

Eye Blink controlled wheel chair system for physically challenged people using virtual mapping

Submitted in partial fulfillment of the requirements for
the award of

ELECTRONICS AND COMMUNICATION ENGINEERING

by

**KOTA VENKATA ASHOK REDDY (37130210)
NALLA JAYADEV (37130271)**



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

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with "A" by NAAC

**JEPPIAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119**

APRIL 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KOTA VENKATA ASHOK REDDY (37130210)** and **NALLA JAYADEV (37130271)** who carried out the project entitled **"EYE BLINK CONTROLLED WHEEL CHAIR SYSTEM FOR PHYSICALLY CHALLENGED PEOPLE USING VIRTUAL MAPPING** under my supervision from .

Internal Guide

Dr.P.Chitra

Head of the Department

(DR. T. RAVI)

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examine

DECLARATION

We, KOTA VENKATA ASHOK REDDY (Reg.No. 37130210) and NALLA JAYADEV (Reg.No.37130271) hereby declare that the Project Report entitled **“EYE BLINK CONTROLLED WHEEL CHAIR SYSTEM FOR PHYSICALLY CHALLENGED PEOPLE USING VIRTUAL MAPPING”** done by us under the guidance of Dr.P.Chitra, Ph.D. is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:
PLACE:

SIGNATURE OF THE CANDIDATE
PLACE:

ACKNOWLEDGEMENT

I am 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. I am grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA M.E.Ph.D., Prof. & Dean, School of Electrical and Electronics Engineering** and **Dr. T. Ravi M.E.Ph.D., Head of the Department, Dept. of ELECTRONICS AND COMMUNICATION ENGINEERING** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.P.Chitra** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

As indicated by another report arranged together by the World Health Organization and the World Bank, 15 percent of the world 's population is incapacitated. The utilization of controlled wheelchairs with high navigational insight is one of the extraordinary strides towards the coordination of seriously physically disabled and intellectually disabled individuals. Driving a wheelchair is a monotonous assignment for severely crippled people, except if they use the tongue to control the joystick. At the same time blind and paraplegic individuals deal with two problems, which causes uncomfortable circumstance for them, i.e movement and limitation. Various frameworks are being created to beat the issues portrayed above, permitting the end-client to perform safe movements and achieve some every day life important tasks. Our Robotic wheelchair uses eye blink and head tilt movement to control the wheelchair. moreover, we can give more freedom to the incapacitated individual by utilizing the same head-tilt movement to speak with the devices in a room for example: a fan. This communication is done using a RF transmitter and receiver. Utilizing this, the person can control various devices easily.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
1	Introduction	9-10
	1.1. Literature Survey	9-10
2	WORKING PRINCIPLE	11-17
3	Requirements	18-24
	3.1. Embedded Systems	18-20
	3.1.1. Overview of embedded systems	18-20
	3.1.2. Block diagram of an embedded system	21
	3.1.3. Characteristics of embedded systems	21-23
4	Materials and Methods	24-33
	4.1 Power Supply	24
	4.1.1 Battery	24
	4.1.2 Regulators	24
	4.2Aurdino Software	25-28
	4.3. Arduino	29-33

6	Conclusion	38
7	References	39

LIST OF FIGURES

FIG NO	TITLE	PAGE NO
1.1.1	Block diagram of proposed system	10
2.1.1	Connection of eye blink sensor with spectacles	12
2.1.2	IR Sensor	12
2.1.3	Eye Demonstration	14
2.1.4	Motor Driver Pin out	17
3.1.1	Block Diagram of typical embedded systems	21
4.3.1	Aurdino nano	29
4.3.2	Aurdino nano pin out	31
4.3.3	Aurdino nano specifications	32
5.1.1	Demonstration	35
5.1.2	Source code	36
5.1.3	Output checking	37

CHAPTER 1

INTRODUCTION

Wheelchairs were designed with the idea to help handicapped people to move around and accomplish daily life tasks, Many brilliant wheelchairs have been designed and with an increasing number of advances in technologies fresher and more intelligent wheelchairs are coming into the market to help the seriously debilitated people. Our wheelchair is likewise an endeavor to utilize straightforward and simple developments of the eye and head to begin, pause and move the wheelchair. In addition, we also connect the various device in a room to the wheelchair and communicate with it through a RF module.

Literature Review:

Existing System:

- In the existing system we can control the wheelchair 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 squint sensor detects whether eye is open or shut. The eye-squint sensor works by enlightening the eye as well as eyelid region with infrared light and afterward checking the adjustments in the mirrored light utilizing a phototransistor and differentiator circuit.
- When the eye can blink two times means wheel chair can move to forward. And three times means left turn four times means right turn.

Drawbacks:

- It is difficult.

Proposed system:

- In this system we used eye blink sensor to control the wheel chair. Firstly, we map 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 sensor detects one blink means it goes to move bedroom 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 wheelchair as per the direction.

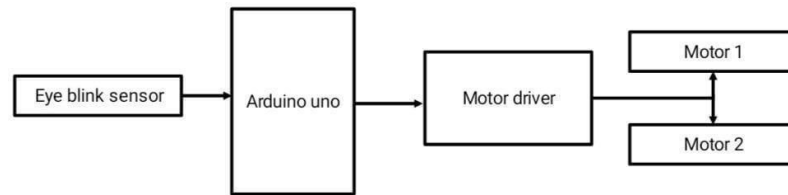


Fig 1.1.1 Block Diagram of Proposed System.

CHAPTER 2

2.1 WORKING PRINCIPLE:

Eye Blink Sensor:

The eye blink sensor detects whether eye is open or shut. The eye blink sensor works by enlightening the eye as well as eyelid region with infrared light and afterward observing the adjustments in the mirrored light utilizing a photometer and differentiator circuit. The exact functionality depends greatly on the positioning and aiming of the emitter and detector with respect to the eye.



Fig.2.1.1. Connection of eye blink sensor with spectacles



Fig 2.1.2. IR Sensor

- **FEATUTRES:**

- EYE BLINK sign by LED Instant yield advanced sign for straightforwardly Connecting to microcontroller Compact Size Working Voltage +5V DC

APPLICATION:

- Digital Eye Blink screen for Vehicle Accident avoidance and. Appropriate for ongoing driving applications

SPECIFICATION:

Operating Voltage :+5V DC regulated Operating Current

:100mA

Output Data Level : TTL Level

Eye Blink Indicated by LED and Output High Pulse

USING SENSOR:

Connect regulated DC power supply of 5 Volts. Dark wire is Ground, Next center wire is Brown which is yield and Red wire is positive supply. These wires are likewise set apart on PCB. To test sensor you just need power the sensor by interface two wires +5V and GND. You can leave the yield wire as its worth.

At the point When Eye closed, LED is off & the output is at 0V. Put Eye blink sensor glass on the face inside 15mm distance, and you can see the LED blinking on each Eye blink. The yield is dynamic high for Eye close and can be offered straight forwardly to microcontroller for connecting applications.

EYE BLINK OUTPUT:

5V (High)	→ LED ON When Eye is close. 0V
(Low)	→ LED OFF when Eye is open.



Fig 2.1.3 Eye Demonstration

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 depicted by its makers as a “programming language and climate for individuals who need to program pictures, animation, and cooperations.” The Arduino programming language use an open source project known as Wiring (wiring.org.co). The Arduino language is based on ordinary C. on the off chance that you are new to this language, don't stress; it's not difficult to learn, and the Arduino IDE gives some input when you commits errors in your programs.

The Arduino Uno has a various number of facilities for speaking with a computer, another Arduino, or other microcontrollers. The ATmega328 gives UART TTL (5V) sequential communication, which is accessible on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this sequential communication over USB and shows up as a virtual com port to software on the computer. The '16U2 firmware utilizes the standard USB COM drivers, and no outside driver is required. In any case, on Windows, a inf file is needed. The Arduino software includes a chronic screen which permits straight forward literary information is being sent through 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 takes into account serial communication on any of the Uno's digital pins. The ATmega328 likewise upholds I2C (TWI) and SPI communication. The Arduino software includes a Wire library to improve on utilization of the I2C bus; see the documentation for details. For SPI communication, utilize the SPI library.

As you go through the rundown of programming articulations available in the Arduino IDE (choose Help->Reference), you may think there isn't lot of power for doing things like running servos, working stepper motors, reading potentiometers, or showing text on an LCD. Like most any language dependent on C, the Arduino upholds the thought of "libraries"

Repositories that expand core programming functionality. Libraries let you re-use code without having to truly reorder it into every one of your projects. The standard Arduino software establishment accompanies a few libraries you may utilize, and you can download others from the Arduino support pages and from outsider websites that distributes Arduino library code. A genuine illustration of a library you'll utilize with the Robot and robot projects is Servo. This library permits you to interface at least one hobby R/C servos to the Arduino's digital I/O pins. The Servo library accompanies the standard Arduino establishment bundle Library->Servo. This adds the line `#include <Servo.h>` Which tells the Arduino IDE that you wish to remember the Servo library for your sketch. With the functionality of the library now accessible to you, you can utilize its different capacities to control at least servos. For example, you can utilize the `write` function to turn a servo to a particular position, from 0 to 180 degrees. The following code

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

Moves a servo to its midpoint, or 90 degree position. Fundamentally, Arduino draws are very direct and are really simple to peruse and understand. The Arduino program contains two fundamental parts: `setup ()` and `loop ()`. These are programming functions that do what their names purpose: `setup ()` sets up the Arduino equipment, such as indicating which I/O lines you intend to utilize, and whether They are inputs or yields. 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.

Motor Driver:

The L293D is a popular 16-Pin **Motor Driver IC**. As the name recommends it is basically used to drive engines. A solitary **L293D IC** is equipped for running two DC engines simultaneously; likewise the direction of these two engines can be controlled autonomously. So if you have motors which has woking voltage under 36V and working current under 600mA, which are to be controlled by advanced circuits like Op-Amp, 555 clocks, digital gates or even Micron rollers like Arduino, PIC, ARM etc.. This IC will be the correct option for me.

Utilising this L293D motor driver IC is extremely simple. The IC deals with the principle of Half H-Bridge, let us not dive excessevely deep into what H-Bridge implies, but for now just realize that H bridge is a set up which is utilized to run motors both in clock wise and hostile to clockwise direction. As said earlier this IC is equipped for running two motors at the any direction simultaneously , the circuit to got the equivalent is shown below.

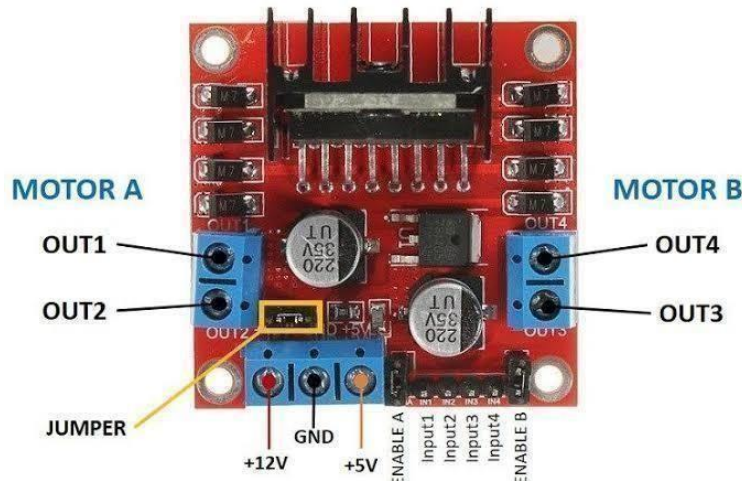


Fig 1.1.4 Motor Driver Pin Out

All the Ground pins ought to be grounded. There are two power pins for this IC, one is the $V_{ss}(V_{cc1})$ which gives the voltage for the IC to work, this should be associated with +5V. The other is $V_s(V_{cc2})$ which gives voltage for the motors to run, on the requirement of my motor i can connect this pin to everywhere 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 by and large we will be utilize both the motors and both the pins are held high ofcourse by associating with +5V supply. The input pins Input 1,2 are utilized to control the motor 1 and Input pins 3,4 are utilized to control the Motor 2. The input pins are associated with the any Digital circuit or microcontroller to bear the speed and direction of the motor. You can flip the input pins dependent on the following table to control your motor.

Applications

- Used to drive high ebb and flow Motors utilizing Digital Circuits.
- Can be utilized to drive Stepper motors .
- High current LED's can be driven.

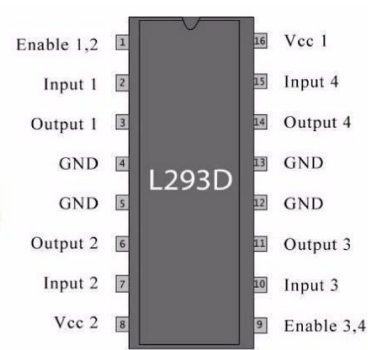
Relay Driver module (Latching Relay is possible)

L298n



vs

L293d



CHAPTER 3

Requirements

Hardware Requirements

- Arduino Board
- Eye blink sensor
- Motor driver
- Motors
- Battery

Software Requirements

- Embedded C
- Arduino IDE

3.1. EMBEDDED SYSTEMS

3.1.1. *Overview of embedded systems*

An embedded system framework is a special-purpose computer system intended to perform one or a few dedicated functions, frequently with real-time computing requirements. It is usually embedded as a component of total device including hardware and mechanical parts. conversely, a useful pc, such as a pc, can do many various assignments relying on programming. Embedded systems have become viral today as they control a significant of the regular devices we use.

Since the embedded system is committed to explicit undertakings, pln designers can enhance it, lessening the size and cost of the item, or expanding the reliability and performance. Some embedded systems are mass-created,profiting 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. Intricacy ranges from low, with a solitary microcontroller chip, exceptionally high with various units, peripherals and networks mounted inside a huge frame or nook.

In general, "embedded system" is not a precisely characterized term, as numerous systems have some element of programmability. For example, Handheld pcs share few components with embedded systems — for example such as the operating systems and microprocessors which power them — but are not genuinely embedded systems, because they permit various applications to be stacked and peripherals to be connected.

Embedded systems provide several functions

- Monitor the environment; embedded systems read information from input sensors. This data is then processed and the outcomes are showed in some configuration.
- Control the environment; embedded systems provide and transmit statements for actuators.
- Transform the data; embedded systems transform the data to in some useful way, such as data compression/decompression .

In spite of interaction with the outside world via sensors and actuators is an essential aspect of embedded systems, these systems also give functionality particular to their applications. Embedded systems commonly execute applications such as control laws, limited state machines, and signal processing algorithms. These systems should also recognise and respond to mistakes in both the interior computing environment as well as the surrounding electromechanical systems.

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

- Communication

devices e.g.: modems, cellular

phones

- Home Appliances

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

- Control Systems

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

3.1.2 Block diagram of an embedded system:

An embedded system mainly consists an embedded processor. Many appliances that have a digital connection -- microwaves, VCRs, cars – use 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 1.1

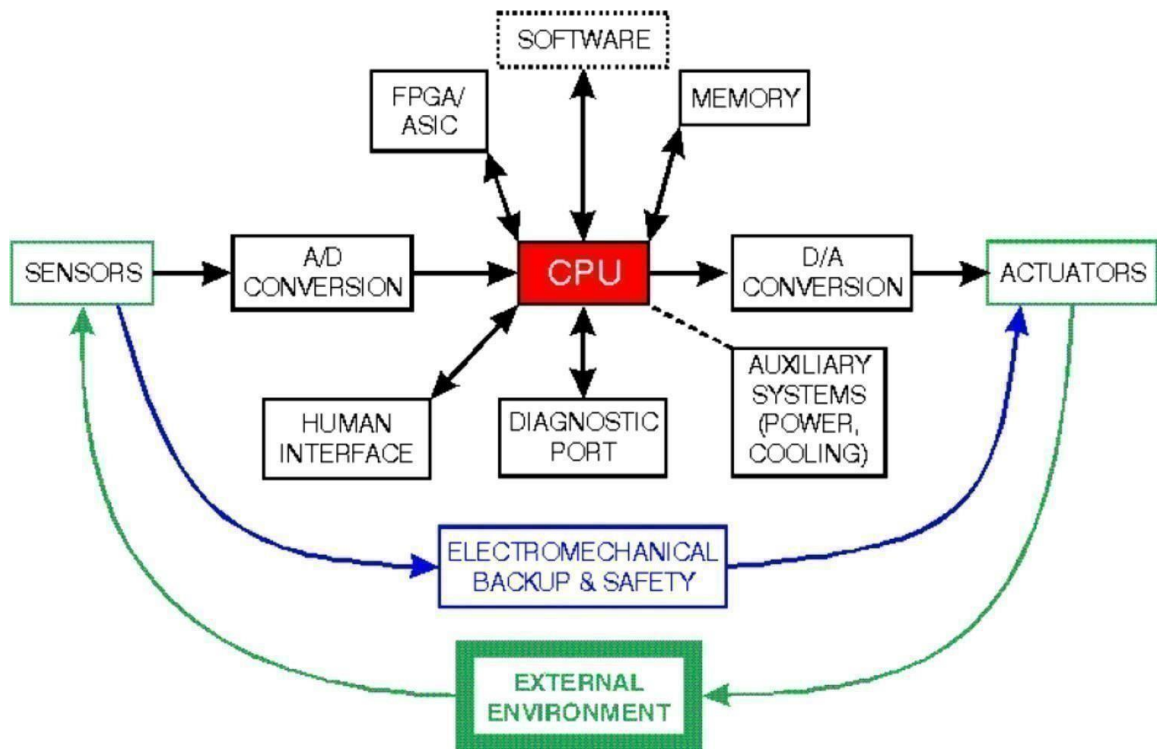


Fig 3.1.1 Block diagram of a typical embedded system

3.1.3. Characteristics of embedded systems

Embedded systems are represented by a unique set of characteristics. Each of these characteristics provide a particular set of design statements on embedded systems designers. The challenge to designing embedded systems is to confirm to the specific set of constraints for the application.

Application Specific Systems:

Embedded systems are not general-purpose pcs. Embedded system designs are upgraded for a specific application. A considerable lot of the work attributes are known before the hardware is planned. This permits the planner to focus on the particular design requirements of a much characterized application. Accordingly, there is limited user reprogrammability. Some embedded systems, however, require the adaptability of reprogrammability. Programmable DSPs are common for such applications.

Reactive Systems

As referenced before, an average embedded systems model reacts to the environment by means of sensors and controls the environment utilizing actuators. This requires embedded systems to run at the speed of the environment. This trait of an embedded system is called “responsive”. Reactive calculation implies that the system (primarily the software component) executes in light of outside occasions. Outside events can be either periodic or aperiodic. Periodic events make it simpler to schedule processing to guarantee performance. Aperiodic events are diligently scheduled. The most extreme arrival rate must be estimated in order to accommodate worst case situations. Most embedded systems have a significant reactive component. Probably 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). Real time system operation means that the correctness of a computation depends, in part, on the time at which it is delivered. This often leads to overly pessimistic estimates erring on the side of caution. Many embedded systems have a significant need 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 typical trait of an embedded system is one that comprises of imparting measures executing on few CPUs or ASICs which are associated by communication links. The justification for this is economy. Efficient 4 8-bit microcontrollers may be less expensive than a 32-bit processors. Even after adding the expense of the communication links, this methodology may be ideal. In this methodology, various processors are usually needed to deal with multiple time-basic tasks. Devices leveled out of embedded systems may likewise be genuinely conveyed.

Heterogeneous Architectures

Embedded systems regularly are made out of heterogeneous models (Fig 1.2). They may contain various processors in the similar system solution. They may also be contradicting signal systems. The combination of I/O interfaces, local and remote memories, and sensors and actuators makes embedded

MATERIALS AND METHODS**4.1. 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. A regulator circuit can utilize this DC input to give DC voltage that significantly has much less ripple voltage but also stays the same DC voltage value even the DC voltage shifts somewhat, or the load associated with the output DC voltage changes.

4.1.1 BATTERY

A battery is a device that stores chemical energy and converts it to electrical energy. The chemical reactions in a battery include the movement of electrons from one material (electrode) to another, through an outside circuit. The movement of electrons gives an electric current that can be utilized to do work.

4.1.2 REGULATORS:

The output voltage from the capacitor is more separated and at least controlled. The voltage regulator is a device, which keeps the output voltage steady regardless of the adjustment in supply variations, load variations & temperature changes. Here we use a fixed voltage regulator to be specific LM7805. The IC LM7805 is a +5V regulator which is utilized for microcontroller.

FEATURES AND DESCRIPTION OF REGULATORS

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection

- Output Transistor Safe Operating Area Protection

4.2 ARDUINO SOFTWARE:

Embedded c

- An Embedded C is quite possibly the most famous and most regularly used Programming Languages in the advancement of Embedded Systems. So, in this article, we will see portion of the Basics of Embedded C Program and the Programming Structure of Embedded C.
- Embedded C is the most famous languages among Embedded Programmers for programming Embedded Systems. There are numerous famous programming languages like Assembly, BASIC, C++ etc. that are frequently utilized for creating Embedded Systems. However Embedded C remains famous due to its productivity, less improvement time and movability.
- Prior to going into the fundamentals of Embedded C Program, we will initially investigate what an Embedded System is and the significance of Programming Language in Embedded Systems.

Embedded system

- An Embedded System can be best portrayed as a framework which has both the hardware and software and is intended to do a particular task. A best example for an Embedded System, which most families have, is a Washing Machine.
- We use washing machines almost everyday except wouldn't get the possibility of idea that it is an embedded system consisting of a Processor (and other hardware as well) and software.
- Embedded Systems can in addition to the fact that stand be separated from everyone else gadgets like washing machines be a piece of a lot bigger framework. A advanced day Car has a few individual embedded systems that work their particular task to make a smooth and safe excursion.
- A portion of the embedded systems in a Car are Anti-lock Braking System (ABS), Temperature Monitoring System, Automatic Climate Control, Tyre Pressure Monitoring System, Engine Oil Level Monitor, and so on.

Programming Embedded Systems:

- Embedded Systems comprises of both Hardware and Software. On the off chance if we consider a simply EmbeddedSystem,the principle Hardware Module is the Processor.The Processor is the core of the Embedded System and it tends to be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).
- Everyone of these devices make them thing in manner: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to show how the device inistially functions.
- Embedded Software or Program pemit Hardware and control external devices (Outputs) accordingly. During this cycle, the program for an Embedded System may need to straight forwardly control the interior and the interior architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Portsetc.
- From the above statement, plainly the Software part of an Embedded System is similarly essential to the Hardware part. There is no reason for having advanced Hardware Components with inadequately composed programs(Software).
- There are many programming languages that are utilized for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high - level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc.
- In the process of making a better embedded system, the programming of the system assumes a crucial part and thus, the choice of the Programming Language is very esseantial.

Factors for selecting the programming language:

The following are few factors that are to be considered while selecting the Programming Language for the advancement of Embedded Systems.

- **Size:** The memory that the program involves is very essential as Embedded Processors like Microcontrollers have a very restricted amount of ROM.
- **Speed:** The programs i.e.theymust run as fast as possible.The hardware should not be slowed down due to a slow running software.
- **Portability:** The similar program can be compiled for vrious processors.
- **Ease of Implementation**

Ease of Maintenance and Readability

Introduction to Embedded C Programming Language:

Prior to going in to the subtleties of Embedded C Programming Language and rudiments of Embedded C Program, we will initially discuss about the C Programming Language.

The C Programming Language, created by Dennis Ritchie in the late 60's and mid 70's, is the most famous and broadly utilized programming language. The C Programming Language gave low level memory access utilizing an easy compiler (a software that converts programs to machine code) and got the reward of efficient measuring to machine instructions.

The C Programming Language turned so famous that it is utilized in a broad range of applications from Embedded Systems to Super Computers.

Embedded C Programming Language, which is broadly utilized in the advancement of Embedded Systems, is an expansion of C Program Language. The Embedded C Programming Language utilizes the similar syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc.

The expansion in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc.

Basic Structure of an Embedded C Programming:

The next thing to understand in the Basics of Embedded C Program is the basic structure or Template of Embedded C Program. This will help us in understanding how an Embedded C Program is written.

The following part shows the basic structure of an Embedded C Program. Multiline

Comments Denoted using `/...../`
SingleLineComments Denoted using `//`
Preprocessor Directives `#include<...>` or `#define`
GlobalVariables Accessible anywhere in the program
Function Declarations Declaring Function
Main Function..... Main Function, execution begins here
{
LocalVariables Variables confined to mainfunction
FunctionCallsCalling other Functions
Infinite LoopLike `while(1)` or `for(;;)`
Statements

```

....
}
Function Definitions .....Defining the Functions
{
Local Variables..... Local Variables confined to this Function
Statements .....
....
....
}

```

Different components of an Embedded C Programme:

- **Comments:** Comments are discernible content that are composed to help us (the reader) understand the code without any problem. They are overlooked by the compiler and do not take up any memory in the final code (after compilation).

There are two different ways you can compose remarks: one is the single line remarks indicated by `//` and the other is multiline remarks denoted by `/*...*/`.

- **Preprocessor Directive:** A Preprocessor Directive in Embedded C is an indication to the compiler that it should glance in to this file for images that are not characterized in the program.

In C Programming Language (also in Embedded C), Preprocessor Directives are typically addressed using `#include...` or `#define...`

In Embedded C Programming, we mostly utilize the preprocessor directive to show a header file explicit to the microcontroller, which composes all the SFRs and the bits in those SFRs.

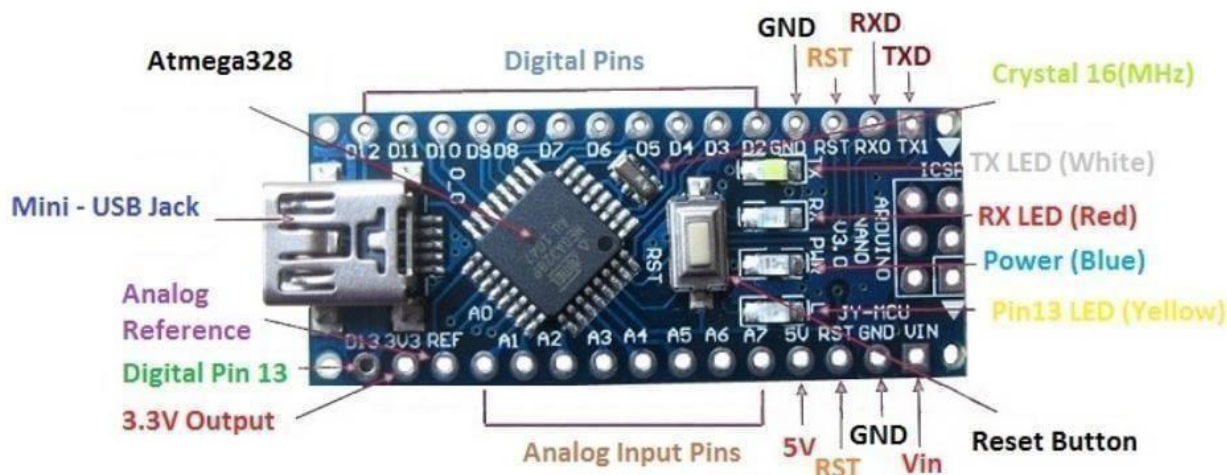
In case of 8051, Keil Compiler has the file “`reg51.h`”, which should be composed at the starting of every Embedded C Program.

- **Global Variables:** Global Variables, as the name recommends, are Global to the program i.e. they can get anywhere in the program.
- **Local Variables:** Local Variables, rather than Global Variables, are confined to their function .
- **Main Function:** Every C or Embedded C Program has one primary function, from where the execution of the program starts.

4.3

Arduino

Introduction to Aurdino Nano:



Arduino Nano

Fig 4.3.1 Aurdino Nano

Arduino Nano is a small, compatible, flexible and breadboard friendly Microcontroller board, developed by Arduino.cc in Italy, based on ATmega328p (Arduino Nano V3.x) / Atmega168 (Arduino Nano V3.x).

It comes with exactly the same functionality as in Arduino UNO but quite in small size.

It comes with an operating voltage of 5V, however, the input voltage can vary from 7 to 12V. Arduino Nano

Pinout contains 14 digital pins, 8 analog Pins, 2 Reset Pins & 6 Power Pins.

Every one of these Digital & Analog Pins are dole out with different functions but their primary function is to be arranged as input or output.

They are gone about as input pins when they are combined with sensors, but if you are giving some load then use them as output.

Functions like `pinMode()` and `digitalWrite()` are utilized to control the tasks of digital pins while `analogRead()` is used to control analog pins.

The analog pins come with a complete resolution of 10bits which measure the worth from zero to 5V.

Arduino Nano accompanies with a crystal oscillator of frequency 16 MHz. It is utilized to deliver a clock of exact frequency using consistent voltage.

There is one limitation utilizing Arduino Nano i.e. it doesn't accompany with DC power jack, means you can not pass outside power source through a battery.

This board doesn't usage standard USB for association with a computer, all things being equal, it comes with Mini USB support.

minusacle size and breadboard well disposed nature settle on this device an ideal decision for greater part of the applications where a size of the electronic segmnts are of extraordinary concern.

Flash memory is 16KB or 32KB that all relies on the Atmega board i.e Atmega168 accompanies with 16KB of flash memory while Atmega328 accompanies with a flash memory of 32KB.

Flash memory is used for storing code. The 2KB of memory out of absolute flash memory is utilized for a bootloader.

The SRAM can vary from 1KB or 2KB and EEPROM is 512 bytes or 1KB for Atmega168 and Atmega328 respectively.

This board is like to other Arduino boards accessible in the market, but the little size makes this board stand apart from others.

Following figure shows the specifications of Arduino Nano Board:

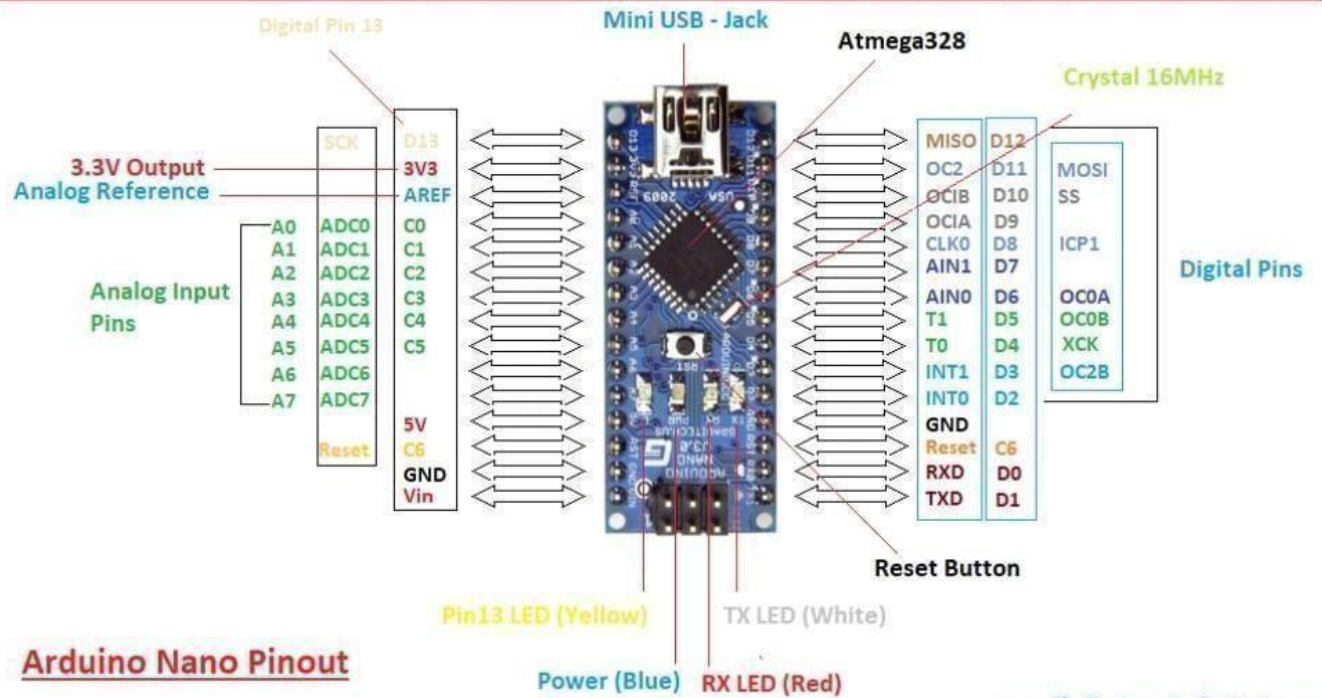
Introduction to arduino nano, intro to arduino nano, pin diagram of arduino nano, applications of arduino nano, arduino nano pinout, difference between Arduino nano and Arduino uno, arduino nano specifications

It is programmed using Arduino IDE which is an Integrated Development Environment that runs both offline and online.

No actions are required to run the board. All you need is board, mini USB cable and Arduino IDE software installed on the computer. USB cable is utilized to move the program from computer to the board.

No different burner is neded to compile and consume the program as this board accompanies an built-in boot-loader.

Aurdino Nano Pin Out:



Each pin on the Nano board accompanies with a particular function related with it. We can see the analog pins that can be utilized as an analog to digital converter where A4 and A5 pins can also be utilized for I2C communication. Relatively, there are 14 digital pins, out of which 6 pins are used for producing PWM.

Pin Description:

- **Vin:** It is input power supply voltage to the board when utilizing an outer power source of 7 to 12V.
- **5V:** It is a managed power supply voltage of the board that is utilized to control the regulator and other parts set on the board.
- **3.3V :** This is a minimum voltage produced by the voltage regulator on the board
- **GND.** These are the ground pins on the board. There are numerous ground pins on the board that can be combined relatively when more than one ground pin is needed.
- **Reset.** Reset pin is added on the board that resets the board. It is very useful when running program goes excessively and hangs up the board. LOW value to the reset pin will reset the regulator.

- **Analog Pins.** There are 8 analog pins associated with the board marked as A0 – A7. These pins are utilized to measure the analog voltage between 0 to 5V
- **Rx, Tx.** These pins are utilized for serial communication where Tx presents the transmission of data while Rx represents the data receiver.

Ardino Nano Specifications:

Microcontroller	Atmega328p/Atmega 168
Operating Voltage	5V
Input Voltage	7 – 12 V
Digital I/O Pins	14
PWM	6 out of 14 digital pins
Max. Current Rating	40mA
USB	Mini
Analog Pins	8
Flash Memory	16KB or 32KB
SRAM	1KB or 2KB
Crystal Oscillator	16 MHz
EEPROM	512bytes or 1KB
USART	Yes

Arduino Nano Specifications

Fig 4.3.3. Aurdino Nano Specifications

Communication and Programming:

The Nano device accompanies with an capacity to set up a communication with different controllers and computers. The sequential communication is done by the digital pins like pin 0 (Rx) and pin 1 (Tx) where Rx is utilizing for taking the Information and Tx is used for the transmission of information. The serial screen is added on the Arduino Software which is utilized to send written information to or from the board. FTDI drivers are also remembered in the software which carry as a virtual com port to the software.

The Tx and Rx pins accompany with an LED which blinks as the information is sent between FTDI and USB connection to the pc.

Arduino Software Serial Library is utilized for carrying out a sequqtial communication between the board and the pc.

Aside from sequential communication the Nano board also supports I2C and SPI communication. The Wire Library inside the Arduino Software is gotten to use the I2C bus.

The Arduino Nano is programmed by Arduino Software called IDE which is a most common software used for practically a wide range of board available. Simply download the software and select the board you are utilising. There are two choices to program the controller i.e either by the bootloader that is included in the software which liberates you free from the utilization of outside burner to compile and dump the program into the controller and other option is by utilizing ICSP(In-circuit serial programming header).

Arduino board software is similarly compatible with Windows, Linux or MAC, however, Windows are liked to use.

CHAPTER 5

RESULT:

Daily we are seeing handicapped people facing difficulties moving one place to another place so we are here to help the handicap people. by using this they can move easily from one place by another place just by blinking eye. This is a sample prototype we are preparing for handicap people. The principle of operation is simple. In this we used Arduino nano, ir sensor, motor driver, 2 motors, battery, regulator. First, we measure the distance required to move from one place to another place and then we write the code for our requirements and dump this code into microcontroller. The sensor detects the eye blink from the eye and give command to microcontroller which is present on Arduino nano. Already. Microcontroller has code which we have written and dumped on to micro controller. Micro controller takes the command from sensor and gives command to motor driver. Motor driver also has controller which drives the two motors. When we blink the eye the eye blink sensor gives command to micro controller. Based on how many blinks for suppose if three blinks are detected then microcontroller understands that the wheelchair should move to room two likewise micro controller gives command to motor driver and the motor driver has two controllers which drives the two motors. Like this our wheelchair moves.

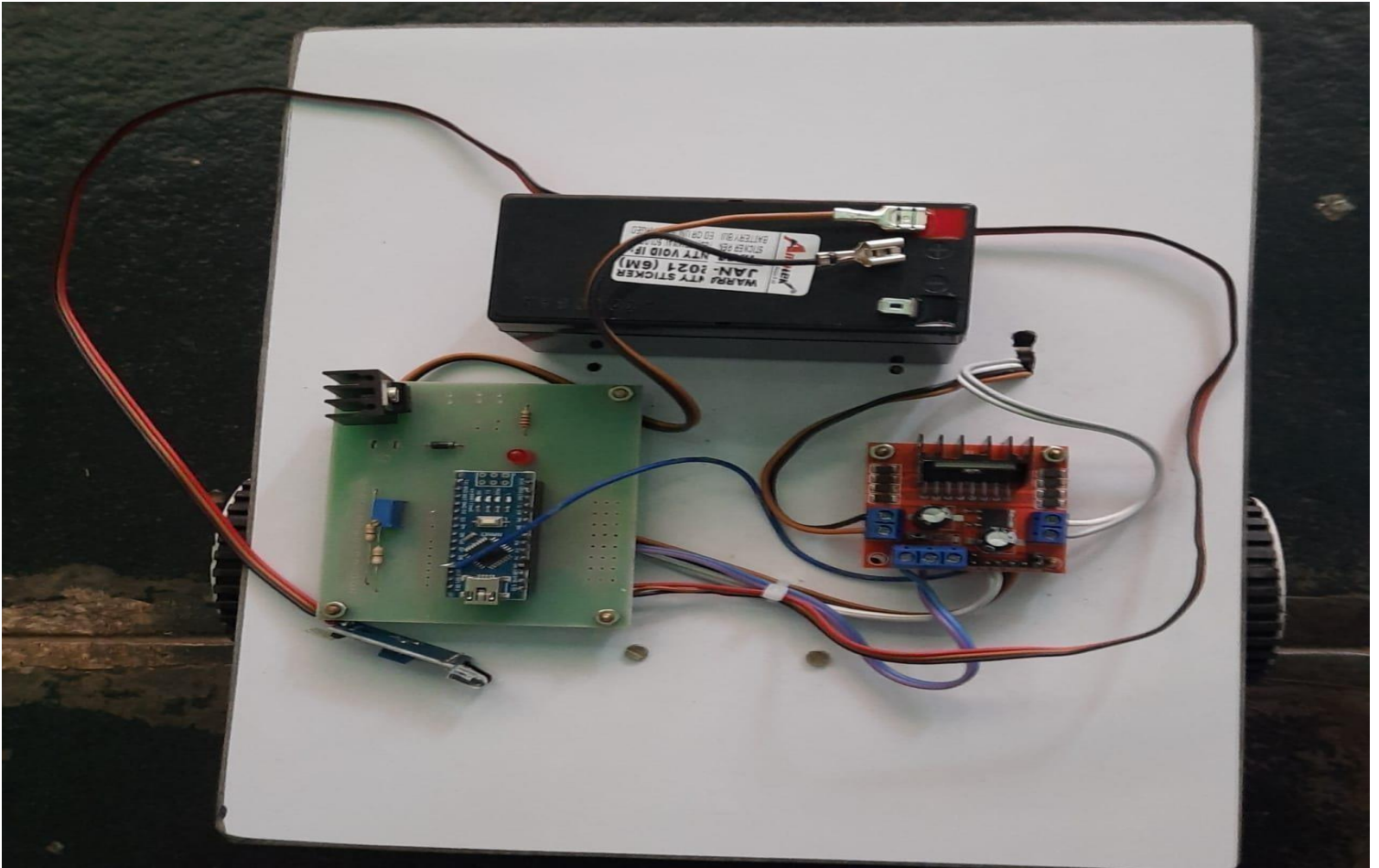


Fig 5.1.1: Demostration

```

eyeblink_virtual_map
byte eye = 14;

const int M11 = 2;
const int M12 = 3;
const int M21 = 4;
const int M22 = 5;
int pwm = 130;
unsigned long startTime;
unsigned long endTime;
unsigned long duration;
byte timerRunning;

int blinkcounter = 0;    // current state of the button
int lastblink = 0;
int lastcount = 0;
int i=0;

int a[200];
int j=2;

void setup() {
    // put your setup code here, to run once:
    Serial.begin (9600);

    pinMode(eye, INPUT);
    pinMode(M11, OUTPUT);
    pinMode(M12, OUTPUT);
    pinMode(M21, OUTPUT);
    pinMode(M22, OUTPUT);

```

Fig 5.1.2:Source Code



Fig 5.1.3 Output Checking

CHAPTER 6

CONCLUSION

- In conclusion, an eye-controlled wheelchair can be designed in a better way by using Arduino and C language.
- This prototype is useful for daily life activities for physically handicapped people.
- This is a simple prototype and is very useful.
- The IR sensor it detects the person's eye blink and moves to the room where ever they want.
- Daily we are seeing handicapped people facing difficulties moving one place to another place so we are here to help the handicapped people.
- By using this they can move easily from one place to another place just by blinking eye.
- This is a sample prototype we are preparing for handicapped people.
- Here with the help of IR sensor, Arduino and motor drivers we have designed this device.

CHAPTER 7

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] N. de Estadística y Censo, “Encuesta nacional sobre discapacidad 2018,” May 2019. [Online]. Available: <https://inec.cr/sites/default/files/documentos-biblioteca-virtual/reenadis2018.pdf>
- [6] S. Shinde, S. Kumar, and P. Johri, “A review: Eye tracking interface with embedded system iot,” in 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Sep. 2018, pp. 791–795.
- [7] U. Garg, K. K. Ghanshala, R. C. Joshi, and R. Chauhan, “Design and implementation of smart wheelchair for quadriplegia patients using iot,” in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Dec 2018, pp. 106–110.
- [8] A. Carrasquilla-Batista, K. Quiros-Espinoza, and C. Gomez-Carrasquilla, “An internet of things (iot) application to control a wheelchair through eeg signal processing,” in 2017 International Symposium on Wearable Robotics and Rehabilitation (WeRob), Nov 2017, pp. 1–1.
- [9] S. Khandani, “Engineering design process, education transfer plan,” 2005. [Online]. Available: <https://www.dphu.org/uploads/attachements/books/books25470.pdf>

- [10] M. Varela, "Raw eeg signal processing for bci control based on voluntary eye blinks," in 2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV), Nov 2015, pp. 1–6.
- [11] D. Purwanto, R. Mardiyanto, K. Arai: " Electric wheelchair control with gaze direction and eye blinking", Proceedings of The Fourteenth International Symposium on Artificial Life and Robotics, GS21-5, B-Con Plaza, Beppu, [2008].
- [12] Poonam S. Gajwani & Sharda A. Chhabria Eye Motion Tracking for Wheelchair Control.
- [13] The 8051 Microcontroller and Embedded Systems using Assembly and C – Second Edition
Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay
- [14] Arai K, Purwanto D (2007) Electric wheelchair control with the human eye. 7th International Conference on Optimization: Tech-niques and Applications, December 2007, Kobe, Japan2.

Plagiarism Checker X Originality Report



Plagiarism Quantity: 15% Duplicate

Date	Thursday, April 15, 2021
Words	769 Plagiarized Words / Total 5257 Words
Sources	More than 43 Sources Identified.
Remarks	Low Plagiarism Detected - Your Document needs Optional Improvement.

?????????CHAPTER 1 INTRODUCTION Wheelchairs were designed with the idea to help handicapped people to move around and accomplish daily life tasks, Many brilliant wheelchairs have been designed and with an increasing number of advances in technologies fresher and more intelligent wheelchairs are coming into the market to help the seriously debilitated people. Our wheelchair is likewise an endeavor to utilize straightforward and simple developments of the eye and head to begin, pause and move the wheelchair. In addition, we also connect the various device in a room to the wheelchair and communicate with it through a RF module. Literature Review: Existing System: ❖ In the existing system we can control 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.

Sources found:

Click on the highlighted sentence to see sources.

Internet Pages

- <1% <https://www.researchgate.net/publication>
- 1% https://www.academia.edu/34257983/loT_Ba
- <1% <https://dokumen.pub/programming-with-stm>
- 1% <https://www.academia.edu/28321108/Making>
- <1% <https://successessays.com/>
- <1% <https://www.scribd.com/doc/95579232/Amph>
- <1% <https://pt.scribd.com/document/222813530>
- 1% <https://www.arduino.cc/en/Main/arduinoBo>
- <1% https://www.academia.edu/31067213/GSM_ba
- <1% <https://www.academia.edu/40090962/FINGER>
- <1% <https://create.arduino.cc/projecthub/jit>
- 1% <https://www.apogeeweb.net/circuitry/L293>

◆ Eye squint sensor detects whether eye is open or shut. The eye-squint sensor works by enlightening the eye as well as eyelid region with infrared light and afterward checking the adjustments in the mirrored light utilizing 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. 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 detect 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. Fig 1.1.1 Block Diagram of Proposed System. CHAPTER 2 2.1 WORKING PRINCIPLE: Eye Blink Sensor: The eye blink sensor detects whether eye is open or shut. The eye blink sensor works by enlightening the eye as well as eyelid region with infrared light and afterward observing the adjustments in the mirrored light utilizing a photometer and diffrentiator circuit. The exact functionality depends greatly on the positioning and aiming of the emitter and detector with respect to the eye. Fig.2.1.1. Connection of eye blink sensor with spectacles Fig 2.1.2.

IR Sensor ◆ FEATUTRES: ◆ EYE BLINK sign by LED Instant yield advanced sign for straightforwardly Connecting to microcontroller Compact Size Working Voltage +5V DC APPLICATION: ◆ Digital Eye Blink screen for Vehicle Accident avoidance and. Appropriate for ongoing driving applications SPECIFICATION: Operating Voltage :+5V DC regulated Operating Current :100mA Output Data Level : TTL Level Eye Blink Indicated by LED and Output High Pulse USING SENSOR: Connect regulated DC power supply of 5 Volts. Dark wire is Ground, Next center wire is Brown which is yield and Red wire is positive supply. These wires are likewise set apart on PCB.To test sensor you just need power the sensor by interface two wires +5V and GND. You can leave the yield wire as its worth. At the point When Eye closed, LED is off & the output is at 0V.Put Eye blink sensor glass on the face inside 15mm distance, and you can see the LED blinking on each Eye blink.The yield is dynamic high for Eye close and can be offered straight forwardly to microcontroller for

<1%	https://www.slideshare.net/mahkamkhan/wi
<1%	https://core.ac.uk/download/pdf/29587917
1%	https://www.arcjournals.org/pdfs/ijirec/
<1%	http://staff.cs.upt.ro/~rbogdan/Content/
<1%	https://easysiteaussie815.weebly.com/dis
<1%	https://users.ece.cmu.edu/~koopman/iccd9
<1%	https://www.scribd.com/document/34589515
<1%	https://epdf.pub/the-firmware-handbook-e
1%	https://www.academia.edu/7654225/Introdu
<1%	https://findanyanswer.com/what-kind-of-e
<1%	https://byjus.com/physics/zener-diode-as
<1%	https://www.apogeeweb.net/productdetail/
2%	https://www.electronicshub.org/basics-of
1%	http://fabacademy.org/2020/labs/vigyanas
<1%	https://en.wikipedia.org/wiki/The_C_Prog
<1%	https://divakarvi.github.io/bk-spca/spca
<1%	https://www.onlineinterviewquestions.com
<1%	https://whataftercollege.com/data-scienc
<1%	https://issuu.com/davetheroyale/docs/pro
<1%	https://www.elprocus.com/basics-and-stru
<1%	https://www.totalphase.com/blog/2019/12/
<1%	https://www.gelbukh.com/clbook/Computati

connecting applications. EYE BLINK OUTPUT: 5V (High) ? LED ON When Eye is close. 0V (Low) ? LED OFF when Eye is open. Fig 2.1.3

Eye Demonstration 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 depicted by its makers as a ♦programming language and climate for individuals who need to program pictures, animation, and cooperations. ♦ The Arduino programming language use an open source project known as Wiring (wiring.org.co). The Arduino language is based on ordinary C. on the off chance that you are new to this language, don♦t stress; it♦s not difficult to learn, and the Arduino IDE gives some input when you commits errors in your programs.

The Arduino Uno has a various number of facilities for speaking with a computer, another Arduino, or other microcontrollers. The ATmega328 gives UART TTL (5V) sequential communication, which is accessible on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this sequential communication over USB and shows up as a virtual com port to software on the computer. The '16U2 firmware utilizes the standard USB COM drivers, and no outside driver is required. In any case, on Windows, a inf file is needed. The Arduino software includes a chronic screen which permits straight forward literary information is being sent through 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 takes into account serial communication on any of the Uno's digital pins. The ATmega328 likewise upholds I2C (TWI) and SPI communication. The Arduino software includes a Wire library to improve on utilization of the I2C bus; see the documentation for details. For SPI communication, utilize the SPI library. As you go through the rundown of programming articulations available in the Arduino IDE (choose Help->Reference), you may think there isn♦t lot of power for doing things like running servos, working stepper

2%	https://www.theengineeringprojects.com/2
1%	https://store.techmaze.ae/store/product/
<1%	https://sites.google.com/site/c70i910I/E
<1%	https://ukdiss.com/examples/system-for-e
<1%	https://www.slideshare.net/selvalakshmi2
<1%	https://en.wikipedia.org/wiki/Comparison
<1%	https://docplayer.net/26687055-For-your-
<1%	https://www.slideshare.net/saurabhbansal

motors, reading potentiometers, or showing text on an LCD. Like most any language dependent on C, the Arduino upholds the thought of libraries code Repositories that expand core programming functionality. Libraries let you re- use code without having to truly reorder it into every one of your projects.

The standard Arduino software establishment accompanies a few libraries you may utilize, and you can download others from the Arduino support pages and from outsider websites that distributes Arduino library code. A genuine illustration of a library you'll utilize with the Robot and robot projects is Servo. This library permits you to interface atleast one hobby R/C servos to the Arduino's digital I/O pins. The Servo library accompanies with the standard Arduino establishment bundle Library- >Servo. This adds the line `#include` Which tells the Arduino IDE that you wish to remember the Servo library for your sketch. With the functionality of the library now accessible to you, you can utilize its different capacities to control atleast servos. For example, you can utilize the write function to turn a servo to a particular position, from 0 to 180 degrees.

The following code `myServo.write(90);` Moves a servo to its midpoint, or 90 degree position. Fundamentally, Arduino draws are very direct and are really simple to peruse and understand. The Arduino program contains two fundamental parts: `setup ()` and `loop ()`. These are programming functions that do what their names purpose: `setup ()` sets up the Arduino equipment, such as indicating which I/O lines you intend to utilize, and whether They are inputs or yields. 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. Motor Driver: The L293D is a popular 16-Pin Motor Driver IC.

As the name recommends it is basically used to drive engines. A solitary L293D IC is equipped for running two DC engines simultaneously; likewise the direction of these two engines can be controlled autonomously. So if you have motors which has woking voltage under 36V and working current under 600mA, which are to be controlled by advanced circuits like Op-Amp, 555 clocks, digital gates or even Micron rollers like Arduino, PIC, ARM etc.. This IC will be the correct option for me. Utilising this L293D motor driver IC is extremely simple. The IC deals with the principle of Half H- Bridge, let us not dive excessevely deep into what H-Bridge implies,

but for now just realize that H bridge is a set up which is utilized to run motors both in clock wise and hostile to clockwise direction.

As said earlier this IC is equipped for running two motors at the any direction simultaneously , the circuit to got the equivalent is shown below. Fig 1.1.4 Motor Driver Pin Out All the Ground pins ought to be grounded. There are two power pins for this IC, one is the Vss(Vcc1) which gives the voltage for the IC to work, this should be associated with +5V. The other is Vs(Vcc2) which gives voltage for the motors to run, on the requirement of my motor i can connect this pin to everywhere 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 by and large we will be utilize both the motors and both the pins are held high ofcourse by associating with +5V supply.

The input pins Input 1,2 are utilized to control the motor 1 and Input pins 3,4 are utilized to control the Motor 2. The input pins are associated with the any Digital circuit or microcontroller to bear the speed and direction of the motor. You can flip the input pins dependent on the following table to control your motor. Applications
Used to drive high ebb and flow Motors utilizing Digital Circuits. Can be utilized to drive Stepper motors .
High current LEDs can be driven. Relay Driver module (Latching Relay is possible) CHAPTER 3
Requirements Hardware Requirements Arduino Board Eye blink sensor Motor driver Motors
Battery Software Requirements Embedded C Arduino IDE 3.1. EMBEDDED SYSTEMS 3.1.1.

Overview of embedded systems An embedded system framework is a special-purpose computer system intended to perform one or a few dedicated functions, frequently with real-time computing requirements. It is usually embedded as a component of total device including hardware and mechanical parts. conversely, a useful pc, such as a pc, can do many various assignments relying on programming. Embedded systems have become viral today as they control a significant of the regular devices we use. Since the embedded system is committed to explicit undertakings, pln designers can enhance it, lessening the size and cost of the item, or expanding the reliability and performance. Some embedded systems are mass-created,profiting 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.

Intricacy ranges from low, with a solitary microcontroller chip, exceptionally high with various units, peripherals and networks mounted inside a huge frame or nook. In general, "embedded system" is not an precisely characterized term, as numerous systems have some element of programmability. For example, Handheld pcs share few components with embedded systems for example such as the operating systems and microprocessors which power them but are not genuinely embedded systems, because they permit various applications to be stacked and peripherals to be connected. Embedded systems provide several functions Monitor the environment; embedded systems read information from input sensors. This data is then processed and the outcomes are showed in some configuration.

Control the environment; embedded systems provide and transmit statements for actuators. Transform the data; embedded systems transform the data took in some useful way, such as data compression/decompression . In spite of interaction with the outside world via sensors and actuators is an essential aspect of embedded systems, these systems also gives functionality particular to their applications. Embedded systems commonly execute applications such as control laws, limited state machines, and signal processing algorithms. These systems should also recognise and respond to mistakes in both the interior computing environment as well as the surrounding electromechanical systems.

There are numerous categories of embedded systems, from communication devices to home appliances to control systems. Examples include; Communication devices e.g.: modems, cellular phones Home Appliances e.g.: CD player, VCR, microwave oven Control Systems e.g.: Automobile anti-lock braking systems, robotics, satellite control. 3.1.2 Block diagram of an embedded system: An embedded system mainly consists an embedded processor. Many appliances that have a digital connection -- microwaves, VCRs, cars use 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 1.1. Fig 3.1.1 Block diagram of a typical embedded system 3.1.3.

Characteristics of embedded systems Embedded systems are represented by a unique set of characteristics. Each of these characteristics provide a particular set of design statements 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 pcs. Embedded system designs are upgraded for a specific application. A considerable lot of the work attributes are known before the hardware is planned. This permits the planner to focus on the particular design requirements of a much characterized application. Accordingly, there is limited user reprogrammability. Some embedded systems, however, require the adaptability of reprogrammability. Programmable DSPs are common for such applications.

Reactive Systems As referenced before, an average embedded systems model reacts to the environment by means of sensors and controls the environment utilizing actuators. This requires embedded systems to run at the speed of the environment. This trait of embedded system is called **responsive**. Reactive calculation implies that the system (primarily the software component) executes in light of outside occasions. Outside events can be either periodic or aperiodic. Periodic events make it simpler to schedule processing to guarantee performance. Aperiodic events are difficult to schedule. The most extreme arrival rate must be estimated in order to accommodate worst case situations. **Most embedded systems have a significant reactive component. Probably 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). Real time system operation means that the correctness of a computation depends, in part, on the time at which it is delivered. This often leads to overly pessimistic estimates erring on the side of caution. Many embedded systems have a significant need 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 typical trait of an embedded system is one that comprises of imparting measures executing on few CPUs or ASICs which are associated by communication links. The justification for this is economy. Efficient 4 8-bit microcontrollers may be less expensive than a 32-bit processors. Even after adding the expense of the communication links, this methodology may be ideal.

In this methodology, various processors are usually needed to deal with multiple time-basic tasks. Devices leveled out of embedded systems may likewise be genuinely conveyed. Heterogeneous Architectures Embedded systems regularly are made out of heterogeneous models (Fig 1.2). They may contain various processors in the similar system solution. They may also be contradicting signal systems. The combination of I/O interfaces, local and remote memories, and sensors and actuators makes embedded CHAPTER 4 MATERIALS AND METHODS 4.1. POWER SUPPLY All electronic circuits works only in low DC voltage, so we need a power supply unit to provide the appropriate voltage supply for their proper functioning. A

regulator circuit can utilize this dc input to give dc voltage that significantly has much less wave voltage but also stays the same dc esteem value even the dc voltage shifts some what, or the load associated with the output dc voltages changes. 4.1.1 BATTERY A battery is a device that stores chemical energy and converts it to electrical energy. The chemical reactions in a battery include the movement of electrons from one material (electrode) to another, through an outside circuit. The movement of electrons gives an electric current that can be utilized to care of work. 4.1.2 REGULATORS: The yield voltage from the capacitor is more separated and at last controlled. The voltage regulator is a device, which keeps the output voltage steady regardless of the adjustment in supply varieties, load variations & temperature changes. Here we use fixed voltage regulator to be specific LM7805. The IC LM7805 is a +5v regulator which is utilized for microcontroller.

FEATURES AND DESCRIPTION OF REGULATORS ♦ Output Current up to 1A ♦ Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V ♦ Thermal Overload Protection ♦ Short Circuit Protection ♦ Output Transistor Safe Operating Area Protection 4.2 ARDUINO SOFTWARE: Embedded C ♦ An Embedded C is quite possibly the

most famous and most regularly used Programming Languages in the advancement of Embedded Systems. So, in this article, we will see portion of the Basics of Embedded C Program and the Programming Structure of Embedded C. ♦ Embedded C is the most famous languages among Embedded Programmers for programming Embedded Systems. There are numerous famous programming languages like Assembly, BASIC, C++ etc. that are frequently utilized for creating Embedded Systems. However Embedded C remains famous due to its productivity, less improvement time and movability.

♦ Prior to going into the fundamentals of Embedded C Program, we will initially investigate what an Embedded System is and the significance of Programming Language in Embedded Systems. Embedded system ♦ An Embedded System can be best portrayed as a framework which has both the hardware and software and is intended to do a particular task. A best example for an Embedded System, which most families have, is a Washing Machine. ♦ We use washing machines almost everyday except wouldn't get the possibility of idea that it is an embedded system consisting of a Processor (and other hardware as well) and software. ♦ Embedded Systems can in addition to the fact that stand be separated from everyone else gadgets like washing machines be a piece of a lot bigger framework.

A advanced day Car has a few individual embedded systems that work their particular task to make a smooth and safe excursion. ♦ A portion of the embedded systems in a Car are Anti-lock Braking System (ABS), Temperature Monitoring System, Automatic Climate Control, Tyre Pressure Monitoring System, Engine Oil Level Monitor, and so on. Programming Embedded Systems: ♦ Embedded Systems comprises of both Hardware and Software. On the off chance if we consider a simply Embedded System, the principle Hardware Module is the Processor. The Processor is the core of the Embedded System and it tends to be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).

♦ Everyone of these devices make them thing in manner: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to show how the device initially functions. ♦ Embedded

Software or Program permit Hardware and control external devices (Outputs) accordingly. During this cycle, the program for an Embedded System may need to straight forwardly control the interior and the interior architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc. ♦ From the above statement, plainly the Software part of an Embedded System is similarly essential to the Hardware part. There is no reason for having advanced Hardware Components with inadequately composed programs (Software).

♦ There are many programming languages that are utilized for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high - level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc. ♦ In the process of making a better embedded system, the programming of the system assumes a crucial part and thus, the choice of the Programming Language is very essential. Factors for selecting the programming language: The following are few factors that are to be considered while selecting the Programming Language for the advancement of Embedded Systems. ♦ Size: The memory that the program involves is very essential as Embedded Processors like Microcontrollers have a very restricted amount of ROM. ♦ Speed: The programs i.e. they must run as fast as possible.

The hardware should not be slowed down due to a slow running software. ♦ Portability: The similar program can be compiled for various processors. ♦ Ease of Implementation ♦ Ease of Maintenance and

Readability Introduction to Embedded C Programming Language: Prior to going in to the subtleties of Embedded C Programming Language and rudiments of Embedded C Program, we will initially discuss about the C Programming Language. The C Programming Language, created by Dennis Ritchie in the late 60s and mid 70s, is the most famous and broadly utilized programming language. The C Programming Language gave low level memory access utilizing an easy compiler (a software that converts programs to machine code) and got the reward of efficient measuring to machine instructions.

The C Programming Language turned so famous that it is utilized in a broad range of applications from Embedded Systems to Super Computers. Embedded C Programming Language, which is broadly utilized in

the advancement of Embedded Systems, is an expansion of C Program Language. The Embedded C Programming Language utilizes the similar syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc. The expansion in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc.

Basic Structure of an Embedded C Programming: The next thing to understand in the Basics of Embedded C Program is the basic structure or Template of Embedded C Program. This will help us in understanding how an Embedded C Program is written. The following part shows the basic structure of an Embedded C Program.

Multiline Comments Denoted using /* */ Single Line Comments Denoted using // Preprocessor Directives
#include< > or #define Global Variables Accessible anywhere in the program Function Declarations
Declaring Function Main Function Main Function, execution begins here { Local Variables Variables confined to mainfunction Function Calls Calling other Functions Infinite Loop Like while(1) or for(;;) Statements ; .

} Function Definitions Defining the Functions { Local Variables Local Variables confined to this Function
Statements ; . } Different components of an Embedded C Programme: Comments: Comments are discernible content that are composed to help us (the reader) understand the code without any problem. They are overlooked by the compiler and do not take up any memory in the final code (after compilation). There are two different ways you can compose remarks: one is the single line remarks indicated by // and the other is multiline remarks denoted by /* */.

Preprocessor Directive: A Preprocessor Directive in Embedded C is an indication to the compiler that it should glance in to this file for images that are not characterized in the program. In C Programming Language (also in Embedded C), Preprocessor Directives are typically addressed using #include or #define. In Embedded C Programming, we mostly utilize the preprocessor directive to show a header file explicit to the microcontroller, which composes all the SFRs and the bits in those SFRs. In case of 8051, Keil Compiler has the file reg51.h, which should be composed at the starting of every Embedded C Program. Global

Variables: Global Variables, as the name recommends, are Global to the program i.e. they can get anywhere in the program.

❖ Local Variables: Local Variables, rather than Global Variables, are confined to their function . ❖ Main

Function: Every C or Embedded C Program has one primary function, from where the execution of the program starts. 4.3 Arduino Introduction to Aurdino Nano: Fig 4.3.1 Aurdino Nano Arduino Nano is a small, compatible, flexible and breadboard friendly Microcontroller board, developed by Arduino.cc in Italy, based on ATmega328p (Arduino Nano V3.x) / Atmega168 (Arduino Nano V3.x). It comes with exactly the same functionality as in Arduino UNO but quite in small size. It comes with an operating voltage of 5V, however, the input voltage can vary from 7 to 12V. Arduino Nano Pinout contains 14 digital pins, 8 analog Pins, 2 Reset Pins & 6 Power Pins. Every one of these Digital & Analog Pins are dole out with different functions but their primary function is to be arranged as input or output.

They are gone about as input pins when they are combined with sensors, but if you are giving some load then use them as output. Functions like pinMode() and digitalWrite() are utilized to control the tasks of digital pins while analogRead() is used to control analog pins. The analog pins come with a complete resolution of 10bits which measure the worth from zero to 5V. Arduino Nano accompanies with a crystal oscillator of frequency 16 MHz. It is utilized to deliver a clock of exact frequency using consistent voltage. There is one limitation utilizing Arduino Nano i.e. it doesn't accompany with DC power jack, means you can not pass outside power source through a battery. This board doesn't usage standard USB for association with a computer, all things being equal, it comes with Mini USB support.

minusucle size and breadboard well disposed nature settle on this device an ideal decision for greater part of the applications where a size of the electronic segmnts are of extraordinary concern. Flash memory is 16KB or 32KB that all relies on the Atmega board i.e Atmega168 accompanies with 16KB of flash memory while Atmega328 accompanies with a flash memory of 32KB. Flash memory is used for storing code. The 2KB of memory out of absolute flash memory is utilized for a bootloader. The SRAM can vary from 1KB or 2KB and

EEPROM is 512 bytes or 1KB for Atmega168 and Atmega328 respectively. This board is like to other Arduino boards accessible in the market, but the little size makes this board stand apart from others.

Following figure shows the specifications of Arduino Nano Board: Introduction to arduino nano, intro to arduino nano, pin diagram of arduino nano, applications of arduino nano, arduino nano pinout, difference between Arduino nano and Arduino uno, arduino nano specifications It is programmed using Arduino IDE which is an Integrated Development Environment that runs both offline and online. No actions are required to run the board. All you need is board, mini USB cable and Arduino IDE software installed on the computer. USB cable is utilized to move the program from computer to the board. No different burner is needed to compile and consume the program as this board accompanies an built-in boot-loader. Aurdino Nano Pin Out: Fig 4.3.2 Aurdino Nano Pinout Each pin on the Nano board accompanies with a particular function related with it.

We can see the analog pins that can be utilized as an analog to digital converter where A4 and A5 pins can also be utilized for I2C communication. Relatively, there are 14 digital pins, out of which 6 pins are used for producing PWM. Pin Description: ♦ Vin: It is input power supply voltage to the board when utilizing an outer power source of 7 to 12 V. ♦ 5V: It is a managed power supply voltage of the board that is utilized to control the regulator other and parts set on the board. ♦ 3.3V :This is a minimum voltage produced by the voltage regulator on the board ♦ GND. These are the ground pins on the board. There are numerous ground pins on the board that can be combined relatively when more than one ground pin is needed. ♦ Reset. Reset pin is added on the board that resets the board. It is very useful when running program goes excessively and hangs up the board. LOW value to the reset pin will reset the regulator. ♦ Analog Pins.

There are 8 analog pins associated with the board marked as A0 ♦ A7. These pins are utilized to measure the analog voltage between 0 to 5V ♦ Rx, Tx. These pins are utilized for serial communication where Tx presents the transmission of data while Rx represents the data receiver. Aurdino Nano Specifications: Fig 4.3.3. Aurdino Nano Specifications Communication and Programming: The Nano device accompanies with an capacity to set up a communication with different controllers and computers. The sequential communication is

done by the digital pins like pin 0 (Rx) and pin 1 (Tx) where Rx is utilized for taking the Information and Tx is used for the transmission of information.

The serial screen is added on the Arduino Software which is utilized to send written information to or from the board. FTDI drivers are also remembered in the software which carry as a virtual com port to the software. The Tx and Rx pins accompany with an LED which blinks as the information is sent between FTDI and USB connection to the pc. Arduino Software Serial Library is utilized for carrying out a sequential communication between the board and the pc. Aside from sequential communication the Nano board also supports I2C and SPI communication. The Wire Library inside the Arduino Software is gotten to use the I2C bus. The Arduino Nano is programmed by Arduino Software called IDE which is a most common software used for practically a wide range of board available. Simply download the software and select the board you are utilising.

There are two choices to program the controller i.e either by the bootloader that is included in the software which liberates you free from the utilization of outside burner to compile and dump the program into the controller and other option is by utilizing ICSP (In-circuit serial programming header). **Arduino board software is similarly compatible with Windows, Linux or MAC, however, Windows are liked to use.** CHAPTER 5

RESULT: Daily we are seeing handicapped people facing difficulties moving one place to another place so we are here to help the handicap people.by using this they can move easily from one place by another place just by blinking eye.

This is a sample prototype we are preparing for handicap people. The principle of operation is simple. In this we used Arduino nano, ir sensor, motor driver,2motors, battery, regulator. First, we measure the distance required to move from one place to another place and then we write the code for our requirements and dump this code into microcontroller. The sensor detects the eye blink from the eye and give command to microcontroller which is present on Arduino nano. Already. Microcontroller has code which we have written and dumped on to micro controller. Micro controller takes the command from sensor and gives command to motor driver. Motor driver also has controller which drives the two motors. When we blink the eye the eye blink

sensor gives command to micro controller.

Based on how many blinks for suppose if three blinks are detected then microcontroller understands that the wheelchair should move to room two likewise micro controller gives command to motor driver and the motor driver has two controllers which drives the two motors. Like this our wheelchair moves. Fig 5.1.1:

Demostration Fig 5.1.2:Source Code Fig 5.1.3 Output CheckingCHAPTER 6 CONCLUSION ◆ In

conclusion,a Eye controlled wheel chair can be designed in a better way byusing ardino and c language . ◆

This prototype is useful for daily life activities for physically handicappedpeoples. ◆ This is a simple proto type and is very useful. ◆ The IR sensor it detects the person eye blink and moves to the room where ever they want. ◆ Daily we are seeing handicapped people facing diffucties moving one place toanother place so we are here to help the handicaped people. ◆ By using this they can move easily from one place by another place just by blinking eye.

◆ This is a sample prototype we are preparing for handicaped people. ◆ Here with the help of ir sensor,ardino and motor drivers we have designed thisdevice.

