Project report on

DESIGN AND IMPLEMENTATION OF FAST JOT

A THESIS SUBMITTEDIN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING

in

Electronics and Communication Engineering

by

KAMATAM JEMIMA	**- 100513735014**
VURIBINDI KAVYA	**-100513735017**
MALLAKUNTLA SAI ANUSHA	**-100513735033**
P V S VISHNU PRIYA	**-100513735049**

Under the esteemed guidance of

Dr. B.R.Rajendra Naik

Assistant professor

(Dept. Of Electronics and Communication Engineering)

Osmania University

Department of Electronics and Communication Engineering

University College of Engineering (A),

Osmania University, 2013-2017

Department of Electronics and Communication Engineering

University College of Engineering (Autonomous)

Osmania University,Hyderabad-500007

CERTIFICATE

This is to certify that the thesis entitled "**DESIGN AND IMPLEMENTATION OF FAST JOT**" submitted by **K. Jemima (100513735014), V. Kavya (100513735017), M. Sai Anusha (100513735033) and P. V. S. Vishnu Priya (100513735049)** in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in **ELECTRONICS AND COMMUNICATION** at the University College of Engineering (A), Osmania University is an authentic work carried out by them under my supervision and guidance. To the best of my knowledge the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any degree or diploma.

Project guide	**Head of the Department**
Dr.B.R.RajendraNaik	**Dr.B.R.RajendraNaik**
Asst.Professor,ECE	**Asst.Professor,ECE**
UCE,OU,Hyderabad	**UCE,OU,Hyderabad.**

DECLARATION

This is to certify that the work reported in the present thesis titled "DESIGN AND IMPLEMENTATION OF FAST JOT" is a record of work done by us in VLSI Laboratory in Department of Electronics and Communication, UCE(A)OU.

No part of the thesis is copied from books / journals / internet and wherever the portion is taken; the same has been duly referred in the text. The reported are based on the project work done entirely by us and not copied from any other source.

K.JEMIMA	(100513735014)
V.KAVYA	(100513735017)
M.SAI ANUSHA	(100513735033)
P.V.S. VISHNU PRIYA	(100513735049)

Acknowledgement

We express our sincere thanks to Prof.S.Sameen Fatima, Principal, Universiy college of Engineering , Osmania University for providing the opportunity and support to carry out our project.

We gratefully acknowledge Dr.B.R.RajendraNaik, Head of department, Electronics and Communication Engineering, for his encouragement and advice during the course of this work and for his guidance, scholarly advice and inspiration offered in an amiable and pleasant manner that helped us to complete the project successfully.

We would also like to thank all the faculty members and staff of ECE department, UCE OU, Hyderabad for their valuable support and encouragement without which this couldn’t have been possible.

K.JEMIMA	(100513735014)
V.KAVYA	(100513735017)
M.SAI ANUSHA	(100513735033)
P.V.S. VISHNU PRIYA	(100513735049)

ABSTRACT

Traditional note taking often feels laborious for students because it bores them mentally while it exhausts them physically. It is a time consuming process as the students have to manage both listening and taking down the notes at the same time. So, we have come up with an idea of taking notes with ease. A Raspberry Pi, microphone and a webcam are used to achieve this. The entire setup can be placed in a classroom.

An android app with different set of predefined commands is built. This app is used to give commands to Raspberry Pi through pubnub. Pubnub, a global Real-Time Network enables software developers to rapidly build and scale real-time apps by providing the cloud infrastructure, connections and key building blocks for real-time interactivity.

According to the command received, pi will communicate either with an attached microphone to recognize the speech or with the webcam to take a picture of the board. The speech will then be converted into text which is written into a document along with the pictures. This .docx file will be uploaded to drop box. This enables the students to access the notes easily.

Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Introduction

The purpose of taking notes is to help students gain a firmer understanding of the concepts the class covers.Taking structured notes requires more time than jotting down points as you absorb them. Notes may be taken by the students as an outline approach, in which a main idea is noted, followed by bullet points that cover ideas associated with the main idea.

This can make it hard to keep up with a speaker, and you might miss important information in the process and lose your concentration as you focus on form rather than content. If you revert to your personal shorthand to keep up, it might be hard to decipher notes in the future.

Students give the command and the respective subject using an app and the notes will be uploaded to the drop box along with subject name, date and time.

1.2 Motivation

Besides rapidly growing technology and automation, we still prefer the old traditional way of taking notes. Structured note taking does not work for everyone.

A speaker or lecturer who insists on structured note taking can inconvenience those who prefer listening attentively and referring to a written text later. Some students prefer scribbling points in a stream-of-consciousness way rather than following a rigid structure.

Therefore in order to bring revolution in jotting down the points by the students we have come up with an idea of designing and implementing fast jot which decreases the effort of the students in taking notes.

1.3Objective

Some students are able to gain this understanding through simple reading or listening, but others need the additional mental effort of writing down what they read or hear to better retain the information.

Main objective of this project is to provide organized notes and to get information of the lecture received in the class. To reduce the time as well as manual labor dedicated for taking down notes and the students can concentrate on the lecture. It is eco-friendly procedure which helps to cut down the number of pages used. To implement and deploy a speech access based system for students to reduce efforts in taking notes.

1.4 Organization of the Thesis

Chapter 1 is the introduction which includes the importance of our project 'Design and Implementation of Fast Jot'.

In Chapter 2, the complete outline of Raspberry Pi is given. Chapter 3, 4 and 5 discuss the components (USB Sound Card, Microphone and Webcam respectively) which are used in our project. Chapter 6 completely deals with the software called MIT App Inventor which we have used to build the required app. All the blocks and windows that are used to build the app are described in detail. Chapter 7, 8 and 9 explain the features of Cloud Computing, PubNub and Dropbox respectively in detail.

Chapter 10 discusses each block of the block diagram of our Project 'Fast jot'. The complete algorithm is given.Chapter 11 includes the program that we have written to implement this Fast jot. In Chapter 12, the corresponding results are presented. Finally the thesis ends with conclusion and Future Scope in Chapter 13.

CHAPTER 2

RASPBERRY PI

2.1 Introduction

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles

Several generations of Raspberry Pis have been released. The first generation (Raspberry Pi 1 Model B) was released in February 2012. It was followed by a simpler and inexpensive model A.

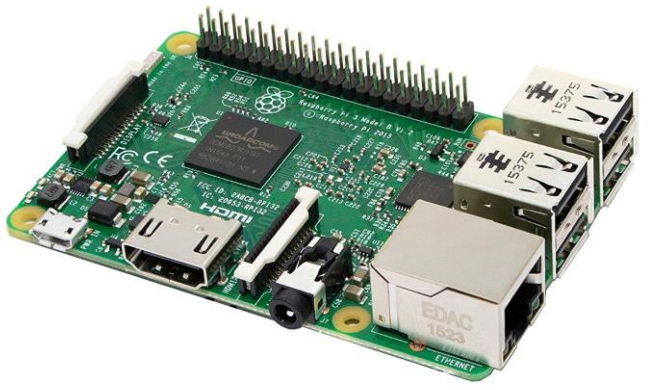


Fig 2.1 Raspberry Pi

In 2014, the foundation released a board with an improved design in Raspberry Pi 1 Model B+. These boards are approximately credit-card sized and represent the standard mainline form-factor[2]. Improved A+ and B+ models were released a year later. The Raspberry Pi 2 which added more RAM was released in February 2015. Raspberry Pi 3 Model B released in February 2016, is bundled with on-board WiFi, Bluetooth and USB boot capabilities. Raspberry Pi boards are priced between US$5–35. As of 28 February 2017, the Raspberry Pi Zero W was launched, which is identical to the Raspberry Pi Zero, but has the Wi-Fi and Bluetooth functionality of the Raspberry Pi 3 for US$10The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support.

2.2 Raspberry Pi Block Function:

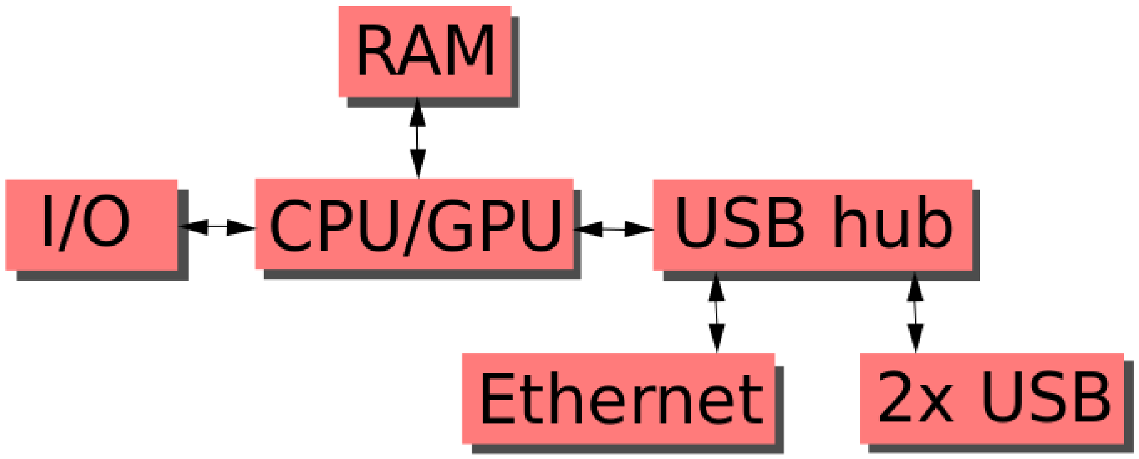


Fig2.2 Raspberry Pi Block Diagram

This block diagram shown in fig 2.2 depicts Models A, B, A+, and B+. Model A, A+, and the Pi Zero lack the Ethernet and USB (Universal Serial Bus) hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the System on a Chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-point USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

2.2.1Processor

The Raspberry Pi 2 uses a 32-bit 900 MHz quad-core ARM Cortex-A7 processor.The Broadcom BCM2835 SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in first generation smartphones (its CPU is an older ARMv6 architecture), which includes a 700 MHz ARM1176JZF-S processor, VideoCore IV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.The Raspberry Pi 2 uses a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor (as do many current smartphones), with 256 KB shared L2 cache.The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

2.2.2 RAM

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p frame buffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC). Comparatively the Nokia 701 uses 128 MB for the Broadcom Video Core IV. For the new Model B with 512 MB RAM initially there were new standard memory split files released(arm256_start.elf, arm384_start.elf, arm496_start.elf) for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). But a week or so later the RPF released a new version of start.elf that could read a new entry in config.txt (gpu_mem=xx) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start.elf worked the same for 256 and 512 MB Raspberry Pi s.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

2.2.3 Peripherals

The current Model B boards incorporate four USB ports for connecting peripherals.The Raspberry Pi may be operated with any generic USB computer keyboard and mouse. It may also be used with USB storage, USB to MIDI converters, and virtually any other device/component with USB capabilities.
Other peripherals can be attached through the various pins and connectors on the surface of the Raspberry Pi.

1. Peripherals are described below:

- **SoC(System On Chip):** Broadcom BCM2835 media processor (datasheet, BCM2835 datasheet errata, unofficial pinout, BCM2835 Register documentation - based on GPU source code) system-on-chip featuring:
- **CPU core**: ARM1176JZF-S ARM11 core clocked at 700 MHz; ARM VFP. The ARM11 core implements the ARMv6 Architecture. For details on ARM instruction

sets and naming conventions, see ARM architecture and List of ARM microprocessor cores.

- **GPU core**: A Broadcom VideoCore IV GPU providing OpenGL ES 1.1, OpenGL ES 2.0, hardware-accelerated OpenVG 1.1, Open EGL, OpenMAX and 1080p30 H.264 high-profile decode. There are 24 GFLOPS of general purpose compute and a bunch of texture filtering and DMA infrastructure. Eben worked on the architecture team for this and the Raspberry Pi team are looking at how they can make some of the proprietary features available to application programmers
- **DSP core**: There is a DSP, but there isn't currently a public API (Liz thinks the BC team are keen to make one available at some point) thread
- 256 MiB of (Hynix Mobile DDR2 or Samsung Mobile DRAM) SDRAM (or 512 MB Mobile DRAM on later boards). The RAM is physically stacked on top of the Broadcom media processor (package-on-package technology). Here is a photo of the SDRAM (left) and BCM2835 (right) ball grid arrays on JamesH's finger. You are looking at the bottom side. The BCM2835 top side has a land grid array which matches the SDRAM ball grid array. Here is a highly magnified side view of the SDRAM stacked on top of the BCM2835 stacked on top of the PCB PoP stack (you can see why its job can only be done by robots!).

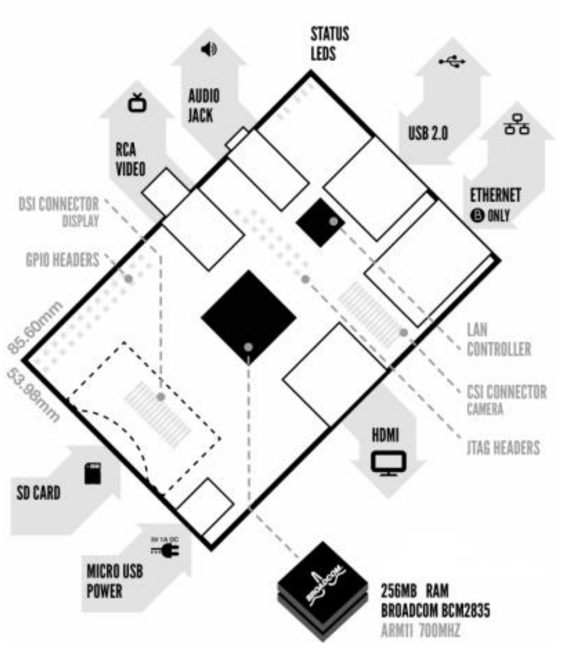


Fig2.3 Raspberry Pi Peripherals

2. LAN9512 (Data Brief | Data Sheet) (Model B) providing:

- 10/100 Mbit/s Ethernet (Auto-MDIX)[7]

- 2x USB 2.0

3. S1: Micro USB power jack (5 V - Power Only)
4. S2: DSI interface. 15-pin surface mounted flat flex connector, providing two data lanes, one clock lane, 3.3 V and GND.
5. S3: HDMI connector providing type A HDMI 1.3a out
6. S4: Composite Video connector: RCA
7. S5: MIPI CSI-2 interface. 15-pin surface mounted flat flex connector.
8. S6: Audio connector: 3.5mm stereo jack (output only)
9. S8: SD/MMC/SDIO memory card slot (underside)
10. S7: Either 1x USB 2.0 (Model A) 2x USB 2.0 (Model B)
11. P1: 26-pin (2x13) 2.54 mm header expansion, providing: see Low-level peripherals

- 8 GPIOs at 3.3 V
- 2-pin UART serial console, 3.3 V TTL (debug); or 2 GPIOs at 3.3 V
- I²C interface (3.3 V); or 2 GPIOs at 3.3 V
- SPI interface (3.3 V); or 5 GPIOs at 3.3 V
- 3.3 V, 5 V and GND supply pins
- ARM JTAG (if pins are reconfigured in software - on Revision1.0 boards one signal would also need to be taken from S5)
- I²S interface (if pins are reconfigured in software, hardware hack may be required[5])

12. P2: 8-pin 2.54 mm header expansion (header not fitted on Revision 2.0 boards), providing GPU JTAG (ARM11 pinout, pin 7 is nofit for locating)
13. P3: 7-pin 2.54 mm header expansion (header not fitted), providing LAN9512 JTAG (pin 6 is nofit for locating)
14. P4: 10/100 Mbit/s RJ45 Ethernet jack (Model B)
15. P5: 8-pin (2x4) 2.54 mm header expansion (header not fitted), on the bottom of the board, providing: see Low-level peripherals (Model B Revision 2.0 and Model A boards only)

- 4 GPIOs at 3.3 V
- 3.3 V, 5 V and GND supply pins
- Second I²C interface (3.3 V) (if pins are reconfigured in software)
- I²S interface (if pins are reconfigured in software)
- Handshake signals for the UART on the P1 header (if pins are reconfigured in software)

16. P6: 2-pin 2.54 mm header expansion (header not fitted), providing an option to connect a hardware-reset button (Revision 2.0 boards only)
17. TP1 and TP2: Test Points giving access to +5 V and GND respectively
18. 5 Status LEDs :

- **D5(Green)** - SDCard Access (via GPIO16) - labelled as "OK" on Model B Rev1.0 boards and "ACT" on Model B Rev2.0 and Model A boards
- **D6(Red)** - 3.3 V Power - labelled as "PWR" on all boards
- **D7(Green)** - Full Duplex (LAN) (Model B) - labelled as "FDX" on all boards
- **D8(Green)** - Link/Activity (LAN) (Model B) - labelled as "LNK" on all boards
- **D9(Yellow)** - 10/100 Mbit/s (LAN) (Model B) - labelled (incorrectly) as "10M" on Model B Rev1.0 boards and "100" on Model B Rev2.0 and Model A boards

19. Board size: 85.60 mm x 53.98 mm. Overall height expected to be less than 25 mm. Production boards measure 85.0 mm x 56.0 mm.

- A Model B between the highest points (USB connector to card slot) measured 21 mm.
- A Model A between the highest points (composite video connector to card slot) measured 18 mm.

20. Weight

- Alpha board weighs approx. 55 g.
- A sample model B weighed 39.45 g.

6 layer PCB

2.3Performance

The Raspberry Pi 3, with a quad-core Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.

Raspberry Pi 2 includes a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It is described as 4–6 times more powerful than its predecessor. The GPU is identical to the original. In parallelized benchmarks, the Raspberry Pi 2 could be up to 14 times faster than a Raspberry Pi 1 Model B+.

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997–99. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capabilities of the Raspberry Pi are roughly equivalent to the performance of the Xbox of 2001.

The LINPACK single node compute benchmark results in a mean single precision performance of 0.065 GFLOPS and a mean double precision performance of 0.041 GFLOPS for one Raspberry Pi Model-B board.A cluster of 64 Raspberry Pi Model B computers, labeled "Iridis-pi", achieved a LINPACK HPL suite result of 1.14 GFLOPS (n=10240) at 216 watts for c. US$4000.

2.3.1Over clocking

The CPU chips of the first and second generation Raspberry Pi board did not require cooling, such as a heat sink, unless the chip was over clocked, but the Raspberry Pi 2 SoC may heat more than usual under over clocking.

Most Raspberry Pi chips could be over clocked to 800 MHz, and some to 1000MHz. There are reports the Raspberry Pi 2 can be similarly over clocked, in extreme cases, even to 1500MHz (discarding all safety features and over-voltage limitations). In the Raspbian Linux distro the over clocking options on boot can be done by a software command running "sudoraspi-config" without voiding the warranty. In those cases the Pi automatically shuts the over clocking down if the chip reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and over clocking settings (voiding the warranty); an appropriately sized heat sink is needed to protect the chip from serious overheating.

Newer versions of the firmware contain the option to choose between five over clock ("turbo") presets that when used, attempt to maximize the performance of the SoC without impairing the lifetime of the board. This is done by monitoring the core temperature of the chip, the CPU load, and dynamically adjusting clock speeds and the core voltage. When the demand is low on the CPU or it is running too hot the performance is throttled, but if the CPU has much to do and the chip's temperature is acceptable, performance is temporarily increased with clock speeds of up to 1 GHz depending on the individual board and on which of the turbo settings is used.The Raspberry Pi Zero runs at 1 GHz.

2.3.2Networking

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the Model B and B+ the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4GHz WiFi 802.11n (150Mbit/s) and Bluetooth 4.1 (24Mbit/s) based on Broadcom BCM43438 Full MAC chip with no official support for Monitor mode but implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Ethernet port.

2.3.3 Video

The early Raspberry Pi 1 Model A, with an HDMI port and a standard RCA composite video port for older displays

The video controller can emit standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions and older standard CRT TV resolutions. As shipped (i.e., without custom overclocking) it can emit these: 640×350 EGA; 640×480 VGA; 800×600 SVGA; 1024×768 XGA; 1280×720 720p HDTV; 1280×768 WXGA variant; 1280×800 WXGA variant; 1280×1024 SXGA; 1366×768 WXGA variant; 1400×1050 SXGA+; 1600×1200 UXGA; 1680×1050 WXGA+; 1920×1080 1080p HDTV; 1920×1200 WUXGA.

Higher resolutions, such as, up to 2048×1152, may work or even 3840×2160 at 15 Hz (too low a frame rate for convincing video)[4]. Note also that allowing the highest resolutions does not imply that the GPU can decode video formats at those; in fact, the Pis are known to not work reliably for H.265 (at those high resolutions), commonly used for very high resolutions (most formats, commonly used, up to Full HD, do work).

Although the Raspberry Pi 3 does not have H.265 decoding hardware, the CPU is more powerful than its predecessors, potentially fast enough to allow the decoding of H.265-encoded videos in software. The GPU in the Raspberry Pi 3 runs at a higher clock frequencies of 300 MHz or 400 MHz, compared to previous versions which ran at 250 MHz.

The Raspberry Pis can also generate 576i and 480i composite video signals, as used on old-style (CRT) TV screens and less-expensive monitors through standard connectors – either RCA or 3.5 mm phone connector depending on models. The television signal standards supported are PAL-BGHID, PAL-M, PAL-N, NTSC and NTSC-J.

2.3.4 Real-time clock

None of the current Raspberry Pi models have a built-in real-time clock, so they are unable to keep track of the time of day independently. As a workaround, a program running on the Pi can retrieve the time from a network time server or from user input at boot time, thus knowing the time while powered on. To provide consistency of time for the file system, the Pi does automatically save the time it has on shutdown, and re-installs that time at boot.

A real-time hardware clock with battery backup, such as the DS1307, which is fully binary coded, may be added (often via the I²C interface).The latest model of Raspberry Pi 3 includes 802.11n WiFi, Bluetooth 4.0, and a quad-core 64-bit ARM Cortex A53 running at 1.2 GHz. It’s a usable desktop computer. Available now at the usual Pi retailers for $35.

The Raspberry Pi 3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) and has complete compatibility with Raspberry Pi 1 and 2.We recommend the Raspberry Pi 3 Model B for use in schools, or for any general use. Those wishing to embed their Pi in a project may prefer the Pi Zero or Model A+, which are more useful for embedded projects, and projects which require very low power.

CHAPTER 3

USB SOUND CARD

3.1 Introduction

A sound card (also known as an audio card) is an internal expansion card that provides inputand output of audio signals to and from a computer under control of computer programs. Theterm sound card is also applied to external audio interfaces used for professional audioapplications. Typical uses of sound cards include providing the audio component formultimedia applications such as music composition, editing video or audio, presentation,education and entertainment (games) and video projection.

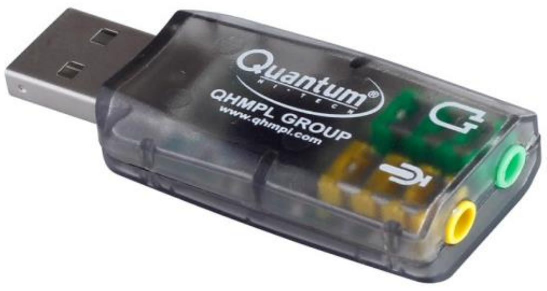


Fig3.1 USB Sound Card

A sound card needs to convert the bits from a computer. For going from the computer tospeakers, a Digital-to- Analog Converter (DAC) receives the train of bits from the computerand transform it into a wave form. The DAC will take samples from the bit train and createthe shape of a wave from it. This wave becomes the electric current to be sent to speakers.The newly created wave starts as more of a ripple. A built-in amplifier makes it grow. Theamplifier provides more power, feeding the wave until it is large enough for speakers. Theelectric current needs enough power to drive the magnet. The speakers and headphones ridethe powered current to make wonderful orchestral performances, rocking concerts andinspiring speeches to be heard.

An external sound card usually has a way to input sound as well. A microphone can capturesound and send it to the sound card. The sound card will then amplify the electric current inorder to translate it to 1’s & 0’s for the computer. This process is done through Analog-to-Digital Converters (ADC).

3.2Types of Sound Cards

- PCI Sound Card
- USB Interface
- Firewire Interface
- PCMCIA / PC Cardbus

3.2.1 PCI Sound Card

PCI sound cards are fitted inside your computer, in a spare PCI slot on the motherboard. These can be a very cost effective and stable solution for a music computer setup. Somecome with a “breakout box” or “breakout cable”, which attaches to the sound card on theoutside of the computer and allows for easy connection to other equipment. These soundcards can only be used in desktop machines, though, and not laptops.

3.2.2 USB Interfaces

These are devices that connect to your computer via USB and function as external soundcards. They are very user-friendly and many are Plug and Play compatible. Suitable forlaptops or desktop computers with USB ports

3.2.3 Firewire Interfaces

FireWire interfaces work in much the same way to USB interfaces, except that FireWireconnections offer a faster data transfer rate and so they can handle more ins and outs at lowerlatency.

3.2.4 PCMCIA / PC Cardbus

This type of card goes into the PCMCIA slot on a laptop computer, and provides a high rate of data transfer. Some include breakout boxes with extra connections.

CHAPTER 4

MICROPHONE

4.1 Introduction

Microphone is the most critical element in the recording chain. Every sound not createdpurely electronically must be transduced through a microphone in order to be recorded. There is a bewildering array of available microphones and each may be optimally suited to a different application. One of the most important areas of knowledge for a recording engineer is the proper application of microphones: selection and placement both require a solid understanding of how microphones work to obtain the best possible results. This is ultimately achieved through experience, but understanding the properties of the various microphones can accelerate the process.

Fig 6 Microphone

Microphones can be divided into 2 types according to the functional principle:

- Condenser mics: These more expensive devices of a higher quality are used in recording studios.

- Dynamic mics. These cheap devices are easy to set up and use.

The main types of commonly used transducers are dynamic, where a conductor moves within a magnetic field in response to the force applied by an incident sound wave, and condensor, where one plate of a capacitor moves relative to the second plate.

Dynamic microphones depend on the principle that a changing magnetic field induces current flow in a conductor in that field. Unlike the dynamic microphone, the capacitor microphone requires a supplied source of voltage to operate. It also requires electronics inthe microphone to convert the capacitance change to a voltage. The advantage of the capacitor microphone comes largely from its low mass moving element, making it easier for small, high-frequency air pressure changes to generate a proportional output voltage.

Microphones can be divided into 2 large groups according to their purposes:

- Professional microphones used in studios and having high sensitivity, noise reduction and directivity levels.
- Custom microphones used for communication and recording of poor quality.

CHAPTER 5

WEBCAM

5.1 Introduction

A webcam, or web camera, is the loosely used term for any camera that generates images that can be accessed by and displayed on the world wide web through a server. A webcam is essentially just a camera that is connected to a computer, either directly or wirelessly, and gathers a series of images for remote display elsewhere.The webcam basically works by capturing a series of digital images that are transferred by the computer to a server and then displayed to the hosting page. There are even sites that allow users to upload and store their webcam images for free, which many individuals choose for personal use. Wbcams vary in their capabilities and features, and the variances are reflected in price. Some webcams capture a still image only once every 30 seconds, while others provide streaming video by capturing 30 images per second.

Fig 5.1 Webcam

A simple Webcam setup consists of a digital camera attached to your computer, typically through the USB port. The camera part of the Webcam setup is just a digital camera -- there's really nothing special going on there. The "Webcam" nature of the camera comes with the software. Webcam software "grabs a frame" from the digital camera at a preset interval (for example, the software might grab a still image from the camera once every 30 seconds) and transfers it to another location for viewing. If you're interested in using your Webcam for streaming video, you'll want a Webcam system with a high frame rate. Frame rate indicates the number of pictures the software can grab and transfer in one second. For streaming video, you need a minimum rate of at least 15 frames per second (fps), and 30 fps is ideal. To achieve high frame rates, you need a high-speed Internet connection.

Once it captures a frame, the software broadcasts the image over your Internet connection. There are several broadcast methods. Using the most common method, the software turns that image into a JPEG file and uploads it to a Web server using File Transfer Protocol (FTP). You can easily place a JPEG image on any Web page (for information on creating Web pages and adding JPEG images.

5.2Specifictaions

- Focus Distance 4cm ~infinity
- Frame Rate Up to 30 fps
- I/O interface USB 1.1
- Image Control Brightness, Contrast, Hue, Saturation, Gamma, White Balance
- Image Flip Horizontal, vertical
- Image Format RGB 24, I420
- Image Resolution 30 Mega pixels (3264X2448) interpolated
- Image sensor 1/7" CMOS sensor
- Lens View angle 54 degree
- Monitor Type CRT, LCD
- Power Consumptions 160mW typical
- ENVIRONMENT Indoor, Outdoor

CHAPTER 6

MIT APP INVENTOR

6.1 Indroduction

App Inventor for Android is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT).

It allows newcomers to computer programming to create software applications for the Android operating system (OS). It uses a graphical interface, very similar to Scratch and the StarLogo TNG user interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as well as work done within Google on online development environments.

MIT App Inventor is an innovative beginner's introduction to programming and app creation that transforms the complex language of text-based coding into visual, drag-and-drop building blocks. The simple graphical interface grants even an inexperienced novice the ability to create a basic, fully functional app within an hour or less.

6.2. Mission

The MIT App Inventor project seeks to democratize software development by empowering all people, especially young people, to transition from being consumers of technology to becoming creators of it.

6.3 History

Google's Mark Friedman and MIT Professor Hal Abelson co-led the development of App Inventor while Hal was on sabbatical at Google in 2009. Other early Google engineer contributors were Sharon Perl, Liz Looney, and Ellen Spertus. App Inventor runs as a web service administered by staff at MIT's Center for Mobile Learning - a collaboration of MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) and the MIT Media Lab.

Apps are built by working with:

- The **App Inventor Designer**, where components are selected for the app.

- The **App Inventor Blocks Editor**, where program blocks are assembled that specify how the components should behave. Programs are assembled visually, fitting pieces together.
- The app appears on the phone step-by-step as pieces are added to it, so that the work can be tested as the app is built. When the app is done, it is packaged and a stand-alone application is produced to install.

If there is no Android phone, then the apps can be built using the *Android emulator*, software that runs on the computer and behaves just like the phone.

The App Inventor development environment is supported for Mac OS X, GNU/Linux, and Windows operating systems, and several popular Android phone models. Applications created with App Inventor can be installed on any Android phone.

6.4 AppInventor Built-in Blocks:

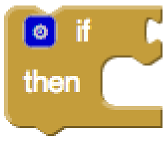
Built-in blocks are available regardless of which components are used in the project. In addition to these *language blocks*, each component in the project has its own set of blocks specific to its own events, methods, and properties. This is an overview of all of the Built-In Blocks available in the Blocks Editor.

- Control blocks
- Logic blocks
- Math blocks
- Text blocks
- Lists blocks
- Colors blocks
- Variables blocks
- Procedures blocks

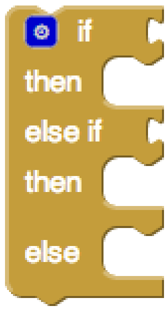
The blocks that are used in the project include:

6.4.1 Control blocks:

- **if & if else**



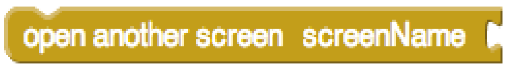
Tests a given condition. If the condition is true, performs the actions in a given sequence of blocks; otherwise, the blocks are ignored.



Tests a given condition. If the result is true, performs the actions in the -do sequence of blocks; otherwise tests the statement in the -else if section. If the result is true, performs the actions in the -do sequence of blocks; otherwise, performs the actions in the -else sequence of blocks.

- **Open another screen**

Opens the screen with the provided name.



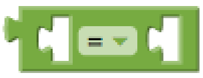
6.4.2 Logic Blocks

- **true**

Represents the constant value true. Use it for setting boolean property values of components, or as the value of a variable that represents a condition.

- **false**

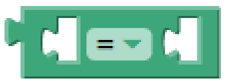
Represents the constant value false. Use it for setting boolean property values of components, or as the value of a variable that represents a condition.

- = 

Tests whether its arguments are equal.

- Two numbers are equal if they are numerically equal, for example, 1 is equal to 1.0.
- Two text blocks are equal if they have the same characters in the same order, with the same case. For example, banana is not equal to Banana.
- Numbers and text are equal if the number is numerically equal to a number that would be printed with that text. For example, 12.0 is equal to the result of joining the first character of 1A to the last character of Teafor2.
- Two lists are equal if they have the same number of elements and the corresponding elements are equal.

Acts exactly the same as the = block found in Math



- ≠

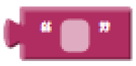
Tests to see whether two arguments are not equal.

6.4.3Math Blocks

- **Basic Number Block**

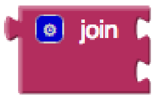
Can be used as any positive or negative number (decimals included). Double clicking on the "0" in the block will allow you to change the number.

6.4.4Text Blocks

- **String:** " " 

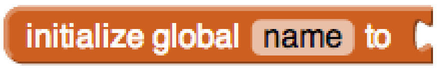
Contains a text string.

This string can contain any characters (letters, numbers, or other special characters). On App Inventor it will be considered a Text object.

- **Join:** 

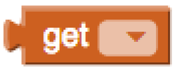
Appends all of the inputs to make a single string. If no inputs, returns an empty string.

6.4.5 Variable Blocks

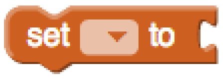
- **initialize global name to** 

This block is used to create global variables. It takes in any type of value as an argument. Clicking on *name* will change the name of this global variable. Global variables are used in all procedures or events so this block will stand alone.

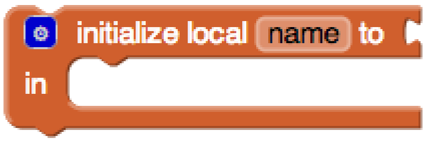
Global variables can be changed while an app is running and can be referred to and changed from any part of the app even within procedures and event handlers. One can rename this block at any time and any associated blocks referring to the old name will be updated automatically.

- **get** 

This block provides a way to get any variables that are created.

- **set to** 

This block follows the same rules as get. Only variables in scope will be available in the dropdown. Once a variable *v* is selected, the user can attach a new block and give *v* a new value.

- **initialize Local name to - in (do)** 

This block is a mutator that allows you to create new variables that are only used in the procedure you run in the DO part of the block. This way all variables in this procedure will all start with the same value each time the procedure is run. One can attach *statements* to it. Statements *do* things. That is why this block has space inside for statement blocks to be attached.

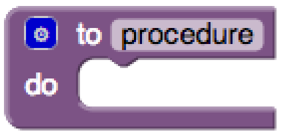
The variables in this block can be renamed at any time and any corresponding blocks elsewhere in the program that refer to the old name will be updated automatically.

AI2 Procedures

A procedure is a sequence of blocks or code that is stored under a name, the name of your procedure block. Instead of having to keep putting together the same long sequence of blocks, you can create a procedure and just call the procedure block whenever you want your sequence of blocks to run. In computer science, a procedure also might be called a function or a method.

6.4.6 Procedure Blocks

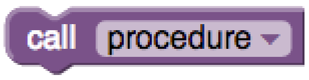
- **procedure do**



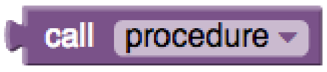
Collects a sequence of blocks together into a group. One can then use the sequence of blocks repeatedly by calling the procedure. If the procedure has arguments, they can be specified by using the block's mutator button. If the blue plus sign is clicked, one can drag additional arguments into the procedure.

When a new procedure block is created, App Inventor chooses a unique name automatically. One can click on the name and type to change it. Procedure names in an app must be unique. App Inventor will not define two procedures in the same app with the same name. A procedurecan be renamed at any time while building the app, by changing the label in the block. App Inventor will automatically rename the associated call blocks to match.

- **Call block**



When a procedure is created, App Inventor automatically generates a call block and places it in the My Definitions drawer. A call block is called to invoke the procedure.



After creating this procedure, a call block that needs to be plugged in will be created. This is because the result from executing this procedure will be returned in that call block and the value will be passed on to whatever block is connected to the plug.

6.5 Types ofComponents

Thisdescribes the components that are used in App Inventor to build an app.

Each component can have methods, events, and properties. Most properties can be changed by apps. These properties have blocks that can be used to get and set the values. Some properties can't be changed by apps. These only have blocks that can be used to get the values, not set them.

- **SpeechRecognizer**

Speech recognizer component is used to listen to the user speaking and convert the spoken sound into text using Android's speech recognition feature.

- **TextToSpeech**

The TestToSpeech component speaks a given text aloud. The pitch and the rate of speech can be set.

One can also set a language by supplying a language code. This changes the pronounciation of words, not the actual language spoken. For example, setting the language to French and speaking English text will sound like someone speaking English with a French accent.

A country can also be specified by supplying a country code. This can affect the pronounciation. For example, British English (GBR) will sound different from US English (USA). Not every country code will affect every language.

The languages and countries available depend on the particular device, and can be listed with the AvailableLanguages and AvailableCountries properties.

6.5.1 Sensor components

- **Clock**

Non-visible component that provides the instant in time using the internal clock on the phone.It can fire a timer at regularly set intervals and perform time calculations, manipulations, and conversions.

Methods to convert an instant to text are also available. Acceptable patterns are empty string, MM/DD/YYYY HH:mm:ss a, or MMM d, yyyyHH:mm. The empty string will provide the default format, which is "MMM d, yyyyHH:mm:ss a" for FormatDateTime, "MMM d, yyyy" for FormatDate.

Date and Time are formatted with Instant inTime and Duration.

- **Instant**: consists of Year, Month, Day ofMonth, Hour, Minute, and Second. An instant can be created by using MakeInstant method.
- **Duration**: time in milliseconds elapsed between instants. Duration can be obtained by Duration method.

6.5.2 Connectivity component

- **Web**

  Non-visible component that provides functions for HTTP GET, POST, PUT, and DELETE requests.

CHAPTER 7

CLOUD COMPUTING

7.1 Introduction

Cloud computing is the delivery of computing services-servers, storage, databases, networking, software, analytics and more-over the Internet (“the cloud”). Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage.

Cloud computing, often referred to as simply “the cloud,” is the delivery of on-demand computing resources-everything from applications to data centers—over the internet on a pay-for-use basis.

Cloud computing has been credited with increasing competitiveness through cost reduction, greater flexibility, elasticity and optimal resource utilization. Here are a few situations where cloud computing is used to enhance the ability to achieve business goals.

7.2 Infrastructure as a service (IaaS) and platform as a service (PaaS)

When it comes to IaaS, using an existing infrastructure on a pay-per-use scheme seems to be an obvious choice for companies saving on the cost of investing to acquire, manage and maintain an IT infrastructure.

7.3 Private cloud and hybrid cloud

Among the many incentives for using cloud, there are two situations where organizations are looking into ways to assess some of the applications they intend to deploy into their environment through the use of a cloud (specifically a public cloud). While in the case of test and development it may be limited in time, adopting a hybrid cloud approach allows for testing application workloads, therefore providing the comfort of an environment without the initial investment that might have been rendered useless should the workload testing fail.

Another use of hybrid cloud is also the ability to expand during periods of limited peak usage, which is often preferable to hosting a large infrastructure that might seldom be of use.

An organization would seek to have the additional capacity and availability of an environment when needed on a pay-as you-go basis.

7.4 Test and development

Probably the best scenario for the use of a cloud is a test and development environment. This entails securing a budget, setting up your environment through physical assets, significant manpower and time. Then comes the installation and configuration of your platform. All this can often extend the time it takes for a project to be completed and stretch your milestones.

With cloud computing, there are now readily available environments tailored for your needs at your fingertips. This often combines, but is not limited to, automated provisioning of physical and virtualized resources.

7.5 Big data analytics

One of the aspects offered by leveraging cloud computing is the ability to tap into vast quantities of both structured and unstructured data to harness the benefit of extracting business value.

Retailers and suppliers are now extracting information derived from consumers' buying patterns to target their advertising and marketing campaigns to a particular segment of the population. Social networking platforms are now providing the basis for analytics on behavioral patterns that organizations are using to derive meaningful information.

7.6 File storage

Cloud can offer you the possibility of storing your files and accessing, storing and retrieving them from any web-enabled interface. The web services interfaces are usually simple. At any time and place you have high availability, speed, scalability and security for your environment. In this scenario, organizations are only paying for the amount of storage they are actually consuming, and do so without the worries of overseeing the daily maintenance of the storage infrastructure.

There is also the possibility to store the data either on or off premises depending on the regulatory compliance requirements. Data is stored in virtualized pools of storage hosted by a third party based on the customer specification requirements.

7.7 Disaster recovery

This is yet another benefit derived from using cloud based on the cost effectiveness of a disaster recovery (DR) solution that provides for a faster recovery from a mesh of different physical locations at a much lower cost that the traditional DR site with fixed assets, rigid procedures and a much higher cost.

7.8 Backup

Backing up data has always been a complex and time-consuming operation. This included maintaining a set of tapes or drives, manually collecting them and dispatching them to a backup facility with all the inherent problems that might happen in between the originating and the backup site. This way of ensuring a backup is performed is not immune to problems such as running out of backup media , and there is also time to load the backup devices for a restore operation, which takes time and is prone to malfunctions and human errors.

Cloud-based backup, while not being the panacea, is certainly a far cry from what it used to be. You can now automatically dispatch data to any location across the wire with the assurance that neither security, availability nor capacity are issues.

While the list of the above uses of cloud computing is not exhaustive, it certainly give an incentive to use the cloud when comparing to more traditional alternatives to increase IT infrastructure flexibility , as well as leverage on big data analytics and mobile computing.

There are different ways to use cloud storage. But then, in this project cloud storage is used in two ways of which the first one is the pubnub and the second one is the dropbox.

CHAPTER 8

PUBNUB

8.1 Introduction

PubNub is a programmable network for developing realtime applications; an evolution fromthree-tier architecture, purpose-built to handle all the complexities of data streams: securingthem, ensuring a realtime experience, guaranteeing reliability, and scaling them to anynumber of devices, anywhere in the world.PubNub's core Data Stream Network is extremely fast; routing a message through the PubNub network takes as little as 4-9 milliseconds of latency, often resulting in an end-to- end experience of 30-40ms for users on a broadband connection.

PubNub can be used to quickly push small messages to one or more devices (smartphones,browsers, microcontrollers, tablets, etc.) – essentially, just about any device that can make aTCP/IP connection to the internet – as well as back again, for bidirectional communicationbetween devices. These messages can be used for human communication (like online chat),machine-to- machine control, IoT, geolocation, smart homes, financial data, multiplayergames, and a lot more.

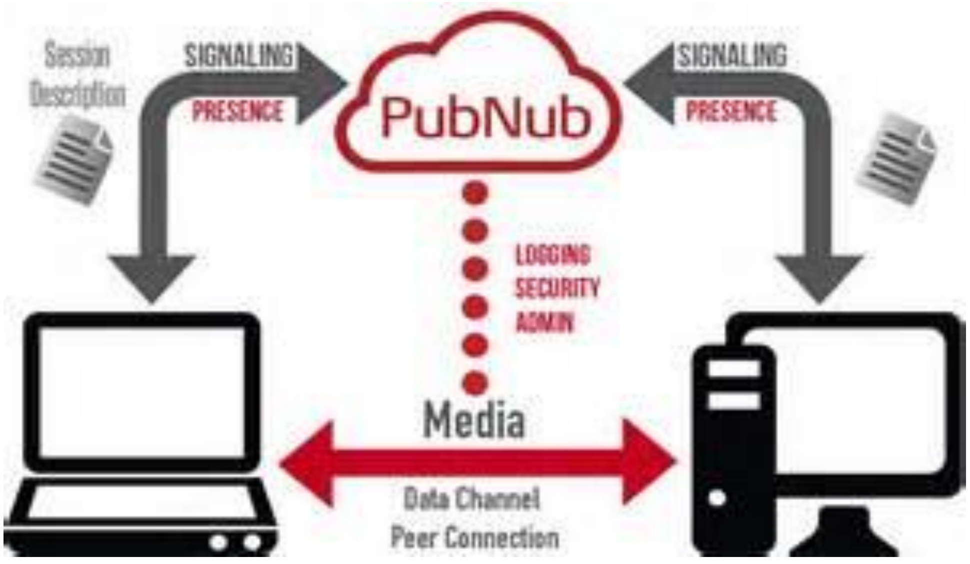


Fig 8.1PubNub

PubNub utilizes a Publish/Subscribe model for real time data streaming and device signaling which lets you establish and maintain persistent socket connections to any device and push data to global audiences in less than ¼ of a second.You can publish messages to any given channel, and subscribing clients receive onlymessages associated with that channel. The message payload can be any JSON data includingnumbers, strings, arrays, and objects. PubNub’s globally distributed network infrastructure powers two core services from which all other features build upon:

- Publish/Subscribe: this is the messaging pattern for how to stream data in realtime acrossthe PubNub network.
- BLOCKS: this enables you to run code directly inside the PubNub network, whichexecutes on your data as it is being streamed between senders and receivers

CHAPTER 9

DROP BOX

9.1 Introduction

Dropbox is a program that we are all familiar with in one way or another. The birth story of this company, however, is not as common knowledge.
Drew Huston, founder and CEO of Dropbox, has been given the title of Internet entrepreneur for his achievements in his field. The creation story of the company, Huston claims, was based upon an idea that came to him after having had to deal with constantly forgetting his USB flash drive during his studies at MIT. While he was a student he found an abundance of problems with already existing storage services and set out to solve this problem for himself before realizing that his solution could benefit others as well. In 2007, he and his co-founder Arash Ferdowsi were able to secure funding to begin the development of the program, and by 2008 they were ready to launch. Dropbox had an enormous success rate, and, after officially being introduced at TechCrunch50, broke records gaining 50 million users in just under three years. By the end of 2013 Dropbox had gained over 200 million users.

Huston’s success did not go unnoticed, however, especially by some of the executives in extremely high positions at the time including none other than Steve Jobs. In 2009, Huston was personally invited to a meeting in the Apple office in California to discuss this new startup, Dropbox, with the CEO himself. Job’s did his best to persuade Huston to partner Dropbox as a new Apple program, but Huston was determined to build this online storage system into a big company. He adamantly refused Job’s offer, disregarding the huge sum being offered for program. Not long after, Apple released its latest addition, the iCloud in a successful attempt at connecting all devices for better file sharing

With this attack from such a major player Huston was shaken with the fear that Dropbox would fall alongside other names such as MySpace, Netscape, or Palm. This is the fear that drove him and quite possibly the deciding factor between his success instead of his failure. The company grew, reaching an even larger user basis while still remaining reliable and keeping its staff relatively small (fewer than 200 employees for its millions of customers). Huston is to this day still incredibly invested in his company personally, his share making up

15% of the company as $600 million on paper, and believes whole heartedly in its growth and success.

We can use Huston's story as an aid in our own entrepreneurial journey. His experiences can help us as we make our own decisions as our own ideas begin to become our realities.

Dropbox lets us take your photos, docs, and videos anywhere and share them easily. Access any file you save to your Dropbox from all of your computers, phones, tablets, and on the web. With Dropbox you'll always have your important memories and work with you. And if something happens to your Windows mobile, tablet or Xbox, your stuff is always safe in Dropbox.

9.2 Features

- It helps to access our photos, docs, and videos from any device.
- It provides 2 GB of free space when we sign up.
- We can even share our biggest files with a simple link without any more attachments.
- We can add files to our "Favorites" for fast and offline viewing.

Fig 9.1 Drop Box

CHAPTER 10

BLOCK DIAGRAM

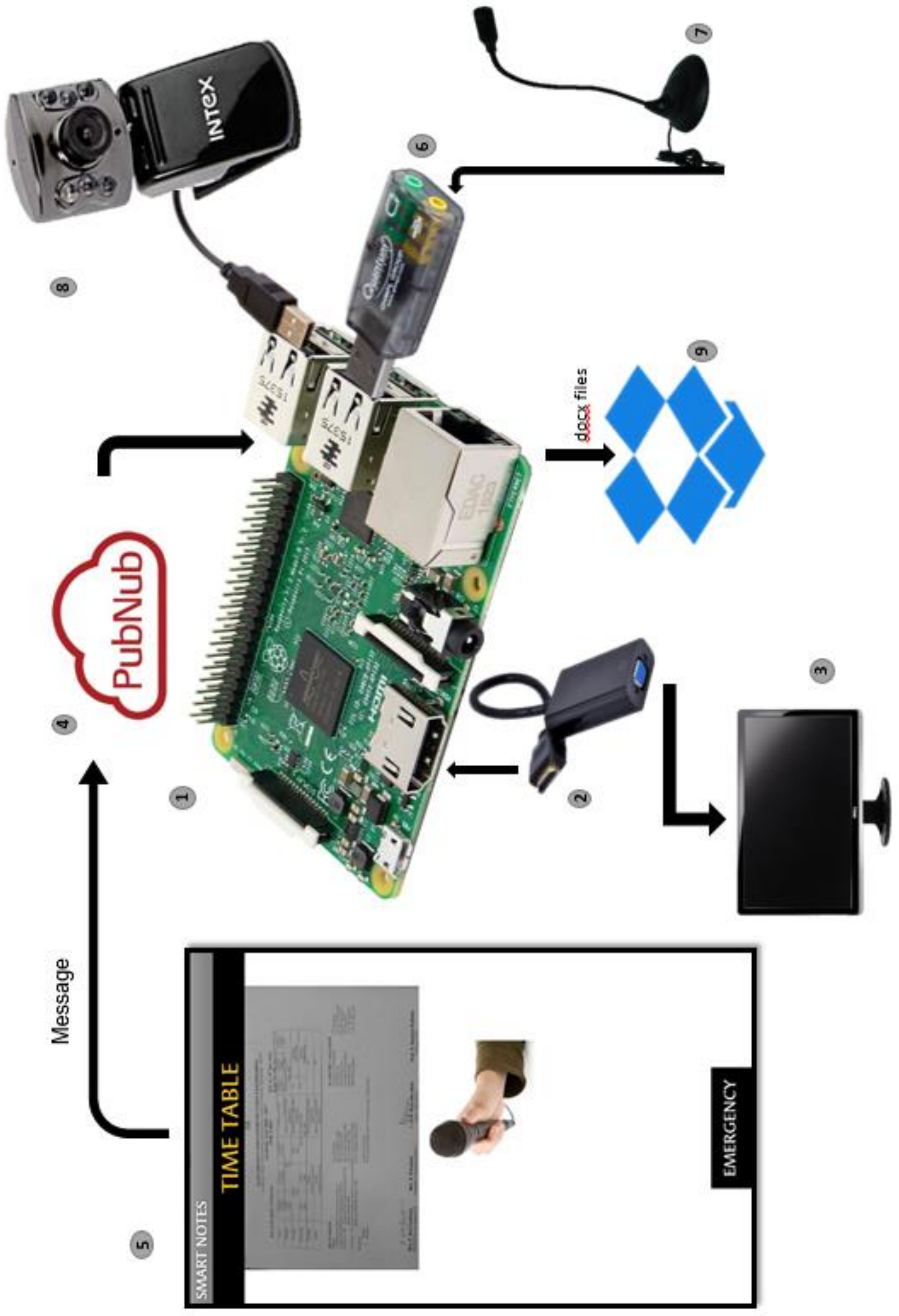


Fig10.1 Block Diagram

10.1 Raspberry Pi

- Raspberry pi is system on chip, so it requires an operating system to work.
- Raspbian has been the standard Raspberry Pi operating system, more commonly referred to as a “distro”, since the Pi arrived in 2012 and we have seen it grow over time into the capable distro that we use today. Raspbian is a fork of the massively popular Debian distribution and it is jointly maintained by the Raspberry Pi Foundation and the community.
- The Raspberry Pi Foundation recommends the use of Raspbian, a Debian-based Linux operating system. Other third party operating systems available via the official website include Ubuntu MATE, Snappy Ubuntu Core, Windows 10 IoT Core, RISC OS and specialised distributions for the Kodi media center and classroom management.
- Many other operating systems can also run on the Raspberry Pi
- Raspbian is a community project under active development, with an emphasis on improving the stability and performance of as many Debian packages as possible.
- Being based on Debian, Raspbian comes with the APT (Advanced Packaging Tool) as it’s package manager, which is used to install software from the vast Raspbian repositories, but Raspbian also comes with raspi-config, a menu based tool that simplifies the act of managing Raspberry Pi configurations such as setting up an SSH, overclocking and enabling the official Raspberry Pi camera.

10.1.1SDCard Classes

There are also different families or generations of SD cards: the first generation SD, the second generation SDHC, and the third general SDXC. In general, each new generation brought increased speeds and capacities, allowing faster and larger cards to be made.

The different families also use different filesystems – SDHC cards came with FAT32, and SDXC cards use the exFAT filesystem. Both FAT32 and exFAT are supported by recent versions of Windows and Mac, although Linux distributions often support only FAT32 for legal reasons.

Each SD or Micro SD card has a speed rating, called a class. Larger class numbers correspond to a faster level of minimum performance, allowing files to be copied or recorded at a higher speed. The class rating system is actually quite simple to remember: the number after each Class corresponds to the minimum sequential read speeds, in MB/s. So Class 2 is 2MB/s, Class 6 is 6MB/s, etc. For example, Class 2 is sufficient for SD video recording, while 4 and 6 support HD recording.

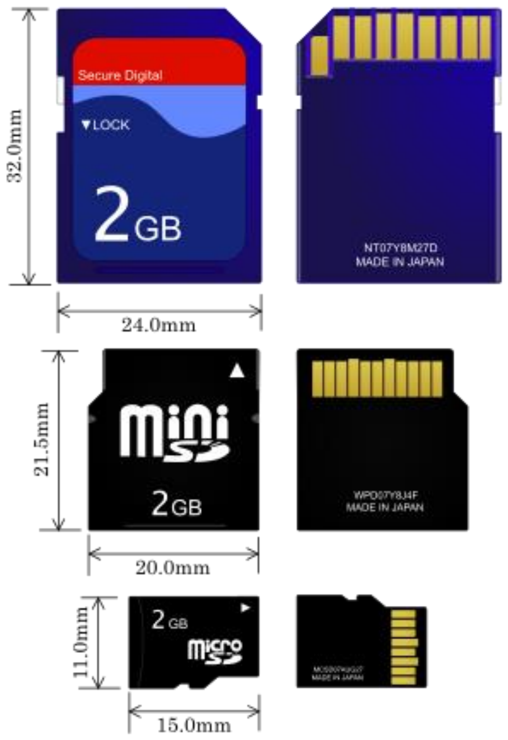


Fig 10.2 SD Cards

10.1.2 Formatting SD Card using SD formatter

Installation of SD Formatter

- Close all application programs on your PC.
- Execute the SD Formatter installation program which you downloaded from SDA website.
- Follow the instructions from the SD Formatter installation program and complete the installation.

10.1.3 Formatting using card reader

Connect the SD card to your computer**.** If your computer has an SD card slot you can insert the SD card directly into the port. If you are using a MicroSD card, you will need an adapter to allow it to fit into standard SD card ports.

- If your computer does not have an SD card port built-in, you can use an external USB card reader that plugs into any USB port on your computer.
- You can link a portable device such as your cell phone or digital camera to your computer (also usually done via USB). You can then insert the SD card into the portable device and it will temporarily act as an SD card reader.
- Open the Computer window.
- Find your SD card which is a removable drive that is connected to your computer.
- Open the Format tool. Right-click on the SD card and select "Format" from the menu that pops up. This will take you to the Format window. Keep "Capacity" and "Allocation unit size" set to default.
- After you confirm, your SD card will be formatted and you will end up with a blank memory card that can be inserted in any device that accepts SD cards.

10.1.4 Installing Operating System (OS)

There are two ways of installing OS:

- ➢ Raw Installation Jessie is used to install operating system images using windows.
- ➢ New Out Of Box Software (NOOBS) is an easy operating system installation manager for the Raspberry Pi

10.1.4. a Raw Installation Jessie

- Insert the SD card into your SD card reader. You can use the SD card slot if you have one, or an SD adapter in a USB port. Note the drive letter assigned to the SD card. You can see the drive letter in the left hand column of Windows Explorer, for example G:
- Download the Win32DiskImager utility from the Sourceforge Project page as an installer file, and run it to install the software.
- Run the Win32DiskImager utility from your desktop or menu.
- Select the image file you extracted earlier.

- In the device box, select the drive letter of the SD card. Be careful to select the correct drive: if you choose the wrong drive you could destroy the data on your computer's hard disk! If you are using an SD card slot in your computer, and can't see the drive in the Win32DiskImager window, try using an external SD adapter.
- Click 'Write' and wait for the write to complete.
- Exit the imager and eject the SD card.

10.1.4.bNOOBS

Installing NOOBS on an SD Card

It is recommend using an SD card with a minimum capacity of 8GB.

- Using a computer with an SD card reader, visit the Downloads page.
- Click on NOOBS, then click on the Download ZIP button under ‘NOOBS (offline and network install)’, and select a folder to save it to.
- Extract the files from the zip.

i. FORMAT SD CARD:

- It is best to format your SD card before copying the NOOBS files onto it. To do this:Visit the SD Association’s website and download SD Formatter 4.0 for either Windows or Mac.
- Follow the instructions to install the software.
- Insert your SD card into the computer or laptop’s SD card reader and make a note of the drive letter allocated to it, e.g. G:/
  In SD Formatter, select the drive letter for your SD card and format it.

ii.DRAG AND DROP NOOBS FILES:

- Once your SD card has been formatted, drag all the files in the extracted NOOBS folder and drop them onto the SD card drive.
- The necessary files will then be transferred to your SD card.
- When this process has finished, safely remove the SD card and insert it into your Raspberry Pi.

iii. FIRST BOOT:

- Now plug the USB power cable into your Pi.

- Your Raspberry Pi will boot, and a window will appear with a list of different operating systems that you can install. We recommend that you use Raspbian – tick the box next to Raspbian and click on Install.
- Raspbian will then run through its installation process. Note that this can take a while.
- When the install process has completed, the Raspberry Pi configuration menu (raspi-config) will load. Here you are able to set the time and date for your region, enable a Raspberry Pi camera board, or even create users. You can exit this menu by using Tab on your keyboard to move to Finish.

iv. LOGGING IN AND ACCESSING THE GRAPHICAL USER INTERFACE:

- The default login for Raspbian is username pi with the password raspberry. Note that you will not see any writing appear when you type the password. This is a security feature in Linux.To load the graphical user interface, type startx and press Enter.

10.1.5Raspi-Config

raspi-config is the Raspberry Pi configuration tool written and maintained by Alex Bradbury. It targets Raspbian.The raspi-config tool helps to change several settings without having to know the correct commands to use. It is written as a bash script, run in a terminal window, and uses whiptail (whiptail is a "dialog" replacement) to create the windows, menus and messages.

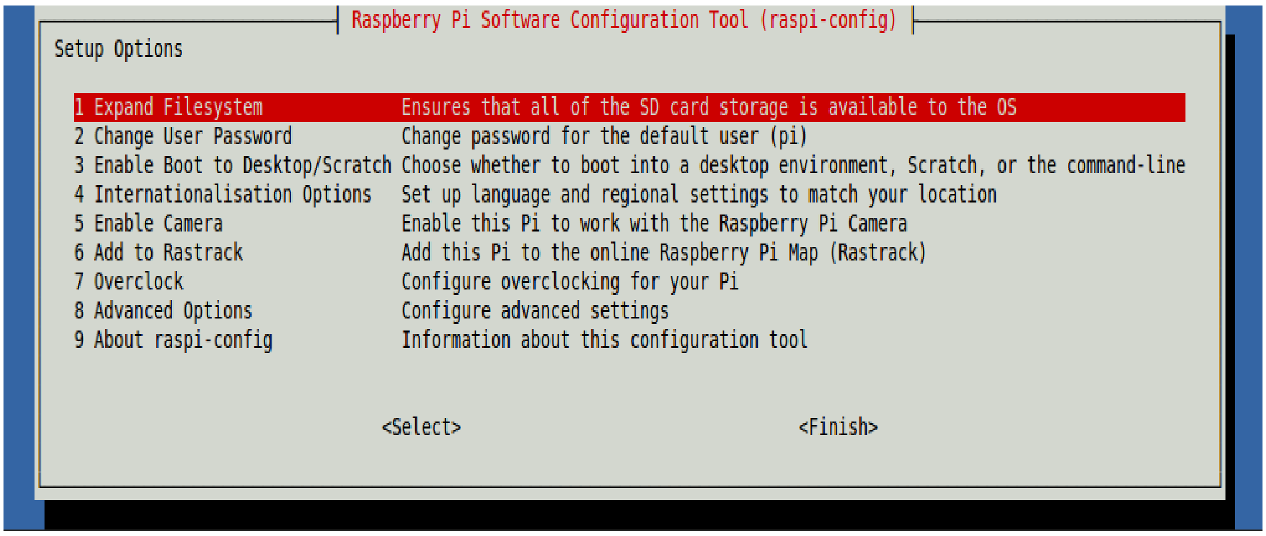
10.1.5.aUsage

You will be shown raspi-config on first booting into Raspbian. To open the configuration tool after this, simply run the following from the command line:

```
sudoraspi-config
```

The sudo is required because you will be changing files that you do not own as the pi user.

A blue screen with options in a grey box in the centre is seen, like so:



```
Raspberry Pi Software Configuration Tool (raspi-config)
Setup Options

1 Expand Filesystem              Ensures that all of the SD card storage is available to the OS
2 Change User Password           Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
4 Internationalisation Options   Set up language and regional settings to match your location
5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camera
6 Add to Rastrack                Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock                      Configure overclocking for your Pi
8 Advanced Options               Configure advanced settings
9 About raspi-config             Information about this configuration tool

                  <Select>                    <Finish>
```

Fig 10.3 Raspberry Pi Software Configuration Tool

The Advanced Options is another menu

```
     Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan          You may need to configure overscan if black bars are
A2 Hostname          Set the visible name for this Pi on a network
A3 Memory Split      Change the amount of memory made available to the
                     GPU
A4 SSH               Enable/Disable remote command line access to your Pi
A5 Device Tree       Enable/Disable the use of Device Tree
A6 SPI               Enable/Disable automatic loading of SPI kernel
module
A7 I2C               Enable/Disable automatic loading of I2C kernel
module
A8 Serial            Enable/Disable shell and kernel message on the seria
A9 Audio             Force audio out through HDMI or 3.5mm jack
A0 Update            Update this tool to the latest version

<Select><Back>
```

10.1.5.bMoving Around the Menu:

- Cursor up/down keys move the highlight up and down menus, stopping at the top or bottom.

- The Tab key switches from the selected menu entry and the "buttons" at the bottom (inside angle brackets), and back again.
- With a menu entry or a "button" highlighted, press Enter to start that option or button.
- Pressing ESC quits from the menus, without confirmation. Whiptail has a 'hotkey' capability which does not apply in raspi-config.

10.1.5.cWhat raspi-config does?

raspi-config aims to provide the functionality to make the most common configuration changes. This tool provides a straightforward way of doing initial configuration of the Raspberry Pi.This may result in automated edits to/boot/config.txt and various standard Linux configuration files. Some options require a reboot to take effect. If you changed any of those, raspi-config will ask if you wish to reboot now when you select the <Finish> button.

10.1.5.dMenu Options

- Expand file system:

If you have installed Raspbian using NOOBS, the filesystem will have been expanded automatically. There may be a rare occasion where this is not the case, e.g. if you have copied a smaller SD card onto a larger one. In this case, you should use this option to expand your installation to fill the whole SD card, giving you more space to use for files. You will need to reboot the Raspberry Pi to make this available. Note that there is no confirmation: selecting the option begins the partition expansion immediately.

- Change user password:

The default user on Raspbian is pi with the password raspberry. You can change that here. Read about other users.

- Enable boot to desktop or scratch:

You can change what happens when your Pi boots. Use this option to change your boot preference to command line, desktop, or straight to Scratch.

- Internationalization Options:

Select Internationalisation Options and press Enter to be taken to a sub-menu containing the following options:

- Change locale:

Select a locale, for example en_GB.UTF-8 UTF-8.

- Change Time zone:

Select your local timezone, starting with the region such as Europe, then selecting a city, for example London. Type a letter to skip down the list to that point in the alphabet.

- Change keyboard layout:

This option opens another menu which allows you to select your keyboard layout. It will take a long time to display while it reads all the keyboard types. Changes usually take effect immediately, but may require a reboot.

- Enable camera:

In order to use the Raspberry Pi Camera Module, you must enable it here. Select the option and proceed to Enable. This will make sure at least 128MB of RAM is dedicated to the GPU.

- Overclock:

It is possible to overclock your Raspberry Pi's CPU. The default is 700MHz but it can be set up to 1000MHz. The overclocking you can achieve will vary; overclocking too high may result in instability. Selecting this option shows the following warning:

10.1.5.eAdvanced Options

Overscan:

Old TV sets had a significant variation in the size of the picture they produced; some had cabinets that overlapped the screen. TV pictures were therefore given a black border so that none of the picture was lost; this is called overscan. Modern TVs and monitors don't need the border, and the signal doesn't allow for it. If the initial text shown on the screen disappears off the edge, you need to enable overscan to bring the border back.

Any changes will take effect after a reboot. You can have greater control over the settings by editing config.txt.On some displays, particularly monitors, disabling overscan will make the picture fill the whole screen and correct the resolution. For other displays, it may be necessary to leave overscan enabled and adjust its values.

Hostname:

Set the visible name for this Pi on a network.

Memory Split:
Change the amount of memory made available to the GPU.

SSH:
Enable/disable remote command line access to your Pi using SSH.SSH allows you to remotely access the command line of the Raspberry Pi from another computer. SSH is disabled by default. Read more about using SSH on the SSH documentation page. If connecting your Pi directly to a public network, you should not enable SSH unless you have set up secure passwords for all users.

Device tree:
Enable/Disable the use of Device Tree. Read more about Device Trees config on the Device Trees documentation page.

SPI:
Enable/Disable SPI interfaces and automatic loading of the SPI kernel module, needed for products such as PiFace.

I2C
Enable/Disable I2C interfaces and automatic loading of the I2C kernel module.

Serial:
Enable/Disable shell and kernel messages on the serial connection.

Audio:
Force audio out through HDMI or a 3.5mm jack. Read more on the audio configuration documentation page.

Update:
Update this tool to the latest version.
Finish:

Use this button when you have completed your changes. You will be asked whether you want to reboot or not. When used for the first time, it's best to reboot. There will be a delay in rebooting if you have chosen to resize your SD card.

10.2HDMI to VGA Convertor

HDMI to VGA converter is a low cost solution to connect VGA monitor with PI.This needs no external power supply & no settings to do.It supports VGA output upto 1080I (60Hz).As VGA doesn’t support audio,this converter has no audio output.

Remove the SD card loaded with Raspbian Operating System & plug it into your Laptop.For Desktop you need to use a Card Reader.Open the folder to locate CONFIG.TXT file.You’ll also need to make a small change to the **config.txt** file that Raspberry Pi uses when it boots.Using a memory card reader on your desktop computer, insert the Raspberry Pi SD card and open config.txt in your preferred text editor.

Look for the following lines:

```
#hdmi_force_hotplug=1
#hdmi_drive=2
```

Both options need to be enabled, which you can do by removing the hash symbol and saving. These options enable VGA output through an HDMI adaptor and sets the screen resolution to a low 640 x 480.

If you want a higher resolution, remove the hash symbols from the following lines:

```
#hdmi_group=1
#hdmi_mode= 4
```

You will also need to edit these two lines, changing **hdmi_group** to **2** and **hdmi_mode** to **16**. Remember to save your changes before safely removing and replacing in your Raspberry Pi.

10.3VGA MONITOR

The Raspberry Pi has an HDMI port which you can connect directly to a monitor or TV with an HDMI cable. This is the easiest solution; some modern monitors and TVs have HDMI ports, some do not, but there are other options.For monitors with VGA only, you can use an HDMI-to-VGA adapter. VGA does not support audio.

10.4 PUBNUB

10.4.1 Installing PubNub

The simplest way to get started is to install PubNub Python SDK via pypi:

```
pip install 'pubnub>=3,<4’
```

10.4.2 Hello World Program using PubNub

```
frompubnub importPubnub
```

```
pubnub = Pubnub(publish_key="demo", subscribe_key="demo")
defcallback(message, channel):
    print(message)


def error(message):
    print("ERROR : " + str(message))


def connect(message):
    print("CONNECTED")
    printpubnub.publish(channel='my_channel', message='Hello 
World')

   def reconnect(message):
    print("RECONNECTED")


def disconnect(message):
    print("DISCONNECTED")

  pubnub.subscribe(channels='my_channel', callback=callback, 
error=callback, connect=connect, reconnect=reconnect, 
disconnect=disconnect)
```

When message ‘Hello World’ is published, it is sent through the channel named ‘my_channel’ and it will be subscribed.

When error signal occurs, it delivers an error message and If callback signal occurs, it will print the message again. Based on the signal received, it will either connect or disconnect to the channel.

10.4.3 Code

```
from pubnub import Pubnub
pubnub = Pubnub(publish_key="pub-c-800c1668-57d0-45ec-9700-
74d251ed134b", subscribe_key="sub-c-5936b224-0eca-11e7-92d3-
02ee2ddab7fe")
def callback(message, channel):
    print(message)
def error(message):
    print("ERROR : " + str(message))
pubnub.subscribe(channels='smart_notes', callback=callback,
error=error)
```

This code is implemented in a new python file created using ***sudonano subscribe.py***

This code is executed using ***python subscribe.py***

Import PubNub publish key and subscribe key. The message is sent through the channel named ‘smart_notes’. If error signal occurs, an ERROR message is delivered and if call back signal occurs, the message is printed again.

10.5MIT APP

There are three options for building apps on MIT App Inventor

Option 1: If you are using an Android device and you have a wireless internet connection, you can start building apps without downloading any software to your computer. You will need to install the App Inventor Companion App on your device.

Option 2: If you do not have an Android device, you'll need to install software on your computer so that you can use the on-screen Android emulator.

Option 3: If you do not have a wireless internet connection, you'll need to install software on your computer so that you can connect to your Android device over USB. The USB Connection option can be tricky, especially on Windows.

Since we have an android device and a working internet connection, we have opted for option1.

We can develop apps on website: ai2.appinventor.mit.edu. To do live testing on Android device just install the MIT App Inventor Companion app on any Android phone or tablet. Once the Companion is installed, one can open projects in App Inventor on the web and the companion on the device, and can test apps as they are built.

The steps to be followed are given below:

Step 1: Download and install the MIT AI2 Companion App on your phone.

Open your device's QR code scanner and scan the QR code on the left below to download the Companion App from the Play Store. If you can't use the Play Store, use the QR code on the right to download the Companion App directly to your phone.

Play Store	APK File
Recommended: Automatic updates	Manual updates required
Scan this QR code	Scan this QR code
to get the app from the Play Store directly	to download the app

After downloading, step though the the instructions to install the Companion app on your device.You need to install the MIT AI2 Companion only once, and then leave it on your phone or tablet for whenever you use App Inventor.

Step 2: Connect both your computer and your device to the SAME WiFi Network

App Inventor will automatically show you the app you are building, but only if your computer (running App Inventor) and your Android device (running the Companion) are connected to the same WiFi Network. See a more detailed explanation of this here.

Step 3: Open an App Inventor project and connect it to your device.

Go to App Inventor and open a project (or create a new one -- use Project > Start New Project and give your project a name).

Then choose "Connect" and "AI Companion" from the top menu in the AI2 browser

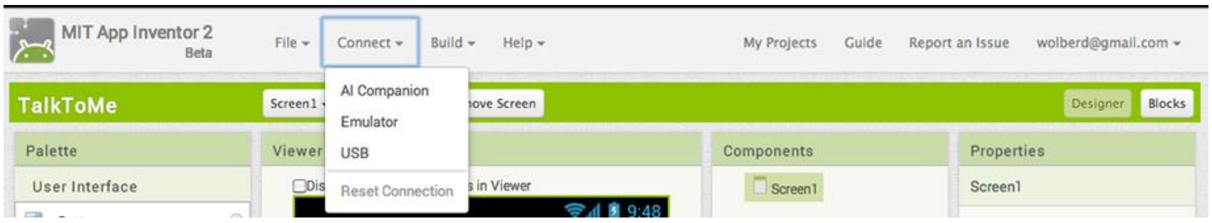
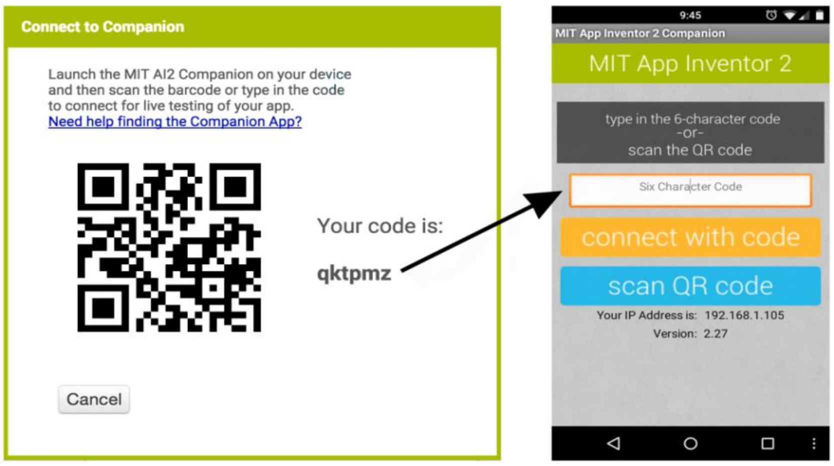


Fig 10.4 AI2 Browser

A dialog with a QR code will appear on your PC screen. On your device, launch the MIT App Companion app just as you would do any app. Then click the "Scan QR code" button on the Companion, and scan the code in the App Inventor window:


Connect to Companion

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
Need help finding the Companion App?

Your code is:
qktpmz

Cancel

Fig 10.5 App Inventor Window

Within a few seconds, you should see the app you are building on your device. It will update as you make changes to your design and blocks, a feature called “live testing”.

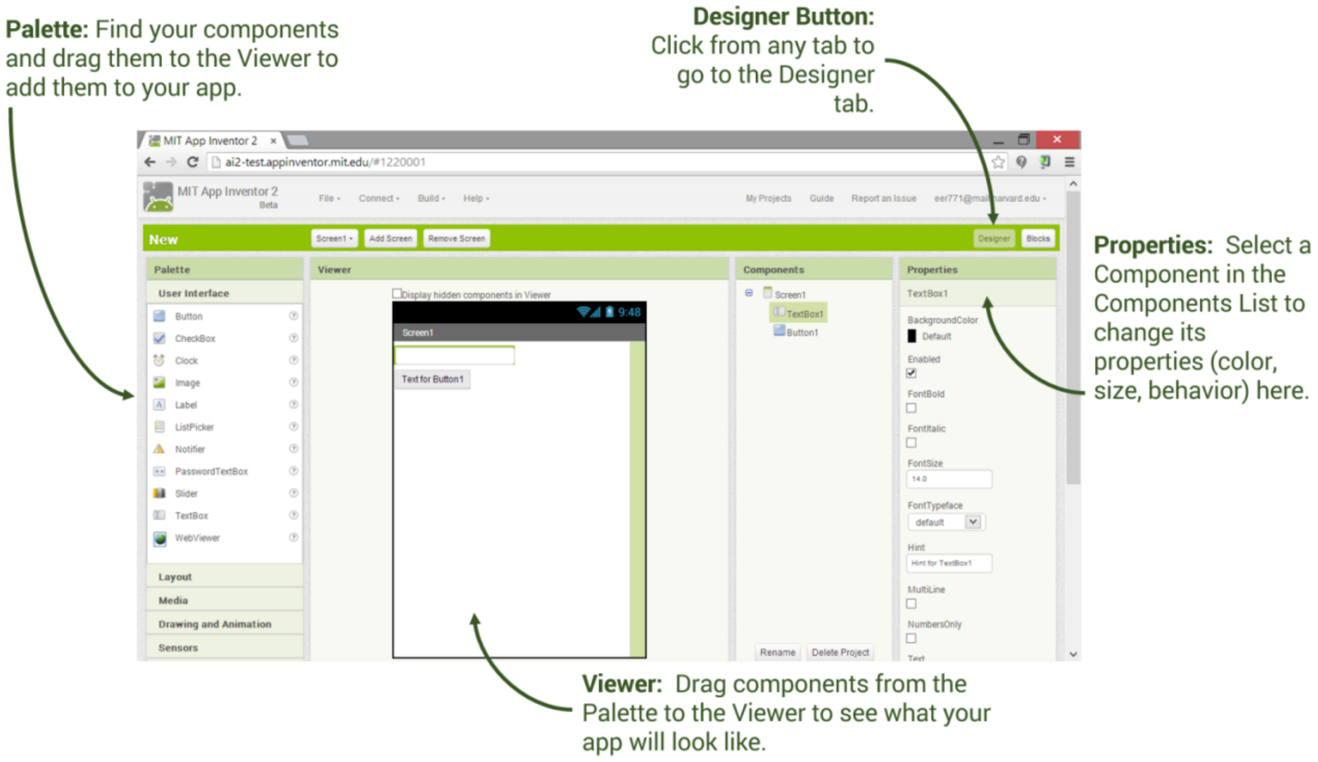
If you have trouble scanning the QR code or your device does not have a scanner, type the code shown on the computer into the Companion's text area on your Android device exactly as shown. The code is directly below where the screen on your PC shows "Your code is" and consists of six characters. Type the six characters and choose the orange "Connect with code". Do not type an Enter or carriage return: type just the six characters followed by pressing the orange button..

10.5.1 Designer and Blocks Editor

App Inventor consists of the Designer and the Blocks Editor. These are described in detail below.

10.5.1.aApp Inventor Designer

Design the App's User Interface by arranging both on- and off-screen components.



Palette: Find your components and drag them to the Viewer to add them to your app.

Designer Button: Click from any tab to go to the Designer tab.

Properties: Select a Component in the Components List to change its properties (color, size, behavior) here.

Viewer: Drag components from the Palette to the Viewer to see what your app will look like.

Fig 10.6 App Inventor Designer

10.5.1.bApp Inventor Blocks Editor

Program the app's behavior by putting blocks together.

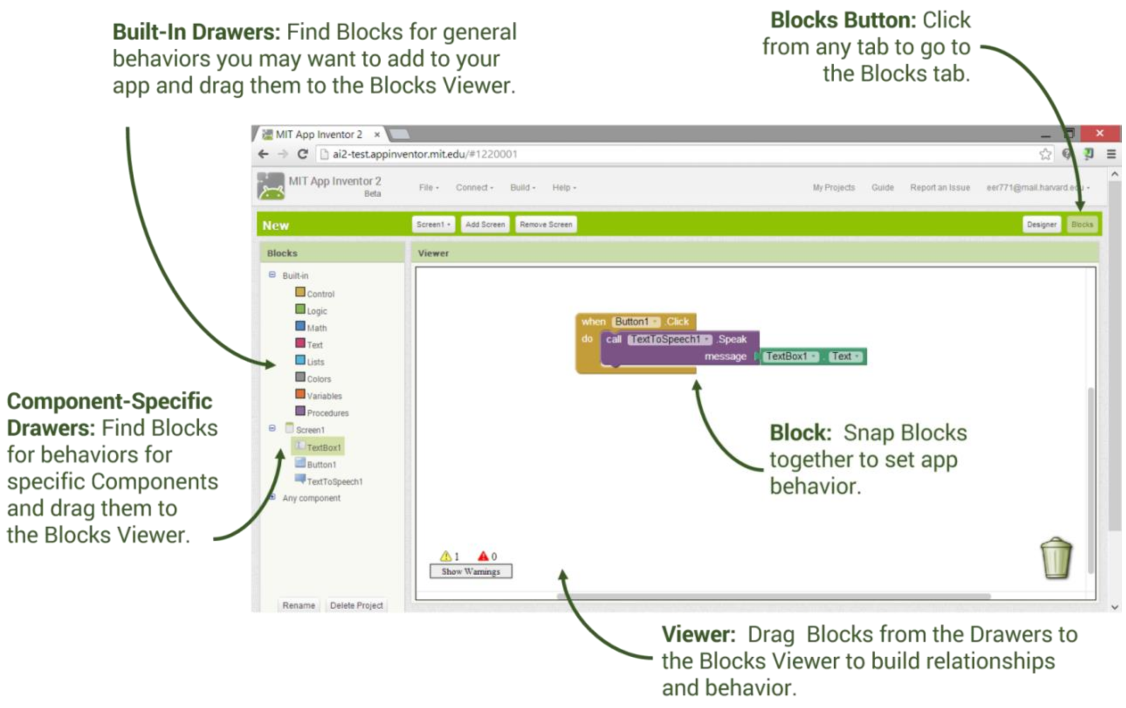

Built-In Drawers: Find Blocks for general behaviors you may want to add to your app and drag them to the Blocks Viewer.
Blocks Button: Click from any tab to go to the Blocks tab.
Component-Specific Drawers: Find Blocks for behaviors for specific Components and drag them to the Blocks Viewer.
Block: Snap Blocks together to set app behavior.
Viewer: Drag Blocks from the Drawers to the Blocks Viewer to build relationships and behavior.
MIT App Inventor 2
ai2-test.appinventor.mit.edu/#1220001
File
Connect
Build
Help
My Projects
Guide
Report an Issue
New
Screen1
Add Screen
Remove Screen
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Colors
Variables
Procedures
Screen1
TextBox1
Button1
TextToSpeech1
Any component
when Button1 Click
do call TextToSpeech1 Speak
message TextBox1 Text
Show Warnings
Rename
Delete Project

Fig 10.7 App Inventor Blocks Editor

10.5.1.cSharing and Packaging Apps

You can share your app in an executable form (.apk) that can be installed on a device, or in source code form (.aia) that can be loaded into App Inventor and remixed. You can also distribute your app on the Google Play Store

Sharing your app so that others can remix (.aia file)

Make sure you are viewing the list of all of your projects (if you are not, choose Projects | My Projects). Select the project you wish to share by checking the box next to it. Choose Project | Export selected project (.aia) to my computer to export the source code (blocks) for your project. The source code is downloaded in a .aia file.

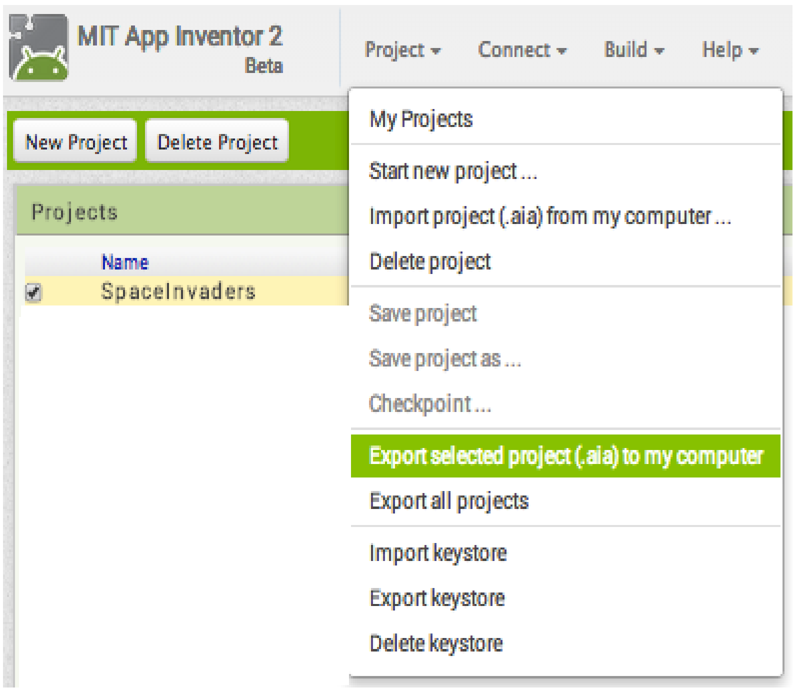


Fig 10.8 Exporting Project

If you send it to a friend, they can open it with Project | Import project (.aia) from my computer.

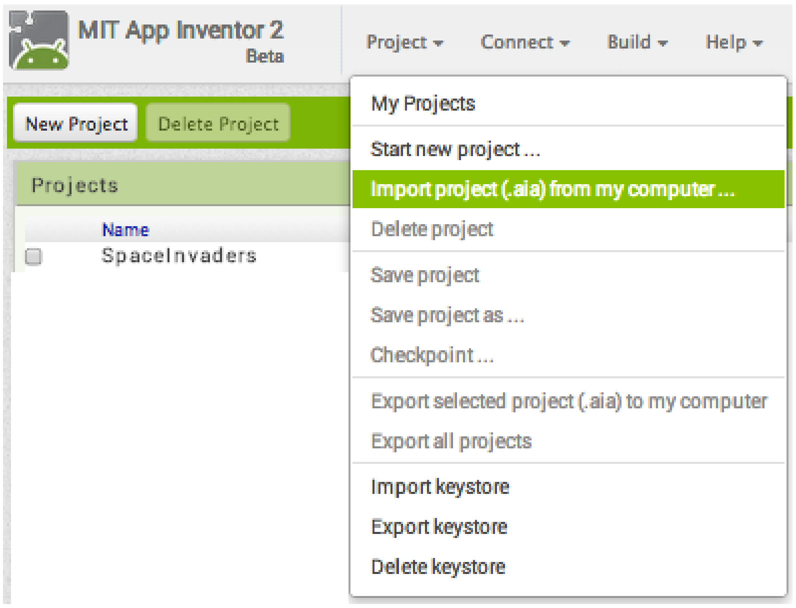


Fig 10.9 Importing Project

Sharing your app for others to install on their phone/tablet (.apk file)

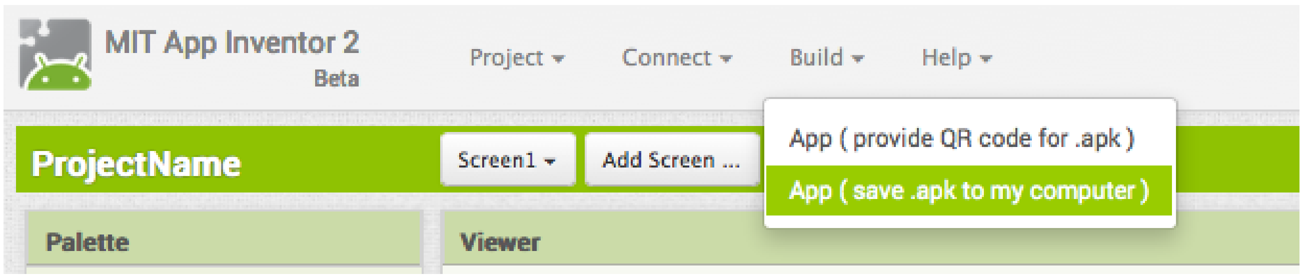


Fig 10.10 Sharing App

Package the app (.apk file) by going to the "Build" menu on the App Inventor toolbar.

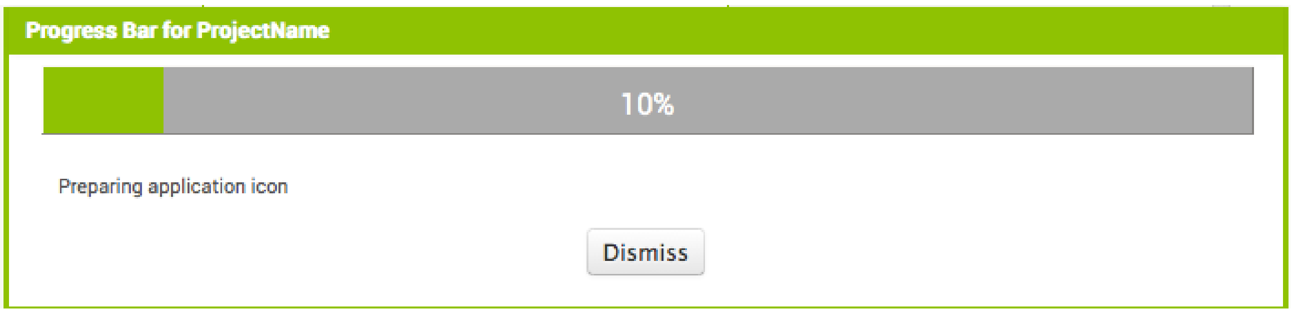


Fig 10.11 Packing App

Select "App (save .apk to my computer)." A pop-up box should alert you that your download has begun. Note: The other option (provide QR code for .apk) produces a scannable QR code that will download the app for two hours. You can share this code with others, but they have to use it within 2 hours of your generating it.

Once the build completes, you can email the app (".apk" file) to your friends who can install it by opening the email from their phone. If you want to distribute it more widely, you can upload it to a website that both you and your friend can access. You can also distribute your app on the Google Play Store.

We have created an app called “SMART NOTES”. This is used to send messages to raspberry pi when commands are given.

Two Screens are created for easy user-interface

Screen 1

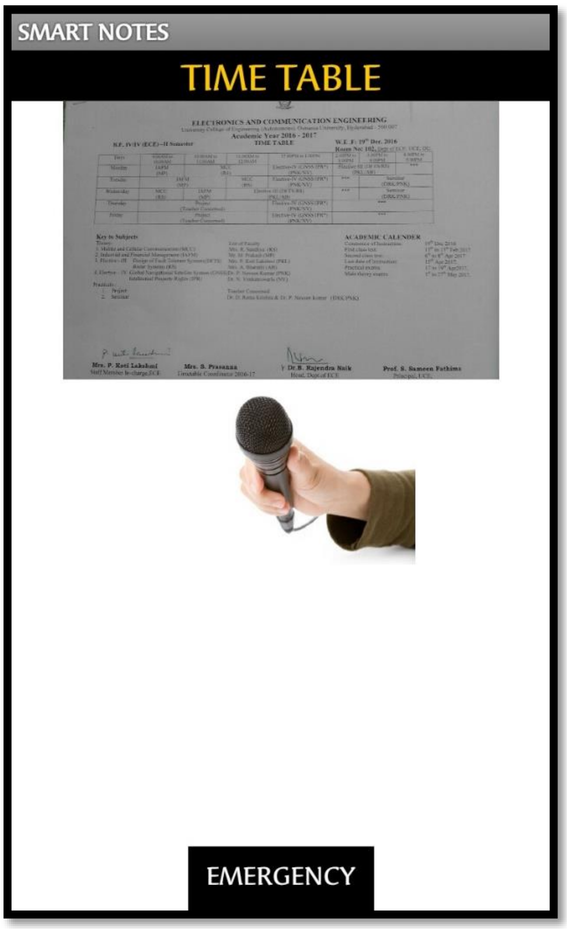


Fig 10.12 Screenshot of Screen 1

A button is created by just dragging and dropping the “Button” component from the pallette to the viewer window. When this button is clicked ‘google speak now’ pops up and waits for the speech. The speech is then coverted to text which is then compared with the predefined set of commands.

- If the text matches with “Start Recording”, The app will then ask for the name of the subject. The messsage will then be sent to pi in a specified format. Pi compares the name of subject with the list of subjects provided. If any of the subjects matches with the text, a new terminal opens and the microphone starts recording the lecture.

- If the text matches with “Take Picture”, A message will be sent to pi and then the camera takes a picture of the board

- If the text matches with “Stop Recording”, A message will be sent to pi and then the mic stops recording until ‘resume’ command is received.

- If the text matches with “Resume”, A message will be sent to pi and then the mic starts recording again.

- If the text matches with “Upload”, A message will be sent to pi which will then upload a docx file consisting of coverted speech along with the pictures taken into our registered account in the dropbox.

Screen2:

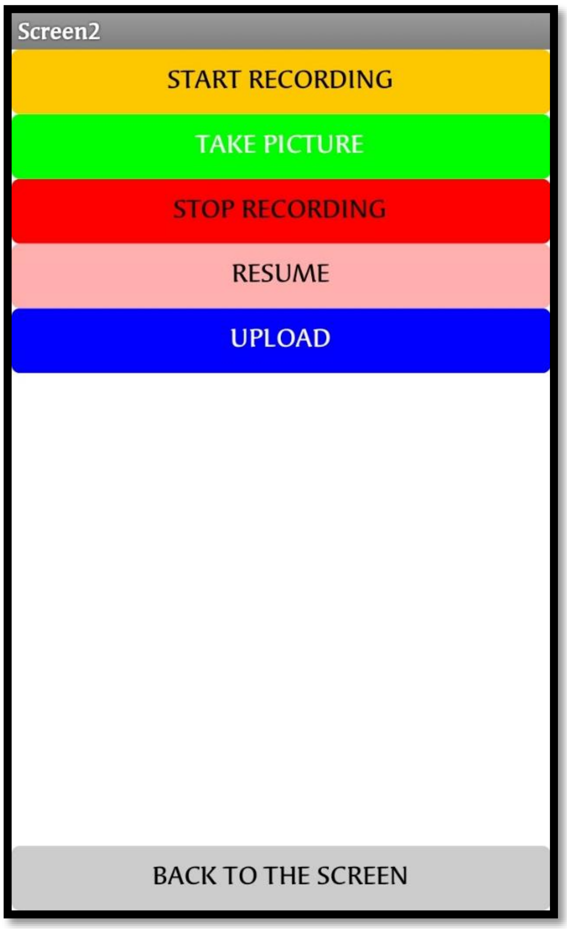


Fig 10.13 Screenshot of Screen 2

The following five buttons are created

1. **Start Recording :**

   When this button is clicked, The app displays a list of subjects. The user is requested to select a subject from the list.Then a message in the form of **“start+subject+date+time”** is sent to raspberry pi which instructs the mic to start recording.

2. **Take Picture**:

   When this button is clicked, Then a message in the form of **“picture+subject+date+time”** is sent to raspberry pi which instructs the webcam to take a picture.

3. **Stop Recording:**

   When this button is clicked, a message in the form of **“stop+subject+date+time”** is sent to raspberry pi which instructs the mic to pause recording until the resume button is pressed.

4. **Resume:**

   When this button is clicked, a message in the form of **“resume+subject+date+time”** is sent to raspberry pi which instructs the mic to resume recording.

5. **Upload:**

   When this button is clicked a message in the form of **“upload+subject+date+time”** is sent to raspberry pi. A docx file is uploaded to the dropbox.

   An additional button named “ Back to the screen” is used to switch from screen2 to screen1.

10.6 USB SOUND CARD

A sound card needs to convert the bits from a computer. For going from the computer tospeakers, a Digital-to- Analog Converter (DAC) receives the train of bits from the computerand transform it into a wave form. The DAC will take samples from the bit train and createthe shape of a wave from it. This wave becomes the electric current to be sent to speakers.

10.7. MICROPHONE

Microphone listens to speech . Here, Pi takes the audio file from the microphone and converts the audio to a text file.

10.7.1Speech to Text Conversion

Google has a great Speech Recognition API. This API converts spoken text (microphone) into written text (Python strings), briefly Speech to Text. You can simply speak in a microphone and Google API will translate this into written text. The API has excellent results for English language.

Speech to text conversion is the process of converting spoken words into written texts. This process is also often called speech recognition. Although these terms are almost synonymous, Speech recognition is sometimes used to describe the wider process of extracting meaning from speech, i.e. speech understanding[1].The term voice recognition should be avoided as it is often associated to the process of identifying a person from their voice, i.e. speaker recognition.

10.7.1.a How does it work?

Speech Recognition using Google Speech API:

All speech-to-text systems rely on at least two models: an acoustic model and a language model. In addition large vocabulary systems use a pronunciation model. It is important to understand that there is no such thing as a universal speech recognizer. To get the best transcription quality, all of these models can be specialized for a given language, dialect, application domain, type of speech, and communication channel.

Like any other pattern recognition technology, speech recognition cannot be error free. The speech transcript accuracy is highly dependent on the speaker, the style of speech and the environmental conditions. Speech recognition is a harder process than what people commonly think, even for a human being. Humans are used to understanding speech, not to transcribing it, and only speech that is well formulated can be transcribed without ambiguity.

From the user's point of view, a speech-to-text system can be categorized based in its use: command and control, dialog system, text dictation, audio document transcription, etc. Each use has specific requirements in terms of latency, memory constraints, vocabulary size, and adaptive features.

10.7.1.bInstallation

Google Speech API v2 is limited to 50 queries per day. Make sure you have a good microphone.
This is the installation guide for Ubuntu Linux. But this will probably work on other platforms is well. You will need to install a few packages: PyAudio, PortAudio and SpeechRecognition. PyAudio 0.2.9 is required and you may need to compile that manually.

```
git clone
http://people.csail.mit.edu/hubert/git/pyaudio.gitcdpyaudio
sudo python setup.py install
sudo apt-get installlibportaudio-dev
sudo apt-get install python-dev
sudo apt-get install libportaudio0 libportaudio2
libportaudiocpp0 portaudio19-dev
sudo pip3 install SpeechRecognition
```

This program will record audio from your microphone, send it to the speech API and return a Python string.The audio is recorded using the speech recognition module, the module will include on top of the program. Secondly we send the record speech to the Google speech recognition API which will then return the output.
r.recognize_google(audio) returns a string.

```
#!/usr/bin/env python3
# Requires PyAudio and PySpeech.

importspeech_recognitionassr
```

```
# Record Audio
r =sr.Recognizer()
withsr.Microphone() as source:
print("Say something!")
audio=r.listen(source)

# Speech recognition using Google Speech Recognition
try:
# for testing purposes, we're just using the default API key
# to use another API key, use `r.recognize_google(audio,
key="GOOGLE_SPEECH_RECOGNITION_API_KEY")`
# instead of `r.recognize_google(audio)`
print("You said: " + r.recognize_google(audio))
exceptsr.UnknownValueError:
print("Google Speech Recognition could not understand audio")
exceptsr.RequestErroras e:
print("Could not request results from Google Speech
Recognition service; {0}".format(e))
```

10.7.1.cInstalling the client library

You can either use a package manager or download and install the Python client library manually:

10.7.1.dManaged installation

Use pip or setuptools to manage your installation (you might need to run sudo first):

pip:

```
$ pip install --upgrade google-api-python-client
```

Setuptools: Use the easy_install tool included in the setuptools package:

```
$ easy_install --upgrade google-api-python-client
```

10.6.2.cManual installation:

Download the latest client library for Python, unpack the code, and run `python setup.py install`

10.8. Webcam

Rather than using the Raspberry Pi camera module, you can use a standard USB webcam to take pictures and video on the Raspberry Pi. The quality and configurability of the pi camera module is highly superior to a standard USB webcam.

10.8.1 Installation of fswebcam

First, install the fswebcam package:

```
sudo apt-get install fswebcam
```

10.8.2 Basic Usage:

Enter the command fswebcam followed by a filename and a picture will be taken using the webcam, and saved to the filename specified:

```
fswebcam image.jpg
```

This command will show the following information:

```
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 384x288 to 352x288.
```

```
--- Capturing frame...
Corrupt JPEG data: 2 extraneous bytes before marker 0xd4
Captured frame in 0.00 seconds.
--- Processing captured image...
Writing JPEG image to 'image.jpg'.
```

10.8.3Specify resolution:

The webcam used in this example has a resolution of 1280 x 720 so to specify the resolution I want the image to be taken at, use the -r flag:

```
fswebcam-s8 1280x720 image2.jpg
```

This command will show the following information:

```
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
--- Capturing frame...
Corrupt JPEG data: 1 extraneous bytes before marker 0xd5
Captured frame in 0.00 seconds.
--- Processing captured image...
Writing JPEG image to 'image2.jpg'.
```

10.9Drop Box

10.9.1 Dropbox Uploader

Dropbox Uploader is a BASH script which can be used to upload, download, delete, list files from Dropbox, an online file sharing, synchronization and backup service.
It is Portable and Secure. It's written in BASH scripting and only needs cURL (curl is a tool to transfer data from or to a server, available for all operating systems and installed by default

in many linux distributions).It's not required to provide your username/password to this script, because it uses the official Dropbox API v2 for the authentication process.

10.9.2 Features

- Cross platform
- Support for the official Dropbox API v2
- No password required or stored
- Simple step-by-step configuration wizard
- Simple and chunked file upload
- File and recursive directory download
- File and recursive directory upload
- Shell wildcard expansion (only for upload)
- Delete/Move/Rename/Copy/List/Share files
- Create share link
- Monitor for changes

10.9.3 Getting started with Drop Box

First, clone the repository using git (recommended)

```
git clone https://github.com/andreafabrizi/Dropbox-
Uploader.git
```

or download the script manually using this command:

```
curl"https://raw.githubusercontent.com/andreafabrizi/Dropbox-
Uploader/master/dropbox_uploader.sh" -o dropbox_uploader.sh
```

Then give the execution permission to the script and run it:

```
$chmod +x dropbox_uploader.sh
 $./dropbox_uploader.sh
```

The first time you run dropbox_uploader, you'll be guided through a wizard in order to configure access to your DropbSox. This configuration will be stored in ~/.dropbox_uploader.

10.9.4 Usage

The syntax is quite simple:

```
./dropbox_uploader.sh [PARAMETERS] COMMAND...
[%%]: Optional parameter
<%%>: Required parameter
```

The format used for dropbox uploader is

```
upload filename<local filename>
/dropboxuploader/dropboxuploader/subject name/home/pi/start.py
```

10.9.5 Available commands

- **upload** <LOCAL_FILE/DIR ...><REMOTE_FILE/DIR>
  Upload a local file or directory to a remote Dropbox folder.
  If the file is bigger than 150Mb the file is uploaded using small chunks (default 50Mb); in this case a. (dot) is printed for every chunk successfully uploaded and a * (star) if an error occurs (the upload is retried for a maximum of three times). Only if the file is smaller than 150Mb, the standard upload API is used, and if the -p option is specified the default curl progress bar is displayed during the upload process.
  The local file/dir parameter supports wildcards expansion.

- **download** <REMOTE_FILE/DIR> [LOCAL_FILE/DIR]
  Download file or directory from Dropbox to a local folder

- **delete** <REMOTE_FILE/DIR>
  Remove a remote file or directory from Dropbox

- **move** <REMOTE_FILE/DIR><REMOTE_FILE/DIR>
  Move or rename a remote file or directory

- **copy** <REMOTE_FILE/DIR><REMOTE_FILE/DIR>
  Copy a remote file or directory

- **mkdir** <REMOTE_DIR>
  Create a remote directory on Dropbox

- **list** [REMOTE_DIR]
  List the contents of the remote Dropbox folder

- **monitor** [REMOTE_DIR] [TIMEOUT]
  Monitor the remote Dropbox folder for changes. If timeout is specified, at the first change event the function will return.

- **share** <REMOTE_FILE>
  Get a public share link for the specified file or directory

- **saveurl** <URL><REMOTE_DIR>
  Download a file from an URL to a Dropbox folder directly (the file is NOT downloaded locally)

- **search** <QUERY> Search for a specific pattern on Dropbox and returns the list of matching files or directories

- **info**
  Print some info about your Dropbox account

- **space** Print some info about the space usage on your Dropbox account

- **unlink**
  Unlink the script from your Dropbox account

10.9.6 Optional parameterss

- **-f <FILENAME>**
  Load the configuration file from a specific file
- **-s**
  Skip already existing files when download/upload. Default: Overwrite

- **-d**
  Enable DEBUG mode

- **-q**
  Quiet mode. Don't show progress meter or messages

- **-h**

  Show file sizes in human readable format

- **-p**

  Show cURL progress meter

- **-k**

  Doesn't check for SSL certificates (insecure)

CHAPTER 11

PROGRAM

When a message is received on rasbperry pi, it starts executing the following program

11.1 start.py

```
frompubnub import Pubnub
importsubprocess
import pickle
pubnub=Pubnub(publish_key='pub-c-800c1668-57d0-45ec-9700-
74d251ed134b', subscribe_key='sub-c-5936b224-0eca-11e7-92d3-
02ee2ddab7fe')

def write(start,pic,upload,subject,date,time):
	v={
		'start':start,
		'pic':pic,
		'upload':upload,
		'sub':subject,
		'date':date,
		'time':time
		}
	with open('var.txt','wb')as handle:
		pickle.dump(v,handle)

def start(cmd):
	subs=['gnss','iafm','mcc','rs','project','seminar']
	ifcmd[1] in subs:
		write(True,False,False,cmd[1],cmd[2],cmd[3])
		Subprogram = subprocess.Popen(['xterm','-hold',
'-e', 'python ./speech1.py'],stdout=subprocess.PIPE)
# Only when the subject sent through the message matches with
one of the prelisted subjects, # the subprocess speech1.py
will be executed.
```

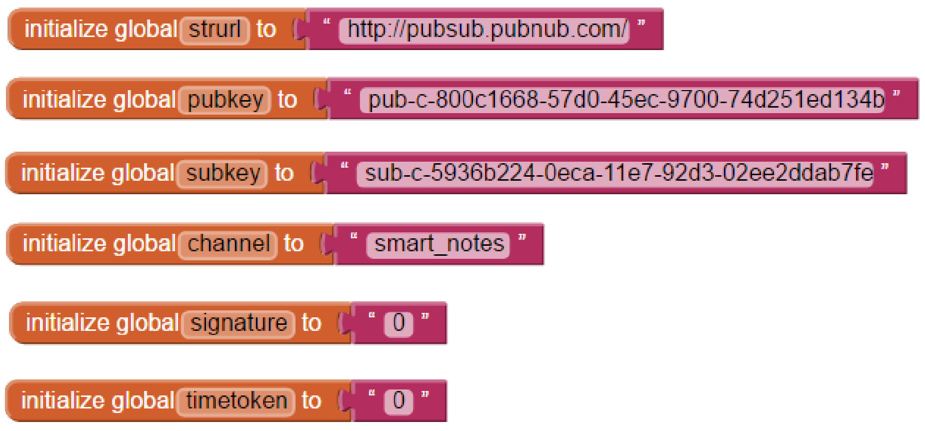
```
def _callback(m,channel):
       print(m)
cmd=m.split('+')
# each word in the message is separated by a '+' .
       #print(cmd[0])
       if(cmd[0]=='Start'):
              start(cmd)
       elif(cmd[0]=='stop'):
              write(False,False,False,cmd[1],cmd[2],cmd[3])
       elif(cmd[0]=='resume'):
              write(True,False,False,cmd[1],cmd[2],cmd[3])
       elif(cmd[0]=='picture'):
              write(True,True,False,cmd[1],cmd[2],cmd[3])
       elif(cmd[0]=='upload'):
              write(False,False,True,cmd[1],cmd[2],cmd[3])

#the first instruction given will be stored in cmd[0].
#If cmd[0]= 'start' or 'resume', then {start, pic,
upload}={1,0,0}. This is when the subprocess speech1.py will
be executed.
#If cmd[0]= 'picture', then {start, pic, upload}={1,1,0}
#If cmd[0]= 'stop', then {start, pic, upload}={0,0,0}
#If cmd[0]= 'upload', then {start, pic, upload}={0,0,1}

def _error(e):
print('error: '+str(e))

pubnub.subscribe(channels='smart_notes', callback=_callback,
error=_error)
```

11.2 speech1.py

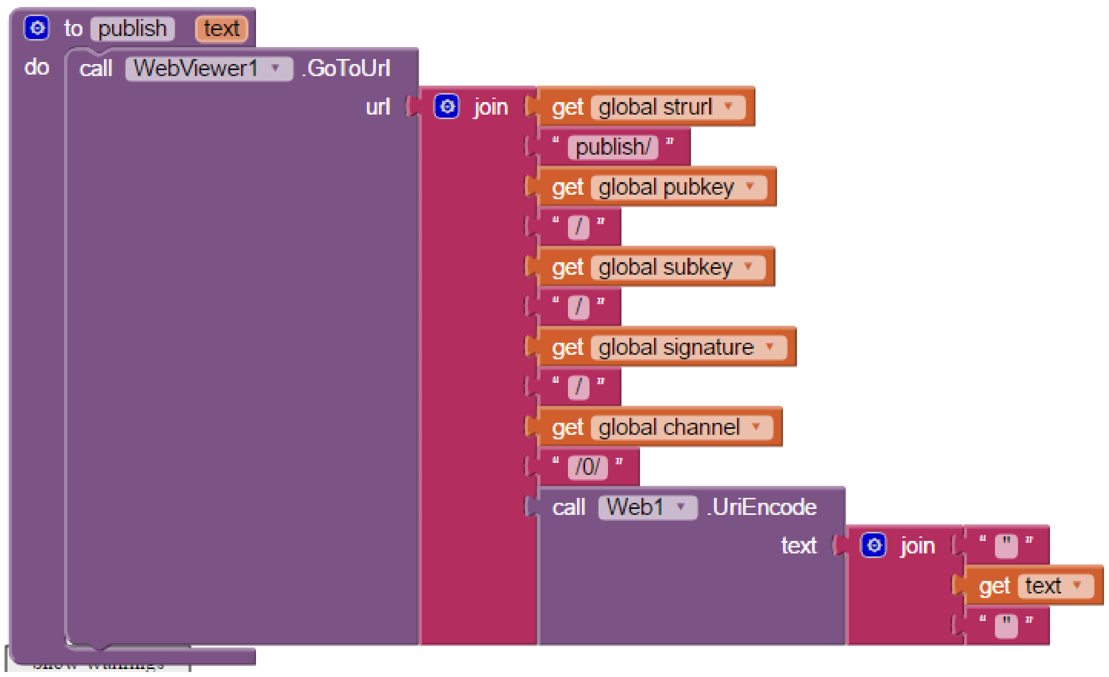
```
import pyaudio
from subprocess import call
import speech_recognition as sr
import pickle
# "Pickling" is the process whereby a Python object hierarchy
is converted into a byte stream
from docx import Document.
# This pickle module keeps track of the objects it has already
serialized, so that later references to the same object won't
be serialized again.

from docx.shared import Inches
import os

global start,pic,upload,sub,date,time

def create_file(file):
	doc=Document()
	doc.add_heading(file)
	doc.save(file)

def write_to_file(file,msg):
	doc=Document(file)
	doc.add_paragraph(msg)
	doc.save(file)

def take_picture(file):
	read()
	os.system('sudo fswebcam -r 640x420 test.jpg -S 8')
	doc=Document(file)
	doc.add_picture('/home/pi/test.jpg',width=Inches(1.25))
	doc.save(file)
```

```
def write(start,pic,upload,sub,date,time):
        v={
                'start':start,
                'pic':pic,
                'upload':upload,
                'sub':sub,
                'date':date,
                'time':time
                }
        with open('var.txt','wb')as handle:
                pickle.dump(v,handle)

def upload_file(filename):
      read()
      os.system('./Dropbox-Uploader/dropbox_uploader.sh upload
'+filename+' /'+sub+'/'+filename)
      os.system('sudo rm '+filename)

def read():
      with open('var.txt','rb') as handle:
      v=pickle.loads(handle.read())
      global start,pic,upload,sub,date,time
      start=v['start']
      pic=v['pic']
      upload=v['upload']
      sub=v['sub']
      date=v['date']
      time=v['time']

def listen(file):
      r = sr.Recognizer()
      r.energy_threshold=4000
      with sr.Microphone(device_index = 2) as source:
            print 'listening..'
            audio = r.listen(source)
            print 'processing'
      try:
            message = (r.recognize_google(audio, language =
'en-us', show_all=False))
            write_to_file(file,message)
      except:
            print('error')



read()
filename=str(sub)+str(date)+str(time)+'.docx'
create_file(filename)

while(upload==False):
        if(pic==True):
            take_picture(filename)
               print('picture taken')
            write(True,False,False,sub,date,time)
            read()
      elif(start==True):
            listen(filename)
            read()
      elif(start==False):
            read()
else:
      upload_file(filename)
      print('file uploaded')
```

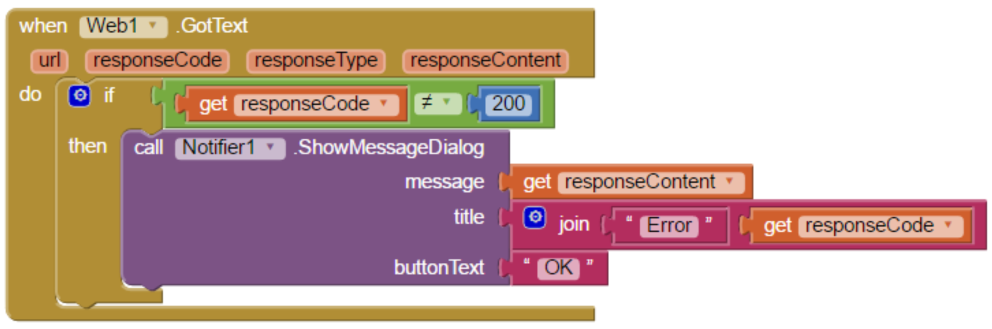
11.3 Program forDeveloping Application



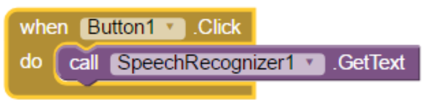
```
initialize global strurl to "http://pubsub.pubnub.com/"
initialize global pubkey to "pub-c-800c1668-57d0-45ec-9700-74d251ed134b"
initialize global subkey to "sub-c-5936b224-0eca-11e7-92d3-02ee2ddab7fe"
initialize global channel to "smart_notes"
initialize global signature to "0"
initialize global timetoken to "0"
```



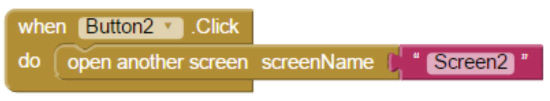
```
to publish text
do  call WebViewer1.GoToUrl
        url  join  get global strurl
                   "publish/"
                   get global pubkey
                   "/"
                   get global subkey
                   "/"
                   get global signature
                   "/"
                   get global channel
                   "/0/"
                   call Web1.UriEncode
                       text  join  """
                                   get text
                                   """
```



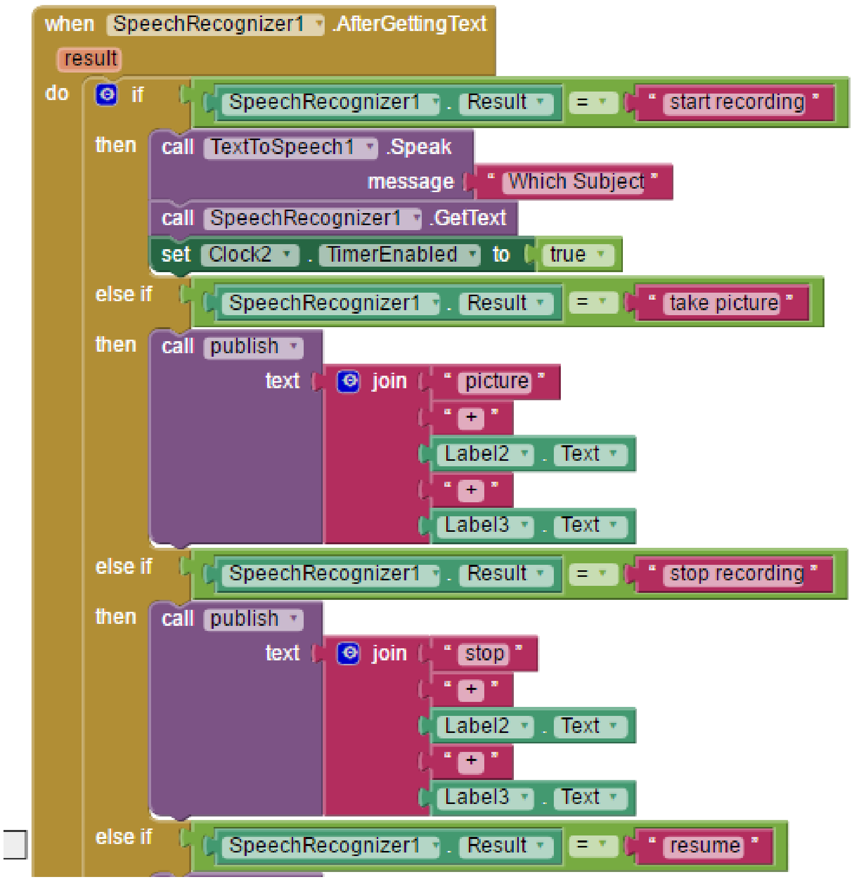
```
when Web1.GotText
  url  responseCode  responseType  responseContent
do  if  get responseCode ≠ 200
    then  call Notifier1.ShowMessageDialog
              message     get responseContent
              title       join  "Error"
                                get responseCode
              buttonText  "OK"
```



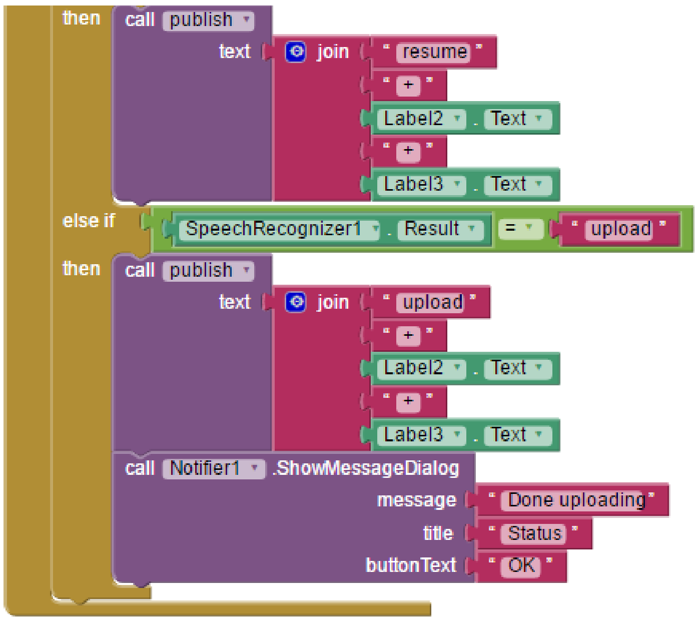
```
when Button1.Click
do  call SpeechRecognizer1.GetText
```



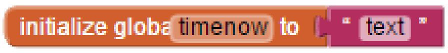
```
when Button2.Click
do  open another screen  screenName  "Screen2"
```



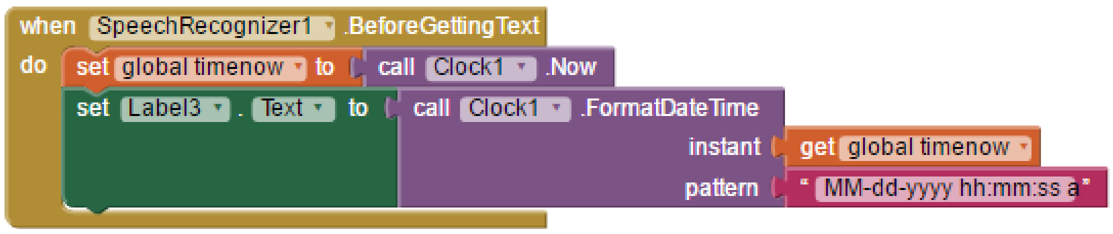
```
when SpeechRecognizer1 .AfterGettingText
 result
do  if  SpeechRecognizer1 . Result  =  " start recording "
    then  call TextToSpeech1 .Speak
                              message  " Which Subject "
          call SpeechRecognizer1 .GetText
          set Clock2 . TimerEnabled to  true
    else if  SpeechRecognizer1 . Result  =  " take picture "
    then  call publish
               text  join  " picture "
                           " + "
                           Label2 . Text
                           " + "
                           Label3 . Text
    else if  SpeechRecognizer1 . Result  =  " stop recording "
    then  call publish
               text  join  " stop "
                           " + "
                           Label2 . Text
                           " + "
                           Label3 . Text
    else if  SpeechRecognizer1 . Result  =  " resume "
```



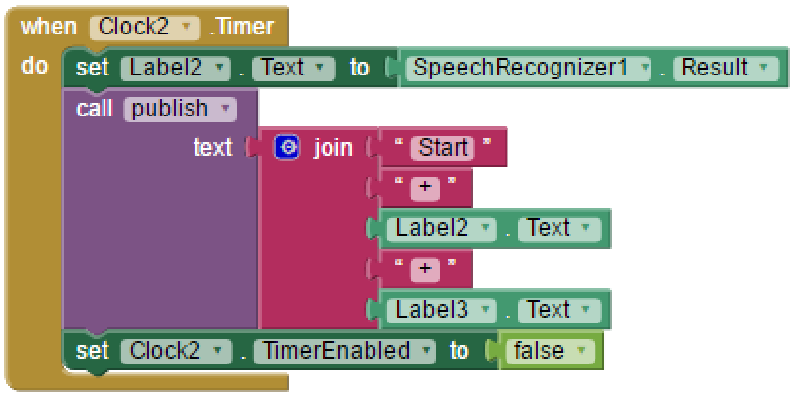
```
    then  call publish
               text  join  " resume "
                           " + "
                           Label2 . Text
                           " + "
                           Label3 . Text
    else if  SpeechRecognizer1 . Result  =  " upload "
    then  call publish
               text  join  " upload "
                           " + "
                           Label2 . Text
                           " + "
                           Label3 . Text
          call Notifier1 .ShowMessageDialog
                            message  " Done uploading "
                              title  " Status "
                         buttonText  " OK "
```



```
initialize global timenow to " text "
```



```
when SpeechRecognizer1 .BeforeGettingText
do  set global timenow to  call Clock1 .Now
    set Label3 . Text to  call Clock1 .FormatDateTime
                                      instant  get global timenow
                                      pattern  " MM-dd-yyyy hh:mm:ss a "
```



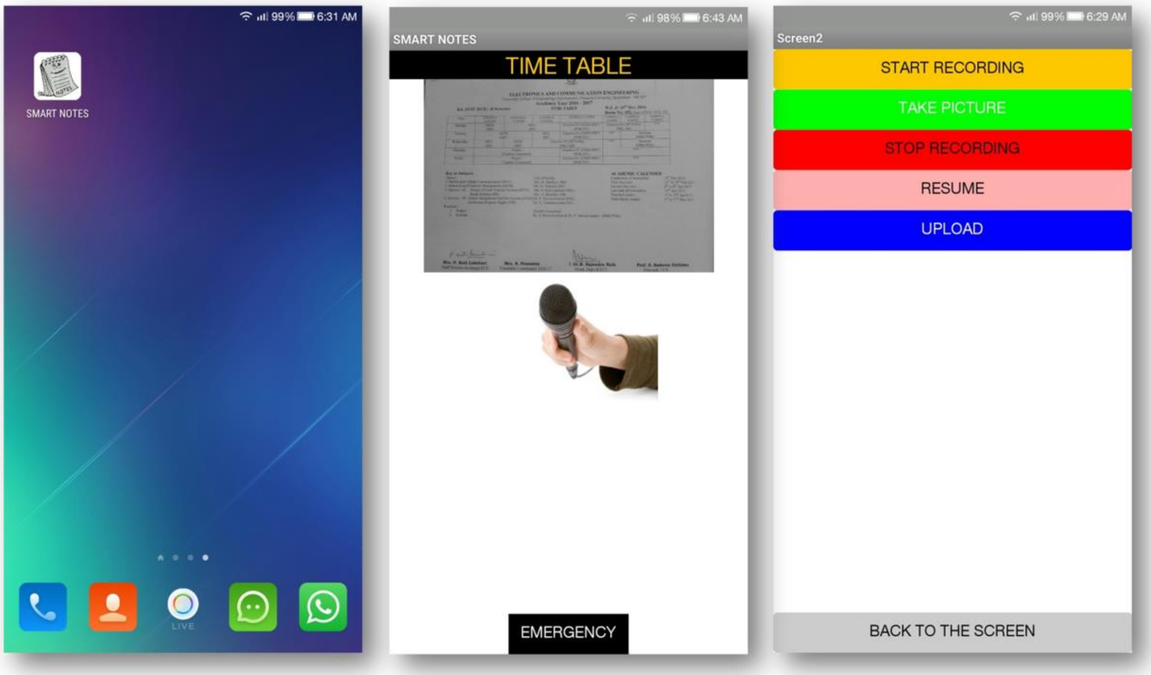
```
when Clock2 .Timer
do  set Label2 . Text to  SpeechRecognizer1 . Result
    call publish
              text  join  " Start "
                          " + "
                          Label2 . Text
                          " + "
                          Label3 . Text
    set Clock2 . TimerEnabled to  false
```

CHAPTER12

RESULT

12.1 Result

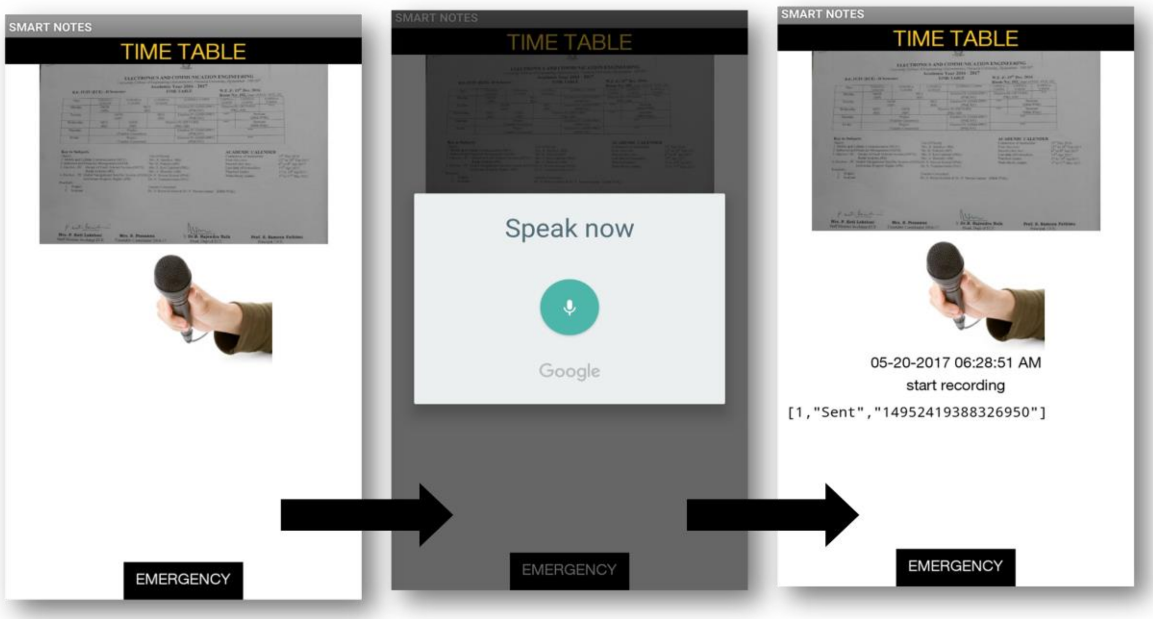
By typing python start.py in putty terminal, the app gets ready to take the input.



SCREEN 1 SCREEN 2

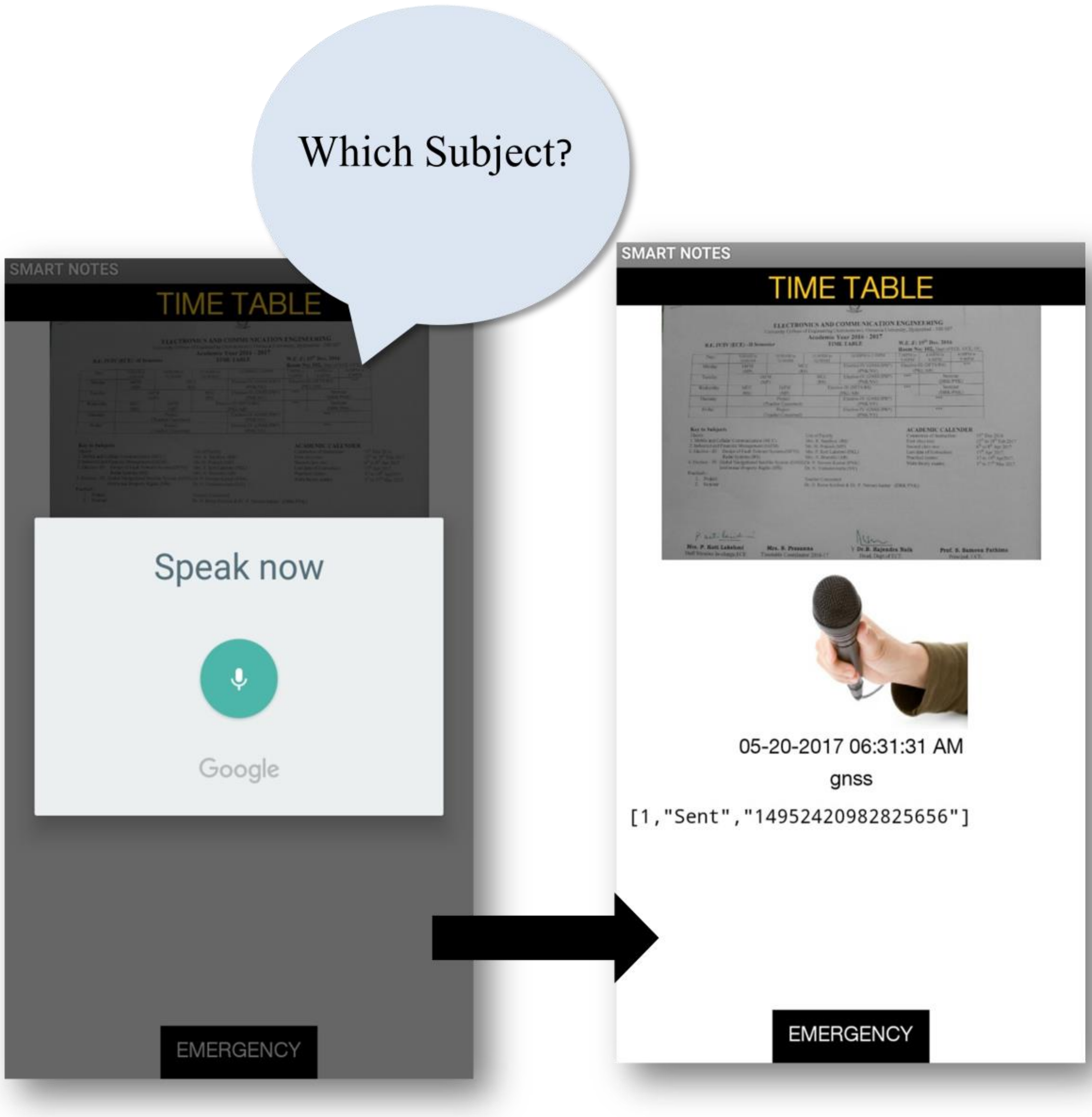
Now open the app and press the mic button. The app starts listening to us.

When "Start Recording" command is given…



```
05-20-2017 06:28:51 AM
start recording
[1,"Sent","14952419388326950"]
```

When it recognizes "start recording" from our speech, It will send a message to raspberry pi which in turn instructs the app to ask "Which Subject?. When we select a subject, It will start recording until either resume or upload commands are given in the app.



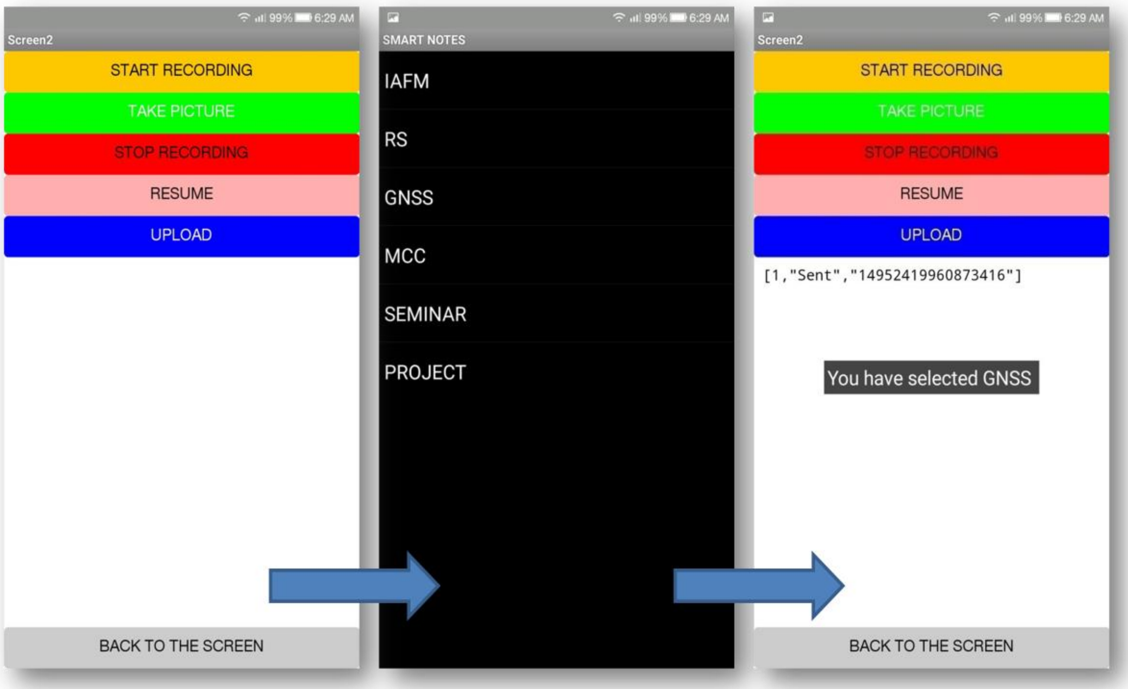
05-20-2017 06:31:31 AM
gnss

```
[1,"Sent","14952420982825656"]
```

If we want to take a picture, we can give a command by saying "Take Picture". Pi will instruct the webcam to take a picture. Now when the "upload" command is given, the audio file which has been recorded will be converted into text in the form of a .docx file and uploads it into the dropbox by creating a file with the subject name(which we have selected) ,date and time. The pictures which are taken during the recording process will also get uploaded in the same docx file.

The same operation can be done using Screen 2 which appears when ‘Emergency’ button is pressed on Screen 1

When “Start Recording” button is pressed in Screen 2, an other window appears showing the list of subjects. A subject has to be selected then.



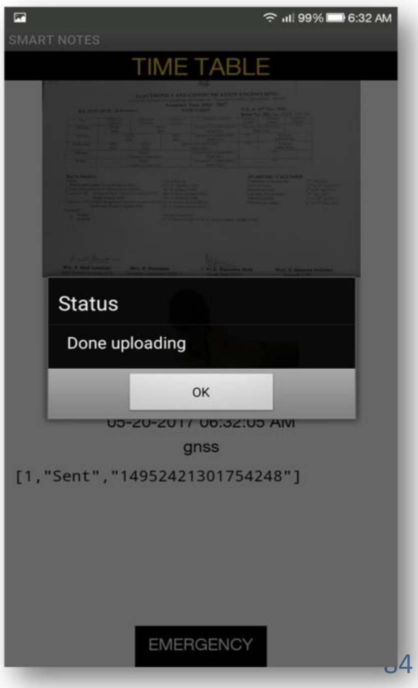
```
[1,"Sent","14952419960873416"]
```

You have selected GNSS

When the upload command is given …

SCREEN 1

SCREEN 2



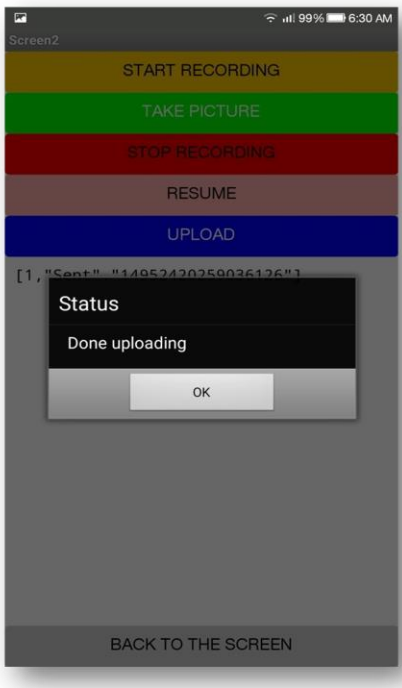
Status

Done uploading

05-20-2017 06:32:05 AM

gnss

```
[1,"Sent","14952421301754248"]
```

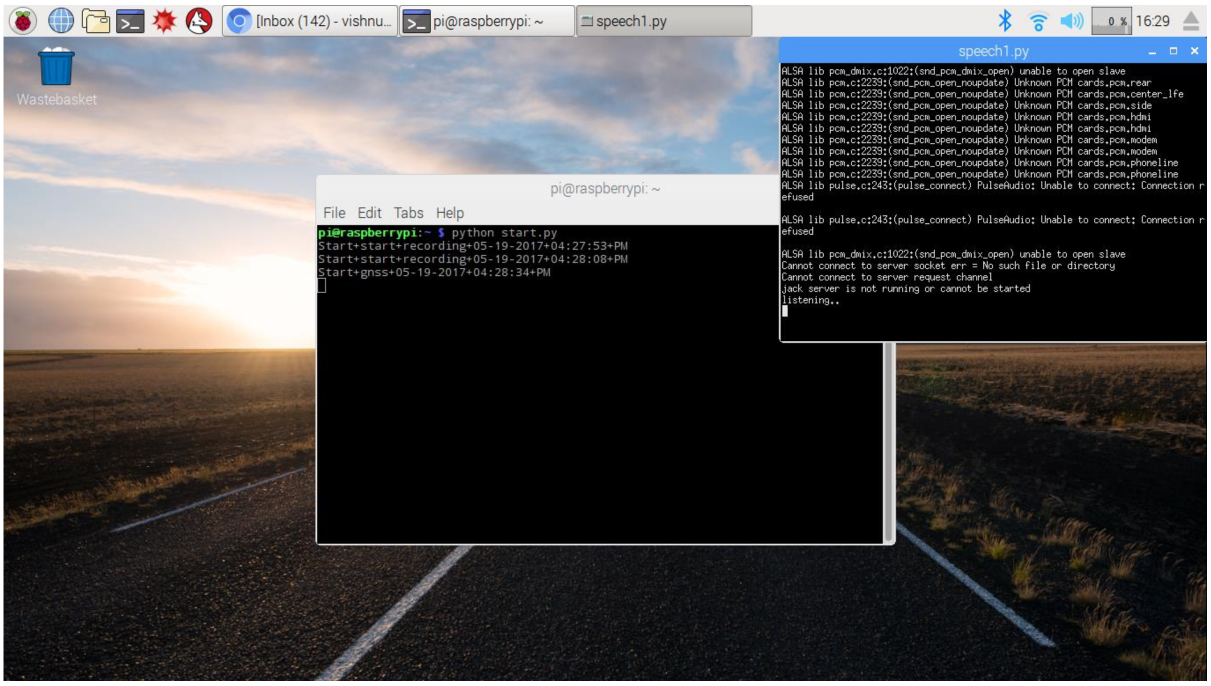


```
[1,"Sent","14952420259036126"]
```

Status

Done uploading

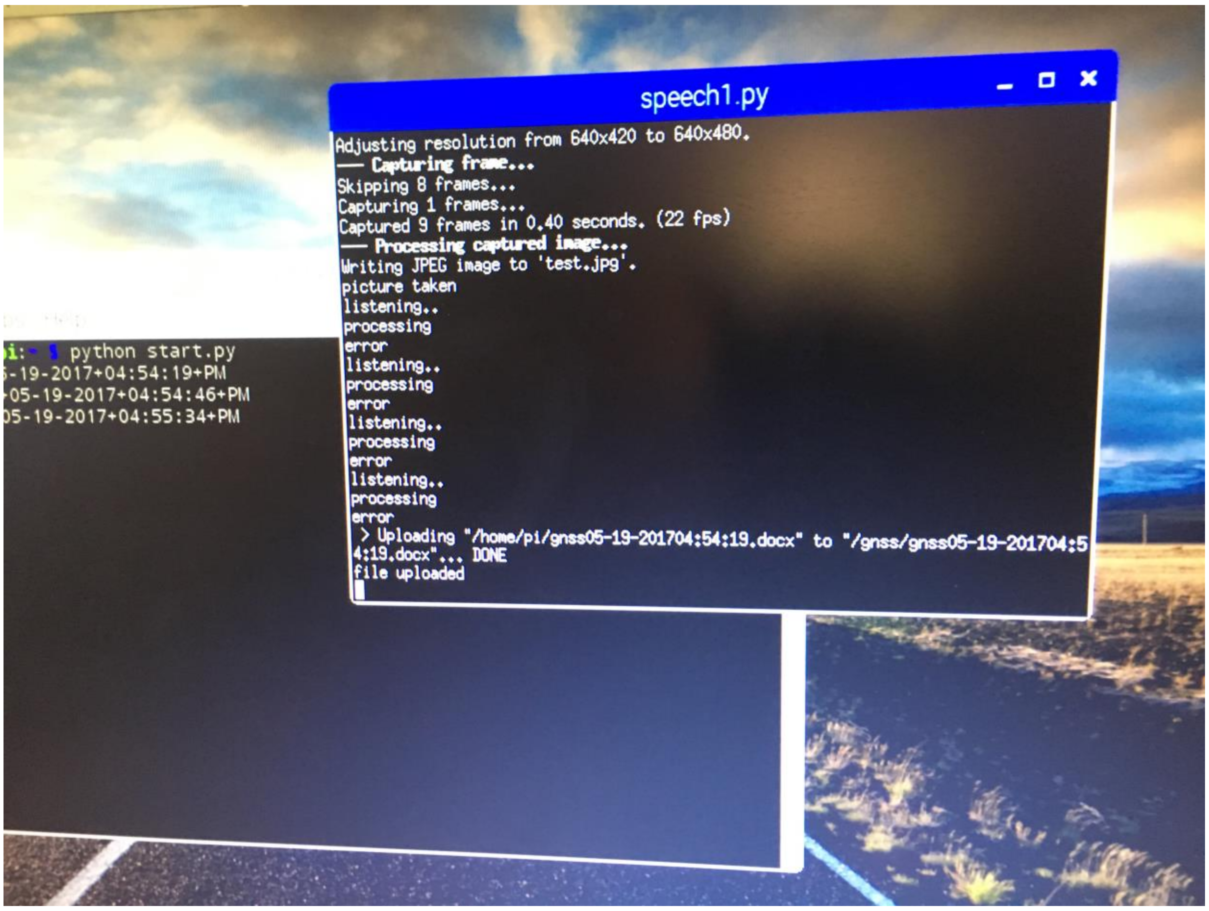
The terminal looks like this after the entire process is done.



```
pi@raspberrypi:~ $ python start.py
Start+start+recording+05-19-2017+04:27:53+PM
Start+start+recording+05-19-2017+04:28:08+PM
Start+gnss+05-19-2017+04:28:34+PM
```

```
ALSA lib pcm_dmix.c:1022:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pulse.c:243:(pulse_connect) PulseAudio: Unable to connect: Connection r
efused

ALSA lib pulse.c:243:(pulse_connect) PulseAudio: Unable to connect: Connection r
efused

ALSA lib pcm_dmix.c:1022:(snd_pcm_dmix_open) unable to open slave
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
listening..
```

Fig 12.1 Result 1



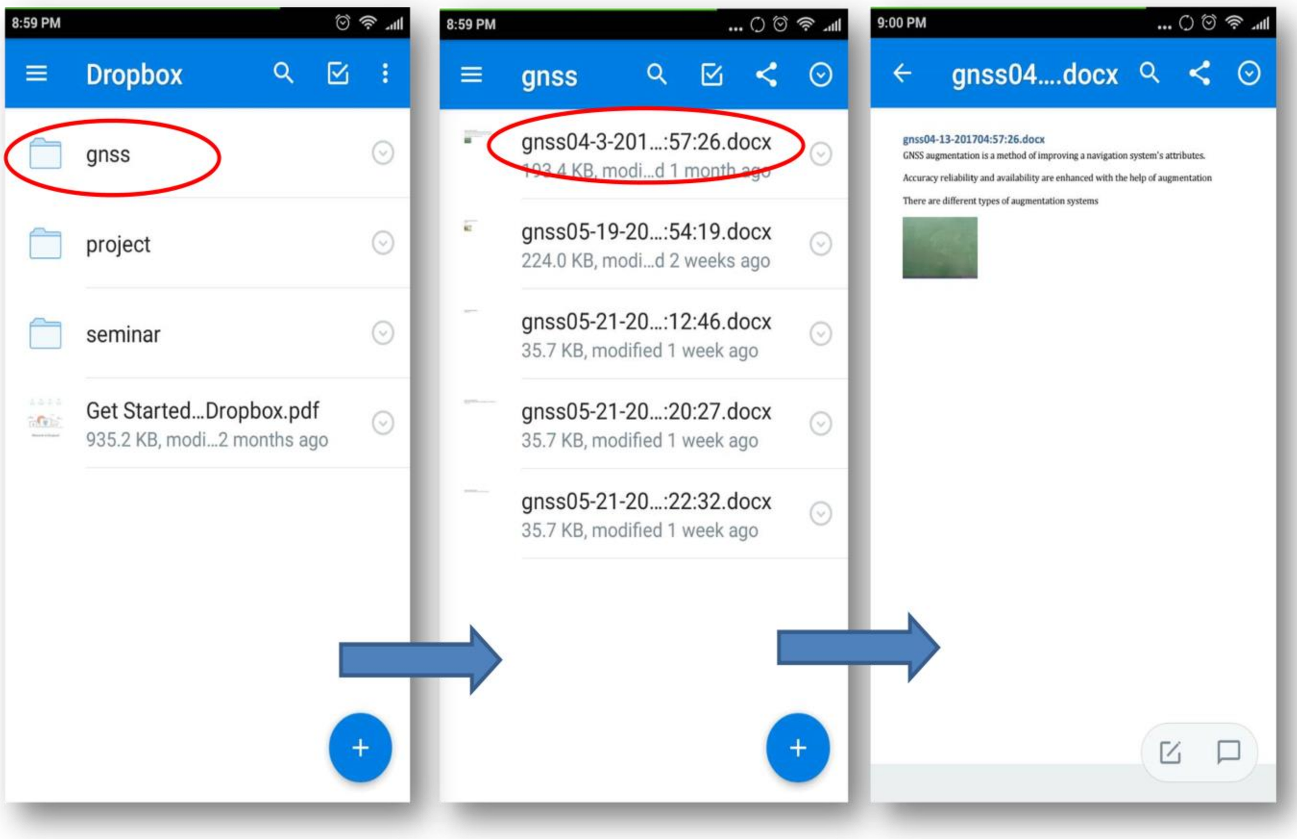
```
Adjusting resolution from 640x420 to 640x480.
--- Capturing frame...
Skipping 8 frames...
Capturing 1 frames...
Captured 9 frames in 0.40 seconds. (22 fps)
--- Processing captured image...
Writing JPEG image to 'test.jpg'.
picture taken
listening..
processing
error
listening..
processing
error
listening..
processing
error
listening..
processing
error
 > Uploading "/home/pi/gnss05-19-201704:54:19.docx" to "/gnss/gnss05-19-201704:5
4:19.docx"... DONE
file uploaded
```

```
i:~ $ python start.py
5-19-2017+04:54:19+PM
+05-19-2017+04:54:46+PM
05-19-2017+04:55:34+PM
```

Fig 12.2 Result 2

12.2 Dropbox Result

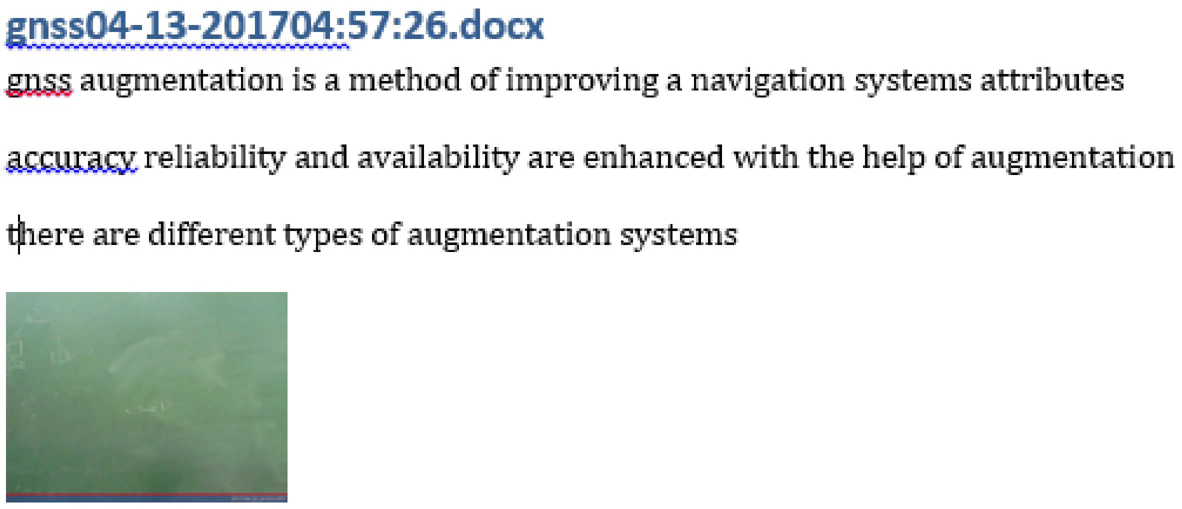
When the upload command is given. The document will be uploaded in the associated dropbox account. Separate folders will be created for each subject and the document will be saved with the name of the subject along with the date and time



gnss
project
seminar
Get Started...Dropbox.pdf
935.2 KB, modi...2 months ago

gnss04-3-201...:57:26.docx
103.4 KB, modi...d 1 month ago
gnss05-19-20...:54:19.docx
224.0 KB, modi...d 2 weeks ago
gnss05-21-20...:12:46.docx
35.7 KB, modified 1 week ago
gnss05-21-20...:20:27.docx
35.7 KB, modified 1 week ago
gnss05-21-20...:22:32.docx
35.7 KB, modified 1 week ago

gnss04-13-201704:57:26.docx
GNSS augmentation is a method of improving a navigation system's attributes.
Accuracy reliability and availability are enhanced with the help of augmentation
There are different types of augmentation systems



gnss04-13-201704:57:26.docx

gnss augmentation is a method of improving a navigation systems attributes

accuracy reliability and availability are enhanced with the help of augmentation

there are different types of augmentation systems

12.3 Fast Jot Kit and Complete Setup

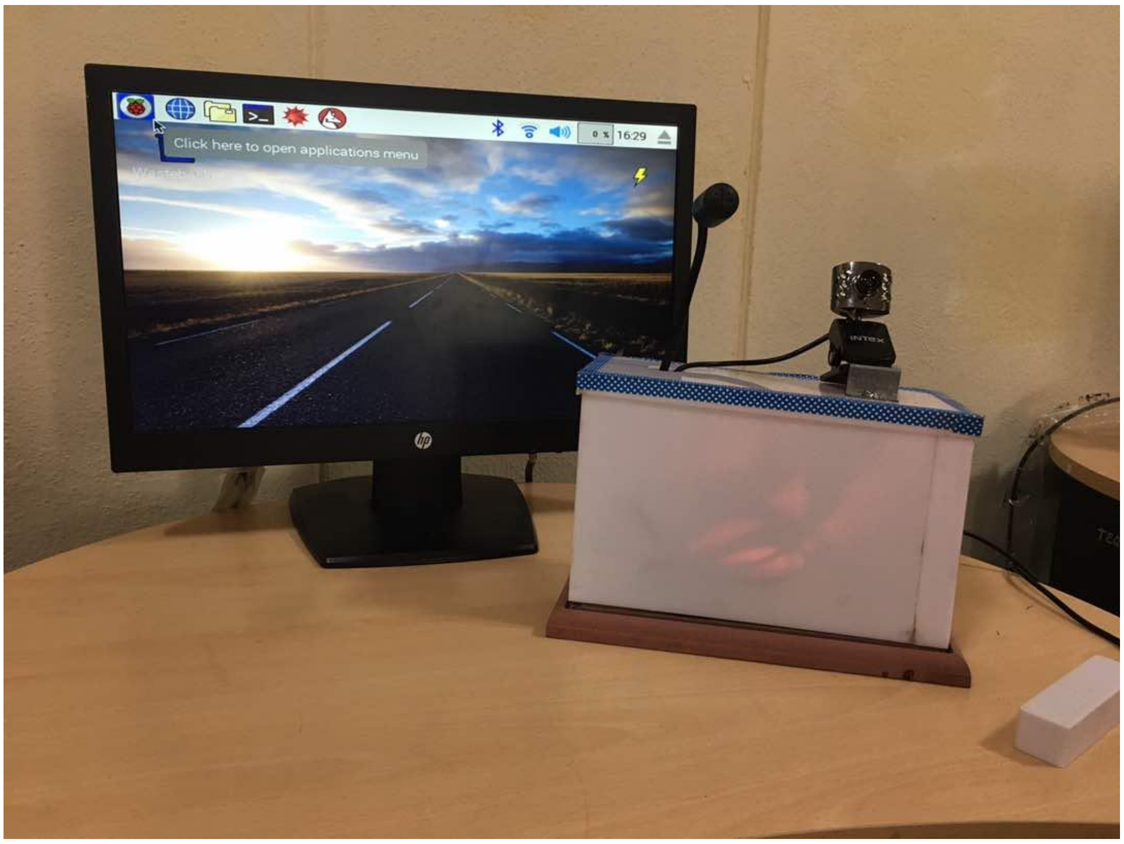


Fig 12.3 Fast Jot

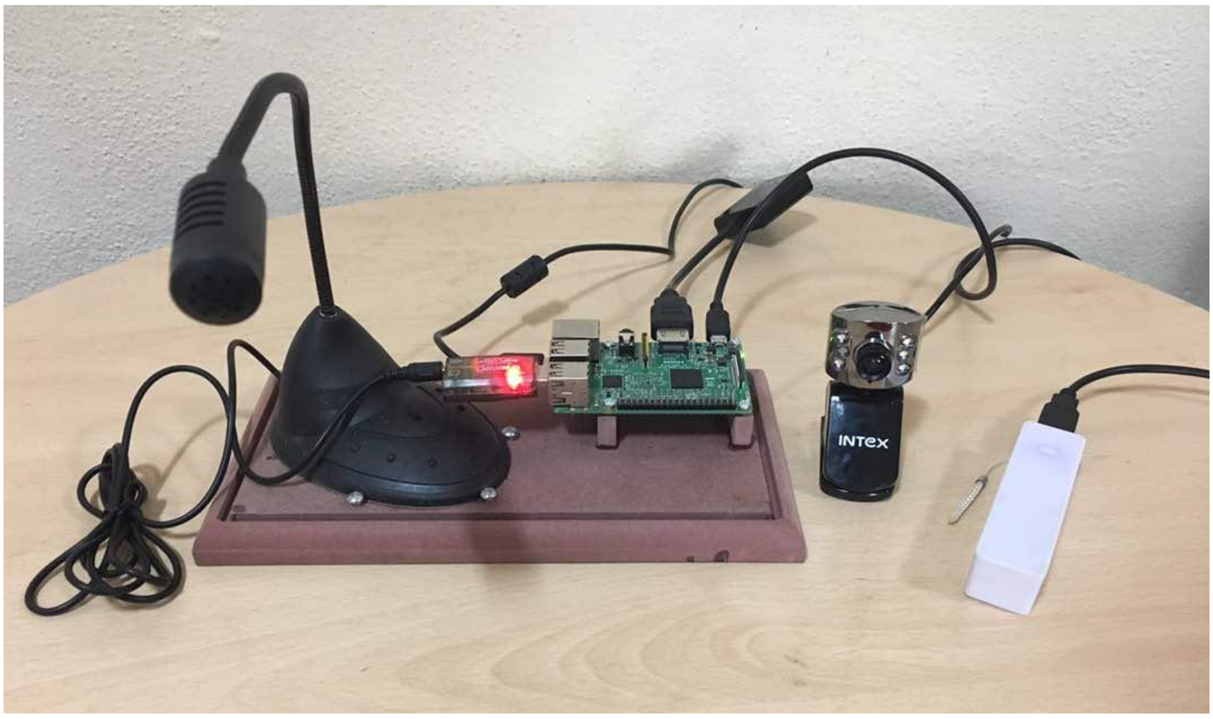


Fig 12.4 Internal Components of Fast Jot

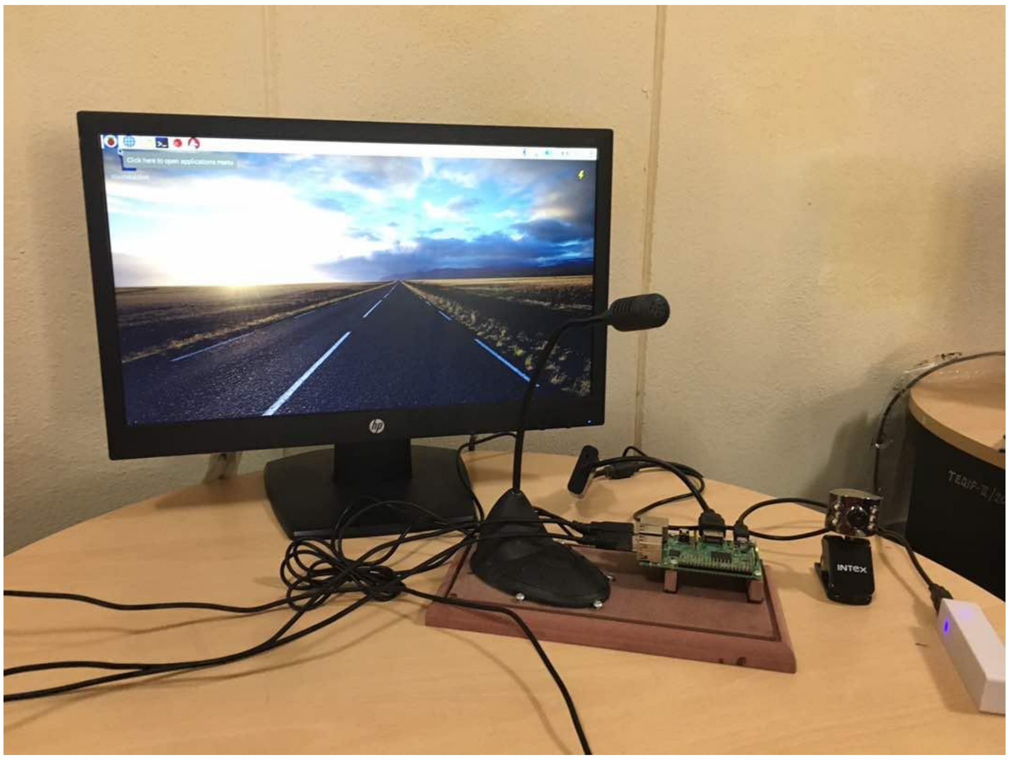


Fig 12.5 Setup

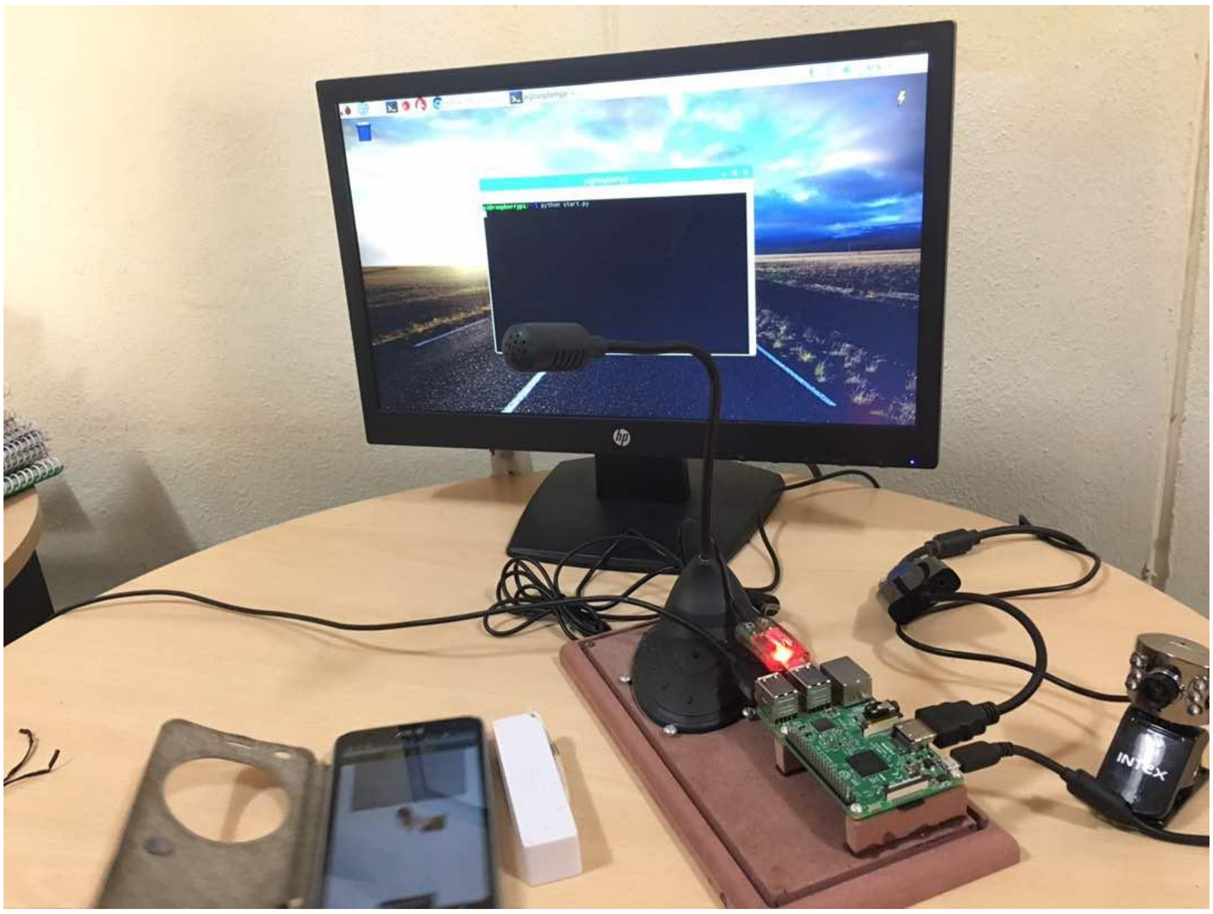


Fig 12.6 Setup

CHAPTER 13

CONCLUSION AND FUTURE SCOPE

13.1 Conclusion

In this project, we have designed and developed a prototype for fast jot which improves the way notes are taken practically by the students. This model requires better and high speed internet access for faster conversion of speech to text and to upload files to dropbox by raspberry pi. This is an efficient and eco-friendly design The results of the execution were as predicted and satisfactory.

Fast jot can be accessed by any student in the class easily by just sending commands through an app named ‘smart notes’, which has been developed using mit app inventor. Students can access organised notes along with pictures through their account in dropbox.

13.2 Future Scope

Stuttering is a speech disorder which is affecting millions of people in their day to day life. Stuttering is a complex speech problem which affects the verbal communication. . A person who stutters may repeat the first part of a word or hold a single sound for a long time, this disorder is characterized by disruptions in the production of speech sounds, called disfluencies. This can be rectified by implementing algorithms of speech repetition and prolongation using Artificial Neural Networks (ANN).

The recognizer is designed to ignore background voices and noise without additional noise-cancelling. However, for optimal results, position the microphone as close to the user as possible, particularly when background noise is present. We use google speech recognition in this project, it listens only for limited amount of time. This limitation can be overcome by purchasing google speech recognition API license. For better quality of pictures in the .docx file created, pi camera can be used.

REFERENCES

[1]Ms. Sneha K. Upadhyay,Mr. Vijay N. Chavda,"Intelligent system based on speech recognition with capability of self learning" ,International Journal For Technological Research In Engineering ISSN (Online): 2347 - 4718 Volume 1, Issue 9, May-2014

[2] Powers, Shawn. "The open-source classroom: your firstbite of raspberry pi." Linux Journal 2012.224 (2012):

[3]G.Senthilkumar1, K.Gopalakrishnan2, V. Sathish Kumar3 Embedded Image Capturing System Using Raspberry Pi System, Volume 3, Issue 2March–April 2014

[4]J.baker,lideng,jim glass "research development in speech recognition and understanding" IEEE signal procession magazine vol23.no.3,pp-75-80,2009

[5] ZhouZhe, "ARM-Based Embedded Linux System For Wireless Video Monitor applications", Department ofInformation Engineer, Beijing University of Post andTelecommunication, Beijing(100876)

[6] .Ryu, Yeonghyeon, JeakyuYoo, and Youngroc Kim. "Cloud services based Mobile monitoring systems." Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4thInternational Conference on. IEEE, 2012.

[7] Daniel Jakubisin,MarshallDavis,CaseyRoberts,Dr. Ivan Howitt,"Real-Time Audio Transceiver Utilizing 802.11b Wireless Technology",IEEE,2007

[8] Matt Richardson, Shawn Wallace, "Getting Started with Raspberry", Brian Jepson, O'Reilly Media Inc., United States of America, first edition, pp.10-31,December 2012.

[9] Prachikhilari, bhope v. (july 2015) "Online speech to text engine" International journal of innovative research in science, engineering andtechnology. vol. 4, issue 7, july 2015.

[10] Ansari ImaranBhatkarFirozChoudhary Faisal and Khan Sharukh "Smart Lecture DeliverySystem using Raspberry Pi" International Journal Of Advanced Research in ComputerEngineeringTechnology(IJARCET) vol 4 issue 3, March 2015.

[11] T. A. M. Phan, J. K. Nurminen, and M. Di Francesco, "Cloud Databases for Internet-of-Things Data," 2014, pp. 117–124.

[12] I. Drago, M. Mellia, M. M. Munafò, A. Sperotto, R. Sadre, and A. Pras, "Inside Dropbox: Understanding Personal Cloud Storage Services," in Proc. of the 12th ACM Internet Measurement Conference, 2012.

[13] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape," in Proc. of the ICSE Workshop on Soft. Eng. Challenges of Cloud Computing, 2009.

[14] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and privacy in cloud computing: A survey," in Proc. of the International Conference on Semantics, Knowledge and Grid, 2010.

[15] R. Baishakhi, D. Posnett, V. Filkov, P. Devanbu, "A Large Scale Study of Programming Languages and Code Quality in Github", Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 155-165, 2014.