

SMART TUTOR FOR VISIONLESS PEOPLE USING KINECT SENSOR

**Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering**

By

G.PRANEETH(37130138)

G.V.J ROHITH (37130143)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119
MARCH 2021**



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide Work of **G.PRANEETH(37130138)** and **G.V.J ROHITH (37130143)** who carried out the project entitled **“SMART TUTOR FOR VISIONLESS PEOPLE USING KINECT SENSOR”** under our supervision from September 2020 to April 2021.

Internal Guide

MS.M.DEVIKA

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **G.PRANEETH** (37130138) and **G.V.J ROHITH** (37130143) hereby declare that the Project Report entitled “SMART TUTOR FOR VISIONLESS PEOPLE USING KINECT SENSOR” done by us under the guidance of **MS.M.DEVIKA** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1)

2)

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them. We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D.** Dean, School of Electrical and Electronics Engineering and **Dr. T. RAVI, M.E., Ph.D.** Head of the Department, Department of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **MS.M.DEVIKA** , Department of Electronics and Communication Engineering for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project. We express our gratitude to our parents for their constant encouragement and support for the completion of the project.

Abstract

Obstacle detection and warning can improve the camera still because the safety of visually impaired and deaf people especially in unfamiliar environments. For this, firstly, obstacles are detected and localized then the knowledge of the obstacles are going to be sent to the visually impaired and deaf people by using different modalities like voice, vibration. during this paper, we present an assistive system for visually impaired people based camera. this technique consists of two main components: environment information acquisition and analysis and knowledge representation. the primary component aims at capturing the environment by employing a camera and analyzing it so as to detect the predefined obstacles for visually impaired and deaf people, while the second component when there's an obstacle to the correct of the user; he or she feels strong frequent pulses on the correct side through the haptic strap. For the user, this results in the perception of an obstacle to the correct. The user can then avoid the obstacle by stepping left. After stepping left, the pulses on the proper side stop and also the user perceives that there's now not an obstacle on his or her right side.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO.
	ABSTRACT	V
	LIST OF FIGURE	VII
	LIST OF TABLE	VIII
1.	INTRODUCTION	01
2.	LITERATURE SURVE AND OBJECTIVE	02
3.	AIM AND SCOPE OF THE PROJECT	04
4.	MATERIALS AND METHODS USED	07
	4.1 RASPBERRY PI	07
	4.2 RASPBERRY PI CONFIGURATION	10
	4.2.2 PI'S CAMERA MODULE	16
	4.3 ARCITECHTUER	19
	4.3.2 SD CARD INTEGATION	24
	4.4 THE GPIO PORT	28
	4.5 DIGITAL CAMERA	40
	4.6 ULTRASONIC SENSOR	43
	4.7 LINUX	46
	4.8 PYTHON IDE	54
	4.9 IMAGE PROCESSING BY OPENCV IN PYTHON	57
5.	RESULT AND LIMITATIONS	63
6.	CONCLUSION AND FUTURE WORKS	65
7.	REFERENCES	66

LIST OF FIGURE

CHAPTER NO	TITLE	PAGE NO
3.1	FLOW CHAT	04
3.2	BLOCK DIAGRAM	05
4.1.1	MODEL B+ POWER SUPPLY	09
4.1.2	RASPBERRY PI MODEL B+ GPIO PORT	11
4.2.1	RASPBERRY PI'S SOFTWARE TOOLS	12
4.3.1	RASPBERRY PI'S LAYOUT	19
4.4.3	GPIO PINOUT DIADRAM	37
4.5.1	DIGITAL CAMERA	40
4.6.1	ULRASONIC SENSOR	43
4.6.2	ULTRASONIC TRANSMITTER TRANSMITS	44
4.6.3	ULTRASONIC PIN NAMES	45

LIST OF TABELS

CHAPTER NO	TITLE	PAGE NO
4.4.1	RASPBERRY PI'S GPIO PORT	29
4.4.2	RASPBERRY PI'S PIN CONFIGURATION	32
4.4.6	ULTRASONIC SENSOR PIN CONFIGURATION	45

CHAPTER -1

Introduction

Blindness could be a state of lacking the perception because of medical specialty or physiological factors. The partial cecity represents the dearth of integration within the growth of the optic visual or nerve centre of the attention, and total cecity is that the full absence of the visual lightweight perception. during this work, cheap, a straightforward friendly user, sensible blind system is meant and enforced to boost the quality of each blind and visually impaired folks during a specific space. The planned work includes a wearable instrumentation consists of sunshine weight blind stick and sensing element based mostly obstacle detection circuit is developed to assist the visually handicapped person to navigate alone safely and to avoid any obstacles that will be encountered, whether or not mounted or mobile, to stop any attainable accident. the most objective of this project is to develop AN application for blind folks to sight the objects in numerous directions, detection pits and manholes on the bottom to create liberal to walk detection objects victimization image process are often utilized in multiple industrial yet as social applications. This project is proposing to use object detection for blind folks and provides them audio/ vocal info concerning it. we have a tendency to square measure detection AN object victimization the mobile camera and giving voice directions concerning the direction of an object. This paper presents a wearable navigation helpful system for the blind and therefore the visually impaired engineered with off-the-peg technology. Microsoft Kinect's onboard depth sensing element is employed to extract Red, Green, Blue and Depth A Support Vector Machine classifier is employed to classify scene objects and obstacles to issue crucial time period info to the user through an external aid (earphone) for safe navigation

CHAPTER -2

LITERATURE SURVEY

1.Wei Zeng ,Guoyuan Liang and Can Wang ,Xinyu Wu Multi-Stage Feature Learning Based Object Recognition and 3D Pose Estimation with Kinect , 2016, Impressive achievements have been achieved over the years in object recognition and 3D pose estimation while detecting objects and estimating their 3D poses reliably is still a very challenging problem, especially for texture-less objects in heavily cluttered scenes. In this paper, we introduce an improved approach based on multi-stage feature learning to identify object as well as their 3D orientation efficiently. Our approach is motivated by visual neuroscience models. Currently similar architectures have been successfully applied to pedestrian detection, traffic recognition, face representation, etc.

2.Witsarut Sriratana,2018, Inspection of the external appearance of the pill is one of the methods to check the basic quality according to the standard of Food and Drug Administration (FDA) such as chipped pills, inappropriate sizes, and unclear letter printed on pills. This paper proposes an application of image processing for inspection of pill production process. In this study, the Image Processing principal including the Grayscale Method, Threshold Method, OTSU Method, Bounding Box Method and Geometric Algorithm was adopted to process images from normal webcam camera. In addition, the Microsoft Visual C# 2008 was used for processing the signals by analyzing the output signals of webcam camera.

3.Aashay Sathe ,Dr. Anjali Deshpande,2017, In India biscuit manufacturing plant largely comprises of various machinery for the entire value chain from dough mixing up to baking. However the inspection and rejection continues to be largely manual and based on human vigilance. This presents critical challenges in quality assurance as well as personal health and safety. In this paper, we present how the process of quality inspection and sorting of cookies can be done in an automated way.

4.Yunus Santur,Mehmet Karaköse,Erhan Akin,2017, It is necessary that periodical inspection and maintenance of railway transportation systems, which is becoming more and more common day by day, is required. A typical rail line can range from a few kilometers to thousands of kilometers. It is not possible to control this length of railway line with human labor. For this purpose, rail inspection is done automatically by machine vision today. The input data of machine vision systems consists of data from high-resolution cameras and other sensors. These data are evaluated by machine learning methods and the diagnosis result is produced

OBJECTIVE

- This system is completely different from different system as a result of it'll be used for indoor further as out of doors navigation by police work obstacle by listening instruction through audio signal.
- This system hopes to permit the visually impaired users to easily press a button, speak the required destination and be radio-controlled there with the employment of audio instruction.
- The system checks for 2 forms of obstacles, those who will fully block progress and smaller ones like chairs that may be avoided by ever-changing direction.

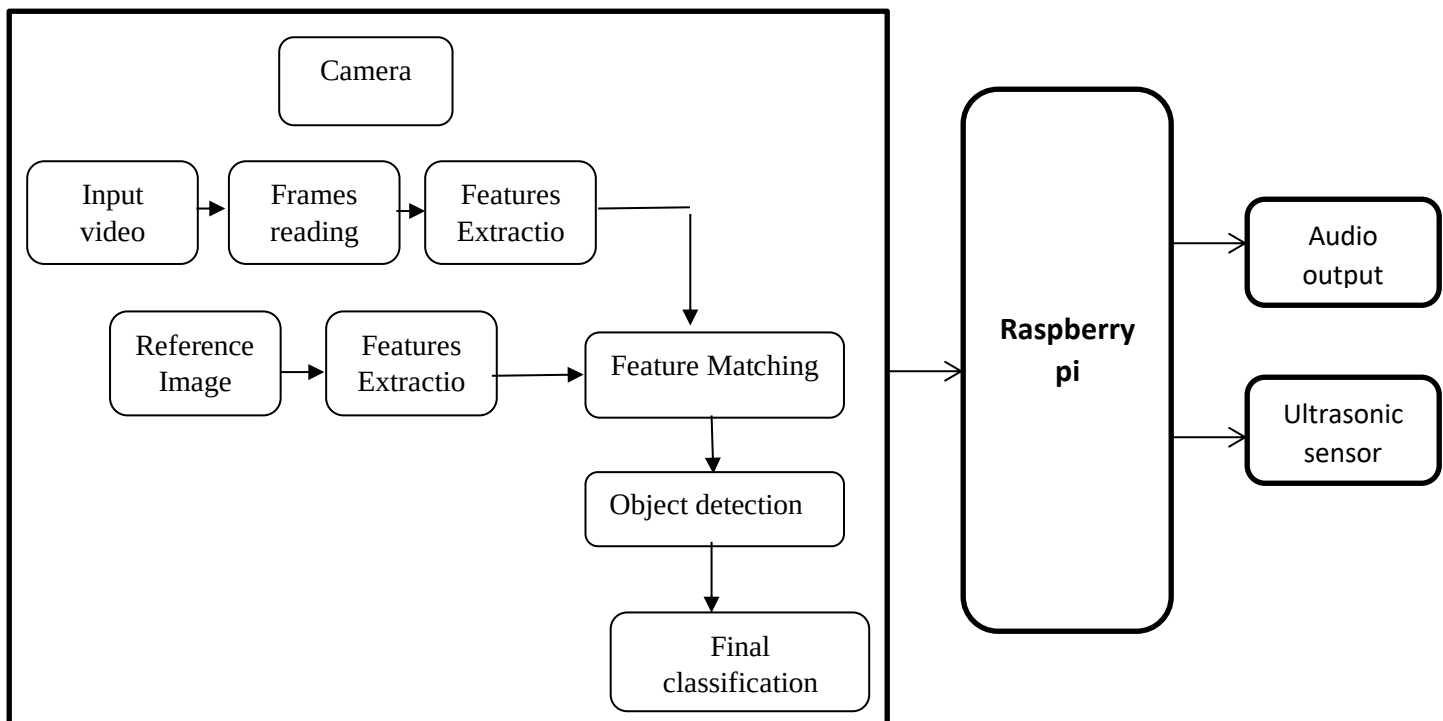
CHAPTER -3

Existing System

The common approach for navigating of visionless person is employing a walking stick cane or walking cane. The walking cane may be a straightforward and computer dedicated to notice static obstacles on the ground, uneven surfaces, and holes via straightforward tactile-force feedback. This device is light-weight, portable but very restricted and it's not usable for the protection from obstacles around head space. Another option that gives the most effective travel aid for the blind is that the guide dogs. The disabled owner and his dog, the coaching and therefore the relationship to the animal square measure the keys to success for this technique. The dog is ready to notice and analyse advanced situations: cross walks, stairs, potential danger, recognize methods and a lot of.

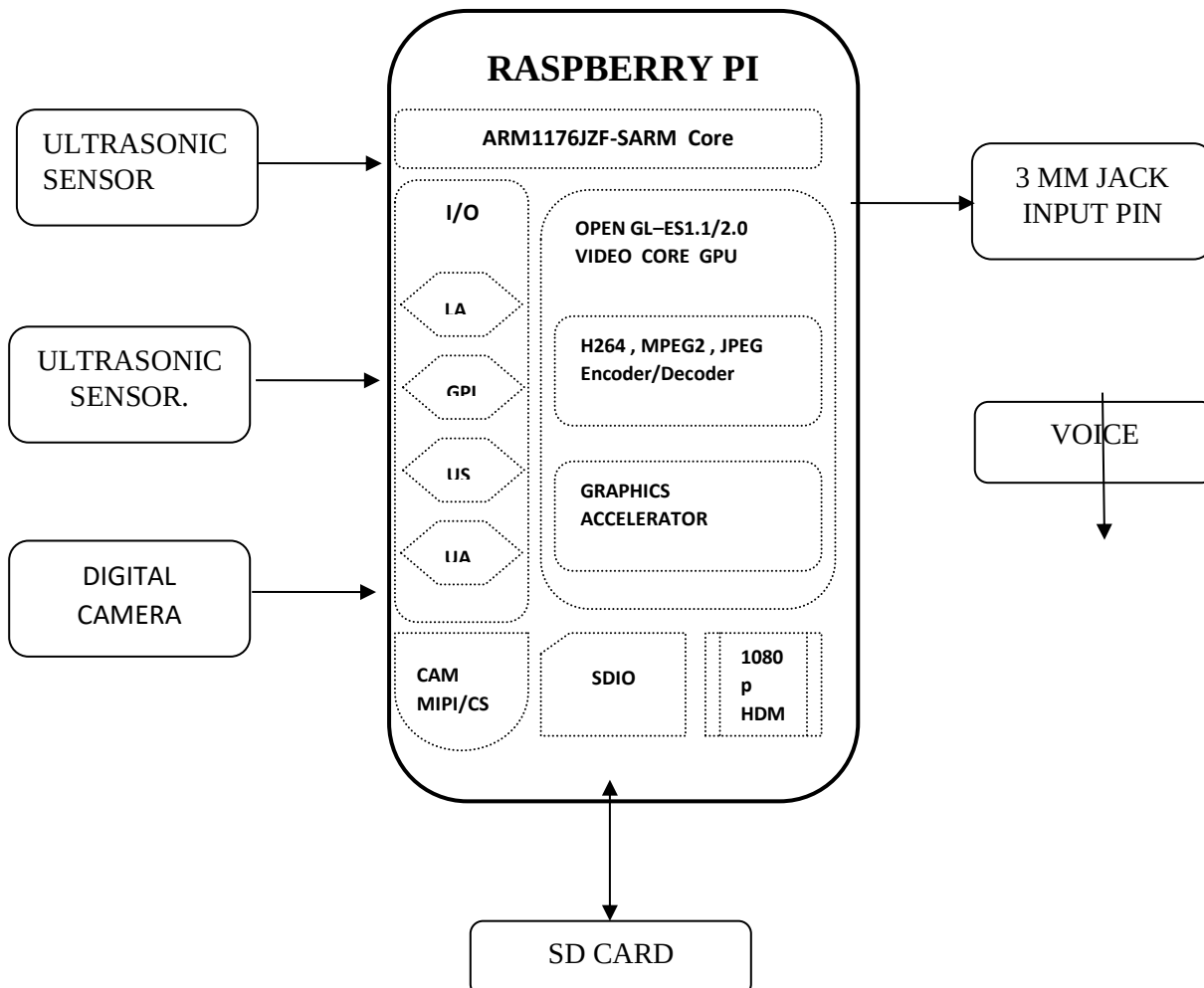
Proposed system

Our planned visual substitution system is predicated on the identification of objects round the blind and handicapped person. we have a tendency to propose a system that acknowledge and find 2D within the video. This system ought to realize the invariant characteristic of objects, offer the popularity and scale back the complexness of detection. we have a tendency to propose a technique supported object extraction and matching in video. A comparison between question frame and info objects is formed to notice object in every frame. for every object detected associate audio file containing the knowledge concerning it's activate.



3.1 Flow chart

The visually handicapped person taking video of the trail where he was walking the appliance will word message to it blind and handicapped person and it'll help thereto person for identifying he's path. the thing gets detected by the key matching technique which is employed within the algorithm. And match that object with the database images to substantiate the obstacle that comes into the way. When object is matched with database objects the appliance gives the voice instruction by using the Speech synthesizer. So, Blind user gets the direction from the applying. They output form the ultrasonic sensor. Shorter distances to obstacles are relayed by increased pulse rates with higher vibration strength. for instance, when there's an obstacle to the correct of the user, he or she feels strong frequent pulses on the proper side through the haptic strap. For the user, this results in the perception of an obstacle to the correct. The user can then avoid the obstacle by stepping left. After stepping left, the pulses on the proper side stop and also the user perceives that there's not an obstacle on his or her right side



3.2 Block diagram

HARDWARE REQUIREMENTS

- Raspberry Pi
- Ultra-Sonic Sensor
- SD card
- Emergency alert button
- Camera

SOFTWARE DESCRIPTION

- Raspberry pi OS: Raspbian stretch
- Programming platform: python 3 IDLE
- Programing language: python 3
- Library: OpenCV

CHAPTER -4

MATERIALS AND METHODS USED

4.1 Raspberry Pi

Raspberry Pi is a small single-board Computer developed in UK by Raspberry Pi foundation to promote the teaching of computer science in schools and in developing countries.

Original model become far more popular than anticipated sealing outside of its target market, for uses such as robots.

History

Raspberry Pi has mainly three generations Raspberry Pi 1, Raspberry Pi 2, Raspberry Pi 3 and also a reduce simple inexpensive Raspberry Pi zero.

- The first model of Raspberry Pi was launched in February 2012 i.e. Raspberry Pi 1 Model B followed by a simple inexpensive Model A.
- In April 2014 “Compute Model” for embedded application Raspberry Pi 1 model B+ improved versions of A and B was launched.
- In November 2015 with reduced I/O and GPIO Raspberry Pi zero came into market.
- In February 2015 advance model with 40 GPIO pins, Ethernet, 4 USB slots Raspberry Pi 2 was launched.
- In February 2016 an upgraded model with inbuilt Bluetooth and Wi-Fi Raspberry Pi 3 Model B was launched.
- Recently in February 2017, “Raspberry Pi – Zero W” with in built Wi-Fi and Bluetooth come into the market.

Features

The heart of the Raspberry Pi is a Broadcom System on Chip (SOC) which includes ARM compatible CPU and on-chip graphic processing unit and VedioCore IV.

The key feature from First generation to the Third generation includes:

- CPU speed ranges from 700 MHz to 1.2 GHz.
- On board Memory (RAM) ranges from 256 MB to 1 GB.
- USB slot differs from 1 slot to USB slots.
- HDMI, composite video output and 3.5mm phone jack.
- Low level output is provided by GPIO pins which support common. protocols like I2C (inter-integrated circuit).
- Ethernet 8 Position 8 Contact (8P8C).

Processor

The processor at the heart of the Raspberry Pi is a Broadcom BCM28XX.

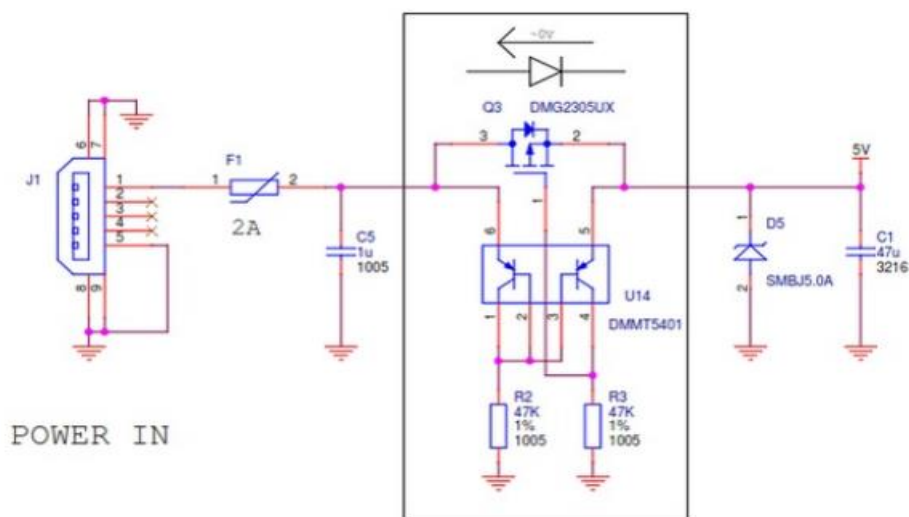
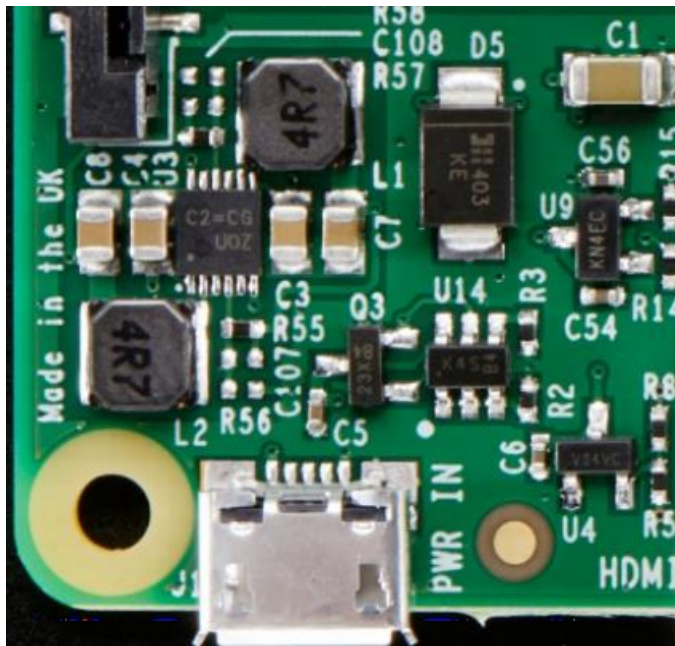
This is the Broadcom System on Chip (SOC) chip use in the Raspberry Pi. The processor from first to third generations include:

- Raspberry Pi 1: Broadcom BCM2835 SOC with 700MHz CPU speed, L2 cache of 128kb with ARM compatibility AR1176JZF-S (ARMv6) 32-bit RISC ARM.
- Raspberry Pi 2: Broadcom BCM 2836 SOC with 900MHz CPU speed, L2 cache of 256kb with 32-bit quad-core ARM cortex-A7 (ARMv7).

- Raspberry Pi 3: Broadcom BCM2837 SOC with 1.2GHz 64-bit quad-core –A53 with 512 kb shared L2 cache (64-bit instruction set ARMv8).

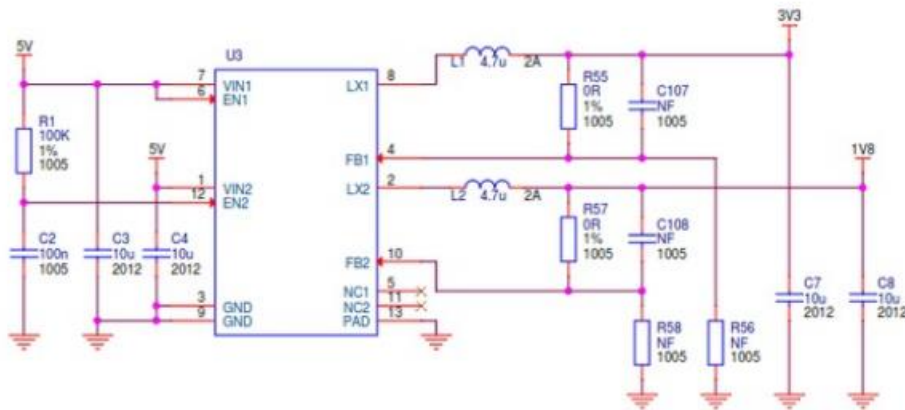
RASPBERRY PI POWER SUPPLY

Model B+ Power Supply To make the B+ more reliable and actually reduce the current draw, the power supply is completely redesigned.

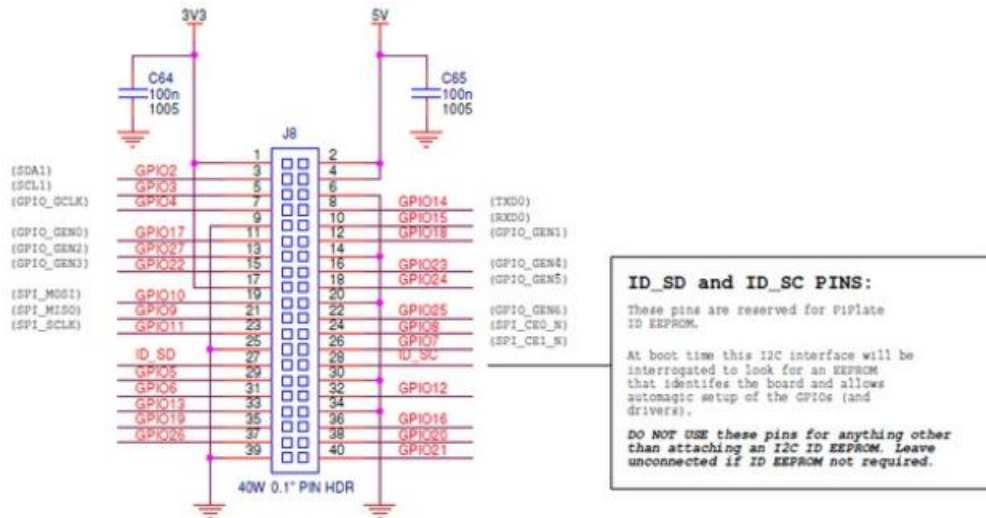


4.1.1 Model B+ Power Supply

There's still the microUSB jack on the left, and the 1A fuse has been upgraded to a 2A fuse. There's also a DMG2305UX (<http://adafru.it/dGU>) P-Channel MOSFET. This acts as a polarity protection switch but is much lower 'drop-out' than a diode. It has only 52mW resistance so @ 2A its about 0.1V voltage drop. Most diodes would be at least 0.5V. Watch this great video about this technique here: To the right is a protection TVS diode (D5 part #SMBJ5) which protects from over-voltages. So not a lot has changed here (other than putting in a protection FET) There is a PNP-matched-pair action going on around the polarity FET, but its 3AM and I'm not 100% sure what it's for so I'll wait till I get some rest before doing any analysis. Let's look at the 3.3V & 1.8V supplies:



Instead of heat-spewing LDO (low dropout) regulators, we now have a dual buck converters. These are high efficiency converters that can take 5V down to 3.3V or 1.8V without as much heat loss. They're more expensive than LDO's but not terribly so! The input to the dual buck is 5V (VIN1 and VIN2) - there's no part number marked here for some reason but it has 12 pins, is a DFN-shaped part (I deal with DFN's all day so I can spot them), and has the marking code C2=CGU0G. with some searching around for a 12-DFN dual buck with 1.8V and 3.3V fixed outputs...



4.1.2 Raspberry Pi Model B+ GPIO Port:

First thing to notice, the top 26 pins of the 40-pin connector are the same as the original. That means that most/many Pi Plates that plug into the Model B will plug into the B+ just fine. They won't sit in the same location - they'll be slid down just a bit but electrically-wise it's the same.

New **GPIOs**

4.2 RASPBERRY PI CONFIGURATION

Setting up Raspberry Pi

As said earlier Raspberry Pi comes without any peripheral devices. The first thing to do is to unpack RasPi and protect it with an enclosure (Figure 3). Raspberry Pi can be installed to the protective enclosure without using any tools. The enclosure has plastic clips which are holding the Raspberry Pi in its place.

After Raspberry Pi has been installed to enclosure and well protected, all the necessary peripherals can be attached to it. Just like any other computer, Raspberry Pi needs some basic devices such as display which is connected via the HDMI cable, the mouse and the keyboard, and the internet connection cable.

Before plugging the power cable, MicroSD-card should be checked if it is flashed and prepared with an operating system. Also it is recommendable to create a backup folder of the MicroSD-card just in case of complications.

The MicroSD-card can be checked with a card-reader. The card-reader can be found from most of the laptops and desktop computers. Insert the MicroSD-card into the card-reader and check that there is something stored in the MicroSD-card. If everything looks good, take the MicroSD-card and plug it into the Raspberry Pi. Now the power cable can be connected.

Raspberry Pi does not have any kind of power switch so it will start up immediately when the power cable is connected to it. At the start up text starts to flow on the monitor and shortly after that there appears a configuration menu. The configuration menu is called Raspi-config (Figure 4). In Raspi-config it is possible to change some of the settings on Raspberry.



4.3.1 RASPBERRY PI SOFTWARE TOOL

The most important settings that should be checked in Raspi-config are:

- ☐ Expand Filesystem, where it is necessary to check that RasPi can use the whole memory capacity of the MicroSD-card. Otherwise the memory can run out fast.
- ☐ Internationalisation Options, where it is possible to choose between different languages and the time zones.
- ☐ Advanced Options, if the internet cable has been plugged in, it is possible to update RasPi to the latest version available. (McManus, S. & Cook, M. 2013, 38.)

It is recommendable that users who do not have so much experience with Linux operating systems should choose the English language because then help and advice can be found more easily from the internet.

It is possible to get back to the Raspi-config and change the settings also after the first setup by typing the following command into the terminal:

`sudo raspi-config`

After making the changes on the Raspberry Pi's settings, the settings can be accepted by choosing the Finish option. Now the terminal view should appear and it might be asking for the username and the password. The username in Raspian Wheezy is by default **pi** and the password should be **raspberrypi**. Notice that these are written in small letters. The Linux is letter case sensitive and it will recognize the difference between small and capital letters.

The next step is logging in to Raspberry and instead of the graphical environment there will be a command console flashing. However, the graphical environment, or so called desktop view, can be started by entering the command:

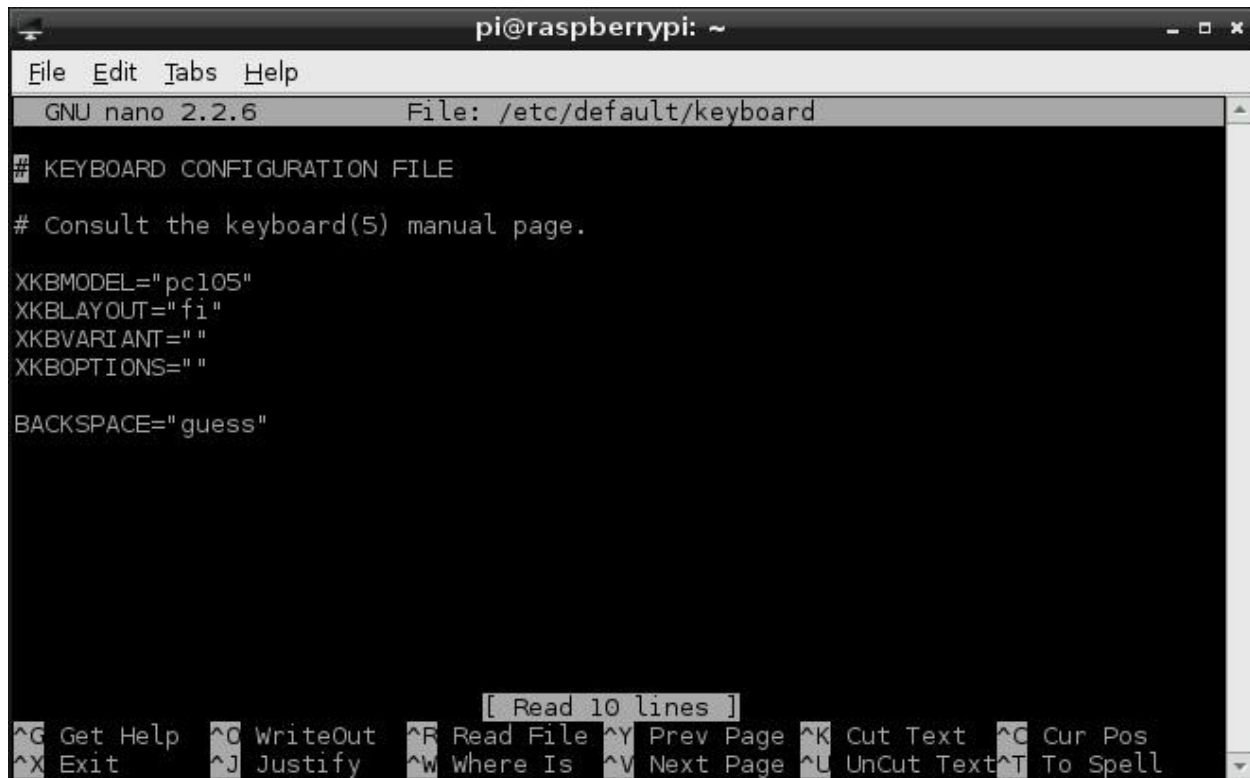
`Startx`

Now Raspberry will be loading for a while and a few seconds later there will appear a more user friendly desktop view. It is recommended to learn how to use the command console as it makes some of the actions faster than doing them in the desktop view.

So far the basic configurations are made for the Raspberry. There might still be some things that are not working correctly. For instance, the keyboard layout might be defined to be in UK style which is the default keyboard layout setting on Raspberry Pi. This can be frustrating and annoying. The layout can be changed easily by opening the LXTerminal which opens the command console. Open the keyboard file in the command console with the nano text editor by typing the following command:

`sudo nano /etc/default/keyboard`

The keyboard configuration file (Figure 5) will appear and it can be modified. The keyboard layout can be changed by replacing the `XKBLAYOUT` value as shown in Figure 5. After the file is edited it can be saved by pressing `CTRL + O` key combination.

A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~'. The terminal shows the GNU nano 2.2.6 text editor editing the file '/etc/default/keyboard'. The content of the file is as follows:

```
KEYBOARD CONFIGURATION FILE

# Consult the keyboard(5) manual page.

XKBMODEL="pc105"
XKBLAYOUT="fi"
XKBVARIANT=""
XKBOPTIONS=""

BACKSPACE="guess"
```

The bottom status bar of the nano editor shows various keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Page, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where Is, ^V Next Page, ^U UnCut Text, and ^T To Spell. A status indicator '[Read 10 lines]' is also visible.

The keyboard configuration, setting the layout to the Finnish “fi”

Controlling the GPIO pins with Python

This chapter discovers the GPIO connector and how it can be used in controlling. The first experiments with the GPIO were to light up a LED (light-emitting diode) through the Python Shell.

The second experiment is little bit more complex and it demonstrates the control-loop of an heating element. The heating element will start to heat the room when room's temperature is getting below the pre-defined lower limit and stops heating when the temperature in the room reaches the second pre-defined upper limit.

Controlling the LED with the GPIO

This is the first experiment with the GPIO connector and it demonstrates how to use it in controlling. This experiment requires a LED and a resistor. The resistor's resistance can be calculated from the Ohm's law which is shown in Formula 1.

Defining the resistance from the Ohm's law:

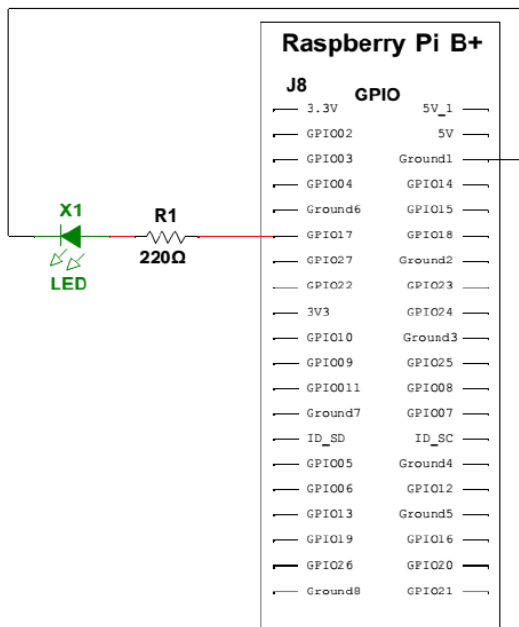
$$U = R * I \quad (1)$$

$$R = \frac{U}{I} = \frac{3.3V_{GPIO \text{ voltage}} - 1.18V_{LED \text{ voltage drop}}}{10mA_{LED \text{ current}}} = 212 \Omega \quad (2)$$

Where

- U is voltage
- R is resistance
- I is current

The resistors above 212 Ω are suitable and can be used for lighting the LED directly from the GPIO. The wiring for the LED and resistor is shown in Figure 11.



After the wirings are done the Python library called python-rpi.gpio needs to be installed. This library allows controlling the GPIO pins. It can be installed with the following command:

```
sudo apt-get install python-rpi.gpio
```

When the installation is finished, open up a Python Shell from the terminal as root user and import the RPi.GPIO library.

```
importRPi.GPIO as GPIO
```

Next thing to do is to set the mode to use the pin numbers from the ribbon cable board and define one of the GPIO pins to be an output. For instance the GPIO 17:

```
GPIO.setmode(GPIO.BCM) # Ribbon cable board
```

```
GPIO.setup(17, GPIO.OUT) # Defines the GPIO17 to be output
```

Now it is possible to control the GPIO17 pin to high and low. The LED will light up when the pin 17 is set to high and when it is set to low the LED will turn off.

```
GPIO.output(17, GPIO.HIGH) # Turns the GPIO17 to high
```

```
GPIO.output(17, GPIO.LOW) # Turns the GPIO17 to low
```

4.2.2 Raspberry Pi's camera module

This chapter is about the Pi NoIR camera module's installation to the Raspberry Pi, and observing the built in functions which are made for it. At the end of this chapter a Python script is created to take resized pictures. The pictures are named with current timestamp and saved to an own directory.

stalling the Pi NoIR camera module

The Raspberry Pi's NoIR camera module board comes in anti-static plastic bag. It is fast and easy to install. The camera module can be mounted to the protective case's cover, where is reserved slot for the camera. (Figure 12) It is screwed with two small screws, and the ribbon cable is connected to the Raspberry Pi's camera connection port. The connection port is located between the 3.5mm audio jack and the HDMI socket. The connection port's clip has to be pulled up before plugging the camera module's ribbon cable on its place After mounting the camera module, it is required to enable the camera module from the Raspi-config configuration tool and then Raspberry Pi has to be rebooted so that the changes will take effect.

Taking the first pictures and videos with the Pi NoIR camera

In Raspbian there are built in functions for the camera module. With these built in functions it is possible to take pictures and record videos, just to try out proper function of the camera module. One of these built in functions or commands is called “raspistill”.

```
raspistill -v -o first_image.jpg
```

After typing the command above into terminal a preview window is started up. The preview window is running for 5 seconds, and then Raspberry takes the picture, and saves it to the file called

first_image.jpg. Parameters -v stands for verbose information during the run and with the -o parameter it is possible to give filename for the output file.

Other simple and useful parameters which can be added into raspistill command are:

- image width **-w**
- image height **-h**
- image quality **-q**
- flip the image vertically **-vf**
- flip the image horizontally **-hf**
- image rotation **-rot**

A complete parameter list can be found from the RaspiCam documentation. (RaspiCam Documentation, 2013, 5-18.)

Creating a Python script for taking pictures

First things to consider before creating the script which takes the picture and stores it automatically are: where the picture is stored, finding the right parameters for the picture so that the image quality and size does not suffer too much.

After a while, some limitations for the pictures are found. The size and quality are reduced to minimize the picture size on the hard drive. The Quality of 75% and the resolution of 1280x720 pixels are sufficient. With these parameters the picture size on the hard drive is around 500KB. That is good starting point, and trade-off between picture quality and available space for picture saving.

All the pictures which are taken by the Python script will be saved to the own folder with current timestamp filename. The folder is located at /var/www/camera/. Apache2 is hosting the folder so that the pictures are available on the website.

Creating the script starts with placing the shebang information and importing the necessary libraries. These libraries are datetime, picamera and time.

```
#!/usr/bin/env/ python
importdatetime
importpicamera
import time
```

On the second step a function called takePicture should be defined. It does not take any input variables. The function consists of three parts. The first part is the general settings, where the location to the saved pictures and the filename are defined.

```
def takePicture():
```

```
    location="/var/www/camera/" #Location to the files
```

```
    date=datetime.datetime.now() #Get current date
```

```
    file_name=date.strftime("%Y-%m-%d %H%M") #Format the string
```

The second part of the function is defining the settings for the picture size and it starts also the preview mode.

```
    #configuration for the pictures
```

```
    camera = picamera.PiCamera()
```

```
    camera.resolution = (1280,720)
```

```
    camera.start_preview()
```

In the last part of the function, the preview mode is kept on for a certain time to warm up the camera. After the warm up time, the function captures the picture and saves it to the predefined location. The picture is named with current timestamp. At the end of the script the preview mode is stopped and the camera is closed.

```
    time.sleep(2) #Camera warm up time
```

```
    # Capture the picture and saved it with the current date
```

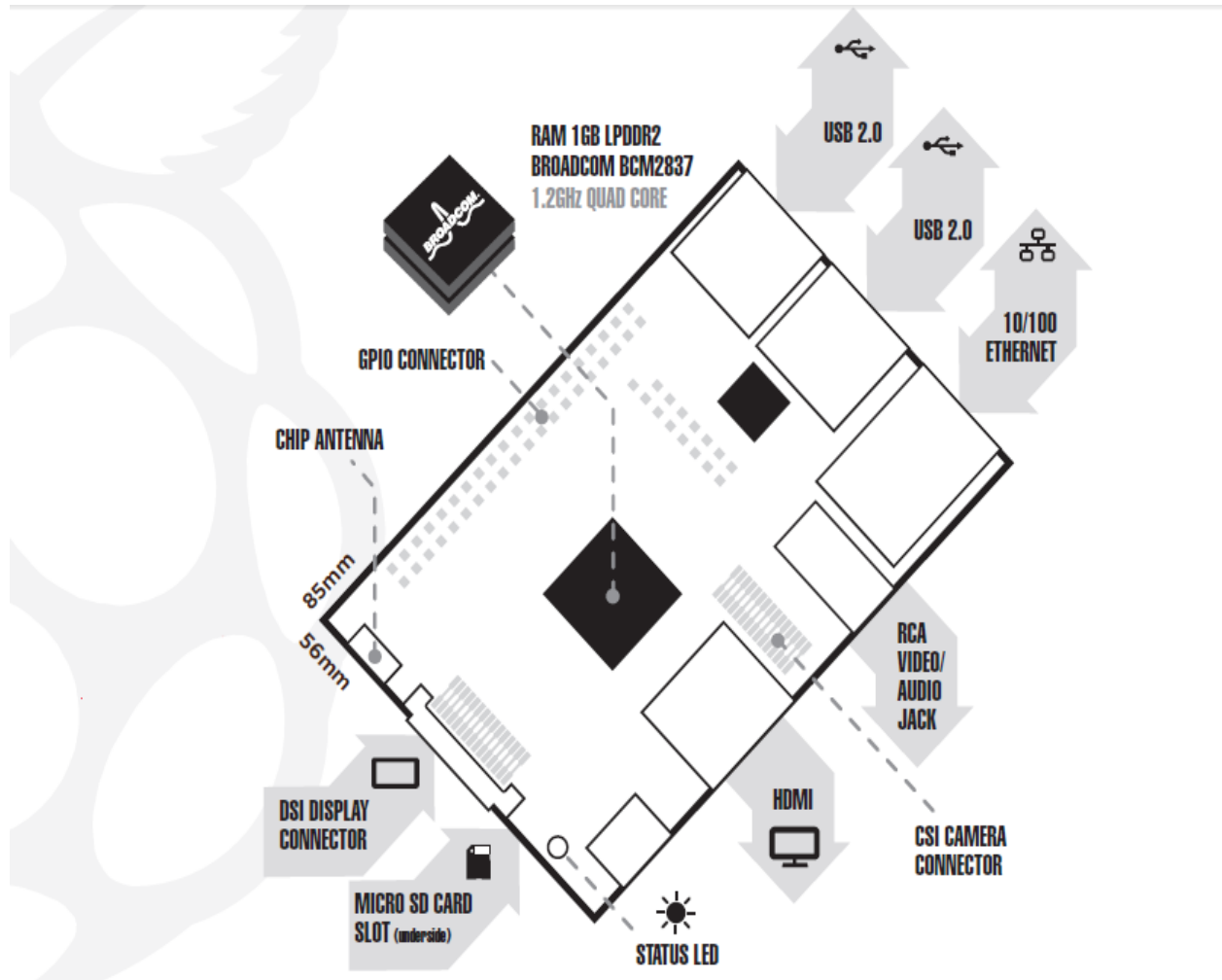
```
    camera.capture("%s%s.jpg" % (location,file_name), quality=75)
```

```
    camera.stop_preview()
```

```
    camera.close()
```

4.3 ARCHITECTURE

The processor at the guts of the Raspberry Pi system may be a Broadcom BCM2837 system-on-chip (SoC) multimedia processor. This suggests that the overwhelming majority of the system's components, including its central and graphics processing units together with the audio and communications hardware, are built onto that single component hidden beneath the 256 MB microchip at the centre of the board.



4.3.1 RASPBERRY PI'S LAYOUT

It's not just this SoC design that produces the BCM2837 different to the processor found in your desktop or laptop, however. It also uses a distinct instruction set architecture (ISA), called ARM. The BCM2837SoC, located beneath a Hynix micro chip Developed by Acorn Computers back within the late 1980s, the ARM architecture may be a relatively uncommon sight within the desktop world. Where it excels, however, is in mobile devices: the phone in your pocket almost

certainly has a minimum of one ARM-based processing core hidden away inside. Its combination of an easy reduced instruction set (RISC) architecture and low power draw make it the proper choice over desktop chips with high power demands and sophisticated instruction set (CISC) architectures. The ARM-based BCM2837 is that the secret of how the Raspberry Pi is ready to work on just the 5V 1A power supply provided by the onboard micro-USB port. It's also the explanation why you won't find any heat-sinks on the device: the chip's low power draw directly translates into little or no waste heat, even during complicated processing tasks. It does, however, mean that the Raspberry Pi isn't compatible with traditional PC software. the bulk of software for desktops and laptops is made with the x86 instruction set architecture in mind, as found in processors from the likes of AMD, Intel and VIA. As a result, it won't run on the ARM-based Raspberry Pi. The BCM2837 uses a generation of ARM's processor design called ARM11, which successively is meant around a version of the instruction set architecture referred to as ARMv6. this can be worth remembering: ARMv6 may be a lightweight and powerful architecture, but includes a rival within the more advanced ARMv7 architecture employed by the ARM Cortex family of processors. Software developed for ARMv7, like software developed for x86, is unfortunately not compatible with the Raspberry Pi's BCM2837—although developers can usually convert the software to create it suitable. That's to not say you're visiting be restricted.

Getting Started with the Raspberry Pi

Now that you have a basic understanding of how the Pi differs from other computing devices, it's time to get started. If you've just received your Pi, take it out of its protective anti-static bag and place it on a flat, non-conductive surface before continuing with this chapter.

Connecting a Display

Before you can start using your Raspberry Pi, you're going to need to connect a display. The Pi supports three different video outputs: composite video, HDMI video and DSI video. Composite video and HDMI video are readily accessible to the end user, as described in this section, while DSI video requires some specialised hardware.

Composite Video

Composite video, available via the yellow-and-silver port at the highest of the Pi referred to as an RCA phono connector is intended for connecting the Raspberry Pi to older display devices. because the name suggests, the connector creates a composite of the colors found within an image—red, green and blue—and sends it down one wire to the display device, typically an old CRT (CRT) TV. The yellow RCA phono connector, for composite video output When no other display device is obtainable, a composite video connection will get you started with the Pi. the standard, however, isn't great. Composite video connections are significantly more vulnerable to interference, lack clarity and run at a limited resolution, meaning that you just can fit fewer icons and features of text on the screen directly. HDMI Video A better-quality picture will be obtained using the HDMI (High Definition Multimedia Interface) connector, the sole port found on the underside of the Pi. Unlike the analogue composite connection, the HDMI port provides a high-speed digital connection for pixel-perfect pictures on both computer monitors and high-definition TV sets. Using the HDMI port, a Pi can display images at the total HD 1920x1080 resolution of hottest HDTV sets. At this resolution, significantly more detail is offered on the screen. If you're hoping to use the Pi with an existing computer monitor, you'll find that your display doesn't have an HDMI input. That's not a disaster: the digital signals present on the HDMI cable map to a typical computer monitor standard called DVI (Digital Video Interconnect). By purchasing an HDMI-to-DVI cable, you'll be able to connect the Pi's HDMI port to a monitor with DVI-D connectivity. If your monitor includes a VGA input—a D-shaped connector with 15 pins, typically coloured silver and blue—the Raspberry Pi can't connect with it. Adapters are available which will soak up a digital DVI signal and convert it to an analogue VGA signal, but these are expensive and hulking. the most effective option here is jus. **While headphones can be connected directly to the Raspberry Pi, you may find the volume a little lacking. If possible, connect a pair of powered speakers instead. The amplifier inside will help boost the signal to a more audible level.**

If you're looking for something more permanent, you can either use standard PC speakers that have a 3.5 mm connector or you can buy some adapter cables. For composite video users, a 3.5 mm to RCA phono cable is useful. This provides the two whiteand- red RCA phono connections that sit alongside the video connection, each carrying a channel of the stereo audio signal to the TV.

For those connecting the Pi to an amplifier or stereo system, you'll either need a 3.5 mm to RCA phono cable or a 3.5 mm to 3.5 mm cable, depending on what spare connections you have on your system. Both cable types are readily and cheaply available at consumer electronics shops, or can be purchased even cheaper at online retailers such as Amazon.

Connecting a Keyboard and Mouse

A webcam could be a video camera that feeds or streams a picture or video in real time to or through a computer to a electronic network, like the web. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams are often used during a video chat session involving two or more people, with conversations that include live audio and video. as an example, Apple's iSight camera, which is made into Apple laptops, iMacs and variety of iPhones, may be used for video chat sessions, using the iChat instant messaging program (now called Messages). Webcam software enables users to record a video or stream the video on the net. As video streaming over the web requires plenty of bandwidth, such streams usually use compressed formats. the utmost resolution of a webcam is additionally less than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions.

4.3.2 SD CARD INTEGRATION

Flashing the SD Card

To prepare a blank SD card for use with the Raspberry Pi, you'll need to flash an operating system onto the card. While this is slightly more complicated than simply dragging and dropping files onto the card, it shouldn't take more than a few minutes to complete.

Firstly, you'll need to decide which Linux distribution you would like to use with your Raspberry Pi. Each has its advantages and disadvantages. Don't worry if you change your mind later and want to try a different version of Linux: an SD card can be flashed again with a new operating system at any point. The most up-to-date list of Linux releases compatible with the Pi is available from the Raspberry Pi website at The Foundation provides BitTorrent links for each distribution. These are small files that can be used with BitTorrent software to download the files from other users. Using these links is an efficient and fast way to distribute large files, and keeps

the Foundation's download servers from becoming overloaded. To use a BitTorrent link, you'll need to have a compatible client installed. If you don't already have a BitTorrent client installed, download one and install it before trying to download the Raspberry Pi Linux distribution. One client for Windows, OS X and Linux is μ Torrent, Which distribution you choose to download is up to you. Instructions in the rest of the book will be based on the Debian Raspberry Pi distribution, a good choice for beginners. Where possible, we'll give you instructions for other distributions as well. Linux distributions for the Raspberry Pi are provided as a single image file, compressed to make it faster to download. Once you've downloaded the Zip archive (a compressed file, which takes less time to download than the uncompressed files would) for your chosen distribution, you'll need to decompress it somewhere on your system. In most operating systems, you can simply double-click the file to open it, and then choose Extract or Unzip to retrieve the contents. After you've decompressed the archive, you'll end up with two separate files. The file ending in sha1 is a hash, which can be used to verify that the download hasn't been corrupted in transit. The file ending in img contains an exact copy of an SD card set up by the distribution's creators in a way that the Raspberry Pi understands. This is the file that needs to be flashed to the SD card.

During the following, you'll be using a software utility called dd. Used incorrectly dd will happily write the image to your main hard drive, erasing your operating system and all your stored data. Make sure you read the instructions in each section thoroughly and note the device address of your SD card carefully. Read twice, write once!

Flashing from Linux

If your current PC is running a variant of Linux already, you can use the dd command to write the contents of the image file out to the SD card. This is a text-interface program operated from the command prompt, known as a terminal in Linux parlance.

Follow these steps to flash the SD card:

1. Open a terminal from your distribution's applications menu.
2. Plug your blank SD card into a card reader connected to the PC.
3. Type `sudo fdisk -l` to see a list of disks. Find the SD card by its size, and note the device address (/dev/sdX, where X is a letter identifying the storage device. Some systems with integrated SD card

readers may use the alternative format `/dev/mmcblkX`—if this is the case, remember to change the target in the following instructions accordingly).

4. Use `cd` to change to the directory with the `.img` file you extracted from the Zip archive.
5. Type `sudo dd if=imagefilename.img of=/dev/sdX bs=2M` to write the file `imagefilename.img` to the SD card connected to the device address from step 3. Replace `imagefilename.img` with the actual name of the file extracted from the Zip archive. This step takes a while, so be patient! During flashing, nothing will be shown on the screen until the process is fully complete.

Figure 1-5: Flashing the SD card using the `dd` command in Linux

Flashing from OS X

If your current PC is a Mac running Apple OS X, you'll be pleased to hear that things are as simple as with Linux. Thanks to a similar ancestry, OS X and Linux both contain the `dd` utility, which you can use to flash the system image to your blank SD card as follows:

1. Select Utilities from the Application menu, and then click on the Terminal application.
2. Plug your blank SD card into a card reader connected to the Mac.
3. Type `diskutil list` to see a list of disks. Find the SD card by its size, and note the device address (`/dev/diskX`, where X is a letter identifying the storage device).
4. If the SD card has been automatically mounted and is displayed on the desktop, type `diskutil unmountdisk /dev/diskX` to unmount it before proceeding.
5. Use `cd` to change to the directory with the `.img` file you extracted from the Zip archive.
6. Type `dd if=imagefilename.img of=/dev/diskX bs=2M` to write the file `imagefilename.img` to the SD card connected to the device address from step 3. Replace `imagefilename.img` with the actual name of the file extracted from the Zip archive. This step takes a while, so be patient!

Connecting External Storage

While the Raspberry Pi uses an SD card for its main storage device—known as a boot device—you may find that you run into space limitations quite quickly. Although large SD cards holding

32 GB, 64 GB or more are available, they are often prohibitively expensive. Thankfully, there are devices that provide an additional hard drive to any computer when connected via a USB cable. Known as USB Mass Storage (UMS) devices, these can be physical hard drives, solid-state drives (SSDs) or even portable pocket-sized flash drives. Two USB Mass Storage devices: a pen drive and an external hard drive. The majority of USB Mass Storage devices can be read by the Pi, whether or not they have existing content. In order for the Pi to be able to access these devices, their drives must be mounted—a process you will learn in Chapter 2, “Linux System Administration”. For now, it’s enough to connect the drives to the Pi in readiness.

Connecting the Network

While the majority of these setup instructions are equally applicable to both the Raspberry Pi Model A and the Model B, networking is a special exception. To keep the component count—and therefore the cost—as low as possible, the Model A doesn’t feature any onboard networking. Thankfully, that doesn’t mean you can’t network the Model A; only that you’ll need some additional equipment to do so.

Wireless Networking

Current Raspberry Pi models don’t feature any form of wireless network capability onboard, but—as with adding wired Ethernet to the Model A—it’s possible to add Wi-Fi support to any Pi using a USB wireless adapter.

Two USB wireless adapters, suitable for use with the Raspberry Pi. Using such a device, the Pi can connect to a wide range of wireless networks, including those running on the latest 802.11n high-speed standard. Before purchasing a USB wireless adapter, check the following:

- Ensure that Linux is listed as a supported operating system. Some wireless adapters are provided with drivers for Windows and OS X only, making them incompatible with the Raspberry Pi. A list of Wi-Fi adapters known to work with the Raspberry Pi.
- Ensure that your Wi-Fi network type is supported by the USB wireless adapter. The network type will be listed in the specifications as a number followed by a letter. If your network type is 802.11a, for example, an 802.11g wireless adapter won’t work.

- Check the frequencies supported by the card. Some wireless network standards, like 802.11a, support more than one frequency. If a USB wireless adapter is designed to work on a 2.4GHz network, it won't connect to a 5GHz network.
- Check the encryption type used by your wireless network. Most modern USB wireless adapters support all forms of encryption, but if you're buying a second-hand or older model, you may find it won't connect to your network. Common encryption types include the outdated WEP and more modern WPA and WPA2. Configuration of the wireless connection is done within Linux, so for now it's enough to simply connect the adapter to the Pi (ideally through a powered USB hub.) You'll learn how to configure the connection in Chapter 4, "Network Configuration".

Connecting Power

The Raspberry Pi is powered by the small micro-USB connector found on the lower left side of the circuit board. This connector is the same as found on the majority of smartphones and some tablet devices. Many chargers designed for smartphones will work with the Raspberry Pi, but not all. The Pi is more power-hungry than most micro-USB devices, and requires up to 700mA in order to operate. Some chargers can only supply up to 500mA, causing intermittent problems in the Pi's operation. Connecting the Pi to the USB port on a desktop or laptop computer is possible, but not recommended. As with smaller chargers, the USB ports on a computer can't provide the power required for the Pi to work properly. Only connect the micro-USB power supply when you are ready to start using the Pi. With no power button on the device, it will start working the instant power is connected and can only be turned off again by physically removing the power cable.

4.4 The GPIO Port

The Raspberry Pi's GPIO port is located on the top-left of the printed circuit board, labelled P1. It's a 54-pin port, fitted with two rows of 13 male 2.54 mm headers at the factory. The spacing of these headers is particularly important: 2.54 mm pin spacing (0.1 inches in imperial measurements) is a very common sight in electronics, and is the standard spacing for prototyping platforms that include stripboards and breadboards. Each pin of the GPIO port has its own purpose, with several pins working together to form particular circuits. The layout of the GPIO port. The Raspberry Pi's GPIO port and its pin definitions Pin numbers for the GPIO port are split into two rows, with the bottom row taking the odd numbers and the top row the even

numbers. It's important to keep this in mind when working with the Pi's GPIO port: most other devices use a different system for numbering pins, and because there are no markings on the Pi itself, it's easy to get confused as to which pin is which.

4.4.1 THE RASPBERRY PI'S GPIO PORT

	Pull	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5
GPIO0	High	SDA0	SA5	<reserved>			
GPIO1	High	SCL0	SA4	<reserved>			
GPIO2	High	SDA1	SA3	<reserved>			
GPIO3	High	SCL1	SA2	<reserved>			
GPIO4	High	GPCLK0	SA1	<reserved>			ARM_TDI
GPIO5	High	GPCLK1	SA0	<reserved>			ARM_TDO
GPIO6	High	GPCLK2	SOE_N / SE	<reserved>			ARM_RTCK
GPIO7	High	SPI0_CE1_N	SWE_N / SRW_N	<reserved>			
GPIO8	High	SPI0_CE0_N	SD0	<reserved>			
GPIO9	Low	SPI0_MISO	SD1	<reserved>			
GPIO10	Low	SPI0_MOSI	SD2	<reserved>			
GPIO11	Low	SPI0_SCLK	SD3	<reserved>			
GPIO12	Low	PWM0	SD4	<reserved>			ARM_TMS

GPIO13	Low	PWM1	SD5	<reserved>			ARM_TCK
GPIO14	Low	TXD0	SD6	<reserved>			TXD1
GPIO15	Low	RXD0	SD7	<reserved>			RXD1
GPIO16	Low	<reserved>	SD8	<reserved>	CTS0	SPI1_CE2_N	CTS1
GPIO17	Low	<reserved>	SD9	<reserved>	RTS0	SPI1_CE1_N	RTS1
GPIO18	Low	PCM_CLK	SD10	<reserved>	BSCSL SDA / MOSI	SPI1_CE0_N	PWM0
GPIO19	Low	PCM_FS	SD11	<reserved>	BSCSL SCL / SCLK	SPI1_MISO	PWM1
GPIO20	Low	PCM_DIN	SD12	<reserved>	BSCSL / MISO	SPI1_MOSI	GPCLK0
GPIO21	Low	PCM_DOUT	SD13	<reserved>	BSCSL / CE_N	SPI1_SCLK	GPCLK1
GPIO22	Low	<reserved>	SD14	<reserved>	SD1_CLK	ARM_TRST	
GPIO23	Low	<reserved>	SD15	<reserved>	SD1_CMD	ARM_RTCK	
GPIO24	Low	<reserved>	SD16	<reserved>	SD1_DAT0	ARM_TDO	

GPIO25	Low	<reserved>	SD17	<reserved>	SD1_DAT1	ARM_TCK	
GPIO26	Low	<reserved>	<reserved>	<reserved>	SD1_DAT2	ARM_TDI	
GPIO27	Low	<reserved>	<reserved>	<reserved>	SD1_DAT3	ARM_TMS	
GPIO28	-	SDA0	SA5	PCM_CLK	<reserved>		
GPIO29	-	SCL0	SA4	PCM_FS	<reserved>		
GPIO30	Low	<reserved>	SA3	PCM_DIN	CTS0		CTS1
GPIO31	Low	<reserved>	SA2	PCM_DOUT	RTS0		RTS1
GPIO32	Low	GPCLK0	SA1	<reserved>	TXD0		TXD1
GPIO33	Low	<reserved>	SA0	<reserved>	RXD0		RXD1
GPIO34	High	GPCLK0	SOE_N / SE	<reserved>	<reserved>		

GPIO41	Low	PWM1	SD5	<reserved>	<reserved>	SPI2_MOSI	RXD1
GPIO42	Low	GPCLK1	SD6	<reserved>	<reserved>	SPI2_SCLK	RTS1
GPIO43	Low	GPCLK2	SD7	<reserved>	<reserved>	SPI2_CE0_N	CTS1
GPIO44	-	GPCLK1	SDA0	SDA1	<reserved>	SPI2_CE1_N	
GPIO45	-	PWM1	SCL0	SCL1	<reserved>	SPI2_CE2_N	
GPIO46	High	<Internal>					
GPIO47	High	<Internal>					
GPIO48	High	<Internal>					
GPIO49	High	<Internal>					
GPIO50	High	<Internal>					
GPIO51	High	<Internal>					
GPIO52	High	<Internal>					
GPIO53	High	<Internal>					

Never connect anything to the pins marked Do Not Connect; these are reserved for internal functions of the Pi's BCM2837 system-on-chip (SoC) hardware. Connecting anything to these will result in damage to the Pi.

4.4.2 RASPBERRY PI'S PIN CONFIGURATION

GPFSEL0:

Bit(s)	Field Name	Description	Type	Reset
31-30	---	Reserved	R	0
29-27	FSEL9	<u>FSEL9 - Function Select 9</u> 000 = GPIO Pin 9 is an input 001 = GPIO Pin 9 is an output 100 = GPIO Pin 9 takes alternate function 0 101 = GPIO Pin 9 takes alternate function 1 110 = GPIO Pin 9 takes alternate function 2 111 = GPIO Pin 9 takes alternate function 3 011 = GPIO Pin 9 takes alternate function 4 010 = GPIO Pin 9 takes alternate function 5	R/W	0
26-24	FSEL8	FSEL8 - Function Select 8	R/W	0
23-21	FSEL7	FSEL7 - Function Select 7	R/W	0
20-18	FSEL6	FSEL6 - Function Select 6	R/W	0
17-15	FSEL5	FSEL5 - Function Select 5	R/W	0
14-12	FSEL4	FSEL4 - Function Select 4	R/W	0
11-9	FSEL3	FSEL3 - Function Select 3	R/W	0
8-6	FSEL2	FSEL2 - Function Select 2	R/W	0
5-3	FSEL1	FSEL1 - Function Select 1	R/W	0
2-0	FSEL0	FSEL0 - Function Select 0	R/W	0

GPFSEL1:

Bit(s)	Field Name	Description	Type	Reset
31-30	---	Reserved	R	0
29-27	FSEL19	<u>FSEL19 - Function Select 19</u> 000 = GPIO Pin 19 is an input 001 = GPIO Pin 19 is an output 100 = GPIO Pin 19 takes alternate function 0 101 = GPIO Pin 19 takes alternate function 1 110 = GPIO Pin 19 takes alternate function 2 111 = GPIO Pin 19 takes alternate function 3 011 = GPIO Pin 19 takes alternate function 4 010 = GPIO Pin 19 takes alternate function 5	R/W	0
26-24	FSEL18	FSEL18 - Function Select 18	R/W	0
23-21	FSEL17	FSEL17 - Function Select 17	R/W	0
20-18	FSEL16	FSEL16 - Function Select 16	R/W	0
17-15	FSEL15	FSEL15 - Function Select 15	R/W	0
14-12	FSEL14	FSEL14 - Function Select 14	R/W	0
11-9	FSEL13	FSEL13 - Function Select 13	R/W	0
8-6	FSEL12	FSEL12 - Function Select 12	R/W	0
5-3	FSEL11	FSEL11 - Function Select 11	R/W	0
2-0	FSEL10	FSEL10 - Function Select 10	R/W	0

Bit(s)	Field Name	Description	Type	Reset
31-30	---	Reserved	R	0
29-27	FSEL29	<u>FSEL29 - Function Select 29</u> 000 = GPIO Pin 29 is an input 001 = GPIO Pin 29 is an output 100 = GPIO Pin 29 takes alternate function 0 101 = GPIO Pin 29 takes alternate function 1 110 = GPIO Pin 29 takes alternate function 2 111 = GPIO Pin 29 takes alternate function 3 011 = GPIO Pin 29 takes alternate function 4 010 = GPIO Pin 29 takes alternate function 5	R/W	0
26-24	FSEL28	FSEL28 - Function Select 28	R/W	0
23-21	FSEL27	FSEL27 - Function Select 27	R/W	0
20-18	FSEL26	FSEL26 - Function Select 26	R/W	0
17-15	FSEL25	FSEL25 - Function Select 25	R/W	0
14-12	FSEL24	FSEL24 - Function Select 24	R/W	0
11-9	FSEL23	FSEL23 - Function Select 23	R/W	0
8-6	FSEL22	FSEL22 - Function Select 22	R/W	0
5-3	FSEL21	FSEL21 - Function Select 21	R/W	0
2-0	FSEL20	FSEL20 - Function Select 20	R/W	0

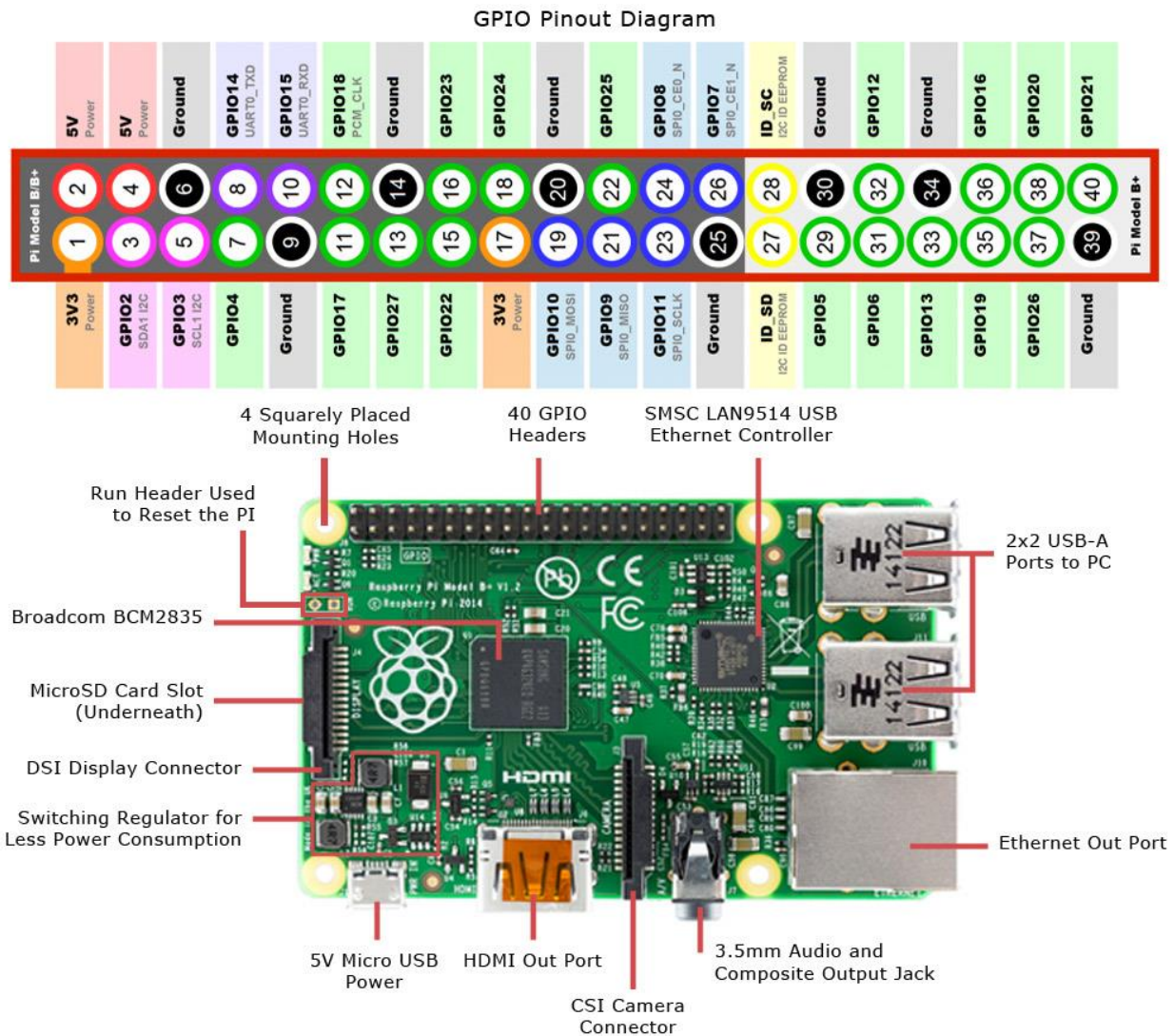
GPIO Alternate function select register 2

Bit(s)	Field Name	Description	Type	Reset
31-30	---	Reserved	R	0
29-27	FSEL39	<u>FSEL39 - Function Select 39</u> 000 = GPIO Pin 39 is an input 001 = GPIO Pin 39 is an output 100 = GPIO Pin 39 takes alternate function 0 101 = GPIO Pin 39 takes alternate function 1 110 = GPIO Pin 39 takes alternate function 2 111 = GPIO Pin 39 takes alternate function 3 011 = GPIO Pin 39 takes alternate function 4 010 = GPIO Pin 39 takes alternate function 5	R/W	0
26-24	FSEL38	FSEL38 - Function Select 38	R/W	0
23-21	FSEL37	FSEL37 - Function Select 37	R/W	0
20-18	FSEL36	FSEL36 - Function Select 36	R/W	0
17-15	FSEL35	FSEL35 - Function Select 35	R/W	0
14-12	FSEL34	FSEL34 - Function Select 34	R/W	0
11-9	FSEL33	FSEL33 - Function Select 33	R/W	0
8-6	FSEL32	FSEL32 - Function Select 32	R/W	0

Bit(s)	Field Name	Description	Type	Reset
31-30	---	Reserved	R	0
29-27	FSEL49	<u>FSEL49 - Function Select 49</u> 000 = GPIO Pin 49 is an input 001 = GPIO Pin 49 is an output 100 = GPIO Pin 49 takes alternate function 0 101 = GPIO Pin 49 takes alternate function 1 110 = GPIO Pin 49 takes alternate function 2 111 = GPIO Pin 49 takes alternate function 3 011 = GPIO Pin 49 takes alternate function 4 010 = GPIO Pin 49 takes alternate function 5	R/W	0
26-24	FSEL48	FSEL48 - Function Select 48	R/W	0
23-21	FSEL47	FSEL47 - Function Select 47	R/W	0
20-18	FSEL46	FSEL46 - Function Select 46	R/W	0
17-15	FSEL45	FSEL45 - Function Select 45	R/W	0
14-12	FSEL44	FSEL44 - Function Select 44	R/W	0
11-9	FSEL43	FSEL43 - Function Select 43	R/W	0
8-6	FSEL42	FSEL42 - Function Select 42	R/W	0
5-3	FSEL41	FSEL41 - Function Select 41	R/W	0
2-0	FSEL40	FSEL40 - Function Select 40	R/W	0

GPIO Alternate function select register 4

Bit(s)	Field Name	Description	Type	Reset
31-12	---	Reserved	R	0
11-9	FSEL53	<u>FSEL53 - Function Select 53</u> 000 = GPIO Pin 53 is an input 001 = GPIO Pin 53 is an output 100 = GPIO Pin 53 takes alternate function 0 101 = GPIO Pin 53 takes alternate function 1 110 = GPIO Pin 53 takes alternate function 2 111 = GPIO Pin 53 takes alternate function 3 011 = GPIO Pin 53 takes alternate function 4 010 = GPIO Pin 53 takes alternate function 5	R/W	0
8-6	FSEL52	FSEL52 - Function Select 52	R/W	0
5-3	FSEL51	FSEL51 - Function Select 51	R/W	0
2-0	FSEL50	FSEL50 - Function Select 50	R/W	0



4.4.3 GPIO PINOUT DIAGRAM

Although the Pi's GPIO port provides a 5 V power supply, tapped from the incoming power on the micro-USB hub, on Pin 2, the Pi's internal workings are based on 3.3 V logic. This means that the components on the Pi work from a 3.3 V power supply. If you're planning on creating a circuit that will interface with the Pi through its GPIO port, make sure you are using components compatible with 3.3 V logic or are passing the circuit through a voltage regulator before it reaches the Pi.

Connecting a 5 V supply to any pin on the Raspberry Pi's GPIO port, or directly shorting either of the power supply pins (Pin 1 and Pin 2) to any other pin will result in damage to the Pi. Because the port is wired directly to pins on the Broadcom BCM2837 SoC processor, you

will not be able to repair any damage you do to it. Always be extra careful when working around the GPIO port.

The GPIO port provides seven pins for general-purpose use by default: Pin 11, Pin 12, Pin 13, Pin 15, Pin 16, Pin 18 and Pin 22. Additionally, Pin 7—while defaulting to providing a clock signal for general purpose use—can also be used as a general purpose pin, giving eight pins in total. These pins can be toggled between two states: high, where they are providing a positive voltage of 3.3 V; and low, where they are equal to ground or 0 V. This equates to the 1 and 0 of binary logic, and can be used to turn other components on or off. You'll learn more about this later in the chapter.

The Pi's internal logic operates at 3.3 V. This is in contrast to many common microcontroller devices, such as the popular Arduino and its variants, which typically operate at 5 V. Devices designed for the Arduino may not work with the Pi unless a level translator or optical isolator is used between the two. Likewise, connecting pins on a 5 V microcontroller directly to the Raspberry Pi's GPIO port will not work and may permanently damage the Pi.

In addition to these general-purpose pins, the GPIO port has pins dedicated to particular buses. These buses are described in the following subsections.

UART Serial Bus

The Universal Asynchronous Receiver/Transmitter (UART) serial bus provides a simple two-wire serial interface. When a serial port is configured in the `cmdline.txt` file (as described in Chapter 6, “Configuring the Raspberry Pi”), it's this serial bus that is used as the port for the messages. Connecting the Pi's UART serial bus to a device capable of displaying the data will reveal messages from the Linux kernel. If you're having trouble getting the Pi to boot, this can be a handy diagnostic tool—especially if nothing is showing on the display. The UART serial bus can be accessed on Pins 8 and 10, with Pin 8 carrying the transmit signal and Pin 10 carrying the receive signal. The speed can be set in the `cmdline.txt` file, and is usually 115,200 bits per second (bps).

I²C Bus

As the name suggests, the Inter-Integrated Circuit (I²C) bus is designed to provide communications between multiple integrated circuits (ICs). In the case of the Pi, one of those integrated circuits is the Broadcom BCM2837 SoC processor at the heart of the system. These pins include access to

pull-up resistors located on the Pi, meaning no external resistors are required to access the I²C functionality. The I²C bus can be accessed on Pins 3 and 5, with Pin 3 providing the Serial Data Line (SDA) signal and Pin 5 providing the Serial Clock (SCL) signal. The I²C bus available on these pins is actually only one of two provided by the BCM2837 chip itself, and is known as I²C0. The second, I²C1, is terminated at resistors on the Raspberry Pi circuit board itself and is not available for general-purpose use.

SPI Bus

The Serial Peripheral Interface (SPI) bus is a synchronous serial bus designed primarily for in-system programming (ISP) of microcontrollers and other devices. Unlike the UART and I²C buses, it's a four-wire bus with multiple Chip Select lines which allow it to communicate with more than one target device. The Pi's SPI bus is available on Pins 19, 21 and 23, with a pair of Chip Select lines on Pin 24 and Pin 26. Pin 19 provides the SPI Master Output, Slave Input (MOSI) signal; Pin 21 provides the SPI Master Input, Slave Output (MISO) signal; Pin 23 provides the Serial Clock (SLCK) used to synchronise communication; and Pins 24 and 26 provide the Chip Select signals for up to two independent slave devices. Although additional buses are present in the Raspberry Pi's BCM2837 SoC processor, they are not brought out to the GPIO port and are thus unavailable for use.

Using the GPIO Port in Python

With the theory out of the way, it's time to get practical. In this section, you'll learn how to install a library to allow easy access to the general-purpose pins on the Raspberry Pi's GPIO port in Python. You'll also be shown two simple electronic circuits which demonstrate how to use the GPIO port for input and output. As you saw in Chapter 11, "Python Basics", Python is a friendly yet powerful programming language. It's not, however, the perfect choice for every scenario. Although it works fine for the simple circuits you'll be creating in this chapter, it does not offer what is known as deterministic real-time operation. For the majority of users, this doesn't matter; if you're planning on using the Pi at the heart of a nuclear reactor or a complex robotics platform, however, you may want to investigate a lower-level language such as C++ or even assembler running on a dedicated real-time microcontroller. If true real-time operation is required for your project, the Pi may be a bad choice. Instead, consider using a microcontroller platform such as the popular open-source Arduino, or one of the MSP430 family of microcontrollers from Texas Instruments. Both of these devices can interface with the Pi either

through the GPIO header or over USB, and provide a specialised real-time environment for control and sensing.

4.5 DIGITAL CAMERA

A webcam could be a video camera that feeds or streams a picture or video in real time to or through a computer to a electronic network, like the web. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams are often used during a video chat session involving two or more people, with conversations that include live audio and video. as an example, Apple's iSight camera, which is made into Apple laptops, iMacs and variety of iPhones, may be used for video chat sessions, using the iChat instant messaging program (now called Messages). Webcam software enables users to record a video or stream the video on the net. As video streaming over the web requires plenty of bandwidth, such streams usually use compressed formats. the utmost resolution of a webcam is additionally less than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions



4.5.1 DIGITAL CAMERA

Optics

Various lenses are available, the most common in consumer-grade webcams being a plastic lens that can be manually moved in and out to focus the camera. Fixed-focus lenses, which have no provision for adjustment, are also available. As a camera system's depth of field is greater for small image formats and is greater for lenses with a large f-number (small aperture), the systems used in webcams have a sufficiently large depth of field that the use of a fixed-focus lens does not impact image sharpness to a great extent.

Most models use simple, focal-free optics (fixed focus, factory-set for the usual distance from the monitor to which it is fastened to the user) or manual focus.

Compression

Digital video streams are represented by huge amounts of data, burdening its transmission (from the image sensor, where the data is continuously created) and storage alike.

Most if not all cheap webcams come with built-in ASIC to do video compression in real-time.

Support electronics read the image from the sensor and transmit it to the host computer. The camera pictured to the right, for example, uses a Sonix SN9C101 to transmit its image over USB. Typically, each frame is transmitted uncompressed in RGB or YUV or compressed as JPEG. Some cameras, such as mobile-phone cameras, use a CMOS sensor with supporting electronics "on die", i.e. the sensor and the support electronics are built on a single silicon chip to save space and manufacturing costs. Most webcams feature built-in microphones to make video calling and videoconferencing more convenient

Interface

Typical interfaces used by articles marketed as a "webcam" are USB, Ethernet and IEEE (denominated as IP camera). Further interfaces such as e.g. Composite video or S-Video are also available

The USB video device class (UVC) specification allows inter-connectivity of webcams to computers without the need for proprietary device drivers.

Software

Various proprietary as well as free and open-source software is available to handle the UVC stream. One could use Gvvcview or GStreamer and GStreamer-based software to handle the UVC stream.

Characteristic:

Webcams are known for their low manufacturing cost and their high flexibility, making them the lowest-cost form of video telephony. As webcams evolved simultaneously with display technologies, USB interface speeds and broadband internet speeds, the resolution went up from gradually 320×240, to 640×480, and some even offering 1280×720 (aka 720p) or 1920×1080 (aka 1080p) resolution.

Despite the low cost, the resolution offered as of 2019 is impressive, with now the low-end webcams offering resolutions of 720p, mid-range webcams offering 1080p resolution, and high-end webcams offering 4K resolution at 60 fps.

Webcams have become a source of security and privacy issues, as some built-in webcams can be remotely activated by spyware. To address this concern, many webcams come with a physical lens cover.

Input control devices

Special software can use the video stream from a webcam to assist or enhance a user's control of applications and games. Video features, including faces, shapes, models and colors can be observed and tracked to produce a corresponding form of control. For example, the position of a single light source can be tracked and used to emulate a mouse pointer, a head-mounted light would enable hands-free computing and would greatly improve computer accessibility. This can be applied to games, providing additional control, improved interactivity and immersiveness.

FreeTrack is a free webcam motion-tracking application for Microsoft Windows that can track a special head-mounted model in up to six degrees of freedom and output data to mouse, keyboard, joystick and FreeTrack-supported games. By removing the IR filter of the webcam, IR LEDs can be used, which has the advantage of being invisible to the naked eye, removing a distraction from the user. TrackIR is a commercial version of this technology.

The EyeToy for the PlayStation 2, PlayStation Eye for the PlayStation 3, and the Xbox Live Vision camera and Kinect motion sensor for the Xbox 360 and are color digital cameras that have been used as control input devices by some games.

Small webcam-based PC games are available as either standalone executables or inside web browser windows using Adobe Flash

4.6 ULTRASONIC SENSOR

Ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception



4.6.1 ULTRASONIC SENSOR

An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.

HC-SR04 Ultrasonic Sensor - Working

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

$$\text{Distance} = \text{Speed} \times \text{Time}$$

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below



4.6.2 ULTRASONIC TRANSMITTER TRANSMITS

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.



4.6.3 ULTRASONIC PIN NAME

Power the Sensor using a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on-board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10µs and then turned off. This action will trigger an ultrasonic wave at frequency of 40kHz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

4.4.6 ULTRASONIC SENSOR PIN CONFIGURATION

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10µs to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: $<15^\circ$
- Operating Current: $<15\text{mA}$
- Operating Frequency: 40Hz

Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

4.7 Linux

Linux is a free open source operating system and it belongs to the Unix operating systems. Actually Linux means the kernel itself which is the heart of the operating system and handles the communication between the user and hardware. Normally Linux is used to refer to the whole Linux distribution. (Upton, E. &Halfacree, G. 2012, 28.)

Linux distribution is a collection of software based on the Linux Kernel. It consists of the GNU-project's components and applications. Because Linux is an open source project, anyone can modify and distribute it. That is the reason why there are many variations of Linux distributions. Most popular distributions are Ubuntu, Red Hat Linux, Debian GNU/Linux and SuSe Linux. (Kuutti, W. &Rantala, A. 2007, 2.)

History of Linux

Linux is a Unix compatible operating system where the operating system's kernel has been reprogrammed. Because of the compatibility most of the free applications programmed for Unix are also available for Linux. In 1973 Unix was reprogrammed in C programming language instead of the assembly code. At that point Unix reached its current outfit. Unix supported multiple users and it was also easy to transfer to new digital machines. Later Unix was given to Universities for further development. (Kuutti, W. &Rantala, A. 2007, 5-6.)

Linux got started in the early 1990s when Linus Torvalds got tired of MS-DOS operating system and decided to create a new operating system for the Intel's cheap x86 processors. At the time there was already available Minix operating system for microcomputers. However, mainly for teaching purposes created Minix was not good enough for Torvalds. (Kuutti, W. &Rantala, A. 2007, 5-6.) In October 1991 Linus Torvalds released the first unofficial Linux and the first official Linux version was released in March 1994. Nowadays Linus Torvalds is still partly developing and supporting the kernel's further development. (Kuutti, W. &Rantala, A. 2007, 5-6.)

Linux compared to Windows

When comparing Linux and Windows as operating systems, one of the major differences are that Linux is an open-source project and Windows is a closed-source project. In the closed-

source project the users sees only the finished product but do not know how it has been done. In open-source projects everything is made fully visible to the public. (Upton, E. &Halfacree, G. 2012, 13-14.)

In practice this can be seen in Linux's easy customization for different platforms. This process is called porting. There are several distributions ported to the Rasp-berry Pi's BCM2835 chip. One of the distributions is called Raspbian Wheezy. (Up-ton, E. &Halfacree, G. 2012, 14.)

Raspbian Wheezy

Raspbian Wheezy is a free operating system based on Debian distribution. It is created by a small team of developers who are fans of Raspberry Pi. Raspbian is optimized for the Raspberry Pi's hardware and it comes with over 35 000 packag-es and pre-compiled software. Raspbian is still under active development and it aims to improve the stability and performance of the Debian packages. (Raspbian [Ref. 15.2.2015])

Raspbian is officially recommended for beginners and it includes the graphical desktop environment called LXDE. Raspbian Wheezy is one of the fastest ways to setup and get the RasPi running. (McManus, S. & Cook, M. 2013, 20.)

Programming languages

There are considerable numbers of programming languages which have been adapted for Raspberry Pi. Python programming language is recommended by The Raspberry Pi foundation especially for the beginners. Basically any programming language which can be compiled for ARMv6 can run on the Raspberry Pi. There-fore the users are not restricted to use only the Python. On the Raspberry Pi there are preinstalled several languages for example C, C++, Java, Scratch and Ruby. (Raspberry Pi Foundation [Ref. 5.2.2015].)

Python programming language

Python programming language is developed in the late 1980s at the National Re-search Institute by Guido van Rossum. Python has grown in popularity, and it is widely used commercially. (Upton, E. &Halfacree, G. 2012, 152.)

Python is a flexible and powerful programming language but still it is easy to learn and follow. The clear syntax of Python makes it a valuable tool for users who wants to learn programming. This is one of the reasons why it is recommended by the Raspberry Pi Foundation. Python is

published under an open-source license and it is available for different operating systems. Python runs on Linux, OS X and Windows computer systems. (Upton, E. &Halfacree, G. 2012, 152.) Cross-platform support guarantees that the programs which are written in Python are also compatible in other platforms. There are few exceptions where the pro-grams are not compatible. For instance, when the Python is addressed to use the specific hardware such like Raspberry Pi's GPIO.(Upton, E. &Halfacree, G. 2012, 152.)

Installing OS to Raspberry pi B+:

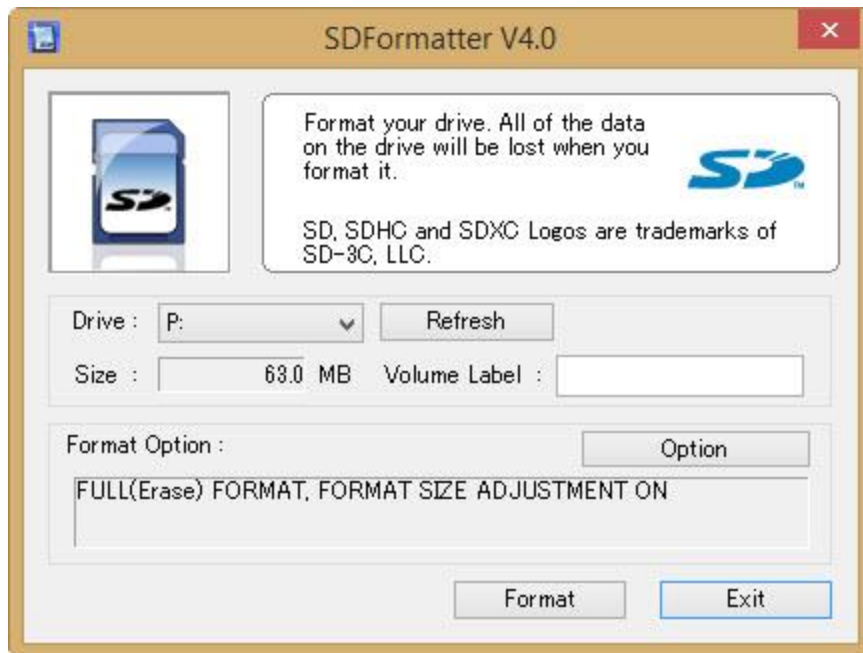
Raspberry Pi doesn't come with an operating system. This is not a weakness, however, rather a feature that means you can choose from a wide of OSs, each of which can be flashed to an SD card (or microSD card for the Raspberry Pi B+) in a few simple steps. Here's how to get a new OS installed and running on your Pi – and how to clone your perfect setup for quick disaster recovery.

Operating systems such as the recommended Raspbian, ArchLinux, Risc OS and even Android come ready to run on your Raspberry Pi. I'll show you the two main ways add an operating system – and once you've got your Pi setup how you want it, we'll look at how to clone the card so that it can be restored following errors (or for temporary reuse of your SD card).

The following tutorials assume that you have a basic Raspberry Pi package and Windows to manage your SD card writing and cloning.

Flash An OS To SD And Boot YourRaspberry Pi

Whichever operating system you download for your Raspberry Pi, the process of writing it to an SD card is the same. However, there are some differences in SD card writing between desktop operating systems. You'll also need to ensure that your card is blank and formatted, and at least 2 GB.



For Windows users (and also those of you with Mac OS X) the tool of choice for writing a Raspberry Pi OS image to SD card is SD Formatter, from the SD Association. With the card inserted into your computer's card reader, and ensuring you have the correct **Drive** letter selected in the drop down menu, open the **Option** menu and select **Full (Erase)** and **On**. This ensures that the full capacity of the storage card will be available.

Click **OK**, then **Format** to begin.

To write the disk image, use Win32DiskImager, available from Sourceforge. You may need to run with administrator privileges. Select the correct drive letter for your SD card, browse to the image file and click **Write** to commence the process. Win32DiskImager will inform you when the data has been written.

If writing the disk image seems too much hassle or is beyond your abilities, it is possible to purchase SD cards with Raspbian pre-installed.

Booting Raspbian For The First Time

With Raspbian installed, you'll need to login with the following credentials:

Username: **pi**

Password: **raspberry**

For other operating systems, check the documentation to find the default login credentials.

Remember that the password will not be displayed as you type it; there are no Windows-style * symbols representing the letters. Instead, it will appear that you haven't entered a password. This is a security feature in Linux to prevent people guessing the length of your passphrase. Just type the password regardless.

In Raspbian, you can change your password by running

```
sudo raspi-config
```

And selecting the Change Password option.

Install Raspbian, XBMC, Risc OS, OpenElec&ArchLinux Easily With NOOBS

For a simpler, more streamlined install of the more popular Raspberry Pi distros such as Raspbian, ArchLinux, XBMC and OpenElec media centres, as well as Risc OS and Pidora – you can employ NOOBS, the New Out Of Box Software installation system.

Perfecting Your Raspberry Pi OS

Different Raspberry Pi projects require a varying amount of software; it really all depends on what you're planning to do.

The SD card contains the Raspberry Pi's operating system (the OS is the software that makes it work, like Windows on a PC or OSX on a Mac). This is very different from most computers and it is what many people find the most daunting part of setting up their Raspberry Pi. It is actually very straightforward—just different!

The following instructions are for Windows users. Linux and Mac users can find instructions at www.raspberrypi.org/downloads

1. Download the Raspberry Pi operating system: The recommended OS is called Raspbian. Download it here: <http://downloads.raspberrypi.org/images/raspbian/2012-12-16-wheezy-raspbian/2012-12-16-wheezy-raspbian.zip>

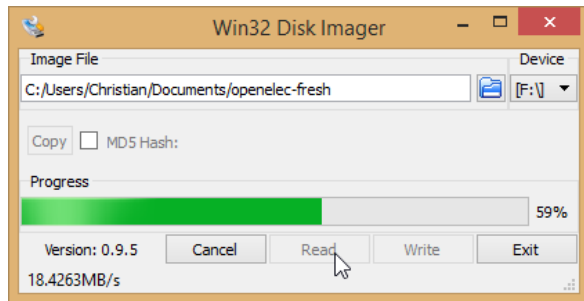
2. Unzip the file that you just downloaded:

a) Right click on the file and choose “Extract all”.

b) Follow the instructions—you will end up with a file ending in .img This .img file can only be written to your SD card by special disk imaging software, so...

3. Download the Win32DiskImager software: a) Download win32diskimager-binary.zip (currently version 0.6) from: <https://launchpad.net/win32-image-writer/+download> b) Unzip it in the same way you did the Raspbian .zip file c) You now have a new folder called win32diskimager-binary You are now ready to write the Raspbian image to your SD card.

4. Writing Raspbian to the SD card a) Plug your SD card into your PC b) In the folder you made in step 3(b), run the file named Win32DiskImager.exe (in Windows Vista, 7 and 8 we recommend that you right-click this file and choose “Run as administrator”). You will see something like this:



c) If the SD card (Device) you are using isn't found automatically then click on the drop down box and select it

d) In the Image File box, choose the Raspbian .img file that you downloaded

e) Click Write

f) After a few minutes you will have an SD card that you can use in your Raspberry Pi 5. Booting your Raspberry Pi for the first time a) Follow the Quick start guide on page 1 b) On first boot you will come to the Raspi-config window c) Change settings such as timezone and locale if you want d) Finally, select the second choice: expand_rootfs and say 'yes' to a reboot e) The Raspberry Pi will reboot and you will see raspberrypi login: f) Type: pi g) You will be asked for your Password h) Type: raspberry i) You will then see the prompt: pi@raspberrypi ~ \$ j) Start the desktop by typing:

With this done, you will then need to install various updates from within the media centre software (much as you would if you had installed Raspbian). There may be other applications that you need, add-ins for the media centre such as Vimeo or YouTube, perhaps the TED Talk channel, etc.

As with any computer system, installing the operating system is only the first step. Updates, optional software and other tools will also need to be added before you're happy that the Raspberry

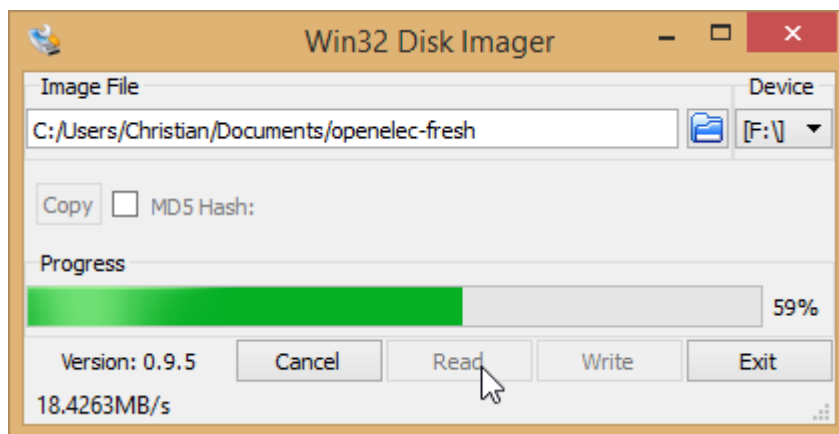
Pi is ready for its intended use. The same process would be followed if you were configuring your Pi as a wireless print server, perhaps, or even webserver.

Once this is done, you should have your Raspberry Pi installation perfected.

Disaster Management: Cloning The OS

The point at which you're happy with the installed OS is precisely when you should make a clone of it. This is a full backup image of the SD card saved to your hard disk drive (or second SD card) that can be easily flashed should you run into trouble.

Windows users can again employ Win32DiskImager here, ensuring that the perfected Raspberry Pi SD card is inserted in your Windows card reader before launching the utility.



Find the **Image File** field, and browse to the folder where you will be saving your SD card image; in the **Device** box, select the drive letter of the SD card, and after confirming both are correct, click **Read** to begin the cloning, following any on-screen instructions. Give the process time to complete before proceeding.

Should you need to revert the SD card back to its “perfect” state later on, you’ll need to remove all partitions (WIN+R to launch Disk Management, right-click the SD card and select **Delete**) before restoring the image. You might also reformat using SD Formatter, as explained above.

Again using Win32DiskImager, browse for the image file, set the **Device** drive letter and then click **Write**. Once the process is complete, your SD card will be reverted to the “perfect” configuration you worked on earlier, and your Raspberry Pi should be ready to put to use once more.

4.8 Python IDE

An Introduction to Python

The Raspberry Pi gets the first half of its name from a long-standing tradition of using fruit to name new computing systems—from classic microcomputers like the Acorn, Apricot and Tangerine to more recognizably modern brands including Apple and BlackBerry—but the second half comes courtesy of the Python programming language.

Introducing Python

Flexible and powerful, Python was originally developed within the late 1980s at the National Research Institute for Mathematics and Computer Science by Guido van Rossum as a successor to the ABC language. Since its introduction, Python has grown in popularity due to what's seen as a transparent and expressive syntax developed with a spotlight on ensuring that code is readable. Python may be a problem-oriented language. This suggests that Python code is written in largely recognizable English, providing the Pi with commands during a manner that's quick to find out and simple to follow. This can be in marked contrast to low-level languages, like assembler, which are closer to how the pc “thinks” but almost impossible for somebody's to follow without experience. The high-level nature and clear syntax of Python make it a valuable tool for anyone who wants to be told to program. It's also the language that is recommended by the Raspberry Pi Foundation for those looking to progress from the easy Scratch. Example : Hello World As you learned in Chapter 10, “An Introduction to Scratch”, the simplest thanks to learn a brand new artificial language is to form a project that prints “Hello World!” on the screen. In Scratch, you only had to tug and drop bricks of prewritten code, but in Python, you wish to put in writing this program entirely by hand. A Python project is, at heart, nothing quite a document containing written instructions for the pc to follow. This file can be created using any text editor. For instance, if you enjoy performing at the console or during a terminal window, you'll be able to use nano; or if you like a graphical programme (GUI), you'll be able to use Leafpad. Another alternative is to use an integrated development environment (IDE) like IDLE, which provides Python-specific functionality that's missing from a customary text editor, including syntax checking, debugging facilities and therefore the ability to run your program without having to go away the editor. This chapter gives you instructions on a way to create Python files using IDLE, but after all, the IDE program that you just value more highly to use for programming is up to you.

The chapter also includes instructions for running your created files directly from the terminal, which can be employed in conjunction with any text editor or other IDE. To begin the Hello World project, open IDLE from the Programming menu within the Debian distribution's desktop environment. If you're not using IDLE, create a blank document in your favourite text editor and skip the remainder of this paragraph. By default, IDLE parades in Python shell mode (see Figure 11-1), so anything you type within the initial window are immediately executed. To open a replacement Python project which might be executed later, click on the File menu and choose New Window to open a blank file. The IDLE Python Shell window. It's good practice to begin all Python programs with a line referred to as a shebang, which gets its name from the # and ! characters at the start of the line. This line tells the software package where it should explore for the Python files. Although this can be not entirely necessary for programs that may be run from within IDLE or will call Python explicitly at the terminal, it's required for programs that are run directly by calling the program's filename. To ensure the program runs irrespective of where the Python executable is installed, the primary line of your program should read as follows: `#!/usr/bin/env python`. This line tells the software system to seem at the `$PATH` environment variable—which is where Linux stores the situation of files that will be executed as programs—for the placement of Python, which should work on any Linux distribution used on the Pi. The `$PATH` variable contains a listing of directories where executable files are stored, and is employed to search out programs after you type their name at the console or in a very terminal window. To achieve the goal of printing out a message, you ought to use Python's print command. As its name suggests, this command prints text to an output device—by default, to the console or terminal window from which the program is being executed. Its usage is simple: any text following the word print and placed between quotation marks are printed to the quality output device. Enter the subsequent line in your new project: `print "Hello, World!"` The final program should appear as if this: `#!/usr/bin/env python print "Hello, World!"` If you're creating the instance program in IDLE instead of a clear text editor, you'll notice that the text is multicolored, (where colours are represented as differing reminder grey within the print edition). this can be a feature referred to as syntax highlighting, and may be a feature of IDEs and therefore the more-advanced text editing tools. Syntax highlighting changes the color of sections of the text per their function, so as to form the program easier to know at a look. It also makes it easy to spot so-called syntax errors caused by forgetting to place an end-quote in an exceedingly print command or forgetting to comment out a remark. For this short example, syntax highlighting isn't necessary—but in larger programs, it may be a useful tool for finding errors. Syntax highlighting in IDLE Before you run your program, put it aside as `helloworld.py` using the File menu. If you're using IDLE, the file are going to be given the extension `.py` automatically. If you're employing a text editor, take care to type `.py` at the tip of the filename (not `.txt`) when you save it. This extension indicates that the file contains Python code—although Python is clever enough to run the program whether or not it's saved with a distinct file extension. How you run the file will depend upon whether you're using IDLE or a text editor. In IDLE, simply choose Run Module from the Run menu, or press the F5 key on the keyboard. this may switch IDLE back to the Python shell window and run the program. You should then see the message Hello, World! appear on screen in blue. If not, check your syntax—in particular, certify you have got quotation marks at both the start and end of the message on the print line. Running `helloworld.py` in IDLE If you created the `helloworld.py` program in an

exceedingly text editor, you'll must open a terminal window from the Accessories menu on the desktop. If you saved the file anywhere except your home directory, you'll even have to use the `cd` command to alter to that directory (see Chapter 2, "Linux System Administration"). Once you're within the right directory, you'll be able to run your program by typing the following: `python helloworld.py`. This tells the package to run Python and so load the `helloworld.py` file for execution. Unlike the Python shell in IDLE, Python will quit when it reaches the tip of the file and return you to the terminal. The result, however, is that the same: the message `Hello, World!` is printed to the quality output.

Running `helloworld.py` at the terminal

Making Python Programs Executable

Normally, the sole thanks to run a Python program is to inform the Python software to open the file. With the shebang line at the highest of the file, however, it's possible to execute the file directly without having to call Python first. This may be a useful way of constructing your own tools which will be executed at the terminal: once copied into a location within the system's `$PATH` environment variable, the Python program will be called just by typing its name.

First, you wish to inform Linux that the Python file should be marked as executable—an attribute which means the file may be a program. To shield the system from malware being downloaded from the net this attribute isn't automatically set, since only files that are marked as executable will run. To create the `helloworld.py` file executable, use the `chmod` command (described intimately in Chapter 2, "Linux System Administration") by typing the following: `chmod +x helloworld.py`

Now try running the program directly by typing the following: `./helloworld.py`. Despite the very fact that you simply didn't call the Python program, the `helloworld.py` program should run just the identical as if you'd typed `python helloworld.py`. The program can only be go past calling it with its full location—`/home/pi/helloworld.py`—or from this directory by using `./` because the location. To make the file accessible within the same way as the other terminal command, it must be copied to `/usr/local/bin` with the subsequent command: `sudo cp helloworld.py /usr/local/bin/`. The `sudo` prefix is required because, for security reasons, non-privileged users cannot write to the `/usr/local/bin` directory. With the `helloworld.py` file located in `/usr/local/bin`, which is included within the `$PATH` variable, it may be executed from any directory by simply typing its name. Try changing to a different directory, then run the program by typing the following: `helloworld.py`

To make your custom-made programs seem more like native utilities, you'll be able to rename them to get rid of the `.py` file extension. To vary the `helloworld.py` program during this way, just type the subsequent line at the terminal as one line: `sudo mv /usr/local/bin/helloworld.py /usr/local/bin/helloworld`. Once renamed, the program may be run just by typing `helloworld` at the terminal or console.

4.9 IMAGE PROCESSING USING OPENCV IN PYTHON

INTRODUCTION

Image processing is the process of manipulating pixel data in order to make it suitable for computer vision applications or to make it suitable to present it to humans. For example, changing brightness or contrast is a image processing task which make the image visually pleasing for humans or suitable for further processing for a certain computer vision application.

PYTHON ENVIRONMENT

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Python's standard library

- Pandas
- Numpy
- Sklearn
- seaborn
- matplotlib
- Importing Datasets
- Opencv

INTRODUCTION TO OPEN CV

OpenCV is Open Computer Vision Library . It was initially launched in 1999 by Intel. With more updates, it has been modified since then to aim for the real-time computer vision. This library has been written under programming languages like C and C+. It can be easily run on operating systems Windows and Linux. This library can be easily interface with programming languages like Python, MATLAB, Ruby and others as well. Along with Numpy and Python image processing (shape & color detection) can be performed at ease.

Install Python-OpenCV

To implement this project, the following packages of Python 3.7 have to be downloaded and installed: Python 2.7.x, NumPy and Matplotlib. According to the default location, Python will be installed to C drive. Open Python IDLE, import all the packages and start working.

Read an Image

First, a sample image in which processing is to be applied is to be read. It's done using a pre-defined Python function: `CV2.imread()`. The sample image should be available in current folder or the full location of the image is to be mentioned as an argument. For reading an image, we can use functions like `imread_color`, `imread_grayscale`, `imread_unchanged` etc.

PANDAS

Pandas is quite a game changer when it comes to analyzing data with Python and it is one of the most preferred and widely used tools in data munging/wrangling if not THE most used one. Pandas is an open source

What's cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example. People who are familiar with R would see similarities to R too). This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

Installation and Getting Started

In order to “get” Pandas you would need to install it. You would also need to have Python 2.7 and above as a pre-requirement for installation. It is also dependent on other libraries (like NumPy) and has optional dependancies (like Matplotlib for plotting). Therefore, I think that the easiest way to get Pandas set up is to install it through a package like the Anaconda distribution , “a cross platform distribution for data analysis and scientific computing.”

In order to use Pandas in your Python IDE (Integrated Development Environment) like Jupyter Notebook or Spyder (both of them come with Anaconda by default), you need to import the Pandas library first. Importing a library means loading it into the memory and then it's there for you to work with. In order to import Pandas all you have to do is run the following code:

- **import pandas as pd**
- **import numpy as np**

Usually you would add the second part ('as pd') so you can access Pandas with 'pd.command' instead of needing to write 'pandas.command' every time you need to use it. Also, you would import numpy as well, because it is very useful library for scientific computing with Python. Now Pandas is ready for use! Remember, you would need to do it every time you start a new Jupyter Notebook, Spyder file etc.

Working with Pandas

Loading and Saving Data with Pandas

When you want to use Pandas for data analysis, you'll usually use it in one of three different ways:

- Convert a Python's list, dictionary or Numpy array to a Pandas data frame
- Open a local file using Pandas, usually a CSV file, but could also be a delimited text file (like TSV), Excel, etc
- Open a remote file or database like a CSV or a JSON on a website through a URL or read from a SQL table/database

There are different commands to each of these options, but when you open a file, they would look like this:

- **pd.read_filetype()**

As I mentioned before, there are different filetypes Pandas can work with, so you would replace “filetype” with the actual, well, filetype (like CSV). You would give the path, filename etc inside the parenthesis. Inside the parenthesis you can also pass different arguments that relate to how to open the file. There are numerous arguments and in order to know all you them, you would have to read the documentation (for example, the documentation for `pd.read_csv()` would contain all the arguments you can pass in this Pandas command).

In order to convert a certain Python object (dictionary, lists etc) the basic command is:

- **`pd.DataFrame()`**

Inside the parenthesis you would specify the object(s) you’re creating the data frame from. This command also has different arguments .

You can also save a data frame you’re working with/on to different kinds of files (like CSV, Excel, JSON and SQL tables). The general code for that is:

- **`df.to_filetype(filename)`**

Viewing and Inspecting Data

Now that you’ve loaded your data, it’s time to take a look. How does the data frame look? Running the name of the data frame would give you the entire table, but you can also get the first n rows with `df.head(n)` or the last n rows with `df.tail(n)`. `df.shape` would give you the number of rows and columns. `df.info()` would give you the index, datatype and memory information. The command `s.value_counts(dropna=False)` would allow you to view unique values and counts for a series (like a column or a few columns). A very useful command is `df.describe()` which inputs summary statistics for numerical columns. It is also possible to get statistics on the entire data frame or a series (a column etc):

- `df.mean()` Returns the mean of all columns
- `df.corr()` Returns the correlation between columns in a data frame
- `df.count()` Returns the number of non-null values in each data frame column
- `df.max()` Returns the highest value in each column
- `df.min()` Returns the lowest value in each column
- `df.median()` Returns the median of each column
- `df.std()` Returns the standard deviation of each column

Selection of Data

One of the things that is so much easier in Pandas is selecting the data you want in comparison to selecting a value from a list or a dictionary. You can select a column (`df[col]`) and return column with label `col` as Series or a few columns (`df[[col1, col2]]`) and returns columns as a new DataFrame. You can select by position (`s.iloc[0]`), or by index (`s.loc['index_one']`) . In order to select the first row you can use `df.iloc[0,:]` and in order to select the first element of the first column you would run `df.iloc[0,0]` . These can also be used in different combinations, so I hope it gives you an idea of the different selection and indexing you can perform in Pandas.

Filter, Sort and Groupby

You can use different conditions to filter columns. For example, `df[df['year'] > 1984]` would give you only the column year is greater than 1984. You can use `&` (and) or `|` (or) to add different conditions to your filtering. This is also called boolean filtering.

It is possible to sort values in a certain column in an ascending order using `df.sort_values(col1)` ; and also in a descending order using `df.sort_values(col2,ascending=False)`. Furthermore, it's possible to sort values by `col1` in ascending order then `col2` in descending order by using `df.sort_values([col1,col2],ascending=[True,False])`.

The last command in this section is `groupby`. It involves splitting the data into groups based on some criteria, applying a function to each group independently and combining the results into a data structure. `df.groupby(col)` returns a groupby object for values from one column while `df.groupby([col1,col2])` returns a groupby object for values from multiple columns.

NUMPY

Numpy is one such powerful library for array processing along with a large collection of high-level mathematical functions to operate on these arrays. These functions fall into categories like Linear Algebra, Trigonometry, Statistics, Matrix manipulation, etc.

Getting NumPy

NumPy's main object is a homogeneous multidimensional array. Unlike python's array class which only handles one-dimensional array, NumPy's ndarray class can handle multidimensional array and provides more functionality. NumPy's dimensions are known as axes. For example, the array below has 2 dimensions or 2 axes namely rows and columns. Sometimes dimension is also known as a rank of that particular array or matrix.

Importing NumPy

NumPy is imported using the following command. Note here np is the convention followed for the alias so that we don't need to write numpy every time.

- `import numpy as np`

NumPy is the basic library for scientific computations in Python and this article illustrates some of its most frequently used functions. Understanding NumPy is the first major step in the journey of machine learning and deep learning.

Data Visualization in Python

Data visualization is the discipline of trying to understand data by placing it in a visual context, so that patterns, trends and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly customized plots python has a excellent library for you.

CHAPTER-5

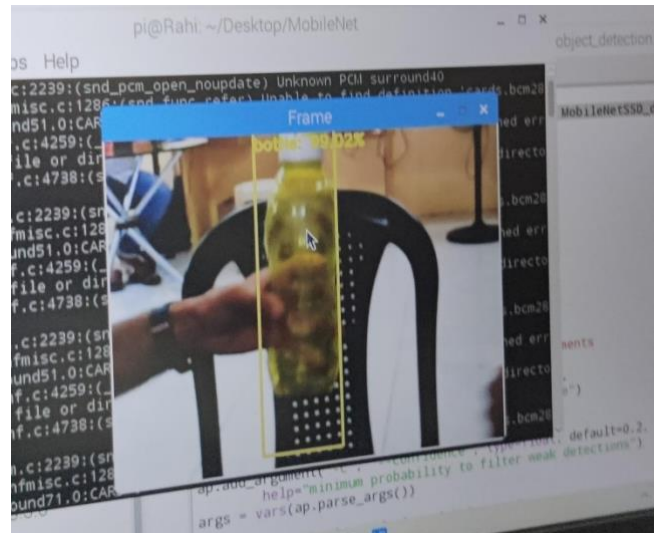
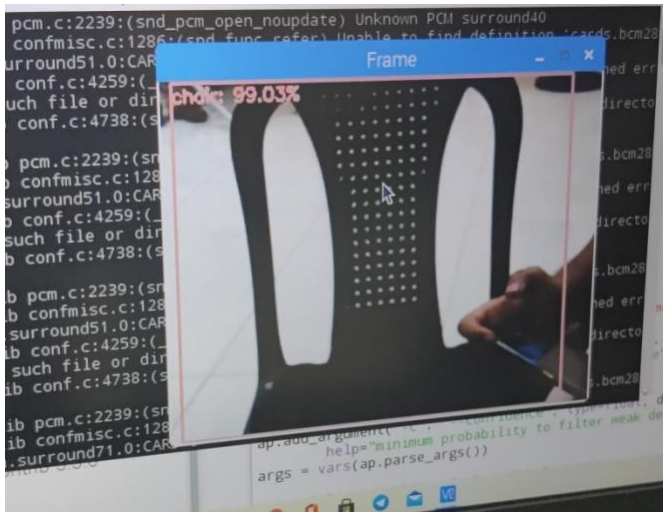
RESULTS

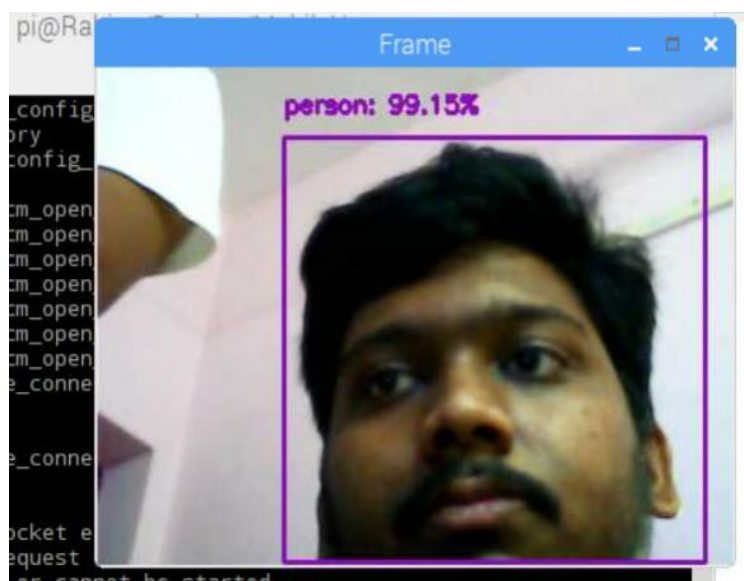
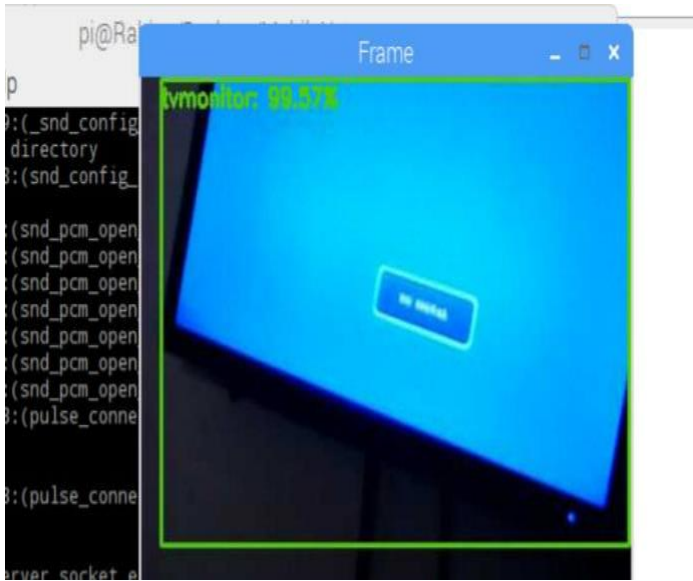
- (1) Determining obstacles around the user body from the ground to the head;
- (2) Affording some instructions to the user about the movement surface consists of gaps or hole on floor and obstacle in front;
- (3) Finding items surrounding the obstacles;
- (4) Affording information to give the ability of self-orientation and mental map of the surroundings.

LIMITATIONS:

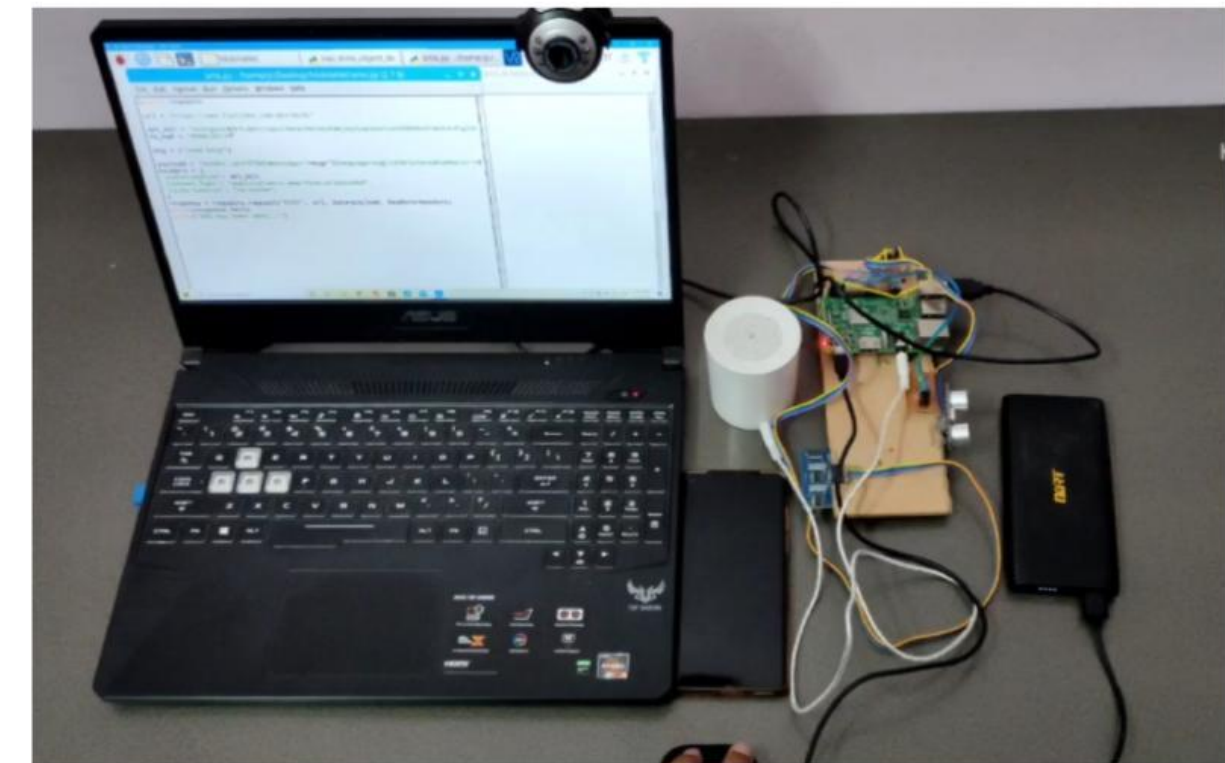
- It takes times to load program.
- The raspberry pi have overclocking of the chip.
- The device does not have the graphics processor.
- This device need a large battery power for working.

Digital output:





Finally developed model:



CHAPTER-6

APPLICATIONS AND FUTURE SCOPE

Computer vision which go beyond image processing, helps to obtain relevant information from images and make decisions based on that information. In other words, computer vision is making the computer see as humans do. Basic steps for a typical computer vision application as follows.

1. Image acquisition
2. Image manipulation
3. Obtaining relevant information
4. Decision making

Computer vision has still not attained a level wherein it can be directly put into use to solve life problems, as it is still in its developmental phase. With passing years and rigorous pace at which research is being done, Computer Vision or to be precise, Object detection will be completely omnipresent. Computer Vision is a sub-part of Machine Learning. Some common and widely used applications of object detection are:

- Accounting Number of Objects
- Automobile Spotting
- Biometric Detection
- Medical Diagnosis
- Supervision
- Machine Man Communication

CONCLUSION

In this paper we have used the functions of opencv. It is free for both commercial and non-commercial use. Therefore you can use the OpenCV library even for your commercial applications. It is a library mainly aimed at real time processing. Now it has several hundreds of inbuilt functions which implement image processing and computer vision algorithms which make developing advanced computer vision applications easy and efficient.

REFERENCES

- [1] Durette, B., Louveton, N., Alleysson, D., and H´erault, J, 2008. Visuoauditory sensory substitution for mobility assistance: testing The VIBE. In Workshop on Computer Vision Applications for the Visually Impaired, Marseille, France.
- [2] Hern´andez, A. F. R. et al, 2009. Computer Solutions on Sensory Substitution for Sensory Disabled People. In Proceedings of the 8th WSEAS International Conference on Computational Intelligence, Manmachine Systems and Cybernetics, pp. 134–138.
- [3] Tang, H., and Beebe, D. J., 2006. An oral tactile interface for blind navigation. IEEE Trans Neural SystRehabilEng, pp. 116–123.
- [4] Auvray, M., Hanneton, S., and O’Regan, J. K., 2007. Learning to perceive with a visuo - auditory substitution system: Localisation and object recognition with ‘The vOICe’. Perception, pp. 416–430.
- [5] Kammoun, S. et al, 2012. Navigation and space perception assistance for the visually impaired: The NAVIG project. In IRBM, Numrospcial ANR TECSAN, 33(2), pp.182–189.
- [6] A. M. Flynn, “Combining sonar and infrared sensors for mobile robot navigation,” The International Journal of Robotics Research, vol. 7, no. 6, pp. 5–14, 1988.
- [7] R. R. Igorevich, P. Park, J. Choi, and D. Min, “Two hand gesture recognition using stereo camera,” International Journal of Computer & Electrical Engineering, vol. 5, no. 1, pp. 69–72, 2013.

project report

ORIGINALITY REPORT

15%

SIMILARITY INDEX

10%

INTERNET SOURCES

0%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1

www.electronicshub.org

Internet Source

5%

2

coecsl.ece.uiuc.edu

Internet Source

5%

3

Submitted to Stratford University

Student Paper

5%

Exclude quotes On

Exclude matches < 5%

Exclude bibliography On