Personal Assistant Robot Using Arduino

Major Project (CS705PC)

Submitted

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Mr. B. SAI SANKEERTH REDDY (17261A0504)

Under the Guidance of

Dr. C.R.K. Reddy

Professor

Department of Computer Science & Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

GANDIPET, HYDERABAD – 500 075

INDIA

2020-2021

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
GANDIPET, HYDERABAD – 500 075. Telangana

CERTIFICATE

This is to certify that the thesis entitled **Personal Assistant Robot Using Arduino** is being submitted by **B. SAI SANKEERTH REDDY** in partial fulfillment for the award of **B. Tech** in **Computer Science & Engineering** to **Jawaharlal Nehru Technological University Hyderabad** is a record of bonafide work carried out by his under our guidance and supervision.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

Supervisor and coordinator
Dr. C. R. K. Reddy
Professor

HOD
Department of CSE

External Examiner

DECLARATION

This is to certify that the work reported in this project titled "**Personal Assistant Robot Using Arduino"** is a record of work done by me in the **Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad.**

No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the work done entirely by me and not copied from any other source.

B. SAI SANKEERTH REDDY

(17261A0504)

ACKNOWLEDGEMENT

I would like to express our sincere thanks to **Dr. K. Jaya Sankar, Principal, MGIT**, for providing the working facilities in college.

I wish to express our sincere thanks and gratitude to **Dr. C. R. K. Reddy, Professor and HOD**, Department of CSE, MGIT, for all the timely support and valuable suggestions during the period of project.

I am extremely thankful to **Dr. C. R. K. Reddy, Professor and B. Prashanthi, Associate Professor,** Department of CSE, MGIT, major project coordinators for their encouragement and support throughout the project.

I am extremely thankful and indebted to my internal guide **Dr. C. R. K. Reddy, Professor,** Department of CSE, for her constant guidance, encouragement and moral support throughout the project.

Finally, I would also like to thank all the faculty and staff of CSE Department who helped us directly or indirectly, for completing this project.

B. SAI SANKEERTH REDDY

(17261A0504)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This robot uses Computer Vision models to dynamically follow humans in real time avoiding particular objects around . The robot truly is a personal assistant and its powered by Arduino UNO and a camera which has Kendryte K210 processor. The most important factor for an autonomous robot is to have a continuous interaction between the person and the robot itself. Our project is about a robot that autonomously follows humans as they move in any direction and assist them. The interaction between the robot and the humans should bring ease to him during any of the normal daily life activities. Our robot is designed by humans to recognizing there faces. The main hindrance in this kind of work is that it might start operating when it detects many faces. This project discusses about a device on the Arduino platform, using an inexpensive computer vision camera and L29-3D chip that is installed on chassis with four DC-motors attached to it controlling the wheels. All the processing is done on Arduino micro-controllers communicating seamlessly with Motor Driver and other parts.

Keywords: Arduino, micro-controllers, Computer Vision, Kendryte K210 processor, DC-motors, L29-3D chip.

1. INTRODUCTION

The **Internet of things** (**IoT**) describes the network of physical objects—a.k.a. "things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet. Things have evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation , and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. The IoT can also be used in healthcare systems.

This robot can use Computer Vision models to dynamically follow humans and particular objects around. The robot truly is a personal assistant and its powered by Arduino UNO and a camera which has Kendryte K210 processor. This robot is design as a mini car which which recognizes faces and follows them by ignoring walls, chairs and other obstacles. This paper discusses about a device on the Arduino platform, using an inexpensive computer vision camera and L29-3D chip that is installed on chassis with four DC-motors attached to it controlling the wheels. All the processing is done on Arduino micro-controllers communicating seamlessly with Motor Driver and other parts.

1.1 Problem Definition

This robot can use Computer Vision models to dynamically follow humans and particular objects around. The robot truly is a personal assistant and its powered by Arduino UNO and a camera which has Kendryte K210 processor. The most important factor for an autonomous robot is to have a continuous interaction between the person and the robot itself.

1.2 Existing System

The customer needs to wait in the supermarket for shopping because it is a highly time-consuming process. A huge crowd in the supermarket at the time of discount offers or weekends makes trouble to wait in long queues because of a bar-code based billing process. In this regard, the Internet of Things (IoT) based Smart Shopping Cart is proposed which consists of Radio Frequency Identification (RFID) sensors, Arduino micro-controller, Bluetooth module, and Mobile application. RFID sensors depend on wireless communication. One part is the RFID tag attached to each product and the other is RFID reader that reads the product information efficiently[1]. After this, each product information shows in the Mobile application. The customer easily manages the shopping list in Mobile application according to preferences. Then shopping information sends to the server wirelessly and automatically generates billing. This experimental prototype is designed to eliminate time-consuming shopping process and quality of services issues. The proposed system can easily be implemented and tested at a commercial scale under the real scenario in the future. That is why the proposed model is more competitive as compared to others.

Disadvantages

1. Radio Frequency signals should be different for each device to differentiate with other devices and on large scale producing many radio frequency without air traffic is very difficult.
2. Not effective on large scale.
3. Cannot recognizes the customer to follow them and needed to be guided by the customer as it’s not fully automated.

1.3 Proposed System

The modern age of technology follows humans and can also be developed to follow path and signs which can be implemented in super markets and hospitals as the robot processes the Artificial Intelligence algorithms which can take advantage of the speed of the robot as movement it moves very fast, which we can implement in real world. Its powered by Arduino UNO and a camera which has Kendryte K210 processor which is cheap and also boosts the performance.

Advantages

1. Uses face recognition which can detect and follows without any issues.
2. This model is convenient to make on large scale because it is very cheap when compared to the other models
3. Very easy to use and it is fully automated

1.4 Requirements Specification

1.4.1 Software Requirements

1. Operating System: Windows 10/Linus/Mac OS
2. Software: Arduinio IDE
3. Language: C, Python

1.4.2 Hardware Requirements

1. Arduino UNO R3 Chip
2. L29-3D Motor Driver
3. Camera with Kendryte K210 processor
4. 4- Dc-Motors and Wheels
5. 7.4V-8.4V - 850mAh Battery
6. Chassis for robot
7. Multi-meter
8. Jumper Wires
9. Other parts (like- Switch, wires, battery holder, battery charger, battery connector)

2. LITERATURE SURVEY

In our literature survey we have investigated various researches on this particular domain some of them are as follows :-

1) **Arduino based human-following IV stand (Ali Kassem Merhi; Murtada Mohammad Hasan; Samih Abdul-Nabi; Ali Bazzi)** [4], The most important factor for an autonomous robot is to have a continuous interaction between the person and the robot itself. Many researches have been done in developing such robots, but none of them were related to improving the IV-Stand making sure it keeps its requirements that meets the hospital standards, which is our main purpose in this research. Our project is about a robot that autonomously follows patients assisting them providing the necessary help needed. The interaction between the robot and the patient should bring ease to him during any of the normal daily life activities. Our robot is designed to follow IR signals, a small leg-band attached to the patient transmits this signal. The main hindrance in this kind of work is that it might start operating when it detects any infrared signal. The infrared has to be unique for the robot to recognize it and start operating. The system works on a mechanism that requires every patient to have his own ID, this allows the robot to follow each patient by itself. This is all possible by using a Wii-camera which detects the IR signal providing its position relative to (x, y, z) system, and an IR reader reads the code being sent. This kind of work requires high attention to the smallest details, due to any possible threat to the patient's life. Keeping in mind, there should be a protective system that collects information about its surroundings while operating; making it capable of avoiding any obstacle in its range. This is done by using an ultrasonic sensor which provides information about the robot surroundings. All the processing is done on two Arduino micro-controllers communicating seamlessly.

2) **A CMUcam5 Computer Vision Based Arduino Wearable Navigation System for the Visually Impaired (Saurav Gandhi; Niketa Gandhi)** [3], This paper presents an engineering design that focuses on creating a wearable, assistive technology, for the visually impaired, that allows them to navigate their indoor surroundings. This paper discusses about a device on the Arduino platform, using an inexpensive computer vision camera, and vibration motors for

vibrotactile feedback, installed in a fanny pack worn around the waist. The test criteria was based off on whether the user is able to identify the obstacle, avoid the obstacle, reach the destination, and make it through the obstacle course without touching a single obstacle. Based on these actions, the success (%) was calculated for one, two and three obstacles. In conclusion, the goal of avoiding obstacles was achieved. It was found that the product was 87.5% successful with one obstacle, 85% successful with two obstacles, and 72.5% successful with three obstacles. The error count increased as the number of obstacles increased. It can be improved with a more sophisticated computer vision camera, with the capability to determine the depth of objects. Additionally, there are numerous qualitative learnings related to the pace of walking, the distance between objects, and external light conditions. Improving these components could make the device safer, and more apt for mass production.

3) **IoT-Based Smart Shopping Cart Using Radio Frequency Identification(Mobeen Shahroz; Muhammad Faheem Mushtaq; Maqsood Ahmad) [1],** The modern age of technology in which most of the customer needs to wait in the supermarket for shopping because it is a highly time-consuming process. A huge crowd in the supermarket at the time of discount offers or weekends makes trouble to wait in long queues because of a barcode-based billing process. In this regard, the Internet of Things (IoT) based Smart Shopping Cart is proposed which consists of Radio Frequency Identification (RFID) sensors, Arduino microcontroller, Bluetooth module, and Mobile application. RFID sensors depend on wireless communication. One part is the RFID tag attached to each product and the other is RFID reader that reads the product information efficiently. After this, each product information shows in the Mobile application. The customer easily manages the shopping list in Mobile application according to preferences. Then shopping information sends to the server wirelessly and automatically generates billing. This experimental prototype is designed to eliminate time-consuming shopping process and quality of services issues. The proposed system can easily be implemented and tested at a commercial scale under the real scenario in the future. That is why the proposed model is more competitive as compared to others.

4) **Personal Assistant Robot (Finds Objects and Follows You) using Raspberry Pi** (Anurag Sinha) [2], This robot is built around the ever popular Raspberry pi, the incredibly

powerful RoboClaw motor controller, and the common Rover 5 robot platform. Furthermore, all the additional physical parts are 3D printed. This robot also uses the Tensor-flow, USB Coral accelerator to speed up the Raspberry Pi's slow object detection. This particular robot combines Machine Learning with Computer Vision. This means it passes the webcam stream through a machine-learnt model in order to detect objects in the frame. For example, based on the 'average cup model' ML example, the computer can look at the camera's video frame and try to fit the average cup image on to your new image.

Table 2.1 Comparative Study of Existing Models

SNO	YEAR	AUTHOR	TITLE	ADVANTAGES	DISADVANTAGES
1	2020	Mobeen Shahroz; Muhammad Faheem Mushtaq; Maqsood Ahmad	IoT-Based Smart Shopping Cart Using Radio Frequency Identification	Technology is advance and cheap to produce.	It is very difficult to produce many Radio Frequency signals for each device in large scale.
2	2020	Saral Taya	Personal Assistant Robot (Finds Objects and Follows You) using Raspberry Pi	Very fast and also fully automated to do many things like voice recognition, face recognition and object pick and drop	Used Raspberry-Pi and Google Coral Accelerator which is very costly.
3	2018	Saurav Gandhi; Niketa Gandhi	A CMUcam5 Computer Vision Based Arduino Wearable Navigation System for the Visually Impaired	Uses pixycam2 detect and RFID to navigate to the location	Fast but costly to use both RFID and pixycam2 to navigate the way and cannot perform face or object recognition properly
4	2017	Ali Kassem Merhi; Murtada Mohammad Hasan; Samih Abdul-Nabi; Ali Bazzi	Arduino based human-following IV stand	This IoT device uses Ultra-Sonic sounds to detect objects and navigate the way. It's easy and very cheap to build this robot	It's is a very basic indoor robot which can not be used at industrial level. RDIF is better than Ultra-Sonic sounds.

3. DESIGN AND IMPLEMENTATION FOR PERSONAL ASSISTANT ROBOT USING ARDUINO

3.1 An approach for Personal Assistant Robot

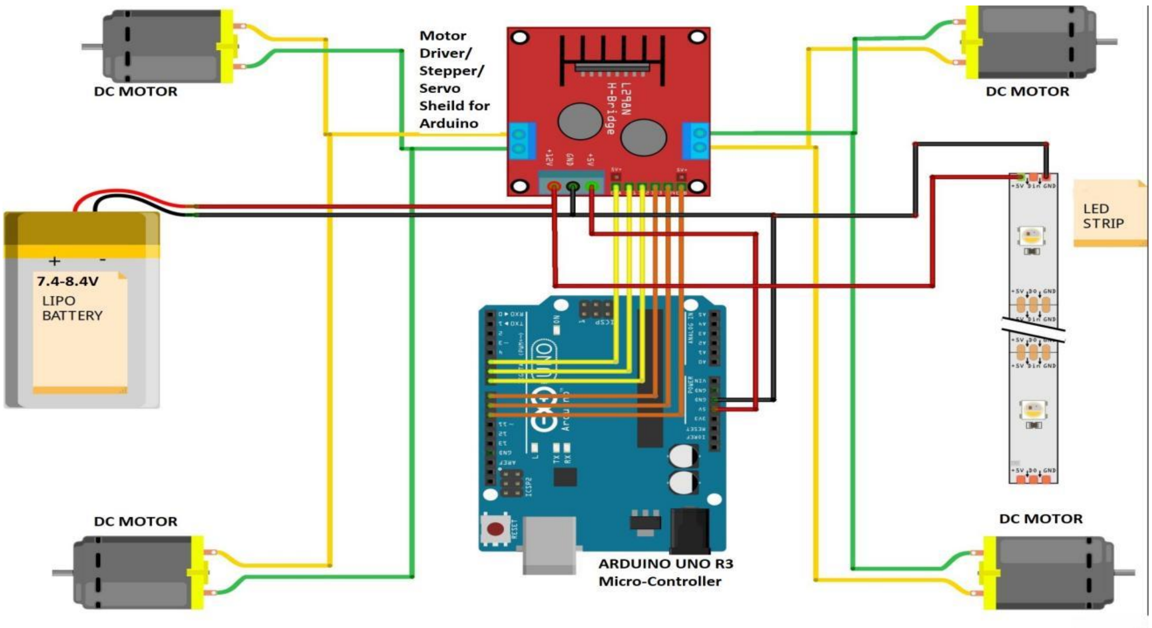


Figure 3.1: Circuit Diagram

The above figure 3.1 represents the circuit diagram (electrical diagram, elementary diagram, electronic schematic) which is a graphical representation of an electrical circuit. A pictorial circuit diagram uses simple images of components, while a schematic diagram shows the components and interconnections of the circuit using standardized symbolic representations. The presentation of the interconnections between circuit components in the schematic diagram does not necessarily correspond to the physical arrangements in the finished device[12].

The 7.4-8.4Volts battery is connected the Motor-Driver which then distributes the current parallel to other parts, such as Arduino UNO R3 chip and four DC-Motors. The Motor-Driver acts as a stepper to reduce the voltage to 5volts and passes it the Arduino. Arduino uses 3.5-5Volts according to the load it's taking. The Arduino has 36,000 Bytes memory which contains the uploaded code in it and Arduino's micro-controller processes and returns command to Motor-Driver which then controls the DC-Motors.

Motor-Driver acts as a shield and stepper for Arduino UNO R3 and is connected to the 5volts and ground(GND) on the right side of Arduino UNO R3 board which gives power to Arduino and 0-13 digital signals on the other side. Then the Motor-Driver will fifth and sixth digital pins to take the output from Arduino and other pins can be used for other purposes through Motor-Driver.

3.1.1 UML Diagram

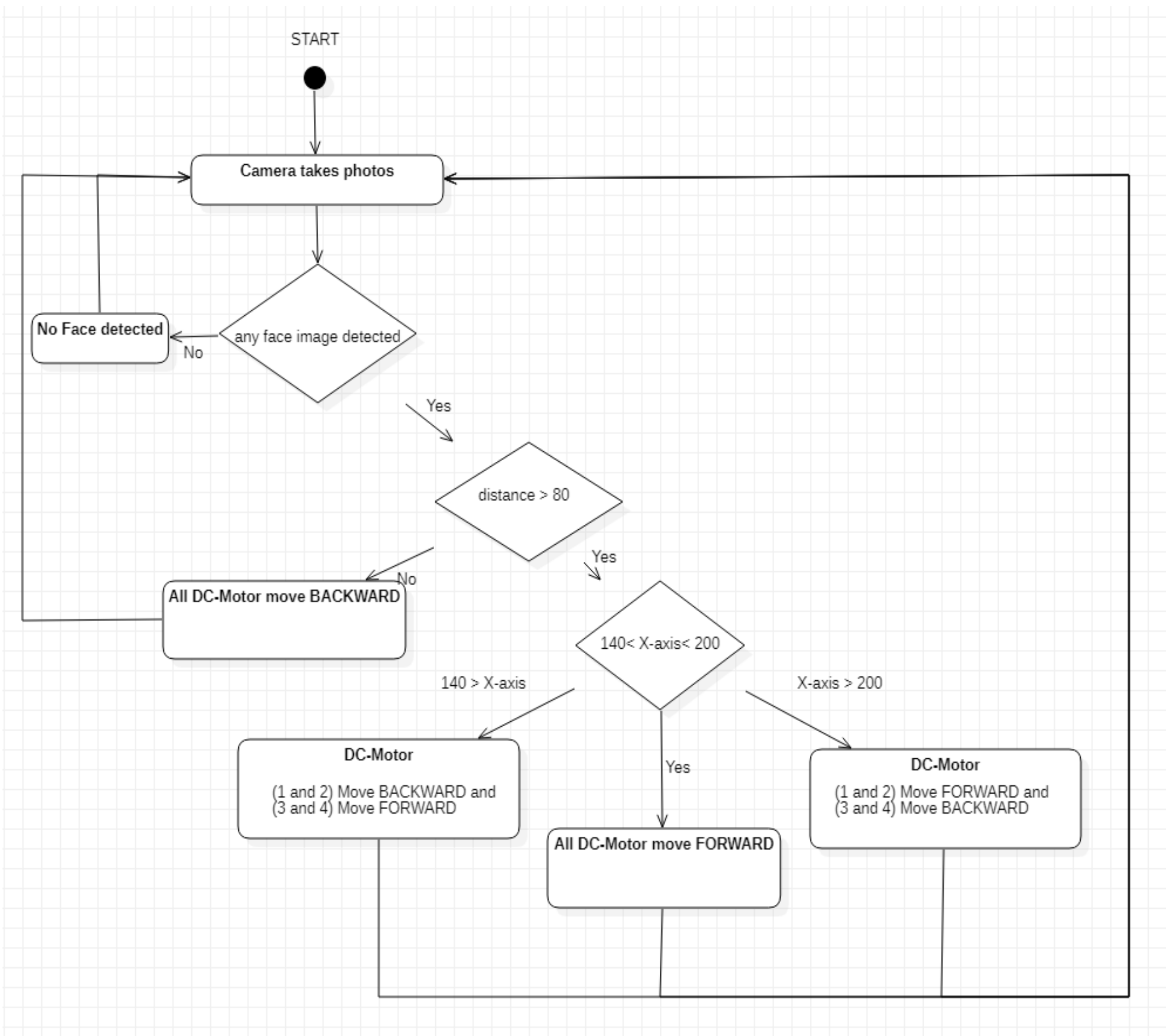


Figure 3.2: UML diagram for Personal Assistant Robot

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc as represented in figure 3.2.

3.1.2 Connecting Camera to Arduino

The camera used in this project is HuskyLens which has Kendryte K210 processor With high computing power at 1TOPS, K210 consumes only 0.3W while other typical devices consume 1W. K210 comes with SRAM and offline database, which allows complete the offline data processing and storage on the device. The camera has many detection and tracking function in its memory and can be accessed by including some huskylens libraries in the code.

Huskylens take Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. The longer the switch is on compared to the off periods, the higher the total power supplied to the load. Along with maximum power point tracking (MPPT), it is one of the primary methods of reducing the output of solar panels to that which can be utilized by a battery. PWM is particularly suited for running inertial loads such as motors, which are not as easily affected by this discrete switching, because their inertia causes them to react slowly. The PWM switching frequency has to be high enough not to affect the load, which is to say that the resultant waveform perceived by the load must be as smooth as possible.

The signal wires of HuskyLens are connected to the Arduino UNO board or the extension in Motor-Driver. The red wire is connected to 5Volts and black wire is connected to ground(GND). The other two wires are transmitter and receiver which is connected to 10, 11 digital signal pin on Arduino UNO R3 board as shown in the figure 3.3 below.

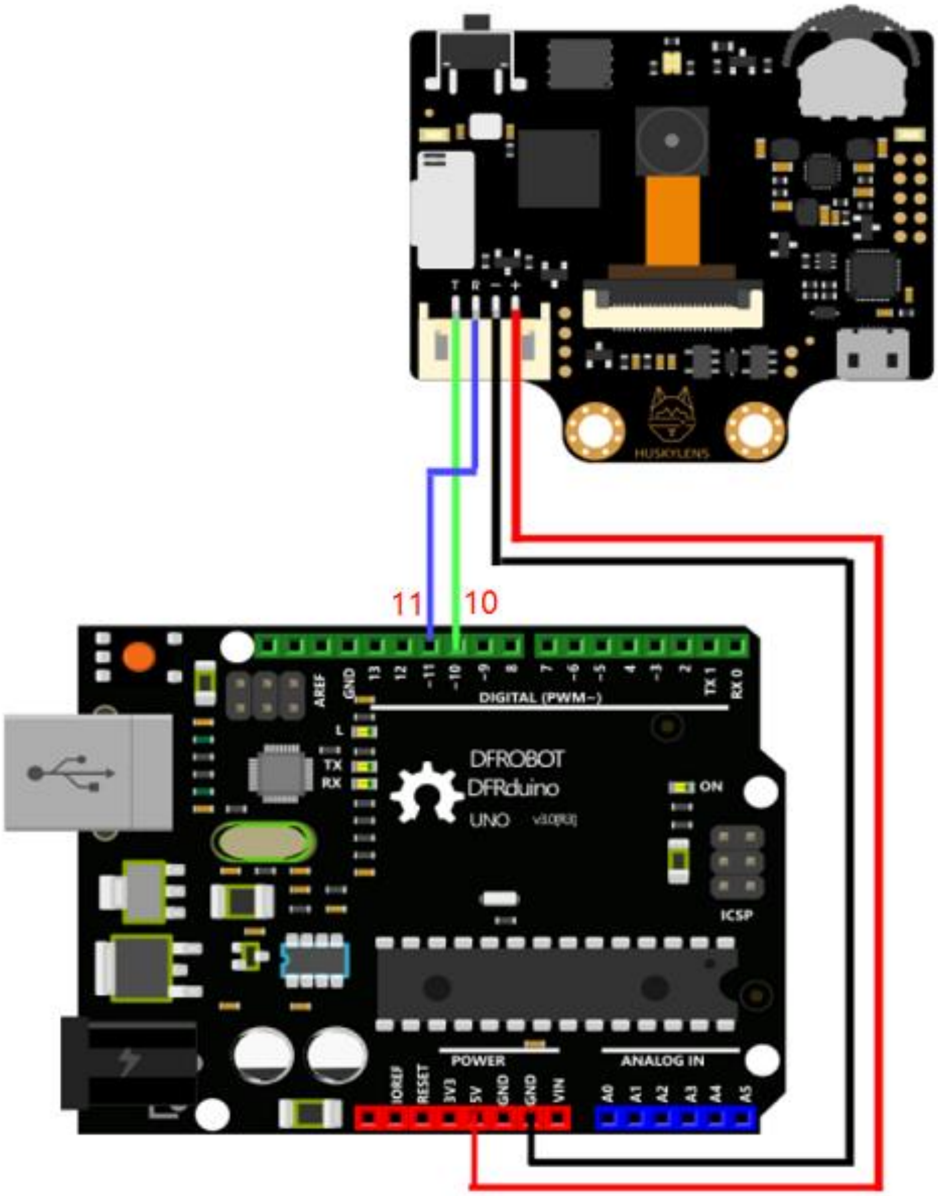


Figure 3.3: Connecting camera to Arduino.

3.2 Algorithms used for Personal Assistant robot

3.2.1 Deep-Face algorithm

The personal assistant robot uses Deep-face algorithm for face detection. Deep Learning is especially suitable for dealing with large training sets and has been recently applied in diverse domains such as vision, speech and language modeling. The Facebook team adopted this tool to apply to their face verification algorithm in lieu of well engineered features which is common in majority of contributions in this field. DeepFace is trained on a large dataset of faces, an identity labeled dataset of four million facial images belonging to more than 4,000 identities, acquired from a population vastly different than the one used to construct the evaluation benchmarks[13].

In modern face recognition there are 4 steps:

1. Detect
2. Align
3. Represent
4. Classify

The DeepFace algorithm, first aligns a face so that the person in the picture faces forward, using a 3-D model of an "average" forward-looking face. Then it uses the deep learning to find a numerical description of the forward-looking face. If two different images have similar enough descriptions, DeepFace decides they must show the same face.

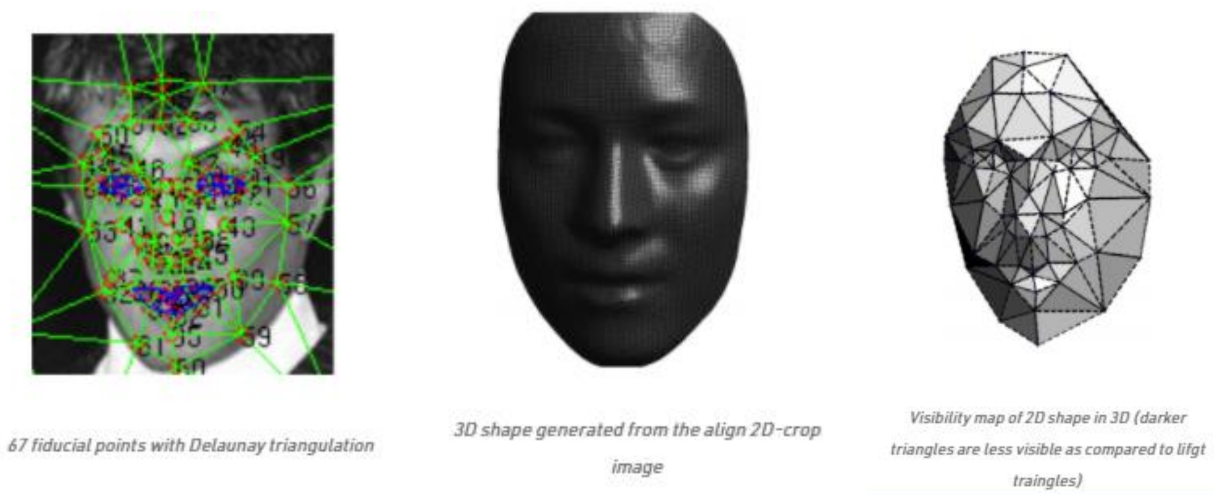
3D Alignment Technique- In order to align faces, DeepFace uses a generic 3D model wherein 2D images are cropped as 3D versions. The 3D image has 67 fiducial points. After the image has been warped, there are 67 anchor points manually placed on the image to match the 67 fiducial points. A 3D-to-2D camera is then fitted that minimizes losses. Because 3D detected points on the contour of the face can be inaccurate, this step is important.

The Weighted distance similarity is defined as :

$$\chi^2_{f_1,f_2} = \sum_i \omega_i \frac{(f_1[i] - f_2[i])^2}{f_1[i] + f_2[i]}$$

Figure 3.4: Distance between face and camera calculation.

where f_1 and f_2 are face representation vector and \omega is weight we can learn using linear SVM.



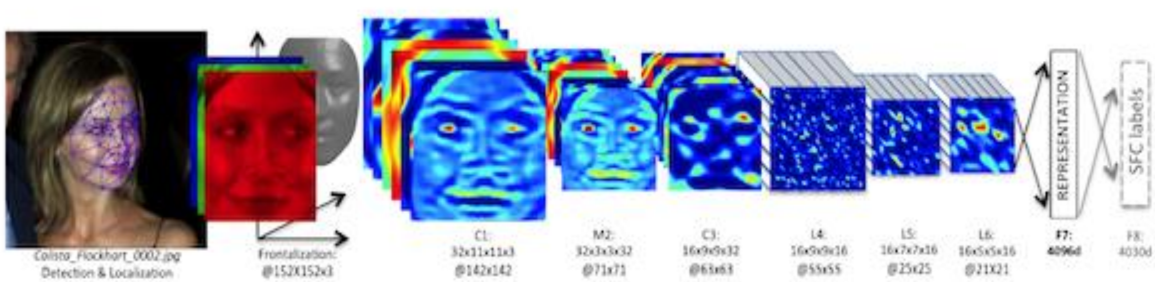


Figure 3.4: Deep-face algorithm processing .

3.3 Software and Hardware component for IoT Personal Assistant Robot

Software Components-

1. **Arduino IDE** - The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.The source code for the IDE is released under the GNU General Public License, version. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program Arduino to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, Arduino is used as the uploading tool to flash the user code onto official Arduino boards[12].

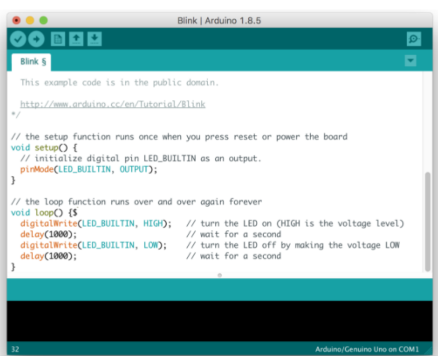


Figure 3.5: Arduino IDE.

2. Programming Languages - The programming languages used are python and C language as they are most common and easy to use.

Hardware Components-

1. **Arduino UNO Rev3 board** - Arduino Uno is a micro-controller board based on the ATmega328P chip. It has 14 digital input/output pins of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the micro-controller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards in figure 3.6 as shown below.

Figure 3.6: Arduino Uno Rev3 board

Arduino Uno board can be used for controlling many hardware parts such as DC-motors, LED lights, Ultra-Sonic sensor, RFID signals and more.

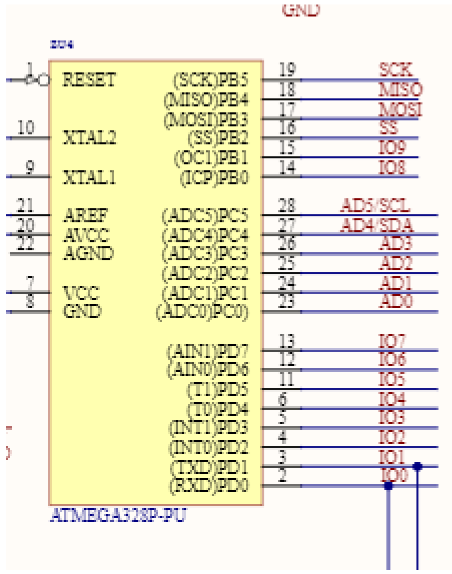


Figure 3.7: ATmega328 is a single-chip micro-controller

The above figure 3.7 is a ATmega328P is a single-chip micro-controller which is 8-bit AVR RISC-based micro-controller combines 32 KB ISP flash memory with read-while-write capabilities, 1 KB EEPROM, 2 KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter , programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz.

2. **L29-3D Motor Driver -** The L293D Motor Driver/Servo Shield for Arduino is probably one of the most versatile on the market and features 2 servo and 4 motor connectors for DC or stepper motors. That makes it a great shield for any robotic project. This Arduino compatible motor Driver shield is a full-featured product that it can be used to drive 4 DC motor or two 4-wire steppers and two 5v servos. It drives the DC motor and stepper with the L293D, and it drives the servo with Arduino pin9 and pin10. The shield contains two L293D motor drivers and one 74HC595 shift register. The shift register expands 3 pins of the Arduino to 8 pins to control the direction of the motor drivers. The output enables the L293D is directly connected to the PWM outputs of the Arduino.

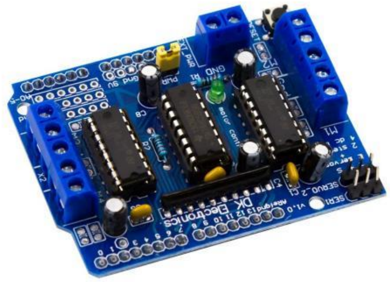


Figure 3.7: The L293D Motor Driver/Servo Shield for Arduino.

3. **HuskyLens Camera -** The camera used in this project is HuskyLens which has Kendryte K210 processor With high computing power at 1TOPS, K210 consumes only 0.3W while other typical devices consume 1W. K210 comes with SRAM and offline database, which allows complete the offline data processing and storage on the device. The camera has many detection and tracking function in its memory and can be accessed by including some huskylens libraries in the code.Huskylens take Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. The longer the switch is on compared to the off periods, the higher the total power supplied to the load. Along with maximum power point tracking (MPPT), it is one of the primary methods of reducing the output of solar panels to that which can be utilized by a battery. PWM is particularly suited for running inertial loads such as motors, which are not as easily affected by this discrete switching, because their inertia causes them to react slowly. The PWM switching frequency has to be high enough not to affect the load, which is to say that the resultant waveform perceived by the load must be as smooth as possible. The Camera needs high brightness to detect faces in real time as it has only 2 mega-pixe[5][6][7][9].

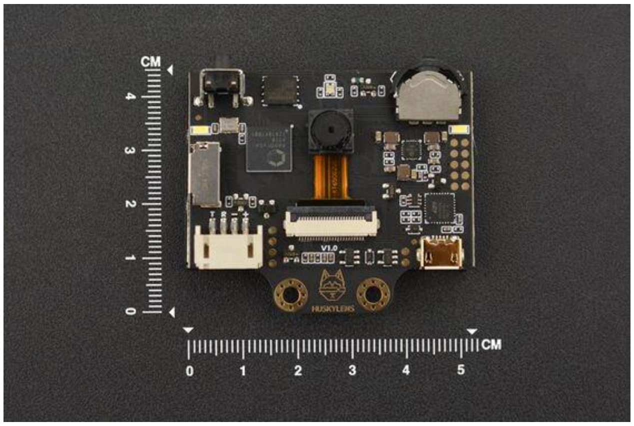


Figure 3.8: Husky Lens.

4. **4-DC Motors and Wheels -** The Operating Voltage of the motor is 3V - 12V DCRPM and load current is 40 - 80 mA Output. There are two motors front and 2 motors back of the chassis.

Figure 3.7: DC-Motors and Wheels.

5. **7.4V-8.4V battery** - Orange 850mAh 2S 30C/60C Lithium Polymer Battery Pack (Lipo) battery are known for performance, reliability, and price. It’s no surprise to us that Orange Lithium polymer packs are the go-to pack for those in the know. The Orange batteries deliver the full rated capacity at a price everyone can afford.Orange 850mAh 2S 30C/60C Lithium Polymer Battery Pack (Lipo) battery are equipped with heavy-duty discharge leads to minimize resistance and sustain high current loads. Orange batteries stand up to the punishing

extremes of aerobatic flight and RC vehicles. All Orange Lithium Polymer batteries packs are assembled using IR match cells.

Figure 3.9:Orange 850mAh 2S 30C/60C Lithium Polymer Battery (Lipo) .

6. **Chassis for the Robot -** In this project the Chassis is made by 3D printing it to make the robot correct size and also look good.

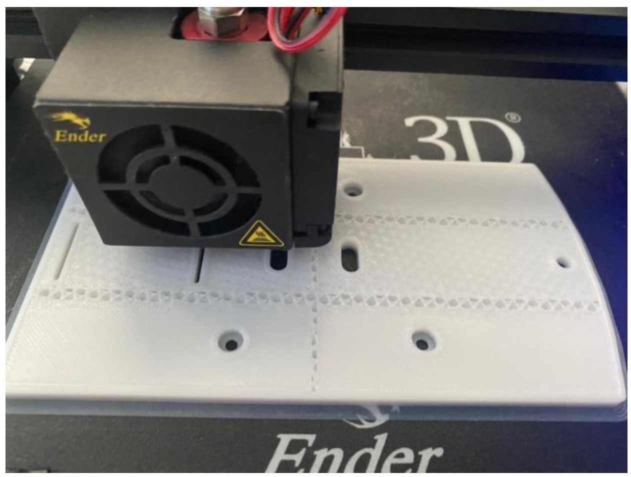


Figure 3.10: Chassis

7. **Multi-meter** - A digital multi-meter is a test tool used to measure two or more electrical values—principally voltage (volts), current (amps) and resistance (ohms). It is a standard diagnostic tool for technicians in the electrical/electronic industries.

Figure 3.10: Multi-meter.

8. **Other parts -** Other parts of the robot such as DPDT switch, jumper wires, 5Amp wires, battery holder, battery charger and dis-charger and battery connector are some important parts of the robot. The jumper wires and 5Amp wires are used to connect one part of the robot to other part. Battery charger and dis-charger is used to charge and dis-charge the battery. The battery holder holds the battery on the chassis and battery connector is used to connect the battery to the switch.

Figure 3.11: Jumper wires, Imax B6 battery charger and dis-charger, DPDT switch.

4. RESULTS

Testing is a very important module in the software and hardware development to verify, validate and provide quality and service for different components of the project. It is used to minimize the risks by efficient use of resources in the development life cycle. This module can be employed at any point of the development process. It is efficient for the testing phase to be implemented at initial level to get good results[11].

4.1 Types of Testing in IoT

1. **Usability Testing:**
There are so many devices of different shape and form factors are used by the users. Moreover, the perception also varies from one user to other. That's why checking usability of the system is very important in IoT testing. The device should be user friendly.

2. **Compatibility Testing:**
There are lots of devices which can be connected though IOT system. These devices have varied software and hardware configuration. Therefore, the possible combination are huge. As a result, checking the compatibility in IOT system is important. As there are many devices, the project use specifically dedicated parts for the Arduino like L29-3D motor driver and Huskylens for improvement of performance.

3. **Reliability and Scalability Testing:**
Reliability and Scalability is important for building an IOT test environment which involves simulation of sensors by utilizing virtualization tools and technologies. The parts used in the device are cutting-edge technologies for good performance and can connect to a wide range of electronic parts such as ultra-sonic sensor, IFRD sensor, Wi-Fi chip and etc. The size of the device can be easily scalable as the main processing part of the device is same and only the hardware parts are changed. This device is capable of performing various other tasks on changing few things and can be implement on industrial level.

4. **Performance Testing:**
Performance testing is important to create strategic approach for developing and implementing an IOT testing plan. The parts used in the device are cutting-edge technologies for good performance. The camera detects the faces easily by using deep-face algorithms and sends to Arduino which is fast in controlling the electrical parts which are responsible for moving the device.

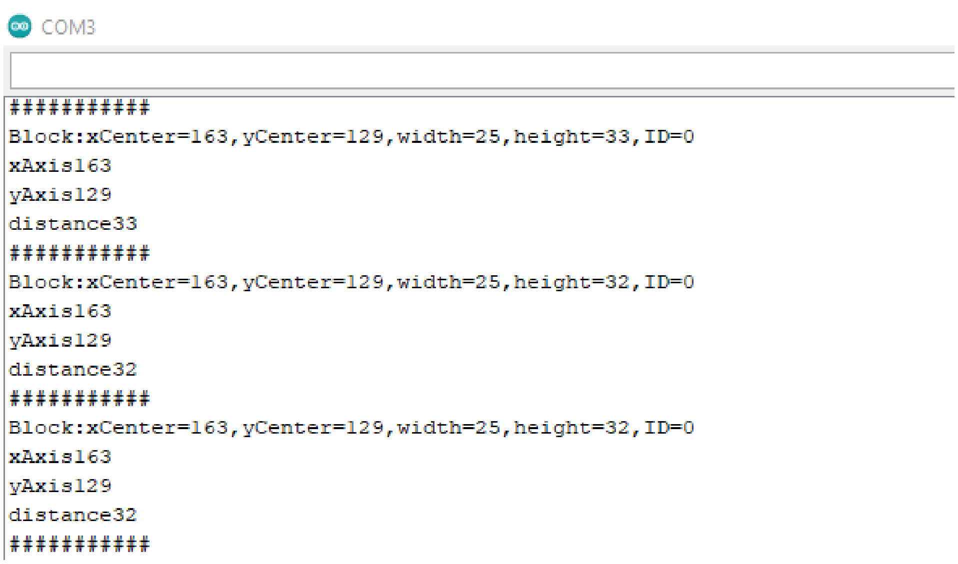
4.2 Testing the performance of IoT parts

1. **Detecting Faces in a Live Video:**

The Huskylens is uses deep-face algorithm which can be controlled by uploading some libraries into Arduino. The deep-face algorithm uses very less computation power for the face recognition in an image with respect to other artificial intelligence algorithms like CNN,ANN and etc. So, the deep-face algorithm can perform at higher speeds then other algorithms with limited resources and can process up to 30 frames per second in a video with the powerful processor Kendryte K210 in it [2][6][8].

Figure 4.1: HuskyLens is detecting the face.

When the distance between Camera and the face is less than 8cm or (80mm) then the robot moves backward adjusting the distance between them. The image below Figure 4.2 shows the x-axis, y-axis and distance parameters sent from the camera to the Arduino.



```
COM3
###########
Block:xCenter=163,yCenter=129,width=25,height=33,ID=0
xAxis163
yAxis129
distance33
###########
Block:xCenter=163,yCenter=129,width=25,height=32,ID=0
xAxis163
yAxis129
distance32
###########
Block:xCenter=163,yCenter=129,width=25,height=32,ID=0
xAxis163
yAxis129
distance32
###########
```

Figure 4.2: Signals sent from HuskyLens to Arduino.

If the distance between Camera and The face is greater than 8cm or (80mm) then the robot moves forward. The image below Figure 4.3 shows the x-axis, y-axis and distance parameters sent from the camera to the Arduino.

```
COM3

###########
Block:xCenter=89,yCenter=85,width=65,height=87,ID=0
xAxis89
yAxis85
distance87
###########
Block:xCenter=87,yCenter=86,width=65,height=110,ID=0
xAxis87
yAxis86
distance110
###########
Block:xCenter=87,yCenter=86,width=65,height=110,ID=0
xAxis87
yAxis86
distance110
###########
Block:xCenter=84,yCenter=88,width=65,height=110,ID=0
xAxis84
yAxis88
distance110
###########
```

Figure 4.3: Signals sent from HuskyLens to Arduino.

When the face is detected but is left side to the camera or less than 140-degrees on the x-axis then the robot moves to left. When the face is detected but is right side to the camera or greater than 200-degrees on the x-axis then the robot moves to right. And the center of the robot or camera is fixed at 170-degrees[13][14].

2. **Testing Arduino and L29-3D Motor Driver:**

Connecting other components to Arduino board with jumper wires and can operate them through uploading code and some libraries like “HUSKLENS.h”, “SoftwareSerial” and “AFMotors”.

The L29-3D motor driver and other parts connection to the Arduino is shown in figure 3.1 .

These libraries are used for:

a. HUSKLENS.h is used for receiving and transmitting data and also control the algorithms used by it.

b. SoftwareSerial is used to connect to the serial monitor which shows output which contains the signals transmission between Arduino and other parts.

c. AFMotors are used for specifically control each motor according to the need.

```
#include "HUSKYLENS.h"
#include "SoftwareSerial.h"
#include <AFMotor.h>

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

HUSKYLENS huskylens;
SoftwareSerial mySerial(10, 9); // RX, TX
//HUSKYLENS green line >> Pin 10; blue line >> Pin 11
void printResult(HUSKYLENSResult result);

void setup() {
    Serial.begin(115200);
    mySerial.begin(9600);
    while (!huskylens.begin(mySerial))
```

Figure 4.4: Importing libraries.

In this project the serial protocol type used is “9600” baud for receiving and transmitting data. and output serial monitor is set to “115200” baud to show the output signals. Baud is a unit of transmission speed equal to the number of times a signal changes state per second. For signals with only two possible states one baud is equivalent to one bit per second as shown in figure 4.4 [10][11][12].

Movement of the robot is controlled by controlling the rotation and speed of the motors. Arduino uses AFMotor libraries to send signals to the motors very precisely to each motor as in figure .

- When the robot want to move forward all the four motors move forward.
- When the robot wants to move backward all the four motors move backwards.

- When the robot wants to move to the left then motor-2 and motor-4 move backwards and motor-1 and motor-3 move forwards.
- When the robot wants to move to the left then motor-2 and motor-4 move forwards and motor-1 and motor-3 move backwards.

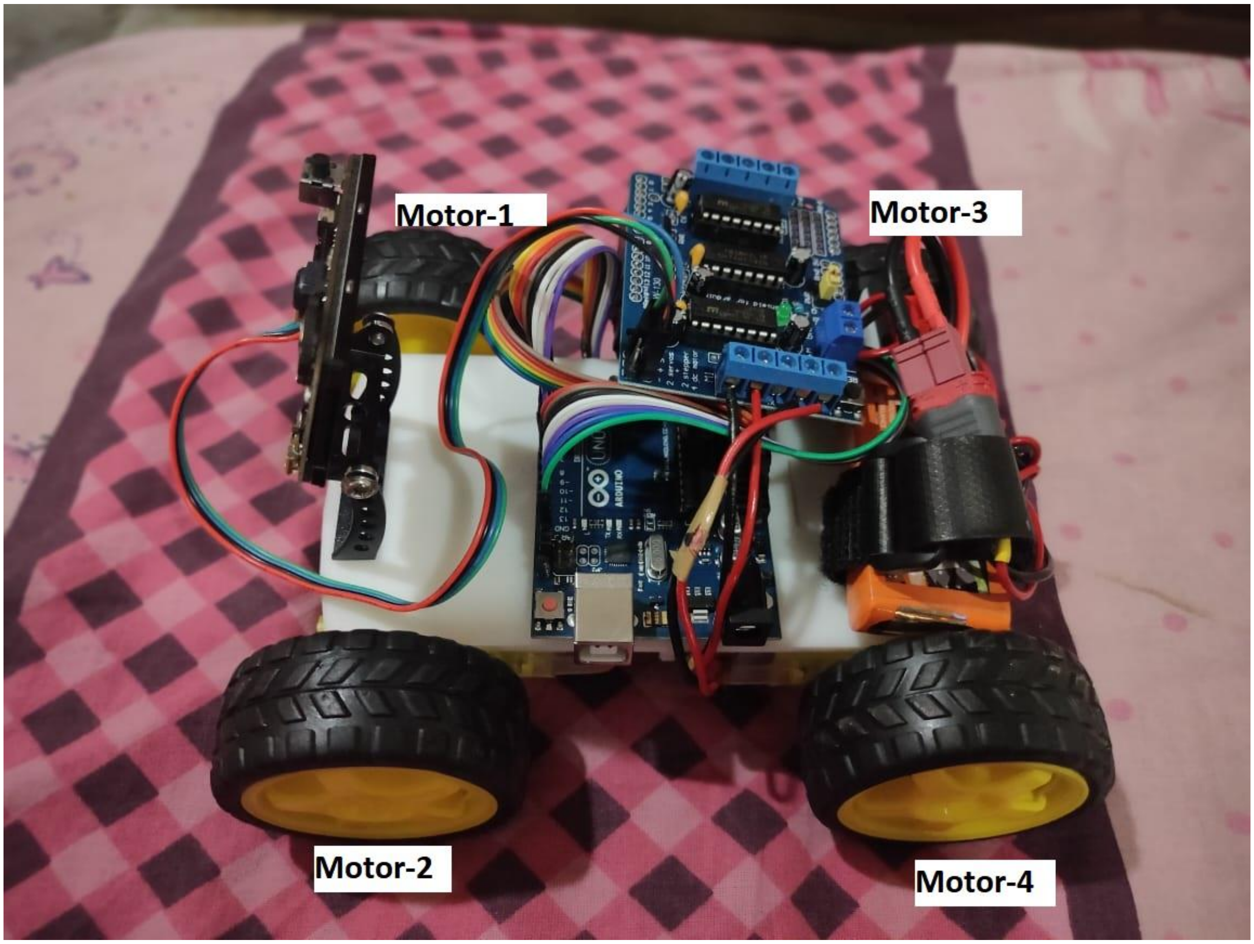


Figure 4.5: Numbering motors in the robot

3. **Regulating Voltage and Current in the Robot :**

All the connections in the robot are done by 5 amperes wires or jumper wires. The voltage in the circuit should be greater than 7.4 volts and less than 8.4 volts. If the voltage is less than 7.4 volts then the battery gets damaged and if it is more than 8.4 volts then the hardware components gets damaged. So, the voltage should be maintained at 7.4volts to 8.volts

4.3 Working Model of the Robot

Figure 4.5: Real Time showcase of the Personal Assistant Robot using Arduino.

This project has successfully built a personal assistant robot using Arduino as shown above in figure 4.6. Which is capable of following humans by recognizing their faces and this robot can move forward, backward, left and right in all directions. This robot can detect objects but has been designed to ignore them and only follow human beings. The robot truly is a personal assistant and its powered by Arduino UNO and a camera with a dedicated Kendryte K210 processor inside. The most important factor for an autonomous robot is to have continuous interaction with the person.

5. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Internet of Things is the concept in which the virtual world of information technology is connected to things in the real world. Modern technological advancements in processors, micro- controllers, cameras, sensors and RFID play an important role in improving the quality of human life. Combining artificial intelligence with IoT is leading the technological revolution of our present times and an attempt has being made to capture the same in the project. This project has implemented these technologies with an aim to assist human beings and pave a better way for the future.

5.2 Future Scope

In the near future, This robot can use Computer Vision models to dynamically follow humans, Signs and particular objects around. The robot truly is a personal assistant and its powered by Arduino UNO and has IR transceiver, Ultrasonic distance sensor, camera and some other parts.

This robot can be used in different ways for example:

1) We can use it in the hospitals, malls and in other activities also.

2) It can perform different tasks like obstacle detection and avoiding etc.,

3) This is the best fit robot which can perform multiple tasks.

BIBLIOGRAPHY

[1] Mobeen Shahroz; Muhammad Faheem Mushtaq; Maqsood Ahmad, “IoT-Based Smart Shopping Cart Using Radio Frequency Identification”A review,” IEEE Sensors Journal, pp.1-9, 2020.

[2] Saral Taya, “Personal Assistant Robot (Finds Objects and Follows You) using Raspberry pi”, An article from Instructables Circuits, 2020.

[3] Saurav Gandhi; Niketa Gandhi, “A CMUcam5 Computer Vision Based Arduino Wearable Navigation System for the Visually Impaired”, International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2-7, 2018.

[4] Ali Kassem Merhi; Murtada Mohammad Hasan; Samih Abdul-Nabi; Ali Bazzi, “Arduino based human-following IV stand ”, A review,” IEEE Sensors Journal, pp. 2-4, 2017.

[5]M. M. Islam, M. Sheikh Sadi, K. Z. Zamli, and M. M. Ahmed, “Developing walking assistants for visually impaired people:IEEE Sensors Journal, vol. 19, no. 8, pp. 2814-2828, 2019.

[6] R. Tapu, B. Mocanu, and T. Zaharia, “Wearable assistive devices for visually impaired: A state of the art survey,” Pattern Recognition Letters, 2018.

[7] M. Shih, Y. Chen, C. Tung, C. Sun, C. Cheng, L. Chan, S. Varadarajan, and M. Sun, “Dlwv2: A deep learning-based wearable vision-system with vibrotactile-feedback for visually impaired people to reach objects,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1-9.

[8] L. Gonzalez-Delgado, L. Serpa-Andrade, K. Calle-Urgilez, A. Guzhnay Lucero, V. Robles-Bykbaev, and M. Mena-Salcedo, “A low-cost wearable support system for visually disabled

people,” in 2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), 2016, pp. 1-5.

[9] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia, “A smartphone based obstacle detection and classification system for assisting visually impaired people,” in 2013 IEEE International Conference on Computer Vision Workshops, 2013, pp. 444-451.

[10] P. Rohit, M. S. Vinay Prasad, S. J. Ranganatha Gowda, D. R. Krishna Raju, and I. Quadri, “Image recognition based smart aid for visually challenged people,” in 2019 International Conference on Communication and Electronics Systems (ICCES), 2019, pp. 1058-1063.

[11] Samuel Greengard, “The Internet of Things”, The MIT Press Essential Knowledge series, pp.34-177, 2015

[12] Simon Monk, “Programming Arduino: Getting Started With Sketches (second edition)”, pp. 21-142, 2012.

[13] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, Lior Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification" University of Toronto, pp. 1-7, 2017.

[14] Gary Bradski and Adrian Kaebler, “Learning Computer Vision with the OpenCV Library”, OpenCV textbook, pp. 16-516, 2008.

APPENDIX

```
/* Code should be executed in Arduino IDE with the computer connected to Arduino UNO R3 board using USB cable  */

 /*  HUSKLENS.h is used for receiving and transmitting data and also control the algorithms used by it. */
#include "HUSKYLENS.h"

 /* SoftwareSerial is used to connect to the serial monitor which shows output which contains the signals transmission between Arduino and other parts.  */
#include "SoftwareSerial.h"

/* AFMotors are used for specifically control each motor according to the need.   */
#include <AFMotor.h>

AF_DCMotor motor1(1, MOTOR12_1KHZ);  /* Assigning each motor  */
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

HUSKYLENS huskylens;
SoftwareSerial mySerial(10, 9); // RX, TX
//HUSKYLENS green line >> Pin 10; blue line >> Pin 11; Transmitting and Receiving signals
void printResult(HUSKYLENSResult result);

/* Setting up serial number between Arduino UNO and HuskyLens   */
void setup() {
   Serial.begin(115200);
   mySerial.begin(9600);
   while (!huskylens.begin(mySerial))
```

```
  {
    Serial.println(F("Begin failed!"));
    Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS (General
Settings>>Protocol Type>>Serial 9600)"));
    Serial.println(F("2.Please recheck the connection."));
    delay(100);
  }
}

/* The code which is present in this function runs in a loop- loop() function*/
void loop() {
/* Checking serial connections */
  if (!huskylens.request()) Serial.println(F("Fail to request data from HUSKYLENS, recheck the
connection!"));
  else if(!huskylens.isLearned()) Serial.println(F("Nothing learned, press learn button on
HUSKYLENS to learn one!"));
  else if(!huskylens.available()) Serial.println(F("No block or arrow appears on the screen!"));
  else
  {
    Serial.println(F("###########"));
    if (huskylens.available())
    {
      HUSKYLENSResult result = huskylens.read();
      printResult(result);

/* Assigning a variables for the data received from HuskyLens */
      int xAxis = result.xCenter;
      int yAxis = result.yCenter;
      int distance = result.height;
```

```
  Serial.print("xAxis");
  Serial.println(xAxis);
  Serial.print("yAxis");
  Serial.println(yAxis);
  Serial.print("distance");
  Serial.println(distance);


 /* When the robot wants to move to the left then motor-2 and motor-4 move backwards and
motor-1 and motor-3 move forwards.*/
  if (xAxis < 140) {
    motor1.setSpeed(220);
    motor1.run(BACKWARD);
    motor2.setSpeed(220);
    motor2.run(FORWARD);
    motor3.setSpeed(220);
    motor3.run(FORWARD);
    motor4.setSpeed(220);
    motor4.run(BACKWARD);
    //delay(10);

/* When the robot wants to move to the right then motor-2 and motor-4 move forwards and motor-
1 and motor-3 move backwards.*/
  }else if (xAxis > 200) {
    motor1.setSpeed(220);
    motor1.run(FORWARD);
    motor2.setSpeed(220);
    motor2.run(BACKWARD);
    motor3.setSpeed(220);
    motor3.run(BACKWARD);
    motor4.setSpeed(220);
```

```
      motor4.run(FORWARD);
      //delay(10);

    }
/*   When the robot want to move forward all the four motors move forward. */
    else if (distance < 80) {
      motor1.setSpeed(200);
      motor1.run(FORWARD);
      motor2.setSpeed(200);
      motor2.run(BACKWARD);
      motor3.setSpeed(200);
      motor3.run(FORWARD);
      motor4.setSpeed(200);
      motor4.run(BACKWARD);
      //delay(10);


    }
/* When the robot wants to move backward all the four motors move backwards. */
else if (distance > 110) {
      motor1.setSpeed(200);
      motor1.run(BACKWARD);
      motor2.setSpeed(200);
      motor2.run(FORWARD);
      motor3.setSpeed(200);
      motor3.run(BACKWARD);
      motor4.setSpeed(200);
      motor4.run(FORWARD);
      //delay(10);

    }
```

```
/*  Robot is idel when it donot detect anything */
    else{
      motor1.setSpeed(0);
      motor1.run(RELEASE);
      motor2.setSpeed(0);
      motor2.run(RELEASE);
      motor3.setSpeed(0);
      motor3.run(RELEASE);
      motor4.setSpeed(0);
      motor4.run(RELEASE);
    }


  } else {
    Serial.println("vision body undetected.");
     motor1.setSpeed(0);
      motor1.run(RELEASE);
      motor2.setSpeed(0);
      motor2.run(RELEASE);
      motor3.setSpeed(0);
      motor3.run(RELEASE);
      motor4.setSpeed(0);
      motor4.run(RELEASE);
  }
        }



    }
```

```
/* The output signals returned by the robot */
/* The output signals can only be accessed in Arduino IDE- Serial Monitor */

void printResult(HUSKYLENSResult result){
   if (result.command == COMMAND_RETURN_BLOCK){

Serial.println(String()+F("Block:xCenter=")+result.xCenter+F(",yCenter=")+result.yCenter+F(",
width=")+result.width+F(",height=")+result.height+F(",ID=")+result.ID);
   }
   else if (result.command == COMMAND_RETURN_ARROW){

Serial.println(String()+F("Arrow:xOrigin=")+result.xOrigin+F(",yOrigin=")+result.yOrigin+F(",
xTarget=")+result.xTarget+F(",yTarget=")+result.yTarget+F(",ID=")+result.ID);
   }
   else{
      Serial.println("Object unknown!");

   }
}
```