**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR nagar, MSRIT post, Bangalore-54

A Dissertation Report on

LANDSLIDE DETECTION SYSTEM

Submitted by

VISHAL KUMAR SAH 1MS12CS133

SUMEET SEN 1MS12CS116

RAKESH ROSHAN 1MS12CS088

SUPRATIK SHARMA 1MS12CS118

*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 internet of things (iot) refers to the ever-growing network of physical objects that feature an ip address for internet connectivity, and the communication that occurs between these objects and other internet-enabled devices and systems, examples of objects that can fall into the scope of internet of things include connected security systems, thermostats, cars, electronic appliances, lights in household and commercial environments, alarm clocks, speaker systems, vending machines and more.  Businesses can leverage iot applications to automate safety tasks (for example, notify authorities when a fire extinguisher in the building is blocked) to performing real-world testing using networked cameras and sensors to detect how customers engage with products.

Landslides cause significant damages to civil infrastructure. Over the years, methods and technologies have been proposed to determine the risk of landslides and to detect hazardous slope movements. There have been increasing interests in developing and landslide monitoring systems to observe movements using sensors installed on the slope. Although providing accurate data, many landslide monitoring systems are not operating in an automated fashion and lack the ability to analyze the collected data in a timely manner.

Use of technology in different areas to get various benefits in itself research of big area. Use of vibration sensor in the landslide detection is not a new trend. This is an IOT project to detect landslides by monitoring various sensor’s data real time and uploading the data on cloud in order to alert people in case of any ongoing landslide situation or a probable landslide situation in landslide prone areas. This is a microcontroller based (using Raspberry pi soc) project which is interfaced with various sensors like vibration sensor, moisture sensor, pressure sensor and the soc is configured to connect to an existing cloud framework provided by a free and open source cloud infrastructure ubidots .

#### Contents

1 **INTRODUCTION**

* 1. General Introduction……………….
  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. **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**
2. **REFERENCES**

**INTRODUCTION**

***General Introduction:***

This project is using a Raspberry Pi board along with various sensors to detect the various landslides in the hilly regions. The data is collected in real time and the data is processed to conclude if a landslide is about to occur in the particular region, so that a warning can be conveyed to the people living in the nearby regions.

***Problem Statement:***

This is an IOT project to detect landslides by monitoring various sensors’ data real time and uploading the data on cloud in order to alert people in case of any ongoing landslide situation or a probable landslide situation in landslide prone areas. This is a microcontroller based (using Raspberry pi soc) project which is interfaced with various sensors like vibration sensor, moisture sensor, pressure sensor and the soc is configured to connect to an existing cloud framework provided by a free and open source cloud infrastructure NIMBUS.

***Objective of the Project:***

* To facilitate the conversion of raw sensor data to a form that is more useful.
* To use the SOC Raspberry Pi
* To provide a accessory that enables users to analyze the data in Cloud in real time.
* To use Ethernet technology to transfer data to cloud
* To understand the uses of sensors and use them effectively.
* To use the latest boom of IOT market to provide safety measures in landslide prone areas.

***Project Deliverables:***

* Installation of Raspberry Pi and running first program.(Done)
* Calibration and interfacing soil moisture sensor and processing soil moisture sensor data.(1/10/2015) (Done)
* Calibration and interfacing vibration sensor and processing vibration sensor data.(4/10/2015) (Done)
* Combining the two sensor data and troubleshooting.(10/10/2015) (Done)
* Configuring and connecting Ethernet module ensuring transmission of data to the cloud. (15/9/2015) (Done)
* Creating web app for analysis of data.(21/10/2015) (Done)
* Troubleshooting and delivery of first prototype.(28/10/2015) (Done)
* Delivery of final project (10/12/2015)

***Current Scope:***

The current project module is going to be used to measure small frequencies in the movements of land in hilly regions. The project is designed to correctly provide a range of data to be analyzed and reported upon in due course of time. The data is also being sent to the cloud for storage and measurement purposes. This data can be useful at later points in time if a prediction is to be made upon for future landslide locations and durations.

***Future Scope:***

* This project can be implemented at a larger scale by analyzing the data sets during the rainy or monsoon months of the year actively to detect landslides.
* If the sets represent a probable set where a possibility of landslide is imminent, a warning can be sent immediately to the nearby location, thus preventing the public from using the danger zone areas.
* A lot of lives can be saved by early warnings

**PROJECT ORGANIZATION**

***Software Process Models;***

The various software module used in the project are-

* Putty
* Vncserver
* Vncviewer
* Win32diskimager
* Ipscan
* Ubidots cloud package
* Python tools

**Putty-**

This software uses a ssh client to connect to the terminal of raspberry pi to install the vnc server.

**Vnc server and viewer-**

Vnc server module is installed on the raspberry pi.it runs a server on the pi which sends out data to the Ethernet so that the display of the pi can be viewed from the monitor of the laptop where the vnc viewer is installed.

**Win32diskimager-**

This software module is used to extract the .img file of the pi os to burn to the sd card.

**Ipscan-**

This software scans the devices connected to the local network and the ip addresses allocated to them.

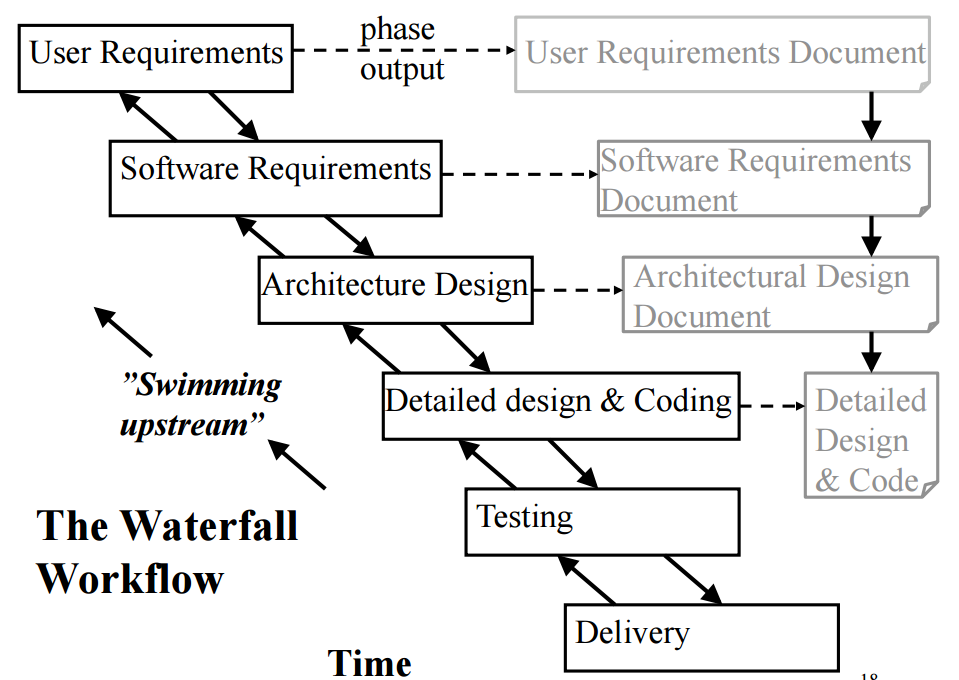
**Python-**

Python library is installed to run python scripts to sense the GPIO pins of the pi.

The software models are interdependent as the ubidots module depends on the python libraries to sense the pi.

The python module after sensing the data pushes it onto the cloud

Some more dependencies come when we talk about the cloud as the cloud may run into problems like memory leak etc.

****

***Roles and Responsibilities:***

***The roles shared by the group members for the project:***

Program Planner

Program Planner

Status and Tracking Reporting

Issues Management

Communication Coordinator

Software Methodology Support

Infrastructure and Technical

Facilities Administrator

Risk Manager

Project Deliverables

Resource administrator

**SOFTWARE REQUIREMENT SPECIFICATIONS**

***1. External Interface Requirements***

* 1. **User Interfaces**

First page would be login page. This page will allow the user to login as manager or volunteer.

The volunteer side mobile application will consist of three buttons start, end and delivered. On pressing the start button, the information about the location of the volunteer is continuously uploaded to the cloud and this uploading will stop on pressing the stop button. Deliver button is used to increment count of vaccinated children.

The manager side mobile application will have track and buttons. Track button is used to monitor the location of volunteer.

* 1. **Hardware Interfaces**

The mobile application takes the data that is stored in cloud by raspberry pi. Raspberry pi makes use of Raspian OS. A python code is written to collect various data from the sensors and upload it to the cloud. The various sensors used include

* + - * 1. Light sensor (BH1750FVI)
        2. Temperature (DS18B20)
        3. Accelerometer (MPU6050)

The digital outputs generated by these sensors are uploaded into the cloud using a python code. The cloud platform used here is ‘CARRIOTS’.

The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

* 1. **Software Interfaces**

Yousup API is installed in the Raspian OS which is used to send whatsapp message to the volunteers when abnormal conditions are found in vaccination box.

The physical location of volunteers is tracked continuously by using maps to ensure that they cover all the assigned areas.

The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while the communication between the database and the mobile application consists of only reading operations.

* 1. **Communication Interfaces**

The communication between the different parts of the system is important since they depend on each other.

Internet connection and a web browser are required in order to access the data from the cloud and analyze it.

1. ***Functional Requirements***
   1. **Functional Requirement 1.1:**

ID: FR1

TITLE: Download mobile application

DESC: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download.

DEP: None

* 1. **Functional Requirement 1.2:**

ID: FR2

TITLE: User log-in - Mobile application

DESC: The user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically. If the user logs in as administrator, he gets full access to volunteer’s location and activities

DEP: FR1

* 1. **Functional Requirement 1.3:**

ID: FR3

TITLE: Get the sensor reading

DESC: The various data readings such as temperature (in Kelvin), light intensity (in luminous) and equilibrium (angle of rotation) inside the vaccine box are obtained using python code.

DEP: NONE

* 1. **Functional Requirement 1.4:**

ID: FR4

TITLE: Upload the sensor reading to cloud

DESC: Sensor data reading are uploaded to cloud using python code. The cloud platform used is ‘CARRIOTS’.

DEP: FR3

* 1. **Functional Requirement 1.5:**

ID: FR5

TITLE: Check for abnormal conditions

DESC: If temperature is not within range specified by WHO or if there is light exposure or if there is too much of shaking, the manager as well as volunteer are notified so that they can take appropriate actions to preserve the vaccines.

DEP: FR4

* 1. **Functional Requirement 1.6:**

ID: FR6

TITLE: Send GPS coordinates to cloud

DESC: The volunteer’s mobile continuously sends GPS coordinates like longitude and latitude to cloud.

DEP: FR2

* 1. **Functional Requirement 1.7:**

ID: FR7

TITLE: Track the volunteer

DESC: The manager side application receives the GPS coordinates from cloud and traces it to ensure that volunteers cover all the assigned areas.

DEP: FR1, FR6

1. ***Software System Attributes***
   1. **Reliability**

The reliability that the system gives precise sensor reading. The cloud platform used should be reliable, secure and fast. There should be proper internet connectivity to cloud.

* 1. **Availability**

The system should be available all the time during vaccination program.

* 1. **Security**

The system should be secure enough to prevent the external attacker from entering malicious data which may affect the normal functioning of the software and generate unnecessary exception conditions.

* 1. **Portability**

The system is portable across different operating system.

* 1. **Maintainability**

The code written is modular, simple and easy to read. Hence the maintenance required is less. The application need not be updated frequently.

* 1. **Performance**

The performance of software should be optimized to ensure that the resource utilization is less and efficiency is very high and we get proper results almost instantly with no delays.

1. ***Performance Requirements***

The performance of the system depends upon the reliability and efficiency of sensors ie how fast the sensors produce readings.

It also depends upon cloud platform we choose ie higher the capacity of the cloud, higher the performance. Also we require high speed internet for instant uploading of data values.

1. ***Database Requirement***

The database should be able to store large number of readings. The database should not crash when the number of hits is high. The database should be flexible and querying should be fast.

1. ***Design Constraints***

The sensor may produce incorrect result or burn when the voltage supplied is varied, so we must ensure that voltage supplied should not exceed specific values. The database may crash when there is large number of querying to it, so the database should be created so that it is capable to store large number of data. The cloud platform may become slow when there is large number processing in it, so the cloud platform should be selected carefully.

1. ***Other Requirements***

Knowledge of android programming, python programming and connection of sensors to raspberry pi.

**DESIGN**

***INTRODUCTION:***

This section describes about the various modules involved in the project and the details about each of them and the design of the project.

**NUMBER OF MODULES-**

The project consists of mainly three modules ie

1. The raspberry pi soc as the computing device

2. The sensors as the data collecting device

3. A cloud component (ubidots cloud)

**MODULE DESCRIPTION:**

**1. COMPUTER SOC MODULE**

The project uses raspberry pi 2 model b+ as its main computing device. Raspberry pi is a system on a chip (soc) which is capable of general purpose computing and is capable of running computation and graphics intensive software and is capable to support java.The operating system installed on raspberry pi is the official Raspbian os provide by the Raspberry pi community.

Raspbian is also a fully fledged operating system which is a slightly modified distribution of Linux optimized to work on the raspberry pi.

**The various hardware components and capabilities of raspberry pi are as follows-**

1GB RAM, Quad Core, Speed 900Mhz. these single board computers provide you with double the ram and a much faster processor. The credit-card sized computer is capable of many of the things that a desktop PC does, like spreadsheets, word-processing and playing high-definition video and games. It can run several flavors of Linux (and even Windows 10 free-of-charge) and is being used to learn how to program worldwide

**Features:**

* Broadcom 900 MHz BCM2836 ARMv7 Quad Core Processor SoC
* Broadcom Video Core IV GPU 1 GB RAM
* 4 x USB2.0 Ports with up to 1.2A output
* Expanded 40-pin GPIO Header Video/Audio Out via 4-pole 3.5mm connector
* full size HDMI
* Raw LCD (DSI display port) CSI camera port
* Storage: microSD
* 10/100 Ethernet (RJ45)
* Low-Level Peripherals: 40 pin GPIO Header
* 27 x GPIO UART I2C bus SPI bus with two chip selects +3.3V +5V Ground
* Power Requirements: 5V @ 600 mA via MicroUSB or GPIO Header Supports Windows 10, Debian GNU/Linux, Fedora, Arch Linux, RISC OS and More!



Fig: Raspberry pi 2 model B

**2. SENSOR MODULE**

Vibration sensor module alarm Motion sensor module vibration switch SW-420

The Vibration module based on the vibration sensor SW-420 and Comparator LM393 to detect if there is any vibration that beyond the threshold. The threshold can be adjusted by the on-board potentiometer. When this no vibration, this module output logic LOW the signal indicate LED light,And vice versa.

* Uses: For a variety of shocks triggering, theft alarm, smart car, an earthquake alarm, motorcycle alarm. This module when compared with normally open shock sensor module, shock triggered much longer can drive relay module
* The use of the company's production of SW-420 normally closed type vibration sensors. comparator output signal clean wave well, driving ability, 15mA
* rated voltage and 3.3V-5V output: digital switching output (0 and 1) a bolt-hole for easy installation
* Small Board PCB dimensions: 3.2cm x 1.4cm. using wide LM393 voltage comparator
* Module description: the product when it is not shock, vibrate switch is closed on-State, output output low level, the green indicator light is on; When vibration, vibration switches disconnected moments, output output line, the Green led is not lit; the output can be directly connected to the microcontroller, by single-chip computer to detect high or low level, to detect whether there is vibration, alarm function

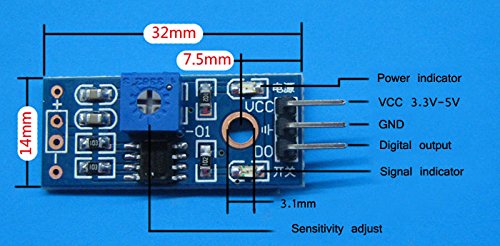


Fig:**Vibration sensor module alarm Motion sensor module vibration switch SW-420**

# DHT11 Temperature and Humidity Sensor

**Description-**

This DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Digital Output Single Wire Output Stable & long term accuracy Full range temperature compensated Relative humidity and temperature measurement Calibrated digital signal outstanding long-term stability pins packaged and fully interchangeable DHT11 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability.

Its sensing elements are connected with 8-bit single-chip computer. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programs in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package.

Technical parameters: Power supply - 3-5.5V DC Output signal- digital signal via single-bus Sensing element- Polymer resistor Measuring range- humidity 20-90%RH; temperature 0-50 Celsius Humidity hysteresis- +-1%RH Long-term Stability- +-0.5%RH/year Sensing period- Average: 2s



# Fig: DHT11 Temperature and Humidity Sensor

**3. CLOUD MODULE**

We use ubidots as our cloud provider which provides free setup to store sensor data which provides easy ways of accessing the data both for storing and retrieving, with good tutorials and code-examples and a nice interfaces.

Tracking sensor-data only makes sense if the data is being stored in a central place for further action to analyze and visualize the data, drive further actions from the data and probably even perform some machine learning tasks to get deeper insights.

This can be done locally with your own server and file-system which most probably would be something hadoop-based in these days with a stack of spark-services on top. But: it will be a huge effort to build a backend-service like this and operate it on a daily basis.

Fortunately a lot of cloud-based IoT services have come to life recently. There are so many that it is even hard to create a list of them because new services are going live every weak – just recently sparkfun and adafruit launched their own tracking-services for instance

I decided not to use thingspeak in my project, but i think it deserves a close look if you are considering a project or even a business in the IoT field – especially because you could prototype with the cloud-service and then establish your own backend based on the Open-Source stack – that can be a very promising approach.

[ubidots](http://ubidots.com) cloud-based IoT Service that is a little bit more business focussed than thingspeak but in the same time less arrogant than xively…

In fact ubidots is a very nice and reliable platform with a well documented API, nice tutorials and really amazing features for inspecting and visualising the data. There are different plans starting with a free-service that gives enough room and features to really start something + 3 levels of paid services starting with 5$/month. For 100$ per month businesses can get a customizable service under their own domain-name + advanced features. Their service is competent and available and the dashboard they offer is probably best in class. Just recently they for instance offered a GPS-trace app (GPS location was already available) that gives you a map with a route based on the latest GPS-coordinates you tracked. Pretty amazing.

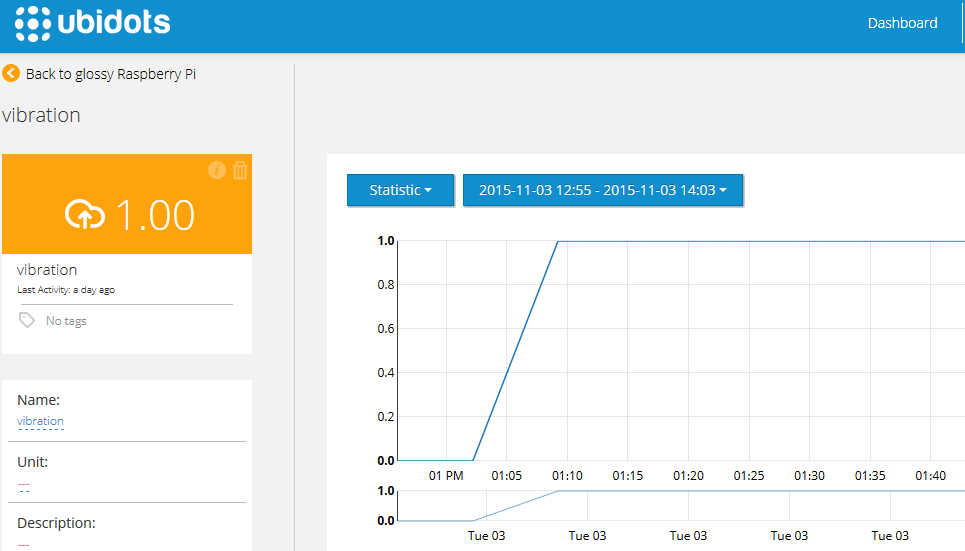
The downside: ubidots is closed source and you are locked into their service with your data.

Fig: snapshots of ubidots cloud interface

**Architecture Design**

We use sensors ie. Vibration sensor, humidity sensor and the data from each of the sensors is uploaded to the cloud in real time via raspberry pi soc.

The sensors are interfaced with the soc where the analog signal collected by the sensors is converted into digitally interpretable data. This digital data is the output from the sensors which act as an input for the raspberry pi board.

The board sends this data to the ubidots cloud. Our board is configured to connect to the ubidots cloud using the proprietary software provided by ubidots which runs on the Raspian OS installed on the raspberry pi soc.

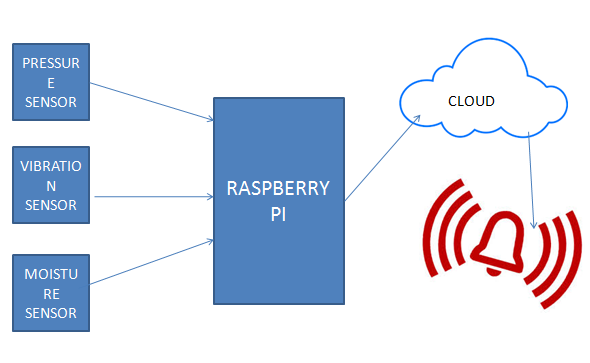


Fig: architecture overview diagram

**Dependencies between modules, hierarchy and partitioning of the software modules**

**4. SOFTWARE MODULES**

The various software module used in the project are-

* Putty
* Vncserver
* Vncviewer
* Win32diskimager
* Ipscan
* Ubidots cloud package
* Python tools

**Putty:**

This software uses a ssh client to connect to the terminal of raspberry pi to install the vnc server

**Vnc server and viewer**

Vnc server module is installed on the raspberry pi.it runs a server on the pi which sends out data to the Ethernet so that the display of the pi can be viewed from the monitor of the laptop where the vnc viewer is installed.

**Win32diskimager**

This software module is used to extract the .img file of the pi os to burn to the sd card

**Ipscan**

This software scans the devices connected to the local network and the ip addresses allocated to them.

**Python**

Python library is installed to run python scripts to sense the GPIO pins of the pi

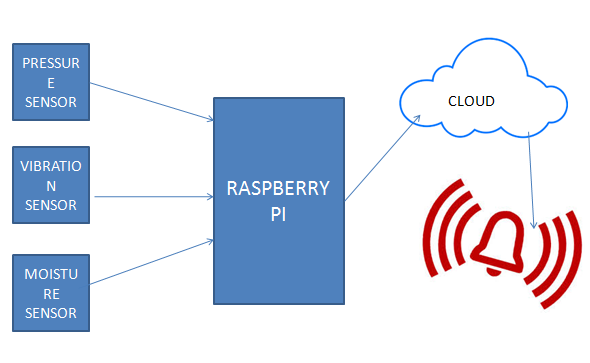
The software modules are interdependent as the ubidots module depends on the python libraries to sense the pi.

The python module after sensing the data pushes it onto the cloud

Some more dependencies come when we talk about the cloud as the cloud may run into problems like memory leak etc.

**IMPLEMENTATION**

**Tools Introduction:**



***Technology Introduction:***

***Details of Soc:***

* NAME: RASPBERRY PI 2 MODEL B
* 900MHz quad-core ARM Cortex-A7 CPU
* 1GB RAM
* Like the (Pi 1) Model B+, it also has:
* 4 USB ports
* 40 GPIO pins
* Full HDMI port
* Ethernet port
* Combined 3.5mm audio jack and composite video
* Camera interface (CSI)
* Display interface (DSI)
* Micro SD card slot
* VideoCore IV 3D graphics core

Raspberry Pi has become the most popular single-board computer on the market and spawned many imitators, but none with the rich community that has grown organically around the Raspberry Pi.

The Raspberry Pi B+ and Pi 2 both come with the same Videocore GPU as before and in our tests there was a small improvement in FPS (Frames Per Second) for the Raspberry Pi 2 largely thanks to the increased RAM present on the board. Our last test was file transfer speeds via Ethernet, for this we used scp to copy a 692MB Big Buck Bunny video file to each Pi. On the B+ we saw an average of 3.8MB/s and on the Pi 2 we saw 4.6MB/s, which is an 0.8MB/s speed increase. The Raspberry Pi Foundation have released an updated Raspbian image which includes the ARMv7 kernel image necessary to use the new CPU. Applications written for the original Raspberry Pi are fully compatible with the Raspberry Pi 2, though -- building upon the rich projects that have been written since the initial launch of the Raspberry Pi.

The technical specifications are:

SoC: Broadcom 2836  
CPU: Quad-core ARM7 800MHz  
GPU: Videocore IV 250MHz  
Memory: 1GB  
GPIO: 40pin  
Ports: 4x USB 2.0, 100BaseT Ethernet, HDMI, MicroSD card  
Size: 85.60 × 56.5mm (about 3.2 x 2.1-inch)

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

|  |  |
| --- | --- |
| START  Run Raspberry Pi  GET READINGS FROM SOIL MOISTURE SENSOR  GET READING FROM GEOPHONE SENSOR  TRANSMIT DATA THROUGH ETHERNET MODULE TO THE CLOUD  RUN THE WEB APPLICATION  PERFORM DATA ANALYSIS ON REAL TIME  WARN FOR LANDSLIDE OCCURRENCE FOR THE AREA  STOP |  |

***Information about the implementation of Modules:***

Soil Moisture Sensor

The Soil Moisture Sensor is used to measure the volumetric water content of soil. This makes it ideal for performing experiments in courses such as soil science

Specifications

* Range: 0 to 45% volumetric water content in soil (capable of 0 to 100% VWC with alternate calibration)
* Accuracy: ±4% typical
* Resolution: 0.1%  
  (LabQuest 2, LabQuest, LabQuest Mini, Go!Link, LabPro)
* Power: 3 mA @ 5VDC
* Operating temperature: –40°C to +60°C
* Dimensions: 8.9 cm × 1.8 cm × 0.7cm (active sensor length 5 cm)

Geophones

Geophones are usually the sensor of choice in most mining applications, because of the low cost, large bandwidth and excellent reliability. IMS routinely manufactures two kinds of geophones, with natural frequencies of 4.5 Hz and 14 Hz. The 4.5 Hz geophone has a usable frequency bandwidth of between 3 Hz and 2000 Hz but must be installed to within two degrees of its pre-set orientation with respect to the vertical. The 14 Hz geophone is omni-directional and can be installed at any angle, with a usable frequency bandwidth of between 8 Hz (-3dB point) and 2000 Hz.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Low and Medium Frequency Geophone Specifications** | | | | |
| Type | LGT-4.5 | LGT-10 | LGT-14 | LGT-28 |
| Natural Frequency (Hz) | 4.5±10% | 10 ± 5% | 14 ± 5% | 28 ± 5% |
| Open Circuit Damping | 0.6 ±10% | 0.271 ± 5% | 0.2 ± 5% | 0.6 ± 5% |
| Damping with Shunt Resistance |  | 0.6 ± 5% | 0.6 ± 5% |  |
| Intrinsic Voltage Senditivity (V/cm/s) | 0.288 ± 10％ | 0.288 ± 5％ | 0.288 ± 5％ | 0.31 ± 5％ |
| Sensitivity with Shunt Resistance (V/cm/s) |  | 0.227 ± 5％ | 0.182 ± 5％ |  |
| Coil Resistance | 375 ± 5％ | 375 ± 5％ | 375 ± 5％ | 385 ± 5％ |
| Close Circuit Resistance (Ω) |  | 296 ± 5％ | 236.5 ± 5％ |  |
| Harmonic Distortion | ≤0.3% | ≤0.2% | ≤0.2% | ≤0.2% |
| Typical Spurious Frequency (Hz) | ≥180 | ≥200 | ≥250 | ≥400 |
| Moving Mass (g) | 11.3 | 11.3 | 11.6 | 10.2 |
| Coil Excursion P-P | 4 | 2 | 2 | 1.6 |
| Maintains Fn Specifications to Tilt Angle | ≤10 | ≤20 | ≤20 | 90 |
| Core Unit Diameter (mm) | 25.4 | 25.4 | 25.4 | 26 |
| Core Unit Height (mm) | 36 | 32 | 32 | 32 |
| Core Unit Mass (g) | 74 | 74 | 74 | 95 |
| Operating Temperature(℃) | -40 to +90 | -40 to +90 | -40 to +90 | -40 to +90 |
| Warranty (year) | 1 | 3 | 3 | 3 |
|  |  |  |  |  |

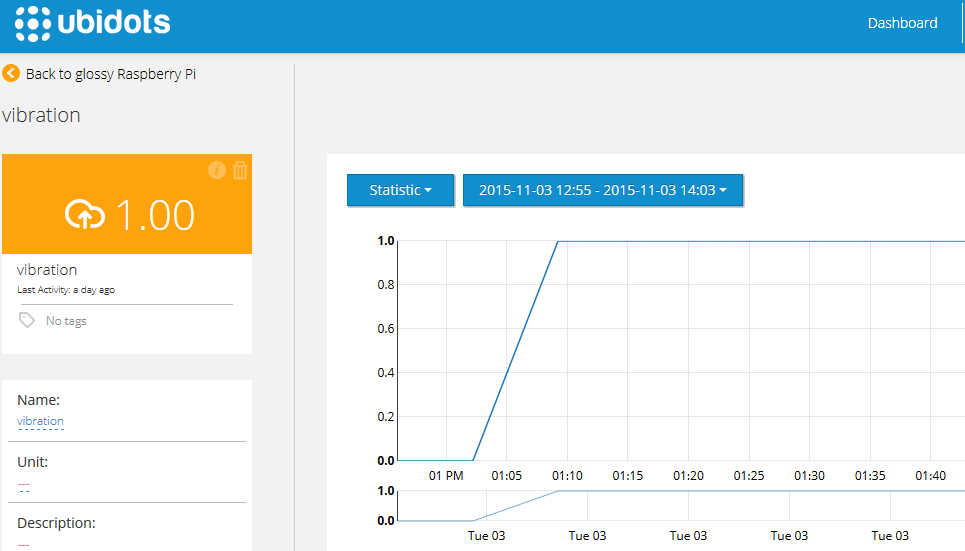
***Sample Data:***

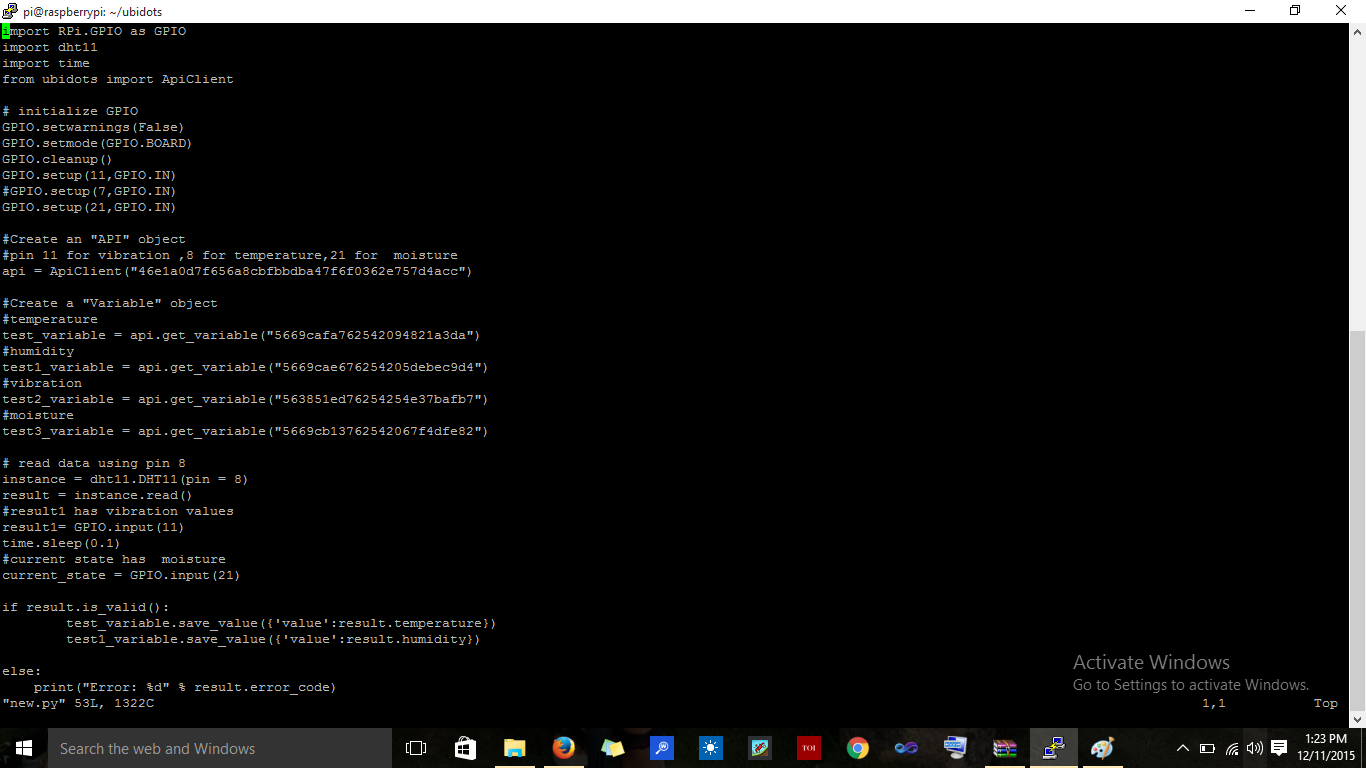
Vibration: 3-4+ richter scale

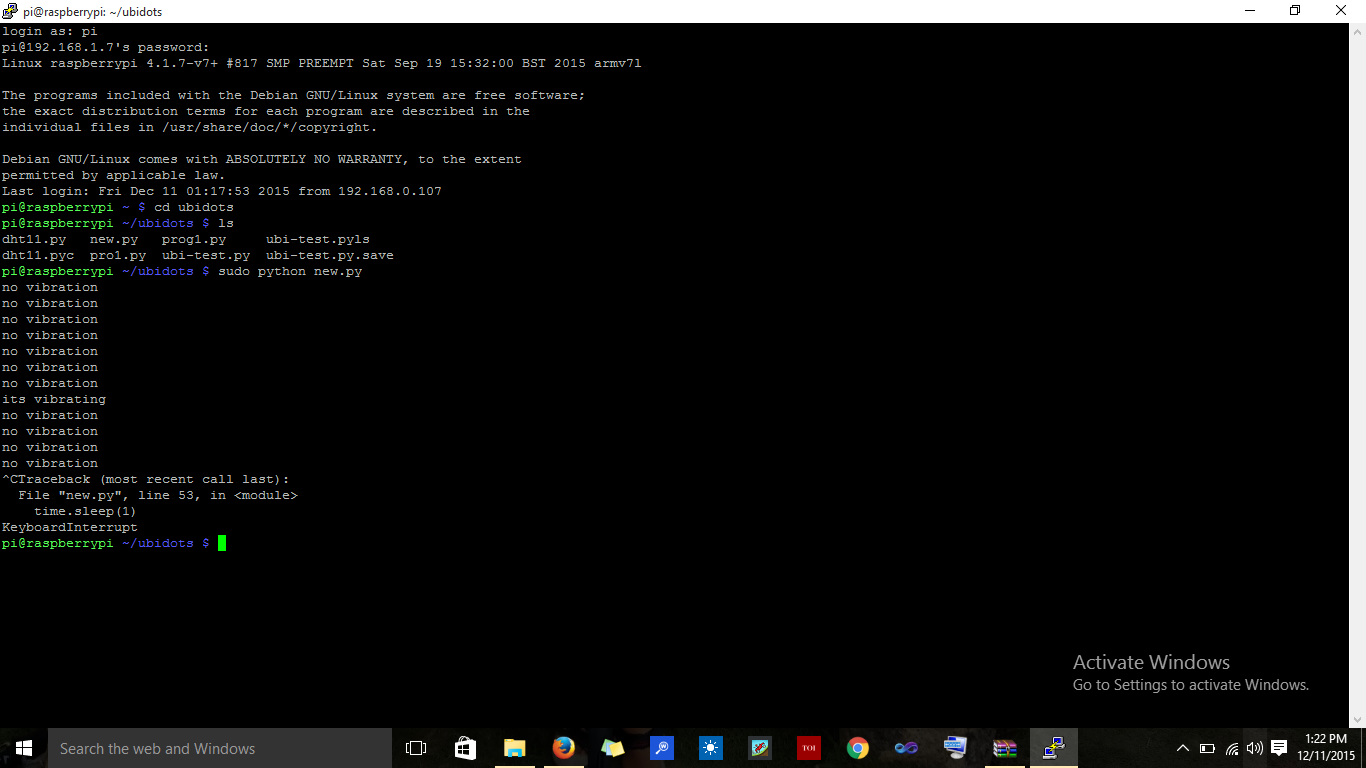
Available Water Capacity: 1.9-2.7ft

**TESTING**

**Results and Snapshots:**



****

****

****

****

**CONCLUSION & SCOPE FOR FUTURE WORK**

This project can be implemented at a larger scale by analyzing the data sets during the rainy or monsoon months of the year actively to detect landslides.

If the sets represent a probable set where a possibility of landslide is imminent, a warning can be sent immediately to the nearby location, thus preventing the public from using the danger zone areas.

A lot of lives can be saved by early warnings.

On a large scale data production can be examined and processed for landslide predictions.

**REFERENCES**

[1]. http://ubidots.com/

[2]. http://www.govtech.com/fs/New-Era-in-Firefighting-Leverages-the-Internet-of-Things.html

[3]. http://www.forbes.com/sites/federicoguerrini/2015/06/29/firefighters-and-the-internet-of-things/

[4]. http://www.jbrehm.com/blog/2015/6/11/smart-landslide-merging-iot-and-big-data-for-fire-safety

[5]. http://www.nist.gov/el/fire\_research/201506\_smart\_landslide\_roadmap.cfm

[6]. IEEE Journals and papers - http://ieeexplore.ieee.org/

[7]. http://blog.ubidots.com/topic/raspberry-pi