Palestine Polytechnic University
College of Engineering &Technology
Electrical Engineering Department
Electronics & Telecommunications Engineering

Graduation Project

Remote Control of vehicles using Wi-Fi technology

Project Team

Abedelfattah H. Drabee

Ahmad A. Drabee

Hothayfa I. Najajreh

Project Supervisor

Eng. Ahmad Qudemat

2013-2014

جامعة بوليتكنك فلسطين
الخليل – فلسطين

كلية الهندسة و التكنولوجيا
دائرة الهندسة الكهربائية و الحاسوب

اسم المشروع

"Remote Control of vehicles using Wi-Fi technology"

أسماء الطلبة

أحمد درابيع حذيفة نجاجرة عبدالفتاح درابيع

بناء على نظام كلية الهندسة و التكنولوجيا و إشراف و متابعة المشرف المباشرة على المشروع وموافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة الالكترونيات والاتصالات.

توقيع المشرف

....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

....

Abstract

This project is about designing a robotic vehicle controlled using Wi-Fi technology which will be used in helping users to discover places that may be difficult to enter and most of environments that people need to know but they can't access. This vehicle will provide alive video to give users a clear vision about the nature of the surrounding environment. The Wi-Fi topology will be used for sending control commands and receiving live video between the vehicle and a smartphone. For controlling vehicle, relays were connected with the raspberry by GPIO pins. Once an order came from the android app to the vehicle through Wi-Fi, relays will give orders to motors to be moved due to the users order. In the side of the user, an android application will be used for displaying received video and controlling the movement of the vehicle.

This project falls within the fields of communication, so a lot of challenges will face us in implementing the system like programing each device, defining needed ports in raspberry that we will use and writing and designing the android application.

المقترح

الهدف من وراء هذا المشروع هو بناء مركبة صغيرة تحمل كاميرا لمساعدة المستكشفين والباحثين بالاضافة الي طواقم الانقاذ في الولوج الى اماكن يصعب على الانسان الوصول اليها بحيث سيتم التحكم بهذه المركبة وعرض الفيديو الحي عن طريق تطبيق اندرويد على هاتف ذكي، بالاضافة الا قطعة رازبيري موصولة مع قطعة واي فاي و متحكمات للتحكم بجسم المركبة بحيث سيتم ربط ربط الهاتف الذكي مع جسم المركبة باستخدام تقتية الواي فاي.

هذا المشروع يندرج تحت نطاقي التحكم والبرمجة لذلك فهناك مجموعة من التحديات التي ستواجهنا أثناء تطبيق هذا المشروع مثل: برمجة القطع وتعريف مداخل قطعة الرازبيري، بالإضافة إلى كتابة وتصميم تطبيق اندرويد يحتوي على شاشة عرض للفيديو وازرار تحكم .

Dedication

We dedicate this work

To our families

To our teachers

To our friends

To everyone who help us

Acknowledgment

At the outset we thank Almighty God for the blessing of the mind and knowledge, Then extend our sincere thanks for all the teachers who have given us science and knowledge in previous years, to our parents and families who supported us and were beside us every time Our special thanks to our supervisor. Eng. Ahmad Qudemat for his great supervision and support.

Table of Contests

Contents

List of Tables & Figures

Tables

Figures

Chapter 1
Introduction

1.1 overview

Robotics is the branch of technology that deals with the design, construction, operation, and application of robots as well as computer systems for their control, sensory feedback, and information processing. The design of a given robotic system will often mix principles of mechanical engineering, electronic engineering, and computer science.

The concept of creating machines that can operate autonomously dates back to classical times, but research into the functionality and potential uses of robots did not grow substantially until the 20th century. Throughout history, robotics has been often seen to mimic human behavior, and often manage tasks in a similar fashion. Today, robotics is a rapidly growing field, as technological advances continue, research, design, and building new robots serve various practical purposes, whether domestically, commercially, or militarily. Many robots do jobs that are hazardous to people such as defusing bombs, mines and exploring shipwrecks.

One of the most important technological challenges that faces humans' digital revolution is the ability of controlling different stuffs from different existence location using computers or other machines. A lot of applications could be done through this specialization like: medical operations, cars industry and movement control at different cases.

1.2 General Idea

This project is based on hardware plus software combination with each other to invent a remote controlled tracked toy car using smartphone. By using this project we will be able to capture live video using a small camera deployed on the tracked body of the toy car, movement of our tracked car will be in four directions Left, Right, forward and revers in order to have the best shot we need.

The connection type between the smartphone and the toy car will be wireless line using Wi-Fi technology. The toy car will be able -by the end of our project- to transfer video synchronously with the movement of the toy car. A small video camera will be deployed on the toy car and connected to processing unit which will redirect the video packets to the transmitting unit in order to display them on the smartphone after doing the needed programming tasks.

Problem statement

In some cases of our life, people may face some difficulties in discovering areas since they could be dangerous or very narrow so that no one could enter them. Our tracked toy car will help in discovering these places due to its small size and its ability to travel across bugged roads and areas. Additionally the video camera deployed on it will be very useful in capturing a live video for these places.

1.4 Motivations

Robots have replaced humans in the assistance of performing those repetitive and dangerous tasks which humans prefer not to do, or are unable to do due to size limitations, or even those such as in outer space or at the bottom of the sea where humans could not survive the extreme environments. This project will be able to exceed most of these difficulties and will be very beneficial in dangerous situation and rescue operations.

1.5 Objectives of the project

The proposed project is assumed to achieve these objectives:

- The ability of the system to process commands quickly and try to reduce delay time response.
- Controlling the toy car in a way enables us to move it in four directions (forward, back, right and left).
- Transferring video between toy car and smartphone with the best possible resolution.
- Trying to minimize the time delay in transferring video packets.

1.6 Time Planning

The time schedule must be determined in specific way and includes the related tasks of study and system analysis. The project was divided in to the following stages.

Table 1: Timeline

Task	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1. Collection of references	■	■	■	■	■	■	■	■	■	■	■	■	■	■
2. Preparation of proposal		■	■											
3.Learning to use equipment's and software				■	■	■								
4.connecting the needed components						■	■	■	■					
5.Writing the suitable program										■	■	■	■	
6.building the connection line between computer and the toy car and also testing the project work									■	■	■	■	■	
7.Writing report													■	■
8.Delivery of report to supervisor for review													■	■
9.Submission of report to department														■

1.7 Budget Table

The following is a list of different costs needed to implement the system**:**

Table 2: cost of components

No.	Component	Cost ($)
1	**Raspberry**	**200**
2	**Memory Card**	**20**
3	**Robot body with track**	**200**
5	**Battery**	**60**
6	**Battery Cable**	**10**
7	**Self-USB connector**	**40**
8	**Wireless adapter**	**100**
9	**Raspberry pi camera**	**80**
10	**Relay (4 pieces)**	**40**
Total Cost		**850$**

1.8 Benefits of the project

There are several benefits for the project as following:

- This project will be an important and beneficial exploring way to explorers and scientists who are specialized in studying the nature, the wildlife and the places that are hard to reach.

- This project also decreases the dangers for wildlife researchers in exploring dangerous areas and locations.

1.9 Technologies

In order to finish our work in this project, we have to use these technologies:

Hardware:

- Raspberry.
- Smartphone.
- Memory Card.
- Robot body with track.
- Relay.
- Battery.
- Battery Cable.
- Self-USB connector.
- Wi-Fi USB adapter.
- Raspberry pi camera.

Software:

The software section is a combination of several programs:

Linux: It is the main operating system for raspberry pi device, since this operator designed to enable applications and the computer operator to access the devices on the computer to perform desired functions. Linux will be installed on the SD memory card in order to be connected to the raspberry to operate it.

Eclipse (Java): in order to design android application to received captured video from raspberry pi and display it in a small window in the application and control the vehicle direction of movement by four button in the application. We will use this software that support Java language.

Python: when we will connect GPIO pins to control circuit we need to programming pins that we used by python language.

Chapter 2
Literature review& background

2.1 introduction

If we look at the world around us, we find that the technology has embraced the human being and took over most areas of his life, especially with the progress of science and innovation in new ways in which access to what meets the human needs and desires that cannot be broken.

In the world of technology, there are two great human innovations, which are computer technology and robotic technology. We use a computer system to perform various difficult tasks, robotic controlling Based on the usefulness of computers. It is very difficult to imagine how people will live without computer systems. The second great human innovation is a robotic system. There are many kinds of robots with such tasks as a robot arm to service an industrial purpose, an army robot to detonate a buried bomb, an android robot to do a human job and children's toy car or an RC car that can be operated with radio waves, etc. Normally, a robotic system needs some computer systems to manage it.

In this chapter, the general concepts is to show the background that related to our project, and discuss what previous researchers have used in their projects from algorithms and techniques.

2.2Previous studies

The principle of robotic was presented many years ago. A lot of people and researchers worked and still working in this field to use it in different applications. By doing group search in the internet and university library, we found different documents and papers in this field as following.

In [1] RC-Car-Control-Programming project. There are two main parts of this project: the PC and the car. Between the two there are RF link for transmitting commands and a separate wireless video link which isn't shown in our project for video transmission. They made the RF link to be a one way communication because the RF components they found support this mode of operation. Used in this project IR LED (infrared LED) and an IR sensor, and connect to special circuit. These circuits are used for avoiding collision with obstacles. Electrical pulses are regularly sent to fire the IR LEDs ON and OFF and two readings are taken during a single period of a pulse, so that the controller "knows" the real reading from the IR sensor. The next thing is an RF receiver. Commands are constantly transmitted from the RF transmitter connected to the serial port of a PC as a form of series of bits and received by the RF receiver located on the car. They have created a simple C program for command bits generation and for video display and replay functions. To control DC motors on the RC car, they used two H-bridges as shown before. These two H-bridges are driven by outputs of micro-controller through one of parallel port pins.

In [2] this project is a robot vehicle that can be controlled through any computer wirelessly using a Wi-Fi link. The video transmitting camera mounted on it is 360° rotatable. This project features a robot that uses Wi-Fi 802.11G standard for its control signals through TCP/IP protocol which has flow control. This robot is capable of moving with speed of 12Km/h up to a distance of 200m from the controlling computer. Wi-Fi link is used here because it is a good and more flexible option for being used as a communication link for controlling robot in comparison to RF link because by using Wi-Fi, we can increase the range of this robot by the use of access points and secondly it can also be integrated to interne by developing a web based console. The project discussed above gives a good way for improving the distance of control at our project which is designing a web based console and using multiple Wi-Fi access points. This way can be used in the future to do some enhancements on our project. This project works as a system tool that depends on the parallel port of the PC .It is designed for search & detective operation. By using this project we capture live video by using the car. We perform mainly four operations Left, Right, Up and Down to move the car in particular direction. We use a Webcam for watching the live video. We are using the Parallel Port to provide signal to the interface card so that it can transmit signals to the car and guide it to move in a particular direction.

In [3] at this project, a transmitter connected to the system sends the signals which are received by the receiver on the toy car. These signals trigger the appropriate circuits and the desired motion is resulted. The receiver of the camera is also connected to the system. The user sees the images sent by the on-board camera and manipulates the motion of the car. The movement of the car lies in the manipulation of the two DC motors. The software sends signals using 4 pins of the 25 pin parallel port. These are the various values sent for various actions.

In [3] Jon Bennett, an Electrical Engineer, designed a Wi-Fi robotic car controlled by computer, He used a WRT54GL router with two serial connections. One of them connected to the microcontrollers witch controlling motors moving the car. He used three software. The Car Server which is written in C and runs on the router running open WRT White Russian v0.9 (Linux), and the microcontroller firmware.

In [4] another project paper called (A Surveillance Robot for Disaster Sites), describes the design of a robotic car controlling via Wi-Fi to help people in disasters and danger places. This project using LPC 2138 microcontroller as a brain of the system, C language as software built in a laptop device and camera and different sensors like temperature, fire detector, sound detector and human body detector. If any of the sensors alert, microcontroller gives order to camera to takes some photo to assess the situation.

In [5] another study paper (Wi-Fi Enabled Robots for Engineering Education) describes a set of workshops in which Wi-Fi enabled robots were constructed in order to introduce students by hands-on experiments on handling electronic components and software system design. Authors show that we have two wireless standards in common used, IEEE 802.1la/b/g (Wi-Fi) and IEEE 802.15 (Bluetooth). They have also described the different between them.

After we explain related works we find that Wi-Fi is a suitable wireless standard to use in our project, because Bluetooth designed to act as a communications link directly between two devices for short distances, while Wi-Fi can connect different devices in the same time. This project is different from the related work that we have discussed since we will use raspberry pi instead of Arduino or pics because of the raspberry high performance and its simplicity to deal with and it's able to do a lot of tasks. It’s just like motherboard in the computer. Another variation, we are going to control the car using a smartphone application while most of the previous projects have used a computer for the user interface.

2.3 Background:

2.3.1 Introduction to Wi-Fi:

Wi-Fi is a type of wireless networking protocol that allows devices to communicate without cords or cables. Wi-Fi is technically an industry term that represents a type of wireless local area network (LAN) protocol based on the 802.11 IEEE network standard. It's the most popular means of communicating data wirelessly, within a fixed location.

The newest standard, dubbed 802.11n, was designed to replace all three of the previous standards (a, b, g). It's up to five times faster than 802.11g, with a range almost twice as far. It also adds multiple-input multiple-output (MIMO) technology which uses multiple antennas to increase data transfer rates. The 802.11n standard was drafted to allow up to four channel configurations with potential speeds up to 600 Mb/sec. It's becoming increasingly popular for its high speeds, which allow for smoother audio and video streaming among devices.

Table 2.1: overview of the four most popular current 802.11 standards.

Standard	Frequency	Data Transfer Rate Typical (Max)	Range (indoor)
802.11a	3.7/5 GHz	20 (54) Mb/sec	about 35 m (115 ft)
802.11b	2.4 GHz	5.5 (11) Mb/sec	38 m (125 ft)
802.11g	2.4 GHz	22 (54) Mb/sec	38 m (125 ft)
802.11n	2.4/5 GHz	110+ (300+) Mb/sec	70m (230ft)

.

Why Wi-Fi?

- Fast data transfer rates:

  With transfer speeds around 150 megabits (Mb) per second (18.75 megabytes per second), 802.11n is currently the fastest commercially available Wi-Fi protocol on the market. It's more than capable of handling the demands of streaming high-definition TV signals, as well as CD-quality audio.

- Quick, easy setup:

  Setting up a wireless network may sound like a daunting task, but it's actually a pretty straight forward process. Wi-Fi networks don't require professional installation, and there are no holes to drill or wires to run through walls. Many new routers are "plug-and-play.

Wi-Fi limitation?

There are some limitations for Wi-Fi:

- Interference from other devices:

  Wi-Fi transmissions take place primarily within the 2.4 GHz spectrum, making them susceptible to interference from Bluetooth wireless enabled devices, cordless telephones.

- Limited range:

  Wi-Fi is a radio technology and so it is affected by things such as walls getting in the way. For example, at home the router might be downstairs and so the signal upstairs may be very weak.

2.3.2 Motor Controller Circuits

DC motors are simple two-lead, electrically controlled devices that convert electrical power into mechanical power through the interaction of two magnetic fields. One field is usually produced by a stationary permanent magnet (on the stator), and the other field is produced by an electric current flowing in the motor coil (on the rotor). The interaction of the two fields results in a torque that tends to rotate the rotor. For this experiment, a reversible, permanent magnet, brushed DC motor is selected. The term 'reversible' means the rotation of the motor can be reversed by simply flipping the terminals of the DC power supply.

In this project we decided that to control the vehicle in four directions (forward, revers, right and left), this tasks need to choose a suitable H-bridge driver motor to do this operate ,below we describe different type of driver motor that can be used..

Using Motor Bridges

Bridges allow your robot to control high current motors. They literally "bridge" between the low-current logic circuitry of your robot's brain to the high current demanded from the motors. Many, but not all, use the common "H" arrangement of control circuitry (usually power transistors) to provide On/off and directional controls to the motors -- hence the oft-cited name H-bridge.

Here's a short collection of additional motor bridge ideas you may want to ponder. Some are scratch built, while others rely on commercial bridge circuits.

L298 Motor H-Bridge

A very popular and reasonably priced all-in-one H-bridge motor control IC is the L298. It can control two motors, not just one.in [6] it can handle 2 amps per motor, though to get the maximum current be sure to add a heat sink. The L298 has a large cooling flange with a hole in it, making it easy to attach a homebrew metal heat sink to it.

If there's a downside to the L298 it's that it comes in a special "Multiwatt 15" package, with 15 offset pins that don't match the standard 0.100" spacing of breadboards. But with care, the pins can be repent as needed. Or you may prefer to simply get a breakout board for the L298, which is a small circuit board with holes drilled in it to accept the chip. You then plug the breakout board into your breadboard. Problem solved.

The schematic below shows a basic connection diagram for controlling two motors using the L298 motor bridge IC. There are three input pins for each motor: Input1, Input2, and Enable1 controls Motor1. Input3, Input4, and Enable2 controls Motor2, the motors connect to Output1/Output2 and Output3/Output4, as shown.

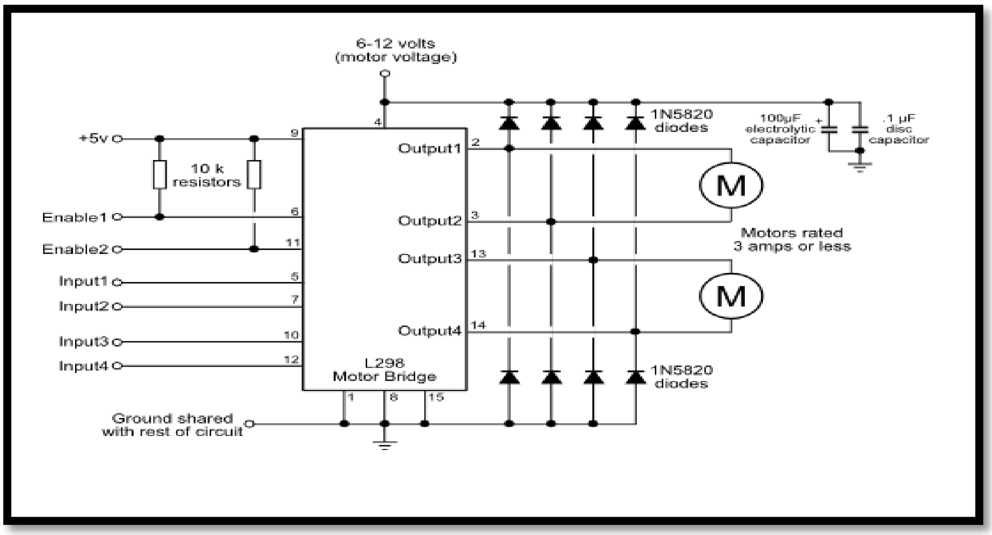


Figure 2.1: L298 motor control

Using an L298 Motor Board

As noted above, the L298 is one of the more popular bridge modules to use on fully-developed commercial motor control boards. For example, the Dual H-Bridge board, contains the L298 Bridge on a heat sink, fly back diodes, screw terminals, indicator LEDs, and other parts on a compact printed circuit board.

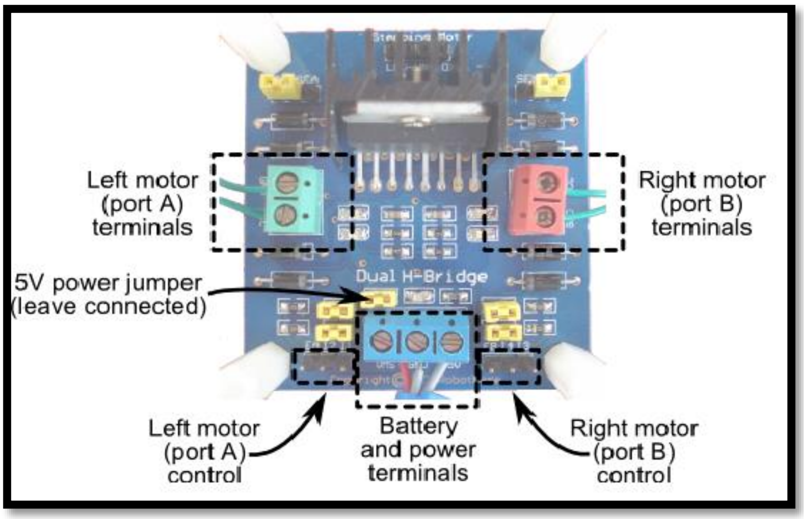


Figure 2.2: L298 motor board

Using a Motor Shield with the Arduino

In [7] popular method of controlling motors with the Arduino is to use a motor shield. This shield likewise is outfitted with an L298 bridge chip. Operation of the bridge is a bit different than the above, as the A/B inputs already have their own logic steering. You instead control each motor by setting a direction pin, and by making a PWM input line HIGH or LOW. The PWM input uses the Enable line for each motor, allowing you to control the speed of the motor was well as whether it is simply on or off.

Relay H-bridge(Relay Motor controller)

Relays require no human interaction in order for the switching to occur. They are electrically controlled mechanical switches. Inside the plastic box is an electro-magnet (coil), a switch, and a spring. The spring holds the switch in one position, until a current is passed through the coil. When the coil is energized with an electric pulse, a magnetic field is generated which moves the switch. The second part of a relay is a system of metallic arms which make up the physical contacts of the switch. When the relay is off, or no electric pulse is given to the relay, the arms of the switch are in one position. When the relay is on, or an electric pulse is sent to the relay, the swing or switching arm of the switch moves to another contact of the switch. The arm moves as the generated magnetic field pulls the swinging arm toward the inductor (or wire coil). When the relay is in the "off" position, the swing arm is in contact with the normally closed contact. When the relay is activated, the magnetic field created by the inductor coil pulls the swing arm until it makes contact with the normally open contact connecting the circuit connected to the normally open contact to the circuit connected to the main contact.

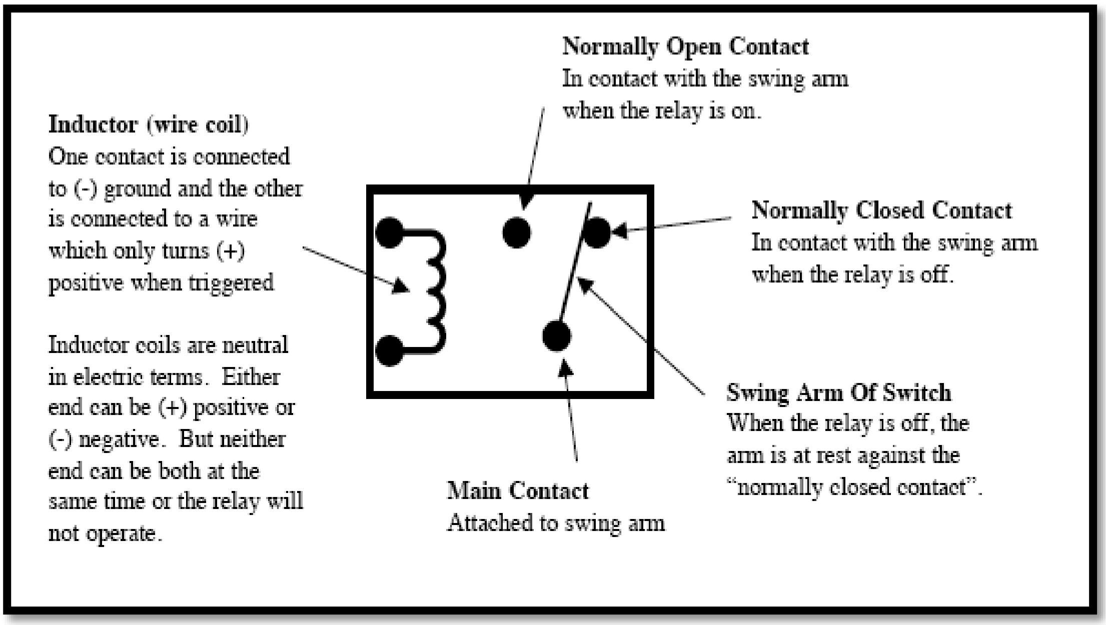


Figure 2.3: Relay pins

Relay Construction:

Relays are amazingly simple devices. There are four parts in every relay:

- Electromagnet
- Armature : that can be attracted by the electromagnet
- Spring
- Set of electrical contacts

2.3.3Language programmable supported from raspberry:

The Raspberry Pi was designed to encourage young people to learn to how to code the Pi in Raspberry Pi even comes from the Python programming language, so the very idea of programming is written into the name of the computer itself.

In the short time that the Raspberry Pi has been around, in [8] a considerable number of programming languages have been adapted for the Raspberry Pi, either by the creator of the language, who wanted to support the Pi by porting their creation, or by enthusiastic users who wanted to see their language of choice available on their platform of choice.

Here's a quick rundown of some of the languages now available for programming on the raspberry:

- Scratch

Scratch is an entry-level programming language that comes as standard with the Raspberry Pi distribution, Raspbian. Scratch was originally created by the Lifelong Kindergarten Group at the MIT Media Lab in Boston, U.S., with an aim to help young people learn mathematical and computational concepts while having fun making things

.

- Python

Python is one of the primary programming languages hosted on the Raspberry Pi. Did you know that Python is named after Monty Python's Flying Circus, the comedy team who brought us Life of Brian? (Which means Raspberry Pi is indirectly named after Monty Python, too) .

References to the comedy show are encouraged in the documentation and examples. Guido Van Resume, the Dutch programmer who created Python, was a big Monty Python fan. Python's supporters have given Guido the title of Benevolent Dictator for Life. Great title, eh

- HTML5

HTML is the mark-up language that makes the World Wide Web tick. It was devised by Tim Berners-Lee while he was working at CERN in Geneva as a means to allow scientists in the organization to share their documents with each other. Before long, it went global.

HTML is the primary building block of the Internet — it tells your browser how to lay out each web page, and lets one website link to another. The latest version is HTML5. Through its radical redesign, it's made embedding videos or audio into webpages or writing apps that will run on any smartphone or tablet easy.

- JavaScript

JavaScript is a scripting language that works alongside HTML to add interactivity to websites. JavaScript was invented, and is maintained by, the World Wide Web Consortium, which also looks after HTML and CSS.

JavaScript adds client-side scripting to web browsers, which means you can create rollover buttons and drop-down menus and do calculations and a million other things. It got a new lease of life when it was combined with XML to become AJAX, which was adopted by companies such as Google and Yahoo! to improve the usability of their online maps, among other things

.

- JQuery

. JQuery is the most popular JavaScript library. It runs on any browser, and it makes the scripting of HTML considerably simpler. With jQuery, you can create rich web interfaces and interactive components with just a small amount of JavaScript knowledge

- Java

When Java arrived on the scene, it was greeted with open arms by developers as the first programming language with which you could write a program that would run on any operating system, Windows machines and UNIX boxes alike, without having to re-write the code.

This was a great leap forward. No longer did developers have to write in different languages for each operating system, or compile different iterations for every computer they wanted their code to run on. They could simply compile the code one time and it would run anywhere.

It was originally designed for Interactive TV by its creators, James Gosling, Mike Sheridan, and Patrick Naught on, and is named after the Java coffee that the creators consumed in quantity.

- C programming language

The C Programming language was written by Dennis Ritchie, using Brian Kernighan's B language as its model. C is one of the most widely used languages in the world, utilized in everything from complete operating systems to simple programming languages. Linux, the operating system that runs the Raspberry Pi, is largely written in C and is built into all Linux and Unix systems..

The design for C influenced a great many other programming languages, including Python, Java, JavaScript, and a programming language called D. It was also extended as Objective C, which is the language used to write apps for iPhones and iPads.

.

- C++

C++ was developed by the Danish developer Bjarne Stroustrup as a way to enhance C. C++ is used in a million different circumstances, including hardware design, embedded software (in mobile phones, for example), graphical applications, and programming video games. C++ adds object-oriented features to C. Other object-oriented languages are Java, Smalltalk, Ruby, and .Net.

2.4 Android

Android is an operating system based on the Linux kernel, and designed primarily for touchscreen mobile devices such as smartphones and tablet computers.

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices. Despite being primarily designed for phones and tablets, it also has been used in televisions, games consoles, digital cameras, and other electronics. Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, [9] which add new features for advanced users or bring Android to devices which were officially released running other operating systems.

In this project we decided to use android to build android application to receive streaming live video and to control vehicles.

Android Features

- Screen capture

  Android supports capturing a screenshot by pressing the power and volume-down buttons at the same time. Prior to Android 4.0, the only methods of capturing a screenshot were through manufacturer and third-party customizations or otherwise by using a PC connection (DDMS developer's tool). These alternative methods are still available with the latest Android

- Media support

- for common audio, video, and image formats (MPEG4 ،H.264, MP3, AAC, AMR, JPG, PNG, GI)

- Open source.
- Based on Linux kernel.
- Hardware dependent Camera, GPS, compass, and accelerometer hardware dependent.
- Technology dependent Bluetooth, EDGE, 3G, and Wi-Fi.
- Integrated browser based on the open source Web Kit engine.

2.5 Streaming live video

There are a lot kinds of streaming:

- Streaming audio or video to iPhone, iPod touch, iPad, or Apple TV
- Streaming live events without special server software
- Sending video on demand with encryption and authentication

2.5.1 HTTP (Hyper Text Transfer Protocol)

Streaming video is content sent in compressed form over the Internet and displayed by the viewer in real time. With streaming video or streaming media, [10] a Web user does not have to wait to download a file to play it. Instead, the media is sent in a continuous stream of data and is played as it arrives. The user needs a player, which is a special program that uncompressed and sends video data to the display and audio data to speakers.

Hypertext Transfer Protocol (HTTP) Live Streaming lets us send audio and video over HTTP from an ordinary web server for playback on IOS-based devices including iPhone, iPad, iPod touch, and Apple TV and on desktop computers, its supports both live broadcasts and prerecorded content (video on demand).

HTTP Live streaming supports multiple alternate streams at different bit rates, and the client software can switch streams intelligently as network bandwidth changes and HTTP Live streaming also provides for media encryption and user authentication over HTTPS, allowing publishers to protect their work.

Conceptually, HTTP Live Streaming consists of three parts: the server component, the distribution component, and the client software.

- The server component is responsible for taking input streams of media and encoding them digitally, encapsulating them in a format suitable for delivery, and preparing the encapsulated media for distribution.

- The distribution component consists of standard web servers. They are responsible for accepting client requests and delivering prepared media and associated resources to the client. For large-scale distribution, edge networks or other content delivery networks can also be used.

- The client software is responsible for determining the appropriate media to request, downloading those resources, and then reassembling them so that the media can be presented to the user in a continuous stream. Client software is included on IOS 3.0 and later and computers with Safari 4.0 or later installed.

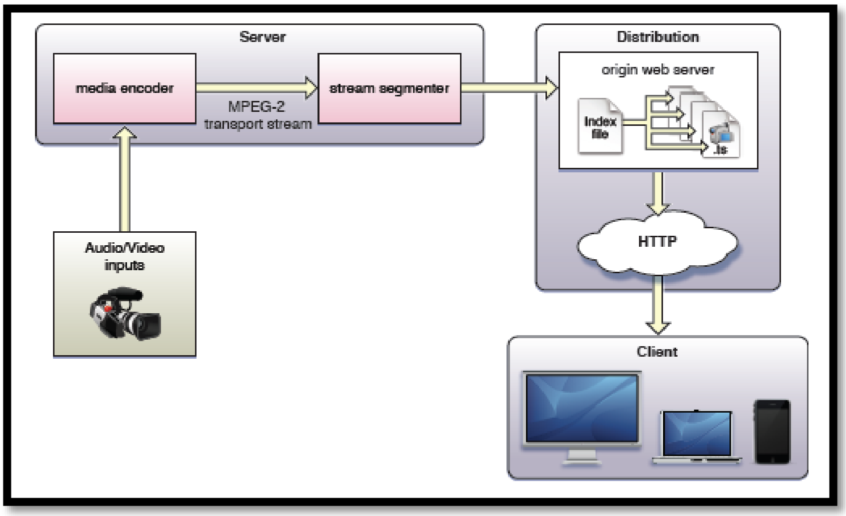


Figure 2.4: video streaming stages

2.5.2 RTSP (Real Time Streaming Protocol)

The Real-Time Streaming Protocol (RTSP) establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It does not typically deliver the continuous streams itself, [11] although interleaving of the continuous media stream with the control stream is possible. In other words, RTSP acts as a "network remote control" for multimedia servers.

RTSP takes advantage of streaming which breaks data into many packets sized according to the bandwidth available between client and server. When enough packets have been received by the client, the user's software can be playing one packet, decompressing another and downloading the third. The user is able to start listening almost immediately without having to get the entire media file. Both live data feeds and stored clips can be the sources of data.

The Real Time Streaming Protocol is more of a framework than a protocol. It's meant to control multiple data delivery sessions, provide a way to choose delivery channels such as UDP, TCP and IP-multicast. The delivery mechanisms are based solely on RTP.

RTSP has been designed to be on top of RTP to both control and deliver real time content. Thus RTSP implementations will be able to take advantage of RTP improvements, such as RTP header compression. Although RTSP can be used with unicast, its use might help to smoothen the change from unicast to IP multicasting with RTP. Real Time Streaming Protocol can also be used with RSVP to set up and manage reserved-bandwidth streaming sessions.

Differences between RTSP and HTTP

The RTSP is intentionally similar in syntax and operation to HTTP. However, it differs in a number of important aspects from HTTP:

- RTSP introduces a number of new methods and has a different protocol identifier.
- An RTSP server needs to maintain state by default in almost all cases, as opposed to the stateless nature of HTTP.
- Both an RTSP server and client can issue requests.
- Data is carried out-of-band by a different protocol. (There is an exception to this.)
- RTSP is defined to use ISO 10646 (UTF-8) rather than ISO 8859-1, consistent with current HTML internationalization efforts.
- The Request-URI always contains the absolute URI.

Chapter 3

Technologies and Methodology

3.1 Introduction

3.2 Hardware

3.2.1 Raspberry

3.2.2 Raspberry pi camera

3.2.3 Multi-Chassis

3.2.4 TP-Link

3.2.5 GPIO cable

3.2.6 Relay

3.3 Software

3.3.1 LINUX system operation

3.3.2 Eclipse program

3.3.3 Python

3.1 Introduction

This chapter provides an overview for technologies used to implement all parts of our project. The set of technologies includes hardware components and programs written by several programming languages suitable for running project operations.

3.2 Hardware

First, we start with the hardware components:

3.2.1 Raspberry

The Raspberry Pi is a single-board computer developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It’s a capable little PC which can be used for many of the things that your desktop PC does, like spreadsheets, word-processing and games [8]. It also plays high-definition video. The design is based around a Broadcom BCM2835SoC, which includes an ARM1176JZF-S 800 MHz processor, Video Core IV GPU, and 256 or 512 Megabytes of RAM. The design does not include a built-in hard disk or solid-state drive, instead relying on an SD card for booting and long-term storage. This board is intended to run Linux kernel based operating systems. The foundation has released two versions, Model A & Model B.

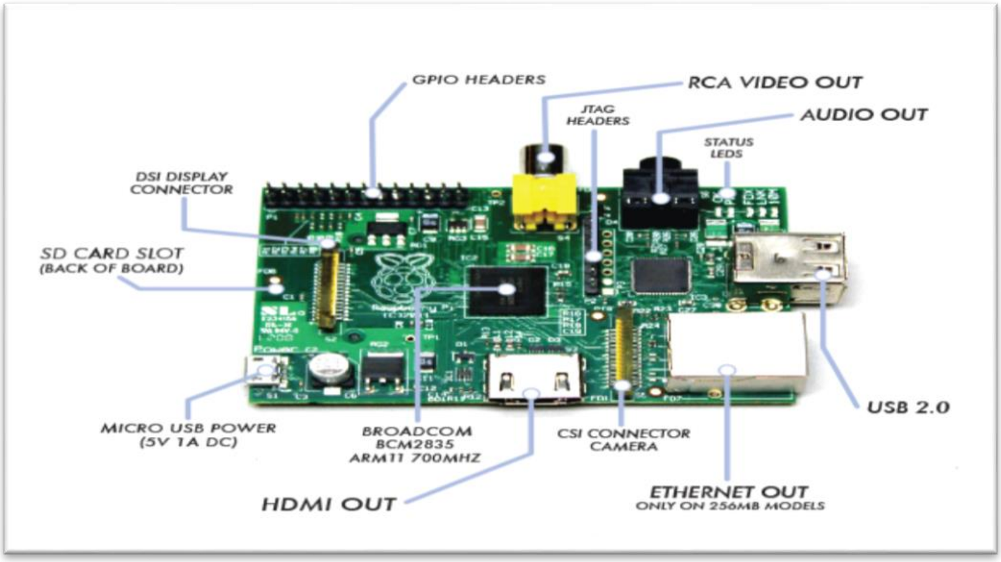


Figure 3.1: raspberry pi device

- **Model B (Revision 2.0)**: It's a latest launched Raspberry Pi Revision 2 with 512MB of RAM memory, two USB ports and a 10/100 Ethernet controller.
- **Model B (Revision 1.0)**: has 256MB RAM memory, two USB ports and a 10/100 Ethernet controller.
- **Model A** - has 256 Megabytes (MB) RAM memory, one USB port and no Ethernet controller. Though the Model A doesn't have an RJ45 Ethernet port, it can connect to a network by using a user supplied USB Ethernet or Wi-Fi adapter. As typical of modern computers, generic USB keyboards and mice are compatible with the Raspberry Pi.

The Raspberry Pi use Linux-kernel based operating systems. Debian GNU/Linux, Ice weasel. The Raspberry Pi does not come with a real-time clock, so an OS must use a network time server, or ask the user for time information at boot time to get access to time and date info for file time and date stamping. However a real time clock (such as the DS1307) with battery backup can be easily added via the I2C interface.

The Raspberry Pi Foundation has released various SD Card images that can be loaded onto an SD Card to produce a preliminary operating system. The image is based upon Linux version of Debian OS (Raspbian) with the LXDE desktop and the Midori browser, plus various programming tools. The image can also run on QEMU allowing the Raspberry Pi to be emulated on various other platforms.

Raspberry Specifications

Raspberry Pi model B microcontroller will be used. It has the following specifications.

- System on a chip: Broadcom BCM2835
- CPU: 700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set).
- GPU: Broadcom Video Core IV @ 250 MHz, OpenGL ES 2.0 (24 GFLOPS), MPEG-2 and VC-1.
- Memory (SDRAM): 512 MB (shared with GPU).
- USB 2.0 ports: 2 (via the built in integrated 3-port USB hub).
- Video input: A CSI input connector allows for the connection of a camera module, USB camera.
- Video outputs: Composite video, HDMI (not at the same time).
- Audio outputs: 3.5 mm jack, HDMI.
- Onboard storage: SD / MMC / SDIO card slot.
- Onboard network: 10/100 wired Ethernet RJ45.
- Low-level peripherals: General Purpose Input/output (GPIO) pins, Serial Peripheral Interface Bus (SPI), Universal asynchronous receiver/transmitter (UART).
- Power source: 5V and 700 mA via Micro USB or GPIO Header
- Power ratings: 700mA, (3.5 W).

Raspberry ports

1. Broadcom BCM2835: This chip represents an integrated computer it contains the central processing unit (CPU), random access memory (RAM) and graphic processing unit (GPU).
2. High-Definition Multimedia Interface Output (HDMI):The output port used in connection raspberry with high quality of television screens with the knowledge that this port out both picture and audio to television screen, shown in figure 3.3.
3. Radio Corporation of America video output (RCA): The same as HDMI but using with old televisions.
4. Audio out (jack): Audio output port with size 3.5mm, size that used in for most types of different speakers.
5. Ethernet port: port using to connect raspberry with computer networks and internet.
6. Universal Serial Bus port (USB): as in computers used to connect keyboards mouse and Wi-Fi adapters.
7. Micro USB ports: used as input voltage port to support raspberry with 5 volt.
8. SD card slot: the place to installing memory that contains operating system and all of needed files.
9. GPIO pins: Electronic control group outlets that makes raspberry work as microcontrollers.
10. CSI camera input: the place to connect a high fineness camera manufactured specially for raspberry.
11. DSI display: used to connect it with touch screen as in smartphone.

3.2.2 Raspberry pi camera

The Raspberry Pi Camera Module is a 5MP CMOS camera with a fixed focus lens that is capable of capturing still images as well as high definition video. Stills are captured at a resolution of 2592 x 1944, while video is supported at 1080p at 30 FPS, 720p at 60 FPS and 640x480 at 60 or 90 FPS.

The camera module measures in at just 25mm x 20mm x 9mm and weighs a mere 3 grams. This makes it ideal for projects such as hidden security cameras, high altitude balloon experiments, and even an on board camera for RC car adventures. The camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system.

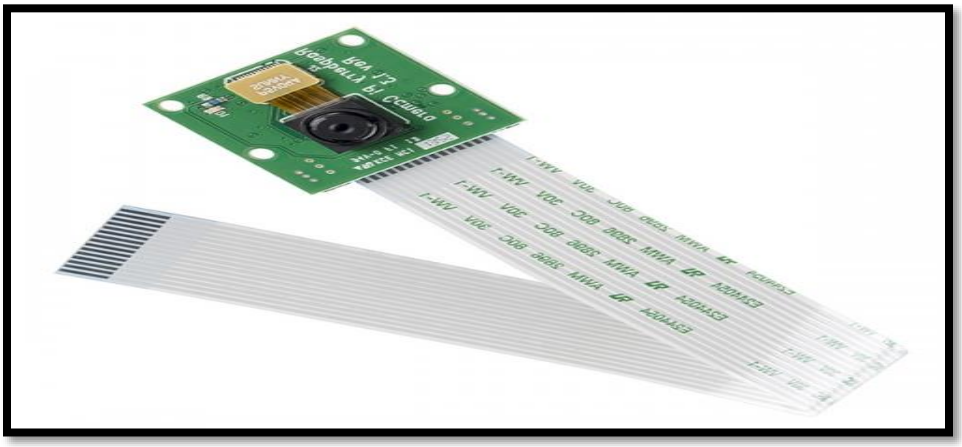


Figure 3.2: Raspberry pi camera.

Specification

- 1.4 µm X 1.4 µm pixel with Omni BSI technology for high performance (high sensitivity, low crosstalk, low noise)
- Optical size of 1/4"
- Automatic image control functions:
- Automatic exposure control (AEC)
- Automatic white balance (AWB)
- Automatic band filter (ABF)
- Automatic 50/60 Hz luminance detection
- Automatic black level calibration (ABLC)
- Programmable controls for frame rate , AEC/AGC 16-zone size/position/weight control, mirror and flip, cropping, windowing, and panning
- Digital video port (DVP) parallel output interface

- 32 bytes of embedded One-Time Programmable (OTP) memory

3.2.3 Multi-Chassis

An easy to assemble and use robot chassis platform (stand). The Multi-Chassis kit provides everything that we need to give our robot a rugged tank tread platform with plenty of area for expansion to add controllers and raspberry pi board.

This Multi-Chassis Tank Kit includes two 48:1 DC gearboxes with a metal final output gear and shaft that independently drive each 52mm tank treads making this chassis able to drive around tough terrain like tall grass and sand. Each side of the 2.5mm thick aluminum frame comes cut with plenty of attachment points to add a multitude of different robot controllers, drivers, and sensors. A standard size servo can be mounted inside the chassis and be used for robotic arms or sensor arrays.

Figure3.3: Multi-Chassis tank kit

Dimensions:

- Completed Kit Size - 157L x 149W x 60H mm.
- Wheel Diameter - 52 mm.

3.2.4 TP-Link

Wireless N USB Adapter TL-WN722N allows to connect a desktop or notebook computer to a wireless network and access high-speed Internet connection. Complies with IEEE 802.11n, they provide wireless speed up to 150Mbps, which is beneficial for the online gaming or even video streaming. Also, wireless security encryption could be established simply at a push of Quick Setup Security (QSS) button, preventing the network from outside threats It provides Wi-Fi Protected Access (WPA) and Wi-Fi Protected Access II (WPA2) encryptions, that are created by the Wi-Fi Alliance industry group, promoting interpretabilities and security for WLAN.

TL-WN722N shows more excellent abilities of mitigating data loss over long distances and through obstacles in a small office or a large apartment, even in a steel-and-concrete building. It delivers performance enhancements, allowing user to sharing files and watching streaming media. Which we need in this project. Clear Channel Assessment (CCA) automatically avoids channel collision using its clear channel selection feature and fully realizes the advantages of channel binding, greatly enhanced the wireless performance.

It offers 4dBi (decibel isotropic) high gain external antenna that can be rotated and adjusted in different directions to fit various operating environments, and can bring better performance than the internal antenna.

TL-WN722N performance:

Table 3.1: TL-WN722N performance

WIRELESS FEATURES	
Wireless Standards	**IEEE 802.11n, IEEE 802.11g, IEEE 802.11b**
Frequency	**2.400-2.4835GHz**
Signal Rate	**11n: Up to 150Mbps(dynamic)** **11g: Up to 54Mbps(dynamic)** **11b: Up to 11Mbps(dynamic)**
EIRP	**<20dBm**
Reception Sensitivity	**130M: -68dBm@10% PER** **108M: -68dBm@10% PER** **54M: -68dBm@10% PER** **11M: -85dBm@8% PER** **6M: -88dBm@10% PER** **1M: -90dBm@8% PER**
Wireless Modes	**Ad-Hoc / Infrastructure mode**
Wireless Security	**Support 64/128 bit WEP, WPA-PSK/WPA2-PSK**
Modulation Technology	**DBPSK, DQPSK, CCK, OFDM, 16-QAM, 64-QAM**
Advanced Functions	**WMM, PSP X-LINK(For Windows XP), Roaming**
Antenna Gain	**4dBi**

Figure 3.4: TP-link WN722N

3.2.5 GPIO cable

The "General Purpose Input and Output" pins, that known as GPIO. One feature that has contributed to the Raspberry Pi's success is the possibility of interfacing the software with the physical world. They are available on the Pi at the P1 connector. We will use a 15 cm GPIO cable to connected raspberry pi with motor control circuit. As shown in figure.

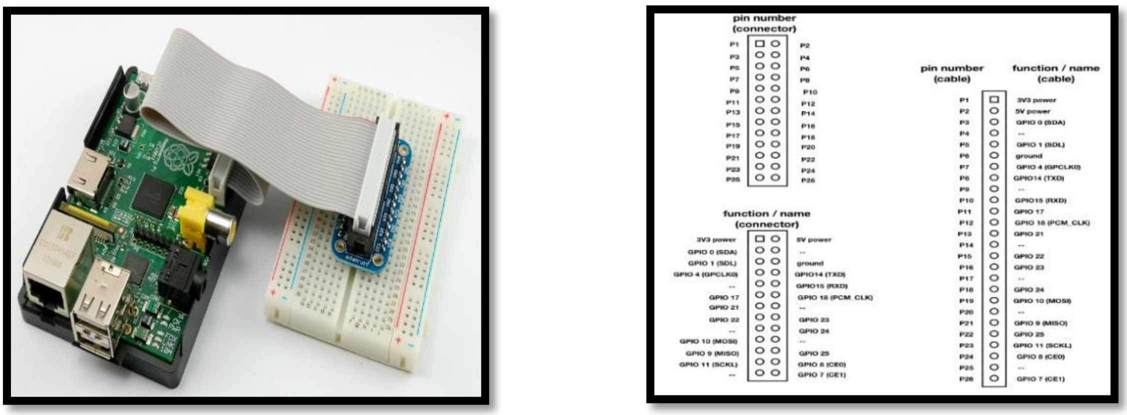


Figure 3.5: GPIO cable and pins.

3.2.6 Relay

Relay is one of the most important electromechanical devices highly used in industrial applications specifically in automation. A relay is used for electronic to electrical interfacing i.e. it is used to switch on or off electrical circuits operating at high AC voltage using a low DC control voltage. A relay generally has two parts, a coil which operates at the rated DC voltage and a mechanically movable switch. The electronic and electrical circuits are electrically isolated but magnetically connected to each other, hence any fault on either side does not affects the other side.

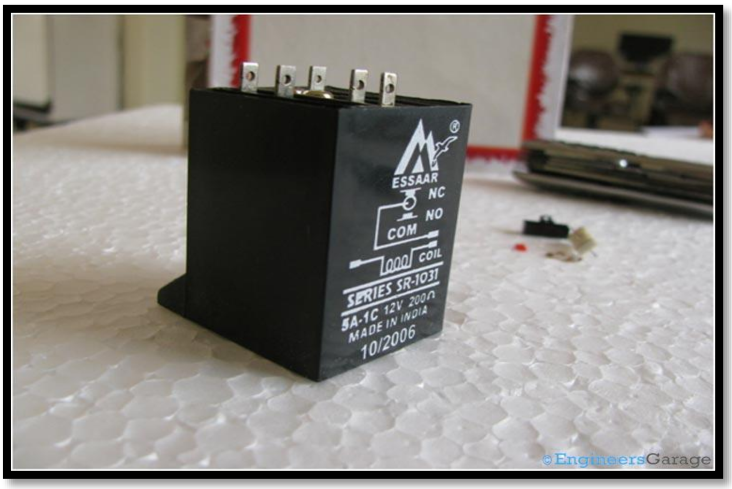


Figure 3.6: Relay switches.

Relay switch shown in the image above consists of five terminals. Two terminals are used to give the input DC voltage also known as the operating voltage of the relay. Relay are available in different operating voltages like 6V, 12V, 24V etc. The rest of the three terminals are used to connect the high voltage AC circuit. The terminals are called Common, Normally Open (NO) and Normally Closed (NC). Relays are available in various types & categories and in order to identify the correct configuration of the output terminals, it is better to see the data sheet or manual. You can also identify the terminals using a multimeter and at times it is printed on the relay itself.

3.3 Software:

A programming language is an artificial language designed to communicate instructions to machine particularly computer. Programming languages can be used to create programs that control the behavior of a machine and / or to express algorithm precisely. There are a lot of programming language that is used to build out a computer program, where choosing any language depends on the tasks of the program.

3.3.1 LINUX system operation:

Is a free and open source software operating system for computers the operating system is a collection of the basic instructions that tell the electronic parts of the computer what to do and how to work. There is a lot of software for Linux, and since Linux is free software it means that none of the software will put any license restrictions on users. This is one of the reasons why many people like to use Linux.

Linux is, in simplest terms, an operating system. It is the software on a computer that enables applications and the computer operator to access the devices on the computer to perform desired functions. The operating system (OS) relays instructions from an application to, for instance, the computer's processor. The processor performs the instructed task, then sends the results back to the application via the operating system, in this project Linux is the operating system of the raspberry pi that help the raspberry to operate well.

Facts about Linux

- Linux is multi-platform:

  This means that by choosing Linux, you choose aportable operating system on which you can run or develop portable applications.

- Linux is multi-user:

  This means that you have the freedom to share the same computer among various tasks launched by various users.

- Linux is open source:

  This means you have access to the code of the operating system. You can actually learn computer science, change the code and experience the result. You can even participate to the development code.

- Linux is cheap:

  Linux, as operating system, is and will always remain free (in the sense of a free beer). There is no license to pay to anybody. More than that, Linux cost of ownership is very low.

3.3.2 Eclipse program:

In the context of computing, Eclipse is an integrated development environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins.

Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PYDEY is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used an IDE to develop Scala applications and PHP eclipse is a plug-in to eclipse that provides complete development tool for PHP.

In this project we will based on eclipse program in order to build android application that include control button and window for capture video.

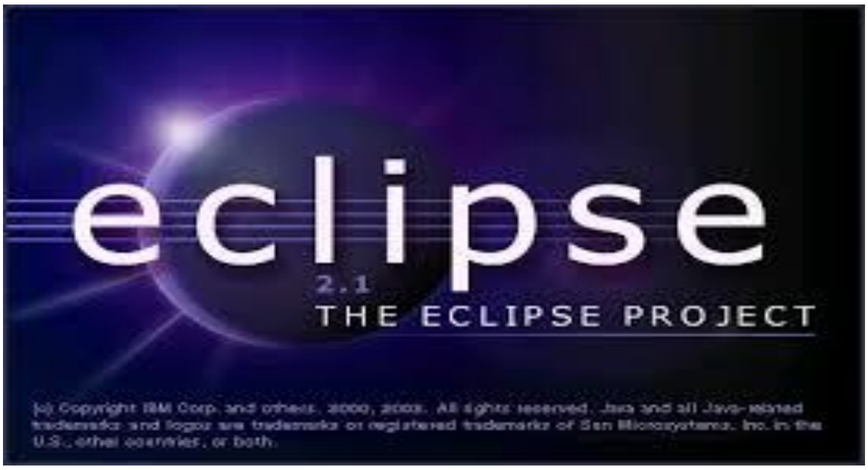


Figure 3.7: Eclipse program

Java language

Java is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or applet for use as part of a Web page.

Characteristics of Java:

- Relative to C++, Java is easier to learn.
- In addition to being executed at the client rather than the server, a Java applet has other characteristics designed to make it run fast.
- Java is object-oriented, which means that, among other characteristics, an object can take advantage of being part of a class of objects and inherit code that is common to the class.
- Java is object-oriented, which means that, among other characteristics, an object can take advantage of being part of a class of objects and inherit code that is common to the class.

3.3.3 Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python was designed to be highly readable which uses English keywords frequently where as other languages use punctuation and it has fewer syntactical constructions than other languages, this language will be used for programming GPIO pins in raspberry pi.

Python's language feature highlights include:

- Easy-to-learn: Python has relatively few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language in a relatively short period of time.

- Easy-to-read: Python code is much more clearly defined and visible to the eyes.

- Easy-to-maintain: Python's success is that its source code is fairly easy-to-maintain.

- A broad standard library: One of Python's greatest strengths is the bulk of the library is very portable and cross-platform compatible on UNIX, Windows and Macintosh.

- Interactive Mode: Support for an interactive mode in which you can enter results from a terminal right to the language, allowing interactive testing and debugging of snippets of code.

- Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

- Databases: Python provides interfaces to all major commercial databases.

- GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh and the X Window system of UNIX.

- Scalable: Python provides a better structure and support for large programs than shell scripting.

Why Python language:

Python is arguably the biggest programming language on the Pi right now, for good reason. It has lots of tools, is well supported and development is really fast, and it’s easy to learn. It’s very powerful and you can even create simple games and graphics with it.

Chapter 4
Design and implementation

4.1 introduction

This project is about a robot vehicle that can be controlled through any smartphone wirelessly using a Wi-Fi link and have special android application. It transmits real-time video to the controlling smartphone using the same link. The video transmitting camera mounted on the top of the vehicle.

Usually robots are controlled through a remote using an RF link which is subject to attenuation, noise and has a very limited range as well. This project features a robot that uses Wi-Fi 802.11 standard to provide good range and give us high data rates which enables good quality uninterrupted video transmission from the robot to the smartphone.

This chapter was made to give an over view on the design and block diagram of the project components which are the main parts of project report because they give the project reader the best indication about the project and work techniques.

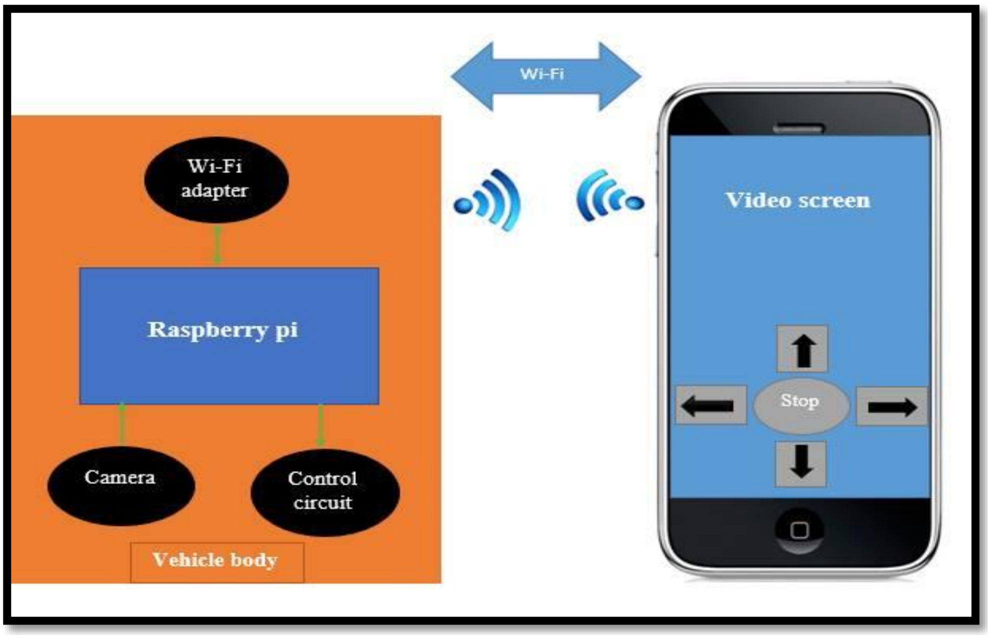


Figure 4.1: General block diagram.

4.2 General System operation

The system consists of three parts. The first part called raspberry pi consists of raspberry pi model B as the brain (PC) of the system, video camera (raspberry pi camera) and Wi-Fi adapter used for wireless communication. The second part its vehicle include Multi-Chassis tank (Vehicle body), battery and control circuit which responsible for vehicle motors control. The third part is the smartphone. This part will hold the android application which used to control the movement of the vehicle and display the captured video.

4.2.1 Raspberry pi

In this section we will describe the connections and interfaces between the raspberry and the others components .We will also describe how does this part operate. The raspberry pi is like motherboard in computer which means that all of the components in this part will be connected to it. First, the Wi-Fi adapter will be connected to the USB port, the camera to the CSI connector camera, battery to the Micro USB port and the control circuit of the motors to the GPIO pins in raspberry pi by GPIO cable. This will be described later in vehicle part. The previous description can be summarized by the figure below:

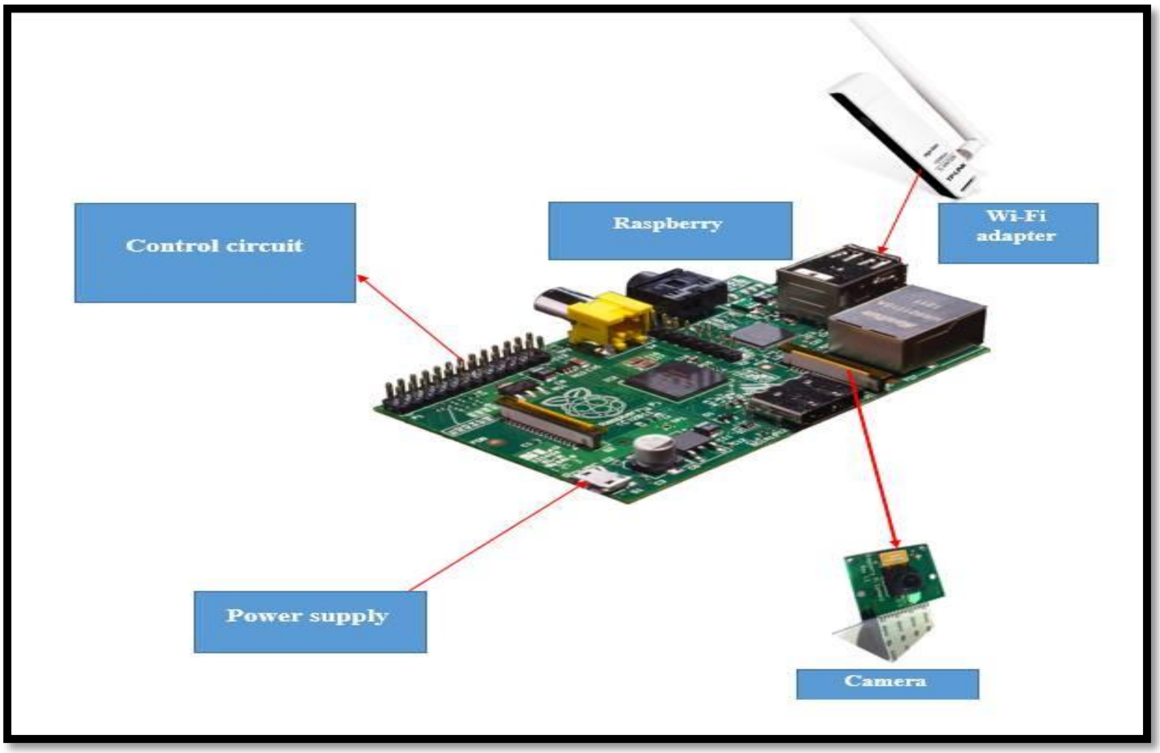


Figure 4.2: raspberry ports

Now we will describe and detail the operations that will be conducted by this part. The first step is preparing and activating the raspberry pi by installing (Linux). The next step is connecting the components to the raspberry pi and enabling raspberry pi camera, USB Wi-Fi and GPIO pins for control circuit by entering to raspberry system (Linux). After that when the raspberry pi turn on, it can be connected to the network by taking IP number and the camera operate and takes a live video (capture video). The raspberry taking video streams from camera and encoding them digitally, and encapsulating them in a format suitable for delivery, and preparing the encapsulated video for distribution video to the smartphone via Wi-Fi.

4.2.2 Vehicle

The vehicle part consist of vehicle body that include two DC motors. It has four wheels. Every two wheel are connected with each other by plastic chain, and it's important for adding controllers and raspberry pi board over it. To control motors well we will design control circuit.

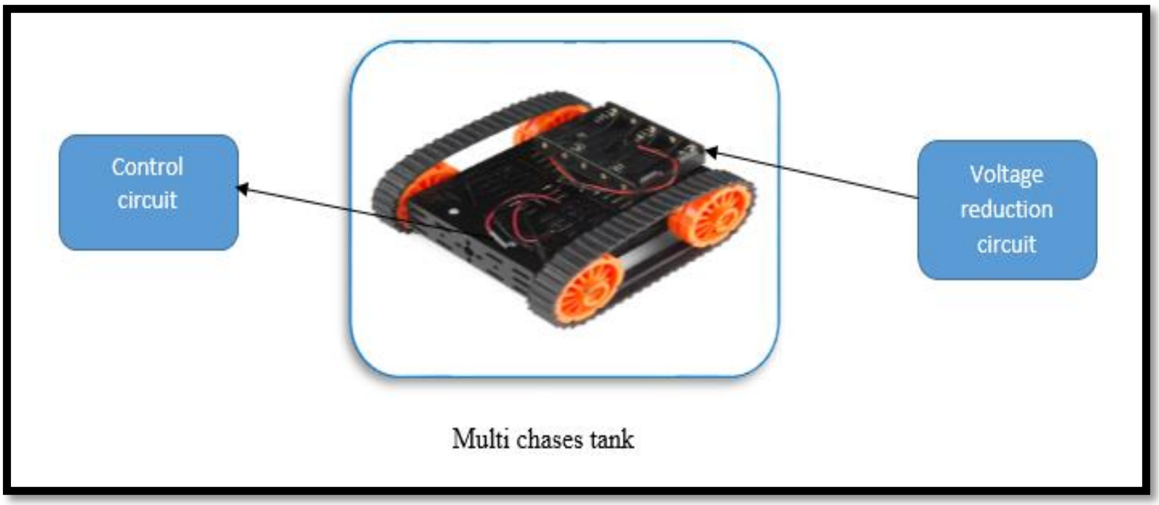


Figure 4.3: vehicle block diagram

Control Motors:

One of the most challenges in this project is controlling motors. We decided to control the vehicle in four direction (forward, revers, right and left), the vehicle have tow DC motor each motor must control to forward and revers in order to achieve control we need tow relay, two transistor and two led for each motor, the circuit below describe of the control circuit for the two motors.

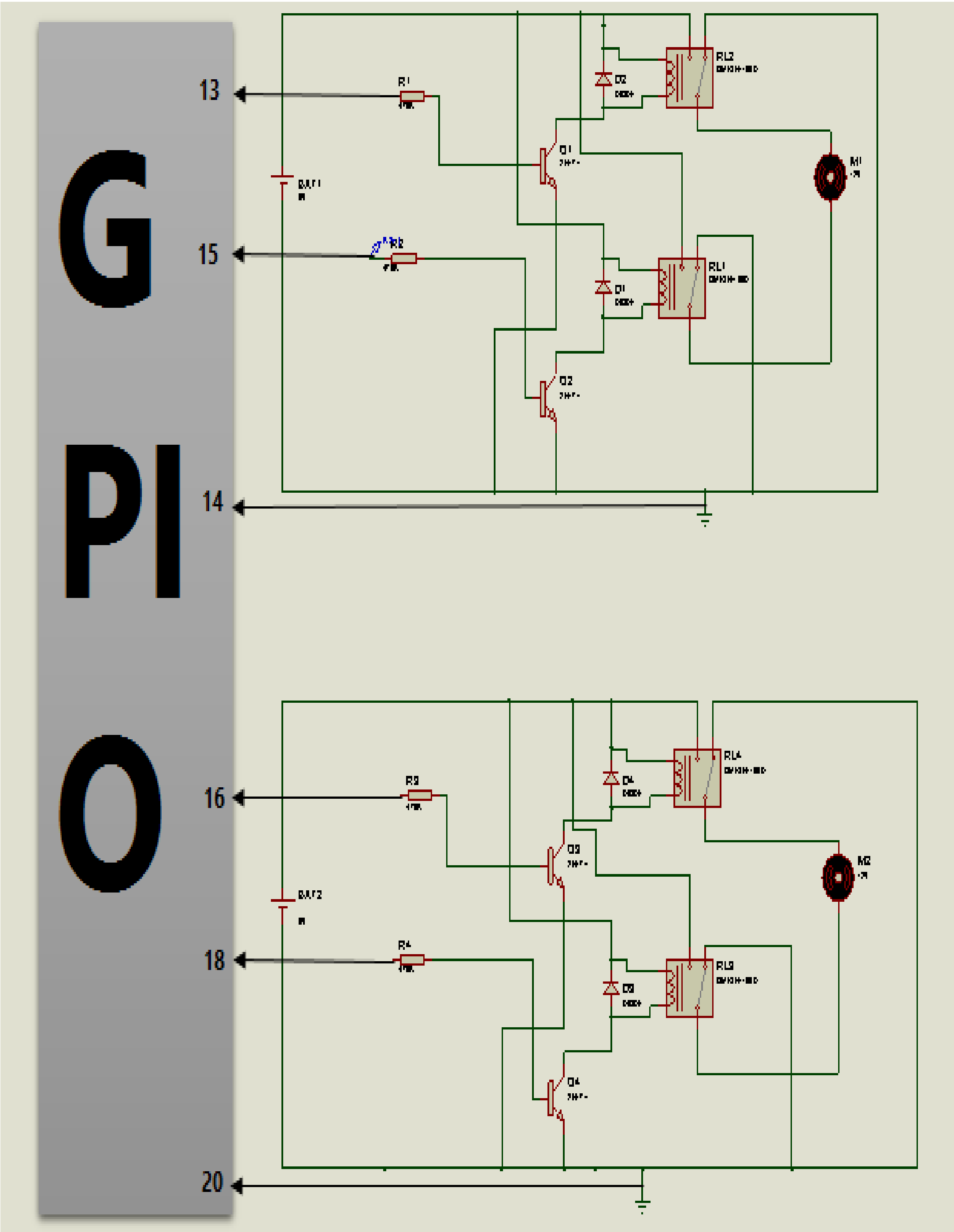


Figure 4.4: control circuit diagram

Now we will describe how does the circuit work. There are two inputs for each circuit of the both motors. The first circuit have two inputs pin (13) and pin (15) from GPIO and the second circuit have two inputs from pin (16) and pin (18).

The pins of the GPIO that connect to control circuit that will be active by programming in python language, the pins (13, 15, 16, 18) programming as output pins from the raspberry and input for control circuit.

When the pin 13 active the motor rotate forward direction and when pin 15 active the motor rotate reverse and the second motor the same if pin 16 active the motor rotate forward and activation pin 18 lead to rotate reverse direction.

Voltage Reduction:

Raspberry pi model B device needs a 5 volt power supply, but since we do not find a 5 volt battery we decided to use a 9 volt battery with regulator (78L05), to produce a 5 volt.

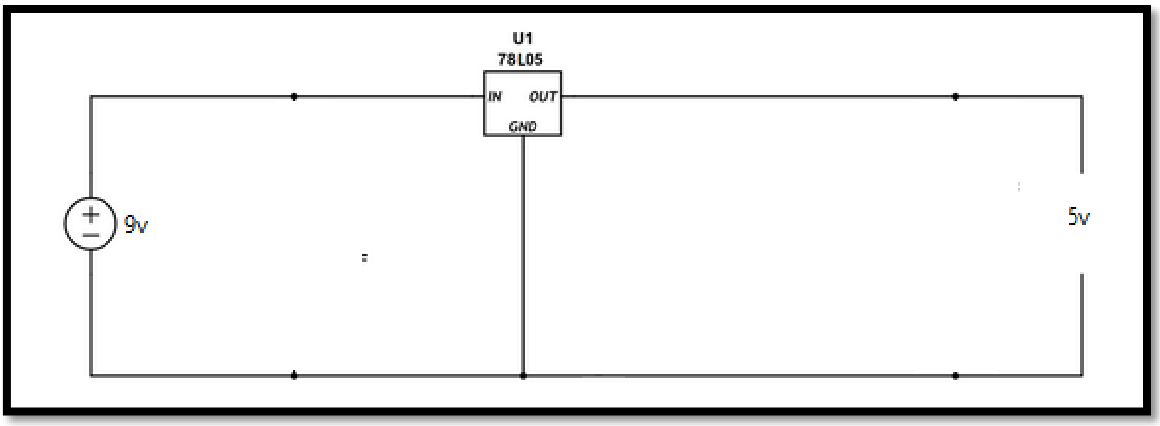


Figure 4.5: voltage reduction circuit

4.2.3 Smartphone

Smartphone device was choose as a tool to control the vehicle and display the captured video. The operation system on this device has to be android in order to install the android application on. We will use the e-clips program to design the needed android application which accepts video from raspberry pi (streaming video live) and can send commands to raspberry. By these commands we will be able to specify the direction of movement of our vehicle. The main parts of the android application will be describe below.

User interface

The user interface of the android application for this project contains five Buttons (forward, revers, right left, and stop) and small screen above the buttons to display the capture video which receive from raspberry pi. This interface will be design by Java language in (e-clips) program.

Figure 4.6: Android app. interface

Streaming video live

The real-time video streaming robot can be used in a wide range of applications including security, spying, or even to have real-time information when the robot works in the garden.

HTTP Live streaming supports multiple alternate streams at different bit rates, and the client software can switch streams intelligently as network bandwidth changes. HTTP Live streaming also provided for media encryption and user authentication over HTTPS. This allows publishers to protect their work.

In this project we can divide HTTP live streaming video to two parts:

- The server component (raspberry pi) is responsible for taking input streams capture video from raspberry pi camera and encoding them digitally and compression then encapsulating them in a format suitable for delivery, and preparing the encapsulated video for distribution it in the Wi-Fi network.

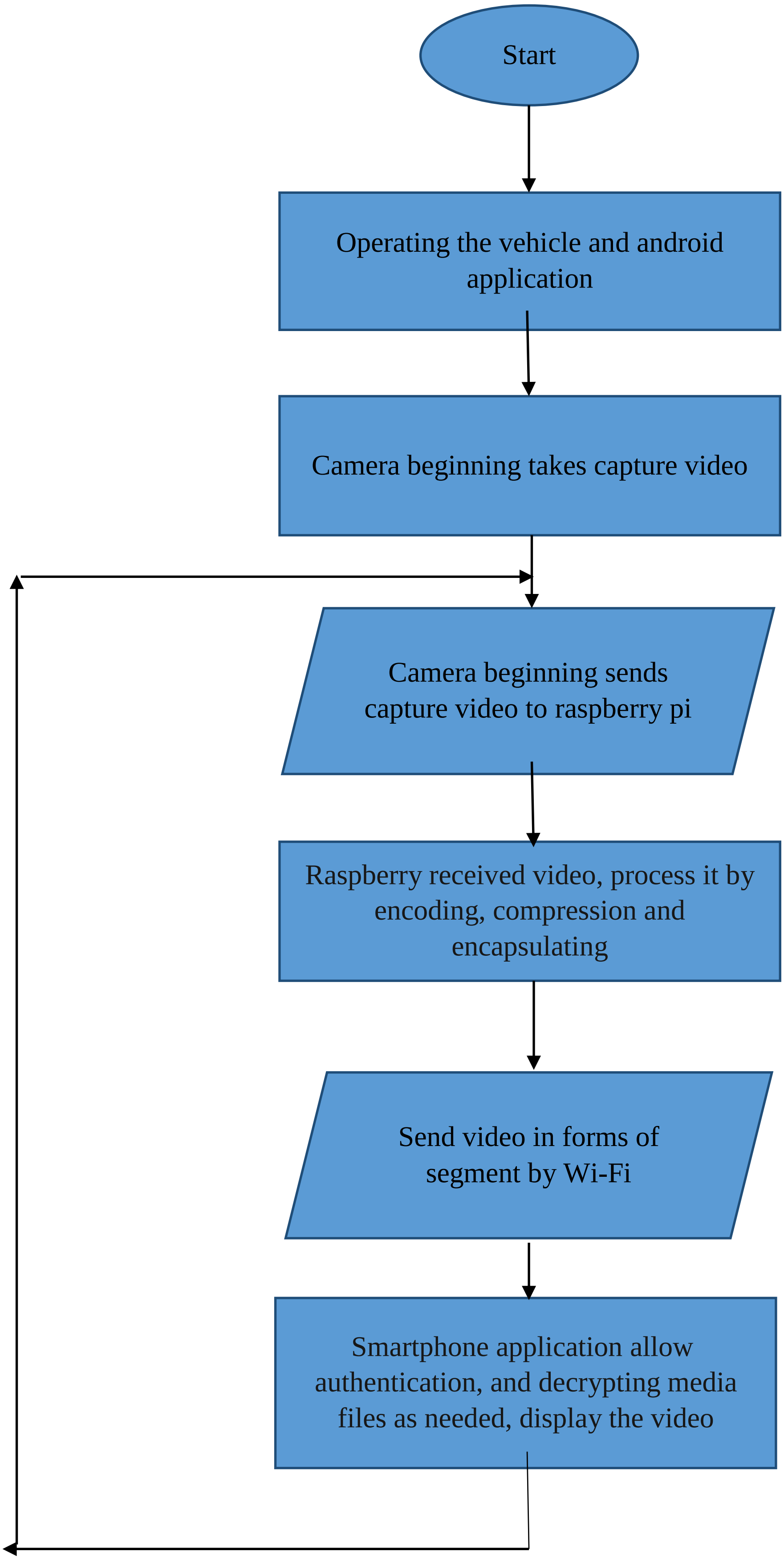
- The client part in this project is android application on a smartphone. The client is responsible for fetching any decryption keys, authenticating or presenting a user interface to allow authentication, and decrypting media files as needed.

This step is transfer capture video that takes by the raspberry pi camera from the raspberry to the smartphone (streaming video). In this step we use (HTTP) protocol because the time delay for display video is acceptable and the quality of the video is good. When the raspberry pi takes video from camera and save it in array, the video will be pulled and displayed on window screen by the android application.

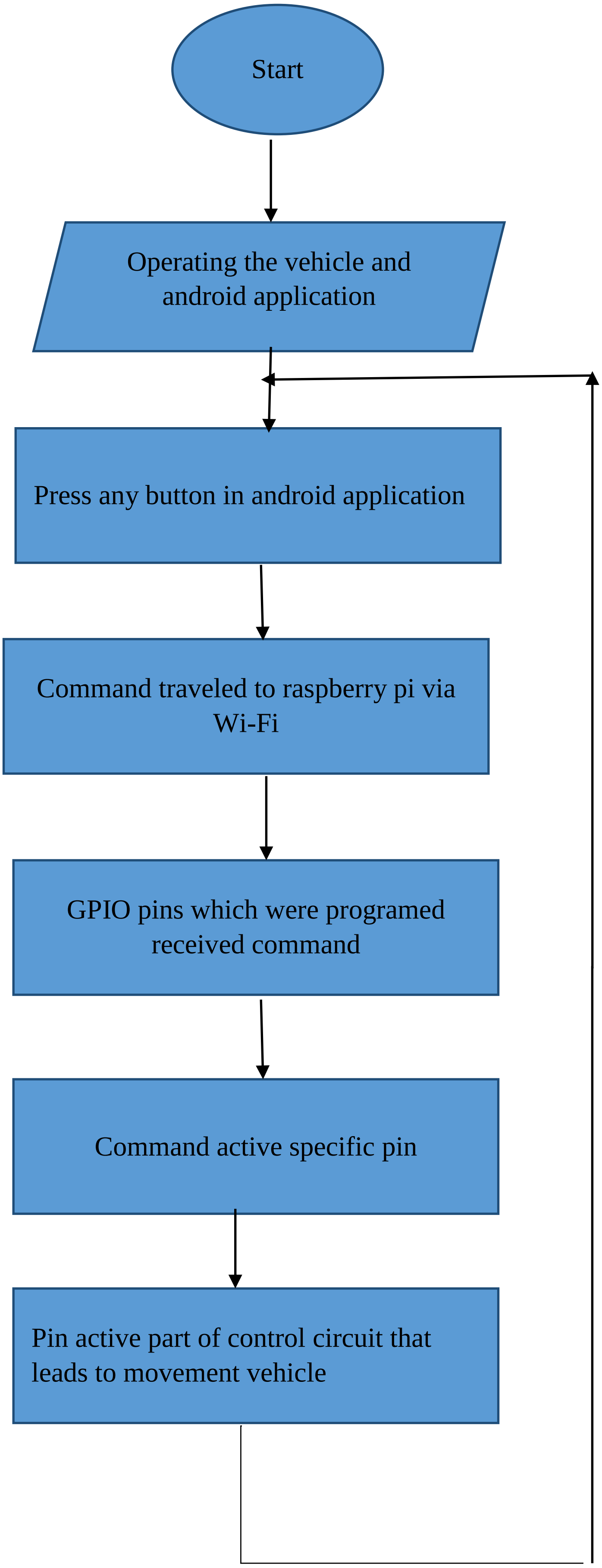
In the beginning we will test the streaming between raspberry pi and computer by VLC program. After that we will start streaming video with the android application on the smartphone.

4.3 Flowchart

The following flowchart describe the streaming video in the system:



Flowchart for controlling vehicle:



Chapter 5
Methodology

5.1 introduction

In this chapter we will explain how we worked at this project step by step. We can summarize this step as follow:

- Raspberry Pi activation.
- Enabling the Raspberry Pi ports.
- Building control circuit.
- Connection the components (USB Wi-Fi, control circuit, raspberry pi camera) to the main component (raspberry pi).
- Design android application in the smartphone to control the vehicle and receive (streaming video from the raspberry).

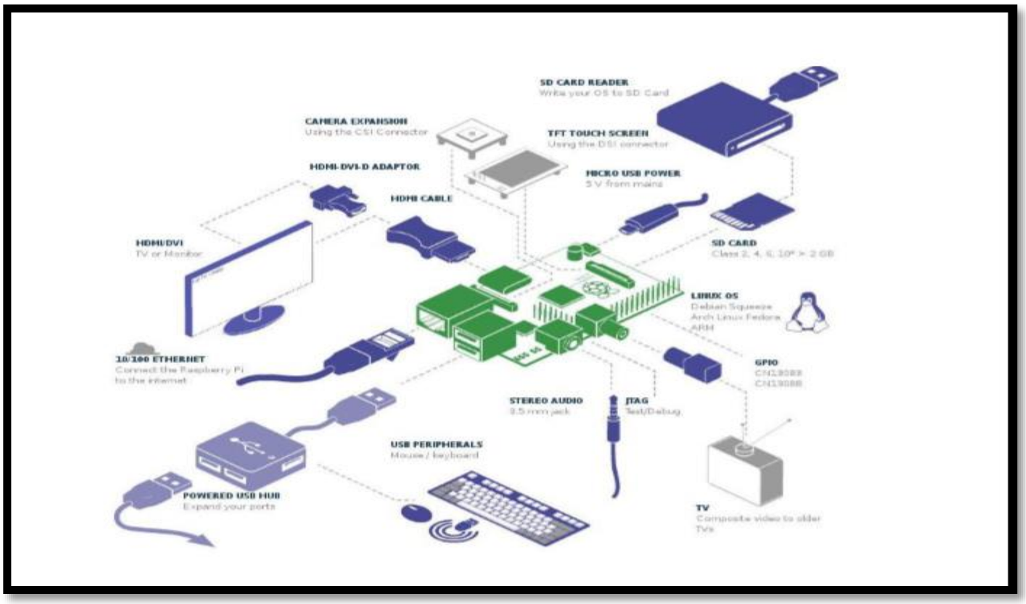


Figure5.1: description raspberry ports

5.2 methodology

In this section we will describe the way that we have followed in operating and programming the components of the project. We have divided the work to:

5.2.1 Operating Raspberry Pi

A lot of working steps needs to be finished in order to operate raspberry Pi so that we finish our work without any error. Working steps are described below:

- The operation system that raspberry pi works on is (Linux), First we store the operation system (Linux) on SD memory card. When the store process complete, we put the SD memory card in SD adapter in the raspberry. SD card size 16 GB which is enough to store all needed files to be used with raspberry.

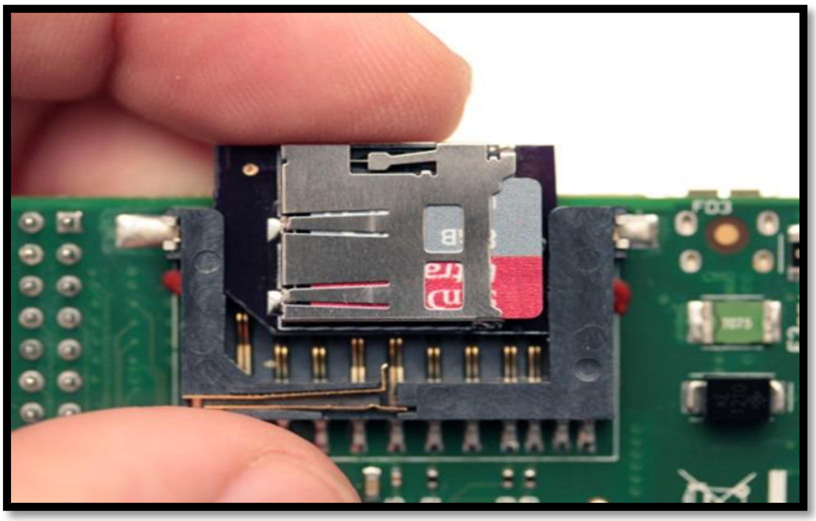


Figure 5.2: insert SD Card memory

- We supply the raspberry pi with the necessary electrical power for operation by a mobile phone charger, and we connect it to the Micro-USB port in raspberry pi. The mobile phone charger supply 5 volt and 700 mille amber or more.

Figure 5.3: power supply with ports

- Connect the keyboard and mouse to USB port to control the raspberry pi in best way.

Figure 5.4: USB ports with input tools

- Connect TV by HDMI cable to HDMI Output port to display raspberry pi desktop.

Figure 5.5: HDMI cable with screen

- Finally we plug the charger into the power jack and then the raspberry began to load system. After that a screen appears like figure below for adjusting setting of the raspberry pi.

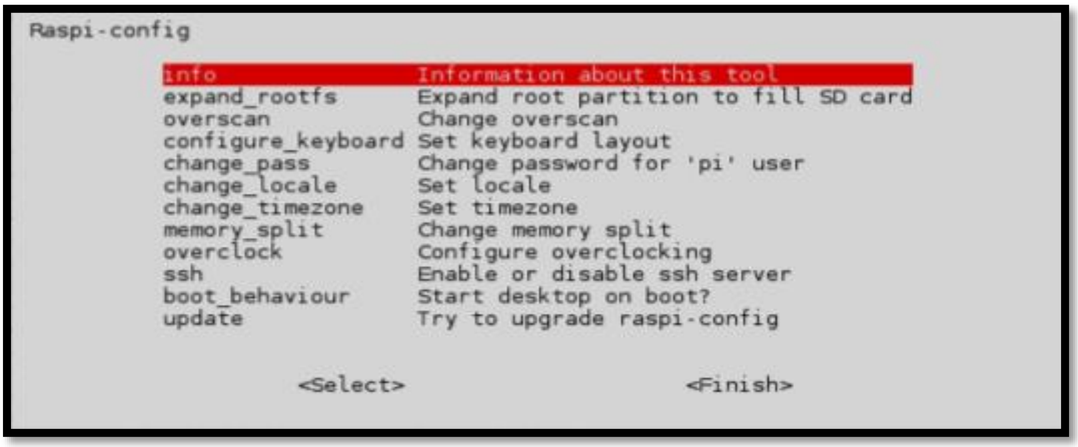


Figure 5.6: raspberry pi configuration interface

We chose the option (Expand root fs), so that the Linux operating system exploit the fully SD memory card. Then new window appears and we select (OK).

Figure 5.7: accept the expand task

After the completion of the expansion card, we will fine-tune raspberry pi to entry automatically to the desktop through select the Boot behavior option, from the main list. After that we press (OK). Then tow step the desktop appear on the TV screen and the raspberry become ready to deal with.

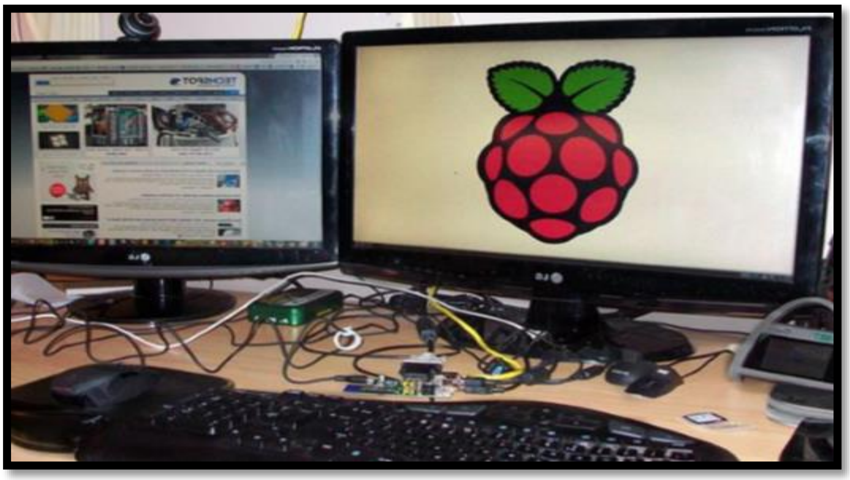


Figure 5.8: the connection of raspberry pi

5.2.2 Activation of high-resolution camera

The CSI camera module for the raspberry pi make it possible to stream high-definition video without having problem with performance. First of all we connect the raspberry pi camera to CSI Camera port.

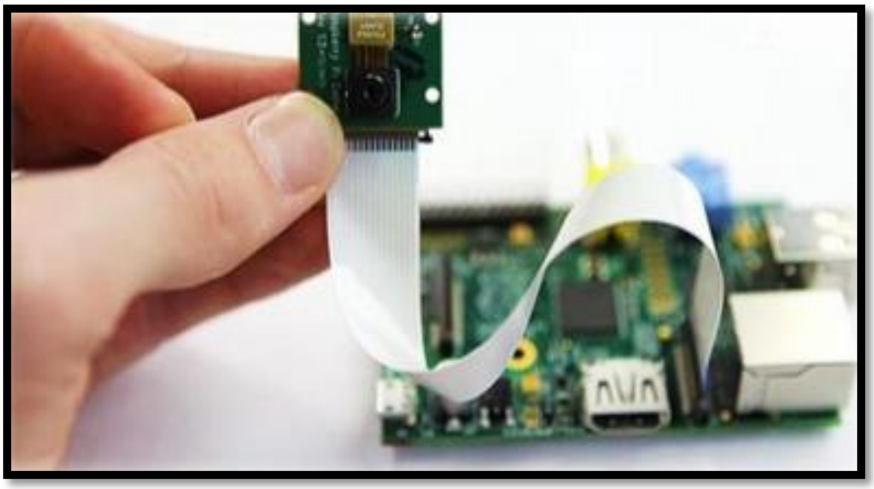


Figure 5.9: raspberry pi camera

Then we enter to (Raspi-config) by typing the command <<sudo raspi-config>>into TX terminal program, after that we enter to (Enable camera). Then we choose (Enable).

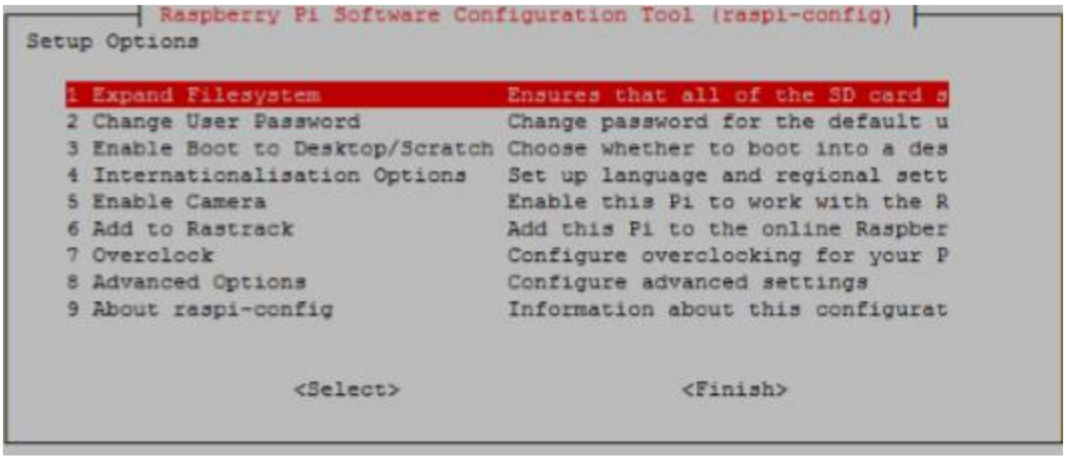


Figure 5.10: setup option

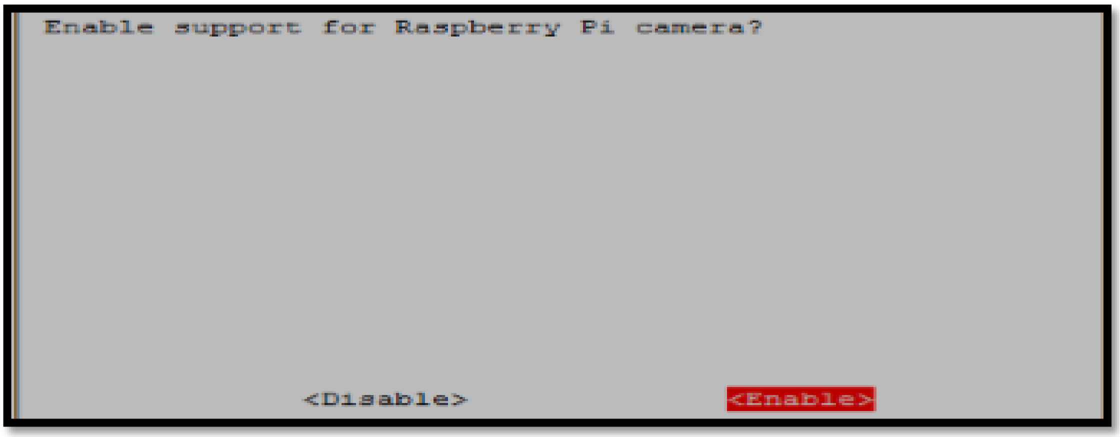


figure5.11: enabling raspberry pi camera

After these steps the camera is enabled and working in good condition, and we test it by tacking some photos by writing some commands in TX terminal program.

5.2.3 Connect raspberry pi to network (LAN)

Raspberry pi supports many wireless cards that can connect to USB port. Some of which is small size and little energy consumption. Another type is able to connect to wireless networks from a distance of 2 km. In this project we used TP-link WN722 as Wi-Fi USB adapter in order to connect the raspberry pi to the special network.

Figure 5.12: connection USB Wi-Fi with raspberry

First we connect TP-link WN722 to USB port. Then opening (Wi-Fi Config. icon) that located on the desktop. After that the interface of the wireless settings are shown as follows. Then we press scan button to search for the special network that established to work on it.

Figure 15.13: connection to network

Then when the special network appear as a result of search, we press double click on it. This results a new window to write the password of the network and the type of encryption. After that we press connect button. Finally the raspberry pi becomes connected to the special network by taking special (IP) address.

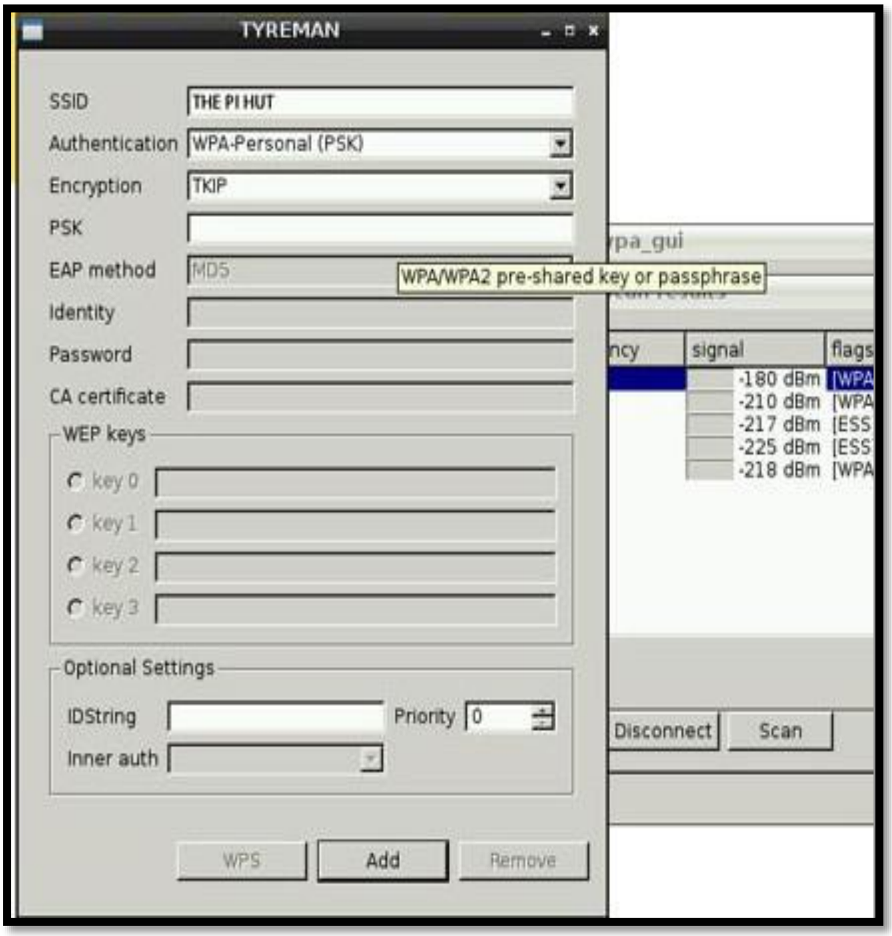


Figure 5.14: advance settings for connection to network

5.2.4 Building the multi chassis tank

The multi chassis tank contains a lot of components, we started compiled the components by using special tools to connect the components with each other ,the figure below show the components that we will compile with each other to form the final form of the multi chassis tank.:

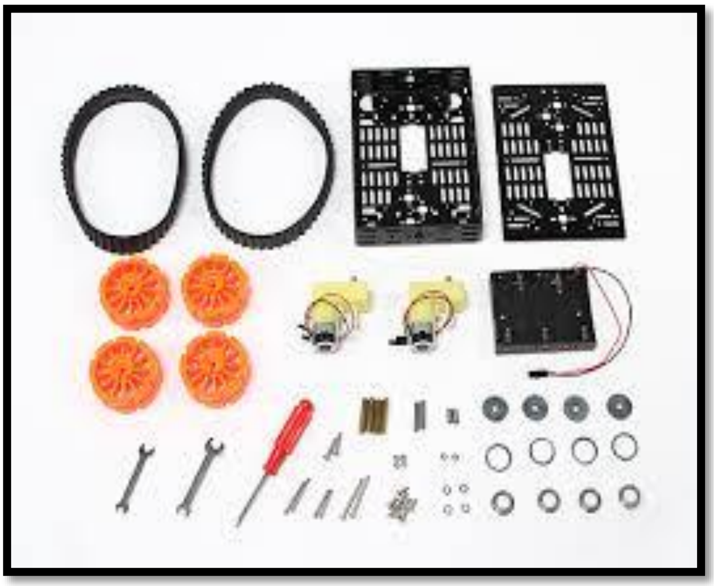


Figure 5.15: components of the multi chassis

The main parts it's the two DC motor, we connect them with the rear wheels to move the tank to four direction, and the rubber chains it's important to overcome of the difficult surfaces that faced the tank, and by helping the datasheet of the multi chassis tank that explained the methodology step by step to we reached to final form that show in the figure below:

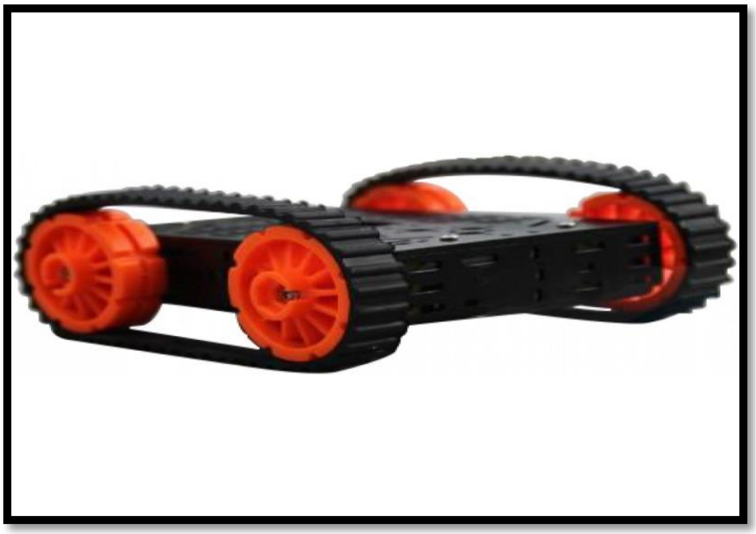


Figure 5.16: multi chassis tank after collect

We put Battery Holder and the two motor internal the box of multi chassis to exploitation the area above the tank to put the raspberry pi, control circuit, USB Wi-Fi, camera.

5.2.5 Control circuit

After we simulate control circuit by Proteus program and test it, we decided the exactly value of the components, and began built the circuit on board, this circuit Responsible for driving motors in for direction (forward, revers, left, right).

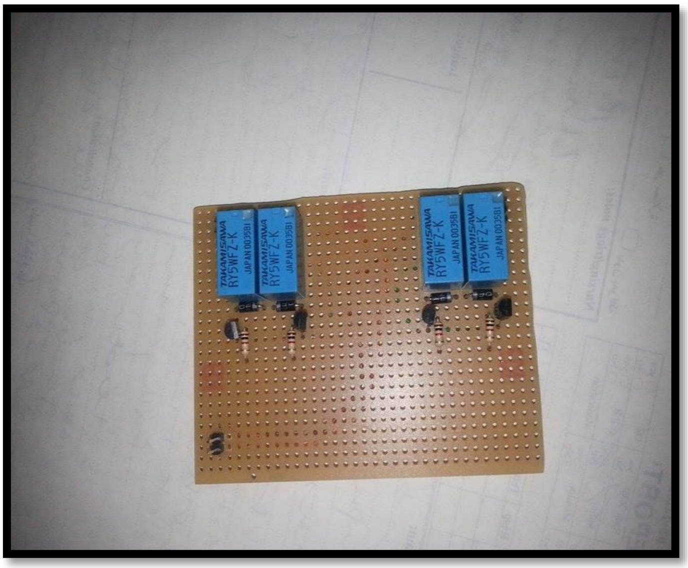


Figure 5.17: control circuit for driving motors

We used four 5V relays, four diodes, four transistors and four resistances which all needed to complete the circuit. The table below describes the type of the components:

Table 5.1: component of control circuit

Resistance	470
Transistor	2N2222
Relay	RY5WFZ- K
Diode	1N4004

After designing the suitable control circuit, we have connected it to the motors so that each relay of the four relays circuit will serve one of the motors in order to ensure having four directions movement. The figure below shows the control circuit after being connected to the body of the vehicle.

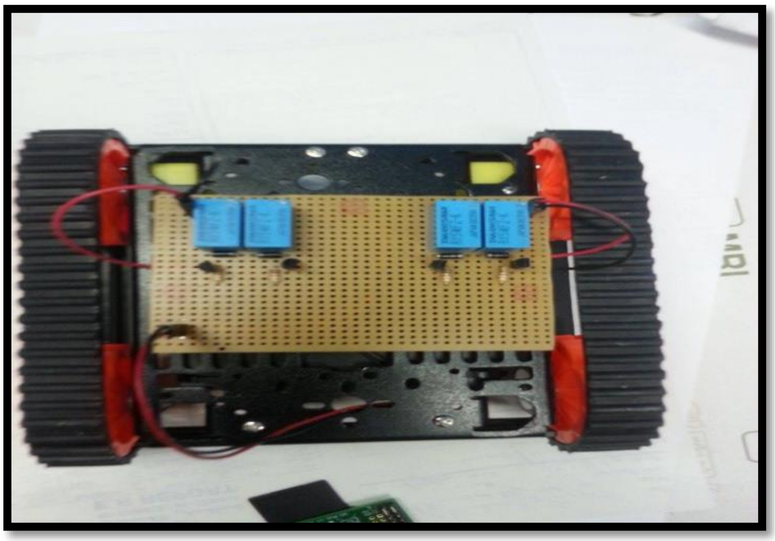


Figure: 5.18 control circuit with motor of vehicle

The last obvious step to run the control circuit is connecting it to the raspberry pi in order to start receiving orders from GPIO pins on raspberry pi for specifying the direction of movement. This will be done using the GPIO cable designed for this action. The figure below shows how control circuit will be connected to the raspberry pi.

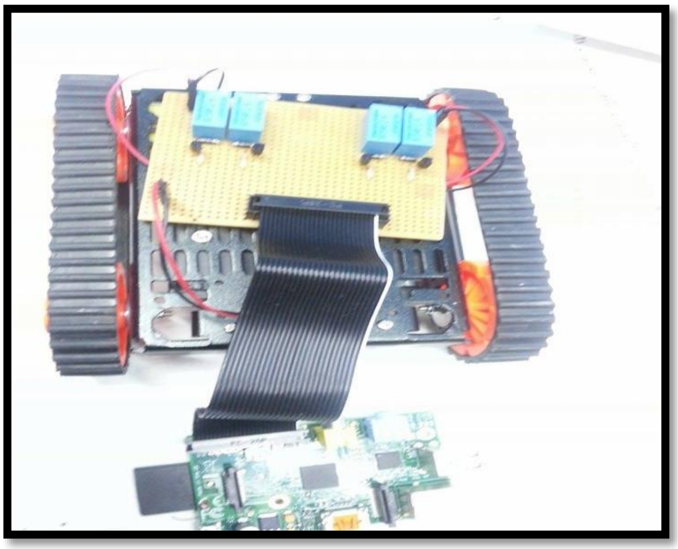


Figure 5.19: control circuit connection to GPIO pins

5.2.6 Installing scratch GPIO:

The Scratch GPIO software does not change or replace scratch, it adds a set of python scripts that integrate with Scratch and add the ability to control and read the GPIO pins at the hardware level. Scratch will continue to function as normal after the install.

There will, however, be a new icon on the graphical desktop of the Raspberry Pi that will launch the GPIO support scripts and Scratch together. The installer is simple and easy to use. We will follow the instructions below to download and install the Scratch GPIO package.

- Double click and launch **LXT.**
- Execute the following commands:

```
Wget https://raw.github.com/cymplecy/scratch_gpio/master/install_scratchgpio4.sh -O
isgh.sh
sudo sh isgh.sh
```

- The installer script will complete politely saying Thank You and Finished. Saying you're welcome is optional but certainly the polite thing to do.

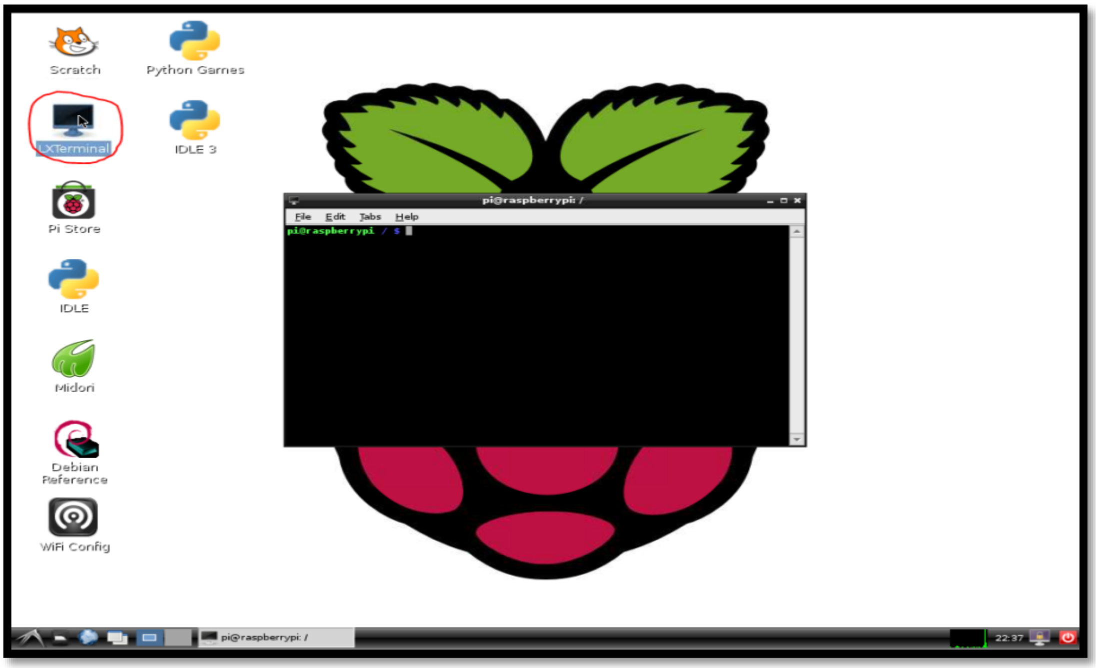


Figure 5.20: LXT home screen

- To get the program we need open the browser in our Raspberry Pi and go to this page on it
- Clicking on the icon as shown below to click on the Raspberry Pi to launch the browser.

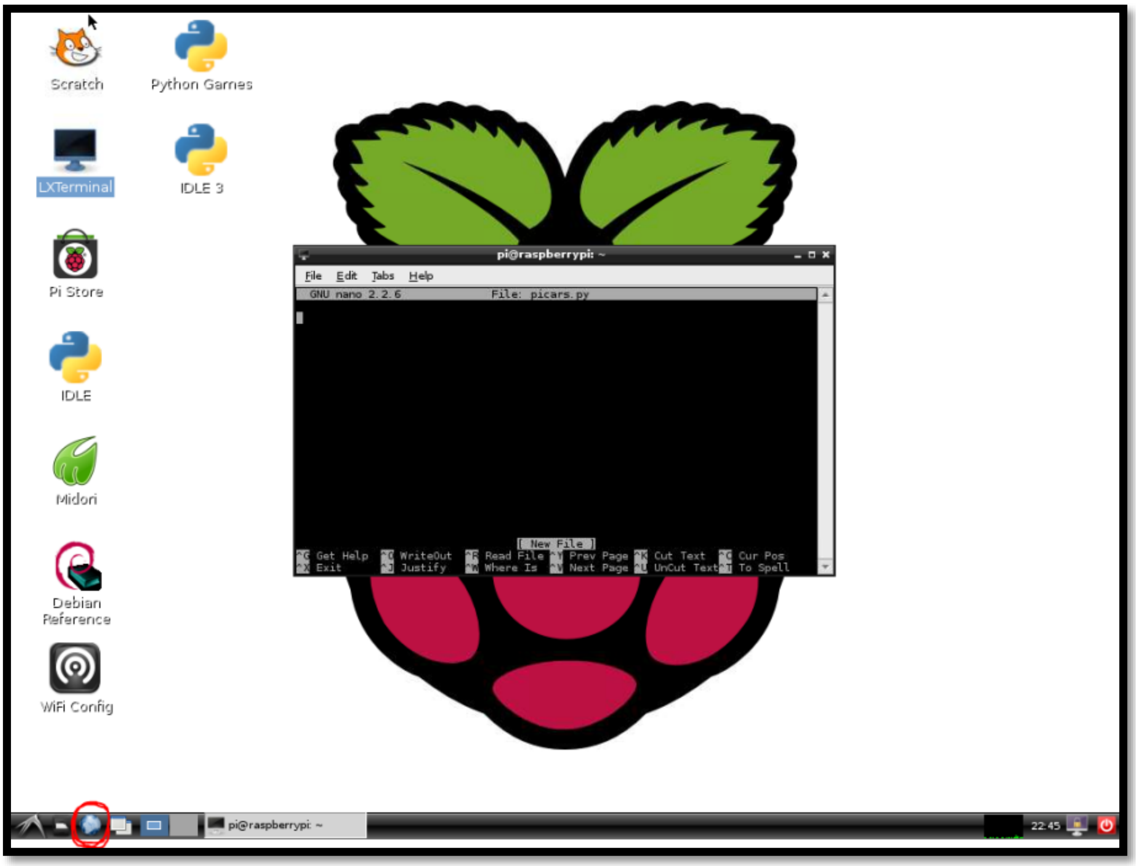


Figure 5.21: commands window of LXT

Then copy the code from the raspberry pi site to activate using scratch to ensure that it will work correctly.

Using Scratch GPIO Run the special Scratch icon (Scratch GPIO) on your desktop. As you can see in the blink11 script, we can simply use a broadcast message telling Pins to go on or off (Up to 3.3V and down to 0V) the valid messages are along with the corresponding pin off messages. We can also replace the word on with high and replace off with low if you want to talk in pure logic levels.

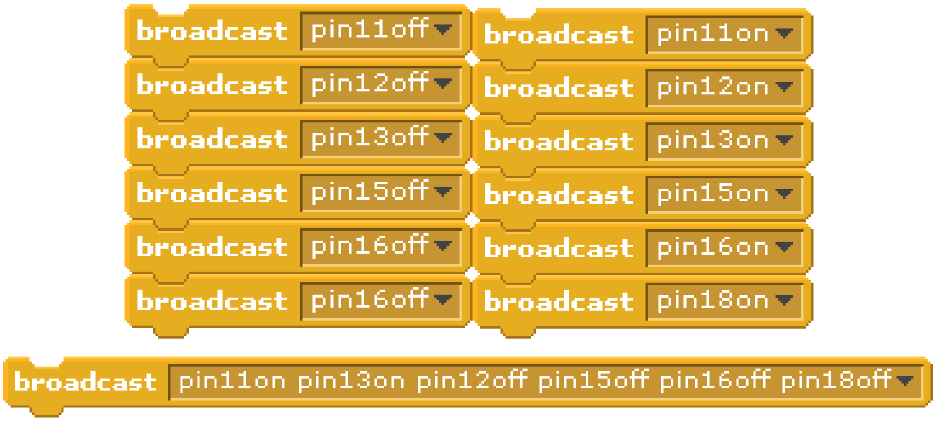


Figure 5.22: way of enabling and disabling GPIO pins

The final scratch program is shown in the figure below. This is the window responsible of specifying the direction of movement. We choose this method because its very simple.

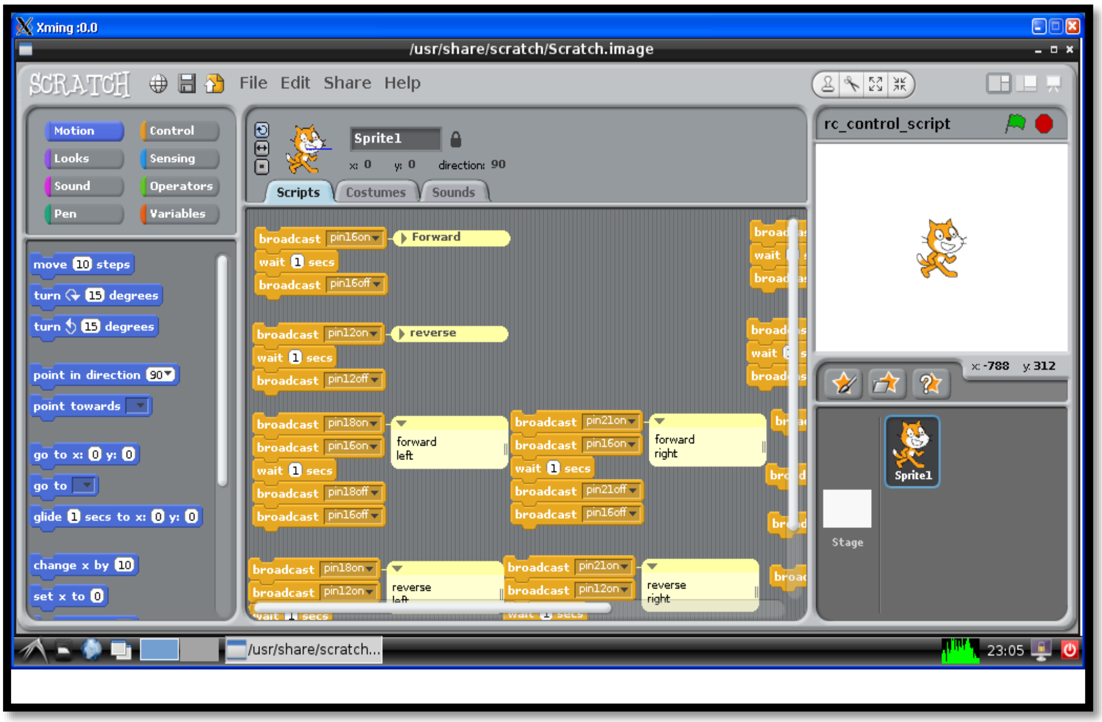


Figure 5.23: whole control program

5.2.7 Streaming:

To get a good live video from raspberry pi to be displayed on the smartphone, we must use specific commands and processes. These processes will be entered in raspberry and smartphone android application as follow:

In raspberry side:

The software that we are going to use to transmit the video stream is called MJPG-streamer. MJPG-streamer acts as a small HTTP server installed on the Raspberry Pi, transmitting the video feed captured using the webcam over HTTP, in motion jpeg format. To view the feed, you can just open a web browser and type the address (IP or domain name) of your Raspberry Pi and the port number associated with MJPG-streamer. What's really great about MJPG-streamer is that you can easily embed the feed into any webpage by using a simple IMG tag, as we will see a bit later.

Installation, configuration, testing

1. Connect the raspberry camera to the Raspberry Pi.To verify that your webcam is recognized by the Pi, just plug it in and check that the /dev/video0 entry is present in the system:
ls dev/video0

2. Make sure your system is up to date:
apt-get update
apt-get upgrade

3. Install the libv41-0 package on the Pi: apt-get install libv41-0

4. We need to download the mjpg-streamer package onto the Raspberry Pi. We have put together a custom package which should work well on the Pi.

5. Start the server application: ./mjpg-streamer.sh start

6. Test the live feed: In order to test the live feed, we open a web browser and in the address bar we type the address of our Raspberry Piand the port number associated with the mjpg-streamer server:
http://example.com:8083/?action=stream
(mjpg- streamer will start on port 8083 by default)

We should now see a live feed from our webcam in a web browser. If we want to embed this feed into a web page, we just put this into the HTML code:
**
Simple and nice! Obviously we can change the width and height parameters, but we should make sure that they are not more than the actual resolution of the feed.

When we start mjpg-streamer, it is possible to tell it which port to listen to, what resolution and what frame rate to use. The syntax of the start-up script looks like this:
Mjpg-streamer.sh [start|stop|restart|status] [port number] [resolution] [framerate]

In smartphone side

For the smartphone, we will use eclipse program to design the suitable Android program needed for controlling the movement direction and displaying captured video by the camera. The following figure shows the home screen of eclipse program.

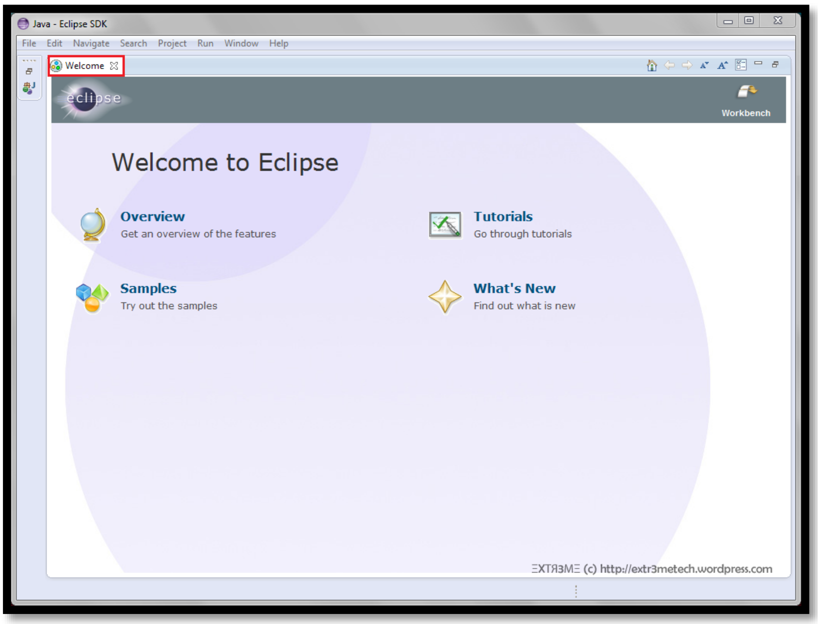


Figure 5.24: eclipse program

We will show now the main parts of the Android Application program code:

1) First we designed the interface screen which includes the movement arrows and the display screen. The first part of the code is for programming and defining these buttons in addition to the video display screen.
2) The second main part of the program is for specifying the IP address for the server and the needed port numbers.
3) The third part of program is responsible for the establishment of connection between raspberry pi and the smartphone.

4) The fourth part is for specifying the dimensions of video displaying screen.

5) The fifth part is used to link the application with the media player in order to start running captured video on the displaying screen.

Chapter 6
Testing

6.1 Introduction

The final stage to complete the project is to test the system to get results and measure the performance of our system, this chapter shows all measurements needed to evaluate the performance of this system such as delay, data rate, range and security.

6.2 Testing and Results

Checking and testing of the Android and control are individually illustrated in this section in addition to the testing result.

6.2.1 Android Testing

We use port and IP for channel and show the picture by smart phone. It works in a very good situation. If any error happened in the network the application display a message provide the user that there is an error in the network connection.

Figure 6.1: Android application testing

6.2.2 Control Testing

In this part we have tested the control circuit which connected with raspberry to move the vehicle body using raspberry pi interface to move it, and it worked correctly.

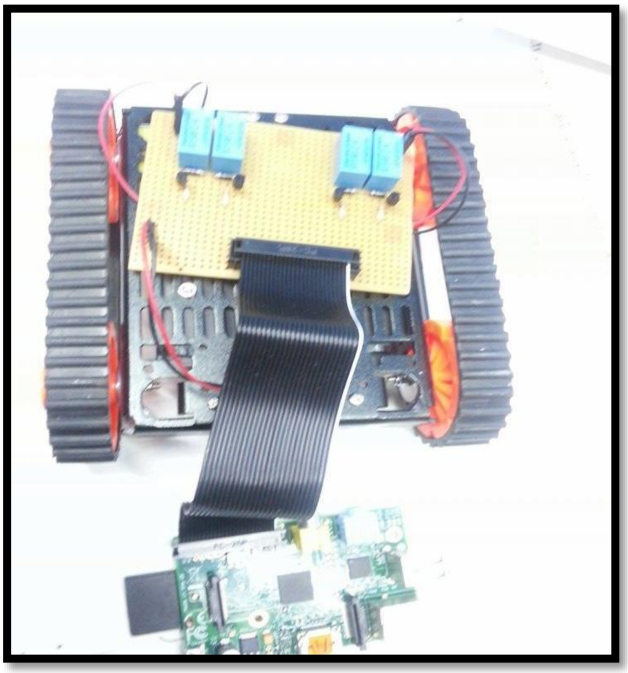


Figure 6.2: control testing

6.3 Performance Evaluation

To evaluate the performance in this project, we have to measure the Delay, Data Rate and Rang. This measurements should be applied to Wi-Fi scenarios.

6.3.1 Delay

The delay is a significant factor in this project as the nature of the sent data required fast transmission.

The delay defined how long it take for an entire message to completely arrive at the destination from the time the first bit is sent out from the source.

The delay is made of four component: propagation time, transmission time, queuing time and processing delay

.

1) Propagation Time: measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.

$$Propagation = \frac{\text{Distance}}{\text{Propagation speed}}$$

2) Transmission Time: the time required for transmission of a massage depends on the size of the message and the bandwidth of the channel.

$$Transmission\ Time = \frac{\text{DistMessage sizeance}}{\text{Bandwidth}}$$

3) Queuing Time: the time needed for each intermediate or end device to hold the message before it can processed. The queuing time is not a fixed factor; it changes with the load imposed on the network. When there is heavy traffic on the network, the queuing time increases.

4) Processing Time: the time it take routers to process the packet header. Processing delay is a key component in network delay.

But in Wi-Fi scenario, we have conducted three experiments to obtain the delay, this delay was measured by using Ping. The Ping is a computer network administration utility used to test the reachability of a host on an internet protocol (IP) network and it to measure the round-trip time for message sent from the originating host to a destination computer.

The first experiment was when we placed the transmitter and the access point and the receiver "personal computer" in the same floor, the average delay in this case was 0.6 msec which was obtain from 15 readings.

The second one when we placed access point and the transmitter in the same floor and placed the receiver "personal computer" in another floor, in this case the delay was 1.1 msec gotten from 15 readings.

The third one when we placed the transmitter in the first floor and the access point in the second floor and placed the receiver "personal computer" in the third floor, in this case the delay was 9 msec obtained from 15readings.

6.3.2 Data rate

In this project, TL-WN722N supports 802.11g standard. It operates at a maximum physical layer bit rate of 300 Mbps or about 115Mbps, average throughput. This is much more comparator to 3G GSM which provides 144Kbps data rate and 80Kbps throughput.6.3.3 Coverage. A Wi-Fi signal provide coverage typically up to 250 feet from an AP depending on factors such as walls.

6.4 Challenges

Several challenges faced the students during the design. The main hardware challenges can be summarized as follows:

1. Finding an appropriate microcontroller that can be controlled using Wi-Fi technology.

2. Offering all suitable components for implementing the project.

3. Finding the suitable design for the Android Application.

4. We faced a problem in transferring video which is delay.

6.5 Recommendations

After completing the hardware design of the project successfully, we suggested some recommendations that following bellow:

- Increase the distance that the car can cross it by using new technology.
- Increase the resolution video that receive in the user side by using good camera.
- Using the principle of this project in other idea or application.

Appendix

References

[1] Volker Graefe and Rainer Bischoff. From Ancient Machines to Intelligent Robots -A Technical Evolution[C]. The Ninth International Conference on Electronic Measurement & Instruments. 2009. Vol.3, 418-431.

[2] Liggesmeyer P., Trapp M.. Trends in Embedded Software Engineering[J]. IEEE Software, 2009, 26(3):19-25.

[3] E. D. Lagnese and D. E. Thomas, Architectural partitioning for system level design[C], in Proc. 26th DAC. June 1989, 62-67.

[4] ATMEL Corporation. ATmega128 Preliminary Summary [EB/OL]. (2010-06-21) [2011-01-13]. http://wenku.baidu.com/view/9a4bb6fb84 ae45c3b358c5d.html.

[5] Kar-Keung D. Young, Yong Quan Ou, Lun Huai Cai, Jason Kam On Ho and Ken Kin Man Cheng. Real Time Embedded Control System Development for Wireless Mobile Platforms[C]. Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on. 2008.11.18,2022-2027.

[6] Jasmin Blanchette, Mark Summerfield. C++ GUI Programming with Qt4(Second Edition) [M]. BeiJing: Electronic Industry Press. 2010.4:11-12.

[7] Alan Cox. Video4 Linux Programming[EB/OL]. (2000)[2011-01-20]. alan@redhet.com.

[8]. Momotaz Begum and Fakhri Karray Visual Attention for Robotic Cognition: A Survey, IEEE Transactions On Autonomous Mental Development, VOL. 3, NO. 1, March 2011

[9]. Ming Gao, Kui Qian and Aiguo Song Design and Implementation of a Mini-Size Search Robot, 2nd International Asia Conference on Automation and Robotics June 2010

[10]. Change Zheng Implementation of Reactive Control for a Miniature Surveillance Robot, International Conference on Information Technology and Computer Science May 2009

[11]. G.e.H.W. Stone, "HANZBOT: A hazardous materials emergency response mobile robot," in Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, 1992

[12] A. Z¨undorf, “Story driven modeling: a practical guide to model driven software development.” in ICSE, G.-C. Roman, W. G. Griswold, and B. Nuseibeh, Eds. ACM, 2005, pp. 714–715.

[13] I. Diethelm, L. Geiger, A. Z¨undorf, “Fujaba goes Mindstorms,” Objekto- rientierte Modellierung zum Anfassen; in Informatik und Schule (INFOS) 2003, Mnchen, Germany, Sep.2003. [14] R. Jubeh, “Simple robotics with Fujaba,” in Fujaba Days. Technische Universit¨at Dresden, Sep. 2008.

[15] L. Geiger and A. Zndorf, “eDOBS - Graphical Debugging for Eclipse,” Natal, Brasil, September 2006.

[16] Leif Geiger, “Design Level Debugging mit Fujaba,” in Informatiktage. Bad Schussenried, Germany: der Gesellschaft f¨ur Informatik, 2002.

[17] P. Van Gorp, R. Mans, and J. Samalikova, “Help Wall•E door de vuilnisberg heen! handleiding voor publieksdag project,” Sep 2009.