A Power Efficient Scheme for Speech Controlled IoT Applications

Thesis · July 2016

DOI: 10.13140/RG.2.1.4659.3523

1 author:

Nisarg Milan Vasavada

Gujarat Technological University

14 PUBLICATIONS **6** CITATIONS

SEE PROFILE

A Power Efficient Scheme for Speech Controlled IoT Applications

By

VASAVADA NISARG MILAN
141060752022

Guided by

ADITYA KUMAR SINHA
Principal Technical Officer at CDAC ACTS, Pune.

A Thesis Submitted to,

Gujarat Technological University

In partial fulfillment for the degree of Masters of Engineering in

Electronics and Communication (VLSI & Embedded Systems)

May 2016

Gujarat Technological University PG School,

Ahmedabad

CERTIFICATE

This is to certify that the research work embodied in this thesis entitled as **"A Power Efficient Scheme for Speech Controlled IoT Applications"** was carried out by **Vasavada Nisarg Milan** bearing enrollment number **141060752022** studying at **Gujarat Technological University PG School (106), Ahmedabad** for fulfillment of Master of Engineering in **Electronics and communication (VLSI & Embedded Systems)** Degree to be awarded by Gujarat Technological University. The work has been carried out under my guidance and supervision and is up to my satisfaction.

Date: _____ **Place: _____**

Signature of Guide
Aditya Kumar Sinha

Signature of Co-Guide
Mr. Swapnil Belhe

Signature of Principal

Seal of Institute

COMPLIANCE CERTIFICATE

This is to certify that the research work embodied in this thesis entitled as **"A Power Efficient Scheme for Speech Controlled IoT Applications"** was carried out by **Vasavada Nisarg Milan** bearing enrollment number **141060752022** studying at **Gujarat Technological University PG School (106), Ahmedabad** for fulfillment of Master of Engineering in **Electronics and communication (VLSI & Embedded Systems)** Degree to be awarded by Gujarat Technological University. He has complied with the comments given by the Dissertation Phase – I and Mid Semester Thesis Reviewer to my satisfaction.

Date: _____ **Place: _____**

Signature of Student
Vasavada Nisarg Milan

Signature of Guide
Aditya Kumar Sinha

PAPER PUBLICATION CERTIFICATE

This is to certify that the research work embodied in this thesis entitled as **A Power Efficient Scheme for Speech Controlled IoT Applications** was carried out by **Vasavada Nisarg Milan** bearing enrollment number **141060752022** studying at **Gujarat Technological University PG School (106), Ahmedabad** for fulfillment of Master of Engineering in **Electronics and communication (VLSI & Embedded Systems)** Degree to be awarded by Gujarat Technological University, has published article entitled **A Power Efficient Scheme for Speech Controlled IoT Applications** for publication by the **International Journal of Engineering Research and Technology** in Volume 5, Issue 1 during January 2016.

Date: _____ **Place: _____**

Signature of Student	**Signature of Guide**
Vasavada Nisarg Milan	Mr. Swapnil Belhe

THESIS APPROVAL CERTIFICATE

This is to certify that the research work embodied in this thesis entitled as **A Power Efficient Scheme for Speech Controlled IoT Applications** carried out by **Vasavada Nisarg Milan** bearing enrollment number **141060752022** studying at **Gujarat Technological University PG School (106), Ahmedabad** is approved for the degree of Master of Engineering in **Electronics & Communication Engineering (VLSI & Embedded System)** by Gujarat Technological University.

Date: _____ **Place: _____**

Name and Signature of Examiners:

_____ _____

( ) ( )

DECLARATION OF ORIGINALITY

We hereby certify that we are the sole authors of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other University or Institution.

We certify that, to the best of our knowledge, the current thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in our thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, we certify that we have obtained a written permission from the copyright owner(s) to include such material(s) in the current thesis and have included copies of such copyright clearances to our appendix.

We declare that this is a true copy of thesis, including any final revisions, as approved by thesis review committee.

We have checked write up of the present thesis using anti-plagiarism database and it is in allowable limit. Even though later on in case of any complaint pertaining of plagiarism, we are sole responsible for the same and we understand that as per UGC norms, University can even revoke Master of Engineering degree conferred to the student submitting this thesis.

Signature of Student	**Signature of Guide**
Vasavada Nisarg Milan	Aditya Kumar Sinha
(141060752022)	(106)

Dedicated to…

The unsung hero of my life,

My ~~Father~~ Pa.

A C K N O W L E D G E M E N T

A Masters Dissertation is never just a University formality, it is a sole opportunity for a research student to express his knowledge and enthusiasm as a well-documented thesis which expresses his expertise and growth as a technical artifact. Being an engineer means standing on the shoulders of the giants and it is not just an honor but also a matter of fascination when the giants are understanding and helpful in both logical and moral aspects. I would like to express my humble gratitude towards my guide **Mr. Aditya Kumar Sinha** and co-guide **Mr. Swapnil Belhe** for their essential guidance and for making the research experience delightful. A vote of thanks to the VLSI & ESD department of GTU PG School and CDAC – ACTS for their systematic management and continuous support.

Small things make perfection but perfection is not a small thing. I would also like to thank my parents for their endless encouragement and appreciation and my colleagues for creating a workaholic, competitive yet supportive environment.

Nisarg M. Vasavada

(141060752022)

A Power Efficient Scheme for Speech Controlled IoT Applications

Submitted by

Vasavada Nisarg Milan

(141060752022)

Guided by

Aditya Kumar Sinha

Principal Technical Officer,

CDAC ACTS, Pune.

Co-Guided by

Mr. Swapnil Belhe

Technical Officer,

CDAC, Pune.

A B S T R A C T

Sound is one of the most natural means of human expression. And humans being the only species having a variety of sound emissions generated a phenomena called speech. Speech recognition has always been an interesting subject of research since decades. With the change in time the tools, algorithms and the accuracy has drastically changed. Currently Speech processing is implemented based on application specific accuracy requirements and regardless of the implementation scheme the ultimate goal remains the same, to enhance speech processing engines up to an extent where natural language can be interpreted accurately in case of word or phoneme recognition and context prediction. Along with all of these needs when we apply speech processing to embedded systems we also need to be aware of the current research directions in it. On one hand the application development of embedded systems are now being massively focused on Internet of Things, the core researchers' concerns are still security and power efficiency.

Here a state of the art scheme is proposed where speech processing is applied to the constrained IoT applications and the system has been made power efficient. For achieving so, this project covers two modules (nodes) of an IoT architecture where one is a major node while other is a constrained node and applies i-vector algorithm to increase PDF of the speech by means of speaker recognition through wake up call.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

- Hz - Hertz
- HMM - The Hidden Markov Model
- ASR - Automatic Speech Recognition
- LPC - Linear Predictive Coding
- OS - Operating System
- I/O - Input and Output
- IDE - Integrated Development Environment
- DSP - Digital Signal Processing
- SR - Speech Recognition
- XML - Extensible Markup Language
- API - Application Program Interface
- JSON - Java Script Object Notification
- GPIO - General Purpose Input Output
- IoT - Internet of Things
- CEO - Chief Executive Officer
- CMU - Carnegie Mellon University
- IETF - Internet Engineering Task Force
- HMM - Hidden Markov Model
- PC - Personal Computer
- IP - Internet Protocol
- TCP - Transmission Control Protocol
- UDP - User Datagram Protocol
- DLL - Data Link Layer

- PHY - Physical Layer
- LoWPAN - Low power Personal Area Network
- EUI - Extended Unique Identifier
- ISA - Industry Standard Architecture
- 6LoWPAN - Low power IpV6 for Personal Area Networks
- WEI - Wireless Embedded Internet
- BSP - Board Support Package
- WiFi - Wireless Fidelity
- MGTI - Management Interface
- JVM - Java Virtual Machine
- M2M - Machine to Machine
- ISV - Independent Software vendor
- API - Application Program Interface
- SRaaS - Speech Recognition as a Service
- HAL - Hardware Abstraction Layer
- IPv6 - Internet Protocol Version 6
- RPi - Raspberry Pi
- MVA – SI - Modified Vector Algorithm for Speaker Identification
- LM - Language Model
- AM - Acoustic Model
- KB - Kilobyte
- 3D - 3 dimensional
- GPOS - General Purpose Operating System

I N D E X

C H A P T E R 1: I N T R O D U C T I O N

"A decent system fulfills its purpose with accuracy while maintaining its simplicity and resource efficiency"

~ Tim Cook, CEO, Apple Inc.

1.1 Summary of the Project

This project fundamentally distinguishes a certain class of IoT applications from others where user input plays an essential role unlike the autonomous ones where only the key triggering inputs are necessary and the system works on its own account. An easy, less dependent, portable Speech Recognition Engine is customized precisely for IoT applications. Also, considering the durability of small objects and socio-technical issues such as electronic waste the whole system is designed with a power efficient approach which includes 6LowPAN adaption for IP allocation and wireless communication.

1.2 Motivation behind the Project

Even though IoT is emerging rapidly around hundreds of applications and various use cases, the implementation of ASR for controlling or accessing nodes of IoT is still something less experimented and definitely never documented. Basically the project is performed to overcome Power efficiency challenge of IoT and to create an ASR environment by customizing open source trained ASR by CMU. This gives a totally new level of accessibility and convenience on the user end which is exactly why IoT consumer applications are designed.

1.3 Expert Sessions and Webinars

1. ASR using "C", a CMU webinar series.
2. 6LowPAN: Embedded Wireless Internet webinar by Mr. Zack Shelby.
3. Advanced Device Drivers session by Mr. Krishnamurthy Babu.

1.4 Thesis Flow

As the title itself suggests, this thesis represents research and implementation performed as a part of Post-graduation dissertation and **1st Chapter** explains summary and motivation of the project along with a brief outline of thesis flow.

2nd Chapter is about the literature review carried out for the research. Out of many academic and industrial references, the most relevant and influential literatures are discussed along with their overview and importance. Their flaws from author's view are also discussed which lead to problem formation.

3rd Chapter entitled as World of Words is a combined discussion of ASR theory which includes its history, working, algorithm and specifications as well as adaption of ASR in current scenarios. Thus it intertwines ASR market as well as ASR process.

4th Chapter encompasses a brief study and introduction to Internet of Things and jumps to Wireless Embedded Internet approach which is the heart of IoT.

5th Chapter links 3rd and 4th chapters with existing systems and enlists observations which along with literature review define the problem statement.

6th Chapter proposes a system level solution for the problem mentioned in Chapter 5 and elaborates the Algorithm developed by the Author as a part of the solution.

7th Chapter is simply the documentation of implementation performed for the project which includes various experiments, configurations, connections, designs and hardware setup.

8th Chapter named as Unity discusses connection of all nodes under a single platform called Node-red.

9th Chapter portrays Research management cycle. Here month wise progress is enlisted and completed tasks are ticked with correct sign.

10th Chapter Concludes the thesis with comparative tabular results and shows a glimpse of its possible impacts on respective ecosystem.

CHAPTER 2: LITERATURE REVIEW

"Great Literature is simply language charged with meaning to the utmost possible degree."

~ Ezra Pound, Poet.

An innovative research is a result of an idea which is generated to solve a particular problem, which can be reasonable only if the core technology and the progress in the same is known to the researcher. Any missing links can either make him work on solutions which are already present in the research community or can make him opt for unrealistic approach. In this chapter the crux of the research and review papers along with other key documentations is described in brief and at the end a tabular representation sources, publication years and usefulness is furnished.

Literature 1

Title: Speech Recognition with HMM: A review

Authors: B. Singh, N. Kapur & P. Kaur

Publication: IJARCSE, 2012.

Review

Overview:

The heart of Speech recognition lies in the recognition algorithm. This paper gives a brief introduction of Recognizer block of speech recognition engine and jumps to HMM. The paper plays an important role in this literature survey since it touches the very essential basics of one of the key algorithms of Speech Recognition.

Author divides the whole ASR process of speech recognition into 3 steps.

1. Pre-processing
2. Feature Extraction
3. Recognition

Hidden Markov Model is used under the design of Recognizer block.

Flaws:

By the year the paper was published the Language model was trained enough to include more than just one particular algorithm, thus for optimized performance multiple techniques could have been used.

Literature 2

Title: PocketSphinx: A free continuous RT-ASR for hand-held devices

Authors: David Huggins, Mohit Kumar, Arthur Clan

Publication: IEEE ICCASSP conference, 2006.

Review

Overview:

The authors from Carnegie Mellon University which served as the birthplace for Sphinx have elaborated how speech recognition including HMM is implemented and how the performance has turned out in a sleek yet strategic way which makes the explanation still useful. Even though Sphinx has evolved a lot since then, major updates were filling the loopholes and providing various supports which makes this paper one of the core references and even the firm itself suggests so on their web documentation.

The paper serves as an introductory article for PocketSphinx which is an open source speech recognition engine written in C. Since the publishing of the paper the library has grown and has been trained a lot but the basics remain the same.

Flaws:

The paper ignores the building and porting instructions for PocketSphinx which makes it time consuming and full of unknown dependencies.

Literature 3

Title: Speech and Language Processing

Authors: Daniel Jurafsky and James H. Martin

Publication: Prentice Hall Series in Artificial Intelligence, 2000.

Review

Overview:

Experts refer to it as the bible of ASR learning. This book covers not just one or two algorithms but almost everything one needs to know before beginning ASR implementation. The initial chapters provide a well arranged timeline description of advancements that ASR gained with time. There is always a gap between theory and implementation. This book fills the gap with suitable examples and the practical writing approach.

Although not much of the content has been added in this report this book is going to serve as one of the fundamental references for the implementation as it is one of the most reliable documents available which gives introduction to the programming approach od DSP and various algorithms in languages such as C and C++ which serve as a great learning platform as well as some basic skeleton to apply in the practical coding conventions.

Flaws:

The approach in this book is useful for dedicated DSPs but not much of the information is provided about how to implement the algorithms on a generic microcontroller such as ARM which may create dependency tree issues.

Literature 4

Title: An improved i-vector extraction algorithm for speaker verification

Authors: Wei Li, Tianfan Fu, Jie Zhu

Publication: EURASIP Journal on Audio, Speech and Music Processing by Springer Publication (Open Access Distribution), 2015.

Review

Overview:

In case of user specific embedded devices, change in user may turn into different interpretation caused by accent problems which is something important to consider. This paper describes improvement in i-vector which stands for intelligent vector algorithm. The algorithm is an updated version of previous one specially designed to exploit the ILP availability in recent microcontrollers. Here the same data of frequency is provided to two computational units at the same time. One for the speech processing and the other one for the user recognition. Thus, without any loss of data or any context remained misinterpreted, the user is recognized which makes the job of recognizer a lot easier and efficient.

Flaws:

The author did not describe any specific microcontroller range to exploit this algorithm thus the implementation turns out as trial and error attempts and a lot of other platform oriented issues need to be solved while trying to implement is without any OS.

Literature 5

Title: 6LoWPAN: The Wireless Embedded Internet

Authors: Zack Shelby and Carsen Bormann

Publication: Wiley series in communication networking and Distribution Systems, 2009

Review

Overview

This book is not an author's effort to explain a concept, it is a sole documentation created by the leaders of "Wireless Embedded Internet" community of IETF (Internet Engineering Task Force). It is derived from IETF standards for the specific purpose to scale down IPv6 requirements to the core embedded level. The layer removal and address scaling is perfectly explained with proper open source available libraries and APIs. Various application specific topologies are also added along with core hardware of 6LoWPAN which is 802.15.4 and the integration is throughout seamless.

Whenever IoT is discussed, a major concern is security. This book also covers bootstrapping and security issues of embedded level IoT design.

Flaws:

IoT was an introductory term in year 2009 and has advanced a lot during past 6 years and so has 6LoWPAN which has turned from a proposed standard to an existing standard. A newer version with important updates is expected soon.

Literature Table

Table 2.1: Literature Review

No.	Reference	Year	Significance
1	Speech Recognition with HMM: A Review	2012	Introduction to HMM
2	PocketSphinx: A free real time continuous SR for hand-held devices	1990	Understanding of ASR Implementation (An Example)
3	Speech and Language Processing	2009	A complete guide for modern ASR
4	An Improved i-vector Extraction algorithm for speaker detection	2015	Base Research
5	6LoWPAN: Wireless Embedded Internet	2000	A complete guide for Embedded level IoT implementation

CHAPTER 3: WORLD OF WORDS

"When I get ready to talk to people, I spend two thirds of the time thinking what they want to hear and one third about thinking what I want to say."

~Abraham Lincoln, President of USA.

Sound is a natural expression that is sensed, interpreted and delivered in a spectrum of frequencies. Fascinatingly, humans are the species that have learned to modulate it in many different ways and have taken a step further where their way of delivering it surpassed almost all other members of ecosystem which we call, Speech.

Figure 3.1: People Talking to each other [2]

3.1 Automatic Speech Recognition

Using the gift of speech as an input and controlling parameter in the embedded platform has been an interest of research and since early 80s [3]. Many methods, models and algorithms have been developed since then among which one of the pioneer algorithms which is still used in most of the speech recognition engines is known as HMM. [3]

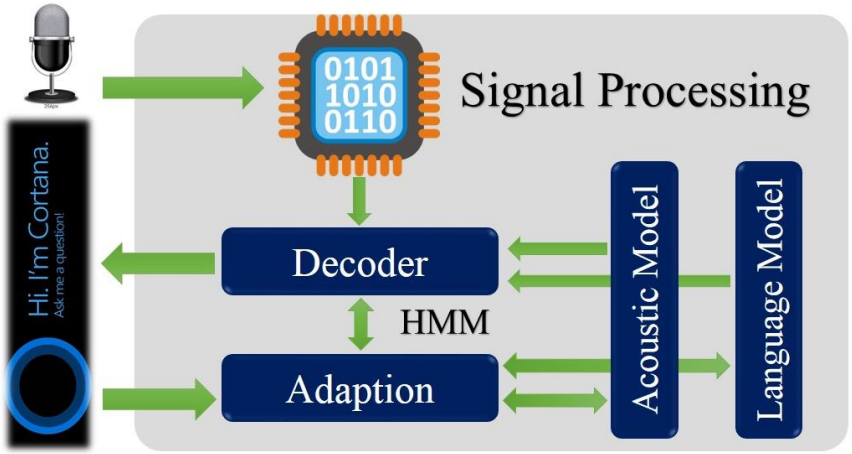


Figure 3.2: ASR Block diagram [1]

3.2 HMM (Hidden Markov Model)

The Hidden Markov Model (HMM) is a variant of a finite state machine having a set of hidden states, an output alphabet (observations), transition probabilities, output (emission) probabilities, and initial state probabilities. The current state is not observable. Instead, each state produces an output with a certain probability.

3.3 Use of HMM in ASR

Hmm can be used to model a unit of speech whether it is a phoneme, or a word, or a sentence. LPC analysis followed by the vector quantization of the unit of speech, gives a sequence of symbols (Vq indices). Hmm is one of the ways to capture the structure in this sequence of symbols. In order to use HMMs in speech recognition, one should have some means to achieve the following.

1. Evaluation of a given sequence having an HMM given.
2. Training to adjust parameters to maximize probability of sequence occurrence.
3. Decoding to find the single best state sequence for given observation.

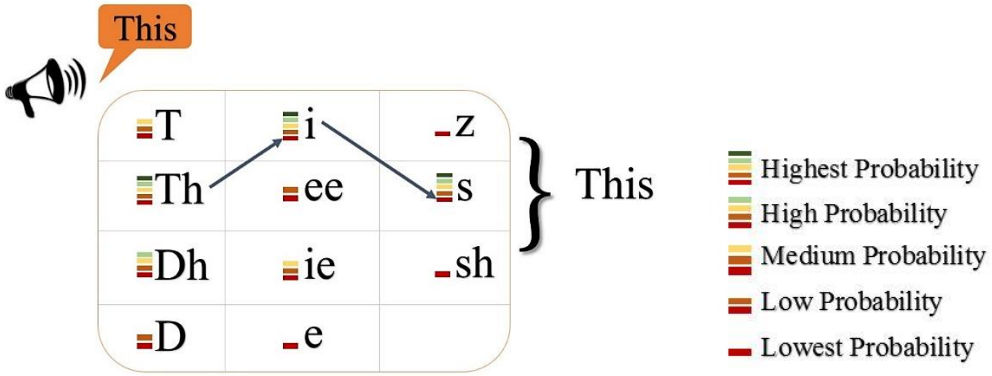


Figure 3.3: ASR example through HMM Probability states

CHAPTER 4: INTERNET OF THINGS

"The Internet is rapidly becoming the town square for the global village of tomorrow"

~Bill Gates, CEO, Microsoft.

As the Internet of servers, routers and PCs has been growing mature, one more revolution was marching on its way – The IoT. The idea behind the IoT is to make small and sensor driven embedded devices IP (Internet Protocol) enabled, and to integrate them as an essential part of the Internet.

Examples of embedded devices and systems using IP today:

1. Mobile phones,
2. Personal health devices and
3. Home automation,
4. Industrial automation,
5. Smart metering
6. Environmental monitoring systems. [17]

4.1 IoT Reference Model

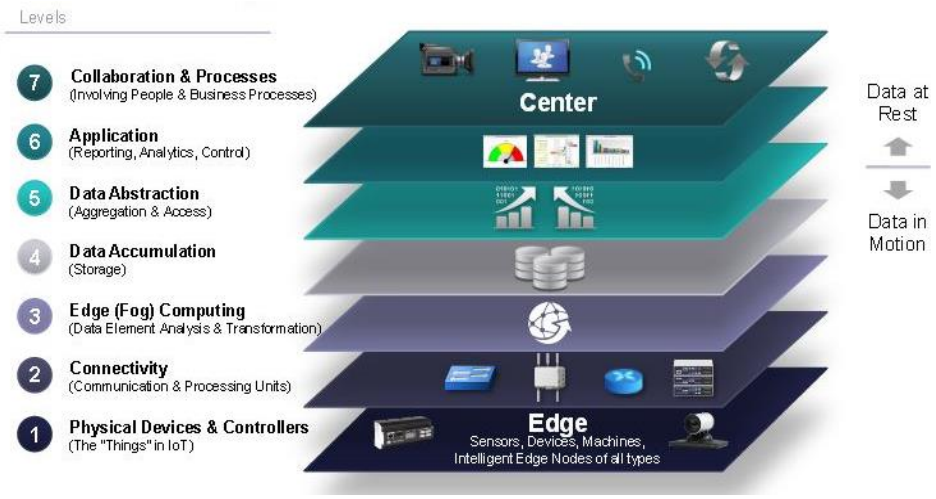


Figure 4.1: Automatic Speech Recognition block diagram [17]

In an IoT system, data is generated from multiple devices spanning various complexity which are processed in different ways, transmitted to different locations, and acted upon by applications. The basic IoT reference model is stacked up of seven levels where each level is defined with its terminology that can be used to standardize to create a globally acceptable frame of reference. This model does not restrict the scope or locality of its components. For example, from a physical perspective, every element could reside in a single rack of equipment or it could be distributed across the world. The IoT Reference

Model also allows the processing occurring at each level to range from trivial to complex, depending on the situation. The model describes how tasks at each level can be handled to maintain simplicity, ensure compatibility, and allow stability. [16]

The above shown figure is a reference model for worldwide developers which is published as an open source reference by Cisco Inc. Here as the embedded system designers we deal with the Physical layer of the model. Physical layer is made of sensors and controllers that might control other devices. These are literally the "Things" in Internet of Things.

4.2 IoT: Wireless Embedded Approach

As the concept of "Things" has a huge ocean of devices in its own category it is one of the most irrational approach treat and consider all of the devices as same. Thus the newest and smallest members of networked world are small sensors and actuators which are embedded devices by nature and do not contain scope or requirement of intelligence similar to fully fledged computing systems. Thus the TCP/IP layered approach is over-sufficient for such devices and it is efficient to remove wrappers of unwanted layers and implement a scaled down approach.

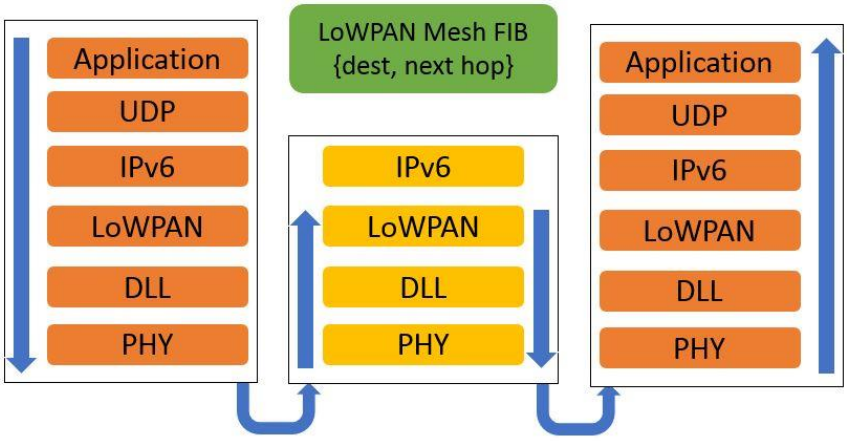


Figure 4.2: 6LoWPAN protocol Stack [16]

The figure shows the layered approach of 6LoWPAN. When routing and forwarding on Data Link Layer, they are performed based on corresponding addresses (64-bit EUI-64 or 16-bit short addresses). The Internet Engineering Task Force is not generally working on "mesh routing" protocols. ISA100 defines one such routing protocol, along with some extensions to the layer-2 that make the fact that routing and forwarding is happening at data link layer is essentially invisible to the LoWPAN adaptation layer, In case the actual link-layer forwarding is not hidden from the LoWPAN adaptation layer.

There is one issue to resolve: the layer-2 headers describe the source and destination addresses for the current layer-2 hop. To forward the packet to its eventual layer-2 destination, the node needs to know its address, the final destination address. Also, to perform a number of services including reassembly, nodes need to know the address of the original layer-2 source, the originator address. Since each forwarding step overwrites the link-layer destination address by the address of the next hop and the link-layer source address by the address of the node doing the forwarding, this information needs to be stored somewhere else. 6LoWPAN defines the mesh header for this. [17]

4.3 IoT: Node Span

The following figure is a skeleton infrastructure of WEI based IoT approach where small embedded devices form a mesh/star topology based ad hoc network with a router which is then connected to the internet which leads it to cloud and billions of other internet enabled devices such as PC or smartphones. In formal IoT terminology, the sensor-actuator rich small embedded systems are nodes, a full-fledged computing system with standard IP stack acts as a Gateway and other members of the internet family act as themselves expanding pervasiveness of connectivity across unprecedented applications and solutions.

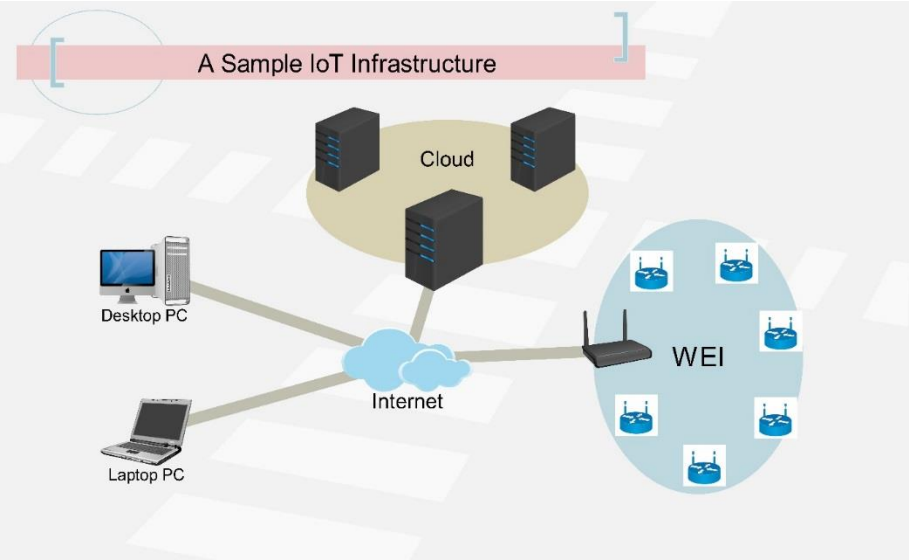


Figure 4.3: SR Constrained IoT [17]

CHAPTER 5: EXISTING SYSTEMS

"Growth never happens as a mere chance, it is a result of multiple forces working together in one direction."

~Warren Buffet, Philanthropist.

This chapter discusses about the systems that are currently in use in terms of IoT and ASR. Ironically, these two have not been merged on a scale where smallest "Things" from the Physical layer of IoT can be governed or monitored by spoken commands as inputs. Also, the industries that involved in the development and investment of IoT are the ones making full-fledged intelligent systems and providing end to end services. Following is an example of market dominant IoT application.

5.1 Recent Implementations

There are three main aspects of any sophisticated IoT implementation.

- Hardware – Software Interface
- Connectivity and Compatibility
- Security

While hardware architecture and OS are virtually free from inter-device compatibility issues, they are the key behind successful running performance of any Embedded system as precision in HAL (Hardware Abstraction Layer) is vital for interrupt handling and I/O GPIO operations. Following figure portrays set of protocols along with their role in the operation.

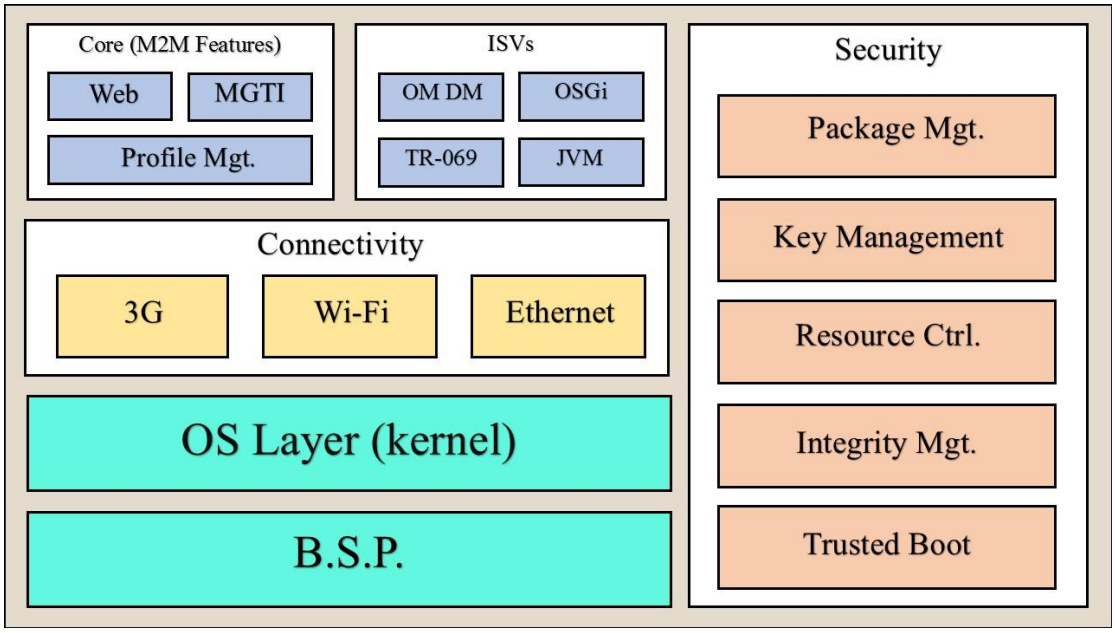


Figure 5.1: Traditional IoT Protocol Stack [16]

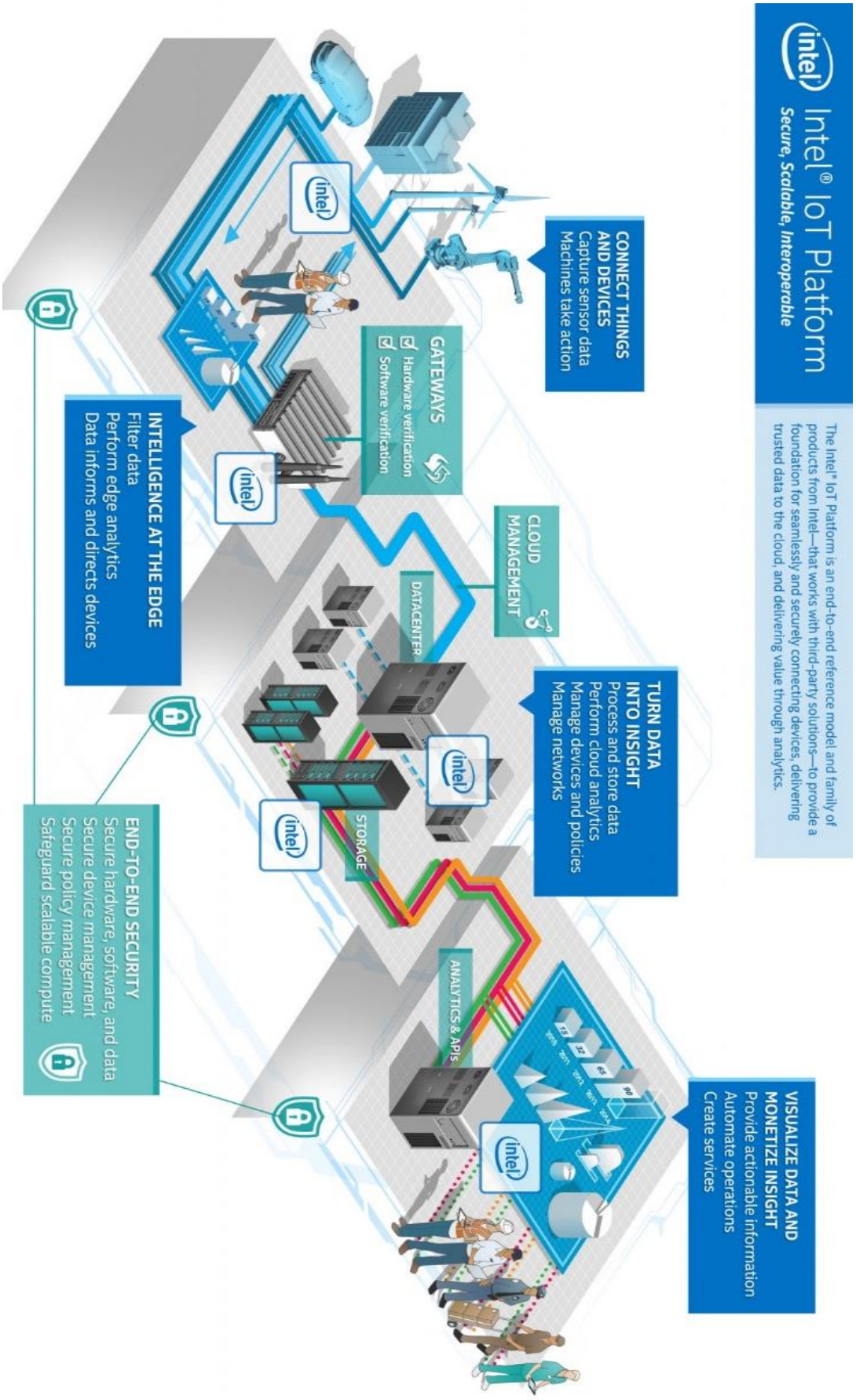


Figure 5.2: Intel Full-Fledged IoT open model [13]

The above figure shows a high-end industrial IoT enabled automation system created by Intel Corporation. Following are the important parameters to note about the system.

1. The data is communicated through cloud platforms.
2. Analysis and actuation is used as a service.
3. Parameter passing to constrained nodes is done through Gateways.
4. There are more than one layers of analysis and security for reliability.
5. The gap is huge between data acquisition and actuation API.
6. Web interface is used to interact with the data.

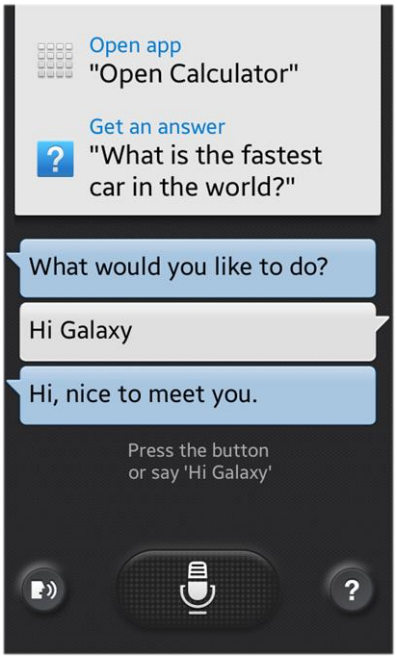


Figure 5.3: ASR Application on a leading smartphone

Switching observations to ASR systems, currently it is largely implemented in smartphones as personal assistant application coupled with web APIs for various applications and in some case to use SRaaS (Speech Recognition as a Service). Typically, dedicated hardware is implemented for Vector speech processing and the adaption is highly OS as well as HAL dependent. In the implementation department, Acoustic and Language models are constructed using java libraries and are trained dynamically. Multiple high-end connectivity protocols and application layer protocols are involved and multi – layered APIs for interaction are dependent on multiple platforms such as top layer application for user interface, network and transport for information exchange and physical resources themselves for running regardless of the configuration and availability of higher protocol stacks.

5.2 Problem Identification

By looking at above examples it is trivial that these systems are convenient, accurate, secure and reliable but the audience for such systems is pretty limited since the embedded domain comprises of many small low end devices which are envisioned as the basic building blocks and source of success for the IoT. The approach for automating, monitoring and designing those devices is certainly different from regular ones and there need to be schemes and frameworks available which can act as bridges between two distinct implementations. The frameworks should be keeping following aspects into consideration.

1. The input-output interface for actuation and monitoring should be coherent and seamless. [17]
2. The bandwidth and battery life should be taken into account.[16]
3. The approach should be cost effective[16]
4. The networking should be achieved with as less layers as possible.[16]
5. As for speech recognition, speaker independence should only be applied if the nature of application demands it·[7]
6. The training should be focused on required words and phrases only. [6]
7. The whole unit should be made out of single platform to avoid as many dependencies as possible.
8. The unit should be made easy to port which can be handled by means of pre-processing and shell scripting.
9. The System and internal IPs should opt for open source implementation as it would encourage hobbyists and developers to move the applications forward.[17]
10. Even though smartphones can be used to overcome many of the above mentioned problems… “Phones are made for CALLING, thus calls will override all the priorities and we may lose the purpose while maintaining balance.

CHAPTER 6: THE SOLUTION

"We cannot solve our problems with the same mindset we had when we created them."

~Albert Einstein, Physicist.

6.1 System Architecture

The problems mentioned in the previous chapter require a design which is made by taking the both resource efficiency and accuracy aspects into consideration. The proposed system is an IoT implementation at constrained embedded level where specific nodes are provided local IP address while major routing nodes are provided proper internet connectivity where all traditional concepts of cloud, gateways, services, security etc. are applied. For achieving so in most optimized manner the hardware requirements are drastically different which is a key difference in implementation of this project. Following figure describes the overall proposed system.

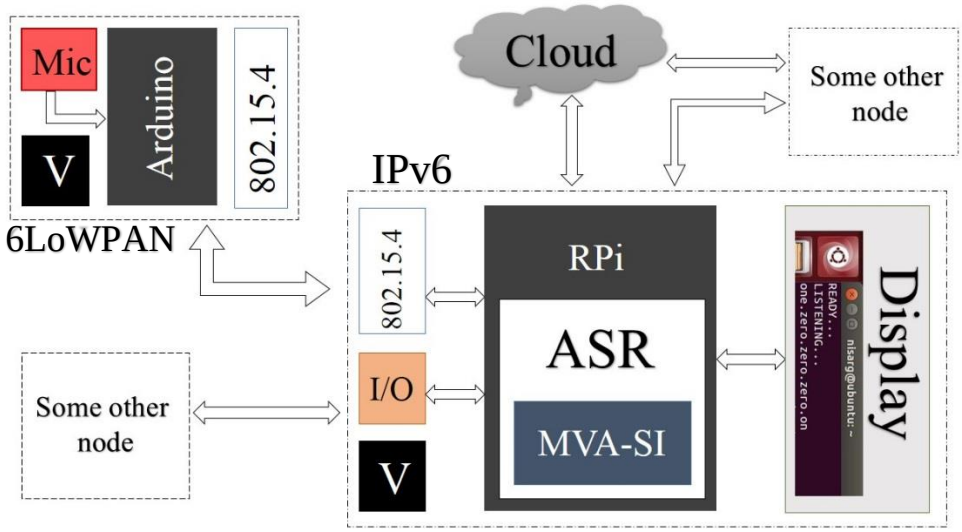


Figure 6.1: System Architecture – Block Diagram

As the above diagram displays, the Mic provides output to the Arduino (Here the reason for choosing Arduino is just ease of prototyping and implementation, for industrial or large scale manufacturing purpose similar microcontroller is suggested since it would be cost effective) which sends it through Bluetooth to the major node in the IoT frame which is governed by Raspberry Pi and is provided full TCP/IP stack for proper internet connectivity across IPv6 [9] [10]. The I/O are connected to the RPi (either in wired manner or in wireless manner depends on the choice of the developer) which are governed by the inputs received from the voice commands sent from smaller nodes. The display is

mainly for prototyping and debugging purpose[10]. The RPi is equipped with open source modified ASR engine which is nothing but a set of files in a proper hierarchy (The Linus Torvalds Philosophy). The speaker dependence (as per the requirement) is handled by the "MVA-SI" algorithm which will be applied in the acoustic model of the ASR engine. The following figure shows the logic of how MVA-SI will work.

6.2 Modified Vector Algorithm for Speaker Identification

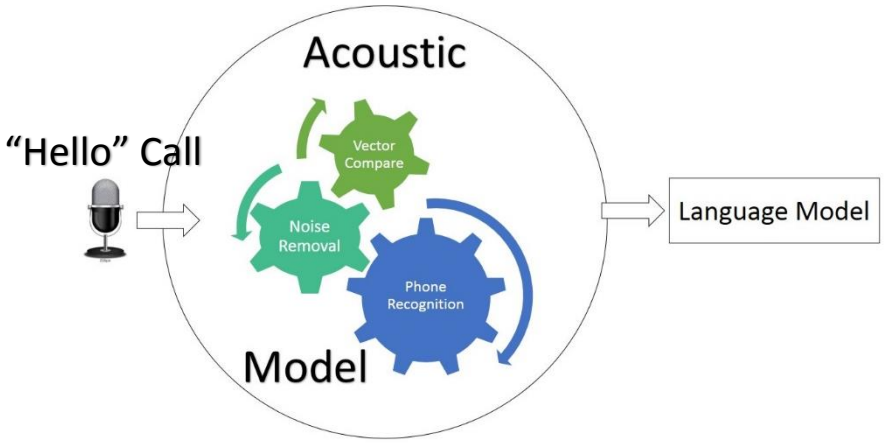


Figure 6.2: MVA-SI Pictorial Representation

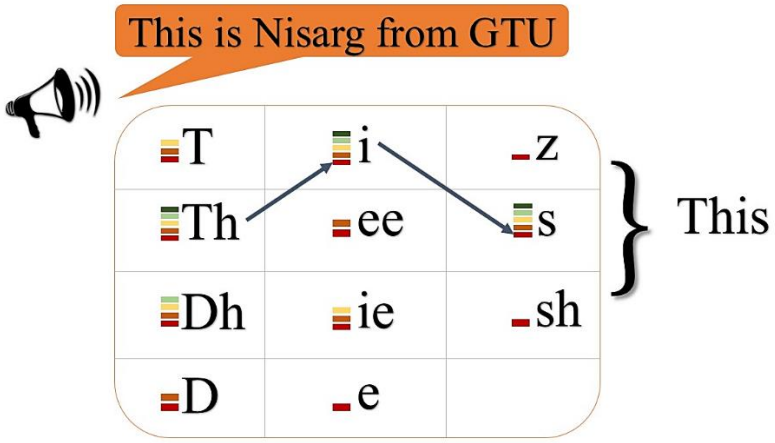


Figure 6.3: ASR using HMM

Figure 6.3 shows how speech is recognized in acoustic models of an ASR engine. First of all the initial frequency is recognized and quantized, the digitized values match a certain set of possible phonemes which have different probability weight. The combination of phonemes with most consecutive probabilities is finalized which is "This" in above demo. Similarly, all the words are recognized by acoustic model but after initial word, language model also plays an important role in correct recognition. In figure 6.4, n-gram language model significance is demonstrated where phonemes which have their dedicated acoustic model probability weight also have n –gram LM

weight. The decision is now cumulative and in case of equal probability of two phonemes (here in this example, it is assumed that the speaker has pronounced a wrong phoneme so the quantized values increase the probability of a different phoneme whereas acoustic model is aware of the possible context so it suggests some other phoneme such as ‘ss’ and ‘s’ for ‘iss’ or ‘is’. Now the language model knows that the previous word was ‘This’ so it would put more probability weight on ‘is’ than ‘iss’ and thus we get corrected recognition.)

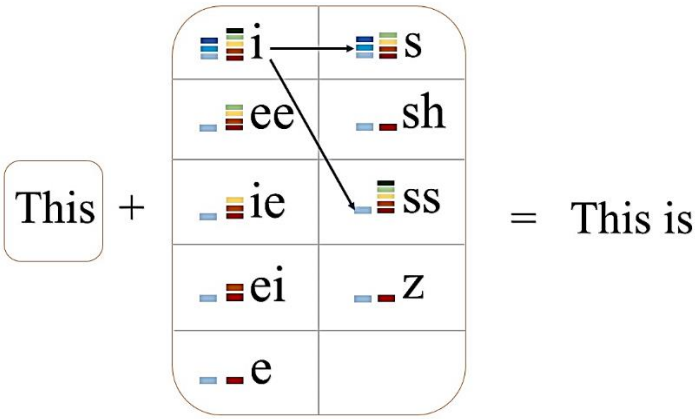


Figure 6.4: ASR using HMM and n-gram weighted model

The MVA-SI actually does nothing but analyzing the phones in the speech context more precisely and giving significant eigenvalues for the vectors generated by the input speech. The inclusion of MVA-SI is the crux of this whole project. Phones are the starting of consonants and endings of vowels which are all distinct from each other.

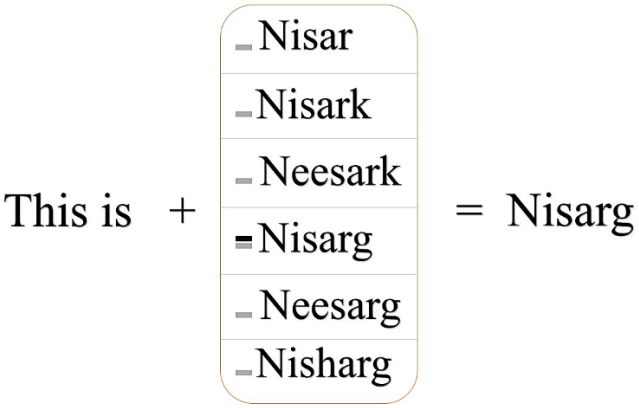


Figure 6.5: ASR using HMM, n-gram and MVA-SI

The algorithm focuses on the phones and the floating point values generated by them and creates an eigenvector. Such eigenvectors are compared to the stored ones and user is recognized. This is all done by one “Hello” wake call which initiates the ASR and also recognizes the user. Since the keyword for this operation is predefined the complexity of the algorithm decreases while the efficiency increases which is the main

aim of not just this but ANY system regardless of the domain and purpose. The requirement of such algorithm arises when words are unconventional and mostly not stored in either acoustic or language models. One of such examples is a name. MVA-SI stores the user specific data and provides binary probability to words. This case is different from AM or LM since here either the word is black or white means either the probability is 0 or one. Such a scenario is expressed and resolved mathematically in the expressions below.

In the equations below, P_1 and P_2 are tentative probabilities of utterance whereas 'A' and 'B' are binary probabilities of MVA-SI (It is worth noting that the number of cases may differ for other utterances). P_1 and P_2 are obtained using trained AM and n-gram LM. Trivially, one will be float and other will be zero so comparison would be easy. Thus P_{final} is derived from MVA-SI.

$$P_1 = [P(1)|P(2)]*[P(12)|P(3)] *[P(12\ldots n\text{-}1)|P(n)]$$

$$P_2 = [P(1)|P(2)]*[P(12)|P(3)] *[P(12\ldots n\text{-}1)|P(n)]$$

$$\text{If } P_1 = P_2$$

$$P_1' = P_1 * A \; ; \; P_2' = P_2 * B$$

$$\text{If } P_1' > P_2'$$

$$P_{final} = P_1'$$

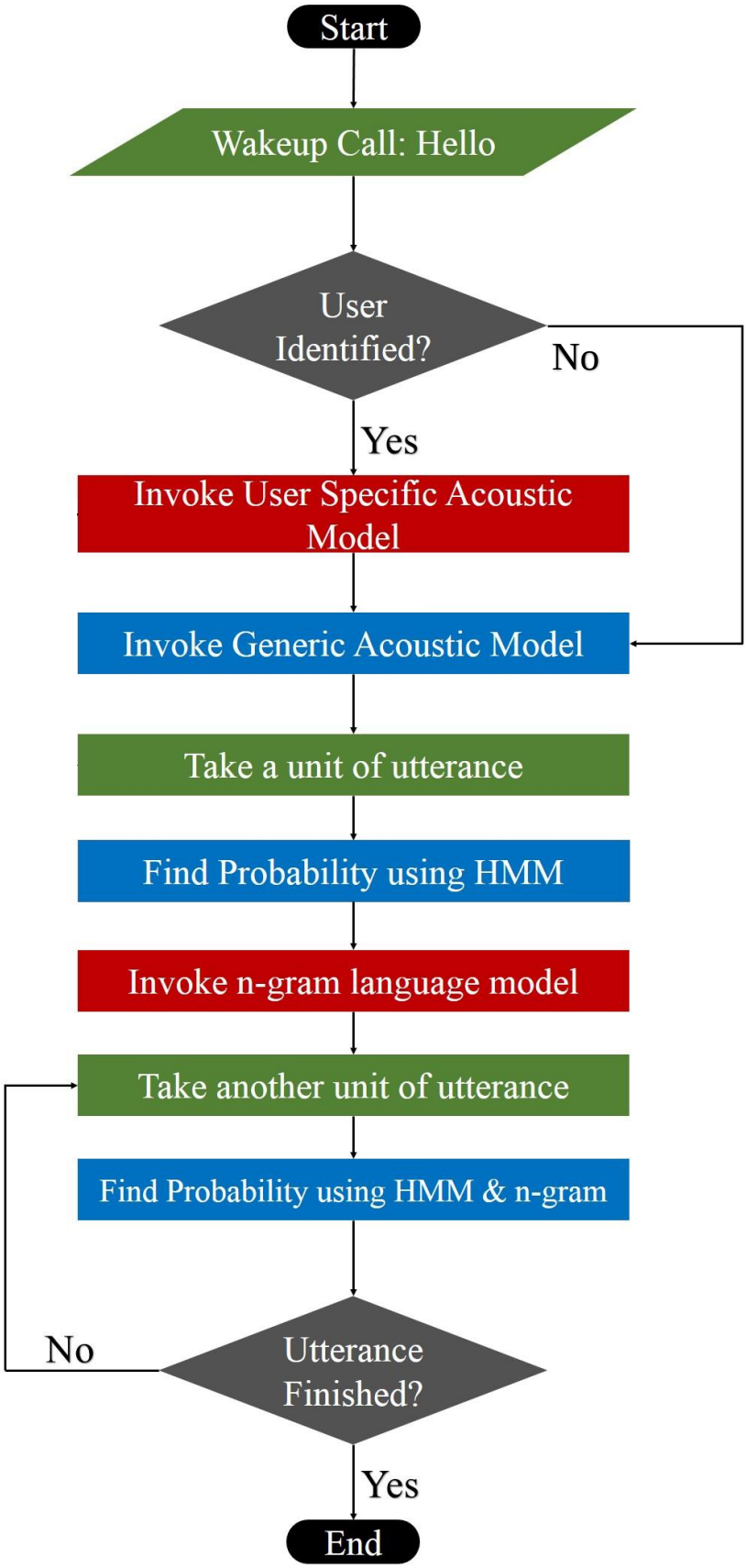


Figure 6.6: Flow chart of MVA-SI

CHAPTER 7: IMPLEMENTATION

"Knowledge by itself is vulnerable, true strength is achieved by molding the knowledge into implementation."

~Sundar Pichai, CEO, Google Inc.

Considering the scope of the project, implementation of MVA-SI on power efficient IoT systems includes a prototype model with 1 node, 1 gateway, ASR modification (building, training and replacing i-vector with MVA-SI) and 802.15.4 XBee radio communication using 6LoWPAN on network layer. In this chapter, designs of all nodes with tool specification and screenshots of working stages with brief elaboration is covered.

7.1 Hardware Design

Hardware is the base of any embedded system regardless of complexity. Bugged software can be debugged but hardware with improper or incomplete design cannot be compensated. Use of open source hardware platforms reduce the efforts and development cycle while increasing simplicity and efficiency. In the prototype of this project, WEI node is created using Arduino Uno whereas gateway is created using Raspberry Pi 2 and both are described in sections below.

7.1.1 Node 1 (WEI Node)

Node 1 or WEI node is a manually operated mobile node (*Note: by the term Mobile, the author does not refer to cellphones, it just signifies its usage mobility*) consisting of Arduino board, electret 3 pin microphone and 802.15.4 XBee radio configured as transmitter. Here the generic router and coordinator configurations of XBee do not work since they are dedicatedly designed for Zig Bee protocol whereas here the network layer is managed by 6LoWPAN.

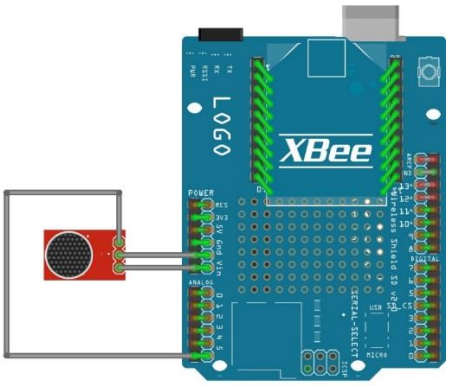


Figure 7.1: Fritzing Board Layout of Node 1

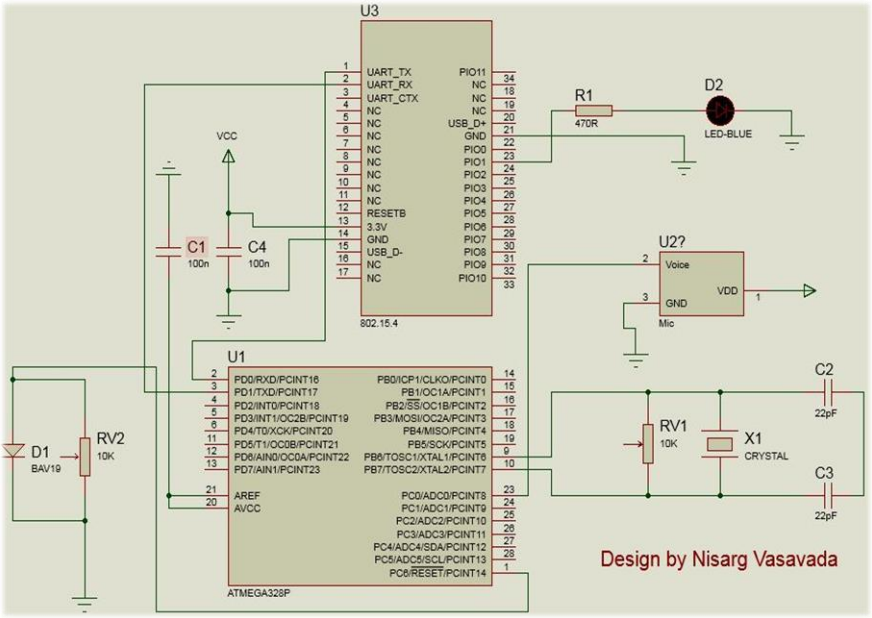


Figure 7.2: ISIS Design of Node 1

7.1.1.1 Arduino Uno R3

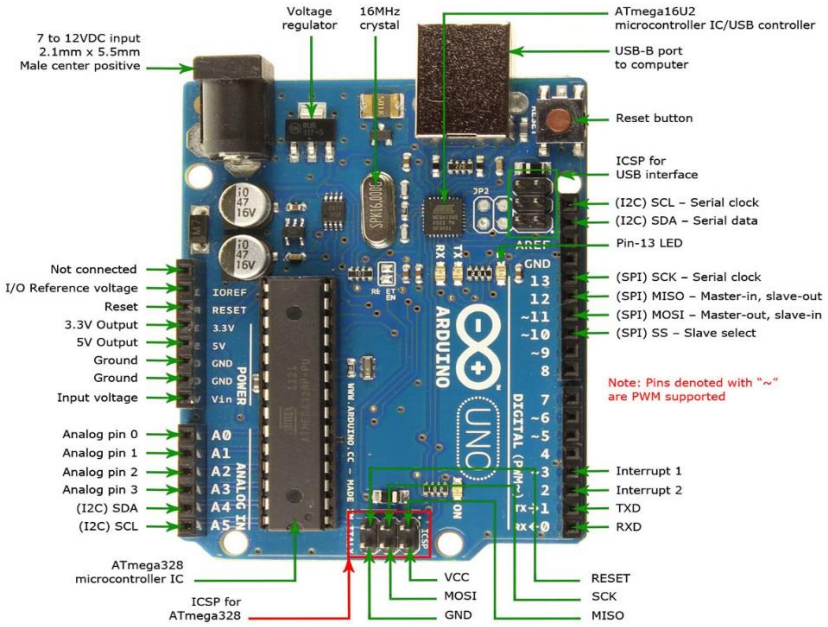


Figure 7.3: Nomenclature of Arduino Uno R3 [3]

This ATMega328P operated, 5v driven open source hardware prototyping board is the game changer for the maker community across the globe. Originated in Italy, this board provides one of the simplest IDEs for coding and plug and run USB interface. The computational capacity, clock rate and 2kb of ram is sufficient to stream chunks of voice commands over wireless media. With constantly updating libraries, 6LoWPAN can also be implemented on Arduino using µIPv6 or pIPv6 lightweight networking

libraries. This protocol implementation using hop-to-hop communication following MAC addresses for node identification thus even the IP allocation and log generation is not required which sets significant amount of RAM bytes free.

7.1.1.2 Electret Microphone

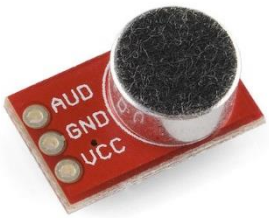


Figure 7.4: 3-pin Mic [3]

This is the simplest and cheapest microphone that can be used for prototyping. Although this low cost microphone costs even less than `100, it is not much useful without an amplifier. Thus it is convenient to use a Mic with the breakout board itself which is shown in the figure. The breakout consist of an op-amp which amplifies the sound to make it more precise. Eagle schematic of the op-amp is shown below.

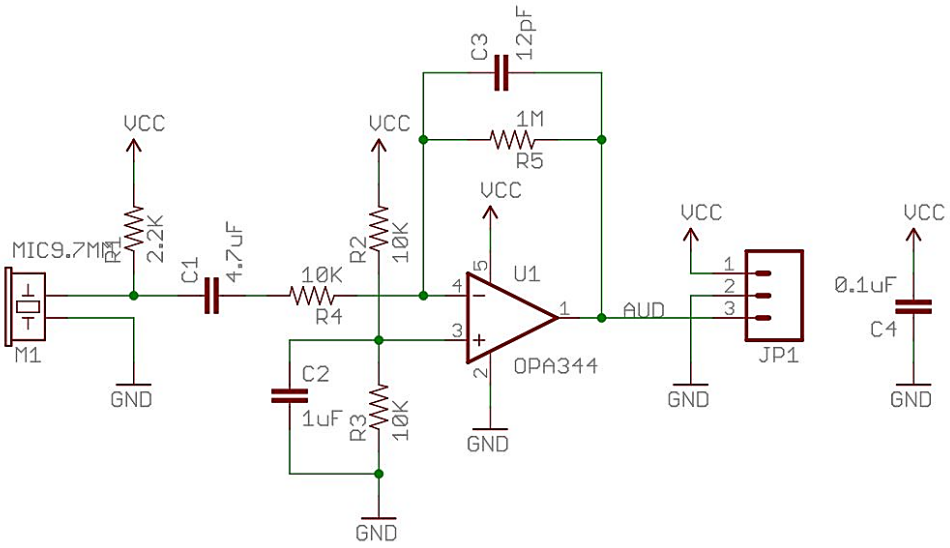


Figure 7.5: Design of Microphone breakout with Amplifier [6]

7.1.1.3 XBee Radio

XBee is a market jargon used to refer to radios falling under compatibility of IEEE 802.15.4 wireless device standard. These low power devices were introduced as PAN

Figure 7.6: Xbee [3]

hardware layer components along with their network layer companion ZigBee protocol. The popular firmware also supported ZigBee compatible Coordinator and router modes. While establishing 6LoWPAN, it was intentionally decided to use a hardware which was already widely in use. This would reduce the initial adaption cost and would also improve the

acceptance ratio since developers would not have to buy new hardware. On a parallel note developers also established libraries for 6LoWPAN compatible to Arduino Uno and various other platforms. Comparing the data headers of conventional ZigBee and 6LoWPAN, it can be seen that the payload of 6LoWPAN is more than two times greater compared to ZigBee. This can be seen by calculations shown below. Here it is worth noting that to increase the power efficiency and keeping WPAN applications in mind, the default transport layer adaption follows UDP instead of TCP which removes scope of acknowledgement signals. In case of critical application the approach can be changed and on the other hand, other IoT targeted transport layer protocols can also be used such as MQTT.

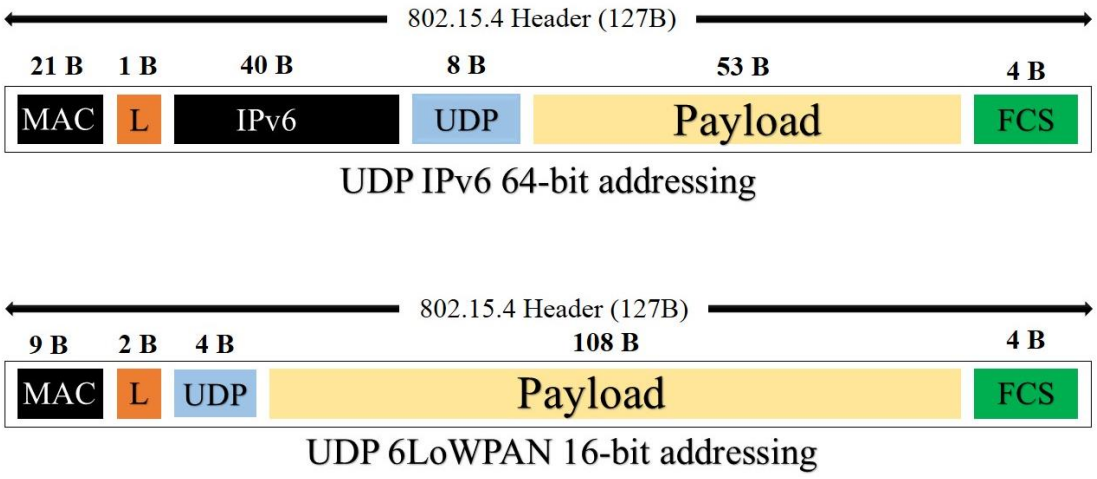


Figure 7.7: 6LoWPAN UDP header [16]

By taking comparative cross ratios of both the protocols we can derive the improved bitrate of 6LoWPAN as follows,

$$6LoWPAN\ BitRate = \frac{6LoWPAN\ Payload * ZigBee\ Bitrate}{ZigBee\ Payload}$$

$$6LoWPAN\ BitRate = \frac{108 * 250}{53} = 519.09\ kbps$$

7.1.2 Node 2 (WEI Gateway)

Similar to previous node, this node also exploits connectivity through 802.15.4 radio using UDP 6LoWPAN but the role of this node is major as it serves as the IoT gateway. it is the link that connects small nodes to the global internet and allows monitoring and data acquisition using browser UI or other such platform. For computing purpose this

node consists of Raspberry Pi 2 which is not only connected to XBee radio for 6LoWPAN but is also provided full TCP/IP stack and Ethernet or Wi-Fi connection. Debian variant of Linux manages necessary software support. 6 times faster to its predecessor, Raspberry Pi 2 is powered by Cortex A7 microprocessor and 1GB of RAM and a full Gigahertz of clock rate if run in turbo mode. These powerful specifications along with vast community support makes it a proper choice for open source hardware prototyping. The core function of RPi in this system setup is to perform speech recognition from its natively installed ASR engine (CMU Sphinx). Apart from that, the scheduling of ASR has been optimized using a wakeup call activation with MVA-SI. In most of the cases Pi acts as receiver and Xbee radio is connected to its GPIO using D1 and D2 pins. The power supply of Pi is managed by local 5v adapter and nature of this node's application demands it to remain immobile.

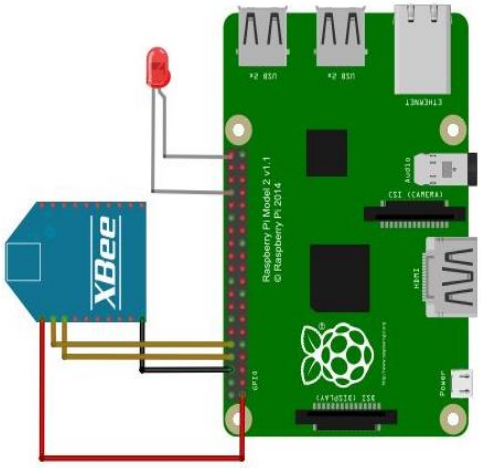


Figure 7.8: Design of Node 2 [3]

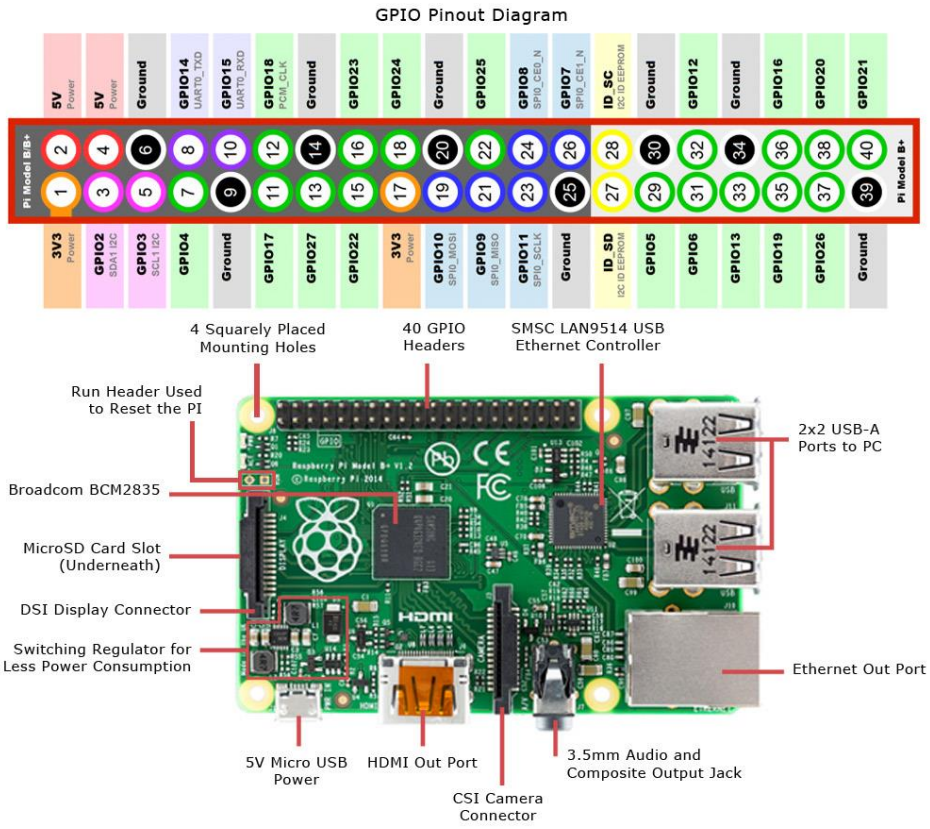


Figure 7.9: Nomenclature and GPIO map of RPi 2 [3]

7.2 Software Setup

Being an Embedded system project, the software also plays an important part in the prototype design. This section describes and elaborates choice and functionality of software, elaboration of sequential software workflow and discussion of initial results.

7.2.1 Arduino IDE

Open source and free Arduino Integrated Development Environment is one of the key reasons which made this hardware board product line so popular. The coding philosophy simply divides whole program into 2 segments (core functions) among which one is used for setup which executes only once whereas the other one is a loop function which runs as an infinite loop until some interrupt causes it to stop. The inclusion of libraries is simple and developer support is sufficiently large thus users mostly don't need to develop new libraries.

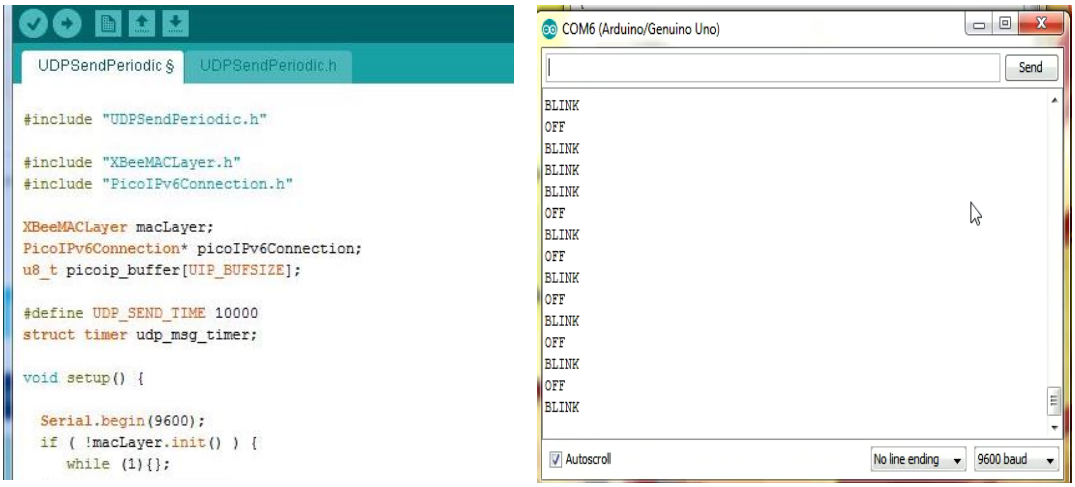


Figure 7.10 6LoWPAN state communication

Figure 7.3 shows snapshot of serial terminal of Arduino on my system. In this setup for testing purpose, two Arduinos are communicating to each other where one is sending the state of its LED on pin 13 to other which are defined as 'BLINK' and 'OFF'.

7.2.2 PocketSphinx ASR

The Automatic Speech Recognition engine is setup in virtual Raspberry Pi's Ubuntu patch and the actual speech recognition is tested and implemented. Following are the snapshots of the same. The procedure has simple implementation and constraints. The figure below indicates successful building of PocketSphinx after handling prerequisites and dependencies such as Bison parser and ALSA player. The sample code was written in C to execute the functionality.

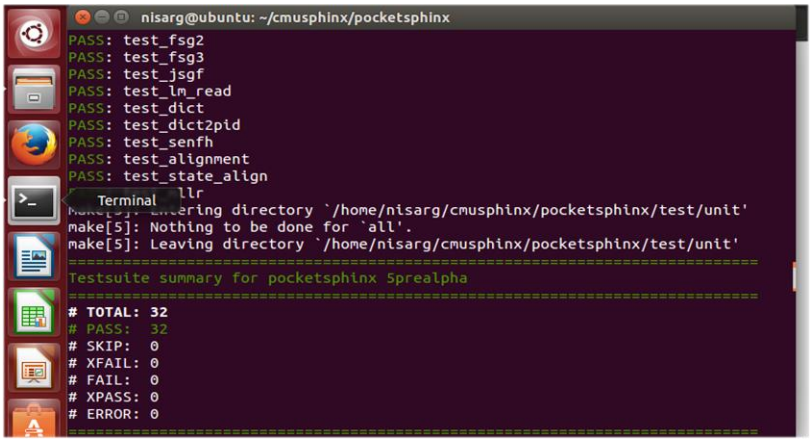


Figure 7.11: Build of PocketSphinx ASR

The input was a standard file which got executed accurately whereas the voice input of author himself as a user was interpreted inaccurately. This is exactly what happens when a constrained system tries to become speaker independent.

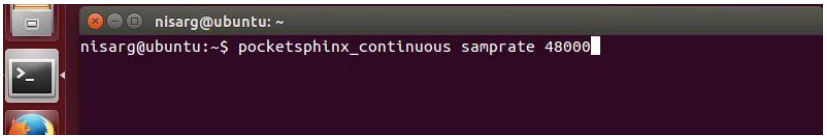


Figure 7.12: Initiation of Speech Recognition

In the figure shown above, two command line parameters are vitally important. Continuous suggests that the input speech will be a structured utterance and not just fragmented words which creates requirement of inclusion of n-gram language model. If left undefined, by default the value of ‘n’ in the n-gram model is 7. This means while interpreting a word in an utterance, it will take 7 previous words into account for constitution of proper sentence. The logic behind decision of sampling rate value falls under Nyquist’s sampling rate theorem. Since the voice from Arduino’s Mic is transmitted over the rate of 16000, 48000 is sufficient to produce proper samples.

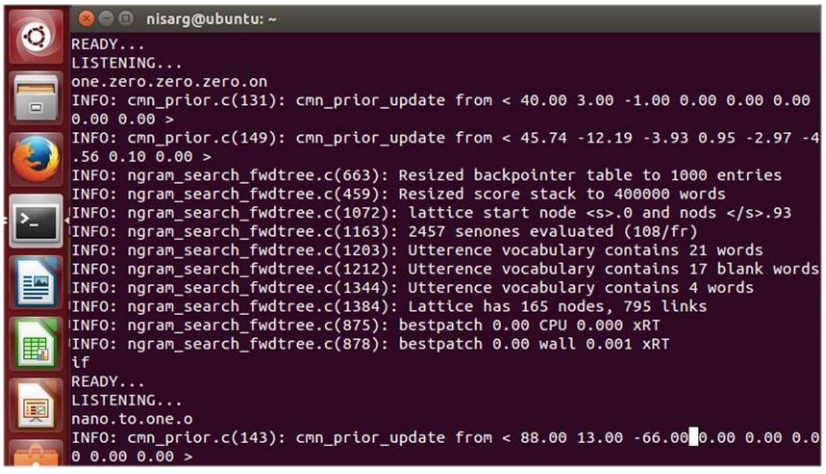


Figure 7.13: Speech Recognition

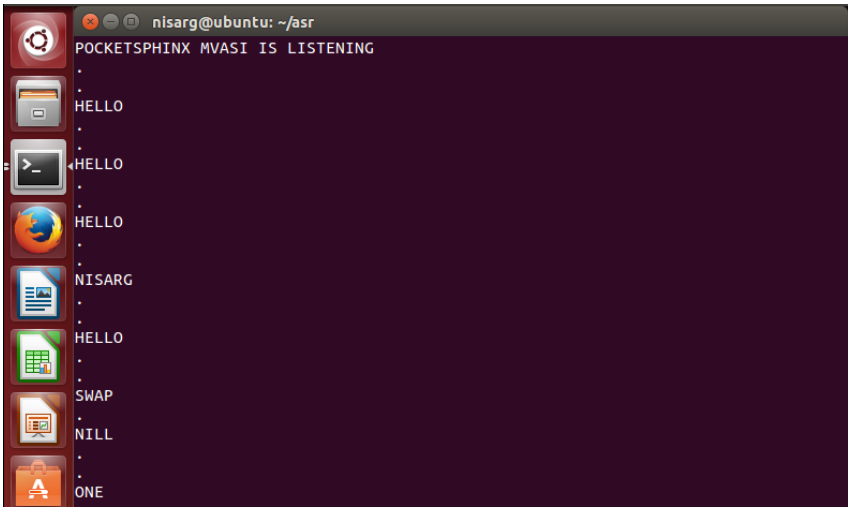


Figure 7.14: ASR with MVA – SI

The input was a standard file which got executed accurately whereas the voice input of author himself as a user was interpreted inaccurately. This is exactly what happens when a constrained system tries to become speaker independent.

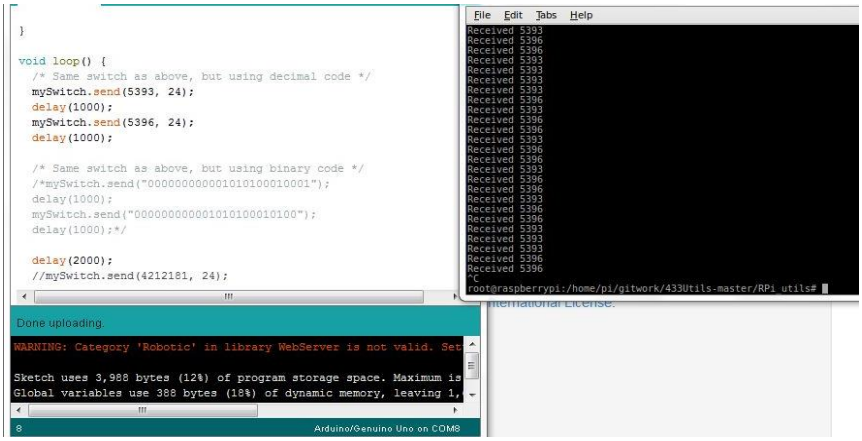


Figure 7.15: 6LoWPAN over Nodes

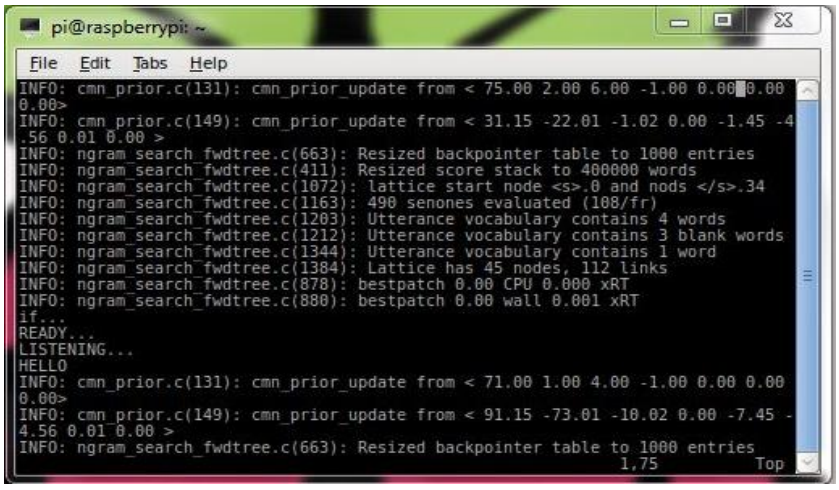


Figure 7.16: Testing on RPi

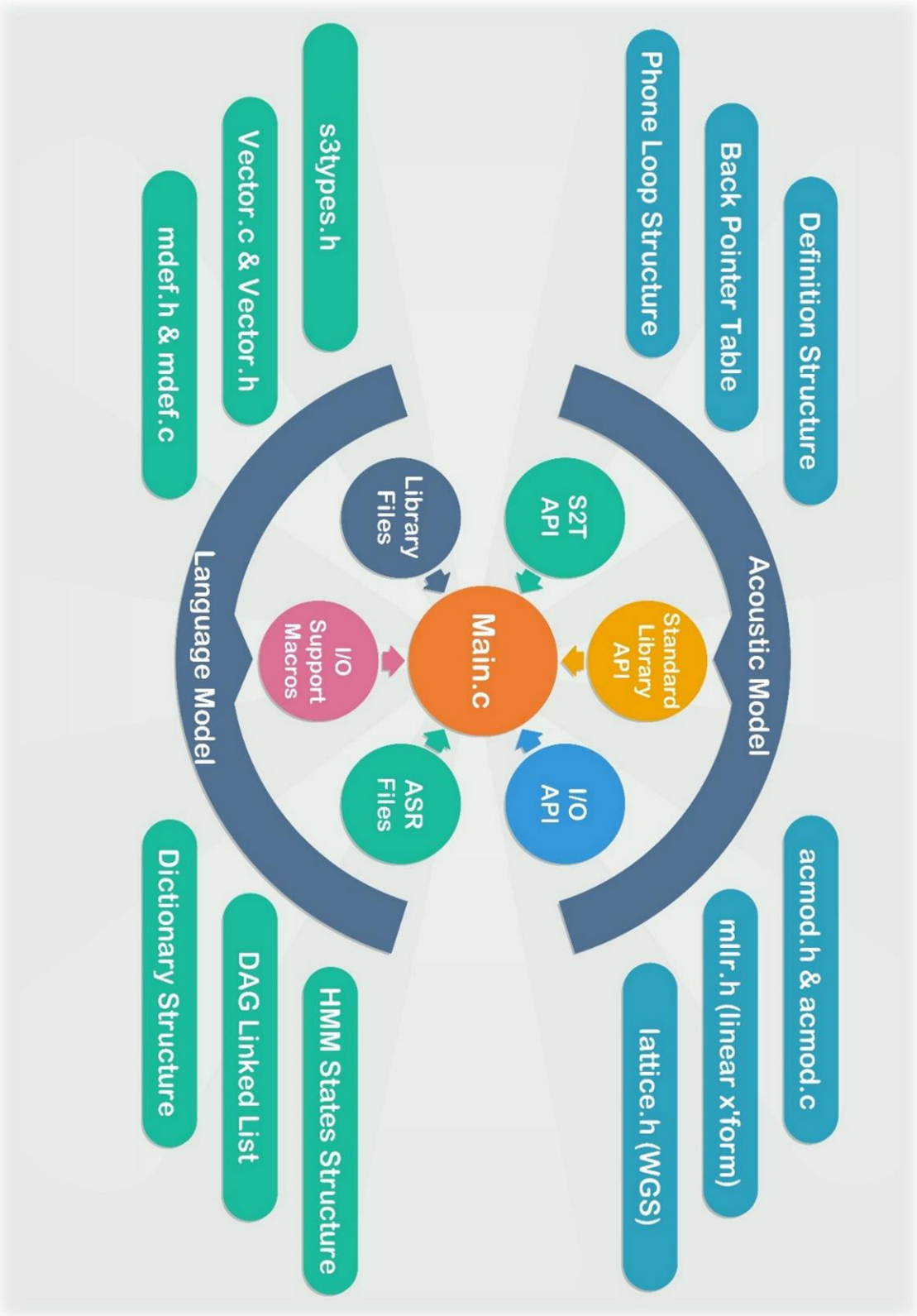


Figure 7.17: File management of PocketSphinx [14]

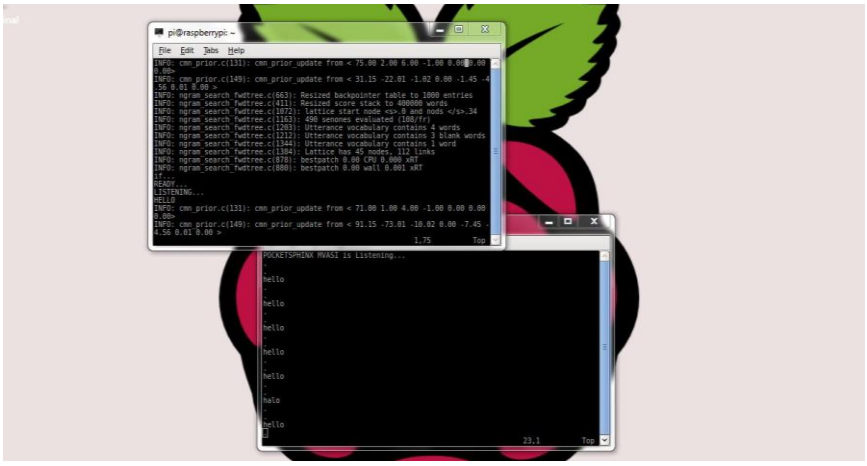


Figure 7.18: MVA – SI on RPi

The similar experiments along with MVA – SI and 6LoWPAN with 802.15.4 were performed on RPi 2 and to extend the virtual applicability, 6LoWPAN network with similar configuration is simulated in Cooja Network Simulator which I shown below.

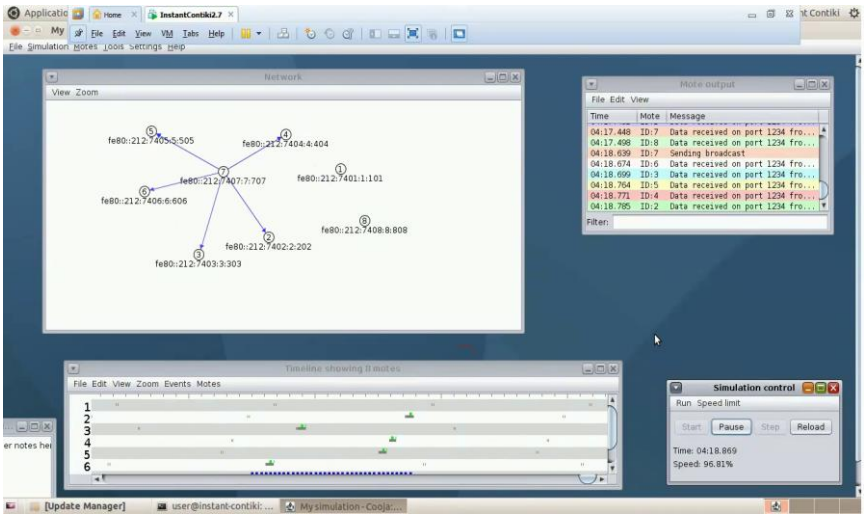


Figure 7.19: 6LoWPAN IoT in Cooja

Following are the observations done accordingly before Dissertation Phase-1 on PC:

1. 40kb of ram was used during the process as stack.
2. According to sampling rate which was set as 48000 (To maintain internal ILP as 1) 21 words were detected.
3. Out of those 17 did not contain any utterances thus were avoided.
4. 4 actual words were detected (Which is actually accurate)
5. The first effort was quite accurate following American accent whereas the second which was in Indian accent got quite a lot of errors. (Which is the problem to be solved).

Following are the observations done accordingly recently on Raspberry Pi 2:

1. 40kb of ram was used during the process as stack. (It turns out that most of this stack remains unused and is refreshed often to ensure efficiency).
2. According to sampling rate which was set as 48000 (To maintain internal ILP as 1) 4 words were detected.
3. Out of those 3 did not contain any utterances thus were avoided.
4. Hello was detected 28 times correctly out of 31 times which makes WER 9.7% which is even less than Google Voice Search's WER recorded in 2014 (Although they have decreased it during 2015 by implementing machine learning via neural networks but that is out of the context).
5. The network simulation worked seamless with 8 IoT 6LoWPAN nodes in Cooja.
6. The Arduino boards communicated successfully via pIPv6 stack and transferred LED blink states by means of serial strings.

CHAPTER 8: SINGULARITY

"Simplicity is the ultimate form of sophistication"

-Leonardo Da Vinci, Engineer.

From the previous chapters it is trivially visible that despite of MVA-SI recognizing its users and enhancing the ASR accuracy, it takes an irritating amount of tools, platforms and methods to look at or monitor the whole system. This defeats its application crux since for a commercial application user interface is as important as performance. Combining all of this with 6LoWPAN, maintenance engineer will spend most of his time switching between Arduino serial port monitor, Jessie's terminal and hardware indicators such as LED. This issue is solved by integrating all of the nodes and gateway using single IoT platform, Node Red.

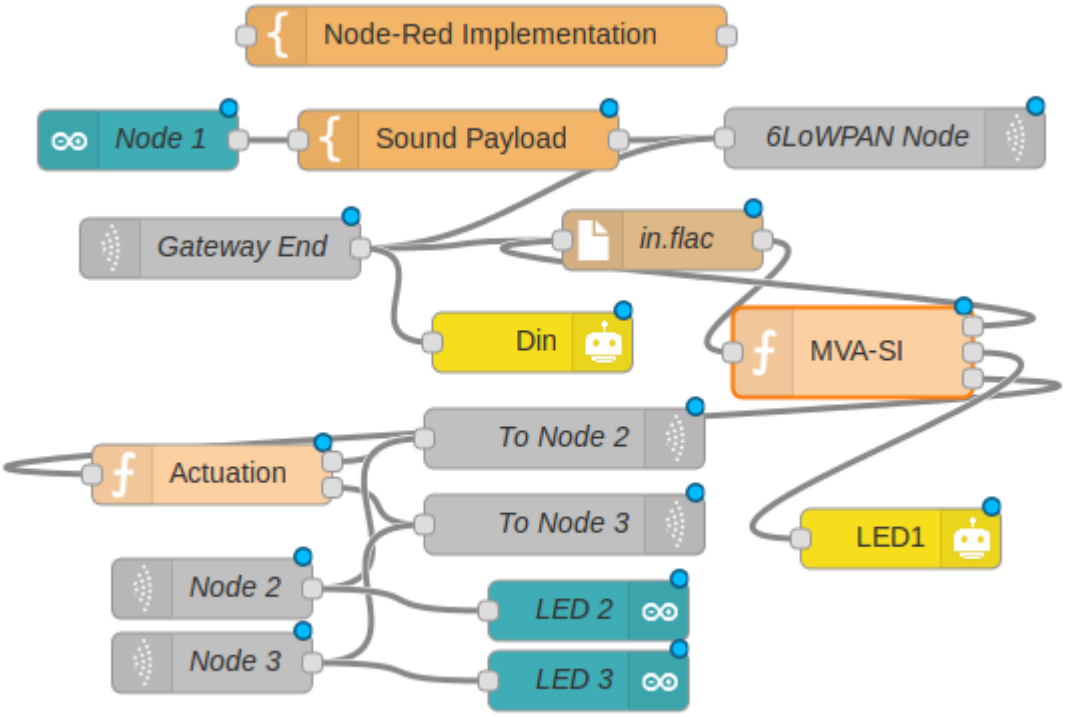


Figure 8.1: Node-Red Deployment of Project

Open sourced under MIT license, Node-Red is the output of collective research efforts from MIT, IBM and vast Linux and Macintosh IoT developer community. Instead of jumping from board to board and layer to layer, Node-Red simplifies and visualizes the design as a single product and connects it all with or without wires using NodeJS. It is one of the initial GUI ever made available to understand and implement Internet of Things on a large spectrum. With accessibility to IBM Bluemix along with support of

all the needed Network and Transport Layer protocols, it effortlessly connects electronics with information. One of the best part of using Node-Red designs is that they can be made available to clone on Git and the UI allows simultaneous debugging and troubleshooting which gives us a sophistication to look at both software and hardware errors in one tiny display.

This approach redefines the possibilities and success rate of IoT and what developer community can do with it. And with such power and performance efficient solutions applications can be boundless while designs can be simple yet seamless.

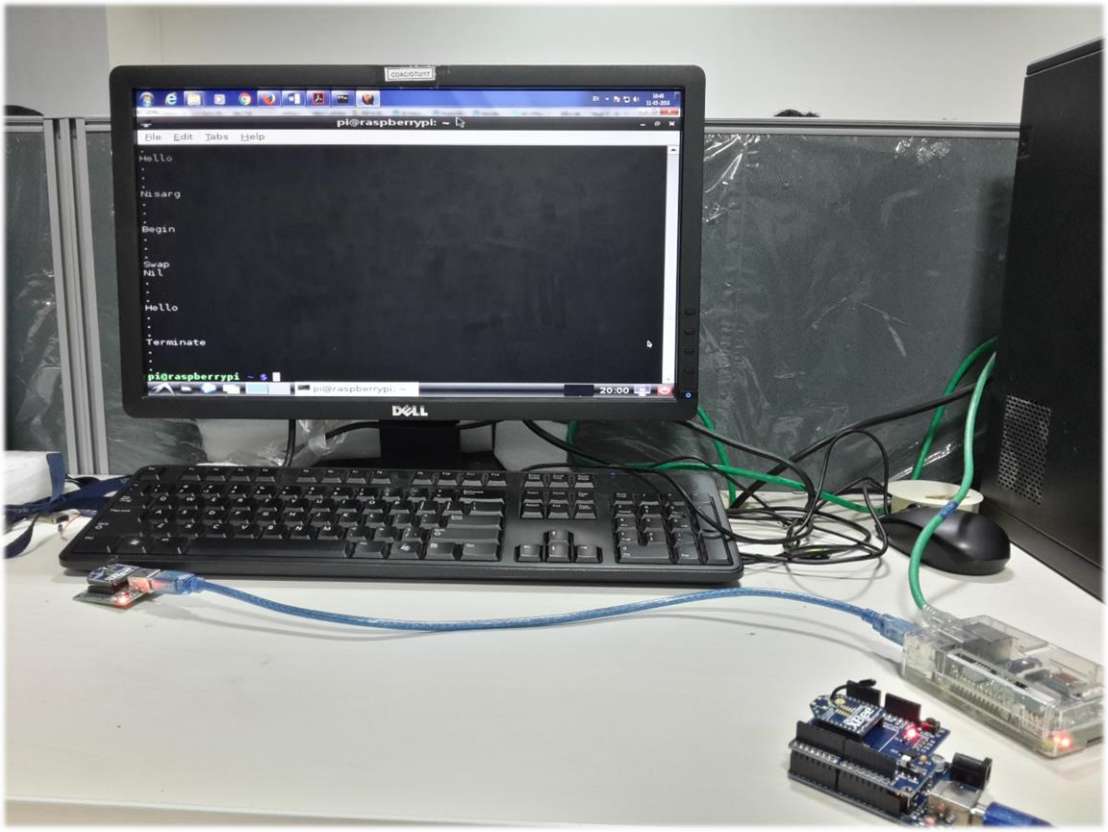


Figure 8.2: Hardware deployment

CHAPTER 9: RESEARCH MANAGEMENT

Aug
- Guide Allocation & Domain Selection ✓
- Project Definition Selection ✓

Sept
- Seminar Completion ✓
- Literature Survey Preparations ✓

Oct.
- Literature Review Presentation & Review Paper ✓
- LRP Corrections ✓

Nov.
- Final Proposed scheme and module design ✓
- ASR Setup and Execution (Testing) ✓

Dec.
- Dissertaion Phase 1 Documantation & presentation ✓
- Logic Building of MVA SI ✓

Jan.
- Research Paper & Emulation of Raspberry Pi ✓
- Testing of 6LoWPAN on Network Simulator ✓

Feb.
- Implementation of pIPv6 stack on Arduino ✓
- Building PocketSphinx on RPi and testing with .wav file ✓

Mar.
- Mid sem review Presentation and Poster ✓
- Completion with prototype and on board testing ✓

Apr.
- Dissertation Thesis Completion ✓
- Single platform integration ✓

May
- Final Submission ✓

CHAPTER 10: CONCLUSION

"If you become comfortable with continuous uncertainty, Infinite possibilities open up in your life"

-Eckhart Tolle, Social Author

At the end of the journey, with successful implementation and singular integration of proposed scheme and designed algorithm the expected betterment has been achieved and comparison with existing systems reflects distinguishable differences which are plotted below.

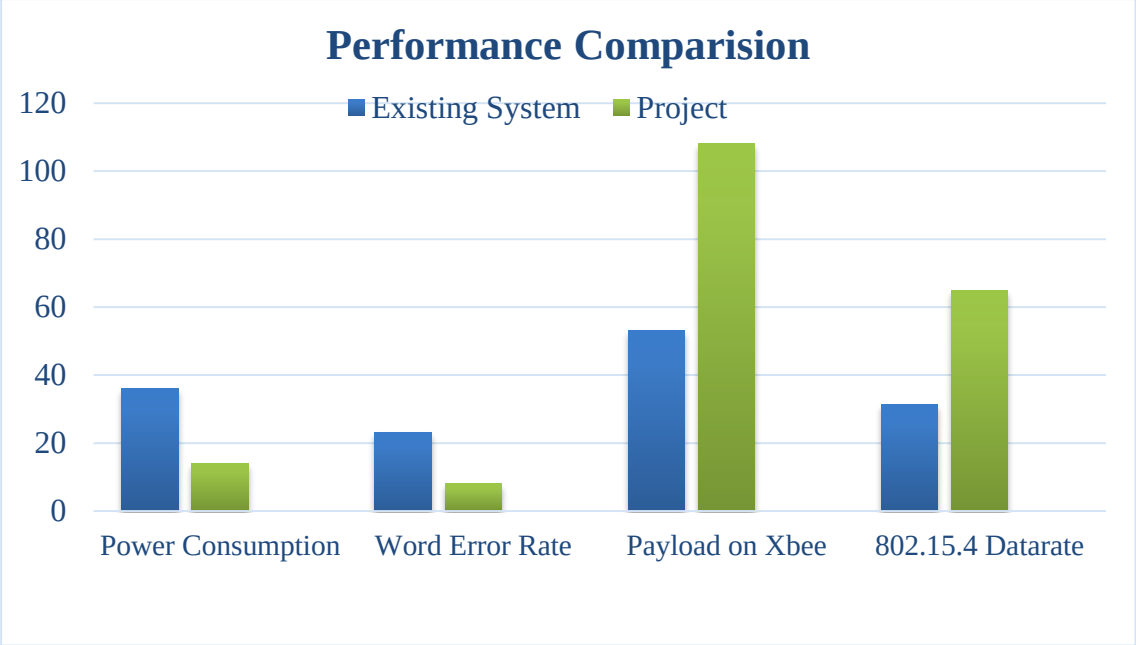


Figure 10.1: Performance evaluation

Graph indicates decrement in Power consumption and Word error rate (Compared to Sphinx 4's WER disclosed by Google in 2014) while increment in payload and data rate compared to traditional Zigbee communication.

These facts propose following opportunities for IoT and ASR developers.

- Extension of this concept to multi-enterprise level which would open possibilities of emergency central remote management and ASR in terms of Big data analytics.
- System design approach can opt for more cost efficient hardware design in terms of controllers instead of Arduino.

- Algorithms can be developed for faster training and update of Acoustic and Language models.
- The scheme itself is a platform designed to carry various applications on top of it, thus creativity and coding are the only judges of what the applications might bring.
- Security algorithm and modified frame headers can be designed for 6LoWPAN which would provide better reliability with least compromise in payload.

R E F E R E N C E

[1] Nisarg M. Vasavada, Swapnil Belhe, **"A power efficient Scheme for Speech Controlled IoT Applications"**, IJERT, vol. 5, Issue 1, p.p 446- 449, January 2016. DOI: http://dx.doi.org/10.17577/IJERTV5IS010382

[2] Andrew Kehler et al. "**Spoken Language Processing**", Prentice Hall New Jersey, ISBN: 978-0131873216.

[3] B. Sing et al. "**Speech recognition with Hidden Markov Model: A review"**, IJCASS 2012.

[4] J.P. Haton, "**Speech analysis for automatic speech recognition: A review**," Proc. 5-th Conf. on Speech Technology and Human-Computer Dialogue, 2009, vol., no., pp. 1-5, June 2009.

[5] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero, "**An Introduction to Voice Search: A look at the technology, the technological challenges, and the solutions**", IEEE Signal Processing Magazine, p.p 29-38, May 2008.

[6] Michelle Cutajar, Edward Gatt et al. "**Comparative study of automatic speech recognition techniques**", IET Signal Process., 2013, Vol. 7, Iss. 1, pp. 25–45

[7] M.J.F. Gales, "**Acoustic Modelling for Speech Recognition: Hidden Markov Models and Beyond?**" IEEE ASRU 2009, p.no 44.

[8] Douglas O'Shaughnessy, "**Acoustic Analysis for Automatic Speech Recognition**", IEEE Proceedings Vol. 101, No. 5, pp. 1038-1043, May 2013.

[9] Jinyu Li, Li Deng et al., "**An Overview of Noise-Robust Automatic Speech Recognition**", IEEE/ACM Transactions on Audio, Speech and Language processing, vol. 22, no. 4, pp. 745-777, April 2014.

[10] Dhwani P. Sametriya and Nisarg M. Vasavada," Comprehensive Survey of MtM Scaling Parameters for deep submicron Analog Mixed Signal Design", in Proceddings of NCRTEMP, March 2016. DOI: 10.13140/ RG.2.1.1179.5608

[11] Badamasi Y. A., "**The working Principal of an Arduino**", 11th International conference on Electronics, Computer and Computing, 2014.

[12] Severence C., "**Eben Upton: Raspberry Pi**", Computer Conversations by IEEE, p.p 14-16, October 2013.

[13] Willie Walker, Paul Lamere et al., "**Sphinx-4: A Flexible Open Source Framework for Speech Recognition**" a white paper by Sun Microsystems Inc., 2004.

[14] Guangguang Ma1, Wenli Zhou et al., "**A Comparison between HTK and SPHINX on Chinese Mandarin**", IEEE Computer Society International joint conference on Artificial Intelligence, p.p 394-397, 2009.

[15] David Huggins-Daines, Mohit Kumar et al., "**Pocketsphinx: A free, Real-time continuous Speech recognition system for hand-held devices**", IEEE ICASSP, pp. 185-188, 2006.

[16] Ferdian Thung, Tegawende F. Bissyande et al., "**Network Structure of Social Coding in GitHub**" IEEE Computer Society 17th European Conference on Software Maintenance and Reengineering, pp. 323-326, 2013.

[17] Wei Li, Tianfan Fu, Jie Zhu, "**An improved i-vector extraction algorithm for speaker verification**", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.

[18] "**6LoWPAN: Wireless Embedded Internet**", Z Shelby, C Bormann, Wiley series in communication networking and distributed systems, ISBN: 978-0-470-74799-5.

[19] "**IoT Reference Model**", A white paper by Cisco Inc.

[20] Dhwani P. Sametriya, Aesha Zala et al., "**LVPLL MCSS Charge Pump in 90nm CMOS for SoCs**", in IRJET volume 03, issue 02, February 2016.

[21] Dhwani P. Sametriya, Nisarg M. Vasavada, **"HC-CPSoC: Hybrid Cluster NoC Topology for Cyber-Physical System-on-Chip"**, 978-1-4673-9338-6/16, IEEE WiSPNET 2016 conference proceedings, p.p 240-243.

APPENDIX 1: PROGRESS

Review Card

Hall No. 12 22/12/15

GUJARAT TECHNOLOGICAL UNIVERSITY
(Established Under Gujarat Act No.: 20 of 2007)
ગુજરાત ટેકનોલોજીકલ યુનિવર્સિટી
(ગુજરાત અધિનિયમ ક્રમાંક : ૨૦/૨૦૦૭ દ્વારા સ્થાપિત)
ESTD - 2007

Master of Engineering

(Dissertation Review Card)

Name of Student : Nisarg M. Vasavada

Enrollment No. : 1 4 1 0 6 0 7 5 2 0 2 2

Student's Mail ID:- nisargmvasavada@gmail.com

Student's Contact No. : 9909740210

College Name : GTU PG School

College Code : 1 0 6

Branch Code : 5 2 **Branch Name :** E.C. (VLSI & Embedded System Design)

Theme of Title : Embedded System Design

Title of Thesis : "A power efficient scheme for Speech controlled IoT devices"

Supervisor's Detail	
Name :	Aditya Kumar Sinha
Institute :	CDAC-ACTS
Institute Code :	106
Mail Id :	saditya@cdac.in
Mobile No. :	020-25503155

Co-supervisor's Detail	
Name :	Mr. Swapnil Belhe
Institute :	CDAC
Institute Code :	
Mail Id :	swapnilb@cdac.in
Mobile No. :	

~1~

Hall No. 72 22/12/15

❖ **Comments For Internal Review (2730002) (Semester 3)**

Sr. No.	Comments given by Internal review panel (Please write specific comments)	Modification done based on Comments
1)	The proposed work is acceptable with minor modifications as discussed.	
2)	The title is covering non-relevant terms, so it needs to get revised	The Inclusion of 6LoWPAN has solved this issue
3)	More emphasis needs to be provided on core ASR	- Literature Review is revised with more emphasis on core ASR.

Particulars	Internal Review Panel	
	Expert 1	Expert 2
Name :	K. G. Vaghela	
Institute :	GEC, Modasa	
Institute Code :	016	
Mobile No. :		
Sign :		

Particulars	Internal Guide Details	
	Expert 1	Expert 2
Name :	Swapnil Bedhe	Aditya Kumar Sinha
Institute :	CDAC, Pune.	CDAC ACTS, Pune.
Institute Code :	106	106
Mobile No. :		020-25503155
Sign :		

~2~

Hall No. 12 22/12/15

Comments For Literature Review (730002) (Semester 3)

Comments Given By Internal Review Panel (Pl Write specific comments in bullets).	Modification done based on Comments
- The proposed work is acceptable with minor modifications as discussed - The title is covering non relavent terms, so it needs to get revised. - More emphasis needs to be provided on core ASR	- The implementation of 6LoWPAN has solved this issue. - Literature review is revised with more emphasis on core ASR.
Internal Review Panel *Name:* K G Vaghela *Sign:* *Inst. Name & Code:* GREM, 016 *Contact No.:*	*Name:* *Sign:-* *Inst. Name & Code:* *Contact No.:*
Guide *Name:* Swapnil Belhe *Sign:*	

Enrollment No. of Student : 1 4 1 0 6 0 7 5 2 0 2 2

❖ **Comments of Dissertation Phase-1 (2730003)** (Semester 3)

Exam Date : 22/12/2015

Hall No : 12

Title : A power efficient scheme for speech controlled IoT Applications

1. Appropriateness of title with proposal. (Yes/ No) Yes
2. Justify rational of proposed research. (Yes/ No) Yes
3. Clarity of objectives. (Yes/ No) Yes

Hall No. 12 22/12/15

Sr. No.	Comments given by External Examiners (DP-I) : (Please write specific comments)	Modification done based on Comments
1.	Accepted the progress.	The progress is as per project plan.

- *Approved* ☑
- *Approved with suggested recommended changes* ☐
- *Not Approved* ☐

} ***Please tick on any one***

➢ ***Details of External Examiners :***

Particulars	*Name*	*University / College Name & Code*	*Mobile No.*	*Sign.*
Expert 1	Bhaskar Thakkar	GCET COII)	9574387285	
Expert 2	Kirtan Rajhdev	GTU (016)	9924343433	
Expert 3				

~ 4 ~

Enrollment No. of Student : 1 4 1 0 6 0 7 5 2 0 2 2

❖ **Comments of Mid Sem Review (2740001)** (Semester 4)

Exam Date : 12 / 04 / 2016

Hall No : 05

Sr. No.	**Comments given by External Examiners :** i) The appropriateness of the major highlights of work done; State here itself if work can be approved with some additional changes. ii) Main reasons for approving the work. iii) Main reasons if work is not approved.	**Modification done based on Comments**
1.	Completion of the Implementation on Ri-pi 3 to be done.	RPi Implemented along with Node-Red is completed.

- *Approved* ☑
- *Approved with suggested recommended changes* ☐
- *Not Approved* ☐

Please tick on any one

➢ ***Details of External Examiners :***

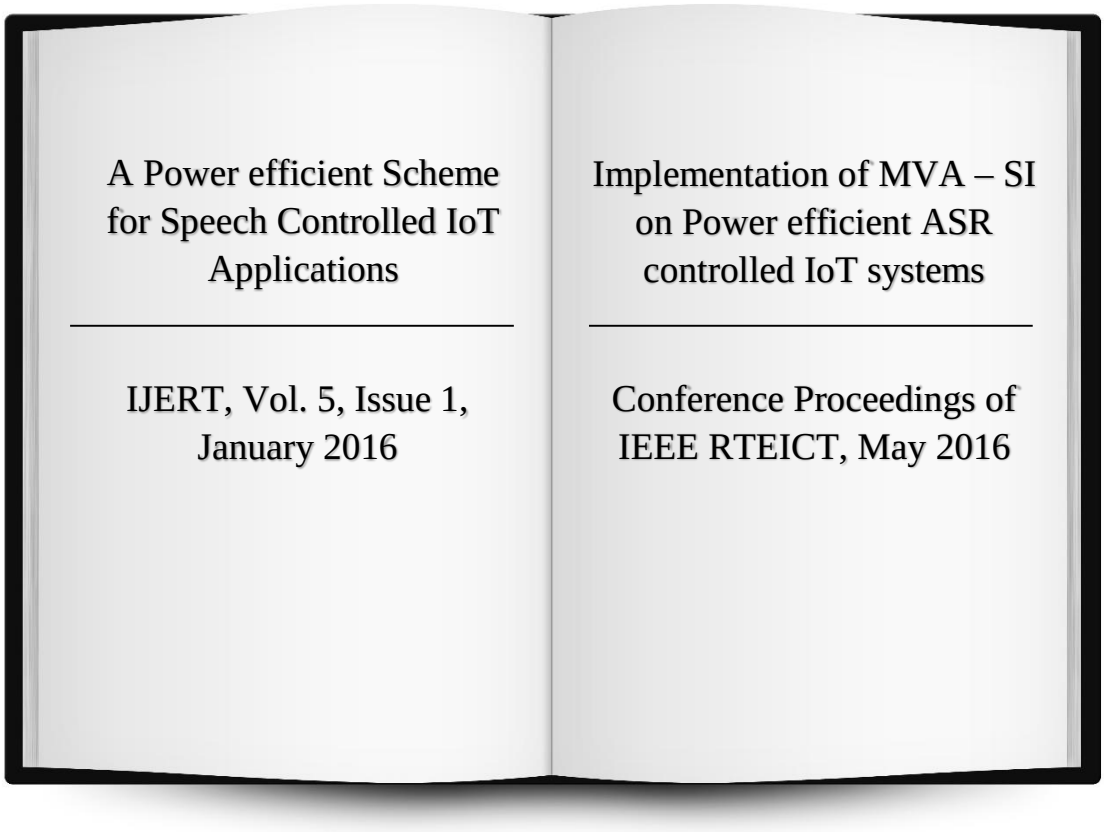
Particulars	*Name*	*University / College Name & Code*	*Mobile No.*	*Sign.*
Expert 1	K.S. Gurumurthy	REVA Univ. Blore-64	9845120845	
Expert 2	KG Vaghela	LDCE, Ahmedabad	9924342433	
Expert 3				

Compliance Report

Table A.1 Compliance Report Table

Event	Comments	Compliance
Literature Review Presentation	The proposed work is acceptable with minor modifications as discussed	Inclusion of 6LoWPAN has solved this issue
	The title is covering irrelevant terms so it needs to be revised	
	More emphasis required on core ASR	Literature review has been revised with more emphasis on core ASR
Dissertation Phase – I	Accepted the progress	The progress is as per the project plan
Mid Semester Review	The completion of implementation on RPi is to be done	Work is now implemented on RPi and Node-Red as planned.

APPENDIX 2: PUBLICATIONS



A Power Efficient Scheme for Speech Controlled IoT Applications

Article *in* International Journal of Engineering and Technical Research · January 2016

DOI: 10.17577/IJERTV5IS010382

1 author:

Nisarg Milan Vasavada

Gujarat Technological University

8 PUBLICATIONS **1** CITATION

SEE PROFILE

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 01, January-2016

A Power Efficient Scheme for Speech Controlled IoT Applications

Nisarg M. Vasavada
GTU PG School – CDAC ACTS
Pune, India.

Swapnil Belhe
CDAC
Pune, India.

***Abstract*— Speech recognition has been a subject of research since decades. Although it has wide applications in Artificial Intelligence and modern user interfaces, when speech processing is applied to embedded systems we also need to consider the constraints which are normally faced and algorithms to overcome the same. While the applications of embedded systems are now being massively focused on Internet of Things, still the prime research concerns are power efficiency and security. Here a state of the art scheme is proposed where speech processing is applied to the constrained IoT applications and the wireless communication is made power efficient. For achieving so, the 6LoWPAN protocol is implemented and Modified Vector Algorithm for Speaker Identification (MVA-SI) is used to increase PDF of the correct interpretation input speech through predefined wake up call.**

Keywords*—Speech Recognition, IoT, Power Efficiency, Speaker Identification

I. INTRODUCTION TO ASR

Sound is an electromagnetic expression that is sensed, interpreted and delivered in a spectrum of frequencies and humans are the species that have learned to modulate it in many different ways. Speech is a combination of modulated sound and interpreted linguistics.[1] Initially concept of Automatic Speech Recognition (ASR) was limited to speech to text conversion which had many overheads that were resolved with time and dedicated research.[2] Today, ASR has taken an advance form where it has become one of the key pillars behind the success of Natural User Interface (NUI) and Artificial Intelligence (AI).[3]

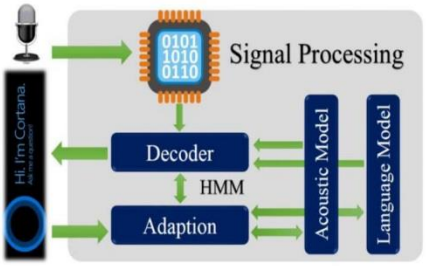


Figure 1. ASR Block Diagram [2]

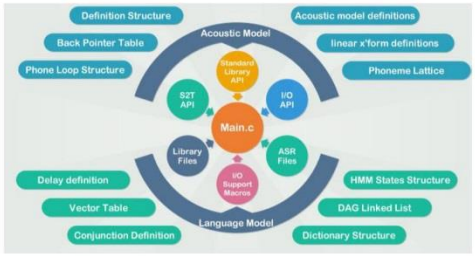


Figure 2. ASR engine file management*

Figure 1 shows the basic building blocks of a system consisting of ASR. The Microphone is used for input which is a transducer converting analog speech waves into voltage pulses that are digitized and provided to speech decoder. Considering the Linux philosophy of understanding systems, the blocks in the ASR engine are files which are interdependent and Operating System (OS) architecture dependent.[15] A file called "main" manages the order in which all the other files, functions and values defined in the files are called and used. Figure 2 describes, the Acoustic Model (AM) and the Language Model (LM) in context of files and internal contents are libraries that are referred and updated which is called training.[12] The AM recognizes phonemes, derives words from lattice and passes to LM which recognizes sentences by measuring delays and applying interpretation of conjunctions. The wider the structures and tables of AM and LM, the more they are trained which in turn makes the ASR engine more efficient.[8]

Mathematically, most of the ASR engines work on a combination of Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM).[4][13] Markov Model is a statistical method of assuming the behavior of the system by the states it passes through and the values obtained from such states. The word "Hidden" focuses on the nature of systems where sequences of occurrence of states are obtained instead of values. This is the key behind word generation from phonemes. Each spoken input has a certain probability of being recognized which is calculated from its Probability Density Function (PDF).[5] The higher the probability, the finer the recognition becomes.

***Note:** Here it is considered that the ASR engine is written in C, if not so, the module management may differ a little but the basic concepts remain the same.[15]

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 01, January-2016

II. IoT: Wireless Embedded Internet (WEI)

As the Internet of personal computers, mobile devices and high performance systems has been growing mature, one more revolution in the Internet was marching on its way–The Internet of Things (IoT).[10] The idea behind the IoT is to make small sensor driven smart devices IP enabled. In an IoT system, data is generated from multiple devices spanning various complexity which are processed in different ways. The basic IoT reference model is derived from conventional OSI network model.[10]

The newest and smallest members of Internet in IoT context are small sensors and actuators which are embedded devices by nature and do not contain scope or requirement of intelligence similar to fully fledged computing systems. Thus the TCP/IP layered approach is over-sufficient for such devices. Routing on Data Link Layer is performed based on corresponding addresses (64-bit EUI-64 or 16-bit short addresses).[19] There is one issue to resolve: as the MAC headers describe the source and destination addresses for the current layer-2 hop, in order to forward the packet to destination MAC, the node needs to know its address. Since each forwarding step overwrites the layer-2 destination address by the address of the next hop and the layer-2 source address by the address of the node doing the forwarding, this information needs to be stored somewhere else. 6LoWPAN defines the mesh header for this. Figure 3 shows layers to be implemented for using 6LoWPAN which is the official standard for wireless embedded internet defined by IETF. Figure 4 shows how the address is compressed in 6LoWPAN where LoWPAN header is larger whereas MAC address is smaller and payload is wider. Since communication is hop to hop IPv6 address is skipped. This approach is especially applicable for sensors and actuators where security is not a primary concern.

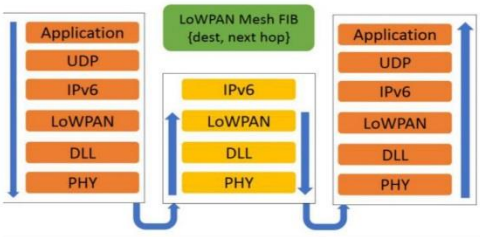


Figure 3. 6LoWPAN layers [19]

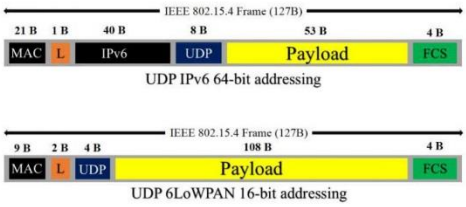


Figure 4. Example of 6LoWPAN Address Compression [19]

III. Constrinted IoT Infrastructure

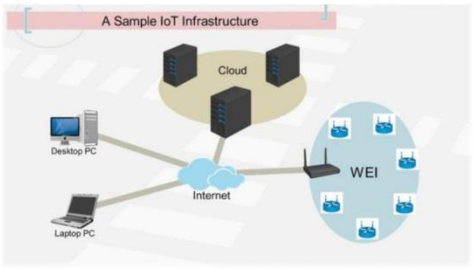


Figure 5. IoT integration with WEI

Figure 5 is a skeleton infrastructure of WEI based IoT approach where small embedded devices form a mesh/star topology based ad hoc network with a router which is connected to the internet which leads it to cloud and billions of other internet enabled devices such as PCs or smartphones.

A. *Observations*

Currently used IoT systems are convenient, accurate, secure and reliable but the audience for such systems is pretty limited, since the embedded domain comprises of many small low-end devices which are envisioned as the basic building blocks and source of success for the IoT. The approach for automating, monitoring and designing those devices is certainly different from regular ones and there need to be schemes and frameworks available which can act as bridges between two distinct implementations. The frameworks should be keeping following aspects into consideration.

1. The I/O interface for actuation and monitoring should be coherent and seamless.
2. The bandwidth, battery life, memory, portability and system cost should be taken into account.
3. The networking should be achieved with as less layers as possible.
4. Speaker independence in ASR should only be applied if the nature of application demands it.[18]

B. *Proposed Solution*

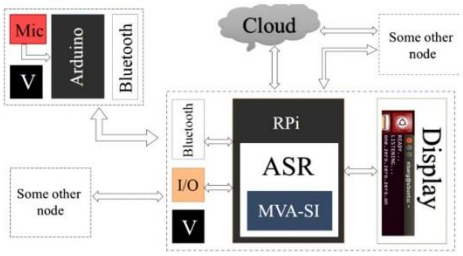


Figure 5. Proposed Scheme Prototyping Model

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 01, January-2016

The proposed scheme is an IoT implementation at constrained embedded level where sensor nodes are provided local IP address while major routing nodes are provided full-fledged internet protocol support which also solves the security issue. For achieving so, the WEI tiny nodes are made independent of OS while mainstream nodes are built on top of OS which provides them full TCP/IP stack.

As the Figure 5 displays, the microphone provides speech output to the Arduino** which transmits it through Bluetooth using 6LoWPAN hopping to the major node which consists of Raspberry Pi (RPi)** and is provided full TCP/IP stack for proper internet connectivity across IPv6.[9][10][11] The I/O are connected to the RPi (either in wired manner or in wireless manner depends on the choice of the developer) which are governed by the inputs received from the voice commands sent from smaller nodes. The display is mainly for prototyping and debugging purpose. The RPi is equipped with open source modified ASR engine. The speaker identification is handled by the "Modified Vector Algorithm for Speaker Identification (MVA-SI)" algorithm which is applied in the acoustic model of the ASR engine. Then the output of AM is provided to LM. The inclusion or removal of LM depends on the nature of application. The MVA-SI actually does nothing but analyzing the phones in the speech context more precisely and giving significant eigenvalues for the vectors generated by the input speech. Phones are the starting of consonants and endings of vowels which are all distinct from each other. The algorithm focuses on the phones and the floating point values generated by them and creates an eigenvector. Such eigenvectors are compared to the stored ones and user is recognized. This is all done by one "Hello" wake up call which initiates the ASR and also recognizes the user. Since the keyword for this operation is predefined, the complexity of the algorithm decreases while the probability of correct speaker identification increases.

C. *MVA-SI Algorithm*

In the i-vector algorithm proposed by N. Dehak, a single space was created for speaker and channel.[16] This approach was later found inefficient and was modified with separate spaces.[14] In this proposed scheme where the system wakes up by only one specified word, Dehak's approach remains simple and efficient. The 512MB RAM and 700MHz ARM11 is sufficient enough to perform ASR initiation and eigenvector calculation for speaker identification thus no unexpected delay is faced.[17]

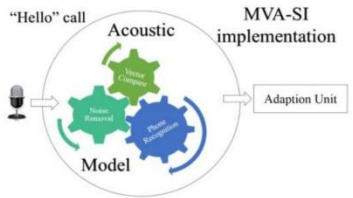


Figure 6. AM with MVA-SI [6][7][8]

***Note:** The open source Hardware platforms such as Arduino and Raspberry Pi are used for prototyping simplicity. For manufacturing purpose standalone controllers are suggested.

The inclusion of MVA-SI as displayed in the Figure 6 fundamentally acts as pre-processing in AM which dismisses probabilities of other speakers and focuses on a defined speaker only. By doing so, the AM needs to process less phonemes, thus less operating cycles are required and the context specific interpretation of words become more accurate. The mathematics of the algorithm is out of the scope of this paper but following equation of supervector M which is used for modelling the GMM justifies the behavior of MVA-SI.

$$M = m + Tw \quad (1)$$

In the equation 1, m is speaker independent base supervector, T is the speaker dependent factor matrix with low ranks and w notifies the count of utterance. For less trained engines, LM can be skipped if only a few tens of words need to be trained whereas in rest of the cases LM will have less eigenvectors to process and will act faster.[20]

IV. Conclusion

Parameter	T.S.	P.S.
Convenient UI		✓
Efficient ASR		✓
Power Efficiency		✓
Less code density		✓
Security	✓	
Authenticity	✓	✓

Table 1. Parameter Survey of schemes

Table 1 is a comparative survey of various parameters and their availability in both Traditional Scheme (T.S) and Proposed Scheme (P.S). Trivially it is the trade-off between constrained implementation and quality features that system developers have to make according to the requirements. The future work can consist of making the proposed scheme secure and more authentic while maintaining other parameters on constrained embedded system design domain.

References

[1] Andrew Kehler et al. "Spoken Language Processing", Prentice Hall New Jersey, ISBN: 978-0131873216.
[2] J.P. Haton, "Speech analysis for automatic speech recognition: A review," Proc. 5-th Conf. on Speech Technology and Human-Computer Dialogue, 2009, vol., no., pp. 1-5, June 2009.
[3] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero, "An Introduction to Voice Search: A look at the technology, the technological challenges, and the solutions", IEEE Signal Processing Magazine, p.p 29-38, May 2008.
[4] Michelle Cutajar, Edward Gatt et al. "Comparative study of automatic speech recognition techniques", IET Signal Process., 2013, Vol. 7, Iss. 1, pp. 25–45
[5] M.J.F. Gales, "Acoustic Modelling for Speech Recognition: Hidden Markov Models and Beyond?" IEEE ASRU 2009, p.no 44.
[6] Douglas O'Shaughnessy, "Acoustic Analysis for Automatic Speech Recognition", IEEE Proceedings Vol. 101, No. 5, pp. 1038-1043, May 2013.
[7] Jinyu Li, Li Deng et al., "An Overview of Noise- Robust Automatic Speech Recognition", IEEE/ACM Transactions on Audio, Speech and Language processing, vol. 22, no. 4, pp. 745-777, April 2014.
[8] Zhen-Hua Ling, Shi-Yin Kang et al. "Deep Learning for Acoustic Modeling in Parametric Speech Generation", IEEE Signal Processing Magazine, pp. 35-52, May 2015.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 01, January-2016

[9] Badamasi Y. A., "The working Principal of an Arduino", 11th International conference on Electronics, Computer and Computing, 2014.
[10] "IoT Reference Model", A white paper by Cisco Inc, 2014.
[11] Severence C., "Eben Upton: Raspberry Pi", Computer Conversations by IEEE, p.p 14-16, October 2013.
[12] Sunyi Hu, David Mulvaney, S. Datta, "Modification of Sphinx 3 for Embedded System Implementation", International Conference on Multimedia, Signal Processing and Communication Technologies, p.p 137-140, 2011.
[13] Willie Walker, Paul Lamere et al., "Sphinx-4: A Flexible Open Source Framework for Speech Recognition" a white paper by Sun Microsystems Inc., 2004
[14] Anthony Chun, Jenny X. Chang et al., "ISIS: An Accelerator for Sphinx Speech Recognition", IEEE Symposium on Application Specific Processors, pp. 58-61, June 2011.
[15] David Huggins-Daines, Mohit Kumar et al., "Pocketsphinx: A free, Real-time continuous Speech recognition system for hand-held devices", IEEE ICASSP, pp. 185-188, 2006.
[16] Wei Li, Tianfan Fu, Jie Zhu, "An improved i-vector extraction algorithm for speaker verification", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.
[17] Hynek Hermansky, "Multistream Recognition of Speech: Dealing With Unknown Unknowns", IEEE Proceedings Vol. 101, no. 5, pp. 1076-1088, May 2013.
[18] Feng Deng, Chang-Chun Bao, "Speech enhancement based on Bayesian decision and spectral amplitude estimation", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.
[19] "6LoWPAN: Wireless Embedded Internet", Z Shelby, C Bormann, Wiley series in communication networking and distributed systems, ISBN: 978-0-470-74799-5.
[20] Javier Tejedor, Doroteo T. Toledano et al., "Spoken term detection ALBAYZIN 2014 evaluation: overview, systems, results, and discussion", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.

IJERT International Journal of Engineering Research & Technology
ISSN : 2278 - 0181, www.ijert.org
(Published by : ESRSA Publication)

Certificate Of Publication

This is to certify that

Nisarg M. Vasavada

Has published a research paper entitled

A Power Efficient Scheme for Speech Controlled IoT Applications

In IJERT, Volume. 5, Issue. 01 , January - 2016

Registration No: IJERTV5IS010382 **Date:** 22-01-2016 **Chief Editor, IJERT**

International Journal of Engineering Research & Technology

Implementation of MVA-SI on Power Efficient ASR controlled IoT Systems

Nisarg M. Vasavada[1], Dhwani P. Sametriya[1], Dipika S. Vasava[1], Devanshi M. Desai[2]
[1]Department of VLSI & ESD, GTU PG School, Ahmedabad, India.
[2]Department of HPC, GTU PG School, Ahmedabad, India.

***Abstract* — Speech processing is implemented based on application specific accuracy requirements and regardless of the implementation scheme the ultimate goal remains the same, to enhance speech processing engines up to an extent where natural language can be interpreted accurately in case of word or phoneme recognition and context prediction. ASR training and adaption is always application specific and in the considered application the ASR engine is trained for a power efficient IoT application. The nodes communicate via 802.15.4 6LoWPAN which makes wireless sound transmission very lightweight and power efficient. MVA-SI is a customized version of i-vector speaker identification algorithm which identifies the speaker with system wake up call. In this paper, the logic and implementation of MVA-SI are discussed in detail along with elaboration of prototype system architecture.**

***Keywords*— ASR, MVA-SI, Arduino, 6lowpan, Raspberry Pi**

I. INTRODUCTION TO ASR

Automatic Speech Recognition (ASR) is one of the most intuitive methods of interaction with electronic devices which are mostly targeted to be a part of Artificial Intelligence community [1]. Although the output is fundamentally similar to old Speech-to-Text (STT) converters, the recent ASR engines are more context specific, accent neutral and user centric which makes them more efficient and reliable. The basic ASR system consists of a microphone input, a general purpose or dedicated digital signal processor, memory, display device and IO [2]. Many methods, models and algorithms have been developed since then among which many have succeeeded and others could only find their space in research documentations and old text books. One of the pioneer algorithms which is still used in most of the speech recognition engines is known as Hidden Markov Model (HMM) [3].

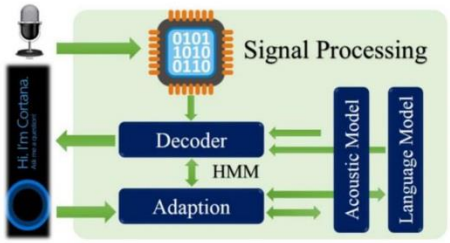


Figure 1 Block diagram of ASR

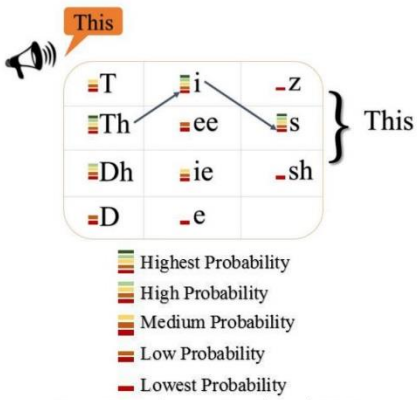


Figure 2 Word recognition through HMM

The Hidden Markov Model (HMM) is a variant of FSM (Finite State Machine) having discrete hidden states, output state (alphabet), transition probabilities (next alphabet), output (phoneme) probabilities, and initial state probabilities [5]. Instead of having an observable current state it produces outputs with certain probabilities. In the example above, word "This" is broken down in respective phonemes and most probable entries are selected. Other probable utterances with respective priorities are also shown. So, when a user with general accent speaks an utterance starting with "This", mathematically there is also a low probability of him having spoken 'D' instead of 'Th' which depends on frequency created by him but the training of Acoustic model enables ASR to determine the correct phonemes which in turn become key to determine correct word, phrase and eventually the whole utterance. In advanced ASR such as CMU's sphinx, GMM which stands for Gaussian Mixture Model also plays an important role in the STT algorithm by complementing HMM. While HMM finds the probability of each state (in this case, phoneme), GMM finds the probability of weighted sums of various HMM states[10] [12]. This part of ASR is known as Language Model (LM) and depth HMM of GMM depends on the rigour required by the Acoustic Model (AM) and LM. For example, one word utterances may not even need LM while continuous language processors may need more than one

The Project was performed at CDAC ACTS, Pune as a part of Masters Dissertation.

layers of LM, for this reason N-gram ASR engines are used since Sphinx2.

II. Low Power IoT

The idea behind the IoT is to make the small and sensor driven embedded devices pervasive and IP (Internet Protocol) enabled, and to integrate them as an essential part of the Internet by making them authentic, area specific and quasi-real-time data suppliers[16]. In an IoT system, data is generated from multiple devices spanning a spectrum of complexities (varying from sensors to servers) which are processed in different ways (varying from small ASICs to sophisticated parallel processors), transmitted to different locations (varying from users to servers), and acted upon by applications (varying from mobile application to enterprise applications). The basic IoT reference model is stacked up of seven levels where each level is defined with its terminology that can be used for standardization to create a globally acceptable frame of reference [16]. As the concept of "Things" has a huge ocean of devices in its own category it is one of the most irrational approaches to treat all of the devices equally. Thus since small sensors and actuators which are embedded devices by nature do not contain scope or requirement of intelligence similar to fully fledged computing systems the TCP/IP layered approach is over-sufficient for such devices and it is efficient to remove wrappers of unwanted layers and implement a scaled down and lightweight protocol. In the figure shown below, 6LoWPAN UDP header is compared to normal 64-bit UDP header. By removing the wrappers of IPv6 addressing the devices are identified by MAC addresses which are unique for every device and is provided by the manufacturer [15]. This allows 127 byte frame to store more payload which in turn gives it flexibility to send more data over the same frame and use less frames for the total amount of data. The 6LoWPAN stands for Low Power IPv6 which means all of the nodes will be a part of IPv6 only and the communication will work hop-hop due to neighborhood discovery [15].

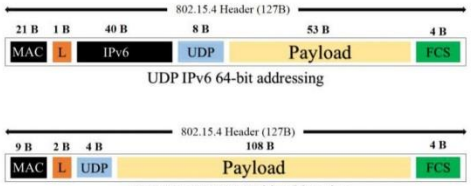


Figure 3 6LoWPAN header [15]

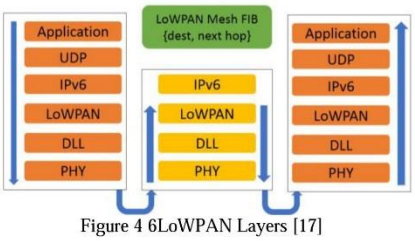


Figure 4 6LoWPAN Layers [17]

III. Test System Architecture

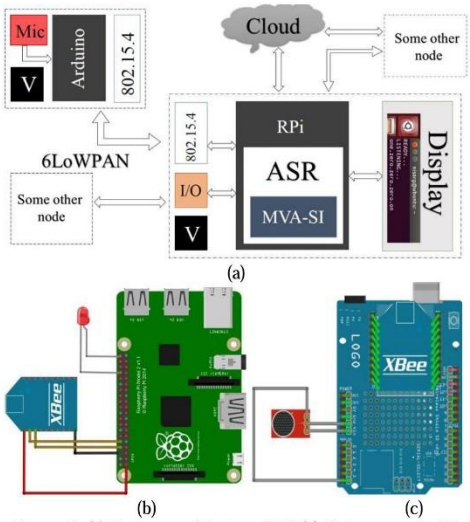


Figure 5: (a) System architecture [17] (b) Gateway node [9] (c) small node [8]

MVA-SI stands for Modified Vector Algorithm for Speaker Identification which is an efficient solution for optimization of bandwidth, battery life, memory, portability and system cost which is applied on a test system shown in figure (a) [17]. The proposed system is an IoT implementation at constrained embedded level where small nodes connect with gateways using 6LoWPAN. Gateways are provided proper internet over multiple protocols such as IpV6 on network layer and UDP or MQTT (for lightweight UDP) are applied. To communicate with other routers and gateways, TCP or SCTP are applied. The small nodes from 6LoWPAN communicate via hop to hop in personal area so naturally the number if hops are supposed to be limited. Their MAC addresses are stored in the devices only so they do not need to know respective IPv6 IPs. Such a node here is designed using Arduino development board as shown in (c) where mic is connected via GPIO and wireless connectivity is achieved using 802.15.4 xbee radio. On the other hand, Gateway is implemented using Raspberry Pi 2 as shown in (b) with another 802.15.4 xbee radio. It is equipped with CMU Pocketsphinx ASR engine which enables it to recognize the speech utterance provided to Arduino as an input. The generic raspbian Jessie incorporates full featured internet connectivity which can be achieved using Ethernet or Wi-Fi module. The wide range of 1.2km of Xbee radios allow small nodes to remain mobile flexibly whereas larger payload of 6LoWPAN increases data (in Kbps) rate as following.

$$6LoWPAN\ Datarate = \frac{6LoWPAN\ payload\ *\ Xbee\ Datarate}{Xbee\ Payload}$$

$$6LoWPAN\ Datarate = \frac{108 * 250}{53} = 519.09$$

IV. MVA-SI

A. *Algorithm and Process Flow*

In section 1, it was explained how words are recognized using HMM and GMM in ASR. Speaker identification semantically improves the ASR accuracy by reducing WER (Word Error Rate) and creating more context specific relevance. The MVA-SI actually does nothing but analyzing the phones in the speech context more precisely and giving significant eigenvalues for the vectors generated by the input speech. Phones are the starting of consonants and endings of vowels which are all distinct from each other. The algorithm focuses on the phones and the floating point values generated by them and creates an eigenvector. Such eigenvectors are compared to the stored ones and user is recognized. This is all done by one "Hello" wake call which initiates the ASR and also recognizes the user. Since the keyword for this operation is predefined the complexity of the algorithm decreases while the efficiency increases which is the main aim of not just this but ANY system regardless of the domain and purpose. The requirement of such algorithm arises when words are unconventional and mostly not stored in either acoustic or language models. One of such examples is a name. MVA-SI stores the user specific data and provides binary probability to words. This case is different from AM or LM since here either the word is black or white means either the probability is 0 or one. Such a scenario is expressed and resolved mathematically in the expressions below. P1 and P2 are tentative probabilities of utterance whereas 'A' and 'B' are binary probabilities of MVA-SI (It is worth noting that the number of cases may differ for other utterances). P1 and P2 are obtained using trained AM and n-gram LM. Trivially, one will be float and other will be zero so comparison would be easy. Thus P_{final} is derived from MVA-SI.

$$P1 = [P(1)|P(2)] * [P(12)|P(3)] * [P(12 \dots n-1)|P(n)] \quad \dots (1)$$

$$P2 = [P(1)|P(2)] * [P(12)|P(3)] * [P(12 \dots n-1)|P(n)] \quad \dots (2)$$

$$If\ P1 = P2$$
$$P1' = P1 * A\ ;\ P2' = P2 * B$$
$$If\ P1' > P2'$$
$$Pfinal = P1' \quad \dots (3)$$

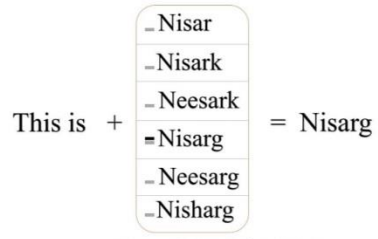


Figure 6 ASR with MVA SI

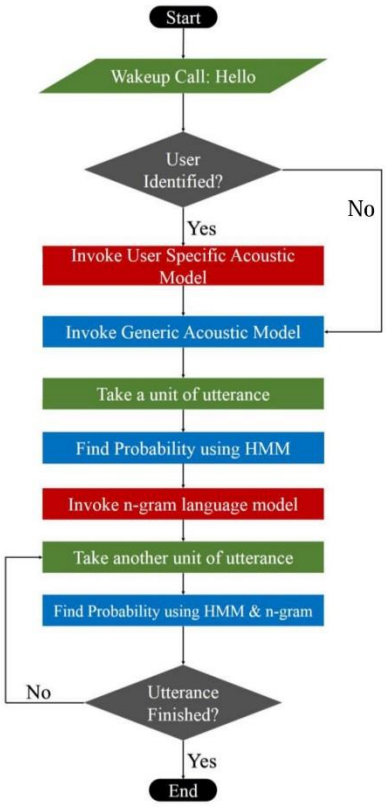


Figure 7 MVA-SI Flowchart

The flowchart elaborates the execution of MVA-SI in above mentioned environment. It is governed by the APIs written in C only. First of all the ASR engine is initiated. The RPi receives sound input from Arduino node which will be seamless as bitrate over 500 is sufficient enough to stream sound without having the buffer partially filled. And since this is the only user enabled key process that will be running at a particular instance the bandwidth will not be shared between the applications. From the "Hello" .wav file, eigenvector is created which, if matches a certain range (provided environmental and natural variance of ±10ve, the pointer of that value will lead to the user member of the user entry union. This is how the user is recognized. We can say that MVA-SI is built on top of i-vector speaker verification algorithm but is a constrained and thus more efficient version of it [14]. Consequently, the AM and LM are invoked and STT conversion takes place and the process runs in a loop till the

utterance is finished. Sphinx STT APIs put EOU character at the end of every array of vectors automatically so it is not user or programmer's concern.

B. Implementation

In figure 8, ASR implementation on Ubuntu Linux is displayed which was performed for testing purpose. It is visible that while trying to perform continuous speech recognition, words that are pronounced without gaps are misinterpreted due to ASR engine being unfamiliar with the accent of the user. On the other hand the same experiment with the same results was performed on RPi2 and afterwards, MVA-SI APIs were added to it. The logic and codes of the APIs are out of the scope of the paper. As it can be seen in figure 9, MVA-SI is running as a separate process and word "hello" was spoken using different accents and different voices which was interpreted accurately. On the other hand 6LoWPAN communication was established between two nodes made of Arduino. To do so, pIPv6 library was imported to Arduino toolkit and serial IPv6 in LoWPAN header APIs were used. The baud rate was kept as default 9600 which is enough to transmit sounds and sampling rate for ASR was kept as 48000. For testing of 6LoWPAN connectivity over Arduino the states of LED were transmitted and it becomes crucial because of the less RAM available in the Arduino which is 2kb only. The difference in results are portrayed in tabular format in conclusion section.

Figure 8 ASR Testing on Ubuntu Linux

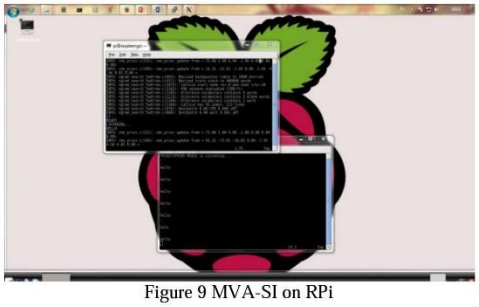


Figure 9 MVA-SI on RPi

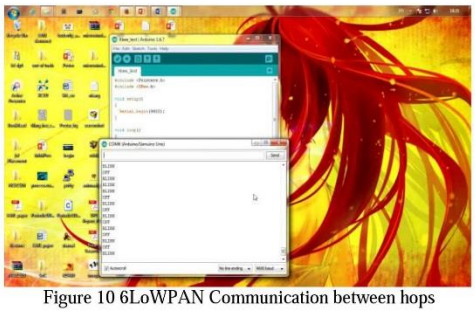


Figure 10 6LoWPAN Communication between hops

CONCLUSION

Thus by implementing MVA-SI on a power efficient ASR controlled IoT system, it is observed that user specific speech recognition improves with decrement in WER (Word Error Rate). The fact is furnished in the table below.

Parameter	Existing System	MVA-SI
Power Efficient	No	Yes
WER	23%[7]	8%
Payload on Xbee	53B	108B
Load on Gateway	More	Less
Load on X'mitter	More	Less
802.15.4 Bandwidth	~250kbps[16]u	519kbps

Table 1: Comparative study of two implementations

REFERENCES

[1] Andrew Kehler et al. "Spoken Language Processing", Prentice Hall New Jersey, ISBN: 978-0131873216.

[2] J.P. Haton, "Speech analysis for automatic speech recognition: A review," Proc. 5-th Conf. on Speech Technology and Human-Computer Dialogue, 2009, vol., no., pp. 1-5, 18-21 June 2009.

[3] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero, "An Introduction to Voice Search: A look at the technology, the technological challenges, and the solutions", IEEE Signal Processing Magazine, p.p 29-39, May 2008.

[4] Michelle Cutajar, Edward Gatt et al., "Comparative study of automatic speech recognition techniques", IET Signal Process., 2013, Vol. 7, Iss. 1, pp. 25–46

[5] M.J.F. Gales, "Acoustic Modelling for Speech Recognition: Hidden Markov Models and Beyond?", IEEE ASRU 2009, p.no 44.

[6] Douglas O'Shaughnessy, "Acoustic Analysis for Automatic Speech Recognition", IEEE Proceedings Vol. 101, No. 5, May 2013, pp. 1038-1044

[7] C. Y. Fook, M. Hariharan et al., "A Review: Malay Speech Recognition and Audio Visual Speech Recognition", International Conference on Biomedical Engineering (ICoBE) pp. 479-485 ,27-28 February 2012,Penang

[8] Badamasi Y. A., "The working Principal of an Arduino", 11th International conference on Electronics, Computer and Computing, 2014.

[9] Severence C., "Eben Upton: Raspberry Pi", Computer Conversations by IEEE, p.p 14-16, October 2013.

[10] Willie Walker, Paul Lamere et al., "Sphinx-4: A Flexible Open Source Framework for Speech Recognition" a white paper by Sun Microsystems Inc., 2004.

[11] Guangguang Ma1, Wenli Zhou et al., "A Comparison between HTK and SPHINX on Chinese Mandarin", IEEE Computer Society International joint conference on Artificial Intelligence, p.p 394-397, 2009.

[12] David Huggins-Daines, Mohit Kumar et al., "Pocketsphinx: A free, Real-time continuous Speech recognition system for hand-held devices", IEEE ICASSP, pp. 185-188, 2006.

[13] Ferdian Thung, Tegawende F. Bissyande et al., "Network Structure of Social Coding in GitHub" IEEE Computer Society 17th European Conference on Software Maintenance and Reengineering, pp. 323-326, 2013.

[14] Wei Li, Tianfan Fu, Jie Zhu, "An improved i-vector extraction algorithm for speaker verification", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.

[15] "6LoWPAN: Wireless Embedded Internet", Z Shelby, C Bormann, Wiley series in communication networking and distributed systems, ISBN: 978-0-470-74799-5.

[16] "IoT Reference Model", A white paper by Cisco Inc.

[17] Nisarg M. Vasavada, Swapnil Belhe, " A power efficient Scheme for Speech Controlled IoT Applications", IJERT, vol. 5, Issue 1, p.p 446-449, January 2016. DOI: http://dx.doi.org/10.17577/IJERTV5IS010382

APPENDIX 3: CONSOLIDATED REPORT

Date	:	May 10th, 2016.
Enrollment Number	:	141060752022
Branch Name	:	VLSI and Embedded Systems Design
Name of Student	:	Vasavada Nisarg Milan
Title of Thesis	:	A Power Efficient Scheme for Speech Controlled IoT Applications
Theme of Thesis	:	Internet of Things, Speech Recognition
Name of Supervisor	:	Aditya Kumar Sinha

Signature of Guide
Aditya Kumar Sinha
P.T.O., CDAC ACTS, Pune.

Overview of Project:

The scheme which is proposed and demonstrated with prototype implementation is a result of collective obsession to simplify its essential components and to create the most efficient design possible. It is an approach that could not exist without innovation over many disciplines. Internet of Things is one of the most familiar and prospective trends in research and development community. It defined the theme of dissertation. To make it power and bandwidth efficient, we centralized the focus to most delicate as well as untouched members of IPv6 community, wireless low configuration nodes. 6LoWPAN reinforces the stability and efficiency over IoT in terms of battery life and power. On the gateway end, MVA-SI is an algorithm that is not just designed but deliberately engineered to exploit the sophistication of Raspberry Pi hardware along with accuracy of Pocket Sphinx ASR engine and to add the luxury of artificial intelligence to Internet of Things. While keeping the lower layers tightly scheduled and controlled, the upper layers are left open to expand the applications to virtually endless extent.

The Problem:

While exploiting the pervasive adaption of IoT, it is worth noting that Zigbee and other wireless communication protocols falling under 802 series are least power efficient and are not suitable for state and signal transfer. Operating such networks with speech instead of commands is certainly interesting but smartphones as ASR modules are not suitable for such systems as ASR run on low priorities. And since the span of commands would be limited to a few, speaker dependent independent universal acoustic and language models are not only power consuming (in terms of operation cycles and resource management) but also less efficient. These problems as a whole are targeted to pursue most optimized design covering these fields yet.

The Solution:

In terms of solution, A power efficient scheme for speech controlled IoT applications is designed which includes usage of 6LoWPAN instead of other network layer protocols. As a part of the project, a new and modified version of i-vector algorithm has been developed which compromises its scope to a single word (here “Hello”). The Hello word also acts as a wakeup call for the ASR system which would put it into sleep in any otherwise case.

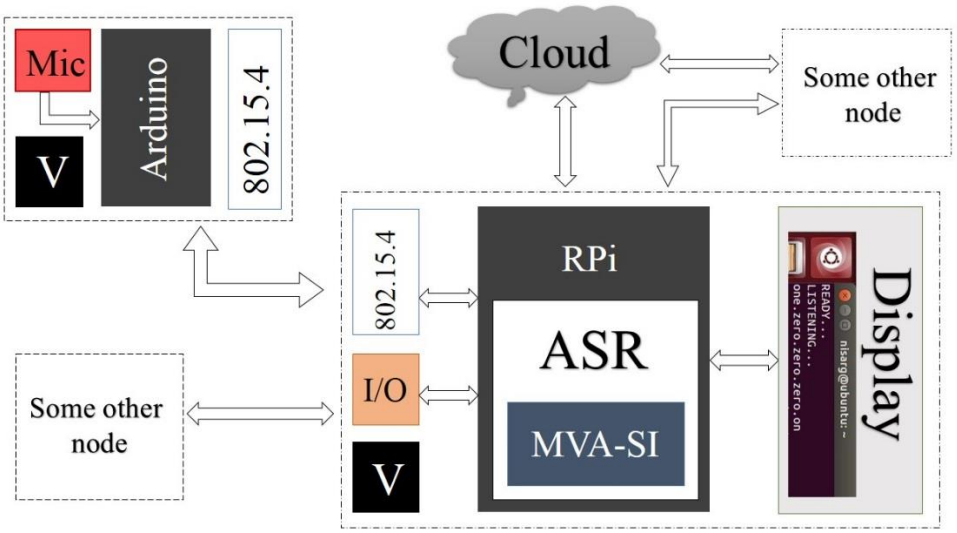


Figure A.1: Proposed Scheme

The above Figure shows proposed design on block diagram level which is implemented in the project. The whole system is then deployed using Node-Red which gives end to end singularity and ease of debugging to the system. The results are noted and compared in graphical format shown below.

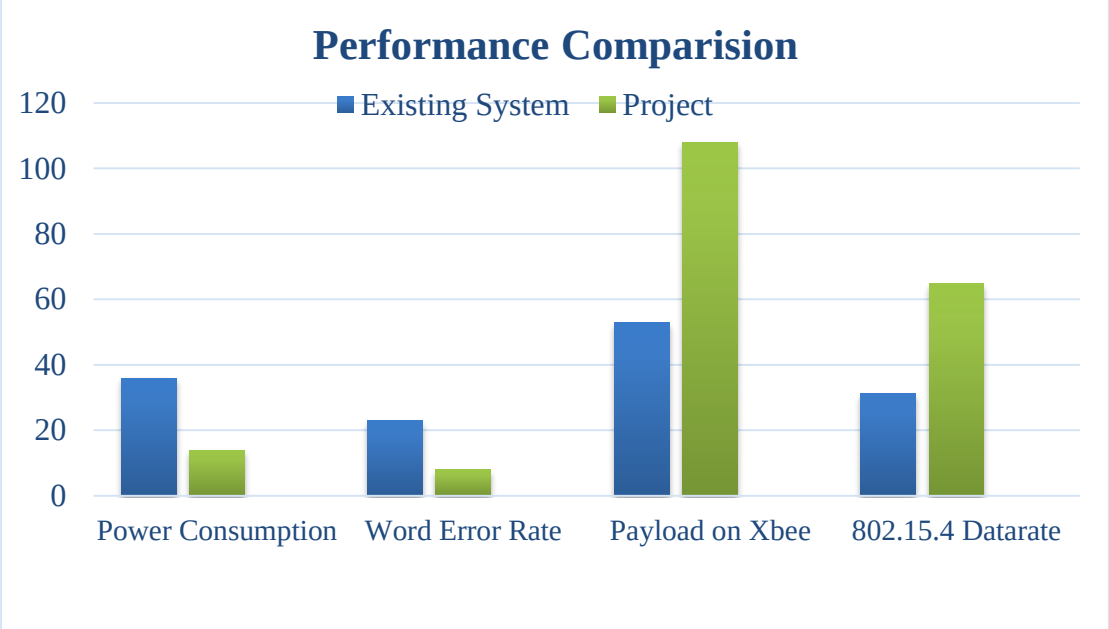
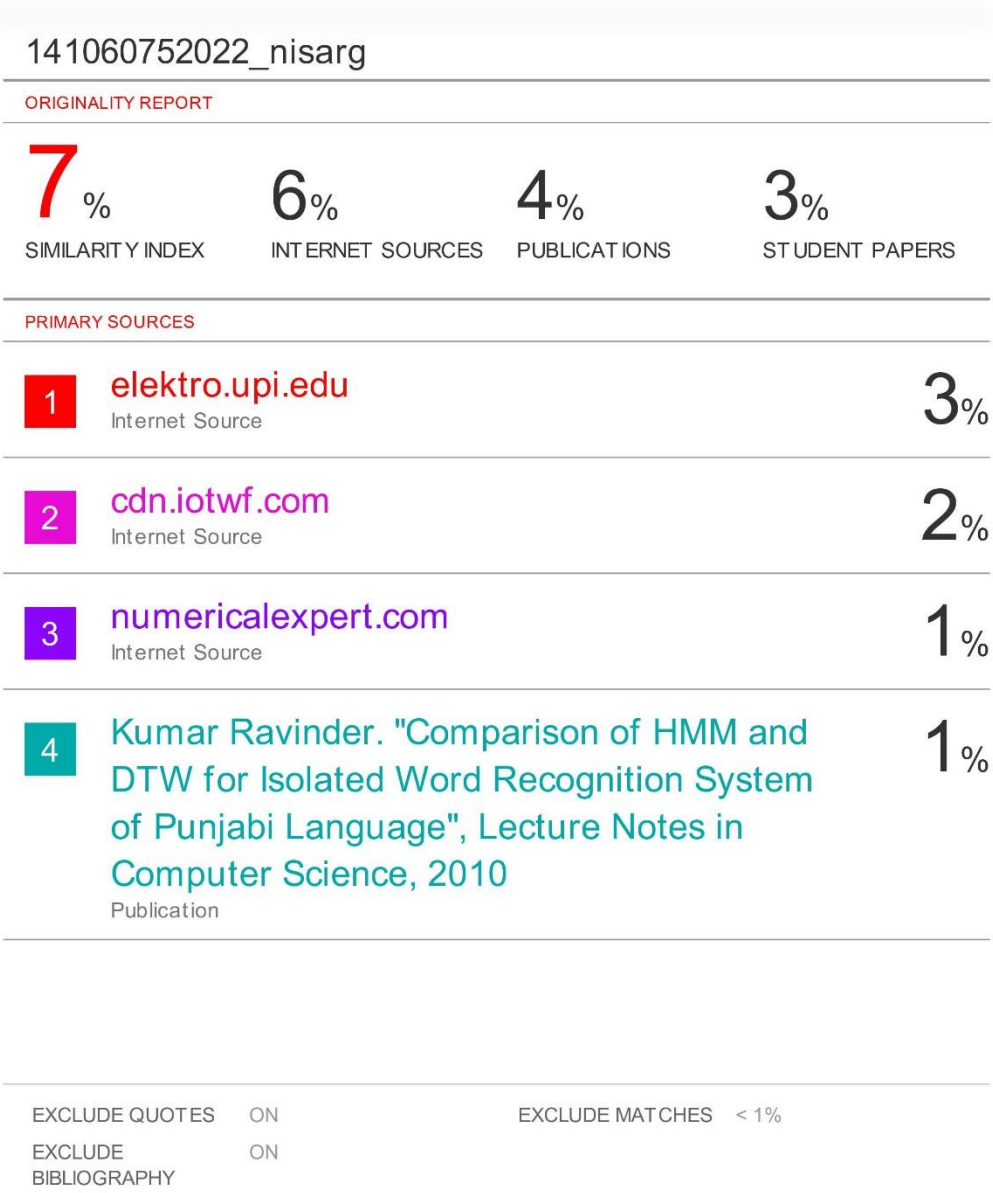


Figure A.2: Performance Graph

APPENDIX 4: ORIGINALITY



141060752022_nisarg

ORIGINALITY REPORT

7%	6%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	elektro.upi.edu Internet Source	3%
2	cdn.iotwf.com Internet Source	2%
3	numericalexpert.com Internet Source	1%
4	Kumar Ravinder. "Comparison of HMM and DTW for Isolated Word Recognition System of Punjabi Language", Lecture Notes in Computer Science, 2010 Publication	1%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 1%
EXCLUDE BIBLIOGRAPHY	ON		

Signature of Guide
Aditya Kumar Sinha
P.T.O., CDAC ACTS, Pune.