

Internet of Things (on Android Platform)

Submitted By:

Varun Prasad

B. E. Second Year

Instrumentation and Electronics Engg
Jadavpur University, Kolkata

Submitted To:

Dr. P.K.Mishra

Senior Scientist
AMT Department
CSIR – CIMFR, Dhanbad



CSIR – Central Institute of Mining & Fuel Research

Barwa Road, Dhanbad – 826001

ACKNOWLEDGEMENT

Every project begins with an idea and materializes with concrete efforts. In the beginning, I would like to thank the almighty God who gave me the strength and capability to work on this topic and complete it successfully.

I would like to extend my sincere thanks to *Dr. P.K. Mishra (Sr. Scientist), Advanced Mining Technology Division, CSIR-CIMFR, Dhanbad* for his incessant encouragement, inspiring supervision and valuable guidance during the course of project.

I am grateful to *Director, CSIR-CIMFR Dhanbad* for granting me the permission to do the project work.

I am extremely thankful to *Subhash Kumar (Senior Research Fellow)* for his help and guidance. His constant support, experience, and technical knowledge helped me complete the project in time.

I will end this by appreciating the support and cooperation, I received from my family, without them none of my work would have been possible.

Varun Prasad

B. E. (Second Year)
Instrumentation and Electronics Engg
Jadavpur University, Kolkata

Content

1. Introduction	3
2. Objective	4
3. Design and Implementation	5
a. Proposed Monitoring System	6
b. Software Design	7
i. <i>HTML Code</i>	
ii. <i>Arduino Code</i>	
iii. <i>Java Code</i>	
c. Hardware Design	16
i. <i>Sending Data</i>	
ii. <i>Receiving Data</i>	
d. Result	18
4. Conclusion	19
5. References	20

Introduction

Internet of Things represents a general concept for the ability of network devices to sense and collect data from the world around us, and then share that data across the Internet where it can be processed and utilized for various interesting purposes.

What is the “Things” in “Internet of Things”?

Any device that can be embedded with electronics, software, sensors to communicate with other device is “Things”. A Thing, in the Internet of Things, can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low -- or any other natural or man-made object that can be assigned an IP address and provided with the ability to transfer data over a network. These devices collect useful data with the use of various existing technologies and then autonomously flow the data between other devices

What Internet of Things can do for us?

Some future consumer applications envisioned for IoT sound like science fiction, but some of the more practical and realistic sounding possibilities for the technology include:

- Receiving warnings on your phone or wearable device when IoT networks detect some physical danger nearby.
- Self-parking automobiles.
- Automatic ordering of groceries and other home supplies.
- Automatic tracking of exercise habits and other day-to-day personal activity including goal tracking and regular progress reports.
- Fuel savings from intelligent environmental modeling of gas-powered engines.
- New and improved safety controls for people working in hazardous environments.

And a lot of other interesting things.

Benefits of IoT in Mining Industry

- ***IoT improving safety:*** IoT can help prevent the collapse of unstable shafts for example, because the sensors will pick up real-time data, and predict the faulty equipment/where issues may occur, which allows for adjustments to be made before anything goes wrong.
- ***Energy and cost benefits:*** IoT reduces energy expenditure and maintenance cost for mining companies. By having a transparent system, where all parts are monitored. The cost spent on workers will reduce, allowing the industry to become more profitable.
- ***Automation advances:*** By having standalone products, i.e. vehicles and equipment work together, more data is collected, which can improve accuracy rates.
- ***Predictive Maintenance:*** Having a fully integrated network, monitoring every aspect of an operation becomes far easier, and only leads to greater productivity and safety. This includes detecting wear and tear on vital pieces of equipment, as well as projecting when repairs or maintenance is required.

Objective

The objective of the project was to design, implement and build a working model of wireless real-time data transmission from multiple sensors in a remote location to a specified web server host and monitor the sensor's reading on an android smartphone and also store the data simultaneously.

Core components used in the project:

- 2 Arduino Mega
- Multiple Environment Sensors
- XBee pair (for data transmission)
- WiFi Shield (as a server host)
- Android Smartphone (for monitoring purpose)
- SD Card (for data storage)
- Arduino IDE
- Android Studio 2.3.3

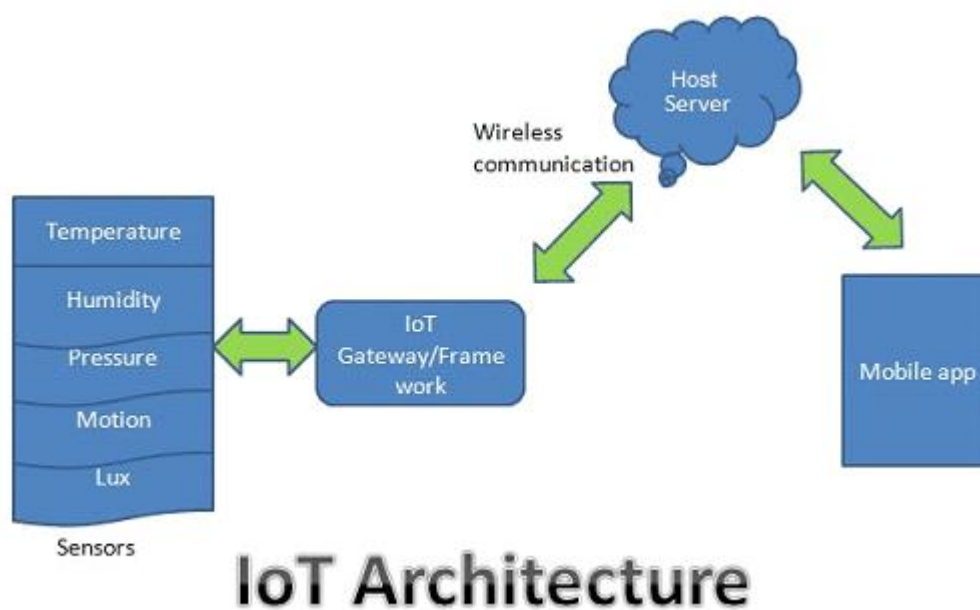


Fig. 1 IoT Architecture

Design and Implementation

A. Proposed Monitoring System:

The proposed model of the monitoring system is as shown in the figure below. The model consists of different Environmental Sensors. Initially, the Arduino MEGA is connected to aforementioned sensors to read the required data.

The data from the sensors are read at a particular interval and send to serial, from there the data is read by XBee module and further transmitted to the receiving XBee pair. This XBee starts reading the parameters of sensors like p1, p2, p3 etc. The Arduino MEGA (on the receiving end) is connected to the internet through WiFi Shield. When the connection is established the sensor data are sent to the web server and simultaneously stored in SD Card attached to WiFi Shield.

In the proposed model the Environmental Sensors is monitored. The data can be analyzed anywhere anytime.

Preferably the user can monitor the data through dedicated android app.

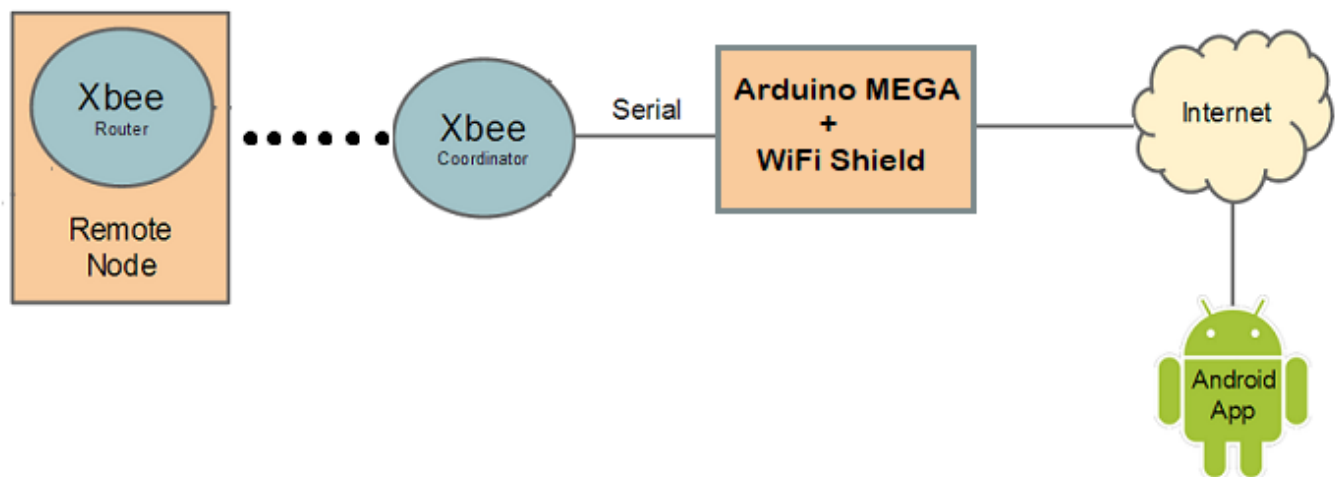


Fig. 2 Working Diagram of the project.

B. Software Design:

- **HTML Code:**

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

A simple HTML code to display live sensor's data, as well as data, is included in receiver Arduino MEGA's code which is uploaded to the server whenever a new client connects to the server.

- **Arduino IDE Code:**

Two separate codes for the two Arduino MEGA boards are required for proper functioning of the system.

- The code which sends the sensor data is fairly simple, it collects all the data from the Analog pins and encapsulates them in a single string. The collection of data is done at an interval of 20 seconds and only if the previous data has changed, it is printed to the serial monitor. The data is then transmitted by the XBee to its receiving counterpart.
- The code which receives the sensor data includes multiple libraries from the IDE i.e. SPI.h, WiFi.h, WiFiUdp.h, SD.h.
The XBee module receives the signal whenever it is available and parses it into appropriate variables. Then the program waits for any client to connect to the server. If any client is available, it sends the HTML Code along with the values from the sensor for live monitoring.

- **Android APP Code:**

An android app with a webView tool is designed to specifically access the web server hosted by WiFi Shield. Thus the app is capable of connecting and reading live data of sensor in a remote location.

There are two primary screens in the app:

- One to access the live data for monitoring and also to view complete datalog.
- Second to save the datalog or a particular data on the phone memory.

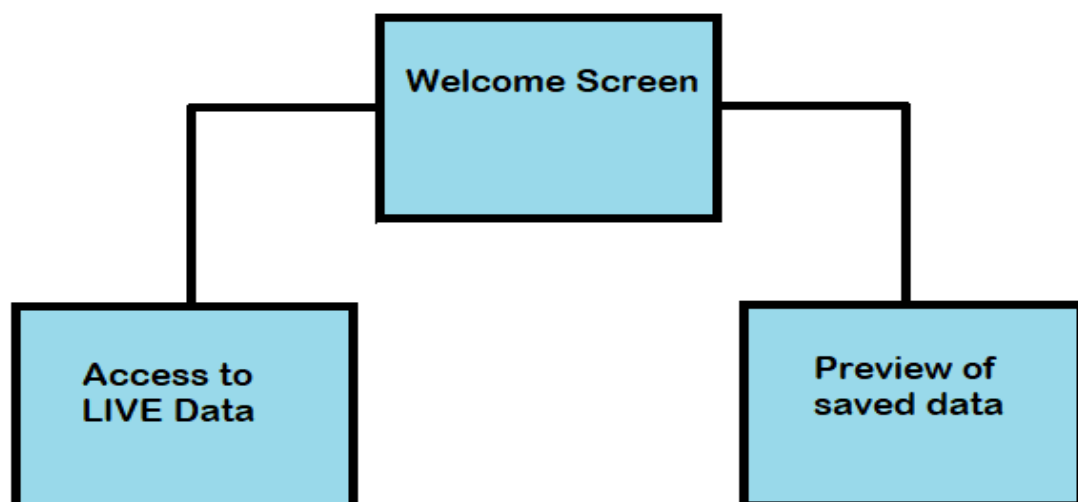


Fig. 3 Basic Layout of Android APP.

The fundamental building blocks of an app on the Android platform are Activities. They serve as the entry point for a user's interaction with an app.

Three such activity is used in our app:

1. Main Activity 2. Live Data Activity 3. Saved Data Activity

Android files contain multiple files out of which only the important XML and Java file are mentioned and explained here.

1) Main Activity:

This serves as the entry point to our app and also as a navigation page.

The options provided in this activity are: Get Live Data, Saved Screens, Exit and Credits.

Screenshot of activity along with its Java Code and XML code are mentioned below:

MainActivity.java

```
1  package com.example.varun.csir_trainee;
2
3  import android.content.Intent;
4  import android.support.v7.app.AppCompatActivity;
5  import android.os.Bundle;
6  import android.view.View;
7  import android.widget.EditText;
8  import android.widget.Toast;
9
10 public class MainActivity extends AppCompatActivity {
11
12     EditText url_f, url_l;
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         url_f = (EditText) findViewById(R.id.url_first);
18         url_l = (EditText) findViewById(R.id.url_last);
19     }
20
21
22     public void get_live_data(View view){
23         String urlf = url_f.getText().toString().trim();
24         String url_l = url_l.getText().toString().trim();
25         if((urlf.length() != 0) && (url_l.length() != 0)) {
26             String urladd = "192.168." + urlf + "." + url_l;
27             Intent i = new Intent(this, live_data.class);
28             i.putExtra("url", urladd);
29             startActivity(i);
30         }
31         else{
32             Toast.makeText(this, "Please Enter Proper URL.", Toast.LENGTH_SHORT).show();
33         }
34     }
35
36     public void get_saved_data(View view){
37         Intent i = new Intent(this, saved_data.class);
38         startActivity(i);
39     }
40
41     public void credit(View view){
42         Intent i = new Intent(this, credits.class);
43         startActivity(i);
44     }
45
46     public void exit(View view){
47         finish();
48     }
49 }
```


activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout
3  xmlns:android="http://schemas.android.com/apk/res/android"
4  xmlns:app="http://schemas.android.com/apk/res-auto"
5  xmlns:tools="http://schemas.android.com/tools"
6  android:layout_width="match_parent"
7  android:layout_height="match_parent"
8  tools:context="com.example.varun.csir_trainee.MainActivity"
9  android:background="@drawable/csirlogo">
10
11  <TextView
12      android:id="@+id/welcome_sign"
13      android:layout_width="wrap_content"
14      android:layout_height="wrap_content"
15      android:text="@string/arduino_s_data"
16      android:textAlignment="center"
17      android:textColor="@color/colorPrimaryDark"
18      android:textSize="26sp"
19      android:textStyle="bold"
20      app:layout_constraintBottom_toBottomOf="parent"
21      app:layout_constraintLeft_toLeftOf="parent"
22      app:layout_constraintRight_toRightOf="parent"
23      app:layout_constraintTop_toTopOf="parent"
24      app:layout_constraintVertical_bias="0.03" />
25
26  <LinearLayout
27      android:id="@+id/layout_button"
28      android:layout_width="0dp"
29      android:layout_height="wrap_content"
30      android:orientation="horizontal"
31      app:layout_constraintLeft_toLeftOf="parent"
32      app:layout_constraintRight_toRightOf="parent"
33      app:layout_constraintTop_toBottomOf="@+id/layout_url">
34
35      <Button
36          android:id="@+id/live_data"
37          android:layout_width="wrap_content"
38          android:layout_height="wrap_content"
39          android:backgroundTint="@color/button_background"
40          android:layout_weight="1"
41          android:onClick="get_live_data"
42          android:text="@string/get_live_data"
43          android:textSize="18sp"
44          android:textStyle="bold"
45          app:layout_constraintLeft_toLeftOf="parent"
46          tools:ignore="ButtonStyle" />
47
48      <Button
49          android:id="@+id/saved_data"
50          android:layout_width="wrap_content"
51          android:layout_height="wrap_content"
52          android:backgroundTint="@color/button_background"
53          android:layout_weight="1"
54          android:onClick="get_saved_data"
55          android:text="@string/get_saved_data"
56          android:textSize="18sp"
57          android:textStyle="bold"
58          app:layout_constraintRight_toRightOf="parent"
59          tools:ignore="ButtonStyle" />
60
61  </LinearLayout>
62
63  <LinearLayout
64      android:id="@+id/layout_url"
65      android:layout_width="wrap_content"
66      android:layout_height="wrap_content"
67      android:orientation="horizontal"
68      app:layout_constraintLeft_toLeftOf="parent"
69      app:layout_constraintBottom_toBottomOf="parent"
70      app:layout_constraintTop_toTopOf="parent"
71      app:layout_constraintVertical_bias="0.70">
72
73      <TextView
```

```
74         android:id="@+id/url_initial"
75         android:layout_width="wrap_content"
76         android:layout_height="wrap_content"
77         android:text="@string/enter_url_192_168"
78         android:textColor="@android:color/black"
79         android:textSize="16sp"
80         android:textStyle="bold" />
81
82     <EditText
83         android:id="@+id/url_first"
84         android:layout_width="wrap_content"
85         android:layout_height="wrap_content"
86         android:background="@color/white"
87         android:hint="   xxx   "
88         android:inputType="number"
89         android:textAlignment="center"
90         android:textSize="18sp"
91         android:textStyle="bold"
92         tools:ignore="HardcodedText,TextFields" />
93
94     <TextView
95         android:id="@+id/url_dot"
96         android:layout_width="wrap_content"
97         android:layout_height="wrap_content"
98         android:text="@string/dot"
99         android:textColor="@android:color/black"
100        android:textSize="16sp"
101        android:textStyle="bold" />
102
103     <EditText
104         android:id="@+id/url_last"
105         android:layout_width="wrap_content"
106         android:layout_height="wrap_content"
107         android:background="@color/white"
108         android:hint="   xxx   "
109         android:inputType="number"
110         android:textAlignment="center"
111         android:textSize="18sp"
112         android:textStyle="bold"
113         tools:ignore="HardcodedText,TextFields" />
114
115 </LinearLayout>
116
117 <Button
118     android:id="@+id/exit"
119     android:layout_width="wrap_content"
120     android:layout_height="wrap_content"
121     android:backgroundTint="@color/button_background"
122     android:onClick="exit"
123     android:text="@string/exit"
124     android:textAllCaps="false"
125     app:layout_constraintBottom_toBottomOf="parent"
126     app:layout_constraintLeft_toLeftOf="parent" />
127
128 <Button
129     android:id="@+id/credits"
130     android:layout_width="wrap_content"
131     android:layout_height="wrap_content"
132     android:backgroundTint="@color/button_background"
133     android:onClick="credit"
134     android:text="@string/main_credits"
135     android:textAllCaps="false"
136     app:layout_constraintBottom_toBottomOf="parent"
137     app:layout_constraintRight_toRightOf="parent" />
138 </android.support.constraint.ConstraintLayout>
```

2) Live Data Activity:

This activity is used for live monitoring of data sent by Arduino to the web server.

An option to save the data is also provided in this activity.

Screenshot of activity along with its Java Code and XML code are mentioned below:

live_data.java

```
1  package com.example.varun.csir_trainee;
2
3  import android.content.Intent;
4  import android.graphics.Bitmap;
5  import android.net.Uri;
6  import android.support.v7.app.AppCompatActivity;
7  import android.os.Bundle;
8  import android.view.KeyEvent;
9  import android.view.View;
10 import android.webkit.DownloadListener;
11 import android.webkit.WebView;
12 import android.webkit.WebViewClient;
13 import android.widget.Button;
14 import android.widget.ImageButton;
15 import android.widget.LinearLayout;
16 import android.widget.ProgressBar;
17 import android.widget.TextView;
18 import android.widget.Toast;
19 import java.io.File;
20 import java.text.DateFormat;
21 import java.util.Date;
22
23 public class live_data extends AppCompatActivity {
24
25     private WebView mWebView;
26     String currentDateTimeString = "";
27     ProgressBar p;
28     TextView connection;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_live_data);
34         Intent intent = getIntent();
35         String url;
36         url=intent.getStringExtra("url");
37         url="http://" +url;
38
39         Thread t = new Thread() {
40
41             @Override
42             public void run() {
43                 try {
44                     while (!isInterrupted()) {
45                         runOnUiThread(new Runnable() {
46                             @Override
47                             public void run() {
48                                 updateTime();
49                             }
50                         }); // updating time every 5 sec
51                         Thread.sleep(5000);
52                     }
53                 } catch (InterruptedException e) {
54                     e.printStackTrace();
55                 }
56             }
57         };
58
59         t.start();
60         p = (ProgressBar) findViewById(R.id.progressBar);
61         connection = (TextView) findViewById(R.id.connection);
62         p.setVisibility(View.VISIBLE);
63         connection.setVisibility(View.VISIBLE);
```

```
64
65     mWebView = (WebView) findViewById(R.id.webView);
66     mWebView.setWebViewClient(new MyBrowser());
67     mWebView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
68     connection.setText("Connecting to: "+url);
69     mWebView.loadUrl(url);
70     mWebView.setDownloadListener(new DownloadListener() {
71         @Override
72         public void onDownloadStart(String url, String userAgent, String
contentDisposition, String mimetype, long contentLength) {
73             //download file using web browser
74             Intent i = new Intent(Intent.ACTION_VIEW);
75             i.setData(Uri.parse(url));
76             startActivity(i);
77         }
78     });
79
80     ImageButton refresh = (ImageButton) findViewById(R.id.refresh);
81     refresh.setOnClickListener(new View.OnClickListener() {
82         @Override
83         public void onClick(View v) {
84             mWebView.reload();
85         }
86     });
87 }
88
89 private class MyBrowser extends WebViewClient {
90     @Override
91     public boolean shouldOverrideUrlLoading(WebView view, String url) {
92         view.loadUrl(url);
93         return true;
94     }
95     @Override
96     public void onPageFinished(WebView view, String url) {
97         p.setVisibility(View.GONE);
98         connection.setVisibility(View.GONE);
99     }
100 }
101
102 public boolean onKeyDown(int keyCode, KeyEvent event) {
103     if ((keyCode == KeyEvent.KEYCODE_BACK) && mWebView.canGoBack()) {
104         mWebView.goBack(); // Go to previous page
105         return true;
106     }
107     // Use this as else part
108     return super.onKeyDown(keyCode, event);
109 }
110
111
112 private void updateTime(){
113     TextView datetime = (TextView) findViewById(R.id.datetime);
114
115     currentDateTimeString = DateFormat.getDateTimeInstance().format(new Date());
116     datetime.setText("Log Time: " + currentDateTimeString);
117 }
118
119 public void backbutton(View view) { finish(); }
120
121
122
123 public void log_data(View view) {
124     Button back = (Button) findViewById(R.id.back_button_live);
125     Button logger = (Button) findViewById(R.id.log_data);
126     logger.setVisibility(View.INVISIBLE);
127     back.setVisibility(View.INVISIBLE);
128     Bitmap b = null;
129     LinearLayout layout = (LinearLayout) findViewById(R.id.layout_up);
130     b = ScreenshotUtils.getScreenShot(layout);
131
132     if (b != null) {
133         File saveFile = ScreenshotUtils.getMainDirectoryName(this);
134         File file = ScreenshotUtils.store(b, currentDateTimeString+".png", saveFile);
135         Toast.makeText(this, R.string.save_done, Toast.LENGTH_SHORT).show();
136     } else
137         //If bitmap is null show toast message
138         Toast.makeText(this, R.string.save_fail, Toast.LENGTH_SHORT).show();
139     logger.setVisibility(View.VISIBLE);
140     back.setVisibility(View.VISIBLE); }}
```

```
activity_live_data.xml
1  <?xml version="1.0" encoding="utf-8" ?>
2  <android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/root_content"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context="com.example.varun.csir_trainee.live_data">
9
10     <LinearLayout
11         android:id="@+id/layout_up"
12         android:layout_width="0dp"
13         android:layout_height="0dp"
14         android:orientation="vertical"
15         app:layout_constraintTop_toTopOf="parent"
16         app:layout_constraintBottom_toTopOf="@+id/layout_bottom"
17         app:layout_constraintLeft_toLeftOf="parent"
18         app:layout_constraintRight_toRightOf="parent">
19
20         <TextView
21             android:id="@+id/datetime"
22             android:layout_width="match_parent"
23             android:layout_height="wrap_content"
24             android:text="@string/date_and_time_appears_here"
25             android:textColor="@color/colorAccent1"
26             android:textSize="18sp"
27             android:textStyle="bold"
28             app:layout_constraintBottom_toTopOf="@+id/webView"
29             app:layout_constraintTop_toTopOf="@+id/layout_up" />
30
31         <TextView
32             android:id="@+id/connection"
33             android:layout_width="match_parent"
34             android:layout_height="wrap_content"
35             android:text="@string/connection_status"
36             android:visibility="gone" />
37
38         <ProgressBar
39             android:id="@+id/progressBar"
40             style="?android:attr/progressBarStyle"
41             android:layout_width="match_parent"
42             android:layout_height="wrap_content"
43             android:visibility="gone"
44             app:layout_constraintBottom_toTopOf="@+id/webView"
45             app:layout_constraintTop_toBottomOf="@+id/datetime" />
46
47         <WebView
48             android:id="@+id/webView"
49             android:layout_width="match_parent"
50             android:layout_height="match_parent"
51             app:layout_constraintBottom_toTopOf="@+id/layout_bottom"
52             app:layout_constraintLeft_toLeftOf="parent"
53             app:layout_constraintRight_toRightOf="parent"
54             app:layout_constraintTop_toBottomOf="@+id/datetime">
55
56         </WebView>
57
58     </LinearLayout>
59
60     <LinearLayout
61         android:id="@+id/layout_bottom"
62         android:layout_width="0dp"
63         android:layout_height="wrap_content"
64         android:orientation="horizontal"
65         app:layout_constraintBottom_toBottomOf="parent"
66         app:layout_constraintLeft_toLeftOf="parent"
67         app:layout_constraintRight_toRightOf="parent">
68
69         <Button
70             android:id="@+id/back_button_live"
71             android:layout_width="wrap_content"
72             android:layout_height="wrap_content"
73             android:layout_weight="1"
74             android:backgroundTint="@color/button_background"
```

```
75         android:onClick="backbutton"
76         android:text="@string/back_live"
77         android:textAllCaps="false"
78         app:layout_constraintBottom_toBottomOf="parent"
79         app:layout_constraintLeft_toLeftOf="parent"
80         tools:ignore="ButtonStyle" />
81
82     <ImageButton
83         android:id="@+id/refresh"
84         android:layout_width="wrap_content"
85         android:layout_height="wrap_content"
86         android:layout_weight="1"
87         android:backgroundTint="@color/refresh_background"
88         app:layout_constraintBottom_toBottomOf="parent"
89         app:layout_constraintLeft_toRightOf="@+id/back_button_live"
90         app:layout_constraintRight_toLeftOf="@+id/log_data"
91         app:srcCompat="@android:drawable/ic_menu_rotate"
92         android:contentDescription="@string/refresh_button" />
93
94     <Button
95         android:id="@+id/log_data"
96         android:layout_width="wrap_content"
97         android:layout_height="wrap_content"
98         android:layout_weight="1"
99         android:backgroundTint="@color/button_background"
100        android:onClick="log_data"
101        android:text="@string/save_this_data"
102        android:textAllCaps="false"
103        app:layout_constraintBottom_toBottomOf="parent"
104        app:layout_constraintRight_toRightOf="parent"
105        tools:ignore="ButtonStyle" />
106
107 </LinearLayout>
108
109 </android.support.constraint.ConstraintLayout>
```

3) Saved Data Activity:

This activity is used to view data saved by user in phone memory.

Screenshot of activity along with its Java Code and XML code are mentioned below:

```
saved_data.java
1  package com.example.varun.csir_trainee;
2
3  import android.content.Intent;
4  import android.graphics.Bitmap;
5  import android.graphics.BitmapFactory;
6  import android.support.v7.app.AppCompatActivity;
7  import android.os.Bundle;
8  import android.view.View;
9  import android.widget.AdapterView;
10 import android.widget.AdapterView.OnItemClickListener;
11 import android.widget.ArrayAdapter;
12 import android.widget.ImageView;
13 import android.widget.Spinner;
14 import android.widget.Toast;
15 import java.io.File;
16 import java.util.ArrayList;
17 import java.util.List;
18
19 public class saved_data extends AppCompatActivity implements
AdapterView.OnItemClickListener{
20
21     File listFile;
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_saved_data);
26
27         Spinner spinner = (Spinner) findViewById(R.id.spinner);
28         spinner.setOnItemSelectedListener(this);
29
30         List<String> DateTime = new ArrayList<String>();
31         DateTime.add("Choose Date and Time Here.");
32
33         listFile = ScreenshotUtils.getMainDirectoryName(this);
34         File[] files = listFile.listFiles();
35
36         for(int i=(files.length-1); i>=0; i--){
37             DateTime.add(files[i].getName());
38         }
39         ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this,
R.layout.support_simple_spinner_dropdown_item, DateTime);
40
41         dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
42         spinner.setAdapter(dataAdapter);
43     }
44
45     public void onItemSelected(AdapterView<?> parent, View view, int pos, long id){
46         String item = parent.getItemAtPosition(pos).toString();
47
48         ImageView dataView = (ImageView) findViewById(R.id.dataView);
49         File imgFile = new File(listFile+"/"+item);
50         Bitmap bmp = BitmapFactory.decodeFile(imgFile.getAbsolutePath());
51
52         if(imgFile.exists()){
53             dataView.setImageBitmap(bmp);
54         }
55         else{
56             dataView.setImageBitmap(null);
57         }
58         if(!item.equals("Choose Date and Time Here.)){
59             Toast.makeText(parent.getContext(), "Selected: " + item,
Toast.LENGTH_SHORT).show();
60         }
61     }
62
63     public void onNothingSelected(AdapterView<?> arg0){ }
64
65     public void backButtonsaved(View view) {
66         finish();
67     }
68 }
```

activity_saved_data.xml

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <android.support.constraint.ConstraintLayout "
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context="com.example.varun.csir_trainee.saved_data">
8
9      <LinearLayout
10         android:id="@+id/layout_up_saved"
11         android:layout_width="0dp"
12         android:layout_height="0dp"
13         android:orientation="vertical"
14         app:layout_constraintTop_toTopOf="parent"
15         app:layout_constraintBottom_toTopOf="@+id/layout_bottom_saved"
16         app:layout_constraintLeft_toLeftOf="parent"
17         app:layout_constraintRight_toRightOf="parent">
18
19         <Spinner
20             android:id="@+id/spinner"
21             android:layout_width="match_parent"
22             android:layout_height="wrap_content"
23             android:backgroundTint="@color/colorPrimary"
24             app:layout_constraintBottom_toTopOf="@+id/dataView"
25             app:layout_constraintTop_toTopOf="@+id/layout_up_saved" />
26
27         <ImageView
28             android:id="@+id/dataView"
29             android:layout_width="match_parent"
30             android:layout_height="match_parent"
31             android:background="@color/saved_background"
32             app:layout_constraintBottom_toTopOf="@+id/layout_bottom_saved"
33             app:layout_constraintLeft_toLeftOf="parent"
34             app:layout_constraintRight_toRightOf="parent"
35             app:layout_constraintTop_toBottomOf="@+id/spinner" />
36     </LinearLayout>
37
38     <LinearLayout
39         android:id="@+id/layout_bottom_saved"
40         android:layout_height="wrap_content"
41         android:background="@color/saved_background"
42         android:orientation="horizontal"
43         app:layout_constraintBottom_toBottomOf="parent"
44         app:layout_constraintLeft_toLeftOf="parent"
45         app:layout_constraintRight_toRightOf="parent">
46
47         <Button
48             android:id="@+id/back_button_saved"
49             android:layout_width="0dp"
50             android:layout_height="wrap_content"
51             android:layout_weight="1"
52             android:backgroundTint="@color/button_background"
53             android:onClick="backbuttonsaved"
54             android:text="@string/back"
55             android:textAllCaps="false"
56             android:textStyle="bold"
57             app:layout_constraintBottom_toBottomOf="parent"
58             app:layout_constraintLeft_toLeftOf="parent"
59             tools:ignore="ButtonStyle" />
60     </LinearLayout>
61 </android.support.constraint.ConstraintLayout>
```


C. Hardware Design:

The hardware components consist of two Arduino MEGA.

1. Arduino which send data using XBee.

An Arduino MEGA board connected to the sensors (Potentiometer and LM35) and a sender XBee. This Arduino is used to read sensor data.

All the sensor reading are combined in single String format and transmitted using serial.print() function by the XBee module.

The data is programmed to be sent at interval of 30 second only if data has changed.

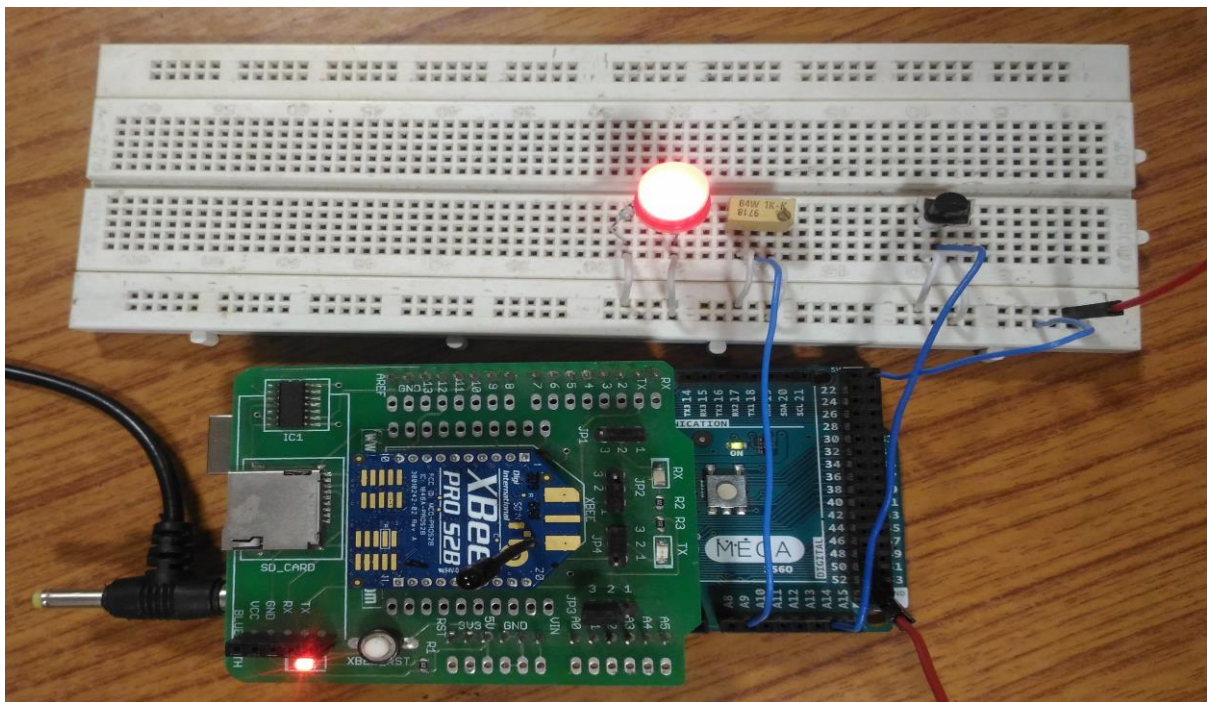


Fig. 4 The actual circuit used for a demonstration with XBee, Potentiometer and LM35.

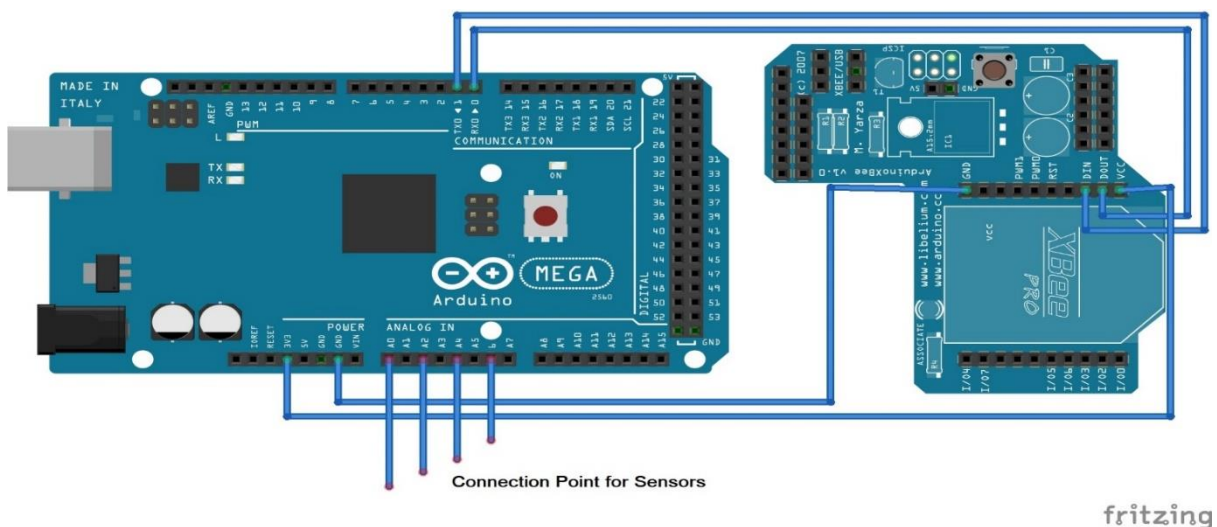


Fig. 5 Circuit Diagram for above figure

2. Arduino which receive data using XBee.

The other board is attached to a WiFi Shield and XBee module. This XBee module acts as a receiver and collects the received data in String format using `serial.read()` function. The data received is parsed into separated assigned variables, stored in the log file (in SD Card) in a predefined format and also sent to the web server hosted by WiFi Shield for the user to monitor.

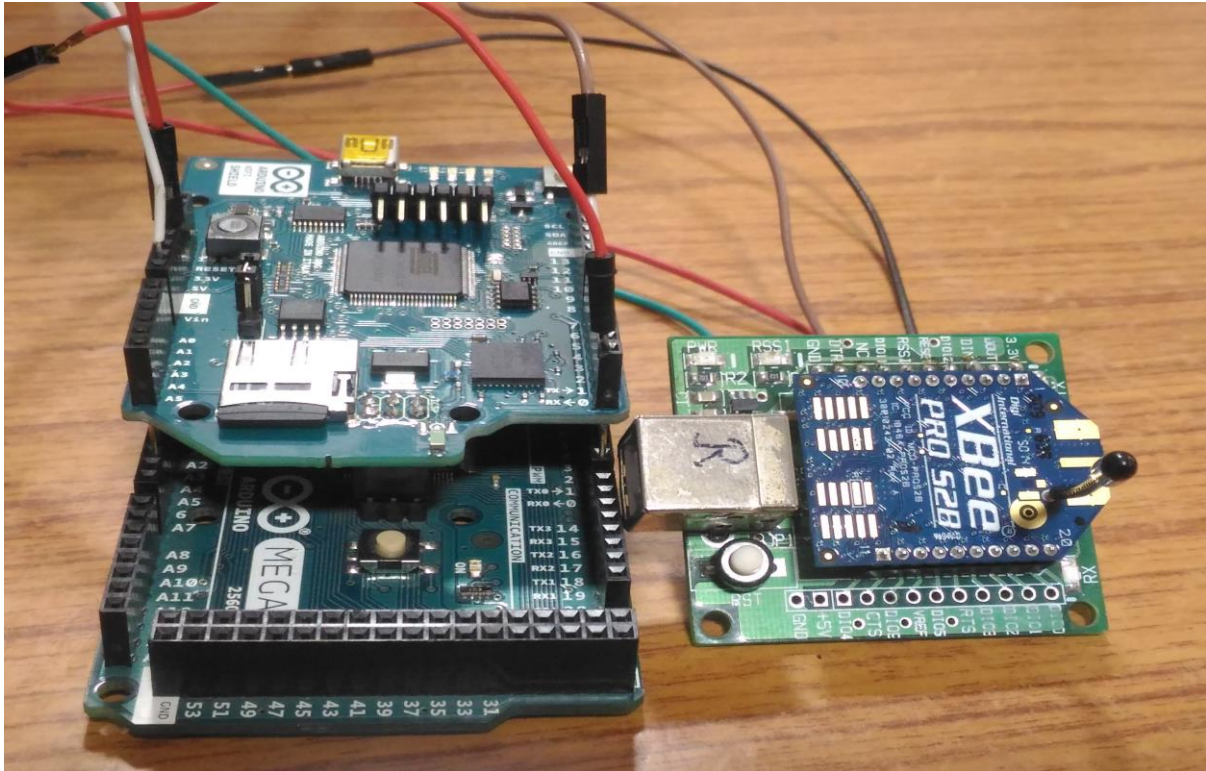


Fig. 6 The actual circuit used for a demonstration with XBee and WiFi Shield.

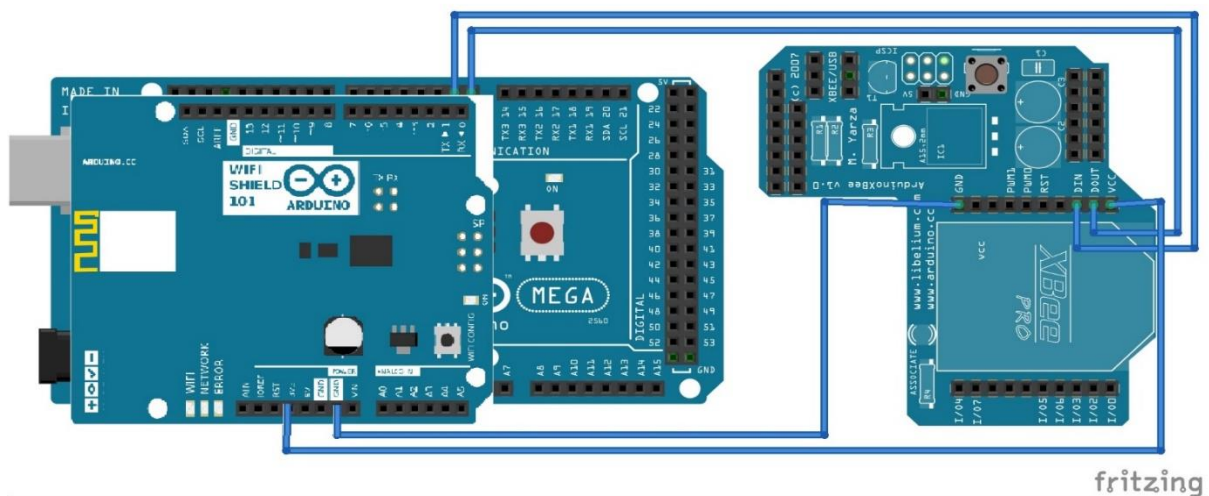


Fig. 7 Circuit Diagram for above figure

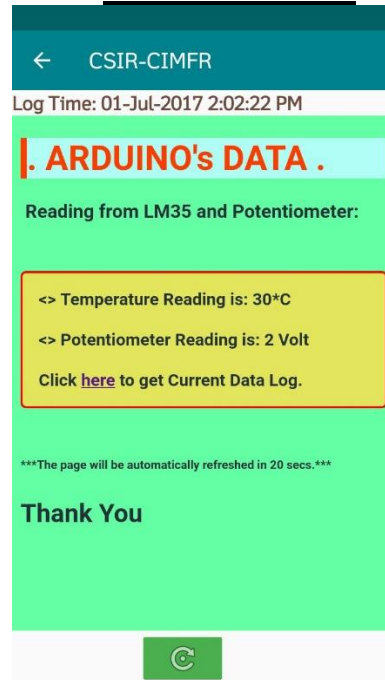
D. Result:

For the demonstration of proper working of the project, LM35 and a potentiometer is used.

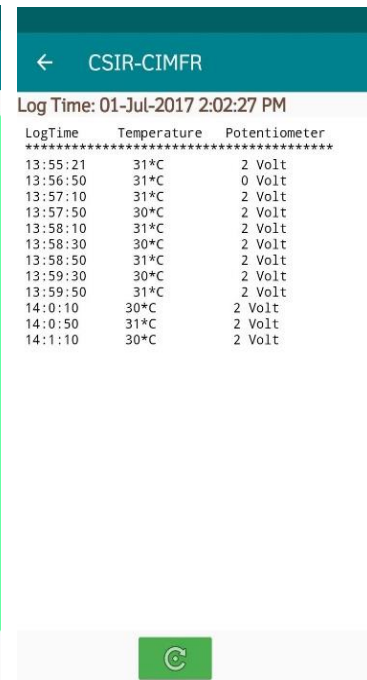
Main Screen:



GET LIVE DATA:



◁ Live Monitoring Screen

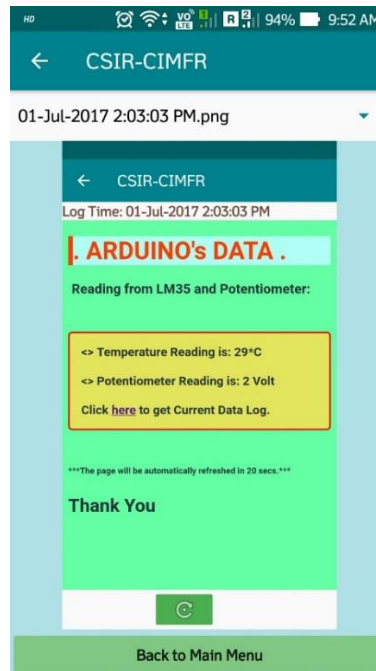


◁ Complete Log of Data

SAVED SCREENS:



◁ Choose Date and Time



◁ Saved Screen of Single Data



◁ Saved Screen of Complete Log

Along with the saved screen the data is being simultaneously stored in the attached SD Card.

Conclusion

The monitoring system using Internet of Things has been experimentally proven to work satisfactorily by connecting simple sensors to it and the readings were successfully monitored remotely through intranet. The designed system not only monitors the data from various Environmental Sensors but also stores the data regularly according to the requirement and format.

- **Future Work:**

Using this system as a framework, the system can be expanded to include various other options which could include home automation features like switching on the light when it gets dark, home security features like capturing the photo of a person moving around the house and storing it onto the cloud. This will reduce the data storage than using the CCTV camera which will record all the time and stores it. The system can be expanded for energy monitoring or weather stations. This kind of a system with respective changes can be also implemented in the hospitals for disable people or in industries where the human invasion is impossible or dangerous, and it can also be implemented for environmental monitoring.

The scope of Internet of Things is only limited by the imagination of people working with it.

References

- https://en.wikipedia.org/wiki/Internet_of_things
- https://tutorialspoint.com/arduino/arduino_quick_guide.htm/
- <https://forum.arduino.cc>
- <https://arduino.stackexchange.com/>
- <http://www.instructables.com/id/How-to-connect-your-Arduino-WiFi-shield-to-a-custo/>
- <http://electronicsforu.com/electronics-projects/humidity-temperature-monitoring-using-arduino-iot>
- <https://www.arduino.cc/en/Main/ArduinoWiFiShield>
- <http://www.microelectronicos.com/datasheets/ArduinoWiFiShield.pdf>
- <https://www.arduino.cc/en/Main/ArduinoXbeeShield>
- <https://www.digi.com/support/forum/41668/receiving-data-via-xbee-serial-port-buffer-overflow>
- <http://www.mantech.co.za/datasheets/products/A000047.pdf>
- <http://www.microelectronicos.com/datasheets/ArduinoWiFiShield.pdf>
- https://cdn.sparkfun.com/datasheets/Wireless/Zigbee/ds_xbeezbmodules.pdf