PPU College Of Engineering and Technology

The Home of Competent Engineers and Researchers

COLLEGE OF ENGINEERING & TECHNOLOGY

ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT

Graduation Project

Using Wireless Technologies For Electricity Meters Reading And Management

Project Team:

Mahmoud Emad Abdeh

Yasmeen Izzat Alashhab

Thalji Khaled T. Qawasma

Rimah Abd Alrahman Natsheh

Project Supervisor:

Dr. Murad Abu Subaih

2012

اهداء

إلى من هدانا لهذا وما كُنا لنهتدي لولا هُداه
إلى من وفقنا لهذا وما توفيقُنا إلا به
إلى المولى العظيم جلّ في علاه.
إلى من اصطفاهُ اللهُ على العالمين
إلى المصطفى -صلّ الله عليه وسلم-.
إلى فلسطين: أرضاً وشعباً وحجراً مُقاوِم
إلى الذين زرعوا بذرةَ الهندسة فكان الثمر
اربعة مُهندسين ابتكروا الطريقَ إلى القلوب
إلى النُجوم المَنيرة لنا سبيلنا والتي ما فتئت
تنحرق لتظلَ منيرةً ومضيئةً كسراجٍ مُنير
إلى أمهاتنا أدامهنّ اللهُ تاجاً على رؤوسنا
إلى والدينا ولا سيّما لروحِ المغفور لهُم
بإذنه تعالى:
(عماد محمود عابده)
(عبد الرحمن شحاده النتشه)
وجعل اللهُ هذا التواضعُ الهندسي صدقةً لأرواحهم
في الملكوت الأعلى.
إلى السند الذي شدّ الوثاقُ به
إلى من كانَ أباً وأخاً إلى د. المشرف(د.مراد ابو صبيح)
الذي ما بخلِ علينا بثانيةٍ من وقته وعلمه حفظهُ الله
إلى جامعة بولتكنيك فلسطين لاسيما عمادة كلية الهندسة
ودائرة الهندسة الكهربائية والحاسوب
إلى الأخوة الصادقين الرائعين المُخلصين
إلى من كانت معهم أجمل لحظات حياتنا
إلى طلاب كلية الهندسة دفعة2007م
لإخوتنا وأحبتنا والقاطنينَ قلوبنا
إلى الذين عرفتهم ولم يعرفوني

أهديكم هذا النقشَ المتواضعَ
لوحةً كلماتيةً ترسمُ ظلنا بقلوبكم

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 their 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 supervisor **Dr. Murad Abu Subaih** for his wise and amazing supervision.

Abstract:

Because of the manual low accurate electricity meters reading method, we purpose to replace it with a system design based on GPRS ,SMS and ZigBee technologies using Arduino microcontroller as a core processor, in order to collect readings wirelessly and store them in a database and return the electricity bills to the customers without a human need. Moreover, a two way communication between the electricity meters and the electricity company allows us to manage meters, receive alarms and send commands to make the desired change.

This system leads to an easy and accurate readings collecting and managing method. Clustering structure of the network to get a reduced data traffic and saved energy will be achieved.

ملخص المشروع

نظرا للأخطاء اليدوية في جمع قراءات عدادات الكهرباء, نسعى الى تصميم نظام يعتمد على الاتصالات اللاسلكية باستخدام تكنولوجيا GPRS و تكنولوجيا ZigBee و الربط بين الاجزاء المختلفة بواسطة المعالج الدقيق الذي يعتبر حلقة الوصل الاساسية في النظام, حيث ترسل القراءات لا سلكيا و تحفظ في قاعدة بيانات في شركة الكهرباء دون تدخل بشري مما يسهل الحصول على قراءات دقيقة للعدادات.

هذا النظام يمكننا ايضا من التحكم بعدادات الكهرباء عن بعد, عن طريق وصل او فصل الكهرباء عن مشترك معين عن طريق ارسال امر لاسلكيا الي العداد وذلك في حالات معينة مثل التأخر في تسيد الفواتير المستحقة , او محاولة التلاعب بالعداد لسرقة الكهرباء او ما شابه.

للتقليل من حركة مرور المعلومات على شبكة ال GSM و تقليل الطاقة اللازمة سنقوم بتقسيم المنطقه الجغرافيه الى مجموعات لكل منها نقطة رئيسية تمثل البوابة لهذه المجموعة,حيث يتم نقل القراءات في داخل المجموعه من نقطه الى اخرى الى ان تصل للنقطة الرئيسية.

Table of contents

Table of Figures

List of Tables

1

CHAPTER ONE

INTRODUCTION

1. 1 Overview

Recent advances in wireless technologies enable human to design and spread many applications which are able to monitor and control environments.

One of the applications that can be built over wireless networks is electricity meters reading and management. This application is intended to replace the existing manual methods of gathering data from meters and transferring them to a central database for billing, troubleshooting, analyzing and tele-controlling, by an accurate automatic technique using General Packet Radio Service (GPRS) and Small Message Service (SMS), provided by Global System for Mobile (GSM).

The proposed design consists of three main parts: the Electricity meters, wireless transmission network and the graphical user interface. Data will be transmitted to the electricity company when requested, and saved automatically in a database.

1. 2 Objectives

1) To utilize different wireless technologies to collect electricity meters readings without the need of human.
2) Saving time and efforts for the electricity companies and their employees.
3) To collect readings from unreachable and far areas.
4) To send electricity bills to each customer wirelessly.
5) To achieve high level of readings' accuracy with no more estimations, which leads to an accurate future plan for energy consumption.
6) To decrease the errors and mistakes in entering data at the server.
7) Managing meters and controlling the security to prevent electricity stealing.

1. 3 Project Motivations

In order to overcome the problems of the traditional meter reading system, efforts are underway around the world to automate meter reading. The work of a human being employed can't give an accurate reading because of the estimation that happens when some meters cannot be easily reached. As known, home privacy is an important issue for the local people and the entry of the meter-readers into their homes for recording electricity consumption during day time, is not always welcomed, especially when men of the family are at work and only the women are there. So the company estimates the readings depending on previous bills, which may cause some problems between customers and the company. The billing system will not be based on estimation or monthly average but on actual readings.

Moreover, since we live in an occupied country, problems of reaching some houses that are near settlements will face the company employees, and it's another reason to estimate readings. With automatic meter reading, the billing system will not be based on estimation or monthly average but on actual readings, and the meters will be controlled wirelessly.

Besides, the input data at the server will be automatically entered and saved into a database that assures data correction, without a need for the human to enter readings one by one and making some potential mistakes. Calculations, customer's information, dates and time will be monitored.

This project is expected to be very useful for electricity companies, it's also useful for us as students, it will add to our information, skills and experiences. Moreover, it opens many fields to be developed in the near future, and could be applied in our country since it's not yet.

1. 4 Project idea and approach

As previously mentioned, the main idea of this project is to automate the electricity meters reading and management. In our project, an external electronic circuit will be connected to the meter in order to deliver the reading. Readings will be saved at a microcontroller considered as the main connector between the system parts, the transmission of the data will be done over the GSM network using GPRS technology, which has many features and advantages that will be explained later. A GSM modem is connected to the microcontroller at the meter side and at the main server's side to transmit and receive.

Practically, using a GSM modem for each meter has two disadvantages: first of all, consider that we have hundreds of thousands of meters; it will be very expensive to connect each meter to a GSM modem. Secondly, the traffic on GSM network will increase when number of transmitters (modems) increase.

To overcome these two problems we will use ZigBee technology which will make it possible to connect multiple meters to only one GSM modem. A number of neighboring meters can access this technology using ZigBee modems (XBee). They send their readings to the meter that is connected to the GSM modem and has the responsibility of sending the group cluster readings.

At the company side, the PC server is also connected to a GSM modem that receives the customer's readings. Then, data is passed through a microcontroller which interfaces the modem with the PC. A programmed database at the server will store the received readings for each customer. It will also store the time and date of each reading and gives the needed calculations and information. After that, the electricity bill will be sent using SMS for each customer.

Additionally, managing meters can be accomplished by detecting any abnormal action, or by disconnecting the electricity for a specific meter.

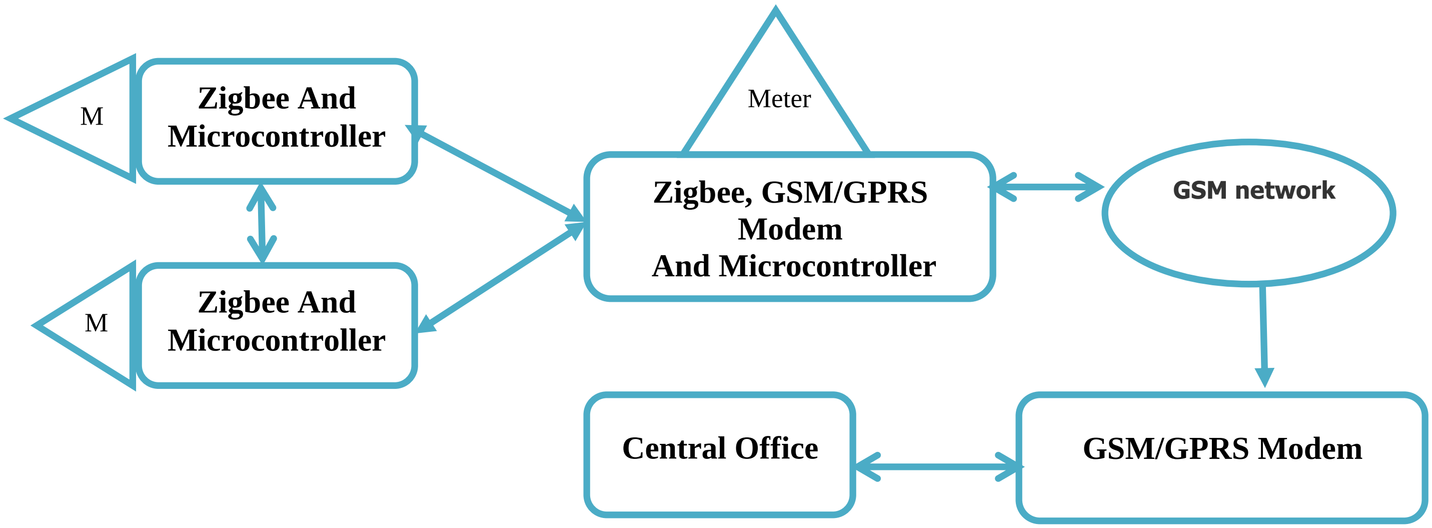


Figure 1.1: General block diagram of the system

1. 5 Previous studies

In [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 displayed on the LCD in two lines. The communication is done using GM862 cellular Quad band 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 Arduino microcontroller because it is easier to be programmed and to be connected to the PC's.

In [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 to 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 as Wi-Fi.

In [3] the authors presented Remote Meter Reading using the GSM/SMS System. This system is using a standard electricity meter that has pulse output. By connecting this meter to a microprocessor based unit, the number of pulses is received to the data logger which will integrate their number over time, and store the summation in a separate file, one for each 24 hours. They integrate the file every 60 minutes because of the fact that the KWh is treated at the electricity exchange on an hourly basis.

A built in GSM modem in the data logger is used to call the datalogger for initialization to send SMS every night randomly over 30 minutes from 00.30 to 1.00 to avoid sending data at the same time.

As will be mentioned later, GPRS technology provides higher data rates than SMS service. Moreover, SMS will cost more than GPRS, so using GPRS is more preferred to be used here.

1. 6 Estimated Cost

The following table contains the hardware components that will be used in our system and its costs. The costs are in shekel (₪).

1. 6.1 Hardware resources

Table 1.1 Development hardware resources cost.

Component Name	**(NIS)**	**Quantity**	**Total Cost**
Arduino (UNO)	192₪	4	768₪
Arduino(Duemilanove)	110₪	1	110₪
XBEE pro module	180₪	5	900₪
XBEE pro shield	60₪	5	300₪
GPRS module	440₪	3	1320₪
LDR	3₪	12	36₪
Arduino parts	100₪	1	100₪
Resistors	0.1₪	50	5₪
Transistors(BC337.25)	3₪	10	30₪

Inverter (74LS04)	3₪	4	12₪
SIM card	15₪	3	45₪
Decoder (74LS138)	3₪	2	6₪
NAND gate (74LS00)	3₪	5	15₪
3S buffer (74LS125)	3₪	4	12₪
Counter(74LS192)	5₪	42	210₪
Diode	1₪	25	25₪
Polar capacitor	2₪	5	10₪
Non-polar capacitor	2₪	5	10₪
Regulator	3₪	4	12₪
Heat sink	5₪	4	20₪
Transformer (6-18)V	50₪	7	350₪
Arduino Jumper Cables	24.00₪	5	120₪
JUMPER WIRES 9" F/F PA OF 10	12.00₪	4	48₪
Heat gun(Soldering iron)	75₪	1	75₪
Side cutters	10₪	1	10₪
Small pliers	10₪	1	10₪
Track cutter	15₪	1	15₪
Small flat-blade screwdriver	10₪	1	10₪
Soldering iron stand	30₪	1	30₪
Aluminum wire (Reel of sol	60₪	1	60₪
Desoldering pump (solder sucker)	20₪	1	20₪
Boards	20₪ - (20 cm)	3	60₪
Switch	3₪	3	9₪
Relay	10₪	4	40₪
LED	0.5₪	10	5₪
Stackable	0.1₪	300	30₪
Hardware cost 4838₪			

1. 6.2 Human resources

Table 1.2 Operating Human resources

Kind of resource	Cost (NIS)	Tota cost
Movement and transportation	500₪	500₪
Stationery and printing supplies	400₪	400₪
Human resources cost 900₪		

1. 6.3 Total Development Costs

Table 1.3 Total development cost

Resource	**Cost ($)**
Hardware resources	4838₪
Human resources	900₪
Total	**5738₪**

1. 7 Time Schedule

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

Table 1.4: Time Planning

Tasks \ Weeks	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										■	■	■	■	■	■	
Documentations	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Project delivery															■	

2

CHAPTER TWO

THEORITICAL BACKGROUND

2.1 Introduction

2.2 Digital electricity meters

2.3 Microcontrollers

2.4 ZigBee technology

2.5 Global System for Mobile communications (GSM)

2.6 Database

2. 1 Introduction

Electricity meters reading and management project can be designed and achieved using different methods. Wireless technologies, meters and even microcontrollers have many options. Thus, each part in the project should be studied deeply in order to decide the most suitable choice.

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

2. 2 Digital electricity meters

Electric meters are devices that measure the amount of electrical power used. They are typically used by power companies to measure how much electricity a household or apartment uses. The electric companies take readings on these electric meters, typically monthly, and charge the electricity user for the amount of power consumed for the month. There are kinds of electricity meters: analog meters and digital meters. Digital meters have numeric displays that show numbers in kilowatt hours.

The challenge in this project is the variety of digital electricity meter. There are many types of meters and different manufacturers. Each one has a special protocol for meter reading, so we explored to find a proper method to read most of them.

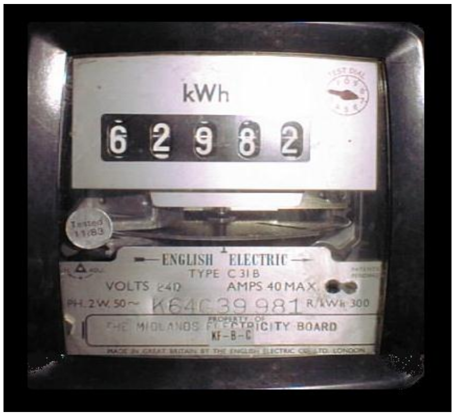


Figure 2.1: Different kinds of electricity meters.

2. 3 Microcontrollers

A microcontroller is "a highly integrated chip that contains all the components comprising a controller, this includes a CPU, RAM, some form of ROM, I/O ports, and timers". [4]

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.

There are many microcontroller types and architectures different in length of register and instruction word. We can mention here the most known types of microcontrollers:

1) PIC (8-bit PIC16, PIC18, 16-bit dsPIC33/PIC24)
2) Intel 8051
3) MIPS
4) ATmega

The microcontroller used in this system, is the Arduino microcontroller. Arduino is an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs.

The main advantages of Arduino microcontrollers [5]**:**

1) Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms.
2) Simple, clear programming environment - The Arduino programming environment is easy-to-use, yet flexible enough for advanced users.
3) Open source and extensible software- The Arduino software is published as open source tools, available for extension by experienced programmers.
4) Open source and extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers.

There are several kinds of Arduino which is largely similar in hardware: Arduino Uno, Arduino Duemilanove, Arduino Diecimila, Arduino Pro, Arduino Mega…..act.

<u>The main features of the Arduino Uno are:</u>

1) ATmega328 Microcontroller.
2) 5V Operating Voltage.
3) 7 - 12 V Input Voltage.
4) 14 Digital I/O Pins.
5) 6 Analog Input Pins.
6) 40 mA DC Current per I/O Pin.
7) Flash Memory 32 KB
8) KB SRAM.
9) 1 KB EPROM.
10) 16 MHz Clock Speed.

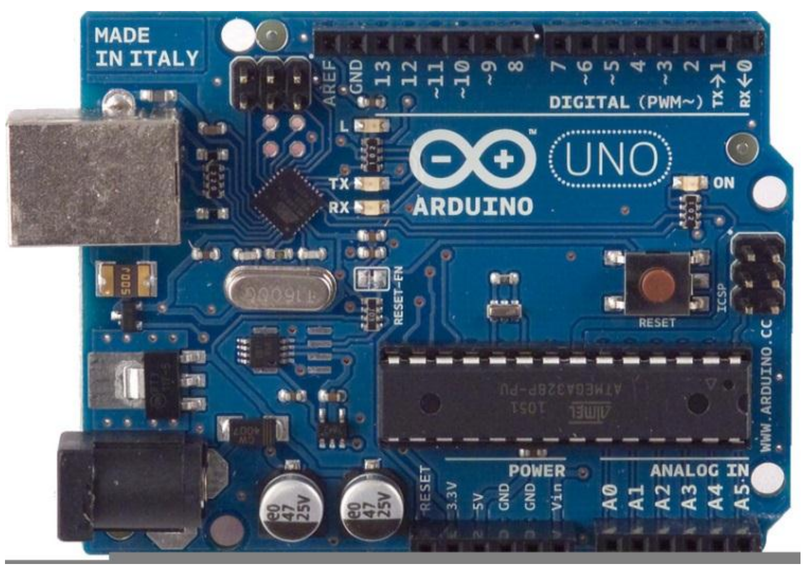


Figure 2.2: Arduino Uno Microcontroller

2. 4 ZigBee Technology

In this system, a clustering structure will be designed by ZigBee routers to reduce the number of GPRS gateway nodes.

ZigBee has been developed to meet the growing demand for capable and self- organizing Ad-hoc wireless networking between different low power devices.

The ZigBee standard operates on the IEEE 802.15.4 physical radio specification which supports multiple network topologies such as point to point, point to multipoint and mesh networks.

It uses FM and spread spectrum technologies to work in the 2.4 GHz, 868 MHz and 915 MHz, and these three bands can transit with 250 Kbps, 20 Kbps and 40 Kbps. When we use the 2.4 GHz band, ZigBee technology can transmit 10 meters indoor; while in the outdoor transmission distance can reach 200 meters. In other uses spectrum, the indoor distance is 30 meters, while in the outdoor transmission distance can reach 1000 meters. We can notice that these distances are suitable for the expected distances between houses that will be parts this of system.

The main advantages for the ZigBee technology are [6]**:**

1) Low power: it can work with a battery for several months or years such as the device is activated when the data is transmitted or received only.
2) Low cost: according to the low complexity comparing with the Bluetooth technology.
3) Short time delay.
4) Reliable and secure over sniping Bluetooth devices.
5) Transmits for short distances which are suitable for our project.

There are three different types of ZigBee devices:

1) ZigBee coordinator: it forms the root of the network tree and might bridge to other networks. There is exactly one ZigBee coordinator in each network since it is the device that started the network originally. It is able to store information about the network, including acting as the trust center and repository for security keys.

2) ZigBee router: it can act as an intermediate router, passing data from device to another.

3) ZigBee end device: it contains a functionality to talk with either the coordinator or a router. This relationship allows the node to sleep for a significant amount of time, thereby giving long battery life.

Figure 2.3: XBee Series 2 modem.

2. 5 Global System for Mobile communications (GSM)

2. 5.1 Foundations and Definitions

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

It is one of several types of cellular technology, and is the second generation (2G) of the mobile communication networks. GSM allows for voice and data transmissions across several frequencies, and supports cellular devices from basic to smart phones. One of the most popular cellular services, GSM service is available across almost 80 percent of the world.

GSM when used in the standard operation mode was able to provide data transfer speed up to 9.6 Kbps. Through the years a new technique was introduced in the GSM standard called High Speed Circuit Switched Data (HSCSD). 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:[7]

1) The switching system (SS); which is responsible for performing calls processing and subscriber-related functions.
2) The base station system (BSS); all radio-related functions are performed in the BSS.
3) The operation and support system (OSS); it is the functional entity from which the network operator monitors and controls the system.

GSM Network Areas:

The GSM network is made up of 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 service 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 service area is an area served by one network operator.

2. 5.2 Why have we used GSM Network?

There are many advantages for GSM that lead us to use it in our system:

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 devices and services.

3) **Extensive Coverage:**
GSM has a widespread use throughout the world. So even though different countries may operate on different frequency bands, users can transfer seamlessly between networks and keep the same number. As a result, GSM users 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 a greater variety of phones 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 phones only made in their respective country.

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

2. 5.3 Short Messaging Service (SMS)

In this system, SMS service will be used at the server's PC, to send the electricity bill for each customer.

SMS is a way of sending short messages to mobile telephones. “Short” means a maximum of 160 bytes. According to the 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 services is that not only does it provide an alternative means of transmission in a cellular communication system but it is a more versatile and convenient option since all new phones are SMS and capable. Additionally, under certain circumstances, SMS can be the only means 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 messages at times when s/he cannot get access to the network for voice calls. This is because SMS uses control channels of a cellular system, which are used for the initial call setup rather than the regular voice traffic channels.

2. 5.4 General Packet Radio Service (GPRS)

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

GPRS is a packet oriented mobile data service that applies a packet radio principle to transfer user data packets between GSM mobile stations and external packet data networks in an efficient way. In 2G systems, 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 continuous 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 extremely efficient ,which means higher data rates -in 2G systems, GPRS provides data rates of 56-114 kbit/second- . For example, If SMS over GPRS is used, an SMS transmission speed of about 30 SMS messages 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 messages 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-packets-are-needed basis rather than dedicated to only one user at a time. [8] [9]

2. 5.5 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.

2. 5.6 GSM Modem (SM5100B)

The most suitable GSM 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 2.4, is called a "Cellular Shield with SM5100B".

The Cellular Shield includes all the parts needed to interface Arduino microcontroller with an SM5100B cellular module. This allows adding SMS and GSM functionalities to Arduino-based project. [10]

Figure 2.4: Cellular Shield with SM5100B.

2. 6 Database

After data being transmitted over GSM network, it reaches the server at the electricity company. Managing and organizing them will be needed. This could be done in a special database.

A database is a collection of related data organized for one or more purposes in such a way that a computer program can easily and quickly chooses desired pieces of data to be accessed, managed and updated. Databases can be classified according to types of content: bibliographic, full-text, numeric, and images. To access information from a database, you need a database management system (DBMS), which is a software package with computer programs that control the creation, maintenance, and the use of a database. [11]

Database design is done before building it to meet needs of end-users within a given application/information-system that the database is intended to support. The database design defines the needed data and data structures that such a database comprises. It can be programmed using different programming languages such as C#, visual basic, F# ...etc. [12]

.

3

CHAPTER THREE

SYSTEM DESIGN

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

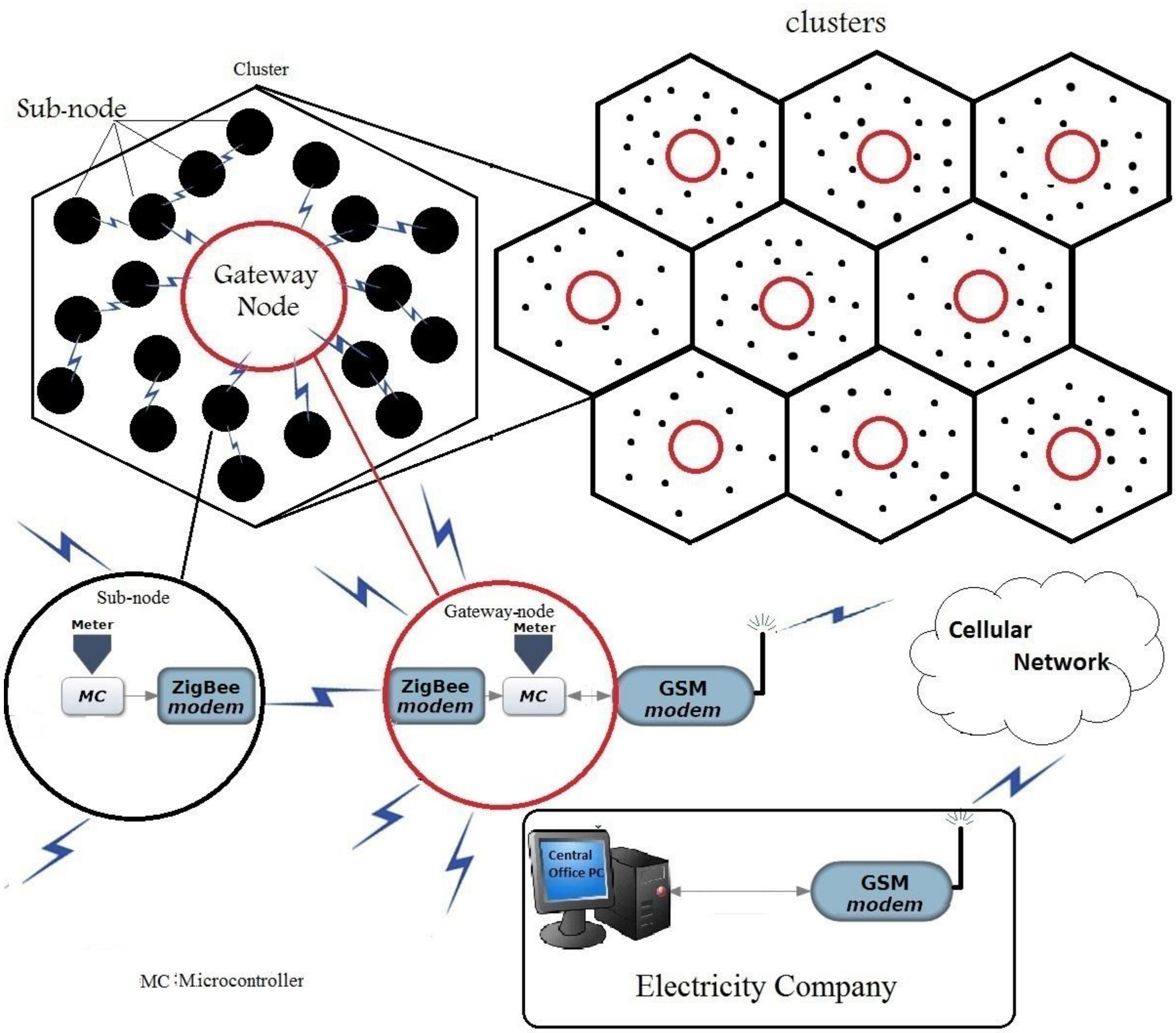


Figure 3.1: General block diagram of the system design.

3.3 System Scenario

As shown in Figure 3.1, the system is based on dividing a certain wide region into multiple clusters, each cluster has a different size depending on the number of electricity meters existing in such a region. It contains a number of neighboring meters that can communicate with an elected meter that performs as a gateway node.

The system has three communication parts:

The first part is inside the cluster, it is based on ZigBee network which connects all meters to the gateway node using ZigBee modems located at each meter including the gateway, and connected to a microcontroller (MC). Then data will be routed daily from a meter to another till it reaches the gateway node.

The second communication part is based on GPRS technology using GSM modems which wait for a request from the electricity company either to transmit the collected data stored at the gateway node to the GSM modem at the central office's PC, or to manage a specific meter. It also used to receive an alarm from meters that have been opened for security purposes. According to a specific programming, a database at the company's PC will report, store and analyze the received data.

The third communication part is based on SMS technology using GSM modems which send the electricity bills for each customer after receiving the reading and calculating the electricity bill.

3.4 System Operation

Figure 3.2 presents a flow chart that explains the system operation at different levels. As shown, it starts with measuring power consumption using electricity meter interfacing circuits, then microcontrollers will process the received signals according to the programs installed in each one, and send them to ZigBee modems that will make the appropriate routing to the gateway node to update the saved data daily.

If a request from a graphical user interface at the PC of the electricity company arrives to the gateway nodes (or a specific one) or to a specific meter, data will be sent using GSM modems to the central office, where it will be stored in a database at the PC to be analyzed.

According to the results at the database, the graphical user interface enables us to send SMS containing the electricity bill and the deadline date for payment. It also enables us to manage meters by connect/cut off the electricity of the meter. In a case of an abnormal action on the meter box, the electricity will be cut off and an alarm will be sent to the electricity company.

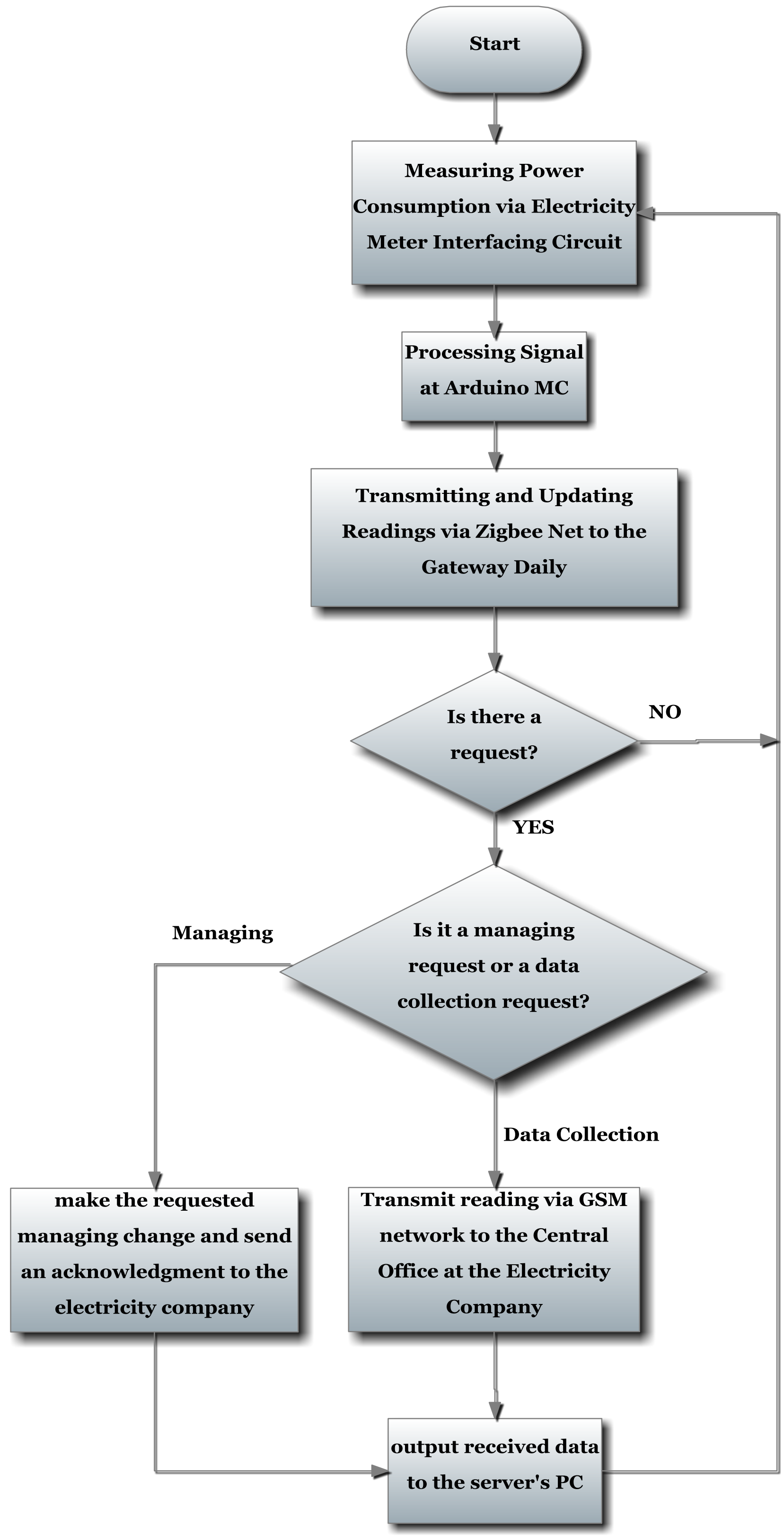


Figure 3.2: A flowchart of the system scenario.

3.5 Digital Electricity Meters

Digital electricity meter is the main element of the system since it's the desired equipment to be read and managed. It measures the power consumption in Kilo Watts per Hour and displays the value on its LCD screen.

Due to the variability of electricity meters, a digital meter has been used in the system called ACE5000 it's shown in Figure 3.3. It's a three phase meter, but in this system it will be connected to measure the one phase power lines. It has been chosen for many reasons since it's widely used, also because of its similarity compared to other meters.

Figure 3.3: ACE5000 Digital Electricity Meter.

As shown in Figure 3.4, an external circuit is needed to perform the task of delivering reading from the meter to a microcontroller.

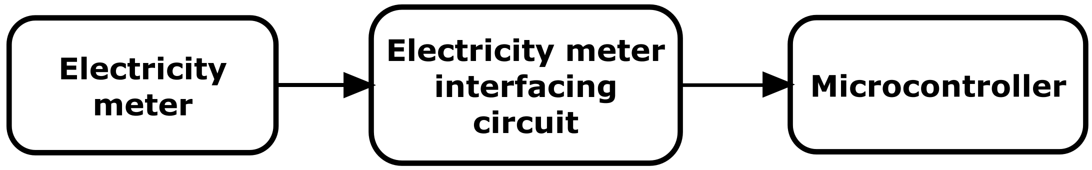


Figure 3.4: A block diagram of the electricity meter interfacing circuit connection.

3.6 Electricity meter interfacing circuit

As mentioned before, a challenge point is the variety of digital meters types, so we worked to find a proper method to read most of them. From the datasheet of the used meter in this project (ACE 5000), it is mentioned that it contains a LED (Light Emitting Diode) that produces pulses, a specific number of pulses represents the consumed power in kilowatt. In the case of ACE 5000 meter, 1000 pulses represent one Kilo Watts of consumed power**.**

3.7 Arduino Microcontroller

The whole system is built around the microcontroller. Thus we can consider it as the central unit in the project. As shown in Figure 3.5, the microcontroller receives the power consumption signals from the electricity meter interfacing circuit, and processes them, and receives collecting data and management requests. It takes the right decision of sending the signals to the transceivers or not. This decision depends on some parameters which are programmed in the software.

Figure 3.5: A block diagram of the microcontroller terminals at a sub-node.

We have used microcontrollers with every electricity meter including the GSM gateways; also it's used with the GSM modem at the central office.

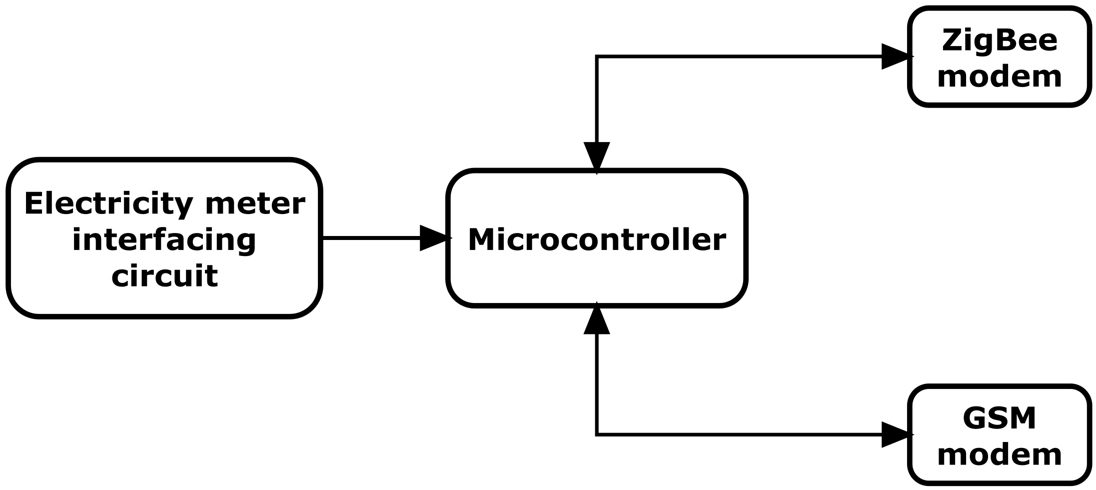


Figure 3.6: A block diagram of the microcontroller terminals at a gateway node.

Arduino Uno is used in this project. It is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz 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 an AC-to-DC adapter or battery to get started.

Figure 3.7: Arduino UNO module.

Description of the Arduino UNO Pins:

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 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). We can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

1) **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.
2) **3.3V:** A 3.3 volt supply generated by the on-board regulator.
3) Maximum current draw is 50 mA.
4) **GND:** Ground pins.

Input and Output:

Each of the 14 digital pins on the Uno can be used as an input or output. 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 kΩ. In addition, some pins have specialized functions [13]:

1) **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.

2) **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.

3) **PWM: 3, 5, 6, 9, 10, and 11:** Provide 8-bit PWM output.

4) **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK):** These pins support SPI communication.

5) **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.

6) 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. Additionally, some pins have specialized functionality:

7) **TWI: A4 or SDA pin and A5 or SCL pin:** Support TWI communication.

There are a couple of other pins on the board:

1) **AREF:** Reference voltage for the analog inputs.

2) **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.

3.8 ZigBee Modem

The ZigBee modem plays the role of routing the electricity meters' readings and alarms to the gateway node daily. It also routes the requests of data collection and meters management from the Gateway node to the desired meter. Each meter in the cluster is connected to a ZigBee modem using a microcontroller. The readings and alarms are routed from ZigBee modem to the next until they reach the gateway node. and in an opposite operation for requests –from the gateway to the meter- As shown in Figure 3.8, the input data of ZigBee modem will be taken from the microcontroller and the output will be routed to the next ZigBee node or to the gateway node as a final destination. Figure 3.8 shows the connection of ZigBee modems at each sub-node and gateway nodes.

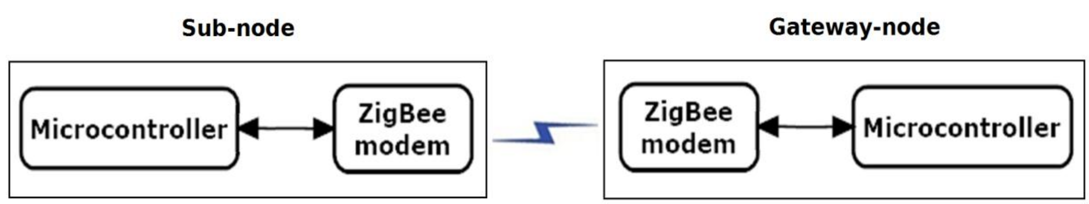


Figure 3.8: A block diagram of ZigBee modems connection.

The most appropriate ZigBee modem we found is “XBee Chip Antenna”. Since it’s compatible with Arduino microcontroller, this is a very popular 2.4GHz XBee module.

This module takes the 802.15.4 stack (the basis for ZigBee) and wraps it into a simple form to use serial command set. It allows a very reliable and simple communication between microcontrollers, computers, systems and anything with a serial port. Point to point and multi-point networks are supported.

<u>It has the following characteristics</u>[14]**:**

1. 3.3V @ 40mA

2. 250kbps Max data rate
3. 2mW output (+3dBm)
4. 400ft (120m) range
5. Built-in antenna
6. Fully FCC certified
7. 6 10-bit ADC input pins
8. 8 digital IO pins
9. 128-bit encryption
10. Local or over-air configuration
11. AT or API command set

Figure 3.9: The XBEE module.

3.9 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 a GSM modem at the monitoring PC in the central office. To send data containing the status of the power consumption, we have to program the GSM modem with a particular language at the Arduino program. Figure 3.10 shows the connection of GSM modems at gateway nodes and central office.

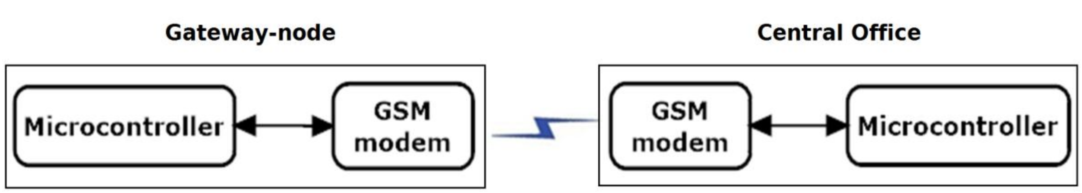


Figure 3.10: A block diagram of GSM modems connection.

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 called 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 allows adding GPRS and GSM functionalities to the project.

As shown in Figure 3.11, the main components of the Cellular Shield are a 60-pin SM5100B connector, a SIM card socket, and an SPX29302 voltage regulator configured to regulate the Arduino's raw voltage to 3.8V. The board's red LED indicates power. The Arduino's reset button is also brought out on the shield. Two jumpers on the board allow selecting which serial pins interface with the cellular module - software (D2, D3) or hardware (D0, D1).

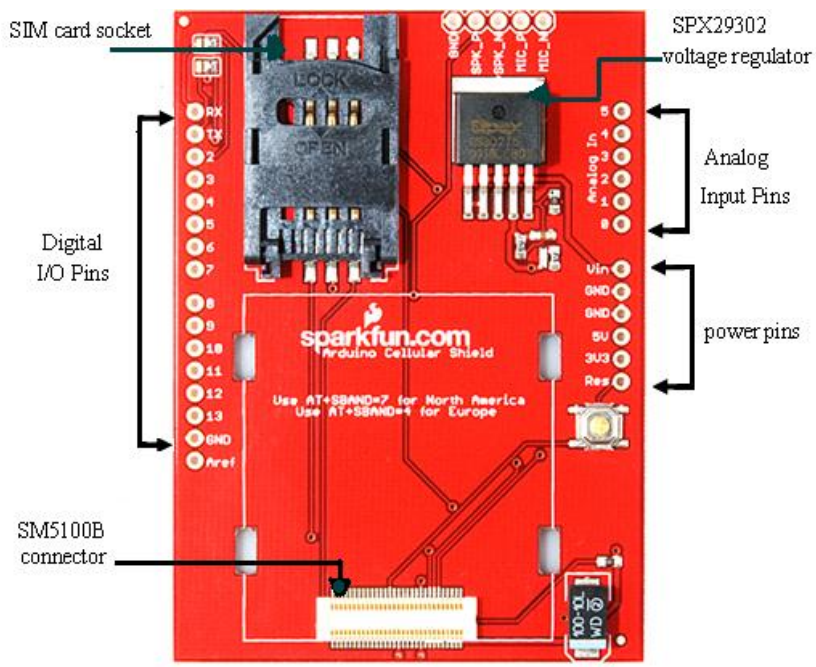


Figure 3.11: Components of the cellular shield.

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

To perform this connection, detailed schematic diagrams are needed for each component in the cellular board.

The following Figures show the required schematic diagrams for the components:

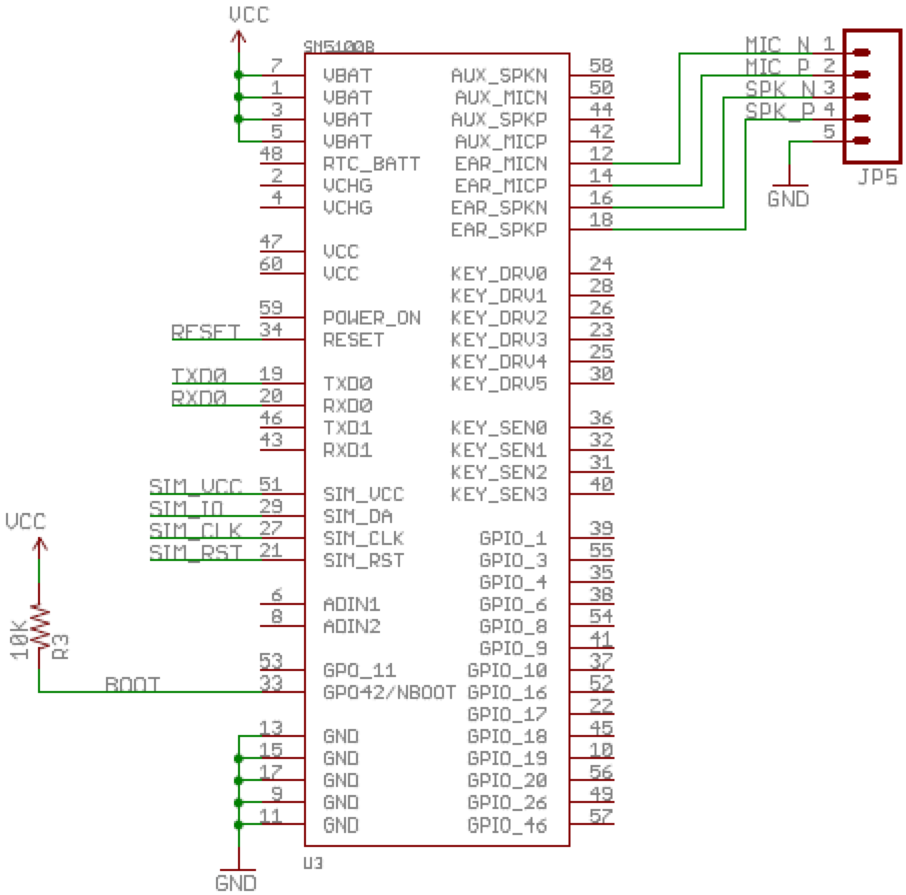


Figure 3.12: SM5100B GSM Modem Pins

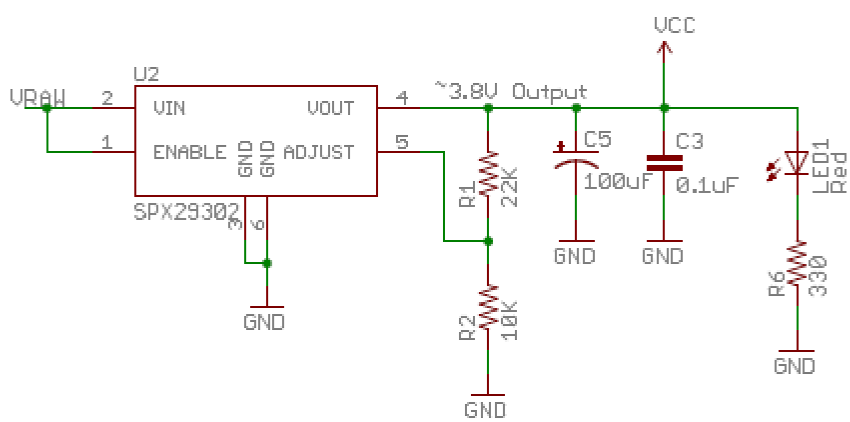


Figure 3.13: SPX29302 Voltage Regulator

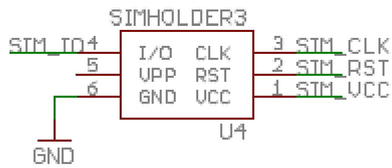


Figure 3.14: SIM Card

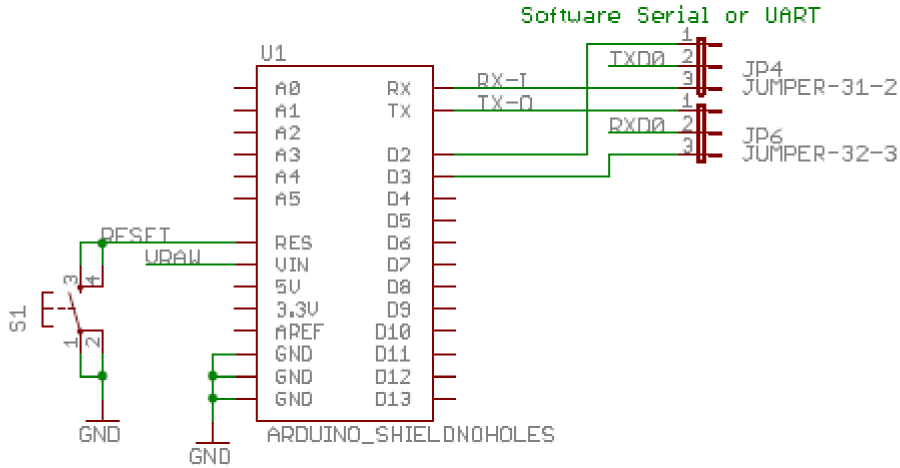


Figure 3.15: Arduino Cellular Shield Pins

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

3.10 Database

The following block diagram shows the connection of the central office PC with the GSM modem.

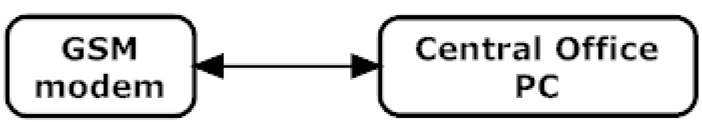


Figure 3.16: A block diagram of GSM-Central office PC connection.

Once the readings arrived at the server side, it appears a need to a program on the server's PC to store ,analyze and manage the data . Building this database and a graphical user interface can be implemented by a program called Microsoft Visual Studio.

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010[4]). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

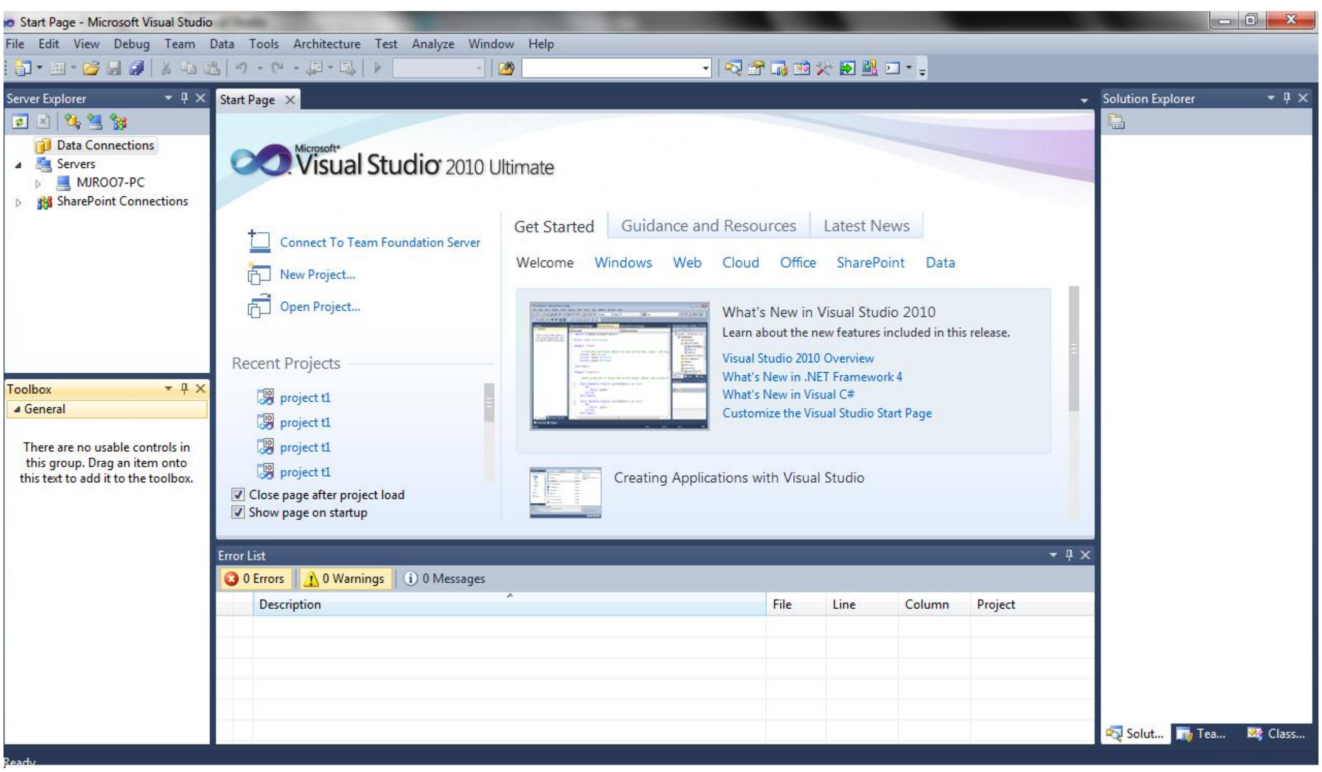


Figure 3.17: The interface of Visual Studio Program

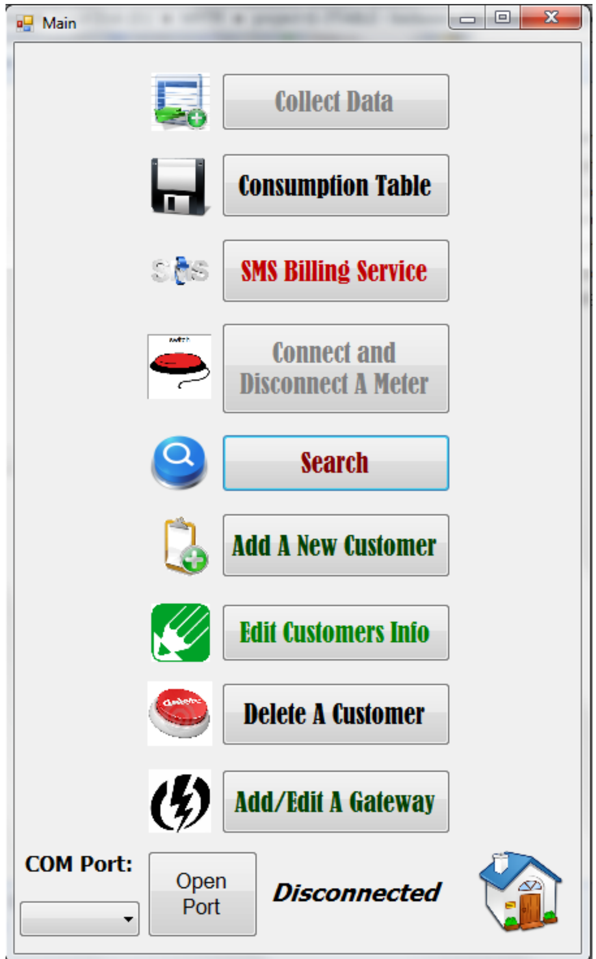


Figure 3.18: The contents of the database at the central office PC.

3.11 Arduino Programming

Arduino software is the IDE processor, but can send the Arduino program (or “sketch”) to the micro controller in the same board. The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board. As the Arduino system is open source, anyone can purchase a blank micro controller and put the boot loader on it, or even write their own boot loader.

The Arduino IDE comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++, although users only need define two functions to make a runnable program:

- “Setup ( )”: a function run once at the start of a program that can initialize settings.

- “Loop ( )”: a function called repeatedly until the board powers off.

Currently, there are 22 IDE versions of software Arduino program, in this project version 21 has been used. Figure shows the interface of Arduino program.

Figure 3.19: Arduino Program Interface.

3.12 Software design

After introducing the hardware design of the system in the previous sections, the software design is presented using flowcharts and Pseudocode for different algorithms. The hardware part has to be programmed is the microcontroller, and the system requires the following software module.

3.12.1 Electricity meter interfacing circuit module

Pseudocode module:

```
Start
{
        Set a system counter to zero;
        Set a KiloWattHour counter of zero;
        While (the output of the electricity meter interfacing circuit equals 1)
        {
        Increment the system counter;
        If the system counter equals 1000;
            {
            Increment the KiloWattHour counter;
            Set the system counter to zero;
            }
        }
        endwhile
}
End
```

This pseudocode can be presented in the following flowchart, as shown in Figure 3.20

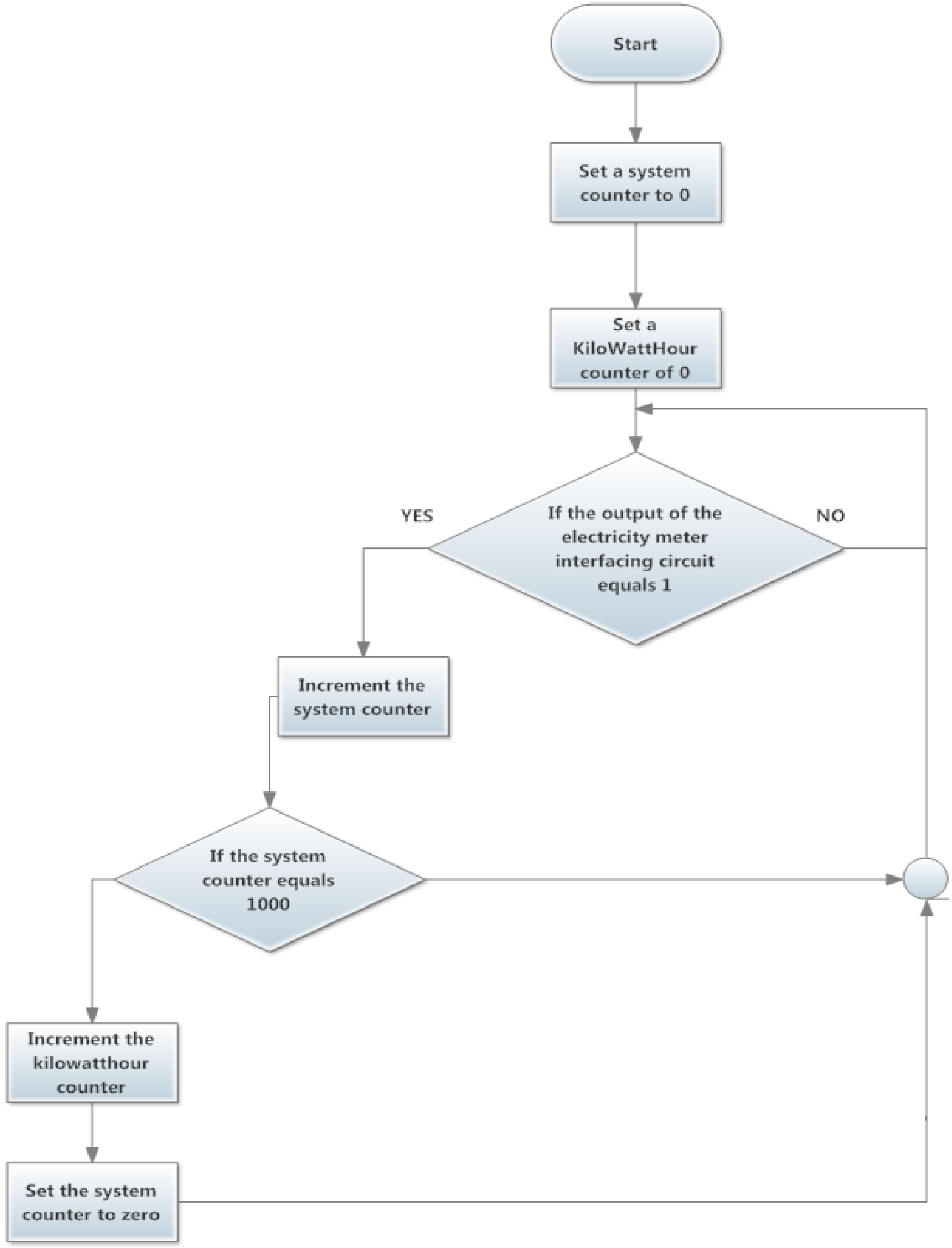


Figure 3.20: Electricity meter interfacing circuit flowchart.

3.12.2 ZigBee modems module

Pseudocode module:

```
Start
{
Set time (Td)
    While current time equals Td
    {
            Activate ZigBee modems at each node;
            If (gateway node)
                    {
                        save data at microcontroller;
                    }
            Else
                    {
                        Send data to ZigBee modems;
                        Route data to gateway node;
                    }
            Deactivate ZigBee modems
    }
    endwhile
}
End
```

This pseudocode can be presented in the following flowchart, as shown in Figure 3.21.

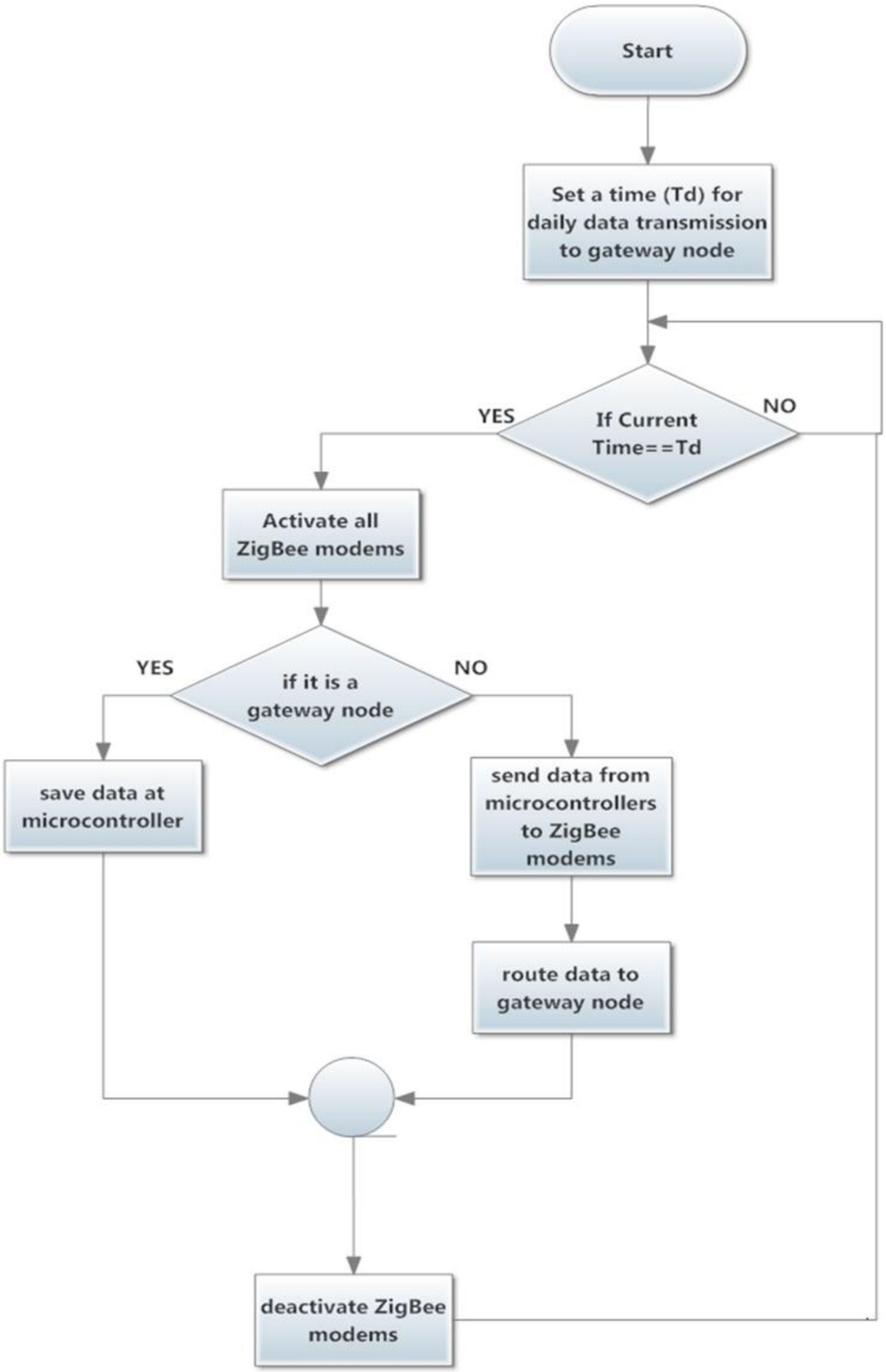


Figure 3.21: ZigBee modems flowchart.

3.12.3 GSM modems of the gateway nodes module

Data Collection Pseudocode module:

```
Start
{
While (request to send data)
        {
          Send AT commands to GSM modem;
          Send data to GSM modem;
          Transmit Data over GSM network;
        }
}
endwhile
End
```

This pseudocode can be presented in the following flowchart, as shown in Figure 3.22.

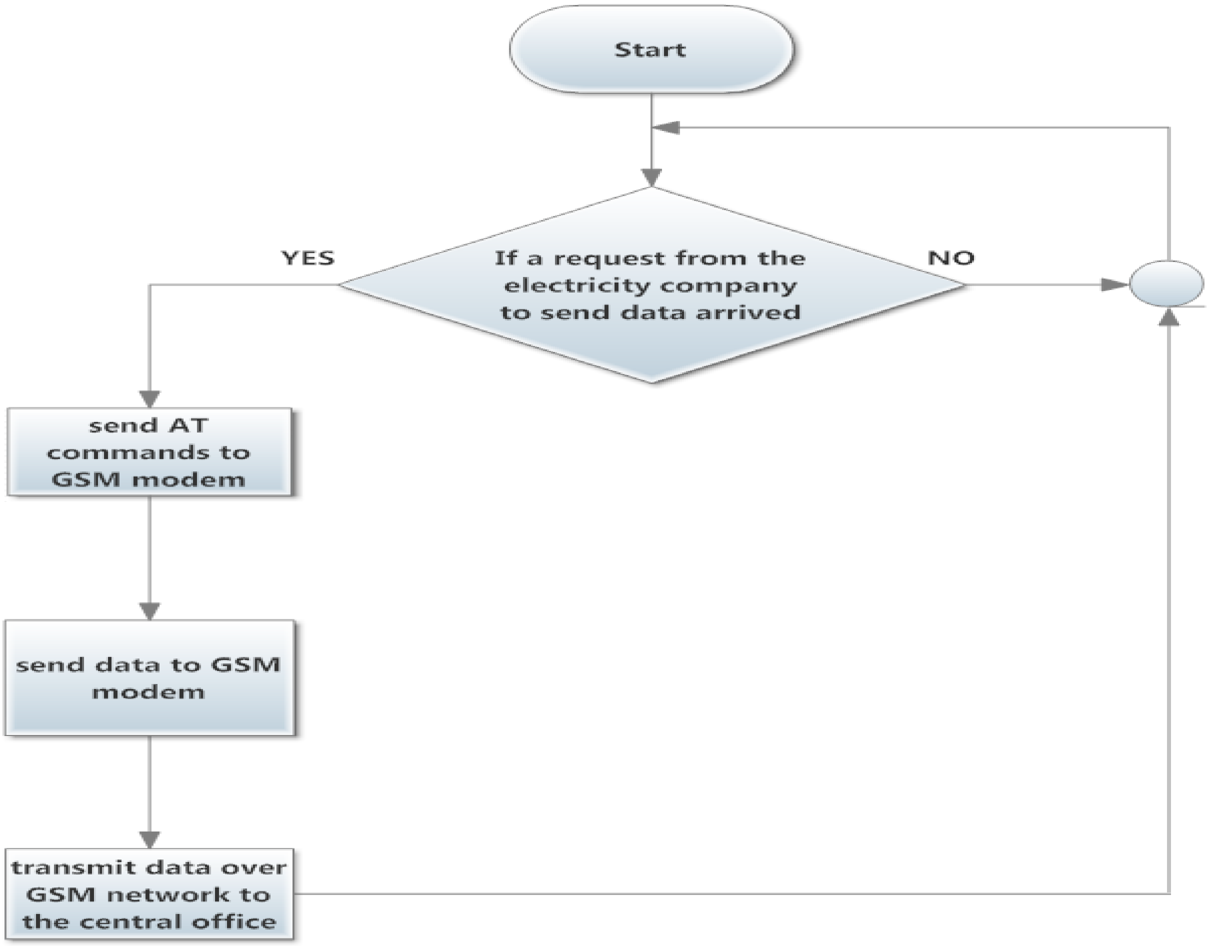


Figure 3.22: GSM modems of data collection at the gateway nodes

3.12.4 Management Requests programing :

Managing Pseudocode module:

```
Start
{
While(managing request)
{
        If(Gateway node managing request )

         Make the connect/disconnect;

         Else;
         Route the request to the wanted Xbee modem's meter;

}
endwhile

}
end
```

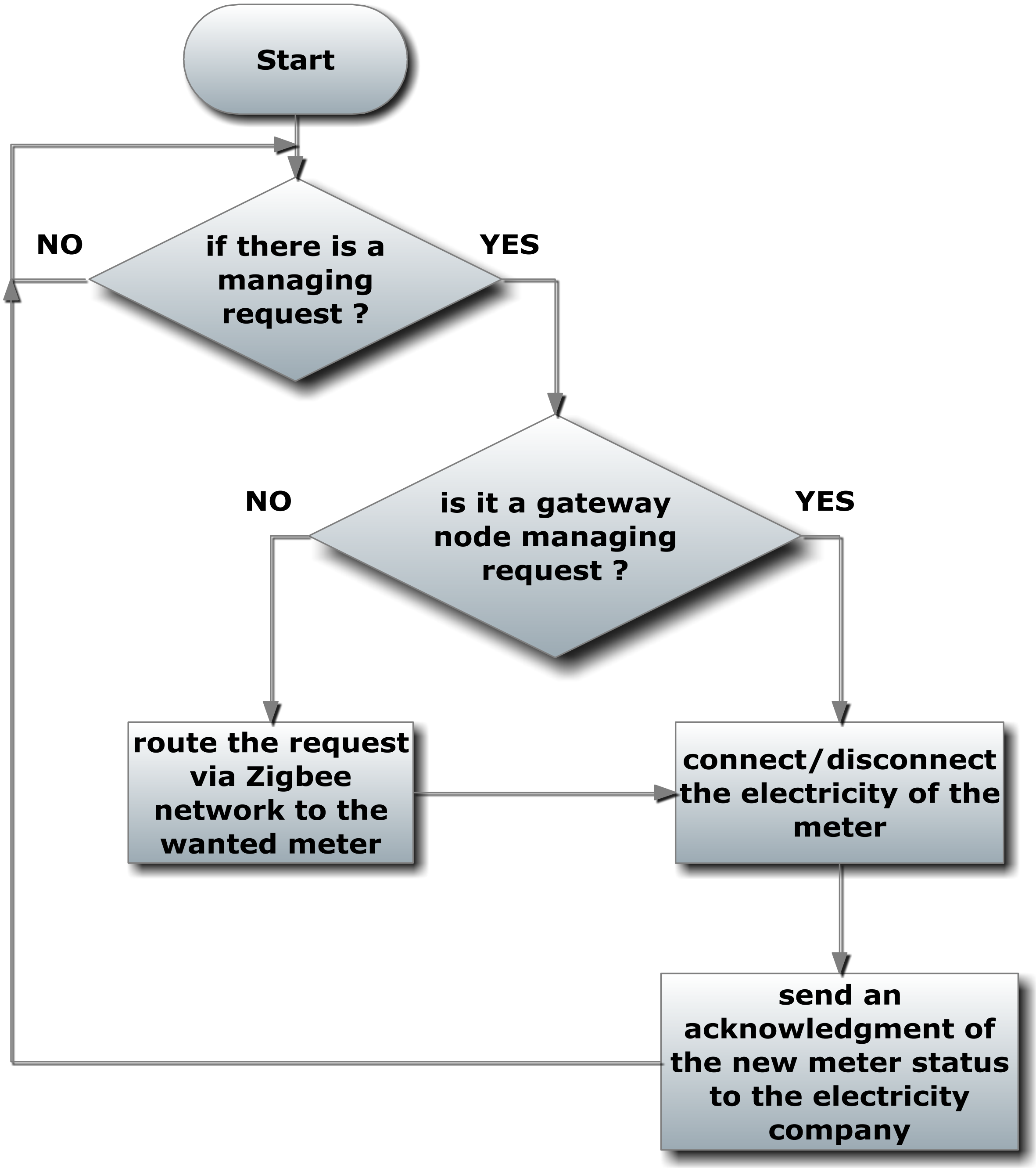


Figure 3.23 : A flowchart of management requests.

3.12.5 Central office module

Pseudocode module:

```
Start
   {
      Set i as a gateway nodes counter;
      Identify gateway nodes by their ID’s with ID[i];
      Start form i=1;
      While (i is less than or equals the maximum number of gateway nodes)
      {
      Send a request from the central office over GSM network to the gateway node with ID [i];
      Wait until the arrival of data to the database;
      Update database (ID [i]);
      Increment i;
      }
      endwhile
}
End
```

This pseudocode can be presented in the following flowchart, as shown in Figure 3.23

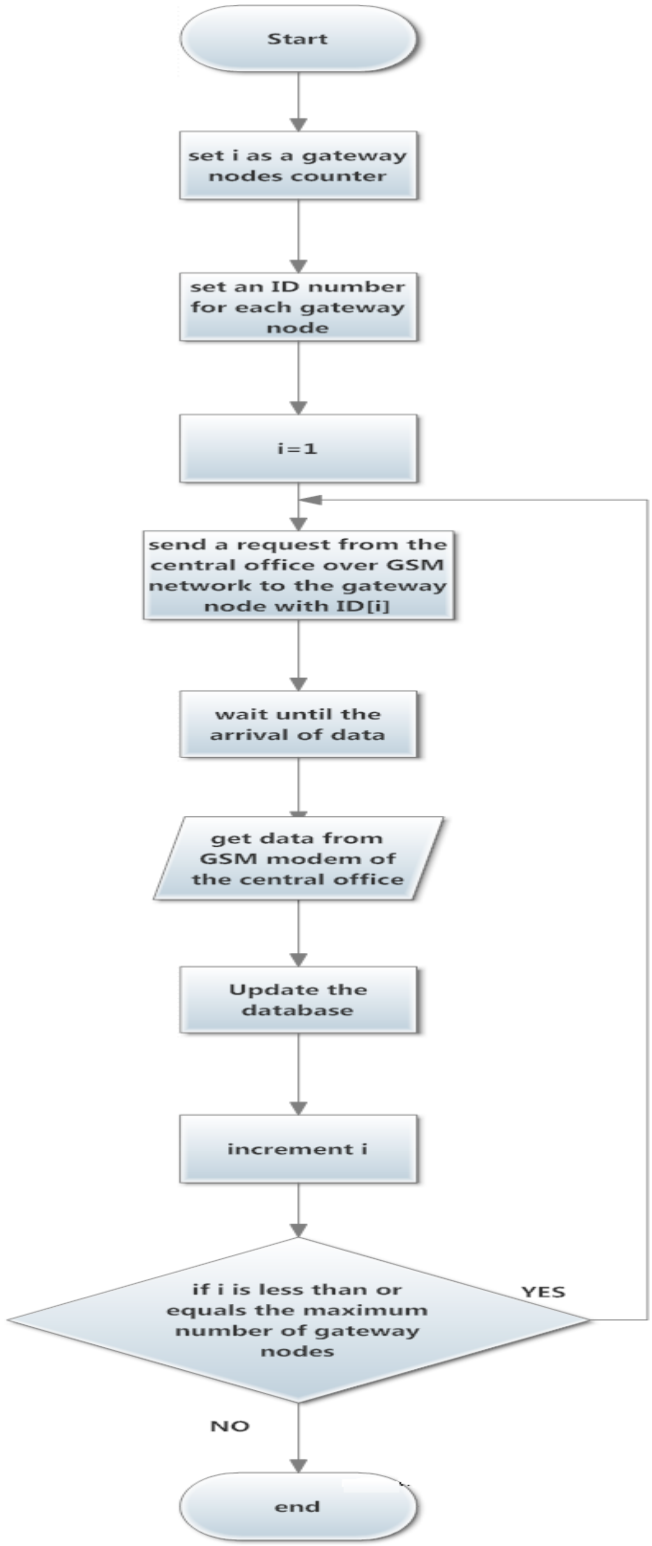


Figure 3.24: Central office flowchart.

3.12.6 Security programming :

Security Pseudocode:

```
Start
{
If(opened box)
{
Disconnect the electricity of the meter;
   If(Gateway node)
       {
     Send an alarm via the GSM network to the electricity company;
   Else;
      {
       Send an alarm via the Zigbee network to the Gateway node;
      Send the alarm via the GSM network to the electricity company;
      }
      }
}
}
end
```

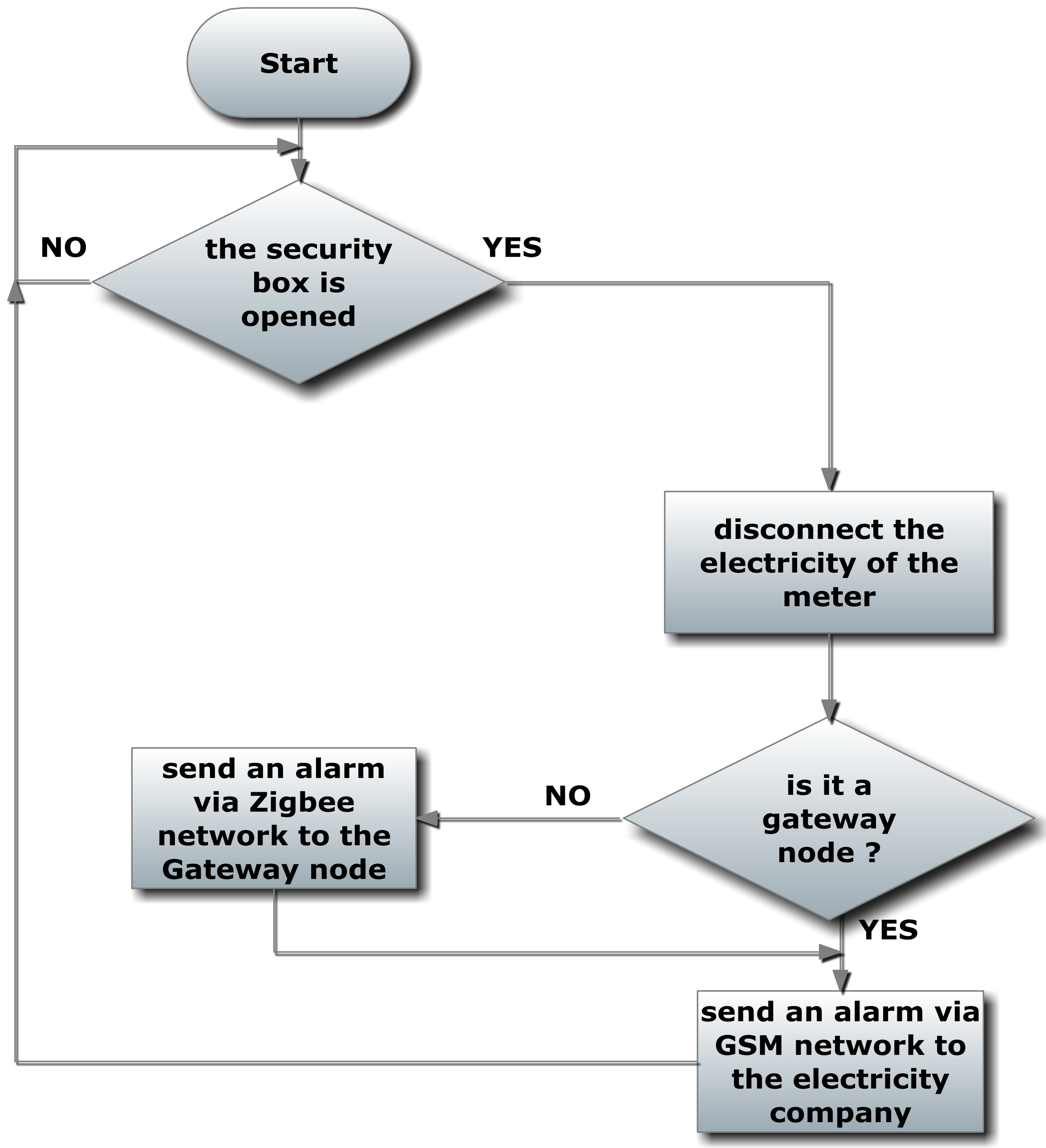


Figure 3.25: Security flowchart.

4

CHAPTER FOUR

HARDWARE DESIGN IMPLEMENTATION

\

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 connections and interfacing several components in the system.

This chapter presents the hardware details by pointing to each pin and showing how it is combined with other pins in each device. It also presents the detailed flowcharts of each part of the system.

4.2 Electricity meter interfacing circuit

4.2.1 LDR circuit

An external circuit based on LDR (**Light Dependent Resistor**) has been designed and connected to the meter in order to recognize the light pulses of the meter. LDR is a device which has a resistance that varies according to the amount of light falling on its surface, LDR has a high resistor value in the darkness, and this value decreases smoothly when the light increases. This feature helps to build the interfacing circuit.

The following Figures (4.1) and (4.2) illustrate LDR operation.

Figure 4.1: LDR with absence of light.

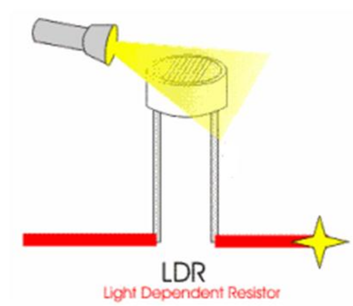


Figure 4.2: LDR with light.

There is a basic circuit using **light dependent resistor** activated by light, the circuit just requires an **LDR**, some resistors and a small signal transistor.

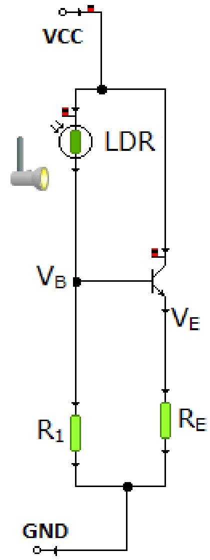


Figure 4.3: LDR circuit

To use the LDR in the interfacing circuit, a driver circuit using BC337-25 transistor has been used. This transistor will get the light changes on the LDR and output the results. When the light level is low, the resistance of the LDR will be high. This prevents the current from flowing to the base of the transistor which will lead to zero emitter current due to relation:

($I_C=\beta*I_B$) …………….. Equation (4.1)

Where α is the transistor current gain. Consequently, the output of the transistor will be **almost** zero.

However, when light shines onto the LDR, its resistance falls and current flows into the base of the transistor which allows the emitter current to pass. And the output of the transistor will be existed.

Experimentally, the suitable value of the resistor R_1 that will achieve the purpose is 62 KΩ, this value has the suitable sensitivity of the LDR changes according to the amount of light emitted from the meters' LED. If this value is decreased, and the circuit sensitivity to the light will be lower.

A 1 KΩ resistor is connected to the emitter of the transistor in order to reduce the current flowing through the transistor, and also to get a fixed voltage value at the output.

The output of the LDR circuit is a voltage analog value. So in order to digitize this value before sending it to the microcontroller as an input one 74LS04 chip which includes 6 inverters inside it has been connected to the output of the LDR circuit to give a value of (1) when a pulse emitted from the meter and a value of (0) when there are no pulses by using two inverters from the chip.

This circuit takes the power from the same power supply of the microcontroller (+5 volt).

4.2.2 Electricity meter interfacing circuit calculations

Arduino Microcontroller needs low current signals, If (Is) is defined as the maximum saturated current that passes through the interfacing circuit, and from the microcontroller datasheet, a 5mA is chosen as a suitable amount of current to go through it. To find (R_E):

- V_{CC}=5 V, I_S=5 mA, R_C=0 Ω , what is the value of R_E?
- $I_S = \frac{V_{CC}}{(R_C + R_E)}$ …………….. Equation(4.2)
- By substituting the previous values in this equation we get:
- $R_E = \frac{V_{CC}}{I_S} = 1K\,\Omega$ …………….. Equation(4.3)

From experimental measurements that we made, we found that LDR resistivity in full darkness equals 1848 KΩ and in high light intensity equals 6.48 KΩ. As known for any digital equipment characteristics, (0-2.5) V analog signals known as logical zero and (2.5-5) V knows as logic one. Then two values at the transistor circuit output (V_E) are needed, the first one is between (0-2.5) V, and the other is between (2.5-5) V. If a value of 3.8 V is chosen for V_E which drives the first 74LS04 inverter, so the question is: what is the value of R_1 that achieves a value of V_E =3.8 V?

- $V_B = V_E + 0.7$ …………….. Equation(4.4)
- $V_B = 3.82 + 0.7 = 4.52V$

And using the LDR experimental value in voltage divider biasing rule:

- $V_B = \left(\frac{R_1}{(R_1 + LDR)} \right) * V_{CC}$ …………….. Equation(4.5)
- $4.52 = \left(\frac{R_1}{(R_1 + 6.48K)} \right) * 5\text{ V}$

From equation 4.4

- $R_1 = 62\text{ K}\Omega$

When checking the value of V_E in the absence of light:
Sub in equation (4.5)

- $V_B = \left(\frac{62}{(62 + 315\text{ K})} \right) * 5$ // 315KΩ because there is no absolute full darkness.

- V_B= 0.82 V …………….. Equation(4.5)
- $V_E = V_B - 0.7$

- $V_E = 0.12$ V   // this value can't drive the 74LS04 inverter, and then the output will be Zero.

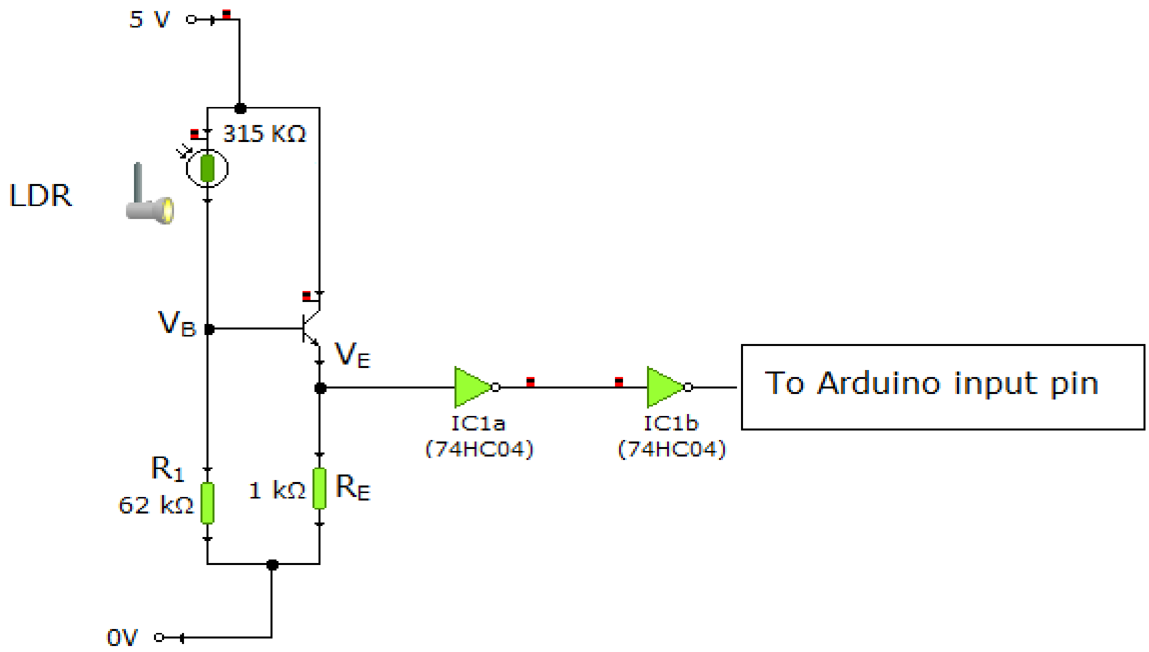


Figure 4.4: Electricity meter interfacing circuit with the absence of light.

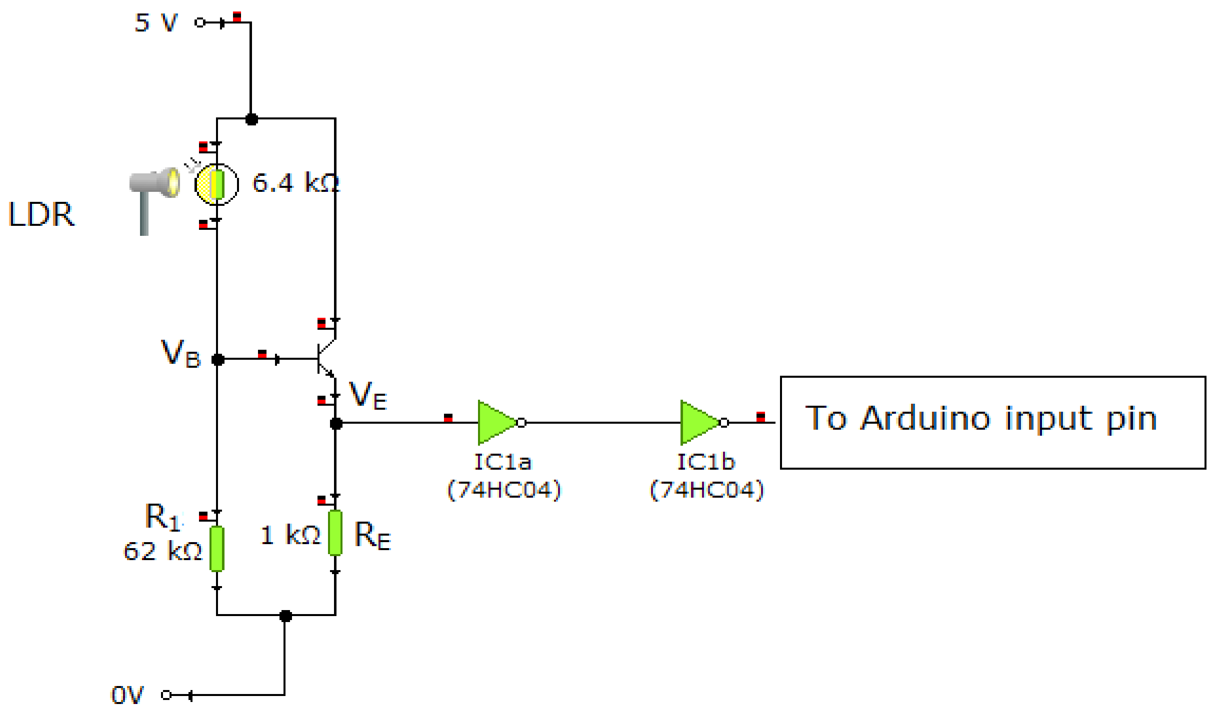


Figure 4.5: Electricity meter interfacing circuit with the existence of light.

4.3 Interfacing Arduino Microcontroller

In this project, three Arduino microcontrollers based on ATmega 328p chip will be been used. One is jointed with ZigBee shield at each node, one is connected to cellular shield with SM5100B at the gateway node and one is connected with the cellular shield and the PC at the central office.

Arduino microcontroller is a central unit in this project, it receives the power consumption signals from the electricity meter interfacing circuits, processes them and sending the results to the transceivers.

The electricity meter interfacing circuit has been connected to Arduino via digital pins on the Arduino board which has 14 digital input/output pins and 6 analog input/output pins.

As shown in Figure 4.6, a power supply has been used to power the Arduino, so we have connected the power pins {VCC, GND, AVCC, AGND, and AREF} to the power supply as shown in Figure (4.6) as follows:

1) VCC is connected to +5 volt.
2) GND is connected to 0 volt.
3) AVCC (analog VCC) is connected to +5.
4) AGND (analog GND) is connected to 0 volt.
5) AREF (analog reference) is connected to +5 volt to have an analog rang from (0 to 5) volt.

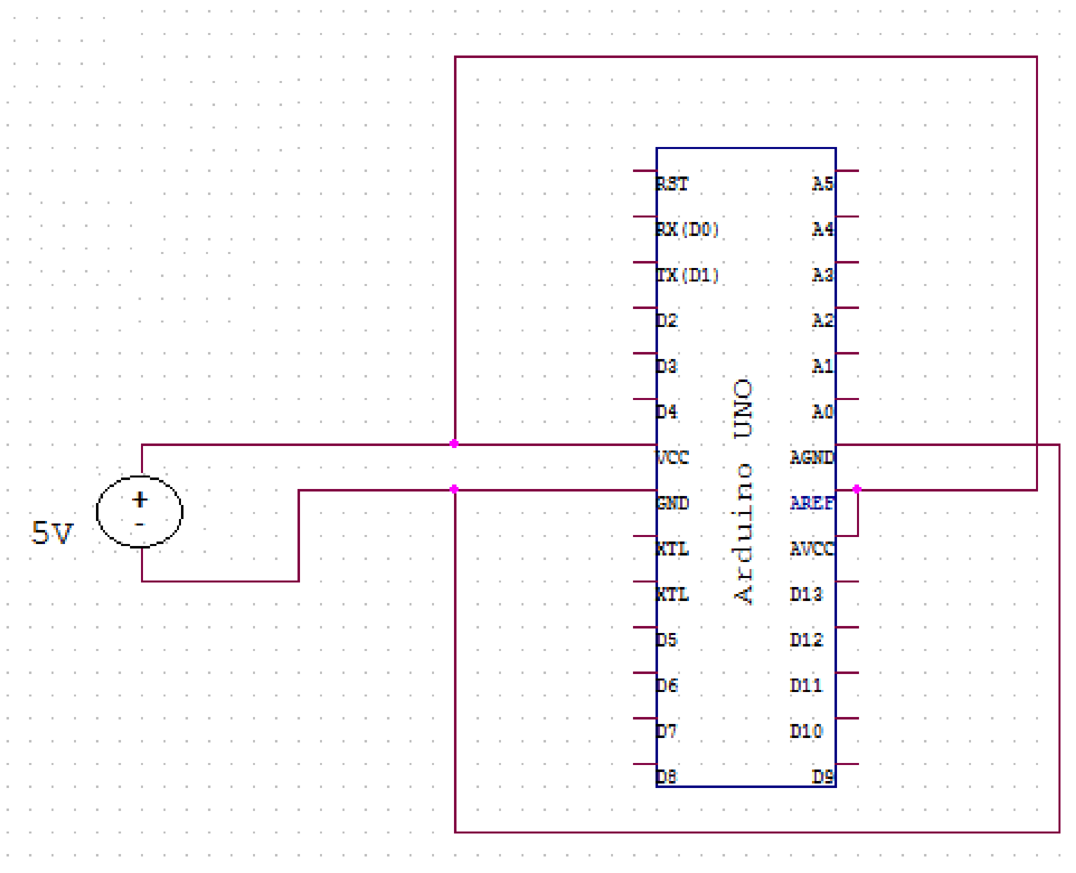


Figure 4.6: Interfacing Arduino Microcontroller with a Power supply

4.4 Electricity meter interfacing circuit connection

Electricity meter interfacing circuit has three pins, input voltage pin, ground pin and the output voltage pin.

Connecting the circuit to the Arduino is done through the digital and analog pins from the output voltage pin (V_{out}) to the digital pin 9 on the Arduino board.

This circuit operates at 5V. So to power it, the input voltage pin is connected to the voltage pin (AVCC) in Arduino which in turn provides circuit with the right amount of power which equals 5 volt and the third pin is the ground pin which is connected to pin (AGND). See Figure (4.7).

The microcontroller is programmed to take the output from this circuit and input it at digital pin, it will count the pulses received and saves the results as 1 Kilo Watts for 1000 pulses.

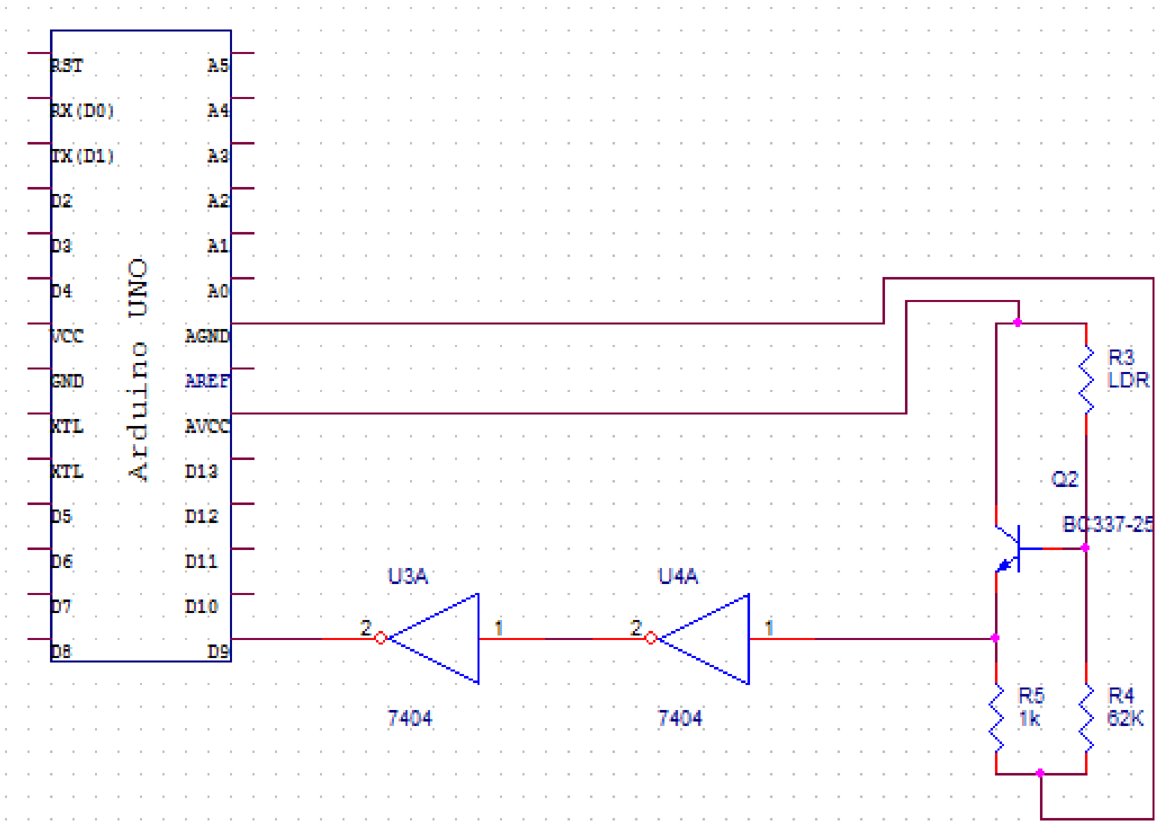


Figure 4.7: Interfacing electricity meter interfacing circuit.

A relay will be connected to the feeder of the meter for management purposes. It operates as a switch that connect or disconnect the electricity of the meter.

The following figure shows how the relay is connected with the meter, where we deal with the lamp in the figure instead of the meter.

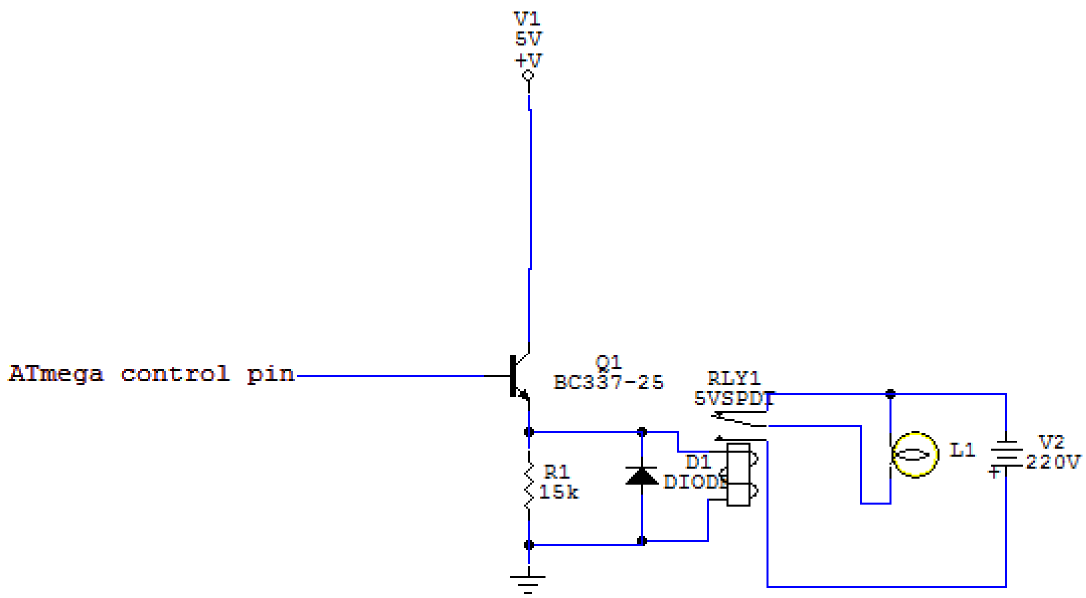


Figure 4.8: The relay-meter connection

4.5 Shields Interfacing

In order to connect the Shields to Arduino Uno microcontrollers, shield can be plugged on the top of Arduino, and then connecting pins to each others can be done.

In this project, two shields have been used. One of them is XBee shield and the other is cellular shield.

For XBee shield, connecting (VIN) and (GND) pins of Arduino with those of the shield in order to take power from the Arduino. In this case, XBee shield has a specific power operating value equals to 3.3 V. Instead of using a voltage regulator, a voltage divider can be used to achieve the purpose.

In order to use this method, two resistors are used, one of them is R2= 100 KΩ and the other can be found using the voltage dividing rule:

- $V_{OUT} = \frac{R_2}{R_2 + R_1} * V_{IN}$ …………….. Equation(4.7)
- $3.3V = \frac{100K}{100K\Omega + R1} * 5V$
- R1= 50 KΩ

The following Figure shows the connection of resistors and power supply with XBee shield Figure (4.8).

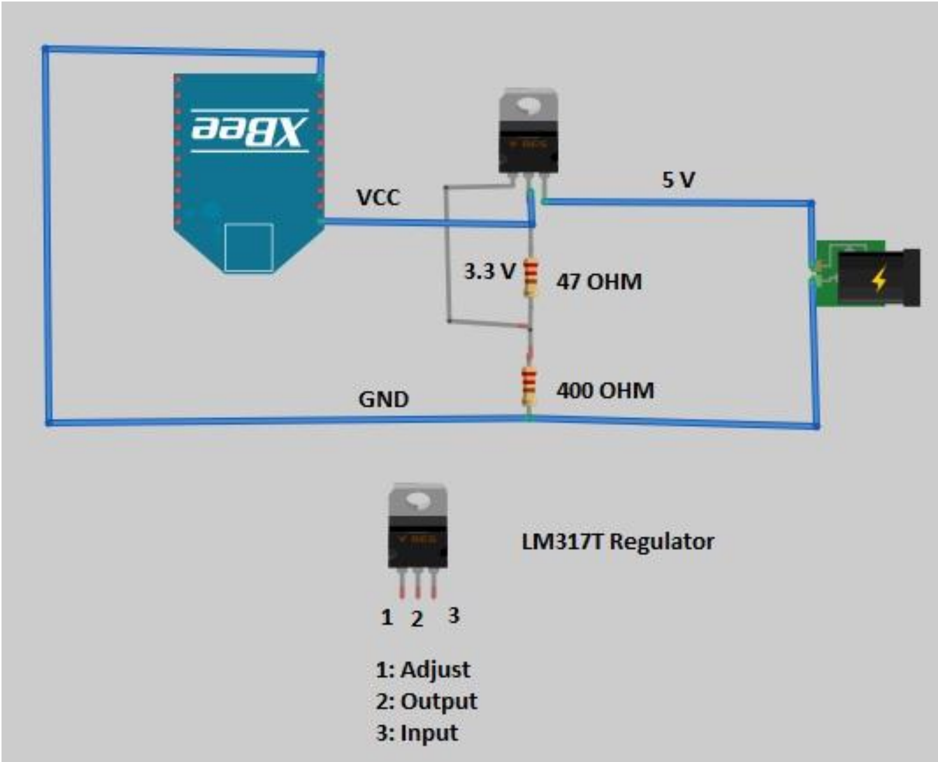


Figure4.9: XBee shield connection with a voltage divider.

An interface can be made easily between XBee shield and any hosting device through a logic-level asynchronous serial port. Through its serial port, the shield can communicate with any logic and voltage compatible Universal Asynchronous Receiver/Transmitter UART, or through a level translator to any serial device .Two options for the command mode are supported:

- AT Command Mode
- API Command Mode.

In this system, AT commands will be used.

As shown in the Figure (4.9), TX is connected to input data pin (D_{IN}) and RX is connected to output data pin (D_{OUT}).

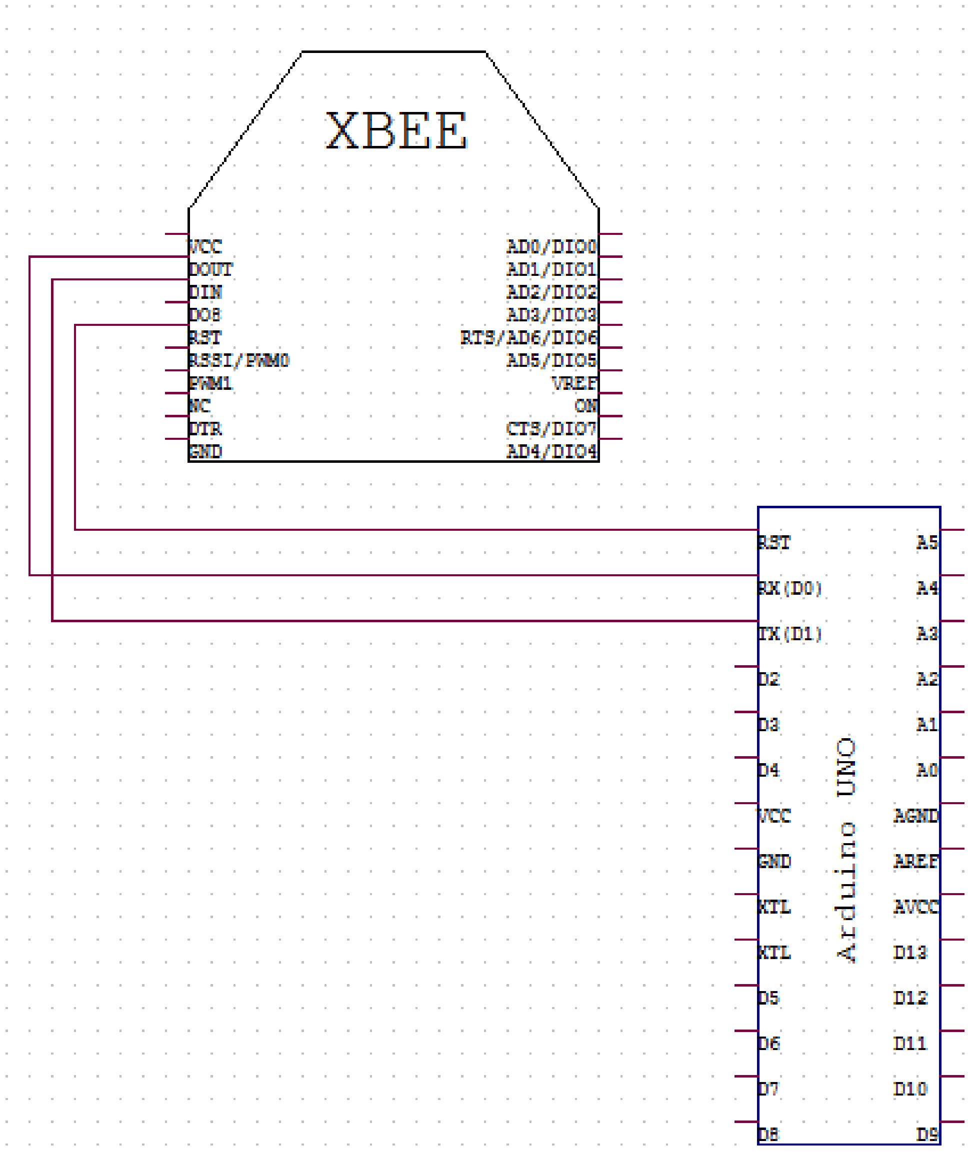


Figure 4.10: Interfacing XBee shield

For Cellular shield, which is compatible with Arduino microcontroller, it takes its power from the microcontroller power supply. This shield has an internal regulator that makes the needed voltage equals to 3.8V.

As shown in Figure (4.10), the serial communication can be achieved by two ways: using software pins (D2, D3), or hardware pins (D0, D1).

In this system, we want to connect this shield by the software serial pins (D2, D3) by introducing a new library in the main Arduino IDE that allows to use the digital pins of the microcontroller as additional serial ports, and to use the same programming commands of the main serial ports for the new ones in the program. Cellular shield can be programmed by transferring suitable AT commands from Arduino to it.

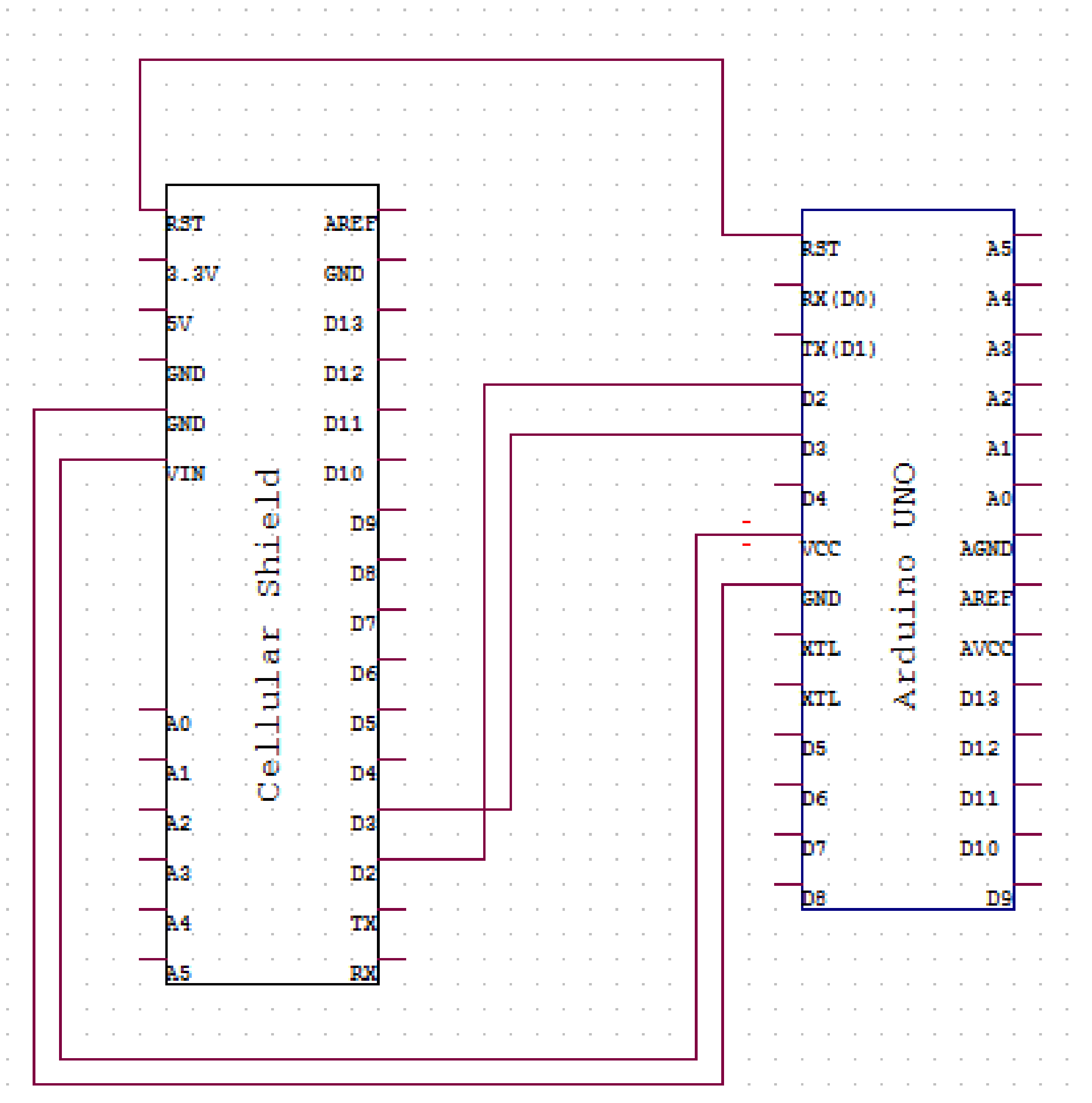


Figure 4.11: Cellular shield interfacing.

The combination of the microcontroller, XBee shield and the cellular shield is shown in Figure (4.11).

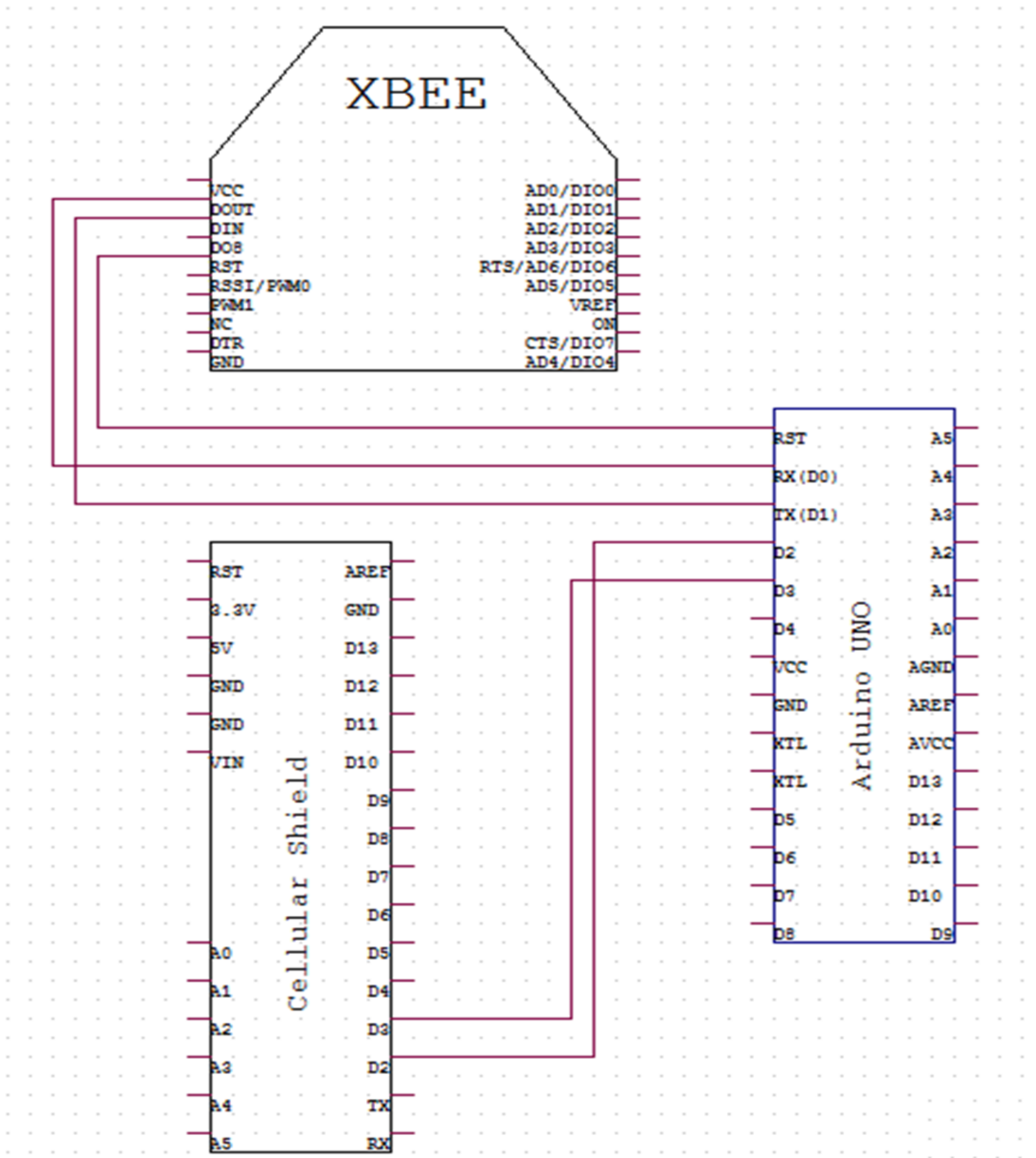


Figure 4.12: XBee shield and cellular shield interfacing.

4.6 Hardware Connection of GSM modem with the central office PC

At the electricity company's side, the GSM modem is connected directly to the PC as USB connection to a certain com port.

5

CHAPTER FIVE

SYSTEM IMPLEMENTATION

5.1 Introduction

5.2 Arduino Programming

5.3 Electricity Meter Interfacing Circuit Programming

5.4 Xbee series 2 modems Programming

5.5 Cellular Shield with SM5100B Programming

5.6 Management code Programming

5.7 Security code Programming

5.8 Graphical user interface and database Programming

5.1 Introduction

Software is an important part in any technological system. Operating and controlling any component in this project requires software handling.

This chapter explains the required software with detailed steps and procedures of programming every used component in this project.

5.2 Arduino Programming

It should be noted that most of the designed system has been programmed using the Arduino software which has controlled the microcontrollers and the transceivers.

Arduino Programming Steps:

- First Step: Selecting Arduino Board.

Selecting the entry in the Tools >> Board >> menu that corresponds to the chosen Arduino. We choose the Arduino Uno (5V, 16 MHz) / ATmega 328, as shown in Figure 5.1.

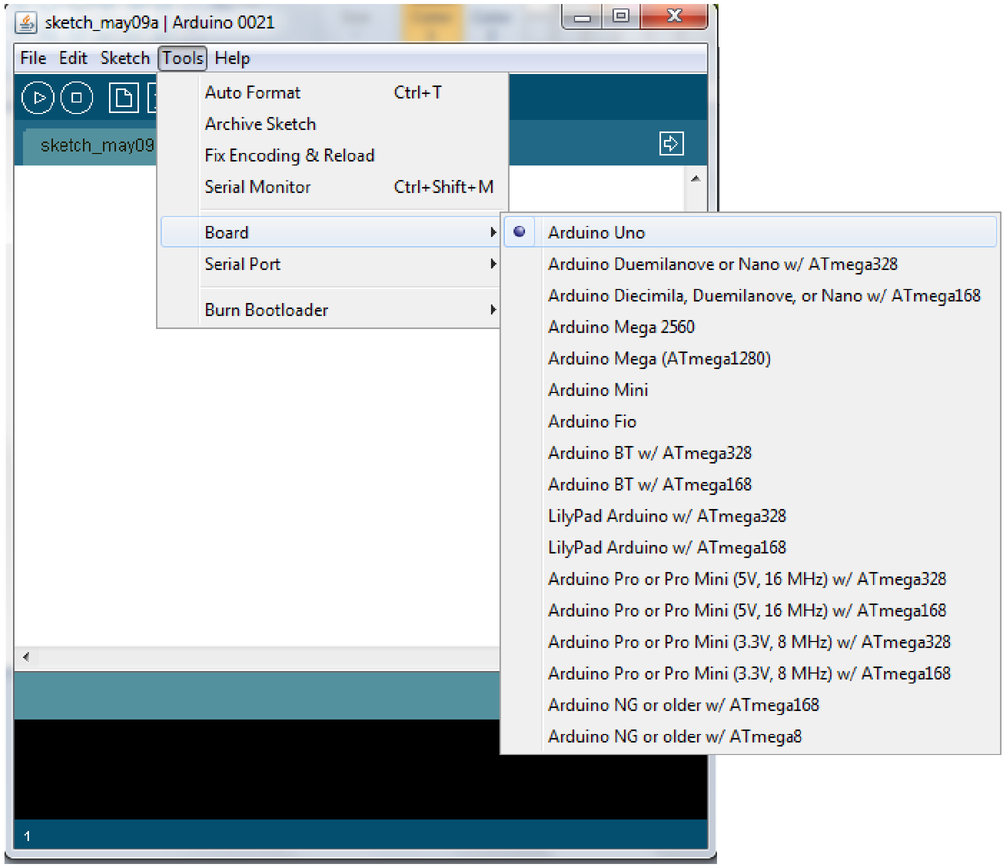


Figure 5.1: Select Arduino Uno Board

- Second Step: Selecting the Serial Port

The board communicates with the computer via a USB interface, but with a serial protocol. This entire means is that we must select the correct serial port number.

Selecting the serial device of the Arduino board is from Tools >> Serial Port menu. As shown in Figure 5.2, COM8serial port is selected in this project.

To find out, Arduino board can be disconnected and the menu should be re-opened; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

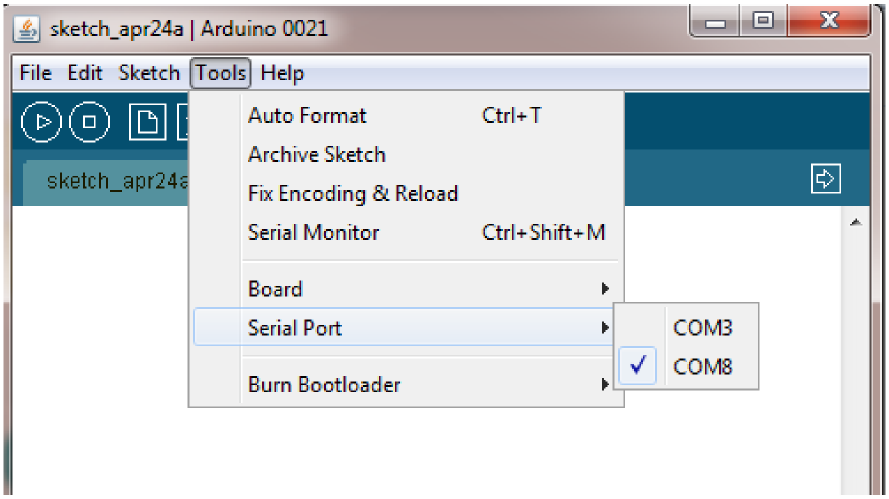


Figure 5.2: Selecting Serial Port

- Third Step: Saving the Program

After writing the code, saving is done by selecting File >> Save, show Figure 5.3.

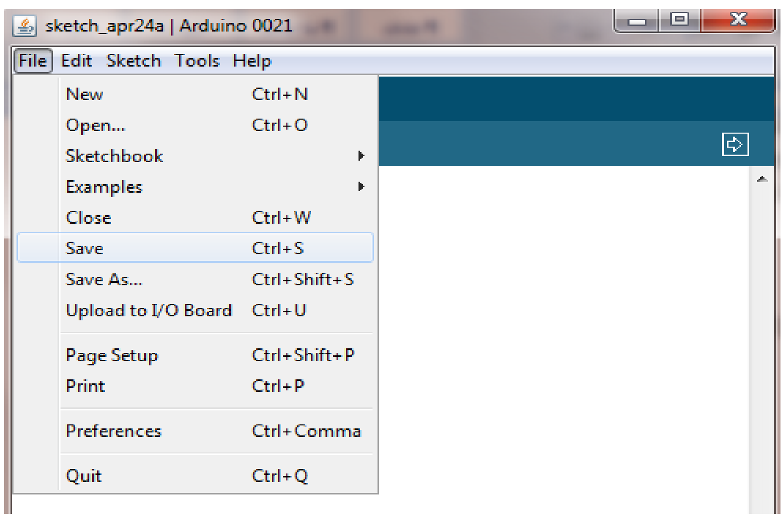


Figure 5.3: Saving the Program

- Forth Step: Compiling the Program

Compiling can be selected from Sketch >> Verify/Compile, as shown in Figure 5.4.

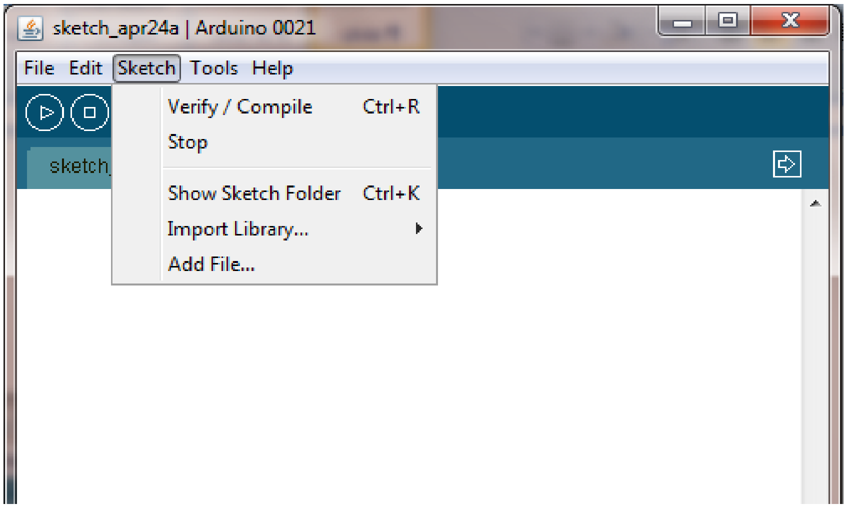


Figure 5.4: Compiling the Program

- Fifth Step: Uploading the Program

Uploading is done by selecting File >> Upload to I/O Board, as shown in Figure 5.5. After waiting a few seconds; we can see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

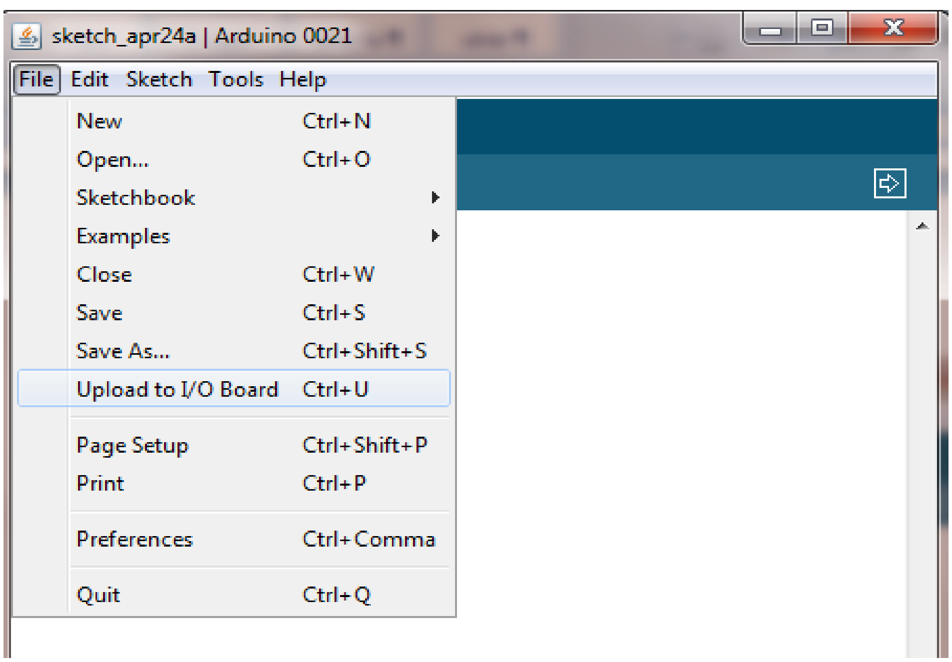


Figure 5.5: Uploading the Program

- Sixth Step: Displaying Result on Serial Monitor

Serial Monitor displays serial data being sent from the Arduino board (USB or serial board). It is also used to send data to the board.

Result on the Serial Monitor is displayed by selecting Tools >> Serial Monitor, as shown in Figure 5.6.

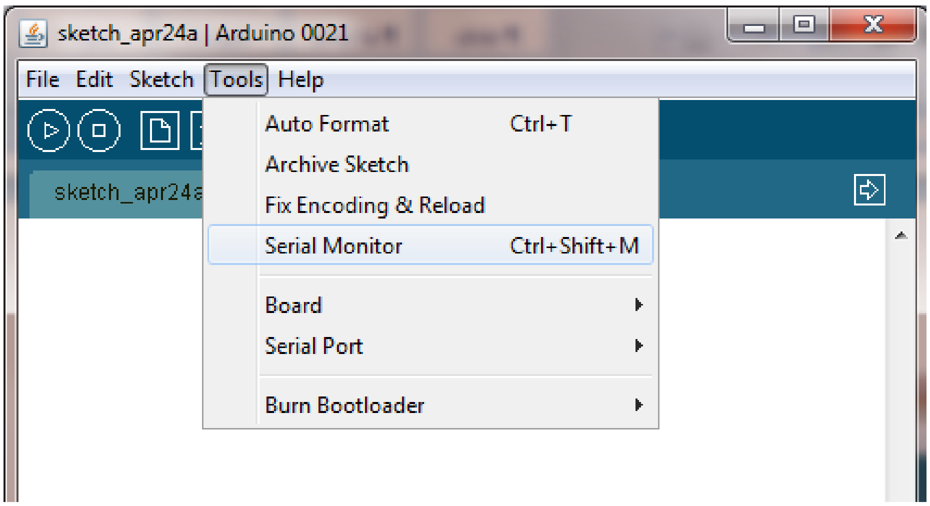


Figure 5.6: Displaying Serial Monitor

Figure 5.7 shows the serial monitor on which the results are displayed to make sure that the code works properly:

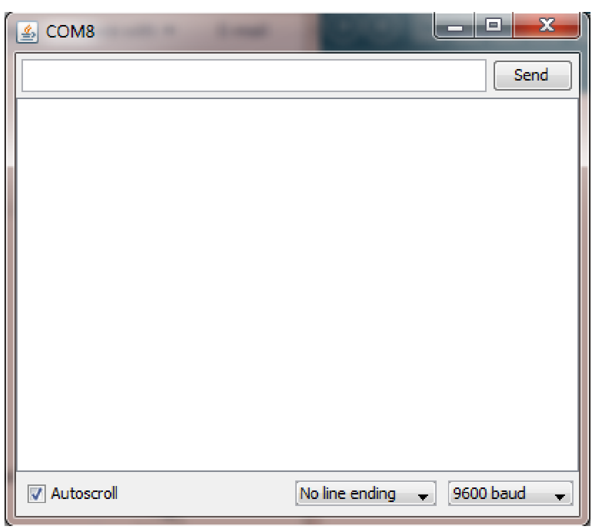


Figure 5.7: Serial Monitor

Arduino Libraries

Libraries are files written in C or C++ (.c, .cpp) which provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, it can be selected from Sketch >> Import Library, as shown in Figure 5.8.

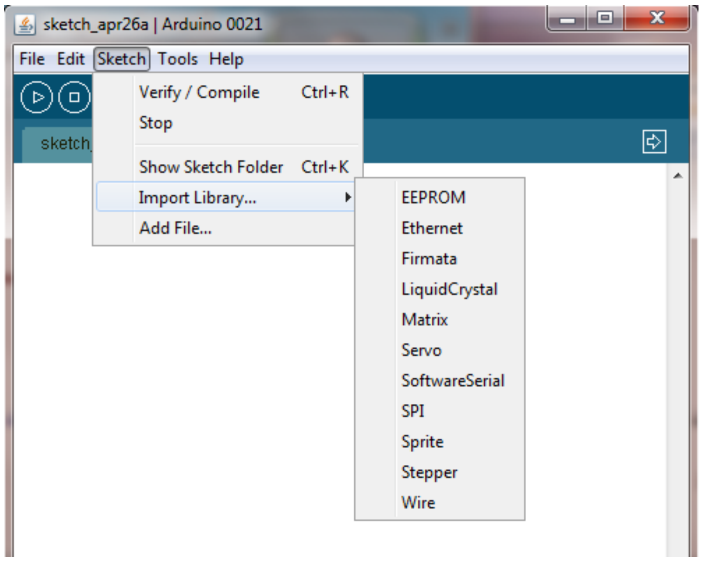


Figure 5.8: Adding Library

If the library does not exist, it must be installed first. To do so, the library should be downloaded.

5.3 Electricity Meter Interfacing Circuit Programming:

The resulted digital signal from the electricity meter interfacing circuit has been entered to the Ardiuno microcontroller through digital Pin 7. To read this signal, digitalRead( ) command should be used.

If the returned value from the digitalRead( ) command is HIGH, then the pulses counter (m) will increase by one. For every 1000 pulses the Kilowatt counter (kW) will also increase by one, and the pulses counter will start counting from zero. See Figure(5.9).

Figure 5.9: Electricity meter interfacing circuit code.

5.4 Xbee series 2 modules programming :

5.4.1 Xbee configuration:

Before we connect the Xbee modules to communicate with each other , we should configure them to specify each one's type. This can be implemented by a program called X-CTU.

Configuration & Test Utility Software (X-CTU) is a Windows-based application provided by Digi. This program was designed to interact with the firmware files found on Digi’s RF products and to provide a simple-to-use graphical user interface to them

Steps of Xbee configuration:

- First step: com port test

After Running the X-CTU software. Single-click on the USB COM port that the XBee is connected to. Then click on “Test/Query” button make sure that it's the right com port as shown un figure(5.10).

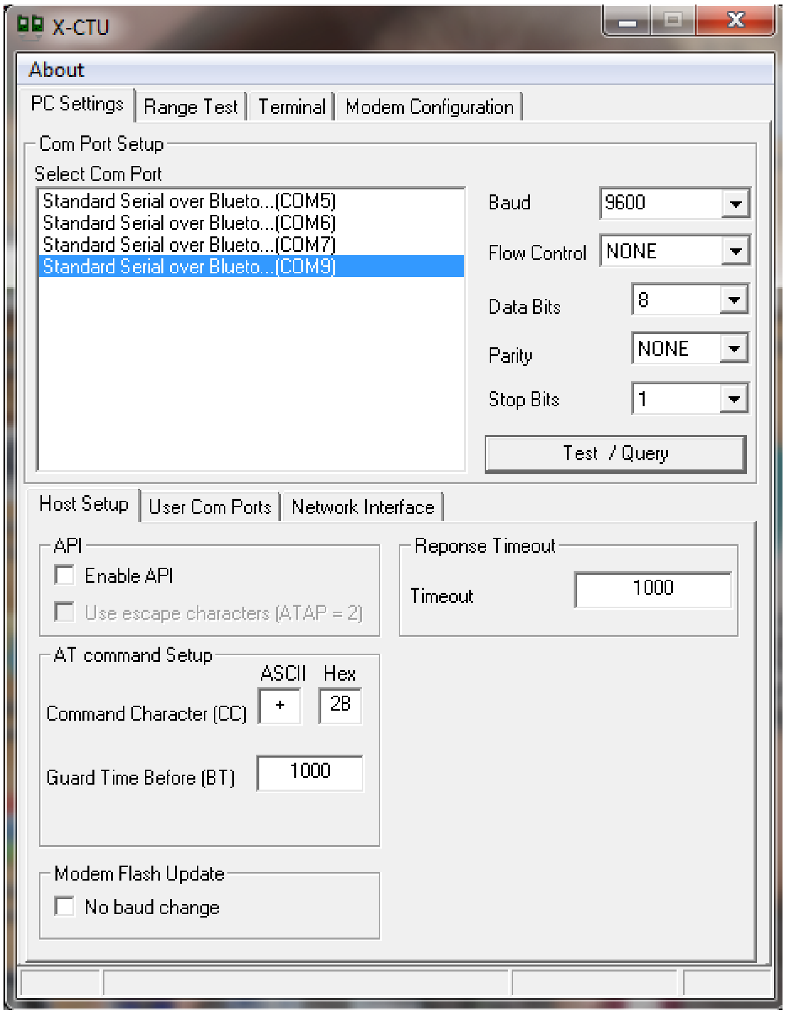


Figure 5.10: com port test

- Second step: Choosing the function of the modem

Choosing “Modem Configuration” >> "always update firmware". Then we choose the Xbee modem type " XB24-B" and one of the function as the firmware in the pull-down menu . For coordinator we choose " ZigBee Coordinator Device AT" as shown in figure (5.11) , and for the router or end device we choose " ZigBee Router/End device AT " as shown in figure(5.12).

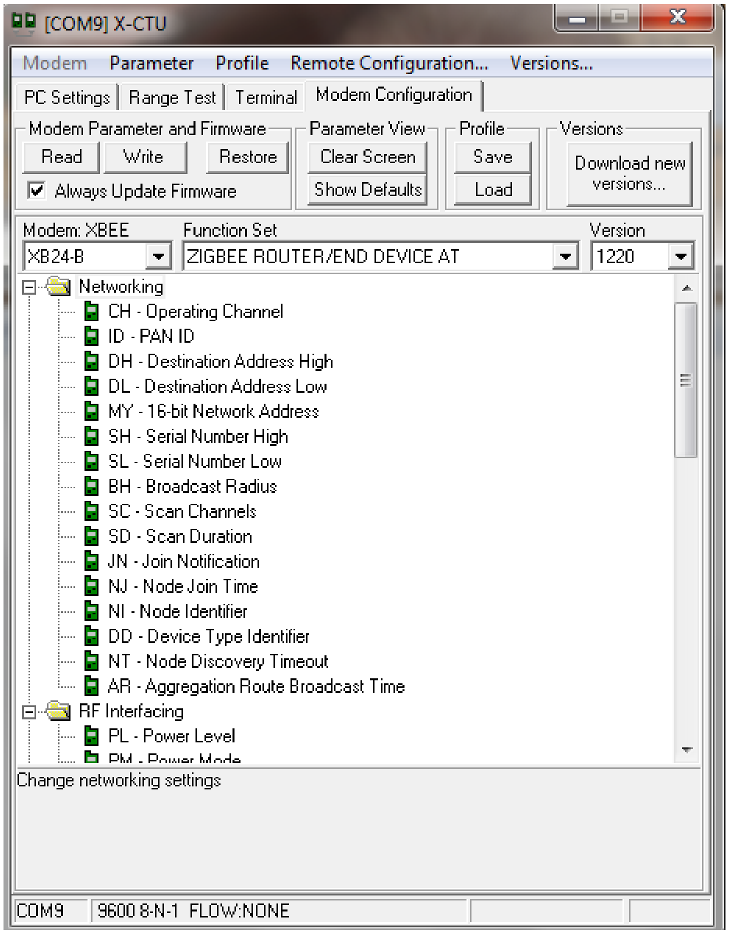


Figure 5.11: Coordinator device configuration

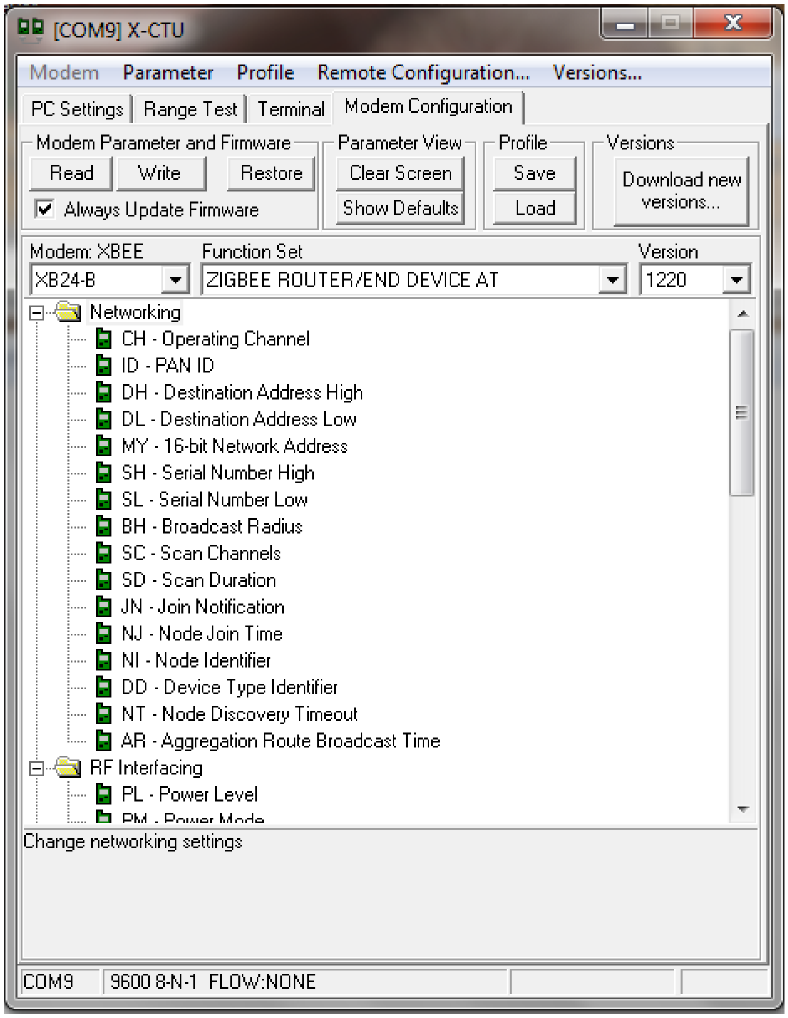


Figure 5.12: Router/End device configuration

- Third Step: Uploading the Xbee function type

Changing the “PAN ID” to a unique number (1FFE) to make our own Xbee network.
And clicking the “Write” button.
As a result, a window assures that the configuration has completed correctly will appear, which means that the Xbee modules are ready to communicate with each other’s.

5.4.2 Xbee modems programming using microcontrollers :

At each node in the system, the microcontroller is programmed to deal with the Xbee modem using Pin 2 as receiver and Pin 3 as transmitter. It sends the data to the Xbee network when it’s time for sending readings to the gateway node or when it receives a request for reading or managing a specific meter. See Figure (5.13).

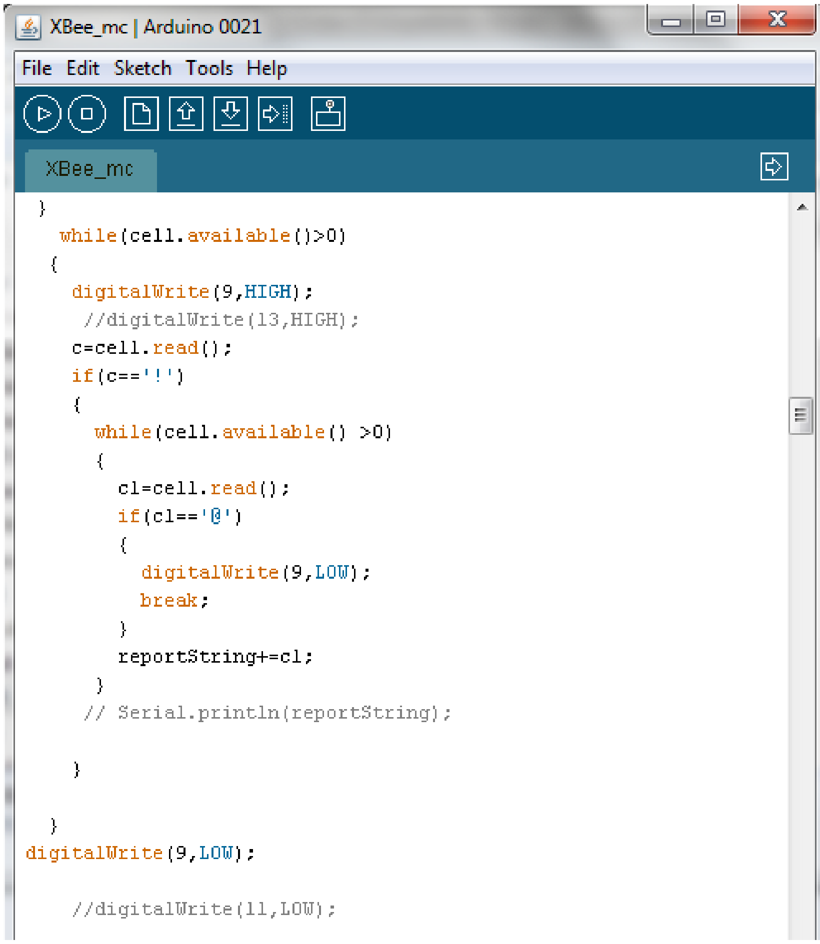


Figure 5.13: Xbee modems code

5.5 Programming Cellular Shield with SM5100B using microcontrollers :

At the gateway node we have programmed the microcontroller to deal with the Xbee modem and also with the GSM modem. The microcontroller listens to the GSM modem serial port using (while (cell. Available()>0)) command, when it receives a message from the GSM modem, data in the message will be read from the first character (!) to the last one (@) and analyzed . This message has three choices:

1- A request for sending readings of all meters that are connected to the same gateway. We used "gett" an indication for this purpose.
2- A request for sending the reading of a specific meter.

3- A managing message to connect or disconnect the electricity for a specific meter.

Steps for Programming Cellular shield with SM5100B

The Arduino has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL serial communication, which is available on digital pins 0 (RX) and 1 (TX). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board via a USB connection.

Serial communication has been used for communication between the Arduino board and the cellular shield. The Arduino hardware has built-in support for serial communication on pins 0 and 1 (which also goes to the computer via the USB connection). The native serial support happens via a piece of hardware (built into the chip) called a UART.

A universal asynchronous receiver/transmitter is a type of "asynchronous receiver/ transmitter", a piece of computer hardware that translates data between parallel and serial forms.

The NewSoftSerial library has been developed to allow serial communication on other digital pins of the Arduino, using software to replicate the functionality.

First we added the NewSoftSerial library to the Arduino program, by selecting sketch >> import library >>NewSoftSerial, as shown in Figure 5.14.

NewSoftSerial library used to send serial commands to the cellular module from any digital pin on Arduino.

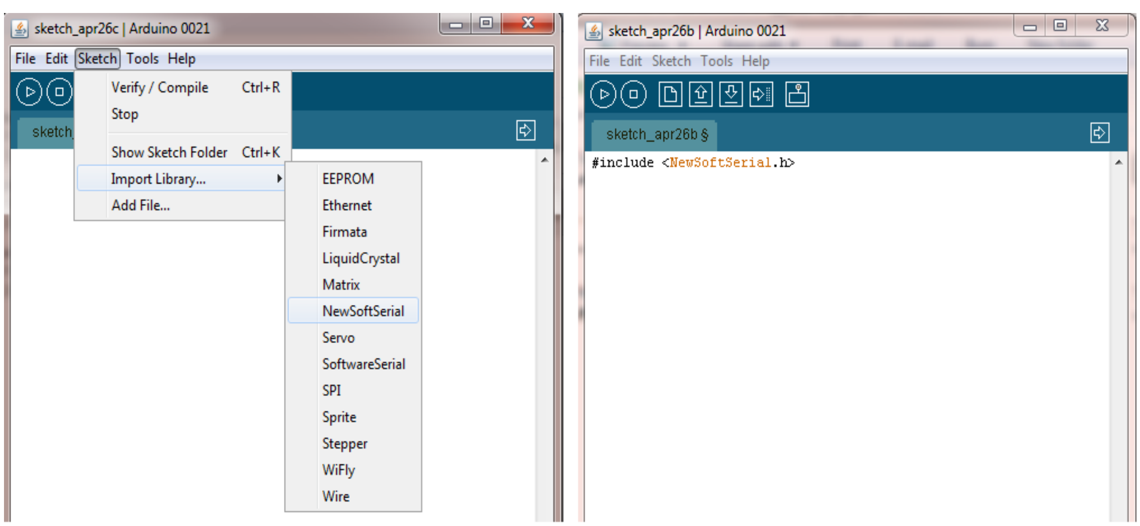


Figure 5.14: Adding NewSoftSerial Library

After adding the library, we need to create a serial port on D2/D3 (rxPin, txPin) to communicate with the GSM modem by the NewSoftSerial cell (2,3) command.

To send data from Arduino microcontroller to the transmitter pin “D3” of the software serial port in GSM modem by using the cell.print ("Data that you want to send") command.

After that we checked if the setup is functioning correctly, this is done through specific code. We made sure the SIM card is set to not require a PIN when the phone is turned on. After the code is uploaded to Arduino, the following screen - Figure 5.15 - can be shown by opening the serial monitor box in the Arduino program.

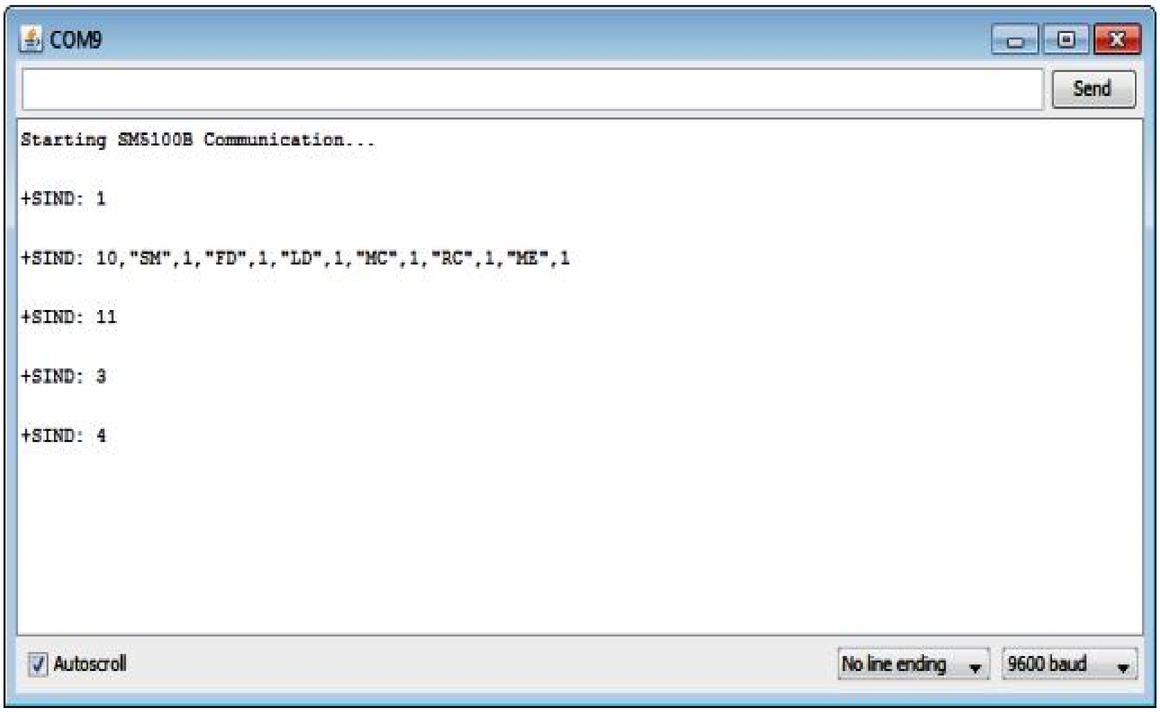


Figure 5.15: Results of GSM Modem Initialization

+SIND: 1, means the SIM card has been inserted.
+SIND: 10 line, shows the status of the in-module phone book. Nothing to worry about there for us at the moment.
+SIND: 11, means the module has registered with the cellular network.
+SIND: 3, means the module is partially ready to communicate.
+SIND: 4, means the module is registered on the network, and ready to communicate.

Now we can control the cellular shield with SM5100B by AT commands. The AT commands are instructions used to control a modem. AT is the abbreviation of ATtention.

Every command line starts with "AT" or "at". That's why modem commands are called AT commands.

In this system we are interested in receiving a request as a data call from the electricity company, then sending the appropriate data as a data call also, through the cellular shield with SM5100B (GSM modem) that support GPRS technology.

GPRS Registration:

Figure (5.16) shows the code of GPRS initialization that programmed in the microcontroller.

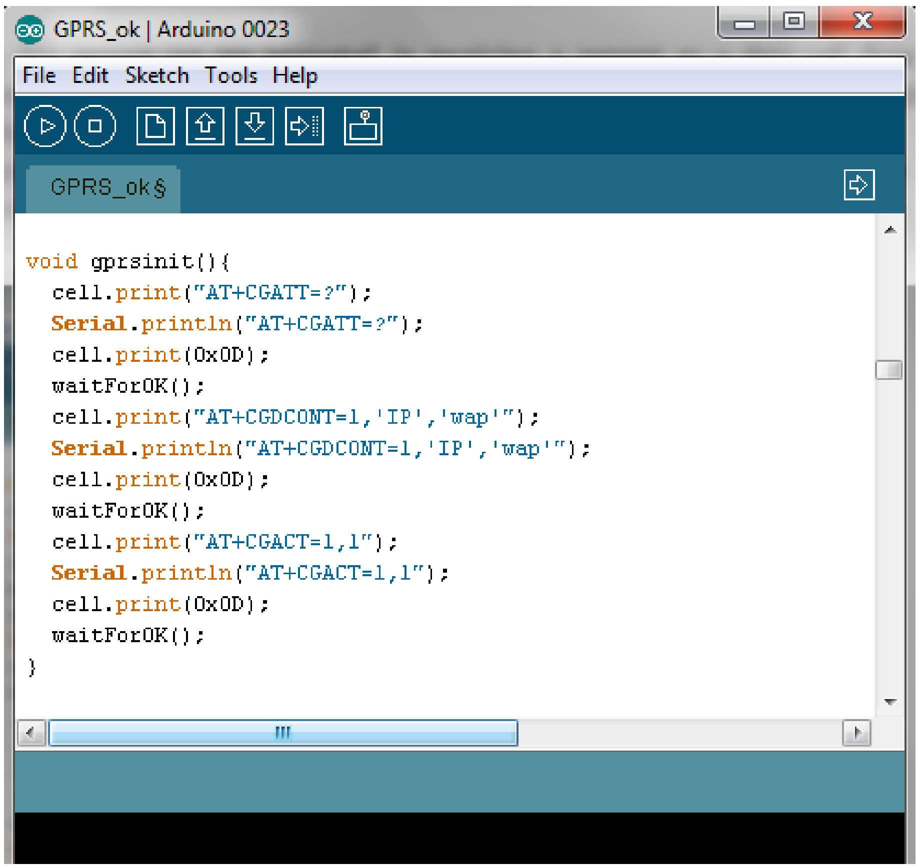


Figure 5.16: GPRS initialization code

The result can be seen in the serial monitor icon in the Arduino program as shown in figure (5.17). The

1) AT+CGATT: means that GPRS attachment is done.
2) AT+ACGCONT: shows the used parameters
1: connection ID.
IP: we use IP connection.
Wap: the access point name.
3) AT+CGACT: means that GPRS is activated.

4) At+sdataconf: means that configuration for TCP/IP is done and shows the port number with GSM network.
5) At+sdatastart: start the data call communication.
6) At+sdatastatus: Query sockets status and every socket communication information.

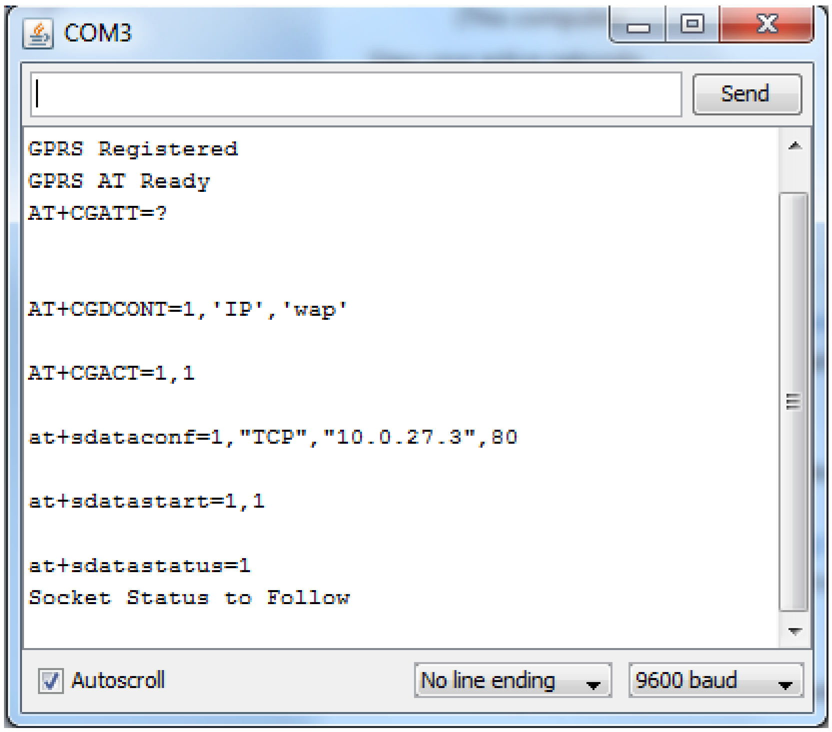


Figure 5.17: The results of GPRS registration.

5.6 Management code Programming:

Once the managing request arrives at the Gateway node's microcontroller through the GSM modem, data will be analyzed. If it's a managing request, a decision should be taken. If it's a managing request for a Gateway node meter, the appropriate new status will be achieved, if not, it will be routed through the Zigbee network to the Xbee modem that connected to the wanted meter.

When the request arrives the microcontroller that connected to the wanted meter , a HIGH signal will be sent to the relay that connected to the feeder of the electricity meter, to change its status from N.O(normally open) to N.C(normally close) , or vice versa .

5.7 Security code programming:

All components that connected to the meter will be saved in a security box. Using a wire with a V_{cc} voltage connected to the screw of the box, and connected with the microcontroller with a HIGH signal, we can achieve a touch sensor. Once the box is opened, the wire gives a LOW signal arrives to the microcontroller which will disconnect the electricity of the meter and send an alarm to the electricity company through the Zigbee and GSM networks. The alarm appears at the graphical user interface and the appropriate security procedure will be made.

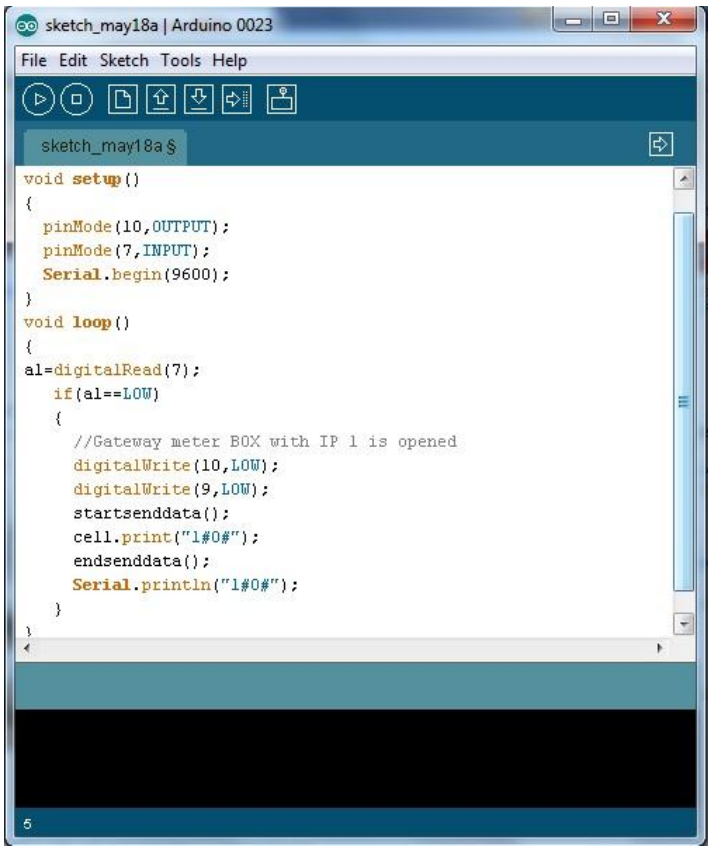


Figure5.18: The Security code

5.8 Graphical user interface and database Programming

Programming the database and the graphical user interface is implemented by Visual Studio program with C# language. The main window of the graphical user interface is shown in figure (5.19). To know more about the main functions in the code of each window, check the appendix.

First of all, we should check the com port that connected with the GSM modem. As shown in the figure, there is an icon for opening the com port and a status icon to view if it's connect or disconnect.

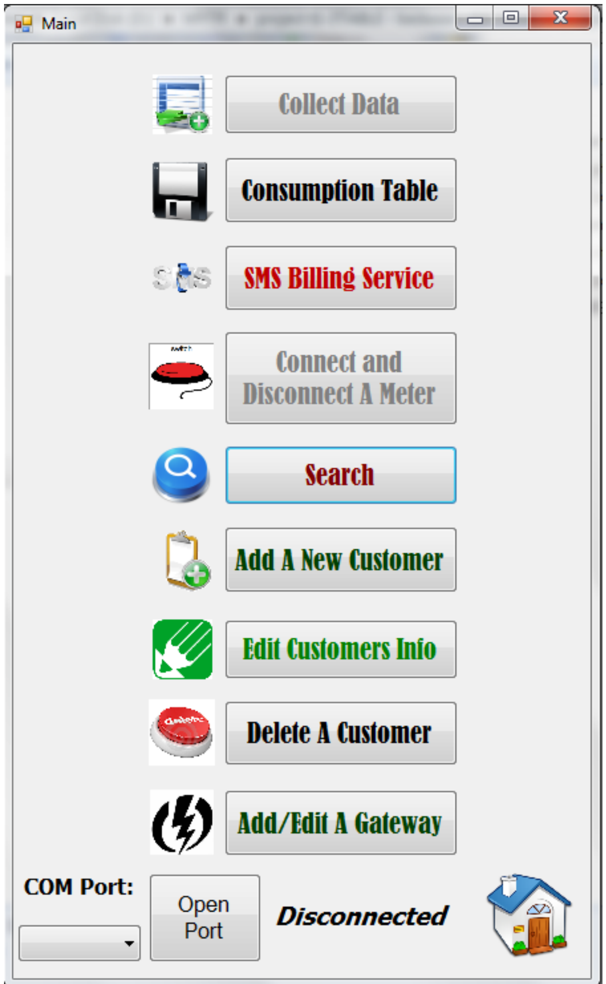


Figure 5.19: The main window of the graphical user interface

The contents of the main window are:

1) Collect Data: after making sure that the com port that is connected to the GSM modem is open, this window enables us to send requests for all gateway nodes or for a specific meter to send their readings.

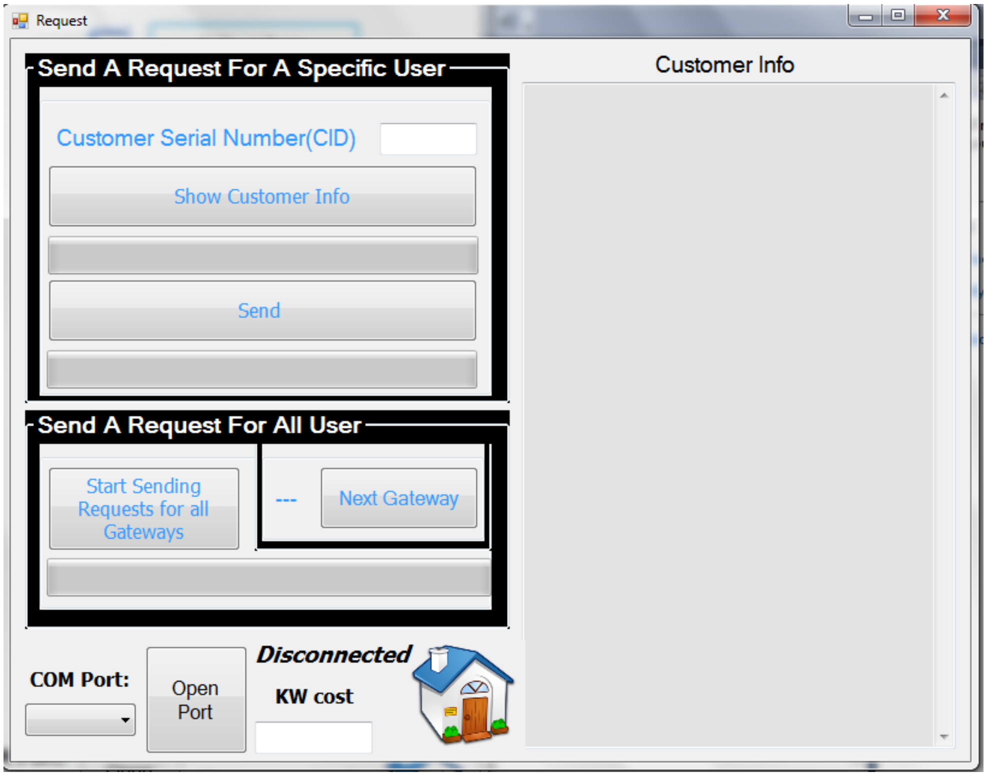


Figure 5.20: Collect data window

2) Consumption Table: it's used to schedule the received readings from meters, showing the date and time of recieption. The bills will be calculated and saved before sending them the customers.

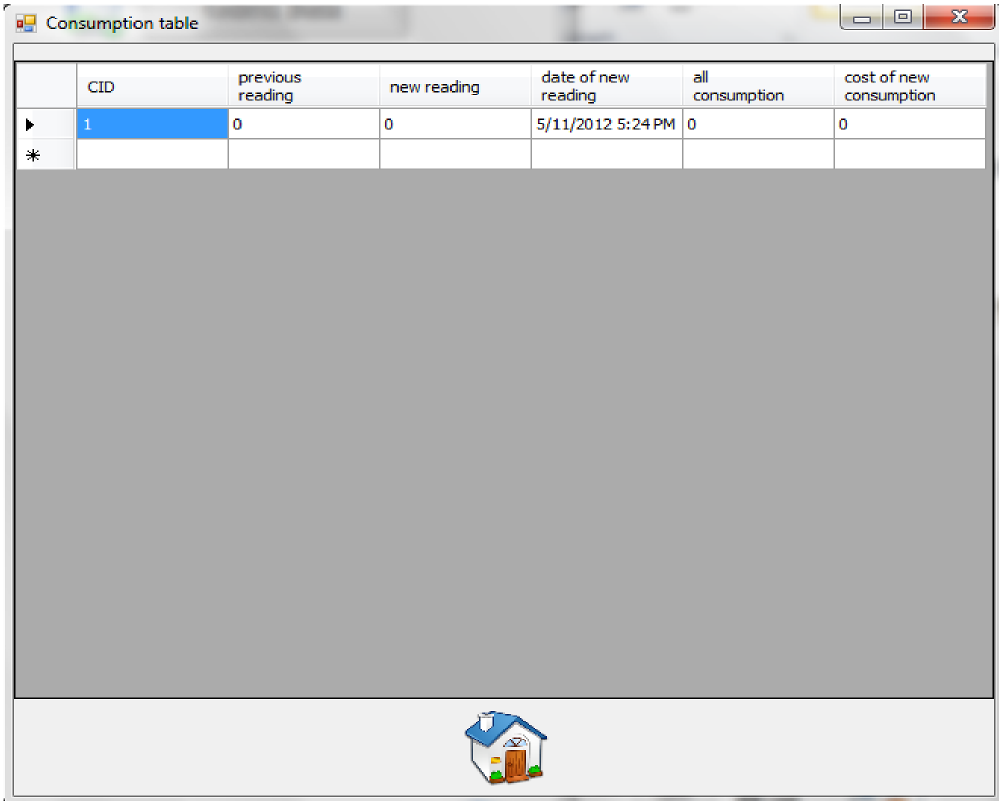


Figure 5.21: Consumption table window

3) SMS billing service: to send the electricity bill for each customer using SMS service or to resend it for a specific user.

Figure 5.22: SMS billing service window.

4) Connect/Disconnect electricity: controlling the meters in the system can be implemented since we have a two way communication. This window enables us to connect or disconnect the electricity for a specific meter.

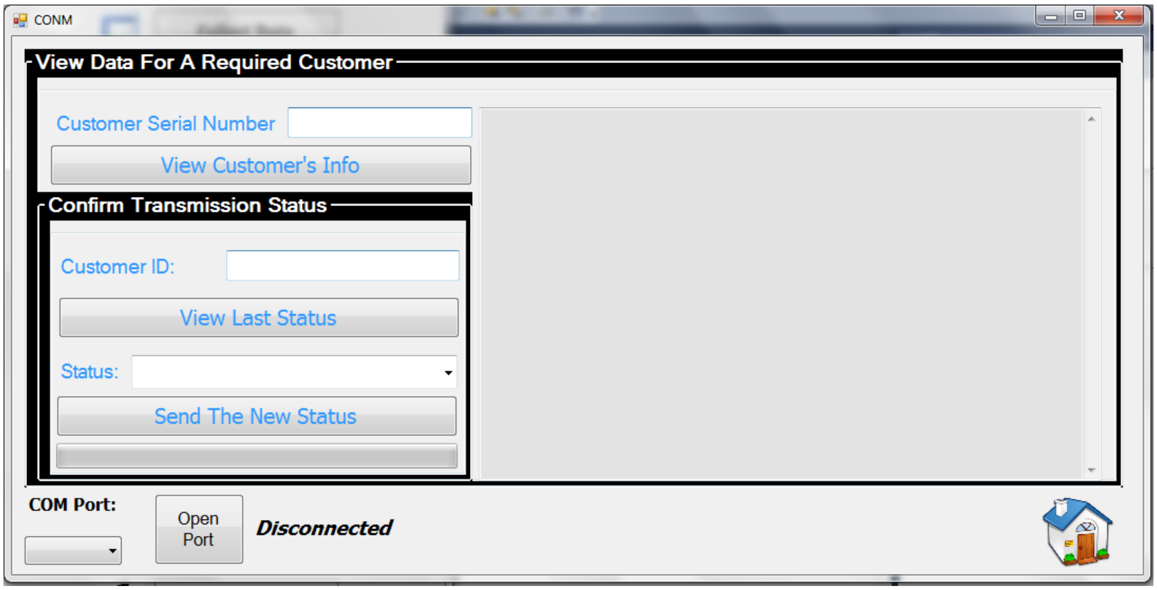


Figure 5.23: Connect/Disconnect window.

5) Editing users' information, Adding/deleting a customer and searching for a specific customer can be achieved as shown in the following windows.

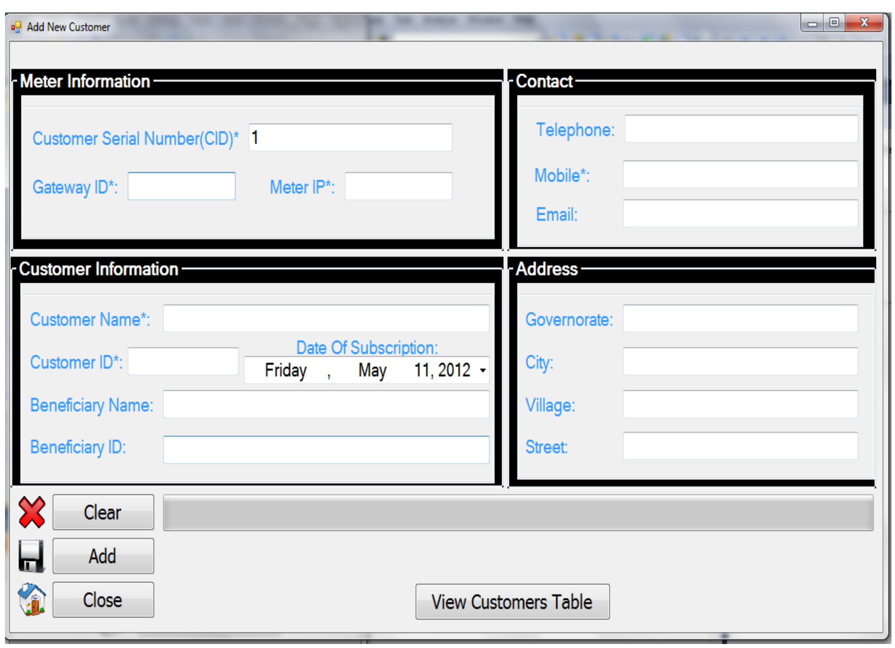


Figure 5.24: Adding a new customer window.

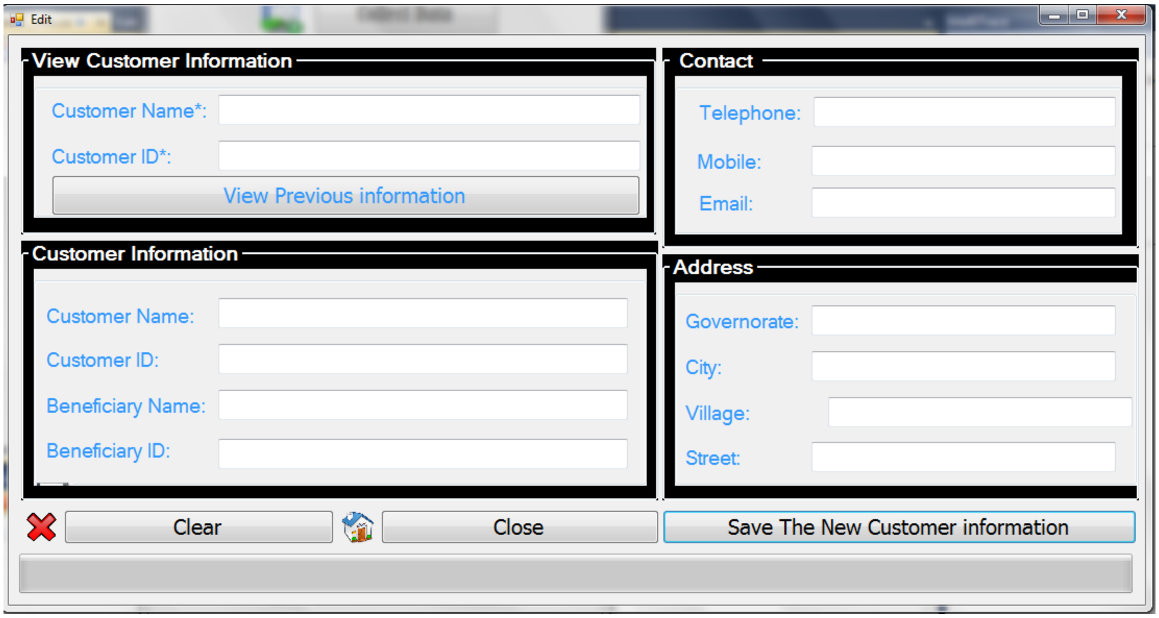


Figure 5.25: Edit customer info window.

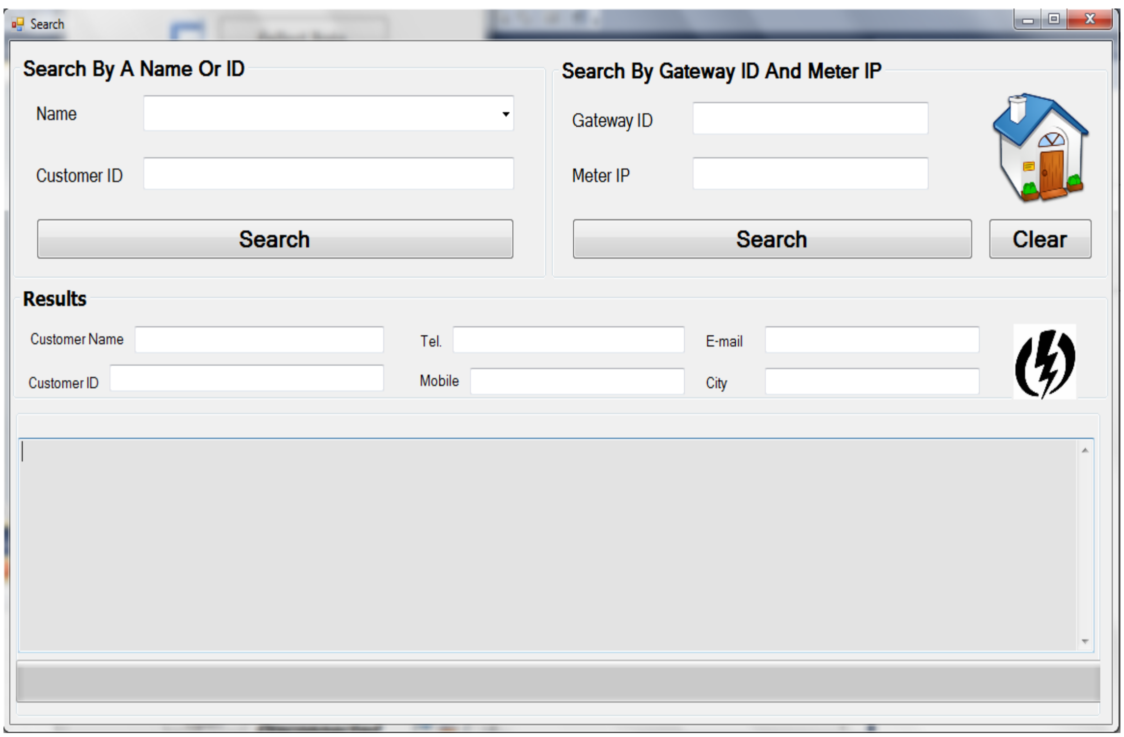


Figure 5.26: Search window.

Figure 5.27: Delete window.

6) Adding/Editing a Gateway node, showing SIM number and IP of GSM modem .

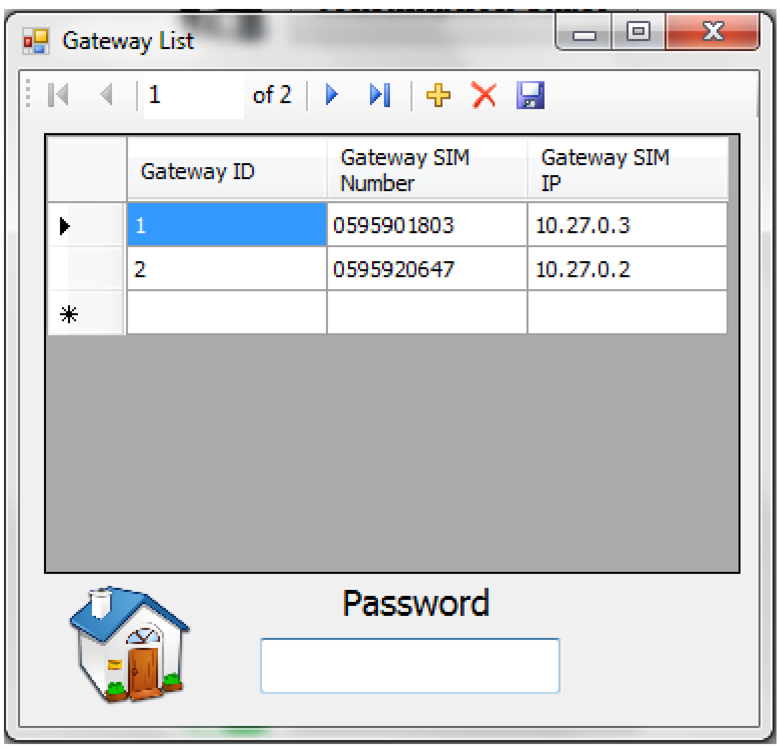


Figure 5.28: Adding/Editing a Gateway node window.

6

CHAPTER SIX

TESTING AND RESULTS

6.1 Introduction

6.2 System Testing

6.3 Testing results

6.1 Introduction:

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

This chapter shows all measurements needed to evaluate the performance of this system such as delay, and coverage.

6.2 System testing:

Checking and testing of the LDR interfacing circuit, microcontrollers and transceivers are individually illustrated in this section in addition to the testing results.

6.2.1 Electricity meter interfacing circuit testing:

LDR interfacing circuit has been implemented and tested if it gives the results in volts. After that the connection between this interfacing circuit and Arduino microcontroller has been tested. See Figure 6.1.

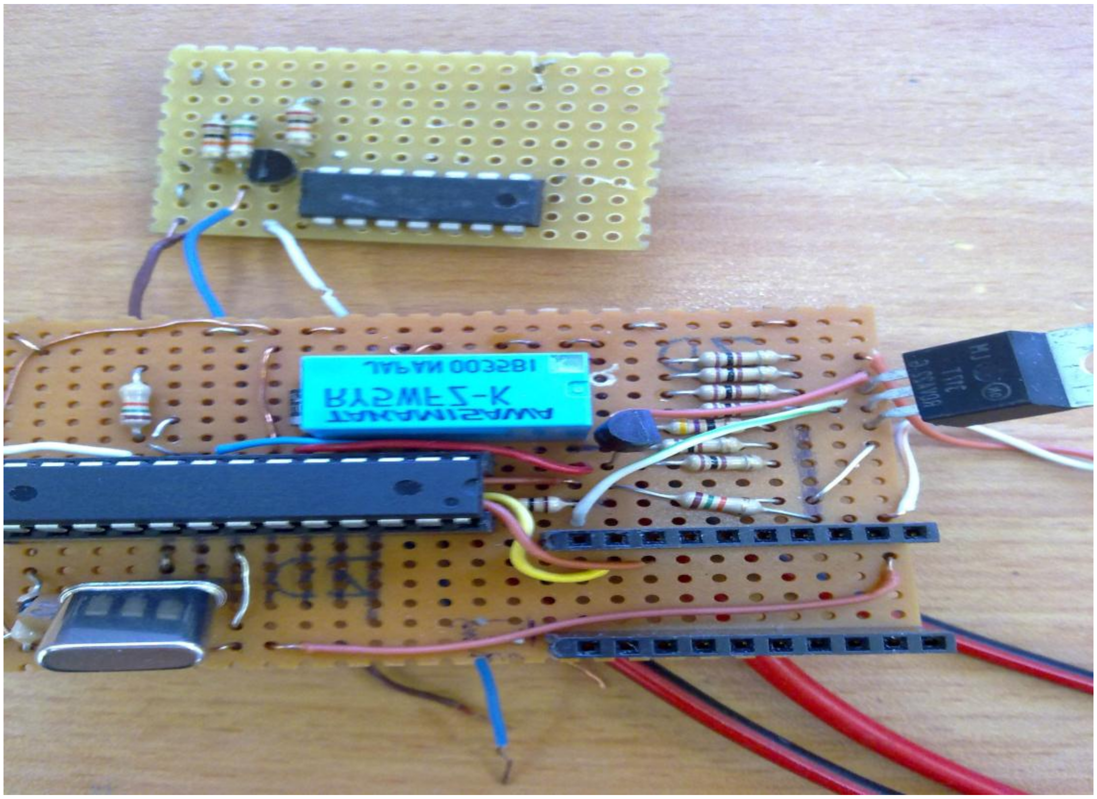


Figure 6.1: Connection between electricity meter interfacing circuit and Arduino microcontroller.

The resulted pulses were added together and used to calculate the kilowatt consumption by a certain process at the Arduino microcontroller

A test was performed on a digital electricity meter, where we fixed the LDR at the wanted led on the meter .The circuit gives a pulse, which equals to one watt of electricity consumption, only when the led is on as it is expected. Figure 6.2 shows the result at the serial monitor of the Arduino program.

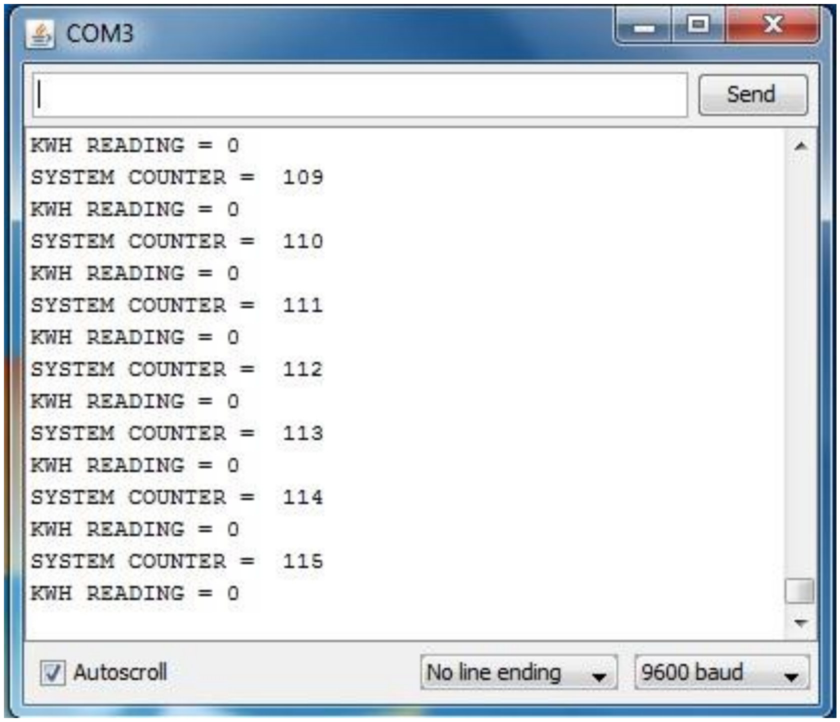


Figure 6.2: Result of electricity meter interfacing circuit

6.2.2 GSM modem :

The connection between the Cellular Shield with SM5100B transceiver and the Arduino was successfully achieved. After that we applied a test to initialize the GSM module.

After initializing the GSM module, the module has registered in the cellular network. As a result the module is partially ready to communicate and locate the network. See Figure 6.3.

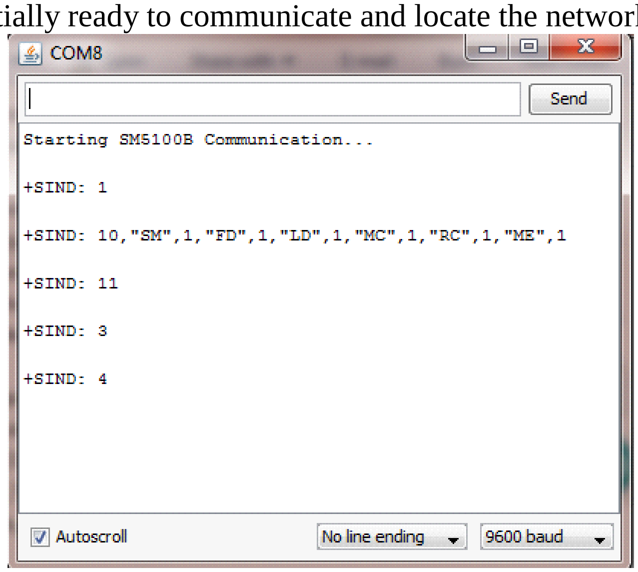


Figure 6.3: Result of Initializing GSM Module

Then the GPRS communication has started, as a result the module is partially ready to communicate and locate the network. See Figure 6.4.

.

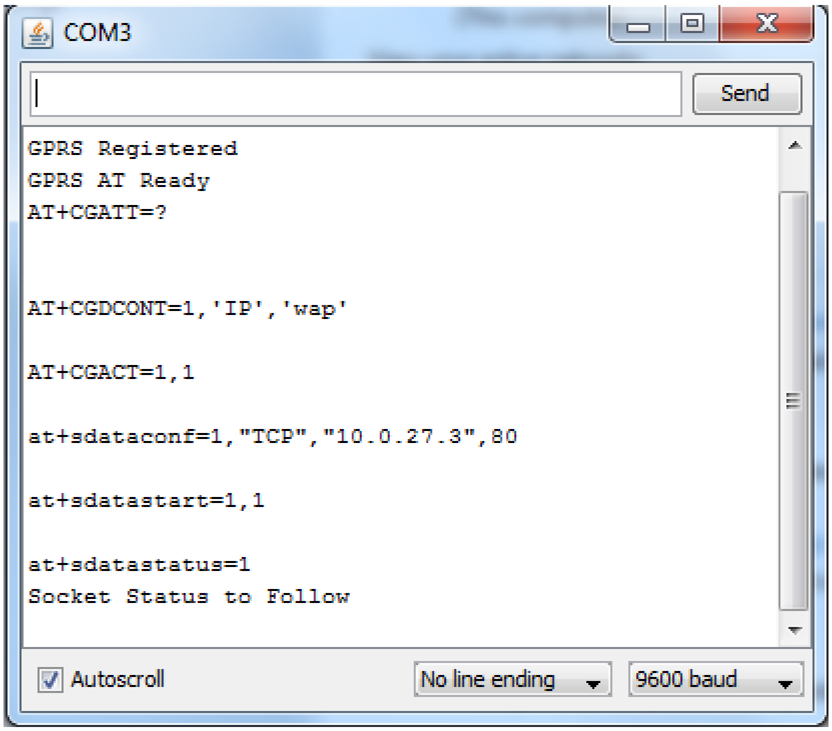


Figure 6.4: Result of GPRS connection.

The next step was to test the transmission of the data call from the GSM module at the Gateway node to the GSM module at the electricity company. Figure (6.5) result of receiving data for a testing code.

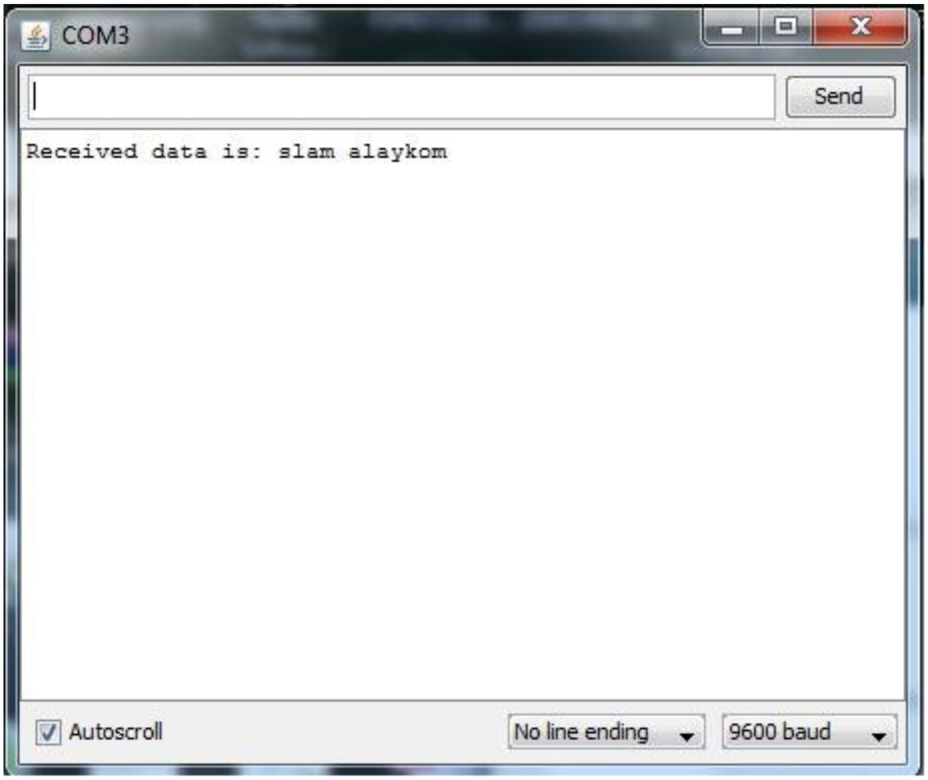


Figure 6.5: result of receiving data for a testing code

6.2.3 ZigBee Testing :

First, connection between the xbee module and the Arduino was tested. Then, the communication between the xbee modules was done successfully as shown in the terminal window of X-CTU program in Figure 6.6.

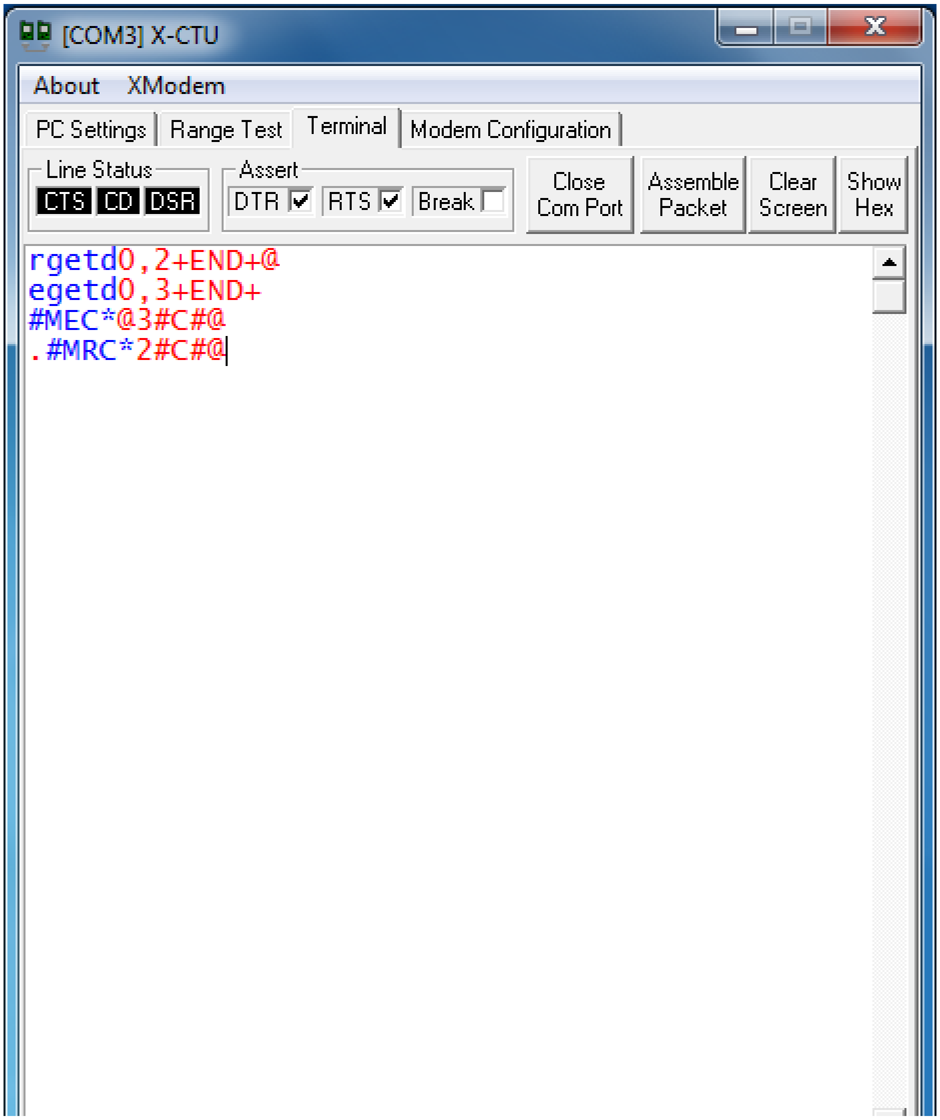


Figure 6.6: Result of xbee modules communication.

A connection between the xbee module and the Arduino microntroller as shown in figure 6.7.

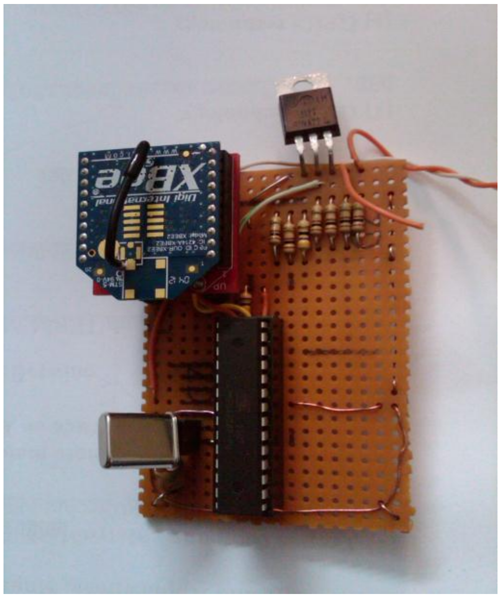


Figure 6.7: connecting Xbee with Arduino .

At the gateway node, we have added the GSM module to the whole system as shown in figure 6.8.

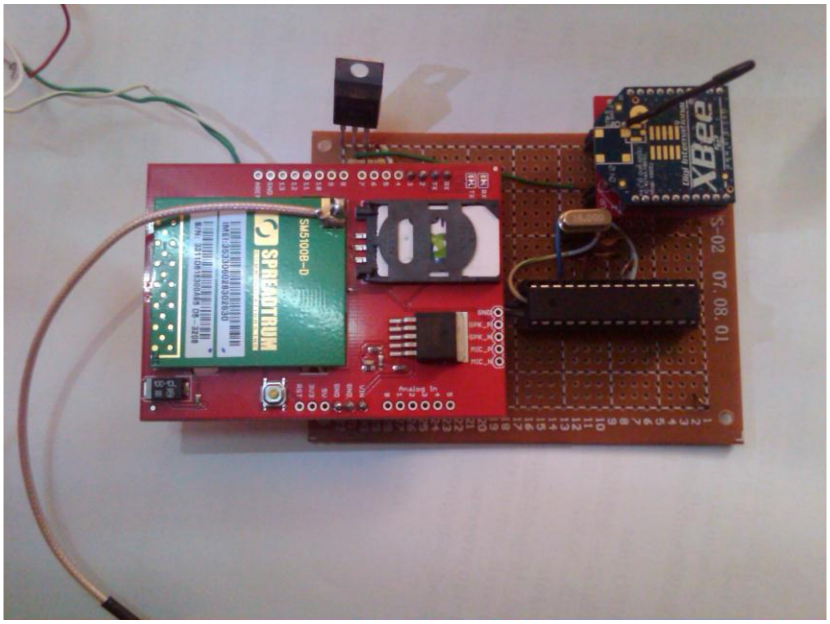


Figure 6.8: adding the GSM module to the gateway node

6.2.4 Management code testing :

After we have programmed the microcontrollers to analyze the requests, the result of a connect/disconnect request for the first meter is shown in figure (6.9).

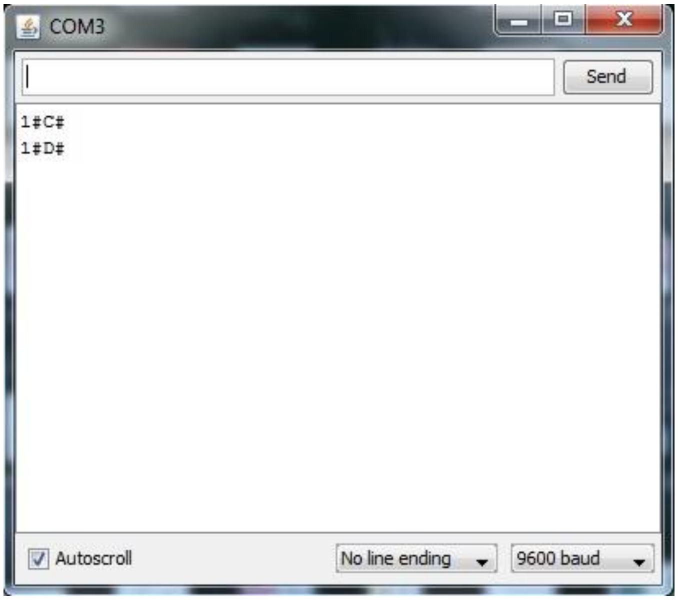


Figure 6.9: The result of a connect and disconnect request

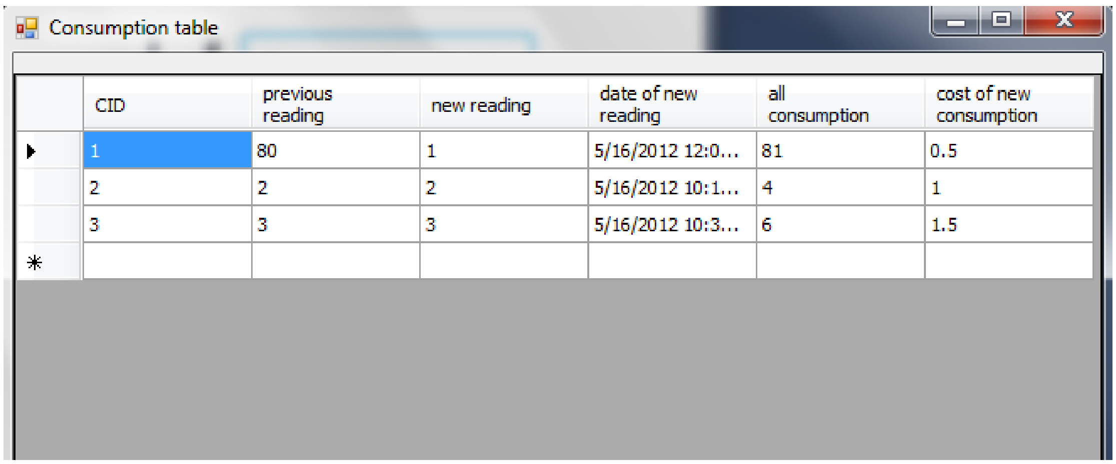
6.2.5 Safety code testing :

Figure (6.10) shows the received alarm of the first meter's opened box on the serial monitor of the Arduino.

Figure 6.10: A result of an opened box.

6.2.6 Database testing :

The data was sent if a request has been arrived to the GSM module at the gateway node from the electricity company. This data contains the meter number and the electricity consumption for each meter. Figure 6.11 shows an example of three electricity meters data at the database of the electricity company's PC.


Consumption table

CID	previous reading	new reading	date of new reading	all consumption	cost of new consumption
1	80	1	5/16/2012 12:0...	81	0.5
2	2	2	5/16/2012 10:1...	4	1
3	3	3	5/16/2012 10:3...	6	1.5

Figure 6.11: The collected data of three electricity meters.

If an alarm is received to the electricity company, a warning window will appear on the graphical user interface, contains the serial number of the meter that has been opened and the gateway that the meter belongs to. See Figure (6.12).

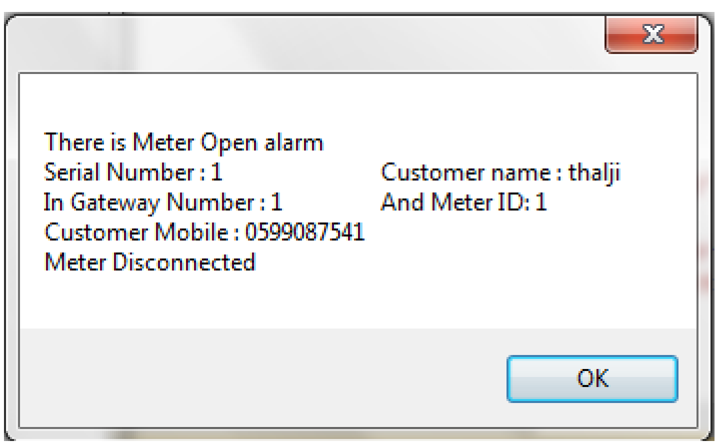


Figure 6.12: alarm warning window.

6.3 Testing Results:

6.3.1 Delay:

The delay defines how long it takes for an entire data call to completely arrive at the destination from the time the first bit is sent out from the source.

In this system, we have five delay forms. The first is measured when the data is collected at the Gateway node, this delay is about 200 ms.

the second is measured from the time the data collection request is sent from the electricity company to a Gateway node, and the data is completely sent back. This delay is about 7 seconds.

The second form is measured when a specific meter request to a sub node is sent, and the data is completely sent back. This delay is about 9 seconds.

The third is measured when a managing request is sent to a specific meter at a sub node, and an acknowledgment is sent back to the electricity company shows that the request is done. This delay is about 4 seconds.

The fourth is measured when a security box is opened, and an alarm is sent back to the company This delay is about 8 seconds.

It should be noted that the measurement is the average of multiple trials have been done.

6.3.2 Coverage

	GSM	ZigBee (2 mW)
Coverage	Several miles	180 m (non line of site)
Data rate	9.6 Kbps	250 Kbps

7

CHAPTER SEVEN

Recommendations

CHAPTER SEVEN

RECOMMENDATIONS

These are some developmental recommendations for moving forward:

1- Expanding the application to be implemented for several public services such as water and gas meters.

2- Building a new system contains an electricity meter with build in wireless system, having the features, capabilities and infrastructure of this system.

3-Including more parameters and information that received from the meter to the database at the company.

APPENDIX

DATA SHEETS

For

ACE 5000 digital electricity meter

ATMEGA 328P microcontroller

Arduino UNO board

SM5100B GSM Module

XBee module

ACE 5000 digital electricity meter

1 GENERAL

Meter Type	ACE 5000 type 5.1 Three Phase Time Switch Static Meter	
LifeTime	15 years (<0.5% drop-out rate/year)	
Enclosure Protection Degree	Overall Terminal Block	IP 54 IP xxB
Temperature Range	Specified Operating Range Limit Range of Operation Storage & Transport	-25°C to 60°C -40°C to 70°C -40°C to 80°C
Humidity Range	Specified Operating+C22 Range Limit Range of Operation Storage & Transport	45% to 90%RH ≤ 95% RH over 95% RH non condensing
Connection Type	3phases 3wires or 4 wires	Internal or External IP Link
Metrology Accuracy	Active Energy Reactive Energy	Class 1 IEC 62052/53 Class 2 IEC 62052/53
LED Constant	1000 pulses / kWh	In test mode 5000 pulses available per kWh or kvarh
Terminal Connection	DIN 46 857 part2	(asymmetric connection)
Voltage Operating Range	Vn (Un) = 3x230 (400)V / 3x400V or Vn (Un) = 3x127 (220)V / 3x220V Fully Redundant Power Supply	From -30% up to +25% (Operation without loss of accuracy with only two wires connected)
Limit Voltage Ranges	(Un) 0V to 500 V	(Withstands neutral rupture)
LV/DC Nominal / Maximum Current	5A / 120A	(Other Ib currents available for commercial purposes)
LV/CTNominal / Maximum Current	1A / 10A	(Other Ib currents available for commercial purposes)
Limit Current Ranges	Starting current Limit max current Withstands :	Ib/250 1.2 Imax 30.Imax for ½ cycle nominal frequency
Frequency	50, 60 Hz ± 10%	
Metrology Start Up	< 2.5s	
Continue Operation on V drops	0.5s @ [0.7 Vn .. 1.25 Vn]	0.2s - if only two wires [0.9 Vn .. 1.25 Vn]
Back-Up Power Supply Unit	Lithium Battery Super-Capacitor	3 years / 23°C ; Rated shelf life of 10years 7 days capacity; recharge less than 12hours
Overall Consumption	2W and 4VA capacitive at Vn	
Real Time Clock	± 0.5s/day @ 23°C with Main Power and + 0.1s/day/ °C for Specified Operating Range ± 1s/day @ 23°C with Back-Up Power Clock accuracy	According to IEC 61038 typically < 9s/month and < 3min/year for Specified Operating Range
Time Switch & Registers	4 Seasons with 8 Day Profiles & up to 8 Rates 12 End of Billing dates 72 Special Days	Leap year and Day-light Saving Time management Up to 18 historical Billing data
Energy Registers NR = non resetable [1..n] registers per Rate n = 8 for TSW only 3 energies possible, either P+, Q+, P- or P+, Q+, Q-	P+[1..n] (NR) P+Total (NR) Q+[1..n] (NR) Q+Total (NR) P- or Q- [1..n] (NR) P- or Q-Total (NR)	Bill.P+[1..n] — Hist.BillP+ [1..n] (kWh) Bill.P+Total — Hist.BillP+Total Bill.Q+[1..n] — Hist.BillQ+ [1..n] (kvarh) Bill.Q+Total — Hist.BillQ+Total Bill.P- or Q-[1..n] — Hist.BillP- or Q-[1..n] (kWh or kvarh) Bill.P- or Q-Total — Hist.BillP- or Q-Total
Maximum Demand Registers	MDI.P+ or S+ (W or VA) MDI.P+ or S+[1..n] (W or VA)	Hist.MDIP+ or S+ (W or VA) Hist.MDIP+ or S+[1..n] (W or VA) Number of Sub-intervals N = 1 to 5, 10 and 15 Sub-Interval Duration in minutes ΔT= 1, 3, 5, 10, 15, 20, 30, 60 Block or sliding operation
Maximum Registers	Max.I1/2/3 (A) filtered MaxI1/2/3 [1..n] (A) filtered Max.P+ (W) filtered Max U1/2/3 (V) not filtered	Filter Time Constant = 20s, 5 or 10 minutes
InstantaneousValues	RMS Current per phase (A) RMS Voltage per phase (V)	Active Power (W) Apparent Power (VA) Power Factor
Display	Energy Resolution : 6+0 , 6+1 , 5+2 Digit size : 6mm x 8mm LCD display	Displaying information : Normal and Autoscrolling mode Special Function mode Test mode 2 Push Buttons : A: Scrolling button B: Reset button - sealable
Optical Port	IEC 61107 sealable optical port	Readout (all data) according to IEC 61107 Read/Write (single data values) according to DLMS/COSEM (IEC 62056)
Communication Speed	Communication : 9600 bps	
Data Bus Output or Pulse Output	Serial unidirectional Data Bus Ouput with information of Readout @ 1200 bps (set by default) Up to 2 Pulse Outputs S0 (DIN 43 864 & IEC 62053-31 type A) Programmable pulse constant and pulse length.	
Load Management	0, 1 or 2 Relay out-puts 5 A 250 V	Relays can be controlled directly, by rate change or by current threshold.
Insulation Strength	4 kV 50Hz for 1 min	
Impulse Voltage Insulation Strength	Mains connections Output connections	10 kV 12 kV
Immunity to HF Fields	30 V/m ≤ 2 GHz	
Size (W x H x D)	218 x 179 x 106 (without Terminal Cover)	272 x 179 x 106 (with Terminal Cover)
Weight	1.7 kg	
Anti-fraud devices	Optical port password protection Number of times in programming mode Number of reverse energy detections Check read index	Number of power outages Working time (total & by rate) Detection Terminal Cover Opening
Configuration Tool	AIMS 5000 with Cripkey protection	

ATMEGA 328P microcontroller

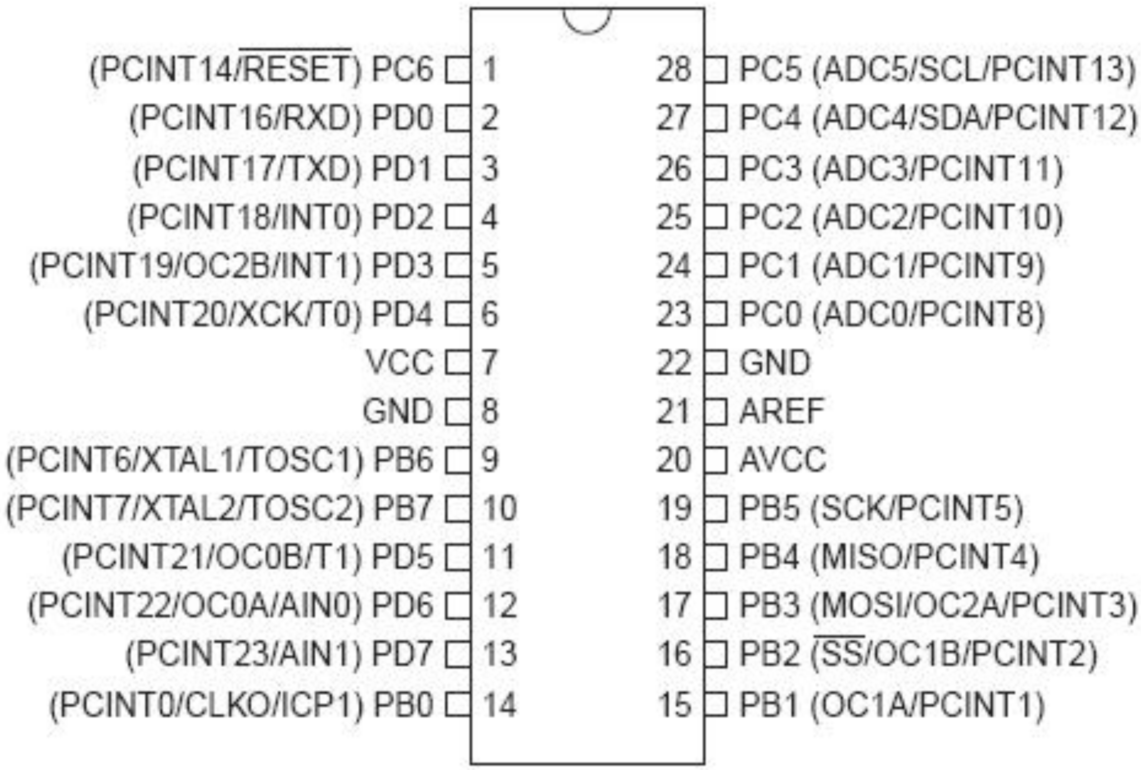
Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C[1]
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I^2C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (Including 32 kHz RTC)

8-bit AVR® Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

Rev. 8161D–AVR–10/09


(PCINT14/RESET) PC6 1
(PCINT16/RXD) PD0 2
(PCINT17/TXD) PD1 3
(PCINT18/INT0) PD2 4
(PCINT19/OC2B/INT1) PD3 5
(PCINT20/XCK/T0) PD4 6
VCC 7
GND 8
(PCINT6/XTAL1/TOSC1) PB6 9
(PCINT7/XTAL2/TOSC2) PB7 10
(PCINT21/OC0B/T1) PD5 11
(PCINT22/OC0A/AIN0) PD6 12
(PCINT23/AIN1) PD7 13
(PCINT0/CLKO/ICP1) PB0 14
28 PC5 (ADC5/SCL/PCINT13)
27 PC4 (ADC4/SDA/PCINT12)
26 PC3 (ADC3/PCINT11)
25 PC2 (ADC2/PCINT10)
24 PC1 (ADC1/PCINT9)
23 PC0 (ADC0/PCINT8)
22 GND
21 AREF
20 AVCC
19 PB5 (SCK/PCINT5)
18 PB4 (MISO/PCINT4)
17 PB3 (MOSI/OC2A/PCINT3)
16 PB2 (SS/OC1B/PCINT2)
15 PB1 (OC1A/PCINT1)

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 88.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

Arduino UNO board

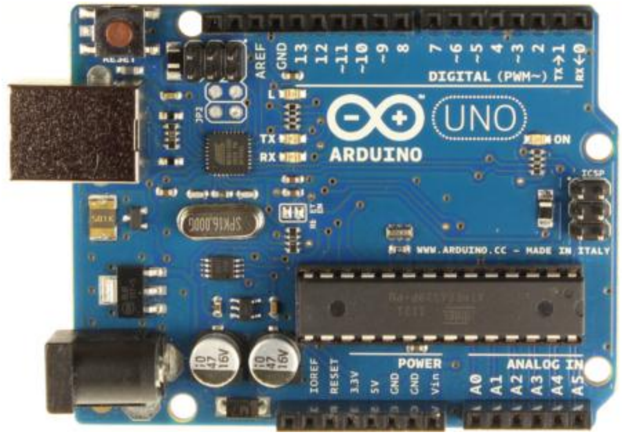


Figure A.1: The Arduino UNO board.

Overview

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz 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 Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

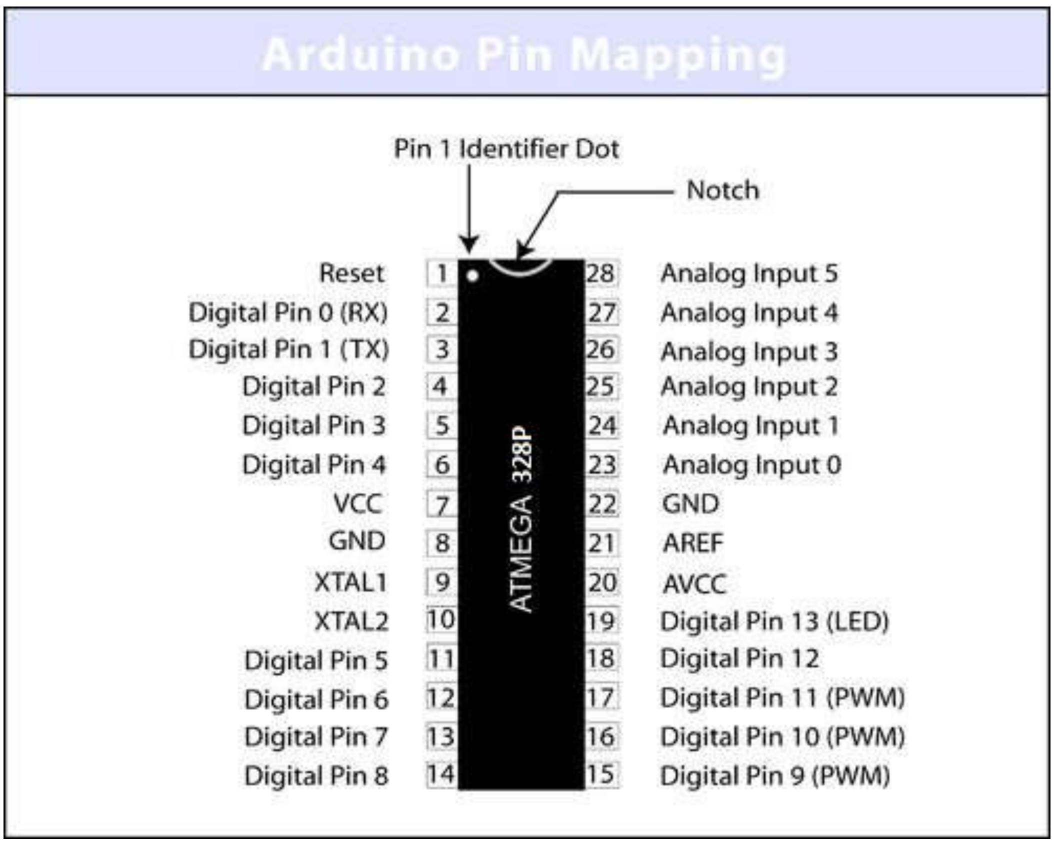
The board has the following new features:

- 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 both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate 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.

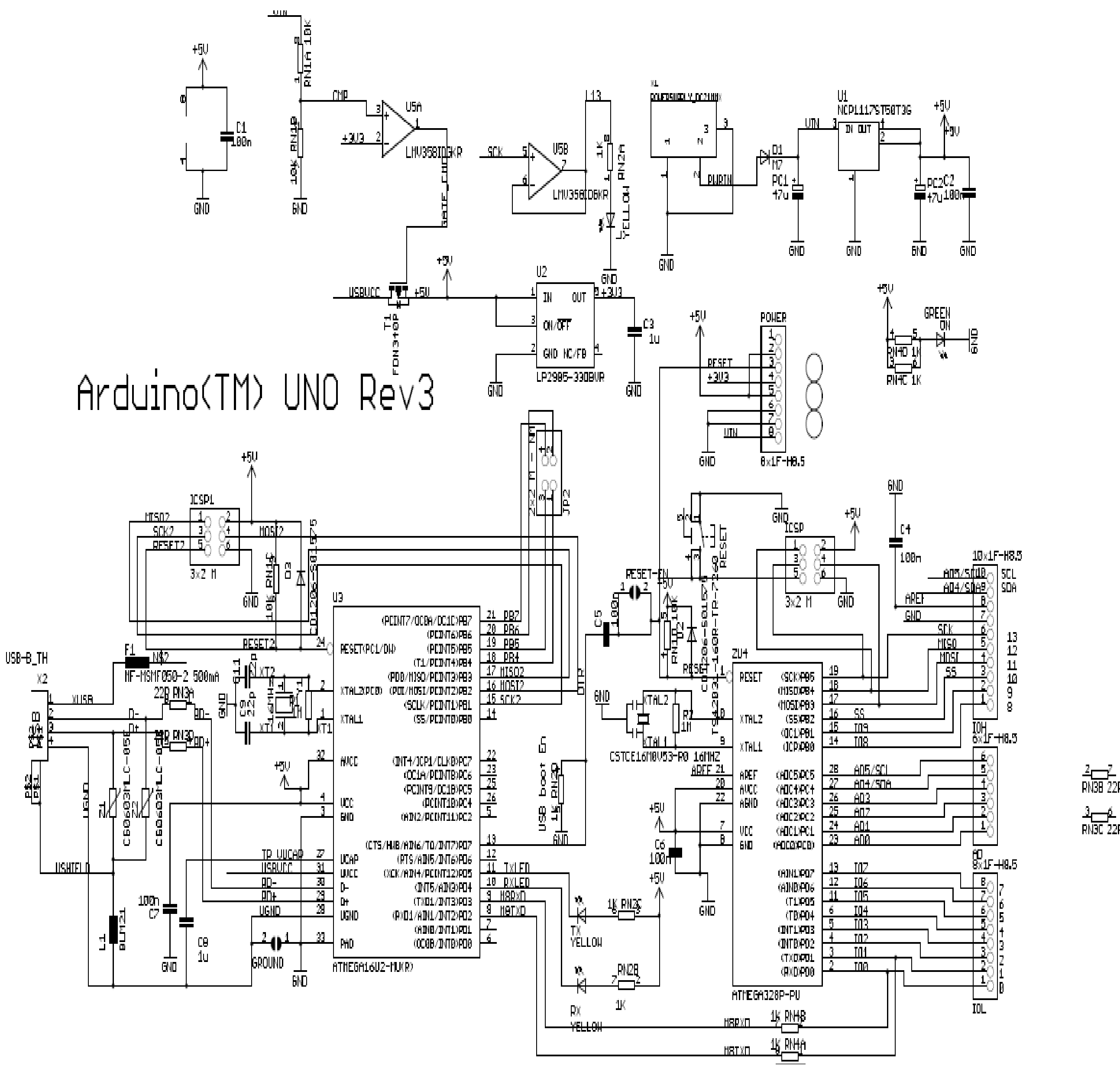
Summary:

- 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



SM5100B GSM modem

SM5100B-D GSM/GPRS Module Specification(Preliminary)

1 OVERVIEW

1.1 OBJECT OF THE DOCUMENT

This document gives an overview of the SM5210 module: a miniature, single-side board, quad-band GSM 850/EGSM 900/DCS 1800/PCS 1900 module, ready for integration in various kinds of Fix wireless phones and other wireless devices.

1.2 SYSTEM BLOCK DIAGRAM

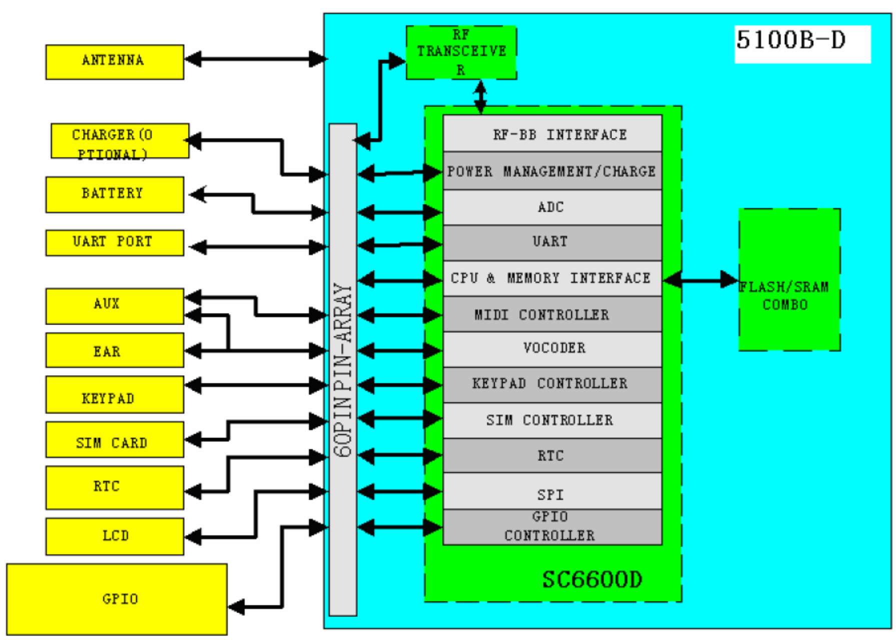


Figure 1: System block diagram

1.3 PRODUCT OVERVIEW

Temperature range	Normal range: -10°C to +55° C (full compliant) Storage: -40°C to +85°C
Weight	< 9g
Physical dimensions	35.0X39.0X2.9 mm (typical)

Connection	60 pins
Power supply	VBAT: 3.3V to 4.2V range, 3.6V typical.
Power consumption	Off mode: <100uA Sleep mode: <2.0mA Idle mode: <7.0mA (average) Communication mode: 350 mA (average,GSM) Communication mode: 2000mA (Typical peak during TX slot,GSM)
Li-ion Battery charging management and interface (OPTION)	Li-ion Battery charging management is included. The charger interface is provided on 60-pin connector. (only for 3.7V Li-ion Battery)
Frequency bands	EGSM900 +GSM850+ DCS1800+PCS1900
Transmit power	Class 4 (2W) for EGSM900/GSM850 Class 1 (1W) for DCS1800/PCS1900
Supported SIM card	3V/1.8V SIM card. (auto recognise)
Keyboard interface	4x6 keyboard interface is provided
UART0 interface with flow control	Up to 460 kbps Full hardware flow control signals (+3.0V) are provided on 60 pins.
UART1 interface without flow control	2-Wire UART interface Up to 460 kbps
LCD interface	Support standard SPI interface,

1.4 FUNCTIONAL DESCRIPTION

1.4.1 RF Functionalities

The RF part of this module converts RF signals to baseband for receiver chain and translates base band signals into RF frequency spectrum.
The operating frequencies are:

Rx (EGSM 850): 869 to 894MHz
Tx (EGSM 850): 824 to 849MHz

Rx (EGSM 900): 925 to 960MHz
Tx (EGSM 900): 880 to 915MHz

Rx (DCS 1800): 1805 to 1880MHz
Tx (DCS 1800): 1710 to 1785MHz

Rx (PCS 1900): 1930 to 1990MHz
Tx (PCS 1900): 1850 to 1910MHz

1.4.2 Baseband Functionalities

The baseband part of SM5210 is composed of a SPREADTRUM's SC6600D chip. This chipset is using 0.18µm mixed signal CMOS technology which allows massive integration as well as low power consumption. SC6600D provides single-chip solution to wireless Quad-band telephone handsets and data modems confirming to the EGSM 900, GSM 850, DCS 1800 and PCS 1900.

XBee module

Specifications

Specifications of the XBee®/XBee-PRO® ZB RF Module

Specification	XBee	XBee-PRO (S2)	XBee-PRO (S2B)
Performance			
Indoor/Urban Range	up to 133 ft. (40 m)	Up to 300 ft (90 m), up to 200 ft (60 m) international variant	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant
Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled	50mW (+17 dBm) 10mW (+10 dBm) for International variant	63mW (+18 dBm) 10mW (+10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps	250,000 bps
Data Throughput	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)
Serial Interface Data Rate (software selectable)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)
Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled	-102 dBm	-102 dBm
Power Requirements			
Supply Voltage	2.1 - 3.6 V	3.0 - 3.4 V	2.7 - 3.6 V
Operating Current (Transmit, max output power)	40mA (@ 3.3 V, boost mode enabled) 35mA (@ 3.3 V, boost mode disabled)	295mA (@3.3 V) 170mA (@3.3 V) international variant	205mA, up to 220 mA with programmable variant (@3.3 V) 117mA, up to 132 mA with programmable variant (@3.3 V), International variant
Operating Current (Receive))	40mA (@ 3.3 V, boost mode enabled) 38mA (@ 3.3 V, boost mode disabled)	45 mA (@3.3 V)	47 mA, up to 62 mA with programmable variant (@3.3 V)
Idle Current (Receiver off)	15mA	15mA	15mA
Power-down Current	< 1 uA @ 25°C	3.5 µA typical @ 25°C	3.5 µA typical @ 25°C
General			
Operating Frequency Band	ISM 2.4 GHz	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960 x 1.297 (2.438cm x 3.294cm)	0.960 x 1.297 (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector	Integrated Whip, Chip, RPSMA, or U.FL Connector	Integrated Whip, PCB Embedded Trace, RPSMA, or U.FL Connector
Networking & Security			
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Number of Channels	16 Direct Sequence Channels	14 Direct Sequence Channels	15 Direct Sequence Channels
Channels	11 to 26	11 to 24	11 to 25
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
Agency Approvals			
United States (FCC Part 15.247)	FCC ID: OUR-XBEE2	FCC ID: MCQ-XBEEPRO2	FCC ID: MCQ-PROS2B
Industry Canada (IC)	IC: 4214A-XBEE2	IC: 1846A-XBEEPRO2	IC: 1846A-PROS2B
Europe (CE)	ETSI	ETSI (International variant)	ETSI (10 mW max)

* Function is not supported at the time of this release

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 50k Ω pull-up resistor attached to $\overline{\text{RESET}}$
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

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] Tropos company , MetroMesh Tropos . Available at
http://www.tropos.com/pdf/technology_briefs/tropos_techbrief_amr.pdf

[3] Henrik Vikelgaard, Remote meter reading using the GSM / SMS system , 17th International Conference on Electricity Distribution, Barcelona, 2003.

[4] Microcontroller
http://www.webopedia.com/TERM/M/microcontroller.html

[5] Arduino microcontroller
http://www.arduino.cc/en/Main/arduinoBoardUno

[6] ZigBee technology
http://en.wikipedia.org/wiki/ZigBee

[7] GSM technology
http://www.3gamericas.org/index.cfm?fuseaction=page&pageid=371#_2 .GSM

[8] GPRS technology
http://www.mobilein.com/GPRS.pdf

[9] GPRS technology
http://en.wikipedia.org/wiki/General_Packet_Radio_Service

[10] GSM modem
http://www.nowsms.com/faq/what-is-a-gsm-modem

[11] Database
http://searchsqlserver.techtarget.com/definition/database

[12] Database
http://en.wikipedia.org/wiki/Database

[13] Arduino UNO pins
http://www.arduino.cc/en/Main/arduinoBoardUno

[14] XBee module
www.sparkfun.com/products/8742

[15] GoBetwino program
http://arduino.cc/playground/Interfacing/GoBetwino