P.E.S COLLEGE OF ENGINEERING,
MANDYA.

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi.)

A Report on

"HAND GESTURES RECOGNITION SYSTEM AND HOME AUTOMATION FOR BLIND,DEAF AND MUTE PEOPLE"

Submitted in partial fulfilment of the requirement for
the **PROJECT** of the
BACHELOR OF ENGINEERING DEGREE

Submitted by

SHREYAS.P	**[4PS15CS107]**
NISCHAL.C.S	**[4PS15CS066]**
SHUBHASHREE.M	**[4PS15CS108]**
SANJAY.S	**[4PS15CS095]**

Under the guidance of

RAMYASHREE H P
Assistant Professor

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany successful completion of any task would be incomplete without the mention of people who made it possible. We take this opportunity to express our sincere gratitude to all those who have helped us in this project.

We have immense pleasure in expressing our thanks to our Principal **Dr.H.V.Ravindra,** PESCE, Mandya for providing all the facilities for the successful completion of the project.

It is our great privilege to express our sincere and heartfelt thanks to our Head of the department **Dr.M.C.Padma**, Department of Computer Science and Engineering, for her constant encouragement, valuable suggestions and support in bringing out this dissertation successfully.

We would also like to take this opportunity to acknowledge the guidance from **Mrs.Ramyashree.H.P, Associate Professor**, who extended full support and cooperation at every stage of our project period.

Also, we would like to express our gratitude to all the teaching and non-teaching staff for their kind co-operation and support during the course of our project work.

Finally, we would like to thank our parents and all our friends for their constant support.

SHREYAS.P [4PS15CS107]

NISCHAL.C.S [4PS15CS066]

SHUBHASHREE [4PS15CS108]

SANJAY .S [4PS15CS095]

DECLARATION

We, **SHREYAS.P, NISCHAL.C.S, SHUBHASHREE.S, SANJAY.S,**

students of Eighth Semester, B.E, Computer Science & Engineering, from PES College of Engineering, Mandya, hereby declare that this project work entitled "**Hand Gestures Recognition System for Deaf, Mute and Blind People**" has been independently carried out by us under the guidance of **Mrs.Ramyashree.H.P, Associate Professor**, Department of CS&E, PESCE, Mandya and submitted in partial fulfilment of the requirement for the award of the degree of Bachelor o of Engineering in Computer Science Engineering during the academic year 2019-20.

We further declare that the matter embedded in this report has not been submitted previously for the award of any degree to any other university.

SHREYAS.P [4PS15CS107]

PLACE: **NISCHAL.C.S [4PS15CS066]**

DATE: **SHUBHASHREE [4PS15CS108]**

SANJAY .S [4PS15CS095]

Table of Contents

ABSTRACT

Visual impairment is one of the biggest limitation for humanity, especially in this day and age when information is communicated a lot by text messages (electronic and paper based) rather than voice. The device we have proposed aims to help people with visual impairment. The device helps deaf, mute and blind people to have basic home automation using hand gestures. In this project, we developed a device that converts an image's text to speech and basic control of some devices. The basic framework is an embedded system that captures an image, extracts only the region of interest (i.e. region of the image that contains text) and converts that text to speech and controlling of devices. It is implemented using a Raspberry Pi and a Raspberry Pi camera. The captured image undergoes a series of image pre-processing steps to locate only that part of the gestures and removes the background. Two tools are used convert the new image (which contains only the text) to speech. They are OCR (Optical Character Recognition) software and TTS (Text-to-Speech) engines. The audio output is heard through the raspberry pi's audio jack using speakers or earphones.

Keywords: Text to speech,smart home controls,gesture recognition,deaf and mute,

Blind,

CHAPTER 1

INTRODUCTION

In our planet of 7.4 billion humans, 285 million are visually impaired out of whom 39 million people are completely blind, i.e. have no vision at all, and 246 million have mild or severe visual impairment (WHO, 2011). It has been predicted that by the year 2020, these numbers will rise to 75 million blind and 200 million people with visual impairment. As reading is of prime importance in the daily routine (text being present everywhere from newspapers, commercial products, sign-boards, digital screens etc.) of mankind, visually impaired people face a lot of difficulties. Our device assists the visually impaired by reading out the text to them. There have been numerous advances in this area to help visually impaired to read without much difficulties. The existing technologies use a similar approach as mentioned in this paper, but they have certain drawbacks. Firstly, the input images taken in previous works have no complex background, i.e. the test inputs are printed on a plain white sheet. It is easy to convert such images to text without pre-processing, but such an approach will not be useful in a real-time system. Also, in methods that use segmentation of characters for recognition, the characters will be read out as individual letter and not a complete word. This gives an undesirable audio output to the user. For our project, we wanted the device to be able to detect the text from any complex background and read it efficiently. Inspired by the methodology used by Apps such as “Cam Scanner”, we assumed that in any complex background, the text will most likely be enclosed in a box eg billboards, screens etc. By being able to detect a region enclosing four points, we assume that this is the required region containing the text. This is done using warping and cropping. The new image obtained then undergoes edge detection and a boundary is then drawn over the letters. This gives it more definition. The image is then processed by

the OCR and TTS to give audio output. Another use of our device is it also helps in home automation. Basic control of devices such as on/off of light, fan, locking/unlocking of door, play/pause of music and other few automation can be achieved. The ADXL sensor uses its onboard gyroscope to recognize the hand gestures made by deaf, mute and blind people. Raspberry pi converts these gestures into predefined controls. The pre-defined controls are then processed to get the desired output.

OBJECTIVE OF PROPOSED SYSTEM

Our work aims to help visually impaired people to easily read and helping deaf and mute people by providing basic home automation.

The steps followed are as follows-:

- Hand Gesture Recognition
- Image acquisition
- Image pre-processing
- Image to text conversion
- Text to speech conversion
- Colour recognition
- Control of devices

CHAPTER 2

LITERATURE SURVEY

Literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and extent of the project .

- **Paper 1:**A smart reader for visually impaired people using Raspberry PI[2017]. International journal of Engineering science on volume 6 by D.Velmurugan, M.S.Sonam, S.Umamaheshwari, K.R.Arun.
- **Paper 2:** Image text to speech conversion using OCR technique in Raspberry Pi,[2018] International journal of advanced research in electrical,electronics and instrumentation engineering of volumes by K.Nirmala kumara, meghana reddy.
- **Paper 3:** Sivo Ferreirn , c'eline Thillou,Bernald Gosselin. From picture to speech:[2018]An innovative application for embedded environment and the circuits . et Traitement du signals.
- **Paper 4**:Mirjana, marksinovic, Vladimir vyjrvic, Nikola davidovic, Vladimir Milosevic and Branko Perisic [2018] Raspberry PI as interest of things hardware performances and constraints.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing system and its limitations

In all of the Project on deaf, mute and blind the system which was mainly concentrated on the security and safety of the disabled and using their own predefined gestures which used for operating and maintaining their requirements and necessary precaution for their regular livelihood. They also worked on the health issue on how to monitoring the health and their regular interaction with the desired and respective doctors or care takers.

3.2 Proposed system

In the project we have taken up, we mainly concentrate on the basic home automation which can be done by the disabled by considering with the lowest cost IOT device which must be free handled for all. We have also concentrated on the basic interaction among normal people to make it user friendly. The text to speech was a most interested task which was mainly taken into consideration for the disabled to interact among themselves and with the normal humans.

3.3 Problem Statement

To make the disabled people to get awareness on the basic home automation uses which has to be used by them like normal people and to make them feel like they're not different. That they are one among like all the others who are able to listen, speak and recognise.

3.3.1 Functional Requirements

Functional Requirement describes what the system should do, i.e. the service provided for the Users. Many embedded system have substantially different design according to their functions and utilities.

3.3.2 Non- Functional Requirements

Non-functional requirements is a requirement that specifies criteria that can be used to judge

The operation of a system, rather than specific behaviors. Functional requirements define

What a system is supposed to do whereas non-functional requirements define how a system is

Supposed to be. The non-functional requirements are the constraints or the environment in Which the software is developed.

Reliability-

The Application should be reliable in providing 24/7 uninterrupted service.

3.4 Hardware Requirements:

- RASPBERRY PI
- ADXL SENSOR
- 5V RELAY MODULES
- PI CAMERA
- COLOUR BANDS
- EARPHONES
- CONNECTING CABLES

3.5 Software Requirements:

- RASPBIAN OS
- PYTHON
- VNC VIEWER
- TESSERACT OCR
- WINDOWS 7 OR HIGHER

CHAPTER 4

4.1 SYSTEM DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us toward how

To satisfy the needs. The design of a system is perhaps the most critical factor affecting the Quality of the software; it has a major impact on the later phases particularly testing and Maintenance.

The design activity often results in three separate outputs – • Architecture design.

- High level design.
- Detailed design.

Architecture Design:

Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired result. The focus is on Identifying components or subsystems and how they connect. In other words, the focus is on What major components are needed?

High Level Design:

In high level design identifies the modules that should be built for developing the system and the specifications of these modules. At the end of system design all major data structures, file format, output formats, etc., are also fixed. The focus is on identifying the modules. In other words, the attention is on what modules are needed.

Detailed Design:

In the detailed design the internal logic of each of the modules is specified. The focus is on Designing the logic for each of the modules. In other words how modules can be implemented in software is the issue.

A design methodology is a systematic approach to creating a design by application of a set of Techniques and guidelines. Most methodologies focus on high level design.

4.2 Architecture of Internet of Things

Architecture of internet Of Things contains basically 4 layers:

• Application Layer

• Gateway and the network layer

• Management Service layer

• Sensor layer

➢ Application Layer

• Lowest Abstraction Layer

• With sensors we are creating digital nervous system.

• Incorporated to measure physical quantities

• Interconnects the physical and digital world

• Collects and process the real time information

➢ Gateway and The Network Layer

• Robust and High performance network infrastructure

• Supports the communication requirements for latency, bandwidth or security

• Allows multiple organizations to share and use the same network independently

➢ **Management Layer**

• Capturing of periodic sensory data

• Data Analytics (Extracts relevant information from massive amount of raw data)

• Streaming Analytics (Process real time data)

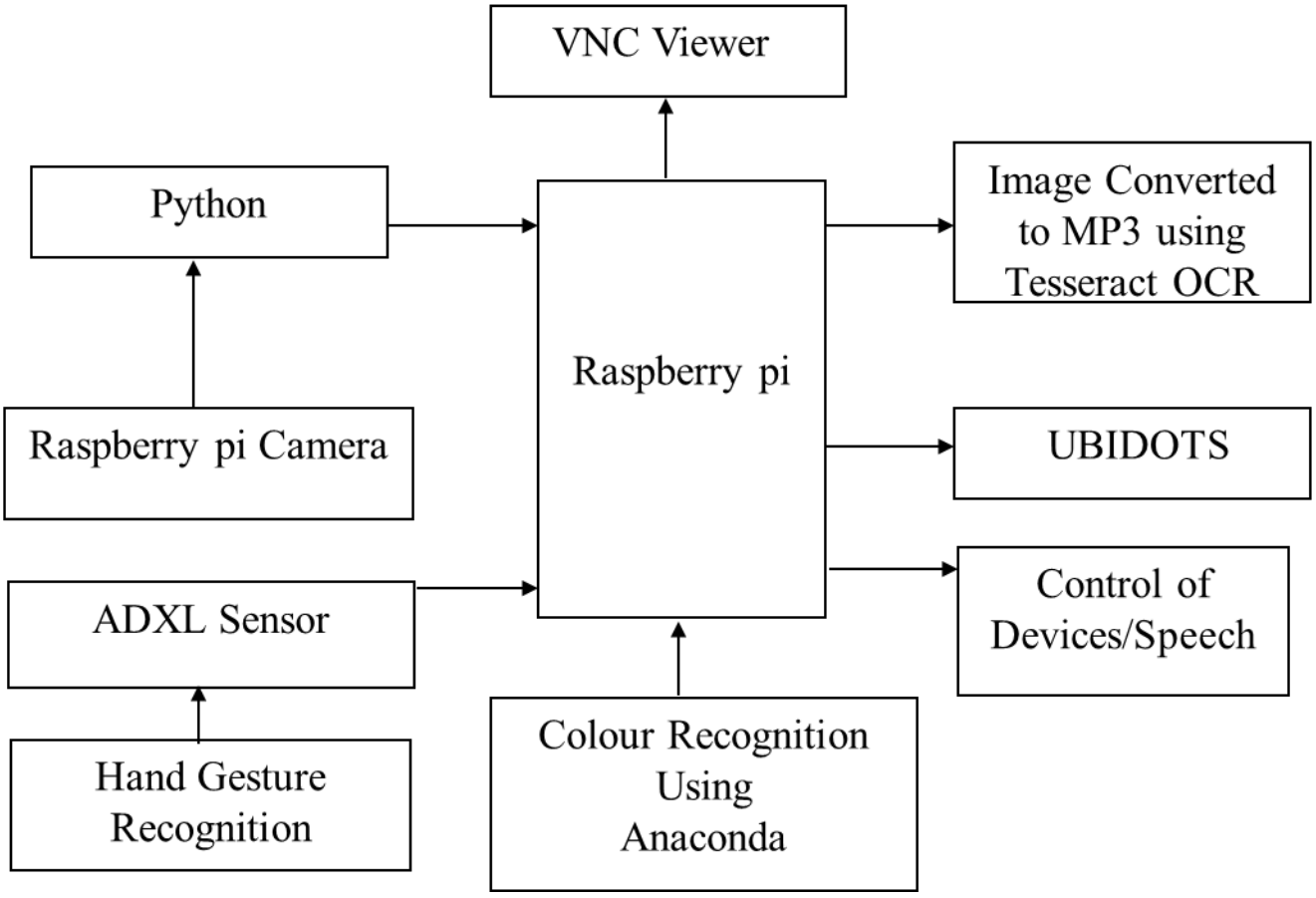
• Ensures security and privacy of data.

➢ **Sensor Layer**

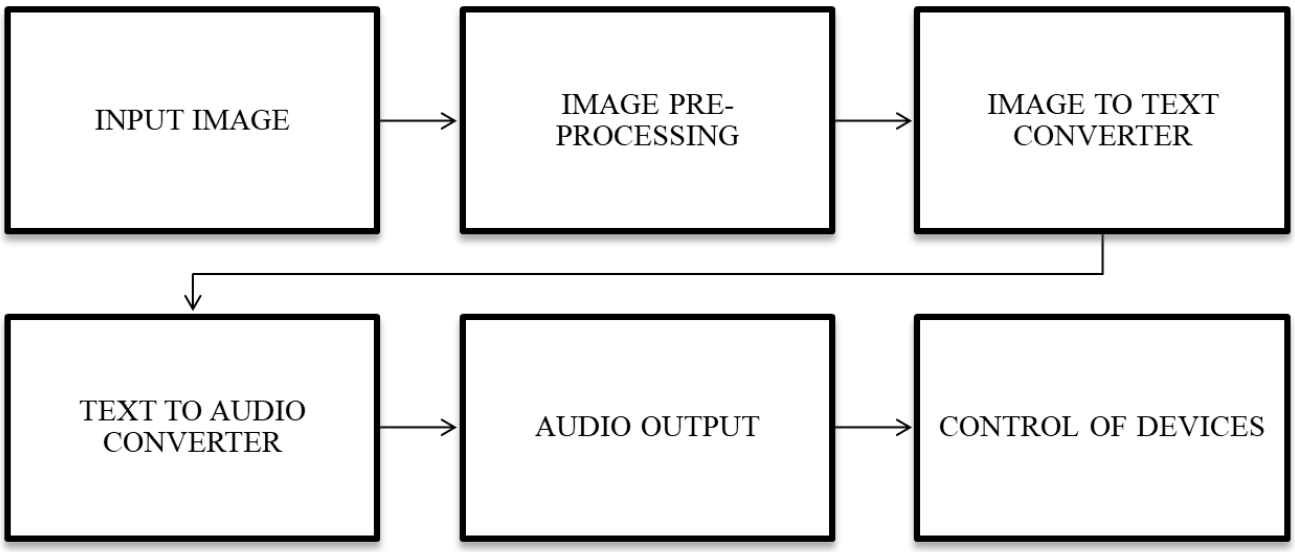
• Provides a user interface for using IoT.

• Different applications for various sectors like Transportation, Healthcare, Agriculture, Supply chains, Government, Retail etc.

SYSTEM ARCHITECTURE



4.6 WORKING



Hand Gesture Recognition: The ADXL sensor recognizes the hand gestures by using its 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface.

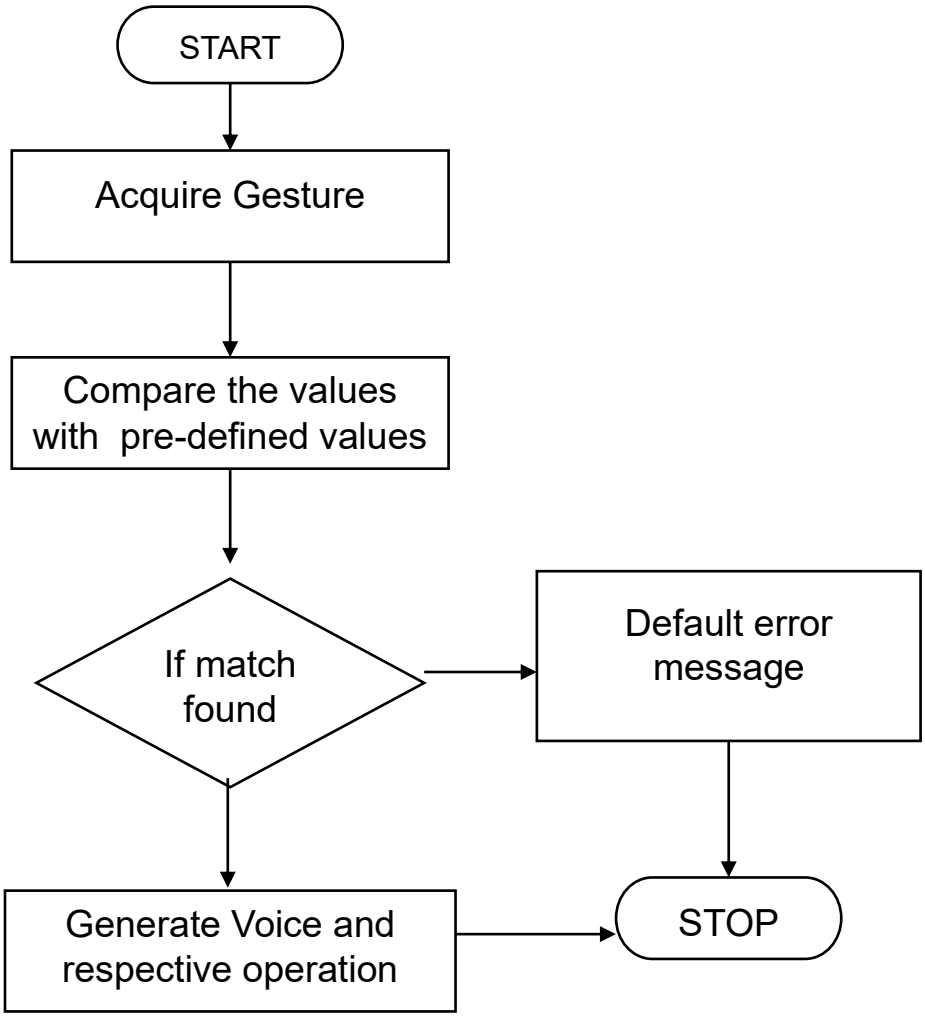
Image acquisition: In this step, the inbuilt camera captures the images of the text or the gestures. The quality of the image captured depends on the camera used. We are using the Raspberry Pi's camera which 5MP camera with a resolution of 2592x1944.

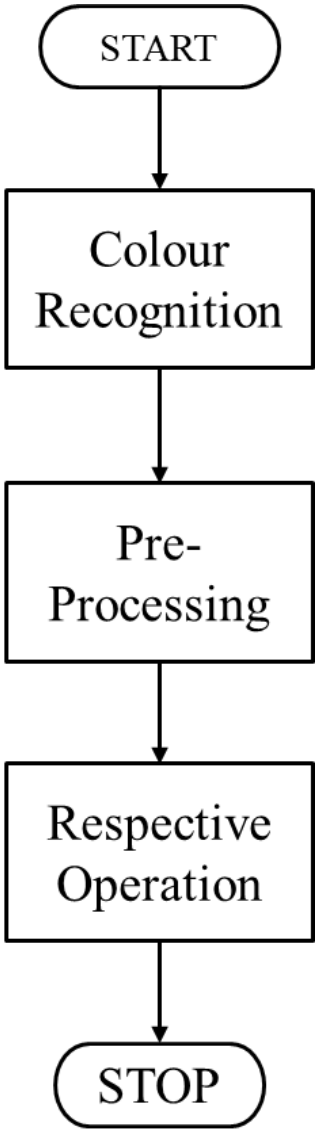
Image pre-processing: This step consists of color to gray scale conversion, edge detection, noise removal, warping and cropping and thresholding. The image is converted to gray scale as many OpenCV functions require the input parameter as a gray scale image. Noise removal is done using bilateral filter. Canny edge detection is performed on the gray scale image for better detection of the contours. The warping and cropping of the image are performed according to the contours. This enables us to detect and extract only that region which contains text and removes the unwanted background. In the end, Thresholding is done so that the image looks like a scanned document. This is done to allow the OCR to efficiently convert the image to text.

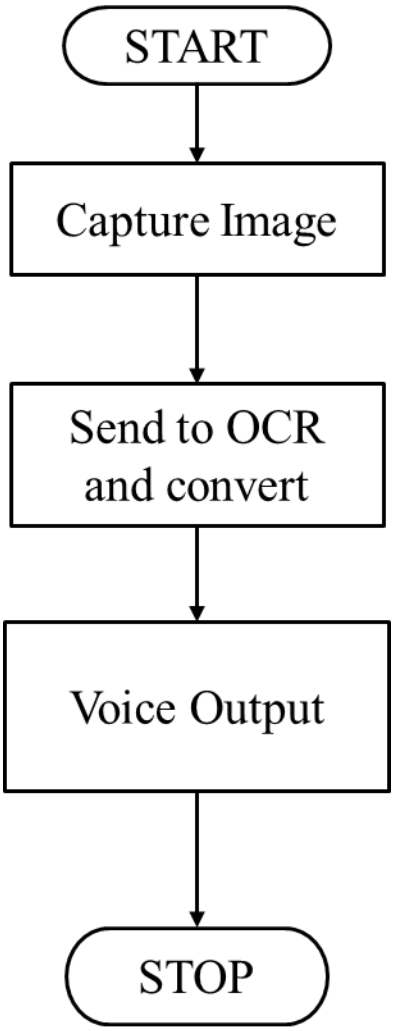
Image to text conversion: The above diagram shows the flow of Text-To-Speech. The first block is the image pre-processing modules and the OCR. It converts the pre-processed image, which is in .png form, to a .txt file. We are using the Tesseract OCR. Text to speech conversion: The second block is the voice processing module. It converts the .txt file to an audio output. Here, the text is converted to speech using a speech synthesizer called Festival TTS. The Raspberry Pi has an on-board audio jack, the on-board audio is generated by a PWM output.

Control of Devices: Basic control of devices such as on/off of light, fan, locking/unlocking of door, play/pause of music and other few automation can be achieved using control from raspberry pi output.

4.6 Flow Chart:



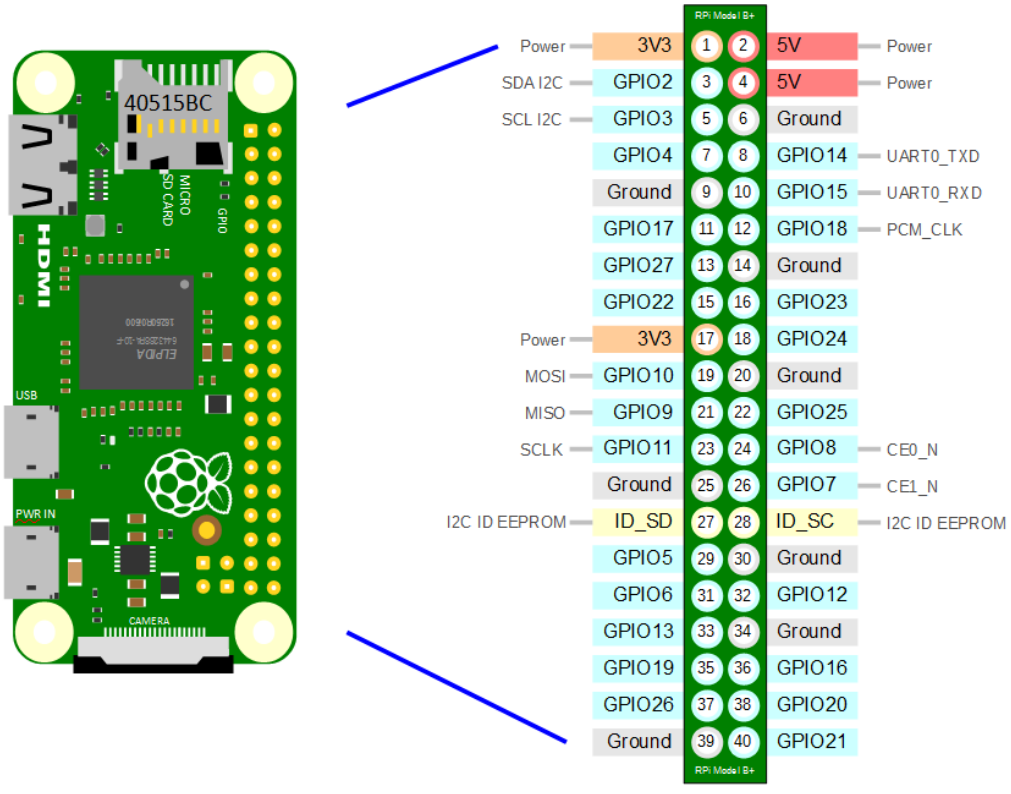

START
Colour Recognition
Pre-Processing
Respective Operation
STOP


START
Capture Image
Send to OCR and convert
Voice Output
STOP

CHAPTER 5

HARDWARE COMPONENTS

5.1 RASPBERRY PI ZERO

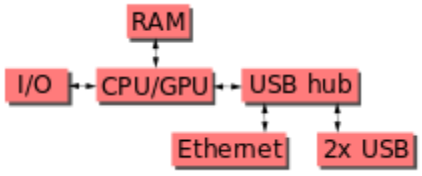


The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It is now widely used even in research projects, such as for weather monitoring because of its low cost and portability. It does not include peripherals (such as keyboards and mice) or cases. However, some accessories have been included in several official and unofficial bundles.

Features

- The Raspberry Pi uses the ARM processor architecture, which is also used by many modern mobile phones
- The Raspberry Pi can use one of the many Linux-based operating systems, instead of Microsoft Windows or Mac OS X like most computers.
- An operating system can be installed using the Raspberry Pi Imager, or by copying operating system 'images' to an SD card
- Add-on boards called 'HATs' (Hardware Attached on Top) can be used to give a Raspberry Pi more features, such as LED lights, extra sensors and Power over Ethernet (PoE).

Hardware of Raspberry Pi



The Raspberry Pi hardware has evolved through several versions that feature variations in the type of the central processing unit, amount of memory capacity, networking support, and peripheral-device support.

This block diagram describes Model B and B+; Model A, A+, and the Pi Zero are similar, but lack the Ethernet and USB 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-port 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. Unlike all other Pi models, the 40 pin GPIO connector is omitted on the Pi Zero with solderable through holes only in the pin locations. The Pi Zero WH remedies this.

Home automation

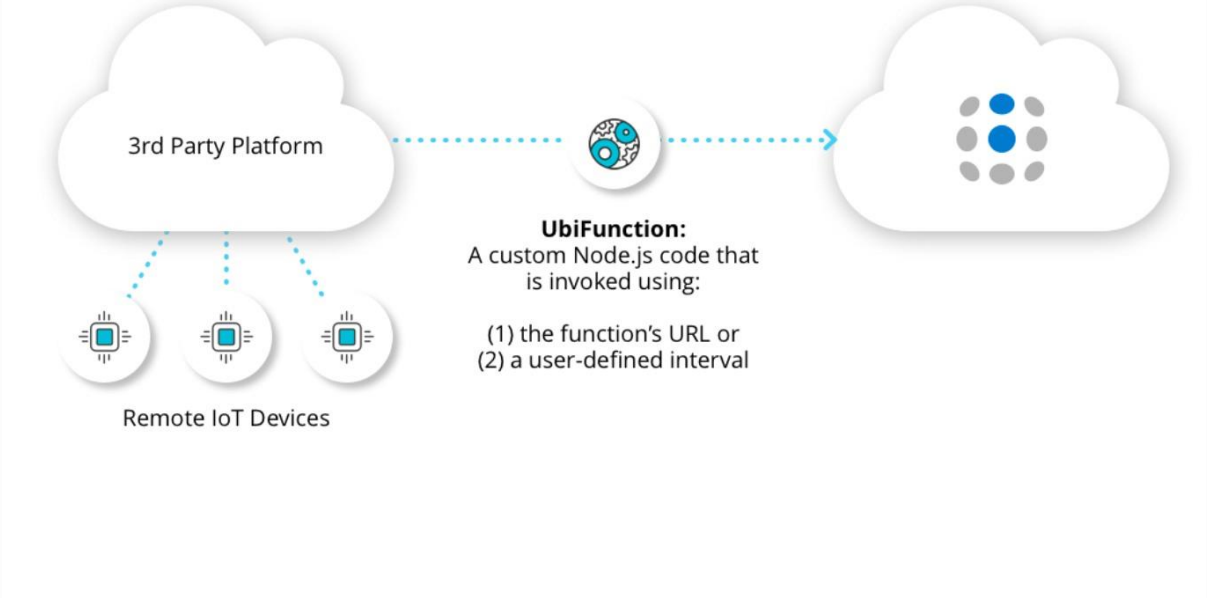
There are a number of developers and applications that are using the Raspberry Pi for home automation. These programmers are making an effort to modify the Raspberry Pi into a cost-affordable solution in energy monitoring and power consumption. Because of the relatively low cost of the Raspberry Pi, this has become a popular and economical alternative to the more expensive commercial solutions.

5.2 UBIDOTS

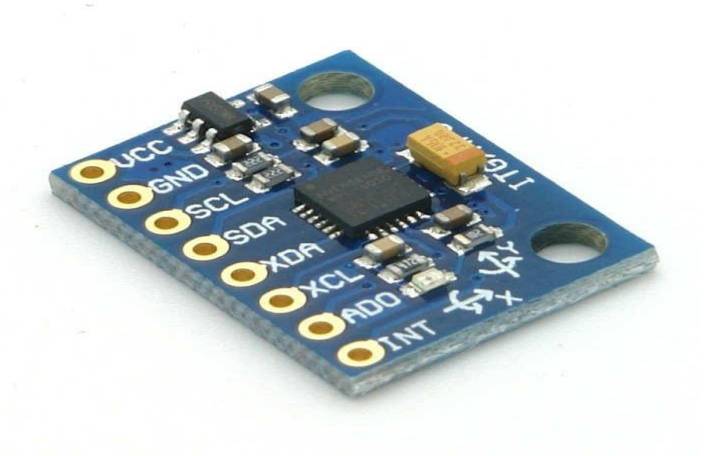
Ubidots is an IoT Platform empowering innovators and industries to prototype and scale IoT projects to production. Use the Ubidots platform to send data to the cloud from any Internet-enabled device. You can then configure actions and alerts based on your real-time data and unlock the value of your data through visual tools. Ubidots offers a REST API that allows you to read and write data to the resources available: data sources, variables, values, events and insights. The API supports both HTTP and HTTPS and an API Key is required.

Your data will be protected with two more replication, encrypted storage and optional TLS/SSL data support. You can also customize permission groups to each module of the platform, making sure the right information is shown to the right user.

Ubidots engineering stack delivers a secure, white-glove experience for our users with device friendly APIs (accessed over HTTP/MQTT/TCP/UDP protocols) that provide a simple and secure connection for sending and retrieving data to/from our IoT data performance optimized time-series backend (cloud). Ubidots application enablement platform supports interactive, real-time data visualization (widgets), and an IoT App Builder that allows developers to extend the platform with their own HTML/JS code for private customization when appropriate. Ubidots exists for businesses and researchers to connect devices efficiently, manage data, and economize an environment.



5.4 ADXL SENSOR



MPU6050 Module

MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package. Also, it has additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers.

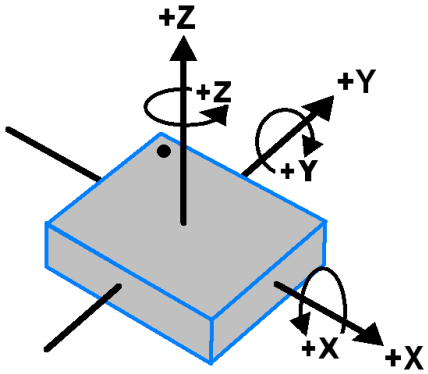
It has Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc.

If 3-axis Magnetometer is connected to auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.

Let's see MPU6050 inside sensors.

3-Axis Gyroscope

The MPU6050 consist of 3-axis Gyroscope with Micro Electro Mechanical System(MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.



MPU-6050
Orientation & Polarity of Rotation

- When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.

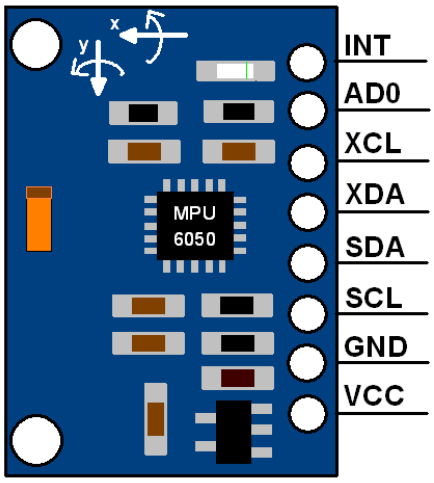
- The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate.

- This voltage is digitized using 16-bit ADC to sample each axis.

- The full-scale range of output are +/- 250, +/- 500, +/- 1000, +/- 2000.

- It measures the angular velocity along each axis in degree per second unit.

MPU-6050 MODULE



The MPU-6050 module has 8 pins,

INT: Interrupt digital output pin.

AD0: I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.

XCL: Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.

XDA: Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.

SCL: Serial Clock pin. Connect this pin to microcontrollers SCL pin.

SDA: Serial Data pin. Connect this pin to microcontrollers SDA pin.

GND: Ground pin. Connect this pin to ground connection.

VCC: Power supply pin. Connect this pin to +5V DC supply.

Calculations

Note that gyroscope and accelerometer sensor data of MPU6050 module consists of 16-bit raw data in 2's complement form.

Temperature sensor data of MPU6050 module consists of 16-bit data (not in 2's complement form).

Now suppose we have selected,

- Accelerometer full scale range of +/- 2g with Sensitivity Scale Factor of 16,384 LSB(Count)/g.

- Gyroscope full scale range of +/- 250 °/s with Sensitivity Scale Factor of 131 LSB (Count)/°/s.

then,

To get sensor raw data, we need to first perform 2's complement on sensor data of Accelerometer and gyroscope.

After getting sensor raw data we can calculate acceleration and angular velocity by dividing sensor raw data with their sensitivity scale factor as follows,

Accelerometer values in g (g force)

Acceleration along the X axis = (Accelerometer X axis raw data/16384) g.

Acceleration along the Y axis = (Accelerometer Y axis raw data/16384) g.

Acceleration along the Z axis = (Accelerometer Z axis raw data/16384) g.

Gyroscope values in °/s (degree per second)

Angular velocity along the X axis = (Gyroscope X axis raw data/131) °/s.

Angular velocity along the Y axis = (Gyroscope Y axis raw data/131) °/s.

Angular velocity along the Z axis = (Gyroscope Z axis raw data/131) °/s.

5.5 RELAY MODULE

In layman terms, a relay is a switch. Technically speaking, a relay is an electromagnetic switch where a small control signal (usually from a microcontroller) at the input of the Relay will control a high voltage supply (usually AC mains).

Since this is a Raspberry Pi based project, let us talk with respect to Raspberry Pi. The Raspberry Pi computer, although a powerful device, works on a 3.3V Logic.

If you want this powerful computer to control your electrical loads, like an LED strip running along your garden or kitchen, you cannot interface them directly as the electrical loads work on AC Mains supply and the Raspberry Pi works on 3.3V DC (technically).

Here comes the Relay. A simple electromechanical device that consists of a coil and few electrical contacts. When the coil is energized, it acts as an electromagnet and closes a switch. If the coil is de-energized, the coil loses its magnetic nature and releases the switch.

So, by controlling the coil, you can control a switch, which in turn will control an electrical load. You can control the coil of the relay with the help of Raspberry Pi (although not directly, but with additional circuitry) as all you need is a small current to energize the coil.

CHAPTER 6

CODE

6.1 PYTHON CODE FOR COLOUR RECOGNITION AND TTS CONVERTION

```
#importing Modules
from ubidots import ApiClient
import cv2
import numpy as np
import time
from PIL import Image
import pytesseract
from gtts import gTTS
import os
from pygame import mixer
pytesseract.pytesseract.tesseract_cmd=r'C:\Program Files\Tesseract-OCR\tesseract'

api=ApiClient(token="BBFF-XRMEqnm38YiLDK9qUiokVIYodThntL")
dat=api.get_variable("5f1be99c1d847276e76092e5")


cap = cv2.VideoCapture(0)

z = 0

x = 0

while(1):

    _, frame = cap.read()

    hsv_frame=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
```

```
low_red=np.array([161,155,84])
high_red=np.array([179,255,255])
red_mask=cv2.inRange(hsv_frame,low_red,high_red)
red=cv2.bitwise_and(frame,frame,mask=red_mask)
a = red.any()
#print(a)
if(a ==True and z == 0):
    a=1
    response=dat.save_value({"value":a})
    time.sleep(1)
    print("FAN ON")
    z = 1
    mixer.init()
    mixer.music.load("music3.mp3")
    mixer.music.play()
    pass

elif(a == True and z == 1):
        a=2
        response=dat.save_value({"value":a})
        time.sleep(1)
```

```
            print("FAN OFF")
            z = 0

            mixer.init()
            mixer.music.load("music4.mp3")
            mixer.music.play()
            pass



    #  low_blue=np.array([110,50,50])
    #  high_blue=np.array([130,255,255])
    #  blue_mask=cv2.inRange(hsv_frame,low_blue,high_blue)
    #  blue=cv2.bitwise_and(frame,frame,mask=blue_mask)
    #  b=blue.any()
    #  print(b)
    #  if(b ==True) :
    #      print("blue")

      low_green=np.array([0,0,200])
      high_green=np.array([180,255,255])
```

```
green_mask=cv2.inRange(hsv_frame,low_green,high_green)
white=cv2.bitwise_and(frame,frame,mask=green_mask)
c=white.any()
# print(c)
if(c == False):
    print("remove the white sheet")
    mixer.init()
    mixer.music.load("music2.mp3")
    mixer.music.play()
    time.sleep(5)
#    camera = cv2.VideoCapture(0)
    for i in range(1):
        return_value,image = cap.read()
        cv2.imwrite('data'+'.png',image)
        image = Image.open('data.png')
        text1 = pytesseract.image_to_string(image)
        print(text1)
        mytext = text1
    var = gTTS(text=mytext,slow=100)
    var.save('music1.mp3')
    os.system("mpg321 music1.mp3")
```

```
        mixer.init()
        mixer.music.load("music1.mp3")
        mixer.music.play()
        print(return_value)
 #      del(camera)
        print("image captured")


 #   cv2.imshow("white",white)
     cv2.imshow("redmask",red)
   # cv2.imshow("whitemask",white)
   # cv2.imshow("greenmask",green)
 #   print("red",red)
#    print("blue",blue)
#    print("green",green)
    time.sleep(2)
    key=cv2.waitKey(1)
    if(key==27):
        break
```

PYTHON CODE FOR GESTURE RECOGNITION

```
import RPi.GPIO as GPIO
import time
from ubidots import ApiClient
GPIO.setmode(GPIO.BOARD)
import smbus                    #import SMBus module of I2C
from time import sleep
api=ApiClient(token="BBFF-XRMEqnm38YiLDK9qUiokVIYodThntL")
dat=api.get_variable("5f1be99c1d847276e76092e5")
rl1=7
rl2=8
rl3=10
rl4=11
GPIO.setup(rl1,GPIO.OUT)
GPIO.setup(rl2,GPIO.OUT)
GPIO.setup(rl3,GPIO.OUT)
GPIO.setup(rl4,GPIO.OUT)
GPIO.output(rl1,1)
GPIO.output(rl2,1)
GPIO.output(rl3,1)
```

```
GPIO.output(rl4,1)


#some MPU6050Registers and their values


PWR_MGMT_1 = 0*6B
SMPLRT_DIV = 0*19
CONFIG     = 0*1A
GYRO_CONFIG= 0*1B
ACCEL_XOUT_H = 0*3B
ACCEL_YOUT_H = 0*3D
ACCEL_ZOUT_H = 0*3F
GYRO_XOUT_H  = 0*43
GYRO_YOUT_H  = 0*45
GYRO_ZOUT_H  = 0*47
def MPU_Init():
        #write to sample rate register
bus.write_byte_data(Device_Address, SMPLRT_DIV, 7)
        #write to power management register
bus.write_byte_data(Device_Address, PWR_MGMT_1, 1)
        #write to configuration register
bus.write_byte_data(Device_Address, CONFIG, 0)
```

```
        #write to Gyro configuration register
bus.write_byte_data(Device_Address, GYRO_CONFIG, 24)
        #write to interrupt enable register
bus.write_byte_data(Device_Address, INT_ENABLE, 1)
def read_raw_data(addr):
        #Accelero and gyro value are 16bit
high=bus.read_byte_data(Device_Address, addr)
low=bus.read_byte_data(Device_Address, addr+1)
        #concatenate higher and lower value
value=((high<<8)|low)
        #to get signed value from mpu6050
if(value>32768):
value=value-65536
return value
bus=smbus.SMBus(1)     #or bus=smbus.SMBus(0) forolder version boards
Device_Address=0*68    #MPU6050 device address
MPU_Init()
print("reading data of gyroscope and accelerometer")
while True:
      #read accelerometer raw value
acc_x = read_raw_data(ACCEL_XOUT_H)
```

```
acc_y = read_raw_data(ACCEL_YOUT_H)
acc_z = read_raw_data(ACCEL_ZOUT_H)
    #read gyroscope raw value
gyro_x = read_raw_data(GYRO_XOUT_H)
gyro_y = read_raw_data(GYRO_YOUT_H)
gyro_z = read_raw_data(GYRO_ZOUT_H)
    #full scale range +/- 250 degree/c as per sensitivity scale factor
Ax = acc_x/16384.0
Ay = acc_y/16384.0
Az = acc_z/16384.0
print "\tx=%.2f g" %Ax, "\ty=%.2f g " %Ay,"\tz=%.2f g" %Az
sleep(3)
if(Ax <= 0):
print("LIGHT OFF")
GPIO.output(rl2,1)
GPIO.OUTPUT(rl3,1)
elif(Ax>0 and Ax<0.98):
print("LIGHT ON")
GPIO.output(rl2,0)
elif(Ax>1):
print("FRIDGE ON")
```

```
GPIO.output(rl3,0)
else:
print("CONSTANT")
data=dat.get_values(1)
z=data[0]['value']
if(z==2):
print("Fan off")
GPIO.output(rl1,1)
elif(z==1):
print("Fan on")
GPIO.output(rl1,0)
```

REFERENCES

- [1] D.Velmurugan, M.S.Sonam, S.Umamaheswari, S.Parthasarathy, K.R.Arun[2016]. A Smart Reader for Visually Impaired People Using Raspberry PI. International Journal of Engineering Science and Computing IJESC Volume 6 Issue No. 3.
- [2] K Nirmala Kumari, Meghana Reddy J [2016]. Image Text to Speech Conversion Using OCR Technique in Raspberry Pi. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 5, Issue 5, May 2016.
- [3] Silvio Ferreira, C´eline Thillou, Bernard Gosselin. From Picture to Speech: An Innovative Application for Embedded Environment. Faculté Polytechnique de Mons, Laboratoire de Théorie des Circuits et Traitement du Signal Bˆatiment Multitel - Initialis, 1, avenue Copernic, 7000, Mons, Belgium.
- [4] Nagaraja L, Nagarjun R S, Nishanth M Anand, Nithin D, Veena S Murthy [2015]. Vision based Text Recognition using Raspberry Pi. International Journal of Computer Applications (0975 – 8887) National Conference on Power Systems & Industrial Automation.
- [5] Poonam S. Shetake, S. A. Patil, P. M. Jadhav [2014] Review of text to speech conversion methods.
- [6] International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-2, Issue-8, Aug.-2014.
- [7] S. Venkateswarlu, D. B. K. Kamesh, J. K. R. Sastry, Radhika Rani [2016] Text to Speech Conversion. Indian Journal of Science and Technology, Vol 9(38), DOI: 10.17485/ijst/2016/v9i38/102967, October 2016.
- [8] World Health Organization. 10 facts about blindness and visual impairment. 2015. Available from: http://www.who. int/features/factfiles/blindness/blindness_facts/en/

- [9] http://elinux.org/RPi_Text_to_Speech_(Speech_Synthesis)
- [10] https://en.wikipedia.org/wiki/Visual_impairment
- [11] https://en.wikipedia.org/wiki/Braille
- [12] https://www.classycyborgs.org/braille-literacy-statistics-india/
- [13] www.raspberrypi.org
- [14] http://www.zdnet.com/article/raspberry-pi-11-reasons-why-its-the-perfect-small-server/ [15]. http://aishack.in/tutorials/opencv/
- [16] http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/p