VENTILATION CONTROL SYSTEM FOR UNDERGROUND FACILITIES

CONTROL SYSTEM FOR UNDERGROUND FACILITIES VENTILATION

ABSTRACT

* Nowadays the construction of underground facilities is popular in order to attack or for defense and the use of existing natural caves and artificial underground facilities for military purposes.
* Tunnels can be used to undermine [fortifications](https://en.wikipedia.org/wiki/Fortification) and slip into territory, while it can strengthen a [defense](https://en.wikipedia.org/wiki/Defence_(military)) by creating the possibility of ambush, counterattack and the ability to transfer troops from one portion of the battleground to another unseen and protected. Also, tunnels can serve as shelter for [combatants](https://en.wikipedia.org/wiki/Combatants) and [non-combatants](https://en.wikipedia.org/wiki/Non-combatants) from enemy attack.
* It can be sometimes linked with [urban warfare](https://en.wikipedia.org/wiki/Urban_warfare), as tunnels are often found in urban areas and [cities](https://en.wikipedia.org/wiki/City), although urban warfare as a whole usually dominates any tunnel considerations.
* Tunnels, due to their nature, restrict [fields of fire](https://en.wikipedia.org/wiki/Field_of_fire_(weaponry)) and thus any troops in a tunnel usually only have a few areas exposed to fire or sight at any one particular time. Also they can be part of an extensive [labyrinth](https://en.wikipedia.org/wiki/Labyrinth) and have [cul-de-sacs](https://en.wikipedia.org/wiki/Cul-de-sac) as well as reduced [lighting](https://en.wikipedia.org/wiki/Lighting) that can create a closed-in night environment.
* Ventilation for Tunnels is also an important factor. So the precise and intelligent control system of Ventilation is essential.
* In this Research, a propose system of Ventilation for Tunnel will have been designed and developed.
* This System is made up of four components: Ventilation Fans and controller circuit, Raspberry Pi, Wireless Router, Mobile Phone Control apps. The goal of this research is to develop mobile phone and computer controlled ventilation system for underground facilities.

Keywords: Ventilation control system, Raspberry Pi as Embedded System, Industrial Control and Automation, Building Automation.

ABSTRACT

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

LIST OF EQUATIONS

LIST OF FIGURES

LIST OF TABLES

CHAPTER TITLE

1. INTRODUCTION

1.1 Motivation

* 1. Objective of Thesis
  2. Organization of Thesis

1. LITERATURE REVIEW AND BACKGROUND THEORY

2.1 Control System Theory

2.2 Control circuit and Ventilation fans

2.3 Raspberry Pi

2.4 Android App Development

3. PROPOSED SYSTEM

3.1 Components of Proposed System

3.2 Software and Hardware Requirements

3.3 Proposed Architecture

4. CONCLUSION AND FUTURE WORKS

4.1 Conclusion

4.2 Future works

REFERENCE

AUTHOR’S PUBLICATION

1. INTRODUCTION

-Important of Tunnel warfare (to get advantages over, to use nature as our defense system, mountain).

-Important of Ventilation system (not only for human comfort but also for important machines: moisture, oxidizing, decay, humidity)

-To add intelligence to the system, Rpi is used as a server.

-To use advantage of android phones, to alert operator and to monitor, portable

-Also computers can be used as Control panel of the system

-both wireless communication and wired communication are used

-For each platform, different programming languages are used for compatibility and operation ability.

* 1. Motivation

Reliability, automation, monitoring, portable, management.

* 1. Objective of Thesis

The objective of thesis is

1. to create Control System for Underground Facilities Ventilation.
2. To monitor continuously
3. To alert when threshold is reach
4. To support automation (eg. Automatic operation of reserve devices)
   1. Organization of Thesis

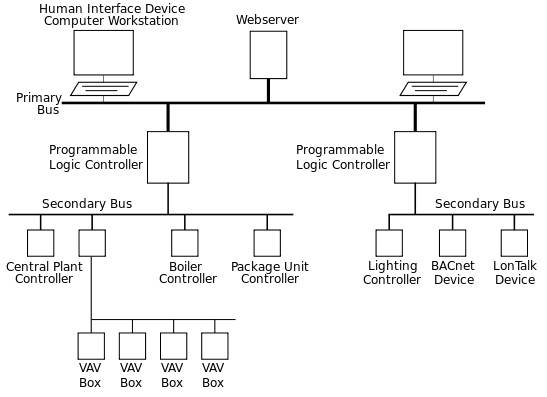
– Theories and devices introduction

**Building automation** is the automatic centralized control of a building's [heating, ventilation and air conditioning](https://en.wikipedia.org/wiki/HVAC), lighting and other systems through a [Building Management System](https://en.wikipedia.org/wiki/Building_Management_System) or Building Automation System (BAS). The objectives of building automation are improved occupant comfort, efficient operation of building systems, and reduction in energy consumption and operating costs.

Building automation is an example of a [distributed control system](https://en.wikipedia.org/wiki/Distributed_control_system) - the [computer networking](https://en.wikipedia.org/wiki/Computer_networking)of electronic devices designed to monitor and control the mechanical, security, fire and flood safety, lighting (especially emergency lighting), [HVAC](https://en.wikipedia.org/wiki/HVAC) and humidity control and ventilation systems in a building.[[1]](https://en.wikipedia.org/wiki/Building_automation#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Building_automation#cite_note-hvac-2)

BAS core functionality keeps building climate within a specified range, provides light to rooms based on an occupancy schedule (in the absence of overt switches to the contrary), monitors performance and device failures in all systems, and provides malfunction alarms to building maintenance staff. A BAS should reduce building energy and maintenance costs compared to a non-controlled building. Most commercial, institutional, and industrial buildings built after 2000 include a BAS. Many older buildings have been retrofitted with a new BAS, typically financed through energy and insurance savings, and other savings associated with pre-emptive maintenance and fault detection.

A building controlled by a BAS is often referred to as an intelligent building,[[3]](https://en.wikipedia.org/wiki/Building_automation#cite_note-Dragoicea2013-3) "[smart building](https://en.wikipedia.org/wiki/Smart_building)", or (if a residence) a "[smart home](https://en.wikipedia.org/wiki/Smart_home)". Commercial and industrial buildings have historically relied on robust proven protocols (like [BACnet](https://en.wikipedia.org/wiki/BACnet)) while proprietary and poorly integrated purpose-specific protocols (like [X-10](https://en.wikipedia.org/wiki/X10_(industry_standard)) or those from [Johnson Controls](https://en.wikipedia.org/wiki/Johnson_Controls), [Honeywell](https://en.wikipedia.org/wiki/Honeywell),[Siemens](https://en.wikipedia.org/wiki/Siemens) or other major manufacturers of [smart thermostats](https://en.wikipedia.org/wiki/Smart_thermostat), etc.) were used in homes. Recent [IEEE](https://en.wikipedia.org/wiki/IEEE) standards (notably [IEEE 802.15.4](https://en.wikipedia.org/wiki/IEEE_802.15.4), [IEEE 1901](https://en.wikipedia.org/wiki/IEEE_1901) and [IEEE 1905.1](https://en.wikipedia.org/w/index.php?title=IEEE_1905.1&action=edit&redlink=1), [IEEE 802.21](https://en.wikipedia.org/wiki/IEEE_802.21),[IEEE 802.11ac](https://en.wikipedia.org/wiki/IEEE_802.11ac), [IEEE 802.3at](https://en.wikipedia.org/wiki/IEEE_802.3at)) and consortia efforts like [nVoy](https://en.wikipedia.org/w/index.php?title=NVoy&action=edit&redlink=1) (which verifies IEEE 1905.1 compliance) or [QIVICON](https://en.wikipedia.org/wiki/QIVICON) have provided a standards-based foundation for heterogeneous networking of many devices on many physical networks for diverse purposes, and [quality of service](https://en.wikipedia.org/wiki/Quality_of_service) and [failover](https://en.wikipedia.org/wiki/Failover) guarantees appropriate to support human health and safety. Accordingly commercial, industrial, military and other institutional users now use systems that differ from home systems mostly in scale. *See*[*home automation*](https://en.wikipedia.org/wiki/Home_automation)*for more on entry level systems, nVoy, 1905.1, and the major proprietary vendors who implement or resist this trend to standards integration.*



Automation system[[edit](https://en.wikipedia.org/w/index.php?title=Building_automation&action=edit&section=1)]

*Main article:*[*Building management system*](https://en.wikipedia.org/wiki/Building_management_system)

The term "Building Automation System", loosely used, refers to any electrical control system that is used to controls a buildings heating, ventilation and air conditioning (HVAC) system. Modern BAS can also control indoor and outdoor lighting as well as security, fire alarms, and basically everything else that is electrical in the building. Old HVAC control systems, such as 24VDC wired thermostats or pneumatic controls, are a form of automation but lack the modern systems flexibility and integration.

## Buses and protocols[[edit](https://en.wikipedia.org/w/index.php?title=Building_automation&action=edit&section=2)]

Most building automation networks consist of a *primary* and *secondary* [bus](https://en.wikipedia.org/wiki/Bus_(computing)) which connect high-level controllers (generally specialized for building automation, but may be generic[programmable logic controllers](https://en.wikipedia.org/wiki/Programmable_logic_controller)) with lower-level controllers, [input/output](https://en.wikipedia.org/wiki/Input/output) devices and a [user interface](https://en.wikipedia.org/wiki/User_interface) (also known as a human interface device). [ASHRAE](https://en.wikipedia.org/wiki/ASHRAE)'s open protocol [BACnet](https://en.wikipedia.org/wiki/BACnet) or the open protocol [LonTalk](https://en.wikipedia.org/wiki/LonTalk) specify how most such devices interoperate. Modern systems use[SNMP](https://en.wikipedia.org/wiki/SNMP) to track events, building on decades of history with SNMP-based protocols in the computer networking world.

Physical connectivity between devices was historically provided by dedicated [optical fiber](https://en.wikipedia.org/wiki/Optical_fiber),[ethernet](https://en.wikipedia.org/wiki/Ethernet), [ARCNET](https://en.wikipedia.org/wiki/ARCNET), [RS-232](https://en.wikipedia.org/wiki/RS-232), [RS-485](https://en.wikipedia.org/wiki/EIA-485) or a low-bandwidth special purpose [wireless network](https://en.wikipedia.org/wiki/Wireless_network). Modern systems rely on standards-based multi-protocol heterogeneous networking such as that specified in the [IEEE 1905.1](https://en.wikipedia.org/w/index.php?title=IEEE_1905.1&action=edit&redlink=1) standard and verified by the [nVoy](https://en.wikipedia.org/w/index.php?title=NVoy&action=edit&redlink=1) auditing mark. These accommodate typically only IP-based networking but can make use of any existing wiring, and also integrate [powerline networking](https://en.wikipedia.org/wiki/Powerline_networking) over AC circuits, [power over Ethernet](https://en.wikipedia.org/wiki/Power_over_Ethernet) low power DC circuits, high-bandwidth wireless networks such as [LTE](https://en.wikipedia.org/wiki/LTE_(telecommunication)) and [IEEE 802.11n](https://en.wikipedia.org/wiki/IEEE_802.11n) and [IEEE 802.11ac](https://en.wikipedia.org/wiki/IEEE_802.11ac)and often integrate these using the building-specific wireless mesh open standard [ZigBee](https://en.wikipedia.org/wiki/ZigBee)).

[Proprietary hardware](https://en.wikipedia.org/wiki/Proprietary_hardware) dominates the controller market. Each company has controllers for specific applications. Some are designed with limited controls and no interoperability, such as simple packaged roof top units for HVAC. Software will typically not integrate well with packages from other vendors. Cooperation is at the Zigbee/BACnet/LonTalk level only.

Current systems provide interoperability at the application level, allowing users to mix-and-match devices from different manufacturers, and to provide integration with other compatible building control systems. These typically rely on [SNMP](https://en.wikipedia.org/wiki/SNMP), long used for this same purpose to integrate diverse computer networking devices into one coherent network.

## Types of inputs and outputs[[edit](https://en.wikipedia.org/w/index.php?title=Building_automation&action=edit&section=3)]

Analog inputs are used to read a variable measurement. Examples are [temperature](https://en.wikipedia.org/wiki/Temperature), [humidity](https://en.wikipedia.org/wiki/Humidity)and [pressure](https://en.wikipedia.org/wiki/Pressure) [sensors](https://en.wikipedia.org/wiki/Sensors) which could be [thermistor](https://en.wikipedia.org/wiki/Thermistor), [4-20 mA](https://en.wikipedia.org/wiki/Current_loop#Process-control_use), 0-10 [volt](https://en.wikipedia.org/wiki/Volt) or platinum [resistance thermometer](https://en.wikipedia.org/wiki/Resistance_thermometer) (resistance temperature detector), or wireless [sensors](https://en.wikipedia.org/wiki/Sensor).

A digital input indicates if a device is turned on or not. Some examples of an inherently digital input would be a 24VDC/AC signal, current switch, an air flow [switch](https://en.wikipedia.org/wiki/Switch), or a volta-free[relay](https://en.wikipedia.org/wiki/Relay) contact (Dry Contact). Digital inputs could also be pulse type inputs counting the frequency of pulses over a given period of time. An example is a turbine flow meter transmitting rotation data as a frequency of pulses to an input.

Analog outputs control the speed or position of a device, such as a [variable frequency drive](https://en.wikipedia.org/wiki/Variable_frequency_drive), an I-P ([current](https://en.wikipedia.org/wiki/Current_(electricity)) to [pneumatics](https://en.wikipedia.org/wiki/Pneumatics)) [transducer](https://en.wikipedia.org/wiki/Transducer), or a valve or damper [actuator](https://en.wikipedia.org/wiki/Actuator). An example is a hot water valve opening up 25% to maintain a [setpoint](https://en.wikipedia.org/wiki/Setpoint_(control_system)). Another example is a [variable frequency drive](https://en.wikipedia.org/wiki/Variable_frequency_drive) ramping up a motor slowly to avoid a hard start.

Digital outputs are used to open and close relays and switches as well as drive a load upon command. An example would be to turn on the parking lot lights when a [photocell](https://en.wikipedia.org/wiki/Photocell) indicates it is dark outside. Another example would be to open a valve by allowing 24VDC/AC to pass through the output powering the valve. Digital outputs could also be pulse type outputs emitting a frequency of pulses over a given period of time. An example is an energy meter calculating kWh and emitting a frequency of pulses accordingly.

## Infrastructure[[edit](https://en.wikipedia.org/w/index.php?title=Building_automation&action=edit&section=4)]

### Controller**[**[**edit**](https://en.wikipedia.org/w/index.php?title=Building_automation&action=edit&section=5)**]**

Controllers are essentially small, purpose-built computers with input and output capabilities. These controllers come in a range of sizes and capabilities to control devices commonly found in buildings, and to control sub-networks of controllers.

Inputs allow a controller to read temperatures, humidity, pressure, current flow, air flow, and other essential factors. The outputs allow the controller to send command and control signals to slave devices, and to other parts of the system. Inputs and outputs can be either digital or analog. Digital outputs are also sometimes called discrete depending on manufacturer.

Controllers used for building automation can be grouped in 3 categories. Programmable Logic Controllers (PLCs), System/Network controllers, and Terminal Unit controllers. However an additional device can also exist in order to integrate 3rd party systems (i.e. a stand-alone AC system) into a central Building automation system).

Terminal Unit controllers usually are suited for control of lighting and/or simpler devices such as a package rooftop unit, heat pump, VAV box, or fan coil, etc. The installer typically selects 1 of the available pre-programmed personalities best suited to the device to be controlled, and does not have to create new control logic.

Advantages and Disadvantages of Building Automation System

The advantages are obviouse, see the status and have full control of the buildings infrastructure from one easy to use system, gone are the days of having to run around the building to turn things on/off or tocheck status, it also helps to reduce operating and maintenance costs, the system can alert you to an impending problem before it becoes an irreversable expensive repair, and more. 

Disadvantages, high up front costs, can be a bigger pain if not designed and/installed/configured correctly, users and maintenance crew need to be more tech savvy than in the past.

– Detail explanation of components of proposed system

### Fan - Air supply,exhaust/Exhaust system with vertical shaft

### Applicable for any length of tunnel

* This system exhausts polluted air inside a tunnel through a shaft and/or a duct and supplies external fresh air into the tunnel by using fans installed in a ventilation section or a ventilation tower.
* The shaft type ventilation system is a system applicable for any length of tunnel by replacing polluted air with external fresh air through a shaft which is constructed at the place where hazardous substances in air inside the tunnel reach the permissible limit.
* The exhaust system emits polluted air in a tunnel from a ventilation tower through a shaft, resulting in substantial reduction of hazardous substances at the wayout of tunnel.
* The fan in this system can change the ventilation flow rate by variable pitch control mechanism.

### **Variable pitch axial fan**

## H:\Lt Col Soe Win Myint\Air supply_exhaust system with vertical shaft _ Tunnel Air Purification _ Air Conditioning and Purification _ Eco Solutions _ Business _ Panasonic Global_files\variablepitchaxialfan.jpg

## **Air supply exhaust system with vertical shaft**

## H:\Lt Col Soe Win Myint\Air supply_exhaust system with vertical shaft _ Tunnel Air Purification _ Air Conditioning and Purification _ Eco Solutions _ Business _ Panasonic Global_files\airsupplyexhaustsystem.jpg

## Tunnel Ventilation System

## H:\Lt Col Soe Win Myint\Tunnel Ventilation System _ Kawasaki Heavy Industries, Ltd._files\img_machinery_A10_02.jpg

## H:\Lt Col Soe Win Myint\Tunnel Ventilation System _ Kawasaki Heavy Industries, Ltd._files\img_machinery_A10_03.jpg

### Spec

### Jet Fans

Jet Fans are installed in venous type tunnels. Jet Fans have the advantage of not only being constructed in a short time, but also of being easily installed in tunnels. KHI can design and provide reliable Jet Fans.

|  | **Model** | **KJF 600** | | **KJF 1000** | | **KJF 1000** | | **KJF 1250** | | **KJF 1250** | | **KJf 1500** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Specifications** |  | | **standard model** | | **high-speed model** | | **standard model** | | **high-speed model** | |  | |
| **Ventilator** | **Type** | Horizontal type axial flow ventilator | | | | | | | | | | | |
| **Aperture (mmΦ)** | 630 | | 1030 | | 1030 | | 1250 | | 1250 | | 1530 | |
| **Length (mm)** | 3000 | | 4250 | | 4250 | | 4250 | | 4250 | | 4250 | |
| **Exterior diameter (mmΦ)** | 800 | | 1200 | | 1200 | | 1450 | | 1450 | | 1750 | |
| **Nominal average air flow speed (m/s, min.)** | 30 | | 30 | | 35 | | 30 | | 35 | | 30 | |
| **Discharge air volume (m3/s, min.)** | 8 | | 25 | | 29 | | 37 | | 43 | | 55 | |
| **Effective discharge areas (m²)** | 0.267 | | 0.833 | | 0.833 | | 1.227 | | 1.227 | | 1.838 | |
| **Fan Efficiency (%, min.)** | 60 | | 65 | | 75 | | 75 | | 75 | | 70 | |
| **Noise (db(A), min.) ※** | 90 | | 95 | | 95 | | 95 | | 95 | | 95 | |
| **Venting direction** | Both directions | | | | | | | | | | | |
| **Electric motor** |  | Fully-enclosed three-phase induction motor | | | | | | | | | | | |
| **Type** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Rated voltage (V)** | 400 | 440 | 400 | 440 | 400 | 440 | 400 | 440 | 400 | 440 | 400 | 440 |
| **Frequency (Hz)** | 50 | 60 | 50 | 60 | 50 | 60 | 50 | 60 | 50 | 60 | 50 | 60 |
| **Rated output (kW)** | 9.5 | 9.5 | 25 | 25 | 33 | 33 | 30 | 30 | 50 | 50 | 50 | 50 |
| **Ratings** | Continuous | | | | | | | | | | | |
| **Insulation type** | F type | | | | | | | | | | | |
| **Approximate weight (kg)** | 600 | | 1300 | | 1300 | | 2000 | | 2000 | | 2500 | |

### **Large Axial-Fan**

In the case of particularly long tunnels when adequate ventilation is not possible with the Jet Fans only alone, the Large Axial Fans can be used, either alone or in combination with the Jet Fans.  
KHI can design highly efficient, low noise, sufficiently reliable and economical Large Axial Fans. KHI has provided various types of Large Axial Fans.

### **Electrostatic Precipitator (ESP) for Road Tunnel**

ESP can remove drifting SPM (Suspended Particulate Matter) in the atmosphere of road tunnels. So that, ESP for road tunnel improves visibility in road tunnels, cleans up exhaust gas from them, and contributes environmental preservation.

| **Type** |  | **Horizontal air flow, Two stages (separate Ionization and precipitation), Wash by water** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ESP** | | **TYPE I** | **TYPE II** | **TYPE III** | **TYPE IV** | **TYPE V** | **TYPE VI** | **TYPE VII** | **TYPE VIII** |
| **(Air flow rate)** | | 7.2m³/s | 14.4m³/s | 14.4m³/s | 21.6m³/s | 28.8m³/s | 28.8m³/s | 43.2m³/s | 57.6m³/s |
|  | |  | Series | Parallel | Series | Parallel | Parallel |  |  |
| **STD** | **Width** | 1,040 | 1,040 | 2,000 | 1,040 | 1,040 | 2,000 | 2,000 | 2,000 |
| **(mm)** | **Height** | 940 | 1,670 | 940 | 2,460 | 3,230 | 1,670 | 2,460 | 3,230 |
|  | **Depth** | 1,700 | 1,700 | 1,700 | 1,700 | 1,700 | 1,700 | 1,700 | 1,700 |
| **Number of units\*** | | 1 | 2 | 2 | 3 | 4 | 4 | 6 | 8 |
| **Efficiency\*\*** | | More than 80% (PM10)/The result of factory test using actual ESP unit | | | | | | | |
| **Press Loss\*\*\*** | | 250 Pa or less/The result of factory test using actual ESP unit and cage | | | | | | | |
| **Ionization** | | DC-11kV or less | | | | | | | |
| **Precipitation** | | DC-11kV or less | | | | | | | |

\*

One unit can work up to 7.2m³/s which means that air velocity is up to approx. 13m/s inside ESP unit.

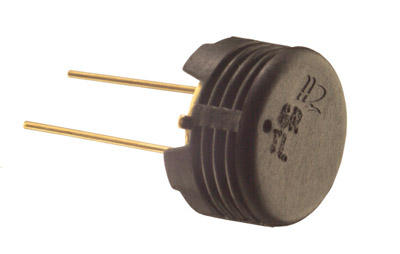
\*\*

Test gas (includes PM10 about 0.2-1.0 mg/m³ consists of Diesel Engine Exhaust. Measurement is JIS B 9910 and JIS Z 8808 with 3-6 hours sampling.

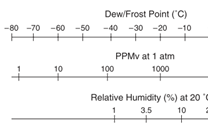
\*\*\*

Measurement condition is standard. (Atmospheric Press. 101.3Pa, the temperature 20 degree Celsius, relative humidity 65%, air flow rate is 7.2m3/s per ESP unit.)

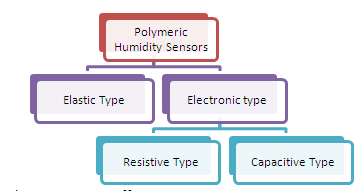
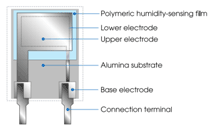
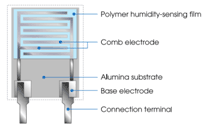
### Sensors (Humidity, Current, …)

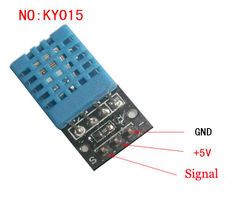
* Humidity is the presence of water in air. The amount of water vapor in air can affect human comfort as well as many manufacturing processes in industries. The presence of water vapor also influences various physical, chemical, and biological processes. Humidity measurement in industries is critical because it may affect the business cost of the product and the health and safety of the personnel. Hence, **humidity sensing** is very important, especially in the control systems for industrial processes and human comfort.
* Controlling or monitoring humidity is of paramount importance in many industrial & domestic applications. In semiconductor industry, humidity or moisture levels needs to be properly controlled & monitored during wafer processing. In medical applications, humidity control is required for respiratory equipments, sterilizers, incubators, pharmaceutical processing, and biological products. Humidity control is also necessary in chemical gas purification, dryers, ovens, film desiccation, paper and textile production, and food processing. In agriculture, measurement of humidity is important for plantation protection (dew prevention), soil moisture monitoring, etc. For domestic applications, humidity control is required for living environment in buildings, cooking control for microwave ovens, etc.  In all such applications and many others, **humidity sensors** are employed to provide an indication of the moisture levels in the environment.
* **RELEVANT MOISTURE TERMS**
* To mention moisture levels, variety of terminologies are used. The study of water vapour concentration in air as a function of temperature and pressure falls under the area of psychometrics. Psychometrics deals with the thermodynamic properties of moist gases while the term “humidity’ simply refers to the presence of water vapour in air or other carrier gas.
* Humidity measurement determines the amount of water vapor present in a gas that can be a mixture, such as air, or a pure gas, such as nitrogen or argon. Various terms used to indicate moisture levels are tabulated in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Term** | **Definition** | **Unit** |
| 1 | Absolute Humidity  (Vapor Concentration) | Ratio of mass(vapour) to volume. | grams/m3 |
| 2 | Mixing Ratio OR Mass Ratio | Ratio of mass(vapour) to mass(dry gas) | grams/m3 |
| 3 | Relative Humidity | Ratio of mass(vapour) to mass(saturated vapour) OR ratio of actual vapor pressure to saturation vapor pressure. | % |
| 4 | Specific Humidity | Ratio of mass(vapour) to total mass. | % |
| 5 | Dew Point | Temperature(above 0°C) at which the water vapor in a gas condenses to liquid water) | °C |
| 6 | Frost Point | Temperature(below 0°C) at which the water vapor in a gas condenses to ice |  |
| 7 | Volume Ratio | Ratio of partial pressure(vapour) to partial pressure (dry gas) | % by volume |
| 8 | PPM by Volume | Ratio of volume(vapour) X 106 to volume(dry gas) | PPMV |
| 9 | PPM by Weight | PPMVX | PPMW |

* Most commonly used units for humidity measurement are Relative Humidity (RH), Dew/Frost point (D/F PT) and Parts Per Million (PPM). RH is a function of temperature, and thus it is a relative measurement. Dew/Frost point is a function of the pressure of the gas but is independent of temperature and is therefore defined as absolute humidity measurement. PPM is also an absolute measurement.
* Dew points and frost points are often used when the dryness of the gas is important. Dew point is also used as an indicator of water vapor in high temperature processes, such as industrial drying.
* Mixing ratios, volume percent, and specific humidity are usually used when water vapor is either an impurity or a defined component of a process gas mixture used in manufacturing.
* Correlation among RH, Dew/Frost point and PPMvis shown below:
* 
* **HUMIDITY SENSING – CLASSIFICATION & PRINCIPLES**
* According to the measurement units, humidity sensors are divided into two types: Relative humidity(RH)sensors and absolute humidity(moisture) sensors. Most humidity sensors are relative humidity sensors and use **different sensing principles**.
* A table showing important parameters of different types of humidity sensors is given below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Active Material | Thermo-set  Polymer | Thermoplastic  Polymer | Thermoplastic  Polymer | Bulk Thermoplastic | Bulk  AlO3 | Lithium  Chloride Film |
| Substrate | Ceramic or  Silicon | Ceramic or  silicon | Polyester or  mylar film | N/A | N/A | Ceramic |
| Sensed Parameter | Capacitance | Capacitance | Capacitance | Resistance | Resistance | Conductivity |
| Measured Parameter | %RH | %RH | %RH | %RH | %RH | %RH |
| RH Change | 0% to 100% | 0% to 100% | 0% to 100% | 20% to 100% | 2% to 90% | 15% to <100% |
| RH Accuracy | ±1% to ±5% | ±3% to ±5% | ±3% to ±5% | ±3% to ±10% | ±1% to ±5% | ±5% |
| Interchangability | ±2% to  ±10% RH | ±3% to  ±20% RH | ±3% to  ±20% RH | ±5% to  ±25% RH | poor | ±3% to  ±10% RH |
| Hysterisis | <1% to 3% RH | 2% to 5% RH | 2% to 5% RH | 3% to 6% RH | <2% RH | very poor |
| Linearity | ±1% RH | ±1% RH | ±2% RH | poor | poor | Very poor |
| Risetime | 15 s to 60 s | 15 s to 90 s | 15 s to 90 s | 2 min to 5 min | 3 min to 5 min | 3 min to 5 min |
| Temperature  Range | -40 °C to  185 °C | -30 °C to  190 °C | -25°C to  100 °C | 10 °C to  40 °C | -10 °C to  75 °C | - |
| Long Term  Stability | ±1%RH/5 yr | ±1%RH/yr | ±1%RH/yr | ±3%RH/yr | ±3% RH/yr | >1% RH/°C |

* **·         Sensing Principle**
* Humidity measurement can be done using dry and wet bulb hygrometers, dew point hygrometers, and electronic hygrometers. There has been a surge in the demand of electronic hygrometers, often called humidity sensors.
* Electronic type hygrometers or humidity sensors can be broadly divided into two categories: one employs capacitive sensing principle, while other use resistive effects
* 
* **Sensors based on capacitive effect:**
* Humidity sensors relying on this principle consists of a hygroscopic dielectric material sandwiched between a pair of electrodes forming a small capacitor. Most capacitive sensors use a plastic or polymer as the dielectric material, with a typical dielectric constant ranging from 2 to 15. In absence of moisture, the dielectric constant of the hygroscopic dielectric material and the sensor geometry determine the value of capacitance.
* At normal room temperature, the dielectric constant of water vapor has a value of about 80, a value much larger than the constant of the sensor dielectric material. Therefore, absorption of water vapor by the sensor results in an increase in sensor capacitance.
* At equilibrium conditions, the amount of moisture present in a hygroscopic material depends on both the ambient temperature and the ambient water vapor pressure. This is true also for the hygroscopic dielectric material used on the sensor.
* By definition, relative humidity is a function of both the ambient temperature and water vapor pressure. Therefore there is a relationship between relative humidity, the amount of moisture present in the sensor, and sensor capacitance. This relationship governs the operation of a capacitive humidity instrument.
* Basic structure of capacitive type humidity sensor is shown below:
* 
* On Alumina substrate, lower electrode is formed using gold, platinum or other material. A polymer layer such as PVA is deposited on the electrode. This layers senses humidity. On top of this polymer film, gold layer is deposited which acts as top electrode. The top electrode also allows water vapour to pass through it, into the sensing layer .  The vapors enter or leave the hygroscopic sensing layer until the  vapour content is in equilibrium with the ambient air or gas.Thus capacitive type sensor is basically a capacitor with humidity sensitive polymer film as the dielectric.
* **Sensors based on Resistive effect:**
* Resistive type humidity sensors pick up changes in the resistance value of the sensor element in response to the change in the humidity. Basic structure of resistive type humidity sensor from TDK is shown below
* 
* Thick film conductor of precious metals like gold, ruthenium oxide is printed and calcinated in the shape of the comb to form an electrode. Then a polymeric film is applied on the electrode; the film acts as a humidity sensing film due to the existence of movable ions. Change in impedance occurs due to the change in the number of movable ions.



### Current sensor

From Wikipedia, the free encyclopedia

A **current sensor** is a device that detects [electric current](https://en.wikipedia.org/wiki/Electric_current) (AC or DC) in a wire, and generates a signal proportional to it. The generated signal could be analog voltage or current or even digital output. It can be then utilized to display the measured current in an ammeter or can be stored for further analysis in a data acquisition system or can be utilized for control purpose.

The sensed current and the output signal can be:

* [Alternating current](https://en.wikipedia.org/wiki/Alternating_current) input,
  + analog output, which duplicates the wave shape of the sensed current
  + bipolar output, which duplicates the wave shape of the sensed current
  + unipolar output, which is proportional to the average or RMS value of the sensed current
* [Direct current](https://en.wikipedia.org/wiki/Direct_current) input,
  + unipolar, with a unipolar output, which duplicates the wave shape of the sensed current
  + digital output, which switches when the sensed current exceeds a certain threshold

## Technologies[[edit](https://en.wikipedia.org/w/index.php?title=Current_sensor&action=edit&section=1)]

* [Hall effect](https://en.wikipedia.org/wiki/Hall_effect) [IC](https://en.wikipedia.org/wiki/Integrated_circuit) sensor.
* [Transformer](https://en.wikipedia.org/wiki/Transformer) or [current clamp](https://en.wikipedia.org/wiki/Current_clamp) meter, (suitable for AC current only).
* [Resistor](https://en.wikipedia.org/wiki/Resistor), whose voltage is directly proportional to the current through it.
* [Fiber optic current sensor](https://en.wikipedia.org/wiki/Fiber_optic_current_sensor), using an [interferometer](https://en.wikipedia.org/wiki/Interferometry) to measure the phase change in the light produced by a magnetic field.
* [Rogowski coil](https://en.wikipedia.org/wiki/Rogowski_coil), electrical device for measuring alternating current (AC) or high speed current pulses.

Hall Effect current sensor is a type of current sensor which is based on phenomenon of Hall Effect discovered by Edwin Hall in 1879.

Hall Effect current sensors can measure all types of current signals i.e. AC,DC or pulsating current.

These sensors are currently being used widely in almost all the industries because of their vast applications and the type of output they provide, which can be manipulated and can be used for various application.

he technique is called high-side current sensing. However, measurements at that location are fraught with problems. Of primary concern is that the measurement is at main line voltage, a problem if the current being measured feeds an electronic circuit that checks it for control or overload sensing.

A new generation of small Hall-effect-based current sensors simplify the task. Hall-effect devices measure current via the intensity of the magnetic field generated by the current flow. Of course, higher currents produce stronger magnetic fields. These fields now provide the means to measure high-side current in applications where it has not been physically or economically feasible.

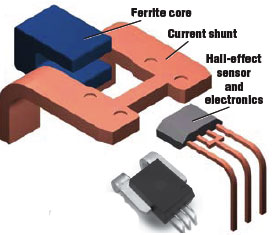
Hall-effect devices are named for Edwin Herbert Hall. He found that if you place a conductor carrying an electric current in a magnetic field, a voltage is induced across the conductor at right angles to the current flow. The strength of that induced voltage indicates the relative strength of the magnetic field.

In Hall-effect sensors, current to be measured passes through a shunt path. This path is extremely low resistance, typically only a few milliohms. The shunt produces little insertion loss when placed in a current path, so it works well with high-current, low-voltage circuits.

The Hall-effect sensor and calibration electronics sit in a small notch next to the shunt. A ferrite core wrapped around the shunt and Hall-effect sensor completes the magnetic path and helps concentrate the magnetic field through the sensor.

The sensor produces an output voltage directly related to the strength of the magnetic field generated by the current flowing through the shunt. Many such sensors operate from a single 5-Vdc supply so their output does not need level shifting before connecting to an a/d converter, microprocessor, or microcontroller.

Hall-effect current sensors are virtually immune to environmental contaminants. They're rugged and suitable for use under severe conditions. These devices work well with dc motors, automotive applications, robotic, and heavy-machinery sensing applications, or any applications in harsh environments. There is no electrical connection between the current path and sensor and, unlike current transformers, Hall-based sensors measure dc as well as ac currents.



Hall-effect current sensors measure the strength of the magnetic field generated by the current flow through a shunt path, producing an output relative to the amount of current. They also provide electrical isolation between the voltage of the sensed current and the output of the sensor. Current sensor graphic courtesy Allegro Microsystems Inc., Worcester, MA

### Controller circuit

Relays and Microcontroller are composed.

# **Four Channel Relay Module**



# **PIC microcontroller**

**PIC** is a family of [modified Harvard architecture](https://en.wikipedia.org/wiki/Modified_Harvard_architecture)[microcontrollers](https://en.wikipedia.org/wiki/Microcontroller) made by [Microchip Technology](https://en.wikipedia.org/wiki/Microchip_Technology), derived from the PIC1650[[1]](https://en.wikipedia.org/wiki/PIC_microcontroller#cite_note-1)[[2]](https://en.wikipedia.org/wiki/PIC_microcontroller#cite_note-2)[[3]](https://en.wikipedia.org/wiki/PIC_microcontroller#cite_note-PICmicrofamilytree-3) originally developed by [General Instrument](https://en.wikipedia.org/wiki/General_Instrument)'s Microelectronics Division. The name PIC initially referred to *Peripheral Interface Controller*.[[4]](https://en.wikipedia.org/wiki/PIC_microcontroller#cite_note-1976databook-4)[[5]](https://en.wikipedia.org/wiki/PIC_microcontroller#cite_note-1977catalog-5) The first parts of the family were available in 1976; by 2013 the company had shipped more than twelve billion individual parts, used in a wide variety of [embedded systems](https://en.wikipedia.org/wiki/Embedded_system).

Early models of PIC had read-only memory (ROM) or field-programmable EPROM for program storage, some with provision for erasing memory. All current models use [Flash memory](https://en.wikipedia.org/wiki/Flash_memory) for program storage, and newer models allow the PIC to reprogram itself. Program memory and data memory are separated. Data memory is 8-bit, 16-bit and in latest models, 32-bit wide. Program instructions vary in bit-count by family of PIC, and may be 12, 14, 16, or 24 bits long. The instruction set also varies by model, with more powerful chips adding instructions for [digital signal processing](https://en.wikipedia.org/wiki/Digital_signal_processing) functions.

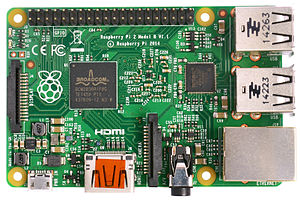
The hardware capabilities of PIC devices range from 8-pin [DIP](https://en.wikipedia.org/wiki/Dual_in-line_package) chips up to 100-pin [SMD](https://en.wikipedia.org/wiki/Surface-mount_technology) chips, with discrete I/O pins, ADC and DAC modules, and communications ports such as [UART](https://en.wikipedia.org/wiki/UART), [I2C](https://en.wikipedia.org/wiki/I2C), CAN, and even [USB](https://en.wikipedia.org/wiki/Universal_serial_bus). Low-power and high-speed variations exist for many types.

The manufacturer supplies computer software for development known as [MPLAB](https://en.wikipedia.org/wiki/MPLAB), assemblers and C/C++ compilers, and programmer/debugger hardware under the [MPLAB](https://en.wikipedia.org/wiki/MPLAB_devices) and [PICKit](https://en.wikipedia.org/wiki/PICKit) series. Third party and some open-source tools are also available. Some parts have in-circuit programming capability; low-cost development programmers are available as well has high-production programmers.

PIC devices are popular with both industrial developers and hobbyists due to their low cost, wide availability, large user base, extensive collection of application notes, availability of low cost or free development tools, serial programming, and re-programmable Flash-memory capability.



### Raspberry Pi (PHP web services, Python GPIO control, JSON object, HTTP post, HTTP get, MySQL)



The **Raspberry Pi** is a series of [credit card](https://en.wikipedia.org/wiki/Credit_card)–sized [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) developed in the [UK](https://en.wikipedia.org/wiki/United_Kingdom) by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation)with the intention of promoting the teaching of basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and third world countries.[[5]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-6)[[7]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-7)

The original Raspberry Pi and Raspberry Pi 2 are manufactured in several board configurations through licensed manufacturing agreements with [Newark element14](https://en.wikipedia.org/wiki/Newark_element14) ([Premier Farnell](https://en.wikipedia.org/wiki/Premier_Farnell)), [RS Components](https://en.wikipedia.org/wiki/RS_Components) and Egoman. These companies sell the Raspberry Pi online.[[8]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-8) Egoman produces a version for distribution solely in Taiwan, which can be distinguished from other Pis by their red colouring and lack of [FCC](https://en.wikipedia.org/wiki/FCC_Declaration_of_Conformity)/[CE marks](https://en.wikipedia.org/wiki/CE_marking). The hardware is the same across all manufacturers.

The original Raspberry Pi is based on the [Broadcom](https://en.wikipedia.org/wiki/Broadcom) BCM2835 [system on a chip](https://en.wikipedia.org/wiki/System_on_a_chip) (SoC),[[2]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-Broadcom-BCM2835-Website-2) which includes an[ARM1176JZF-S](https://en.wikipedia.org/wiki/ARM11) 700 [MHz](https://en.wikipedia.org/wiki/Hertz) processor,[VideoCore](https://en.wikipedia.org/wiki/VideoCore) IV GPU,[[9]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-grandmax_brose_2012-9) and was originally shipped with 256 megabytes of [RAM](https://en.wikipedia.org/wiki/Random-access_memory), later upgraded (models B and B+) to 512 [MB](https://en.wikipedia.org/wiki/Mebibyte).[[3]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-MB-3)[[10]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-512MB-10) The system has [Secure Digital](https://en.wikipedia.org/wiki/Secure_Digital) (SD) (models A and B) or MicroSD (models A+ and B+) sockets for boot media and persistent storage.[[11]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-VerifiedPeripheralList-11)

In 2014, the Raspberry Pi Foundation launched the *Compute Module*, which packages a BCM2835 with 512 MB RAM and an [eMMC](https://en.wikipedia.org/wiki/MultiMediaCard#eMMC) flash chip into a module for use as a part of embedded systems.[[12]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-12)

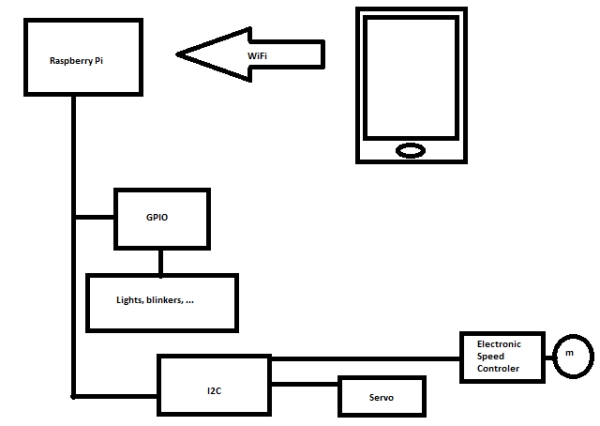
The Foundation provides Debian and Arch Linux ARM [distributions](https://en.wikipedia.org/wiki/Linux_distribution) for download.[[13]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-raspberrypi_downloads-13) Tools are available for[Python](https://en.wikipedia.org/wiki/Python_(programming_language)) as the main programming language, with support for [BBC BASIC](https://en.wikipedia.org/wiki/BBC_BASIC)[[14]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-edge_david_braben-14) (via the [RISC OS](https://en.wikipedia.org/wiki/RISC_OS) image or the Brandy Basic clone for Linux),[[15]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-15) [C](https://en.wikipedia.org/wiki/C_(programming_language)),[C++](https://en.wikipedia.org/wiki/C%2B%2B), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)),[[16]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-java_on_raspberry_pi-16) [Perl](https://en.wikipedia.org/wiki/Perl) and [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)).[[17]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-Raspberry_Pi_GPIO_Programming_with_pi_piper_Ruby_gem-17)

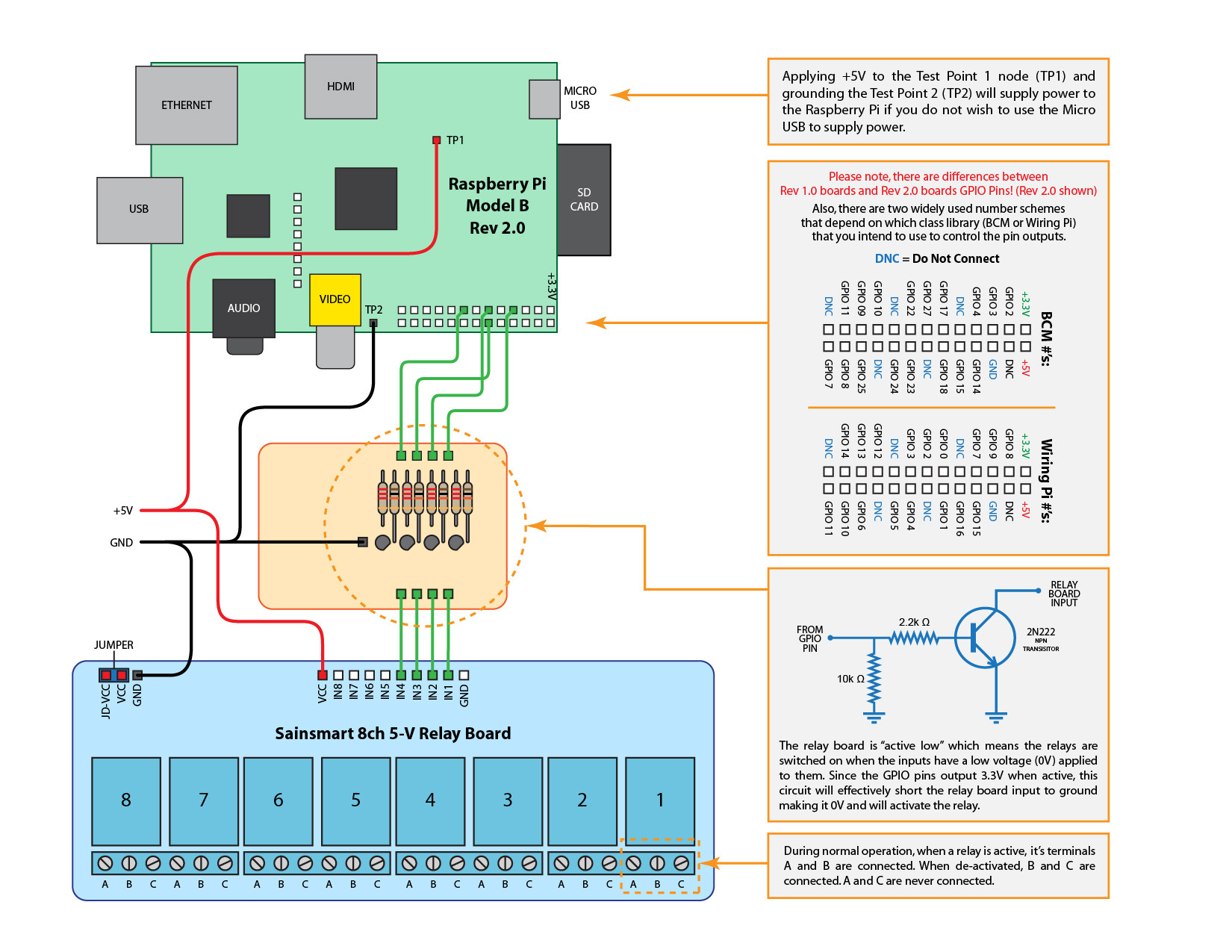
As of 8 June 2015, about five to six million Raspberry Pis have been sold.[[18]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-18)[[19]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-19) While already the fastest selling British [personal computer](https://en.wikipedia.org/wiki/Personal_computer), it has also shipped the second largest number of units behind the [Amstrad PCW](https://en.wikipedia.org/wiki/Amstrad_PCW), the "Personal Computer [Word-processor](https://en.wikipedia.org/wiki/Word-processor)", which sold eight million.

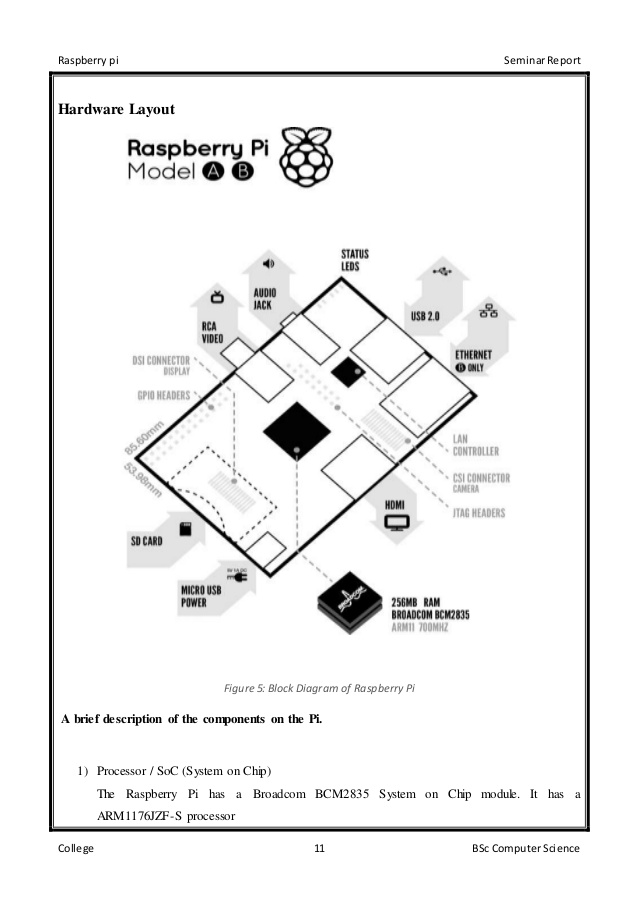
In early February 2015, the next-generation Raspberry Pi, Raspberry Pi 2, was released.[[20]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-20) The new computer board is initially available only in one configuration (model B) and features a Broadcom BCM2836 SoC, with a [quad-core](https://en.wikipedia.org/wiki/Multi-core_processor) [ARM Cortex-A7](https://en.wikipedia.org/wiki/ARM_Cortex-A7) CPU and a VideoCore IV dual-core GPU; 1 GB of RAM with remaining specifications being similar to those of the previous generation model B+. The Raspberry Pi 2 retains the same US$35 price point of the model B,[[21]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-BBC_RaspPi_2-21) with the US$20 model A+ remaining on sale

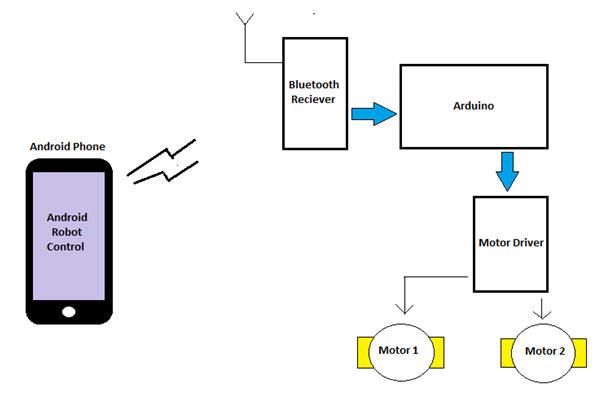
### Android Programming (JSON parsing, HTTP post, HTTP get)

### - Block diagram and Design of proposed system









## LITERATURE REVIEW AND BACKGROUND THEORY

### 2.1 Control System Theory

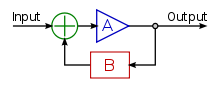
A **control system** is a device, or set of devices, that manages, commands, directs or regulates the behavior of other devices or systems. [Industrial control systems](https://en.wikipedia.org/wiki/Industrial_control_system) are used in industrial production for controlling equipment or machines.

There are two common classes of control systems, open loop control systems and closed loop control systems. In open loop control systems output is generated based on inputs. In closed loop control systems current output is taken into consideration and corrections are made based on feedback. A closed loop system is also called a feedback control system. The human body is a classic example of feedback systems.

The term "control system" may be applied to the essentially manual controls that allow an operator, for example, to close and open a [hydraulic press](https://en.wikipedia.org/wiki/Hydraulic_press), perhaps including logic so that it cannot be moved unless [safety guards](https://en.wikipedia.org/wiki/Safety_barrier) are in place.

An automatic [sequential control](https://en.wikipedia.org/wiki/Sequential_logic) system may trigger a series of mechanical [actuators](https://en.wikipedia.org/wiki/Actuator) in the correct sequence to perform a task. For example various electric and pneumatic transducers may fold and glue a cardboard box, fill it with product and then seal it in an automatic packaging machine. [Programmable logic controllers](https://en.wikipedia.org/wiki/Programmable_logic_controller) are used in many cases such as this, but several alternative technologies exist.

In the case of linear [feedback](https://en.wikipedia.org/wiki/Feedback) systems, a **control loop**, including [sensors](https://en.wikipedia.org/wiki/Sensor), control algorithms and actuators, is arranged in such a fashion as to try to regulate a variable at a [setpoint](https://en.wikipedia.org/wiki/Setpoint_(control_system)) or[reference value](https://en.wikipedia.org/wiki/Reference_value). An example of this may increase the fuel supply to a furnace when a measured temperature drops. [PID controllers](https://en.wikipedia.org/wiki/PID_controller) are common and effective in cases such as this. Control systems that include some sensing of the results they are trying to achieve are making use of feedback and so can, to some extent, adapt to varying circumstances. [Open-loop control systems](https://en.wikipedia.org/wiki/Open-loop_controller) do not make use of feedback, and run only in pre-arranged ways.



## Logic control

Logic control systems for industrial and commercial machinery were historically implemented at [mains voltage](https://en.wikipedia.org/wiki/Mains_electricity) using interconnected [relays](https://en.wikipedia.org/wiki/Relay), designed using[ladder logic](https://en.wikipedia.org/wiki/Ladder_logic). Today, most such systems are constructed with [programmable logic controllers](https://en.wikipedia.org/wiki/Programmable_logic_controller)(PLCs) or [microcontrollers](https://en.wikipedia.org/wiki/Microcontroller). The notation of ladder logic is still in use as a programming idiom for PLCs.[[1]](https://en.wikipedia.org/wiki/Control_system#cite_note-Kuphaldt_LADDER_LOGIC-1)

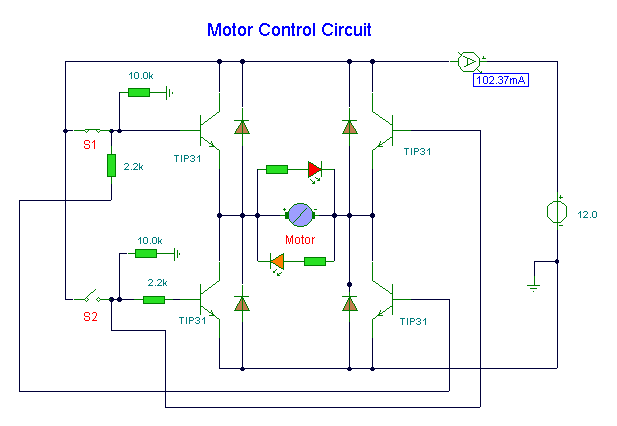


Logic controllers may respond to switches, light sensors, pressure switches, etc., and can cause the machinery to start and stop various operations. Logic systems are used to sequence mechanical operations in many applications. PLC software can be written in many different ways – ladder diagrams, SFC – sequential function charts or in language terms known as statement lists.[[2]](https://en.wikipedia.org/wiki/Control_system#cite_note-Ian_Brady_PLCs-2)

Examples include elevators, washing machines and other systems with interrelated stop-go operations.

Logic systems are quite easy to design, and can handle very complex operations. Some aspects of logic system design make use of [Boolean logic](https://en.wikipedia.org/wiki/Boolean_logic).

### 2.2 Control circuit and Ventilation fans



### 2.3 Raspberry Pi

### 2.4 Android App Development

Android apps are written in the Java programming language. The Android SDK tools compile your code—along with any data and resource files—into an APK: an *Android package*, which is an archive file with an .apk suffix. One APK file contains all the contents of an Android app and is the file that Android-powered devices use to install the app.

Once installed on a device, each Android app lives in its own security sandbox:

* The Android operating system is a multi-user Linux system in which each app is a different user.
* By default, the system assigns each app a unique Linux user ID (the ID is used only by the system and is unknown to the app). The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.
* Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps.
* By default, every app runs in its own Linux process. Android starts the process when any of the app's components need to be executed, then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

In this way, the Android system implements the principle of least privilege. That is, each app, by default, has access only to the components that it requires to do its work and no more. This creates a very secure environment in which an app cannot access parts of the system for which it is not given permission.

However, there are ways for an app to share data with other apps and for an app to access system services:

* It's possible to arrange for two apps to share the same Linux user ID, in which case they are able to access each other's files. To conserve system resources, apps with the same user ID can also arrange to run in the same Linux process and share the same VM (the apps must also be signed with the same certificate).
* An app can request permission to access device data such as the user's contacts, SMS messages, the mountable storage (SD card), camera, Bluetooth, and more. All app permissions must be granted by the user at install time.

That covers the basics regarding how an Android app exists within the system. The rest of this document introduces you to:

* The core framework components that define your app.
* The manifest file in which you declare components and required device features for your app.
* Resources that are separate from the app code and allow your app to gracefully optimize its behavior for a variety of device configurations.

## App Components

App components are the essential building blocks of an Android app. Each component is a different point through which the system can enter your app. Not all components are actual entry points for the user and some depend on each other, but each one exists as its own entity and plays a specific role—each one is a unique building block that helps define your app's overall behavior.

There are four different types of app components. Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed.

Here are the four types of app components:

**Activities**

An *activity* represents a single screen with a user interface. For example, an email app might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. Although the activities work together to form a cohesive user experience in the email app, each one is independent of the others. As such, a different app can start any one of these activities (if the email app allows it). For example, a camera app can start the activity in the email app that composes new mail, in order for the user to share a picture.

An activity is implemented as a subclass of [Activity](http://developer.android.com/reference/android/app/Activity.html) and you can learn more about it in the [Activities](http://developer.android.com/guide/components/activities.html)developer guide.

**Services**

A *service* is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface. For example, a service might play music in the background while the user is in a different app, or it might fetch data over the network without blocking user interaction with an activity. Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it.

A service is implemented as a subclass of [Service](http://developer.android.com/reference/android/app/Service.html) and you can learn more about it in the [Services](http://developer.android.com/guide/components/services.html)developer guide.

**Content providers**

A *content provider* manages a shared set of app data. You can store the data in the file system, an SQLite database, on the web, or any other persistent storage location your app can access. Through the content provider, other apps can query or even modify the data (if the content provider allows it). For example, the Android system provides a content provider that manages the user's contact information. As such, any app with the proper permissions can query part of the content provider (such as[ContactsContract.Data](http://developer.android.com/reference/android/provider/ContactsContract.Data.html)) to read and write information about a particular person.

Content providers are also useful for reading and writing data that is private to your app and not shared. For example, the [Note Pad](http://developer.android.com/resources/samples/NotePad/index.html) sample app uses a content provider to save notes.

A content provider is implemented as a subclass of [ContentProvider](http://developer.android.com/reference/android/content/ContentProvider.html) and must implement a standard set of APIs that enable other apps to perform transactions. For more information, see the [Content Providers](http://developer.android.com/guide/topics/providers/content-providers.html) developer guide.

**Broadcast receivers**

A *broadcast receiver* is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Apps can also initiate broadcasts—for example, to let other apps know that some data has been downloaded to the device and is available for them to use. Although broadcast receivers don't display a user interface, they may [create a status bar notification](http://developer.android.com/guide/topics/ui/notifiers/notifications.html) to alert the user when a broadcast event occurs. More commonly, though, a broadcast receiver is just a "gateway" to other components and is intended to do a very minimal amount of work. For instance, it might initiate a service to perform some work based on the event.

A broadcast receiver is implemented as a subclass of [BroadcastReceiver](http://developer.android.com/reference/android/content/BroadcastReceiver.html) and each broadcast is delivered as an [Intent](http://developer.android.com/reference/android/content/Intent.html) object. For more information, see the [BroadcastReceiver](http://developer.android.com/reference/android/content/BroadcastReceiver.html) class.

A unique aspect of the Android system design is that any app can start another app’s component. For example, if you want the user to capture a photo with the device camera, there's probably another app that does that and your app can use it, instead of developing an activity to capture a photo yourself. You don't need to incorporate or even link to the code from the camera app. Instead, you can simply start the activity in the camera app that captures a photo. When complete, the photo is even returned to your app so you can use it. To the user, it seems as if the camera is actually a part of your app.

When the system starts a component, it starts the process for that app (if it's not already running) and instantiates the classes needed for the component. For example, if your app starts the activity in the camera app that captures a photo, that activity runs in the process that belongs to the camera app, not in your app's process. Therefore, unlike apps on most other systems, Android apps don't have a single entry point (there's no main()function, for example).

Because the system runs each app in a separate process with file permissions that restrict access to other apps, your app cannot directly activate a component from another app. The Android system, however, can. So, to activate a component in another app, you must deliver a message to the system that specifies your intent to start a particular component. The system then activates the component for you.

### **Activating Components**

Three of the four component types—activities, services, and broadcast receivers—are activated by an asynchronous message called an intent. Intents bind individual components to each other at runtime (you can think of them as the messengers that request an action from other components), whether the component belongs to your app or another.

An intent is created with an [Intent](http://developer.android.com/reference/android/content/Intent.html) object, which defines a message to activate either a specific component or a specific type of component—an intent can be either explicit or implicit, respectively.

For activities and services, an intent defines the action to perform (for example, to "view" or "send" something) and may specify the URI of the data to act on (among other things that the component being started might need to know). For example, an intent might convey a request for an activity to show an image or to open a web page. In some cases, you can start an activity to receive a result, in which case, the activity also returns the result in an[Intent](http://developer.android.com/reference/android/content/Intent.html) (for example, you can issue an intent to let the user pick a personal contact and have it returned to you—the return intent includes a URI pointing to the chosen contact).

For broadcast receivers, the intent simply defines the announcement being broadcast (for example, a broadcast to indicate the device battery is low includes only a known action string that indicates "battery is low").

The other component type, content provider, is not activated by intents. Rather, it is activated when targeted by a request from a [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html). The content resolver handles all direct transactions with the content provider so that the component that's performing transactions with the provider doesn't need to and instead calls methods on the [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html) object. This leaves a layer of abstraction between the content provider and the component requesting information (for security).

There are separate methods for activating each type of component:

* You can start an activity (or give it something new to do) by passing an [Intent](http://developer.android.com/reference/android/content/Intent.html) to [startActivity()](http://developer.android.com/reference/android/content/Context.html#startActivity(android.content.Intent)) or[startActivityForResult()](http://developer.android.com/reference/android/app/Activity.html#startActivityForResult(android.content.Intent, int)) (when you want the activity to return a result).
* You can start a service (or give new instructions to an ongoing service) by passing an [Intent](http://developer.android.com/reference/android/content/Intent.html) to[startService()](http://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)). Or you can bind to the service by passing an [Intent](http://developer.android.com/reference/android/content/Intent.html) to [bindService()](http://developer.android.com/reference/android/content/Context.html#bindService(android.content.Intent, android.content.ServiceConnection, int)).
* You can initiate a broadcast by passing an [Intent](http://developer.android.com/reference/android/content/Intent.html) to methods like [sendBroadcast()](http://developer.android.com/reference/android/content/Context.html#sendBroadcast(android.content.Intent)),[sendOrderedBroadcast()](http://developer.android.com/reference/android/content/Context.html#sendOrderedBroadcast(android.content.Intent, java.lang.String)), or [sendStickyBroadcast()](http://developer.android.com/reference/android/content/Context.html#sendStickyBroadcast(android.content.Intent)).
* You can perform a query to a content provider by calling [query()](http://developer.android.com/reference/android/content/ContentProvider.html#query(android.net.Uri, java.lang.String[], java.lang.String, java.lang.String[], java.lang.String)) on a [ContentResolver](http://developer.android.com/reference/android/content/ContentResolver.html).

For more information about using intents, see the [Intents and Intent Filters](http://developer.android.com/guide/components/intents-filters.html) document. More information about activating specific components is also provided in the following documents: [Activities](http://developer.android.com/guide/components/activities.html), [Services](http://developer.android.com/guide/components/services.html),[BroadcastReceiver](http://developer.android.com/reference/android/content/BroadcastReceiver.html) and [Content Providers](http://developer.android.com/guide/topics/providers/content-providers.html).

## The Manifest File

Before the Android system can start an app component, the system must know that the component exists by reading the app's AndroidManifest.xml file (the "manifest" file). Your app must declare all its components in this file, which must be at the root of the app project directory.

The manifest does a number of things in addition to declaring the app's components, such as:

* Identify any user permissions the app requires, such as Internet access or read-access to the user's contacts.
* Declare the minimum [API Level](http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels) required by the app, based on which APIs the app uses.
* Declare hardware and software features used or required by the app, such as a camera, bluetooth services, or a multitouch screen.
* API libraries the app needs to be linked against (other than the Android framework APIs), such as the [Google Maps library](http://code.google.com/android/add-ons/google-apis/maps-overview.html).
* And more

### **Declaring components**

The primary task of the manifest is to inform the system about the app's components. For example, a manifest file can declare an activity as follows:

<?xml version="1.0" encoding="utf-8"?>  
<manifest ... >  
    <application android:icon="@drawable/app\_icon.png" ... >  
        <activity android:name="com.example.project.ExampleActivity"  
                  android:label="@string/example\_label" ... >  
        </activity>  
        ...  
    </application>  
</manifest>

In the [<application>](http://developer.android.com/guide/topics/manifest/application-element.html) element, the android:icon attribute points to resources for an icon that identifies the app.

In the [<activity>](http://developer.android.com/guide/topics/manifest/activity-element.html) element, the android:name attribute specifies the fully qualified class name of the[Activity](http://developer.android.com/reference/android/app/Activity.html) subclass and the android:label attributes specifies a string to use as the user-visible label for the activity.

You must declare all app components this way:

* [<activity>](http://developer.android.com/guide/topics/manifest/activity-element.html) elements for activities
* [<service>](http://developer.android.com/guide/topics/manifest/service-element.html) elements for services
* [<receiver>](http://developer.android.com/guide/topics/manifest/receiver-element.html) elements for broadcast receivers
* [<provider>](http://developer.android.com/guide/topics/manifest/provider-element.html) elements for content providers

Activities, services, and content providers that you include in your source but do not declare in the manifest are not visible to the system and, consequently, can never run. However, broadcast receivers can be either declared in the manifest or created dynamically in code (as [BroadcastReceiver](http://developer.android.com/reference/android/content/BroadcastReceiver.html) objects) and registered with the system by calling [registerReceiver()](http://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)).

For more about how to structure the manifest file for your app, see [The AndroidManifest.xml File](http://developer.android.com/guide/topics/manifest/manifest-intro.html) documentation.

### **Declaring component capabilities**

As discussed above, in [Activating Components](http://developer.android.com/guide/components/fundamentals.html#ActivatingComponents), you can use an [Intent](http://developer.android.com/reference/android/content/Intent.html) to start activities, services, and broadcast receivers. You can do so by explicitly naming the target component (using the component class name) in the intent. However, the real power of intents lies in the concept of implicit intents. An implicit intent simply describes the type of action to perform (and, optionally, the data upon which you’d like to perform the action) and allows the system to find a component on the device that can perform the action and start it. If there are multiple components that can perform the action described by the intent, then the user selects which one to use.

The way the system identifies the components that can respond to an intent is by comparing the intent received to the *intent filters* provided in the manifest file of other apps on the device.

When you declare an activity in your app's manifest, you can optionally include intent filters that declare the capabilities of the activity so it can respond to intents from other apps. You can declare an intent filter for your component by adding an [<intent-filter>](http://developer.android.com/guide/topics/manifest/intent-filter-element.html) element as a child of the component's declaration element.

For example, if you've built an email app with an activity for composing a new email, you can declare an intent filter to respond to "send" intents (in order to send a new email) like this:

<manifest ... >  
    ...  
    <application ... >  
        <activity android:name="com.example.project.ComposeEmailActivity">  
            <intent-filter>  
                <action android:name="android.intent.action.SEND" />  
                <data android:type="\*/\*" />  
                <category android:name="android.intent.category.DEFAULT" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>

Then, if another app creates an intent with the [ACTION\_SEND](http://developer.android.com/reference/android/content/Intent.html#ACTION_SEND) action and pass it to [startActivity()](http://developer.android.com/reference/android/app/Activity.html#startActivity(android.content.Intent)), the system may start your activity so the user can draft and send an email.

For more about creating intent filters, see the [Intents and Intent Filters](http://developer.android.com/guide/components/intents-filters.html) document.

### **Declaring app requirements**

There are a variety of devices powered by Android and not all of them provide the same features and capabilities. In order to prevent your app from being installed on devices that lack features needed by your app, it's important that you clearly define a profile for the types of devices your app supports by declaring device and software requirements in your manifest file. Most of these declarations are informational only and the system does not read them, but external services such as Google Play do read them in order to provide filtering for users when they search for apps from their device.

For example, if your app requires a camera and uses APIs introduced in Android 2.1 ([API Level](http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels) 7), you should declare these as requirements in your manifest file like this:

<manifest ... >  
    <uses-feature android:name="android.hardware.camera.any"  
                  android:required="true" />  
    <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="19" />  
    ...  
</manifest>

Now, devices that do not have a camera and have an Android version lower than 2.1 cannot install your app from Google Play.

However, you can also declare that your app uses the camera, but does not require it. In that case, your app must set the [required](http://developer.android.com/guide/topics/manifest/uses-feature-element.html#required) attribute to "false" and check at runtime whether the device has a camera and disable any camera features as appropriate.

More information about how you can manage your app's compatibility with different devices is provided in the[Device Compatibility](http://developer.android.com/guide/practices/compatibility.html) document.

## 3. PROPOSED SYSTEM

### 3.1 Components of Proposed System

### 3.2 Software and Hardware Requirements

### 3.3 Proposed Architecture

## 4. CONCLUSION AND FUTURE WORKS

### 4.1 Conclusion

### It is necessary to keep the air inside tunnels clean. Each tunnel needs the most suitable ventilation system.

### 4.2 Future works

* Security for Platform and Encryption for transmission

### REFERENCE

<http://machinedesign.com/sensors/sensor-sense-hall-effect-current-sensors>

<http://www.engineersgarage.com/articles/humidity-sensor>

<http://www.rcjournal.com/contents/05.04/05.04.0507.pdf>

<http://www.agf.gov.bc.ca/resmgmt/publist/300Series/306420-1.pdf>

<http://www.oregon.gov/energy/cons/bus/dcv/docs/dcvguide.pdf>

<http://resources.hkedcity.net/downloadResource.php?rid=663161337&pid=1290088494>

<https://developer.android.com/training/index.html>

### AUTHOR’S PUBLICATION