Release Manual

PES Institute of Technology
Department of Electronics and Communication

Driverless Car

Rakshith Vishwanatha
Aditya T
Pavan Gurudath

Project:	Driverless Car (Using Machine Learning and OpenCV)
Department:	*Electronics and Communication, Electrical and Electronics Engineering*
Date:	*August 4th, 2015*
Written by:	*Rakshith Vishwanatha*

Review and approval				
Name	**Designation**	**Signature**	**Date**	**Function**

Document History			
Version	Date when completed	Date if modified	Reason for Change
1.0	03/08/2015	04/08/2015	Addition of a Few Specifications and Changes in Operating Environment
1.0	28/08/2015	28/08/2015	Correction of Motor and Power Supply Ratings and Requirements

Contents

1 Scope

This document has been created to define the user requirements for the purchase of Artificial Intelligence Drive (AID) Car for use as a driverless car on flat, one way, paths not containing any traffic*.* As identified by relevant guidelines:

1. The user must have a valid Driver Licence, as there may be situations where the car needs to be used manually
2. Used on flat white roads with a black strip representing the path to be followed. The path thickness needs lie between 1.5cm and 1.8 cm. The path is allowed to have a maximum inclination of 5 degree, and there cannot be a change its inclination at any point.
3. The lighting conditions need to be in the range of 10 lx to 640 lx.
4. No obstructions or obstacles are to be allowed in the area of view of the camera.(Area of view has been defined in the Release Manual under section 4.1).
5. Temperature can vary in the range 0decC to 40degC.
6. The product has been tested for use only indoors, and may not fare well in outdoor gust, rain, or snow.
7. The connecting wires shall not be tampered with by customer

1.1 Objectives

The objective of this product is to help the user in his/her commute. It has the ability to see the path lying ahead of it and decide the direction that it needs to traverse. The flat, black and white path that has been made, acts as the road that the car is to navigate. The car also gives the user the liberty to shift between automatic and manual mode at any given point of time.

1.2 Benefits

The purchase of the product would benefit the user in the following manner:

1. Help the user in his commute, by taking over the driving, allowing the user to relax, talk to other passengers of the car or do some other work
2. Reduce cramping and ache in the shoulder and knee due to long distance driving
3. Allow transport of goods between cities on well defined highways without the need for a full time driver

2 Terminology

This document refers to the features and attributes the product should have and how it should perform from the user perspective. It should describe what the proposed system will do, but nothing about how it will be built or implemented.

Narrative descriptions and diagrams can be used to show the services the proposed system will deliver and its limitations.

The document can be used as a “contract” between the client and the equipment supplier who will provide and install the system.

Key words used within the user manual include “must”, “must not”, “should”, “should not”, “recommended”, and “may” or “optional” and are to be interpreted as follows:

Must this word means that the definition is an absolute requirement of the specification.

Must not This phrase means that the definition is an absolute prohibition of the specification.

Should This word or the adjective “Recommended” means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

Should not This phrase means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful but the full implications should be understood and the case carefully weighed.

May This word, or the adjective “optional” means that an item is truly optional

3 Regulations

1. Voltage requirement of 230V AC at mains and 5V/12V DC supply to car’s H-bridge must be met
2. The user must have a valid Driver Licence, as there may be situations where the car needs to be used manually
3. Used on flat white roads with a black strip representing the path to be followed. The path thickness needs lie between 1.5cm and 1.8 cm. The path is allowed to have a maximum inclination of 5 degree, and there cannot be a change its inclination at any point.
4. The lighting conditions need to be in the range of 10 lx to 640 lx.
5. No obstructions or obstacles are to be allowed in the area of view of the camera.(Area of view has been defined in the Release Manual under section 4.1).
6. Temperature can vary in the range 0decC to 40degC.
7. The product has been tested for use only indoors, and may not fare well in outdoor gust, rain, or snow.
8. The connecting wires shall not be tampered with by customer

4 Use

This product will be used to help the user in his/her commute. It has the ability to see the path lying ahead of it and decide the direction that it needs to traverse. The flat, black and white path that has been made, acts as the road that the car is to navigate. The car also gives the user the liberty to shift between automatic and manual mode at any given point of time. It will benefit the user by allowing him/her to relax during the journey avoiding the need of being attentive the whole way.

The local environment consists of a flat black path on a white sheet, in the temperature range of 20 degC to 35degC, with lighting conditions of 10 lx to 640 lx. The purpose of this System in the local environment is to act as a proof of concept for driverless cars.

4.1 Specification

1. Car
   a. Dimensions (lxbxh): 24.5cms x 14.5 cm x 11 cm
   b. Min Radius of Turning: 56cm or greater
   c. Motors: Mabuchi Motors, 6V, 1A
   d. H-bridges: L293D, 4.5V-12V input, output, with peak output current: 1.2A
   e. Lights: CE ROHS, operating voltage:12V

2. Camera
   a. Model: Microsoft Lifecam hd 6000
   b. Height and Inclination of placement: 12cm, 45deg to the normal (inclined forward)
   c. Field of View: 66deg diagonal view
   d. Area of View: blind spot for 6cm from front of car, extends to 51cm from front end of car, extends to 15cm on either side. Total area of view = 675 sq. cm
   e. Resolution 1280x800 sq. pixels
3. Path
   a. Strip thickness: 1.5 cm to 1.8 cm
   b. Colour: Black strip on White background
   c. Allowed Radius: 56cm or greater
   d. Allowed inclination of path: 5 degrees
   e. Clearance area: Same as Area of view of camera = 675 sq. cm
4. Power Supply
   a. Mains: 230 volt 50/60 Hz to adapter of the ULK board, PC, and Voltage regulator
   b. Car: 9V to h-bridge of front wheels and 5V to the h-bridges of the rear wheels.
5. Additional Specifications:
   a. Orientation: For the car to be able to drive automatically, it needs to be placed/driven manually such that the path is in the Field of View of the camera
   b. Lighting Conditions: 10 lx to 640 lx

4.2 Operational Requirements.

Operationally, the car needs to sense the path/road lying ahead and decide which direction to turn. Once it makes the decision, it needs to follow up, and actuate it. The path appearing ahead is sensed by using the camera mounted at the front. A coloured image is seen by the camera, but a black and white image is captured. It has a resolution of 1280x800 sq.pixels This is compressed to a 20x20 image. The image is then converted to its corresponding bitmap matrix form, where white pixels are stored as the 1's and black pixels are stored as 0's. The capturing of the image and its conversion to bitmap form uses OpenCV.

Once the image has been obtained in the bitmap format, the machine learning algorithm is used to decide the direction in which the car is to turn. This involves pixel-wise multiplication with the trained parameter values, and applying the sigmoid function to the sum of the multiplied result obtained. When this kind of multiplication is done for parameter trained for left, straight, and right turns, it yields the probabilities of the image representing the left, straight or right turn. The turn yielding the maximum probability is picked and that is the direction the car is made to actuate.

To actuate the car, we use the ULK board. The decision made is then sent to the board using USB port on the PC to the UART port on the ULK board. Sending the command from the PC is achieved using Libsub in synchronous bulk transfer mode. Messages sent and received are in the form of characters, where ‘l’ represents left turn, ‘r’ represents right turn and ‘s’ represents straight. In accordance to the character received by the ULK board, signals are sent through the GPIO ports to the h-bridges, which then drive the motors of the car.

Finally, once the motors are driven as desired, a confirmation message is sent back to the PC from the ULK’s UART port. This message allows the camera to take a new photo and make the next move.

Requirements:

i. *ULK board*
ii. *PC with Ubuntu (12.04 or 12.10)*
iii. *OpenCV (2.4.9)*
iv. *Libusb (1.0.9)*
v. Webcam (Microsoft Lifecam hd 6000)
vi. H-bridge (L293D)
vii. 12V, 9V and 5V supply
viii. 230V for laptop and board adapter

4.3 Process Requirements

First off, is image capture, done using OpenCV. The main Functions and API’s used form OpenCV, along with their reason of use are given in the table below:
(The order in which the contents of the table is organized is the general order in which they are used as the program is executed)

OpenCV libraries used:

i. Opencv_core
ii. Opencv_highgui
iii. Opencv_imgproc

Function or Variable Name	Description, Purpose of Use
CvCapture* c	Helps obtain image/video feed from the camera
IplImage* p , p1, p2, p3 cvCreateImage(int, int, int)	Pointers to various images captured by camera p – a variable using which images from the camera can be referred to p1 – a coloured 1280x800 sq. pixel image p2 – a black and white 1280x800 sq.pixel image p3 – a 20x20 compressed image of p2
cvGetSize(pointer)	Gets the size of the image pointed to
CvSize(int, int)	Creates an image of size mentioned
cvWaitKey(int)	Waits for set number of milliseconds for users input

cvShowImage(pointer)	Creates a window and displays the mentioned image in that window
cvSmooth(pointer, pointer, int, int, int)	Applies Guassian blur to the image to take the average values of a group of pixels while reducing an image's size
cvCvtColour(pointer, pointer, int)	Converts RGB values to HSV values
cvInRangeS(pointer, struct, struct, pointer)	Isolates the part of the image which falls in the given HSV range
cvReleaseCapture(pointer)	Stops getting the feed from the camera
cvReleaseImage(pointer)	Clears p1, p2, and p3. Thus stops displaying the images in windows
Mat mat (pointer)	Creates an array

Using the functions given above, we are able to obtain the camera feed and get it to the desired bitmap matrix form.

Then comes the Machine Learning Algorithm that is used to decide the direction in which the car needs to turn. Here we do pixel-wise multiplication with the trained parameter values, and applying the sigmoid function to the sum of the multiplied result obtained. When this kind of multiplication is done for parameter trained for left, straight, and right turns, it yields the probabilities of the image representing the left, straight or right turn. The turn yielding the maximum probability is picked and that is the direction the car is made to actuate. This is accomplished using simple, additions, multiplication, logarithmic values and 'for' loops.

(Explanation of the Machine Learning relating to creating a data base of images and training the system has been given in section 6.1 Calibration)

Finally, we have the Libusb stage where the signal is sent the ULK board for actuation. The main Functions and API's used form Libusb, along with their reason of use are given in the table below:

Libusb header file included used:

1. libusb.h

Function or Variable Name	Description, Purpose of Use
libusb **devs	Pointer to a list of devices connected to the PC
libusb_device_handle *dev_handle	Initializes a variable to point to some device handle
libusb_context *ctx	Creates a variable to point to the current session
libusb_init(pointer)	Begins a USB session
libusb_get_device_list(pointer, pointer)	Get all the devices that are connected to the PC during the session mentioned
open_device_with_vid_pid(pointer, int, int)	Stores the device handle in a pointer using the vendor and product ID of that device
libusb_kernel_driver_active(pointer, int)	Checks if the kernel has occupied the device handle mentioned
libusb_detach_kernel_driver(pointer, int)	Detaches the kernel

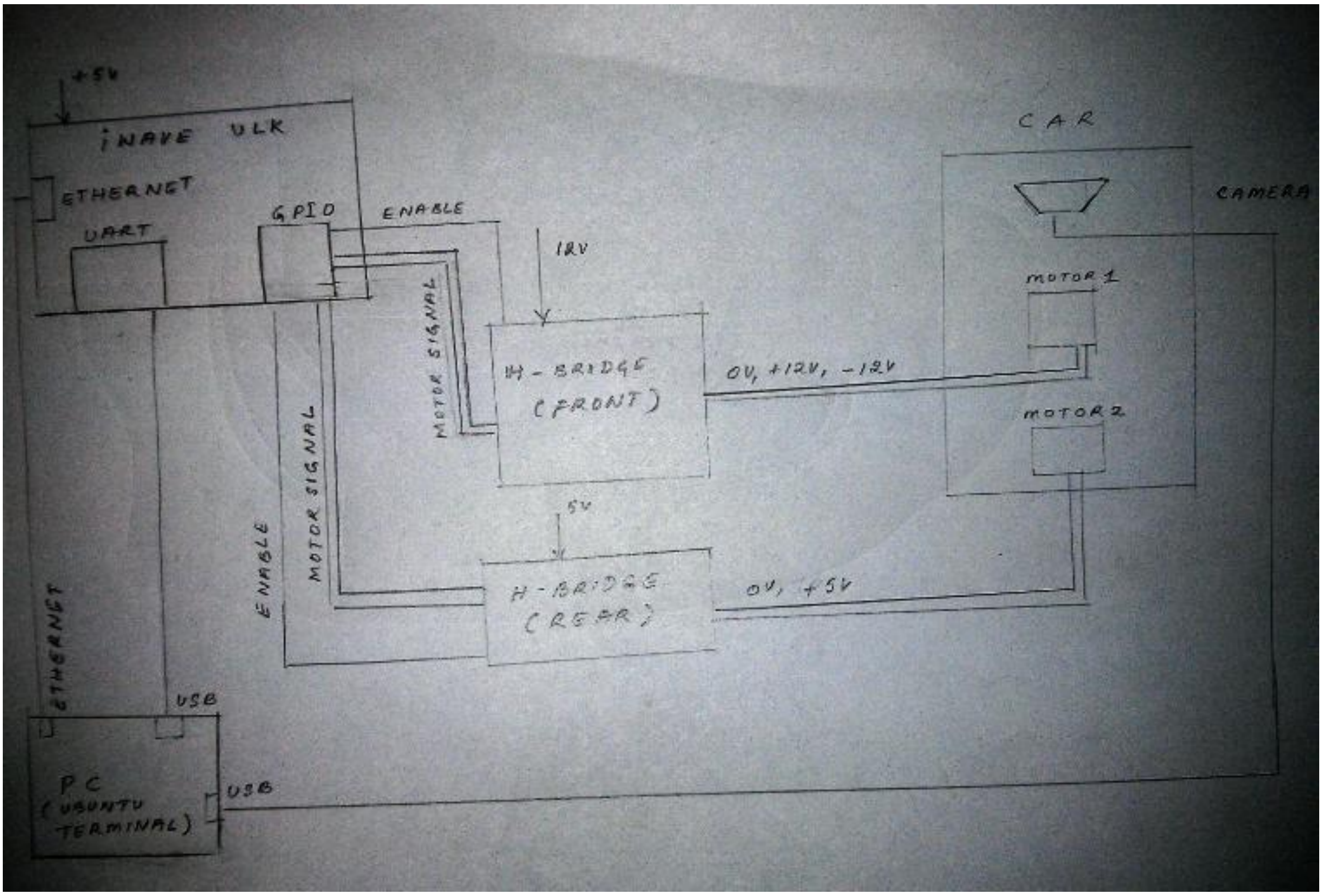
libusb_claim_interface(pointer, int)	Claims the device handle mentioned
libsub_bulk_transfer(pointer, int, pointer, int, pointer, int)	Sends or receives data (char type)through the USB port
libusb_release_interface(pointer, int)	Releases the device handle that it had claimed
libusb_close(pointer)	Clears the pointer holding the device handle
libusb_exit(pointer)	Destroys the USB session which had been created

Using the functions mentioned above, we are able to send data to the ULK board, which is then used to drive the car in the desired direction. By changing the endpoint's direction bits, we are also able to read the confirmation message sent from the board to the PC.

5 Functions

5.1 Operation

Hardware Setup:



Software Installation:

1. OpenCV
   i. Download the zip/tar folder for the required OS. We have used OpenCV 2.4.9 for Linux (Ubuntu) [opencv.org/downloads.html]

ii. Extract the contents into some known location. We have extracted it into /home/linux/Downloads/wipro/opencv-2.4.9
iii. Change directory to the location where you extracted the files
iv. Then, we need to install cmake. Install cmake from the terminal using sudo apt-get install cmake
v. Now, we create two directories in the opencv directory as follows
mkdir _build
mkdir _install
vi. We then change directory to the _build folder.
vii. Now we use the following code to install the minimum components required to by this application
cmake .. -DCMAKE_INSTALL_PREFIX=/home/linux/Downloads/wipro/opencv-2.4.9/_install -DWITH_IPP=OFF -DWITH_PTHREADS_PF=OFF -DBUILD_TESTS=OFF -DBUILD_PREF_TESTS=OFF -DBUILD_opencv_apps=OFF
viii. Then type the following commands
make
make install

ix. On running the OpenCV code, if you experience a problem where the terminal complains about not being able to find opencv .so files, then type the following in the terminal
export LD_LIBRARY_PATH=~/Downloads/wipro/opencv-2.4.9/_install/lib

x. Note, the above export command needs to be typed each time a new terminal session is opened. To make it a permanent change, modifications can be made to the .bashrc file
xi. This will setup OpenCV

2. Libusb
   i. Download the libusb tar file from the libusb sourceforge webpage. We have used libusb-1.0.9
   [http://sourceforge.net/projects/libusb/files/libusb-1.0/libusb-1.0.9/]

   ii. Extract the files into a known folder. We have extracted it into /home/linux/Downloads/wipro/libusb-1.0.9
   iii. Here, you will find a file named ‘configure’. It will create the required Makefile, which in turn will create all the libraries/binaries required. Change directory to the path where you extracted the contents
   iv. In the command window, now type; configure
   v. Now a Makefile will appear in the same directory. Thus type the command; make
   vi. This will setup Libusb
   vii. Note, to run libusb programs for data transfer, super user access maybe required.

3. Running the Code
   i. Once the car has been powered on (ULK board switched on, Power supplies are switched on) the usbuart.c needs to be uploaded to the board and run
   ii. On the PC, the wiproFinal code needs to be run. Since it utilizes libusb which requires root access, we need to the following
      1. Get root access; sudo su

2. Set the environment variable for OpenCV to access using the same export command given above

export LD_LIBRARY_PATH=~/Downloads/wipro/opencv-2.4.9/_install/lib

3. Now change directory to the place where the executable lies and run the executable using ./wiproFinal

5.2 Control System requirements

1. Camera: 1280x800 sq.pixel
2. Motors: 24V 1000 rpm motors
3. H-bridges: L293D, 4.5V-36V output, peak output current: 1.2A
4. Lights: CE ROHS, operating voltage:12V
5. Power Supply-Mains: 230 volt 50/60 Hz to adapter of the ULK board, PC, and Voltage regulator
6. Power Supply-Car: 9V to h-bridge of front wheels and 5V to the h-bridges of the rear wheels.
7. USB-UART cable connecting PC and board
8. Ubuntu terminal, ULK API

5.3 Power Failure/Recovery

If power is lost, the system will need to be driven manually. To get it to work in its fully functional automatic mode, the following steps will need to be followed:

1. Upload the usbuart.c code to the ULK
2. On the PC, change to the directory where the executable lies. In our case, /home/linux/Downloads/wipro/mystuff
3. Log in as root user using sudo su
4. Set the path variable using
   export LD_LIBRARY_PATH=~/Downloads/wipro/opencv-2.4.9/_install/lib
5. Rerun the executable on the terminal of the PC using the ./wiproFinal
6. When the program prompts you to, select the mode in which you would like it to work
7. This is will have it running properly

5.4 Recovery methods

The system may malfunction and need to be reset/recovered for various reasons:

1. Gets stuck due to Camera Malfunction:
   Some errors may arise with the camera connection. These errors are reflected as ‘select timeout’ errors. Ways to fix this error:
   i. Press the esc key and quit the program running on the PC. Reset the program.
   ii. If pressing the esc key doesn’t work, use ctrl+c to force quit the program running on the PC. Reset the program.
   iii. If none of the above work, as a last resort, press the esc / ctrl+c key and quit the program running on the PC. Then, disconnect the camera and reconnect it. Now rerun the wiproFinal program

2. Loss of files:

i. The executable file maybe lost or damaged due to mishandling of the file system. If that is the case, using the ‘make’ command to compile and re-create the executable will fix the problem
ii. The source file itself maybe lost. In this case, the the contents of the wipoFinal.backup file are to be copied and pasted into a file labelled wiproFinal.cpp Then, the ‘make’ command needs to be used to compile and re-create the executable
iii. If all files and code is missing, the Support persons mentioned in the User Manual can be contacted, to obtain the files. Once the source files have been obtained, step ii can be followed.

5.5 Alarms/Warnings

The main form of communication between the car and the user is the the direction of movement. The car outputs the direction it plans to turn on the LCD panel and then turns. If the user feels something is going wrong with the car’s decision, wants to go in a different direction he can switch to manual mode.

5.6 User interfaces

The user can interact with the system in the following ways:

1. The PC terminal:

   It is used both as an input and an output. The input that the system takes is the mode of operation of the car. Initially the user is asked which mode he/she would like the car to operate in. After making their initial choice, the user has the ability to switch modes at any given point in time.

   It shows the output in the sense, it displays the current mode in which the car is, displays other errors that may occur such as ‘select timeout’, and also displays an upcount, which can be used to find out if the system is in good health or is there exists a fault. The counter must continuously count up. If it gets stuck at a particular number for more than 3 seconds, it means there is malfunction in the system, and manual mode is the only mode available to the user (The system can of course, be reset to get full access to all the modes, as mentioned in the the recovery section).

2. The LCD panel of the ULK is used only as an output. It displays the systems state in brief. It shows the current operational mode and the desicion made by the car.

3. Other than the esc button which relies on the software to detect and disengage the car from automatic mode, hardware buttons/switches are also a means by which the user can interact with the system. These help in achieving emergency manual mode, and cutting off complete power supply to the wheels in case of major malfunctioning of the autonomous software.

5.7 Environment

The product will be run on flat white roads with a black strip representing the path to be followed. The path thickness needs lie between 1.5cm and 1.8 cm. The path has 0deg of inclination, and will be present indoors within a lighting of 10 lx to 320 lx. The temperature will be around 28 degC

5.8 Physical conditions

Threshold operating conditions for the whole system, taking into consideration all the components being used in it:

1. Temperature: 0 degC to 40 degC
2. Voltage and Current Mains: 230V, 15A
3. Voltage and Current Car: 5V/12V, 1.2A

5.10 Biohazard level

Biohazard, specifically to other human beings on the road could be cause by reasons mentioned below:

1. Incapability to adapt to changes in road and whether conditions.
2. Inability to judge pedestrian behaviour and signals.
3. Drivers being inexperienced if situations arose requiring manual driving

Thus, to operate the system, it is essential to meet the requirements as mentioned under section 5.12 of this document

5.11 Cleaning Requirements

1. Cleaning Requirements by user:
   a. Camera's front glass. Use glass-wipe type of good quality wipes to prevent scratches on camera panel. Cleaner camera facilitates lesser errors.
   b. Headlights: To be wiped/washed when the system is turned off. Having clean headlights allows path to be better lit, and thus helps lessens errors.
2. Cleaning requirements by the developer:
   a. Cleaning and regreasing/oiling of the wheel's axis
   b. Regreasing/oiling internal gears and motors joints
   c. Testing continuity of the wires, and resoldering if required

5.12 Intended Operating Environment

For optimal results, the product has been designed to be used of flat white roads with a black strip representing the path to be followed. The path thickness needs lie between 1.5cm and 1.8 cm. The path is allowed to have a maximum inclination of 5 degree, and there cannot be a change its inclination at any point. The lighting conditions need to be in the range of 10 lx to 640 lx. No obstructions or obstacles are to be allowed in the area of view of the camera.(Area of view has been defined in the Release Manual under section 4.1). The product has been tested for use only indoors, and may not fare well in outdoor gust, rain, or snow.

6 Requirements for Installation

6.1 Calibration

There are 3 main calibrations that can be made for the car;

1. Machine Learning Image Database:
   The car has been trained using supervised learning on a set of about 400 images. The images taken have covered situations that the car may face on a day to day basis such as straight path, right/left turns, and U-turns. Logistic Regression

has been used to train the system. These images have helped achieve accuracy greater than 80%.

Image Samples:

There can be cases where the car may not behave as expected or manuver incorrectly. In such cases, to help improve the system, the user can report to us the situation, (road and location) where he faced the problem. The service team would then go to that road and figure out what it was that went wrong. If it was a new situation the car had faced, for which it had not been trained, the service and calibration team would manually drive the car into the same situation, and update the training image data set. This would enable the car to navigate correctly the next time it encountered similar situations.

2. OpenCV HSV Calibrations:
   The system uses black and white images as described above. These images are generated based on particular range of HSV values. These values have been set for the case where lighting varies between 10lx to 640lx with the headlights on at all times. Beyond 640 lx, there will be considerable reflection from the sheet causing the algorithm to detect even the black portion as white. Depending on the weather conditions, season and manner and place of use of the car, a different HSV range may be required to optimise performance. This can be adjusted as per the user's needs.

3. Wheel Rotation Speed and Frequency of Photo Capture:
   The car has been designed to run such that its motors are driven for a total of 0.3 seconds for each image that it captures. Corresponding to this is the speed of the car. This allows a speed of about 2cm/s to 3cm/s.
   If the user desires the car drive slower or faster, the time for which the motors are driven per photo and the frequency of photo capture can be altered.

6.2 Deliverables

Deliverables identified in this document will be identified in advance and at the time of installation. They include:

1. The car key (Power switches in our case)
2. The installed software on the PC and ULK board
3. The camera
4. The USB-UART cable
5. The path on which the Car can be run

6.3 Documentation

Appropriate user manuals and schematics will be provided as in the "User Manual – Driverless Car" document

6.4 Training

Appropriate training along with documentation will be provided on the date the product is taken for a test drive, and/or bought by the customer.

6.5 Tools and Spares

Tools required:
1. Soldering gun
2. Screw driver with a star bit
3. Wire stripper
4. ULK, OpenCV, Libusb Software on Ubuntu

Spares part include:
1. L293 h-bridges
2. DCMHBY DC motor
3. Led headlights
4. Camera
5. Additional wires

6.5 Calibration/Servicing

There are 4 main service and calibrate the car. The different ways are mentioned below;

1. Machine Learning Image Database:
   The car has been trained using supervised learning on a set of about 400 images. The images taken have covered situations that the car may face on a day to day basis such as straight path, right/left turns, and U-turns. Logistic Regression has been used to train the system. These images have helped achieve accuracy greater than 80%.
   There can be cases where the car may not behave as expected or manuver incorrectly. In such cases, to help improve the system, the user can report to us the situation, (road and location) where he faced the problem. The service team would then go to that road and figure out what it was that went wrong. If it was a new situation the car had faced, for which it had not been trained, the service and calibration team would manually drive the car into the same situation, and update the training image data set. This would enable the car to navigate correctly the next time it encountered similar situations.

2. OpenCV HSV Calibrations:
   The system uses black and white images as described above. These images are generated based on particular range of HSV values. These values have been set for the case where lighting varies between 10 lx to 640 lx with the headlights on at all times. Depending on the weather conditions, season and manner and place of use of the car, a different HSV range may be required to optimise performance. This can be adjusted as per the user’s needs.

3. Wheel Rotation Speed and Frequency of Photo Capture:
   The car has been designed to run such that its motors are driven for a total of 0.3 seconds for each image that it captures. Corresponding to this is the speed of the car. This allows a speed of about 2cm/s to 3cm/s.

If the user desires the car drive slower or faster, the time for which the motors are driven per photo and the frequency of photo capture can be altered.

4. Wheel Alignment:
   This is essential to ensure proper turning of the car. The screws attaching the wheels to their axis are to be tightened, and the gears allowing the front wheels to turn need to be cleaned and oiled. Additionally, the restoring spring present at the front wheels may need to be replaced. This spring is essential to restore the wheels to their 0deg position, and thus allow it to drive straight.

CONSTRUCTION FEATURES:

1. Car
   a. Dimensions (lxbxh): 24.5cms x 14.5 cm x 11 cm
   b. Min Radius of Turning: 56cm or greater
   c. Motors: DCMHBY, 24V 1000 rpm motors
   d. H-bridges: L293D, 4.5V-36V output, peak output current: 1.2A
   e. Lights: CE ROHS, operating voltage:12V

2. Camera
   a. Model: Microsoft Lifecam hd 6000
   b. Height and Inclination of placement: 12cm, 45deg to the normal (inclined forward)
   c. Field of View: 66deg diagonal view
   d. Area of View: blind spot for 6cm from front of car, extends to 51cm from front end of car, extends to 15cm on either side. Total area of view = 675 sq. cm
   e. Resolution 1280x800 sq. Pixels

3. Path
   a. Strip thickness: 1.5 cm to 1.8 cm
   b. Colour: Black strip on White background
   c. Allowed Radius: 56cm or greater
   d. Allowed inclination of path: 5 degrees
   e. Clearance area: Same as Area of view of camera = 675 sq. Cm

4. Power Supply
   a. Mains: 230 volt 50/60 Hz to adapter of the ULK board, PC, and Voltage regulator
   b. Car: 12V to h-bridge of front wheels and 5V to the h-bridges of the rear wheels.

5. Additional Construction Features:
   a. Orientation: For the car to be able to drive automatically, it needs to be placed/driven manually such that the path is in the Field of View of the camera
   b. Lighting Conditions: 10 lx to 640 lx

CONTROL FEATURES:

1. Automatic Mode (‘a’)

This mode is activated when the user presses the key ‘a’ on the keyboard. In this mode, the car is in full control, and the user cannot drive it. The car uses its camera and trained machine learning code, to predict the direction, and take the respective turns.

In the automated mode, the options available to the user are the other 3 modes of operation of the car, as mentioned in section 5.2 to 5.4. At any given point in time, is the user feels he wants to switch to another mode, he may do so by pressing the corresponding key. While in this mode, the terminal outputs the status ‘automatic’, along with the upcount.

2. Manual Mode (‘m’)

This mode is activated when the user presses the key ‘m’ on the keyboard. It allows the user to control the car manually, without any interference from the system’s AI. The car behaves like a non-self-driving car in this mode.

In the manual mode, the options available to the user are the other 3 modes of operation of the car, as mentioned in section 5.1, 5.3, and 5.4. At any given point in time, is the user feels he wants to switch to another mode, he may do so by pressing the corresponding key.
While in this mode, the terminal outputs the status ‘manual’, along with the upcount

3. Testing/Semi-automatic Mode (‘t’)

This mode is activated when the user presses the key ‘t’ on the keyboard. In this mode, the car’s self-driving ability kicks in for a short duration of time (about half a second). This mode thus drives the car automatically for half a second each time the key ‘t’ is pressed. The reason this mode has been provided is to help in testing the behaviour and decision making of the car, step-by-step, when training it, or during debugging when it is faced with the different or new paths. This mode can be used by the customer to get a feel for the autonomous driving capability of the car in short bursts, if they are not comfortable being in the full automatic mode. This would help in cases where customers may not fully trust the car’s behaviour, and would want time to get used to its working. (eg, while test driving an autonomous car for the first time).

In the manual mode, the options available to the user are the other 3 modes of operation of the car, as mentioned in section 5.1, 5.2, and 5.4. At any given point in time, is the user feels he wants to switch to another mode, he may do so by pressing the corresponding key.
While in this mode, the terminal outputs the status ‘testing’, whenever the ‘t’ key is pressed.

4. Exit

This is the method by which the customer can exit the system. By pressing the ‘esc’ key or the ctrl+c key combination, the customer can power down the car, at the end of their use, or in cases of emergency.

GENERAL:

- ENERGY CONSUMPTION:- Electricity at 5V, 9V, and 12V
- EASY CLEAN SURFACES (EXTERNAL AND INTERNAL):- External: Wheels, Car body, Headlights   Internal:- None are easy to clean
- HIGHLY VISIBLE DISPLAYS:- ULK LCD Panel for direction display
- CONNECTIVITY TO EXTERNAL SITE ALARM SYSTEMS - None
- ALARM :- Visible only, ULK LCD Panel and Terminal