



جامعة بوليتكنك فلسطين

Palestine Polytechnic University

College of Engineering

Electrical Engineering Department

Detection and localization of faults in power distribution lines

Prepared by :

SherinJabari

SabreenQwasmeh

Rami Salah

Supervisor

Eng. AymanWazwaz

May , 2014

الى من هداانا لهذا وما كنا لنهتدي لولا هدااه
الى من وفقنا لهذا وما توفيقنا الا به
الى المولى العظيم جل في علاه
الى من اصطفاه الله على العالمين
الى فلسطين ارضا وشعبا وحجرا مقاوم
الى الذين زرعوا بذره الهندسة فكان الثمر
ثلاثه مهندسين ابتكروا الطريق الى القلوب
الى النجوم المنيره لنا سبيلنا والتي ما فتنت
تتحرق لتظل منيرة ومضيئة كسراج منير
الى امهاتنا ادامهن الله تاجا على رؤوسنا
الى السند الذي شد الوثاق به

الى من كان ابا واخا الى الاستاذ المشرف ايمن وزوز والاستاذ سامي السلمين والاستاذ مكاوي حريز

الذي ما بخل علينا بثانيه من وقته وعلمه

الى جامعة بوليتكنك فلسطين لا سيما عمادة كلية الهندسة

ودائرة الهندسة الكهربائيه

الى الاخوة الصادقين الرانعين المخلصين

الى من كانت معهم أجمل لحظات حياتنا

الى طلاب كلية الهندسة دفعة ٢٠٠٩

لاخوتنا وأحبتنا والقاطنين قلوبنا

Acknowledgments

We could not forget our families, who stood by us with their support ,love and care for our whole lives; they were with us with there bodies and souls, believed in us and helped us to accomplish this project.

We would like to thank our amazing teachers at Palestine polytechnic university, to whom we would carry our gratitude our whole life. Special thanks to our supervisors eng ayman wazwaz .

Abstract

Our project is **Detection and localization of faults in power distribution lines** which aims to transmit the value of fault and status of the environment readings (humidity, water leakage, heat) through a microcontroller unit and a modem (GSM\GPRS modem) and communicate with the main server in Electricity Company headquarters, and design a program which has a representation (map) of the network covered by the electric power company, give a notification to the server and communicate automatically with the employee in charge of the site to correct the fault in the specified location.

ملخص المشروع

هذا المشروع "كشف الأعطال وتحديد مكانها في خطوط توزيع الكهرباء" يتعلق بتحديد مواقع الأعطال الكهربائية على شبكات الجهد المتوسط من خلال تركيب محولات التيار على قواطع المحطات الفرعية أو أعمدة الضغط العالي بهدف لإرسال قيمة العطل الكهربائي وحالة البيئة المحيطة بما يشمل الحرارة والرطوبة وتسرب المياه إلى الخطوط الكهربائية بحيث يتم الاتصال بالخادم الرئيسي ويتم تحديد العطل عن طريق برنامج يوجد عليه تمثيل للشبكة الكهربائية وإعطاء إنذار نصي من خلال تغير حالة التمثيل للون المفتاح على مخطط الشبكة الأحادية .

Table of contents

Chapter One : Introduction	1
1.1 Overview	2
1.2 Objectives	2
1.3 project motivation	2
1.4 project idea and approach	2
1.5 Literature Review	5
1.6 Estimated cost	8
1.7 Time schedule.....	9
Chapter Two :	10
2.1 Introduction	11
2.2 Current Transformer (CT)	11
2.3 Inductor	11
2.4 Capacitor	13
2.5 Resistance	14
2.6 Humidity- temp Sensor.....	15
2.7 Grove- gas sensor	16
2.8 Microcontroller	16
2.9 Global system for mobile communication (GSM).....	19
2.10 Data base	23
2.11 Java	23
2.12 Tiny RTC 12C.....	24
2.13 ASB.NET.....	25
Chapter Three	26
3.1 Introduction	27
3.2 General Block diagram	27
3.3 system function	27

3.4 system operation	27
3.5 Arduino Microcontroller	29
3.6 GSM Modem	29
3.7 Database	31
3.8 Software Design	33
3.9 Failure Detection	34
3.10 Environmental sensing	37
Chapter Four	39
4.1 Introduction	40
4.2 Software Design	40
4.2.1 Office software	40
4.3 Fault Detection System	53
4.4 Environmental Circuit	55
4.4.1 Grove-gas sensor	55
4.4.2 DHT11	56
4.5 Multiple serial communication	58
4.6 Tiny RTC.....	59
4.7 GPRS\GSM	64
4.8 Message format	68
4.9 Web server.....	68
Chapter five.....	70
5.1 Introduction.....	71
5.2 System testing	71
5.2.1 DHT11.....	71
5.2.2 MQ_2.....	72
5.2.3 RTC.....	72
5.2.4 AC to DC circuit converter	73
5.2.5 Arduino measurement voltage	73
5.2.6 Multiple serial communication	74
5.2.7 SMS communication	74
5.2.8 GPRS communication.....	76
5.3 Testing results.....	78

Table of Figures

Figure 1.1 : schemetic digram of system .	3
Figure 1.2 : General block diagram of system .	4
Figure 2.1. : schemetic digram forcurrent transformer	11
Figure 2.2 : schemetic digram for inductor .	13
Figure 2.3: schemetic digram for capictor .	14
Figure 2.4:schemetic digram for resistor .	15
Figure 2.5:schemetic digram for DHT11.	16
Figure 2.6: schemetic digram for MQ-2.	16
Figure 2.7: schemetic digram for orduino.	19
Figure 2.8: schemetic digram for sim card.	22
Figure 2.9:schemetic digram for GPRS\GSM modem.	23
Figure 2.10: schemetic digram for Tiny RTC	24
Figure 2.11: schemetic digram for Tiny RTC with orduino .	25
Figure 3.1.: flowchart of the system operation	28
Figure 3.2 : A block diagram of the microcontroller at node.	29
Figure 3.3 : A block diagram of GSM modems connection	30
Figure 3.4: component of the cellular shield.	30
Figure 3.5:circuit diagram for fault detection .	37
Figure 3.6:A flowchart of the system operation for fault detection .	34
Figure 3.7: General Block diagram of environmental sensing .	35
Figure 3.8 : flowchart of the system operation of sensing environmental.	36
Figure 3.9 : a block diagram for main part of program .	37
Figure 3.10: a block diagram for main part for web server .	38
Figure 4.1: setup frame of program.	40
Figure 4.2: select the path of alarm sound.	41
Figure 4.3: select the path of alarm sound continue.	41

Figure 4.4: select the path of data base.	42
Figure 4.5:check that the port is available.	42
Figure 4.6:login frame.	43
Figure 4.7:map frame.	44
Figure 4.8:frame for monitoring each node online.	44
Figure 4.9:node frame to add a node .	45
Figure 4.10 :node frame for modifying the information of the node.	45
Figure 4.11:node frame to delete a node.	46
Figure 4.12 : alarm frame to add an alarm.	47
Figure 4.13: alarm frame to modify an alarm.	47
Figure 4.14: alarm frame to delete an alarm.	48
Figure 4.15: user frame to delete user.	49
Figure 4.16: user frame to modify information of a user .	49
Figure 4.17:user frame to set authentication for a user .	50
Figure 4.18:reading frame represent the node which has not connect .	51
Figure 4.19: Reading frame to get the reading of alarm .	51
Figure 4.20:reading frame to get all readings.	52
Figure 4.21:draw the relations between the current and time.	52
Figure 4.22: logout icon.	53
Figure 4.23:the frame after the logout .	53
Figure 4.24: signal to current transformer .	54
Figure 4.25: conversion from Ac to D C.	54
Figure 4.26: The circuit connection of Grove-Gas sensor (MQ2) with Arduino.	55
Figure 4.27: circuit connection between DHT11 sensor with Arduino.	56
	59

Figure 4.28: block diagram of multiple serial communication .	59
Figure 4.29: Tiny RTC with Arduino.	59
Figure 4.30: schemetic digram diagram of connection between RTC ana arduino mega.	60
Figure 4.31: message format.	68
Figure 4.32: print screen of web server.	69
Figure 5.1: the result of DHT11 .	71
Figure 5.2: the result of smoke detector .	72
Figure 5.3: the result of RTC.	73
Figure 5.4: the result of Arduino measurement voltage .	73
Figure 5.5: the result of Arduino multiple serial .	74
Figure 5.6:send SMS of Arduino to employee .	75
Figure 5.7: received SMS on mobile .	76
Figure 5.8:send GPRS MSG of arduino to public IP .	77
Figure 5.9: received GPRS msg in socket server.	78
Figure 5.10:SMS message alarm.	79
Figure 5.11: GPRS MSG receiving on server.	80
Figure 5.12: representation danger state.	80
Figure 5.13: insert reading in data base .	81

Table for Tables

Table 1.1 Estimated Cost	8
Table 1.2 Time planning	9
Table 3.1 log file to all IP Nodes	31
Table 3.2 log file to all Nodes	32
Table 3.3 log file to all alarm type	32
Table 3.4 log file to all reading	32
Table 3.5 log file to all user	35

Chapter One

Introduction

1.1 Overview

1.2 Objectives

1.3 project motivation

1.4 project idea and approach

1.5 Literature Review

1.6 Estimated cost

1.7 Time schedule

1.1 Overview

The electrical company faces a lot of problems in identifying the specific place of faults in electrical lines however ,by cutting off electrical lines in many location. And that will be a waste of time , efforts, and cause a problem for customers . Our system will solve all this problems especially in the companies , which haven't the SCADA system in all nodes . Because the SCADA is expensive, we need another system with less cost .

1.2 Objectives

- Transmit the value of current through microcontroller unit and modem (GSM\GPRS modem).
- Communicate with the main server in Electricity Company headquarters.
- Communicate automatically with the employee in charge of the site to correct the fault in the location .
- Send status of the environmental and communicate with stakeholders e.g. fire , humidity , water leakage ,heat .

1.3 project motivation

- Saving time , instead of cutting off the electrical in the all location, they can use our system to identify the site of the fault.
- Our system includes a map will determine the specific location of fault , that leading to save effort .
- Avoiding complications in the network.
- Our system is cheaper than other fault system detection like SCADA system .
- We can develop the system in Anti-theft of electricity.

1.4 Project idea and approach

As previously mentioned , our project is the monitoring of electrical line nodes , which can detect and identify the electrical fault location .

The monitoring system in our project identify electrical fault sites through the change of value of voltage , and adding current transformers on sites in sub-stations or high-pressure poles as shown in figure 1.1. Then transmitted the value of fault through microcontroller unit connect with GSM/GPRS modem and communicate

with GSM network using GPRS technology with the PC main server in Electricity Company workers As shown in figure 1.2 .

At the company side , the PC server is that receive reading and measurement from electrical line nodes from public ip . A programmed database at the server will store the received reading for each nodes . it will also store the time and data for each reading and gives the needed calculations and information of electrical fault location . after that , the electricity will be sent SMS using GSM network for employee charge of the site.

Additionally the program which have the map is responsible to status received from GSM modem in electrical fault site , then the program analysis status and give a text warning and changes the status of representation of the key color on the scheme electrical grid map.

additionally , the system send status of environmental (fire , humidity , water leakage ,heat)for each nodes by built sensor circuit and communicate with stakeholders by sending SMS automatically using GSM network .

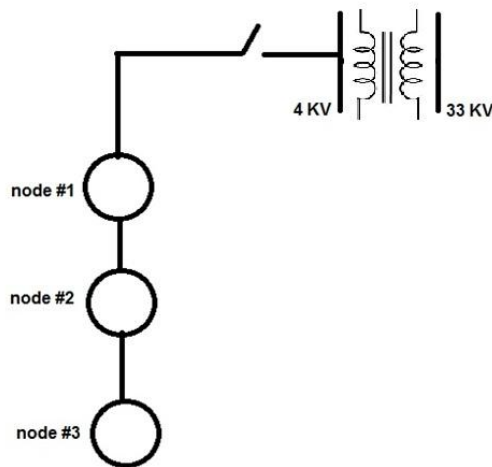


Figure 1.1 : schematic digram of system

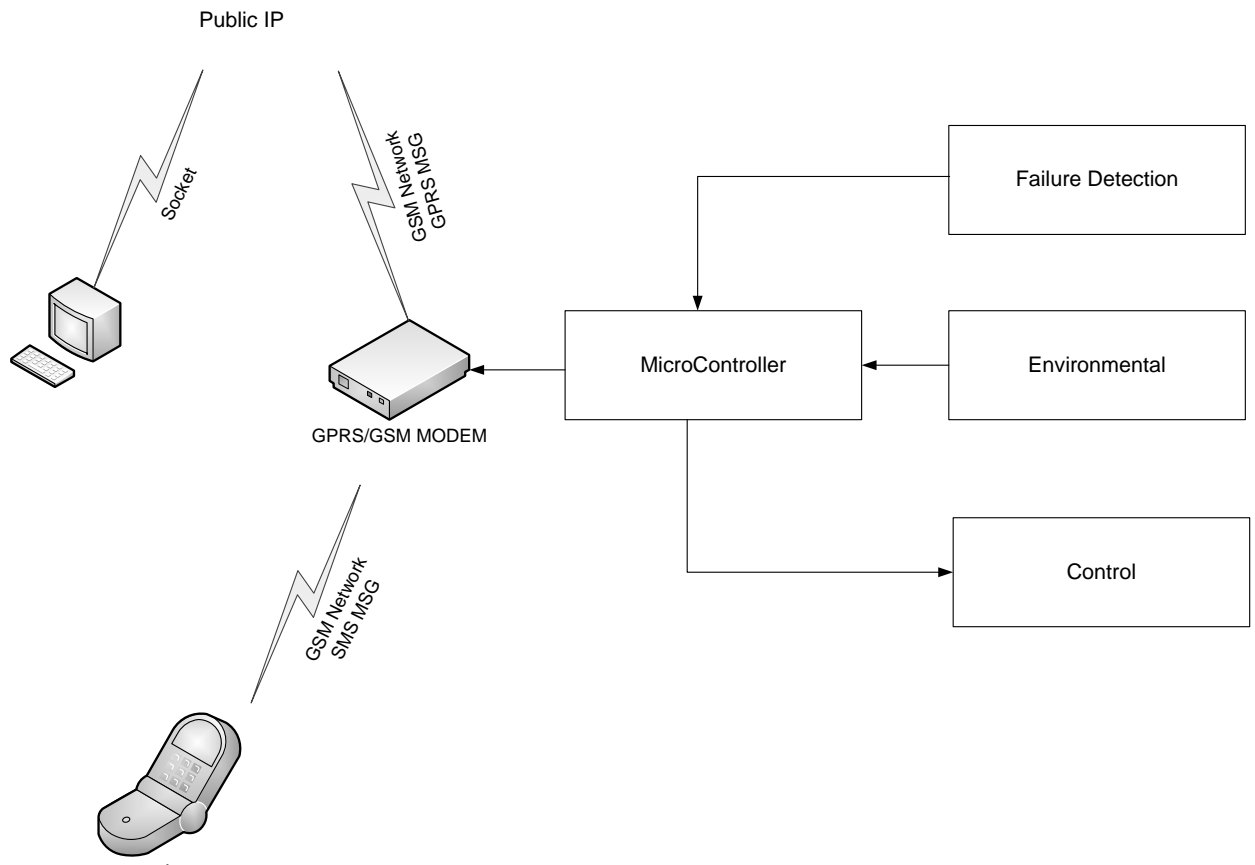


Figure 1.2 : General block diagram of system

1.5 Literature Review

In a project with the title Design and implementation of secure low cost AMR system using GPRS technology [1], the authors presented a design and implementation of a secure low cost AMR (automatic meters reading) system using GPRS technology. It's a bidirectional system that transmits and receives data from and to the electricity company. They designed a digital meter which is constructed using components such as evaluation board(MCP3905A), a microcontroller (PIC16F84A), and LCD(2 lines with 16 characters per line). The power can be read by sensing the load that is connected to the evaluation board. The PIC microcontroller is used to accumulate the consumed energy by saving the readings in the memory of the microcontroller ; readings will be module and Quad-band wired cellular antenna SMA.

The meter reading is sent automatically to the server and customers can remotely get their consumption at any time.

In our project, we will not design a new electricity meter that is capable with GPRS technology. We use the same meters already used by customers, but we add additional equipment's to it. Additionally the microcontroller that will be used is mbed microcontroller because it is easier to be programmed and to be connected to the pc's.

In a study about tropes[2], the author proposed automatic meters reading devices that have embedded Wi-Fi capability operating in the 2.4 GHZ unlicensed spectrum.

The design uses smart meters that record consumption of electric energy every hour or less and communicate the information at least daily with the company in two way communication.

It offers a reliable and secure data transfer from all parts of the city to the center office using 802.11 security standards.

In comparison to our system, it is based on GPRS technology and we intend use a Zigbee technology to have a clustering structure to reduce the data traffic .Besides ,the data will be transmitted when requested not daily. Wi-Fi technology will face several problems such that its limited coverage and interference from microwaves or cordless phones which use the same frequency aWi-Fi.

In SCADA system (supervisory control and data acquisition)[3], is a type of industrial control system (ICS). Industrial control systems are computer controlled systems that monitor and control industrial processes that exist in the physical world. SCADA systems historically distinguish themselves from other ICS systems by being large scale processes that can include multiple sites, and large distances. the SCADA system has advantages like : Real-time reporting of the physical variables of the system, Ease of maintenance ,Control of contingencies ,Reliability of measurement, Efficiency in the operation, Storage of historical data. And has disadvantages like : TCP/IP based SCADA systems are (extremely),vulnerable to cyber warfare /cyber terrorism attacks which, in the worst ,scenario case, could cause not only financial loss but also loss of life, directly or indirectly, Requires significant oversight by Owner to ensure compliance with contract specifications, needs high costs.

From the scientific paper named review of fault location techniques for distribution system [4],the author discussed methods , proposed in literature or implemented in practice, for estimating the location of transmission line faults consist of using voltages and currents measured at one or both terminals of a line. The methods can be divided into three categories: methods that are based on traveling waves, methods that use high frequency components of currents and voltages and, methods that use the fundamental frequency voltages and currents measured at the terminals of a line. The last method, also classified as impedance-based method, consists of calculating line impedances as seen from the line terminals and estimating distances of the faults.

Fault locators provide estimate for both sustained and transient faults. Generally, transient faults cause minor damage that is not easily visible on inspection. Fault locators help identify those locations for early repairs to prevent recurrence and consequent major damages.

From the scientific paper with the article Fault Detection of Electric Power Transmission Line by Using Neural Network[5],the authors discussed the use of Neural Network to detect faults, The major advantage of the neural networks is its self-learning capability. First, the network is presented with a set of correct input and output values. Then it adjusts the connection strength among the internal network nodes until proper transformation is learned. Second the network is presented with only the input data, and then it produces a set of output values.

Power systems are prone to frequent faults, which may occur in any of its components, such as generating units, transformers, transmission network and/or loads. It is well

known that faults can cause significant disruption of supply, so using of the neural network will solve this problem.

1.6 Estimated cost

The table below show the cost of the project for one Node .

Type	Unit Price(NIS)	Amount	Total price(NIS)
GSM/GPRS MODEM.	920	1	920
MICROCONTROLLER (Arduino mega).	350	1	350
MICROCONTROLLER (Arduino atmega).	60	4	240
sensors and electrical components	550	1	550
Movement and transportation	1320	-	1320
Stationery and printing supplies	400	-	400

Table 1.1 : estimated cost

TOTAL COST = 3780 NIS \approx 1080 \$; (1 \$=3.5)

1.7 Time schedule

The table below show the activities and efforts done on the project and the time associated with each one .

Table 1.2 : Time planning

Tasks \ Week	Week															
	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
writing project software	█	█	█	█	█	█	█	█								
implementation hardware	█	█	█	█	█	█	█									
unit testing					█	█	█	█	█							
system testing									█	█	█	█	█	█	█	
maintenance and development									█	█	█	█	█	█	█	
documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
project delivery															█	

2

Chapter Two

Introduction

2.1 Introduction

2.2 Current Transformer (CT)

2.3 inductor

2.4 capacitor

2.5 resistance

2.6 Humidity-Temp Sensor

2.7 Grove - Gas Sensor

2.8 Microcontroller

2.9 Global system for mobile communication (GSM)

2.10 Database

2.11 JAVA

2.12 Tiny RTC I2C

2.13 ASB. NET

2.1 Introduction

In the following sections these option are listed , showing the advantages and disadvantages of each component used in this system , explanation of its characteristics ,uses and type.

2.2 Current Transformer (CT)

is used for measurement of alternating electric currents. Current transformers, together with voltage transformers (VT) (potential transformers (PT)), are known as instrument transformers. When current in a circuit is too high to directly apply to measuring instruments, a current transformer produces a reduced current accurately proportional to the current in the circuit, which can be conveniently connected to measuring and recording instruments.[6]

Like any other transformer, a current transformer has a primary winding, a magnetic core and a secondary winding. The alternating current flowing in the primary produces an alternating magnetic field in the core, which then induces an alternating current in the secondary winding circuit.

Current transformers are used extensively for measuring current and monitoring the operation of the power grid. Along with voltage leads, revenue-grade CTs drive the electrical utility's watt-hour meter on virtually every building with three-phase service and single-phase services greater than 200 amps .

The CT is typically described by its current ratio from primary to secondary. Often, multiple CTs are installed as a "stack" for various uses. For example, protection devices and revenue metering may use separate CTs to provide isolation between metering and protection circuits, and allows current transformers with different characteristics (accuracy, overload performance) to be used for the devices.



Figure 2.1:schematic diagram for current transformer

2.3 Inductor:

An inductor, also called a coil or reactor, is a passive two-terminal electrical component which resists changes in current passing through it. It consists of a conductor such as a wire, usually wound into a coil. When a current flows through it, energy is stored temporarily in a magnetic field in the coil. When the current flowing through an inductor changes, the time-varying

magnetic field induces a voltage in the conductor, according to Faraday's law of electromagnetic induction, which opposes the change in current that created it.

An inductor is characterized by its inductance, the ratio of the voltage to the rate of change of current, which has units of henneries (H). Inductors have values that typically range from 1 μ H (10–6H) to 1 H. Many inductors have a magnetic core made of iron or ferrite inside the coil, which serves to increase the magnetic field and thus the inductance. Along with capacitors and resistors, inductors are one of the three passive linear circuit elements that make up electric circuits. Inductors are widely used in alternating current (AC) electronic equipment, particularly in radio equipment. They are used to block the flow of AC current while allowing DC to pass; inductors designed for this purpose are called chokes. They are also used in electronic filters to separate signals of different frequencies, and in combination with capacitors to make tuned circuits, used to tune radio and TV receivers.

Inductors are used extensively in analog circuits and signal processing. Inductors in conjunction with capacitors form tuned circuits which can emphasize or filter out specific signal frequencies. Applications range from the use of large inductors in power supplies, which in conjunction with filter capacitors remove residual hums known as the mains hum or other fluctuations from the direct current output, to the small inductance of the ferrite bead or torus installed around a cable to prevent radio frequency interference from being transmitted down the wire. Smaller inductor/capacitor combinations provide tuned circuits used in radio reception and transmission, for instance. Inductors are used as the energy storage device in many switched-mode power supplies to produce DC current. The inductor supplies energy to the circuit to keep current flowing during the "off" switching periods.

Two (or more) inductors in proximity that have coupled magnetic flux (mutual inductance) form a transformer, which is a fundamental component of every electric utility power grid. The efficiency of a transformer may decrease as the frequency increases due to eddy currents in the core material and skin effect on the windings. The size of the core can be decreased at higher frequencies. For this reason, aircraft use 400 hertz alternating current rather than the usual 50 or 60 hertz, allowing a great saving in weight from the use of smaller transformers.

Inductors are also employed in electrical transmission systems, where they are used to limit switching currents and fault currents. In this field, they are more commonly referred to as reactors.

Because inductors have complicated side effects (detailed below) which cause them to depart from ideal behavior, because they can radiate electromagnetic interference (EMI), and most of all because of their bulk which prevents them from being integrated on semiconductor chips, the use of inductors is declining in modern electronic devices, particularly compact portable devices. Real inductors are increasingly being replaced by active circuits such as the gyrator which can synthesize inductance using capacitors.



Figure 2.2:schematic diagram for Inductor

2.4 capacitor :

A capacitor (originally known as a condenser) is a passive two-terminal electrical component used to store energy electrostatically in an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors separated by a dielectric (insulator); for example, one common construction consists of metal foils separated by a thin layer of insulating film. Capacitors are widely used as parts of electrical in many common electrical devices.[7]

When there is a potential difference across the conductors, an electric field develops across the dielectric, causing positive charge to collect on one plate and negative charge on the other plate. Energy is stored in the electrostatic field. An ideal capacitor is characterized by a single constant value, capacitance. This is the ratio of the electric charge on each conductor to the potential difference between them. The SI unit of capacitance is the farad, which is equal to one coulomb per volt.

The capacitance is greatest when there is a narrow separation between large areas of conductor, hence capacitor conductors are often called plates, referring to an early means of construction. In practice, the dielectric between the plates passes a small amount of leakage current and also has

an electric field strength limit, the breakdown voltage. The conductors and leads introduce an undesired inductance and resistance.

Capacitors are widely used in electronic circuits for blocking direct current while allowing alternating current to pass. In analog filter networks, they smooth the output of power supplies. In resonant circuits they tune radios to particular frequencies. In electric power transmission systems they stabilize voltage and power flow.

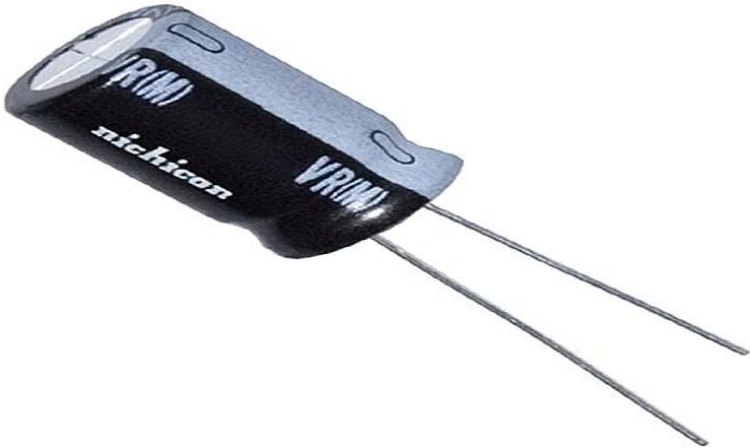


Figure 2.3:schematic diagram for Capacitor

2.5 Resistance

The electrical resistance of a circuit component or device is defined as the ratio of the voltage applied to the electric current which flows through it:

$$R = V/I$$

If the resistance is constant over a considerable range of voltage, then Ohm's law, $I = V/R$, can be used to predict the behavior of the material. Although the definition above involves DC current and voltage, the same definition holds for the AC application of resistors.

Whether or not a material obeys Ohm's law, its resistance can be described in terms of its bulk resistivity. The resistivity, and thus the resistance, is temperature dependent. Over sizable ranges of temperature, this temperature dependence can be predicted from a temperature coefficient of resistance. The electrical resistance of a wire would be expected to be greater for a longer wire, less for a wire of larger cross sectional area, and would be expected to depend upon the material out of which the wire is made. Experimentally, the dependence upon these properties is a straightforward one for a wide range of conditions, and the resistance of a wire can be expressed as :

$$R = \frac{\rho L}{A}$$

ρ = resistivity
 L = length
 A = cross sectional area

The factor in the resistance which takes into account the nature of the material is the resistivity . Although it is temperature dependent, it can be used at a given temperature to calculate the resistance of a wire of given geometry[10].



Figure 2.4:schematic diagram for Resistor

2.6 DHT11-Humidity-TempSensor

Digital output temperature and humidity sensor DHT11

Needs 4.7K to 10K pull-up resistor from +5 to Signal pin. Connect to Digital pin, not Analog pin.

- Full range of calibration, in-line digital output;
- Humidity measuring range: 20% ~ 90% RH (0-50 °C temperature compensation)
- Temperature measuring range: 0 ~ +50 °C ;
- Humidity measuring accuracy: 5.0% RH
- Temperature measurement accuracy: 2.0 °C
- Response time: <5S ;
- Low power consumption

Features :

1. single wire digital interface (the most simple system integration, ultra-low prices).
2. ultra-small size (12X15.5X5.5 mm).
3. high reliability.
4. optimized long-term stability.

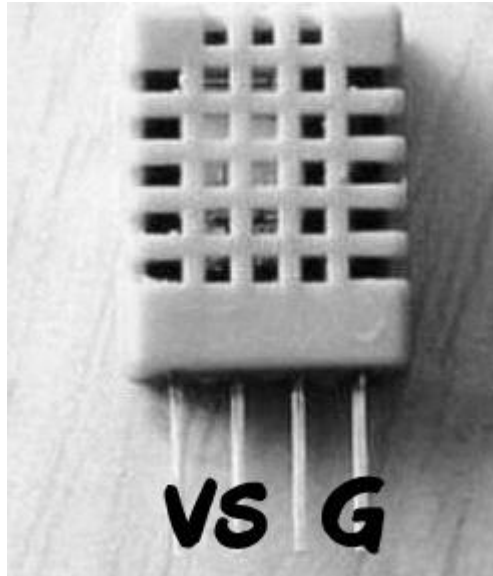


Figure 2.5:schematic diagram for DHT11

2.7 Grove - Gas Sensor(MQ2)

The Grove - Gas Sensor(MQ2) module is useful for gas leakage detecting(in home and industry). It can detect LPG, i-butane, methane, alcohol, Hydrogen, smoke and so on. Based on its fast response time. measurements can be taken as soon as possible. Also the sensitivity can be adjusted by the potentiometer.



Figure 2.6:schematic diagram for MQ-2

2.8 Microcontroller

Microcontroller is a highly integrated chip that contains all the components comprising a controller. Typically this includes a CPU, RAM, some form of ROM, I/O ports, and timers.

Another definition for microcontroller is" a small computer on a single integrated circuit containing a processor core ,memory ,and programmable input/output peripherals".

As microcontroller is an integrated circuit, the cost of the total system decreases, a smaller and cheaper circuit board used, the effort required to assemble and test the circuit board reduces, and the number of chips and the amount of wiring reduces.

Microcontrollers are designed for using in embedded systems, which mean that they are part of embedded systems, so they are sometimes called "embedded microcontrollers".

A microcontroller is designed for A very specific task to control a particular system and is used in automatically controlled products and devices, such as automobile engine control systems ,implantable medical devices ,remote controls ,office machines ,appliances ,power tools ,and toys .By reducing the size and cost compared to a design that uses a separate microprocessor ,memory, and input/output devices ,microcontrollers make it economical to digitally control even more devices and processes.

Most microcontrollers deal with a digital data, so analog-to-digital converter(ADC)must be exist to convert analog data to digital, but in some microcontrollers there is a digital-to-analog converter(DAC)that allows the processor to output analog signals or voltage levels.

The microcontroller used in this system, is The Arduino microcontroller . Arduino was the first hardware used to bootstrap the Arduino platform, used for creating smart devices based on ARM Cortex-M microcontrollers.

Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible.[1]The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. Current models feature a USB interface, 6 analog input pins, as well as 14 digital I/O pins which allows the user to attach various extension boards.

Introduced in 2005, it was designed to give students an inexpensive and easy way to program interactive objects. It comes with a simple integrated development environment (IDE) that runs on regular personal computers and allows to write programs for Arduino using C or 2B%2B_programming_language"C++.

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected 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 2%B2C"I²C serial bus, allowing many shields to be stacked and used in parallel. Official Arduino have used the mega AVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other

processors have been used by Arduino compatibles. Most boards include a 5 volt linear [HYPERLINK "http://en.wikipedia.org/wiki/Linear_regulator"](http://en.wikipedia.org/wiki/Linear_regulator) regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino 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.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a level shifter circuit to convert between RS-232-level and [transistor_logic](#)"TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FT232RL. Some variants, such as the Arduino Mini and the unofficial Boarding, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs. These pins are on the top of the board, via female 0.10-inch (2.5 mm) headers. Several plug-in application shields are also commercially available.

The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board to be plugged into solderless breadboards.

There are many Arduino-compatible and Arduino-derived boards. Some are functionally equivalent to an Arduino and may be used interchangeably. Many are the basic Arduino with the addition of commonplace output drivers, often for use in school-level education to simplify the construction of buggies and small robots. Others are electrically equivalent but change the form factor, sometimes permitting the continued use of Shields, sometimes not. Some variants use completely different processors, with varying levels of compatibility.

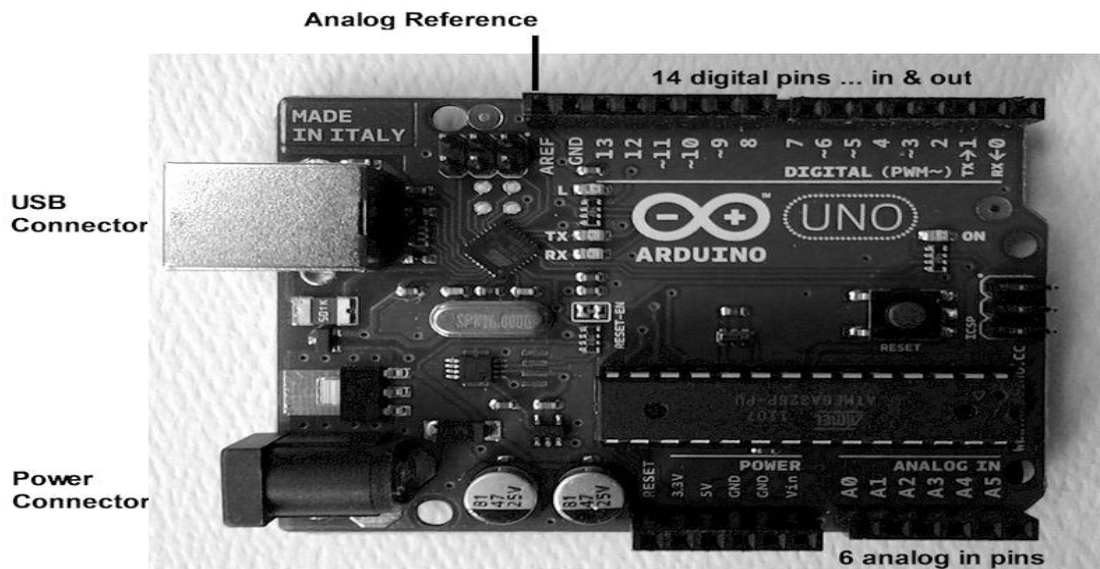


Figure 2.7:schematic diagram for Arduino

2.9 Global system for mobile communication (GSM)

2.9.1 Foundation and definitions

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services.

it is one of several type of cellular technology , and is the second generation (2G) of the mobile communication network . GSM allows for voice and data transmission across several frequencies, and support cellular devices from basic to smart phone . one of the most popular cellular services , GSM services is available across almost 80 percent of the word .

GSM when used in the standard operation mode was able to provide data transfer speed up to 9.6 Kbps. Through the year a new technique was introduced in the GSM standard called high speed circuit switched data (HACSD) . this technology makes it possible to use several time slots simultaneously when sending or receiving data , so the user can increase the data transmission to 14.4 Kbps (an increase of 50%) or even triple 43.3 Kbps the transmission speed .

the GSM network is divided into three major systems :[9]

- The switching system (SS): which is responsible for performing call processing and subscriber-related functions.
- The Base Station System (BSS): All radio-related functions are performed in the BSS.
- The Operation and Support System (OSS): it is the functional entity from which the network operator monitors and control the system .

GSM network Areas :

The GSM network is made up geographic areas . these areas include cells ,location areas , mobile switching center / visitor location register (MSC/VLR) service areas , and public land mobile network (PLMN areas).

The cell is the area given radio coverage by one base transceiver station . the location area is a group of cells . it is the area in which the subscriber is paged .

An MSC/VLR services area represents the part of the GSM network that is covered by one MSC and which is reachable , as it is registered in the VLR of the MSC .

The PLMN services area is an area served by one network operator .

2.9.2 Why have we used GSM network?

1) Security:

GSM services are highly secure, with technologies in place that can protect against both eavesdropping and service hacking. The SIM card or subscriber identity module card which carries subscriber and contact information ,secures customer information.

2) Affordable devices and services:

GSM providers control a large share of the cellular market and therefore are able to provide a large variety of affordable device and services.

3) Extensive coverage:

GSM has widespread use throughout the world. So even though different countries may GSM operate on different frequency band, user can transfer seamlessly between networks and keep the same number .as a result GSM user essentially have coverage in over 218 countries.

4) Greater phone variety:

Another advantage of GSM is that because it is used throughout the world , there is greater variety of phone that operate on GSM ,therefore consumers have more flexibility in choosing a handset that fits their specific desires ,and they are not limited to purchasing phone only made in their respective country.

There are other advantages for GSM such as ,ability to use repeaters , less signal deterioration inside building , and maturity of GSM which mean a more stable network robust features .

2.9.3 Short messaging service (SMS)

In this system , SMS services will be used at the server's PC ; to send the alarm to the employee charge of the site to correct the fault in the location .

SMS is way of sending short messages to mobile telephones "short" mean a maximum of 160 bytes. According to GSM association "each short message is up to 160 characters in length when Latin alphabets are used and 70 characters in length when non Latin alphabets such as Arabic and Chinese are used.

Our motivation to use mobile messaging service is that not only does it provide an alternative mean of transmission in a cellular communication system but it is a more versatile and convenient option since all new phone are SMS and capable. Additionally ,under certain circumstances, SMS can be the only mean of transmission in a cellular system .This can be evident when considering the vulnerable nature of the traditional wireless voice channel used by a cellular phone to establish a connection with the serving base station including dropped calls and service denials during peak hours. SMS uses different sets of channels that are more robust than those assigned for voice ,which can enable a user to send and receive SMS message at time when s/he cannot get access to the network for voice calls .These is because SMS uses control channel of a cellular system which are used setup rather than the regular voice channel.

2.9.4 General packet radio service (GPRS)

in this system , the wireless communication between the server at the electricity company and the microcontroller in the nodes that are distributed over a region , will be based on General packet radio service which is second generation (2G) and third generation (3G) – or sometimes referred to as between both generations (2.5G) – wireless data service . the GPRS system is an integrated part of the GSM network switching subsystem and it extend its data capabilities for accessing internet , multimedia messaging services and some internet applications .

GPRS is a packet oriented mobile data service that applied a packet radio principle to transfer user data packet between GSM mobile station and external packet data networks in an efficient way . in 2G system , GPRS provides data rates of 56 -114 K bit /second .

Our **motivation** to use General packet radio service is that not only does it provide an alternative means of transmission in a cellular communication system , but also it is a more versatile and convenient option since it gives almost continues connection to the internet and there is no need to a physical end to end connection , because network bandwidth is only used when data is actually transferred . usage of the radio bandwidth will be external efficient , which mean higher data rates – in 2 G systems , GPRS provides data rate of 56 -114 Kbit /second -. For example , if SMS over GPRS is used , an SMS transmission speed of about 30 SMS message per minute may be achieved . this is much faster than using the ordinary SMS over GSM , whose SMS transmission speed is about 6 to 10 SMS message per minute . in theory , GPRS packet-based services cost users less than circuit-switched services since communication

channels are being used on a shared use ,as –packet – are needed basis rather than dedicated to only one user at a time . [10].

2.9.5 SIM card :

The Subscriber Identity Module (SIM) card is a wafer-thin, thumbnail-size microchip used by all GSM devices, including phones and GSM-GPRS PC card modems. Like a credit card or smart card, the SIM card securely stores information about the user's account and subscription services, and it can be used to support services such as wireless e-commerce, or m-commerce.

The SIM card is removable, allowing customers to switch GSM devices when purchasing a new phone or adding a PC card – without the hassle of configuring the new device or the loss of personalized subscription services, such as messaging. The SIM card makes it easier for users to change GSM operators and keep the same phones. This flexibility makes GSM-based data networks, such as GPRS and UMTS-HSPA, attractive for a wide variety of data applications.



Figure 2.8:schematic diagram for SIM card

2.9.6 GSM Modem(SM5100B)

The most suitable modem for the proposed system, is the cellular modem(SM5100B)which is associated with a cellular shield on the same board. This board ,shown in figure below, is called "cellular shield with SM5100B". [11]

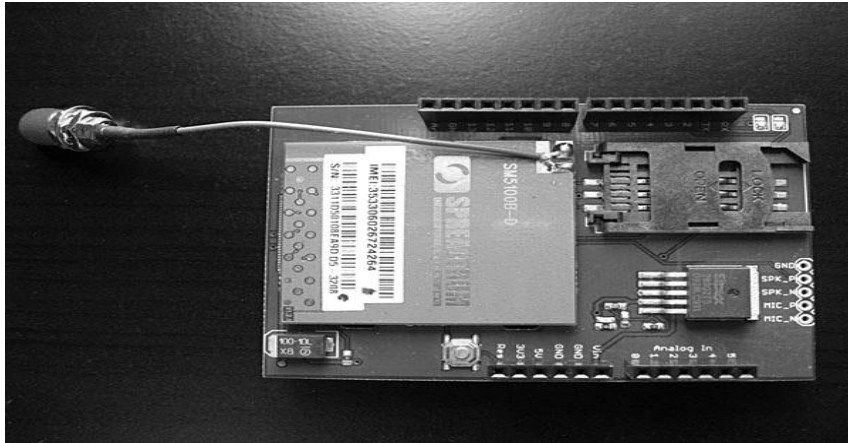


Figure 2.9:schematic diagram for GPRS/GSM modem

2.10 Database

After the value of current and status of each nodes transmitted over GSM network . reaches the server at the electricity company .

A database is collection of the status and alarms for each nodes, data for mapping on PC server screen . to access information from a database , you need a database management system , which is software package with computer programs that control the creation , maintenance ,and the use of a database. It can programmed using different programming languages such as C# ,visual basic ,JAVA etc.

2.11 JAVA

Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems(which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.[12]

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1991 and first released in 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU. Others have also developed alternative implementations of

these Sun technologies, such as the GNU Compiler for Java(byte code compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plug-in for applets).

Why have we used JAVA ?

- simple, object-oriented and familiar.
- robust and secure.
- architecture-neutral and portable.
- high performance.

2.12 Tiny RTC I2C

Tiny RTC I2C module for Arduino. Once it's configured and launched, it allows us to know the current time and day. This module consists of a DS1307 clock , an EEPROM 24C32AN memory and a holder for CR2032 lithium battery.



Figure 2.10:schematic diagram for Tiny RTC

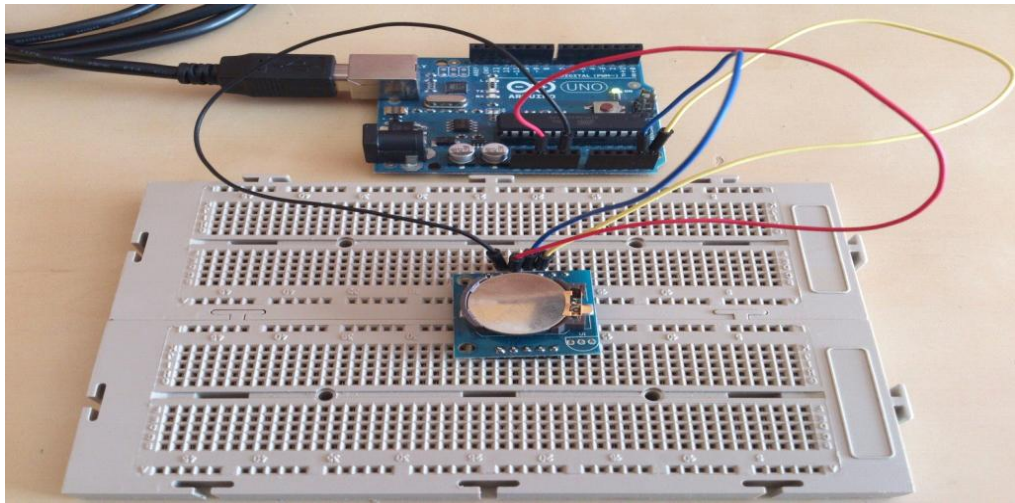


Figure 2.11:schematic diagram for Tiny RTC with Arduino

2.13 ASB. NET

ASP.NET is a server-side Web application framework designed for Web development to produce dynamic Web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

3

Chapter Three

System Design

3.1 Introduction

3.2 General Block Diagram

3.3 system function

3.4 system operation

3.5 Arduino microcontroller

3.6 GSM Modem

3.7 Database

3.8 software design

3.9 failure detection

3.10 Environmental sensing

3.1 introduction

This chapter discusses the system concepts and its different entities. It also describes the subsidiary block diagrams, explaining how each entity is connected to the others. Flowcharts are added to explain the system operation .Additionally, features and detailed schematic diagrams for each component will be shown.

3.2 General Block Diagram

Figure 1.1 and figure 1.2 from chapter one shows the suggested network structure of the whole project.

3.3 system function

As shown in figure 1.2 in chapter 1 Our project is Detection and localization of faults in power distribution lines measures the current in each node and transmits the value of fault and status of environmental (humidity , water leakage ,heat) through microcontroller unit and modem (GSM\GPRS modem) to the main server in Electricity Company headquarters, and in the same time send SMS message to Mobile station for employee charge .

3.4 system operation

Figure 3.1 a presents the flow chart that explain the system operation at different levels .As shown

1. it starts with reading the current line to the current transformer .
- 2 then read current to current voltage converter .
3. read measurement of environmental of case node .
4. make a processing on this signal at Arduino controller .
5. transmit the output of Arduino to GSM /GPRS modem in the node using serial communication.
6. transmitted the reading via GSM/GPRS modems to central office at the electrical company .
7. store data measured in a log file in pc server .
8. represent on the map.
- 9.If any reading above a specific threshold the program will change the color of key on map ,send SMS to the employee in the region via GSM network, at the same time microcontroller in the node will send SMS to the employee .

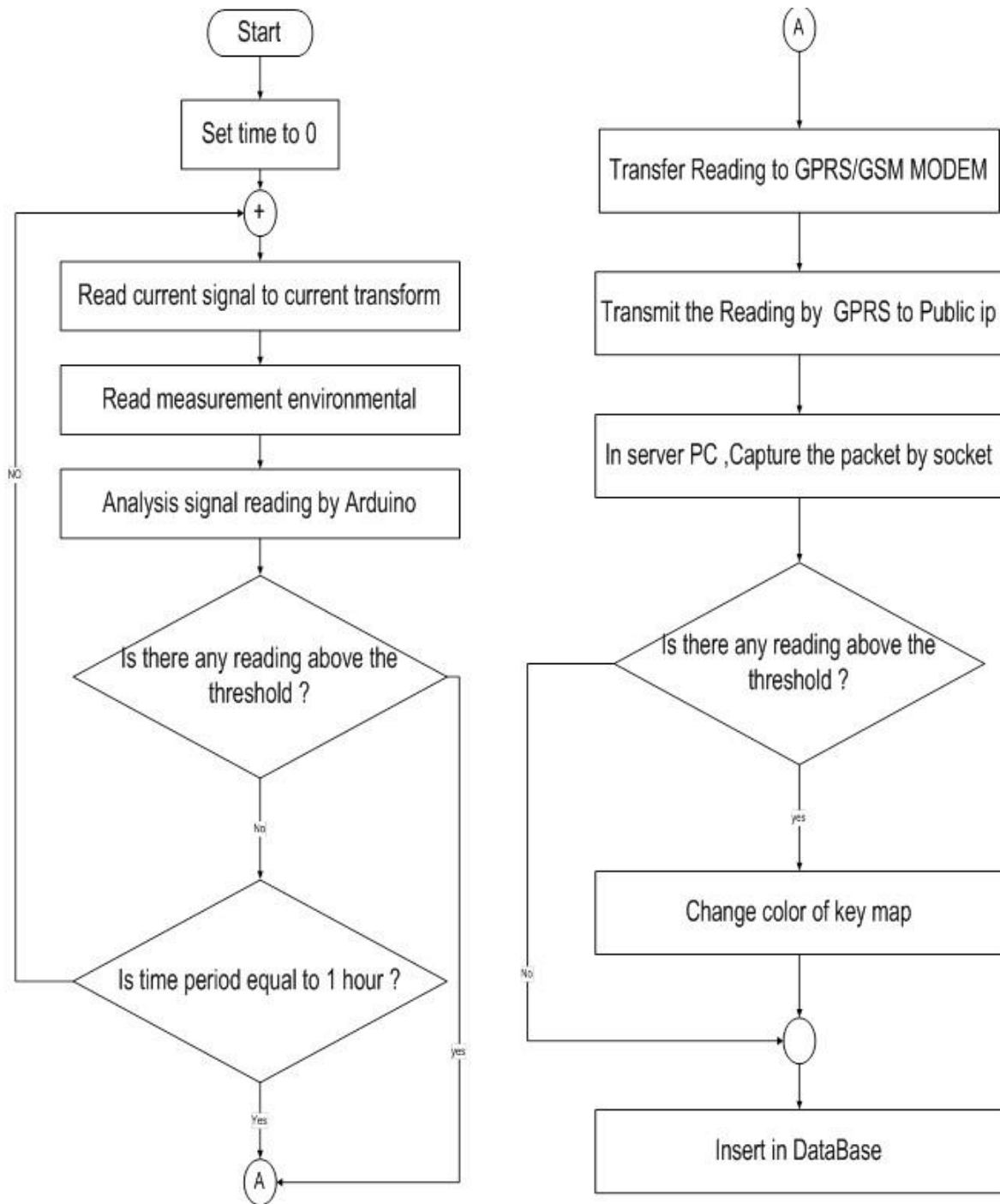


Figure 3.1 A flowchart of the system operation (mentoring system)

3.5 Arduino microcontroller

The whole system is built around microcontroller .Thus we can consider it as the central unit in the project as shown in the figure 3.2 ,the Arduino controller receives the reading of(current ,alarm of humidity ,fire ,water leakage, heat),for each node and processing them ,and output this reading to the GSM/GPRS modem input.

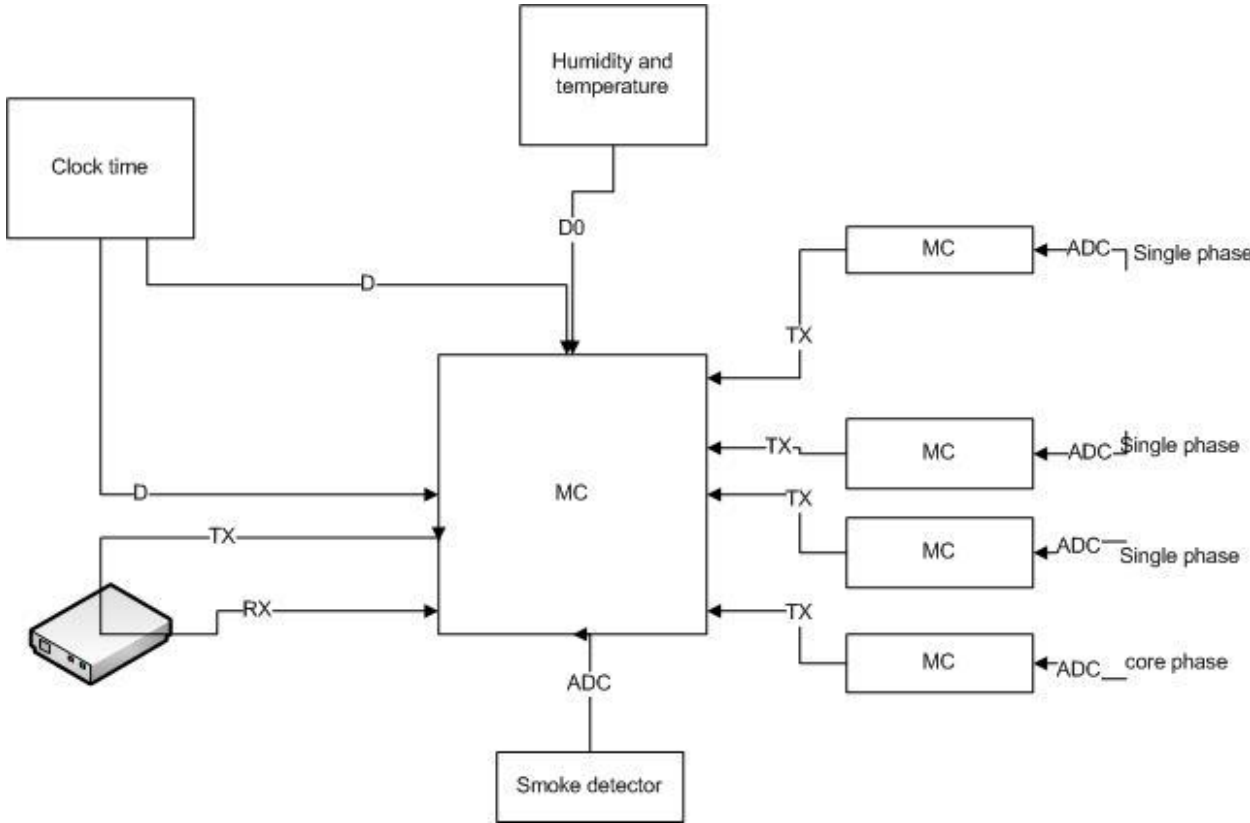


Figure 3.2 : A block diagram of the microcontroller at node

3.6 GSM Modem

As we have seen before , The GSM modem is an essential part of the second communication part, this modem is controlled by the Arduino microcontroller.

Data is sent from GSM modem located at the gateway nodes, to public ip at the monitoring pc in the central office as shown in figure 3.3 . To send data containing the status of environmental and current.

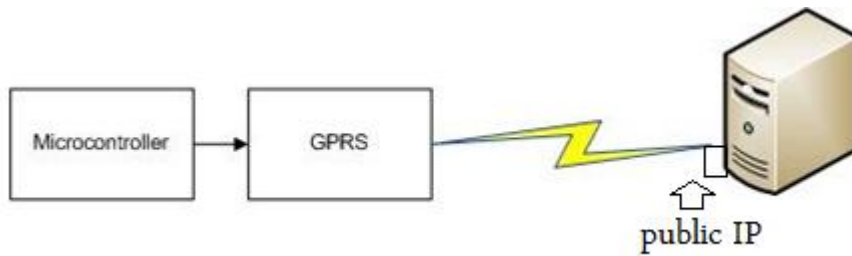


Figure 3.3 : A block diagram of GSM modems connection.

The most appropriate cellular modem we found is "SM5100B", since it exists jointly with a cellular shield of an Arduino microcontroller altogether on the same board a cellular shield with SM5100B.

The advantage of this board stems from including all the parts needed to interface the Arduino with an sm5100B cellular module, and that easily allow GPRS and GSM functionalities to the project.

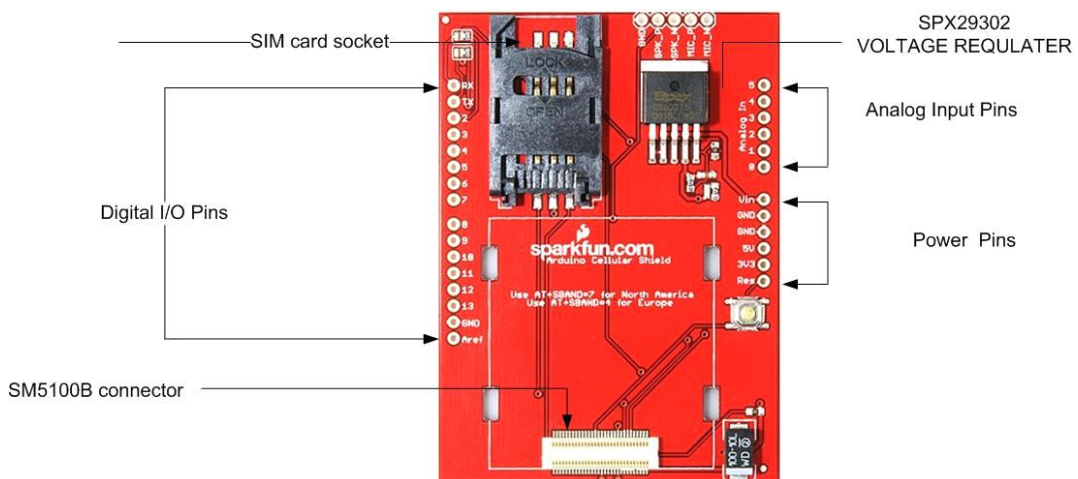


Figure 3.4 : component of the cellular shield

The main component of the cellular shield are 60 pin SM5100B connector a SIM card socket and an SPX29302 voltage regulator configured to regulate the Arduino raw voltage to 3.8 V. the board red LED indicates power .the Arduino reset button is also brought out on the shield .tow jumpers on the board allow selecting which serial pins interface with the cellular module .

On the one hand this board will be wired the microcontroller ,and connected wirelessly with other GSM modem.

TO perform this connection detailed schematic diagrams are needed for each some potent in the cellular board.

The following figures show the required schematic diagrams for the component:

This GSM modem works on -10 Celsius to +55 Celsius temperature range, and 3.3 v to 4.2 v power supply range(3.6v typical).It operates on EGSM900,GSM850,DCS1800,PCS1900 frequency bands and supported with 3v/1.8v SIM card.

3.7 Database:

Once the readings arrive at the server side, it appears a need to a program on the server pc to store and analyze the data. Building this data base and graphical user interface can be implemented by a program called JAVA.

The table 3.1 below show the log file in data base ,store the ip for GPRS node , update every received data .

Table 3.1 : Log file to all IP Node

nodenum	ip
1	/127.0.0.1:55256
2	/127.0.0.1:54834
3	/127.0.0.1:49973
6	/127.0.0.1:64891
8	/127.0.0.1:54645
9	/127.0.0.1:55279
120	/127.0.0.1:56404
121	/127.0.0.1:64567
123	/127.0.0.1:55469
129	/127.0.0.1:50559

The table 3.2 below show the log file in data base for information of all Node .

Table 3.2 : Log file to all Node

NodeNum	NodeName	X	Y	delay6
1	120	23334	3445	196
2	121	23232	23223	286
3	122	2321	121	137
4	wade harieya	80	4	287
5	no	0	0	286
6	no	0	0	286
7	126	34	332	286
8	127	7676	55445	286
9	128	7676	4343	183
10	129	80	433	367
11	no	0	0	65
(جديد)				0

The table 3.3 below show the log file in data base for Alarm Type .

Table 3.3 : Log file to all Alarm Type

ratypealarm	ranumalarm
smoke	11
high current	12
cuurent earth	13
rami	111111
humidity	1
hh	3232
nent one ha3	41
*	

The table 3.4 below show the log file in data base for all Reading received from all node

Table 3.4 : Log file to all Reading

RID	Time222	NumNode	currentvaluephase1	currentvaluephase2	currentvaluephase3	currentvaluecorephase	AlarmType
296	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
295	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
294	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
293	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
292	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
291	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
290	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
289	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
288	11:23:40 23/10/2014	4	1700.35	1333.45	44.33	134.29	
287	11:23:40 23/10/2014	3	1700.35	1333.45	44.33	134.29	
286	11:23:40 23/10/2014	3	1700.35	1333.45	44.33	134.29	
285	11:23:40 23/10/2014	121	1700.35	1333.45	44.33	134.29	
284	11:23:40 23/10/2014	120	1700.35	1333.45	44.33	134.29	
283	11:23:40 23/10/2014	9	1700.35	1333.45	44.33	2134.29	
282	11:23:40 23/10/2014	9	1700.35	1333.45	44.33	134.29	
281	11:23:40 23/10/2014	121	1700.35	1333.45	944.33	134.29	
280	11:23:40 23/10/2014	1	1700.35	1333.45	44.33	134.29	
279	11:23:40 23/10/2014	121	1700.35	1333.45	44.33	134.29	

The table 3.5 below show the log file in data base for all user information .

Table 3.5 : Log file to all User

UID	UName	Permissionn	pas	انقر للإضافة
190	rami	1	22	
192	rami	1	22	
193	rami	1	22	
194	rami	1	22	
195	rami	1	22	
196	rami	1	22	
198	rami	1	22	
199	rami	1	22	
200	rami	1	22	
201	rami	1	22	
202	rami	1	22	
203	rami	1	22	
204	rami	1	22	
205	rami	1	22	
207	rami	1	22	
208	rami	1	22	
209	rami	1	22	
210	rami	1	22	

3.8 software design

* figure 3.9 show the main block diagram of program:

- Socket Server : which is wait for packet on public ip .
- Data Base : store all reading and information .
- graphical user interface and processing : make processing on the reading and make alarm when happen accident .

*figure 3.10 show the main block diagram of web server :

- Data Base : store reading that represented on the web server .
- Page : represent the reading on the page using tool (Grid View).
- Data Source : the method for transfer data between data base and page is data source .

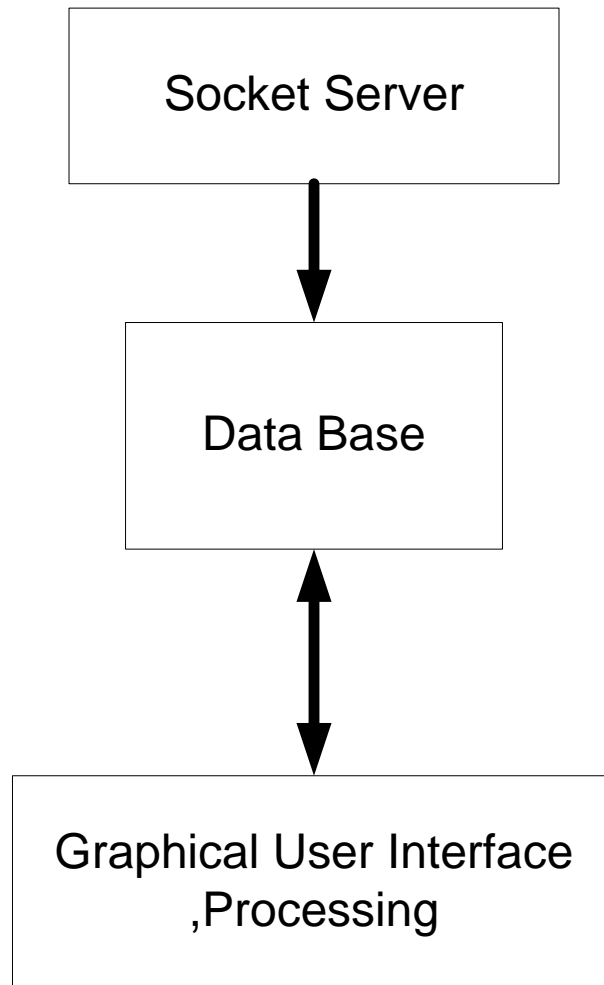


Figure 3.9 Block diagram for main part of Program

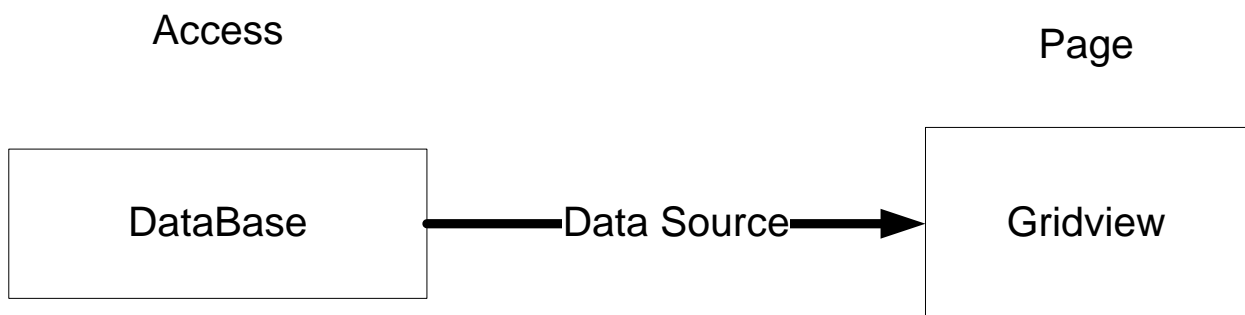


Figure 3.10 Block diagram for main part of web Server

3.9 failure detection

As shown in fig 3.6 the flow chart of system operation of fault detection which it starts with reading the current line to the single current transformer and 3-phase to core C.T then read current to circuit of figure 3.5 ,then and the output of it read to ADC of microcontroller .

In microcontroller the current signal for single lines phase compared with a threshold x over time interval (eg.100 ms) ,and with threshold y over time interval (eg.200 ms) for core phase ; and if current exceed this thresholds over time interval send alarm and the value of current to central office , send SMS to employee charge of this site.

In the Situation normal The system transmitted the value of current each 1 hour .

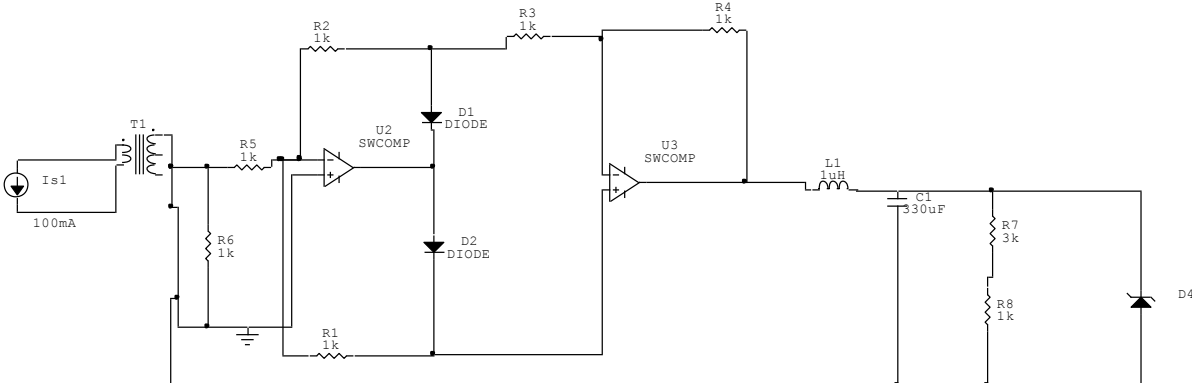


Figure 3.5 circuit diagram for fault detection

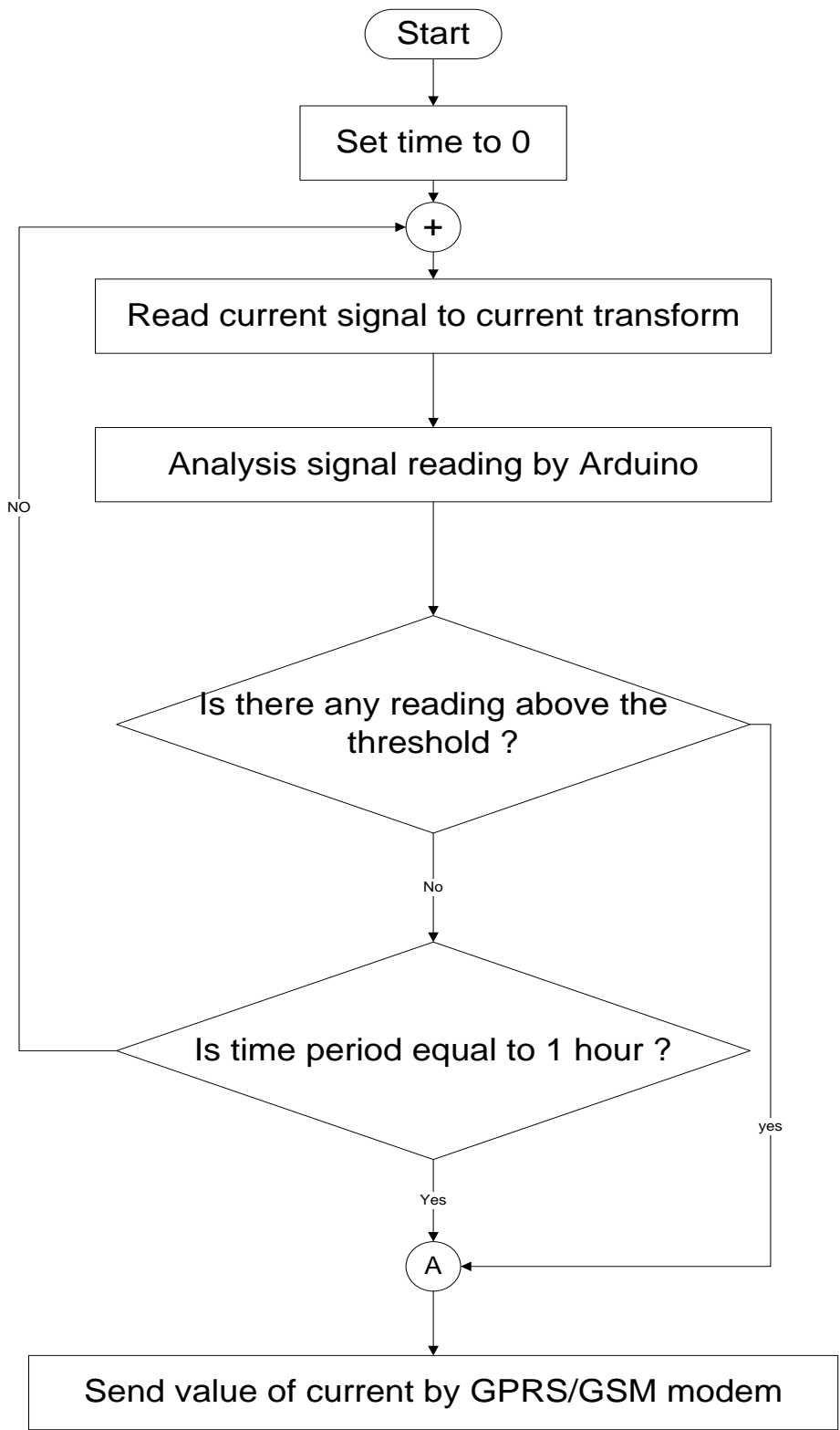


Figure 3.6 A flowchart of the system operation for fault detection

3.10 Environmental sensing

The following block diagram (fig 3.7) show the main part for environmental sensing the humidity , temperature ,fire ,water leakage for case node .

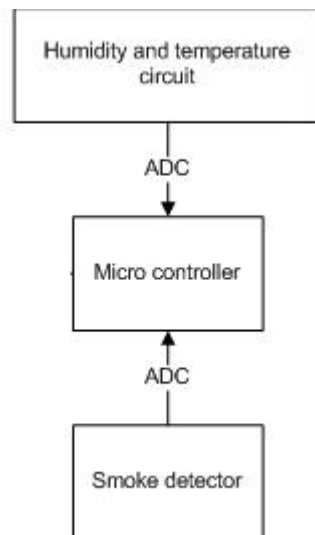


Figure 3.7 : General block diagram of environmental sensing

As shown in the following flow chart (fig 3.8) the system measurement the environmental and if any of measurements exceeds thresholds to each environments ,the system transmitted a alarm , type and value of fault to central office , send SMS to employee charge of this site and send MSG to Civil defense .

In the Situation normal The system transmitted the status of environmental each 1 hour .

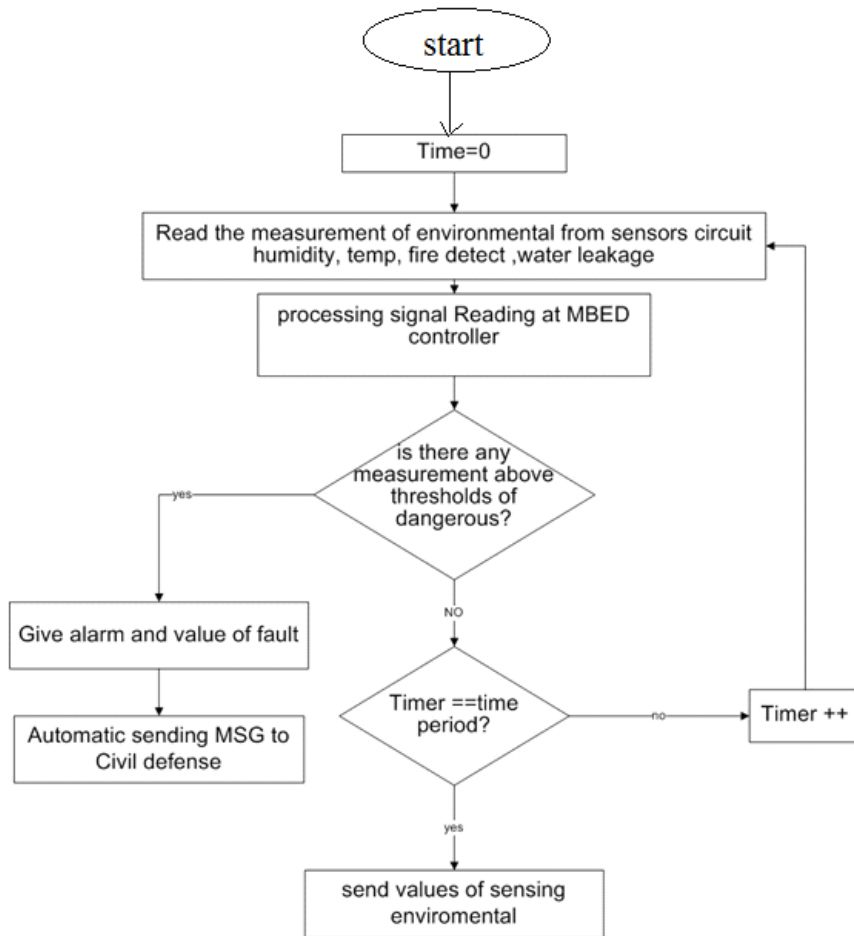


Figure 3.8 flowchart of system operation of sensing environmental

4

Chapter Four

Hardware and Software Design Implementation

4.1 Introduction

4.2 Software Design

4.3 Fault Detection System

4.4 Environmental Circuits

4.5 Multiple serial communication

4.6 Tiny RTC

4.7 GPRS/GSM

4.8 Message Format

4.9 web server

4.1 introduction

After viewing the general system design in the previous chapter , it is now the time for presenting the specific details of the connection and interfacing several component in the system.

This chapter presents the hardware details by pointing to each pins and showing how its combined other pins in each device .

4.2 Software design

This section describes the software part in the project which includes the PC (central office) software.

4.2.1 Central office software

The central office software can be divided into two parts, the basic functions that includes adding and modifying information, and the communication process that includes the communication details.

- At the beginning, the user of the software program must select the path of data base and the path of Alarm Sound , then the program check the port if it available as shown in figure 4.1,4.2,4.3,4.4,4.5 .



Figure 4.1 setup frame of program

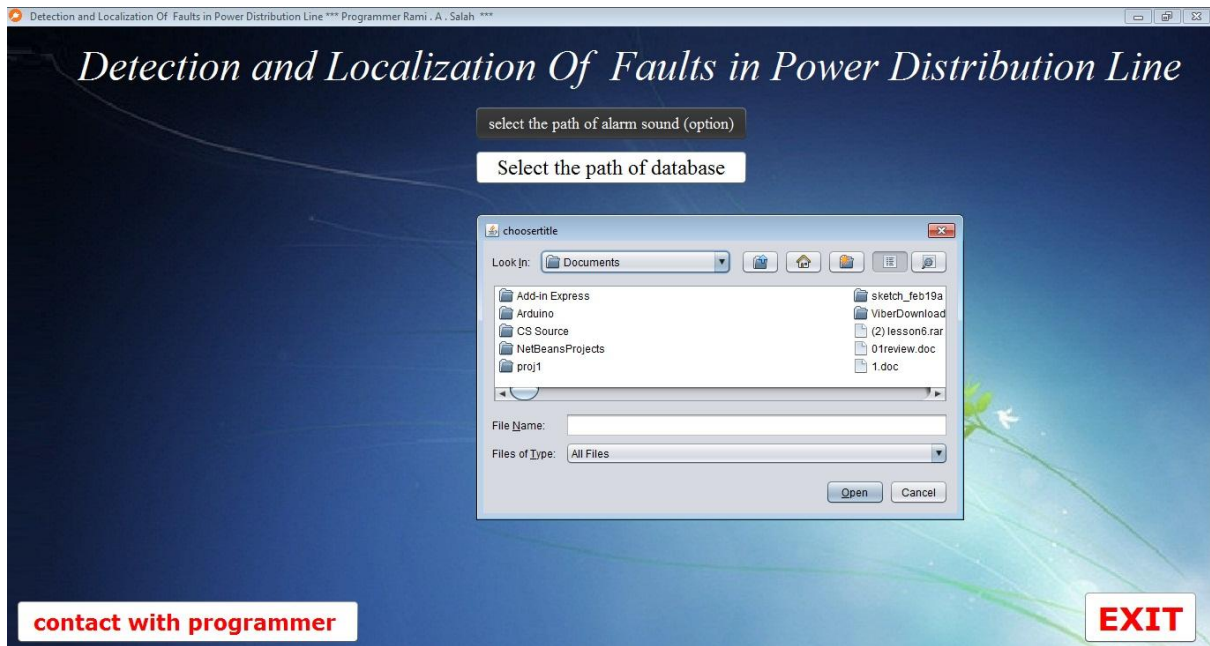


Figure 4.2 select the path of alarm sound

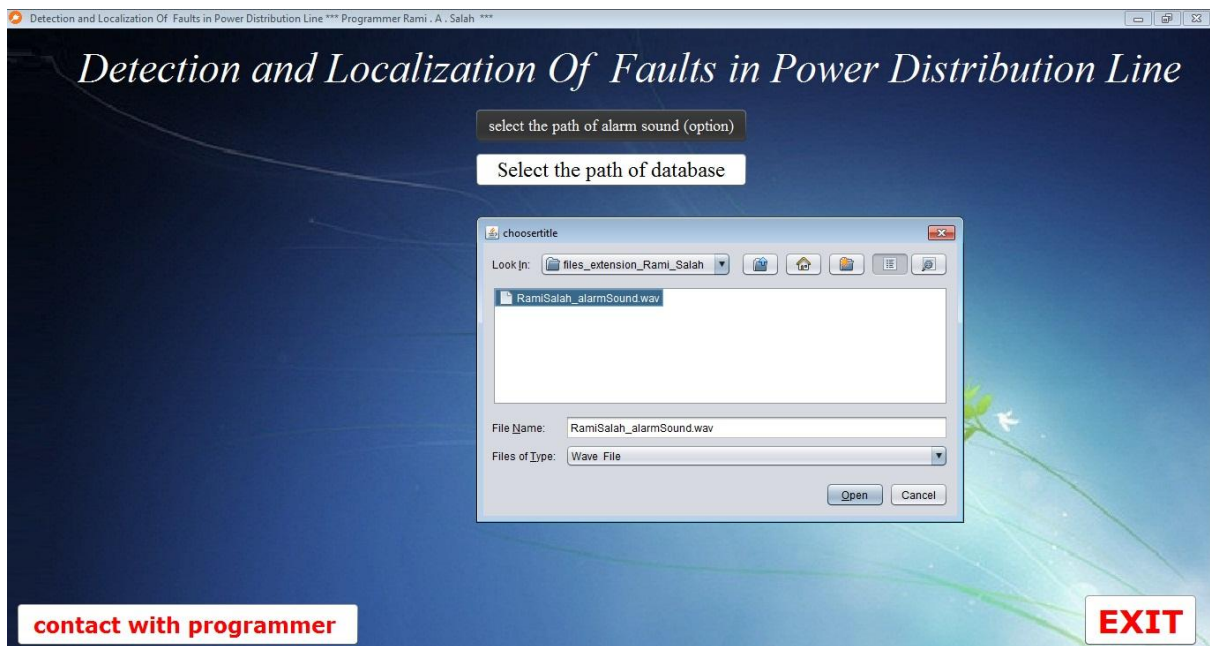


Figure 4.3 select the path of alarm sound continue

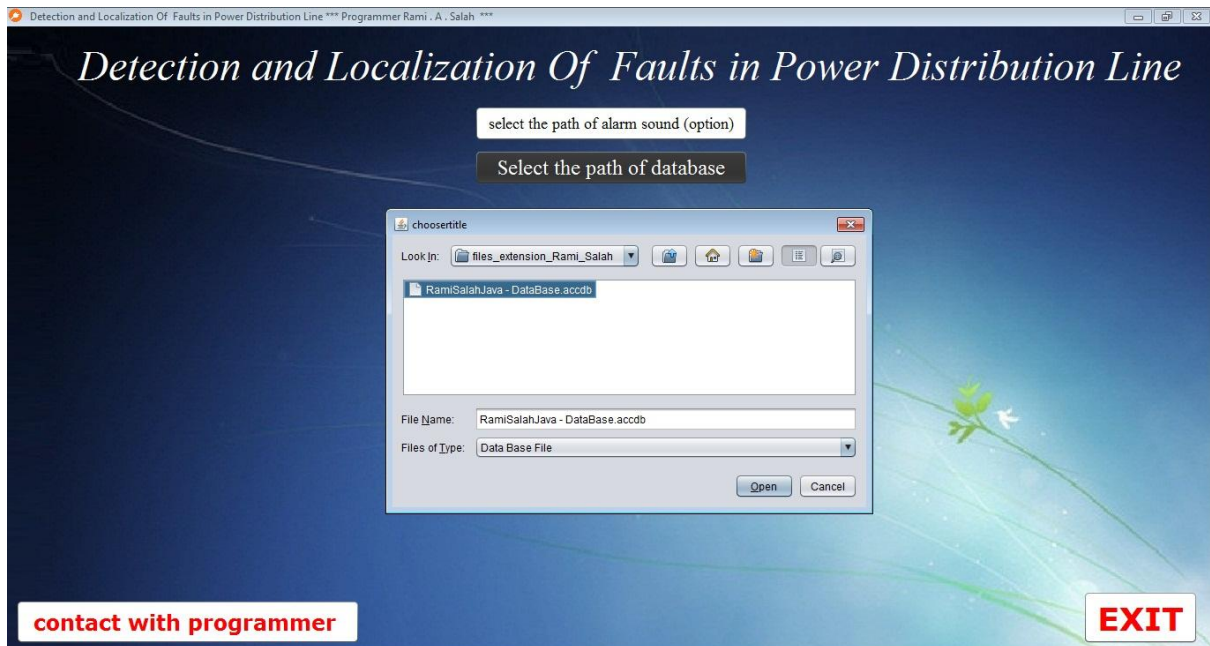


Figure 4.4 select the path of database

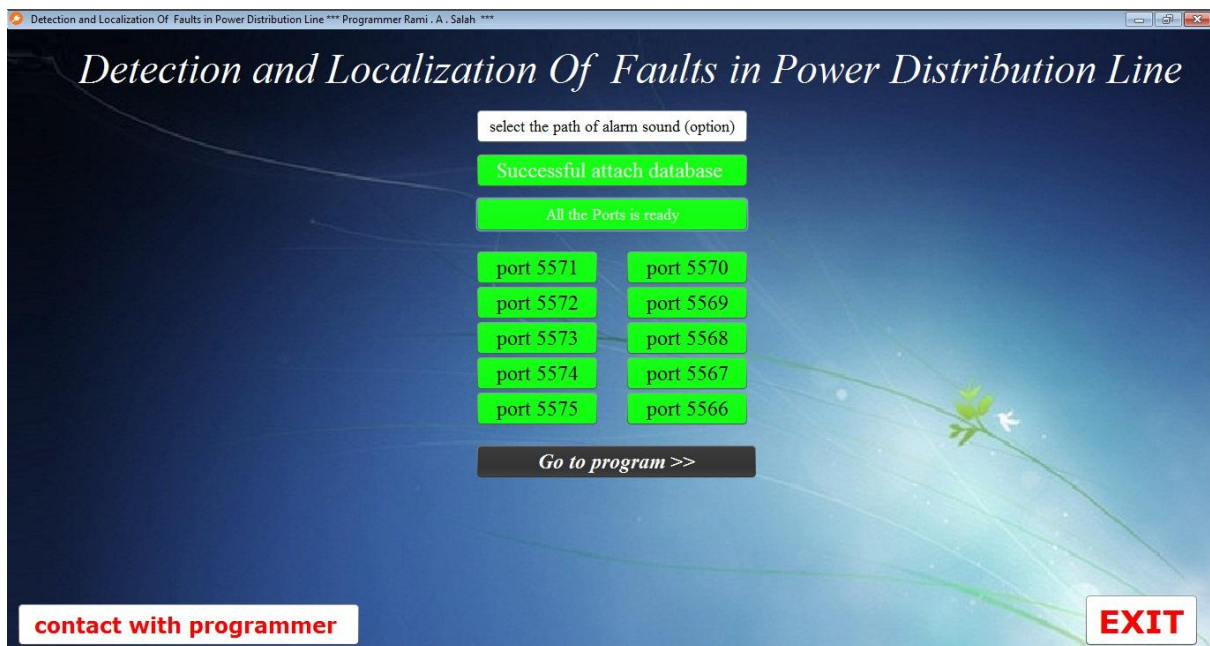


Figure 4.5 check that the port is available

- the user of the software program must log in the program with his name and specific password in order to use Program. After checking the user authority as shown in figure 4.6 .

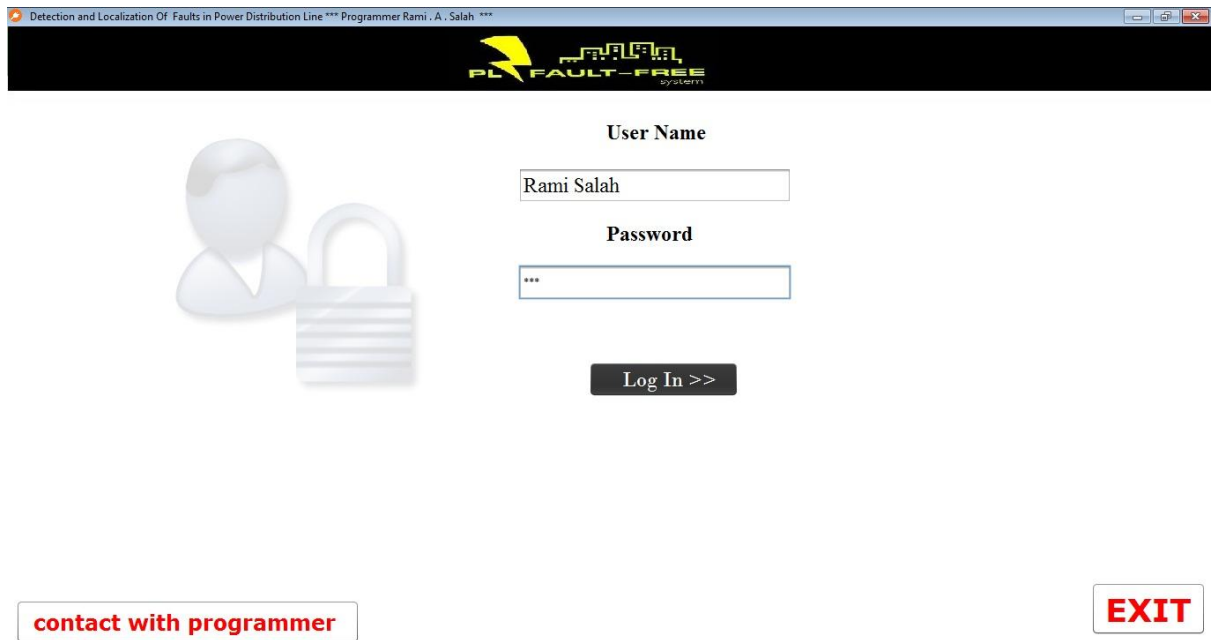


Figure 4.6 login frame

- Map Frame :
 - Present all the node as shown in figure 4.7 and the Measurement for each node by click on the node button as shown in figure 4.8 .
 - Indicate of status nodes by color :
 - Red : Danger .
 - Orange : Not connect.
 - Green : Safe .
 - When Dangerous or Not Connect the program make this step :
 - Output flash light .
 - Output alarm sound.
 - Output the name of node in Text Area .
 - Change Color of node to Red or Orange .
 -

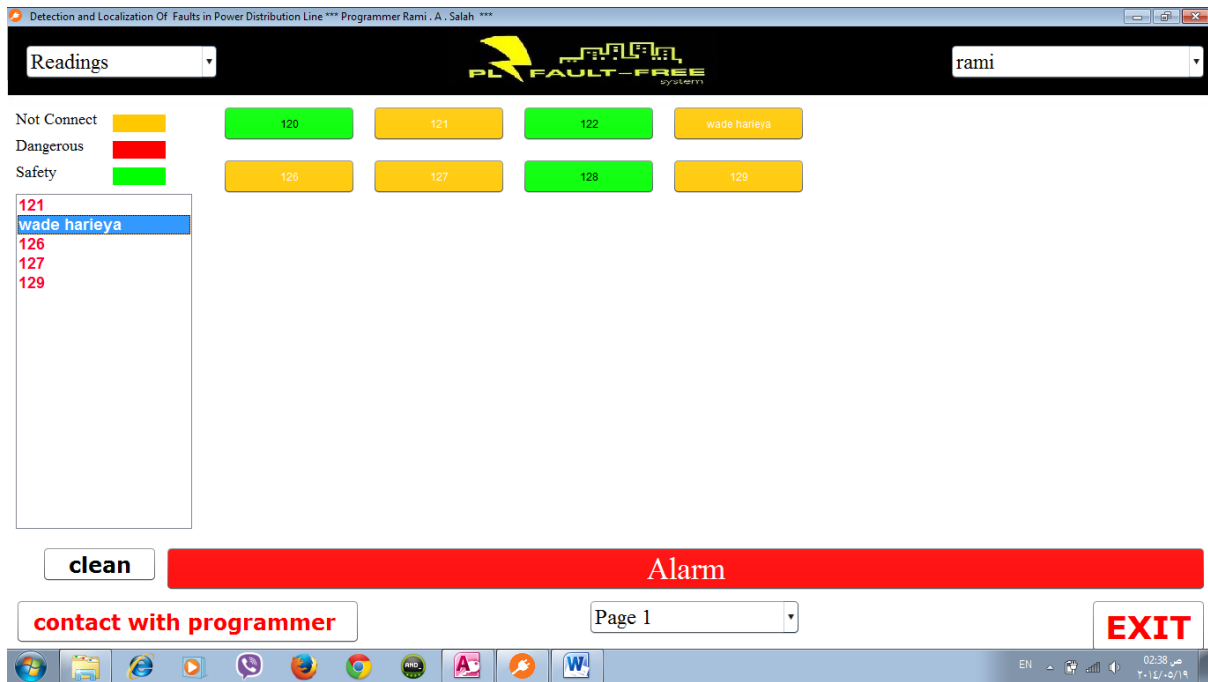


Figure 4.7 map frame

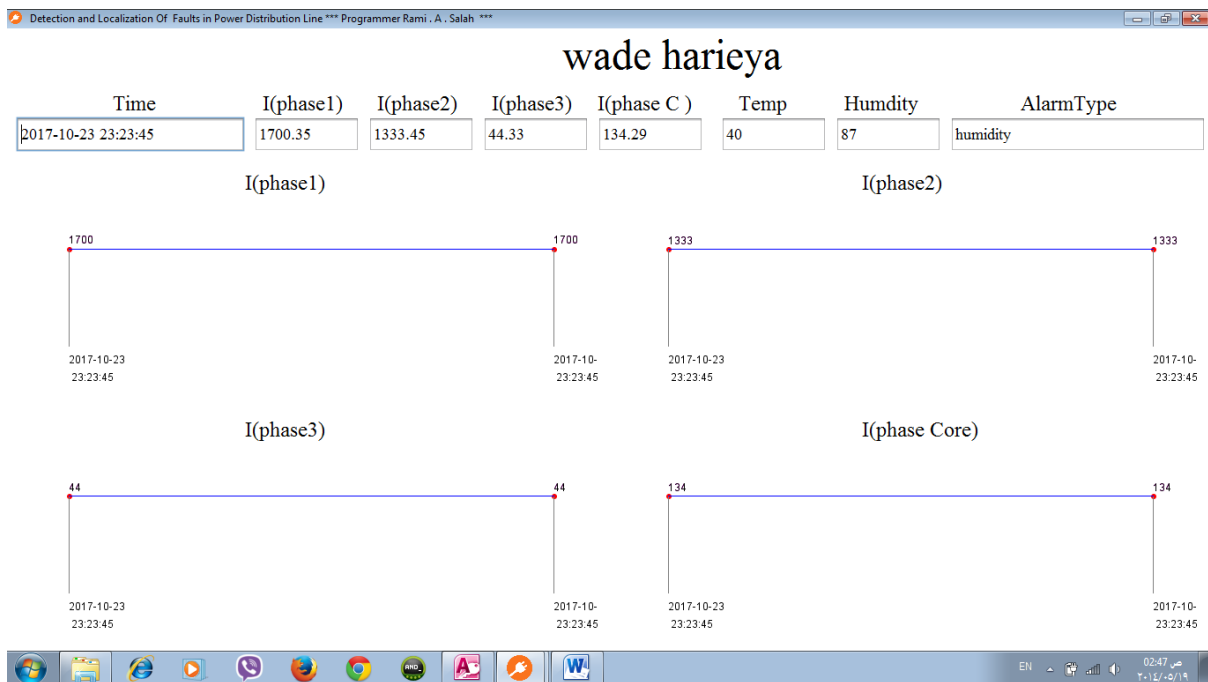


Figure 4.8 frame for monitoring each node online

- Node Frame :
 - Represent all the Node with all information .
 - Allow user to make 3 method :
 - Add Node as shown in figure 4.9 .
 - Modify Node as shown in figure 4.10.
 - Removes Node as shown in figure 4.11.

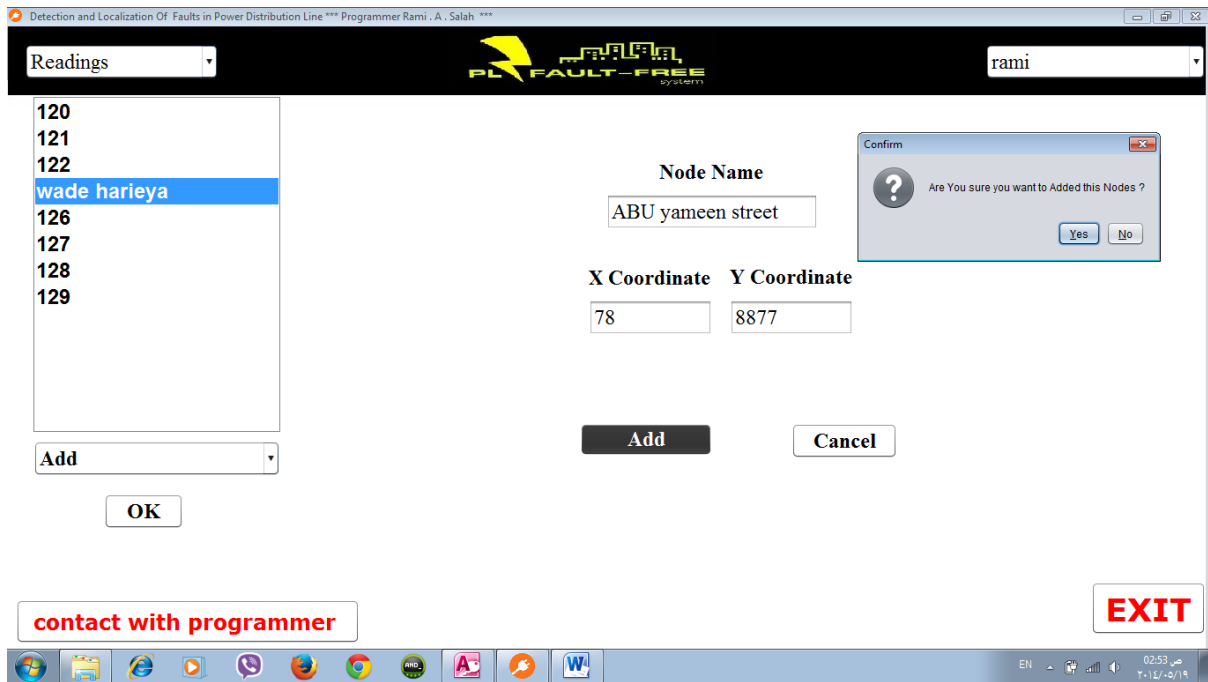


Figure 4.9 node frame to add anode

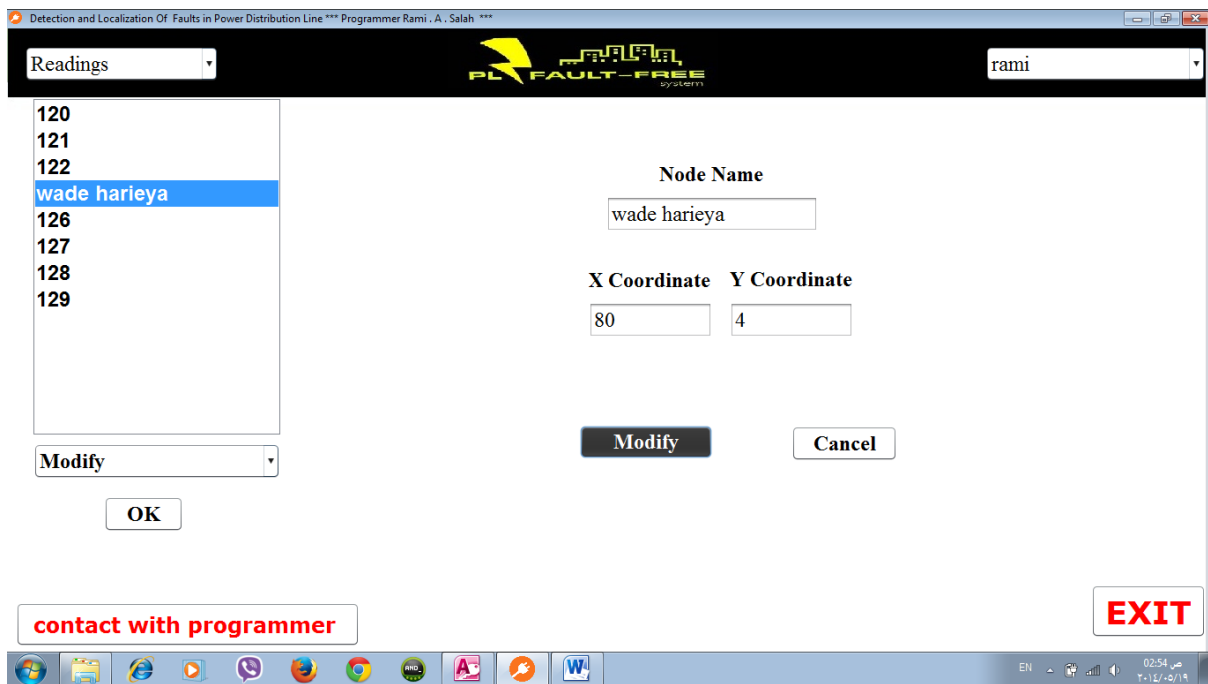


Figure 4.10 node frame for modifying the information of the node

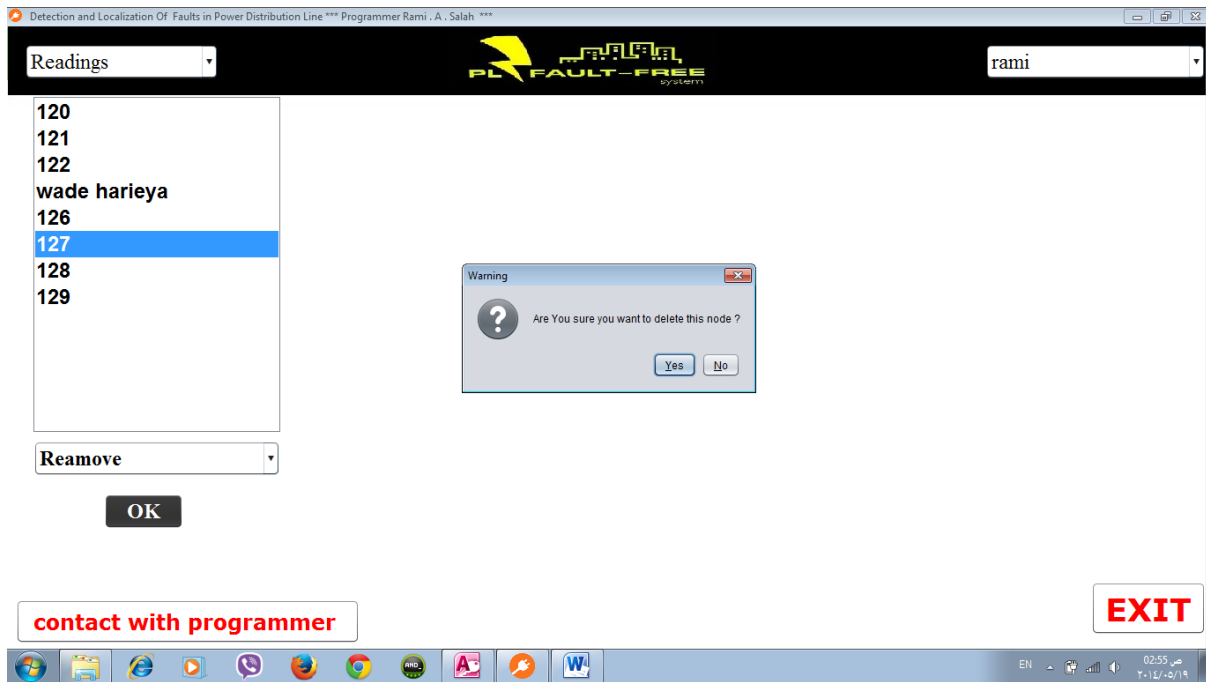


Figure 4.11 node frame to Delete anode

- Alarm Frame :
 - Represent all the Alarm type with all information .
 - Allow user to make 3 method :
 - Add Alarm as shown in figure 4.12.
 - Modify Alarm as shown in figure 4.13.
 - Removes Alarm as shown in figure 4.14.

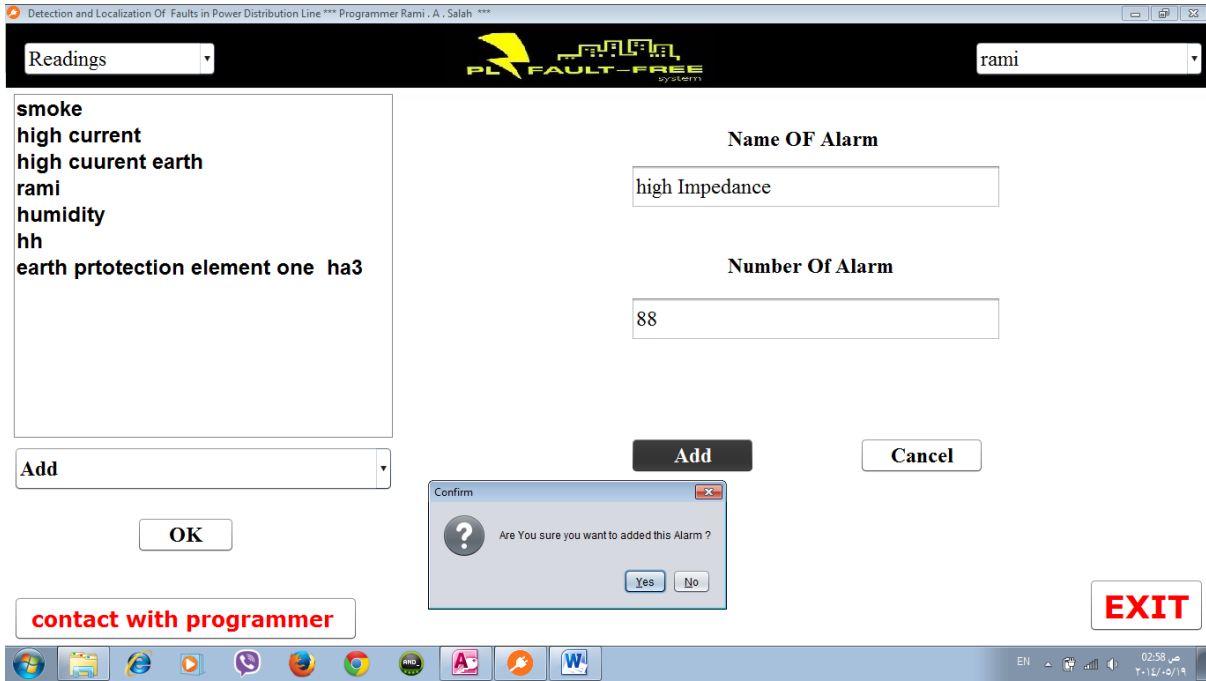


Figure 4.12 alarm frame to add an alarm

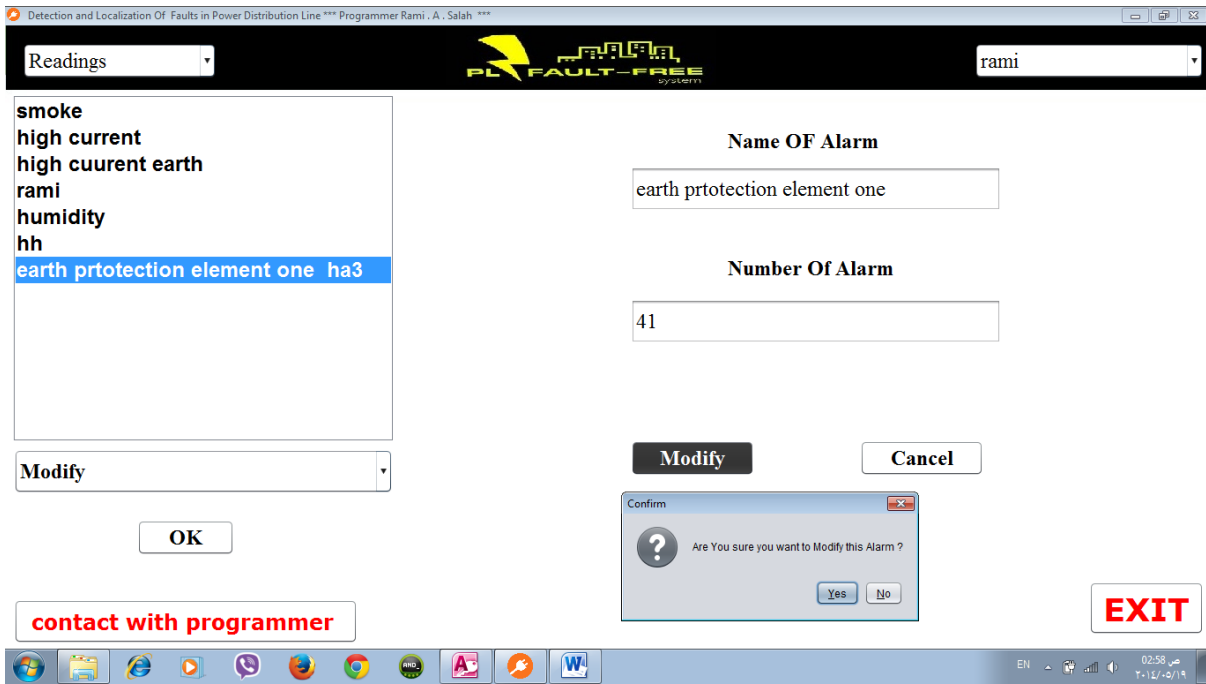


Figure 4.13 alarm frame to modify an alarm

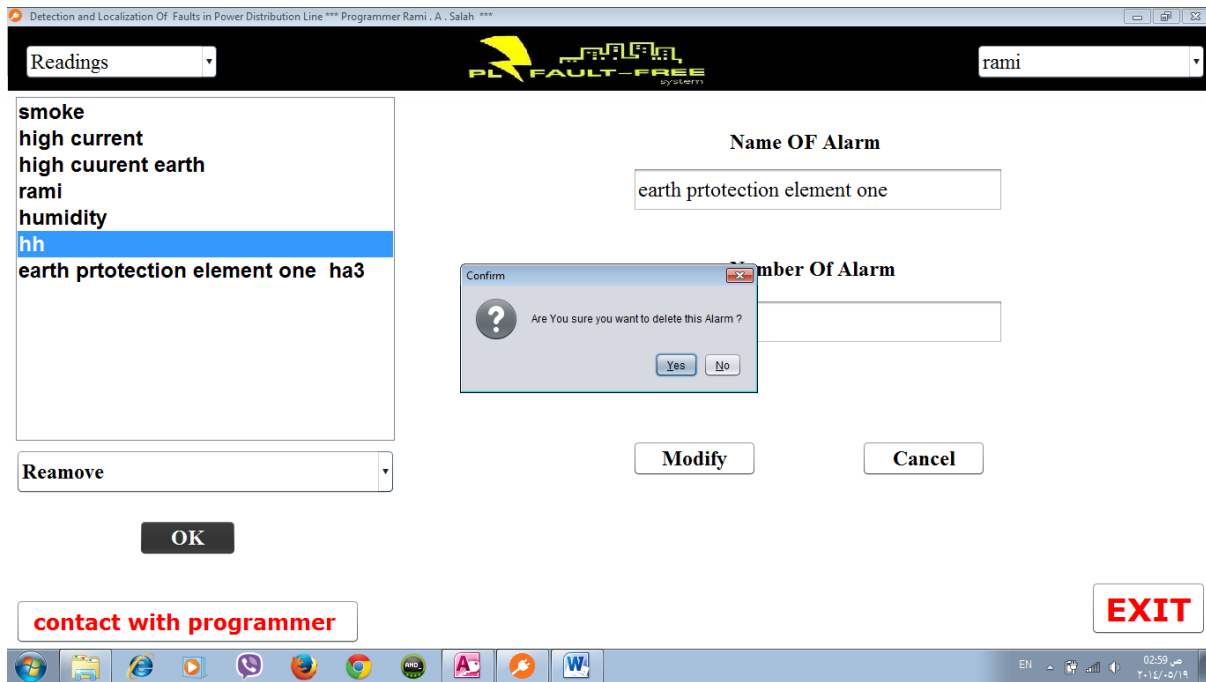


Figure 4.14 alarm frame to delete an alarm

- User Frame :
 - Represent all the User type with all information .
 - Allow user to make 3 method :
 - Add User .
 - Modify User as shown in figure 4.16 .
 - Removes User as shown in figure 4.15.
 - The program sort users in 3 level as shown in figure 4.17:
 - Administrator can access to this :
 - User frame .
 - Node frame.
 - Alarm frame.
 - Map frame.
 - Reading frame.
 - Deputy of Administrator can access to this :
 - Node frame.
 - Alarm frame.
 - Map frame.
 - Reading frame.
 - Supervisor can access to this :
 - Map frame.
 - Reading frame.

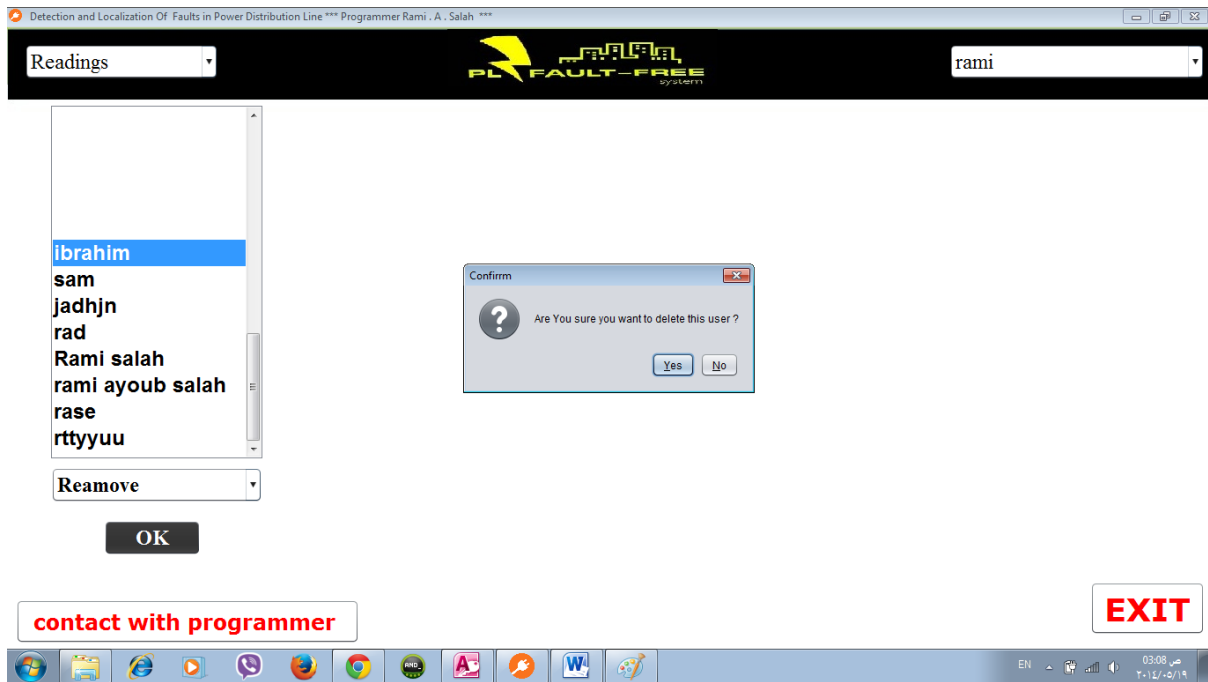


Figure 4.15 user frame to delete a user

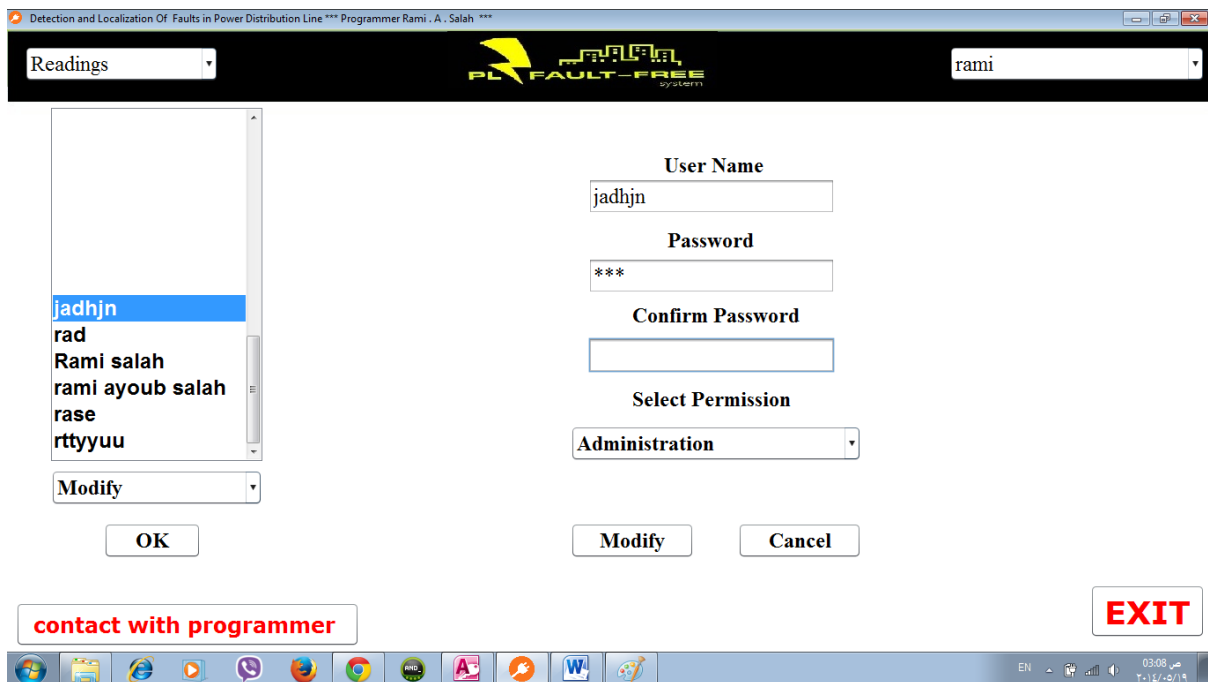


Figure 4.16 user frame to Modify info of a user

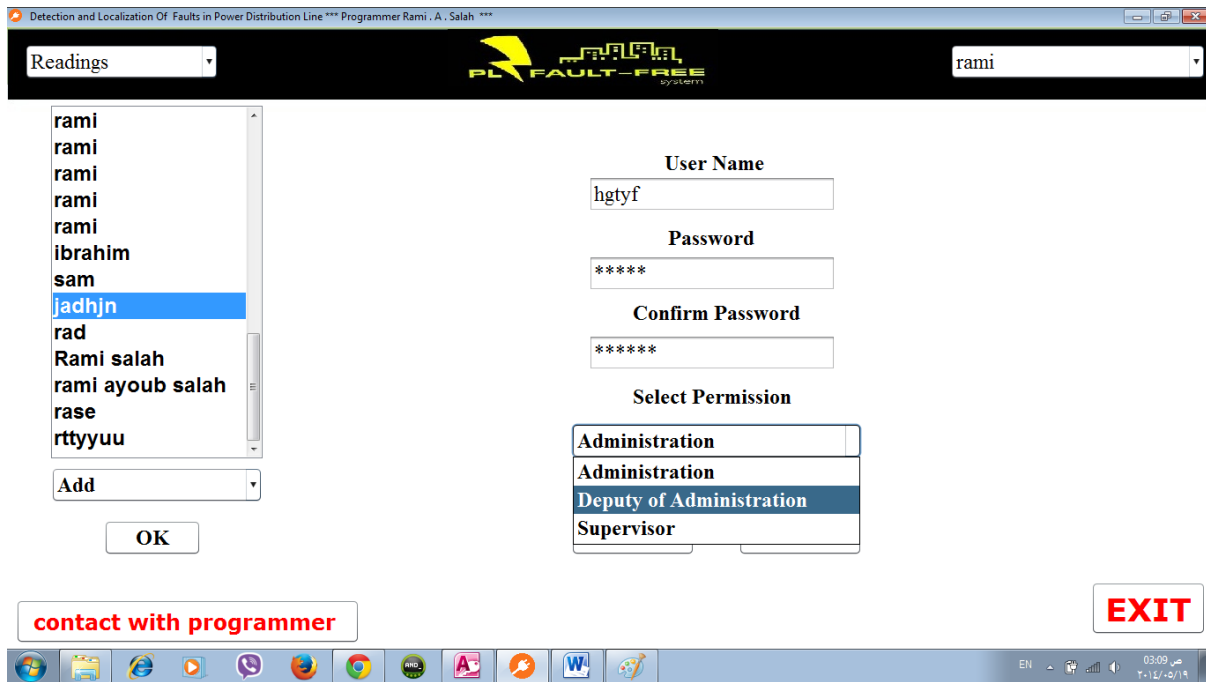


Figure 4.17 user frame to set authentication for the user

- Reading Frame :
 - Represent all the Reading with all of details .
 - Allow user to make 4 method :
 - Draw the relations between the current and time for specified Node as shown in figure 4.21.
 - Represent the reading by date and by name of node searched by user as shown in figure 4.20 .
 - Represent the reading which has a dangerous as shown in figure 4.19 .
 - Represent the Node which has not connect as shown in figure 4.18.

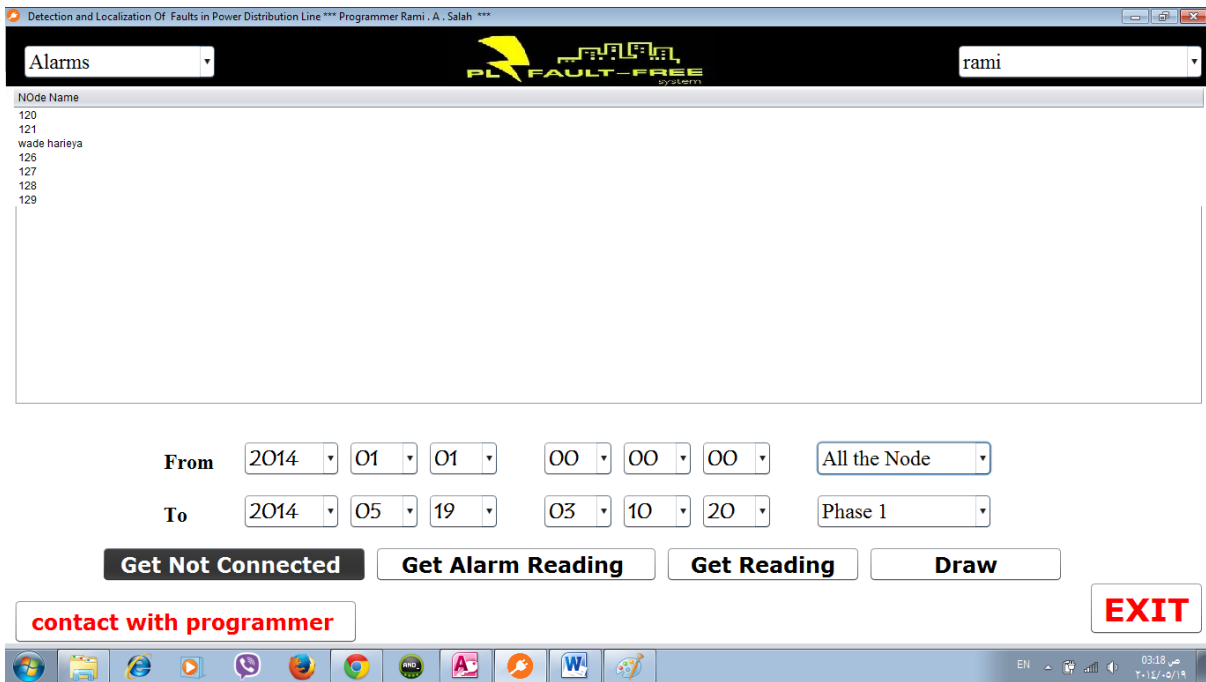


Figure 4.18 reading frame Represent the Node which has not connect

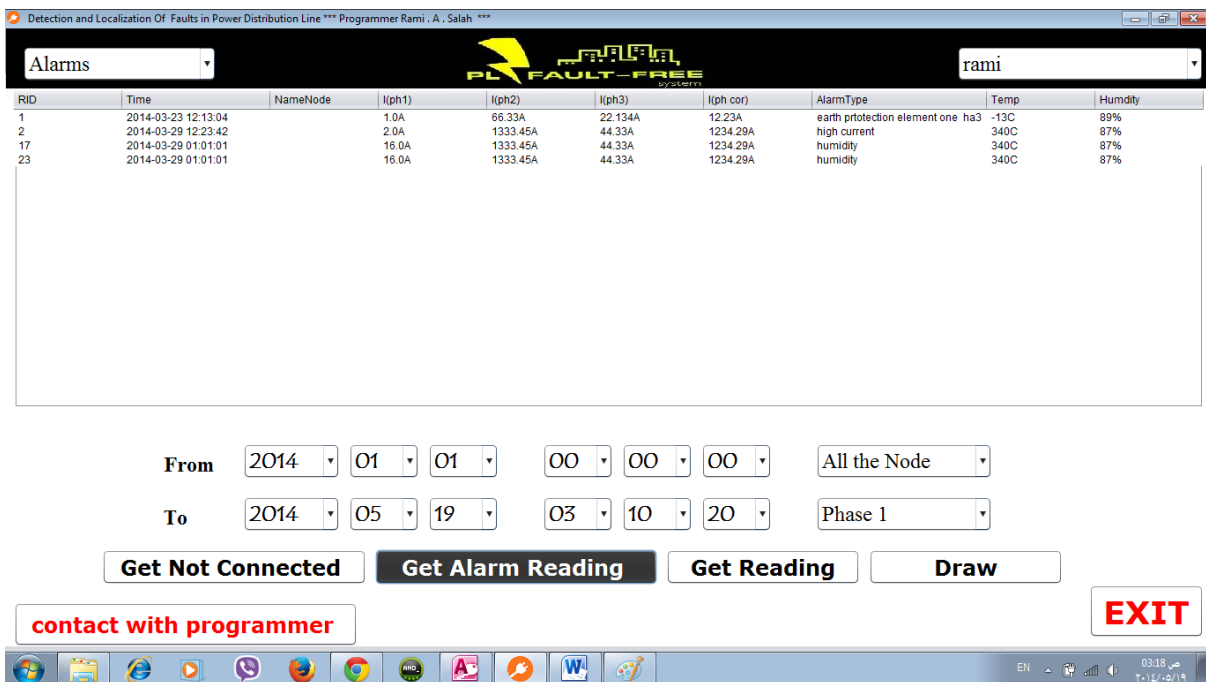


Figure 4.19 reading frame to get the reading of alarm

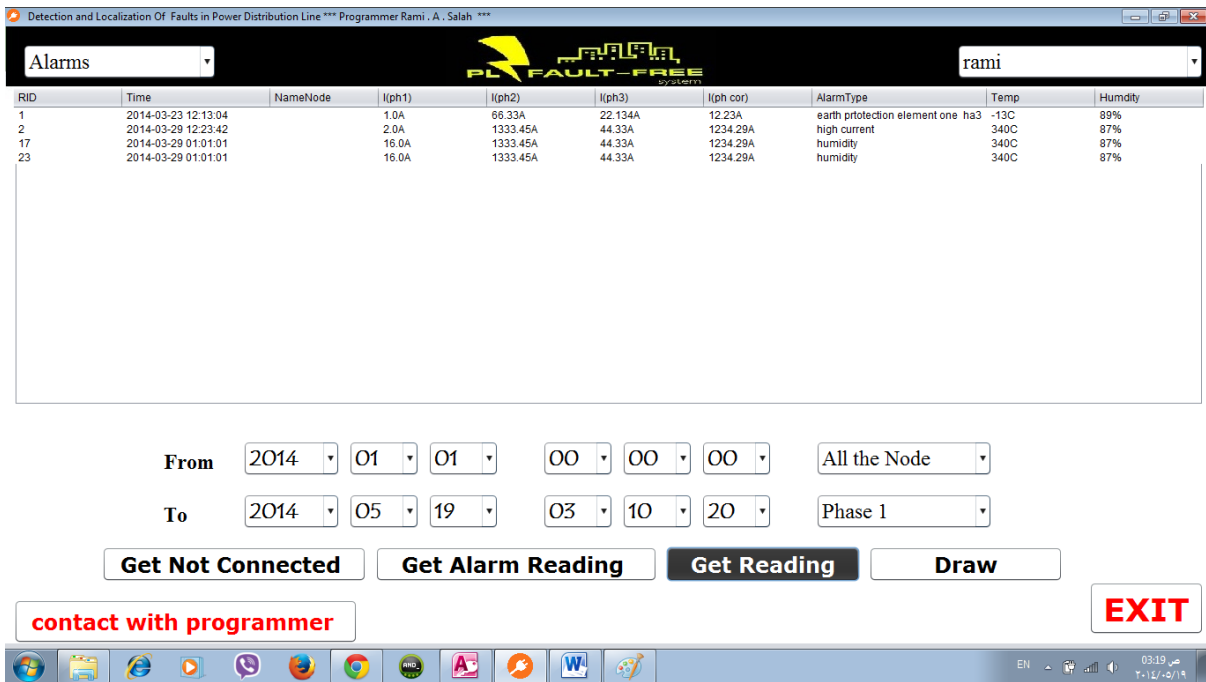


Figure 4.20 reading frame to get all readings

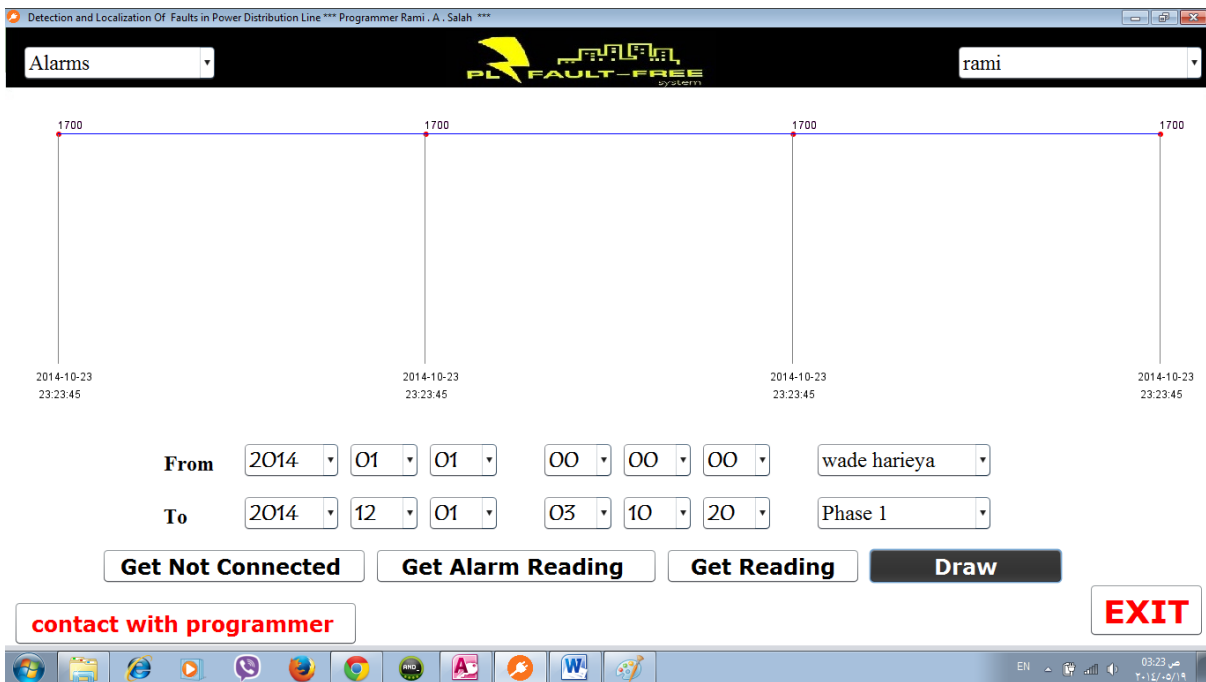


Figure 4.21 Draw the relations between the current and time

- Logout Icon : as shown in figure 4.22 when the user press Logout icon ,the program will be logout from user account as shown in figure 4.23 .

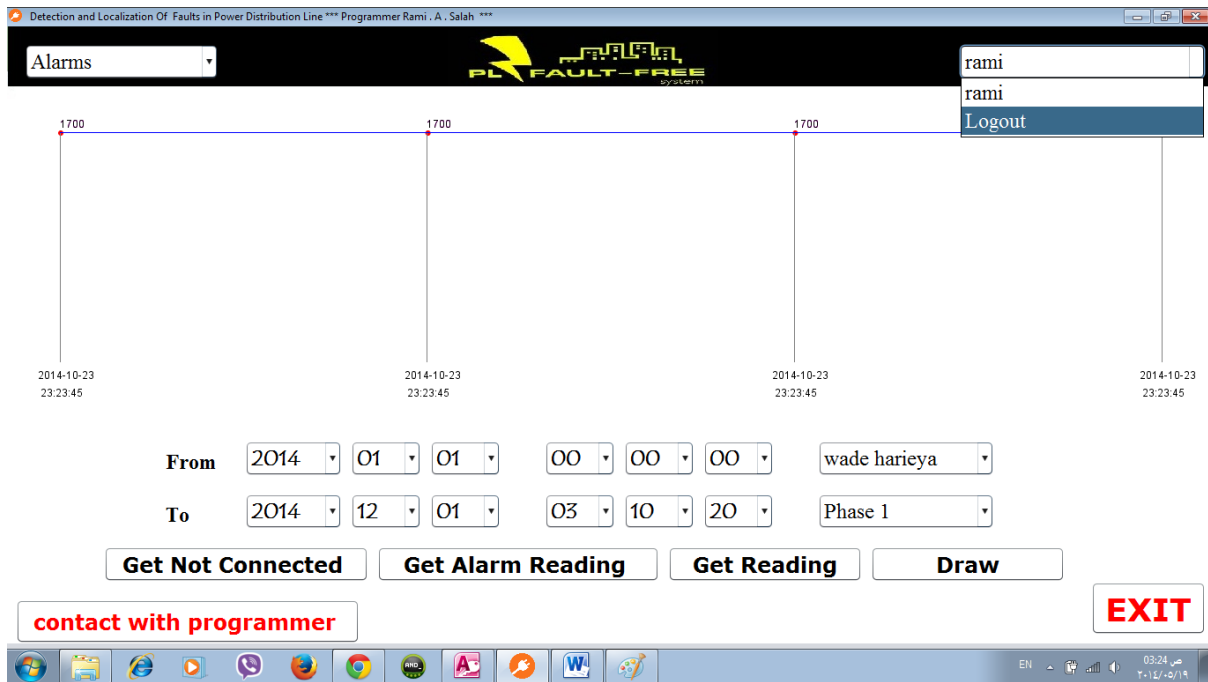


Figure 4.22 Logout icon

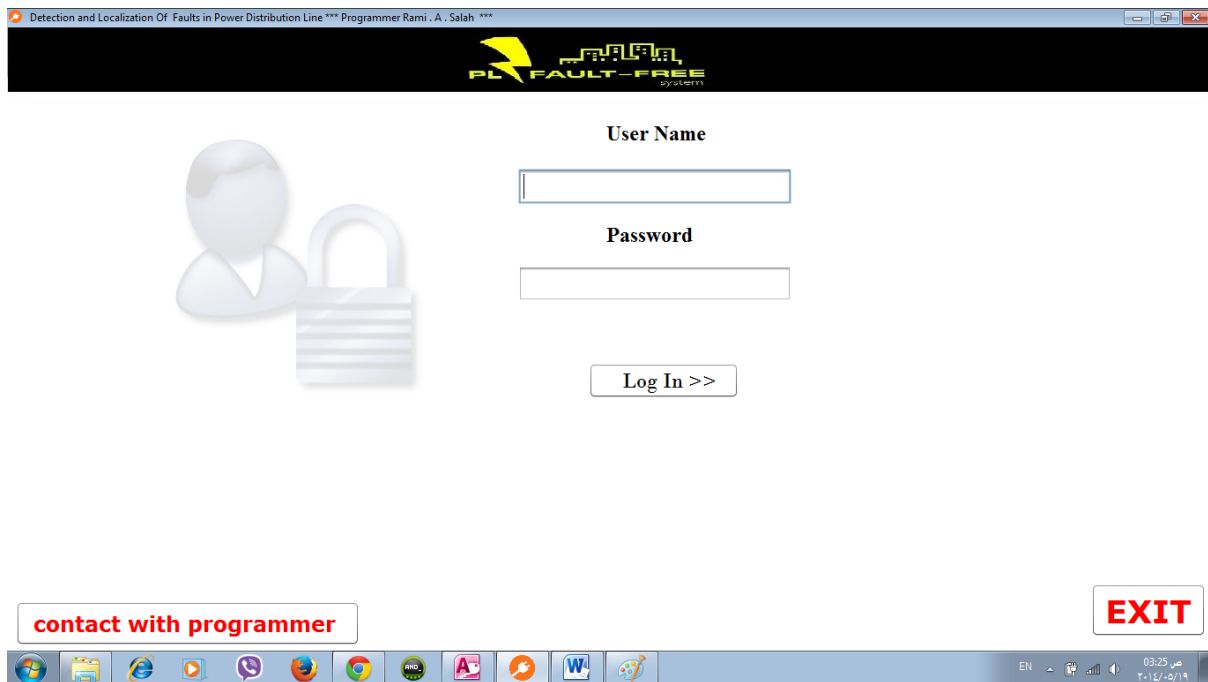


Figure 4.23 the frame after the logout

4.3 fault detection system

As shown circuit in figure 3.5 in chapter 3 ,this circuit convert Ac signal voltage to Dc VRMS signal .

- First reading the current signal to current transformer (400:5 A) .
 - Figure 4.24 is a diagram for AC signal .

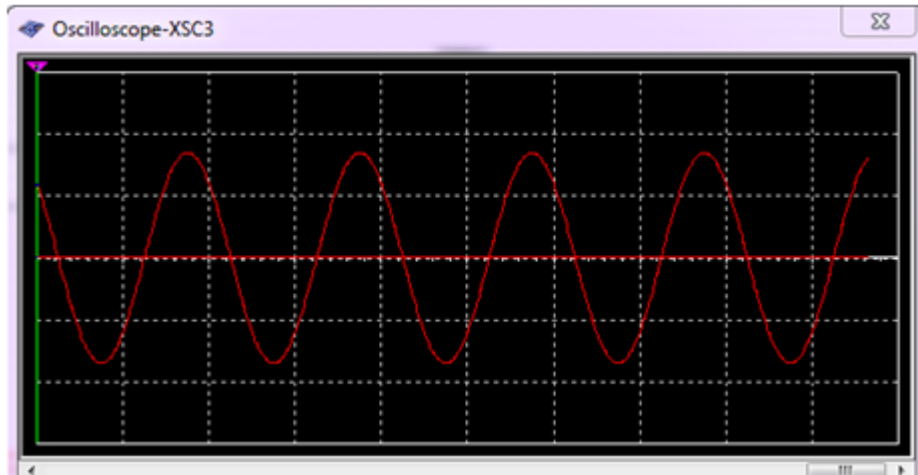


Figure 4.24 signal to current transformer

- Read the current signal to resistance, To convert the current to voltage .
- Read current signal to rest of circuit to convert current signal from AC signal to DC signal.
 - Figure 4.25 is the output of circuit .

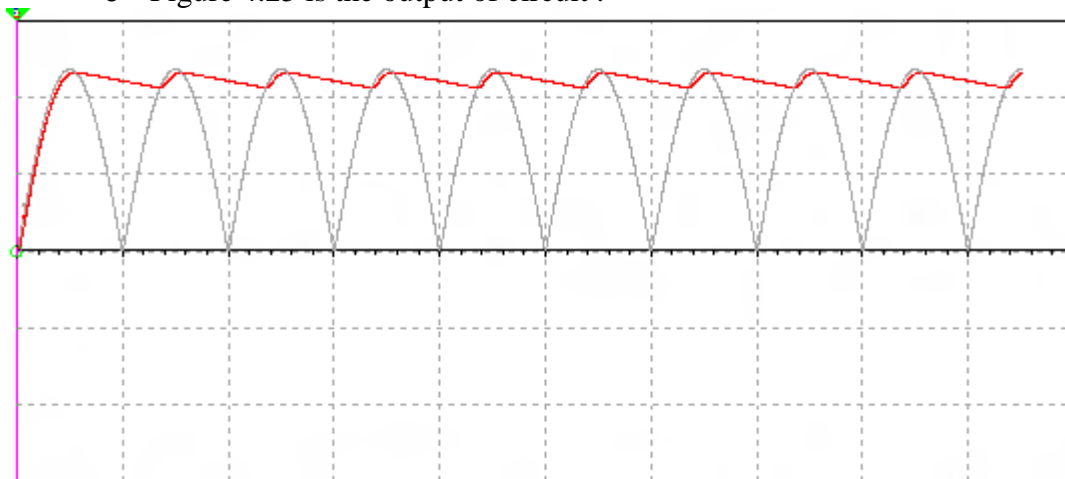


Figure 4.25 conversion from AC to DC

- We use two resistance (1K ohms,3K ohms) to divide the voltage to fourth.
- We want to protect the Microcontroller from high current may happen. so , I will use zener diode.
- In microcontroller :
 - Define a specific ADC pin as analog voltage reading .
 - Measurement the voltage .
- Compare the reading with thresholds and substitution in equation .
- Transfer the value of current from the Arduino to the central Arduino by serial.

- Code for reading analog signal :
 - This code reading analog voltage in Pin A0 (ADC) and print the value on Serial monitoring .

```

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage);
}

```

4.4 environmental circuits

- **4.4.1 Grove - Gas Sensor(MQ2)**
 - The Grove - Gas Sensor(MQ2) module is useful for gas leakage detecting(in home and industry). It can detect LPG, i-butane, methane, alcohol, Hydrogen, smoke and so on. Based on its fast response time. measurements can be taken as soon as possible. Also the sensitivity can be adjusted by the potentiometer. Figure 4.26 show the connection between MQ2 and Arduino .

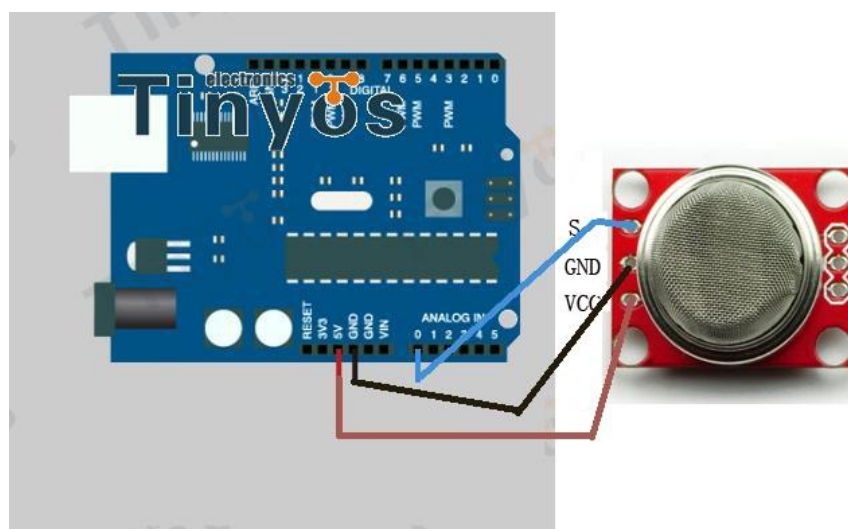


Figure 4.26 the circuit connection of Grove - Gas Sensor(MQ2)with Arduino

- Code for reading MQ2 :
 - This code reading analog output from MQ2 in pin A0(ADC) and print the value on screen , when happened smoke the value of analog will be increase .

```

void setup() {
  Serial.begin(9600);
}

void loop() {
  float vol;
  int sensorValue = analogRead(A0);
  vol=(float)sensorValue/1024*5.0;
  Serial.println(vol,1);
}

```

- **4.4.2 Digital output temperature and humidity sensor DHT11**

- Figure 4.27 show the connection between DHT11 and Arduino .
- Needs 4.7K to 10K pull up resistor from +5 to Signal pin. Connect to Digital pin, not Analog pin.

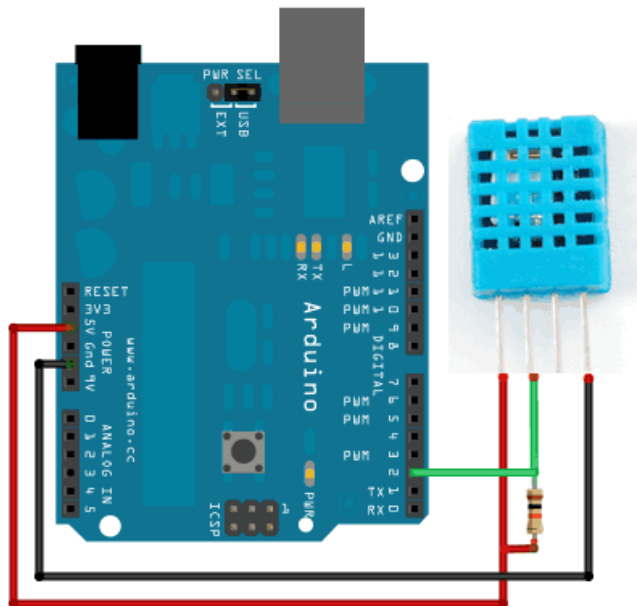


Figure 4.27 circuit of connection between DHT11 sensor with Arduino

- Code for **DHT11** :

-

- This code reading humidity and temperature reading from DHT11 in digital pin 2 ,and print the value of humidity and temperature on serial monitor .

```

/*-----( Import needed libraries )-----*/
#include <dht11.h>

/*-----( Declare objects )-----*/
dht11 DHT11;

/*-----( Declare Constants, Pin Numbers )-----*/
#define DHT11PIN 2

```

```

void setup() /*----( SETUP: RUNS ONCE )----*/
{
  Serial.begin(9600);
  Serial.println("DHT11 TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT11LIB_VERSION);
  Serial.println();
}/*--(end setup )---*/

void loop() /*----( LOOP: RUNS CONSTANTLY )----*/
{
  Serial.println("\n");

  int chk = DHT11.read(DHT11PIN);

  Serial.print("Read sensor: ");
  switch (chk)
  {
    case 0: Serial.println("OK"); break;
    case -1: Serial.println("Checksum error"); break;
    case -2: Serial.println("Time out error"); break;
    default: Serial.println("Unknown error"); break;
  }

  Serial.print("Humidity (%): ");
  Serial.println((float)DHT11.humidity, 2);

  Serial.print("Temperature (oC): ");
  Serial.println((float)DHT11.temperature, 2);

  Serial.print("Temperature (oF): ");
  Serial.println(Fahrenheit(DHT11.temperature), 2);

  Serial.print("Temperature (K): ");
  Serial.println(Kelvin(DHT11.temperature), 2);

  Serial.print("Dew Point (oC): ");
  Serial.println(dewPoint(DHT11.temperature, DHT11.humidity));

  Serial.print("Dew PointFast (oC): ");
  Serial.println(dewPointFast(DHT11.temperature, DHT11.humidity));

  delay(2000);
}/* --(end main loop )-- */

/*----( Declare User-written Functions )----*/
//
//Celsius to Fahrenheit conversion
double Fahrenheit(double celsius)

```

```

{
    return 1.8 * celsius + 32;
}

//Celsius to Kelvin conversion
double Kelvin(double celsius)
{
    return celsius + 273.15;
}

// dewPoint function NOAA
// reference: http://wahiduddin.net/calc/density_algorithms.htm
double dewPoint(double celsius, double humidity)
{
    double A0= 373.15/(273.15 + celsius);
    double SUM = -7.90298 * (A0-1);
    SUM += 5.02808 * log10(A0);
    SUM += -1.3816e-7 * (pow(10, (11.344*(1-1/A0)))-1) ;
    SUM += 8.1328e-3 * (pow(10,(-3.49149*(A0-1)))-1) ;
    SUM += log10(1013.246);
    double VP = pow(10, SUM-3) * humidity;
    double T = log(VP/0.61078); // temp var
    return (241.88 * T) / (17.558-T);
}

// delta max = 0.6544 wrt dewPoint()
// 5x faster than dewPoint()
// reference: http://en.wikipedia.org/wiki/Dew_point
double dewPointFast(double celsius, double humidity)
{
    double a = 17.271;
    double b = 237.7;
    double temp = (a * celsius) / (b + celsius) + log(humidity/100);
    double Td = (b * temp) / (a - temp);
    return Td;
}

/* ( THE END ) */

```

4.5 Multiple serial communication :

- As shown in figure 4.28 each Arduino connect with central Arduino by 5 link as it :
 - RTS Bit (request to send) : request from Arduino central to send by serial .
 - CTS Bit (clear to send) : Allow to Arduino send which is send RTS.
 - SRK Bit (Send Reading Know) : tell to Arduino to send Reading .
 - TX Bit : transmitted to central Arduino .
 - RX Bit : received from Arduino central .

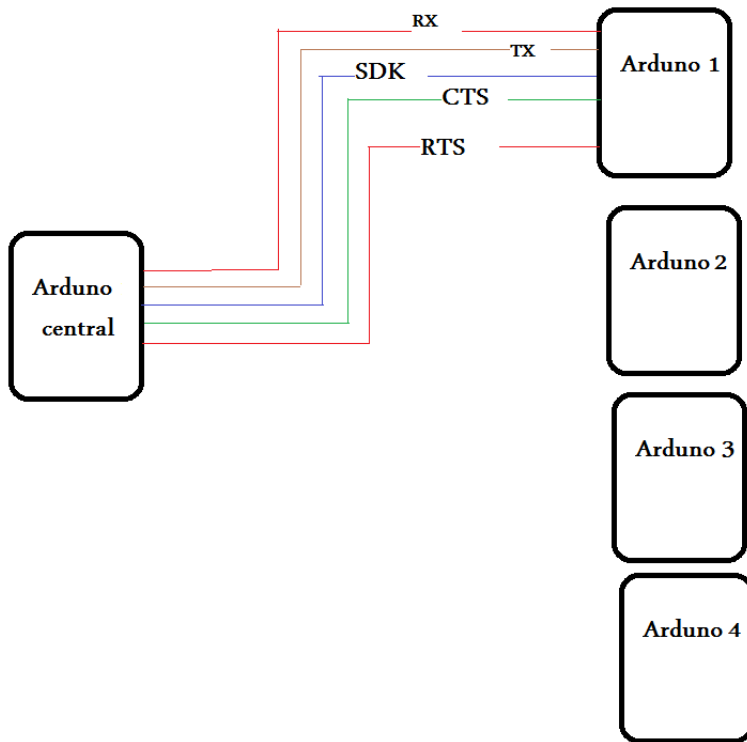


Figure 4.28 block diagram of multiple serial communication

4.6 Tiny RTC

The module I will be using is a Tiny RTC v1 module. Communication is done over an I2C interface.as shown in figure 4.29 It has a battery on-board that can supposedly last for several years.

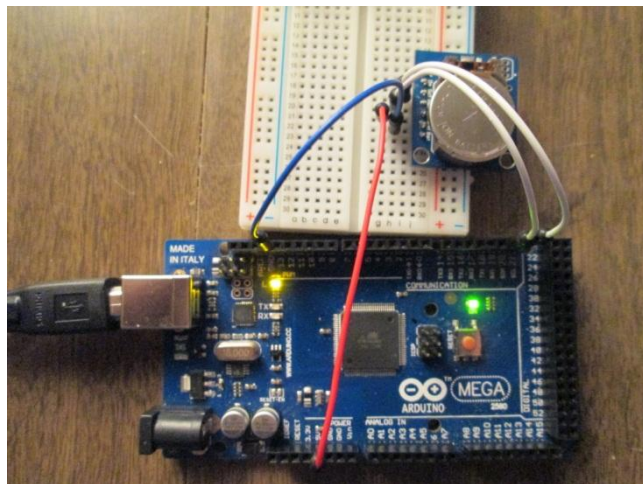


Figure 4.29 Tiny RTC with Arduino

Figure 4.30 show the connection between RTC with Arduino .

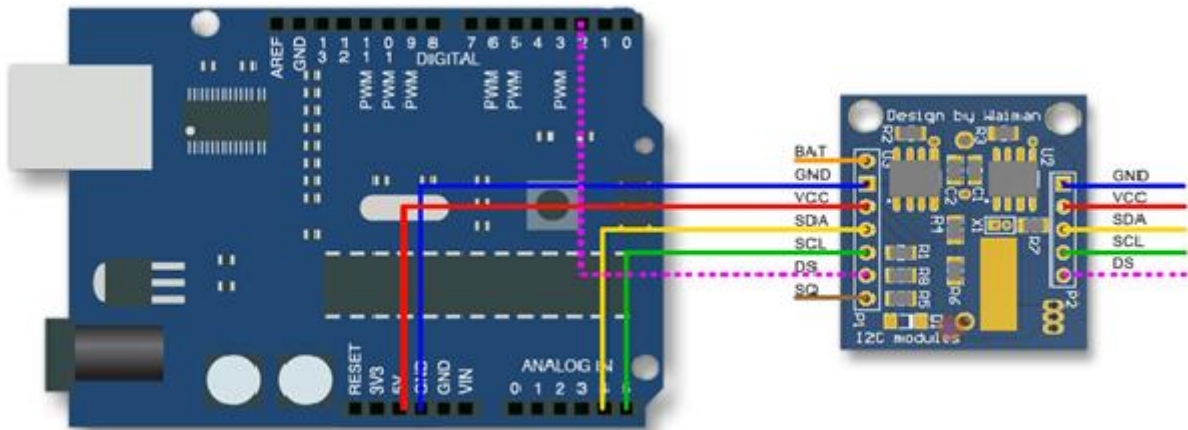


figure 4.30 Schematic diagram of connection between RTC and Arduino Mega

- Code for Get Time :
 - This code will get a time from RTC and print the time on serial monitor .

```
#include <Wire.h>

#include <Time.h>

#include <DS1307RTC.h>

void setup() {

  Serial.begin(9600);

  while (!Serial) ; // wait for serial

  delay(200);

  Serial.println("DS1307RTC Read Test");

  Serial.println("-----");

}

void loop() {

  tmElements_t tm;

  if (RTC.read(tm)) {

    Serial.print("Ok, Time = ");

    print2digits(tm.Hour);
```

```

Serial.write(':');

print2digits(tm.Minute);

Serial.write(':');

print2digits(tm.Second);

Serial.print(", Date (D/M/Y) = ");

Serial.print(tm.Day);

Serial.write('/');

Serial.print(tm.Month);

Serial.write('/');

Serial.print(tmYearToCalendar(tm.Year));

Serial.println();

} else {

  if (RTC.chipPresent()) {

    Serial.println("The DS1307 is stopped. Please run the SetTime");

    Serial.println("example to initialize the time and begin running.");

    Serial.println();

  } else {

    Serial.println("DS1307 read error! Please check the circuitry.");

    Serial.println();

  }

  delay(9000);

}

delay(1000);

}

void print2digits(int number) {

  if (number >= 0 && number < 10) {

    Serial.write('0');

```



```
}  
  
Serial.print(number);  
  
}
```

- Code for Set Time :
 - This code will be download on Arduino only I time ,when set the time of PC to RTC .

```
#include <Wire.h>  
  
#include <Time.h>  
  
#include <DS1307RTC.h>  
  
const char *monthName[12] = {  
  
  "Jan", "Feb", "Mar", "Apr", "May", "Jun",  
  
  "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"  
  
};  
  
tmElements_t tm;  
  
void setup() {  
  
  bool parse=false;  
  
  bool config=false;  
  
  // get the date and time the compiler was run  
  
  if (getDate(__DATE__) && getTime(__TIME__)) {  
  
    parse = true;  
  
    // and configure the RTC with this info  
  
    if (RTC.write(tm)) {  
  
      config = true;  
  
    }  
  
  }  
  
  Serial.begin(9600);  
  
  while (!Serial) ; // wait for Arduino Serial Monitor
```

```

delay(200);

if (parse && config) {

    Serial.print("DS1307 configured Time=");

    Serial.print(__TIME__);

    Serial.print(", Date=");

    Serial.println(__DATE__);

} else if (parse) {

    Serial.println("DS1307 Communication Error :-{");

    Serial.println("Please check your circuitry");

} else {

    Serial.print("Could not parse info from the compiler, Time=\"");

    Serial.print(__TIME__);

    Serial.print("\", Date=\"");

    Serial.print(__DATE__);

    Serial.println("\");

}

}

void loop() {

}

bool getTime(const char *str)

{

    int Hour, Min, Sec;

    if (sscanf(str, "%d:%d:%d", &Hour, &Min, &Sec) != 3) return false;

    tm.Hour = Hour;

    tm.Minute = Min;

    tm.Second = Sec;

    return true;
}

```

```

}

bool getDate(const char *str)
{
    char Month[12];
    int Day, Year;
    uint8_t monthIndex;
    if (sscanf(str, "%s %d %d", Month, &Day, &Year) != 3) return false;
    for (monthIndex = 0; monthIndex < 12; monthIndex++) {
        if (strcmp(Month, monthName[monthIndex]) == 0) break;
    }
    if (monthIndex >= 12) return false;
    tm.Day = Day;
    tm.Month = monthIndex + 1;
    tm.Year = CalendarYrToTm(Year);
    return true;
}

```

4.7 GPRS/GSM :

- Code for Active GPRS/GSM :
 - This code download on Arduino to activate GPRS/GSM modem , which use pin 3,4,5 and TX ,Rx pin .

```

Active void setup()
{
    //Init the driver pins for GSM function
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
}

```

```

//Output GSM Timing
digitalWrite(5,HIGH);

delay(1500);

digitalWrite(5,LOW);

}

void loop()

{

// Use these commands instead of the hardware switch 'UART select' in order to enable each mode

// If you want to use both GMS and GPS. enable the required one in your code and disable the other
one for each access.

digitalWrite(3,LOW);//enable GSM TX、 RX

digitalWrite(4,HIGH);//disable GPS TX、 RX

}

```

- Code for SMS :
 - This code download on Arduino to send SMS.

```

byte gsmDriverPin[3] = {
    3,4,5};//The default digital driver pins for the GSM and GPS mode

//If you want to change the digital driver pins

//or you have a conflict with D3~D5 on Arduino board,

//you can remove the J10~J12 jumpers to reconnect other driver pins for the module!

void setup()

{

//Init the driver pins for GSM function

for(int i = 0 ; i < 3; i++){

    pinMode(gsmDriverPin[i],OUTPUT);

}

digitalWrite(5,HIGH);//Output GSM Timing

```

```

delay(1500);

digitalWrite(5,LOW);

digitalWrite(3,LOW);//Enable the GSM mode

digitalWrite(4,HIGH);//Disable the GPS mode

delay(2000);

Serial.begin(9600); //set the baud rate

}

void loop()

{

  Serial.println("AT");

  delay(2000);

  //Send message

  Serial.println("AT+CMGF=1");

  delay(1000);

  Serial.println("AT+CMGS=\"15800449871\"");//Change the receiver phone number

  delay(1000);

  Serial.print("HELLO");//the message you want to send

  delay(1000);

  Serial.write(26);

  while(1);

}

```

- Code for GPRS message :
 - This code download on Arduino to send GPRS MSG.

```

void start_GSM(){

  Serial.println("AT");

  delay(2000);

```

```
Serial.println("AT+CGATT=1");

delay(2000);

Serial.println("AT+CSTT=\"Wataniya_internet\", \"\", \"\");

delay(2000);

Serial.println("AT+CIICR");

delay(2000);

Serial.println("AT+CIPSTART=\"TCP\", \"195.3.191.27\", \"12349\");

delay(2000);

Serial.println("AT+CIPSEND"); // start sending ..

    Serial.println("\r");

    delay(2000);

    // the data to be sent.

    Serial.print("Rami"); // the letter "P" indicates for Police. the server use it to distinguish
between three cases.

    delay(2000);

    Serial.write(26);

}
```

4.8 Message Format :

Figure 4.31 represent the message format that send from node to pc server .

- Date : 19 bits (b1-b19) .
- Node Number : 3 bits (b20-b22).
- Current value of phase 1 : 7 bits (b23-b29).
- Current value of phase 2 : 7 bits (b30-b36).
- Current value of phase 3 : 7 bits (b37-b43).
- Current value of core phase : 7 bits (b44-50).
- Alarm Type : 2 bits (b51-b52).
- Temperature : 4 bits (b53-b56).
- Humidity : 4 bits (b57-b60).
- Password : 9 bits (b61-b69).

b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	Date
y	y	y	y	-	m	m	-	d	d	
b11	b12	b13	b14	b15	b16	b17	b18	b19		
	H	H	:	M	M	:	S	S		
b20	b21	b22								
int	int	int								
b23	b24	b25	b26	b27	b28	b29				
int	int	int	int	int	int	int				
b30	b31	b32	b33	b34	b35	b36				
int	int	int	int	int	int	int				
b37	b38	b39	b40	b41	b42	b43				
int	int	int	int	int	int	int				
b44	b45	b46	b47	b48	b49	b50				
int	int	int	int	int	int	int				
b51	b52									
int	int									
b53	b54	b55	b56							
int	int	int	int							
b57	b58	b59	b60							
int	int	int	int							
b61	b62	b63	b64	b65	b66	b67	b68	b69		
char	char	char	char	char	char	char	char	char		

Node Num

I(ph1)

I(ph2)

I(ph3)

I(core)

Alarm Type

Temperature

Humidity

Password

figure 4.31 message format

4.9 web server

The figure 4.32 show the web server which represent the all reading in data base in the server .

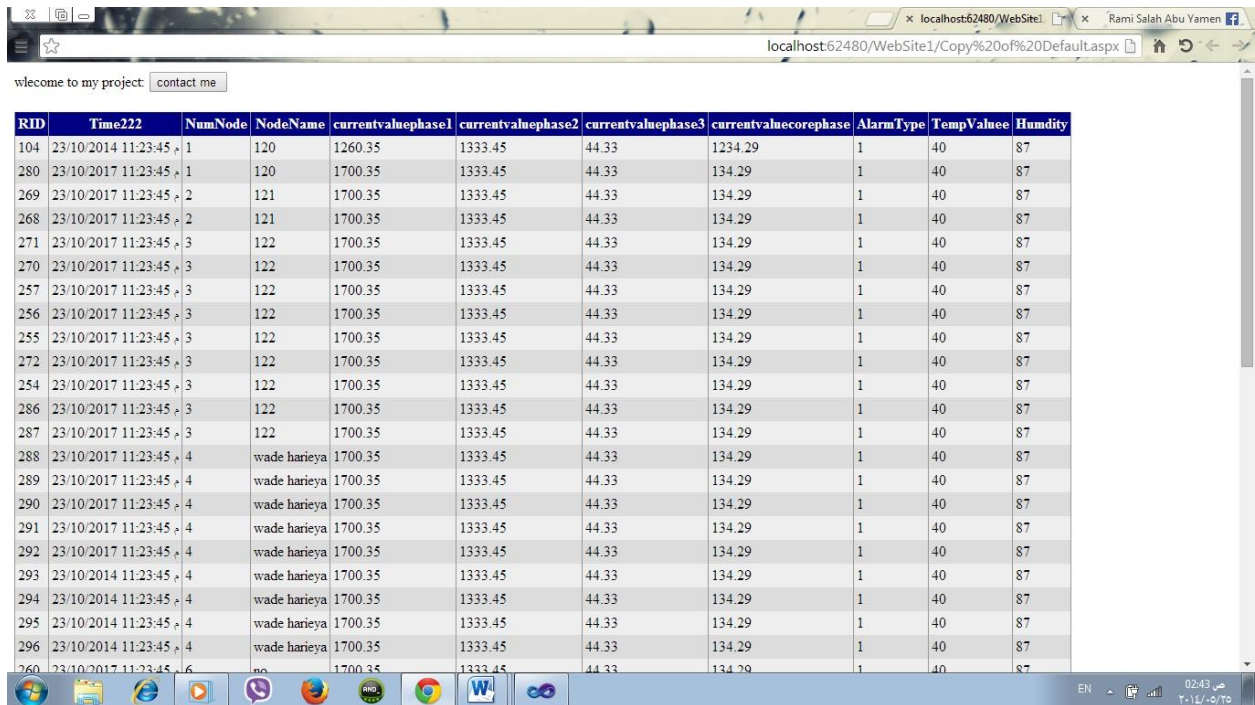


figure 4.32 Print Screen of Web Server

5

Chapter Five

Testing And Result

5.1 Introduction

5.2 System Testing

5.3 Testing results

5.1 Introduction :

the final stage to complete the project is to test the system to get result and measure its performance .

this chapter show all measurements needed to evaluate the performance of this system .

5.2 System testing :

Checking the testing of the DHT11 ,MQ-2 ,RTC, AC to DC circuit converter , Arduino measurement voltage , multiple serial communication , SMS , GPRS .

5.2.1 DHT11 (Humidity-Temperature Sensor) :

DHT11 has been implemented and tested if it gives the result as the same as other Temperature sensor and humidity sensor .

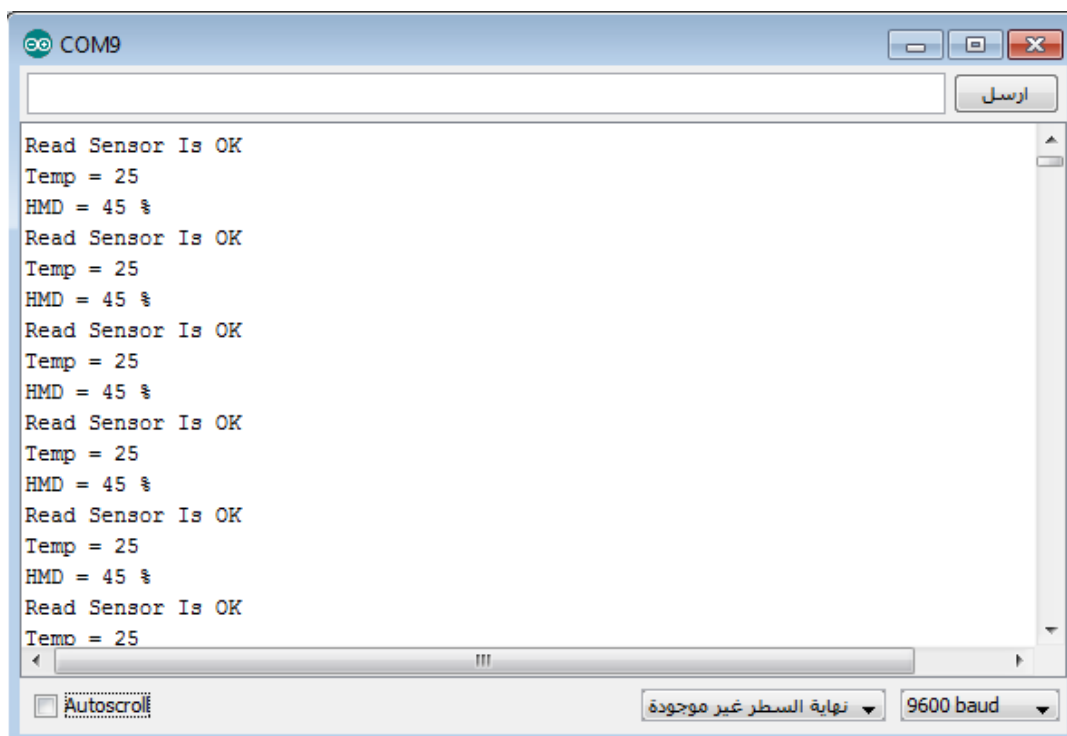
The Temperature on the other sensor = 27 c.

The humidity on the other sensor = 44%.

As shown in the Figure 5.1 the result of DHT11 is :

The Temperature on the other sensor = 25 c.

The humidity on the other sensor = 45%.



```
COM9
ارسل
Read Sensor Is OK
Temp = 25
HMD = 45 %
Read Sensor Is OK
Temp = 25
HMD = 45 %
Read Sensor Is OK
Temp = 25
HMD = 45 %
Read Sensor Is OK
Temp = 25
HMD = 45 %
Read Sensor Is OK
Temp = 25
HMD = 45 %
Autoscroll
نهاية السطر غير موجودة
9600 baud
```

Figure 5.1 the result of DHT11

5.2.2 MQ_2 (smoke detector) :

MQ_2 has been implemented and tested if it gives the Alarm when the smoke spreads , and when I make a smoke ,MQ-2 give alarm as shown in figure 5.2.

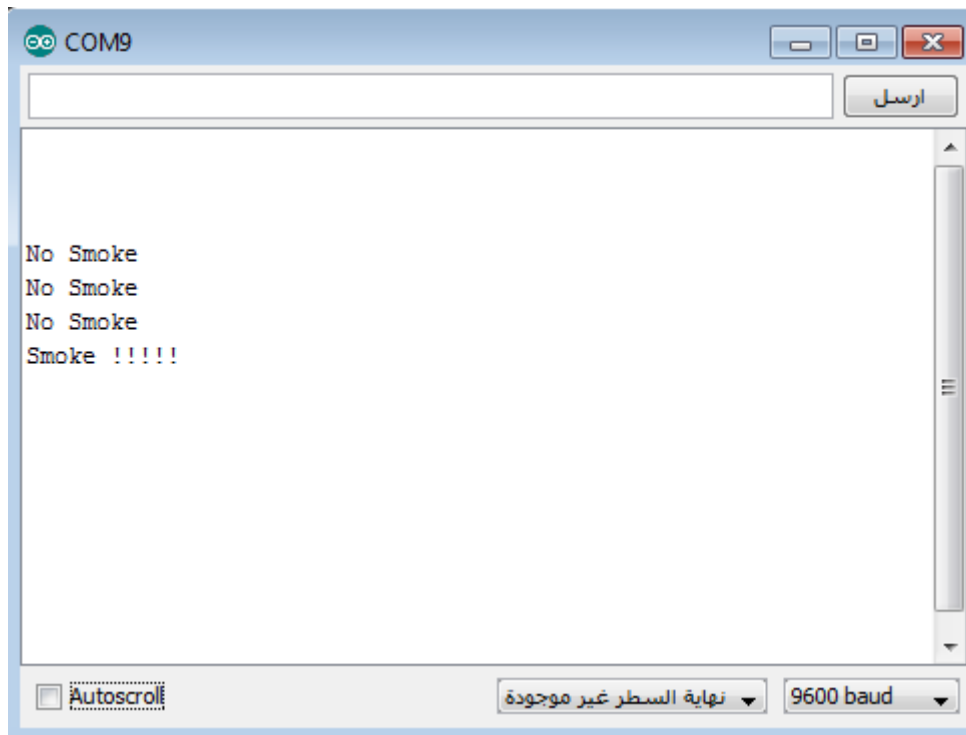


figure 5.2 the result of smoke detector

5.2.3 RTC :

RTC has been implemented and tested if it gives a real and true data and time, as shown in figure 5.3 it get same time .

```
COM11
-----
DS1307RTC Read Test
-----
Ok, Time = 00:06:36, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:37, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:38, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:39, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:40, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:41, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:42, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:43, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:44, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:45, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:46, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:47, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:48, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:49, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:50, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:51, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:52, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:53, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:54, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:55, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:56, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:57, Date (D/M/Y) = 20/5/2014
Ok, Time = 00:06:58, Date (D/M/Y) = 20/5/2014
```

figure 5.3 the result of RTC

5.2.4 AC to DC circuit converter:

AC to DC circuit has been implemented on pcb and tested it. When I connect to it sine wave (2 Vp_p) it give 1 volt DC (VP-P /4) .

AC : Analog Current .

DC: Digital Current .

VP-P: voltage Peak to Peak .

5.2.5 Arduino measurement voltage :

Arduino measurement voltage has been implemented and tested if it gives the result as the voltmeter .

And I connect to voltmeter 2.7 volt DC and voltmeter read it 2.7 Volt ,when I connect it to Arduino in ADC 0 ,the Arduino give a result 2.68 V as shown in figure 5.4.

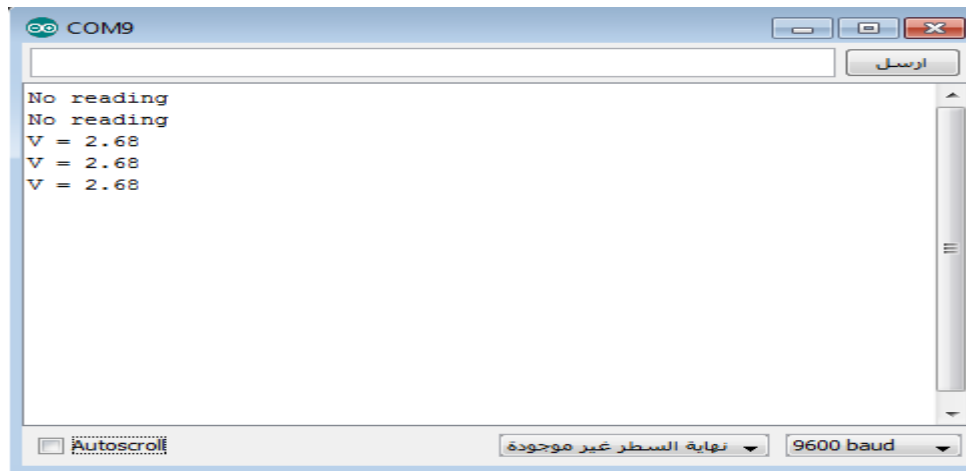


figure 5.4 the result of Arduino measurement voltage

5.2.6 Multiple serial communication :

Arduino mater has been implemented and tested if it can management the communication serial between 4 Arduino, and it pass in it as shown in figure 5.5 .

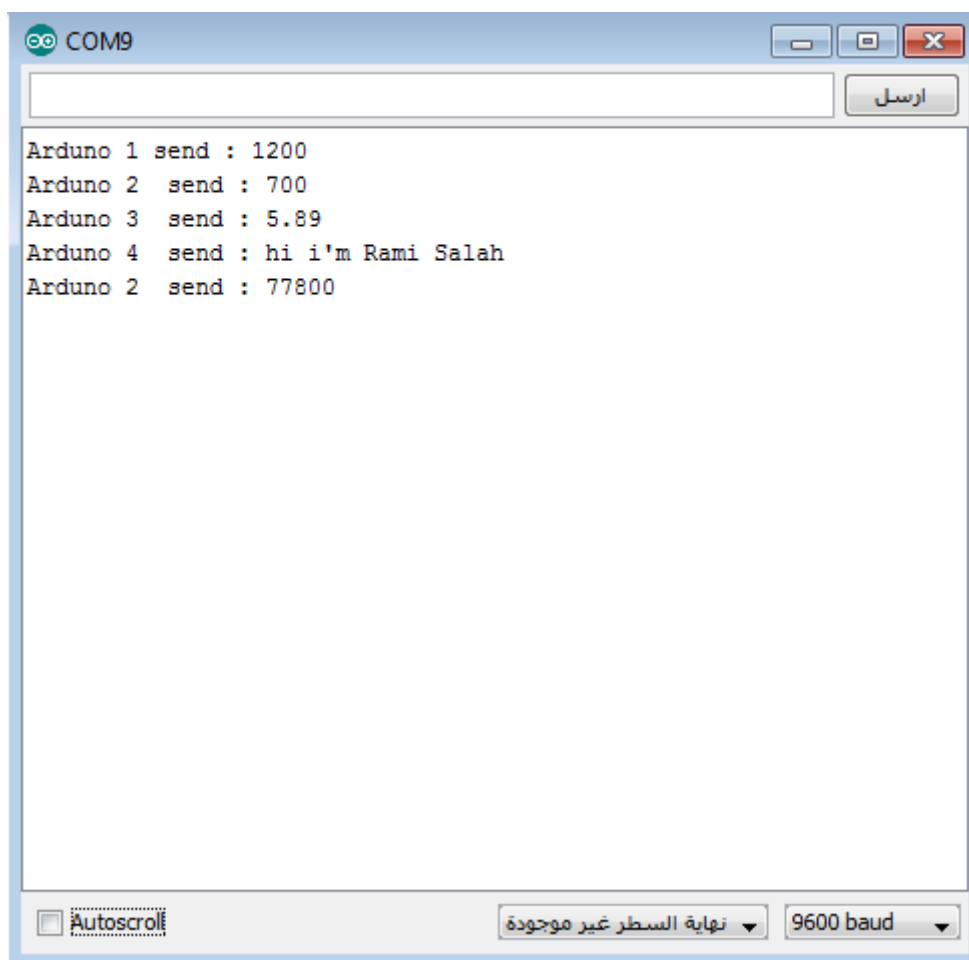


figure 5.5 the result of Arduino multiple serial

5.2.7 SMS communication :

Arduino master has been implemented and tested if it can send, I sent " the current of phase 1 in wade harieh =1200.34 " as shown in figure 5.6 ,and I received the same message as shown in figure 5.7

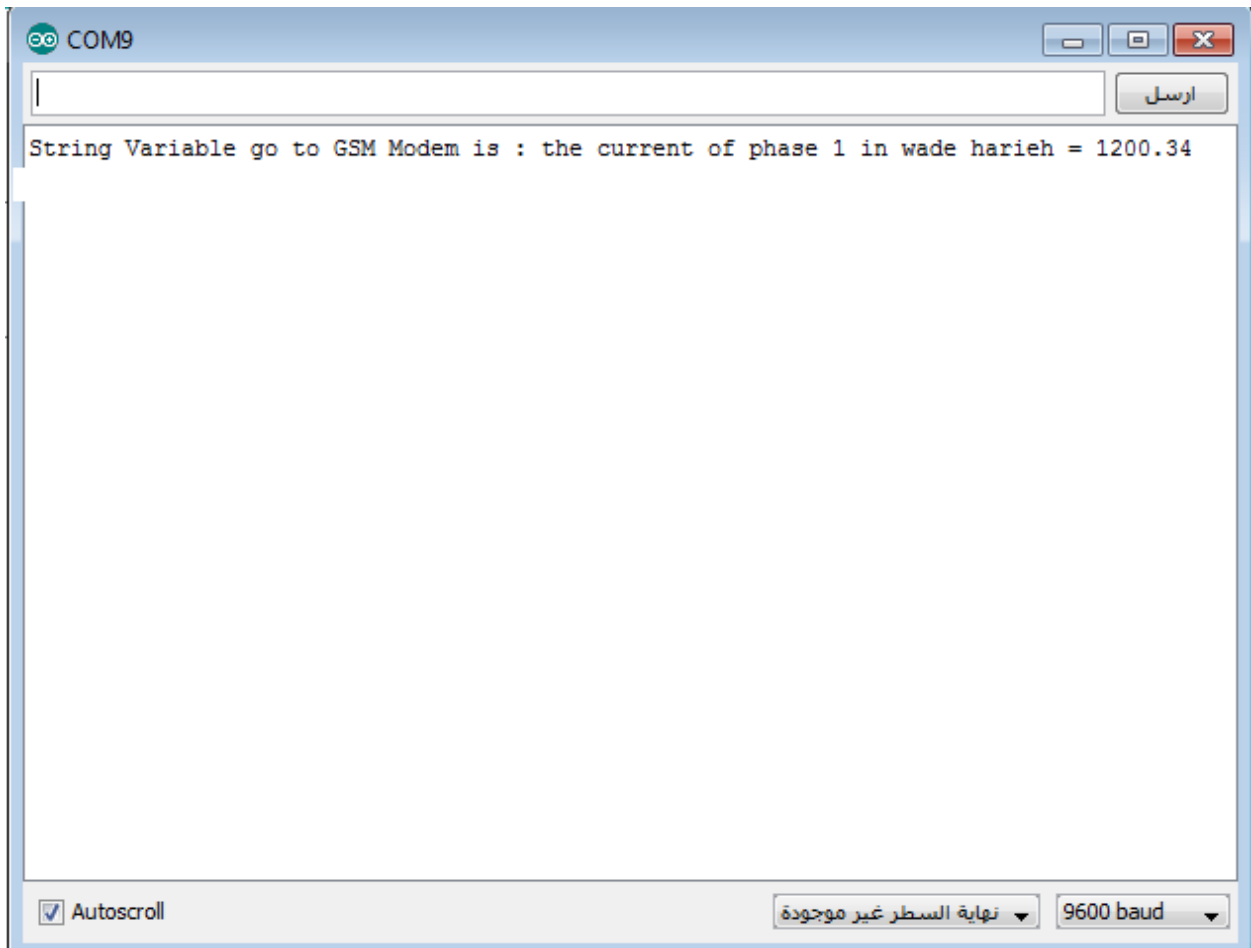


figure 5.6 send SMS of Arduino to any person

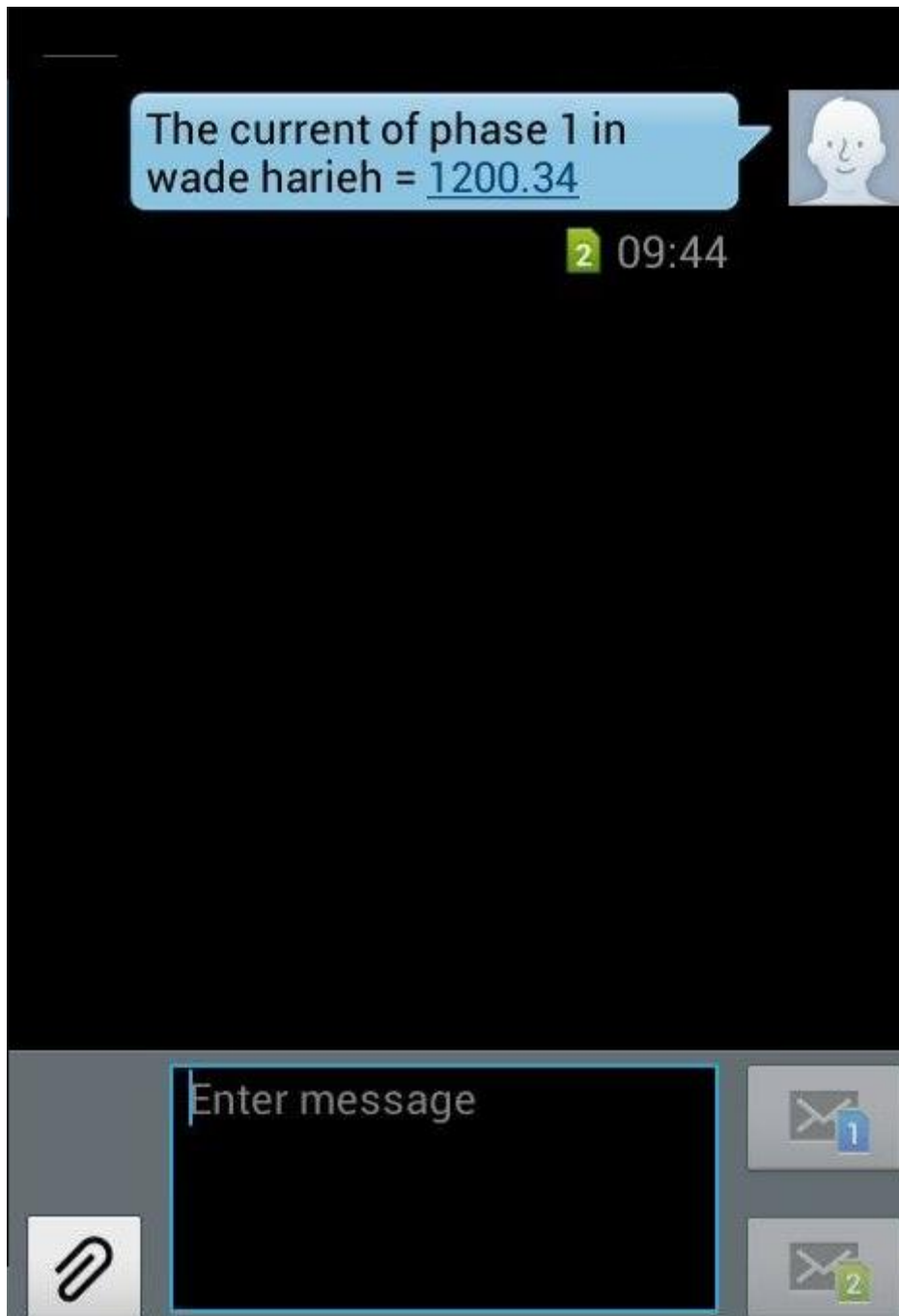


figure 5.7 received SMS on mobile

5.2.8 GPRS communication :

Arduino master has been implemented and tested if it can send , I sent " the current of phase 1 in wade hariéh =1200.34" as shown in figure 5.8 ,and I received the same message in socket server in net Beans as shown in figure 5.9 .

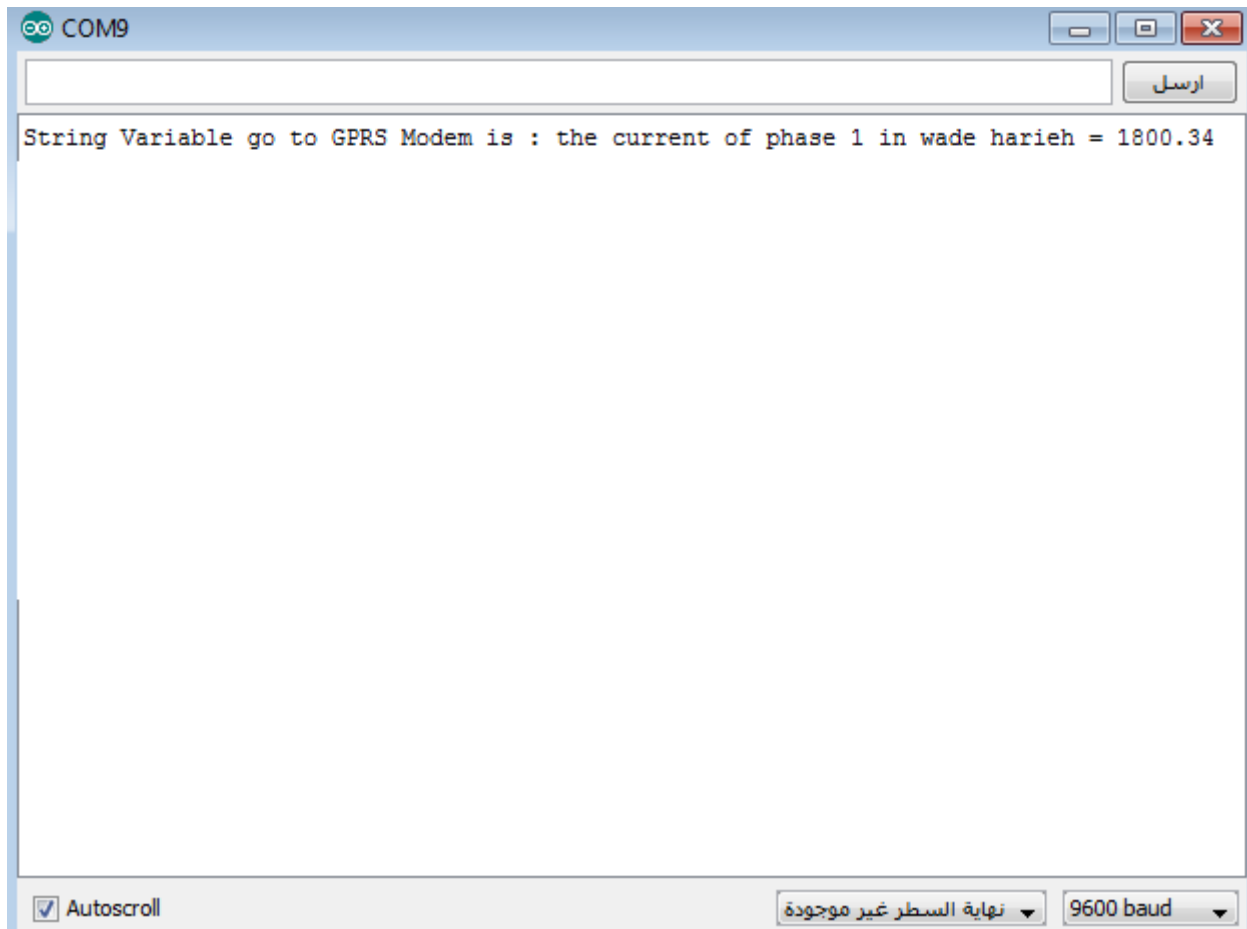


figure 5.8 send GPRS MSG of Arduino to public ip

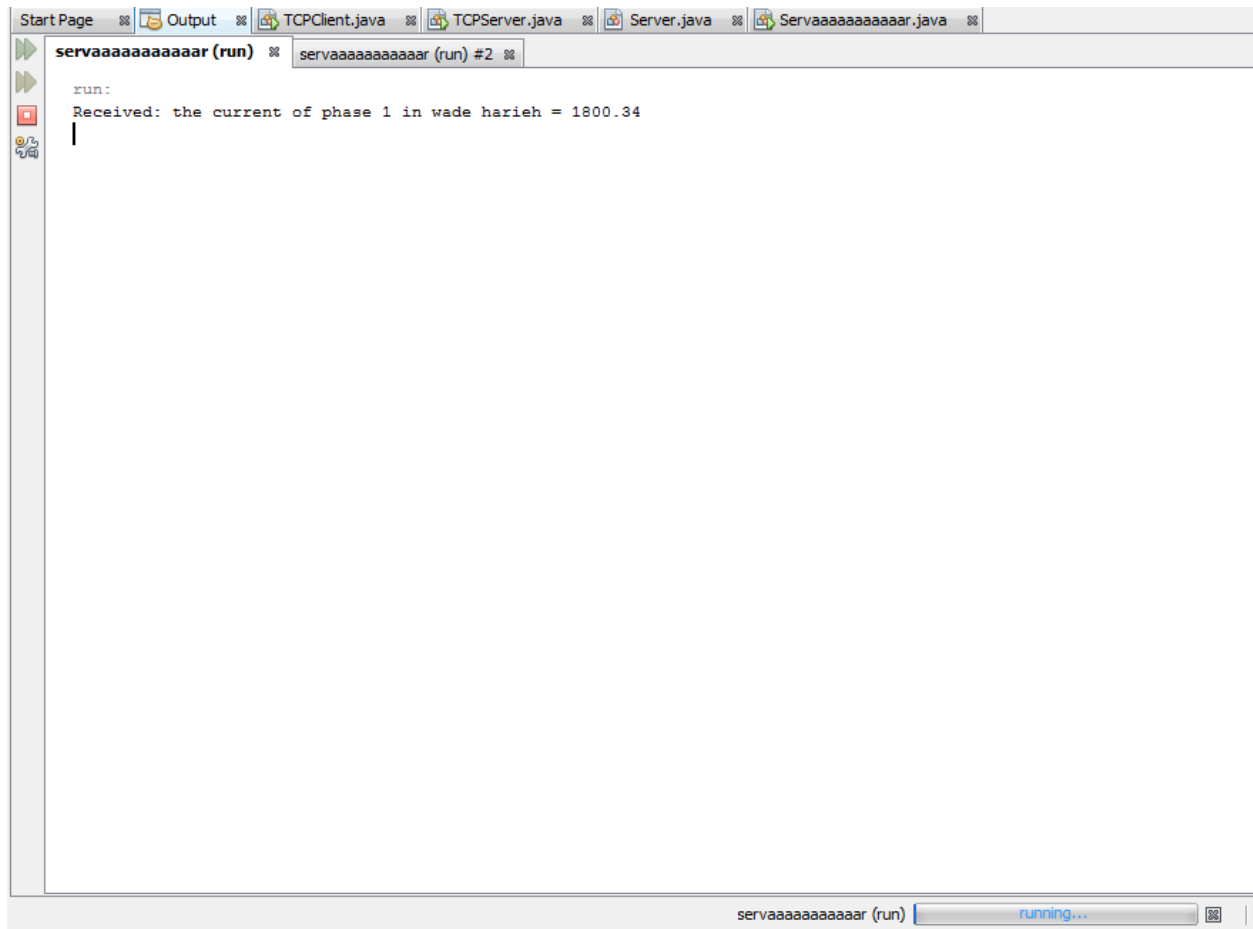


figure 5.9 received GPRS MSG in socket server

5.3 Testing Results :

* we connected 4 function generator to our project as 3 phase and core phase , and generated on phase 2 overload current above threshold of safety and make the current as the following at 2014/05/25 10:29:32 and the Node Number is 102 :

I(phase 1) = 3 A.

I (phase 2)= 11 A .

I(phase 3)= 2.5 A.

I(core phase) =1.5 A.

Temperature = 22 C.

Humidity =36 % .

The result is :

- Send SMS to Mobile as shown in figure 5.10



figure 5.10 SMS MSG Alarm

- Send GPRS MSG to server as shown in figure 5.11 .

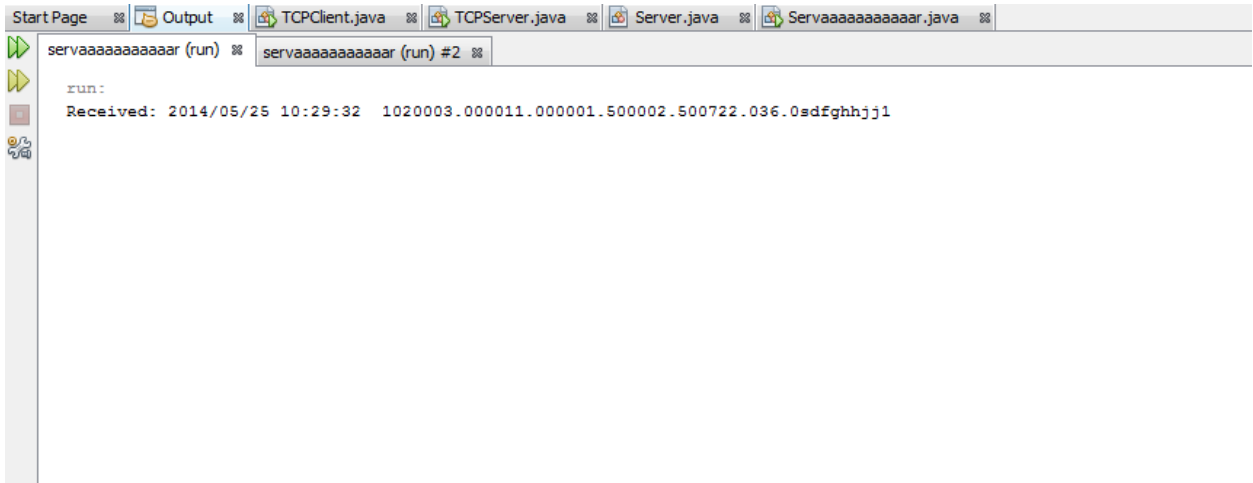


figure 5.11 GPRS MSG receiving on server

- Change color of key map to red as shown in figure 5.12 .

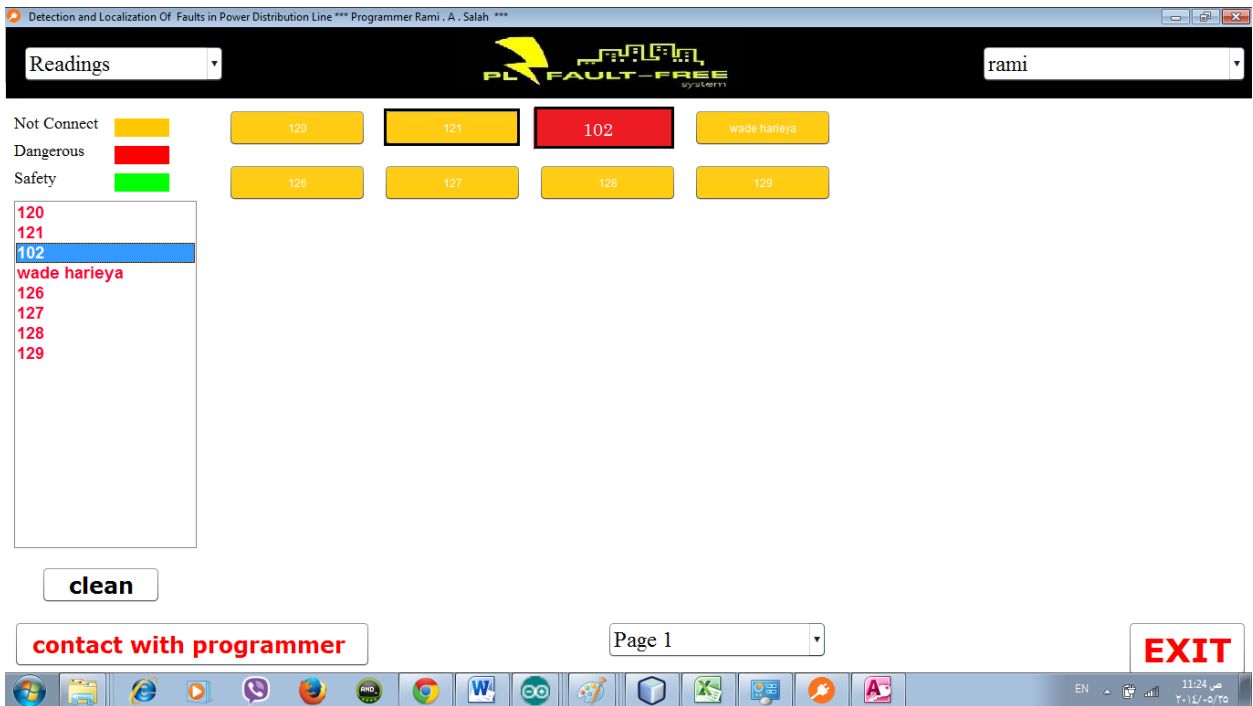


figure 5.12 representation danger state

- Insert Reading in Data Base as shown in figure 5.13 .

currentvaluephase2	currentvaluephase3	currentvaluecorephase	AlarmType	TempValuee	Humdity	انقر لإضافة
1333.45	44.33	1234.29	41	340	87	
1333.45	44.33	1234.29	13	340	87	
1333.45	44.33	1234.29	12	340	87	
66.33	22.134	12.23	41	-13	89	
11	2.5	1.5	7	22	36	
*						

figure 5.13 insert reading in database

6

Chapter Six

Outcomes and Future work

5.1 Introduction

6.2 Future work

6.3 Acquired learning outcomes.

6.1 Introduction

This chapter describes the learning outcomes that have been acquired during the work on the project, suggestions for future work and development.

6.2 Future work

- 1- Anti-theft of electricity .
- 2- control remotely in the node from server .

6.3 Acquired learning outcomes.

- 1- learning java .
- 2- learning database .
- 3-learning web Server Design .
- 4-learning programming Arduino .
- 5-learning programming PIC .
- 6-learning programming GPRS Modem .
- 7-learning reading data sheet of IC .

APPENDIX

DATA SHEETS

For

DHT11 Humidity & Temperature Sensor

Gas sensor (MQ2)

Arduino mega

Arduino UNO

GPS/GPRS/GSM Module V3.0

RTC tiny l2c

DHT11 Humidity & Temperature Sensor

DHT11 - Humidity and Temperature Sensor

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.

Features

- ☐ Full range temperature compensated
- ☐ Relative humidity and temperature measurement
- ☐ Calibrated digital signal
- ☐ Outstanding long-term stability
- ☐ Extra components not needed
- ☐ Long transmission distance
- ☐ Low power consumption
- ☑ 4 pins packaged and fully interchangeable

Details

This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration.

The

calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process.

The single-wire serial interface makes system integration quick and easy. Its small size, low power

consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package.

Sunrom Technologies Your Source for Embedded Systems Visit us at www.sunrom.com

Specifications

Sunrom Technologies Your Source for Embedded Systems Visit us at www.sunrom.com

Typical Application

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

SDK (Software Development Kit)

Download source code + project articles by clicking following link

<http://www.sunrom.com/files/3732.zip>

It contains details for AVR, PIC and Arduino projects.

Sunrom Technologies Your Source for Embedded Systems Visit us at www.sunrom.com

Communication Process: Serial Interface (Single-Wire Two-Way)

The interesting thing in this module is the protocol that uses to transfer data. All the sensor readings are sent using a single wire bus which reduces the cost and extends the distance. In order to send data over a bus you have to describe the way the data will be transferred, so that transmitter and receiver can understand what says each other. This is what a protocol does. It describes the way the data are transmitted. On DHT-11 the 1-wire data bus is pulled up with a resistor to VCC. So if nothing is occurred the voltage on the bus is equal to VCC. Communication Format can be separated into three stages

- 1) Request
- 2) Response
- 3) Data Reading

1) Request: To make the DHT-11 to send you the sensor readings you have to send it a request. The request is, to pull down the bus for more than **18ms** in order to give DHT time to understand it and then pull it up for **40uS**.

2) Response: What comes after the request is the DHT-11 response. This is an automatic reply from DHT which indicates that DHT received your request. The response is ~54uS low and 80uS high.

3) Data Reading: What will come after the response is the sensor data. The data will be packed in a packet of 5 segments of 8-bits each. Totally $5 \times 8 = 40$ bits.

First two segments are Humidity read, integral & decimal. Following two are Temperature read in

Celsius, integral & decimal and the last segment is the Check Sum which is the sum of the 4 first Sunrom Technologies Your Source for Embedded Systems Visit us at www.sunrom.com segments. If Check Sum's value isn't the same as the sum of the first 4 segments that means that data received isn't correct.

How to Identify Bits: Each bit sent is a follow of ~54uS Low in the bus and ~24uS to 70uS High depending on the value of the bit.

Bit '0' : ~54uS Low and ~24uS High

Bit '1' : ~54uS Low and ~70uS High

End Of Frame: At the end of packet DHT sends a ~54uS Low level, pulls the bus to High and goes to sleep mode.

Sunrom Technologies Your Source for Embedded Systems

Logic Analyzer Snapshots: In the following image you can see the request sent from the MCU to the DHT and following the packet. Because about 20mS and packet received is in uS we can't view the data bits view.

If we zoom at the data bits we can read the values. You can see after the Request follows the Response, and Data bits. I have drawn so

If we decode the above data we have.

Humidity 0b00101011.0b00000000 = 43.0%

Temperature 0b00010111 = 23 C.

The last two segments can't be seen in this image because of zoom.

Implementation:

What we have to do to read a DHT-11 sensor is:

- 1) Send request

2) Read response

3) Read each data segment and save it to a buffer

4) Sum the segments and check if the result is the same as CheckSum

If the CheckSum is correct, the values are correct so we can use them. If CheckSum is wrong we discard the packet.

To read the data bits can use a counter and start count uSeconds of High level. For counts > 24uS we replace with bit '1'. For counts <=24 we replace with bit'0'

Visit us at www.sunrom.com

the request has very long duration as you can see is

So it is expanded in next

some color notes to be more understandable.

(43 is integral part and .0 is decimal part)

)

Sunrom Technologies Your Source for Embedded Systems Visit us at www.sunrom.com

Dimensions (mm)

Gas sensor (MQ2)

HANWEI ELETRONICS CO.,LTD MQ-2 <http://www.hwsensor.com>

TEL: 86-371- 67169070 67169080 FAX: 86-371-67169090 E-mail: sales@hwsensor.com

TECHNICAL DATA MQ-2 GAS SENSOR

FEATURES

Wide detecting scope Fast response and High sensitivity

Stable and long life Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting

of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

SPECIFICATIONS

A. Standard work condition

Symbol Parameter name Technical condition Remarks

Vc Circuit voltage $5V \pm 0.1$ AC OR DC

VH Heating voltage $5V \pm 0.1$ AC OR DC

RL Load resistance can adjust

RH Heater resistance $33\Omega \cdot \pm 5\%$ Room Tem

PH Heating consumption less than 800mw

B. Environment condition

Symbol Parameter name Technical condition Remarks

Tao Using Tem $-20^{\circ}\text{C}-50^{\circ}\text{C}$

Tas Storage Tem $-20^{\circ}\text{C}-70^{\circ}\text{C}$

RH Related humidity less than 95%Rh

O2 Oxygen concentration 21%(standard condition)Oxygen concentration can affect sensitivity

minimum value is

over 2%

C. Sensitivity characteristic

Symbol Parameter name Technical parameter Remarks

Rs Sensing

Resistance

$3K\Omega-30K\Omega$

(1000ppm iso-butane)

α

(3000/1000)

isobutane

Concentration

Slope rate ≤ 0.6

Standard

Detecting

Condition

Temp: $20^{\circ}\text{C} \cdot \pm 2^{\circ}\text{C}$ Vc: $5V \pm 0.1$

Humidity: $65\% \pm 5\%$ Vh: $5V \pm 0.1$

Preheat time Over 24 hour

Detecting concentration

scope :

200ppm-5000ppm

LPG and propane

300ppm-5000ppm
butane
5000ppm-20000ppm
methane
300ppm-5000ppm H₂
100ppm-2000ppm
Alcohol

D. Structure and configuration, basic measuring circuit

Structure and configuration of MQ-2 gas sensor is shown as Fig. 1 (Configuration **A** or **B**), sensor composed by

micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a

Parts Materials

- 1 Gas sensing layer
SnO₂
- 2 Electrode Au
- 3 Electrode line Pt
- 4 Heater coil Ni-Cr alloy
- 5 Tubular ceramic Al₂O₃
- 6 Anti-explosion network
Stainless steel gauze
(SUS316 100-mesh)
- 7 Clamp ring Copper plating Ni
- 8 Resin base Bakelite
- 9 Tube Pin Copper plating Ni

A向A向

Fig. 1

Configuration **A** Configuration **B**

Fig.2

HANWEI ELETRONICS CO.,LTD MQ-2 <http://www.hwsensor.com>

TEL: 86-371- 67169070 67169080 FAX: 86-371-67169090 E-mail: sales@hwsensor.com

crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of

sensitive components. The enveloped MQ-2 have 6 pin ,4 of them are used to fetch signals, and other 2 are used

for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-2

SENSITIVITY ADJUSTMENT

Resistance value of MQ-2 is difference to various kinds and various concentration gases.

So,When using

this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for

1000ppm liquified petroleum gas<LPG>,or 1000ppm iso-butane<i-C₄H₁₀>concentration in air and use value of

Load resistance that(RL) about 20 KΩ(5KΩ to 47 KΩ).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

MQ-2

0.1

1

10

100 1000 10000

ppm

Rs/Ro

H2

LPG

CH4

CO

alcohol

smoke

propane

air

0

0.2

0.4

0.6

0.8

1

1.2

1.4

1.6

1.8

-20 -10 0 10 20 30 40 50 60

degree

Rs/Ro

33%RH

85%RH

Fig.3 is shows the typical sensitivity characteristics of the MQ-2 for several gases.

in their: Temp: 20°C、

Humidity: 65%、

O2 concentration 21%

RL=5kΩ

Ro: sensor resistance at 1000ppm of H2 in the clean air.

Rs:sensor resistance at various concentrations of gases.

Fig.4 is shows the typical dependence of the MQ-2 on temperature and humidity.

Ro: sensor resistance at 1000ppm of H2 in air at 33%RH and 20 degree.

Rs: sensor resistance at 1000ppm of H2

at different temperatures Fig.4 and humidities.

Arduino mega

The Arduino Mega is a microcontroller board based on the ATmega1280 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 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 Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: [arduino-mega-reference-design.zip](#)

Schematic: [arduino-mega-schematic.pdf](#)

Summary

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega 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.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- GND. Ground pins.

Memory

The ATmega1280 has 128 KB of flash memory for storing code (of which 4 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- I²C: 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega1280 provides four hardware UARTs for TTL (5V) serial communication. An FTDI FT232RL on the board channels one of these over USB and the [FTDI drivers](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega1280 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega1280 datasheet.

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega1280 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega1280 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). 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.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](#).

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (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	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

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.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of

the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the [SPI library](#).
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function.

Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available .

The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

GPS/GPRS/GSM Module V3.0

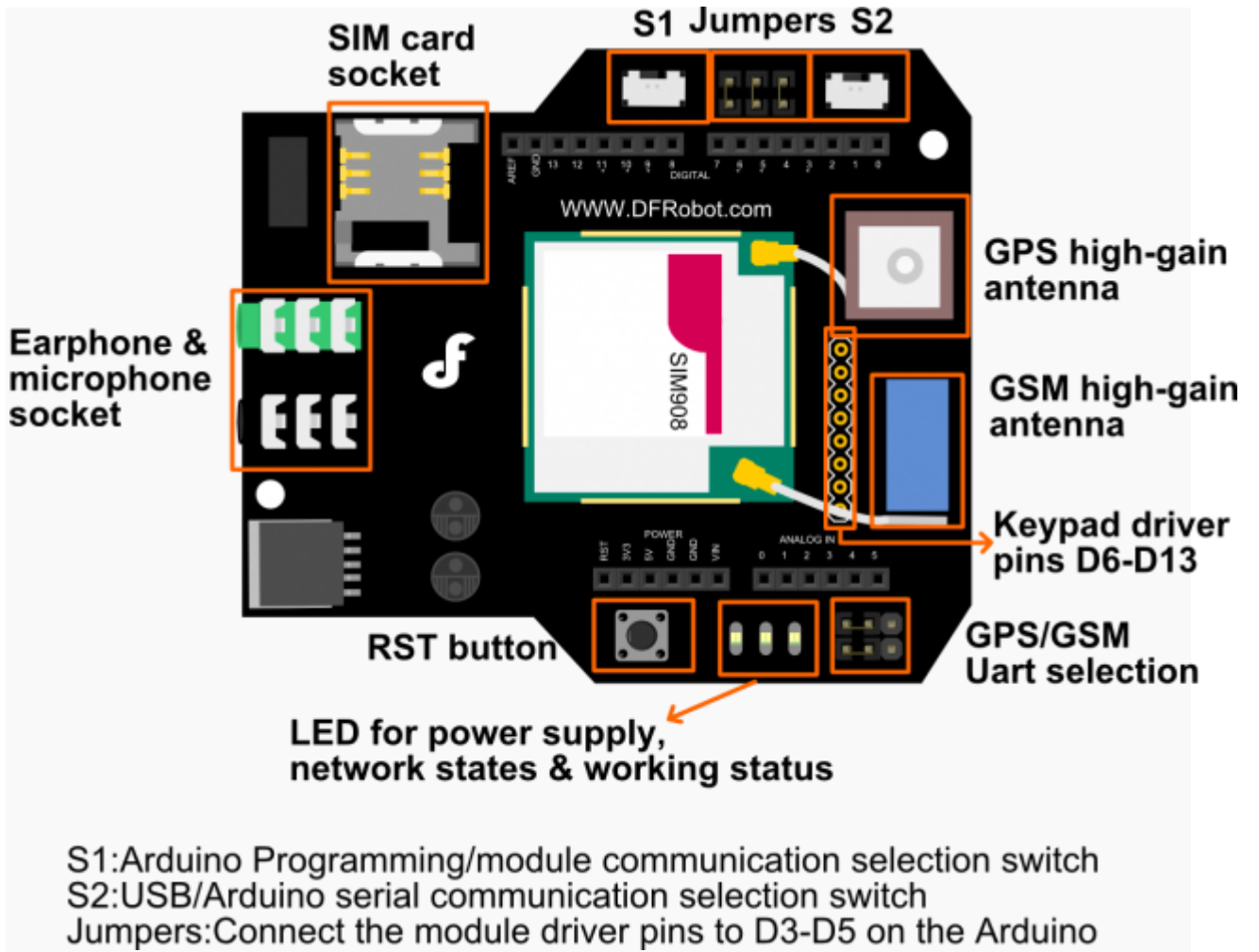
This is a GPS/GPRS/GSM shield from DFRobot. This shield with a Quad-band GSM/GPRS engine works on frequencies EGSM 900MHz/DCS 1800MHz and GSM850 MHz/PCS 1900MHz. It also supports GPS technology for satellite navigation. It's possible for your robot and control system to send messages and use the GSM network.

It is controlled via AT commands(GSM07.07 ,07.05 and SIMCOM enhanced AT Commands). And the design of this shield allows you to drive the GSM & GPS function directly with the computer and the Arduino Board. It includes a high-gain SMD antenna for GPS & GSM.

This GPS/GPRS/GSM shield uses an embedded SIM908 chip from SIMCom. Featuring an industry-standard interface and GPS function, the combination of both technologies allows goods, vehicles and people to be tracked seamlessly at any location and anytime with signal coverage.

Specification

- Power supply: 6-12v
- Low power consumption (100mA@7v - GSM mode)
- Quad-Band 850/900/1800/1900MHz
- GPRS multi-slot class 10
- Support GPS technology for satellite navigation
- Embeded high-gain SMD antennas for GPS & GSM
- Directly support 4*4 button pad
- USB/Arduino control switch
- Board Surface:Immersion Gold
- Size: 81x70mm



RTC TINY L2C

DESCRIPTION:

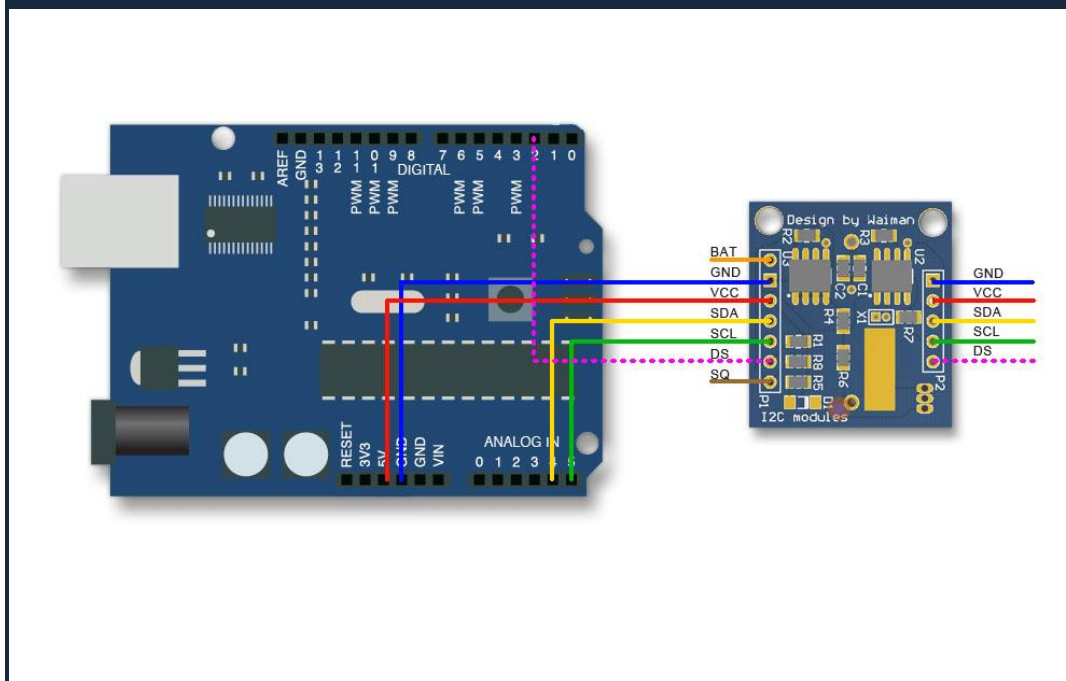
Arduino Tiny RTC I2C Real Time Clock

This Arduino Tiny RTC I2C module incorporates the [DS1307](#) I2C real time clock IC and the [24C32](#) 32K I2C EEPROM storage. What's more, it has a [DS18B20](#) temperature sensor on board. All of this in a tiny package of 25mm x 28mm x 8.4mm. It comes with a LIR2303 rechargeable lithium battery, and a charging circuit is included in the module. When the temperature sensor is off, the RTC module can run for 1 year on a single charge.

Specs of DS18B20 Temperature Sensor

- 3.0-5.5V input voltage
- Waterproof
- -55°C to +125°C temperature range
- ±0.5°C accuracy from -10°C to +85°C
- 1 Wire interface

Wiring the RTC to Arduino



The most useful pins are duplicated from P1 to P2. If needed, the pin "BAT" can be fed into an ADC pin for monitoring the battery voltage. The pin-outs are explained below. Connecting "VCC" to 5 V will trickle charge the onboard battery.

Arduino Tiny RTC I2C Real Time Clock Pinout

PIN	Description	Comment
BAT	Battery voltage	To monitor the battery

		voltage, or not connected
GND	Ground	Ground
VCC	5V supply	Power the module and charge the battery
SDA	I2C data	I2C data for the RTC
SCL	I2C clock	I2C clock for the RTC
DS	DS18B20 Temp. Sensor output	One wire interface
SQ	Square wave output	Normally not used

The I2C wires "SDA" and "SCL" are the data line and clock line, they should be connected to the corresponding pins depending on the Arduino board.

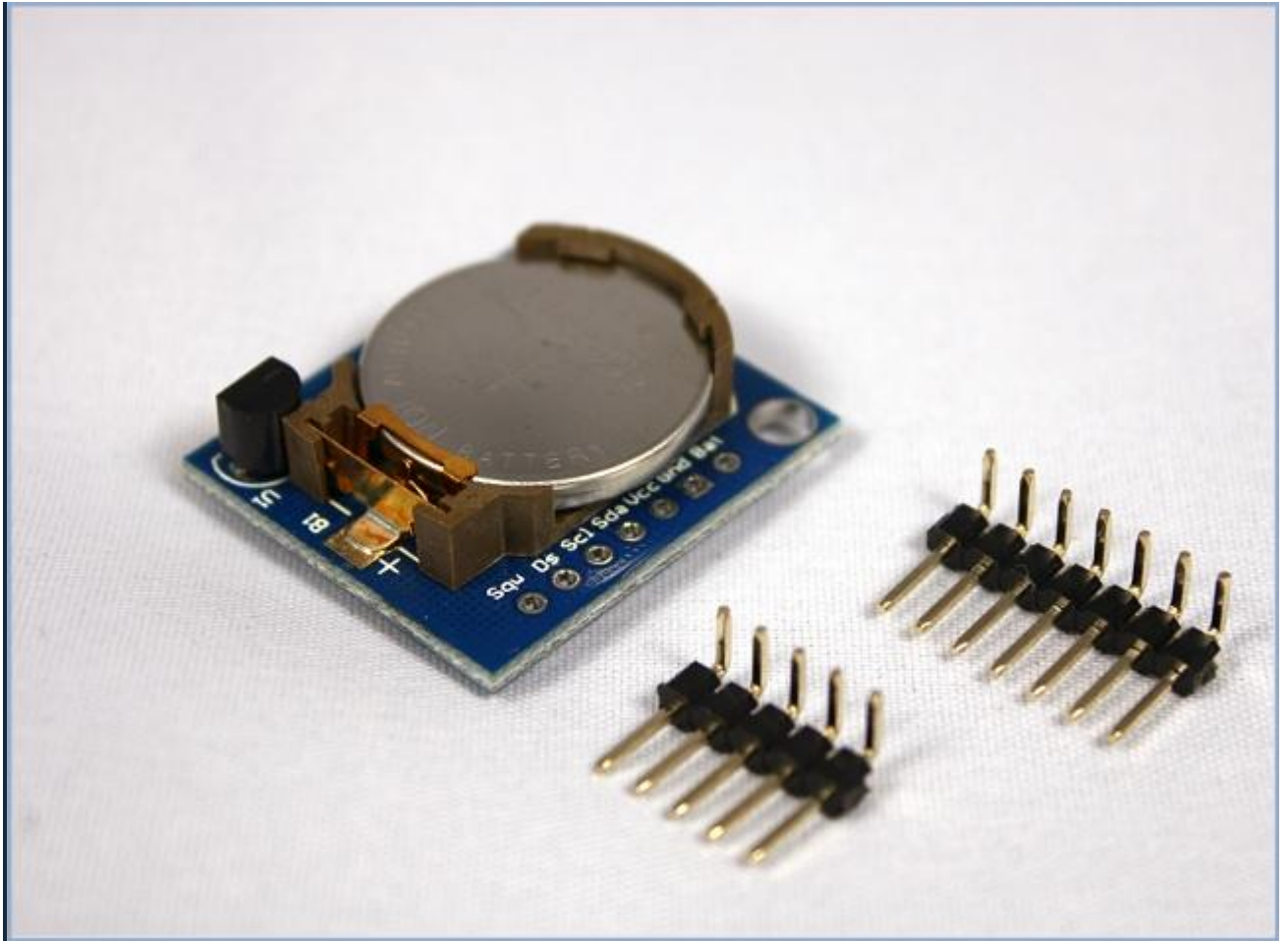
Board	I2C / TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

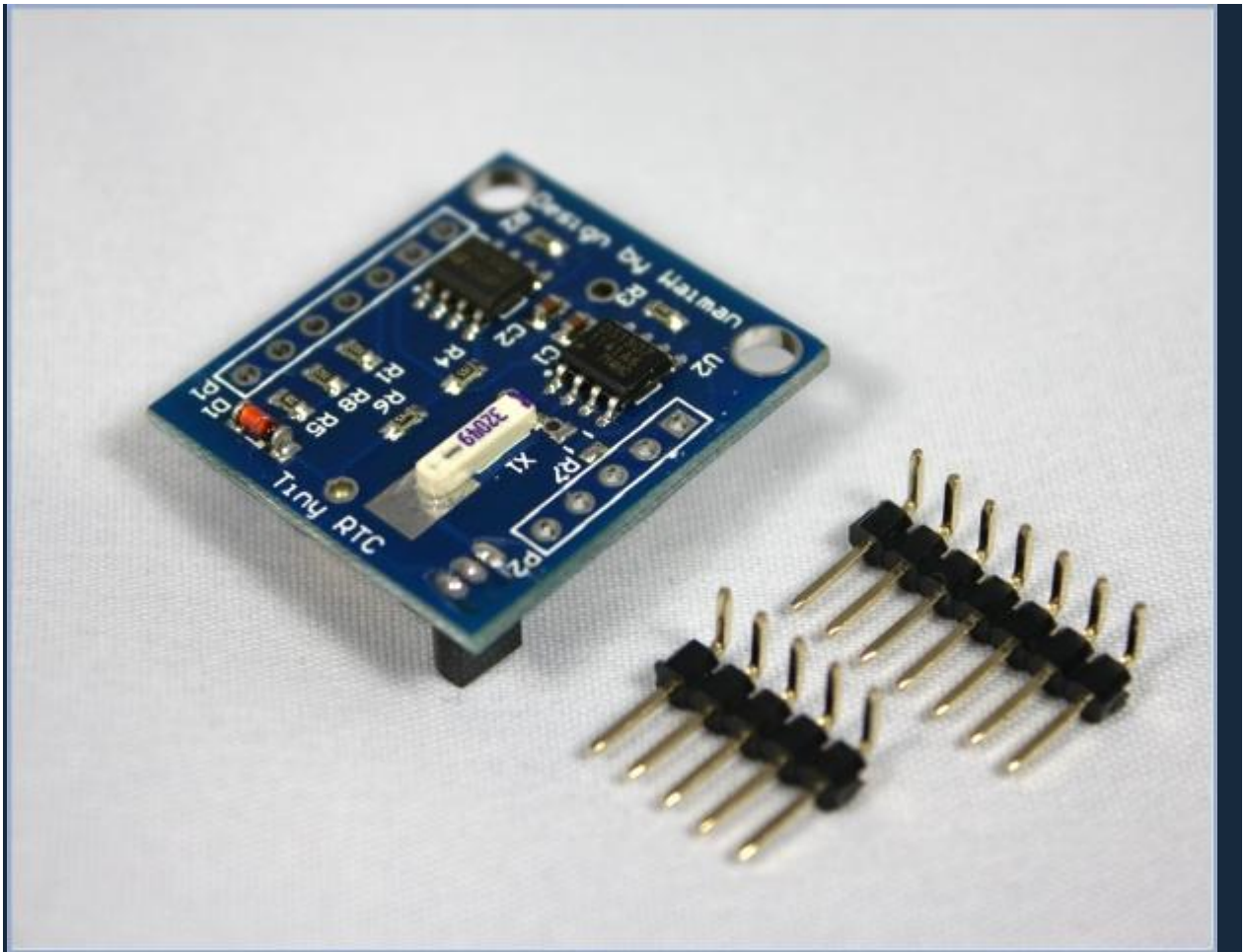
[Example code is available here.](#)

For more detail, please visit the [Arduino Wire Library Schematics](#)

The schematics of this RTC module can be downloaded [here](#).

IMAGE:





CATALOG:

References

- ❖ [1] Alauddin Al-omary, Wael El-medany and Sufyan Al-Irhayim, Design and implementation of secure low cost AMR system using GPRS technology, international conference on telecommunication technology and applications, Singapore, 2011.
- ❖ [2] Raian Ali, Fabiano Dalpiaz, Paolo Giorgini, Location-Based Software Modeling and Analysis: Tropos-Based Approach, 2008.
- ❖ [3] Stuart A. Boyer, Supervisory Control And Data Acquisition 4th, June 15, 2009.
- ❖ [4] Murari Mohan Saha Ratan Das Pekka Verho Damir Novosel, Power Systems and Communications Infrastructures for the future, Beijing, September 2002.
- ❖ [5] International Journal of Emerging Technology and Advanced Engineering, December 2012, available at www.ijetae.com, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 2, Issue 12.
- ❖ [6] McLaren, P.G.; Dirks, E.; Jayasinghe, R.P.; Parker, A.D., A current transformer model based on the Jiles-Atherton theory of ferromagnetic hysteresis.
- ❖ [7] Alexander L. Schulz, Capacitors: Theory Types and Applications, Nova Science Publishers, Incorporated, Jul 1, 2010.
- ❖ [8] Edwin Fitch Northrup, Methods of Measuring Electrical Resistance, McGraw-Hill book Company, 1912.
- ❖ [9] Joachim Tisal, GSM cellular radio telephony, Wiley, 1997.
- ❖ [10] Regis "Bud" J. Bates, GPRS: General Packet Radio Service (Professional Telecom) November 12, 2001.
- ❖ [11] Spreadtrum, SM5100B-D GSM/GPRS Module Specification, 2006-4-4.
- ❖ [12] Steve Suehring, JavaScript Step by Step, 3rd Edition, Microsoft Press 2013.