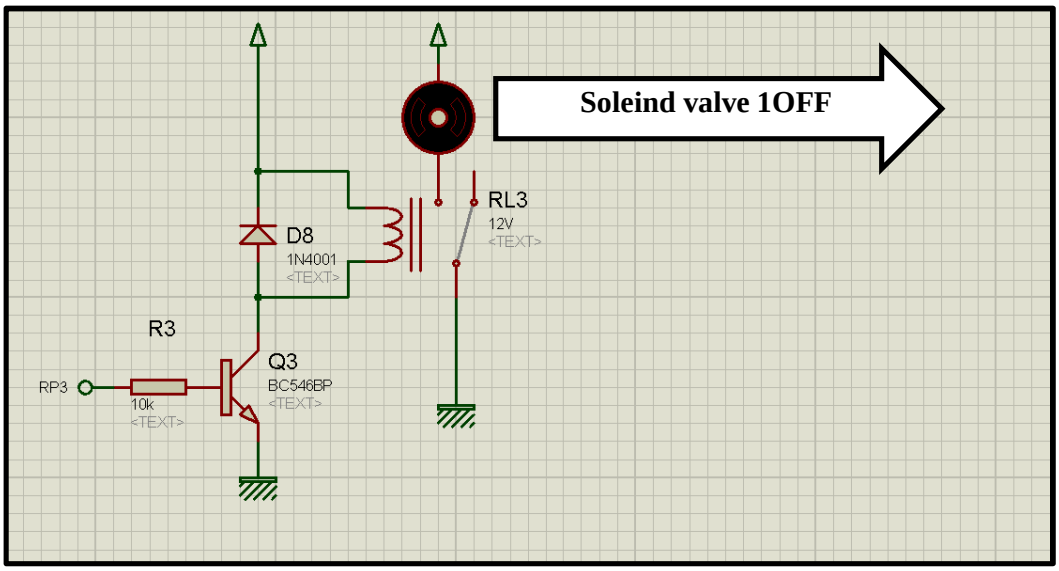


Figure (4.23): simulation of Soleind Valve 1 OFF

if (Pressure in main node > 50 , pressure2 = 0 &Flow2 >0) then turn Soleind valve2 OFF as show in figure (4.24).

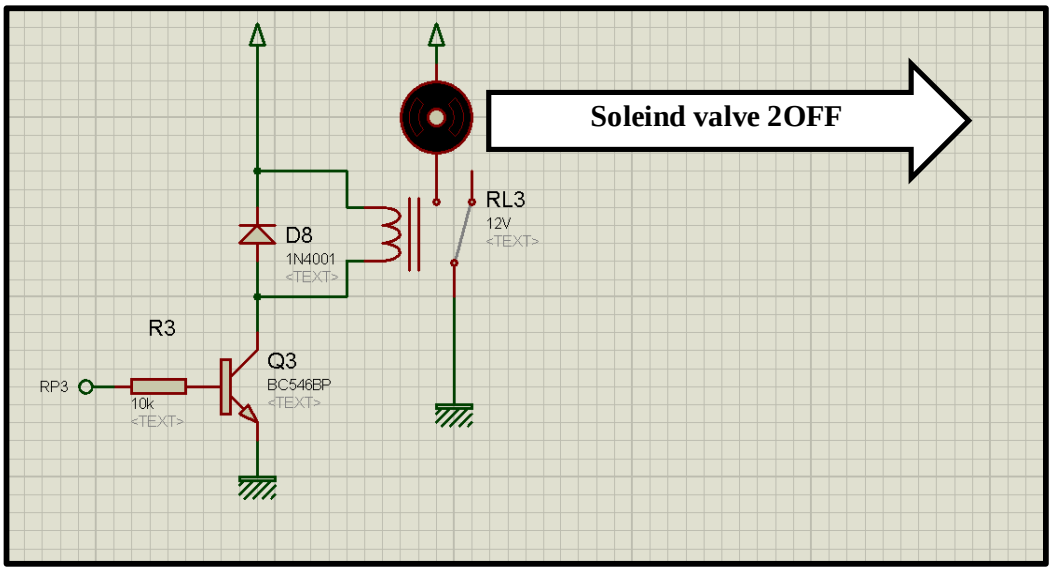
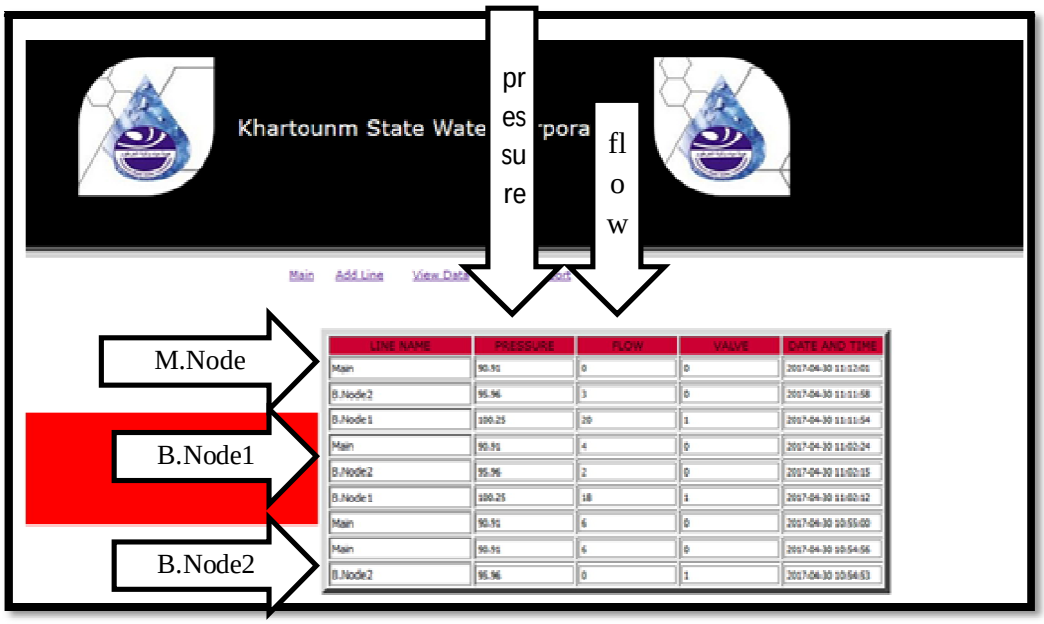


Figure (4.24): simulation of Soleind Valve 2 OFF

Also store(pressure sensor of main node , pressure sensor of branch node,flow rate) in database and directly present in webpage as show in figure (4.25).



Figure(4.25): webpage present pressure value of M.node &B.NodesP& flow rate

When the designed program is executed and after measure sensor value, MainForm is displayed to perform the following:

First: The XBees s1 that connect with each other to open the connection with the node.

Second: after connection establish the webpage send frame to main node. Main node received frame that as shown in Figure (4.26).

1	1	1	1
Start of frame(*)	**Node ID**	**Valve instance**	**End of Frame(#)**

Figure (4.26): Frame Received by Main node from Webpage

Third: after connection establish the main node send frame to the all branches nodes. all the branches nodes receive that frame as shown in Figure(4.27):

1	1	1	1
Start of frame(*)	**Node ID**	**Valve instance**	**End of Frame(#)**

Figure (4.27): Frame Received by All Branches Nodes from the Main node

Fourth: each branch node start measuring the network parameters if it has the same ID in the received frame as shown in Figure (4.28):

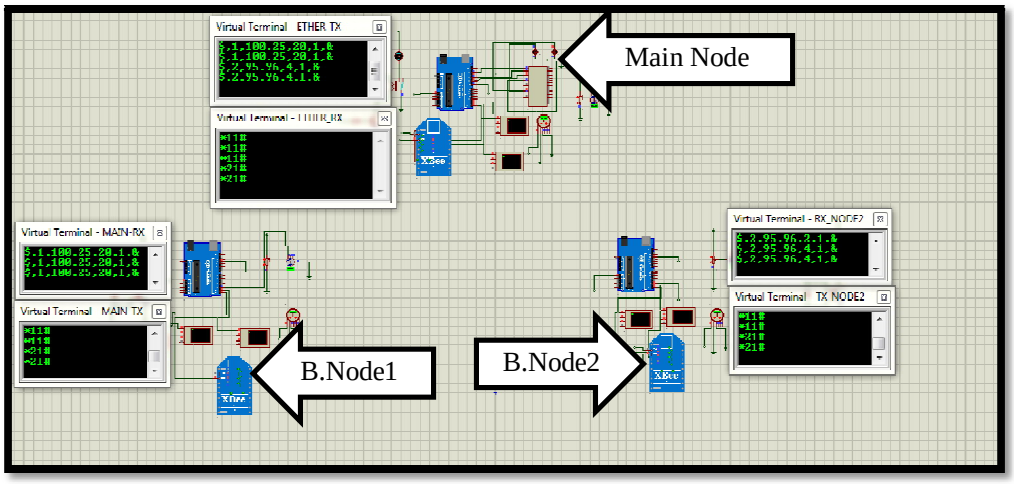


Figure (4.28): Measure network parameters

Fifth: each branch node responses by frame sends wirelessly to the main node via XBee s1 module as shown in Figure (4.29).

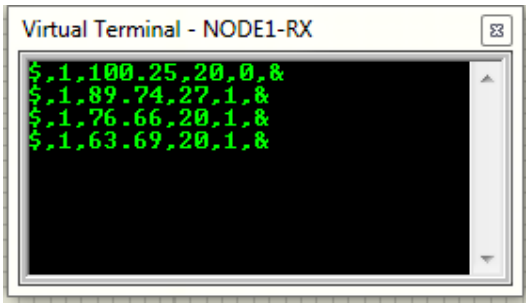


Figure 4.29(a): FrameSends from Branch Node-1 to the Main node

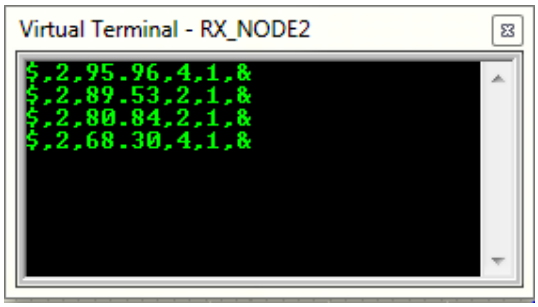


Figure 4.29(b): Frame Sends from Branch Node-2 to the Main node

Sixth:Main node will receive the sensor data sent by XBee wirelessly, as shown in Figure 4.30(a) and it will send them via Ethernet and store in the Data Base, as shown in Figure 4.30(b). That can be done in an automatic way.

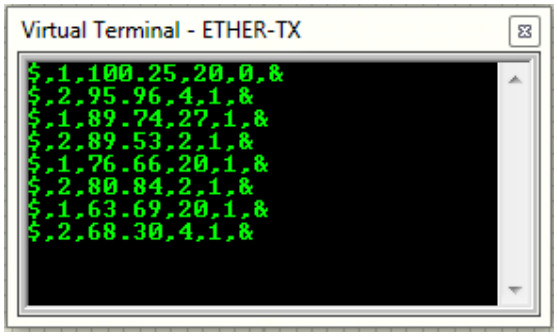
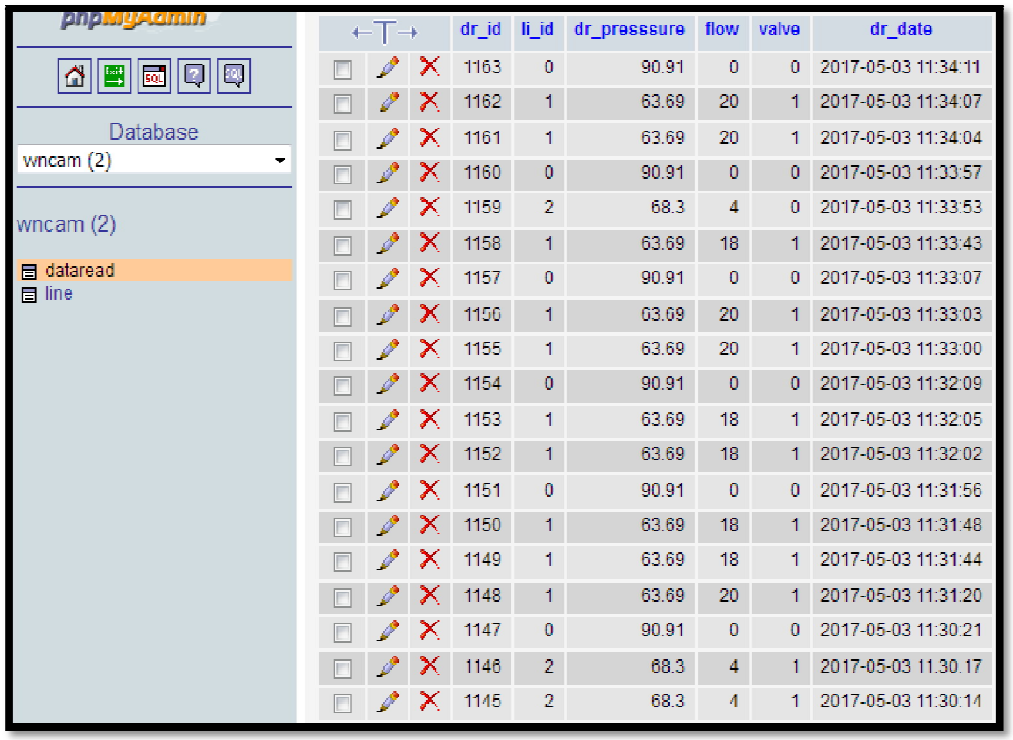


Figure 4.30(a) : All frames received by main node



dr_id	li_id	dr_presssure	flow	valve	dr_date
1163	0	90.91	0	0	2017-05-03 11:34:11
1162	1	63.69	20	1	2017-05-03 11:34:07
1161	1	63.69	20	1	2017-05-03 11:34:04
1160	0	90.91	0	0	2017-05-03 11:33:57
1159	2	68.3	4	0	2017-05-03 11:33:53
1158	1	63.69	18	1	2017-05-03 11:33:43
1157	0	90.91	0	0	2017-05-03 11:33:07
1156	1	63.69	20	1	2017-05-03 11:33:03
1155	1	63.69	20	1	2017-05-03 11:33:00
1154	0	90.91	0	0	2017-05-03 11:32:09
1153	1	63.69	18	1	2017-05-03 11:32:05
1152	1	63.69	18	1	2017-05-03 11:32:02
1151	0	90.91	0	0	2017-05-03 11:31:56
1150	1	63.69	18	1	2017-05-03 11:31:48
1149	1	63.69	18	1	2017-05-03 11:31:44
1148	1	63.69	20	1	2017-05-03 11:31:20
1147	0	90.91	0	0	2017-05-03 11:30:21
1146	2	68.3	4	1	2017-05-03 11:30:17
1145	2	68.3	4	1	2017-05-03 11:30:14

Figure 4.30(b): Stores (sensor nodes) the network parameters in the Database

Seventh: Then display the readings on (web site) the monitoring screen as shown in Figure 4.31.

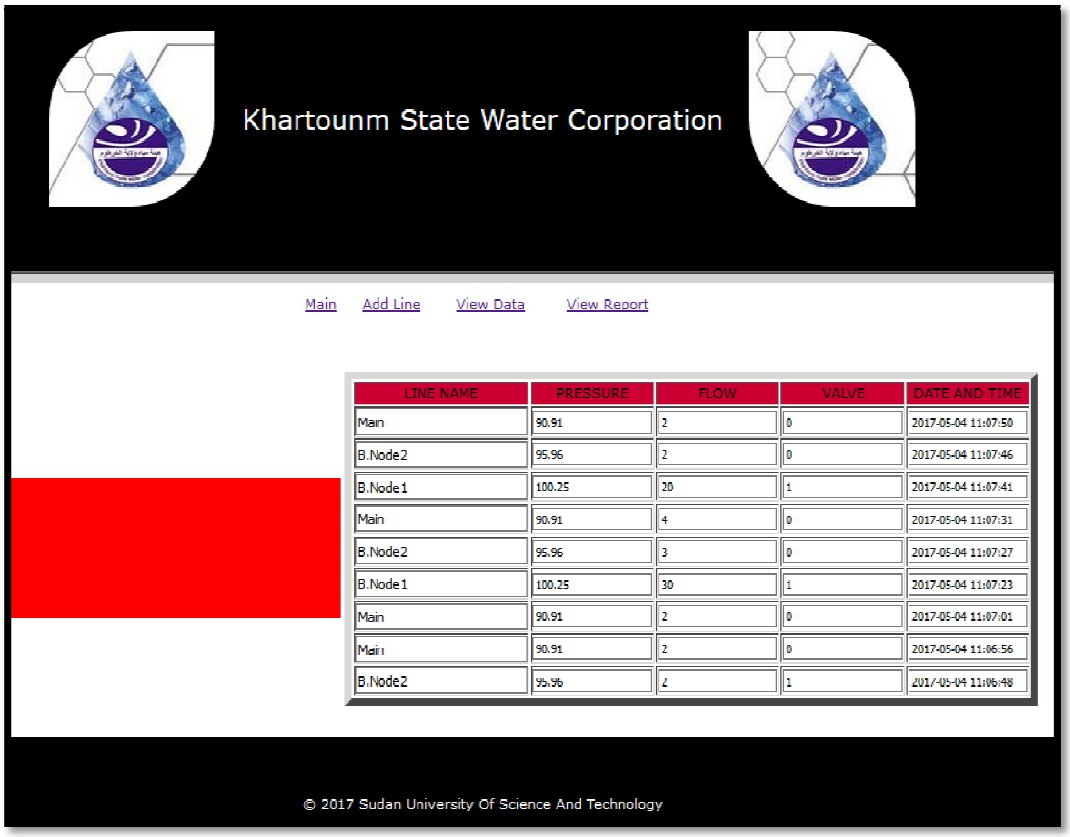
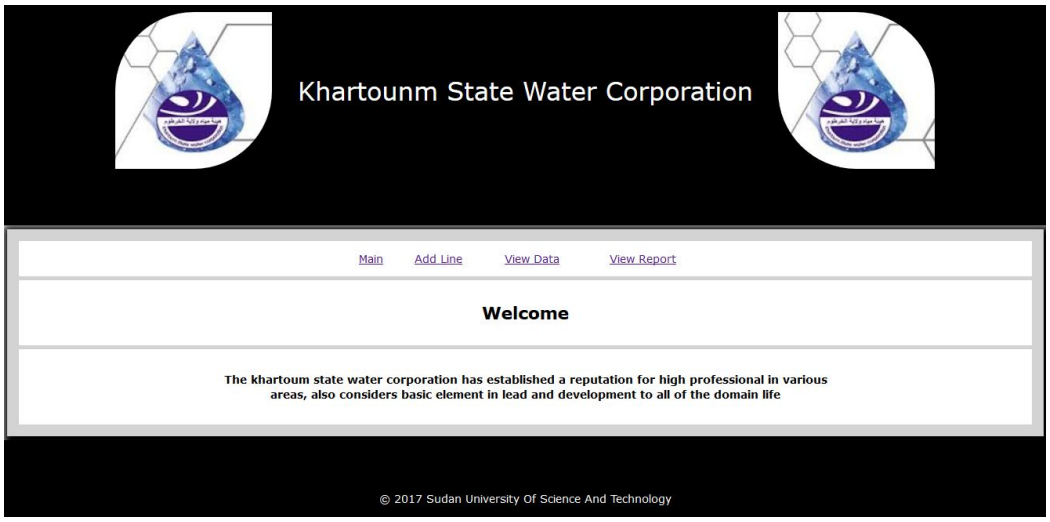


Figure (4.31): sensor nodes values on web site

The Web Site Consist of:

First: main page: contain of introduction of Khartoum state water corporation. Also hybrid links to remain pages [Data come from network page(data page) ,Report page, Add line page]. As shown in figure (4.32).



Figure(4.32): Main page for Khartoum state water corporation

Second: view data page: contain of all data come from circuits found in lines. The first field consist of name of lines and the rest fields represents the data. The last field consist of date and time when received the data.as shown in figure 4.31.

Third: view report: consist of list of lines that participate in the service as shown in figure 4.33(a). also consist of display button when click on to selecting line as show in figure 4.33(b). also it display all data of the line that selects if wont to consult .as shown in figure 4.33(c).

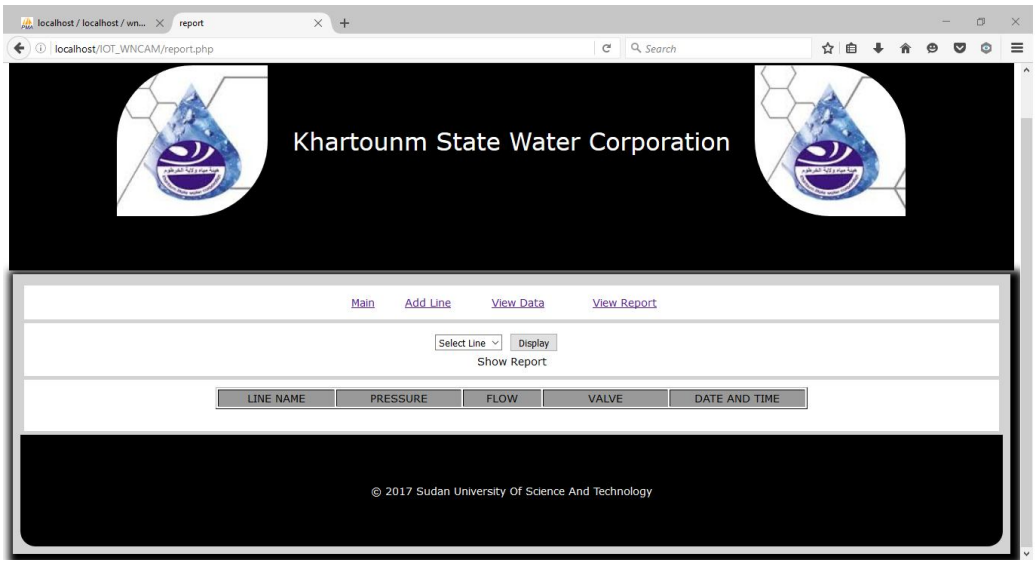


Figure 4.33(a) : View report page

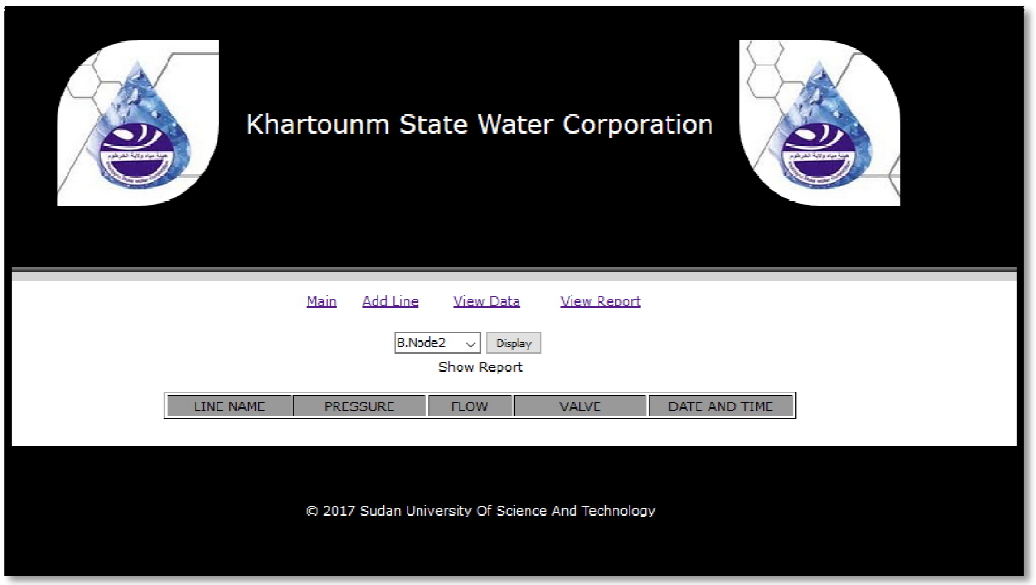


Figure 4.33(b) : select B.Node2

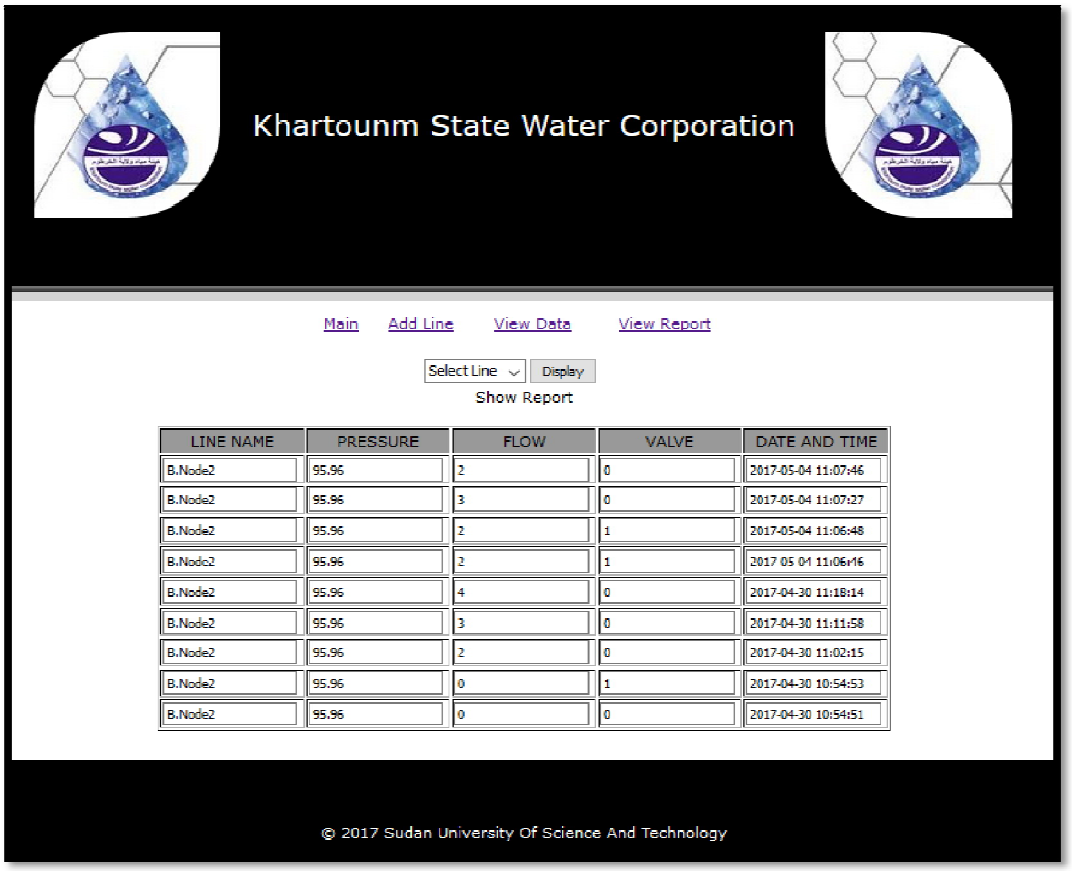


Figure 4.33(c) : View report of B.Node2

Fourth: Add Line : insert the data of Line that wont to participate in services, and it consist of line name, number of house, and number of person .as show on figure 4.27.

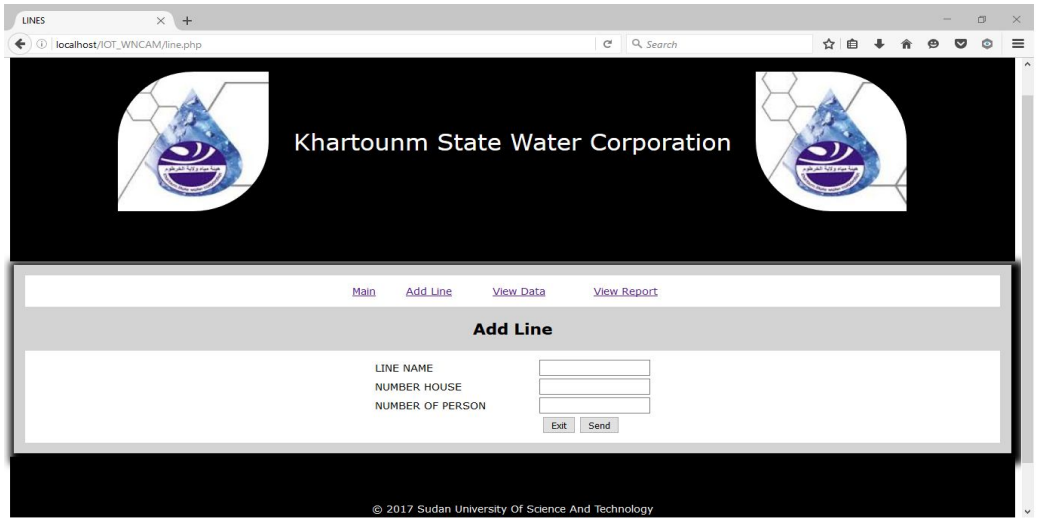


Figure 4.34: Add line to network

4.3 Discussion

- Internet of things is main important things to remote monitor therefore the first step design must prepared the Internet.
- The whole wireless XBee used in this design is about 1.5 kilo meter coverage that areas to monitoring system must in not than 1.5 kilo meter far from main node.
- Whole nodes depending on ID that require to send the frame contain of all network parameters that measure by sensors to the main node.
- the main node recieved these value, send it via ethernet to data base.
- This system can reducing risk of network, fully perceivable , reliable transmission and information processing.
- also provides service interfaces to the people. Hence people easily consult different kinds of smart interfaces of Internet about the status and information of things.
- The IoT provide dynamically selecting sensor information, as well as for structuring this information on daily operations in purpose for predictive maintenance, reduction of unplanned stoppages or other critical processes to reduce cost of network operation.

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions:

In this work, a proposed method based on internt of things Technology to design and implement a simple, inexpensive, and easy movement remote water distribution network control and monitoring system consist of three sensors nodes(main node, two branches nodes) located in different places to measure all of the surrounding network parameters via two sensors (pressure and flow sensor),control via SolenoidValve and main node consider data collected. two branches nodes transmit the measured values to a wireless receiver board (xbeeS1) on the main node. main node transmit the measured values and data sending by xbee via Ethernet to server. By inverse web page can send frame to main node to controlling ON/OF in specific solenoid valve, then short Message Service(SMS) message special of water distribution network instance display on monitoring screen. Simulation result shows the sensor values, store it in a database, and displayed it in web site on monitoring screen in host PC. This system is more accurate, simple, less used of complex circuit, and highly flexible and open to further development.

5.2 Recommendations:

This research recommends to:

1- implement this system as hardware and apply in real network because is solve the problem of network.

2- Use another wireless communication module with wide rage such as very high frequency device (VHF) to control and monitoring network from anywhere and anytime.

3- Communication all node with GIS system because GIS program clears all data in network clearly(descriptive data, where data) .

5.3 Reference

1) N.B.Bhawarkar, N.M.Verulkar, R.R.Ambalkar, K.U.Pathak, (2016) ” Automatic Water Distribution System Using ARM Controller ” , IJRE- International Journal of Research in Electronics.

2)Pranita Vijaykumar Kulkarni, Mrs. M. S. Joshi, (2016), ”An IOT based Water Supply Monitoring and Controlling System with Theft Identification ”, International Journal of Innovative Research in Science, Engineering and Technology. Vol. 5, Issue 9,September,2016.

3) Sokratis Kartakis, Edo Abraham, Julie A. McCann, (2015), ” A Testbed forMonitoring and Controlling Smart Water Networks”.

4) Harish K M, Chaitra R, Deepika R, Divya K S, Nandini M (2016), ” Centralised Water Distribution Monitoring and Controlling System Using PLC and SCADA” International Journal of Science, Engineering and Technology Research(IJSETR) Volume5, Issue6, June 2016.

5) AdityaRaj Hemrajan, Bindal Patel, Asst. Prof. Shreeji Sheth, (2016) ” AutomaticWater Distribution System Using Arduino UNO ”, International Journal of Engineering Research and Development (IJERD) ISSN: 2278-067X, (RTEECE 08t – 09th April 2016).

6) Kainka, F. Das Franzis Starterpaket Arduino Mega 2560. Haar: (2013).

Jie Yang(2013), ”A Harsh Environment Wireless Pressure Sensing Solution Utilizing High Temperature Electronics ”, sensorsISSN 1424-8220,www.mdpi.com/journal/sensor.

7) Ria Sood, Manjit Kaur, Hemant Lenka(2013), ”Design andDevelopment of Automatic Water FlowMerer”,International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.3, No.3, June 2013.

8) MAPLA, (2013), ”A survey of non-invasive and semi-invasive flow meters for mining applications ”.

9) Devika, S. V., et al. "Arduino Based Automatic Plant Watering System. "*International Journal of Advanced Research in Computer Science and Software Engineering* 4.10 (2014).

10) I.S.Tawade, M.S.Pendse, H.P.Chaudhari ,Design and Development of Saline Flow Rate Monitoring System Using

11) Flow Sensor, Microcontroller and RF ZigBee Module , International Journal of Engineering Research and General Science Volume 3, Issue 3, May-June, 2015

ISSN 2091-2730

12) W B Bache (British Pressure Gauge Manufacturers Association), P Clow, T J Thompson (both UKAS), C Duncombe (BSI), L March (Kistler Instruments Ltd), N A Morgan (Theta Systems Ltd) and R White (Pfeiffer). Published by the Institute of Measurement and Control. Further copies are available from the Institute.

13) © Crown Copyright 1998. Reproduced by permission of the Controller of HMSO. ISBN 0 904457 29 X

14) Alldatasheet.com.(2016).*MPX4115 Datasheet(PDF)-Motorola, Inc*. Available at:

15)http://www.alldatasheet.com/datasheet,pdf/pdf/5178/Motorola/MPX4115.html[Ac c-essed 2 Novomber 2015]

5.4 APPENDIX

Code for Main Node and Branches Nodes

```
# include <SPI.h>
#include <EtherCard.h>
#include <IPAddress.h>
#include <net.h>
 #include <EthernetUdp.h>
include <EEPROM.h>#
//////////////////////////AVR/////////////////////////////////////////////////////
const int FAN1=8;
const int pre_sensor = A0;  //PIN PRE
#define kpa2atm 0.00986923267
int pre_val = 0;       // value read from the pot
float pkPa; // pressure in kPa
float pAtm; // pressure in Atm
//////////////////////////////////////////////////////////////////////////////////
const int encoder_pin = 2;  // The pin the encoder is connected
unsigned int rpm;     // rpm reading
volatile byte pulses;  // number of pulses
unsigned long timeold;
//The number of pulses per revolution //
//depends on your index disc!!
unsigned int pulsesperturn = 20;
int FAN1_ST=0x30;
//////////////////////////////////////////////////////////////////////////////////////
unsigned char forme[255],frame_bit;
char forme1[20],forme2[20],forme3[20],forme4[20];
int i,j,n;
long count=0;
```

```
//////////////////////////////////////////////

char form2[50];

//char buffer[500];

int aa;

int tt,flag;

;int *p

//////////pressure////////////////////////////////////////////////

void read_pressure(){

);pre_val = analogRead(pre_sensor

;pkPa = ((float)pre_val /(float)1023+0.095)/0.009

;pAtm = kpa2atm*pkPa*100

));EEPROM.put(0xB31,char(pAtm

pAtm=pAtm*100;

));//Serial.println(String(pAtm

);delay(200

}

//////////flowmeater//////////////////////////////////////////////////////////

void Read_flow(){

if (millis() - timeold >= 1000){  /*Uptade every one second, this will be equal to reading
frecuency (Hz).*/

//Don't process interrupts during calculations

);detachInterrupt(0

//Note that this would be 60*1000/(millis() - timeold)*pulses if the interrupt

//happened once per revolution

;rpm = (60 * 1000 / pulsesperturn )/ (millis() - timeold)* pulses

;//EEPROM.put(0xA40,(String)rpm

();timeold = millis

;pulses = 0
```

```
//Write it out to serial port
Serial.print("RPM//
//Serial.println(rpm,DEC
//Restart the interrupt processing
);attachInterrupt(0, counter, FALLING
}
}
//////////////////MOTOR/////////////////////////////////////
void fan_control(unsigned char f1 ){
;}if(f1=='1'){ digitalWrite(FAN1, HIGH);FAN1_ST=0x31
;}}if(f1=='0'){digitalWrite(FAN1, LOW);FAN1_ST=0x30
////////////xbee read/////////////////////////////////////////
void Read_XPEE(){
()){while (Serial.available
frame_bit= Serial.read();
{=='$')if( frame_bit
;i=1
{do
()){if(Serial.available
();frame_bit= Serial.read
;forme[i] = frame_bit
//if(forme[2]=='1'){  ////Node 1
EEPROM.write(0x00+i, forme[i];
//}
//if(forme[2]=='2'){  /////Node2

EEPROM.write(0x20+i, forme[i];
```

```
// }
i++ ;
}
while (frame_bit!='$');}}
;frame_bit=0
}
}
//////////////////read/////////////////////////////////////////
void Read_EEprom(){
////////// node 1 ////////////
for(j=0x01;j<=0x0E;j++){
forme1[n]= EEPROM.read(j);
++;}n
;n=1
////////// node 2 ////////////
++ //for(j=0x21;j<=0x2A;j
//
//forme2[n]= EEPROM.read(j);
}
//////////////////////////ethernet//////////////////////////////
#define STATIC 1  // set to 1 to disable DHCP (adjust myip/gwip values below )
#if STATIC
//ethernet interface ip address
static byte myip[] = { 192,168,1,122 };//192,168,1,122
//gateway ip address
static byte gwip[] = { 192,168,1,1};//192,168,1,1
static byte subnet[]={255, 255, 255, 0};
```

```
;unsigned int localPort = 1337

#endif

//unsigned int localPort = 8888 ;

//ethernet mac address - must be unique on your network

static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31};

//////////////////////////////////////////////////////////

//////////////////////////////////////////////

char
form[50];//{0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x3
0};

byte Ethernet::buffer[500]; // tcp/ip send and receive buffer

//BufferFiller bfill ;

//EthernetUDP Udp;

void udpSerialPrint(uint16_t dest_port, uint8_t src_ip[IP_LEN], uint16_t src_port, const char
*data, uint16_t len){

]);IPAddress src(src_ip[0],src_ip[1],src_ip[2],src_ip[3

//

:");//Serial.print("dest_port

);//Serial.println(dest_port

:");//Serial.print("src_port

//Serial.println(src_port);

//

//

//Serial.print("src_port:");

//ether.printIp(src_ip);

();//Serial.println

:");//Serial.println("data

);Serial.println(data

//////////////////////////////////////////////
```

```
]flag=data[1
){if(  flag!=0x30
);}ether.sendUdp(forme1 ,sizeof(forme1), 1337, src_ip, src_port
//////////////////////////////////////////////
){if(  flag==0x30
);}ether.sendUdp(forme2 ,sizeof(forme2), 1337, src_ip, src_port
//////////////////////////////////////////////
}
void setup(){
://put your setup code here, to run once
);Serial.begin(9600
);pinMode(FAN1,OUTPUT
);pinMode(encoder_pin, INPUT
//Interrupt 0 is digital pin 2, so that is where the IR detector is connected
)//Triggers on FALLING (change from HIGH to LOW
);attachInterrupt(0, counter, FALLING
//Initialize
;pulses = 0
;rpm = 0
;timeold = 0
");//Serial.println("\nHI
){if (ether.begin(sizeof Ethernet::buffer, mymac) == 0
");//Serial.println( "Failed to access Ethernet controller
}
//Serial.println( "step1");
#if STATIC
//Serial.println( "step1");
```

);ether.staticSetup(myip, gwip

#else

())if (!ether.dhcpSetup

");//Serial.println("DHCP failed

#endif

ether.printIp("IP: ", ether.myip);

ether.printIp("GW: ", ether.gwip);

ether.udpServerListenOnPort(&udpSerialPrint, 1337);

}

void loop(){

://put your main code here, to run repeatedly

());ether.packetLoop(ether.packetReceive

Read_XPEE ();

();Read_EEprom

();read_pressure

();Read_flow

//

){if(forme[1]==0x30

]);}fan_control(form[2

//

]='$';forme2[0

forme2[1]=',';

//

]='$';forme1[0

forme1[1]=',';

//////////////////////////////

//if(form2[0]=='0';){

';forme2[2]='0

```
]=',';forme2[3
forme2[4]=int(pAtm)/1000    + 48;
;forme2[5]=int(pAtm/100)%10 + 48
]=',';forme2[6
;forme2[7]=int(pAtm/10)%10  + 48
forme2[8]=int(pAtm)%10 + 48;
]=',';forme2[9
forme2[10]=char (rpm);   ////(rpm)
]',';forme2[11
;forme2[12]=FAN1_ST
]='$';forme2[13
//forme2[15]='6';
);Serial.println(rpm
//}
////////////////////////////////
//forme1[10]='&';
//////////////////////////////////////////////////////
//forme2[0]='$';
//forme2[1]=',';
//forme2[10]='&';
////////////////////////////////////
//count==){
if(count==1000){
;("*")//Serial.print
//  //  delay(30);
");//Serial.print("1
//  //  Serial.print("3");
//  //  Serial.print("5 ");
```

```
// // delay(30);
// Serial.println ("#");
);//delay(30
////count=0 ;
//}
////////////////////////////////
//if(count==5000){
//Serial.print("*")    ;
//Serial.print("2");
//Serial.println("#")    ;
// //// count=0;
//}
////////////////////////////////}
()void counter
}
//Update count
++;pulses
}
////////////////////////////////Node1/////////////////////////////////////////////
#include <EEPROM.h>
//////////////////////////AVR/////////////////////////////
const int FAN1=12;
const int pre_sensor = A0;  //PIN PRE
#define kpa2atm 0.00986923267
int pre_val = 0;       // value read from the pot
float pkPa; // pressure in kPa
float pAtm; // pressure in Atm
////////////////////////////////////////////////////////
```

```
const int encoder_pin = 2;  // The pin the encoder is connected

unsigned int rpm;     // rpm reading

volatile byte pulses;  // number of pulses

unsigned long timeold;

//The number of pulses per revolution

!!//depends on your index disc

unsigned int pulsesperturn = 20;

//////////////////////////////////////////////////////////

int frame_bit,I;

unsigned char form[255];

int FAN1_ST=0;

//////////pressure/////////////////////////////////////////////

void read_pressure(){

);pre_val = analogRead(pre_sensor

;pkPa = ((float)pre_val /(float)1023+0.095)/0.009

;pAtm = kpa2atm*pkPa*100

//Serial.println(pAtmك)

);delay(200

}

//////////flowmeater/////////////////////////////////////////////////////////

void Read_flow(){

if (millis() - timeold >= 1000){  /*Uptade every one second, this will be equal to reading
frecuency (Hz).*/

//Don't process interrupts during calculations

);detachInterrupt(0

//Note that this would be 60*1000/(millis() - timeold)*pulses if the interrupt

//happened once per revolution

;rpm = (60 * 1000 / pulsesperturn )/ (millis() - timeold)* pulses
```

```
timeold = millis();
;pulses = 0
//Write it out to serial port
="");Serial.print("RPM//
);//Serial.println(rpm,DEC
//Restart the interrupt processing
);attachInterrupt(0, counter, FALLING
}
}
//////////////////MOTOR/////////////////////////////////////
void fan_control(unsigned char f1){
;}if(f1=='1'){ digitalWrite(FAN1, HIGH);FAN1_ST=1
;}}if(f1=='0'){digitalWrite(FAN1, LOW);FAN1_ST=0
//////////////////READ XBEE/////////////////////////////////
void read_xbee(){
while(Serial.available()){
();frame_bit = Serial.read
=='*'){if(frame_bit
;i=1
{do
if (Serial.available)){
();frame_bit = Serial.read
form[i]=frame_bit;
;)]EEPROM.write(0x09+i,form[i
++;i
}
!='#');}}while(frame_bit
```

```
);delay(1
;frame_bit=0
}}
//////////////////////send data////////////////////////////////
void send_data(){
'){if(form[1]=='1
Serial.print("$");
Serial.print  (",") ;
');Serial.print('1
Serial.print(",")  ;
);Serial.print(pAtm
Serial.print(",")  ;
);Serial.print(rpm
Serial.print(",");
);Serial.print(FAN1_ST
Serial.print (",");
Serial.println("&");
;form[1]=0
}
}
//////////////////////////////////////////////////////////////////
void setup(){
//put your setup code here, to run once
);Serial.begin(9600
);pinMode(FAN1,OUTPUT
);pinMode(encoder_pin, INPUT
//Interrupt 0 is digital pin 2, so that is where the IR detector is connected
```

```
//Triggers on FALLING (change from HIGH to LOW )
);attachInterrupt(0, counter, FALLING
Initialize//
;pulses = 0
;rpm = 0
timeold = 0;
////////////////////////////////////////////////////////////
}
void loop(){
//put your main code here, to run repeatedly
read_xbee();
send_data():
 read_pressure();
();Read_flow
]);fan_control(form[2
//FAN1_ST=EEPROM.read(0x0B)-48;
}
void counter()
{
//Update count
++;pulses
}
//////////////////////////////////////////Node2//////////////////////////////
#include <EEPROM.h>
//////////////////////////AVR//////////////////////////
const int FAN1=12;
const int pre_sensor = A0;  //PIN PRE
```

```
#define kpa2atm 0.00986923267
int pre_val = 0;       // value read from the pot
float pkPa; // pressure in kPa
float pAtm; // pressure in Atm
//////////////////////////////////////////////////////////
const int encoder_pin = 2;  // The pin the encoder is connected
unsigned int rpm;     // rpm reading
volatile byte pulses;  // number of pulses
unsigned long timeold ;
//The number of pulses per revolution
//depends on your index disc
unsigned int pulsesperturn = 20;
//////////////////////////////////////////////////////////
int frame_bit,I;
unsigned char form[255];
int FAN1_ST=0;
//////////pressure/////////////////////////////////////////////
void read_pressure(){
);pre_val = analogRead(pre_sensor
;pkPa = ((float)pre_val /(float)1023+0.095)/0.009
;pAtm = kpa2atm*pkPa*100
);//Serial.println(pAtm
);delay(200
}
//////////flowmeater//////////////////////////////////////////////////////////
void Read_flow(){
if (millis() - timeold >= 1000){  /*Uptade every one second, this will be equal to reading
frecuency (Hz).*/
```

```
//Don't process interrupts during calculations
);detachInterrupt(0
//Note that this would be 60*1000/(millis() - timeold)*pulses if the interrupt
//happened once per revolution
;rpm = (60 * 1000 / pulsesperturn )/ (millis() - timeold)* pulses
();timeold = millis
;pulses = 0
//Write it out to serial port
");=//Serial.print("RPM
);//Serial.println(rpm,DEC
//Restart the interrupt processing
attachInterrupt(0, counter, FALLING);
}
}
//////////////////MOTOR//////////////////////////////////////
void fan_control(unsigned char f1){
;}(f1=='1'){ digitalWrite(FAN1, HIGH);FAN1_ST=1
;}}if(f1=='0'){digitalWrite(FAN1, LOW);FAN1_ST=0
//////////////////////READ XBEE//////////////////////////////////
void read_xbee(){
()){while(Serial.available
();frame_bit = Serial.read
        I =1;=='*'){if(frame_bit
Do{
()){if (Serial.available
();frame_bit = Serial.read
;form[i]=frame_bit
```

```
]);EEPROM.write(0x09+i,form[i
++;i
}
!='#';}}while(frame_bit
);delay(1
;frame_bit=0
}}
//////////////////////send data////////////////////////////////
void send_data(){
'){if(form[1]=='2
Serial.print("$") ;
Serial.print(",") ;
');Serial.print('2
Serial.print(",") ;
);Serial.print(pAtm
Serial.print(",")  ;
);Serial.print(rpm
Serial.print(",") ;
);Serial.print(FAN1_ST
Serial.print(",");
Serial.println("&");
;form[1]=0
}
}
//////////////////////////////////////////////////////////////////
void setup(){
//put your setup code here, to run once
```

```
);Serial.begin(9600
);pinMode(FAN1,OUTPUT
pinMode(encoder_pin, INPUT);
 //Interrupt 0 is digital pin 2, so that is where the IR detector is connected
//(Triggers on FALLING (change from HIGH to LOW)
attachInterrupt(0, counter, FALLING);
//Initialize
;pulses = 0
;rpm = 0
;timeold = 0
////////////////////////////////////////////////////////////
}
void loop(){
://put your main code here, to run repeatedly
read_xbee();
send_data();
();read_pressure
();Read_flow
]);fan_control(form[2
//FAN1_ST=EEPROM.read(0x0B)-48 ;
}
void counter()
{
//Update count//
Pulses++;
}
////////////////////////////////////////////////
```

Web site code

```
<?php
mysql_connect("localhost","root","root");
mysql_select_db("wncam");
?>
<?php
include("conn.php");
;(error_reporting(E_ALL//
echo "<h2>UDP</h2>\n ;
$service_port = "1337 " ;
$address = "192.168.1.122";
$socket = socket_create(AF_INET, SOCK_DGRAM, SOL_UDP);
if ($socket === false) {
;echo "socket_create() failed: reason: " . socket_strerror(socket_last_error()) . "\n
} else {
";echo "OK.\n
}
result = socket_connect($socket, $address, $service_port);
if ($result === false) {
echo "socket_connect() failed.\nReason: ($result) "
.socket_strerror(socket_last_error($socket)) . "\n.
} else {
;echo "OK.\n
}
//################################B.Node2###################################;
$tow='*21# ;
$nodtow=socket_write($socket,$tow, strlen($tow)) ;
if(false !== $nodtow){
$buftow = 'This is my buffer.';
if (false !== ($bytes2 = socket_recv($socket,$buftow, 2048, MSG_WAITALL)))  {
...";//echo "Read $bytes bytes from socket_recv(). Closing socket
;echo  $buftow
include("conn.php");
$d2=explode(",",$buftow) ;
```

```
$da=date("y-m-d h:i:s");
'){if($d2[1]=='1
mysql_query("insert into
dataread"."(li_id,dr_presssure,flow,valve,dr_date)"."Values('$d2[1]','$d2[2]','$d2[3]','$d2[4]'
,'$da')");}
");mysql_connect("localhost","root","root
mysql_select_db("wncam");
$lines=mysql_query("SELECT * FROM line where li_id=2");
$row=mysql_fetch_array($lines);
'];$nhouse=$row['nhouse
;//echo $nhouse
'];$nperson=$row['nperson
;echo $nperson
);$q2=($nhouse*$nperson*115*1.45)/(86400000
;//echo $q
echo"<br>";
];$flwo2=$d2[3
$;x2=.0081*$flwo2
{)if($x2<$q2
";$qtwo="line 2decrease the pump by  170
}
)if($d2[1]>16.31&&$d2[3]=0&&$d2[4]=1
{
';$NoConsumptions2='decrease the pump by 270 line 2
}
];$xnode2=$d2[2]+$d2[3
];$valve2=$d2[4
if($xnode2<=16.31&&$valve2=1&&$d2[1]=2
{
line2='limited reached line2';
}
} else {
;";echo "socket_recv() failed; reason: " . socket_strerror(socket_last_error($socket)) . "\n
}
```

```
}

//#################################B.Node1#############################
$presssure2=$d2[2];
$flow2=$d2[2];
sleep(2);
$nodone='*11#';
$one=socket_write($socket,$nodone, strlen($nodone));
if($one=== false) {
$bufone = 'This is my buffer.';
}
else{
if (false !== ($bytes1 = socket_recv($socket,$bufone, 2048, MSG_WAITALL))) {
echo  $bufone;
include("conn.php");
);$d1=explode(",",$bufone
");$da=date("y-m-d h:i:s
mysql_query("insert into
dataread"."(li_id,dr_presssure,flow,valve,dr_date)"."Values('$d1[1]','$d1[2]','$d1[3]','$d1[4]'
;("(','$da
");mysql_connect("localhost","root","root
mysql_select_db("wncam");
$lines=mysql_query("SELECT * FROM line where li_id=1");
$row=mysql_fetch_array($lines);
;$nhouse=$row['nhouse']
'];$nperson=$row['nperson
);$q1=($nhouse*$nperson*115*1.45)/(86400000
];$flwo=$d[3
;$Q1=.0081*$flwo
if($Q1<$q1)
{
;$one="line1 decrease the pump by  170
}
)if($d1[2]>16.31&&$d1[3]=0&&$d1[4]=1
{
```

```
';$NoConsumptions1='decrease the pump by 270 line 1
}
];$xnode1=$d1[2]+$d1[3
$valve1=$d1[4];
if($xnode1<=16.31&&$valve1=1 && $d1[1]=1)
{
$line1='limited reached line1';
}
} else {
;echo "socket_recv() failed; reason: " . socket_strerror(socket_last_error($socket)) . "\n
}
}
$presssure1=$d1[2];
$flow1=$d1[3];
//##############################main##############################
sleep(2);
$nodmain='*00#;
$nmain=socket_write($socket,$nodmain, strlen($nodmain));
if(false !== $nmain){
$bufmain;
if (false !== ($bytesm = socket_recv($socket,$bufmain, 2048, MSG_WAITALL))) {
...";//echo "Read $bytes bytes from socket_recv(). Closing socket
;echo  $bufmain
");include("conn.php
);$dm=explode(",",$bufmain
");$da=date("y-m-d h:i:s
mysql_query("insert into
dataread"."(li_id,dr_presssure,flow,valve,dr_date)"."Values('$dm[1]','$dm[2]','$dm[3]','$dm
'[4]','$da')");
");mysql_connect("localhost","root","root
mysql_select_db("wncam");
$maiflow=$dm[3];
$mainpres=$dm[2];
av=maiflow/2;
```

```
////////////////////////////////////////////////////////////////////////////////////////
if($mainpres==$presssure2+presssure1)
{
$prmain="STABLEST INSTANCE";
}
if($flow2<$av)
}
socket_write($socket,'*20#', strlen($in));
;"echo "22222222222222222222222222222
{
(if($flow1<$av
}
;((socket_write($socket,'*10#', strlen($in//
;"echo "11111111111111111111111111111
}
////////////////////////////////////////////////////////////////////////////////////////
mysql_connect("localhost","root","root");
mysql_select_db("wncam");
;$presssure
$lines=mysql_query("SELECT *FROM dataread WHERE li_id =0 ORDER BY dr_id DESC LIMIT 0
, 1");
if($row=mysql_fetch_array($lines))
{
$'];presssure=$row['dr_presssure
;echo $presssure
;"<echo"<br
;['flow=$row['flow$
;echo $flow
>";echo"<br
'];$valve=$row['valve
    ;echo $valve
$;fmai=$flow/2
        }
>";echo"<br
;$presssuren1
```

```
$nod=mysql_query("SELECT *FROM dataread WHERE li_id =1 ORDER BY dr_id DESC
LIMIT 0 , 1");
))if($row=mysql_fetch_array($nod
{
'];$presssuren1=$row['dr_presssure
;echo $presssuren1
>";echo"<br
'];$flown1=$row['flow
;echo $flown1
>";echo"<br
'];$valven1=$row['valve
;echo $valven1
}
$presssuren2
>";echo"<br
$nod2=mysql_query("SELECT *FROM dataread WHERE li_id =2 ORDER BY dr_id
DESC LIMIT 0 , 1")
if($row=mysql_fetch_array($nod2))
{
$presssuren2=$row['dr_presssure'];
;echo $presssuren2
>";echo"<br
'];$flown2=$row['flow
;echo $flown2
>";echo"<br
'];$valven2=$row['valve
;echo $valven2
}
){if($presssure>= 50 && $presssuren1==16.31 &&$flown1>0
socket_write($socket,'*10#', strlen('*10#'));
>";echo"<br
)   ";echo "close line1(burst
}
);sleep(2
```

```
){if($presssure>50 && $presssuren2 ==16.31 && $flown1>0
#'));socket_write($socket,'*20#', strlen('*20
>";echo"<br
)";echo "close line2(burst
}
//-----------------------------------------------------------------------------------------------------------
){if($flown2<$fmai
socket_write($socket,'*20#', strlen('*20#'));
>";echo"<br
)";echo "close line2 (leakage
}
){if($flown1<$fmai
'));socket_write($socket,'*10#', strlen('*10#
>";echo"<br
)";echo "close line1 (leakage
}
}
//*******************************
}
echo "Closing socket...";
socket_close($socket);
echo "OK.\n\n";
?>
DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" !>
<""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd
<"html xmlns="http://www.w3.org/1999/xhtml>
<head>
</ "meta http-equiv="Content-Type" content="text/html; charset=windows-1256>
<title>sensor</title>
<style>
Body {
;font-family: Verdana,sans-serif
;font-size: 0.9em
}
```

```
div#header {
;padding: 50px
;color: #FFFFFF
;background-color: #000000
;box-shadow: 5px 5px 10px 10px
;border-radius: 20px 20px 0px 0px
}
div#footer {font-size:16px ;
padding: 50px;
;color: #FFFFFF
;background-color: #000000
;box-shadow: 0px 0px 0px 0px
;border-radius: 0px 0px 20px 20px
}
div#content {font-size:16px ;
;margin: 5px
;padding: 10px
;background-color: lightgrey
;box-shadow: 5px 5px 10px 10px
}
div.article { font-size:16px
;margin: 5px
;padding: 10px
;background-color: white
}
}
</style>
</head>
<body>
<center>
<"div id="header>
<"table border="0" width="1054" height="200>
<TR>
```

```
TD width="200"><img src="do (1).jpg" / height="200" width="200" style=" border-radius:
100px 0px 100px 0px;"></TD>
<"TD width="632" align="center"><font color="#FFFFFF" size="+3>
>Khartounm State Water Corporation </font
</div></TD><TD width="200"><img src="do (1).jpg" / height="200" width="200" style=
"border-radius: 0px 100px 0px 100px;"></TD>
</TR>
</table><br /></div>
<"div id="content>
<"div class="article"><table width="455" border="0" align="center>
</tr>
<td><div align="center"><a href="auto.php">Main</a></div></td>
<td><div align="center"><a href="line.php">Add Line</a></div></td>
<td><div align="center"><a href="sensor2.php">View Data</a></div></td>
<td><div align="center"><a href="report.php">View Report</a></div></td>
</tr>
</table></div>
<"div class="article>
</div>
<div class="article"><br>
<table>
<"tr><td width="500" style="font-size:18px>
<marquee  bgcolor="#FF0000" s>
<?php
echo "<br>";
echo $line1;
echo "<br>";
echo $line2;
echo "<br>";
echo  $NoConsumptions1;
echo "<br>";
echo $qtwo;
echo "<br>";
echo $one;
```

```
echo "<br>";
echo   $NoConsumptions2;
echo "<br>";
echo $prmain;
?>
</marquee>
</td>
<"td width="810" align="center>
<"table width="785" border="10>
<"tr bgcolor="#CC0033>
<td width="119"><div align="center">LINE NAME </div></td>
<"td width="127"><div align="center>
PRESSURE</div></td>
<td width="79"><div align="center">FLOW</div></td>
<td width="127"><div align="center">VALVE</div></td>
<td width="139"><div align="center">DATE AND TIME </div></td>
tr><font size="+2"><?php</
mysql_connect("localhost","root","root");
mysql_select_db("wncam");
*li=mysql_query(" SELECT$
FROM dataread, line
WHERE line.li_id = dataread.li_id
ORDER BY dataread.dr_date  DESC
LIMIT 0 , 9 ");
){if($li
)){while($row=mysql_fetch_array($li
?>
<tr>
<td>
<input type="text" name="textfield" value="<?php echo $row[li_name]?>"  style="font-
size:16px"/>
</td>
<td><input type="text" name="textfield" value="<?php echo $row[dr_presssure ? >"
<td/>
```

```
<?>"/td><input type="text" name="textfield" value="<?php echo $row[flow>
</td>
<]?>"/td><input type="text" name="textfield" value="<?php echo $row[valve>
</td>
<]?>/td><input type="text" name="textfield" value="<?php echo $row[dr_date>
</td>
}tr><?php</
>}?></font
</ table><br/>
</td></tr></table></td></tr></table></div>
<"div id="footer>
<p>&copy; 2017 Sudan University Of Science And Technology </p>
</div>
</center>
</body>
</html>
```