PROJECT REPORT

COURSE:
HUMAN COMPUTER INTERACTION

INSTRUCTOR:
PROF. DR. TABIN HASAN

GROUP MEMBERS:

NAME	ID
Habib, Md Amir	19-41429-3
Rahman, Ashiqur	18-38779-3
Emon, Abdur Rahman	18-36603-1
Momu, Sharmin Akter	18-36567-1

Department of Computer Science

Faculty of Science & Technology

American International University Bangladesh

SUMMER, 2022

❑ Introduction

Today, gaming experience is a different story than what it was 20 years ago. Gaming imitates the real world and provide a virtual world, allowing users to receive and disperse information between the two worlds through graphical user interfaces. Such communication with a virtual world can be enhanced in a more complete imitation of the real space by the introduction of an artificial support technology called haptics. Haptic research related to home entertainment and computer games has blossomed and impacted the development of technology during the past few years. A haptic interface is a device that allows a user to interact with a computer by receiving tactile and force feedback. Force feedback technology enhances the game experience by creating a more realistic physical feeling of playing a game, providing players a higher sense of involvement as well as new and interesting ways to interact with the game environment. This physical experience can contribute in improving the physical skills of the players, improving driving training simulation and more.

In this project, we have worked together to assemble a working gaming steering wheel with force feedback technology integrated. This technology is achieved with the help of electric motor inside the steering wheel. The motorized motion creates resistance in accordance to how you are handling the wheel.

❑ Project background

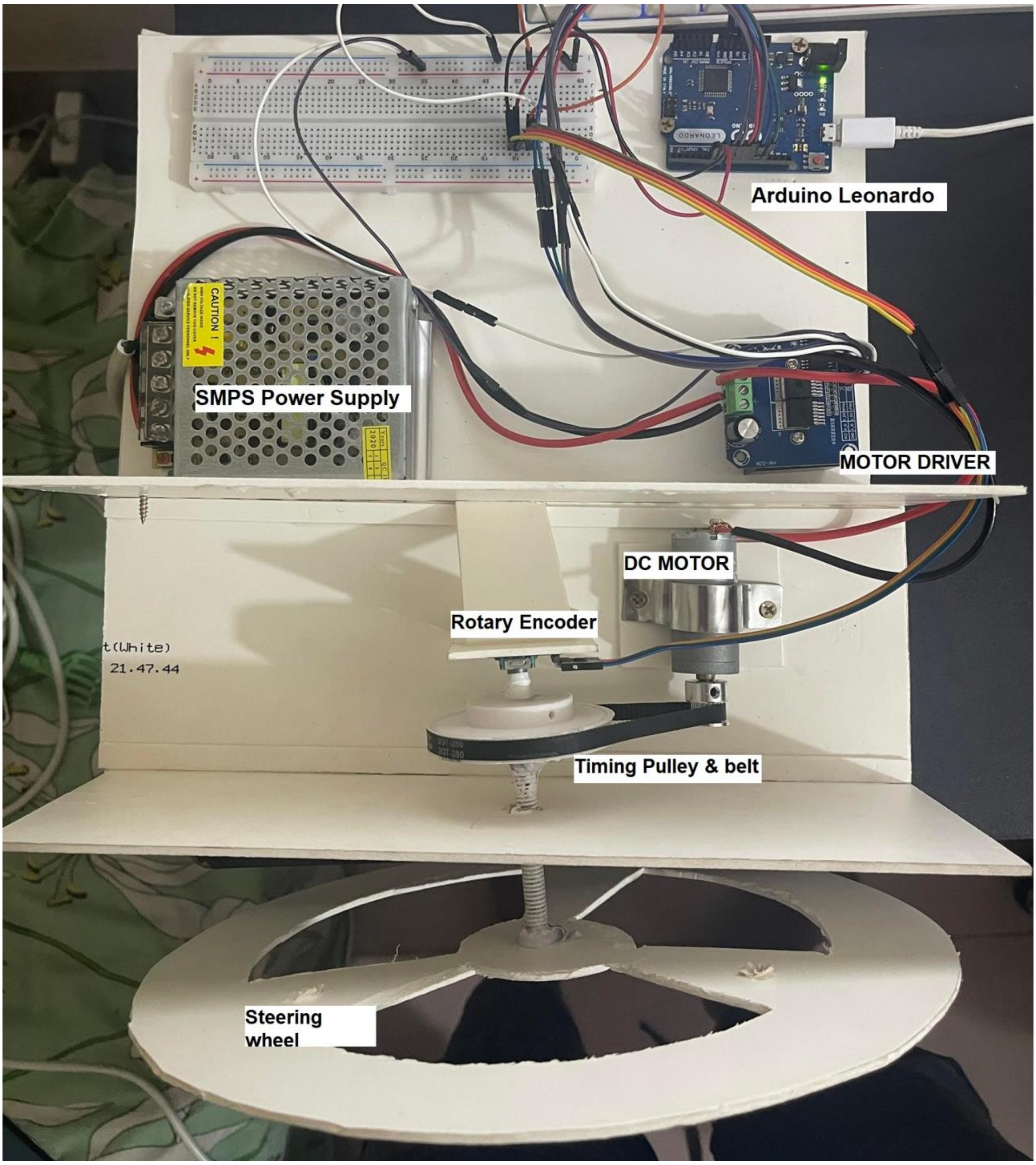
For many past years in racing game or simulation games, players and users would interact with the games work with the help of signals and interactions. Normally, games send you signals with the help of visuals and audio, and you react to them by pushing buttons on your keyboard or controller. However, in an attempt to achieve a more immersive experience many haptics technology have been explored, particularly force feedback. Force feedback adds an extra layer for both signals and interactions. Not only the game can send you extra signals with the help of motorized motion in your steering wheel, you also get extra opportunities to react to those signals very precisely by turning your wheel or pushing the pedals. The extra dimension of physical signals, interactions and opportunities that you experience with a force feedback steering wheel leads to a new extent of communication with the game. With added communication, you feel much more absorbed and the accurate steering movement allowed by a wheel helps achieve the simulation of real driving experience from your computer desk.

Racing steering wheels are the perfect input device for both arcade-style and simulation-style racing. They are already commercially available in the market, of different features and capabilities for the use in virtual environments. But they can be highly costly and exclusive. Our goal in this project, to ‘DIY’ a basic force feedback steering wheel from scratch with cheap expenses, where the electric motor alters as the driving conditions change, providing mainly realistic resistance to the wheel rotation and providing precise throttle response.

❑ Methodology

Our gaming steering wheels support force feedbacks are which typically done by equipping with a single motor that applies a resistive force to the steering wheel using a belt-driven system attached to the wheel shaft. A rotary encoder is connected to the wheel shaft, so whenever we play the game, the encoder sensor record the rotation measurement and sent the information to the microprocessor, where we have uploaded our written code. Using the code and incoming data, the microprocessor decides how the motor must act to apply the feedback resistive force, and using the motor driver the motor is controlled to apply a force feedback to the wheel depending how much we have turned the wheel.

The circuit for our whole project is given below.



Tools we have used:

Hardware:

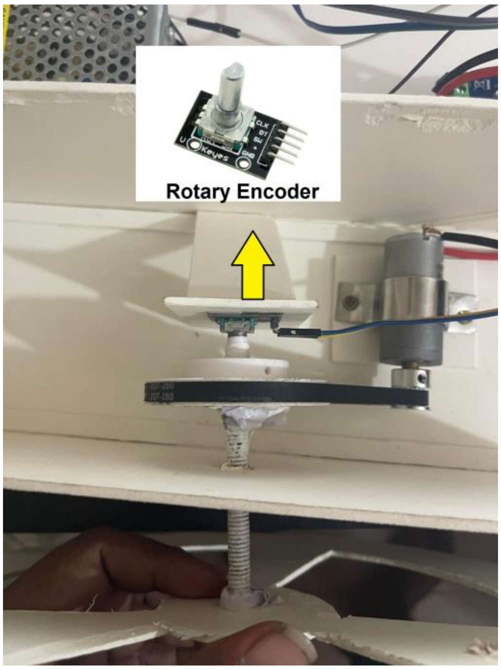
1. DC Motor 12V, 100rpm
2. Timing Pulley and timing belt
3. Arduino Leonardo
4. Rotary Encoder
5. H-bridge Motor Driver BTS7960
6. SMPS power supply 12V 5A
7. Connecting wires and jumper wires
8. Breadboard

Software:

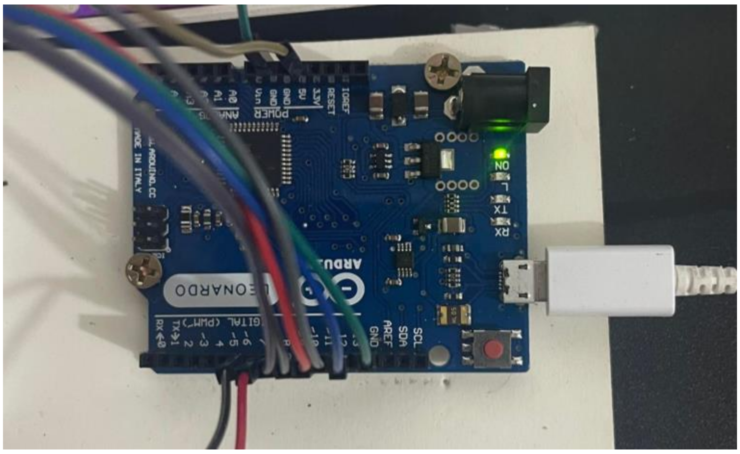
1. Arduino IDE
2. Python

Implementation:

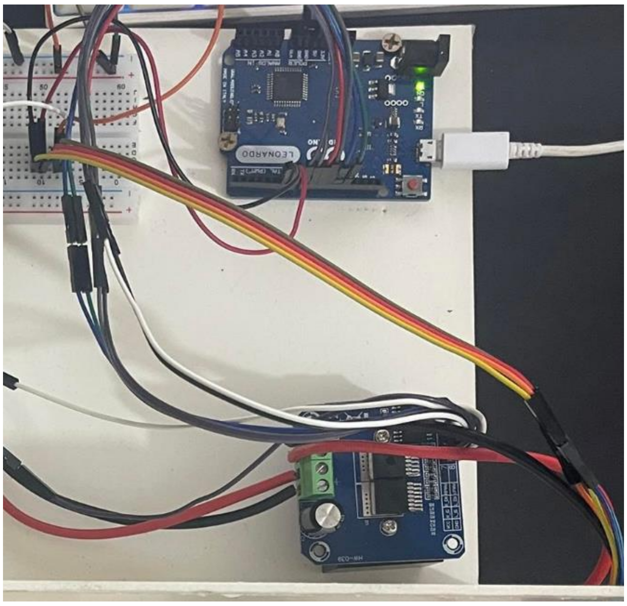
Our steering wheel is connect to a shaft. There is a rotary encoder at the end of the shaft. Whenever we rotate the wheel, the encoder reads the rotation made by the wheel.



The rotary encoder is connected to a microprocessor, in our case Arduino Leonardo. The encoder feeds the input data from wheel to the Leonardo. From where, inputs are transferred to computer as game inputs. In Leonardo we have also integrated custom codes, where using the data from encoder, the code logic computed how much the wheel has rotated and how much resistive force should be applied on the wheel by the feedback motor.

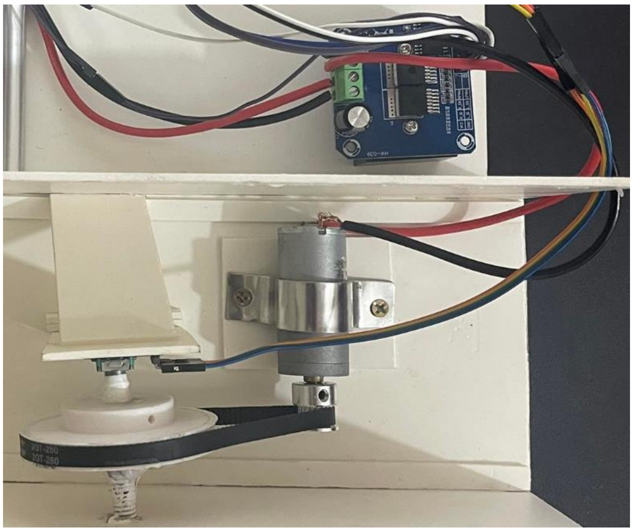


Now the Leonardo signal the motor driver, in what direction and force the motor should rotate to apply the desired force feedback.

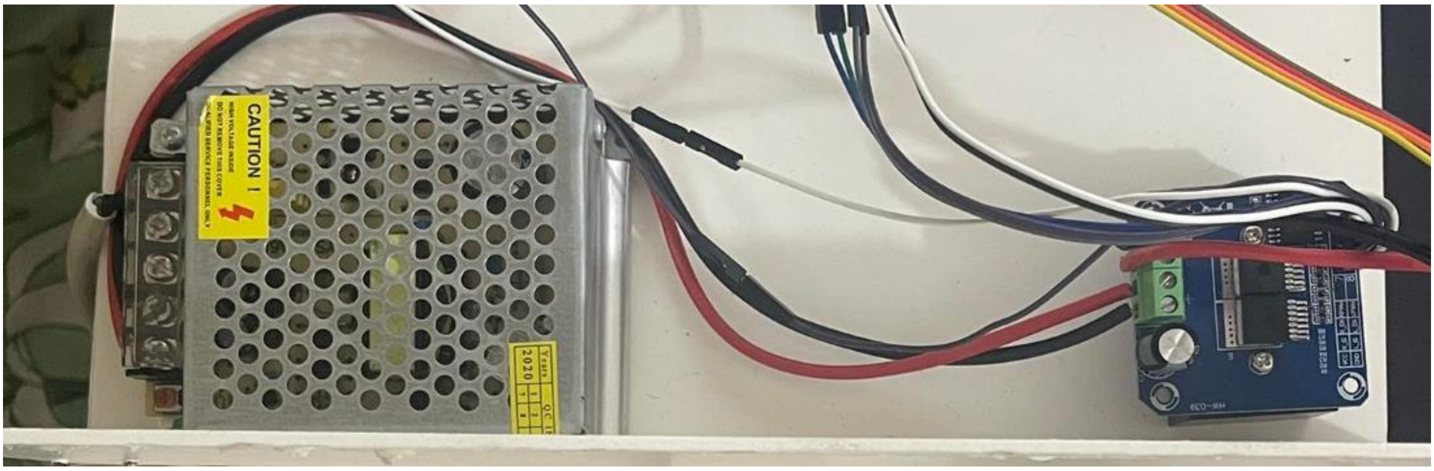


The motor is connected to the shaft of the wheel through a belt-driven system, where we have used timing pulleys and timing belt. When the force feedback force is applied by the motor, it rotates the shaft in opposite direction, creating a resistive force on the wheel while driving.

For example, it the rotation input from the wheel is not at 0 position, and the force feedback wants the rotation of the wheel at 0 position, the motor will rotate and apply resistive force in a way to bring the wheel rotation at 0 position.



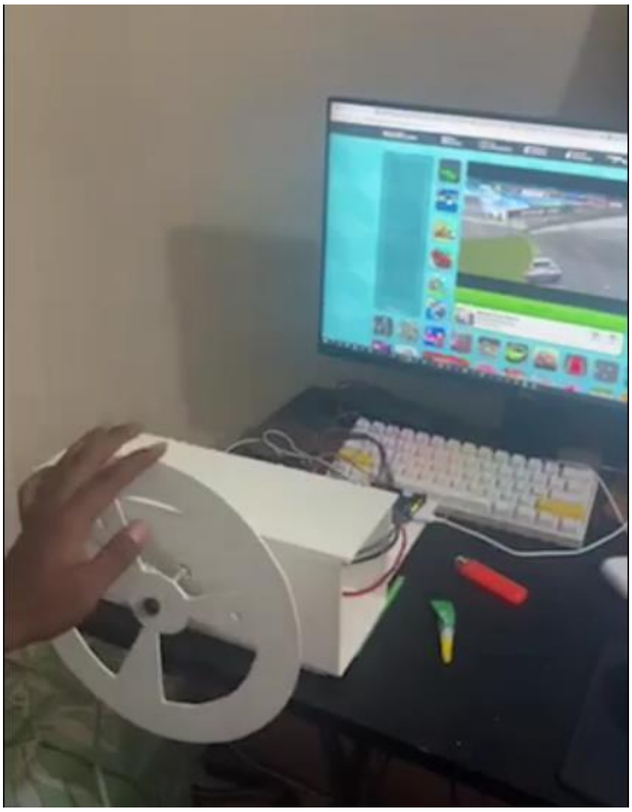
To power the motor, we have used a 12V power supply



❑ Result

After assembling the hardware and running the codes, we have applied it to a game to check for results. The game took inputs properly of the wheel rotation coming through the Leonardo using our python code in computer. The control of the car using the wheel was achieved.

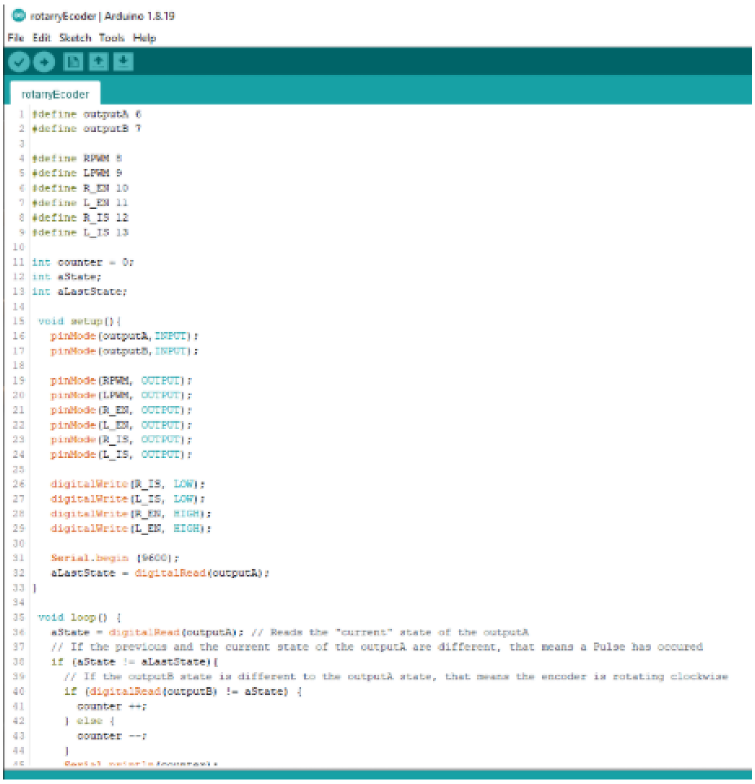
We can also experience force feedback by the motor. The more we rotate the wheel, the more torque we experience by the motor trying to bring the wheel to initial position



```
import serial
import pyautogui
import time

ArduinoSerial = serial.Serial('COM7', 9600)
nowDown = 0
#data = 100;

while 1:
    data = str(ArduinoSerial.readline().decode('ascii'))
    #data = data-1
    pos = int(data)
    pyautogui.keyDown("w")
    print(pos)
    if pos>0 and nowDown==0:
        pyautogui.keyDown("d")
        nowDown = -1
        #print('pos1')
    elif pos<0 and nowDown==0:
        pyautogui.keyDown("a")
        nowDown = 1
        #print('pos2')
    elif pos==0:
        if nowDown == -1:
            pyautogui.keyUp("d")
            nowDown = 0;
            #print('pos3')
        elif nowDown == 1:
            pyautogui.keyUp("a")
            nowDown = 0;
            #print('pos4')
```



```
#define outputA 6
#define outputB 7

#define RPWM 8
#define LPWM 9
#define R_EN 10
#define L_EN 11
#define R_IS 12
#define L_IS 13

int counter = 0;
int aState;
int aLastState;

 void setup(){
   pinMode(outputA,INPUT);
   pinMode(outputB,INPUT);

   pinMode(RPWM, OUTPUT);
   pinMode(LPWM, OUTPUT);
   pinMode(R_EN, OUTPUT);
   pinMode(L_EN, OUTPUT);
   pinMode(R_IS, OUTPUT);
   pinMode(L_IS, OUTPUT);

   digitalWrite(R_IS, LOW);
   digitalWrite(L_IS, LOW);
   digitalWrite(R_EN, HIGH);
   digitalWrite(L_EN, HIGH);

   Serial.begin (9600);
   aLastState = digitalRead(outputA);
}

 void loop() {
   aState = digitalRead(outputA); // Reads the "current" state of the outputA
   // If the previous and the current state of the outputA are different, that means a Pulse has occured
   if (aState != aLastState){
     // If the outputB state is different to the outputA state, that means the encoder is rotating clockwise
     if (digitalRead(outputB) != aState) {
       counter ++;
     } else {
       counter --;
     }
     Serial.println(counter);
```

❑ Discussion

Whether you’re a beginner or a professional sim racer, you’ll immediately notice the difference when playing with racing wheels equipped with Force Feedback effects as it enables you to react faster and more accurately when compared to a keyboard or controller. Steering wheels with great Force Feedback effects allow you to feel the difference between huge turns or small tricky turns when driving or oversteer. Steering wheels with the optimized Force Feedback effects also simulate the same levels of force as that of a real race car making involvement truly enjoyable in terms of gameplay when playing your sim racing games.

However, we have come across some limitations while building the project. We have worked and altered to overcome those limitations. On first attempt we tried to use gears cut out from wood for rotating the wheel shaft, but it wasn’t giving satisfying results. Hence, we shifted to another approach of using belt driven mechanism, for which we 3d printed some timing pulley. Moreover, belt driven wheels has more advantages than gear driven wheels. It utilizes a small motor with a belt and pulley system to generate Force Feedback effects. This system produces more torque and stronger Force Feedback than gear-driven wheels. Gear driven are chunkier and less smooth and lends to create lags if not properly fitted.

❑ Conclusion

So finally we have achieved our goal to build a basic force feedback gaming steering wheel from scratch with cheap expenses. Our wheel tries to replicate what the driver would feel if he were to race under the same conditions in the same location in real life and, ultimately, provide you with a ton of extra information on how your car is performing so you can make minor adjustments to improve it. But if you’re expecting force feedback to create a 1:1 simulation of real life driving, you’re being overly optimistic. To accomplish that, technology still has a long way to go. Although, there are many FFB racing steel commercially sold in the market, usually direct-driven that can give you near 1:1 simulation of real life driving, but ultra-realistic expensive force feedback wheel is probably not worth it unless you’re a professional racer. For most people, spending a lot of money on a belt driven force feedback wheel is much better option. Our force feedback gaming wheel aims to create the most enjoyable driving experience. Realistic simulation does not necessarily mean it’s enjoyable. Every person has a unique driving style and the term realistic is subjective. Besides, many racing games provide a plenty of options like: understeer, vertical load, vehicle weight, spring and damper, slip effect and many more. You can adjust each of these settings to the intensity of your liking to achieve your definition of the term “realistic”.