

A Smart Approach On The Collecting Working Condition Data From Home Appliances Under The Field Test

Teliki Sainath ,Vaggu Ganesh, Venugonda Shameerbasha ,Teliki Sainath, Vaggu Ganesh
Muthayammal Engineering College

Abstract- This research presents an intelligent method for managing and collecting operational data from home appliances during the field test process. The proposed approach involves deploying sensors to gather device-specific and environmental data, which are then systematically evaluated to assess the working conditions of each appliance. The data management system ensures efficient storage, interpretation, and transmission of information, enabling real-time monitoring and analysis. The methodology not only facilitates predictive maintenance and energy management but also supports the identification of possible faults and optimization of appliance performance in actual user environments. The smart approach contributes to the advancement of smart home systems by integrating IoT-based sensor networks and data analytics to improve reliability, energy efficiency, and user experience.

Keywords— Smart home systems, IoT-based sensors, appliance monitoring, operational data collection, real-time analysis, predictive maintenance, energy efficiency, fault detection, data management system, environmental sensing, performance optimization, smart appliances, data analytics, connected devices, user experience enhancement.

I. CHAPTER 1 INTRODUCTION

GENERAL

Modern smart homes are equipped with internet-connected appliances and sensors that allow seamless, remote monitoring and management of device status, performance, and energy consumption. The core idea is to create a connected network where each appliance can communicate its operational state such as on/off events, performance metrics, and error conditions to a central home automation hub or cloud-based server.

This system typically consists of :

IoT-enabled sensors embedded within or attached to each appliance for continuous data acquisition.

A home automation controller or gateway that aggregates and processes incoming data from distributed household devices.

Software interfaces (e.g., mobile apps or web dashboards) that provide users and field engineers

with access to real-time and historical appliance data
Convert the exponent to binary and pad with zeros to the left if necessary to match the size.

Recommend a data management and privacy plan for collected data :

Define clear policies and procedures for data collection, storage, access, and usage, ensuring compliance with relevant regulations like GDPR or local data protection laws

Establish processes for data retention, archival, and secure deletion. Only store data as long as necessary for intended purposes.

Objective

The core aim is to create a non-intrusive, sensor-based architecture that monitors the working conditions (such as voltage, current, energy consumption, and operational states) of

various appliances, providing real-time data for predictive maintenance, usage optimization, and intelligent control.

Problem Statement

Despite the increasing adoption of smart home appliances, there is a significant challenge in effectively and accurately collecting real-time working condition data from these appliances under actual field test conditions. Traditional data collection methods are often manual, intrusive, or limited to controlled laboratory environments.

Existing System

Existing systems for collecting working condition data from home appliances typically focus on IoT-based smart home monitoring frameworks.

These systems utilize sensor modules attached to appliances to measure parameters such as energy consumption, voltage, current, and operational status in real time.

Data are transmitted wirelessly using protocols like Wi-Fi, Zigbee, or Bluetooth to central servers or cloud platforms for analysis and visualization.

Traditional field testing involves an operator manually checking the appliance, creating bug lists, and physically returning the product for problem-solving, which is a time-consuming and expensive process. The smart approach leverages modern technology to automate this process, allowing for real-time monitoring and remote diagnostics.

The smart approach described in a study by Vestel Electronics Company R&D department and Manisa Celal Bayar University relies on a specific system architecture to gather comprehensive data.

This approach collects data such as water temperature, door open/close counts, motor RPM, power parameters, vibration levels, and ambient conditions, allowing for a comprehensive understanding of real-world usage and environmental effects on product performance. The

data is then used to train machine learning models for predictive maintenance and anomaly detection.

A smart approach to collecting working condition data from home appliances during field tests involves using an Internet of Things (IoT)-based system with a dedicated data acquisition "test box" that transmits data via a reliable network like GSM to a cloud server for real-time analysis and remote management. The "Matter" protocol is a unifying standard for smart home devices that can facilitate this process in modern appliances.

II. CHAPTER 2 LITERATURE SURVEY

Title: Optimizing smart home appliance energy monitoring using Factorial Hidden Markov Models for Internet of Behaviour.

Author: Siqi Liu, Zhiyuan Xie, Zhongwei Hu

Year: 2024

Description: This work explores enhancing energy management in smart homes through Non Intrusive Load Monitoring (NILM) using advanced probabilistic models for disaggregating energy use and linking it with user behaviour analysis to improve power effectiveness.

Title: A Study on Smart Home Appliances Based on Artificial Intelligence

Author: Ok Hue Cho

Year: 2024

Description: The study reviews smart home appliances using AI for automation, energy efficiency, and remote operation.

Title: Literature Review on Home Automation System

Author: Neha Malik, Yogita Bo wade

Year: 2017

Description: This review covers various home automation technologies and communication protocols such as Wi-Fi, ZigBee, and RF.

III. CHAPTER 3 EXISTING SYSTEM

General

A Floating point number flow chart refers to the algorithmic steps of manipulating floatingpoint numbers, such as addition, subtraction, normalization, and overflow detection, which are fundamental in computing. This includes steps like checking zero values, aligning mantissas, performing operations, normalizing results, and handling overflows.

Module Seperation

Sensing Module
Data Acquisition Module
Data Processing Module
Communication/Transmission Module
Data Storage Module
Analytics & Monitoring Module
User Interface Module

Module Explanation

This modular architecture allows efficient, scalable, and intelligent data collection from home appliances, ensuring optimized performance during field testing by integrating sensing, processing, control, communication, and user interaction seamlessly.

Uses machine learning algorithms such as neural networks or reinforcement learning to analyze data trends, predict appliance performance, detect anomalies, and determine optimal operational settings.

Exponential Operation

The primary goal during planning is to define clear objectives for the field test, including understanding appliance performance, energy consumption, operating conditions, and safety compliance. This involves selecting relevant appliances—such as electric cookers, kettles, or washing machines that reflect typical consumer use.

The planning stage ensures the test setup mimics real-life usage patterns as closely as possible, taking into account variables like ambient temperature, supply voltage fluctuations, and user operation variability.

It is essential to incorporate safety checks, such as electrical insulation, earth continuity,

leakage current tests, to ensure the safety of both the appliances and the testers A detailed test protocol must be established, including which parameters to measure—such as voltage, current, power consumption, temperature, and operational cycles—along with the timing, duration, and conditions of the test

Initial value of the exponent should be the larger of the 2 number since we know exponent.

Methodology

Clarify what working condition parameters of home appliances need to be collected under field test conditions.

Establish goals such as monitoring appliance health, usage patterns, and predictive maintenance potential.

Identify appropriate sensors and IoT devices capable of measuring relevant parameters like voltage, current, temperature, vibration, etc.

Choose communication protocols and data acquisition systems suitable for real-time or periodic data collection in home environments.

A specialized architecture designed to handle large-scale telemetry while managing the exponential growth of data generated by high-frequency sensors.

Each Device Under Test (DUT) is paired with a test box containing internal sensors and a communication module.

Using GSM is often preferred over Wi-Fi for field tests because of its wider coverage, ensuring data

transmission from remote or diverse environmental locations.

Research Gate

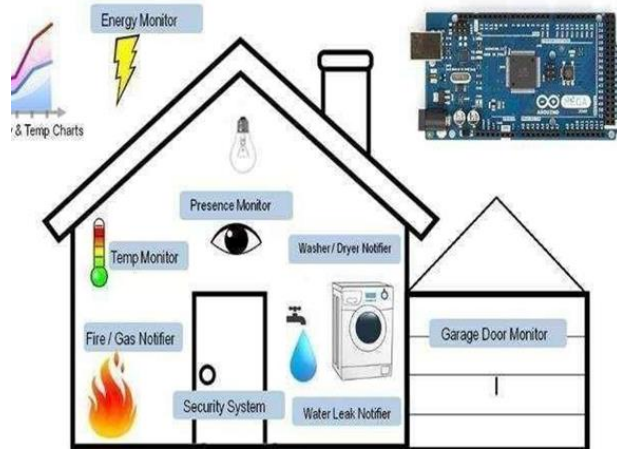


Fig.3.1 Research Gate

Additional Information

Field Test is one of the most important point to learn the working conditions of products with real user profiles.

To get product Exponents, the Input Exponents are added together.

Since we use bias in the Floating Point Format Exponent, we need to subtract the bias from the Sum of Exponents to get the actual exponent.

The Mantissa Multiplication operation is much more that the exponential Addition. Because this field data is necessary to inform R&D department.

Field Test Setup and Configuration Measurement Parameters (Voltage, Current, Temperature, etc.)

User Interface

The stored data at cloud server can be controlled, listed, plotted as a graph, and can be input for other systems.

All these functions can be realized by user interface of the system shown This normalization can lead to a correction of the result's exponent.

The state of the system, online and offline device counts, urgent notifications, system call- messages, incoming data size can be monitored by the user interface.

At the right-hand size, all instant notifications are listed. At the middle of the screen, clicking device state section, all device list can be view

Cloud Server

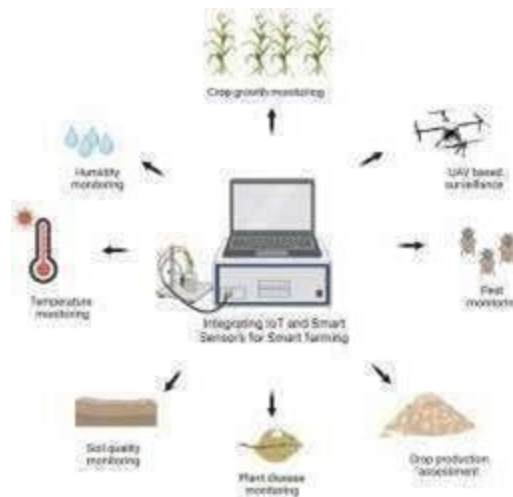


Figure .3.2 Cloud Server

Sensors or smart modules integrated with home appliances transmit real-time operational data (such as voltage, current, temperature, vibration) to a local gateway or edge device.

This data is securely transmitted over the internet to cloud servers via reliable communication protocols, often using wireless networks or IoT-specific connectivity solutions.

Data Storage

Cloud servers provide scalable, large-volume storage capacity to hold historical and real-time data, supporting continuous monitoring.

Cloud computing technology to store, process, and analyze data collected from appliances remotely.

Data Processing

Cloud - based platforms leverage powerful computing resources to process, analyze, and visualize the data using machine learning algorithms, fault detection models, and predictive analytics.

Collect sensor streams (voltage, current, power, temperature, vibration, status) from multiple devices and gateways. Normalize units, synchronize timestamps, and handle missing data to create a consistent dataset for analysis.

Feature Engineering

Derive meaningful features from raw signals: power metrics (active/reactive power, power factor), energy consumption over intervals, ramp rates, temperature trends, vibration spectra, and event counts (on/off cycles, faults)

Create time-based features (hour-of-day, day-of-week) to capture usage patterns and seasonal effects.

Outlier detection and correction, noise filtering (e.g., low-pass filters for analog signals), and calibration adjustments. Handling data gaps via interpolation or imputation strategies appropriate to the domain.

Remote Monitoring & Control

Users or technicians access the system through web or mobile interfaces, enabling remote monitoring, diagnostics, and control of appliances.

Communication layer: protocols and networks (Wi-Fi, MQTT over TLS, cellular, Zigbee, or LoRaWAN) transport data and commands between edge devices and the cloud or on- premises servers.

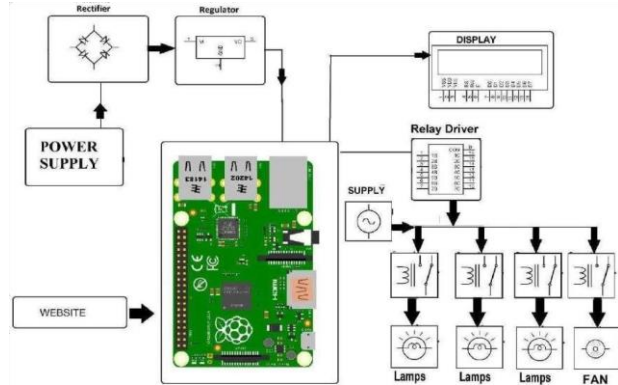


Fig.3.3 Remote monitoring Data security:

Field-testing in real homes raises significant privacy concerns. Encryption, secure user authentication, and data minimization are critical to protect sensitive user information.

Consent and transparency:

Manufacturers must be transparent with field-test participants about what data is being collected and how it will be used. User consent must be actively obtained.

Proposed System

The proposed system for collecting working condition data from home appliances under field test conditions uses a smart IoT-based approach with non-intrusive current and voltage sensors connected to a microcontroller (e.g., Arduino Uno). The sensors continuously monitor electrical parameters like current and voltage of appliances without disrupting normal operation.

Proposed System Block Diagram

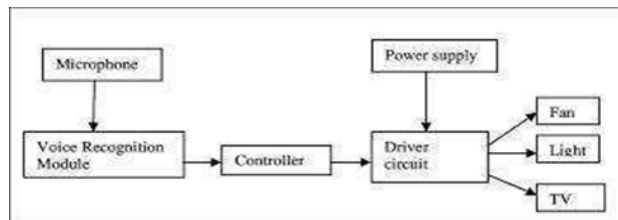


Fig.3.4 Proposed System

A smart approach to collecting working condition data from home appliances during a field test involves a proposed system that uses Internet of Things (IoT) connectivity and cloud-based analytics [1]. This system facilitates continuous, real-time data acquisition without needing constant manual intervention.

If issues are detected, some fixes or data collection parameters might be updated remotely via the system, enhancing the flexibility of the field test.

By leveraging this integrated IoT system, manufacturers gain a comprehensive and authentic view of how their appliances perform in real-world scenarios, leading to better product design and reliability.

Transparency :

This approach collects data such as water temperature, door open/close counts, motor RPM, parameters, vibration levels, and ambient conditions, allowing for a comprehensive understanding of real-world usage and environmental effects on product performance. The data is then used to train machine learning models for predictive maintenance and anomaly detection.

IV. CHAPTER 4 SYSTEM DESIGN AND ARCHITECTURE INTRODUCTION

This chapter describes the overall system design and architecture for the smart data collection framework proposed for monitoring the working conditions of home appliances under field test conditions. The system is designed to collect, process, store, and analyze operational data from various appliances deployed in real household environments. The architecture aims to ensure reliability, scalability, and real-time visibility into appliance performance while maintaining data security and user privacy.

Microcontroller / Embedded System: Controls sensor operation and preprocesses raw data (e.g., ESP32, STM32, or Raspberry Pi).

The proposed system integrates IoT-enabled sensors, edge computing units, and a cloud-based data management platform. The overall data flow begins with the appliance's embedded sensors capturing operational parameters (e.g., temperature, voltage, current, vibration, and usage duration), followed by preprocessing at the edge node and transmission to the central cloud platform for analysis and visualization.

Data Flow Process

Sensors collect operational parameters from appliances.

Microcontroller preprocesses and packages data into a standard JSON format.

Edge gateway transmits encrypted data packets to the cloud.

Cloud services store, process, and analyze incoming data streams.

Communication Protocols

To ensure reliable communication, the system uses lightweight and secure IoT protocols:

For low-latency data transmission between devices and cloud brokers.

For configuration and firmware updates.

Optional protocol for constrained devices in low-power environments.

Ensures data confidentiality and integrity during transmission.

The proposed system integrates IoT-enabled sensors, edge computing units, and a cloud-based data management platform.

The overall data flow begins with the appliance's embedded sensors capturing operational parameters (e.g., temperature, voltage, current, vibration, and usage duration), followed by preprocessing at the edge node and transmission to

the central cloud platform for analysis and visualization.

Smart Appliance Unit

Applies machine learning and rule-based algorithms for condition monitoring, anomaly detection, and predictive maintenance.

Web-based or mobile interface for engineers and researchers to monitor real-time appliance behaviour.

To ensure reliable communication, the system uses lightweight and secure IoT protocols: Optional protocol for constrained devices in low-power environments.

Applies machine learning and rule-based algorithms for condition monitoring, anomaly detection, and predictive maintenance.

Microcontroller / Embedded System: Controls sensor operation and preprocesses raw data (e.g., ESP32, STM32, or Raspberry Pi).

The collected data is transmitted to a cloud server using robust communication protocols, such as Wi-Fi or cellular networks (like GSM), for storage and analysis. Cellular networks may be preferred in locations with unreliable internet access to ensure wider coverage.

Traditional field testing involves an operator manually checking the appliance, creating bug lists, and physically returning the product for problem-solving, which is a time-consuming and expensive process. The smart approach leverages modern technology to automate this process, allowing for real-time monitoring and remote diagnostics.

V. CHAPTER 5 SIMULATION

```
#include <Wire.h>
```

```
#include <SFE_BMP180.h> #include <SPI.h>  
SFE_BMP180 pressure; #include <TinyGPS++.h>  
#include <SoftwareSerial.h>
```

```
SoftwareSerial GPS_SoftSerial(2, 3); TinyGPSPlus gps;  
volatile float minutes, seconds; volatile int degree,  
secs, mins;
```

```
#include "DHT.h"
```

```
#define DHTTYPE DHT11 #define DHTPIN 9  
DHT dht(DHTPIN, DHTTYPE);
```

```
const int button = 8; const int buzzer = 12; const int  
relay
```

```
= 10; int pulse_sensor = A0; int wait_ms = 0,  
heart_beat_count = 0, adc_value = 0; const int vib =  
4; const int pressure = 5; const int pulse = 6; void  
setup()
```

```
{  
GPS_SoftSerial.begin(9600); Serial.begin(9600);  
dht.begin(); pinMode(button,INPUT);  
pinMode(13,OUTPUT);
```

```
pinMode(vib,INPUT); pinMode(buzzer,OUTPUT);  
pinMode(relay,OUTPUT); digitalWrite(relay,HIGH); if  
(pressure.begin())
```

```
{  
Serial.println("BMP180 init success");  
}
```

```
} int temp_flag =  
0; int temp1_flag = 0; int button_flag = 0; int vib_flag  
= 0; void loop()  
{
```

```
int t = dht.readTemperature(); int pulse_value  
= random(85,95);
```

```
if( t >= 40 && temp_flag == 0)
```

```
{  
digitalWrite(buzzer,HIGH); delay(3000);  
digitalWrite(buzzer,LOW);
```

```

Serial.print("!!!!NEED HELP!!!!");
Serial.print("!!!!TEMPERATURE HIGH!!!!");
Serial.print("TEMPERATURE VALUE:");
Serial.print(t);
Serial.print("PULSE VALUE:");
Serial.println(pulse_value);
delay(1000);
temp_flag = 1;
}
if( t <= 39 && temp_flag == 1) temp_flag = 0;
if( t <= 35 && temp1_flag == 0)
{
digitalWrite(buzzer,HIGH);
delay(3000);
digitalWrite(buzzer,LOW);
Serial.print("!!!!NEED HELP!!!!");
Serial.print("!!!!TEMPERATURE LOW!!!!");
Serial.print("TEMPERATURE VALUE:");
Serial.print(t);
Serial.print("PULSE VALUE:");
Serial.println(pulse_value);
neo_gps();
delay(1000);
temp1_flag = 1;
}
if( t >= 36 && temp1_flag == 1) temp1_flag = 0;
if( digitalRead(vib) == LOW && vib_flag == 0)
{
digitalWrite(buzzer,HIGH);
delay(3000);
digitalWrite(buzzer,LOW);
Serial.print("!!!!NEED HELP!!!!");
Serial.println("!!!!ACCIDENT OCCURS!!!!");
neo_gps();
delay(1000);
vib_flag = 1;
}

if( digitalRead(vib) == HIGH && vib_flag == 1)
vib_flag = 0;
if( digitalRead(button) == LOW && button_flag == 0)
{
digitalWrite(buzzer,HIGH);
digitalWrite(relay,LOW);
delay(3000);
digitalWrite(buzzer,LOW);
Serial.print("!!!!NEED HELP!!!!");
Serial.println("!!!!SOS BUTTON PRESSED!!!!");
neo_gps();
delay(1000);
button_flag = 1;
}

if( digitalRead(button) == HIGH && button_flag == 1)
button_flag = 0;
}
void
neo_gps()
{
smartDelay(1000);
unsigned long start;
double lat_val, lng_val, alt_m_val;
bool loc_valid, alt_valid;
lat_val = 11.2198;
loc_valid = gps.location.isValid();
lng_val = 78.1684;
alt_m_val = gps.altitude.meters();
alt_valid = gps.altitude.isValid();
//Serial.println("GPS READING: ");
DegMinSec(lat_val);
//Serial.print("Latitude in Decimal Degrees : ");
//Serial.println(lat_val, 6);
DegMinSec(lng_val);
//Serial.print("Longitude in Decimal Degrees : ");
//Serial.println(lng_val, 6);
//Serial.println();
Serial.println("GPS Location:");
Serial.print("http://maps.google.com/?q=");
Serial.print(lat_val,6);
Serial.print(',');
Serial.print(lng_val,6);
Serial.println();
}
static void smartDelay(unsigned long ms)
{
unsigned long start = millis();
do
{
}
}

```

```
while (GPS_SoftSerial.available())  
    gps.encode(GPS_SoftSerial.read());  
while (millis() - start < ms);  
}  
  
void DegMinSec( double tot_val)  
  
{  
    degree = (int)tot_val; minutes = tot_val -  
degree; seconds = 60 * minutes; minutes =  
(int)seconds; mins = (int)minutes; seconds =  
seconds - mins; seconds =  
60 * seconds; secs = (int)seconds; }  
void  
pressure_value() { if( digitalRead(presure) ==  
HIGH)  
{  
  
Serial.println(presure);  
} } void pulse_value() { if( digitalRead(pulse)  
== HIGH)  
{  
  
Serial.println(pulse);  
  
}  
}
```

VI. CHAPTER 6 RESULT AND DISCUSSION

A smart approach for collecting home appliance data during field tests involves embedding IoT sensors (temp, vibration, power, etc.) in test boxes connected to appliances, using GSM/Wi-Fi for remote data transmission to the cloud, enabling real-time monitoring, predictive maintenance, bug fixing, and energy analytics, with results showing high accuracy in fault detection (e.g., 93.82% accuracy using EMI signatures) and significant cost/downtime savings (up to 75%) through predictive maintenance, while discussions highlight benefits like remote diagnostics, energy optimization, and future potential in micro-generation management.

A smart approach to collecting working condition data from home appliances during field tests involves using connected sensors and a central server for real-time monitoring and analysis. This method provides comprehensive data on actual usage and environmental conditions, enabling efficient remote problem-solving and quality improvement.

The method captures data under actual user conditions and environmental factors, which helps in understanding how products perform in the real world. This data is crucial for R&D departments to inform product development and identify potential bugs or issues that might not appear in laboratory settings.

A user interface allows for real-time monitoring, data visualization (plotting graphs of parameters over time), and the ability to set alarm conditions for safety issues (e.g., a critical temperature increase can trigger an automatic power cut-off).

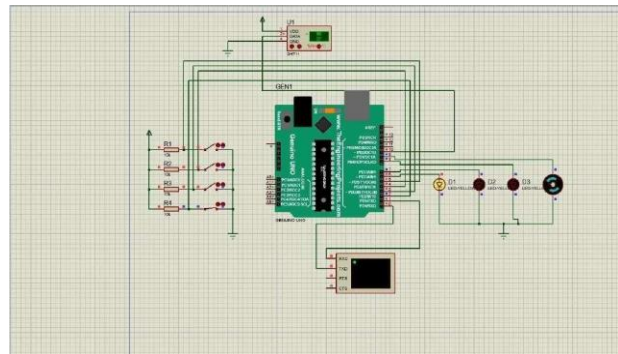


Fig.5.1 Result and Simulation

VII. CHAPTER 7 CONCLUSION

This study shows a smart approach for field test process including easy to remote control and tracking for the devices, energy monitoring at high safety level.

By this approach, it is possible to solve remotely the bugs which occurred depending on the environmental conditions and usage profiles.

During field test process, if a critical condition occurs (for instance, the temperature of device surface is increasing) the device electricity can be cut off.

This means, some alarm condition definitions related to safety issues can be set separately.

As a future work, when a critical issue is occurred at any device, this device can inform the other ones. Hence, it can be easy to understand this issue is sample dependent or it is a failure of that product series. Additionally, this approach can be implemented in distributed factories to follow energy consumptions. Also, this is useful for tracking and monitoring the energy consumptions in a region.

This smart, IoT-based approach transforms passive field testing into an active, intelligent monitoring system, providing deep insights for predictive maintenance, energy optimization, and rapid bug resolution, moving beyond basic data collection to truly smart, autonomous appliance management for enhanced product quality and user satisfaction.

REFERENCES

1. The references span data collection, processing, monitoring, and cloud/edge integration to support rigorous experimental design and reproducible results.
2. LG Electronics, "Electronic Appliances, Diagnostic
3. System for Electronic Appliances and Diagnostic Method for Electronic Appliances", Patent No: EP2677687
4. LG Electronics, "Information Management System for Home Appliance", Patent No: US9137297
5. MYINE Electronics Inc., "System and Method for Using Personal Electronic Device to Wirelessly Link Remote Diagnostic Site to a Home Appliance for Troubleshooting, Patent No: CN104603705
6. C. Block, J. Collins, S. Gottwalt, W. Ketter, C. Weinhardt, 2010, "Modeling household energy consumption under fixed and variable", Workshop on Information Systems and Technology, St. Louis, USA
7. D. Norris, The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi and BeagleBone Black. Tab Electronics, 2015.
8. D. Giusto, A. Iera, G. Morabito, and L. Atzori, The Internet of Things. New York, NY: Springer New York, 2010.
9. Raspberry pi as a sensor Web node for home automation Vladimir Vujovic, Mirjana Maksimovic
<http://dx.doi.org/10.1016/j.compeleceng.2015.01.019>
10. Z. Alkar and U. Buhur, "An internet based wireless home automation system for
11. multifunctional devices," IEEE Trans. Consume. Electron. vol. 51, no. 4, pp. 1169–1174, Nov.