**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR nagar, MSRIT post, Bangalore-54

A Dissertation Report on

Smart Doorbell using IOT

Submitted by

**Sunayana Bhowmik** 1MS12CS117

**Annapurna T** 1MS13CS401

**Santhoshi R** 1MS12CS145

**Bindu Shree G** 1MS13CS403

*in partial fulfillment for the award of the degree of*

# *Bachelor of Engineering in Computer Science & Engineering*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S.RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[www.msrit.edu](http://www.msrit.edu), **May 2015**

# ABSTRACT

The project basically deals with connecting doorbell to the internet. Whenever someone presses our doorbell, our smart phones or tablet will chime so we will never miss a delivery or visitor. Motion sensor that detects any kind of activity on your property and trigger instant **mobile alerts**. See and speak with visitors from anywhere using your smart phone, and watch recorded footage anytime with Cloud Recording.

With the development of technology and the continuous improvement of people's living standard, people are in pursuit of automated, intelligent and convenient home control systems. At present, the PC is used as the remote control terminal for most home control systems; however, there are some problems in the PC monitor terminal, such as its great bulk, inconvenience to carry, high cost, limited monitoring range and so on. Therefore, it‟s a good choice to design a terminal based on phone. With the popularity of smart phones, particularly, the phone based on Android system is rapidly developed. At its I/O developer conference, Google showed a sneak preview of its Android Home project, which will extend the Android platform into household objects. It means that the remote control based on Android phone will become a mainstream way. After logging into the control interface, users can easily control the lights, TVs and air conditionings anytime, anywhere, which brings great convenience to people and improves the quality of life.

#### Contents

***Declaration i***

***Acknowledgements ii***

***Abstract iii***

***List of Figures***

***List of Tables***

1 **INTRODUCTION**

* 1. General Introduction………………. **<Page No>**
  2. Statement of the Problem…………..
  3. Objectives of the project……………
  4. Project deliverables……………
  5. Current Scope………………………
  6. Future Scope……………………….

1. **PROJECT ORGANIZATION**
   1. Software Process Models
   2. Roles and Responsibilities
2. **LITERATURE SURVEY**

3.1….Introduction

3.2…Main Body

3.3 Conclusion of Survey

1. **SOFTWARE REQUIREMENT SPECIFICATIONS**

5.1 Product Overview

5.2 External Interface Requirements

5.2.1 User Interfaces

5.2.2 Hardware Interfaces

5.2.3 Software Interfaces

5.2.4 Communication Interfaces

5.3 Functional Requirements

5.3.1 Functional Requirement 1.1

:

5.3.n Functional Requirement 1.n

5.4 Software System Attributes

5.4.1 Reliability

5.4.2 Availability

5.4.3 Security

5.4.4 Portability

5.4.5 Maintainability

5.4.6 Performance

5.5 Performance Requirements

5.6 Database Requirement

5.7 Design Constraints

5.8 Other Requirements

1. **DESIGN**
   1. Introduction
   2. Architecture Design
   3. Graphical User Interface
   4. Class Diagram and Classes (represent Inheritance, Aggregation and Association)
   5. Sequence Diagram
   6. Data flow diagram
   7. Metric calculation
2. **IMPLEMENTATION**
   1. Tools Introduction
   2. Technology Introduction
   3. Overall view of the project in terms of implementation
   4. Explanation of Algorithm and how it is been implemented
   5. Information about the implementation of Modules
3. **TESTING**

**7.1** Results and Snapshots

1. **CONCLUSION & SCOPE FOR FUTURE WORK <No Page Numbers>**
2. **REFERENCES**

**INTRODUCTION**

**General Introduction**

**Statement of the Problem :-**

The “Smart Doorbell” uses internet of things to notify people about their visitor through smart phones. Whenever someone presses our doorbell, our smartphone or tablet will chime so we will never miss a delivery or visitor.The project is properly represented using SRS,classdigram,sequencedigrametc.We keep in notice that the requirements of the client are fulfilled.

**Objectives of the project**

* To Enhance the security
* To see the live streaming across world.

**Project deliverables**

* Initial project report on requirement
* SRS
* Design document
* Implementation code
* Final project
* Output of the project
* Final project report

**Current Scope**

The System Test Plan(STP)is necessary to ensure product design and implementation meets the complete product as specified in the System Requirements Specification(SRS), and the Detail Design Specification (DDS). It will be used as away to validate and verify our product solves the original problem given.

The users can manipulate appliances anytime, anywhere, letting our houses become more and more automated and intelligent. At present, the PC is used as the remote control terminal for most home control systems [however, there are some problems in the PC monitor terminal, such as its great bulk, inconvenience to carry, high cost, and limited monitoring range and so on. Therefore, it‟s a good choice to design a terminal based on phone.

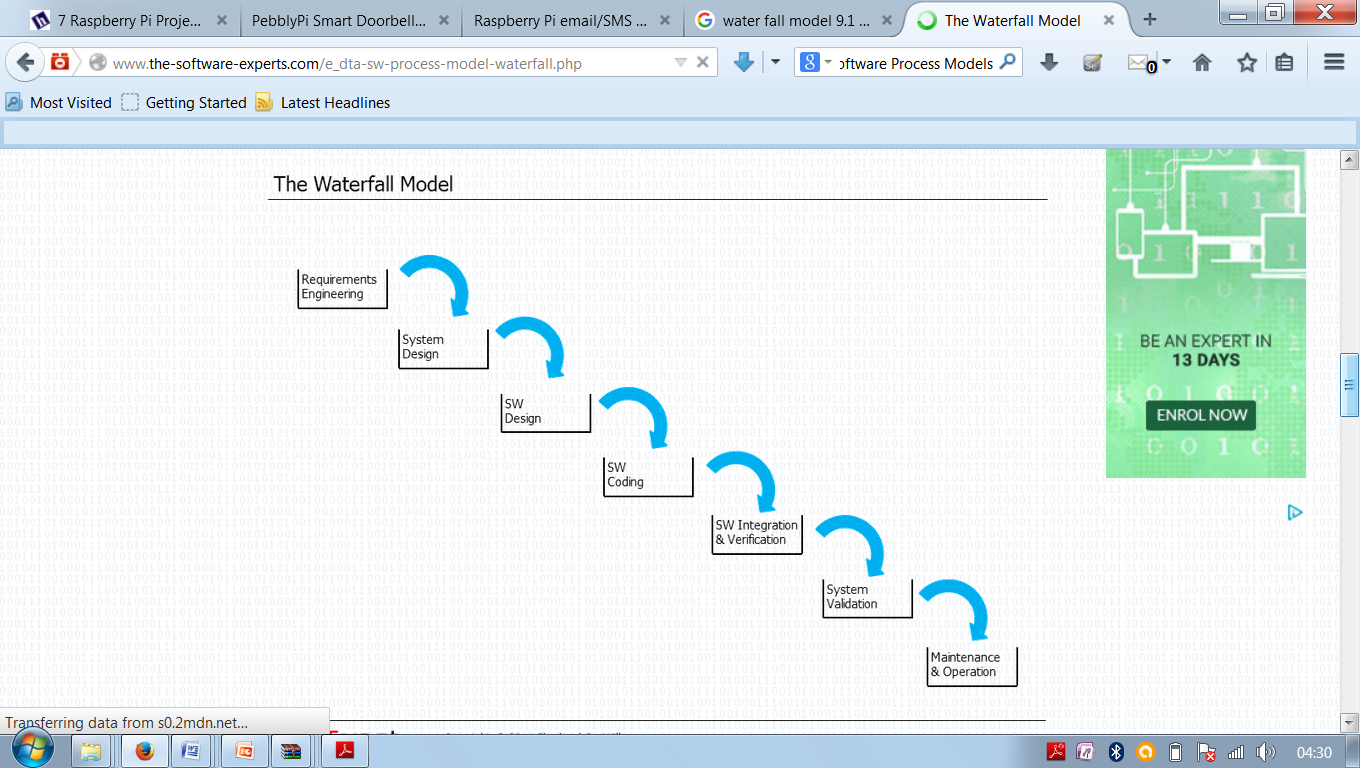
**Future Scope**

Furthermore technology is not the solution to create a perfect home environment but has the ability to make a useful contribution. The environment as a whole, including for instance social contacts and location of the home, is responsible for the overall satisfaction of the residents.

After this introduction the chapter continues with an overall description of smart home technology, relevant trends and stakeholders, referred to as the smart home framework. In the next three sections several aspects of the smart home environment are described in more detail. Each of these sections uses a different viewpoint. The viewpoints are: technological, products and services and user interaction. The sixth section is devoted to realized projects, which are divided in projects with residents and research facilities. The chapter ends with a section discussing the challenges for smart home technology in the near future.

**PROJECT ORGANIZATION**

# SOFTWARE PROCESS MODELS- The Waterfall Model



The waterfall model is believed to have been the first process model which was introduced and widely followed in software engineering. The innovation was that the first time software engineering was divided into separate phases. When I did my first programs in PL/1 and RPG in the early 1970's there was no awareness of splitting up software development into different phases. Programs were very small, the requirements only a few and after punching a pile of cards the program was done and could be tested by inserting it into the card reader and observing what it did.   
  
As programs became bigger the need for a better requirements phase, some more thoughts on the design, etc. were needed. Programmers found it more and more difficult to keep an abstract of the program in their mind and transfer it into code. Also the thought of having a separate testing phase performed by dedicated testers evolved. The different phases of software engineering were identified and simply cascaded in each other, allowing for loops in case it was found in a subsequent phase that the previous phase was not done properly.

Phases:

Requirement Analysis & Definition:

All requirements of the system which has to be developed are collected in this step. Like in other process models requirements are split up in functional requirements and constraints which the system has to fulfill. Requirements have to be collected by analyzing the needs of the end user(s) and checking them for validity and the possibility to implement them. The aim is to generate a Requirements Specification Document which is used as an input for the next phase of the model.   
  
System Design:

The system has to be properly designed before any implementation is started. This involves an architectural design which defines and describes the main blocks and components of the system, their interfaces and interactions. By this the needed hardware is defined and the software is split up in its components. E.g. this involves the definition or selection of a computer platform, an operating system, other peripheral hardware, etc. The software components have to be defined to meet the end user requirements and to meet the need of possible scalability of the system. The aim of this phase is to generate a System Architecture Document this serves as an input for the software design phase of the development,.   
  
Software Design:

Based on the system architecture which defines the main software blocks the software design will break them further down into code modules. The interfaces and interactions of the modules are described, as well as their functional contents. All necessary system states like startup, shutdown, error conditions and diagnostic modes have to be considered and the activity and behavior of the software has to be defined. The output of this phase is a Software Design Document which is the base of the following implementation work.

Coding:

Based on the software design document the work is aiming to set up the defined modules or units and actual coding is started. The system is first developed in smaller portions called units. They are able to stand alone from an functional aspect and are integrated later on to form the complete software package.   
  
Software Integration & Verification:

Each unit is developed independently and can be tested for its functionality. This is the so called Unit Testing. It simply verifies if the modules or units to check if they meet their specifications. This involves functional tests at the interfaces of the modules, but also more detailed tests which consider the inner structure of the software modules. During integration the units which are developed and tested for their functionalities are brought together. The modules are integrated into a complete system and tested to check if all modules cooperate as expected.

System Validation:

After successfully integration including the related tests the complete system has to be tested against its initial requirements. This will include the original hardware and environment, whereas the previous integration and testing phase may still be performed in a different environment or on a test bench.Operation & Maintenance:

The system is handed over to the customer and will be used the first time by him. Naturally the customer will check if his requirements were implemented as expected but he will also validate if the correct requirements have been set up in the beginning. In case there are changes necessary it has to be fixed to make the system usable or to make it comply to the customer wishes. In most of the "Waterfall Model" descriptions this phase is extended to a never ending phase of "Operations & Maintenance". All the problems which did not arise during the previous phases will be solved in this last phase.

**ROLES AND RESPONSIBILITIES**

* Cheaper
* Part of a totality
* Savings also for parts of conventional installations

The installation can be concealed

**LITERATURE SURVEY**

**INTRODUCTION**

For this proposed Seminar, following IEEE papers were studied as part of literature survey. Smart Home System for Disabled People Via Wireless Bluetooth gives moneywise concept by using GPRS as the medium to control and monitor home appliances.

Design and Realization of Home Appliances Control System Based on The Android Smartphone present the information about the remote appliances control system based on the Android smart phone is designed and realized. A user logs into the smart phone interface, and clicks the buttons gently to send message commands which will be transmitted to home information Centre through the GSM network. Then the PIC processor recognizes the specified command, and controls the home appliance switches in the wireless radio frequency manner to achieve remote control of appliances ultimately. Exploiting Bluetooth on android mobile devices for home security application present the information about mobile devoice has been integrated into our everyday life. Home automation and security are becoming increasingly prominent features on mobile devoices the mobile devoice and security system communicates via Bluetooth because a short-range-only communication system was desired. With the help of android mobile we can control task such as locking the doors, turning on/off lights remotely. According to , home automation can be useful to those who need to access home appliances while away from their home and can improve the lives of the disabled

**SOFTWARE REQUIREMENT SPECIFICATIONS**

# 1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “Smart Doorbell Project” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications

## Scope

The “Smart Doorbell project” helps us to notify if there is some delivery on the door and we don’t miss it if we are not present there. The project basically deals with connecting doorbell to the internet. Whenever someone presses our doorbell, our smart phone or tablet will chime so we will never miss a delivery or visitor. Motion sensor that detects any kind of activity on your property and trigger instant **mobile alerts**. See and speak with visitors from anywhere using your smart phone, and watch recorded footage anytime with Cloud Recording.

## 2. External Interface Requirements

### 2.1 User Interfaces:

The “smart doorbell” has two user interface. The one is sms application and email which provides us notification about the presence of delivery people. The other is doorbell button or camera which senses the presence of delivery man.

### 2.2 Hardware Interfaces

Hardware interfaces are doorbell itself, Raspberry pi, LCD, Sensors,USB Cables and connecting cables.

### 2.3 Software Interfaces:

The software interface includes various kind of IDE’s using which we going to code and feed the code in the memory. The various IDE’s can be used are Blue IDE or Code blocks IDE. The other software interface is mobile sms apps and how it is communicating with OS and network.

### 2.4 Communications Interfaces:

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and the web portal.

## 3.Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

### 3.1 Functional Requirement 1.1

3.1.1 TITLE: Doorbell-Presses the doorbell switch

3.1.2 DESCRIPTION: The delivery person or person from outside presses the key.This is the initial step. On pressing the switch rest of the step is carried out and the owner is informed.

3.2 Functional Requirement 1.2

3.2.1.1TITLE:Camera-Imaging

3.2.2DESCRIPTION:On sensing the doorbell ringing the image of the person may be taken by our camera system and is delivered to the owner .

**3.3 Functional Requirement 1.3**

3.3.1TITLE:Smartphone-Message and email Notification

3.3.2DESCRIPTION:We get a message and email when our doorbell senses that the switch is pressed.

## 4.Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level.

### 4.1 Performance:

The system performs efficiently everywhere if there is presence of Internet.If the speed of internet is less then we may get notification late i.e not at the time of delivery.

### 4.2 Reliability:

System is not reliable only in absence of Internet.It is not reliable if the person is unable to get the notification at correct time.

### 4.3 Availability:

Resources are easily available i.e they are not costly and can be easily bought.

### 4.4 Security:

The system is secure and there is no issues regarding it.

### 4.5 Maintainability:

Easy to maintain.

### 4.6 Portability:

Devices are portable because the connections can be easily unplugged and carried to different places. The output device smart phone is the best portable device.

## 5. Design Constraints

## Title: Application Memory Usage

## Description: The amount of Operate System memory occupied by the application.

## SCALE: MB.

## METER: Observations done from the performance log during testing

## MUST: No more than 20 MB.

## PLAN: No more than 16 MB

## WISH: No more than 10 MB

## Operate System: DEFINED: The mobile Operate System which the application is running on. MB: DEFINED: Megabyte3.8 Logical Database Requirements

**6.Database Requirement:**

Incase we miss the visitor to receive and we are not at the place of delivery the information about the delivery organization,or the person we can store the information in database.

**DESIGN**

**1. Modules:**

* 1. **Interaction between visitor and the device:**

When we press the doorbell switch this function is called and executed. The hardware components used implement this module mostly act as user interface.

* 1. **Sending signal from doorbell to microcontroller:**

After executing above module this module is called. In this module sound or motion detected is converted into electrical signal.

**1.3displaying the message in LCD:**

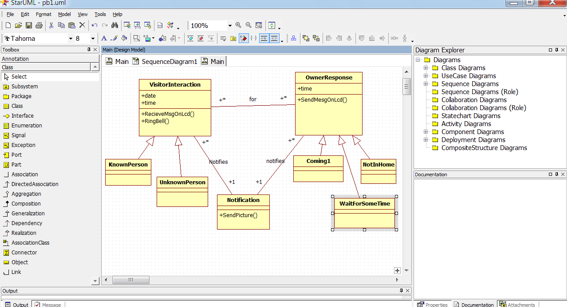
As the above module is executed the code in the flash memory is loaded and executed and the output is displayed on the LCD.

**1.4Sending notification to smartphone:**

After the module 1.2 is executed this module is executed simultaneously along with module 1.3 this module fails to execute in the absence of internet.

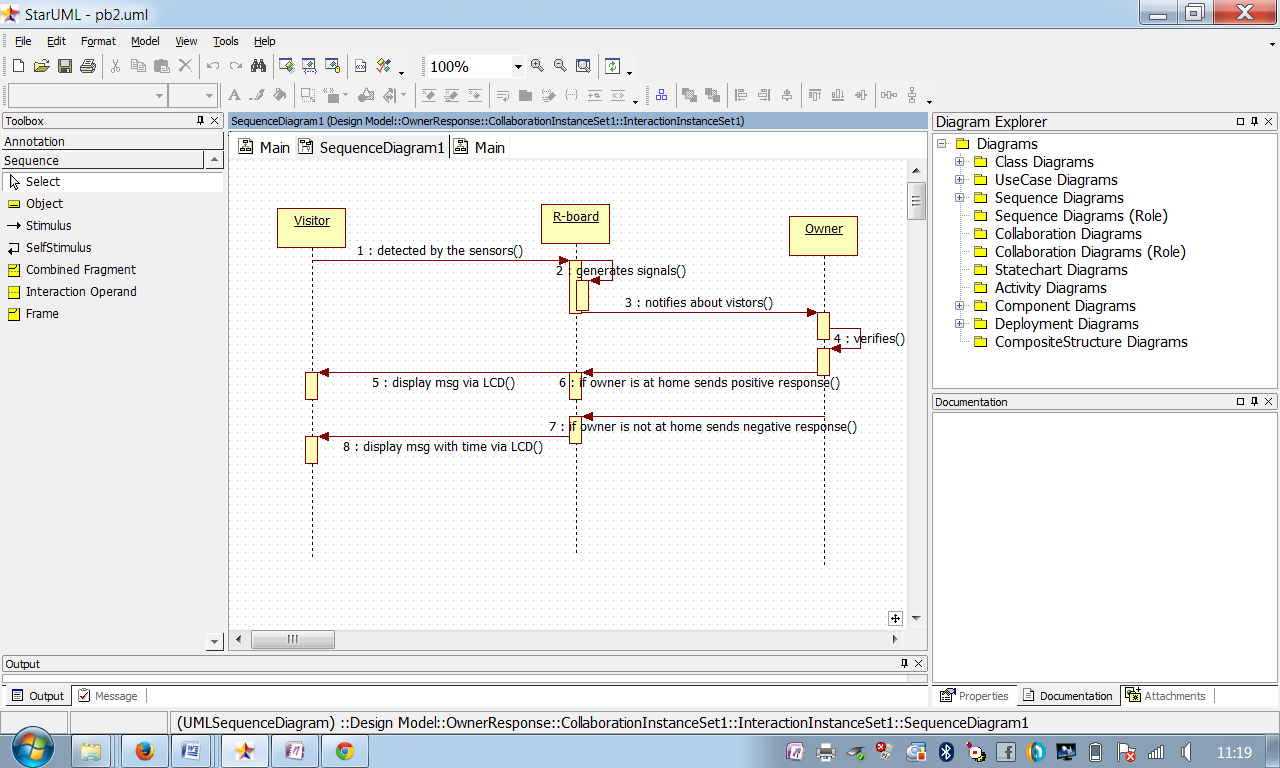
**2. Class diagram:**

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), a **class diagram** in the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a type of static structure diagram that describes the structure of a system by showing the system's [classes](https://en.wikipedia.org/wiki/Class_(computer_science)), their attributes, operations (or methods), and the relationships among objects.

****

**3.Sequence Digram:**

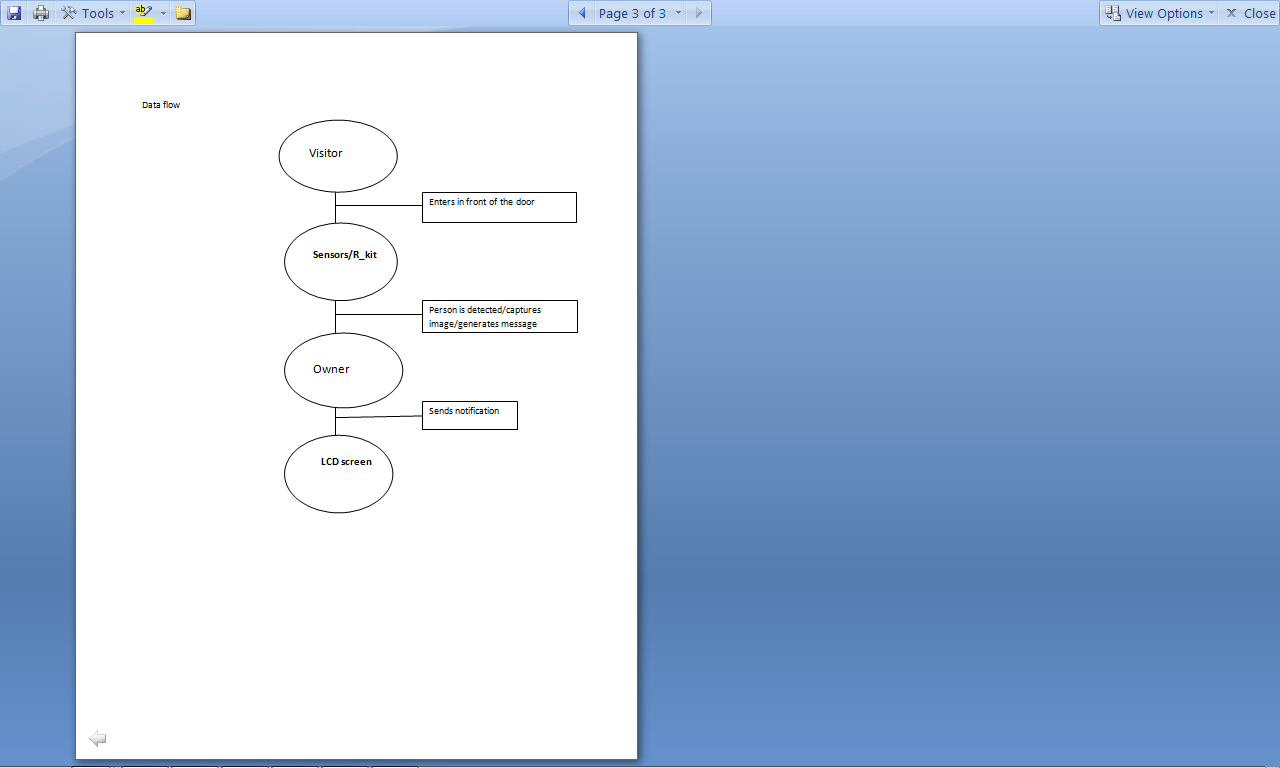
A **Sequence diagram** is an [interaction diagram](https://en.wikipedia.org/wiki/Interaction_diagram) that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.



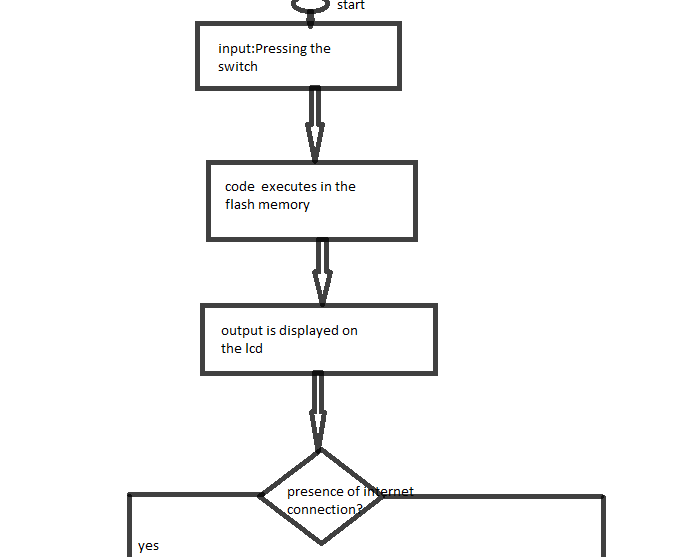
**4.Data Flow Digram:**

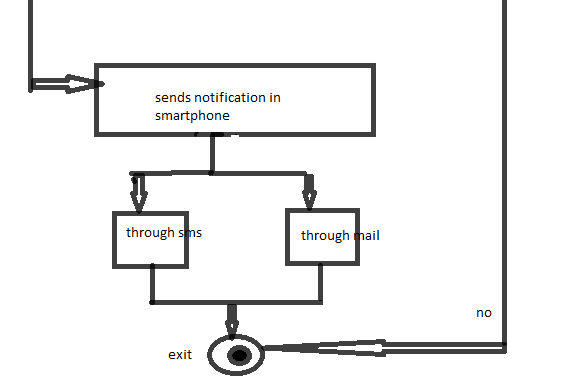
A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

1)



2)

****

****

**IMPLEMENTATION**

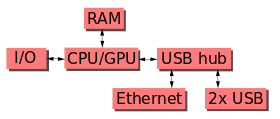
**TOOLS INTRODUCTION**

**PuTTY** is a popular SSH and **Telnet client** that helps you establish secure connections over the Internet and doesn't even require installation. It's especially aimed for programmers and network administrators, which means that newcomers won't find it easy to use. The program features a simple, straightforward interface with no included documentation. Despite its apparent simplicity, PuTTY is highly configurable and includes many options to tweak connections, sessions, SSH security features and even the window's appearance.

* 1. **TECHNOLOGY INTRODUCTION**

**Raspberry Pi**

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity, and peripheral device support.

[](https://en.wikipedia.org/wiki/File:Raspberrypi_block_function_v01.svg)

This block diagram depicts models *A*, *B*, *A+*, and *B+*. Model *A* and *A+* and *Zero* lack the Ethernet and USB hub components. The Ethernet adapter is connected to an additional USB port. In model *A* and *A+* the USB port is connected directly to the SoC. On model *B+* the chip contains a five-point USB hub, of which four ports are available, while model *B* only provides two. On the model *Zero*, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

**Python (programming language)**

**Python** is a widely used [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language) Its design philosophy emphasizes code [readability](https://en.wikipedia.org/wiki/Readability), and its syntax allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Lines_of_code) than would be possible in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B) or [Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29). The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming) or [procedural](https://en.wikipedia.org/wiki/Procedural_programming) styles. It features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management) and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using [third-party](https://en.wikipedia.org/wiki/Third-party_software_component) tools, such as [Py2exe](https://en.wikipedia.org/wiki/Py2exe) or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

[CPython](https://en.wikipedia.org/wiki/CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) and has a community-based development model, as do nearly all of its alternative implementations. CPython is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation).

Python's developers strive to avoid [premature optimization](https://en.wikipedia.org/wiki/Premature_optimization), and moreover, reject patches to non-critical parts of CPython that would offer a marginal increase in speed at the cost of clarity.When speed is important, Python programmers use [PyPy](https://en.wikipedia.org/wiki/PyPy), a [just-in-time compiler](https://en.wikipedia.org/wiki/Just-in-time_compilation), or move time-critical functions to extension modules written in languages such as C. [Cython](https://en.wikipedia.org/wiki/Cython) is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of the Python developers is making Python fun to use. This is reflected in the origin of the name, which comes from [Monty Python](https://en.wikipedia.org/wiki/Monty_Python), and in an occasionally playful approach to tutorials and reference materials, such as using examples that refer to spam and eggs instead of the standard [foo and bar](https://en.wikipedia.org/wiki/Foobar).

A common [neologism](https://en.wikipedia.org/wiki/Neologism) in the Python community is *pythonic*, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called *unpythonic*.

Users and admirers of Python—especially those considered knowledgeable or experienced—are often referred to as *Pythonists*, *Pythonistas*, and *Pythoneers*.

### Statements and control flow

Python's statements include (among others):

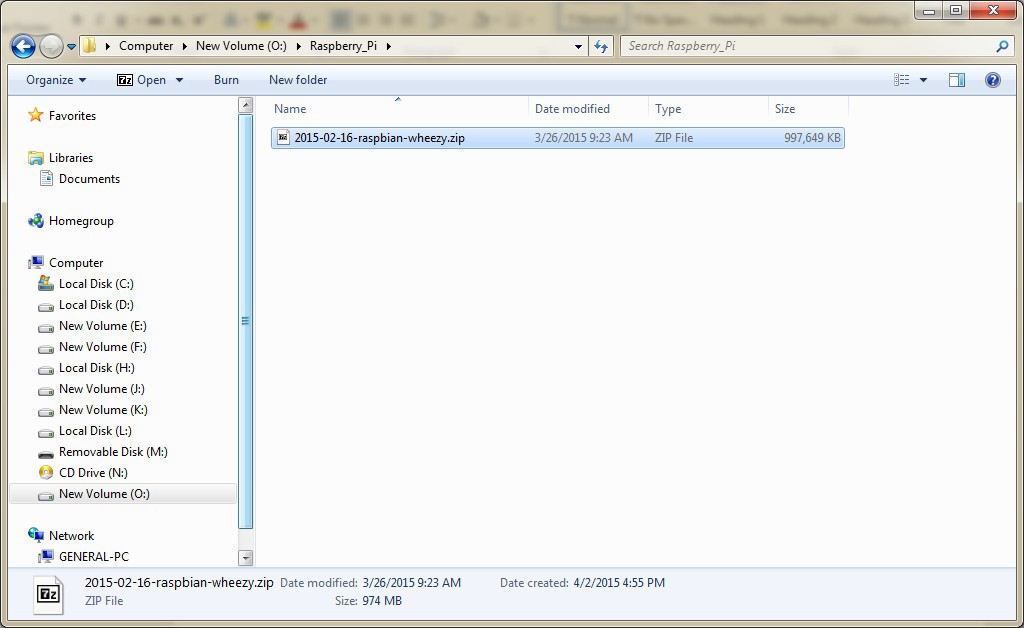
* The [if statement](https://en.wikipedia.org/wiki/If-then-else), which conditionally executes a block of code, along with else and elif (a contraction of else-if).
* The [for statement](https://en.wikipedia.org/wiki/Foreach#Python), which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
* The [while statement](https://en.wikipedia.org/wiki/While_loop#Python), which executes a block of code as long as its condition is true.
* The [try](https://en.wikipedia.org/wiki/Exception_handling_syntax#Python) statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
* The class statement, which executes a block of code and attaches its local namespace to a [class](https://en.wikipedia.org/wiki/Class_%28computer_science%29), for use in [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming).
* The def statement, which defines a [function](https://en.wikipedia.org/wiki/Function_%28computing%29) or [method](https://en.wikipedia.org/wiki/Method_%28computing%29).
* The with statement (from Python 2.5), which encloses a code block within a context manager (for example, acquiring a [lock](https://en.wikipedia.org/wiki/Lock_%28computer_science%29) before the block of code is run and releasing the lock afterwards, or opening a [file](https://en.wikipedia.org/wiki/Computer_file) and then closing it), allowing [RAII](https://en.wikipedia.org/wiki/Resource_Acquisition_Is_Initialization)-like behavior.
* The pass statement, which serves as a [NOP](https://en.wikipedia.org/wiki/NOP). It is syntactically needed to create an empty code block.
* The [assert statement](https://en.wikipedia.org/wiki/Assertion_%28programming%29), used during debugging to check for conditions that ought to apply.
* The yield statement, which returns a value from a [generator](https://en.wikipedia.org/wiki/Generator_%28computer_science%29#Python) function. From Python 2.5, yield is also an operator. This form is used to implement [coroutines](https://en.wikipedia.org/wiki/Coroutine).
* The import statement, which is used to import modules whose functions or variables can be used in the current program.
* The print statement was changed to the print() function in Python 3.[[](https://en.wikipedia.org/wiki/Python_%28programming_language%29#cite_note-diff_py2-3-51)

**OVERALL VIEW OF THE PROJECT IN TERMS OF IMPLEMENTATION**

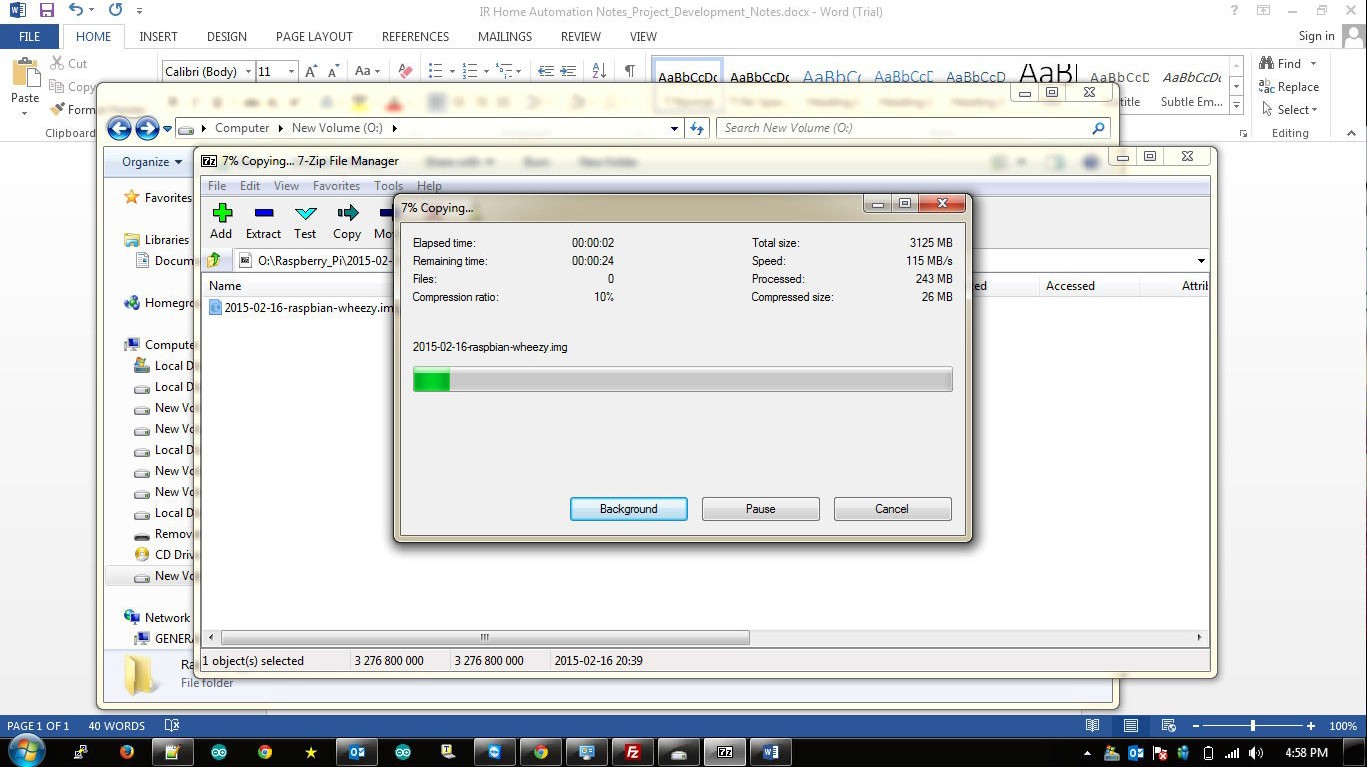
Getting started with Raspberry Pi(andWebIoPi framework)

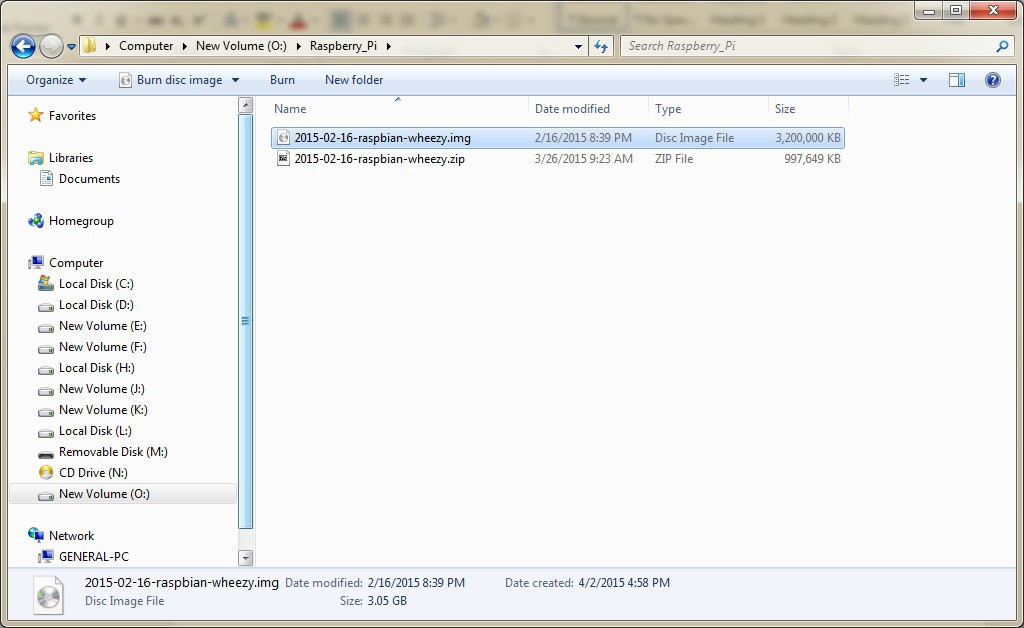
1. Installing the OS on the Raspberry Pi

Download the image file from theRaspberry Pi website. It’ll be a zip file as shown below:

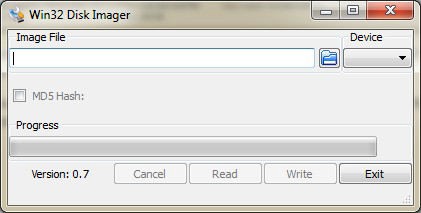


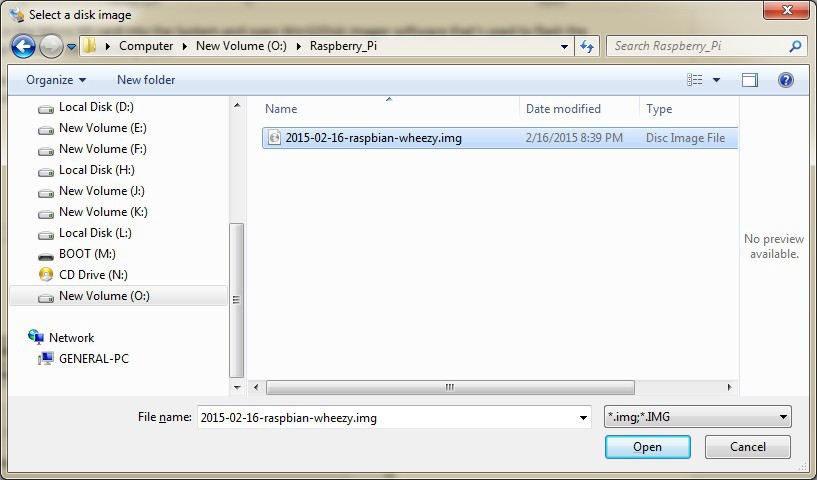
Unzip the file to reveal the image file with.I mg extension



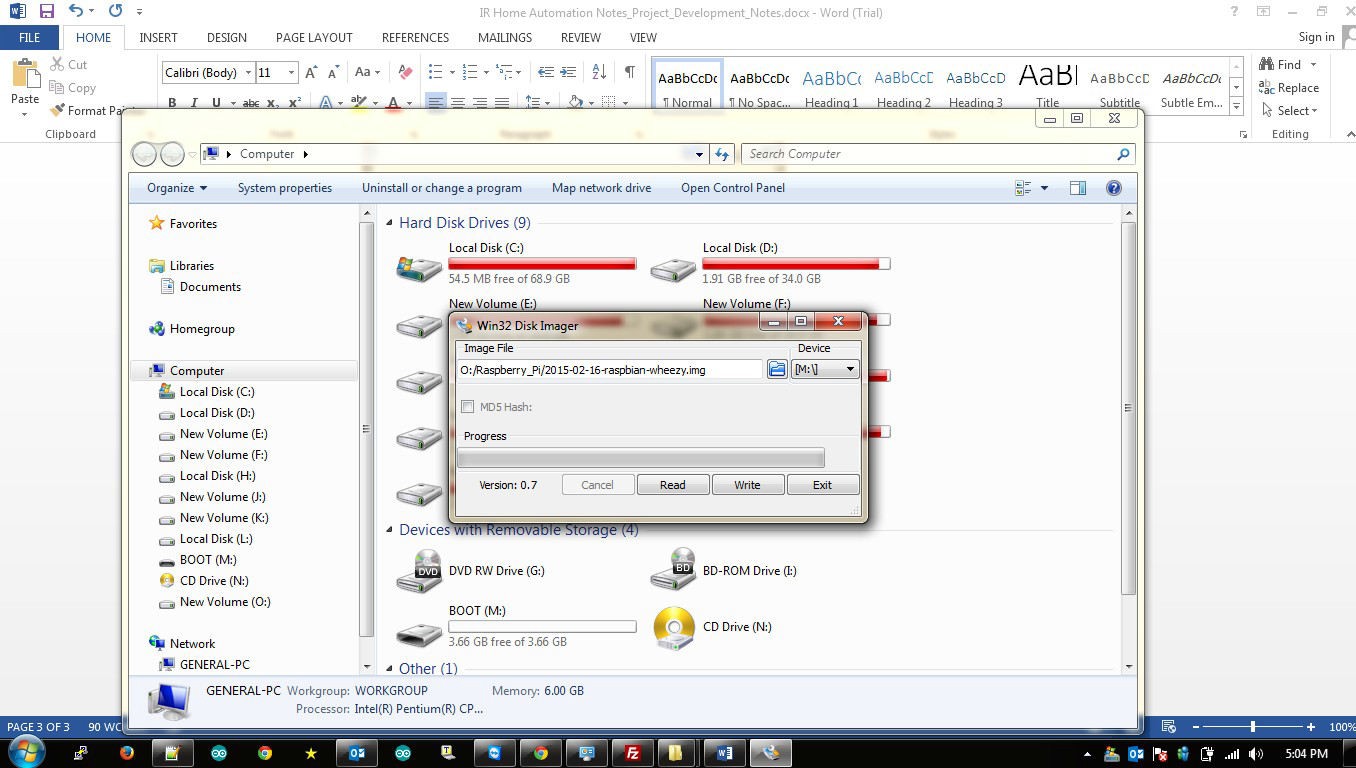


Plugin the MicroSD card into the System and open Win32 Disk image software that’s used to flash the Image file on to the Memory card.Inside theapplication,selecttheDestination driveastheSDcardandinthesourcefile,selecttherecentlyextracted.IMGfileasshownbelow:

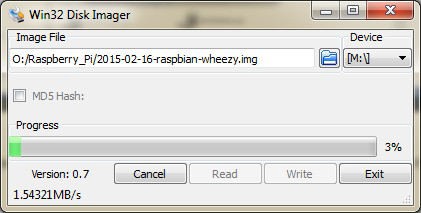




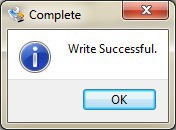
4



Now,hitthe Write button.TheApplicationwillbeginwritingtheOSontotheMemoryCardasshownbelow:



Oncedone,you’ll seethisconfirmationdialog box:



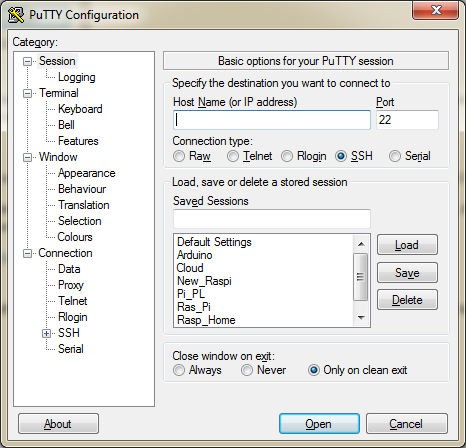
Now,plug intheSDcardbacktotheRaspberry Pi.Use aLANcabletobeabletoSSH intothe RaspberryPiifadisplay deviceisnotconnectedtothe RaspberryPi.

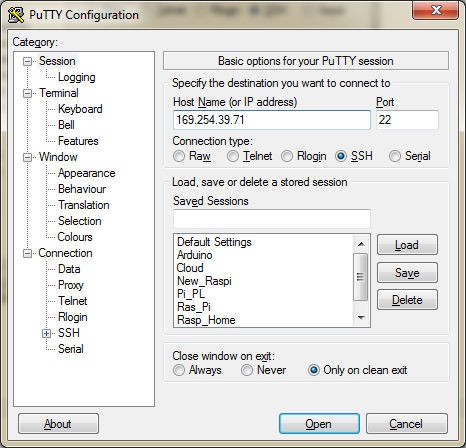
TheIPaddressassignedtomyRaspberry PiafterconnectingittotheNetworkis169.254.39.71.

InordertoaccesstheremoteterminaloftheRaspberryPi,weneedtousePuTTYsoftware,details ofwhichareshownbelow.

SSHingintotheRaspberryPi

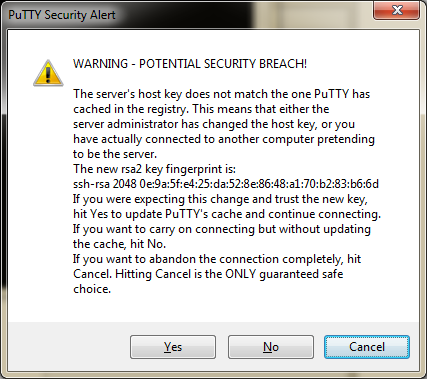
OpenPuTTYsoftwareandkeyintheIPaddressoftheRaspberryPiasdeterminedfromtheNetwork:





HitOpen

Ifyougetthiswarning,pleaseselecttheYesbutton.

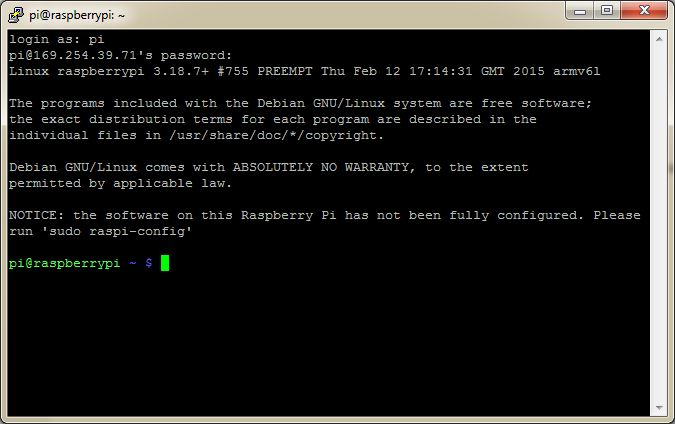




Loginwiththefollowingdefaultcredentials:

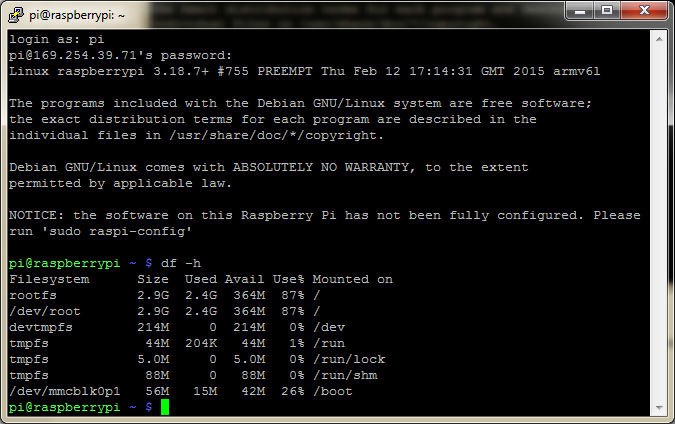
Username:piPassword:raspberry

Ifyoucorrectlyprovidedthecredentials,youshouldbeloggedinandshouldbeable tosee shellpromptasshownbelow:

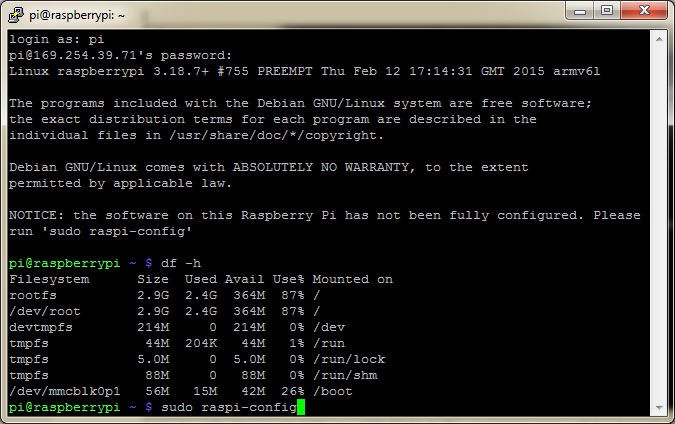


Ifthisisafirstinstallation,weneedtoruntheraspi-configcommandthat’llconfigure theRaspberry Piforfirstuse andalsoexpandtherootfile system.

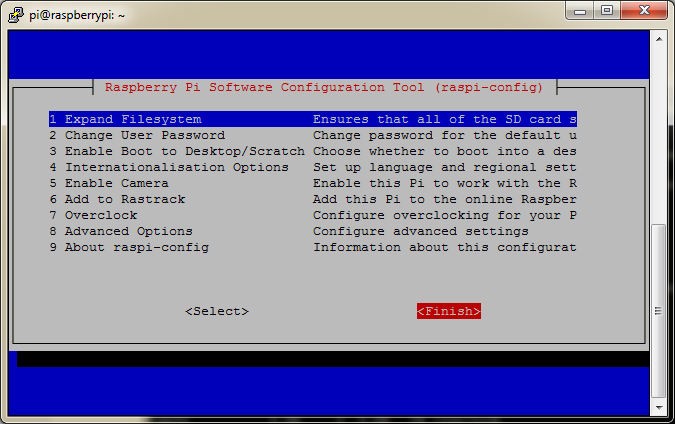
(



Note:howthe Sizeisonly2.9Gona4GBSDCard.Theremainingspacewillbe claimedafterrunningtheabovecommand)



SelectExpandrootfsandhitEnter



**Explanation of Algorithm and how it is been implemented**

import webiopi

import subprocess

import RPi.GPIO as GPIO\_PY

import os

import glob

import time

import sys

import smtplib

GPIO = webiopi.GPIO

PIR = 17 # GPIO pin using BCM numbering

from email.mime.text import MIMEText

motion = False

# setup function is automatically called at WebIOPi startup

def setup():

# set the GPIO used by the light to output

GPIO.setFunction(PIR, GPIO.IN)

# loop function is repeatedly called by WebIOPi

def loop():

# retrieve current datetime

global motion

if GPIO.digitalRead(PIR):

if motion == False:

print("Motion Detected")

SendEmail("Motion Detected!")

motion = True

else:

motion = False

webiopi.sleep(1)

# destroy function is called at WebIOPi shutdown

def destroy():

GPIO.digitalWrite(PIR, GPIO.LOW)

def SendEmail(MessageText):

#enter the e-mail account username between the quotes

smtp\_user = "raspi.arduino.bbb"

#enter the e-mail account password between the quotes

smtp\_pass = "raspberryarduino"

#sys.argv[1] is the 1st parameter that is passed to #this program and it contains the text for the body #of the e-mail

msg = MIMEText(MessageText)

#enter the target e-mail address between the quotes

msg['To'] = "captanwaar@gmail.com" #enter the e-mail account username between the quotes

msg['From'] = "raspi.arduino.bbb" #enter the message subject between the quotes

msg['Subject'] = "Motion Detection Alert" #enter the SMTP server URL or IP Address between the quotes

s = smtplib.SMTP\_SSL("smtp.gmail.com", 465)

s.login(smtp\_user,smtp\_pass)

s.sendmail(msg['From'], msg['To'], msg.as\_string())

s.quit()

@webiopi.macro

def shutDown():

**Information about the implementation of Modules**

**E-mail notification**

def SendEmail(MessageText):

#enter the e-mail account username between the quotes

smtp\_user = "raspi.arduino.bbb"

#enter the e-mail account password between the quotes

smtp\_pass = "raspberryarduino"

#sys.argv[1] is the 1st parameter that is passed to #this program and it contains the text for the body #of the e-mail

msg = MIMEText(MessageText)

#enter the target e-mail address between the quotes

msg['To'] = "captanwaar@gmail.com" #enter the e-mail account username between the quotes

msg['From'] = "raspi.arduino.bbb" #enter the message subject between the quotes

msg['Subject'] = "Motion Detection Alert" #enter the SMTP server URL or IP Address between the quotes

s = smtplib.SMTP\_SSL("smtp.gmail.com", 465)

s.login(smtp\_user,smtp\_pass)

s.sendmail(msg['From'], msg['To'], msg.as\_string())

s.quit()

**TESTING**

**ID Requirement Description Priority**

|  |  |  |  |
| --- | --- | --- | --- |
| **3.1** | Smart phone application control | The system shall be controlled By a smartphone application installed On the users’smartphone, Providing a Means for the users to Interact with the system. | 1 – Critical |
| **3.2** | Snapshot of guest satdoor | Photo images Shallbetakenand Stored By the system when doorbell is Rung. | 1 – Critical |
| **3.3** | Keep activities Log | The system activitylog shall differentiate between users when logging all in teraction sand Activitiesofthesystem. | 1 – Critical |
| **3.4** | Emergency power supply | The systemshallhave An Emergencypowersupply. | 4 – Low |
| **3.5** | Video monitoring | The system shal lallow register eduserst omonitoractivitiesvialivevideofeed On Smartphones. | 2 – High |
| **3.6** | Systemportability | Thesystemshallbea Portabledevicethat Thecustomercantake Withthemwhen They Move. | 2 – High |
| **3.7** | Smartphone Pairing | Thesystemshallpairuseractivitywith Aparticularsmartphoneto Keeplogs Of Whointeractswiththesystem. | 2 – High |
| **3.8** | Motionsensor | Thesystemshallhave Ashort-Rangemotionsensor.The Motionsensorwilltake A Pictureand Uploadittotheservereverytime Thesensoristriggered. | 5 – Future |
| **3.9** | Localstorage | Thedeviceshallhave An On-Boardstoragedevice. | 2 – High |
| **3.10** | Hardwaresecurity | Thedeviceshallbe Enclosed In Asecuredcase. | 2 – High |
| **3.11** | Z-Wave | Thesystemshallbecompatiblewithz-Wavewirelesscommunicationsprotocol. | 5 – Future |
| **3.12** | Lock/Unlock | Thesystemshallbeabletolockand Unlock the Doorremotely. | 5 – Future |
| **3.13** | Open source | The source code of the system be released underanopensourcelicense. | 1 – Critical. |
| **3.14** | Nonintrusive | The system shall not interfere with normal functionalities And operations Of the existing door And doorbell. | 1 -Critical |
| **3.15** | Account setup | The system shall provide account setup capability.The account setup will be responsible for pairing a Phone Toa Particular smart doordevice. | 2 – High |
| **3.16** | Video Log downloads | The system shall allow register edusers to download videos from the video Logon Thewebsiteinterface. | 4 – Low |

**Table1:Customerrequirements**

**RESULT**

****

**REFERENCES:**

**[1] Peter A. Blume, The LabVIEW Style Book. New jersey, United**

**states: Prentice Hall,2007**

**[2]http://www.engineersgarage.com/electroniccircuit/ infrared-ir-sensor .**

**[3]http://electroneecs-project.blogspot.in/ 2013-03-01-archive.html**

**(2002) The IEEE website. [Online]. Available: http://www.ieee.org/**

**[4] http://research.ijcaonline.org/volume119/number4/pxc3903700.pdf**

Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |
| --- |
| [**Home**](http://www.the-software-experts.com/index.php) |
| [**Contact us**](http://www.the-software-experts.com/e_dta-sw-team-contact.php) |
| [**Training Overview**](http://www.the-software-experts.com/e_dta-sw-training.php) |
| [**Training Access**](http://www.the-software-experts.com/moodle) |
| [**SW-Design**](http://www.the-software-experts.com/e_dta-sw-design.php) |
| [**SW Architecture**](http://www.the-software-experts.com/e_dta-sw-design-high-level.php) |
| [**SW Module Design**](http://www.the-software-experts.com/e_dta-sw-design-module.php) |
| [**Coding in C**](http://www.the-software-experts.com/e_dta-sw-coding.php) |
| [**Safe Coding in C**](http://www.the-software-experts.com/e_dta-sw-coding-safe.php) |
| [**Refactoring**](http://www.the-software-experts.com/e_dta-sw-refact.php) |
| [**Software Testing**](http://www.the-software-experts.com/e_dta-sw-test-principles.php) |
| [**Static Code Analysis**](http://www.the-software-experts.com/e_dta-sw-test-swb.php) |
| [**Software Inspection**](http://www.the-software-experts.com/e_dta-sw-test-inspection.php) |
| [**Software Module Tests**](http://www.the-software-experts.com/e_dta-sw-test-unit.php) |
| [**Integration Tests**](http://www.the-software-experts.com/e_dta-sw-test-int.php) |
| [**Test Documentation**](http://www.the-software-experts.com/e_dta-sw-test-doku.php) |
| [**SW-Documentation**](http://www.the-software-experts.com/e_dta-sw-doku.php) |
| [**SGML Principles**](http://www.the-software-experts.com/e_dta-sw-doku-principles.php) |
| [**Printing SGML**](http://www.the-software-experts.com/e_dta-sw-doku-printing.php) |