CHAPTER-1

Introduction

Snake robots are a class of hyper-redundant mechanisms that locomote through internal shape changes. Snake robots' unique shape and ability to navigate highly variable environments, such as the pipe in Figure 1, make them suitable for urban search and rescue missions. There are also various military uses. However, only strong, well-designed, and reliable robots will be useful for these tasks. We have designed a modular architecture that allows for both a wide range of gaits and resilience to failures. This architecture incorporates a well-refined mechanical design coupled with advanced electronics and software. These design requirements have led to the development of the versatile Super Servo which proactively preserves itself as it monitors its internal status. This combination allows the robots to effectively perform their tasks. We considered common failures in other modular snake robots and designed accordingly.

1.1 ABSTRACT:

SNAKE BOT

This Project is to design a snake-like robot that can provide the locomotion as the real biological snake, and possessed the ability to cross over the obstacles with a certain height's limit, or find another alternative ways instead of climb over it if the height of the obstacles is over the limit. Snake-like robot is a biomorphic hyper redundant robot that resembles a snake. The shape and sizes of the snake-like robot is depend on its own application, different application may require different sizes and shapes, since this project mainly target is to design a snake-like robot that can avoid the obstacles, so the snake-like robot is design to a moderate size with 4-5 segments, so that the snake-like robot can move flexible in the terrain that have a lot of obstacle. In order to make the snake-like robot function and move like a real biological snake, snake-like robot may construct of multiple joints which enable the snake-like robot to have multiple degree of freedom, which give it the ability to flex, reach and approach a huge volume in its workspace with infinite number of configurations.

This mobility can enable the robot to move around in more complex environments.
So, the application for this snake-like robot could be very useful in hard to reach places or hazardous environments. For the locomotion of this snake-like robot, it is move in a specific gait, which is a periodic of sine wave motion, just like a lateral undulation motion. Lastly, the special feature such as snake-like locomotion, ability to climb over obstacles or stair, estimate the height, making decisions, and able to remotely control are applied to the design.

1.2 WORKING PRINCIPLE:

This Snake Bot is working under Raspberry pi concept that it is controlled by our Laptop using Linux OS. This Snake Bot will hold a camera. Raspberry pi is the heart of this project. 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. It has micro USB port for power, programming and debugging.

Existing System:

1.3 BLOCK DIAGRAM:

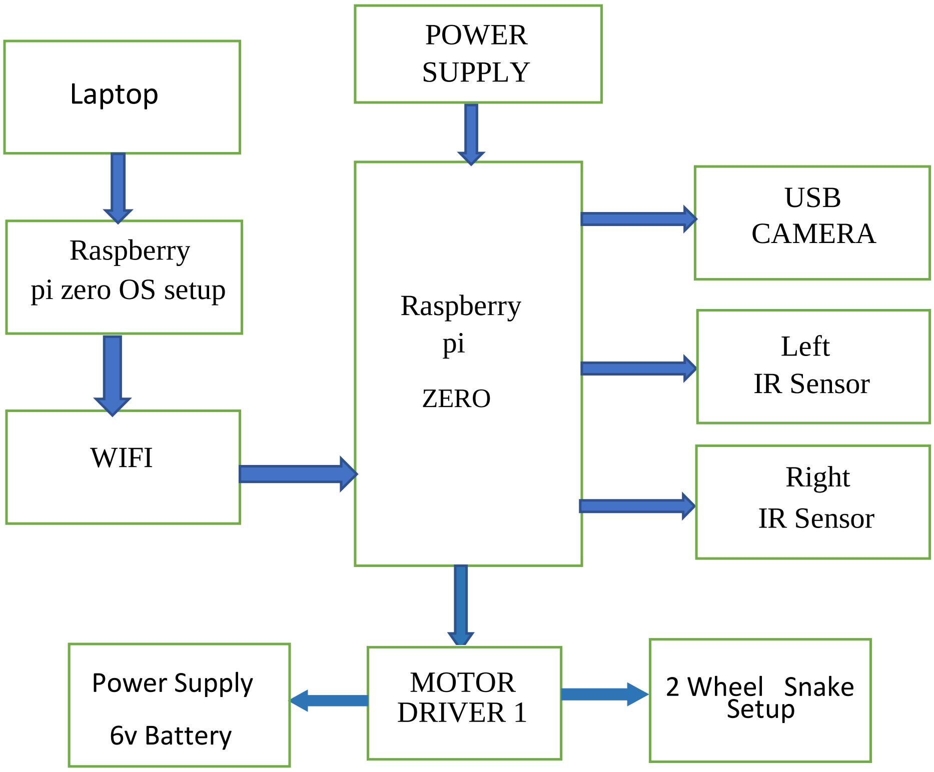


FIG1.1

Proposed System:

BLOCK DIAGRAM:

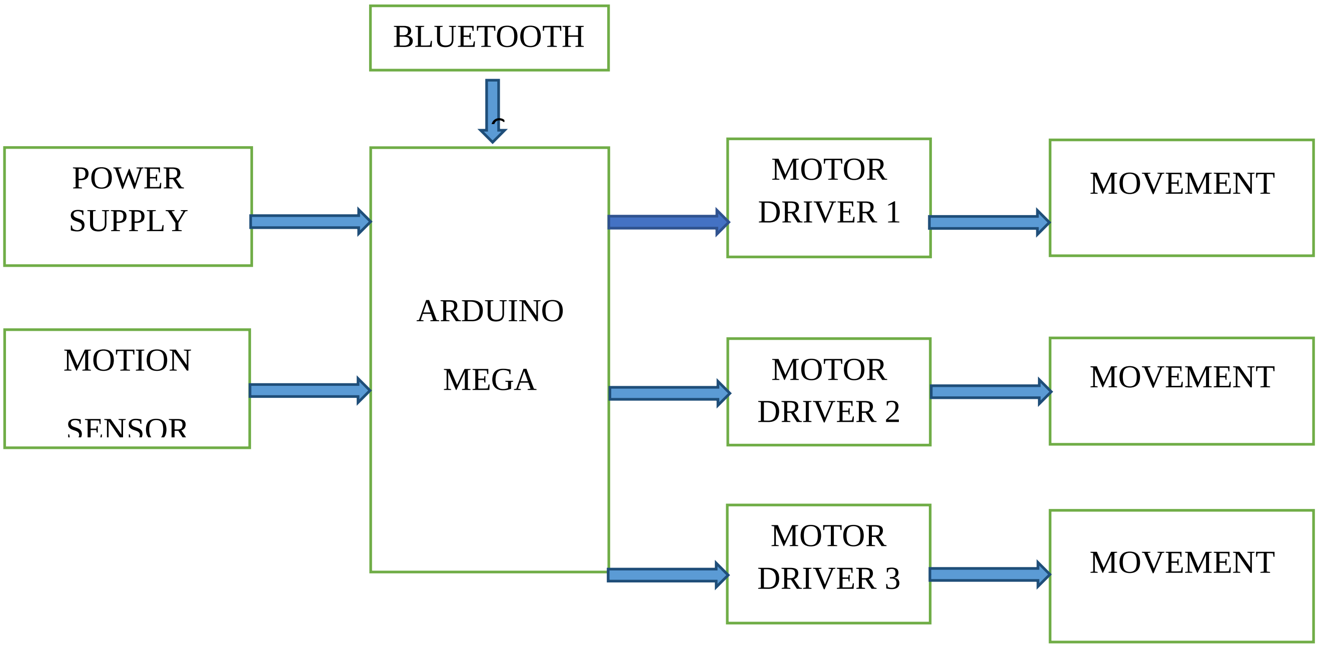


FIG1.2

CHAPTER-2

2.HARDWARE DESCRIPTION:

- Raspberry Pi Zero
- Raspberry pi camera
- IR Sensor
- L293D motor driver module
- 6V lithium ion Battery
- DC Geared Motor

2.SOFTWARE DESCRIPTION:

Language	:	Python
Compiler	:	GCC Complier.
OS	:	Linux

2.1 Power Supply:

Battery is the primary power source for any electronics wireless gadget, be it a smartphone, laptop, watch or remote. Can you imagine the situation without these energy sources? We wouldn’t be able to build any wireless electronic device and have to rely on wired power source only, even electric cars and space missions would not be possible without Batteries. Today in this tutorial we discuss briefly about various types of batteries, their classification, terminology and specifications.

What is Battery and why it is used?

Let’s see the basic **difference between a battery and a cell**. Also let’s find out why we exactly need a battery and why can’t we use the Alternating power (i.e., AC power from the wall sockets) instead of DC power.

Cell: A cell is an energy source which can deliver only DC voltage and current which are in very small quantities. For example if we take cells that we use in watches or remote controls, it can give maximum of 1.5 – 3V.

Battery: The functionality of the battery is exactly same as that of a cell but a **battery is a pack of cells arranged is a series/parallel fashion** so that the voltage can be raised to desired levels. The best known example for a battery is a power bank which is used to charge up smart phones. If we ever see the inside of a power bank we can find set of batteries arranged serially/parallel based on the requirement. Batteries are arranged in series to increase the voltage and in parallel to increase the current.

Now **Why DC is preferred over AC**? In most of the portable electronics, AC can't be stored where as DC can be stored without any difficulty. Even the losses due to AC power are more when compared to the DC power. That is the reason DC is preferred for portable electronic devices.

2.2Technical terms used while dealing with batteries

We can't just keep on using voltage and current alone to explain about a battery's functionality, there are some unique terms that defines the characteristics of a battery like Watt-hour (mAh), C-rating, nominal voltage, charging voltage, charging current, discharging current, cut off voltage, shelf life, cycle life are the few terms used to define a batteries performance.

Power capacity:

It is the **energy stored in a battery** which is measured in **Watt-hour**

Watt-hour = V * I * hours {since voltage is kept constant, so it is measured in Ah/mAh}

Power capability:

It means the **amount of current that the battery can deliver**. It is also known as **C-rating.**

Nominal voltage:

While defining power capacity we have a unit called **Wh** which can be elaborated as V * I * hour but where did the voltage gone? As the voltage of the battery will be constant and will not be varied, it is considered as **nominal voltage** (fixed voltage). So since the voltage is fixed only Ampere and hour are considered as the unit (**Ah/mAh)**.

Charging current:

It is the **maximum current that can be applied to charge the battery** i.e., practically maximum of 1A/2A can be applied if a battery protecting circuit is in-built but still 500 mA is the best the range for charging the battery.

Charging voltage:

It is the maximum voltage that should be applied to the battery to efficiently charge a battery. Basically 4.2 V is the best/standard charging voltage. Though we apply 5 V to the battery it accepts only 4.2 V.

Discharging current:

It is the **current that can be drawn from the battery and is delivered to the load**. If the current drawn by the load is greater than the rated discharging current, the battery drains very fast which causes the battery heat up quickly which also causes the battery to explode. So it is cautious to determine the amount of current drawn by the load as well as the maximum discharging current a battery can withhold.

Shelf life:

There might be a situation where the batteries are kept idle/sealed especially in the stores/shops for a long period of time. *So* **shelf life defines the time period a battery can be stay powered up and should be able to use it for a rated time period**. Shelf life is mainly considered for non-rechargeable batteries because those are of use and throw. For rechargeable batteries even if the shelf time is less, we can still recharge it.

Cut-off voltage:

It is the **voltage at which the battery can be considered as fully discharged**, after which if we still try to discharge from it the battery gets damaged. So beyond the cut-off voltage the battery should be disconnected from the circuit and should be charged appropriately.

Cycle life:

Let's consider a battery is fully charged and it is discharged to 80% of its actual capacity, then the battery is said to be completed one cycle. Likewise the **number of such cycles that a battery can charge and discharge defines the cycle life**. The more the cycle life the better will be the battery's quality. But if a battery is discharged to say 40% of its actual capacity considering the battery is fully charged initially, it cannot be considered as a cycle life.

Power density:

It defines power capacity of battery for a given mass of volume.

Types of Batteries

Batteries are basically classified into 2 types:

- Non-rechargeable batteries (primary batteries)
- Rechargeable batteries (secondary batteries.

Non-rechargeable Batteries

These are basically considered as **primary batteries** because they can be used only once. These batteries cannot be recharged and used again. Let's see about the regular, daily life primary batteries that we see.

- **Alkaline batteries:** It is basically constructed with the chemical composition of Zinc (Zn) and Manganese dioxide (MnO_2), as the electrolyte used in it is potassium hydroxide which is purely an alkaline substance the battery is named as alkaline battery having he power density of 100 Wh/Kg.

Advantages:

1. Cycle life is more
2. More compatible and efficient for powering up portable devices.
3. Shelf life is more.
4. Small in size.
5. Highly efficient.
6. Low internal resistance so that discharge state in idle state is less.
7. Leakage is low.

Disadvantages:

1. Cost is a bit high. Except it everything is an advantage.

Applications:

It can used in torches, remotes, wall clocks, small portable gadgets etc.

- **Coin cell batteries:** The chemical composition of coil cell batteries is also alkaline in nature. Apart from alkaline composition, lithium and silver oxide chemicals will be used to manufacture these batteries which are more efficient in providing steady and stable voltage in such a small sizes. It has Power density of 270 Wh/Kg.

Advantages:

1. Light in weight
2. Small in size
3. High density
4. Low cost
5. High nominal voltage (up to 3V)
6. Easy to get high voltages by arranging serially
7. Long shelf life

Disadvantages:

1. Needs a holder
2. Low current draw capability

Applications:

Used in watches, wall clocks, miniature electronic products etc.

vRechargeable Batteries

These are generally called as **secondary batteries** which can be recharged and can be reused. Though the cost is high, but they can be recharged and reused and can have a huge life span when properly used and safely charged.

Lead-acid batteries

It consists of lead-acid which is very cheap and seen mostly in cars and vehicles to power the lighting systems in it. These are more preferable in the products where the size/space and weight doesn’t matter. These comes with the nominal voltage starting 2V to24V and most commonly seen as 2V, 6V, 12V and 24V batteries. It has Power density of 7 Wh/Kg.

Advantages:

1. Cheap in cost
2. Easily rechargeable
3. High power output capability

Disadvantages:

1. Very heavy
2. Occupies much space
3. Power density is very low

Applications:

Used in cars, UPS (uninterrupted Power Supply), robotics, heavy machinery etc..

Ni-Cd batteries

These batteries are made of Nickel and Cadmium chemical composition. Though these are very rarely used, these are very cheap and their discharge rate is very low when compared to NiMH batteries. These are available in all standard sizes like AA, AAA, C and rectangular shapes. The nominal voltage is 1.2V, often connected together in a set of 3 which gives 3.6V. It has Power density of 60 Wh/Kg.

Advantages:

1. Cheap in cost
2. Easy to recharge

3. Can be used in all environments
4. Comes in all standard sizes

Disadvantages:

1. Lower power density
2. Contains toxic metal
3. Needs to be charged very frequently in order to avoid growth of crystals on the battery plate.

Applications:

Used in RC toys, cordless phones, solar lights and mostly in the applications where price is important.

Ni-MH batteries

The Nickel – Metal Hydride batteries are much preferable than Ni-Cad batteries because of their lower environmental impact. Its nominal voltage is 1.25 V which is greater than Ni-Cad batteries. It has less nominal voltage than alkaline batteries and they are good replacement due to its availability and less environmental impact. The power density of Ni-MH batteries is 100 Wh/Kg.

⊕ KAYO Ni-MH AA 1600mAh 1.2V ⊖
⊕ KAYO Ni-MH AA 1600mAh 1.2V ⊖

Advantages:

1. Available in all standard sizes.
2. High power density.
3. Easy to recharge.
4. A good alternative to alkaline which has almost all similarities and also it is rechargeable.

Disadvantages:

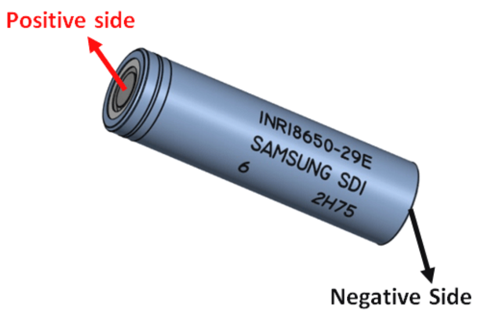
1. Self-discharge is very high.
2. Expensive than Ni-Cad batteries.

Applications:

Used in all applications similar to the alkaline and Ni-Cad batteries.

Li-ion batteries

These are made up of Lithium metal and are latest in rechargeable technology. As these are compact in size they can be used in most of the portable applications which need high power specifications. These are the best rechargeable batteries available. These have a nominal voltage of 3.7V (most commonly we have 3.6V and 7.2V) and have various ranges of power capacity (starting from 100s of mAh to 1000s of mAh). Even the C-rating ranges from 1C to 10C and Power density of Li-ion batteries is 126 Wh/Kg.



Parameter	**Lead Acid**	**Ni-Cd**	**Ni-MH**	**Alkaline**	***Li-Ion***	**Li-Polymer**
Cell Voltage	2V	1.2V	1.2V	1.5V	3.6V	3.7V
Cost	Low	Moderate	High	Very Low	Very High	Very High
Internal Resistance (IR)	Low	Very Low	Moderate	Varies	High	Low
Self Discharge (%/month)	2% to 4%	15% to 30%	18% to 20%	0.3%	6% to 10%	5%
Charge Cycle	500 to 2000	500 to 1000	500 to 800	Low	1000 to 1200	>1000
Overcharge Tolerance	High	Medium	Low	Medium	Very Low	Very Low
Energy Density (Wh/kg)	30 to 45	45 to 50	55 to 65	80	90 to 110	130 to 200
Memory Effect	No	Yes	Yes	Yes	No	No
Maintenance	Very High	High	Low	Low	Low	Low
Safety	Highly Safe	Safe	Safe	Safe	Un-Safe	Un-Safe

18650 Cell Features and Technical Specifications

- Nominal Voltage: 3.6V
- Nominal Capacity: 2,850 mAh
- Minimum Discharge Voltage: 3V
- Maximum Discharge current: 1C
- Charging Voltage: 4.2V (maximum)
- Charging current: 0.5C
- Charging Time: 3 hours (approx)
- Charging Method: CC and CV
- Cell Weight: 48g (approx)
- Cell Dimension: 18.4mm (dia) and 65mm (height)

Where to use an 18650 Cell

The 1**8650 Cell** is a **Li-ion type battery** which has found its application in many fields such as Portable electronics like torch lights, Electric Vehicles/Cars like Tesla and much more. The main reason for this battery being successful is its properties compared to its competitors. These properties include current carrying capability, voltage, cycle life, storage life, safety, and operating temperature and much more. Below table shows the comparison between popular batteries for key parameters.

Lead Acid vs Ni-Cd vs Ni-MH vs Alkaline vs Li-ion vs Li-Polymer Batteries

Advantages:

1. Very light in weight.
2. High C-rating.
3. Power density is very high.

4. Cell voltage is high.

Disadvantages:

1. These are a bit expensive.
2. If the terminals are short circuited the battery might explode.
3. Battery protection circuit is needed.

Li-Po batteries

These are also called as Lithium Ion polymer rechargeable batteries because it uses high conductivity polymer gel/polymers electrolyte instead of liquid electrolyte. These come under the Li-ion technology. These are a bit costly. But the battery is very highly protected when compared to the Li-ion batteries. It has Power density of 185 Wh/Kg.

Advantages:

1. These are highly protective compared to Li-ion batteries.
2. Very light in weight
3. Thin in structure when compared to Li-ion batteries.
4. Power density, nominal voltages are comparatively very high compared to Ni-Cad and Ni-MH batteries.

Disadvantages:

1. Expensive.

2. Might explode if wrongly connected.

3. Should not be bent or exposed to high temperature which may cause to explosion.

Applications: Can be used in all the portable devices which need rechargeable advantage like drones, robotics, RC toys etc.

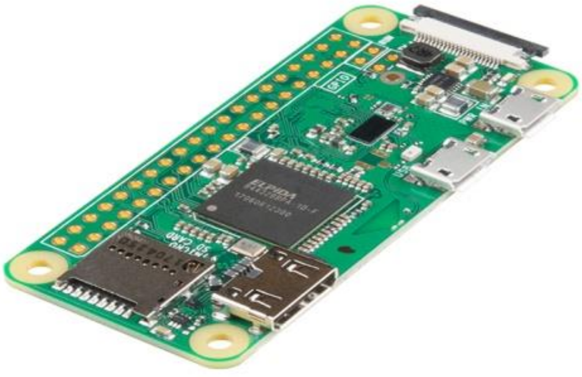
2.3 RASPBERRY PI MICRO CONTROLLER

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.[5][6][7] The original model became far more popular than anticipated,[8] selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) and cases. However, some accessories have been included in several official and unofficial bundles.[8]

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

According to the Raspberry Pi Foundation, more than 5 million Raspberry Pis were sold by February 2015, making it the best-selling British computer.[9] By November 2016 they had sold 11 million units,[10][11] and 12.5m by March 2017, making it the third best-selling "general purpose computer".[12] In July 2017, sales reached nearly 15 million.[13] In March 2018, sales reached 19 million



Generations of released model:

Several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+; on-board memory ranges from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either SDHC or MicroSDHC sizes. The boards have one to four USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm phono jack for audio output. 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. Prices range from US$5 to $35.

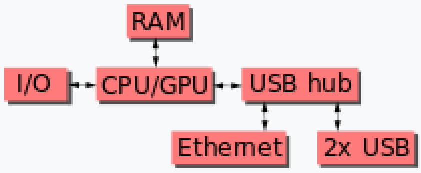
The first generation (Raspberry Pi 1 Model B) was released in February 2012, followed by the simpler and cheaper Model A. In 2014, the Foundation released a board with an improved design, Raspberry Pi 1 Model B+. These boards are approximately credit-card sized and represent the standard *mainline* form-factor. Improved A+ and B+ models were released a year later. A "Compute Module" was released in April 2014 for embedded applications. The Raspberry Pi 2, which added more RAM, was released in February 2015.

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. By 2017, it became the newest mainline Raspberry Pi. On 28 February 2017, the Raspberry Pi Zero W was launched, a version of the Zero with Wi-Fi and Bluetooth capabilities, for US$10.[16][17] On 12 January 2018, the Raspberry Pi Zero WH was launched, the same version of the Zero W with pre-soldered GPIO headers.[18]

Raspberry Pi 3 Model B was released in February 2016 with a 64 bit quad core processor, on-board WiFi, Bluetooth and USB boot capabilities.[19] On Pi Day 2018 model 3B+ appeared with a faster 1.4 GHz processor and a three times faster network based on gigabit Ethernet (300 Mbit / s) or 2.4 / 5 GHz dual-band Wi-Fi (100 Mbit / s).[1] Other options are: Power over Ethernet (PoE), USB boot and network boot (an SD card is no longer required). This allows the use of the Pi in hard-to-reach places (possibly without electricity).

Hardware:

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



This block diagram describes Model B and B+; Model A, A+, and the Pi Zero are similar, but 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 2B uses a 32-bit 900 MHz quad-core ARM Cortex-A7processor.

The Broadcom BCM2835 SoC used in the first generation Raspberry Pi[20] includes a 700 MHz ARM1176JZF-S processor, VideoCore IV graphics processing unit(GPU),[21] 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 stackedunderneath the RAM chip, so only its edge is visible. The 1176JZ(F)-S is the same CPU used in the original iPhone,[22] although at a higher clock rate, and mated with a much faster GPU.

The earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache.[23] The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor,[24] the same SoC which is used on the Raspberry Pi 3, but underclocked (by default) to the same 900 MHz CPU clock speed as the V1.1. The BCM2836 SoC is no longer in production (as of late 2016).

The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.[1]

Performance:

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS.[25][26] 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.

Raspberry Pi 2 V1.1 included a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It was described as 4–6 times more powerful than its predecessor. The GPU was identical to the original.[23] In parallelised benchmarks, the Raspberry Pi 2 V1.1 could be up to 14 times faster than a Raspberry Pi 1 Model B+.[27]

The Raspberry Pi 3, with a quad-core ARM Cortex-A53 processor, is described as having ten times the performance of a Raspberry Pi 1.[28] 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 parallelised tasks.[29]

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.[30] In those cases the Pi automatically shuts the overclocking down if the chip temperature reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and overclocking settings (voiding the warranty); an appropriately sized heat sink 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 maximise the performance of the SoC without impairing the lifetime of the board. 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 seven overclock presets are:

- none; 700 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 overvolting
- modest; 800 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 overvolting,
- medium; 900 MHz ARM, 250 MHz core, 450 MHz SDRAM, 2 overvolting,
- high; 950 MHz ARM, 250 MHz core, 450 MHz SDRAM, 6 overvolting,
- turbo; 1000 MHz ARM, 500 MHz core, 600 MHz SDRAM, 6 overvolting,
- Pi 2; 1000 MHz ARM, 500 MHz core, 500 MHz SDRAM, 2 overvolting,
- Pi 3; 1100 MHz ARM, 550 MHz core, 500 MHz SDRAM, 6 overvolting. In system information the CPU speed will appear as 1200 MHz. When idling, speed lowers to 600 MHz.[30][31]

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.

The CPU on the first and second generation Raspberry Pi board did not require cooling, such as a heat sink or fan, even when overclocked, but the Raspberry Pi 3 may generate more heat when overclocked.[32]

RAM:

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU.[33] 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).[34] Comparatively the Nokia 701 uses 128 MB for the Broadcom VideoCore IV.[35]

For the later Model B with 512 MB RAM new standard memory split files (arm256_start.elf, arm384_start.elf, arm496_start.elf) were initially released for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM) respectively. 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 MB and 512 MB Raspberry Pis.[36]

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM.[37][38] 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.[39] The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4 GHz WiFi 802.11n (150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on the Broadcom BCM43438 FullMAC chip with no official support for monitor mode but implemented through unofficial firmware patching[40] and the Pi 3 also has a 10/100 Mbit/s Ethernet port. The Raspberry Pi 3B+ features dual-band IEEE 802.11b/g/n/ac WiFi, Bluetooth 4.2, and Gigabit Ethernet (limited to approximately 300 Mbit/s by the USB 2.0 bus between it and the SoC).

Peripherals:

The Model 2B boards incorporate four USB ports for connecting peripherals.

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse.[41] 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.[42]

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 generate standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions as well as older NTSC or PAL standard CRT TV resolutions. As shipped (i.e., without custom overclocking) it can support the following resolutions: 640×350 EGA; 640×480 VGA; 800×600 SVGA; 1024×768 XGA; 1280×720 720p HDTV; 1280×768 WXGA variant; 1280×800 WXGAvariant; 1280×1024 SXGA;

1366×768 WXGA variant; 1400×1050 SXGA+; 1600×1200 UXGA; 1680×1050 WXGA+; 1920×1080 1080p HDTV; 1920×1200 WUXGA.[43]

Higher resolutions, up to 2048×1152, may work[44][45] or even 3840×2160 at 15 Hz (too low a frame rate for convincing video).[46] Note also that allowing the highest resolutions does not imply that the GPU can decode video formats at these resolutions; in fact, the Pis are known to not work reliably for H.265 (at those high resolutions), commonly used for very high resolutions (however, most common formats 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.[47] The GPU in the Raspberry Pi 3 runs at higher clock frequencies of 300 MHz or 400 MHz, compared to previous versions which ran at 250 MHz.[48]

The Raspberry Pis 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 phono connector depending on models. The television signal standards supported are PAL-BGHID, PAL-M, PAL-N, NTSC and NTSC-J.[49]

Real-time clock:

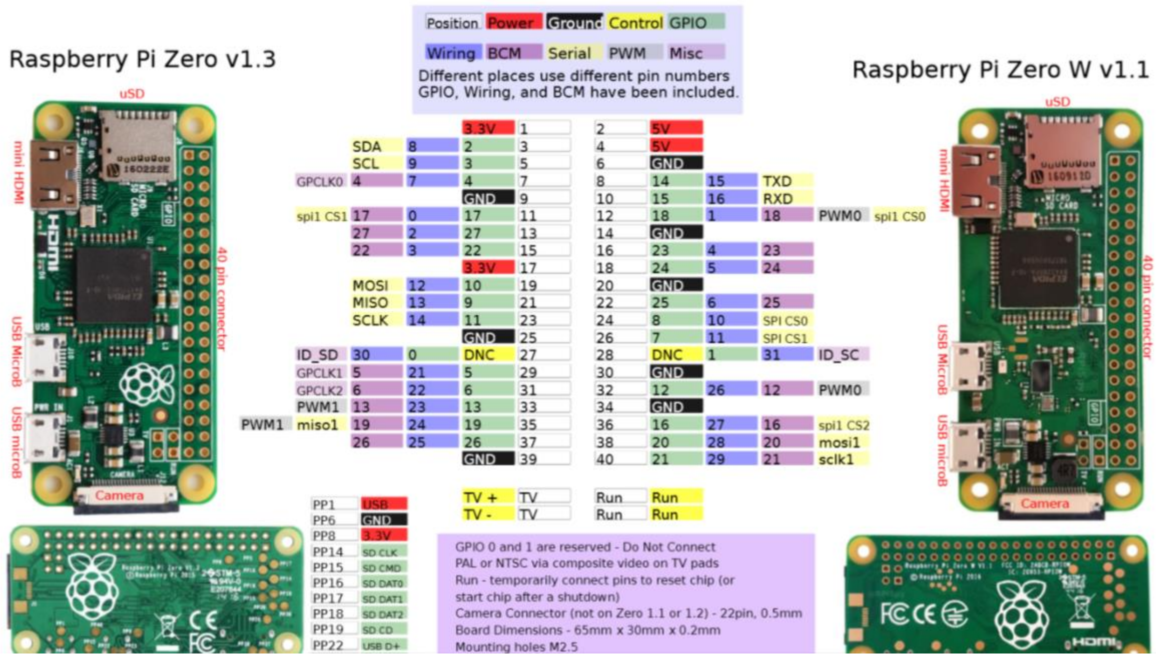
None of the current Raspberry Pi models have a built-in real-time clock, so they are unable to keep track of the time of day independently. As a workaround, a program running on the Pi can retrieve the time from a network time server or from user input at boot time, thus knowing the time while powered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

A real-time hardware clock with battery backup, such as the DS1307, may be added (often via the I²C interface).

Operating systems

Various operating systems for the Raspberry Pi can be installed on a MicroSD, MiniSD or SD card, depending on the board and available adapters; seen here is the MicroSD slot located on the bottom of a Raspberry Pi 2 board.

The Raspberry Pi Foundation provides Raspbian, a Debian-based Linux distributionfor download, as well as third-party Ubuntu, Windows 10 IoT Core, RISC OS, and specialised media centre distributions.[106] It promotes Python and Scratch as the main programming languages, with support for many other languages.[107] The default firmware is closed source, while an unofficial open source is available.[108][109][110] Many other operating systems can also run on the Raspberry Pi. Other third-party operating systems available via the official website include Ubuntu MATE, Windows 10 IoT Core, RISC OS and specialised distributions for the Kodi media centre and classroom management.



Position | Power | Ground | Control | GPIO
Wiring | BCM | Serial | PWM | Misc

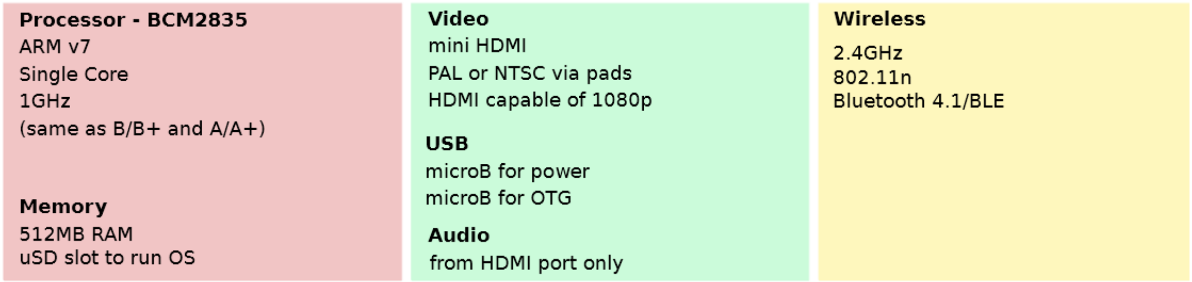
Different places use different pin numbers GPIO, Wiring, and BCM have been included.

				3.3V	1	2	5V				
		SDA	8	2	3	4	5V				
		SCL	9	3	5	6	GND				
	GPCLK0	4	7	4	7	8	14	15	TXD		
				GND	9	10	15	16	RXD		
	spi1 CS1	17	0	17	11	12	18	1	18	PWM0	spi1 CS0
		27	2	27	13	14	GND				
		22	3	22	15	16	23	4	23		
				3.3V	17	18	24	5	24		
		MOSI	12	10	19	20	GND				
		MISO	13	9	21	22	25	6	25		
		SCLK	14	11	23	24	8	10	SPI CS0		
				GND	25	26	7	11	SPI CS1		
	ID_SD	30	0	DNC	27	28	DNC	1	31	ID_SC	
	GPCLK1	5	21	5	29	30	GND				
	GPCLK2	6	22	6	31	32	12	26	12	PWM0	
	PWM1	13	23	13	33	34	GND				
PWM1	miso1	19	24	19	35	36	16	27	16	spi1 CS2	
		26	25	26	37	38	20	28	20	mosi1	
				GND	39	40	21	29	21	sclk1	

PP1	USB
PP6	GND
PP8	3.3V
PP14	SD CLK
PP15	SD CMD
PP16	SD DAT0
PP17	SD DAT1
PP18	SD DAT2
PP19	SD CD
PP22	USB D+

TV +	TV	Run	Run
TV -	TV	Run	Run

GPIO 0 and 1 are reserved - Do Not Connect
PAL or NTSC via composite video on TV pads
Run - temporarily connect pins to reset chip (or start chip after a shutdown)
Camera Connector (not on Zero 1.1 or 1.2) - 22pin, 0.5mm
Board Dimensions - 65mm x 30mm x 0.2mm
Mounting holes M2.5



Processor - BCM2835
ARM v7
Single Core
1GHz
(same as B/B+ and A/A+)

Memory
512MB RAM
uSD slot to run OS

Video
mini HDMI
PAL or NTSC via pads
HDMI capable of 1080p

USB
microB for power
microB for OTG

Audio
from HDMI port only

Wireless
2.4GHz
802.11n
Bluetooth 4.1/BLE

2.4 Raspberry Pi Camera +

5 megapixel camera capable of taking photographs of 3280 x 2464 pixels⍰Capture video at 1080p30, 720p60 and 640x480p90 resolutions⍰All software is supported within the latest version of Raspbian Operating System.

2.5 Raspberry Pi Camera

5 megapixel camera capable of taking photographs of 3280 x 2464 pixels□Capture video at 1080p30, 720p60 and 640x480p90 resolutions□All software is supported within the latest version of Raspbian Operating System.

The Camera v2is the new official camera board released by the Raspberry Pi foundation.The Raspberry Pi Camera Module v2 is a high quality 8megapixel Sony IMX219 imagesensorcustom designed add-onboardfor Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and

640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras.

- 5 megapixel camera capable of taking photographs of 3280 x 2464 pixels
- Capture video at 1080p30, 720p60 and 640x480p90 resolutions
- All software is supported within the latest version of Raspbian Operating System

The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable.The high quality Sony IMX219 imagesensor itself has a native resolution of 8megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

Applications

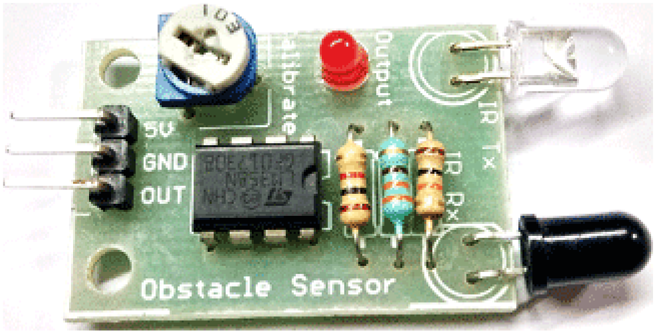
-CCTV security camera

-motion detection

-time lapse photography

2.6 IR Sensor Module

The IR sensor module consists mainly of the IR Transmitter and Receiver, Opamp, Variable Resistor (Trimmer pot), output LED in brief.



IR LED Transmitter

IR LED emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. IR LEDs have light emitting angle of approx. 20-60 degree and range of approx. few centimeters to several feets, it depends upon the type of IR transmitter and the manufacturer. Some transmitters have the range in kilometers. IR LED white or transparent in colour, so it can give out amount of maximum light.

Photodiode Receiver

Photodiode acts as the IR receiver as its conducts when light falls on it. Photodiode is a semiconductor which has a P-N junction, operated in Reverse Bias, means it start conducting the current in reverse direction when Light falls on it, and the amount of current flow is proportional to the amount of Light. This property makes it useful for IR detection. Photodiode looks like a LED, with a black colour coating on its outer side, Black colour absorbs the highest amount of light.

LM358 Opamp

LM358 is an Operational Amplifier (Op-Amp) is used as voltage comparator in the IR sensor. the comparator will compare the threshold voltage set using the preset (pin2) and the photodiode's series resistor voltage (pin3).

Photodiode's series resistor voltage drop > Threshold voltage = Opamp output is High

Photodiode's series resistor voltage drop < Threshold voltage = Opamp output is Low

When Opamp's output is **high** the LED at the Opamp output terminal **turns ON** (Indicating the detection of Object).

Variable Resistor

The variable resistor used here is a preset. It is used to calibrate the distance range at which object should be detected.

Pin Name	**Description**
VCC	Power Supply Input
GND	Power Supply Ground
OUT	Active High Output

.

IR Sensor Module Features

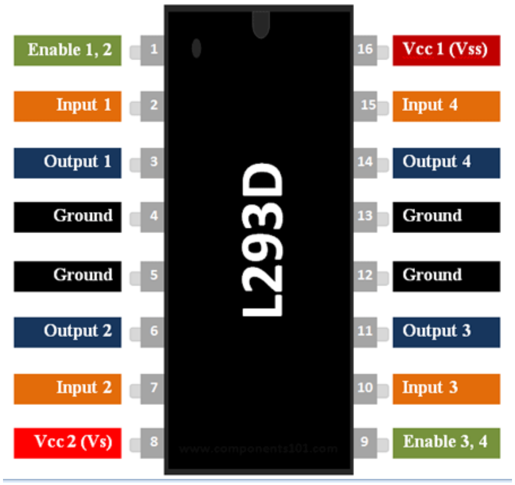
- 5VDC Operating voltage
- I/O pins are 5V and 3.3V compliant
- Range: Up to 20cm
- Adjustable Sensing range
- Built-in Ambient Light Sensor
- 20mA supply current
- Mounting hole

Applications

- Obstacle Detection
- Industrial safety devices
- Wheel encoder

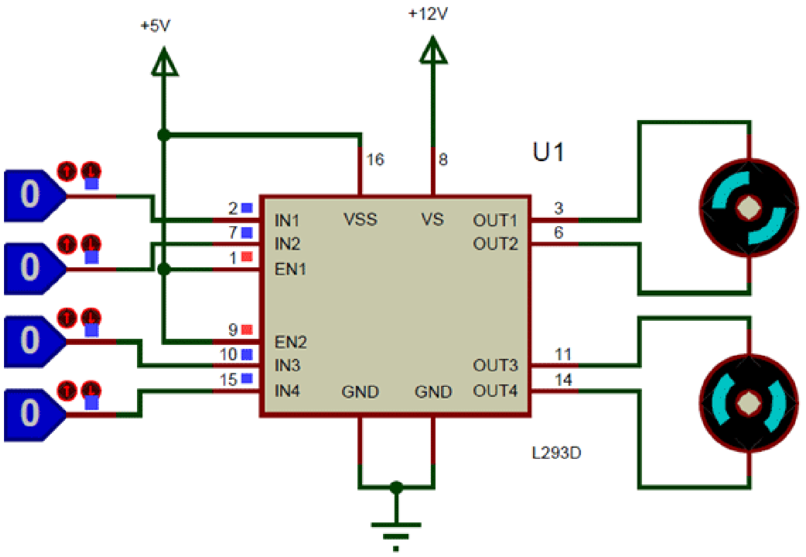
2.7 L293D Motor Driver:

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micron rollers like Arduino, PIC, ARM etc..



How to use a L293D Motor Driver IC:

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set up which is used to run motors both in clock wise and anti clockwise direction. As said earlier this IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.



All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1,2 and Enable 3,4) are used to Enable Input pins for Motor 1 and Motor 2 respectively. Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1,2 are used to control the motor 1 and Input pins 3,4 are used to control the Motor 2. The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor. You can toggle the input pins based on the following table to control your motor.

Input 1 = HIGH(5v)	Output 1 = HIGH	Motor 1 rotates in Clock wise Direction
Input 2 = LOW(0v)	Output 2 = LOW	
Input 3 = HIGH(5v)	Output 1 = HIGH	Motor 2 rotates in Clock wise Direction
Input 4 = LOW(0v)	Output 2 = LOW	

Applications:

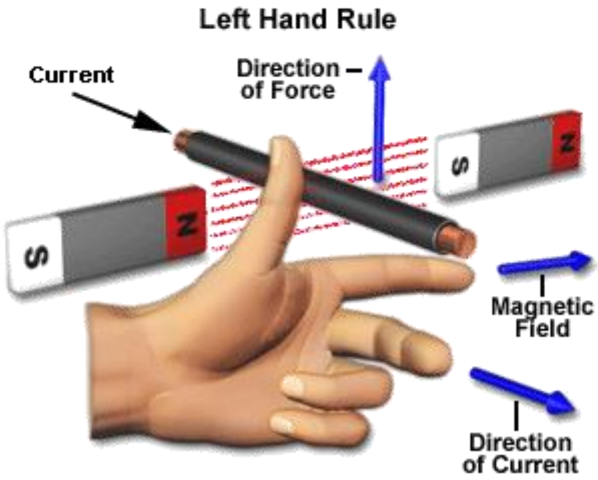
- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

2.8 DC Motor or Direct Current Motor:

The electric motor operated by dc is called **dc motor**. This is a device that converts DC electrical energy into a mechanical energy.

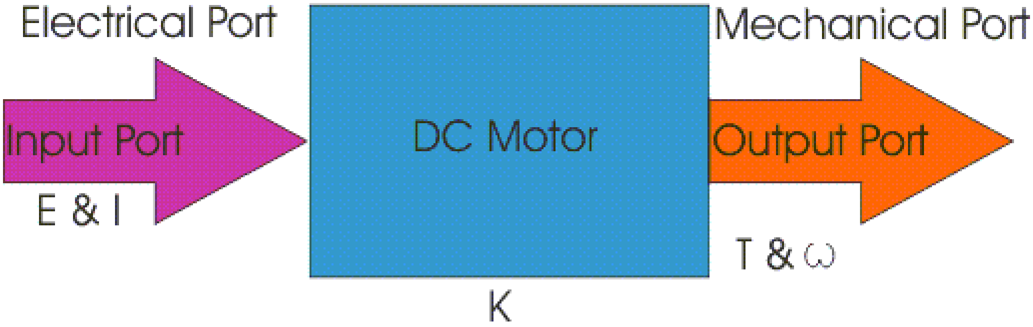
Principle of DC Motor

When a current carrying conductor is placed in a magnetic field, it experiences a torque and has a tendency to move. In other words, when a magnetic field and an electric field interact, a mechanical force is produced. The **DC motor** or **direct current motor** works on that principal. This is known as motoring action.



The direction of rotation of a this motor is given by Fleming's left hand rule, which states that if the index finger, middle finger, and thumb of your left hand are extended mutually perpendicular to each other and if the index finger represents the direction of magnetic field, middle finger indicates the direction of current, then the thumb represents the direction in which force is experienced by the shaft of the **DC motor**.

Structurally and construction wise a direct current motor is exactly similar to a DC generator, but electrically it is just the opposite. Here we unlike a generator we supply electrical energy to the input port and derive mechanical energy from the output port. We can represent it by the block diagram shown below.



Here in a DC motor, the supply voltage E and current I is given to the electrical port or the input port and we derive the mechanical output i.e. torque T and speed ω from the mechanical port or output port.

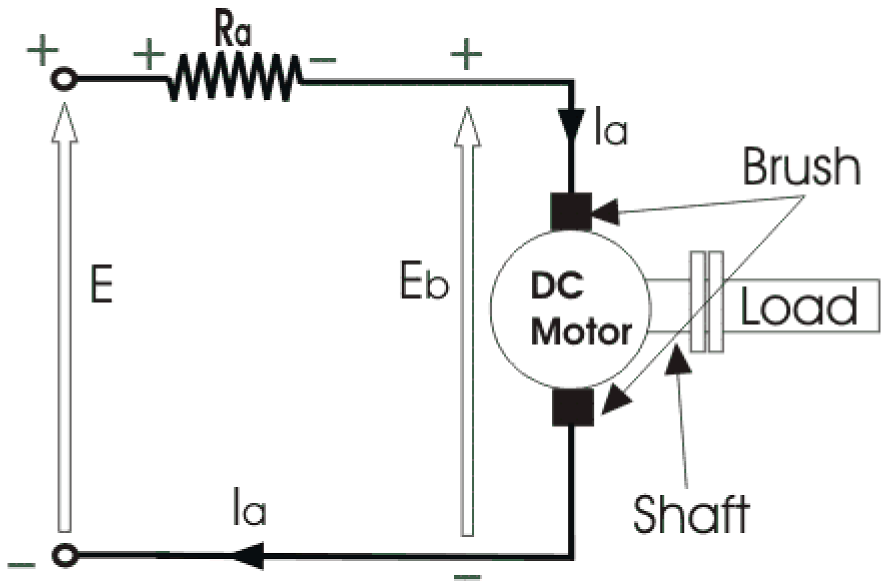
the parameter K relates the input and output port variables of the **direct current motor**.

$$T = KI \ and \ E = K\omega$$

So from the picture above, we can well understand that motor is just the opposite phenomena of a DC generator, and we can derive both motoring and generating operation from the same machine by simply reversing the ports.

Detailed Description of a DC Motor

To understand the DC motor in details lets consider the diagram below,



The circle in the center represents the direct current motor. On the circle, we draw the brushes. On the brushes, we connect the external terminals, through which we give the supply voltage. On the mechanical terminal, we have a shaft coming out from the center of the armature, and the shaft couples to the mechanical load. On the supply terminals, we represent the armature resistance R_a in series.

Now, let the input voltage E, is applied across the brushes. Electric current which flows through the rotor armature via brushes, in presence of the magnetic field, produces a torque T_g. Due to this torque T_g the dc motor armature rotates. As the armature conductors are carrying currents and the armature rotates inside the stator magnetic field, it also produces an emf E_b in the manner very similar to that of a generator. The generated Emf E_b is directed opposite to the supplied voltage and is known as the back Emf, as it counters the forward voltage.
The back emf like in case of a generator is represented by

Where, P = no of poles

ϕ = flux per pole

Z= No. of conductors

A = No. of parallel paths

and N is the speed of the DC Motor.

So, from the above equation, we can see E_b is proportional to speed 'N.' That is whenever a direct current motor rotates; it results in the generation of back Emf. Now let's represent the rotor speed by ω in rad/sec. So E_b is proportional to ω.

So, when the application of load reduces the speed of the motor, E_b decreases. Thus the voltage difference between supply voltage and back emf increases that means $E - E_b$ increases. Due to this increased voltage difference, the armature current will increase and therefore torque and hence speed increases. Thus a DC Motor is capable of maintaining the same speed under variable load.

Now armature current I_a is represented by

Now at starting,speed $\omega = 0$ so at starting $E_b = 0$.

Now since the armature winding electrical resistance R_a is small, this motor has a very high starting current in the absence of back Emf. As a result we need to use a starter for starting a DC Motor. Now as the motor continues to rotate, the back emf starts being generated and gradually the current decreases as the motor picks up speed.

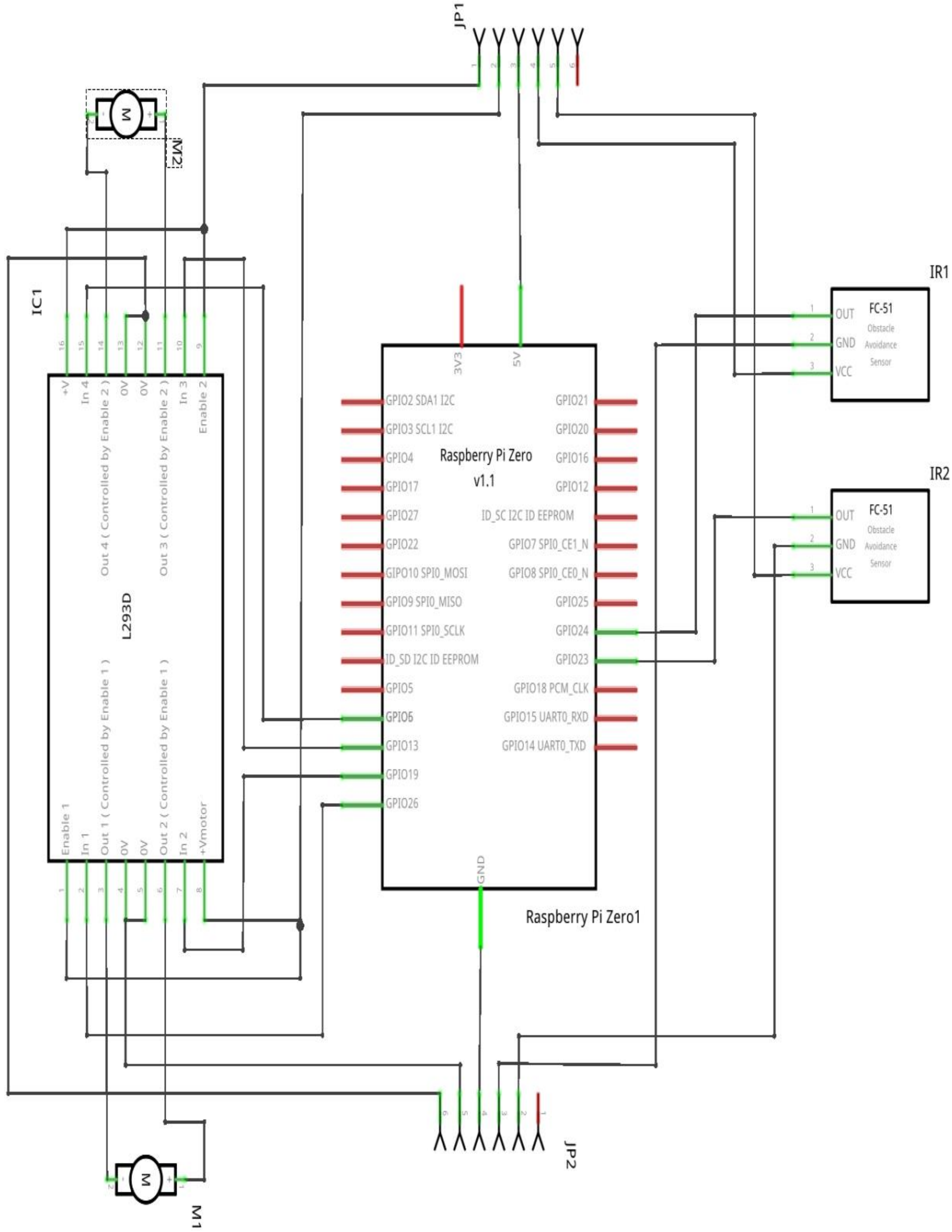
Types of DC Motors

Direct motors are named according to the connection o the field winding with the armature. There are 3 types:

1. Shunt wound DC motor
2. Series wound DC motor
3. Compound wound DC motor

CHAPTER 3

Circuit Diagram:



CHAPTER 4

Introduction to Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.
- Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End Of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained.
- Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python’s Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.
- It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python is Interpreted

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
- This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.
- One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.
- For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.
- Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.
- Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

CHAPTER 5

5. CODING

```
import cv2

import time

import RPi.GPIO as GPIO

key = cv2. waitKey(1)

webcam = cv2.VideoCapture(0)

GPIO.setmode(GPIO.BOARD)

Motor11 = 33    # Input1 Pin1

Motor12 = 31    # Input1 Pin2

Motor21 = 35    # Input2 Pin1

Motor22 = 37    # Input2 Pin2

IRleft  = 16    # Left IR sensor

IRright = 18    # Right IR sensor


GPIO.setup(Motor11,GPIO.OUT)

GPIO.setup(Motor12,GPIO.OUT)

GPIO.setup(Motor21,GPIO.OUT)

GPIO.setup(Motor22,GPIO.OUT)

GPIO.setup(IRleft,GPIO.IN)

GPIO.setup(IRright,GPIO.IN)

while True:


    check, frame = webcam.read()

    #print(check) #prints true as long as the webcam is runningq

    #frame = cv2.rotate(frame, cv2.cv2.ROTATE_180_CLOCKWISE)

    #print(frame) #prints matrix values of each framecd
```

```
(h, w) = frame.shape[:2]

center=(w/2, h/2)

c=cv2.getRotationMatrix2D(center, 180, 1.0)

frame=cv2.warpAffine(frame, c, (w,h))


cv2.imshow("Capturing", frame)

key = cv2.waitKey(1)

time.sleep(0.01)

#print(key)

x=GPIO.input(IRleft)

y=GPIO.input(IRright)

#print(x)

#print(y)

if key == ord('q'):

   print("Turning off camera.")

   webcam.release()

   print("Camera off.")

   print("Program ended.")

   print("STOP")

   GPIO.output(Motor11,GPIO.LOW)

   GPIO.output(Motor12,GPIO.LOW)

   GPIO.output(Motor21,GPIO.LOW)

   GPIO.output(Motor22,GPIO.LOW)

   cv2.destroyAllWindows()

   break

elif key == ord('s'):

   print('image saved')
   cv2.imwrite('image.jpg',frame)
elif (key==82 and key!=81 and key!=83 and key!=84):
   if (x==1 and y==1):

      print("UP")
      GPIO.output(Motor11,GPIO.LOW)
      GPIO.output(Motor12,GPIO.HIGH)
      GPIO.output(Motor21,GPIO.LOW)
      GPIO.output(Motor22,GPIO.HIGH)
      time.sleep(0.5)
elif ((key==81 and key!=82 and key!=83 and key!=84)):
   if (x==1):
      print("LEFT")
      GPIO.output(Motor11,GPIO.LOW)
      GPIO.output(Motor12,GPIO.HIGH)
      GPIO.output(Motor21,GPIO.LOW)
      GPIO.output(Motor22,GPIO.LOW)
      time.sleep(0.5)
elif key==84 and key!=81 and key!=83 and key!=82:
   print("DOWN")
   GPIO.output(Motor11,GPIO.HIGH)
   GPIO.output(Motor12,GPIO.LOW)
   GPIO.output(Motor21,GPIO.HIGH)
   GPIO.output(Motor22,GPIO.LOW)
   time.sleep(0.5)
elif (key==83 and key!=81 and key!=82 and key!=84):
   if (y==1):
```

```
            print("RIGHT")
            GPIO.output(Motor11,GPIO.LOW)
            GPIO.output(Motor12,GPIO.LOW)
            GPIO.output(Motor21,GPIO.LOW)
            GPIO.output(Motor22,GPIO.HIGH)
            time.sleep(0.5)
    else:
    #elif key == ord('s'):
        print("STOP")
        GPIO.output(Motor11,GPIO.LOW)
        GPIO.output(Motor12,GPIO.LOW)
        GPIO.output(Motor21,GPIO.LOW)
        GPIO.output(Motor22,GPIO.LOW)
import cv2
import time
key = cv2. waitKey(1)
webcam = cv2.VideoCapture(0)
while True:
    check, frame = webcam.read()
    cv2.imshow("Capturing", frame)
    key = cv2.waitKey(1)
    time.sleep(0.01)
    print(key)
    if key == ord('q'):
        print("Turning off camera.")
        webcam.release()
        print("Camera off.")
        print("Program ended.")
        cv2.destroyAllWindows()
        break
from getkey import getkey, keys


while 1:
    key = getkey()
    if key == keys.UP:
        print("U")
    elif key == keys.DOWN:
        print("D")
    elif key == keys.LEFT:
        print("L")
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
Motor11 = 33    # Input1 Pin1
Motor12 = 31    # Input1 Pin2
Motor21 = 35    # Input2 Pin1
Motor22 = 37    # Input2 Pin2
IRleft  = 16    # Left IR sensor
IRright = 18    # Right IR sensor

GPIO.setup(Motor11,GPIO.OUT)
GPIO.setup(Motor12,GPIO.OUT)
GPIO.setup(Motor21,GPIO.OUT)
GPIO.setup(Motor22,GPIO.OUT)

GPIO.setup(IRleft,GPIO.IN)

GPIO.setup(IRright,GPIO.IN)

while True:

    x=GPIO.input(IRleft)

    y=GPIO.input(IRright)

    print("IRL"+str(x))

    print("IRR"+str(y))

    '''GPIO.output(Motor11,GPIO.LOW)

    GPIO.output(Motor12,GPIO.HIGH)

    GPIO.output(Motor21,GPIO.LOW)

    GPIO.output(Motor22,GPIO.HIGH)

    time.sleep(2)

    GPIO.output(Motor11,GPIO.LOW)

    GPIO.output(Motor12,GPIO.LOW)

    GPIO.output(Motor21,GPIO.LOW)

    GPIO.output(Motor22,GPIO.LOW)'''

    time.sleep(2)
```

Type your text

CHAPTER 6

6. COST OF ESTIMATION

SNO	HARDWARES	QUANTITY	PRICE
1.	CHASIS	1 SET	RS. 300
2.	Raspberry pi zero	1 NO	RS.2000
3.	Raspberry pi camera	1 NO	RS.800
4.	IR Sensor	2 NO	RS. 250
5.	L293D Motor driver module	1 NO	RS.300
6.	6V lithium ion Battery	2 NO' s	RS. 500
7.	DC Geared Motor	2 NO'S	RS.200
8.	External memory 16gb	1 no	RS.400
TOTAL			RS.4,750

CHAPTER 7

7. CONCULSION

The target of this project is to explain the usage of Snake Bot for surveillance and search

The objective of this program is to build a snake like robot that can traverse many different and difficult terrain types, such as small holes, tunnels and gaps and mapping the information in developing the modular snake robot design, we considered several factors such as size, weight, power while designing and calculating the necessary torque in every joint.

This work aimed at facilitating the snake inspired robots in several areas. A specific class of gaits was analysed, and the model was designed in order to understand the mechanics of the locomotion as it relates to the mechanical design, as well as gait design. This model can move both in narrow place and in place with a height difference.

Finally, a snake inspired robot was designed to be implemented, is a full modular architecture and an innovative manufacturing process aimed at reducing costs.

The system is simple. Power efficient

CHAPTER 8

8. PROJECT MEMBERS ADDRESS

SNO	NAME	ADDRESS	EMAIL ID & MOBILE NO.
1	ASWIN.K	No.34b, Nehru Street, Vinayagapuram, Kolathur, Chennai- 600099.	aswin.24it@licet.ac.in 7338739552
2	S. GANESH	No.1********, 5th ******** Che********.	********il.com 8********
3	S. KARTHIKEYAN	No.13/4 R******** t********, Chennai- ********.	ki********@gmail.com 9********
4	K. LOKESHWARAN	No.10******** St********r, A********.	lo********@gmail.com 9********
5	B. NARAYANAN	No.1******** ******** ******** 60********.	na********om 8********
6	T. PURSOTHAMAN	No********eet, V********ai- ********9.	t.p********l.com ********8

CHAPTER 9

9. REFERENCES

LITRATURE REVIEW

1: Mechatronics – A. Preumont

- Findings-
- Control of Active Structures, was to cross the bridge between Structural Dynamics and Automatic Control. To insist on important control structure interaction issues, the book often relied on "ad-hoc" models and intuition (e.g. a thermal analogy for piezoelectric loads), and was seriously lacking in accuracy and depth on transduction and energy conversion mechanisms which are essential in active structures. The present book project was initiated in preparation for a new edition, with the intention of redressing the imbalance, by including a more serious treatment of the subject. As the work developed, it appeared that this topic was broad enough to justify a book on its own.

- 2: Amphibionics Build Your Own Biologically Inspired Reptilian Robot – Karl Williams

- Findings-
- The robots in this book were designed to imitate biological **life forms**. Watching the snake robot moving through a room, it is interesting to observe the surprised reactions of people when it quickly turns towards them. People actually regard the robot as being alive. I am struck with the thought that although these machines are not alive in our biological sense, they actually are alive, but as life-forms unto themselves.

- **3: Design of a modular snake robot. – Cornell Wright, Aaron Johnson, Aaron Peck, Zachary McCord, Allison Naaktgeeboren, Phillip Gianfortoni, Ross Hatton, and Howie Choset.**

- Findings-

- Many factors such as size, power, and weight constrain are an integral part of the design of modular snake robots. Meeting these constrains requires implementing a complex mechanical and electrical architecture, this paper presents a solution of construction of 16 aluminium modules and creation of a super servo, a modified hobby servo. The paper also explains about some sensors which are connected to the snake bot. Also to protect the snake robot from the external environment the outer body or the skin of the snake bot should be robust which is well explained in this paper.

CHAPTER 10

Project picture