Mansoura University
Faculty of Engineering
Dep. Of Electronics & Communications Engineering

Graduation Project On:
VEHICLE COLLISION AVOIDANCE SYSTEM

Prepared by:

- Serag Mounir Manaa
- Marina Samir Anton
- Mohamed Ahmed Abo Zaid
- Mohamed Tarek Shalaby
- Mohamed Mahmoud Abd-Elaaty
- Nada Ahmed Sadek
- Nora Ebrahim Sharaf

Supervised by:

Assis.Prof. / Ahmed Elnakib

2016

Table of Contents

Chapter 4

Chapter 5

Chapter 1

Introduction

Egypt occupies the tenth position globally in road accidents[1]. Therefore there is a need to build collision system. The goal of this project is to provide highway obstacle detection in a low cost and makes an automatic response of brakes of car. First Technique introduced the obstacle detection which uses combination of ultrasound, infrared lasers and radars.

1.1 Problem statement

A collision avoidance system is an automobile safety system designed to reduce the severity of an accident. Also known as precrash system, forward collision warning system or collision mitigating system, it can use radar (influenced by all weather) and sometimes laser and camera (both sensor types are ineffective during bad weather) to detect an imminent crash. Once the detection is done, these systems either provide a warning to the driver when there is an imminent collision or take action autonomously without any driver input (by braking or steering or both). Collision avoidance by braking is appropriate at low vehicle speeds (e.g. below 50 km/h), while collision avoidance by steering is appropriate at higher vehicle speeds[2]. Cars with collision avoidance may also be equipped with adaptive cruise control, and use the same forward-looking sensors.

1.2 Objective of Work

This project aims to create a low cost, retrospective solution that can be implemented in large scale to help reduce a significant number of accidents. This is by no means a fully autonomous system as it is, but an effective driver assistance system which helps the driver use an automobile in a safe way without getting into a crash situation from which the driver may find it hard to get out of. The system can also help a panicking driver to safely get out of the crash scenario. This system contains two levels of assistance, the first level being a driver alerting system followed by a controlled braking process.

1.3 Review on collision systems

The world's first demonstration of forward collision avoidance was performed in 1995 by a team of scientists and engineers at Hughes Research Laboratories in Malibu, California. The project was funded by Delco Electronics, and was led by HRL physicist Ross D. Olney. The technology was labeled for marketing purposes as FOREWARN. The system was radar based - a technology that was readily available at Hughes Electronics, but virtually no place else in the world. A small custom fabricated radar-head was developed specifically for this automotive application at 77 GHz. The forward radar-head, plus the signal processing unit and visual-audio-tactile feedbacks were first integrated into a Lexus SC400, and shortly thereafter into a Cadillac STS[3].

An SUV-style (The "***SUV***" term is defined as "a large vehicle that is designed to be used on rough surfaces but that is often used on city roads or highways.) concept vehicle known as SSC (Safety Security & Communications). At the North American International Auto Show at Cobo Hall in Detroit in 1996. This was a fully functional vehicle, and demonstrations were concurrently being provided by a duplicate vehicle[5],[6]. While primarily a warning system, with various feedbacks, the system did have minor control of the brakes which were pulsed to begin a braking action in the event of a potential collision, making it also the beginning of avoidance systems. These SSC vehicles were sent around the world, including Europe and Asia, to share this very important life-saving technology with all the major automotive manufacturers in an effort to quick-start their individual development efforts. It took almost 20 years for this important technology to reach the consumer marketplace.

In the early-2000, the U.S. National Highway Traffic Safety Administration (NHTSA) researched whether to make frontal collision warning systems and lane departure warning systems mandatory[7].

In 2011, a question was submitted to the European Commission regarding stimulation of these "collision mitigation by braking" systems[8]. The mandatory fitting of Advanced Emergency Braking Systems in commercial vehicles will be implemented on 1 November 2013 for new vehicle types and on 1 November 2015 for all new vehicles in the European Union[9]. This could, according to the impact assessment[10], ultimately prevent around 5,000 fatalities and 50,000 serious injuries per year across the EU.

A 2012 study[11] by the Insurance Institute for Highway Safety examined how particular features of crash-avoidance systems affected the number of claims under various forms of insurance coverage. The findings indicate that two crash-avoidance features provide the biggest benefits: (a) *autonomous braking* that would brake on its own, if the driver does not, to avoid a forward collision, and (b) *adaptive headlights* that would shift the headlights in the direction the driver steers. They found lane departure systems to be not helpful, and perhaps harmful, at the circa 2012 stage of development.

A 2015 Insurance Institute for Highway Safety study found forward collision warning and automatic braking systems reduced rear collisions[12]. collision avoidance features are rapidly making their way into the new vehicle fleet.

1.4 Review on Automobile :

Table “1” Safety systems development in vehicles :

	Year	*Project*	*Uses*
Audi		Braking guard	Radar
	2010	Pre sense	Twin 7.radar and monocular camera sensors[13]
	2011	Pre Sense Plus[14]	N.A * (denote as non-available)
	2015	Avoidance Assistant[15]	N.A
BMW	2012	Active Protection	N.A
	2013	Driving Assistant Plus	Camera ,radar ,sensors
Cadillac	2012	Forward Collision Alert	Camera ,radar[16]
Chrysler	2010:2011	Forward Collision Warning[17]	N.A

Chrysler	2014	Collision Warning	N.A
Fiat	2013	City Brake Control[18],[19]	Laser Lidar sensor
Ford	2009	the Collision Warning with Brake Support[20]	a mix of sensors, including a camera behind the rearview mirror[21]
General Motors	2012	GM's collision alert system[22]	a camera to provide warning when there is a vehicle ahead or there is a lane departure
	2013	Forward collision alert[23],[24]	Camera, radar sensors on both sides of the vehicle
Honda	2003	Collision Mitigation Brake System CMBS, originally CMS[25]	Radar[26],[27]
	2004	Night vision system[28]	N.A
Hyundai		Hyundai Genesis DH[29]	radar and windshield camera
		Smart City Brake Support (SCBS)	lasers to detect vehicles or obstacles in front of the car

Mazda		Smart Brake Support (SBS)[30],[31].	millimeter wave radar to detect vehicles up to 200 metres
Mercedes-Benz	2002	Pre-Safe[32]	Electronic Stability Control sensors to measure steering angle, vehicle yaw, and lateral acceleration and brake assist(BAS) sensors to detect emergency braking
	2005	Brake Assist BAS plus[33]	radar based forward collision warning
	2013	radar based forward collision warning[34]	combination of stereo camera and radar sensors to detect pedestrians in front of the vehicle
Mitsubishi		Mitsubishi's Forward Collision Mitigation (FCM), Adaptive Cruise Control (ACC), and Lane Departure Warning System (LDW)	N.A
Nissan		Intelligent Brake Assist (IBA) with Forward Emergency Braking (FEB) (on QX80)	Laser based and Radar based systems

Nissan		Predictive Forward Collision Warning system	Laser based and Radar based systems
		Forward Emergency Braking	
Peugeot	2014	Introduced active emergency braking system on Peugeot 308	N.A
Renault	2014	Introduced radar active emergency braking on 2015 Renault Espace[35]	N.A
Skoda	2013	Introduced four new security system[36]	N.A
Subaru	2008	Introduced the EyeSight system to Japanese drivers on the Legacy sedan, wagon and Outback[37]	Two CCD stereo cameras mounted to the roof beside the rear view mirror
	2014	updated the system[38]	cameras that can detect color (brake lights) and reducing the bulk of the system.
Tesla		has a feature known as Collision Avoidance Assist that automatically engages the brakes to reduce the impact of a	N.A

		frontal collision[39].	
Toyota	2008	Toyota's "Pre-Collision System (PCS) was the first production forward collision warning system.	Radar (blue) and Stereo camera (red) coverage.
	2003 February	Toyota launched PCS in on the redesigned Japanese domestic market Harrier	N.A
	August 2003	added an automatic partial pre-crash braking system to the Celsior[40].	N.A
	September 2003	PCS made its first appearance in North America on the lexus LS 430 , becoming the first radar-guided forward collision warning system offered in the US[41].	N.A
	July 2004	The Crown Majesta Radar PCS added a single digital camera to improve the accuracy of collision forecast and warning and control levels[42][43][44]	Digital camera

Toyota	2006	Pre-Collision System with the first Driver Monitoring System	CCD Camera on the steering column
		The Lexus LS featured a further advanced version of the PCS. Advanced Pre-Collision System (APCS)[45]	twin-lens stereo camera and a more sensitive radar to detect for the first time smaller "soft" objects such as animals and pedestrians.
	2008	Updated Driver Monitoring System added on the Crown for detecting whether the driver's eyes are properly open. This system is designed to work even if the driver is wearing sunglasses, at night[46].	N.A
	2009	adding a front-side millimeter-wave radar to detect potential side collisions primarily at intersections or when another vehicle crosses	N.A

Toyota		the center line[47],[48]	
	2012	Higher Speed A-PCS on facelifted Lexus LS: enables deceleration of up to 37 MPH as compared to the current generation system of 25 MPH[49].	The same technologies as the current A-PCS.
	2013	Pre-Collision System with Pedestrian-avoidance Steer Assist and Steering Bypass Assist can help prevent collisions in cases where automatic braking alone is not sufficient[50][51].	The system issues an audio and visual alarm to encourage the driver to take evasive action
volkswagen	2010	"Front Assist" on the 2011 Vollkswagen Touareg can brake to a stop in case of an emergency and tension the seat belts as a precautionary measure[52].	N.A
	2012	Vollkswagen Golf MK7 introduced their "Proactive Occupant Protection", which will close the windows and retract the safety belts to	N.A

volkswagen		remove excess slack if the potential for a forward crash is detected.	
	2014	Vollkswagen Passat B8 introduces pedestrian recognition a part of the system.	Sensor fusion between a camera and the radar sensor.
Volvo	2006	Volvo's "Collision Warning with Auto Brake", developed in cooperation with Mobileye , was introduced on the 2007 S80 .	This system is powered by a radar/camera sensor fusion and provides a warning through a head up display that visually resembles brake lamps.
	2013	Volvo introduced the first cyclist detection system[53].	Lidar laser sensor that monitors the front of the roadway
	2015	"IntelliSafe" with Auto brake at intersection. The Volvo XC90 is the first technology .	N.A

1.5 Proposed System

Road safety has become a major social issue with the number of cars and trucks increasing day by day. This project focuses on development of a crash warning and avoidance system that monitors the environment of the vehicle constantly and assisting the driver in avoiding a collision. This idea targets a low cost, retrofit scheme which can be used in roads to prevent the worst case scenarios in road crashes. This system incorporates an ARM based processing unit running RTOS which uses a camera to detect a crash scenario and send control signals to corresponding Electronic Control Units.

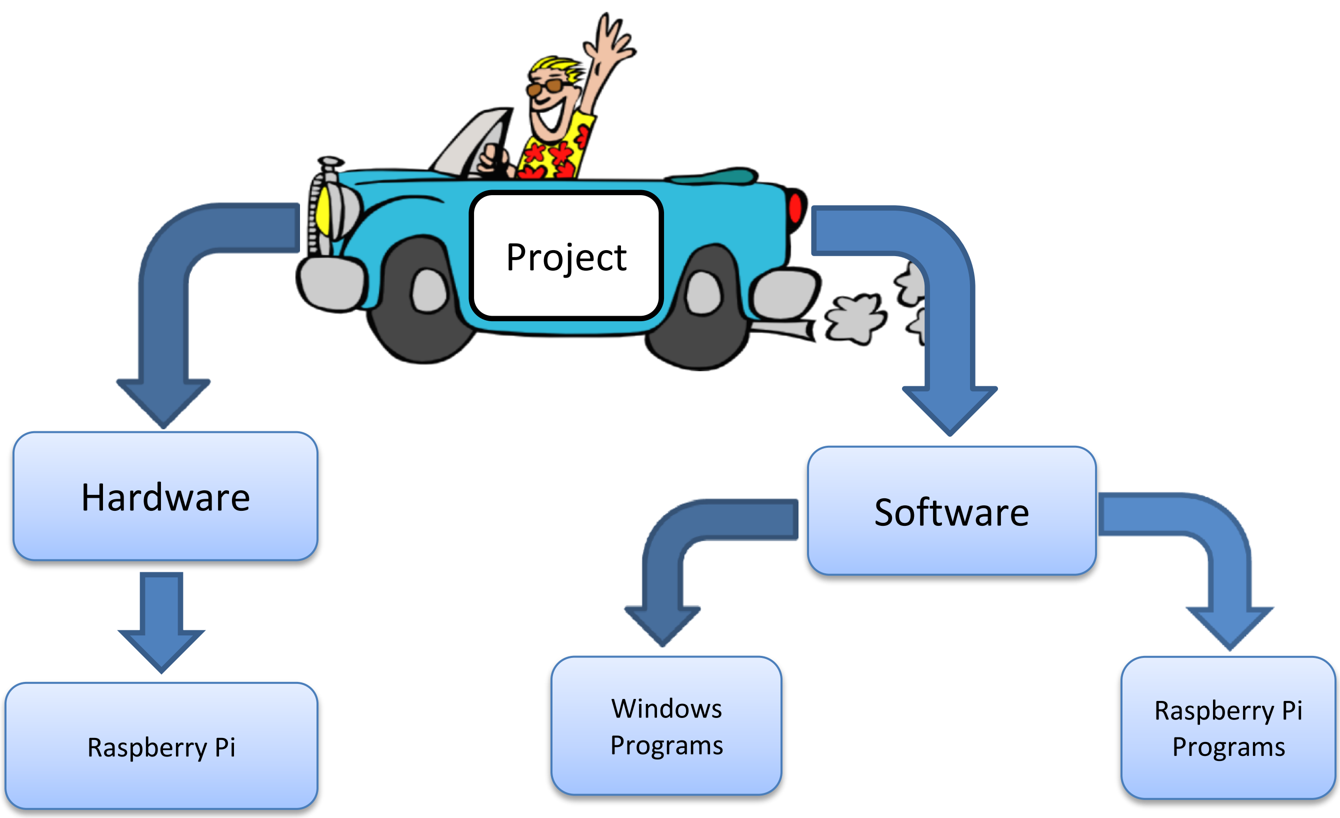


Figure (1) Basic diagram of our project

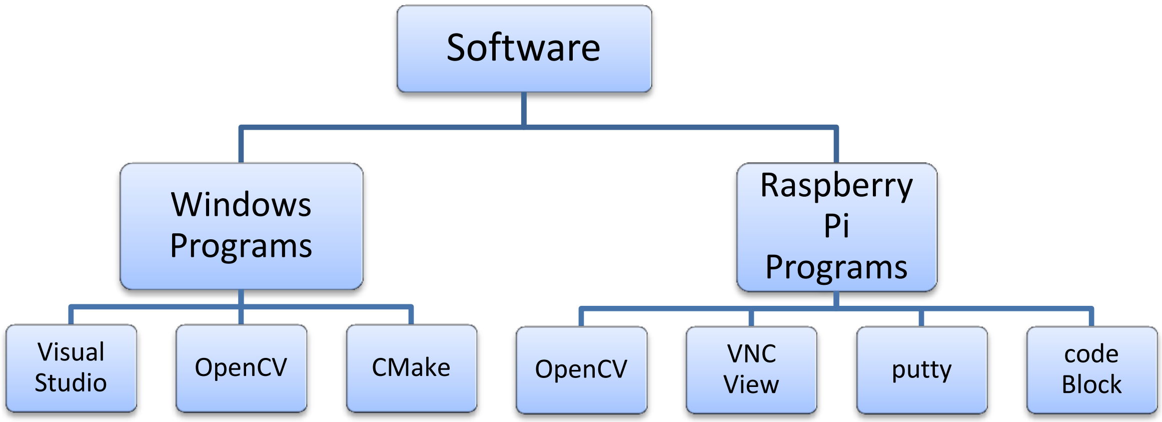


Figure (2) Block diagram of software

1.6 Organization

This project is organized as follows :
Chapter (2) : Hardware (Raspberry pi 3) .
Chapter (3) : Software .
Chapter (4) : Integration Between Software and Hardware .

1.7 References

1. https://www.google.com.eg/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=0ahUKEwjbldeSpdHNAhUDuRQKHf7ZDj4QFgg-MAQ&url=http%3A%2F%2Fwww.mei.edu%2Fcontent%2Farticle%2Ftraffic-accidents-egypt-need-reform&usg=AFQjCNFCqwZosdbK6O96UxYihzlfx7_Dzg
2. Kanarachos, Stratis (2009). "A new method for computing optimal obstacle avoidance steering manoeuvres of vehicles". International Journal of Vehicle Autonomous Systems**7** (1): 73–95. doi:10.1504/IJVAS.2009.027968. Retrieved 29 July 2015.
3. "U.S. DOT and IIHS announce historic commitment of 20 automakers to make automatic emergency braking standard on new vehicles". U.S. Department of Transportation National Highway Traffic Safety Administration. 17 March 2016. Retrieved17 March 2016.
4. Olney R.D.; et al. (November 1995), "Collision Warni
5. ng System Technology", Intelligent Transport Systems World Congress (Yokohama, Japan)
6. "The Mensa-Mobile". Car & Driver: 144–145. August 1996.
7. Wiesenfelder, Joe (February 1996). "The Information Superhighway (This is not a metaphor)". Wired (4.02).
8. Keenan, Tim (May 1, 1996), "Delco's dream machine", Wards Auto
9. "Forward Collision Warning Requirements Project Final Report - Task 1" (PDF). National Highway Traffic Safety Administration. January 2003. Retrieved 29 July 2015.
10. "Written question - Rear-end traffic collisions in the European Union - E-011477/2011".europa.eu. Retrieved 25 January 2015.
11. "Answer to a written question - Rear-end traffic collisions in the European Union - E-011477/2011". europa.eu. Retrieved 25 January 2015.
12. "Annex to the proposal for a regulation of the European Parliament and of the Council concerning type-approval requirements for the general safety of motor vehicles - Impact Assessment" (PDF). Commission of the European Communities. 23 May 2008. Archived from the original (PDF) on 23 June 2015. Retrieved 31 March 2016.
13. "Crash avoidance features cut insurance claims". iihs.org. Retrieved 4 April 2015.

14. Beene, Ryan (2016-01-28). "Automatic braking reduces rear-end crashes, IIHS study finds". Automotive News. Retrieved 2016-03-10.
15. "Extensive safety in the new Audi A8" (Press release). Bosch Media Services. 27 April 2010. Retrieved 29 July 2015.
16. "The new Audi A8" (PDF) (Press release). Retrieved 17 February 2010.
17. "The new Audi Q7 – Sportiness, efficiency, premium comfort" (Press release). Audi Media Center. 12 December 2014. Retrieved 20 July 2015.
18. "Cadillac 'Virtual Bumpers' Can Help Avoid Crashes". media.gm.com (Press release). Retrieved 25 January 2015.
19. Chrysler Group LLC (2 June 2010). "All-new 2011 Jeep Grand Cherokee Earns 'Top Safety Pick' From the Insurance...". prnewswire.com (Press release). Retrieved25 January 2015.
20. "City Brake Control wins the 'Euro NCAP Advanced' award". FGA EMEA Press site(Press release). June 26, 2013. Retrieved June 30, 2014.
21. "Euro NCAP Advanced: FIAT City Brake Control". Euro NCAP official site. RetrievedJune 30, 2014.
22. "Ford's latest safety brakethrough – collision warning with brake support – coming in 2009". Media Ford (Press release). 6 April 2009. Archived from the original on 20 March 2012. Retrieved 31 March 2016.
23. "New Toyota, Ford Systems Can Steer Clear of Pedestrians". thedetroitbureau.com. Retrieved 25 January 2015.
24. "New Camera-Based Collision Alert Debuts on GMC Terrain". media.gm.com (Press release). Retrieved 25 January 2015.
25. "Chevrolet News - United States - Impala" (Press release). Media.gm.com. 2014-12-15. Retrieved 2016-03-10.
26. "2016 Malibu" (Press release). Media.gm.com. 2014-12-15. Retrieved 2016-03-10.
27. "Honda Announces a Full Model Change for the Inspire" (Press release). Honda. 2003-06-18. Retrieved 2015-01-19.
28. "Honda Worldwide". honda.com (Press release). Retrieved 25 January 2015.
29. "Honda Worldwide - World News - News Releases". honda.com (Press release). 20 May 2003. Retrieved 25 January 2015.
30. Archived August 8, 2008, at the Wayback Machine.

31. "Features | Safety Technology | The All-New 2015" (Press release). Hyundai Genesis. Archived from the original on March 11, 2016. Retrieved 2016-03-10.
32. "MAZDA:Mazda to Feature Smart City Brake Support Advanced Safety Technology in All-New Mazda CX-5 Crossover SUV". mazda.com (Press release). Retrieved 25 January2015.
33. "SCBS & SBS | Pre-Crash Safety Technology" (Press release). Mazda. Retrieved2016-03-10
34. (Press release) https://web.archive.org/20071008151232/http://www.mercedes-benz.ca/index.cfm?NewsID=121&id=2959. Archived from the original on October 8, 2007. Retrieved March 14, 2013. Missing or empty |title= (help)
35. Daimler Media: Innovation as a tradition Archived December 29, 2014, at theWayback Machine.
36. "Mercedes-Benz TecDay Special Feature: PRE-SAFE And PRE-SAFE Brake".emercedesbenz.com (Press release). Retrieved 25 January 2015.
37. "Extended PRE-SAFE protection: Prevention is better than cure". daimler.com(Press release). May 2013. Retrieved 25 January 2015.
38. "ADAS: a range of technologies promoting safety and easier driving experience".renault.com (Press release). Retrieved 25 January 2015.
39. "Euro NCAP 2013 Award - Skoda Multi-Collision Brake". Retrieved 2013-04-17.
40. Neff, John. "2013 Subaru models to see with optional EyeSight safety system". Autoblog. Retrieved 25 January 2015.
41. http://media.subaru.com/newsrelease.do;jsessionid=BC4CE9E4A87FD0ABBF77D742AC4BBE54?&id=562&mid=1
42. EVino (27 March 2015). "Version 6.2 Release Notes". Tesla Forums. Retrieved31 March 2016.
43. "Safety matters: advanced crash avoidance technology finds its way into production vehicles in Japan". Automotive Industries. 2004.
44. "Toyota - Technical Development - Electronics Parts". toyota-global.com (Press release). Retrieved 4 April 2015.
45. "Toyota Crown Majesta undergoes complete redesign". theautochannel.com. Retrieved4 April 2015.
46. "(Really Playing it Safe)". Designnews.com. Retrieved 2016-03-10.

47. "Toyota: News Releases". toyota.co.jp (Press release). Retrieved 25 January 2015.
48. Matsubayashi, Kiyoka; Yamada, Yukinori; Iyoda, Motomi; Koike, Shin; Kawasaki, Tomoya; Tokuda, Masanori. "Development of Rear Pre-Crash Safety System For Rear-End Collisions" (PDF). Toyota Motor. Retrieved 2016-03-10.
49. "Toyota Enhances Pre-crash Safety System with Eye Monitor" (Press release). Toyota.co.jp. 2008-01-22. Retrieved 2016- 03-10.
50. "Toyota Launches Redesigned Crown Majesta in Japan". Worldcarfans. Retrieved25 January 2015.
51. "Toyota Adds to Pre-crash Safety Technologies" (Press release). Toyota.co.jp. 2009-02-26. Retrieved 2016-03-10.
52. "Toyota Global Site - Technology File". toyota-global.com (Press release). Retrieved25 January 2015.
53. "Toyota Develops New Pedestrian Safety Technology" (Press release). Toyota.com. 2013-10-11. Retrieved 2016-03-10.

"Lexus and Toyota Will Make Automated Braking Standard on Nearly Every Model and Trim Level by End of 2017". Toyota Press Room (Press release). 21 March 2016. Retrieved 31 March 2016."Volvo Trucks - Emergency braking at its best!". YouTube. Retrieved 25 January 2015

2.1 Raspberry Pi

The Raspberry Pi 3 Model B is out now. This latest model includes 802.11n WiFi, Bluetooth 4.0, and a quad-core 64-bit ARM Cortex A53 running at 1.2 GHz. It's a usable desktop computer. Available now at the usual Pi retailers for $35.

The Raspberry Pi 3 Model B features a quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz. This puts the Pi 3 roughly 50% faster than the Pi 2. Compared to the Pi 2, the RAM remains the same – 1GB of LPDDR2-900 SDRAM, and the graphics capabilities, provided by the VideoCore IV GPU, are the same as they ever were. As the leaked FCC docs will tell you, the Pi 3 now includes on-board 802.11n WiFi and Bluetooth 4.0. WiFi, wireless keyboards, and wireless mice now work out of the box[1].

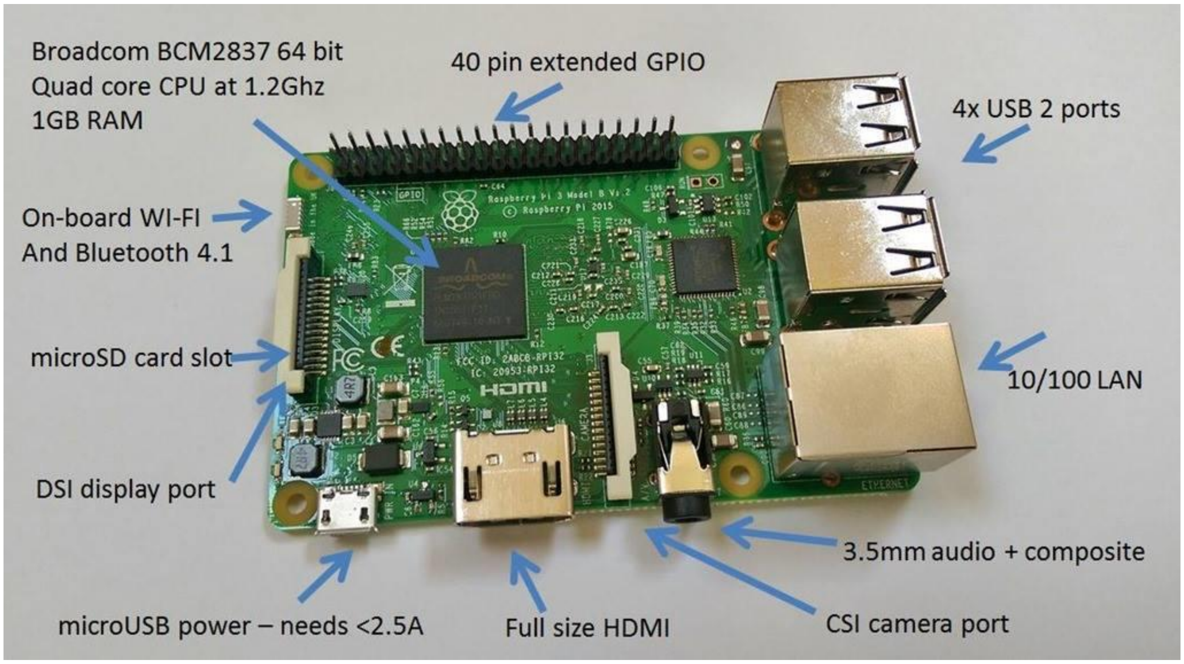
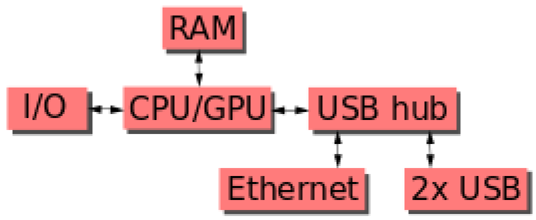


Figure (2.1) Structure of Raspberry pi 3

2.1.1 Hardware

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



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

The Raspberry pi contains five objects:

2.1.1.1 Processor

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

The Raspberry Pi 2 uses a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache[4].

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

2.1.1.2 RAM

On the older beta model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release model B (and model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p frame buffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC). Comparatively the Nokia 701 uses 128 MB for the Broadcom Video Core IV. For the new model B with 512 MB RAM initially there were new standard memory split files released for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). But a week or so later the RPF released a new version of start .lf that could read a new entry in config.txt (gpu_mem=xx) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start .lf worked the same for 256 and 512 MB Raspberry Pis.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero has 512 MB of RAM[5]

2.1.1.3 Networking

Though the model A and A+ and Zero do not have an 8P8C ("RJ45") Ethernet port, they can be connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the model B and B+ the

Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip [6]**.** The Raspberry Pi 3 is equipped with 2.4 GHz 0WiFi 802.11n (600 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) in addition to the 10/100 Ethernet port.

2.1.1.4 Peripherals

The Raspberry Pi may be operated with any generic USB computer keyboard and mouse [7]**.**

2.1.1.5 Video

The video controller is capable of standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions and older standard CRT TV resolutions.

Higher resolutions, such as, up to 2048×1152 [8][9],may work or even 3840×2160 at 15 Hz (too low a framerate for convincing video). Note also that allowing the highest resolutions does not imply that the GPU can decode video formats at those; in fact, the Pis are known to not work reliably for H.265 (at those high resolution, at least), commonly used for very high resolutions (most formats, commonly used, up to full HD, do work).

Although the Raspberry Pi 3 does not have H.265 decoding hardware, the CPU, more powerful than its predecessors, is potentially able to decode H.265-encoded videos in software.

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35

Table (2.1) A comparison between types of Raspberry pi

2.1.2 Operating systems supported by Raspberry

The Raspberry Pi primarily uses Linux-kernel-based operating systems.

The ARM11 chip at the heart of the Pi (first generation models) is based on version 6 of the ARM. The primary supported operating system is Raspbian, although it is compatible with many others. The current release of Ubuntu supports the Raspberry Pi 2, while Ubuntu, and several popular versions of Linux, do not support the older Raspberry Pi 1 that runs on the ARM11. Raspberry Pi 2 can also run the Windows 10 IoT Core operating system, while no version of the Pi can run traditional Windows. The Raspberry Pi 2 currently also supports Open ELEC and RISC OS [10].

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

Raspbian (recommended for Raspberry Pi 1) [10] is maintained independently of the Foundation ; based on the Debian ARM hard-float (armhf) architecture port originally designed for ARMv7 and later processors (with Jazelle RCT/ThumbEE and VFPv3), compiled for the more limited ARMv6 instruction set of the Raspberry Pi 1. A minimum size of 4 GB SD card is required for the Raspbian images provided by the Raspberry Pi Foundation.

2.1.3 Driver Applications

Scheme of the implemented APIs: OpenMAX , OpenGL ES and OpenVG

Raspberry Pi can use a VideoCore IV GPU via a binary blob, which is loaded into the GPU at boot time from the SD-card, and additional software, that initially was closed source. This part of the driver code was later released. However, much of the actual driver work is done using the closed source GPU code. Application software use calls to closed source run-time libraries (OpenMax, OpenGL ES or OpenVG) which in turn calls an open source driver inside the Linux kernel, which then calls the closed source Video Core IV GPU driver code. The API of the kernel driver is specific for these closed libraries. Video applications use OpenMAX, 3D applications use OpenGL ES and 2D applications use OpenVG which both in turn use EGL. OpenMAX and EGL use the open source kernel driver in turn [11].

2.2 USB webcam

The webcam used to detect the objects and capture images is a Logitech USB webcam. The webcam has a still sensor image resolution of 5 megapixel, image capture resolution 640 x 480, the frame rate is 30 fps with video capture resolution of 1024 x 768. The camera is easy to plug in raspberry pi board and supports all Linux based OS [12].

2.3 References

1- http://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/

2-"BCM2835 Media Processor; Broadcom". Broadcom.com. 1 September 2011. Archived from the original on 13 May 2012. Retrieved 6 May 2012

3-Brose, Moses (30 January 2012). "Broadcom BCM2835 SoC has the most powerful mobile GPU in the world?". Grand MAX. Retrieved 13 April 2012

4- Upton, Eben (2016-02-29). "Raspberry Pi 3 on sale now at $35 - Raspberry Pi". Raspberry Pi. Retrieved 2016-02-29.

5- Raspberry Pi boot configuration text file

6- "Microchip/SMSC LAN9514 data sheet;" (PDF). Microchip. Retrieved 15 July 2014

7- "Verified USB Peripherals and SDHC Cards;". Elinux.org. Retrieved 6 May 2012

8- "Raspberry Pi, supported video resolutions". eLinux.org 30 November 2012. Retrieved 11 December 2012

9- "Raspberry Pi, supported video resolutions". eLinux.org. 30 November 2012. Retrieved 11 December 2012.

10- Merz, Alexander (4 February 2015). "Raspberry Pi 2: Schnell rechnen, langsam speichern" (in German). Retrieved 8 February 2015

11- "Raspbian – Debian optimized for the Raspberry Pi hardware".

12- APRIL 2015 ISSN 1819-6608 ARPN Journal of Engineering and Applied Sciences ©2006-2015 Asian Research Publishing Network (ARPN).

We will discuss software we use in both operating systems (windows 10 and Raspbian).

3.1 For windows 10

3.1.1 Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API ,Windows Forms ,Windows Presentation Foundation , Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor Visual Studio supports different programming languages and allows the code editor and debugger to support nearly any programming language, provided a language-specific service exists. Built-in languages include C,C++ and C++/CLI (via Visual C++) , Support for other languages such as Python , Ruby, Node.js .
In our project we used Microsoft Visual Studio 2013we used visual C++ platform to make our code using language of C ++[1].

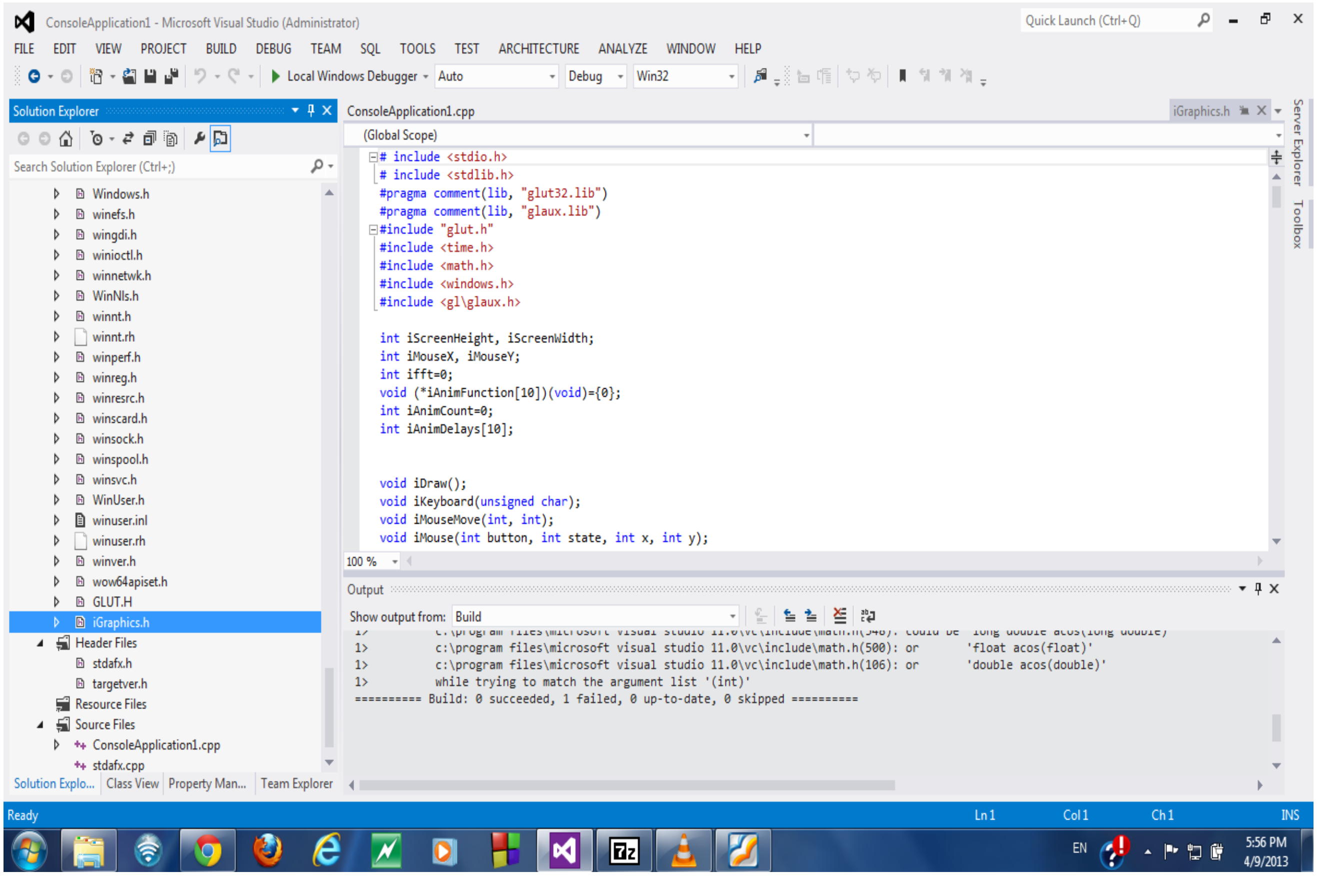


Figure (3.1) Interface of visual studio

3.1.2 OpenCv

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, originally developed by Intel's research center in Nizhny Novgorod (Russia), later supported by Willow Garage and now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license[2].

3.1.2.1 Applications

in our project we used OpenCV For Some Applications such as Motion understanding and Object identification .

3.1.2.2 OS support

OpenCV runs on a variety of platforms. Desktop: Windows, Linux, OS X, FreeBSD, NetBSD, OpenBSD; Mobile: Android, iOS, Maemo, BlackBerry 10. The user can get official releases from Source Forge or take the latest sources from GitHub. OpenCV uses CMake.
We used OpenCv especially Version 3.1 and our operating system is Linux .

3.1.2.3 Programming language

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 and MATLAB/OCTAVE.

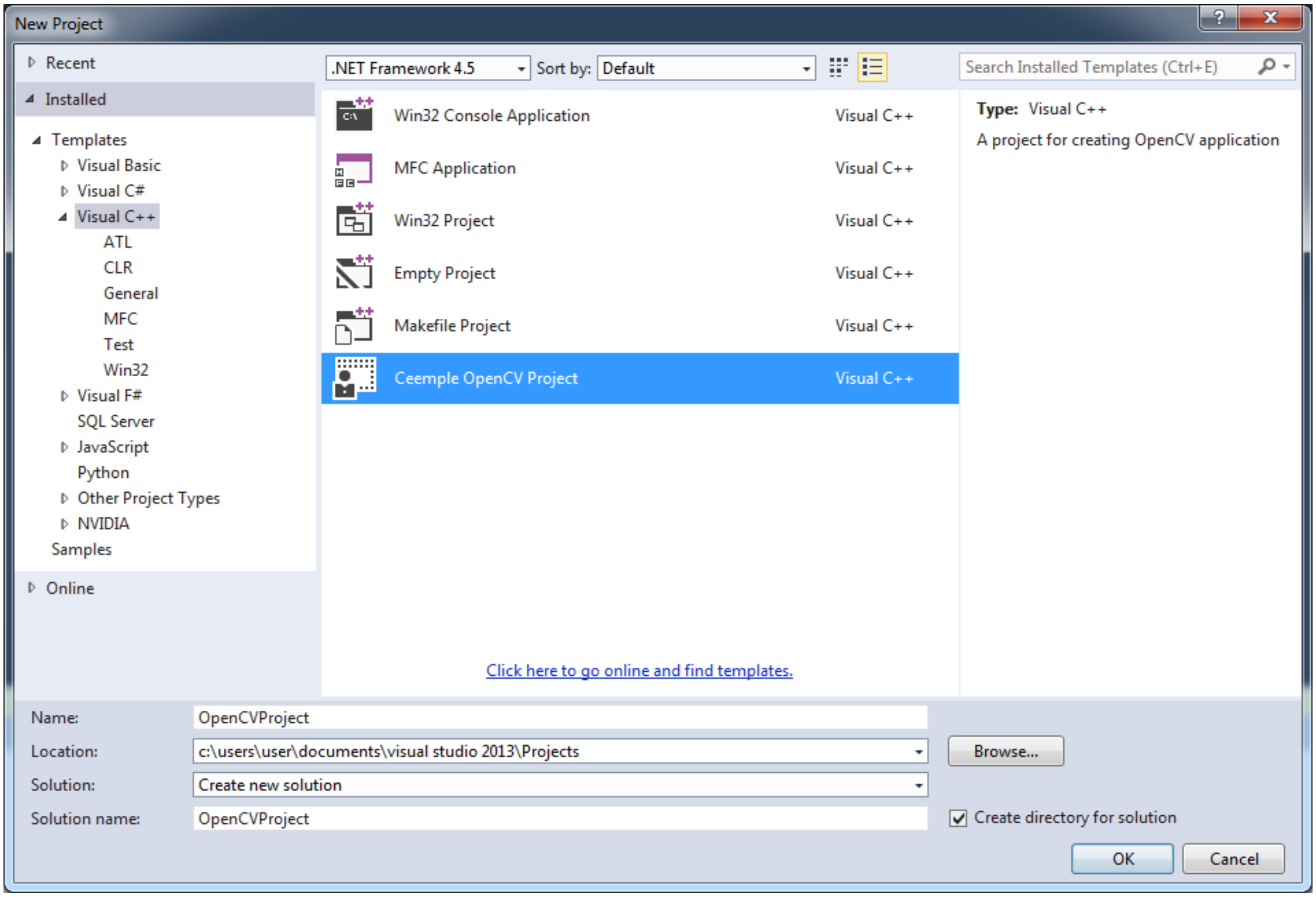


Figure (3.2) creating new project

3.1.3 CMake

CMake is cross-platform free and open-source software for managing the build process of software using a compiler-independent method. It is designed to support directory hierarchies and applications that depend on multiple libraries. It is used in conjunction with native build environments such as make, Apple's Xcode, and Microsoft Visual Studio. It has minimal dependencies, requiring only a C++ compiler on its own build system[3].

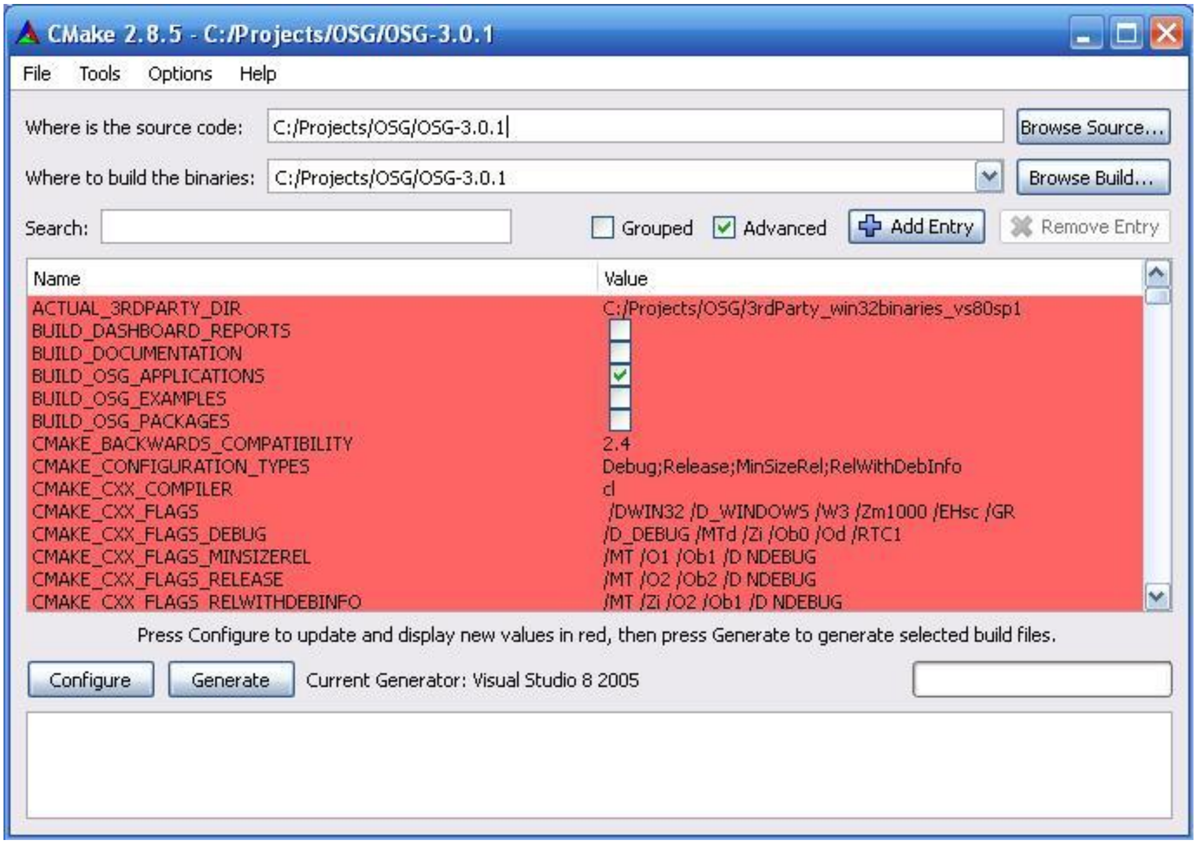


Figure (3.3) Generate and configure tools

CMake scripts can produce Microsoft Visual Studio project and solution files. However, CMake's syntax is more oriented towards Unix/Linux make files which is rather unfriendly for visual studio development which relies primarily on project- properties GUI .

3.2 For Raspbian

3.2.1 Code Blocks

Code Blocks is a free , open-source cross platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C , C++ , and Fortran. It has a custom build system and optionalMake support.

Code::Blocks is being developed for Windows, Linux, and Mac OS X and has been ported to FreeBSD, OpenBSD and Solaris[4].

3.2.2 VNC Viewer (virtual Network Computing)

Sometimes it is not convenient to work directly on the Raspberry Pi. Maybe you would like to work on it from another computer by remote control.

VNC is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer from another. It transmits the keyboard and mouse events from the controller, and receives updates to the screen over the network from the remote host[5].

You will see the desktop of the Raspberry Pi inside a window on your computer. You'll be able to control it as though you were working on the Raspberry Pi itself.

On your Pi (using a monitor or via SSH), install the TightVNC package:

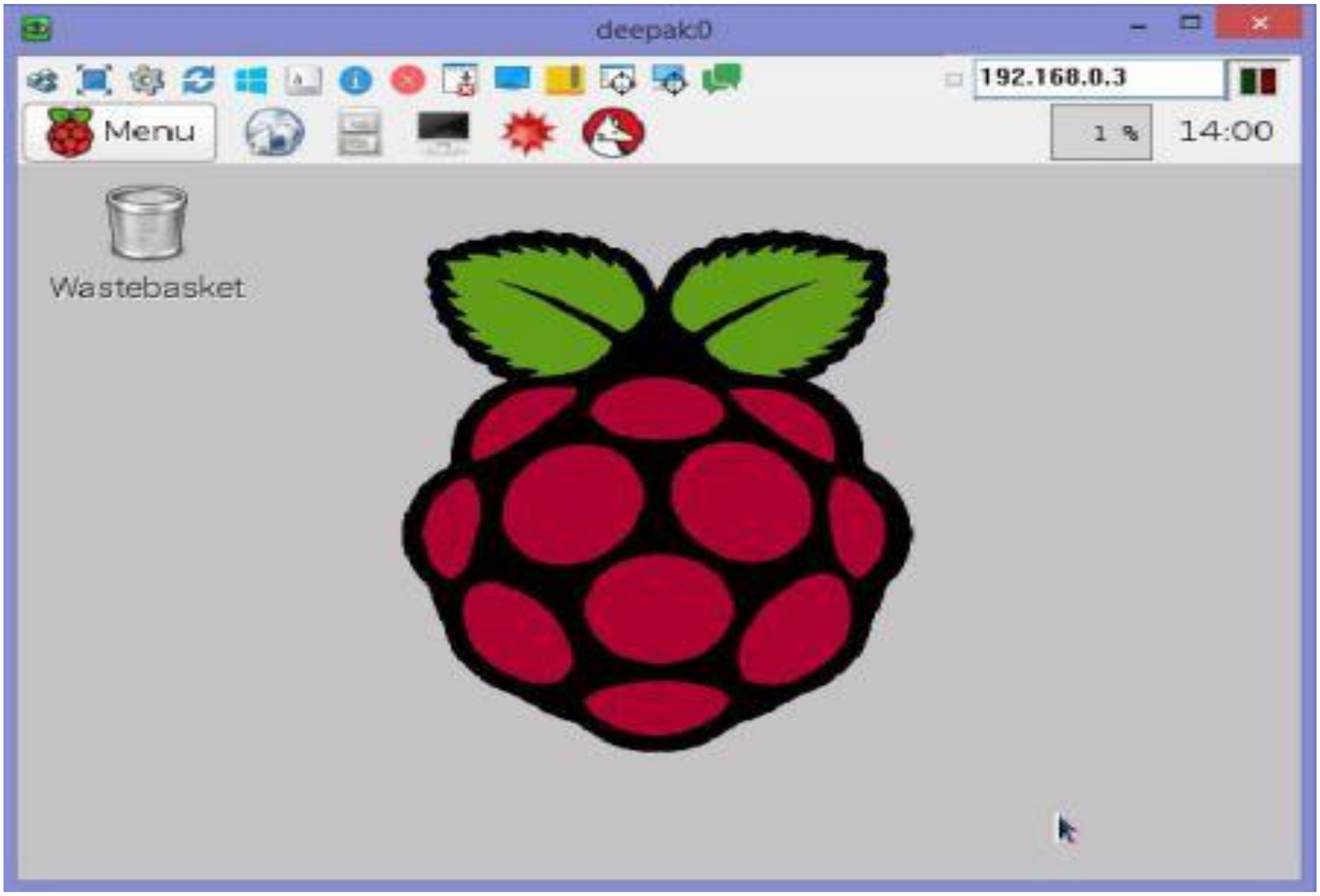
```
sudo apt-get install tightvncserver
```

```
pi@deepak:~$ sudo apt-get install x11vnc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  x11vnc
0 upgraded, 1 newly installed, 0 to remove and 112 not upgraded.
Need to get 0 B/1,044 kB of archives.
After this operation, 2,011 kB of additional disk space will be used.
Selecting previously unselected package x11vnc.
(Reading database ... 78596 files and directories currently installed.)
Unpacking x11vnc (from .../x11vnc_0.9.13-1_armhf.deb) ...
Processing triggers for desktop-file-utils ...
Processing triggers for man-db ...
Setting up x11vnc (0.9.13-1) ...
pi@deepak:~$
```

Next, run Tight VNC Server which will prompt you to enter a password and an optional view-only password

Tightvncserver

```
pi@deepak:~$ x11vnc -storepasswd
Enter VNC password:
Verify password:
Write password to /home/pi/.vnc/passwd?  [y]/n y
Password written to: /home/pi/.vnc/passwd
pi@deepak:~$
```



(Figure 3.8)

You'll now use a VNC **client** program on your PC/laptop to connect to the VNC server and take control of it. Follow instructions for your computer's operating system:

- Linux
- Mac OS
- Windows

Install and Run Putty on your Raspberry Pi .

3.2.3 Putty

Putty is a very useful application that can be used to connect to serial ports and Secure Shell(SSH) to Raspberry Pi's[6].

Putty is mostly used on Windows to connect to remote devices but it can also run on a Raspberry Pi.

Now that Putty has been installed you can use it to connect to all kinds of devices. I use it to connect to GRBL from my Raspberry Pi.

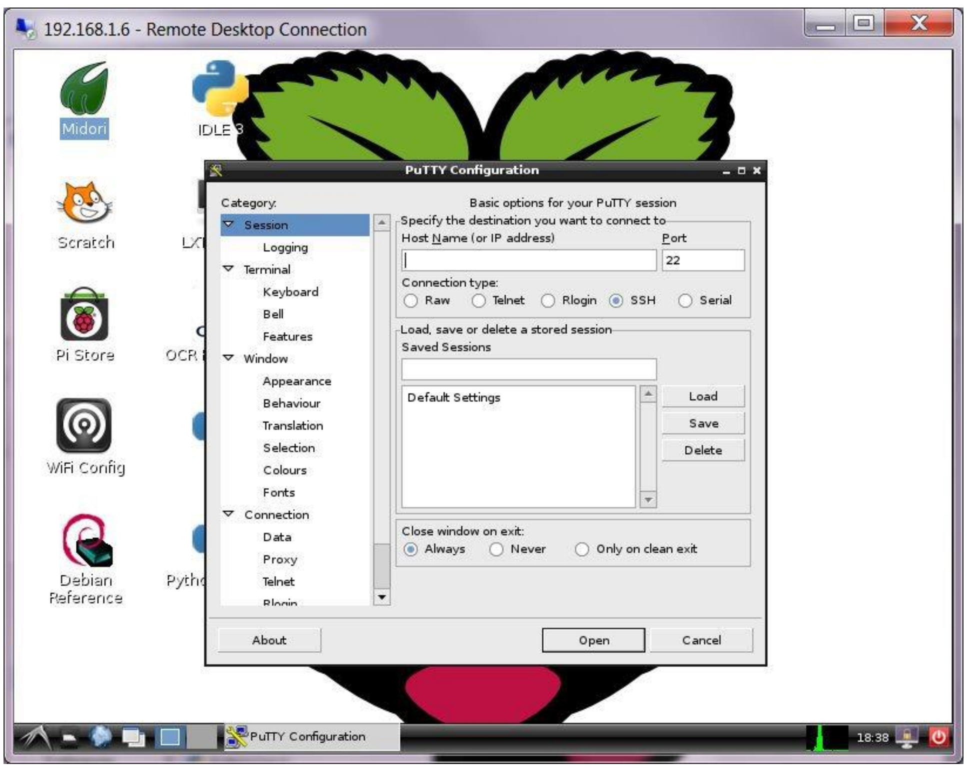


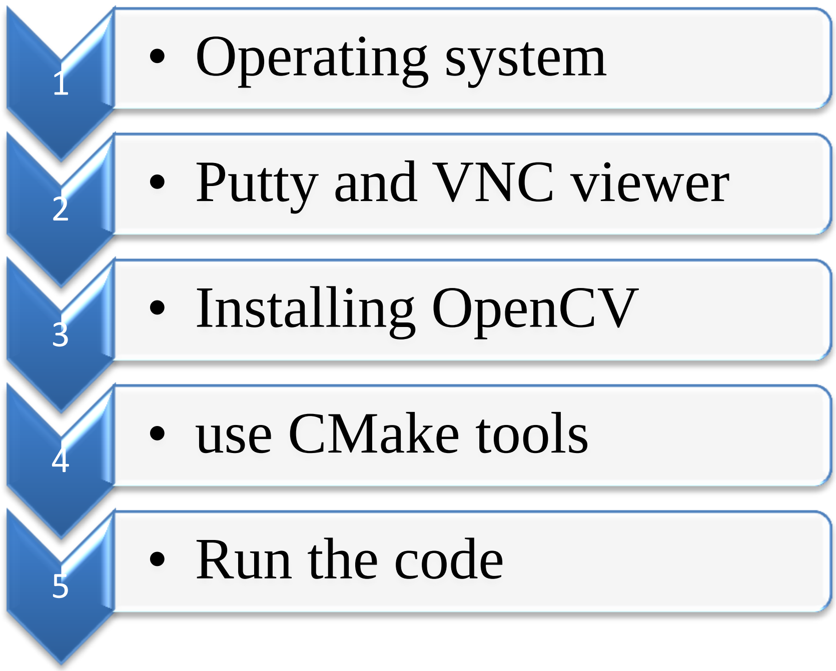
Figure (3.9) Using PUTTY

There are two main methods for building the kernel. You can build locally on a Raspberry Pi which will take a long time; or you can cross-compile, which is much quicker, but requires more setup.
In our project we will use cross compile method .

3.3 References

1- Root, Randal; Romero Sweeney, Mary (2006) visual studio
2- Itseez leads the development of the renowned computer vision library OpenCV http://itseez.com
Jump up ^ Bradski, Gary; Kaehler, Adrian (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc. p. 6. opencv
3- Maynard, Robert (24 March 2016) Kitware. Retrieved 2016-04- 09 cmake
4- https://en.wikipedia.org/wiki/Code::Blocks
5- blog on the following link https://www.raspberrypi.org/documentation/remote-access/vnc/ vnc viewer
6. blog on the following link http://blog.protoneer.co.nz/install-and-run-putty-on-your-raspberry-pi/Putty

4.1 Software on Raspberry pi



4.1.1 Operating system

We install the Raspbian as operating system on these steps :

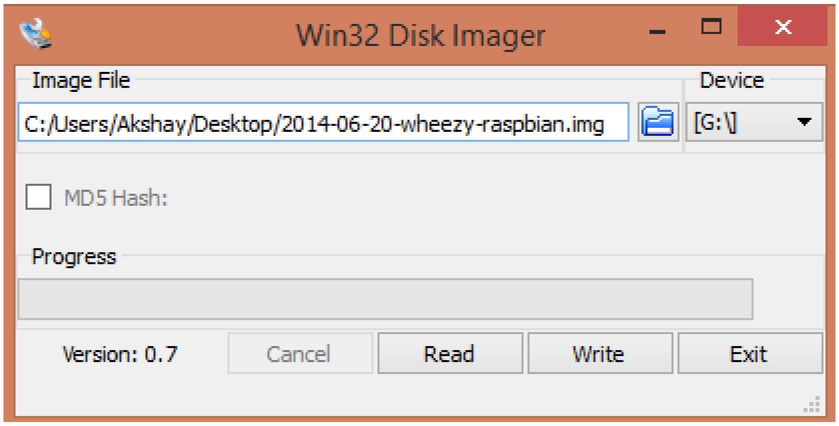
1- Downloading Raspbian and Image writer

Download the latest version of Raspbian , we will be needing an image writer to write the downloaded OS into the SD card (micro SD card in case of Raspberry Pi B+ model). So download the "win32 disk imager"

2- Writing the image

Insert the SD card into the laptop/pc and run the image writer. Once open, browse and select the downloaded Raspbian image file. Select the correct device, that is the drive representing the SD card. If the drive (or device) selected is different from the SD card then the other selected drive will become corrupted

After that, click on the "Write" button in the bottom. As an example, see +the image below, where the SD card (or micro SD) drive is represented by the letter "G:\"



Figure(4.1) writing the image

Once the write is complete, eject the SD card and insert it into the Raspberry Pi and turn it on. It should start booting up.

3- Setting up the pi

remember that after booting the Pi, there might be situations when the user credentials like the "username" and password will be asked. Raspberry Pi comes with a default user name and password and so always use it whenever it is being asked. The credentials are:

```
login: pi

password: raspberry
```

When the Pi has been booted for the first time, a configuration screen called the "Setup Options" should appear and it will look like the image below.

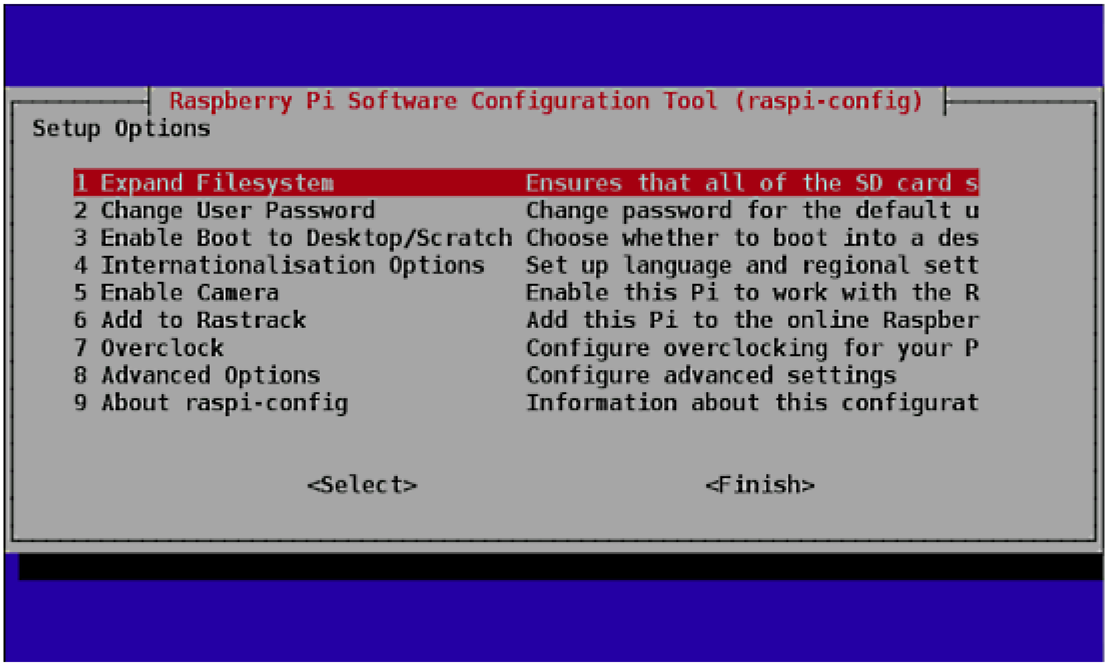


Figure (4.2) set up options

If you have missed the "Setup Options" screen, it’s not a problem, you can always get it by typing the following command in the terminal

```
sudo raspi-config
```

Once you execute this command the "Setup Options" screen will come up as shown in the image above.

Now that the Setup Options window is up, we will have to set a few things. After completing each of the steps below, if it asks to reboot the - Pi, please do so. After the reboot, if you don't get the "Setup Options" screen, then follow the command given above to get the screen/window.

- **The first thing to do:**

select the first option in the list of the setup options window, that is select the "Expand File system" option and hit the enter key. We do this to make use of all the space present on the SD card as a full partition. All this does is, expand the OS to fit the whole space on the SD card which - can then be used as the storage memory for the Pi.

- **The second thing to do:**

select the third option in the list of the setup options window, that is select the "Enable Boot To Desktop/Scratch" option and hit the enter key. It will take you to another window called the"choose boot option"

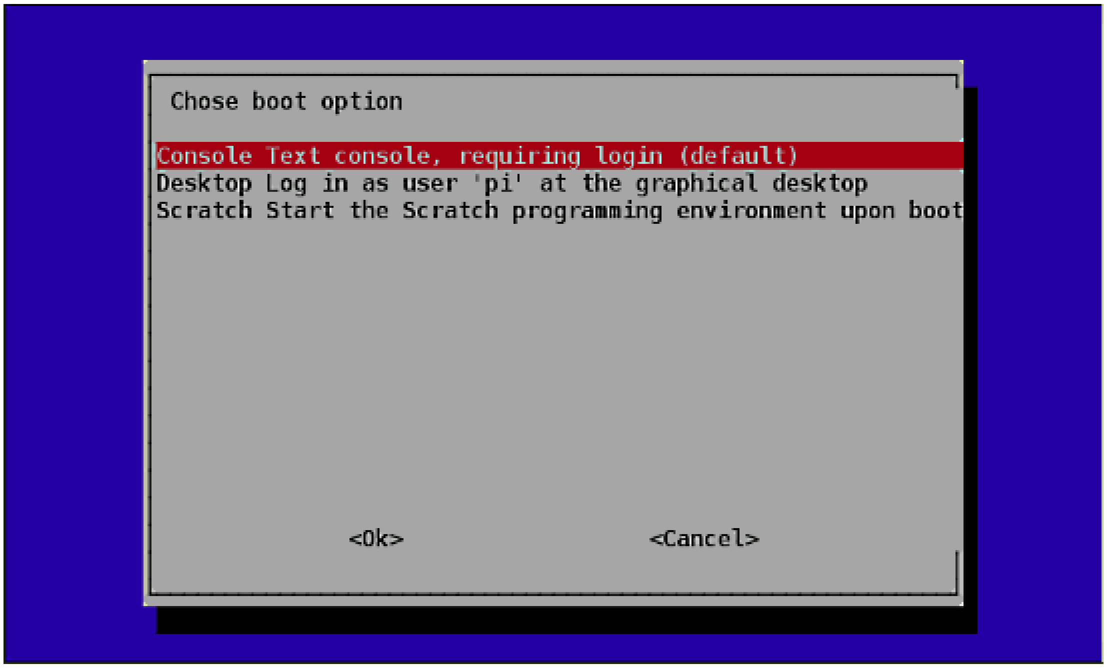


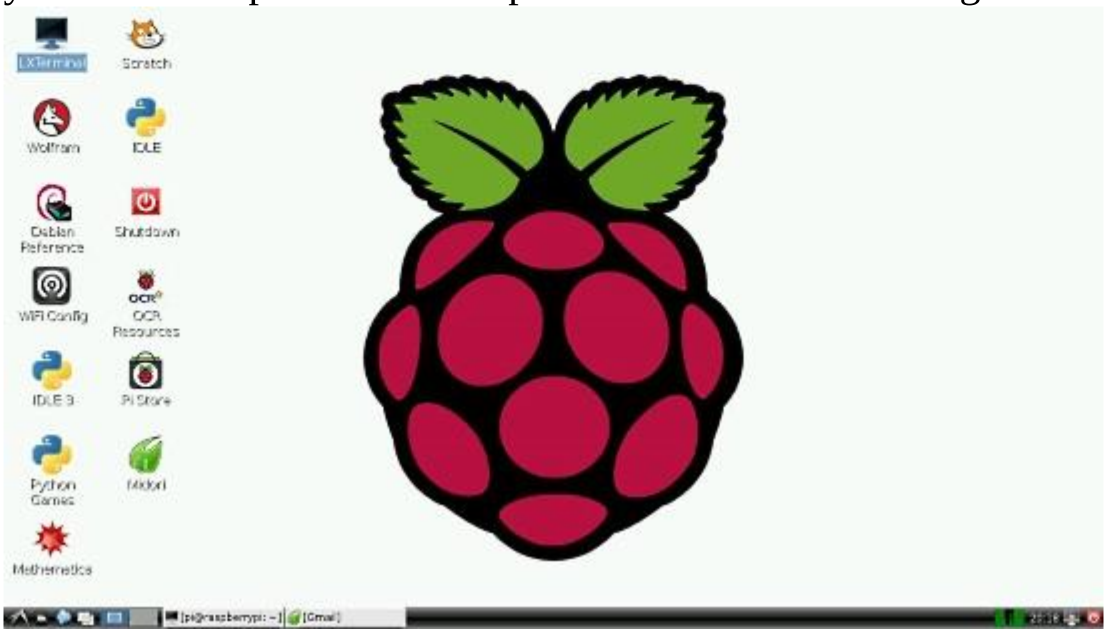
Figure (4.3) the boot option

- In the "choose boot option window" , select the second option, that is, "Desktop Log in as user 'pi' at the graphical desktop" and hit the enter button. Once done you will be taken back to the "Setup Options" page, if not select the "OK" button at the bottom of this window and you will be taken back to the previous window. We do this because we want to boot into the desktop environment which we are familiar with. If we don't do this step then the Raspberry Pi boots into a terminal each time with no GUI options.

Once, both the steps are done, select the "finish" button at the bottom of the page and it should reboot automatically. If it doesn't, then use the following command in the terminal to reboot.

4- Updating the firmware

After the reboot from the previous step, if everything went right, then you will end up on the desktop which looks like the image below.



Once you are on the desktop , open a terminal and enter following command to update the firmware of the Pi.

```
sudo rpi-update
```

Updating the firmware is necessary because certain models of the Pi might not have all the required dependencies to run smoothly or it may have some bug. The latest firmware might have the fix to those bugs, thus it's very important to update it in the beginning itself [1].

4.1.2 Interface between raspberry and pc

We use Putty programme to remote controle the raspberry from PC to dispense the lcd display with it's a commond line and VNC viewer to deal with raspberry , as we explain in chapter 3 .

4.1.3 Installing OpenCV 3 on Raspbian Jessie]

Step #1: Install dependencies

The first thing we should do is update and upgrade any existing packages, followed by updating the Raspberry Pi firmware.

Installing OpenCV 3 on Raspbian Jessie Shell

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo rpi-update
```

Timing: 3m 33s

You'll need to reboot your Raspberry Pi after the firmware update:

Installing OpenCV 3 on Raspbian Jessie Shell

```
$ sudo reboot
```

Now we need to install a few developer tools:

Installing OpenCV 3 on Raspbian Jessie Shell

```
$ sudo apt-get install build-essential git cmake pkg-config
```

Timing: 51s

Now we can move on to installing image I/O packages which allow us to load image file formats such as JPEG, PNG, TIFF, etc.:

Installing OpenCV 3 on Raspbian Jessie Shell

```
$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
```

Timing: 42s

Just like we need image I/O packages, we also need video I/O packages. These packages allow us to load various video file formats as well as work with video streams:

```
Installing OpenCV 3 on Raspbian Jessie                                    Shell
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
$ sudo apt-get install libxvidcore-dev libx264-dev
```

Timing: 58s

We need to install the GTK development library so we can compile the `highgui` sub-module of OpenCV, which allows us to display images to our screen and build simple GUI interfaces:

```
Installing OpenCV 3 on Raspbian Jessie                                    Shell
$ sudo apt-get install libgtk2.0-dev
```

Timing: 2m 48s

Various operations inside of OpenCV (such as matrix operations) can be optimized using added dependencies:

```
Installing OpenCV 3 on Raspbian Jessie                                    Shell
$ sudo apt-get install libatlas-base-dev gfortran
```

Timing: 50s

Step #2: Grab the OpenCV source code

At this point we have all of our prerequisites installed, so let's grab the `3.0.0` version of OpenCV from the OpenCV repository. (*Note*: As future versions of OpenCV are released just replace the `3.0.0` with the most recent version number):

```
Installing OpenCV 3 on Raspbian Jessie                                    Shell
$ cd ~
$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.0.0.zip
$ unzip opencv.zip
```

Timing: 2m 29s

For the full install of OpenCV 3 (which includes features such as SIFT and SURF), be sure to grab the opencv_contrib repo as well. (*Note:* Make sure your `opencv` and `opencv_contrib` versions match up, otherwise you will run into errors during compilation. For example, if I download v3.0.0 of `opencv` , then I'll want to download v3.0.0 of `opencv_contrib` as well):

```
Installing OpenCV 3 on Raspbian Jessie                                    Shell
$ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.0.0
$ unzip opencv_contrib.zip
```

Timing: 1m 54s

Step #4: Compile and install OpenCV

At this point, we are ready to compile OpenCV.

First, make sure you are in the `cv` virtual environment:

Installing OpenCV 3 on Raspbian Jessie — Shell

```
$ workon cv
```

Followed by setting up the build:

Installing OpenCV 3 on Raspbian Jessie — Shell

```
$ cd ~/opencv-3.0.0/
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D INSTALL_C_EXAMPLES=ON \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.0.0/modules \
    -D BUILD_EXAMPLES=ON ..
```

Update (3 January 2016): *In order to build OpenCV `3.1.0`, you need to set `-D INSTALL_C_EXAMPLES=OFF` (rather than `ON`) in the `cmake` command. There is a bug in the OpenCV v3.1.0 CMake build script that can cause errors if you leave this switch on. Once you set this switch to off, CMake should run without a problem.*

Before you move on to the compilation step, ***make sure you examine the output of CMake!***

Now that our build is all setup, we can compile OpenCV:

Installing OpenCV 3 on Raspbian Jessie — Shell
```
$ make -j4
```

Timing: 1h 35m

The `-j4` switch stands for the number of cores to use when compiling OpenCV. Since we are using a Raspberry Pi 2, we'll leverage all four cores of the processor for a faster compilation.

However, if your `make` command errors out, I would suggest starting the compilation over again and only using one core:

Installing OpenCV 3 on Raspbian Jessie — Shell
```
$ make clean
$ make
```

Using only one core will take much longer to compile, but can help reduce any type of strange race dependency condition errors when compiling.

Assuming OpenCV compiled without error, all we need to do is install it on our system:

Installing OpenCV 3 on Raspbian Jessie — Shell
```
$ sudo make install
$ sudo ldconfig
```

Step #5: Finishing the install

Step #6: Verifying your OpenCV 3 install

At this point, OpenCV 3 should be installed on your Raspberry Pi running Raspbian Jessie!

But before we wrap this tutorial up, let's verify that your OpenCV installation is working by accessing the `cv` virtual environment and importing `cv2`, the OpenCV + Python bindings:

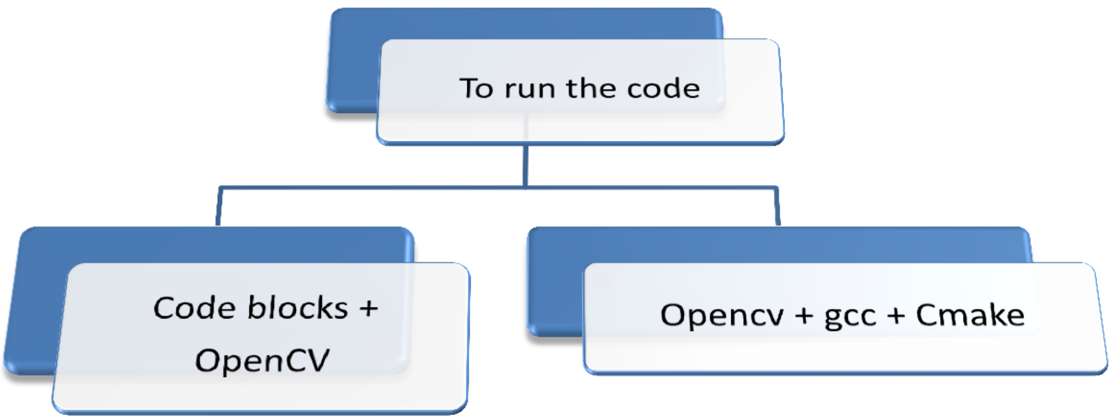
Installing OpenCV 3 on Raspbian Jessie — Shell
```
$ workon cv
$ python
>>> import cv2
>>> cv2.__version__
'3.0.0'
```

You can see a screenshot of my terminal below, indicating that OpenCV 3 has been successfully installed:

4.1.4 Using CMake tool

After we install openCV we use CMake tools and check the parameters of it

4.1.5 Run the cod



We use Code Blocks and OpenCV as it's a fast method .

```
#include <opencv2/opencv.hpp>
#include <iostream>

const int KEY_SPACE = 32;
const int KEY_ESC = 27;

CvHaarClassifierCascade *cascade;
CvMemStorage            *storage;

void detect(IplImage *img);

int main()
{
    std::cout << "Using OpenCV " << CV_MAJOR_VERSION << "." << CV_MINOR_VERSION << "." << CV_SUBMINOR_VERSION << std::endl;

    CvCapture *capture;
    IplImage  *frame;
    int input_resize_percent = 100;


    cascade = (CvHaarClassifierCascade*)cvLoad("cars.xml", 0, 0, 0);
    storage = cvCreateMemStorage(0);
    capture = cvCaptureFromAVI("video1.avi");

    assert(cascade && storage && capture);

    cvNamedWindow("video", 1);

    IplImage* frame1 = cvQueryFrame(capture);
    frame = cvCreateImage(cvSize((int)((frame1->width*input_resize_percent) / 100), (int)((frame1->height*input_resize_percent) / 100)), fram

    int key = 0;
    do
```

```
void detect(IplImage *img)
{
    CvSize img_size = cvGetSize(img);
    CvSeq *object = cvHaarDetectObjects(
        img,
        cascade,
        storage,
        1.1, //1.1,//1.5, //-------------------SCALE FACTOR
        1, //2        //-----------------MIN NEIGHBOURS
        0, //CV_HAAR_DO_CANNY_PRUNING
        cvSize(0, 0),//cvSize( 30,30), // ------MINSIZE
        img_size //cvSize(70,70)//cvSize(640,480)  //---------MAXSIZE
        );

    std::cout << "Total: " << object->total << " cars detected." << std::endl;
    for (int i = 0; i < (object ? object->total : 0); i++)
    {

        CvRect *r = (CvRect*)cvGetSeqElem(object, i);

        std::cout << r->width << std::endl;

            CvPoint Centerpt = cvPoint(r->x + r->width/2, r->y + r->height/2);
            if (r->width > (img->width / 2)){

                std::cout << "Be careful   \a !!";
            }

        cvRectangle(img,
            cvPoint(r->x, r->y),
            cvPoint(r->x + r->width, r->y + r->height),
            CV_RGB(0, 255, 255), 2, 8, 0);
    }
```

opencvTestProgram | (Global Scope)

```
    do
    {
        frame1 = cvQueryFrame(capture);

        if (!frame1)
            break;

        cvResize(frame1, frame);

        detect(frame);

        key = cvWaitKey(10);

        if (key == KEY_SPACE)
            key = cvWaitKey(0);

        if (key == KEY_ESC)
            break;

    } while (1);

    cvDestroyAllWindows();
    cvReleaseImage(&frame);
    cvReleaseCapture(&capture);
    cvReleaseHaarClassifierCascade(&cascade);
    cvReleaseMemStorage(&storage);

    return 0;
}
```

4.2 EXPERIMENTAL SETUP

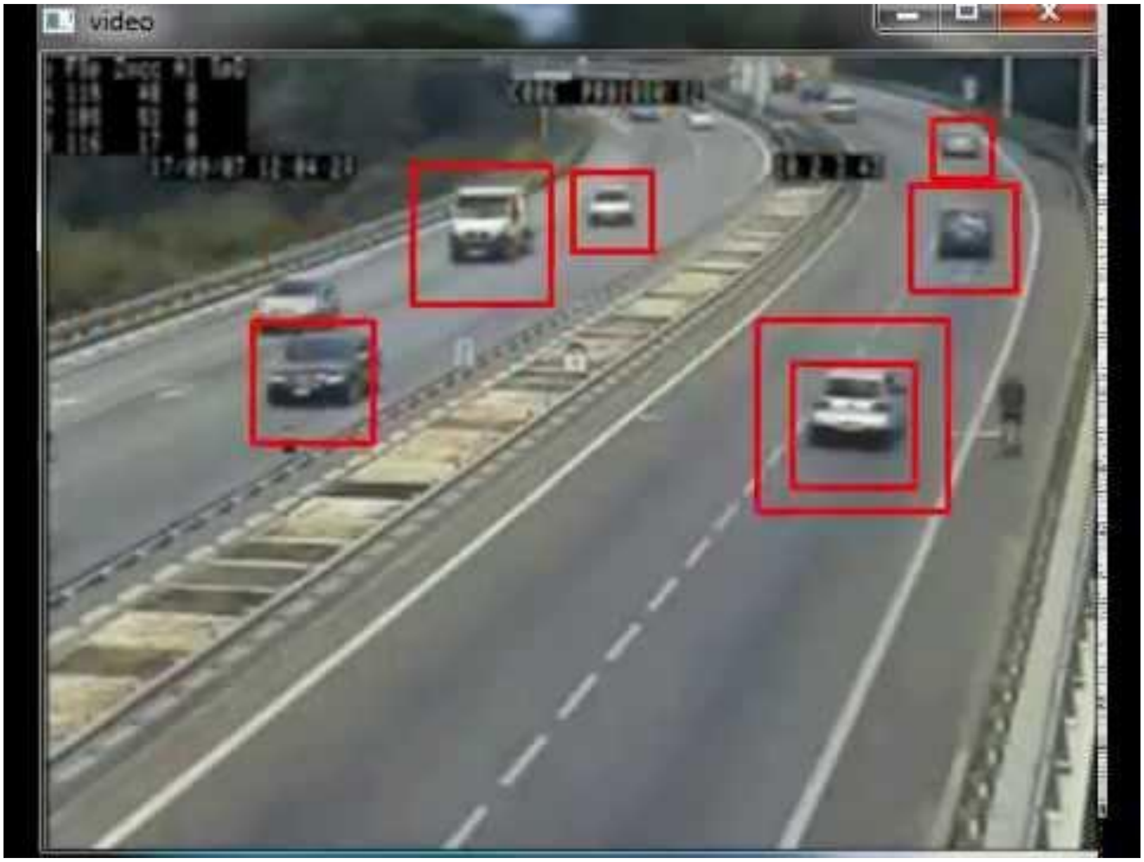


Figure (4.4) test image

4.2 References

1. https://www.howtoforge.com/tutorial/howto-install-raspbian-on-raspberry-pi/
2. http://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/

CONCLUSIONS AND FUTURE WORK

We introduce a system with low cost and real time image processing . An illustrative comparative analysis is performed on the test video captured by USB webcam and the Haar cascade classifier operations are performed in OpenCV on Raspberry Pi board. Based on the results it can be concluded that detection technique provides better and legible edges than its other counterparts. This can be seen from the results that are obtained after performing detection the results of which leads to distinguish objects and better depth analysis of the image. This entire work has been performed in OpenCV by adopting c++ as the programming language.

The future scope of this above work is to a generate a novel distance measuring algorithm based on canny edge detection principle to measure the distance between the objects and vehicles and efficiently minimize the risk of accidents especially in areas densely infested with people .We can use RADAR and LIDAR to overcome weather changes to make a better performance for getting a detailed images.

Reference

ARPN Journal of Engineering and Applied Sciences©2006-2015 Asian Research Publishing Network (ARPN)

MANSOURA UNIVERSITY - FACULTY OF ENGINEERING

كلية الهندسة - جامعة المنصورة

Vehicle Collision Avoidance System