# AUTONOMOUS SELF DRIVING CAR

A Report submitted in partial fulfilment of the requirement for the award of degree of

Bachelor of Technology

In

Electrical and Electronics Engineering

Under the Supervision of

Ms. Mamta Tholia

By


Vikrant Thakur (35515004915)

Naveen Meena (021150014915)

Chetan Atrai (41915004915)

Amit Aryan (41015004915)

**MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY**

C-4, Janakpuri, New Delhi-58

Affiliated to Guru Gobind Singh Indraprastha University, Delhi


2019

# DECLARATION

We, students of Bachelor of Technology pursing Electrical and Electronics Engineering hereby declare that the project work done on "**Autonomous Self Smart Car**" submitted to Maharaja Surajmal Institute of Technology, Janakpuri Delhi in partial fulfilment of the requirement for the award of degree of Bachelor of Technology comprises of our original work and has not been submitted anywhere else for any other degree to the best of our knowledge.


Vikrant Thakur (35515004915)

Naveen Meena (02115004915)

Chetan Atrai (41915004915)

Amit Aryan (41015004915)

# CERTIFICATE

This is to certify that the project work done on "**Autonomous Self Smart Car**" submitted to Maharaja Surajmal Institute of Technology, Janakpuri Delhi by "Vikrant Thakur, Naveen Meena, Chetan Atrai, Amit Aryan," in partial fulfillment of the requirement for the award of degree of Bachelor of Technology, is a bonafide work carried out by him/her under my supervision and guidance. This project work comprises of original work and has not been submitted anywhere else for any other degree to the best of my knowledge.

| | |
|---|---|
| Signature of Supervisor | Signature of HOD |
| (Project Supervisor) | (HOD, EEE) |

# ACKNOWLEDGEMENT

Team effort together with precious words of encouragement and guidance makes daunting tasks achievable. It is a pleasure to acknowledge the direct and implied help we have received at various stages in the task of developing the project. It would not have been possible to develop such a project without the furtherance on part of numerous individuals. We find it impossible to express our thanks to each one of them in words, for it seems too trivial when compare to the profound encouragement that they extended to us.

We are grateful to **Dr. Meena Tushir, HOD, EEE**, for having given us opportunity to do this project, which was of great interest to us. Our sincere thanks to **Ms. Mamta Tholia, Assistant Professor, EEE** for believing in us and providing motivation all through. Without her guidance this project would not be such a success.

Finally, we would like to thank our project head **Ms. Rakhi Kamra, Assistant Professor, EEE**, our parents, all friends, and well-wishers for their valuable help and encouragement throughout the project. At last we thank the almighty, who had given the strength to complete this project on time.

Vikrant Thakur (35515004915)

Naveen Meena (02115004915)

Chetan Atrai (41915004915)

Amit Aryan (41015004915)

# Contents

## ABSTRACT

The project aims to build a monocular vision autonomous car prototype using Raspberry Pi as a microprocessor development board. A Pi camera along with the ultrasonic sensors is used to provide real-time to the controlling mechanism of the car. The car is capable of reaching the given destination safely and intelligently, thus avoiding the risk of human errors. Many existing algorithms like lane detection, obstacle detection is combined together to provide the necessary control to the car.



Vikrant Thakur (35515004915)

Naveen Meena (02115004915)

Chetan Atrai (41915004915)

Amit Aryan (41015004915)

## CHAPTER-1

### 1.1 INTRODUCTION

- A self-driving car, also known as a robot car, autonomous car, or driverless car, is a vehicle that is capable of sensing its environment and moving with little or no human input.

- Autonomous cars combine a variety of sensors to perceive their surroundings, such as radar, Lidar, sonar, GPS, odometry and inertial measurement units. Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage.

- Potential benefits include reduced costs, increased safety, increased mobility, increased customer satisfaction and reduced crime. Safety benefits include a reduction in traffic collisions, resulting in injuries and related costs, including for insurance. Automated cars are predicted to increase traffic flow; provide enhanced mobility for children, the elderly,] disabled, and the poor; relieve travellers from driving and navigation chores; increase fuel efficiency of vehicle, significantly reduce needs for parking space; reduce crime;]and facilitate business models for transportation as a service, especially via the sharing economy.

- Problems include safety, possible technological errors, liability, legal framework and government regulations; risk of loss of privacy and security concerns, such as hackers or cyber terrorism; concern about the loss of driving-related jobs in the road transport industry; and risk of increased suburbanization as travel becomes more convenient.

- The first truly automated car was developed in 1977, by Japan's Tsukuba Mechanical Engineering Laboratory. The vehicle tracked white street markers, which were interpreted by two cameras on the vehicle, using an analog computer for signal processing. The vehicle reached speeds up to 30 kilometres per hour (19 mph), with the support of an elevated rail.

- Autonomous prototype cars appeared in the 1980s, with Carnegie Mellon University's Navlab and ALV projects funded by DARPA starting in 1984 and Mercedes-Benz and Bundeswehr University Munich's EUREKA Prometheus Project in 1987. By 1985, the ALV had demonstrated self-driving speeds on two-lane roads of 31 kilometres per hour (19 mph) with obstacle avoidance added in 1986 and off-road driving in day and nighttime conditions by 1987. From the 1960s through the second DARPA Grand Challenge in 2005, automated vehicle research in the U.S. was primarily funded by DARPA, the US Army, and the U.S. Navy, yielding incremental advances in speeds, driving competence in more complex conditions, controls, and sensor systems.] Companies and research organizations have developed prototypes.

- The U.S. allocated $650 million in 1991 for research on the National Automated Highway System, which demonstrated automated driving through a combination of automation, embedded in the highway with automated technology in vehicles and cooperative networking between the vehicles and with the highway infrastructure. The program concluded with a successful demonstration in 1997 but without clear direction or funding to implement the system on a larger scale. Partly funded by the National Automated Highway System and DARPA, the Carnegie Mellon University Navlab drove 4,584 kilometres (2,848 mi) across America in 1995, 4,501 kilometres

(2,797 mi) or 98% of it autonomously. Navlab's record achievement stood unmatched for two decades until 2015 when Delphi improved it by piloting an Audi, augmented with Delphi technology, over 5,472 kilometres (3,400 mi) through 15 states while remaining in self-driving mode 99% of the time. n 2015, the US states of Nevada, Florida, California, Virginia, and Michigan, together with Washington, D.C., allowed the testing of automated cars on public roads.

- In 2017, Audi stated that its latest A8 would be automated at speeds of up to 60 kilometres per hour (37 mph) using its "Audi AI." The driver would not have to do safety checks such as frequently gripping the steering wheel. The Audi A8 was claimed to be the first production car to reach level 3 automated driving, and Audi would be the first manufacturer to use laser scanners in addition to cameras and ultrasonic sensors for their system.

- In November 2017, Waymo announced that it had begun testing driverless cars without a safety driver in the driver position; however, there was still an employee in the car. In October 2018, Waymo announced that its test vehicles had traveled in automated mode for over 10,000,000 miles (16,000,000 km), increasing by about 1,000,000 miles (1,600,000 kilometres) per month. In December 2018, Waymo was the first to commercialize a fully autonomous taxi service in the U.S.

Vehicular communication systems:-

- Vehicle networking may be desirable due to difficulty with computer vision being able to recognize brake lights, turn signals, buses, and similar things. However, the usefulness of such systems would be diminished by the fact current cars are equipped with them; they may also pose privacy concerns.

- Individual vehicles may benefit from information obtained from other vehicles in the vicinity, especially information relating to traffic congestion and safety hazards. Vehicular communication systems use vehicles and roadside units as the communicating nodes in a peer-to-peer network, providing each other with information. As a cooperative approach, vehicular communication systems can allow all cooperating vehicles to be more effective. According to a 2010, study by the National Highway Traffic Safety administration, vehicular communication systems could help avoid up to 79 percent of all traffic accidents.

- There has so far been no complete implementation of peer-to-peer networking on the scale required for traffic: each individual vehicle would have to connect with potentially hundreds of different vehicles that could be going in and out of range.

- In 2012, computer scientists at the University of Texas in Austin began developing smart intersections designed for automated cars. The intersections will have no traffic lights and no stop signs, instead of using computer programs that will communicate directly with each car on the road.

- In 2017, Researchers from Arizona State University developed a 1/10 scale intersection and proposed an intersection management technique called Crossroads. It was shown that the Crossroads is resilient to network delay of V2I communication and Worst-case Execution time of the intersection manager. In 2018, a robust approach was introduced which is resilient to model mismatch and external disturbance like wind and bump.

- Among connected cars, an unconnected one is the weakest link and will be increasingly banned from busy high-speed roads, predicted a Helsinki think tank in January 2016.

## 1.2 PROPOSED PROJECT

- Rushing around, trying to get errands done, thinking about the things to be bought from the nearest grocery store has become a part of our daily schedule. Driver error is one of the most common cause of traffic accidents, and with cell phones, in car entertainment systems, more traffic and more complicated road systems, it isn't likely to go away. With the number of accidents increasing day by day, it has become important to take over the human errors and help the mankind. All of this could come to an end with self-driving cars which just need to know the destination and then let the passengers continue with their work. This will avoid not only accidents but also bring a self-relief for minor day to day driving activities for small items.

- Modern vehicles provide partly automated features such as keeping the car within its lane, speed controls or emergency braking. Nonetheless, differences remain between a fully autonomous self-driving car on one hand and driver assistance technologies on the other hand. According to the BBC, confusion between those concepts leads to deaths.
- Association of British Insurers considers the usage of the word autonomous in marketing for modern cars to be dangerous, because car ads make motorists think 'autonomous' and 'autopilot' means a vehicle can drive itself, when they still rely on the driver to ensure safety. Technology alone still is not able to drive the car.
- When some car makers suggest or claim vehicles are self-driving, when they are only partly automated, drivers risk becoming excessively confident, leading to crashes, while fully self-driving cars are still a long way off in the UK.

How do self-driving cars work?

- The technological leaps and bounds we have made in recent years have finally culminated in making self-driving cars a reality. Self-driving cars are only possible today due to the existence of three technologies:

**Figure 1**

1.IoT sensors

- There are many types of sensors available today that make autonomous cars a reality. Sensors for blind-spot monitoring, forward collision warning, radar, camera, LIDAR, and ultrasonic all work together to make navigation of a self-driving car possible.

2. IoT connectivity

- Self-driving cars use cloud computing to act upon traffic data, weather, maps, adjacent cars, and surface conditions among others. This helps them monitor their surroundings better and make informed decisions. Self-driving cars must be connected to the internet even if edge computing hardware can solve small computing tasks locally.

3. Software algorithms

- All the data the car collects needs to be analysed to determine the best course of action. This is the main function of the control algorithms and software. This is the most complex part of the self-driving car since it has to make decisions flawlessly. A "flaw," like in Uber's self-driving accident, can be fatal.

4. What do they have to offer?

- Autonomous cars are much safer than human-driven cars. They're unaffected by factors like driver fatigue, emotion, or illness. This makes them very safe.
- Self-driving cars are always attentive and active, observing their environments and scanning multiple directions. It would be difficult to make a move that the car has not anticipated.
- Adoption of self-driving cars would mean safer roadways, which in turn would mean less demand for emergency response services, pricey insurance premiums, not to mention savings in accident-related health care for everyone.

### 1.3 Field of application:

Automated trucks: -

- Several companies are said to be testing automated technology in semi trucks. Otto, a self-driving trucking company that was acquired by Uber in August 2016, demonstrated their trucks on the highway before being acquired. In May 2017, San Francisco-based startups Embark announced a partnership with truck manufacturer Peterbilt to test and deploy automated technology in Peterbilt's vehicles. Waymo has also said to be testing automated technology in trucks, however no timeline has been given for the project.

- In 2016, Anheuser-Busch Inc. and Uber Technologies Inc. joined together and successfully made the first commercial delivery of beer using a self-driving truck. No human action had involved in the driving process and the truck travelled 120 miles.

- In March 2018, Starsky Robotics, the San Francisco-based automated truck company, completed a 7-mile (11 km) fully driverless trip in Florida without a single human in the truck. Starsky Robotics became the first player in the self-driving truck game to drive in fully automated mode on a public road without a person in the cab.

- In Europe, truck platooning is being considered with the Safe Road Trains for the Environment approach.
- Lockheed Martin with funding from the U.S. Army developed an automated truck convoying system that uses a lead truck operated by a human driver with a number of trucks following autonomously. Developed as part of the Army's Autonomous Mobility Applique System (AMAS), the system consists of an automated driving package that has been installed on more than nine types of vehicles and has completed more than 55,000 hours of driving at speeds up to 64 kilometres per hour (40 mph) as of 2014. As of 2017 the Army was planning to field 100–200 trucks as part of a rapid-fielding program.

Transport systems:-

- In Europe, cities in Belgium, France, Italy and the UK are planning to operate transport systems for automated cars, and Germany, the Netherlands, and Spain have allowed public testing in traffic. In 2015, the UK launched public trials of the LUTZ Pathfinder automated pod in Milton Keynes. Beginning in summer 2015, the French government allowed PSA Peugeot-Citroen to make trials in real conditions in the Paris area. The experiments were planned to be extended to other cities such as Bordeaux and Strasbourg by 2016.The alliance between French companies THALES and Valeo (provider of the first self-parking car system that equips Audi and Mercedes premi) is testing its own system. New Zealand is planning to use automated vehicles for public transport in Tauranga and Christchurch.
- In China, Baidu and King Long produce automated minibus, a vehicle with 14 seats, but without driving seat. With 100 vehicles produced, 2018 will be the first year with commercial automated service in China. Those minibuses should be at level 4, that is driverless in closed road

## CHAPTER-2

### 2.1 BLOCK DIAGRAM OF PROJECT:

- SMART CAR- OVERVIEW PART: 1

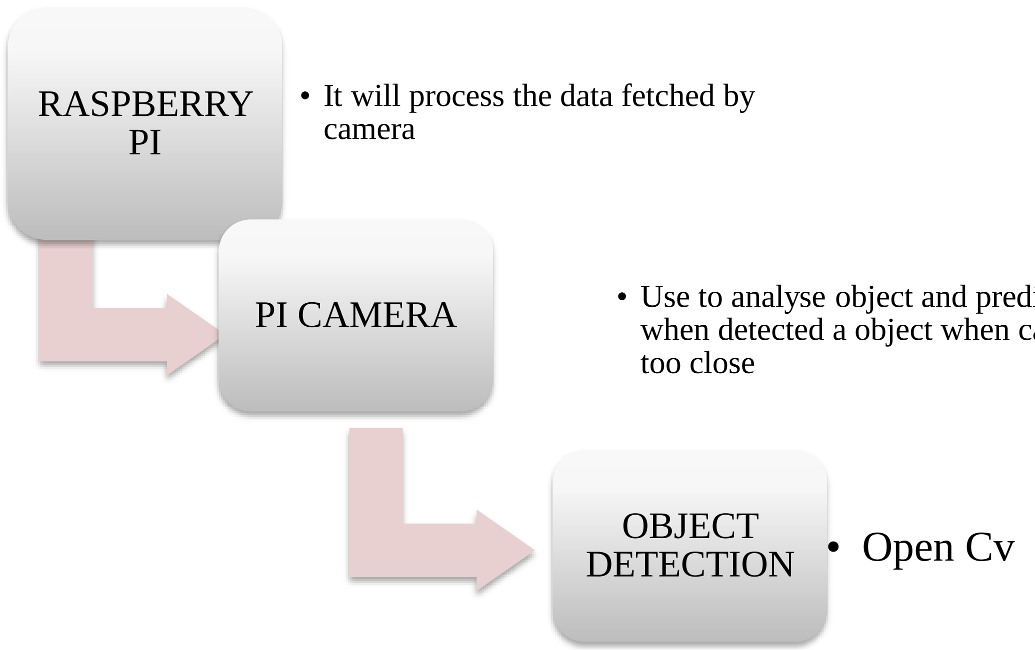



**Figure 2**

- SMART CAR- OVERVIEW PART 2

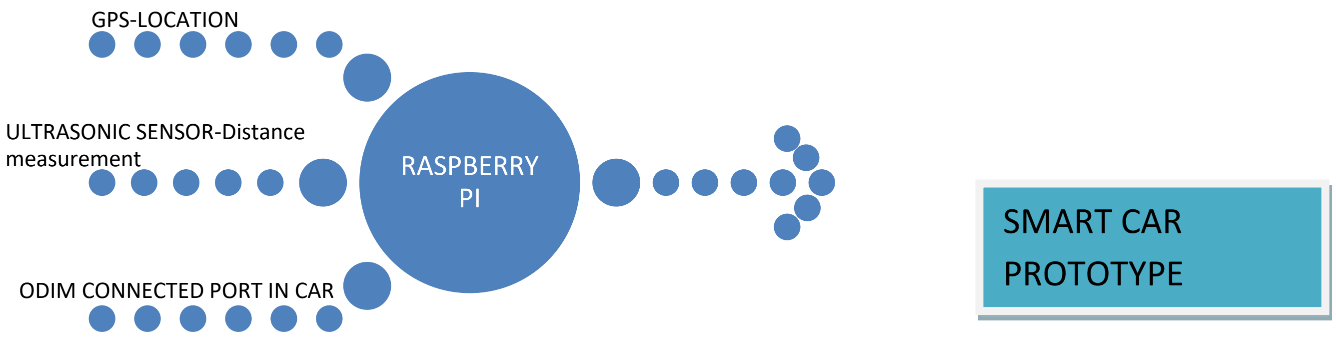



Figure 3

2.2 Circuit Diagram:

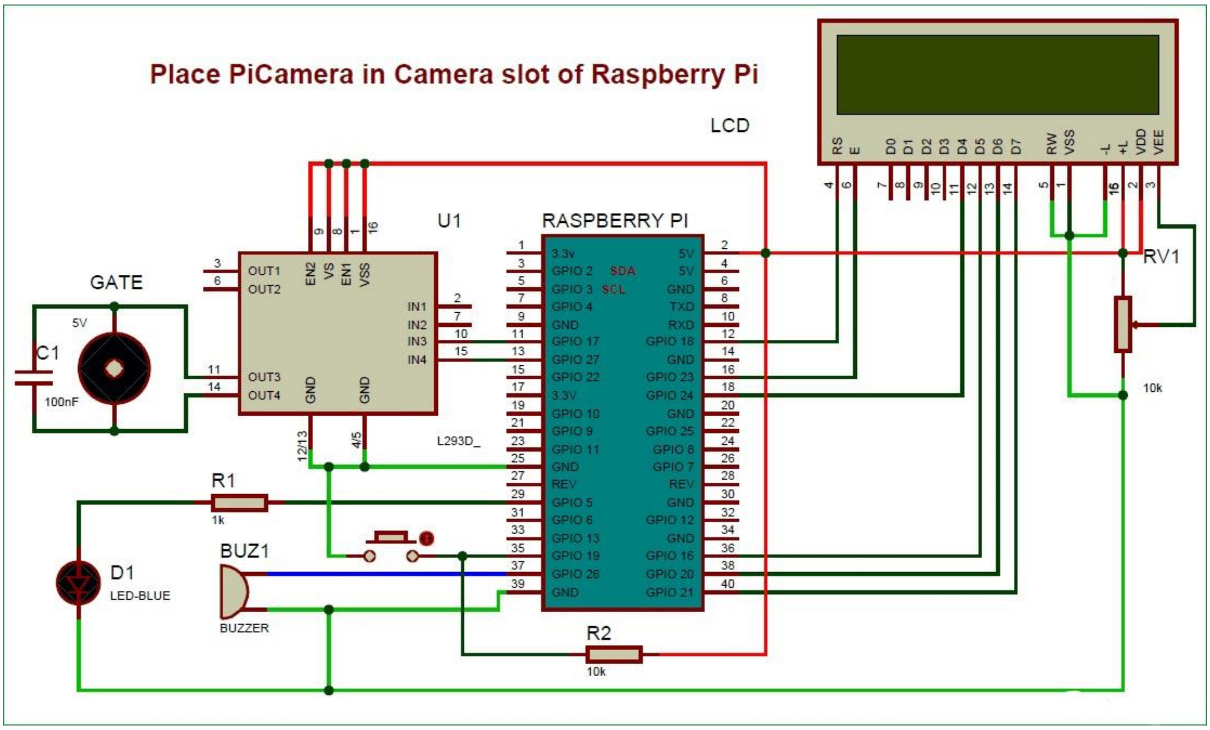



Figure 4 (With Pi camera installation)

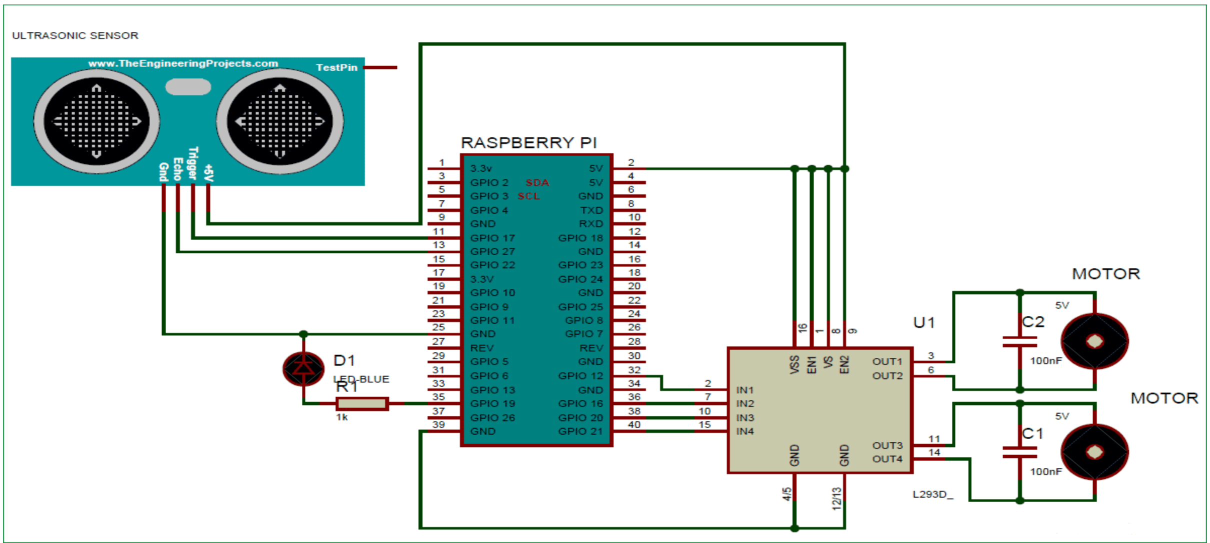



Figure 5 (With ultrasonic sensor connection)

## 2.3 Autonomous Vehicle Hardware:

The hardware components of the autonomous car are analogous to the physical parts of the human body, which allow us to interact with the stimuli of the outside world. The hardware components enable the car to complete such tasks as seeing (through sensors), communicating (through V2V technology), and moving (through actuators).

- **Sensors:** Sensors are the components that allow the autonomous to take in raw information about the environment. Sensors are like your eyes, which enable you to understand what's going on in your surroundings. The main sensors in autonomous cars include GPS/Inertial Measurement Unit (IMUs), camera, Lidar, and radar. Each of these sensors has their respective advantages and disadvantages. Lidar, for example, is great at capturing information in various types of ambient light (whether night or day), whereas cameras may have difficulty in handling certain occlusions caused by shadows or other poor lighting conditions. Accordingly, most autonomous vehicles combine the readings of multiple sensor types to add extra redundancy and compensate for the weaknesses of the various sensors in a process called sensor fusion.
- **V2X technology (V2V and V2I technology):** V2V and V2I components enable the autonomous vehicle to talk and receive information from other machine agents in the environment, such as transmitted information from a city light that it has turned green or warnings from an oncoming car. You can think of V2X technology as akin to your mouth and ears. Your mouth allows you to communicate to other humans, and your ears allow you to understand what other humans are communicating to you.
- **Actuators:** Actuators are the components of a machine responsible for controlling and moving the system. Actuators are like muscles of your body, responding to electrochemical signals from your brain so that you move such parts as your arm or leg.

### 2.4 Autonomous Vehicle Software

Whereas the hardware components of the autonomous car enable the car to perform such functions as see, communicate, and move, the software is like the brain, which processes information about the environment so that the car understands what action to take—whether to move, stop, slow down, etc. Autonomous vehicle software can be categorized into three systems: perception, planning, and control.

- Perception: The perception system refers to the ability of the autonomous vehicle to understand what the raw information coming in through the sensors or V2V components mean. It enables the car to understand from a given picture frame whether a certain object is another car, a pedestrian, or something else entirely. This process is analogous to how our brains process the information we obtain through sight into meaning. The photoreceptors of our eyes (the sensors) absorb light waves emanating from the environment and converts those light waves into electrochemical signals. Networks of neurons pass these electrochemical signals all the way back to the visual cortex of the brain, where our brain processes what these electrochemical signals mean. In this way, our brain can understand whether a certain light pattern hitting our retina represents a chair, a plant, or another person.

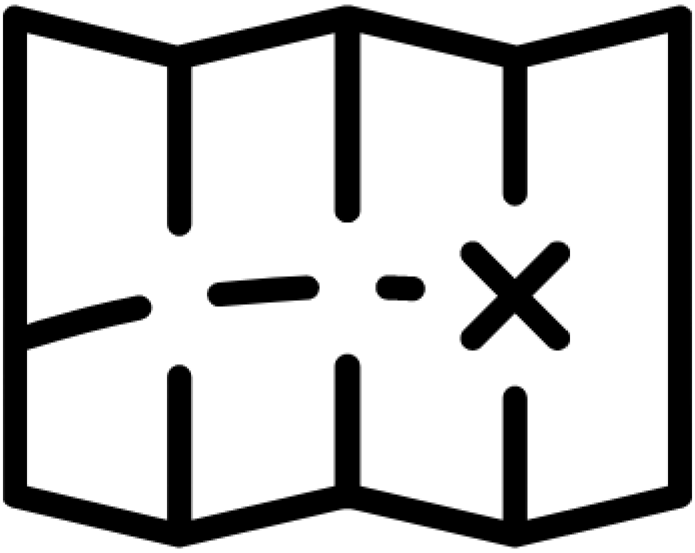


**Figure 6**

- Planning: The planning system refers to the ability of the autonomous vehicle to make certain decisions to achieve some higher order goals. This is how the autonomous vehicle knows what to do in a situation—whether to stop, go, slow down, etc. The planning system works by combining the processed information about the environment (i.e. from the sensors and V2X components) with established policies and knowledge about how to navigate in the environment (e.g. do not run over pedestrians, slow down when approaching a stop sign, etc.) so that the car can determine what action to take (e.g. overtake another car, how to reach the destination, etc.). Analogously, just like the planning system in the autonomous car, the processes in the frontal lobe of the human brain enable us to reason and make decisions, such as what to wear in the morning or what we should do for fun on the weekend.

- Control: The control system pertains to the process of converting the intentions and goals derived from the planning system into actions. Here the control system tells the hardware (the actuators) the necessary inputs that will lead to the desired motions. For example, an autonomous vehicle, knowing that it should slow down when approaching a red light, translates this knowledge into the action of applying the brakes. In humans, the processes that occur in the cerebellum play the analogous role. The cerebellum is responsible for the important function of motor control. It enables us, for example, to chew when the desired intention is to eat.

**How They All Work Together**

Now that we have a good understanding of the main components of an autonomous vehicle, let's review a scenario of how they all work together.

**Scenario:** The car has stopped at an intersection in front of the red light.

**Mission:** The car should move forward when the traffic light turns green without violating any traffic laws or hurting other beings.

1. **Sensors:** The car’s sensors take in raw information about the environment. It does not know what this information means yet—at least not until it gets to the perception stage.

2. **V2X technology:** The traffic light communicates to the car that it has just turned green. Other surrounding cars communicate their position in the environment.

3. **Perception Stage:** The vehicle turns the raw information coming in from the perception stage into actual meaning. The camera information reveals that the light has just turned green and that there is a pedestrian crossing in front of the vehicle into the street.

4. **Planning Stage:** The vehicle combines the sensing information processed during the perception stage with the incoming V2X information to determine how to behave. The car’s policy is to generally move when the light turns green; however, it has an overriding policy that it should not run over pedestrians. What should the car do in this scenario? The car decides that, based on the combination of environmental information and the general policies of how it should operate, it should not move.

5. **Control Stage:** The car must translate its decision to not move into an action. In this case, this action (or rather, inaction in this case) is to stay still and keep the brakes applied.

6. **Actuators:** The car keeps the brake applied, which is the result of its decision-making process stated above.

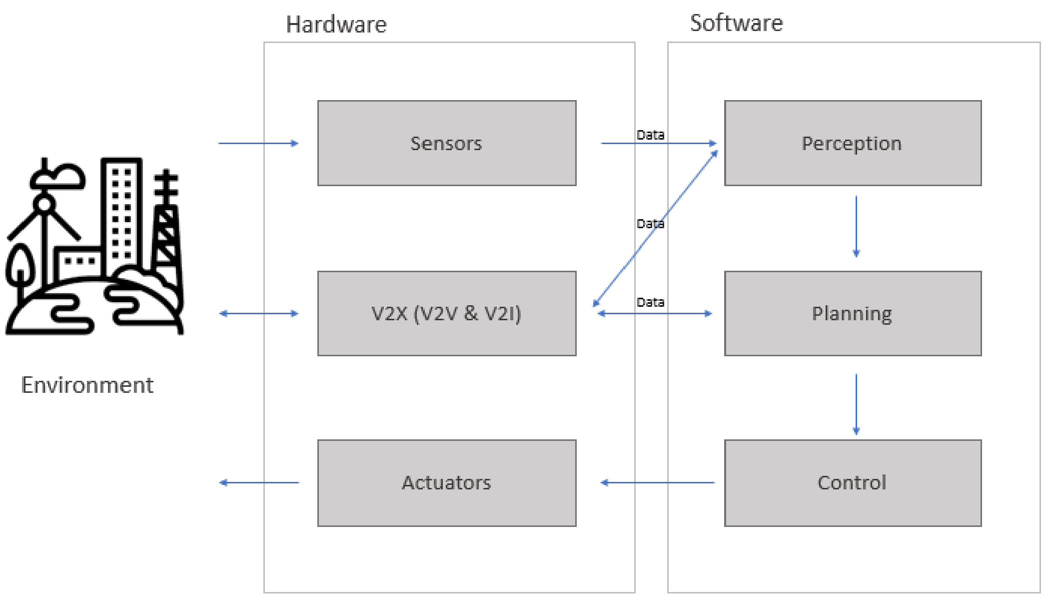



**Figure 7**

## 2.5 Six Levels of Autonomy

- It may come as a surprise to learn that vehicle automation is not an either/or proposition. And not everyone wants complete automation. Putting your car on automatic pilot in fair weather along a standard stretch of highway may be fine, but any driver would want the option to regain control of the car should circumstances demand it. In reality, the autonomy of vehicles is graded on a scale from zero to five, with zero meaning no autonomy and five signifying complete autonomy.

### Level Zero: No Automation

- Readers who are a bit more experienced know that automobiles didn't always come with all those computerized controls. There was a time when cars had no computers at all, and in the early days they didn't even have power steering or power brakes. At level zero, all aspects of the driving task are in the hands of the driver. Think of the car your father or grandfather may have driven – or even your car if it is an older model.
- "A vehicle that fits into this category," says an article from CNET, "relies on a human to dictate every driving action." The driver has full control. Level zero doesn't just apply to an old Ford Model T. CNET offers as an example: "your uncle Rick's 2005 Honda."

### Level One: Driver Assistance

- At this level, the automobile includes some built-in capabilities to operate the vehicle. The vehicle may assist the driver with tasks like steering or acceleration/deceleration. For several years now cars have been manufactured with controls on the steering column that allow the driver to maintain a constant speed or gradually increase or decrease speed. These functions are enacted by the driver and not automatically performed by the automobile.
- Most modern cars fit into this level. If your vehicle has adaptive cruise control or lane-keeping technology, it's probably at level one.
- You might be surprised to learn that cruise control was invented by a blind mechanical engineer named Ralph Teetor. Chrysler was the first car company to offer cruise control in 1958 (they called it Speedostat). Later the technology became standard in all Cadillacs.

### Level Two: Partial Automation

- In 2017, the NHTSA adopted SAE's six levels in their Automated Vehicles Policy. (They previously defined only five levels.) At this level of automation, two or more automated functions work together to relieve the driver of control. An example is a system with both adaptive cruise control and automatic emergency braking. The driver must remain fully engaged with the driving task, but you will notice the gradual transfer of control from man to machine. NHTSA refers to this as an advanced driver assistance system (ADAS).
- Examples of level two include Tesla Autopilot and General Motors Super Cruise. A GM Super Cruise ad says "Bring us your doubts and we'll bring you the future,"

and calls the technology "the first true hands-free driving system for the freeway." Another is the Mercedes-Benz Distronic Plus, as demonstrated by an owner who seems to have some reservations about its capabilities.

Level Three: Conditional Automation

- This level is marked by both the execution of steering and acceleration/deceleration and the monitoring of the driving environment. In levels zero through two, the driver does all the monitoring. At level three, the driver is still required, but the automobile can perform all aspects of the driving task under some circumstances. Levels three and higher qualify as automated driving systems (ADS), according technology is demonstrated in this video.

Level Four: High Automation

- Level four vehicles don't need a human driver. The vehicle can essentially do all the driving, but the driver can intervene and take control as needed. This level of automation means that the car can perform all driving functions "under certain conditions." The test vehicles currently on the road would fall under this category.
- Waymo LLC (once known as the Google self-driving car project) is testing level four vehicles. An article from DesignNews, "Autonomous Cars Will Move to Level 4 in 2018,"says that new sensor technology from Velodyne Lidar amounts to "a huge step forward" for autonomous cars. But they say that level five is still a decade away.

Level Five: Full Automation

- A completely automated vehicle can perform all driving functions under all conditions. In this situation, humans are just passengers. But as CNET authors Kyle Hyatt and Chris Paukert tell us, "It's hard to imagine a world where Level 5 autonomous vehicles become the norm, available to all. If that happens, how would that change the way that we live?"
- One writer for The Atlantic speculates that fully automated cars could result in free transportation for everybody – with a catch. Judith Donath pictures a world where a free driverless car will chauffeur you and your family across town, so long as you stop 15 minutes or so at a sponsor's store. She thinks that "targeted stops" at places like McDonald's, Starbucks or a bookstore would be enough to entice the advertisers to pay for the ride.
- Of course, we already have automated transportation in some places. Have you ever been to an airport that shuttles passengers from one terminal to another without a driver? Such automation is being considered for other mass transit systems to take the place of buses and taxis in town.
- In the formal SAE definition below, note in particular what happens in the shift from SAE 2 to SAE 3: the human driver no longer has to monitor the environment. This is the final aspect of the "dynamic driving task" that is now passed over from the human to the automated system. At SAE 3, the human driver still has the responsibility to intervene when asked to do so by the automated system. At SAE 4 the human driver is relieved of that responsibility and at SAE 5 the automated system will never need to ask for an intervention.

| SAE Level | Name | Execution of steering and acceleration/ deceleration | Monitoring of driving environment | Fallback performance of dynamic driving task | System capability (driving modes) | | |
|---|---|---|---|---|---|---|---|
| 0 | No Automation | Human driver | Human driver | Human driver | n/a | | |
| 1 | Driver Assistance | The driving mode-specific execution by a driver assistance system of "either steering or acceleration/deceleration" | | Human driver and system | | | Some driving modes |
| 2 | Partial Automation | The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration | | System | | | |
| 3 | Conditional Automation | The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task | | System | System | Human driver | Some driving modes |
| 4 | High Automation | | | | | System | Many driving modes |
| 5 | Full Automation | | | | | | All driving modes |

**Table-1**

- **Note:** using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving

## 2.6 Difficulty Challenges

### 2.6.1 Technical Challenges

- There are different systems that help the self-driving car control the car. The car navigation system, the location system, the electronic map, the map matching, the global path planning, the environment perception, the laser perception, the radar perception, the visual perception, the vehicle control, the perception of vehicle speed and direction, the vehicle control method. These systems need to get improvement.

- The challenge for driverless car designers is to produce control systems capable of analyzing sensory data in order to provide accurate detection of other vehicles and the road ahead. Modern self-driving cars generally use Bayesian simultaneous localization and mapping (SLAM) algorithms, which fuse data from multiple sensors and an off-line map into current location estimates and map updates. Waymo has developed a variant of SLAM with detection and tracking of other moving objects (DATMO), which also handles obstacles such as cars and pedestrians. Simpler systems may use roadside real-time locating system (RTLS) technologies to aid localization. Typical sensors include Lidar, stereo vision, GPS and IMU. Control systems on automated cars may use Sensor Fusion, which is an approach that integrates information from a variety of sensors on the car to produce a more consistent, accurate, and useful view of the environment.

- Driverless vehicles require some form of machine vision for the purpose of visual object recognition. Automated cars are being developed with deep neural networks, a type of deep learning architecture with many computational stages, or levels, in which neurons are simulated from the environment that activate the network. The neural network depends on an extensive amount of data extracted from real-life driving scenarios, enabling the neural network to "learn" how to execute the best course of action.

- In May 2018, researchers from MIT announced that they had built an automated car that can navigate unmapped roads. Researchers at their Computer Science and Artificial Intelligence Laboratory (CSAIL) have developed a new system, called MapLite, which allows self-driving cars to drive on roads that they have never been on before, without using 3D maps. The system combines the GPS position of the vehicle, a "sparse topological map" such as OpenStreetMap, (i.e. having 2D features of the roads only), and a series of sensors that observe the road conditions.

### 2.6.2 Human Factor Challenges

- Self-driving cars are already exploring the difficulties of determining the intentions of pedestrians, bicyclists, and animals, and models of behavior must be programmed in to driving algorithms. Human road users also have the challenge of determining the intentions of autonomous vehicles, where there is no driver with which to make eye contact or exchange hand signals. Drive.ai is testing a solution to this problem that involves LED signs mounted on the outside of the vehicle, announcing status such as "going now, don't cross" vs. "waiting for you to cross".

- Two human-factor challenges are important for safety. One is the handoff from automated driving to manual driving, which may become necessary due to unfavorable or unusual road conditions, or if the vehicle has limited capabilities. A sudden handoff could leave a human driver dangerously unprepared in the moment. In the long term, humans who have less practice at driving might have a lower skill level and thus be more dangerous in manual mode. The second challenge is known as risk compensation: as a system is perceived to be safer, instead of benefiting entirely from all of the increased safety, people engage in riskier behavior and enjoy other benefits. Semi-automated cars have been shown to suffer from this problem, for example with users of Tesla Autopilot ignoring the road and using electronic devices or other activities against the advice of the company that the car is not capable of being completely autonomous. In the near future, pedestrians and bicyclists may travel in the street in a riskier fashion if they believe self-driving cars are capable of avoiding them.

- In order for people to buy self-driving cars and vote for the government to allow them on roads, the technology must be trusted as safe. Self-driving elevators were invented in 1900, but the high number of people refusing to use them slowed adoption for several decades until operator strikes increased demand and trust was built with advertising and features like the emergency stop button.

## CHAPTER-3

### 3.1 PROJECT COMPONENTS

#### 3.1.1 Raspberry pi3

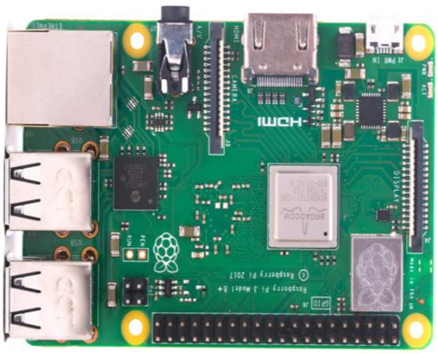


**Figure 8**

The Raspberry Pi Foundation has revealed its latest leap-forward, introducing a Wi-Fi-enabled Pi 3 computer with twice the performance of the previous generation

The Raspberry Pi 3 will cost the same as its predecessor, but feature much more powerful hardware. Bluetooth will be built into the board for the first time, and is powered by a Quad Core Broadcom BCM2837 64bit ARMv8 processor.

The Pi 3 runs at 1.2 GHz, compared to the Pi 2's 900MHz, and also has an upgraded power system, and the same four USB ports and extendable 'naked board' design as the Pi 2. "Four years ago today, we launched the first Raspberry Pi with our friends at Premier Farnel," Pi founder Eben Upton said.

So successful has the board been, it is now almost certainly the best-selling British-made and designed computer in history, Upton told the BBC. "Today we’re launching Raspberry Pi 3: it’s still $35 and it’s still the size of your credit card, but now it comes with on-board wireless LAN and Bluetooth, 50 percent more processing power, and a Quad Core 64bit processor. The new Raspberry Pi opens up even more possibilities for IoT and embedded projects; we hope you like it as much as we do."

With its built-in wireless connectivity, the new Raspberry Pi is clearly positioned as a low-cost hub for Internet of Things devices, or as the flexible, low-cost basis of new types of connected gadgets. The new bump to a 2.5 amps power source means it will be able to power more complex USB devices without the need for a second power cable.

The Raspberry Pi 3 Model B is available immediately, for just over £27. Both the Raspberry Pi 1 Model B+ and Raspberry Pi 2 Model B will still be made and sold.

More than five million Raspberry Pis have been sold since the computer was first launched, while the recent announcement of the absurdly inexpensive £4 Pi Zero gave a new burst of momentum to the low-cost, hackable computer project.

Claire Doyle, head of Raspberry Pi at partners Element 14, said: "Today, as we celebrate the 4th birthday of Raspberry Pi, we reach another major milestone on the Pi journey with the launch of an even faster, technically advanced board for an amazing price of USD35. The new Raspberry Pi 3 Model B offers a higher level of performance than any other Raspberry Pi board on the market."

The full specs for the Raspberry Pi 3 include:

- CPU: Quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz
- GPU: 400MHz VideoCore IV multimedia
- Memory: 1GB LPDDR2-900 SDRAM (i.e. 900MHz)
- USB ports: 4
- Video outputs: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
- Network: 10/100Mbps Ethernet and 802.11n Wireless LAN
- Peripherals: 17 GPIO plus specific functions, and HAT ID bus
- Bluetooth: 4.1
- Power source: 5 V via MicroUSB or GPIO header
- Size: 85.60mm × 56.5mm
- Weight: 45g (1.6 oz)

**Raspberry Pi-3 Pin Configuration:**

RASPBERRY PI 3 is a development board in PI series. It can be considered as a single board computer that works on LINUX operating system. The board not only has tons of features it also has terrific processing speed making it suitable for advanced applications. PI board is specifically designed for hobbyist and engineers who are interested in LINUX systems and IOT (Internet of Things).

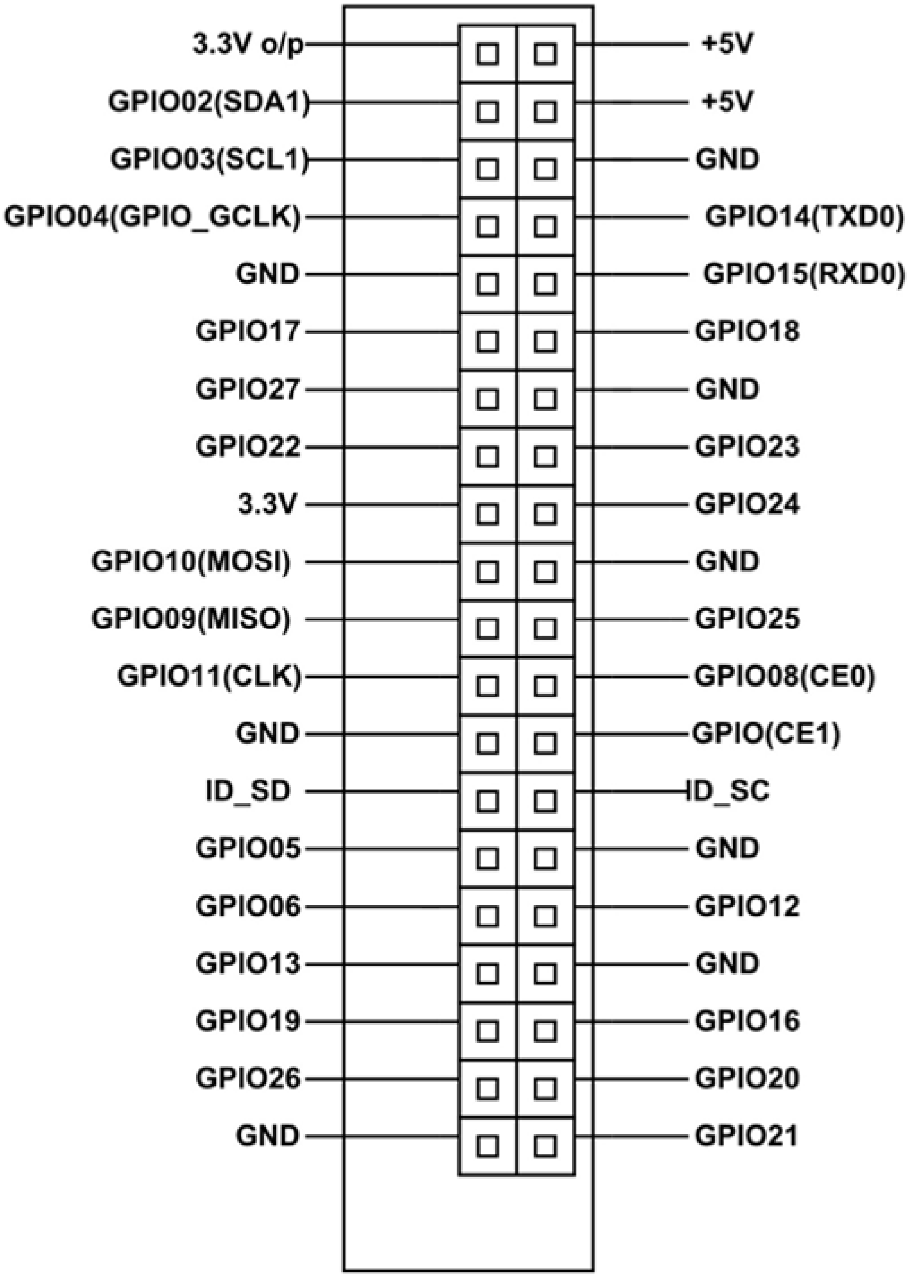

3.3V o/p
+5V
GPIO02(SDA1)
+5V
GPIO03(SCL1)
GND
GPIO04(GPIO_GCLK)
GPIO14(TXD0)
GND
GPIO15(RXD0)
GPIO17
GPIO18
GPIO27
GND
GPIO22
GPIO23
3.3V
GPIO24
GPIO10(MOSI)
GND
GPIO09(MISO)
GPIO25
GPIO11(CLK)
GPIO08(CE0)
GND
GPIO(CE1)
ID_SD
ID_SC
GPIO05
GND
GPIO06
GPIO12
GPIO13
GND
GPIO19
GPIO16
GPIO26
GPIO20
GND
GPIO21

**Figure 9**

| **PIN GROUP** | **PIN NAME** | **DESCRIPTION** |
|---|---|---|
| POWER SOURCE | +5V, +3.3V, GND and Vin | +5V -power output<br>+3.3V -power output<br>GND – GROUND pin |
| COMMUNICATION INTERFACE | UART Interface (RXD, TXD) [(GPIO15, GPIO14)] | UART (Universal Asynchronous Receiver Transmitter) used for interfacing sensors and other devices. |
| | SPI Interface (MOSI, MISO, CLK, CE) x 2<br>[SPI0-(GPIO10, GPIO9, GPIO11, GPIO8)]<br>[SPI1--(GPIO20, GPIO19, GPIO21, GPIO7)] | SPI (Serial Peripheral Interface) used for communicating with other boards or peripherals. |
| | TWI Interface (SDA, SCL) x 2 [(GPIO2, GPIO3)]<br>[(ID_SD, ID_SC)] | TWI (Two Wire Interface) Interface can be used to connect peripherals. |
| INPUT OUTPUT PINS | 26 I/O | Although these some pins have multiple functions.They can be considered as I/O pins. |
| PWM | Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19 | These 4 channels can provide PWM (Pulse Width Modulation) outputs.<br>*Software PWM available on all pins |
| EXTERNAL INTERRUPTS | All I/O | In the board all I/O pins can be used as Interrupts. |

Table-2

### Raspberry Pi 3 Technical Specifications:

| | |
|---|---|
| **Microprocessor** | Broadcom BCM2837 64bit Quad Core Processor |
| **Processor Operating Voltage** | 3.3V |
| **Raw Voltage input** | 5V, 2A power source |
| **Maximum current through each I/O pin** | 16mA |
| **Maximum total current drawn from all I/O pins** | 54mA |
| **Flash Memory (Operating System)** | 16Gbytes SSD memory card |
| **Internal RAM** | 1Gbytes DDR2 |
| **Clock Frequency** | 1.2GHz |
| **GPU** | Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GLES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high- profile decode.<br>Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure. |
| **Ethernet** | 10/100 Ethernet |
| **Wireless Connectivity** | BCM43143 (802.11 b/g/n Wireless LAN and Bluetooth 4.1) |
| **Operating Temperature** | -40ºC to +85ºC |

**Table-3**

### Board Connectors:

| Name | Description |
|---|---|
| **Ethernet** | Base T Ethernet Socket |
| **USB** | 2.0 (Four sockets) |
| **Audio Output** | 3.5mm Jack and HDMI |
| **Video output** | HDMI |
| **Camera Connector** | 15-pin MIPI Camera Serial Interface (CSI-2) |
| **Display Connector** | Display Serial Interface (DSI) 15-way flat flex cable connector with two data lanes and a clock lane. |
| **Memory Card Slot** | Push/Pull Micro SDIO |

**Table-4**

**Where RASPBERRY PI 3 is used?**

RASPBERRY PI platform is most used after ADRUINO. Although overall applications of PI are less it is most preferred when developing advanced applications. Also, the RASPBERRY PI is an open source platform where one can get a lot of related information so you can customize the system depending on the need.

Here are few examples where Raspberry pi3 is chosen over other microcontrollers and development boards:

1. Where the system processing is huge. Most Arduino boards all have clock speed of less than 100MHz, so they can perform functions limited to their capabilities. They cannot process high end programs for applications like Weather Station, Cloud server, gaming console etc. With 1.2GHz clock speed and 1GB ram Raspberry pi can perform all those advanced functions.

2. Where wireless connectivity is needed. Raspberry pi 3 has wireless LAN and Bluetooth facility by which you can setup wifi hotspot for internet connectivity. For Internet of Things this feature is best suited.

3. RASPBERRY PI had dedicated port for connecting touch LCD display which is a feature that completely omits the need of monitor.

4. RASPBERRY PI also has dedicated camera port so one can connect camera without any hassle to the PI board.

5. RASPBERRY PI also has PWM outputs for application use.

There are many other features like HD steaming which further promote the use of RASPBERRY PI.

**How to Use RASPBERRY PI 3:**

As mentioned earlier PI is simply a COMPUTER ON A SINGLE BOARD so it cannot be used like ARDUINO development boards. For the PI to start working we need to first install OPERATING SYSTEM. This feature is similar to our PC. The PI has dedicated OS for it; any other OS will not work.

We will discuss the programming of PI in step by step below.

1. Take the 16GB micro SD card and dedicate it specifically for PI OS.
2. Choose and Download OS software. [https://www.raspberrypi.org/downloads/] ('NOOBS' recommended for beginners)
3. Format the SD card and install OS on to the SD memory card using convenient methods.
4. Take the SD card after OS installation and insert it in PI board.
5. Connect monitor, keyboard and mouse
6. Power the board with micro USB connector
7. Once the power is tuned ON the PI will run on the OS installed in the memory card and will start from boot.
8. Once all drivers are checked the PI will ask for authorization, this is set by default and can be changed.
9. After authorization you will reach desktop where all application program development starts.

On the PI you can download application programs required for your use and can directly install as you do for your PC. After that you can work on developing required program and get the PI run the developed programs.

**Applications**

- Hobby projects.
- Low cost PC/tablet/laptop
- IoT applications
- Media centre
- Robotics
- Industrial/Home automation
- Server/cloud server
- Print server
- Security monitoring
- Web camera

- Gaming
- Wireless access point
- Environmental sensing/monitoring (e.g. WEATHER STATION).

### 3.1.2 PI Camera:

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs. It’s easy to use for beginners, but has plenty to offer advanced users if you’re looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion and other video cleverness. You can also use the libraries we bundle with the camera to create effects.

If you’re interested in the nitty-gritty, you’ll want to know that the module has a five-megapixel fixed-focus camera that supports 1080p30, 720p60 and VGA90 video modes, as well as stills capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Pi camera Python library.

The camera module is very popular in-home security applications, and in wildlife camera traps.

Features

- 5MP sensor
- Wider image, capable of 2592x1944 stills, 1080p30 video
- 1080p video supported
- CSI
- Size: 25 x 20 x 9 mm

Camera Details:

The camera consists of a small (25mm by 20mm by 9mm) circuit board, which connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's image sensor has a native resolution of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90. The camera module is shown below:

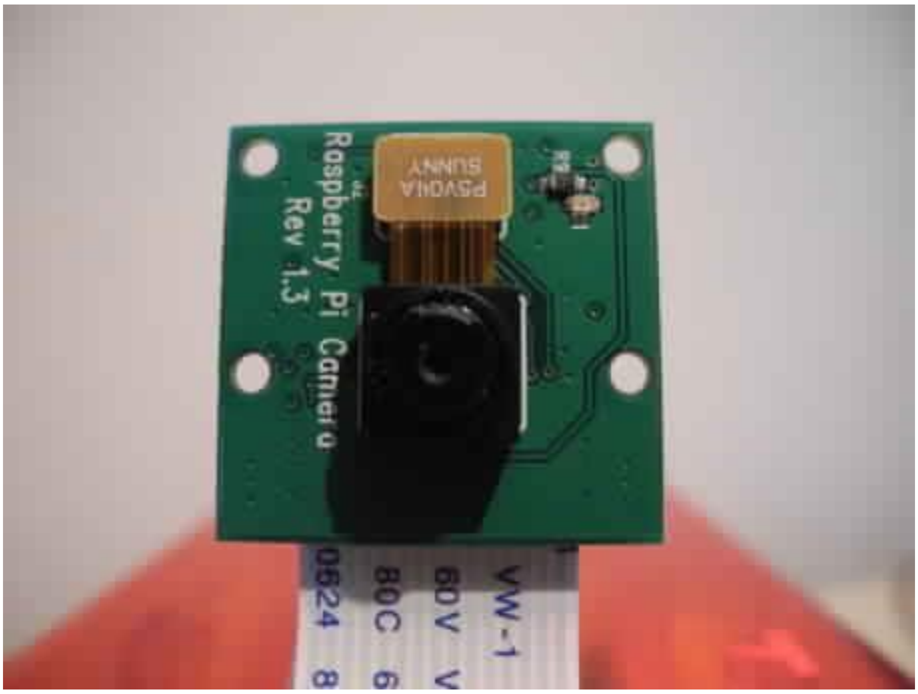



**Figure 10**

Installation involves connecting the ribbon cable to the CSI connector on the Raspberry Pi board. This can be a little tricky, but if you watch the videos that demonstrate how it is done, you shouldn't have any trouble.

When you purchase the camera, you will receive a small camera board and cable. You'll want to devise some method of supporting the camera in order to use it. Some camera stands and Raspberry Pi cases are now available. You can also rig up something simple yourself if you wish. I attached mine to a case using a small piece of plastic and double-sided tape.

Camera connection:

The flex cable inserts into the connector situated between the Ethernet and HDMI ports, with the silver connectors facing the HDMI port. The flex cable connector should be opened by pulling the tabs on the top of the connector upwards then towards the Ethernet port. The flex cable should be inserted firmly into the connector, with care taken not to bend the flex at too acute an angle. The top part of the connector should then be pushed towards the HDMI connector and down, while the flex cable is held in place.

Update the SD card:

In order to use the camera you must be using a recent operating system that knows that the camera exists. The easiest way to do this is to grab the latest Raspbian image from the RaspberryPi.org site and create a fresh SD card.Enable camera in raspi-config settings

Reboot. If you are using a fresh image the raspi-config utility should load. If it doesn't then you can run it maually using : sudo raspi-config Sekect the “Camera” option and press “Enter”.


**Figure 11**

Select “Enable” and press “Enter”.

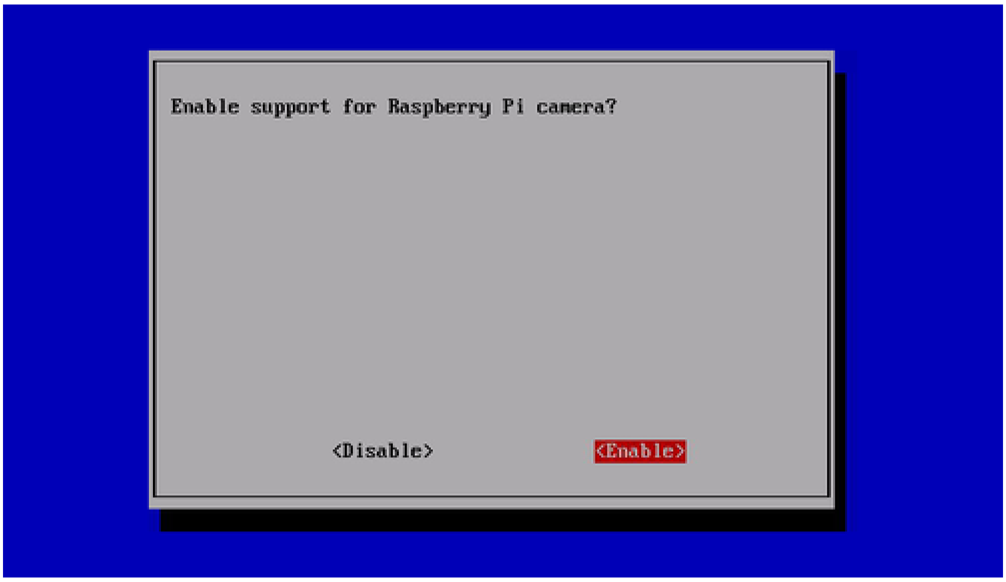



**Figure 12**

Select “Yes” and press “Enter”. Your Pi will reboot.

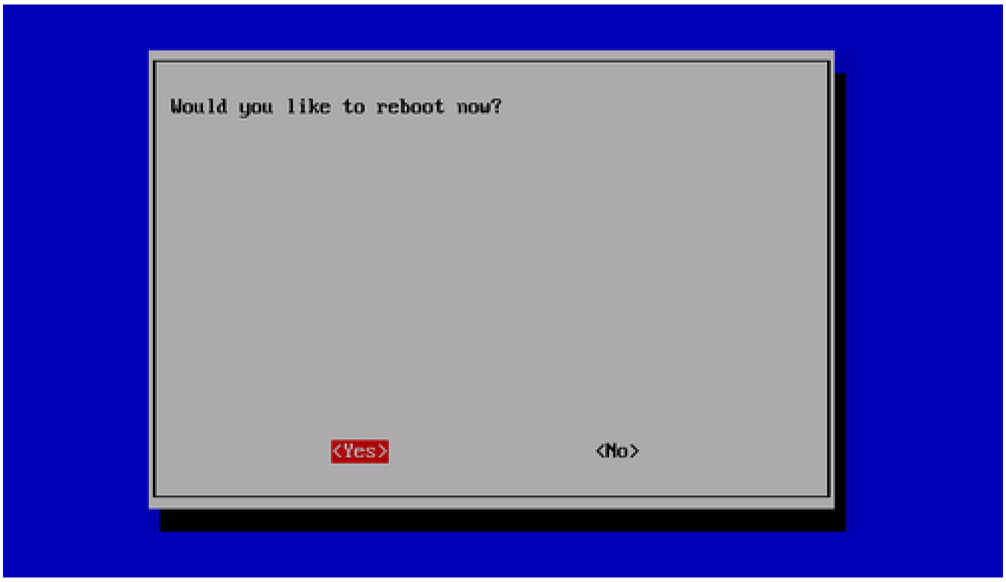



**Figure 13**

Updating your operating and enabling the camera using raspi-config did two things. It told your Pi that there is a camera attached and it added two command line utilities. rasp still rasped These allow you to capture still photos and HD video respectively.

### 3.1.3 Ultrasonic sensors:

**Figure 14**

Ultrasonic transducers or ultrasonic sensors are a type of acoustic sensor divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

In a similar way to radar and sonar, ultrasonic transducers are used in systems which evaluate targets by interpreting the reflected signals. For example, by measuring the time between sending a signal and receiving an echo the distance of an object can be calculated. Passive ultrasonic sensors are basically microphoning that detect ultrasonic noise that is present under certain conditions.

Ultrasonic probes and ultrasonic baths apply ultrasonic energy to agitate particles in a wide range of materials.

Ultrasonic sensors can detect movement of targets and measure the distance to them in many automated factories and process plants. Sensors can have an on or off digital output for detecting the movement of objects, or an analog output proportional to distance. They can sense the edge of material as part of a web guiding system.

Ultrasonic sensors are widely used in cars as parking sensors to aid the driver in reversing into parking spaces. They are being tested for a number of other automotive uses including ultrasonic people detection and assisting in autonomous UAV navigation.

Because ultrasonic sensors use sound rather than light for detection, they work in applications where photoelectric sensors may not. Ultrasonic are a great solution for clear object detection, clear label detection[5] and for liquid level measurement, applications that photoelectric struggle with because of target translucence. As well, target colour or reflectivity do not affect ultrasonic sensors, which can operate reliably in high-glare environments.

Passive ultrasonic sensors may be used to detect high-pressure gas or liquid leaks, or other hazardous conditions that generate ultrasonic sound. In these devices, audio from the transducer (microphone) is converted down to human hearing range.

High-power ultrasonic emitters are used in commercially available ultrasonic cleaning devices. An ultrasonic transducer is affixed to a stainless steel pan which is filled with a solvent (frequently water or isopropanol). An electrical square wave feeds the transducer, creating sound in the solvent strong enough to cause cavitation.

Ultrasonic technology has been used for multiple cleaning purposes. One of which that is gaining a decent amount of traction in the past decade is ultrasonic gun cleaning.

Ultrasonic testing is also widely used in metallurgy and engineering to evaluate corrosion, welds, and material defects using different types of scans.

**Circuit:**

The Ultrasonic transmitter and receiver circuit is built with 555 timer IC or CMOS (complementary metal-oxide-semiconductor devices). This circuit works with 9 volts to 12 volts DC.These are preset controlled variable oscillators are specific controlled variable oscillators. The working frequencies specific value is likely to drift due to changes in temperature. The drift in frequency affects the transmission range from the ultrasonic transducer.

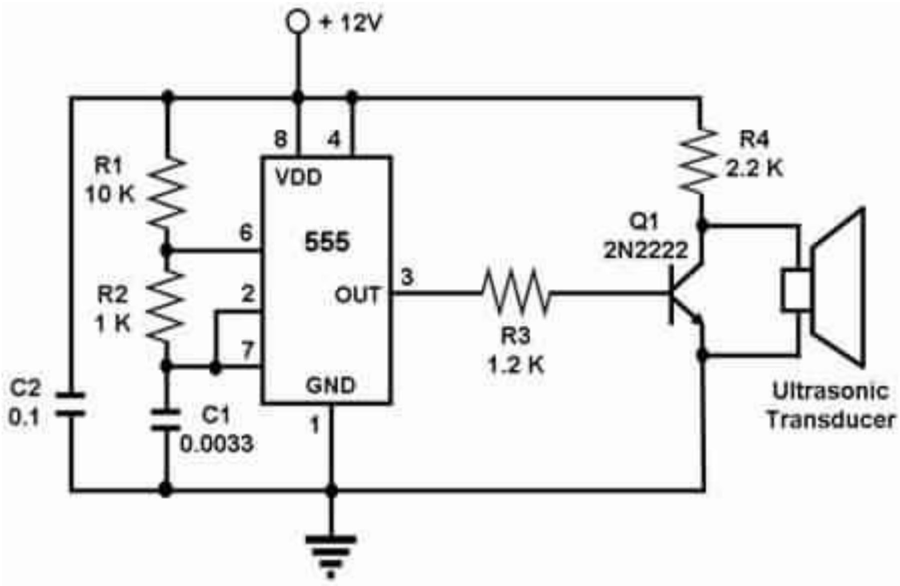



**Figure 15**

The circuit of an ultrasonic transmitter and receiver uses an IC CD4017 decade counter. The transmitter circuit is designed with D- flip flop IC and two-decade counter ICs and a few major components. The circuit design generates stable 40kHz signals, which are transferred by the transducer. The RF oscillator built with transistor T1 to produce an 8MHz signal, that works as an i/p to the first decade counter that is built around IC1. The oscillator frequency can be divided to 800 kHz by the decade counter.

The o/p of the first decade counter is fed to the second decade counter, then the frequency can be divided to 80 kHz. The flip flop divides the frequency to half of the frequency of decade counter, then it is transferred by the ultrasonic transducer. The L coil is made with enameled copper wire 36SWG that is looped 15 times around an 8 millimeter diameter of plastic former, which has a ferrite rod and this is used for radio oscillators.

**Features**

Detecting range: 3cm-4m

Best in 30 degree angle

Electronic brick compatible interface

5VDC power supply

Breadboard friendly

Dual transducer

Arduino library ready

**Specifications**

- Working Voltage: DC 5V
- Working Current: 15mA

- Working Frequency: 40Hz
- Max Range: 4m
- Min Range: 2cm
- Measuring Angle: 15 degree
- Trigger Input Signal: 10µS TTL pulse
- Echo Output Signal Input TTL lever signal and the range in proportion
- Dimension 45 * 20 * 15mm

### 3.1.4 Chassis

A **vehicle frame**, also known as its *chassis*, is the main supporting structure of a motor vehicle, to which all other components are attached, comparable to the skeleton of an organism.

Until the 1930s virtually every car had a structural frame, separate from its body. This construction design is known as *body-on-frame*. Over time, nearly all passenger cars have migrated to unibody construction, meaning their chassis and bodywork have been integrated into one another.

In the case of vehicles, the term rolling chassis means the frame plus the "running gear" like engine, transmission, drive shaft, differential, and suspension.

An under body (sometimes referred to as "coachwork"), which is usually not necessary for integrity of the structure, is built on the chassis to complete the vehicle.

For commercial vehicles, a rolling chassis consists of an assembly of all the essential parts of a truck (without the body) to be ready for operation on the road.[3] The design of a pleasure car chassis will be different from one for commercial vehicles because of the heavier loads and constant work use.[4] Commercial vehicle manufacturers sell "chassis only", "cowl and chassis", as well as "chassis cab" versions that can be outfitted with specialized bodies. These include motor homes, fire engines, ambulances, box trucks, etc.

In particular applications, such as school buses, a government agency like National Highway Traffic Safety Administration (NHTSA) in the U.S. defines the design standards of chassis and body conversions.[5]

An armoured fighting vehicle's hull[6] serves as the chassis and comprises the bottom part of the AFV that includes the tracks, engine, driver's seat, and crew compartment. This describes the lower hull, although common usage might include the upper hull to mean the AFV without the turret. The hull serves as a basis for platforms on tanks, armoured personnel carriers, combat engineering vehicles, etc.

Function

The main functions of a frame in motor vehicles are:

1. To support the vehicle's mechanical components and body
2. To deal with static and dynamic loads, without undue deflection or distortion.

   These include:

- Weight of the body, passengers, and cargo loads.
- Vertical and torsional twisting transmitted by going over uneven surfaces.
- Transverse lateral forces caused by road conditions, side wind, and steering the vehicle.
- Torque from the engine and transmission.
- Longitudinal tensile forces from starting and acceleration, as well as compression from braking.
- Sudden impacts from collisions.

Types of frame according to the construction:

- Ladder type frame
- X-Type frame
- Off set frame
- Off set with cross member frame
- Perimeter Frame

### 3.1.5 L293D Motor Driver

SunFounder L293D is a monolithic integrated, high voltage, high current, 4-channel driver. Basically this means using this chip you can use DC motors and power supplies of up to 16 Volts, thats some pretty big motors and the chip can supply a maximum current of 600mA per channel, the L293D chip is also what's known as a type of H-Bridge. The H-Bridge is typically an electrical circuit that enables a voltage to be applied across a load in either direction to an output, e.g. motor.

The L293D is quadruple high-current half-H drivers.It is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo- Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

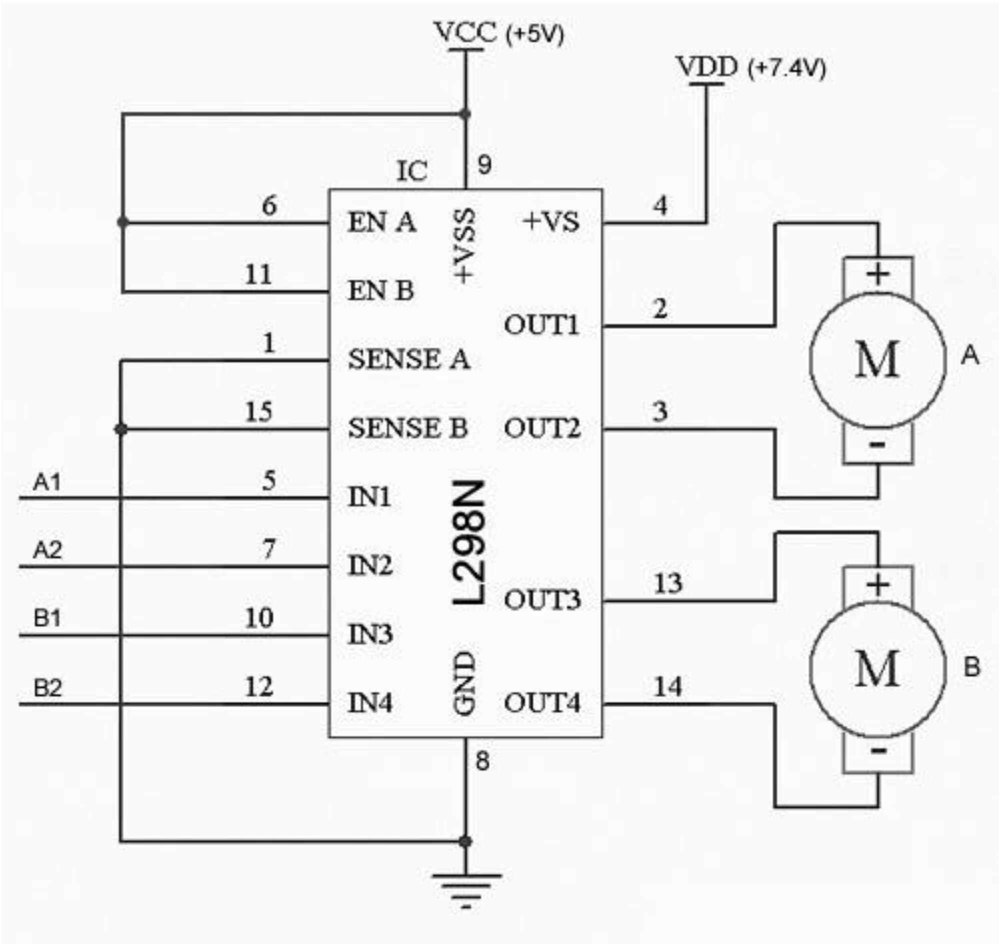

VCC (+5V)
VDD (+7.4V)
IC
9
4
6
EN A
+VSS
+VS
11
EN B
OUT1
2
1
SENSE A
M
A
15
SENSE B
OUT2
3
A1
5
IN1
A2
7
IN2
L298N
B1
10
IN3
OUT3
13
B2
12
IN4
GND
OUT4
14
M
B
8

**Figure 16**

## PIN FUNCTION

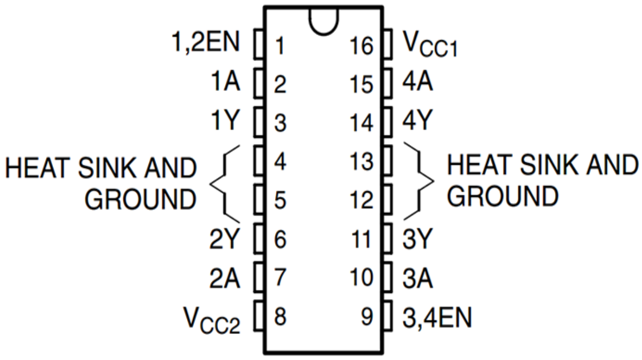



**Figure 17**

| **Pin** | **Name** | **Function** |
|---|---|---|
| 1 | Enable1,2 | Enable pin to control 1,2 driver |
| 2 | Input 1A | Input to control 1Y |
| 3 | Output 1Y | Output,connect to motor |
| 4 | GND | Ground and heat sink |
| 5 | GND | Ground and heat sink |
| 6 | Output 2Y | Output,connect to motor |
| 7 | Input 2A | Input to control 2Y |
| 8 | Vcc2 | Output supply voltage |
| 9 | Enable3,4 | Enable pin to control 3,4 |

| | | |
|---|---|---|
| | | driver |
| 10 | Input 3A | Input to control 3Y |
| 11 | Output 3Y | Output,connect to motor |
| 12 | GND | Ground and heat sink |
| 13 | GND | Ground and heat sink |
| 14 | Output 4Y | Output, connect to motor |
| 15 | Input 4A | Input to control 4Y |
| 16 | Vcc1 | Supply voltage(7 max) |

**Table-5**

Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Output Current 600 mA Per Channel
- Peak Output Current 1.2 A Per Channel

### Motors-60rpm

60RPM Centre Shaft Economy Series DC Motor is high-quality low-cost DC geared motor. It has steel gears and pinions to ensure longer life and better wear and tear properties. The gears are fixed on hardened steel spindles polished to a mirror finish. The output shaft rotates in a plastic bushing. The whole assembly is covered with a plastic ring. Gearbox is sealed and lubricated with lithium grease and require no maintenance. The motor is screwed to the gear box from inside.

Although motor gives 60 RPM at 12V but motor runs smoothly from 4V to 12V and gives wide range of RPM, and torque. Tables below gives fairly good idea of the motor’s performance in terms of RPM and no load current as a function of voltage and stall torque, stall current as a function of voltage.

For compatible wheels refer to Wheels and Accessories product category.

You can also mount this motor on the chassis using Motor Mount for Centre Shaft Economy Series DC Motor

For adding Position Encoder, refer to Encoder Kit for Centre Shaft Economy Series DC Motor

**Specifications**

- DC supply: 4 to 12V
- RPM: 60 at 12V
- Total length: 46mm
- Motor diameter: 36mm
- Motor length: 25mm
- Brush type: Precious metal
- Gear head diameter: 37mm
- Gear head length: 21mm
- Output shaft: Centred
- Shaft diameter: 6mm
- Shaft length: 22mm
- Gear assembly: Spur
- Motor weight: 100gms

**Inside view of Centre Shaft Economy Series DC Motor**

**Figure 18**

**Motor Mounting Clamp and Position Encoder Kit for Centre Shaft Economy Series DC Motor**

**Figure 19**

**Motor performance in terms of RPM and no load current as a function of input voltage**

| Voltage (V) | RPM (No Load) | Current (A) |
|---|---|---|
| 4 | 16 | 0.013 |
| 5 | 21 | 0.014 |
| 6 | 28 | 0.014 |
| 7 | 36 | 0.016 |

**Motor performance in terms of stall torque and stall current as a function of input voltage**

| Voltage (V) | Stall torque (Kg/cm) | Stall Current (A) |
|---|---|---|
| 4 | 3.440 | 0.403 |
| 5 | 4.020 | 0.516 |
| 6 | 4.966 | 0.618 |
| 7 | 5.762 | 0.715 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | 42 | 0.0 | | 8 | 6.772 | 0.820 |
| 9 | 47 | 0.019 | | 9 | 7.912 | 0.922 |
| 10 | 52 | 0.020 | | 10 | 8.557 | 1.006 |
| 11 | 59 | 0.022 | | 11 | 9.030 | 1.115 |
| 12 | 64 | 0.024 | | 12 | 9.352 | 1.203 |

**Note: Motors's data can vary by ±10%**

**Table-6**

## CHAPTER-4

### 4.1 Raspbian OS

Raspbian is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie. Since 2015 it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the family of Raspberry Pi single-board computers.[1] Raspbian was created by Mike Thompson and Peter Green as an independent project.[4] The initial build was completed in June 2012.[5] The operating system is still under active development. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.

Raspbian uses PIXEL, **P**i **I**mproved **X**-Window **E**nvironment, **L**ightweight as its main desktop environment as of the latest update. It is composed of a modified LXDE desktop environment and the Openbox stacking window manager with a new theme and few other changes. The distribution is shipped with a copy of computer algebra program Mathematica and a version of Minecraft called Minecraft Pi[6] as well as a lightweight version of Chromium as of the latest version.

Pi was introduced as an educational gadget to be used for protyping by hobbyists and for those who want to learn more about programming. It certainly cannot be a substitute for our day to day Linux, Mac or Windows PC.

Pi is based on a Broadcom SoC (System of Chip) with an ARM processor [~700 MHz], a GPU and 256 to 512 MB RAM. The boot media is an SD card [which is not included], and the SD card can also be used for persist data. Now that you know that the RAM and processing power are not nearly close to the power house machines you might have at home, these Pi's can be used as a Cheap computer for some basic functions, especially for experiments and education. The Pi comes in three Configurations and we will discuss the specifications of those in the coming sections. The cost of a Pi is around $35 for a B Model and is available through many online and physical stores.Below are the basic things you would need to get started with using a Pi.

| | |
|---|---|
| Computer | A Raspberry Pi |
| Storage | SD Card and a SD card reader to image the OS [These days laptops have inbuilt card readers] |
| Power supply | 5 volt micro USB adapter, mostly your android phone charger would work |
| Display | An TV/Monitor with DVI or HDMI port |
| Display connector | HDMI cable or HDMI to DVI converter cable |
| Input | USB Mouse |
| Input | USB Keyboard |
| Network | Ethernet cable |
| Case | If you really need one, you can get them online based on the model you have |

Table-7

Raspberry Pi system specifications

As discussed earlier the Pi comes in three configurations. Below is a table that gives the details about all three models namely A, B and B+.

| Description | Model A | Model B | Model B+ |
|---|---|---|---|
| Chip | Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, and single USB port) | | |
| Processor | 700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set) | | |
| RAM | 256 MB | 512 MB | 512 MB |
| USB | 1 (direct from BCM2835 chip) | 2 on board | 4 on board |
| Storage | SD Card | SD Card | MicroSD card |
| Voltage | 600mA upto 1.2A @ 5V | 750mA upto 1.2A @ 5V | 600mA upto 1.8A @ 5V |
| GPO | 26 | 26 | 40 |

**Table-8**

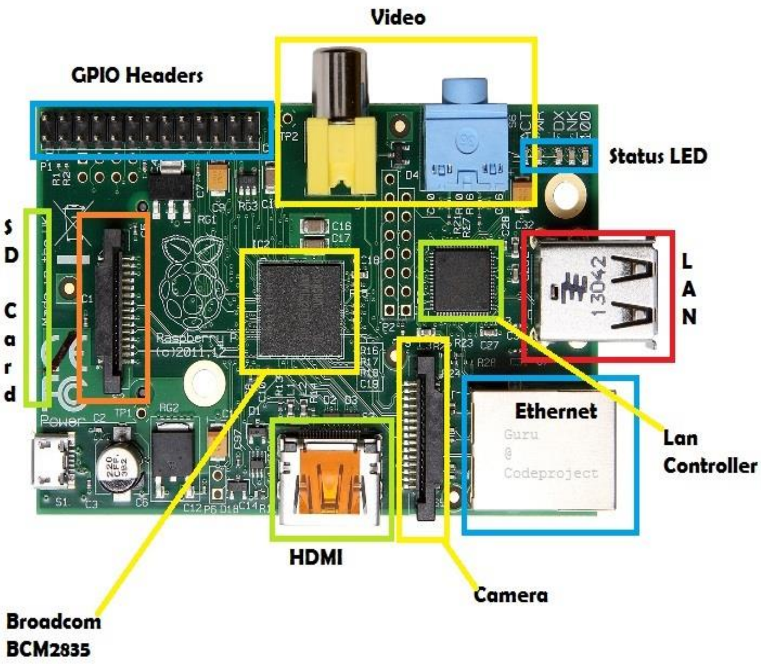



**Figure 20**

## Ports, Pins and their uses

Below are the ports on the Raspberry Pi board and some of their uses. The ports may also be used for other purposes than listed below.

| | |
|---|---|
| USB | Mainly used for pheriperals like Keyboard, mouse and a Wifi Adapter. A powered USB hub can be connected and be expanded |
| HDMI | This is the High Definition Multimedia Interface [HDMI] and is use to connect to a Display unit like TV or Monitor or sometimes a projector |
| Stereo Audio | Audio connections using a 3.5 mm jack |
| SD Card | SD card is used as a boot device and also persistent storage. More stoage can be attached to the USB |
| Micro USB | The micro usb port is used for supplying power to the unit |
| CSI Connector | CSI [ Camera serial Interface] is used for connecting a camera to the unit |
| Ethernet | Used for connecting to a network using a network cable |
| DSI Connector | DSI [ Digital serial Interface] is used for connecting aLCD to the unit |

**Table-9**

One other important pin is the GPIO.

GPIO stands for General Purpose Input and Output.

There are 26 Pins on a Model B in total.

- There are three power supply pins [3.3v, 5.0v and 0v].
- 6 Do not Connect (DNC) pings
- 17 GPIO pins

We shall build a simple circuit towards the end of this tutorial to understand more about how to use these GPIO. Before that let's get our Pi setup with an Operating system.

## Raspberry Pi with Raspian OS

Below are some of the Operating systems that a Pi can run but in this article we will only learn about Raspian.

| | |
|---|---|
| Linux | There are three official Linux flavors available for download namely |
| | Debian [Raspbian] *Recommended |
| | ArchLinux |
| | Pidora [Based on Fedora] |
| RISC OS | A retro looking 1080p GUI designed by the ARM designers. RISC was more common during the 90's |
| Firefox OS | A new OS by the Firefox team. Pretty much a combination of Firefox and PTXdist-built Linux |
| Plan 9 | Unix like OS by the by the Bell Labs, created by the UNIX creators |
| Android | No explanation necessary, but this hasn't gone beyond a 2.3 build and a bit too slow. |

**Table-10**

Raspbian OS is one of the official Operating systems available for free to download and use. The system is based on Debian Linux and is optimized to work efficiently with the Raspberry Pi computer. As we already know an OS is a set of basic programs and utilities that runs on a specified hardware, in this case the Pi. Debian is very lightweight and makes a great choice for the Pi. The Raspbian includes tools for browsing, python programming and a GUI desktop.

The Raspbian desktop environment is known as the "Lightweight X11 Desktop Environment" or in short LXDE. This has a fairly attractive user interface that is built using the X Window System software and is a familiar point and click interface. We shall look more into how to install and use this OS in the next section.

### 4.2 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

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

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

Features and

python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogrammingand metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.[46]

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

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

The language's core philosophy is summarized in the document *The Zen of Python* (*PEP 20*), which includes aphorisms such as:[49]

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

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

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

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

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

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

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

**Uses:**

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

Large organiations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify and some smaller entities like ILM and ITA. The social news networking site Reddit is written entirely in Python.

Python can serve as a scripting language for web applications, e.g., via mod_wsgi for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitatetheseapplications. Webframeworks like Django, Pylons, Pyramid, TurboGears, web2 py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used as data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.

Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a mathematical software with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus.

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro,[138] and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go. Python is also used in algorithmic trading and quantitative finance. Python can also be implemented in APIs of online brokerages that run on other languages by using wrappers.

Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

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

Python is used extensively in the information security industry, including in exploit development.

### 4.3 RPi.GPIO Python Library:

Stands for "General Purpose Input/Output." GPIO is a type of pin found on an integrated circuit that does not have a specific function. While most pins have a dedicated purpose, such as sending a signal to a certain component, the function of a GPIO pin is customizable and can be controlled by software.

Not all chips have GPIO pins, but they are commonly found on multifunction chips, such as those used in power managers and audio/video cards. They are also used by system-on-chip (SOC) circuits, which include a processor, memory, and external interfaces all on a single chip. GPIO pins allow these chips to be configured for different purposes and work with several types of components.

A popular device that makes use of GPIO pins is the Raspberry Pi, a single-board computer designed for hobbyists and educational purposes. It includes a row of GPIO pins along the edge of the board that provide the interface between the Raspberry Pi and other components. These pins act as switches that output 3.3 volts when set to HIGH and no voltage when set to LOW. You can connect a device to specific GPIO pins and control it with a software program. For example, you can wire an LED to a GPIO and a ground pin on a Raspberry Pi. If a software program tells the GPIO pin to turn on, the LED will light up.

Most computer users will not encounter GPIO pins and do not need to worry about configuring them. However, if you are a hobbyist or computer programmer, it can be helpful to learn what chips have GPIO pins and how to make use of them.

RPIO is an advanced GPIO module for the Raspberry Pi.

- PWM via DMA (up to 1µs resolution)
- GPIO input and output (drop-in replacement for RPi.GPIO)
- GPIO interrupts (callbacks when events occur on input gpios)

- TCP socket interrupts (callbacks when tcp socket clients send data)
- Command-line tools rpio and rpio-curses
- Well documented, fast source code with minimal CPU usage
- Open source (LGPLv3+)

RPIO consists of two main components:

- *RPIO* – Python modules which you can import in Python 2 or 3 with import RPIO, import RPIO.PWM, etc.
- *rpio* – command-line tools for inspecting and manipulating GPIOs system-wide.

Documentation:

- *RPIO* – Python modules which you can import in Python 2 or 3 with import RPIO, import RPIO.PWM, etc.
- *rpio* – command-line tools for inspecting and manipulating GPIOs system-wide.

The easiest way to install/update RPIO on a Raspberry Pi is with either easy_install or pip:

- $ sudo apt-get install python-setuptools
- $ sudo easy_install -U RPIO

You can also get RPIO from Github repository, which is usually a step ahead of pypi:

- $ git clone https://github.com/metachris/RPIO.git
- $ cd RPIO
- $ sudo python setup.py install

Or from Github but without Git:

- $ curl -L https://github.com/metachris/RPIO/archive/master.tar.gz | tar -xz
- $ cd RPIO-master
- $ sudo python setup.py install

Debian packages are available at metachris.github.com/rpio/download.

After the installation you can use import RPIO as well as the command-line tool rpio.

### 4.4 OpenCV

**OpenCV** (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

OpenCV supports the deep learning frameworks TensorFlow, Torch/ PyTorch and Caffe

OpenCV's application areas include:

- 2D and 3D feature toolkits
- Ego motion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking
- Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

- Boosting
- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbour algorithm
- Naive Bayes classifier
- Artificial neural networks
- Random forest
- Support vector machine (SVM)
- Deep neural networks (DNN)

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, Haskell, and Ruby have been developed to encourage adoption by a wider audience.

Since version 3.4, **OpenCV.js** is a JavaScript binding for selected subset of OpenCV functions for the web platform.

All of the new developments and algorithms in OpenCV are now developed in the C++ interface.

OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV runs on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10. The user can get official releases from Source Forge or take the latest sources from GitHub. OpenCV uses C Make.

## CHAPTER-5

### 5.1 LANE DETECTION

Lane Detection Algorithm Traditionally, lane could be detected by two approaches namely feature based technique and model based technique. The feature based technique localizes the lanes in the road images by combining the low-level features, such as painted lines or lane edges etc. Accordingly, this technique requires well studied road having well-painted lines or strong lane edges, otherwise it will fail. Moreover, it has the disadvantage of not imposing any global constraints on the lane edge shapes, this technique may suffer from occlusion or noise.

On the other hand, the model-based technique just uses a few parameters to represent the lanes. Assuming the shapes of lane can be presented by either straight line or parabolic curve, the processing of detecting lanes is approached as the processing of calculating those model parameters. This way, the model-based technique is much more robust against noise and missing data, compared with the feature-based technique. To estimate the parameters of lane model, the likelihood function, Hough transform, and the chi-square fitting, etc. are applied into the lane detection. However, as the most lane models are only focused on certain shapes of road, thus they lack the flexibility to modelling the arbitrary shape of road.

In the proposed algorithm to detect the lanes, a combination of feature and model base is used. In general, this algorithm is valid for all kind of roads (whether they are marked with white lanes or not).

**Application:**

- Digital camera image
- Computer graphics
- Computer vision
- Digitizing
- Free boundary condition
- GPGPU
- Homomorphic filtering
- Image analysis
- IEEE Intelligent Transportation Systems Society
- Multidimensional systems
- Remote sensing software
- Standard test image
- Super resolution

**5.2 The overall method consists of 7 major parts:**

5.2.1 Extract the colour range for the road Extract the appropriate upper and lower range to determine the colour of the portion on which the car is standing. This is the primary and most important part of this algorithm. Using this range, a binary image of the current view is created.

5.2.2 Define the region of interest A region of interest is defined starting from the bottom towards upward. As the view is taken from the camera on the car, the road surface closest to the car is at the bottom of the image. Thus the region of interest is defined from the nearest region to the farther distances by moving upwards in the image created. The height of the region of interest is usually not more than half the height of image. This removes the sky area from the visual field and saves unnecessary computations and is better than the method proposed by Dahlkemper. As we move away from the car, the width of the road seems to be narrowing. So, the actual region of interest is in a shape of a trapezium.

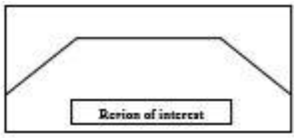



**Figure 21**

5.2.3 Convert complex region of interest into simple shape.

This is an important step in determining the boundary of the road. In the region of interest, the contours are determined. Since the car is on the road, the largest contour which contains the central part of the bottom region is the road. Simplify the shape of this contour using approximations.

5.2.4 Determine the shape of the road.

We draw Hough lines on the manipulated contour. A number of Hough lines are obtained along the left and right edges of the road. Out of these, only few lines represent the actual edge. Rests of the lines are due to the noise (due to irregularities in the road) along the edge.

5.2.5 Filtering the noise.

The lines along the left edge of the road are tilted towards right and vice versa. So, any line which is tilted towards left and lies entirely in the left half the image is discarded. Similar lines are also discarded from the right half.For the left edge, a line with the smallest angle or having the least positive x intercept or y intercept is chosen as the left edge of the road. Similarly, find the right edge of the road. A general case could be represented as shown in Fig 20.

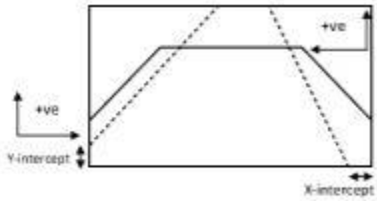



**Figure 22**

5.1.6 Make it robust and unaffected from noise.

For the moving car, the change in the direction of road cannot be abrupt. The edges of the road cannot change the angle in discrete fashion. So, any line which is far away from the line in previous frame is simply discarded. Another factor called "tolerance" is considered. Basically, it is the count of the continuous frames which could be accepted without being able to determine the edge in entire above-mentioned process. Its maximum value is 3. If we are not able to determine the edge of the road, tolerance value is decremented by 1. If it reaches 0, we retry to find the new colour range of the road at the run time [6, 7].

5.1.7 Determine the turns in the road, if any, and give directions to the car.

The different cases of the road (left turn, right turn, divergence, round - about etc.) are considered. Suppose there is a left turn as shown in Fig 6. Divide the region of interest in 3 parts in ratio 2:3:5. Compare the lines obtained in all three parts with the possible shape of the road. The dotted lines in each section in figure 20 show the lines that are obtained applying Hough lines in these sections separately.

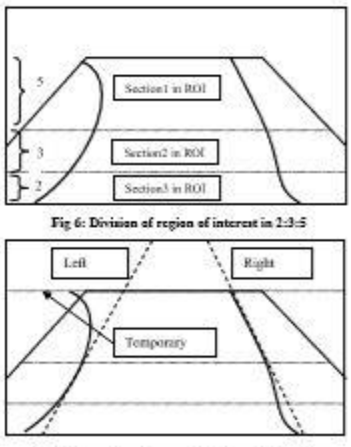



**Figure 23**

Lines making same angle with the x axis and lying in either all three sections or two sections namely top and middle or even only one section is its middle or bottom section are chosen and predicated to be the edges of the road shown in figure 7. Depending upon the continuation or break of same line in 3 different sections, the final ROI is modified which further assists to decide the turn present on the road. The above fig 6 shows a road that has a left turn ahead. Applying the steps mentioned above,the three parts of the region of interest are divided as shown. Finding the left and right edges in these different sections can be used to find the possible edges of the road as is depicted by two dotted lines connecting the bottom of the ROI to the top (Fig 7).

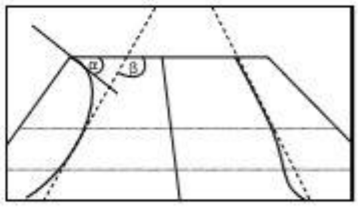
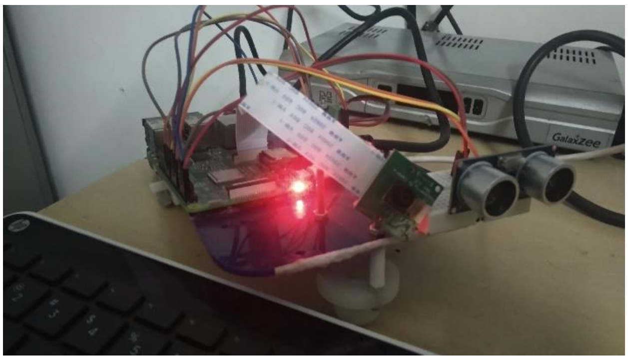


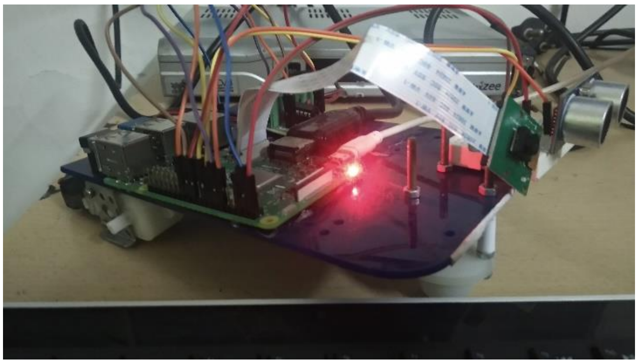

**Figure 24**

Next challenge lies in determining the turns and changing the line of motion of the car. A temporary line, as shown in fig 7, overlapping the top edge of the ROI from left to right is drawn. A tangent to the contour is also drawn at the intersection point of the temporary line with contour. If the intersection point lies outside the ROI, the ROI is modified to accommodate this point within itself. This could be seen in Fig 8, where the top point of the left edge of the road is within the ROI which was lying outside the ROI in Fig 7. In Fig 8, the tangent is making an angle α with the top of ROI. And β is the angle between the determined left edge and top of ROI. For the left lane these angles are different. Similarly, these angles are calculated for right lane and are found same. These parameters are used to come to a conclusion that there is a turn ahead and that turn is a left turn. After this step there is a change in line of motion of the car depending upon the new line obtained by joining the mid points of the top and bottom line of the modified ROI.

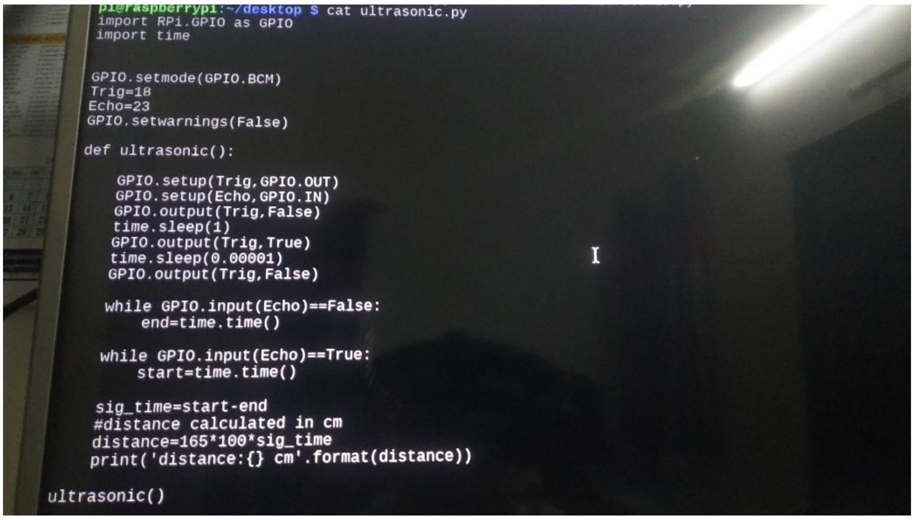
## CHAPTER-6

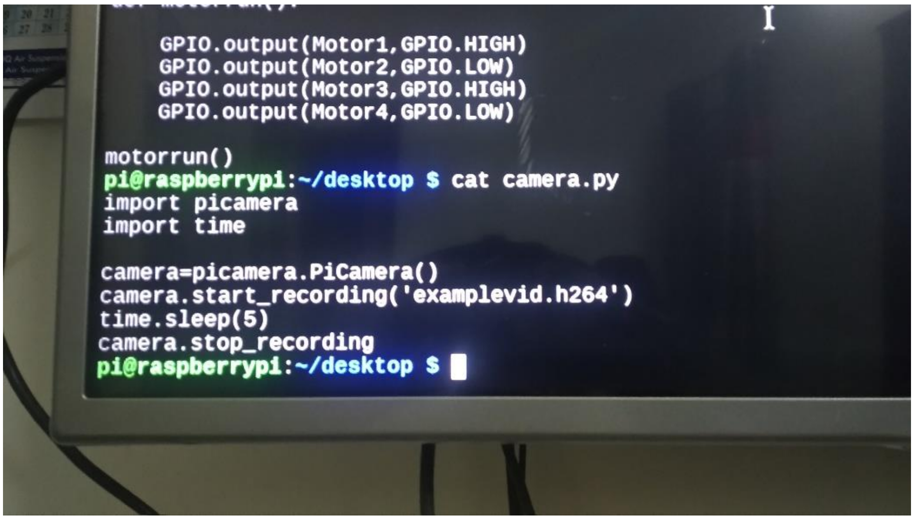
### 6.1 RESULT

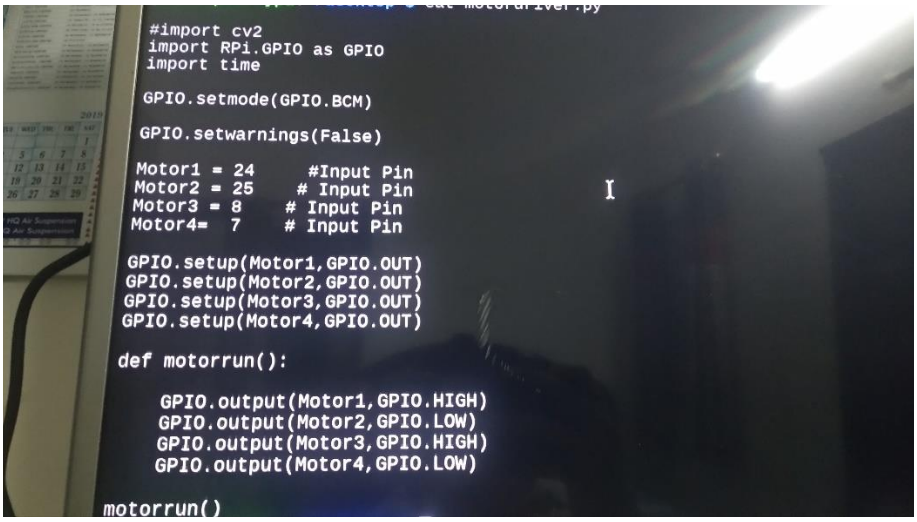




Bot Pictures






Program Pictures

### 6.2 CONCLUSION

In this report, a method to make a self-driving robot car is presented. The different hardware components and their assembly are clearly described. A novel method to determine the uneven, marked or unmarked road edges is explained in details relying upon OpenCV. Using ultrasonic sensors, the collisions with obstacles is avoided. The algorithm mentioned in the paper has been successfully implemented on a small autonomous car.

### 6.3 FUTURE WORK

The work could be enhanced by improving the algorithm by adding machine learning to it. The present algorithm performs the operations on all the frames. It is accurate but its efficiency could be further enhanced if it starts learning by itself and avoid unnecessary calculations of the regions which are already known or familiar. Once the car starts travelling on the roads, it determines the obstacles (mainly static) on the way and note their characteristic features. An XML file is generated every time the car travels. It stores the following information:

- The distance between the two nodes.
- The number of roads diverging from a particular node.
- The number of speed breakers and other static obstacles on the road joining two nodes.
- The distance of speed breakers and other static obstacles from a specific node.

- Height and the width of an obstacle. The information stored in the XML helps the car understand and remember the path being followed for the next time.

## REFERENCES


[1] Johann Borenstein & Yoram Koren, Obstacle Avoidance with Ultrasonic Sensors, IEEE JOURNAL OF ROBOTICS AND AUTOMATION, VOL. 4, NO. 2, APRIL I988, pp. 213-218

[2] Yue Wanga, Eam Khwang Teoha & Dinggang Shenb, Lane detection and tracking using B-Snake, Image and Vision Computing 22 (2004) , available at:www.elseviercomputerscience.com, pp. 269–280.

[3] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. G. Sukhatme, S. Schaal, W. Burgard, and D. Fox, editors& Proceedings of the Robotics Science and Systems Conference, Philadelphia, PA, 2006.

[4] Joel C. McCall &Mohan M. Trivedi, Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation, IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 1, March 2006, pp. 20-37.

[5] Tushar Wankhade & Pranav Shriwas, Design of Lane Detecting and Following Autonomous Robot, IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 2, Issue 2 (July-Aug. 2012), pp. 4548.

[6] Xiaodong Miao, Shunming Li & Huan Shen, On-Board lane detection system for intelligent vehicle based on monocular vision, International Journal on Smart Sensing and Intelligent Systems, vol. 5, no. 4, December 2012, pp. 957-972.

[7] A. Bar Hillel, R. Lerner, D. Levi, & G. Raz. Recent progress in road and lane detection: a survey. Machine Vision and Applications, Feb. 2012, pp. 727–745

[8] Narathip Thongpan, & Mahasak Ketcham, The State of the Art in Development a Lane Detection for Embedded Systems Design, Conference on Advanced Computational Technologies & Creative Media (ICACTCM'2014) Aug. 14-15, 2014

[9] Narayan Pandharinath Pawar & Minakshee M. Patil, Driver Assistance System based on Raspberry Pi, International Journal of Computer Applications (0975 – 8887) Volume 95– No.16, June 2014, pp. 36-39.

[10] J.M.A. Alvarez, A.M. Lopez & R. Baldrich, IlluminantInvariant Model-Based Road Segmentation. Intelligent Transportation Systems, IEEE Transactions on, 12, 2008, pp 184–193.

[11] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, & P. Newman, Illumination Invariant Imaging: Applications in Robust Vision-based Localization, Mapping and Classification for Autonomous Vehicles, Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014.

[12] Neha A Ghaisas & R. R. Sedamkar, Inter-vehicular Communication using Packet Network Theory, IJRET: International Journal of Research in Engineering and Technology, ISSN: 2319-1163 | ISSN: 2321-7308, Volume: 03 Issue: 09 | Sep-2014, available at: http://www.ijret.org, pp. 72-78

[13] Trifan, A., Neves, A.J.R. & Cunha, B, Evaluation of color spaces for user-supervised color classification in robotic vision.17th International Conference on Image Processing, Computer Vision, & Pattern Recognition, Las Vegas, Nevada, USA (July 2013).

[14] R. Cucchiara, C. Grana, M.Piccardi& A. Prati, Detecting moving objects, ghosts, and shadows in video streams, IEEETransactions on Pattern Analysis and Machine Intelligence(PAMI), Vol. 25(10), 1337 - 1342, 2003.pp.25-28.

[15] S. Tuohy, D. O'Cualain, E. Jones, & M. Glavin, Distance determination for an automobile environment using inverse perspective mapping in OpenCV, in Proc. Irish Signals and Systems Conference 2010.

[16] Li, M., Zhao, C., Hou, Y. & Ren, M. , A New Lane Line Segmentation and Detection Method based on Inverse Perspective Mapping, International Journal of Digital Content Technology and its Applications. Volume 5, Number 4, April 2011, pp. 230-236

[17] Dhaval Chheda, Divyesh Darde & Shraddha Chitalia, Smart Projectors using Remote Controlled Raspberry Pi, International Journal of Computer Applications (0975 – 8887),Volume 82 – No. 16,2013, pp.

[18] Stewart Watkiss, Design and build a Raspberry Pi robot [Online], available at: http://www.penguintutor.com/electronics/robot/rubyrobo t-detailedguide.pdf

[19] David Hayward, Raspberry Pi operating systems: 5 reviewed and rated [Online], available at: http://www.in.techradar.com/news/software/

[20] Matt Richardson, Shawn Wallace, Getting Started with Raspberry Pi, 2nd Edition, Published by Maker Media, Inc., USA, 2014. Book ISBN: 978-1-457-18612-7.

[21] Gary Bradski, Adrian Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, "O'Reilly Media, Inc.". Copyright.September 2008, 1st Edition, Book ISBN: 978-0-596-51613-0

[22] Ch. Krishnaveni , Ms. A. Siresha , Mr. B. J. Prem Prasana Kumar & Mr. K. Sivanagireddy, Implementation of embedded systems for pedestrian safety using haar features, IJEC: International Journal of Electrical Electronics and Communication,ISN 2048 – 1068,Volume: 06 Issue: 20 l Oct -2014, pp. 761-766

[23] Tan-Hung Duong , Sun-Tae Chung , Seongwon Cho, Model-Based Robust Lane Detection for Driver Assistance, available at: http://www.kpubs.org/article/articleMain.kpubs?spotTyp e=low&articleANo=MTMDCW_2014_v17n6_655