DESIGN AND IMPLEMENTATION OF INTELLIGENT HOME USING GSM NETWORK

OMAR TALAL ALGOIARE

SEPTEMBER 2014

DESIGN AND IMPLEMENTATION OF INTELLIGENT HOME USING GSM NETWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY
OMAR TALAL ALGOIARE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING

SEPTEMBER 2014

Title of the Thesis: **Design and Implementation of Intelligent Home Using GSM Network.**

Submitted by **Omar Talal ALGOIARE**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements of a thesis for the degree of Master of Science.

Assist. Prof. Dr. Murat SARAN
Head of Department

This is to certify that we have read this thesis, and that in our opinion, it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV
Supervisor

Examination Date: 23.09.2014

Examining Committee Members

Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV	(Çankaya Univ.)	
Assoc. Prof. Dr. Hadi Hakan MARAŞ	(Çankaya Univ.)	
Assist. Prof. Dr. Fahd JARAD	(UTAA)	

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name	:	Omar Talal ALGOIARE
Signature	:	
Date	:	23.09.2014

ABSTRACT

DESIGN AND IMPLEMENTATION OF INTELLIGENT HOME USING GSM NETWORK

ALGOIARE, Omar Talal
M.Sc., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV

September 2014, 56 pages

Intelligent home devices such as weather, safety or security systems have become more advanced with the latest developments in Microcontroller and wireless technology, such as Wi-Fi, GSM, Bluetooth and ZigBee. For most of these devices, the difficulty lies in their graphical user interfaces and the number of controlling devices.

In the last few years with the development of wireless technologies, a good solution is yet to be found for the smart home system. In this thesis, an intelligent home system has been designed and implemented using the GSM network.

Generally, this system consists of two parts: hardware and software. The hardware consists of many units, mobile phones, computers, smart card readers, temperature,

gas, motion and rain sensors, a GSM module, a 9V charger, a Microcontroller, an SDRAM module and a buzzer alarm.

The first important hardware unit is the Microcontroller; it is the brain of this system. All hardware units connect to this Microcontroller. The Arduino Mega 2560 is used to build this system. Moreover, a SIM900 GSM/GPRS modem is used to connect with the GSM network. This modem is responsible for sending and receiving instructions and alerts. The SDRAM is used to save all events that occur in the home. Furthermore, the smart card reader using RFID technologies used to open and close the doors to specific persons and also to save any events in a specific text file in the SDRAM.

The software is the second part of this system. It is the GUI of the system and is built in the VB6 environment. The software is connected to hardware devices through USB ports. Moreover, it works to control and monitoring this system from the computer. The software consist of three parts: sensors reading values, electronic devices turning other devices on/off and real-time reading for monitoring instructions from the USB port.

This system works in two ways: portably with a 9-volt charger and with the computer. The job of the system is to monitor heat and gas values in the home and determine whether there are any high proportions. The system works in many procedures, such as sending SMS alerts to mobile phones, sounding the buzzer inside the home and saving these events to SDRAM. In the event of problems in the GSM network, such problems are solved by using SDRAM to save all events to a specific text file. Moreover, the system uses power-saving devices including a motion sensor. This sensor turns off lights automatically when there is no one in the room. Finally, the system monitors outside weather: for example, in the event of rain, the system closes the roof of the garage automatically.

Keywords: Intelligent Home, Smart Home, Automation Home, Embedded Systems, Home Automation and Security.

ÖZ

GSM ŞEBEKESİ KULLANARAK AKILLI EVİN DİZAYNI VE UYGULAMASI

ALGOIARE**,** Omar Talal
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi: Yrd. Doç. Dr. Yuriy ALYEKSYEYENKOV
Eylül 2014, 56 sayfa

Akıllı ev donanımı olan hava durumu, emniyet ve güvenlik sistemleri, son zamanların en yeni gelişmesi olarak mikrodenetleyici ve kablosuz teknoloji (wifi, gsm, bluetooth, zigbee) ile ileri derecede gelişmiştir. Bu donanımların çoğunun grafiksel kullanıcı arayüzünün kullanım zorlukları vardır.

Son birkaç yılda kablosuz teknolojilerin gelişiminde akıllı ev sistemi için iyi sonuçlar bulunmamaktadır. Bu tezde GSM şebekesi (network) kullanarak akıllı ev sisteminin dizaynını ve uygulaması yapılmıştır.

Bu sistem, genellikle donanım ve yazılımdan oluşan iki bölüm içerir. Donanım, cep telefonları, bilgisayar, akıllı kart okuyucu, ısı derecesi, gaz, hareket ve yağmur sensörleri, GSM modülü, 9 voltluk şarj cihazı, mikrodenetleyici ve SDRAM modülü birimlerinden oluşur.
Donanımın birimlerinden en önemlisi bir mikrodenetleyicidir. Mikrodenetleyici bu sistemin bir beyni gibidir. Donanımın bütün birimleri bu mikrodenetleyici ile

birbirine bağlıdır. Bu sistemi oluşturmak için Arduino Mega 2560 kullanılır. Ayrıca SIM900; GSM/GPRS modemini GSM şebekesine bağlamak için kullanılır. Bu modem, talimatları yollama, alma ve uyarma ile sorumludur. Bunlara ek olarak SDRAM, evin içinde gerçekleşen bütün olayları kaydetmek için kullanılır. Akıllı kart okuyucu da RFID(radyo frekansı ile tanımlama) teknolojisini kullanır. Bu modül, belirli insanlara kapıları açma ve kapama işlemi yapar. Ayrıca bütün bunları SDRAM`de özel metin dosyasına kaydeder.

Yazılım ise bu sistemin ikinci bölümüdür. Bu sistemin grafiksel kullanıcı arayüzü (GUI [graphical user interface]) yazılımdır. Bu yazılım VB6 kullanılarak yazılmıştır. Bu sistem, donanım aygıtları ile USB port aracılığı sayesinde bağlantılıdır. Bu sistemin bilgisayar haricindeki denetim ve gözetimini yazılım çalıştırır. Yazılım, sensörlerin değerini okuma, elektronik cihazların açma ve kapama geçiş kontrolü ve USB porttan geldiği zaman gözetim talimatının gerçek zamanını okuma gibi 3 birimden oluşur.

Bu sistem iki yolla çalışır: 9 voltluk şarj aletinde ve bilgisayarla taşınabilir şekilde. Ayrıca bu sistem, ısının gözlemi ve evdeki yüksek oranda olabilecek gazı gözlemek için çalışır. Eğer bahsedilenlerin herhangi biri normalin üstündeyse, bu kez sistemden telefonumuza uyarı SMS`i gelir, evdeki alarm cihazı çalışır ve bu yüksek oran SDRAM`deki özel metin dosyasına kaydedilir. Böylece, GSM şebekesinde herhangi bir problem olduğunda, biz bu problemi, SDRAM kullanarak ve bütün olayları özel metin dosyasına kaydederek çözdük. Ayrıca güç tasarrufu kullanan bu sistem bir hareket sensörü içerir. Eğer odada kimse yoksa, bu hareket sensörü ışıkları otomatik olarak kapatır. Son olarak, bu sistem dışarıdaki hava durumunu gözler. Eğer yağış varsa sistem otomatik olarak garajın çatısını kapatır.

Anahtar Kelimeler: Akıllı Ev, Otomatik Kontrollü Ev, Ankastre (Gömülü) Sistemler, Ev Güvenliği ve Otomatik Kontrolü.

ACKNOWLEDGEMENTS

First of all, praise to GOD "**ALLAH**" on all the blessings, one of these blessings was the help in achieving this research to the end.

I would like to express my sincere gratitude to **Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV** for his supervision, special guidance, suggestions, and encouragement through the development of this thesis.

Finally, I would like to express my sincere gratitude to my family, especially my mother for her endless help, patience and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

FIGURES

FIGURES

LIST OF TABLE

TABLE

LIST OF ABBREVIATIONS

GSM	Global System for Mobile Communications
GUI	Graphical User Interface
SMS	Short Message Service
RF	Radio Frequency
ECG	Electrocardiography
PC	Personal Computer
RFID	Radio Frequency Identification
XML	Extensible Markup Language
ADK	Accessory Development Kit
PIC	Peripheral Interface Controller
VB	Visual Basic
DTMF	Dual Tone Multi Frequency
I/O	Input /Output
SPI	Serial Peripheral Interface
PWM	Pulse Width Modulation
USB	Universal Serial Bus
PCB	Printed Circuit Board
LCD	Liquid Crystal Display
FTDI	Future Technology Devices International
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
MOSI	Master Out Slave In
MISO	Master In Slave Out
SCK	Serial Clock
SS	Slave Select
UART	Universal Asynchronous Receiver/Transmitter

ICSP	In Circuit Serial Programming
TTL	Time To Live
IDE	Integrated Development Environment
GCC	Gnu C Compiler
API	Application Programming Interface
SIM	Subscriber Identity Module
SMSC	Short Message Service Center
GPRS	General Packet Radio Service
USART	Universal Synchronous Asynchronous Receiver and Transmitter
TCP / IP	Transfer Control Protocol / Internet Protocol
LPG	Liquefied Petroleum Gas
ROM	Read Only Memory
RAM	Random Access Memories
EC	Electronic Cash
PIR	Passive Infrared Sensor
IR	Infrared
GPS	Global Positioning System
DOS	Disk Operating System
M2M	Machine 2 Machine
IVRS	Interactive Voice Response System

CHAPTER 1

INTRODUCTION

1.1 Background

The first prototypes of the first business products were developed over 44 years ago. A number of attempts have been made to develop, implement and maintain a standardized system of control in home automation [1].

Intelligent homes are of interest to many researchers and businesses today. The way of life in modern society, along with human behavior and thought is dramatically changing with the development of technology. The advancement of technology has increased the safety and security of people and their belongings. Some of the reasons for the appearance of intelligent homes include the increased risk of theft and violation of property/the home and a busy lifestyle.

The busy lifestyle of the people is leading to the need to control devices remotely and maintain monitoring over their house. Therefore, it is necessary to define how to understand a smart home. Frances Aldrich states that a "*smart home could be defined as a residence equipped with computing technology and information* [which] *anticipated and respond to the needs of the occupiers, which works to promote comfort, convenience, security and entertainment through the technology management in the home and connections to the world beyond* " [2]. This definition describes the objectives of the intelligent home in some important aspects. Firstly, the intelligent home is packed with different technologies and sensors in order to gather information on the current situation in the intelligent home. Second, increase the quality of life of people. Intelligent home with remote control function became necessary today [3].

Home automation systems integrate electrical devices in the house [4]. There are many different types of intelligent homes on the market, and consumers have the flexibility of choosing the type and number of sensors and price ranges. A house may include intelligent sensors (such as temperature, gas, daylight or motion detectors).

There are many attempts to standardize electronic communication interfaces in intelligent home systems. Some standards use radio frequency (RF) signals, external power supplies or wiring. Control wiring is more difficult to change in an existing house.

In recent years, changes in smartphone technology have been seen. Smart phones come with many cheap and powerful embedded sensors [1]. These intelligent devices are programmable, allowing developers anywhere to develop any application. A point has been reached such that cellphones are being used not only for making calls or for sending SMS's but also for scheduling, time-keeping, notes, control and so on. Today smartphones with different applications and hardware are available in all parts of the world. In addition, cellular networks (GSM) which use new technology. For it is easy to use in our application. With the help of the GSM network, a mobile phone can easily be used for intelligent home applications in order to control electronic devices and receive alerts for theft and burglary.

1.2 Motivation

The research carried out above demonstrates the importance of the implementation of intelligent homes for elderly or handicapped people. Intelligent homes allow such people to stay in places where they feel more relaxed and do not need help from others. Intelligent houses give the elderly or alternatively opportunities for independence that they may have not had before. The aim of this thesis is to explain upon the design and implementation of an intelligent home system that communicates with a mobile phone using a GSM network. The main service of this system collects signals from sensors with data analysis being performed using the Microcontroller and then sending messages such as "Check the temperature at home" or "Check the gas at home."

1.3 Problems

Most existing solutions lack certain significant features. The focus here is on addressing most of the problems associated with existing solutions in home security and smart systems.

Some (examples) of problems in most home automation systems are listed below.

1. Existing systems contain no plans to increase the number of controlled / monitored devices.
2. Requirements of different levels of security for different users. A solution will be provided that can alter the security level.
3. Many solutions do not provide an easy-to-use computer interface to control and monitor home [5].
4. Most existing systems are not affordable for most users due to high costs and difficult maintenance.
5. Some systems provide solutions that are not very useful for household applications [6].
6. Most network problems in existing systems are not resolved.
7. Some current systems provide a view of the house from a web application; however, this can inconvenience the user, who must access the Web each time the user wishes to view the status of the house.

1.4 Literature Survey

The introduction of intelligent homes and home automation (which began in the 1970s) had failed to improve the lifestyles of users for several reasons. First, it is difficult to determine the economic benefits of home automation technologies. The cost of implementation of home automation technology must be justified by the benefits of installation [7]. There is a need for home automation technologies to be cost effective, simple to install and comprising flexible network infrastructures and appliances. Research on intelligent homes constructed the last decade is reviewed in this section.

Ogawa M., Tamura T., Yoda M. and Togawa T. [8], in Japan, built a test system for medical monitoring in the home. The aim of this demonstration system functions to measure automatically and store ECG in the personal computer during bathing.

Sriskanthan N., Tan F. and Karande A. [9], busy families and individuals with physical limitations represent an attractive market for home automation and networking. He explained a model of home automation via Bluetooth through the PC. Unfortunately, the system lacks support for mobile technology.

Al-Ali A. R. and Al-Rousan M. [10], presented a layout and possible implementation of an automation system based on Java via the World Wide Web with an independent board embedded system integrated into a PC-based home server. Home appliances are connected to an input / output board integrated system with status passed to the server. Engine monitoring and control software is based on the combination of Java Server Pages, JavaBeans and Interactive C. Appliances can be monitored and monitored locally through the embedded system board, or remotely through a web browser from anywhere in the world provided that an Internet connection is available.

Deepti S. [11], has developed an application in a universal XML format that can be easily ported to other mobile devices instead of targeting a single platform. Home automation systems is one of the operating systems and the establishment of a single XML document that can be placed on the server that can be adapted by any other mobile device without any problems on the platform. The XML format that controls the layout of the display remains common, except for the part that must be encoded on each platform which is downloaded as an XML file from the server and analyzed. This reduces a large amount of coding effort as the design phase is encoded only once, and the same file can be used by any other platform. All household activities are controlled by a smart phone that is connected to a server that contains the XML file via an Internet connection.

Piyare R. and Tazil M. [12], have introduced the design and implementation of an inexpensive, flexible and wireless home automation solution. The design is based on a separate Arduino BT and appliances are connected to the input / output of this

Forum via relays. Communication occurs between the cell phone and the Arduino BT's wireless. This system is designed to be inexpensive and scalable, thus allowing many devices that to be controlled with minimal changes to its core. Password protection is used to allow only authorized users to gain access to home appliances.

Javale D., Mohsin M., Nandanwar S. and Shingate M. [13], produced a design that is based on an embedded system board Android ADK (Accessory Development Kit) at home. The appliances connected to the ADK and communication between the ADK and Android mobile or tablet is set. The appliances are connected to the input / output board embedded system and their status is passed to the ADK. Low cost devices with less scalable modification of the nucleus are much more important. The design and implementation of the automation system that can monitor and control appliances via Android phone or tablet is presented.

Teymourzadeh R., Ahmed S. A., Kok W. C. and Mok V. H. [14], carried out a study which focuses on the functionality of the GSM protocol and allows the user to control the target system out of homes using frequency bandwidths. Communication concepts and serial AT commands are applied to the development of an intelligent home automation system based on GSM. Property owners can receive status feedback from any household under their control if they connect or disconnect remotely using their mobile phones. A Microcontroller PIC16F887 with GSM integration provides the automated smart home system with a desired baud rate of 9600 bps.

Zhai Y., Cheng X. [3], invented a technique which combined with an embedded GSM system. The design adopted a good platform for the master control system with an Xscale PXA270 processor core as the expansion module and a single chip for information gathering, analysis and processing. A GSM module is connected to transmit all information collected by this system. The design also comprises the acquisition of video data, which can be transmitted over a wireless or wired network to the monitoring center in order to understand status of the home remotely.

Peijiang C. and Xuehua J. [15], contrived a system wherein the GSM network is a medium for transmitting the remote signal. The system includes two parts: the monitoring center and the remote monitoring station. The monitoring center consists of a computer and a TC35 communication module for the GSM. The computer and TC35 are connected via RS232. The remote monitoring station includes a TC35 communication module for the GSM, a MSP430F149 MCU Microcontroller, a display unit, various sensors, data gathering and processing units. The software for the control center and the remote monitoring station is designed using VB. The result of the demonstration shows that the system can monitor and control the distance communication between the monitoring center and the remote monitoring station, as well as the performance of the remote control function.

ElKamchouchi H. and ElShafee A. [16], this system consists of two main components: the GSM modem, which is the communication interface between the home automation system and the user. The GSM Modem uses SMS technology to share data and to signal between users and the home automation system. The second module is the Microcontroller, which is the core of the home automation system, and acts as a bridge between the GSM network (the user) and the sensors and actuators of the home automation system. Sensors and actuators are connected directly to the Microcontroller hardware through the appropriate interface. Security is based on user authentication for each SMS exchange system since each SMS contains the user name and password (with comments). The user can easily configure the settings of the home automation system through the RS232 protocol using a user-friendly interface.

Coskun I. and Ardam H. [17], designed and developed a remote control based on the phone for home and office automation. The circuit has a design based on Turkish standards and is connected to the telephone network as any normal telephone handsets. Any dial tone dual tone multi-frequency (DTMF) telephone set or hand dial tone can be used to send commands to the control unit in order remotely to control a wide range of network devices in homes and offices. The designed circuit can also detect user identification numbers in order to prevent unauthorized use of the control unit. The feedback signal informs the user about the results of the commands.

CHAPTER 2

BACKGROUND THEORY

2.1 Introduction

The intelligent home system consists of two important parts: a SIM900 GSM/GPRS module and an Arduino Mega 2560 Microcontroller. The SIM900 GSM/GPRS is used to send and receive instructions between the system and the mobile station. The Arduino Mega 2560 is a “brain” unit of the system because it is responsible for controlling all parts of the system. All intelligent homes consist of four important parts:

1. Infrastructure: the communication environment for transmission of digital information;
2. Sensors: it’s used to monitor the conditions inside the home;
3. Actuators: these devices run the instruction of the application; for example, they open and close door; and
4. Applications: these enable people to control the smart home system.

2.2 Arduino Platform

Arduino is one of the famous electronic devices in the world due to being open source, flexible software and hardware. The Arduino platform is a simple device built in the AT-mega Microcontroller. Moreover, the software for Arduino works with many operating systems, such as Mac, Windows and Linux. Therefore, the Arduino is better than many types of Microcontroller. The Arduino software language is based on C. There are many types of Arduino modules available on the market [18]. There are a variety of the number of ports and type of Microcontroller along with many purposes for the Arduino.

1. The software and hardware arefree and open source, very flexible and easy to expand.
2. It includes a variety of analog and digital ports, such as I2C, SPI serial interface.
3. It is easy to use and connects to a computer through a USB port.
4. It is cheap and comes with free software and IDE environment.
5. It is bundled with copious amounts of ready-to-use source code.

The Arduino platform is a great tool for developing interactive objects. It receives input from outside, for example, sensors and switches and then controls lights, doors or other outputs. The Arduino platform works as a stand-alone device (?) or connects to a PC through a USB port. Therefore, the Arduino platform canbe used in projects requiring serial communication.

2.3 Arduino Hardware

The Arduino hardware board is a Microcontroller board, which is a small circuit that consists of many electronic components. The Arduino board is a whole computer on a small chip. There are many types of Arduino hardware available on the market. These may vary in size, use, the number of ports and type of Microprocessor. Examples of these types of Arduino hardware include **Arduino Nano, Arduino Mega, Arduino Mini, Arduino Galileo and Arduino UNO.** The Arduino name is a registered trademark, so it is not possible to name a cloned board Arduino. Figure 1 shows some examples of Arduino boards.

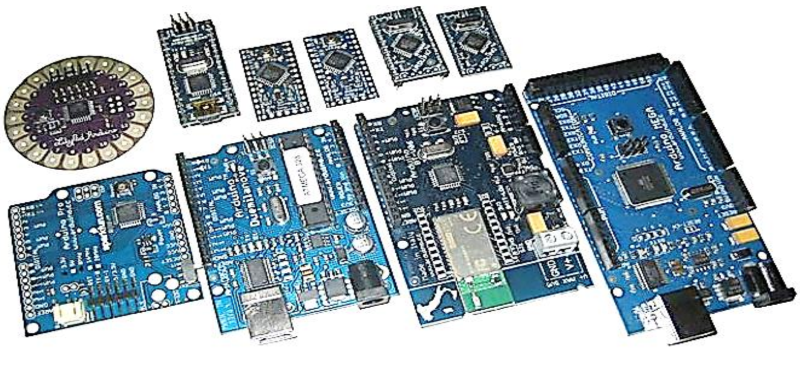


Figure 1: Some examples of Arduino boards

2.3.1 Arduino Mega2560 modules

The Arduino Mega 2560 board is a Microcontroller board included in the AT-mega 1280 Microprocessor. It has 54 digital input/output pins, 14 pins of these 54 pins are used in PWM output. Included also are 16 analog pins and 4 UARTs pins to hardware serial ports.

The crystal oscillator in the Arduino Mega is 16MHz and includes everything to support a Microcontroller, such as a USB connection, reset button, ICSP header, and a power jack. The Arduino Mega 2560 is simply connected to the PC via a USB cable and an AC to DC adapter or battery to start. Figure 2 shows the Arduino Mega2560 board.

Figure 2: The Arduino Mega2560 board

The Arduino Mega 2560 can be powered by a USB port or an external power supply. The power supply is optional. The board can function between 6V and 20V. There are four pins that work with the power. **VIN**: the input power to the Arduino board when we use the external power supply, **5V**: the regulated power supply used to power the Microcontroller, **3.3V**: power supply generated by the on-board FTDI chip and **GND**: the Ground pins [20].

The ATmega1280 has 128KB of flash memory to store code, 4 KB of which are used for the Bootloader, 8 KB for SRAM and 4 KB for EEPROM [20]. Each of the 54 digital pins can be used for input or output. To select the input or output pin, we use the pinMode(), digitalWrite(), anddigitalRead() functions. Each of these pins works at a maximum current of 40mA. These 54pins have some specialized functions. Serial 0: pin 0(RX), pin 1(TX), Serial 1: pin 19(RX), pin 18(TX), Serial 2: pin 17(RX), pin 16(TX), Serial 3: pin 15(RX), pin 14(TX). The TX is used to transmit and the RX is used to receive. The other type of specialized pin is the SPI pin, SPI: MISO pin 50, MOSI pin 51, SCK pin 52, SS pin 53. Each of the 16 analog input pins provides 10 bits of resolution (1024 different values) and uses the analogRead() function [20].

There are many facilities for having the Arduino Mega 2560 communicate with the computer, the other Microcontroller or another Arduino. The Microcontroller ATmega1280 has four hardware UARTs for TTL (5V) serial communication. Also, the ATmega1280 has I2C pins and SPI pin communication [20].

2.3.2 Comparison between Arduino modules

There are many types of Arduino modules. The table shows a brief comparison between Arduino Boards. Table 1 shows the comparison between Arduino modules.

Arduino Type	Mega2560	UNO	Nano	Galileo
Digital I/O Pins	54 pins	14 pins	14 pins	14 pins
Analog Input Pins	16 pins	6 pins	8 pins	6 pins
Processor	ATmega1280	ATmega328	ATmega168	Intel
Memory Size	128 KB	32 KB	32 KB	8MByte
Clock Speed	16 MHz	16 MHz	16 MHz	400 MHz
Cost	Normal	Cheap	So Cheap	Expensive

Table 1: Comparison Between Arduino Modules

From this comparison, the difference is seen in cost and the number of port pins. Therefore, that which has low cost and a high number in port number are used.

2.4 Arduino Software

The other part of the Arduino platform is the Arduino Integrated Development Environment (IDE). This software comes with the Arduino or can be downloaded at no cost from the Arduino site. This software is built in Java. With this software, we can edit, write sketches and upload code to the Arduino [21]. The basis of this software is the C- language. This software is set up in the computer. This compiler is open source.

A first step in the IDE checks whether the C or C++ code is correct. After checking whether the code is correct, the IDE passes the code to a compiler (avr-gcc) to change it from human readable to machine readable instructions. Then, the code is combined with the Arduino libraries code, which provides a basic function. The result of this process is a single hex file. This file contains the specific bytes and it is ready to write to the program memory on the Arduino board, which is transmitted to the Arduino over a USB port or serial connection. Figure 3 shows the home page of the IDE environment.

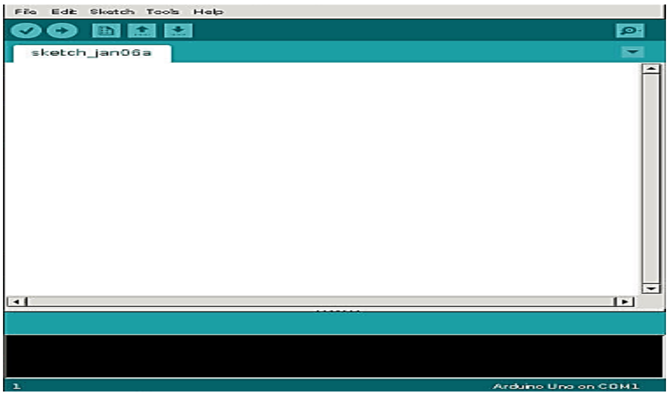


Figure 3: Arduino IDE environment

The IDE software consists of two important functions, **setup** and **loop.** The **setup()** function is called when a sketch starts. It is used to initialize variables, pin modes, start using libraries, and so on. The **setup()** function will only run one time, just after each power up or reset of the Arduino.

The **loop()** function allows the program to respond and change when running.

2.5 Hardware Shields

In the same project, we need some external pins to save data or the type of connection that is not found on this Arduino board. In this case, we must find a way to extend the Arduino by using a shield. The shield has the same number of pins and the same PCB design of the Arduino layout. The shield can be stacked above the Arduino which adds extra functionality. Figure 4 shows examples of the Arduino shield.

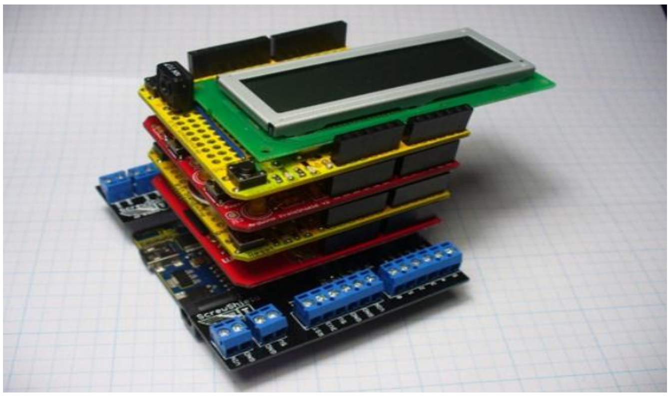


Figure 4: Examples of Arduino shields

The Arduino is placed on the bottom and the other shields are stacked over it. This picture is by John Boxall. There are a large number of shields available on the market, each with a unique purpose. Some are developed by the Arduino Company and others are developed by other persons or companies. For examples the GSM/GPRS shield, the Motor shield, the Ethernet shield, the Analog video output, LCD displays and so on. Therefore, the idea is to use a shield to add a specific feature to the Arduino without developing a new circuit to implement this feature

2.5.1 SIM900 GPRS/GSM shield

This is one example of an Arduino shield. It works to connect with the GSM network. Before discussing this shield, it is important to know what it means, how the GSM network operates and how the SIM900 GSM/GPRS works with this network. GSM is an acronym for the Global System for Mobile Communication. The GSM network differs from the analog mobile network such that subscription and mobile are separated. The subscription data are stored in the Subscriber Identity

Module (SIM). The SIM is a smart card. With this card, we can use any device if it is accessible. The radio devices are called mobile equipment (ME). From these, we can say the Mobile Station consist of two parts: MS = SIM + ME [23, 212]. Figure 5 shows the structure of the GSM network [23].

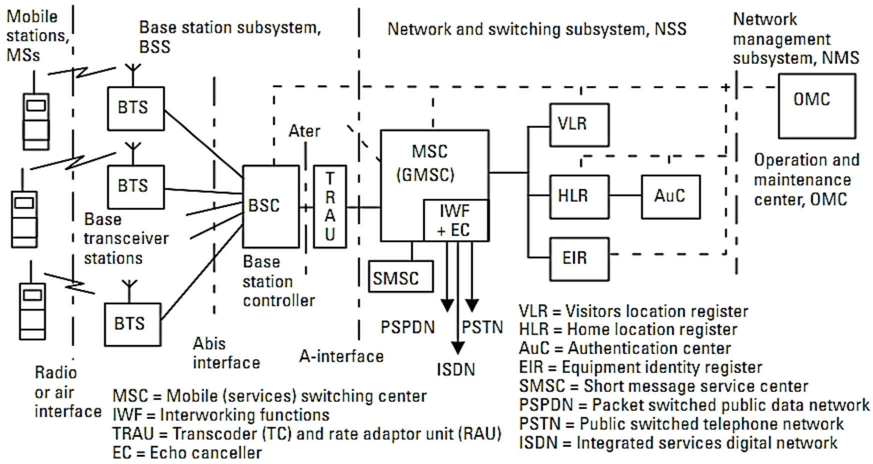


Figure 5: Structure of the GSM network

After knowing what it means, we know what the GSM network consists of and what the Structure of this network consists of. Now, we want to know about GPRS. GPRS is an acronym for General Packet Radio Service. GPRS is a mobile data service for 2G and 3G cellular communication for mobile communications (GSM). The data rate in 2G systems is 56 to 114 kbit per second. In the 3G system, a moderate speed data transfer is provided by using Time Division Multiple Access (TDMA) channels. Figure 6 shows the structure of the GPRS network [23].

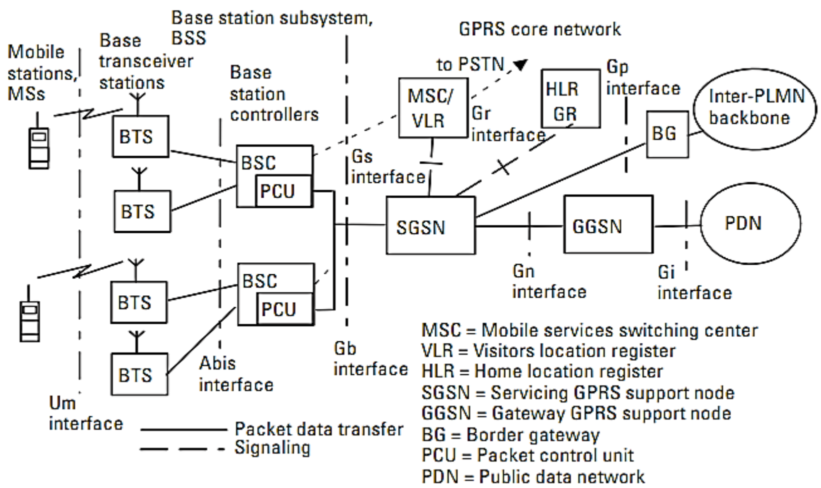


Figure 6: GPRS system architecture

GPRS is a cheaper mobile data service than SMS and data is transmitted more quickly than SMS. The GPRS splits the information when it is transmitted, but related packets are assembled at the other end of GPRS [24].

After understanding how GSM and GPRS operate, it is now easy to understand the SIM900 GSM/GPRS shield. It is small in size and easy to use. This shield is a TTL-Modem that operates in serial and the baud rate can be from 9600 to 115200. The SIM900 operates at many frequencies including 850 MHz, 900 MHz and 1800 MHz. The shield is designed to operate at 3.3V or 5V and works easily with, and connects directly to the Microcontroller. This modem is suitable for SMS as well. The modem can also be connected to the Arduino by using a UART connection [25]. Figure 7 shows the SIM900 GSM/GPRD and Figure 8 shows the GSM/GPRS Shield.

The many features of the SIM900 GSM/GPRS are listed below [25,33].

1- Built-in TTL (serial communication).

2- Built-in SIM Card holder.

3- Audio Interface Connectors.

4- Speaker and Headphone jacks.

5- Normal Operation Temperature: -20°C to +55°C.

6- Input Voltage:5V to 12V DC.

Figure 7: SIM900 GSM/GPRS

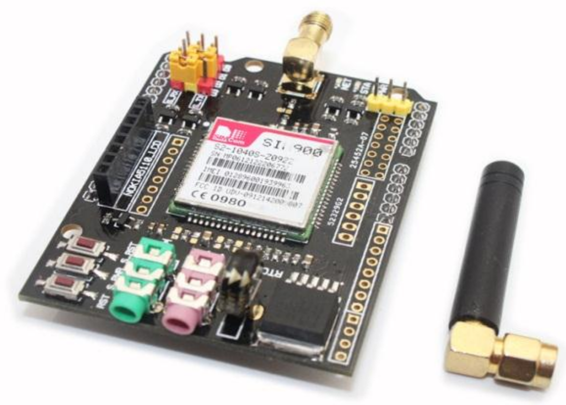


Figure 8: SIM900 GSM/GPRD shield

The SIM900 GSM/GPRS has many uses in many applications [33], including transfer of data between two machines (Machine 2 Machine) in different places and remote control of a device. It also uses a remote in Wireless Sensor Networks to transfer sensor data to a web server and vehicle tracking systems to trace everything with GPS.

2.5.2 RFID shield

RFID is an acronym for Radio-frequency identification. It is used in electromagnetic fields to transfer data and in automatic identification and tracking. Information is saved electronically onto the cards. In the last year, RFID has been very popular with many services, such as manufacturing, industry, tracking systems, animals and products in transit [29, 1-2].

The smart card is an electronic data storage medium made from plastic and includes a microprocessor. The connection between the smart card and the reader is a serial port. One of the important advantages of the smart card is that the data stored inside the smart card isprotected very well. Therefore, it is used in banking. The smart card market is one of the fastest growing markets in the industry [29, 4- 6]. Figure 9 shows an FRID shield.

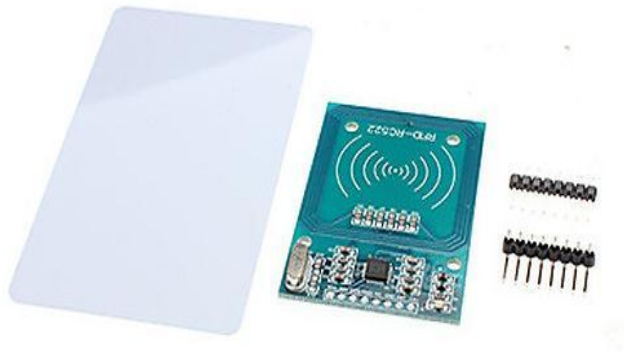


Figure 9: RFID shield (type- RC522)

The RFID consists of two parts: the reader and transponder. The reader depends on the design company to just read or read and write. The transponder is an object which can be identified. Figure 10 shows the reader and the transponder for the RFID system.

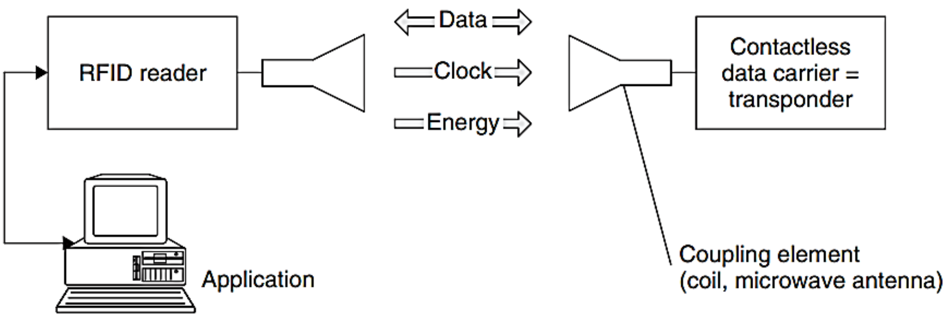


Figure 10: The reader and transponder for the RFID system

2.5.3 SD card shield

The memory size of the Arduino is small – merely at the magnitude of KBs. For large electronic projects requiring large memory to save data such as images, video, audio or so on, major problems occur. Furthermore, if we need to sort any data, a minimum of one megabyte of storage is required. To get acquire such storage capacty, we are going to be using the same type of memory found in cameras or MP3 players. This type of memory is known as SD RAM or an SD card. The advantages of this type of SD RAM is many gigabyte of storage in a small size and its availability on the market. In addition, it is easy to use and easy to save and back up the data from [31]. Figure 11 shows an SD card memory.

Figure 11: SD card memory

To solve the problem and to save a huge amount of data in Arduino projects, many electronics companies build shields to work with SD memory. One companies has produced a new SD shield to operate with SD and MicroSD memory. This product can interface with any type of Arduino. Figure 12 shows an SD card shield.

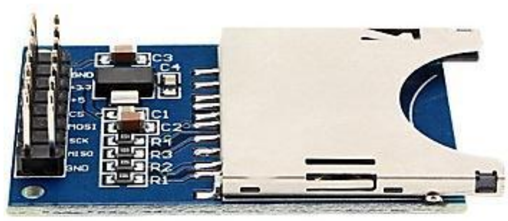


Figure 12: SD card shield

This shield is easy to connect to the Arduino board. It uses an SPI bus interface. This type of connection operates with all types of Microcontroller.

2.6 Sensors

There are many definitions for the Sensors. The sensors are devices that detect events or changes in amounts and give us an output value, usually as a visual or electrical signal. These devices also respond to and receive signals or stimuli [26,19-20]. The sensor's sensitivity refers to how much the output of the sensor changes when the amounts of the input being measured changes, or the ratio between the output signal and the amount of property. The sensors are used every day in our life, such as the LM35 sensor, which is used to convert temperature to output voltage. There are two types of sensors: digital and analog. The digital sensitivity is constant with the unit.

Therefore, this type of sensor is called a linear sensor since the ratio is constant at all points of the measurement. An analog sensor is needed to convert to the digital format by using the ADC.

In general, there are two kinds of sensors: direct and complex sensors. Direct sensors are easy to use and convert a stimulus to an electrical signal (pulse). The complex sensor cannot be used directly. It requires more power before converting to an electrical signal [26, 22-23]. Figure 13 shows the sensor transducers.

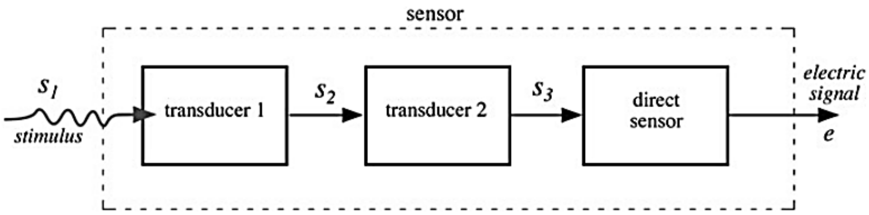


Figure 13: Sensor transducers

In Figure 13, we see that S1, S2, S3 implement different kinds of energy. The last part of this figure implements the direct sensor. The most important properties of the sensors are the response time, dynamic range, sensitivity, cost and size.

2.6.1 Temperature sensor

The temperature sensor is an integrated circuit to read temperature value. The output voltage of this type of sensor is linearly proportional. Some properties of this sensor include greater ease of use, better accuracy than a thermistor, and higher output voltage than thermocouples. There are many kinds of temperature sensor: LM35, LM35D, LM35CA, and LM35C. They are also available in different transistor packages: TO-92, T0-46, TO-202 [27]. Figure 14 shows the basic temperature sensor centigrade and Figure 15 shows a full-range temperature sensor. These photos from [27].

The full-range temperature sensors can read between -55 ºC to +150 ºC. The circuit connection to the full-range sensor is shown in Figure 15. The +V is the power supply to LM35 sensor. R1 is a resistance that is connected between the output voltage and –Vs. Finally, we can obtain a temperature from the output voltage of the sensor Vout in degrees centigrade.

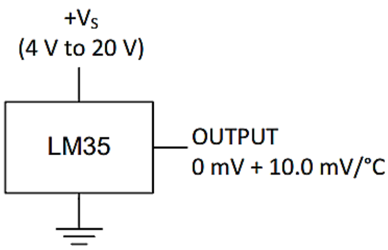


Figure 14: Temperature sensor centigrade

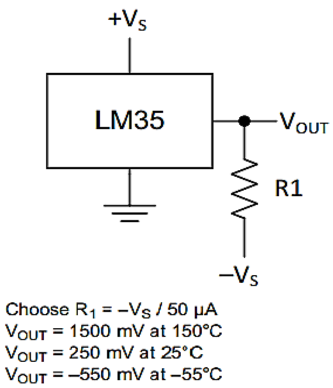


Figure 15: Full range temperature sensor

2.6.2 Gas sensor

A gas sensor is a device to detect various gases in an area. Most usages occur in security systems since it detects many types of gases, including hydrogen, LPG, smoke, methane and so on. Generally, the gas sensor consists of two parts: a steel exoskeleton and ionized material from the gases. When the sensor is exposed to a gas, the ionized material becomes heated and closes the sensing element. This change in resistance of the sensing element changes the current exiting the sensor.

There are many types of gas sensor available in the market: MQ2, MQ306, MQ135, and MQ137. Every one of these works with a specific type of gas. The MQ2 semiconductor sensor is used for general combustible gas.

The MQ2 sensor detects smoke and combustible gas at concentrations between 300 and 10,000 ppm. The output of this sensor is analog. The MQ2 sensor has a high sensitivity to hydrogen, LPG, propane [28]. The advantages of this sensor are low cost, small size and high sensitivity. Therefore, it is used in various devices such as portable gas detectors, combustible gas detectors in industry, and in domestic gas leakage detectors.

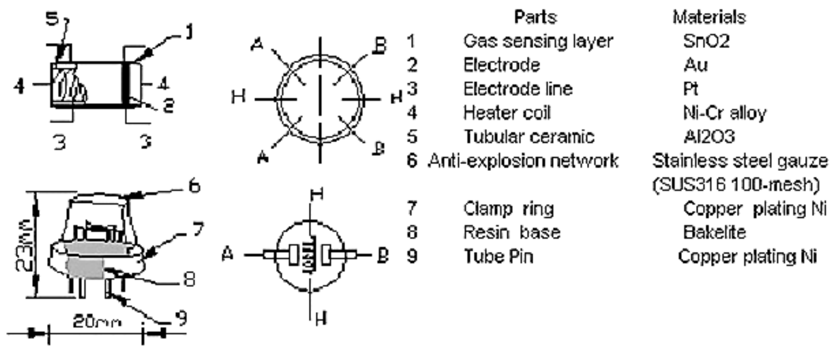


Figure 16: Structure and configuration of the MQ2 sensor

Figure 16 shows the structure and configuration of the MQ2 sensor [28]. The sensor consists of a micro AL2O3 ceramic tube and a measuring electrode. A heater is fixed into a crust made of plastic and stainless steel. The heater, which contains sensitive components, is an compnent in the sensor.

2.6.3 Rain sensor

The Rain sensor is a simple tool to detect rain. This sensor can be used in measuring rainfall intensity and it works a switch when raindrops fall onto the sensor board. This sensor consists of two parts: the controller and the rain board. This sensor is one of the members of the analog sensor family. Some properties of this type of sensor include the following: it operates on 5 volts; it includes two types of output: digital and analog; it is easy to instal and use ; it has anti-oxidation properties; it is of high quality and is constructed with double sided material. Figure 17 shows a rain sensor module.

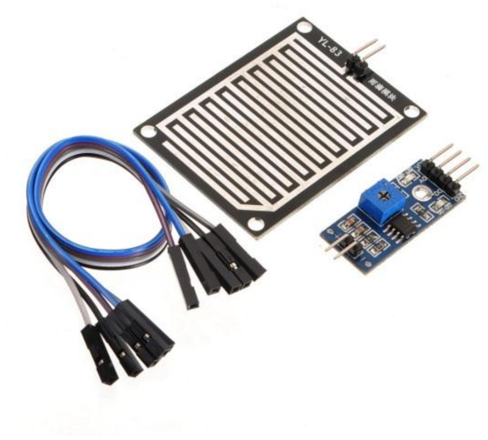


Figure 17: Rain sensor module

This sensor consists of four pins, the important of which is Aout (analog output). This pin connects to the analog port of the Microcontroller and converts it to digital format [32].

2.6.4 Motion sensor

A motion sensor is an electronic device that is used to detect moving objects. It is often integrated into many systems in order to provide an automatic alert to the user if there is any motion in a specific area. The motion detector is a vital element for automation, security, energy efficiency, lighting control and other systems. There are many types of motion sensors: passive infrared (PIR), the tomographic motion detector, video camera software, ultrasonic,tomographic motion detector, and microwave.

The Passive infrared sensor has a potentiometer to calibrate distance and delay time [35]. The PIR sensor is located in a closed box made from metal to improve temperature, noise and humidity. This box includes a window made from silicon and IR-transmissive material. Behind this window are two balanced sensors. Figure 18 shows the PIR element window.

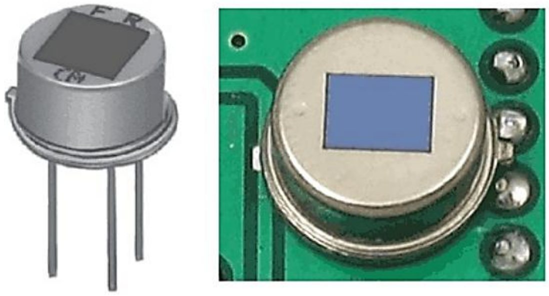


Figure 18: PIR element window

Furthermore, it includes a type of transistor known as JFET. This type of transistor is low- noise, high resistance and low cost [30]. In this case, the sensitivity and detection area is just two rectangles, as can be seen clearly in Figure 19. These two rectangular areas are not sufficient for our needs. We want to increase this area by using a simple lens such as a camera lens or Fresnel lenses. Figure 20 shows a Fresnel lenses. Now the range has increased. Figure 21 show a Fresnel lens used in conjunction with the PIR sensor.

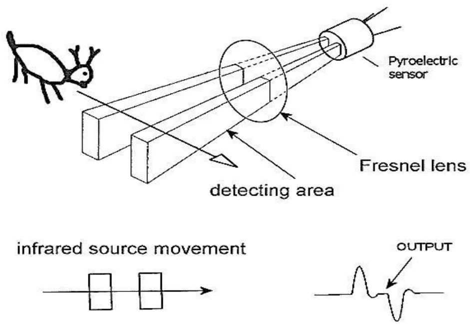


Figure 19: PIR rectangles

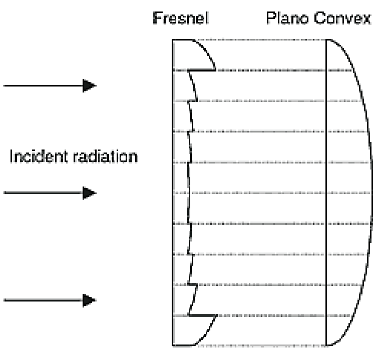


Figure 20: Fresnel lens

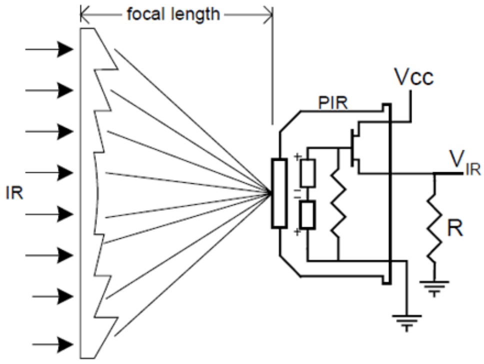


Figure 21: Fresnel lenses with PIR sensor

2.7 Buzzer Alarm

A buzzer is a sound signalling device. The buzzer alarm is used in many devices, such as timers, clocks, security systems and so on. The DFRobot is one type of buzzer. This type is very popular because of its small size and low cost and and because it connects directly to the Microcontroller [35]. Figure 22 shows the buzzer alarm.

Figure 22: Buzzer alarm

CHAPTER 3

STRUCTURE OF THE SYSTEM

3.1 System Design

The system consists of two parts: First, the mobile phone, which is responsible for giving the control instruction to home devices and receiving responses from the sensors. It is the same a GUI and does not connect to the devices directly. The second part is the Microcontroller unit, which is responsible for controlling the devices and sensors. It is connected all these parts and processes information from the sensors and mobile station. The Microcontroller is the brain of the system which controls and processes all information. Figure 23 shows a block diagram of the system.

From this block, we see two important parts: the Microcontroller Board and the Sim900 GSM/GPRS Module. The Microcontroller part connects to four sensors: the Smart card reader (RFID), the SD memory and Lights Home, whereas on one side, it connects to the GSM modem.

The LM35 sensor is a temperature sensor and a Heat Detector in the system; the PIR sensor is a motion detector that uses a passive infrared sensor; the YL-83 module is a Rain sensor; and the MQ2 sensor is a gas / smoke detector. The data from all these sensors are continually processed by the Microcontroller and an alert is given if there is any event in the home by sending an SMS to our mobile phone directly. In addition, an audio alert is given and this event is saved on the SD card. The MFRC522 module is a smart card reader onto which all events in the home are saved.

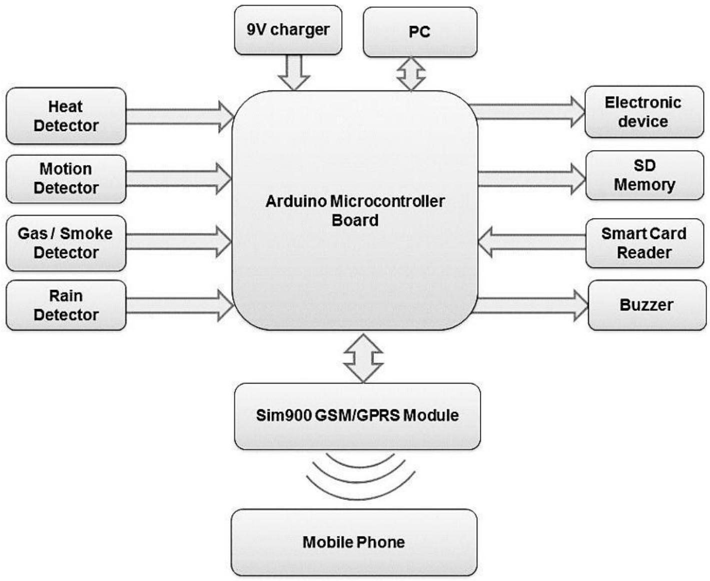


Figure 23: Block diagram of the system

These units are responsible for the security of the home. The system uses an LED light to show the demo of the remote device management by using the mobile station. The user can control the state of the LED and remotely turn it on or off.

In fact, the Sim900 GSM/GPRS module is the interface between the Microcontroller and mobile station. The Sim900 GSM/GPRS works to make a connection between the Microcontroller and the mobile station. During operations of this unit, the information transfer from the Microcontroller to the mobile station is sent from the mobile station to the Microcontroller. The commands sent by the user from a mobile phone are executed by the Microcontroller. The commands are sent to the Microcontroller via text messaging. This system works in an intelligent home system providing security and a remote management system for the devices inside the home.

3.2 Interfacing the Sim900 GPRS/GSM Module

The Sim900 GSM/GPRS shield is an important unit in the system. It works to make connections between the Microcontroller and the mobile station. Now, we must know how to program theSim900 GSM/GPRS and connect it to the Arduino Mega 2560. Moreover, we must know which library is used to send and receive a text messages. In this work, we used what was the latest version of the GSM/GPRS library [36]. The library is open source and uses Google hosting at:

https://code.google.com/p/gsm-shield-arduino/. It was created by Marco Martines. This library is constantly updated and improved with the addition a new features. The main important feature is TPC/IP communication support through GPRS.

At the first step in order to work with the GSM/GPRS library, we must put this library into a folder of libraries in the Arduino environment directories. The GSM/GPRS library folder includes all necessary functions. For basic operation, some functions need to be included. There are many types of important functions in the GSM/GPRS library, including ***SoftwareSerial.h***, ***SIM900.h*** and ***sms.h***.

- **SoftwareSerial.h**: Development of this library allows for serial communication on the digital pins on the Arduino. It is used to replicate functionality, hence the name "SoftwareSerial") [41].

- **SIM900.h**: This function is important and is put first due to its inclusion of basic functions of starting and configuring the GSM/GPRS modem. This library includes an important function: **GSM.begins()**. It turns the power on and sets the baud rate of the GSM module for serial communication.

- **Sms.h library**: This functions with a text message when it is necessary to send or receive an SMS. In this library, there are two important functions: First, SendSMS (), with two parameters: the mobile phone number and the text message. It uses this function to send an SMS to any mobile phone. For example, SendSMS ("039258695","Welcome!"). The second function, GetSMS(), is used when we need to receive an SMS. It consists of four parameters: the SMS position (the position of SMS in the SIM card from 1 to 20), the phone number (this is the number that sends this SMS), the SMS text itself (i.e., the text message), and an integer number (the number of characters in the SMS). For example, this function **sms.GetSMS(sms_position, phone_number, sms_text, 100)** is written in the Arduino IDE in order to send and receive SMS code as shown in Figure 24.

GSM_GPRSLibrary_SMS §

```
#include "SIM900.h"
#include <SoftwareSerial.h>
#include "sms.h"
SMSGSM sms;
int numdata;
boolean started=false;
char smsbuffer[160];
char n[20];

//debug begin
char sms_position;
char phone_number[20]; // array for the phone number string
char sms_text[100];
int i;
//debug end

void setup()
{
    //Serial connection.
    Serial.begin(9600);
    if(started)
    {
        if (sms.SendSMS("0539558894", "Wellcome!"))
        Serial.println("SMS sent Successfully.");
    }
};

void loop()
{
            // read new SMS
            Serial.print("SMS postion:");
            Serial.println(sms_position,DEC);
            sms.GetSMS(sms_position, phone_number, sms_text, 100);
            // now we have phone number string in phone_num
            Serial.println(phone_number);
            // and SMS text in sms_text
            Serial.println(sms_text);
            delay(1000);

};
```

Figure 24: Arduino IDE code to send and receive SMS

This is a simple code to send and receive SMS by using GSM/GPRS module.

3.3 Interfacing and Implementing Sensors

There are many different regions that need to be monitored repeatedly and devices which have to be checked in the home. For example, doors and windows need to be monitored for thieves attempting breaking and entering. Monitoring the movement of strangers around the homeis also needed. In addition, in order to know the status of the electronic devices after leaving home, to turn certain devices on and off and to adjust the temperature at home, it is necessary to monitor these and for us to receive an alert upon reaching the critical point. Control of the temperature, movement, rain and opening and closing of the doors and windows are carried out by specific sensors. The software is responsible for managing sensor operations. There are many different types of sensors which interface according to the output and properties of the sensors. Some of the sensors need an external circuit in order to interface with an application and some others do not.

3.3.1 Heat detector

Temperature is one of many important areas to be controlled in the house; therefore, it must be known at any time. Moreover, a temperature sensor can be put to use in the house in many ways. For example, it can be used as a fire alert. It can also be used in cold places where the temperature needs to be controlled so that the temperature does not increase beyond a set limit. A temperature sensor is used to check the heating system at certain points and achieve more energy savings. Finally, it helps to save money.

The Lm35 temperature sensor is used as the heat detector in this system. This sensor is better when it is used in the home due to its low cost and highest sensitivity between +2 C^{o} and +250 C^{o} and a low power consumption at only +5v DC. The Lm35 is easy to connect to the Arduino as depicted in Figure 25.

Three pins on theLM35 sensor are shown in figure 25. The analog pin for the LM35 connects to an input port on the Arduino Board, such as A15. The two others pins on the LM35 are used for power and connect to a 5-volt power source and ground output port on the Arduino board. Therefore, this type of sensor does not need an external circuit as a power supply and the output of the sensor can be fed directly into the input port of the Arduino Mega2560.

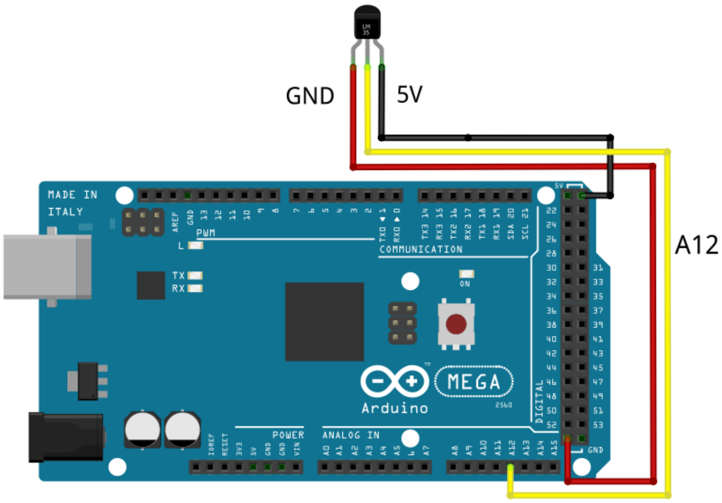


Figure 25: Block diagram of LM35 with Arduino

The code is used to interface between the LM35 and Arduino board and calculate the temperature value as shown in the next figure. Figure 26 shows the IDE code to read heat sensor values.

```
sketch_aug13a §

float temp;

void loop()
{
  temp = analogRead(A15); //reads the output of LM35 sensor
  temp = temp * 0.48828125;
  Serial.print("TEMPRATURE = ");
  Serial.print(temp);
  Serial.print("*C");
  Serial.println();
  delay(1000);
}
```

Figure 26: Arduino IDE code to read LM35 value

The Arduino reads the output value from the LM35 every one second by using the analog function **analogRead()**. The temperature is calculated from the output voltage by using the equation **temp= temp * 0.48828125.** The float value in this equation is derived from **(SUPPLY_VOLTAGE * 1000 / 1024) / 10** where **SUPPLY_VOLTAGE** is 5.0V (the voltage used to power the LM35) and 1024 is 2^{10}, the value where the analog value can be represented by the ATmega. 1000 is used to change the unit from V to mV. 10 is constant. Each 10 mV is directly proportional to 1°Celsius. Simple math equation: **(5 * 1000 / 1024) / 10 =**

0.48828125. The result of this equation is the real temperature value in digital format after converting from analog. The last line that is printed is the value of the result in the serial port and the delay.

3.3.2 Motion detector

Motion detectors, or movement sensors, are used to detect any motion of people inside or outside the home. Hence, it uses a Passive Infrared sensor (PIR) to find any movement and send an alert to the user regarding this movement. The Panasonic company manufactured this type of the PIR sensor. This sensor consists of three pins as shown in Figure 27.

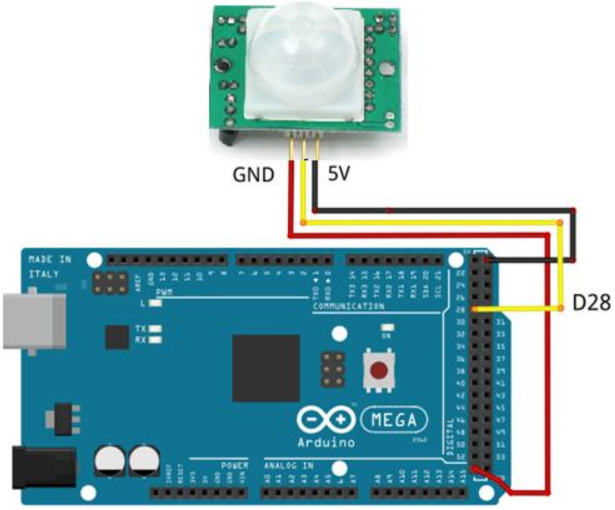


Figure 27: Block diagram of PIR sensor with Arduino

The 5V and GND connects the Arduino board to the PIR sensor and output sensor connects to the digital input port in the Microcontroller as in D22. The output of this sensor is a digital pulse, 30 seconds maximum – the stability time for this value to turn on and off [37]. The interfacing code for the PIR sensor with an Arduino Mega2560 board is shown in Figure 28:

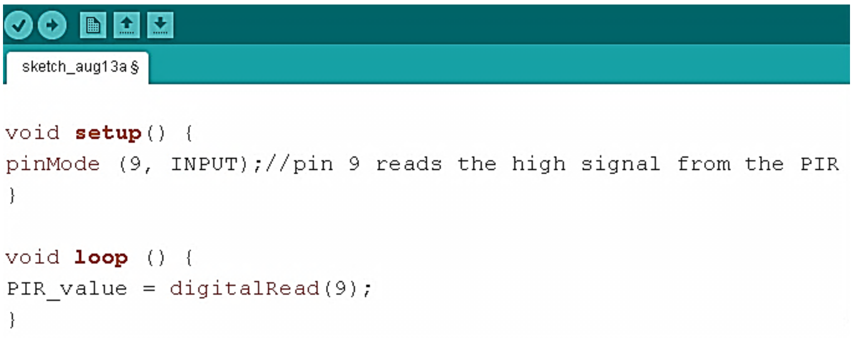


Figure 28: Arduino IDE code to read PIR value

There are just two output possible statuses in the PIR sensor, high or low. To read the status from the PIR sensor, we set the digital pin port to input mode and then use the **digitalRead()** function. Of note, this sensor does not need an external power supply.

3.3.3 Gas detector

The MQ2 (gas sensor) is one of many types of gas detector. It uses a small heater inside with an electrochemical sensor. It has fast response time. There are many features, including a wide range of detection, fast response time, and a stable and long life [38]. Furthermore, it is more sensitive. Therefore, it can be used inside a room to detect smoke and gas inside the home.

The output of this sensor is an analog signal and we can read it in the Arduino analog port. Figure 29 shows how to connect the MQ2 sensor to the Arduino Mega. The gas sensor has three pins, two pins used for the power supply and the remaining pin used for the output signal. The output pin (AOUT) connects to the analog input port of the Arduino Mega board (A14).

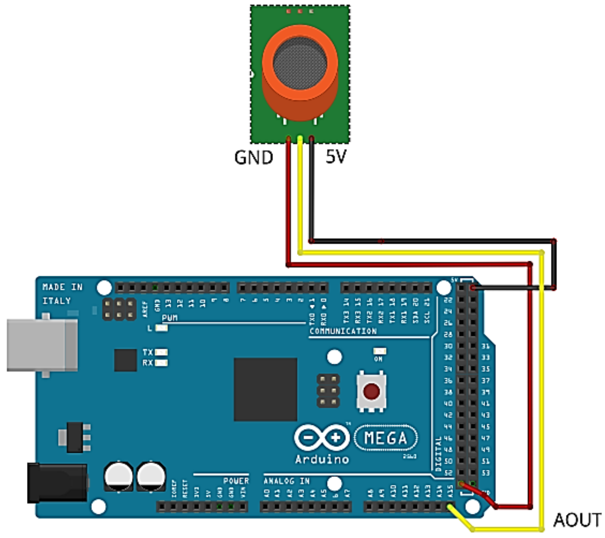


Figure 29: Block diagram of gas sensor with the Arduino

In order to read the gas sensor value in the Arduino Mega, we use the **analogRead()** function as shown in Figure 30. The analog-to-digital converter equation is identical to the analog-to-digital converter equation of the heat detector.

```
void loop() {

  // Read value on A14
  Gas_sensor_Value = analogRead(A14);

  Gas_sensor_Value = Gas_sensor_Value * 0.48828125; // convert to digital format

  // Display value
  Serial.println(Gas_sensor_Value);

  // Loop 10 times per second
  delay(100);
}
```

Figure 30: Arduino IDE code to read gas sensor value

3.3.4 Rain detector

The rain sensor is a simple tool to detectrain. This sensor can be used to measure rainfall intensity and works when rain falls onto the rain sensor board. This sensor consists of four pins: 5V, GND, D0 and A0. This sensor is easy to use and it is easy to interface with any type of Microcontroller. Figure 31 shows the connection between the rain sensor and the Arduino Mega 2560. Figure 32 shows the code in IDE to read the rain sensor value.

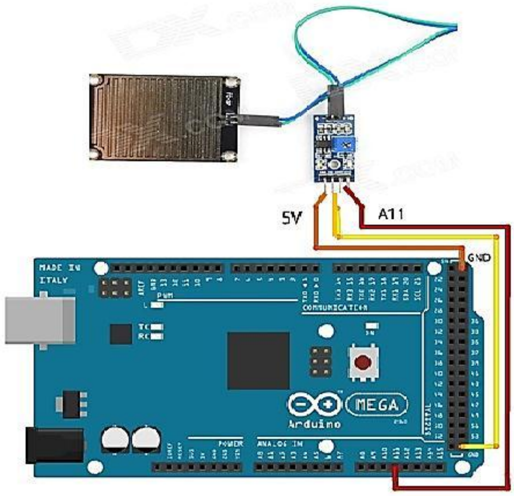


Figure 31: Block diagram of the rain sensor with the Arduino

```
void setup ()
{
Serial.begin(9600);
}
void loop ()
{
int val = analogRead(A11); // read sensor value and send to variable val
val = val * 0.48828125; // convert to digital format
Serial.print("sensor = " );
Serial.print(val); // serial print variable data
delay(100);
}
```

Figure 32: Arduino IDE code to read rain sensor values

3.4 Manage Electronic Devices

Most electronic devices and lights in the home need to change status on or off continuously. At times, we forget to turn off our electronic devices or lights when we leave home. This is a problem; therefore, in this work we solve this problem to creating a remotely managed devices system inside the home. With this system, it is unnecessary to return home to turn off ones devices or lights.

There are many ways to implement remotely managed electronic devices; for example, management by using ethernet or using a GSM/GPRS module. The lighting system is managed by using the GSM/GPRS module. In this type of project, SMS services are used. Figure 33 shows how to connect the LED with an Arduino board.

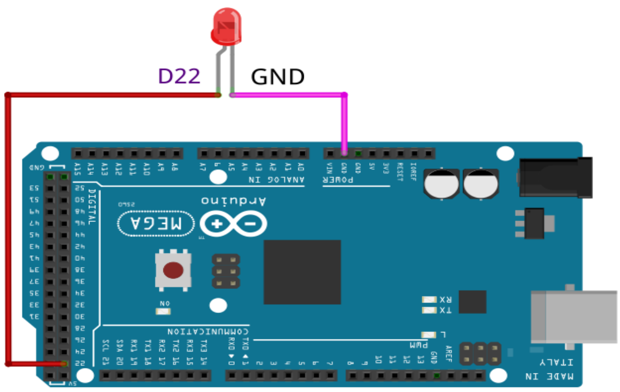


Figure 33: Block diagram of the LED with the Arduino

The LED consist of two legs: the long leg being (+) and another leg (-). To connect to the Arduino, we set the digital port to output mode after connecting the positive wire (?) (+) of the LED with the digital pin of the Arduino. To turn the LED on or off, we use the **digitalWrite()** function. If the value is **HIGH,** it means ON; if **LOW,** it means OFF. The Arduino code for interfacing with the LED as shown in the Figure 34.


```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);  // LED on
  delay(1000);
  digitalWrite(13, LOW);   // LED off
  delay(1000);
}
```

Figure 34: Arduino IDE code to LED control

3.5 FRID Reader

The Radio Frequency Identification (RFID) is a very cheap, small and can be found at any electronics vendor. It operates at 13.56 MHz and can be used in many applications such as automatic identification, robotics, payment systems, and so on. It operates at a low voltage of 3.3 volts, therefore it does not require external power.

To the FRID easily interfaces with an Arduino Mega board through a Serial Peripheral Interface (SPI) bus. Figure 35 shows a block diagram of how to connect the RFID module to the Arduino Mega.

In the RFID module, there are three important pins: SCK, MOSI and MISO [39]. To transmit data in the SPI bus, clock initiated by the master to check whether the slave is ready to receive data.

The master sends a bit on the MOSI line and the slave device receives it in the same line. The slave sends one bit to the MISO line and the master receives it in the same line.

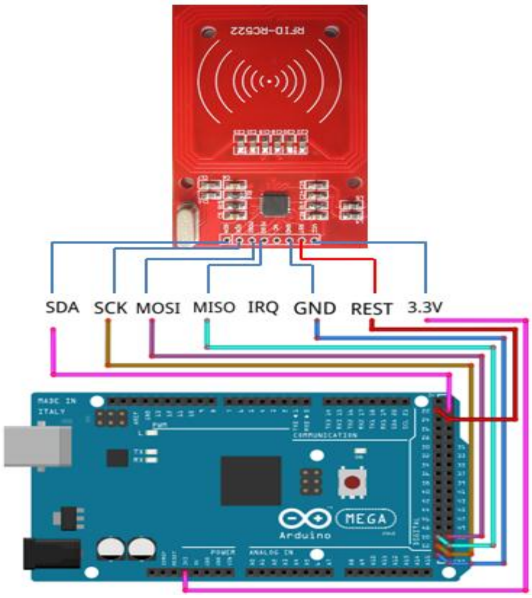


Figure 35: Block diagram of the RFID with the Arduino

SPI code in the Arduino IDE is shown in Figure 36.


```
#include <SPI.h>
#include <RFID.h>
#define SS_PIN 23   //SDA
#define RST_PIN 22   //resset
RFID rfid(SS_PIN, RST_PIN);
// Setup variables:
    int serNum;
    char stat;
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  rfid.init();
}

void loop()
{  if (rfid.isCard()) {
        if (rfid.readCardSerial()) {

                /* With a new cardnumber, show it. */
                serNum = rfid.serNum[0];
                Serial.println("\nCardnumber:");
                Serial.print(rfid.serNum[0]);
                delay(2000);
              }
      }
    rfid.halt();
}
```

Figure 36: Arduino IDE code to read the RFID number

First of all, it must call SPI and RFID libraries in the IDE. Now, it is able to operate with RFID functions. The **rfid.isCard()** function is used to check whether there is any card nearby, and reads the serial number of this card by using the **rfid.readCardSerial()**function. In addition, the **rfid.halt()** function is used if there is anything wrong with the RFID. Finally, the card number is printed.

3.6 SD Card Module

The SD Card Shield is a simple solution to save huge amounts of data. The pin out of this shield connects directly to the Arduino and any type of Microcontroller. The SD card operates at 3.3V and it has a power regulator when we use it at 5V. This module can also be used as a Micro SD socket.

Figure 37 shows the block diagram to interface between the SD Card modules and the Arduino Mega2560. This shield works with an SPI bus. SPI is a protocol that is used to synchronize serial data between the Microcontroller and any other peripheral device in a short time and over small distances [40]. Moreover, it can be used to connect between two Microcontrollers. With this type of connection, there is one master device, the Microcontroller, that monitors all peripheral devices.

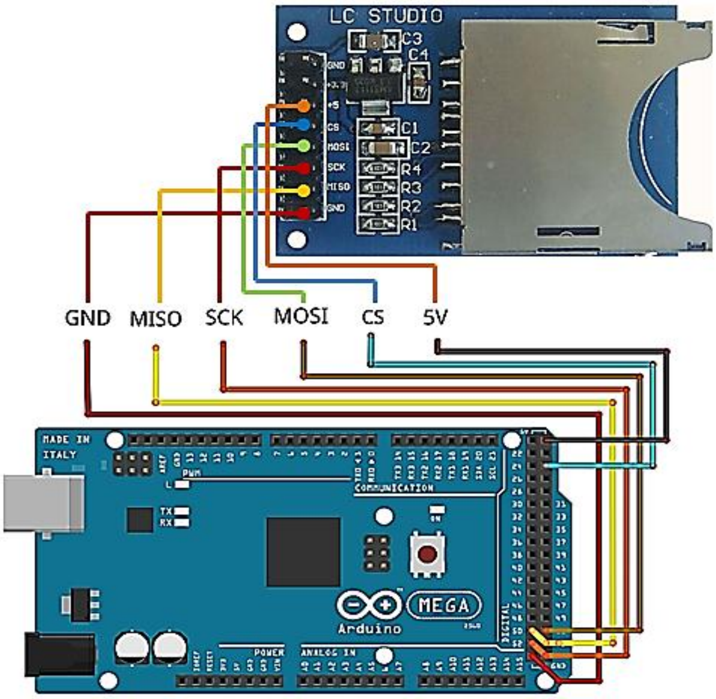


Figure 37: Block diagram of SD card with Arduino

In this connection there are three line:

1- MISO (Master In Slave Out): The Slave line for sending data to the master.
2- MOSI (Master Out Slave In): The Master line for sending data to the peripherals.
3- SCK (Serial Clock): The clock pulses which synchronize data transmission generated by the master and one line specific for every device: SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices.

In Figure 38, the simple code uses the SD library and SPI bus to interface the Arduino Mega with the SD module to work with SD card, we must call the SD library at the beginning. This library allows us to read and write from the SD card. At the first step, we call the **SD.begin()** function to start a connection with the module. After initialization of the card, we open the file by using the **SD.open()** function. Once available, the data can be written into the file by using the **myFile.println()** function. Finally, all data is saved in the SD file by using the **myFile.read()** function.

ReadWrite §

```
#include <SD.h>

File myFile;

void setup()
{
 // Open serial communications and wait for port to open:
  Serial.begin(9600);
   while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
  Serial.print("Initializing SD card...");
   pinMode(53, OUTPUT);
  if (!SD.begin(23)) {
    Serial.println("initialization failed!");
    return;
  }

      Serial.println("initialization done.");
      myFile = SD.open("test.txt", FILE_WRITE);

      // if the file opened okay, write to it:
      if (myFile) {
        Serial.print("Writing to test.txt...");
        myFile.println("testing 1, 2, 3.");
            // close the file:
        myFile.close();
        Serial.println("done.");
      } else {
        // if the file didn't open, print an error:
        Serial.println("error opening test.txt");
      }

   myFile = SD.open("test.txt");
   if (myFile) {
     Serial.println("test.txt:");

     // read from the file until there's nothing else in it:
     while (myFile.available()) {
         Serial.write(myFile.read());
     }
     // close the file:
     myFile.close();
   } else {
         // if the file didn't open, print an error:
     Serial.println("error opening test.txt");
   }
 }
```

Figure 38: IDE code to read and write from the SD module

3.7 Buzzer Alarm

The buzzer alarm is connected to the digital output port. When we make the output port high, the buzzer emits a tone. Furthermore, we can connect it to an analog pulse-width modulation to produce a different tone. In Figure 39, we can see the block diagram of how to connect the buzzer module to the Arduino and Fgure 40 shows the sample code to generate sound.

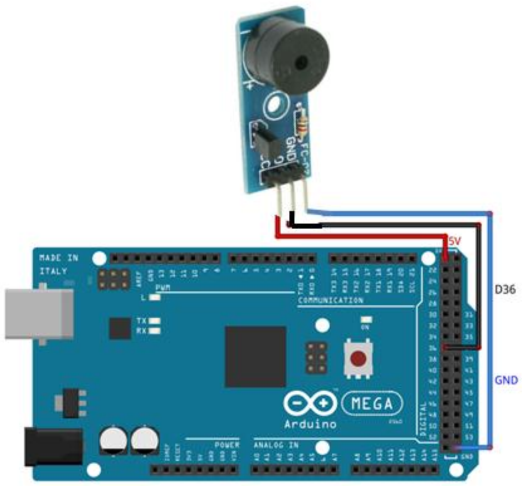


Figure 39: Block diagram of buzzer alarm with Arduino

sketch_aug18a §

```
void setup()  {
  pinMode(9, OUTPUT);  // declare pin 9 to be an output:
  beep(50);
  beep(50);
  beep(50);
  delay(1000);
}
void loop()  {
  beep(200);
}
void beep(unsigned char delayms){
  analogWrite(9, 20);       // Almost any value can be used except 0 and 255
                            // experiment to get the best tone
  delay(delayms);           // wait for a delayms ms
  analogWrite(9, LOW);        // 0 turns it off
  delay(delayms);           // wait for a delayms ms
}
```

Figure 40: IDE code to the buzzer alarm

CHAPTER 4

IMPLEMENTATION RESULTS AND DISCUSSION

4.1 Introduction

This chapter explains the implementation of the intelligent home system in details, how it is works and discusses all of the results. In this system, we used two LEDs to show the operations of remote electronic devices in the home, and the four sensors, namely the movement detector (PIR) to detect motion, the LM35 heat sensor to detect heat, the Gas/Smoke sensor and the Rain sensor to close the roof of the garage when it rains. The number of sensors can be increased or decreased depending on the requirement of the application. Moreover, it includes a smart card to open and close doors to specific persons. The system saves all events on an SD card.

The system can connect with a personal computer to carry out real time monitoring and controlling in order to show the current climate inside the home. The system can run without a PC. It works with a 9-volt external charger. The system includes a buzzer to sounds an alert if there are any increases in heat or gas. The system gives an alert if there are any increases in gas or heat inside the home to our mobile phone directly and saves it in a text file in SDRAM, which can then be read at the other time. The connection environment in this project is the GSM network, and to send instructions, we use the SMS service.

This system has been implemented on the Arduino Mega 2560 platform. The system uses a SIM900 GSM/GPRS module to interface with the GSM network to send and receive SMSs. Furthermore, it uses a USB port to connect to a PC for real time monitoring and control. Figure 41 shows the final image to the system.
The system consist of the Arduino Mega2560, a GSM shield, lights, RFID, an SD card, a heat sensor, a gas sensor, a rain sensor, a PIR sensor, a buzzer, a USB cable, a

9-volt charger. All of these connect to the Microcontroller. The whole system is implemented in the IDE programming language and uploaded to the Microcontroller using the same environment.

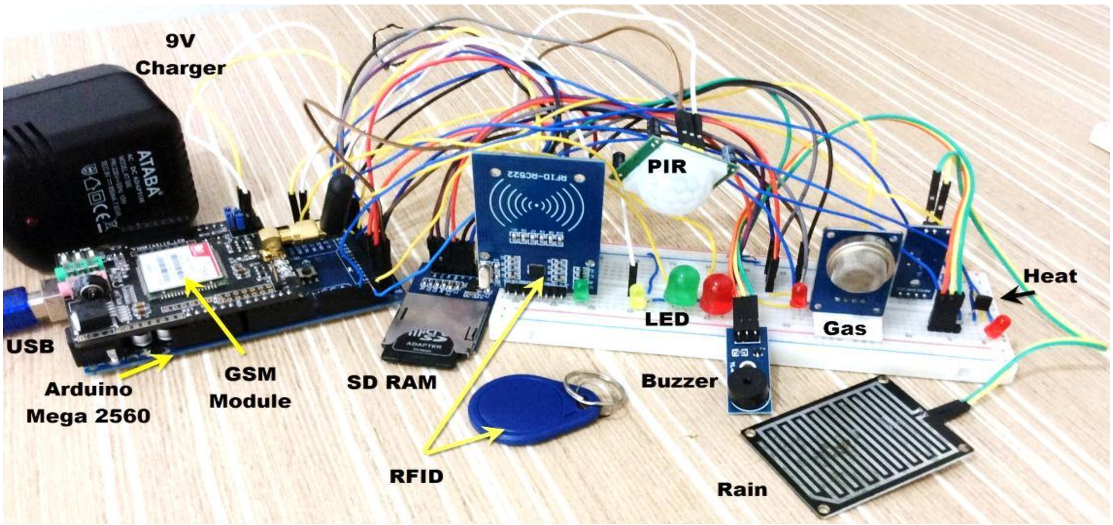


Figure 41: Final intelligent home system

4.2 Testing the Hardware System

At first, all hardware units were tested and checked to ensure that everything was in a good working condition. After checking, all hardware units were tested individually with the Arduino Mega 2560 to determine the necessary software in this application and to detect any errors in each unit more easily. The following unit was not tested until the current unit gave us the correct result and worked well according to the necessary functionality in the application. After testing and checking all hardware units, it was found that all were working correctly. The next step was to connect all these units together and test the whole system.

The first step was to test the first important unit in the system: the connection of the GSM module to the GSM network and whether it worked correctly with the Arduino Mega2560 platform. From this step, we were able to know whether the system was ready to use. Figure 42 shows the workings of the GSM software. Figure 43 shows the result of the GSM testing.

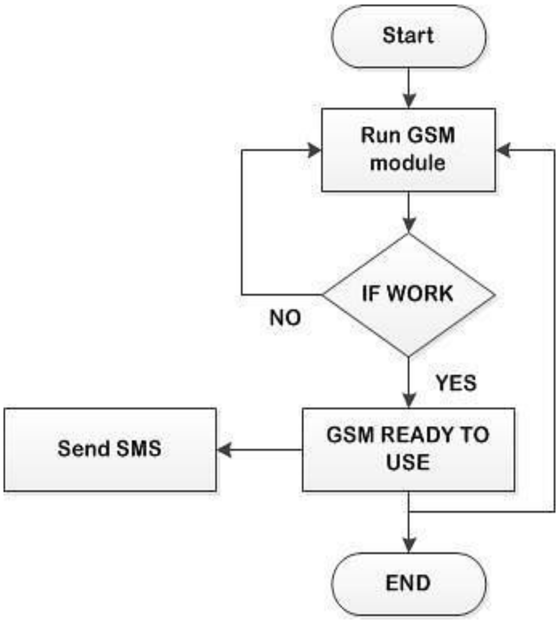
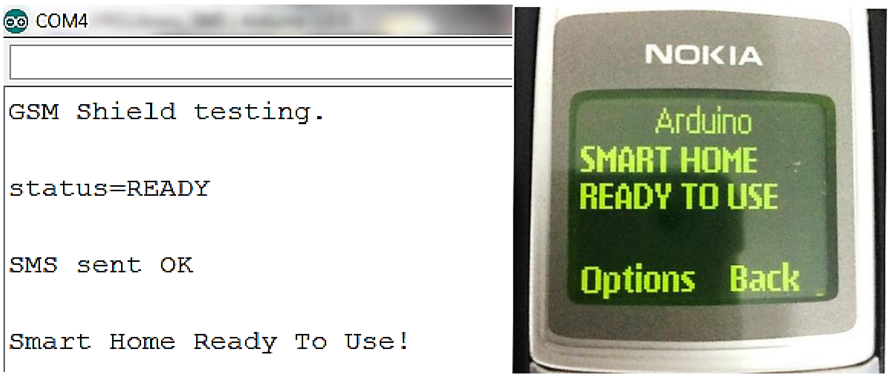


Figure 42: Workings of the GSM software



(A) **(B)**

Figure 43: GSM testing module

A GSM test module consists of two parts: (A) the serial port monitoring the GSM module test; and (B) the mobile phone receiving the SMS from the GSM module. After testing the GSM module and its functionality and connection to the network, the testing the second important unit in the system, SDRAM initialization followed. Checks were made to ascertain whether the SDRAM would save the sensor value. SDRAM includes three files to save data separately: temp.txt, rfid.txt and gas.txt. The system uses Figure 44 to check whether the SDRAM ready. Figure 45 shows the initialization of SDRAM.

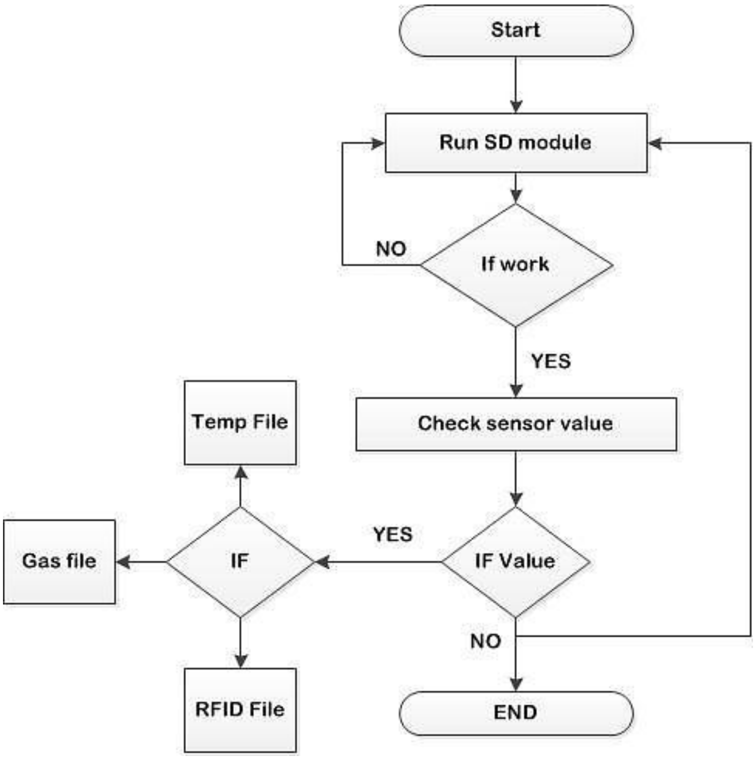
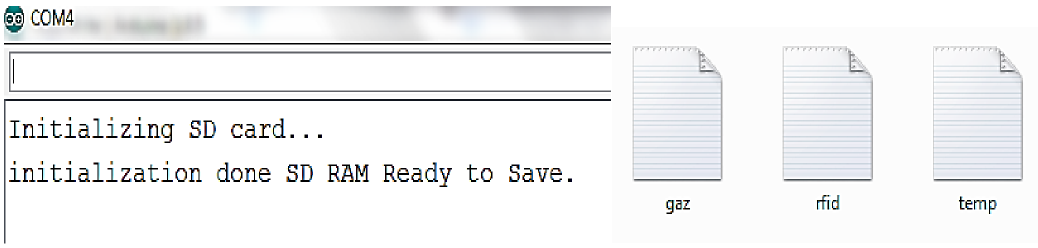


Figure 44: Flowchart for SDRAM software



(A) (B)

Figure 45: Test SDRAM module

Figure 45 consists of two parts: (A) the serial port monitoring the SDRAM module test and initialization files and (B) the creation of three files to save the data in our system.

After testing these two important units in the system, the system is now ready for testing the other hardware units. We start with the sensors: heat, gas and rain sensors. All of these sensors are analog sensors. To work with these sensors, it is necessary to convert from analogue format to digital format. The system monitors this value. If the value increases beyond the limited value, the system sends an alert SMS to the mobile, saves this value to SDRAM in a text file and give us sounds an alert using the buzzer. The flowchart in Figure 46 shows how the software sensors work. Figure

47 shows the reading result value from the sensor in the serial port and saves it to SDRAM.

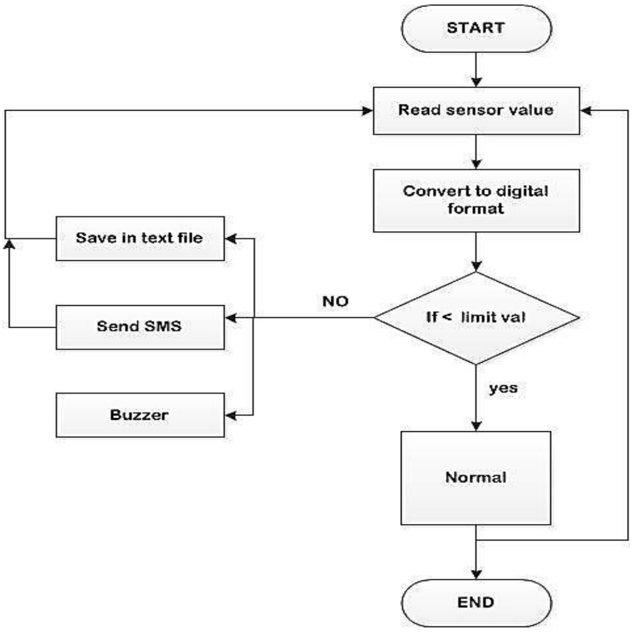


Figure 46: Flowchart for sensor software

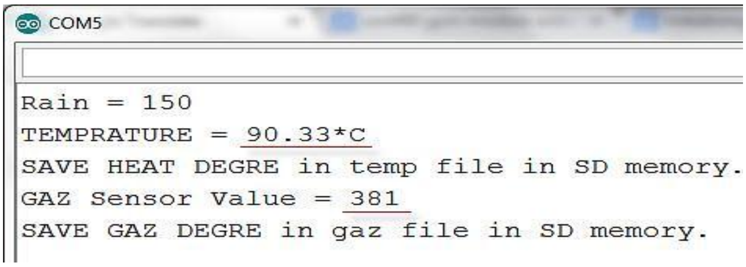


Figure 47: Reading sensor value and save it in SDRAM

(A) **(B)** **(C)**

Figure 48: Testing high sensors value

Figure 48 consists of three parts: (A) the hardware LEDs for gas and heat sensors to turn ON when exceeding the limit value; (B) this SMS alert is sent when the system finds any gas or smoke in the home; and (C) this SMS alert is sent when the system detects overheating in the home. The testing of every sensor, GSM module, SDRAM was deemed a success. Now, testing the other units of the system proceeds.

RFID is an important unit in the system. Its function is to control doors. It reads a smart card issued to people who are authorized to enter the home and saves any entry and exit events to SDRAM. The flowchart in Figure 49 shows how the software works with the RFID module and Figure 50 shows RFID testing while reading the card ID and saving it to DSRAM.

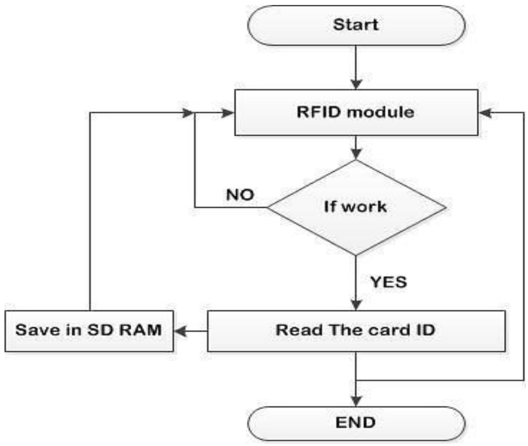


Figure 49: Flowchart for RFID software

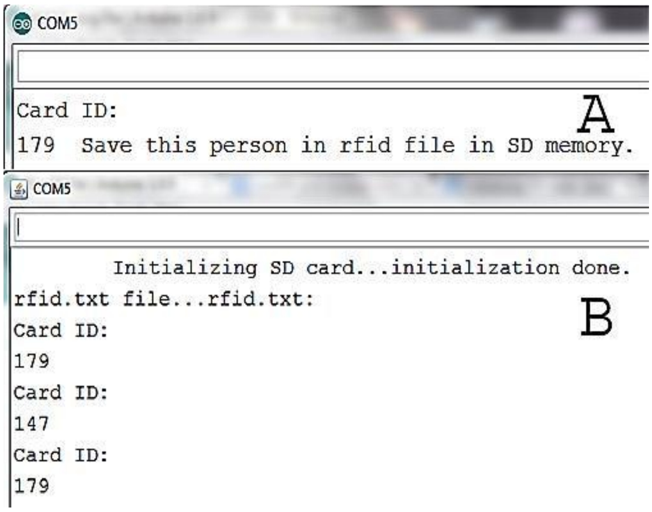


Figure 50: Serial port to RFID testing

Figure 50 consists of two parts: (A) this part of the serial port reads the smart card and saves it to SDRAM; (B) this part of the serial port is reading the rfid.txt file. The figure 51 shows the RFID unit and FRID LED when finds smart card.

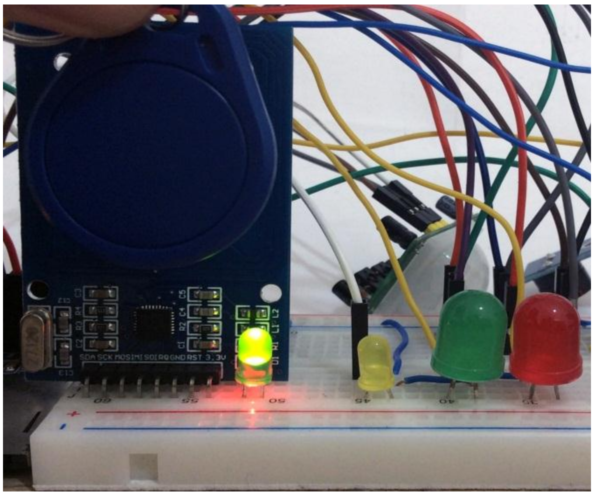


Figure 51: Hardware LED for RFID testing

Now, the other units of this system are tested. The PIR sensor is a digital sensor. It is connected directly to the Arduino Mega 2560 in the digital port. Figure 52 shows how the software works with the PIR sensor and Figure 53 shows the test for sensing motion.

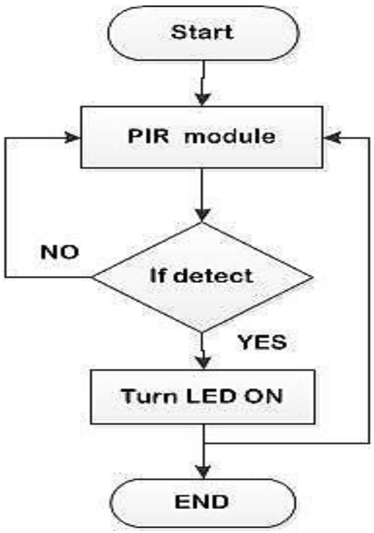


Figure 52: Flowchart for PIR software

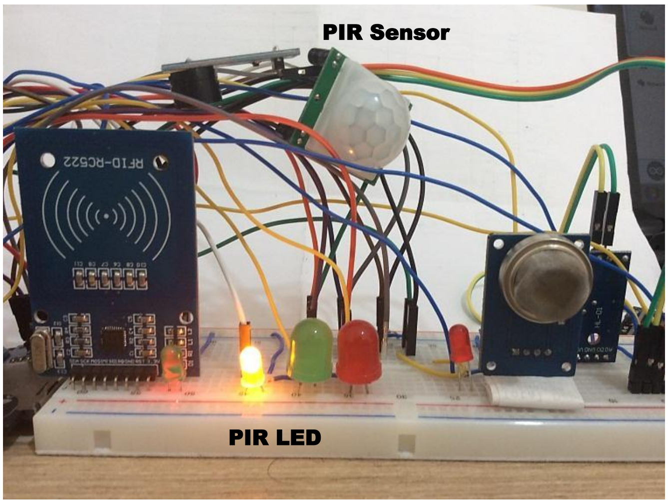


Figure 53: PIR sensor and PIR LED upon detection of motion

Now, the manual ON/OFF electronic device is tested using the SMS service. In this state, using two demos LEDs RED/GREEN. Every LED is the same as the electronic device. When an SMS is sent from any mobile phone to the system, the system turns these LEDs on or off depending on which devices want to control it. The flowchart in Figure 54 explains how the software works in this case. Figure 55 shows the result of these steps.

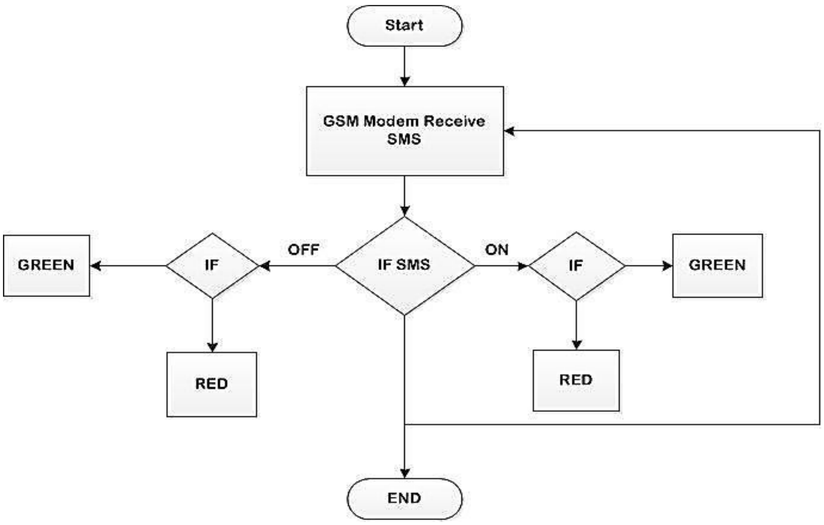


Figure 54: Flowchart for manual ON/OFF LEDs software

(A) **(B)** **(C)**

Figure 55: Manual ON/OFF LEDs testing

Figure 55 consists of three parts: (A) the sending of SMSs from the mobile; (B) the serial port when an SMS is received; (C) the hardware showing the Red LED when turned ON and Green LED when turned off.

The final step of the hardware part is the PCB of the system. The PCB is designed with the Eagle software package. This software is especially used for PCB design. The PCB of the intelligent home system consists of two layers: the top and bottom. Figure 56 shows the PCB circuit in the Eagle program. Figure 57 shows the printed PCB hardware.

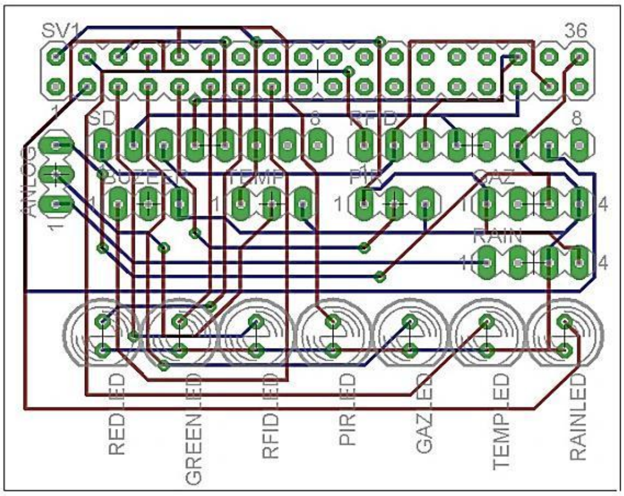
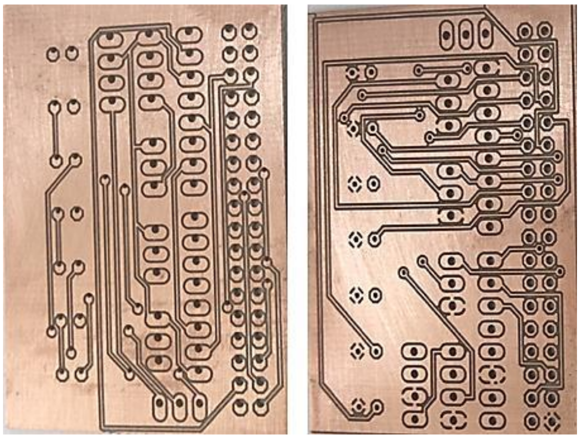


Figure 56: PCB circuit design



(A) **(B)**

Figure 57: Hardware PCB design

Figure 57 consists of two parts: the top face (A) and the bottom face (B).

4.3 Testing the Software System

The program interface of a PC is used with the hardware system to carry out real time monitoring and controlling in this system via the USB port. This software was built in the VB6 environment. With this software, one can read the current sensor value, read instructions from the Microcontroller, and turn electronic devices on and off. Two important tools are used: the MSComm Control tool to send and receive data via the serial port and Windows Media Player to play the alarm sound on the computer.

Many images are shown of software testing with the hardware device. Figure 58 shows the home page of the program before starting to work. This page consists of three parts: the first three rectangles represent the current reading sensor value. The second shows the switch button with ON/OFF electronic devices. The third rectangle shows the real-time instructions from the Microcontroller.

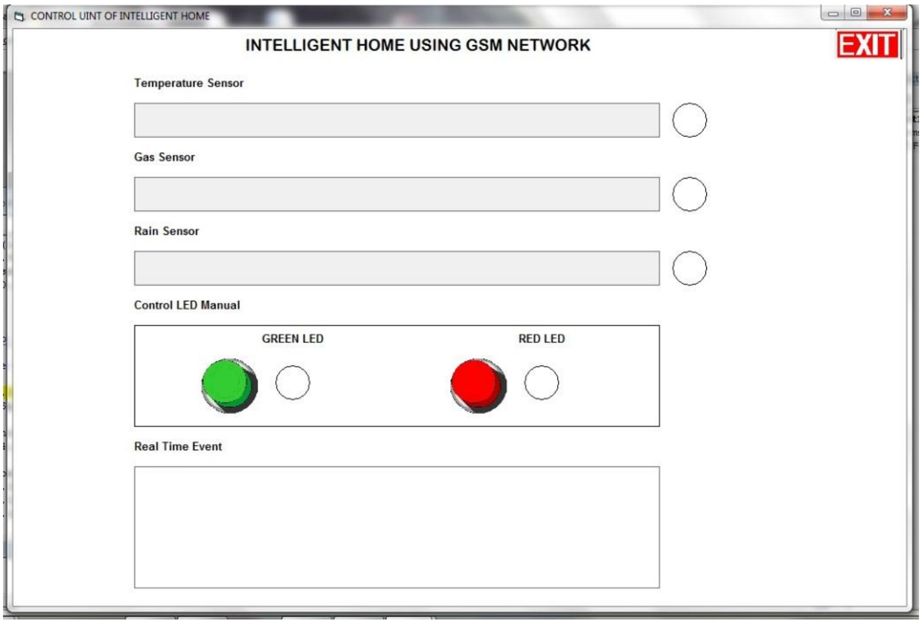


Figure 58: GUI for intelligent home software

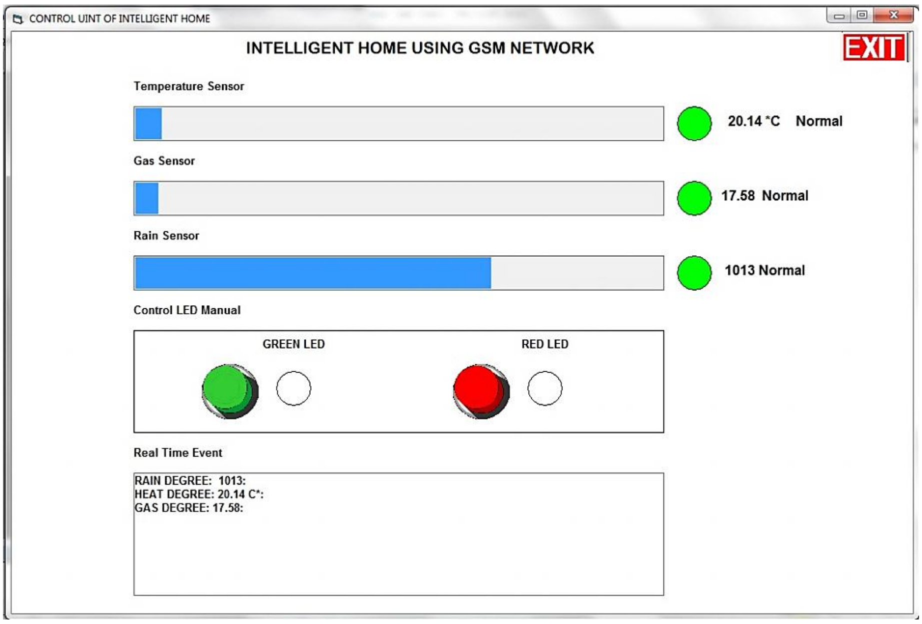


Figure 59: Software starting to operate

Figure 59 shows this software starting when data is received. It shows a normal current sensor value with devices turning off when instructions are received from the Microcontroller. Figure 60 shows software systems testing of increases in the heat sensor value.

Figure 60: Software testing for increases in temperature

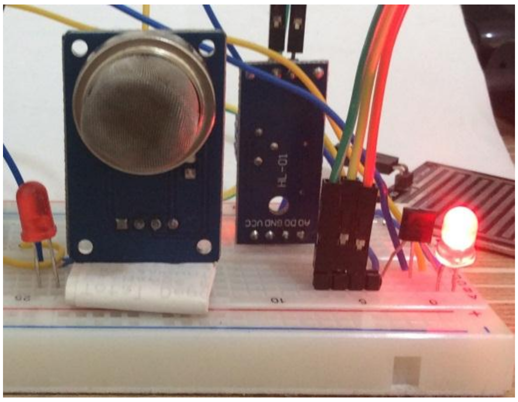


Figure 61: Hardware testing, when increased in heat value

Figure 62 shows software systems testing for increases in the gas sensor value.

Figure 62: Software testing for increases in gas value

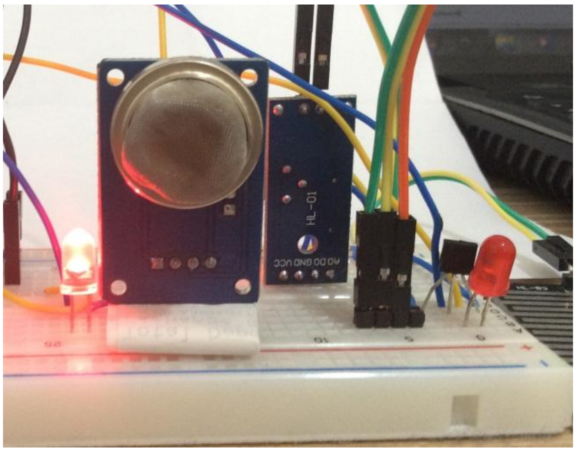


Figure 63: Hardware testing for increases in gas value

Figure 64 shows testing of software systems. Here, an increased rain sensor value is shown.

Figure 64: Software testing for rain

Figure 65 shows instructions from the USB port.

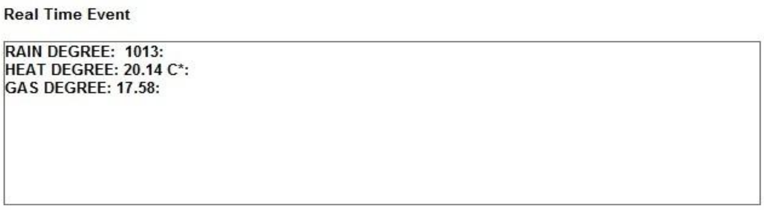


Figure 65: Instructions from the USB port

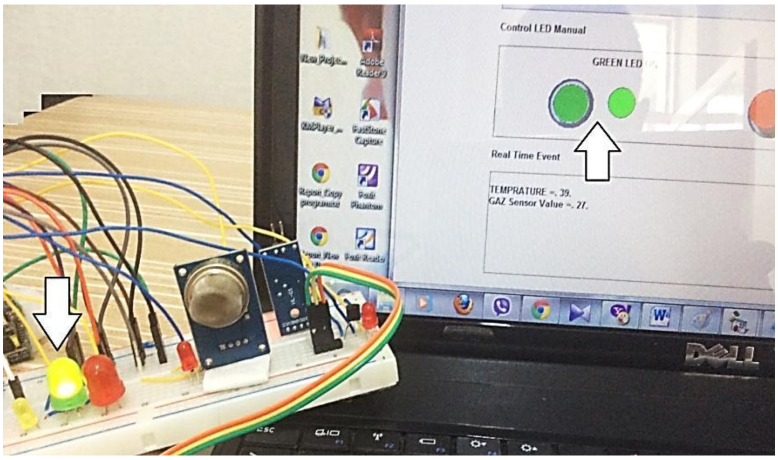


Figure 66: Software testing to turn on the green LED

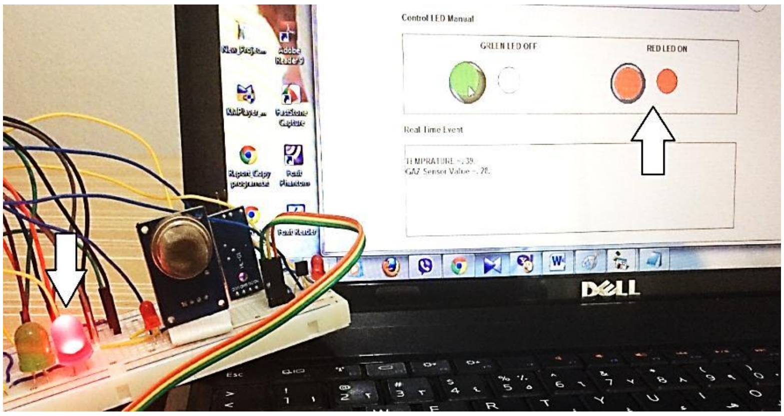


Figure 67: Software testing to turn on the red LED

CHAPTER 5

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORKS

5.1 Conclusions

The purpose of this research was to design and implement an intelligent home system by remotely controlling electronic devices in the home and monitoring of the temperature and gas degree inside the home by using a mobile phone. This work used a GSM network due to its low cost and coverage of a large area. It is used to send instructions to the system to control electronic devices and receive alerts. The goal has been achieved successfully. These devices are controlled by sending SMS messages. A temperature sensor uses a heat detector which turns on the LED when the heat increases. As a result, the system sends an SMS alert to the mobile. Moreover, the gas and rain sensors work correctly with very good results. Furthermore, in this work the SDRAM was successfully added to the system in order to save all events in the home. It used to check sensor values in the event of any problems with the GSM network, SIM or SMS service. This system also uses power saving by means of a PIR sensor in order to switch off lights in unoccupied rooms. An important feature in this system is that it saves into a text file the details of all persons who enter the home by using a Smart card.

The SIM 900 GSM/GPRS module, which uses an interface between the Arduino Mega 2560 and the mobile station, work correctly by using a new function: GSM/GPRS. The mobile phone does not require any special hardware or application to be used as a mobile station. Therefore, any mobile phone can be used as a remote control.

There are many types of high cost intelligent home systems on the market. In spite of this, designing and customizing one’s own intelligent home system and security will reduce costs. Not only on the economic side but also the technical side, there will be more flexibility with this concept. With this (customization) concept, the designer decides the number of sensors, the coverage area and the number of electronic devices.

The main advantages of this system are flexibility, economy and technical customization. All previous smart home systems are limited to available GSM networks and will not work if GSM networks are unavailable. The intelligent home system will work if a GSM network is not available as it takes many types of SMS alerts and Buzzers. In addition to the alerts, the system saves all events and the sensor values can be backed up. The smart card is successfully added to save all people who entered into the home in the SDRAM.

5.2 Suggestions for Future Works

Some suggestions for future works include:

1- Add a camera to this system for face recognition to identify persons automatically.

2- Design the mobile application to control an electronic device remotely by using a WIFI network.

3- Add a sound sensor to the children’s room to determine whether they are sleeping.

4- Add LCD touch screen to the system.

REFERENCES

1. **Atukorala K., Wijekoon D., Tharugasini M., Perera I., Silva C., (2009),** "*SmartEye Integrated Solution to Home Automation, Security and Monitoring Through Mobile Phones*", Next Generation Mobile Applications, Services and Technologies, IEEE Third International Conference on, pp.64–69.

2. https://www.snet.tu-berlin.de/fileadmin/fg220/courses/WS1112/snet- roject/. (Data Download Date: 20 July 2014).

3. **Zhai Y., Cheng X., (2011),** "*Design of Smart Home Remote Monitoring System Based on Embedded System"*, Control and Industrial Engineering, IEEE 2nd International Conference, pp.41-44 .

4. **Gurek A., Gur C., Gurakin C., Akdeniz M., Metin S. K., Korkmaz I., (2013),** "*An Android Based Home Automation System*", High Capacity Optical Networks and Enabling Technologies, IEEE 10th International Conference on, pp.121-125.

5. **Tan K. K., Lee T. H., Soh C. Y., (2002),** "*Internet Based Monitoring of Distributed Control Systems an Undergraduate Experiment*", Education, IEEE Transactions on, vol.45, no.2, pp.128–134.

6. **Yüksekkaya B., Kayalar A. A., Tosun M. B., Özcan M. K., Alkar A. Z., (2006),** "*A GSM, Internet and Speech Controlled Wireless Interactive Home Automation System*", Consumer Electronics, IEEE Transactions on, vol.52, no.3, pp.837-843.

7. **Yamazaki T., (2006),** *"Beyond The Smart Home",* Hybrid Information Technology, IEEE International Conference on, vol.2, pp.350-355.

8. **Ogawa M., Tamura T., Yoda M., Togawa T., (1997),** "*Fully Automated Biosignal Acquisition System For Home Health Monitoring"*, Engineering in Medicine and Biology Society, IEEE Proceedings of the 19th Annual International Conference on, vol.6, pp.2403-2405.

9. **Sriskanthan N., Tan F., Karande A., (2002),** "*Bluetooth Based Home Automation*", Microprocessors and Microsystems, Elsevier Science B.V. vol.26, no.6, pp.281–289.

10. **Al-Ali A. R., Al-Rousan M., (2004),** "*Java Based Home Automation System*", Consumer Electronics, IEEE Transactions on, vol.50, no.2, pp.498-504.

11. **Deepti S., (2014),** "*Home Automation System with Universally Used Mobile Application Platform*", IOSR Journal of Electronics and Communication Engineering, vol.9, no.2, pp.01-06.

12. **Piyare R., Tazil M., (2011),** "*Bluetooth Based Home Automation System Using Cell Phone*", Consumer Electronics, IEEE 15th International Symposium on, vol.45, no.3, pp.192-195.

13. **Javale D., Mohsin M., Nandanwar S., Shingate M., (2013),** "*Home Automation and Security System Using Android ADK*", International Journal of Electronics Communication and Computer Technology, vol.3, pp.382-385.

14. **Teymourzadeh R., Ahmed S. A., Kok W. C., Mok V. H., (2013),** "*Smart GSM Based Home Automation System*", Systems, Process & Control, IEEE Conference on, pp.306–309.

15. **Peijiang C., Xuehua J., (2008),** "*Design and Implementation of Remote Monitoring System Based on GSM*", Computational Intelligence and Industrial Application, IEEE Pacific Asia Workshop on, vol.1, pp.678-681.

16. **ElKamchouchi H., ElShafee A., (2012),** "*Design and Prototype Implementation of SMS Based Home Automation System*", Electronics Design, Systems and Applications, IEEE International Conference on, pp.162–167

17. **Coskun I., Ardam H., (1998),** "*A Remote Controller for Home and Office Appliances by Telephone*", Consumer Electronics, IEEE Transactions on, vol.44, no.4, pp.1291–1297.

18. http://arduino.cc/en/Guide/Introduction. (Data Download Date: 26 July 2014).

19. http://arduino.cc/en/Main/ArduinoBoardUno. (Data Download Date: 26 July 2014).

20. http://arduino.cc/en/Main/ArduinoBoardMega2560. (Data Download Date: 26 July 2014).

21. http://processing.org/. (Data Download Date: 26 July 2014).

22. http://playground.arduino.cc/. (Data Download Date: 26 July 2014).

23. **Anttalainen T., (2003),** "*Telecommunications Network Engineering*", 2nd Edition, United States of America, pp.212-216.

24. http://www.activexperts.com/mmserver/cellular/gprsintro/. (Data Download Date: 27 July 2014).

25. **Campus Component Pvt. Ltd., (2011),** "*GSM Interfacing Board*", India, pp.3-4.

26. **Jacob F., (2010),** "*Handbook of Modern Sensors*", 4th Edition, New York, pp.19.

27. **Instruments T., (1999),** "*LM35 Precision Centigrade Temperature Sensors*", Texas Instruments Incorporated, pp.1.

28. **Punniamoorthy D., Vimal P. K., (2014),** "*Modified Toll Gate System with Enhanced Security Using FPGA*", IOSR Journal of VLSI and Signal Processing, vol.4, no.2, pp.52-61.

29. **Klaus F., (2010),** "*RFID HandBook"*. 3rd Edition, Germany, pp.1-6.

30. https://learn.adafruit.com/pir-passive-infrared-proximity-motion-ensor/how-pirs- work. (Data Download Date: 29 July 2014).

31. https://learn.adafruit.com/downloads/pdf/adafruit-micro-sd-breakout-board-card-tutorial.pdf. (Data Download Date: 29 July 2014).

32. http://www.tinkbox.ph/sites/tinkbox.ph/files/downloads/RAIN_SENSOR_MODULE.pdf. (Data Download Date: 29 July 2014).

33. http://www.seeedstudio.com/wiki/GPRS_Shield_V1.0. (Data Download Date: 30 July 2014).

34. http://arduino.cc/en/Main/ArduinoBoardNano. (Data Download Date: 31 July 2014).

35. **Budijono S., Andrianto J., Noor M. A. N., (2014),** "*Design and Implementation of Modular Home Security System with Short Messaging System*", EPJ Web of Conferences, EDP Sciences, vol. 68, pp.1-5.

36. http://www.open-electronics.org/gsm-gps-shield-for-arduino/.(Data Download Date: 30 July 2014).

37. http://www.farnell.com/datasheets/911651.pdf. (Data Download Date: 14 August 2014).

38. http://www.seeedstudio.com/wiki/Grove_-_Gas_Sensor%28MQ2%29. (Data Download Date: 15 August 2014).

39. **Kugelstadt T., (2011),** "*Extending the SPI Bus for Long Distance Communication*", Analog Applications Journal, Texas, pp. 16-22.

40. http://arduino.cc/en/Reference/SPI. (Data Download Date: 16 August 2014).

41. http://arduino.cc/en/Reference/SoftwareSerial. (Data Download Date: 16 August 2014).

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ALGOIARE, Omar Talal
Date and Place of Birth: 02 July 1987, Diyala
Marital Status: Single
Phone: +90 539 201 93 62
Email: omar_talal20022003@yahoo.com

EDUCATION

Degree	**Institution**	**Year of Graduation**
M.Sc.	Çankaya Univ., Computer Engineering	2014
B.Sc.	Technical College Kirkuk, Software Engineering	2011
High School	Al-Maaref High School	2007

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2011	Al-Yarmok Univ. Department of Computer Engineering.	Lecturer

FOREIN LANGUAGES

Advanced Arabic, English, Beginner Turkish

HOBBIES

Football, Reading, Swimming.