Palestine Polytechnic University

College Of Information Technology and Computer Engineering

Department of Information Technology

Smartphone Wi-Fi Controlled Toy Car

<u>Team Members</u>

Aya Nidal Ahmaro
Fahmiea Mohammad Karaki

<u>Supervisor</u>

Dr. Amal Al-Dweik

This project is to complement the graduate attributes of bachelor degree for Information Technology field of study

2014 – 2015

Smartphone Wi-Fi Controlled Toy Car

ملخص المشروع

هذا المشروع يتمحور حول فكرة التحكم بسيارة ألعاب عن بعد عن طريق شبكة الانترنت اللاسلكية بدلا من الامواج الراديوية باستخدام هاتف ذكي، يقوم بقراءة حركات يد المستخدم وترجمتها لأفعال إما للتحكم باتجاه تحرك السيارة او التحكم بسرعتها، حيث تتم قراءة حركة اليد عن طريق سنسور موجود في معظم الهواتف الذكية (الاكسلروميتر) مخصص لقراءة هذه الحركات. سيتم تصمييم وتركيب دائرة لاستقبال الشبكة اللاسلكية ووصلها مع موتور السيارة من جهة، وستقوم من جهة أخرى بالتواصل مع الهاتف الذكي الذي يعمل بنظام أندرويد لاستقبال وارسال المعلومات منه وإليه. بالإضافة إلى ذلك، سيتم تركيب كاميرا امامية على السيارة لتمكين المتحكم من رؤية المنطقة والاتجاه الذي تسير به عن طريق فيديو سيتم ارساله بشكل مباشر الى هاتف المتحكم، مما سيمكنه من القيام بعملية التحكم دون الحاجة لأن يكون بنفس المكان مع السيارة. سيتم أيضا تركيب مجسات لاكتشاف الحرائق أو الدخان والظلام، وإضافة إضاءة ليتم التفاعل مع البيئة المظلمة. يمكن استخدام هذا المشروع في اغراض ميدانية استكشافية، أو اغراض أمنية، أو حتى ترفيهية.

Smartphone Wi-Fi Controlled Toy Car

Aya Ahmaro & Fahmiea Karaki

Faculty of Information Technology and Computer Engineering, University of Palestine Polytechnic University, Hebron, Palestine, e-mail: anahmaro@gmail.com & fefemkaraki@gmail.com

Smartphone Wi-Fi controlled toy car is a project that involves designing a control system to remotely drive a toy car. Since an RC toy car has been chosen, the radio frequency circuit module used in the car is replaced with a microcomputer circuit that will be customized to receive and transmit wireless signals. An android application is developed as the main interface between the user and the car. The data is processed by the microcomputer (Raspberry Pi) and is transmitted through a Web-Socket server.

Moreover, a camera is attached to the front-end of the car to allow remote vision of the surroundings; eliminating the need of following the car everywhere. Finally, flame and dark sensors are added too, to provide the car with special capabilities to act according to the surrounding environment.

The system is able to move the 2 DC-Motors successfully according to the accelerometer readings from the phone in the four directions: left, right, forward, and backward, the webcam successfully sends live video streaming to the android phone (with two seconds lag), and the sensors works perfectly: if darkness is detected according to a tested threshold (we chose it to be 1650) of the LDR readings, the system will automatically turn on lights on the car. In addition, if fire is surrounding the car, the flame sensor will successfully detect it and directly alarm the user on his device.

In a nutshell, this project- as we have stated earlier, aims to replace the traditional radio frequency technology to remotely control objects, with a modern one: Wi-Fi; which is considered to be cheap, widely available, has a wide range, and more consistent.

Keywords: Raspberry Pi, Wi-Fi, Video-Streaming, Accelerometer, Android, Toy-Car.

Smartphone Wi-Fi Controlled Toy Car

By:

Aya Ahmaro
Fahmiea Karaki

Supervised By:

Dr.Amal Al-Dweik

Acknowledgement

All praise and thanks are only for Allah, the One who, by His blessing and favor, perfected goodness, and good works are accomplished, the One who helped us get through this project, and gave us strength whenever we stumbled and faced hardships.

As an Arabic proverb says: “If you can’t reward then you should thank.”, we would like to express our gratitude and thanks to people that helped us produce this humble work.

First of all, we are thankful for our devoted supervisor, Dr.Amal Al-Dweik, for sharing her expertise, her sincere and helpful guidance and support during the entire project, for eight months long.

To Eng.Sami Salamin, whom he shared his expertise and patience with us throughout the whole semester, as well as the lab resources.

We want to thank the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University, for their kindness and financial support to our project.

Our deepest love and appreciation go to our parents and families, to Grandma Izdihar, whom they shared our hopes, prayed for us, and supported us in good and bad.

To our sincere friends, brothers and sisters: Mohammad, Amal, Aseel, Iman, Ameena, Karim, Abdullah; thank you all for every small and big detail that you contributed to our project, for being with us in our ups and downs, for your precious time that you spent with us to present it as it is now, if it weren’t for you, it would not have been like this.

"We should certainly count our blessings, but we should also make our blessings count."- Neal A. Maxwell. We hope you are all proud of this achievement as much as we are...

Abstract

Smartphone Wi-Fi controlled toy car is a project that involves designing a control system to remotely drive a toy car. Since an RC-Toy Car has been chosen, the radio frequency circuit module used in the car is replaced with a microcomputer circuit that will be customized to receive and transmit wireless signals. An android application is developed as the main interface between the user and the car. The data is processed by the microcomputer (Raspberry Pi) and is transmitted through a Web-Socket server.

Moreover, a camera is attached to the front-end of the car to allow remote vision of the surroundings; eliminating the need of following the car everywhere. Finally, flame and dark sensors are added too, to provide the car with special capabilities to act according to the surrounding environment.

The system is able to move the 2 DC-Motors successfully according to the accelerometer readings from the phone in the four directions: left, right, forward, and backward, the webcam successfully sends live video streaming to the android phone (with two seconds lag), and the sensors works perfectly: if darkness is detected according to a tested threshold (we chose it to be 1650 ms) of the LDR readings, the system will automatically turn on lights on the car. In addition, if fire is surrounding the car, the flame sensor will successfully detect it and directly alarm the user on his device.

In a nutshell, this project- as we have stated earlier, aims to replace the traditional radio frequency technology to remotely control objects, with a modern one: Wi-Fi; which is considered to be cheap, widely available, has a wide range, and more consistent.

Glossary

DC electrical machines that converts direct current electrical power into mechanical power. 2, 5, 9, 10, 12, 15, 29, 31, 32, 34

GPIO General Purpose Input Output . 9, 11, 15, 29, 36

HDMI High Definition Multimedia Interface. 9

IC Integrated Circuit. 10

LDR Light Dependent Resistor. iii, 10, 21, 28, 32

MCB Motor Control Board. 15

PWM Pulse Width Modulation. vi, 8, 16

RC Resistor Capacitor Circuit. vi, 1, 8

RC-Toy Car Radio Controlled. iii, 4, 5, 9, 11, 16

SD Secure Digital Card. 9, 17

TCP Transmission Control Protocol. 1

USB Universal Serial Bus. 9, 11, 14, 16, 22, 28

List of Figures

Contents

Chapter 1

Introduction

1.1 Overview of the project

Smartphone Wi-Fi controlled toy car is a project that involves designing a control system to remotely drive a toy car. Since an RC toy car has been chosen for the project, the radio frequency circuit module that is used in the car has been replaced with a microcomputer that will be customized to receive and transmit wireless signals. In addition to the designing process, an android application is developed to be the main interface between the user and the Wi-Fi module. The data will be processed by the microcomputer (Raspberry Pi) and be transmitted through a WebSocket server (A protocol for full-duplex communication channels over a single TCP connection, used by any client or server application). Moreover, a camera will be attached to the front-end of the car to allow remote vision of the surroundings; eliminating the need of following the car everywhere. Finally, flame and dark sensors will be added too, to provide the car with special capabilities to adapt according to the surrounding environment no matter how tough the conditions were. It should also act according to it.

1.2 Motivation

This project is considered to be challenging on many levels: the idea itself is considered to be among the latest trends in the tech world nowadays, such as robots, flying drones and smartly built devices and appliances that can be remotely controlled either manually or automatically. Also, we have recently been hearing a lot about similar projects with more advanced capabilities that are being turned into commercial products and actually being deployed in many scenarios, such as military missions, security purposes, police surveillance, exploring activities of areas in disadvantaged environments. In addition to using such products for personal entertainment, such as organizing toy-cars or drone races between multiple players. As a result, we have been highly motivated to get involved in this remarkable trend and become part of the makers, not only the consumers.

1.3 Importance

The idea of this project may be considered innovative in our local community and targeted sectors; it is a result of an inspiration from similar ideas and projects that are still being massively developed abroad. Such products are massively in demand these days, where they serve as smart technical solutions in the industrial sector and service industry, as well as being attractive toys that can be used for personal entertainment purposes. Locally, according to some research and studies that has been conducted, this project will serve the industrial sector and service industry, as well as community individuals: it will attract individual customers, such as kids or car and drone racers. More importantly, this project is considered to be a smart solution when exploited for research purposes, such as exploring remote and disadvantaged environments, military missions, police surveillance, search and rescue activities.

1.4 Objectives

1. Exploit wireless networks instead of radio waves to remotely control the devices.
2. Control DC motors that can be deployed in many robotic and motion projects.
3. Build a system that is capable of exploring remote areas, and record collected data instead of sending out humans for such missions.
4. Build a system that is capable of detecting fires, operates in difficult environmental conditions, and acts as an informant agent that sends back data when access to such areas is limited.
5. Build devices that are remotely controlled for personal or commercial use.

1.5 Description of the Project

This proposed project is about controlling a toy car over a WiFi network from a smartphone. This is basically done using the accelerometer sensor that is built in all smartphones: a WiFi circuit is be attached to the motor of the car, which communicates with the Android based smartphone through a WebSocket. The phone sends the accelerometer readings of the three coordinates, x, y and z to the car, these readings will constantly change according to the phone motion, which will cause the car to change its direction and speed in consistency of the movement of the hand. In addition, a camera is added to the front head of the car to send real time video to the phone, meaning that the user will be granted distant/remote controlling capability. Sensors to detect fire and darkness are added to upgrade the functionality of the car to operate in difficult and disadvantaged environmental conditions.

1.6 Report Outline

Going further in this report, chapter two will talk about the main problem statement of the system, the main requirements of the system and the expected results. Chapter three talks briefly about the theoretical background of the project, literature review, hardware parts, specifications and design constraints.
Chapter four discusses the conceptual design of the system, block diagrams, flowchart, and detailed hardware connection and subsystems. Chapter five handles the software requirements and design, shows the code used and main interfaces of the android application. Chapter six shows validation and testing steps of the implemented system, with graphical demonstrations of results of each unit testing in the system. It also talks about implementation issues that were faced, and the overall system results of implementation. Last but not least, conclusion and suggested future work is included in chapter seven. Followed an appendix and complete list of references.

Chapter 2

Problem Statement

2.1 Problem Analysis

The system will utilize a commercially available smartphone (an Android phone in this case) to control a vehicle and its sensors. These capabilities are applicable to multiple scenarios including military missions, police surveillance, search and rescue activities. The prototype should be developed and demonstrated using commercially available products.

2.2 List of Requirements

The system should have the following requirements:

- Functional Requirements

    1. Transmit all data between RC-Toy Car and Android Phone in a WiFi covered range.
    2. Ability to change car location according to accelerometer readings in the phone.
    3. Ability to process streaming video at 15 fps minimum with 16 bit color (minimum color requirements for an android phone)
    4. Use flame sensor to detect fires and alarm user.
    5. Use light sensors to detect darkness and turn on the lights on the car to enhance vision.

6. Ability to drive the car in accordance to the available battery life.
7. The RC car must be able to maintain a reasonable speed.

- Non-Functional Requirements
    1. The RC-Toy Car shall run on two DC electric motors for four directions.
    2. The user must use Android to control the system.
    3. Communication protocol is IEEE 802.11n standard for WebSockets.

2.3 Expected Results

The following are the results of the projects that are expected to be accomplished:

1. Control vehicle with smartphone via a WI-FI connection.
2. Car moves in four directions: forward, backward, left and right according to phone motion.
3. Meet the video streaming requirements.
4. Flame sensors to detect fire and alarm user.
5. Use Light sensors to detect darkness, and thus turn on the lights of the car.
6. Maintain a controllable speed.
7. The user control system must use Android.

Chapter 3

Background

3.1 Overview

This chapter talks briefly about the theoretical background of the project, , short description of the hardware parts used in the system, specification and design constraints.

3.2 Theoretical Background

This section will provide some information about some technologies used in the project.

Sensors

A sensor is a device that detects and responds to some type of input from the physical surrounding environment. The specific input could be light, heat, motion, moisture, pressure...etc. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or for further processing.[1]

Raspberry Pi Microcomputer

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the Internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. Model B+ was used due to its high data processing capabilities; refer to Appendix A for detailed specifications.[2]

Android operating system

Android is an open source operating system, a large community of companies and developers maintain it and contribute toward developing newer versions of it. The Android Operating System is a Linux-based OS developed by the Open Handset Alliance (OHA), the user interface of android is based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects. Android's source code is released by Google under the Apache Lances this permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Android software development is the process by which new applications are created for the Android operating system, applications are usually developed in the Java programming language using the Android Software Development Kit[3]

Wi-Fi

Is a local area wireless technology that allows an electronic device to exchange data or connect to the Internet using 2.4 GHz and 5 GHz radio waves. Many devices can use Wi-Fi, e.g., personal computers, video-game consoles, smartphones , digital cameras, tablet computers and digital audio players. These can connect to a network resource such as the Internet via a wireless network access point. Such an access point has a range of about 46 meters indoors and a greater range outdoors. Hotspot coverage can comprise an area as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points.[4, 5]

Smartphone

Is a mobile phone with more advanced computing capability and connectivity than basic feature phones. Smartphones typically include the features of a phone with those of another popular consumer device, such as a personal digital assistant, a media player, a digital camera, and/or a GPS navigation unit. Later smartphones include all of those plus the features of a touchscreen computer, including web browsing, Wi-Fi, 3rd-party apps, motion sensor, mobile payment and 3G.[6]

PWM

Is a technique used to encode a message into a pulsing signal. It is a type of modulation. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors, that's why we use it to control speed of the motor.[7]

RC-Circuit

Is an electric circuit composed of resistors and capacitors driven by a voltage or current source. A first order RC circuit is composed of one resistor and one capacitor and is the simplest type of RC circuit. In this project, an RC-Circuit is used for darkness detection. Refer to Appendix B for more details on how RC-Circuit was used for darkness detection.[8]

3.3 Literature Review

In 2012, there was a graduation project in the Electrical and Computer Engineering Department, called "Movable Camera for Special Purposes"; in which they redesigned an RC-Toy Car, they used an ordinary infrared car remote controller. A remote receiver was attached to the DC motor circuit, with an L293 bridge. The car was only meant to be moved forwards or backwards in a straight line, and a wireless camera board was used with a transmitter on the car, and a receiver module in the remote. No clear results are shown, the car did not move and no video was transmitted.

Our project idea differs from this one in the use of a sophisticated microcomputer with more advanced technologies, such as the Wi-Fi data transmission, the webcam, the smartphone as controller, flame and light sensors. The system will be able to stream live video back to the phone, the sensors will detect fire and darkness, alarm the user of fire, and trigger car lights if darkness is detected.

3.4 Hardware Components of the System

This section shows the hardware components used in the project, and how they are used.

1. **One RC Toy Car:** The car we have consists of two simple DC-motors, one that drives the vehicle forward/backward, and another one that steers the car: front wheels left/right. The radio control circuit will be removed from the car.

2. **One Raspberry Pi, Model B+:** This is the type of microcomputer that will be used. It Runs Linux (and Java), has 512MB RAM, USB ports (for keyboard, mouse, WiFi), takes an SD card, has an HDMI out, and has a handful of GPIO (General Purpose Input/Output) pins to drive external devices, like lights, motors.[2]

3. **Regular Webcam:** It is used to record video of the surrounding place to stream it back to the smartphone.

4. **Dongle, Wi-Fi, USB, for Raspberry Pi:** It is used to set up connection with the WiFi access point of any router.

5. **Wifi Router:** To generate Wifi waves, and thus create the transmitting media that will connect both the phone and the Pi together in the same network.

6. **One Breadboard:** This simplifies connecting the building blocks, sensors and LDR will be placed on it.

7. **Sensors:** Two types of sensors will be used:

    a) Light Dependent Resistor (LDR) : to detect darkness.

    b) Flame Sensor: to detect fire.

8. **5 LEDs:** To provide car with lights that turns on once darkness is detected through the LDR.

9. **Connection Series:**Set of male to male, male to female, female to female jumper leads.

10. **One Smartphone:** This is the controlling device, the built-in phone accelerometer sensor will be used to detect motion of the hand and translates this motion to signals to be transmitted over Wi-Fi network to instruct the car direction. Video streaming from the Pi will be transmitted and displayed on the phone.

11. **Batteries:**

    - 9-volts, 2A battery pack to power the motors.
    - 5-volts, 2A battery pack to power the Raspberry Pi.

12. **L293D Motor Driver IC:** L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that two DC motor can be controlled with a single L293D IC.[9]

3.5 Design Specifications & Constraints

This section talks briefly about design specifications and constraints.

1. **Design Specifications:**

   The system must demonstrate the ability to transmit all data between RC-Toy Car and Android Phone in a Wi-Fi covered range, it must be able to process streaming video, use temporal sensors to detect fire, and light sensors to detect darkness and thus, turn on lights on the car to enhance the vision.Finally, the system must show ability to autonomously drive the car in accordance to the available battery life and provide the car with ability to maintain a reasonable speed.

2. **Design Constraints:**

   a) Type of microcomputer must be B+ model, because of the extended general input output pins (GPIOs) provided, the USB ports, and enhanced functionality and capabilities compared to previous models.

   b) The distance between the user and the car should be within the Wi-fi range, which is 46 meters indoors and 92 meters outdoors according to the IEEE 802.11 protocol of the access point.[5]

   c) The Wi-Fi range may vary depending on the strength of the device transmitter, the physical obstructions, and/or the radio interference in the surrounding area.

   d) Use inexpensive off-the-shelf products as much as possible.

   e) Budget of 2600 NIS approximately.

Chapter 4

System Design

4.1 Overview

This chapter discusses the conceptual design of the system, it shows a block diagram of its hardware components, a flow chart, software and hardware requirements.

4.2 Brief Description of the System

An android application streams the accelerometer data from the phone to the Raspberry Pi over a simple websocket. The Pi then uses this data to drive the two DC motors. A Wi-Fi dongle is attached to the Pi to enable Wi-Fi functionality in the microcomputer. Thus, once the car is connected to the phone through Wi-Fi, it will be able to send video streaming to the car, information from the flame and light sensors to inform the user of the surrounding conditions of the car. Lights will be turned on if the car detects darkness and the vision becomes blurry.

4.3 Flowchart

Figure 4.1 represents the flowchart of system activity

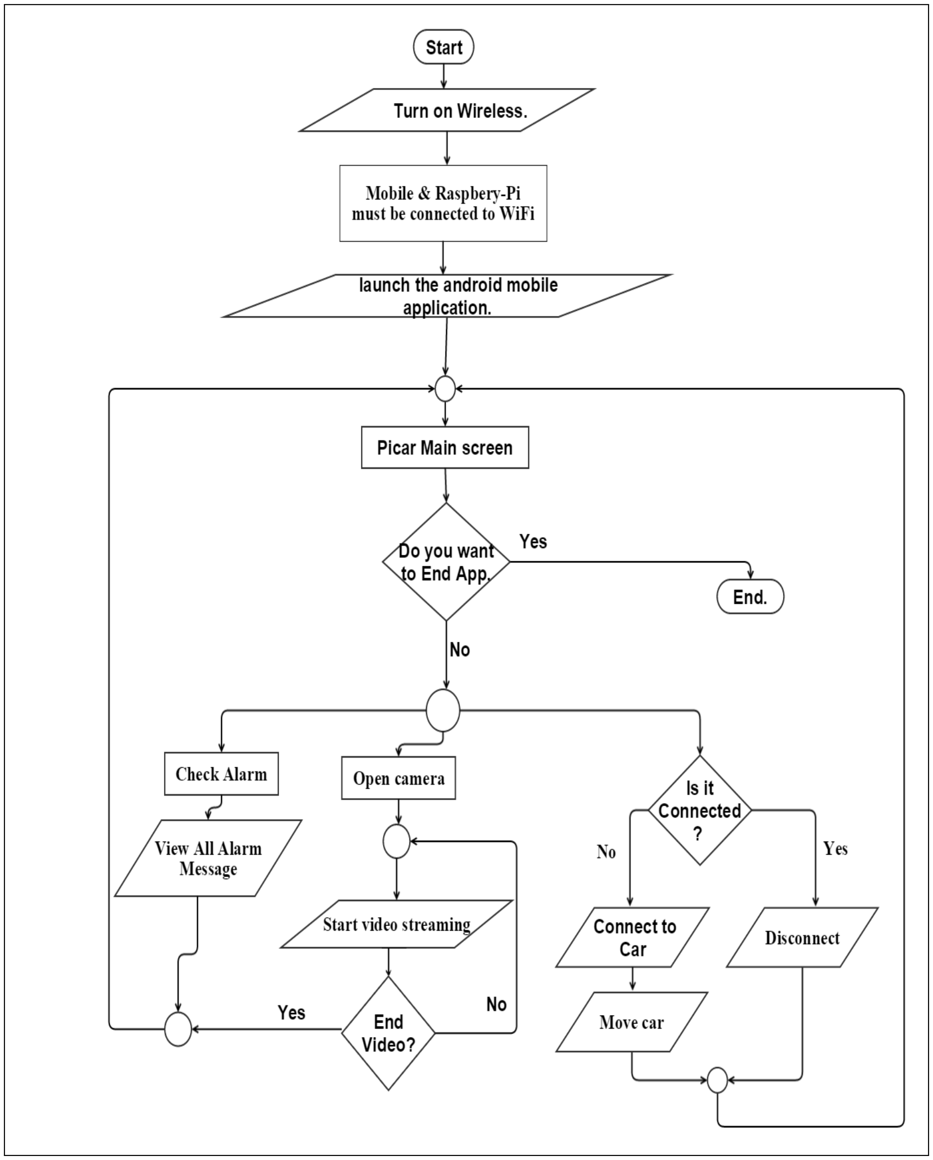

Figure 4.1: System flowchart

4.4 Description Diagram

Figure 4.2 represents a brief description diagram of the system components.

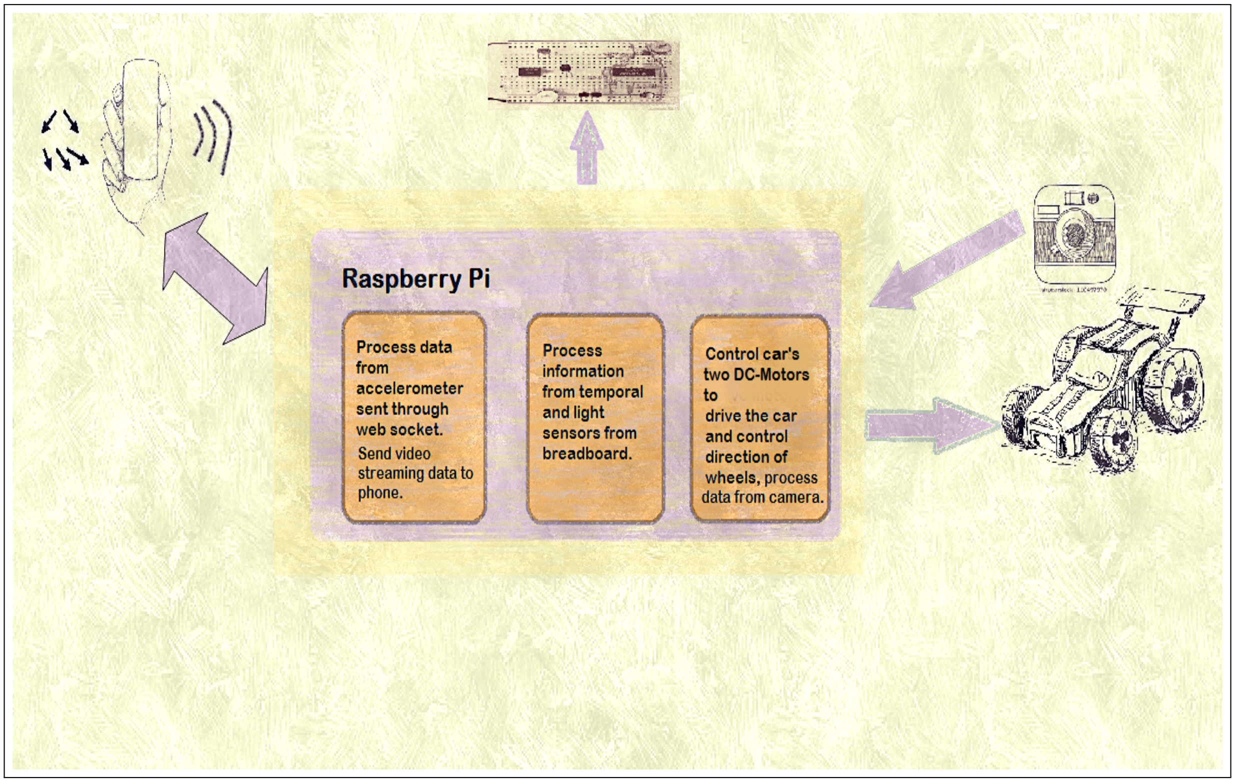

Figure 4.2: Description Diagram of Raspberry Pi controlled car

4.5 System Design

Design Options

During our research, we have encountered many alternatives regarding on how to construct a Raspberry Pi motor controlling circuit. These include:

- Camera Options:
  1. **First Design Option:** Use an ordinary webcam to transmit video streaming to the phone,and attach it to the Pi through a USB port.

2. **Second Design Option:** Use a special camera Pi module to transmit the video streaming to the phone.
3. **Chosen Design Option:** We have chosen the first approach, because it is cheaper, more widely available, plus, given the high resolution at which the Pi camera can work, it is possible that the performance of Raspberry PI be not enough (processing of large images is CPU-hungry) considering that we only have one Pi with multiple tasks, not to mention that only one Pi camera module may be attached to the Pi, whereas we can set multiple webcams on the car.[10]

- Motor controller Options:

1. **First Design Option:** Use the L293DH-bridge integrated circuit to control the motors.
2. **Second Design Option:** Use the Pi MCB/Servo Board to control the motors.
3. **Chosen Design Option:** The first approach provides easier software programming, highly compatible with the Raspberry Pi, giving more consistent results, and it is cheaper, thus it was chosen.

- Sensors Options:

1. **First Design Option:** Use electronic integrated circuits of flame and light sensors and attach them to a breadboard, which will be connected to the Pi externally.
2. **Second Design Option:** Use Pi sensor modules for temperature sensors and light sensors.
3. **Chosen Design Option:** The first approach was chosen, because it is easier and cheaper, and getting more modules will mean more software programming on the board, all of the GPIO pins on the Pi will be consumed, and thus we will need an extension GPIO pins board to complete the connections.

- Chassis Options:

1. **First Design Option:** We can construct our own car chassis,or any other figure that we wish, like Lego robots, and get separate DC motors, servo motors and wheels,connect them to whatever we have built,and control it.

2. **Second Design Option:** Buy an RC-Toy Car, remove the upper body,the radio control, and attach the Pi and all of the new components on the chassis of the car.
3. **Chosen Design Option:** We have chosen the second approach, due to the availability of RC toy cars in the market, with already installed and connected wheels and motors to the body of the car; instead of worrying about the availability of the pieces, and ordering them separately, whether they work together or not...etc.

Main Hardware connection

The main component in the project is the Raspberry Pi microcomputer, it will be attached on top of the body of the car, powered by a 5 volts battery. An L293D (H-bridge) is connected between the Pi and the motors. Moreover, the PWM pins (pulse width modulation) s on the Pi are used to control the speed of the car. A webcam is used to stream input video to the phone. A Wi-Fi adapter is inserted in a USB port on the Pi and it connects to the smartphone that is acting as a wireless hotspot to the Pi.

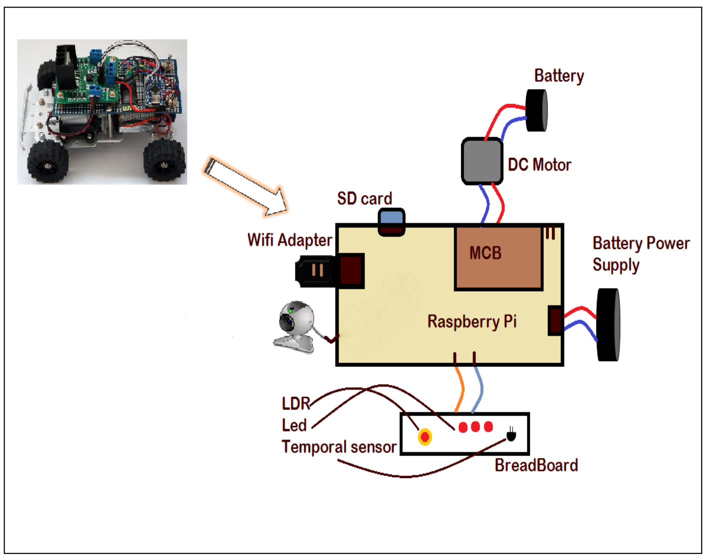

Figure 4.3: System Internal Design

Flame and light sensors are added to the Pi using a breadboard connected directly to the Pi through the pins. Leds are attached as well and will be turned on whenever the light sensors detects darkness. An SD card is needed to install the OS on the Raspberry Pi, most likely it is going to be Raspbian operating system. The controlling program software can be written in any language, C, Scratch, python, Java, php, etc; python is used in our case.[9, 11, 12, 10, 8, 13]

Block Diagrams of the System

Figure 4.4 represents the overall System Block Diagram

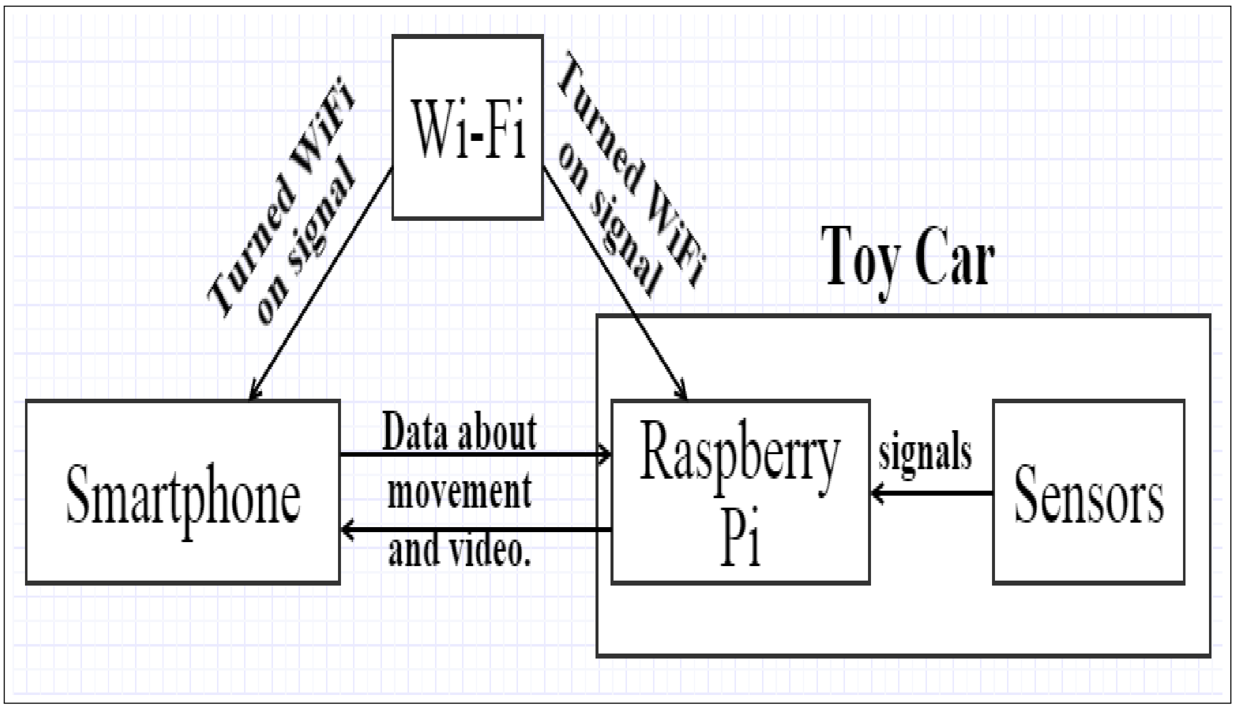

Figure 4.4: Overall System Block Diagram

Figure 4.5 represents the Smartphone Block Diagram

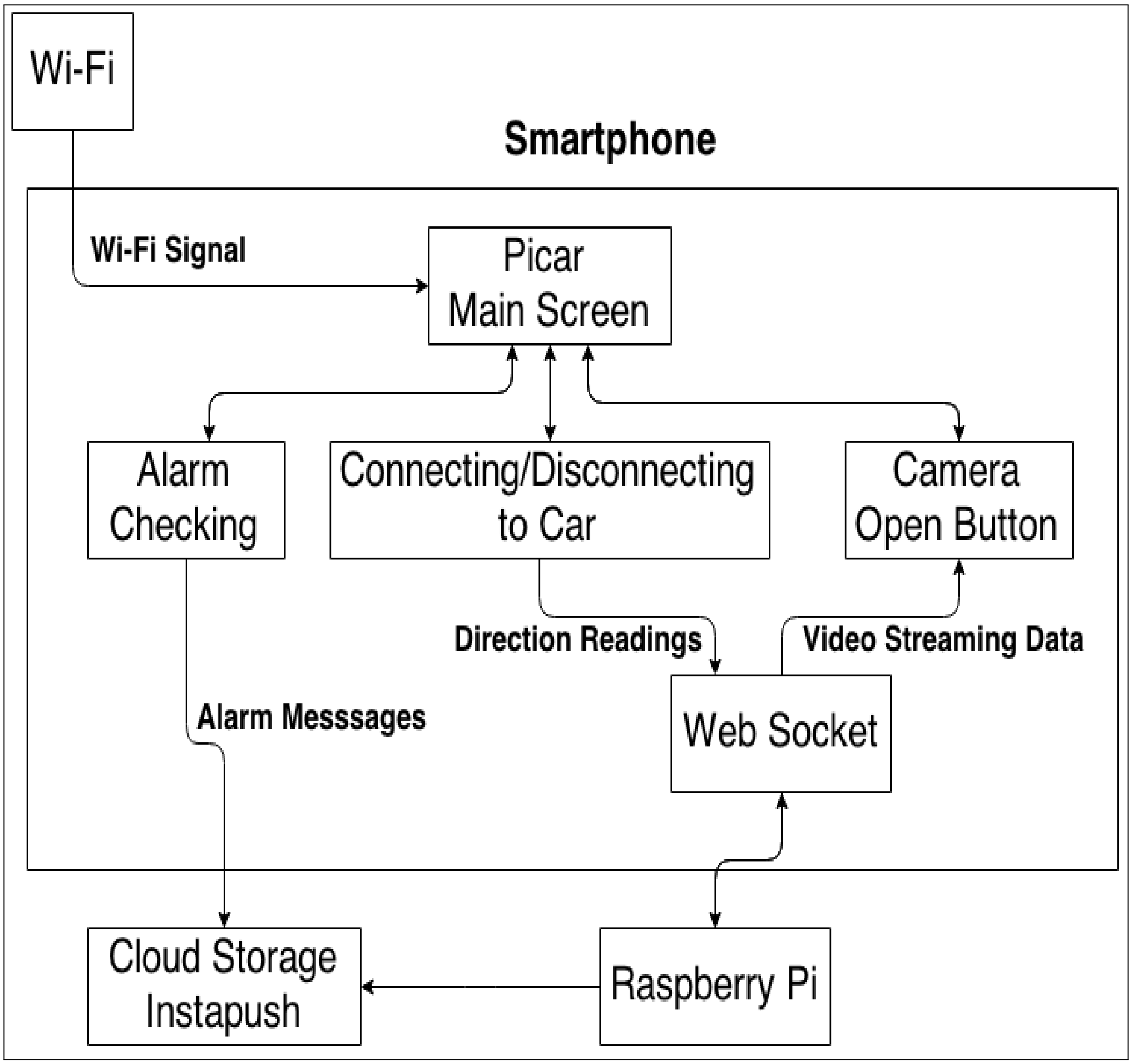

Figure 4.5: Smartphone Block Diagram

Figure 4.6 represents the Raspberry Pi Block Diagram

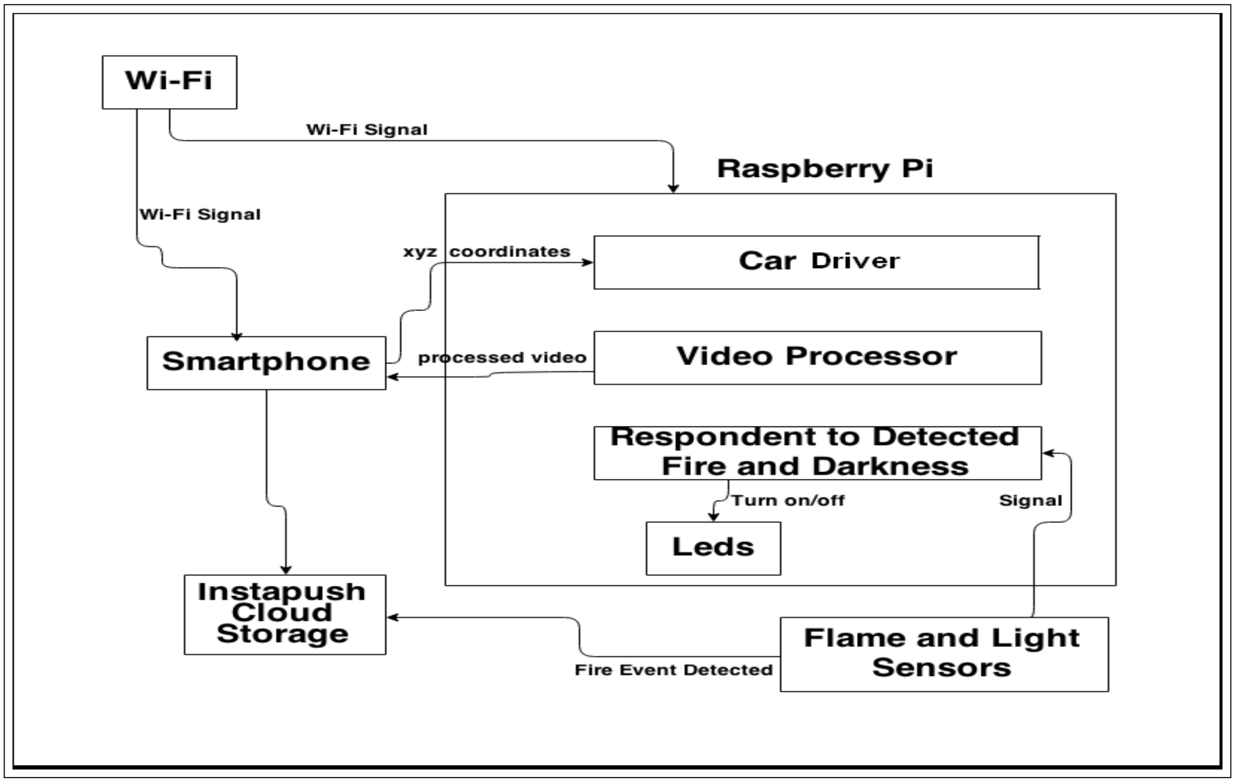

Figure 4.6: Raspberry Pi Block Diagram

Figure 4.7 represents the Sensors Block Diagram

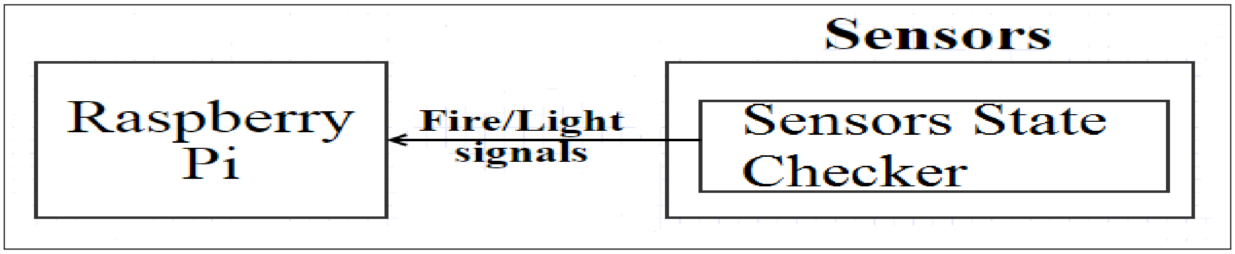

Figure 4.7: Sensors Block Diagram

Detailed Hardware Connection

Figure 4.8 represents the detailed hardware connection

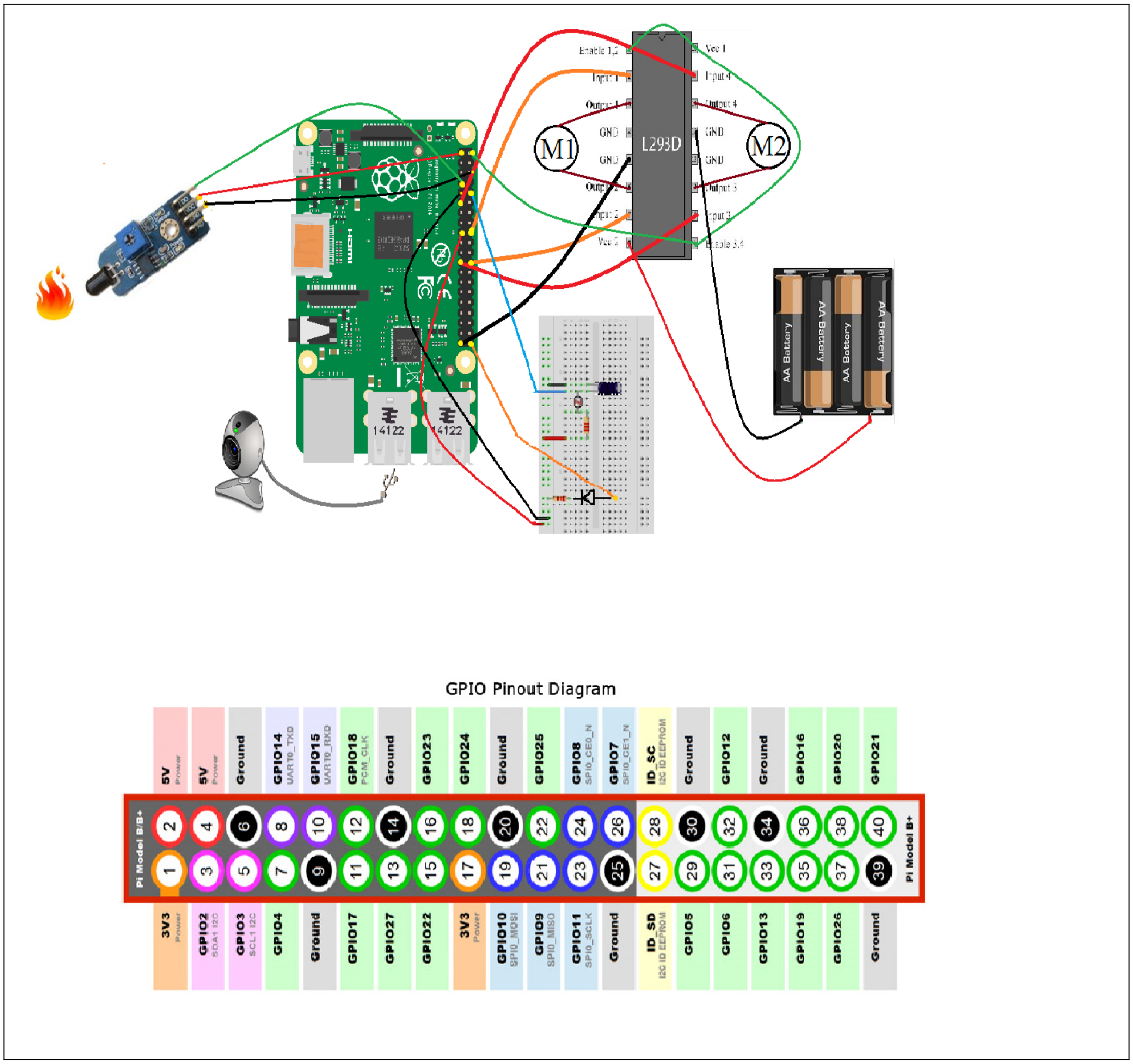

Figure 4.8: Detailed Hardware Connection

Detailed connections:

- **Motors Driving Subsystem:** Connect IN1 of the H-bridge to pin 18 on the Pi, IN2 to pin 24, IN3 to pin 23, IN4 to pin 11.
  Connect one of the ground pins of the H-bridge to pin 39 on the Pi, Vcc1 and the two enables of the bridge with pin 2 on the Pi (5V). Finally, outputs 1 and 2 for the first motor, and outputs 3 and 4 to the second motor.

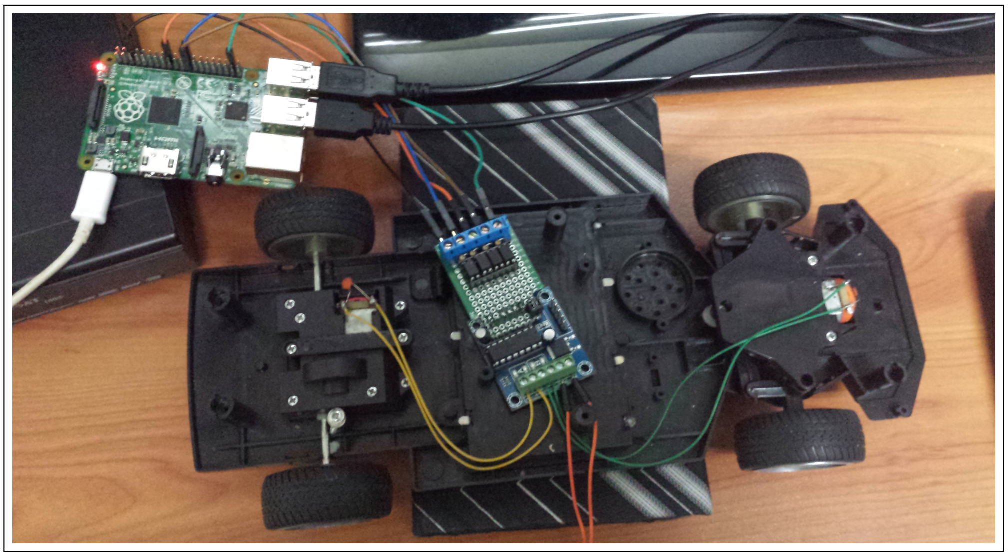

Figure 4.9: Motors Subsystem

- **Darkness Detection Subsystem:** Connect pin 17 of the Pi of output 3.3V to Vcc of the breadboard, and the ground to pin 9, between the positive leg of the capacitor and one leg of the LDR we have a wire connecting to pin 10 on the Pi, the negative leg of the capacitor is connected the ground of the breadboard; between the second leg of the LDR and the Vcc of the breadboard 3.3V, a resistor of value 2.2Kohms has been inserted for protection. To light the led, output from pin 40 on the Pi is connected to the positive leg of the led, and the negative leg should be connected to ground, with a 100ohms resistor in between for protection. Refer to Appendix B for more theoretical details.

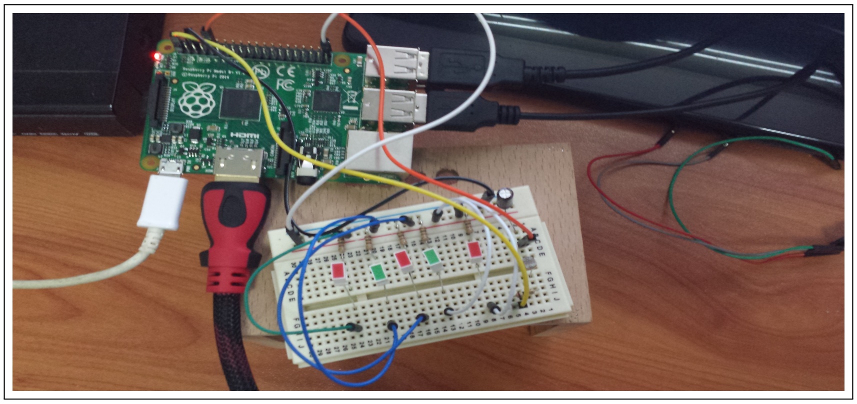

Figure 4.10: Darkness Detection Subsystem

- **Fire Detection Sub-system:** The Vcc of the flame sensor that has been used was connected to pin 1 of the Pi, with 3.3V, the ground of the sensor has been connected to pin 6 of the Pi, and finally, the DC output of the flame was connected to pin 7 on the Pi.

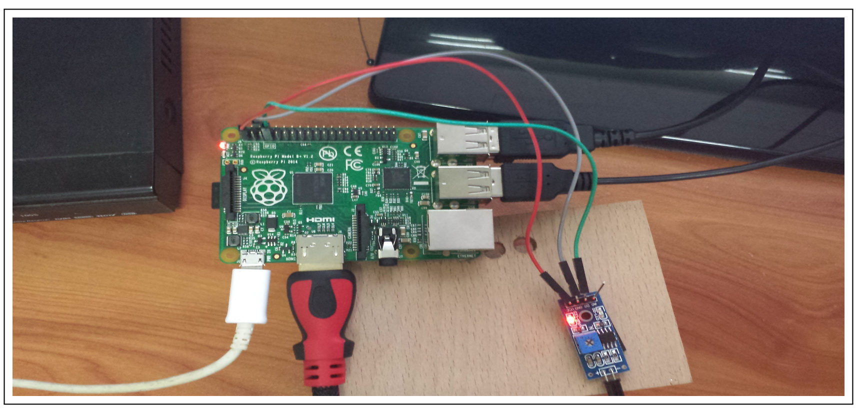

Figure 4.11: Fire Detection Subsystem

- **Camera Subsystem:** Just plug in the webcam to the USB port, and follow the instructions shown in Appendix C on how to configure motionEye, which is a linux based web interface for surveillance, to turn on the webcam with Raspberry Pi.

Chapter 5

Software

5.1 Overview

In this chapter, user interfaces of the android application are presented. It is worth nothing that it is necessary for the user to have two essential android apps on his phone in order for this project to work. These two apps are: Wireless IMU; to hack into the phone accelerometer sensor, and Instapush; which acts as the main interface for the cloud service used to store fire detection events. Below are the downloadable links, directly from Google play.

- Wireless IMU: "https://play.google.com/store/apps/details?id=org.zwiener.wimuhl=en"
- Instapush: "https://play.google.com/store/apps/details?id=im.instapush.pushbotshl=en"

5.2 User Interfaces

In this section, the interfaces of the android app "Picar" are presented, with detailed transition between the screens:

- Figure 5.1 is the main screen of Picar.
- If the user presses "Connect to Car" button, screen of figure 5.2 will appear.
- After the connection, the "Disconnect" button will appear as shown in figure 5.3.
- If the user presses "Disconnect button, figure 5.4 is the result.

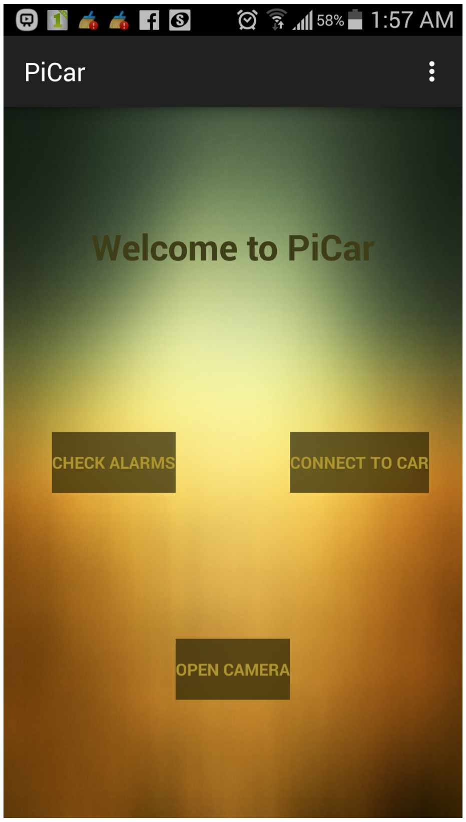

Figure 5.1: Picar Mainscreen

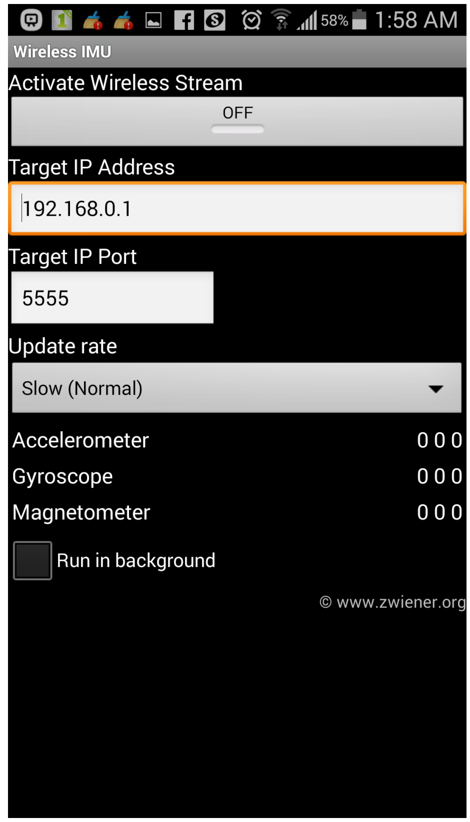

Figure 5.2: IMU Connect

- If fire is detected, a direct notification will be sent to the user instantly, as shown in figure 5.5.
- If the user wants to check out the fire events that were detected within the past 24 hours, he can press "Check Alarms" button, screen in figure 5.6, and figure 5.7 will appear.
- If the user wants to turn on camera, he can press "Open Camera" button, and the camera starts streaming, as shown in a screen-shot in figure 5.8.

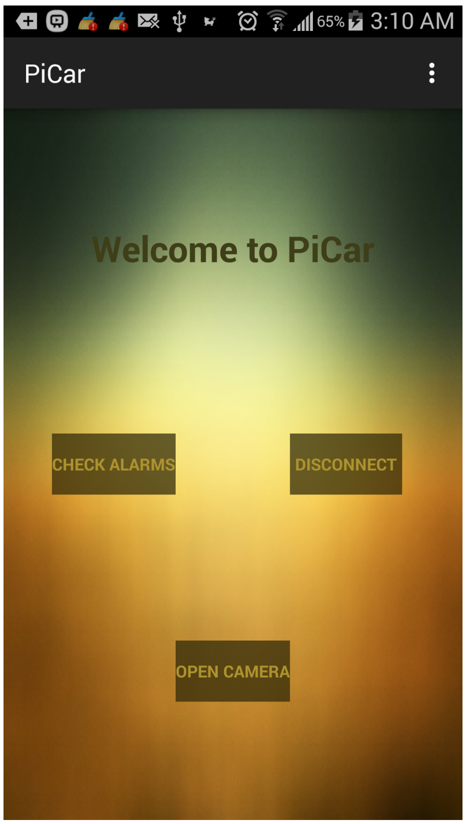

Figure 5.3: Picar Disconnect

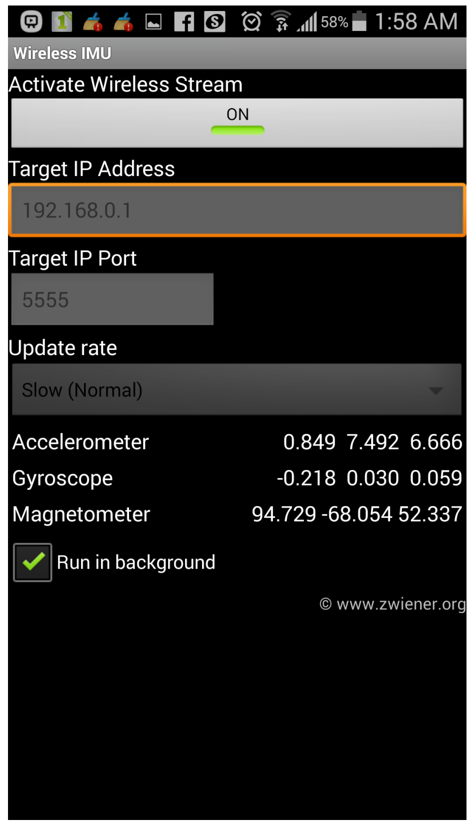

Figure 5.4: IMU Disconnect

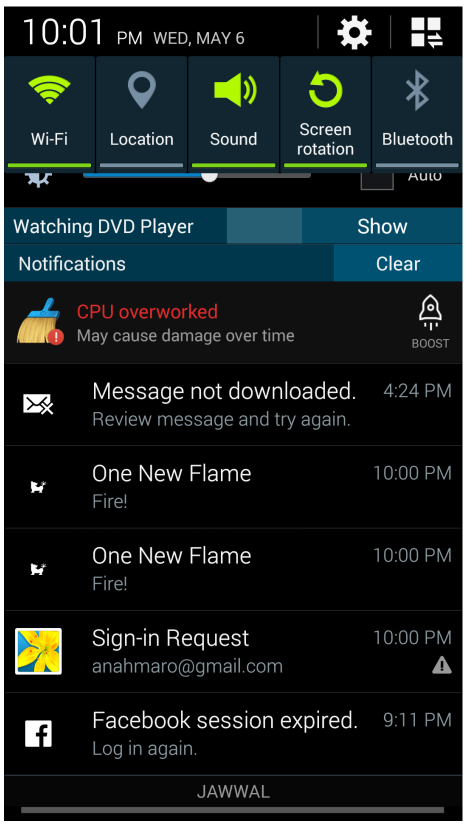

Figure 5.5: Fire Notification

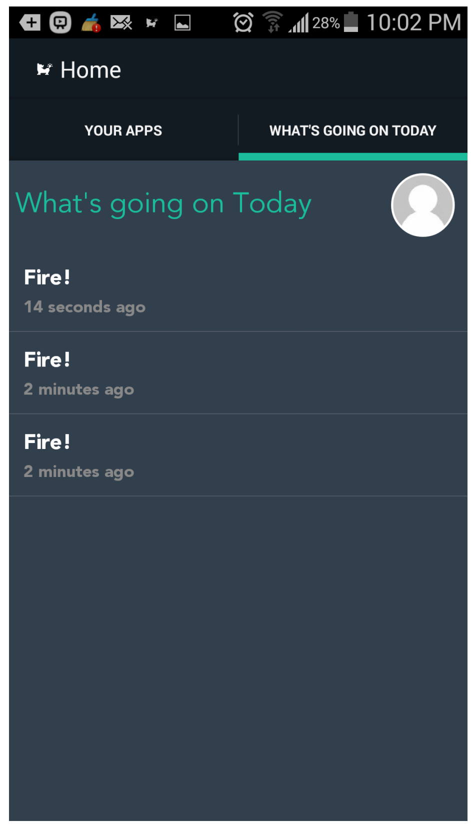

Figure 5.6: Instapush Fire-Check(1)

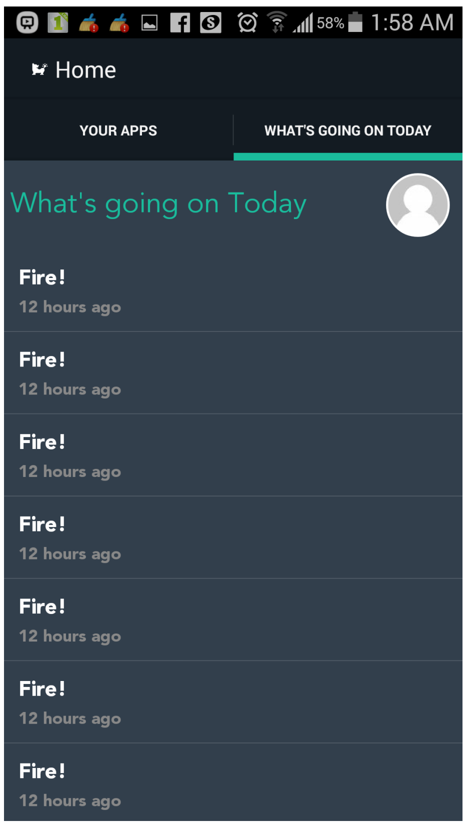

Figure 5.7: Instapush Fire-Check(2)

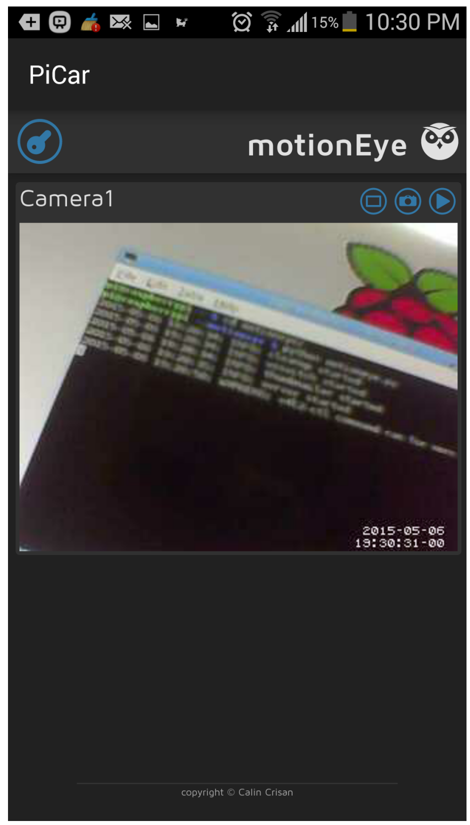

Figure 5.8: Video Streaming

Chapter 6

Validation and Testing

Since the project consists of four main tasks for the Pi, each task was implemented and tested individually; after debugging and obtaining successful results from each, the four were all combined together, to form an integrated system that drives motors, send live stream to an android phone, detect darkness to turn on leds, and fire to alarm the user.

1. Video Streaming Test The webcam USB was plugged to the USB port at the Raspberry Pi, and the motioneye.py script was run, the results appeared on the phone directly as shown in figure 6.1:

2. Fire Test The fire detection subsystem was tested, and a notification was sent successfully to the phone, as shown in figures 6.2 and 6.3:

3. Darkness Test The darkness detection subsystem was tested under different light circumstances, until it was found out the best threshold is 1650 millisecond, meaning that if the LDR value is 1650 ms or above, darkness is surrounding the car and the lights should be turned on automatically; else, they remain in the off state. Figure 6.4 shows the lights turned on in the darkness:

4. Motors Test After the circuit of motors was connected as explained in the previous chapter, power supply was applied to the motors, the motors python scripts was run, and the motors moved according to the motion of the phone. The test was passes successfully and recorded in a video.

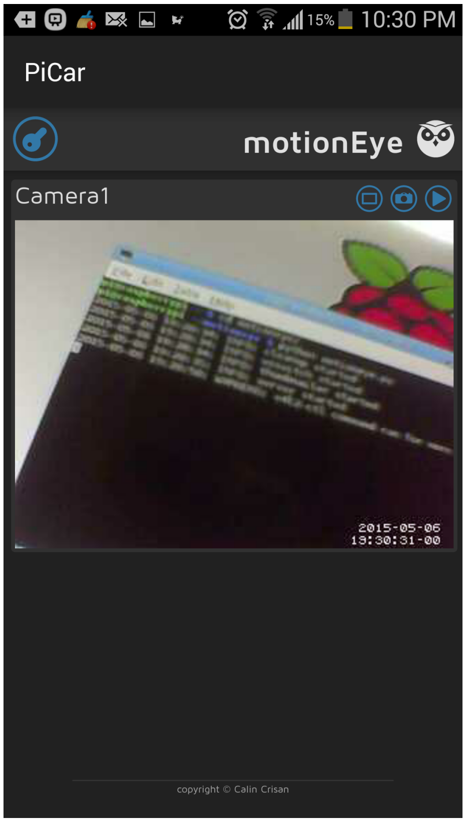

Figure 6.1: Video Streaming test

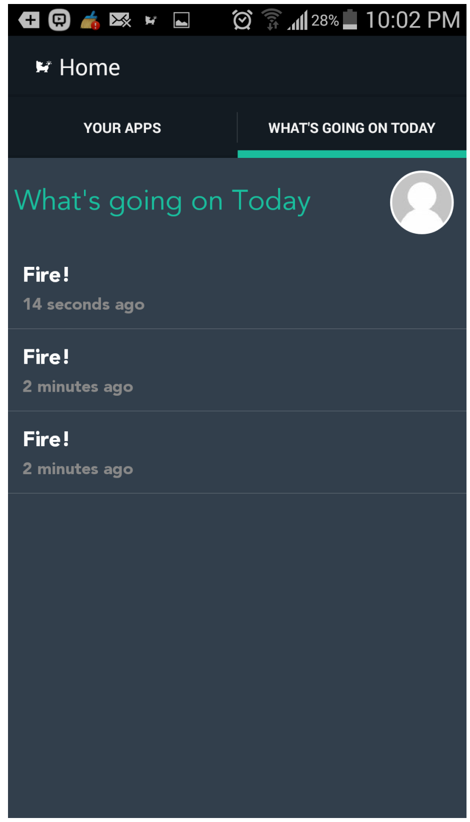

Figure 6.2: Fire Notification Test

Implementation Issues

- We faced serious trouble while testing the motor movement and their interaction with the GPIO pins of the Pi, there seemed to be an overshoot voltage coming from the DC motors while moving suddenly to the GPIO pins, and thus causing the Pi to stop and crash. Therefore, a protection circuit was used eventually, consisting of 4 optocouplers, to isolate the motor driving circuit from the Pi, and thus, preventing sudden signals that MAY be generated from the motors to reach the Pi. The optocouplers transferred the signals between the two electrical circuits using light.

  The circuit after configuration is shown in figure 6.5:

- The second implementation issue is considered to be a minor one: there is a video lag of 2 seconds approximately on the phone; this is the best

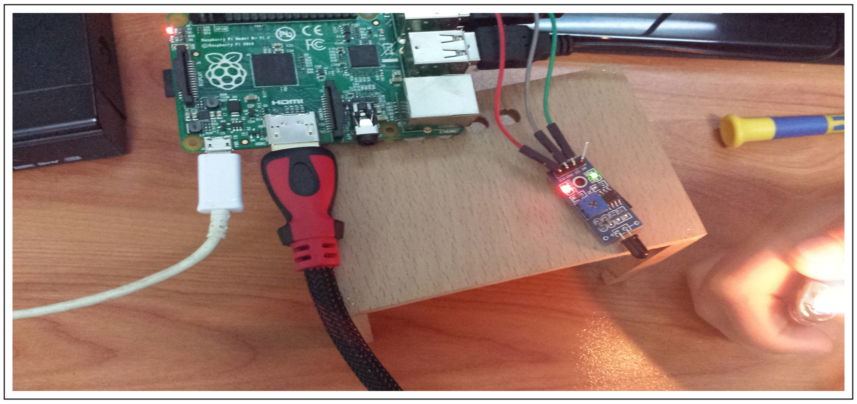

Figure 6.3: Fire Test

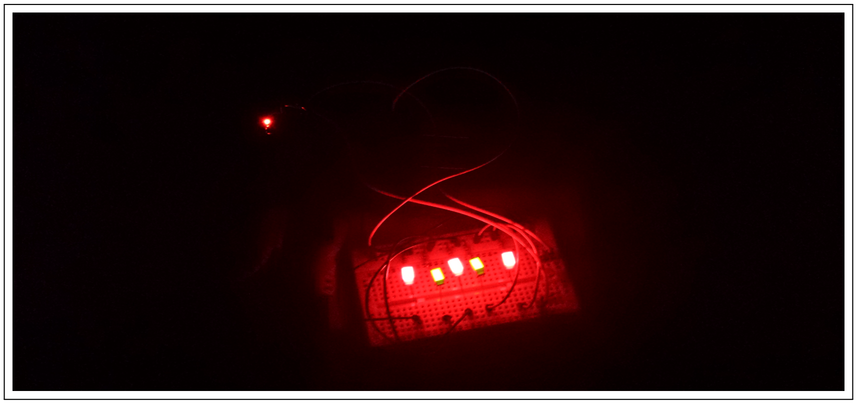

Figure 6.4: Leds turned on in the dark

lag that anyone could reach so far, and the Raspberry Pi community and developers are working on it, improving libraries of python camera and other related packages, to improve the video streaming to zero lag on any network.

Figure 6.5: Motor Driving Subsystem with 4 optocouplers

Results of Implementation

The system was able to move the 2 DC-Motors successfully according to the accelerometer readings in the four directions: left, right, forward, and backward in a controllable speed. Moreover, the webcam successfully sent live video streaming to the android phone (with two seconds lag) with a 15 fps precision, and the sensors worked perfectly: if darkness was detected according to a tested threshold of a time counter that counts till the capacitor discharges and sets pin 40 high-more details in the code, the system will automatically turn on lights on the car (we chose the threshold to be 1650 ms). In addition, if fire was surrounding the car, the flame sensor would successfully detect it and directly alarm the user on his device. All the data transmission and receiving happened through a Wi-Fi network according to the IEEE 802.11n protocol. An android application was developed to be the main interface for the user to drive the car, view the video, and receive notifications of fire detection. Thus, all expected results and system requirements were achieved.

Chapter 7

Conclusion

Bringing It All Together

This project- as we have stated earlier, aims to replace the traditional radio frequency technology to remotely control objects, with a modern one: Wi-Fi; which is considered to be cheap, widely available, has a wide range, and more consistent.

More challenges were added to the idea to make it even more appealing: live video streaming was chosen to the requirements, and sensors to detect dark and fire were added, which are both typical events that may occur in any environment that the car or any similar model, might be or used in.

Tests were run on each subsystem individually, and then combined all together in the end to form a full functional system that can move motors, detect fire and darkness, and stream video back to the phone; the following results were achieved:

1. 2 DC-Motors were moved successfully according to the accelerometer readings in the four directions: left, right, forward, and backward in a controllable speed.

2. Live streaming video was sent successfully through the network back to the phone from the webcam on the car with 2 seconds lag.

3. If fire was surrounding the car, the flame sensor detected it and sent a notification to the phone.

4. If the darkness is detected by the LDR, leds on the car are turned on.

5. An android application Picar is developed to be the main user interface on the driving phone and the car.

During the development and implementation stages, despite that there were serious challenges and obstacles regarding the hardware pieces and software implementation, we yet managed to overcome them all. New programming languages were learned, we tested many options before deciding on using python as the main language for our software and socket programming, such as java, and node.js; we became more familiar with linux environment and shell scripting. Definitely more experts with wiring and dealing with hardware pieces.

Lastly, the Internet has countless resources on the Raspberry Pi and how to implement such project with more countless suggested methods and different approaches, we tried many of them: some were easy, some were complex, some worked flawlessly, others did not. We had to choose the best and more convenient approaches for us and for future designers and developers.

Future Work

- Attach more than one camera to the car, and thus, give a full view of the four directions of the car surrounding.
- Work on improving the video streaming lag over networks.
- If 3G and 4G were ever enabled in West-Bank, better and more flexible communication options may be used between the Pi and the android device, such as setting the phone as a Wi-Fi hotspot and configure the Pi to connect to it directly, instead of being required to connect to a specific area Wifi network and stick to it.
- More sensors can be attached to the Pi, the only limit is the user desired options.
- The accelerometer readings of the phone may be more customized to allow the user to control speed more easily and efficiently.
- An iOS application may be developed for iOS devices such as iPhones and iPads, according to the sensors available in these devices: Gyroscope may be used instead of the accelerometer.
- Use Servo Motors instead of or in addition to the DC-Motors in the car.
- Extend this project concept and apply it to building remote controlled boats, drones or even robots.

Appendix A

Raspberry Pi Model B+

The Raspberry Pi Model B+ incorporates a number of enhancements and new features. Improved power consumption, increased connectivity and greater IO are among the improvements to this powerful, small and lightweight ARM based computer.

- **Specifications:**
  - **Chip:** Broadcom BCM2835 SoC .
  - **Core architecture:** ARM11.
  - **CPU:** 700 MHz Low Power ARM1176JZFS Applications Processor.
  - **GPU:** Dual Core VideoCore IV® Multimedia Co-Processor
    Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode.
    Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
  - **Memory:** 512MB SDRAM
  - **Operating System:** Boots from Micro SD card, running a version of the Linux operating system
  - **Dimensions:** 85 x 56 x 17mm
  - **Power:** Micro USB socket 5V, 2A
- **Connectors:**
  - **Ethernet:** 10/100 BaseT Ethernet socket

- **Video Output:** HDMI (rev 1.3 1.4)
  Composite RCA (PAL and NTSC)
- **Audio Output:** 3.5mm jack, HDMI
- **USB:** 4 x USB 2.0 Connector
- **GPIO Connector:** 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip
  Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
- **Camera Connector:** 15-pin MIPI Camera Serial Interface (CSI-2)
  JTAG Not populated
- **Display Connector:** Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
- **Memory Card Slot:** SDIO[2]

Appendix B

RC-Circuit

As explained in the darkness detection circuit, a resistor is placed in series with a Capacitor. When a voltage is applied across these components the voltage across the capacitor rises. The time it takes for the voltage to reach 63% of the maximum is equal to the resistance multiplied by the capacitance. When using a Light Dependent resistor this time will be proportional to the light level. This time is called the time constant :

$t = RC$

where t is time,
R is resistance (ohms)
and C is capacitance (farads)
So the trick is to time how long it takes a point in the circuit the reach a voltage that is great enough to register as a "High" on a GPIO pin. This voltage is approximately 2 volts, which is close enough to 63% of 3.3V. So the time it takes the circuit to change a GPIO input from Low to High is equal to 't'.

With a 1Kohm resistor and a 1uF capacitor t is equal to 1 millisecond. In the dark the LDR may have a resistance of 1Mohm which would give a time of 1 second.

In order to guarantee there is always some resistance between 3.3V and the GPIO pin, a 2.2Kohm resistor is inserted in series with the LDR.[8]

Appendix C

motionEye with Raspberry Pi

motionEye is a web frontend for a Linux video surveillance program called motion. This tutorial will walk you through the necessary steps to install and configure motionEye on a Raspberry PI. You will build yourself an intelligent video surveillance system based on affordable components and opensource software.

motionEye Features

- web interface, responsive design
- user/password protection (administrator and normal user)
- mjpg streaming
- motion detection, output to jpeg and avi files
- timelapse capturing
- browsing, previewing and downloading of media files
- advanced camera settings

Required Hardware

- a Raspberry PI
- a micro USB power supply (5V, 1A)

- an SD memory card (at least 4GB)
- an USB webcam or CSI camera board
- an USB WiFi adapter (optionally, if an ethernet connection is not available)

motionEye requires a few libraries and extra programs. Install them with the following command:

```
sudo aptitude install python-tornado python-jinja2 python-imaging mot
```

Installing motionEye

Choose the latest version from the downloads list in:
"https://bitbucket.org/ccrisan/motioneye/downloads"
(use the Tags tab) and download it to /home/pi. Unpack (replacing xyz with the code in the filename):

```
cd /home/pi
tar zxvf ccrisan-motioneye-xyz.tar.gz
mv ccrisan-motioneye-xyz motioneye
cd motioneye
```

After this step you'll have motionEye installed in /home/pi/motioneye, your current directory.

Configuring motionEye

Create a settings.py file from the existing template:

```
cp settings_default.py settings.py
```

Runnig motionEye

motionEye does not need root privileges to run; it can be started directly from the directory where it was extracted:

```
./motioneye.py
```

If everything was properly installed and configured, motionEye should emit an info message saying that the server started. Now point your browser to http://raspberrypi:8765 (replacing raspberrypi with the IP adress of your device). The motionEye web interface should show up. Use admin with no password, when prompted for authentication. Start by adding a new camera and feel free to experiment with the various available settings.

When you're done "experimenting", hit

```
ctrl-c
```

[10]

For more details, refer to "http://www.howtoembed.com/projects/raspberry-pi/95-motioneye-with-raspberry-pi"

References

[1] Sensors. http://en.wikipedia.org/wiki/Sensor. Accessed: October 21, 2014.

[2] Raspberry pi model b+. https://www.adafruit.com/products/1914. Accessed: September 15, 2014.

[3] Android operating system. http://en.wikipedia.org/wiki/Android $_{Accessed:October 21,2014.}$

[4] Wi-fi. http://en.wikipedia.org/wiki/Wi-Fi. Accessed: October 21, 2014.

[5] What is the range of a typical wi-fi network? ://compnetworking.about.com/cs/wirelessproducts/f/wifirange.htm.

[6] Smartphones. http://en.wikipedia.org/wiki/Smartphone. Accessed: October 21, 2014.

[7] Rpi.gpio 0.5.2a now has software pwm – how to use it. http://raspi.tv/2013/rpi-gpio-0-5-2a-now-has-software-pwm-how-to-use-it. Accessed: October 1, 2014.

[8] Reading analogue sensors with one gpio pin. http://www.raspberrypi-spy.co.uk/2012/08/reading-analogue-sensors-with-one-gpio-pin. Accessed: November 1, 2014.

[9] Controlling a dc motor with the raspberry pi. https://www.youtube.com/watch?v=W7cV9$_W 12sM. \quad Accessed \quad : January 21, 2015.$

[10] motioneye with raspberry pi. http://www.howtoembed.com/projects/raspberry-pi/95-motioneye-with-raspberry-pi. Accessed: March 1, 2015.

[11] Push notifications with raspberry pi. http://videos.cctvcamerapros.com/digital-io-alarm-in-out/send-push-notifications-from-raspberry-pi.html. Accessed: April 15, 2015.

[12] using flame sensor with the raspberry pi. https://www.raspberrypi.org/forums/viewtopic.php?t=60707p=453093. Accessed: April 15, 2015.

[13] Awesome raspberry pi iphone wifi radio controlled car toy with pov video. https://www.youtube.com/watch?v=JSP6VKiU7F4. Accessed: September 15, 2014.