

EYE CONTROLLED WHEEL CHAIR SYSTEM FOR PHYSICALLY CHALLENGED PEOPLE

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

MD.DHARIQ REFAI (37130255)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RA JIV GANDHI SALAI, CHENNAI - 600 119

MARCH 2021



SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in

**DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING**

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of MD.DHARIQ REFAI (37130255) who carried out the project entitled **“EYE CONTROLLED WHEEL CHAIR SYSTEM FOR PHYSICALLY CHALLENGED PEOPLE”** under our supervision from December 2020 to March 2021.

Internal Guide

Dr.S. KARTHIKEYAN

Associate professor, Department of ECE

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **MD.DHARIQ REFAI (37130255)**

hereby declare that the Project Report entitled “**EYE CONTROLLED WHEEL SYSTEM FOR PHYSICALLY CHALLENGED PEOPLE**” done by us under the guidance of **Dr.S .KARTHIKEYAN** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1)

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D. Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E., Ph.D. Head of the Department, Department of Electronics and Communication Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. S. KARTHIKEYAN ASSOCIATE PROFESSOR ,Department of Electronics and Communication Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the project work.

We wish to express our thanks to all teaching and Non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We express our gratitude to our parents for their constant encouragement and support for the completion of the project.

ABSTRACT

According to a new report prepared jointly by the World Health Organization and the World Bank, 15 percent of the world's population is disabled.

The use of powered wheelchairs with high navigational intelligence is one of the great steps towards the integration of severely physically disabled and mentally handicapped people.

Driving a wheelchair is a tedious task for severely handicapped persons, unless they use the tongue to control the joystick.

Simultaneously blind and paraplegic people deal with two problems, which creates uneasy situation for them, i.e locomotion and localization. Different systems are being developed to overcome the problems described above, allowing the end-user to perform safe movements and accomplish some daily life important tasks.

Our Robotic wheelchair uses eye blink and head tilt movement to steer the wheelchair. In addition, we can give more independence to the disabled person by using the same head-tilt movement to communicate with the devices in a room for example: a fan.

This communication is done using a RFtransmitter and receiver. Using this, the person can control various devices easily.

Keywords: Arduino UNO,IR sensor, Motor driver.

TABLE OF CONTENT

CONTENT	PAGE NO:
1. INTRODUCTION	1
2. LITERATURE SURVEY	1
3. MATERIALS AND METHODS	4
3.1 EXISTING SYSTEM	4
3.2 PROPOSED SYSTEM	4
3.3 BLOCK DIAGRAM	5
3.4 HARDWARE REQUIREMENTS	5
3.5 HARDWARE IMPLEMENTATION	6
4. HARDWARE DESIGN	10
4.1 ARDUINO	10
4.2 ARDUINO UNO AND ITS PROGRAMMING	13
4.2.1 ARDUINO SOFTWARE	19
4.3 EYE BLINK SENSOR	24
4.3.1 EYE ATTACHED TRACKING	27
4.3.2 OPTICAL TRACKING	28
4.3.3 ELECTRIC POTENTIAL MEASUREMENT	28
4.3.4 EYE TRACKING OF YOUNGER AND ELDERLY PEOPLE WHILE WALKING	31
4.3.5 THE FUTURE OF THE EYE TRACKING TECHNOLOGY	31
4.3.6 UBIQUITOUS EYE TRACKING	33
4.4 MOTOR DRIVER	35
4.4.1 APPLICATIONS	36
5. RESULT	37
6. CONCLUSION	37

LIST OF FIGURES

FIG NO.	NAME OF FIGURE	PAGE NO.
3.1	BLOCK DIAGRAM	5
3.2	BLOCK DIGRAM OF A TYPICAL EMBEDDED SYSTEM	7
3.3	GENERAL BLOCK DIAGRAM OF POWER SUPPLY UNIT	9
4.1	ARDUINO	14
4.2	OPEN ARDUINO	19
4.3	SELECT COM PORT	20
4.4	SELECT REQUIRED ARDUINO	21
4.5	SKETCH ARDUINO CODE	22
4.6	COMPILE AND EXECUTION	23
4.7	EYE TRACKING	30
4.8	IR SENSOR IN GOOGLES	34
4.9	MOTOR DRIVER	35
4.10	MOTOR APPLICATIONS	36
5.1	OUTPUT	37

CHAPTER 1

INTRODUCTION

Wheelchairs were designed with the idea to help handicapped people to move around and accomplish daily life tasks, but what use would it be if the person is severely handicapped? Many smart wheelchairs have been designed and with more and more advances in technologies newer and smarter wheelchairs are coming into the market to help the severely handicapped persons. Our wheelchair is also an attempt to make use of simple and easy movements of the eye and head to start, stop and move the wheelchair. In addition we also connect the various device in a room to the wheelchair and communicate. We are introducing the design and implementation of wheelchair controlled using eye movement. This wheelchair is easy for differently abled and paralyzed people to make their life more easier. The IR sensor is mounted on goggle for capturing the eye blink movement. This sensor captured his eye blink movement and this signal sends to motor driver circuit to move and stop the wheelchair. As per the counting of number of eye blink movement the circuit will perform different operations like left, right, forward, backward and stop.

CHAPTER 2

LITERATURE SURVEY

2.1 Voice Controlled Wheelchair:

This presents an automatic wheel chair using voice recognition. A voice controlled wheelchair makes it easy for physically disabled person who cannot control their movements of hands. The powered wheel chair depends on motors for locomotion and voice recognition for command. The circuit comprises of an Arduino, HM2007 Voice recognition module and Motors. The voice recognition module recognizes the command by the user and provides the corresponding coded data stored in the memory to ArduinoMicrocontroller. Arduino Microcontroller controls the locomotion accordingly. The wheelchair also has provision for joystick for physically disabled people Smart Electronic Wheelchair Using Arduino and Bluetooth Module who can move their hands.

2.2 Smart Electronic Wheelchair Using Arduino and Bluetooth Module:

The design of a smart, motorized, voice controlled wheelchair using embedded system. Proposed design supports voice activation system for physically differently abled persons incorporating manualoperation. This paper represents the "Voice-controlled Wheel chair" for the physically differently abled person where the voice command controls themovements of the wheelchair. The voice command is given through a cellular device having Bluetooth and the command is transferred and converted to string by the BT Voice Control for Arduino and is transferred to the Bluetooth Module SR-04connected to the Arduino board for the control of the Wheelchair. For example, when the user says „Go' then chair will move in forward direction and when he says „Back" then the chair will move in backward direction

and similarly „Left“, „Right“ for rotating it in left and right directions respectively and „Stop“ for making it stop. This system was designed and developed to save cost, time and energy of the patient. Ultrasonic sensor is also made a part of the design and it helps to detect obstacles lying ahead in the way of the wheelchair that can hinder the passage of the wheelchair.

2.3 Voice Controlled Wheelchair using Arduino:

The needs of many individuals with disabilities can be satisfied with traditional, manual or powered wheelchairs. A segment of the disabled community finds it difficult or impossible to use wheelchairs. There is extensive research on computer controlled chairs where sensors and intelligent control algorithms have been used to minimize the level of human intervention. Here I describe design of a smart, motorized, voice controlled wheelchair for physically disabled people by using embedded system. Proposed design supports voice activation system for physically disabled persons incorporating manual operation. Arduino microcontroller and speaker dependent voice recognition processor have been used to support the navigation of the wheel chair. The wheelchair does not respond to a false speech command. Depending on the direction selected on the joystick, microcontroller controls the wheelchair directions. This can be controlled through the voice commands also. By using ultrasonic sensors we can avoid obstacles.

2.4 Design and Development of Smart Wheel Chair using Voice Recognition and Head Motion

The challenging problem faced by the paralyzed people is their independent mobility. They need an external help to perform their daily activities. Electric wheelchairs are designed to aid paraplegics. Unfortunately, these cannot be used by persons with higher degree of impairment, such as quadriplegics, i.e. persons that, due to age or illness, cannot move any of the body parts, except of the head. The main objective of this project is to provide an automated system for disabled people. The wheel chair will work based on the head movement of the user. The recognized gestures are used to generate motion control commands to the controller so that it can control the motion of the wheel chair according to the user intention. Design and development of Head motion controlled wheelchair has been achieved using MEM sensors and microcontroller. The system is implemented practically and works well. The MEM Sensor senses the change in direction of head and accordingly the signal is given to microcontroller. Depending on the direction of the Acceleration, microcontroller controls the wheel chair directions like LEFT, RIGHT, FRONT, and BACK with the aid of DC motors.

2.5 Speech recognition based robotic system of wheelchair for disabled people

This system aims to design, assimilate, combine and test a motorized, speech recognized protocol of wheelchair. This idea helps physically disabled patient to

control the direction without any external manual force. There are many people who are unable to control the standard joystick interfaced wheelchair. speech controlled wheelchair can provide easy access uses raspberry pi for the speech recognition. In this prototype, Bluetooth interface is used between android phone and Raspberry Pi. Android phone is used which is connected to the HC-05 Bluetooth module via BT connection. HC-05 is the Bluetooth module which is connected to the Raspberry Pi. The objective is to design a system which will make the life of disable people easier. In this way, a navigation framework for wheelchairs in which integrates various alternative input interfaces where voice control is done with android Bluetooth Smartphone to them who cannot control the wheelchair with hand movement. This robotic system.

Chapter 3

MATERIALS AND METHODS

3.1 EXISTING SYSTEM:

- In the existing system we can controlled the wheel chair using eye blink sensor based on direction of the eye lid motion. The wheel chair can be started and stopped by the eye blink movements.
- Eye blink sensor senses whether eye is open or closed. The eye-blink sensor works by illuminating the eye and/or eyelid area with infrared light and then monitoring the changes in the reflected light using a phototransistor and differentiator circuit.
- When the eye can blink two time means wheel chair can move to forward. And three times means left turn four times means right turn.

Drawbacks:

- It is difficult.

3.2 PROPOSED SYSTEM:

- In this system we used eye blink sensor to control the wheel chair. Firstly we mapping the house or any area and then we give the command to the wheel chair by eye blink sensor and the controller.
- When the eye blink sensors detects one blink means it goes to move bed room and then it detects two blinks means it goes to move kitchen, three times means goes to move toilets and etc...
- The proposed system is very useful and no need to control the wheel chair as per the direction.

3.3 Block diagram:

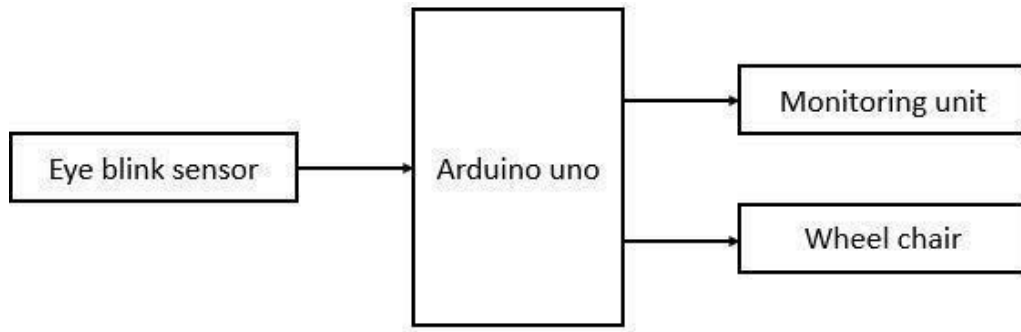


Fig 3.1 Block diagram

3.4 Hardware required:

- Arduino uno
- Eye blink sensor
- Monitoring unit
- Wheel chair

3.5 Hardware implementation:

EMBEDDED SYSTEMS

Overview of embedded systems

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and network mounted inside a large chassis or enclosure.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems — such as the operating systems and microprocessors which power them — but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

Embedded systems provide several functions

Monitor the environment; embedded systems read data from input sensors. This data is then processed and the results displayed in some format to a user or users
Control the environment; embedded systems generate and transmit commands for actuators.

Transform the information; embedded systems transform the data collected in some meaningful way, such as data compression/decompression.

Although interaction with the external world via sensors and actuators is an important aspect of embedded systems, these systems also provide functionality specific to their applications. Embedded systems typically execute applications such as control laws, finite state machines, and signal processing algorithms. These systems must also detect and react to faults in both the internal computing

environment as well as the surrounding electromechanical systems.

There are many categories of embedded systems, from communication devices to home appliances to control systems. Examples include;

Communication devices

e.g.: modems, cellular phones

e.g.: CD player, VCR, microwave oven

e.g.: Automobile anti-lock braking systems, robotics, and satellite control

Block diagram of an embedded system:

An embedded system usually contains an embedded processor. Many appliances that have a digital interface -- microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an operating system. Others are very specialized resulting in the entire logic being implemented as a single program. These systems are embedded into some device for some specific purpose other than to provide general purpose computing. A typical embedded system is shown in Fig 4.1

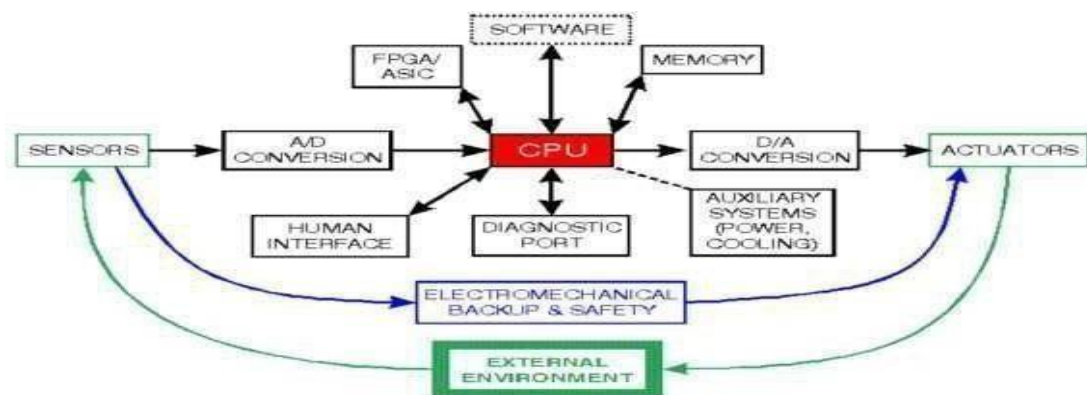


Fig 3.2 Block diagram of a typical embedded system

Characteristics of embedded systems

Embedded systems are characterized by a unique set of characteristics. Each of these characteristics imposed a specific set of design constraints on embedded systems designers. The challenge to designing embedded systems is to conform to the specific set of constraints for the application.

Application Specific Systems

Embedded systems are not general-purpose computers. Embedded system designs are optimized for a specific application. Many of the job characteristics are known before the hardware is designed. This allows the designer to focus on the specific design constraints of a well-defined application. As such, there is limited user reprogrammability. Some embedded systems, however, require the flexibility of reprogrammability. Programmable DSPs are common for such applications.

Reactive Systems

As mentioned earlier, a typical embedded systems model responds to the environment via sensors and control the environment using actuators. This requires embedded systems to run at the speed of the environment. This characteristic of embedded system is called “reactive”. Reactive computation means that the system (primarily the software component) executes in response to external events. External events can be either periodic or aperiodic. Periodic events make it easier to schedule processing to guarantee performance. Aperiodic events are harder to schedule. The maximum event arrival rate must be estimated in order to accommodate worst case situations. Most embedded systems have a significant reactive component. One of the biggest challenges for embedded system designers is performing an accurate worst case design analysis on systems with statistical performance characteristics (e.g., cache memory on a DSP or other embedded processor. Systems with this requirement must often design to worst case performance. But accurately predicting the worst case may be difficult on complicated architectures. This often leads to overly pessimistic estimates erring on the side of caution. Many embedded systems have a significant requirement for real time operation in order to meet external I/O and control stability requirements. Many real-time systems are also reactive systems.

Distributed Systems:

A common characteristic of an embedded system is one that consists of communicating processes executing on several CPUs or ASICs which are connected by communication links. The reason for this is economy. Economical 4 8-bit microcontrollers may be cheaper than a 32-bit processors. Even after adding the cost of the communication links, this approach may be preferable.

Heterogeneous Architecture

Embedded systems often are composed of heterogeneous architectures (Fig 1.2). They may contain different processors in the same system solution. They may also be mixed signal systems. The combination of I/O interfaces, local and remote memories, and sensors and actuators makes embedded system design truly unique. Embedded systems also have tight design constraints, and heterogeneity provides better design flexibility.

POWER SUPPLY

All electronic circuits work only in low DC voltage, so we need a power supply unit to provide the appropriate voltage supply for their proper functioning. This unit consists of transformer, rectifier, filter & regulator. AC voltage of typically 230volts rms is connected to a transformer voltage down to the level to the desired ac voltage. A diode rectifier that provides the full wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide dc voltage that not only has much less ripple voltage but also remains the same dc value even the dc voltage varies somewhat, or the load connected to the output dc voltages changes.

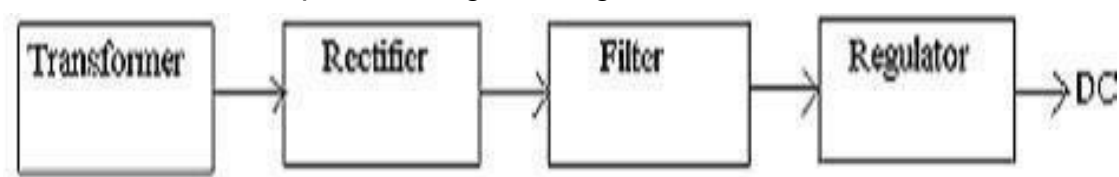


Fig 3.3 General Block of Power Supply Unit

Transformer:

A transformer is a static piece of which electric power in one circuit is transformed into electric power of same frequency in another circuit. It can raise or lower the voltage in the circuit, but with a corresponding decrease or increase in current. It works with the principle of mutual induction. In our project we are using a step down transformer to providing a necessary supply for the electronic circuits. Here we step down a 230volts ac into 12volts ac.

Rectifier:

A dc level obtained from a sinusoidal input can be improved 100% using a process called full wave rectification. Here in our project for full wave rectification we use bridge rectifier. From the basic bridge configuration we see that two diodes (say D2 & D3) are conducting while the other two diodes (D1 & D4) are in off state during the period $t = 0$ to $T/2$. Accordingly for the negative cycle of the input the conducting diodes are D1 & D4. Thus the polarity across the load is the same. In the bridge rectifier the diodes may be of variable types like 1N4001, 1N4003, 1N4004, 1N4005, 1N4007 etc... can be used. But here we use 1N4007, because it can withstand up to 1000v.

Filters:

In order to obtain a dc voltage of 0 Hz, we have to use a low pass filter. So that a capacitive filter circuit is used where a capacitor is connected at the rectifier output & a dc is obtained across it. The filtered waveform is essentially a dc voltage with negligible ripples & it is ultimately fed to the load.

Regulators:

The output voltage from the capacitor is more filtered & finally regulated. The voltage regulator is a device, which maintains the output voltage constant irrespective of the change in supply variations, load variations & temperature changes. Here we use fixed voltage regulator namely LM7805. The IC LM7805 is a +5v regulator which is used for microcontroller.

Chapter 4

HARDWARE DESIGN

4.1 ARDUINO

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

What does it do?

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a huge variety of Arduino-based projects.

For everything from robots and a heating pad hand warming blanket to honest fortune-telling machines, and even a Dungeons and Dragons dice-throwing gauntlet, the Arduino can be used as the brains behind almost any electronics project.

What's on the board?

There are many varieties of Arduino boards (explained on the next page) that can be used for different purposes. Some boards look a bit different from the one below, but most Arduino have the majority of these components in common.

Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labelled **(1)** and the barrel jack is labelled.

NOTE: Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labelled on the board and used for different functions.

- **GND (3):** Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5):** As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6):** The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- **Digital (7):** Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8):** You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM).
- We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).

- **AREF (9):** Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

4.2 ARDUINO UNO AND ITS PROGRAMMING

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer. The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

Overview

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

This guide covers the Arduino Uno board (Spark fun DEV-09950, \$29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the

This is what the Arduino board looks like.



FIG 4.1 ARDUINO

The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16MHz, which translates to about executing about 300,000 lines of C source code per second.

The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer.

Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button.

Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

SUMMARY:

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MH

Communication

Microcontrollers depend on a host computer for developing and compiling programs. The software used on the host computer is known as an integrated development environment, or IDE. For the Arduino, the development environment is based on the open source Processing platform(www.processing.org) which is described by its creators as a “programming language and environment for people who want to program images, animation, and interactions.” The Arduino programming language leverages an open source project known as Wiring (wiring.org.co). The Arduino language is based on good old-fashioned C. If you are unfamiliar with this language, don’t worry; it’s not hard to learn, and the Arduino IDE provides some feedback when you make mistakes in your programs.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins

0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

As you go through the list of programming statements available in the Arduino IDE (choose Help->Reference), you might think there isn’t much power for doing things like running servos, operating stepper motors, reading potentiometers, or displaying text on an LCD. Like most any language based on C, the Arduino supports the notion of “libraries” code. Repositories that extend core programming functionality. Libraries let you re-use code without having to physically copy and paste it into all your programs.

The standard Arduino software installation comes with several libraries

you may use, and you can download others from the Arduino support pages and from third-party websites that publish Arduino library code. A good example of a library you'll use with the Robot and likely many other robot projects is Servo.

This library allows you to connect one or more hobby R/C servos to the Arduino's digital I/O pins. The Servo library comes with the standard Arduino installation package Library->Servo. This adds the line

```
#include <Servo.h>
```

Which tells the Arduino IDE that you wish to include the Servo library in your sketch. With the functionality of the library now available to you, you can use its various functions to control one or more servos.

For example, you can use the write function to rotate a servo to a specific position, from 0 to 180 degrees. The following code

```
myServo.write(90);
```

Moves a servo to its midpoint, or 90 degree position. Structurally, Arduino sketches are very straightforward and are pretty easy to read and understand. The Arduino program contains two main parts: setup () and loop ().

These are programming functions that do what their names suggest: setup () sets up the Arduino hardware, such as specifying which I/O lines you plan to use, and whether

They are inputs or outputs. The loop () function is repeated endlessly when the Arduino is operating.

Arduino IDE (Integrated development environment) is used to write the program and dump into the Arduino board.

4.2.1 ARDUINO SOFTWARE:

4.2.1.1 Open Arduino IDE as shown below

Open Arduino IDE

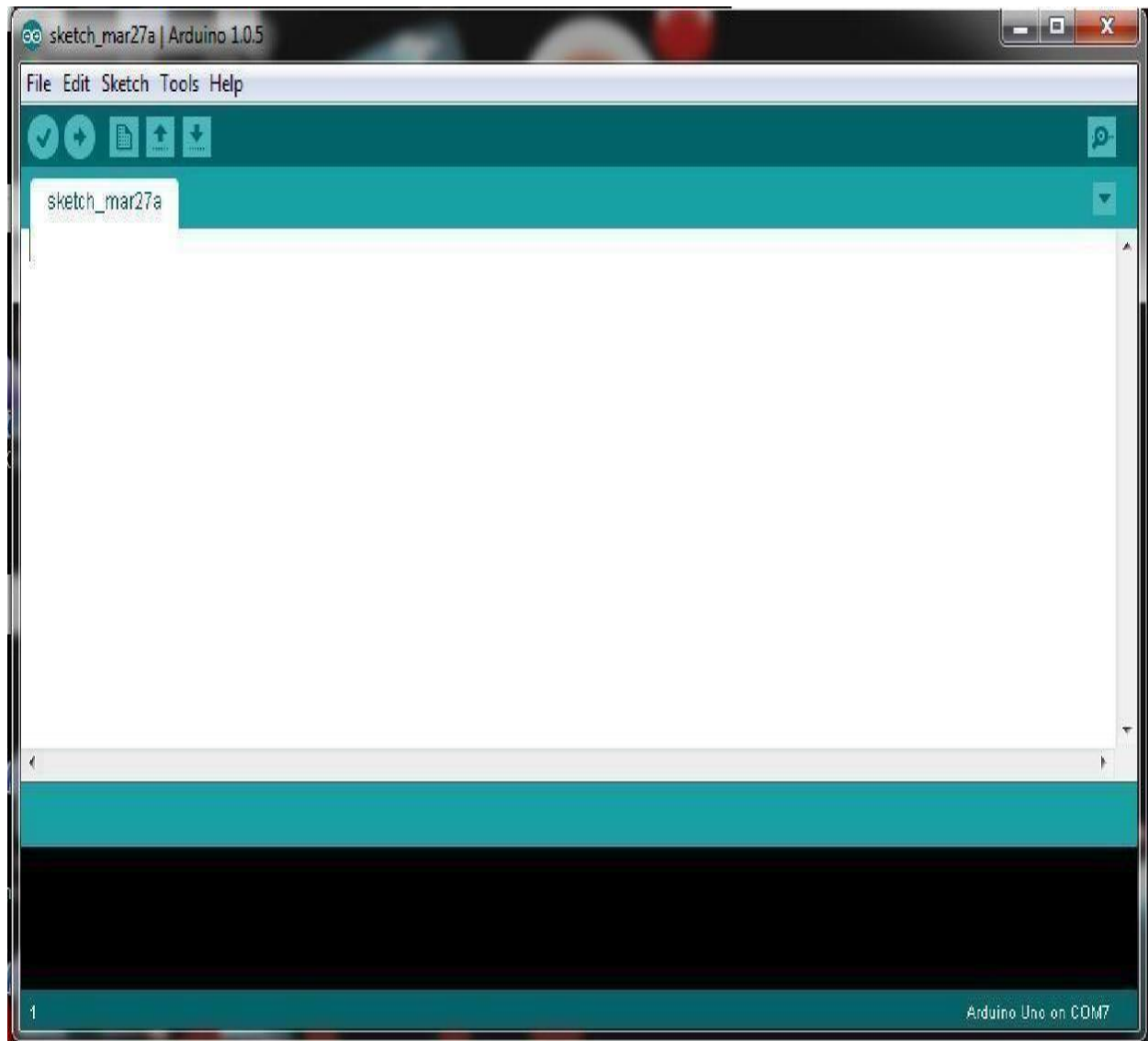


Fig 4.2 open Arduino

4.2.1.2 Select the COM Port from tool

Select the COM Port

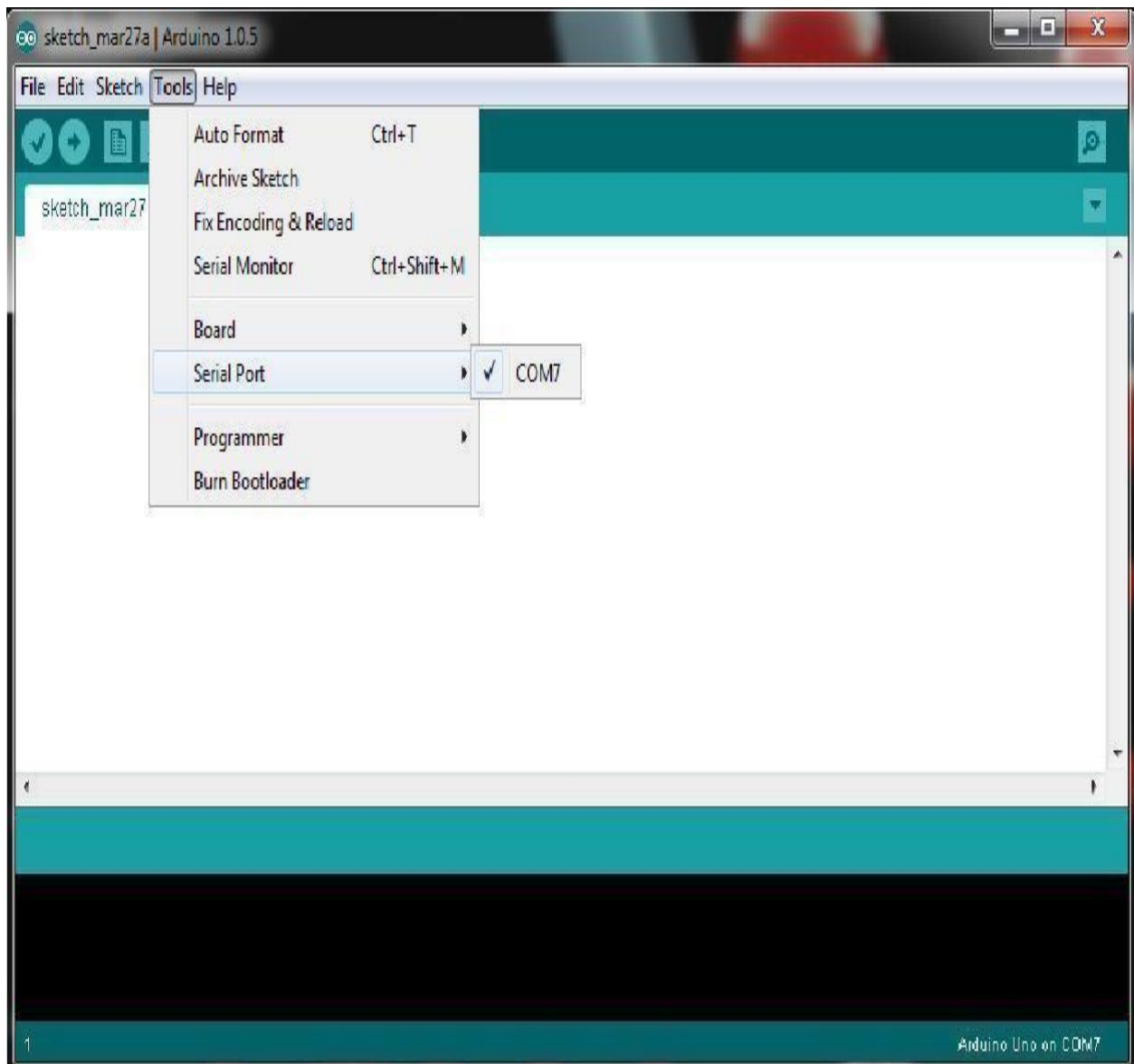


Fig 4.3 Select COM Port

4.2.1.3 Select the required Arduino board from Tools as shown below

Select the required Arduino board

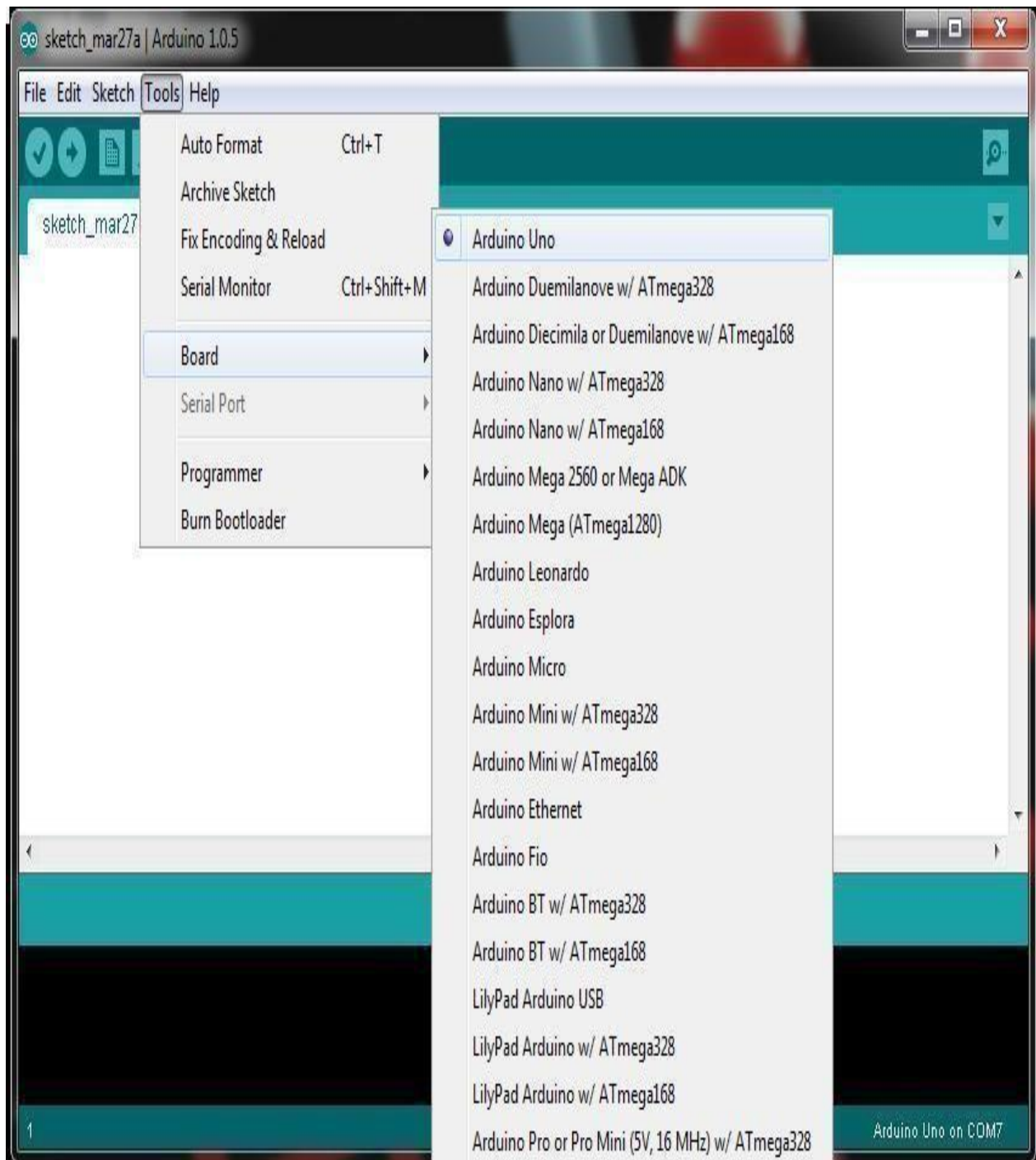
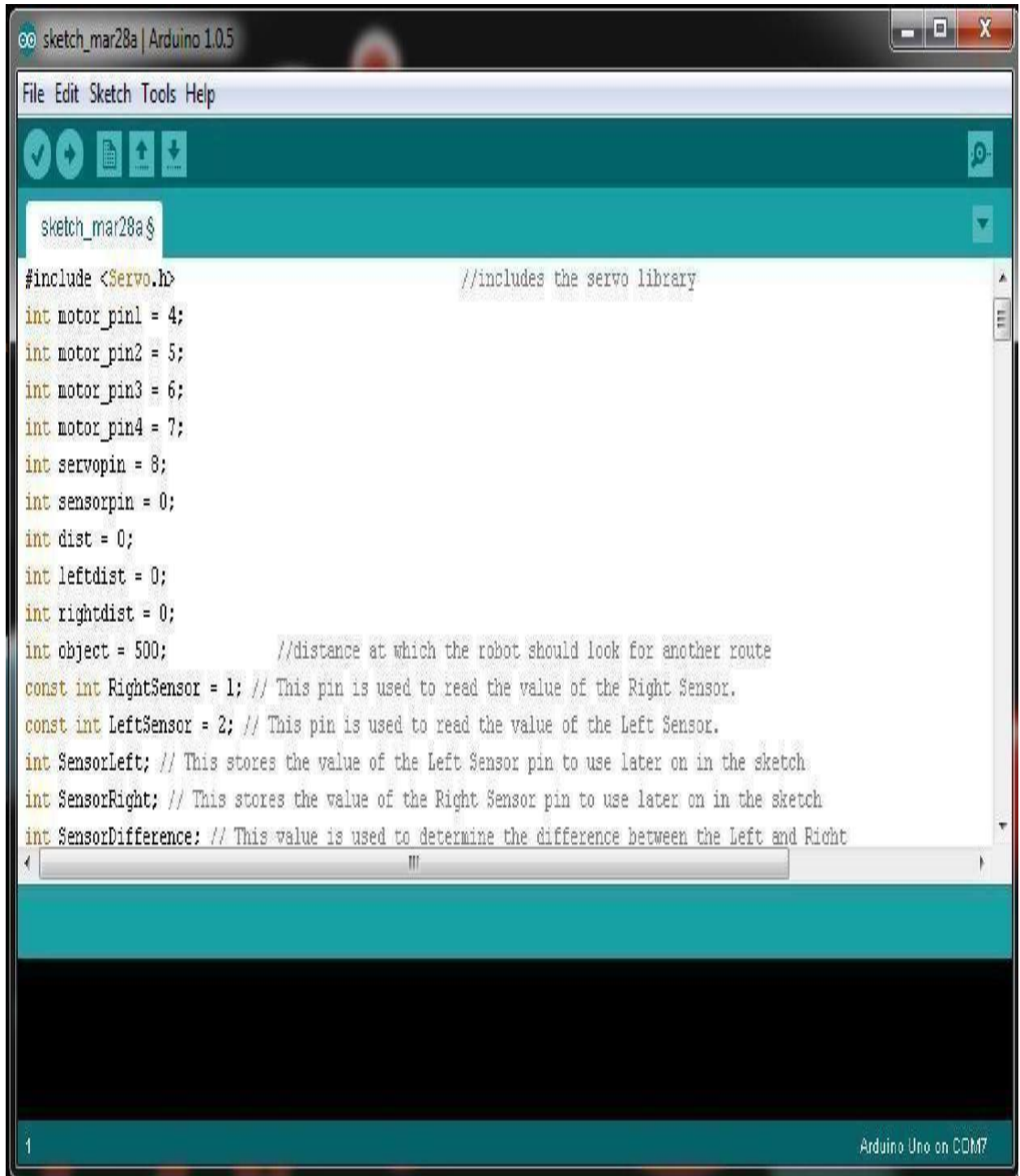


Fig 4.4 Select required Arduino

4.2.1.4 Write the sketch in Arduino IDE

Sketch in Arduino IDE



```
sketch_mar28a$
#include <Servo.h> //includes the servo library
int motor_pin1 = 4;
int motor_pin2 = 5;
int motor_pin3 = 6;
int motor_pin4 = 7;
int servopin = 8;
int sensorpin = 0;
int dist = 0;
int leftdist = 0;
int rightdist = 0;
int object = 500; //distance at which the robot should look for another route
const int RightSensor = 1; // This pin is used to read the value of the Right Sensor.
const int LeftSensor = 2; // This pin is used to read the value of the Left Sensor.
int SensorLeft; // This stores the value of the Left Sensor pin to use later on in the sketch
int SensorRight; // This stores the value of the Right Sensor pin to use later on in the sketch
int SensorDifference; // This value is used to determine the difference between the Left and Right
```

1 Arduino Uno on COM7

Fig 4.5 Sketch in Arduino code

4.2.1.5 Compile and upload the Sketch to Arduino board upload the Sketch to Arduino board



Fig 4.6 Compile and execution

4.3 EYE BLINK SENSOR:

This article is about the study of eye movement. For the tendency to visually track potential prey, see eye-stalking.

Scientists track eye movements in glaucoma patients to check vision impairment while driving.

Eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. An eye tracker is a device for measuring eye positions and eye movement. Eye trackers are used in research on the visual system, in psychology, in psycholinguistics, marketing, as an input device for human-computer interaction, and in product design.

Eye trackers are also being increasingly used for rehabilitative and assistive applications (related for instance to control of wheel chairs, robotic arms and prostheses). There are a number of methods for measuring eye movement. The most popular variant uses video images from which the eye position is extracted. Other methods use search coils or are based on the electrooculogram.

In 1879 in Paris, Louis Émile Javal observed that reading does not involve a smooth sweeping of the eyes along the text, as previously assumed, but a series of short stops (called fixations) and quick saccades. This observation raised important questions about reading, questions which were explored during the 1900s: On which words do the eyes stop? For how long? When do they regress to already seen words?

Edmund Huey built an early eye tracker, using a sort of contact lens with a hole for the pupil. The lens was connected to an aluminum pointer that moved in response to the movement of the eye. Huey studied and quantified regressions (only a small proportion of saccades are regressions), and he showed that some words in a sentence are not fixated.

The first non-intrusive eye-trackers were built by Guy Thomas Buswell in Chicago, using beams of light that were reflected on the eye and then recording them on film. Buswell made systematic studies into reading and picture viewing.

In the 1950s, Alfred L. Yarbus did important eye tracking research and his 1967 book is often quoted. He showed that the task given to a subject has a very large influence on the subject's eye movement. He also wrote about the relation between fixations and interest:

"All the records ... show conclusively that the character of the eye movement is either completely independent of or only very slightly dependent on the material of the picture and how it was made, provided that it is flat or nearly flat." The cyclical pattern in the examination of pictures "is dependent on not only what is shown on the picture, but also the problem facing the observer and the information that he hopes to gain from the picture."

"Records of eye movements show that the observer's attention is usually held only by certain elements of the picture. Eye movement reflects the human

thought processes; so the observer's thought may be followed to some extent from records of eye movement (the thought accompanying the examination of the particular object). It is easy to determine from these records which elements attract the observer's eye (and, consequently, his thought), in what order, and how often."

"The observer's attention is frequently drawn to elements which do not give important information but which, in his opinion, may do so. Often an observer will focus his attention on elements that are unusual in the particular circumstances, unfamiliar, incomprehensible, and so on."^[8]

" When changing its points of fixation, the observer's eye repeatedly returns to the same elements of the picture. Additional time spent on perception is not used to examine the secondary elements, but to re-examine the most important elements."

In the 1970s, eye-tracking research expanded rapidly, particularly reading research. A good overview of the research in this period is given by Rayner.

In 1980, Just and Carpenter formulated the influential *Strong eye-mind hypothesis*, that "there is no appreciable lag between what is fixated and what is processed".

If this hypothesis is correct, then when a subject looks at a word or object, he or she also thinks about it (process cognitively), and for exactly as long as the recorded fixation.

The hypothesis is often taken for granted by researchers using eye-tracking. However, gaze-contingent techniques offer an interesting option in order to disentangle overt and covert attentions, to differentiate what is fixated and what is processed.

During the 1980s, the eye-mind hypothesis was often questioned in light of covert attention, ^{[15][16]} the attention to something that one is not looking at, which people often do. If covert attention is common during eye-tracking recordings, the resulting scan-path and fixation patterns would often show not where our attention has been, but only where the eye has been looking, failing to indicate cognitive processing.

The 1980s also saw the birth of using eye-tracking to answer questions related to human-computer interaction. Specifically, researchers investigated how users search for commands in computer menus. ^[17] Additionally, computers allowed researchers to use eye-tracking results in real time, primarily to help disabled users.

More recently, there has been growth in using eye tracking to study how users interact with different computer interfaces. Specific questions researchers ask are related to how easy different interfaces are for users. ^[17] The results of the eye tracking research can lead to changes in design of the interface.

Yet another recent area of research focuses on Web development. This can include how users react to drop-down menus or where they focus their attention on a website so the developer knows where to place an advertisement.

According to Hoffman, current consensus is that visual attention is always slightly (100 to 250 ms) ahead of the eye. But as soon as attention moves to a new position, the eyes will want to follow.

We still cannot infer specific cognitive processes directly from a fixation on a particular object in a scene. ^[21] For instance, a fixation on a face in a picture may indicate recognition, liking, dislike, puzzlement etc.

Therefore, eye tracking is often coupled with other methodologies, such as introspective verbal protocols.

Thanks to advancement in portable electronic devices, portable head-mounted eye trackers currently can achieve excellent performance and are being increasingly used in research and market applications targeting daily life settings.

In the 21st century, the use of artificial intelligence (AI) and artificial neural networks has become a viable way to complete eye-tracking tasks and analysis. In particular, the convolutional neural network lends itself to eye-tracking, as it is designed for image-centric tasks.

With AI, eye-tracking tasks and studies can yield additional information that may not have been detected by human observers.

The practice of deeplearning also allows for a given neural network to improve at a given task when given enough sample data. This requires a relatively large supply of training data, however.

The potential use cases for AI in eye-tracking cover a wide range of topics from medical applications to driver safety to game theory. While the CNN structure may fit relatively well with the task of eye-tracking, researchers have the option to construct a custom neural network that is tailored for the specific task at hand.

In those instances, these in-house creations can outperform pre-existing templates for a neural network. In this sense, it remains to be seen if there is a way to determine the ideal network structure for a given task.

Eye-trackers measure rotations of the eye in one of several ways, but principally they fall into one of three categories:

(i) measurement of the movement of an object (normally, a special contact lens) attached to the eye; (ii) optical tracking without direct contact to the eye; and (iii) measurement of electric potentials using electrodes placed around the eyes.

4.3.1 Eye-attached tracking

The first type uses an attachment to the eye, such as a special contact lens with an embedded mirror or magnetic field sensor, and the movement of the attachment is measured with the assumption that it does not slip significantly as the eye rotates.

Measurements with tight-fitting contact lenses have provided extremely sensitive recordings of eye movement, and magnetic search coils are the method of choice for researchers studying the dynamics and underlying physiology of eye movement.

This method allows the measurement of eye movement in horizontal, vertical and torsion directions.

4.3.2 Optical tracking

The second broad category uses some non-contact, optical method for measuring eye motion. Light, typically infrared, is reflected from the eye and sensed by a video camera or some other specially designed optical sensor. The information is then analyzed to extract eye rotation from changes in reflections. Video-based eye trackers typically use the corneal reflection (the first [Purkinje image](#)) and the center of the pupil as features to track over time.

A more sensitive type of eye-tracker, the dual-Purkinje eye tracker, uses reflections from the front of the cornea (first Purkinje image) and the back of the lens (fourth Purkinje image) as features to track. A still more sensitive method of tracking is to image features from inside the eye, such as the retinal blood vessels, and follow these features as the eye rotates. Optical methods, particularly those based on video recording, are widely used for gaze-tracking and are favored for being non-invasive and inexpensive.

4.3.3 Electric potential measurement

The third category uses electric potentials measured with electrodes placed around the eyes. The eyes are the origin of a steady electric potential field which can also be detected in total darkness and if the eyes are closed. It can be modelled to be generated by a dipole with its positive pole at the cornea and its negative pole at the retina. The electric signal that can be derived using two pairs of contact electrodes placed on the skin around one eye is called Electrooculogram (EOG).

If the eyes move from the centre position towards the periphery, the retina approaches one electrode while the cornea approaches the opposing one. This change in the orientation of the dipole and consequently the electric potential field results in a change in the measured EOG signal.

Inversely, by analysing these changes in eye movement can be tracked. Due to the discretisation given by the common electrode setup, two separate movement components – a horizontal and a vertical – can be identified.

A third EOG component is the radial EOG channel, which is the average of the EOG channels referenced to some posterior scalp electrode. This radial EOG channel is sensitive to the saccadic spike potentials stemming from the extra-ocular muscles at the onset of saccades, and allows reliable detection of even miniature saccades.

Due to potential drifts and variable relations between the EOG signal amplitudes and the saccade sizes, it is challenging to use EOG for measuring slow eye movement and detecting gaze direction. EOG is, however, a very robust technique for measuring saccadic eye movement associated with gaze shifts and detecting blinks. Contrary to video-based eye-trackers, EOG allows recording of eye movements even with eyes closed, and can thus be used in sleep research. It is a very light-weight approach that, in contrast to current video-based eye-trackers, requires only very low computational power; works under different lighting conditions; and can be implemented as an embedded, self-contained wearable system. It is thus the method of choice for measuring eye movement in mobile daily-life situations and REM phases during sleep. The major disadvantage of EOG is its relatively poor gaze-direction accuracy compared to a video tracker. That is, it is difficult to determine with good accuracy exactly where a subject is looking, though the time of eye movements can be determined.

Technologies and techniques:

The most widely used current designs are video-based eye-trackers. A camera focuses on one or both eyes and records eye movement as the viewer looks at some kind of stimulus. Most modern eye-trackers use the center of the pupil and infrared / near-infrared non-collimated light to create corneal reflections (CR). The vector between the pupil center and the corneal reflections can be used to compute the point of regard on surface or the gaze direction. A simple calibration procedure of the individual is usually needed before using the eye tracker.

Two general types of infrared / near-infrared (also known as active light) eye-tracking techniques are used: bright-pupil and dark-pupil. Their difference is based on the location of the illumination source with respect to the optics. If the illumination is coaxial with the optical path, then the eye acts as a retroreflector as the light reflects off the retina creating a bright pupil effect similar to red eye. If the illumination source is offset from the optical path, then the pupil appears dark because the retroreflection from the retina is directed away from the camera.^{1]}

Bright-pupil tracking creates greater iris/pupil contrast, allowing more robust eye-tracking with all iris pigmentation, and greatly reduces interference caused by eyelashes and other obscuring features. It also allows tracking in lighting conditions ranging from total darkness to very bright.

Another, less used, method is known as passive light. It uses visible light to illuminate, something which may cause some distractions to users.^[34] Another challenge with this method is that the contrast of the pupil is less than in the active light methods, therefore, the center of [iris](#) is used for calculating the vector instead.

This calculation needs to detect the boundary of the iris and the white sclera (limbus tracking). It presents another challenge for vertical eye movements due to obstruction of eyelids

Eye-tracking setups vary greatly: some are head-mounted, some require the head to be stable (for example, with a chin rest), and some function remotely and automatically track the head during motion.

Most use a sampling rate of at least 30 Hz. Although 50/60 Hz is more common, today many video-based eye trackers run at 240, 350 or even 1000/1250 Hz, speeds needed in order to capture fixational eye movements or correctly measure saccade dynamics.

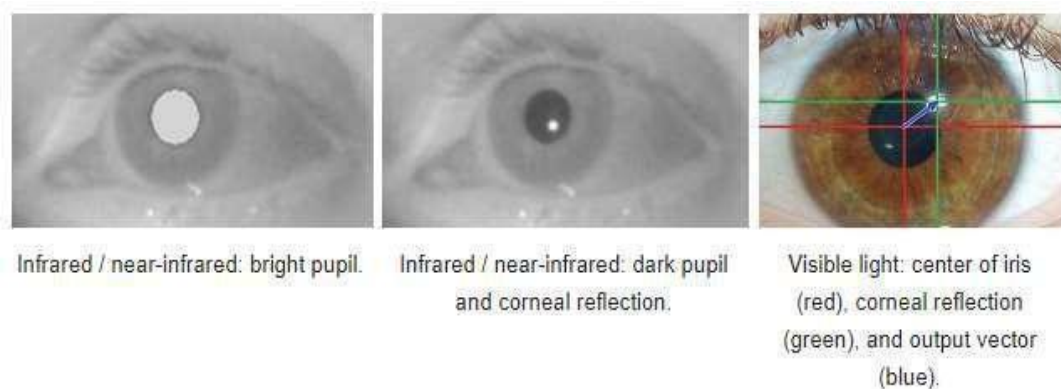


Fig 4.7 Eye tracking

Eye movements are typically divided into fixations and saccades – when the eye gaze pauses in a certain position, and when it moves to another position, respectively. The resulting series of fixations and saccades is called a scanpath. Smooth pursuit describes the eye following a moving object. Fixational eye movements include microsaccades: small, involuntary saccades that occur during attempted fixation. Most information from the eye is made available during a fixation or smooth pursuit, but not during a saccade.

Scanpaths are useful for analyzing cognitive intent, interest, and salience. Other biological factors (some as simple as gender) may affect the scanpath as well.

Eye tracking in human-computer interaction (HCI) typically investigates the scanpath for usability purposes, or as a method of input in gaze-contingent displays, also known as gaze-based interfaces.

4.3.4 Eye tracking vs. gaze-tracking

Eye-trackers necessarily measure the rotation of the eye with respect to some frame of reference. This is usually tied to the measuring system. Thus, if the measuring system is head-mounted, as with EOG or a video-based system mounted to a helmet, then eye-in-head angles are measured. To deduce the line of sight in world coordinates, the head must be kept in a

constant position or its movements must be tracked as well. In these cases, head direction is added to eye-in-head direction to determine gaze direction.

If the measuring system is table-mounted, as with scleral search coils or table-mounted camera (“remote”) systems, then gaze angles are measured directly in world coordinates. Typically, in these situations head movements are prohibited. For example, the head position is fixed using a bite bar or a forehead support. Then a head-centered reference frame is identical to a world-centered reference frame. Or colloquially, the eye-in-head position directly determines the gaze direction.

Some results are available on human eye movements under natural conditions where head movements are allowed as well. [\[42\]\[43\]](#) The relative position of eye and head, even with constant gaze direction, influences neuronal activity in higher visual areas.

4.3.4 Eye-tracking of younger and elderly people while walking

While walking, elderly subjects depend more on foveal vision than do younger subjects. Their walking speed is decreased by a limited visual field, probably caused by a deteriorated peripheral vision.

Younger subjects make use of both their central and peripheral vision while walking. Their peripheral vision allows faster control over the process of walking.

4.3.5 The future of the Eye-Tracking Technology:

Physiological response measurements coupled with eye tracking have entered the UX field. These techniques are in their infancy for UX researchers, but we expect these technologies to continue to improve and be as easy to use as a modern eye trackers. By 2020, companies large and small will be more aggressive in seeking new ways to engage their users, and it will be common for marketers and UX practitioners to use eye tracking with electroencephalography or electrodermal activity to assess the user experience. Competition in the digital advertising space will go beyond traditional website banner ads and click-through rates.

While much of the rich user experience still involves traditional links, innovative products are emerging that allow UX researchers new opportunities. Current eye-tracking technology allows UX researcher to track users' eyes through untethered headset units, such as the Tobii Glasse

These eye trackers allow users complete freedom of movement and can track everything the user looks at from mobile devices to the menu in their local café. By allowing users to freely move about their environment, we are able to observe more natural behaviors compared to when participants sit in front of an eye tracker in a lab.

Marketers are now able to overlay eye-tracking data with shopping shelves to assess what people look at while they shop, and Google has recently patented a process for measuring pay-per-gaze as a way to measure the effectiveness of advertising.

The patent describes a process in which a head-mounted gaze-tracking device can identify ads that a user looks. These data can be used to understand what types of advertisements people attend to and in what situations people pay attention to them.

Advertisements will likely continue to be a revenue source for both digital and offline content in the future. Eye tracking will help designers to understand a user's level of engagement with ads by fixation counts and duration in the context of the user's actions. Others are creating technology whereby the eyes control the device. Taking advantage of built-in webcam technology, manufacturers are introducing eye-tracking capabilities into mobile devices.

A *Wall Street Journal* article published in May 2013 broke the news that Amazon has been developing a new mobile phone that would include retina-tracking technology that will allow users to navigate through content by simply using their eyes. Software already exists that allows users to merely look at the corner of an iPad to turn the page of an eBook, or look at the corner of a television screen to change the station. This modern technology enables accessibility for disabled people who cannot interact with technology with their hands. The smartwatch has recently been patented, which uses eye trackers to decide what content to display on a tiny watch interface, and other researchers are using eye tracking to diagnose Alzheimer's disease, schizophrenia, and memory disorders. Thus, we are now at a time when eye tracking is used not only to evaluate products but to interact with them and to diagnose disorders.

As eye-tracking technology continues to become miniaturized and cost less, we will see many more devices with this capability built in. Google Glass presents numerous opportunities for interaction designers and user researchers. While the first generation of this technology will be somewhat limited, by 2020, we can expect these types of devices to be commonplace.

Eye tracking is likely to become an integral part of head-mounted displays where most user interactions will take place with eye movements. In addition to being an input device, user researchers can see through the eyes of their users with the existing external camera and track where they are looking with a camera facing the eye.

4.3.6 Ubiquitous Eye Tracking

We are entering a brave new world for eye tracking. New technologies are emerging that will allow UX researchers to track users' eyes anywhere and everywhere. Obviously this will come with significant concerns about privacy and the ability for the user to control when they are being tracked.

While webcam-based eye tracking has emerged, there are still significant limitations to using this technology. Aga Bojko^[53] (2013) identified several concerns about this methodology related to accuracy:

"Webcam eye tracking has much lower accuracy than real eye trackers. While a typical remote eye tracker (Tobii T60) has accuracy of 0.5 degrees of visual angle, a webcam will produce accuracy of 2-5 degrees, provided that the participant is NOT MOVING.

To give you an idea of what that means, five degrees correspond to 2.5 inches (6 cm) on a computer monitor (assuming viewing distance of 27 inches), so the actual gaze location could be anywhere within a radius of 2.5 inches from the gaze location recorded with a webcam.

I don't know about you but I wouldn't be comfortable with that level of inaccuracy in my studies".

As both the quality of webcams increases and the algorithms used for tracking the movement of the eyes improves, we can expect to see more of this type of eye tracking used in the future.

A short distance in the future will be a time of ubiquitous eye tracking with the ability to track a user's eyes on nearly every device they use. For UX researchers, this opens up enormous opportunities for large-scale quantitative eye-tracking studies.

In addition to reaching broad audience groups currently not practical with today's technology, we will also have true freedom to collect eye-tracking data in the user's natural environment.

Taking the next step:

Currently eye-tracking is used by industry professionals to assess and improve the user experience of websites, forms, surveys, and games, both on desktop and mobile devices. The case studies and topics the wide range of applications of eye tracking in user experience design. It's by no means an exhaustive assembly of all possible uses for eye tracking and interactive media.

Many of the applications referenced are still their infancy and further research is required to better understand these areas



Fig 4.8 IR SENSOR IN GOOGLES

4.4 Motor driver:

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micro controllers like Arduino, PIC, ARM etc.. This IC will be the right choice for you.

Using this L293D motor driver IC is very simple. The IC works on the principle of Half H-Bridge, let us not go too deep into what H-Bridge means, but for now just know that H bridge is a set up which is used to run motors both in clock wise and anti clock wise direction. As said earlier this IC is capable of running two motors at the any direction at the same time, the circuit to achieve the same is shown below.

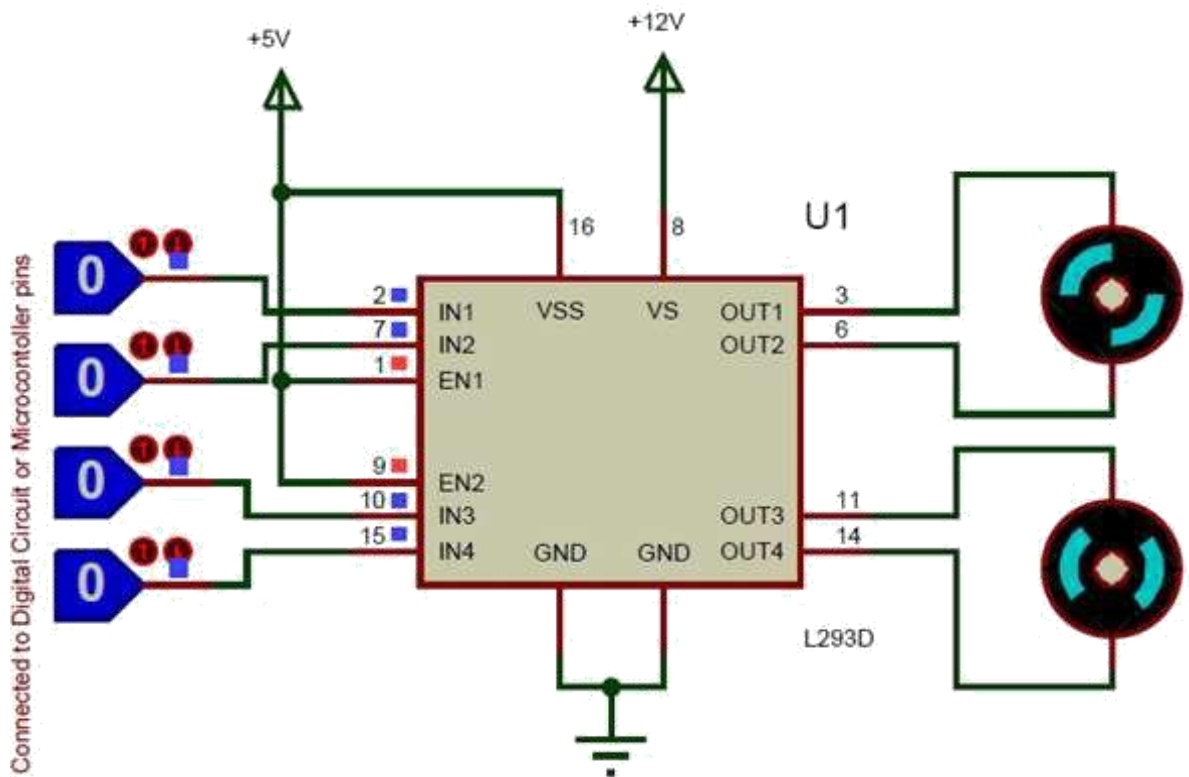


Fig 4.9 Motor Driver

All the Ground pins should be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which provides the voltage for the IC to work, this must be connected to +5V. The other is Vs(Vcc2) which provides voltage for the motors to run, based on the specification of your motor you can connect this pin to anywhere between 4.5V to 36V, here I have connected to +12V.

The Enable pins (Enable 1,2 and Enable 3,4) are used to Enable Input pins for Motor 1 and Motor 2 respectively.

Since in most cases we will be using both the motors both the pins are held high by default by connecting to +5V supply. The input pins Input 1,2 are used to control the motor 1 and Input pins 3,4 are used to control the Motor2.

The input pins are connected to the any Digital circuit or microcontroller to control the speed and direction of the motor. You can toggle the input pins based on the following table to control your motor.

4.4.1 Applications

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

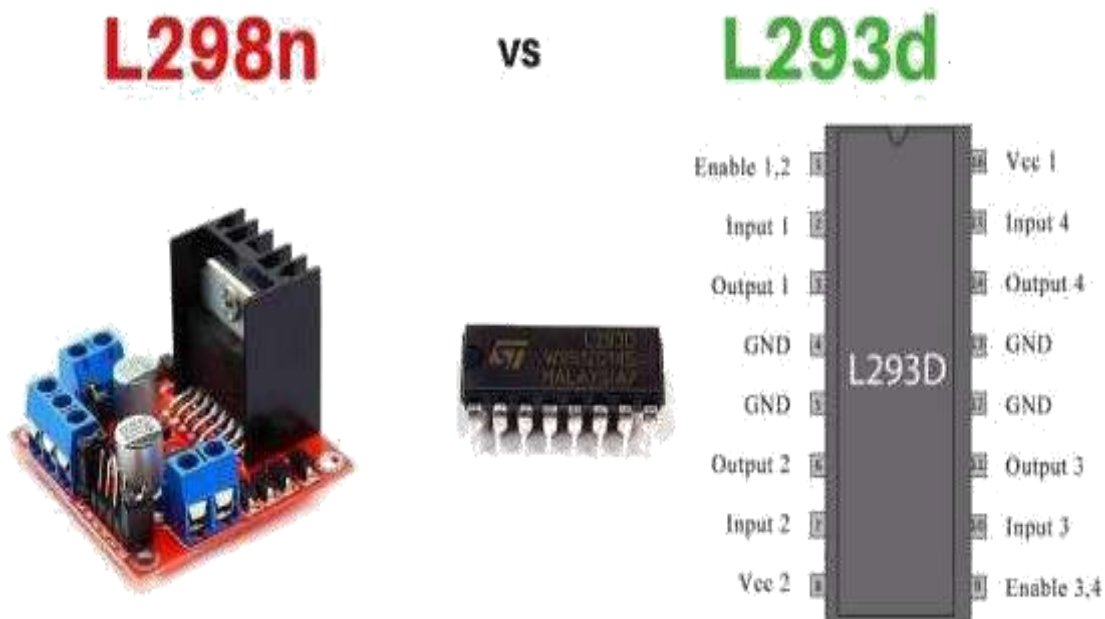


Fig 4.10 Motor applications

Chapter 5

RESULT

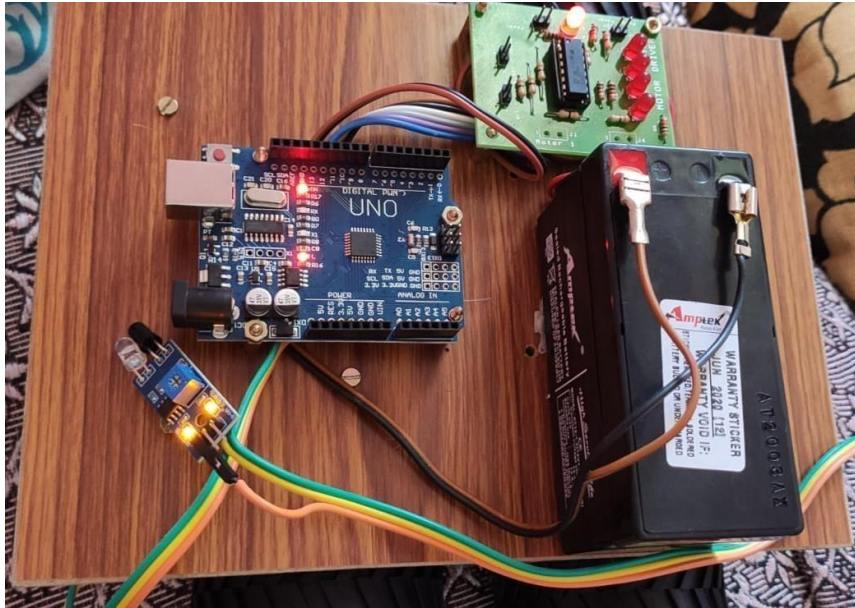


Fig 5.1 output

The proposed system is working perfectly without any issues. Output image is shown above in the fig 5.1.

Chapter 6

CONCLUSION

In conclusion, a Eye controlled wheel chair can be designed in a better way by using arduino and c language . This wheelchair is easy for differently abled and paralyzed people to make their life more easier. The IR sensor is mounted on goggles for capturing the eye blink movement. This sensor captured his eye blink movement and this signal sends to motor driver circuit to move and stop the wheelchair. As per the counting of number of eye blink movement the circuit will perform different operations like left, right, forward, backward and stop.

References:

- [1.] Teodiano F. Bastos-Filho, Fernando Auat Cheein, Member, IEEE, Sandra M.T. M^uller, Wanderley C. Celeste, Celso de la Cruz, Daniel C. Cavalieri, M^ario Sarcinelli Filho, Paulo F. S. Amaral, Elisa Perez, Carlos Soria, Ricardo Carelli, Senior Member, IEEE, "Towards a New Modality Independent Interface for a Robotic Wheelchair", May 2013
- [2.] Poonam S. Gajwani & Sharda A. Chhabria "Eye Motion Tracking for Wheelchair Control"
- [3.] Elisa Perez, Natalia López, Eugenio Orosco, Carlos Soria, Vicente Mut, and Teodiano Freire-Bastos "Robust Human Machine Based on Head Movements Applied to Assistive Robots"
- [4.] Manuel Mazo, Francisco J. Rodriguez, Josi I. L. Zaro, Jesiis Ureia, Juan C. Garcia "Wheelchair for Physically Disabled People with Voice, Ultrasonic and Infrared Sensor Control", Autonomous Robots, 1995, Volume 2, Issue 3, pp 203-224
- [5.] Maurizio Caon, Stefano Carrino, Simon Ruffieux, Omar Abou Khaled, and Elena Mugellini —Augmenting Interaction Possibilities between People with Mobility Impairments and Their Surrounding Environment
- [6.] <https://learn.sparkfun.com/tutorials/accelerometer-basics/all>
- [7.] <http://www.keil.com/c51/c51.asp>
- [8.] <https://www.pantechsolutions.net/sensors/user-manual-for-eye-blinksensor#sthash.Rv5vb3yy.dpuf>
- [9.] ADC0808/ADC0809 8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer Texas Instrumentation
- [10.] P89V51RB2/RC2/RD2 8-bit 80C51 5 V low power 16/32/64 kB flash microcontroller with 1 kB RAM Rev. 05 — 12 November 2009 Product data sheet
- [11.] The 8051 Microcontroller and Embedded Systems using Assembly and C – Second Edition Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay
- [12.] <http://www.engineersgarage.com/electroniccomponents/rf-module-transmitter-receiver>
- [13.] https://www.sparkfun.com/datasheets/Components/ADXL330_0.pdf
- [14.] <https://learn.sparkfun.com/tutorials/accelerometer-basics/all>