*

*

*

ADAMA SCIENCE AND TECHINOLOGY UNIVERSTY
School of Electrical Engineering And Computing
Electronics and Communication Engineering Program

Title: Design And Implementation Of Car Driving Simulator for Driving Trainee(Phase II)

A Senior project submitted for the requirement of Bachelor of Science Degree in Electronics and Communication Engineering

Prepared by:

1) Yonas Taye.....R/01041/07
2) Yidnekachwe Mekuria.....R/01022/07
3) Wogderes Abebe.....R/00992/07
4) Abraham Tadesse.....R/00042/07
5) Meharu Abera.....R/00668/07

Advisor Name: Mr. Kedir Beshir
Submission Date: July 22,2019

Declaration

We declare that this project is the original work of us which has not been presented for a degree program in this or any other universities, and all sources of materials used for the project have been fully acknowledged.

Students Name	Signature
Yonas Taye	.....
Yidnekachwe Mekuria	.....
Wogderes Abebe	.....
Abraham Tadesse	.....
Meharu Abera	.....

This Project has been submitted for examination with my approval as an advisor of the project titled as **Design And Implementation Of Car Driving Simulator for Driving Trainee**.

Project Advisor	Signature
Mr. Kedir Beshir	_____

Acknowledgment

First and for most, we are extremely grateful to the program of Electronics and Communication Engineering, for giving us the opportunity to carry out this project, which is an integral part of the curriculum in Adama science and Technology university. Also, we would like to extend our gratefulness to one and all who are directly or indirectly involved in the successful completion of the project.

Abstract

This paper presents the design and implementation of car driving simulator for training purpose. Simulator is a device which places an individual in a situation which resembles real world, our driving simulator is a virtual environment which is used to teach and evaluate the driver similar to the actual car. Our main goal of this project is to create a realistic driving simulator for the driving trainee school and to contribute something to reduce the risk and difficulty faced by the new drivers in the learning process of driving.

In this project we have designed and developed a driving simulator by using a hardware and software devices. The simulator includes a steering wheel,shifter, accelerator and brake pedals that has thee same function. Our project focus on designing a car simulator that help trainee with previous practical training with virtual world. This enables learning the basic of car controls like accelerating, braking and changing driving direction of a vehicle with the help of simulator before they begin the practical driving training.

Key words: driving simulator, school trainee, static base, driving in a risk-free environment, virtual reality.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

3D	Three-dimensional
ADC	Analog-to-digital converter
ASCII	American Standard Code for Information Interchange
CAMP	Crash Avoidance Metrics Partnership
GND	Ground
IDE	Integrated development environment
LED	light-emitting diode
NADS	National Advanced Driven Simulator
PWM	Pulse Width Modulation
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

Chapter One

1 Introduction

Car simulator is a device which places an individual in a situation which resembles real driving. The individual controls the commands which are the same as those of a real car. With the development of computer-generated images, it has become possible to recreate a complete virtual environment (including roads, indications, vehicles, etc.) in an entirely interactive way. The computer recreates a virtual landscape with vehicles, and the learner driver can move in this universe with complete freedom.

The simulator provides realistic artificial environment and situations in which allows practice driving in a risk-free environment and with possibility to repeat exercises without restrictions[1]. Now a days, western countries use driving simulators for training driver's education courses taught in educational institutions as well as in entertainment and private businesses. They are also used for research purposes in the area of human factors and medical research, to monitor driver behavior, performance, and attention and in the car industry to design and evaluate new vehicles or new advanced driver assistance systems.

Driving simulators are being increasingly used for training drivers all over the world. Research has shown that driving simulators are proven to be excellent practical and effective educational tools to impart safe driving training techniques for all drivers[2]. it provides an opportunity to educate staff in the driving skills that achieve reduced maintenance costs, improved productivity and, most importantly, to ensure the safety of their actions in all possible situations.

Research proves that the use of driving simulators in novice driver training may greatly enhance safe driving. A large scale study, performed in the United States(R. Wade Allen et al. 2007), has shown that when a good simulator is used, the accident rate during the two year period after getting the license is reduced to 34 percent of the national average accident rate for novice drivers. Since teen drivers accident rates are about five times higher than those for mature drivers, the application of simulators in teenager driver training should be seriously considered. A driving simulator indisputably improves driver training, resulting in safer drivers and a reduction in the accident rate of up to 66 percent for novice drivers[3].

1.1 background

Ethiopia is one of the countries that ranked highest in the number of road traffic accidents. There are many factors that contributed to this statistic; one of them is the deficiency in pre-driving skill training in the overall driver education and training curriculum. According to the country's transport authority, the number of people killed as well as the rate of accidents occurring per year increased in a considerable way. About 5,118 people died in traffic accidents in Ethiopia during the 2017/18 fiscal year[4]. The number of people killed increased from 4,500 to 5,118 while a record number of close to 41,000 traffic accidents were registered during the 2017/18 fiscal year. Another 7,754 people had also sustained serious physical injuries from traffic accidents across the country, while some 7,775 others had light injuries, according to the report of the Ethiopian Federal Transport Authority. The authority also indicated that Ethiopia's rate of traffic incidents during the just concluded fiscal year has registered a 6-percent increase as compared with the previous year. Even Though Ethiopia has one of the lowest per capita car ownerships in the world, deadly traffic accidents are fairly common due to bad roads and flawed driving license issuance system[4].

It is quite easy to argue that if we are able to produce knowledgeable and competent driver, our road will be safer and the number of traffic accident can be reduced.

Previous training enables learning the basic road traffic rules, shifting gears, accelerating, braking and changing driving direction of a vehicle (driving the vehicle). The training system and periodic testing of drivers does not allow any possibilities for testing human behaviors in emergencies that pose a threat of an accident it is impossible to acquire higher level driving skills in high-risk situations. Being a driving trainee, one cannot face more complicated and riskier situation while training[13].

1.2 Statement of problem

From the above declaration's poor-quality education in the driving schools is considered to be the factor behind the ever-increasing fatal traffic accidents in and around the outskirts of Ethiopia. The problem rising with driver license quality in Ethiopia is related to limited infrastructure, limited practice area, limited practice time. the process of licensing drivers based on existing curriculum mainly focus on developing theoretical knowledge and leaving some hour for practical training. Due to the number of driver trainee and limited infrastructure school trainee can't get more practical time. Not only that this practical training schedule is also

used for familiarization. one driving trainee may not get the chance to drive long distances and practice with some circumstances. With this problem a trainee will not get a perfect driving skill and control while in school. Also, currently there is limited on-road training areas. This will also limit trainee from experiencing the real driving environment. Our project focus on implementing previous practical training with virtual world enables learning the basic of car controls like shifting gears, accelerating, braking and changing driving direction of a vehicle (driving the vehicle) with the help of simulator before they begin the practical driving training.

1.3 Objective

1.3.1 General objective

The general objective of this project is to design and implement driving simulator for driving trainee that is used to increase skill of driver.

1.3.2 Specific objective

In order to achieve the general objective, the following specific objectives are needed.

- To study existing system
- To design electrical control and communication system
- To design the system
- To simulation of hardware and software
- To select appropriate implementation software and hardware

1.4 Significance of the study

The driving simulation has two main advantages:

- Develop pre-driving skill - Because relevant driver skills can be consistently practiced much better than in a real car.
- Any environment can be created in the simulator and reproduced over and again, in the exact same way.
- Develop self-confidence- After the simulator training phase, the trainee can start driving in a learner car with much more self-confidence, because the most important driving skills have already been learned.

- The use of a driving simulator increases capacity of the driving school
- Training in a safe environment without stress. Stress inhibits the efficient development of skills.

1.5 Limitation

- The game is currently working in Windows operating system
- The Virtual environment we created don't resemble driving space in our country
- Virtual Environment we created lacks traffic light system

1.6 Data collection methodology

1.6.1 Primary data source

Observation

- we have observed some car driving training schools in Adama.

1.6.2 Secondary data source

Document analysis

- we saw some training document used in training schools, also related car driving simulator implemented in different areas.

1.7 Scope of our project

The scope of our project is:

- Design virtual environment for the simulator
- Configuration of gas, brake, steering, hand brake with electrical elements
- Basic car control script used for accelerating, braking and steering
- Represent car controlling function with respect to voltage and current
- Simulation of virtual car controlled with corresponding electrical element

1.8 Organization of document

This project is organized in to five chapters. The first chapter provides with an overview introduction of car driving simulator. It is followed with related literature reviews in which previous studies and reports related to the car driving simulator specifically for driving training schools are reviewed. The third chapter illustrates the methodology which describe System block diagram including its brief description of the system design. The fourth chapter presents System design and implementation which illustrates mechanical, electrical and software desgin and implementation.
The Last chapter gives conclusion and possible recommendations based on the results of the project.

Chapter Two

2 Literature View

Simulators have existed since the 1960s, for a long time they remained too expensive to be used widely for training purposes. The fast development in technologies such as computers has become ever more powerful, and their price-performance ratio has been getting increasingly better. There are a large number of driving simulators throughout the world today, and they are rapidly evolving. The older ones have constantly been improved in an attempt to reproduce real-world driving conditions as accurately as possible. In Netherland for example in 2010, it is estimated around 100 simulators were being used in basic driving training[5].

2.1 NADS at the University of Iowa (USA)

The National Advanced Driving Simulator (NADS) is the most sophisticated research driving simulator in the world. Developed by the NHTSA, and located at the University of Iowa's Oakdale Research Park, the NADS offers high-fidelity, real-time driving simulation. It consists of a large dome in which entire cars and the cabs of trucks and buses can be mounted. The vehicle cabs are equipped electronically and mechanically using instrumentation specific to their make and model. At the same time, the motion system, on which the dome is mounted, will provide 400 squaremeters of horizontal and longitudinal travel and nearly 360 degrees of rotation in either direction. The effect will be that the driver will feel acceleration, braking and steering cues as if he or she were actually driving a real car, truck or bus. Fig shows the configuration of the NADS.

Figure 1: NADS configuration

As a national leading simulator, the NADS was charged with many kinds of research projects and contributed a lot of valuable publications. They are focused on five research areas: a) driver distraction relating to wireless voice communication devices; b) driver behavior including young driver risk, driver

validation and driver reaction to thread separation scenarios; c) drugs and driver impairment including vision validation test and pharmaceutical project; d) advanced vehicle system including electronic stability control, evaluating lane change collisions, safety vehicles using adaptive interface technology, agricultural virtual proving ground, and Crash Avoidance Metrics Partnership (CAMP) advanced simulation including visual database development, advanced simulator research consulting, and software support.

2.2 Simulator III at VTI (Sweden)

Simulator III at the VTI was introduced in April 2004, after several years of intensive development work. Simulator III is built around a real vehicle chassis and a sophisticated motion system, which enables fast acceleration. The surroundings are simulated and displayed to the driver via three main screens and three rear view mirrors. Under the chassis is a vibration table to simulate contact with the road surface, providing a more realistic driving experience. The driving dynamics are also very advanced and on the forefront of what can be done with current technology. Together this creates a unique simulator that provides an extremely realistic experience. Because of the modular construction, the various subsystems can be adapted to suit the needs of each individual research project. The driving simulator can be fully adapted for private cars and trucks by means of a chassis interchange system. In the future, it will also be possible to use it to simulate rail traffic.

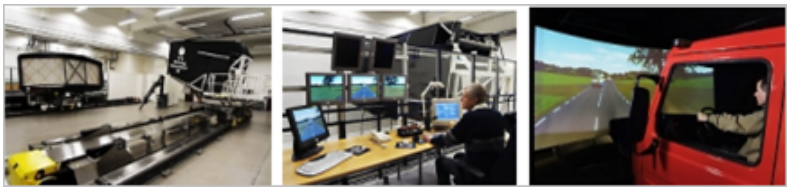

Figure 2: Simulator

The application of Simulator III was broad from studies concerning driver behavior, the human-machine interface and the effects of tiredness and drugs to projects concerning environmental issues, road and landscape design, tunnel design, the reactions of the body, drivers with reduced functionality and new subsystems in vehicles. The effects of noise and vibrations on driver performance are examples of other areas that may be studied using the simulator. One exciting area is how the new technology influences driving, for example the use of mobile telephones. VTI has performed this type of test in the simulator, which would have been impossible in a real traffic environment for safety reasons. Recently, the

simulator has also been used for the planning of the Northern and Southern Link Roads in Stockholm, for example, in order to determine the positioning of road signs and for reasons of aesthetic design.

A company called SimDrive found in South Africa is one of the companies that implement car driving simulator for driving trainee. SimDrive specializes in simulator driver education. The company has already started training young learners at seven schools in Gauteng. They have a formal standardized syllabus that shows the progress made during training as well as an end result. SimDrive also facilitates night driving, which no other traditional teaching method addresses. Simulator driver training has been proven to produce better and safer drivers. We have the means and technology to reduce the carnage on our roads, roads that you and your family drive on every day[3].

Also, Malaysia is one of the countries that ranked highest in the number of road traffic accidents. There are many factors that contributed to this statistic; one of them is the deficiency in pre driving skill training in the overall driver education and training curriculum. This is also true for Ethiopia. Hence adapting driving simulators overall driving education and training will be one of many ways to reduce road traffic accidents. In this project we tried to adopt driving simulator technology based on our country context in cost effective way[12].

Chapter Three

3 Methodology

Our main task for creating a car driving simulators is to create and place the driver in an artificial environment believed to be a valid substitute of the actual driving environment. This is done by using 3D gaming software like unity 3D, Unreal. Typical training environment found in real world is modeled and constructed in virtual world. This artificial environment is used to train the driver with the corresponding lesson. In addition to the environment we also model a real-world car to virtual car found virtual world. The main part in a driving simulator is the real-time vehicle dynamic simulation and the interaction of driver with the car. With the above process we can have artificial environment as well as virtual car. A driver can manipulate its artificial car in artificial environment basic functionally of car like accelerating, braking, steering, horn. This simulation effects are expected to be similar to the real-world car driving scenario and user can visualize its action effect through computer screen.

The interaction of driver with virtual car is guided with some electrical sensors.These sensors are used to transducer physical change made by driver to corresponding electrical signal. The driver's physical interaction with the system through the steering wheel, gas and brake pedal and various car controller is captured by sensors. This sensor data is used to control virtual car with the corresponding action.

After completing the task in virtual environment our control circuit is designed and simulated. Proteus ISIS 8.6 professional is used to design the circuit and simulation. One major task in our project was to find a way for communication between virtual environment with our circuit.

3.1 Block Diagram

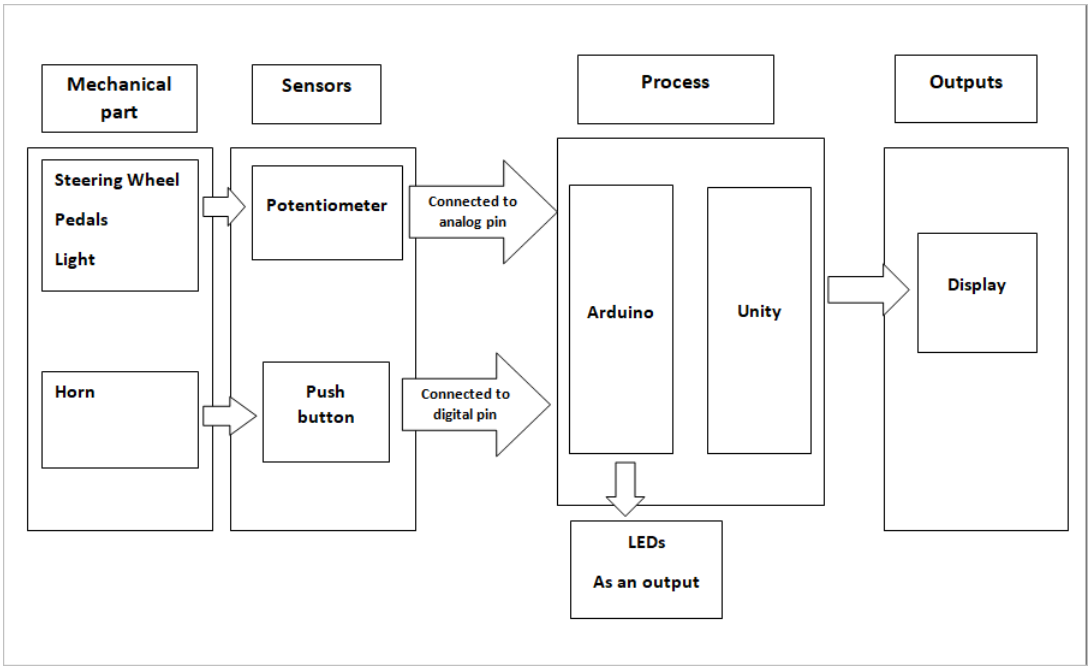

Figure 3: Block diagram of our system

3.2 Description of block diagram

3.2.1 Mechanical part

Steering wheel

A steering wheel (also called a driving wheel or a hand wheel) is a type of steering control in vehicles.Steering wheels are used in most modern land vehicles, including all mass-production automobiles, as well as buses, light and heavy trucks, and tractors. The steering wheel is the part of the steering system that is manipulated by the driver; the rest of the steering system responds to such driver inputs. This can be through direct mechanical contact as in recirculating ball.in our case we have designed the steering wheel by using some basic material such as bearing, metal stick(shaft),potentiometer, gear and wood.

Bearing: It is used to reduce friction, generally on a rotating shaft. it is made up of a material which helps in holding oil on the surface to reduce friction. it has balls or rods in between two metals to make a sliding contact to rolling surface.

Bearings are used basically for performing three important tasks as mentioned below

1. Reducing Friction
2. Supporting the load
3. Providing the guide for moving components such as shafts or wheels

Figure 4: Ball Bearing

Shaft: It is an element which supports rotating parts(steering wheel)and it turn themselves supported by bearings resting in the rigid body. The shafts perform the function of transmitting power from one rotating member to another supported by it or connected to it .The length of the shaft is 40 cm and the perimeter of the steering wheel is 92 cm.

Shifter

Conventionally, in order to select the transmission operating mode, the driver moves a selection lever to different states, which are Park (P), drive (D), neutral (N) and reverse(R).

Park(P)

This selection locks the speed of the car in to limited range which is suitable for parking.

Reverse(R)

This engages reverse gear within the transmission, permitting the vehicle to be driven backward. The driver should avoid engaging reverse while the vehicle is moving forwards, and likewise avoid engaging any forward gear while traveling backwards.

Neutral/No gear

This disengages all gear trains within the transmission,effectively disconnecting the transmission from the driven wheels except the steering wheel angle.

Drive

This position allows the transmission to engage the full range of available forward gear ratios, allowing the vehicle to move forward and accelerate through its range of gears.

Pedal design

Since the simulator is automatic we have designed two pedals parts these are the gas (accelerator) and brake pedal.to design this two pedal we use some material such as spring, wood, potentiometer, connecting wire and bolt

Gas pedal

When we hit the gas pedal , there are a number of forces at play to get it going. Here is a basic run-down of what happens when the car accelerates. when some forces apply to this pedal the spring change from its initial position or the force extend or compress a spring by some distance x scales linearly with respect to that distance. That is: F=kx where k is a constant factor characteristic of the spring: its stiffness, and x is small compared to the total possible deformation of the spring. Since the spring expanded to some distance the potentiometer value change from the previous value and the speed of the car increase

Brake pedal

The brake pedal is the pedal that you press with your foot in order to make a vehicle go slower or stop. When the driver puts his foot on the brake pedal, the system automatically applies the optimum pressure required to avoid hitting the car in front.

3.2.2 Sensors

User module mainly consists of sensors used to sense driver interaction with the system. They can be classified as Analog sensor and Digital sensr depending on, they are subjected to sense.

Analog sensor

Analogue Sensors produce a continuous output signal or voltage which is generally proportional to the quantity being measured.

Potentiometer

A potentiometer is a manually adjustable variable resistor with 3 terminals. Two terminals are connected to both ends of a resistive element, and the third terminal connects to a sliding contact, called a wiper, moving over the resistive element. The position of the wiper determines the output voltage of the potentiometer. The potentiometer essentially functions as a variable voltage divider. The resistive element can be seen as two resistors in series(potentiometer resistance), where the wiper position determines the resistance ratio of the first resistor to the second resistor.

A potentiometer is also commonly known as a potmeter or pot. The most common form of potmeter is the single turn rotary potmeter. This type of pot is often used in audio volume control (logarithmic taper) as well as many other applications. Different materials are used to construct potentiometers, including carbon composition, cermet, wire wound, conductive plastic or metal film.

Figure 5: Potentiometer

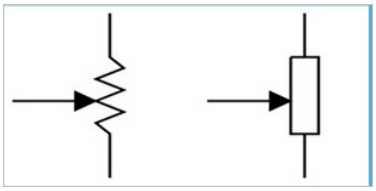

Figure 6: Schematic Symbols of a Potentiometer

Working Principle

The potentiometer can be used as a voltage divider to obtain a manually adjustable output voltage at the slider (wiper) from a fixed input voltage applied across the two ends of the potentiometer.

The potentiometer can be used as a voltage divider to obtain a manually adjustable output voltage at the slider (wiper) from a fixed input voltage applied across the two ends of the potentiometer.

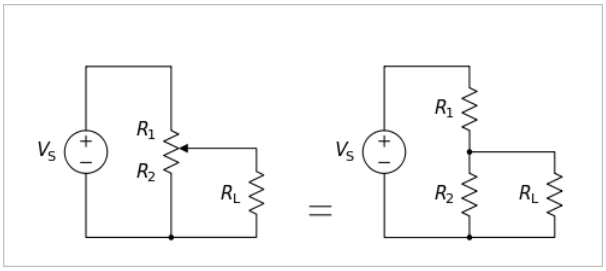

Figure 7: Working principle of potentiometer

$$\mathbf{V}L = (R2 * Vs)/(R1 + R2) \tag{1}$$

Digital Sensors

Push Button

A push button is a simple type of switch that controls an action in a machine or some type of process. It is a momentary or non-latching switch which causes a temporary change in the state of an electrical circuit only while the switch is physically actuated. An automatic mechanism (i.e. a spring) returns the switch to its default position immediately afterwards, restoring the initial circuit condition. A push button switch is a small, sealed mechanism that completes an electric circuit when you press on it.

Figure 8: Push button

Working principle

In order to make push button as digital sensor it should be connected install either to Ground or +5V. in order to implement this we use pull-up resistor or pull-down resistor. This will make our microcontroller initially at determined value.

3.2.3 Process

Micro controller

One of the most available microcontrollers in our compound is Arduino UNO. Arduino is an open source hardware/software programming platform based around Atmel microcontrollers. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. The Arduino platform has become quite popular with people just starting out with electronics, with ATMEGA 328 board. Unlike most previous programmable circuit boards, the Arduino UNO does not need a separate piece of hardware (called a programmer) in order to load new code onto the board, It simply use a USB cable[6]. Additionally, the Arduino IDE uses a simplified version of C languages, making it easier to learn to program.

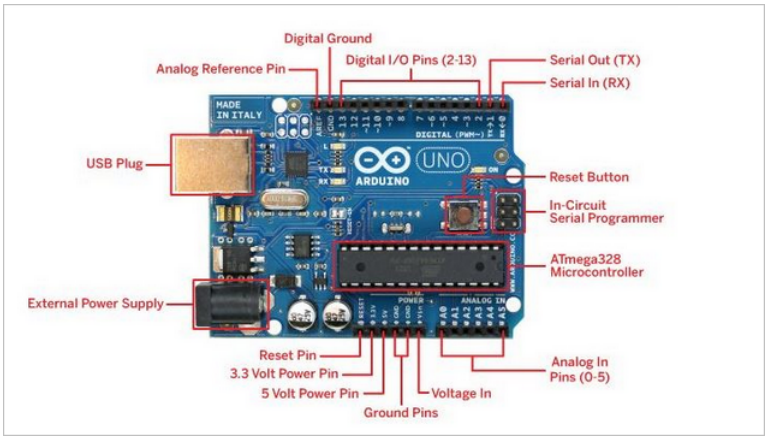

Figure 9: Arduino UNO

Power (USB) and pin configuration and description

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply. It's not allowed to use a power supply greater than 20 Volts as you will overpower and thereby destroy the Arduino UNO. The recommended voltage for

most Arduino models is between 6 and 12 Volts. The pins on your Arduino are the places where it will connect wires to construct a circuit probably in conjunction with a breadboard and some wire. They usually have black plastic headers that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions[7].

GND(3): Short for Ground. There are several GND pins on the Arduino, any of which can be used to ground the circuit.
5V(4) AND 3.3V(5): As we might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
Analog(6): The area of pins under the "Analog In" label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
Digital(7): Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
PWM(8) the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but it can also be used for something called Pulse-Width Modulation (PWM).
AREF(Stands for Analog Reference)(9): Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Detail Pin Description

Table 1: Arduino UNO pin configuration

Pin Category	**Pin Name**	**Details**
Power	Vin,3.3V,5V,GND	Vin:Input voltage to Arduino when using an external power source 5V:Regulated power supply used to power microcontroller and other components on the board 3.3V:3.3V supply generated by on-board voltage regulator. Maximum current draws 50mA. GND:ground pins
Reset	Reset	Resets the microcontroller
Analog Pins	A0-A5	Used to provide analog input in the range of 0-5V
Input/Output Pin	Digital Pins 0-13	Can be used as input or output pins.
Serial	0(Rx),1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2,3	To trigger an interrupt
PWM	3,5,6,9,11	Provides 8-bit PWM output
SPI	10(SS),11(MOSI),12 (MISO) and 13(SCK)	Used for SPI communication
In built LED	13	To turn on the in built LED
TWI	A4(SDA),A5(SCA)	Used for TWI communication
AREF	AREF	To provide reference voltage for input voltage

Arduino UNO Technical Specifications

Table 2: Arduino UNO technical specification

Microcontroller	**ATmega- 8-bit AVR family microcontroller**
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6(A0-A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

3.3 Unity

Unity3D is a powerful cross-platform 3D engine and a user-friendly development environment. Unity engine can be used to create both three-dimensional and two-dimensional games as well as simulations for its many platforms. Unity gives users the ability to create games and interactive experiences in both 2D and 3D, and the engine offers a primary scripting in C, JavaScript for both the Unity editor in the form of plugins, and games themselves, as well as drag and drop functionality[8]. There are five main views used in the Unity editor to get all the work done, the project view, scene view, game view, hierarchy view and inspector view, all of which are explained in more detail below.

The Unity game engine works in conjunction with MonoDevelop for controlling the behavior of objects. MonoDevelop is an open source Integrated Development Environment or IDE developed by Xamarin and the Mono community, which is primarily used for development in the C programming language. With MonoDevelop, developers are able to write scripts in C, JavaScript, or Boo and attach them to objects in the Unity engine; these scripts enable developers to control the logic and behaviors of objects within the Unity environment [13]. For example, if a developer was implementing the braking system on the wheels of the player-controlled car, she/he would write and attach a script that takes keyboard input from the user and applies the input value to the wheel's velocity. In this way, the combined tools of Unity and MonoDevelop enable developers to focus on the development of the

AI components of the game rather than on issues related to 3D graphics rendering and physics calculations. Figure 4 shows the script inside one of the icons shown on the bottom panel of the Unity editor; development of the script was supported by the MonoDevelop IDE.

3.3.1 Terms used in unity

Game Object

Gameobject is the fundamental object in Unity that represent characters and it is a base class for all entities in Unity Scenes.

- In our game, the game object is car

Terrain

Unity's Terrain system allows you to add vast landscapes to your game.

- In our game, our Terrain is driver training environment.

Rigid bodies

Rigid bodies enable Game objects to act under the control of physics. The Rigid body can receive forces and torque to make your objects move in a realistic way. Any Gameobject must contain a Rigidbody to be influenced by gravity, act under added forces via scripting.

- In our case car and road are some examples of Rigidbodies.

Vector3

Representation of 3D vectors and points. This structure is used throughout Unity to pass 3D positions and directions around. It also contains functions for doing common vector operations.

Wheel Collider

A special collider for vehicle wheels. Wheel collider is used to model vehicle wheels. It simulates a spring and damper suspension setup, and uses a slip-based tire friction model to calculate wheel contact forces.

- Practically in cars a wheel is colliding with the ground holding the eight of the car, similarly we have wheel collider that support the weight of the car and make contact with the ground.

Transfrom

Position, rotation and scale of an object. Every object in a Scene has a Transform. It's used to store and manipulate the position, rotation and scale of the object. Every Transform can have a parent, which allows you to apply position, rotation and scale hierarchically. This is the hierarchy seen in the Hierarchy pane.

- In our game we have transformation. Considering a moving car, it has translation motion in its movement direction also the wheel perform rotational motion.

Camera

A Unity scene is created by arranging and moving objects in a three-dimensional space. Since the viewer's screen is two-dimensional, there needs to be a way to capture a view and "flatten" it for display. This is accomplished using Cameras.

3.3.2 Views

The Scene View

The scene view is one of the most used views as this is where all the game objects are Placed and scenes for the game are built.

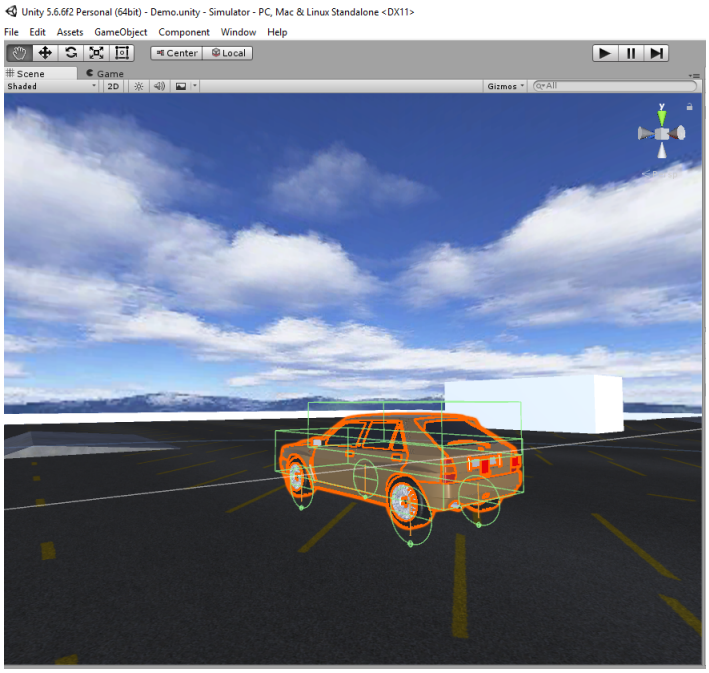

Figure 10: Scene view

The Game View

The game view is what user will see when the game is started. There are several options for this window. Across the top of the window there are several button/drop down menus which can change things from the perspective, full screen, and gizmos shown in the game view.

Figure 11: Game View

The Hierarchy View

The hierarchy view is where all the objects in the game can be created, accessed, grouped and manipulated to make the game. When the project is saved, the objects are saved in a scene file.

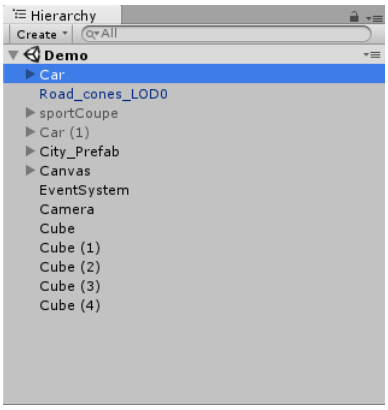

Figure 12: Hierarchy View

The Project View

The project view is where all the scripts and scenes are accessible from. This view is exactly like the file explorer on Windows or Mac and allows creating files and folders to help organize the projects assets.

Figure 13: Project View

The Inspector View

The inspector view is where all the physics and properties of the objects are stored and accessed from. Every game object has a transform; this is what holds properties of the object such as rotation, position and scale. Other properties are the physics affecting the object, textures to load on the object and sound.

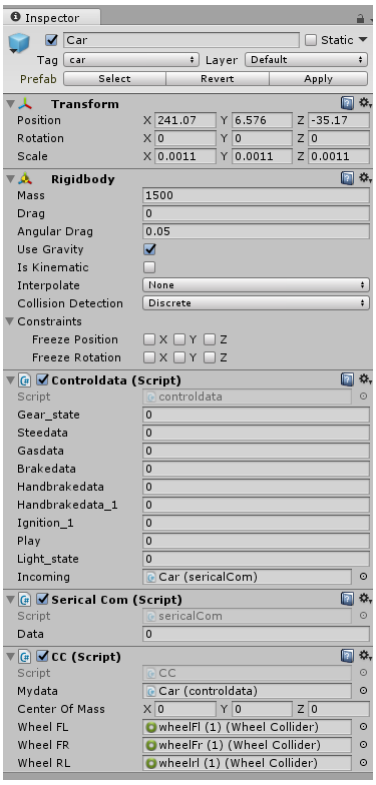

Figure 14: Inspector View

Chapter Four

4 System design and Implementation

4.1 Mechanical design and implementation

4.1.1 Steeering wheel

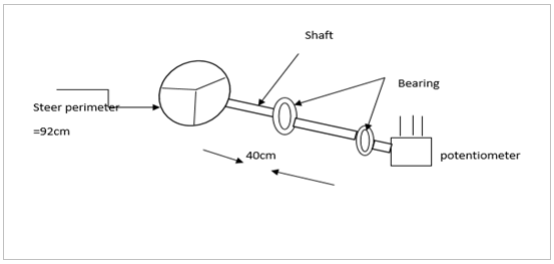

Figure 15: Steering wheel

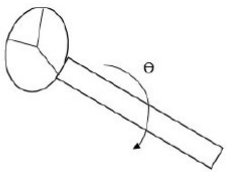

Figure 16: Potentiometer change with respect to steering wheel

$$\textbf{Voltage}(V) = (5/270) * (\triangle\theta)v \tag{2}$$

The above figure shows that the design of steering wheel when the steer rotate either clockwise direction or counter clockwise the potentiometer range value read from (0-1023). We have used different material to design and implement such as bearing , potentiometer, and some wasted wood material.

4.1.2 Shifter

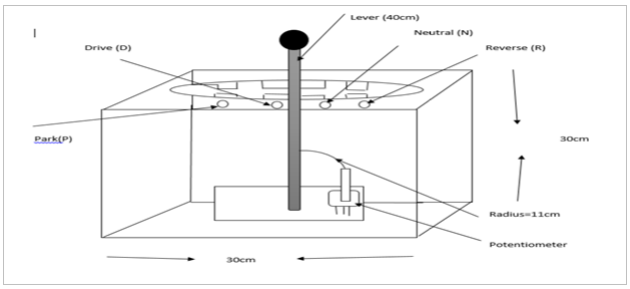

Figure 17: Shifter

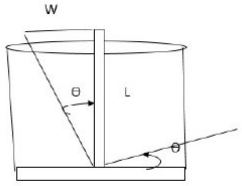

Figure 18: The change of potentiometer with respect to shifter

The transmission (shifter) designed 30cm by 30cm wood material and 40 cm lever length which is used to convert the condition of the car.

The lever changes between the four positions(Drive, park, reverse and neutral) based its location LEDs connected to each of the position LEDs will be on and the value of the potentiometer changes between the ranges of (200-300).

4.1.3 Pedal

To design this two pedal we use some material such as spring, wood, potentiometer, connecting wire and bolt.
The gas pedal and brake pedal have similar design structure in our case and the potentiometer connected with 18 cm wood bar. when some forces apply to this pedal the spring change from its initial position or the force extend or compress a spring by some distance x scales linearly with respect to that distance due to the expand of the spring the potentiometer value changes the ranges between (400-500) ohms and the brake pedal also the range between (600-700) ohms.

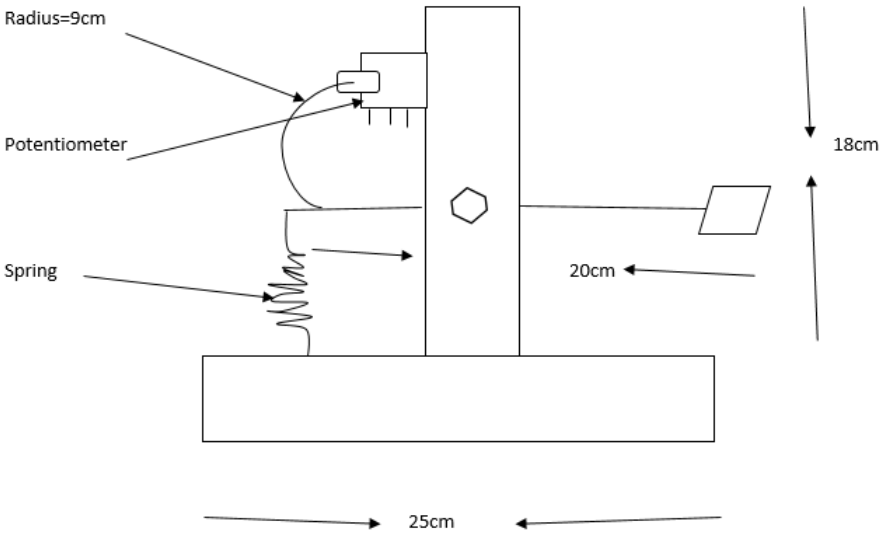

Figure 19: Data flow diagram of the pedals

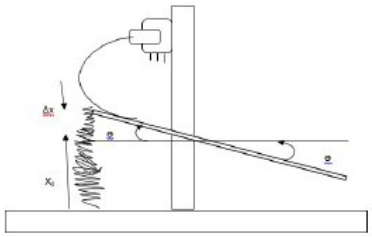

Figure 20: Potentiometer change with respect to pedal

$$\textbf{Voltage}(V) = (cos(\triangle x/l)) * (5/270)v \tag{3}$$

$$\textbf{tan}\ (\theta) = \triangle x/12cm \tag{4}$$

V is a voltage which is caused by the value of potentiometer

Table 3: Arduino UNO technical specification

$\triangle x(cm)$	$\triangle \theta(degee)$	V
0	0	0
1	0.0831	0.00153
2	0.165	0.003
3	0.244	0.004
4	0.321	0.0059

4.2 Electrical design and implementation

4.2.1 potentiometer

potentiometer for steering Wheel

Here the steering wheel is connected to the potentiometer through steering shaft. When the steering is rotated the wiper of the potentiometer is displaced. The variation in position of wiper of potentiometer results in variation of resistance of the potentiometer output. This variation in resistance results in variation of voltage at the output from 0V to 5V. This output voltage is in analog form; the microprocessor receives only digital input, hence the analog to digital converter is used to convert analog output of the potentiometer to digital input of microprocessor.

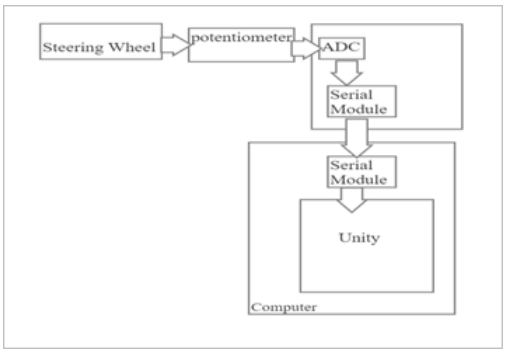

Figure 21: Data flow diagram of Steering wheel

Potentiometer for Brake pedal

Whenever gas pedal is pressed corresponding potentiometer produce an analog voltage proportional to a displacement.

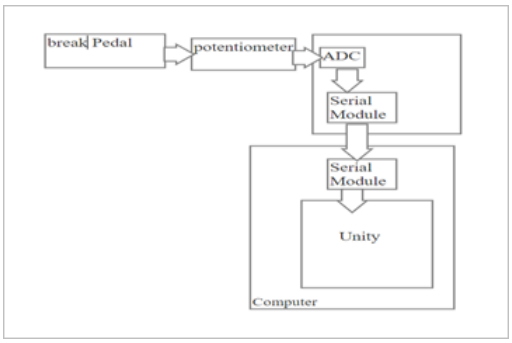

Figure 22: Data flow diagram of Brakepedal

Potentiometer for Gas pedal

Whenever gas pedal is pressed corresponding potentiometer produce an analog voltage proportional to a displacement.

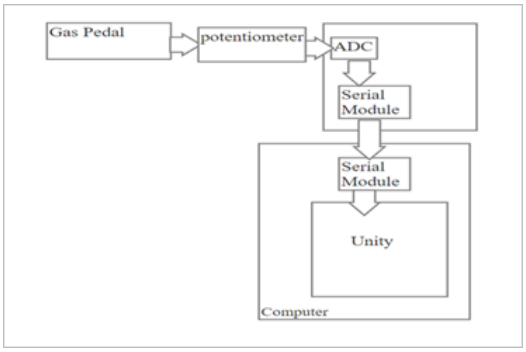

Figure 23: Data flow diagram of Gas pedal

4.2.2 Push button for car ignition

when a driver pushes this button, the car will start ignition. Ignition sound of the car will start.

4.2.3 Push button for horn

After the car ignition is started, when a driver pushes this button, the car will make horn

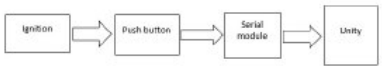

Figure 24: Data flow diagram of the car ignition and horn

4.2.4 Resistor

is a component that resists the flow of direct or alternating electric circuit. Resistors can limit or divide the current, reduce the voltage, protect an electric circuit, or provide large amounts of heat or light.
In our project we use 220 ohm resistor to connect with the leds used in shifter to indicate the position of the shifter(park, drive, neutral, reverse,)

4.2.5 Serial Communication

This concept used for communicating Arduino and Unity 3D. one of the serial communication protocols supported by Arduino UNO is UART. This serial data transmission formats are available on Arduino Used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART), and some have several[9].

UART stands for Universal Asynchronous Reception and Transmission and is a simple communication protocol that allows the Arduino to communicate with serial devices. The UART system communicates with digital pin 0 (RX), digital pin 1 (TX), and with another computer via the USB port. This peripheral, found on all Arduino boards, allows the Arduino to directly communicate with a computer thanks to the fact that the Arduino has an onboard USB-to-Serial converter.

Steps for Unity And Arduino Uno Serial Communication

Arduino Part

- Set a baud rate in the Arduino code by **Serial.begin(9600)**;
- **Serial.println()**; Prints data to the serial port as human-readable ASCII text

Unity Part

- Use the **System.IO.Ports** library

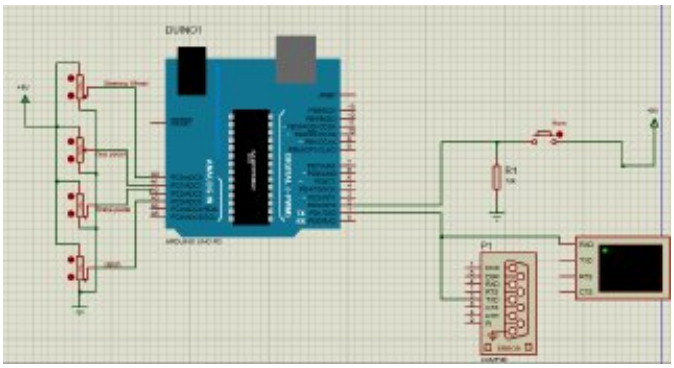

Figure 25: Electical desgin

- Set the port and baud rate by **SerialPort stream = new SerialPort ("device name here", 9600)**;
- Initialize: **stream.Open()**;
- In update, read the stream data by **stream.ReadLine()**;

4.2.6 Power Supply

The power supply circuit design is one of the important parts of this project, without a power supply the electronic devices such as sensors not function. Basically, we have two system which need power supply i.e. personal computer in which Unity 3D is installed and Arduino UNO. In our project we use directly mains (220 v AC) for computer power source and a computer USB port to power Arduino board. Major devices attached with Arduino board requiring power supply in our project are push buttons, and potentiometer. For this purpose, we use Arduino +5V pin to power such connected device.

4.2.7 Brief Description of Electrical Design

The Arduino board contains 10-bit analog to digital converter. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, 0.0049 volts (4.9 mV) per unit.
Mapping of the input voltage computed using this formula:

$$\textbf{Mapped value}(MP) = (Vin/5V) * (1023) \tag{5}$$

Vin = input voltages which are given to the analog ports of the arduino.
5V= which is the operating voltage of arduino

1023= the decimal value of 10bit
Since we are using potentiometer as analog sensor and for every rotation we get different value of which varies between 0v-5v and that cause MP varies between 0-1023.This change will be used to manipulate virtual world car with the corresponding value. But as we can see from the analog port, we read value from 0 to 1023 through all analog pin. This will make confusion especially when this data is sent through serial communication and used by another software. In order to solve this problem boundary is set for ever sensors data reading. for example boundary for Steering 0 up to 360, Gas pedal 400 up to 500 etc.This will help us to compare incoming data through serial and compare it with some boundary values. so that we can easily differentiae specific value of each sensors.

There are also digital sensors that is used for sensing ignition state, horn and lighting switch. These sensors are connected to digital pin of Arduino UNO. The state of these sensors is continuously checked. In addition to these each digital sensor is connected to ground using pull-down resistor. This will help our processor to not confuse to determine initial state of switch.
The following table describes the boundary values of corresponding analog and digital sensors.

Table 4: Analog sensors and data send through serial communication

Analog sensors	**Data send through serial communication**
Steering wheel	Between 0 and 360
Gas Pedal	Between 400 and 500
Brake pedal	Between 600 and 700
Hand Break	Between 900 and 920

Table 5: Digital sensor and data send through serial communication

Digital sensors	**Data send through serial Communication**
Start ignition on rising edge	930
Stop ignition on falling edge	935
Horn start (on rising edge)	970
Horn stop (on falling edge)	975
Turning to left	950
No light	955
Turning to right	960

4.2.8 Analysis on the boundary data

Using MP equation that is given above

- Steering data = (vin/5v*360), this is because the boundary should have to be mapped between 0 and 360. When we change vin between 0 up to 5 volts the steering date changes between 0 up to 360.
  - Inside the steering data the boundary between 0 and 179 used for turning the steering on the left side.
  - The boundary between 180 and 360 used for turning the steering on the right side
  - If the steering data is 180 the steering becomes neutral or steady state
- Gas pedal data = (vin/5v*100)+400, this is because the boundary should have to be mapped between 400 and 500. when vin changes between 0 up to 5v the Gas pedal data changes between 400 up to 500.
- Brake pedal = ((vin)/5v*100) + 600, this is because the boundary should have to be mapped between 600 and 700.
- Hand brake = (vin/5v*20)+900 , this is because of the boundary.

4.3 Software Implementation

As the project is car simulator type it is obvious to use 3D gaming software like Unity 3D, Unreal engine,In this project we used Unity because of its user-friendly environment and with a lot of tutorials for beginners, well documented and easy to refer manual and forum and availability of expertise help.Unity is a powerful cross-platform game engine and a user-friendly development environment and is used to create both three-dimensional and two-dimensional games as well as simulations for its many platforms. As the engine supports a primary scripting in C and JavaScript the project is done by using mainly C language . As Unity's built-in physics engines provide components that handle the physical simulation.by controlling the physics from scripts (code) an object in a game must accelerate correctly and be affected by collisions, gravity and other forces also you can give an object the dynamics of a vehicle, a machine, or even a piece of fabric. As this project is Thing which have done inside unity3d to fit our general objective

4.3.1 Menu

Which is the front page of the game which allows the player to interact with the game. It contains

Figure 26: Menu

Figure 27: Inside start option

- **start bar** is pressed it begin to play the game
- **Option bar** is pressed it allows the player to choice form available game scenes(which are parallel parking test, s-test driving test, free driving test and English 8 driving test)
- **Quit bar** is pressed it allows the player to exist for the game

4.3.2 Parallel parking test game scene

In this game scene the driver required to park the car in the proper allotted parking area. To cheek whether the driver is doing well we have used two mechanisms

- Cones with collision effects placed at the right and left ends of the road to detect collision, based on how many cones the driver collide with the performance changes.
- By determining the x and z coordinates of the parking area, we can detect whether the driver parks in the right location.

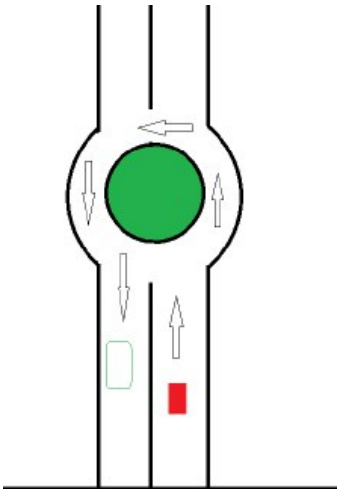

Figure 28: Parallel parking test game scene

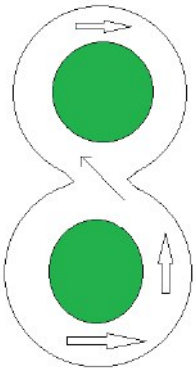

Figure 29: English 8-test game scene

4.3.3 English 8-Test game scene

In this game scene the driver required to drive in number eight shaped road and roundabout to enhance the driver's ability to derive in real roads which has many curves and roundabouts. Maximum score given to when the car properly pass the roundabouts without colliding.

4.3.4 S driving Test game scene

In this game scene the driver is required to drive in s shaped road to enhance and test the ability to drive in roads which has many curves. Similar to the

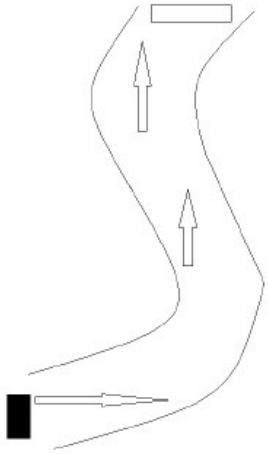

Figure 30: S driving test game scene

above environments, the score reduced when the driver hits the cones that are placed in s shape.

4.3.5 Free Driving game scene

In this game scene the driver required to drive in highways. This game scene is different from the others that the driver in not alone here. There are cars in addition to the drivers car. The performance measured here is not by colliding with the other cars.

The second method that the game system employs in controlling the car is the use of trigger detection and artificial environment perception. Trigger detection is a popular mechanism used in contemporary video games, and it supports detecting a specific game object when it is within the vicinity of another game object. This method can determine if the game-controlled car may run into some object such as a wall. To prevent the collision from happening, the game system can adjust the steering and braking output levels applied to the car. The mechanism that supports this feature is the trigger area added to the car. If a wall enters into this trigger area, the game system becomes aware of where the wall is with respect to the car and it would adjust its speed. With this addition of the trigger area, the game system is able to control the car so that it can navigate the track satisfactorily. When an obstacle or another car enters into the trigger detection area, the output levels of the highlighted car will be adjusted so as to steer the high lighted car away from the obstacle.

4.4 Simulation Results

4.4.1 Analog sensors data (steering, gas, brake, handbrake) on Proteus

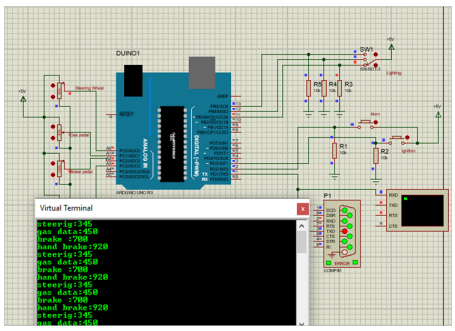

Figure 31: Analog data sent from Proteus

The above simulation result shows the analog data sent from the Proteus serially and the data displayed in the virtual terminal. the sent data is steering , gas data, hand brake and brake data.

4.4.2 Digital sensors data (ignition, horn, light) on Proteus

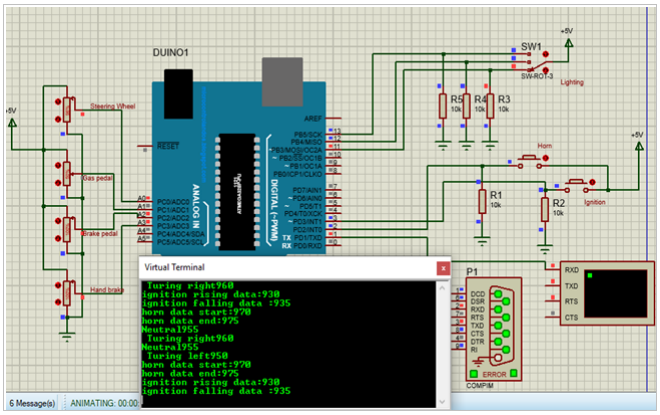

Figure 32: Digital data sent from proteus

The above simulation result shows the digital data sent from the Proteus serially and the data displayed in the virtual terminal. the sent data is ignition rising /falling data , horn data start/end, and turning right/left.

4.4.3 Analog sensors data (steering, gas, brake, handbrake) on Unity

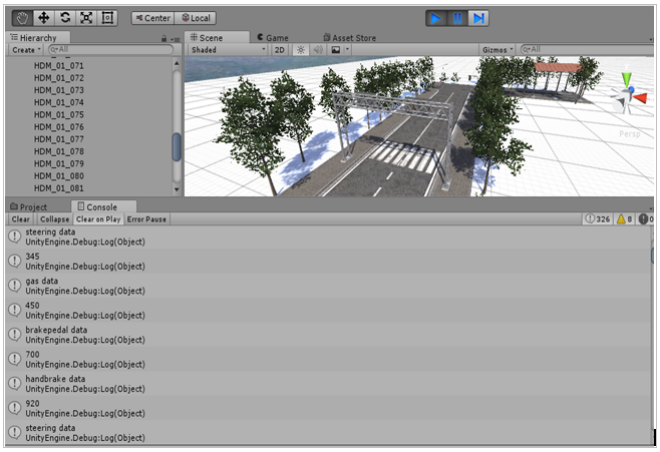

Figure 33: Analog data received in Unity

This figure express the transfer of analog data from protues and received by unity and show the virtual environment. the received data will be steering ,gas, brake, and handbrake .

4.4.4 Digital sensors data (ignition, horn, light) on Unity

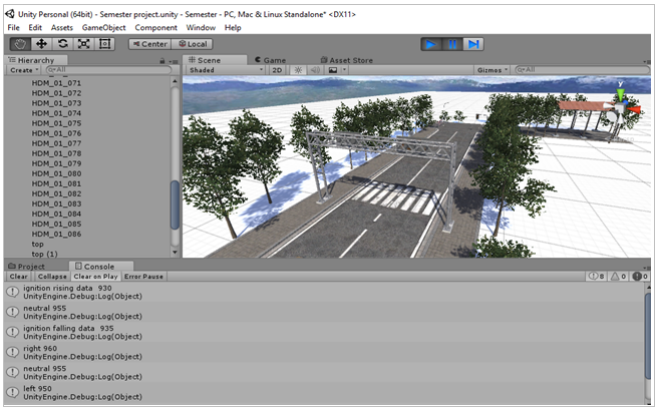

Figure 34: Digital data received in Unity

The above simulation result shows the digital data sent from the protues serially and the data displayed in the virtual terminal. the sent data is ignition rising /falling data , horn data start/end, and turning right/left.

4.5 Hardware Implementation

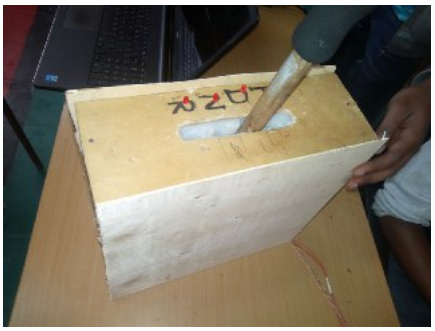

Figure 35: Hardware Implementation of shifter

Figure 36: Hardware Implementation of pedals

Figure 37: Hardware Implementation of steering

Chapter Five

5 Conclusions and Recommendations

5.1 Conclusion

This paper describes plementation of car driving simulator with Arduino UNO and Unity 3D. both hardware and software part of listed functionality are designed, simulated, implemented and tested practically. We found that simulation behavior of electrical component is the same as practical one. Artificial driving training environment is modeled and created in unity 3D software which is used to communicate and controlled with Arduino UNO. Communication between Arduino and Unity is tested. Using sensors attached with Arduino, driver can communicate with virtual environment effectively all the while driver can continuously interact with the change. this project is illustration of how to control virtual world car taking the input from real world. Finally, we can conclude that, if one can create virtual environment and have controller hardware it is possible to represent virtual car property related with voltage and current this will help to make driving simulator that is useful for driving trainee. And this will help trainee to grasp pre-driving skill before practical on road training.

5.2 Recommendation

In addition to work done in the project, some improvements need to be made in the future work.

- Our aim is to make this project useful for driving trainees, it would be better if the environment resembles the actual environment of our country..
- Include trafficking system so that driver will also learn road traffic rules
- AI based traffic system.
- Inaddition to basic features we included in our project other features including more realistic environment like rain, dark so that a driver can be more familiar with real-world scenario.
- Implement Complete examination and computer-based assessment that is similar with on road examination.

References

[1].http://https://itcl.es/en/simulation-and-virtual-reality-products/driving-simulator-for-dr
[2].https://en.wikipedia.org/wiki/Driving-simulator
[3].https://www.arrivealive.mobi/simulator-driver-training-and-road-safety
[4].http://www.xinhuanet.com/english/2018-08/16/c-137393273.htm9
[5].https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2966567/
[6].https://www.arduino.cc/en/Main/arduinoBoardUno¿
[7].https://en.wikipedia.org/wiki/Unity-(game-engine)
[8].https://www.alanzucconi.com/2015/10/07/how-to-integrate-arduino-with-unity/
[9].https://learn.sparkfun.com/tutorials/analog-to-digital-conversion/all
[10].https://en.wikipedia.org/wiki/Potentiometer
[11].https://en.wikipedia.org/wiki/Push-button
[12].https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3561885/
[13]. https://www.safedrivingforlife.info/using-road

6 Appendix

6.1 Flow chart

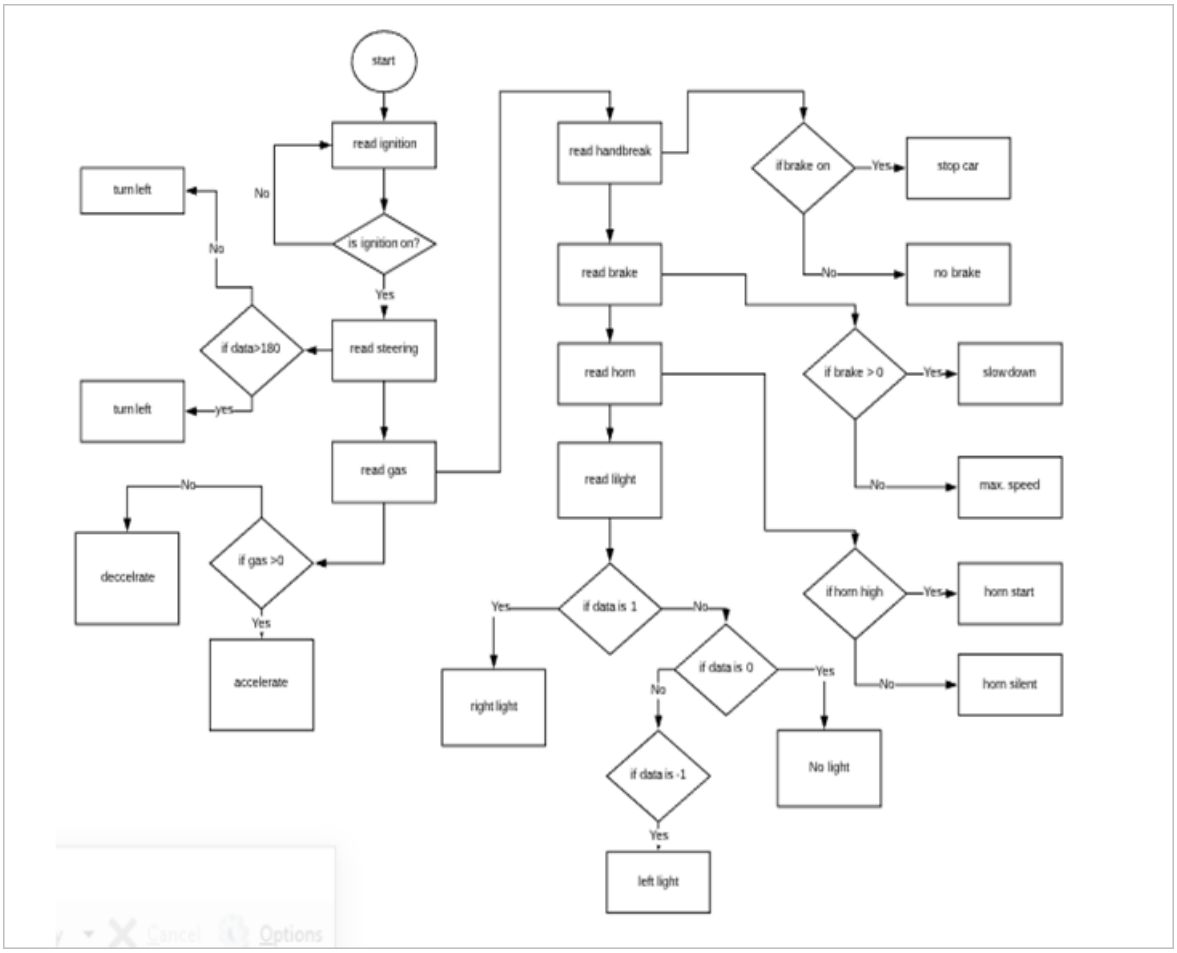

Figure 38: Flow chart of the system

Sample code

6.2 Sample arduino code

void setup()

pinMode(2,INPUT);
pinMode(3,INPUT);
pinMode(11,INPUT);
pinMode(12,INPUT);
pinMode(13,INPUT);
Serial.begin(9600);

void loop()

steering = analogRead(A0);
gas = analogRead(A1)
brake = analogRead(A2);
handbrake = analogRead(A3);
steering$_o ut = map(steering, 0, 1023, 0, 360)$;
$gas_o ut = map(gas, 0, 1023, 400, 500)$;
$brake_o ut = map(brake, 0, 1023, 600, 700)$;
$handbrake_o ut = map(handbrake, 0, 1023, 900, 920)$;
$Serial.print("steerig : ")$;
$Serial.println(steering_o ut)$;
$Serial.print("gasdata : ")$;
$Serial.println(gas_o ut)$;
$Serial.print("brake : ")$;
$Serial.println(brake_o ut)$;
$Serial.print("handbrake : ")$;
$Serial.println(handbrake_o ut)$;
$if(digitalRead(2) == 1prev_h orn == 0)$

6.3 Sample C sharp code for unity

Controller class

public class controller : MonoBehaviour
public Slider gasslider;
public Slider brakeslider;

```
    public Slider handbrakeslider;

    public controldata mydata = new controldata ();
    public AudioSource horn;
    public AudioSource Car¡gnintion;

    int prev-1=0;
    int prev-2=0;

    int prev-lig=0;

    int prev-light1 = 0;
    intprev − light2 = 0;
    intprev − light3 = 0;

bool car-lgnition=false;

public Vector3 centerOfMass;
public WheelCollider wheelFL;
public WheelCollider wheelFR;
public WheelCollider wheelRL ;
public WheelCollider wheelRR;

void Start ()

gasslider.value = 0;
brakeslider.value = 0;
handbrakeslider.value = 0;
rb = GetComponent¡Rigidbody¿();
rb.centerOfMass = centerOfMass;
SetValue ();

void SetValue()
myForwardFriction = wheelRR.forwardFriction.stiffness;
mySidewayFriction = wheelRR.sidewaysFriction.stiffness;
slipForwardFriction = 0.04f;
slipSidewayFriction = 0.08f;
```