الآية

فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ ۗ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَىٰ إِلَيْكَ وَحْيُهُ ۖ وَقُلْ رَبِّ زِدْنِي عِلْمًا ﴿١١٤﴾

سورة طه (١١٤)

Dedication

All praise to Allah, today we fold the days ' tiredness and the errand summing up between the cover of this humble work.

To the utmost knowledge lighthouse, to our greatest and most honored prophet Mohamed - May peace and grace from Allah be upon him.

To the spring that never stops giving, to my mother who weaves my happiness with strings from her merciful heart . . . .

To my mother

To whom he strives to bless comfort and welfare and never stints what he owns to push me in the success way who taught me to promote life stairs wisely and patiently. . . .

To my dearest father

To whose love flows in my veins and my heart always remembers them. . . .

To my brothers and sisters

To those who taught us letters of gold and words of jewel of the utmost and sweetest sentences in the whole knowledge. Who reworded to us their knowledge simply and from their thoughts made a lighthouse guides us through the knowledge and success path. . . .

To our honored teachers and professors

Acknowledgment

We give all our regards and wishes to all the
Academic supervisors and the employees at
Sudan University of science and technology
whom encouraged and supported.

Also We are heartily thankful to our supervisor A. *Gaffar babiker*
for his guidance and support from the initial to the final level which enabled us to complete this project.

To Eng. Waddah Mohammed thank you we highly appreciate every minute you spend with us.

Abstract

The research is designed and implemented, it can be used in a variety of areas and uses for various purposes, which is controlled remotely about 100 meters away. This is useful for espionage and natural disasters and helps keep the controlling person away from danger or out of sight in case of espionage. The robot has several DC motor to move from one place to another and several sensors collect the data and send it to the microcontroller (Arduino). There is a base-mounted camera with two servo motors to control the direction of the camera with 180 degree rotational motion and 90 degrees vertical. The camera and ultrasonic sensors are used to provide the necessary data from the surrounding environment of the robot to Robot in automatic condition, manual condition control by press button on application. The car along together with camera can wirelessly transfer video in real time. The car is able to reach a particular destination smoothly and accurately. The main objective of the research is to study different types of robot and how this robots operate in observation and exploration.

مستخلص

صــمم ونفذ المشــروع ، ويمكن اســتخدامه في مجموعة متنوعة من المجالات واســتخدامات لأغراض متنوعة، والتي يتم التحكم فيها عن بعد على بعد حوالي 100 متر .هذا مفيد في التجسس والكوارث الطبيعية ويساعد على إبقاء الشخص المسيطر بعيداً عن الخطر أو بعيداً عن الأنظار في حالة التجسس .لدى الروبوت عدة محركات تيار مستمر للانتقال من مكان إلى آخر، وتقوم العديد من أجهزة الاستشعار بجمع البيانات وإرسالها إلى وحدة التحكم الدقيقة (اردوينو .)توجد كاميرا مثبتة في القاعدة مع محركين مؤازرين للتحكم في اتجاه الكاميرا بحركة دائرية 180 درجة و 90 درجة عمودية .يتم استخدام الكاميرا وأجهزة الاستشعار بالموجات فوق الصوتية لتوفير البيانات اللازمة من البيئة المحيطة للروبوت إلى الروبوت في حالة تلقائية، والتحكم في الحالة اليدوية عن طريق الضغط على زر التطبيق .يمكن للسيارة جنبا إلى جنب مع الكاميرا نقل الفيديو لاسلكيا في الوقت الحقيقي .السيارة قادرة على الوصول إلى وجهة معينة بسلاسة وبدقة. الهدف الأساسي من البحث هو دراسة أنواع مختلفه من الروبوتات وكيفية استخدامها للمراقبه والاستكشاف.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	**Alternating Current**
App	**Application**
DC	**Direct Current**
GPS	**Global Positioning System**
GSM	**Global System for Mobile**
HD	**High Definition**
HDMI	**High Definition Multimedia Interface**
I/O	**Input/ Output**
IDE	**Integrated Development Environment**
LED	**Light Emitting Diode**
Modem	**Modulator demodulator**
OS	**Operating System**
PC	**Personal Computer**
PWM	**Pulse Width Modulation**
RF	**Radio Frequency**
RX	**Receive**
TX	**Transmit**
Wi-Fi	**Wireless Fidelity**

CHAPTER ONE
INTRODUCTION

1. Over view

Surveillance is the process of monitoring a situation, an area or a person [2],[3]. This generally occurs in a military scenario where surveillance of borderlines and enemy territory is essential to a country's safety. Human surveillance is achieved by deploying personnel near sensitive areas in order to constantly monitor for changes. However, humans do have their limitations, and deployment in inaccessible places is not always possible. There are also added risks of losing personnel in the event of getting caught by the enemy. With advances in technology over the years, however, it is possibly to remotely monitor areas of importance by using robots in place of humans[7],[8], building a small robot for testing and research purposes proves to be extremely expensive. because a security robot would require certain components such as a GPS module, High resolution cameras[3],[4],[5], Each of these components are quite expensive and piecing them together for the purpose of a robot is a very costly and time consuming affair, thus used the smartphone with the required features such as a GPS module, a high resolution camera and internet connectivity.

2. Research Problem

When Humans exploring places some problems may be encountered, these are:

- In exploring wildlife, explorers have some problems and risks, including exposure to predators and animal fear when feeling explorers.
- In the exploration of the radioactive places, dilapidated buildings, places of waste and the bastions of criminals, humans may be exposed to danger.

3. Objectives

The main objective of the research to design a robot to keep humans away from the bastions of criminals and the dangers they may encounter in the strongholds of criminals and provide a comprehensive picture of strongholds to look at the smallest victims and locate the exact victims in dilapidated buildings. Wildlife conservation and non-disturbance by wildlife recorders with the ability to take pictures of predators safely.

4. Research Methodology

Modelling a robotic system using a microcontroller (Arduino Uno), Power shield driver (L298N), WI-FI module, camera, Ultrasonic sensor HC-sr04, four Dc motors, and two servomotors.

5. Research structure

The research is composed from an abstract and five chapters.

Chapter one is an introduction deals with over view, research problem, objectives, research methodology and research structure. Chapter two is robotic systems deals with introduction and main type of robots. Chapter three is robotic software deals with an introduction, Arduino IDE, Android (operation system), Proteus. Chapter four is an application consist of introduction, system flowchart, the system block diagram, microcontroller (Arduino Uno), motor shield drive (L298N), WI-FI module, camera, DC motor, Servo motor, Ultrasonic sensor (HC-sr04),System operation. Chapter five is conclusion and recommendation.

CHAPTER TWO
ROBOTIC SYSTEMS

2. 1 Introduction

A robot is a machine designed to execute one or more tasks repeatedly, with speed and precision [18]. There are as many different types of robots as there are tasks for them to perform.

A robot is a mechanical or virtual artificial agent, usually an electro-mechanical machine that is guided by a computer program or electronic circuitry. Robots can be autonomous or semiautonomous. Robots have replaced human in performing repetitive and dangerous tasks which humans prefer not to do, or are unable to do because of size limitations, or which take place in extreme environments such as outer space or the bottom of the sea.

The advent of new high-speed technology and the growing computer Capacity provided realistic opportunity for new robot controls and realization of new methods of control state.

This technical improvement together with the need for high performance robots created faster, more accurate and more intelligent robots using new robots control devices[1], In recent years, the applications of mobile robot have gradually become more diverse, which makes the robot closer to people's daily life. At present, the middle and small-scale motion robot are usually designed based on single chip microcomputer without operating system. This is why we are doing a study of the surrounding environment with simple research to see the world's need for robots and their importance and needs in daily and practical life, therefore need robots with the advent of the age to reduce the risks against human life.

2. 2 Main type of robots

There are many types of robots used for various purpose [19] such as:

- Articulated Robots

The figure (2.1) represent articulated robot:

Figure 2.1 Articulated Robot

An articulated robot is the robot equipped with joins capable of rotating (for example: an industrial robot or a robot with legs)

Flexibility, dexterity, and reach make articulated robots ideally suited for tasks that span non-parallel planes, such as machine tending. Articulated robots can also easily reach into a machine tool compartment and under obstructions to gain access to a work piece (or even around an obstruction, in the case of a 7-axis robot).

Sealed joints and protective sleeves allow articulated robots to excel in clean and dirty environments alike. The potential for mounting an articulated robot on any surface (e.g., a ceiling, a sliding rail) accommodates a wide range of working options.

The sophistication of an articulated robot comes with a higher cost compared to other robot types with similar payloads. And articulated robots are less suited than other types of robots for very high-speed applications due to their more complex kinematics and relatively higher component mass.

Advantages

- High speed
- Large work envelope for least floor space
- Easier to align to multiple planes

Disadvantages

- Requires dedicated robot controller
- Complicated programming
- Complicated kinematics

Application

- Food packaging
- Arc welding
- Spot welding
- Material handling
- Machine tending
- Automotive assembly
- Steel bridge manufacturing
- Steel cutting
- Glass handling

- Selective Compliance Articulated Robot Arm (SCARA)

Figure (2.2) represent SCARA robot:

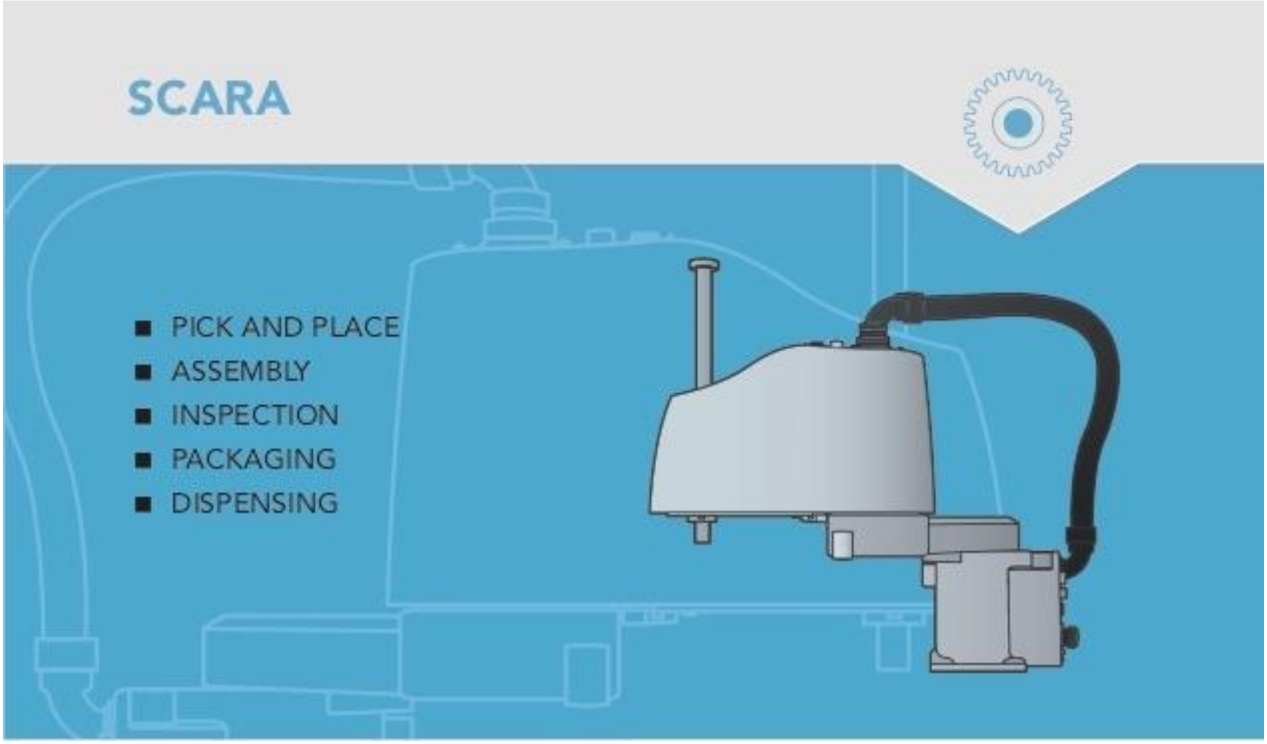

Figure 2.2 SCARA robot

A Selective Compliance Articulated Robot Arm (SCARA) is a good and cost-effective choice for performing operations between two parallel planes (e.g., transferring parts from a tray to a conveyor). SCARA robots excel at vertical assembly tasks such as inserting pins without binding due to their vertical rigidity.

SCARA robots are lightweight and have small footprints, making them ideal for applications in crowded spaces. They are also capable of very fast cycle times.

Due to their fixed swing arm design, which is an advantage in certain applications, SCARA robots face limitations when it comes to tasks that require working around or reaching inside objects such as fixtures, jigs, or machine tools within a work.

Advantages

- High speed
- Excellent repeatability
- Large workspace

Disadvantages

- Requires dedicated robot controller
- Limited to planar surfaces
- Hard to program offline

Application

- Assembly applications
- Semiconductor wafers handling
- Biomed applications
- Packaging
- Palletizing
- Machine loading

- Delta robots

Figure (2.3) represent Delta robot

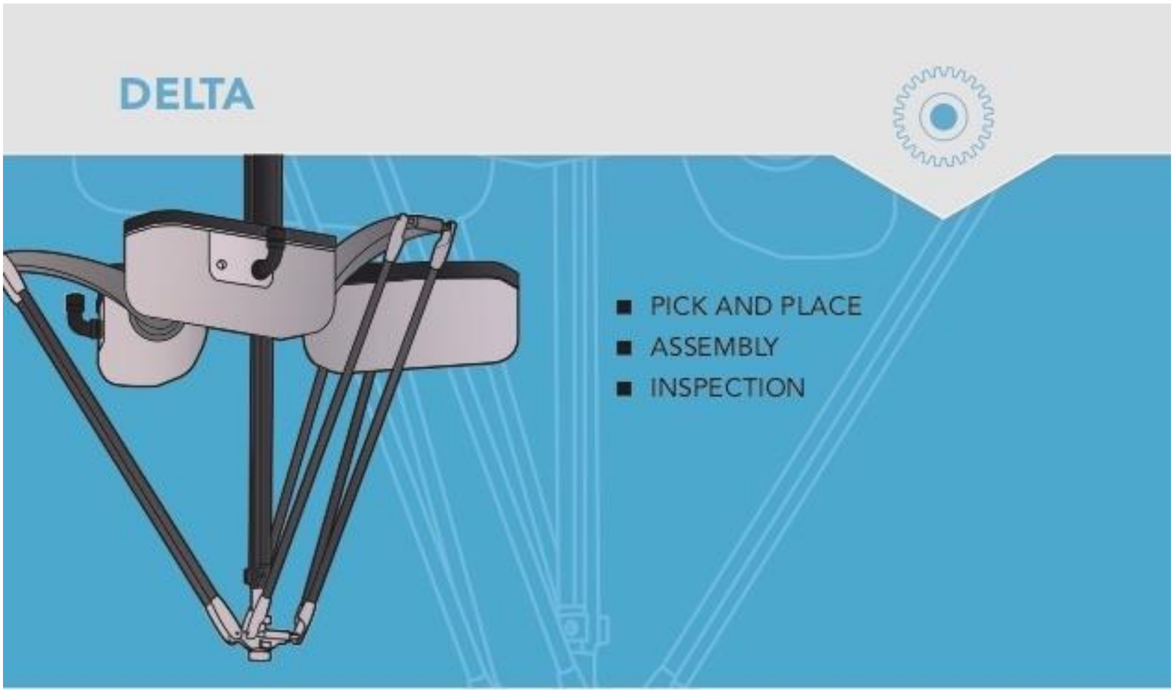

Figure 2.3 Delta robot

Delta robots, also referred to as “spider robots,” use three base-mounted motors to actuate control arms that position the wrist. Basic delta robots are 3-axis units but 4- and 6-axis models are also available.

By mounting the actuators on, or very close to, the stationary base instead of at each joint (as in the case of an articulated robot), a delta robot’s arm can be very lightweight. This allows for rapid movement which makes delta robots ideal for very high-speed operations involving light loads.

An important thing to note as you compare delta robots to other robot types: Reach for delta robots is typically defined by the diameter of the working range, as opposed to the radius from the base, as in the case of articulated and SCARA units. For example, a delta robot with a 40” reach would only have half the reach (20” on a radius) of a 40” articulated or SCARA unit.

Advantages

- Very high speed
- High operational accuracy

Disadvantages

- Complicated operation
- Requires dedicated robot controller

Application

- Food industry
- Pharmaceutical industry
- Electronic industry
- Flight simulators
- Automobile simulators
- Optical fiber alignment

- Cartesian Robots

Figure (2.4) represent Cartesian Robots

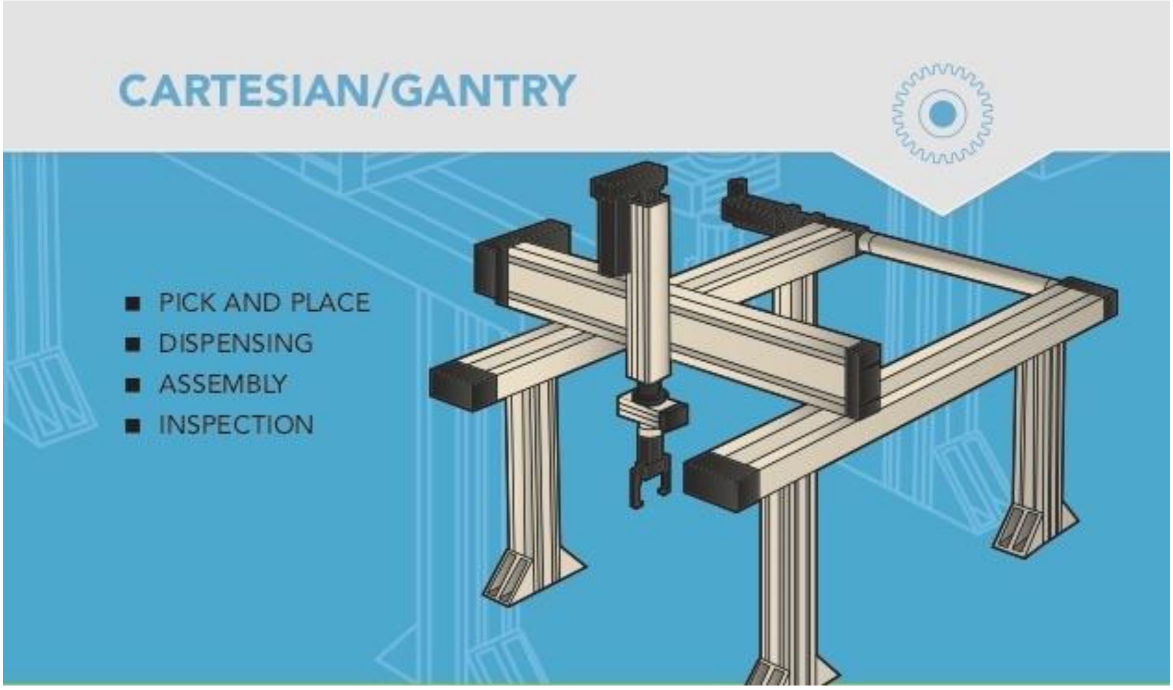

Figure 2. 4 Cartesian Robot

Cartesian robots typically consist of three or more linear actuators assembled to fit a particular application. Positioned above a workspace, cartesian robots can be elevated to maximize floor space and accommodate a wide range of workpiece sizes. (When placed on an elevated structure suspended over two parallel rails, cartesian robots are referred to as "gantry robots.")

Cartesian robots typically use standard linear actuators and mounting brackets, minimizing the cost and complexity of any "custom" cartesian system. Higher capacity units can also be integrated with other robots (such as articulated robots) as "end- effectors" to increase system capabilities. That said, the custom nature of cartesian robots can make design, specification, and programming challenging or out of reach for smaller manufacturers intent on a "DIY" approach to robotics implementation.

Advantages

- Provides high positional accuracy
- Simple operation
- Easy to program offline
- Highly customizable
- Can handle heavy loads
- Less cost

Disadvantages

- Requires large operational and installation area
- Complex assembly
- Movement limited to only one direction at a time

Application

- Pick and place operations
- Loading and unloading
- Material handling
- Assembly and sub-assembly
- Nuclear material handling
- Adhesive application

▪ Industrial robot

An industrial robot is a robot system used for manufacturing. Industrial robots are automated, programmable and capable of movement on three or more axes.

Typical applications of robots include welding, painting, assembly, disassembly, pick and place for printed circuit boards, packaging and labeling, palletizing, product inspection, and testing; all accomplished with high endurance, speed, and precision. They can assist in material handling.

In the year 2020, an estimated 1.64 million industrial robots were in operation worldwide according to International Federation of Robotics (IFR).

Figure (2.5) represents Industrial Robot:

Figure 2.5 Industrial Robot

CHAPTER THREE
ROBOTIC SOFTWARE

3. 1 Introduction

Robot software is the set of coded commands or instructions that tell a mechanical device and electronic system, known together as a robot, what tasks to perform. Robot software is used to perform autonomous tasks [20]. Many software systems and frameworks have been proposed to make programming robots easier.

Some robot software aims at developing intelligent mechanical devices. Common tasks include feedback loops, control, path finding, data filtering, locating and sharing data.

There are many robot software used to control robotic system such as: Arduino ide, Android (operation system), Proteus.

3. 2 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, mac OS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Figure (3.1) represents Arduino Logo:

Figure3.1 Arduino IDE Logo

Figure (3.2) represents the Arduino IDE home page:

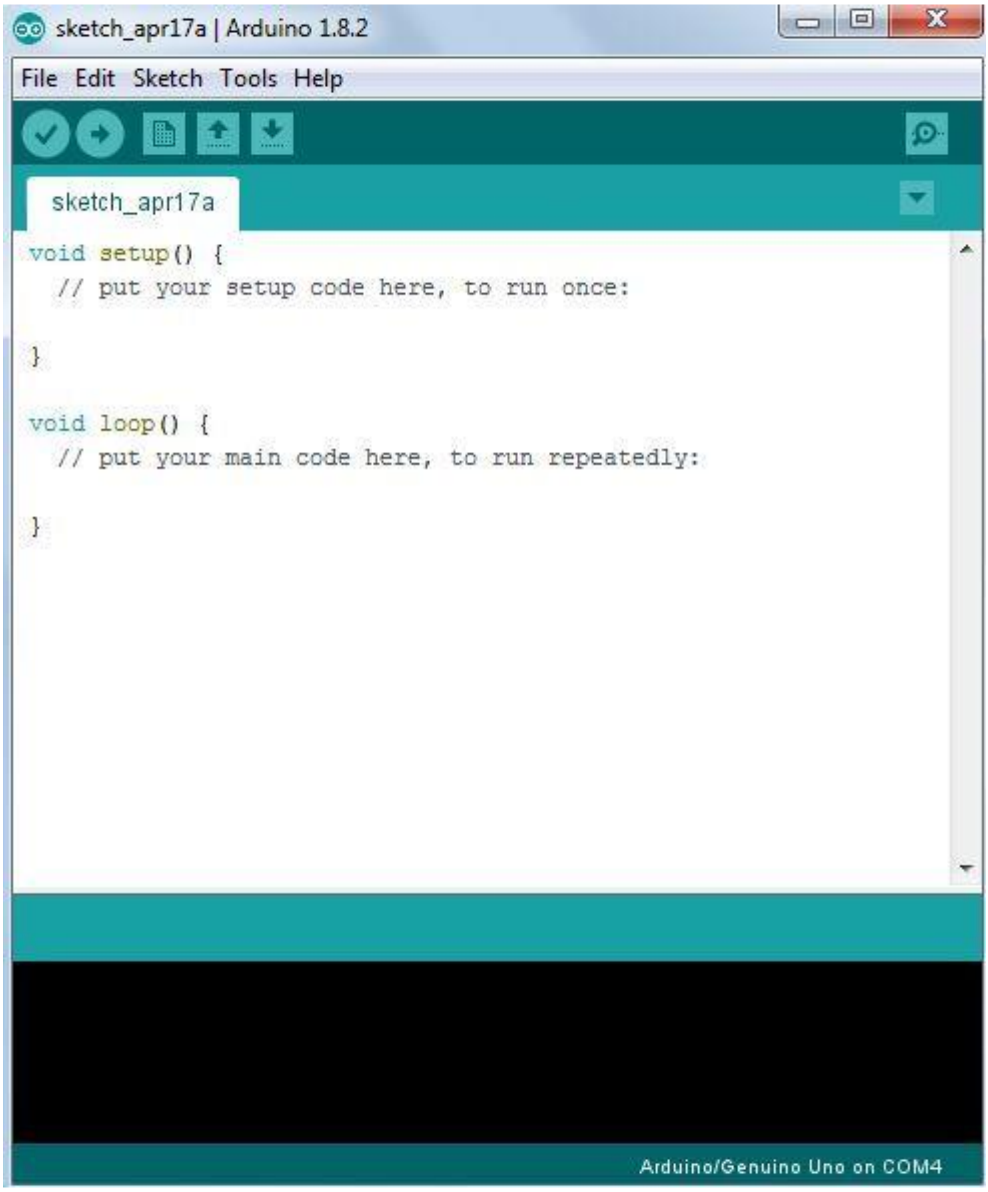

Figure 3.2 Arduino IDE home page 6

3. 3 **Android (operating system)**

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in game consoles, digital cameras, regular PCs (e.g. the HP Slate 21) and other electronics.

As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded. A developer survey conducted in April “May 2013 found that 71 % of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over one billion active monthly Android users, up from 538 million in June 2013. As of 2015, Android has the largest installed base of all general purpose operating systems.

Android’s source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software, including proprietary software developed and licensed by Google.

Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007,along with the founding of the Open Handset Alliance a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

Figure (3.3) represents the home screen of android app:

Figure 3.3 App home screen 7

Figure (3.4) represents the Code settings of android app:

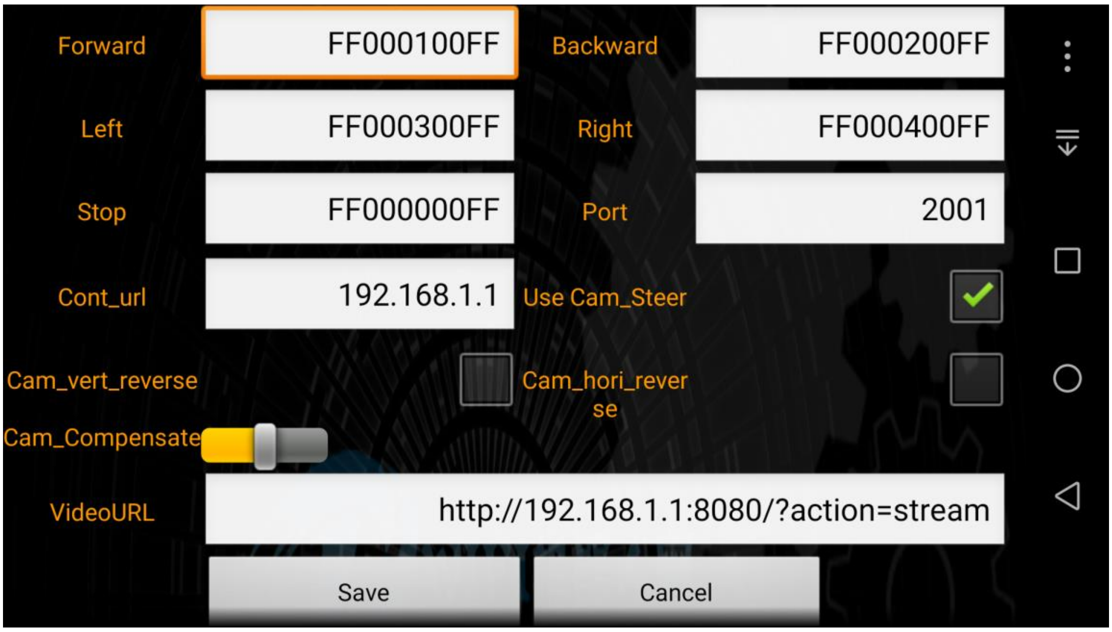

Figure 3.4 Code settings 8

Figure (3.5) represent home control screen of android app:

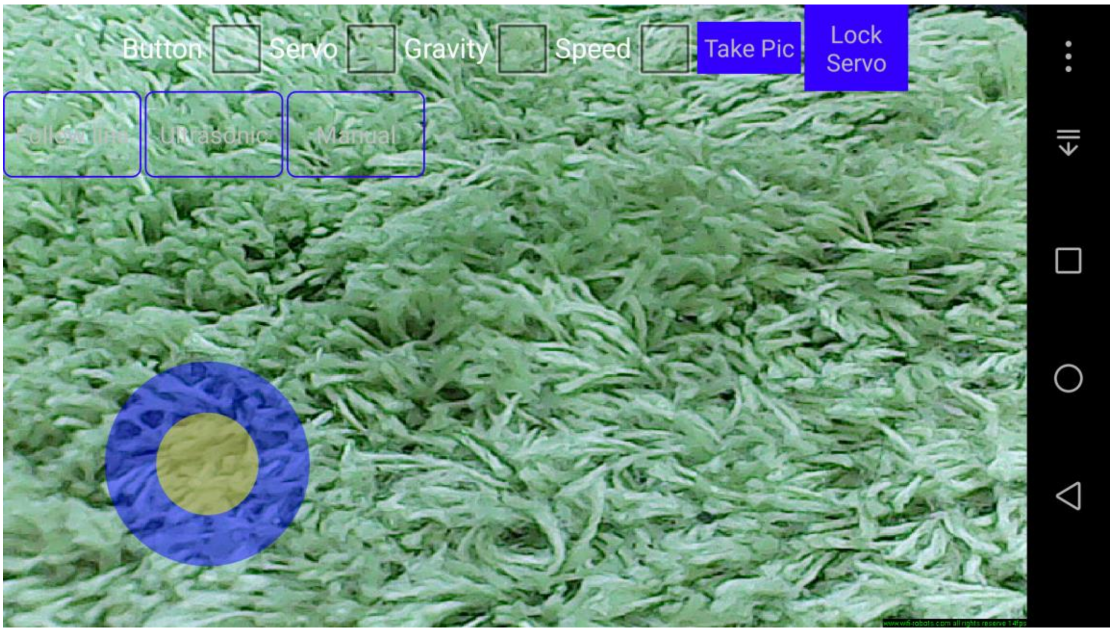

Figure3.9 Home control screen

Figure (3.6) represents the button control of android app:

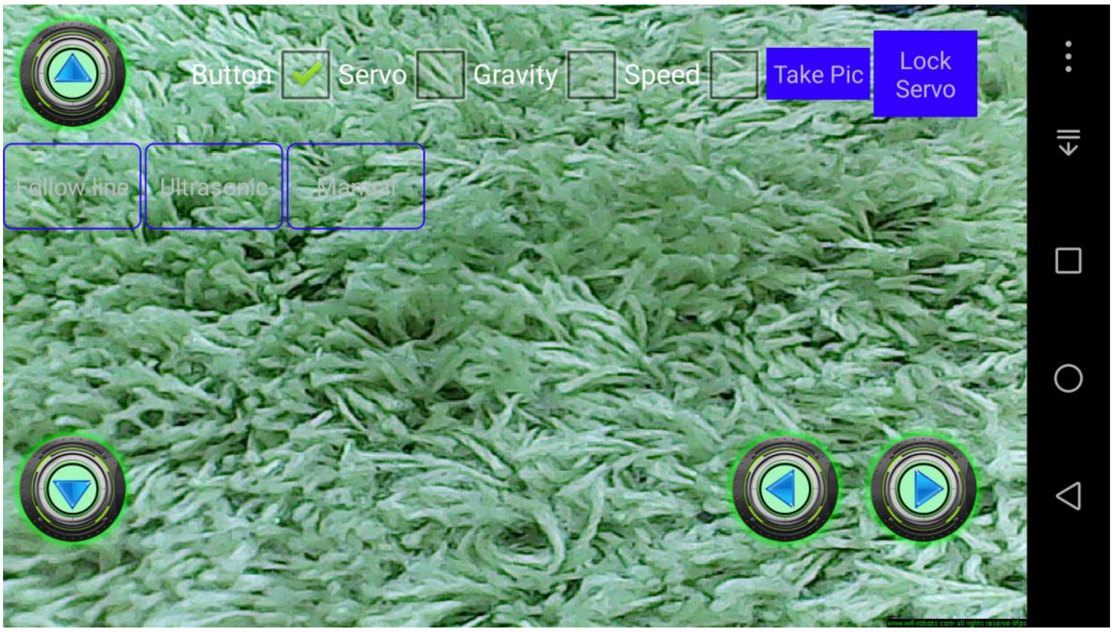

Figure3.10 Button control screen

Figure (3.7) represents the speed control screen of android app:

Figure3.11 Speed control screen

3. 4 Proteus

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

It was developed in Yorkshire, England by Labcenter Electronics Ltd and is available in English, French, Spanish and Chinese languages.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high-speed design was added in 2017. Feature led product releases are typically biannual, while maintenance based service packs are released, as it is required.

Product Modules

The Proteus Design Suite is a Windows application for schematic capture, simulation, and PCB (Printed Circuit Board) layout design. It can be purchased in many configurations, depending on the size of designs being produced and the requirements for microcontroller simulation. All PCB Design products include an auto router and basic mixed mode SPICE simulation capabilities.

The Figure (3.8) represents the proteus home screen

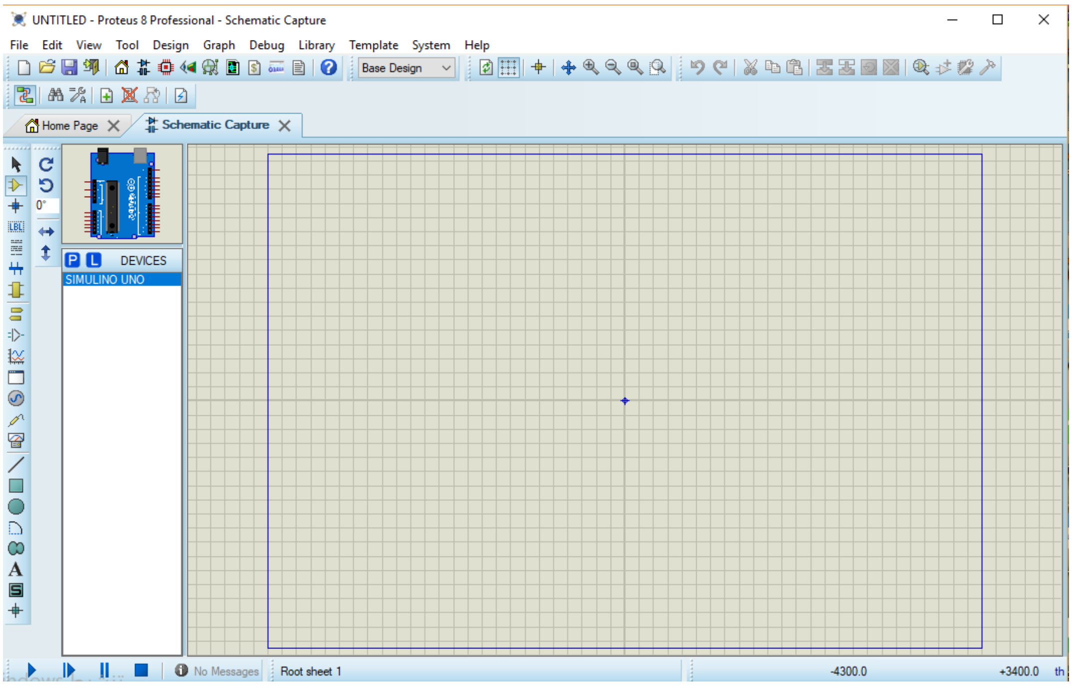

Figure3.12 Proteus home screen

CHAPTER FOUR
APPLICATION

4. 1 Introduction

The observation robotic system is used in many fields such as: Chemical field, Exploring wildlife (explorers have some problems and risks, including exposure to predators and animal fear when feeling explorers), radioactive places, dilapidated buildings, places of waste and the bastions of criminals humans may be exposed to danger.

The robot based Wi-Fi remote control using phone (Android) with a vector for the picture and video to the phone, and consist of: WI-FI module, Camera, four Dc motor, two-servo motor, motor shield drive (L298N), microcontroller (Arduino UNO), ultrasonic sensor (HC-SR04).

4. 2 System Flowchart

The Figure (4.1) represents the system Flowchart

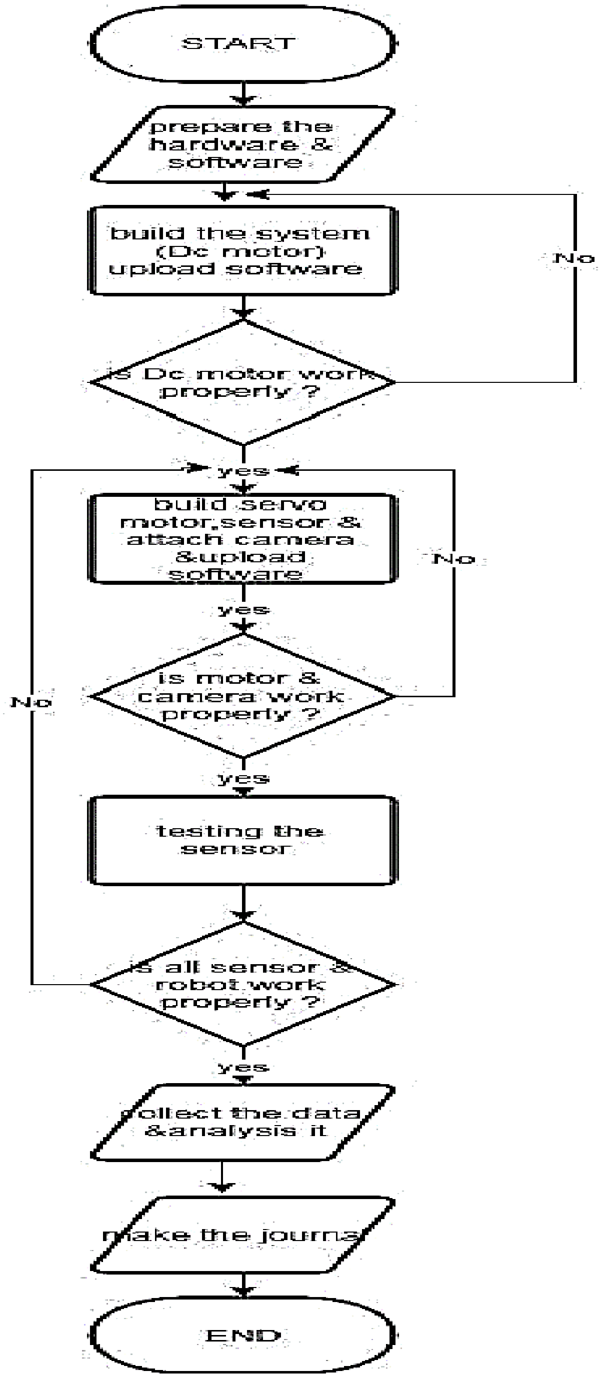

Figure 4.1: Flow Chart

4. 3 The system Block diagram

The Figure (4.2) represents the system block diagram

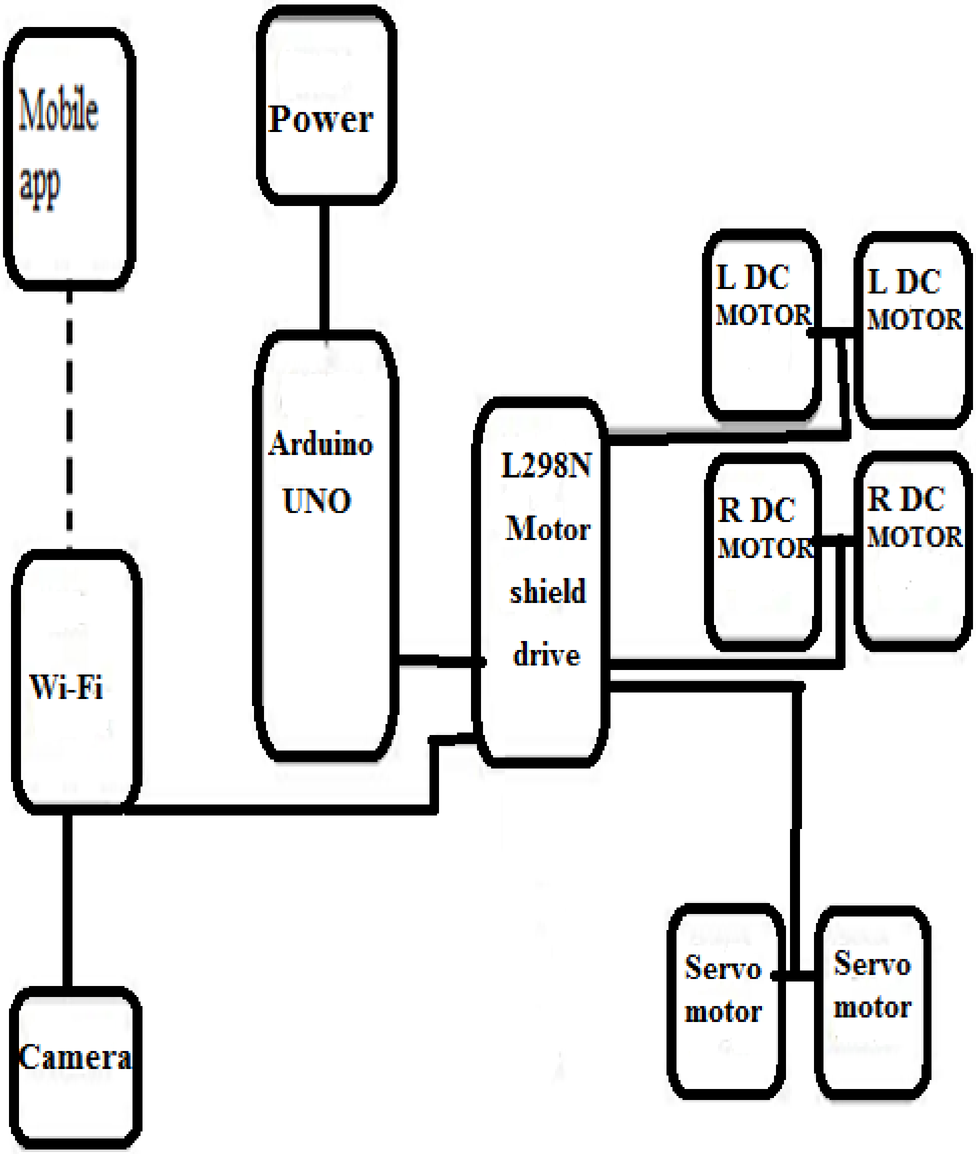

Figure 4.2: Block diagram for camera robot

4.3.1 Microcontroller (Arduino Uno)

Arduino is an electronic board based on Atmega328 IC, Arduino UNO consist 32 pins(input/output), 6 of these can be used analog input (A_0-A_5), 14 used digital output, 6 of these can be used PWM output (3,5,6,9,10and11). It can easily connect with computer and upload the code by USB cable, operating on a 5-volt voltage source.

Arduino can communicate with the surrounding environment by connecting to sensor devices, or by connecting it to motors, small LED and other electronic devices.

Figure (4.3) represents the Arduino Uno R3:

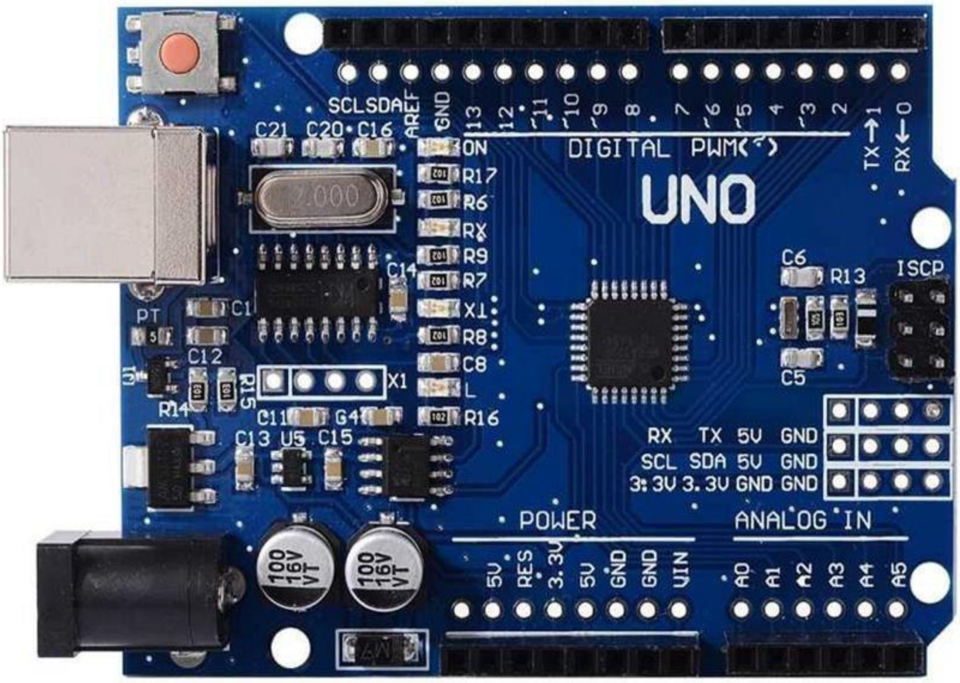

Figure 4.3: Arduino Uno R3

4.3.2 Motor shield drive (L298N)

L298N is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L298N is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L298N.

Figure (4.4) represents L298N Motor shield drive:

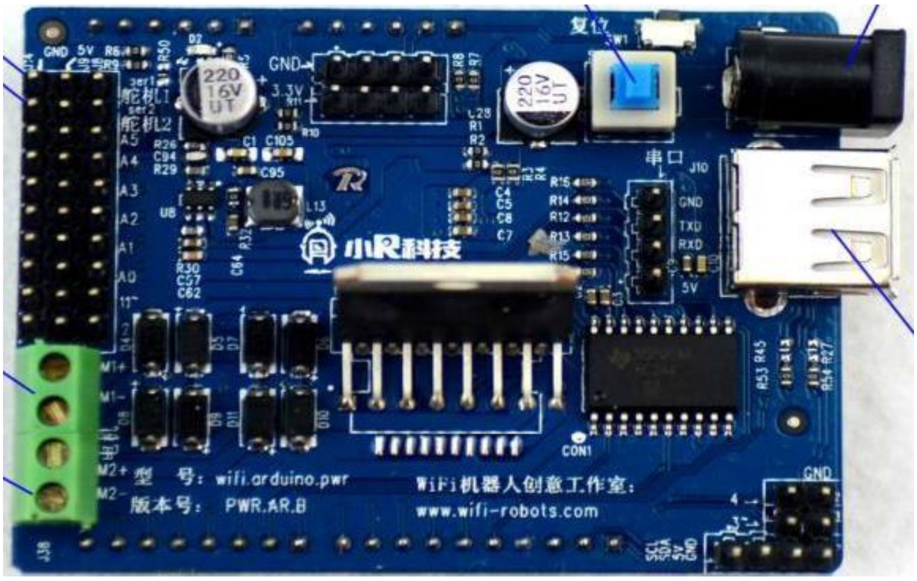

Figure 4.4: L298N Motor shield drive

4.3.3 WI-FI module

WI-FI is a popular electronic device that sends radio waves over the network to allow the exchange of information and data wirelessly .Covers an area ranging from 50 meters outdoors and less inside the walls

Features

- Through the WIFI transmission of video.
- Forwarding instructions from the network to serial output.
- Dual antenna 300Mbps wireless throughput. Drive the camera default 640x480 30fps.
- Each module has been calibrated to ensure performance.

Module function

- The acquisition of USB camera image, sent to the client through the jpg format display.
- The network - serial port forwarding

Application scenario

- Wi-Fi video car robot.
- Smart home.

Figure (4.5) represents the WI-FI module:

Figure 4.5: WIFI MODULE

4.3.4 Camera

The Robot-Eyes USB Video Camera is used in robotic car, shown in figure (4.6): Camera

Figure 4.6: Camera

The camera plugs into your system via a USB connector. An excellent camera for when you are using First Person View (FPV) to drive your car around or just for videoing where you go.

Specifications:

- Focal Length: 100mm to infinity.
- Hardware Pixels: 300K.
- Resolution: 640x480.
- Focus: Manual.

4.3.5 DC Motor

12 V DC motor it can be used in many areas, especially in robots, with relatively lightweight and high torque, allowing it to climb the slopes and some hills.

Figure (4.7) represents the DC motor with wheel:

Figure 4.7: DC motor with wheel

4.3.6 Servo motor

The servomotor have gears, which allows to control them accurately. Because of that can allow the base to be placed at different angles, usually between 0 and 180 degrees.

Figure (4.8) represents the Servomotor:

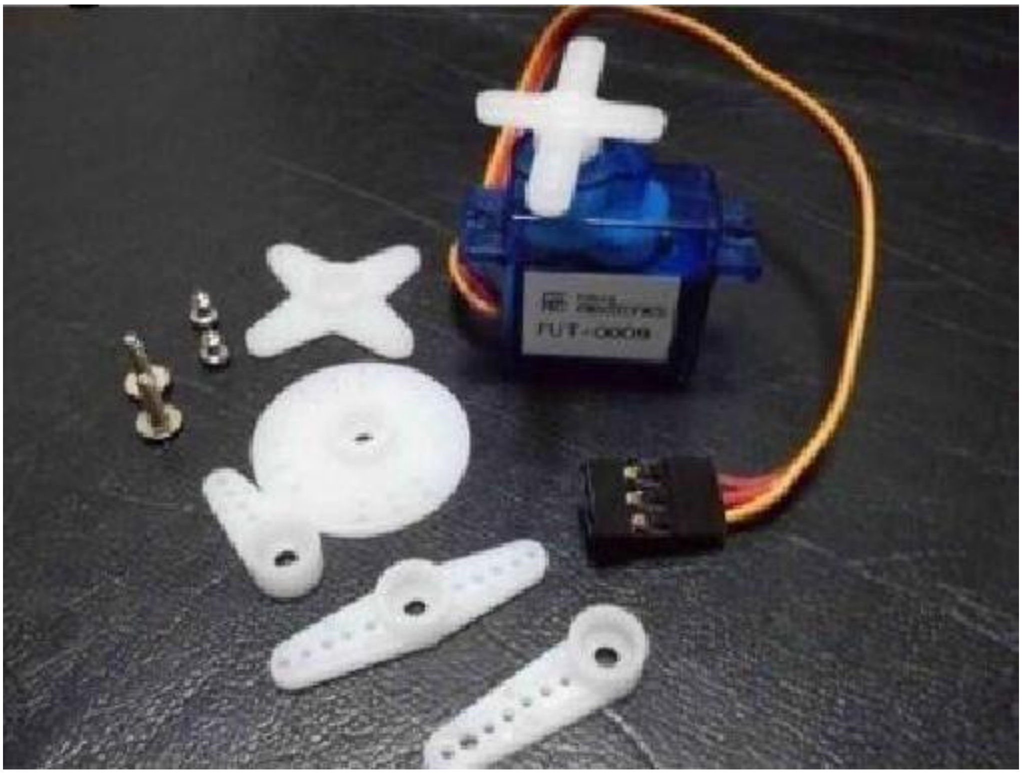

Figure 4.8: Servomotor

4.3.7 Ultrasonic sensor HC-sr04

This sensor is attached to detect the distance of the obstacle from the robot. It uses sonar to govern distance of an object. It inaugurate non- contact range detection, and provide stable reading in an easy to use package. Its range varies from 2 cm to 400 cm or 1’’ to 13 feet. Sensor is not affected by sunlight or black material but it is difficult to detect the distance from any soft material like cloth.

It is a combination of both ultrasonic transmitter and receiver module. Its output is greatly perturbed by Echo signal, so the output never goes Low if Echo is not received. Even timeout parameters are needed to alter the output according to the user aspirations. Its resolution is 0.3cm and trigger input pulse width is 10µS.

Figure (4.9) represents the Ultrasonic sensor:

Figure 4.9: Ultrasonic sensor

4. 4 System Operation

The circuit diagram of the system is shown in the figure(4.10)

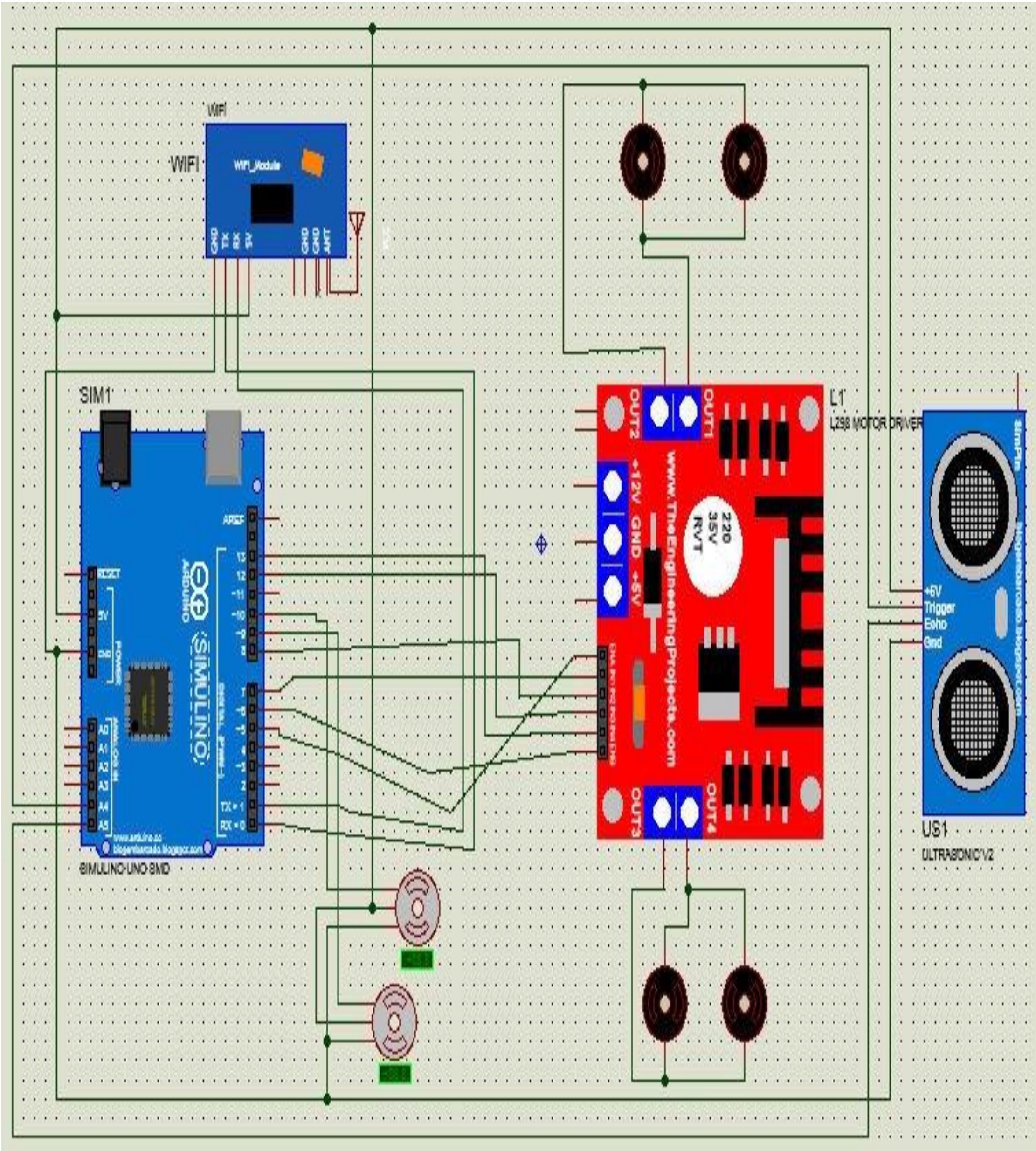

Figure 4.10: Circuit diagram of the system

When the power switch is turned on, all devices (Arduino, servo motor, Wi-Fi and camera) Will be operated through the power shield, Wi-Fi transmits video and receive control commands from the application and sends them to the controller, As arduino receives the signal from Wi-Fi Module, it translate the signals using the code, then Arduino send commands to the

components to be executed.
Different signals such as forward, back, left and right movement of the Robot.

Wi-Fi based robot car Remote control using phone (Android) with a vector for the picture and video to the phone is designed and implemented.

The results in table (4.1) and (4. 2) Represents the motion and the speed of the system respectively were taken during system running.

Table (4.1): The robot motion

Condition of motor	(M1) LEFT	(M2) LEFT	(M3) RIGHT	(M4) RIGHT
Left & Right Motor-Stop	Stop	Stop	Stop	Stop
Forward	Anticlockwise	Anticlockwise	Anticlockwise	Anticlockwise
Back	Clockwise	Clockwise	Clockwise	Clockwise
Left	Anticlockwise	Anticlockwise	Clockwise	Clockwise
Right	Clockwise	Clockwise	Anticlockwise	Anticlockwise

The Servo motor, which controls the base of the camera ,It is controlled by the phone by passing the finger on the screen with a horizontal movement of 180 degrees or vertical movement of 90 degrees. The video is taken from the camera and sent to the controller and Wi-Fi model, as shown in figure 4.11

Figure 4.11: side view of the system

Table (4.2): speed of robot

Experiment number	Time	Distance	Speed
I	17.07 s	10 m	0.58 m/s
II	16.65 s	10 m	0.60 m/s
III	17.03 s	10 m	0.587 m/s
IV	16.77 s	10 m	0.59 m/s
V	16.80 s	10 m	0.59 m/s

There are four buttons as shown as in figure 4.14, forward, right, left and back. When the buttons are pressed, the signal is transferred to the robot and then to the controller .The signal is analyzed, the transmitted step is taken if it is forward, backward, right or left the robot start move .Table 4.1 explains the condition of motor and its motion when the robot move.

Figure (4.12) represents the Screenshot of android app:

Figure 4.12: Screenshot of android app

The reason for multiple values in table 4.2 is due to several

Reasons:

- The ground is not straight.
- The floor is a little bumpy.
- Human errors.

If the robot was to be driven to very far away, more than it has rang, the camera wouldn't be able to send back data.

The robot is able to wrap around itself at a 360 degree angle. The car avoids obstacles and objects in the case of automatic control. If the car meets a small object vertically (base of chair), it's can't recognize it.

The robot can be connected with Smartphone up to distance of about 40-50M with good controlling.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATION

5. 1 Conclusion

This system implemented so that the robot can be controlled by the smart phone, smart phone and robot are connected to the cordless wireless network, so the smart phone can connect with the robot up to a distance of 100 meters away. This robot can move freely in all directions forward, backward, left and right, with the possibility to rotate at 360 degrees at the same point. The robot has sensors on the front to avoid obstacles and objects. The robot has a camera mounted on the base and can move horizontally 180 degrees and 90 degrees vertically. The camera transfers photos and videos directly to the smart phone and displays them to smart phone or other wireless display device. The robot is not affected by airflow. The robot works in the closed and dingy places. The robot enjoys great freedom of movement. The robot has the ability to control speed, as the speed of the robot can be reduced in small places with safety.

5. 2 Recommendations

A great result obtained but with bellow recommendation better advantages could get; such as:

- Add an arm and pickup to the robot to perform tasks during observation.
- Add metal sensors for mineral exploration.
- Add night photography technology and thermal cameras.
- Enable navigation system by GPS system.
- Enable navigation system by maps system.

REFERENCES

[1] Hou-Tsan Lee, Wei-Chuan Lin, Ching-Hsiang Huang, Yu-Jhih Huang, “Wireless indoor surveillance robot”, in 2011 Proceedings of SICE Annual Conference (SICE), 2011.

[2] Change Zheng, “Mechanical design and control system of a miniature surveillance robot”, in ICIA '09, International Conference on Information and Automation, 2009.

[3] Kyunghoon Kim, “Intelligent surveillance and security robot systems”, in IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), 2010.

[4] Ki Sang Hwang, Kyu Jin Park, Do Hyun Kim, Sung-Soo Kim, Sung Ho Park, “Development of a mobile surveillance robot”, in ICCAS '07, International Conference on Control, Automation and Systems, 2007.

[5] Christian Hernández, Raciel Poot, Lizzie Narváez, Erika Llanes and Victor Chi, “Design and Implementation of a System for Wireless Control of a Robot”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 5, September 2010

[6] Phey Sia Kwek, Zhan Wei Siew, Chen How Wong, BihLiiChua, Kenneth Tze Kin Teo ,Development Of A Wireless Device Control Based Mobile Robot Navigation System. IEEE 2012.

[7] Pavan.C, Dr. B. Sivakumar Wi-Fi Robot For Video Monitoring Surveillance System International Journal of Scientific Engineering Research Volume 3, Issue 8, August-2012.
[8] A.Sivasoundari, S.Kalaimani, M.Balamurugan Wireless Surveillance Robot With Motion
Detection And Live Video Transmission International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319â“6378, Volume-I, Issue-6 April 2013.
[9] Article ID 197105 Indoor Surveillance Security Robot with a Self-Propelled Patrolling Vehicle Volume 2011.
[10] Shingo Ito and Andrew A. Goldenberg ,A mobile robot with autonomous climbing and descending of stairs.
[11] Sebastian van Delden and Andrew Whigham, A Bluetooth-based Architecture for Android Communication with an Articulated Robot , IEEE.
[12] Arpit Sharma, Reeteshverma, SaurabhGupta,Sukhdeepkaurbhatia, Android phone controlled robot using Bluetooth , IJEEE,Vol.7, Nov-2014.
[13] M. Selvam, Smart phone based robotic control for surveillance application ,IJRET,Vol.3, Issue-3, pp-229-232, Mar- 2014.
[14] Gu, Yi, et al. Design and Implementation of UPnP-Based Surveillance Camera System for Home Security." Information Science and Applications (ICISA), 2013 International Conference on. IEEE, 2013.

[15] "Updated: Arduino announces FPGA board, ATmega4809 in Uno Wi-Fi mk2, cloud-based IDE and IoT hardware". Electronics Weekly. 2018-05-18. Retrieved 2018-06-14

[16] Android Developers Site [Online]. Available: http://developer.android.com/index.html, September 2018.

[17] Mejdl Safran and Steven Haar, "Arduino and Android Powered Object Tracking Robot". [Online]. Available: http://faculty.ksu.edu.sa/mejdl/Publications/Android_Arduino_Robot.Pdf.

[18] https://en.wikipedia.org/wiki/Robot

[19] https://www.nist.gov/blogs/manufacturing-innovation-blog/4-types-robots-every-manufacturer-should-know

[20] https://en.wikipedia.org/wiki/Robot_software

APPENDIX A

Hardware structure

The basic structure of robot top view, robot button view and camera robot used to shape the basic of the robot, it is made from aluminum alloy with weight 1500 g and lengthwise dimensions of 28 cm and 17 cm width. The chassis is convenient for robot already have microcontroller, Wi-Fi Module and camera, sensors on the platform.

The module have, above chassis there's microcontroller and other, under chassis there's four DC motors and sensor. Microcontroller, driver motor, Wi-Fi Module and battery are placed on chassis. The camera is attached at the front end on the upper of the chassis.

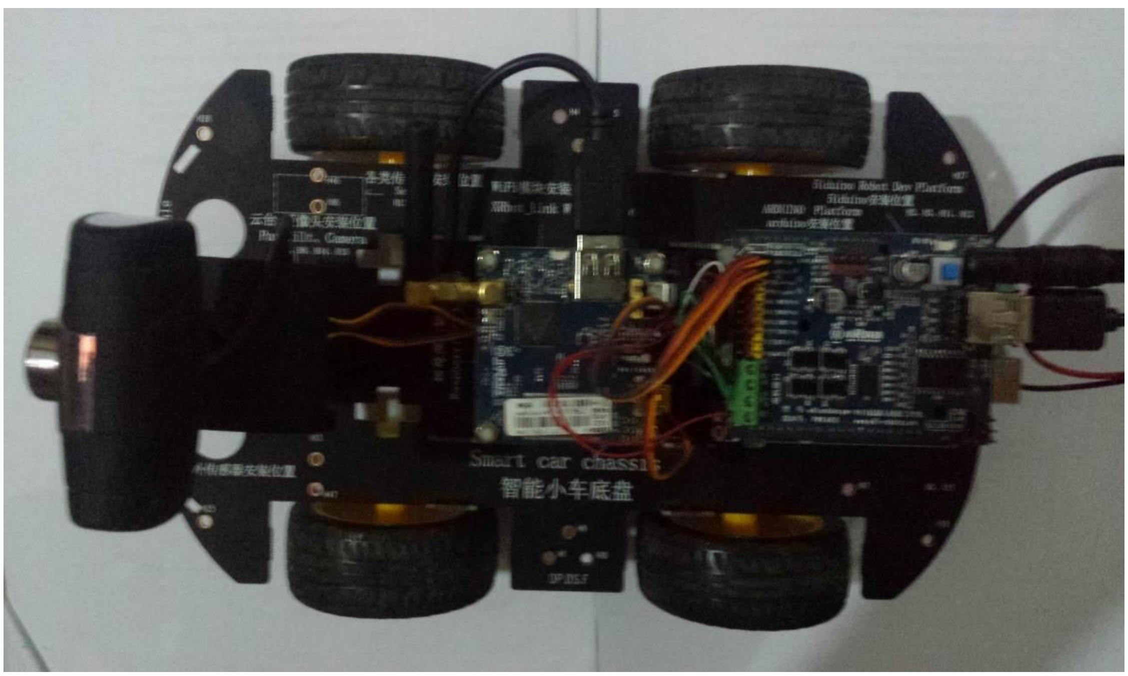

Figure A1 Basic Structure of the robot top view

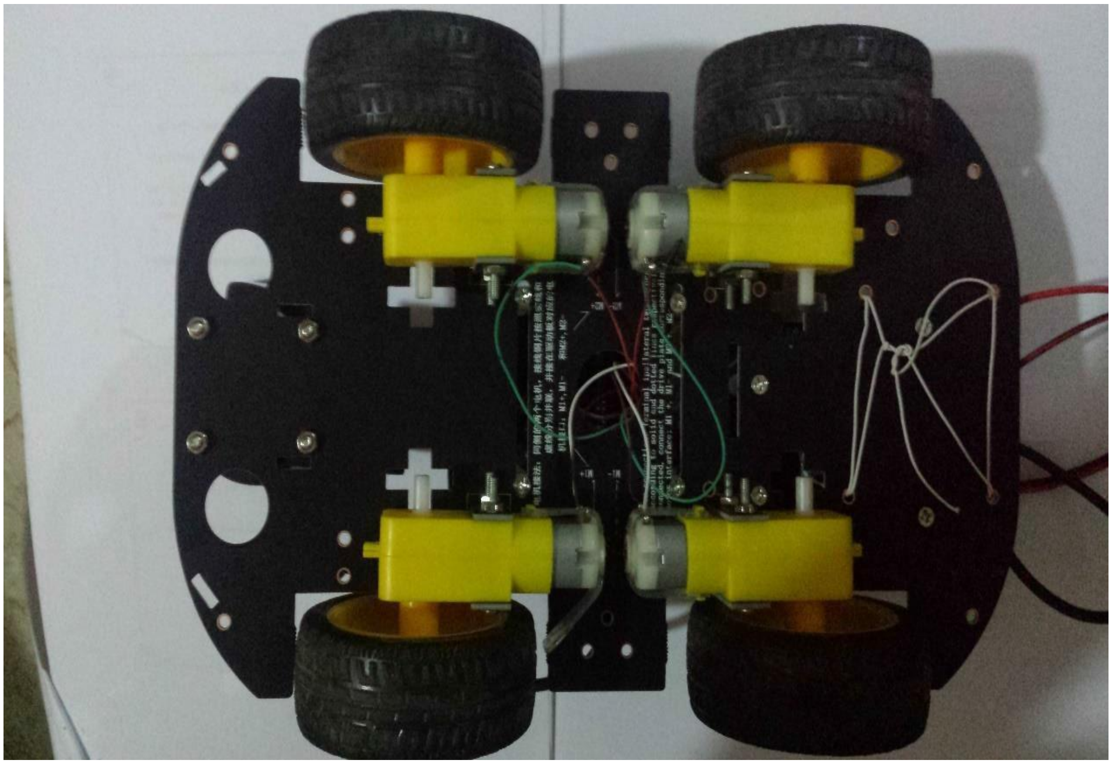

Figure A2 Basic structure of the robot bottom view

Figure A3 Camera robot

Appendix B

Arduino code

```

#include <Servo.h>
#include <EEPROM.h>
int ENA = 5;//L298 Enable A
int ENB = 6;//L298 Enable B
int INPUT2 = 7;//Motor interface 1
int INPUT1 = 8;//Motor interface 2
int INPUT3 = 12;//Motor interface 3
int INPUT4 = 13;//Motor interface 4
int num;//Define motor flag
int Echo = A5;  // Define the ultrasonic signal receiving pin
int Trig = A4;  // Define the ultrasonic signal emitter
int Carled = A0;//Defining the car light interface
int Cruising_Flag = 0;
int Pre_Cruising_Flag = 0 ;
int Left_Speed_Hold = 255;//Defining the left side speed variable
int Right_Speed_Hold = 255;//Define right speed variable

//Macro defines the car steering direction
#define MOTOR_GO_FORWARD {digitalWrite(INPUT1,LOW);digitalWrite(INPUT2,HIGH);digitalWrite(INPUT3,LOW);digitalWrite(INPUT4,HIGH);}    //Car body advance
#define MOTOR_GO_BACK {digitalWrite(INPUT1,HIGH);digitalWrite(INPUT2,LOW);digitalWrite(INPUT3,HIGH);digitalWrite(INPUT4,LOW);}    //Car body advance
#define MOTOR_GO_RIGHT {digitalWrite(INPUT1,HIGH);digitalWrite(INPUT2,LOW);digitalWrite(INPUT3,LOW);digitalWrite(INPUT4,HIGH);}    //Car body advance
#define MOTOR_GO_LEFT {digitalWrite(INPUT1,LOW);digitalWrite(INPUT2,HIGH);digitalWrite(INPUT3,HIGH);digitalWrite(INPUT4,LOW);}    //Car body advance
#define MOTOR_GO_STOP {digitalWrite(INPUT1,LOW);digitalWrite(INPUT2,LOW);digitalWrite(INPUT3,LOW);digitalWrite(INPUT4,LOW);}    //Car body advance


void forward(int num)
{
  switch(num)
  {
    case 1:MOTOR_GO_FORWARD;return;
    case 2:MOTOR_GO_FORWARD;return;
```

```
    case 3:MOTOR_GO_BACK;return;
    case 4:MOTOR_GO_BACK;return;
    case 5:MOTOR_GO_LEFT;return;
    case 6:MOTOR_GO_LEFT;return;
    case 7:MOTOR_GO_RIGHT;return;
    case 8:MOTOR_GO_RIGHT;return;
    default:return;
  }
}

void back(int num)
{
    switch(num)
  {
    case 1:MOTOR_GO_BACK;return;
    case 2:MOTOR_GO_BACK;return;
    case 3:MOTOR_GO_FORWARD;return;
    case 4:MOTOR_GO_FORWARD;return;
    case 5:MOTOR_GO_RIGHT;return;
    case 6:MOTOR_GO_RIGHT;return;
    case 7:MOTOR_GO_LEFT;return;
    case 8:MOTOR_GO_LEFT;return;
    default:return;
  }
}
void left(int num)
{
    switch(num)
  {
    case 1:MOTOR_GO_LEFT;return;
    case 2:MOTOR_GO_RIGHT;return;
    case 3:MOTOR_GO_LEFT;return;
    case 4:MOTOR_GO_RIGHT;return;
    case 5:MOTOR_GO_FORWARD;return;
    case 6:MOTOR_GO_BACK;return;
    case 7:MOTOR_GO_FORWARD;return;
    case 8:MOTOR_GO_BACK;return;
    default:return;
  }
}
void right(int num)
{
    switch(num)
  {
    case 1:MOTOR_GO_RIGHT;return;
    case 2:MOTOR_GO_LEFT;return;
```

```
   case 3:MOTOR_GO_RIGHT;return;
   case 4:MOTOR_GO_LEFT;return;
   case 5:MOTOR_GO_BACK;return;
   case 6:MOTOR_GO_FORWARD;return;
   case 7:MOTOR_GO_BACK;return;
   case 8:MOTOR_GO_FORWARD;return;
   default:return;
 }
}


Servo servo7;// Create servo #7
Servo servo8;// Create servo #8
byte angle7=60;//Servo #7 initial value
byte angle8=60;//Servo #8 initial value

char Get_Distence()//Measuring distance
 {
    digitalWrite(Trig, LOW);   // Let the ultrasonic wave emit low voltage 2μs
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);  // Let the ultrasonic wave emit high voltage 10μs,
 here is at least 10μs
    delayMicroseconds(10);
        digitalWrite(Trig, LOW);    // Maintain ultrasonic emission low voltage
        Ldistance = pulseIn(Echo, HIGH);  // Reading difference time
        Ldistance= Ldistance/5.8/10;      // Turn time into distance (unit: cm)
      //  Serial.println(Ldistance);      //Display distance
        return Ldistance;

     }


void Avoid_wave()//Ultrasonic obstacle avoidance function
{
  Get_Distence();
  if(Ldistance < 15)
    {
      MOTOR_GO_STOP;
    }
    else
    {
       forward(num);
    }
}

void  Delayed()    //Delay 40 seconds and other WIFI modules started
{
```

```
    int i;
    for(i=0;i<20;i++)
    {

      delay(1000);

      delay(1000);
    }
  }
130, void Init_Steer()//Servo initialization (angle is the last saved value)
  {
    angle1 = EEPROM.read(0x01);//Read the value in register 0x01
    angle2 = EEPROM.read(0x02);//Read the value in register 0x02
    angle3 = EEPROM.read(0x03);//Read the value in register 0x03
    angle4 = EEPROM.read(0x04);//Read the value in register 0x04
    angle7 = EEPROM.read(0x07);//Read the value in register 0x07
    angle8 = EEPROM.read(0x08);//Read the value in register 0x08

    if(angle7 == 255 && angle8 == 255)
    {
      EEPROM.write(0x01,60);//Store the initial angle in address 0x01
      EEPROM.write(0x02,60);//Store the initial angle in address 0x02
      EEPROM.write(0x03,60);//Store the initial angle in address 0x03
      EEPROM.write(0x04,60);//Store the initial angle in address 0x04
      EEPROM.write(0x07,60);//Store the initial angle in address 0x07
      EEPROM.write(0x08,60);//Store the initial angle in address 0x08
      return;
    }

    servo1.write(angle1);//Assigning Save Angle to Servo 1
    servo2.write(angle2);//Assigning Save Angle to Servo 2
    servo3.write(angle3);//Assigning Save Angle to Servo 3
    servo4.write(angle4);//Assigning Save Angle to Servo 4
    servo7.write(angle7);//Assigning Save Angle to Servo 7
    servo8.write(angle8);//Assigning Save Angle to Servo 8
    num = EEPROM.read(0x10);//Read the value in register 0x10
    if(num==0xff)EEPROM.write(0x10,1);

    Left_Speed_Hold = EEPROM.read(0x09);//Read the value in register 0x03
    Right_Speed_Hold = EEPROM.read(0x0A);//Read the value in register 0x04
    if(Left_Speed_Hold<55|Right_Speed_Hold<55)
    {
     Left_Speed_Hold=255;
     Right_Speed_Hold=255;
    }
    analogWrite(ENB,Left_Speed_Hold);//Assign L298 to Enable B
```

```
    analogWrite(ENA,Right_Speed_Hold);//Assign L298 to Enable A
}

void setup()
{

    pinMode(ENA,OUTPUT);
    pinMode(ENB,OUTPUT);
    pinMode(INPUT1,OUTPUT);
    pinMode(INPUT2,OUTPUT);
    pinMode(INPUT3,OUTPUT);
    pinMode(INPUT4,OUTPUT);
    pinMode(Carled, OUTPUT);
    pinMode(Echo,INPUT);
    pinMode(Trig,OUTPUT);

    Delayed();//Delay 40 seconds and other WIFI modules started

    servo1.attach(3);//Define servo 1 control port
    servo2.attach(4);//Define servo 2 control port
    servo3.attach(2);//Define servo 3 control port
    servo4.attach(11);//Define servo 4 control port
    //servo5.attach(SDA);//Define servo 5 control port
    //servo6.attach(SCL);//Define servo 6 control port
    servo7.attach(9);//Define servo 7 control port
    servo8.attach(10);//Define servo 8 control port
    Serial.begin(9600);//Serial baud rate is set to 9600 bps
    Init_Steer();
}
void Cruising_Mod()//Mode function switching function
    {

   if(Pre_Cruising_Flag != Cruising_Flag)
   {
       if(Pre_Cruising_Flag != 0)
     {
         MOTOR_GO_STOP;
     }

       Pre_Cruising_Flag =  Cruising_Flag;
   }
  switch(Cruising_Flag)
    {
```

```
         case 4:Avoid_wave();return;//Ultrasonic obstacle avoidance mode
               case 5:Send_Distance();//Ultrasonic distance PC display
         default:return;
        }

    }

void loop()
  {
    while(1)
    {
        Get_uartdata();
        UartTimeoutCheck();
        Cruising_Mod();
     }
  }
void Communication_Decode()
{
    if(buffer[0]==0x00)
    {
        switch(buffer[1])   //Motor command
        {
            case 0x01:MOTOR_GO_FORWARD; return;
            case 0x02:MOTOR_GO_BACK;    return;
            case 0x03:MOTOR_GO_LEFT;    return;
            case 0x04:MOTOR_GO_RIGHT;   return;
            case 0x00:MOTOR_GO_STOP;    return;
            default: return;
        }
    }
   else if(buffer[0]==0x01)//Servo Command
    {
        if(buffer[2]<1)return;
        switch(buffer[1])
        {
            case 0x01:if(buffer[2]>170)return;angle1 = buffer[2];servo1.write(angle1);return;
            case 0x02:if(buffer[2]>170)return;angle2 = buffer[2];servo2.write(angle2);return;
            case 0x03:if(buffer[2]>170)return;angle3 = buffer[2];servo3.write(angle3);return;
            case 0x04:if((buffer[2]<110)||(buffer[2]>178))return;angle4 = buffer[2];servo4.write(angle4);return;
            //case 0x05:if(buffer[2]>170)return;angle5 = buffer[2];servo5.write(angle5);return;
            //case 0x06:if(buffer[2]>170)return;angle6 =
```

```
buffer[2];servo6.write(angle6);return;
            case 0x07:if(buffer[2]>170)return;angle7 = buffer[2];servo7.write(angle7);return;
            case 0x08:if(buffer[2]>170)return;angle8 = buffer[2];servo8.write(angle8);return;
            default:return;
        }
    }

    else if(buffer[0]==0x02)//Speed regulation
    {
     if(buffer[2]>100)return;

     if(buffer[1]==0x01)//Left shift
     {
                  Left_Speed_Hold=buffer[2]*2+55;//Speed gear is 0~100 Convert to pwm Speed pwm is less than 55 motor does not turn
                  analogWrite(ENB,Left_Speed_Hold);
                  EEPROM.write(0x09,Left_Speed_Hold);//Storage speed
     }
              if(buffer[1]==0x02)//Right side shift
              {
                  Right_Speed_Hold=buffer[2]*2+55;//Speed gear is 0~100 Convert to pwm Speed pwm is less than 55 motor does not turn
                  analogWrite(ENA,Right_Speed_Hold);
                  EEPROM.write(0x0A,Right_Speed_Hold);//Storage speed
              }else return;
        }
     else if(buffer[0]==0x33)//Read the servo angle and assign value
    {
      Init_Steer();return;
        }
     else if(buffer[0]==0x32)//Save command
     {
        EEPROM.write(0x01,angle1);
        EEPROM.write(0x02,angle2);
        EEPROM.write(0x03,angle3);
        EEPROM.write(0x04,angle4);
        //EEPROM.write(0x05,angle5);
        //EEPROM.write(0x06,angle6);
        EEPROM.write(0x07,angle7);
        EEPROM.write(0x08,angle8);
        return;
     }
       else if(buffer[0]==0x13)//Mode switch
    {
```

```
        switch(buffer[1])
      {

        case 0x02: Cruising_Flag = 2; return;
        case 0x03: Cruising_Flag = 3; return;
        case 0x04: Cruising_Flag = 4; return;
        case 0x05: Cruising_Flag = 5; return;
        case 0x00: Cruising_Flag = 0; return;
        default:Cruising_Flag = 0; return;
      }
    }
        else if(buffer[0]==0x04)
      {
        switch(buffer[1])
        {


            default: return;
        }
      }
        else if(buffer[0]==0x40)
    {
       num=buffer[1];
       EEPROM.write(0x10,num);
    }
  }
  void Get_uartdata(void)
  {
      static int i;

      if (Serial.available() > 0) //Determine whether the serial buffer has data loaded
      {
          serial_data = Serial.read();//Read the serial port
          if(rec_flag==0)
          {
              if(serial_data==0xff)
              {
                  rec_flag = 1;
                  i = 0;
                 Costtime = 0;
              }
          }
          else
          {
              if(serial_data==0xff)
```

```
            {
                rec_flag = 0;
                if(i==3)
                {
                    Communication_Decode();
                }
                i = 0;
            }
            else
            {
                buffer[i]=serial_data;
                i++;
            }
        }
    }
}
void UartTimeoutCheck(void)
{
    if(rec_flag == 1)
    {
       Costtime++;
      if(Costtime == 100000)
      {
          rec_flag = 0;
      }
    }
}
```