|| Volume 3 || Issue 6 || June 2018 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

Pattern Recognition in Autonomous Vehicle using IoT

Tushar Bobade[1], Yash Naidu[2], Rushikesh Hire[3], Suraj Lad[4], Prof. Supriya Sarkar[4]
Department of Computer Science Engineering, SKNSITS Lonavala [1 2 3]
Department of Computer Science Engineering, SKNSITS Lonavala[4]

Abstract— **Recently, there have been significant advances in self-driving cars, which will play key roles in future intelligent transportation systems. In order for these cars to be successfully deployed on real roads, they must be able to autonomously drive along collision-free paths while obeying traffic laws. In contrast to many existing approaches that use prebuilt maps of roads and traffic signals, we propose algorithms and systems using Unified Map built with various onboard sensors to detect obstacles, other cars, traffic signs, and pedestrians. The proposed map contains not only the information on real obstacles nearby but also traffic signs and pedestrians as virtual obstacles. Using this map, the path planner can efficiently find paths free from collisions while obeying traffic laws.**

The proposed algorithms were implemented on a commercial vehicle and successfully validated in various environments, including the 2012 Hyundai Autonomous Ground Vehicle Competition

Keywords :*Autonomous Car, Smart Transportation, IOT, Path Detection, Driverless System.*

I INTRODUCTION

AUTONOMOUS driving technologies are expected to significantly improve driving safety and convenience by alleviating the burden of a driver, particularly under adverse conditions. Currently, they are implemented as a form of an advanced driver assistance system to partially aid drivers. It is also anticipated that, in the near future, fully autonomous cars will emerge as the key component of future transportation systems, replacing human drivers. In 2013, Mercedes Benz announced their plan to commercialize autonomous vehicles by 2020, and Nissan also announced their target year as 2020. Autonomous driving technology took a quantum leap, owing to the Defence Advanced Research Project Agency (DARPA) Grand Challenge held in 2005, which required autonomous vehicles to drive in a 367-km off-road course in a desert without any help from outside [1]–[3]. The DARPA urban challenge, which was held in 2007, evaluated autonomous navigation technologies for urban environments such as merging, intersection handling, parking lot navigation, and lane change. The speed of vehicles was limited to 48.2 km/h for safety, and a very detailed Route Network Definition File about the competition environment (e.g., lane markings, stop signs, and special check points, with a high-resolution aerial image of the area) was given to participants [4]–[7]. Google increased the chance of the commercialization of autonomous vehicles by developing Google cars, i.e., the first licensed robot driver [8]. The VisLab Intercontinental Autonomous Challenge (VIAC) highlighted the reliability of vehicle-following autonomous driving, without any prior knowledge of the course, during a 13 000-km intercontinental trip [9], [10]. Considering the long distance covered, this experiment exhibited great autonomous driving performance. Although the competitions and the experiment were performed in challenging environments, the developed vehicles function when detailed information about the driving environment was provided. Hence, autonomous driving in traffic scenarios such as traffic lights and crosswalks, without prior information of the course, remains a challenge. An autonomous vehicle should drive considering the overall situation, and many kinds of detection algorithms are necessary for autonomous driving in unpredictable real environments. In order to search for traffic information, we propose color based detectors for artificial markers on the road and an obstacle detector based on Light Detection and Ranging (LiDAR) sensors, and we implement pedestrian and vehicle detectors. However, the outputs of many different detectors complicate the decision-making process of the autonomous vehicle. To manage and consolidate the traffic information obtained from various detection algorithms, we also propose the Unified Map representation.

Unified Map converts the information of traffic environments into imaginary obstacles. Since this map represents the results from various algorithms as obstacles, the behaviour of the vehicle can be easily determined by a path planner with only local information. The path planner can be also easily applied to an autonomous driving system without complicated decision rules. The structure of Unified Map is similar to existing map representations [11]–[14]. Although these map representations only deal with the geometric information about the environment, our map representation additionally uses high-level information (e.g., traffic information) for the path planning of autonomous vehicles. We developed a test driving vehicle, i.e., EURECAR, to implement the proposed system for autonomous driving in unknown environments. In this paper, we describe the Unified Map representation, real-time algorithms for detecting traffic events, and the system architecture. This integrated system successfully completed the full course of the 2012 Autonomous Vehicle Competition (AVC), which was organized by the Hyundai–Kia Motor Group.

II LITERATURE REVIEW

TITLE: Vehicle Crash Sensors: Obstacle Detector.

A blind spot is a location that is invisible to the driver from the driving seat or with the use of the three mirrors. This spot is found at the vehicle sides or directly behind the vehicle. Vans, SUVs, trucks and higher vehicles tend to have larger blind spots. Blind spot causes numerous accidents, resulting in fatalities, injuries and property damage. Obstacle detection sensor, also known as parking systems, detects an obstacle in the blind spot and warns the driver before an accident occurs. This system operates by receiving or sending out high frequency sound waves. A microprocessor within a control module of the sensor registers sound waves that bounce off the surface of the obstacle. The sensor measures the type and length of the waves to interpret the size and distance of the obstacle. Large objects usually emit different sizes of energy wave.

TITLE: Obstacle Avoidance Robotic Vehicle Using Ultrasonic Sensor for Obstacle Detection.

Now day's many industries are using robots due to their high level of performance and reliability and which is a great help for human beings. The obstacle avoidance robotics is used for detecting obstacles and avoiding the collision. This is an autonomous robot. The design of obstacle avoidance robot requires the integration of many sensors according to their task.

TITLE: Collision Avoidance For Cognitive Automobiles Using A 3D PMD Camera.

Collision avoidance is one of the most important capabilities for autonomous vehicles. During driving, collisions must be avoided in all situations. With the availability of 3d cameras which rely on the time-of-flight principle, it is possible to get a very rich perception of the environment. This paper shows, how obstacles can be detected in the vehicle's surrounding using a 3d PMD-camera (photonic mixing device). The obstacle detection is composed of two separated steps. First, segmentation and a clustering of pixels takes place. Secondly, each group of pixels is analysed in order to decide whether it is an obstacle or not. The result of the detection is a list of obstacles which is then used for behaviour execution. The execution is done with a behaviour network and it generates recommendations for path planning.

III SYSTEM ARCHITECTURE

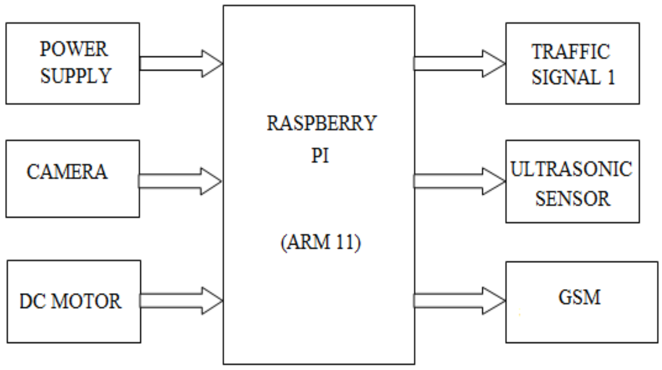


Figure 1 :Block Diagram

Specifications of Raspberry pi:

- Broadcom BCM2837 Arm7 Quad Core Processor powered Single Board Computer running at 900MHz
- 1GB RAM
- 40pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Micro USB power source.

Specifications of Raspberry pi:

- Broadcom BCM2837 Arm7 Quad Core Processor powered Single Board Computer running at 900MHz
- 1GB RAM
- 40pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Micro USB power source.

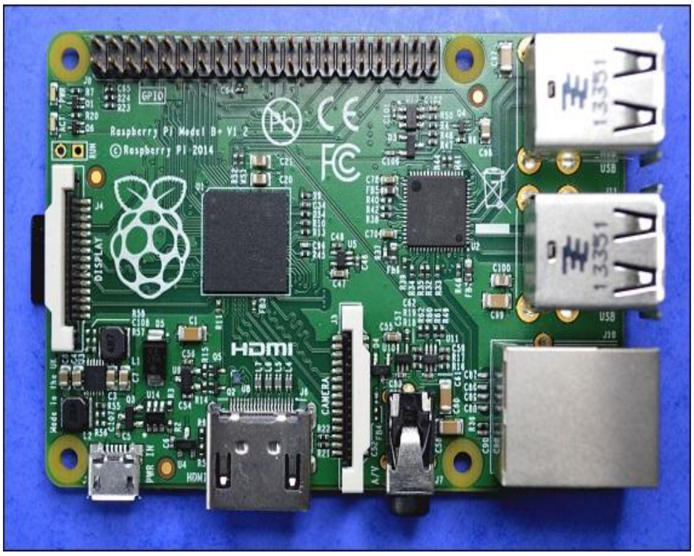
Features of Raspberry Pi:

- Broadcom BCM2837Arm7 Quad Core Processor powered Single Board Computer running at 900MHz
- 1GB RAM so you can now run bigger and more powerful applications
- Identical board layout and footprint as the Model B+, so all cases and 3rd party add-on boards designed for the Model B+ will be fully compatible.

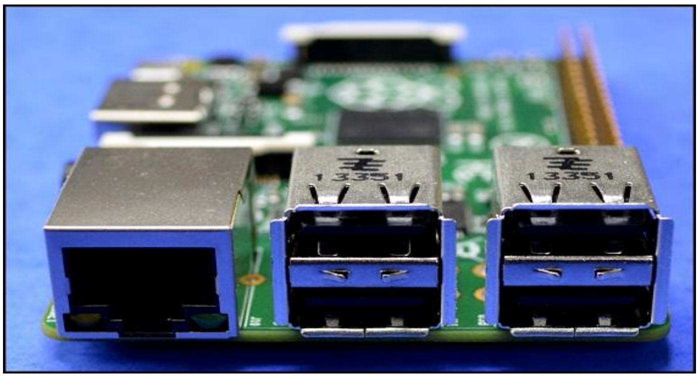
- Fully HAT compatible
- 40pin extended GPIO to enhance your "real world" projects. GPIO is 100% compatible with the Model B+ and A+ boards. First 26 pins are identical to the Model A and Model B boards to provide full backward compatibility across all boards.
- Connect a Raspberry Pi camera and touch screen display (each sold separately)
- Stream and watch Hi-definition video output at 1080P
- Micro SD slot for storing information and loading your operating systems.
- Advanced power management:
- You can now provide up to 1.2 AMP to the USB port – enabling you to connect more power hungry USB devices directly to the Raspberry PI. (This feature requires a 2Amp micro USB Power Supply)
- 10/100 Ethernet Port to quickly connect the Raspberry Pi to the Internet
- Combined 4-pole jack for connecting your stereo audio out and composite video out

Pin diagram:

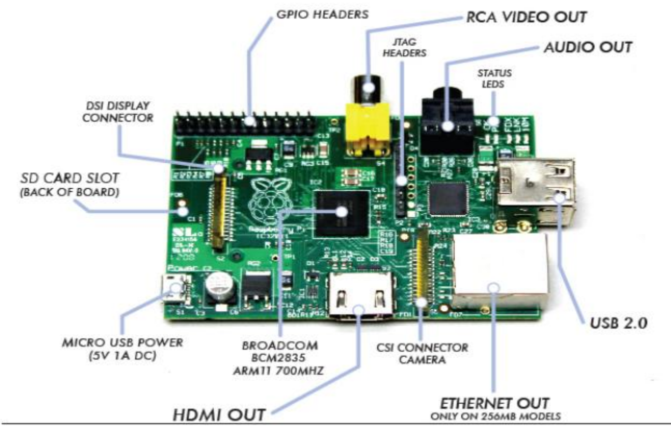
If you enjoy interfacing your Raspberry Pi with the real world by (for example) connecting sensors or controlling LEDs and motors, there will have been times when you have wanted to have more inputs and outputs on the GPIO header, or a couple more USB ports for connecting peripherals. Well, the good news is that the folks at the Raspberry Pi Foundation have obviously listened to the growing community of Raspberry Pi users and developed an enhanced version of the Raspberry Pi called the model B+.



On first sight the new Raspberry Pi model B+ looks quite different to previous models and indeed nearly all of the new features and enhancements introduced on the model B+ relate to connectivity and physical (as opposed to electronic) design.

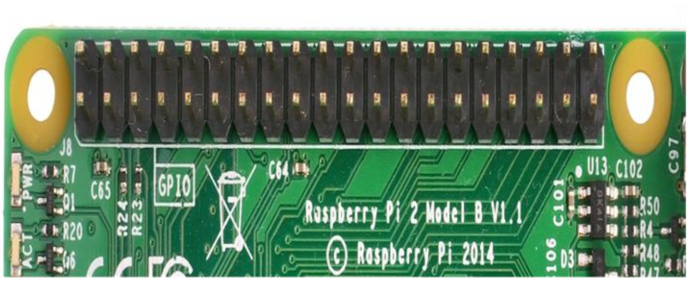


Two more USB ports have been provided which in many applications negate the need for an external USB hub. I often want to connect a keyboard, mouse and Wi-Fi adapter to my Raspberry Pi, and on the new Raspberry Pi B+ module I can do so without requiring a USB hub and still have one USB port free.



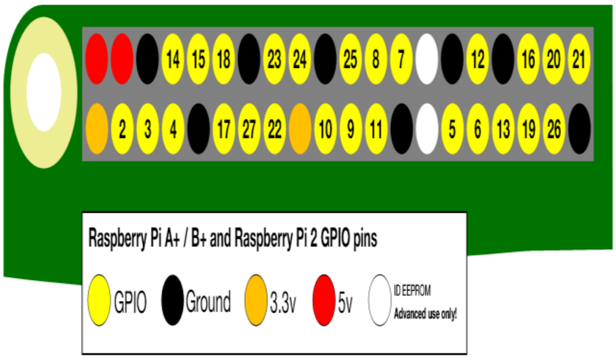
An introduction to gpio and physical computing on the raspberry pi.

One powerful feature of the Raspberry Pi is the row of GPIO (general purpose input/output) pins along the top edge of the board.



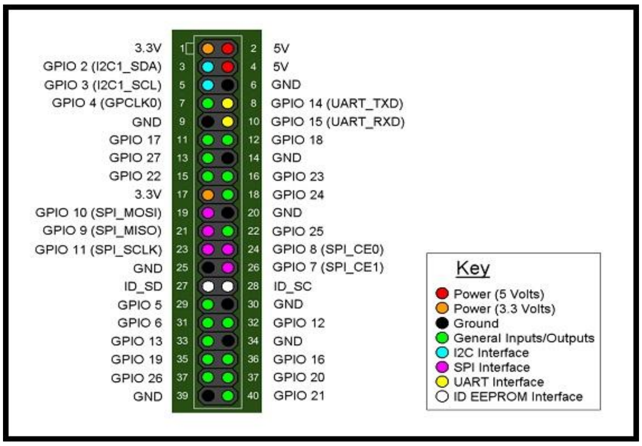
These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins which you should not play with unless you know your stuff!).

|| Volume 3 || Issue 6 || June 2018 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS



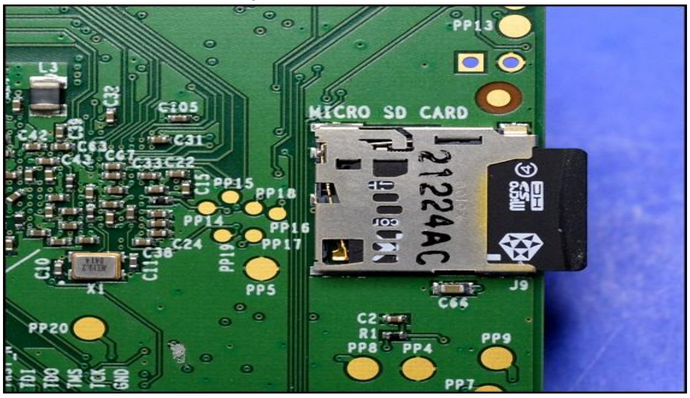
There are now a total of 40 GPIO pins, 26 of which can be used as digital inputs or outputs. Perhaps more importantly, 9 of the 14 new GPIO pins are dedicated inputs/outputs (pins which do not have an alternative function), so now if you want to use the onboard UART, I2C or SPI bus you can do so and still have plenty of free GPIO inputs and outputs to play with. The new expanded GPIO pin out is as shown below.

Pin out:



Pins 3 and 5 (GPIO 2 and GPIO 3) both have on board 1.8KOhm pull-up resistors fitted to them (and they also double up as the I2C interface pins). Pins 27 and 28 (ID_SD and ID_SC) are reserved exclusively for ID EEPROM use and cannot be used as input/output pins. The layout of the GPIO pins is backwards compatible with previous Raspberry Pi models – pins 1 to 26 are directly compatible with previous Raspberry Pi GPIO headers, although it should be noted that the whole GPIO header has been moved away from the corner of the board to allow room for an additional mounting hole - therefore any plug in board designed for previous Raspberry Pi models may be compatible, but will not sit directly above the Raspberry Pi B+ board because the GPIO header has been repositioned.

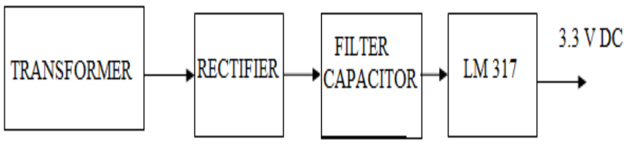
As far as other onboard connectors are concerned, the 3.5mm audio jack output socket and RCA composite video output socket (as found on previous Raspberry Pi models) have been replaced with a single 3.5mm 4-Pole A/V socket located next to the HDMI socket (which itself has been moved slightly on the PCB) and the power connector on the new Raspberry Pi model B+ has been relocated next to the HDMI socket. This means that all audio video and power connectors are now located along one side of the PCB which will help keep all connected cables tidy.



Amongst other changes introduced on the new Raspberry Pi model B+, the SD memory card slot used on previous Raspberry Pi models has been replaced with a Micro SD memory card slot, the status LEDs have been moved to the opposite end of the PCB and now consist only of a red “PWR” LED and a green “ACT” LED, and the PCB now has 4 mounting holes laid out in a rectangular pattern, which will make mounting the PCB securely so much easier.

However, please note the new Raspberry Pi model B+ uses the same CPU and GPU architecture as the model B and has the same 512MB of onboard SDRAM, so I’m afraid if you are expecting enhanced performance or computing power you may be disappointed. The only significant change in the onboard electronics is that linear power circuitry used on previous Raspberry Pi boards has been replaced with a more efficient (and significantly “beefed up”) switch mode design.

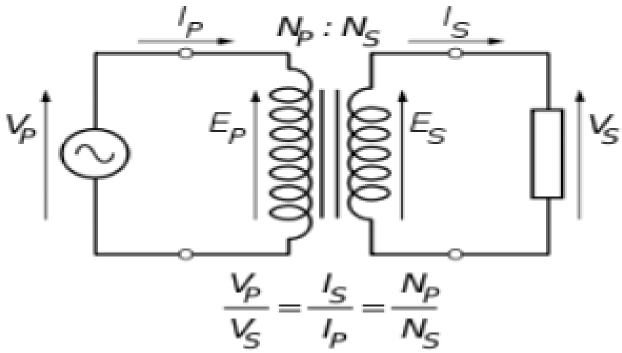
POWER SUPLLY (3.3 V):



One of the basic steps in the designing of any system is to design the power supply required for that system. The power supply is consisting of Transformer, Bridge Rectifier, Filter capacitor, and Voltage regulator IC.

1. TRANSFORMER

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors without changing its frequency. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a 34 varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction. If a load is connected to the secondary, an electric current will flow in the secondary winding and electrical energy will be transferred from the primary circuit through the transformer to the load.



Step-Down Transformer.

The voltage induced in the secondary is determined by the TURNS RATIO.

$$\frac{\text{Primary Voltage}}{\text{Secondary Voltage}} = \frac{\text{Number of Primary Turns}}{\text{Number of Secondary Turns}} \quad \text{.....} (6)$$

For example, if the secondary has half the primary turns; the secondary will have half the primary voltage 35. Another example is if the primary has 5000 turns and the secondary has 500 turns, then the turn's ratio is 10:1. If the primary voltage is 240 volts then the secondary voltage will be x 10 smaller = 24 volts.

Transformer Rating = 12V, 300mA.

Primary Voltage = 230V.

Secondary Voltage = 12V.

2. RECTIFIER

Rectifier is used to rectify the positive and negative half cycles of the output signal of the secondary of the transformer. So at the input of the rectifier. We have AC signal with both positive and negative cycles and at the output of the rectifier we have signal with only positive cycles. And at the rectifiers output we get Dc signal. There are mainly three types of rectifiers namely half wave, Full wave and Bridge rectifier. Out of these three we have used Bridge rectifier since it give more efficiency.

Reasons for choosing Bridge rectifier are :

a) The TUF is increased to 0.812 as compared the full wave rectifier.

b) The PIV across each diode is the peak voltage across the load $=V_m$, not $2V_m$ as in the two diode rectifier

Output of the bridge rectifier is not pure DC and contains some AC some AC ripples in it. To remove these ripples we have used capacitive filter, which smoothens the rippled output that we apply to 7805 regulators IC that gives 5V DC. We preferred to choose capacitor filters since it is cost effective, readily available and not too bulky.

1) FILTER CAPACITOR

As mentioned above we have to use filter capacitor to remove the AC signal from the output of rectifier. A capacitor is an electrical device that can store energy in the electric field between a pair of closely spaced conductors (called 'plates'). When voltage is applied to the capacitor, electric charges of equal magnitude, but opposite polarity, build up on the plate.

2) VOLTAGE REGULATOR

Voltage regulator is used after the filter capacitor so as to generate constant DC voltage supply of 3.3 volts.

LM 317 series:

The popular IC of 3 terminal regulators is 7805 series. The series 7805 is a series of 3- terminal positive voltage regulator. These ICs are provided with adequate heat sinking and can deliver output current more than 1A. These ICs do not require external components. These are provided with internal thermal protecting, overload and short circuit protection.

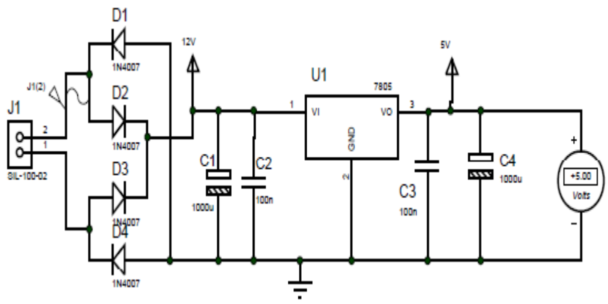
Features

- Output Current up to 1A.
- Output Voltages of 3.3 V.
- Thermal Overload Protection.
- Short Circuit Protection.
- Output Transistor Safe Operating Area Protection.

System specification

- Supply voltage of 5 V and current of 1.2 mA.
- Back Light voltage of 5 V and current 60 mA.

POWER SUPPLY CIRCUIT



IV HARDWARE AND SOFTWARE

HARDWARE SPECIFICATION

- ASPBERRY PI
- RANSISTOR
- IODE
- APACITORS
- SB CAMERA
- ED
- RAFFIC SIGNAL (R-Y-B)

|| Volume 3 || Issue 6 || June 2018 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

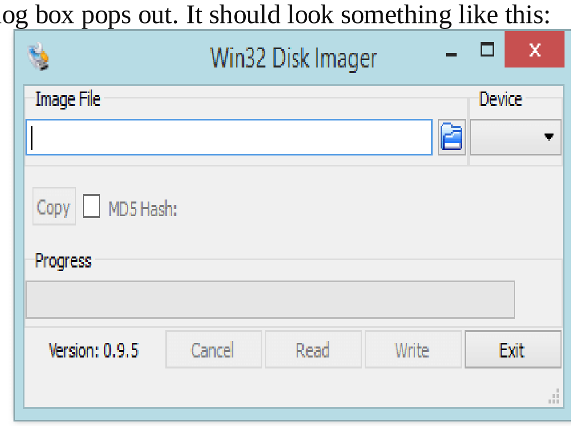
- C MOTOR
- CB
- IRES
- ULTRASONIC SENSOR

SOFTWARE SPECIFICATION:

Burning the OS Image

The Raspbian OS iso can be downloaded here or here. Extract the zip file after downloading.

Next, download & install the Win32DiskImager utility from the Sourceforge Project page here: http://sourceforge.net/projects/win32diskimager/. This software is to write the Raspbian OS into the MicroSD card. Once the installation in done, open up the program. Press yes when a dialog box pops out. It should look something like this:



Insert the MicroSD card into the MicroSD Adapter, and then into your computer. A drive name should appear in the Device combobox. After that, press the folder image & select the image file (ends with .iso) you extracted. Click *Write* and wait for the write to complete. Once done, eject the SD card & transfer the MicroSD into the Raspberry Pi 2.

Booting Up

Connect your Pi to a HDMI output & power source. As you turn on the switch, the Pi would boot up automatically. By default, the Pi would boot into terminal, but you can change this in the raspi-config screen. As this is the first time you up turning on the Pi, the screen will display the raspi-config screen for you to do your configurations. Select *Finnish* to exit the configuration screen.

Before doing anything, you will have to login first. But, what is my username & password?? By default, they are:

Username: pi

Password: raspberry

[Note: When you are typing your password, do not be alarmed when there is no text there. Linux does NOT display passwords!]

Once you have login, the terminal will display:

pi@raspberry ~ $

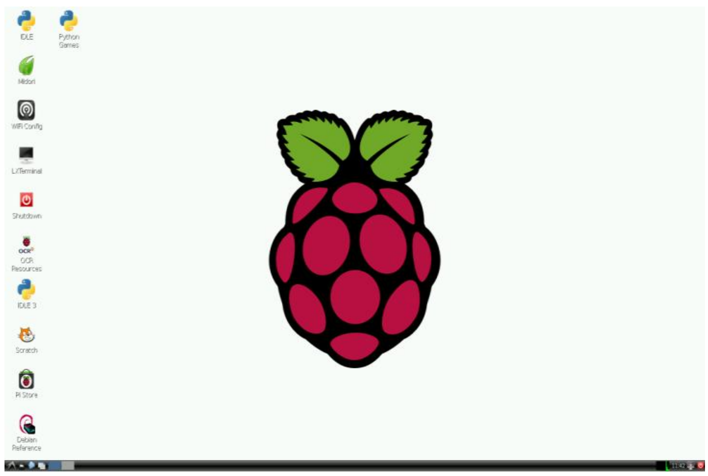
This means that the terminal is ready for the next command!

Starting the GUI

If you are not familiar with the Linux line, using the GUI (Graphical User Interface) will be a better option. But how? Just type *startx* into the command line. It should look like this:

pi@raspberry ~ $ startx

After that, the Raspberry Pi will exit the terminal view & enter the GUI. It will look something like this:



ULTRASONC SENSOR:

Ultrasonic Ranging Module HC - SR04 Product features:λ Ultrasonic ranging module HC - SR04 provides 2cm

- 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work: (1) Using IO trigger for at least 10us high level signal, (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back. (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning. Test distance = (high level time velocity of sound (340M/S) / 2,

Wire connecting direct as following:

5V Supply
Trigger Pulse Input
Echo Pulse Output
0V Ground

Features of Ultrasonic Sensor:

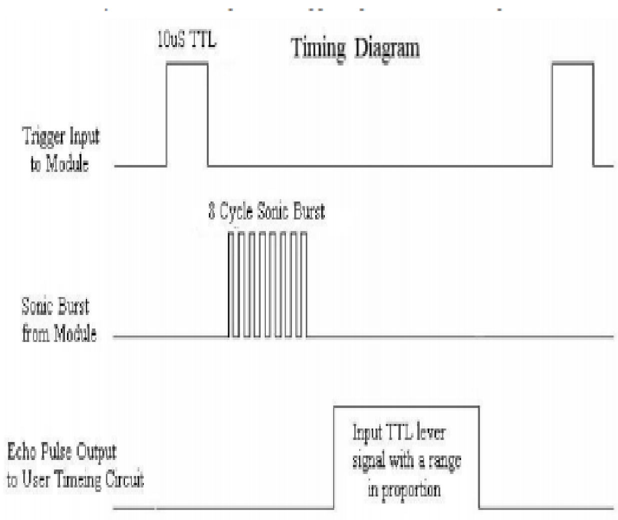
- Compact and light weight
- High sensitivity and high pressure
- High reliability
- Power consumption of 20mA
- Pulse in/out communication
- Narrow acceptance angle
- Provides exact, non-contact separation estimations within 2cm to 3m
- The explosion point LED shows estimations in advancement
- 3-pin header makes it simple to connect utilizing a servo development link.

Electric Parameter:

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



DC MOTOR:

The speed of a DC motor is directly proportional to the supply voltage, so if we reduce the supply voltage from 12 Volts to 6 Volts, the motor will run at half the speed. How can this are achieved when the battery is fixed at 12 Volts. The speed controller works by varying the average voltage sent to the motor. It could do this by simply adjusting the voltage sent to the motor, but this is quite inefficient to do. A better way is to switch the motor's supply on and off very quickly. If the switching is fast enough, the motor doesn't notice it, it only notices the average effect

TRAFFIC SIGNAL

- Turning arrows at traffic lights
- Signs at traffic lights
- B signals
- Bicycle riders
- T signals
- Red light speed cameras
- Pedestrian signals
- **Red** means stop. Wait behind the stop line.
- Do not go through the intersection.

Yellow (amber) means stop. You can enter the intersection if you are so close that sudden braking might cause a crash.

Green means proceed through the intersection carefully.

Turning arrows at traffic lights

Some traffic lights have arrows to control traffic turning right or left.

A green arrow means you can only turn in that direction.

A red arrow means traffic waiting to turn must wait behind the **stop** line until the arrow turns green or disappears, even if there is no traffic on the road.

You must not go straight ahead or turn left. You may turn right.

You must not turn right but you can go straight ahead or turn left if the way is clear.

When the yellow (amber) arrow is flashing, you may proceed but you must give way to any pedestrian who is crossing the road into which you are turning.

Note:

Some intersections that have traffic lights contain bicycle storage areas. These painted areas on the road allow cyclists to stop in safety. When traffic lights or arrows are red, drivers must not allow any part of the vehicle to enter the bicycle storage area.

Pedestrian Crossing:

When making a turn at an intersection you must give way to pedestrians on the road into which you are turning. Where the traffic at the intersection is controlled by traffic lights. The Camera will also detect the zebra crossing pattern and act accordingly.

IV PROJECT IMPLEMENTATION

- Tools and Technologies Used
- HARDWARE REQUIREMENTS :

_ Raspberry pi Processor
_ Optical Camera
_ Battery 12v
_ Ultra-sonic Sensor
_ Power-Bank for Power Supply

- SOFTWARE REQUIREMENTS :

_ Operating system : Raspberry Pi
_ Coding Language : Python
_ IDE : Python (OpenCV)

- Methodologies/Algorithm Details
- Hough Transform Algorithm

The Hough transform is a feature extraction technique used in image anal-
ysis, computer vision, and digital image processing.[1] The purpose of the technique is to _nd imperfect instances of objects within a certain class of
shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.The classical Hough transform was concerned with the identication of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses. The Hough transform as it is universally used today was invented by Richard Duda and Peter Hart in 1972, who called it a "generalized Hough transform"[2] after the related 1962 patent of Paul Hough.[3][4] The trans-form was popularized in the computer vision community by Dana H. Ballard through a 1981 journal article titled "Generalizing the Hough transform to detect arbitrary shapes". In automated analysis of digital images, a sub problem often arises of detecting simple shapes, such as straight lines, circles or ellipses. In many cases an edge detector can be used as a pre-processing stage to obtain image points or image pixels that are on the desired curve in the image space. Due to imperfections in either the image data or the edge detector, however, there may be missing points or pixels on the desired curves as well as spatial deviations between the ideal line/circle/ellipse and the noisy edge points as they are obtained from the edge detector. For these reasons, it is often non-trivial to group the extracted edge features to an appropriate set of lines, circles or ellipses. The purpose of the Hough transform is to address this problem by making it possible to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects (Shapiro and Stockman, 304). The simplest case of Hough transform is detecting straight lines. In general, the straight line y = mx + b can be represented as a point (b, m) in the parameter space. However, vertical lines pose a problem. They would give rise to unbounded values of the slope parameter m. Thus, for computational reasons, Duda and Hart[5] proposed the use of the Hesse normal form

r = x cos + y sin r = x cos + y sin r = x cos + y sin

where r rristhedistancefromtheorigintotheclosestpointonthestraightline; and(theta)istheanglebetweenthexxxaxisandthelineconnectingthe originwiththatclosestpoint:

It is therefore possible to associate with each line of the image a pair

(r ,) (r;)(r;):T he(r;)(r;)(r;)

plane is sometimes referred to as Hough space for the set of straight lines in two dimensions. This representation makes the Hough transform conceptually very close to the two-dimensional Radon transform.

The dimension of the accumulator equals the number of unknown parameters, i.e., two, considering quantized values

|| Volume 3 || Issue 6 || June 2018 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS

of r and in the pair (r,).For each pixel at (x,y) and its neighborhood, the Hough transform algorithm determines if there is enough evidence of a straight line at that pixel. If so, it will calculate the parameters (r,) of that line, and then look for the accumulator's bin that the parameters fall into, and increment the value of that bin. By nding the bins with the highest values, typically by looking for local maxima in the accumulator space, the most likely lines can be extracted, and their (approximate) geometric denitions read o. (Shapiro and Stockman,304) The simplest way of ending these peaks is by applying some form of threshold, but other techniques may yield better results in di
erent circumstances determining which lines are found as well as how many. Since the lines returned do not contain any length information, it is often necessary, in the next step, to nd which parts of the image match up with which lines. Moreover, due to imperfection errors in the edge detection step, there will usually be errors in the accumulator space, which may make it non-trivial to and the appropriate peaks, and thus the appropriate lines. The final result of the linear Hough transform is a two-dimensional array (matrix) similar to the accumulator one dimension of this matrix is the quantized angle and the other dimension is the quantized distance r. Each element of the matrix has a value equal to the sum of the points or pixels that are positioned on the line represented by quantized parameters (r,). So the element with the highest value indicates the straight line that is most represented in the input image.

Color Separation Algorithm

The Color separation operation allows you to extract di
erent 'bands' for
instance from a scanned or digital color photo as if using color lters when
taking the picture. After color extraction, you can perform the normal Image
Processing operations like Filtering, Classification, etc. on these bands.
Maps that have a Picture domain or the (24 bit) Color domain store
for each pixel three values: Red, Green and Blue. The Color separation oper-
ation allows you to retrieve for each pixel either the Red, Green or Blue value
and store these in a separate map. You can also retrieve Yellow, Magenta,
Cyan, combined Gray values, or Hue, Saturation or Intensity values for each
pixel.
For each pixel the selected color is retrieved. Red, Green and Blue
values are directly retrieved from the map or the picture domain. When another color is selected, one of the following formulas is applied:

Yellow = 255 - Blue
Magenta = 255 - Green
Cyan = 255 - Red
Hue = (255/2p) * arctan2 (1/2 3 * (Green-Blue) , Red - (Green+Blue)
/ 2) * 240/255
Saturation = (Red2 + Green2 + Blue2 - Red*Green - Red*Blue -Green*Blue) * 240/255
Intensity = 1/3 * (Red + Green + Blue) * 240/255
Gray = 0.3 * Red + 0.59 * Green + 0.11 * Blue

Red, Green, and Blue values range from 0 to 255. Hue, Saturation, and
Intensity values however range from 0 to 240; this range complies with the
Windows color scheme definition. In the formulas for Hue, Saturation and
Intensity above, multiplication factor 240/255 is used to obtain that range.

8.3 Verication and Validation for Acceptance Verification and validation are independent procedures that are used together
for checking that a product, service, or system meets requirements and speccations and that it fullls its intended purpose. These are critical compo-
nents of a quality management system such as ISO 9000. The words verification and validation are sometimes preceded with independent, indicating that the verication and val- idation is to be performed by a disinterested
third party. Independent verication and validation can be abbreviated as
IV and V.

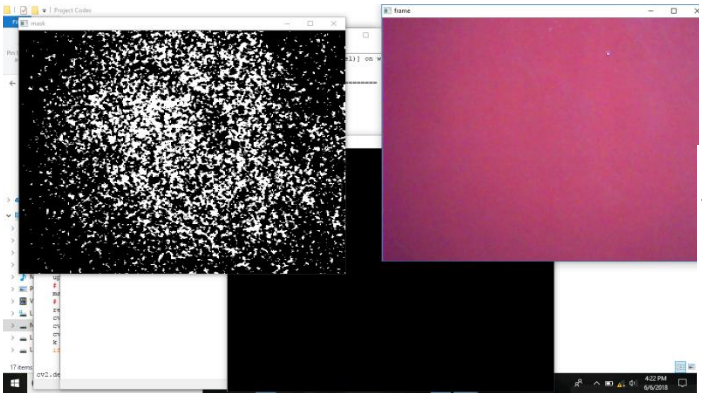
Validation : The assurance that system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with verication.

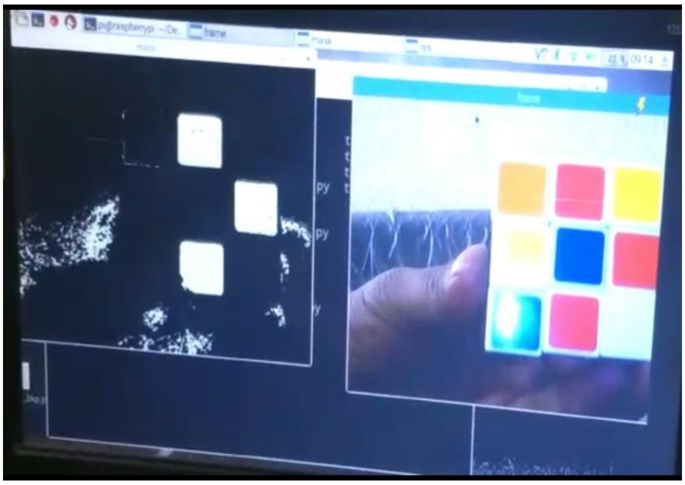
Verication : The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with validation.

V RESULTS

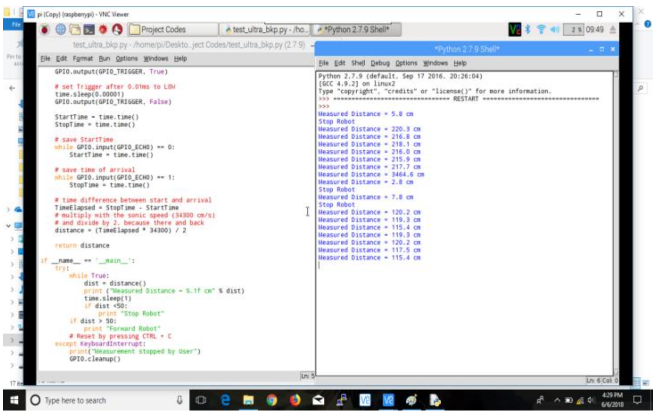
1. COLOR DETECTION :

|| Volume 3 || Issue 6 || June 2018 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

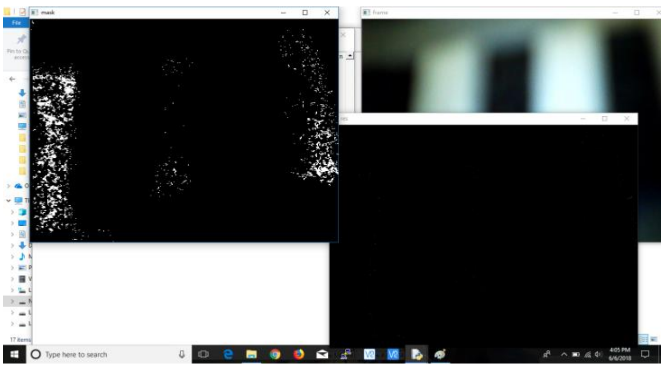




2. OBJECT DETECTION :



3. ZEBRA CROSS DETECTION :



Advantages:

1) Maximize car utilization
2) Customer satisfaction
3) Ensure car safety
4) To help drivers reversing vehicles into tight parking spaces and unfamiliar docks
5) Such systems minimize repair costs, vehicle downtime and injuries by avoiding collisions.
6) This technology is easy to install
7) Acting as an advanced warning about obstacles under low-visibility conditions like fog, snow and rain.

VI CONCLUSION

In this work, an approach for collision avoidance for cognitive automobiles was presented. For the detection of obstacles, a 3d PMD-camera was used which delivers a full 3d image of the surrounding. The detection was done using a clustering algorithm which groups pixels from the 3d-camera based on their spatial relations. The result is a list of possible obstacles. For vehicle control, a behavior-based architecture is used. The main principle is the decoupling of behaviors as separate modules and an ordering of the behaviors at different levels of abstraction. The result of the behavior execution consists of a corridor which describes lateral control parameters. Additionally, hints for speed and acceleration are calculated. Furthermore, a front line as the latest possible stop line is provided.

REFERENCES

[1] S. Thrun et al., "Stanley: The robot that won the DARPA grand challenge," J. Field Robot. (JFR), vol. 23, no. 9, pp. 661–692, Sep. 2006.

[2] U. Ozguner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The DARPA grand challenge experience," Proc. IEEE, vol. 95, no. 2, pp. 397–412, Feb. 2007.

[3] M. Buehler, K. Iagnemma, and S. Singh, The 2005 DARPA Grand Challenge: The Great Robot Race. Berlin, Germany: Springer-Verlag, 2007, vol. 36.

[4] M. Montemerlo et al., "Junior: The Stanford entry in the urban challenge," J. Field Robot. (JFR), vol. 25, no. 9, pp. 569–597, Sep. 2008.

[5] J. Leonard et al., "A perception-driven autonomous urban vehicle," J. Field Robot. (JFR), vol. 25, no. 10, pp. 727–774, Oct. 2008.

[6] B. J. Patz, Y. Papelis, R. Pillat, G. Stein, and D. Harper, "A practical approach to robotic design for the DARPA urban challenge," J. Field Robot. (JFR), vol. 25, no. 8, pp. 528–566, Aug. 2008.

[7] C. Urmson et al., "Autonomous driving in urban environments: Boss and the urban challenge," J. Field Robot. (JFR), vol. 25, no. 8, pp. 425–466, Aug. 2008.

[8] J. Markoff, Google Cars Drive Themselves, in Traffic. New York, NY, USA: The New York Times, 2010, vol. 10, p. A1.

[9] M. Bertozzi, A. Broggi, A. Coati, and R. I. Fedriga, "A 13,000 km intercontinental trip with driverless vehicles: The VIAC experiment," IEEE Intell. Transp. Syst. Mag., vol. 5, no. 1, pp. 28–41, Spring 2013.

[10] A. Broggi, L. Bombini, S. Cattani, P. Cerri, and R. I. Fedriga, "Sensing requirements for a 13,000 km intercontinental autonomous drive," in

[11] W. Burgard and M. Hebert, World Modeling. Berlin, Germany: SpringerVerlag, 2008. [12] P. Pfaff, R. Triebel, and W. Burgard, "An efficient extension to elevation maps for outdoor terrain mapping and loop closing," Int. J. Robot. Res., vol. 26, no. 2, pp. 217–230, Feb. 2007.

[13] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Path planning in 3-D environments using the normal distributions transform," in Proc. IEEE/RSJ Int. Conf. IROS, 2010, pp. 3263–3268.

[14] Y. Choe, I. Shim, and M. J. Chung, "Urban structure classification using the 3-D normal distribution transform for practical robot applications," Adv. Robot., vol. 27, no. 5, pp. 351–371, Apr. 2013.

[15] S. Huh and D. H. Shim, "A vision-based landing system for small unmanned aerial vehicles using an airbag," Control Eng. Pract., vol. 18, no. 7, pp. 812–823, Jul. 2010.

[16] G. C. Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. New York, NY, USA: Springer-Verlag, 2011, vol. 24.

[17] D. P. Bovet and M. Cesati, Understanding the Linux kernel. Sebastopol, CA, USA: O'Reilly Media, Inc., 2005.

[18] B. Chapman, G. Jost, and R. Van Der Pas, Using OpenMP: Portable Shared Memory Parallel Programming. Cambridge, MA, USA: MIT Press, 2008, vol. 10.

[19] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in Proc. IEEE/RSJ Int. Conf. IROS, 2004, vol. 3, pp. 2301–2306.

[20] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, Machine Learning, Neural and Statistical Classification. New York, NY, USA: Ellis Horwood, 1994.

[21] "Pattern Recognition in Autonomous vehicle using IOT."Tushar Bobade, Yash Naidu, Rushikesh Hire, Suraj Lad.Department of Computer Science Engineering SKNSITS Lonavala