Mehhatroonikainstituut

Mehhanosüsteemide komponentide õppetool

MHE70LT

Riva

JUHTMEVABA KODU AUTOMATISEERIMISE SÜSTEEM

MSc Lõputöö

Autor taotleb

tehnikateaduste magistri

akadeemilist kraadi

Tallinn
2016

Department of Mechatronics
Chair of Mechanosystem Components

MHE70LT

Riva

WIRELESS HOME AUTOMATION SYSTEM

MSc Thesis

The author applies for
The academic degree
Masters of Science in Mechatronics Engineering

Tallinn
2016

AUTHOR'S DECLARATION

I declare that I have written this graduation thesis independently.

These materials have not been submitted for any academic degree.

All the works of other authors used in this thesis have been referenced.

The thesis was completed under Professor Vu Trieu Minh supervision

“22nd May, 2016

Author: Riva

.... Signature

The thesis complies with the requirements for graduation theses.

“.......”....................201….

Supervisor: Vu Trieu Minh

.... Signature

Accepted for defence.

.... Chairman of the defence committee

“.......”....................201….

.... Signature

TUT Department of Mechatronics
Chair of Mechatronics

MASTERS THESIS TASK

2016 (spring) Semester

Student: Riva (A145848)
Study programme: MAHM02/13
Speciality: Mechatronics
Supervisor: Professor, Head of Mechanosystem Components, Vu Trieu Minh

THESIS TOPIC:

(in Estonian) : **JUHTMEVABA KODU AUTOMATISEERIMISE SÜSTEEM**

(in English) : **WIRELESS HOME AUTOMATION SYSTEM**

Assignments to be completed and the schedule for their completion:

NR.	Description of tasks	Timetable
1	Research on background possibilities, learning about Arduino ,Xbees other hardware's to be used or not	**Week 5-6**
2	Android Application making user interface and making it able to connect to server.	**Week 7-8**
3	Processing with MATLAB. working on model with Arduino and Xbees	**Week 9-10**
4	Getting data into Arduino from sensor and Sending the data through the 2 Xbees to the MATLAB.	**Week 11-12**
5	Working with databases and compiling work (writing summary and adding forms)	**Week 13-14**

Engineering and economic problems to be solved:
The overall thesis task is divided into 3 main parts Part 1: Sensor to Computer which includes getting data into Arduino from sensor and Sending the data through the 2 Xbees to the laptop. Part 2: Processing in Computer and upload to server which includes getting data from Xbees into MATLAB, MATLAB processing and uploading data to server/internet. Part 3: App in android which includes User interface and Phone downloading data from server and displaying

Language of the thesis: English

Deadline for submission of the application for defence **12/02/2016**
Deadline for submission of the thesis 01/06/2016

Student …………………….. /signature/ ……….. Date………

Supervisor……………………… /signature/ ………. date………

TABLE OF CONTENTS

Foreword

First of all, I would like to thank the Tallinn University of Technology, which gave me the opportunity to acquire a master's degree. The curriculum gave me a chance to increase my practical knowledge through this innovative program. I obtained my bachelor's degree in electronics and communication so it was a new experience and solid mechanics professional learning opportunity.

Secondly, I would like to thank Professor Mart Tamre, Chair of Mechatronic, who was supportive and helpful in the past two years. He has not only demonstrated professionalism, but also always tried to understand the problems with students and find the best solution for them.

I would also like to thank my supervisor, Professor Vu Trieu Minh, Head of Mechanosystem Components. My sincere thanks to him for motivating me to write till end of the work. He was interested in my previous education and proposed a similar theme. He was always helpful and helped me with my doubts and confusion in the case. Also the techniques he suggested were always practical, and I thank, Professor Vu Minh Trieu for the opportunity to prepare thesis under his supervision.

Finally, I should like to thank my family and friends, who were the main reasons for choosing this theme, the development, and the execution. Despite the difficulties, their support was very important, and sympathizing.

Eessõna

Esiteks ma sooviksin tänada Tallinna Tehnikaülikooli, kes andis mulle võimaluse omandada magistrikraadi. Õppekava andis mulle võimaluse suurendada praktilisi teadmisi läbi selle innovatiivse programmi. Ma omandasin oma bakalaureuse kraadi elektroonika ja kommunikatsiooni alal. See oli uus kogemus ja hea mehaanika eriala õppimisvõimalus.

Teiseks, ma sooviksin tänada professor Mart Tamret, mehhatroonikasüsteemide õppetooli juhatajat, kes oli toetav ja abivalmis viimase kahe aasta jooksul. Ta ei ole mitte ainult demonstreerinud professionaalsust, vaid ka alati proovinud mõista tudengite probleeme ning leida neile parim lahendus.

Ma sooviksin samuti tänada oma juhendajat, professor Vu Minh Trieud, ehhanosüsteemide komponentide õppetooli juhatajat. Minu siiras tänu talle motiveerimaks mind seda lõputööd kirjutama. Ta oli huvitatud minu eelevast haridusest ja pakkus välja sarnase teema, mida saaksin teostada ja õppida. Ta oli alati abivalmis ja aitas mind mu kahtluste ja segaduse puhul. Samuti tema pakutud tehnikad olid alati praktilised ning tänan professor Vu Minh Trieu-d võimaluse eest see lõputöö tema juhendamisel valmistada.

Viimaks sooviksin veel tänada oma pere ja sõpru, kes olid peamised põhjused selle teema valimisel, arendusel ning täideviimisel. Vaatamat raskustele oli nende toetus ja kaasaelamine väga oluline.

1 INTRODUCTION

Home automation is the next step in the endeavour of the community of engineers to improve the quality of life, so that people can concentrate on the more important things in their lives. These systems combines all the electronics in any given home into a single system. The home automation systems are getting more and more popular these-days as they are becoming cheaper and more affordable for the masses. Their reliability has also improved in the recent years and with the advent of smartphones, tablets and the improved internet connectivity all over the world has made it possible for people to control the devices in their homes to be controlled from any corner of the world.

Here in this work of mine I will introduce my system for kitchen automation. It aimed at helping physically disabled and the elderly. This system can also great help for people with busy schedules and hectic lifestyles which includes almost everyone these days. It gives the user the ability to control the kitchen appliances using just their smartphones/tablets. These appliances may include microwave ovens, kettles, refrigerators, stoves, electric cookers etc.

Assistive domotics is a field in home automation particularly focused on the elderly and people with disabilities, aimed at making their lives a whole lot easier and comfortable. These system can give them the sense of safety and ease of use by providing them with features like voice control and gesture controls for those who have disabilities.

The elderly and differently abled go through hard time working, cooking and moving around the kitchen and they often need to hire help to make their lives easier. Automatic kitchen gives them the choice of using technology to accomplish their needs instead of depending on others. As lives nowadays are getting excessively occupied and individuals for the most part don't have time for cooking and investing a lot of energy in kitchen .So appliances ought to be that cutting-edge as far as sparing time and remind them about vital things such as their food items in fridge is going to lapse soon so they can purchase a few staple goods soon.

The sensors from the appliances collect necessary data like temperature, pressure etc., and these collected data are then transferred to the microcontroller. Here Arduino Uno module is used to do the same. It collects and processes the signals and send them wirelessly to another module, the raspberry pi, using a zig bee module. The raspberry pi runs an Open HAB server using which the appliances are controlled with smartphones/tablets or any other device that can access the internet as a user interface device.

In this task I am going to build up an android application which I named as Kool kitchen .It helps distinctively capable individuals to remotely utilize their kitchen apparatuses. Likewise warning on advanced mobile phones offers them some assistance with updating about what is occurring in their kitchens. I will explain more about this application later in this report with all data and programming used. The basic idea and the theory is given below.

1.1 Fantastic Fridge

As the name suggests it's a fridge with fantastic features. The fantastic features involves information about items in their refrigerator like expiry dates, amount of food left in the fridge. The information will be provided with the help of an android app

For that we need a barcode reader in the fridge. Barcodes initially were examined by unique optical scanners called barcode Readers. Later applications programming got to be accessible for gadgets that could read pictures, for example, cell phones with cameras. In this project I am going to make an android app called fantastic fridge which tells us about the expiry dates of remaining items and how much items are still left in the fridge.

Barcode printer will print date, year and month. So user can pick any dates for packed or cooked food.

Barcode reader will read that date and save that date in the data base and accordingly user will get notifications on their smart phones

For the items those has no expiry dates on them In the fantastic fridge there is barcode printer user can chose expiry dates like 1 day, 2 days for cooked food and 1 week for fruits and vegetables. All other dairy products like milk, butter, cream etc. soft drinks and medicines barcode will read the expiry dates and we get a notification on our android stating how much items is still left in the fridge and it's time to get groceries from the shop plus whether items are expired or not .

Below is the 2 d diagram of the fantastic fridge with barcode reader and barcode printer.

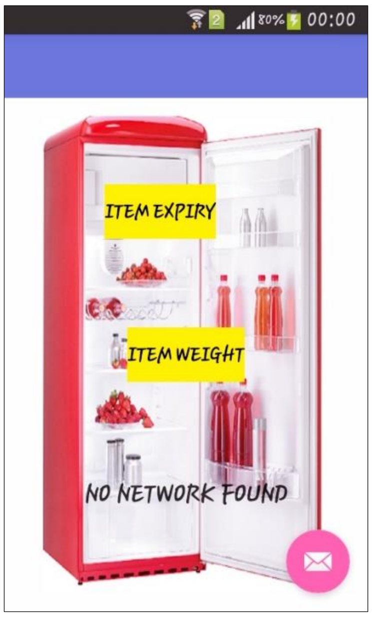


Figure 1 Fantastic Fridge

1.2 Smart Stove

Smart stove is another advancement in the stove I am trying to make to help my targeted customers.

First advancement here in the smart stove we have some timers installed user can set their own timers and after that time the stove will turn off. Through using the above stated automatic system we can turn off and turn on and off our stove remotely. Another advanced features that we can add to smart stove is some vessels attached with temperature sensors when the vessel temperature is of certain temperature stove will turn off automatically.

This system works in the same way Fantastic fridge works in which Temperature sensor is attached to microcontrollers and microcontrollers is attached to central hub which sends signals to smart phones and user will be able to control device remotely. In this work I have made prototype using stove and we will also see Simulink simulations towards the ends of this task. Temperature will be constantly given to android application and to the central HUB for monitoring.

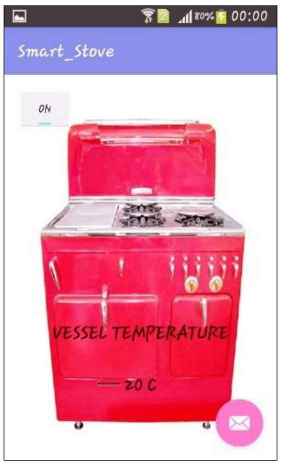


Figure 2 Smart Stove

1.3 Master Microwave

Using the above stated automatic system we can turn off and turn on and off our microwave remotely. Also as a help to physically challenged people master microwave will send a text / notification on smart phones when task is completed inside the microwave.

For that to happen we can use some position sensors or cameras. With the help of temperature sensor constant monitoring of data will be done and the same prototype which is used for stove could be used here.

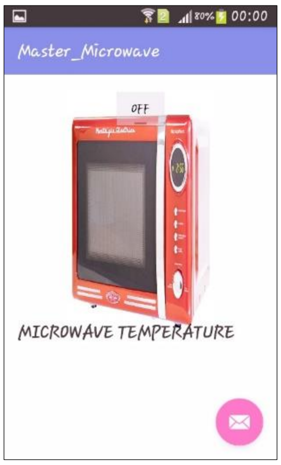


Figure 3 Master Microwave

1.4 Background Research

A standardized identification is an optical machine-clear representation of information identifying with the article to which it is connected. Initially standardized tags methodically spoke to information by changing the widths and spacing's of parallel lines, and might be alluded to as direct or one-dimensional (1D). Later two-dimensional (2D) codes were created, utilizing rectangles, specks, hexagons and other geometric examples in two measurements, for the most part called standardized tags despite the fact that they don't utilize bars in that capacity. Standardized identifications initially were checked by extraordinary optical scanners called standardized identification peruses. Later applications programming got to be accessible for gadgets that could read pictures, for example, cell phones with cameras.

An early utilization of one sort of scanner tag in a modern connection was supported by the Association of American Railroads in the late 1960s. Created by General Telephone and Electronics (GTE) and called Karaka ACI (Automatic Car Identification), this plan included setting shaded stripes in different mixes on steel plates which were attached to the sides of railroad moving stock. Two plates were utilized per auto, one on every side, with the game plan of the hued stripes encoding data, for example, proprietorship, sort of gear, and distinguishing proof number.] The plates were perused by a trackside scanner, situated for occurrence, at the passageway to a characterization yard, while the auto was moving past. The venture was relinquished after around ten years on the grounds that the framework demonstrated inconsistent after long haul use.

Standardized identifications turned out to be economically fruitful when they were utilized to mechanize grocery store checkout frameworks, an assignment for which they have turned out to be verging on general. Their utilization has spread to numerous different errands that are blandly alluded to as programmed ID and information catch (AIDC). The principal examining of the now omnipresent Universal Product Code (UPC) standardized identification was on a pack of Wrigley Company biting gum in June 1974. [1]

System like this is already available in the market like LG home chat which are using android interface to work on it however system provided by me facilitates some other advancement as it is not just chatting also weighing and barcode printer in the fridge gives user freedom to choose expiry date of certain items as it is very important to eat healthy food.

LG Home Chat incorporates the popular LINE application to allow users to receive recommendations and control settings when away from home. With an intuitive interface, Home Chat makes communicating with LG's smart refrigerator, washing machine or oven much like chatting with a close friend. For extra convenience, the Quick Button feature enables fast and easy access to each appliance's most commonly used functions. Home Chat also gives users the choice of three different modes. [2]

The advantage my system has on the system already available in market is that user need not to communicate by themselves with their devices however devices send user notifications on their smart phones as reminder. Another advantage is in the fridge I have installed a barcode reader user can choose expiry dates of items which are cooked. For packed food user can just print dates mentioned on food containers.

As I am going to use barcodes in one of my devices there is a standardized identification standard called GS1 that encodes the best before dates. GS1 is available on a few pharmaceuticals right now. The scanner tags most as often as possible found in grocery stores use UPC and EAN standardized identification organizes and don't encode the expiry dates. The measure of the data that can be incorporated into an UPC scanner tag is regularly considered too little to incorporate the article distinguishing proof number and a date. The GS1 codes utilizes Code 128 configuration which can store 128 bytes. All that anyone could need for dates. [3] For making this system back ground research on zig bee is done. [4] Connections between Arduino and zig bee are made. [5]

2 HOME AUTOMATION

The sensors from the appliances collect necessary data like temperature, pressure etc., and these collected data are then transferred to the microcontroller. Here Arduino Uno module is used to do the same. It collects and processes the signals and send them wirelessly to another module, the raspberry pi, using a zig bee module. The raspberry pi runs an Open HAB server using which the appliances are controlled with smartphones/tablets or any other device that can access the internet as a user interface device.

In this task I am going to build up an android application which I named as Kool kitchen .It helps distinctively capable individuals to remotely utilize their kitchen apparatuses. Likewise warning on advanced mobile phones offers them some assistance with updating about what is occurring in their kitchens. I will explain more about this application later in this report with all data and programming used.

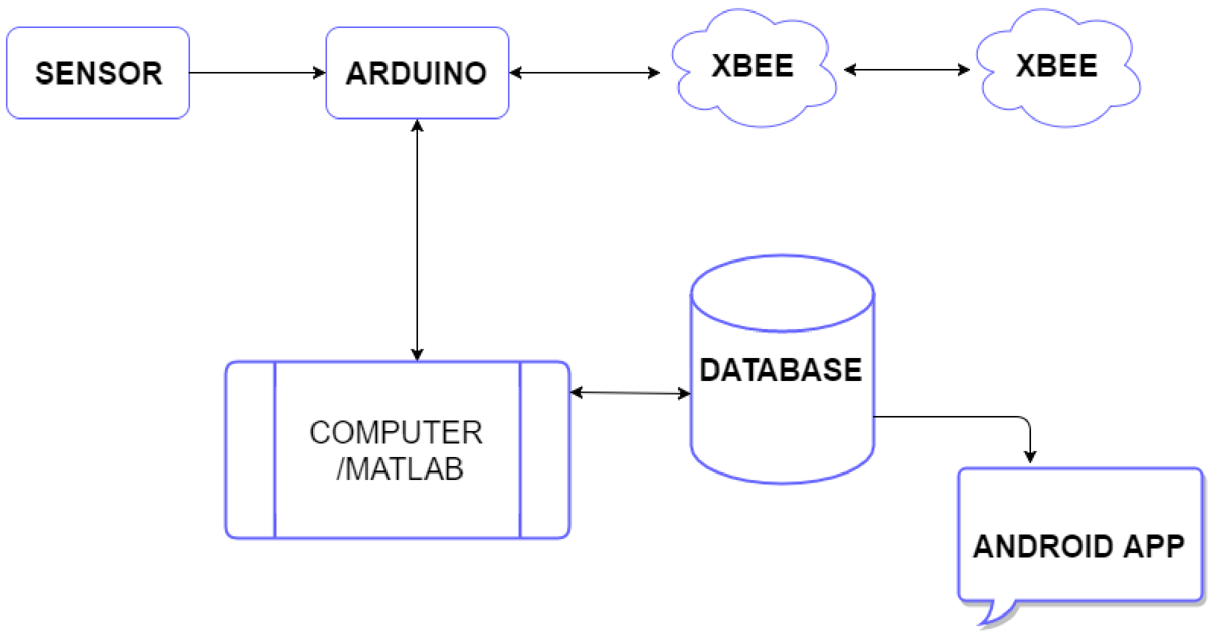


Figure 4 Home Automation System

The system diagram above is self-explanatory as this is a wireless system we need Arduino, zig bee and raspberry pi to connect to internet and used in phones later. Different kinds of sensors are used in my system like weighing sensor in fridge.

The diagram shown above was my initial plan when I started working on this research thesis however when I started executing my plan for thesis that changed a bit have not used Raspberry pi and instead of open hub I have used Web Server to store my data.

So the overall thesis task is divided into 3 main parts

Part 1: Sensor to Computer

1. Getting data into Arduino from sensor

2. Sending the data through the 2 Xbees to the laptop

Part 2: Processing in Computer and upload to server

1. Getting data from Xbees into MATLAB

2. MATLAB processing

3. Upload data to server/internet

Part 3: App in android

1. User interface

2. Phone downloading data from server and displaying.

For the experiment I am going to use temperature sensor. The temperature sensor as the name suggests is used to measure the temperature. It converts heat energy into electrical energy and gives a current signal as an output. In my system it is to be placed inside the oven, refrigerator, kettle etc., to measure the temperature of the food that is being cooked so that it can be displayed to the user who can use the data to get to know the status of the food, also this temperature is used to control the ON/OFF state of the appliance as pre-programmed in the system.
Arduino is the chip produced by a company called Arduino, which is an open source hardware and software company. It can be used to design and build digital devices that can be used to obtain data from sensors and this data can be used to control various devices and their operations.
They have a set of digital and analogue I/O pins which can be programmed to generate signals with the help of the integrated development environment specific to these Arduino boards.

The role of the Arduino Uno in this project is to get input from the sensors to get data about the system and this data is compared to the set values using the programs that have been loaded into the Arduino Uno. For example the program can be used to compare the current temperature of the food to the present value so that once it reaches the present value the appliance can be turned off and a notification can be given to the user that the food is ready.

The zig bee sends the data that is processed by the Arduino to the raspberry pi module wirelessly so that the location of sensors are not constrained by wiring problems. [6]
The Raspberry Pi is credit-card sized computer, without a display or a keyboard or a mouse. It has ports which plugs into a computer monitor or TV, and we can use a standard keyboard and mouse which can be plugged into the USB ports. It is very capable and in getting more and more popular these-days among electronic hobbyists and in prototype developments. It runs Open HAB software which is an open source server specifically designed for automation applications.
There are many types of systems and devices in the market and in order to make interaction between these devices easier we need a common middle ground to talk in. The open HAB server plays that role of a translator between systems that cannot otherwise speak to each other directly.

It is very important interfacing between Arduino and zig bee as our devices has micro controllers

2.1 Hardware to be used

Below is the list of hardware I will be using in my whole work.

2.1.1 Sensors

Sensors or transducers convert various types of energy into electric signals which can be used to measure either the presence or the intensity of the observed phenomenon.
The main sensors to be used in this project includes

- **Temperature sensor**

The temperature sensor as the name suggests is used to measure the temperature. It converts heat energy into electrical energy and gives a current signal as an output.
In my system it is to be placed inside the oven, refrigerator, kettle etc., to measure the temperature of the food that is being cooked so that it can be displayed to the user who can use the data to get to know the status of the food, also this temperature is used to control the ON/OFF state of the appliance as pre-programmed in the system.

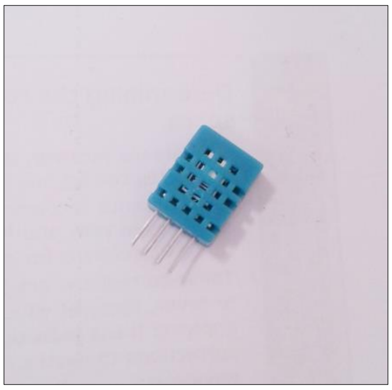


Figure 5 DHT11 Temperature Sensor

2.1.2 Arduino Uno

This chip is produced by a company called Arduino, which is an open source hardware and software company. It can be used to design and build digital devices that can be used to obtain data from sensors and this data can be used to control various devices and their operations.
They have a set of digital and analogue I/O pins which can be programmed to generate signals with the help of the integrated development environment specific to these Arduino boards.

Role in the System

The role of the Arduino Uno in this project is to get input from the sensors to get data about the system and this data is compared to the set values using the programs that have been loaded into the Arduino Uno. For example the program can be used to compare the current temperature of the food to the present value so that once it reaches the present value the appliance can be turned off and a notification can be given to the user that the food is ready.

2.1.3 Xbee

Xbee is an IEEE 802.15.4 wireless standard that is very energy efficient and is suitable for use in home automation for the following reasons.

● Low power consumption, a couple of AA batteries can run it for 1-2 years

● has a long range (up to 1000m) as it uses mesh routing where each device using zig bee can work as an access point

● Bitrate of up to 250kbps which is enough bandwidth to send necessary information across devices in home automation.

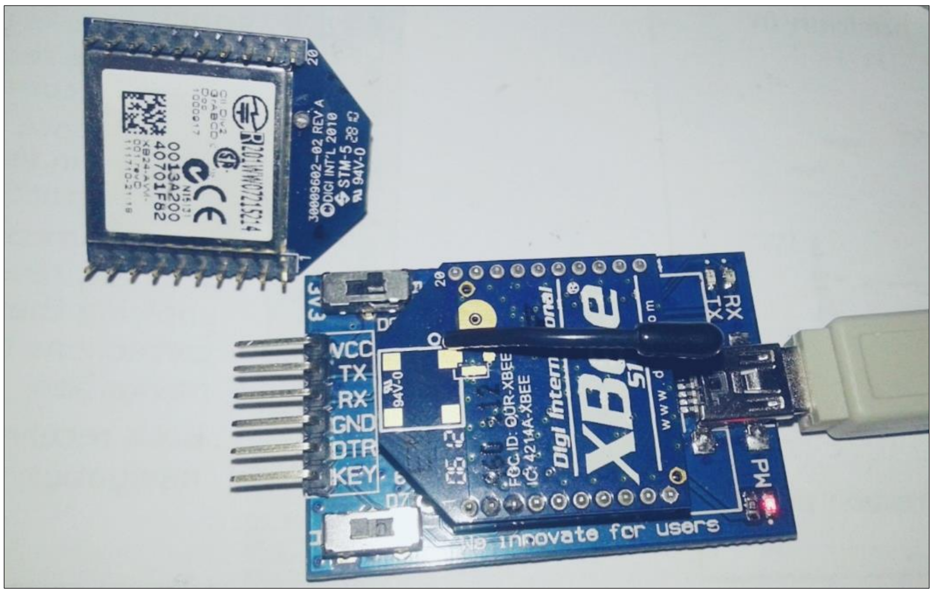


Figure 6 XBee

Role in the System

The zig bee sends the data that is processed by the Arduino to the raspberry pi module wirelessly so that the location of sensors are not constrained by wiring problems.

2.1.4 OPEN SERVER

There are many types of systems and devices in the market and in order to make interaction between these devices easier we need a common middle ground to talk in. The open HAB server plays that role of a translator between systems that cannot otherwise speak to each other directly. In the initial stages of my thesis I thought of using this server but as the development of the project went forward I have used WAMP server and online servers to connect to my android device.

So what is the role of Wamp server in my thesis .It is to connect to local host as machine wants to operate on PHP codes I need an operating system and sever. My SQL is used with PHP script so I downloaded Wamp Server.

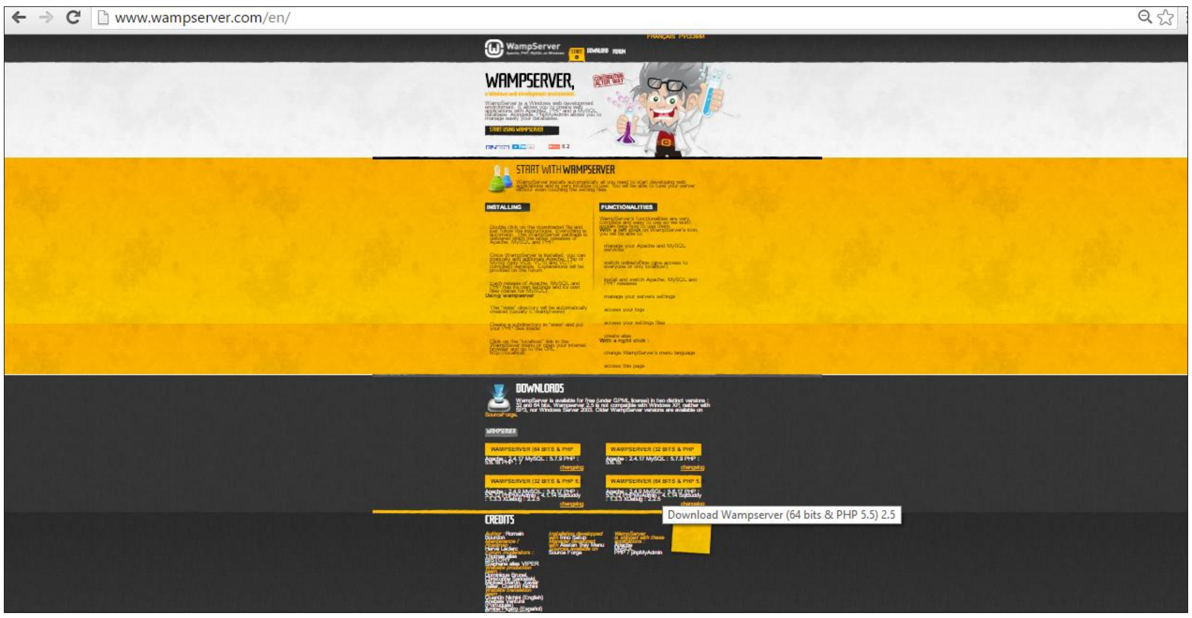


Figure 7 Wamp Server Download

For that go to Google >> type Wamp server >> select the operating system you are using .I have selected 64 bits according to my computer configuration. Click on download>> Click on executable file >> Install the file >> After Install is done.

Start the Wamp server .Add PHP scripts and add database Table and the server will be up and running.

3 KOOL KITCHEN (ANDROID APPLICATION)

Kool kitchen is the name given to the android application I am going to make through which we can control kitchen appliances like fridge microwave and stove.

I have designed logo for Kool kitchen. Kool kitchen is further divided into 3 subunits named as

- Fantastic fridge
- Smart stove
- Master microwave

As soon as we click on the icon of Kool kitchen app will take us to the new window from which user can chose devices they want to work with. So the user screen looks like the picture below. We can see in the pictures symbols for all the smart devices.

For android application I have learnt android programming. Application is made in Android studio with java and xml codes. For doing the same our computer should have java jdk downloaded and then we can download Android Studio to work on it.

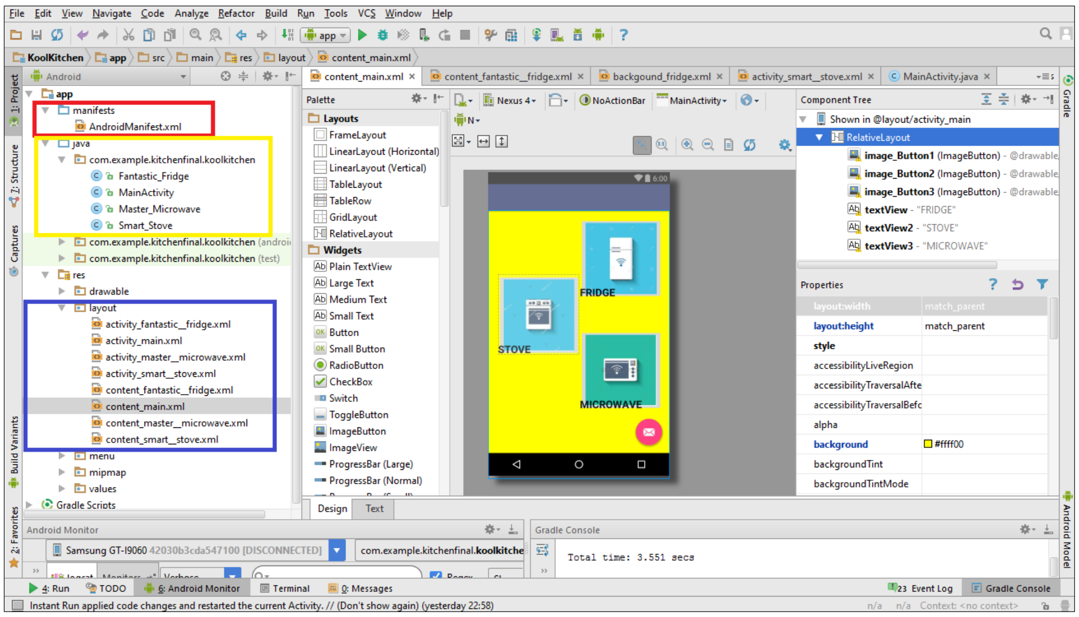
I have designed logo for Kool kitchen which we can see in the Figure 8, below.

Figure 8 Icon for Kool Kitchen

3.1 Application Development

As to start some important things to keep in mind is the android fest is the main manager with xml files. It will have bunch of activities which will manages our application. As soon app start phone doesn’t know what to do. Phone looks for manifest .Phone will look all of the activities and look for properties called launcher now launcher is app starting point. Phone will find out and it will hop to main activity .All other activities which are not main activity which have default settings not launcher setting. [7]

Three very important files which when I was making android application should be taken care of is manifest file, java file and layout. Without working on these files and without knowledge about these files no application can work .Let see the files in the Android Studio. Refer to Figure 9, for more clear understanding of the concept.[8]



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
KoolKitchen > app > src > main > res > layout > content_main.xml
app
manifests
AndroidManifest.xml
java
com.example.kitchenfinal.koolkitchen
Fantastic_Fridge
MainActivity
Master_Microwave
Smart_Stove
res
drawable
layout
activity_fantastic_fridge.xml
activity_main.xml
activity_master_microwave.xml
activity_smart_stove.xml
content_fantastic_fridge.xml
content_main.xml
content_master_microwave.xml
content_smart_stove.xml
menu
mipmap
values
Gradle Scripts
FRIDGE
STOVE
MICROWAVE
Component Tree
RelativeLayout
image_Button1 (ImageButton)
image_Button2 (ImageButton)
image_Button3 (ImageButton)
textView - "FRIDGE"
textView2 - "STOVE"
textView3 - "MICROWAVE"
Properties
layout:height match_parent
background #ffff00
Design Text
Android Monitor
Samsung GT-I9060 42030b3cda547100 [DISCONNECTED]
Gradle Console

```
Total time: 3.551 secs
```

Instant Run applied code changes and restarted the current Activity. // (Don't show again) (yesterday 22:58)

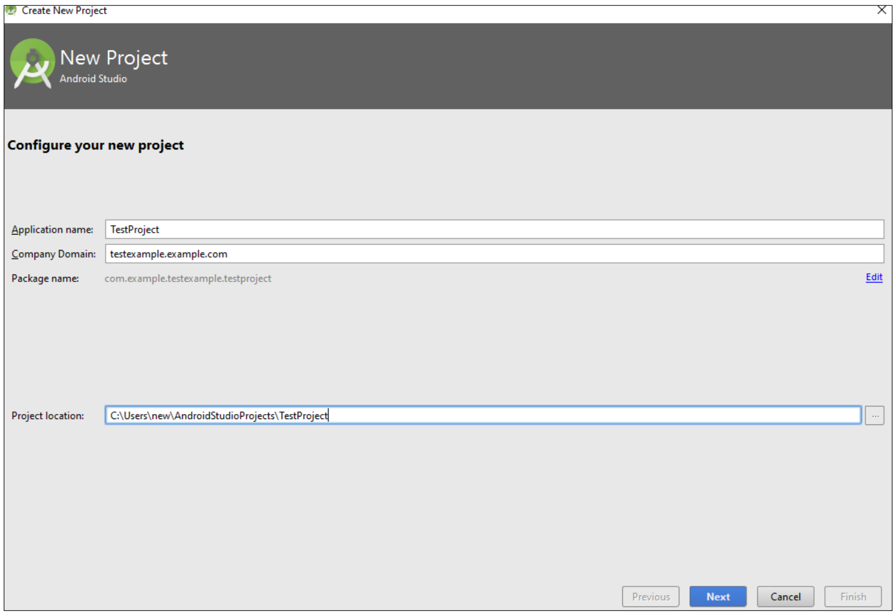
Figure 9 Three Important Files of Android App

All these files shown in Figure 9 ,is not the initial stage of the development of application it is later stages when application is almost ready to work with all data feed in .For that lets first see how to open an simple activity or main screen to work on in android studio.

3.1.1 Working with Main Activity

The first screen is called MainActivity, however user can change its name. Making the project for Kool kitchen so for that open android studio add name to app n company domain name which is a virtual package installation and click next choose android version .For that open android studio open New project .The window will appear which is shown below in Figure 10.Once I open new project all components and SDK files will load up in the android Studio to make it work smoothly. For that in the SDK Manger in the tool bar user can load up all the files. [9]

Here user can give name to his application. Select a unique package and location where he wants to store the project wish to keep it in default location as it is easy to find from there. So I named my application name as Kool Kitchen. Hence this will be name appear on the screen when I will install my application in phone or tablets. So for refer figure 10.Figure 11, actually shows the logo and the name which I have given to my application.



Configure your new project

Application name: TestProject
Company Domain: testexample.example.com
Package name: com.example.testexample.testproject

Project location: C:\Users\new\AndroidStudioProjects\TestProject

Figure 10 New Project Android

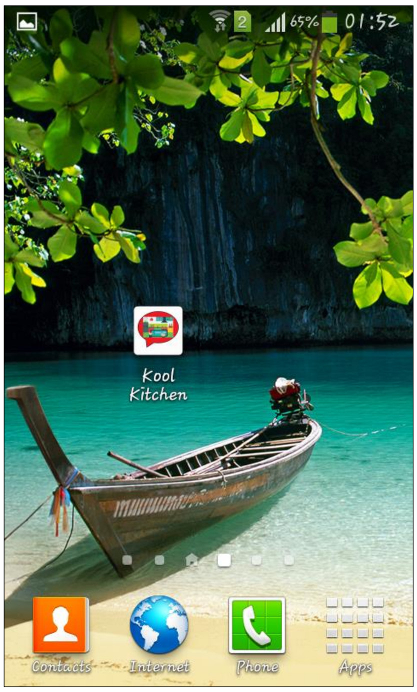
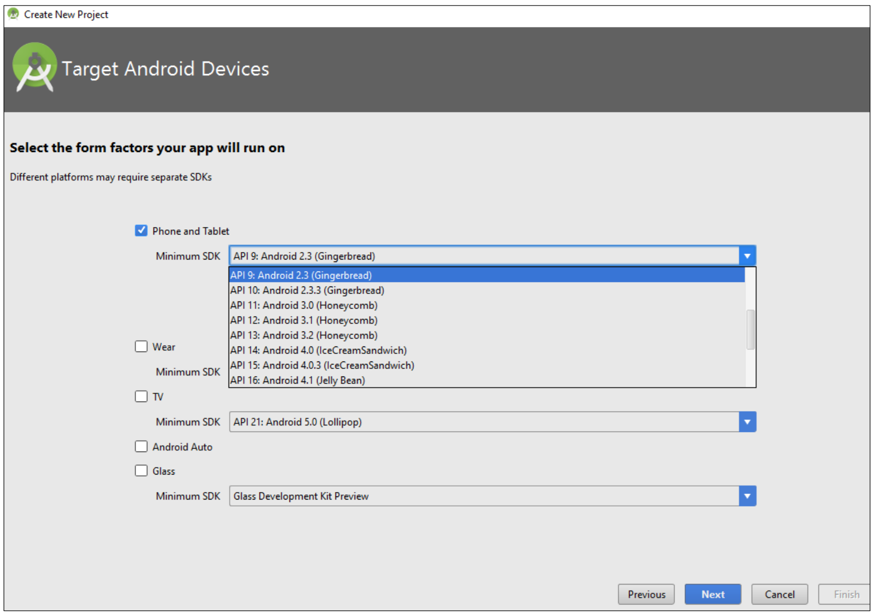


Figure 11 App Name and Logo

Second window I see when I click on next is given in figure 12 and from that I can select what version of android I want my use to use so for making Kool Kitchen I have used Gingerbread version of android which not very basic and not very advanced and when checked 100% of phones and tablets can manage to work on this version according to current status.



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 9: Android 2.3 (Gingerbread)

- API 9: Android 2.3 (Gingerbread)
- API 10: Android 2.3.3 (Gingerbread)
- API 11: Android 3.0 (Honeycomb)
- API 12: Android 3.1 (Honeycomb)
- API 13: Android 3.2 (Honeycomb)
- API 14: Android 4.0 (IceCreamSandwich)
- API 15: Android 4.0.3 (IceCreamSandwich)
- API 16: Android 4.1 (Jelly Bean)

☐ Wear

Minimum SDK

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass

Minimum SDK: Glass Development Kit Preview

Figure 12 Selecting Android Version

Important note in the above picture is that I am for now just working on phones and tablets so wear, TV and other tabs in the window I left unattended. When I clicked next in the to the window in Figure 12, I got another window which is shown in Figure 13 .In Figure 10 it is made possible for user to select an activity preferred Basic activity though out my project.

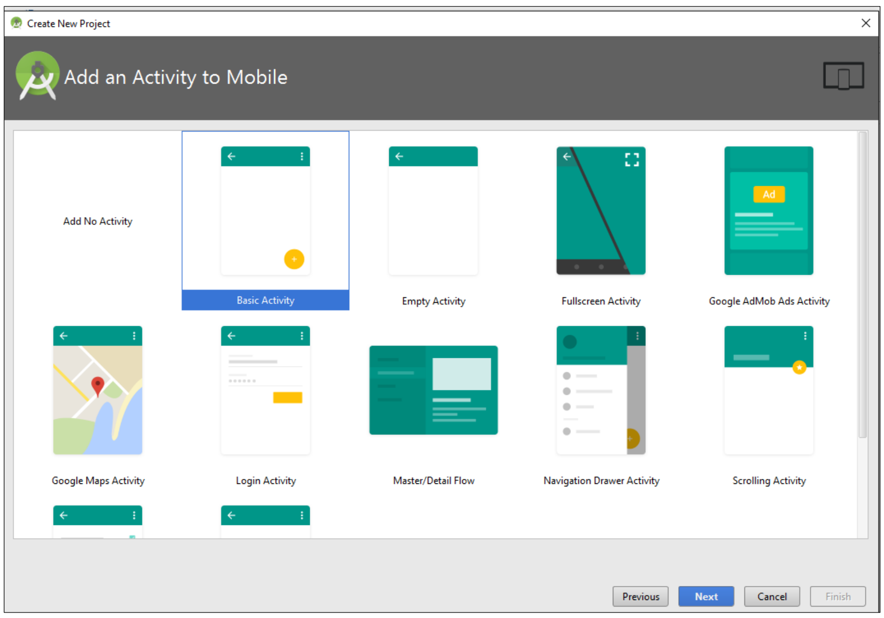
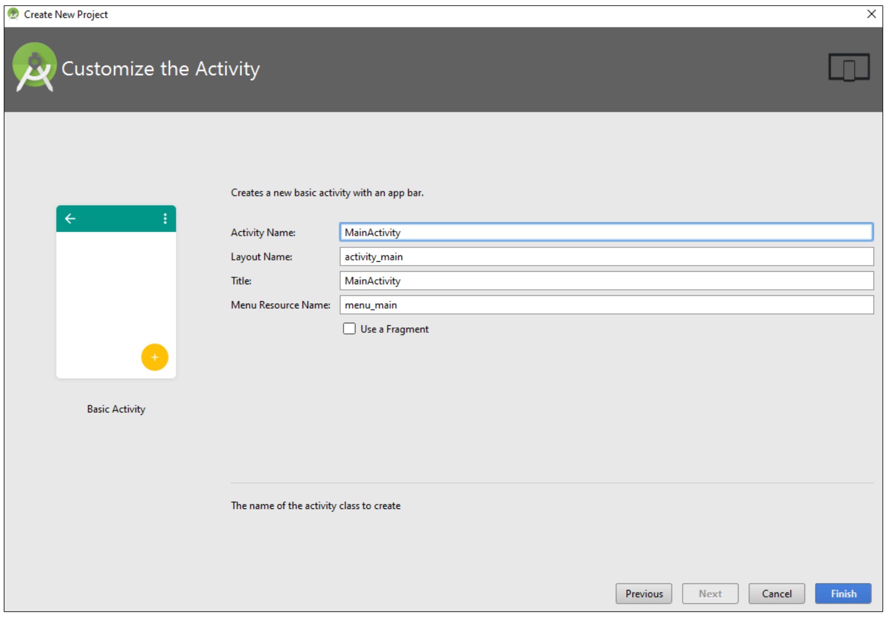


Figure 13 Adding Activity

Now in figure 13, I will show you how to actually customise name of activities and go to the platform to work on java and xml files and actually make activity do something. And after choosing Activity name, Title and layout name user can click on finish to let it build for few minutes in Android Studio.


Create New Project

Customize the Activity

Creates a new basic activity with an app bar.

Activity Name: MainActivity
Layout Name: activity_main
Title: MainActivity
Menu Resource Name: menu_main
☐ Use a Fragment

Basic Activity

The name of the activity class to create

Figure 14 Customize Activity

3.2 Making Kool Kitchen

The above all example used a different package name and application name as Test Project ,the same work is done for making Kool Kitchen and I get into the Android Studio. In figure 15, it is shown that how the first ever page of my application looks like so to make it little fancy and make it look good I have added background colour to my application. In this case I chose yellow. So for giving colour hexadecimal values of colour should be given as shown in the figure 15.And this file has Java codes. In layout folder two files content_main.xml and activity_main.xml and at top manifest which has data from all the activities which I am going to add.

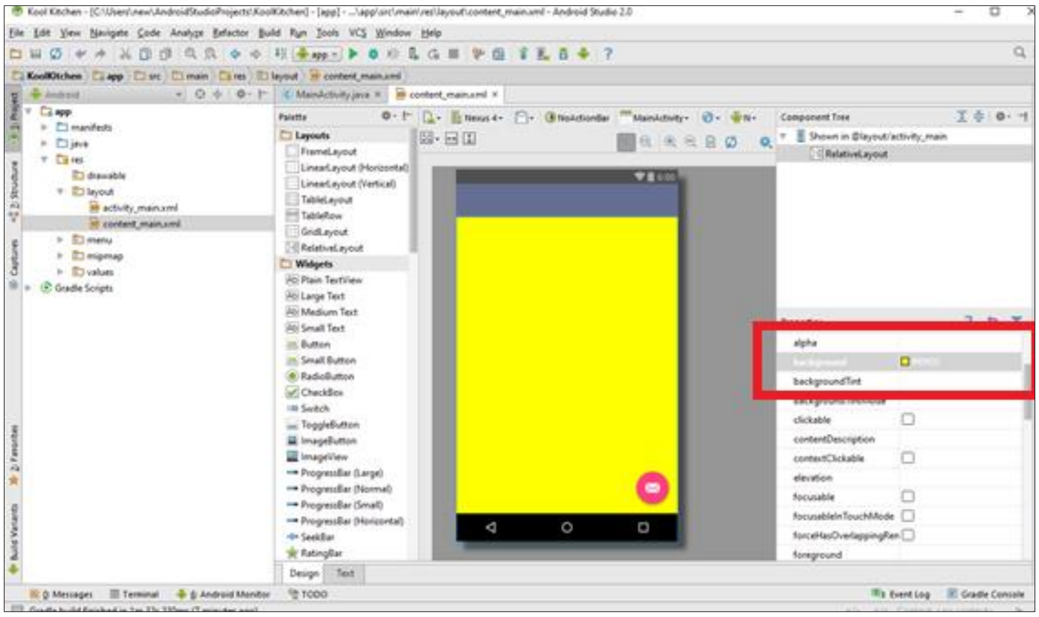


Figure 15 Background for APP

3.2.1 Giving Logo to an APP

Before I have done something with the application I have given logo to my application which I have shown in Figure 11.

Go to resource file >> right click on new>> image asset >> select image>>check folder >>pick configuration and then click on finish have used XHDI configuration to make my application look better in the phone.

After doing we can check our application have checked my application so far everything is working well.

3.2.2 Adding Image Buttons to APP

First of all images which I need to used should be uploaded to Drawable folder with a suitable size given to it. In this project I have uploaded three images that is fantastic_fridge, smart_stove and master_microwave. All in lower case as Drawable only allows files with lower cases. After I have all the files in Drawable I need to create three image buttons.

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/image_Button1"
    android:src="@drawable/fridge_click"
    android:layout_alignParentTop="true"
    android:layout_toRightOf="@+id/image_Button2"
    android:layout_toEndOf="@+id/image_Button2"
    android:onClick="callFridge" />
```

In the text design of the content_main.xml file I need to write this code. Here each line written in code has its meaning I will explain each. First of all I need image button so open <ImageButton/>

Height and width of image button could be user defined so

```
android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

Every Button should have a unique id from which it could be called in main file of java and manifest.I have given image_Button 1 as id to my first image switch.

```
android:id="@+id/image_Button1"
```

Source of my image button as explained above is from Drawable so

```
android:src="@drawable/fridge_click"
```

The below is nothing just the alignment position of the switch which says it is on the top ,right of image_Button2 and when it ends image Button 2 starts from there.

```
android:layout_alignParentTop="true"
    android:layout_toRightOf="@+id/image_Button2"
    android:layout_toEndOf="@+id/image_Button2"
```

Second step is to give a name to this image Button just to make application look little better. And hence for that again I need to write code in xml file. Note that in these cases we are working on two views Design view and Text view. Codes are written in Text view in xml and pictures are dragged and dropped and aligned in Design view.

```
<TextView
    android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="FRIDGE"
android:id="@+id/textView"
android:textStyle="bold"
android:layout_alignBottom="@+id/image_Button1"
android:layout_alignLeft="@+id/textView3"
android:layout_alignStart="@+id/textView3" />
```

The above is almost same as the Image Button switch however instead of displaying message in the form of image I have used text .So this text will be seen on the image when user will see full interface.

android:text="FRIDGE"

Hence when codes for all three button will be added I can see below images in phone or in design view. Full code will be given in Appendix 1.

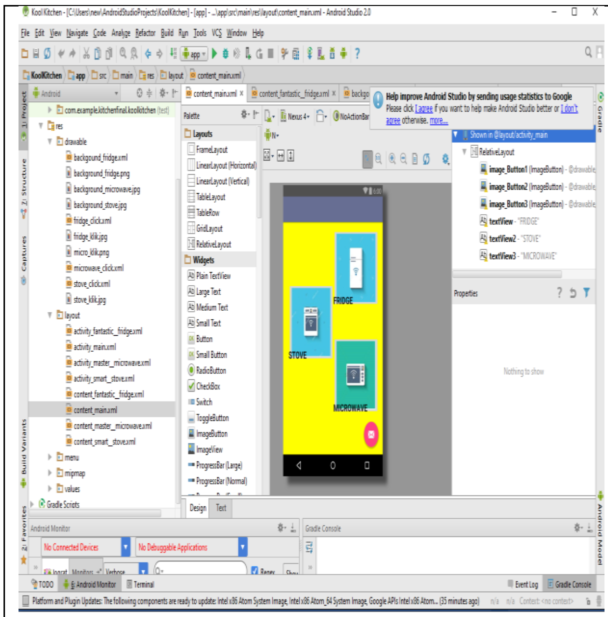
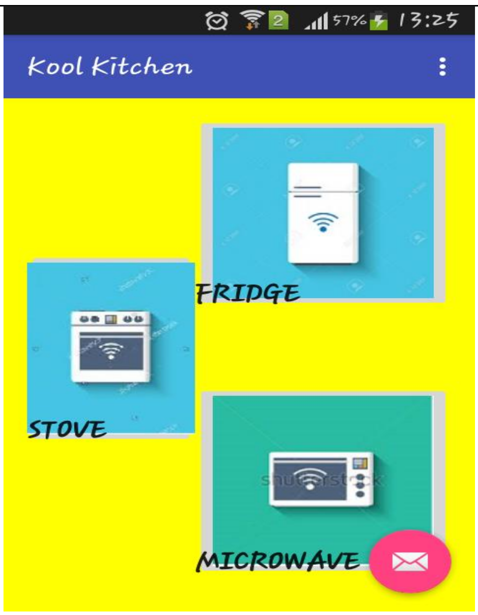
	 57% 13:25 Kool Kitchen FRIDGE STOVE MICROWAVE
DESIGN VIEW	APP IN PHONE

Figure 16 Adding Image Buttons

3.2.3 Adding Three Activities to App

Adding 3 activities to android means when user click on each button it should open a new activity or in simple language new page. For opening any new activity user need to create three new layout so it would be like App>>New>> Activity>>Basic activity.

This will add one java class and one xml file so user can work on it

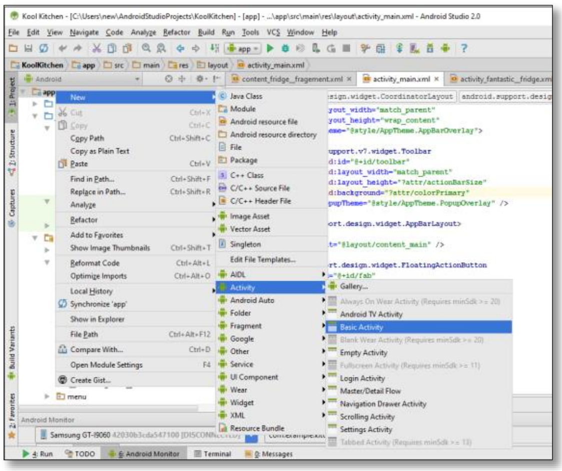


Figure 17 Creating New Activity

3.2.4 Opening Activities from Buttons

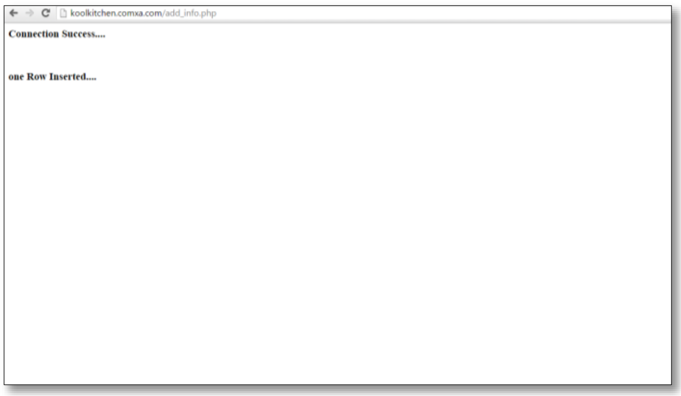
For that I have created on click methods for the above defined buttons that one can see in Appendix 1.Hence on click method is defined in xml files but they are called in java code .Refer Appendix 2 for that. And looking at 3 activities their java and xml codes are given in Appendix3, 4, 5.And xml files and other design I will be explaining in bit details with SQL programing and data bases.

3.3 Uploading Data from Android to Server (Databases)

Android actually comes with the database. For making a client and server connection I will use PHP language and first in the work I will explain client and server relationship which will help understand my work more clearly. As on the client side say PC which is connected to phone has Java and xml files and one the sever site I have .NET or PHP so how it works is .client send data to sever and sever process that data and send it back to client in the same language client is using.PHP creates a good layer of feedback between client and the server. Server is capable of handling much more hard tasks and delivers back to client in HTML. [10].Let’s check what we have there at the server site .it is a block which basically has 3 important filed in it .

Wamp server is the software which I installed for the making my own client.PHP my admin is user interface for my SQL data base. After downloading user can start the Wamp server and check status if everything is working alright and server is ready to connect to client

First we need to create a database on my SQL data base. For that in browser I typed local host It will show the Wamp server .It is home page of the Wamp server. Here one can create there data base so for that click on phpMyAdminl>> from this interface one can manage databases on mysql



```
Connection Success....


one Row Inserted....
```

Figure 18 Connection with localhost

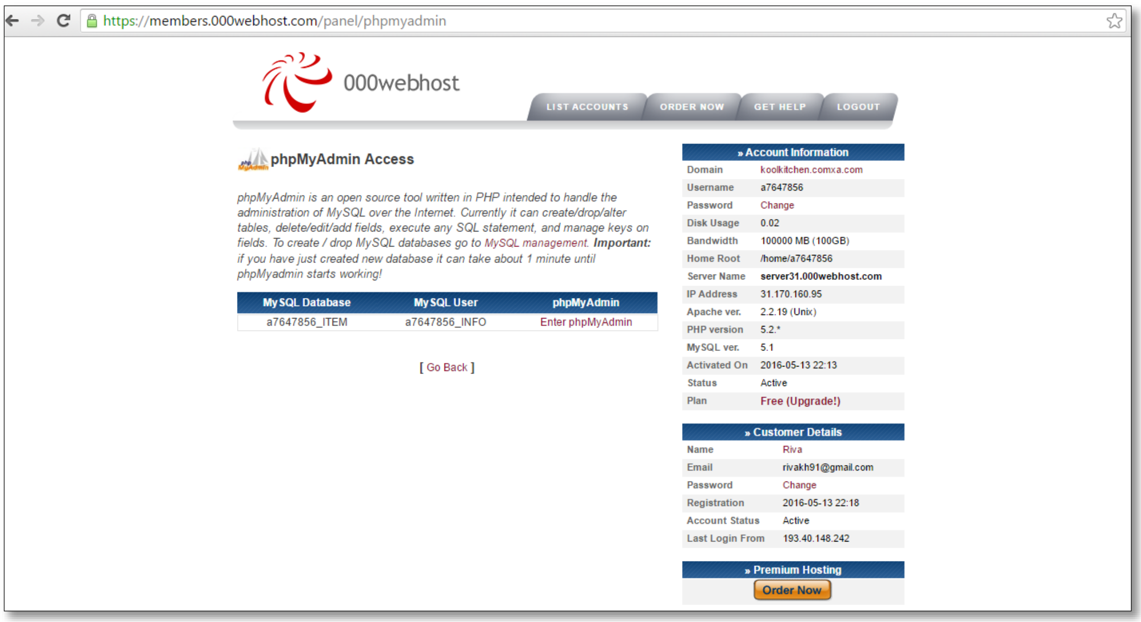
First I will connect my smart fridge with databases for that in this I will use JASON.JASON is most commonly used data interchange mechanism in android application.JSAON means JavaScript Object Notation. Format of JASON is easily read by humans. Most importantly JASON is a text format and supported by all programming languages. JASON is built on 2 structures that is object and array. Like all other programming languages. JASON is most of the commonly using parsing using with android application .Now as I want my fridge to take some data from databases and send some data to databases I am going to do that through JASON. [11]

Fantastic fridge as discussed above will have two main thing expiry dates of the item and weight of the item.

For this we need to connect to online database for that I need server space, some domain name and an android application to communicate with the database. First set up the server and get some domain name. Lot of website provide free PHP hosting and many of them provide free domain name also.

Domain created online is http://koolkitchen.comxa.com/

For creating data base online I have used 000webhost as it is easy and free. Data base is created online called as item and the table which is created inside the database is called as product_info


https://members.000webhost.com/panel/phpmyadmin

000webhost

LIST ACCOUNTS | ORDER NOW | GET HELP | LOGOUT

phpMyAdmin Access

phpMyAdmin is an open source tool written in PHP intended to handle the administration of MySQL over the Internet. Currently it can create/drop/alter tables, delete/edit/add fields, execute any SQL statement, and manage keys on fields. To create / drop MySQL databases go to MySQL management. ***Important:*** *if you have just created new database it can take about 1 minute until phpMyadmin starts working!*

MySQL Database	MySQL User	phpMyAdmin
a7647856_ITEM	a7647856_INFO	Enter phpMyAdmin

[Go Back]

» Account Information	
Domain	koolkitchen.comxa.com
Username	a7647856
Password	Change
Disk Usage	0.02
Bandwidth	100000 MB (100GB)
Home Root	/home/a7647856
Server Name	server31.000webhost.com
IP Address	31.170.160.95
Apache ver.	2.2.19 (Unix)
PHP version	5.2.*
MySQL ver.	5.1
Activated On	2016-05-13 22:13
Status	Active
Plan	Free (Upgrade!)

» Customer Details	
Name	Riva
Email	rivakh91@gmail.com
Password	Change
Registration	2016-05-13 22:18
Account Status	Active
Last Login From	193.40.148.242

» Premium Hosting

Order Now

Figure 19 SQL Database Online

At the left hand side of the Figure 19, all useful imformation is seen like Server name, Host ,IP Address etc .

$mysql_host="mysql9.000webhost.com";

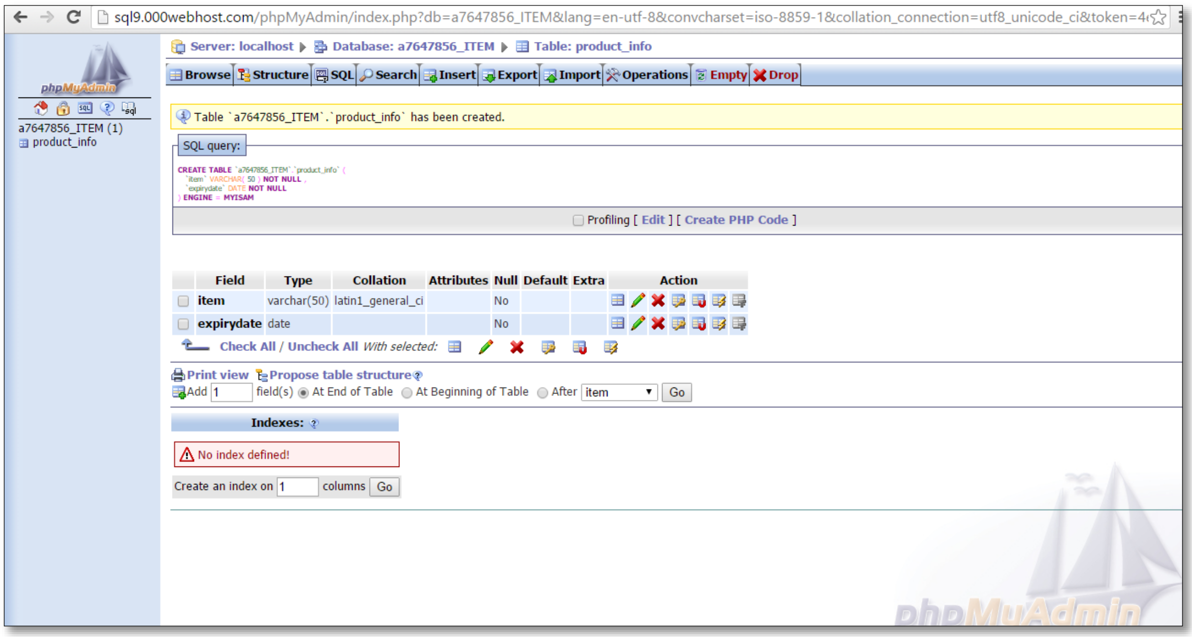
$mysql_database="a7647856_ITEM";

$mysql_user ="a7647856_INFO";

$mysql_password="jesus099"

Domaian name :http://koolkitchen.comxa.com/

These all imformations will be later used by android system to connect wirelessly.so in the table I have added two fields those two fiels are item and expiry date.As item is a variable character so the type is given as Varchar and expiry date is a Date so the type for the table is given ad date.


sql9.000webhost.com/phpMyAdmin/index.php?db=a7647856_ITEM&lang=en-utf-8&convcharset=iso-8859-1&collation_connection=utf8_unicode_ci&token=4

Server: localhost ▸ Database: a7647856_ITEM ▸ Table: product_info

Table `a7647856_ITEM`.`product_info` has been created.

SQL query:

```
CREATE TABLE `a7647856_ITEM`.`product_info` (
 `item` VARCHAR( 50 ) NOT NULL ,
 `expirydate` DATE NOT NULL
) ENGINE = MYISAM
```

Field	Type	Collation	Attributes	Null	Default	Extra
item	varchar(50)	latin1_general_ci		No		
expirydate	date			No		

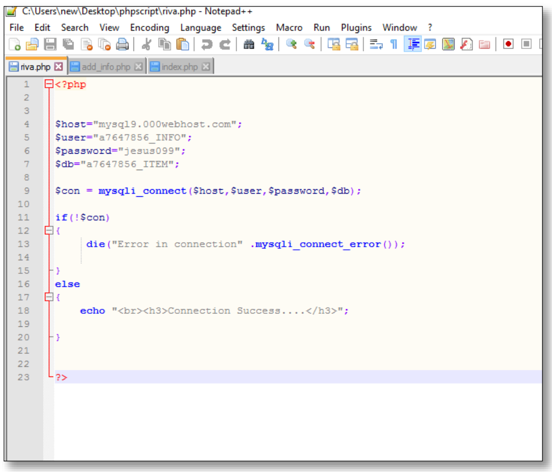
No index defined!

a7647856_ITEM (1)
product_info
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Profiling [Edit] [Create PHP Code]
Action
Check All / Uncheck All With selected:
Print view
Propose table structure
Add 1 field(s) At End of Table At Beginning of Table After item Go
Indexes:
Create an index on 1 columns Go

Figure 20 Table in Database

In the above image Figure 20, we see the table is empty so if I want to add data to sever manually first I need to add some PHP scripts as this server works on PHP scripts.

So just for making the connection I have used one script as riva.php which is given below.


C:\Users\new\Desktop\phpscript\riva.php - Notepad++

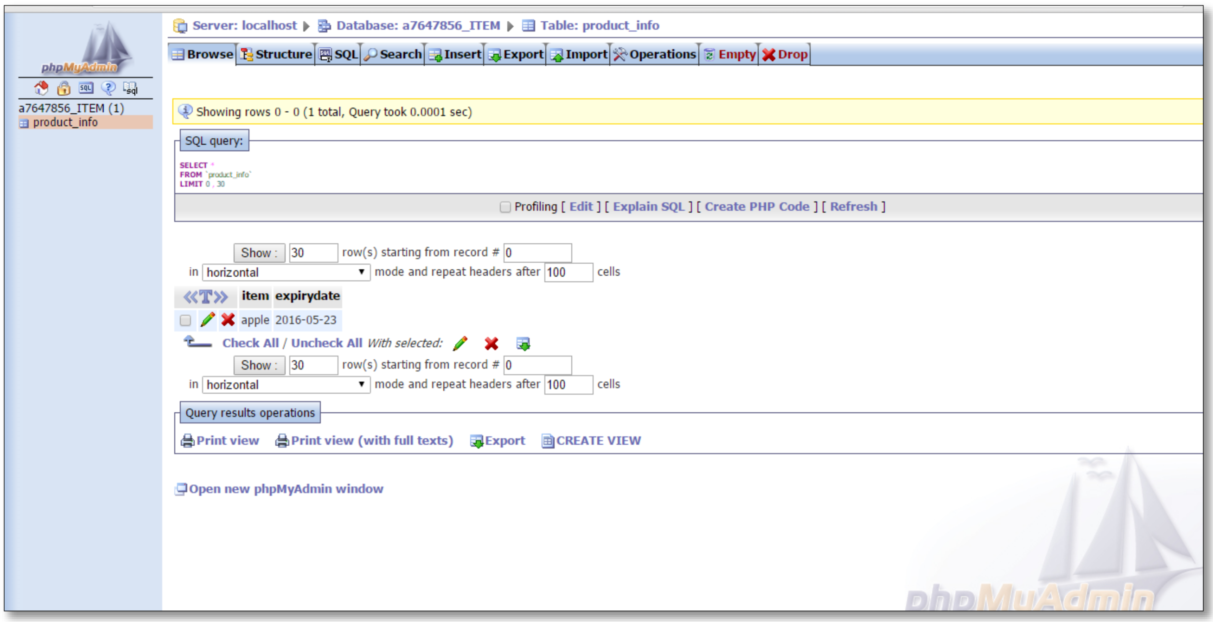
```
<?php


$host="mysql9.000webhost.com";
$user="a7647856_INFO";
$password="jesus099";
$db="a7647856_ITEM";

$con = mysqli_connect($host,$user,$password,$db);

if(!$con)
{
    die("Error in connection" .mysqli_connect_error());

}
else
{
    echo "<br><h3>Connection Success....</h3>";

}


?>
```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
riva.php
add_info.php
index.php

Figure 21 PHP Script (riva.php)

This PHP script should be added to online data bases .For adding scripit go to phpMyAdmin>>File manager>>upload >>New>>choose file >>select the check Button.

Similarly other files Add_info.php and index.php is added to server.These codes are given in Appendix 6 and 7 respectively. When all files are added to server I can check and add a table to my server manually. In which I have added item apple and of expiry is 23rd May, 2016.



Showing rows 0 - 0 (1 total, Query took 0.0001 sec)

SQL query:

```
SELECT *
FROM `product_info`
LIMIT 0 , 30
```

Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

item	expirydate
apple	2016-05-23

Check All / Uncheck All *With selected:*

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Figure 22 Manual Data Insertion

For connecting android with webhost I used java and xml codes in android. And added an activity which is called Fridge_Fragment .As this activity will be open from the fridge activity. For the background task in Fridge_Fragment one can refer to Appendix 8 for java codes. To explain it first see I worked on the activity I have that is Fantastic _Fridge in that activity I have added two buttons and one text field to show the network connectivity.so in the Figure 23, we can see the fantastic fridge and also the xml codes for the activity which is given in Appendix 9.

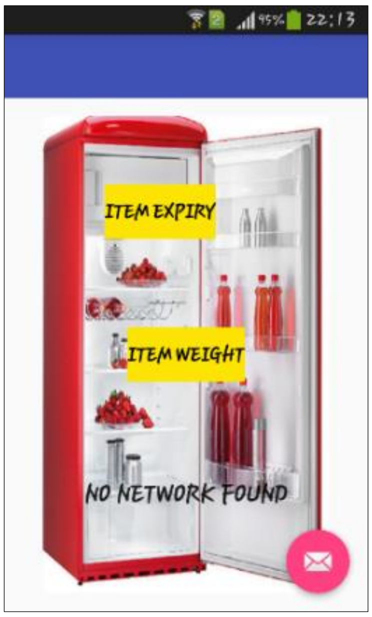


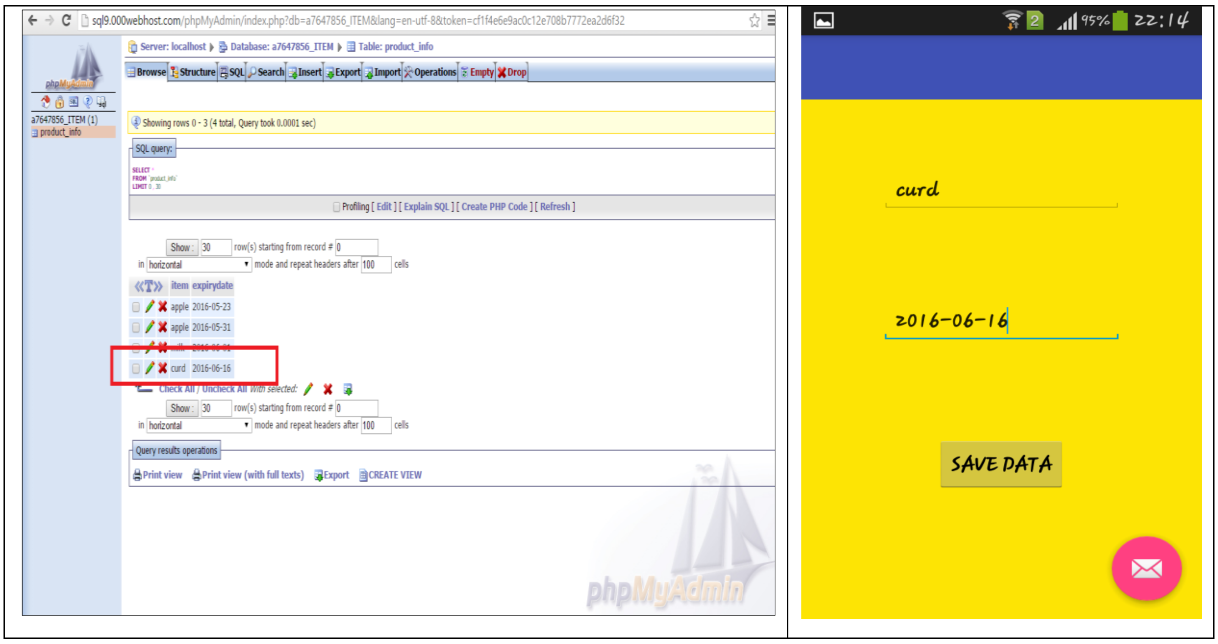
Figure 23 Fantastic_Fridge Activity

The important thing there to keep in mind is two buttons with their ids and onclick method added for the button which is ITEMEXPIRY Button.

When the user clicks on item expiry new activity is open that is Fridge_Fragament and there I have added item as curd and date of expiry as 2016-06-16.And saved that information in the android as the Internet permissions is on in the android so it will add this information to the SQL online.

```
<uses-permissionandroid:name="android.permission.INTERNET"/>

<uses-permissionandroid:name="android.permission.ACCESS_NETWORK_STATE"></uses-
permission>
```

The above code is the example how internet permission is given to android. Also in the Figure 24, we will see the Android application actually sending data to SQL online


sql9.000webhost.com/phpMyAdmin/index.php?db=a7647856_ITEM&lang=en-utf-8&token=cf1f4e6e9ac0c12e708b7772ea2d6f32

Server: localhost ▸ Database: a7647856_ITEM ▸ Table: product_info

Showing rows 0 - 3 (4 total, Query took 0.0001 sec)

```
SELECT *
FROM `product_info`
LIMIT 0 , 30
```

item	expirydate
apple	2016-05-23
apple	2016-05-31
milk	2016-06-01
curd	2016-06-16

curd

2016-06-16

Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
a7647856_ITEM (1)
product_info
Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]
Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Check All / Uncheck All With selected:
Query results operations
Print view
Print view (with full texts)
Export
CREATE VIEW
95%
22:14
SAVE DATA

Figure 24 Adding data from android to online SQL

4 MODELLED SYSTEM MADE FOR THE HOME AUTOMATION SYSTEM

For modelling the home system first thing I need to do is to make connection between Arduino and sensor. In the Modelled home automation system which I have made a display of it below we have a home PC in which data processing is taking place constantly as user cannot use phone all the time.

We need a home system to process and save data .Anther change I made here is all the devices are connected to microcontrollers and are in hand shake with each other and after that it is connected to the mobile phones. It will be used for to optimize the energy. So the full idea of the Automatic Kitchen will come into picture. So the main idea here is to include a central hub as Pc where microcontroller is attached to it Here data will be saved and processed as user cannot use phone all the time system needs a central Hub to collect save and process data in a way user get the error message if the functioning is not proper for any of the system. Hence the work is divided into 3 main parts that is making android app. Application should download the data from the server. Connecting sensors to microcontrollers and get data in real time and uploading data to the server.

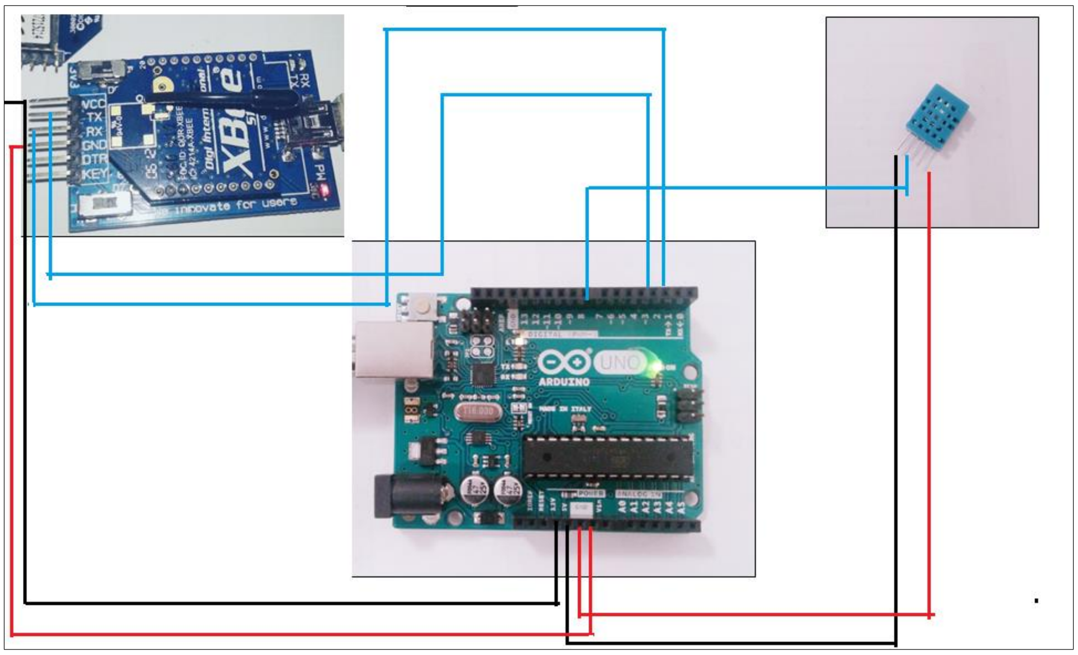


Figure 25 Model Home System required connections

4.1 Arduino to Sensor Connection

The sensor I am using here is temperature and humidity sensor which is DHT11.It has two main parts a capacitive humidity sensor and a thermistor. It is cheaper and easily available also one can use similar sensor, DHT22 which is bit more accurate with results. Practically DHT11 cannot work for temperature ranges more than 50°C but for my prototype I have used this sensor. For higher temperature ranges I prefer DHT22

It is a digital temperature and humidity sensor. So for making the connections with Arduino Uno and sensor we need to know all the pins of both .And connect those pins then connect Arduino to PC. And then a code must be written in Arduino to get the required output.DHT11 has four pins. If I start explaining it from right to left it is shown in the Figure 26.

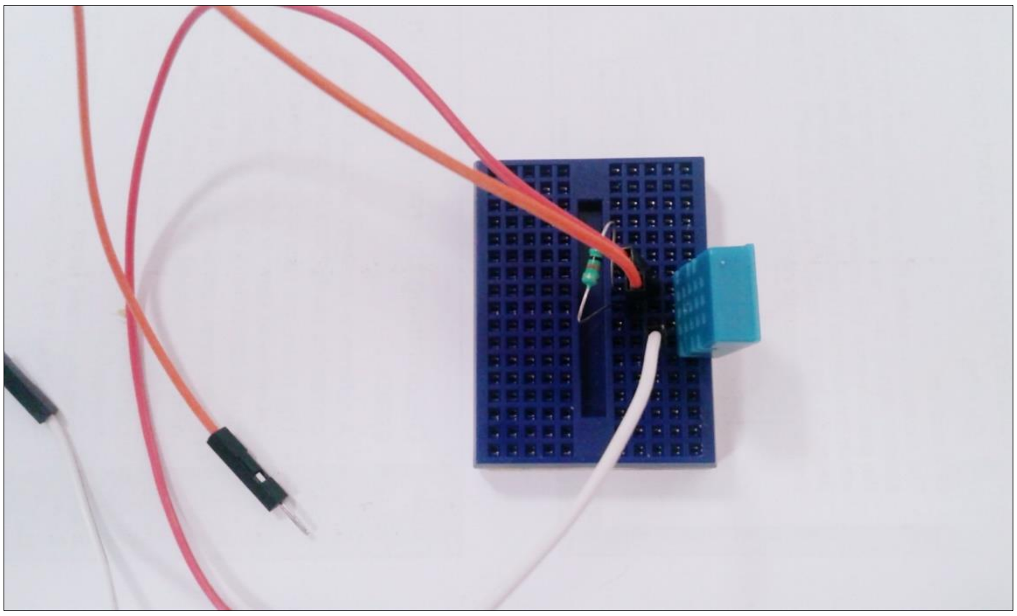


Figure 26 DHT11 Sensor Connections

Technical details of sensor [12]

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy

- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

The final connection would look like the image below, shown in Figure 27.

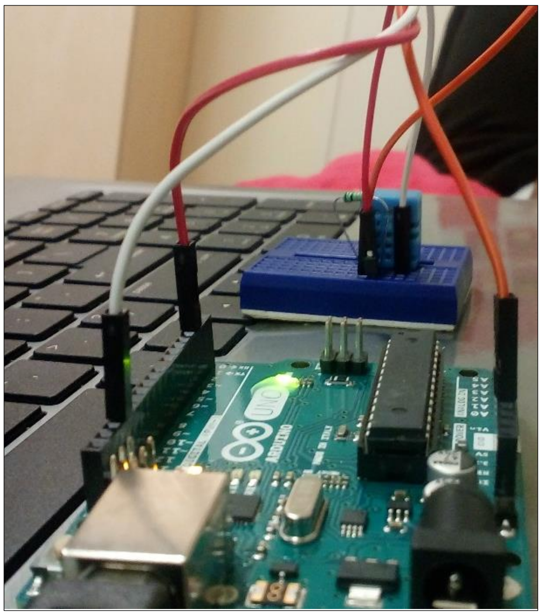


Figure 27 Connection between Arduino and Sensor

As I am going to work with Arduino so I should know how to work with it so with basic I started and explaining it in here. So Arduino hardware in that I will take look and give a pretty general view. So I have an Arduino made by Arduino .It is open source that means all the design files for the board is made public.so anyone can go and make their own clones of Arduino so our Arduino has digital pins header that is a plastic lines with bunch of holes in it and numbers written next to it and number are 0 up to 13 and a pins are Ground. Those give access to chip which is below there. The pins can be used as input or output for example they can apply 5 volts voltage .If we look on the board closely we will find transmitter and receiver LEDS so when we load the data these pins will work efficiently. There are 6 Analog pins headers in Arduino board marked 0 to 6.Another header talk at here is 3.3V which stands for 3.3 volts and 5V which stands for 5 volts .Another important button is reset button when I push that button the Arduino is going to start from the beginning of the program. It will not delete the code which we write but it will reboot. Another way to reset the board is applying 0 volts to reset button which is next to 3.3V.Arduino board could be power supplied by external battery or by using USB cable however whenever board is on we will see LED light ON .

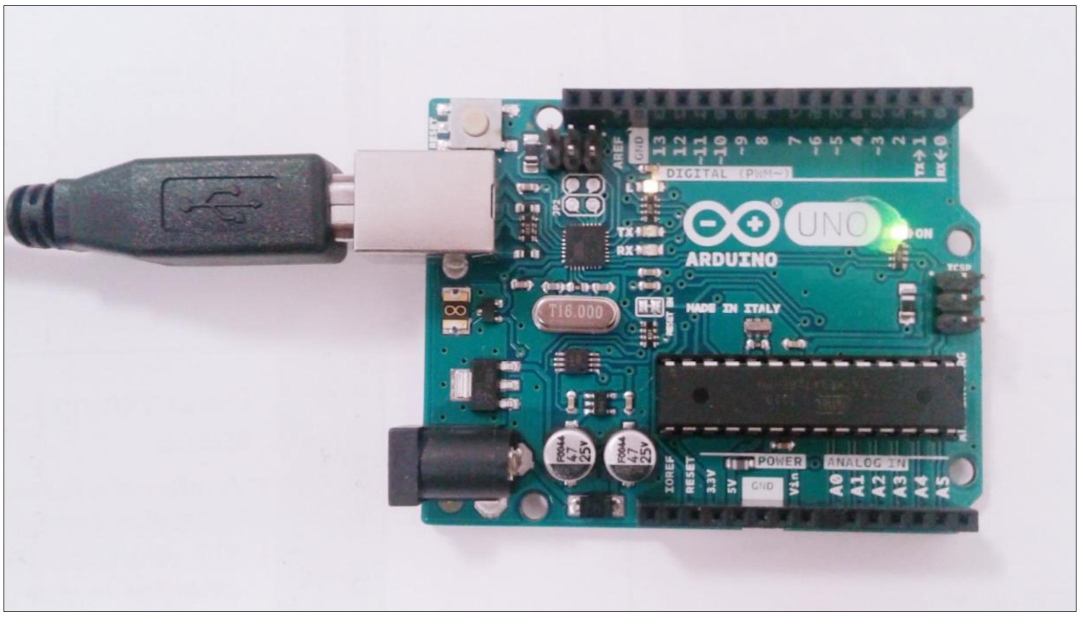
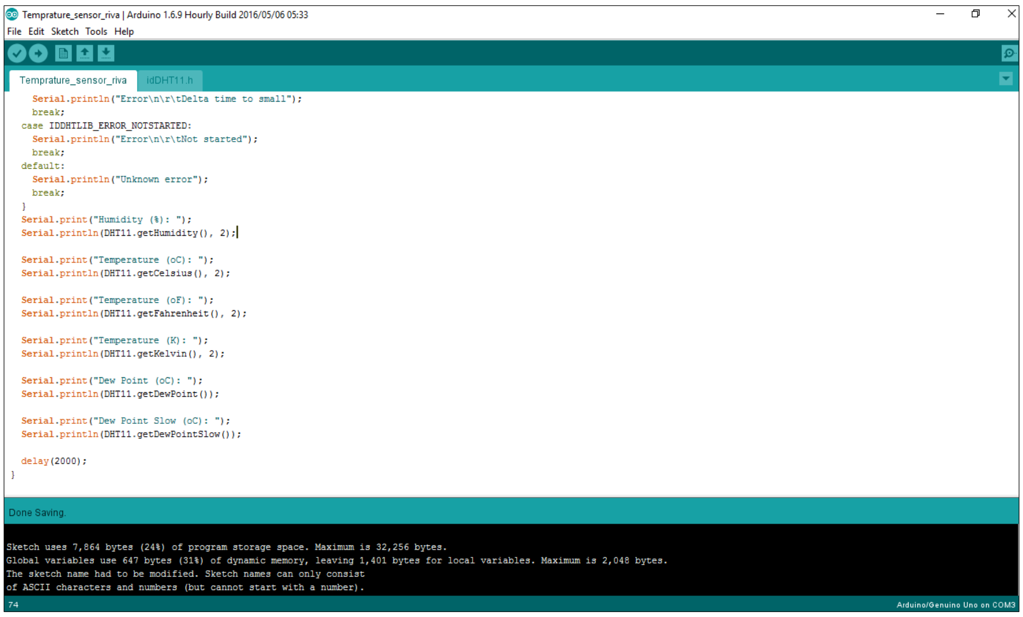


Figure 28 Arduino Used

One of the best thing with Arduino is how easy it is to get started. The software is completely free and design is completely free. The program called integrated development environment and (IDE) for short .For installing Arduino in computer >> Go to Browser >> Arduino Home page >> arduino.cc (one should see this on the top) >> Go to downloads >> select operating system.

It will take a minute to download .One can save this Arduino this file applications.so it will easy for me use anytime I want to use it next .So it will default saved my file in

C:\Users\new\Documents\Arduino.When I opened my IDE for the first time that is what I got in Figure 29 below.


Temprature_sensor_riva | Arduino 1.6.9 Hourly Build 2016/05/06 05:33
File Edit Sketch Tools Help
Temprature_sensor_riva
idDHT11.h

```
    Serial.println("Error\n\r\tDelta time to small");
    break;
  case IDDHTLIB_ERROR_NOTSTARTED:
    Serial.println("Error\n\r\tNot started");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
  Serial.print("Humidity (%): ");
  Serial.println(DHT11.getHumidity(), 2);

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

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

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

  Serial.print("Dew Point (oC): ");
  Serial.println(DHT11.getDewPoint());

  Serial.print("Dew Point Slow (oC): ");
  Serial.println(DHT11.getDewPointSlow());

  delay(2000);
}
```

```
Done Saving.

Sketch uses 7,864 bytes (24%) of program storage space. Maximum is 32,256 bytes.
Global variables use 647 bytes (31%) of dynamic memory, leaving 1,401 bytes for local variables. Maximum is 2,048 bytes.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers (but cannot start with a number).
74                                                    Arduino/Genuino Uno on COM3
```

Figure 29 Arduino Sketch

As discussed in section for making the connection between Arduino and sensor all pins should be known properly have used DHT11 with same method one can use DHT22.So four pins coming out of my sensor and I have connected first pin to the left of the sensor to 5V of Arduino. Second step is to attach a 10 K Ω resistor between first pin and second pin from the left. The trick here is one side of the sensor is with hole and one side without holes or checks so the side with holes should face the user. Following the second pin with the resistor and connect it to pin number 2 on Arduino. The third pin is left alone and fourth pin is connected to Ground. Full code for the Arduino will be given in Appendix 10.

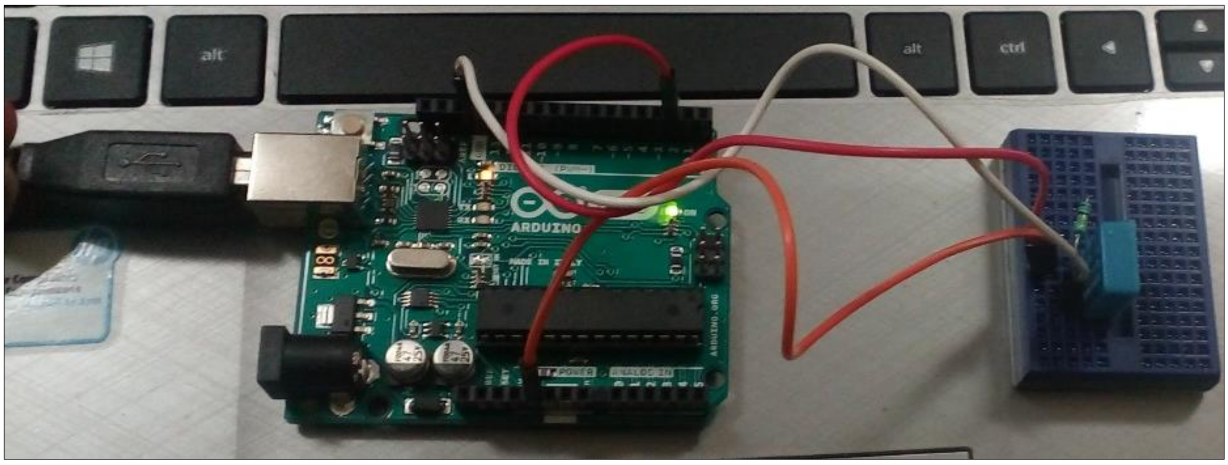
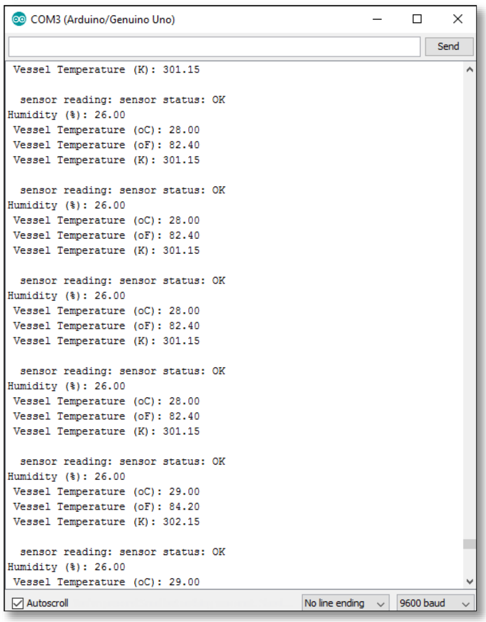


Figure 30 Arduino And Sensor Connection

The output from the sensor can be shown below in Figure 31.It is so evident that temperature is increasing and decreasing to get the apt output I have placed ice pack near the sensor so temperature decreased to 20°C and high as 30°C .I have made one video of updating data and will show it in presentation.



```
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 29.00
 Vessel Temperature (oF): 84.20
 Vessel Temperature (K): 302.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 29.00
```

Figure 31 Output from Sensor

4.2 Arduino to PC via 2 XBees

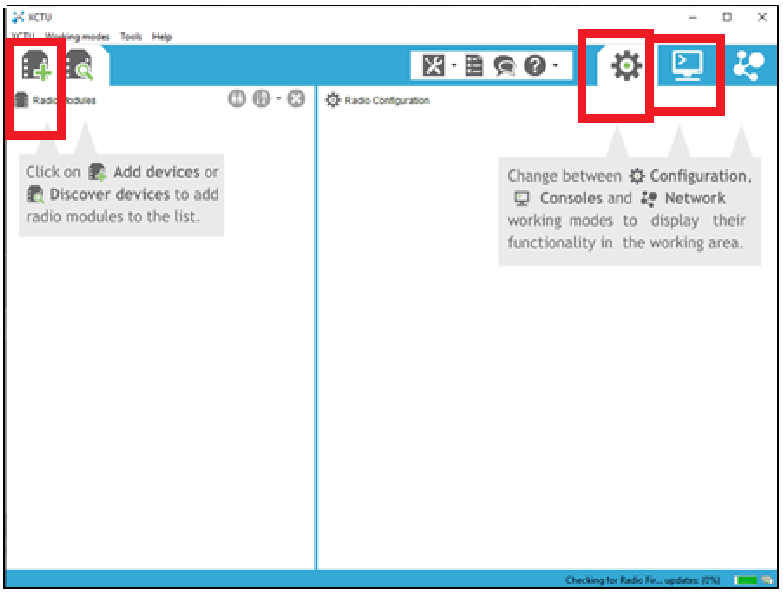
Now when I am getting data from sensor to Arduino next logical step is to work on sending this data wirelessly through XBees

Xbee is a wireless module as mentioned above in section .It uses low power applications of radio frequency .one can use it as transmitter or receiver both. The communication used to send and receive data is serial. Here in this section I will talking more about how technology works. So why I am using zig bee for working as it has higher range as compared to other wireless modules. Before I write the procedure about Xbee there are few important things I need to add about Xbee .So Xbee is a wireless microcontroller made by digi.it utilizes 802.14.5 protocol. It is used to communicate between two wireless radios. It consist of 20 pins and a little antenna. It has 11 digital I/o pins and 4 Analog pins am using Xbee S1 which requires 3.3 V and it has indoor range of 40 meters and line of set range around 120 meters. It uses 2.4 Gigahertz frequency. Important feature about Xbee is it can interfaces with Arduino very easily

It is interesting to note how it can be used in the network so that one should decide one coordinator Xbee only one Xbee can be used as the coordinator Xbee other Xbees can be used as routers or end pints that defends hoe network is defined.

There can be two modes of AT and API mode in AT mode communication goes through the Xbee.In API mode you can connect to Xbee to receive and transmit data through Xbee itself have used USB cable to attach my Xbee to computer .It have not used FTDI cable .one can also use FDTI cables .

So for configuring two XBees I am using a software provided by Digi X-CTU. This softaware is easily available on internet and it is compatible both with windows and Mac.To configure and test serial communication taking part in two Xbees one can use same computer or different computer am using my computer to make connection at first .so when the user is looking for ata X-CTU softaware for the first time it looks like the Figure 32.Here I have marked three blocks which I have used to test two XBees


XCTU
XCTU Working modes Tools Help
Radio Modules
Radio Configuration
Checking for Radio Fir... updates (0%)

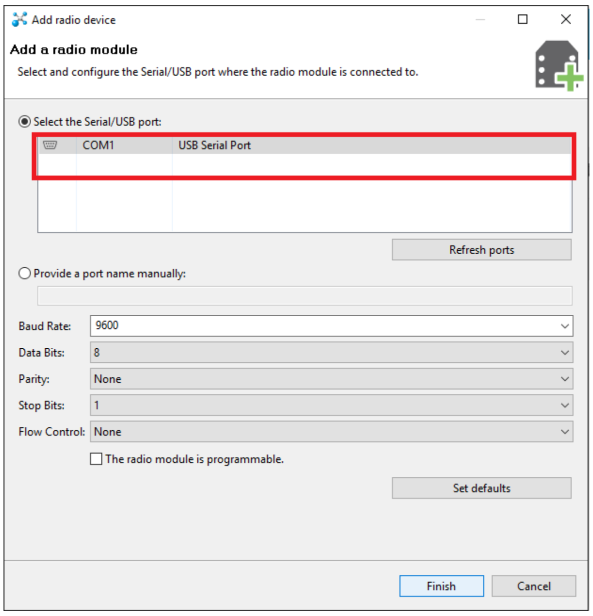
Click on Add devices or Discover devices to add radio modules to the list.

Change between Configuration, Consoles and Network working modes to display their functionality in the working area.

Figure 32 X-CTU software

The symbol on the left hand side is for add devices called as: Add device icon by clicking on it I got Xbees connected to my computer so that I can configure it [13].

I have connected one XBee for now to my computer and click on Add device icon it shows the dialogue box given in Figure 33, where Baud rate and parity can be checked. Also I know that now it is connected to COM PORT 1.after checking all properties I have clicked on finished so that this device will be added to X-CTU.


Add radio device

Add a radio module

Select and configure the Serial/USB port where the radio module is connected to.

(●) Select the Serial/USB port:

COM1 USB Serial Port

( ) Provide a port name manually:

Baud Rate: 9600

Data Bits: 8

Parity: None

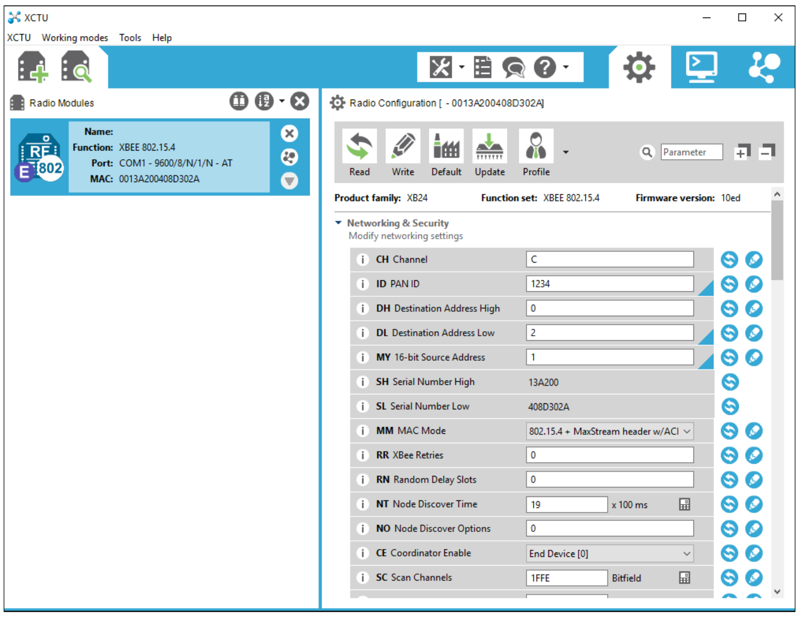
Stop Bits: 1

Flow Control: None

☐ The radio module is programmable.

Figure 33 Adding Device to X-CTU

Now click on the newly added device on the left hand side when I have clicked it start to read information and took me to next dialogue box which is shown in Figure 34.In the parameters device I have checked PAN Ids it is important to make a note if I want two Xbees to communicate with each other I need same PAN Ids.So I have given PAN id as 1234 similarly when configuring the next Xbee I have to make sure it is using same PAN Id


XCTU

XCTU Working modes Tools Help

Radio Modules

Name:
Function: XBEE 802.15.4
Port: COM1 - 9600/8/N/1/N - AT
MAC: 0013A200408D302A

Radio Configuration [- 0013A200408D302A]

Read Write Default Update Profile

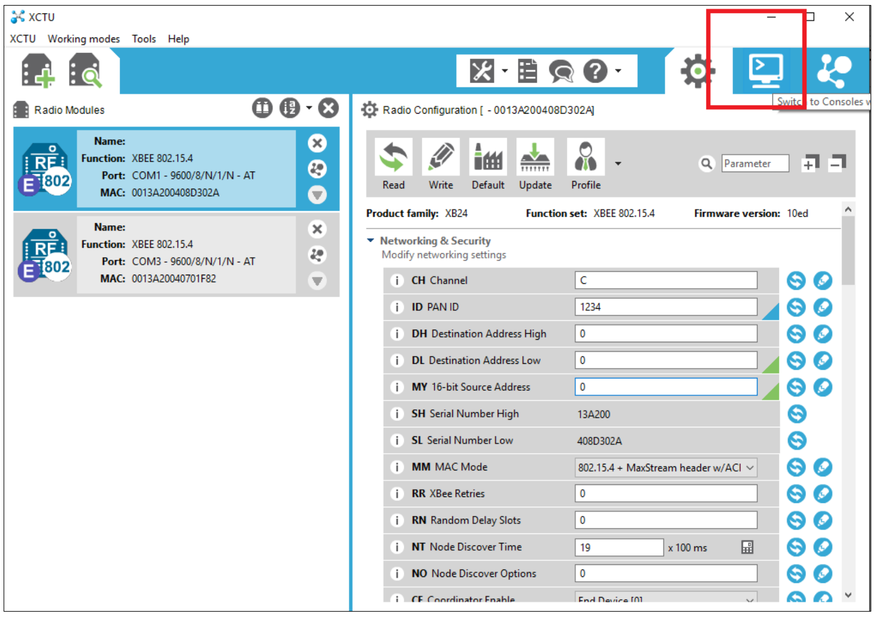
Product family: XB24 Function set: XBEE 802.15.4 Firmware version: 10ed

Networking & Security
Modify networking settings

CH Channel: C
ID PAN ID: 1234
DH Destination Address High: 0
DL Destination Address Low: 2
MY 16-bit Source Address: 1
SH Serial Number High: 13A200
SL Serial Number Low: 408D302A
MM MAC Mode: 802.15.4 + MaxStream header w/ACI
RR XBee Retries: 0
RN Random Delay Slots: 0
NT Node Discover Time: 19 x 100 ms
NO Node Discover Options: 0
CE Coordinator Enable: End Device [0]
SC Scan Channels: 1FFE Bitfield

Figure 34 Parameters of added device

And for adding second device one can use any other computer in which X-CTU is installed or use the same computer and any other port. Repeating all the above step would help user to configure second XBee.As soon a s plug in next Xbee and click on Add devices option it will start to read new device to computer and also give it a COM PORT and in this case COM 3 is given to it .Setting PAN ids enable me to make a communication between two Xbees .So to send data to and froth click on the symbol highlighted in Figure 35.


XCTU

XCTU Working modes Tools Help

Switc to Consoles

Radio Modules

Name:
Function: XBEE 802.15.4
Port: COM1 - 9600/8/N/1/N - AT
MAC: 0013A200408D302A

Name:
Function: XBEE 802.15.4
Port: COM3 - 9600/8/N/1/N - AT
MAC: 0013A20040701F82

Radio Configuration [- 0013A200408D302A]

Read Write Default Update Profile

Product family: XB24 Function set: XBEE 802.15.4 Firmware version: 10ed

Networking & Security
Modify networking settings

CH Channel: C
ID PAN ID: 1234
DH Destination Address High: 0
DL Destination Address Low: 0
MY 16-bit Source Address: 0
SH Serial Number High: 13A200
SL Serial Number Low: 408D302A
MM MAC Mode: 802.15.4 + MaxStream header w/ACI
RR XBee Retries: 0
RN Random Delay Slots: 0
NT Node Discover Time: 19 x 100 ms
NO Node Discover Options: 0
CE Coordinator Enable: End Device [0]

Figure 35 Making connection

For making the connections one I clicked on the connection switch named as Open in green.

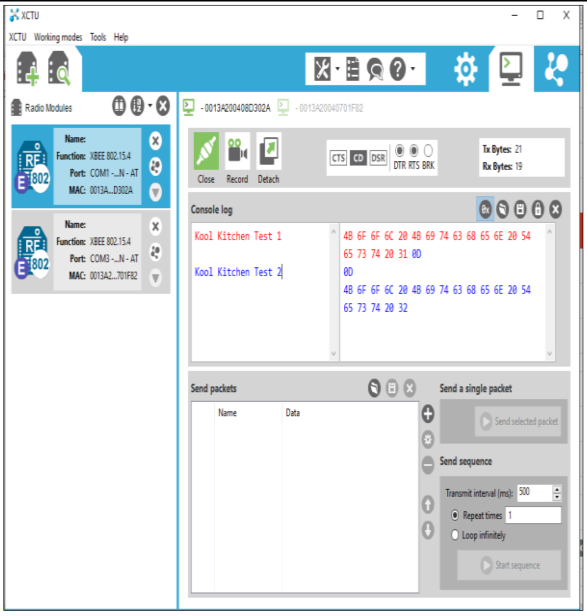
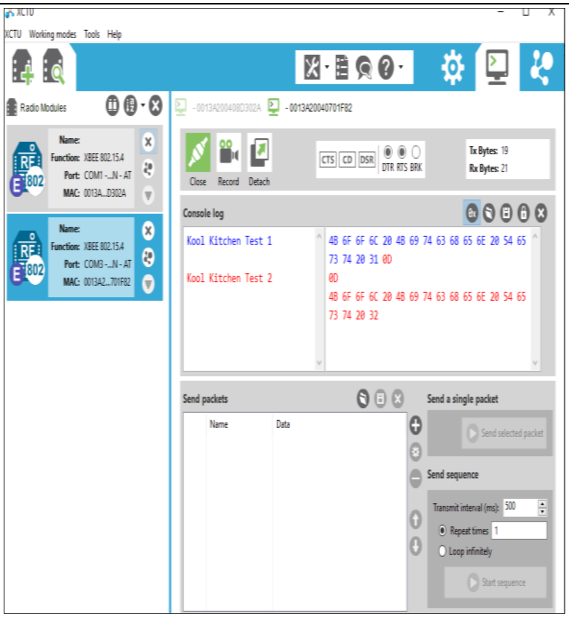
 Kool Kitchen Test 1 Kool Kitchen Test 2 4B 6F 6F 6C 20 4B 69 74 63 68 65 6E 20 54 65 73 74 20 31 0D 0D 4B 6F 6F 6C 20 4B 69 74 63 68 65 6E 20 54 65 73 74 20 32 Tx Bytes: 21 Rx Bytes: 19 Function: XBEE 802.15.4 Port: COM1 -...N - AT MAC: 0013A...D302A Function: XBEE 802.15.4 Port: COM3 -...N - AT MAC: 0013A2...701F82	 Kool Kitchen Test 1 Kool Kitchen Test 2 4B 6F 6F 6C 20 4B 69 74 63 68 65 6E 20 54 65 73 74 20 31 0D 0D 4B 6F 6F 6C 20 4B 69 74 63 68 65 6E 20 54 65 73 74 20 32 Tx Bytes: 19 Rx Bytes: 21 Function: XBEE 802.15.4 Port: COM1 -...N - AT MAC: 0013A...D302A Function: XBEE 802.15.4 Port: COM3 -...N - AT MAC: 0013A2...701F82
COM 1 sending and receiving data Left hand side shows which COM is selected Hexadecimal values can be viewed on right hand side	COM 3 sending and receiving data Left hand side shows which COM is selected Hexadecimal values can be viewed on right hand side

Figure 36 Devices Communicating

XBees are amazing in light of the fact that they're exceedingly – and effortlessly – configurable. The greater part of the XBee setup settings boil down to controlling which different XBees it can converse with. On this page, we'll demonstrate to you generally accepted methods to arrange three of the most essential XBee settings there are: PAN ID, MY location, and destination address.

There are a couple levels to XBee systems. To start with, there's the channel. This controls the recurrence band that your XBee conveys over. Most XBee's work on the 2.4GHz 802.15.4 band, and the channel further aligns the working recurrence inside that band. You can for the most part allow the channel setting to sit unbothered, or possibly ensure each XBee you need to have on the same system works on the same channel.

The following level of an XBee system is the individual region system ID (PAN ID). The system ID is some hexadecimal quality somewhere around 0 and 0xFFFF. XBees can just speak with each other in the event that they have the same system ID. There being 65536 conceivable ID's, there's a little risk that your neighbour will work on the same system (the length of you change it from the default!).

At last there are MY and destination addresses. Each XBee in a system ought to be doled out a 16-bit address (again somewhere around 0 and 0xFFFF), which is alluded to as MY location, or the "source" address. Some other setting, the destination address, figures out which source address an XBee can send information to. For one XBee to have the capacity to send information to another, it must have the same destination address as the other XBee's source.

For instance, if XBee 1 has a MY location of 0x1234, and XBee 2 has a proportionate destination location of 0x1234, then XBee 2 can send information to XBee 1. Yet, in the event that XBee 2 has a MY location of 0x5201, and XBee 1 has a destination location of 0x5200, then XBee 1 can't send information to XBee 2. For this situation, stand out way correspondence is empowered between the two XBee's (lone XBee 2 can send information to XBee 1) [8].
As I have already given unique PAN ids of my choice. Now it's time to work on other communication parameters like MY Address (MY) as now I am using two Xbees so any one I can use as 0 and other as 1.
The destination address defines which XBee your source XBee is talking to. There are actually two values used to set the destination: destination high (DH) and destination low (DL). You can use that pair of values in one of two ways to set your XBee's mate:

1. Leave DH set to 0, and set DL to the MY address of the receiving XBee.
2. Set DH to the Serial Number High (SH) and DL to the Serial Number Low (SL) of your destination XBee.[8]

Hence in the case my receiving Xbee is at COM1 and COM3 is my Sending Xbee.

And for configuring the two Xbees Refer to Table 1.

Table 1 Configuration of two Xbees

	XBEE 1	**XBEE 2**
CH	C	C
ID	1234	1234
DH	0	0
DL	1	0
MY	0	1

So now I made sure Xbees are communicating with each other than I will do remote sensing. For the serial communication I have added one code in Arduino which is given in Appendix 11.The hardware used for the making this connection is Arduino Uno, shield needed I have not used shields in my demonstrations as I don't need one .2 Xbees as I have used S1 series with Antenna and XBee shield have used for powering up the Xbees. Also for making connection I have used USB mini-B cables .Software to work on is X-CTU. First of all I will write about the connection I have made.The connection between Arduino and XBee can be shown below in Figure 37.

The connection is viewed below

3.3v to VCC (At XBee)

GND to GND (At XBee)

D03 to TX (At XBee)

D02 to RX (At XBee)

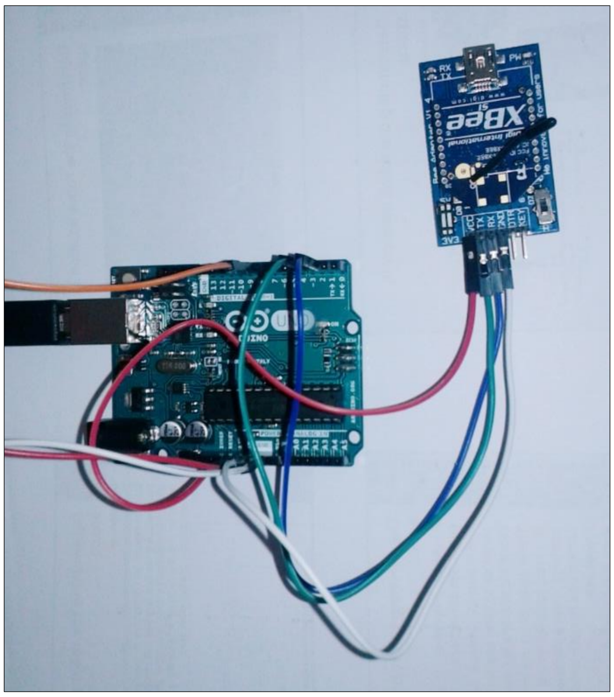


Figure 37 Connection between XBee and Arduino

Now to give inputs to Arduino I have used the same temperature sensor DHT11 and the connection is made and shown in the Figure 38.Also I have powered up the Arduino so to check Xbee and Arduino are working.Here what I have one XBee is connected to Arduino and Arduino is taking data from the temperature sensor and then sends this data wirelessly through another Xbee.And the connection between Arduino and sensor is made above. To recall sensor have 4 pins

Pin 1 Vcc

Pin 2 Data Input

Pin 3 not in use

Pin 4 GND

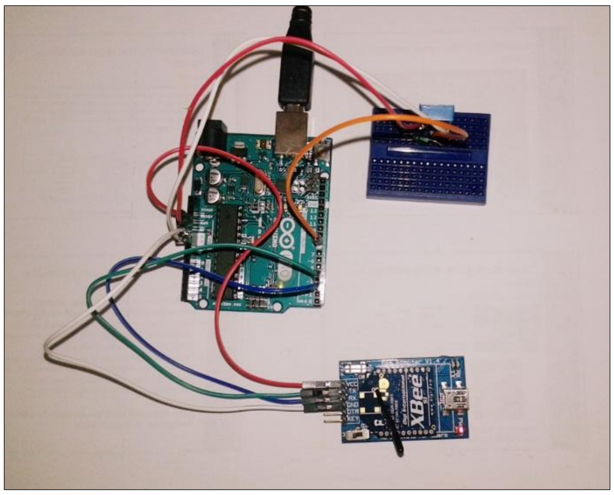


Figure 38 Full connection between Arduino Xbee and Sensor

Now the full connection is made and codes are written .Connection can be viewed in Figure 39.So the main thing here to see this for giving input to sensor pin D8 from so to power Xbee 3.3volts power supply is used and TX and RX made connected to D3 and D2 respectively.

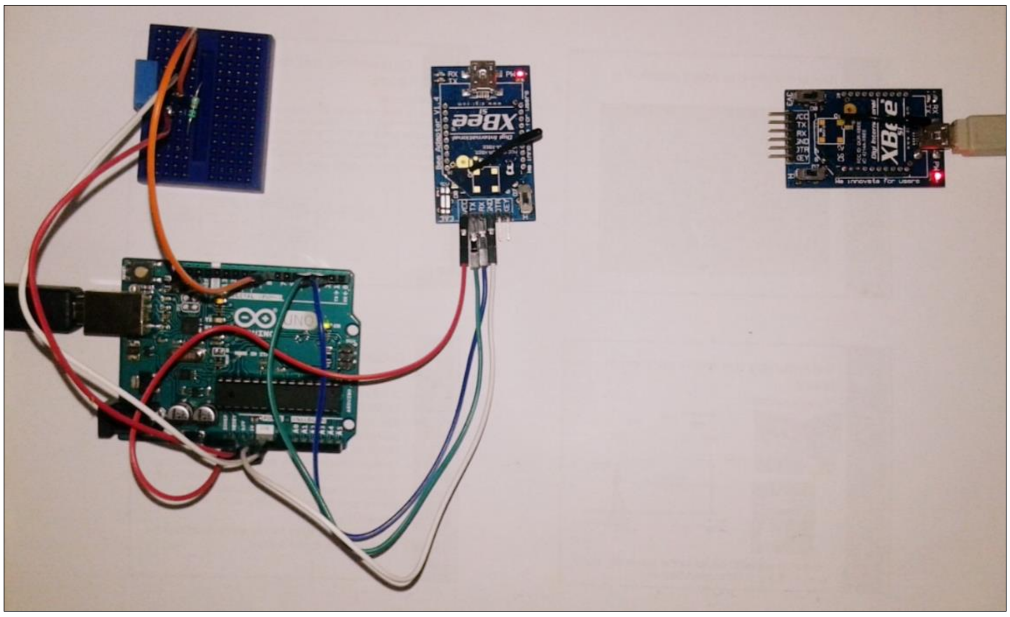


Figure 39 Full connection

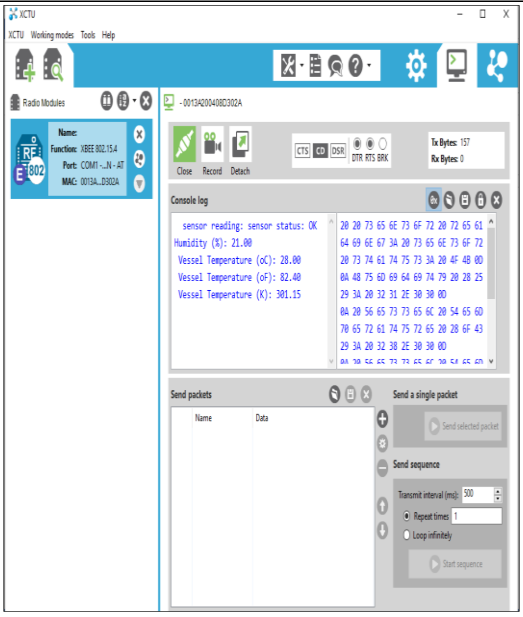
After you've transferred the code, tail this arrangement of ventures to check that everything is working:

1. Open the Arduino's Serial Monitor. Ensure the baud rate is set to 9600.

2. Switch to XCTU and snap over to reassure mode.

3. Type something in the console view, it ought to appear on the Serial Monitor.

3. Type something into the Serial Monitor (and press "Send"), it ought to appear in the console view. [14]



Name:
Function: XBEE 802.15.4
Port: COM1 -...N - AT
MAC: 0013A...D302A

- 0013A200408D302A

Tx Bytes: 157
Rx Bytes: 0

Console log

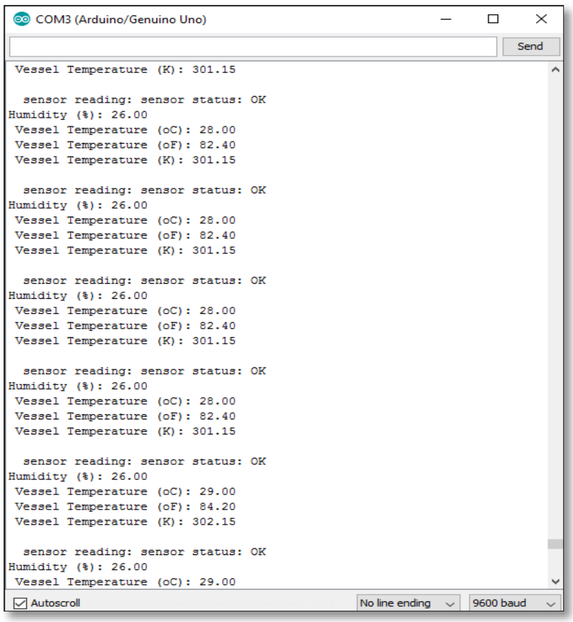
```
  sensor reading: sensor status: OK
Humidity (%): 21.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15
```

```
20 20 73 65 6E 73 6F 72 20 72 65 61
64 69 6E 67 3A 20 73 65 6E 73 6F 72
20 73 74 61 74 75 73 3A 20 4F 4B 0D
0A 48 75 6D 69 64 69 74 79 20 28 25
29 3A 20 32 31 2E 30 30 0D
0A 20 56 65 73 73 65 6C 20 54 65 6D
70 65 72 61 74 75 72 65 20 28 6F 43
29 3A 20 32 38 2E 30 30 0D
```

COM 1 sending and receiving data

Left hand side shows which COM is selected

Hexadecimal values can be viewed on right hand side



COM3 (Arduino/Genuino Uno)

```
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 28.00
 Vessel Temperature (oF): 82.40
 Vessel Temperature (K): 301.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 29.00
 Vessel Temperature (oF): 84.20
 Vessel Temperature (K): 302.15

  sensor reading: sensor status: OK
Humidity (%): 26.00
 Vessel Temperature (oC): 29.00
```

Autoscroll: ✓
No line ending
9600 baud

COM 3 sending and receiving data

Left hand side shows which COM is selected

Hexadecimal values can be viewed on right hand side

Figure 40 Remote sensing

5 MATLAB MODELLING

I am using MATLAB in this project because a constant check should be given to all devices and for that I will create small prototype for sensing computer. As above in examples I have used Temperature sensor DHT11 I will use same temperature sensor only. And an interface of MATLAB, Arduino and temperature sensor will be shown here. From the datasheet it specified that DHT11 yields 40bits of information. It contains 8-bits fundamental RH data+ 8-bits Decimal RH data+8-bits Integral Temperature data+8-bits decimal Temperature data+8-bits check total. Perused datasheet for more points of interest. The library for DHT11 sensor is accessible in the assets. Include the library in this way "C:\Program Files (x86)\Arduino\libraries\DHT11" (For my computer). [15]

For making the connections I have used same connection between Arduino and sensor to get temperature data and humidity data with the MATLAB.In this demonstrations I will be getting graphs which can be later saved to server for comparison and risk management. So code which is written in MATLAB can be viewed in Appendix12 and the Code written for Arduino can be viewed in Appendix 13.Also refer graph in Figure 41.Also as the room temperature was coming constant to make it look real and do some work I again kept ice-pack near my sensor to show variation in data.

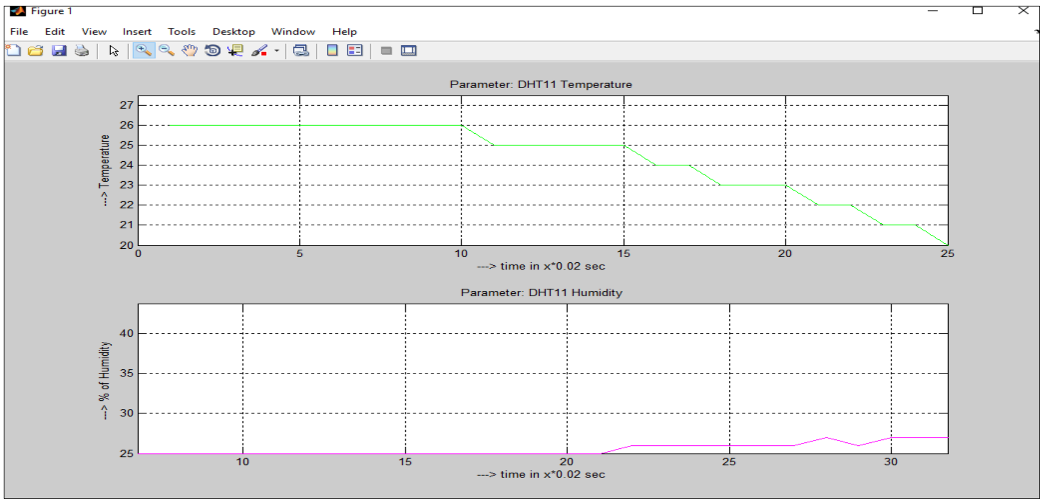


Figure 41 Temperature and Humidity Graph

I made the above illustration to show the temperature sensor can sense temperature out constantly and saves this data in the central Hub where I am using MATLAB so the constantly data can be monitored .

5.1 Simulink Model

To make my stove work through the MATLAB I have made one Simulink.

But the question arise how it will work in the end I will connect this Simulink model with databases and as I have already made connections between databases and android.

When android is sending a signal for OFF/ON it sends signal to data bases and that is connected to MATLAB and Arduino so the stove will be switched off. As the system is made wirelessly constant data will be read in the mobile.

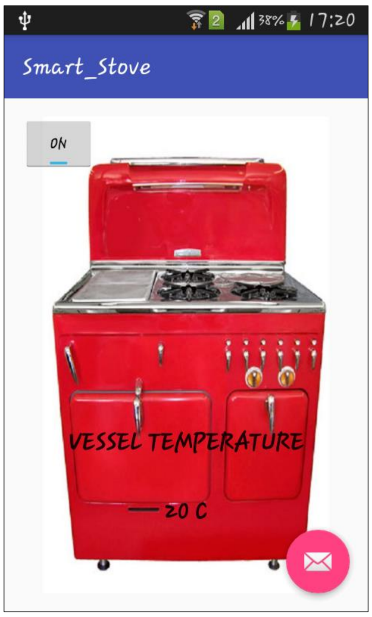


Figure 42 Smart_Stove Activity

Model which is given below In Figure 43 .So the model has three main parts.

I have mentioned three things: stove, vessel and controller.

As vessel temperature should be controlled hence the model serves the purpose and stove will go off when the temperature increases from the given set temperature points

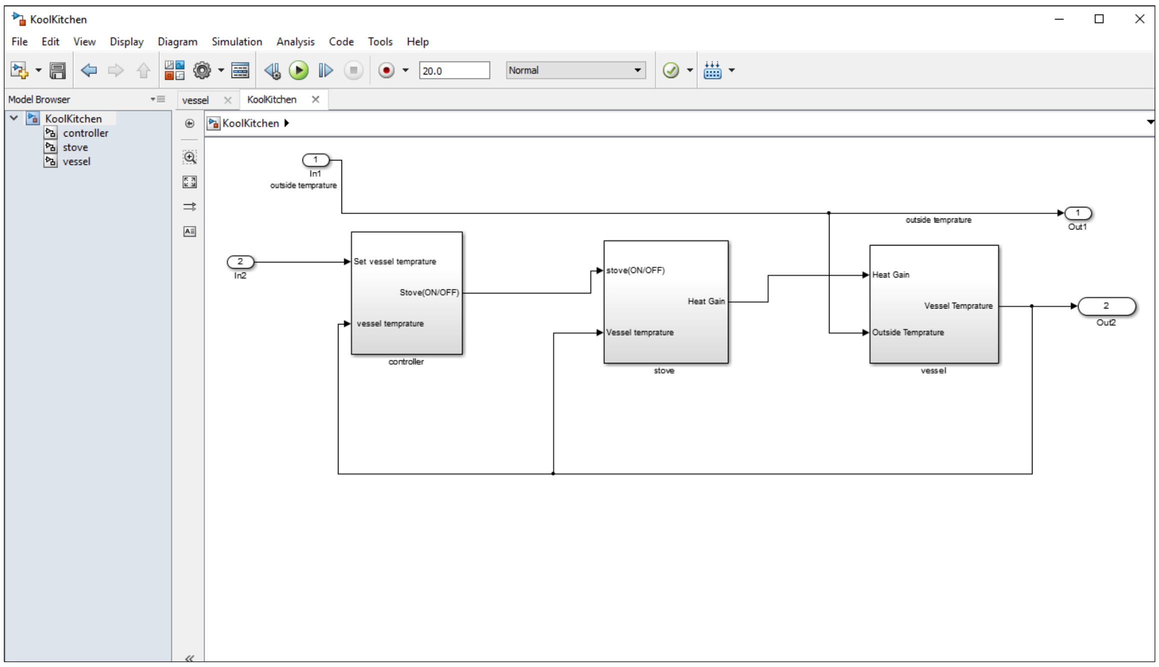


Figure 43 Simulink For stove

When the temperature rises and decreases from the given set point we see the variable that variation is shown below in the graph. Hence the set temperature of the vessel would be say 25℃ so variations are shown in Figure 44

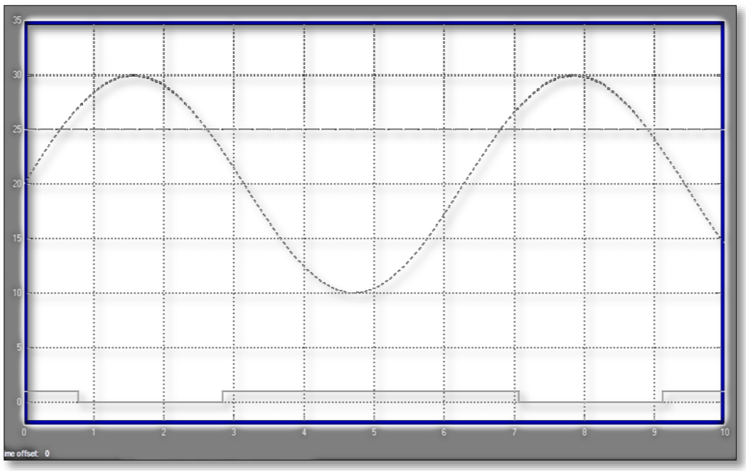


Figure 44 Temperature variations

For the smooth functioning of the model the set data should be fed into the controller to make necessary calculations and setting a set point. To simulate the stove and controller subsystems without the vessel subsystem, we need a signal for the changing vessel temperature. Using a constant block to set the controlled temperature and a sine wave block for a realistic outside temperature signal, the simulations are shown in Figure 45.

From about 0 to 1.5 hours, the stove is turned on. Heat gain is not constant but changes because heat gain is a function of the difference between the stove heat temperature and the vessel temperature. From 1.5 to 5.6 hours, the stove is turned off and the heat gain (top graph) is zero. The simulation confirms the expected behavior.

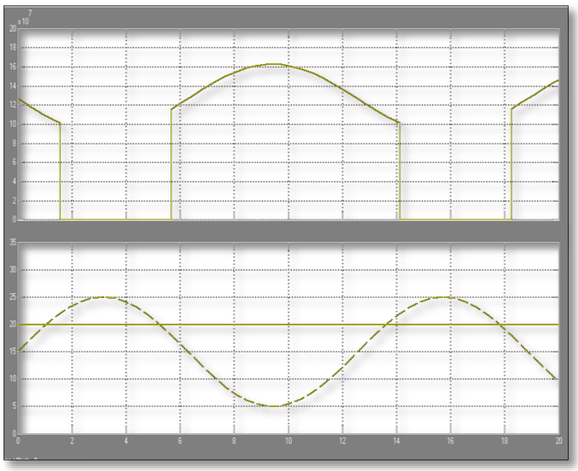


Figure 45 Changing vessel temperature

To simulate the stove and controller subsystems with the vessel subsystem, we need a signal for the changing outside temperature. Simulating the model allows you to observe how the controller setting and outdoor temperature affect the indoor temperature as shown in Figure 46.

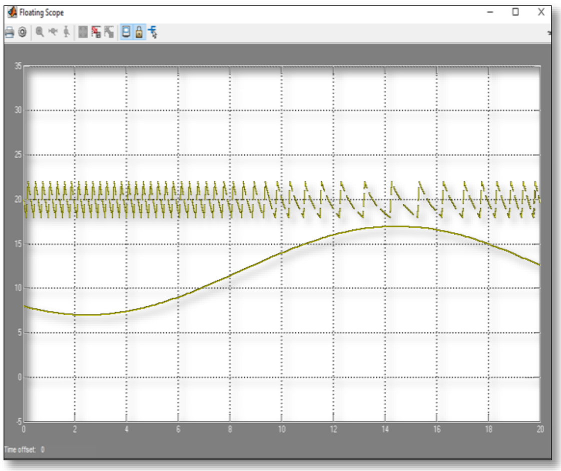


Figure 46 Changing vessel temperature

For the model to work on data .we need to feed data in the controller so that stove can get off when maximum temperature ceases .For that I made some measurements of temperature actually increasing in the vessel or pot. As 100 °C is the boiling point of the water and outside temperature and temperature of vessel without stove is on would be 20°C. I have considered it as room temperature. The room temperature is taken 25°C for the simulations

Also the subparts of the models are defined here for more clarity,

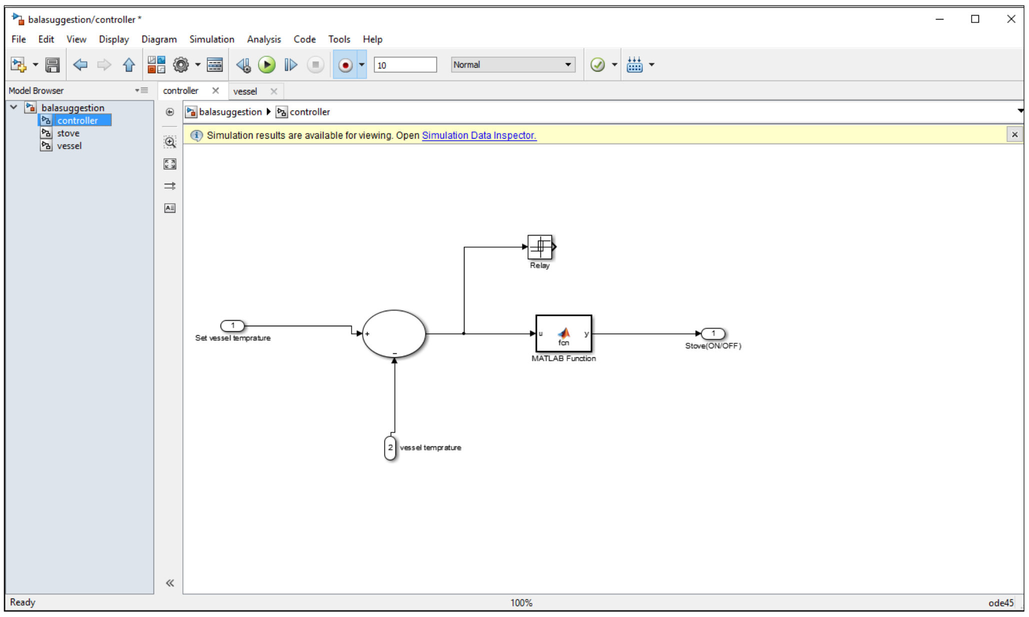


Figure 47 Controller Sub block

In the subpart controller I am using microcontroller which feeds in the data to vessel to set temperature of the vessel and which in end decide stove needs to be OF/ON depending upon the variable used. Important thing to observer here is stove will not be OFF instantly it has the default ability to fluctuate between above and below 2 degrees.

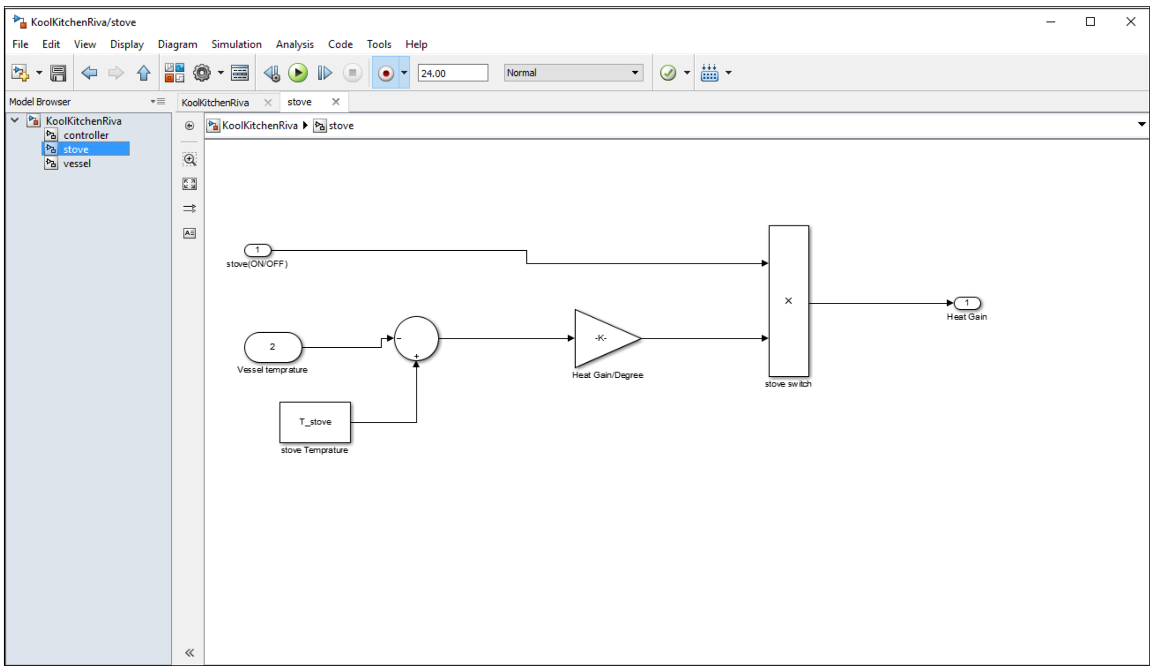


Figure 48 Stove Sub Block

Stove is the main part in this Simulink as this the main thing which takes in all operations to work on it and as I have mentioned that we are going to take in account vessel temperature to measure what is the temperature on stove I have added last block as vessel temperature.

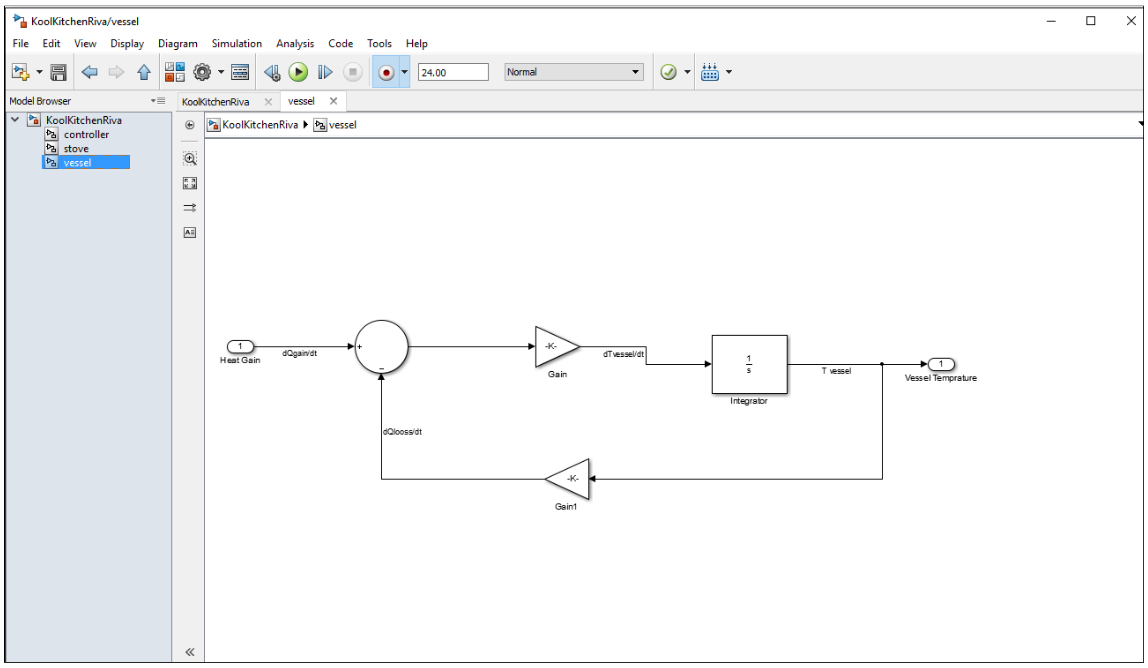


Figure 49 Vessel sub block

Let’s compare the data with the measured results’ will getting results in Simulink data inspector.

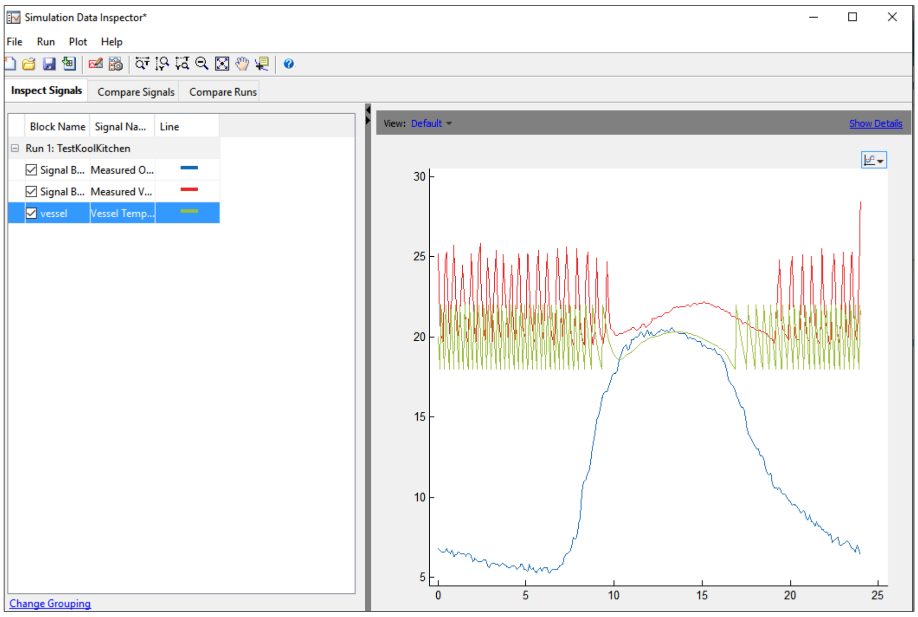


Figure 50 Comparing results

Finally when the stove is turning OFF/ON

Basically it's safer when it gets turn off automatically when it ceases the set temperature. But to be able to automate it with just a click of a mobile we should be able to on stove from our smart phones. And now we determine changes to the model

One obvious change to the model is the hysteresis of the thermostat. The simulated room temperature oscillates between 18 and 22 degrees around the temperature set point of 20 degrees. The measured room temperature oscillates between 20 and 25 degrees with the same set point.

1. Open the Relay block in the Thermostat subsystem.
2. Change Switch on point from 2 to 0 because the difference between the room temperature and set point is 0.
3. Change Switch off point from -2 to -5. When the room temperature is 5 degrees above the set point, you want to turn the heater off. The set point is –5 degrees below the room temperature.

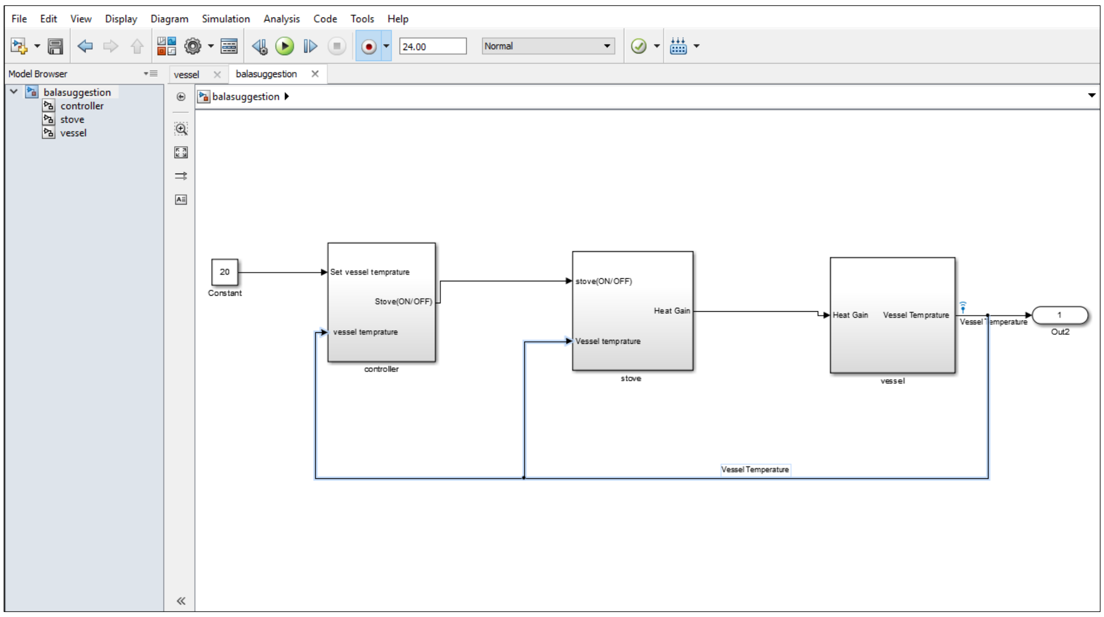


Figure 51 Determining changes

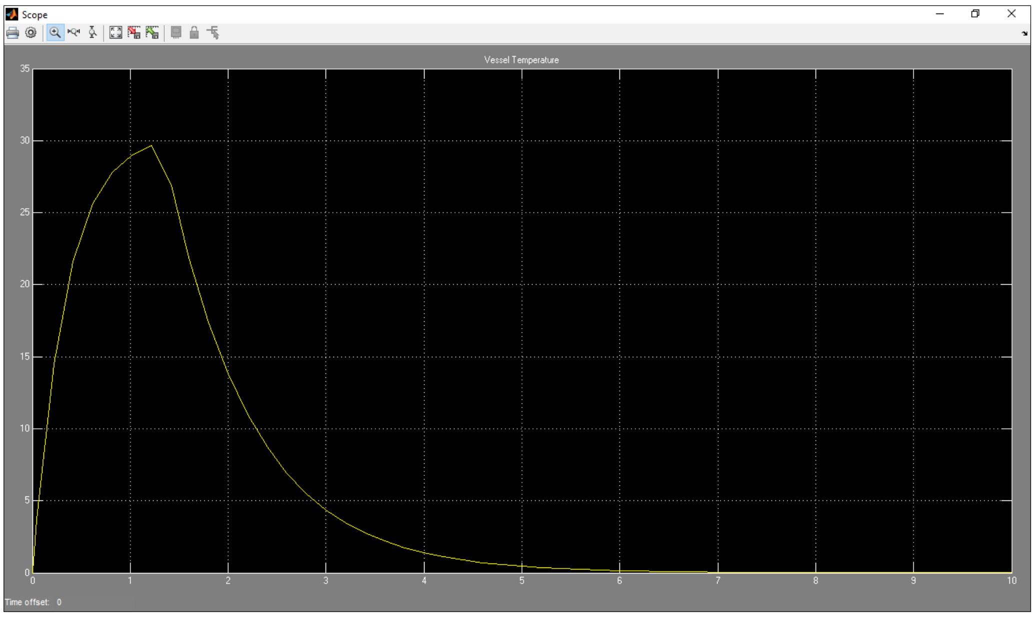


Figure 52 Final output

In Figure 52, it is clearly shown that stove goes off when the temperature reaches the set point .For my sensor the temperature range is not too high so I selected 30 degrees to demonstrate and after it reaches the set point the stove goes off that is pure logically step.

Also the stove will stay off till user ON it manually or by the application.

5.2 Connecting Databases to MATLAB

It is very important to connect MATLAB to database as my data which is temperature data and expiry date data is sent to server .MATLAB should be able to pull that data from the server .for that I need a MATLAB and SQL table created.

Driver to associate with the MySQL database.
Step 1. Check the driver establishment.
Step 2. Set up the information source utilizing Database Explorer.
Step 3. Associate utilizing Database Explorer or the charge line.
The ODBC driver is regularly preinstalled on your PC. For insights about the driver establishment or investigating the establishment, contact your database director or allude to your database documentation on ODBC drivers.

To make the work together I need my-sql.java connector file which has all libraries we need and save the connector file in MATLAB .Host name user name and the password is already being defined in the server section of the report when the connection is made between server and android here I can use same data and can check if the server is online or not

By going to http://koolkitchen.comxa.com/riva.php

Hence I have checked yes I am online or Active so everything is ready .Make sure the sql java file is called correctly in the code.

Next step is to select name of the table product_info is the table name I have defined and a7647856_ITEM (1) is the SQL database name. Hence with a code I can connect to MATLAB to online data base. And in the left hand side I will be able to see all the values defined.

6 Kokkuvõte

Kodu automatiseerimine on järgmine samm, millega insenerid parandavad elukvaliteeti, et inimesed saaksid keskenduda rohkem olulisematele asjadele elus. Need süsteemid ühendavad kodus kõik eraldiseisvad süsteemid ühte süsteemi. Kodu automaatikasüsteemid on saanud rohkem ja rohkem populaarsemaks, sest need on odavamad ja rohkem kättesaadavamad massidele. Nende usaldusväärsus on samuti paranenud viimastel aastatel, mis on seotud nutiseadmete, nutitelefonide kasutusega. Samuti on paranenud interneti kättesaadavus ülemaailma, mis on teinud võimalikuks selle, et inimesed saavad kontrollida koduseadmeid interneti vahendusel erinevatest maailmaosadest.

Antud lõputööks on android rakendus, mis ühendab kasutaja ja seadme distantsilt. See ei ole abiks ainult erinevate puuetega inimestele vaid samuti tudengitele ja inimestele, kes on eriti hõivatud igapäevaste tegemistega. See on uue kontseptiga külmkapp, mille kasutaja teab, mis on külmikus ja samuti teab nende toodete säilivustähtaegu. Jaemüüjad tänapäeval lihtsalt kasutavad triipkoode selleks, et hind määrata tootele. Kodu automaatikasüsteem sisaldab põhimõtteliselt kolme lihtsat asja, nagu masinad mida kasutame igapäevaselt, külmikud, ahjud ja mikrolaineahjud.

Arduino Uno moodul teeb selle ära. See kogub ja töötleb signaalid ja saadab need traaditaühenduse abil teise moodulisse, SQL andmebaasid ja MATLAB kesk arvutis, kasutavad zig bee moodulit. Arvutuse tulemused MATLAB-ist saadetakse tagasi autonoomsesse mobiilirakendusse nende käskude täidesaatmiseks reaalajas.

Töö on jaotatud mitmetesse osadesse, milleks on androidi rakenduse arendus ja andmete lisamine pilve. Teine ja kõige olulisem osa on riistvara, andurid on ühendatud mikrokontrolleritega ja need saadavad andmed edasi. Oluline on väljatuua, et need seadmed on kasutajasõbralikud ja neid seadmeid saab tulevikus proovida. Esiteks on uuritud triipkoodi kasutamise võimalust androidi rakenduse asemel, aga see osutus natukene ebamugavaks, seega on parem rakendus, mis lisab info serverisse.

7 SUMMARY

Home automation is the next step in the endeavour of the community of engineers to improve the quality of life, so that people can concentrate on the more important things in their lives. These systems combines all the electronics in any given home into a single system. The home automation systems are getting more and more popular these-days as they are becoming cheaper and more affordable for the masses. Their reliability has also improved in the recent years and with the advent of smartphones, tablets and the improved internet connectivity all over the world has made it possible for people to control the devices in their homes to be controlled from any corner of the world.

This thesis work concludes the making the android application which enable user to operate his devices from distance. It is not only a help for differently abled but for students and people who are busy with hectic life styles. This works includes a new concept of fridge by which user can be reminded of food items in the fridge and their expiry dates. These days' retailers are just including prize in the barcodes information. This home automation system which basically includes three basic kitchen appliances which we use in our home daily such as refrigerators, stoves and microwave.

Here Arduino Uno module is used to do the same. It collects and processes the signals and send them wirelessly to another module, the SQL Databases and MATLAB in central Hub computer, using a zig bee module. Computational results/solutions from MATLAB are sent back to those autonomous mobile application for their executions and interactions as in the real time.

The work is divided into many parts like development of android application and adding data on cloud .Second and most important part which requires core hardware is Sensor connecting to microcontroller and microcontroller sending data remotely. Then the computational part with the MATLAB

It is interesting to know how user friendly these devices will be prove in near future. First I have researched on using barcode instead of android application but it proved to bit inconvenient for the user so I switched on making application adding data to server.

8 REFERENCES

[1] *Barcode*. (2016, May 15). Retrieved from Wikimedia Foundation, Inc.: 1https://wikimediafoundation.org/

[2] Electronics, L. (2014, June 05). LG Rolls Out Premium Smart Appliances that Chat. *LG Rolls Out Premium Smart Appliances that Chat*.

[3] Pennington, O. (2015, July 28). *Does the bar code on groceries include the expiry date?* Retrieved from www.quora.com: https://www.quora.com/Does-the-bar-code-on-groceries-include-the-expiry-date/answer/Orit-Pennington

[4] Sheikh Ferdoush, Xinrong Li,*Department of Electrical Engineering, University of North Texas, Denton, Texas, 76203, USA* ,August 2014

[5] J.-Y. Son, et al, "*Resource-Aware smart home management system by constructing resource relation graph*," IEEE Trans. On Consumer Electronics, vol. 57, pp. 1112-1119, 2011.

[6] Vu Trieu Minh, *Development of a Wireless Sensor Network Combining MATLAB and Embedded Microcontrollers* Sensor Letters 13(12):1091-1096, 2015

[7] Meier, R. (2009). Professional Android Application Development. Canada: Wilsey Publishing,Inc.

[8] Jonathan Stark, B. J. (Jan 13, 2012). Building Android Apps with HTML, CSS, and JavaScript: Making Native Apps with Standards-Based Web Tools. O'Reilly Media, Inc.

[9] Friesen, J. (2014). *Learn Java for Android Development: Java 8 and Android 5 Edition.* Springer, Part of Springer Science+Business Media.

[10] Meloni, J. C. (2012). *Sams Teach Yourself PHP, MySQL and Apache All in One.* Sams Publishing.

[11] Peterson, M. P. (2014). *Mapping in the Cloud.* Guilford Publications.

[12] Adafruit [WWW] https://www.adafruit.com/product/386 (14.03.2016).

[13] Spark Fun Electronics (US) [WWW] https://www.sparkfun.com/ (7.04.2016).

[14] Spark Fun Electronics (US) [WWW] https://learn.sparkfun.com/blog (10.04.2016).

[15] A. Thati(November 7,2014) DHT11 Interfacing with MATLAB and Arduino http://embedded-electronics.blogspot.com.ee/

[16] Joshi, S. (2015, August 31). Courses /Learning Elementary MATLAB (One on One Course). Learning Elementary MATLAB .

9 APPENDICES

Appendix 1

```
?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   app:layout_behavior="@string/appbar_scrolling_view_behavior"
   tools:context="com.example.kitchenfinal.koolkitchen.MainActivity"
   tools:showIn="@layout/activity_main"
   android:background="#ffff00">

   <ImageButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/image_Button1"
      android:src="@drawable/fridge_click"
      android:layout_alignParentTop="true"
      android:layout_toRightOf="@+id/image_Button2"
      android:layout_toEndOf="@+id/image_Button2"
      android:onClick="callFridge" />

   <ImageButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/image_Button2"
      android:src="@drawable/stove_click"
      android:layout_marginBottom="116dp"
      android:layout_alignBottom="@+id/image_Button3"
      android:layout_toLeftOf="@+id/image_Button3"
      android:layout_toStartOf="@+id/image_Button3"
      android:onClick="callStove" />

   <ImageButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/image_Button3"
      android:src="@drawable/microwave_click"
      android:layout_marginTop="66dp"
      android:layout_below="@+id/image_Button1"
      android:layout_alignParentRight="true"
      android:layout_alignParentEnd="true"
      android:onClick="callMicrowave" />
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="FRIDGE"
        android:id="@+id/textView"
        android:textStyle="bold"
        android:layout_alignBottom="@+id/image_Button1"
        android:layout_alignLeft="@+id/textView3"
        android:layout_alignStart="@+id/textView3" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="STOVE"
        android:id="@+id/textView2"
        android:textStyle="bold"
        android:layout_alignBottom="@+id/image_Button2"
        android:layout_alignLeft="@+id/image_Button2"
        android:layout_alignStart="@+id/image_Button2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="MICROWAVE"
        android:id="@+id/textView3"
        android:textStyle="bold"
        android:layout_alignBottom="@+id/image_Button3"
        android:layout_alignLeft="@+id/image_Button3"
        android:layout_alignStart="@+id/image_Button3" />

</RelativeLayout>
```

Appendix 2

```
package com.example.kitchenfinal.koolkitchen;

import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageButton;

public class MainActivity extends AppCompatActivity {

    private static ImageButton button_sbm;
    private static ImageButton button_sbm1;




    public void init(){}


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);


        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();

            }
        });
    }


    public void callFridge(View view){
```

```
                    Intent intent=new Intent(this,Fantastic_Fridge.class);
                    startActivity(intent);

    }


    public void callStove(View view){


        Intent intent=new Intent(this,Smart_Stove.class);
        startActivity(intent);

    }
    public void callMicrowave(View view){


        Intent intent=new Intent(this,Master_Microwave.class);
        startActivity(intent);

    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }




    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

Appendix 3

Java file for Fantastic_fridge

```
package com.example.kitchenfinal.koolkitchen;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class Fantastic_Fridge extends Activity {

    Button B1;
    Button B2;
    TextView textView;





    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fantastic__fridge);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
                B1=(Button) findViewById(R.id.b1);
                B2=(Button) findViewById(R.id.b2);
                textView=(TextView) findViewById(R.id.textView6);

                ConnectivityManager
connectivityManager=(ConnectivityManager)getSystemService(Context.CONNECTIVITY_
SERVICE);
                NetworkInfo networkInfo=connectivityManager.getActiveNetworkInfo();
      /* if(networkInfo!=null&& networkInfo.isConnected())
```

```
          {
           textView.setVisibility(View.INVISIBLE);

           }
               else
        {
          B1.setEnabled(false);
           B2.setEnabled(false);

        }
*/
               B1.setEnabled(false);
               B2.setEnabled(false);

            }
         });
     }

   public void itemexpiry(View view)
   {

startActivity(new Intent(this,Fridge_Fragement.class));

   }



}
```

Appendix 4

Java file for Smart_stove

```
package com.example.kitchenfinal.koolkitchen;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;

public class Smart_Stove extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_smart__stove);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }

}
```

Appendix 5

Java file for Master_microwave

```
package com.example.kitchenfinal.koolkitchen;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;

public class Master_Microwave extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_master__microwave);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }

}
```

Appendix 6

```
////////Add_info.php////////

<?php
 require "riva.php";

$item = $_POST["item"];
$expirydate =  $_POST["expirydate"];

$sql = "insert into product_info values ('$item' , '$expirydate');";

if(mysqli_query($con,$sql) )
{

 echo "<br><h3>one Row Inserted....</h3> ";
}
 else
 {
 echo "Error in insertion.......". mysqli_error($con);
 }
 ?>
```

Appendix 7

```
////////Index.php////////
<html>
<head><title>Add info....</title></head>
<body>

<form action="add_info.php" method="post">

<table>
<tr>
<td>Item: </td>
<td><input type="varchar" name="item" /></td>
</tr>



<tr>
<td>ExpiryDate: </td>
<td><input type="date" name="expirydate" /></td>
</tr>
</table>

<input type="submit" value="Submit Info" />

</form>
</body>


</html>
```

Appendix 8

Java file for Fragment_fridge

```
package com.example.kitchenfinal.koolkitchen;

import android.app.Activity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.common.api.GoogleApiClient;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;

public class Fridge_Fragement extends Activity {

    EditText Item, ExpiryDate;
    String item, expirydate;
    /**
     * ATTENTION: This was auto-generated to implement the App Indexing API.
     * See https://g.co/AppIndexing/AndroidStudio for more information.
     */
    private GoogleApiClient client;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fridge__fragement);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
                Item = (EditText) findViewById(R.id.et_item);
                ExpiryDate = (EditText) findViewById(R.id.et_date);
                BackgroundTask backgroundTask = new BackgroundTask();
                backgroundTask.execute(item, expirydate);
                finish();

            }
        });

        // ATTENTION: This was auto-generated to implement the App Indexing API.
        // See https://g.co/AppIndexing/AndroidStudio for more information.
        client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
    }

    public void saveInfo(View view) {

        item = Item.getText().toString();
        expirydate = ExpiryDate.getText().toString();
        BackgroundTask backgroundTask=new BackgroundTask();
        backgroundTask.execute(item,expirydate);
        finish();

    }





    class BackgroundTask extends AsyncTask<String, Void, String> {
        String add_info_url;

        @Override
        protected void onPreExecute() {

            add_info_url = "http://koolkitchen.comxa.com/add_info.php";


        }

        @Override
        protected String doInBackground(String... args) {
            String item, expirydate;
            item = args[0];
            expirydate = args[1];
            try {
                URL url = new URL(add_info_url);
                HttpURLConnection
httpURLConnection=(HttpURLConnection)url.openConnection();
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                OutputStream outputStream=httpURLConnection.getOutputStream();
                BufferedWriter bufferedWriter=new BufferedWriter(new
OutputStreamWriter(outputStream,"utf-8"));
                String data_string= URLEncoder.encode("item","utf-
8")+"="+URLEncoder.encode(item,"utf-8")+"&"+
                        URLEncoder.encode("expirydate","utf-
8")+"="+URLEncoder.encode(expirydate,"utf-8");

                bufferedWriter.write(data_string);
                bufferedWriter.flush();
                bufferedWriter.close();
                outputStream.close();
                InputStream inputStream=httpURLConnection.getInputStream();
                inputStream.close();
                httpURLConnection.disconnect();
                return "One row of Data inserted...";


            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }

            return null;
        }

        @Override
        protected void onProgressUpdate(Void... values) {
            super.onProgressUpdate(values);
        }

        @Override
        protected void onPostExecute(String result) {
            Toast.makeText(getApplicationContext(),result,Toast.LENGTH_LONG).show();


        }
    }


}
```

Appendix 9

Xml code for Fantastic_fridge

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  app:layout_behavior="@string/appbar_scrolling_view_behavior"
  tools:context="com.example.kitchenfinal.koolkitchen.Fantastic_Fridge"
  tools:showIn="@layout/activity_fantastic__fridge">

  <ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageView"
    android:src="@drawable/background_fridge"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ITEM EXPIRY"
    android:id="@+id/b1"
    android:background="#fde404"
    android:textStyle="bold"
    android:onClick="itemexpiry"
    android:layout_marginTop="60dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginLeft="72dp"
    android:layout_marginStart="72dp" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ITEM WEIGHT"
    android:id="@+id/b2"
    android:background="#fde404"
    android:textStyle="bold"
    android:onClick="itemweight"
    android:layout_centerVertical="true"
```

```
        android:layout_centerHorizontal="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="NO NETWORK FOUND"
        android:id="@+id/textView6"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="73dp"
        android:textStyle="bold" />
</RelativeLayout>
```

Appendix 10

Arduino Code for Sensor

```
#include <idDHT11.h>


/*
  Board         int.0      int.1   int.2   int.3   int.4   int.5
 Uno, Ethernet  2          3
 Mega2560       2          3       21      20      19      18
 Leonardo       3          2       0       1
 Due            (any pin, more info http://arduino.cc/en/Reference/AttachInterrupt)
 */
int idDHT11pin = 2; //Digital pin for comunications
int idDHT11intNumber = 0; //interrupt number (must be the one that use the previus defined pin (see table above)
//declaration
void dht11_wrapper(); // must be declared before the lib initialization
// Lib instantiate
idDHT11 DHT11(idDHT11pin,idDHT11intNumber,dht11_wrapper);
void setup()
{
  Serial.begin(9600);
  Serial.println("idDHT11 Example program");
  Serial.print("LIB version: ");
  Serial.println(IDDHT11LIB_VERSION);
  Serial.println("---------------");
}
// This wrapper is in charge of calling
// mus be defined like this for the lib work
void dht11_wrapper() {
  DHT11.isrCallback();
}
void loop()
{
  Serial.print("\n  sensor reading: ");
  Serial.print("sensor status: ");
  //delay(100);
  DHT11.acquire();
  while (DHT11.acquiring())    ;
int result = DHT11.getStatus();
  switch (result)
  {
  case IDDHTLIB_OK:
    Serial.println("OK");
    break;
  case IDDHTLIB_ERROR_CHECKSUM:
    Serial.println("Error\n\r\tChecksum error");
    break;
  case IDDHTLIB_ERROR_ISR_TIMEOUT:
    Serial.println("Error\n\r\tISR Time out error");
    break;
  case IDDHTLIB_ERROR_RESPONSE_TIMEOUT:
    Serial.println("Error\n\r\tResponse time out error");
    break;
  case IDDHTLIB_ERROR_DATA_TIMEOUT:
    Serial.println("Error\n\r\tData time out error");
    break;
  case IDDHTLIB_ERROR_ACQUIRING:
    Serial.println("Error\n\r\tAcquiring");
    break;
  case IDDHTLIB_ERROR_DELTA:
    Serial.println("Error\n\r\tDelta time to small");
    break;
  case IDDHTLIB_ERROR_NOTSTARTED:
    Serial.println("Error\n\r\tNot started");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
  Serial.print("Humidity (%): ");
  Serial.println(DHT11.getHumidity(), 2);
Serial.print(" Vessel Temperature (oC): ");
```

```
  Serial.println(DHT11.getCelsius(), 2);
Serial.print(" Vessel Temperature (oF): ");
  Serial.println(DHT11.getFahrenheit(), 2);
Serial.print(" Vessel Temperature (K): ");
  Serial.println(DHT11.getKelvin(), 2);
delay(2000);
}
```

Appendix 11

```
#include "DHT.h"

#include <SoftwareSerial.h>


#define DHTTYPE DHT11

#define DHTPIN 8


// XBee's DOUT (TX) is connected to pin 2 (Arduino's Software RX)

// XBee's DIN (RX) is connected to pin 3 (Arduino's Software TX)

SoftwareSerial XBee(2, 3); // RX, TX


// Initialize DHT sensor.

// Note that older versions of this library took an optional third parameter to

// tweak the timings for faster processors.  This parameter is no longer needed

// as the current DHT reading algorithm adjusts itself to work on faster procs.

DHT dht(DHTPIN, DHTTYPE);


void setup() {

    // Set up both ports at 9600 baud. This value is most important

  // for the XBee. Make sure the baud rate matches the config

  // setting of your XBee.

  XBee.begin(9600);

  Serial.begin(9600);

  dht.begin();

}
```

```
void loop() {
  // Wait a few seconds between measurements.
delay(1000);

float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
if ( isnan(t) ) {
Serial.println("Failed to read from DHT sensor!");
return;
  }
    XBee.println(t);
    Serial.println(t);
  }
```

Appendix 12

Temperature data through Arduino to MATLAB

```
>> s = serial('COM3');
time=100;
i=1;
while(i<time)

fopen(s)
fprintf(s, 'Your serial data goes here')
out = fscanf(s)

Temp(i)=str2num(out(1:4));
 subplot(211);
 plot(Temp,'g');
 axis([0,time,20,50]);
title('Parameter: DHT11 Temperature');
xlabel('---> time in x*0.02 sec');
ylabel('---> Temperature');
grid
Humi(i)=str2num(out(5:9));
 subplot(212);
 plot(Humi,'m');
axis([0,time,25,100]);
title('Parameter: DHT11 Humidity');
xlabel('---> time in x*0.02 sec');
ylabel('---> % of Humidity ');
grid

fclose(s)
i=i+1;
drawnow;
end
```

```
delete(s)

clear s

out =

26.0025.00

out =

26.0025.00
out =

26.0025.00

out =

26.0025.00

out =

26.0025.00

out =

26.0025.00

out =

26.0025.00

out =

26.0025.00

out =

26.0025.00

out =

26.0025.00

out =

25.0025.00

out =

25.0025.00

out =

25.0025.00

out =

25.0025.00

out =

25.0025.00
```

```
out =

24.0025.00

out =

24.0025.00

out =

23.0025.00

out =

23.0025.00

out =

23.0025.0

out =

22.0025.00
```

Appendix 13

Arduino code for DHT11

```
The circuit:

 * LCD RS pin to digital pin 12

 * LCD Enable pin to digital pin 11

 * LCD D4 pin to digital pin 5

 * LCD D5 pin to digital pin 4

 * LCD D6 pin to digital pin 3

 * LCD D7 pin to digital pin 2

 * LCD R/W pin to ground

 * 10K resistor:

 * ends to +5V and ground

 * wiper to LCD VO pin (pin 3)

 */

// include the library code:

#include <DHT11.h>

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);



int pin=8;

DHT11 dht11(pin);

void setup()

{

 lcd.begin(16, 2);
```

```
  Serial.begin(9600);
  while (!Serial) {
      ; // wait for serial port to connect. Needed for Leonardo only
    }
}

void loop()
{
  int err;
  float temp, humi;
  if((err=dht11.read(humi, temp))==0)
  {
    //Serial.print("temperature:");
    Serial.print(temp);
    lcd.setCursor(0, 1);
    lcd.print("T:");
    lcd.print(temp);
    lcd.print("C");

    //Serial.print(" humidity:");
    Serial.print(humi);
    lcd.setCursor(8,1);
    lcd.print("H:");
    lcd.print(humi);
    lcd.print("%");
    Serial.println();
  }
  else
  {
    Serial.println();
    Serial.print("Error No :");
    lcd.print("Error No:");
    Serial.print(err);
    lcd.print(err);
    Serial.println();
  }
  delay(DHT11_RETRY_DELAY); //delay for reread
  }
```

Appendix 14

CH=Channel

ID =PAN ID

DH =Destination Address High

DL= Destination Address Low

MY=16-bit Source Address

VCC=Power Supply

GND=Ground

D02=Digital output Pin 2

D03=Digital Output Pin 3

TX=Transmitter

RX=Receiver