Car Automation

Prepared by

Mit Patel (15IT093)

Under the supervision of

Prof. Mrudang Pandya

A Report Submitted to
Charotar University of Science and Technology
for Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology
in Information Technology
IT318 Software Group Project-III (5th sem.)

Submitted at

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY
Chandubhai S. Patel Institute of Technology
At: Changa, Dist: Anand – 388421
November 2017

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**Car Automation**" is a bonafied work carried out by **Mit Patel (15IT093)** under the guidance and supervision of **Prof. Mrudang Pandya** for the subject **Software Group Project-III (IT318)** of **5th** Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Mrudang Pandya.
Assistant Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Prof. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

Chandubhai S Patel Institute of Technology

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

Acknowledgement

I fall short of words to pen down anything for the co-operation and encouragement extended by parents through the completion of this project. A more thanks is little for their love, encouragement and support.

I hereby have the utmost privilege to thank my Software Group Project teacher Prof. Mrudang Pandya for his constant support and guidance. I would also like to thank Dr. A.D. Patel, the principal and Prof. Parth Shah, the head of department for giving me this opportunity to present this project. Last but not the least I would like to thank all my friends and my colleagues for encouraging me to put my best efforts in this project. I am making this project not only for marks but also to increase my knowledge.

THANKS AGAIN TO ALL WHO HELPED ME.

-Mit Patel (15IT093)

Abstract

In this Software Group Project-3 I want to make Embedded System project, in which I am trying to make system for Car Automation (level2) which we can use in future.

We use intelligent instruments in every part of our lives. It won't take much time that we realize that most of our tasks are being done by electronics. Very soon, as we shall see, they will perform one of the most complicated tasks that a person does in a day, that of driving a vehicle.

SAE, the Society of Automotive Engineers, defines six levels of automation, starting at Level 0 for no automation, and leading up to Level 5 for full automation.

Level 1 (The simplest automated tasks), with features including adaptive cruise control and parking assistance, where vehicles are capable of performing specific tasks but a driver is still required to take control at any time.

Level 2 is called partial automation, and this is the automation level that can be seen in the market today with the Mercedes-Benz E-Class or Tesla S. It's a system capable of driving the vehicle by itself under certain conditions and on certain roads, but the driver still needs to supervise what the car does. The driver, and not the OEM, is accountable for road safety, which makes it much easier for an OEM to deploy such a system, as less validation is needed. However, it's important that the driver understands that he or she needs to supervise the car even if it feels as though the vehicle can handle everything.

In which we include concepts of **Python, PHP, HTML Programming, Raspberry Pi and Sensors** so we can learn about these concepts.

Chapter 1: Introduction

1.1 Project Summary:

- Car Automation provides wireless accessibility to user of car functionalities like turning headlights on/off, opening/closing doors, driving a car using wireless controller using the power of web, etc. Main use of this project can be to provide an Artificial Intelligence way of interacting/controlling with car (e.g. Autonomous Car).

1.2 Scope:

- For autonomous cars access to physical interface of car to implement their AI is going to be very helpful for everyone.
- This helps in a way so we can focus more on working on AI rather than struggling to build physical interface for our AI.

1.3 Objective:

- To access facilities of car easily with any wireless device using power of web.
- To provide physical interface for AI to interact with car.

Chapter 2: Software and Hardware requirement

2.1 User characteristics:

- User should know how to access a web application on any wireless device that he prefers.

2.2 Tools and Technology used:

Hardware:

- Raspberry Pi 3 Model B
- SD card
- Jumper cables (M2F)
- LED/LCD Screen
- Relay
- L293D motor controller circuit

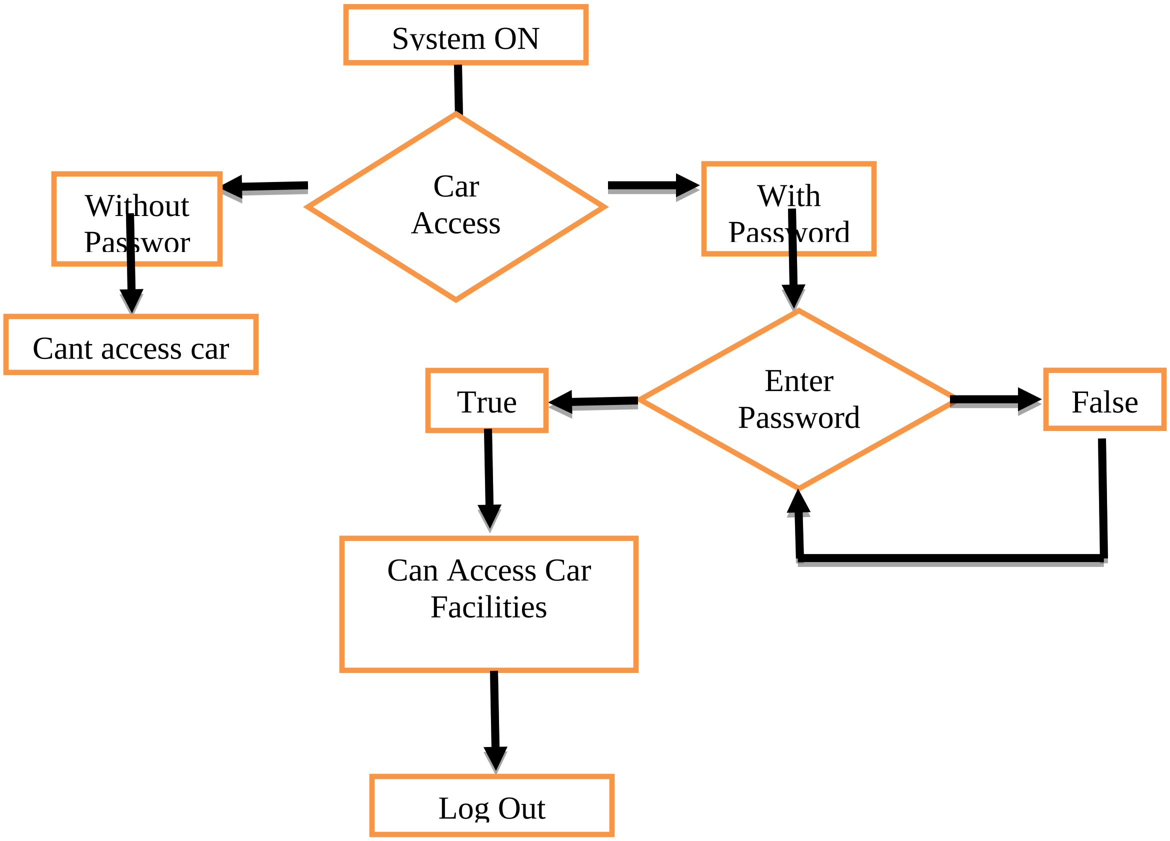
Software:

- WinSCP (Windows) / SCP (Linux)
- PuTTy (Windows) / OpenSSH (Linux)
- XRDP
- PHP 7.0 or greater, Python 3.5 or greater, Apache2, Ajax

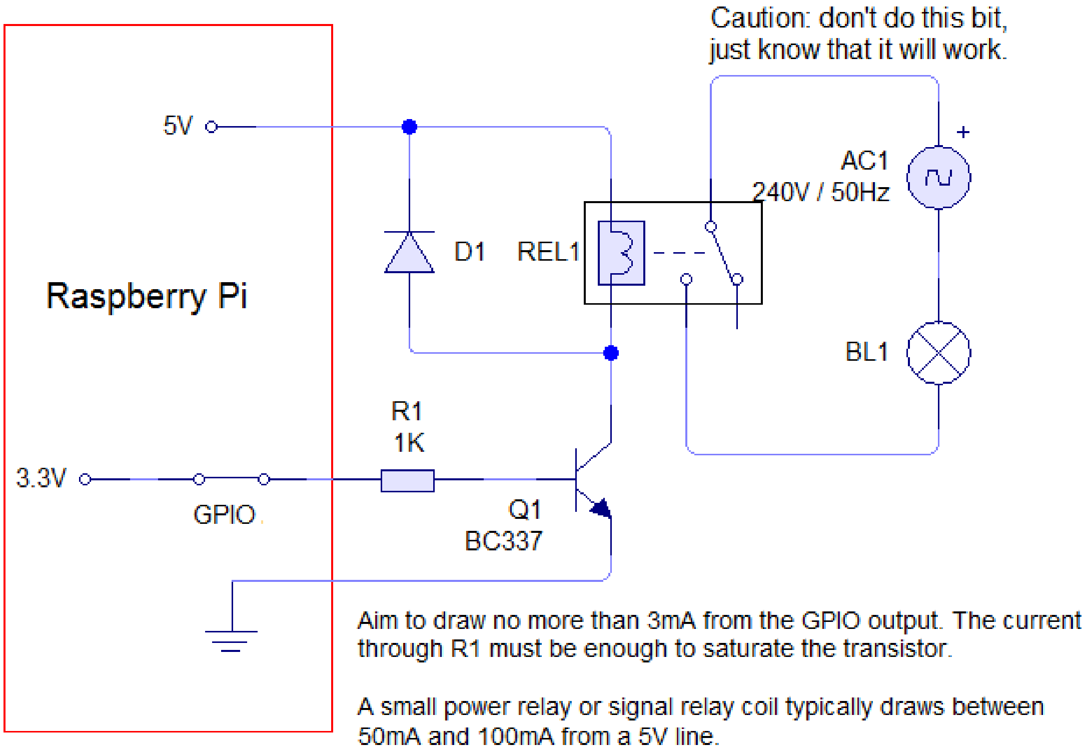
Chapter 3: System Design

3.1 Project Flow:

Flow Chart I (User Interface):



Flow Chart II (Relay):



Caution: don't do this bit,
just know that it will work.

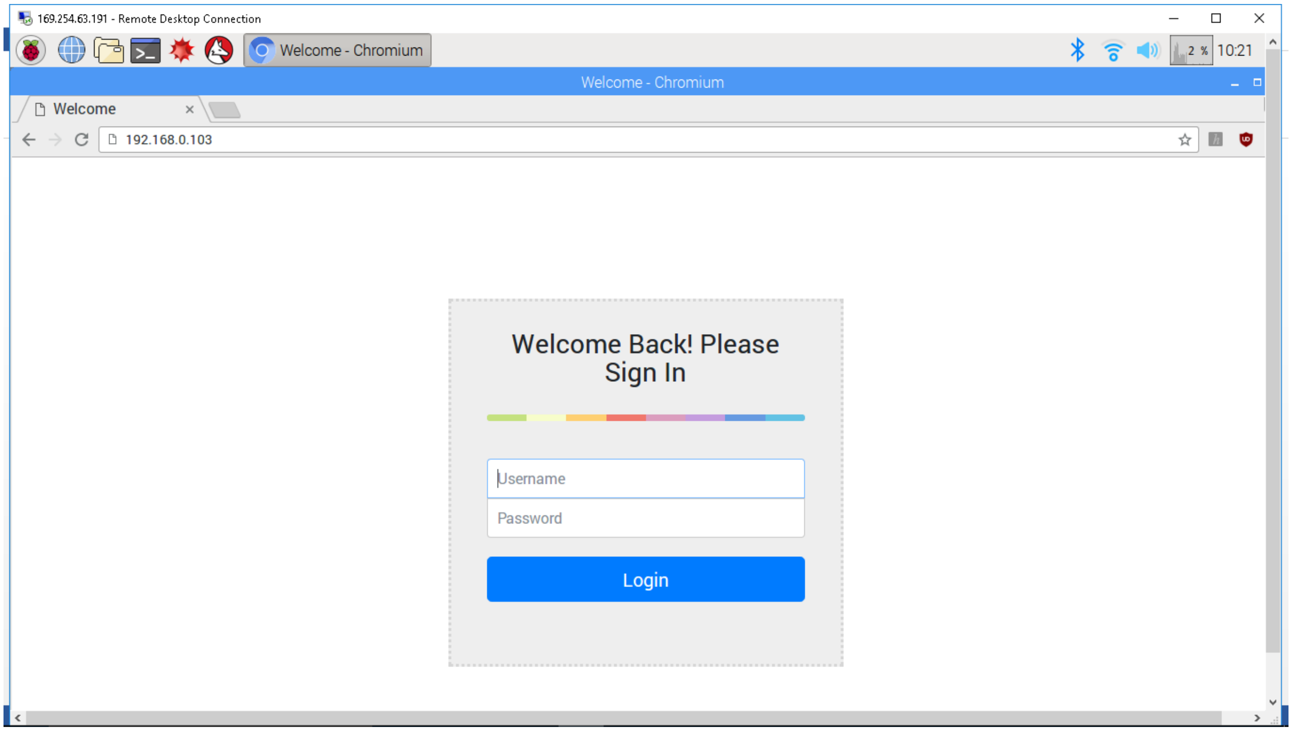
Aim to draw no more than 3mA from the GPIO output. The current through R1 must be enough to saturate the transistor.

A small power relay or signal relay coil typically draws between 50mA and 100mA from a 5V line.

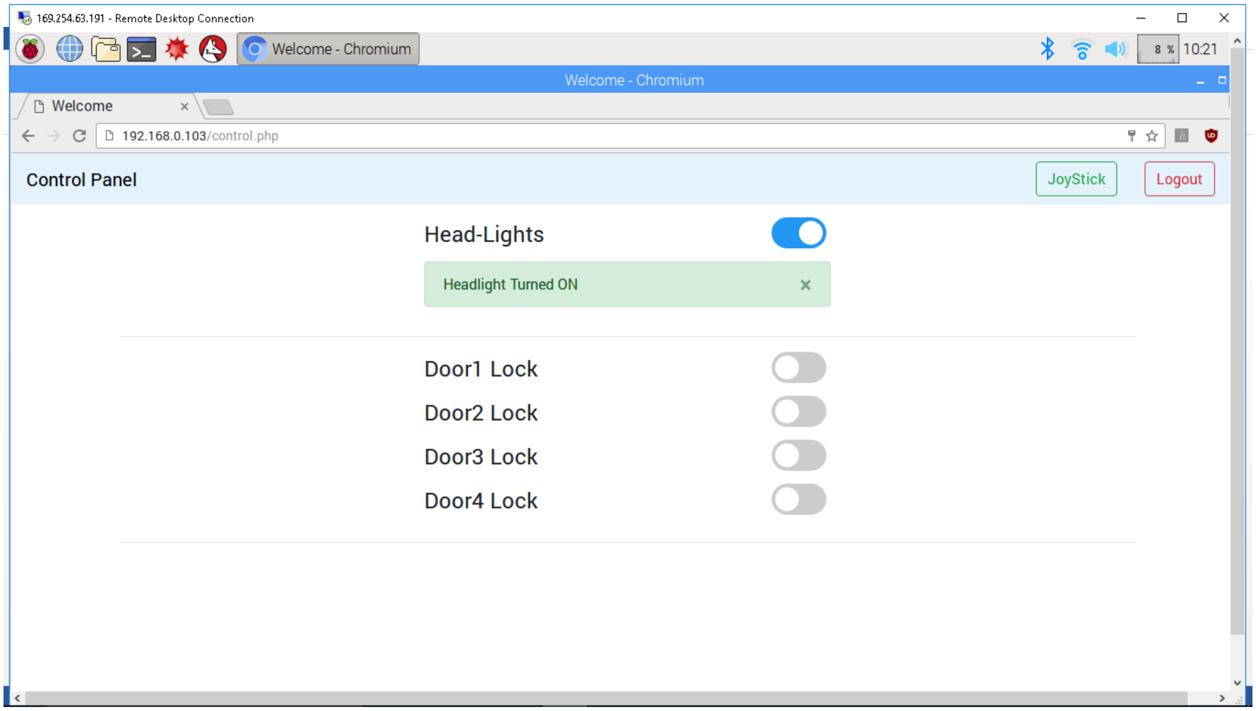
3.2 Major functionality:

- Provides physical interface for an AI.
- User can control his/her car remotely (e.g. Parking).

3.3 GUI forms:



Home page: To Log into system.


169.254.63.191 - Remote Desktop Connection
Welcome - Chromium
8 %
10:21
Welcome - Chromium
Welcome
192.168.0.103/control.php
JoyStick
Logout

Control Panel

Head-Lights ●

Headlight Turned ON ×

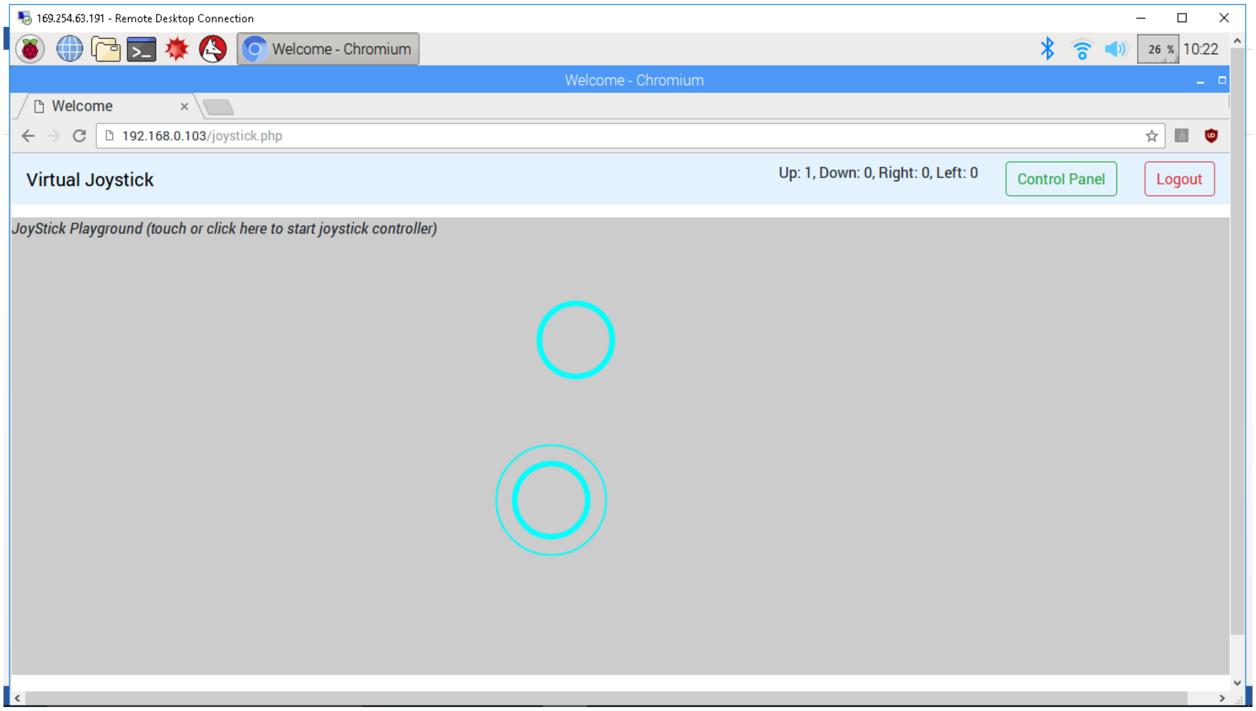
Door1 Lock ○

Door2 Lock ○

Door3 Lock ○

Door4 Lock ○

After Successful login Control page.



Virtual Joystick

Up: 1, Down: 0, Right: 0, Left: 0

JoyStick Playground (touch or click here to start joystick controller)

Joystick to control car.

Chapter 4: Implementation Planning

4.1 Implementation Environment:

- As this project uses login forms for user, owner can create multiple users and allow them to access car.
- All allowed users would have accessibility to car. This might cause problem because of multiple access of car functions at the same time, we've tested this issue and it doesn't affect system in any way but experience of user.
- Project uses apache2 as server and web interface for interacting with system.

4.2 Module Specifications:

- Our control panel page (used to control headlight, doors etc) uses AJAX for sending GET/POST request to server that runs PHP as scripting language for handling requests on raspberry pi and shows success or failure message after getting response from server.
- Joystick page uses javascript canvas for animations and to simulate real time joystick controller for car and also shows live results on the page using AJAX.
- In PHP, Command line utility of Wiring PI is used to handle GPIO on/off requests as quick as possible.
- Web interface uses bootstrap for styling which is responsive meaning it works on variety of screen sizes (e.g. mobile, tablet, laptop).

Working:

- When we open website we have to login first, if login is successful then we can see Control page and if login not successful we have to try again.
- Now after successful login we can see many options are available to control Head light, Door lock, etc. By that buttons we can operate functionalities of car. When we on/off any button from wireless device it will give signal to Rpi and it will run cade accordingly.
- We also have Joy stick to control car, its same as remote control car.

4.3 Snapshot of project:

- **Connections :**

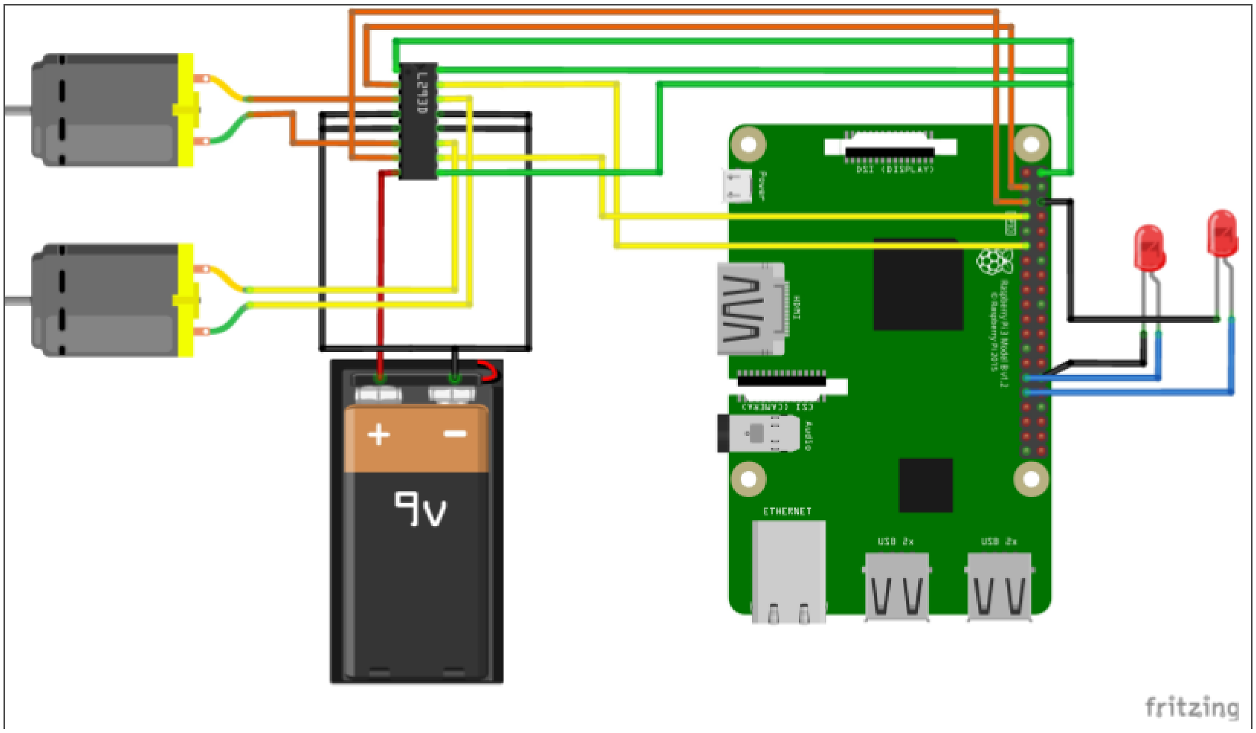


Fig.4.1 Connections (Pin no. as board)…

GPIO as BOARD
GPIO 3 and 5 - Control Motor1
GPIO 7 and 11 - Control Motor2
GPIO 29 and 31 - Control Head lights

4.4 Coding standards:

```
<?php                                          //Control.php
  session_start();
  // checks for session for security purpose.
  if(isset($_SESSION['raspi_carauto_admin_name'])) {
?>
// if true then show following content to user else go to login page
// checkbox calls javascript function when clicked that requests the
server
<input type="checkbox" onclick="headlightAjax()" id="headlight">

<?php
      } else {
            header("Location: ./index.html");
        exit;
      }
?>



<?php                                          //joystick.php
  session_start();
  // checks for session for security purpose.
  if(isset($_SESSION['raspi_carauto_admin_name'])) {
?>
// if true then allow user to interact with joystick controller else
send back to login page
  <div class="container-fluid">
    <div class="row">
      <div class="col-12" id="joy_container">
        <i><p>JoyStick Playground (touch or click here to start
joystick controller)</p></i>
      </div>
    </div>
  </div>

<?php
    // set pin modes
    $setmode3 = shell_exec("/usr/local/bin/gpio mode 8 out");
    $setmode5 = shell_exec("/usr/local/bin/gpio mode 9 out");
    $setmode7 = shell_exec("/usr/local/bin/gpio mode 7 out");
    $setmode11 = shell_exec("/usr/local/bin/gpio mode 0 out");
    $setmode13 = shell_exec("/usr/local/bin/gpio mode 2 out");
    $setmode15 = shell_exec("/usr/local/bin/gpio mode 3 in");

  } else {
            header("Location: ./index.html");
        exit;
      }
```

```
?>

#!/usr/bin/env python                                          //headlight.py
# exception handling
try:
    # importing required libraries
    import RPi.GPIO as GPIO
    import time
    from sys import argv
    GPIO.setwarnings(False)

    headlight 1= 29
    headlight 2= 31

    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(headligh1t, GPIO.OUT)
    GPIO.setup(headlight2, GPIO.OUT)

    # storing command line arguement for output logic
    OutputLogic = bool(int(argv[1]))
    GPIO.output(headlight1, OutputLogic)
    GPIO.output(headlight2, OutputLogic)

    print("Headlight Turned {}".format("ON" if OutputLogic else "OFF"))

except:
    print("Something went wrong!")




<?php                                               //headlightAjax.php
  // handles ajax request from javascript as per sent data
  if($_GET['action'] == 'TURN_ON_HEADLIGHT'){
    // for sudo to work we will need to add apache username to sudoers
file
    // sudo echo "apache_username ALL=(ALL) NOPASSWD: ALL" >>
/etc/sudoers
    // executes shell command and prints output
    $command=escapeshellcmd("sudo ../pyscripts/headlight.py 1");
    $output=shell_exec($command);
    echo $output;
  } else if ($_GET['action'] == 'TURN_OFF_HEADLIGHT') {
      $command=escapeshellcmd("sudo ../pyscripts/headlight.py 0");
      $output=shell_exec($command);
      echo $output;
    } else {
      echo "Something went wrong!";
    }
?>
```

```
<?php                                     //joystickAjax.php
  // setting variables as per requested data
  $up = ($_GET['up'] == 'true') ? 1 : 0;
  $down = ($_GET['down'] == 'true') ? 1 : 0;
  $right = ($_GET['right'] == 'true') ? 1 : 0;
  $left = ($_GET['left'] == 'true') ? 1 : 0;

  // python
  // $command=escapeshellcmd("sudo ../pyscripts/joystick.py ".$up."
".$down." ".$right." ".$left);
  // $output=shell_exec($command);
  // install wiring pi by executing following set of commands
  // `git clone git://git.drogon.net/wiringPi`
  // `cd wiringPi`
  // `./build`

  // uses WiringPi command line utility to handle GPIO requests
 $upOutput=shell_exec("/usr/local/bin/gpio write 8 ".$up);
 $downOutput=shell_exec("/usr/local/bin/gpio write 9 ".$down);
 $rightOutput=shell_exec("/usr/local/bin/gpio write 7 ".$right);
 $leftOutput=shell_exec("/usr/local/bin/gpio write 0 ".$left);
 echo "Up: ".$up.", Down: ".$down.", Right: ".$right.", Left: ".$left;
?>
```

```
<?php                               //login_session_create.php
  // creates session if username and password is authenticated.
  session_start();
  session_regenerate_id(true);
  $_SESSION['raspi_carauto_admin_name']=$_POST['Username'];
  $_SESSION['raspi_carauto_admin_passcode']=$_POST['Password'];

  if(isset($_SESSION['raspi_carauto_admin_name']))
  {
    include 'credentials.php';
      if($_SESSION['raspi_carauto_admin_name'] == $uname &&
$_SESSION['raspi_carauto_admin_passcode'] == $passcode)
      {
      header("Location: ../control.php");
      exit;
    }
    else {
      header("Location: ./login_session_destroy.php");
      exit;
    }
  }
?>
```

```
<?php                               //credentials.php
```

```
  $uname = 'admin';
  $passcode = '123';
?>



<?php                              //login_session_destroy.php
  // Initialize the session.
  // If you are using session_name("something"), don't forget it now!
  session_start();

  // Unset all of the session variables.
  $_SESSION = array();

  // If it's desired to kill the session, also delete the session
cookie.
  // Note: This will destroy the session, and not just the session
data!
  if (ini_get("session.use_cookies")) {
      $params = session_get_cookie_params();
      setcookie(session_name(), '', time() - 42000,
          $params["path"], $params["domain"],
          $params["secure"], $params["httponly"]
      );
  }

  // Finally, destroy the session.
  session_destroy();

  header("Location: ../index.html");
  exit;
?>
```

Chapter 5: Limitations and Future Enhancement

Limitation:

- We must have internet connection in car as well as in mobile.
- If any fault occurs in circuit then that functionality wouldn't work, so we have to suffer from that.
- More power usage, so less battery life.
- In critical situation isn't useful.
- There's always a risk of getting system malicious or hacked.

Future Enhancement:

- This system can be used as physical interface to AI ,with this automation platform we can give base for AI, so system interact with car very efficiently and faster, Of course we need to be worry about how circuit failures can be handled and also how to nullify critical situations, these all can be future enhancements for this project.

Chapter 6: Conclusion

From this project We have tried to make such a system which is trending nowadays. It’s a base for self-driving car, It’s more accurate than humans, faster, more efficient, more comfortable and might get more reliable over time.

So, by this project we learn how to make any small object autonomous and combine PHP, Python, JS and Ajax together to work as a fluent system. This also helps to learn about internet of things and also connect them together so they can communicate with other devices to execute different tasks.

References

http://www.hellointernetofthings.com
http://www.raspberrypi.org/Downloads
http://hellorapberrypi.blogspot.com
http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html
http://circuitbasics.com
https://www.youtube.com
https://en.wikipedia.org/wiki/Autonomous_car
https://www.google.co.in/search?q=L293d+connections&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjh8rfjtaLXAhWJ6Y8KHTZIBmYQ_AUICigB&biw=1438&bih=727
http://learn.jquery.com/about-jquery/
http://wiringpi.com/pins/
http://v4-alpha.getbootstrap.com/layout/grid/
http://jeromeetienne.github.io/virtualjoystick.js/examples/basic.html