AUTOMATIC BOOK SCANNER

Group Members

Zumar Shahnawaz	**(190558)**
Hira Younus	**(190528)**
Zersheena Gul	**(190394)**

BACHELOR OF ELECTRICAL ENGINEERING

(2019-2023)

Project Supervisor

DR. MUMAJJED UL MUDASSIR

Assistant Professor

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

A Final Year Project Report
presented to
AIR University
in partial fulfillment of the requirements
for the Degree of

Bachelor of Electrical Engineering

AUTOMATIC BOOK SCANNER

SUBMITTED BY

Zumar Shahnawaz	**(190558)**
Hira Younus	**(190528)**
Zersheena Gul	**(190394)**

(2019-2023)

Department Of Electrical and Computer Engineering

Date: 24-05-2023

Approval For Submission

It is to certify that the project report titled
AUTOMATIC BOOK SCANNER
has met the required standard for submission
in partial fulfillment of the requirements
for the award of degree of
Bachelor of Electrical Engineering
at Air University, Islamabad

Project Supervisor

____

Dr. Mumajjed ul Mudassir
(Assistant Professor)

Head OF Department

____

Dr.Hafiz Aashiq Hussain

Acknowledgement

FIRST and most of all we would like to thank our supervisor for his guidance, assistance and patience throughout the Project. His constant support, suggestions and encouragement made all this possible. Without his help this would not have been possible. We would like to extend our sincere gratitude to our faculty and teaching staff and lastly our families for their constant support.

Abstract

THIS report was aimed to explain our project of creating a prototype for the Automatic Book Scanner. A book scanner is a device that could scan any book or document page by page and converted it into a digital formed. A regular book scanner user needed to flip each page manually, which was hectic and time-consuming. Our primary goal is to develop a method of automatic page-turned that would work without human intervention. In today's world, kept books and documents in hard formed had become hectic in libraries and other offices because of less storage space and large data. So kept them in digital form had became a necessity. The technical challenged was to construct an automatic, easy-to-use, cost-efficient, and good-quality scanning system which could easily scan any document without human effort. We chose a robotic arm and vacuum suction method for the Automatic page-turned mechanism. Our design consists of the robotic armed, vacuum suction tube, and flipper. We have built a prototype that is able to scan the book and convert it into text form and save it in a file. We interfaced an external camera with the Arduino for the conversion process. Also the user has the ease of selecting the desired number of pages he/she wants to scan. The pictures taken from a book processed and converted to text with the Tesseract OCR. The processed images of the pages are combined and converted into an e-book after the desired number of pages is scanned. The entire process is automatic including an automatic page-turning mechanism with the hardware implementation.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The 21st century is characterized by technological advancements and a digital revolution that has significantly impacted our lifestyles and professional endeavors. The practice of digitizing books and documents has become increasingly prevalent, particularly in libraries and large-scale offices with extensive data storage needs. This shift towards digitization offers several advantages, including cost-effectiveness, increased accessibility, and space efficiency. Traditional libraries often face limitations in terms of user access, physical storage constraints, and the retrieval of specific information. By storing information in digital formats, these challenges can be overcome. Digital documents require minimal physical space, are protected from physical damage, and offer enhanced search ability. In the contemporary digital landscape, the majority of documents are stored in digital form. However, a challenge arises when dealing with older books and documents that exist only in printed copies, necessitating their conversion into digital formats. Digitized books provide improved security and reliability. To accomplish this conversion, book scanners are employed, enabling the scanning of individual document pages and their subsequent transformation into digital files. The resulting scanned books can be accessed and read on various digital devices.

1.2 Motivation

Nearly 4 billion trees worldwide were cut down each year for paper, representing about 35 percent of all harvested trees. It had a terrible impact on our environment. To solve this problem world moved toward digitization from hardcover documents. As we saw from this, a massive numbered of pages produced each year. Indeed it needed much physical space, which was limited where as when we saw a digital world, we could easily stored thousands of booked and documents on a 2 cm SD card. Our Libraries are usually limited by the numbered of people who could access them, storage space, and the challenged of obtained the specific information requested by the user. Stored information in digital formed can solve these problems. Booked scanning is a process of converted hardcover books to digital formed. Moreover, manually scanning each page one by one is hectic and time-consuming. So an automatic book scanner had became the needed of almost all institutes and offices, so it would have been a handy tool to solve these problems.

1.3 Problem Statement

Nearly 4 billion trees worldwide were cut down each year for paper production. Still, booked and documents limited to access for a numbered of people.Neither are they safe nor reliable for users to carried hardcover booked in their pocket. They consume large storage space as well. That was why we needed a scanning system that would converted these hardcover documents to digital formed. However, in a regular book scanner user needed to manually flipped each page to scan, which was hectic and time-consuming. We need an automatic book scanner that would automatically scan the documents and converted them into texted files(digital form). Many commercial companies tried to solve the problem and had built book scanners that could automatically convert a given paper book to a digital formed. The price of a document scanner presently available in the market is too high, started from 700 to a few thousand US Dollars. So there is a need to made a cost-efficient book scanner.

1.4 Objectives

To develop a prototype that

1. Can automatically scan books without human intervention.

2. Can convert images into text file.

3. Should be cost efficient.

Chapter 2

Background and Literature Review

Our main focus was to select the best method for the page-turning mechanism for our project. We have done this task by studying the previously tested methods and reviewing them by their specifications and accuracy.

2.1 Page Turning Mechanism Using Wheels

Our primary objective was to identify the optimal page-turning mechanism for our project. To accomplish this, we conducted a thorough analysis of previously tested methods, carefully evaluating their specifications and accuracy. One method that we extensively studied is the wheel-based page-turning mechanism. Considering our project's emphasis on cost-effectiveness, we focused on implementing a hardware solution utilizing readily accessible components at an affordable price.

2.1.1 Construction

1. They constructed a box-shaped framework using corrugated sheets and attached a stand to it.

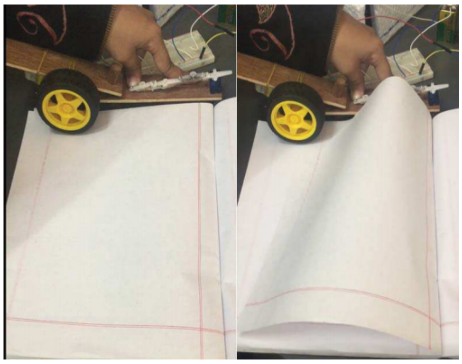
2. Within the box, they securely positioned the wires and Raspberry Pi.

3. A webcam with adjustable settings, boasting a 1.0-megapixel resolution, was affixed to the stand to capture images of the book.

To facilitate the automated page-turning mechanism, they employed a slender plastic board structure featuring a small wheel powered by a DC motor. Additionally, a servo motor with an extended head and a motor driver were utilized for motor control, which was managed by the Raspberry Pi.

2.1.2 Page Turner construction and working

Two rectangular plastic boards are found in the page-turner of the book scanner. The one board has a servo motor with an expanded head and is fixed to a mobile construction. The structure can be shifted in a horizontal manner to accommodate different book lengths. The second board can be moved vertically upward or downward and is positioned on top of the previous one. It can be altered to fit the height of the book. To turn a page, a small wheel that is fixed to this component is turned by the page-turner initially in the opposite direction from the way that the page is bound. As a result, the front page moves slightly upward before being successfully turned by the servo head pushing the page away.



The camera was used to take an initial image. The OCR software is then used to turn the image into a text file. The page-turning mechanism, which is seen in figure, is then used to turn the page. The process is repeated until all required number of pages got scanned. at the end the processed images are converted into a single digital document.

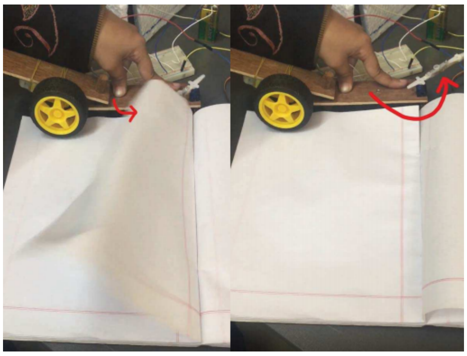


Figure 2.1: Page turning Using Wheels

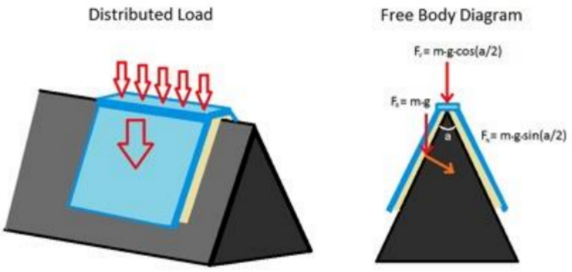
2.1.3 Advantages

1. It was easy to build.
2. It was reasonably priced.

2.1.4 Drawback

One significant limitation of this prototype was occasional page skipping. Instead of turning a single page, there were instances when the wheel movement resulted in turning 2 or 3 pages at once, leading to errors in the process.

2.2 Linear Book Scanner



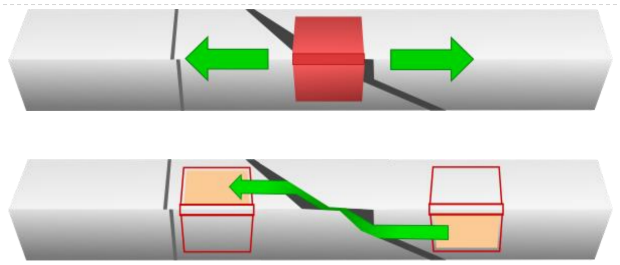


Figure 2.2: Linear Book Scanner

A book scanning machine was made using a variety of technologies and techniques. The way these devices turn the page of a book or flip through it is a key distinction between them. The linear book scanner was one of the methods. There have been several prototypes constructed before, but the most current or fresh one is far better. A saddle, a motor, a hoover and cameras are all part of it. The machine's base is shaped like an inverted V, and its top has a small opening that allows a book page to fall into it. First, the user places the book on the saddle with the cover facing down. The saddle will then be moved to the other side of the body by the motor. It will grab one book page and stuff it into the hole using the hoover. Therefore, that page will be detached from the book when the saddle starts to move again to return to its original position. There are cameras inside the body that will take pictures for the book.

2.2.1 Working

To make this procedure better, a lot of study was conducted. To complete all of the analysis for future work, a probe was carried out. There were various characteristics that must be present in a scanner.

a) Page Turning:

The primary objective was to achieve a damage-free page-flipping process with cost-effectiveness as a crucial factor. Improving the page-turning mechanism was of utmost importance for this scanner. To determine and finalize the overall geometry of the page-turning machine, an engineering analysis was conducted. The width of the book plays a vital role in selecting the appropriate materials for constructing the scanner. The size of the channel or hole in the upper part of the scanner body, through which the book pages will slide, was determined. In designing the interior flap, materials were carefully chosen to minimize friction and avoid potential damage to the pages. Additionally, the top of the slit was designed with a curved shape to optimize the page movement speed and further safeguard the pages.

b) Page Separation:

The second crucial aspect of the design involves effectively separating the pages without skipping any. Vacuum suction proves to be one of the most efficient methods for achieving this. To estimate the required power for page separation, calculations were conducted to determine the force exerted by the vacuum pump, taking into account the contact area between the pages and the pump. This estimation was then compared with the weight of a single piece of paper. Based on these calculations, it was concluded that a standard vacuum pump possesses sufficient power to drive the book scanner's motors and successfully flip the pages. Managing friction between the pages and the scanner's body was another significant consideration. Various calculations were performed using different book types and materials to address this concern. The analysis revealed that utilizing slippery plastic, such as Delrin, resulted in reduced friction compared to aluminum. Delrin, known for its cost-effectiveness and environmental friendliness, was combined with vinyl for the inner flap construction. Furthermore, it was determined that an apex angle of 60 degrees, as opposed to 90 degrees, would exert more force on the book's spine rather than the pages themselves, optimizing the separation process.

c) Error Detection:

Making a system capable of mistake detection was the third most crucial criterion. The idea behind this component was to include a sensor that can determine whether the machine turns more than one page or not. When a page enters a channel, the sensor's role is to alert the machine by sending signals. The object detection uses a proximity sensor. As a result, the error detection method used to send signals when a page enters a channel is the proximity sensor. After that, the capacitive sensor is employed to determine how many pages have been turned. The voltage between a pair of parallel plates is altered as the page enters between the plates.

d) Final Prototype:

Therefore, the machine consists of two main components: the saddle and the body. The body was built of the plastic delrin, and the top of the body is curved. 60 degrees is the apex angle. The page or hoover will grab the page from this side of the machine and move it to the other side, hence one area of the body is known as the page intake channel. On the channel intake side, the vacuum pump is attached. The machine's edges are all smooth to prevent harm to the book. The body panel has the motor fastened to it. It provides the saddle with power so that it may move across the body. The area where we put our book is called the saddle. Through a power belt, power is delivered to the saddle. For error detection, some sensors are employed.

e) Drawback:

Not all book kinds can be scanned by this scanner. Because it was incompatible with a wide range of book sizes and weights. It functions flawlessly with light weight volumes, but it had trouble turning the pages properly and jams with large books. Thus, it was unable to scan the majority of book-sized pages. The sensor's precision is really poor. As a result, it makes page detection difficult.

2.2.2 OCR (Optical character recognition)

OCR(optical character recognition) software is used to convert the image or handwritten documents into the text file. It automatically extract the text from the images without Human help. Now a days this software is very popular specially in the office document work. There are many different applications of this software such as license plate reading, image reading, handwritten document reading. There are many different software available in OCR such as Ocular which is used for the conversion of hand text document. Also Plate-scanner which is used for license plate recognition. In automatic book scanner it need to read the image file and convert it into text file so for this purpose Tesseract is used. It is a OCR engine. It can easily convert the image file. There are two major steps. In the first argument it take the image file and in the second argument it extract the text file from that software and store it in file. It is compatible with different programming language. It is used in such a way that it first recognize the text from a document or it can also contain some external text detector which detect the text from the single line of the image. There are some stages in which it work such as find word, find line and in the last character classification. First it do word finding by placing the text line in sequence with the help of blobs. These text lines were break into different words depending upon the characters and spaces available between them. Then all the words were recognize separately and then it goes to the training data which also recognize words more accurately. After that tesseract commands are used which then write these words in text form and store it.

Chapter 3

System design

3.1 Methodology

There were many different methods through which we can design our automatic book scanner. We already discussed some of the methods in literature review but all those methods had some drawbacks. After all the analysis and research we chose the book scanner with robotic arm and vacuum suction method for our project.

3.1.1 Book Scanner with Robotic Arm and Vacuum Suction

a) Introduction

There were some properties that are very necessary to be present in the automatic book scanner. It should flip the page automatically, it shouldn't damage the pages of the book and also detect error. There were many different methods to implement these properties in the project but the main difference was how they flipped the pages. The flipping technique was chosen on the bases of speed, less cost and should be reliable. Just by looking at the name of the method we got the idea that in our project we are going to use the robotic arm to flip the page with the help of vacuum pump. We also used a flipping rod or slider to slide the page on the other side.

b) Working

There were some main components that are present in our automatic book scanner such as
• Robotic arm with suction pipe.
• Flipping rod.
• Arduino along with camera.
• Cradle.
Our first step was to place the book in the cradle. There was a counter attached and the user will enter the numbers. These numbers are the total number of pages user wanted to scan. Then the robotic arm will move in the downward direction to lift the page. The suction pipe is attached to the robotic arm and it's other end is attached to the vacuum pump. With the help of this vacuum suction the robotic arm will the graph the page of the book. Then the arm will lift the page. The arm will move at the angle of 90 degree which is its initial position and then the flipping rod with flip the page. Flipping rod will rotate at the angle of 180 degree. This flipping rod was used to slide the page. After the page is flipped. Then the camera will take the picture of the page. After taking the picture of the page it will be converted to text. We used OCR (optical character recognition) software to convert the image into the text. If there were more pages left then again the robotic arm will move the page and camera will click the pic but if there are no pages remain then the text will we converted to the PDF and will be store or save to some external memory.

3.1.2 Construction

a) Page Separation

To do a page separation we first needed a robotic arm and a servo motor. One servo motor is linked with the suction pipe so that it will move or lift the page up and down. The other servo motor is used by the flipping rod so that it will flip the lifted page. To make this scanner available for all types of book sizes, we placed the rod and arm in a sliding rail so that it will move. Then it is adjusted according to the size of the book. There is camera attached to the upper part of the scanner that will capture the image. We used Arduino micro controller for our system as we were using the stepper motor.

b) Error Detection

To do error detection and to make the scanner work more efficient sensors are used. Sensors are used so that all the steps are performed in a loop. There are different sensors used in this automatic book scanner. These sensors are used to make sure that rod and robotic arm will flip the pages and doesn't skip the page in any cycle. Sensor are used so that it will detect when the book is placed and send signals to the system. Another sensor is used which send signal to the micro controller so that the camera will capture the image after the page is flipped. If the system skip any one step it will not perform the next step because sensor will not send the signal to the system which make this scanner more efficient.

c) Keypad

There is a Keypad attached to the micro controller which acts as an input. The purpose of this Keypad is to define how many pages user needed to scan. First the user enters the number in the Keypad and then the Keypad send this input to the micro controller and the system scans the specific page number not more than that. With the help of this keypad the user can scan the book from anywhere.

3.1.3 OCR (Optical Character Recognition)

We needed to convert the image into the text file. OCR software is use to convert the images into the text file. We used tesseract for this purpose. Tesseract is an optical character recognition engine. First the image file was provided to the tesseract as an input which is a command base tool after that the tesseract command will use. There are two major steps. First the tesseract command reads and take the image file as an argument and in second step it extracts the text and store it in a file. This tesseract can be used for different languages. It means that it can convert the images of the book into text file without having any problem of language.

3.1.4 Text To Image

In the last step all the text file was needed to be converted into the pdf and to be stored in the external memory. First there was some fixed directory where all this data will store after that imagemagick was used. It converts the text file into the pdf. Then we used the SD card so that it can store all the files. SD Card is used as an output which will store the final file or pdf.

3.2 Block Diagram

The functional block diagram of automatic Book Scanner is shown below:

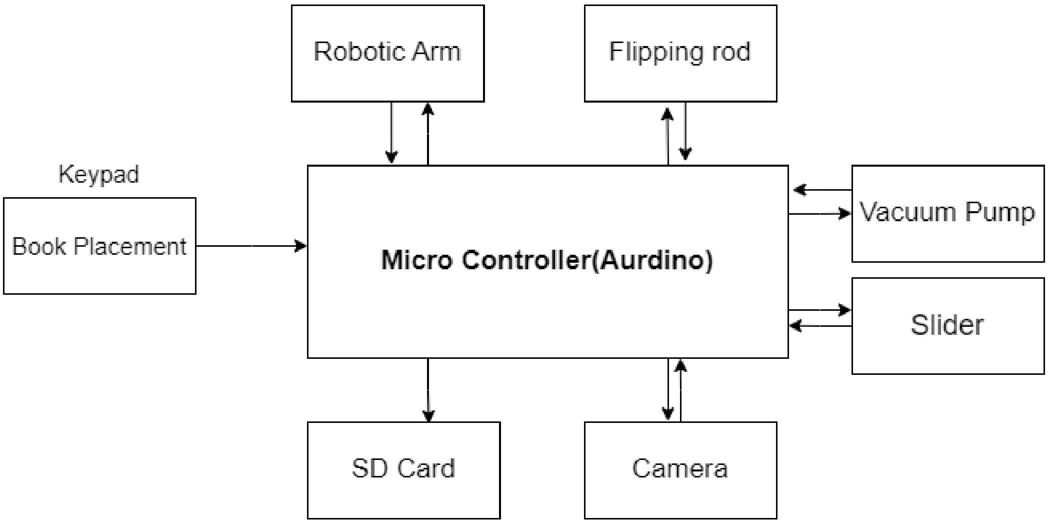


Figure 3.1: Block Diagram

3.3 Process Flowchart

The process flow chart:

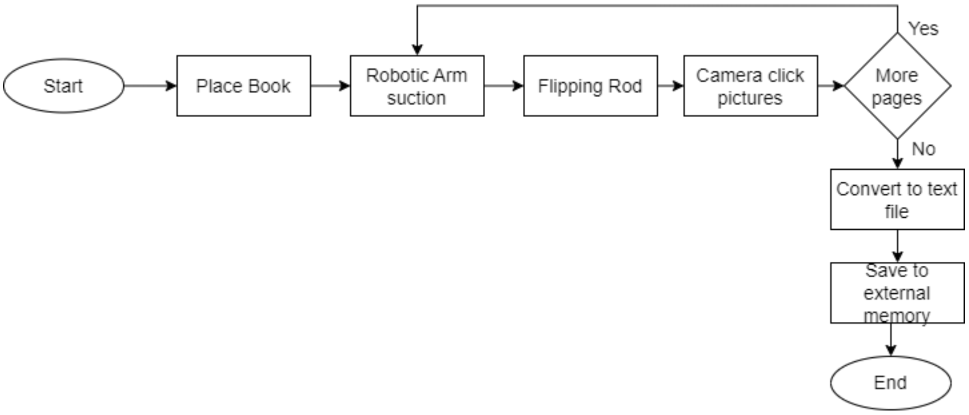


Figure 3.2: Process Flowchart

3.4 Design Procedure

We designed an Automatic Book Scanner. We came through many different methods by research through which we can design an Automatic Book Scanner and chose a method to design our project. Let's discuss why we didn't chose the alternative methods for this project and why we are choosing this method for our project. These were the three methods through which we can design our project of Automatic Book Scanner:

- Page Turning mechanism using wheel
- Linear Book Scanner
- Automatic Book Scanner with Robotic Arm and Vacuum Suction

Alternative methods that were not chosen and why?

3.4.1 Page Turning Mechanism

Page turning mechanism was one of the methods that can be used to make automatic book scanner. It contains two rectangular plastic boards, servo motor and a wheel. A rubber wheel is placed on the page. And as it rotates it pushes the page toward the center. As the page is slightly lifted a stick on a servo is attached that sweeps across and turn the page over. The whole project is repeated this way. But there is a drawback to this method that makes it very inefficient and error full. The drawback is that it can sometimes turn more than one page that means a page is skipped that makes it error full. So we cannot use this method as our main idea of the project is automatically flipping and scanning pages but if there is error in the flipping of the page then it is not very appropriate to use this method.

3.4.2 Linear Book Scanner

Another method of an automatic book scanner was Linear Book Scanner. It contains a horizontal slider and vacuum suction. Book slides horizontally over a slider and at the end point vacuum suction intake and outtake the page in the slider. There are two cameras inside the slider and it takes pictures of each side of the page. But it also has a drawback that it can only pick thick pages. It cannot pick up thin pages. Also two cameras are needed for this method that does not makes it cost efficient hence this method cannot be used for making automatic book scanner.

Method that was chosen and why

3.4.3 Book Scanner with Robotic Arm and Vacuum Suction

Final and the chosen method for making the Automatic Book Scanner was the book scanner that uses a robotic arm with vacuum suction to flip pages automatically without the use of human efforts. It contains a robotic arm with vacuum suction to lift the page and a flipper was used to turn it over. Then a camera was used for taking the picture of the page and then after that the whole process was continued. The reason why we used this process is that it turns one page at a time and can turn all type of pages and can scan all types of books.

3.5 Design Detail

The following was the labeled diagram of our project that gives the overview of what our Project was going to be and also what components we used

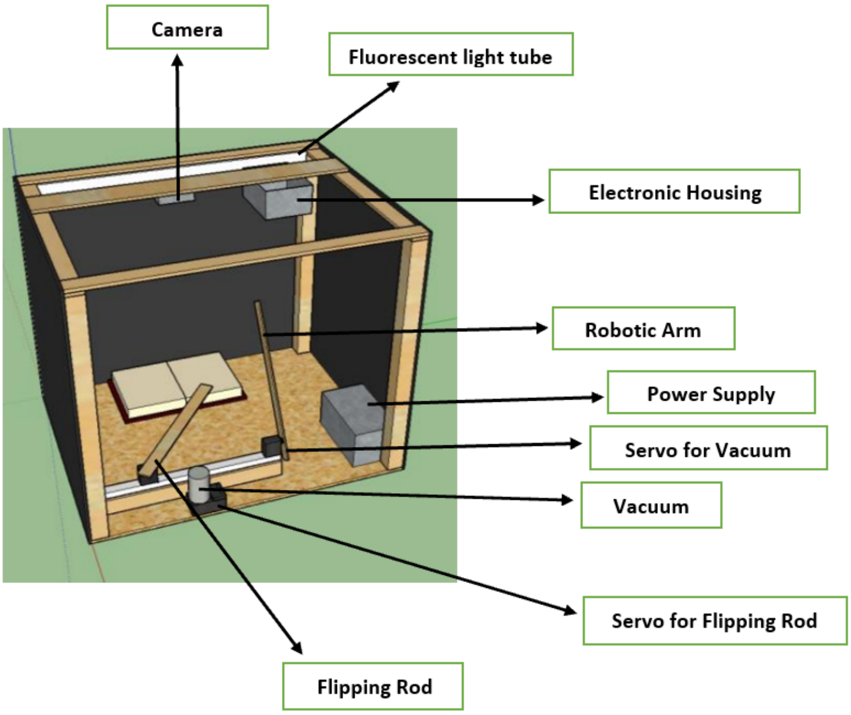


Figure 3.3: Hardware Prototype

3.6 List Of Components

These were the main components that were used in our project.

3.6.1 Arduino

We used an Arduino micro controller. We used Arduino because we were using stepper motors and many sensors that can easily be implemented on Arduino boards. Also it used less power and code had easy access to hardware.

3.6.2 Robotic Arm

A Robotic Arm with Vacuum suction was used for picking the page up. It was made of total 5 servo motors. After the book was placed it will move down and stick to the page due to the vacuum suction and then it will move up with the page attached.

3.6.3 Servo Motors MG66R

Two Servo motors were used in this project. One is for the flipping rod and the other for the vacuum suction.

3.6.4 Vacuum Pump

A vacuum pump kit was used to lift the pages of the book. The kit includes a silicone suction cup, PWM electronic switches, a vacuum pump, an 800mm silicone hose, and a Tee-joint electronic valve.

3.6.5 Slider

A slider was used as it allows us to move our camera back and forth so the camera will take picture of both pages one by one without compromising on quality of image.

3.6.6 Keypad 4x3

A 4x3 keypad was used so that the user can select the desired number of pages he/she wants to scan. By entering a 3 digit number the whole process will start working.

3.6.7 Camera

A Camera was used for clicking pictures of the pages.

3.6.8 Flipping Rod

After the sticking of the page the flipping rod comes into action. It moves horizontally and slides the page and in this way the page is turned..

3.6.9 Power Supply

A Power Supply was needed for all the motors to work. So a power supply is used.

3.6.10 LED Lights

LED lights are used to illuminate the pages so that are pictures are clear even in dim light.

3.6.11 Electronic Housing

Our whole Arduino setup is in a box. So the whole circuitry will not be visible and our project will be more presentable.

3.7 Working Of Components

3.7.1 Arduino Mega 2560

Arduino mega is a microcontroller board that depends on the ATmega2560 microcontroller. It has total of 54 digital input and output pins. The configuration of these pins are:

- 16 pins are analog inputs.
- 14 pins can be used as PWM outputs
- 4 UARTS (Serial ports)
- 16 MHz crystal oscillator
- A USB connection
- A Power Jack
- An ICSP Header
- A reset button.

This board almost includes everything that is essential for supporting the microcontroller. The Power can be supplied by connecting the Arduino to the PC using a cable or an USB or battery or AC DC adaptor.

Arduino Mega Specifications

The specifications of Arduino Mega include the following.

- The ATmega2560 is a microcontroller
- The operating voltage of this microcontroller is 5volts
- The recommended Input Voltage will range from 7volts to 12volts
- The input voltage will range from 6volts to 20volts
- The digital input/output pins are 54 where 15 of these pins will supply PWM o/p.
- Analog Input Pins are 16
- DC Current for each input/output pin is 40 mA
- DC Current used for 3.3V Pin is 50 mA
- Flash Memory like 256 KB where 8 KB of flash memory is used with the help of boot loader
- The static random access memory (SRAM) is 8 KB
- The electrically erasable programmable read-only memory (EEPROM)

is 4 KB

- The clock (CLK) speed is 16 MHz
- The USB host chip used in this is MAX3421E
- The length of this board is 101.52 mm
- The width of this board is 53.3 mm
- The weight of this board is 36 g

Arduino Mega Pin Configuration

The pin configuration of the Arduino mega 2560 board is given below. Each pin on this board has its own specific function. All the analog pins of this board can be used as digital Input/Output pins. The Arduino Mega Board offer Flexible work memory space. When we compare with other types of Arduino boards, these boards are physically superior.

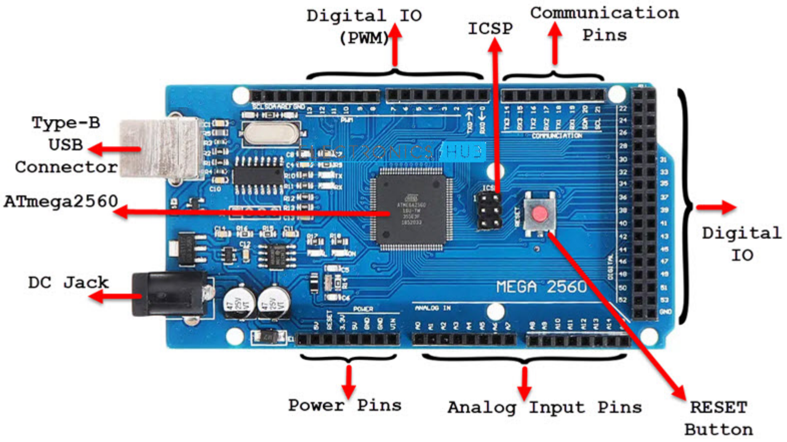
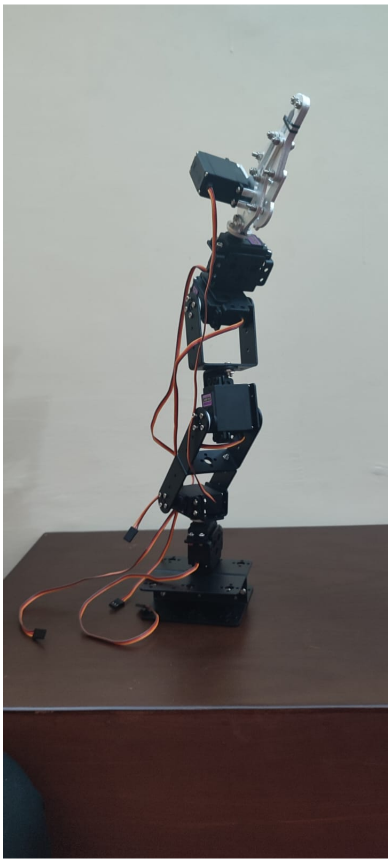


Figure 3.4: Arduino Mega 2560

3.7.2 Robotic Arm

Robotic Arm is a machine that is programmed to execute certain task or job rapidly, efficiently and accurately. In general the robotic arms are motor driven. They are mostly used for fast and consistent performance that involves lifting of heavy things or procedures that are repeated over an extended period of time. They are especially valued in the industrial production, manufacturing, machining and assembly sectors. Robotic arms can perform several tasks for human workers like pick and place, palletizing and other product handling applications that can be dull and injury-inducing. They also can do tasks like welding and removing materials that give rise to fumes, dust and particles that can be hazardous to human beings. The Robotic Arm that we designed in our project is a 4 DOF (degree of freedom) Metallic Robotic Arm and is made of total 4 servo motors. The motors that we used are MG 996R. One servo motor is used for the operation of the Flipping rod. They are assembled together by different screws and nuts, brackets and bearing to make the whole robotic arm.



Now first we'll explain the motor that we used

3.7.3 Servo Motor MG66R

The servo motor MG996R is a metal gear servo motor which has a maximum stall torque of 11 kg/cm. As there are two types of servo motors continuous and standard. The continuous one goes from 0 to 360 degrees and has no stopping point and the standard one goes from 0 to 180 degrees. We are using standard motors so the motor rotates from 0 to 180 degree based on the duty cycle of the PWM wave supplied to its signal pin. This is the general diagram of the servo motor MG996R.

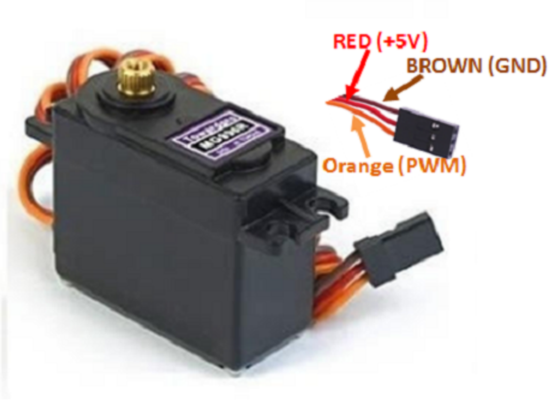
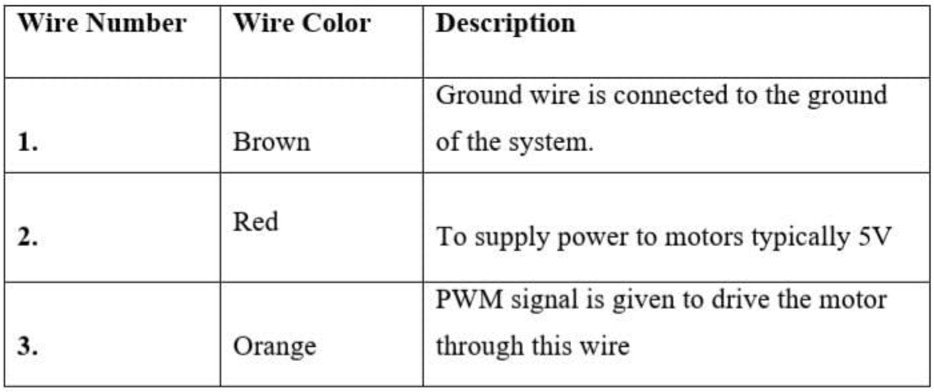


Figure 3.5: Servo Motor used

MG996R SERVO MOTOR FEATURES

- Operating Voltage is +5V typically
- Current: 2.5A (6V)
- Stall Torque: 9.4 kg/cm (at 5V)
- Maximum Stall Torque: 11 kg/cm (6V)
- Operating speed is 0.17 s/60°
- Gear Type: Metal
- Rotation : 0°-180°
- Weight of motor : 55gm
- Package includes gear horns and screws



Wire Number	Wire Color	Description
1.	Brown	Ground wire is connected to the ground of the system.
2.	Red	To supply power to motors typically 5V
3.	Orange	PWM signal is given to drive the motor through this wire

Figure 3.6: Servo Motor Wire Configuration

- Operating Voltage: 4.8V-7.2V

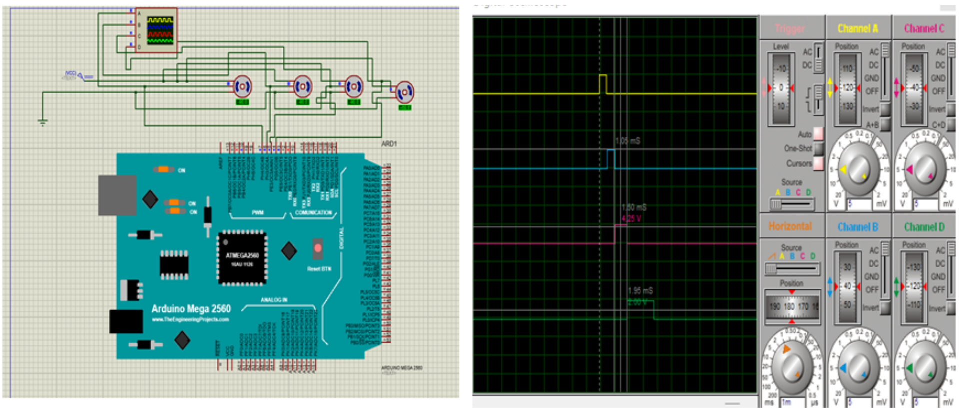


Figure 3.7: Servo Motors Schematics

Servo Motors have 3 wires coming out of it and the wire description above shows what are there uses. In order to make this motor work or we can say rotate, we give a power supply to the motor using the Red wire and brown is connected to the ground. And the orange colour wire receives the PWM signal to drive the motor. PWM signals can be generated from anything like 555 Timer or Microcontroller like Arduino, PIC etc. In our Project we are using Arduino Mega 2560 to generate PWM signals and give it to the motor to make it work. The total servo motors used in our project is five. Four servo motors are used to design the Robotic Arm and one is used to rotate the flipping rod.

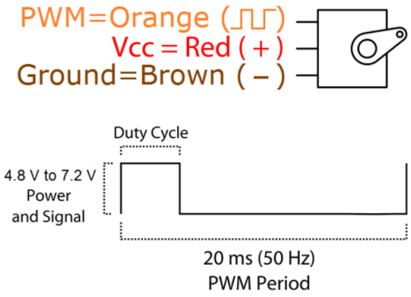


Figure 3.8: PWM of servo Motor

a) Construction of Robotic Arm

Our Robotic Arm consisted of the following parts:
- Multi-functional bracket
- Long U shape bracket
- L shape bracket
- Cup bearing
- Screw nut
- U beam bracket
- Metal servo horn We are using the robotic arm that consists of total four servo motors MG996R. We constructed the robotic arm by screwing the motors, the brackets, bearing and nuts together.

b) Working Of Robotic Arm

In robotic arm we adjusted our servo motors in such a way that all the motors were in the reverse direction of each other. This created a balance in the robotic arm, ensuring equal weight distribution. The servo motors moved in either a positive or negative direction, primarily depending on the gear adjustments we made.
In our robotic arm, servo motors 3 and 4 moved in a negative direction. Initially, these motors were closer to 180 degrees, and we moved them towards 0 degrees to achieve the desired motion. We operated the robotic arm by controlling the servo motors according to our specific requirements.

Our project consisted of four different stages for the robotic arm, which defined its movements. Firstly, we had the initial stage where the robotic arm remained in a steady state. Then, in the next stage, it moved towards the book. The third stage involved moving the arm downward to touch the book's page before returning to the second stage. Subsequently, the flipping rod flipped the page of the book. Finally, in the last stage, the robotic arm moved out of the camera's view, allowing the camera to capture images

Throughout these stages, we provided specific directions to the servo motors, and they rotated accordingly. To ensure optimal performance, we supplied the motors with an AC source of 20V. Since servo motors require a sufficient voltage supply to function properly, the AC source was necessary for precise movements.

This entire process operated in a loop, allowing the robotic arm to transition from one stage to another with user-defined delays.

3.7.4 Vacuum Pump Kit

The Vacuum Pump Kit is a specialized component designed for robotics that requires the use of vacuum suction. The kit includes a silicone suction cup, PWM electronic switches, a vacuum pump, an 800mm silicone hose, and a Tee-joint electronic valve. The vacuum pump is designed to generate a maximum vacuum pressure of 85kPa, which is suitable for a wide range of applications. The pump operates using a DC voltage of 12V and is connected to the servo motors using the 25T spline provided. The silicone suction cup provides the gripping power needed to stick and lift the page. The cup can be easily attached to the vacuum hose provided in the kit. The two PWM electronic switches allow for the precise control of the suction power. They are connected to Arduino to automate the suction operation. The 800mm silicone hose serves as a connection between the suction cups and the vacuum pump. It is flexible and durable, allowing for easy maneuverability during operation. The Tee-joint electronic valve provides an additional level of control over the suction power. It allows for quick and easy adjustments to the air pressure.

a) Assembling of Vacuum Kit

The vacuum pump is connected to a power source, and the suction cups are attached to the end of the vacuum hose. The electronic switches and the Tee-joint electronic valve are then connected to the pump and the servo motors, and the suction power can be controlled using the switches.

b) Working

The entire process involved controlling a pump and a solenoid valve to adhere to and lift the page using suction power. Initially, both the pump and solenoid valve were kept turned off. The process commenced by activating the air pump while keeping the solenoid valve off, generating the necessary suction power to stick to and lift the page. Once the page was lifted, the air pump was turned off, and the solenoid valve was opened, discontinuing the suction power and ultimately releasing the page from the suction cup.

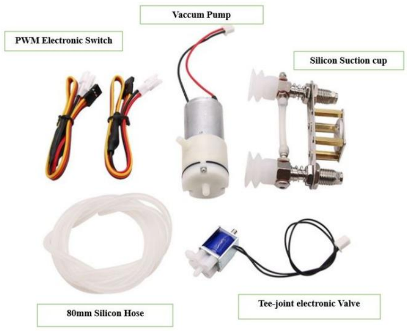


Figure 3.9: Vacuum Pump Kit

3.7.5 Slider

The slider was one of the most significant parts of our project as it allowed us to move our camera back and forth. This enabled the camera to capture pictures of both pages one by one, ensuring that the image quality was not compromised. Additionally, the slider allowed us to complete our work using just one camera instead of needing two different cameras for each page. To design the slider, we utilized a 12V DC stepper motor along with a linear screw nut slider and an H bridge.

a) H-Bridge

We utilized an H-bridge as an electronic circuit to control both the direction and speed of a DC motor. The H-bridge earned its name due to

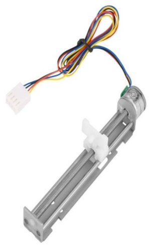


Figure 3.10: Slider

its schematic resemblance to the letter ”H”. This configuration consisted of four switching elements, usually transistors or MOSFETs, arranged in pairs. The DC motor was connected to the central junction of the H-bridge, and the four switches were used to regulate the flow of current through the motor.

By selectively turning the switches on and off in a specific sequence, the H-bridge was able to reverse the direction of the current flowing through the motor, thereby changing the motor’s rotation direction. Additionally, by modifying the pulse width of the signals that controlled the switches, the H-bridge could adjust the motor’s speed.

We employed the L9110S Dual Channel Motor Driver Module (H-Bridge) in our project. This compact module was equipped with two independent L9110S motor driver chips, each capable of driving up to 800 mA of continuous current. The module could be powered from 2.5V to 12V, making it compatible with both 3.3V and 5V microcontrollers. To connect the module to a microcontroller, we utilized a set of male header pins. The module provided two sets of screw terminals for attaching motors.

We regulated the module’s speed using a PWM (Pulse Width Modulation) signal, while the direction could be changed using a digital output. Additionally, the module had the capability to drive a single four-line, two-phase stepper motor. For easy attachment to a robot or other projects, the module included four mounting holes.

Specification

- Motor Voltage: 2.5 12V
- Motor channels: 2
- Max Continuous Current per Channel: 800 mA
- Size: 31mm x 22mm x 12mm

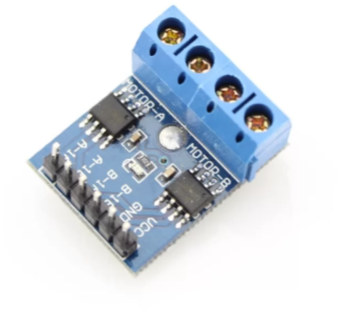


Figure 3.11: H-Bridge

b) Working and Interfacing with Slider:

The motor was connected between the central junctions of the H-bridge in our completed project, and the four switches were utilized to control the flow of current through the motor. The H-bridge played a crucial role in changing the direction of the current flowing through the 12V DC motor to control the linear screw of the slider. By selectively turning on and off the switches in a specific sequence, the H-bridge altered the motor's direction of rotation, effectively controlling the linear motion of the slider.

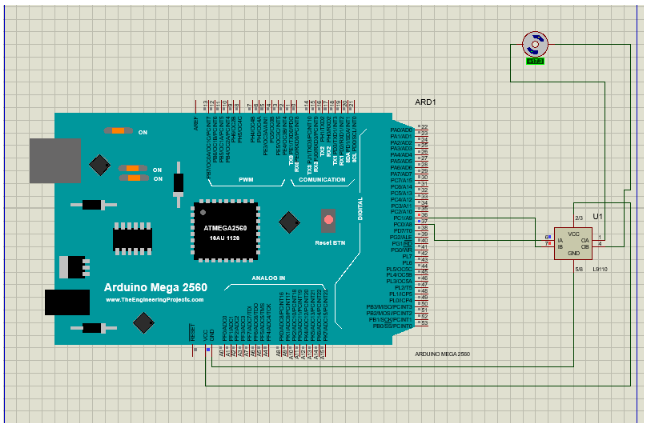


Figure 3.12: H-Bridge Schematics

12V DC Stepper Motor with Linear Screw Nut Slider:

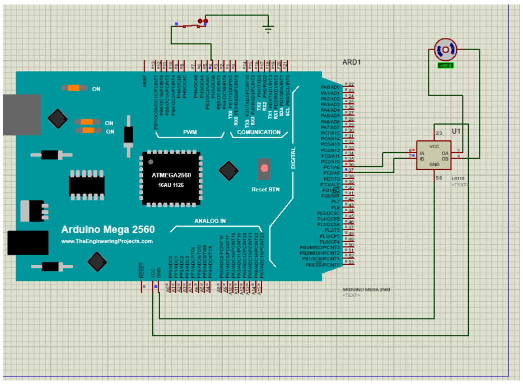
It is widely used for small devices such as medical equipment, laser engraving machine, etc. The motor needs a A4988 driver to drive, and cannot directly connect to power. The maximum linear speed is 25mm per second. This slider consists of 3 main parts

- Linear screw
- 12 V DC motor
- Limiter (Linear limit switch)

The 12V DC motor is attached to the linear screw. This motor will be responsible for driving the screw to move the camera along the rod.

A linear screw, also known as a lead screw, is a type of mechanical component used for converting rotational motion into linear motion. The linear screw is typically composed of a shaft, and a matching slider that moves along the length of the rod. The main use of a linear screw is to provide precise and smooth linear motion in a wide range of applications. They are commonly used in machinery and automation systems for linear actuation, such as moving a camera slider or positioning a robotic arm. One of the advantages of using a linear screw is that they can provide a high degree of accuracy and repeatability in positioning. The lead screw has a precise slider moves smoothly along the length of the rod, resulting in a linear motion that is consistent and repeatable.

The limit switch acts as a safety feature to prevent the motor from continuing to run when the camera has reached the required position on the rod. This prevents the motor from getting damaged due to overloading.



c) Assembling and Working

We connected the motor and limit switch to an Arduino board by using a motor driver. The limit switch was connected to a digital pin on

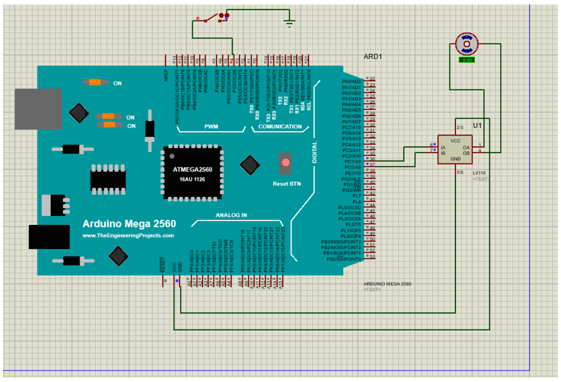
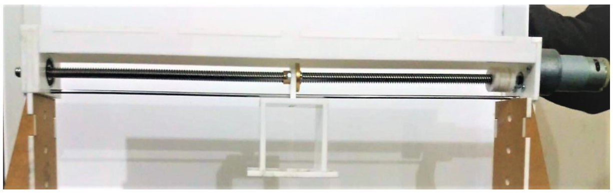


Figure 3.13: Limiter Switch

the Arduino so that its state could be monitored. By doing so, we were able to control the motor and read the status of the limit switch using the Arduino. The Arduino board was programmed to detect when the camera had reached its intended position on the rod by monitoring the state of the limit switch. When the limit switch was triggered, the controller stopped the motor to prevent the camera from moving any further, ensuring that it remained in the desired position.



3.7.6 flipping Rod

The flipping rod is then attached to the linear screw. This will act as a guide for the screw to move the camera along the rod.

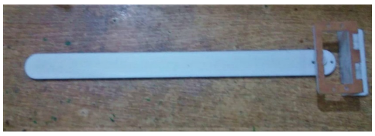


Figure 3.14: Flipping Rod

3.7.7 Keypad 4x3

Keypads are widely used in electronics. A 4x3 Keypad have a total number of 12 buttons; Numbers from 0 to 9, a asterisk (*) and a Hash (#). We have used a 4x3 Keypad in our project.

Figure 3.15: Keypad 4x3

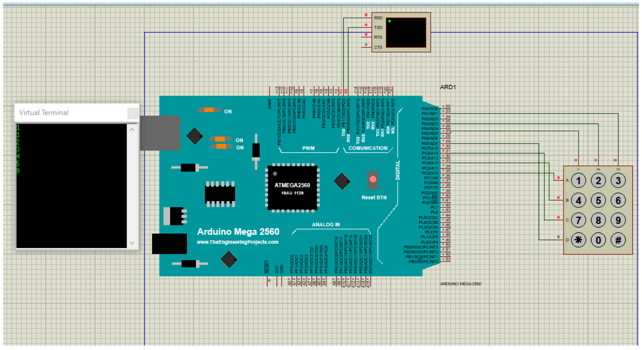
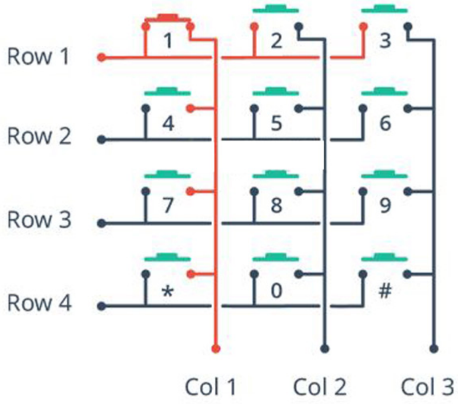


Figure 3.16: Keypad Schematics

There are 4 rows and three columns. Each button is in form of switch. And one side of each switch is connected to a row and the other side to a column. If we press switch 2, the input of this row is saved at the output of its column. We interfaced the keypad with Arduino using Pin numbers 35, 33, 31, 29,27, 25, 23. We are using keypad in our project so that the user can select the number of pages he/she wants to scan.



3.7.8 Web Cam

A webcam is an input device that captures digital images. These are transferred to the computer, which moves them to a server. From there, they can be transmitted to the hosting page. Laptops and desktops are often equipped with a webcam. They are commonly connected to a device using USB or wireless protocols. the systems used in webcams have a sufficiently large depth of field that the use of a fixed-focus lens does not impact image sharpness to a great extent. webcam contains some built-in image processing, separate from compression. Support electronics read the image from the sensor and transmit it to the host computer. The camera pictured to the right to transmit its image over USB.

a) Optocoupler

Optocouplers that are also called photo- couplers or opto-isolators consist of two components in single package. One is the LED and a photosensitive device that is used to optically couple an electrical signal without an electrical connection. Optocoupler is uni-directional means that it can pass signal in one direction only. Optocouplers prevent the circuits receiving the signal from high voltages. There are many different types of optocoupler but we used the PC817.
To trigger the webcam if it has physical button then it is better to trigger it from Arduino via a photocoupler. The photocouplers or optocouplers are typically contained in a single package, often about the size of an integrated circuit, although there is a large degree of variation according to

the intended application.

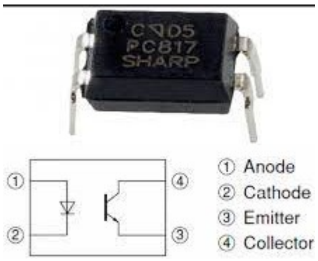


Figure 3.17: Optocoupler

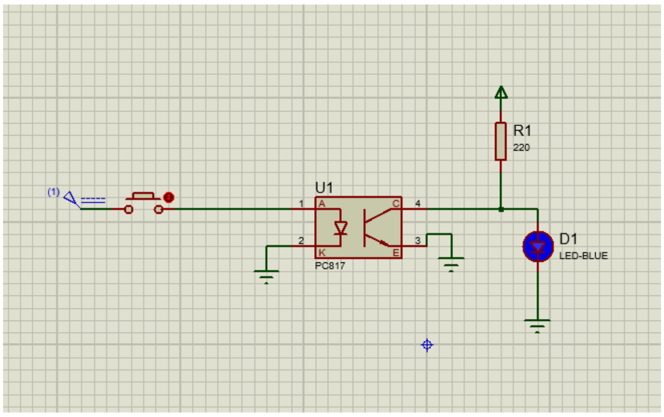


Figure 3.18: Optocoupler Schematics

b) PC817

PC817 is an optocoupler that consists of infrared LED and a phototransistor. This optocoupler works in this way that first the current is applied to the optocoupler which turns on the LED. The LED then generates infrared radiation or light that is proportional to the current flowing through it. After that this light strikes on the phototransistor which conducts current. Basically, this device works as a switch. As soon as the current stops flowing, the LED turns off and the phototransistor also stops conducting and it is turned off. Also, at the input side of the optocoupler, a 1k ohms resistor is connected in series to the anode to limit the current flowing across the IR LED.

In our project we used C120 webcam to capture images and we used photocoupler to trigger the webcam. In an OptoCoupler/Photo Transistor

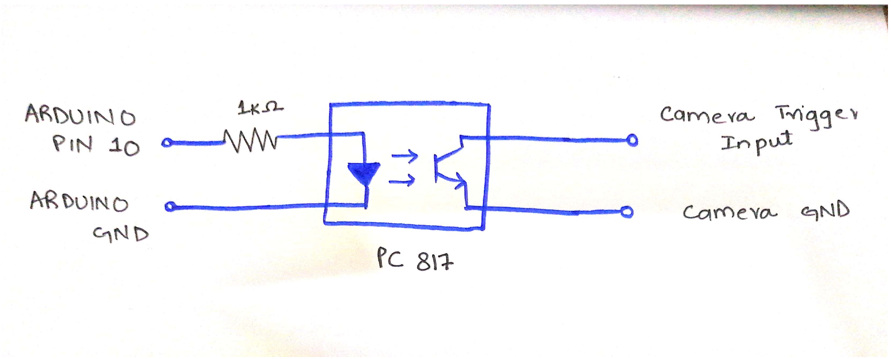
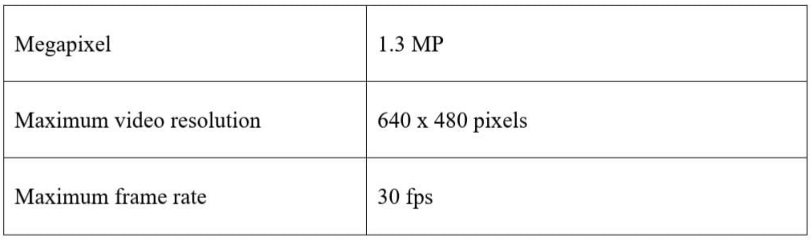


Figure 3.19: Camera Trigger Schematics

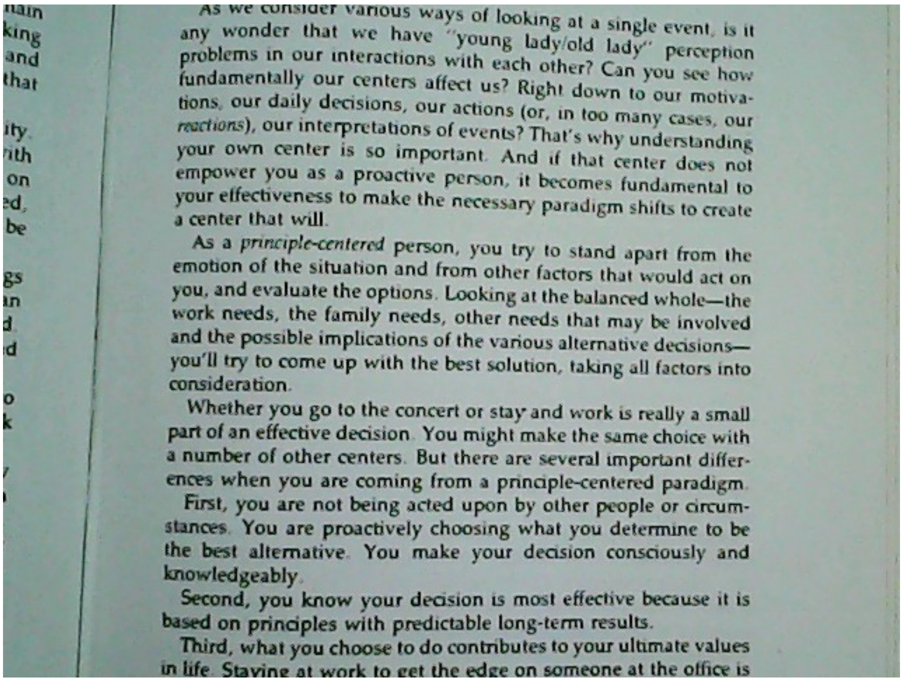
an LED triggers a photo sensitive transistor.

Figure 3.20: Camera used

Camera Specification



Megapixel	1.3 MP
Maximum video resolution	640 x 480 pixels
Maximum frame rate	30 fps



As we consider various ways of looking at a single event, is it any wonder that we have "young lady/old lady" perception problems in our interactions with each other? Can you see how fundamentally our centers affect us? Right down to our motivations, our daily decisions, our actions (or, in too many cases, our *reactions*), our interpretations of events? That's why understanding your own center is so important. And if that center does not empower you as a proactive person, it becomes fundamental to your effectiveness to make the necessary paradigm shifts to create a center that will.

As a *principle-centered* person, you try to stand apart from the emotion of the situation and from other factors that would act on you, and evaluate the options. Looking at the balanced whole—the work needs, the family needs, other needs that may be involved and the possible implications of the various alternative decisions—you'll try to come up with the best solution, taking all factors into consideration.

Whether you go to the concert or stay and work is really a small part of an effective decision. You might make the same choice with a number of other centers. But there are several important differences when you are coming from a principle-centered paradigm.

First, you are not being acted upon by other people or circumstances. You are proactively choosing what you determine to be the best alternative. You make your decision consciously and knowledgeably.

Second, you know your decision is most effective because it is based on principles with predictable long-term results.

Third, what you choose to do contributes to your ultimate values in life. Staying at work to get the edge on someone at the office is

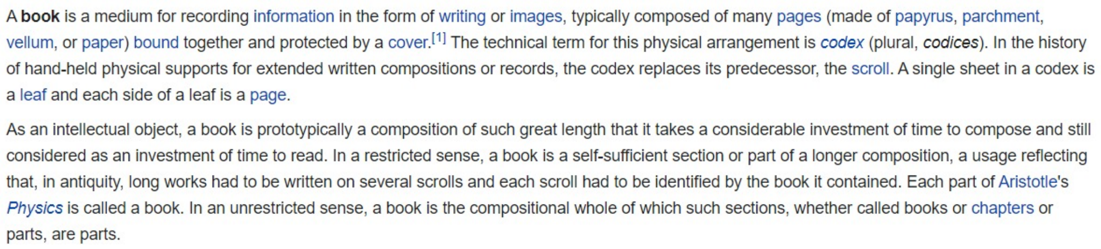
Figure 3.21: Camera Result

3.7.9 Image To Text

One of our tasks was to automatically convert images into text files. Although there are various software options available to perform this function, they typically require user intervention and are not automatic. To address this problem, we incorporated Tesseract, which automatically converts images into text format. We had two primary objectives. Firstly, the program needed to automatically retrieve an image from a stored file, convert it into text, and save it to another file. These functions were designed to run continuously in a loop without any human assistance. For this purpose, we utilized Visual Studio Code (VS Code), a streamlined code editor that supports development operations such as debugging and task running.

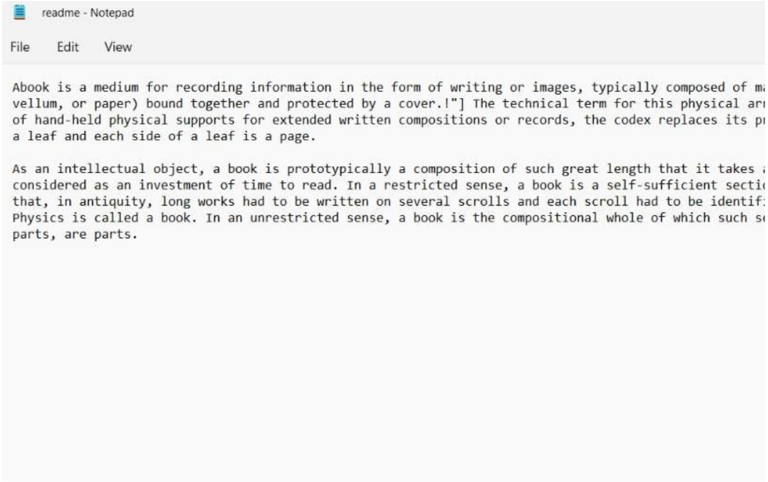
In our project, we used Python programming language and the Tesseract OCR library to perform the image-to-text conversion. We also incorporated the PIL (Python Imaging Library) library, which enhanced image processing capabilities in Python. Image processing involves applying various operations on images to extract useful data or enhance the image itself. In image processing, images are typically treated as 2D signals while applying different processing methods.

Tesseract, the library we employed, utilizes a neural network called a line recognizer. It excels at detecting text within large documents, processing one line at a time in a downward direction. It can also recognize text from images containing a single text line. Additionally, Tesseract employs character recognition patterns to identify specific characters within images. To implement the functionality, we provided paths to the Tesseract OCR and image files. This allowed the program to locate the specified image. It then opened the file, read its contents, extracted the text, and finally stored the extracted text in another file. This flexibility allowed us to save the file anywhere and specify any desired path for the input image.



A **book** is a medium for recording information in the form of writing or images, typically composed of many pages (made of papyrus, parchment, vellum, or paper) bound together and protected by a cover.[1] The technical term for this physical arrangement is *codex* (plural, *codices*). In the history of hand-held physical supports for extended written compositions or records, the codex replaces its predecessor, the scroll. A single sheet in a codex is a leaf and each side of a leaf is a page.

As an intellectual object, a book is prototypically a composition of such great length that it takes a considerable investment of time to compose and still considered as an investment of time to read. In a restricted sense, a book is a self-sufficient section or part of a longer composition, a usage reflecting that, in antiquity, long works had to be written on several scrolls and each scroll had to be identified by the book it contained. Each part of Aristotle's *Physics* is called a book. In an unrestricted sense, a book is the compositional whole of which such sections, whether called books or chapters or parts, are parts.



```
Abook is a medium for recording information in the form of writing or images, typically composed of m
vellum, or paper) bound together and protected by a cover.!"] The technical term for this physical ar
of hand-held physical supports for extended written compositions or records, the codex replaces its p
a leaf and each side of a leaf is a page.

As an intellectual object, a book is prototypically a composition of such great length that it takes
considered as an investment of time to read. In a restricted sense, a book is a self-sufficient secti
that, in antiquity, long works had to be written on several scrolls and each scroll had to be identif
Physics is called a book. In an unrestricted sense, a book is the compositional whole of which such s
parts, are parts.
```

Figure 3.22: Text file

This the a image that was converted to the text form. You can easily see in the next picture that the image is converted to text.

3.7.10 GUI (Graphical User Interface)

GUI, or graphic user interface, stands for. It is a sort of user interface that enables users to interact with electronic devices or software using graphical elements rather than text-based commands, like icons, buttons, windows, and menus. GUI's offer visual representations of tasks and options to make it simpler for users to deal with complex systems. They frequently appear in operating systems, programmers, and websites to offer a simple and clear user interface. GUI's typically include various graphical components such as windows, buttons, check boxes, radio buttons. In our project, we created a GUI using Python. The GUI featured a start and stop button, allowing the user to initiate or halt the processing. Additionally, the GUI provided visual feedback indicating that the pictures were being processed.

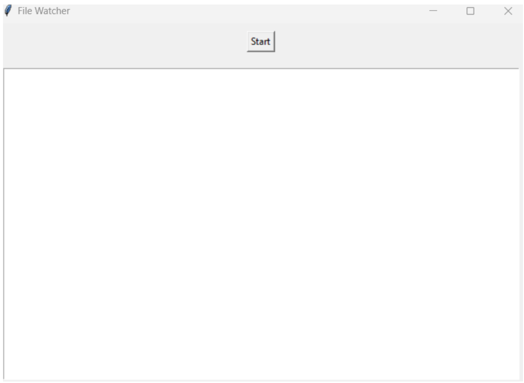
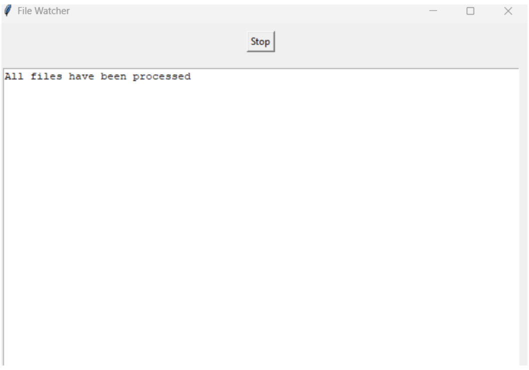


Figure 3.23: GUI Start Button

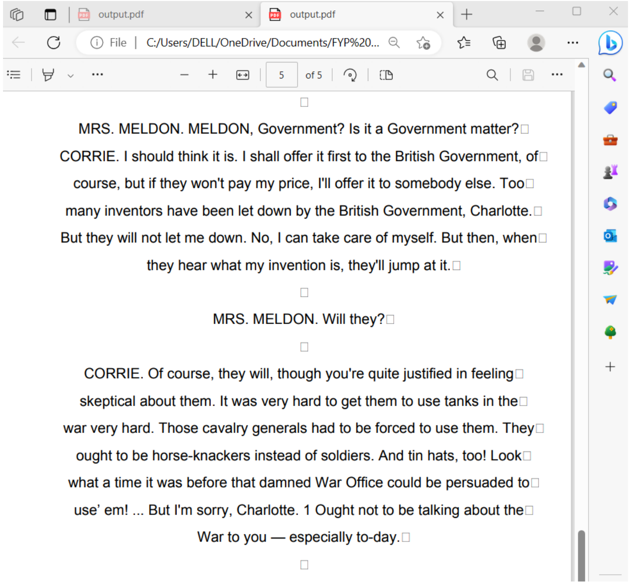


```
All files have been processed
```

Figure 3.24: GUI Stop Button

3.7.11 PDF

A PDF (Portable Document Format) is a file format used to represent documents in a manner that is independent of application software, hardware, and operating systems. Text, photos, vector graphics, hyperlinks, forms, and interactive components can all be found in PDF files. They are made to ensure that a document seems the same to all users by maintaining the formatting, layout, and fonts of the content across various devices and operating systems. Many other uses for PDF exist, including: Document sharing: Since they can be opened and viewed on practically any device with a PDF reader, PDF are frequently used to share documents with others. Printing: PDF are frequently used for high-quality printing because they can maintain a document's layout and typefaces, guaranteeing that it prints reliably on various printers. E-books: Since many e-books are distributed in PDF format, users can view and browse the book on a variety of gadgets. PDF are used for document archiving because they can hold a lot of data in a small, self-contained package. The second major part of our project was to form a PDF. We utilized a Python library for PDF processing to generate the PDF file after extracting the text. This PDF served as the final output of our entire project, providing the user with the text file in a PDF format.



□

MRS. MELDON. MELDON, Government? Is it a Government matter?□
CORRIE. I should think it is. I shall offer it first to the British Government, of□
course, but if they won't pay my price, I'll offer it to somebody else. Too□
many inventors have been let down by the British Government, Charlotte.□
But they will not let me down. No, I can take care of myself. But then, when□
they hear what my invention is, they'll jump at it.□

□

MRS. MELDON. Will they?□

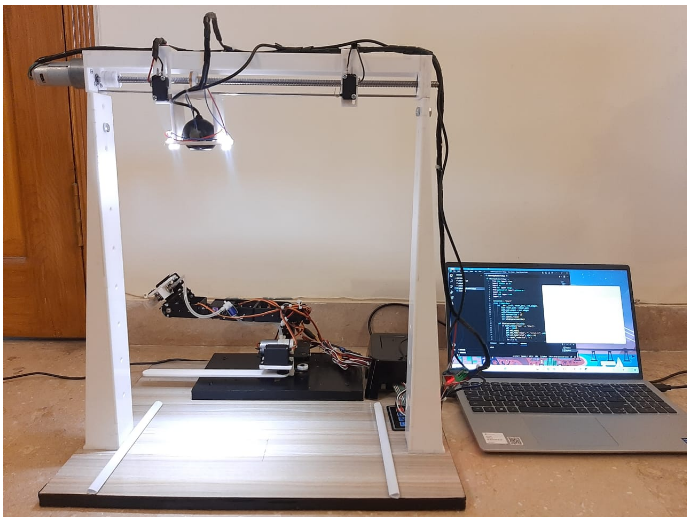
□

CORRIE. Of course, they will, though you're quite justified in feeling□
skeptical about them. It was very hard to get them to use tanks in the□
war very hard. Those cavalry generals had to be forced to use them. They□
ought to be horse-knackers instead of soldiers. And tin hats, too! Look□
what a time it was before that damned War Office could be persuaded to□
use' em! ... But I'm sorry, Charlotte. 1 Ought not to be talking about the□
War to you — especially to-day.□

□

Figure 3.25: PDF

3.8 Hardware Model



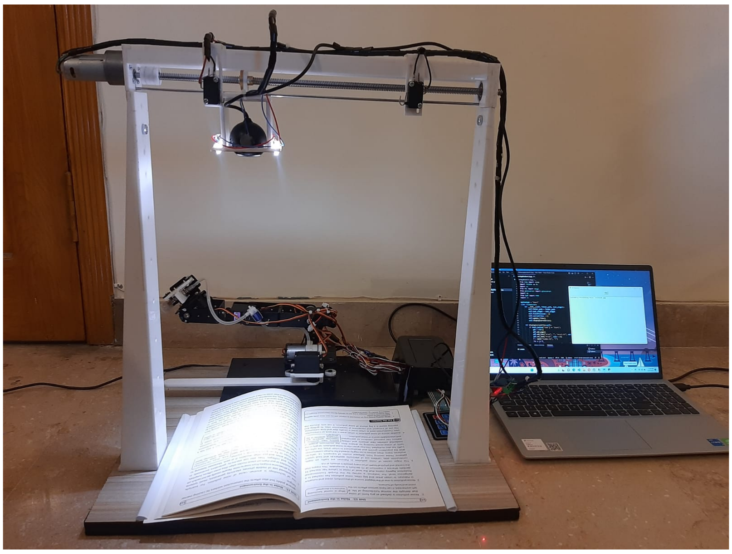


Figure 3.26: Hardware Model

3.9 Working Of Hardware

We have made an Automatic Book Scanner that can scan the pages of a book and turn them into text files. The scanner starts working when the user types the desired number of pages he/she wants to scan using a 4x3 keypad. As soon as the user has typed the number of pages the robotic Arm starts working. First the Robotic Arm moves to its initial position. Then it moves towards the book. Now the robotic arm is at the top of the book and then it starts moving towards the book to pick the page. There is a touch sensor at the end of the robotic arm. When this sensor detects the page it turns on the vacuum pump. If it does not detect anything then the robotic arm stops working. The vacuum pump stick to the page and after that the robotic arm start moving to its second position with the page lifted by the vacuum pump. After that the flipping rod starts working which will flip the lifted page. The flipping rod moves from 0 to 180 degrees. If the page is not flipped then the robotic arm stops working. If the page is flipped then the vacuum will turn off and the robotic arm moves out of the frame of the camera. After that the camera will click the picture of one page of the book. After that the slider will move the camera to the second page of the book and camera clicks the picture of the second page. This process repeats until all the desired pages of the book are scanned. When the whole scanning process completes then the scanned images of the book is converted into text form in a single file. And this is the end of the process.

3.10 Hardware Code Explanation

We designed our hardware in such a way that our robotic arm along with vacuum pump help in the flipping of book. We also added a flipping rod in our hardware design. There is slider and camera to take pictures. We did Arduino coding according to our requirement. It includes the necessary libraries for using a keypad and controlling servos. These libraries are servo and keypad. We are using 4 servo for robotic arm movement. There is another servo motor for flipping rod. There is a solenoid and a vacuum pump. In last there is a DC motor for slider movement. We first declared the pin numbers. Then in the setup() function serial communication is initialized. Pin modes are set for input and output. Servos are attached to their respective pins. Initial positions of servos are set. In the loop() function the user will enter a three-digit number using the keypad. The number is converted to an integer. If the number is greater than zero, a loop is executed. scanLtPage() is called to scan the left page. scanRtPage() is called to scan the right page. changePage() is called to perform a page change operation. After the loop ends, the position of some servos is reset. Now there are three types of operations that will be performed. Change page operation is used to flip the page. We specify different rotations and positions of servo motors. Flipping rod will move 180 degree and vacuum will turn on and off at specific time to pick the page. Next The scanLtPage() function. We will Increments i. And then it activates the left camera (camLeft). Waits until the limit switch (limit) is activated. Deactivates the left camera. Prints the current page number (i).Triggers the camera (camTrig) sand in the end it will decrements num.

The variable i is incremented to keep track of the current page number. The camRight pin is set to LOW to deactivate the right camera (assuming it was previously activated).The camLeft pin is set to HIGH to activate the left camera. There's a delay of 2000 milliseconds (2 seconds) to allow time for the camera to stabilize or adjust. The function enters a while loop that checks the state of the limit pin using digitalRead(limit).The loop continues as long as the limit pin is LOW, indicating that the limit switch is not activated. This loop effectively waits until the limit switch is triggered, indicating that the camera has reached the desired position or end of the page. Once the limit switch is activated, the camLeft pin is set back to LOW to deactivate the left camera .The current page number (i) is printed to the serial monitor using Serial.println(). There's a delay of 500 milliseconds to allow for any necessary processing or stabilization. The camTrig pin is set to HIGH to trigger the camera to capture an image or perform a specific action. There's a delay of 500 milliseconds to ensure that the trigger signal is sent for an adequate duration. The camTrig pin is set back to LOW to deactivate the trigger signal. There's a delay of 500 milliseconds before moving to the next step or iteration. The variable num is decremented, which indicates that one page has been scanned or processed.

Essentially, the scanLtPage() function sets up and activates the left camera, waits for the limit switch to indicate the end of the page, triggers the camera to perform an action (such as capturing an image), and updates the necessary variables to keep track of the progress. The function will perform on the scanRtpage(). That will trigger the right camera .This whole process will repeat depending upon the number of digits enter on the keypad.

Flow chart of Hardware Implementation

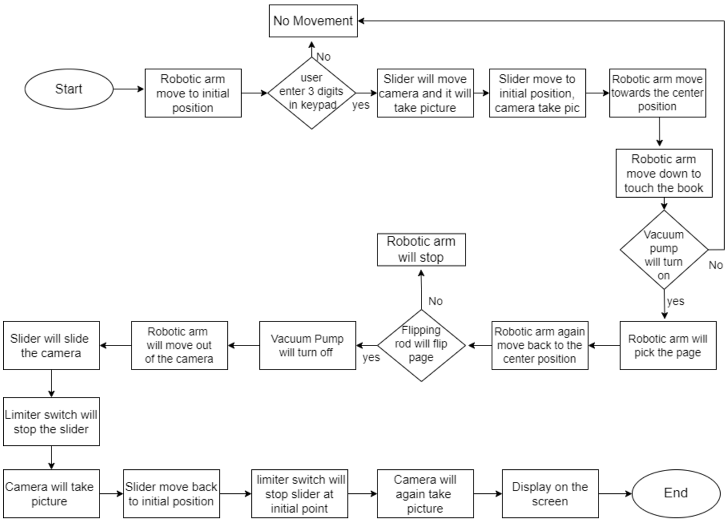

Start
Robotic arm move to initial position
user enter 3 digits in keypad
No
No Movement
yes
Slider will move camera and it will take picture
Slider move to initial position, camera take pic
Robotic arm move towards the center position
Robotic arm move down to touch the book
Vacuum pump will turn on
No
yes
Robotic arm will pick the page
Robotic arm again move back to the center position
Flipping rod will flip page
No
Robotic arm will stop
yes
Vacuum Pump will turn off
Robotic arm will move out of the camera
Slider will slide the camera
Limiter switch will stop the slider
Camera will take picture
Slider move back to initial position
limiter switch will stop slider at initial point
Camera will again take picture
Display on the screen
End

Figure 3.27: Flow chart of Hardware Implementation

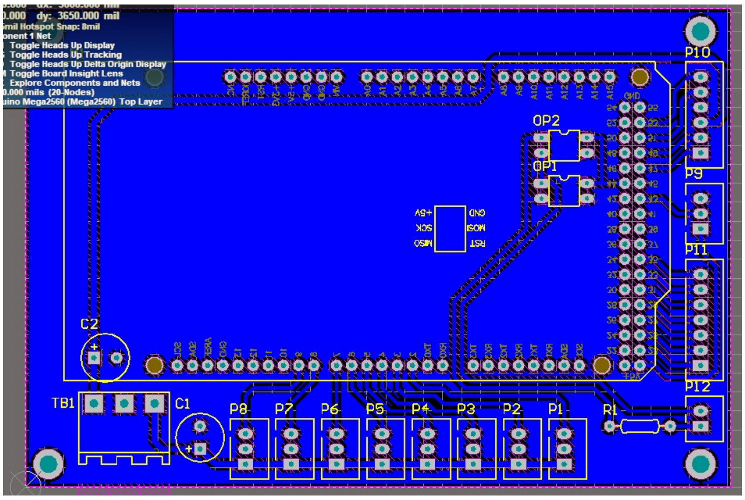

dy: 3650.000 mil
Toggle Heads Up Display
Toggle Heads Up Tracking
Toggle Heads Up Delta Origin Display
Toggle Board Insight Lens
Explore Components and Nets
Mega2560 (Mega2560) Top Layer
P10
OP2
OP1
P9
P11
C2
P12
TB1
C1
P8
P7
P6
P5
P4
P3
P2
P1
R1

Figure 3.28: PCB Layout

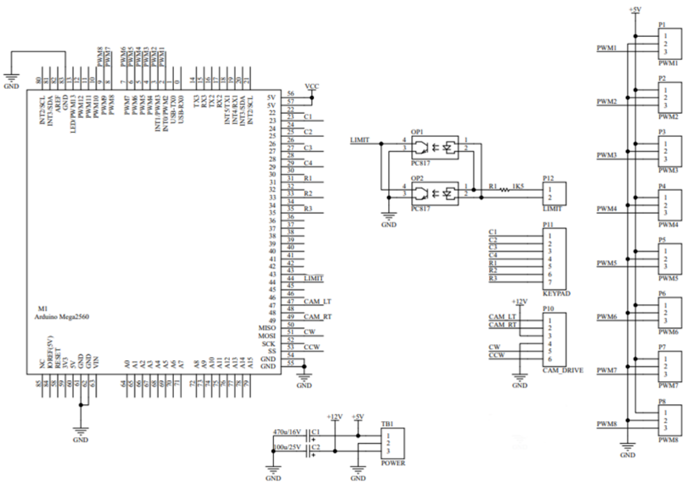


Figure 3.29: PCB Schematics

3.11 Software Code Explanation

3.11.1 Image to Text

Our next major task was to perform a text extraction on the camera images and save it to some external memory or in the form of pdf . Our code is a Python script that monitors a specified folder for new image files, performs optical character recognition (OCR) on each image file using Tesseract OCR, and displays the processed file names and extracted text in a Tkinter GUI window and in the end it will create the pdf. A common Python library for building graphical user interfaces (GUIs) is Tkinter. You can create windows, buttons, labels, text boxes, and other GUI components using the classes and functions that are provided. To create a Tkinter window we Import the tkinter module. Then we create an instance of the Tk class, which represents the main window. In Tkinter, a Text widget is a GUI element that provides a multi-line text editing area. It allows users to enter, view, and manipulate text content. The Text widget is versatile and supports various text-related operations, such as inserting text, deleting text, selecting text, and applying formatting. In Tkinter tk.END is a constant that represents the special index used to refer to the end of a text widget's content. It is used to specify positions or ranges within a Text widget for various text-related operations. The tk.END constant is typically used as an index to indicate the end position of the text content within the widget. It allows you to perform operations such as inserting text at the end, deleting text from a specific position to the end, or retrieving text from a specific position to the end.

First the necessary libraries are imported: sleep from the time module, tkinter for GUI, os for file operations, Image from the PIL (Python Imaging Library) module, pytesseract for OCR,FPDF from fpdf as it will imports the FPDF class, which is used to create PDF documents programmatically and io. io will provides functions for working with streams, which are used to handle PDF output. The sleep() function is a part of the time module in Python. It provides a way to pause the execution of a program for a specified number of seconds. It is commonly used for introducing delays or creating time intervals in code. Inside a class, we can define attributes (variables) and methods (functions) that belong to the class. The attributes store data associated with the class, and the methods define the behaviors or actions that the class can perform.

The FileWatcher class is defined, which will handle monitoring the folder and processing the files. It watches the specified folder for new files. It initializes with the folder path and a text widget where the output is displayed. It keeps track of the files that have been processed and the current files in the folder. We have form two types of function in it. One is the displayCurrentFile(self) and other is update files(self). displayCurrentFile displays the name of the file that is currently being processed and calls the mainOcrFunction to perform OCR on the image. Once the OCR is completed, it updates the processed files list and schedules itself to be called again after a delay of 10 seconds. The text widget is updated with the most recent information regarding file processing via the displayCurrentFile method of the FileWatcher class. The old text widget content is removed, and the name of the currently processed file is substituted. It calls the mainOcrFunction to perform OCR on the image file if it has never been handled previously. After the OCR is finished, it refreshes the list of files that have been processed and sets a 10-second wait before calling itself again. A notification stating that all files have been processed is displayed when there are no more files to process. update files(self) updates the list of current files in the folder.

The mainOcrFunction function uses the Pytesseract library to conduct OCR on the picture file and as an output we retrieved a text file. When a user presses the “Start” button on the GUI, the start watching function is called. It sets up a FileWatcher object with the text widget and folder location. The GUI consists of Start button and a text widget where the output is shown. The start watching function start working when the user hits the Start button initialising the FileWatcher object and starting to watch the specified folder for new files.

In the display current file if the start button's text is Start it means the watcher is stopped, and the method creates a PDF document named "output.pdf" using the FPDF class and the content of a "readme.txt" file.If the start button's text is "Stop," it means the watcher is active, and the method updates the text widget with information about the currently processing file. In the update file if the start button's text is "Start," the method returns early. If the start button's text is "Stop," the method updates the current files list every 1 second using the after method. Using this code, a GUI was created that enables users to specify a folder path and check it frequently for new picture files. Tesseract OCR is used to process each new file, and the GUI displays the file names and text that was extracted in real-time. In the end it will create the PDF as an output.

Flow chart of Software (Image to Text)

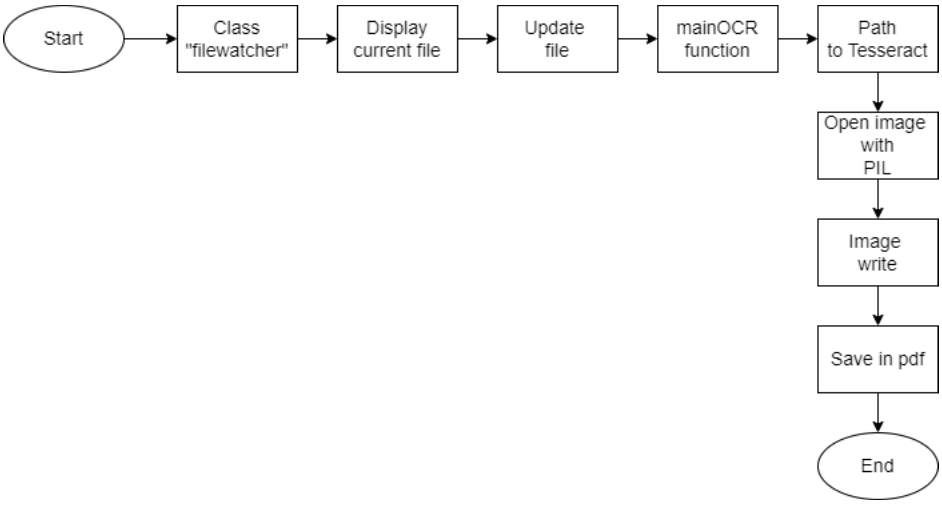


Figure 3.30: Flow Chart of Image to Text

Chapter 4

Results and Discussion

In our FYP III we have completed all the tasks and objectives of our project. As the name indicates our project have 2 major portions one is hardware part includes the whole setup of robotic arm, flipping rod, power supply, sensors, camera and whole framing of book scanner. And our software part includes the whole storage of images and their compilation and conversions. The whole process is automatic and does not require any human effort. In our FYP 3 we have achieved the following milestones
1. Robotic arm and its required rotations.
2. Servo motors interfacing with Arduino.
3. Keypad interfacing
4. Vacuum Pump interfacing
5. Slider designing and Interfacing 6. Flipping Rod
7. Camera interfacing with Arduino and displacing the snap shots on laptop screen.
8. Images to text file conversion.
9. PDF generation 10. Made an user friendly GUI

We have successfully achieved 85% of our deliverable. Our machine can scan 5 pages per minute and 300 pages per hour.
Firstly we have interfaced a 4x3 keypad so that the user can select the number of pages he or she wants to scan. The user can select any 3 digit number.
Secondly we have interfaced the robotic arm with Arduino and attached the required servos with it. Our robotic arm is moving in all designated rotation one by one and its mechanical rotations are visible in our project video. We are using our Robotic Arm with a vacuum pump to lift the pages of the book. And used a flipping rod to flip that page.

Thirdly we have successfully interfaced camera C130 with Arduino Mega2560 using software VC90. Also we have attached a slider with our camera. Firstly the slider will move the camera to take picture of first page of the book then the slider will move back to its initial position to take the picture of the second page. This process is working in loop according to the number of pages. The pictures are displayed on laptop screen.
The second part of our project is image to text conversion. We are using different libraries of python to extract the text from images and to store all these images we have designed a GUI which will show the image conversion process and also stores the files in a proper sequence in a specified folder.

Chapter 5

Impact on Environment

Digitization of hardcover documents will reduce the human dependency on paper and automatically reduce the deforestation caused for paper products every year. In the process of making physical books the paper come directly from the trees. And not from some recycled materials. Speaking of the production of the books globally, the process of producing books itself releases more than 40 million metric tons of CO2. So the use of digital book is much more sustainable and in addition to that it also helps to protect our environment by getting rid of the excessive use of paper. In digital form, extensive data can be stored in a small chip. They are portable and easily accessible for every user. Scanning book manually by turning each page one by one is really hectic and time consuming. Our project of scanning book automatically will help the user to save time and energy. The challenge of obtaining the specific information requested by the user can be easily searchable. In digital form, data will be secure, and no physical damage to the documents will affect the digital copy. As soon as we realize that by digitizing our books we really don't have to worry about the printing and re printing activities. And also by digitizing your books we can save a great amount of paper and therefore leaving a green carbon footprint. Our selected prototype of an automatic book scanner design is cost efficient and beneficial for users by saving their time and money and also beneficial for the environment and it will have a positive impact on environment and sustainability.

Chapter 6

Reference

1.https://www.researchgate.net/publication/328398085AutomaticBookScanner
2.https://www.researchgate.net/publication/333310275DevelopmentofanAutomatedLow-costBookScannerandTranslator
3.https://iopscience.iop.org/article/10.1088/1757-899X/417/1/012032
4.https://digitalcommons.calpoly.edu/cpesp/123/
5.https://nanonets.com/blog/ocr-with-tesseract/
6.https://python-bloggers.com/2022/05/extract-text-from-image-using-python/
7.https://mschoeffler.com/2021/07/17/arduino-tutorial-mg-996r-servo-motor/
8.https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/
9.http://www.martyncurrey.com/activating-the-shutter-release/
10.https://www.mega.pk/webcamerasproducts/2443/Logitech-Webcam-C120.html

Appendix A

First

```
#include <Keypad.h>
#include <Servo.h>
Servo srvo1;                // create servo object to control a servo
Servo srvo2;                // create servo object to control a servo
Servo srvo3;                // create servo object to control a servo
Servo srvo4;                // create servo object to control a servo
Servo srvoRod;              // Flipping Rod
Servo srvoPump;             // Pump
Servo srvoSol     ;         // Solenoid
const int camTrig = 9;      // Camera Trigger

const int limit = 44;       // limit sensing pin
int touchState = 0;

const int camIn = 38;
const int camLeft = 47;
const int camRight = 49;
const int cw = 51;
const int ccw = 53;

const byte ROWS = 4;
const byte COLS = 3;

char keys[ROWS][COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte rowPins[ROWS] = {35, 33, 31, 29};  //connect to the row pinouts of t
byte colPins[COLS] = {27, 25, 23};      //connect to the column pinouts o

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
String cmd;
int i,num;
int angle;

void setup(){
  Serial.begin(9600);
  pinMode(camLeft, OUTPUT);
  pinMode(camRight, OUTPUT);
  pinMode(camTrig, OUTPUT);
  pinMode(limit, INPUT_PULLUP);

  srvo1.attach(2);
  srvo2.attach(3);
  srvo3.attach(4);
  srvo4.attach(5);
  srvoRod.attach(6);
  srvoPump.attach(7);
  srvoSol.attach(8);
  srvo1.write(0);
  srvo2.write(20);        // Fix
  srvo3.write(45);
  srvo4.write(0);
  srvoRod.write(180);
  srvoPump.write(0);
  srvoSol.write(0);
}

void loop(){
  char key;
  String input = "";
  int numDigits = 0;

  while (numDigits < 3){
    key = keypad.getKey();
    if (key) {
      input += key;
      numDigits++;
      Serial.print(key);
    }
  }

  Serial.println(" - Input complete");
  num = input.toInt();
  Serial.println("Number entered: " + String(num));
  if (num > 0){
    i = 0;
    while(num > 0){
      scanLtPage();
      while(num > 0){
        scanRtPage();
        break;
      }
      while(num > 0){
        changePage();
        break;
      }
    }
  }
  else{
    Serial.println("Invalid Value");
  }
  delay(500);
  srvo2.write(20);
  srvo3.write(45);
  srvo4.write(0);
  delay(1000);
  srvo1.write(0);
}

void changePage(){
  srvo1.write(55);
  delay(2000);
  srvo2.write(5);
  srvo4.write(20);
  srvo3.write(45);
  delay(500);
  for (int angle = 45; angle > -1; angle--){
    srvo3.write(angle);
    delay(50);
  }
  srvoPump.write(235);
  srvoSol.write(0);
  delay(2000);
  for (int angle = 0; angle < 46; angle++){
    srvo3.write(angle);
    delay(50);
  }
  delay(1000);

  srvoPump.write(0);
  srvoSol.write(0);
  delay(500);
  srvoRod.write(150);
  delay(1000);


  srvoRod.write(20);
  delay(2000);
  srvoRod.write(180);
  delay(2000);
  cmd="";

srvo1.write(30);
  delay(2000);
}

void scanLtPage(){
  i++;
  digitalWrite(camRight, LOW);
  digitalWrite(camLeft, HIGH);
  delay(2000);
  while(digitalRead(limit)==LOW){}
  digitalWrite(camLeft, LOW);
  Serial.println("Page#: " + String(i));
  delay(500);
  digitalWrite(camTrig, HIGH);
  delay(500);
  digitalWrite(camTrig, LOW);
  delay(500);
  num--;
}

void scanRtPage(){
  i++;
  digitalWrite(camLeft, LOW);
  digitalWrite(camRight, HIGH);
  delay(2000);
  while(digitalRead(limit)==LOW){}
  digitalWrite(camRight, LOW);
  Serial.println("Page#: " + String(i));
  delay(500);
  digitalWrite(camTrig, HIGH);
  delay(500);
  digitalWrite(camTrig, LOW);
  delay(500);
  num--;
}
```

Appendix B

Second

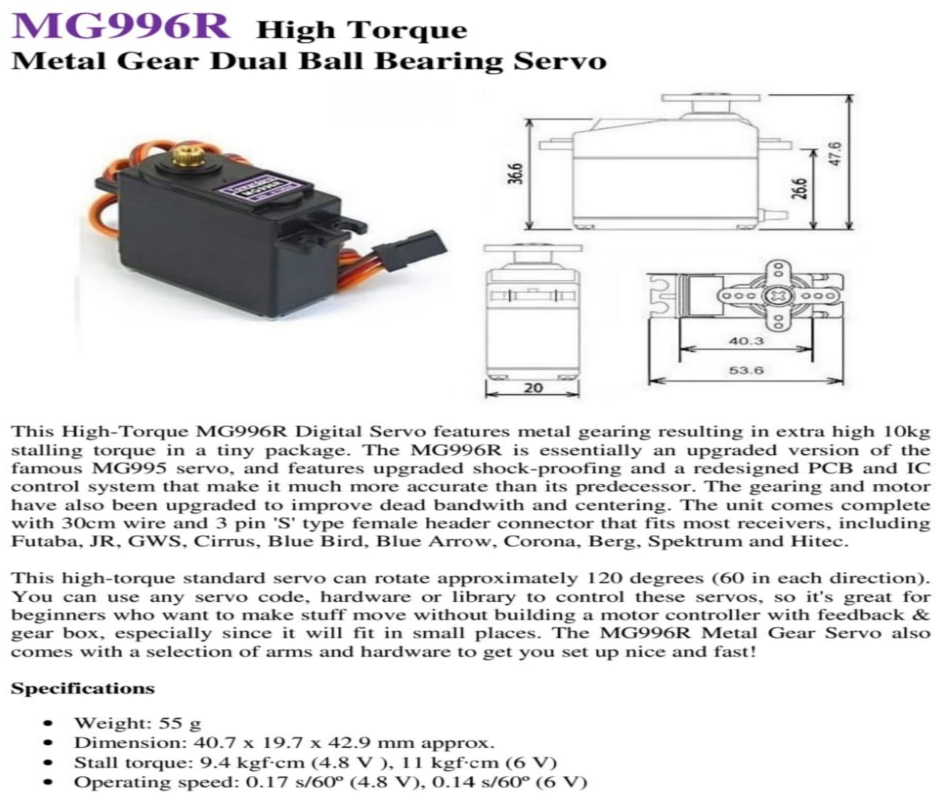
```
from time import sleep
import tkinter as tk
import os
from PIL import Image
from pytesseract import pytesseract
import os

class FileWatcher:
def _init_(self, folder_path, text_widget):#Initializes the FileWatcher
object.
self.folder_path = folder_path #Stores the folder path.
self.text_widget = text_widget #Stores the Text widget.
self.filesProcessed = []  #Keeps track of files been processed.
self.currentFiles = [] #Stores the list of current files in the folder.
self.update_files()  # it call this file to update the file list
self.displayCurrentFile()  #method to display current file being processe

def displayCurrentFile(self): #Displays the current file being
                               #processed in the Text widget.

        self.text_widget.delete('1.0', tk.END)
        for i in range(0, len(self.currentFiles)):
            if self.currentFiles[i] not in self.filesProcessed:
                print(self.currentFiles[i])
self.text_widget.insert(tk.END,'Currently Processing File:'+
self.currentFiles[i] +'\n')
                self.filesProcessed.append(self.currentFiles[i])
```

```
filePath = self.folder_path + "\\" + self.currentFiles[i]
mainOcrFunction(filePath)
self.text_widget.after(10000, self.displayCurrentFile)
    return
    self.text_widget.insert(tk.END, 'All files have been processed\n')
        self.text_widget.after(10000, self.displayCurrentFile)
        return;



def update_files(self):
self.currentFiles = os.listdir(self.folder_path)#to get list of files
#in the specified folder and assigns it to self.currentFiles.

        self.text_widget.after(1000, self.update_files)


def mainOcrFunction(filePath):
    print(filePath)
    #Define path to tessaract.exe
    path_to_tesseract = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
    pytesseract.tesseract_cmd = path_to_tesseract
    #Open image with PIL
    img = Image.open(filePath)
    #Extract text from image
    text = pytesseract.image_to_string(img)
    print(text)
    f = open("readme.txt", 'a')
    f.write(text)
    f.close()
    return
def start_watching():
    folder_path = r'C:\Users\Hammas Ahmed\Downloads\finalzip2\finalzip'  #
    text_widget.delete('1.0', tk.END)
    watcher = FileWatcher(folder_path, text_widget)

root = tk.Tk()  #This line creates the main Tkinter window object, which
#is typically referred to as root
root.title("File Watcher")#This line sets title of window to File Watche

start_button = tk.Button(root, text="Start", command=start_watching)
```

```
#it create a button name start
start_button.pack(pady=10) # pack automatically arrange the widget in
#vertical manner
# pady argument specifies the vertical padding (space) around the button

text_widget = tk.Text(root)
text_widget.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)
root.mainloop()
```

Appendix C

Data Sheets



MG996R High Torque
Metal Gear Dual Ball Bearing Servo

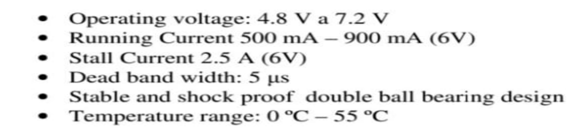
36.6
26.6
47.6
40.3
53.6
20

This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwith and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

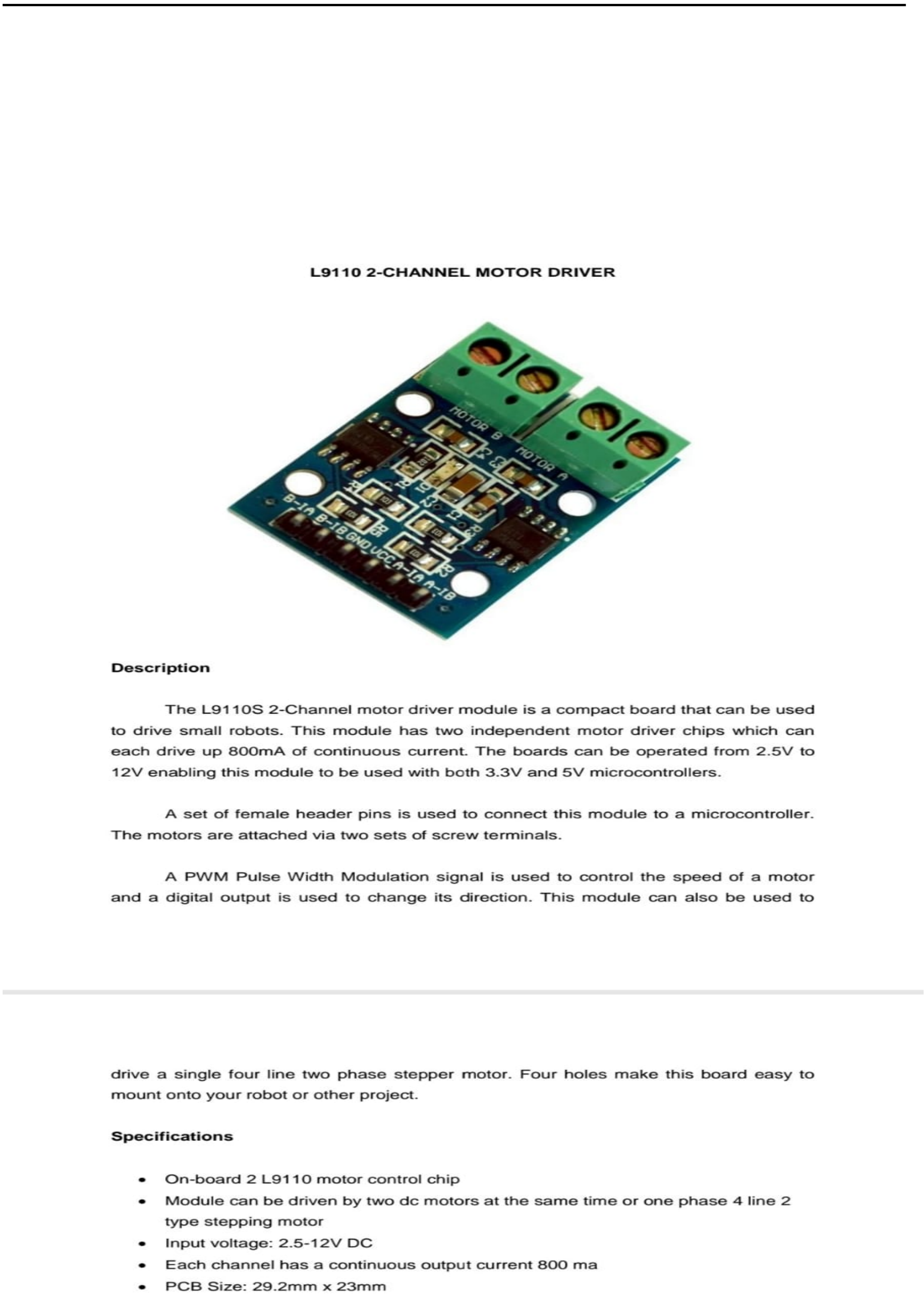
This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60º (4.8 V), 0.14 s/60º (6 V)



- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Temperature range: 0 ºC – 55 ºC



MOTOR B
MOTOR A
B-1A
B-1B
GND
VCC
A-1A
A-1B

L9110 2-CHANNEL MOTOR DRIVER

Description

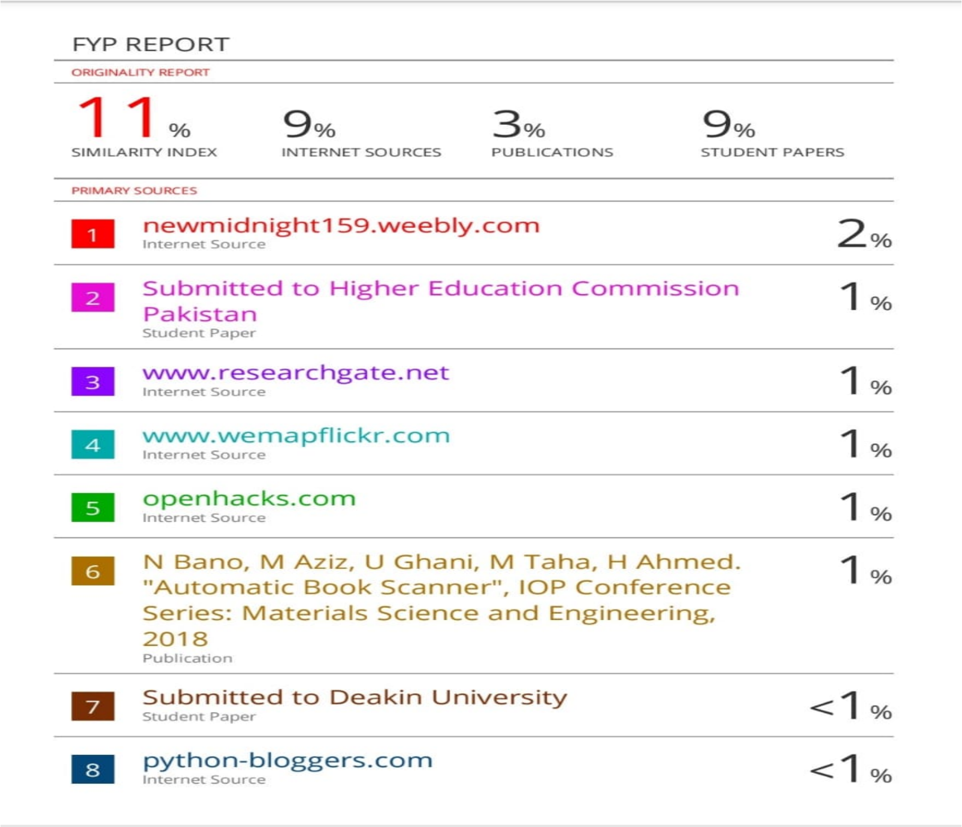
The L9110S 2-Channel motor driver module is a compact board that can be used to drive small robots. This module has two independent motor driver chips which can each drive up 800mA of continuous current. The boards can be operated from 2.5V to 12V enabling this module to be used with both 3.3V and 5V microcontrollers.

A set of female header pins is used to connect this module to a microcontroller. The motors are attached via two sets of screw terminals.

A PWM Pulse Width Modulation signal is used to control the speed of a motor and a digital output is used to change its direction. This module can also be used to drive a single four line two phase stepper motor. Four holes make this board easy to mount onto your robot or other project.

Specifications

- On-board 2 L9110 motor control chip
- Module can be driven by two dc motors at the same time or one phase 4 line 2 type stepping motor
- Input voltage: 2.5-12V DC
- Each channel has a continuous output current 800 ma
- PCB Size: 29.2mm x 23mm



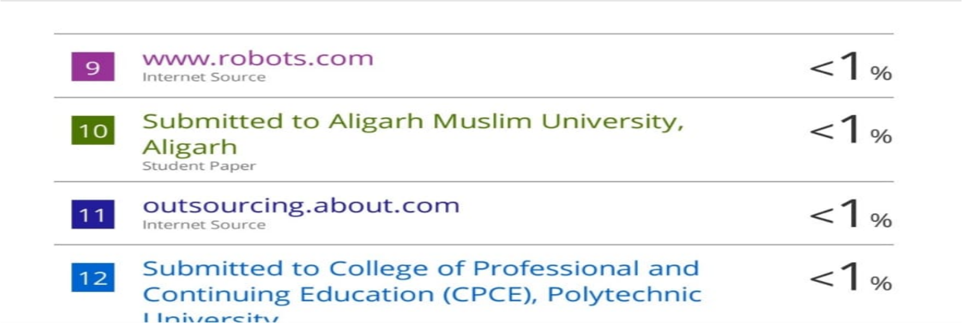
FYP REPORT

ORIGINALITY REPORT

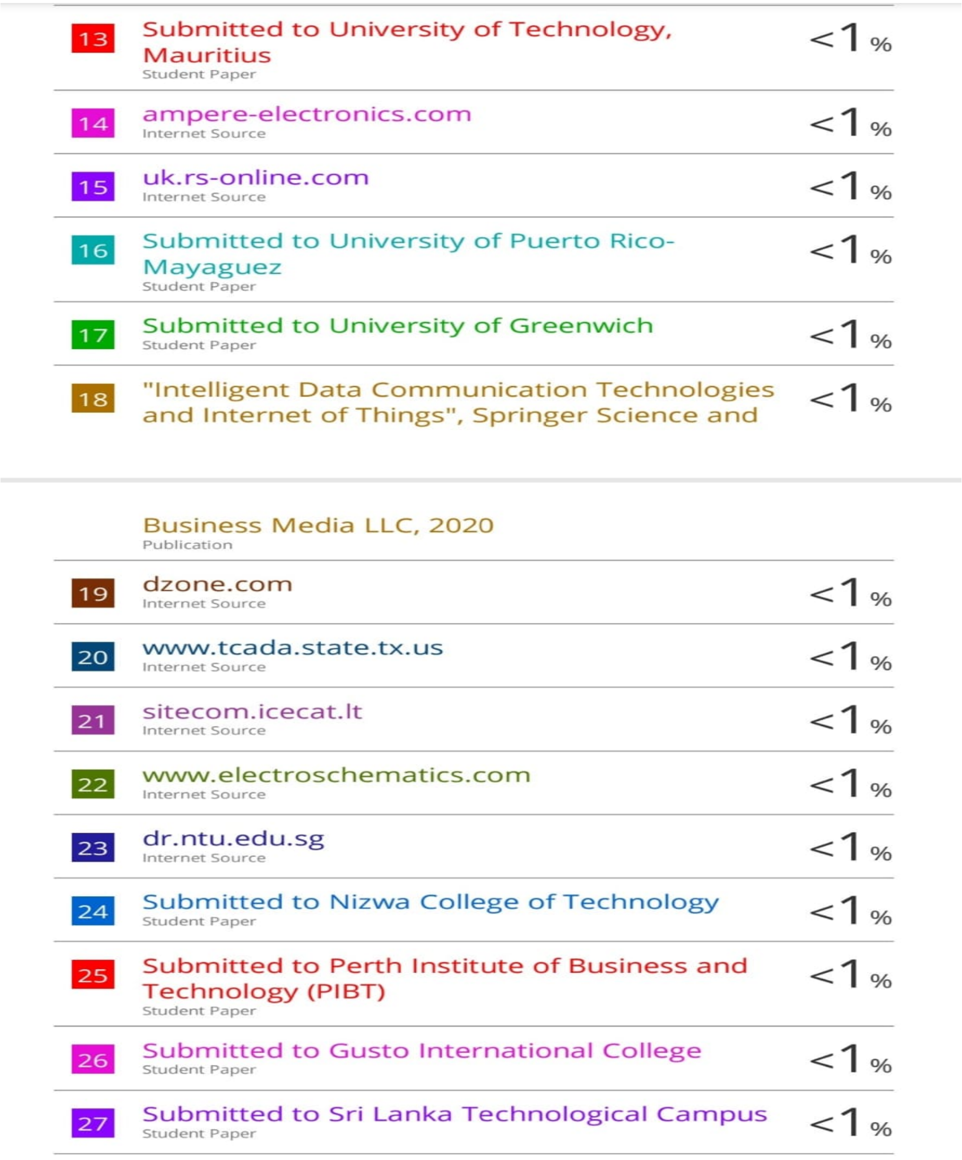
11%	9%	3%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

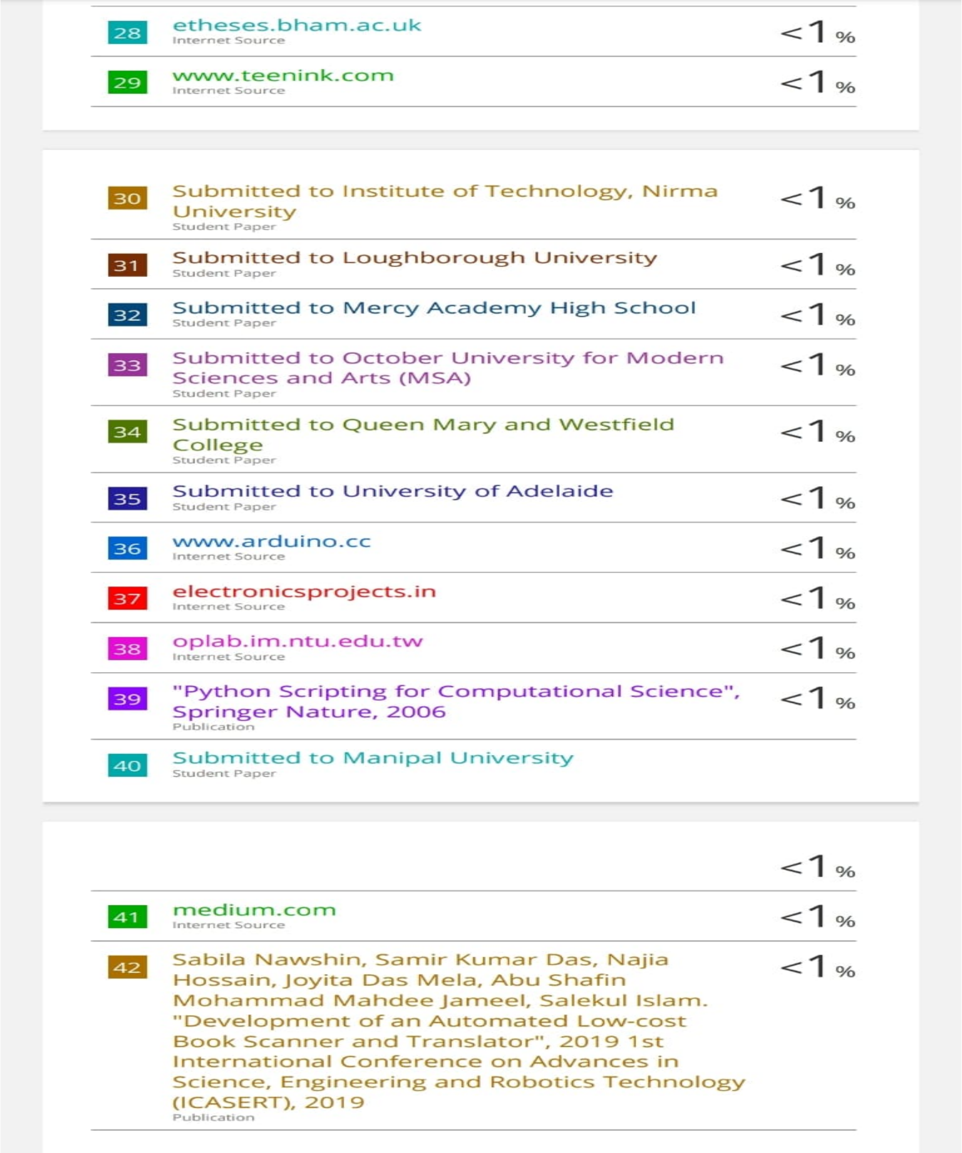
1	newmidnight159.weebly.com Internet Source	2%
2	Submitted to Higher Education Commission Pakistan Student Paper	1%
3	www.researchgate.net Internet Source	1%
4	www.wemapflickr.com Internet Source	1%
5	openhacks.com Internet Source	1%
6	N Bano, M Aziz, U Ghani, M Taha, H Ahmed. "Automatic Book Scanner", IOP Conference Series: Materials Science and Engineering, 2018 Publication	1%
7	Submitted to Deakin University Student Paper	<1%
8	python-bloggers.com Internet Source	<1%



9	www.robots.com Internet Source	<1%
10	Submitted to Aligarh Muslim University, Aligarh Student Paper	<1%
11	outsourcing.about.com Internet Source	<1%
12	Submitted to College of Professional and Continuing Education (CPCE), Polytechnic University	<1%



13	Submitted to University of Technology, Mauritius Student Paper	<1 %
14	ampere-electronics.com Internet Source	<1 %
15	uk.rs-online.com Internet Source	<1 %
16	Submitted to University of Puerto Rico-Mayaguez Student Paper	<1 %
17	Submitted to University of Greenwich Student Paper	<1 %
18	"Intelligent Data Communication Technologies and Internet of Things", Springer Science and Business Media LLC, 2020 Publication	<1 %
19	dzone.com Internet Source	<1 %
20	www.tcada.state.tx.us Internet Source	<1 %
21	sitecom.icecat.lt Internet Source	<1 %
22	www.electroschematics.com Internet Source	<1 %
23	dr.ntu.edu.sg Internet Source	<1 %
24	Submitted to Nizwa College of Technology Student Paper	<1 %
25	Submitted to Perth Institute of Business and Technology (PIBT) Student Paper	<1 %
26	Submitted to Gusto International College Student Paper	<1 %
27	Submitted to Sri Lanka Technological Campus Student Paper	<1 %



28	etheses.bham.ac.uk Internet Source	<1 %
29	www.teenink.com Internet Source	<1 %
30	Submitted to Institute of Technology, Nirma University Student Paper	<1 %
31	Submitted to Loughborough University Student Paper	<1 %
32	Submitted to Mercy Academy High School Student Paper	<1 %
33	Submitted to October University for Modern Sciences and Arts (MSA) Student Paper	<1 %
34	Submitted to Queen Mary and Westfield College Student Paper	<1 %
35	Submitted to University of Adelaide Student Paper	<1 %
36	www.arduino.cc Internet Source	<1 %
37	electronicsprojects.in Internet Source	<1 %
38	oplab.im.ntu.edu.tw Internet Source	<1 %
39	"Python Scripting for Computational Science", Springer Nature, 2006 Publication	<1 %
40	Submitted to Manipal University Student Paper	
		<1 %
41	medium.com Internet Source	<1 %
42	Sabila Nawshin, Samir Kumar Das, Najia Hossain, Joyita Das Mela, Abu Shafin Mohammad Mahdee Jameel, Salekul Islam. "Development of an Automated Low-cost Book Scanner and Translator", 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019 Publication	<1 %

Appendix D

Sustainable Development Goals (SDG's)

1) Quality Education

2) Climate Action