HEAD MOVEMENT BASED WHEEL CHAIR FOR PARALYSED PEOPLE

A PROJECT REPORT

Submitted by

BHARATHVAJAN R	**211419105021**
KISHOREKUMAR V	**211419105067**
DINESH KANNA B	**211419105032**
DEEPAK KUMAR K	**211419105025**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRICAL AND ELECTRONICS ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"HEAD MOVEMENT BASED WHEELCHAIR FOR PARALYSED PEOPLE"** is the bonafide work of **"BHARATHVAJAN R (211419105021) , KISHOREKUMAR V (211419105067), DINESH KANNA B (211419105032), DEEPAK KUMAR K (211419105027)"** who carried out the project work under my supervision.

Dr. S.SELVI, M.E, Ph.D.	**Mr. K.KIRUBAKARAN, M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	**ASST PROFESSOR**
Department of Electrical and	Department of Electrical and
Electronics Engineering,	Electronics Engineering,
Panimalar Engineering College,	Panimalar Engineering College,
Chennai-600 123.	Chennai-600 123.

Submitted for End Semester Project Viva Voce held on 12.04.23 at Panimalar Engineering College, Chennai**.**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Our sincere thanks to our Honourable Founder and Chairman, **Dr.JEPPIAAR,M.A.,B.L.,Ph.D.,** for his sincere endeavour in educating us in his premier institution.

We would like to express our deep gratitude to our beloved **Secretary and Correspondent, Dr.P.CHINNADURAI, M.A.,M.Phil.,Ph.D** for his enthusiastic motivation which inspired us a lot in completing this project and our sincere thanks to our Directors **Mrs.C.VIJAYA RAJESWARI, Dr.C.SAKTHI KUMAR, M.E.,Ph.D** and **Dr.SARANYASREE SAKTHIKUMAR,B.E,M.B.A,Ph.D** for providing us with the necessary facilities for the completion of this project.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our **Head of the Department**, **Dr.S.SELVI, M.E., Ph.D., Professor and Head, Department of Electrical and Electronics Enginering** for her encouragement.

Personally, we thank our Guide **Mr.K.KIRUBAKARAN,M.E., in Department of Electrical and Electronics engineering** for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our sincere thanks to the project coordinators **Dr.S.DEEPA & Dr.N.MANOJ KUMAR,M.E.,Ph.D., in Department of Electrical and Electronics Engineering** for the Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, faculty and non-teaching staff members of our department, friends, well-wishers who have helped us for the successful completion of our project.

ABSTRACT

With the advancement of science and Technology an innovative mechanism is adopted to provide the assistance for physically challenged personalities. Spinal card injury persons and physically challenged persons required human assistance to move from one point to another point. The spinal card injury makes the people to paralysed, by this they need to depend on others for their movement. Electronics interfaced Wheel chair had designed to move without any assistance. Micro Electro Mechanical Systems (MEMS) based accelerometer sensor is used to direct the movement of the device. The wheel chair may move forward, backward, left and right directions based on the output signal of the MEMS device. And also wheel chair is controlled by voice commands through google assistant. The developed wheel chair is best suited to meet the challenges of the physically challenges personalities and also provide them an easy control of wheel chair.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF ABBREVIATIONS

WHO	World Health Organization
IEEE	Institute of Electrical and Electronics Engineers
FCR	False Command Rate
EEG	Electro Encephalo Gram
ESP	Extrasensory Perception
Wi-Fi	Wireless Fidelity
MPU	Memory Protection Unit
LED	Light Emitting Diode
MEMS	Micro- elctromechanical Systems
IOT	Internet Of Things
SOC	System of Chip
TSMC	Taiwan Semiconductor Manufacturing Company limited
SAR	Specific Absorption Rate
PWM	Pulse Width Modulation
IC	Integrated Circuit
IDE	Integrated Development Environment

LIST OF TABLE

CHAPTER 1

INTRODUCTION

It’s a widespread nature of a human being to walk independently as a by-born nature. But sometimes, some reason causes the interruption of this character. Superior stage of spiral cord damage, accident and brain as well as nervous system anarchy are such types of cases which causes inhabitants to loss control of arms or legs or together arms. In addition, physical disorder at the time of born also makes people immobilize. However, wheelchair is the way to make this group to being able to move. But, in the occasion of upper limb injury, it restricts some to the use of manual wheelchair, which is operated by arm’s muscular strength. So, they have to depend on others to give the manual force to move the wheelchair. To remove this enslavement and progress the feature of their life, present science made a progression on wheelchair history, which eliminates some of the mobility problems. Electric Powered Wheelchair is the solution, which is commonly branded as Automatic Wheelchair. The wheelchair where an electric motor moves the chair in spite of labor-intensive power as known as motorized wheelchair, electric wheelchair power chair, or electric-powered wheelchair.

Old citizens or disabled persons become dependent on other members of the family to navigate through their habitat or within residence. A smart wheel chair can be a useful assistant for them. Recent development in the field of robotics, automation, embedded system, artificial intelligence etc. can be combinedly utilised to design such a wheel chair. It can be controlled wirelessly adopting proper communication system. The chair can be controlled by head gesture as well as hand gesture method with directions as needed. The previous development of this kind of wheel chair is using a laptop or PC on the wheel chair [1]. By this development the recent wheel chairs are gesture controlled or voice controlled [2]. But the limitations of this kind of technologies is that the wheel chair is getting too bulky and it is to be controlled only by sitting on it. That’s why these types of wheel chairs are not giving satisfactory feedback from the users. The proposed model makes the wheel chair a lot easier to assemble and simple in the use, in addition the cost of manufacturing also gets reduced.

A major task of designing a smart wheelchair control is to produce sufficient commands rapidly and accurately including left and right turns, forward and backward motions, stopping, acceleration and deceleration. The facilities provided by a typical wheelchair require people with disabilities having intact manipulation ability like to use a joystick and operate the vehicle. However, many people with disabilities can not manipulate mechanical device and thus unable to use typical wheelchair. People with limited muscular function usually suffer from Amylotropic Lateral Sclerosis (ALS), paralysis and other Motor disabilities which causes loss of body movement.

1.1 TYPES OF WHEELCHAIR

1.1.1 All-Terrain Wheelchair

One of the most unusual types of power wheelchairs, these are niche products, but well worth a mention. They feature large inflated tires (and sometimes tank tracks!) with deep treads that enable them to go over just about anyterrain.

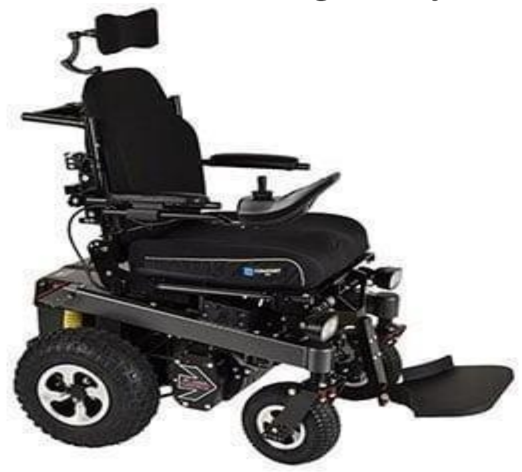


Fig1.1 All-Terrain Wheelchair

1.1.2 Manual Wheelchair

These are what most people think about when you mention the phrase. Most commonly seen in hospitals and nursing homes, they are the most economical choice for most people.

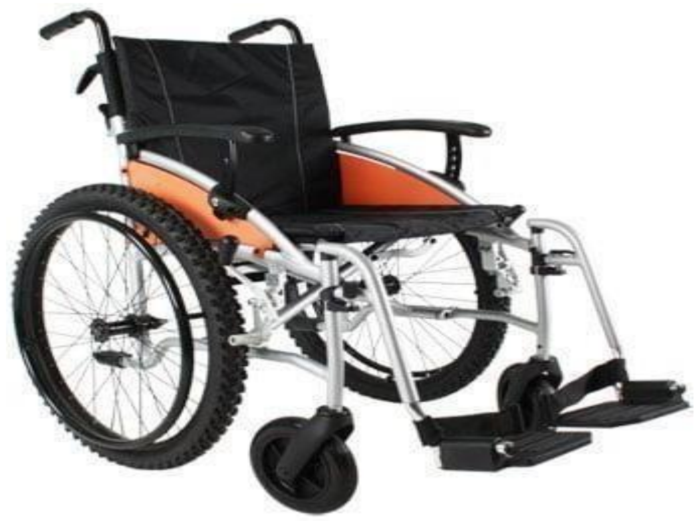


Fig 1.2 Manual Wheelchair

1.1.2.1 Types of Manual Wheelchair

We can also divide this sub-category of wheelchairs according to their use:

1. Lightweight
2. Sports
3. Pediatric
4. Reclining
5. Transport
6. Heavy Duty
7. Hemi-Height

1.1.3 Electrical Wheelchair

Electric wheelchairs are far and away the most popular variety because they allow those who use them greater freedom of mobility without having to rely on a nurse or other assistant, except perhaps for getting into the chair.

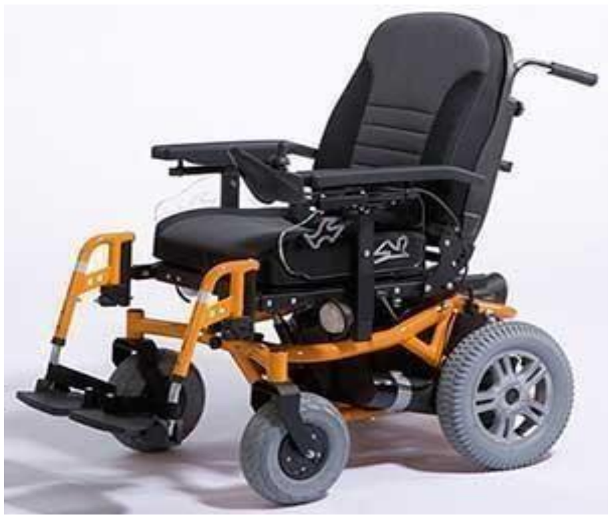


Fig 1.3:Electrical Wheelchair

1.1.4 Airplane Wheelchair

This is another niche product, primarily used by airlines to facilitate transport of people who have disabilities, but if you have mobility issues and travel frequently,you can certainly invest in one privately as well.

They tend to be smaller and lighter than traditional manual wheelchairs, being designed to pass between the rows of seats on an airplane, and come withsafety buckles so that you can buckle up while in flight, just like all the other passengers.

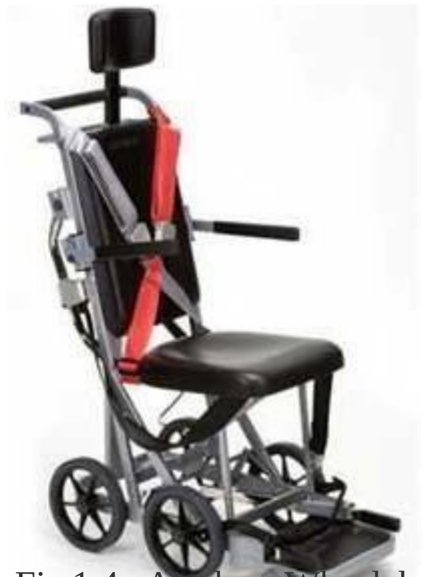


Fig 1.4 : Airplane Wheelchair

1.1.5 Beach Wheelchair

Another highly specialized design, these are almost always made from PVC or hollow aluminum tubes, with almost cartoonishly large wheels to make it easier to traverse the sand. Just because you have mobility issues doesn't mean you have to be limited in the things you can do or enjoy!

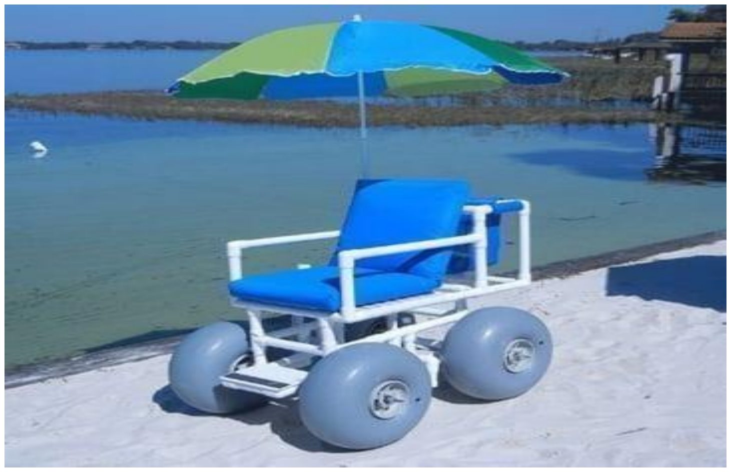


Fig 1.5: Beach Wheelchair

1.1.6 Bariatric Wheelchair

This is another highly-specialized type of chair, expressly designed with obese patients in mind. Constructed on a sturdier frame with a wider seat, these chairs provide the full range of mobility options to people who suffer from extreme weight issues.

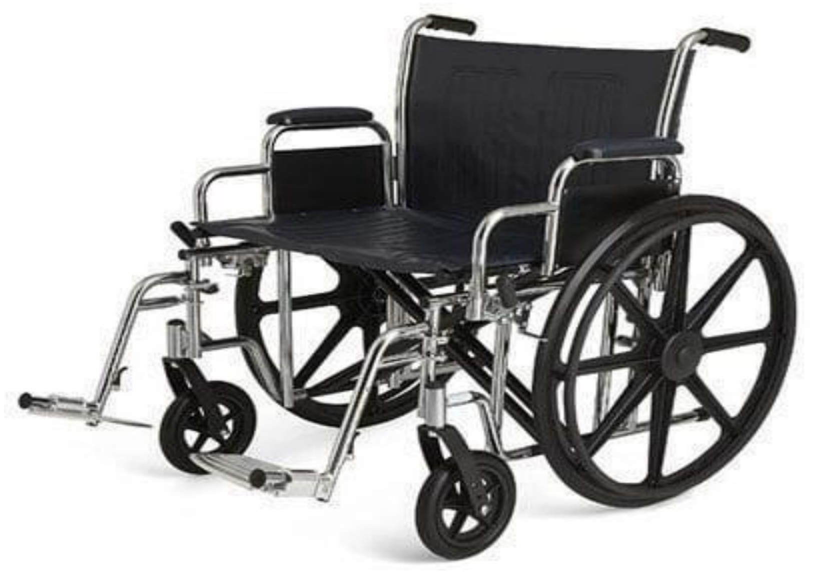


Fig 1.6 : Bariatic Wheelchair

1.1.7 Ergonomic Wheelchair

An ergonomic wheelchair is designed as a synthesis of a standard, manual wheelchair, merged with an ergonomic office chair with superior lumbar support. They're ideal chairs for people who spend extended periods of time sitting down**.**

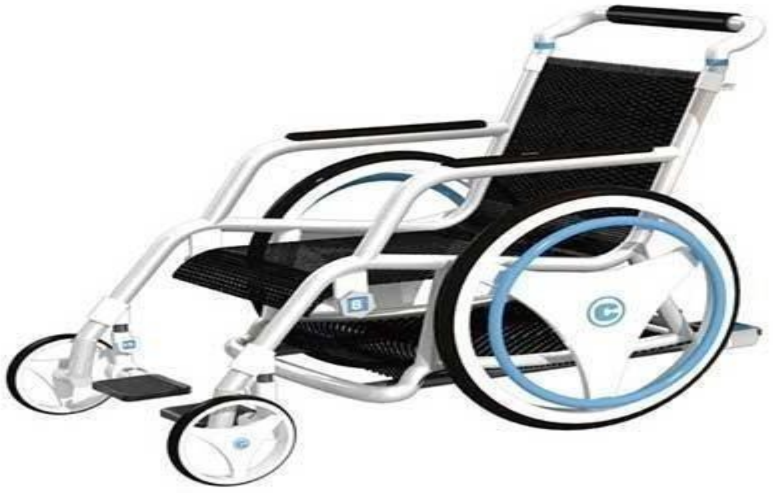


Fig 1.7: Ergonomic Wheelchair

1.1.8 Pediatric Wheelchair

Kid-sized wheelchairs, these are functionally identical to their full-sized cousins, just made smaller to better fit their users.

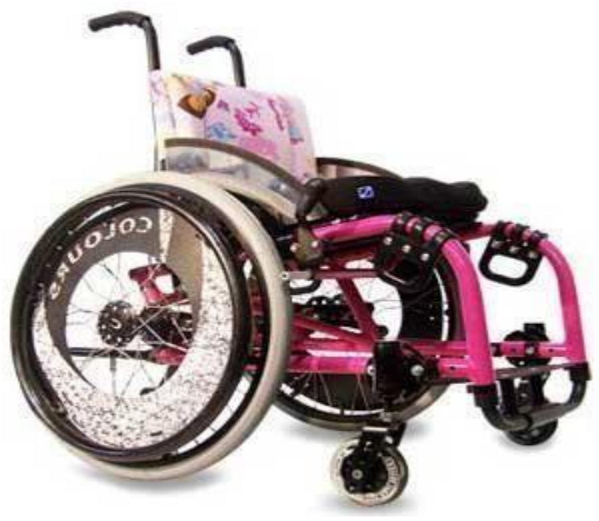


Fig 1.8 : Pediatric Wheelchair

1.1.9 Reclining Wheelchair

An exceptional niche design, offering all the benefits of a recliner, rolled into a wheelchair. These designs come with a pillow to offer greater comfort when resting in a reclined position. While it’s possible to nap in chairs of this type, you’ll almost certainly want extra padding if this is your intention.

1.1.10 Single Arm Wheelchair

If there’s one limitation of manual wheelchairs, it is that the user, by necessity, needs both arms to propel the chair. If you try to just use a single arm, you’ll wind up going in circles and get nowhere fast. The single-arm drive wheelchair solves this problem by connecting the two wheels by a specially-shapedaxle, allowing you to move forward using only one arm.

1.1.11 Sports (Racing) Wheelchair

Sports wheelchairs are constructed the same way as a standard wheelchair, except that the large rear wheels are angled in toward the seat**,** allowing for tighter turns, and enabling users to engage in a full range of sports activities (tennis, basketball, etc.).

1.1.12 Standing Wheelchair

As the name implies, this type of wheelchair is designed for use while standing. Most models allow for reconfiguration so they can be utilized in both seated and standing positions. While not the most technically advanced design currently available, this type of chair has the distinction of turning the user into a kind of living Transformer, just like in the movies!

1.1.13 Tilt Wheelchair

In the world of wheelchairs, there’s a raging debate about which is superior: reclining or tilt wheelchairs? They both accomplish the same basic goal,although they achieve it via different means.
These are the best types of wheelchairs for cerebral palsy patients if a reclinefeature is desired.

1.1.14 Ultralight Wheelchair

Ultra-light wheelchairs are, as their name implies, significantly lighter than theirmainstream counterparts. To accomplish this, they make certain concessions, including a shortened seat back, and being constructed with lighter-weight materials.

1.1.15 Wheelchair Stretcher

An intriguing design not typically purchased for home use, but used by EMTs and hospital professionals. They’re lightweight wheelchairs that can recline 180 degrees, effectively becoming a stretcher when the catch is released. In the picture below, note the presence of two sets of handles. One for use when in wheelchair mode, and another for use when in stretcher mode.

1.1.16 The iBot

There is one additional type of wheelchair worth mentioning. We didn’t list it among the major types, because it’s cutting edge technology and not yet widely available, but that’s changing.

Fig 1.9 : iBot

1.2 Disabled peoples around the world

There are currently more than 1 billion disabled people in the world. According to the World Health Organization (WHO) a disabled person is anyone who has "a problem in body function or structure, an activity limitation, has a difficulty in executing a task or action; with a participation restriction".

75 million people need a wheelchair on a daily basis. This represents 1% of the world's population. That's twice Canada's population!

In addition to born disabilities, due to accidents there are many people lost their legs due to which their mobility are affected. The need to depend on others for their movement. Spinal card injury may cause people paralysed. Partial paralysed or fully paralysed made them to need wheelchair for their movement.

Other than these things due to aging many people lack the stamina to move. These elderly people need wheelchair for their movement.

CHAPTER 2

LITERATURE SURVEY

2.1 WIRELESS HEAD GESTURE CONTROLLED WHEEL CHAIR FOR DISABLE PERSONS

Abstract

This paper describes an indigenously developed hands-free wheel chair for physically disabled persons. The proposed device works based on the Head Gesture Recognition technique using Acceleration sensor. Conventional electric powered wheel chairs are usually controlled by joysticks or hand gesture technology which cannot fulfil the needs of an almost completely disabled person who has restricted limb movements and can hardly move or turn his head only. Acceleration sensor is used for the head gesture recognition and RF (radio frequency) module is used for the smart wireless controlling. With the change of head gesture, data is sent wirelessly to the microcontroller based motor driving circuit to control the movement of the Wheel Chair in five different modes, namely FRONT, BACK, RIGHT, LEFT and a special locking system to STAND still at some place. The proposed device is fabricated using components collected from local market and tested in lab for successful functioning, test results are included in this paper.

PUBLISHED IN: 2017 IEEE .

AUTHOR: Shayban Nasif, Muhammad Abdul Goffar Khan**.**

2.2 STEERING CONTROL IN MULTI-DEGREES-OF-FREEDOM TWO-WHEELED WHEEL CHAIR ON SLOPE ENVIRONMENT

Abstract

As the development of electrical transportation systems, two-wheeled mobile robots have been widely studied because of the high mobility. Among them, the two - wheeled wheel chair is the wheel chair without caster wheels, and moves and stabilizes with only two wheels. However, wheel chairs cannot move lateral direction in general, and they sometimes fall down and cause into serious accidents. In this paper, lateral directional mode and its control method are proposed to solve this problem, and the validity of proposal is checked through simulations. The results are evaluated by the movement distance and the change of body's pitch angle. Index Terms—Two-wheeled wheel chair, Stabilized Control, Observers, Sensorless Control, Mobile Robots, Steering Control, Multi-degrees-of-freedom

AUTHOR: Sakurako HAMATANI, Takahiro NOZAKI .

2.3 EYE BLINK CONTROLLED LOW COST SMART WHEEL CHAIR AIDING DISABLED PEOPLE

Abstract

In this paper, a smart wheelchair is proposed to help out physically disabled people by using their eye blink activities. A prototype is implemented by processing eye ball images of the subject to detect intentional eye blinks. An algorithm is proposed to produce wheelchair movement command from consecutive intentional eye blinks. To identify intentional eye blinks from eye ball images, advanced image

processing techniques are applied using raspberry-pi. The commands are wirelessly transferred to the motor driver of the wheel chair.The HC-05 Buletooth receiver is connected to the wheelchair which receives transmitted data from raspberry-pi and motor driver implements the command according to predefined decision table. It can also sense obstacle using APDS 9960 proximity sensor to avoid collision with surroundings. The proposed smart wheelchair has shown False Command Rate (FCR) of 2.1% to 13.2%. The experiment is performed on four subjects and each of them has attended three sessions. Overall, the eye blink controlled aiding wheelchair is going to create a humanitarian impact on physically disabled people.

AUTHOR: Md Ashiqur Rahman Apu, Imran Fahad, S. A. Fattah, Celia Shahnaz.

2.4 WHEEL THERAPY CHAIR: A SMART SYSTEM FOR DISABLED PERSON WITH THERAPY FACILITY

Abstract

Each and every person in this world has a desire to live a normal human life but accidents, diseases, elder-ship make their desire into disability. Moreover, there are lots of handicaps and elders as well as the number of paralyzed people are increasing day by day. They always need another person in moving and have to go under some physical therapies under the guidance of a therapist to recuperate their strain back. In this paper the proposed system helps them to move freely & safely and also takes the activities of a therapist in a cost effective manner. This system is a combination of different controlling features, has the ability to detect obstacle and provides few kinds of therapies. A smart wheelchair is developed by using voice recognition system to control the movement of wheelchair and also with Arduino interfaced joystick. Besides, an ultrasound system provides the facility of automatic obstacle detection. The aim of research is to compact many facilities in a single wheelchair at low cost.

AUTHOR. . Md. Mamunur Rahman, Swarup Chakraborty

2.5 DEVELOPMENT OF A SMART WHEELCHAIR WITH DUAL FUNCTIONS: REAL-TIME CONTROL AND AUTOMATED GUIDE

Abstract

Wheelchairs are devices used for mobility by people whose walking is difficult or impossible, due to illness or disability. There are many types of wheelchairs, including basic, light-weight, folding, multi-function, special types, and so on. Each of them has its advantages and disadvantages. Development of brain-computer interface applications has become quite popular in recent year. It allows direct communication between brain and computer. One type of the communication is by imagination. By analyzing the EEG signals under imagination, different features could be calculated to transform into instructions for the control of back-end robotic device. Since brain-computer interface allows users to control the instruments without the need of physical input, it may be suitable for patients with physical defects or limb weakness. Therefore, this study aimed to design a smart wheelchair which has real-time control and automated guided functions. We based on the advantages of simple and rapid construction in LabVIEW to develop the smart wheelchair. This system was able to control our own-made wheelchair model with dual functions. In the real-time control function, our results show that the average accuracy of the imagination to the left was 70% and the average accuracy of the imagination to the right was 60%. In the automated guided function, our results

tested in small-area (20m · 20m in space) shows that wheelchair could move along the pre-setting path. In the future, we want to improve the whole system and make this prototype wheelchair more practical as using a real wheelchair instead of a wheelchair model. We hope to make this functional wheelchair more humane, so that the wheelchair is not only comfortable and safe but also has multiple functions to improve patient's quality of life and reduce the burden of care from their families.

Author: Chung-Kang Huang; Zuo-Wen Wang

2.6 IoT Based Smart Wheelchair for Disabled People

Abstract

Smart Wheelchair is known as a Power Wheelchair that is integrated into multiple sensors, assistive technology, and computers that give the user with a disability such as impairment, handicaps, and permanent injury, the required mobility to move freely and safely. These types of wheelchairs are gradually replacing the traditional wheelchairs; however, their expensive costs are preventing a large size of disabled people from having one. According to the organization of World Health (WHO), only 5 to 15% out of 70 million disabled people have access to wheelchairs. Therefore, we need to offer a cost-effective Smart that not only minimized the cost but also provides plenty of features that use the latest components and technologies. In the last years, there have been many pleasant efforts that serve this purpose. They have adopted various technologies such as artificial intelligence, where they have designed an autonomous wheelchair that used machine learning concepts to navigate, and some also used Internet of Thing technology to control the wheelchair-using voice recognition system. This report will present a cost-effective

Smart Wheelchair-based Arduino Nano microcontroller and IoT technology that have several features to gain disabled people, especially poor people who cannot afford expensive Smart Wheelchair, the required help to finish daily life tasks without external help. To conclude this project will make the Smart Wheelchair affordable to a wide range of disabled people and will be based on Arduino Nano, ESP-12e module to give Wi-Fi access, MPU6050 to detect fall with Voice message notification using IFTTT platform, obstacle detection with buzzer and LED to work as hazards, voice recognition system, and joysticks to control the wheelchair.

Author: Maryam Amur Khalfan Al Shabibi; Suresh Manic Kesavan.

2.7 Smart Wheelchair for Physically Challenged People

Abstract

According to the World Health Organization (WHO), in the world population, there around 70 million people who requires the need for wheelchairs, yet only 5-15% have access to it. The wheelchair we developed here is very cost effective and aids the people who are disabled and physically challenged. The wheelchair allows them to maneuver around their surrounding with ease without the intervention of another person, allowing them to be independent.

Author: Petson Varghese Baiju; Kevin Varghese; Jacob Mathew Alapatt; Sanjo Jame Joju

2.8 Designing of a Smart Wheelchair for People with Disabilities

Abstract

There are many people with disabilities in this day and age who find it is impossible to learn or perform day to day tasks. Some of these kinds of people rely on others for their assistance. Whichever way, they can become autonomous and conduct a few routine exercises with the assistance of helpful gadgets. The most used support devices are wheelchairs. The main purpose of wheelchairs is to provide help to people who are unable to walk because of sickness, handicaps or damage. However, there are also elderly persons with weak appendages and seals who cannot move the wheelchair. Smart wheelchairs will therefore gain a tone for them, and anyone else in the public eye. Intelligent wheelchairs are electrically operated wheelchairs with many other components, such as a PC and sensors, which help the consumer and clockmaker to handle wheelchairs safely and efficiently. Signal growth and autonomous mechanics contribute to the creation of new wheelchairs in the fields of artificial intelligence. This paper is designed to improve the maneuvering tasks of an intelligently self-contained wheelchair. In addition to computer-based treatment, no human intervention during navigation and perception is needed in the wheelchair.

Author: Vaibhavi Kengale; Kalyani Bansod; Chaitali Sure

CHAPTER 3
SYSTEM DESIGN

3.1 EXISITNG SYSTEM

In existing system the wheel chair is operated using Bluetooth module HC 05 which is interfaced with the controller. The main disadvantage is it can lose connection in certain conditions. It has low bandwidth as compared to other WSN networks. It allows only short range communication between devices. This is considered as the main disadvantage of the existing method.

3.2 PROPOSED SYSTEM

Assistive technology for physically challenged personalities had wide scope in present research era. The major concept of our project is to provide an assistive wheel chair working based on MEMS sensor which is connected to controller. The MEMS sensors may be positioned to sense the head movement or to sense the gestures of the hand. The motor driver which is connected with the controller is used to control the movement of the wheel chair based on the MEMS sensor. And also it is controlled by Google assistant.

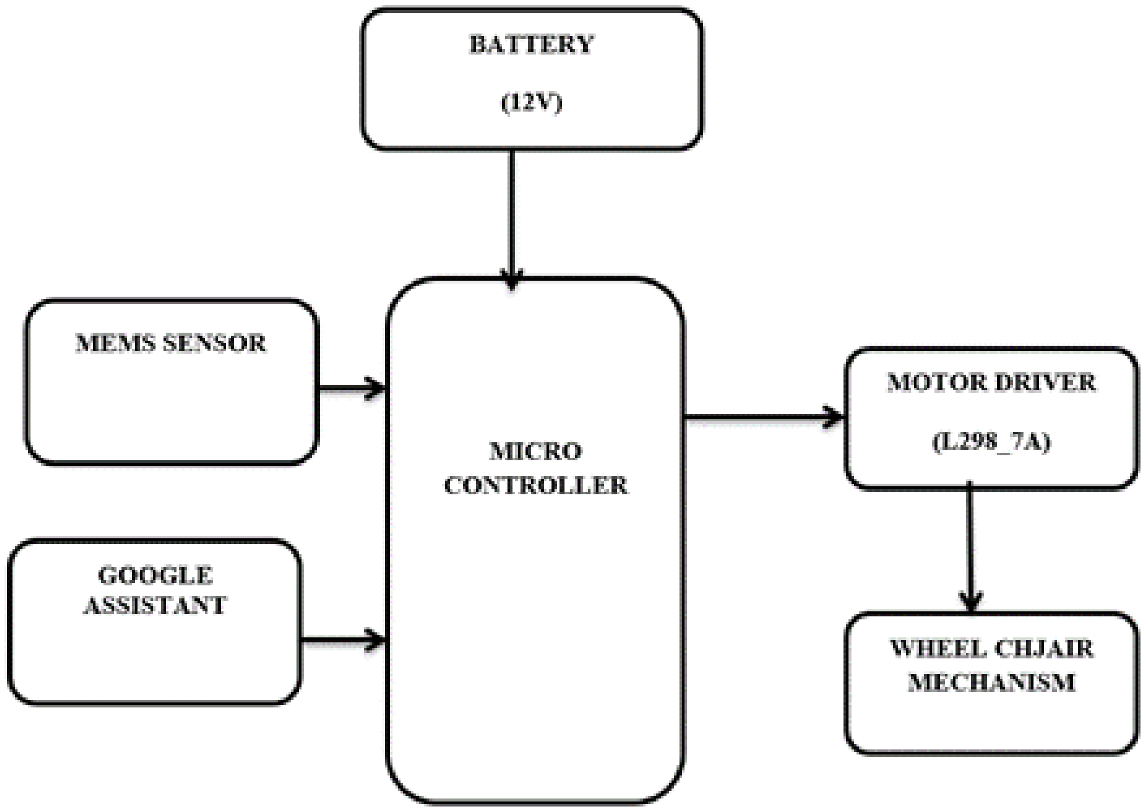


Fig 3.1 : Block diagram of proposed system

The wheelchair is controlled by head movement and voice control. The head movement is controlled my mems sensor and the voice control is controlled by google assistant. The input is given by MEMS sensor or google assistant to the micro controller. According to the input by sensors micro controller gives input to the driver and the driver conrols the wheelchair. The input for the system is given by 12V battery which can be rechargeable.

CHAPTER 4

HARDWARE REQUIREMENTS

4.1 ESP32 Microcontroller

Arduino is a great platform for beginners into the World of Microcontrollers and Embedded Systems. With a lot of cheap sensors and modules, you can make several projects either as a hobby or even commercial.

As technology advanced, new project ideas and implementations came into play and one particular concept is the Internet of Things or IoT. It is a connected platform, where several "things" or devices are connected over internet for exchangeof information.

In DIY community, the IOT projects are mainly focused on Home Automation and Smart Home applications but commercial and industrial IoT projects have far complex implementations like Machine Learning, Artificial Intelligence, Wireless Sensor Networks etc.

The important thing in this brief intro is whether it is a small DIY project by a hobbyist or a complex industrial project, any IoT project must have connectivity to Internet. This is where the likes of ESP8266 and ESP32 come into picture.

If you want to add Wi-Fi connectivity to your projects, then ESP8266 is a great option. But if you want build a complete system with Wi-Fi connectivity, Bluetooth connectivity, high resolution ADCs, DAC, Serial Connectivity and many other features, then ESP32 is the ultimate choice.

4.1.1 Architecture of ESP32

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica's 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.

Another important thing to know about ESP32 is that it is manufactured using TSMC's ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.

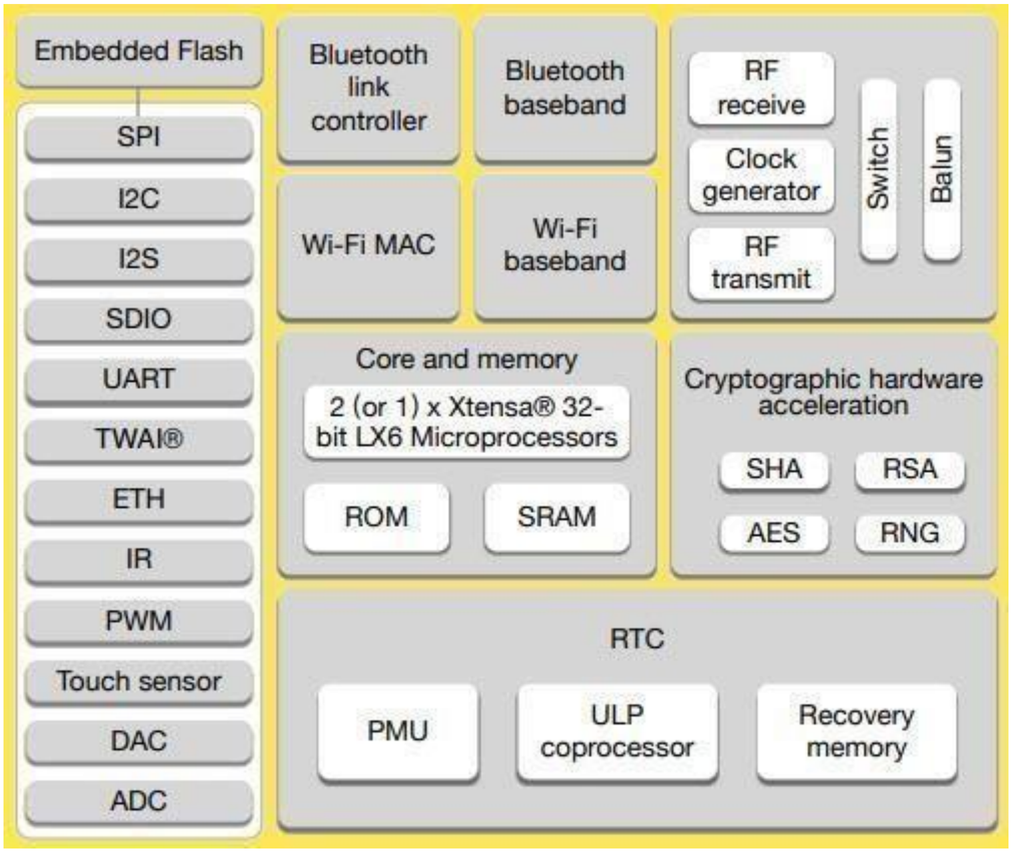


Fig 4.1 : Architecture of ESP32 microcontroller

4.1.2 General Discription

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm processor.

Fig 4.2 : ESP32 microcontroller

4.1.3 Specifications of ESP32

ESP32 has a lot more features than ESP8266 and it is difficult to include all the specifications in this Getting Started with ESP32 guide. So, I made a list of some of the important specifications of ESP32 here. But for complete set of specifications, I strongly suggest you to refer to the Datasheet.

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.

- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I^2C, 2 x I^2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16-channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.

4.2 24 V Wipers Motors

4.2.1 General Description

The car wiper motor is the component that powers the windshield wipers. As it spins, a mechanism built to it rotates a worm gear, arm and, finally, the windshield or windscreen wiper blades. The wiper blades then rid the windscreen of water, snow, dust, or any other debris that may affect visibility when driving.

Car wiper motors form part of the wiper system. Because they help clear the windshield, they are usually viewed as one of the car safety components. Other parts of the wiper system include wiper linkage, wiper washer pump, and wiper switch

Fig 4.4 : Wipers Motor

4.2.2 Features

- Supply voltage: 24 vdc
- Long Lifetime, Low Noise, Smooth Motion
- Equipped with high efficiency

4.3 DRIVER CIRCUIT

4.3.1 BTS7960 Motor Driver

The BTS7960 is a fully integrated high current H bridge module for motor drive applications. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against overtemperature, overvoltage, undervoltage, overcurrent and short circuit. The BTS7960 provides a cost optimized solution for protected high current PWM motor drives with very low board space consumption.

4.3.2 Specifications

- Input Voltage: 6 ~ 27Vdc.
- Driver: Dual BTS7960 H Bridge Configuration.
- Peak current: 43-Amp.
- PWM capability of up to 25 kHz.
- Control Input Level: 3.3~5V.
- Control Mode: PWM or level
- Working Duty Cycle: 0 ~100%.
- Over-voltage Lock Out.

- Under-voltage Shut Down.
- Board Size (LxWxH): 50mm x 50mm x 43mm.
- Weight: ~66g.

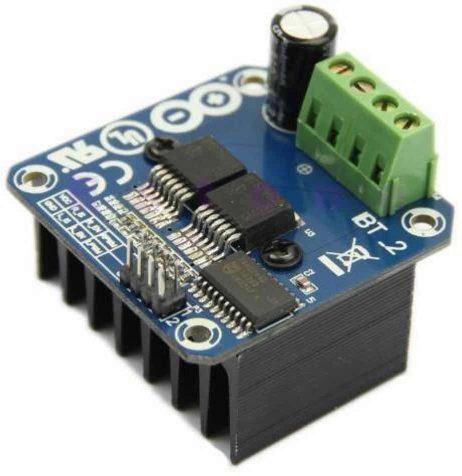
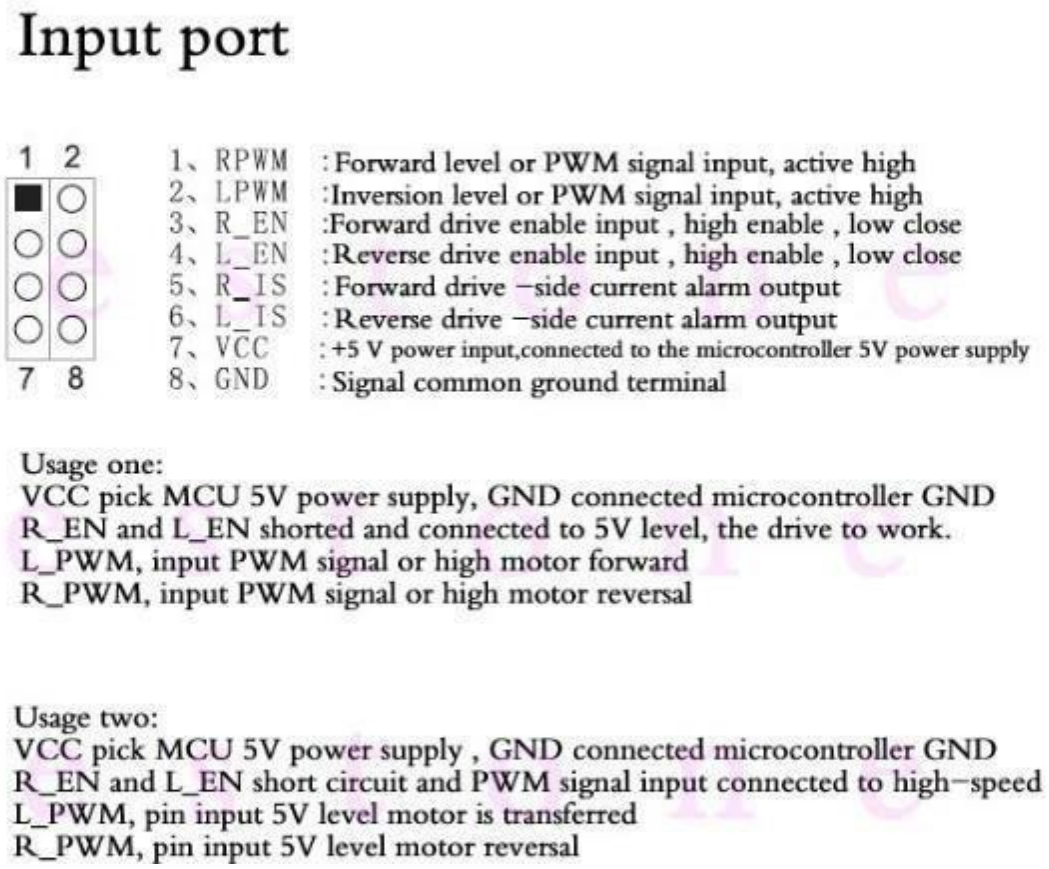


Fig 4.5 : BTS7960 Motor Driver



Input port

1、	RPWM	: Forward level or PWM signal input, active high
2、	LPWM	:Inversion level or PWM signal input, active high
3、	R_EN	:Forward drive enable input , high enable , low close
4、	L_EN	:Reverse drive enable input , high enable , low close
5、	R_IS	: Forward drive −side current alarm output
6、	L_IS	: Reverse drive −side current alarm output
7、	VCC	: +5 V power input,connected to the microcontroller 5V power supply
8、	GND	: Signal common ground terminal

Usage one:
VCC pick MCU 5V power supply, GND connected microcontroller GND
R_EN and L_EN shorted and connected to 5V level, the drive to work.
L_PWM, input PWM signal or high motor forward
R_PWM, input PWM signal or high motor reversal

Usage two:
VCC pick MCU 5V power supply , GND connected microcontroller GND
R_EN and L_EN short circuit and PWM signal input connected to high−speed
L_PWM, pin input 5V level motor is transferred
R_PWM, pin input 5V level motor reversal

Fig 4.6 : Input port of Motor Drive

4.4 MEMS Accelerometer

4.4.1 General Description

The accelerometer is a low power, low profile capacitive micro machined Accelerometer featuring signal conditioning, a 1-pole low pass filter, temperature Compensation, self test, 0g-Detect which detects linear freefall, and g-Select which Allows for the selection between 2 sensitivities Zero-g offset and sensitivity is Factory set and requires no external devices. This includes a Sleep Mode that makes it ideal for handheld battery powered electronics.

4.4.2 Product Description

You can use an accelerometer's ability to sense acceleration to measure a variety of things that are very useful to electronic and robotic projects and designs:

- Acceleration
- Tilt and tilt angle
- Incline
- Rotation
- Vibration
- Collision
- Gravity

Acceleration is a measure of how quickly speed changes. Just as a speedometer is a meter that measures speed, an accelerometer is a meter that measures acceleration. Accelerometers are useful for sensing vibrations in systems or for orientation applications. Accelerometers can measure acceleration on one, two, or three axis. 3-axis units are becoming more common as the cost of development for them decreases. You can use an accelerometer's ability to sense acceleration to measure a variety of things that are very useful to electronic and robotic projects.

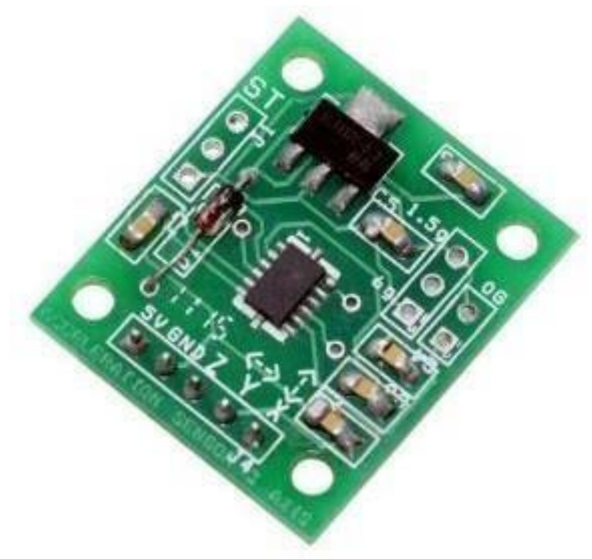


Fig 4.7 : MEMS Accelerometer

4.4.3 FEATURES

- Low Current Consumption: 400 μA
- Sleep Mode: 3μA
- Low Voltage Operation: 2.2 V – 3.6 V
- High Sensitivity (800 mV/g @ 1.5g)
- Selectable Sensitivity (±1.5g, ±6g)
- Fast Turn on Time (0.5 ms Enable Response Time)
- Self-Test for Freefall Detect Diagnosis

4.4.4 APPLICATIONS

- Self-balancing robots
- Tilt-mode game controllers
- Model airplane auto pilot
- Car alarm systems

CHAPTER 5

SOFTWARE REQUIREMENT

5.1 Google Assistant

Google Assistant is Google's voice assistant. When it launched, Google Assistant was an extension of Google Now, designed to be personal while expanding on Google's existing "OK Google" voice controls.

Originally, Google Now smartly pulled out relevant information for you. It knew where you worked, your meetings and travel plans, the sports teams you liked, and what interested you so that it could present you with information that mattered to you.

Google has long killed Google Now, but Assistant lives in the same space, fusing these personalized elements with a wide-range of voice control. Google Assistant supports both text or voice entry and it will follow the conversation whichever entry method you're using.

5.1.1 Google Assistant on Phone

Google expanded its Google Assistant service in 2017 so that it would be available on more mobile devices. That saw the roll-out of Assistant to most Android phones, with all recent launches offering the AI system. Even devices that offer another AI system, like Samsung's Bixby, also offer Google Assistant. Essentially, if your phone has Android, your phone has Google Assistant, so the user base for Google Assistant is huge.

It's possible to have Assistant respond to you even when your Android phone is locked too, if you opt-in through your settings and you can also opt in to see answers to personal queries too.

Google Assistant is also available on the iPhone, although there are some restrictions. So, Google Assistant is no longer the preserve of Pixel phones; it's something that all Android users and even iOS users can enjoy.

5.1.2 Features

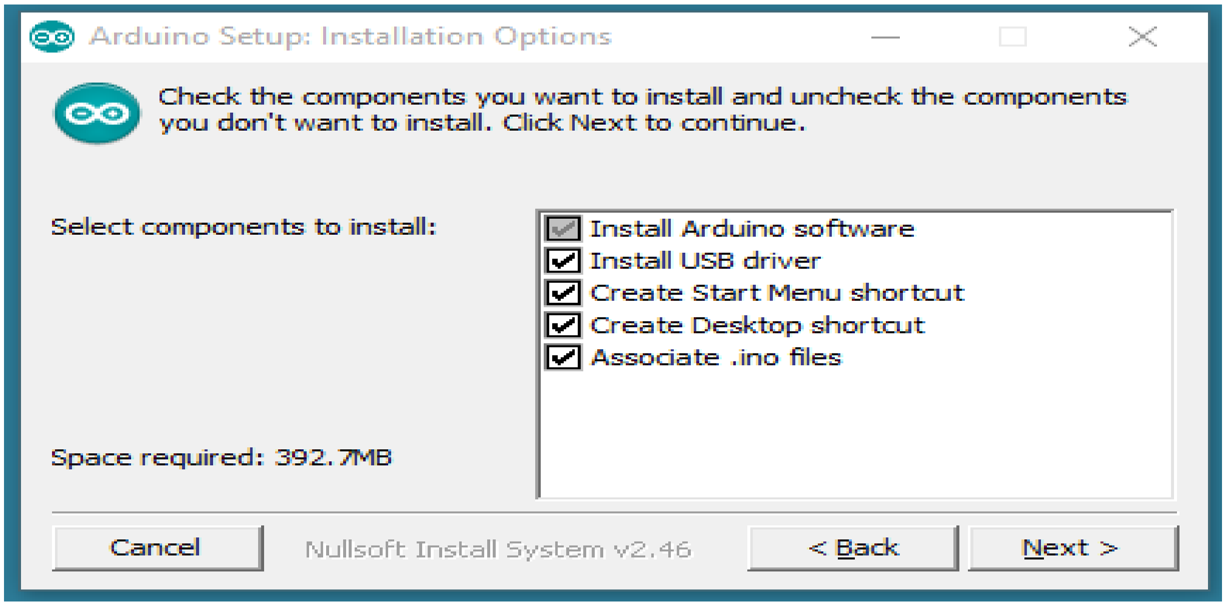
- Ease to Check-in Flights
- Opportunity to Book a Hotel
- See Traffic on Locked Device
- Make Phone Calls for Appointments
- Do Image Search
- Find Nearby Places
- Get Weather Forecast
- To Practice a New Language
- Convert Currencies and Measurements

5.2 Software Description

5.2.1 Arduino Software (IDE)

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.


Arduino Setup: Installation Options

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

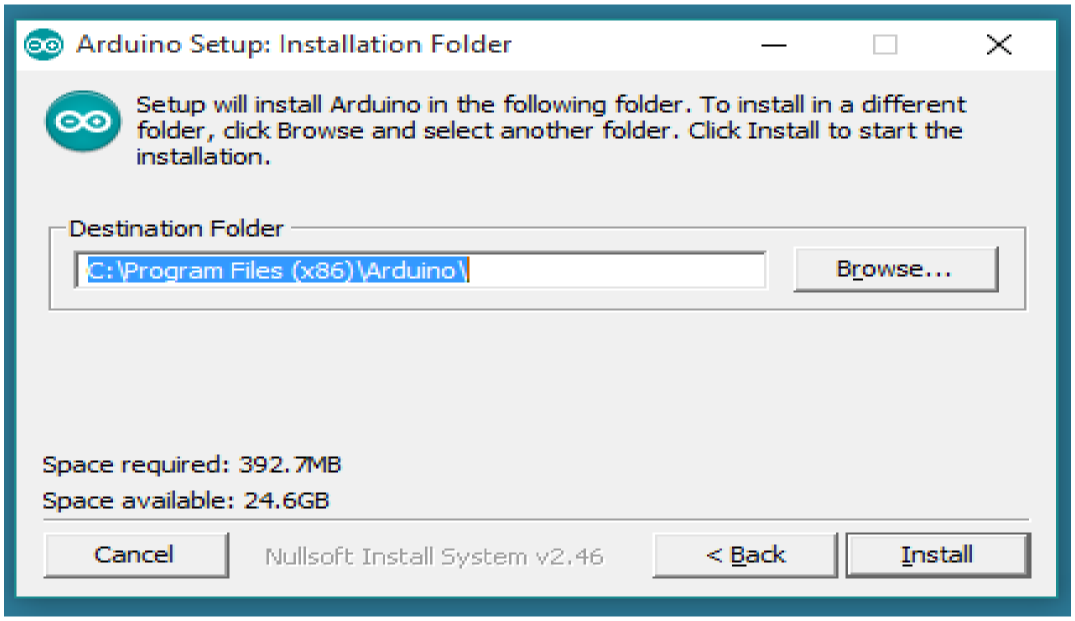
Select components to install:

- [✓] Install Arduino software
- [✓] Install USB driver
- [✓] Create Start Menu shortcut
- [✓] Create Desktop shortcut
- [✓] Associate .ino files

Space required: 392.7MB

Fig5.1 : Arduino setup: Installation Options

Choose the components to install


Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino\

Space required: 392.7MB

Space available: 24.6GB

Fig5.2 : Arduino Setup: Installation Folder

Choose the installation directory (we suggest to keep the default one)

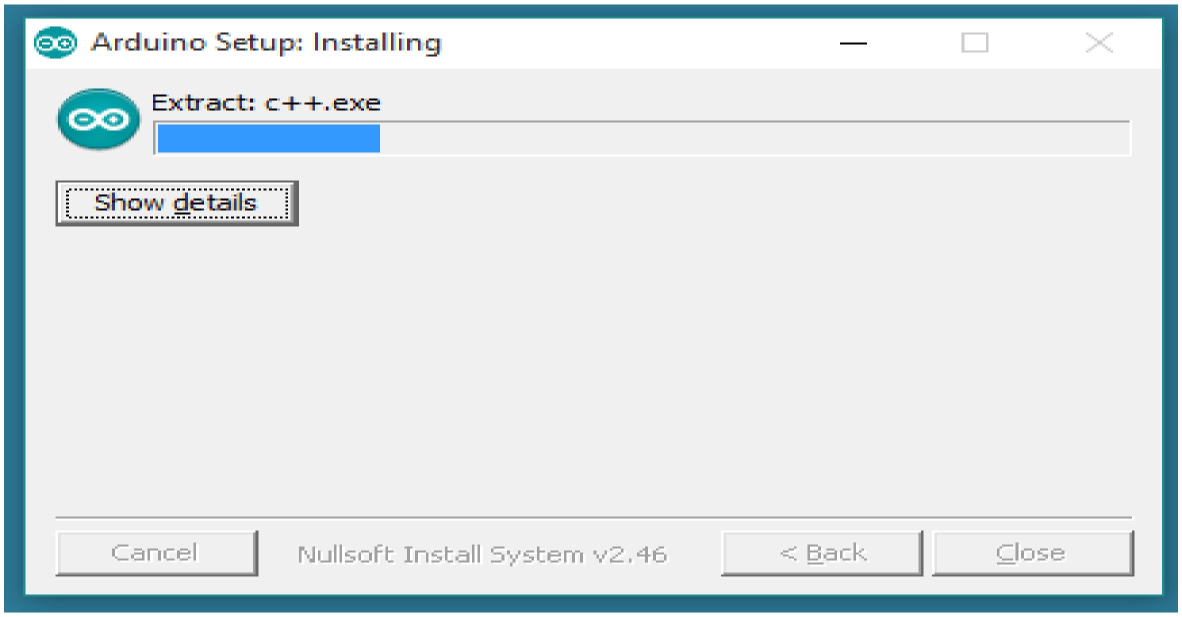


Fig5.3 : Arduino Setup: Installing

The process will extract and install all the required files to execute properly the Arduino Software (IDE)

5.2.2 Arduino Boot loader Issue

The current boot loader burned onto the Arduino UNO is not compatible with ROBOTC. In its current form, you will be able to download the ROBOTC Firmware to the ArduinoUNO, but you will not able to download any user programs.

The reason for this is because there is a bug in the Arduino UNO firmware that does not allow flash write commands to start at anywhere but the beginning of flash memory (0x000000). See the bottom of this page for more technical details.

Because ROBOTC is not able to burn a new bootloader as of today, you will need to use the Arduino's Open Source language with a modified bootloader file to re-burn your bootloader on your Arduino UNO boards. The enhanced bootloader is backwards compatible with the original one. That means you'll still be able to program it through the Arduino programming environment as before, in addition to ROBOTC for Arduino.

5.2.3 Hardware Needed

To burn a new version of the Arduino boot loader to your UNO, you'll need an AVR ISP Compatible downloader.

5.2.3.1 Using an AVR ISP (In System Programmer)

- Your Arduino UNO (to program)
- An AVR Programmer such as the AVR Pocket Programmer
- An AVR Programming Cable (the pocket programmer comes with one)

If you have extra Arduino boards, but no ISP programmer, SparkFun.com has a cool tutorial on how to flash a bootloader using an Arduino as an ISP.

5.2.3.2 Using another Arduino as an ISP

- Your Arduino UNO (to program)
- A Working Arduino (doesn't matter what kind)
- Some Male-to-Male Jumper Cables

For instructions on this method, take a look at the SparkFun.com website: http://www.sparkfun.com/tutorials/247

5.2.4 Software Needed

ROBOTC is not currently able to burn a bootloader onto an Arduino board, so you'll need to download a copy of the latest version of the Arduino Open-Source programming language.

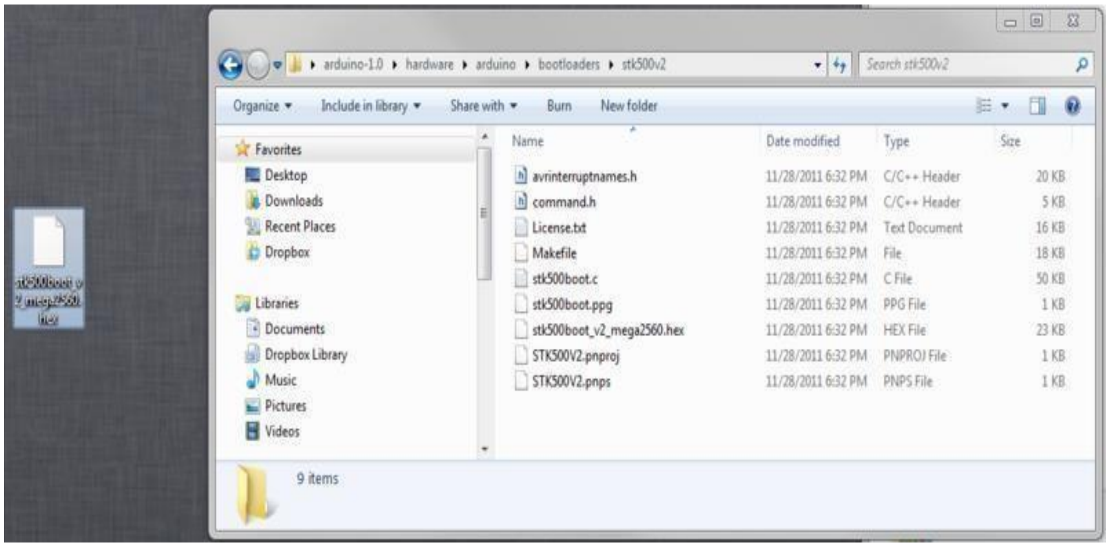
- Arduino Official Programming Language - Download Page

In addition, you'll need the ROBOTC modified bootloader. You can download that here:

- ROBOTC Modified UNO Bootloader - Modified Bootloader

5.2.4.1 Bootload Download Instructions

- Download the Arduino Open Source Software and a copy of the Modified Bootloader File
- Copy the Modified Bootloader File into the /Arduino-1.0/hardware/arduino/bootloaders/stk500v2/ and overwrite the existing bootloader.



Name	Date modified	Type	Size
avrinterruptnames.h	11/28/2011 6:32 PM	C/C++ Header	20 KB
command.h	11/28/2011 6:32 PM	C/C++ Header	5 KB
License.txt	11/28/2011 6:32 PM	Text Document	16 KB
Makefile	11/28/2011 6:32 PM	File	18 KB
stk500boot.c	11/28/2011 6:32 PM	C File	50 KB
stk500boot.ppg	11/28/2011 6:32 PM	PPG File	1 KB
stk500boot_v2_mega2560.hex	11/28/2011 6:32 PM	HEX File	23 KB
STK500V2.pnproj	11/28/2011 6:32 PM	PNPROJ File	1 KB
STK500V2.pnps	11/28/2011 6:32 PM	PNPS File	1 KB

Fig5.4 : Bootland Download

- Power up your Arduino UNO (either via USB or external power)
- Plug in your AVR ISP Programmer to your computer (make sure you have any required drivers installed)
- Connect your AVR ISP Programmer into your Arduino UNO Board via the ISP Header (the 2x3 header pins right above the Arduino Logo)
- Launch the Arduino Open Source Software

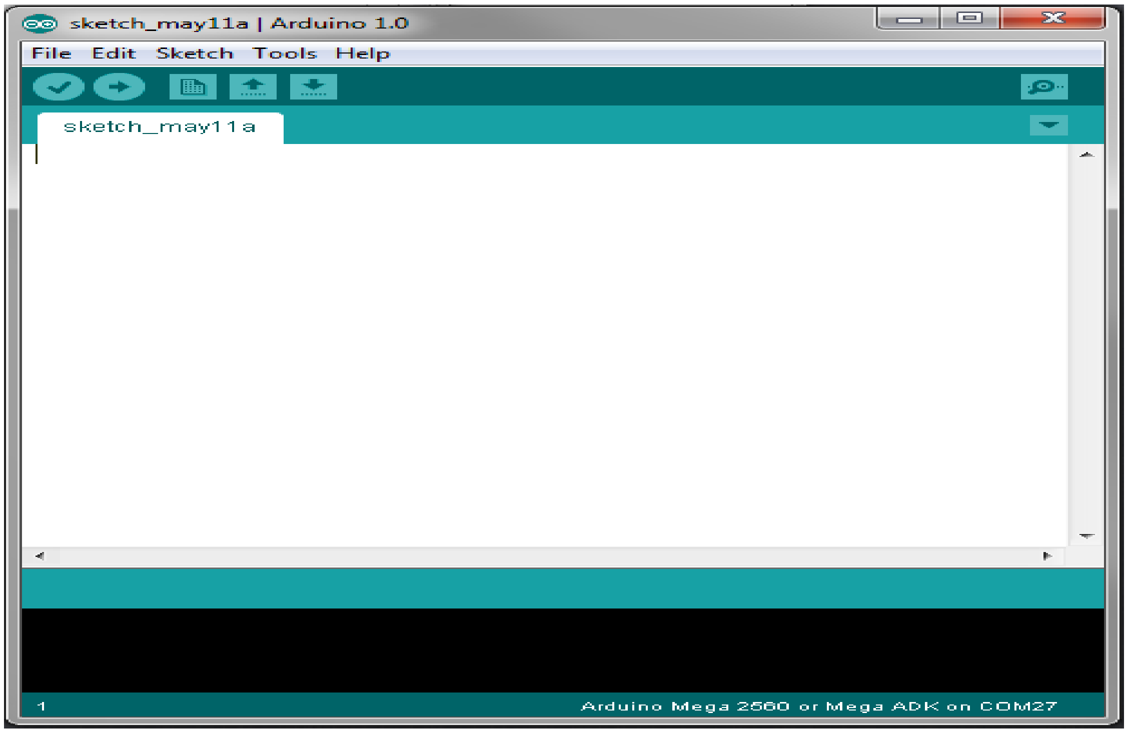
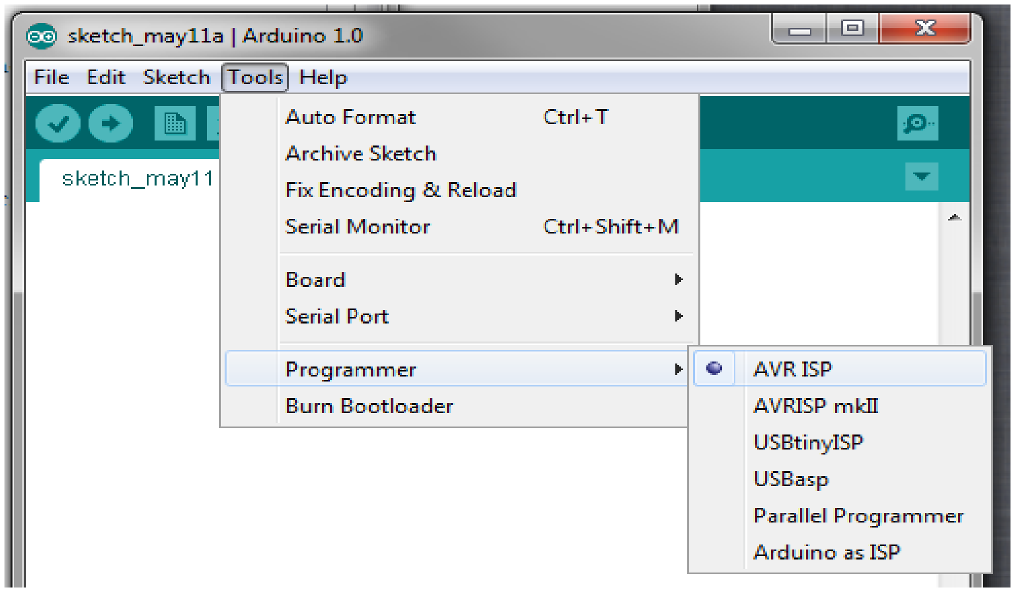
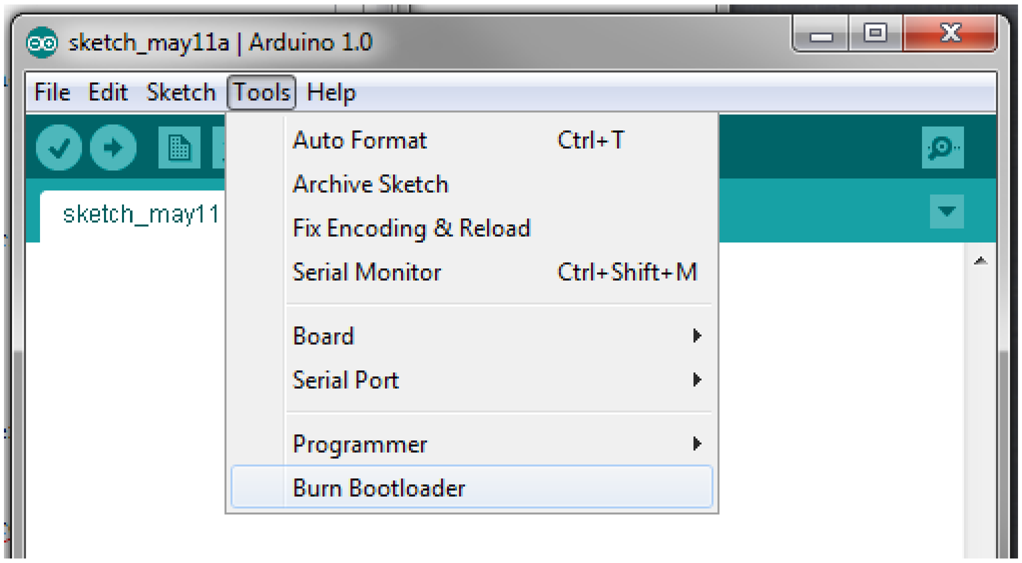

sketch_may11a | Arduino 1.0
File Edit Sketch Tools Help
sketch_may11a
1
Arduino Mega 2560 or Mega ADK on COM27

Fig 5.5 : Bootland setup

- Change your settings in the Arduino Software to look for an Arduino UNO
- Change your settings in the Arduino Software to select your ISP Programmer Type (Check your programmer's documentation for the exact model)



- Select the "Burn Bootloader" option under the "Tools" menu. The modified bootloader will now be sent to your Arduino. This typically take a minute or so.



- You should be all set to download ROBOTC firmware and start using your Arduino UNO with ROBOTC.

5.2.4.2 Technical Details

The Arduino Boot loader sets the "erase Address" to zero every time the boot loader is called. ROBOTC called the "Load Address" command to set the address in which we want to write/verify when downloading program.

When writing a page of memory to the arduino, the Arduino boot loader will erase the existing page and write a whole new page.

In the scenario of downloading firmware, everything is great because the Erase Address and the Loaded Address both start at zero.

In the scenario of writing a user program, we start writing at memory location 0x7000, but the Boot loader erases information starting at location zero because the "Load Address" command doesn't update where to erase.

Our modification is to set both the Load Address and the Erase Address so the activity of writing a user program doesn't cause the firmware to be accidentally erased.

Summary

Microcontroller	Arduino UNO
Operating Voltage	5V Input Voltage (recommended)
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40mA
DC Current for3.3VPin	50mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8KB
EEPROM	4KB
Lock Speed	16MHz

Table5.1: Summary of Technical Details

The Arduino UNO can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and V_{in} pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

They differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via a non-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.**A3.3voltsupplygeneratedbytheon-boardregulator.Maximumcurrentdrawis50mA.
- **GND.** Ground pins.

The ATMEGA has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50k Ohms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATMEGA USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a changing value. See the attach Interrupt() function for details.
- **PWM: 0to13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^{2}C: 20 (SDA) and 21 (SCL).** Support I^{2}C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^{2}C pins on the Duemilanove.

The Arduino UNO has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analog Reference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analog Reference().
- **ESET.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

5.2.4.3 Communication

The Arduino UNO has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The Arduino UNO provides four hardware UARTs for TTL (5V) serial communication.

An ATMEGA on the board channels one of these over USB and provides a virtual comport to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and1).

A Software Serial library allows for serial communication on any of the digital pins.

The Arduino UNO also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the Arduino UNO datasheet.

5.2.4.4 Programming

The Arduino UNO can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Arduino UNO on the Arduino UNO comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

5.2.4.5 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino UNO is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the Arduino UNO via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Arduino UNO is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the UNO. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytesof data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

5.3 USB Over current Protection

5.3.1 Physical Characteristics and Shield Compatibility

The Arduino UNO has a resettable poly fuse that protects your computer's USB ports from shorts and over current. Although most computers provide their own internal protection, the fuse provides an extra layer ofprotection.Ifmorethan500mAisappliedtotheUSBport, the fuse will automatically break the connection until the short or overload is removed.

The maximum length and width of the UNO PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The UNO is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila.

Please note that I^2C is not located on the same pins on the Mega (20and21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

5.3.2 Instructions to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You’ll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Once you have downloaded/unzipped the arduino IDE, you can plug the Arduino to your PC via USB cable.

5.4 EMBEDDED C

5.4.1 ABOUT EMBEDDED C

High-level language programming has long been in use for embedded-systems development. However, assembly programming still prevails, particularly for digital-signal processor (DSP) based systems. DSPs are often programmed in assembly language by programmers who know the processor architecture inside out. The key motivation for this practice is performance, despite the disadvantages of assembly programming when compared to high-level language programming.

If the video decoding takes 80 percent of the CPU-cycle budget instead of 90 percent, for instance, there are twice as many cycles available for audio processing. This coupling of performance to end-user features is characteristic of many of the real-time applications in which DSP processors are applied. DSPs have a highly specialized architecture to achieve the performance requirements for signal processing applications within the limits of cost and power consumption set for consumer applications. Unlike a conventional Load-Store (RISC) architecture, DSPs have a data path with memory-access units that directly feed into the arithmetic units. Address registers are taken out of the general-purpose register file and placed next to the memory units in a separate register file.

A further specialization of the data path is the coupling of multiplication and addition to form a single cycle Multiply-accumulate unit (MAC). It is combined with special-purpose accumulator registers, which are separate from the general-purpose registers. Data memory is segmented and placed close to the MAC to achieve the high bandwidths required to keep up with the streamlined data path. Limits are often placed on the extent of memory-addressing operations. The localization of resources in the data path saves many data movements that typically take place in a Load-Store architecture.

The most important, common arithmetic extension to DSP architectures is the handling of saturated fixed-point operations by the arithmetic unit. Fixed-point arithmetic can be implemented with little additional cost over integer arithmetic. Automatic saturation (or clipping) significantly reduces the number of control-flow instructions needed for checking overflow explicitly in the program.Changes in technological and economic requirements make it more expensive to continue programming DSPs in assembly. Staying with the mobile phone as an example, the signal-processing algorithms required become increasingly complex. Features such as stronger error correction and encryption must be added. Communication protocols become more sophisticated and require much more code to implement. In certain markets, multiple protocol stacks are implemented to be compatible with multiple service providers. In addition, backward compatibility with older protocols is needed to stay synchronized with provider networks that are in a slow process of upgrading.

Today, most embedded processors are offered with C compilers. Despite this, programming DSPs is still done in assembly for the signal processing parts or, at best, by using assembly-written libraries supplied by manufacturers. The key reason for this is that although the architecture is well matched to the requirements of the signal-processing application, there is no way to express the algorithms efficiently and in a natural way in Standard C. Saturated arithmetic.

For example, is required in many algorithms and is supplied as a primitive in many DSPs. However, there is no such primitive in Standard C. To express saturated arithmetic in C requires comparisons, conditional statements, and correcting assignments. Instead of using a primitive, the operation is spread over a number of statements that are difficult to recognize as a single primitive by a compiler.

5.4.2 DESCRIPTION

Embedded C is designed to bridge the performance mismatch between Standard C and the embedded hardware and application architecture. It extends the C language with the primitives that are needed by signal-processing applications and that are commonly provided by DSP processors. The design of the support for fixed-point data types and named address spaces in Embedded C is based on DSP-C. DSP-C [1] is an industry-designed extension of C with which experience was gained since 1998 by various DSP manufacturers in their compilers. For the development of DSP-C by ACE (the company three of us work for), cooperation was sought with embedded-application designers and DSP manufacturers.

The Embedded C specification extends the C language to support freestanding embedded processors in exploiting the multiple address space functionality, user-defined named address spaces, and direct access to processor and I/O registers. These features are common for the small, embedded processors used in most consumer products. The features introduced by Embedded C are fixed-point and saturated arithmetic, segmented memory spaces, and hardware I/O addressing. The description we present here addresses the extensions from a language-design perspective, as opposed to the programmer or processor architecture perspective.

5.4.3 MULTIPLE ADDRESS SPACES

Embedded C supports the multiple address spaces found in most embedded systems. It provides a formal mechanism for C applications to directly access (or map onto) those individual processor instructions that are designed for optimal memory access. Named address spaces use a single, simple approach to grouping memory locations into functional groups to support MAC buffers in DSP applications, physical separate memory spaces, direct access to processor registers, and user-defined address spaces.

The Embedded C extension supports defining both the natural multiple address space built into a processor's architecture and the application-specific address space that can help define the solution to a problem.

Embedded C uses address space qualifiers to identify specific memory spaces in variable declarations. There are no predefined keywords for this, as the actual memory segmentation is left to the implementation. As an example, assume that **X** and **Y** are memory qualifiers. The definition:

```
X int a[25] ;
```

Means that **a** is an array of 25 integers, which is located in the **X** memory. Similarly (but less common):

```
X int * Y p ;
```

Means that the pointer **p** is stored in the **Y** memory. This pointer points to integer data that is located in the **X** memory. If no memory qualifiers are used, the data is stored into unqualified memory.

For proper integration with the C language, a memory structure is specified, where the unqualified memory encompasses all other memories. All unqualified pointers are pointers into this unqualified memory. The unqualified memory abstraction is needed to keep the compatibility of the **void *** type, the **NULL** pointer, and to avoid duplication of all library code that accesses memory through pointers that are passed as parameters.

5.4.4 NAMED REGISTERS

Embedded C allows direct access to processor registers that are not addressable in any of the machine's address spaces. The processor registers are defined by the compiler-specific, named-register, storage class for each supported processor. The processor registers are declared and used like conventional C variables (in many cases volatile variables). Developers using Embedded C can now develop their applications, including direct access to the condition code register and other processor-specific status flags, in a high-level language, instead of inline assembly code.

Named address spaces and full processor access reduces application dependency on assembly code and shifts the responsibility for computing data types, array and structure offsets, and all those things that C compilers routinely and easily do from developers to compilers.

5.4.5 I/O HARDWARE ADDRESSING

The motivation to include primitives for I/O hardware addressing in Embedded C is to improve the portability of device-driver code. In principle, a hardware device driver should only be concerned with the device itself. The driver operates on the device through device registers, which are device specific. However, the method to access these registers can be very different on different systems, even though it is the same device that is connected. The I/O hardware access primitives aim to create a layer that abstracts the system-specific access method from the device that is accessed. The ultimate goal is to allow source-code portability of device drivers between different systems. In the design of the I/O hardware-addressing interface, three requirements needed to be fulfilled:

1. The device-drive source code must be portable.
2. The interface must not prevent implementations from producing machine code that is as efficient as other methods.
3. The design should permit encapsulation of the system-dependent access method.

The design is based on a small collection of functions that are specified in the <iohw.h> include file. These interfaces are divided into two groups; one group provides access to the device, and the second group maintains the access method abstraction itself.

To access the device, the following functions are defined by Embedded C:

```
unsigned int iord(  ioreg_designator );
void iowr( ioreg_designator, unsigned int value );
void ioor( ioreg_designator, unsigned int value );
void ioand( ioreg_designator, unsigned int value );
void ioxor( ioreg_designator, unsigned int value );
```

These interfaces provide read/write access to device registers, as well as typical methods for setting/resetting individual bits. Variants of these functions are defined (with **buf** appended to the names) to access arrays of registers. Variants are also defined (with l appended) to operate with **long** values.

All of these interfaces take an I/O register designator **ioreg_designator** as one of the arguments. These register designators are an abstraction of the real registers provided by the system implementation and hide the access method from the driver source code. Three functions are defined for managing the I/O register designators. Although these are abstract entities for the device driver, the driver does have the obligation to initialize and release the access methods. These functions do not access or initialize the device itself because that is the task of the driver. They allow, for example, the operating system to provide a memory mapping of the device in the user address space.

```
void iogroup_acquire( iogrp_designator );
void iogroup_release( iogrp_designator );
void iogroup_map( iogrp_designator, iogrp_designator );
```

The **iogrp_designator** specifies a logical group of I/O register designators; typically this will be all the registers of one device. Like the I/O register designator, the I/O group designator is an identifier or macro that is provided by the system implementation.The map variant allows cloning of an access method when one device driver is to be used to access multiple identical devices.

5.4.6 EMBEDDED C PORTABILITY

By design, a number of properties in Embedded C are left implementation defined. This implies that the portability of Embedded C programs is not always guaranteed. Embedded C provides access to the performance features of DSPs. As not all processors are equal, not all Embedded C implementations can be equal For example, suppose an application requires 24-bit fixed-point arithmetic and an

Embedded C implementation provides only 16 bits because that is the native size of the processor. When the algorithm is expressed in Embedded C, it will not produce outputs of the right precision.

In such a case, there is a mismatch between the requirements of the application and the capabilities of the processor. Under no circumstances, including the use of assembly, will the algorithm run efficiently on such a processor. Embedded C cannot overcome such discrepancies. Yet, Embedded C provides a great improvement in the portability and software engineering of embedded applications. Despite many differences between performance-specific processors, there is a remarkable similarity in the special-purpose features that they provide to speed up applications.

Writing C code with the low-level processor-specific support may at first appear to have many of the portability problems usually associated with assembly code. In the limited experience with porting applications that use Embedded C extensions, an automotive engine controller application (about 8000 lines of source) was ported from the eTPU, a 24-bit special-purpose processor, to a general-purpose 8-bit Freescale 68S08 with about a screen full of definitions put into a single header file. The porting process was much easier than expected. For example, variables that had been implemented on the processor registers were ported to unqualified memory in the general-purpose microprocessor by changing the definitions in the header definition and without any actual code modifications. The exercise was to identify the porting issues and it is clear that the performance of the special-purposeprocessor is significantly higher than the general-purpose target.

SYSTEM IMAGE

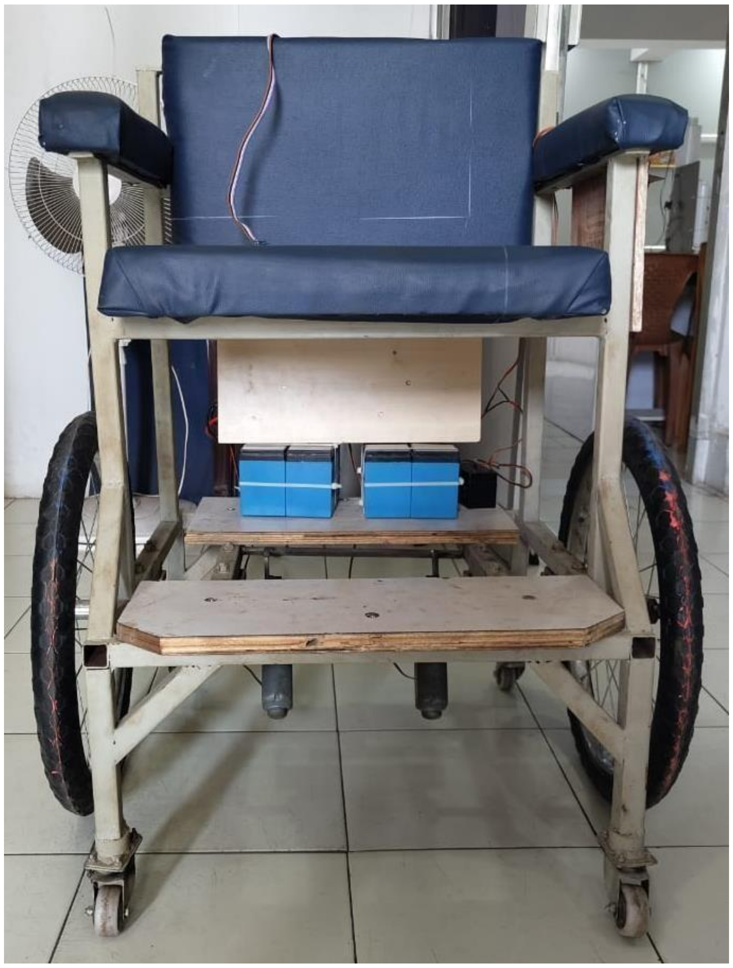


Fig5.5: Front view of system

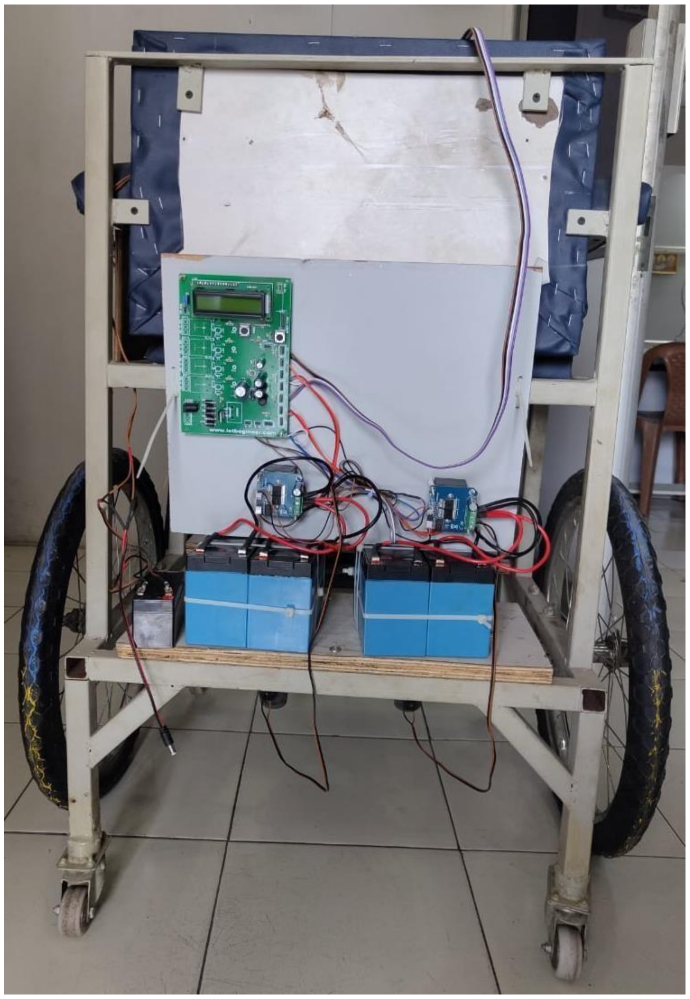
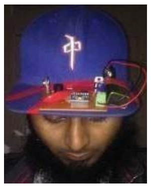
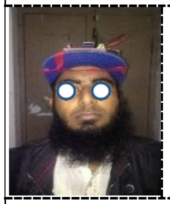
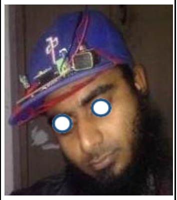
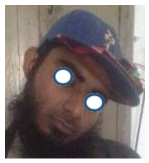


Fig5.6: Back view of system

WORKING

The working of the wheelchair is based on the input given by either MEMS sensor which is used to detect the head movement and google assistant connected by Bluetooth to the system. The input is given to the micro controller. The microcontroller gives the motor driver the command according to the input and the direction of the wheelchair is to be moved . The motor moves by the command by the motor driver. The input for the entire system is given by the 12V battery. The motor used here is 12V viper motor. The wheelchair can also be moved manually.

Head Movement Directions	**Corresponding Analog Value**	**Wheelchair Directions**
	Accelerometer axis Value Y>100	**Forward Direction**
	Accelerometer axis Value Y<-150	**Backward Direction**
	Accelerometer axis Value X>100	**Right Direction**
	Accelerometer axis Value X<-150	**Left Direction**

	Accelerometer axis Value Y>150	**Lock**

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

A major challenge in the design of the smart wheelchair is to rapidly, accurately, and sufficiently produce control commands. The cost of the system is less and it gives the reliable output as compared to another system which useful forsociety. To have safe and it is mainly implemented on a long scale for the better results and problem free solutions in the future.

Due this fast growing world due accident and other reason some may lost the movement and need to depend on others for their movement. Other than accidents due to improper cell division and malnutrition due to the food habits of now a days children may lack limbs. For their movement wheelchair is needed. This voice controlled and head movement controlled wheelchair will be affordable for those disabled peoples. With further development to this system advancement can be done more features can be added.

REFERENCES

[1] T. F. Bastos-filho, F. A. Cheein, S. Mara, T. Muller, W. C. Celeste, C. D. ¨ Cruz, D. C. Cavalieri, M. Sarcinelli-filho, P. Faria, S. Amaral, E. Perez, C. M. Soria, R. Carelli, and S. Member(2014) “Towards a New ModalityIndependent Interface for a Robotic Wheelchair,” vol. 22, no. 3, pp. 567–584.

[2] Q. Huang, S. He, Q. Wang, Z. Gu, N. Peng, K. Li, and Y. Zhang(2017) “An EOG- Based Human-Machine Interface for Wheelchair Control,” vol. 9294, no. c, pp. 1– 11.

[3] J. Ma, Y. Zhang, A. Cichocki, and F. Matsun(2014) “A Novel EOG / EEG Hybrid Human-Machine Interface Adopting Eye Movements and ERPs : Application to Robot Control,” vol. 9294, no. c, pp. 1–14.

[4] A. Maksud, R. I. Chowdhury, T. T. Chowdhury, S. A. Fattah, C. Shahanaz, and S. S. Chowdhury, Dec(2017) “Low-cost EEG based electric wheelchair with advanced controlfeatures,” vol. 2017-December, pp. 2648–2653.

[5] S. K. Sivanath, S. A. Muralikrishnan, and P. Thothadri(2012) “Eyeball and Blink Controlled Firing System for Military Tank Using LabVIEW”.

[6] Y. Wang, G. Zhai, S. Zhou, S. Chen, X. Min, and Z. Gao, “Eye Fatigue Assessment Using Unobtrusive Eye Tracker(2013)” IEEE Access, vol. PP, no. c, p. 1, J. Xiong, W. Xu, W. Liao, Q. Wang, J. Liu, and Q. Liang, “Eye Control SystemBase on Ameliorated Hough Transform Algorithm,” vol. 13, no. 9, pp. 3421–3429.

[7] W. Zhang, B. Cheng, and Y. Lin,(2012) “Driver Drowsiness Recognition Based on Computer Vision Technology *,” vol. 17, no. 3, pp. 354–362.

[8] R. Zhang, S. He, X. Yang, X. Wang, K. Li, Q. Huang, Z. Gu, and Z. Yu,(2018) “An EOG-based Human Machine Interface to Control a Smart Home Environment for Patients with Severe Spinal Cord Injuries,” vol. 9294, no. c, pp. 1–12.