IOT BASED HOME SECURITY USING RASPBERRY PIE

Submitted in partial fulfillment of the requirements

Of the degree of

Bachelor of Engineering

By

KHAN SAMEER NASIR

PATHAN UMAIR SHAHNAZ

KHOT AARAF ABDULLAH

Supervisor:

Asst Prof. Riyaz Pathan

Department of Electronics & Telecommunication

AIKTC

2017

Project Report Approval for B.E.

This project report entitled***Iot Based Home Security Using Raspberry Pie***by***khan sameer nasir, pathan umair shahnaz, khot aaraf abdullah***is approved for the degree of ***B.E. in Electronics & Telecommunication***.

Examiners

1.---------------------------------------

2.---------------------------------------

Supervisor

1.---------------------------------------

Chairman

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sameer Nasir Khan (12ET81)

Pathan Umair Shahnaz (12ET64)

Khot Aaraf Abdulla (12ET65)

Date:

ABSTRACT

Home security and automation are becoming increasingly prominent features on mobile devices. The aim of this paper is to design and implement affordable, flexible and fast monitoring home security system using Raspberry pi with IOT technology. The system is designed to detect burglary, water level detection, motor, lights, fan control and sends to mobile and email as well as alarm gets on; and leaking in harmful gas, the smoke caused by fire as such suspicious activity is detected. Also the user can activate all the alarm system while going outside through the mobile. The messenger has the feasibility of activating and deactivating the alarm system with the additional control for some home appliances switching using relays. For controlling, raspberry pi module, IOT technology is used along with, Gas sensor, door sensor, window sensor. It is only applicable for magnetic door lock system.

TABLE OF CONTENTS

INTRODUCTION

PROBLEM STATEMENT

When the Raspberry pie is switched on it need to be connected to the internet using Wi-Fi or Ethernet cable.

Coding is done using C program and uploaded on the Cloud.

Here we are using Particle Cloud.

Then we have to install an application in our Android or iOS device called Blynk explained in the previous slide.

Then register on the app.

Using the Blynk app. we have to configure our different sensors and load switches accordingly.

After configuring the app. we are ready to go.

As both our Raspberry pie and the Blynk app. are connected to the Internet we can access our sensors and switch from any corner of the world very easily.

1.2 Project objectives

The objectives of this project are:

- We will be using Raspberry pi as our main module to connect different sensors.
- And the Raspberry pi will be connected to the internet using Wi-Fi or Ethernet.
- Coding is done in the cloud i.e. Particle Cloud using C program.
- Blynk is been used in our mobile phone to communicate with the Raspberry Pi to which our sensors and devices are connected wirelessly.

2. SPECIFICATION FOR SYSTEM DEVELOPMENT

In this section, we would like to highlight the targeted hardware platform, software platform and programming languages which are intended to use for development of propose embedded system.

What is IOT?

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 that enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "the infrastructure of the information society. The IoT allows objects to be sensed or controlled remotely across existing network infrastructure,[4] 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 almost 50 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.[12] 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,[13] 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 to look at

"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. The term "the Internet of Things" was coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999.

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).

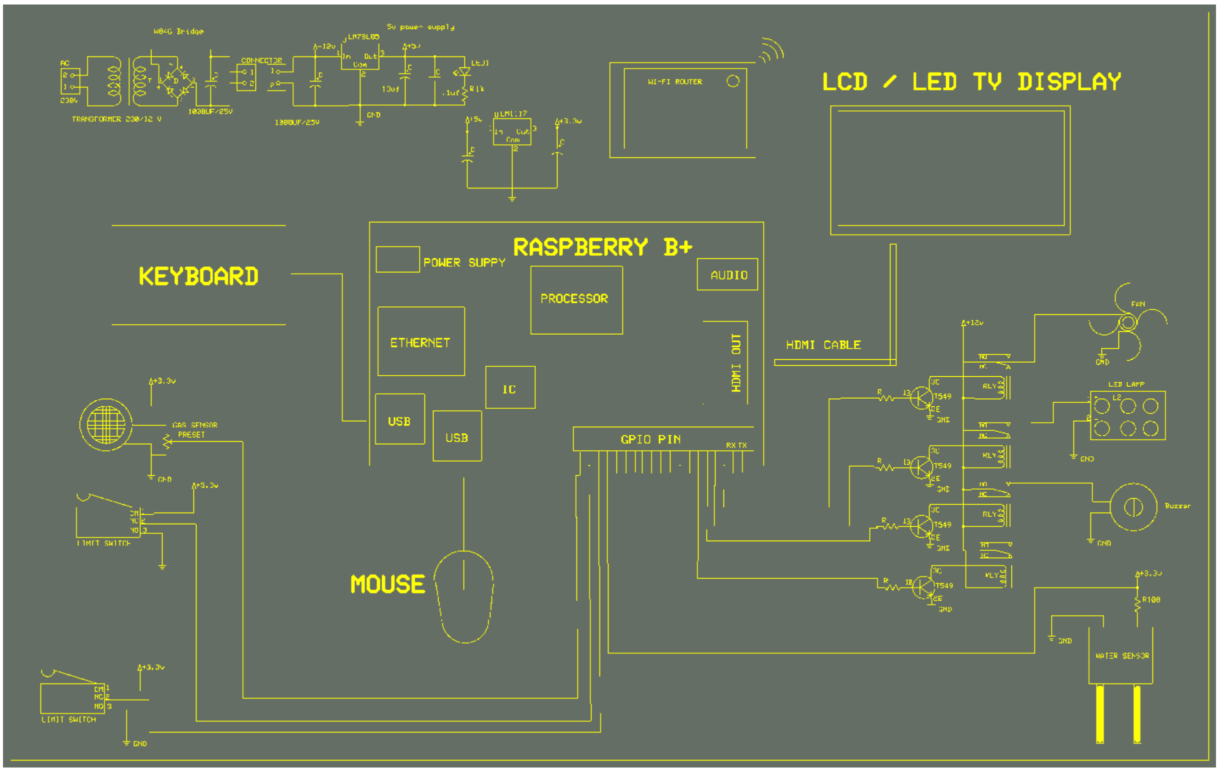
The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold. Mark Weiser's seminal 1991 paper on ubiquitous 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".[28] 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 Joyen visioned 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 Center 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 Center) as a prerequisite for the Internet of things at that point.[31] 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.

HARDWARE SPECIFCATION

CIRCUIT DIAGRAM

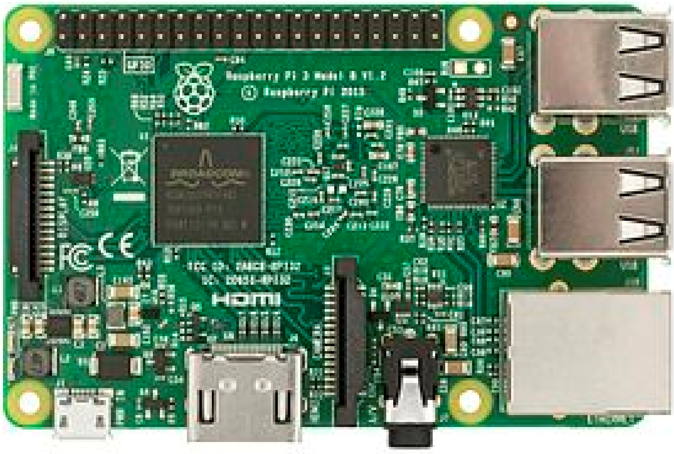


Raspberry Pi

The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles.

According to the Raspberry Pi Foundation, over 5 million Raspberry Pi is have been sold before February 2015, making it the best-selling British computer. By November 2016 they had sold 11 million units.

Overview



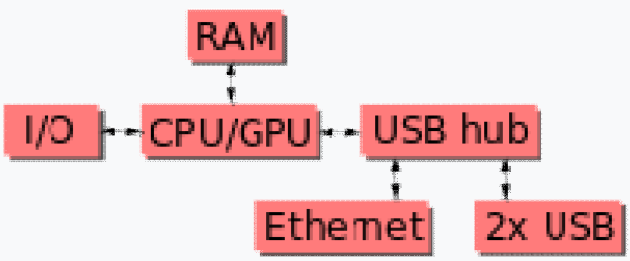
Several generations of Raspberry Pi have been released. The first generation (**Raspberry Pi 1 Model B**) was released in February 2012. It was followed by a simpler and inexpensive model **Model A**. In 2014, the foundation released a board with an improved design in **Raspberry Pi 1 Model B+**. These boards are approximately credit-card sized and represent the standard *mainline* form-factor. Improved A+ and B+ models were released a year later. A "compute module" was released in April 2014 for embedded applications, and a **Raspberry Pi Zero** with smaller size and reduced input/output (I/O) and general-purpose input/output (GPIO) capabilities was released in November 2015 for US$5. The **Raspberry Pi 2** which added more RAM was released in February 2015. **Raspberry Pi 3 Model B** released in February 2016, is bundled with on-board Wi-Fi, Bluetooth and USB boot capabilities.[11] As of January 2017, **Raspberry Pi 3 Model B** is the newest mainline Raspberry Pi. Raspberry Pi boards are priced between US$5–35. As of 28 February 2017, the **Raspberry Pi Zero W** was launched, which is identical to the Raspberry Pi Zero, but has the Wi-Fi and Bluetooth functionality of the Raspberry Pi 3 for US$10.

All models feature a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a Video Core IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes. Most boards have between one and four USB slots, HDMI and composite video output, and a 3.5 mm phone jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I²C. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on board Wi-Fi 802.11n and Bluetooth.

The Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third party Ubuntu, Windows 10 IOT Core, RISC OS, and specialized media center distributions. It promotes Python and Scratch as the main programming language, with support for many other languages. The default firmware is closed source, while an unofficial open source is available.

Hardware

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



This block diagram depicts Models A, B, A+, and B+. Model A, A+, and the Pi Zero lack the Ethernet and USB hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the system on a chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-point USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

Processor

The Raspberry Pi 2 uses a 32-bit 900 MHz quad-core ARM Cortex-A7processor.

The Broadcom BCM2835 SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in first generation smartphones (its CPU is an olderARMv6 architecture),[16] which includes a 700 MHz ARM1176JZF-S processor, VideoCore IV graphics processing unit (GPU), and RAM. It has a level 1 (L1)cache of 16 KB and a level 2 (L2) 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.

The Raspberry Pi 2 uses a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor (as do many current smartphones), with 256 KB shared L2 cache.

The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

Performance

The Raspberry Pi 3, with a quad-core Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1.[19] This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.

Raspberry Pi 2 includes a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It is described as 4–6 times more powerful than its predecessor. The GPU is identical to the original.[18] In parallelized benchmarks, the Raspberry Pi 2 could be up to 14 times faster than a Raspberry Pi 1 Model B+.

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 1997–99. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capabilities of the Raspberry Pi are roughly equivalent to the 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-pi", achieved a LINPACK HPL suite result of 1.14 GFLOPS (n=10240) at 216 watts for c. US$4000.

Overclocking

The CPU chips of the first and second generation Raspberry Pi board did not require cooling, such as a heat sink, unless the chip was overclocked, but the Raspberry Pi 2 SoC may heat more than usual under overclocking.

Most Raspberry Pi chips could be overclocked to 800 MHz, and some to 1000 MHz. There are reports the Raspberry Pi 2 can be similarly overclocked, in extreme cases, even to 1500 MHz (discarding all safety features and over-voltage limitations). In the Raspbian Linux distro 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 if the chip reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and overclocking settings (voiding the warranty); an appropriately sized heatsink is needed to protect the chip from serious overheating.

Newer versions of the firmware contain the option to choose between five overclock ("turbo") presets that when used, attempt to maximize the performance of the SoC without impairing the lifetime of the board. This is done by monitoring the core temperature of the chip, 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 seven overclock presets 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,
- Pi2; 1000 MHz ARM, 500 MHz core, 500 MHz SDRAM, 2 overvolt,
- Pi3; 1100 MHz ARM, 550 MHz core, 500 MHz SDRAM, 6 overvolt. In system information CPU speed will appear as 1200 MHz. When in idle speed lowers to 600 MHz.

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 250 MHz.

The Raspberry Pi Zero runs at 1 GHz.

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 only a 1080p framebuffer, 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 VideoCore 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 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

Networking

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly 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 using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi

Zero W (wireless) are equipped with 2.4 GHz WiFi802.11n (150Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on Broadcom BCM43438 FullMAC chip with no official support forMonitor mode but implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Ethernet port.

Peripherals

The current Model B boards incorporate four USB ports for connecting peripherals.

The Raspberry Pi may be operated with any generic USB computer keyboard andmouse. It may also be used with USB storage, USB to MIDI converters, and*virtually* any other device/component with USB capabilities.

Other peripherals can be attached through the various pins and connectors on the surface of the Raspberry Pi.

Video

The early Raspberry Pi 1 Model A, with an HDMI port and a standard RCA composite video port for older displays

The video controller can emit 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 can emit these: 640×350 EGA; 640×480 VGA; 800×600 SVGA; 1024×768 XGA; 1280×720 720p HDTV; 1280×768WXGA variant; 1280×800 WXGA variant; 1280×1024 SXGA; 1366×768 WXGAvariant; 1400×1050 SXGA+; 1600×1200 UXGA; 1680×1050 WXGA+; 1920×10801080p HDTV; 1920×1200 WUXGA.[38]

Higher resolutions, such as, up to 2048×1152, may work or even 3840×2160 at 15 Hz (too low a framerate for convincing video). Note also that allowing the highest resolutions does not imply that the GPU can decode video formats at those; in fact, the Pi are known to not work reliably for H.265 (at those high resolutions), commonly used for very high resolutions (most formats, commonly used, up to Full HD, do work).

Although the Raspberry Pi 3 does not have H.265 decoding hardware, the CPU is more powerful than its predecessors, potentially fast enough to allow the decoding of H.265-encoded videos in software.[42] The GPU in the Raspberry Pi 3 runs at a higher clock frequencies of 300 MHz or 400 MHz, compared to previous versions which ran at 250 MHz.[43]

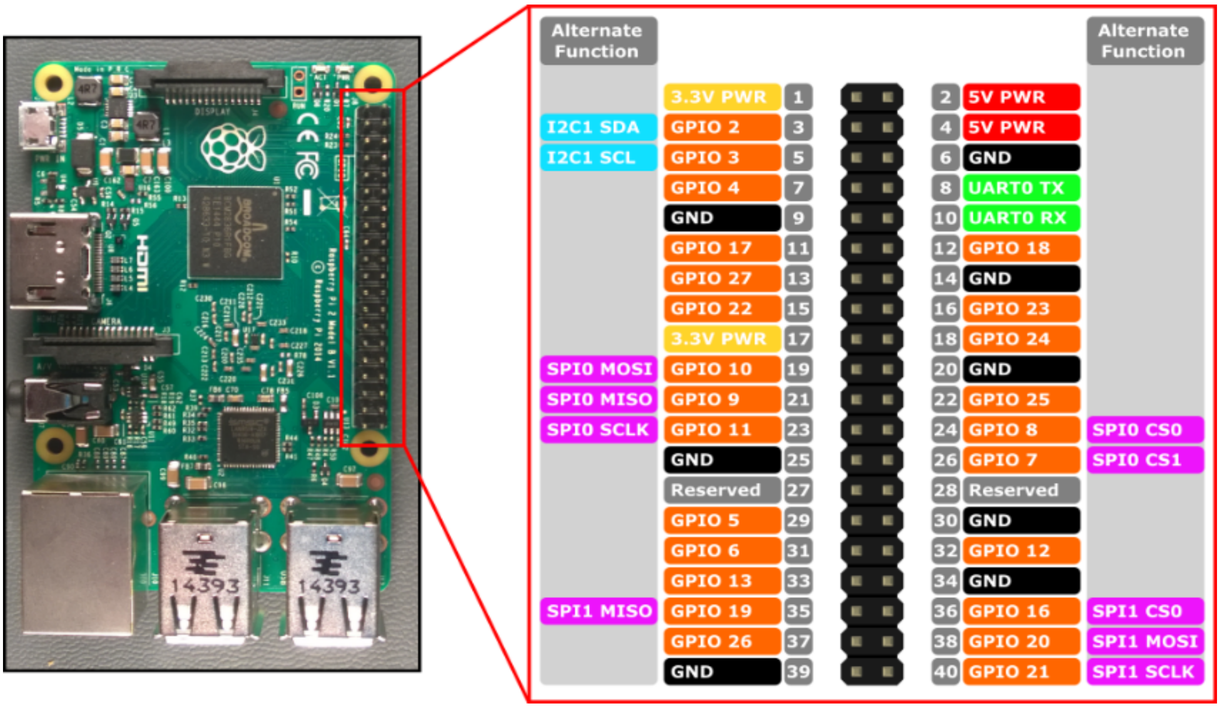
The Raspberry Pi can also generate 576i and 480i composite video signals, as used on old-style (CRT) TV screens and less-expensive monitors through standard connectors – either RCA or 3.5 mm phone connector depending on models. The television signal standards supported are PAL-BGHID, PAL-M, PAL-N, NTSC and NTSC-J.

Real-time clock

None of the current Raspberry Pi models have a built-in real-time clock, so they are unable to keep track of the time of day independently. As a workaround, a program running on the Pi can retrieve the time from a network time server or from user input at boot time, thus knowing the time while powered on. To provide consistency of time for the file system, the Pi does automatically save the time it has on shutdown, and re-installs that time at boot.

A real-time hardware clock with battery backup, such as the DS1307, which is fully binary coded, may be added (often via the I^2C interface).

General purpose input-output (GPIO) connector

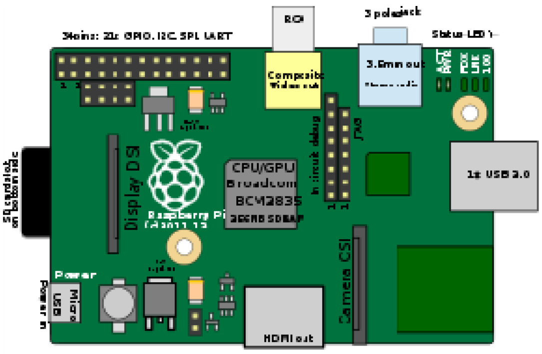


Raspberry Pi 1 Models A+ and B+, Pi 2 Model B, Pi 3 Model B and Pi Zero (and Zero W) GPIO J8 have a 40-pin pinout. Models A and B have only the first 26 pins.

Function	2nd func.	Pin#		Pin#	2nd func.	Function
N/A	+5 V	1		2	+3.3 V	N/A
GPIO28	GPIO_GEN7	3		4	GPIO_GEN8	GPIO29
GPIO30	GPIO_GEN9	5		6	GPIO_GEN10	GPIO31
N/A	GND	7		8	GND	N/A

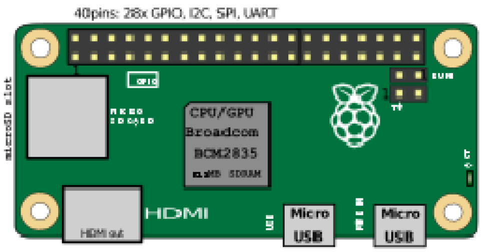
Model B rev. 2 also has a pad (called P5 on the board and P6 on the schematics) of 8 pins offering access to an additional 4 GPIO connections.

Model A

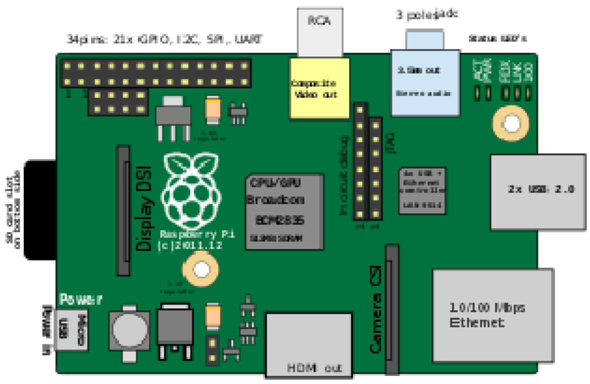


Location of connectors and main ICs on Raspberry Pi 1 Model A

Pi Zero

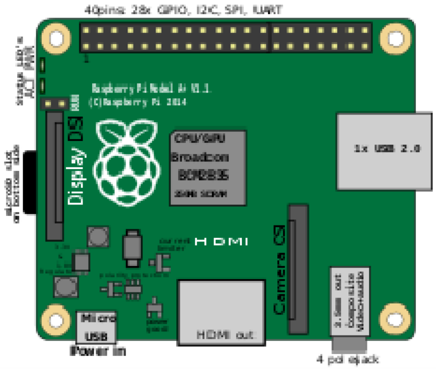


Model B



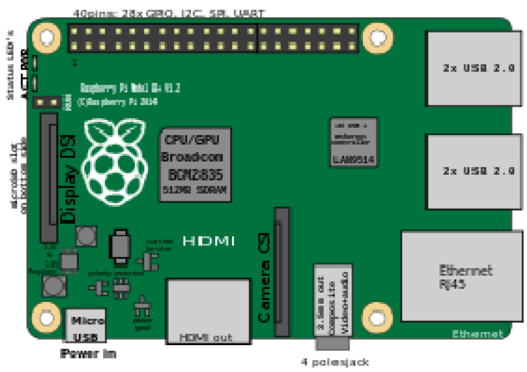
Location of connectors and main ICs on Raspberry Pi 1 Model B revision 1.2

Model A



Location of connectors and main ICs on Raspberry Pi 1 Model A+ revision 1.1

Model B



Location of connectors and main ICs on Raspberry Pi 1 Model B+ revision 1.2 and Raspberry Pi 2

Operating systems

The Raspberry Pi Foundation recommends the use of Raspbian, a Debian-based Linux operating system. Other third party operating systems available via the official website include Ubuntu MATE, Snappy Ubuntu Core, Windows 10 IoT

Core, RISC OS and specialized distributions for the Kodi media center and classroom management.

Many other operating systems can also run on the Raspberry Pi.

Other operating systems (not Linux-based)

- RISC OS Pi (a special cut down version RISC OS Pico, for 16 MB cards and larger for all models of Pi 1 & 2, has also been made available.)
- FreeBSD
- NetBSD
- Plan 9 from Bell Labs and Inferno (in beta)
- Windows 10 IoT Core – a no-cost edition of Windows 10 offered by Microsoft that runs natively on the Raspberry Pi 2.
- xv6 – is a modern reimplementation of Sixth Edition Unix OS for teaching purposes; it is ported to Raspberry Pi from MIT xv6; this xv6 port can boot from NOOBS.
- Haiku – is an open source BeOS clone that has can be compiled for the Raspberry Pi and several other ARM boards. Work on Pi 1 began in 2011, but only the Pi 2 will be supported.
- HelenOS – a portable microkernel-based multiserver operating system; has basic Raspberry Pi support since version 0.6.0
- Genode OS Framework – supports the Raspberry Pi platform with the base-hw
- kernel since release 13.05

Other operating systems (Linux-based)

- Android Things – an embedded version of the Android operating system designed for IoT device development.
- Arch Linux ARM – a port of Arch Linux for ARM processors.
- Xbian[108] – using the Kodi (formerly XBMC) open source digital media center
- openSUSE
- Raspberry Pi Fedora Remix
- Pidora, another Fedora Remix optimized for the Raspberry Pi

- Gentoo Linux
- Diet Pi, includes a diverse range of servers for media, VPN, Minecraft and many others\
- CentOS for Raspberry Pi 2 and later
- RedSleeve (a RHEL port) for Raspberry Pi 1
- Slackware 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 Slackware Linux on an ARM or i386 system.(Whereas the majority of Linux systems boot into a graphical user interface, Slackware's default user environment is the textual shell / command line interface.) The Fluxbox window manager running under the X Window System requires an additional 48 MB of RAM.
- Moebius – is a light ARM HF distribution based on Debian. It uses Raspbian repository, but it fits in a 128 MB SD card. It has only minimal services and its memory use is optimized to be small.
- OpenWrt – is primarily used on embedded devices to route network traffic.
- Kali Linux – is a Debian-derived distro designed for digital forensics and penetration testing.
- SolydXK – is a light Debian-derived distro with Xfce.
- Pardus ARM – is a Debian-based operating system which is the light version of the Pardus (operating system).
- Instant WebKiosk – is an operating system for digital signage purposes (web and media views).
- Ark OS – is designed for website and email self-hosting.
- ROKOS– is a Raspbian-based operating system with integrated clients for the Bitcoin and OKCash cryptocurrencies.
- Kano OS
- Nard SDK – is a software development kit (SDK) for industrial embedded systems.
- Sailfish OS with Raspberry Pi 2 (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.
- IPFire – is a dedicated firewall/router distribution for the protection of a SOHO LAN; runs only on a Raspberry Pi 1; porting to the Raspberry Pi 2 is not planned for now.

- Alpine Linux – is a Linux distribution based on musl and BusyBox, primarily designed for "power users who appreciate security, simplicity and resource efficiency".
- Void Linux – a rolling release Linux distribution which was designed and implemented from scratch, provides images based on musl or glibc.
- Tingbot OS – based on Raspbian, primarily designed for use with the Tingbot addon and running Tide apps.
- WTware for Raspberry Pi – is a free operating system for creating Windows thin client from Pi 2 and Pi 3.
- Fedora 25 – supports Pi 2 and later (Pi 1 is supported by some unofficial derivatives, e.g. listed here.).
- Media center operating systems: OSMC, OpenELEC, LibreELEC, Xbian, Rasplex
- Audio operating systems : Volumio, Pimusicbox, Runeaudio, moOdeaudio
- Retrogaming operating systems: Retropie, Recalbox, Happi Game Center, Lakka, ChameleonPi, Piplay
- Puppy Linux

Particle Cloud

The Particle Cloud is the powerful centerpiece of the Particle platform, handling many of the most complex pieces of creating an IoT product. From robust security, to reliable infrastructure, to a flexible integrations system, the Particle Cloud has everything you need to move quickly and succeed. With Particle's Raspberry Pi agent, you can program in C/C++ to trigger a pin, blink an LED or read sensor values, among many other things. Particle is a full-stack IoT device platform.Particle includes everything you need to deploy an IoT product: a device cloud platform, connectivity hardware, and even SIMs for cellular products.IoT is hard. Particle makes it easy. According to Gartner, 80% of IoT projects are significantly delayed and over budget due to the complexity of IoT. Particle addresses the biggest challenges of building an IoT product so that you can be in the 20% that get to market quickly.

Secure By Default

Particle devices are inherently secure due to our encrypted communications protocol and the limited attack vectors of a microcontroller.

Sourcing Done For You

We select best-in-class, reliable, and affordable connectivity modules and chips from top semiconductor companies so you don't have to.

Close Your Skills Gap

Building an IoT product typically requires deep expertise. With Particle, a small team of engineers and developers without previous IoT experience can ship a product in months.

Oversee Your Fleet

Ship your IoT product with the peace of mind that you can always see your devices, push software updates, and make changes and improvements on an ongoing basis.

Installing Particle Cloud

In order to connect your Raspberry Pi to the Particle Cloud you'll need the following things. Note that these are all included in the Particle Pi Starter Kit with Raspberry Pi v3, which is available for purchase in the Particle Store.

- Raspberry Pi (Raspberry Pi v2 and v3 preferred)
- Power supply (5V, 2A+ preferred)
- Micro SD card and SD adapter
- Ethernet cable (for wired connections)

If you do not have access to a wired network cable, you will need to connect your Pi to an active Wi-Fi network, which will require the following:

- Keyboard
- Mouse
- Monitor
- HDMI Cable (to connect the Pi to your monitor)

Download and install Raspbian

Before you boot up your Pi for the first time, you'll need to make sure you have the latest Raspbian image from the Raspberry Pi Foundation. Note that flashing a fresh version of Raspbian Jessie with Pixel (GUI) can take as long as 10-15 minutes.

Do you already have a Pi with Raspbian installed? Click here to skip the download and setup steps and update your existing Raspbian image.

I DON'T HAVE AN SD CARD WITH RASPBIAN

If you don't already have an SD card with Raspbian on it, you'll need to follow these steps:

1. Make sure your SD card is FAT32 formatted
2. Install an operating system image on the SD card. We recommend Raspberry Pi's preferred operating system, Raspbian Jessie with Pixel, which you can download here.
3. Install the operating system onto your SD card by following the Raspberry Pi Foundation's official installation instructions, here.

Note: There are many different tools and resources available on the Internet to make the process of burning a new image for your Raspberry Pi easier. If you have issues with the instructions above from the Raspberry Pi
Foundation, elinux.org has compiled a great list of alternatives for Mac, Windows, and Linux.

- Mac setup options
- Windows setup options
- Linux setup options

4. Insert the SD card into your Raspberry Pi, and apply power using a 5V, 2A+ power supply.

I HAVE AN SD CARD WITH RASPBIAN

If you already have a Pi set up, run the following commands from your Raspberry Pi's command line to update your OS to the most recent version of Raspbian:

1. sudo apt-get update, which will update your local package database with the upstream one.
2. sudo apt-get upgrade, which will actually upgrade your Raspbian image to the most recent from the Raspberry Pi Foundation.

Note that these steps may take **up to 10 minutes** to complete, so please have patience.

Connect your Pi to the Internet

There are two primary ways to connect your Raspberry Pi to the web--using a wired connection (Ethernet) or using a wireless connection (Wi-Fi preferred).

CONNECTING WITH A WIRED CONNECTION (ETHERNET)

If your Raspberry Pi has an Ethernet port, connecting it to the Internet is as simple as plugging in a cable to the on-board RJ-45 jack on your Pi. The operating system should automatically work with your router to obtain an IP address and connect to the web.

You'll also want to add a blank file named ssh (open a text editor and hit Save as: ssh) to the boot drive of the SD card to allow connecting to your Raspberry Pi remotely.

Note: The Pi Zero does not have an on-board Ethernet port, but can be connected with a Ethernet --> USB adapter.

CONNECTING OVER WI-FI (HEADLESS SETUP)

You can put the Wi-Fi connection information to the SD card before starting the Raspberry Pi if you don't want to connect a monitor and keyboard.

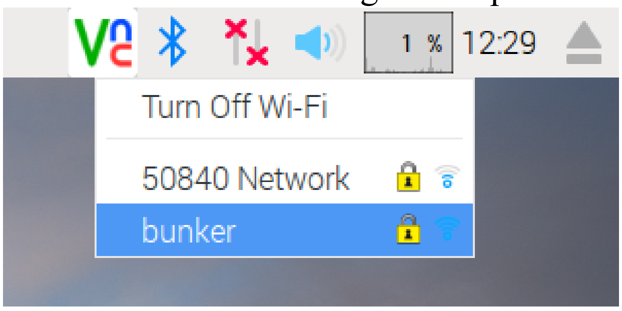
1. Insert the SD card into your computer.
2. Open the drive called boot
3. Add a file named wpa_supplicant.conf with this content:

4. network={

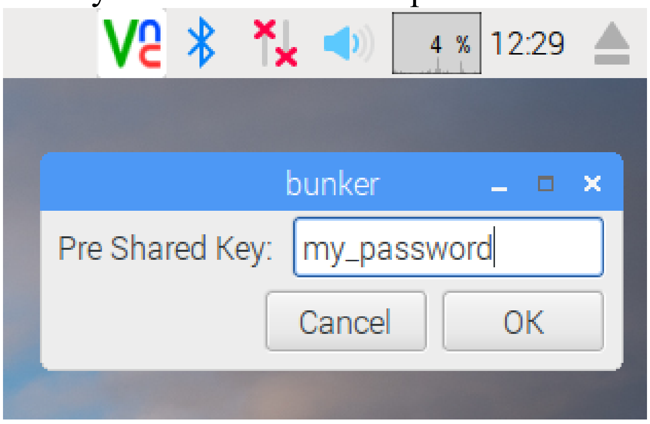
5. ssid="YOUR_SSID"
6. psk="YOUR_PASSWORD"
7. key_mgmt=WPA-PSK
}
8. Add a blank file named ssh (open a text editor and hit Save as: ssh).
9. Insert the SD card into your Raspberry Pi.

CONNECTING OVER WI-FI (GUI SETUP)

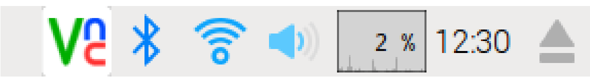
1. Connect a USB keyboard, USB mouse and monitor to your Raspberry Pi.
2. Click on the icon on the left of the volume symbol to scan for Wi-Fi networks and start the Wi-Fi configuration process.



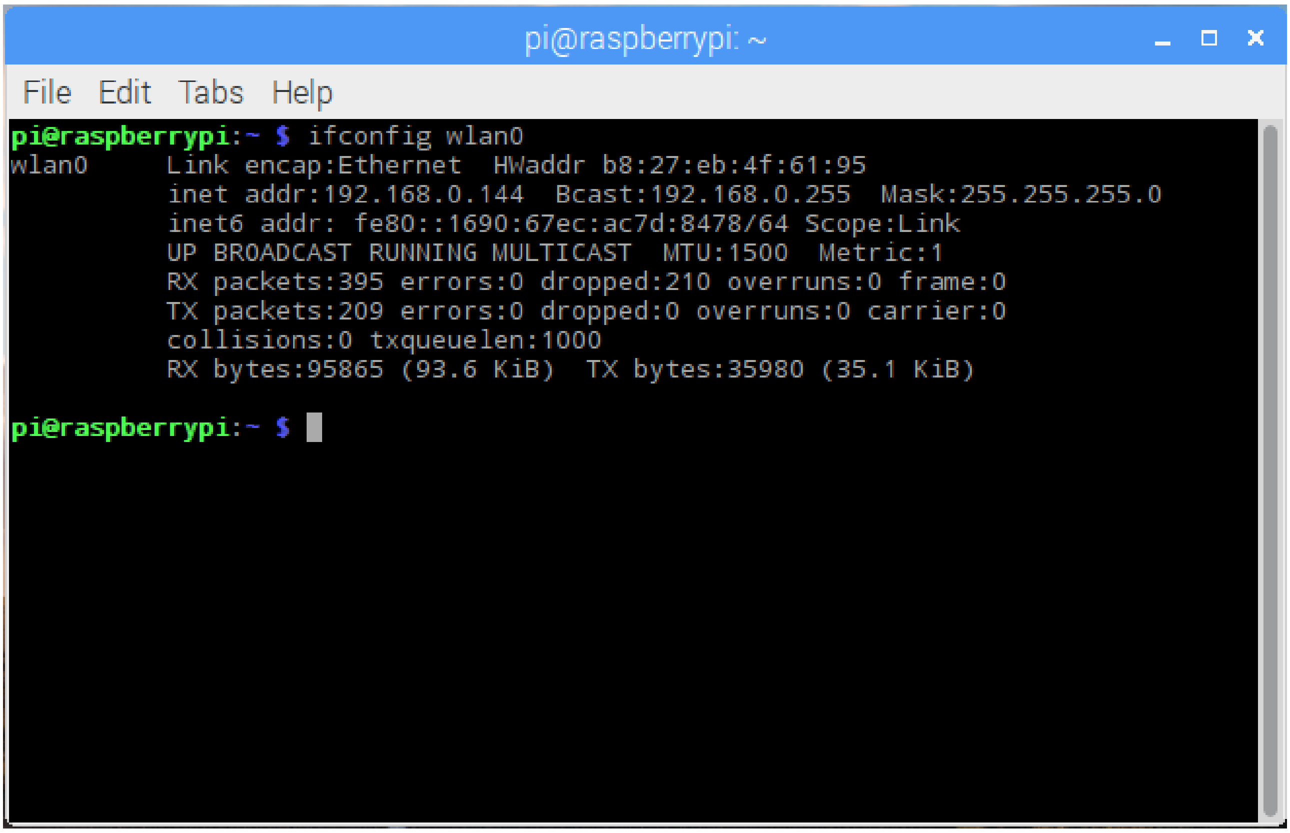
3. Enter your network's Wi-Fi password.



4. When your Pi has successfully connected to the Wi-Fi network, you will see a blue Wi-Fi icon next to the volume icon at the top right hand corner of your screen.



Note that it's also possible to obtain the IP Address of your Raspberry Pi after you've connected it to the Internet. To do so, click on the black terminal icon at the top left hand side of your screen, and type ifconfig wlan0.
Your Pi's IP Address should be displayed next to the label, inet addr and look something like 192.168.X.XXX.



Install Particle Pi software

Now that your Pi is online, it's finally time to download and install the Particle Pi software. If your Pi has a monitor and keyboard connected, you can skip directly to Install the Particle Agent. If would like to install the Particle Pi software without using a monitor and keyboard, please read the following section, Instructions for headless setup.

INSTRUCTIONS FOR HEADLESS SETUP

Note that if you are using a wired connection without a monitor and keyboard (headless) you will have to SSH (secure shell) into your Pi in order to install the Particle software. If you are using a keyboard and monitor, you can head directly to Install the Particle Agent.

- The first step is to obtain the IP address for your Raspberry Pi once it is connected to the Internet. You can find instructions for obtaining your Pi's IP address using Raspberry Pi's official tutorial, here.

The easiest method find your Raspberry Pi's IP address and SSH into it is to use its mDNS hostname followed by .local. The default hostname for your Pi is raspberrypi, so on macOS and Linux, you can simply SSH into your Pi by running the following command in your computer's terminal:

```
ssh pi@raspberrypi.local
```

- If you are using Windows, you can download and use Putty to SSH into your Pi.

The default password for Raspberry Pi is raspberry. **We strongly recommend you change the default password.** You can do so by running the following command inside of your Raspberry Pi's terminal:

```
passwd
```

If you want to change the hostname of your Raspberry Pi to something more meaningful, or if you have multiple Raspberry Pi's on your network, you can do so by running the following command inside of your Pi's terminal:

```
echo newHostname | sudo tee /etc/hostname
```

- You will need to reboot your Pi for the new hostname to be used.

An alternate method for finding the IP address in a headless setup configuration is to ensure that your computer is connected to the same network as your Raspberry Pi device, and to run the following command in your computer's terminal:

```
arp -a | grep b8:27:eb | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'
```

As it turns out, the Raspberry Pi Foundation has their own range of MAC addresses all to themselves. The command above will scan your network for devices whose MAC address starts with the prefix, b8:27:eb and report their IP address. Assuming you only have one Raspberry Pi connected to the network, you should be able to easily identify your Pi's network address and SSH into it in the next step.

- Once you have your Pi's IP address, you can connect to your Pi through a secure shell (SSH). If you are using macOS or Linux, you can simply create an SSH tunnel using your Terminalapplication. If you are using Windows, download PuTTY.

- SSH into your Pi using the following command, where 192.168.X.XXX is the IP address of your Pi.

```
ssh pi@192.168.X.XXX
```

INSTALL THE PARTICLE AGENT

You will not be able to complete this step of the process if you have not already received your beta activation email. If that's the case, hang tight--you'll receive your email in the upcoming weeks as we expand access to the Raspberry Pi provisioning endpoint. For more information, visit particle.io/particle-pi

To connect your Raspberry pi to the Particle Cloud, you need to install the Particle Agent. The Particle Agent is a software service that runs in the background on the Raspberry Pi and allows you to write and run firmware (software that interacts with the GPIO pins on the Pi).

Install the agent by pasting this command in a terminal on your Raspberry Pi, either while connected remotely through SSH or locally with a keyboard and monitor.

```
bash <( curl -sL https://particle.io/install-pi )
```

The installation process should look like this:

When the installation is over, the Particle Agent setup will ask you to sign in to your Particle account. If you don't have one yet, create a Particle account at https://login.particle.io/signup.

Once the Particle Agent is installed, you will have a number of commands available to you to start firmware, stop firmware, and manage your connection to the Cloud. For a full list of available particle-agent commands, type particle-agent help into your terminal.

Particle-agent setup

This is a super useful command! If you find yourself in an unknown or unintended device state at any point in development, you can type this command to reset your device and return it to "factory conditions" running **Tinker**, our default device firmware. The Pi will remain claimed to your Particle account.

When you have successfully completed setup of your Pi, you will see the following confirmation message:

```
Done! Your Raspberry Pi is now connected to the Particle Cloud.

Your Raspberry Pi is running the default Particle app called Tinker.
Tinker allows you to toggle pins with the Particle Mobile App.
  https://docs.particle.io/guide/getting-started/tinker/raspberry-pi/

When you are ready to write your own apps, check out the code examples.
  https://docs.particle.io/guide/getting-started/examples/raspberry-pi/

For more details about the Particle on Raspberry Pi, run:
  sudo particle-agent help
  https://docs.particle.io/reference/particle-agent/
pi@raspberrypi:~ $
```

If you see the message above, **congratulations!** You've successfully connected your Pi to the Particle Cloud!

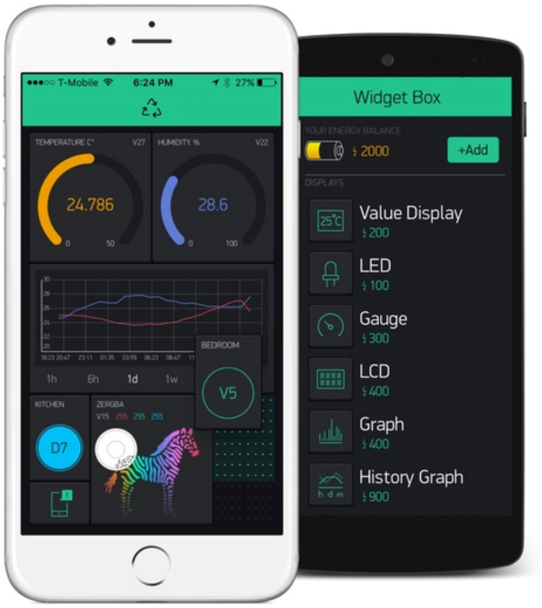
Blynk

Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet.

It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

It's really simple to set everything up and you'll start tinkering in less than 5 mins.

Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you online and ready for the **Internet of Your Things**.



How Blynk Works

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.

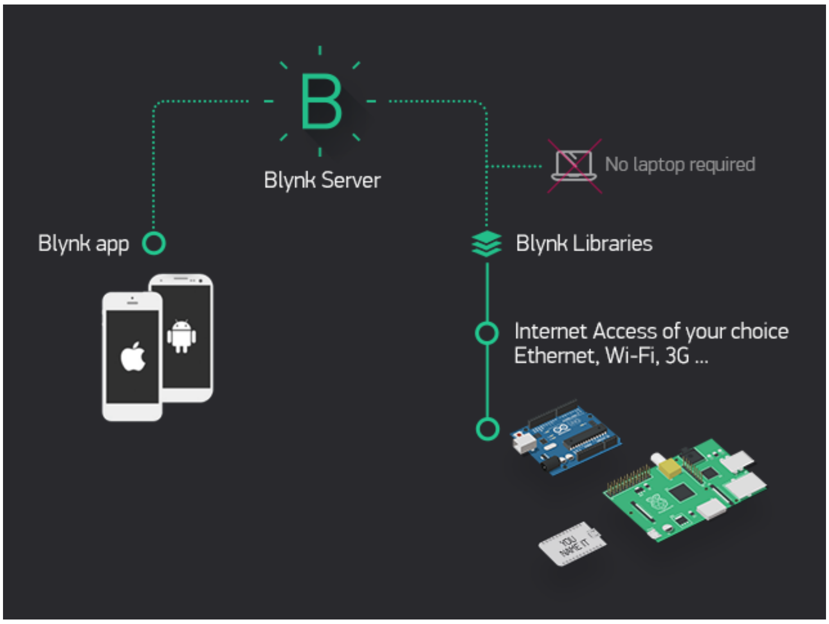
There are three major components in the platform:

Blynk App - allows to you create amazing interfaces for your projects using various widgets we provide.

Blynk Server - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

Now imagine: every time you press a Button in the Blynk app, the message travels to ~~space~~ the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.



Features

- Similar API & UI for all supported hardware & devices
- Connection to the cloud using:
  - Ethernet
  - Wi-Fi
  - Bluetooth and BLE
  - USB (Serial)
  - …
- Set of easy-to-use Widgets
- Direct pin manipulation with no code writing
- Easy to integrate and add new functionality using virtual pins
- History data monitoring via History Graph widget
- Device-to-Device communication using Bridge Widget
- Sending emails, tweets, push notifications, etc.
- … new features are constantly added!

You can find example sketches covering basic Blynk Features. They are included in the library. All the sketches are designed to be easily combined with each other.

What do I need to Blynk?

At this point you might be thinking: “Ok, I want it. What do I need to get started?” **– Just a couple of things, really:**

1. Hardware.

An Arduino, Raspberry Pi, or a similar development kit.

Blynk works over the Internet. This means that the hardware you choose should be able to connect to the internet. Some of the boards, like Arduino Uno will need an Ethernet or Wi-Fi Shield to communicate, others are already Internet-enabled: like the ESP8266, Raspberry Pi with Wi-Fi dongle, Particle Photon or SparkFun Blynk Board. But even if you don’t have a shield, you can connect it over USB to your laptop or desktop (it’s a bit more complicated for newbies, but we got you covered). What’s cool, is that the list of hardware that works with Blynk is huge and will keep on growing.

2. A Smartphone.

The Blynk App is a well-designed interface builder. It works on both iOS and Android, so no holywars here, ok?

Blynk connect with Raspberry Pi

1. Connect your Raspberry Pi to the Internet and open its console.
2. Run this command (it updates your OS package repository to include the required packages):

```
curl -sL "https://deb.nodesource.com/setup_6.x" | sudo -E bash -
```

3. Download and build Blynk JS library using npm:

```
4.    sudo apt-get update && sudo apt-get upgrade
5.    sudo apt-get install build-essential
6.    sudo npm install -g npm
7.    sudo npm install -g onoff
```

```
sudo npm install -g blynk-library
```

8. Run Blynk test script (put your auth token):

```
blynk-client 715f8cafe95f4a91bae319d0376caa8c
```

9. You can write our own script based on examples
10. To enable Blynk auto restart for Pi, find /etc/rc.local file and add there:

```
node full_path_to_your_script.js <Auth Token>
```

Blynk Main Operations

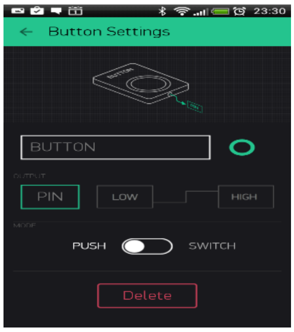
Virtual Pins

Blynk can control Digital and Analog I/O Pins on you hardware directly. You don't even need to write code for it. It's great for blinking LEDs, but often it's just not enough…
We designed Virtual Pins to send **any** data from your microcontroller to the Blynk App and back.
Anything you connect to your hardware will be able to talk to Blynk. With Virtual Pins you can send something from the App, process it on microcontroller and then send it back to the smartphone. You can trigger functions, read I2C devices, convert values, control servo and DC motors etc.
Virtual Pins can be used to interface with external libraries (Servo, LCD and others) and implement custom functionality.
Hardware may send data to the Widgets over the Virtual Pin like this:

```
Blynk.virtualWrite(pin, "abc");
Blynk.virtualWrite(pin, 123);
Blynk.virtualWrite(pin, 12.34);
Blynk.virtualWrite(pin, "hello", 123, 12.34);
```

Button

Works in push or switch modes. Allows to send 0/1 (LOW/HIGH) values. Button sends 1 (HIGH) on press and sends 0 (LOW) on release.



SENSORS USED

GAS Sensors

A GAS sensor or a GAS Detector is a type of chemical sensor which detects/measures the concentration of gas in its vicinity. Gas sensor interacts with a gas to measure in concentration. They are used in various industries ranging from medicine to aerospace. Various technologies are used to measure
Gas concentration such as semiconductors, oxidation, catalytic, infrared, etc. The most common types are as follows

1. Metal Oxide Based GAS Sensor.
2. Capacitance Based GAS Sensor.
3. Acoustic Based GAS Sensor.
4. Calorimetric GAS Sensor.
5. Optical GAS Sensor.
6. Electrochemical GAS Sensor.

Over here we will focus on the most commonly available GAS sensor which is Metal Oxide Gas Sensor or Metal Oxide Semiconductor (MOS) also called as "**Chemiresistors**". The detection is based upon change of resistance of the sensing material when the Gas comes in contact with the material. These Metal Oxide Gas Sensor are extensively used in industry because of their low cost, flexibility in production; simplicity of their use; large number of detectable gases/possible application fields.

Various gas sensors are available in the market but the most commonly available series is the "**MQ Series**". Various gasses like, LPG, Carbon Monoxide (CO), Methane, Smoke, Alcohol, etc can be monitored using these sensors. A good thing about these series is that all are 6 pin sensors with same footprint, same interfacing circuit and are easily available at low cost.

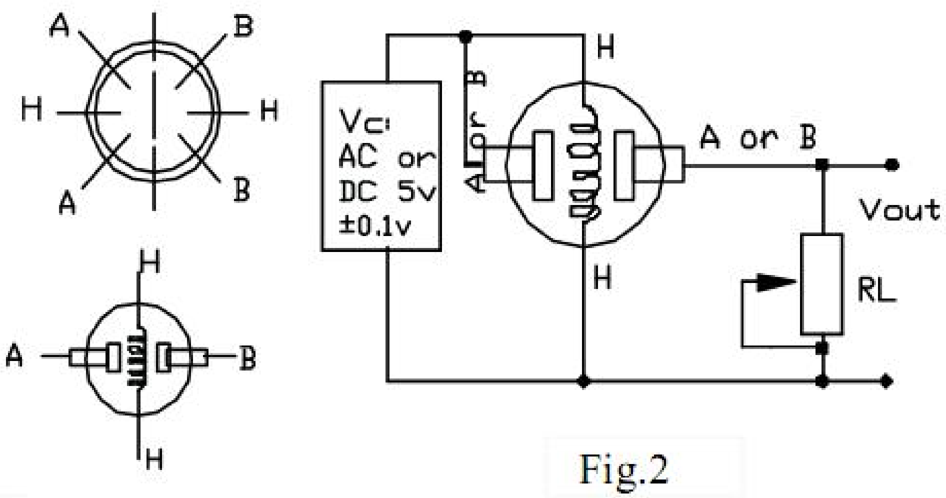


Fig.2

As you can see in the above diagram the MQ series of gas sensors use a small heater inside with an electro-chemical sensor. They are sensitive for a range of gasses.

The preferred wiring is to connect both 'A' pins together and both 'B' pins together. It is safer and it is assumed that is has more reliable output results. Although many schematics and datasheets show otherwise, you are advised to connect both 'A' pins together and connect both 'B' pins together. In the picture, the heater is for +5V and is connected to both 'A' pins. This is only possible if the heater needs a fixed +5V voltage.

The variable resistor in the picture is the load-resistor and it can be used to determine a good value. A fixed resistor for the load-resistor is used in most cases.

The sensor needs a load-resistor at the output to ground. Its value could be from 2kΩ to 47kΩ. Lower the value, the less sensitive. Higher the value less accurate for higher concentrations of gas.

If only one specific gas is measured, the load-resistor can be calibrated by applying a know concentration of that gas. If the sensor is used to measure any gas (like in air quality detector) the load-resistor could be set for a value of about 1V output with clean air.

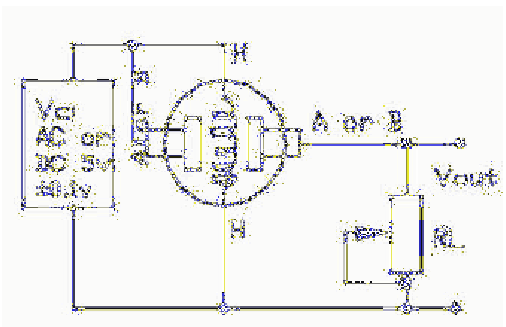
Choosing a good value for the load-resistor is only valid after the burn-in time.

Burn-in

Some datasheets use the term "preheat", but it is the time to burn-in the sensor. This is meant to make the sensor readings more consistent. A time of 12 or 48 hours is usually used for the burn-in time.

The Burn-in is achieved by applying normal power to the sensor (to the heater and with the 'A' and 'B' pins connected, and with a load-resistor). In some special cases a specific burn-in is needed. See the datasheet if the sensor needs such a specific burn-in.

Interfacing Circuit



Since the output of the gas sensors is resistive a resistor has to be connected between the output pin and ground as shown in the adjoining circuit. As you can see no other component is required. There is no specific value for the load resistor.

Its value could be from 2kΩ to 47kΩ. Lower the value, the less sensitive. Higher the value less accurate for higher concentrations of gas. This output voltage can directly be given to any ADC or any comparator circuit and accordingly the gas value can be calculated using a lookup table. These sensors can be easily directly connected to micro controllers with internal ADC or with Arduino.

List of MQ Series GAS Sensors

- MQ2 Gas Sensor

Sensitive for Methane, Butane, LPG, smoke.This sensor is sensitive to flammable and combustible gasses.

- MQ3 Gas Sensor

Sensitive to Alcohol, Ethanol, smoke

- MQ4 Gas Sensor

Sensitive to Methane and CNG/Natural Gas.

- MQ5 Gas Sensor

Sensitive to Natural gas and LPG.

- MQ6 Gas Sensor

Sensitive to LPG, Natural Gas, coal gas and butane

- MQ7 Gas Sensor

Sensitive for Carbon Monoxide

- MQ8 Gas Sensor

Sensitive for Hydrogen Gas

- MQ9 Gas Sensor

Sensitive for Carbon Monoxide, flammable gasses.

- MQ131 Gas Sensor

Sensitive to Ozone

- MQ135 Gas Sensor

For Air Quality. Sensitive to Benzene, Ammonia, Alcohol and smoke.

- MQ136 Gas Sensor

Sensitive to Hydrogen Sulfide gas.

- MQ137 Gas Sensor

Sensitive to Ammonia.

- MQ138 Gas Sensor

Sensitive to Benzene, Toluene, Alcohol, Acetone, Propane, Formaldehyde gas and Hydrogen gas.

Applications

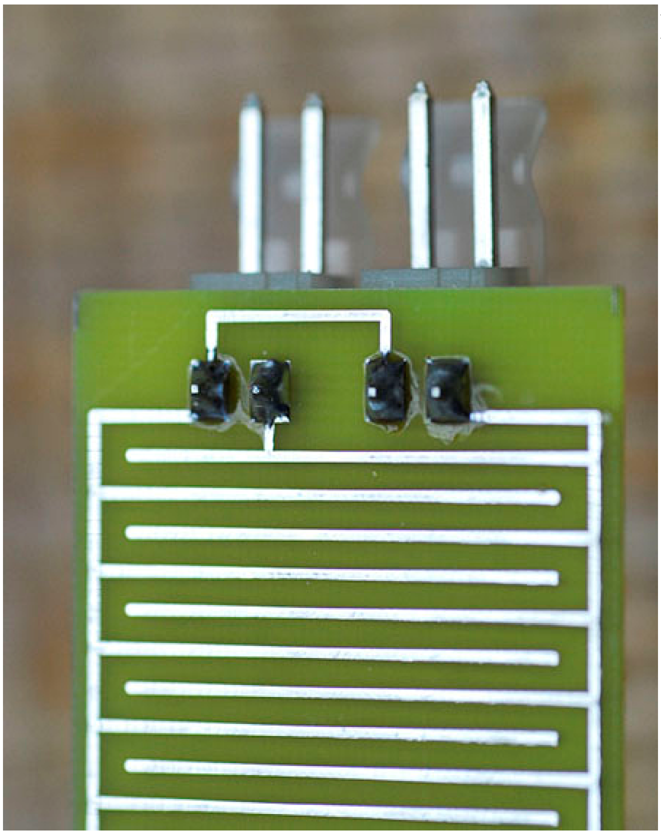
- Process control industries
- Environmental monitoring
- Boiler control
- Fire detection
- Alcohol breath tests
- Detection of harmful gases in mines
- Home safety
- Grading of agro-products like coffee and spices.

Features

- High sensitivity
- Fast response
- Wide detection range
- Stable performance and long life
- Simple drive circuit

WaterLevelSensor

Submersible Pressure Transducer for Level &Pressure



FEATURES:

- High accuracy and reliability
- Completely submersible pressure transducer and cable
- Compact, rugged design for easy installation
- Minimal maintenance and care
- Water level sensor compatible with most monitoring equipment
- 4-20mA output
- Vented cable for automatic barometric compensation
- Multiple level ranges available from 3' to 500'
- Wet-wet water level sensor eliminates vent tube concerns
- Dynamic temperature compensation system
- Submersible pressure transducer not affected by foam, wind, or rain
- Monitors levels in groundwater wells, rivers, streams, tanks, lift stations and open channels
- CE Certified

Water Level Sensor Product Description

Global Water's WL400 Water Level Sensor provides highly accurate water level measurement for a wide variety of applications, including those in severe environments. The submersible pressure transducers have a dynamic temperature compensation system, enabling high accuracy measurements over a wide temperature range. The water level sensor is easily adapted to all dataloggers, telemetry, monitoring equipment, and displays.

Each of the water level sensors consist of a solid state submersible pressure transducer encapsulated in a stainless steel 13/16" diameter housing. The water level sensor has a molded-on waterproof cable and a two-wire 4-20 mA high level output for connection to a monitoring device. A 25' cable is standard, and optional cable lengths are available up to 500'.

The Water Level Sensor's submersible pressure transducer is fully encapsulated with marine-grade epoxy so that moisture can never leak in or work its way down the vent tube to cause drift or level sensor failure (as is the case with other pressure sensors). The water level sensor uses a unique, highly flexible silicon diaphragm to interface between water and the sensing element. This silicon diaphragm protects the water level sensor's electronics from moisture and provides each sensor with exceptional linearity and very low hysteresis. The design of the submersible pressure transducers eliminates the issues associated with metal foil diaphragms, which tend to crinkle and stretch out over time causing drift, linearity, and hysteresis problems. The water level sensor is also has automatically barometric compensation due to the attached vent cable and protected by a stainless steel micro-screen cap, which makes fouling with silt, mud, or sludge virtually impossible. The water level sensor's design is great for all saltwater applications including tide level monitoring, floating docks, and others.

Pressure transducer level ranges of 0-3', 0-15', 0-30', 0-60', 0-120', 0-250', and 0-500' are available. The 0-3' low-level range is ideal for measuring shallow flows or small water level changes like those encountered in sewers, storm drains, weirs, and flumes. The 0-3' water monitoring sensor accurately measures small changes in water, even when the water's depth is only a few inches deep. Other metal foil-type water level sensors typically have serious problems at low-level ranges because of sensor crinkling, stretching, and drifting.

Each submersible pressure transducer has a two-wire 4-20 mA output signal that is linear with water depth. From 10 to 36 VDC is required to operate the water level sensor, so the level sensors can be operated from common 12 VDC battery systems. The 4-20 mA signal can run up to 3,000' from the submersible pressure transducer to the logging device. Common twisted pair or electrical extension cord wire may be spliced to the vented cable once the cable is out of the water. The 4-20 mA signal may be converted to 0.5 to 2.5 VDC by dropping the current signal across a 125 ohm resistor.

Water Level Sensor Specifications

Sensing Element

Sensor Element: Silicone Diaphragm, Wet/Wet Transducer
Pressure Range: 0-3', 0-15', 0-30', 0-60', 0-120', 0-250', 0-500'
Linearity and Hysteresis: ±0.1% FS
Accuracy: ±0.1% of full scale at constant temperature, ±0.2% over 35°F to 70°F range
Overpressure: Not to exceed 2 x full scale range
Resolution: Infinitesimal (Analog)
Outputs: 4-20 mA or 0.5 to 2.5 VDC across 125 ohms
Supply Voltage: 8 to 36 VDC
Current Draw: Same as sensor output
Warm Up Time: 3 seconds recommended
Operating Temperature: -40° to +185°F
Compensation: Uses dynamic temperature compensation 30-70 F. Automatic barometric pressure compensation
Weight: 1/2 lb (227 g)

Housing

Material: 304L Stainless Steel, SS microscreen (hundreds of holes to prevent fouling), electronics are fully encapsulated in marine grade epoxy, guaranteed not to leak
Size: up to 13/16" diameter x 5 1/2" long (2 cm dia. x 14 cm long)(small enough for a 1" well)

Cable

Conductors: 4 each 22 AWG
Jacket Material: Marine Grade Polyurethane

Vent Tube Material: HD Polyethylene
Shield: Aluminum Mylar
Outer Diameter: 0.306"
Temperature Range: -22° to +185°F
Weight: 0.7 oz/ft.
Length: Standard 25' (up to 500' from the factory)

Water Level Sensor Installation

The submersible pressure sensor may be placed slightly below the lowest expected water level. Water level sensor level ranges of 0-3', 0-15', 0-30', 0-60', 0-120', 0-250' and 0-500' are available. When ordering, select the level sensor range that will cover the maximum water level change (this is not necessarily the total depth of water). Selecting the smallest water level range possible will ensure the greatest accuracy. The 4-20 mA signal can run up to 3,000' from the water level sensor to the logging device. Common twisted pair or electrical extension cord wire may be spliced to the vented cable once the cable is out of the water. The 4-20 mA signal may be converted to 0.5 to 2.5 VDC by dropping the current signal across a 125 ohm resistor.

Water Level Sensor Applications

The WL400 Water Level Sensor can be used to measure water level, well depth, groundwater level, surface water flow, pipe flow, and water pressure, etc... NOTE: Polyurethane cable is not recommended for use with hydrocarbons or high concentrations of chlorine. Please look at theWL450 Water Level Transmitter to meet the requirements of these types of applications.

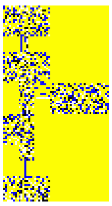
Tactile Bump Sensor Circuit

Tactile Bump Sensors are great for **collision detection**, but the circuit itself also works fine for user buttons and switches as well.
There are many designs possible for bump switches, often depending on the design and goals of the robot itself. But the circuit remains the same. They usually implement a mechanical button to short the circuit, pulling the signal line high or low. An example is the **microswitch** with a lever attached to increase its range, as shown above.

There are several versions below, depending on how you plan to use the circuit and your available switches. For the **resistor** use a very high value, such as **40kohms**.

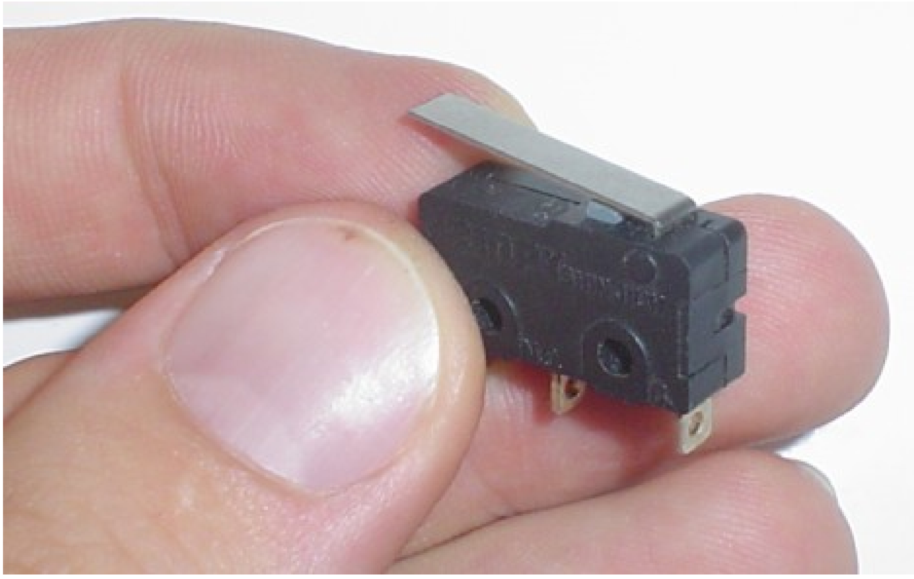
Tactile Bump Sensor Circuits

Voltage goes high with contact

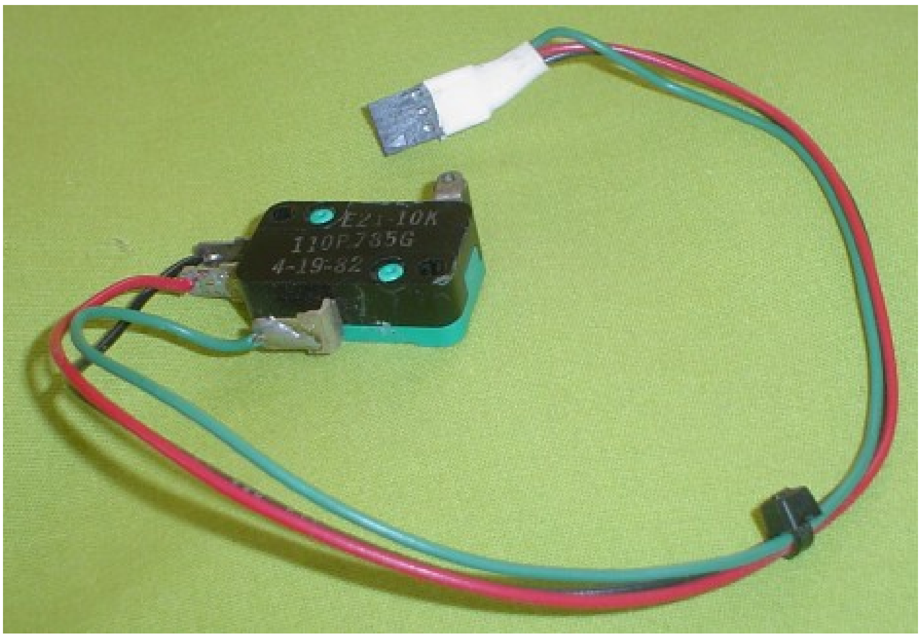


Voltage goes low with contact

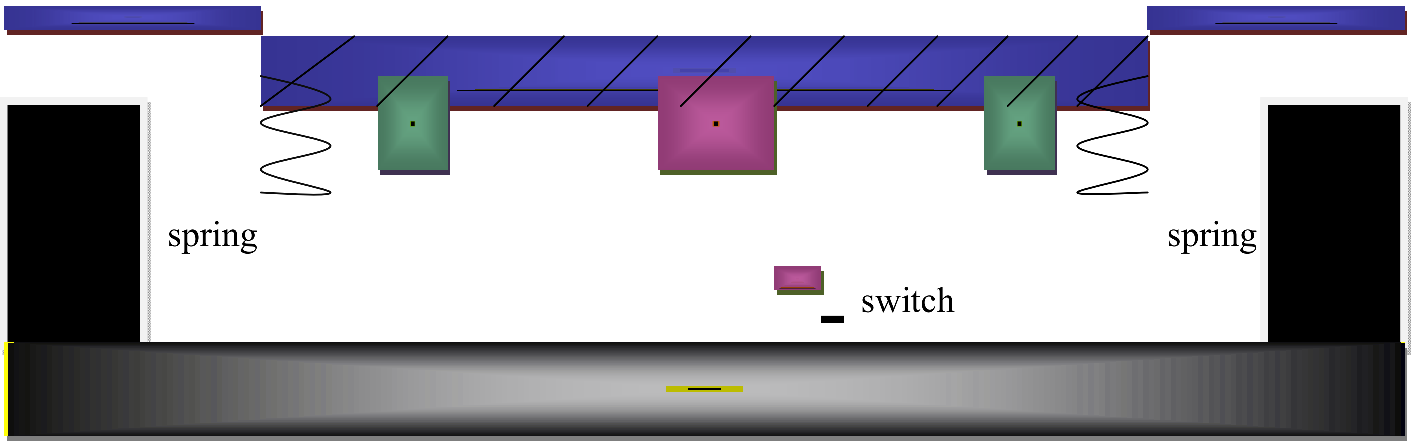
More efficient switch for 3 lead switches
(use for microswitches)



Tips and Uses
Tactile switches only work if your robot can stop instantaneously (like when moving slowly). There is no point ramming the wall, then the switch saying 'oops, wall here.' This is why more advanced robots often use sonar and IR because it gives a slowing down buffer zone. You will need several to cover the front and/or back of your robot.



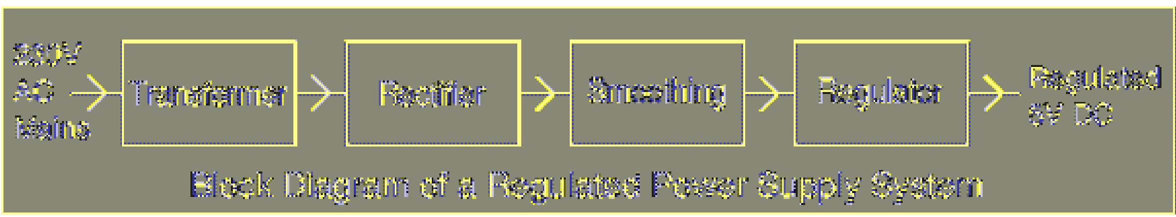
Road level: front view



Power Supply

There are many types of power supply. Most are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function.

For example a 5V regulated supply:



Block Diagram of a Regulated Power Supply System

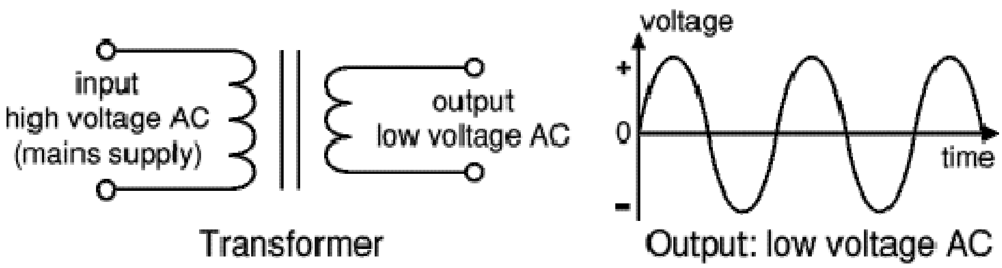
Each of the blocks is described in more detail below:

- Transformer - steps down high voltage AC mains to low voltage AC.
- Rectifier - converts AC to DC, but the DC output is varying.
- Smoothing - smoothes the DC from varying greatly to a small ripple.
- Regulator - eliminates ripple by setting DC output to a fixed voltage.

Power supplies made from these blocks are described below with a circuit diagram and a graph of their output:

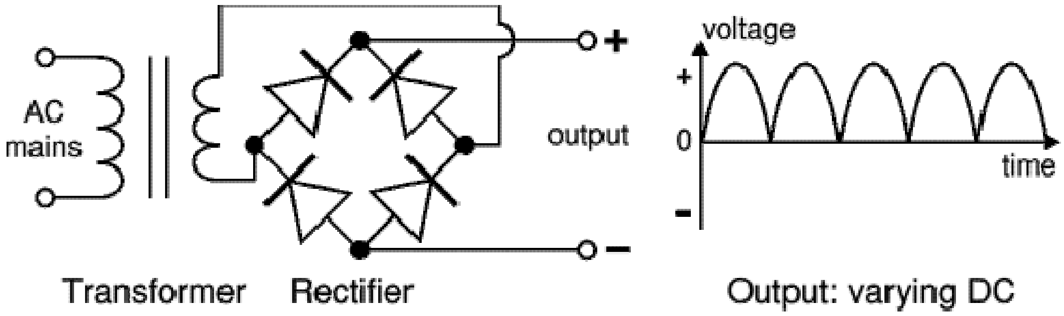
- Transformer only
- Transformer + Rectifier
- Transformer + Rectifier + Smoothing
- Transformer + Rectifier + Smoothing + Regulator

- Transformer only



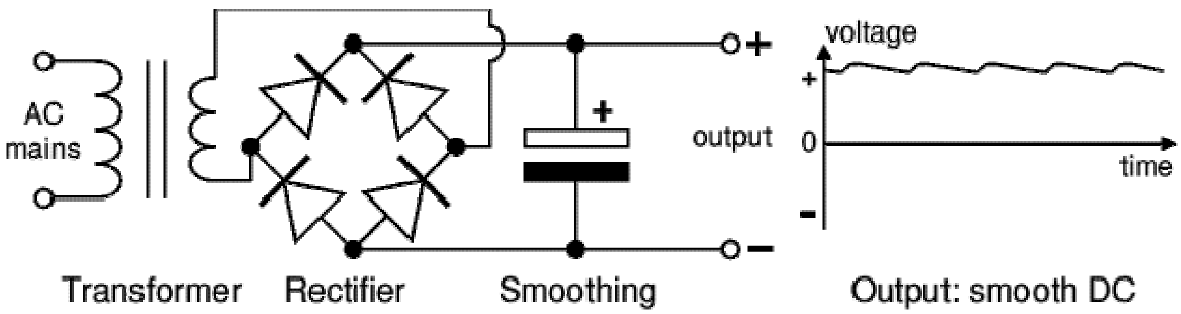
The low voltage AC output is suitable for lamps, heaters and special AC motors. It is not suitable for electronic circuits unless they include a rectifier and a smoothing capacitor.

- Transformer + Rectifier



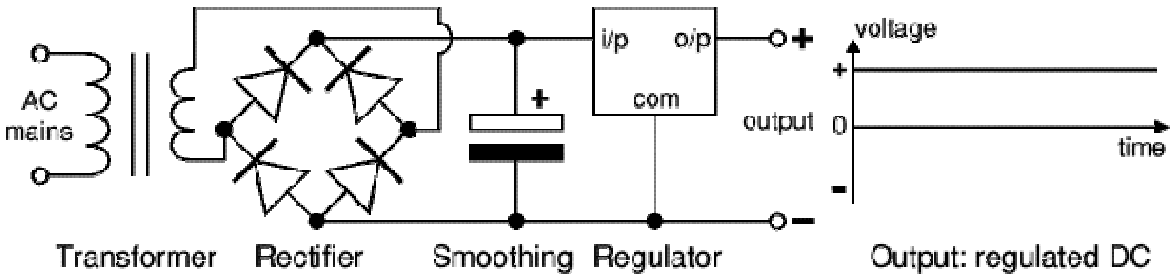
The varying DC output is suitable for lamps, heaters and standard motors. It is not suitable for electronic circuits unless they include a smoothing capacitor.

- Transformer + Rectifier + Smoothing



The smooth DC output has a small ripple. It is suitable for most electronic circuits.

- Transformer + Rectifier + Smoothing + Regulator



The regulated DC output is very smooth with no ripple. It is suitable for all electronic circuits.

The fig. above shows the circuit diagram of the power supply unit. This block mainly consists of a two regulating IC 7805 and a bridge rectified and it provides a regulated supply approximately 5V.

The transformer used in this circuit has secondary rating of 7.5V. The main function of the transformer is to step down the AC voltage available from the main. The main connections are given to its primary winding through a switch connected to a phase line. The transformer provides a 7.5V AC output at its secondary terminals and the maximum current that can be drawn form the transformer is 1 Amp which is well above the required level for the circuit.

The bridge rectified the AC voltage available from the secondary of the transformer, i.e. the bridge rectifier convert the AC power available into DC power but this DC voltage available is not constant. It is a unidirectional voltage with varying amplitude.

To regulate the voltage from the bridge rectifier, capacitors are connected. Capacitors C1 filter the output voltage of the rectifier but their output is not regulated and hence 7805 is connected which is specially designed for this purpose.

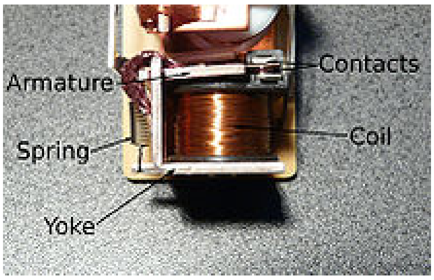
Although voltage regulators can be designed using op-amps, it is quicker and easier to use IC voltage regulator. Further more, IC voltage regulators are available with features such as programmable output current/ voltage boosting, internal short circuit current limiting, thermal shut down and floating operation for high voltage applications.

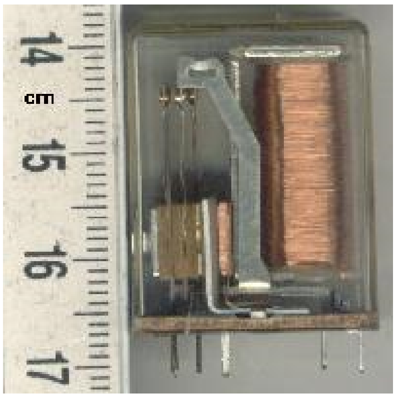
The 78 XX series consists of three terminals viz, input, output & ground. This is a group of fixed positive voltage regulator to give and output voltage ranging form 5V to 24V. These IC's are designed as fixed voltage regulators and with adequate heat sinking, can delivery output current in excess of 1 Amp although these devices do not require external components and such components can be used to obtain adjustable voltage and current limiting. In addition, the difference between the input and output voltages (V in Vo) called the dropout voltage must be typically 2V even from a power supply filter. Capacitors C2, C3, C4, and C5 are small filters which are used for extra filtering.LED1& LED2 are used for Power ON indicator for IC1 and IC2, current-limiting resistors R2&R4, which prevents the LED's from getting heated and thus damaged.

RELAY

A **relay** is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism, but other operating principles are also used. Relays find applications where it is necessary to control a circuit by a low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays found extensive use in telephone exchanges and early computers to perform logical operations. A type of relay that can handle the high power required to directly drive an electric motor is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protection relays

Basic design and operation





A simple electromagnetic relay consists of a coil of wire surrounding a soft iron core, an iron yoke, which provides a low reluctance path for magnetic flux, a movable iron armature, and a set, or sets, of contacts; two in the relay pictured. The armature is hinged to the yoke and mechanically linked to a moving contact or contacts. It is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB.

When an electric current is passed through the coil, the resulting magnetic field attracts the armature, and the consequent movement of the movable contact or contacts either makes or breaks a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low voltage application, this is to reduce noise. In a high voltage or high current application, this is to reduce arcing.

When the coil is energized with direct current, a diode is often placed across the coil to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a voltage spike dangerous to circuit components. Some automotive relays already include a diode inside the relay case. Alternatively a contact protection network, consisting of a capacitor and resistor in series, may absorb the surge. If the coil is designed to be energized with alternating current (AC), a small copper ring can be crimped to the end of the solenoid. This "shading ring" creates a small out-of-phase current, which increases the minimum pull on the armature during the AC cycle.

FUTURE SCOPE

Further extensions and feature enhancements are always inevitable in the present generation trending technologies. As the future scope this system can be extended further by adding additional infrared emitting system to detect the people face if they wore the mask on his/her face. By adding this additional system we can easily identify the person even though the person covered his/her face. Apart from this we can interface sensors like Gas sensors, Smoke sensors, and Fire sensors to give alerts respectively.

CONCLUSION

Thedesign and implementation of a low cost but secure home security system for general users.The security level is increased due to the usage of Raspberry pi which sends the information to the user, has in built capabilities and is easily connectible to external devices. Raspberry pi proves to be smart economic and efficient platform for implementing the home security system. Two advantages provided by the system is that, Necessary action can be taken in short span of time in the case of emergency condition and design of a PCBboard which is also small in size. Reduced size makes it more applicable for commercial manufacturing and distribution. A raspberry pi and open source applications with its ever growing community and development provides a great hope in the near future.

REFRENCE

http://www.blynk.cc/getting-started/

https://docs.particle.io/guide/getting-started/start/raspberry-pi/

https://www.particle.io/

https://www.particle.io/what-is-iot

https://en.wikipedia.org/wiki/Internet_of_things

https://www.slideshare.net/KrishnaKumar272/smart-wireless-surveillance-monitoring-using-raspberry-pi

https://image.slidesharecdn.com/fiinalppt-150305045607-conversion-gate01/95/smart-wireless-surveillance-monitoring-using-raspberry-pi-33-638.jpg?cb=1425540758