Fingerprint Sensor Vehicle Starter

A PROJECT REPORT

Submitted by

Pallavi Kumari	(19011445012)
Prity Kumari Singh	(19011445015)
Nidhi Kumari	(19011445011)
Akash Aman	(19011440002)
Aniket Singh	(19011440006)
Ravi Shankar Thakur	(19011440014)

Under the guidance of

Prof. Priya Ranjan Satapathi

Assistant Professor, Dept of CSE
(Department of Computer Science and Engineering)

in partial fulfillment of the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING
(NBA ACCREDITED)
of

CHAIBASA ENGINEERING COLLEGE
(Estd. by Government of Jharkhand & run by Techno India under PPP)

Bistumpur, Jhinkpani, West Singhbhum,
Jharkhand, 833215

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented, fabricated, or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

.....

Signature of the Students

CHAIBASA ENGINEERING COLLEGE

CERTIFICATION

Certified that this project report titled "**Fingerprint Sensor Vehicle Starter"** is the bonafide work of "Pallavi Kumari-19011445012, Prity Kumari Singh-19011445015, Nidhi Kumari-19011445011, Akash Aman-19011440002, Aniket Singh-19011440006, Ravi Shankar Thakur-19011440014" who carried out the project work under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Mr. PRIYA RANJAN SATAPATHI
ASSISTANT PROFESSOR
DEPT. OF COMPUTER SCIENCE & ENGINEERING

SIGNATURE
Ms. BEAS BHADRA
HEAD OF THE DEPARTMENT
DEPT. OF COMPUTER SCIENCE & ENGINEERING

ABSTRACT

Common vehicle locks are insufficient in providing adequate security for vehicle owners. Conventional vehicle locks are widely recognized by thieves, and they can be effortlessly compromised.

With the inadequacy of traditional key locks on motorcycles, there is a necessity for more advanced security alternatives. Biometric systems, which include fingerprint identification, offer a reliable and efficient option. As every person's fingerprint is unique, it can be utilized in various security measures.

The purpose of this paper is to explore the possibility of using fingerprint recognition to replace traditional key locks for starting motorcycles. The paper provides an extensive comparison between these two methods. Additionally, prior research in this field is reviewed, which discusses various methods of improving motorcycle security, including adding different locks and alarm units that can notify the bike owner of potential risks.

ACKNOWLEDGEMENTS

It gives us a great sense of pleasure to present the report of the project work undertaken during B.Tech final year. We owe a special debt of gratitude to our project supervisor Prof. Priya Ranjan Satapathi, Department of Computer Science and Engineering, Chaibasa Engineering College, Chaibasa for his constant support and guidance throughout the completion of our project. It is only through his cognizant efforts that our endeavors have seen the light of the day.

We would also like to convey our sincerest gratitude and appreciation to all other faculty members and the staff of the Department of Computer Science and Engineering, who conferred their efforts and guidance at appropriate times. Without them, it would have been very difficult to complete the project on time.
This is the appropriate time to express our gratitude to all of our friends and well-wishers who have been a constant source of encouragement and moral support during the entire period of our project.

……..

Signature of the Students

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The conventional key locks found on motorcycles are widely recognized by thieves, making it simple for skilled thieves to pick them. This makes it easy for thieves to unlock the locks on motorcycles using a master key.

This creates the demand for such type of lock which is new and provides an additional security level. The new and modern lock must be unique in itself i.e. it must be only unlocked by a special and specific key. This type of feature is available in the biometrics locks i.e. the lock which can only be locked and unlocked by the human body features. Biometrics can include face recognition, voice recognition, fingerprint recognition, and eye (iris) recognition. Of all these types of special biometric recognition techniques fingerprint recognition is the most widely used because the fingerprint of every person on the earth is unique and can provide good reliability. Also, the implementation of the fingerprint recognition system is easy and cheap than the other ones. Thus fingerprint recognition locking system can provide better reliability than traditional locks and also is cheaper and easy than the other biometric locking system.

Thus, here we are proposing a model which utilizes the concept of fingerprint recognition in motorcycles to enhance the security level of the vehicle. Some other related work to this model is also reviewed in the next section.
The thesis is divided into 5 sections. The first section provides an introduction to the idea of the thesis. The second section is dedicated to the literature review which provides the related work done about the proposed idea. The third section gives a comparison between the existing models based on the literature review. The fourth section is about the design implementation of the idea and the fifth section is the conclusion.

CHAPTER 2

LITERATURE SURVEY

Security Vulnerabilities Against Fingerprint Biometric System (cs.CR) was proposed by M.Joshi, B.Mazumdar, and S.Dey in May 2020 provides an overview of the fingerprint biometric system and gives details about various current security aspects related to the system.

Faceiris multimodal biometric system using multi-resolution Log-Gabor filter with spectral regression kernel discriminant analysis, IET Biometrics in July 2018 was proposed by B.Ammour, T.Bouden, L.Boubchir, this paper based on 2D Log-Gabor filter bank features extraction from the face and two iris images. SRKDA technique reduces the high dimensionality of the extracted feature vectors. The combination of face and two iris modalities is achieved in different levels of fusion to investigate the performance improvement of each level.

A Conceptual Study on User Identification and Verification Process Using Face Recognition Techniques was proposed by K.Krishna Prasad and S.P Aithal in June 2017 provides the contribution of other researchers in the feature extraction process of fingerprint images using Level 1 or Level 2 features or sometimes both, and how these features are helpful in the broad classification of fingerprint images or in the enhancement of images by reducing noise.

A self-banking biometric machine with face detection applied to the fingerprint and iris along with GSM technology was proposed by Joyce Soares and A.N Gaikwad in April 2016 provides the samples of the fingerprint and iris along with the registered mobile number of the customer need to be collected and saved in the database by the banker if the customer is to access the ATM.

Two Wheeler with Locking Options was proposed by Prashant Kumar R in February 2014 that provides good and effective ways of securing the two-wheeler vehicle with a combination of different types of locking options provided in the vehicle.

CHAPTER 3

PROPOSED METHODOLOGY

Secure with biometrics - This all-in-one optical fingerprint sensor will make adding fingerprint detection and verification super simple. These modules are typically used in safes – a high-powered DSP chip does the image rendering, calculation, feature finding, and searching. Connect to any microcontroller or system with TTL serial, and send packets of data to take photos, detect prints, hash, and search. You can also enroll new fingers directly up to 162 fingerprints can be stored in the onboard FLASH memory. There's a red LED in the lens that lights up during a photo.

Fingerprint processing is divided into two parts i.e., fingerprint enrollment and fingerprint matching. When enrolling, the user needs to enter the finger two times. The system will process the two-time finger images, generate a template of the finger based on processing results and store the template. When matching, the user enters the finger through the optical sensor, and the system will generate a template of the finger and compare it with templates of the finger library. For 1:1 matching, the system will compare the live finger with the specific template designated in the Module; for 1: N matching, or searching, the system will search the whole finger library for the matching finger. In both circumstances, the system will return the matching result, success or failure.

Image processing:- Capture the fingerprint images and process them through a series of image processing algorithms to obtain a clear unambiguous skeletal image of the original grey tone impression, clarifying the smudged areas, removing extraneous artifacts, and healing more scars, cuts, and breaks.

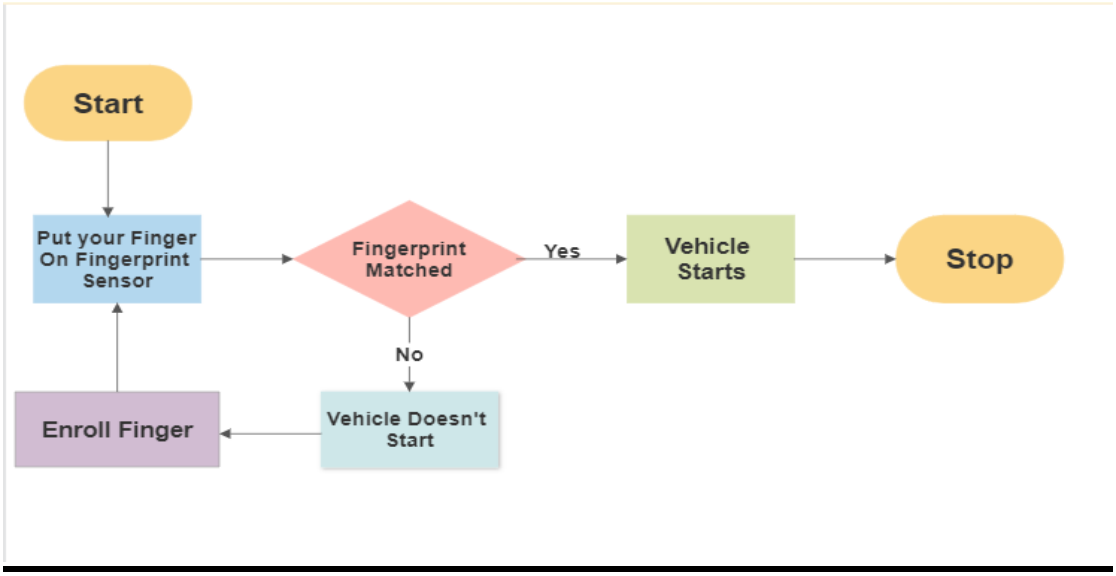

Fig 3.1 Workflow Diagram

- Firstly, we start the vehicle's ignition system using an ignition key switch.
- The user needs to hold the thumb in the fingerprint sensor.
- If the finger is matched, then the vehicle is started to drive.
- If the finger is not matched, if an unauthorized person is trying to start the vehicle, the vehicle will not start and the flow of the petrol is stopped.

CHAPTER 4

SYSTEM DESIGN

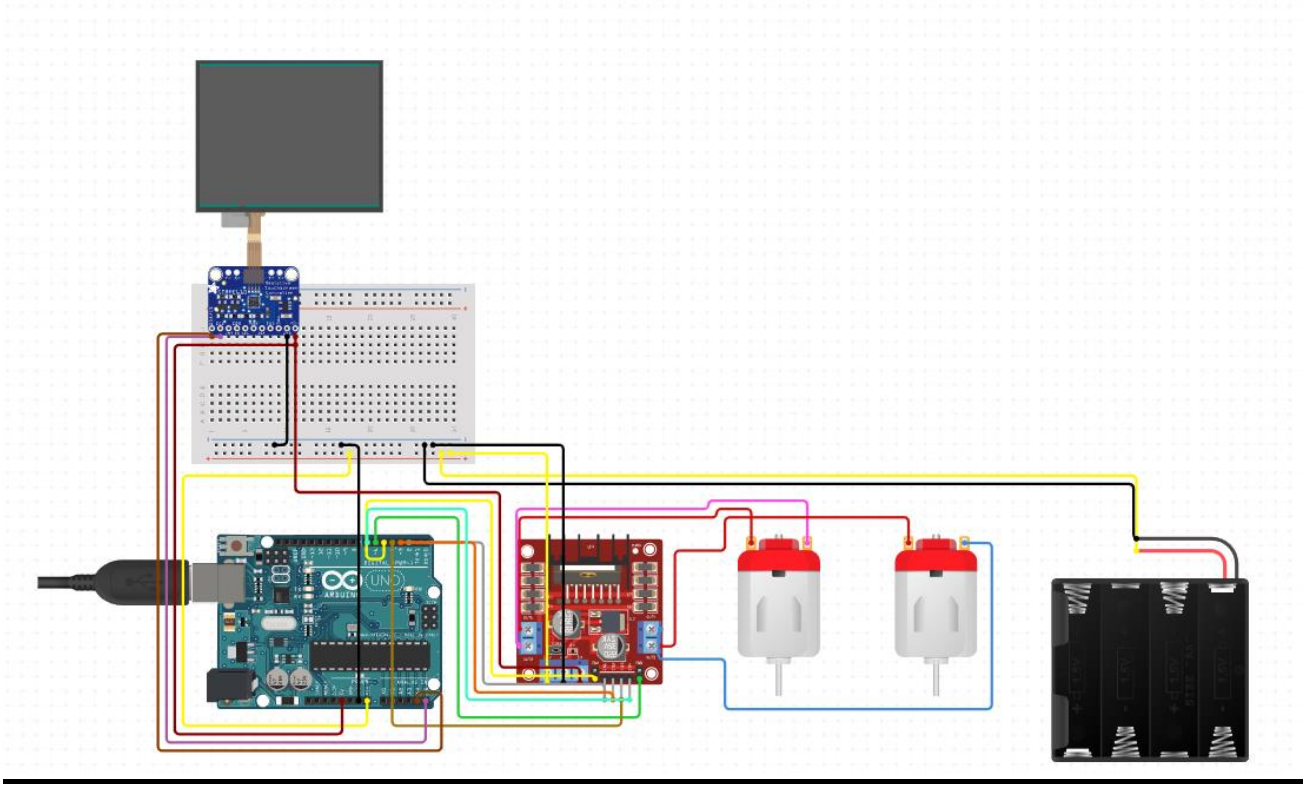

Fig 4.1: Circuit Design of Fingerprint Vehicle Starter

4.1 **Arduino Uno:** Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or

power it with an AC-to-DC adapter or battery to get started. The Arduino Uno board is the most popular board and is mostly referred to the beginners as they are super easy, to begin with, it does not require any specific arduino uno software instead of that all you need is to select the Arduino Uno in the device option before uploading your program.

Fig: 4.2 Arduino Uno

4.2: <u>Fingerprint Sensor R307</u>: Fingerprinting was first created by Dr.Henry Fault, a British surgeon in 1882. It is based on “Key”. Fingerprint recognition is used because of fingerprint of every person on the earth is unique and can provide good reliability. Also, the implementation of the fingerprint recognition system is easy and cheap than the other ones. Thus fingerprint recognition locking system can provide better reliability than traditional locks and also is cheaper and easy than the other biometric locking system. Thus, here we are proposing a model which utilizes the concept of fingerprint recognition in motorcycles to enhance the security level of the vehicle.

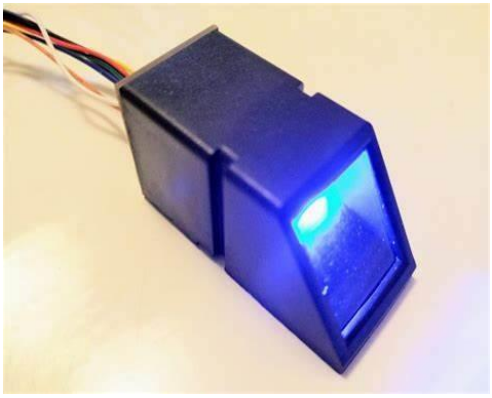

Fig4.3: Fingerprint Sensor R307

Fingerprint Patterns :

- **Arch:-** The ridges enter from one side of the finger, rise in the center forming an arc, and then exit the other side of the finger.

- **Loop:-** The ridges enter from one side of a finger, from a curve, and then exit on that same side.

- **Whorl:-** Ridges from circularly around a central point on the finger.

Fig 4.4: Fingerprint Patterns

Fingerprint matching technique: There are two techniques of fingerprint matching technique:-

- Minutiae-based techniques first find the minutiae points and then map their relative placement on the finger.
- The correlation-based method is able to overcome some of the difficulties of the minutiae-based approach.

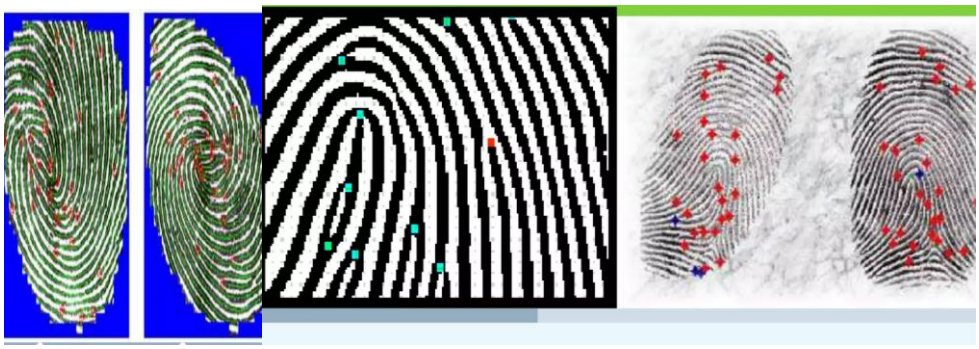

Fig 4.5: Fingerprint Matching Techniques

4.3 L298N Motor Driver: L298N Motor Driver Module is a high-power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control. The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, and 5V jumper in an integrated circuit.

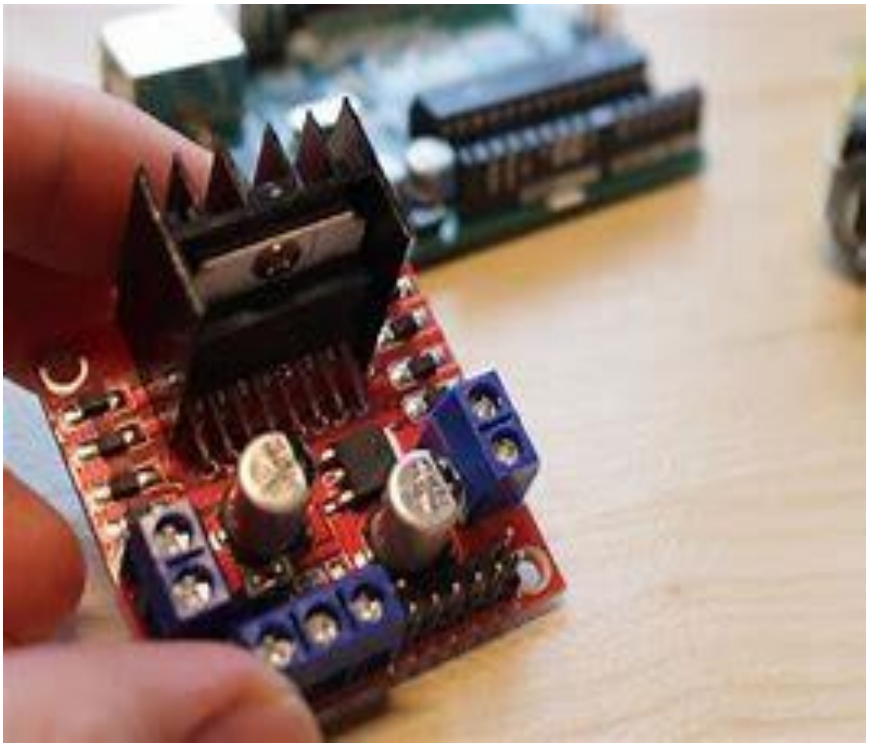

Fig 4.6: L298N Motor Driver

To Enroll:

```
#include <Adafruit_Fingerprint.h>
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup()
{
  Serial.begin(9600);
  while (!Serial);  // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1);

uint8_t readnumber(void) {
  uint8_t num = 0;

  while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
  }
  return num;
}

void loop()                                          // run over and over again
{
  Serial.println("Ready to enroll a fingerprint!");
  Serial.println("Please type in the ID # (from 1 to 127) you want to save this
finger as...");
```

```
  id = readnumber();
  if (id == 0) {// ID #0 not allowed, try again!
     return;
  }
  Serial.print("Enrolling ID #");
  Serial.println(id);

  while (!  getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println(".");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      break;
    default:
      Serial.println("Unknown error");
      break;
    }

  // OK success!

  p = finger.image2Tz(1);
  switch (p) {
    case FINGERPRINT_OK:
```

```
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
```

```
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");

  default:
    Serial.println("Unknown error");
    break;
  }}

// OK success!

p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
Serial.print("Creating model for #");  Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
  Serial.println("Prints matched!");
```

```
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  Serial.print("ID "); Serial.println(id);
  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
  }

else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

}
```

Main Activity (Fingerprint-Motor Connection):

```
#include <Adafruit_Fingerprint.h>

/***************************************************
  This is an example sketch for our optical Fingerprint sensor

  Designed specifically to work with the Adafruit BMP085 Breakout
  ----> http://www.adafruit.com/products/751

  These displays use TTL Serial to communicate, 2 pins are required to
  interface
  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!

  Written by Limor Fried/Ladyada for Adafruit Industries.
  BSD license, all text above must be included in any redistribution
 ****************************************************/


#include <Adafruit_Fingerprint.h>


#if (defined(__AVR__) || defined(ESP8266)) &&
!defined(__AVR_ATmega2560__)
// For UNO and others without hardware serial, we must use software serial...
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino  (WHITE wire)
// Set up the serial port to use softwareserial..
SoftwareSerial mySerial(2, 3);

#else
// On Leonardo/M0/etc, others with hardware serial, use hardware serial!
// #0 is green wire, #1 is white
#define mySerial Serial1

#endif
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
// connect motor controller pins to Arduino digital pins
// motor one
int enA = 10;
int in1 = 9;
int in2 = 8;
// motor two
int enB = 5;
int in3 = 7;
int in4 = 6;

void setup()
{
  Serial.begin(9600);
   // set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  delay(1000);           //don't need to run this at full speed.

  while (!Serial);  // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);
  delay(5);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }
```

```
  finger.getTemplateCount();

 /* if (finger.templateCount == 0) {
    Serial.print("Sensor doesn't contain any fingerprint data. Please run the 'enroll' example.");
  }
  else {*/

     Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
     Serial.println("Waiting for valid finger...");
}

void demoOne()
{
  // this function will run the motors in both directions at a fixed speed
  // turn on motor A
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enA, 150);
  // turn on motor B
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enB, 150);
  delay(2000);
  // now change motor directions
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(2000);
  // now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

```
void demoTwo()
{
  // this function will run the motors across the range of possible speeds
  // note that maximum speed is determined by the motor itself and the operating voltage
  // the PWM values sent by analogWrite() are fractions of the maximum speed possible
  // by your hardware
  // turn on motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  // accelerate from zero to maximum speed

 for (int i = 0; i < 256; i++) {
analogWrite(enA, i);
analogWrite(enB, i);
delay(20);
}
// decelerate from maximum speed to zero
for (int i = 255; i > 0; --i)
  {
    analogWrite(enA, i);
    analogWrite(enB, i);
    delay(20);
  }


  // now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
void demoFour()
{
  // this function will run the motors in both directions at a fixed speed
  // turn on motor A
```

```
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  // set speed to 200 out of possible range 0~255
  analogWrite(enA, 150);
  // turn on motor B
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  // set speed to 200 out of possible range 0~255
  analogWrite(enB, 150);
  delay(1000);
  // now change motor directions
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(1000);
  // now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}

/*void demoTwo()
{
  // this function will run the motors across the range of possible speeds
  // note that maximum speed is determined by the motor itself and the operating voltage
  // the PWM values sent by analogWrite() are fractions of the maximum speed possible
  // by your hardware
  // turn on motors
  digitalWrite(in1, HIGH);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  // accelerate from zero to maximum speed

 for (int i = 0; i < 256; i++) {
```

```
analogWrite(enA, i);
analogWrite(enB, i);
delay(20);
}
// decelerate from maximum speed to zero
for (int i = 255; i > 0; --i)
  {
    analogWrite(enA, i);
    analogWrite(enB, i);
    delay(20);
  }


  // now turn off motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
*/

void loop()                     // run over and over again
{
  getFingerprintID();
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  delay(1000);

}

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
```

```
    Serial.println("No finger detected");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
```

```
  if (p == FINGERPRINT_OK) {
   Serial.println("Found a print match!");
   demoOne();
   delay(1000);

  /* demoThree();
   delay(1000);*/
   demoFour();
   delay(500);
   demoTwo();
   delay(1000);



  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);

  return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();


  if (p != FINGERPRINT_OK)  return -1;
```

```
    p = finger.image2Tz();
    if (p != FINGERPRINT_OK)  return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK)  return -1;

    // found a match!
     demoOne();
     delay(1000);
     /*demoThree();
     delay(1000);*/
     demoFour();
     delay(500);
     demoTwo();
     delay(1000);


    Serial.print("Found ID #"); Serial.print(finger.fingerID);
    Serial.print(" with confidence of "); Serial.println(finger.confidence);
    return finger.fingerID;
}
```

CHAPTER 5

RESULT AND ANALYSIS

A fingerprint vehicle starter's primary aim is to ensure that only approved personnel can start and operate the vehicle, underscoring the need for a secure system that is not easily bypassed or vulnerable to hacking.

Our system has a high degree of reliability and consistency, achieving an 85% accuracy. However, the presence of false data within the system can potentially lead to situations where authorized individuals are denied access to the vehicle or unauthorized individuals are able to gain access. This highlights the importance of ensuring the accuracy and integrity of the data that the system processes.

This project is designed with the user in mind and provides a simple and straightforward experience. The process of scanning a fingerprint to initiate the vehicle is effortless and instinctive**.**

CHAPTER 6

CONCLUSION

The main motive for implementing the fingerprint sensor for the two-wheelers is to provide security for vehicles. It enhances the level of security for vehicles. As the unique finger impression is a promising biometric design for recognition it is used in case of both security and usability. This technology easily reduces theft cases.

This thesis mainly focuses on the ignition of vehicles using sensors, which would provide ease to users in different circumstances, such as in case they forget the keys inside the vehicle or at the other current place. The use of fingerprint sensors provides authentication to valid and registered users. There are many improvements or functionalities that could be added to the current version of this system to make it more efficient in terms of security and portability. The vehicle ignition is highly affected in case the registered user's finger is defaced or defected or colored, the system won't allow the user to ignite the vehicle. It improves the existing system by reducing the cost of the existing and overcoming the conventional system. As an application of it, we can implement the same basic concept in other domains also, which requires more security from thefts such as authorized user entry only, and verified users access in unauthorized regions only.

CHAPTER 7

FUTURE SCOPE

To address any potential issues with fingerprint recognition, our system can be expanded to include keypad password options as an alternative method for starting the vehicle.

Furthermore, we have incorporated an additional layer of security by implementing iris scanning technology. Once the iris of the authorized user has been scanned, access to the vehicle is granted, significantly enhancing the overall security of the system.

With these added features, our project offers enhanced security measures and provides users with multiple for accessing their vehicles, ensuring a seamless and secure experience every time.

CHAPTER 8

BIBLIOGRAPHY

- http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6100565&queryText=finger_print%20car&newsearch=true

- http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4218432&queryText=finger print%20car&newsearch=true

- R. Raman, S. Valarmathy, Dr. N. SuthanthiraVanitha, S.Selvarju, M. Thiruppathi, R. Thangam, "Vehicle Tracking and Locking system based on GSM and GPS", I. J. Intelligent Systems and Applications,86-93, August 2013.

- KUMAR, M. V. & KUMAR, S. R. 2014. Fingerprint-Based Licensing System for Driving.

- MUDHOLKAR, S. S., SHENDE, P. M. & SARODE,M. V. 2012. Biometrics Authentication Technique For Intrusion Detection Systems Using Fingerprint Recognition. *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)* Vol.2, No.1, February

- PINGAT, S., RAKHECHA, S., AGRAWAL, R. MHETRE, S. & RAUSHAN, P. 2013. Real Time Smart Car Security System by Using Biometrics. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*