

Enhancing women safety and Security Smart Identification System using Internet of Things (IOT)

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

DONDA YASWANTH SAI(Reg. No. 37110200)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

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

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **DONDA YASWANTH SAI (Reg.No. 37110200)** project who carried out the project entitled **"Enhancing women safety and Security Smart Identification System using Internet of Things (IOT)"** under my supervision from November 2020 to March 2021.

Internal Guide

Ms T. ANANDHI, M.E.,

Head of the Department

Dr. S. Vigneswari, M.E., Ph.D.,

Dr.L.LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **DONDA YASWANTH SAI** hereby declare that the Project Report entitled **“Enhancing women safety and Security Smart Identification System using Internet of Things (IOT)”** is done by me under the guidance of **Ms T. ANANDHI, M.E.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:08-04-2021

PLACE: CHENNAI

D.Yaswanth sai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms T. ANANDHI, M.E., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Abstract

The software or application has access to GPS and Messaging services which is pre-programmed in such a way that whenever it receives emergency signal, it can send help request along with the location co-ordinates to the nearest Police station, relatives and the people in the near radius who have application. This action enables help instantaneously from the Police as well as Public in the near radius who can reach the victim with great accuracy. The only thought haunting every girl is when they will be able to move freely on the streets even in odd hours without worrying about their security. This paper suggests a new perspective to use technology for women safety. Indian Women Are Harassed, Raped, Killed Every Day!!” That’s a way beyond HUGE number! We propose an idea which changes the way everyone thinks about women safety. A day when media broadcasts more of women’s achievements rather than harassment, it’s a feat achieved! Since we (humans) can’t respond aptly in critical situations, the need for a device which automatically senses and rescues the victim is the venture of our idea in this paper. We propose to have a device which is the integration of multiple devices, hardware comprises of a wearable “Smart band” which continuously communicates with Smart phone that has access to the internet. The application is programmed and loaded with all the required data which includes Human Behavior and reactions to different situations like anger, fear and anxiety. This generates a signal which is transmitted to the smart phone

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE No
	ABSTRACT	v
	LIST OF FIGURES	vii
1.	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 MANAGEMENT OF SECURITY	2
2.	LITERATURE SURVEY	3
	2. LITERATURE SURVEY	3
3.	SYSTEM DESIGN	5
	3.1 INTERNET OF THINGS	5
	3.2 INTRASTRUCTURE MANAGEMENT	12
	3.3 MEDICAL AND HEALTHCARE	14
	3.4 TRENDS AND CHARACTERISTICS	18
4.	HARDWARE SETUP DETAILS	32
	4.1 RASPBERRY PI	35
	4.2 PROCESSOR	37
	4.3 HARDWARE	39
	4.4 BLOCK DIAGRAM	42
	4.5 SOFTWARE SETUP DETAILS	64

4.6 EXPERIMENTAL RESULT	84
CONCLUSION	85
REFERENCES	88

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.4	BLOCK DIAGRAM	42
5.1	UBIDOTS LOGIN PAGE	85
5.2	USER INTERFACE OF VNC VIEWER	86
5.3	FILE SYSTEM IN VNC	86
5.4	DEVICE DATA	87
5.5	CLOUD DATA OF DEVICE IN UBIDOTS	87
5.6	LOCATION OF DEVICE	88

CHAPTER 1

INTRODUCTION

INTRODUCTION

This paper focuses on a security system that is designed merely to serve the purpose of providing security to women so that they never feel helpless while facing such social challenges. An advanced system can be built that can detect the location and health condition of person that will enable us to take action accordingly based on electronic gadgets like GPS receiver, body temperature sensor, GSM, Pulse rate sensor. We can make use of number of sensors to precisely detect the real time situation of the women in critical abusive situations. The heartbeat of a person in such situations is normally higher which helps make decisions along with other sensors like motion sensors to detect the abnormal motion of the women while she is victimized. The idea to develop a smart device for women is that it's completely comfortable and easy to use a compared with already existing women security solutions such as a separate garment, bulky belts and infamous mobile apps that are just very abstract and obsolete. The Smart band integrated with Smart phone has an added advantage so as to reduce the cost of the device and also in reduced size. The GPS and the GSM can be used of a smart phone. This also enables in reduced power use and that the watch can be installed with Bluetooth 4.0 BLE (Bluetooth Low Energy) which comes in handy for several days on a single shot of charge.

Development of the Internet of Things (IOT) is going on quicker, it is uniting a colossal number of points of interest, including devices, people and spots, to pass on and offer information, enhancing business quality and high ground and making new business opportunities. It is presently just a matter of time to see that the virtual and veritable match in a more cutting edge structure. Lately there has been a significant move in the IOT which moved from a unified structure to a perplexing network of decentralized smart devices.

This movement expanded networking and cloud enablement of a wide range of physical devices from machines through autos to home apparatuses. Through this change everything around us will be coordinated towards viable working, controlling, checking, reporting and data storing powered by smart chips. These will doubtlessly

going to make our life super straightforwardness. As IOT is based on a wide range of semiconductor technologies, including power management devices, sensors and microchips execution so security prerequisites change extensively starting with one application then onto the next. The way that the accomplishment of smart homes, connected autos and Industry 4.0 production lines depend all that much on client certainty, simple to-utilize, and safeguard security capacities. The more prominent the volume of delicate information exchange over the IOT is occurring the more noteworthy the danger of information and data fraud, device control, information adulteration, IP robbery and much server/system control. In this paper the talk will be on what is going ahead to make the IOT secured, what are the measures so far taken in development of dependable framework and what is the imminent answers for IOT in the setting of secured utilization of IOT devices in our regular life.

MANAGEMENT OF SECURITY

This is a security system that is designed to providing security to women so that they never feel helpless while facing such critical situations. An advanced system can be built that can detect the location and health condition of women that will enable us to take action accordingly.

We can make use of number of sensors to precisely detect the real time situation of the women in critical situations. The temperature of a women in such situations is normally higher which helps to make decisions.

This paper focuses on a security system that is designed merely to serve the purpose of providing security to women so that they never feel helpless while facing such social challenges. An advanced system can be built that can detect the location and health condition of person that will enable us to take action accordingly based on electronic gadgets like IOT, body temperature sensor , GSM, Pulse rate sensor, buzzer and node MCU. We can make use of number of sensors to precisely detect the real time situation of the women in critical abusive situations. The heartbeat of a person in such situations is normally higher which helps make decisions along with other sensors like motion sensors to detect the abnormal motion of the women while she is victimized.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY:

D. Mahesh Kumar et.al [1] in his paper on “**Healthcare Monitoring System Using Wireless Sensor Network**”. As of late, enormous endeavors have been made to utilize Wireless Sensor Networks (WSN) in medical services frameworks. Nonetheless, for most investigations, assignments, for example, sensor information handling, wellbeing state dynamic, and crisis messages are totally done by distant workers. The transmission and conveyance of a lot of information from sensors burns-through a lot of correspondence assets, loads far off workers, and defers choice time and notice time. In this article, we give a representation of an astute entryway that we have just utilized. The door is a stage for correspondence and administration the executives, particularly for the WSN home medical care framework.

Himadri Nath Saha et.al [2] in his paper on “**Health Monitoring using Internet of Things (IoT)**” proposed a life is a level of ease of use and metabolic capability of organic entities. For individuals, whenever they have gone through physical, mental, mental and social change, it is the force of people or networks to adjust and control themselves. Following the existence of a patient beneficiary can be an overwhelming assignment. Patients with intermittent ailments ought to be given extraordinary consideration and their friends and family ought to be instructed about their wellbeing at work. In this article, we will probably screen patients' wellbeing with the assistance of nervous system specialists and the Internet. The Internet is utilized to tell their friends and family if there is opposition. A wellbeing registration framework can follow a patient's pulse, pulse, pressure, temperature, and so forth.

K Ranjitha Pragnya1 et.al [3] in her paper on “**Wireless Sensor Network Based Healthcare Monitoring System for Homely Elders**”. In this article, we report on the best way to gauge the prosperity of the older dependent on the utilization of home machines associated with different hearing units. A locally situated conduct home checking program dependent on a remote sensor network incorporates execution examination of day-by-day exercises. MEMS, temperature and pressing factor sensors are utilized to decide the wellbeing status of more established individuals as per their everyday exercises. This archive presents the ZIGBEE nursing home observing system. The program depends on everyday exercises. Its sensor hubs

comprise of temperature sensors, MEMS sensor and sensor sensors, ATMEL microcontrollers, LCD modules, and ZIGBEE handset models working on the 2.45 GHZ recurrence band. The development of a clinical band.

Niket Patil et.al [4] in his paper on “**Health Monitoring and Tracking System for Soldiers Using Internet of Things (IoT)**”. This paper gives an account of a military wellbeing observing and checking framework dependent on Internet of Things (IoT). The proposed framework can be embedded into a fighter's body to follow their wellbeing and area utilizing GPS. This data will be moved to the control room through the Internet of Things. The proposed framework incorporates compact actual gadgets, sensors, and transmission modules. Consequently, by utilizing the proposed gear, a more affordable cycle can be performed to ensure valuable lives on the combat zone.

Ahmed Abdelgawad et.al [5] in his paper on “**IoT-Based Health Monitoring System for Active and Assisted Living**”. The Internet of Things (IoT) has been generally used to interface accessible clinical benefits and to give astute, solid and viable clinical benefits for the older. Observing of dynamic and helped living is one of the models that can utilize the Internet of Things to improve the personal satisfaction of more seasoned individuals. In this article, we propose redid IoT plan for wellbeing applications. Proposed structures gather information and move it to the cloud, where it is handled and examined. Reaction information dependent on investigation information can be sent back to the client. An illustration of a proposed development has been created to show its operational advantages.

B.Vijaylakshmi et.al [6] in her paper on “**Self-defense system for women safety with location tracking and SMS alerting through GSM network**”. This framework chapter explains a low-cost, quick-response mechanism for individuals, especially women, in which a woman in trouble may request assistance by pressing a button on a smart device. The Women's Self-Defense System resembles a wearable device for women. With innovation embedded into a small unit, it has the potential to benefit women. When a girl wearing this piece of technology as a watch or band is harassed or suspects that someone is about to harass her, When she clicks a button on her watch or device, or even when she falls, the details of the attack, as well as her body position and location, are sent as an SMS warning to a few pre-defined helplines.

Niti shree el.al [7] in her paper on “**A Review on IOT Based Smart GPS Device for Child and Women Safety**”. This paper proposes an Android based solution to aid parents to track their children in real time. Different devices are connected with a single device through channels of internet. The concerned device is connected to server via internet. The device can be used by parents to track their children in real time or for women safety.

Ms Deepali M. Bhavale el.al [8] in her paper on “**IOT Based Unified Approach for Women and Children Security Using Wireless and GPS**”. The proposed system intends to a device wireless technique in the form of embedded device namely Arduino for women that will serve the purpose of alerts and way of communicating with secure channels and it captures the image using electronic camera. There are many android applications for women safety but they as not as much as efficient. So to solve this issue of women safety we develop a wireless sensor kit which is easy to use and which is efficient to provide help to that victim.

Hung Nguyen el.al [9] in his paper on “**Using Inertial Sensors to Automatically Detect and Segment Activities of Daily Living in People with Parkinson’s Disease**”. Wearable sensors such as inertial measure- ment units (IMUs) have been widely used to measure the quantity of physical activities during daily living in healthy and people with movement disorders through activity clas- sification. These sensors have the potential to provide valu- able information to evaluate the quality of the movement during the activities of daily living (ADL), such as walking, sitting down, and standing up, which could help clinicians to monitor rehabilitation and pharmaceutical interventions.

Prof. R.A.Jain1 el.al [10] in his paper on “**Women’s safety using IOT**”. Nowadays, women and children safety is a prime issue of our society. The count of the victim are increasing day by day. In this paper, we are proposing a model which will help to ensure the safety of women and children’s all over the global. We have used different sensors like heartbeat sensor, temperature sensor, and accelerometer sensor for detecting heartbeat, temperature and sudden change in motion of user. We have also used GPS which will help to detect location of the device. GSM used in the model is used to send alert message to guardians, relatives and police station. We have

proposed IOT (internet of things) based device which will help to continuously monitor values of different sensors and GPS used in device.

CHAPTER 3

SYSTEM DESIGN

3. SYSTEM DESIGN

INTERNET OF THINGS:

The Internet of things (IOT) is the inter-networking of physical devices,

vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IOT-GSI) defined the IOT as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" and for these purposes a "thing" is "an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks". The IOT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention.

When IOT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IOT will consist of about 30 billion objects by 2020. Typically, IOT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications. The interconnection of these embedded devices (including smart objects), is expected to usher in automation in nearly all fields, while also enabling advanced applications like a smart grid, and expanding to areas such as smart cities. "Things", in the IOT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist firefighters in search and rescue operations. Legal scholars suggest regarding "Things" as an "inextricable mixture of hardware, software, data and service".

These devices collect useful data with the help of various existing technologies

and then autonomously flow the data between other devices. Current market examples include home automation (also known as smart home devices) such as the control and automation of lighting, heating (like smart thermostat), ventilation, air conditioning (HVAC) systems, and appliances such as washer/dryers, robotic vacuums, air purifiers, ovens, or refrigerators/freezers that use Wi-Fi for remote monitoring. As well as the expansion of Internet-connected automation into a plethora of new application areas, IOT is also expected to generate large amounts of data from diverse locations, with the consequent necessity for quick aggregation of the data, and an increase in the need to index, store, and process such data more effectively. IOT is one of the platforms of today's Smart City, and Smart Energy Management Systems.

History

As of 2016, the vision of the Internet of things has evolved due to a convergence of multiple technologies, including ubiquitous wireless communication, real-time analytics, machine learning, commodity sensors, and embedded systems. This means that the traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things (IoT). The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold.

Mark Weiser's seminal 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of IoT. In 1994 Reza Raji described the concept in IEEE Spectrum as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1996 several companies proposed solutions like Microsoft's at Work or Novell's NEST. However, only in 1999 did the field start gathering momentum. Bill Joy envisioned Device to Device (D2D) communication as part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999. The concept of the Internet of things became popular in 1999, through the Auto-ID Centre at MIT and related market-analysis publications. Radio-frequency identification (RFID) was seen by Kevin Ashton (one of the founders of the original Auto-ID Centre) as a prerequisite

for the Internet of things at that point.

Ashton prefers the phrase "Internet for Things. "If all objects and people in daily life were equipped with identifiers, computers could manage and inventory them. Besides using RFID, the tagging of things may be achieved through such technologies as near field communication, barcodes, QR codes and digital watermarking. In its original interpretation, one of the first consequences of implementing the Internet of things by equipping all objects in the world with minuscule identifying devices or machine-readable identifiers would be to transform daily life.

For instance, instant and ceaseless inventory control would become ubiquitous. A person's ability to interact with objects could be altered remotely based on immediate or present needs, in accordance with existing end-user agreements. For example, such technology could grant motion-picture publishers much more control over end-user private devices by remotely enforcing copyright restrictions and digital rights management, so the ability of a customer who bought a Blu-ray disc to watch the movie could become dependent on the copyright holder's decision, similar to Circuit City's failed DIVX.

Applications

According to Gartner, Inc. (a technology research and advisory corporation), there will be nearly 20.8 billion devices on the Internet of things by 2020. ABI Research estimates that more than 30 billion devices will be wirelessly connected to the Internet of things by 2020. As per a 2014 survey and study done by Pew Research Internet Project, a large majority of the technology experts and engaged Internet users who responded—83 percent—agreed with the notion that the Internet/Cloud of Things, embedded and wearable computing (and the corresponding dynamic systems will have widespread and beneficial effects by 2025. As such, it is clear that the IoT will consist of a very large number of devices being connected to the Internet. In an active move to accommodate new and emerging technological innovation, the UK Government, in their 2015 budget, allocated £40,000,000 towards research into the Internet of things. The former British Chancellor of the Exchequer George Osborne, posited that the Internet of things is the next stage of the information revolution and referenced the inter-connectivity of everything from urban transport to medical devices

to household appliances. The ability to network embedded devices with limited CPU, memory and power resources means that IoT finds applications in nearly every field. Such systems could be in charge of collecting information in settings ranging from natural ecosystems to buildings and factories, thereby finding applications in fields of environmental sensing and urban planning. On the other hand, IoT systems could also be responsible for performing actions, not just sensing things.

Intelligent shopping systems, for example, could monitor specific users' purchasing habits in a store by tracking their specific mobile phones. These users could then be provided with special offers on their favorite products, or even location of items that they need, which their fridge has automatically conveyed to the phone. Additional examples of sensing and actuating are reflected in applications that deal with heat, water, electricity and energy management, as well as cruise-assisting transportation systems.

Other applications that the Internet of things can provide is enabling extended home security features and home automation. The concept of an "Internet of living things" has been proposed to describe networks of biological sensors that could use cloud-based analyses to allow users to study DNA or other molecules. However, the application of the IoT is not only restricted to these areas. Other specialized use cases of the IoT may also exist. An overview of some of the most prominent application areas is provided here.

Media

In order to hone the manner in which things, media and big data are interconnected, it is first necessary to provide some context into the mechanism used for media process. It has been suggested by Nick Couldry and Joseph Turow that practitioners in media approach big data as many actionable points of information about millions of individuals. The industry appears to be moving away from the traditional approach of using specific media environments such as newspapers, magazines, or television shows and instead tap into consumers with technologies that reach targeted people at optimal times in optimal locations. The ultimate aim is of course to serve, or convey, a message or content that is (statistically speaking) in line with the consumer's mind-set. For example, publishing environments are increasingly

tailoring the messages (advertisements) and content (articles) to appeal to consumers that have been exclusively gleaned through various data-mining activities. The media industries process big data in a dual, interconnected manner: Targeting of consumers (for advertising by marketers) Data-capture Thus, the Internet of things creates an opportunity to measure, collect and analyses an ever-increasing variety of behavioral statistics. Cross-correlation of this data could revolutionize the targeted marketing of products and services.

For example, as noted by Danny Meadows-Klue, the combination of analytics for conversion tracking with behavioral targeting has unlocked a new level of precision that enables display advertising to be focused on the devices of people with relevant interests. Big data and the IoT work in conjunction. From a media perspective, data is the key derivative of device interconnectivity, whilst being pivotal in allowing clearer accuracy in targeting. The Internet of things therefore transforms the media industry, companies and even governments, opening up a new era of economic growth and competitiveness. The wealth of data generated by this industry (i.e. big data) will allow practitioners in advertising and media to gain an elaborate layer on the present targeting mechanisms used by the industry.

Environmental monitoring

Environmental monitoring applications of the IoT typically use sensors to assist in environmental protection by monitoring air or water quality, atmospheric or soil conditions, and can even include areas like monitoring the movements of wildlife and their habitats. Development of resource constrained devices connected to the Internet also means that other applications like earthquake or tsunami early-warning systems can also be used by emergency services to provide more effective aid. IoT devices in this application typically span a large geographic area and can also be mobile. It has been argued that the standardization IoT brings to wireless sensing will revolutionize this area.

Infrastructure management

Monitoring and controlling operations of urban and rural infrastructures like bridges, railway tracks, on- and offshore- wind-farms is a key application of the IoT. The IoT infrastructure can be used for monitoring any events or changes in structural

conditions that can compromise safety and increase risk. It can also be used for scheduling repair and maintenance activities in an efficient manner, by coordinating tasks between different service providers and users of these facilities. IoT devices can also be used to control critical infrastructure like bridges to provide access to ships. Usage of IoT devices for monitoring and operating infrastructure is likely to improve incident management and emergency response coordination, and quality of service, up-times and reduce costs of operation in all infrastructure related areas. Even areas such as waste management can benefit from automation and optimization that could be brought in by the IoT.

Manufacturing Network

Control and management of manufacturing equipment, asset and situation management, or manufacturing process control bring the IoT within the realm of industrial applications and smart manufacturing as well. The IoT intelligent systems enable rapid manufacturing of new products, dynamic response to product demands, and real-time optimization of manufacturing production and supply chain networks, by networking machinery, sensors and control systems together. Digital control systems to automate process controls, operator tools and service information systems to optimize plant safety and security are within the purview of the IoT. But it also extends itself to asset management via predictive maintenance, statistical evaluation, and measurements to maximize reliability.

Smart industrial management systems can also be integrated with the Smart Grid, thereby enabling real-time energy optimization. Measurements, automated controls, plant optimization, health and safety management, and other functions are provided by a large number of networked sensors. The National Science Foundation established an Industry/University Cooperative Research Center on intelligent maintenance systems (IMS) in 2001 with a research focus to use IoT-based predictive analytics technologies to monitor connected machines and to predict machine degradation, and further to prevent potential failures. The vision to achieve near-zero breakdown using IoT-based predictive analytics led the future development of e-manufacturing and maintenance activities. The term IOT (Industrial Internet of Things) is often encountered in the manufacturing industries, referring to the industrial subset of the IoT. IOT in manufacturing could generate so much business value that it will

eventually lead to the fourth industrial revolution, so the so-called Industry 4.0. It is estimated that in the future, successful companies will be able to increase their revenue through Internet of things by creating new business models and improve productivity, exploit analytics for innovation, and transform workforce. The potential of growth by implementing IOT will generate \$12 trillion of global GDP by 2030 while connectivity and data acquisition are imperative for IOT, they should not be the purpose, rather the foundation and path to something bigger. Among all the technologies, predictive maintenance is probably a relatively "easier win" since it is applicable to existing assets and management systems.

The objective of intelligent maintenance systems is to reduce unexpected downtime and increase productivity. And to realize that alone would generate around up to 30% over total maintenance costs. Industrial big data analytics will play a vital role in manufacturing asset predictive maintenance, although that is not the only capability of industrial big data. Cyber-physical systems (CPS) is the core technology of industrial big data and it will be an interface between human and the cyber world. Cyber-physical systems can be designed by following the 5C (connection, conversion, cyber, cognition, and configuration) architecture, and it will transform the collected data into actionable information, and eventually interfere with the physical assets to optimize processes. An IoT-enabled intelligent system of such cases has been demonstrated by the NSF Industry/University Collaborative Research Center for Intelligent Maintenance Systems (IMS) at University of Cincinnati on a band saw machine in IMTS 2014 in Chicago.

Band saw machines are not necessarily expensive, but the band saw belt expenses are enormous since they degrade much faster. However, without sensing and intelligent analytics, it can be only determined by experience when the band saw belt will actually break. The developed prognostics system will be able to recognize and monitor the degradation of band saw belts even if the condition is changing, so that users will know in near real time when is the best time to replace band saw. This will significantly improve user experience and operator safety, and save costs on replacing band saw belts before they actually break. The developed analytical algorithms were realized on a cloud server, and was made accessible via the Internet and on mobile devices.

Energy management

Integration of sensing and actuation systems, connected to the Internet, is likely to optimize energy consumption as a whole. It is expected that IoT devices will be integrated into all forms of energy consuming devices (switches, power outlets, bulbs, televisions, etc.) and be able to communicate with the utility supply company in order to effectively balance power generation and energy usage. Such devices would also offer the opportunity for users to remotely control their devices, or centrally manage them via a cloud based interface, and enable advanced functions like scheduling (e.g., remotely powering on or off heating systems, controlling ovens, changing lighting conditions etc.).

Besides home based energy management, the IoT is especially relevant to the Smart Grid since it provides systems to gather and act on energy and power-related information in an automated fashion with the goal to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity. Using advanced metering infrastructure (AMI) devices connected to the Internet backbone, electric utilities can not only collect data from end-user connections, but also manage other distribution automation devices like transformers and recloses.

Medical and healthcare

IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fit bit electronic wristbands, or advanced hearing aids. Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is attempting to get up. It can also adjust itself to ensure appropriate pressure and support is applied to the patient without the manual interaction of nurses.

Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well. Other consumer devices to encourage healthy living, such as, connected

scales or wearable heart monitors, are also a possibility with the IoT. More and more end-to-end health monitoring IoT platforms are coming up for antenatal and chronic patients, helping one manage health vitals and recurring medication requirements.

Building and home automation

IoT devices can be used to monitor and control the mechanical, electrical and electronic systems used in various types of buildings (e.g., public and private, industrial, institutions, or residential) in home automation and building automation systems. In this context, three main areas are being covered in literature: The integration of the internet with building energy management systems in order to create energy efficient and IOT driven “smart buildings”. The possible means of real-time monitoring for reducing energy consumption and monitoring occupant behaviors. The integration of smart devices in the built environment and how they might be used in future applications.

Transportation

The IoT can assist in integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these components of a transport system enables inter and intra vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistic and fleet management, vehicle control, and safety and road assistance.

Metropolitan scale deployments

There are several planned or ongoing large-scale deployments of the IoT, to enable better management of cities and systems. For example, Song do, South Korea, the first of its kind fully equipped and wired smart city, is near completion. Nearly everything in this city is planned to be wired, connected and turned into a constant stream of data that would be monitored and analyzed by an array of computers with little, or no human intervention. Another application is a currently undergoing project in Santander, Spain. For this deployment, two approaches have been adopted. This city of 180,000 inhabitants, has already seen 18,000 city application downloads for their smartphones. This application is connected to 10,000 sensors that enable services like

parking search, environmental monitoring, and digital city agenda among others. City context information is used in this deployment so as to benefit merchants through a spark deals mechanism based on city behavior that aims at maximizing the impact of each notification. Other examples of large-scale deployments underway include the Sino-Singapore Guangzhou Knowledge City; work on improving air and water quality, reducing noise pollution, and increasing transportation efficiency in San Jose, California; and smart traffic management in western Singapore.

French company, Sigfox, commenced building an ultra-narrowband wireless data network in the San Francisco Bay Area in 2014, the first business to achieve such a deployment in the U.S. It subsequently announced it would set up a total of 4000 base stations to cover a total of 30 cities in the U.S. by the end of 2016, making it the largest IoT network coverage provider in the country thus far. Another example of a large deployment is the one completed by New York Waterways in New York City to connect all the city's vessels and be able to monitor them live 24/7. The network was designed and engineered by Fluid mesh Networks, a Chicago-based company developing wireless networks for critical applications. The NYWW network is currently providing coverage on the Hudson River, East River, and Upper New York Bay. With the wireless network in place, NY Waterway is able to take control of its fleet and passengers in a way that was not previously possible. New applications can include security, energy and fleet management, digital signage, public Wi-Fi, paperless ticketing and others.

Consumer application

A growing portion of IoT devices are created for consumer use. Examples of consumer applications include connected car, entertainment, residences and smart homes, wearable technology, quantified self, connected health, and smart retail. Consumer IoT provides new opportunities for user experience and interfaces. Some consumer applications have been criticized for their lack of redundancy and their inconsistency, leading to a popular parody known as the "Internet of Shit." Companies have been criticized for their rush into IoT, creating devices of questionable value, and not setting up stringent security standards.

Unique addressability of things

The original idea of the Auto-ID Center is based on RFID-tags and unique

identification through the Electronic Product Code however this has evolved into objects having an IP address or URI. An alternative view, from the world of the Semantic Web focuses instead on making all things (not just those electronic, smart, or RFID-enabled) addressable by the existing naming protocols, such as URI. The objects themselves do not converse, but they may now be referred to by other agents, such as powerful centralized servers acting for their human owners.

Integration with the Internet implies that devices will use an IP address as a unique identifier. Due to the limited address space of IPv4 (which allows for 4.3 billion unique addresses), objects in the IoT will have to use the next generation of the Internet protocol (IPv6) to scale to the extremely large address space required. Internet of things devices additionally will benefit from the stateless address auto-configuration present in IPv6, as it reduces the configuration overhead on the hosts, and the IETF 6LoWPAN header compression. To a large extent, the future of the Internet of things will not be possible without the support of IPv6; and consequently the global adoption of IPv6 in the coming years will be critical for the successful development of the IoT in the future. A combination of these ideas can be found in the current GS1/EPCglobal EPC Information Services (EPCIS) specifications. This system is being used to identify objects in industries ranging from aerospace to fast moving consumer products and transportation logistics.

Trends and characteristics

Intelligence:

Ambient intelligence and autonomous control are not part of the original concept of the Internet of things. Ambient intelligence and autonomous control do not necessarily require Internet structures, either. However, there is a shift in research to integrate the concepts of the Internet of things and autonomous control, with initial outcomes towards this direction considering objects as the driving force for autonomous IoT. In the future the Internet of things may be a non-deterministic and open network in which auto-organized or intelligent entities (Web services, SOA components), virtual objects (avatars) will be interoperable and able to act independently (pursuing their own objectives or shared ones) depending on the context, circumstances or environments. Autonomous behavior through the collection and reasoning of context information as well as the objects ability to detect changes in

the environment, faults affecting sensors and introduce suitable mitigation measures constitute a major research trend, clearly needed to provide credibility to the IoT technology. Modern IoT products and solutions in the marketplace use a variety of different technologies to support such context-aware automation but more sophisticated forms of intelligence are requested to permit sensor units to be deployed in real environments.

Architecture

The system will likely be an example of event-driven architecture, bottom-up made (based on the context of processes and operations, in real-time) and will consider any subsidiary level. Therefore, model driven and functional approaches will coexist with new ones able to treat exceptions and unusual evolution of processes (multi-agent systems, B-ADSc, etc.). In an Internet of things, the meaning of an event will not necessarily be based on a deterministic or syntactic model but would instead be based on the context of the event itself: this will also be a semantic web. Consequently, it will not necessarily need common standards that would not be able to address every context or use: some actors (services, components, avatars) will accordingly be self-referenced and, if ever needed, adaptive to existing common standards (predicting everything would be no more than defining a "global finality" for everything that is just not possible with any of the current top-down approaches and standardizations).

Building on top of the Internet of things, the web of things is an architecture for the application layer of the Internet of things looking at the convergence of data from IoT devices into Web applications to create innovative use-cases. In order to program and control the flow of information in the Internet of things, a predicted architectural direction is being called BPM Everywhere which is a blending of traditional process management with process mining and special capabilities to automate the control of large numbers of coordinated devices.

Network architecture

The Internet of things requires huge scalability in the network space to handle the surge of devices. IETF 6LoWPAN would be used to connect devices to IP networks. With billions of devices being added to the Internet space, IPv6 will play a major role in handling the network layer scalability. IETF's Constrained Application

Protocol, ZeroMQ, and MQTT would provide lightweight data transport. "MQ" in "MQTT" came from IBM's MQ Series message queuing product line. Fog computing is a viable alternative to prevent such large burst of data flow through Internet. The edge devices' computation power can be used to analyses and process data, thus providing easy real time scalability.

Complexity

In semi-open or closed loops (i.e. value chains, whenever a global finality can be settled) IoT will often be considered and studied as a complex system due to the huge number of different links, interactions between autonomous actors, and its capacity to integrate new actors. At the overall stage (full open loop) it will likely be seen as a chaotic environment (since systems always have finality). As a practical approach, not all elements in the Internet of things run in a global, public space. Subsystems are often implemented to mitigate the risks of privacy, control and reliability. For example, Domestic Robotics (Domestics) running inside a smart home might only share data within and be available via a local network.

Size considerations

The Internet of things would encode 50 to 100 trillion objects, and be able to follow the movement of those objects. Human beings in surveyed urban environments are each surrounded by 1000 to 5000 track able objects.

Space considerations

In the Internet of things, the precise geographic location of a thing—and also the precise geographic dimensions of a thing—will be critical. Therefore, facts about a thing, such as its location in time and space, have been less critical to track because the person processing the information can decide whether or not that information was important to the action being taken, and if so, add the missing information (or decide to not take the action). (Note that some things in the Internet of things will be sensors, and sensor location is usually important. The GeoWeb and Digital Earth are promising applications that become possible when things can become organized and connected by location. However, the challenges that remain include the constraints of variable spatial scales, the need to handle massive amounts of data, and an indexing for fast search and neighbor operations. In the Internet of things, if things are able to take

actions on their own initiative, this human-centric mediation role is eliminated. Thus, the time-space context that we as humans take for granted must be given a central role in this information ecosystem. Just as standards play a key role in the Internet and the Web, geospatial standards will play a key role in the Internet of things.

Sectors

There are three core sectors of the IoT: enterprise, home, and government, with the Enterprise Internet of Things (EIoT) being the largest of the three. By 2019, the EIoT sector is estimated to account for nearly 40% or 9.1 billion devices.

A Solution to "basket of remotes"

According to the CEO of Cisco, the commercial opportunity for "connected products ranging from cars to household goods" is expected to be a \$USD 19 trillion. Many IoT devices have a potential to take a piece of this market. Jean-Louis Gassée (Apple initial alumni team, and BeOS co-founder) has addressed this topic in an article on Monday Note, where he predicts that the most likely problem will be what he calls the "basket of remotes" problem, where we'll have hundreds of applications to interface with hundreds of devices that don't share protocols for speaking with one another.

There are multiple approaches to solve this problem, one of them called the "predictive interaction", where cloud or fog based decision makers will predict the user's next action and trigger some reaction. For user interaction, new technology leaders are joining forces to create standards for communication between devices. While AllJoyn alliance is composed of the top 20 World technology leaders, there are also big companies that promote their own protocol like CCF from Intel. Manufacturers are becoming more conscious of this problem, and many companies have begun releasing their devices with open APIs. Many of these APIs are used by smaller companies looking to take advantage of quick integration.

Frameworks

IoT frameworks might help support the interaction between "things" and allow for more complex structures like distributed computing and the development of distributed applications. Currently, some IoT frameworks seem to focus on real-time data logging solutions, offering some basis to work with many "things" and have them

interact. Future developments might lead to specific software-development environments to create the software to work with the hardware used in the Internet of things. Companies are developing technology platforms to provide this type of functionality for the Internet of things.

Newer platforms are being developed, which add more intelligence. REST is a scalable architecture that allows things to communicate over Hypertext Transfer Protocol and is easily adopted for IoT applications to provide communication from a thing to a central web server. MQTT is a publish-subscribe architecture on top of TCP/IP that allows bidirectional communication between a thing and an MQTT broker.

Enabling technologies for IoT

There are many technologies that enable IoT. Crucial to the field is the network used to communicate between devices of an IoT installation, a role that several wireless or wired technologies may fulfil:

Short-range wireless

- Bluetooth low energy (BLE) – Specification providing a low power variant to classic Bluetooth with a comparable communication range.
- Light-Fidelity (Li-Fi) – Wireless communication technology similar to the Wi-Fi standard, but using visible light communication for increased bandwidth.
- Near-field communication (NFC) – Communication protocols enabling two electronic devices to communicate within a 4 cm range.
- QR codes and barcodes – Machine-readable optical tags that store information about the item to which they are attached.
- Radio-frequency identification (RFID) – Technology using electromagnetic fields to read data stored in tags embedded in other items.
- Thread – Network protocol based on the IEEE 802.15.4 standard, similar to Zigbee, providing IPv6 addressing.
- Transport Layer Security (network protocol) TLS – Network security protocol.
- Wi-Fi – Widely used technology for local area networking based on the IEEE 802.11 standard, where devices may communicate through a shared access point.

- Wi-Fi Direct – Variant of the Wi-Fi standard for peer-to-peer communication, eliminating the need for an access point.
- Z-Wave – Communication protocol providing short-range, low-latency data transfer at rates and power consumption lower than Wi-Fi. Used primarily for home automation.
- Zigbee – Communication protocols for personal area networking based on the IEEE 802.15.4 standard, providing low power consumption, low data rate, low cost, and high throughput.

Medium-range wireless

- HaLow – Variant of the Wi-Fi standard providing extended range for low-power communication at a lower data rate.
- LTE-Advanced – High-speed communication specification for mobile networks. Provides enhancements to the LTE standard with extended coverage, higher throughput, and lower latency.

Long-range wireless

- Low-power wide-area networking (LPWAN) – Wireless networks designed to allow long-range communication at a low data rate, reducing power and cost for transmission.
- Very small aperture terminal (VSAT) – Satellite communication technology using small dish antennas for narrowband and broadband data.

Wired

- Ethernet – General purpose networking standard using twisted pair and fiber optic links in conjunction with hubs or switches.
- Multimedia over Coax Alliance (MoCA) – Specification enabling whole-home distribution of high definition video and content over existing coaxial cabling.
- Power-line communication (PLC) – Communication technology using electrical wiring to carry power and data. Specifications such as Home Plug utilize PLC for networking IoT devices.

Simulation

IoT modeling and simulation (and emulation) is typically carried out at the design stage before deployment of the network. Network simulators like OPNET, NetSim and NS2 can be used to simulate IoT networks. Digital Twins may also be implemented to produce updates on the status and health of an asset, based upon sensor readings

integrated with a computational model of the asset.

Politics and civic engagement

Some scholars and activists argue that the IoT can be used to create new models of civic engagement if device networks can be open to user control and interoperable platforms. Philip N. Howard, a professor and author, writes that political life in both democracies and authoritarian regimes will be shaped by the way the IoT will be used for civic engagement. For that to happen, he argues that any connected device should be able to divulge a list of the "ultimate beneficiaries" of its sensor data and that individual citizens should be able to add new organizations to the beneficiary list. In addition, he argues that civil society groups need to start developing their IoT strategy for making use of data and engaging with the public.

Government regulation on IoT

One of the key drivers of the IoT is data. The success of the idea of connecting devices to make them more efficient is dependent upon access to and storage & processing of data. For this purpose, companies working on IoT collect data from multiple sources and store it in their cloud network for further processing. This leaves the door wide open for privacy and security dangers and single point vulnerability of multiple systems.

The other issues pertain to consumer choice and ownership of data and how it is used. Presently the regulators have shown more interest in protecting the first three issues identified above. Current regulatory environment: A report published by the Federal Trade Commission (FTC) in January 2015 made the following three recommendations:

- Data security – At the time of designing IoT companies should ensure that data collection, storage and processing would be secure at all times. Companies should adopt a “defense in depth” approach and encrypt data at each stage.
- Data consent – users should have a choice as to what data they share with IoT companies and the users must be informed if their data gets exposed.

Data minimization – IoT companies should collect only the data they need and retain the collected information only for a limited time.

However, the FTC stopped at just making recommendations for now. According to an FTC analysis, the existing framework, consisting of the FTC Act, the Fair Credit Reporting Act, and the Children's Online Privacy Protection Act, along with developing consumer education and business guidance, participation in multistakeholder efforts and advocacy to other agencies at the federal, state and local level, is sufficient to protect consumer rights.

A resolution passed by the Senate in March 2015, is already being considered by the Congress. This resolution recognized the need for formulating a National Policy on IoT and the matter of privacy, security and spectrum. Furthermore, to provide an impetus to the IoT ecosystem, in March 2016, a bipartisan group of four Senators proposed a bill, The Developing Innovation and Growing the Internet of Things (DIGIT) Act, to direct the Federal Communications Commission to assess the need for more spectrum to connect IoT devices

Several standards for the IoT industry are actually being established relating to automobiles because most concerns arising from use of connected cars apply to healthcare devices as well. In fact, the National Highway Traffic Safety Administration (NHTSA) is preparing cyber security guidelines and a database of best practices to make automotive computer systems more secure

Criticism and controversies

Platform fragmentation

IoT suffers from platform fragmentation and lack of technical standards a situation where the variety of IoT devices, in terms of both hardware variations and differences in the software running on them, makes the task of developing applications that work consistently between different inconsistent technology ecosystems hard. Customers may be hesitant to bet their IoT future on a proprietary software or hardware devices that uses proprietary protocols that may fade or become difficult to customize and interconnect. IoT's amorphous computing nature is also a problem for security, since patches to bugs found in the core operating system often do not reach users of older and lower-price devices. One set of researchers say that the failure of vendors to support older devices with patches and updates leaves more than 87% of active devices vulnerable.

Privacy, autonomy, and control

Philip N. Howard, a professor and author, writes that the Internet of things offers immense potential for empowering citizens, making government transparent, and broadening information access. Howard cautions, however, that privacy threats are enormous, as is the potential for social control and political manipulation. Concerns about privacy have led many to consider the possibility that big data infrastructures such as the Internet of things and Data Mining are inherently incompatible with privacy. Writer Adam Greenfield claims that these technologies are not only an invasion of public space but are also being used to perpetuate normative behavior, citing an instance of billboards with hidden cameras that tracked the demographics of passers-by who stopped to read the advertisement. The Internet of Things Council compared the increased prevalence of digital surveillance due to the Internet of things to the conceptual panopticon described by Jeremy Bentham in the 18th Century.

The assertion was defended by the works of French philosophers Michel Foucault and Gilles Deleuze. In *Discipline and Punish: The Birth of the Prison* Foucault asserts that the panopticon was a central element of the discipline society developed during the Industrial Era. Foucault also argued that the discipline systems established in factories and school reflected Bentham's vision of panopticism. In his 1992 paper "Postscripts on the Societies of Control," Deleuze wrote that the discipline society had transitioned into a control society, with the computer replacing the panopticon as an instrument of discipline and control while still maintaining the qualities similar to that of panopticism.

The privacy of households could be compromised by solely analyzing smart home network traffic patterns without dissecting the contents of encrypted application data, yet a synthetic packet injection scheme can be used to safely overcome such invasion of privacy. Peter-Paul Verbeek, a professor of philosophy of technology at the University of Twente, Netherlands, writes that technology already influences our moral decision making, which in turn affects human agency, privacy and autonomy. He cautions against viewing technology merely as a human tool and advocates instead to consider it as an active agent. Justin Brookman, of the Center for Democracy and Technology, expressed concern regarding the impact of IoT on consumer privacy, saying that "There are some people in the commercial space who say, 'Oh, big data

— well, let's collect everything, keep it around forever, we'll pay for somebody to think about security later.' The question is whether we want to have some sort of policy framework in place to limit that." Tim O'Reilly believes that the way companies sell the IoT devices on consumers are misplaced, disputing the notion that the IoT is about gaining efficiency from putting all kinds of devices online and postulating that "IoT is really about human augmentation.

The applications are profoundly different when you have sensors and data driving the decision-making." Editorials at WIRED have also expressed concern, one stating "What you're about to lose is your privacy. Actually, it's worse than that. You aren't just going to lose your privacy, you're going to have to watch the very concept of privacy be rewritten under your nose." The American Civil Liberties Union (ACLU) expressed concern regarding the ability of IoT to erode people's control over their own lives.

The ACLU wrote that "There's simply no way to forecast how these immense powers – disproportionately accumulating in the hands of corporations seeking financial advantage and governments craving ever more control – will be used. Chances are big data and the Internet of Things will make it harder for us to control our own lives, as we grow increasingly transparent to powerful corporations and government institutions that are becoming more opaque to us".

In response to rising concerns about privacy and smart technology, in 2007 the British Government stated it would follow formal Privacy by Design principles when implementing their smart metering program. The program would lead to replacement of traditional power meters with smart power meters, which could track and manage energy usage more accurately. However the British Computer Society is doubtful these principles were ever actually implemented. In 2009 the Dutch Parliament rejected a similar smart metering program, basing their decision on privacy concerns. The Dutch program later revised and passed in 2011.

Data storage and analytics

A challenge for producers of IoT applications is to clean, process and interpret the vast amount of data which is gathered by the sensors. There is a solution proposed

for the analytics of the information referred to as Wireless Sensor Networks. These networks share data among sensor nodes that are send to a distributed system for the analytics of the sensory data.

Another challenge is the storage of this bulk data. Depending on the application there could be high data acquisition requirements which in turn lead to high storage requirements. Currently the internet is already responsible for 5% of the total energy generated and this consumption will increase significantly when we start utilizing applications with multiple embedded sensors.

Security:

Concerns have been raised that the Internet of things is being developed rapidly without appropriate consideration of the profound security challenges involved and the regulatory changes that might be necessary. Most of the technical security issues are similar to those of conventional servers, workstations and smartphones, but the firewall, security update and anti-malware systems used for those are generally unsuitable for the much smaller, less capable, IoT devices.

According to the Business Insider Intelligence Survey conducted in the last quarter of 2014, 39% of the respondents said that security is the biggest concern in adopting Internet of things technology. In particular, as the Internet of things spreads widely, cyber-attacks are likely to become an increasingly physical (rather than simply virtual) threat. In a January 2014 article in Forbes, cyber-security columnist Joseph Steinberg listed many Internet-connected appliances that can already "spy on people in their own homes" including televisions, kitchen appliances, cameras, and thermostats. Computer-controlled devices in automobiles such as brakes, engine, locks, hood and trunk releases, horn, heat, and dashboard have been shown to be vulnerable to attackers who have access to the on-board network.

In some cases, vehicle computer systems are Internet-connected, allowing them to be exploited remotely. By 2008 security researchers had shown the ability to remotely control pacemakers without authority. Later hackers demonstrated remote control of insulin pumps and implantable cardioverter defibrillators. David Pogue wrote that some recently published reports about hackers remotely controlling certain

functions of automobiles were not as serious as one might otherwise guess because of various mitigating circumstances; such as the bug that allowed the hack having been fixed before the report was published, or that the hack required security researchers having physical access to the car prior to the hack to prepare for it. The U.S. National Intelligence Council in an unclassified report maintains that it would be hard to deny "access to networks of sensors and remotely-controlled objects by enemies of the United States, criminals, and mischief makers... An open market for aggregated sensor data could serve the interests of commerce and security no less than it helps criminals and spies identify vulnerable targets.

Thus, massively parallel sensor fusion may undermine social cohesion, if it proves to be fundamentally incompatible with Fourth-Amendment guarantees against unreasonable search." In general, the intelligence community views the Internet of things as a rich source of data. As a response to increasing concerns over security, the Internet of Things Security Foundation (IoTSF) was launched on 23 September 2015. IoTSF has a mission to secure the Internet of things by promoting knowledge and best practice.

Its founding board is made from technology providers and telecommunications companies including BT, Vodafone, Imagination Technologies and Pen Test Partners. In addition, large IT companies are continuously developing innovative solutions to ensure the security for IoT devices. As per the estimates from KBV Research, the overall IoT security market would grow at 27.9% rate during 2016-2022 as a result of growing infrastructural concerns and diversified usage of Internet of things. In 2016, a distributed denial of service attack powered by Internet of things devices running the Mirai malware took down a DNS provider and major web sites.

Design

Given widespread recognition of the evolving nature of the design and management of the Internet of things, sustainable and secure deployment of IoT solutions must design for "anarchic scalability." Application of the concept of anarchic scalability can be extended to physical systems (i.e. controlled real-world objects), by virtue of those systems being designed to account for uncertain management futures. This "hard anarchic scalability" thus provides a pathway forward to fully realize the

potential of Internet of things solutions by selectively constraining physical systems to allow for all management regimes without risking physical failure.

Brown University computer scientist Michael Littman has argued that successful execution of the Internet of things requires consideration of the interface's usability as well as the technology itself. These interfaces need to be not only more user-friendly but also better integrated: "If users need to learn different interfaces for their vacuums, their locks, their sprinklers, their lights, and their coffeemakers, it's tough to say that their lives have been made any easier."

Environmental sustainability impact:

A concern regarding Internet of things technologies pertains to the environmental impacts of the manufacture, use, and eventual disposal of all these semiconductor-rich devices. Modern electronics are replete with a wide variety of heavy metals and rare-earth metals, as well as highly toxic synthetic chemicals. This makes them extremely difficult to properly recycle. Electronic components are often incinerated or placed in regular landfills. Furthermore, the human and environmental cost of mining the rare-earth metals that are integral to modern electronic components continues to grow. With production of electronic equipment growing globally yet little of the metals (from end-of-life equipment) are being recovered for reuse, the environmental impacts can be expected to increase.

Also, because the concept of Internet of things entails adding electronics to mundane devices (for example, simple light switches), and because the major driver for replacement of electronic components is often technological obsolescence rather than actual failure to function, it is reasonable to expect that items that previously were kept in service for many decades would see an accelerated replacement cycle if they were part of the IoT. For example, a traditional house built with 30 light switches and 30 electrical outlets might stand for 50 years, with all those components still original at the end of that period. But a modern house built with the same number of switches and outlets set up for IoT might see each switch and outlet replaced at five-year intervals, in order to keep up to date with technological changes. This translates into a ten-fold increase in waste requiring disposal.

Intentional obsolescence of devices:

The Electronic Frontier Foundation has raised concerns that companies can use the technologies necessary to support connected devices to intentionally disable or "brick" their customers' devices via a remote software update or by disabling a service necessary to the operation of the device. In one example, home automation devices sold with the promise of a "Lifetime Subscription" were rendered useless after Nest Labs acquired Revolv and made the decision to shut down the central servers the Revolv devices had used to operate. As Nest is a company owned by Alphabet (Google's parent company), the EFF argues this sets a "terrible precedent for a company with ambitions to sell self-driving cars, medical devices, and other high-end gadgets that may be essential to a person's livelihood or physical safety".

Owners should be free to point their devices to a different server or collaborate on improved software. But such action violates the United States DMCA section 1201, which only has an exemption for "local use". This forces tinkerers who want to keep using their own equipment into a legal grey area. EFF thinks buyers should refuse electronics and software that prioritize the manufacturer's wishes above their own. Examples of post-sale manipulations include Google Nest Revolv, disabled privacy settings on Android, Sony disabling Linux on PlayStation 3, and enforced EULA on Wii U.

Confusing terminology:

Kevin Lonergan at Information Age, a business-technology magazine, has referred to the terms surrounding IoT as a "terminology zoo". The lack of clear terminology is not "useful from a practical point of view" and a "source of confusion for the end user". A company operating in the IoT space could be working in anything related to sensor technology, networking, embedded systems, or analytics. According to Lonergan, the term IoT was coined before smart phones, tablets, and devices as we know them today existed, and there is a long list of terms with varying degrees of overlap and technological convergence: Internet of Things (IoT), Internet of Everything (IoE), Industrial Internet, Pervasive Computing, Pervasive Sensing, Ubiquitous Computing, Cyber-Physical Systems (CPS), Wireless Sensor Networks (WSN), Smart Objects, Cooperating Objects, Machine-to-Machine (M2M), Ambient Intelligence (Aml), Operational Technology (OT), and Information Technology (IT).

Regarding IIoT, an industrial sub-field of IoT, the Industrial Internet Consortium's Vocabulary Task Group has created a "common and reusable vocabulary of terms" to ensure "consistent terminology" across publications issued by the Industrial Internet Consortium. IoT One has created an IoT Terms Database including a New Term Alert to be notified when a new term is published. As of March 2017, this database aggregates 711 IoT-related terms, however, without any attempts to reduce terminological ambiguity and complexity.

IoT adoption barriers

Complexity and unclear value propositions:

Despite a shared belief in the potential of IoT, industry leaders and consumers are facing barriers to adopt IoT technology more widely. Dan Yarmoluk from ATEK Access Technologies has written that "the IoT industry appears heavily focused on gadgets and not making them relevant to the particular business verticals" and "can appear expensive and intimidating." Mike Farley has argued in Forbes that many IoT solutions are either too complex or lack a clear use case for end-users. "Instead of convincing consumers that they need complex systems to serve needs they don't have, we should fix real problems people struggle with every day."

Many gadgets in the consumer IoT space have appealed to early adopters, yet failed to demonstrate relevance to ordinary people's lives. In order to overcome barriers, "we need to stop making toys no one cares about and instead work on building simple solutions to real, everyday problems for real people." A recent study by Ericsson regarding the adoption of IoT among Danish companies, has suggested that many are struggling "to pinpoint exactly where the value of IoT lies for them". A company must identify where the value of IoT lies in order to capture it, otherwise non-action is the consequence. This indicates that a major roadblock to IoT adoption is not technical but analytical in nature.

Privacy and security concerns:

According to a recent study by Noura Aleisa and Karen Renaud at the University of Glasgow, "the Internet of Things' potential for major privacy invasion is a concern" with much of research "disproportionally focused on the security concerns of IoT."

Among the "proposed solutions in terms of the techniques they deployed and the extent to which they satisfied core privacy principles", only very few turned out to be fully satisfactory. Louis Basenese, investment director at Wall Street Daily, has criticized the industry's lack of attention to security issues: "Despite high-profile and alarming hacks, device manufacturers remain undeterred, focusing on profitability over security. Consumers need to have ultimate control over collected data, including the option to delete it if they choose...Without privacy assurances, wide-scale consumer adoption simply won't happen".

In a post-Snowden world of global surveillance disclosures, consumers take a more active interest in protecting their privacy and demand IoT devices to be screened for potential security vulnerabilities and privacy violations before purchasing them. According to the 2016 Accenture Digital Consumer Survey, in which 28000 consumers in 28 countries were polled on their use of consumer technology, security "has moved from being a nagging problem to a top barrier as consumers are now choosing to abandon IoT devices and services over security concerns".

The survey revealed that "out of the consumers aware of hacker attacks and owning or planning to own IoT devices in the next five years, 18 percent decided to terminate the use of the services and related services until they get safety guarantees." This suggests that consumers increasingly perceive privacy risks and security concerns to outweigh the value propositions of IoT devices and opt to postpone planned purchases or service subscriptions.

Traditional governance structures:

A study issued by Ericsson regarding the adoption of Internet of things among Danish companies identified a "clash between IoT and companies' traditional governance structures, as IoT still presents both uncertainties and a lack of historical precedence." Among the respondents interviewed, 60 percent stated that they "do not believe they have the organizational capabilities, and three of four do not believe they have the processes needed, to capture the IoT opportunity."

This has led to a need to understand organizational culture in order to facilitate

organizational design processes and to test new innovation management practices. A lack of digital leadership in the age of digital transformation has also stifled innovation and IoT adoption to a degree that many companies, in the face of uncertainty, "were waiting for the market dynamics to play out", or further action in regards to IoT "was pending competitor moves, customer pull, or regulatory requirements." Some of these companies risk being 'kodaked' - "Kodak was a market leader until digital disruption eclipsed film photography with digital photos" - failing to "see the disruptive forces affecting their industry" and "to truly embrace the new business models the disruptive change opens up." Scott Anthony has written in Harvard Business Review that Kodak "created a digital camera, invested in the technology, and even understood that photos would be shared online" but ultimately failed to realize that "online photo sharing was the new business, not just a way to expand the printing business.

CHAPTER 4

4. HARDWARE SETUP DETAILS:

RASPBERRY PI:

The Raspberry Pi is a series of credit card–sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools and developing countries. The original Raspberry Pi and Raspberry Pi 3 are manufactured in several board configurations through licensed manufacturing agreements with Newark element14 (Premier Farnell), RS Components and Ego man. These companies sell the Raspberry Pi's online. Ego man produces a version for distribution solely in Taiwan, which can be distinguished from other Pi's by their red coloring and lack of FCC/CE marks.

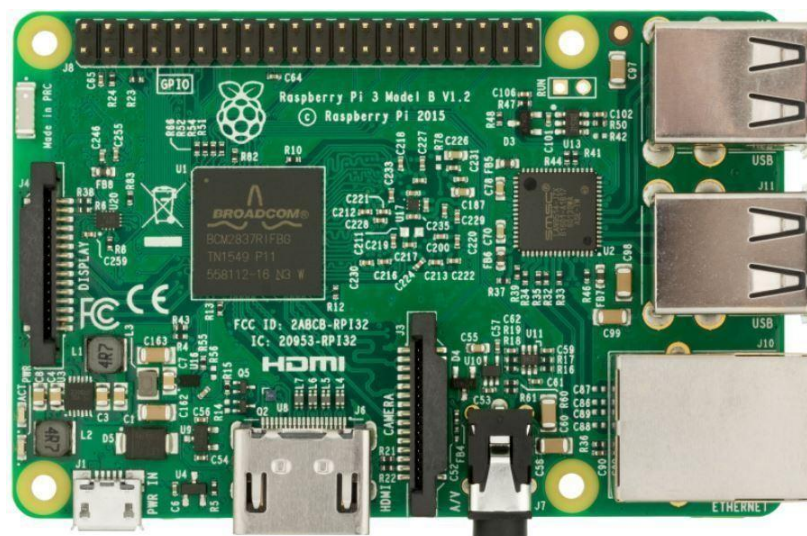
The hardware is the same across all manufacturers. The original Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, Video Core IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded (models B and B+) to 512 MB. The system has Secure Digital (SD) (models A and B) or Micro SD (models A+ and B+) sockets for boot media and persistent storage. In 2014, the Raspberry Pi Foundation launched the Compute Module, which packages a BCM2835 with 512 MB RAM and an eMMC flash chip into a module for use as a part of embedded systems.

The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with support for BBC BASIC (via the RISC OS image or the Brandy Basic clone for Linux), C, C++, Java, Perl and Ruby. As of 8 June 2015, about five to six million Raspberry Pi have been sold. While already the fastest selling British personal computer, it has also shipped the second largest number of units behind the Amstrad PCW, the —Personal

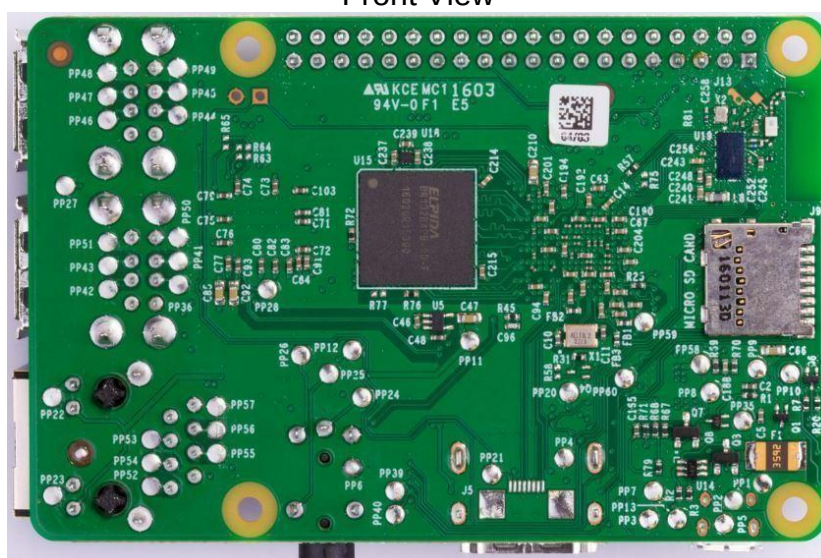
Computer Word-processor", which sold eight million. In early February 2015, the next-generation Raspberry Pi 3, was released.

The new computer board is initially available only in one configuration (model B) and features a Broadcom BCM2836 SoC, with a quad-core ARM Cortex-A7 CPU and a Video Core IV dual-core GPU; 1 GB of RAM with remaining specifications being similar to those of the previous generation model B+. The Raspberry Pi 3 retains the same US\$35 price point of the model B, with the US\$20 model A+ remaining on sale.

Hardware



Front View



Back View

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

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

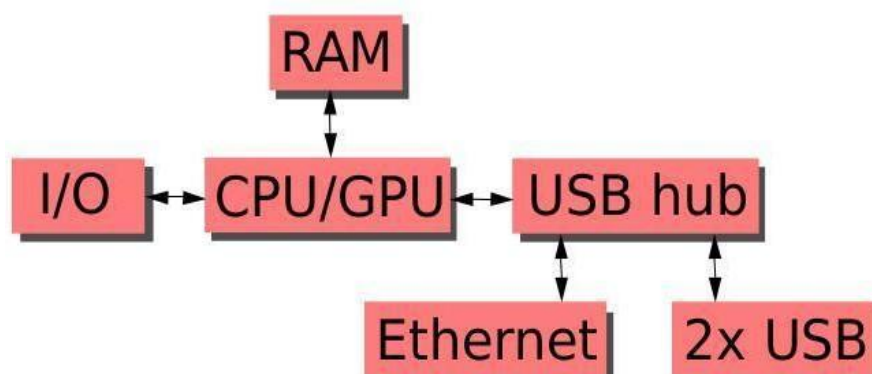
Processor

The SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in older smartphones (such as iPhone / 3G / 3GS). The Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes a 700 MHz ARM1176JZF-S processor, Video Core IV GPU, and RAM. It has a Level 1 cache of 16 KB and a Level 2 cache of 128 KB. The Level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997-1999. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphics capabilities of the Raspberry Pi are roughly equivalent to the level of performance of the Xbox of 2001.

The LINPACK single node compute benchmark results in a mean single precision performance of 0.065 GFLOPS and a mean double precision performance of 0.041 GFLOPS for one Raspberry Pi Model-B board. A cluster of 64 Raspberry Pi Model-B computers, labeled Iridis-pill, achieved a LINPACK HPL suite result of 1.14 GFLOPS (n=10240) at 216 watts for c.

Raspberry Pi 3 is based on Broadcom BCM2836 SoC, which includes a quad-core Cortex-A7 CPU running at 900MHz and 1 GB RAM. It is described as 4–6 times more powerful than its predecessor. The GPU is identical.



Overclocking

The first generation Raspberry Pi chip operated at 700 MHz by default and did not become hot enough to need a heat sink or special cooling, unless the chip was overclocked. The second generation runs at 900 MHz by default, and also does not become hot enough to need a heat sink or special cooling. Again overclocking may heat up the SoC more than usual.

Most Raspberry Pi chips could be overclocked to 800 MHz and some even higher to 1000 MHz. There are reports the second generation can be similarly overclocked, in extreme cases, even to 1500 MHz (discarding all safety features and over voltage limitations). In the Raspbian Linux desktop the overclocking options on boot can be done by a software command running `sudo raspi-config` without voiding the warranty. [25] In those cases the Pi automatically shuts the overclocking down in case the chip reaches 85 °C (185 °F), but it is possible to overrule automatic over voltage and overclocking settings (voiding the warranty). In that case, one can try putting an appropriately sized heat sink on it to keep the chip from heating up far above 85 °C.

Newer versions of the firmware contain the option to choose between five overclock (turbo) presets that when turned on try to get the most performance out of the SoC without impairing the lifetime of the Pi. This is done by monitoring the core temperature of the chip, and the CPU load, and dynamically adjusting clock speeds and the core voltage.

When the demand is low on the CPU, or it is running too hot, the performance is throttled, but if the CPU has much to do, and the chip's temperature is acceptable, performance is temporarily increased, with clock speeds of up to 1 GHz, depending on the individual board, and on which of the turbo settings is used. The five settings are:

- none; 700 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 overvolt,
- modest; 800 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 overvolt,
- medium; 900 MHz ARM, 250 MHz core, 450 MHz SDRAM, 2 overvolt,

- high; 950 MHz ARM, 250 MHz core, 450 MHz SDRAM, 6 overvolt,
- turbo; 1000 MHz ARM, 500 MHz core, 600 MHz SDRAM, 6 overvolt.

In the highest (turbo) preset the SDRAM clock was originally 500 MHz, but this was later changed to 600 MHz because 500 MHz sometimes causes SD card corruption. Simultaneously in high mode the core clock speed was lowered from 450 to 250 MHz, and in medium mode from 333 to 250MHz.

HARDWARE

RAM

On the older beta model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release model B (and model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with just a 1080p frame buffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC). Comparatively the Nokia 701 uses 128 MB for the Broadcom Video Core IV.

For the new model B with 512 MB RAM initially there were new standard memory split files released(arm256_start.elf, arm384_start.elf, arm496_start.elf) for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). But a week or so later the RPF released a new version of start. Elf that could read a new entry in config.txt (gpu_mem=xx) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start. Elf worked the same for 256 and 512 MB Raspberry Pi the Raspberry Pi 3 has 1 GB of RAM.

Networking

Though the model A and A+ do not have an 8P8C (RJ45II) Ethernet port, they can be connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the model B and B+ the Ethernet port is provided by a built-in USB

Ethernet adapter.

Peripherals

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse.

Video

The video controller is capable of standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions and older standard CRT TV resolutions. As shipped (i.e. without custom overclocking) it is capable of the following: 640×350 EGA; 640×480 VGA; 800×600 SVGA; 1024×768 XGA; 1280×720 720p HDTV; 1280×768 WXGA variant; 1280×800 WXGA variant; 1280×1024 SXGA; 1366×768 WXGA variant; 1400×1050 SXGA+; 1600×1200 UXGA; 1680×1050 WXGA+; 1920×1080 1080p HDTV; 1920×1200 WUXGA. It can generate 576i and 480i composite video signals for PAL-BGHID, PAL-M, PAL-N, NTSC and NTSC-J.

Real-time clock

The Raspberry Pi does not come with a real-time clock, which means it cannot keep track of the time of day while it is not powered on. As alternatives, a program running on the Pi can get the time from a network time server or user input at boot time. A real-time clock (such as the DS1307) with battery backup can be added (often via the I2C interface).

GPIO connector

RPi A+, B+ and 2B GPIO J8 40-pin pinout. Models A and B have only the first 26 pins.

Model B rev 2 also has a pad (called P5 on the board and P6 on the schematics) of 8 pins offering access to an additional 4 GPIO connections.

Models A and B provide GPIO access to the ACT status LED using GPIO 16. Models A+ and B+ provide GPIO access to the ACT status LED using GPIO 47, and the power status LED using GPIO 35.

Accessories

Camera – On 14 May 2013, the foundation and the distributors RS Components & Premier Farnell/Element14 launched the Raspberry Pi camera board with a firmware update to accommodate it. [65] The camera board is shipped with a flexible flat cable

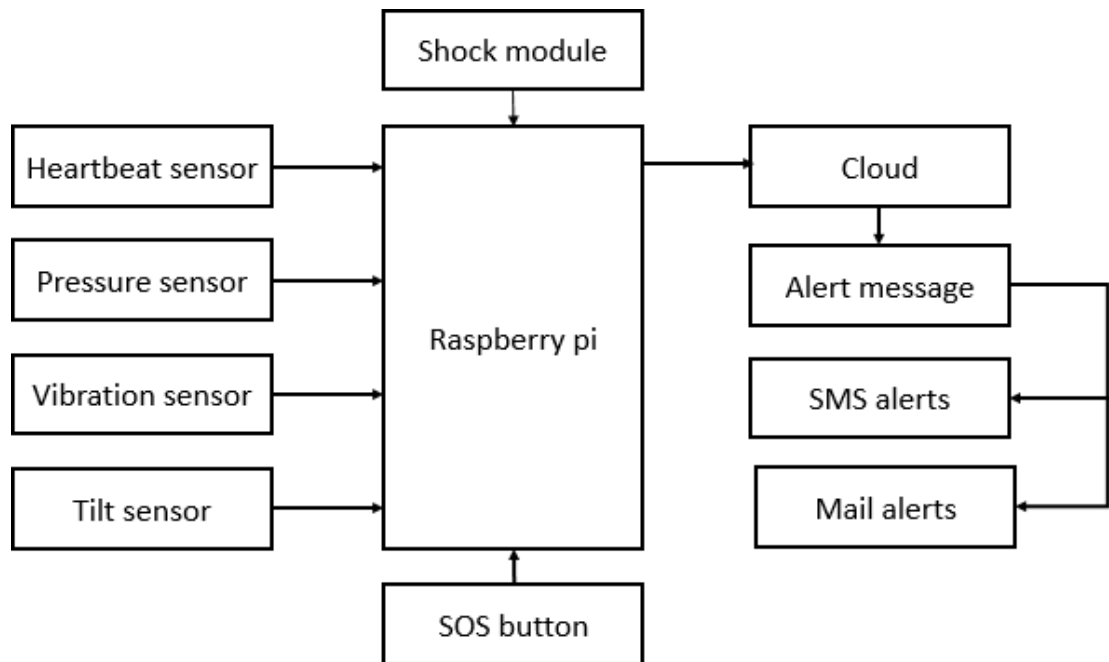
that plugs into the CSI connector located between the Ethernet and HDMI ports. In Raspbian, one enables the system to use the camera board by the installing or upgrading to the latest version of the operating system (OS) and then running Raspi-config and selecting the camera option. The cost of the camera module is €20 in Europe (9 September 2013). It can produce 1080p, 720p and 640x480p video. The footprint dimensions are 25 mm x 20 mm x 9 mm.

- Gertboard – A Raspberry Pi Foundation sanctioned device, designed for educational purposes, that expands the Raspberry Pi's GPIO pins to allow interface with and control of LEDs, switches, analog signals, sensors and other devices. It also includes an optional Arduino compatible controller to interface with the Pi.

- Infrared Camera – In October 2013, the foundation announced that they would begin producing a camera module without an infrared filter, called the Pi NoIR

HAT (Hardware Attached on Top) expansion boards – Together with the model B+, inspired by the Arduino shield boards, the interface for HAT boards was devised by the Raspberry Pi Foundation. Each HAT board carries a small EEPROM (typically a CAT24C32WI-GT3) containing the relevant details of the board so that the Raspberry Pi's OS is informed of the HAT, and the technical details of it, relevant to the OS using the HAT. Mechanical details of a HAT board, that use the four mounting holes in their rectangular formation.

BLOCK DIAGRAM:



HARDWARE USED:

- Raspberry Pi
- heartbeat sensor
- pressure sensor
- tilt sensor
- vibration sensor
- GPS
- SOS button

SOFTWARE USED:

- Arduino c

Software

Operating systems

The Raspberry Pi primarily uses Linux-kernel-based operating systems. The ARM11 chip at the heart of the Pi (first generation models) is based on version 6 of the ARM. The current releases of several popular versions of Linux, including Ubuntu, will not run on the ARM11. It is not possible to run Windows on the original Raspberry Pi, though the new Raspberry Pi 3 can also run on Windows 10 IoT Core operating system. The Raspberry Pi 3 currently only supports Ubuntu Snappy Core, Raspbian, OpenELEC and RISC OS.

The install manager for the Raspberry Pi is NOOBS. The operating systems included with NOOBS are:

- Arch Linux ARM
- OpenELEC
- Pidora (Fedora Remix)
- Puppy Linux
- Raspbmc and the XBMC open source digital media center
- RISC OS – is the operating system of the first ARM-based computer.

SOFTWARE

Raspbian (recommended for Raspberry Pi 1) – Maintained independently of the Foundation; based on the Debian ARM hard-float (armhf) architecture port originally designed for ARMv7 and later processors (with Jazelle RCT/ThumbEE and VFPv3), compiled for the more limited ARMv6 instruction set of the Raspberry Pi

- A minimum size of 4 GB SD card is required for the Raspbian images provided by the Raspberry Pi Foundation. There is a Pi Store for exchanging programs
- The Raspbian Server Edition is a stripped version with fewer software packages bundled as compared to the usual desktop computer oriented Raspbian.

- The Wayland display server protocol enable the efficient use of the GPU for hardware accelerated
- GUIdrawing functions. On 16 April 2014, a GUI shell for Weston called Maynard was released.
- PiBang Linux is derived from Raspbian.
- Raspbian for Robots – is a fork of Raspbian for robotics projects with LEGO, Grove, and Arduino.

Other operating systems

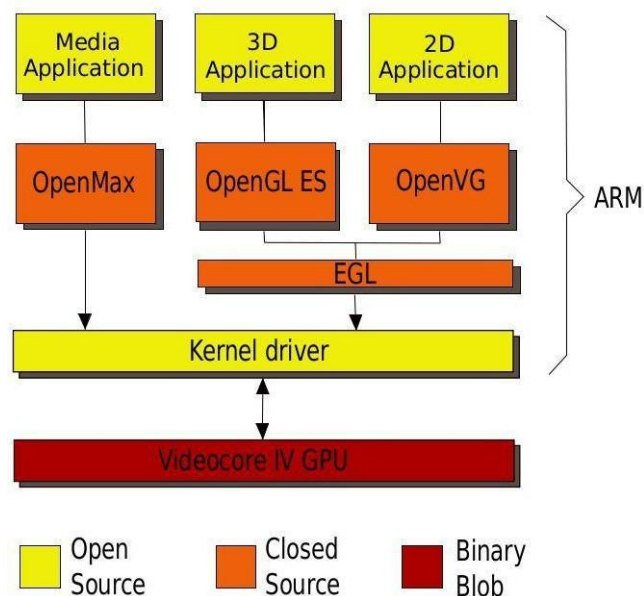
- Q4os
- Xbian – Using the Kodi (formerly XBMC) open source digital media center
- Open SUSE
- Raspberry Pi Fedora Remix
- Slack ware ARM – Version 13.37 and later runs on the Raspberry Pi without modification. The 128–496 MB of available memory on the Raspberry Pi is at least twice the minimum requirement of 64 MB needed to run Slack ware Linux on an ARM or i386 system. (Whereas the majority of Linux systems boot into a graphical user interface, Slack ware's default user environment is the textual shell /command line interface. The Flux box window manager running under the X Window System requires an additional 48MB of RAM.
- FreeBSD and NetBSD are general operating systems.
- Plan 9 from Bell Labs and Inferno (in beta)
- Moebius – A light ARM HF distribution based on Debian. It uses Raspbian repository, but it fits in a 128MB SD card. It has just minimal services and its memory usage is optimized to keep a small footprint.
- OpenWrt – is primarily used on embedded devices to route network traffic.
- Kali Linux – is a Debian-derived destro designed for digital forensics and penetration testing.
- Pardus ARM – is a Debian-Based operating system which is the light version of the Pardus (operating system).
- Instant Web Kiosk – is an operating system for digital signage purposes (web and media views).
- Ark OS – is designed for website and email self-hosting.
- Minepion – is a dedicated operating system for mining crypto currency.

- Kano OS
- Nard SDK for industrial embedded systems
- Sailfish OS with Raspberry Pi 3 (due to use ARM Cortex-A7 CPU; Raspberry Pi 1 uses different ARMv6 architecture and Sailfish requires ARMv7)
- Tiny Core Linux – a minimal Linux operating system focused on providing a base system using Busy Box and FLTK. Designed to run primarily in RAM.
- Windows 10 IoT Core – Microsoft offers a free edition of Windows 10, known as Windows 10 IoT Core that runs natively on the Raspberry Pi 3.
- WTware of Raspberry Pi 3 – is a free operating system for creating Windows thin client from Pi 3.

Driver APIs

- IPFire – a dedicated firewall/router distribution for the protection of a SOHO LAN; runs only on a RaspberryPi 1; porting to the Raspberry Pi 3 is not planned for now.
- xv6 – a modern reimplementation of Sixth Edition Unix OS for teaching purposes; it is ported to RaspberryPi from MIT xv6; this xv6 port can boot from NOOBS. Planned operating systems
- Haiku – This open source BeOS clone has been targeted for the Raspberry Pi and several other ARM boards. Work began in 2011, on the model 1 but only the model 2 will be supported.

Driver APIs



Raspberry Pi can use a VideoCore IV GPU via a binary blob, which is loaded into the GPU at boot time from the SD-card, and additional software, that initially was

closed source. This part of the driver code was later released, however much of the actual driver work is done using the closed source GPU code. Application software use calls to closed source run-time libraries (OpenMax, OpenGL ES or OpenVG) which in turn calls an open source driver inside the Linux kernel, which then calls the closed source VideoCore IV GPU driver code. The API of the kernel driver is specific for these closed libraries. Video applications use OpenMAX, 3D applications use OpenGL ES and 2D applications use OpenVG which both in turn use EGL. OpenMAX and EGL use the open source kernel driver in turn.

RECEPTION AND USE

Third party application software

- AstroPrint – Since August 2014, AstroPrint's Wireless 3D Printing software can be run on the Pi 3
- Mathematica – Since 21 November 2013, Raspbian includes a full installation of this proprietary software for free. As of 24 August 2015, the version is Mathematica 10.2.
- Mine craft – Released 11 February 2013, a version for the Raspberry Pi, in which you can modify the game world with code.
- User Gate Web Filter – On 20 September 2013, Florida-based security vendor Entensys announced porting UserGate Web Filter to Raspberry Pi platform.
- Julia – Since May 2015, Julia programming language compiles and runs on the Pi 3, and the original, in an alpha state.

Tracking Raspberry Pi online on a global map

Ryan Walmsley, a UK school student created a site in 2012, to register and track any Raspberry Pi across the globe. It became very popular soon after its launch. The current site is powered by Google Maps and Digital Ocean and is free. It has a limitation of registering only one Raspberry Pi per unique email id. It uses IP based basic location tracking and is fairly accurate up to Locale or City level.

Reception and use

Technology writer Glyn Moody described the project in May 2011 as a potential BBC Micro 2.0, not by replacing PC compatible machines but by supplementing them. In March 2012

Stephen Pritchard echoed the BBC Micro successor sentiment in ITPRO. Alex Hope, co-author of the Next Gen report, is hopeful that the computer will engage children with the excitement of programming. Co-author Ian Livingstone suggested that the BBC could be involved in building support for the device, possibly branding it as the BBC Nano. Chris Williams, writing in The Register sees the inclusion of programming languages such as Kids Ruby, Scratch and BASIC as a good start to equip kids with the skills needed in the future – although it remains to be seen how effective their use will be. The Centre for Computing History strongly supports the Raspberry Pi project, feeling that it could usher in a new era. Before release, the board was showcased by ARM's CEO Warren East at an event in Cambridge outlining Google's ideas to improve UK science and technology education.

Harry Fairhead, however, suggests that more emphasis should be put on improving the educational software available on existing hardware, using tools such as Google App Inventor to return programming to schools, rather than adding new hardware choices. Simon Rockman, writing in a ZDNet blog, was of the opinion that teens will have —better things to do, despite what happened in the 1980s. In October 2012, the Raspberry Pi won T3's Innovation of the Year award, and futurist Mark Pesce cited a (borrowed) Raspberry Pi as the inspiration for his ambient device project MooresCloud.

In October 2012, the British Computer Society reacted to the announcement of enhanced specifications by stating, —it's definitely something we'll want to sink our teeth into.

In February 2015, a switched-mode power supply chip, designated U16, of the Raspberry Pi 3 model B version 1.1 (the initially released version) was found to be vulnerable to flashes of light, particularly the light from xenon camera flashes and green and red laser pointers. However, other bright lights, particularly ones that are on continuously, were found to have no effect.

The symptom was the Raspberry Pi 3 spontaneously rebooting or turning off when these lights were flashed at the chip. Initially, some users and commenters suspected that the electromagnetic pulse from the xenon flash tube was causing the

problem by interfering with the computer's digital circuitry, but this was ruled out by tests where the light was either blocked by a card or aimed at the other side of the Raspberry Pi 3, both of which did not cause a problem. The problem was narrowed down to the U16 chip by covering first the system on a chip (main processor) and then U16 with opaque poster mounting compound. Light being the sole culprit, instead of EMP, was further confirmed by the laser pointer tests, where it was also found that less opaque covering was needed to shield against the laser pointers than to shield against the xenon flashes

The U16 chip seems to be bare silicon without a plastic cover (i.e. a chip-scale package or wafer-level package), which would, if present, block the light. Based on the facts that the chip, like all semiconductors, is light-sensitive (photovoltaic effect), that silicon is transparent to infrared light, and that xenon flashes emit more infrared light than laser pointers (therefore requiring more light shielding), it is currently thought that this combination of factors allows the sudden bright infrared light to cause an instability in the output voltage of the power supply, triggering shutdown or restart of the Raspberry Pi 3.

Unofficial workarounds include covering U16 with opaque material (such as electrical tape, lacquer, poster mounting compound, or even balled-up bread), putting the Raspberry Pi 3 in a case, and avoiding taking photos of the top side of the board with a xenon flash. This issue was not caught before the release of the Raspberry Pi 3 because while commercial electronic devices are routinely subjected to tests of susceptibility to radio interference, it is not standard or common practice to test their susceptibility to optical interference.

Use in education

As of January 2012, enquiries about the board in the United Kingdom have been received from schools in both the state and private sectors, with around five times as much interest from the latter. It is hoped that businesses will sponsor purchases for less advantaged schools. The CEO of Premier Farnell said that the government of a country in the Middle East has expressed interest in providing a board to every schoolgirl, in order to enhance her employment prospects.

In 2014, the Raspberry Pi Foundation hired a number of its community members

including ex-teachers and software developers to launch a set of free learning resources for its website. The resources are freely licensed under Creative Commons, and contributions and collaborations are encouraged on social coding platform GitHub.

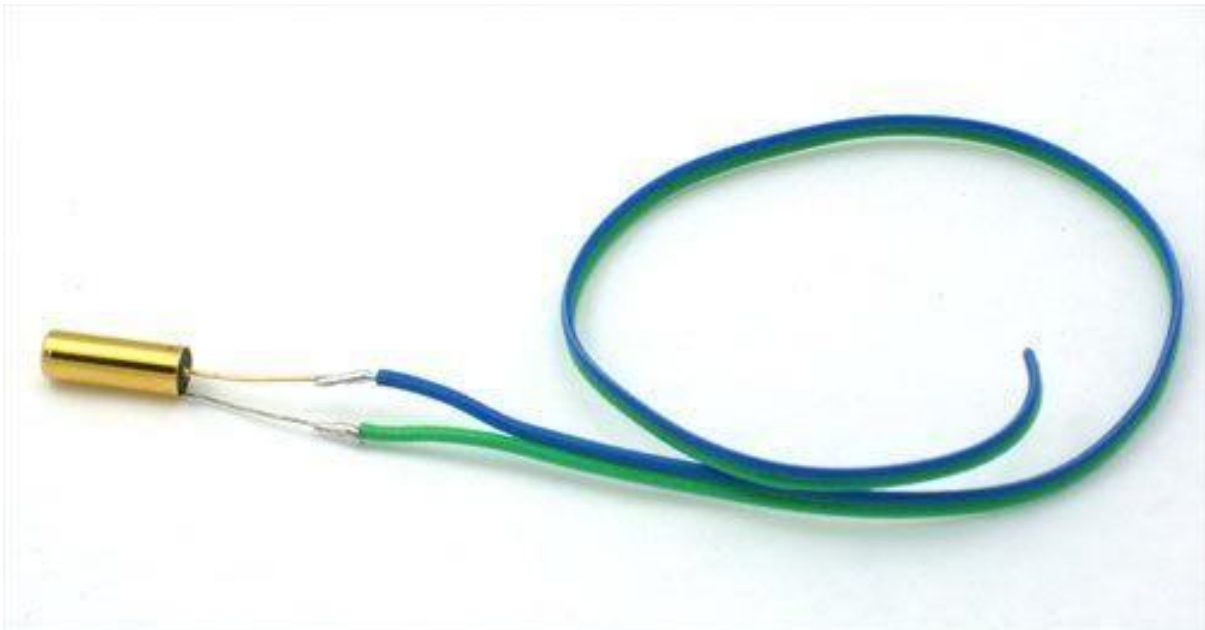
The Foundation also started a teacher training course called Picademy with the aim of helping teachers prepare for teaching the new computing curriculum using the Raspberry Pi in the classroom. The continued professional development course is provided free for teachers and is run by the Foundation's education team.

Tilt Sensors

Tilt sensors allow you to detect orientation or inclination. They are small, inexpensive, low-power and easy-to-use. If used properly, they will not wear out. Their simplicity makes them popular for toys, gadgets and appliances. Sometimes they are referred to as "mercury switches", "tilt switches" or "rolling ball sensors" for obvious reasons.

They are usually made by a cavity of some sort (cylindrical is popular, although not always) and a conductive free mass inside, such as a blob of mercury or rolling ball. One end of the cavity has two conductive elements (poles). When the sensor is oriented so that that end is downwards, the mass rolls onto the poles and shorts them, acting as a switch throw. Tilt switches used to be made exclusively of mercury, but are rarer now since they are recognized as being extremely toxic. The benefits of mercury is that the blob is dense enough that it doesn't bounce and so the switch isn't susceptible to vibrations. On the other hand, ball-type sensors are easy to make, wont shatter, and pose no risk of pollution.

While not as precise or flexible as a full accelerometer, tilt switches can detect motion or orientation simply. Another benefit to them is that the big ones can switch power on their own. Accelerometers, on the other hand, output digital or analog voltage that must then be analyzed with extra circuitry.



Introduction:

A tilt sensor is an instrument that is used for measuring the tilt in multiple axes of a reference plane. Tilt sensors measure the tilting position with reference to gravity, and are used in numerous applications. They enable the easy detection of orientation or inclination. Similar to mercury switches, they may also be known as tilt switches or rolling ball sensors.

These instruments have become more popular over the years, and are being adapted for increasing numbers of high end applications. For example, the sensor provides valuable information about both the vertical and horizontal inclination of an airplane, which helps the pilot to understand how to tackle obstacles during the flight. By knowing the current orientation of the plane, and the angle at which the plane is inclined to the earth's surface, stunt pilots, i.e. the Red Arrows, can put on a fascinating air show. Tilt sensors are an essential decision making tool for the pilots.

Specifications of Tilt Sensors

The functionality of tilt sensors is influenced by factors such as gravity, vibration, temperature, zero offset, linearity, cross-axis sensitivity, acceleration/deceleration, and shock, clear line of sight between the user and the measured point, and calibration of tilt sensors.

The key specifications of tilt sensors include:

- **Number of axes:** The number of axes is an important factor as it varies from application to application. In robotics, a dual axes tilt sensor is used, in video game controllers and joysticks, a 3 axes tilt sensor is required, and in some smartphones, 4 axes sensors are used
- **Resolution:** The minimum inclination detected by the sensor is called its resolution
- **Sensitivity:** The sensor's ability to react to small changes is called its sensitivity
- **Measuring range:** The range of inclination that can be handled by the sensor is important. Some sensors are capable of measurements up to 10° while others can cover a range of up to 60°
- **Noise tolerance:** Noise tends to cause harmonic distortions in the function of the sensor, resulting in output variation and reduction of system efficiency. Manufacturer's guidelines regarding noise levels should be adhered to.
- **Vibration:** Vibrations can disrupt the sensor's functionality, and hence vibration resistance measures are required, especially when the sensors are used in harsh conditions like off-road vehicles or construction sites.

Benefits of Tilt Sensors

The key benefits of tilt sensors include:

- Compact and simple to use and adapt
- High resolution and accuracy
- Very cost effective
- Low power consumption
- Can be read by industry standard data loggers

Working Principle

A tilt sensor has a metallic ball that is designed to move the two pins of the instrument from the 'on' to the 'off' position, and vice versa, if the sensor reaches a pre-determined angle. Tilt sensors are the environment-friendly version of a mercury-switch.

Applications

The following are key applications of tilt sensors:

- To monitor the angle at which a mobile phone or tablet is held for the auto-rotate function
- To detect the position of hand-held game systems and in game controllers
- To indicate roll of boats, vehicles and aircraft
- To measure the angle at which a satellite antenna 'looks' toward a satellite
- To estimate the height of a tree or building using
- To measure the steepness of a ski slope
- As a warning system for the tilt angle of the surface cryogenic liquids during transportation
- To monitor laser levels and seismic activity

HEARTBEAT SENSOR

A person's heartbeat is the sound of the valves in his/her's heart contracting or expanding as they force blood from one region to another. The number of times the heart beats per minute (BPM), is the heart beat rate and the beat of the heart that can be felt in any artery that lies close to the skin is the pulse.

Two Ways to Measure a Heartbeat

- **Manual Way:** Heart beat can be checked manually by checking one's pulses at two locations- wrist (the **radial pulse**) and the neck (**carotid pulse**). The procedure is to place the two fingers (index and middle finger) on the wrist (or neck below the windpipe) and count the number of pulses for 30 seconds and then multiplying that number by 2 to get the heart beat rate. However pressure should be applied minimum and also fingers should be moved up and down till the pulse is felt.
- **Using a sensor:** Heart Beat can be measured based on optical power variation as light is scattered or absorbed during its path through the blood as the heart beat changes.

Principle of Heartbeat Sensor

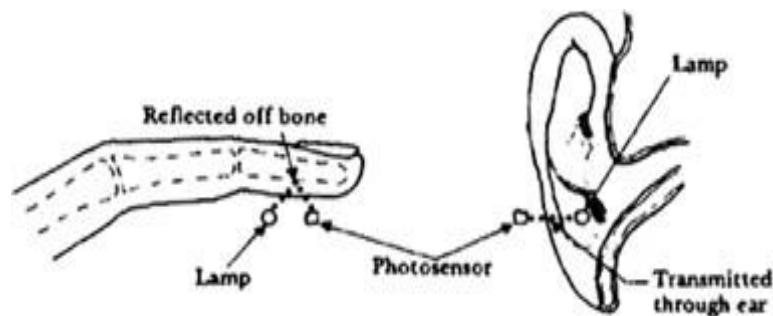
The heartbeat sensor is based on the principle of photo plethysmography. It measures the change in volume of blood through any organ of the body which causes a change in the light intensity through that organ (a vascular region). In case of applications where heart pulse rate is to be monitored, the timing of the pulses is more important.

The flow of blood volume is decided by the rate of heart pulses and since light is absorbed by blood, the signal pulses are equivalent to the heart beat pulses.

There are two types of photoplethysmography:

Transmission: Light emitted from the light emitting device is transmitted through any vascular region of the body like earlobe and received by the detector.

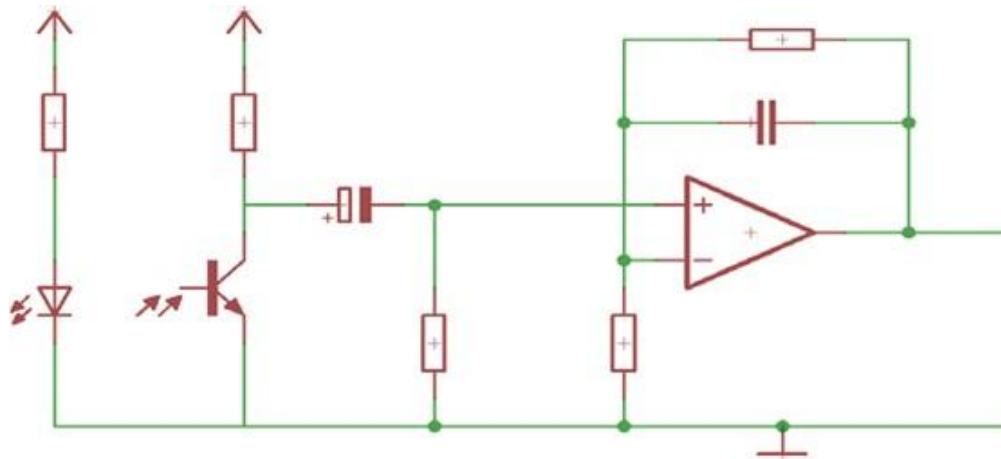
Reflection: Light emitted from the light emitting device is reflected by the regions.



Working of a Heartbeat Sensor

The basic heartbeat sensor consists of a light emitting diode and a detector like a light detecting resistor or a photodiode. The heart beat pulses causes a variation in the flow of blood to different regions of the body. When a tissue is illuminated with the light source, i.e. light emitted by the led, it either reflects (a finger tissue) or transmits the light (earlobe). Some of the light is absorbed by the blood and the transmitted or the reflected light is received by the light detector. The amount of light absorbed depends on the blood volume in that tissue. The detector output is in form of electrical signal and is proportional to the heart beat rate.

This signal is actually a DC signal relating to the tissues and the blood volume and the AC component synchronous with the heart beat and caused by pulsatile changes in arterial blood volume is superimposed on the DC signal. Thus the major requirement is to isolate that AC component as it is of prime importance.



To achieve the task of getting the AC signal, the output from the detector is first filtered using a 2 stage HP-LP circuit and is then converted to digital pulses using a comparator circuit or using simple ADC. The digital pulses are given to a microcontroller for calculating the heart beat rate, given by the formula-

$$\text{BPM (Beats per minute)} = 60 * f$$

Where f is the pulse frequency

Practical Heartbeat Sensor

Practical heartbeat Sensor examples are **Heart Rate Sensor (Product No PC-3147)**. It consists of an infrared led and an ldr embedded onto a clip like structure. The clip is attached to the organ (earlobe or the finger) with the detector part on the flesh.



Another example is **TCRT1000**, having 4 pins-

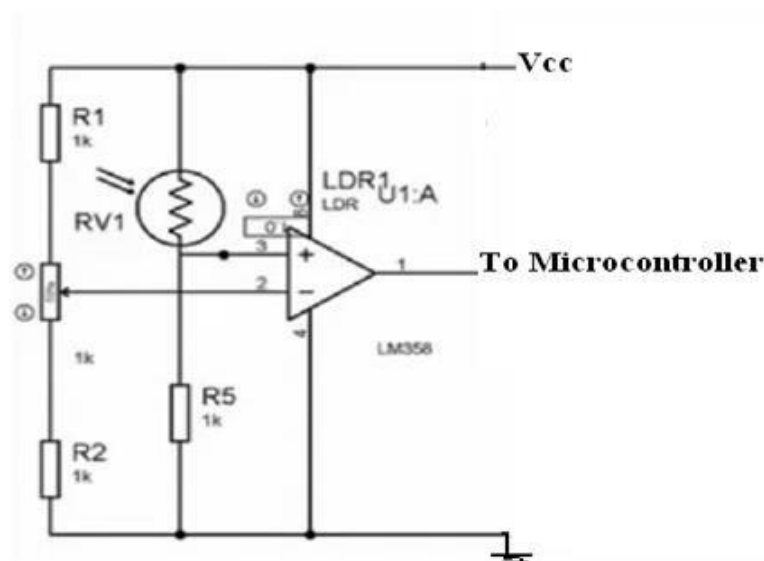
Pin1: To give supply voltage to the LED

Pin2 and 3 are grounded. Pin 4 is the output. Pin 1 is also the enable pin and pulling it high turns the LED on and the sensor starts working. It is embedded on a wearable device which can be worn on the wrist and the output can be sent wirelessly (through Bluetooth) to the computer for processing.



Application developing your own Heartbeat Sensor System

A basic Heartbeat Sensor system can also be built using basic components like an ldr, comparator IC LM358 and a Microcontroller as given below



As described above regarding the principle of heart beat sensor, when the finger tissue or the earlobe tissue is illuminated using a light source, the light is transmitted after getting modulated i.e. a part getting absorbed by the blood and the rest being transmitted. This modulated light is received by the light detector.

Here a Light Dependent Resistor (LDR) is used as a light detector. It works on the principle that when light falls on the resistor, its resistance changes. As the light intensity increases, the resistance decreases. Thus the voltage drop across the resistor decreases.

Here a comparator is used which compares the output voltage from the LDR to that of the threshold voltage. The threshold voltage is the voltage drop across the LDR when the light with fixed intensity, from the light source falls directly on it. The inverting terminal of the comparator LM358 is connected to the potential divider arrangement which is set to the threshold voltage and the non-inverting terminal is connected to the LDR. When a human tissue is illuminated using the light source, the intensity of the light reduces. As this reduced light intensity falls on the LDR, the resistance increases and as a result the voltage drop increases. When the voltage drop across the LDR or the non-inverting input exceeds that of the inverting input, a logic high signal is developed at the output of the comparator and in case voltage drop being lesser a logic low output is developed. Thus the output is a series of pulses. These pulses can be fed to the Microcontroller which accordingly processes the information to get the heart beat rate and this is displayed on the Display interfaced to the Microcontroller.

VIBRATION SENSORS

The piezoelectric sensor is used for flex, touch, vibration and shock measurement. Its basic principal, at the risk of oversimplification, is as follows: whenever a structure moves, it experiences acceleration. A piezoelectric shock sensor, in turn, can generate a charge when physically accelerated. This combination of properties is then used to modify response or reduce noise and vibration.

Why is that important? Because vibration and shock can shorten the life of any electronic and electromechanical system. Delicate leads and bond wires can be stressed, especially after exposure to long term vibration. Solder joints can break free and PCB traces can ever so slightly tear from impact and impulse shock, creating the hardest type of system failure to debug; an intermittent failure.

This article discusses piezoelectric shock and vibration sensors and sensor technology, focusing on available products (all parts mentioned here can be found on

the Digi-Key website — links are provided), as well as design issues and design techniques.

HOW IT WORKS:

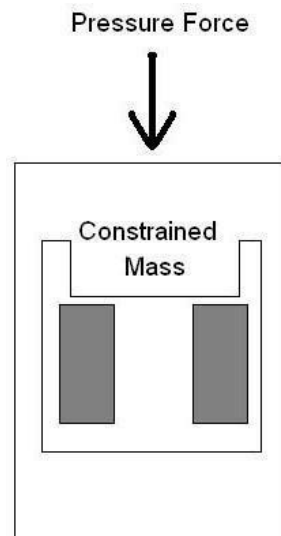
The piezoelectric effect was discovered by Pierre and Jacques Curie in the latter part of the 19th century. They discovered that minerals such as tourmaline and quartz could transform mechanical energy into an electrical output. The voltage induced from pressure (Greek: piezo) is proportional to that applied pressure, and piezoelectric devices can be used to detect single-pressure events as well as repetitive events.

Still, the ability of certain crystals to exhibit electrical charges under mechanical loading was of no practical use until very-high-input impedance amplifiers enabled engineers to amplify the signals produced by these crystals.

Several materials can be used to make piezoelectric sensors, including tourmaline, gallium phosphate, salts, and quartz. Most electronic applications use quartz since its growth technology is far along, thanks to development of the reverse application of the piezoelectric effect; the quartz oscillator.

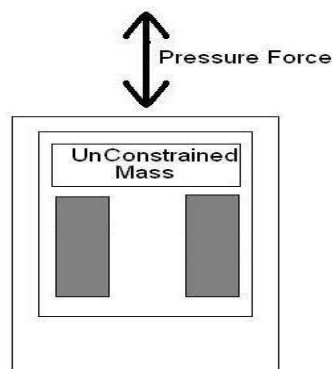
Sensors based on the piezoelectric effect can operate from transverse, longitudinal, or shear forces, and are insensitive to electric fields and electromagnetic radiation. The response is also very linear over wide temperature ranges, making it an ideal sensor for rugged environments. For example, gallium phosphate and tourmaline sensors can have a working temperature range of 1,000°C.

The physical design of the piezoelectric sensor depends on the type of sensor you wish to create. For example, the configuration of a pressure sensor, or a shock (impulse) sensor, would arrange a smaller, but well-known mass of the crystal in a transverse configuration, with the loading deformation along the longest tracks to a more massive base (Figure 1). This assures that the applied pressure will load the base from only one direction.



A constrained mass is allowed to deform the crystal sensor in one axis. This configuration is good for force and pressure.

An accelerometer based on the piezoelectric effect, would use a known mass to deform the sensing crystal part in either a positive or negative direction depending on the excitation force. It should be noted that you need a known modulus of elasticity in the sensor substrate.



Because the modulus of elasticity is known for a substrate material, the unconstrained mass is allowed to move with vibration making this type of piezoelectric sensor ideal for detecting shock and vibration.

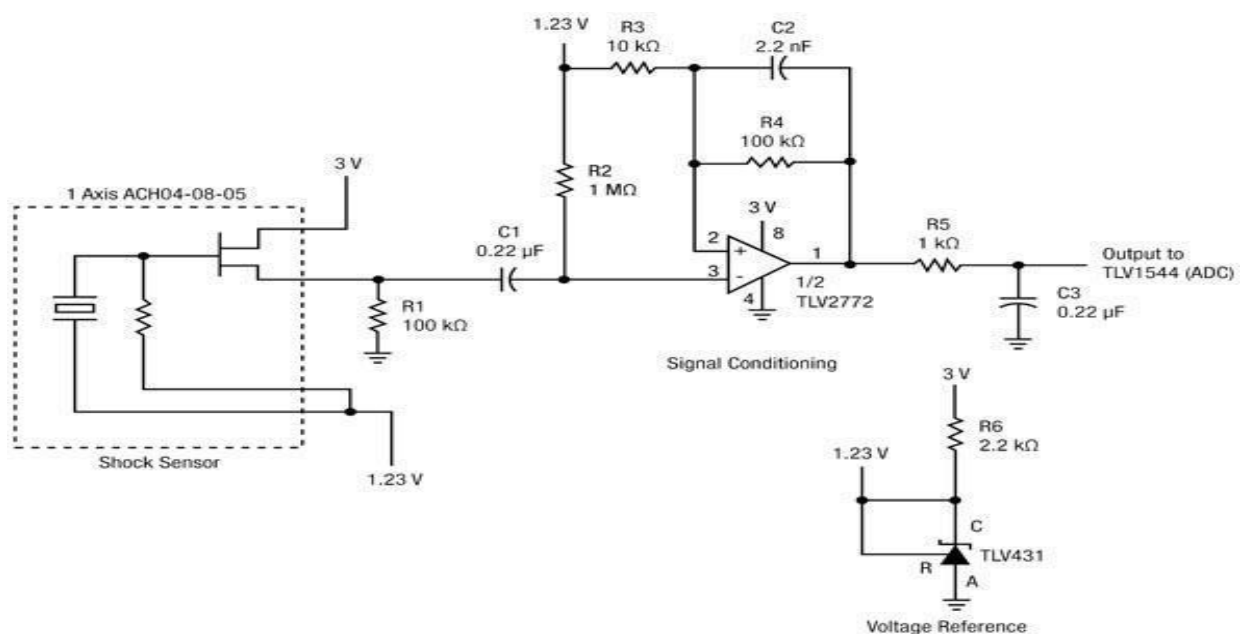
Designing with piezoelectric sensors

piezoelectric sensors require some precautions when connecting to sensitive electronic components. First and foremost, the voltage levels created by hard shock

can be very high, even around 100-V spikes.

More than likely, an op amp will be used to interface these sensors to an A/D converter, either discrete or on a microcontroller. One tip is to choose a high-input-impedance op amp to minimize current. One possible candidate is the Linear Technology JFET input dual op amp. It has $10^{12} \Omega$ input resistance and a 1 MHz gain bandwidth product, good enough to easily handle the vibration ranges of piezoelectric sensors.

Another suitable part is the TLV2771 from Texas Instruments. This rail-to-rail low-power op-amp also has a $10^{12} \Omega$ differential input resistance and a 5 MHz unity-gain bandwidth. Signal conditioning in a single stage can prepare the input from the shock sensor directly into an A/D converter.

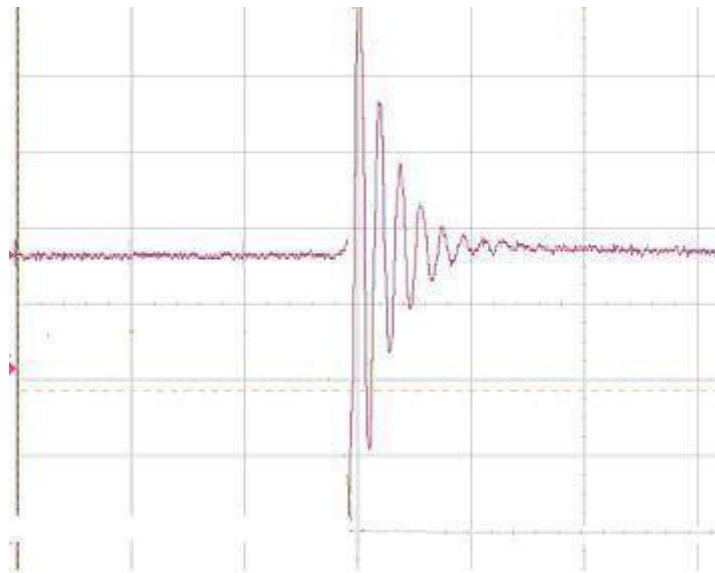


Op amps such as the TI TLV2772 feature high input impedances to help minimize current from the potentially high-voltage inputs from the piezoelectric sensors.

Op-amp circuits can be designed to operate in voltage mode or charge mode. Charge mode is used when the amplifier is remote to the sensor. Voltage mode is used when the amplifier is very close to the sensor.

Another tip is to attenuate the input signal and use the op amp's gain to bring into the desired range. Be aware that you may need snubbing protection on the inputs of the op amp, especially if the design could be subjected to harsh hits.

Also note that you may think that a pressure sensor would generate only a positive voltage, but, in reality, the signal from the sensor can ring and introduce negative voltage spikes (Figure 4). This means that you may need to squelch negative voltage levels on the op-amp inputs, especially if using only a single rail power supply on the op amp.



Care must be taken when using single rail op amps since shock can cause negative voltage spike ringing that can damage op-amp inputs if not squelched.

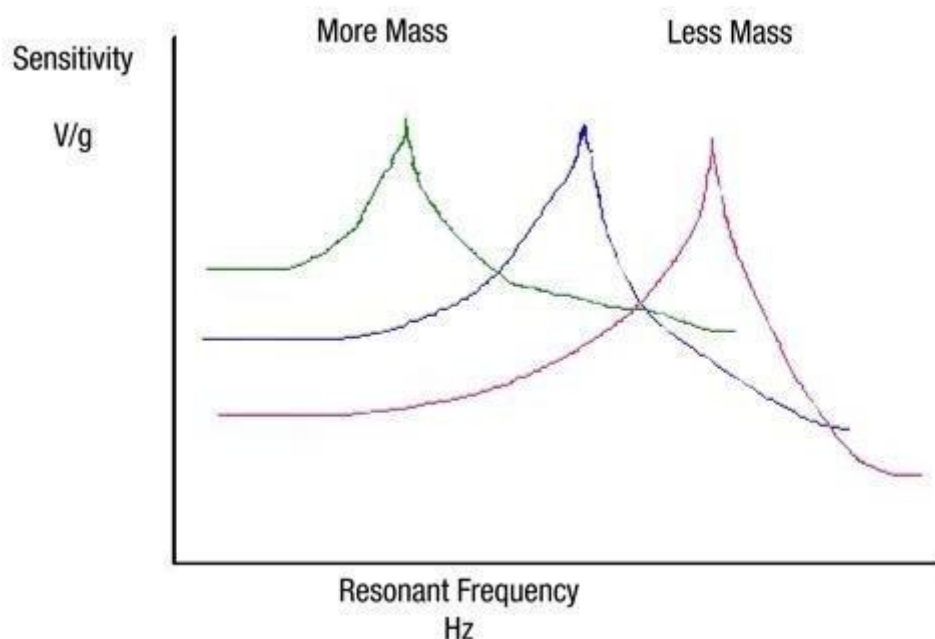
Parts for the design

Many off the shelf piezoelectric sensors are readily available to use in your designs. A case in point is the Parallax 605-00004, which is a piezo vibra tab sensor capable of acting as a switch, or as a vibration sensor (Figure 5). A polymer film laminate uses crimped contacts and features a sensitivity of 50 mV/g.



The flexible through-hole LDTO polymer film piezoelectric sensors can be hard mounted or free floating to detect strain, shock, or vibration.

You should be aware that adding mass to a piezoelectric sensor can change its resonant frequency as well as change its baseline sensitivity. Many piezoelectric sensors like the 605-00004 are characterized to be used this way and provide supporting tables and graphs.



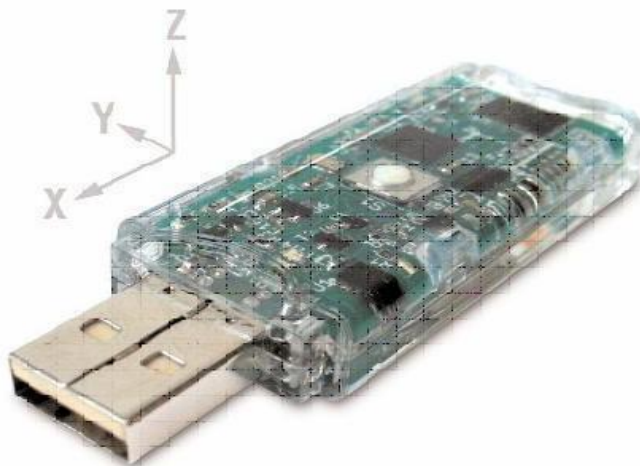
Loading a sensor with mass changes its resonant frequency and sensitivity in a predictable and repeatable way.

Another part worth considering is the Measurement Specialties 0-1002794-0 cantilever piezo film sensor. This is also a vibra tab sensor capable of hard mounting to a surface, floating in an axis of inertia, or mass loaded to prebias and calibrate. The output voltage swings can directly trip a FET or CMOS input, and a multiaxis response can be obtained by offsetting the mass center.

Other uses:

In addition to sensing vibration and shock, a piezoelectric device can also be used to extract ambient energy. Take for example the MideV22BL, which is a hermetically sealed piezoelectric sensor capable of sensing from 26 to 100 Hz vibrations.

It can be evaluated using the company's VR001 data logger, which is a portable rechargeable data logger that can measure acceleration and vibration in three axes (Figure 7). As a USB peripheral, it can be configured and have its data transferred to a PC or other USB host device. It can also be used to help you develop your own designs based on the piezoelectric sensors you choose. A Made training module is available on the Digit-Key website to help bring you up to speed quickly when starting a new design.



The USB plug in three-axis sensor can be used as a development tool and evaluation tool when learning or starting a new design.

Finally, before you select a piezoelectric sensor for shock or vibration duty, evaluate the criteria specific to your intended application (we've provided some examples here), then, spend some time with the manufacturer data sheets and use the links above to get further information from the product and technology pages on the Digit-Key website.

Pressure sensor

A pressure sensor measures pressure, typically of gases or liquids. Pressure is an expression of the force required to stop a fluid from expanding, and is usually stated in terms of force per unit area. A pressure sensor usually acts as a transducer; it generates a signal as a function of the pressure imposed. For the purposes of this article, such a signal is electrical.

Pressure sensors are used for control and monitoring in thousands of everyday applications. Pressure sensors can also be used to indirectly measure other variables such as fluid/gas flow, speed, water level, and altitude. Pressure sensors can alternatively be called pressure transducers, pressure transmitters, pressure senders, pressure indicators, piezometers and manometers, among other names.

Pressure sensors can vary drastically in technology, design, performance, application suitability and cost. A conservative estimate would be that there may be over 50 technologies and at least 300 companies making pressure sensors worldwide.

There is also a category of pressure sensors that are designed to measure in a dynamic mode for capturing very high speed changes in pressure. Example applications for this type of sensor would be in the measuring of combustion pressure in an engine cylinder or in a gas turbine. These sensors are commonly manufactured out of piezoelectric materials such as quartz.

Some pressure sensors, such as those found in some traffic enforcement cameras, function in a binary (off/on) manner, i.e., when pressure is applied to a pressure sensor, the sensor acts to complete or break an electrical circuit. These types of sensors are also known as a pressure switch.



4.4 SOFTWARE SETUP DETAILS:

PYTHON:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code,[25][26] notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. [27]

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. [28]

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software[29] and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Python was conceived in the late 1980s,[30] and its implementation began in December 1989[31] by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL)[32] capable of exception handling and interfacing with the Amoeba operating system.[6] Van Rossum remains Python's principal author. His continuing central role in Python's development is reflected in the title given to him by the Python community: Benevolent

Dictator For Life (BDFL).

On the origins of Python, Van Rossum wrote in 1996:[33]

...In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus).

— Guido van Rossum

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release, the development process became more transparent and community-backed.[34]

Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008 after a long testing period. It is a major revision of the language that is not completely backward-compatible with previous versions.[35] However, many of its major features have been backported to the Python 2.6.x[36] and 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.[37]

Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.[38][39]

Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and Python 3.7.0b1 (PEP 540) adds a new "UTF-8 Mode" (and overrides POSIX locale).

In January 2017, Google announced work on a Python 2.7 to go transcompiler

to improve performance under concurrent workloads. [40]

Features and philosophy

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by met programming [41] and met objects (magic methods)). [42] Many other paradigms are supported via extensions, including design by contract [43] [44] and logic programming. [45]

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has `filter()`, `map()`, and `reduce()` functions; list comprehensions, dictionaries, and sets; and generator expressions.[46] The standard library has two modules (`itertools` and `functools`) that implement functional tools borrowed from Haskell and Standard ML.[47]

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as: [48]

Beautiful is better than ugly

Explicit is better than implicit

Simple is better than complex

Complex is better than complicated

Readability counts

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite

approach. [30]

While offering choice in coding methodology, the Python philosophy rejects exuberant syntax (such as that of Perl) in favor of a simpler, less-cluttered grammar. As Alex Martelli put it: "To describe something as 'clever' is not considered a compliment in the Python culture." [49] Python's philosophy rejects the Perl "there is more than one way to do it" approach to language design in favor of "there should be one—and preferably only one—obvious way to do it". [48]

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of CPython that would offer marginal increases in speed at the cost of clarity. [50] When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python [51]—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar. [52] [53]

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

Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as *Pythonists*, *Pythonistas*, and *Pythoneers*. [54] [55]

Syntax and semantics

Main article: Python syntax and semantics

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal. [56]

Indentation

Main article: [Python syntax and semantics § Indentation](#)

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. [57] This feature is also sometimes termed the off-side rule.

Statements and control flow

Python's statements include (among others):

The assignment statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g., `x = 2`, translates to "typed variable name `x` receives a copy of numeric value 2". The (right-hand) value is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address. The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example, `x = 2`, translates to "(generic) name `x` receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable. Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g., `x = 2; y = 2; z = 2` result in allocating storage to (at most) three names and one numeric object, to which all three names are bound. Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a name will be bound to some object, which will have a type;

thus there is dynamic typing.

The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).

The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.

The while statement, which executes a block of code as long as its condition is true.

The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.

The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.

The def statement, which defines a function or method.

The with statement (from Python 2.5), which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior.

The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.

The assert statement, used during debugging to check for conditions that ought to apply.

The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.

The import statement, which is used to import modules whose functions or variables can be used in the current program. There are four ways of using import: import <module name> or from <module name> import * or import numpy as np or from numpy import pi as Pie.

The print statement was changed to the print () function in Python 3. [58]

Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will.[59][60] However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators.[61] Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through

multiple stack levels. [62]

Expressions

Some Python expressions are similar to languages such as C and Java, while some are not:

Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division and integer division. [63] Python also added the `**` operator for exponentiation.

From Python 3.5, a new infix operator intended to be used by libraries such as NumPy for matrix multiplication was introduced, which uses the `@` operator. [64][65]

In Python, `==` compares by value, versus Java, which compares numerics by value [66] and objects by reference. [67] (Value comparisons in Java on objects can be performed with the `equals()` method.) Python's `is` operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example `a <= b <= c`.

Python uses the words `and`, `or`, `not` for its Boolean operators rather than the symbolic `&&`, `||`, used in Java and C.

Python has a type of expression termed a list comprehension. Python 2.4 extended list comprehensions into a more general expression termed a generator expression. [46]

Anonymous functions are implemented using lambda expressions; however, these are limited in that the body can only be one expression.

Conditional expressions in Python are written as `x if c else y` [68] (different in order of operands from the `c? x: y` operator common to many other languages).

Python makes a distinction between lists and tuples. Lists are written as `[1, 2, 3]`, are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be immutable in Python). Tuples are written as `(1, 2, and 3)`, are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The `+` operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided tuples. Thus, given the variable `t` initially equal to `(1, 2, 3)`, executing `t = t + (4, 5)` first evaluates `t + (4, 5)`, which yields `(1, 2, 3, 4, 5)`, which is then assigned back to `t`, thereby effectively "modifying the contents" of `t`, while conforming to the

immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts. [69]

Python features sequence unpacking where multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc.), are associated in the identical manner to that forming tuple literals and, as a whole, are put on the left hand side of the equal sign in an assignment statement. The statement expects an iterable object on the right hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through, and will iterate through it, assigning each of the produced values to the corresponding expression on the left. [citation needed]

Python has a "string format" operator `%`. This functions analogous to `printf` format strings in C, e.g. `"spam=%s eggs=%d" % ("blah", 2)` evaluates to `"spam=blah eggs=2"`. In Python 3 and 2.6+, this was supplemented by the `format()` method of the `str` class, e.g. `"spam= {0} eggs= {1}".format ("blah", 2)`. Python 3.6 added "f-strings": `blah = "blah"; eggs = 2; f'spam= {blah} eggs= {eggs}'`. [70]

Python has various kinds of string literals:

Strings delimited by single or double quote marks. Unlike in Unix shells, Perl and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (`\`) as an escape character. String interpolation became available in Python 3.6 as "formatted string literals". [70]

Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like here documents in shells, Perl and Ruby.

Raw string varieties, denoted by prefixing the string literal with an `r`. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as regular expressions and Windows-style paths. Compare `"@-quoting"` in C#.

Python has array index and array slicing expressions on lists, denoted as a `[key]`, a `[start: stop]` or a `[start: stop: step]`. Indexes are zero-based, and negative indexes are relative to the end. Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted, for example a `[:]` returns a copy of the entire list. Each element of a slice is a shallow copy.

In Python, a distinction between expressions and statements is rigidly enforced,

in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality. For example:

List comprehensions vs. for-loops

Conditional expressions vs. if blocks

The `eval ()` vs. `exec ()` built-in functions (in Python 2, `exec` is a statement); the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or lambda expressions, all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as `a = 1` cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator `=` for an equality operator `==` in conditions: `if (c = 1) {...}` is syntactically valid (but probably unintended) C code but `if c = 1: ...` causes a syntax error in Python.

Methods

Methods on objects are functions attached to the object's class; the syntax `instance.method (argument)` is, for normal methods and functions, syntactic sugar for `Class.Method (instance, argument)`. Python methods have an explicit `self`-parameter to access instance data, in contrast to the implicit `self` (or `this`) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).[71]

Typing

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being dynamically typed, Python is strongly typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

Python allows programmers to define their own types using classes, which are most often used for object-oriented programming. New instances of classes are constructed by calling the class (for example, `SpamClass ()` or `EggsClass ()`), and the classes are instances of the metaclass type (itself an instance of itself), allowing

metaprogramming and reflection.

Before version 3.0, Python had two kinds of classes: old-style and new-style.[72] The syntax of both styles is the same, the difference being whether the class object is inherited from, directly or indirectly (all new-style classes inherit from object and are instances of type). In versions of Python 2 from Python 2.2 onwards, both kinds of classes can be used. Old-style classes were eliminated in Python 3.0.

The long term plan is to support gradual typing [73] and from Python 3.5, the syntax of the language allows specifying static types but they are not checked in the default implementation, CPython. An experimental optional static type checker named mypy supports compile-time type checking. [74]

Summary of Python 3's built-in types

Type	mutable	Description	Syntax example
bool	immutable	Boolean value	True False
byte array	mutable	Sequence of bytes	byte array (b'Some ASCII') byte array (b"Some ASCII") bytearray([119, 105, 107, 105])
bytes	immutable	Sequence of bytes	b'Some ASCII' b"Some ASCII" bytes ([119, 105, 107, 105])
complex	immutable	Complex number with real and imaginary parts	3+2.7j
dict	mutable	Associative array (or dictionary) of key and value pairs; can contain mixed types (keys and values), keys must be a hashable type	{'key1': 1.0, 3: False}
Ellipsis		an ellipsis placeholder to be used as an index in NumPy arrays ...	
Float	immutable	Floating point number, system-defined precision	3.1415927
Frozen set	immutable	Unordered set, contains no duplicates; can contain mixed types, if hashable	frozenset ([4.0, 'string', True])

int	immutable	Integer of unlimited magnitude[75]	42
list	mutable	List, can contain mixed types	[4.0, 'string', True]
set	mutable	Unordered set, contains no duplicates; can contain mixed types, if hashable	{4.0, 'string', True}
str	immutable	A character string: sequence of Unicode codepoints	
		'Wikipedia'	
		"Wikipedia"	
		"""Spanning	
		multiple	
		lines"""	
tuple	immutable	Can contain mixed types	(4.0, 'string', True)But we can
append elements using <code>__add__</code> .			
<code>a = (4.0, 'string' , True).__add__(('hi' ,))</code>			
now a gives (4.0, 'string', True , 'hi')			

Mathematics

Python has the usual C arithmetic operators (+, -, *, /, %). It also has `**` for exponentiation, e.g. `5**3 == 125` and `9**0.5 == 3.0`, and a new matrix multiply `@` operator is included in version 3.5.[76] Additionally, it has a unary operator (`~`), which essentially inverts all the bits of its one argument. For integers, this means `~x=-x-1`. [77] other operators include bitwise shift operators `x << y`, which shifts `x` to the left `y` places, the same as `x*(2**y)`, and `x >> y`, which shifts `x` to the right `y` places, the same as `x/(2**y)`. [78]

The behavior of division has changed significantly over time: [79]

Python 2.1 and earlier use the C division behavior. The `/` operator is integer division if both operands are integers, and floating-point division otherwise. Integer division rounds towards 0, e.g. `7/3 == 2` and `-7/3 == -2`.

Python 2.2 changes integer division to round towards negative infinity, e.g. `7/3 == 2` and `-7/3 == -3`. The floor division `//` operator is introduced. So `7//3 == 2`, `-7//3 == -3`, `7.5//3 == 2.0` and `-7.5//3 == -3.0`. Adding `from __future__ import division` causes a module to use Python 3.0 rules for division (see next).

Python 3.0 changes `/` to be always floating-point division. In Python terms, the pre-3.0 `/` is classic division, the version-3.0 `/` is real division, and `//` is floor division.

Rounding towards negative infinity, though different from most languages, adds consistency. For instance, it means that the equation $(a + b) // b == a // b + 1$ is always true. It also means that the equation $b * (a // b) + a \% b == a$ is valid for both positive and negative values of a . However, maintaining the validity of this equation means that while the result of $a \% b$ is, as expected, in the half-open interval $[0, b)$, where b is a positive integer, it has to lie in the interval $(b, 0]$ when b is negative.[80]

Python provides a `round` function for rounding a float to the nearest integer. For tie-breaking, versions before 3 use round-away-from-zero: `round(0.5)` is 1.0, `round(-0.5)` is -1.0.[81] Python 3 uses round to even: `round(1.5)` is 2, `round(2.5)` is 2.[82]

Python allows Boolean expressions with multiple equality relations in a manner that is consistent with general use in mathematics. For example, the expression $a < b < c$ tests whether a is less than b and b is less than c . C-derived languages interpret this expression differently: in C, the expression would first evaluate $a < b$, resulting in 0 or 1, and that result would then be compared with c . [83] [page needed]

Python has extensive built-in support for arbitrary precision arithmetic. Integers are transparently switched from the machine-supported maximum fixed-precision (usually 32 or 64 bits), belonging to the python type `int`, to arbitrary precision, and belonging to the python type `long`, where needed. The latter have an "L" suffix in their textual representation. [84] (In Python 3, the distinction between the `int` and `long` types was eliminated; this behavior is now entirely contained by the `int` class.) The `Decimal` type/class in module `decimal` (since version 2.4) provides decimal floating point numbers to arbitrary precision and several rounding modes. [85] The `Fraction` type in module `fractions` (since version 2.6) provides arbitrary precision for rational numbers. [86]

Due to Python's extensive mathematics library, and the third-party library NumPy that further extends the native capabilities, it is frequently used as a scientific scripting language to aid in problems such as numerical data processing and manipulation.

Libraries

Python's large standard library, commonly cited as one of its greatest strengths, [87] provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals,[88] manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation `wsgiref` follows PEP 333[89]), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of September 2017, the Python Package Index, the official repository for third-party Python software, contains over 117,000[90] packages with a wide range of functionality, including:

- Graphical user interfaces

- Web frameworks

- Multimedia

- Databases

- Networking

- Test frameworks

- Automation

- Web scraping [91]

- Documentation

- System administration

- Scientific computing

- Text processing

- Image processing

- Development environments

See also: Comparison of integrated development environments § Python

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and I Python, add further abilities such as auto-completion, session state retention and syntax highlighting.

As well as standard desktop integrated development environments (see Wikipedia's "Python IDE" article), there are Web browser-based IDEs; Sage Math (intended for developing science and math-related Python programs); Python Anywhere, a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing scientific computing.[92]

Implementations

See also: List of Python software § Python implementations

Reference implementation

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features. [93] It compiles Python programs into an intermediate bytecode [94] which is then executed by its virtual machine. [95] CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities. [96]

Other implementations

PyPy is a fast, compliant [97] interpreter of Python 2.7 and 3.5. Its just-in-time compiler brings a significant speed improvement over CPython. [98] A version is under development that will use software transactional memory to take advantage of multi-core processors. [99]

Stackless Python is a significant fork of Python that implements micro threads; it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a stackless version. [100]

MicroPython is a Python 3 variant optimised for microcontrollers.

Unsupported implementations

Other just-in-time Python compilers have been developed, but are now unsupported:

Google began a project named Unladen Swallow in 2009 with the aim of speeding up the Python interpreter fivefold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores. [101]

Psyco is a just-in-time specializing compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialised for certain data types and is faster than standard Python code.

In 2005, Nokia released a Python interpreter for the Series 60 mobile phones named PyS60. It includes many of the modules from the CPython implementations and some additional modules to integrate with the Symbian operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device. [102]

Cross-compilers to other languages

There are several compilers to high-level object languages, with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

Python compiles into Java byte code, which can then be executed by every Java virtual machine implementation. This also enables the use of Java class library functions from the Python program.

IronPython follows a similar approach in order to run Python programs on the .NET Common Language Runtime.

The RPython language can be compiled to C, Java bytecode, or Common Intermediate Language, and is used to build the PyPy interpreter of Python.

Pyjs compiles Python to JavaScript.

Cython compiles Python to C and C++.

Pythran compiles Python to C++.

Somewhat dated Pyrex (latest release in 2010) and Shed Skin (latest release in 2013) compile to C and C++ respectively.

Google's Grumpy compiles Python to Go.

Nuitka compiles Python into C++ [103]

Performance

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13. [104]

Development

Python's development is conducted largely through the Python Enhancement Proposal (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues and documenting Python design decisions.[105] Outstanding PEPs are reviewed and commented on by the Python community and Guido Van Rossum, Python's Benevolent Dictator For Life.[105]

Enhancement of the language corresponds with development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues are discussed in the Roundup bug tracker maintained at python.org. [106] Development originally took place on a self-hosted source-code repository running Mercurial, until Python moved to GitHub in January 2017. [107]

CPython's public releases come in three types, distinguished by which part of the version number is incremented:

Backward-incompatible versions, where code is expected to break and need to be manually ported. The first part of the version number is incremented. These

releases happen infrequently—for example, version 3.0 was released 8 years after 2.0.

Major or "feature" releases, about every 18 months, are largely compatible but introduce new features. The second part of the version number is incremented. Each major version is supported by bugfixes for several years after its release. [108]

Bugfix releases, which introduce no new features, occur about every 3 months and are made when a sufficient number of bugs have been fixed upstream since the last release. Security vulnerabilities are also patched in these releases. The third and final part of the version number is incremented. [109]

Many alpha, beta, and release-candidates are also released as previews and for testing before final releases. Although there is a rough schedule for each release, they are often delayed if the code is not ready. Python's development team monitors the state of the code by running the large unit test suite during development, and using the BuildBot continuous integration system. [110]

The community of Python developers has also contributed over 86,000[111] software modules (as of 20 August 2016) to the Python Package Index (PyPI), the official repository of third-party Python libraries.

The major academic conference on Python is PyCon. There are also special Python mentoring programmers, such as Pyladies.

Naming

Python's name is derived from the British comedy group Monty Python, whom Python creator Guido van Rossum enjoyed while developing the language. Monty Python references appear frequently in Python code and culture;[112] for example, the metasyntactic variables often used in Python literature are spam and eggs instead of the traditional foo and bar.[112][113] The official Python documentation also contains various references to Monty Python routines.[114][115]

The prefix Py- is used to show that something is related to Python. Examples of the use of this prefix in names of Python applications or libraries include Pygame, a

binding of SDL to Python (commonly used to create games); Python for S60, an implementation for the Symbian S60 operating system; PyQt and PyGTK, which bind Qt and GTK to Python respectively; and PyPy, a Python implementation originally written in Python.

Uses

Main article: List of Python software

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index. As of January 2018, it is the fourth most popular language. [116] it was selected Programming Language of the Year in 2007 and 2010. [117] it is the third most popular language whose grammatical syntax is not predominantly based on C.

An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".[118]

Large organizations that use Python include Wikipedia, Google, [119] Yahoo!, [120] CERN, [121] NASA, [122] and some smaller entities like ILM [123] and ITA. [124] the social news networking site Reddit is written entirely in Python.

Python can serve as a scripting language for web applications, e.g., via `mod_wsgi` for the Apache web server. [125] With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used as data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.

Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python

in scientific computing, [126] [127] with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a mathematical software with a "notebook" programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus. The Python language re-implemented in Java platform is used for numeric and statistical calculations with 2D/3D visualization by the DMelt project. [128][129]

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, [130] Inkscape, Scribus and Paint Shop Pro, [131] and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. [132] It has also been used in several video games, [133] [134] and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go. [135] Python is also used in algorithmic trading and quantitative finance. [136] Python can also be implemented in APIs of online brokerages that run on other languages by using wrappers. [137]

Python has been used in artificial intelligence projects. [138][139][140][141] as a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing. [142]

Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD, NetBSD, OpenBSD and macOS, and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

Python is used extensively in the information security industry, including in

exploit development. [143][144]

Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. [145]

The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.

LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature [146] since Version 4.0 from 7 February 2013.

Languages influenced by Python

Python's design and philosophy have influenced many other programming languages:

Boo uses indentation, a similar syntax, and a similar object model. [147]

Cobra uses indentation and a similar syntax, and its "Acknowledgements" document lists Python first among languages that influenced it. [148] However, Cobra directly supports design-by-contract, unit tests, and optional static typing. [149]

CoffeeScript, a programming language that cross-compiles to JavaScript, has Python-inspired syntax.

ECMAScript borrowed iterators, generators and list comprehensions from Python. [150]

Go is described as incorporating the "development speed of working in a dynamic language like Python". [151]

Groovy was motivated by the desire to bring the Python design philosophy to Java. [152]

Julia was designed "with true macros [... and to be] as usable for general programming as Python [and] should be as fast as C". [21] Calling to or from Julia is

possible; to with PyCall.jl and a Python package pyjulia allows calling, in the other direction, from Python.

OCaml has an optional syntax named twt (The Whitespace Thing), inspired by Python and Haskell. [153]

Ruby's creator, Yukihiro Matsumoto, has said: "I wanted a scripting language that was more powerful than Perl, and more object-oriented than Python. That's why I decided to design my own language." [154]

Swift, a programming language developed by Apple, has some Python-inspired syntax. [155]

Python's development practices have also been emulated by other languages. For example, the practice of requiring a document describing the rationale for, and issues surrounding, a change to the language (in Python, a PEP) is also used in Tcl [156] and Erlang. [157]

Python received TIOBE's Programming Language of the Year awards in 2007 and 2010. The award is given to the language with the greatest growth in popularity over the year, as measured by the TIOBE index. [158]

4.6 Experimental Results:

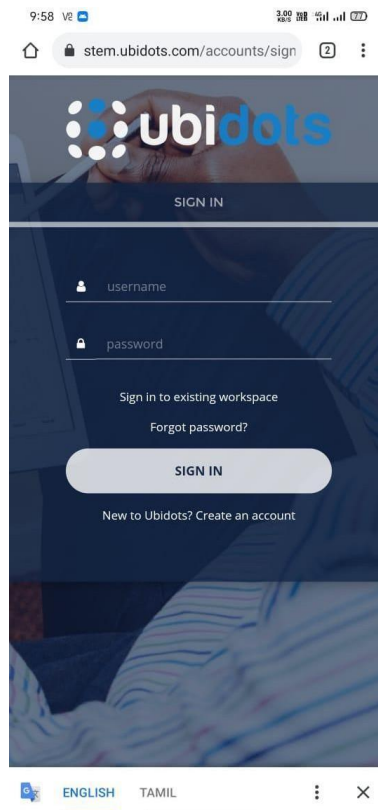
Sensors data will be uploaded to the IOT so that variations in the sensors, means the the sensors value goes beyond the threshold value at that time from internet only messages/email will be sent concerned persons/police station.

CONCLUSION:

This type of an idea being the first of its kind plays a crucial role towards ensuring Women Safety in the fastest way possible automatically. The proposed design will deal with critical issues faced by women in the recent past and will help solve them through technologically sound gadgets.

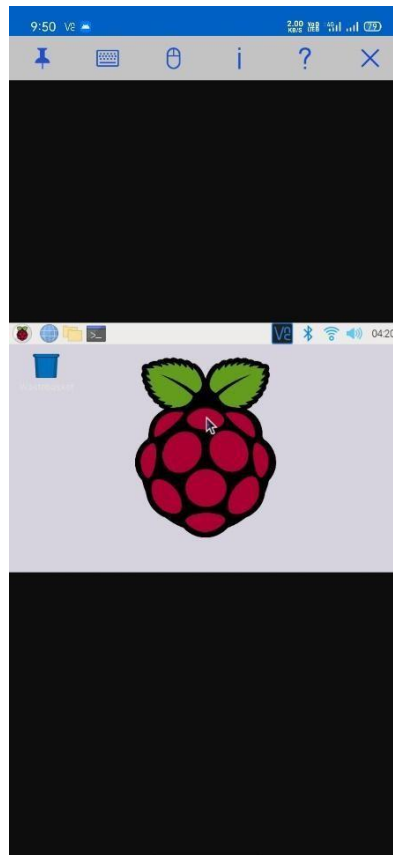
With further research and innovation, this project can be implemented in different areas of security and surveillance. The system can perform the real time monitoring of desired area and detect the violence with a good accuracy.

OUTPUT SCREENSHOTS:

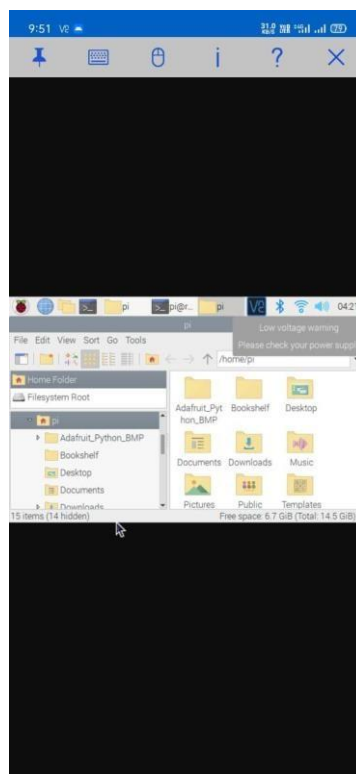


Ubidots Login page

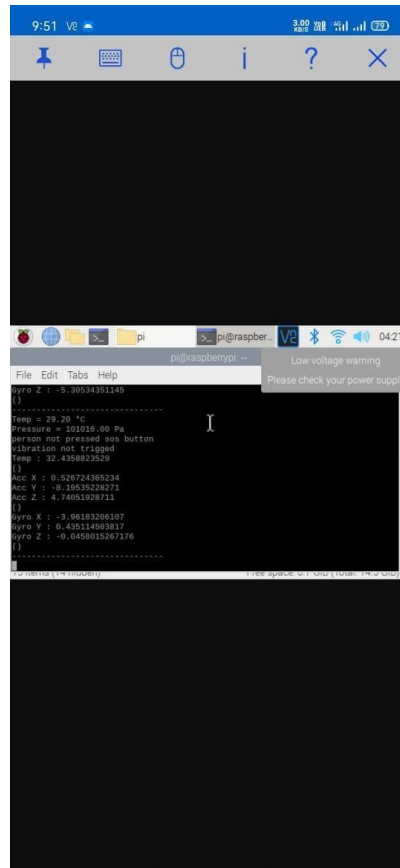
Ubidots is used to access the cloud platform of the device where all device information is stored



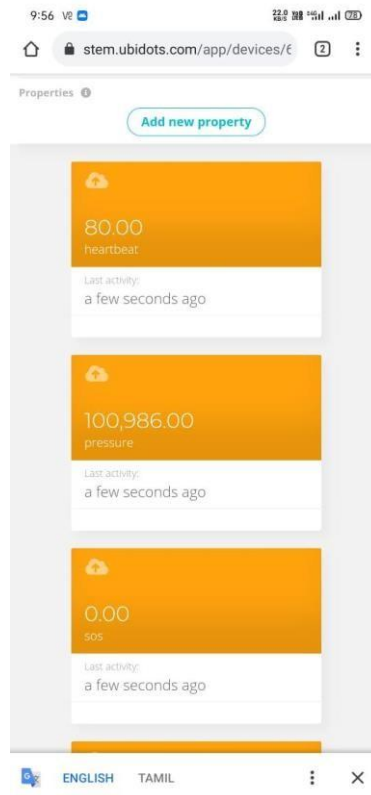
User interface of VNC viewer



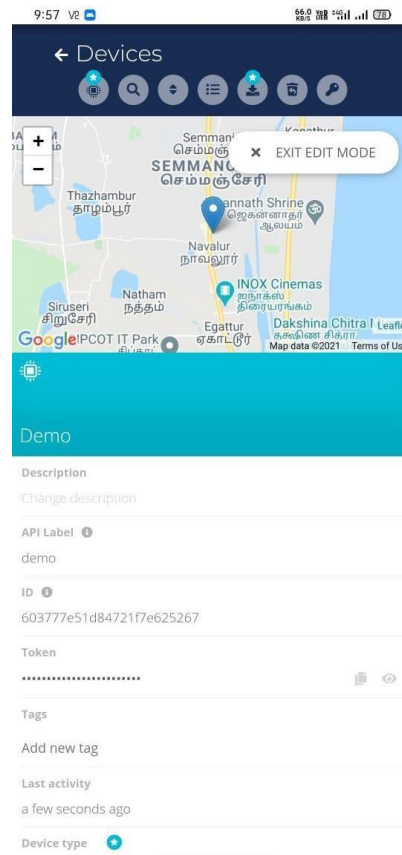
File system in VNC



Device data



Cloud data of device in ubidots



Location of device

REFERENCES:

- [1] B.Vijaylashmi¹, Renuka.S², Pooja Chennur³, Sharangowda.Patil⁴, "Self defence system for women safety with location tracking and SMS alerting through Gsm network.IJRET: International Journal of Research in Engineering and Technology ISSN: 2319-1163 ISSN: 2321-7308.
- [2] Niti shree "A Review on IOT Based Smart GPS Device for Child and Women Safety", IJERGS, May-June 2016.
- [3] Swapnali N.Gadhav¹, Saloni D. Kale², Sonali N. Shinde³, Prof. Amol C. Bhosale⁴, "Electronic Jacket For Women Safety", IRJET, May 2017.
- [4] Hung Nguyen, Karina Lebel, Sarah Bogard, Etienne Goubault, Patrick Boissy, and Christian Duval," Using Inertial Sensors to Automatically Detect and Segment Activities of Daily Living in People With Parkinson's Disease", Transactions On Neural Systems

- [5] Shubham Sharma¹, Fasil Ayaz², Rajan Sharma³, Divya Jain⁴, "IoT Based Women Safety Device using ARM7", IJESC, May 2017.
- [6] Ms. Deepali M. Bhavale, Ms. Priyanka S. Bhawale, Ms. Tejal Sasane, Mr. Atul S. Bhawale, "IOT Based Unified Approach for Women and Children Security Using Wireless and GPS", IJARCET, August 2016.
- [7] Prof. R.A.Jain¹, Aditya Patil², Prasenjeet Nikam³, Shubham More⁴, Saurabh Totewar⁵, "Women's safety using IOT", IRJET, May 2017.
- [8] SeokJu Lee, Girma Tewolde, Jaerock Kwon ||Design and Implementation of Vehicle Tracking System Using GPS/GSM/GPRS Technology and Smartphone Application||IEEEWorld Forum on Internet of Things (WF-IoT), March 2015, Seoul
- [9] D. Thierer, ||The internet of things and wearable technology: Addressing privacy and security concerns without derailing innovation||Rich. J. Law Technol., vol. 21, pp. 615, 2015.
- [10] Muruganandham, ||Real TimeWeb based Vehicle Tracking using GPS||, World Academy of Science, Engineering and Technology, 37, 2018
- [11] R.Ramani, S.Valarmathy, N.SuthanthiraVanitha, S.Selvaraju, and M.Thiruppathi, Vehicle Tracking and Locking System Based on GSM and GPS, I.J. Intelligent Systems and Applications, 2019, 09, 86-93
- [12] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, —Identify authentication and capability based access control (IACAC) for the internet of things, || J. Cyber Security Mobility, vol. 1, pp. 309–348, and 2018
- [13] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, —Privacy in the internet of things: Threats and challenges, || Security Commun. Netw. vol. 7, no. 12, pp. 2728–2742, 2019
- [14] D. Thierer, —The internet of things and wearable technology: Addressing privacy

and security concerns without derailing innovation, || Rich. J. Law Technol., vol. 21, pp. 6–15, 2018.

[15] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, —Constrained application protocol (CoAP), draft-it-core-coap- 13, || in The Internet Engineering Task Force (IETF), 2019.

[16] M. Brachmann, S. L. Keoh, O. Morchon, and S. Kumar, —End-toend transport security in the ip-based internet of things, || in Proc. 21st Int. Conf. Comput. Commun. Netw., 2017, pp. 1–5.

[17] R. Seggelmann, —Sctp: Strategies to secure end-to-end communication, || Ph.D. dissertation, Univ. Duisburg-Essen, Essen, Germany, 2019.

[18] F. McKeen, I. Alexandrovich, A. Berenzon, C. Rozas, H. Shafi, V. Shanbhogue, and U. Savagaonkar, —Innovative instruction and software model for isolated execution,|| in Proc. 2nd Int. Workshop Hardware Architectural Support Security Privacy, 2018.

[19] Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata, —Innovative technology for CPU based attestation and sealing, || in Proc. 2nd Int.Workshop Hardware Architectural Support Security Privacy, 2019.