



République Algérienne Démocratique Et Populaire

Université kasdi merbah Ouargla

**Faculté des Nouvelles Technologies de l'Information et
de la Communication**

Département d'Electronique et des Télécommunications



Mémoire de fin d'étude

Présenté pour l'obtention du **diplôme** de **MASTER**

Domain : Science et Technologies

Spécialité : Electronique

Option : Electronique des systèmes embarqués

Réalisé par :

Mazouzi Abdessatar

Kouinini Ilyes

Sujet :

**Realization of an object color tracker based
on the raspberry pi board**

Membres du jury composé de :

M. AOUF Anouar essadate

MAA

Président

UKM Ouargla

M. ABDESSEMED Djalal

MAA

Examineur

UKM Ouargla

M. BENACER Imad

MCB

Encadreur

UKM Ouargla

2019/2020

Dedication

*With a special feeling of gratitude, I dedicate this work to my beloved family
and to those who were the source of my inspiration.*

To those who set me on the path of life, my dear parent.

To my dear mother who has never stopped praying for me.

To my dear father for his encouragement and support.

To all professors and teachers who followed me along my way to success.

To my dear brothers and sisters.

To all my friends.

Mazouzi Abdessatar

*Every challenging work needs self-effort as well as guidance of elders
especially those who are very close to your heart.*

*My humble effort i dedicate to my sweet and loving father and mother,
whose affection, love, encouragement and prays of day and night make me
able to get such success and honor.*

And to all professors and teachers who followed me along my way to success.

To my dear brothers and sisters.

To all my friends.

Kouinini Ilyes

Contents

List of Figures	vi
List of Tables	x
Acronyms	xi
General Introduction	1
I State of the art in autonomous mobile robots	3
I.1 Introduction	3
I.2 Definition and history of robots	3
I.2.1 Definition of a robot	3
I.2.2 Robot history	4
I.2.2.1 Ancient and preindustrial period	4
I.2.2.2 Industrial and robosapiens Period	5
I.3 Autonomous mobile systems	7
I.3.1 Autonomous mobile systems classifications	7
I.3.1.1 Ground mobile systems	7
I.3.1.2 Aerial mobile systems	8
I.3.1.3 Water and underwater mobile systems	8
I.3.2 Autonomous mobile systems components	8
I.3.3 Mobile systems applications	8
I.4 Ground robot locomotion	9
I.4.1 Legged locomotion	9
I.4.1.1 Bipedal robots	9
I.4.1.2 Many-leg robots	10
I.4.2 Wheeled locomotion	11
I.4.2.1 Wheel types	11
I.4.2.2 Drive types	13

I.5	Mobile robot sensors	17
I.5.1	Sensor classification and characteristics	17
I.5.1.1	Sensors classification	17
I.5.1.2	Sensor characteristics	17
I.5.2	Position and velocity sensors	18
I.5.2.1	Position sensors	18
I.5.2.2	Velocity sensors	18
I.5.3	Distance sensors	19
I.5.3.1	Sonar sensors	19
I.5.3.2	Infrared sensors	20
I.5.3.3	Laser sensors	21
I.5.4	Robot vision	21
I.6	Conclusion	21
II	Introduction to Digital Image Processing	22
II.1	Introduction	22
II.2	What is digital image processing?	22
II.3	The origin of digital image processing	22
II.4	Image analysis tasks	23
II.4.1	Digital image processing levels	25
II.4.1.1	Low-level Processing	25
II.4.1.2	Mid-level processing	25
II.4.1.3	High-level processing	25
II.5	Digital Images	25
II.5.1	Definition of Digital image	25
II.5.2	Image Pixel	26
II.5.3	Preview of digital image	26
II.5.4	Image histogram	27
II.5.5	Binary image	27
II.6	Color image processing	28
II.6.1	Color Fundamentals	29
II.6.2	Color Models	29
II.6.2.1	The RGB color spaces	30
II.6.2.2	The HSV color space	30
II.6.2.3	The CMY color space	30
II.7	Application of digital image processing	31
II.8	Constraints of digital image processing	31
II.9	Conclusion	32

III Hardware and Software description	33
III.1 Introduction	33
III.2 Hardware part	33
III.2.1 The Raspberry pi	33
III.2.1.1 What is Raspberry Pi?	34
III.2.1.2 Raspberry Pi components	34
III.2.1.3 Raspberry Pi released models	35
III.2.1.4 GPIO on the Raspberry Pi	37
III.2.1.5 Review Raspberry Pi 3 model B+ used in our project	38
III.2.2 Raspberry Pi Camera	39
III.2.2.1 Raspberry Pi Camera rev 1.3	40
III.2.3 Arduino board	40
III.2.3.1 what is the arduino?	41
III.2.3.2 Arduino Nano	43
III.2.4 Dc motors	44
III.2.4.1 Equivalent circuit of DC motor	44
III.2.4.2 DC Gearbox Motor "TT Motor"	45
III.2.5 Motor Drives	46
III.2.5.1 DC Motor Driver	46
III.2.5.2 The MX1508 Motor Drive	46
III.2.6 distance sensors	47
III.2.6.1 Sharp IR GP2Y0A41SK0F distance sensor	48
III.2.7 Camera Module Pan Tilt Mount	49
III.2.8 The 4W robotic vehicle chassis	49
III.2.8.1 description	49
III.2.8.2 Features	50
III.2.8.3 kit assembly	50
III.3 Software part	50
III.3.1 Raspberry Pi Operating System	50
III.3.1.1 Raspbian Linux OS	51
III.3.2 Python Programming language	52
III.3.2.1 Python and Artificial Intelligent	53
III.3.3 An introduction to OpenCV	53
III.4 Conclusion	54
IV Raspberry Pi setup, and components assembly	55
IV.1 Introduction	55
IV.2 Raspberry Pi Setup	55
IV.2.1 Installing the Raspbian OS on the Raspberry Pi	55

IV.2.1.1	Download Raspbian	55
IV.2.1.2	Writing Raspbian on SD card	56
IV.2.1.3	Connection Raspberry Pi	57
IV.2.2	Raspberry Pi configuration	58
IV.2.2.1	Using Raspi-config	58
IV.2.2.2	change the Raspbian Hostname	59
IV.2.2.3	Enable Raspberry Pi features	60
IV.2.3	Setup the the remote access	61
IV.2.3.1	Remote desktop with xrdp	61
IV.2.3.2	Connect to a wireless network	62
IV.2.3.3	Using Windows remote desktop	62
IV.3	Installing OpenCV in raspbian OS	63
IV.3.1	Python in raspbian	63
IV.3.2	Installing OpenCV	63
IV.4	Set up the Pi camera module	64
IV.4.1	Installing the Pi camera module	64
IV.4.2	Programming the Pi Camera Module	66
IV.5	Interfacing Arduino with Raspberry Pi	67
IV.5.1	serial communication	68
IV.5.2	Connecting to Serial	69
IV.5.3	Sending data through Serial	69
IV.5.4	Reading data using Arduino	69
IV.6	Control the wheels with Arduino	70
IV.6.1	Connection	70
IV.6.2	Control speeds the wheel	71
IV.6.3	Pulse width Modulation (PWM)	71
IV.6.4	programme Arduino board to control the wheels	72
IV.6.4.1	Arduino script code:	72
IV.7	Sharp IR with Arduino	75
IV.7.1	Connections	75
IV.7.2	Arduino code	75
IV.8	Robot parts installation	76
IV.9	Conclusion	77
V	Object tracking using color detection with raspberry pi board	78
V.1	Introduction	78
V.2	Color feature	79
V.3	Color spaces conversions	80
V.4	Comparative of color detection in different color spaces	81

V.4.1	RGB color space	82
V.4.2	YCrCb color space	83
V.4.3	HSV color space	83
V.4.4	Results of color detection	85
V.5	Path planning for color abject following	88
V.6	Conclusion	89
	CONCLUSION	90
	Bibliography	91
	Annex	94
	ABSTRACT	98

List of Figures

I.1	Model of a bicycle and a programmable cart built based on Leonardo da Vinci's notes.	5
I.2	The wheeled mobile robot Shakey of SRI	6
I.3	The Waseda University robots (a) WABOT-2, (b) WABOT-1.	6
I.4	The Dante II explorer robot	7
I.5	The Honda's humanoid ASIMO.	7
I.6	NASA's Robonaut robot based on human appearance	10
I.7	The quadruped robot "Kotetsu" (leg length at standing 18-22 cm).	10
I.8	Conventional wheels: (a) Powered fixed wheels, (b) Castor wheels	11
I.9	Three designs of a universal wheel.	12
I.10	the design of Mecanum wheel.	12
I.11	A practical implementation of a ball wheel.	13
I.12	Locomotion of differential drive. (A) the straight path, (B) curved path, (C) circular path	13
I.13	Tricycle WMR locomotion modes (A, B, C, D), (E) a tricycle example	14
I.14	Omnidirectional WMRs, (A), Three-wheel case, (B), Four-wheel case.	14
I.15	Standard setup of a four-mecanum-wheel omnidirectional WMR.	15
I.16	Six basic locomotion modes of a WMR with four mecanum wheels	15
I.17	(A) Illustration of synchro drive WMR motion, (B) illustration of the two Synchronization belt systems, and (C) a synchro drive WMR example.	16
I.18	(A) Illustration of the effective contact point and (B) a typical tracked platform	16
I.19	An encoder disk with 16 white and 16 black segments.	19
I.20	A DC tachometer	19
I.21	(A) A commercial sonar sensor. (B) A cone of about 30° is mapped by each sonar sensor. (C) If the obstacle in front of the sensor is far from the perpendicular, then no signal will be reflected toward the sensor and so the object will not be detected.	20

I.22	Infrared distance measurement uses the phase-shift method.	20
II.1	(a) a digital picture produced in 1921 from a coded tape by a telegraph printer with 5 grey levels, (b) Unretouched cable picture of Generals Pershing (right) and Foch, transmitted in 1929 from London to New York, with 15 grey levels.	23
II.2	four possible levels of image representation suitable for image analysis problems in which objects have to be detected and classified.	24
II.3	Representation of digital images by arrays of discrete points on a rectangular grid	26
II.4	(Left) Image of a flower, (right) Subimage of 8x8 pixels from the flower image	27
II.5	(a) an grey-scale image. (b) brightness histogram of that image	28
II.6	A binary Image	28
II.7	Color spectrum has seen by passing white light through a prism	29
II.8	Wavelengths comprising the visible range of the electromagnetic spectrum.	29
II.9	Schematic of the RGB color cube. Points along the main diagonal have grey values, from black at the origin to white at point (1, 1, 1).	30
II.10	the HSV color space illustrated as a cylinder and unfolded cylinder.	31
III.1	Raspberry PI components	34
III.2	The Raspberry Pi Model B board	35
III.3	The Raspberry Pi Model B+ board	36
III.4	The Raspberry Pi 3 board	36
III.5	The Raspberry Pi Zero board	37
III.6	The Raspberry Pi's GPIO port and its pin definitions	37
III.7	Raspberry Pi 3 model B+	38
III.8	Physical specifications of Raspberry Pi 3 Model B+	38
III.9	Raspberry Pi Camera rev 1.3	40
III.10	labelled Arduino Uno board	41
III.11	Labelled Arduino IDE	42
III.12	Arduino Nano Board	43
III.13	Conventional (brushed) d.c. motor scheme	44
III.14	Equivalent circuit of a DC Motor.	45
III.15	DC Gearbox Motor "TT Motor".	45
III.16	Motor drive bloc diagram	46
III.17	The MX1508 Motor Drive and the MX1508 IC circuit	47
III.18	Distance sensor's position control of a mobile robot	48
III.19	Sharp IR GP2Y0A41SK0F	48

III.20Pi Camera Mount Module Pan Tilt	49
III.21(a)unassembled 4w Robotic Vehicle Chassis, (b) assembled 4w robotic vehicle chassis	50
III.22Raspbian Desktop	52
IV.1 Raspbian download page	56
IV.2 SDFormatter	57
IV.3 Win32 Disk Imager	57
IV.4 Raspberry pi connections	58
IV.5 Raspbian Terminal widow	59
IV.6 The Raspbian configuration table	59
IV.7 Network Options of the Raspbian OS.	60
IV.8 Interfacing Options of the Raspbian OS.	61
IV.9 Windows Remote desktop Connection.	62
IV.10XRDP remote desktop login screen.	63
IV.11The Raspberry Pi 3 Model B CSI connectors.	65
IV.12The Camera Module's ribbon cable connected with Raspberry via CSI connectors.	65
IV.13Picture captured via Raspberry Pi camera	67
IV.14Arduino Nano with Raspberry Pi	68
IV.15DC Motors controlled with Arduino board using the MX1508 Motor drive.	70
IV.16DC Motor Speed Control using Variable resistor	71
IV.17different PWM Signals.	72
IV.18sharp GP2Y0A41SK0F connections with Arduino Nano board	75
IV.19connection of the 4 wheels with the motor drives	77
IV.20Complete assembled robot	77
V.1 (a) color image, (b) grayscale image, (c) binary image, (d) grayscale his- togram.	79
V.2 result of threshold for two images taken under different illumination.	80
V.3 Flow Chart for Object Tracking by Color Detection	81
V.4 Different Channels Blue (B), Green (G), Red (R) of the RGB color space shown separately.	82
V.5 histogram of R, G, B for the same images with different lightness.	82
V.6 YCrCb color space components for two image with different lightness conditions.	83
V.7 Histogram Y, Cr, Cb component for two images with different illumina- tion.	84
V.8 Hue (H), Saturation (S) and Value (V) components in HSV color space.	84

V.9 Histogram of the HSV color space component for different colors. . . .	85
V.10 detection of different colors in the RGB color space.	86
V.11 detection of different colors in the YCrCb color space.	87
V.12 detection of different colors in the HSV color space.	87
V.13 Find contours and draw a red rectangle around the objects.	88
V.14 color abject following algorithm.	89
V.15 Technical data sheet of Raspberry Pi camera v1.3	94
V.16 Arduino Nano Pin Layout.	95
V.17 Technical data sheet of the MX1508 DC motor drive	96
V.18 Technical data sheet of GP2Y0A41SK0F	97

List of Tables

II.1	Representation of the intensity level at every pixel in the 8x8 sub-image.	27
III.1	the specifications of the Raspberry Pi 3 Model B+	39
III.2	Specifications of the Raspberry Pi camera rev 1.3	40
III.3	Arduino Nano specifications	43
III.4	technical specification of the DC Gearbox Motor - "TT Motor" - 200RPM - 3 to 6VDC	45
V.1	RGB min and max threshold values of red, blue, green, yellow colors. . .	85
V.2	Y, Cr, Cb min and max threshold values of red, blue, green, yellow colors.	86
V.3	H, S, V min and max threshold values of red, blue, green, yellow colors.	86

Acronyms

- **AC:** Alternating current
- **AGV:** Autonomous guided vehicle
- **AMR:** Autonomous mobile robotics
- **ARM:** Advanced RISC Machine,
- **ASIC:** Application specific integrated circuit
- **ASIMO:** Advanced Step in Innovative Mobility
- **AVR:** Alf and Vegard's RISC
- **BC:** Before Christ
- **CMY:** (Cyan, Magenta, Yellow).
- **CPU:** Central processing unit
- **CRT:** Cathode Ray Tubes
- **CSI:** Camera Serial Interface
- **DC:** Direct current
- **DIP:** Digital image processing
- **DIY:** Do it your self
- **DSI:** Display Serial Interface
- **EMF:** Electromotive force
- **FPS:** Frame per second
- **GB:** Gigabyte

- **GH:** Gigahertz
- **GPIO:** General-purpose input/output
- **GPS:** Global positioning system
- **GPU:** Graphics processing unit
- **GUI:** Graphical user interface
- **HDMI:** High-Definition Multimedia Interface
- **HSV:** (Hue, Saturation, Value).
- **I2C:** A serial communication protocol
- **IC:** Integrated circuit
- **IDE:** Integrated development environment
- **IEEE:** Institute of Electrical and Electronics Engineers
- **IR:** Infra-red
- **LAN:** Local Area network
- **LASER:** Light Amplification by Stimulated Emission
- **LDR:** Light Dependent Resistor
- **LED:** light-emitting diode
- **LIDAR:** Light direction and ranging
- **LMR:** Legged Mobile robot
- **LPDDR:** Low-Power Double Data Rate Synchronous Dynamic Random Access
- **MB:** MegaByte
- **MCU:** Microcontroller unit
- **MMF:** Magnetomotive force
- **NOOBS:** New Out of the Box Software
- **OpenCV:** Open Source Computer Vision
- **OpenGL:** Open Graphics Library.

- **OS:** Operating System
- **POE:** Power of Ethernet
- **PWM:** (pulse-width modulation)
- **RAM:** Random Access memory
- **RCA:** Radio Corporation of America
- **RGB:** Red, Green, Blue
- **RIA:** Robotics Institute of America
- **RISC:** Reduced instruction set computer
- **RPI:** Raspberry Pi
- **RPM:** Rotation Per Minute
- **SD:** Secure Digital
- **SDRAM:** Synchronous Dynamic Random Access Memory
- **SOC:** System on ship
- **SONAR:** Sound navigation and ranging sensors
- **SPI:** Serial Peripheral Interface.
- **SRI:** Stanford Research Institute
- **SSH:** Secure Shell
- **UAV:** Unmanned aerial vehicles
- **USB:** Universal Serial Bus
- **UV:** Ultraviolet radiation
- **VNC:** Virtual Network Computing
- **WiFi:** Wireless Fidelity
- **WMR:** Wheeled Mobile Robots
- **XRDP:** Remote Desktop Protocol

General Introduction

Robotics has made their way into our lives, that due to the fast development of technology in this field, robotics have been a dominant area of researches, from a variant types of robots, Autonomous Mobile Robots (AMRs) have achieved the greatest success over the recent years and it takes a big place in the world of research and development nowadays, on the other hand, earlier robots regardless of their success they have a big disadvantage which was: the lack of mobility and autonomous navigation, furthermore, AMRs can move around their environment and they are distinguished in a variant types already in use some of which are the wheeled mobile robots (WMRs), AMRs are designed for a structured environment to rely on a guided path to navigate around the desired workspace, mobile robots were designed for point-to-point tasks where the path doesn't change over the time. Nowadays researchers are focusing on the development of a mobile robots that can navigate in their workspace autonomously.

Autonomous mobile robots can't perform any task without communication with the outside, it needs to collect useful data from the environment to use that data to make decisions, and these data are collected using sensors, there are a wide variety of sensors used in mobile robots some which are: distance sensors, velocity and position sensors, global position system (GPS), gyroscope sensor and robot vision. Each type is distinguished in several other types classified according to their principle of working and has different characteristics, for example, distance sensors are distinguished in Ultrasonic, Infrared and laser sensors each one of them has its own characteristics. Sensors either used to measure simple values like temperature, speed ... etc. or to acquire more complex information about the robot's environment like robot vision. Robot vision provides robots with a powerful automation capability than other technology could, in simple term robot vision is the combination of using camera and computer vision and image processing algorithms which enable to process the visual data and extract information from the surrounded environment, which make mobile robots capable of identifying their path, avoid barriers, identify specific objects ... etc.

and perform other tasks autonomously. that process includes motion tracking, object recognition . . . etc.

The objective of this work is to design a prototype of an autonomous mobile robot that capable of identifying and track objects by their colors using robot vision, which uses a camera to acquire visual information of the surrounded environment, this information analyzed and processed based on image processing algorithms to detect objects according to their colors, by applying color segmentation method using python programming language and OpenCV.

This robot consists of several components (Raspberry Pi, Arduino board, IR proximity sensor, Camera) and each one plays an important role, this robot has a powered fixed four wheels' chassis, as a processing unit we used the Raspberry Pi 3 Model B+ with a camera module utilized as a vision of this robot, the speed and direction of the robot wheels are controlled via an Arduino Nano board which receive orders from the Raspberry Pi via serial communication, in addition to the camera, the robot equipped with a Sharp IR distance sensor to measure the distance between robot and objects continuously.

Organization of the memory

This work is organized into five chapters. In Chapter I, we present the State of art of autonomous mobile robot, in order to better understand the work present in this memory. In Chapter II, we present an introduction to image processing. In Chapter III, we present briefly the different Hardware and software description that we use in this work. In Chapter IV and V, we introduce a Raspberry Pi setup, and components assembly, the robot was tested in Linux operating system and executed in Raspberry Pi. The algorithm used was built-in OpenCV and Python as the programming language

Finally, we will conclude this work with a general conclusion in which we will discuss about obtained results and performance of our robot, and give our perspective and proposals for feature enhancement, with the view that this work will help as an introduction for further researches.

State of the art in autonomous mobile robots

I.1 Introduction

Autonomous mobile robotics (AMR) field is a popular field of research and development nowadays. These robots are designed to make human life easier and to do tasks that human cannot do (e.g. exploring dangerous or contaminated areas), mobile robots are capable to move around their environment and not fixed to a physical location, unlike those industrial robots that can move only in a specific workspace, this mobility makes them suitable for a large scale of applications, mobile robots are distinguished in Wheeled Mobile Robots (WMRs) and Legged Mobile Robots (LMRs) that also include unmanned aerial vehicles (UAVs), and wheeled autonomous robots are very popular because they have a low mechanical complexity and energy consumption. Mobile robots have already in use in our homes in the form of robotic floor cleaners and autonomous lawnmowers, or automatic guided vehicles (AGV) and the prototype of the self-driving car is already driving, and shortly many applications of autonomous mobile robots are expected to appear.

Robotics science is a combination of several engineering branches and that includes mechanical and electronics engineering and computer science, an autonomous mobile robot is a field of robotics that focuses on designing fully autonomous robots, that are capable to performs tasks without external influence.

I.2 Definition and history of robots

I.2.1 Definition of a robot

The roots of The term robot is in Slavic languages. The meaning of the word “robota” in Polish is work or labour, and in Czech or Slovenian language it means statute

labour.and The word mobile has the roots in the Latin word of the same meaning, “mobilis.” [1]

Technologically there is no unique definition of a Robot. When Joseph Engelberger, the father of modern robotics was asked to define a robot he said: “I can’t define a robot but I know one when I see one.” The Robotics Institute of America (RIA) defines an industrial robot as “a reprogrammable multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks which also acquire information from the environment and move intelligently in response.” This definition does not capture mobile robots. [2]

In conclusion, literately a robot is referring to a machine that performs actions, and an autonomous robot is a programmed machine that can do work without human influence in s specific workspace, and an autonomous mobile robot is a machine that is programmed and moves around its workspace with the ability to extract information from the environment, today’s mobile robot can understand speech, recognize objects and plan their path by themselves in real-time.

I.2.2 Robot history

The history of robots is divided into two main periods[2]:

- Ancient and preindustrial period
- Industrial and Robosapiens period

I.2.2.1 Ancient and preindustrial period

Around 2500-300 BC the first robot in the worldwide history was in the Greek mythological mechanical called Talos. and the mechanical artificial entity constructed by Hephaestus, under the order of Zeus, Talos body was made with bronze and had a single vein from the neck up to the ankle, Talos was gifted by Zeus to Europe who gave him afterwards to her son Minos. Talos name was given to Asteroid 5786 discovered by Robert McNaught on September 31, 1991. [2]

The Earlier were no present robotic in fiction, there are some inventors tried to construct some mechanical automata, the Greek mathematician Archytas constructed a mechanical bird propelled by steam in the 4rth century BC. In Leonardo da Vinci’s notes, several mechanical designs can be found, some of his inventions has reconstructed such as a programmable cart and model of bicycle show in Figure I.1.[1]

The fast evolution of industrial technology nowadays led to the robotics we know today



Figure I.1: Model of a bicycle and a programmable cart built based on Leonardo da Vinci's notes.

I.2.2.2 Industrial and robosapiens Period

Industrial period starts in 1954 which was the first modern robot, George Devol patented his multijointed robotic arm. In 1956, together with Joseph Engelberger founded the world's first robot company called Unimation:

1961: The first industrial robot called Unimate joined a die-casting production line at General Motors.

1963: The RanchoArm, the first computer-controlled robotic arm, was put in operation at the Rancho Los Amigos Hospital (Downey, CA). This was a prosthetic arm designed to aid the handicapped.

1969: The first truly flexible arm, known as the Stanford Arm, was developed in the Stanford Artificial Intelligence Laboratory by Victor Scheinman. This arm soon became a standard and is still influencing the design of today's robotic manipulators.

1970: This is the starting year of mobile robotics. The mobile robot Shakey was developed at the Stanford Research Institute (today is known as SRI Technology), controlled by intelligent algorithms that observe via sensors, and react to their actions (Figure I.2).

1980-1989: This decade was dominated by the development of advanced Japanese robots, especially walking robots (humanoids). Among them, the WABOT-2 humanoid is mentioned (Figure I.3a) which was developed in 1984, representing the first attempt to develop a personal robot with a "specialist purpose" to play a keyboard musical instrument, rather than a versatile robot-like WABOT-1 (Figure I.3b).

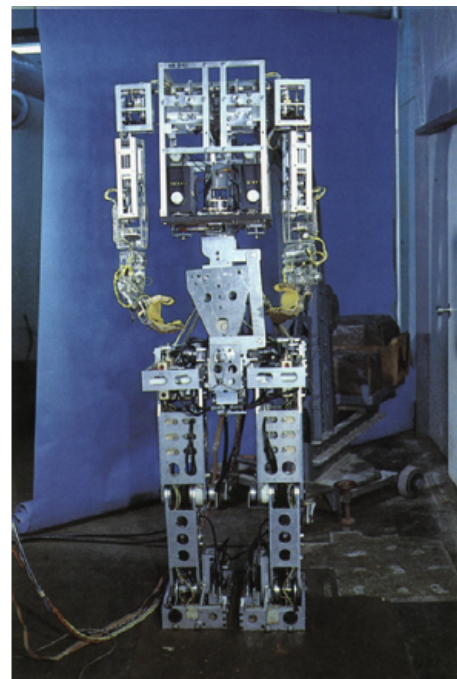
1990-1999: During this decade the emergence of "explorer robots" took place. These robots went where the human did not visit before or it was considered too risky



Figure I.2: The wheeled mobile robot Shakey of SRI



(a)



(b)

Figure I.3: The Waseda University robots (a) WABOT-2, (b) WABOT-1.

or inconvenient. Examples of such robots are Dante (1993) and Dante II (1994) that explored Mt. Erebus in Antarctica and Mt. Spurr in Alaska (Figure I.4).

2000-Present: In the 2000s the development of numerous new intelligent mobile robots capable of almost fully interacting with humans recognizing voices, faces and gestures, and expressing emotions through speech, dexterous walking or performing household chores, hospital chores, microsurgeries and the like, is continuing with growth rates, for example, HONDA humanoid ASIMO (2000) shown in Figure I.5.



Figure I.4: The Dante II explorer robot



Figure I.5: The Honda's humanoid ASIMO.

I.3 Autonomous mobile systems

Mobile systems can be defined as systems that are not attached to the environment and can move in a certain space.[1]

I.3.1 Autonomous mobile systems classifications

Mobile systems can be classified into three principal groups according to the environment that the move across:

I.3.1.1 Ground mobile systems

In this group, several types can be found such as mobile vehicles, legged robots (humanoids) or robots that mimic some other type of animal locomotion. A ground

mobile system with wheels that are not carrying the operator is often referred to as unmanned ground vehicles.

I.3.1.2 Aerial mobile systems

Here we can find mobile systems that can fly in certain aerial space such as aeroplanes, helicopters, drones, rockets, animal-mimicking flying systems when an Aerial mobile system used without a pilot this referred to unmanned aerial vehicles.

I.3.1.3 Water and underwater mobile systems

In this group, we find different types of ships, boats, submarines, autonomous underwater vehicles, etc.

I.3.2 Autonomous mobile systems components

The main mechanical and electronic parts of an autonomous mobile robot are the following:

- **Mechanical parts:** Rigid and moving parts (body, wheels, tracks, legs, etc.)
- **Actuators:** Electrical motors (DC, stepper, servomotor, etc.)
- **Sensors:** Rotation encoders, proximity and distance sensors, inertial navigation unit, global navigation satellite system, etc.
- **Computers:** Micro-controllers, portable personal computer, embedded systems, etc.
- **Power unit:** power supplies such as Batteries, solar panels, etc.
- **Electronics:** electronic components such as capacitors, transistors, integrated Circuits (ICs), Actuator drives etc.

I.3.3 Mobile systems applications

The use of mobiles systems nowadays is expanding due to rapid technological development. They are used in various applications in different areas some are mention below:[1]

- **medical services:** operation support, laboratory analyses etc.
- **cleaning applications:** (floor vacuuming, sweeping and mopping in homes or large buildings, window cleaning, etc.).

- **applications in agriculture:** like automated fruit picking, seed planting, grass mowing, etc.
- **inspection and surveillance of hazardous areas:** (detection and deactivation of mines in minefields, review of nuclear reactors, cleaning of sewer pipes).
- **space missions:** (satellites, inspection and servicing of satellites, planetary exploration).
- **cargo handler robots:** for loading and unloading goods or materials from aeroplanes, ships, or trucks.
- **military robots:** (reconnaissance robots, aeroplanes, and various autonomous projectiles).
- **security robots:** (security guards to control storage, buildings, etc.)

I.4 Ground robot locomotion

Locomotion of ground robots is distinguishing in two principal groups.[2]

- Legged locomotion
- Wheeled locomotion

I.4.1 Legged locomotion

Most of the creators on the planet have legs locomotion, this can be distinguished in two legs or multilegged, Biological multilegged organisms can move in diverse and difficult environments with obstacles, insects can't be achieved as an artificial creator due that they have a small size and strong robustness. The legged robot must be statically stable to use them in real-life. There are two main categories of legged robot locomotion: Two-leg (bipedal) robots and Many-leg robots.

I.4.1.1 Bipedal robots

Two-leg locomotion is consisting of two legs for walking and running such as Humanoids which are considered a bipedal robot with an appearance based on human body (head, legs, arms etc.). such as in the NASA's Robonaut shown in figure I.6 , while others have also a face with eyes And mouth.

Robot with bipedal locomotion needs complex interaction of mechanical and control system features. During locomotion, the legs need to be lifted from and return to the ground. The sequence and way of placing and lifting each foot are synchronized with the body moves to move from one place to another.



Figure I.6: NASA's Robonaut robot based on human appearance

I.4.1.2 Many-leg robots

The initial research on multilegged walking robots was focused on robot locomotion design for smooth or easy rough terrain, bypassing simple obstacles, motion on soft ground, body manoeuvring, and so on. These requirements can be realized via periodic gaits and binary (yes/no) contact information with the ground. Newer studies are concerned with multilegged robots that can move over an impassable road or an extremely complex terrain, such as mountain areas, ditches, trenches, earthquake-damaged areas, etc. In these cases, additional capabilities are needed, as well as detailed support reactions and robot stability prediction. Figures I.7 show an advanced multilegged robot with the capabilities of the above type. The quadruped robot Kotetsu of Figure I.7 is capable of adaptive walking using phase modulations based on leg loading/unloading.

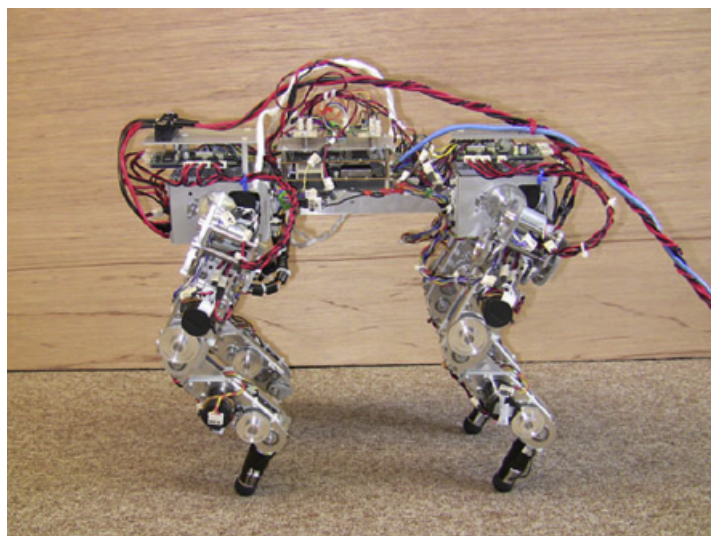


Figure I.7: The quadruped robot "Kotetsu" (leg length at standing 18-22 cm).

I.4.2 Wheeled locomotion

Robots with wheeled locomotion have been widely used because there are principal reasons to achieve wheeled mobility including the simple structure, energy efficiency, fast speed, low fabrication cost.

The mobility and manoeuvrability of a WMR depend on the wheels (number and type of wheels used) and drive used.[2] wheeled mobile robots are characterized to maximal manoeuvrability which is needed for planar motions, such as operating a warehouse floor, a road, a hospital, a museum, etc. wheeled mobile robots are widely used in many applications. In general, wheeled robots consume less energy and move faster than other locomotion mechanisms, less control effort is required, due to their simple mechanisms and reduced stability problems.

I.4.2.1 Wheel types

The wheels used in WMRs are divided into two types: [2]

- Conventional wheels
- Special wheels

I.4.2.1.1 Conventional wheels: These wheels are distinguished in powered fixed wheels, castor wheels, and powered steering wheels. Powered fixed wheels (Figure I.8a) are driven by motors mounted on fixed positions of the vehicle. Their axis of rotation has a fixed direction concerning the platform's coordinate frame, Castor wheels (Figure I.8b) are not powered but they can also rotate freely about an axis perpendicular to their axis of rotation.



(a)



(b)

Figure I.8: Conventional wheels: (a) Powered fixed wheels, (b) Castor wheels

I.4.2.1.2 Special wheels: These wheels are designed such that to have activated traction in one direction and passive motion in another, thus allowing greater manoeuvrability in congested environments, there are three main types of special wheels:

- **Universal wheel:** provides a combination of constrained and unconstrained motion during turning. It contains small rollers around its outer diameter which are mounted perpendicular to the wheel's rotation axis. This way the wheel can roll in the direction parallel to the wheel axis in addition to the normal wheel rotation (Figure I.9).



Figure I.9: Three designs of a universal wheel.

- **Mecanum wheel:** is similar to the universal wheel except that the rollers are mounted at an angle other than 90 (usually $\pm 45^\circ$) (Figure I.10). this type of wheels gives the robot ability to move in any directions by spinning the Mecanum wheels in certain direction Due to the force F produced by the rotation of the wheel acts on the ground via roller that has contact with the. At this roller, the force is decomposed in a force F_1 parallel to the roller axis and a force F_2 perpendicular to the roller axis.

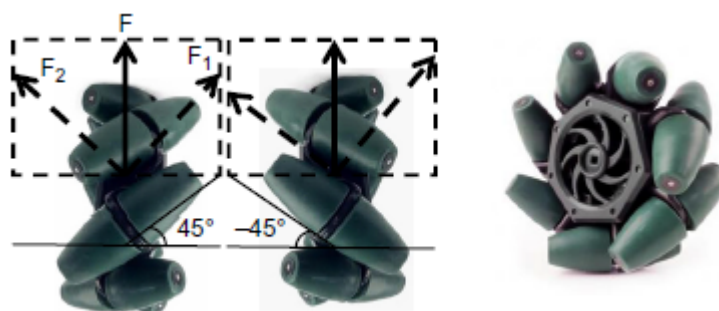


Figure I.10: the design of Mecanum wheel.

- **Ball wheel:** The ball wheel places no direct constraints on the motion, that is, it is omnidirectional like castor or special universal and Mecanum wheels. In other words, the rotational axis of the wheel can have any arbitrary direction. One way to achieve this is by using an active ring driven by a motor and gearbox to transmit power to the ball via rollers and friction, which is free to rotate in any direction instantaneously. The ball wheel is shown in figure I.11.

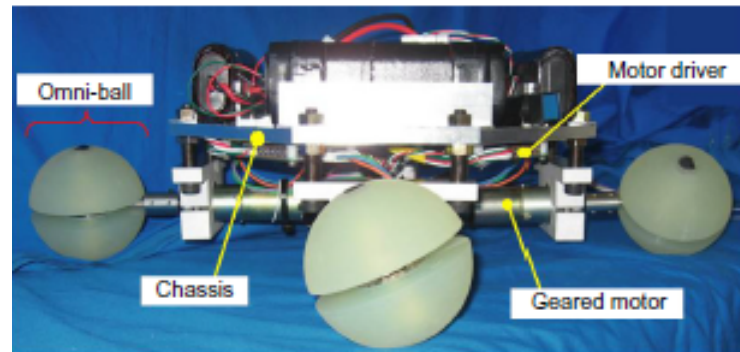


Figure I.11: A practical implementation of a ball wheel.

I.4.2.2 Drive types

The drive of wheeled mobile robot is referring to how the robot is driving and moving. The drive types of WMRs are distinguished in: (Differential, drive Tricycle, Omnidirectional, Synchro drive, Ackerman steering, Skid steering).[2]

I.4.2.2.1 Differential drive: This drive consists of two fixed powered wheels mounted on the left and right side of the robot platform. The two wheels are independently driven. One or two passive castor wheels are used for balance and stability. this type of drive is a simple mechanical drive and it does not need to rotate a driven axis, to move straight forward needs to rotate the powered wheels at the same speed, if one wheel is rotating faster than the other wheel, the robot turn left or right depending on which wheel is running faster. If the two wheels are running at the same speed and in the opposite direction, the robot turns about the midpoint of the wheels. the Differential drive illustration is shown in figure I.12.

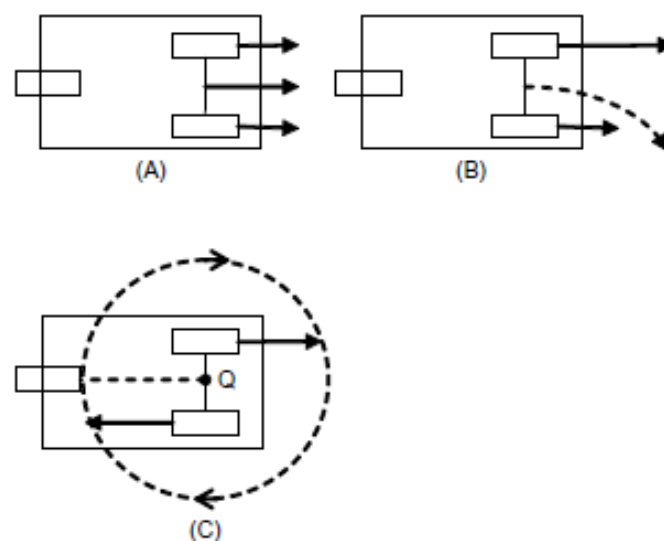


Figure I.12: Locomotion of differential drive. (A) the straight path, (B) curved path, (C) circular path

I.4.2.2.2 Tricycle drive: this type has a single front wheel driven and powered (Figure I.13), in the back, there are two fixed unpowered wheels for stability, the front wheel has the control of directions and speed of the robot, for driving straight forward the wheel is positioned in the middle position with the desired speed (Figure 1.13A). When the front wheel is at an angle the vehicle follows a curved path (Figure 1.13B). if the driven wheel is positioned at 90°, the robot rotates circularly and the centre is in the middle point of the fixed wheels (Figure 1.13C), the WMRs with Tricycle drive cannot achieve parallel parking directly but by some manoeuvres with forward and backward movement as shown in Figure 1.13D.

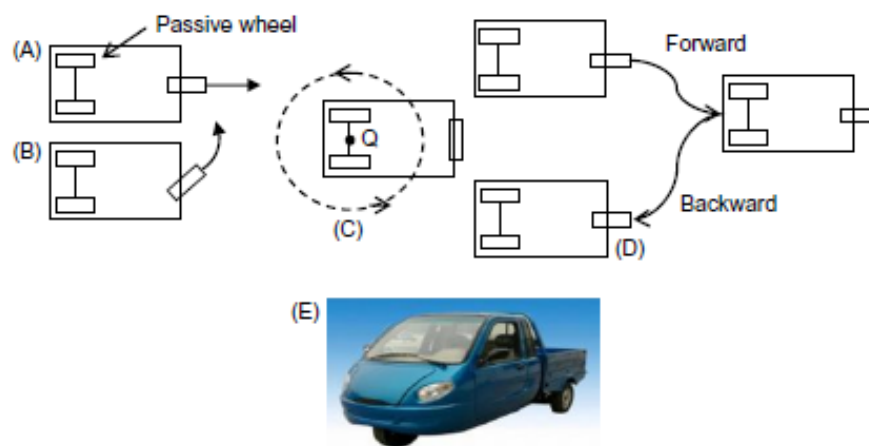


Figure I.13: Tricycle WMR locomotion modes (A, B, C, D), (E) a tricycle example

I.4.2.2.3 Omnidirectional drive: This drive can be obtained using three, four, or more omnidirectional wheels as shown in Figure I.14. WMRs with three wheels use universal wheels that have a 90° roller angle (Figure I.9) as shown in Figure 1.14A. Omnidirectional WMRs with four wheels use Mecanum wheels (Figure I.10) in the configuration shown in Figure 1.14B.

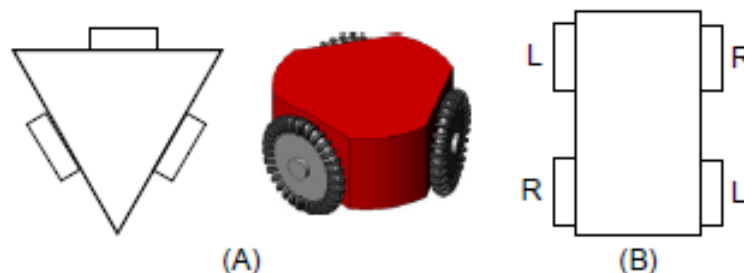


Figure I.14: Omnidirectional WMRs, (A), Three-wheel case, (B), Four-wheel case.

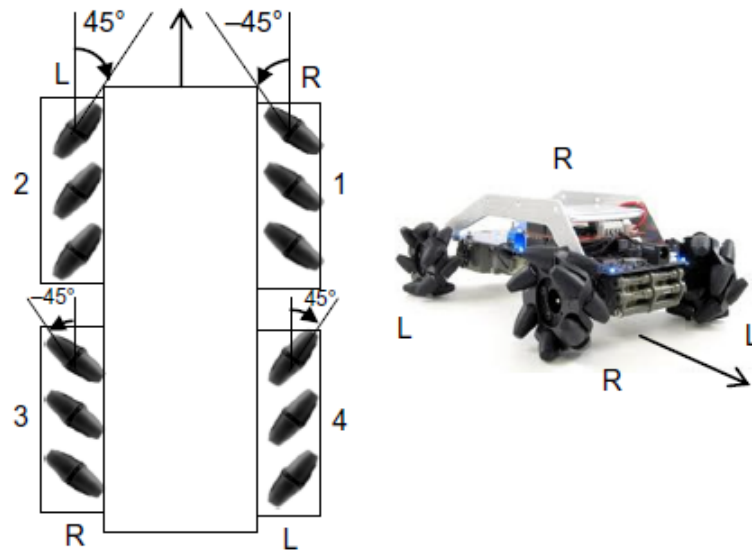


Figure I.15: Standard setup of a four-mecanum-wheel omnidirectional WMR.

Omnidirectional WMRs with four Mecanum wheels shown in Figure I.15. It's able to move in any direction, Figure I.16 shows six basic motions of a four-wheel omnidirectional robot, namely (A) forward motion, (B) left sliding, (C) clockwise turning (on the spot), (D) backward motion, (E) right sliding, and (F) anticlockwise turning.

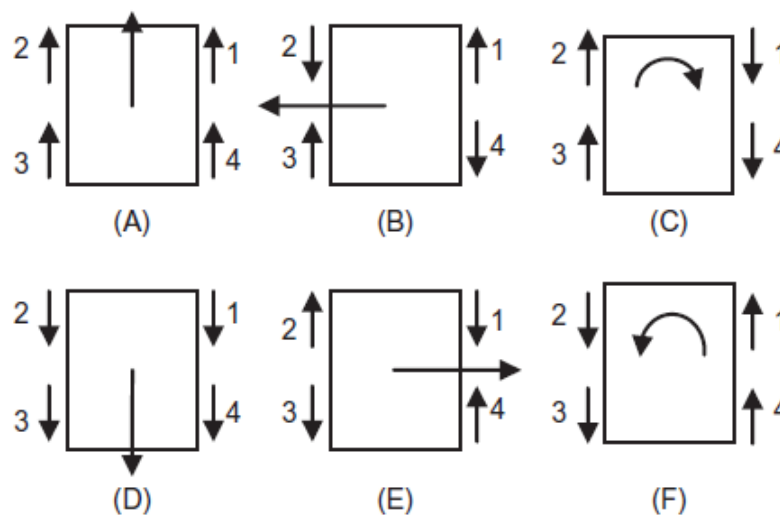


Figure I.16: Six basic locomotion modes of a WMR with four mecanum wheels

I.4.2.2.4 Synchro drive: This drive has three or more wheels that are mechanically coupled such that all of them rotate in the same direction at the same speed, this mechanical steering synchronization can be realized in several ways. for example, using a chain, a belt or gear drive. Synchro drive WMR can move in any direction and

cannot drive and rotate at the same time, to change directions this WMR must stop and realign its wheels. Figure I.17.

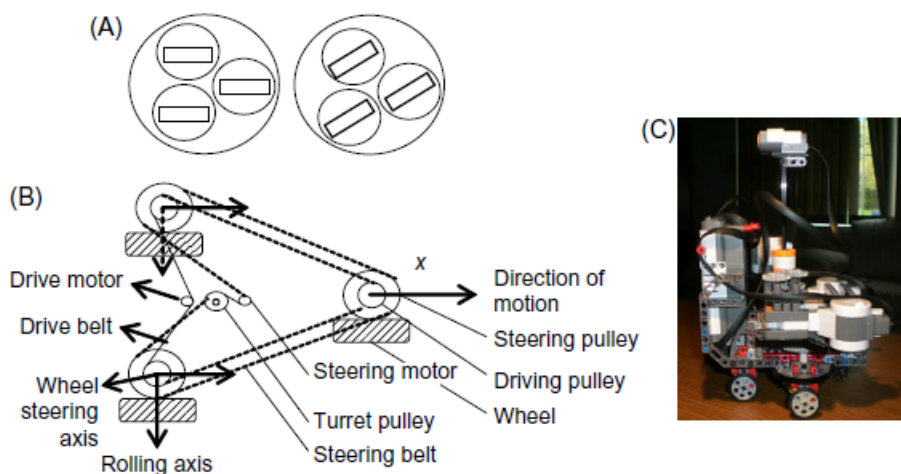


Figure I.17: (A) Illustration of synchro drive WMR motion, (B) illustration of the two Synchronization belt systems, and (C) a synchro drive WMR example.

I.4.2.2.5 Ackerman steering: This is the standard steering used in automobiles. It consists of two combined driven rear wheels and two combined steered front wheels.

I.4.2.2.6 Skid steering: This is a special implementation of differential drive and realized in track form on bulldozers and armoured vehicles. Its difference from a differential drive vehicle is the increased manoeuvrability in uneven terrains and the higher friction which is due to its tracks and the multiple contact points with the terrain (Figure I.18). Figure 1.18A illustrates that the effective point contact for such a skid-steer robot is roughly constrained on either side by a rectangular uncertainty area which corresponds to the track footprint. Figure 1.18B shows a typical tracked robotic platform that can carry a manipulator or special exploration equipment.



Figure I.18: (A) Illustration of the effective contact point and (B) a typical tracked platform

I.5 Mobile robot sensors

The sensors are designed based on the human sensory system such as vision, hearing, touching. etc. this sensor provides inputs signals to the robot brain for processing, and according to these inputs, the robot performs actions such as move, stop. etc. the sensors are importance for autonomous operations, mobility, real-time applications, and intelligence capabilities.

I.5.1 Sensor classification and characteristics

I.5.1.1 Sensors classification

In general, robotic sensors are distinguished into two groups [2]:

- Analogue sensors.
- Digital sensors.

Analogue sensors: analogue sensors provide analogue output signals, which need analogue-to-digital conversion (A/D), an example of analogue sensors are analogue infrared distance sensor, microphone, LDR etc.

Digital sensors: digital sensors provide analogue output signals, which are more accurate than the analogue sensors, and may be of a different form. For example, they may have a “synchronous serial” form (i.e., bit by bit data reading), or a “parallel form” (e.g., 8 or 16 digital output lines).

I.5.1.2 Sensor characteristics

Sensors used in mobile robots have a variety of performance characteristics, the basic characteristics of sensors are as follows:[2]

- **Dynamic range:** is the range between the lower and upper limits of input values for which the sensor is working normally. the dynamic range defined as the algorithm of the ratio of the maximum and minimum values:

$$Range = 20\log\left(\frac{Maximum\ input}{Minimum\ input}\right)(dB)$$

- **Resolution:** the minimum difference of the measured variable that can be recognized by the sensor.

- **Linearity:** it's referred to the property that the output value $F(x + y)$ of a sensor to a sum of inputs x and y are equal to the sum $F(x) + F(y)$ of the output values of the sensor obtained separately by each input.
- **Bandwidth:** the maximum rate or frequency of readings data the sensor can provide. The number of readings (measurements) per second provides the sensor's frequency measured in hertz.
- **Sensitivity:** the degree to which the changes in the input signal affect the output signal.
- **Accuracy:** it refers to the difference between the sensor reading and the true value.

I.5.2 Position and velocity sensors

I.5.2.1 Position sensors

The use of a position sensor in robots is to determine whether the joints of a robot or a mobile platform have moved to the correct position. An example of two basic position sensors are the following: [2]

- Potentiometers
- Encoder

The potentiometer: (linear or angular) is a device that gives an output $V_0(t)$ proportional to the position of the pointer, that is, $V_0 = K_p \theta(t)$, where K_p is the potentiometers coefficient.

The encoder: Encoders constitute a basic feedback sensor for motor control. There are two typical techniques for building an encoder is the Hall effect sensor (magnetic sensor) and the optical encoder (a sector disk with black and white segments together with a light-emitting diode (LED) and a photo-diode; Figure I.19). The photodiode detects reflected light during a white segment, but not during a black segment. Therefore, if this disk has 16 white and 16 black segments, the sensor will receive 16 pulses during a revolution.

I.5.2.2 Velocity sensors

The measurement of the velocity of robot joints or mobile platform is typically done with the aid of tachometers, Tachometers are distinguished in direct current (DC) and alternate current (AC) tachometers. In robotics, the DC tachometer is mostly used

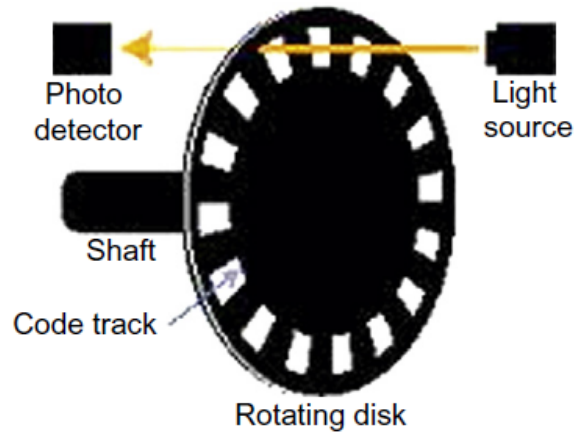


Figure I.19: An encoder disk with 16 white and 16 black segments.

(Figure I.20). The DC tachometer generates a DC voltage at its output proportional to the speed of the motor connected to it. [2]



Figure I.20: A DC tachometer

I.5.3 Distance sensors

Distance sensors which measure the distance between the robot and the object around it, are distinguished in the following types: [2]

- Sonar sensors
- Laser sensors
- Infrared sensors

I.5.3.1 Sonar sensors

Sonar Sensors (or, sound navigation and ranging sensors), these sensors have relatively a narrow cone coverage, so for 360° , typically a mobile robot uses multiple sonar sensors to cover 360° . The working principle of a sonar sensor consists of the

emission of a short acoustic signal (of duration about 1 ms) at an ultrasonic frequency of 50-250 kHz, the signal reflects on an object then the sensor receives the reflected signal, and measure the time from signal emission until the echo returns to the sensor. Figure I.21.

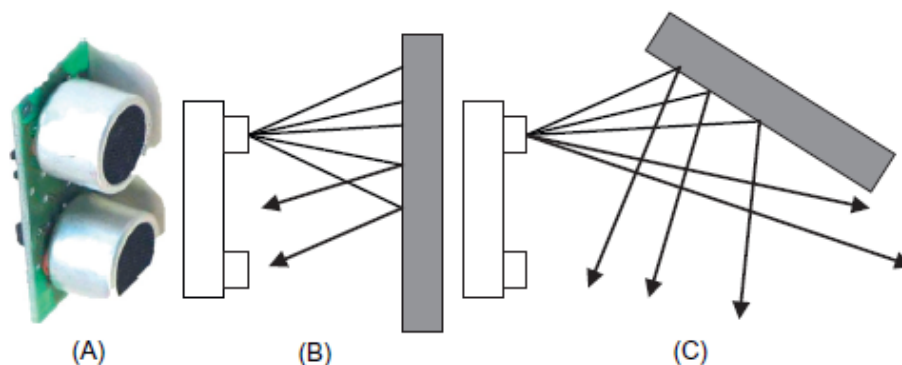


Figure I.21: (A) A commercial sonar sensor. (B) A cone of about 30° is mapped by each sonar sensor. (C) If the obstacle in front of the sensor is far from the perpendicular, then no signal will be reflected toward the sensor and so the object will not be detected.

I.5.3.2 Infrared sensors

Near-infrared light can be produced by LED or a laser, Infrared light has a wavelength between 820 and 880 nm. The sensor sends an amplitude modulated light at a given frequency f and measures the phase shift between the transmitted and reflected signals, (Figure I.22) if ϕ is the measured phase shift and γ the wavelength, then the distance D is equal to:

$$D = \left(\frac{\gamma}{4\pi}\right) \cdot \phi$$

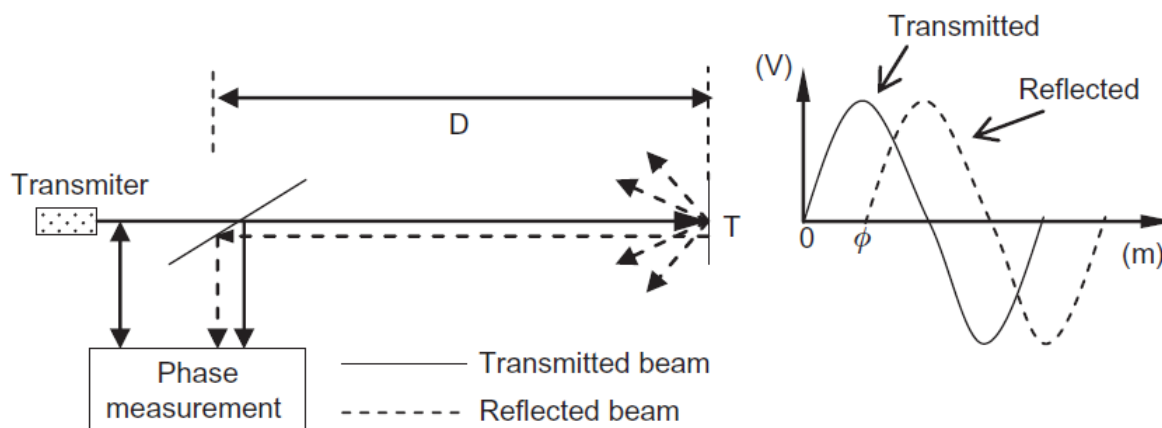


Figure I.22: Infrared distance measurement uses the phase-shift method.

I.5.3.3 Laser sensors

Laser distance sensors are a special case of optical sensors have a range from centimetres to meters. Also, they called laser radars or Lidars (light direction and ranging). Energy is emitted at impulses. The distance is computed from the time-of-flight. Laser sensors provide speed and accuracy, Rotary lasers are based on the rotation of the wave 360o at more than 1 or 2 rpm, this overcomes the problem of hidden areas.

I.5.4 Robot vision

Robot vision can be defined as the ability of a robot to see and recognize objects via the image reflected by those objects with the aid camera, then the images captured via the robot camera need to be interrupted in a result to extract information and features. Robot vision employs optical or visual sensors (cameras) and proper electronic equipment (CPU) to process/analyze visual images and recognize objects of concern in each robotic application.

Sensors are windows for a robot to the environment, they allow robots to understand and measure the geometric and physical properties of objects in their work environment, there are some other important robotic sensors such as Gyroscope, GPS, force sensors etc. [2]

I.6 Conclusion

Robots of all types have been a dominant contributor to the development of human society and economy over the years. Earlier robot were devices with the ability to perform simple tasks in very specialized workspace. Nowadays robots are autonomous and intelligent and cover a very large repertory of the form including wheeled robots, mobile manipulators. Legged robots, animal-like robots etc. and these robots are used to contribute to the development of several fields such as medical, industrial, agricultural, technical, and social life.

Introduction to Digital Image Processing

II.1 Introduction

Computer vision and digital image processing is a well-known technology nowadays, it's a method to perform operations on an image to get an enhanced image or to extract information and features from it. Image processing is a technology used to make an artificial system which its input data obtained from digital images.

processing and understanding visual information is a huge field of research and development nowadays. Intelligent computer vision technology aid humans to interpret image and videos. And the development of embedded systems and computer vision has enabled the implementation of vision-based systems that can detect, recognize and track objects in a real-time.

II.2 What is digital image processing?

In computer science, digital image processing is the use of a digital computer to process digital images through an algorithm. [3], The digital image processing technic used in two principal application areas:

- Extracting information and Improvements of pictorial information for human interpretation.
- Processing images data for storage, transmission, and representation for autonomous machine perception.

II.3 The origin of digital image processing

One of the earliest applications of digital images was in the newspaper industry when pictures were first sent by submarine cable between London and New York.

Introduction of the Bartlane cable picture transmission system in the early 1920s reduced the time required to transport a picture across the Atlantic from more than a week to less than three hours. This technic used to transmit digital newspaper images.[5] Figure II.1a shows digital picture produced in 1921 from a coded tape by a telegraph printer with 5 grey levels [4]

There was a problem of the image quality at that time due to intensity of grey level in image pixel, the image initially was coded with 5 grey levels, which is not good quality, then this was improved to 15 grey levels by 1929. [6] Figure II.1b shows a digital image with 15 grey levels.



(a)



(b)

Figure II.1: (a) a digital picture produced in 1921 from a coded tape by a telegraph printer with 5 grey levels, (b) Unretouched cable picture of Generals Pershing (right) and Foch, transmitted in 1929 from London to New York, with 15 grey levels.

At that time the digital computer was not invented yet, although the previous examples cited above are digital images, this can't be considered as digital image processing, the history of DIP intimately tied to the development of digital computers. Since this field requires much data storage and powerful processor, digital displays, and transmission technology.

The first digital computer, powerful enough to carry out meaningful image processing tasks appeared in the early 1960s. The birth of what we call digital image processing today can be traced to the availability of those machines.[3]

II.4 Image analysis tasks

Image understanding by a machine can be seen as an attempt to find a relation between input image(s) and previously established models of the observed world. The transition from the input image(s) to the model reduces the information contained

in the image to relevant information for the application domain, this process is usually divided into several levels, and several image representations are used passing from the raw image to the data extracted from that image, computer vision creates these intermediate representation within the process to use them to establish the final model.[5]

According to the data organization image representation can be divided into four levels, (Figure II.2), a bottom-up way of information processing, start from signals has almost no abstraction to the high abstraction needed for image understanding, three levels of processing are distinguished a low-level processing, Mid-level processing and high-level processing.

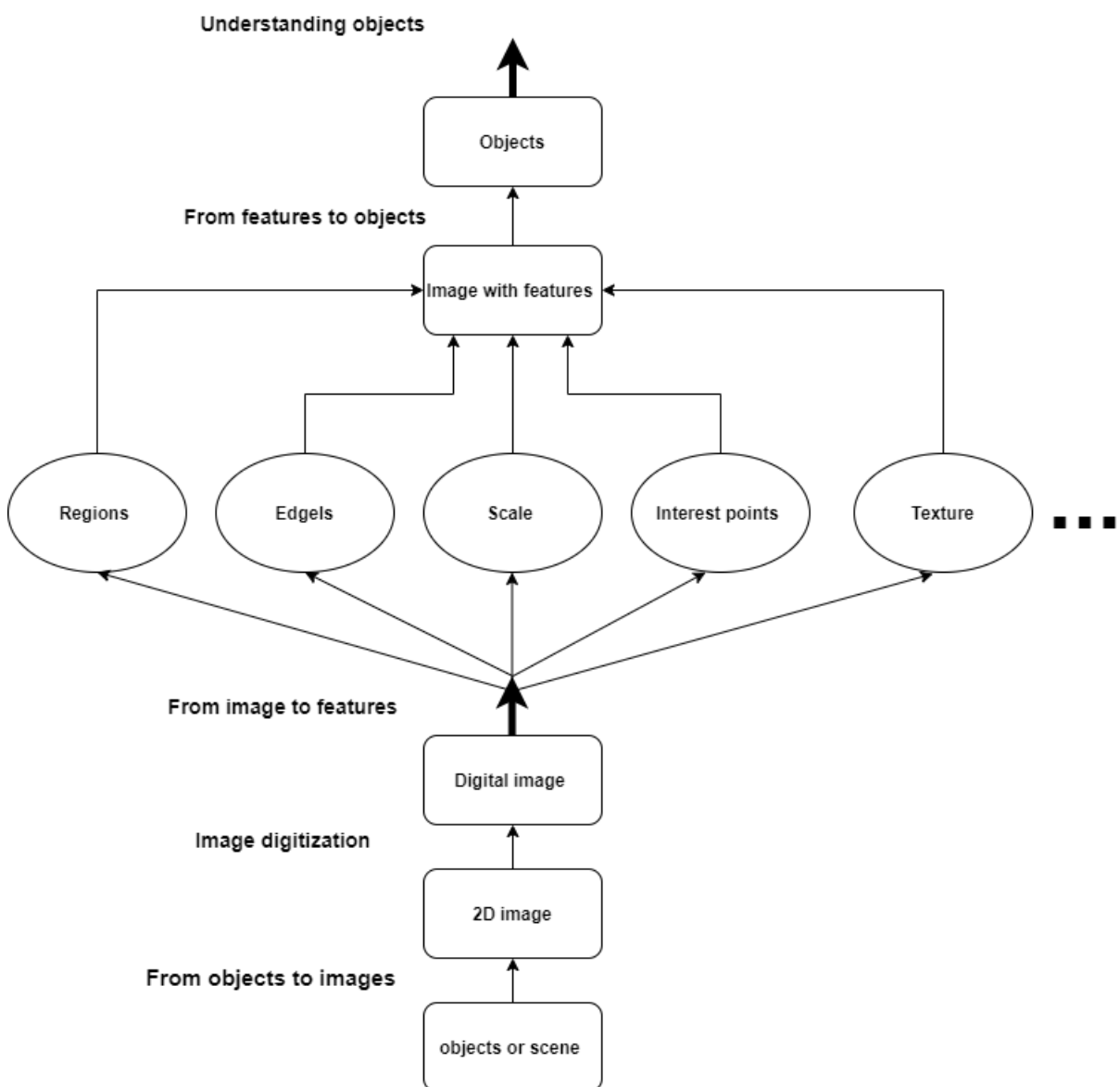


Figure II.2: four possible levels of image representation suitable for image analysis problems in which objects have to be detected and classified.

II.4.1 Digital image processing levels

three types of digital image processing are distinguished: low-, mid, and high-level process.

II.4.1.1 Low-level Processing

in the low-level processing the computer uses a little information about the content of image, this also can be achieved by the high-level image processing, this level often provides, pre-processing methods for noise filtering and blurring, contrast enhancement, and image sharpening. and image compressing.[5], and this level is characterized by the fact that the inputs are an image and the output is also an image.

II.4.1.2 Mid-level processing

Mid-level processing of images involves tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (e.g., edges, contours, and the identity of individual objects), for example, if there is one looking at a picture of a cup of coffee on a desk, the mid-level processing or description would be, on what region from the picture the cup is, where is the cup edges, what the colors on the cup .., and this description apply on the whole image, not just the cup.[3]

II.4.1.3 High-level processing

this level is based on knowledge, goals, and plans of how to achieve those goals, and artificial intelligence methods are widely applicable. High-level computer vision tries to imitate human understanding and the ability to make decisions according to the information contained in the image.[5]

Regardless of the image analysis task presented by levels, there are other forms of grouping, the commonly used are mentioned: by functions, by the level of processing, and by how pixels are mapped from the input to the output.[6]

II.5 Digital Images

II.5.1 Definition of Digital image

a digital image may be defined as a two-dimensional function, $f(x, y)$, where x and y are spatial coordinates, and the amplitude of at any pair of coordinates (x, y) is called the intensity or grey level of the image at that point. This a digital image,

note that a digital image is composed of a finite number of elements, each of which has a particular location and value, these elements are called picture elements, images elements, pixels, Pixel is the term used most widely to denote the elements of a digital image. [3]

II.5.2 Image Pixel

Digital images are represented on computers as arrays of digital numbers because computers cannot handle continuous images, therefore it requires to be represented as two-dimensional arrays of points, a point on the 2-D grid is called a pixel, these pixels are located on a rectangular grid, pixel position is given in the common notation for matrices with $M \times N$ rows and columns, the first index m denote the position of the row, the second, n the position of the column, the index n runs from 0 to $N-1$, and the index m from 0 to $M-1$. [7] (see Figure II.3)

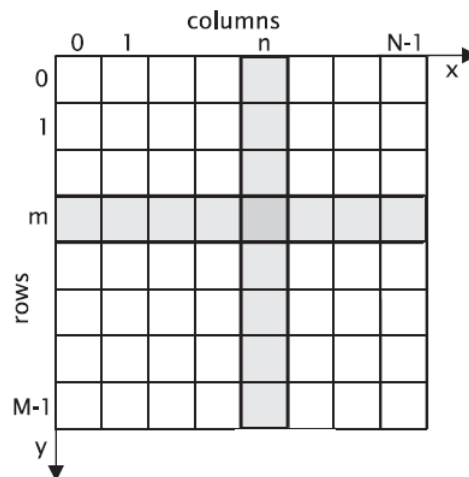


Figure II.3: Representation of digital images by arrays of discrete points on a rectangular grid

II.5.3 Preview of digital image

Everything can the vision of human see, might be represented as a form of digital image (a car, field mountain, sea, text, person's face Etc.), these representation or digital images contain row and column of pixels (picture element), pixels are like little tiles holding quantized value, these values depend on how many bits this image is coded, often digital images are coded with 8bits which in this case the values (intensity level) of pixels are between 0-255, that represent the brightness at the point of the image, 0 could be the darkest and 255 the brightest.[3] Figure II.4 shows a gray image coded with 255 gray level of a flower and an 8×8 sub image, and table II.1, shows the value of the intensity level at every pixel on the 8×8 sub-image.

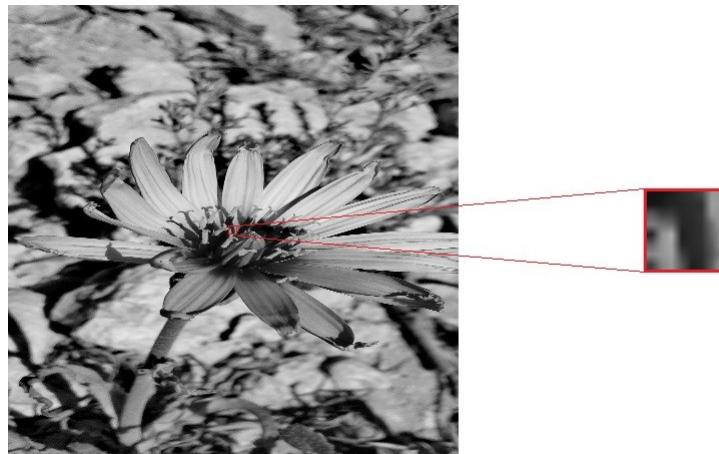


Figure II.4: (Left) Image of a flower, (right) Subimage of 8x8 pixels from the flower image

	0	1	2	3	4	5	6	7
0	177	179	175	175	175	188	189	186
1	183	189	183	182	182	189	188	184
2	186	187	187	184	177	182	183	182
3	183	189	189	183	180	178	183	177
4	184	191	191	183	178	171	177	182
5	187	190	188	181	181	168	168	173
6	187	187	185	172	173	170	171	171
7	182	184	182	179	160	164	164	166

Table II.1: Representation of the intensity level at every pixel in the 8x8 sub-image.

II.5.4 Image histogram

The brightness histogram $H(z)$ of an image provides then the frequency of the brightness value z in the image, the histogram of an image with L gray-levels is represented by a one-dimensional array with L elements. The histogram is usually the only global information about the image which is available. It is used when finding optimal illumination conditions for capturing an image, grey-scale transformation and image segmentation to objects and background. Note that one histogram may correspond to several images.[5] An image and its brightness histogram are shown in Figure II.5b

II.5.5 Binary image

The binary image is the simplest form of a digital image because it coded with 1 bit, and each pixel has 0 or 1 which represent respectively a black or white. Figure II.6 shows an example of a binary image.[6]

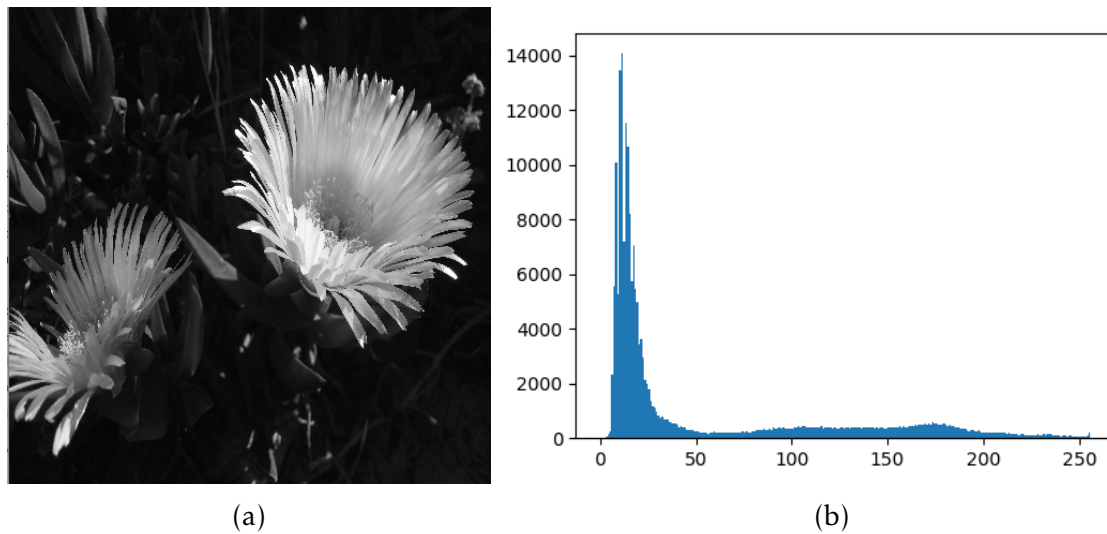


Figure II.5: (a) an grey-scale image. (b) brightness histogram of that image



Figure II.6: A binary Image

II.6 Color image processing

The use of colors in computer vision is the basis of modern digital image processing which make it much powerful technic used currently in several areas such as Medical use, Machines and Robots, security ... etc. Colors is a powerful descriptor that makes the object identification much simpler and the data extraction from scenes easier, we distinguish a thousand of color shades compared to just about two dozen shades of grey.

Color image processing is divided into two major areas: Pseudo and full-color processing. In the first category, the issue is one of assigning a color(s) to a particular grayscale intensity or range of intensities. In the second, images typically are acquired

using a full-color sensor, such as a digital camera, or color scanner. Until just a few years ago, most digital color image processing was done at the pseudo or reduced-color level. However, because color sensors and processing hardware have become available at reasonable prices.[3]

II.6.1 Color Fundamentals

In 1666, Sir Isaac Newton discovered that when a beam of sunlight passes through a glass prism, the emerging light is not white, but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other. As Figure II.7 shows, the color spectrum may be divided into six broad regions: violet, blue, green, yellow, orange, and red. When viewed in full-color no color in the spectrum ends abruptly; rather, each color blends smoothly into the next. (Figure II.8).[3]

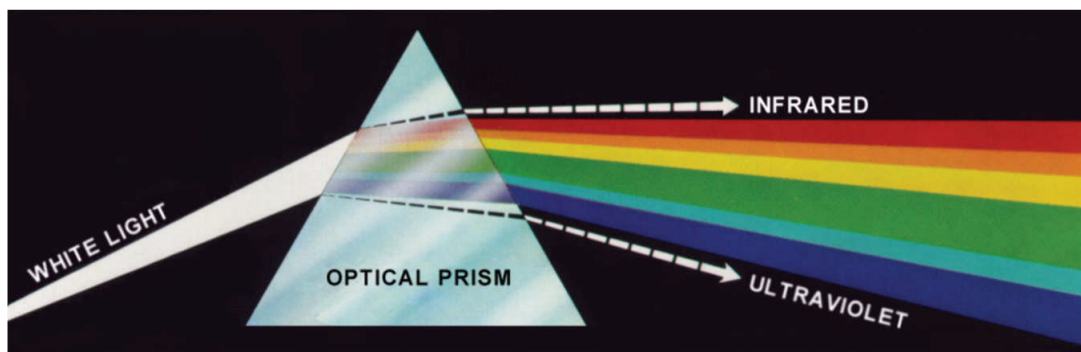


Figure II.7: Color spectrum has seen by passing white light through a prism

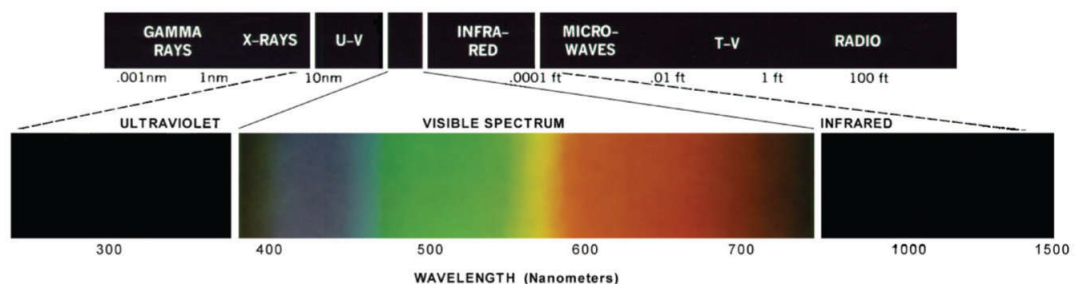


Figure II.8: Wavelengths comprising the visible range of the electromagnetic spectrum.

II.6.2 Color Models

The color models, also called color systems or color spaces, several of color spaces are used to facilitate the specifications of colors and can be transferred from color space to another, nowadays most of them are used in an oriented way either toward hardware such as printers or toward applications, I term of digital image processing the RGB (Red, Green, Blue), the CMY (Cyan, Magenta, Yellow), and the HSV (Hue, Saturation, Value) models, are the most commonly used.[5]

II.6.2.1 The RGB color spaces

The RGB (Red, Green, Blue) color space has its origin in color television where Cathode Ray Tubes (CRT) was used. RGB color space, The RGB model uses additive color mixing to inform what kind of light needs to be emitted to produce a given color. The value of a particular color is expressed as a vector of three elements intensities of three primary colors. the RGB model can be imagined as a cube shown in Figure II.9, in which RGB primary values are at three corners, and the other corners are the secondary color Cyan, Magenta, Yellow. And the two left corners are the black and white, the grayscale in this model, is the point where the RGB value are equal. The other different colors are printed on or inside the cube.

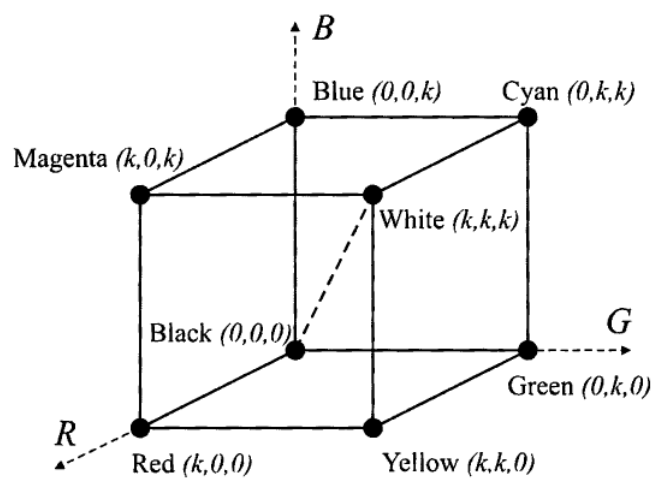


Figure II.9: Schematic of the RGB color cube. Points along the main diagonal have grey values, from black at the origin to white at point (1, 1, 1).

II.6.2.2 The HSV color space

HSV (Hue, Saturation, Value), color space (also known as HSB, hue, saturation, brightness). Is often used by printers it's close to their thinking and technique, HSV decouples intensity information from color, while hue and saturation correspond to the human perception, thus making this representation very useful for developing image processing algorithms, The HSV color model is illustrated in Figure II.10.

II.6.2.3 The CMY color space

the CMY (Cyan, Magenta, Yellow), the color model uses a subtractive color mixing which is used in printing processes. It describes what kind of inks need to be applied so the light reflected from the white substrate (paper) and passing through the inks produced a given color.

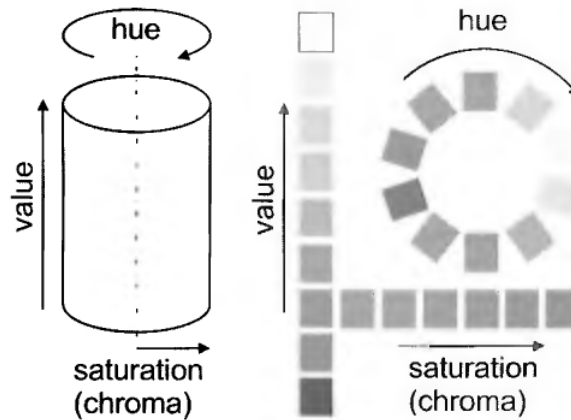


Figure II.10: the HSV color space illustrated as a cylinder and unfolded cylinder.

II.7 Application of digital image processing

The digital image processing is a widely used technique, in many areas for the ease of human daily life, and some of the major fields that used the image processing are mentioned below:[3]

- **Image sharpening and enhancement:** the goal of that process is to enhance those images that have been captured and make them better, or manipulate them to get the desired result, and there are many applications have been created to do this such as photoshop.
- **Medical field:** the following applications are commonly used in the medical field:
 - Gamma-Ray Imaging
 - X-Ray Imaging
 - UV imaging
- **Machine/Robot vision:** recently technology of robots and intelligent machines are growing so fast, especially the vision of robots which make the robot able to see and recognize things which are based on digital image processing.
- **Video processing:** process a video to recognize people or things for security purpose, this also involves noise reduction and motion detection etc.

II.8 Constraints of digital image processing

The computer vision is an extensive scientific, it's too much for some paper to cover its field and branches. And like many other sciences has its difficulties and constraints, we will mention some of them briefly. [5]

- Lose the information in 3D to 2D is a phenomenon which occurs in typical image capture devices such as the camera or the human eye.
- Interpretation of image(s) constitutes the principal tool of computer vision to approach problems which humans solve unwittingly. When a human tries to understand an image then previous knowledge and experience are brought to the current observation. The human ability to reason allows representation of long-gathered knowledge, and its use to solve new problems. Artificial intelligence has invested several decades in attempts to imitate the human perception to provide computers with the capability to understand observations and solve problems.
- Noise is inherently present in each measurement in the real world. Its existence calls for mathematical tools which can cope with uncertainty; an example is probability theory. Of course, more complex tools make the image analysis much more complicated compared to standard (deterministic) methods.
- too much data: Images and video sequences are huge. An A4 sheet of paper scanned monochromatically at 300 dots per inch at 8 bits per pixel corresponds to 8.5 MB. Non-interlaced RGB 24-bit color video 512 x 768 pixels, 25 frames per second, makes a data stream of 225 Mb per second. If the processing we devise is not very simple, then it is hard to achieve real-time performance; to process 25 or 30 images per second.

II.9 Conclusion

Digital image processing plays a vital role in a technological development and it's the core of technology nowadays, it has very wide applications and all of the technical fields are impacted by that and almost there is no technical field where digital image processing is not used such as: communications. Televisions, robotics. Medical, military, security etc. and in the nearest future we will see everything will be operated base on DIP applications.

Digital image processing can make machines acts autonomously, by provides machines the ability to see and recognize objects, perform tasks without any human intervention, autonomous machine based-vision are in already in use such as, Autonomous mobile robot, Self-driving car etc.

Hardware and Software description

III.1 Introduction

Every electronic system is a combination of components in which all its component assembled to work according to a set of rules which define the way of working, organizing or doing one or many tasks according to a fixed plan.

A system is a Hardware having software embedded on it, and perhaps additional parts either mechanic or electronic, the use of Hardware-software combination can easily achieve the purpose wanted with the best performance and less cost, however, it's possible to build a system that does not contain a software by replacing the Software-Hardware combination with a specific integrated circuit (ASIC), which can perform the same functionality. however, the software typical offer more flexibility and much easier and less cost.

This project is based on this Hardware-Software combination, the Hard part consists of the Raspberry Pi which is the brain of the project, Camera, Arduino board, DC motors and motor drives, and the software part consists of the operating system (OS), programming language and library used for image processing.

In this chapter, we will list and discuss each part and component in detail.

III.2 Hardware part

III.2.1 The Raspberry pi

the Raspberry Pi has found a role both as a very low-cost Linux based computer and as a platform for embedded computing. It has proven popular with educators and hobbyists alike, The world of Raspberry Pi is one that moves quickly. With a large active community, new interface boards and software libraries are being developed all the time. So, besides many examples that use specific interface boards or pieces of software. [8]

III.2.1.1 What is Raspberry Pi?

Raspberry Pi is a versatile, small and cheap single-board computer. Its hardware features and the capability of using open source software have allowed different implementations in different areas.[9]

III.2.1.2 Raspberry Pi components

Raspberry Pi (RPi) is a small, single-board computer which consists of several components, The Raspberry Pi board and its components are shown in figure III.1, each number is referring to a component. The board has small holes in the board for the case to be adjusted using screws, and it uses a 5V Micro USB power adapter (6), It has some LEDs (7) that indicate the board situation. It can connect an external projector or monitor via the HDMI port (5). It is also possible to connect a TV via the RCA output (3), The RPi3 can be connected to a screen which improves portability solution, The screen could be connected using the Display Connector (9), the board contain a general-purpose port for input/output GPIO (12), This port has also 40 pins, as in the previous version, and it allows connecting specific sensors (sound, humidity, temperature, etc.). if the application to be developed would require a camera, this could be a USB camera (1) or an integrated camera connected directly to the camera connector (4). RPi 3 has four USB ports (1). And it has also an Ethernet port to connect a network cable (2), but the new model includes integrated Wi-Fi and Bluetooth adaptors (10), regardless to the previous version that depends on an external USB Wi-Fi capabilities. The board has a Micro SD card (8) for installing the operating system and the other software required. And the component (11) is the microprocessor, the RPi 3 has an ARM Cortex-A53 64 bits quad-core processor.[10]

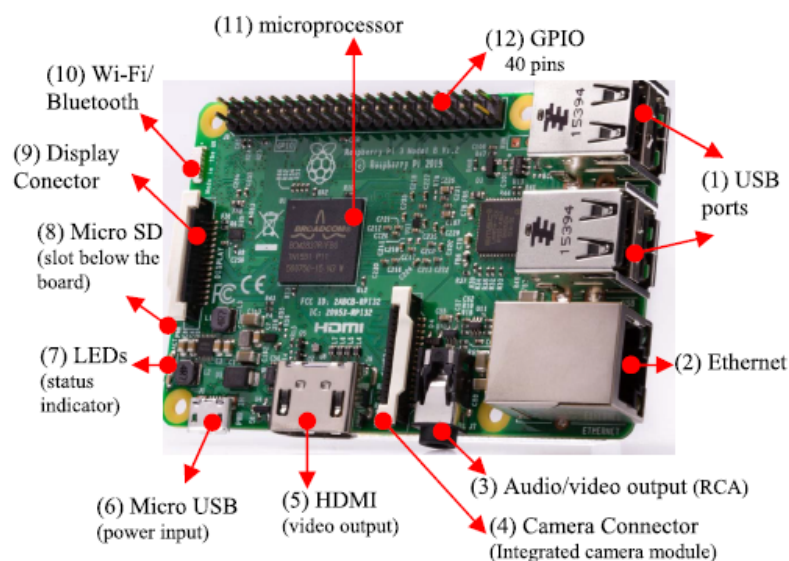


Figure III.1: Raspberry PI components

III.2.1.3 Raspberry Pi released models

with the fast evolution of technologies nowadays, several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU), the difference between all models consist of the processing speed, the memory RAM and ROM, in the size and weight, and how much I/O support is available and GPIO pins.

- **Model A/B:** The original Raspberry Pi models launched were known as the Model A and the Model B (see Figure III.2). Both had the same Broadcom BCM2835 SoC at their heart, but the difference was in some specifications: Model A had a 256 MB amount of RAM, and a single USB port, with no networking capabilities. In the other hand, the Model B had either 256 MB or 512 MB of RAM and two USB ports and a 10/100 wired network port.[11]

These Models has a 26 GPIO pin compared to modern Pi's which had a40 GPIO pins, and both models also use a full-size SD card for storage, rather than the compact micro-SD cards of newer models.

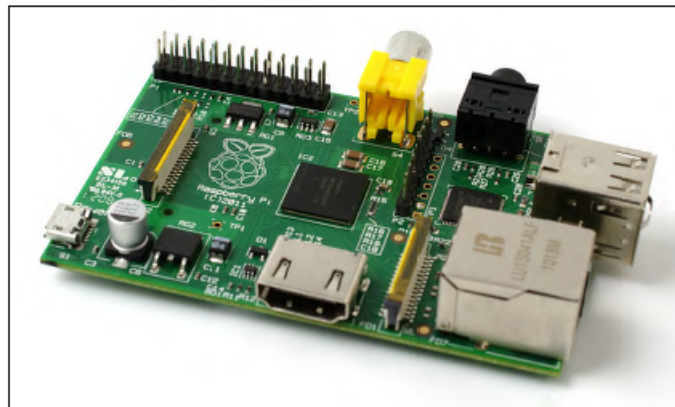


Figure III.2: The Raspberry Pi Model B board

- **Model A+/B+:** Despite that, the original model A/B was very popular, but they replace with the new models released known as the plus, the new board Model A+ and Model B+, comes with 40-pin GPIO and other improved features, Figure III.3 shows the Raspberry Pi B+ board. [11]

The difference between the Model A+ and Model B+ is similar to that of Model B and Model A, Model A+ has either 256 MB or 512 MB of memory depending when it was purchased, and a single USB port, and no network capabilities, the Model B+ he Model B+ has 512 MB of memory, four USB ports, and a 10/100 wired network port.

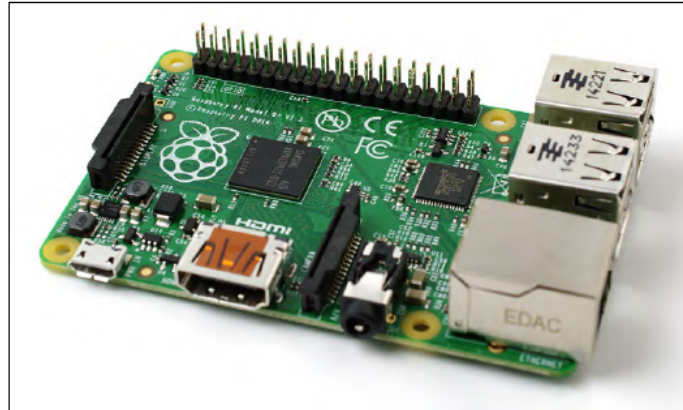


Figure III.3: The Raspberry Pi Model B+ board

- **Raspberry Pi 2:** While the plus and prior boards all used the same BCM2835 SoC, the Raspberry Pi 2 (Figure III.3 show a raspberry Pi 2 Model B+), had the first new processor: the BCM2836 SoC Featuring four processor cores to the original's lone core, the board also comes with 1 GB (1024 MB) of RAM, double the maximum previously available, making multitasking and memory-intensive applications smoother and more responsive. [11]
- **Raspberry Pi 3:** The raspberry Pi 3 (see Figure III.4) comes with a new processor: the Broadcom BCM2837, the first to feature 64-bit, rather than 32-bit, the BCM2837 is significantly faster than the BCM2836 found in the Pi. It has the same 40-pin GPIO header, and four USB ports, a 10/100 wired network port. The Pi 3 is also the first model to get built-in wireless support, featuring a radio capable of connecting to 2.4 GHz Wi-Fi networks and Bluetooth devices. [11]

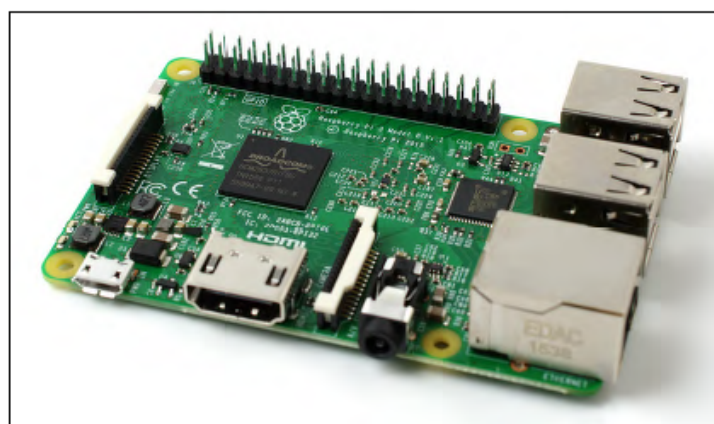


Figure III.4: The Raspberry Pi 3 board

- **Raspberry Pi Zero:** The raspberry Pi Zero (Figure III.5) is the smallest Raspberry Pi and the cheapest, the Pi Zero include the same BCM2835 Soc and 512 MB of RAM, it has a single micro-USB port and mini-HDMI port both require adapters

before they can be connected to standard peripheral. there's no DSI port, and the CSI port requires an adapter, and the GPIO header, requires pins to be purchased and soldered into place before it can be used. [11]

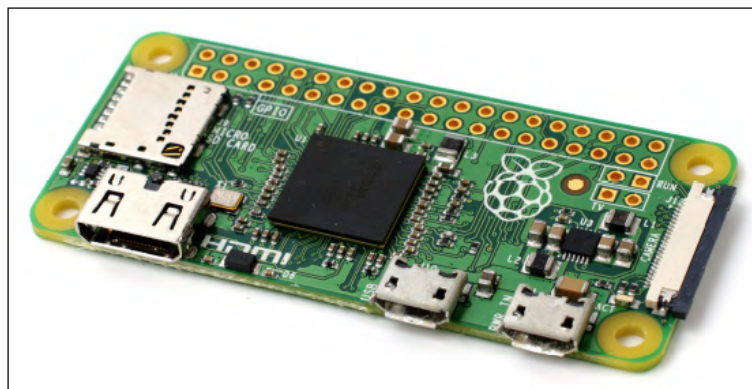


Figure III.5: The Raspberry Pi Zero board

III.2.1.4 GPIO on the Raspberry Pi

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi boards. Prior to the Pi 1 Model B+ (2014), boards comprised a shorter 26-pin header. Each pin of the GPIO port has its own purpose, with several pins working together to form particular interfaces. Figure III.6 shows the layout of the GPIO port. Depending on the model of Raspberry Pi board, the GPIO port provides at least eight pins for general-purpose use by default: Pin 7, Pin 11, Pin 12, Pin 13, Pin 15, Pin 16, Pin 18, and Pin 22. These pins can be toggled between three states: *high*, where they are providing a positive voltage of 3.3 V; *low*, where they are equal to the ground or 0 V; and input. The two outputs equate to the 1 and 0 of binary logic and can be used to turn other components on or off.[11]

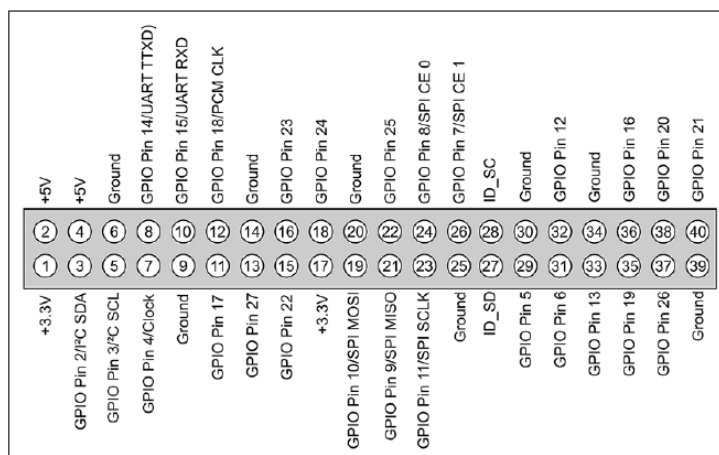


Figure III.6: The Raspberry Pi's GPIO port and its pin definitions

III.2.1.5 Review Raspberry Pi 3 model B+ used in our project

From a variant model Raspberry pi 3 B+ which used in this project because it has the specification that satisfies the project requirements. (Figure III.7)

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad-core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT. The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.[12]

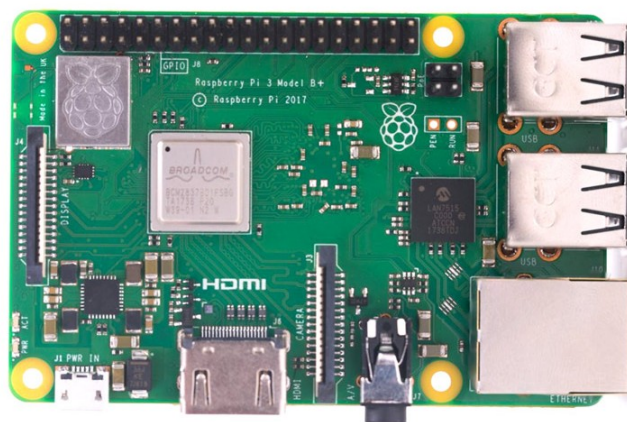


Figure III.7: Raspberry Pi 3 model B+

Technical Specification of the Raspberry Pi 3 Model B+ are illustrated in table III.1, and physical Specification shown in Figure III.8

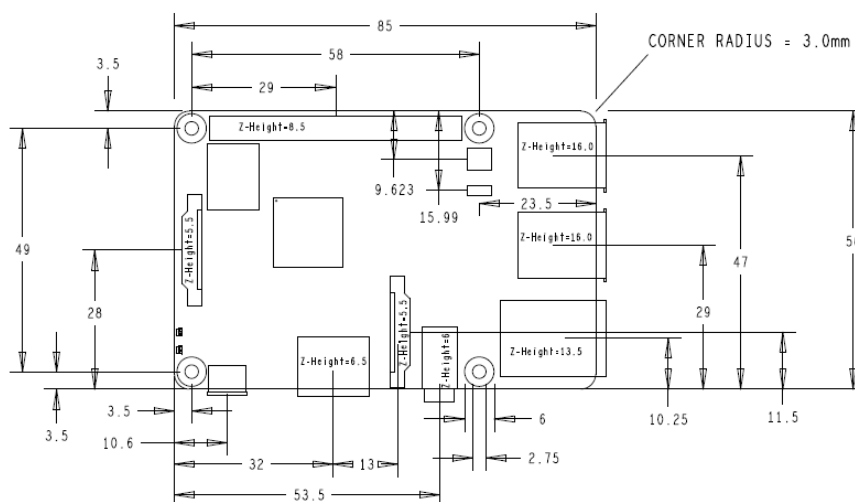


Figure III.8: Physical specifications of Raspberry Pi 3 Model B+

Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Connectivity	<ul style="list-style-type: none"> • 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE • Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) • 4 × USB 2.0 ports
Access	Extended 40-pin GPIO header
Video and sound	<ul style="list-style-type: none"> • 1 × full size HDMI • MIPI DSI display port • MIPI CSI camera port • 4 pole stereo output and composite video port
Multimedia	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card	support Micro SD format for loading operating system and data storage
Input power	<ul style="list-style-type: none"> • 5V/2.5A DC via micro USB connector • 5V DC via GPIO header • Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Operating temperature, 0 – 50°

Table III.1: the specifications of the Raspberry Pi 3 Model B+

III.2.2 Raspberry Pi Camera

High definition camera module compatible with the Raspberry Pi model A and model B. Provides high sensitivity, low noise image capture in an ultra-small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the processor. [13]

III.2.2.1 Raspberry Pi Camera rev 1.3

The Raspberry Pi Camera rev 1.3 shown in Figure III.9 used in this project, The table III.2 show the specification of the Raspberry Pi Camera rev 1.3.

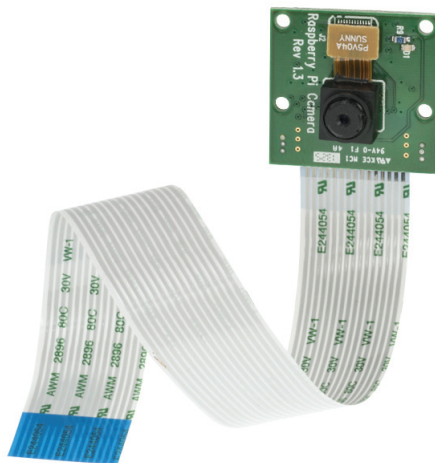


Figure III.9: Raspberry Pi Camera rev 1.3

Resolution	5-megapixel
Picture resolution	2592 x 1944
Max image transfer rate	<ul style="list-style-type: none"> • 1080p: 30fps (encode and decode) • 720p: 60fps
Temp range	<ul style="list-style-type: none"> • Operating: -30° to 70° • Stable image: 0° to 50°
Dimensions	20 x 25 x 10mm
Weight	3g

Table III.2: Specifications of the Raspberry Pi camera rev 1.3

III.2.3 Arduino board

With the rapid evolution of technologies which makes human life easier and to solve problems, the embedded system field has taken its share of that evolution, Microcontroller system is one of the embedded system branches that is used widely in daily life, These days many people try to use the Arduino because it makes things easier due to the simplified version of C++. [15] In simple terms, the Arduino is a

tiny computer system that can be programmed with your instructions to interact with various forms of input and output.

III.2.3.1 what is the arduino?

the Arduino board it's open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. [14] the Arduino board is consisting of two main parts:

Hardware: The Arduino is not a microcontroller but instead, a microcontroller interface board. It has a microcontroller chip on the board, but it also has a whole load of other components (see Figure III.10) that provide: [15]

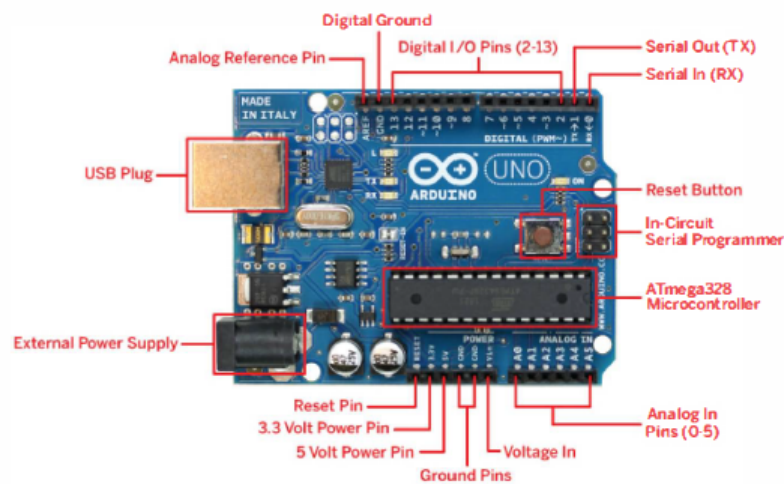


Figure III.10: labelled Arduino Uno board

- USB plug: this an important part of the Arduino which is used to upload the programme to the microcontroller, and used either as power supply because it has a 5-volt regulator to power the board.
- External Power Supply: it has a 9 to 12-volt regulator, used to power the board.
- Reset button: this button resets the Arduino board when it pressed.
- Microcontroller: This is the device that receives and send information or command to the respective circuit.
- Analog Pins(O-5): This are analog input pins from AO to A5.
- Digital I/O Pins: These are the digital input/output Pins 2 to l3
- In-Circuit Programmer: This is another source to upload the programme.

- Digital and analog Ground pins.
- Power Pins: it has 3.3-volt and 5-volt output power pins

Software: An Arduino does not do anything until it has been programmed. That is, you have to use your computer to write some instructions in the programming language C that can control and read the GPIO pins. Special software, called the Arduino IDE, is provided to both write the programs and then upload them onto the Arduino using a USB cable.[15] Figure III.11 shows Arduino IDLE.

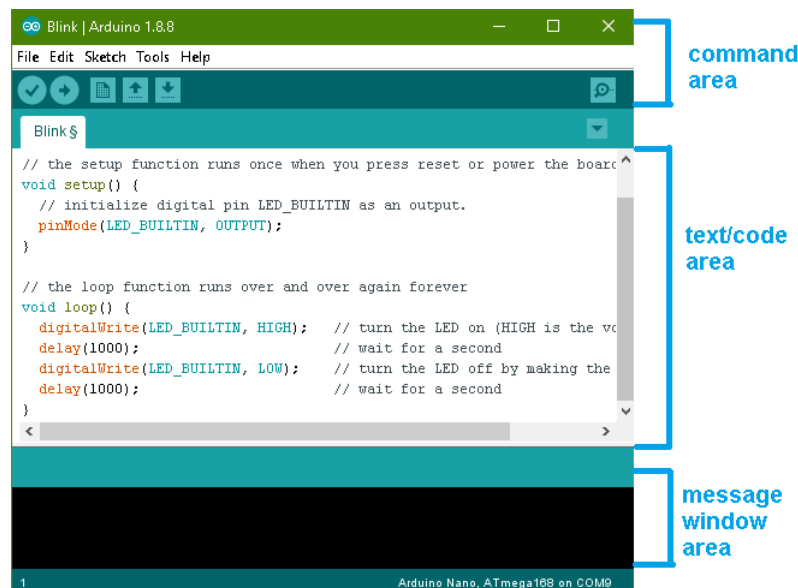


Figure III.11: Labelled Arduino IDE

- **Command Area:** this section contains: Edit, Sketch, Tools, Help and Verify button for verification, Upload button for uploading your code New, Open, Save and Serial Monitor used for sending and receiving of data between the Arduino and the IDE.
- **Text Area:** this is the area where you write your code (programme), with the simplified version of C++ programming language. The structure of the Arduino code is consisting of two parts:
 - The setup function: the area just before this function is used to initialize the variable that you intended to use in your code, then the setup function begins in which you set the initial condition of your variables, and it runs once only when you power the board.
 - Loop function: this the main function that you write your command in, and it keeps running over and over again.

- **Message Window Area:** This shows a message from the IDE in the black area, mostly on verification on your code.

III.2.3.2 Arduino Nano

the purpose of using Arduino Nano board in this project is to control all the motors which execute the orders given by the raspberry pi because Microcontrollers are typically better for low-level tasks that require accurate/fast timing.

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328. (Figure III.12) It has more or less the same functionality of the Arduino Duemilanove but in a different package. It lacks only a DC power jack and works with a Mini-B USB cable instead of a standard one.[16]

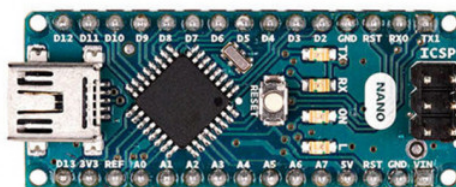


Figure III.12: Arduino Nano Board

Specifications of the arduino Nano board shown in table III.3

Microcontroller	ATmega320
Operating Voltage	5V
Flash Memory	32 KB which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHZ
Analog IN Pins	8
EEPROM	1KB
DC Current per i/o Pins	40 mA (I/O Pins)
Input Voltage	7-12V
Digital I/O Pins	22 (6 of which are PWM)
Power Consumption	19 mA

Table III.3: Arduino Nano specifications

III.2.4 Dc motors

The most of mechanical development exist today are accomplished by electrical motors, DC or AC motors, the electrical motors take the electrical energy and convert it to mechanical energy. these motors are widely used in our lives.

A DC motor consists of a stator, an armature, a rotor and a commutator with brushes. The opposite polarity between the two magnetic fields inside the motor cause it to turn. DC motors are the simplest type of motor and are used in household appliances, such as electric windows in cars. Over a very wide power range from several megawatts at the top end down to only a few watts, all DC machines have the same basic structure, [17] see figure III.13

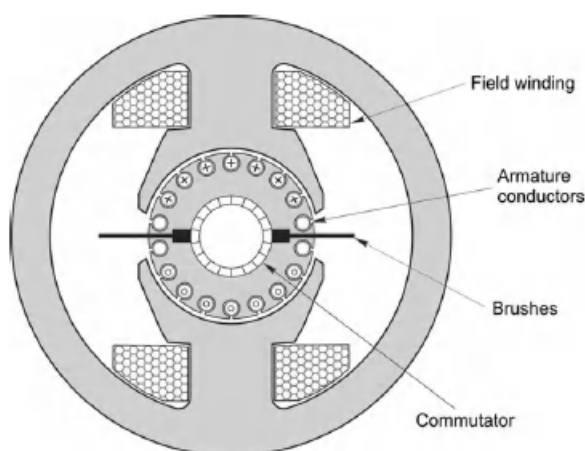


Figure III.13: Conventional (brushed) d.c. motor scheme

the DC Motor is consisting of two separate electrical circuit, the smaller pair of terminals connect to the field windings, which surround each pole. these windings provide the m.m.f. to set up the flux in the air-gap under the poles. The main terminals (usually designated A1, A2) convey current to the brushes which make sliding contact to the armature winding on the rotor. [17]

III.2.4.1 Equivalent circuit of DC motor

The equivalent circuit, using the basic components of electronics is shown in figure III.14

- **V:** is the voltage applied to the armature terminals or across the brushes
- **E:** is the internally developed motional e.m.f .
- **R and L:** are the resistance and the inductance of the complete armature.

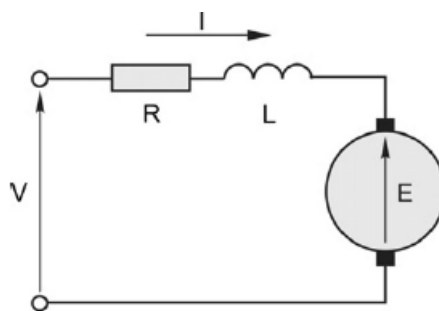


Figure III.14: Equivalent circuit of a DC Motor.

III.2.4.2 DC Gearbox Motor "TT Motor"

DC motors are widely used to run the wheels of the autonomous mobile robot, in this project, a DC Gearbox Motor used to control the robot motion. Figure III.15, Technical specifications of the DC Gearbox Motor - 200RPM are illustrated in table III.4. [18]

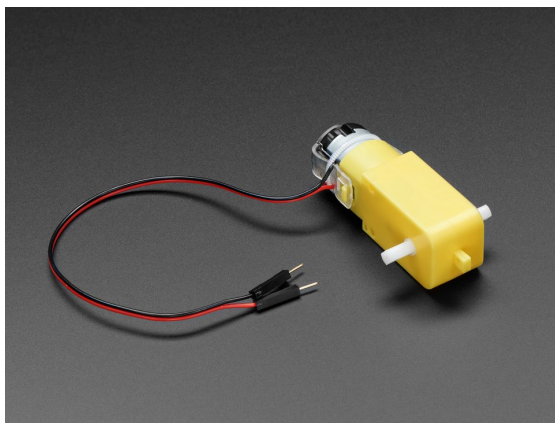


Figure III.15: DC Gearbox Motor "TT Motor".

Rated Voltage	3 to 6V
Continuous No-Load Current	150mA ± 10%
Min Operating Speed (3V)	90 ± 10% RPM
MAX Operating Speed (6V)	200 ± 10% RPM
Continuous No-Load Current	150mA ± 10%
Gear Ratio	1:48
Body Dimensions	70 x 22 x 18mm
Input Voltage	7-12V
Weight	30.6g

Table III.4: technical specification of the DC Gearbox Motor - "TT Motor" - 200RPM - 3 to 6VDC

III.2.5 Motor Drives

The Motor drives are circuits used to run a motor. It's commonly used for motor interfacing. These drives circuits can be easily interfaced with the motor and their selection depends on current, and voltage that the functionality of motor required. Figure III.16 illustrate how the motor drive controls motors.

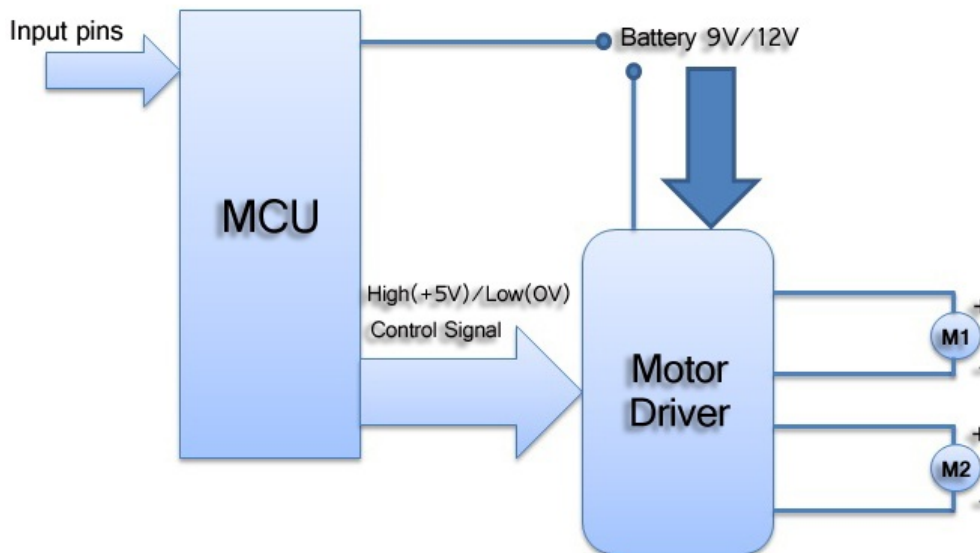


Figure III.16: Motor drive bloc diagram

III.2.5.1 DC Motor Driver

These motor drives are considered as a current amplifier that used as a bridge between the controller and the motor itself, they are made of an integrated circuit, the input to the motor driver IC or motor driver circuit is a low current signal. The function of the circuit is to convert the low current signal to a high current signal. Which run the motor.

III.2.5.2 The MX1508 Motor Drive

The MX1508 board (Figure III.17a) can drive two DC motors independently. It has Dual H-bridge motor driver can drive two DC motors in either direction or a 4-wire two-phase stepper motor. This board control The motor speed with a Pulse Width Modulated (PWM) signal to the IN connections. The schematic of the Mx1508 IC shown in figure III.17b

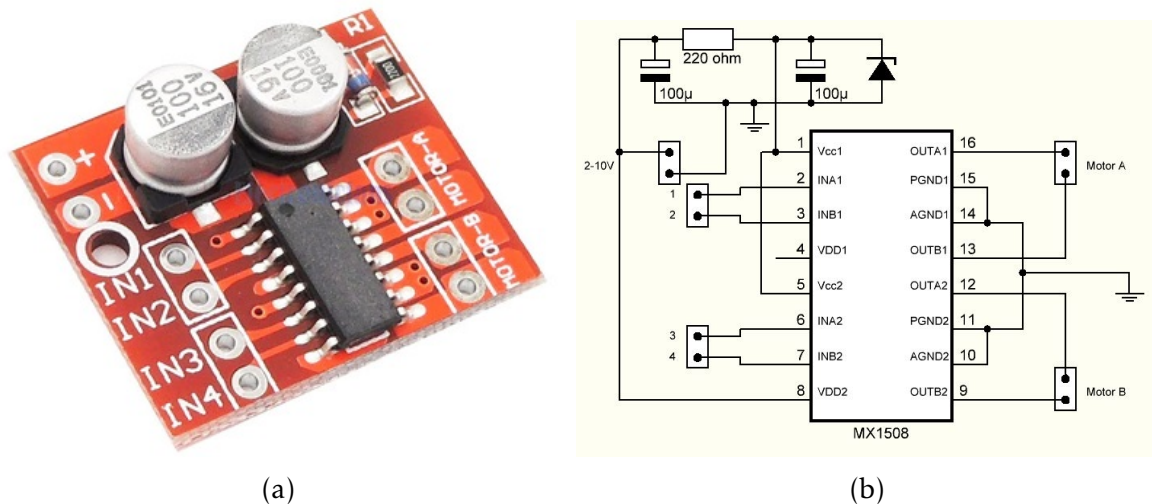


Figure III.17: The MX1508 Motor Drive and the MX1508 IC circuit

III.2.6 distance sensors

Distance measurement of a body or obstacle in front or by the side of a moving object is required in a large number of devices. The distance sensor is one of these devices and is used greatly in automotive vehicles, autonomous robot navigation, blind man's walking stick, medical applications, these devices may be small or large, and also reasonably simple or complex. However, such distance-measuring devices are available. These use several kinds of sensors and systems. In the automotive industry, the latest models, namely, 'smart cars' need sophisticated sensors that can measure distances within the range. During parking or even in moving situation, as well as intelligent suspensions and headlight levelling, there is need for some devices by which distance measurement can be performed. Application of Distance Sensor in AGV Robot Position Control Sensor-based fuzzy control method can be used for the navigation of an autonomous guided vehicle (AGV) robot. Robot navigation is defined as the guiding of a mobile robot to the desired destination or along a desired path following the robot position in an environment characterized by a set of distinct objects, such as obstacles and landmarks. [19]

Figure III.18 illustrates the basic of mobile robot position control, the distance sensor is mounted in front of the robot, the robot can avoid obstacles due to its ability to sense and detect obstacles via the distance sensor that send data to the microcontroller which react immediately by changing the direction or stop the robot.

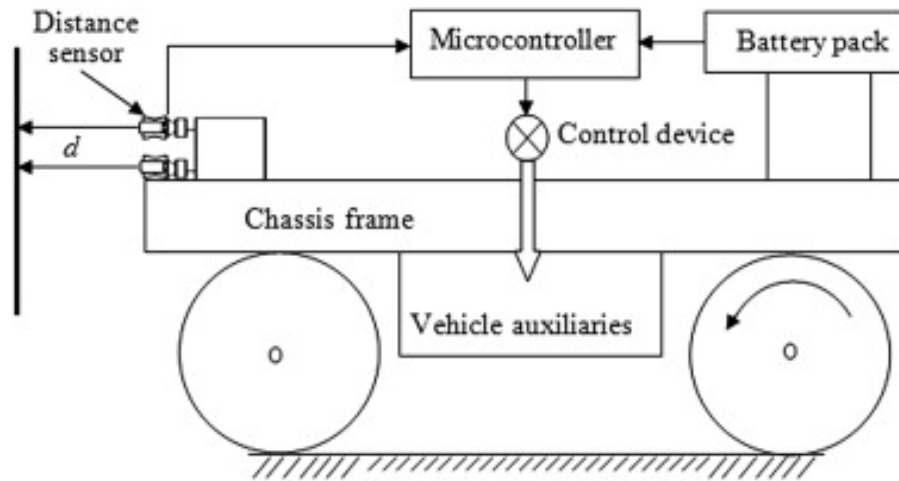


Figure III.18: Distance sensor's position control of a mobile robot

III.2.6.1 Sharp IR GP2Y0A41SK0F distance sensor

In order to measure the distance between the robot and the followed object, and to avoid barriers and control the robot's position, we've used the Sharp IR distance sensor.

The sharp IR is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IR-LED (infrared emitting diode) and signal processing circuit? The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor. [20] ,(see Figure III.19).



Figure III.19: Sharp IR GP2Y0A41SK0F

Features of Sharp IR GP2Y0A41SK0F :

- Distance measuring sensor is united with PSD, infrared LED and signal processing circuit.
- Short measuring cycle (16.5ms).
- Distance measuring range: 4 to 30 cm.
- Package size (29.5 × 13.0 × 13.5mm).
- Analog output type.

III.2.7 Camera Module Pan Tilt Mount

This Pan-tilt mount is a suitable way to this project to customize a full range motion, it includes two servo motors to rotate roughly 250° horizontally and vertically roughly 150°. The module is shown in Figure III.20

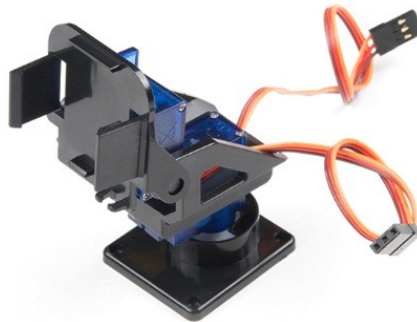


Figure III.20: Pi Camera Mount Module Pan Tilt

III.2.8 The 4W robotic vehicle chassis

III.2.8.1 description

The 4-Wheel robotic vehicle chassis, it's a perfect platform for the mobile robotic projects, the kit includes then mechanical hardware required to build a mobile robot project, that include the motors, wheels, chassis, etc. it offers a large space with holes for mounting sensors and electronics as a project requirement.

III.2.8.2 Features

- 4 Gearbox DC motors operate at 3-6V DC
- Length: 25 cm
- Width: 15 cm
- Tire Diameter: 6.6

III.2.8.3 kit assembly

The kit comes unassembled it requires assembly by the user, and it's a very easy to easy to assemble. See Figure III.21



Figure III.21: (a)unassembled 4w Robotic Vehicle Chassis, (b) assembled 4w robotic vehicle chassis

III.3 Software part

III.3.1 Raspberry Pi Operating System

We can tell that the Raspberry Pi, is a small computer capable to do the work that every computer can do, of course, there are a couple of differences between the raspberry board and a normal desktop computer or laptop, such as in size and the price and the important thing is the operating system which is the software that enables to control the board.

As known that the majority of desktop computer today are running of two operation systems: Microsoft Windows or Apple OS. And both of them are a closed source.

Which mean that the user can obtain the finished software but never to see how it's made. In another hand the Raspberry Pi is mainly designed to run an operating system called GNU/Linux, Linux unlike other OS is open-source, which mean that you are capable to download the source for the whole operating system and make your changes.[11]

Linux OS has several versions known as distributions, have been ported to the Raspberry Pi, including Raspbian and Arch Linux .. etc.

III.3.1.1 Raspbian Linux OS

III.3.1.1.1 Debian, the parent of Raspbian: Raspbian has been created based on other Linux distribution called Debian, Debian is a popular distribution in its right, Raspbian is a fork of Debian adapted for a particular task. not the only one Ubuntu Linux either is based on Debian.

III.3.1.1.2 Introducing Raspbian: Raspbian is the name given to a customized variant of the popular Debian Linux distribution. Debian is one of the longest-running Linux distributions and concentrates on stability, high compatibility, and excellent performance even on modest hardware—making it a great partner for the Raspberry Pi. Raspbian takes Debian as its base, or parent distribution, and adds custom tools and software to make using the Raspberry Pi as easy as possible.[11]

the size of Raspbian is minimum as possible, the raspberry Pi image for Raspbian contains only the software that you'd find on a regular desktop version, that includes the necessary tools such as web browsers, file management and browsing, programming with python. using the GUI with the Pi gives quick access to the tools and quickly install any additional software you might need. Raspbian comes with plenty of pre-installed software you can get started with, the software provided with the Raspbian distribution is split into themed categories. To view these categories, left-click the Menu button found on the top left of the screen as part of the taskbar (see Figure III.22).

The following list to some preinstalled software comes with the Raspbian:

- **Geany Programmer's Editor:** A cross between a simple text editor and a fully-featured IDE, Geany is a popular choice for writing programs in a variety of languages.
- **Python 2 (IDLE):** An IDE written specifically for Python.
- **Python 3 (IDLE):** An IDLE to use the newer Python 3 programming language, rather than the default Python 2.7 language.

- **File Manager:** The file manager provides a graphical browser for files stored on the Pi or any connected storage device.
- **Image Viewer:** Allows you to view images, such as those from a digital camera or on a connected storage device.
- **PDF Viewer:** Opens Portable Document Format (PDF) files.
- **Terminal:** This terminal package allows you to use the Linux command line.
- **Text Editor:** This is a simple text editor, which is useful for making quick notes or writing simple programs.

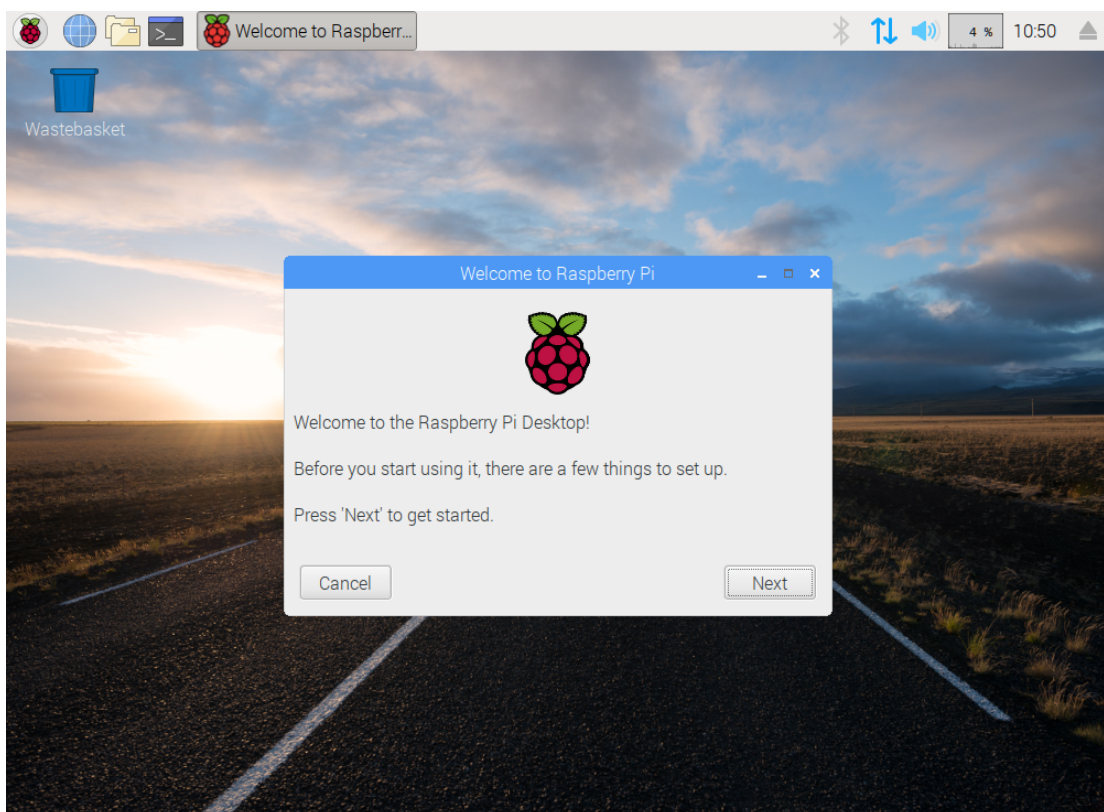


Figure III.22: Raspbian Desktop

III.3.2 Python Programming language

The main part in this project (an object color recognition) is the image processing technic which don via the python programming language using a super-powerful library called OpenCV, so what is python and why Python?

Python is a general-purpose, object-oriented, and open-source computer programming language. It is commonly used for both standalone programs, and scripting applications in a wide variety of domains, by hundreds of thousands of developers. Python

is designed to optimize developer productivity, Software quality, program portability, and components integration, Python programs run on most platforms in common use, Linux, Windows and Macintosh ... etc. [21]

Python is published under an open-source license and is freely available for Linux, OS X, and Windows computer systems. This cross-platform support means that software written using Python on the Pi can be used on computers running almost any other operating system as well—except where the program makes use of Pi-specific hardware such as the GPIO. [11]

III.3.2.1 Python and Artificial Intelligent

Artificial intelligent is considered to be the trending technology of the future. Already there are several applications made based on it. Due to this, many companies and researchers are taking an interest in it. But the main question that arises here is that in which programming language can these AI applications be developed, there are various programming languages used for AI, among them, Python gains huge popularity and the reason are as follows:[22]

- Open source
- Simple syntax and less coding
- inbuilt libraries for AI projects
- Easy to learn

III.3.3 An introduction to OpenCV

Initially developed by Intel, OpenCV (Open Source Computer Vision) is a free cross-platform library for real-time image processing that has become a standard tool for all things related to Computer Vision. The first version was released in 2000 under BSD license and since then, its functionality has been very much enriched by the scientific community. In 2012, the nonprofit foundation OpenCV.org took on the task of maintaining a support site for developers and users. [23]

OpenCV is available for the most popular operating systems, such as GNU/ Linux, OS X, Windows, Android, iOS, and some more. The first implementation was in the C programming language; however, its popularity grew with its C++ implementation as of Version 2.0. New functions are programmed with C++. However, nowadays, the library has a full interface for other programming languages, such as Java, Python, and MATLAB/Octave.

III.4 Conclusion

We have now given a brief explanation about all necessary part and the working principle of each component which this project consists of (hardware and Software). the hardware section that refers to the physical elements such as electronic components (Resistors, Motor drives, Ics etc.), and the software section which knows as a program or applications that contain all the instruction and orders which tells the hardware how to perform tasks. In the next chapter, we are going to set up the Raspberry Pi board and assemble all the components of the project.

Raspberry Pi setup, and components assembly

IV.1 Introduction

This project is about the autonomous robot, which can consist of several components such as (the Raspberry Pi, the Arduino board, wheels, Pi camera... etc). Each component has a very important role in assuring the proper functioning of the project. Whose all the components need to be assembled and connected to provide the required functionality, some of these components are made only to execute commands given by other components while the other would communicate among each other. A brief discussion will be given in this chapter, in which you will be able to understand how those components are connected? and how do they work?

IV.2 Raspberry Pi Setup

IV.2.1 Installing the Raspbian OS on the Raspberry Pi

IV.2.1.1 Download Raspbian

There are two methods of installing the operating system (OS) on the Raspberry Pi.[24]

The first one is to download the Raspbian image and writing the bootable image on the SD card, the advantage is that it takes less space on the SD card. The second method is the easier way, NOOBS (New Out of the Box Software) is designed to make the installation on the Raspberry Pi easier, it allows to select from several operating systems and simple install. And this method takes more space in on the SD card.

In this project the first methods used because it takes less storage space. to install the Raspbian OS, initially, download the Raspbian Image from the download page: <https://www.raspberrypi.org/downloads/> (see Figure IV.1).

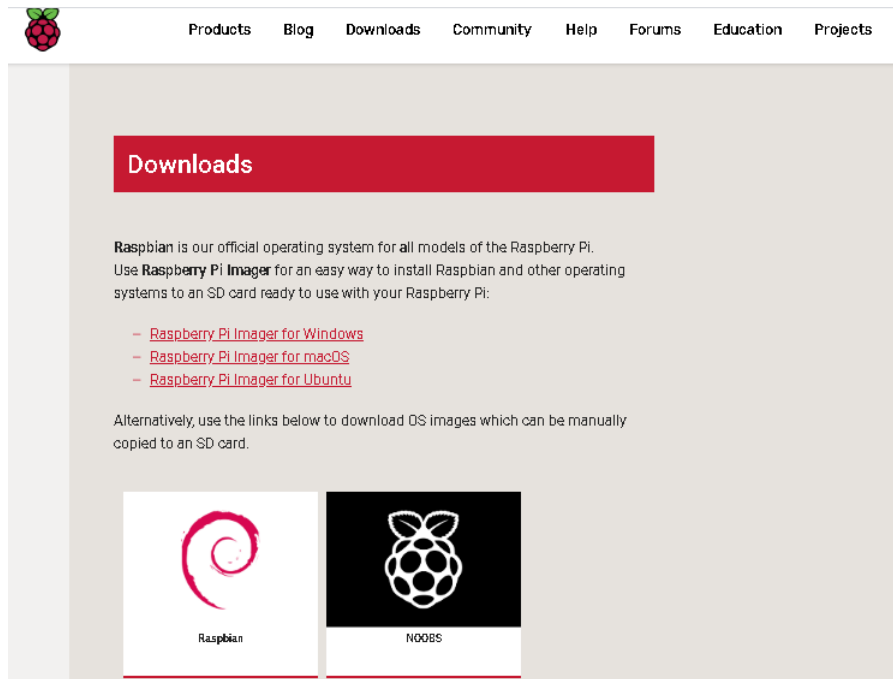


Figure IV.1: Raspbian download page

There are two installation methods shown on the download page, Navigate to the download page and choose the Raspbian distribution then the Raspbian version, in this projects we have chosen the Raspbian Buster With desktop with the size of 1.09 GB.

IV.2.1.2 Writing Raspbian on SD card

The installation process needs two additional application the SDFormatter and the Win32 Disk Imager: [24]

- **SDFormatter:** this application used to format and prepare the SD card. It can be downloaded at:

`www.sdcard.org/downloads/formatter_4/`

- **Win32 Disk Imager:** This application performs the writing of the image file to the SD card. It can be downloaded at:

`https://sourceforge.net/projects/win32diskimager/`

After downloading and installing the SDFormatter, initially, the SD card needs to prepare for the installation, the main window of SDFormatter shown in Figure (IV.2), before formatting the SD card make sure that the drive representing the SD card, to format click start after the process finishes the SD card is ready.

The Win32 disk imager used to write the image file to the SD card, open Win32 disk imager (see figure IV.3), and in the image field, select the Raspbian image make sure that the correct SD card is selected, then click write.

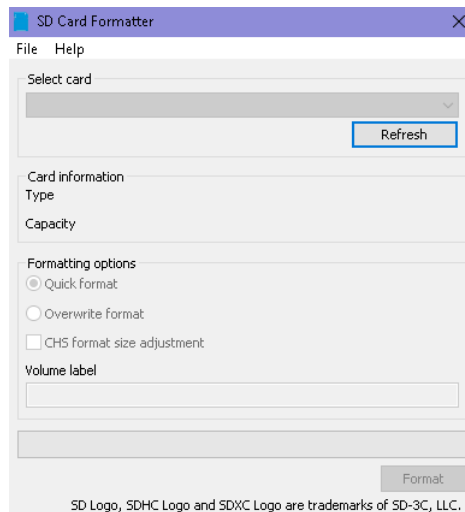


Figure IV.2: SDFormatter

When the writing process is done, inset the SD card on the Raspberry Pi.

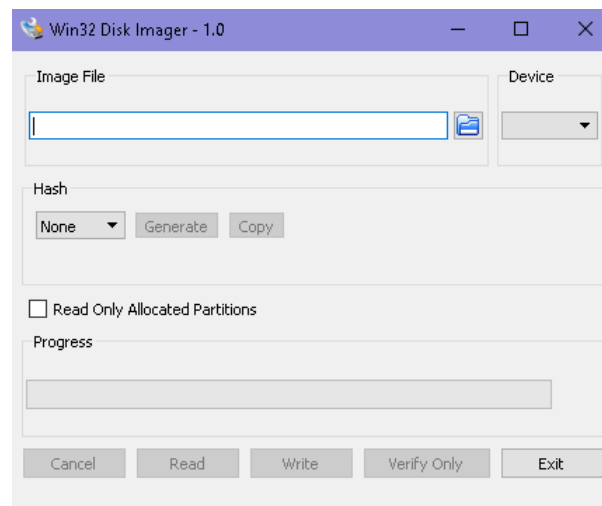


Figure IV.3: Win32 Disk Imager

IV.2.1.3 Connection Raspberry Pi

now the SD card ready for useful exploitation, we need to connect the Raspberry Pi to the necessary peripheral, such as (mouse, keyboard, monitor.. etc.) the raspberry Pi 3 B+ contain 4 USB ports, micro USB port to power the board, and HDMI connector, and an ethernet port:

1. Insert the SD card and Make sure that the Raspbian installed in it.
2. Connect the monitor via the HDMI cable
3. Connect the keyboard and mouse to the USB ports.

4. Connect the Ethernet cable
5. Connect the 5V power to the micro USB port.



Figure IV.4: Raspberry pi connections

Once all connections are done the raspberry Pi shown in Figure IV.4, then boot the board up and it will ask the user to log in, the raspbian default username and password:

- Username: pi
- Password: raspberry

Note that the default username and password are never secure, so one of the first things we are going to do is to change the Username and the password ahead in the configuration.

IV.2.2 Raspberry Pi configuration

The initial installation has finished, now the board needs some configurations as this project require, the Raspberry Pi comes with several features to enable depending on the user utilization, such as the Pi camera and the remote desktop .. etc.

IV.2.2.1 Using Raspi-config

the *Raspi-config* is a command-line used to access the configurations of the Raspberry board through the terminal.

To open the configuration table, click on the Raspberry Pi icon at the top left of the screen, then select Accessories then Terminal, and then the terminal window will display, Figure IV.5. Then type the command `sudo raspi-config`. This command opens the Raspberry Pi Software Configuration Tool as shown in Figure IV.6.

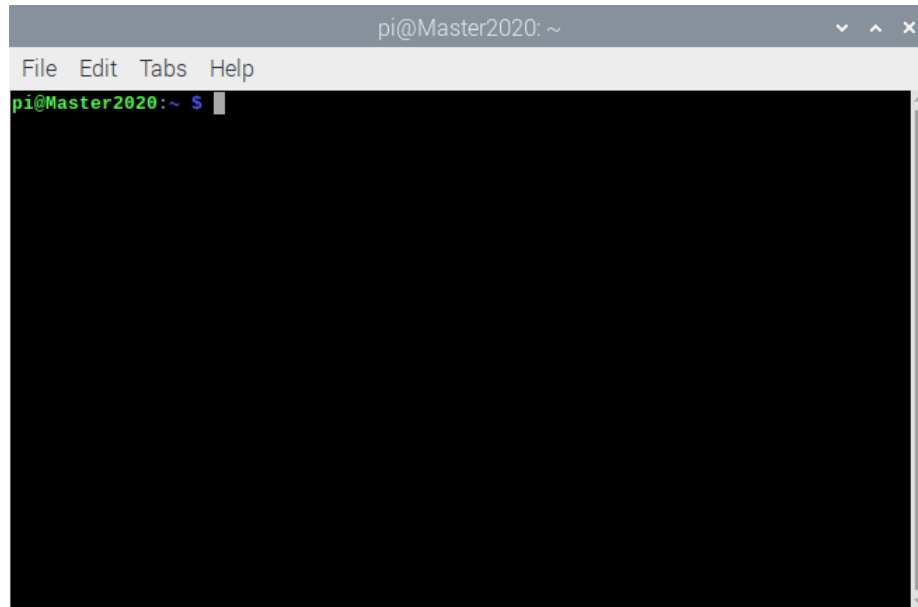


Figure IV.5: Raspbian Terminal widow

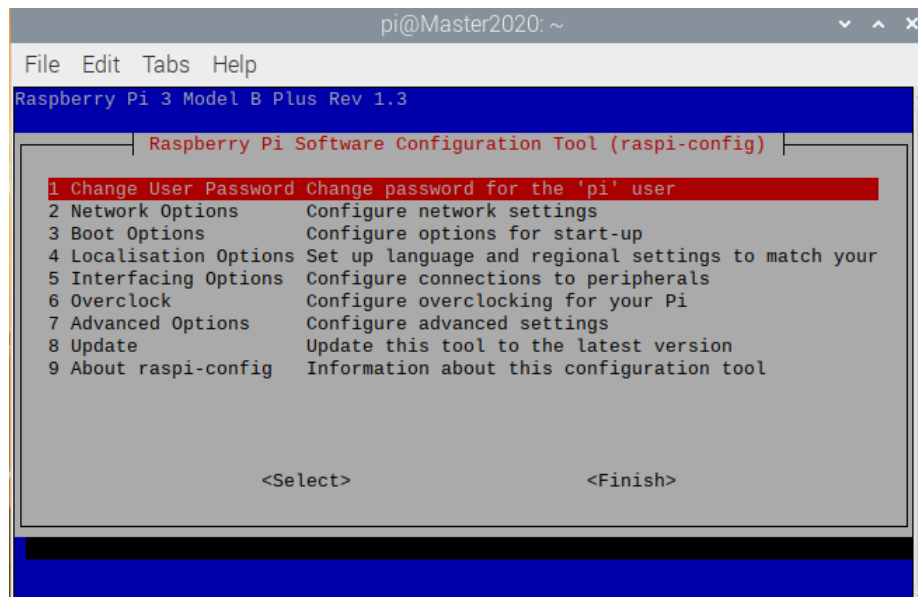


Figure IV.6: The Raspbian configuration table

IV.2.2.2 change the Raspbian Hostname

The first change in the raspberry configuration recommended to do is to change the default username and password, to do so, make sure that change password is highlighted as shown in Figure IV.6, press enter then a message will be displayed saying

“you will now be asked to enter a new password for the pi user”. press **OK** this drops you into the terminal to type the new user password.

The next step is to enable some features used in this project, starting by changing the Hostname of the Raspberry Pi. Make sure that the network options highlighted, then press Enter, the network options shown in Figure IV.7.

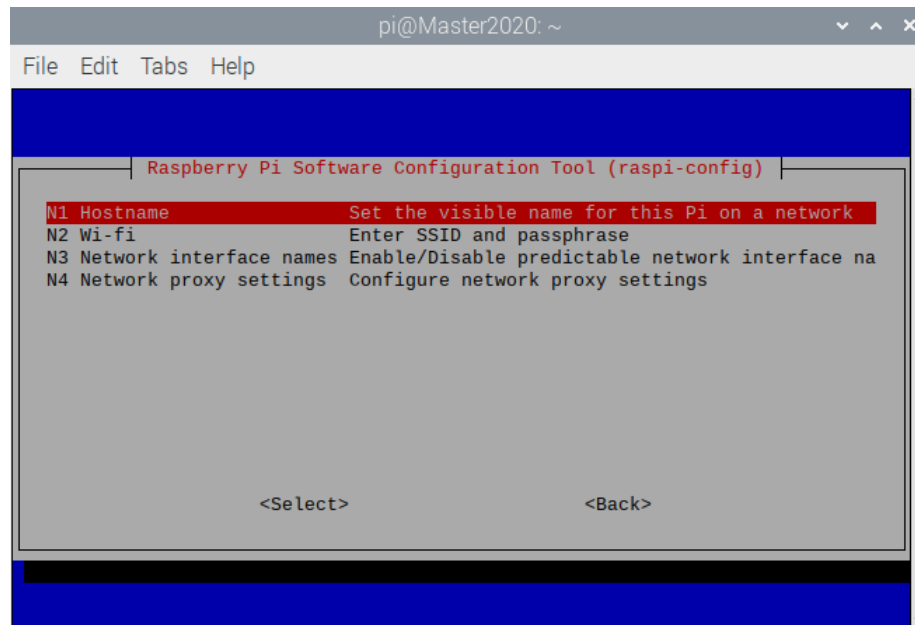


Figure IV.7: Network Options of the Raspbian OS.

the Hostname is how the Raspberry Pi will appear on the network, and it's preferable to give a unique name, below are the instructions to change the Hostname:

- Highlight the Hostname and press enter.
- A dialogue box will be shown explaining the requirement of a hostname, press enter to continue.
- type the hostname and press enter.

IV.2.2.3 Enable Raspberry Pi features

The SSH feature allows the user to access the Raspberry Pi through the terminal window (SSH client) from another computer. On Windows, PuTTY is a very popular free SSH client. SSH does not provide a GUI. All interactions are made using terminal commands. This is helpful if the user wants to quickly execute commands, program, or install software, [24] and here it's how to enable it:

- initially, make sure that Interfacing Options is highlighted, press **enter**, the interfacing options is shown in Figure IV.8 .

- select Enable SSH and press **Enter**.
- Confirm that the SSH is enabled and press **Enter**.

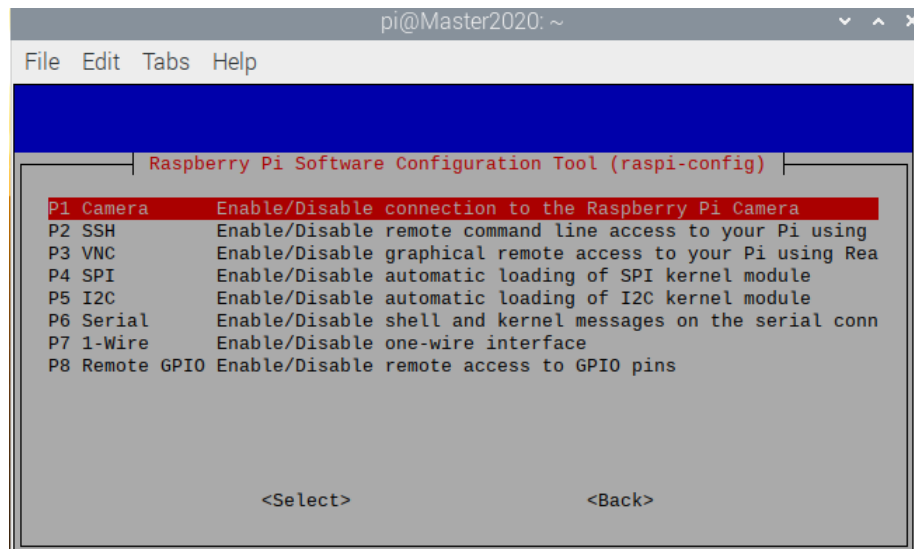


Figure IV.8: Interfacing Options of the Raspbian OS.

with the same menu and with the same way, enable the other features:

- Enable the use of Pi camera.
- Enable the use of GPIO pins.
- Enable Serial communication.

IV.2.3 Setup the the remote access

Essentially, there are two methods to remotely access the Raspberry Pi. One is by using the SSH, which allows you to connect a remote device using a terminal client, and the other method is to set up the remote desktop.[24]

IV.2.3.1 Remote desktop with xrdp

To set the Raspberry Pi headless (without a keyboard or monitor) by accessing Raspberry the Raspberry Pi remotely using another computer. Windows comes with the remote desktop connection already installed, for users to remotely access other desktops.

To achieve the remote access, It's mandatory to install some services on the Raspbian, the TightVNC server and Xrdp. Installing these two services needs to type a few commands on the Raspbian terminal:

- Installing TightVNC server with the command `sudo apt-get install tightvncserver`.

- Installing `xrdp` with command `sudo apt-get install xrdp`.

After installations are finished, the Raspberry pi is read for the remote desktop access. To start the remote desktop using windows remote desktop, the raspberry board must be connected to a wireless network.

IV.2.3.2 Connect to a wireless network

The board has already connected the network with an Ethernet cable, which means it's already on the network. But, it's preferable to connect to a wireless network, because this allows to remote into the robot while it's moving, and send it to commands and view the video stream from the Pi camera.

The Raspberry PI 3 Model B+ comes with a Wi-Fi builtin, which used in this project to connect the Raspberry Pi to the network.

- To get connected to the Wi-Fi click on the network icon on the top right of the screen.
- Select the Wi-Fi from the list.
- Enter the security key of the Wi-Fi.

now the board is connected to the network

IV.2.3.3 Using Windows remote desktop

The remote connection is done using the IP address, Firstly, obtain the Raspberry IP address using the terminal command-line: `sudo ifconfig`.

After getting the Pi IP address, then to start accessing it remotely via Remote desktop connection, shown in Figure IV.9.

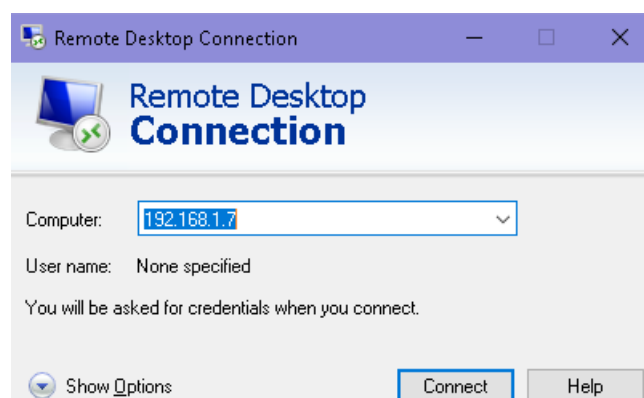


Figure IV.9: Windows Remote desktop Connection.

in the input field, enter the IP address of the Raspberry Pi, then click Connect. Once it's connected, the Xrdp login screen shown on figure IV.10, It asks the user to enter the username and the password.

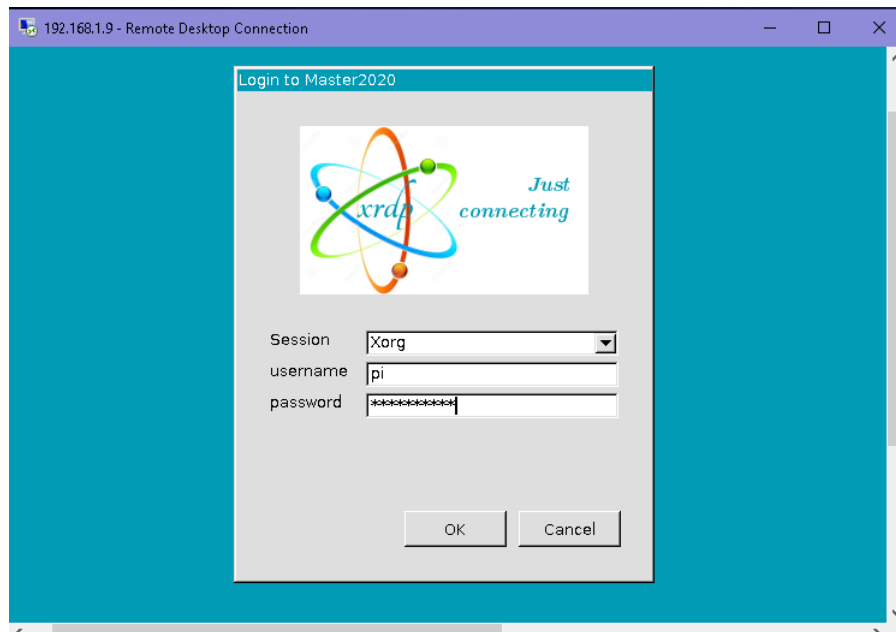


Figure IV.10: XRDP remote desktop login screen.

by now, setting up the remote desktop of the Raspberry Pi has finished, and note that as long as the IP address of the Pi doesn't change, it's no longer need the keyboard and the mouse or the monitor.

IV.3 Installing OpenCV in raspbian OS

IV.3.1 Python in raspbian

There are essentially two versions of python: Python 2.7 and Python 3, creator Guido van Rossum decided to clean up the code without putting a lot of emphasis on backward compatibility; therefore, some of the code for version 2.7 simply won't work on version 3 and vice versa. Python 3 is the current version and everything will eventually move over to it. [24]

Raspbian comes with both Python 2.7 and Python 3 already installed, and in our project, we will use the version 3 of Python because, in term of robotics the big hold-out was OpenCV which is the library we use in our project, and it's compatible with Python3.

IV.3.2 Installing OpenCV

Toward this project, the main work consists of computer vision and dealing with the digital image processing, to use OpenCV, firstly it needs to install on the Raspberry Pi.

there is no simple OpenCV installer for the Raspberry Pi. Because the Pi runs on an ARM processor, and the package has to be compiled from source code.[24]

there are several methods to install OpenCV on Raspberry Pi, and here it is how a method to install it:

1. First, it's mandatory to update the Raspbian System by typing this command on the Terminal:

- *sudo apt-get update*
- *sudo apt-get upgrade*

2. Installing dependencies by typing this commands:

- *sudo apt-get install build-essential cmake pkg-config*
- *sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev*
- *sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev*
- *sudo apt-get install libxvidcore-dev libx264-dev*
- *sudo apt-get install libgtk2.0-dev libgtk-3-dev*
- *sudo apt-get install libatlas-base-dev gfortran*

3. Installing Python3 and Pip3:

The Raspbian comes with python3 preinstalled, but to make sure it's already installed reinstall it.

- *sudo apt-get install python3-dev*
- *sudo apt-get install python3-pip*

4. Installing OpenCV using the pip3 install command:

- *pip3 install opencv-python*

IV.4 Set up the Pi camera module

In this project the camera is the main part, which considered the vision of this Following Robot using the Pi Camera, the Raspberry Pi Camera Module is a compact way of adding the ability to capture a video or images and is designed to connect to the Camera Serial Interface (CSI).

IV.4.1 Installing the Pi camera module

The Pi camera it's like the Raspberry Pi itself is supplied as a bare circuit board. It's connected to the Pi through a ribbon cable that should be inserted into the Raspberry Pi's CSI connectors, initially, the camera comes with one end of the cable has

already inserted to the module, one side of the other end is colored blue, and the other has silver contacts, these contacts connect to pins in the Pi's CSI connectors, which transfer power and data between the Pi and the Camera Module. [11]

the CSI connector of the Raspberry Pi 3 B is located toward the right side of the board just to the left of the Ethernet port (see Figure IV.11).

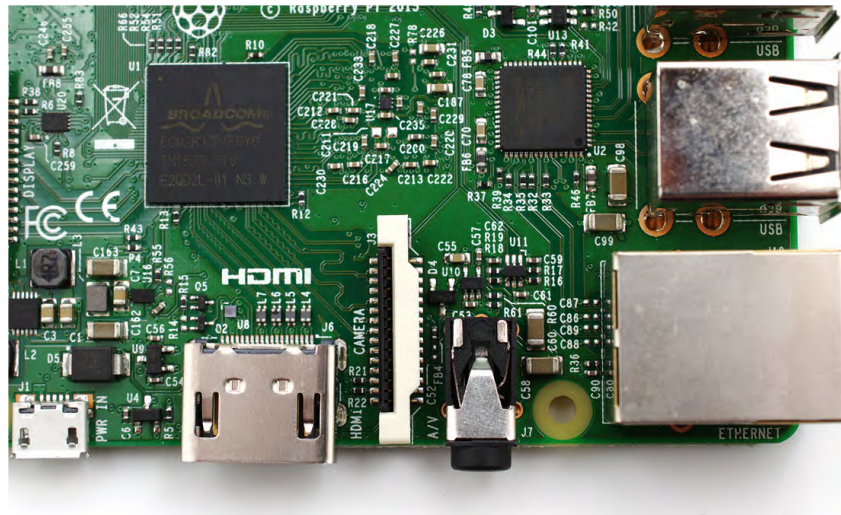


Figure IV.11: The Raspberry Pi 3 Model B CSI connectors.

There is a small lug on either side of the connector must be lifted a couple of millimetres before inserting the cable. Then inserting the free end of the robin cable into the CSI port making sure that the silver contacts are facing towards the left of the Pi and the blue portion of the cable to the right. the final result is shown in Figure IV.12.

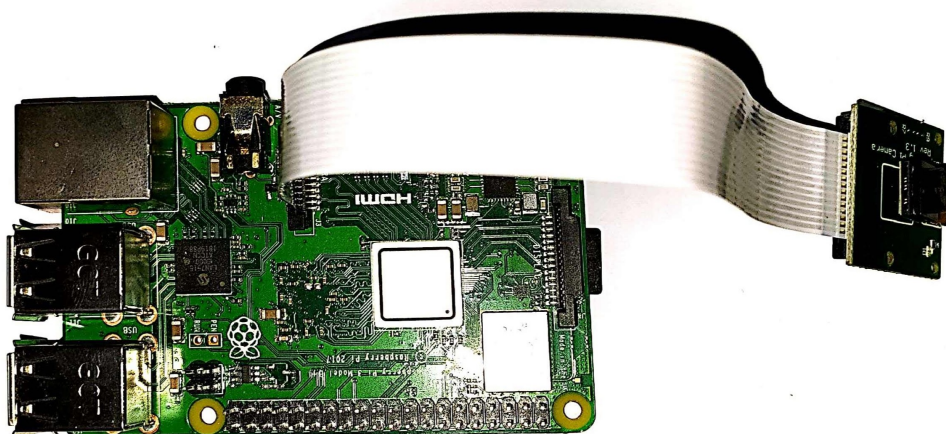


Figure IV.12: The Camera Module's ribbon cable connected with Raspberry via CSI connectors.

IV.4.2 Programming the Pi Camera Module

Before getting started using Pi camera, the Camera will not work unless it's been a camera-ready by using the Raspberry Pi Configuration tool “raspi-config” to enable the use of the Camera Module. Once the Raspberry Pi reboots, the Camera is ready to use.

There are several ways to capture pictures or stream live video through the camera module, in this project the Pi Camera has been programmed via utilization Python programming language. the Python code that gets video stream from the camera is shown below:

```
1 import picamera
2 from picamera.array import PiRGBArray
3 import cv2
4 import time
5
6 camera = picamera.PiCamera()
7 camera.resolution = (640,480)
8 camera.framerate = 25
9 rawCapture =PiRGBArray(camera, size = (640,480))
10 time.sleep(1)
11 for frame in camera.capture_continuous(rawCapture, format='bgr',
    use_video_port = True ) :
12     image = frame.array
13     cv2.imshow('preview',image)
14     rawCapture.truncate(0)
15     key = cv2.waitKey(1) & 0xFF
16
17     if key == ord('a'):
18         break
19
20 cv2.destroyAllWindows()
```

Note that the Picamera library is not among the python standard library, it needs to be installed with the pip command before using it.

- the first four lines are obligatory because their rule is to import necessary packages which will be used later on.
- At lines 6,7 and 8, it refers to the object instantiation of the class PiCamera with the resolution video of 640x480 and the frame rate at 25fps.
- Line 9 is for creating an instance of the class PiRGBArray named rawCapture which takes two parameters as an input, the Camera and the size, and produces a 3-dimensional RGB array organized (rows, columns, colors).
- The code at lines 11-18 loops over a set of instructions:

- in each iteration, the loop takes an image frame returned via the function `Camera.capture_continuous` which takes three parameters as input.
 - Create a variable and store the image frame in it.
 - Preview the image frame at the 13th line using OpenCV function `cv2.imshow`, this later takes two inputs, the window name and the image will be shown.
 - At the 14th line, the truncate function clears the stream to prepare for the next frame.
 - The purpose of the code at lines 15-18 within the for loop is to break the loop if the user presses the key 'a'.
- The last line in the code which is the OpenCV function `it'` has been used to close all the window has been created right after the loop break.

Example of picture captured using the Raspberry Pi Camera shown in Figure IV.13



Figure IV.13: Picture captured via Raspberry Pi camera .

IV.5 Interfacing Arduino with Raspberry Pi

Both Raspberry Pi and Arduino are important parts in this project, Arduino board is excellent at a low process such as reading inputs and sending outputs, that's the reason the Arduino board used to control this robot wheels, in another hand the Raspberry board used for the High-level process that requires high-speed treatment such

as image processing technique. The functionality of this robot requires that these two units must communicate together, to do that the two boards are connected using USB Serial Communication, this is a suitable way to communicate between them. To do that must connect the Arduino Nano board to the Raspberry Pi over a USB cable as shown on Figure IV.14

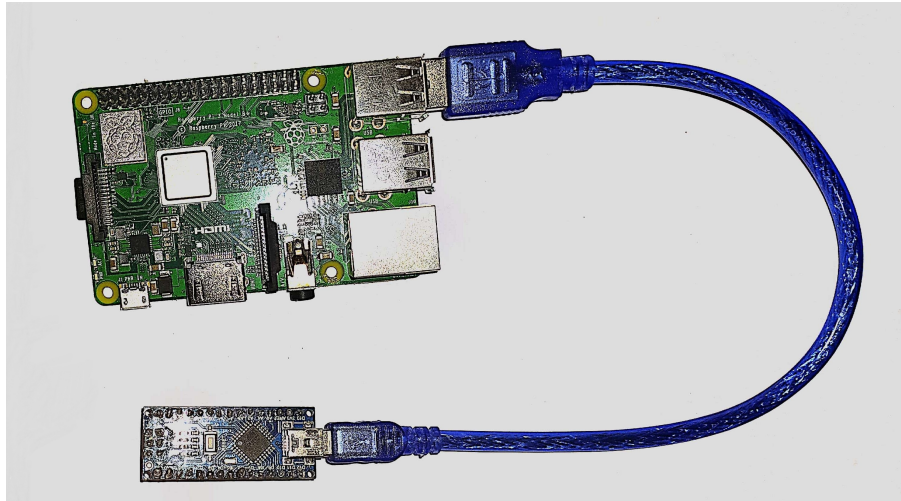


Figure IV.14: Arduino Nano with Raspberry Pi

IV.5.1 serial communication

We can achieve Serial communication with aid of Python programming language which provide a preinstalled library for Serial communication, Python code is shown below:

```
1 import serial
2 import time
3
4 ser = serial.Serial("/dev/ttyUSB0",9600)
5 ser.baudrate = 9600 ;
6
7 ser.write('f'.encode())
8
9 time.sleep(1.5)
10
11 ser.write('s'.encode())
12
13 time.sleep(1)
```

IV.5.2 Connecting to Serial

the code that connects the Raspberry Pi to the Arduino board over Serial shown above:

- initially, the first two lines show the importing of necessary packages.
- then in the next line Create an instance of the class Serial named Ser. which take two parameters as input:
 - `"/dev/ttyUSB0"`: the USB port used
 - `9600`: is the communication speed in bits per second

Note that the serial instant returns a boolean value that indicates whether or not a serial connection is available or not, it returns True or False.

IV.5.3 Sending data through Serial

Sending data to serial is done with the help of the `write()` method, it sends the data within the parenthesis to the serial port, and the data can be seen via the Arduino IDLE serial monitor. with the same way, it's possible to receive and read data using the `Readline()` method.

IV.5.4 Reading data using Arduino

the Arduino code shown below, perform the serial connection and read the data over the serial port:

```
1 byte data ;
2
3 void setup () {
4 // initialize serial:
5 Serial.begin(9600);
6
7 }
8
9 void loop () {
10
11 if (Serial.available () > 0) // check if serial is avialable
12 {
13 data = Serial.read (); // read data
14 } }
```

- Initially, at the first line referring to a variable declaration with type byte to store the input data on it.

- In the setup function, at lines 3-7 initialization of the serial connection and setting the communication speed at 9600.
- The `serial.Available()` function at line 11 returns the number of character arrived. according to that, the if statement used that to check if the serial's available.
- At line 13, and with the aid of `read()` function, it's possible to read the input data and store it in the data variable.

IV.6 Control the wheels with Arduino

the MX1508 DC Motor Driver has been used to control both the speed and the direction of the wheels, it comes incorporates an H-bridge for both motors, therefore it can drive the motors in either direction and the speed of the motors is controlled with a Pulse Width Modulated (PWM) signal.

IV.6.1 Connection

the lithium battery poles are connected to the MX1508 power pins, and the outputs of the motor drive go to the DC motor inputs, the control signal inputs of the motor drive are comes from the PWM Arduino outputs. Figure IV.15.

Note that the MX158 Motor driver able to control two DC motors simultaneously, therefore two of MX1508 has been used to run the four wheels of the mobile robot.

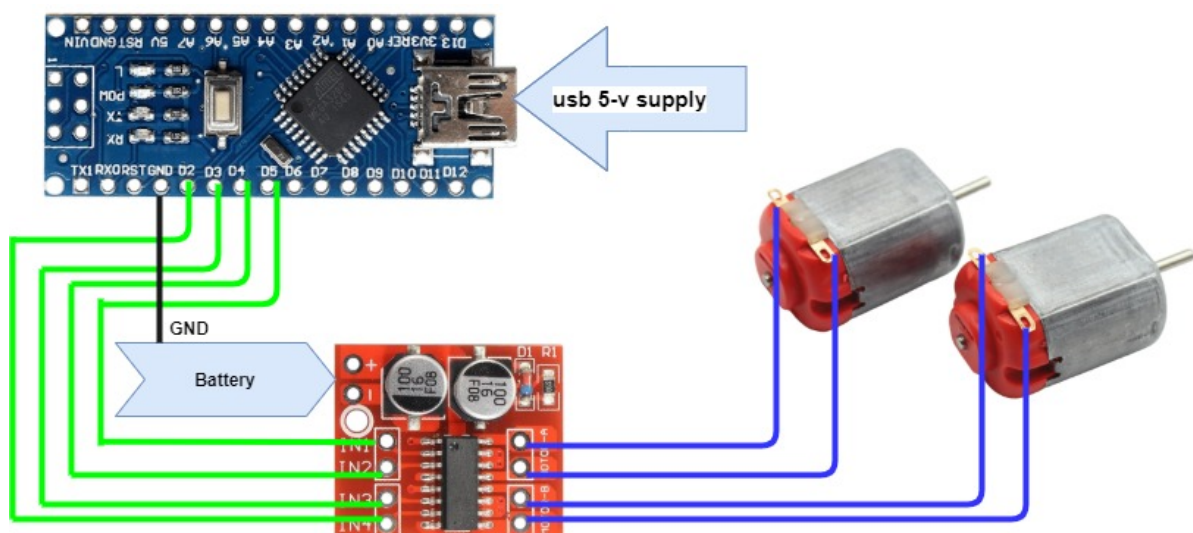


Figure IV.15: DC Motors controlled with Arduino board using the MX1508 Motor drive.

IV.6.2 Control speeds the wheel

the speed of the DC Motor is controlled via controlling the flowing current through the motor, usually, they attempt to do this using a large variable resistor wired in series with the DC Motor as shown in Figure IV.16.

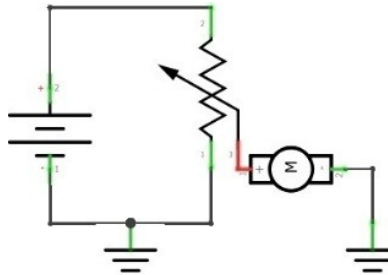


Figure IV.16: DC Motor Speed Control using Variable resistor

Analogue voltages and currents can be used to control things directly like controlling the speed of the motor, the current flowing through the variable resistor increase or decrease according to the value of the resistor, this changes the speed of the Motor, it is not always economically attractive or otherwise practical. Because analogue circuits tend to drift over time and it generates a lot of heat and wasted power. [25]

To control the speed of a motor is to manipulate the value of the voltage flowing across its terminals using PWM (Pulse width Modulation) generated via Arduino.

IV.6.3 Pulse width Modulation (PWM)

PWM is a way of digitally encoding analogue signal levels. Through the use of high-resolution counters, the duty cycle of a square wave is modulated to encode a specific analogue signal level. The PWM signal is still digital because, at any given instant of time, the full DC supply is either fully on or fully off. The voltage or current source is supplied to the analogue load utilizing a repeating series of on and off pulses. The on-time is the time during which the DC supply is applied to the load, and the off-time is the period during which that supply is switched off. Given a sufficient bandwidth, any analogue value can be encoded with PWM. [25]

Figure IV.17 shows three different PWM signals. Figure IV.17a shows a PWM output at a 60% duty cycle. the signal is on for 60% of the period and is off the rest 40%. Figures IV.17b and IV.17c show PWM outputs at 50% and 30% duty cycles, respectively. These three PWM outputs encode three different analogue signal values if the supply voltage assumed is 5-volts the outputs at 60%, 50%, and 30% are 3V, 2.25V, 1.5V, respectively.

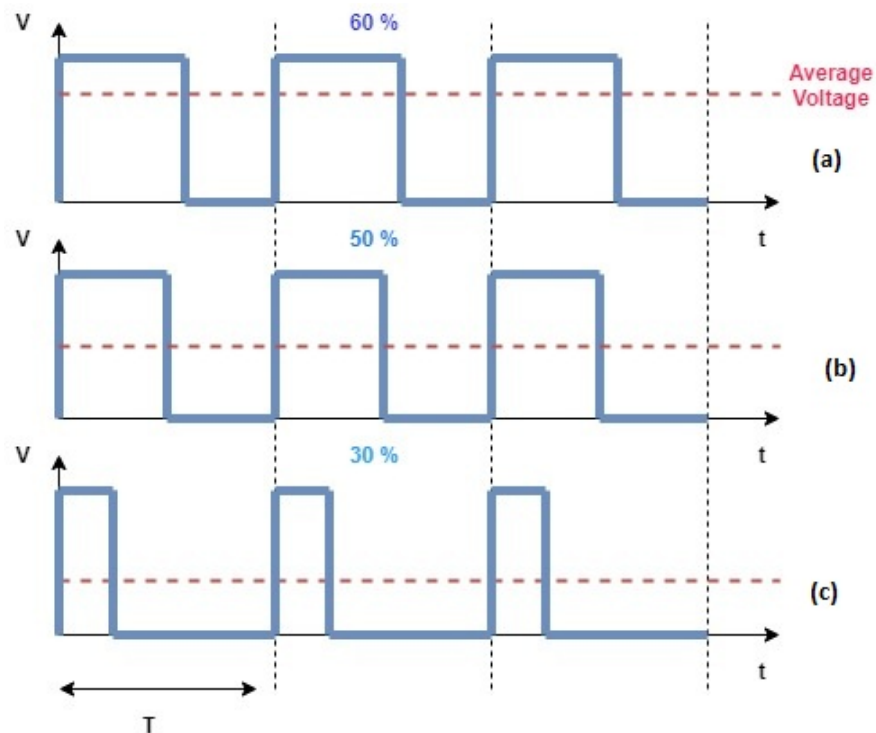


Figure IV.17: different PWM Signals.

IV.6.4 programme Arduino board to control the wheels

Arduino controls the wheels speed via generating a PWM signal through the PWM output Pins and to generate a PWM signal to a pin, the `analogWrite()` function does that, this function takes two parameters as an input, `analogWrite(parameter1,parameter2)`.

- Parameter1: the number of PWM pin (Arduino Nano Board has 6 PWM pins: 3,5,6,9,10,11), must be integer type int.
- Parameter2: is duty cycle value between 0 and 255. allowed data types: int.

IV.6.4.1 Arduino script code:

First thing is to declare the variables that used in this code which refer to the Pins that controls the wheel speeds and directions, each one of the wheels has two pins controlling It. In this mobile robot, There are four wheels which need 8 pins, there for 8 variable the code shown below:

```

1
2 //front
3 const int FrontRight_Fpin = 2; //Wighte
4 const int FrontRight_Bpin = 3; //Green
5 const int FrontLeft_Fpin = 4; // pink wire
6 const int FrontLeft_Bpin = 5; ///blue

```

```

7
8 //Back
9 const int  BackRight_Fpin = 6; // white
10 const int  BackRight_Bpin = 7; //white
11 const int  BackLeft_Fpin = 8; //brown
12 const int  BackLeft_Bpin = 9; //red

```

Note that the Arduino Nan board has just 6 PWM pins, which is insufficient, to solve this dilemma we used four PWM pins to control the Robot Forward and four digital pins to control it backward. that because this mobile robot doesn't need to move backwards, just in one case if the robot gets too closer to the target object then it's moving backwards to stop at the right position. Then in setup function using pin-Mode() function to set all pins as outputs:

```

1
2 void setup ()
3 {
4 pinMode(FrontRight_Fpin , OUTPUT);
5 pinMode(FrontRight_Bpin , OUTPUT);
6 pinMode(FrontLeft_Fpin , OUTPUT);
7 pinMode(FrontLeft_Bpin , OUTPUT);
8 pinMode(BackRight_Fpin , OUTPUT);
9 pinMode(BackRight_Bpin , OUTPUT);
10 pinMode(BackLeft_Fpin , OUTPUT);
11 pinMode(BackLeft_Bpin , OUTPUT);
12 }

```

To make the robot move forward, backwards and turn to the right or left, we must change the directions of the wheels with four cases listed below:

- Move forward: all the wheels moving forward.
- Move Backward: all the wheels moving backwards.
- Turning left: the right wheels are moving forward and the left wheels are moving backwards.
- Turning right: the left wheels are moving forward and the right wheels are moving backwards.
- Stop moving: set all pins to zero.

```

1 void front () {
2   analogWrite(FrontRight_Fpin ,HIGH) ;

```

```
3  analogWrite ( FrontLeft_Fpin ,HIGH) ;
4  analogWrite ( BackLeft_Fpin ,HIGH);
5  analogWrite ( BackRight_Fpin ,HIGH) ;
6  }
7  void back () {
8    digitalWrite ( FrontRight_Bpin ,HIGH) ;
9    digitalWrite ( FrontLeft_Bpin ,HIGH) ;
10   digitalWrite ( BackLeft_Bpin ,HIGH) ;
11   digitalWrite ( BackRight_Bpin ,HIGH) ;
12  }
13 void stopp () {
14   digitalWrite ( FrontRight_Fpin ,LOW) ;
15   digitalWrite ( FrontLeft_Fpin ,LOW) ;
16   digitalWrite ( BackLeft_Fpin ,LOW) ;
17   digitalWrite ( BackRight_Fpin ,LOW);
18
19   digitalWrite ( FrontRight_Bpin ,LOW) ;
20   digitalWrite ( FrontLeft_Bpin ,LOW) ;
21   digitalWrite ( BackLeft_Bpin ,LOW) ;
22   digitalWrite ( BackRight_Bpin ,LOW) ;
23 }
```

In the the loop section, the Switch Case statement used to choose which function needs to be executed according to the data received from the serial bus, the code shown below :

```
1  switch ( data ) {
2    case ( byte) 's' : {
3      stopp () ;
4      break ;}
5    case ( byte) 'b' : {
6      back () ;
7      break ; }
8    case ( byte) 'f' : {
9      front () ;
10     break ; }
11   case ( byte) 'l' : {
12     left () ;
13     break ; }
14   case ( byte) 'r' : {
```

```
15   right();  
16   break; }  
17   default: {  
18     stopp(); }  
19 }
```

IV.7 Sharp IR with Arduino

In this project, the Sharp IR proximity sensor is used to calculate the distances for each instance between our robot and the target object.

IV.7.1 Connections

The sharp IR GP2Y0A41SK0F has three connectors, two used for the power supply (red and the black are connected to the GND and 5-V respectively), and the yellow one is the analogue output, which connected to one of the analogue Arduino inputs. as shown in Figure IV.18

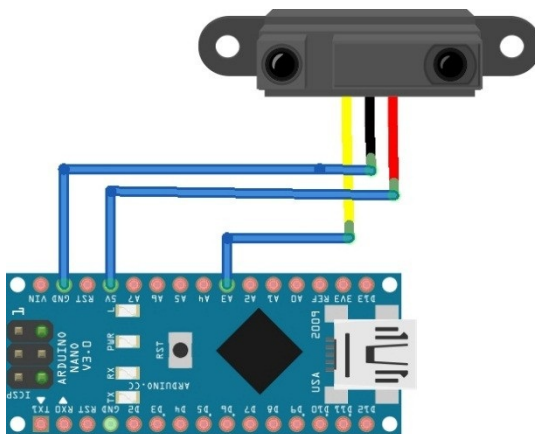


Figure IV.18: sharp GP2Y0A41SK0F connections with Arduino Nano board

IV.7.2 Arduino code

This distance sensor programmed with the Arduino board using a library that needs to be installed, the code that gets the distance is shown below:

```
1 #include <SharpIR.h>  
2  
3 SharpIR sensor( SharpIR::GP2Y0A41SK0F, A0 );  
4  
5 void setup()
```

```
6 {
7 Serial.begin( 9600 ); //Enable the serial communication
8 }
9
10 void loop()
11 {
12 int distance = sensor.getDistance(); //Calculate the distance
    in centimeters and store the value in a variable
13
14 Serial.println( distance ); //Print the value to the serial
    monitor
15 }
```

- Firstly, importing the necessary libraries.
- Create an instance from the class and set the input pins to Analog A0.
- In the setup function, initialization of the serial to print the input data.
- In the loop function, declare a variable to store the data obtained using the **getDistance()** function.
- Then printing the distance on the serial monitor.

IV.8 Robot parts installation

After we discussed briefly each part of our project, these parts needs to be assembled and each part needs to be installed in its place to make this mobile robot work perfectly:

1. First, we need to fix our motor drives to the chassis and connect the 4 motors to it. See figure IV.19
2. The distance sensor is placed in the front of the chassis.
3. attach the Camera Mount on the top front of chassis, and fix the Pi camera to it.
4. On the top of the chassis, we have placed an Arduino board, Raspberry Pi, and the battery pack, the Arduino Nano board is a friendly breadboard, then connect all the necessary wire to the breadboard, Arduino board connection with Raspberry Pi is related via USB cable, to achieve the serial communication, furthermore to provide energy power to the Arduino board . Figure IV.20

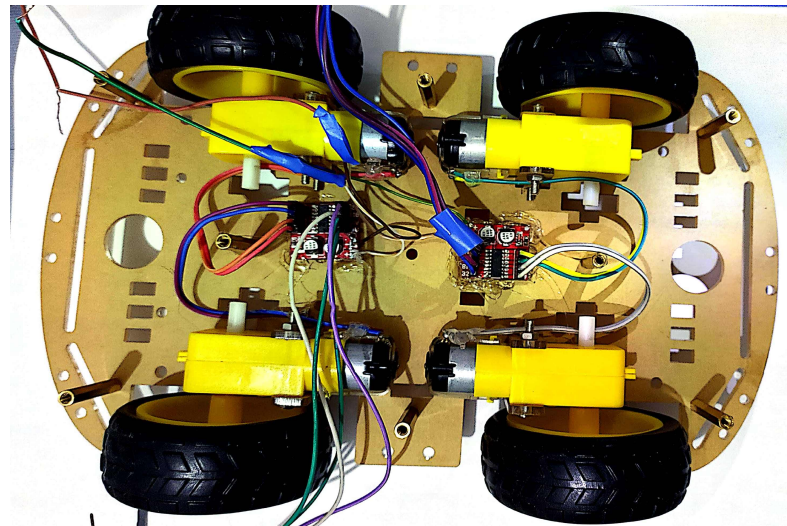


Figure IV.19: connection of the 4 wheels with the motor drives

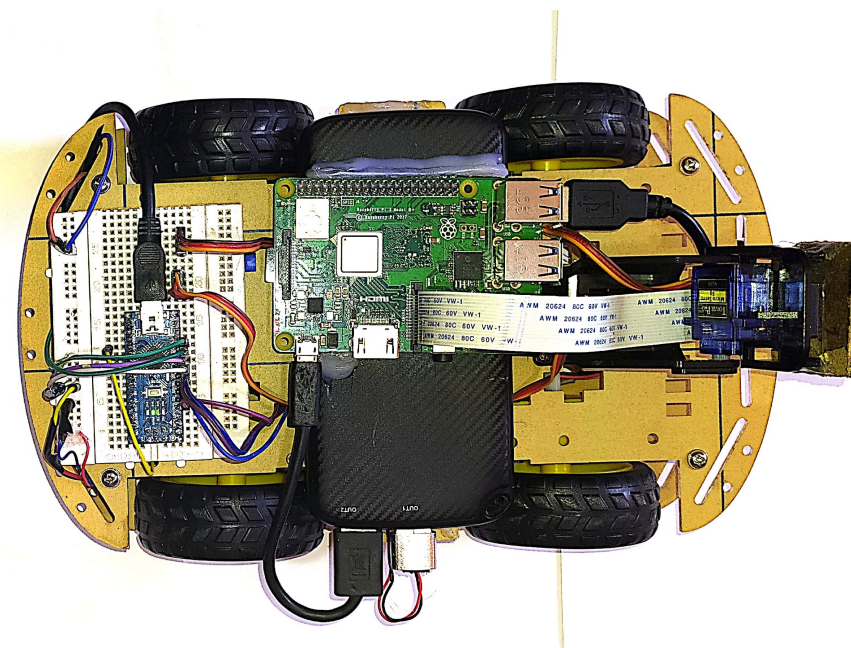


Figure IV.20: Complete assembled robot

IV.9 Conclusion

In this chapter we went through the necessary setups of the Raspberry Pi board, we took the basic installation, starting with the installation of Raspbian OS, which is an important step for the Pi to be used, furthermore, we used the rapti-config to configure the board, and enabled some useful features such as the remote access which allows us to remotely access the robot. Also, we showed programming and assembly of different components which our robot consist of.

In the next chapter, we will use image processing to recognize and track objects using color segmentation method in different color spaces with the raspberry pi board.

Object tracking using color detection with raspberry pi board

V.1 Introduction

Object detection has been a very dominant field of computer vision researches and developments nowadays, it consists of finding the object location in a scene using image features. Digital images have properties known as image feature which plays the main role in the development of advanced computer vision algorithms over the recent years, it makes the camera an extremely powerful sensor used to detect, identify and track objects in the camera view. Applications based object recognition has a variety of uses, some of which are: Self-driving car, Autonomous mobile robots, security and surveillance etc.

Mobile robots can navigate in their environments autonomously, which needs a path planning algorithm to determine its position and direction, this project uses a vision based-navigation to extract features required for the localization process. To keep track the target using color detection, it repeats the following steps:

- the robot captures an image frame sequence continuously.
- process each frame to detect the specific color.
- Path localization for the robot to follow the color desired.

In this chapter we are interested in the detection of colored objects, objects have features such as color and shape, they're used to track the object in a frame sequence. We will discuss the detection of an object with a specific color Using different color spaces which is an important step for color detection, there are a variety of color models some of which are: RGB, HSV, YCrCb, LAB... etc. and finally we will discuss localization of the robot based on object location in the image.

V.2 Color feature

The digital image is a discrete signal can be represented as a 2-D matrix of numbers that represent either light intensity, gray level, color, or other quantity. Digital images are distinguished in several types, (Figure V.1) are color (Figure V.1a), grayscale (Figure V.1b), binary (Figure V.1c). A gray-scale image is used to identify patterns that are not color-dependent. The binary image is the result of image segmentation and used for content masking.

Image segmentation uses a thresholding method, which set the pixels that are above a threshold value to a logical one, and the rest to logical zero, for example, the binary image in figure V.1c is a result of thresholding of the gray-scale image in figure V.1b with a thresholded value at 83.

Detection of color is not an easy task due to the different lighting conditions, shadows and reflections, which mean that the color based feature segmentation is more efficient in the environment where we can control the illumination conditions, (Figure V.2). Colors in a digital image are normally represented with three color components: red, green, and blue; this is known as the RGB color model. The color is restored from a combination of red, green, and blue color components that are filtered through red, green, and blue color filters, respectively. there are other color spaces used in machine vision which gives a natural color description and enables better color segmentation than can be achieved in RGB space.[1]

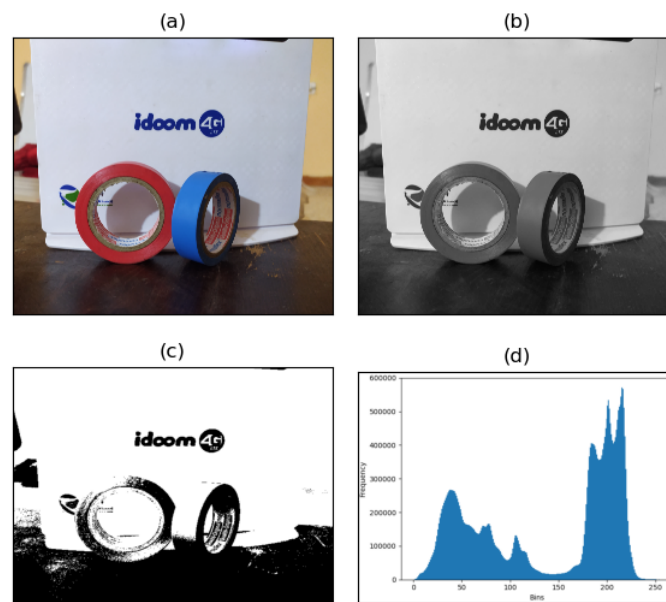


Figure V.1: (a) color image, (b) grayscale image, (c) binary image, (d) grayscale histogram.

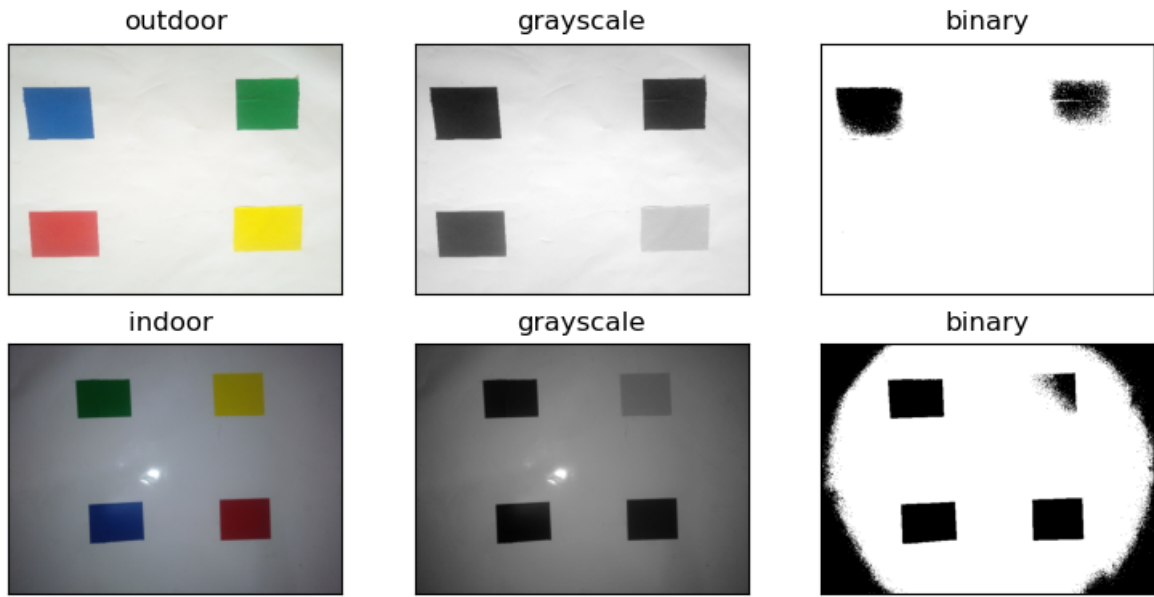


Figure V.2: result of threshold for two images taken under different illumination.

V.3 Color spaces conversions

To display color represented with specific color spaces with other color space, it needed to be converted. Color space is a mathematical model to represent color information as three or four different color components. Different color models are used for different applications such as computer graphics, image processing, TV broadcasting, and computer vision.[26]

- The Transformation of an image with RGB color space to an HSV image using equations V.1, V.2, V.3 :

$$H = \arccos \frac{\frac{1}{2}(2R - G - B)}{\sqrt{(R - G)^2 - (R - B)(G - B)}} \quad (\text{V.1})$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} \quad (\text{V.2})$$

$$V = \max(R, G, B) \quad (\text{V.3})$$

- The transformation between YCrCb and RGB image using the equation V.4 :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.279 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{V.4})$$

V.4 Comparative of color detection in different color spaces

In this section, we will discuss color object detection in some important color spaces used in computer vision and compare between the results of color detection obtained in each color space.[27] The algorithm (Figure V.3) of color image segmentation based in different color space is explained as follows:

- The input image obtained from a digital camera.
- The image initially is represented with RGB color spaces (OpenCV uses the BGR representation), the default RGB image is converted into a specific color space.
- Compute the histogram for the all color spaces component, and according to the histogram, the threshold values is determined.
- Threshold the image to obtain the binary image with the specific color region.
- The binary image is smoothed and filtered.
- Find contours.

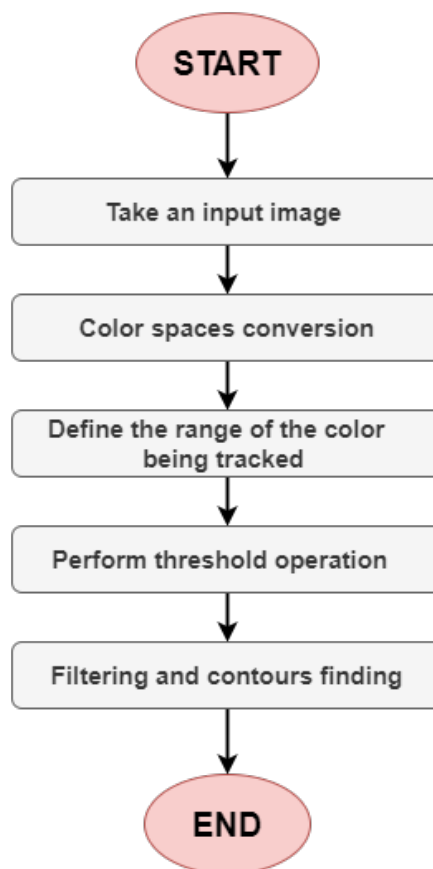


Figure V.3: Flow Chart for Object Tracking by Color Detection

V.4.1 RGB color space

RGB color space is an additive color space where colors are obtained by a combination of Red, Green, and Blue Values, Figure V.4 shows the R, G, B components separately.

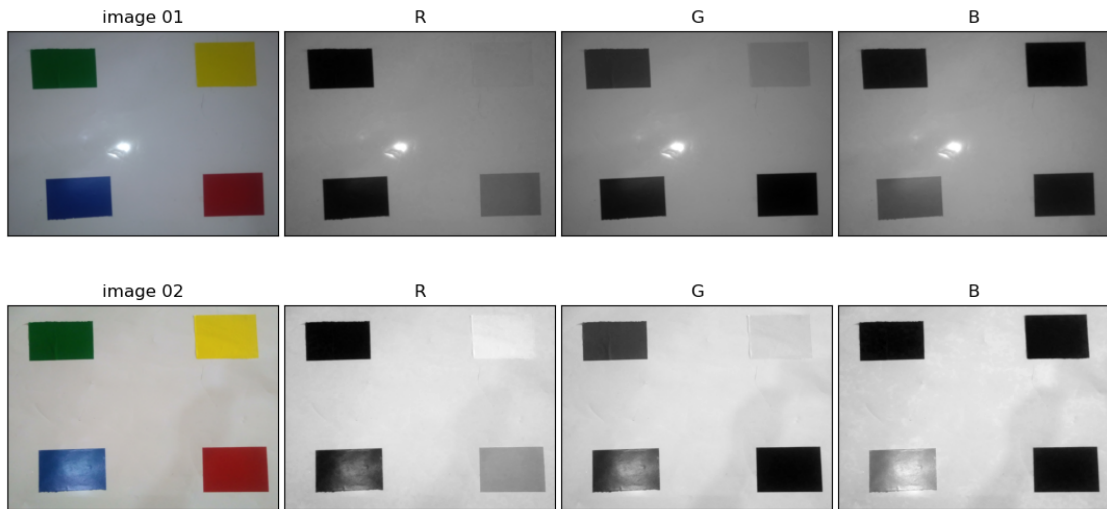


Figure V.4: Different Channels Blue (B), Green (G), Red (R) of the RGB color space shown separately.

In the RGB color space, the information of specific color is separated into three channels and the same channels encode the brightness information which makes color-based segmentation difficult in this color spaces, Figure V.4 show component RGB color space for the same image under two different illumination conditions, and the histograms of the R, G, B channels for both images shown in figure V.5, and It's clear that the change in illumination in both images, affects all the three RGB components.

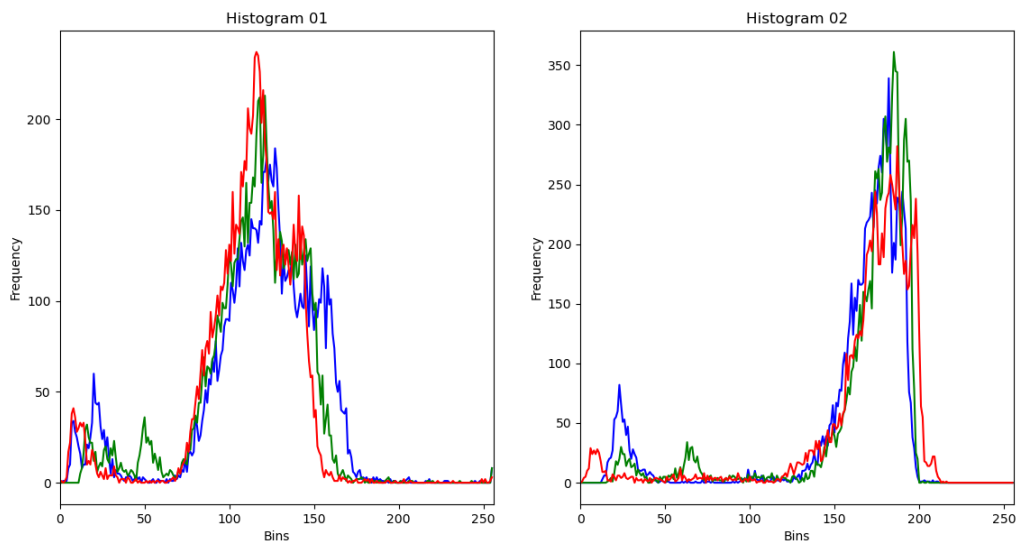


Figure V.5: histogram of R, G, B for the same images with different lightness.

V.4.2 YCrCb color space

The YCrCb color is another color space just like the RGB but, in the YCrCb color spaces the Y channel encodes the luminance which is independent of the color information. Which mean that the luminance and the chrominance are separated into different channels.[27] The YCrCb color space is a way of encoding RGB information, the main colors are displayed depending on the actual RGB. [28] The YCrCb has three components the same as RGB color spaces:

- Y: Luminance component.
- Cr: = R-Y (represent how far is the red from Luma).
- Cb: = B-Y (represent how far is the blue from Luma).

Figure V.6 shows the components of the YCrCb color space separately and their histograms in figure V.7, according to that we observe that the change of illumination in both image mostly affects only the L component and the Cr and Cb component for in both cases are nearly the same because the Cr and Cb encode the color information and independent of the luminance.

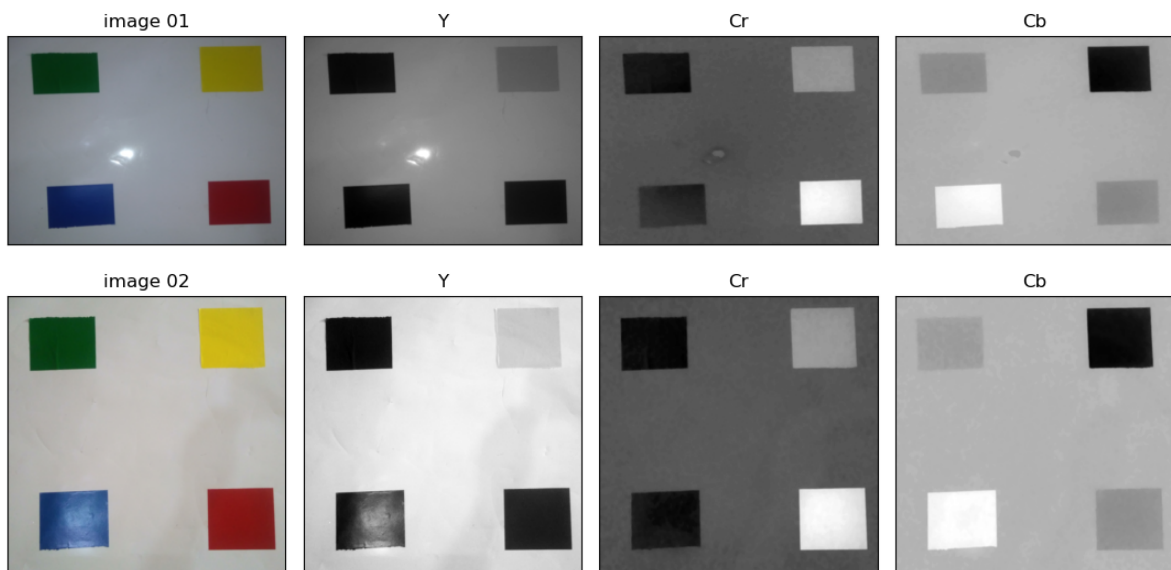


Figure V.6: YCrCb color space components for two image with different lightness conditions.

V.4.3 HSV color space

The HSV The HSV Color space has the following component H: Hue (the Color), S: Saturation (shades of the color), V: Value (Intensity). HSV color space uses one channel to describe the color information (H), which make it flexible to specify a color.

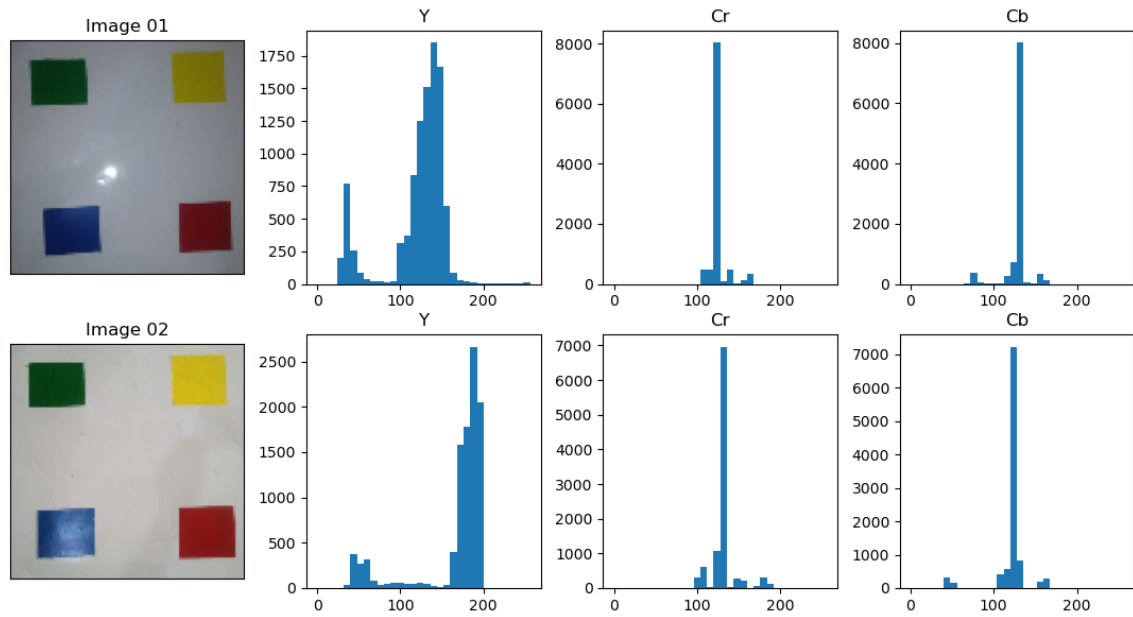


Figure V.7: Histogram Y, Cr, Cb component for two images with different illumination.

Figure V.8 show the components of HSV image under two illumination conditions. We observe that the S and H components are quite similar in both images, and V component is different, the V value represents the amount of light, therefore is change due to illumination conditions.

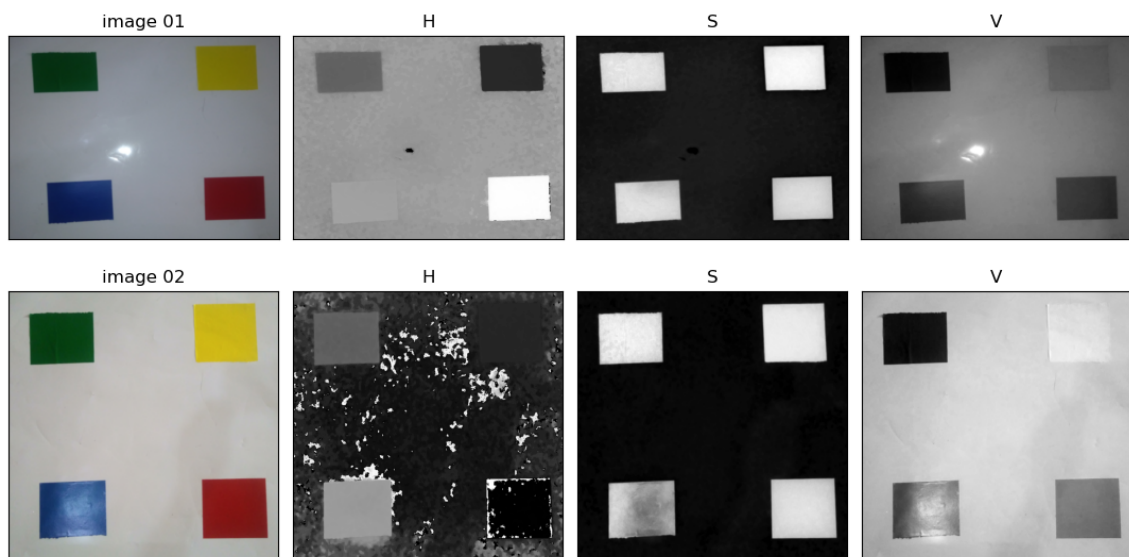


Figure V.8: Hue (H), Saturation (S) and Value (V) components in HSV color space.

Color histogram can be used for color segmentation for regions with a particular color, HSV colors histogram for each component for the red, blue, green and yellow colors are shown in figure V.9. As can be seen the Hue histogram is obviously different from color to another which refer that the H component represents the color itself, which make color identification is quite efficient for the HSV color space.

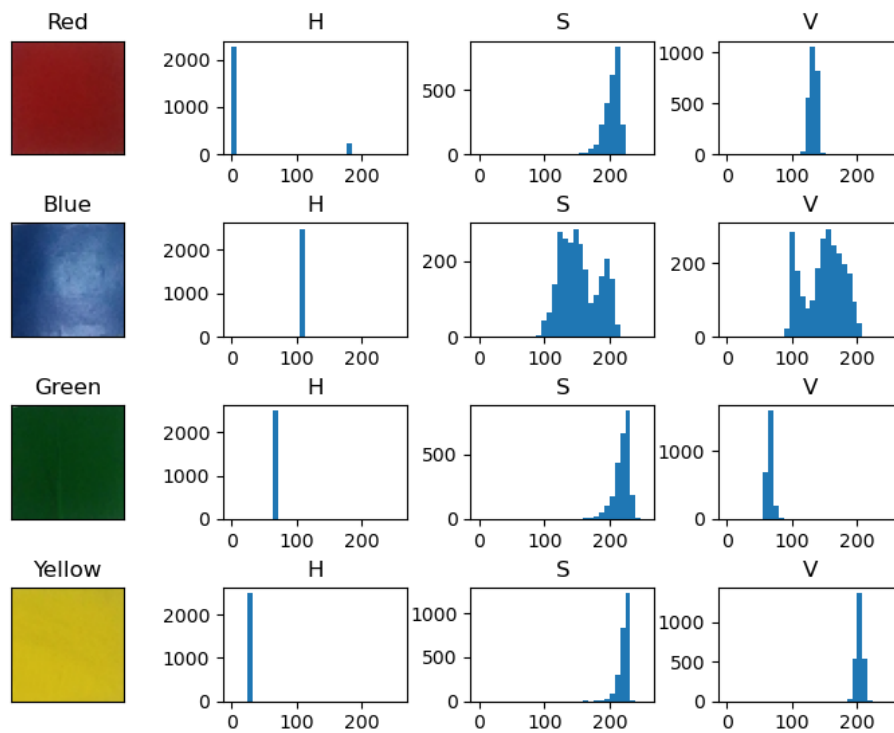


Figure V.9: Histogram of the HSV color space component for different colors.

V.4.4 Results of color detection

Color detection in the RGB, YcrCb, HSV color spaces uses the boundaries (Lower Values and Upper Values) for the specific color to be detected. The obtained results are reported in Tables V.1, V.2, V.3, which are presented the lower and upper boundaries for the red, blue, green, yellow colors in the RGB, YCrCb, HSV color spaces respectively, and The result of detection of the four colors are shown in Figures V.10, V.11, V.12 .

it should be noted that the using fixed threshold values (specified manually) for both images, doesn't obtain a good results in the RGB color space due to the mixing of color information and intensity luminance.

color space	R		G		B	
	THmin	THmax	THmin	THmax	THmin	THmax
Red	17	64	14	41	41	199
Blue	57	220	10	88	5	32
Green	0	27	0	100	0	41
Yellow	0	46	95	179	97	195

Table V.1: RGB min and max threshold values of red, blue, green, yellow colors.

color space	R		G		B	
Boundary	THmin	THmax	THmin	THmax	THmin	THmax
Red	10	75	125	191	5	117
Blue	8	148	26	113	139	168
Green	15	78	98	113	35	135
Yellow	11	225	10	255	16	80

Table V.2: Y, Cr, Cb min and max threshold values of red, blue, green, yellow colors.

color space	R		G		B	
Boundary	THmin	THmax	THmin	THmax	THmin	THmax
Red	0	41	78	255	89	160
Blue	89	130	89	255	8	255
Green	45	79	119	255	7	255
Yellow	13	45	60	255	9	255

Table V.3: H, S, V min and max threshold values of red, blue, green, yellow colors.

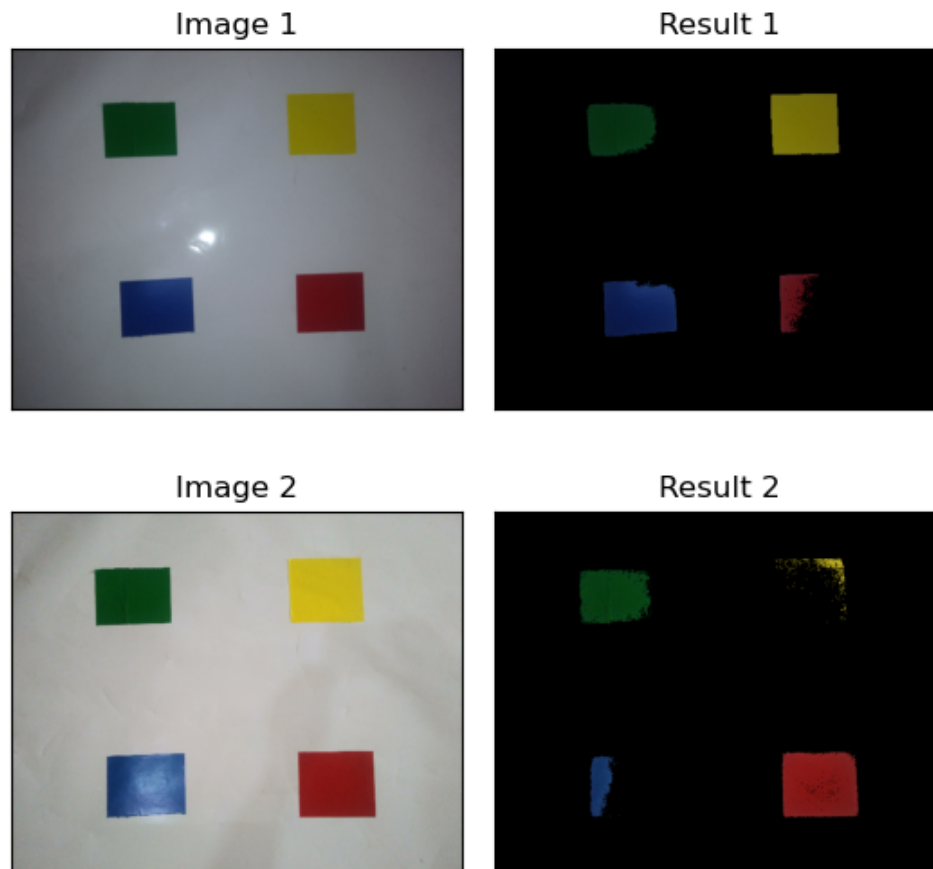


Figure V.10: detection of different colors in the RGB color space.

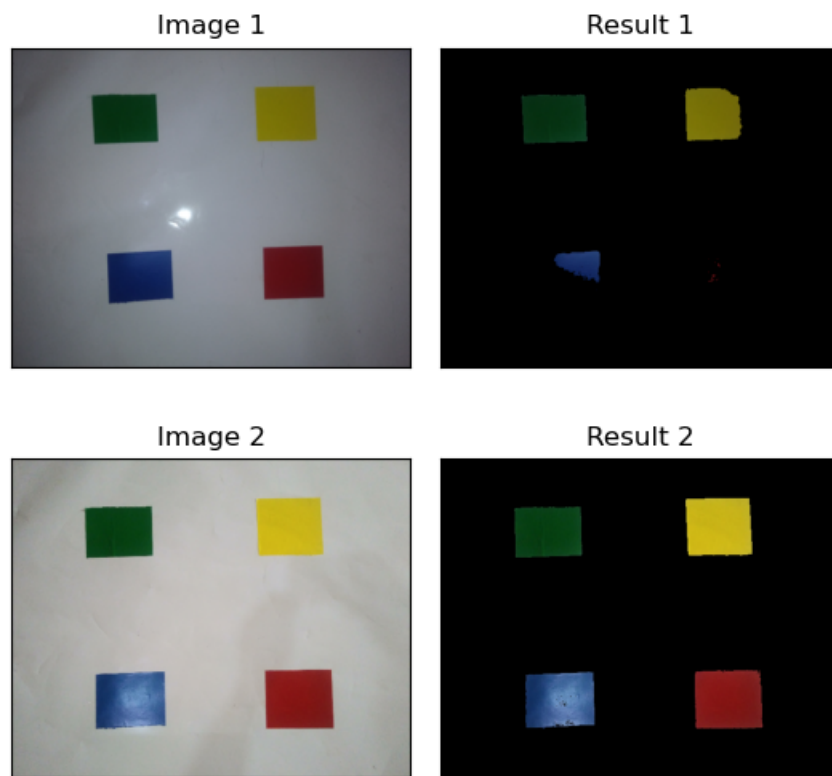


Figure V.11: detection of different colors in the YCrCb color space.

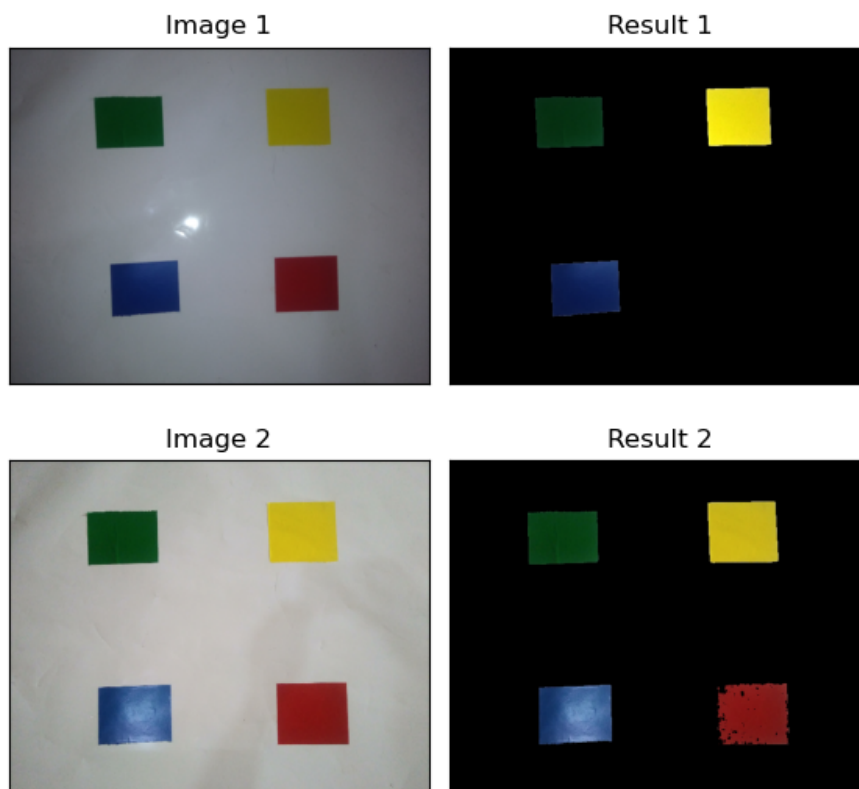


Figure V.12: detection of different colors in the HSV color space.

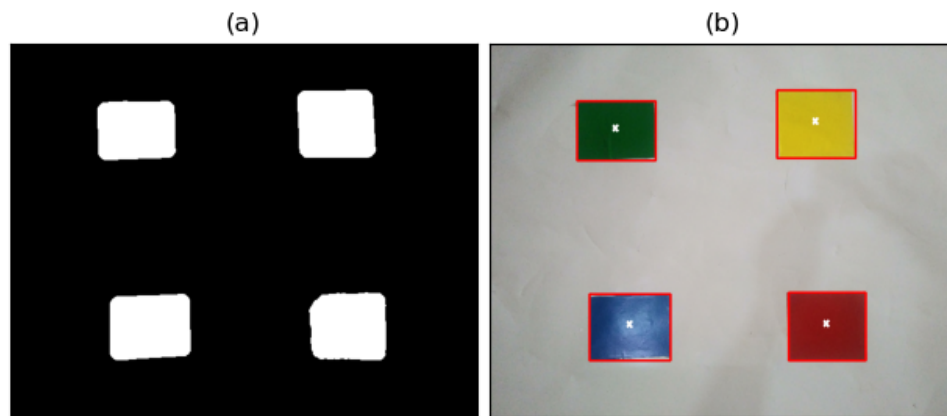


Figure V.13: Find contours and draw a red rectangle around the objects.

V.5 Path planning for color object following

The robot localization and path planning is the final step in this project, Object tracking means estimating the state of the target object present in the scene of our robot, which means that the mobile robot follows a specific object and set its direction according to that object location. After the detection of the desired Object by using color feature, now we must contour coordination which used for the robot navigation. To keep track the color object using Object tracking algorithm (see Figure V.14), it repeats the following steps:

- Search for the object with a specific color.
- Find the object location and calculate the centre of that object.
- set the object centre the same as the image width centre.
- Move forward the object and keep always the centre of the abject the same as the centre of the image width.
- Stop in front of the object with a specific distance.

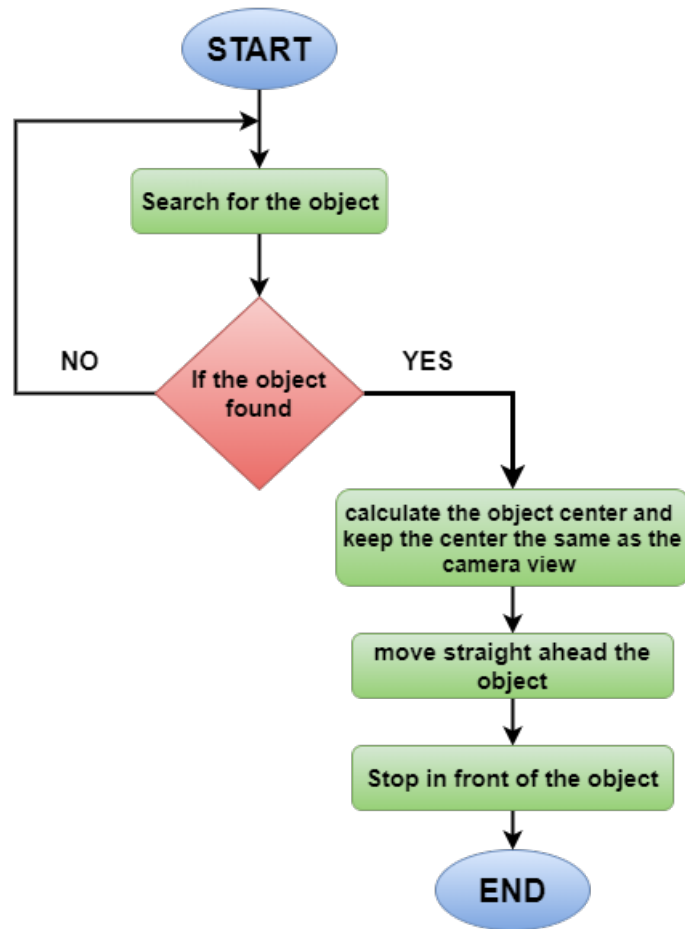


Figure V.14: color object following algorithm.

V.6 Conclusion

in this chapter, we have presented a color reorganization and detection with Raspberry Pi using image processing technique. Several image processing techniques are used to detect the location of the object being tracked in the images.

As seen in this chapter image color segmentation have been described, however the results that obtained in color detection using different color spaces doesn't the same, and due to the change of illumination in the environment the results can be affected.

The robot is designed to track color objects by path planning algorithm to keep the object in sight and driving forward and backward to maintain a constant distance between the robot and the object.

This idea can be used for the surveillance, traffic monitoring, human-computer interaction, intelligent transportation, and robotics and also in the field of medical.

GENERAL CONCLUSION

Autonomous Robots became an essential part of modern life, this field is growing fast and many researchers are focusing on designing a fully autonomous mobile robot that can interact with the environment in real-time, the use of vision system in mobile robots led this field to the next stage of autonomous vision based mobile robots. This work addresses the use of color object detection to autonomously drive a robot from its camera observation.

In this project we designed a prototype of a color object tracker via Raspberry Pi board, using image processing technique to identify and track objects by their colors, we showed a satisfactory result, otherwise, we faced some constraints, the issue we observed that achieving real-time processing which was a big challenge due to some reasons, the low-quality camera used in the project which affects the result of color identification. In a result of that the processing time significantly depends on the performance of the camera and the raspberry pi, in addition to that, the other constraint was the changing of illumination conditions in the environment, which also affects the robot performance, therefore we had to run the robot in an environment where it's possible to control the illumination conditions. In this project we used 2D vision system with a camera that gives a 2D image, however recently robots vision uses a 3D vision system that gives a 3D image which provides depth information thus will give a better result, furthermore, advanced autonomous mobile robots equipped with a powerful vision system with Artificial Intelligence algorithms which gives better interpretation and more efficient autonomous performance.

Despite the complexity of the image processing technique, we showed that the embedded computing platforms such as Raspberry Pi 3 Model B+, are powerful enough to carry out real-time computer vision processing. As can be seen, there are many areas it can push this project further for more advanced applications, this project can be equipped with Artificial Intelligence algorithms and developed as an autonomous vehicle capable of identifying objects, traffic signs and traffic light.

Bibliography

- [1] GREGOR KLANČAR and ANDREJ ZDEŠAR and SAŠO BLAŽIČ and IGOR ŠKRJANC, *WHEELED MOBILE ROBOTICS*, Elsevier 2017.
- [2] Spyros G. Tzafestas, *Introduction To Mobile Robot Control*, Elsevier 2014
- [3] Rafael C.Gonzalez and Richard E. Woods, *Digital image processing 3rd edition*, Pearson, 2018
- [4] mcfarlane, m. d. [1972], *Digital pictures fifty years ago*, *Proceedings of the IEEE*, vol 60, NO 7.
- [5] Milan Sonka , Vaclav Hlavac , Roger Boyle ,*image processing analysis, and machine vision*, Thomson 2007.
- [6] P.K.Sinha, *image Acquisition and Preprocessing for machine vision systems*, Spie Press 2012.
- [7] Bern jahne, *Digital Image Processing 6th revised and extended edition*, Springer 2005.
- [8] Simon Monk, *Raspberry Pi Cook Book, 1st Edition*, O'Reilly Media 2014.
- [9] Vladmimir Vujovic and Mirjana Maksimovic, *Raspberry Pi as a Wireless Sensor Node*, Faculty of Electrical Engineering, East Sarajevo, Bosnia and Herzegovina, 26-30 May 2014.
- [10] Rocío Andrea Rodríguez , Pablo Cammarano , Daniel Alberto Giulianelli, Pablo Martín Vera , Artemisa Trigueros , and Luciano J. Albornoz, *Using Raspberry Pi to Create a Solution for Accessing Educative Questionnaires From Mobile Devices*, *IEEE REVISTA IBEROAMERICANA DE TECNOLOGIAS DEL APRENDIZAJE*, VOL. 13, NO. 4, NOVEMBER 2018.
- [11] Eben Upton and Gareth Halfacree, *Raspberry PI User Guide 4Th Edition*, Wiley 2016.

BIBLIOGRAPHY

- [12] Raspberry Pi 3 Model B+ Datasheet: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [13] Raspberry Pi Camera rev 1.3 Datasheet : <https://www.raspberrypi.org/documentation/hardware/camera/>
- [14] John Boxall, Arduino workshop, No Starch Press 2013.
- [15] Yusuf Abdullahi Badamasi, The Working Principle Of An Arduino, 29 December 2014, IEEE.
- [16] Arduino Nano Datasheet: <https://store.arduino.cc/arduino-nano>
- [17] Austin Hughes, Electric Motors and Drives: Fundamentals,Types and Applications 3rd Edition, ELsevier 2006.
- [18] DC Gearbox Motor - "TT Motor" - 200RPM - Datasheet.
- [19] Saleem Hashmi, Gilmar Ferreira Batalha,Bekir Yilbas, Comprehensive Materials Processing, ELsevier 2014.
- [20] Sharp IR GP2Y0A41SK0F distance sensor Datasheet: <https://www.alldatasheet.com/view.jsp?Searchword=GP2Y0A41SK0F&sField=4>
- [21] Mark Lutz, Python Pocket Reference, 2nd Edition, O'Reilly Media; (November 17, 2001).
- [22] Aditya Tandon, Sonam Soni, Introduction to Artificial Intelligence using Python, Book Bazooka 2020
- [23] Gloria Bueno García,Oscar Deniz Suarez ,José Luis Espinosa Aranda ,Jesus Salido Tercero ,Ismael Serrano Gracia , Learning Image Processing with OpenCV, Packt Publishing (March 26, 2015).
- [24] Jeff Cicolani, Beginning Robotics with Raspberry Pi and Arduino, Apress 2018
- [25] Michael Barr, Introduction to Pulse Width Modulation (PWM), Barr Group (1 September 2001).
- [26] Khamar Basha Shaik, Ganesan, V.Kalist , B.S.Sathish, J.Merlin Mary Jenitha. Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).
- [27] Vikas Gupta, Color spaces in OpenCV (C++ / Python), May 07 2017: <https://www.learnopencv.com/color-spaces-in-opencv-cpp-python/>

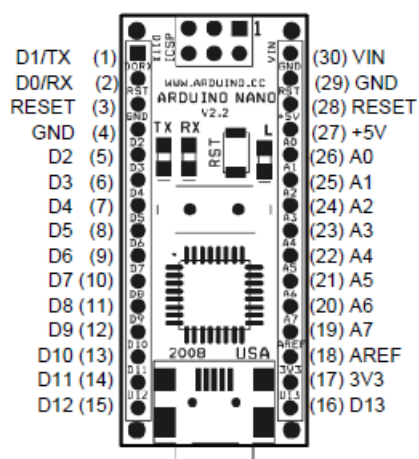
- [28] Amit Kumar Mandal, Dilip Kumar Baruah, Image Segmentation Using Local Thresholding And Ycbr Color Space, Journal of Engineering Research and Applications, Vol 3, Issue 6, Nov-Dec 2013.

Annexes

Product Name	Raspberry Pi Camera Module
Product Description	High definition camera module compatible with the Raspberry Pi model A and model B. Provides high sensitivity, low crosstalk and low noise image capture in an ultra small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor.
RS Part Number	775-7731
Specifications	
Image Sensor	Omnivision 5647 CMOS image sensor in a fixed-focus module with integral IR filter
Resolution	5-megapixel
Still picture resolution	2592 x 1944
Max image transfer rate	1080p: 30fps (encode and decode) 720p: 60fps
Connection to Raspberry Pi	15 Pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2)
Image control functions	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
Temp range	Operating: -30° to 70° Stable image: 0° to 50°
Lens size	1/4"
Dimensions	20 x 25 x 10mm
Weight	3g

Figure V.15: Technical data sheet of Raspberry Pi camera v1.3 .

Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Figure V.16: Arduino Nano Pin Layout.

MX1508 SOP-16 Quad Dual Brushed DC Motor Drive IC

Overview

This product provides an integrated brushed DC motor drive solution for battery powered toy, low voltage or battery powered motion control applications.

The circuit integrates two H-channel drive circuits designed with N-channel and P-channel power MOSFETs, which are suitable for driving the steering of electric toy cars.

Wheel and rear wheel drive, (drive brushed DC motor or drive two windings of stepper motor). The circuit has a wide operating voltage range (from

2V to 9.6V), the steering wheel has a maximum continuous output current of 0.8A and a maximum peak output current of 1.5A. Rear wheel drive maximum continuous output

The flow reaches 1.5A and the maximum peak output current reaches 2A.

The drive circuit has a built-in thermal protection circuit. When the load current through the drive circuit is much larger than the maximum continuous current of the circuit, the package is capable of dissipating heat.

Limit, the junction temperature of the chip inside the circuit will rise rapidly. Once the set value is exceeded (typically 150 ° C), the internal circuit will immediately turn off the output work.

Rate tube, cut off the load current, and avoid the safety hazard such as smoke and fire in the plastic package due to the continuous increase of temperature. Built-in temperature hysteresis circuit, indeed

It is only after the circuit is restored to a safe temperature that the circuit is allowed to be re-controlled.

.

Features:

- a. Low standby current (less than 0.1uA);
- b. Low quiescent operating current;
- c. Integrated H-bridge driver circuit;
- d. Built-in anti-common conduction circuit;
- e. Low-conduction internal resistance power MOSFET;
- f. Built-in thermal protection circuit (TSD) with hysteresis effect;
- g. Antistatic rating: 3KV (HBM).

Typical Applications

- a. 2-6 AA/AAA dry battery powered toy motor drives;
- b. 2-6 nickel-hydrogen/nickel-cadmium rechargeable battery powered toy motor drives;
- c. 1-2 lithium battery powered motors drive

Figure V.17: Technical data sheet of the MX1508 DC motor drive .



■Description

GP2Y0A41SK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IR-LED (infrared emitting diode) and signal processing circuit. The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

■Applications

1. Cleaning robot
2. Personal robot
3. Sanitary

■Features

1. Distance measuring sensor is united with PSD, infrared LED and signal processing circuit
2. Short measuring cycle (16.5ms)
3. Distance measuring range : 4 to 30 cm
4. Package size (29.5 × 13.0 × 13.5mm)
5. Analog output type

Figure V.18: Technical data sheet of GP2Y0A41SK0F

Abstract:

Over recent years, integration of visual sensing system in mobile robots plays a vital role in autonomous navigation, visual sensing provides mobile robots with the ability to acquire visual information of the surrounding environment. Real-time object detection is challenging due to do a change of appearance and illumination. In this work, a prototype of vision-based object follower has been presented, the robot designed to autonomously follow objects, it can identify the desired object by its colors and determine its position using computer vision algorithms, by processing a frame sequence feed by a camera mounted in front of the robot in real-time.

Keywords: color detection, object follower, autonomous mobile robot, computer vision, robot vision, OpenCV, Raspberry Pi.

ملخص:

في السنوات الأخيرة، أصبح دمج الإستشعار البصري و تزويد الروبوتات المتنقلة بالرؤية يلعب دورا هاما في التنقل التلقائي للروبوت. فخاصية الرؤية تزود الروبوتات المتنقلة بالقدرة على جمع و إستخلاص المعلومات من المحيط الخارجي للروبوت من أجل تمييز الأشياء. تقنية التعرف و تمييز الأشياء مصحوبة بمجموعة من التحديات وذلك بسبب التغيير المستمر لمظهر الهدف و الإضاءة الغير ثابتة، في هذا السياق فإن الهدف من هذه الدراسة هو إعداد نموذج لروبوت يتعقب الأشياء بتلقائية تامة، يقوم هذا الروبوت بالتعرف على الأشياء و تحديد موقعها بإستخدام الألوان المميزة للهدف بتطبيق خوارزميات الرؤية الحاسوبية وذلك بالمعالجة في الوقت الفعلي لصور متعاقبة و التي تبث من كاميرا مثبتة في مقدمة الروبوت.

الكلمات المفتاحية: التعرف على الألوان، تتبع الأشياء، الروبوت التلقائي المتنقل، الرؤية الحاسوبية، رؤية الروبوت، راسبيري باي، أوبن سي في.

Résumé:

Au cours des dernières années, l'intégration du système de détection visuelle dans les robots mobiles joue un rôle important dans la navigation autonome, la détection visuelle permet aux robots mobiles d'acquérir des informations visuelles sur l'environnement environnant, La détection d'objets en temps réel est difficile en raison d'un changement d'apparence et d'éclairage. Dans cette thèse, un prototype de suiveur d'objet basé sur la vision a été présenté, le robot conçu pour suivre les objets en manière autonome, il peut identifier l'objet souhaité par ses couleurs et déterminer sa position à l'aide d'algorithmes de vision par ordinateur, en traitant en temps réel une séquence d'images par une caméra montée devant le robot.

les mots clés: détection de couleur, suiveur d'objet, robot mobile autonome, vision par ordinateur, vision de robot, OpenCV, Raspberry Pi.