Digitalization and Development of a Torque Sensor Based Control System and implementation of the algorithm using a Micro-Controller

A Thesis Submitted to the Department of Electrical and Electronic Engineering of BRAC University

By

Omar Shahjalal Shantonu	**ID: 13321034**
Afia Abeeda Shafiq	**ID: 13121074**
Raziul Hassan	**ID: 12221049**
Md. Safayet Ahmed	**ID: 12221088**

Supervised By

Dr. A.K.M Abdul Malek Azad

Professor

Department of Electrical and Electronic Engineering

BRAC University, Dhaka

In partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering

FALL 2016

BRAC University, Dhaka

DECLARATION

We hereby declare that our research work titled "Digitalization and Development of a Torque Sensor Based Control System and implementation of the algorithm using a Micro-Controller", a thesis submitted to the Department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the Bachelors of Science in Electrical and Electronics Engineering, is our own work. The work has not been presented elsewhere for assessment. Where a material has been collected from another source, it has been properly acknowledged and referred to.

Signature of Supervisor

……………………………………
Dr. AKM Abdul Malek Azad

Signature of Authors

....
Omar Shahjalal Shantonu

....
Afia Abeeda Shafiq

....
Raziul Hassan

....
Md. Safayet Ahmed

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. A.K.M. Abdul Malek Azad, Professor, Dept. of Electrical and Electronic Engineering (EEE), BRAC University, for his supervision and counselling intended for the completion of our thesis. Additionally, we would also like to thank BRAC University for providing us with the necessary equipment for completion of this project as well as for funding this project undertaken by Control and Application Research Centre (CARC).

ABSTRACT

Mechanical vehicles are a part of our daily lives. Some of them are driven by electrical power whilst others are driven manually by effort of the driver through pedaling the vehicle. Given that the majority of the population of the world is still under poverty, the dependency of people have on manually driven vehicles is more, as they are both cheaper to own and maintain. However, continuous utilization of man-driven vehicles like, rickshaws, vans and wheelchairs, can have strenuous effect on the rider. Our aim for the project is to reduce this human effort by assisting the rider with an electrical input when the load on the vehicle becomes high. Thus, we developed an intelligent control system, with the help of a micro-controller, which would work in conjunction with a Torque sensor to run a motor during some pre-defined conditions. This makes the operation of manually-driven vehicles easier by assisting the rider, when necessary, with power from a motor. This enables us to change the manually driven vehicle into a hybrid vehicle. The inclusion of the micro-controller is to make the final circuitry simpler and easier to operate. In this circuit design the user can simply replace the chip, instead of changing the entire circuitry as in the former analog circuit, when any wear and tear takes place. The on circuit micro-controller has a pre-installed algorithm which controls the overall decision making of the system. It automatically switches the motor both on and off based on the pre-installed threshold and cutoff voltage values. Our objective is to reduce the riders from the excessive physical exhaustion that mainly occurs when a high load is present on the vehicle or while initiating the momentum from either rest or low speed to generate a moderate speed. A motor helping the pullers/users only during these phases eradicates exhaustion to a significant level, and at the same time saves energy by limiting the over-use of the motor. Therefore, the digital circuit enables us to significantly reduce the human effort needed whilst keeping the identity and driving mechanism of the existing manually driven vehicles.

Our developed algorithm and the designed digital circuitry with the microcontroller are usable with all the existing manually driven vehicles that involve pedaling. In our project, we have implemented it in an electrically assisted wheelchair for physically disabled people.

TABLE OF CONTENTS

ACKNOWLEDGEMENT 3
ABSTRACT 4
LIST OF FIGURES 8
LIST OF TABLES 9
LIST OF ABBREVIATIONS 10
CHAPTER 1 Introduction 11
1.1 Introduction to Wheelchair 11
1.2 Main concern of physically challenged people in Bangladesh 13
1.3 Motivation 15
1.4 Outline of the subsequent chapters 16
CHAPTER 2 Overview of the existing system 17
2.1 Introduction 17
2.2 Hub Motor 19
2.2.1 Comparison with Conventional EV Design in Automobiles 20
2.3 Batteries 21
2.4 The Torque Sensor 21
2.5 Motor Controller 22
2.6 Throttle 23
2.7 The Horn System 24
2.8 The Brake System 25
2.9 The Charging Material 25
2.10 Wheels 26
2.11 The Footrest 27
2.12 Chair & Seat 28
2.13 Conclusion 28
CHAPTER 3 Software and Hardware components of the Digital circuit 29
3.1 Introduction 29
3.2 Overview of the software components 29
3.2.1 AVR Studio 5.0 29
3.2.2 Extreme Burner AVR 34
3.3 Overview of hardware components 37
3.3.1 USBASP AVR Programmer 38
3.3.2 Microcontroller ATmega32A 43
3.3.3 LCD Display (20X4) 46

3.4 Conclusion 49
CHAPTER 4 Integration of the torque sensor into the System 50
4.1 Introduction 50
4.2 Introduction to the Torque Sensor 50
4.3 Types of Torque Transducers and Torque Sensors 51
4.4 Applications 51
4.5 Features of Torque Sensor 52
4.6 Technical Parameter Data 52
4.7 Electrical Connections 53
4.8 Power Management 54
4.9 Conclusion 57
CHAPTER 5 Overview of the Analog circuit 58
5.1 Introduction 58
5.2 The Control Algorithm 58
5.3 Key Observations of the system and Implementation of the results 59
5.4 Design Proposal 61
5.5 Circuit Design I 61
5.5.1. Introduction 61
5.5.2. The Circuit-Diagram and Explanation 62
5.5.3. Simulation and Results 62
5.6 Circuit Design II 64
5.6.1 Introduction 64
5.6.2 The Circuit-Diagram and Explanation 64
5.8 Conclusion 65
CHAPTER 6 Development of the control algorithm for the microcontroller based circuit 66
6.1 Introduction 66
6.2 The Control Algorithm 66
6.2.1 ADC conversion in microcontroller (ATmega32A) 66
6.2.2 Output Calculation 68
6.2.3 Pulse Width Modulation (PWM) 69
6.3 Flowchart of the Control Circuit 73
6.4 Conclusion 75
CHAPTER 7 The microcontroller based circuit design 76
7.1 Introduction 76
7.2 The block diagram for the external circuit 76
7.3 The proposed design 77

7.3.1 Introduction 77
7.3.2 Circuit diagrams and explanations 77
7.3.3 The complete circuit design 81
7.4 Lab Simulation 81
7.4.1 Lab simulation data and Results 83
7.4.2 Comments 84
7.5 Conclusion 85
CHAPTER 8 Field test and analysis 86
8.1 Introduction 86
8.2 Data Acquisition methods 86
8.3 SOC of the Battery 86
8.4 Field test data with the microcontroller based circuit 87
8.4 Field test data collected with the analog circuit 89
8.5 Comparative study 91
8.6 Overall analysis of the microcontroller based digital circuit 93
8.7 Conclusion 94
CHAPTER 9 Conclusion 95
9.1 Summary 95
9.2 The PCB design 95
9.3 Future Works 97
REFEENCES 98
APPENDIX 101
APPENDIX A 102
APPENDIX B 103
APPENDIX C 123
APPENDIX D 126

LIST OF FIGURES

Fig 1.1: Crutches and Artificial limbs
Fig 1.2: Manual wheel chair (foldable)
Fig 1.3: Joystick controlled wheelchair
Fig 1.4: Three-wheeler peddling wheelchair
Fig 1.5: Percentage of total physically challenged people
Fig 1.6: Existing Electrically Assisted Wheelchair with Torque Sensor

Fig 2.1: Side View of the Wheelchair
Fig 2.2: Front View of the Wheelchair and Rear View of Wheelchair
Fig 2.3: Top View of Wheelchair
Fig 2.4: Hub Motor & the Adjusted Rim
Fig 2.5: Two Sets of Batteries
Fig 2.6: Torque Sensor Paddle & Module
Fig 2.7: Motor Controller
Fig 2.8: Wiring Diagram of Wheelchair
Fig 2.9: Throttle
Fig 2.10: Horn System
Fig 2.11: Hand Clutch
Fig 2.12: Charger
Fig 2.13: (1) Front Wheel (2) & (3) Back wheels
Fig 2.14: Footrest
Fig 2.15: Chair & Seat

Fig 3.1: Opening page of AVR Studio 5.0
Fig 3.2: New window for project information
Fig 3.3: Chip selection of AVR Studio
Fig 3.4: AVR Studio New Project Start Screen
Fig 3.5: Sample of a successful compilation of an algorithm in AVR Studio 5
Fig 3.6: A sample of generating hex file in AVR Studio 5.0
Fig 3.7: Extreme Burner AVR opening
Fig 3.8: Burn Progress in Extreme Burner AVR
Fig 3.9: Task Completed Successfully
Fig 3.10: USBASP AVR Programmer
Fig 3.11: List of supported microcontrollers for USBASP
Fig 3.12: USBASP device layout
Fig 3.13: LEDs of USBASP
Fig 3.14: 10 ISP pin-out diagram
Fig 3.15 Connection between the USBASP and ATmega32A on breadboard
Fig 3.16: Microcontroller (ATmega32A)
Fig 3.17: Pin configuration diagram
Fig 3.18: LCD display (20X4)
Fig 3.19: Pin configuration for LCD (20X4)

Fig 4.1: Complete set-view of torque sensor and other components
Fig 4.2: Torque sensor and module
Fig 4.3: The torque-sensor pedal and module attached to the wheelchair
Fig 4.4: Electrical connection diagram for independent operation of the sensor and module
Fig 4.5: Overview of the power management and signal flow of the analog circuit

Fig 4.6: Complete overview of the power management and signal flow

Fig 5.1: Block diagram of the design proposal
Fig 5.2: Circuit design I of the analog circuit
Fig 5.3: Simulation results for the circuit design
Fig 5.4 Circuit design II of the analog circuit

Fig 6.1: Sample ADC conversion
Fig 6.2: 8-bit counter
Fig 6.3: Fast PWM (non-inverting and inverting mode)
Fig 6.4: Flowchart of our control Algorithm

Fig 7.1: Block diagram of the proposed design
Fig 7.2: Power module circuit
Fig 7.3: Microcontroller and the supplementary circuits
Fig 7.4: Complete circuit design
Fig 7.5: Circuit implemented on breadboard
Fig 7.6: Lab simulation results

Fig 8.1: Input voltages (V) from torque sensor module to the microcontroller circuit and the output voltages (V) from the circuit to the C.U of the motor Vs. Time (seconds)
Fig 8.2: Motor Current (A) supplied by battery and the Motor voltage (V) Vs. Time (s)
Fig 8.3: Power consumption profile of the digital circuit
Fig 8.4: Comparison of torque sensor output voltage (V) & input voltage (V) from torque sensor to controller unit with respect to time (s) of the analog circuit
Fig 8.5: Voltage and current profile with the analog circuit
Fig 8.6: Power consumption profile for the analog circuit
Fig 8.7: Time calculation for digital circuit and analog circuit
Fig 8.8: Distance travelled with digital and analog circuit
Fig 8.9: Energy consumption calculation with digital and analog circuit

Fig 9.1: Top silk view of PCB
Fig 9.2: Bottom copper view of PCB
Fig 9.3: PCB implemented in the wheelchair

LIST OF TABLES

Table 3.1: Characteristics of USBASP ISP 10 Pin

Table 7.1 Simulation data

Table 8.1 State of Charge as related to Specific Gravity and Open Circuit Voltage

LIST OF ABBREVIATIONS

ACWF = Autistic Children's Welfare Foundation
ADC = Analogue to DC Conversion
ASCII = American Standard Code for Information Interchange
BEV = Battery Electric vehicle
BLDC = Brushless DC
CARC = Control Application Research Centre
CARG = Control and Research Group
CDD = Center for Disability in Development
CGRAM= Character Generated Random Access Memory
DDRAM= Double Data Rate Dynamic Random Access Memory
DDRx= Port Direction Register
DoD = Depth of Discharge
EEPROM = Electrical Erasable Programmable Read Only Memory
GUI = Graphical User Interface
HV = Hybrid Vehicle
ICSP = In Circuit Serial Programming
IDC = Insulation Displacement Connector
IDE = Integrated Development Environment
ISP = In-System Programming
LCD = Liquid Crystal Display
LVD = Low Voltage Discharge
MCU = Microcontroller Unit
MISO = Master In-Slave out
MOSI = Master Out-Slave In
PHEV = Plug-in Hybrid Vehicle
PINx =Port Input Register
PORTx = Port Output Register
PWM = Pulse with Modulation
RS = Register Select
SCK = Serial Clock
SOC = State of Charge
WHO = World Health Organization

CHAPTER 1

Introduction

1.1 Introduction to Wheelchair

As the days pass by, the livelihood of physically challenged people becomes hard especially in our current fast-paced society where existence is valued by a person's daily income and contribution to his or her family. For them constant attending to is required by another person for simple tasks like mobility from places to places, which in turn become burdensome for the other person. Thus, society as a whole has taken assistive measures, in order to ease some of the problems of the physically challenged people who are constantly fighting a handicapped battle for the rest of their remaining life. These measures include devices like crutches, wheelchair either manual or electric, and replacing with artificial limbs. In Bangladesh, different types of wheelchair are available for utilization which includes the most recent electrically assisted wheelchair that CARG under BRAC University undertook and developed.

Fig 1.1: Crutches and Artificial limbs

For a developing country like ours, where road accidents are frequent and are a serious problem which contributes to the rising number of physically challenged people, the country lacks more efficient method of movement for disabled populace, such as the modern electric wheelchairs which are hard to see. More often we see various types of manual wheelchairs among which majority of the share are held pre-dominantly by folding type wheel chair. These wheelchairs are portable and can be folded when not in use. Manual wheelchairs accommodates its user to rotate the rear wheels with both hands for locomotion, which are less expensive compared to electrical wheelchair but requires large amount of labor from its user.

Fig 1.2: Manual wheel chair (foldable)

The most advanced type of wheelchair is “Joystick Wheelchair”, which shares similar physical structures and features to that of the manual wheelchair. However, electricity from battery source powers the whole wheelchair and the joystick movements. A motor is installed, which makes the wheels rotate whenever the joystick is moved and the whole wheelchair moves along with the direction of the joystick’s movement. This requires complex algorithm and mechanism, which later proved to be expensive when installed in the final product. Moreover, the user needs a lot of mental preparation for smooth controlling of this wheelchair.

Fig 1.3: Joystick controlled wheelchair

The wheelchairs mentioned above are built for indoor purposes only. In Bangladesh, we see a common type of three-wheeler paddling wheelchair, which is used for outdoor purposes. Beggars and disabled people with limited motion of their bodies commonly use this type of wheelchair. These wheelchairs usually consists of three wheels, one at front and two at rear, of which chain is connected in either front or rear wheel only. The wheelchair is driven by paddling using both hands. However, due to inadequate or flawed design, this is unsuitable for either indoor or outdoor use in the terrain of our country.

Fig 1.4: Three-wheeler peddling wheelchair

1.2 Main concern of physically challenged people in Bangladesh

The World Health Organization has defined health as ―a complete physical, mental and social well-being and not merely the absence of disease or infirmity. According to this organization -disability is an umbrella term, covering impairments, activity limitations, and participation restrictions. [32]

Bangladesh is a developing nation and is the home to around 160 million people. As indicated by CDD's (Center for Disability in Development) report estimated 44.3 % of the populace living beneath the poverty line, with insufficient healthcare, education and social security services, low employment rate and at higher peril from natural calamities, especially flooding. CDD's report estimates that 10% of the populace of 16 million is living with disability, and these are a standout among the most vulnerable group, as they get almost no help from the government and not-for-profit organizations, those conveying development projects to address their circumstance. In the nation yet at the same time, these developmental projects penetrate to a little extent in the entire population. [33] Here, we ought to specify the WHO's estimation that 10% of any given populace are disabled. [32] It is observed from a study of the reasons for disability in Bangladesh, which clearly indicates that, 46 percent of the victims, experiences some sort of defects since birth (e.g. like victims affected from polio virus) . Remaining 29 percent reported debilitations due to a disease, 17 percent because of accidents, and 3 percent brought about by malnutrition and 5 percent because of other outside stuns or distressing social circumstances. [34]

A figure demonstrating the insight is given below,

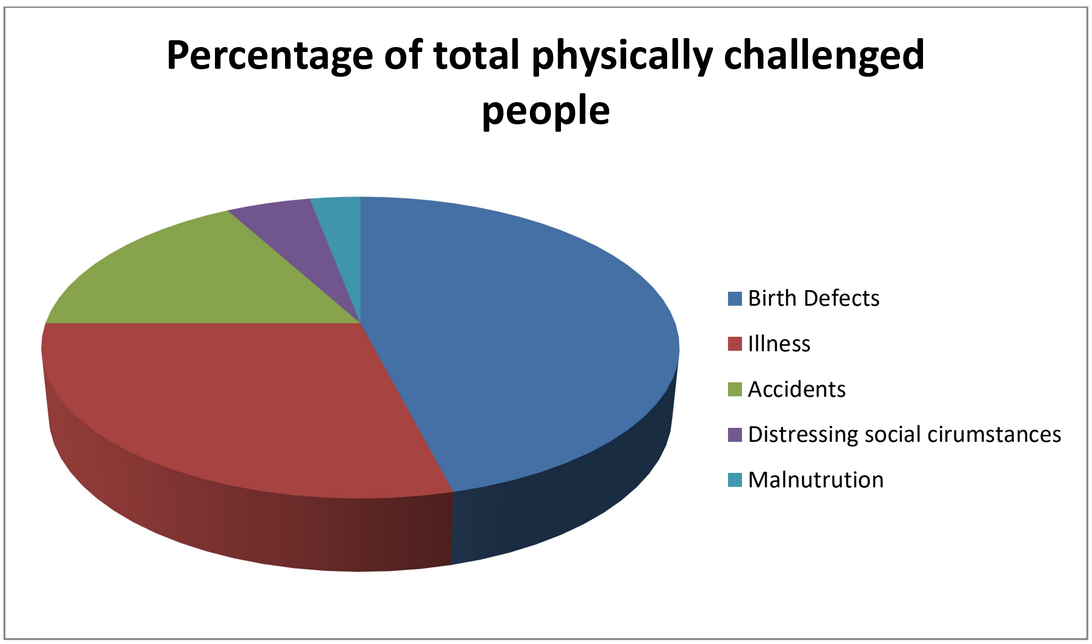


Fig 1.5: Percentage of total physically challenged people

It is found from Autistic Children's Welfare Foundation (ACWF) that out of each 94 young men, one is influenced by Autism. What's more, for young women, it is one in each 150 young women where in Bangladesh. No survey has been done yet, it is accepted that around 3, 00,000 children are affected. [35] These individuals require wheelchair or other kind of instrument to run their life. This venture principally focuses on these 16 million physically challenged people.

1.3 Motivation

CARG under BRAC University undertook a research to develop an Electrically Assisted Wheelchair with Torque Sensor and Dedicated Solar Charger Kit for disabled people in previous year. The project's fruition leads to leap of expectations for incorporating to several other projects where this electrically assistive control mechanism can be used. The new developments that are required to be incorporated in order to enhance the previously existing electrical wheelchair, is to achieve finite control and precision and low power consumption. Furthermore, motivation for this project comes from the desire to improve and modernize the existing wheelchair to increase the comfort level of the patient. We want to develop a control system that would assist the patient, via a motor, only during the extreme times as monitored by a torque sensor thereby saving energy and limiting overuse of the motor. Here by transforming the control mechanism from analog based to digital based embedded system by use of micro controllers, the objectives are met, which will be explained in later chapters.

Our goal is to develop an algorithm by the help of a microcontroller, which can be implemented, in a wheelchair or any other similar non-motorized mechanical vehicles (e.g. in a rickshaw).

Fig 1.6: Existing Electrically Assisted Wheelchair with Torque Sensor

1.4 Outline of the subsequent chapters

In the second chapter of our paper, an overview of the existing system of the electrically assisted wheelchair is given along with a description of its features. The third chapter introduces the hardware and software components of the new control system that has been designed for replacing the analog circuit. The fourth chapter discusses the integration of the torque sensor in to the system and introduces the modified power management system that ensures no malfunction and damage of the circuit component occurs, when power is being supplied. Meanwhile, the fifth chapter portrays the analog circuit implemented earlier. The sixth chapter embodies the control algorithm to be executed using the microcontroller and justifies the algorithm. The consequent chapter presents the circuit design of the digital circuit regulated by the microcontroller along with the auxiliary circuitry (LCD display, gain circuit, torque sensor circuit). The eighth chapter evaluates the prototype in practical road conditions when installed in the wheelchair and represents the data obtained and its analysis. The chapter goes on to further distinguish the differences between the two different circuits, analog and digital circuit, and the advantage of the latter circuit. Finally, our last chapter concludes our thesis paper.

CHAPTER 2

Overview of the existing system

2.1 Introduction

The electric hybrid vehicle that is used to implement and test the completed digital circuit is the electrically assisted wheelchair designed and developed by CARC. This electric wheelchair has been manufactured to have a light weighed modern steel body and its architecture is different from the existing wheelchairs. The massive modernization of such electric wheelchair will improve the lifestyle of huge number of disabled people. One of our main aspects of this vehicle is the use of torque sensor pedal and solar battery charging kit to help conserve power consumption. The electric wheelchair has a hub motor, three 12V 12Ah lead acid batteries connected in series, a controller unit, a throttle, a power key, an LVD charge controller, a front wheel break system and other components. Various views of the customized electric wheelchair used in our experiment are shown below.

Fig 2.1: Side View of the Wheelchair

Fig 2.2: Front View of the Wheelchair and Rear View of Wheelchair

Fig 2.3 Top view of wheelchair

2.2 Hub Motor

The basic working principle of this motor is the turning of stored electricity into magneto motive power; electric current is supplied to a tightly coiled wire that sits between the poles of a magnet and the coil spins around creating a force that can turn a wheel and drive a machine.

Most electric-powered vehicles (electric cars, electric bicycles, and wheelchairs) use on-board batteries and a single, fairly ordinary electric motor to power either two or four wheels. However, some of the recent electric cars and bicycles work in a different way. Instead of using one motor to power all the wheels by either gears or chains, these electric vehicles have a built-in motor directly inside the hub of each wheel—so the motor and wheel are one and the same. This type of motor is called hub motor.

A hub motor is fitted and energized in the front wheel of the wheelchair and has a power of 250W. A rim is associated adjacent to the hub motor for the pedal chain. The fundamental mechanism of a hub motor is almost similar to the normal dc motor. The primary contrast between the two types of motor is that the hub motor stays inside of the wheel and weighs less. Moreover, the hub motor does not consume any additional room to keep it inside the wheelchair. The inside structure of the hub motor could not be shown due to lack of available resources.

Fig 2.4: Hub Motor & the Adjusted Rim

2.2.1 Comparison with Conventional EV Design in Automobiles

1. **Drive by wire**

    - ❖ Vehicles with electronic control of brakes and acceleration provide more opportunities for computerized vehicle dynamics such as:

        - Active cruise control, where the vehicle can maintain a given distance from a vehicle ahead
        - Collision avoidance, where the vehicle can automatically brake to avoid a collision
        - Emergency brake assists, where the vehicle senses an emergency stop and applies maximum braking
        - Active software differentials, where individual wheel speed is adjusted in response to other inputs
        - Active brake bias, where individual wheel brake effort is adjusted in real time to maintain vehicle stability
        - Brake steer, where individual wheel brake bias is adjusted to assist steering (similar to a tracked vehicle like a bulldozer)

    - ❖ As wheel motors can brake and accelerate a vehicle with a single solid state electric/electronic system, many of the above features can be added as software upgrades rather than requiring additional systems/hardware be installed like with ABS etc. This should lead to cheaper active dynamic safety systems for wheel motor equipped road vehicles.

2. **Weight savings**

    Eliminating mechanical transmission including gearboxes, differentials, drive shafts and axles provide a significant weight and manufacturing cost saving, while also decreasing the environmental impact of the product.

3. **Unsparing weight concerns**

    The major disadvantage of wheel hub motors are that the weight of the electric motors would increase the unsparing weight, which adversely affects the handling and ride (the wheels are more sluggish in responding to road conditions, especially fast motions over bumps, and transmit the bumps to the chassis instead of absorbing them). Most conventional electric motors include ferrous material composed of laminated electrical steel. This ferrous material contributes to most of the weight of electric motors. To minimize this weight several recent wheel motor designs have minimized the electrical steel content of the motor by utilizing a coreless design with Litz wire coil windings to reduce eddy current losses. This significantly reduces wheel motor weight and therefore unsparing weight. [1]

2.3 Batteries

Three rechargeable 12V, 12Ah lead-acid batteries have been used for the wheelchair. The batteries are connected in series and supply a total of 36V, 12Ah to the hub motor. Each battery is 6x4x3.56 inch in dimension. Two sets of batteries have been used. One set is for the wheelchair while another is for solar battery charger kit. The batteries will be swapped when necessary. The weight of each set of batteries is 13.75 kg. As shown in fig. 2.5, the batteries are placed under the seat and are attached to the bottom sheet of the wheelchair. This ensures that the batteries will not be able to move when the wheelchair is being driven and makes it easy to change the batteries when required. Each fully charged battery provides a voltage of 12.7V or above and the voltage across the terminals is 38.1V after connection in series combination.

Fig 2.5: Two sets of batteries

2.4 The Torque Sensor

The torque sensor is a device which is used to measure and record the torque of a rotating system. [2] It needs a biasing voltage of 5 volt from the DC source to operate. If the torque applied to the pedal of the torque sensor increases the corresponding output voltage will increase. The speed of the motor is directly proportional to the output voltage of the torque sensor. Fig 2.6 shows the torque sensor pedal that has been installed in the wheelchair.

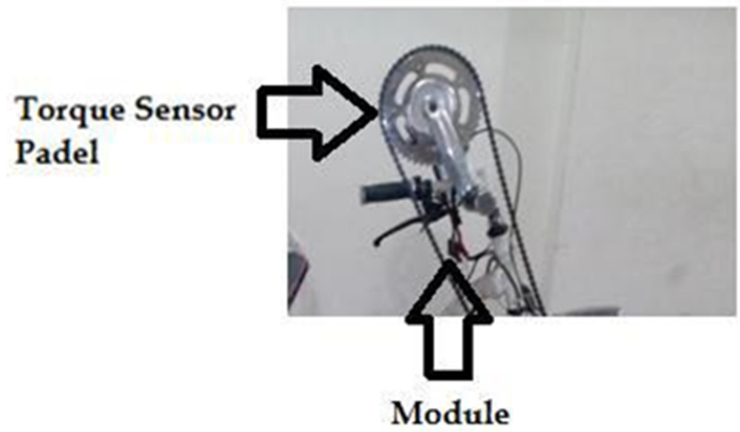


Fig 2.6: Torque Sensor Paddle & Module

2.5 Motor Controller

Motor controller is used to control all electrical part along with motor in parallel. General specification like low voltage protection is 31.5V and maximum current it can carry is 17A. It is kept under the seat of the wheelchair. Motor controller along with wiring diagram has been shown in Fig 2.7 and Fig 2.8.

Fig 2.7: Motor Controller

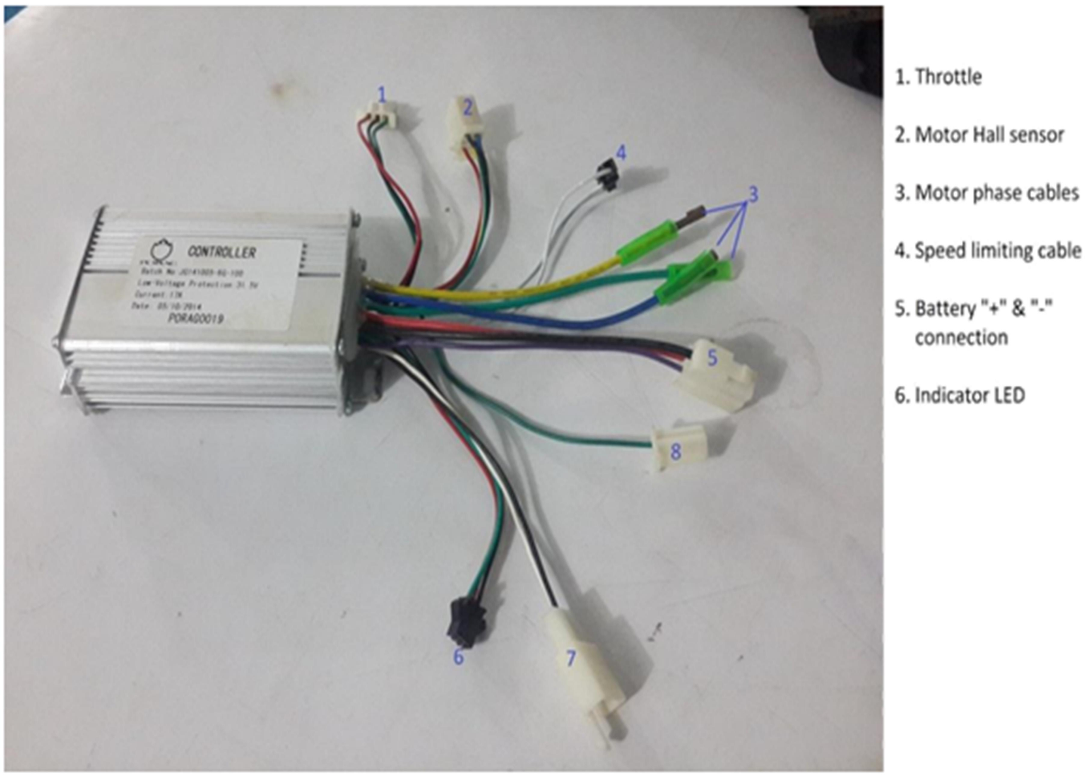


Fig 2.8: Wiring Diagram of Wheelchair

2.6 Throttle

A throttle is a specially designed potentiometer and is used to control the speed of the motor. It needs a biasing voltage of 5V which is provided by the motor controller unit. The output voltage depends on the angle of the throttle and this voltage is supplied to the controller which controls the speed of the motor. The speed of the motor depends on the output voltage and the motor speed increases as the output voltage increases. It has been found from the field test that the when the output voltage is 1.44V the motor starts and when the output voltage is 3.5 volt the motor rotates at its maximum speed.

Fig 2.9: Throttle

2.7 The Horn System

This wheel chair has a system which is rather uncommon in respect to other wheelchairs. Existence of this horn system will help to run the wheelchair properly on main roads of the country. It is near to throttle as can be seen in below in fig 2.13. The horn system is connected through the motor controller and when the button is pressed the system will take 5V from the battery to the horn.

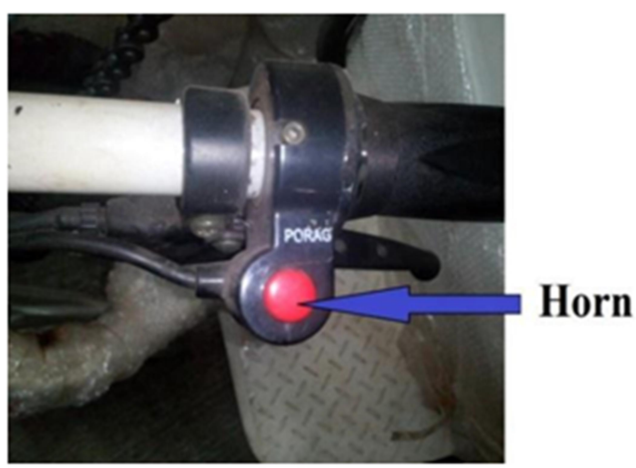


Fig 2.10: Horn System

2.8 The Brake System

Hand brake is used in the wheelchair. The brake system is constructed on the front wheel. Traditional hand clutch is used to stop the front wheel of the wheelchair which is normally used in rickshaw or bi-cycle. This hand-clutch is used to stop both the wheel and the motor at the same time. When the hand-clutch is released, it returns to its original position and allows the motor to start again when needed.

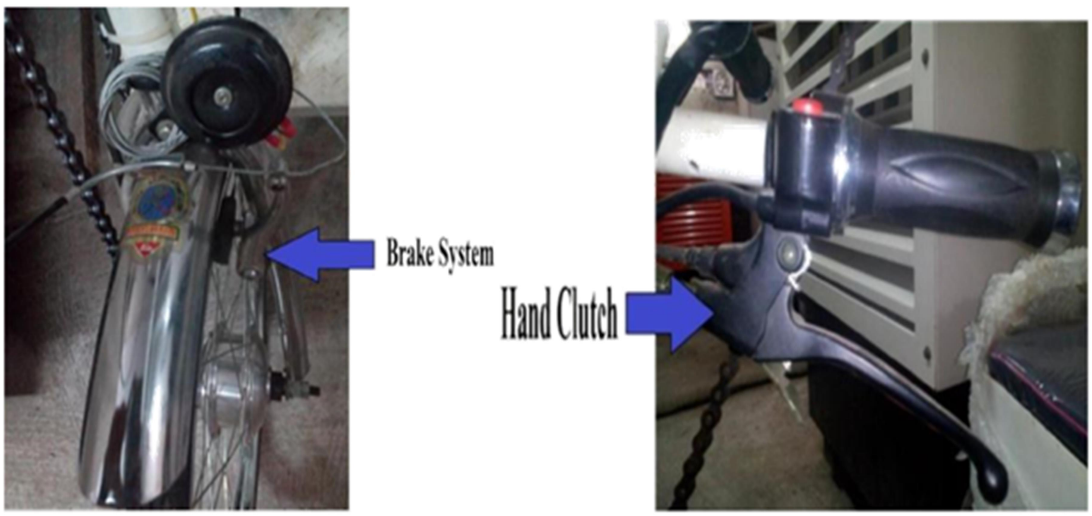


Fig 2.11: Hand Clutch

2.9 The Charging Material

There are two charging systems for the wheelchair. One system is replacing the batteries with the second set of batteries from the solar charger kit. The other charging system is provided by the national grid. For that we need a charger which can carry at least 36V, 12A. The charging point is just corner of the footrest which can be easily accessed whenever it is necessary.

Fig 2.12: Charger

2.10 Wheels

The wheelchair consists of three wheels. One wheel is in the front and two wheels are at the back of the wheelchair. The hub motor is connected in the front wheel and there is no motor connection in the back wheels. The back wheels will get power to run from front wheel motor. Normal rickshaw wheels have been used in the wheelchair. Diameter of each of the wheels is 26 inches which is enough to make the wheelchair run properly when carrying a person.

(1)

(2) (3)

Fig 2.13: (1) Front Wheel (2) & (3) Back wheels

2.11 The Footrest

There is a footrest which provides enough space to place the legs. It was kept in mind during design of the wheelchair that paralyzed people; especially those people who have no feeling in their legs may also utilize the wheelchair. For that reason the footrest was made more spacious and wide in respect to the foot length of an average person. The length of the foot rest is 12 inches.

Fig 2.14: Footrest

2.12 Chair & Seat

A comfortable chair is used in our wheelchair because. It is enough spacious to sit properly for a disabled person. Length of the chair is 18 inches. We used comfortable cushions for the seat. We also used cushion in the back side for more comfort. Cushions are covered with polyethylene as our wheelchair will be used mainly for the outdoor activities where polyethylene will give the protection from the dust and rain.

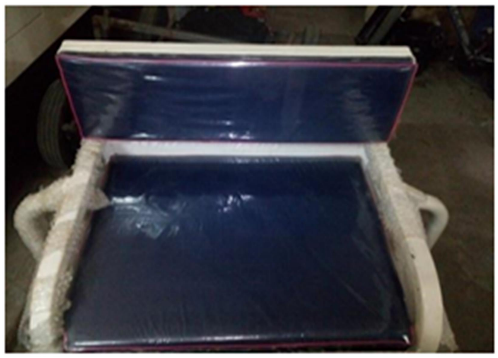


Fig 2.15: Chair & Seat

2.13 Conclusion

This chapter provided information about the electric hybrid vehicle which was taken for field test in order to test the performance of the microcontroller based circuit. Our circuit can replace any of the analog circuit implemented in the electric hybrid vehicles manufactured by CARC. The electrically assisted wheelchair was chosen and the data that was collected during our field tests were compared to the data obtained from the analog circuit taken by the developers of the wheelchair which will discussed in the later chapters.

The upcoming chapter introduces the software and hardware components essential for the development of the microcontroller based circuit design.

CHAPTER 3

Software and Hardware components of the Digital circuit

3.1 Introduction

In this chapter we will discuss about the software programs and hardware components that have been used in writing and implementing the control algorithm that will be discussed in the next chapter. By means of this we will have familiarized ourselves with the understanding of the microcontroller (ATmega32A), starting from its description and features to how it can be incorporated with our system through the help of the software and hardware elements.

3.2 Overview of the software components

AVR programmer is an USB in-circuit programmer for Atmel AVR microcontrollers. With the help of this programmer, it is possible to load hex files in to AVR chips. The AVR chip that we have utilised is the ATmega32A microcontroller. The AVR programmer utilises USB driver software and is directly controlled with port signals since it has no controller is present on the programmer. These programmers can also be worked with AVR Studio. [3]

For our system, AVR Studio 5.0 is chosen as the AVR programmer software which is used to write the code for our microcontroller (ATmega32A). The code written has been constructed on the basis of a control algorithm (discussed in detail in the chapter 5) and the entire code is provided in the Appendix section. In order to burn the written code in to the microcontroller, different software of the AVR programmer is used. It is the Extreme Burner AVR software. Both of these programs are explained in depth in the succeeding subdivisions.

3.2.1 AVR Studio 5.0

Studio AVR 5.0 is a software development environment produced by Atmel and is designed for programming of AVR microcontrollers. Atmel AVR Studio 5.0 is an Integrated Development Environment (IDE) for developing and debugging embedded Atmel AVR applications. Accordingly, this provides us with a seamless and easy-to-use environment for both beginners and experienced developers to write, build, and debug C/C++ and assembler code [10]. A useful advantage of operating this software was the availability of tutorials that aided us to get familiar with the usage and functions of the software.

Features of AVR Studio 5.0: [source: - 4 & 5]

- This software provides a user friendly environment for the editor, simulator, and programmer.
- It has its own integrated C compiler so the AVR GNU C Compiler (GCC) does not require any third party C compiler.
- It permits chip simulation and in-circuit emulation.
- Same User Interface (UI) for simulation and emulation are used here.
- Programs for both 8-bits and 32-bits AVR series of microcontrollers can be developed
- Users can derive supports for this software by its easy access of datasheets, STK500, AVR Dragon etc.
- It has 400 Example Projects.

3.2.1.1 Working with AVR Studio at a Glance

The AVR Studio software is at first downloaded and installed. After installation is complete software is launched, so that a new C project can be created. For creating a new C project, a toolbar named 'New Project' is selected. After selection, a window appears providing the option to save the location of our project along with the project name. Next, we select the chip for our project which is ATmega32A.

Once the above steps have been completed, the main job of writing the code begins. A new window comes into view with the basic algorithm of our selected chip. The header of the algorithm is '#include<avr32/io.h>', which provides various input/output operations such as DDRx, PINx, PORTx. The port names of DDRx, PINx, and PORTx refers to Port Direction Register, Port Input Register and Port Output Register respectively. Within the while loop segment of the basic algorithm, we have to write the algorithm for our system.

The step-by-step working process of AVR Studio 5.0 with appropriate sample pictures are given below-

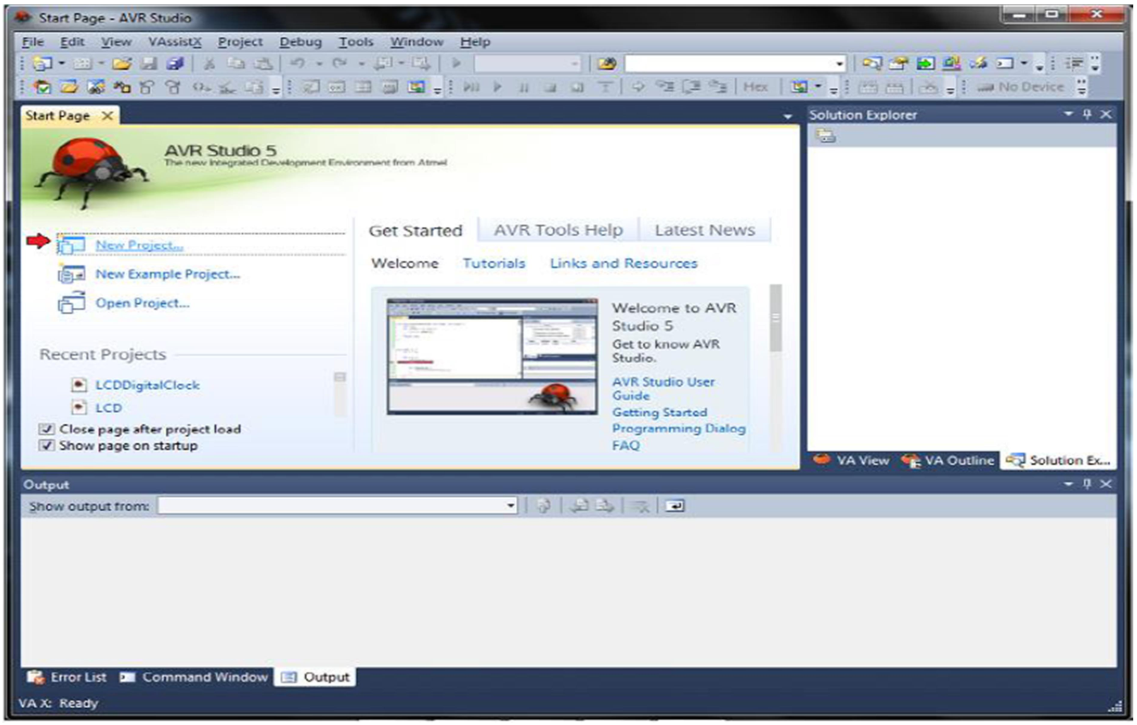


Fig 3.1: Opening page of AVR Studio 5.0 (here we select the 'New Project' as indicated in the figure)

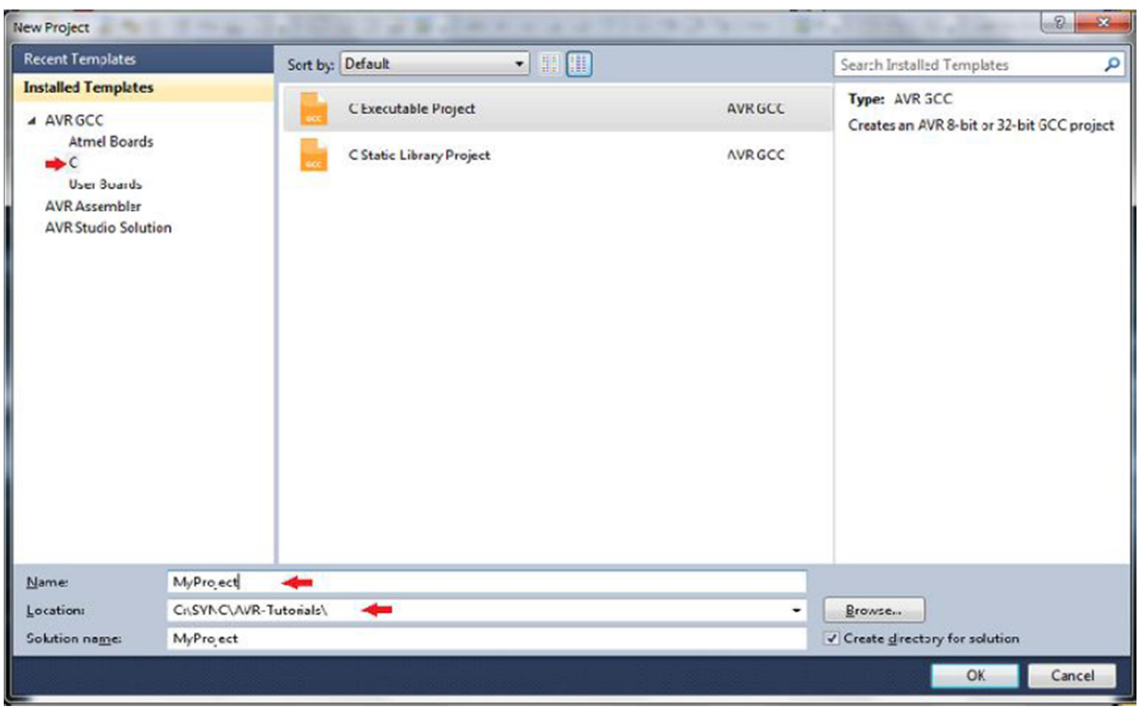


Fig 3.2: New window for project information

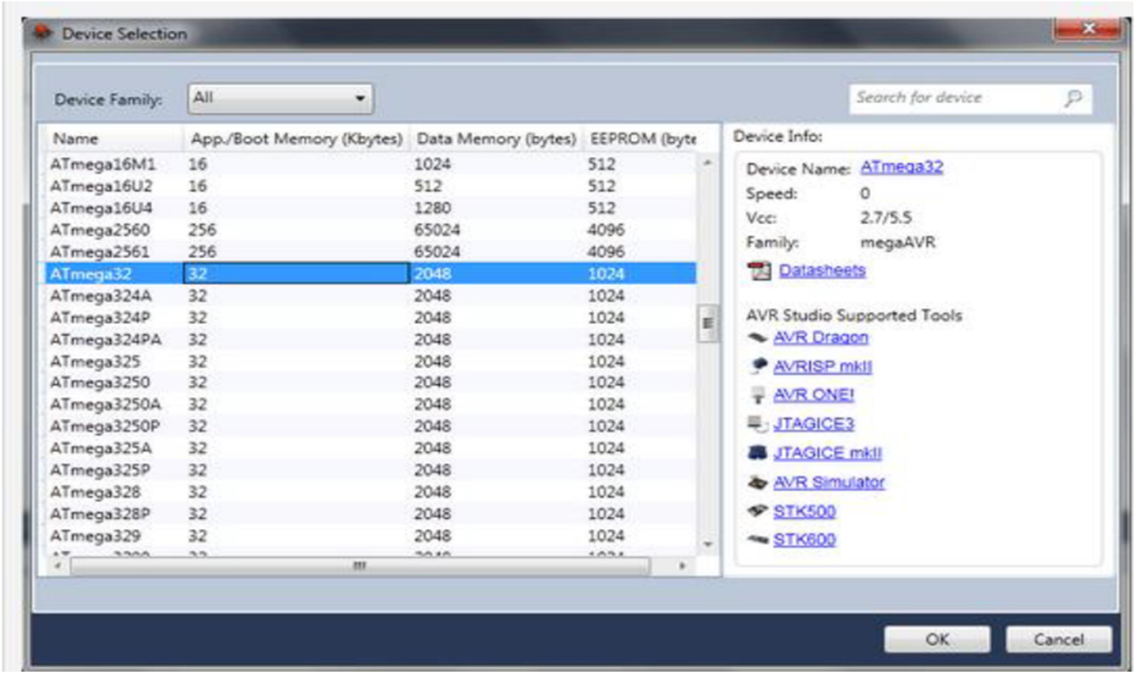


Fig 3.3: Chip selection of AVR Studio

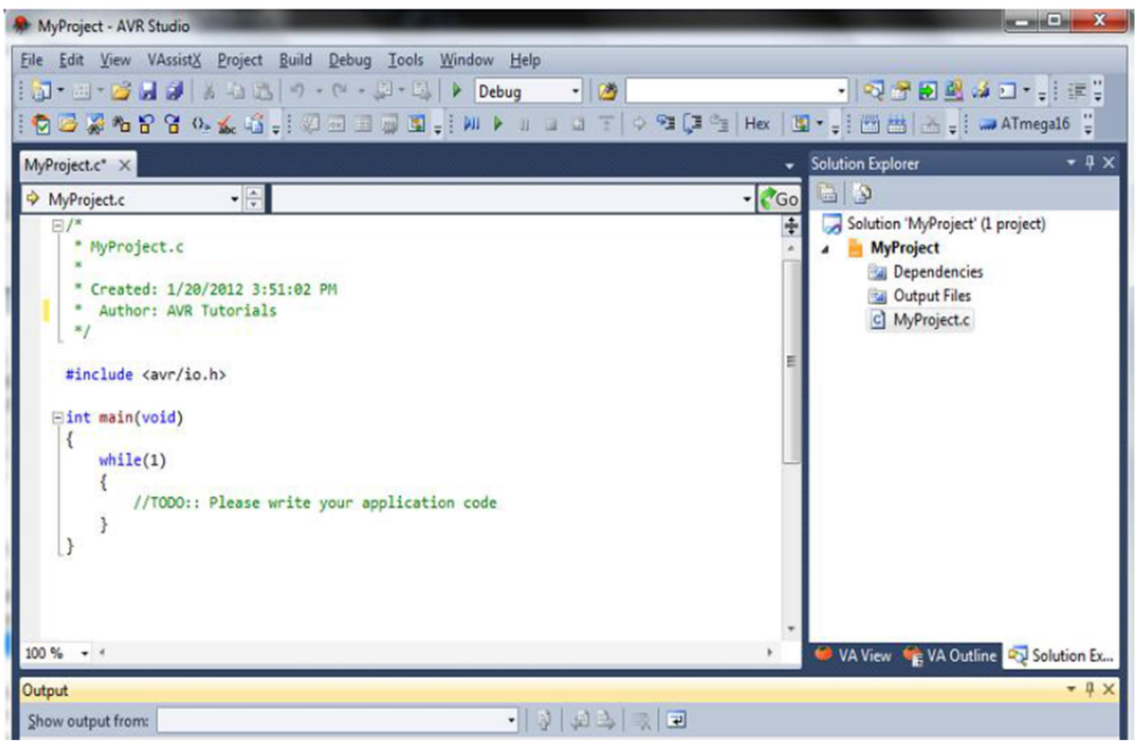


Fig 3.4: AVR Studio New Project Start Screen

After the code representing our algorithm has been written down, it is compiled to check whether there are any errors present in the code. If the code is compiled and built successfully, a message will be displayed at the bottom of the AVR Studio 5.0 Editor indicating 'Build succeeded'. With the code built successfully, we now move on to burning the code into the microcontroller (ATmega32A) using the next software Extreme Burner AVR.

On the other hand, if the message shows an error in the output box we need to recheck the algorithm again and fix the errors that were made while writing the code. In such circumstances, the code needs to be recompiled to ensure that no errors are present and it has been built successfully in order to continue the remainder of the processes.

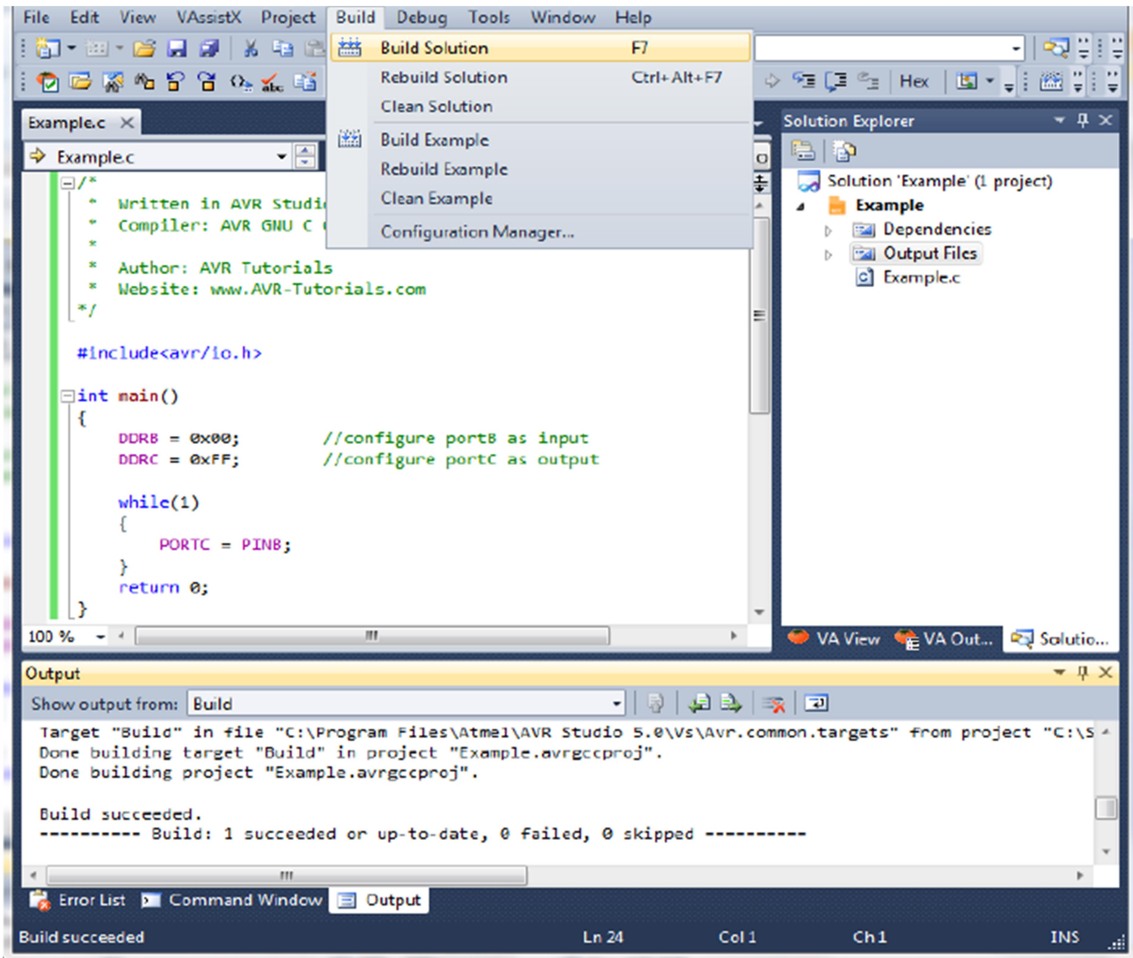


Fig 3.5: Sample of a successful compilation of an algorithm in AVR Studio 5.0 [6]

When compilation of the code is successful, it is then saved as a machine readable file (**.hex** file) for burning of the code into ATmega32A microcontroller using Extreme Burner AVR software. The content of the generated hex file can be viewed by double clicking on the file with the **.hex** extension in the **Solution Explorer**, to the left of the AVR Studio 5.0.

A sample of a generated hex file displayed in the AVR Studio 5.0 Editor shown below-

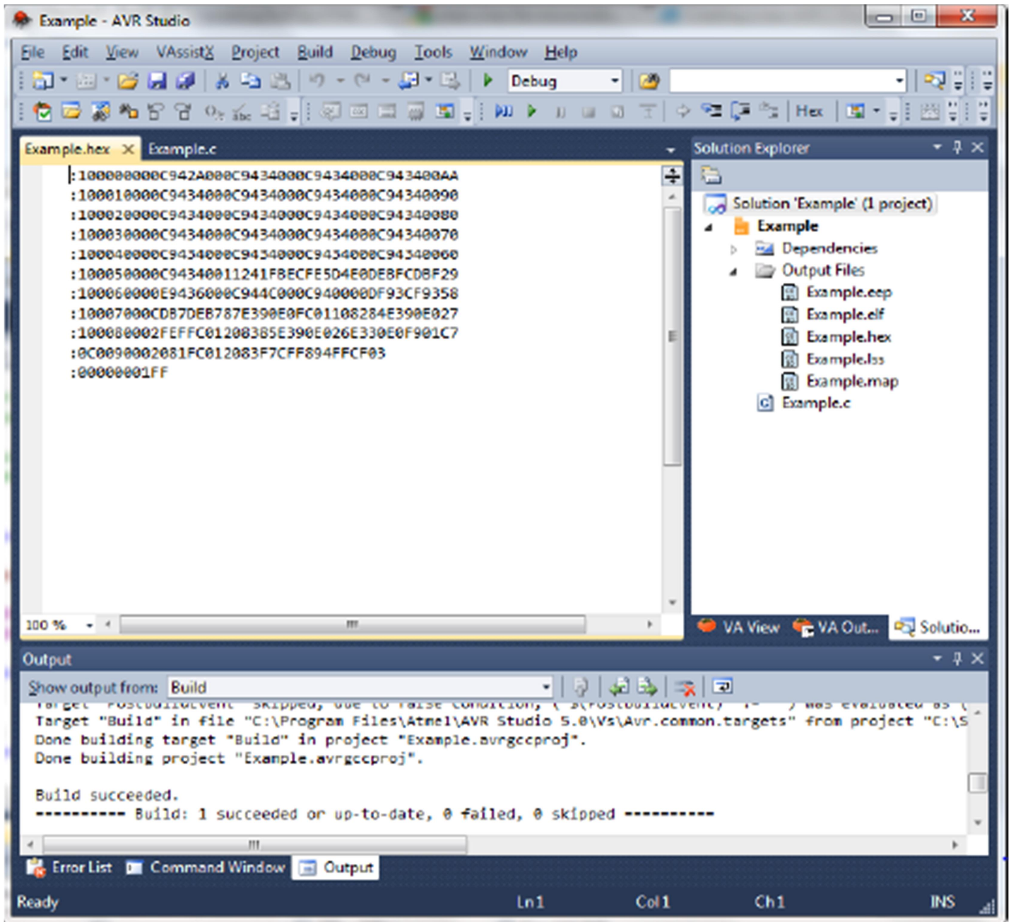


Fig 3.6: A sample of generating hex file in AVR Studio 5.0 [6]

3.2.2 Extreme Burner AVR

Extreme Burner is a low cost USB port that provides for the process of burning hex files of a code in to a microcontroller. It has basic features like In Circuit Serial Programming (ICSP) and High Voltage Programming supports by the Extreme Burner AVR for more than 45 devices [13]. Moreover, Extreme Burner AVR supports several kinds of clock sources for different applications which have graphical user interface (GUI) AVR series of microcontroller. Moreover, this software enables us to read and write a RC oscillator or a perfect high speed crystal oscillator. The clock sources can be selected from the following options:

- external clock
- calibrated Internal RC Oscillator

- external RC oscillator -external low frequency crystal
- external crystal/ceramic resonator

3.2.2.1 Working Process of Extreme Burner AVR

The software needs to be installed at first and during the installation of Extreme Burner Software, USB AVR programmer must be connected to the USB port. After the installation process is complete, Extreme Burner AVR is launched from desktop icon or start menu.

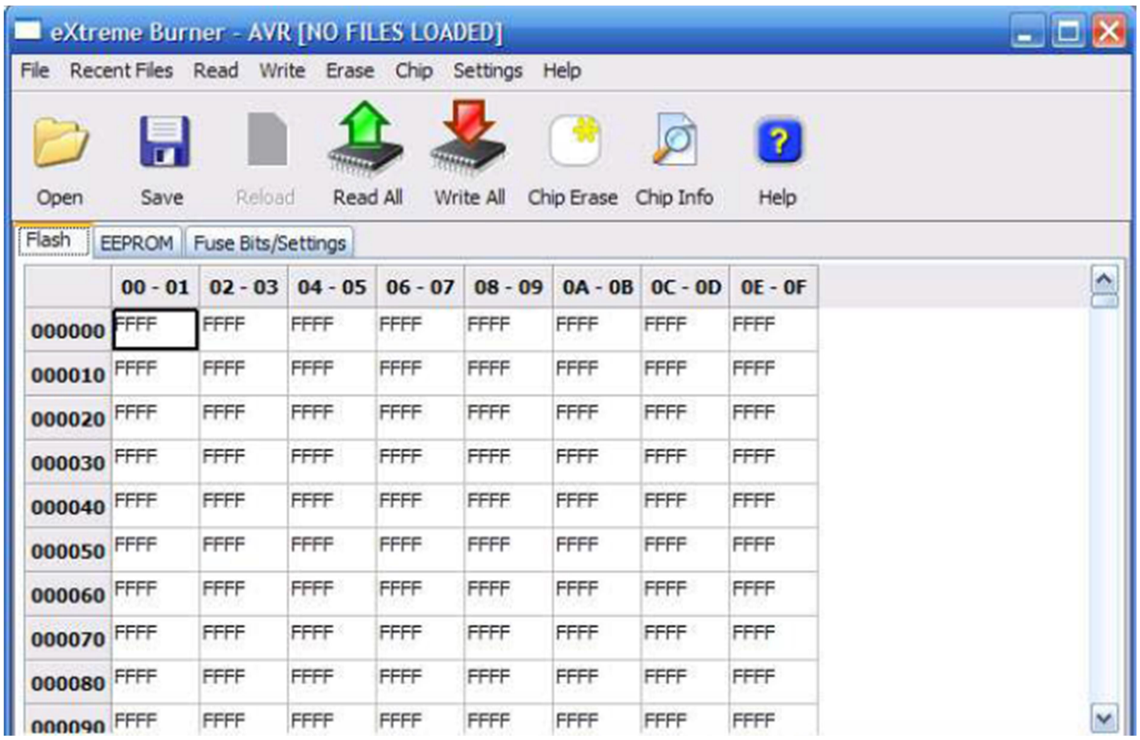


Fig 3.7: Extreme Burner AVR opening page [7]

Before we commence the burning of the algorithm in to the microcontroller, the ATmega32A need to be connected with the USB AVR Programmer according to the pin configuration which will be discussed in section 3.3.

As soon as connection is finalized, the hex file of the code is opened in the Extreme Burner AVR software and the chip (ATmega32A) is selected. After that we select the "**Write All"** icon to burn the program in to the microcontroller. While code is being burned, the Red LED will glow indicating **BUSY** State.

When the chip is programmed properly, a message will appear in the 'Progress' window showing "**ALL TASKS COMPLETED SUCCESSFULLY"**. At this point, we can disconnect the programmer kit from the computer.

The following diagrams exhibit the progress and completion of the burning procedure in Extreme Burner AVR-

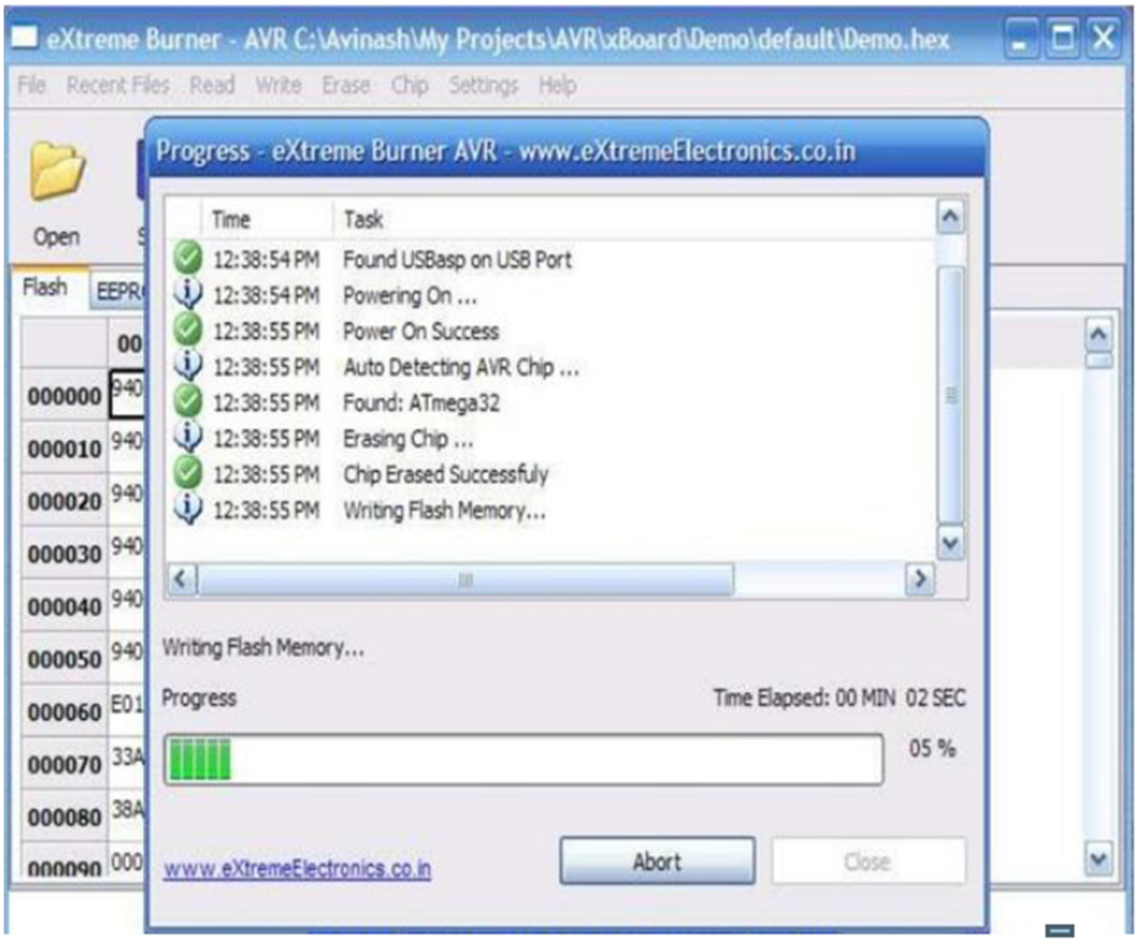


Fig 3.8: Burn Progress in Extreme Burner AVR [7]

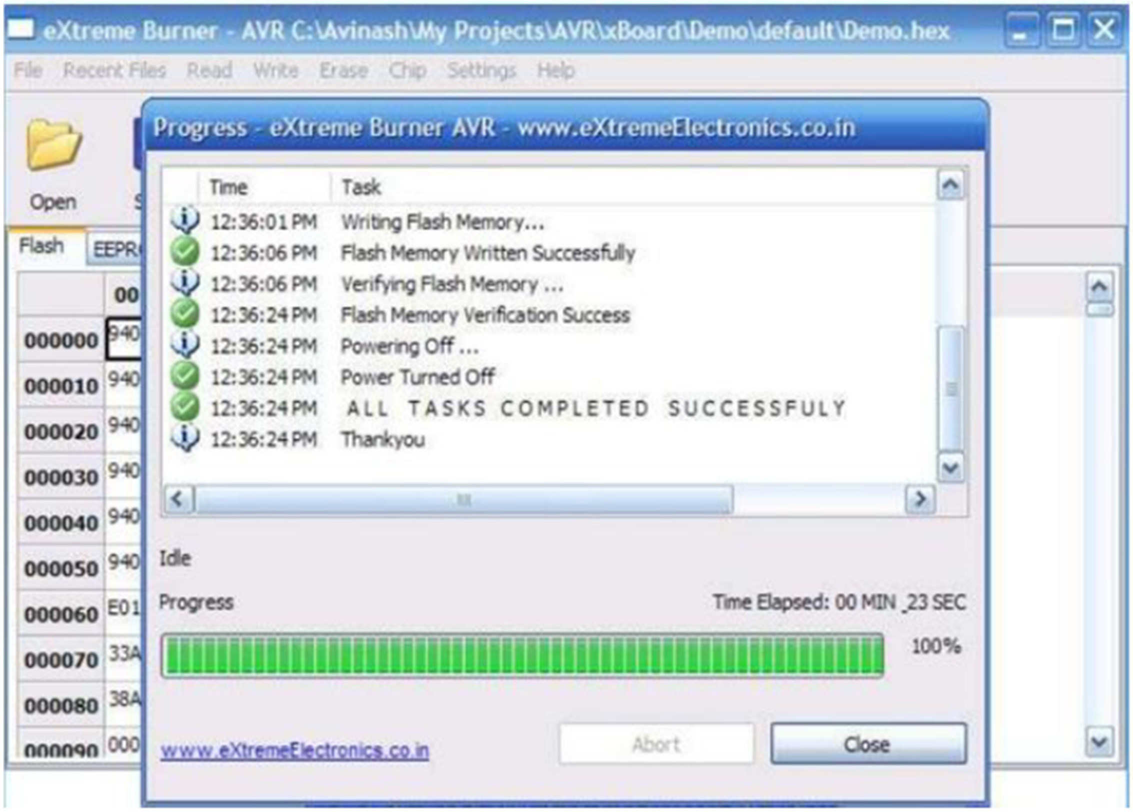


Fig 3.9: Task Completed Successfully [7]

In this section, we have discussed about two different software programs and how both are vital for our project. The AVR Studio 5.0 is used for writing and compiling the algorithm whereas the Extreme Burner AVR software is utilized for burning the written algorithm into the AVR microcontroller. In the following segment we will discuss about the hardware components used in our project.

3.3 Overview of hardware components

Hardware components are the structured combination of physical components which make up a system. The hardware components used for our project and software components mentioned in the earlier section are mutually dependent on each other; without one component the other cannot be utilised to its full extent.

The synopsis of our project is that, we will take a voltage input from the torque sensor pedal and depending on this input the microcontroller will control the C.U of the motor to provide assistance to the user of the electric hybrid vehicle i.e. the wheelchair. Therefore, it can be concluded that the microcontroller and the torque sensor pedal are the core components of our system. Along with the microcontroller some of the other related physical elements will be discussed that were essential for the development of the digital circuit from the analog circuit. The integration of the torque sensor pedal will be described in detail in the next chapter.

The hardware components that are directly related to our digital circuitry and will be explain in this chapter are -

1. USBASP AVR Programmer
2. Microcontroller (ATmega32A)
3. LCD display (20X4)

3.3.1 USBASP AVR Programmer

USBASP is a Universal Serial Bus (USB) in-circuit programmer for Atmel AVR controllers which consists of an ATMega8 and a few passive components. [7] Since it does not require any special USB controller, the programmer uses a firmware driver making the programmer easy-to-use for beginners. USBASP is officially included and supported by WinAVR. Therefore, USBASP is the least expensive option for programming of AVR chips since we are developing a digital circuit that will run an electric hybrid vehicle for use of the poor and the disabled people of Bangladesh.

3.3.1.1Features of USBASP AVR Programmer: [8]

- Designed to read or write the microcontroller EEPROM, firmware, fuse bits and lock bits.
- Supported by Windows, Mac OS X and Linux.
- Programming speed is up to 5kBytes/sec.
- Software controlled SCK (Serial Clock) to support targets with low clock speed (< 1.5MHz)
- 10 pin ISP (In-System Programming) interface with case silver

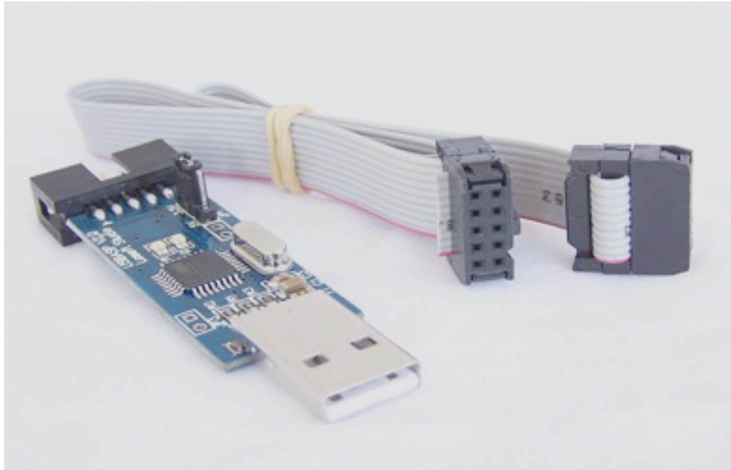


Fig 3.10: USBASP AVR Programmer [8]

The microcontroller ATmega32A can be supported by this programmer as shown below,

Supported Microcontrollers				
Mega Series				
ATmega8	ATmega8A	ATmega48	ATmega48A	ATmega48P
ATmega48PA	ATmega88	ATmega88A	ATmega88P	ATmega88PA
ATmega168	ATmega168A	ATmega168P	ATmega168PA	ATmega328
ATmega328P	ATmega103	ATmega128	ATmega128P	ATmega1280
ATmega1281	ATmega16	ATmega16A	ATmega161	ATmega162
ATmega163	ATmega164	ATmega164A	ATmega164P	ATmega164PA
ATmega169	ATmega169A	ATmega169P	ATmega169PA	ATmega2560
ATmega2561	ATmega32	ATmega32A	ATmega324	ATmega324A
ATmega324P	ATmega324PA	ATmega329	ATmega329A	ATmega329P
ATmega329PA	ATmega3290	ATmega3290A	ATmega3290P	ATmega64
ATmega64A	ATmega640	ATmega644	ATmega644A	ATmega644P
ATmega644PA	ATmega649	ATmega649A	ATmega649P	ATmega6490
ATmega6490A	ATmega6490P	ATmega8515	ATmega8535	
Tiny Series				
ATtiny12	ATtiny13	ATtiny13A	ATtiny15	ATtiny25
ATtiny26	ATtiny45	ATtiny85	ATtiny2313	ATtiny2313A
Classic Series				
AT90S1200	AT90S2313	AT90S2333	AT90S2343	AT90S4414
AT90S4433	AT90S4434	AT90S8515		
AT90S8535				
Can Series				
AT90CAN128				
PWN Series				
AT90PWM2	AT90PWM3			

Fig 3.11: List of supported microcontrollers for USBASP [9]

3.3.1.2 Layout of USBASP

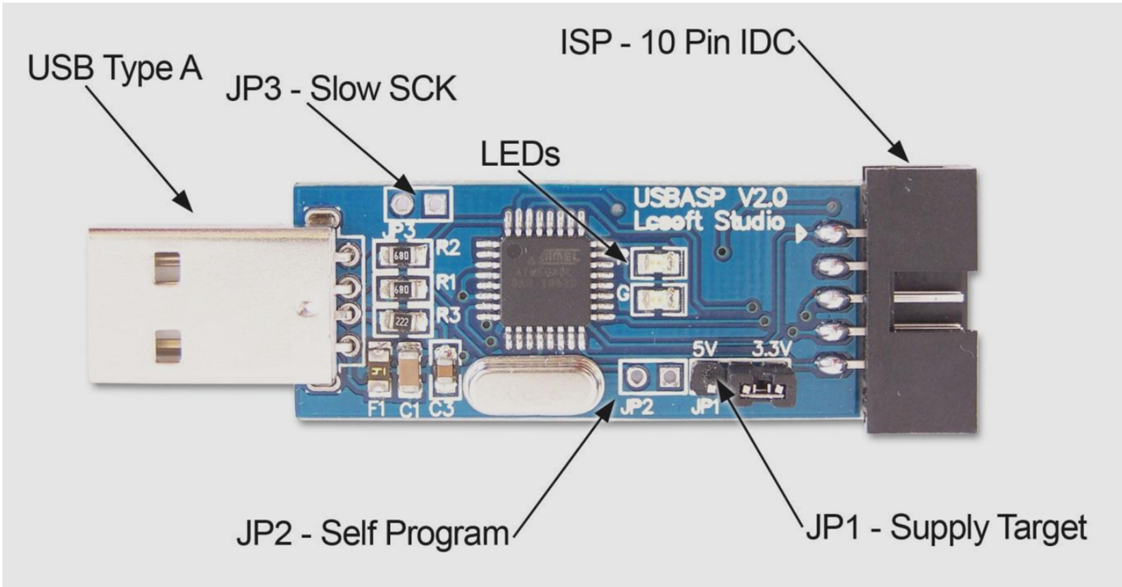


Fig 3.12: USBASP device layout [9]

The USBASP device is composed of several components. They are, [9]

- USB Type A
  - The USB end of the programmer connects directly into your computers USB port.
- JP1-Supply Target
  - This jumper controls the voltage on the ISP VCC connector. It can be set to +3.3V, +5V or disable this jumper if the target device has its own power source.
- JP2-Self Program
  - This jumper is used to update the firmware of the USBASP programmer. In order to update the firmware you will need 2 programmers. One to be programmed and the other to do the programming.
- JP3-Slow SCK
  - When this jumper is selected, the slow clock mode is enabled. If the target clock is lower than 1.5 MHz, you need to set this jumper. Then SCK is scaled down from 375 kHz to about 8 kHz.
- LEDs (Light Emitting Diode)
  - The USBASP programmer has 2 LEDs near the ISP connection. These have the following functions:

    a. LED R – Programmer communicating with target device
    b. LED G – Power

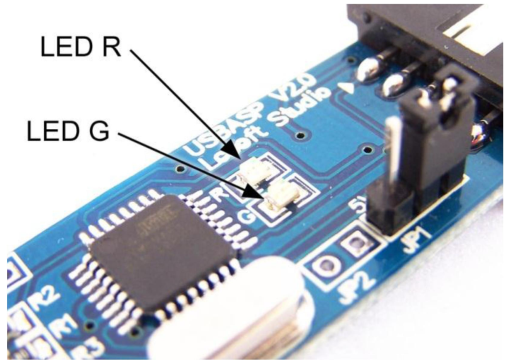


Fig 3.13: LEDs of USBASP [9]

- ISP 10 Pin IDC (Insulation Displacement Connector)
  - The 10 pin ISP connection provides an interface to the microcontroller. This interface uses a 10 pin IDC connector and the pin-out diagram is shown below,

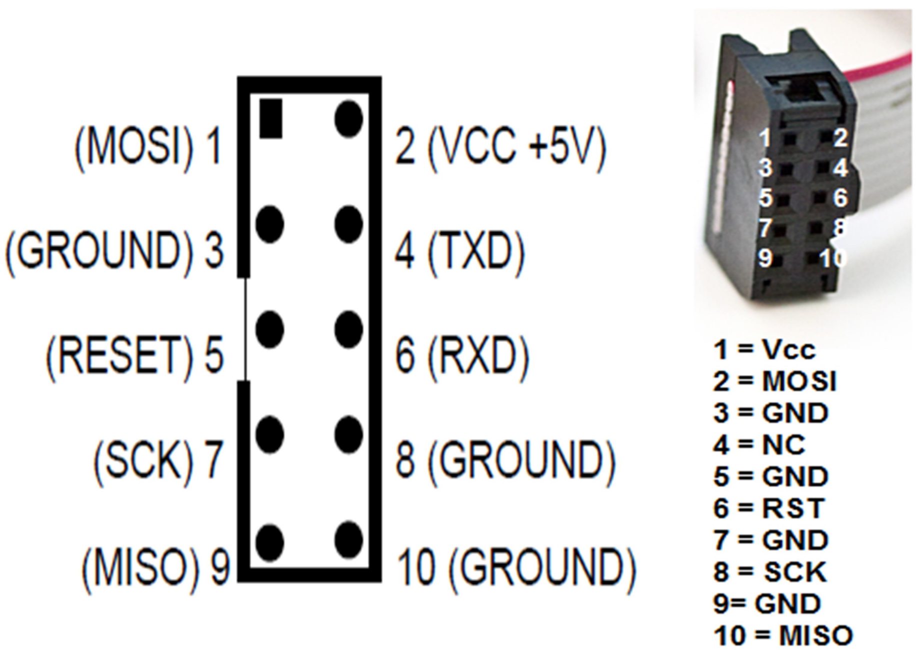


Fig 3.14: 10 ISP pin-out diagram [9 & 10]

3.3.1.3 ISP (In-System Programming) 10 Pin IDC

For interfacing with the microcontroller, 10 pin ISP connections are MOSI, VCC +5V, GND, TXD, Reset, RXD, SCK and MISO. Among these 10 pins only 6 are essential for direct communication between the USBASP and target AVR chip (ATmega32A). Functions of each of these 6 pins are mentioned in the table below,

Pin Name	Description	Comment
MOSI	Master Out-Slave In	This allows master device (the USBASP) to send data to the slave device (target AVR being programmed).
MISO	Master In-Slave Out	This allows the slave device (target AVR being programmed) to send information to master device (the USBASP).
SCK	Serial Clock	This is the mutual clock shared between the master and slave device which provides a common clock signal for synchronized and efficient communication.
Reset	Target AVR Microcontroller Reset	The reset pin for the AVR chip being programmed must be put in active low for programming to occur. When the RST pin is put low, the master slave can communicate on the SCK, MISO, and MOSI lines
VCC	Power	The master (USBASP) and slave (AVR Microcontroller) both need power in order to operate. The highest value they can take is about 5V.
GND	Common Ground	AVR Microcontroller and USBASP must be connected with the common ground for operation. Among the three pins for ground (3, 8 and 10) any of them can be connected to ground because they are internally connected.

Table 3.1: Characteristics of USBASP ISP 10 Pin [10]

The USBASP is the master device and the AVR chip we are programming is the target, or slave device. The master device is does the programming, while the slave device is the subordinate device that is need of being programmed. In order for a master device to program a slave device, they need have communication amongst each other. The master device needs to be able to write data to the slave device's flash memory and be able to receive data from the slave data. This is done through the MOSI and MISO pins. MOSI is the pin by which the master device outputs, or writes, data to the target AVR which is being programmed. MISO is the pin by which the slave device (ATmega32A) can send information to the master device. Both of these communication pathways are essential to program a device. [10]

The connection diagram for connecting the USBASP with the ATmega32A microcontroller is given below,

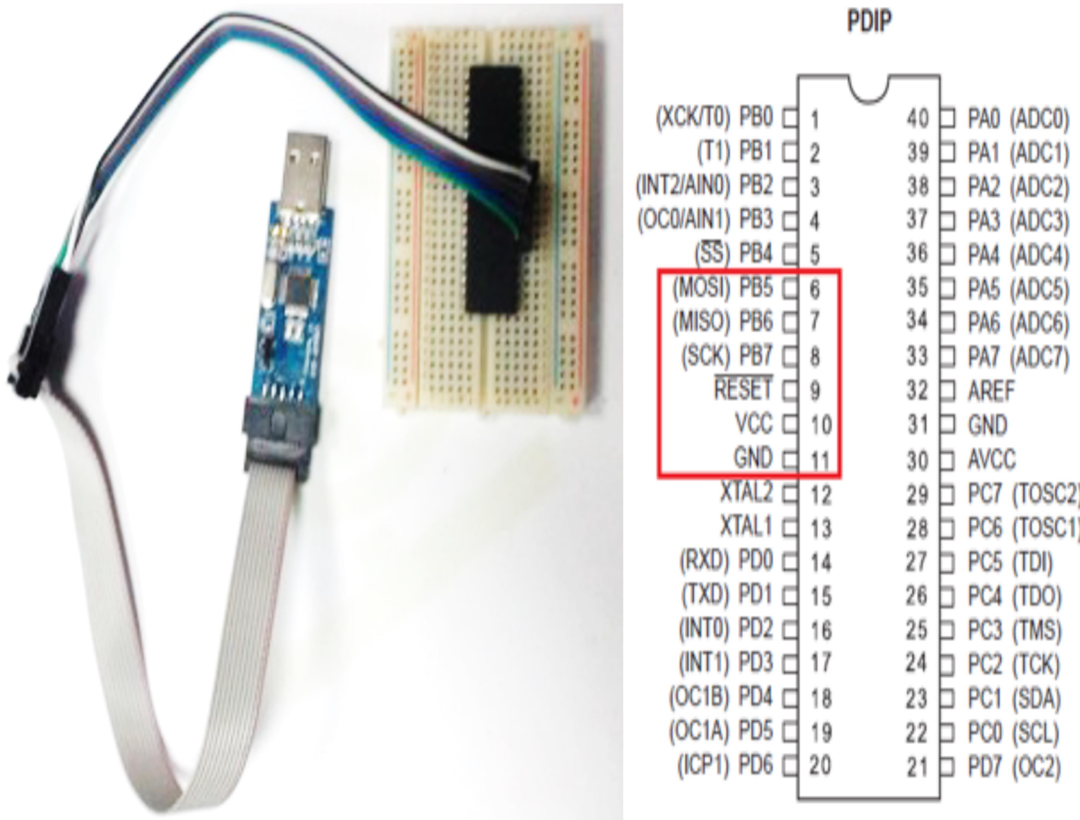


Fig 3.15 Connection between the USBASP and ATmega32A on breadboard [8]

3.3.2 Microcontroller ATmega32A

The Atmel ATmega32A is a low-power CMOS 8-bit high performance microcontroller based on the AVR enhanced RISC (Reduced Instruction Set Computer) architecture. By executing effective directions in a single clock cycle, the ATmega32A accomplishes throughputs near 1MIPS for every MHz frequency. This enables the system programmer to optimize the device for power consumption versus processing speed. [12]

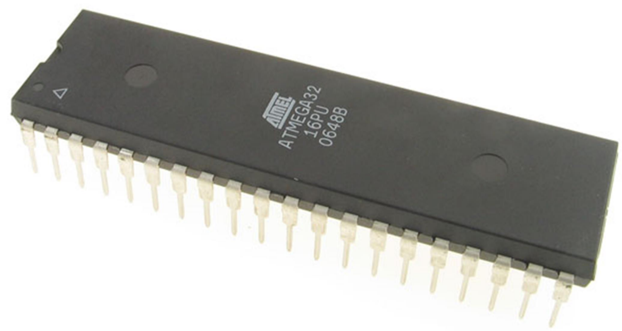


Fig 3.16: Microcontroller (ATmega32A) [13]

3.3.2.1 Features of ATmega32A [12]

- Advanced RISC Architecture
    - 131 Powerful Instructions - Most Single-clock Cycle Execution
    - 32 x 8 General Purpose Working Registers
    - Fully Static Operation
    - Up to 16MIPS Throughput at 16MHz
    - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
    - 32Kbytes of In-System Self-programmable Flash program memory
    - 1024Bytes EEPROM
    - 2Kbytes Internal SRAM
    - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
    - Data retention: 20 years at 85°C/100 years at 25°C(1)
- JTAG (IEEE std. 1149.1 Compliant) Interface
    - Boundary-scan Capabilities According to the JTAG Standard
    - Extensive On-chip Debug Support
    - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
    - Two 8-bit Timer/Counters with Separate Pre-scalars and Compare Modes
    - One 16-bit Timer/Counter with Separate Pre-scalar, Compare Mode, and Capture Mode
    - Real Time Counter with Separate Oscillator
    - Four PWM Channels
    - 8-channel, 10-bit ADC
        - 8 Single-ended Channels
        - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
    - Byte-oriented Two-wire Serial Interface
    - Programmable Serial USART
    - Master/Slave SPI Serial Interface

    - Programmable Watchdog Timer with Separate On-chip Oscillator
    - On-chip Analog Comparator
- Special Microcontroller Features
    - Power-on Reset and Programmable Brown-out Detection
    - Internal Calibrated RC Oscillator
    - External and Internal Interrupt Sources
    - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
    - 32 Programmable I/O Lines
    - 40-pin PDIP
- Operating Voltages
    - 2.7V - 5.5V
- Speed Grades
    - 0 - 16MHz
- Power Consumption at 1MHz, 3V, 25°C
    - Active: 0.6mA
    - Idle Mode: 0.2mA
    - Power-down Mode: < 1µA

3.3.2.2 Pin configuration of ATmega32A

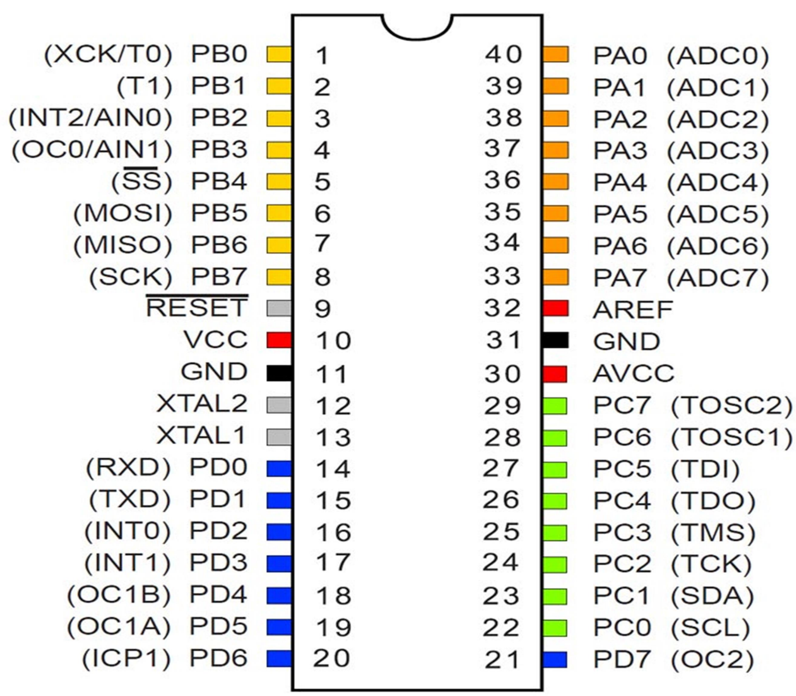


Fig 3.17: Pin configuration diagram [14]

The fig 3.17 shows all the pin configurations of the microcontroller along with the functions of each pin.

PORTA (PA7:PA0) serves the analog inputs to the A/D Converter. It can be also used as an 8-bit bi-directional I/O port, if A/D Converter is not used. In addition, there are also PORTB (PB7:PB0), PORTC (PC7:PC0), PORTD (PD7:PD0) which are also 8-bit bi-directional I/O port with internal pull-up resistors. XTAL1 pin refers the input to the inverting oscillation amplifier and input to the internal clock operating circuit. XTAL2 is the output from the inverting oscillation amplifier. AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. AREF is the analog reference pin for the A/D Converter.

3.3.3 LCD Display (20X4)

The LCD (Liquid Crystal Display) used for our project can show 20 characters in each of its 4 rows. The characters displayed are white in color and lays on a vivid blue background and backlight. This component has proved very useful for our microcontroller based project where the LCD was used to display the input and output voltages along with the percentage of gain. Only two rows out of the four available rows of the LCD were used in our project.

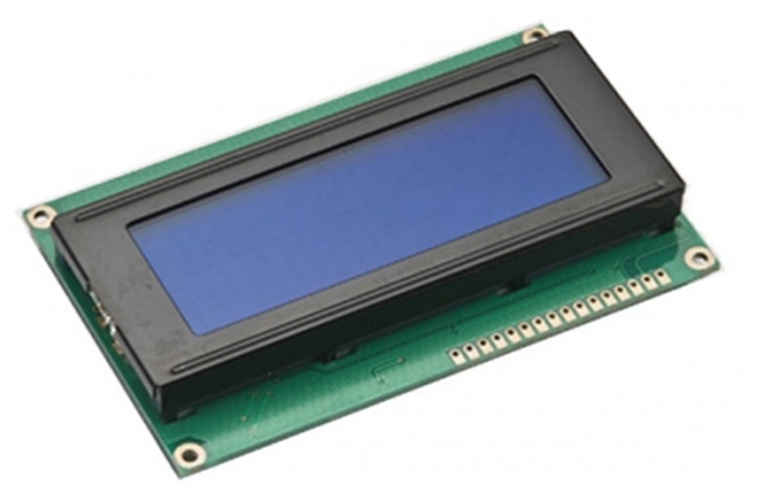


Fig 3.18: LCD display (20X4) [15]

3.3.3.1 Features of LCD display [16]

- Wide viewing angle and high contrast
- Industry standard HD44780 equivalent LCD built-in controller
- +5V DC LED backlight
- Does not need separate power supply for backlight
- Supported 4 or 8 bit parallel interface
- Display 4-line X 20-character
- Operate with 5V DC

3.3.3.2 Pin configuration

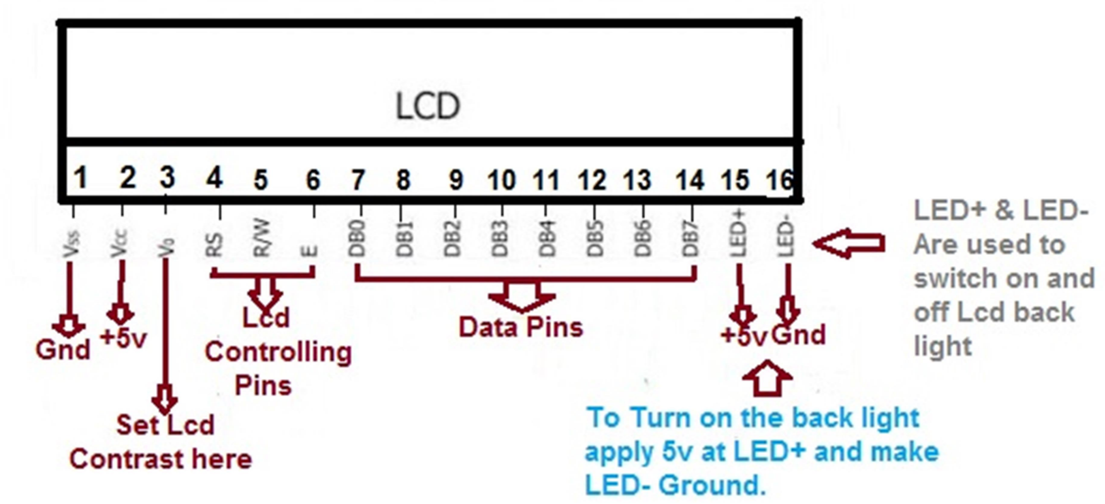


Fig 3.19: Pin configuration for LCD (20X4) [17]

<u>Functions of each pin:</u> [17]

- **DB0-DB7**
  - These are eight bit bi-directional data pins. We send our data and command to the LCD using these lines
- **R/W(Read-Write)**
  - This pin selects if we want to read or write some instruction (Data-Command) to LCD.
    - If R/W=0 Write operation is selected.
    - If R/W=1 Read operation is selected.

- **En (Enable)**
  - This pin is a push to data. It generally remains low but when data is supplied to 8 bit data pins, a high to low pulse must be applied for a few milliseconds to this pin in order for the LCD to latch the data present at the data pins
- **V_{ss} & V_{dd}**
  - V_{ss} pin is grounded and
  - +5V is applied to V_{dd} in order to switch on the LCD.
- **LED+ & LED-**
  - To display the back light of the LCD +5V is supplied to LED+ and LED- is grounded.
- **V_o**
  - V_o sets the contrast for the LCD.
  - Attaching a variable resistor or a potentiometer to this pin and setting the exact contrast makes the text visible to naked eye.
- **RS (Register Select)**
  - This pin selects the LCD register. Each character in LCD has two registers (Data and Command).
    - **Data Register**
      To display data on LCD we have to select data register. To make the data appear on the LCD we have to make the En (Enable) pin high.
      - If RS=1 Data Register is selected.
    - **Command Register**
      Before using LCD we have to set some parameters for it. For example, the font size of data that is going to appear on the LCD.
      - If RS=0 Command register is selected.

LCD accepts two types of signals, one is data, and another is control. These signals are recognized by the LCD module from status of the RS pin. Data can be also read from the LCD display, by pulling the R/W pin high. As soon as the E pin is pulsed, LCD display reads data at the falling edge of the pulse and executes it, and same situation occurs for the case of transmission.

LCD display takes a time of 39-43µs to place a character or execute a command. Except for clearing display and to seek cursor to home position it takes 1.53ms to 1.64ms. Any attempt to send any data before this interval may lead to failure to read data or execution of the current data in some devices. Some devices compensate the speed by storing the incoming data to some temporary registers.

LCD displays have two RAMs, naming DDRAM and CGRAM. DDRAM registers in which position which character in the ASCII chart would be displayed. Each byte of DDRAM represents each unique position on the LCD display. The LCD controller reads the information from the DDRAM and displays it on the LCD screen. CGRAM allows user to define their custom characters. For that purpose, address space for first 16 ASCII characters are reserved for users. After CGRAM has been setup to display characters, user can easily display their custom characters on the LCD screen. [11]

3.4 Conclusion

This chapter covered the software and hardware aspects of our project and discussed the link between these two components. The code for the microcontroller based system is written in AVR Studio 5.0 while the USBASP AVR Programmer connects with the computers USB port in order to burn the hex file of the code in to the microcontroller using the Extreme Burner AVR software. In addition, the LCD is used to display the input and the desired output of the microcontroller. Gaining and understanding of these components is essential for us, so that we are able to move forward with our aim of replacing the existing analog circuitry with our digital circuit.

The next chapter of this paper will comprise of the integration of the torque sensor pedal and its module in to the system.

CHAPTER 4

Integration of the torque sensor into the System

4.1 Introduction

This chapter describes the overall features of torque sensor, its working principle and all aspects of integrating it into the system. This includes mechanical fitting and managing power source of the sensor by both the analog and the digital circuit.

4.2 Introduction to the Torque Sensor

A torque sensor or torque transducer is a device for measuring and recording the torque on a rotating system, such as an engine, crankshaft, transmission, rotor, gearbox, a bicycle crank or cap torque tester [19]. It is a transducer that converts torque based mechanical input into an electrical output signal. There are two types of torque sensors, a reaction that measures static torque and rotary that measures dynamic torque [20]. The fundamental working principle of a torque sensor is to take input from a DC voltage source and generate an output voltage corresponding to the torque applied on a specific crank or shaft. Within the operating region of the sensor, the output voltage is linear with the torque applied.

Fig 4.1: Complete set-view of torque sensor and other components [18]

4.3 Types of Torque Transducers and Torque Sensors [21]

- Brushless Rotary Torque Transducers
- Flange Torque Sensors
- Shaft Torque Sensors
- Multi-Axis Torque and Axial Force Transducers
- High Capacity Torque Transducers (>5000Nm)
- High Speed Rotary Torque Sensors (>55000rpm)
- Low Capacity Torque Transducers (<1N)
- Rotary Slip Ring Torque Sensors
- Miniature Torque Transducers
- Square Drive Torque Sensors
- Static / Reaction Torque Transducers
- Wireless Radio Telemetry Rotary Torque Transducers

4.4 Applications [21]

- Automotive & Motorsport
- Aircraft Component Testing & Development
- Engine Test Stands
- Marine
- Production Process Monitoring
- Pump Development
- Steel Manufacturing
- Torque & Power Measurement on Drive Shafts
- Torque Wrench & Tool Testing / Calibration
- Wind Turbine Development

The torque sensor setup for integration in to the system was purchased from Suzhou Victory Sincerity Technology Company Ltd (http://www.jc-ebike.com) which is located in Suzhou, China. They undertake to develop, design and produce the components of e-bikes, mainly torque intelligent sensor system and relevant parts. Furthermore, the company has a group of experts proficient at designing and developing various e-bikes. The company has developed torque intelligent sensor system which conforms to the European standard ---EN 15194, Japanese JIS standard [22]. Torque Sensor is their national patent product. To use with bicycles they have integrated the sensor technology inside the pedal-system. In our system, the sensor and module were the only components used.

4.5 Features of Torque Sensor [22]

- Brush/Brushless motor controllers are applicable for it
- The hardware may be installed like a normal chain wheel crank
- The electrical system is sensor/sensor-less motor type
- Main body parts are made of aluminum alloy
- Provides instant response while pressure is applied on pedal and pedaling is stopped or pressure on pedal is reduced
- Data collection per crank rotation from 18 to 96 times
- Magnet ring integrated with multi-pole improving greatly the precision of signal sampling
- The system is fully sealed against ingress of water and dirt

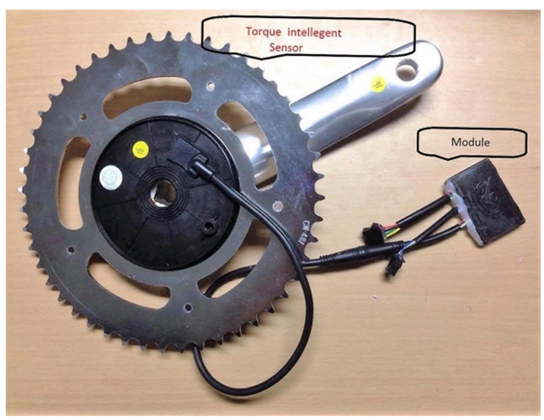


Fig 4.2: Torque sensor and module

4.6 Technical Parameter Data [22]

- Vcc = 5.15 V (+/- 0.15V)
- Output, linear, zero-start, 0.5~4.5V
- Output torque >15N-m
- Delay time < 50ms

4.7 Electrical Connections

The electrical connection diagram for the entire setup was provided by the manufacturing company. This includes the brushless-controller, torque adjustor, chain-wheels, module etc. However, we will be implementing only the torque sensor and the corresponding signal-producing module in the system.

Fig 4.3: The torque-sensor pedal and module attached to the wheelchair

The sensor was built in such a way that it could be fixed in any bicycle. However, assembling it in the tri-wheeler required a few mechanical modifications. The above figure shows the torque sensor pedal and its module when implemented and installed in the electrically assisted wheelchair developed by CARC. This installation was done following the provided connection diagram provided by the manufacturing companies.

According to the figure below, there are separate mechanisms for the torque sensor and module to get input voltages from the controller connections. But in the system, a different controller will be incorporated with the sensor. So, the independent operation connection diagram was extracted from the main diagram.

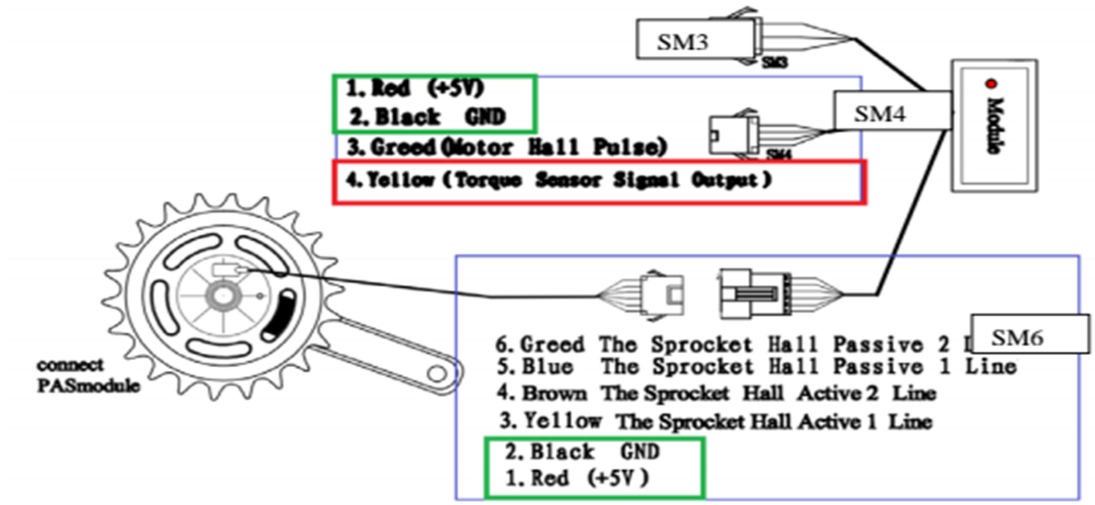


Fig 4.4: Electrical connection diagram for independent operation of the sensor and module [22]

Figure 4.4 indicates that the Red and Black wires from SM6 (marked as +5V and GND) are the input biasing terminals for the sensor. For the module, the input voltages are marked in SM4. The Yellow wire from SM4 is the processed output from the torque sensor which will be used for motor control. This wire gives the output voltage with respect to GND according to applied torque in the pedal-crank of the sensor. So, this is the output which is supposed to be fed to the external control circuit.

4.8 Power Management

For implementation of the system, it was necessary that all power is sourced from the main wheelchair battery which is three 12V batteries in series. The external circuit design and the torque sensor specification setup require that only a +5V source in order to be activated. As a result, to get a fixed output of +5V the LM7805 was used.

The power-source management used in the earlier system that utilized an analogous circuit is illustrated in figure 4.5.

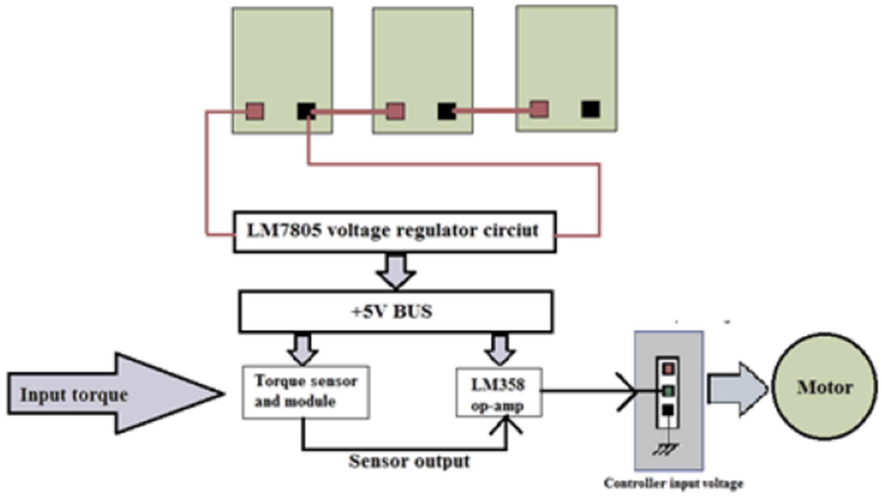


Fig 4.5: Overview of the power management and signal flow of the analog circuit [23]

The system has three 12V batteries connected in series providing 36V. Only 12V is required as input to the voltage regulator circuit. The voltage regulator circuit will give +5V which will be fed to the amplifier circuit as biasing and also to power up torque module. When pressure is applied on the pedal, there will be a certain torque for which the torque sensor and module will provide a certain output voltage to the operational amplifier circuit. The amplifier will output a voltage corresponding to the voltage provided by the sensor and module maintaining the ease while pedaling. This voltage is fed to the motor C.U and eventually to the motor.

The modified and improved, power-source management to be implemented uses a power module along with a microcontroller to control the input from the torque sensor. The block diagram for the present system is shown below,

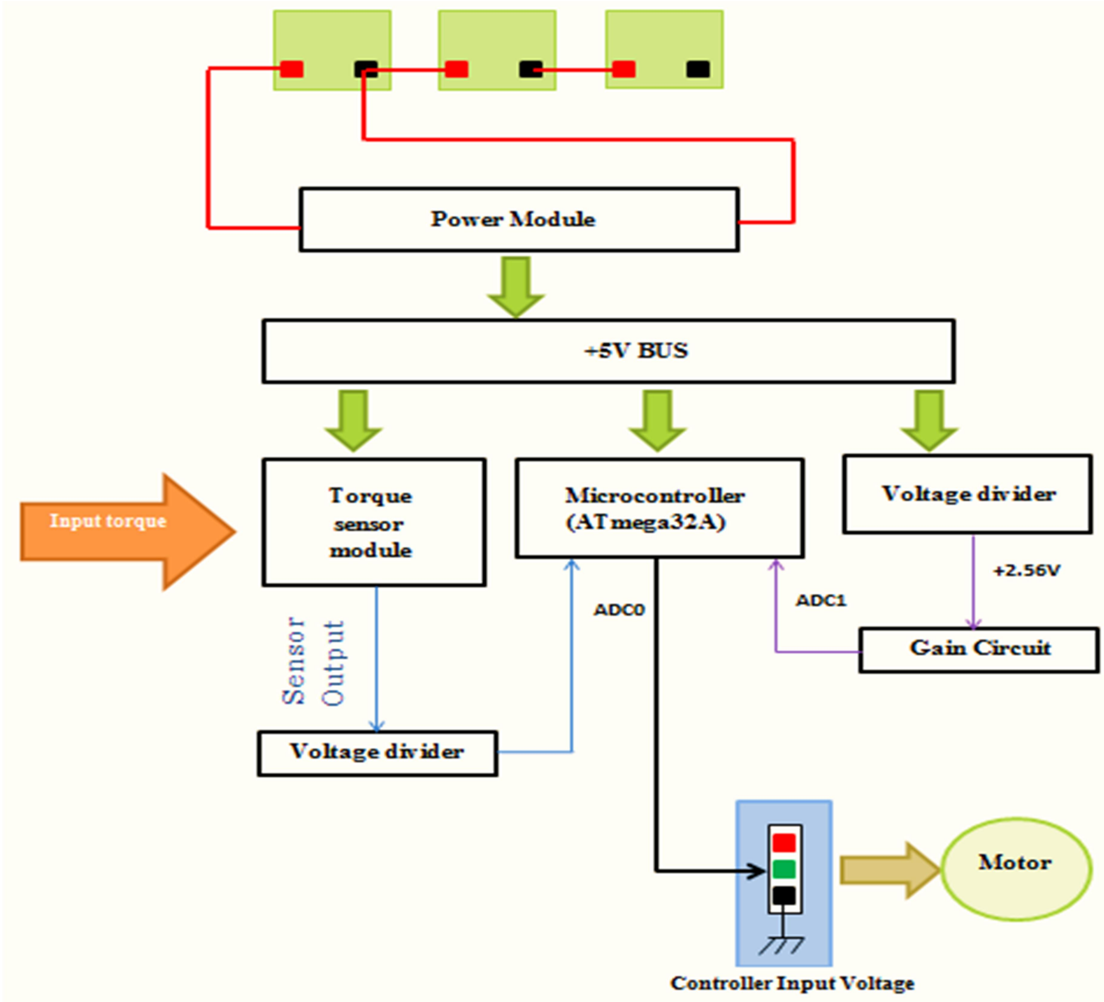


Fig 4.6: Complete overview of the power management and signal flow

Here, same as the previous circuit our input to the system is 12V and this is taken from the first cell of the three 12V lead acid battery connected in series to produce 36V for the motor. This input voltage is fed to a power module to obtain an output voltage of 5V. **The power module consists of a crowbar circuit protecting the voltage regulator LM7805 from overvoltage. This circuit not only provides overvoltage protection in case of power supply malfunction but it also has over current, reverse polarity, and regulator IC protection.** This circuit is particularly important since both the microcontroller ATmega32A and the torque sensor module are sensitive devices which require a well- regulated power supply and a constant +5V for biasing. If by any chance the voltage regulator becomes faulty and supplies more than 5V (for instance 6V) it would damage the main circuit components (microcontroller and torque sensor module). Full description about the power module is given in chapter 7.

Furthermore, 5V is also provided to a voltage divider circuit to obtain a voltage of 2.56 V, goes to the gain circuit. The microcontroller receives two ADC inputs: ADC1, the gain, and ADC0, the output voltage from the torque sensor module. Based on these two inputs, the microcontroller sends a Pulse Width Modulation (PWM) signal to the controller unit of the motor which then drives the motor according to our stated logic.

Benefits of the new design for power management as opposed to the previous design,

1) The new circuit design incorporates the power module with a protection circuit that protects the entire circuit from overvoltage, overcurrent, reverse current flow and protection for the LM7805.
2) This power module ensures that the LM7805 does not provide more than 5V to the main circuit, which would burn the torque sensor module and the microcontroller.
3) This is a major improvement from just using the LM7805 in the previous design where malfunction of the voltage regulator would occur, causing damage to the analogous circuit.
4) The microcontroller being a precision device controls the motor more efficiently according to the algorithm burned in to it.
5) Microcontroller is a high performance and low power appliance as a result of which the overall power consumption is reduced due to the replacement of the analog circuit with the microcontroller based circuit. This will be proved in the section where we perform the field tests and analyze the data we will obtain.

4.9 Conclusion

We discussed about the torque sensor in this chapter and learned about the different aspects of the sensor as well as how it has been integrated in to the system and with the microcontroller based circuit. The following chapter will give us an insight in to the existing circuitry which is the analog circuit, how it works and its control algorithm.

CHAPTER 5

Overview of the Analog circuit

5.1 Introduction

This chapter focuses on the existing circuitry that controls the motor and gives a complete overview of its development, implementation and performance. The circuit is composed of an operational amplifier, a voltage regulator and resistances to adjust the gain. The working principle of this circuit is, it takes input from the torque sensor module and outputs a corresponding voltage. This output voltage is fed to the motor controller which then governs the speed of the motor. The following sections describe the developed control algorithm and key observations made that lead to the making of the current circuit design. In addition, the rate of power consumption from the battery has been mentioned and compared with and without the use of the torque sensor pedal.

5.2 The Control Algorithm [24]

In the logic that has been developed, the motor driver system should provide assistance from the motor only when it is needed by the user; through measurement of the input torque on the pedal. The system has to control the speed of the BLDC motor, fitted in the wheelchair, with an external input signal from the torque sensor. In order to achieve this, there must be a minimum amount of torque (threshold torque) which when applied on the pedals, should run the motor.

If the output voltage from the torque sensor module is V_s, and the output voltage of the sensor module for the Threshold Torque is V_{th}, then as soon as V_s exceeds V_{th}, the motor speed should increase proportionally with respect to the voltage that is excess of V_{th}. Simultaneously it must be ensured that overuse of the motor is restricted, which means the motor cannot be allowed to run at very high speeds. Therefore, to attain these parameters Vs can be the controlling factor for the motor speed and a system will be developed that limits V_s to a maximum level even when the input torque corresponds to a higher voltage.

Considering, that V_{in} is supplied to CU as the input signal that controls the motor speed and $V_{in} = K \times V_s$; where K is a constant indicating the level of ease for the system, and V_{max} is a pre-set value of ($K \times V_s$) that corresponds to a maximum limit of V_{in}, and V_{start} is the voltage at which the motor turns on, then the algorithm is developed as,

Get Vs
$V_{in} = K \times V_s$
% Amplifier designed so that maximum output is V_{max}
If $V_{in} \geq K \times V_{th}$
Motor speed $= M \times (V_{in} - V_{start})$; *% M is constant*
Else, Motor speed = 0
Go to: *Get* V_s

The algorithm was designed and simulated with a voltage source (assuming the voltage was coming from a sensor) before finalizing the design for the system.

5.3 Key Observations of the system and Implementation of the results [24]

The Controller Unit (CU) has an input terminal that comes from the throttle position sensor (TPS). This provides voltage signal which controls the speed of the motor, almost linearly with increasing voltage from where we can access its predefined threshold (V_{start}). The key idea is to provide this voltage, externally, from a torque sensor which outputs voltage signal proportionately with increasing torque. However, it is important to note that the motor should not start at literally any torque input; a minimum amount of torque (threshold) that must be applied in order to get motor assistance. This means the motor speed should be proportional to the output voltage of Torque Sensor (TS) that is access to the minimum voltage. **So, the built-in threshold should correspond to the desired threshold torque output.**

The main idea, as has been described before, was that the motor will assist the puller according to the torque he applies on the pedal. There will be two predefined set-points:

1) A minimum torque at which puller starts getting assistance from motor, *A*N-m.
2) A maximum torque after which the puller will not get any assistance.

Assuming the corresponding voltage for *A*N-m be **X** Volts.

$$\mathbf{V_{start} = K \times X}$$

Where, K is a constant gain factor, which will be defined by the circuit designer as a determinant of *'level of ease'*. If, after amplification, the torque represents V_{start}, the motor will start running, and the speed will increase with increasing input torque, as torque corresponds to voltage. The gain should be such that the minimum torque we set corresponds to V_{start}.

On the other hand, the upper cut-off voltage is set at a value according to motor-stability limit. It is set at the value of 3.6V, as the system tends to become unstable beyond this voltage as the motor rotates with very high speed.

According to the conditions that have been pre-set, the motor will start running and increase in speed linearly as soon as the controller will get more than or equal to V_{start} at the throttle terminals (controller input voltage); irrespective of where it is getting that voltage from. Consequently, **the sensor voltage needs to be manipulated in such a way that for the controller, it will seem that the signal is coming directly from the throttle, but in reality it will be processing the voltage signal from the torque sensor.**

The external hardware to be designed has to have two distinct parts. The first part is an amplifier which transforms the minimum torque to V_{start}. The value of the amplifier gain will depend on sensor characteristics and the value of minimum torque that has been set. The output of the amplifier will be connected to the controller input terminals. However, the amplifier output will always be compared to the cut-off voltage (the voltage at which the motor is to be stopped). The comparator will output a 'high' whenever the voltage exceeds the cut-off (in this case, 3.6V).

A way to limit the speed of the motor is by limiting the maximum output voltage of the amplifier to a reasonable value (say 3.6V) using a reduced biasing voltage. For example, if we use LM358 Operational Amplifier IC, and provide a biasing voltage of +5V, the maximum output voltage will be 3.6V. There are several advantages of using LM358 Operational Amplifier. They are:

1) Negative biasing voltage is not required, reduces power supply complications.
2) Large operating range; can be used with the same biasing voltage as the torque sensor (+5V) and hence, only one power bus can be used.
3) Reduces Complicacy of circuits to a great extent.

5.4 Design Proposal [24]

In this design, a continuous process takes place where the output of the amplifier goes directly to the controller input terminals (between green and black terminals) but is limited to a certain value by the use of a controlled biasing voltage for the amplifier.

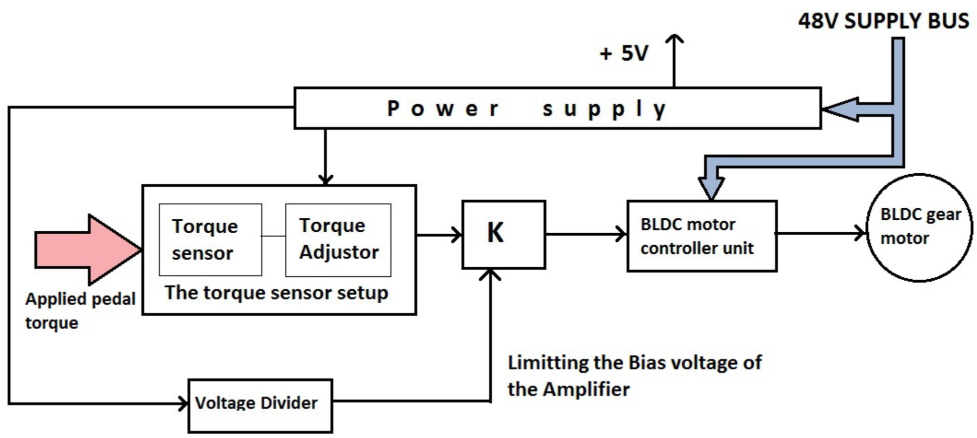


Fig 5.1: Block diagram of the design proposal [24]

In fig 5.1, the amplifier block ‘K’ is not powered from a secondary 5V power supply (or any step-down value), but from a lower divided voltage using a voltage divider. The value of the reduced voltage will depend on the particular op-amp used and its maximum output voltage in 5V biasing. For example, if we want to limit the voltage to 3.0 V, the biasing voltage will be set accordingly. For an LM358 Operational Amplifier IC, a 5V biasing leads to a maximum voltage of 3.5V, and its operating range is 3V-32V.

5.5 Circuit Design I [24]

5.5.1. Introduction

The basic concept of this design is to limit the output of an op-amp by limiting its biasing voltage. For a specific level of biasing voltage, the output of an op-amp is maximized to a value depending on that voltage. In this design, LM358 operational amplifier is used with a biasing of +5V to limit the output to 3.5V and this output will be directly fed in the CU terminals.

5.5.2. The Circuit-Diagram and Explanation

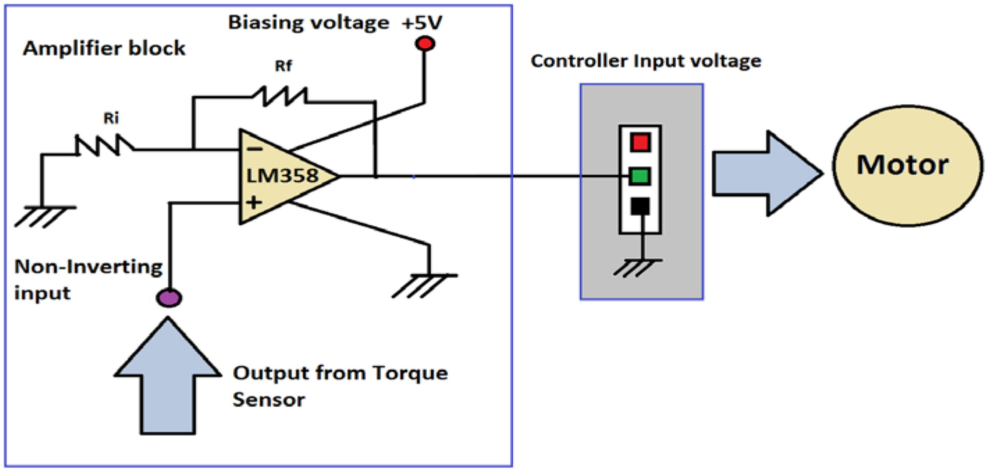


Fig 5.2: Circuit design I of the analog circuit [24]

Fig. 5.2 uses LM358 op-amp as the key device. The LM358 consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. [17]. The LM7805 is a voltage regulator which gives an output of +5 V when a voltage more than 7.5 V is provided as input. The output is always fixed and it is +5 V

With a supply voltage of +5V, the amplifier's maximum output voltage is 3.5V. This characteristic will help the system to maintain a maximum tolerable voltage input to the CU without interrupting the whole system by motor switching or sudden-stop-runs. For example, if the gain is set to 2, an input of 1.5V will give 3V; but an input of 2V will not give 4V as output, but will show 3.5V. We will take the advantage of this in the circuit design.

5.5.3. Simulation and Results

For simulation, the gain was set to 1; the biasing voltage was set to 5V. The system behavior with these settings can be visualized below in fig 5.3. For simulation, voltage signals were provided from a voltage source following the pattern of 0 V to 4.0 V in constant intervals; then 4.0V-to 0V.

As illustrated in fig 5.3, it is observed that, increasing the torque (voltage) causes the motor assistance to increase proportionately. After the cut-off, the motor continues to run in the maximum possible voltage that is 3.5 here. Again, when the voltage starts to drop, the speed starts to drop accordingly with the regular fashion.

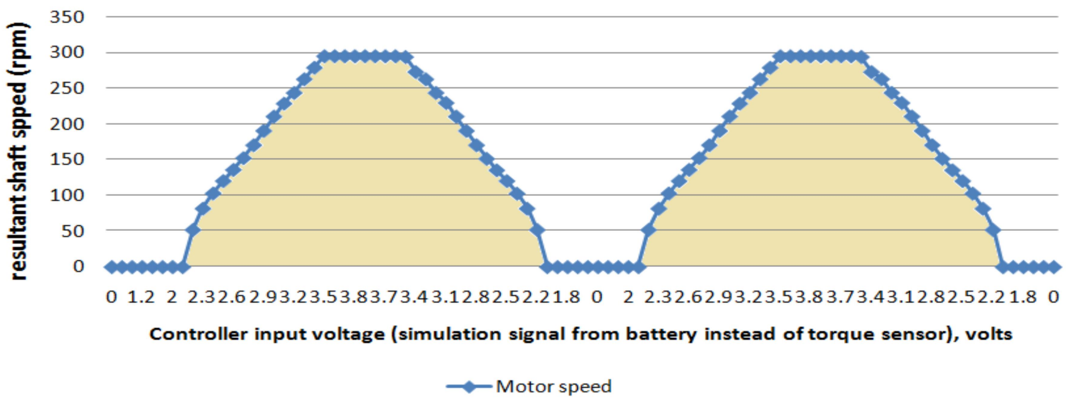


Fig 5.3: Simulation results for the circuit design [24]

Some important benefits of this circuit design are as follows: [24]

1) The circuitry is independent of hand-clutch or power-keys since the motor's use is limited by limiting the biasing voltage of the op-amp.
2) The circuit will consist of only voltage regulator circuit and op-amp IC, reducing complicacy.
3) Torque sensor and all other devices can be biased using a single supply of 5V provided by LM7805.
4) Negative biasing is not required for LM358.
5) Continuity of assistance is maintained in the hybrid vehicle. Sudden drop of rpm does not occur and the motor assists the user whenever required, according to requirement, within its limit. When more assistance is required, it will not stop assisting, but will continue to assist with its maximum limit.
6) No sudden thrust takes place during riding conditions; the ride will be comfortable.
7) Optimization of manual power and electric power can be achieved.
8) Consistent high-assistance can be achieved when required. For example, **while going uphill**, the input torque will frequently cross maximum level, at that time, if the motor stops assisting, it might be fatal. Therefore, even after crossing the maximum torque level, the motor will not "stop responding", but will assist with its maximum power corresponding to the cut-off voltage (3.5V).
9) No frequent motor-switching takes place.

5.6 Circuit Design II [23]

5.6.1 Introduction

The key concept of this design is the use of a voltage divider circuit. Here, the voltage coming from the torque sensor and module is not fed directly to the op-amp, but instead the input signal passes through a voltage divider to reduce the incoming voltage to 0.6 of the input voltage. This reduced voltage is then fed to the op-amp. In this way even if the incoming voltage is high, it will be reduced to a certain value and eventually limit the use of motor.

5.6.2 The Circuit-Diagram and Explanation

The circuit design I was modified because it was noticed that when the vehicle was utilized practically, the voltage generated was much higher around 3-4 V when the puller applied pressure on the torque sensor pedal. Therefore, to minimize this voltage, a voltage divider is used in such a way that the output voltage becomes 0.6 of the input voltage. That is,

$$\mathbf{V_{out} = 0.6 * V_{in}}$$

The circuit diagram is shown in fig 4.4. Here, a voltage regulator circuit has been used in order to limit the torque sensor input voltage.

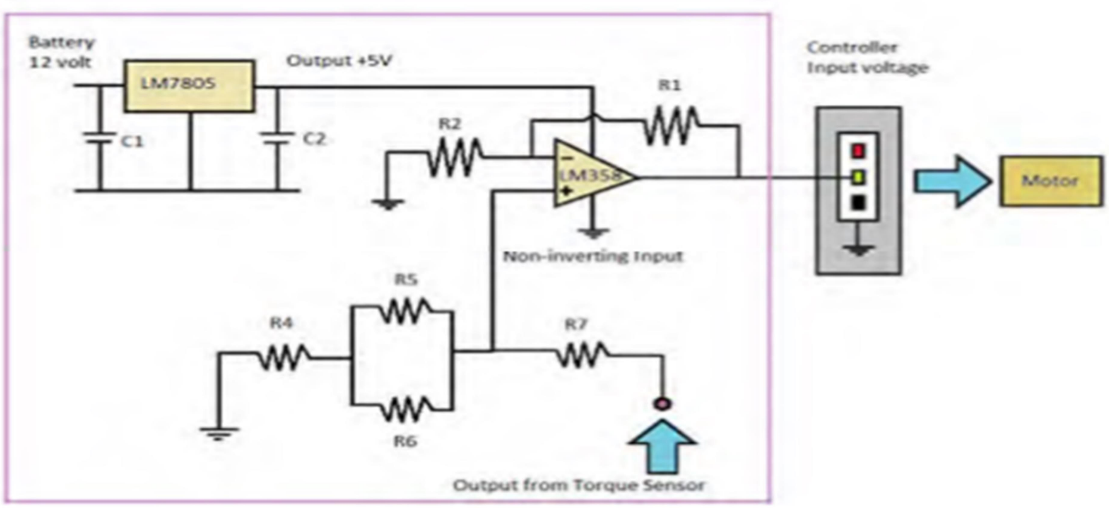


Fig. 5.4 Circuit design II of the analog circuit [23]

Together with retaining all the advantages of the previous circuit design I the modified circuit design II adds another benefit. In case of high input voltage coming from the torque sensor and module, it is first reduced to 0.6 of the input voltage and then fed to the operational amplifier. This way motor use can also be limited. [23]

5.8 Conclusion

The circuit design II is the existing analog circuitry that is being currently implemented in the wheelchair. In the upcoming chapters, we will focus on replacing this analog circuit with a digitalized circuit that is controlled by the output of a microcontroller (ATmega32A). But then again, the digital circuit to be designed will follow the logic used so far in the analog circuit with the motive to bring about much more improvements in accuracy and power consumption of the battery.

The following chapter gives details about various functions of the microcontroller which are essential to understand for developing the algorithm. The chapter also includes the upgraded control algorithm designed specifically for use with the microcontroller.

CHAPTER 6

Development of the control algorithm for the microcontroller based circuit

6.1 Introduction

This chapter describes in detail the upgraded control algorithm we have developed for implementation with the microcontroller (ATmega32A). This algorithm is based on the earlier version of the system discussed in the previous chapter but modifications to the procedure have been added to make the process more reliable and smooth for the digital circuit.

6.2 The Control Algorithm

The algorithm that has been developed can be defined and divided into three steps, as following:

1) ADC (analog to digital converter) conversion
2) Output calculation
3) Pulse width generation

6.2.1 ADC conversion in microcontroller (ATmega32A)

The output from the torque sensor and its module is a variable and continuous signal; fundamentally it is an analog signal. However, the microcontroller will only recognize this signal if it has been converted to a digital signal in the form of binary numbers. This type of conversion is known as analog to digital conversion and is carried out by the analog to digital converter (ADC). ADC is one such hardware which measures analog signals and produces a digital equivalent of the same signal.

AVR ATmega32A microcontroller can convert the analog voltage signal to a digital signal (integer value) since it has inbuilt 8 ADC pins in PORTA (PA0 to PA7). AVR can convert the analog signal into a 10-bit number of ranging from 0 to 1023. There are three ADC registers in ATmega32A microcontroller,

- ADC multiplexer selection register (ADMUX) – selects the reference and the input channel

- ADC control and status register A (ADCSRA) – has the status of ADC and is also used to control it
- ADC data register (ADCL and ADCH) – ADCL and ADCH stores the final output in register A

It needs to be mentioned that there is also an Analog Reference (AREF) Voltage, which is considered to be equivalent to 1023 if 5V is provided and for any voltage value less than this AREF will have an integer number less than 1023. The input range is 0-AREF and digital output is 0-1023. Here 0V will be equal to 0, and AREF/2 will be equal to 512 and so on. [25]

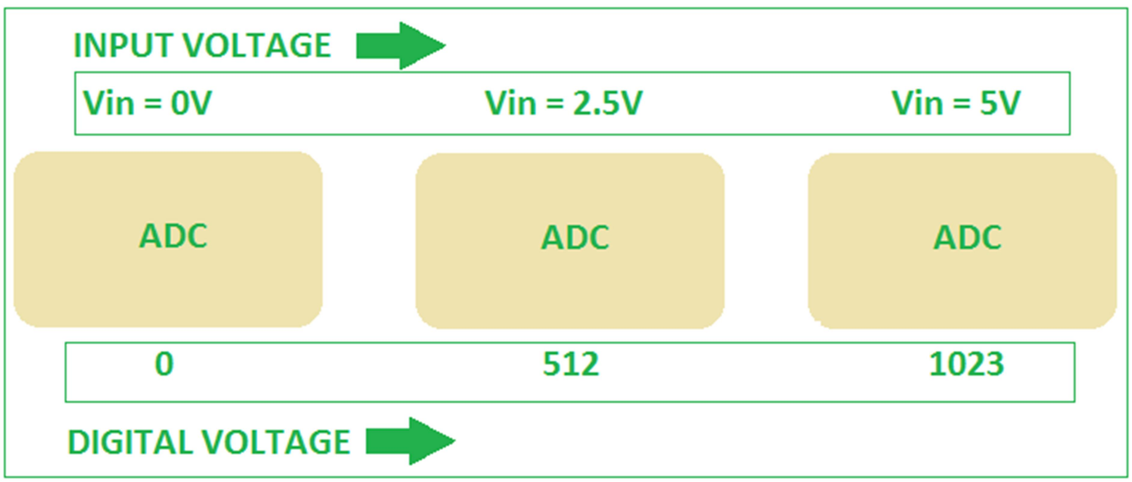


Fig 6.1: Sample ADC conversion [25]

For ADC conversion to take place a clock pulse is required and for this the system clock is divided by a number (2, 4, 16, 32, 64 or128) to get lesser frequency since fine-tuning of the ADC conversion requires a frequency between 50 kHz to 200 kHz. [25] This number is called the pre-scalar division factor.

For our algorithm, we set the pre-scalar division factor to **16**. Since, the microcontroller's internal clock frequency is 1 MHz division by a factor 16 gives us a frequency of 62.5 kHz.

$$Frequency = \frac{1\ MHz}{16} = 62{,}500\ Hz$$

Therefore, the microcontroller analog to digital conversion mode is set to a frequency of 62.5 kHz which means that the microcontroller ATmega32A will initialize ADC

interrupts 62,500 times per second in order to take an analog input from the torque sensor and the gain circuit respectively.

This analog input to the ATmega32A microcontroller can range from 0V to a maximum of 5V. By using the ADMUX function in the microcontroller, we select ADC0 and ADC1 pins in the microcontroller to take inputs from the torque sensor output and the gain circuit respectively. These inputs are converted into 10 bits of digital data, and stored into the 'A' register of the ATmega32A microcontroller. Moreover, to ensure the range of the input conversion, we have set a reference of 5V in the AREF pin.

6.2.2 Output Calculation

As mentioned before, the same logic is followed here as the analog circuit. The key concept remains the unchanged; the system should provide assistance from the motor only when it is needed by the user by measuring the input torque applied on the torque sensor pedal. Similarly, there will be two predefined set points that will govern the motor controller unit. These set points are,

1. A minimum torque at which the puller starts getting assistance from the motor
2. A maximum torque after which the puller will get constant assistance

The system will have a minimum torque input which when applied at the pedal of the torque sensor will produce a corresponding output voltage from the microcontroller that will start our motor. After this minimum torque input has been exceeded the speed of the motor will increase linearly with the increase of applied torque to the pedals. Similarly, after reaching a maximum torque input the motor will not increase in speed anymore as it will receive a constant output voltage from the microcontroller. At this point the motor will maintain a constant speed unless the applied torque declines or the hand clutch is pressed.

According to our requirements, we need to calculate the output voltage that will be fed into the motor controller unit from the microcontroller. If the torque sensor output voltage is within its operating region the following formula will be applied to calculate the output,

$$\mathbf{V_{out} = (torque\ sensor\ output)*(gain)}$$

The gain represents the ‘level of ease’ for the user, which is variable and determined by the circuit designer. The maximum gain that can be varied in our algorithm is set for 10 i.e. the gain can be adjusted from 0 to 10. (The amount of gain that has been varied by the user will be displayed by the LCD display which will show the gain variation from 0 % to 1000 %.)

In accordance to our design statements, the microcontroller should send an output signal to the C.U only if the torque sensor output exceeds the lower threshold input (corresponding voltage to the minimum torque input). In our case, the lower threshold is set to 0.5V, whose corresponding digital value is 0001100111 in binary form and 103 in decimal form. The C.U will receive output from the microcontroller up until the torque sensor output reaches a certain upper threshold input (corresponding voltage to the maximum torque input) which will ensure a maximum output of 3.6 volts, whose digital value is 1011100001 in binary form and 737 in decimal form.

Hence, this input upper threshold is calculated through following formula:

Input upper threshold = (737/gain)

Even if more torque is applied on the torque sensor pedals beyond this input upper threshold, output will remain same at 737 or 3.6 volts, regardless of any excess voltage provided by the torque sensor. This condition has been established to ensure that the motor is not overused for high torque input as motor becomes unstable and rotates at very high speeds. Additionally, this has proved to be easier for controlling the wheelchair.

6.2.3 Pulse Width Modulation (PWM)

Pulse width modulation (PWM) is a powerful technique for controlling analog circuits with a microcontroller’s digital outputs. PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion. [28] In AVR microcontrollers PWM signals are generated by the TIMER units.

A timer (also called counter) is simply a register that counts upon receiving clock pulses. The timer increments or decrements its count with each clock pulse it receives. A timer is usually specified by the maximum value to which it can count (called **MAX**) beyond which it overflows and resets to zero (**BOTTOM**). Thus, an 8-bit timer/counter can count from 0 to

255 and a 16-bit timer/counter can count from 0 to 65,535. The speed of counting can be controlled by varying the speed of clock input to it. This is done by pre-scaling the clock of the microcontroller. By pre-scaling, we feed a fraction of the CPU clock to the timer. [27]

Timer speed =F_CPU / pre-scale

Where, F_CPU is the AVR CPU clock speed.

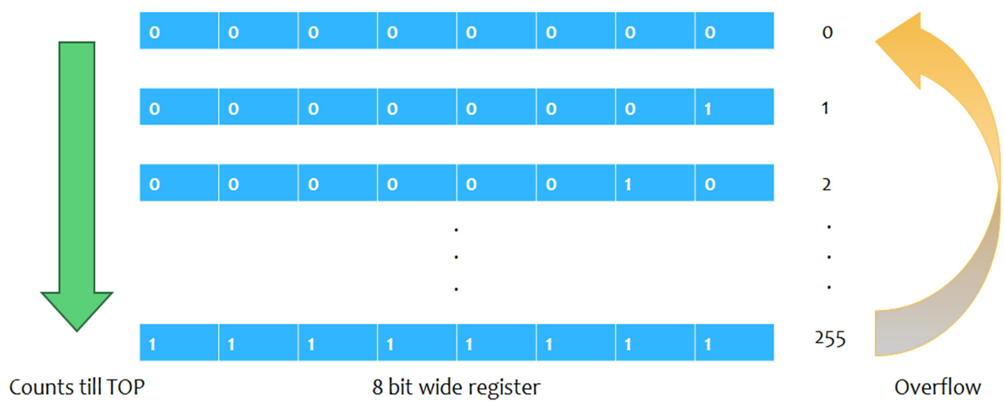


Fig 6.2: 8-bit counter [26]

The timer works independently of the CPU i.e. it runs parallel to the CPU and there is no intervention from the CPU. Therefore, the timer can be used to measure time accurately.

In ATmega32A there are three different kinds of timer,

- TIMER0 – 8-bit timer
- TIMER1 – 16-bit timer
- TIMER2 – 8-bit timer

These timers can be operated in three modes:

- Normal mode (overflow)
- Pulse Width Modulation mode (PWM)
- Clear Timer Compare mode (CTC)

We will operate the timer in PWM mode. In PWM, square waves are generated whose **duty cycle** can be varied. Duty cycle refers to the fraction of the time period of the wave for which the signal is in high state (or simply ON state). For example, for a square wave of period

100ms, if the duty cycle=50%, the signal will be in high state for precisely 50ms and in low state for the next 50ms that make up the period. [29]

$$Duty\ Cycle = \frac{ton}{ton + toff}\ x\ 100\% = \frac{ton}{Period}\ x\ 100\% = \frac{Pulse\ Width}{Period}\ x\ 100\%$$

The microcontroller PWM peripheral depends on the TIMER to provide the PWM signals frequency. The PWM peripheral will use the TIMER counter register (**TCNT**) as a digital step-up /down and continuously compare to the pre-determine duty cycle register (**OCR** – output compare register) value. When **TCNT** equal to **OCR** value the wave generator circuit will set (ON) or reset (OFF) the corresponding microcontroller PWM I/O ports.

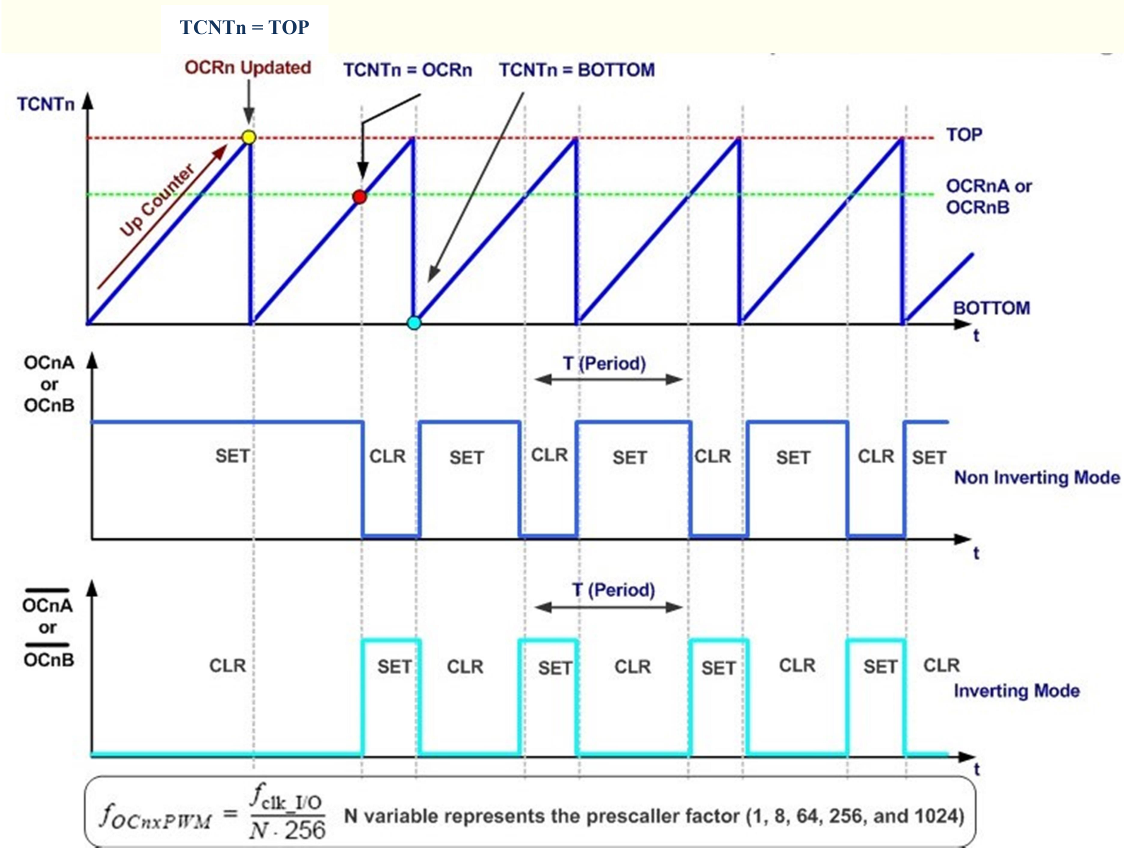
There are also several modes of PWM generation provided by AVR which decides the shape of the square wave generated (the placement of the high section in each cycle). The modes are,

1. Fast PWM
2. Phase correct PWM
3. Phase and Frequency Correct PWM

For our algorithm we will utilise the Fast PWM mode. The AVR fast PWM mode can generate the most high frequency PWM waveform compared to the other two PWM modes (i.e. Phase Correct or Phase and Frequency Correct mode). This PWM mode uses the TIMER counter register **TCNTn**, where n represent the TIMER 0, TIMER1, and TIMER2 respectively, incremental value of which starts from **0x00** (BOTTOM) to 0xFF (8-bit TOP) or **0xFFFF** (16-bit TOP). [30]

When the TIMER counter register reach the output compare register (**OCRnA** or **OCRnB**) value then the output compare bit channel (**OCnA** or **OCnB**) will receive a CLEAR (logical low) signal. When the TIMER counter register value reach the TOP value then it will SET (logical high) the output compare bit channel and the whole process will repeat again from BOTTOM.

There are two modes of operation: the inverted mode and the non- inverted mode. In the non-inverted mode, when the count equals the **compare** value (set in the OCRnA or OCRnB register), the OCnA pin or OCnB is pulled low and the timer continues counting till TOP, resets to zero and repeats the whole cycle again. In inverted mode, OCnA or OCnB is pulled low when counter resets and pulled high when compare match occurs. An example of the PWM generation process is shown below,



$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

Fig 6.3: Fast PWM (non-inverting and inverting mode) [30]

Our algorithm utilizes the inverted mode of operation and the OCR1A register. Here, the OC1A is pulled high when the count equals the compare value set in the OCR1A register. The value that we have set in ICR1A is 3333.33 and when the counter reaches this value the signal will jump up to 5V (creating the start of the pulse). After this the timer resets to zero at which point the OC1A pin is pulled low. All other parameters needed for generation of the PWM is mentioned in the code provided in Appendix C.

In a nutshell, PWM is a way of digitally encoding analog signal levels. Through the use of high-resolution counters, the *duty cycle* of a square wave is modulated to encode a specific analog signal level. The PWM signal is still digital because, at any given instant of time, the full DC supply is either fully on or fully off. [28]

The above descriptions give us a brief idea about PWM generation which we will use to convert our digital output signal from the microcontroller to an analog signal. Once the output is obtained, it is converted in the form of pulses using fast pulse width modulation in the microcontroller. Using inverted ISR vector function, pulses of 3.33ms is created and sent to OC1A pin of the microcontroller, which is then directly feed to the motor controller box. These pulses act similar to the speed control of a DC motor using pulse modulation.

6.3 Flowchart of the Control Circuit

The steps of the algorithm are,

1) Initialize variables: Torque sensor input, gain, gainADC, output, val
2) Take output of torque sensor and store the digital value in the variable "Torque sensor input".
3) Take the output of gain circuit and store the digital value in the variable "gain ADC".
4) Calculate the gain value by: $\frac{digital\ value\ of\ gain\ circuit}{512}$ and corresponding value is stored in the variable "gain"
5) Calculate the upper threshold by: ${}^{737}/_{gain}$
6) Check the '’torque sensor input" value with lower and upper thresholds to determine the operating region.
7) If the torque sensor input is less than lower threshold, then output will be zero.
   If the torque sensor input is beyond the upper threshold, the output will be 737.
   If the torque sensor input is between the lower and upper thresholds, the output will be:

   Output = Torque sensor input * gain

8) Pulse width of maximum 3.33ms will be calculated and stored in variable "val" by

   val = (3333.33 * output)/1023

9) Using Fast wave generation mode, pulses are created in accordance to the variable "val" and sent to motor controller.

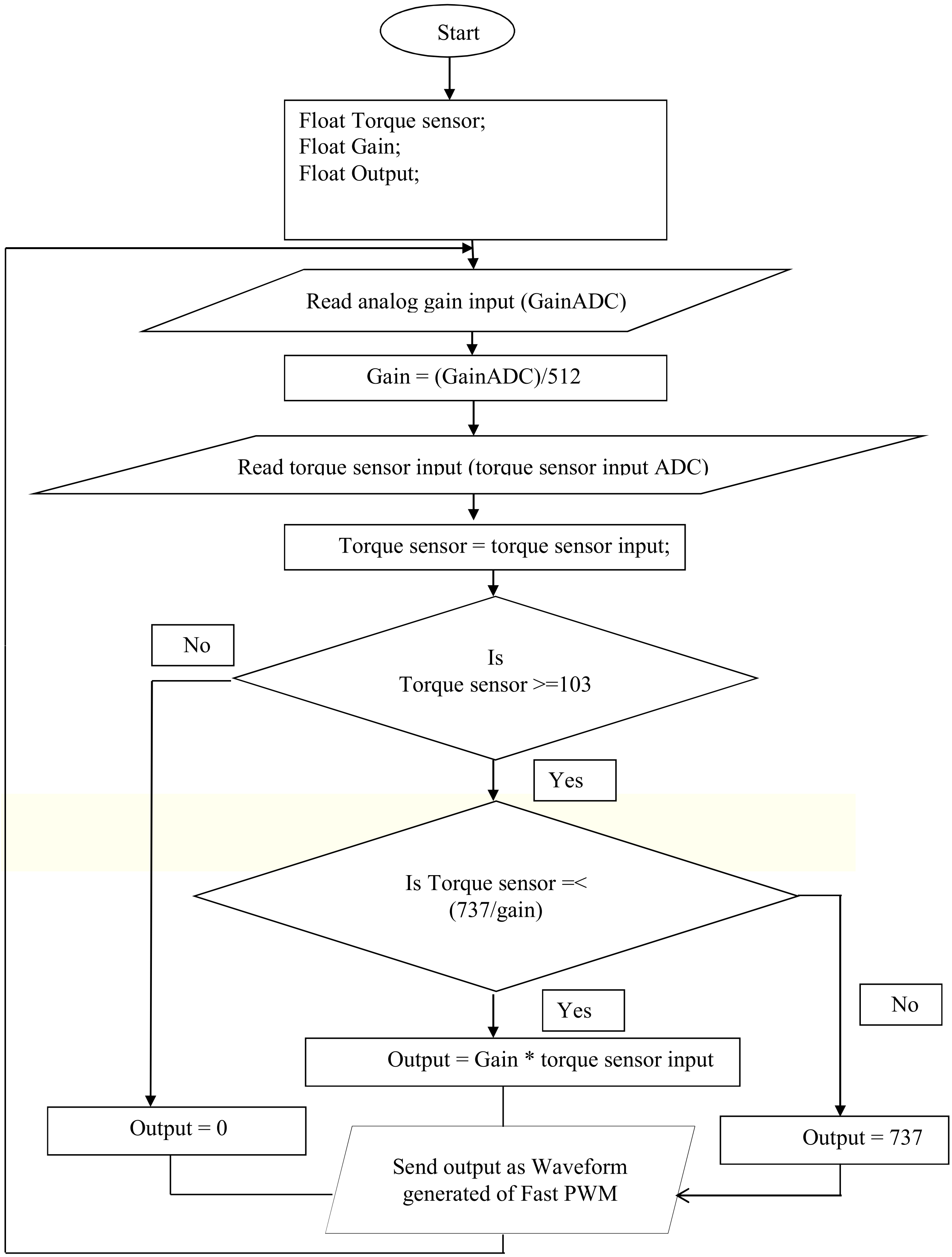


Fig 6.4: Flowchart of our control Algorithm

6.4 Conclusion

The entire chapter discusses the control algorithm that has been established, which will be executed by burning the developed code in to the microcontroller.

The summary of our algorithm for the control circuit is as follows,

1. Read analog torque sensor input and gain input through analog to digital conversion, and feed it to micro controller
2. Calculate the upper threshold for the input received
3. Calculate output using following formula until the output reached to maximum i.e. 3.6V, corresponding digital value is 737.

   Output = Gain * torque sensor input

4. Sends digital output as pulses to the motor controller/ throttle

Based on this algorithm the eternal circuit will be designed which will accommodate to the stated logic of the microcontroller and assist in governing the controller unit of the motor in order to drive the motor. The design of the external circuitry along with the power module is explained in next chapter.

CHAPTER 7

The microcontroller based circuit design

7.1 Introduction

The control algorithm discussed in the earlier chapter has been implemented and burned in to the microcontroller through the program Extreme Burner AVR and according to the stated logic a circuit is designed to meet the conditions of the algorithm. The description of the circuit design is mentioned in this chapter.

7.2 The block diagram for the external circuit

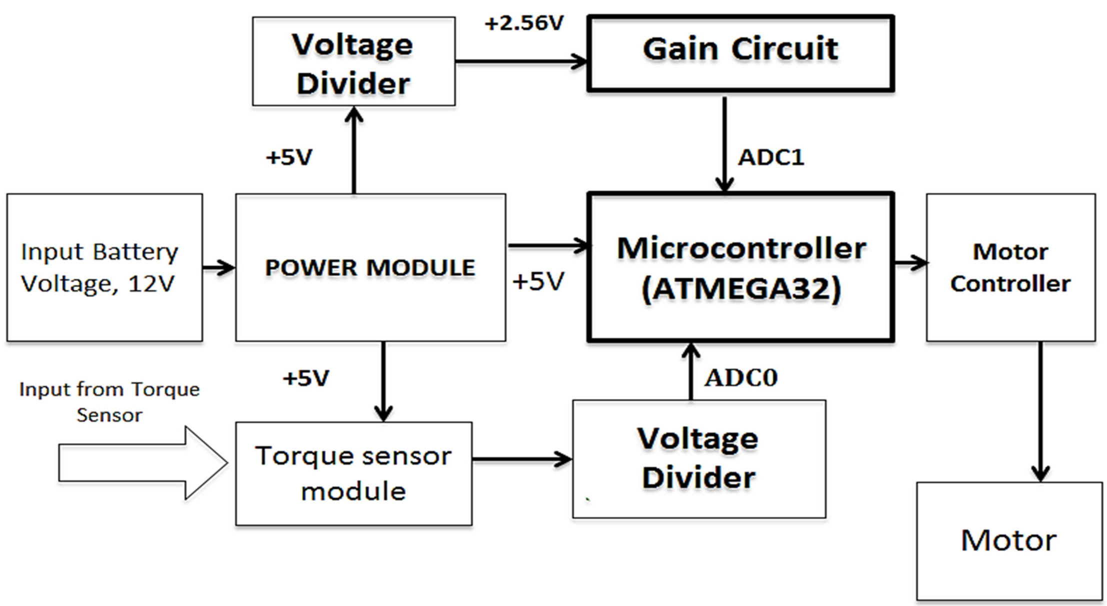


Fig 7.1: Block diagram of the proposed design

The required input voltage to our circuit is 12V, which is taken from the first cell from the three 12V lead acid battery connected in series to produce 36V for the hub motor. This input voltage is fed to a power module from where we obtain an output voltage of 5V. The obtained 5V is chiefly used as the biasing voltage required for the torque sensor module and the microcontroller (ATMEGA32). Furthermore, 5V is also fed to a voltage divider circuit to obtain a voltage of 2.56 V, which is goes in to the gain circuit to vary the overall gain. The microcontroller receives two ADC inputs: ADC0, the input voltage from the torque sensor, and ADC1, being the desired gain. Based on these two inputs, the microcontroller sends an appropriate Pulse Width Modulation (PWM) signal to the controller unit of the motor. The motor controller then drives the motor according to our stated logic.

7.3 The proposed design

7.3.1 Introduction

The core component of this design is the use of the microcontroller (ATmega32A) along with a power module setup consisting of LM7805 and a crowbar circuit. The incoming voltage from the torque sensor module is fed in to one of the microcontrollers input terminals. The logic burned in to the microcontroller manipulates the input data it receives and outputs the desired outcome. Moreover, this design insures a steady flow of 5V from the LM7805 due to the introduction of the crowbar circuit which provides overvoltage protection to the sensitive devices in our circuit. The 5V is required for biasing of the microcontroller as well as to power up the torque sensor and module.

7.3.2 Circuit diagrams and explanations

Our circuit consists of three parts,

I. Power module circuit
II. Supplementary circuits (gain circuit, torque sensor circuit, LCD Interface) and
III. Microcontroller ATmega32A

7.3.2.1 Power module circuit

Microcontrollers are sensitive devices that require a well-regulated power supply as a sudden change or increase in the supply voltage can easily damage these ICs. Therefore, to protect the microcontroller, a circuit design has been introduced where the LM7805 voltage regulator is geared up with an overvoltage protection circuit. Basically, this circuit provides overvoltage protection but it has over current, reverse polarity, and protection for LM7805. So we do not have to worry about the power supply affecting the microcontroller.

To provide biasing voltage for delicate torque sensor, LCD and microcontroller ATmega32A we need to ensure that our power module in any case will **not provide more than 5 volts**. Any voltage above the rating will burn the components, which are expensive.

The circuit design includes a LM7805 5V regulator IC, thyristor NEC2P4M, diodes IN4007, Zener diode IN4738 fuse, capacitor and resistances. This is actually a crowbar circuit.

A crowbar circuit is a method of protecting a circuit against high voltages (overvoltage) in the event of a power supply malfunction or power surge [31]. In addition, if the LM7805 is faulty the supply voltage of 12V could be directly applied to the microcontroller and damage the IC. So, this type of protection circuit is essential for our electric components.
The circuit design is shown below in fig 7.2.

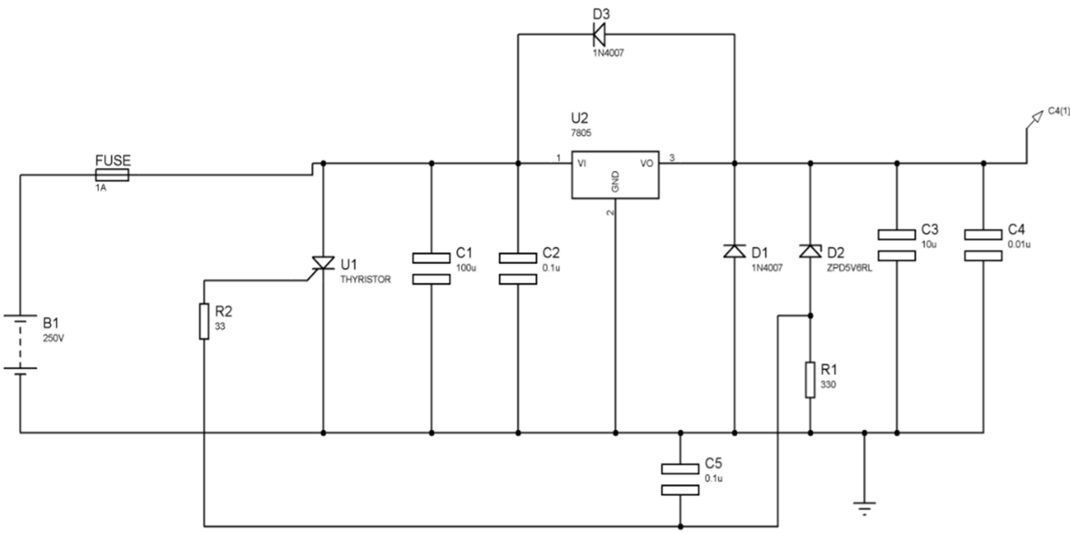


Fig 7.2: Power module circuit

A crowbar circuit works by sensing a voltage that is above a certain threshold and shorting out the power supply. This causes a voltage drop in the rest of the circuit and current surge through the power supply that will trip a circuit-breaker or blow a fuse.

This crowbar circuit has a 12V power supply, and triggers at 5.6V. The zener diode detects the overvoltage condition. At the trigger voltage, its resistance decreases suddenly causing the zener diode to conduct. Current will flow through the Zener diode and trigger the silicon controlled rectifier or thyristor (SCR) to give 2A. This will then provide a short circuit to the ground, thereby protecting the circuitry, that is being supplied to, from any damage and also blow the fuse (fuse rating 1A) that will then remove the voltage from the series regulator.

- The fuse will blow if the current drawn by the circuit is more than 1A.
- The C1 capacitor is present to ensure that short spikes do not trigger the SCR.
- D1 provides output polarity reversal protection to the regulator IC.
- The D3 provides reverse bias protection

In the rated condition, the circuit was tested and voltage found was 5.07 voltage and current to be 440mA. It is observed the regulator does not heat up in the end and continuously burn, which was eminent in the previous models.

7.3.2.2 The supplementary circuits

Torque sensor circuit

Initially, the power module circuit supplies a biasing voltage of +5V to the torque sensor module. When torque is applied on the pedals, the sensor senses the input torque and the module then outputs a corresponding voltage in the form of an analog signal ranging from 0 to 5 volts. This voltage is then passed through a voltage divider circuit i.e. torque sensor circuit. Here, the incoming voltage (the torque sensor module output voltage) is halved since we have set our maximum reference for the ADC (analog to digital conversion) of the microcontroller to be 2.56 volts. The analog voltage of 2.56V gives a digital value of 1023 and this reference has been set internally in the code. In addition, the minimum resolution of voltage remains constant at 4 mV.

The output from the toque sensor circuit is fed to PA0/ADC0 built in pin of the microcontroller ATmega32A for ADC conversion.

Gain Circuit

Again, +5V is derived from the power module circuit which is first passed through a voltage divider circuit, stepping the voltage down to 2.5V and then it is varied by a potentiometer in the range of 0 to 2.5 volts. The varied voltage is fed in to and digitally converted by PA1/ADC1 built in pin of the microcontroller.

LCD interface

In the circuit, a LCD display shows the digitally converted inputs (input to the microcontroller from the torque sensor and the gain circuit) and output (output from the microcontroller to the C.U of the motor) in the form of Vin, Vout and Gain respectively.
Vin displays the digital torque sensor value and Gain displays the amplification factor. Vout displays the digital output as dictated by the algorithm. The LCD has 16 wires all of which are connected, among these 8 data lines (D0 to D7) are from the microcontroller and the other

8 are for the LCD operation. In the midst of the 8 pins needed for LCD operation 3 pins (EN, RS, and RW) are controlled by the microcontroller for enable read, and write operation of LCD. Others are used to power on LCD and varying the contrast ratio.

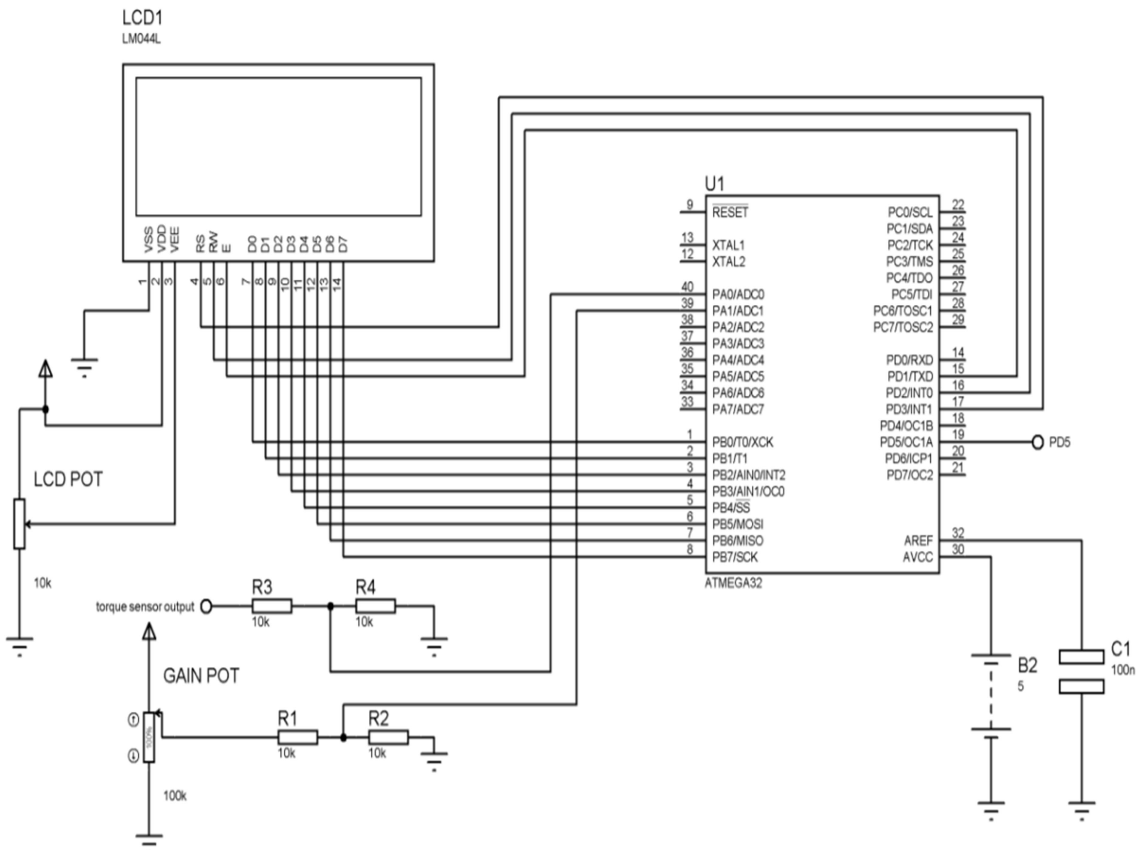


Fig 7.3: Microcontroller and the supplementary circuits

7.3.2.3 Microcontroller circuit

The diagram above shows the relation of the microcontroller with the supplementary circuits. In the circuit design, there are two inputs and one output. Inputs are taken from the toque-sensor and gain circuit via pins PA0/ADC0 and PA1/ADC1 respectively, which are located on pin no. 40 and 39 of the micro controller. Output is obtained from PD5/OC1A pin located on pin no. 19 of the micro controller, which is then fed to motor controller box of the motor.
+5V is the biasing used for the microcontroller, which is provided at the VCC pin located on pin no. 10, and ground is connected to GND pin located on pin no. 11.

For LCD interfacing, data lines for 10 bits of data are sent over from port A (PA0 to PA7). For read/write and enable operation of LCD port D is selected and pins PD1, PD2 and PD3 pins are used, which are located on pins no, 15, 16, 17 respectively.

On the other side of the micro controller, we are required to activate AREF and AVCC pins for setting maximum reference value for the ADC conversion. We have set 5V at AVCC pin located on pin no. 30 and a capacitor of 100uF on AREF pin.

7.3.3 The complete circuit design

Here, the power module is connected to the microcontroller and its supplementary circuits to provide a +5V power supply to these components. The entire finalized circuit design is show in fig 7.3.

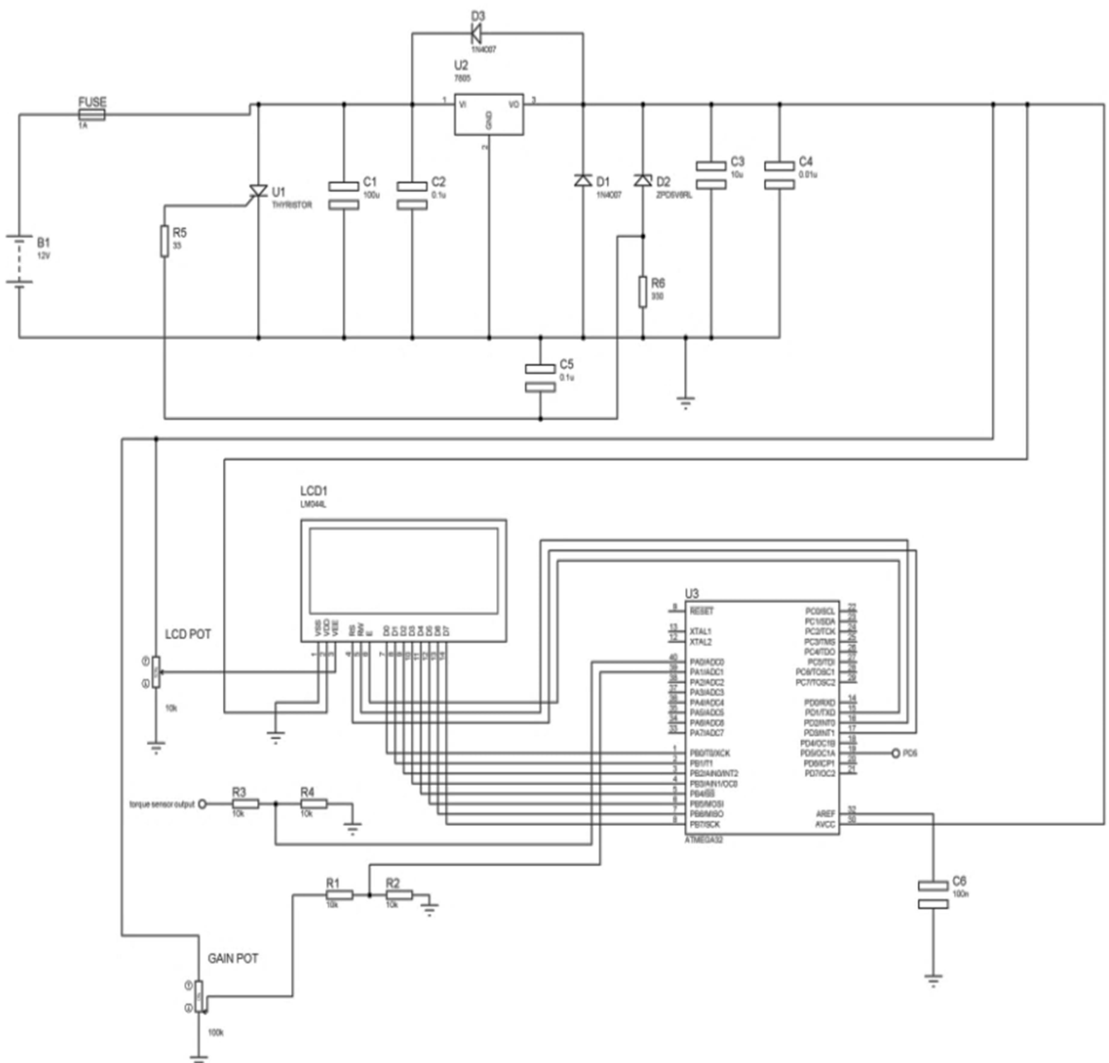


Fig 7.4: Complete circuit design

7.4 Lab Simulation

Now that the circuit has been designed and the code has been burned in to the microcontroller through the Extreme AVR Burner software, the circuit needs to be evaluated in the lab to confirm that our algorithm and circuit design works efficiently under practical lab conditions. It needs to be noted that the circuit design schematic has worked perfectly when the simulation was run in Proteus and only after that have we proceeded to build the circuit on the breadboard.

In the lab, the circuit was built on the bread board as shown below.

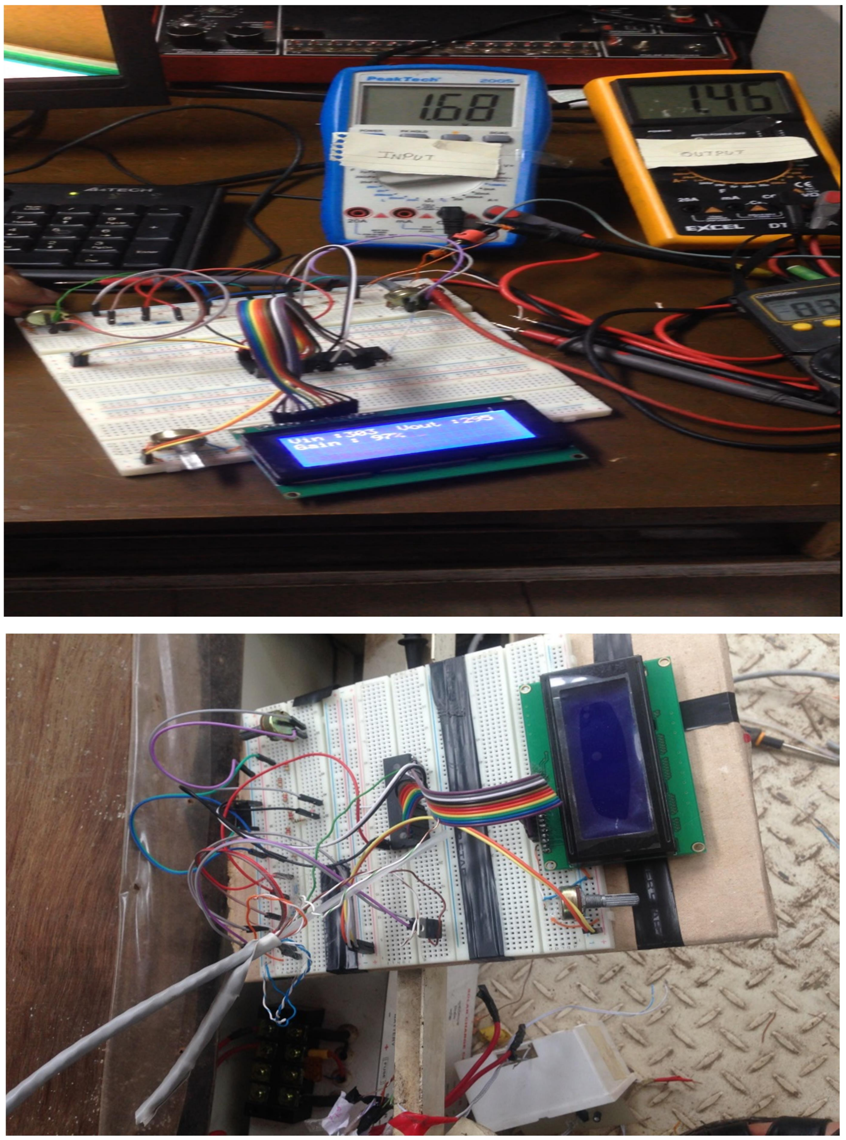


Fig 7.5: Circuit implemented on breadboard

We have supplied the torque sensor circuit with a variable dc voltage which is then connected to a variable potentiometer to mimic the output from the sensor module. The voltage obtained from the potentiometer is then passed through the voltage divider of the torque sensor circuit. The output of the torque sensor circuit is then fed to the ADC0 pin of the microcontroller. 5V is supplied from the power module circuit to provide biasing as well to the gain circuit for varying the gain. For this experiment the gain was set to 1.67 times or 167%.

The table below represents the data obtained from testing in the lab. Vin represents the output of the torque sensor circuit and Vout represents the output from the microcontroller which will be used to drive the motor via the motor controller unit. The voltages were measured using multi-meters, one for measuring the input voltage (Vin) and the other for the output voltage (Vout).

7.4.1 Lab simulation data and Results

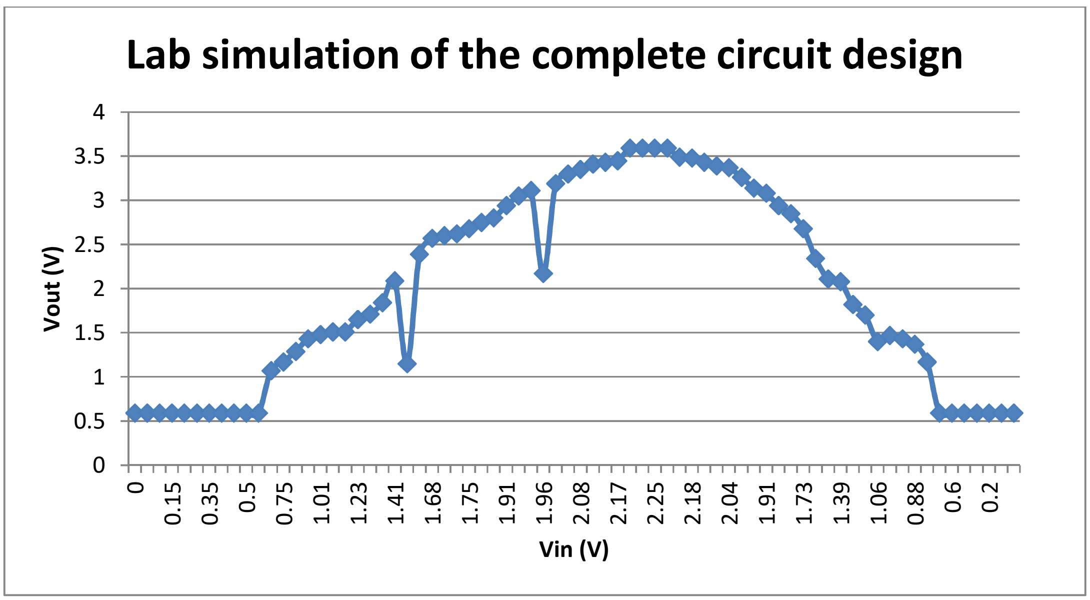


Fig 7.6: Lab simulation results

Vin (V)	Vout (V)		Vin (V)	Vout (V)
0	0.59		2.08	3.35
0.05	0.59		2.11	3.41
0.1	0.59		2.17	3.45
0.15	0.59		2.21	3.59
0.2	0.59		2.23	3.59
0.25	0.59		2.25	3.59
0.35	0.59		2.29	3.59
0.4	0.59		2.19	3.49
0.45	0.59		2.18	3.48

0.5	0.59		2.15	3.43
0.6	0.59		2.1	3.39
0.7	1.07		2.04	3.37
0.75	1.17		2	3.26
0.8	1.29		1.96	3.14
0.95	1.43		1.91	3.08
1.01	1.48		1.85	2.94
1.15	1.51		1.82	2.85
1.17	1.51		1.73	2.68
1.23	1.65		1.61	2.34
1.25	1.71		1.45	2.11
1.29	1.84		1.39	2.08
1.41	2.09		1.3	1.82
1.53	1.15		1.25	1.7
1.64	2.39		1.06	1.4
1.68	2.57		1	1.47
1.69	2.6		0.97	1.43
1.73	2.62		0.88	1.37
1.75	2.68		0.75	1.17
1.79	2.75		0.69	0.59
1.84	2.8		0.6	0.59
1.91	2.94		0.41	0.59
1.92	3.05		0.38	0.59
1.94	3.11		0.2	0.59
1.96	2.17		0.09	0.59
1.97	3.19		0	0.59
2.01	3.3			

Table 7.1 Simulation data

7.4.2 Comments

The data obtained follows the logic that has been set in the microcontroller. We can observe that below the input threshold voltage of 0.5V the microcontroller outputs a constant voltage of 0.59V. As the input voltage gradually exceeds the lower threshold the output voltage increases linearly according to our equation,

Output = Gain * torque sensor input

The output voltage increases only up until the upper threshold voltage which is calculated by, Upper threshold input voltage = 737/gain

The microcontroller does the above calculations as set in the algorithm and after the upper threshold is exceeded the microcontroller outputs a constant voltage of approximately 3.6V. This can be seen in the above data. As the input voltage is decreased gradually the output voltage also decreases and after it decreases below the lower threshold the microcontroller again outputs a constant voltage of 0.59V.

7.5 Conclusion

The complete circuit design for digital circuit meets our expectations and follows the logic and algorithm stated in the microcontroller in the lab conditions. Since, the circuit performs according to requirement it is now time to see the performance of the digital circuit when it is connected to the electric vehicle and to the torque sensor.

The following chapter analyses the operation of the digital circuit when implemented in the electrically assisted wheelchair and also compares the power consumption rate of the digital circuit to the analogue circuit.

CHAPTER 8

Field test and analysis

8.1 Introduction

The goal of our field test was to determine how far wheelchair can keep on running up to 50% SOC voltage discharge with the assistance of torque sensor, after replacing the previous analog circuit by our microcontroller based digital circuit. In addition, we also determined the maximum current that the battery supplied. Multi-meters were utilized to determine voltage across the battery while clamp meter was used to measure current supplied by the battery. The test was carried out near the National Institute of Diseases of the Chest and Hospital area which had a free road for running the wheelchair. The voltage and current readings for the battery were recorded.

8.2 Data Acquisition methods

Multi-meters were used to measure the voltage across the battery. One multi-meter was connected in parallel with 36V battery, another one was connected to torque sensor to find output torque voltage while another one was connected to the output of the microcontroller (pin no. 19). The clamp meter was clipped on to a wire connected in series with the battery. At full load conditions, all of the data were collected. Every one of these readings was recorded by a video camera all through the duration of the field tests. It should be specified that the data were recovered from the video camera at 20 seconds time interval. All data are attached in the appendix part. A GPS tracking system was used to estimate the distance travelled.

8.3 SOC of the Battery

State of charge (**SOC**) is the equivalent of a fuel gauge for the battery pack in a battery electric vehicle (BEV), hybrid vehicle (HV), or plug-in hybrid electric vehicle (PHEV). The units of SOC are percentage points (0% = empty; 100% = full). An alternate form of the same measure is the **depth of discharge** (**DoD**), the inverse of SOC (100% = empty; 0% = full). SOC is normally used when discussing the current state of a battery in use, while DoD is most often seen when discussing the lifetime of the battery after repeated use.[36] Usually, SoC cannot be measured directly but it can be estimated from direct measurement variables in two ways: offline and online. The SOC of the battery has been determined from online source. Measuring SOC is not accurate and so SOC chart obtained from an online source was used as a guideline. The SOC chart is given in the table below,

Percentage of Charge	Specific gravity corrected to	Open – Circuit Voltage					
		6V	8V	12V	24V	36V	48V
100	1.277	6.37	8.49	12.73	25.46	38.20	50.93
90	1.258	6.31	8.41	12.62	25.24	37.85	50.47
80	1.238	6.25	8.33	12.50	25.00	37.49	49.99
70	1.217	6.19	8.25	12.37	24.74	37.12	49.49
60	1.195	6.12	8.16	12.27	24.48	36.72	48.96
50	1.172	6.02	8.07	12.10	24.20	36.31	48.41
40	1.148	5.98	7.97	11.89	23.92	35.87	47.83
30	1.124	5.91	7.88	11.81	23.63	35.44	47.26
20	1.098	5.83	7.77	11.66	23.32	34.97	46.63
10	1.073	5.75	7.67	11.51	23.02	34.52	46.03

Table 8.1 State of Charge as related to Specific Gravity and Open Circuit Voltage [37]

8.4 Field test data with the microcontroller based circuit

Field test was carried out to determine the distance travelled by the wheelchair up to battery discharged down to 50% SOC, the power consumed from the battery as well as to determine the maximum current supplied. Voltage across the battery and current supplied by the battery were recorded. The torque sensor input voltage to the controller unit and output voltage from the torque sensor pedal to the torque adjuster were recorded. All the data were recorded with the help of a video camera and then later we retrieved data at 20 second interval and used to compile the following graphs. Fig 8.2 represents the battery supply current and voltage. Fig 8.3 represents battery power [voltage * current].

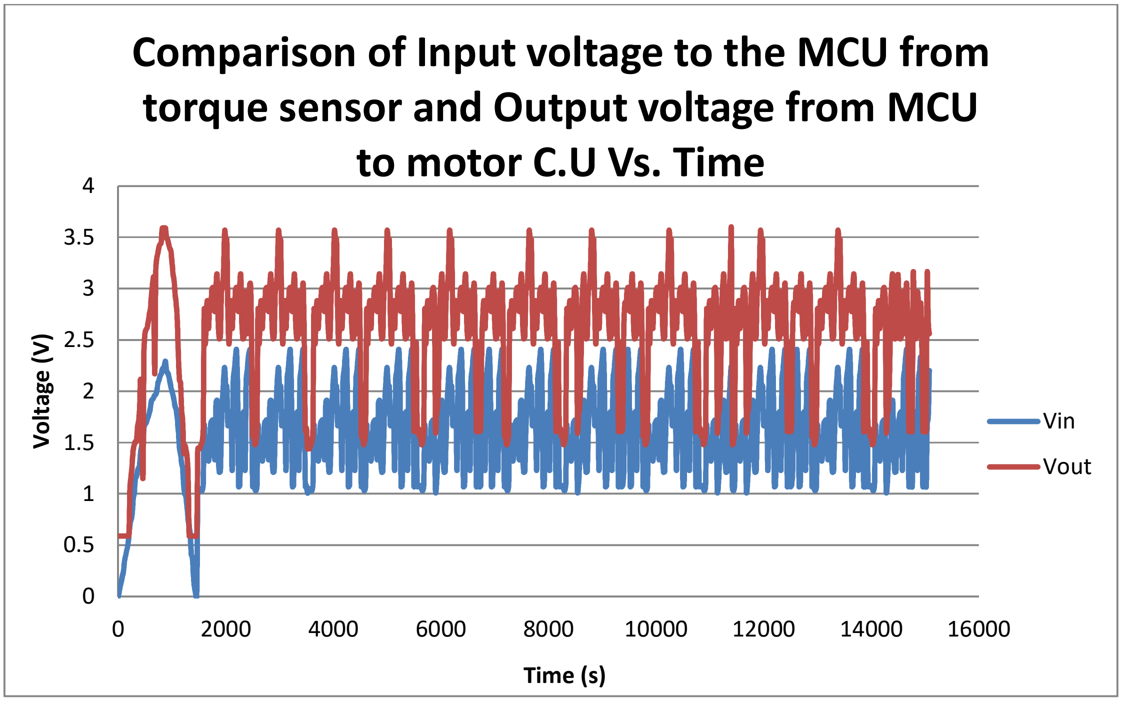


Fig 8.1: Input voltages (V) from torque sensor module to the micro controller circuit and the output voltages (V) from the circuit to the C.U of the motor Vs. Time (seconds)

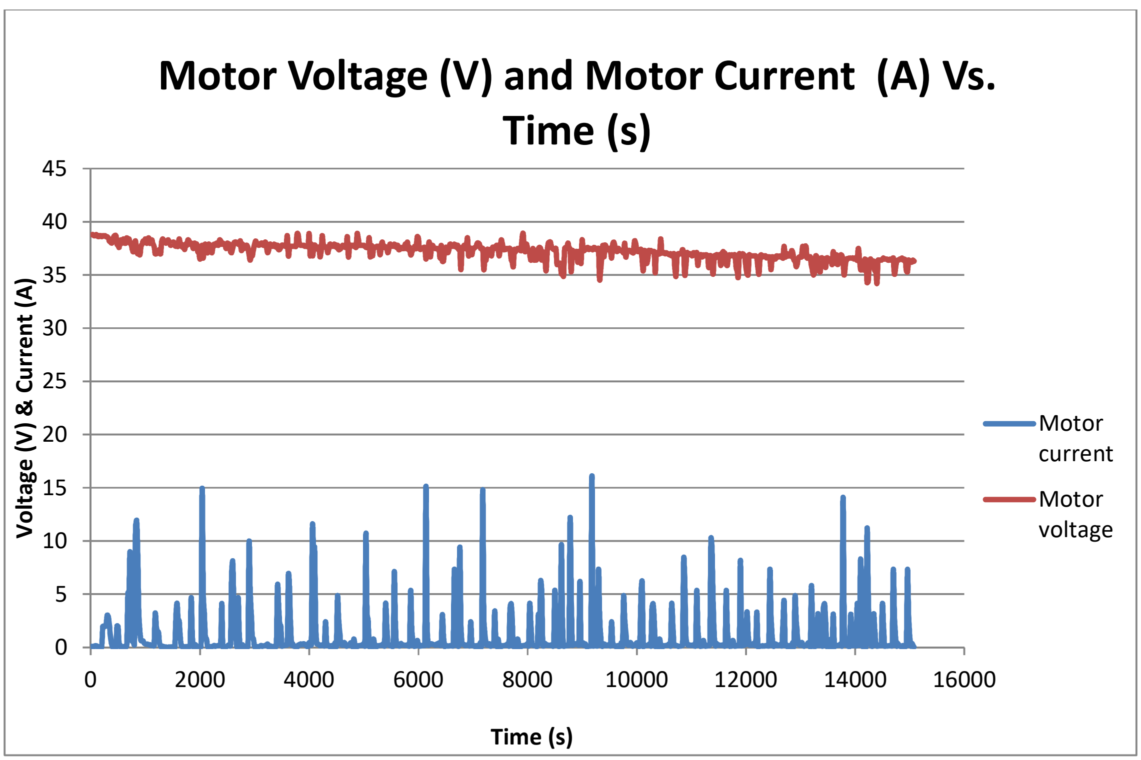


Fig 8.2: Motor Current (A) supplied by battery and the Motor voltage (V) Vs. Time (s)

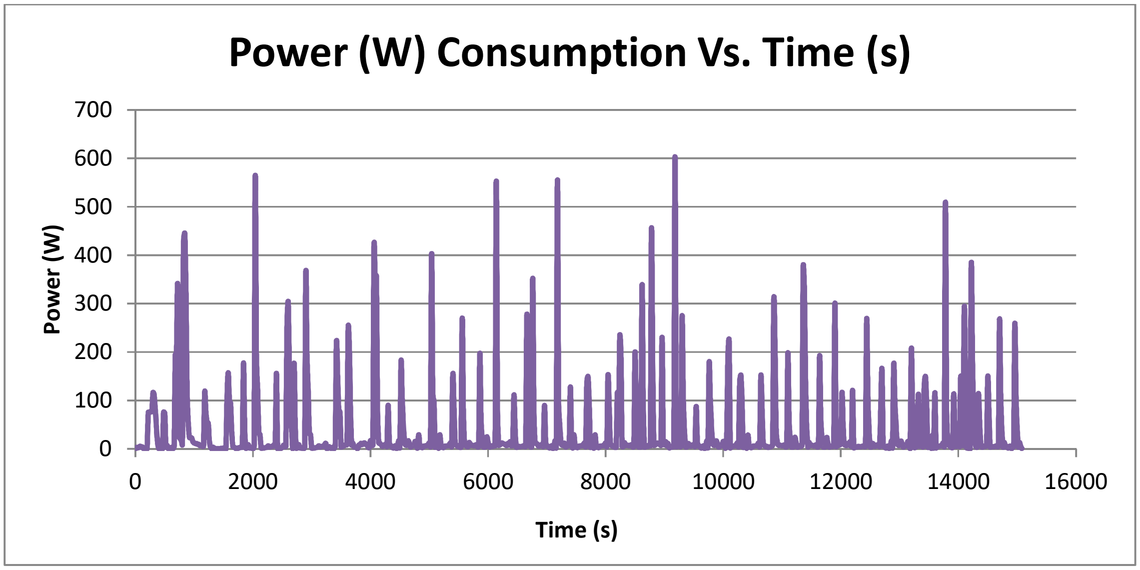


Fig 8.3: Power consumption profile of the digital circuit

The area under the graph in Fig 8.3 for power consumption (W) vs. time (s) is used to calculate the energy consumed by the load from the battery using the trapezoidal rule. In this case, the energy supplied by the battery & the renewable energy supplied by the body force are equal to the energy consumed by the load. The energy that is consumed by the load from battery is 608397.02 Joules.

8.4 Field test data collected with the analog circuit

The data for the analog circuit was collected from the previous tests performed during the initial establishment of the electrically assisted wheelchair. The following graphs are a representation of those data, and display the variation in voltage, current and power consumption of the battery with respect to time.

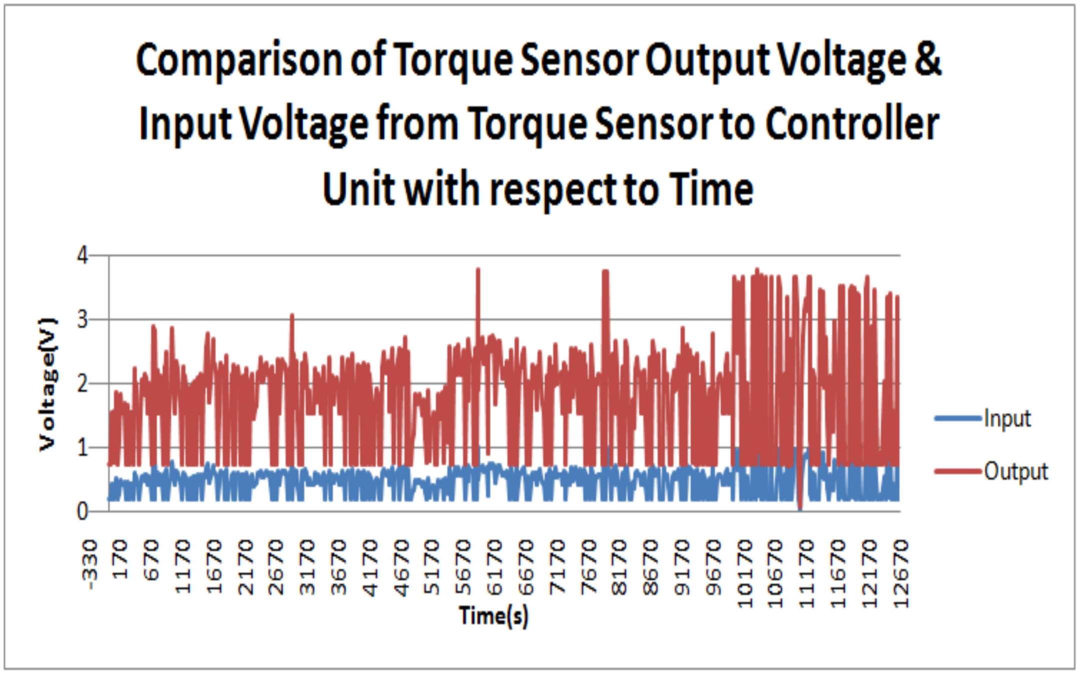


Fig 8.4: Comparison of torque sensor output voltage (V) & input voltage (V) from torque sensor to controller unit with respect to time (s) of the analog circuit [23]

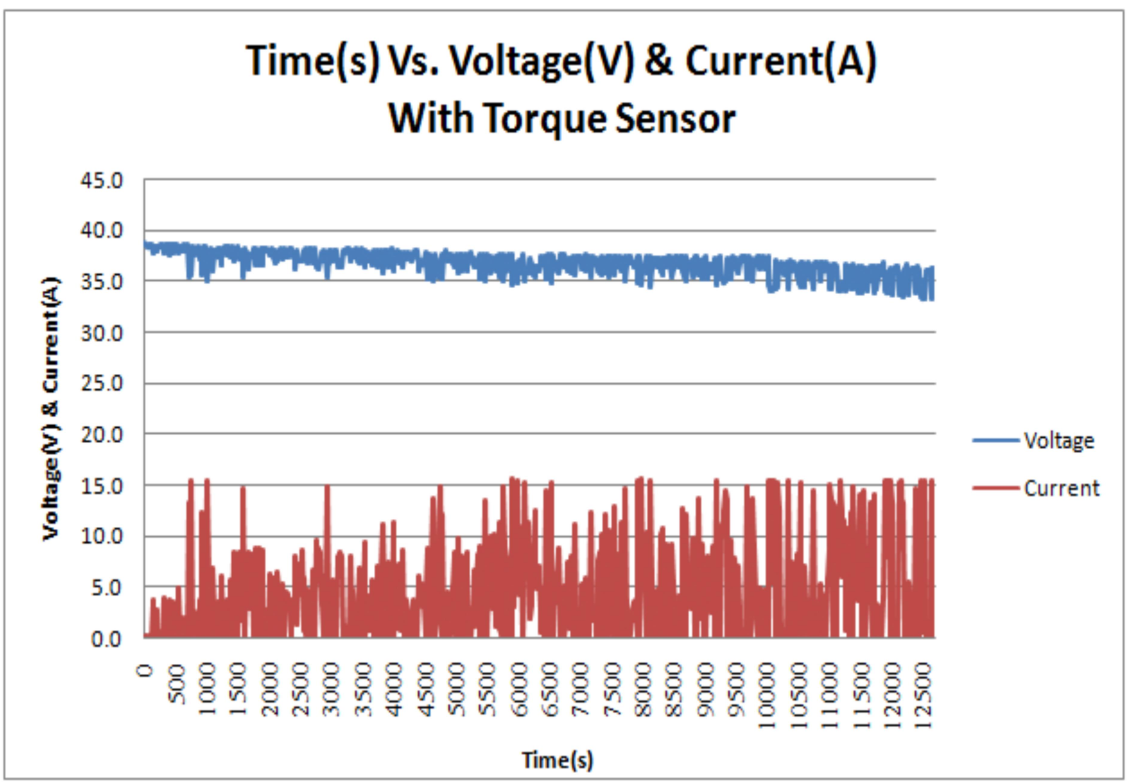


Fig 8.5: Voltage and current profile with the analog circuit [23]

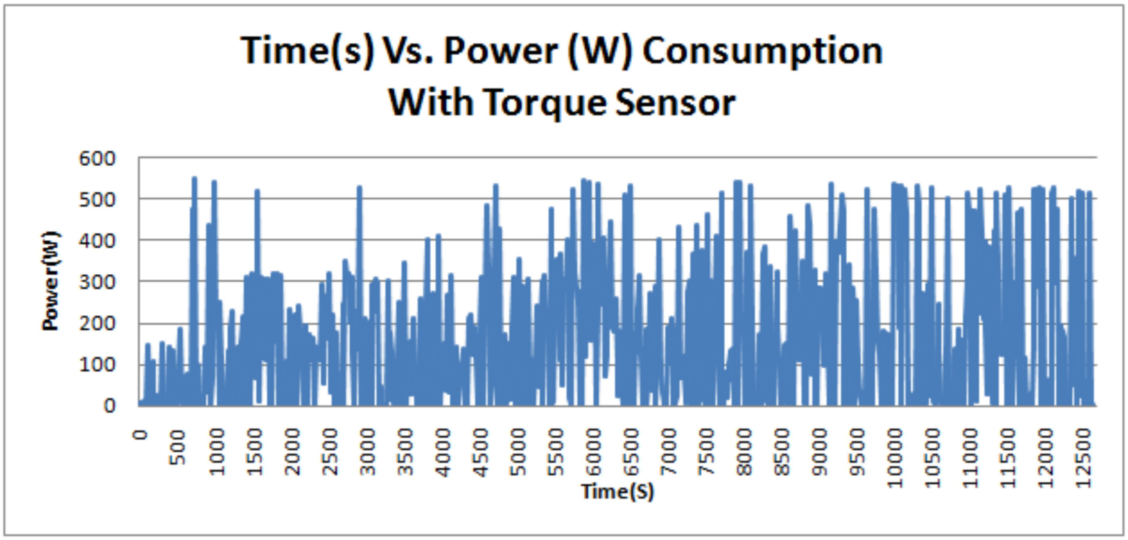


Fig 8.6: Power consumption profile for the analog circuit [23]

The area under the graph in Fig 8.6 power consumption (W) vs. time (s) is used to calculate the energy consumed by the load from the battery. In this case, the energy supplied by the battery & renewable energy supplied by the body force are equal to the energy consumed by the load. The energy that is consumed by the load from battery is 2004674.3 Joules.

8.5 Comparative study

The field test was carried out until the battery had discharged to 50% SOC. The results obtained from our field test reveal that the wheelchair can travel **52 km** in **4.2 hours** (15080 seconds) with the microcontroller based digital circuit whereas the wheelchair had travelled a distance of 42.1 km in 3.52 hours (12660 seconds) with the analog circuit.

The test reveals that with the implementation of the microcontroller based circuit design the total energy consumed from the battery by the load, within the mentioned timeframe, was **608397.02 Joules**. On the contrary, the analog circuit had recorded total energy consumption at 2004674.3 Joules.

From these energy calculations we can find out that the analog circuit had consumed an excess of 1396277.28 Joules. From this value it can be concluded that the digital circuit is more energy efficient since it **consumes 30.35%** energy of the analog design and yet provides sufficient assistance to the user. Hence it **saves 69.65% energy**.

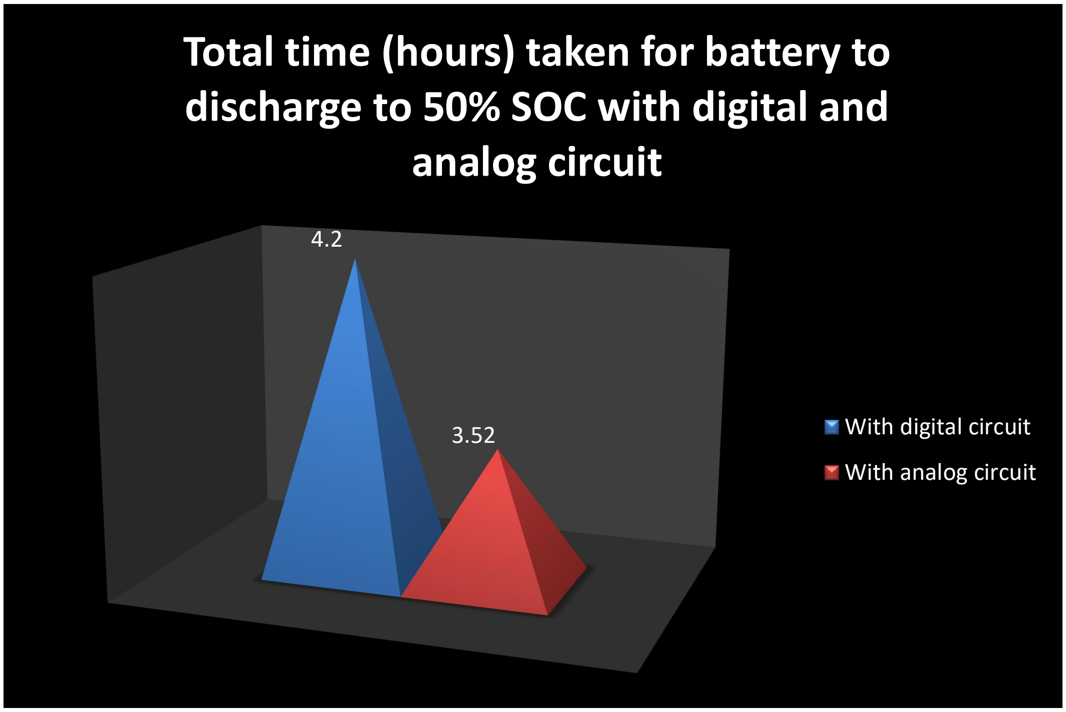


Fig 8.7: Time calculation for digital circuit and analog circuit

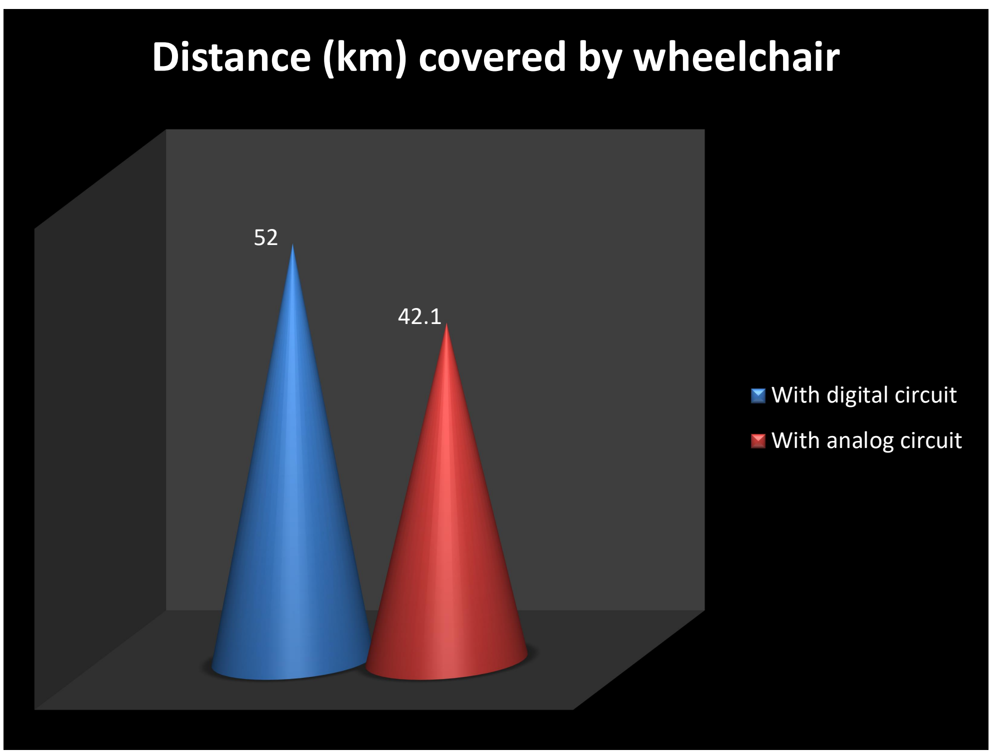


Fig 8.8: Distance travelled with digital and analog circuit

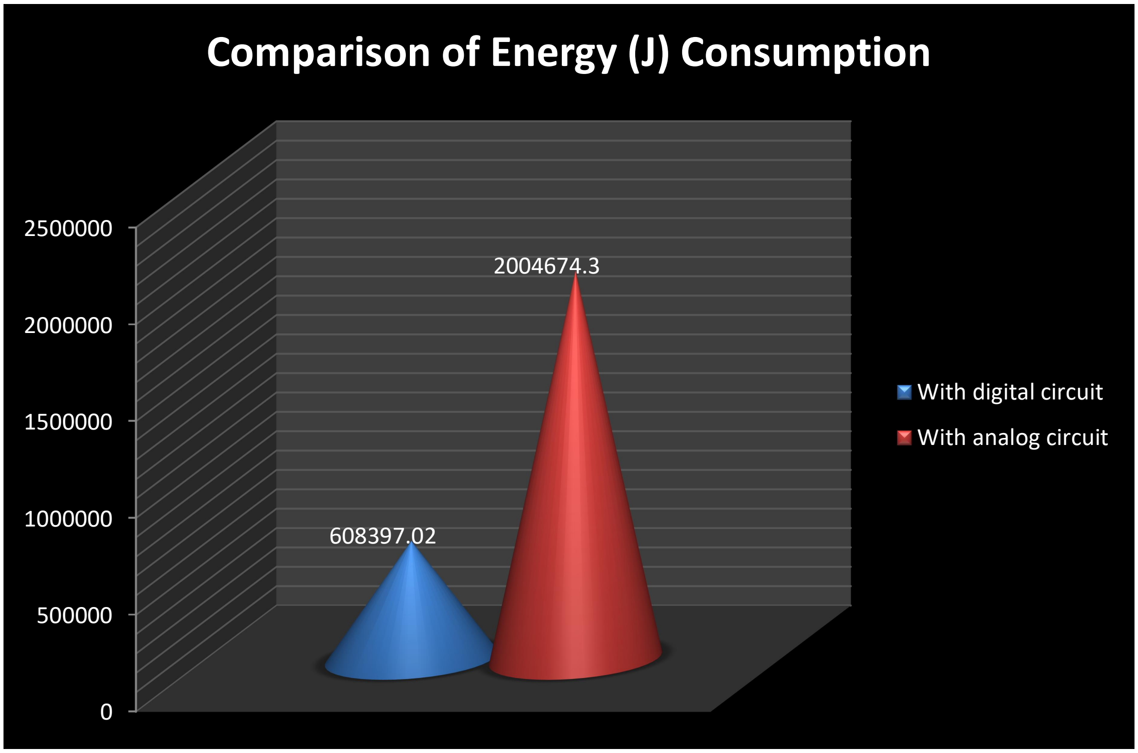


Fig 8.9: Energy consumption calculation with digital and analog circuit

8.6 Overall analysis of the microcontroller based digital circuit

The overall benefits of the digital circuit over the analog circuit,

- The energy saving in our digital circuit is high. Around 70% less energy is consumed in comparison to the previous analog circuit
- Gain control is introduced to control the speed of the vehicle by means of a variable resistor
- Four layered protection is introduced to block any sort of short circuit, over voltage, reverse polarity and over current flow through the circuit to prevent causing harm to the sensitive components of the digital circuitry as well as the torque sensor module.
- The resolution of our digital circuit is more precise (4mV) in comparison to the analog circuit (10mV)
- Overuse of the motor is overcome by the introduction of two voltage levels, below which the motor would not start and over which the motors speed would not increase any further. Such control was not observed in the existing analog circuit.
- In comparison to the analogous circuit, our digital circuit has fewer spikes in the "Current VS Time" curve. This means the motor supplied high current only when the

torque input was sufficiently high. As a result, power drawn from the motor was reduced and the overall power consumption was found to be less in our digital circuit in contrast to the analog circuit. Hence, the making of our digital circuit more cost effective in terms of power consumption.

- Introduction of the speed control system enabled us to provide a smoother ride in contrast to the analog circuit where sudden jerking was felt due to the absence of the speed control system.

Overall drawbacks of the digital circuit,

- Digital circuit has more components hence, the cost of the circuit is higher in comparison to the analog circuit
- Higher initial torque is needed so to compensate this condition, throttle input is taken to rotate the wheels using the motor. Otherwise, providing this initial torque is very difficult.

8.7 Conclusion

This chapter numerically proves the efficiency of the implemented system using practical field test data. It was also evident from the fact that after four long hours of field test 50% charge was still left in the battery (as the battery-level indicator showed). The system also lengthens the battery life as a consequence of using the digital circuit since the close to the rated current flows for a lesser amount of time (as seen from the spikes in the graph displaying current in fig 8.2), which is frequent in the analog circuit controlled system (as seen in fig 8.5).

CHAPTER 9

Conclusion

9.1 Summary

The development of the project "Torque Sensor based and micro-controller controlled Electrically Assisted Mechanical vehicle" was a success after a yearlong research and development. The main objective of this project was to design and develop a prototype of a system that can revolutionize the way human peddled vehicles are driven. After developing the algorithm and implementing it in a micro-controller, the final system was tested using an existing model of an electrically assisted wheelchair for physically disabled people from *The Control and Applications Research Group (CARG)*. The developed prototype then underwent several field tests and the data collected from there was further analyzed. The graphs obtained showed significant improvement in both efficiency and power consumption.

9.2 The PCB design

The microcontroller based digital circuit was implemented in hardware after successful field test analysis. This way it will be a compact system and will eradicate problems arising due to the use of breadboard and a lot of wires connecting the components. When the wheelchair will be used by the patients the roads might not be smooth and due to shaking of the vehicle on uneven roads the wires may become detached. Therefore, hardware implementation will resolve the problems as each of the components are soldered on to the PCB board and there is no chance of any component to fall apart.

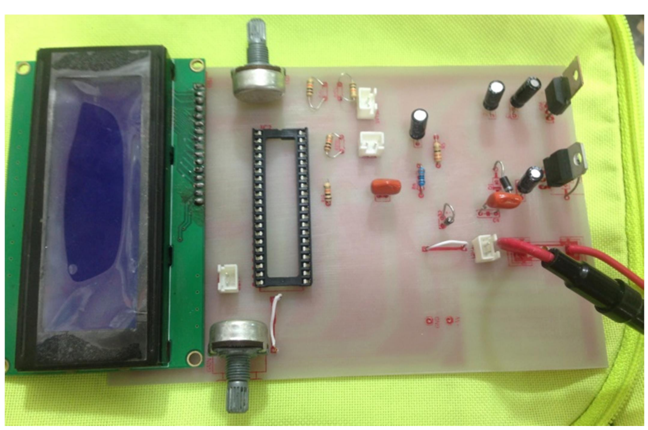


Fig 9.1: Top silk view of PCB

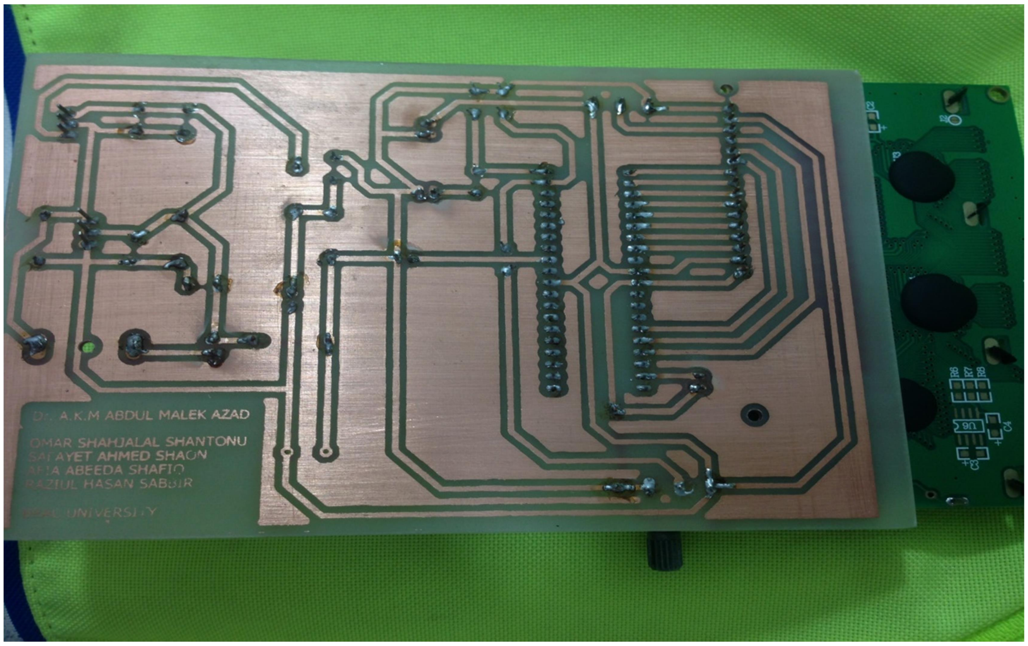


Fig 9.2: Bottom copper view of PCB

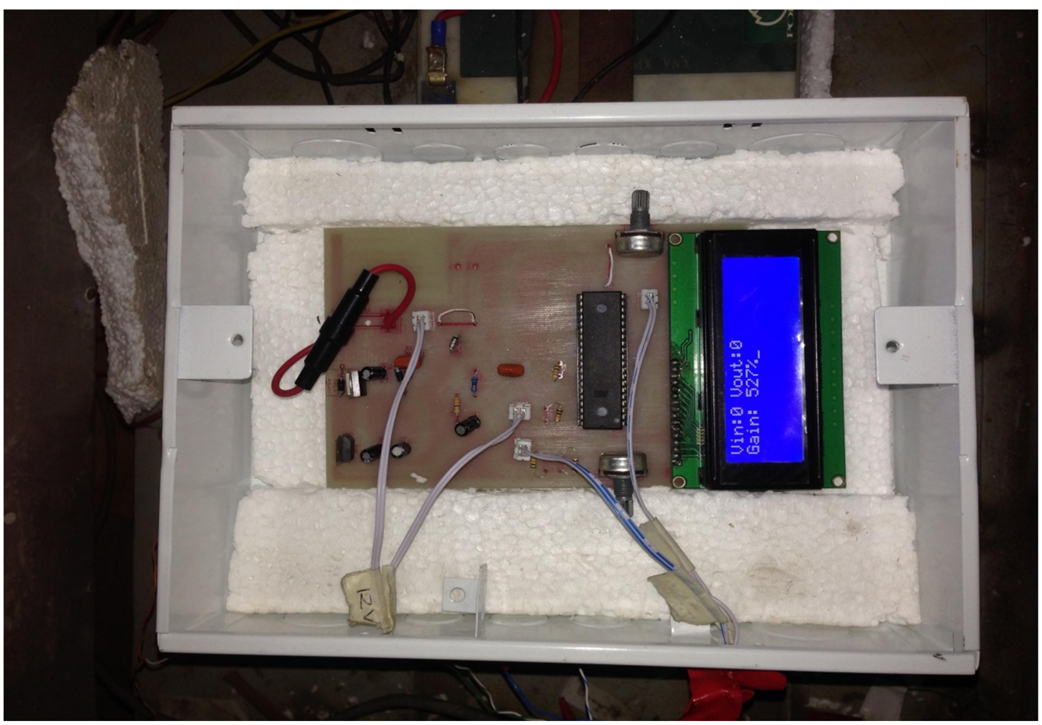


Fig 9.3: PCB implemented in the wheelchair

9.3 Future Works

The Control and Applications Research Group (CARG) has a vision towards taking this as a final project design and implementing the technology in a few more wheelchairs for paraplegic patients. Moreover, our basic goal of the project was to transform the existing ways of driving manually controlled vehicles hence in the future, using the same algorithm, this micro-controller based control system can be implemented in many other manually controlled vehicles like, rickshaws, bicycles, and vans for carrying high amounts of load. Given the cost effectiveness and the longevity of the developed control system, mass production of such wheelchairs for paraplegic patients is also possible in the near future.

REFEENCES

[1] Wheel hub motor (2016). . In *Wikipedia*. Retrieved November 22, 2016 from https://en.wikipedia.org/wiki/Wheel_hub_motor

[2] Torque sensor (2016). . In *Wikipedia*. Retrieved November 22, 2016 from https://en.wikipedia.org/wiki/Torque_sensor

[3] "Simple AVR Programmers". (2015) Retrieved November 20, 2016, from http://elm-chan.org/works/avrx/report_e.html

[4] EXtreme Burner - AVR Free. (2016). Retrieved November 20, 2016, from http://en.freedownloadmanager.org/Windows-PC/eXtreme-Burner-AVR-FREE.html

[5] AVR Studio 5. (n.d.). Retrieved November 20, 2016, from http://www.avr-tutorials.com/avr-studio-5/avr-studio-5

[6] Generating / Creating the AVR HEX File in AVR Studio 5. (n.d.). Retrieved November 20, 2016, from http://www.avr-tutorials.com/avr-studio-5/generating-hex-file-avr-studio-5

[7] A. Gupta. "Tutorial on USBASP AVR Programmer Documentation". (2008). Retrieved November 22, 2016, from www.extremeelectronics.co.in/downloads/usbavrprogrammer/docs/Tutorial.pdf

[8] "USBASP AVR Programmer," in *Future Electronics*. Retrieved November 22, 2016, from http://www.fut-electronics.com/wp-content/uploads/2015/11/USBASP_AVR_Programmer_tutorial.pdf

[9]"AC-PG-USBASP USBASP AVR Programmer" in *ProtoStack*. (2016) Retrieved November 22, 2016, from http://eecs.oregonstate.edu/education/docs/ece375/USBASP-UG.pdf

[10] "How to program an AVR chip using a USBASP (10-pin cable)," in *Learning About Electronics*. (2015). Retrieved November 23, 2016, from http://www.learningaboutelectronics.com/Articles/Program-AVR-chip-using-a-USBASP-with-10-pin-cable.php

[11] Rakesh, "A note on character LCD displays" in *Circuits Today.* (2012). Retrieved November 25, 2016, from http://www.circuitstoday.com/a-note-on-character-lcd-displays

[12] "Atmel- 8255G 8-bit AVR Microcontroller ATmega32A datasheet complete," in *Atmel Corporation*. Retrieved November 29, 2016, from http://www.atmel.com/images/atmel-8155-8-bit-microcontroller-avr-atmega32a_datasheet.pdf, 2015.

[13] Petrykowski, J. (n.d.). Hurtownia części elektronicznych. Retrieved November 29, 2016, from http://micros.com.pl/pl/product/14143/atm32-16pu

[14] ATMEGA32A-PU Atmel 8 Bit 32K AVR Microcontroller, $8.30 : Protostack, AVR Development kits, Boards and Components. (n.d.). Retrieved November 29, 2016, from http://www.protostack.com/microcontrollers/atmega32a-pu-atmel-8-bit-32k-avr-microcontroller

[15] 20x4 LCD Display White/Blue LED Backlight. (n.d.). Retrieved November 29, 2016, from http://www.hobbytronics.co.uk/lcd-20-4-backlight-blue

[16] 20x4 White on Blue Character LCD with Backlight. (n.d.). Retrieved November 29, 2016, from http://www.gravitech.us/20chbllcd.html
[17] Lcd pinout and working. (n.d.). Retrieved November 29, 2016, from http://www.microcontroller-project.com/character-lcd-pinout--working.html
[18] Supplier Homepage. (n.d.). Retrieved November 29, 2016, from http://www.made-in-china.com/showroom/product-images/China-Torque-Sensor-mNlYmlrZQa-jKXmhoWCkJke.html
[19] Torque sensor. (n.d.). Retrieved November 29, 2016, from https://en.wikipedia.org/wiki/Torque_sensor
[20] Rotary Torque Sensors. (n.d.). Retrieved November 29, 2016, from https://www.transducertechniques.com/torque-sensor.aspx
[21] Applied Measurements Ltd Aldermaston, Reading, UK. (n.d.). Retrieved November 29, 2016, from http://www.appmeas.co.uk/torque-transducers-and-torque-sensors.html
[22] Torque sensor operation manual v4.0; Suzhou Victory SincerityTechnology Company Ltd
[23] Khan, A. S. M. M., Islam, T., Urmi, R. J., & Rahman, M. (2015). *Electrically Assisted Wheelchair For Physically Disabled People With Torque Sensor & Dedicated Solar Charger Kit*
[24] Huq, R. M., Shuvo, N. T., Chakraborty, P., & Hoque, M. A. (2012). *Development of Torque Sensor Based Electrically Assisted Hybrid Rickshaw*
[25] Hareendran, T. K. Electronics Projects Circuits. *AVR ADC analog to digital conversion - Tutorial #13*. Retrieved November 24, 2016, from http://www.electroschematics.com/10053/avr-adc/
[26] max Embedded, & Mayank. (2011, June 22). MaxEmbedded » a guide to robotics, embedded electronics and computer vision. *Introduction to AVR timers » maxEmbedded*. Retrieved November 24, 2016, from Atmel AVR, http://maxembedded.com/2011/06/introduction-to-avr-timers/
[27] Sajeev, A. (2010, July 16). *Programming the timers in AVR [avr-gcc] « [Curiosity, Experimentation]*. Retrieved November 24, 2016, from https://appusajeev.wordpress.com/2010/07/16/programming-the-timers-in-avr-avr-gcc/
[28] Barr, M. (2001, September 1). Embedded Systems Programming and Design - Barr Group | Embedded Systems Experts. *Introduction to pulse width modulation (PWM)*. Retrieved November 25, 2016, from http://www.barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation
[29] Sajeev, A. (2010, September 30). *PWM in AVR « [Curiosity, Experimentation]*. Retrieved November 25, 2016, from https://appusajeev.wordpress.com/2010/09/30/pwm-in-avr/
[30] Besinga, R. W.. (2011, April 30). ermicroblog Microcontroller and Electronics Project Blog. *Working with Atmel AVR Microcontroller basic pulse width modulation (PWM) peripheral.* Retrieved November 26, 2016, from http://www.ermicro.com/blog/?p=1971
[31] Practical electronics/Crowbar circuit - Wikibooks, open books for an open world. Retrieved December 4, 2016, from https://en.wikibooks.org/wiki/Practical_Electronics/Crowbar_circuit

[32] World Health Organization Homepage: www.who.int/about/en
[33] Centre for Disability in Development Homepage: www.cdd.org.bd
[34] Centre for Disability in Development Homepage: www.acwf-bd.org
[35]P. Norek, M. Ahmed, et al, Livelihood Challenges for Extremely Poor Disabled People in the Southwest Costal Region of Bangladesh, Shiree working paper 12, January 2013
[36] State of charge (2016). . In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/State_of_charge
[37] Trojan battery company. Retrieved December 9, 2016, from http://www.trojanbattery.com/tech-support/battery-maintenance/

APPENDIX

APPENDIX A

CIRCUIT DESIGN SCHEMATIC

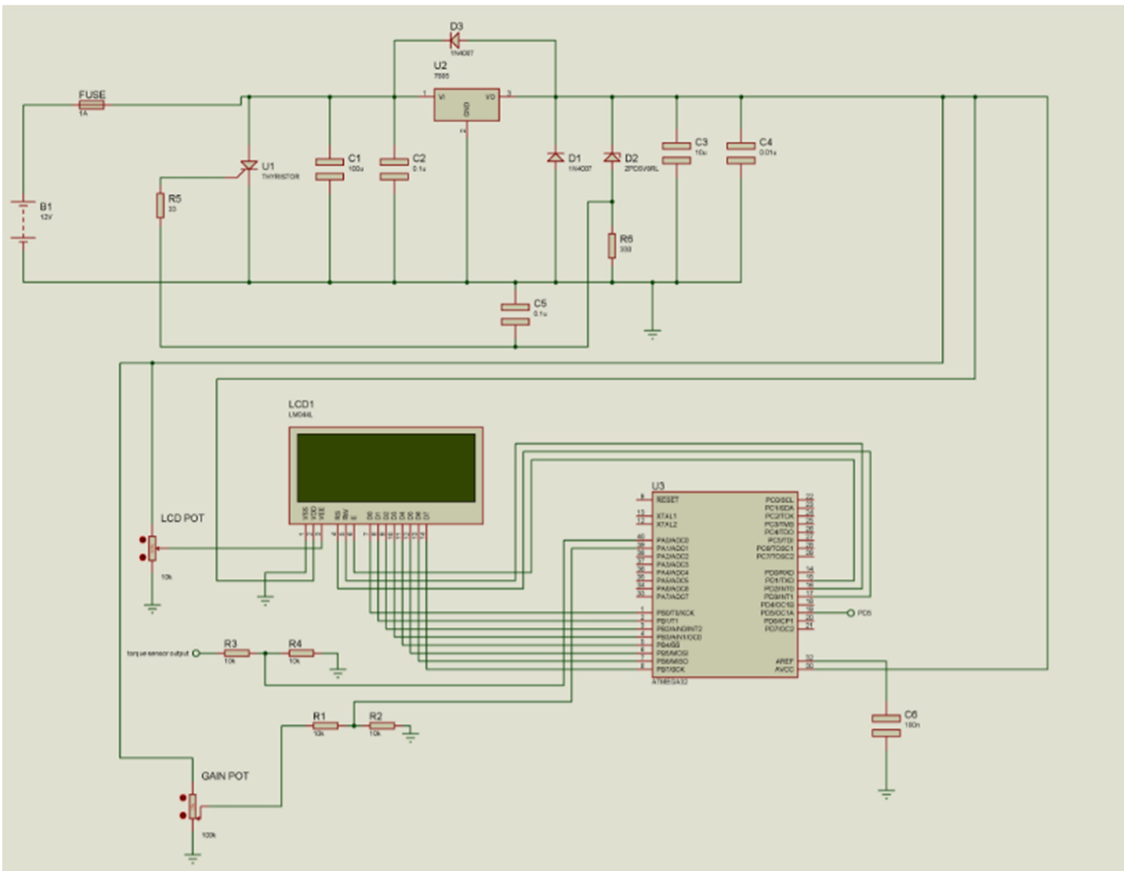


Fig 1: Complete circuit design

APPENDIX B

TABLE 1: FIELD TEST DATA WITH THE DIGITAL CIRCUIT AND THE TORQUE SENSOR PEDAL INSTALLED IN THE ELECRIC WHEELCHAIR

Time (s)	Vin (V)	Vout (V)	Motor current (A)	Motor voltage (V)	Power (W)
0	0	0.59	0	38.8	0
20	0.05	0.59	0.08	38.8	3.104
40	0.1	0.59	0.06	38.8	2.328
60	0.15	0.59	0.08	38.7	3.096
80	0.2	0.59	0.15	38.7	5.805
100	0.25	0.59	0.1	38.8	3.88
120	0.35	0.59	0.11	38.7	4.257
140	0.4	0.59	0.02	38.6	0.772
160	0.45	0.59	0.01	38.7	0.387
180	0.5	0.59	0.02	38.7	0.774
200	0.6	0.59	0.01	38.6	0.386
220	0.7	1.07	1.95	38.7	75.465
240	0.75	1.17	1.96	38.7	75.852
260	0.8	1.29	2	38.7	77.4
280	0.95	1.43	2	38.6	77.2
300	1.01	1.48	3.01	38.6	116.186
320	1.15	1.51	2.9	38.5	111.65
340	1.17	1.51	2.07	38.2	79.074
360	1.23	1.65	1.1	38.1	41.91
380	1.25	1.71	0.32	38	12.16
400	1.29	1.84	0.02	38.1	0.762
420	1.41	2.09	0.03	38.6	1.158
460	1.53	1.15	0.01	38.7	0.387
480	1.64	2.39	2	37.9	75.8
500	1.68	2.57	1.95	37.8	73.71
520	1.69	2.6	0.11	37.6	4.136
540	1.73	2.62	0.19	38.2	7.258
560	1.75	2.68	0.02	38.3	0.766
580	1.79	2.75	0	38	0
600	1.84	2.8	0	38.4	0
620	1.91	2.94	0.09	38.1	3.429

640	1.92	3.05	0.02	38	0.76
660	1.94	3.11	0.14	38.5	5.39
680	1.96	2.17	5.1	38.2	194.82
700	1.97	3.19	5.15	38.4	197.76
720	2.01	3.3	8.9	38	338.2
740	2.08	3.35	0.8	37.9	30.32
760	2.11	3.41	0.57	37.4	21.318
780	2.15	3.43	0.55	37.2	20.46
800	2.17	3.45	0.24	37.9	9.096
820	2.21	3.59	11.23	38.2	428.986
840	2.23	3.59	11.95	37.3	445.735
860	2.25	3.59	8.96	37	331.52
880	2.29	3.59	3.2	37.2	119.04
900	2.19	3.49	1.45	36.9	53.505
920	2.18	3.48	0.63	37.7	23.751
940	2.15	3.43	0.61	37.9	23.119
960	2.1	3.39	0.59	38.2	22.538
980	2.04	3.37	0.39	38.1	14.859
1000	2	3.26	0.32	38.4	12.288
1020	1.96	3.14	0.3	38	11.4
1040	1.91	3.08	0.3	38.1	11.43
1060	1.85	2.94	0.25	38.5	9.625
1080	1.82	2.85	0.25	38	9.5
1100	1.73	2.68	0.18	38.1	6.858
1120	1.61	2.34	0.18	38.3	6.894
1140	1.45	2.11	0.15	37.9	5.685
1160	1.39	2.08	0.15	37.7	5.655
1180	1.3	1.82	3.18	37	117.66
1200	1.25	1.7	2.02	37	74.74
1220	1.06	1.4	1.5	37.1	55.65
1240	1	1.47	1.41	37.2	52.452
1260	0.97	1.43	0.38	37.4	14.212
1280	0.88	1.37	0.1	37	3.7
1300	0.75	1.17	0.01	38.2	0.382
1320	0.69	0.59	0.1	38.4	3.84
1340	0.6	0.59	0.01	38.1	0.381
1360	0.41	0.59	0.03	38.1	1.143
1380	0.38	0.59	0.02	38	0.76
1400	0.2	0.59	0.01	38.2	0.382

1420	0.09	0.59	0.02	38.3	0.766
1460	0	0.59	0.03	37.8	1.134
1480	1.01	1.44	0.02	37.9	0.758
1500	1.02	1.45	0.05	38	1.9
1520	1.04	1.44	0.06	38.1	2.286
1540	1.05	1.48	0.01	37.8	0.378
1560	1.03	1.5	2.9	37.7	109.33
1580	1.07	1.61	4.15	37.9	157.285
1600	1.67	2.78	2.67	37.7	100.659
1620	1.35	2.46	2.45	38.4	94.08
1640	1.41	2.75	0.9	37.9	34.11
1660	1.44	2.88	0.12	37.9	4.548
1680	1.32	2.61	0.1	37.6	3.76
1700	1.44	2.86	0.06	37.8	2.268
1720	1.68	2.79	0	37.4	0
1740	1.71	2.98	0.05	37.9	1.895
1760	1.72	3.01	0.05	37.8	1.89
1780	1.56	2.89	0.08	38.3	3.064
1800	1.34	2.78	0.08	38.2	3.056
1820	1.89	3.03	0.12	37.9	4.548
1840	1.91	3.13	4.67	38	177.46
1860	1.38	2.67	1.14	37.8	43.092
1880	1.21	2.51	0.04	37.5	1.5
1900	1.22	2.53	0.1	38	3.8
1920	1.62	2.96	0.23	37.8	8.694
1940	1.91	3.01	0.06	37.9	2.274
1960	2.11	3.18	0.07	37.2	2.604
1980	2.23	3.56	0.04	37.1	1.484
2000	2.05	3.48	0.01	36.5	0.365
2020	2.05	3.48	0.09	36.9	3.321
2040	1.67	2.78	14.89	37.8	562.842
2060	1.74	2.46	5.23	36.6	191.418
2080	1.79	2.75	3.2	37.8	120.96
2100	1.8	2.88	0.89	37.9	33.731
2120	1.23	2.61	0.79	37.4	29.546
2140	2.11	2.86	0.05	37.1	1.855
2160	2.2	2.79	0.07	37.9	2.653
2180	2.33	2.98	0.05	37.6	1.88
2200	2.4	3.01	0.06	37.4	2.244

2220	2.04	2.89	0.08	37.1	2.968
2240	1.07	2.78	0.1	37.8	3.78
2260	1.08	3.03	0.17	37.9	6.443
2280	1.67	3.13	0.1	37.8	3.78
2300	1.74	2.67	0.13	38	4.94
2320	1.79	2.51	0.14	37.9	5.306
2340	1.8	2.53	0.15	37.8	5.67
2360	1.23	2.96	0.23	38.3	8.809
2380	2.11	2.81	0.31	38.2	11.842
2400	2.2	2.56	4.12	37.9	156.148
2420	2.33	2.86	0.02	38	0.76
2460	2.4	3.04	0.01	37.8	0.378
2480	2.04	2.48	0.04	37.5	1.5
2500	1.07	1.56	0.1	38	3.8
2520	1.08	1.58	0.2	37.8	7.56
2540	1.05	1.48	0.13	37.9	4.927
2560	1.03	1.5	0.14	37.2	5.208
2580	1.07	1.61	6.9	37.1	255.99
2600	1.67	2.78	8.1	37.5	303.75
2620	1.35	2.46	4.3	38	163.4
2640	1.41	2.75	2.1	38.1	80.01
2660	1.44	2.88	1	38.1	38.1
2680	1.32	2.61	0.12	38	4.56
2700	1.44	2.86	4.67	37.9	176.993
2720	1.68	2.79	1.14	37.9	43.206
2740	1.71	2.98	0.04	38.2	1.528
2760	1.72	3.01	0.1	37.3	3.73
2780	1.56	2.89	0.23	37.9	8.717
2800	1.34	2.78	0.06	37.5	2.25
2820	1.89	3.03	0.07	37.6	2.632
2840	1.91	3.13	0.04	37.7	1.508
2860	1.38	2.67	0.01	37.8	0.378
2880	1.21	2.51	0.09	37.5	3.375
2900	1.22	2.53	9.89	36.9	364.941
2920	1.62	2.96	5.23	36.4	190.372
2940	1.91	3.01	3.2	36.9	118.08
2960	2.11	3.18	0.89	36.8	32.752
2980	2.23	3.56	0.79	37.4	29.546
3000	2.05	3.48	0.05	37.9	1.895

3020	2.05	3.48	0.07	38.2	2.674
3040	1.67	2.78	0.05	38	1.9
3060	1.74	2.46	0.06	37.9	2.274
3080	1.79	2.75	0.08	37.8	3.024
3100	1.8	2.88	0.1	37.9	3.79
3120	1.23	2.61	0.17	38	6.46
3140	2.11	2.86	0.1	38	3.8
3160	2.2	2.79	0.13	37.8	4.914
3180	2.33	2.98	0.14	37.9	5.306
3200	2.4	3.01	0.15	38.1	5.715
3220	2.04	2.89	0.23	37.6	8.648
3240	1.07	2.78	0.31	37.1	11.501
3260	1.08	3.03	0.02	37.4	0.748
3280	1.67	3.13	0.1	37.9	3.79
3300	1.74	2.67	0.12	37.8	4.536
3320	1.79	2.51	0.04	37.8	1.512
3340	1.8	2.53	0.05	37.7	1.885
3360	1.23	2.96	0.09	37.9	3.411
3380	2.11	2.81	0.02	37.8	0.756
3400	2.2	2.56	0.14	37.6	5.264
3420	2.33	2.86	5.78	37.8	218.484
3440	2.4	3.04	4.13	37.4	154.462
3460	2.04	2.48	2.18	37.4	81.532
3480	1.07	1.61	2.01	37.6	75.576
3500	1.08	1.64	0.11	37.8	4.158
3520	1.01	1.44	0.14	37.8	5.292
3540	1.02	1.45	0.15	37.8	5.67
3560	1.04	1.44	0.21	37.7	7.917
3580	1.05	1.48	0.23	37.7	8.671
3600	1.03	1.5	0.24	38.7	9.288
3620	1.07	1.61	6.8	36.8	250.24
3640	1.67	2.78	5.6	37.5	210
3660	1.35	2.46	2.1	37.7	79.17
3680	1.41	2.75	0.22	37.7	8.294
3700	1.44	2.88	0.31	37.7	11.687
3720	1.32	2.61	0.11	37.7	4.147
3740	1.44	2.86	0.1	37.7	3.77
3760	1.68	2.79	0.09	37.7	3.393
3780	1.71	2.98	0.02	38.9	0.778

3800	1.72	3.01	0.2	38.8	7.76
3820	1.56	2.89	0.22	37.9	8.338
3840	1.34	2.78	0.31	37.8	11.718
3860	1.89	3.03	0.24	37.8	9.072
3880	1.91	3.13	0.31	37.8	11.718
3900	1.38	2.67	0.33	37.8	12.474
3920	1.21	2.51	0.25	37.7	9.425
3940	1.22	2.53	0.26	37.7	9.802
3960	1.62	2.96	0.08	37.7	3.016
3980	1.91	3.01	0.08	37.7	3.016
4000	2.11	3.18	0.45	38.9	17.505
4020	2.23	3.56	0.23	36.8	8.464
4040	2.05	3.48	0.21	37.9	7.959
4060	2.05	3.48	11.35	36.8	417.68
4080	1.67	2.78	8.94	37.3	333.462
4100	1.74	2.46	9.41	37.8	355.698
4120	1.79	2.75	2.11	37.7	79.547
4140	1.8	2.88	0.34	37.6	12.784
4160	1.23	2.61	0.31	37.4	11.594
4180	2.11	2.86	0.25	37.6	9.4
4200	2.2	2.79	0.15	36.7	5.505
4220	2.33	2.98	0.18	37.3	6.714
4240	2.4	3.01	0.23	38.7	8.901
4260	2.04	2.89	0.19	37.7	7.163
4280	1.07	2.78	0.14	37.7	5.278
4300	1.08	3.03	2.4	37.6	90.24
4320	1.67	3.13	0.37	37.6	13.912
4340	1.74	2.67	0.16	37.1	5.936
4360	1.79	2.51	0.15	37.4	5.61
4380	1.8	2.53	0.06	37.9	2.274
4400	1.23	2.96	0.08	37.8	3.024
4420	2.11	2.81	0.17	37.8	6.426
4440	2.2	2.56	0.03	37.7	1.131
4460	2.33	2.86	0.08	37.9	3.032
4480	2.4	3.04	0.45	37.8	17.01
4500	2.04	2.48	0.09	37.6	3.384
4520	1.07	1.56	4.8	37.8	181.44
4540	1.08	1.58	2.68	37.4	100.232
4560	1.05	1.48	1.31	37.4	48.994

4580	1.03	1.5	0.33	37.5	12.375
4600	1.07	1.61	0.21	37.7	7.917
4620	1.67	2.78	0.22	37.8	8.316
4640	1.35	2.46	0.45	37.7	16.965
4660	1.41	2.75	0.33	37.6	12.408
4680	1.44	2.88	0.11	37.7	4.147
4700	1.32	2.61	0.15	38.6	5.79
4720	1.44	2.86	0.21	36.7	7.707
4740	1.68	2.79	0.19	37.8	7.182
4760	1.71	2.98	0.42	37.8	15.876
4780	1.72	3.01	0.31	37.8	11.718
4800	1.56	2.89	0.05	37.8	1.89
4820	1.34	2.78	0.78	37.7	29.406
4840	1.89	3.03	0.11	37.7	4.147
4860	1.91	3.13	0.12	37.6	4.512
4880	1.38	2.67	0.07	38.9	2.723
4900	1.21	2.51	0.15	38.4	5.76
4920	1.22	2.53	0.16	37.8	6.048
4940	1.62	2.96	0.17	37.8	6.426
4960	1.91	3.01	0.18	37.8	6.804
4980	2.11	3.18	0.23	37.8	8.694
5000	2.23	3.56	0.33	37.8	12.474
5020	2.05	3.48	0.13	37.7	4.901
5040	2.05	3.48	10.7	37.6	402.32
5060	1.67	2.78	3.1	37.6	116.56
5080	1.74	2.46	2.56	37.6	96.256
5100	1.79	2.75	0.11	38.5	4.235
5120	1.8	2.88	0.34	36.7	12.478
5140	1.23	2.61	0.31	37.8	11.718
5160	2.11	2.86	0.05	36.8	1.84
5180	2.2	2.79	0.78	37.6	29.328
5200	2.33	2.98	0.11	37.6	4.136
5220	2.4	3.01	0.12	37.6	4.512
5240	2.04	2.89	0.07	37.7	2.639
5260	1.07	2.78	0.15	37.7	5.655
5280	1.08	3.03	0.16	37.6	6.016
5300	1.67	3.13	0.17	36.9	6.273
5320	1.74	2.67	0.18	37.7	6.786
5340	1.79	2.51	0.23	38.1	8.763

5360	1.8	2.53	0.33	37.6	12.408
5380	1.23	2.96	0.13	37.7	4.901
5400	2.11	2.81	4.1	37.6	154.16
5420	2.2	2.56	2.3	37.6	86.48
5440	2.33	2.86	0.31	37.1	11.501
5460	2.4	3.04	0.22	37.4	8.228
5480	2.04	2.48	0.14	37.9	5.306
5500	1.07	1.61	0.15	37.8	5.67
5520	1.08	1.64	0.09	37.8	3.402
5540	2.04	2.48	0.1	37.7	3.77
5560	1.07	1.61	7.11	37.9	269.469
5580	1.08	1.64	2.15	37.8	81.27
5600	1.07	1.56	0.21	37.6	7.896
5620	1.08	1.58	0.25	37.8	9.45
5640	1.05	1.48	0.14	37.4	5.236
5660	1.03	1.5	0.13	37.4	4.862
5680	1.07	1.61	0.36	37.6	13.536
5700	1.67	2.78	0.21	37.5	7.875
5720	1.35	2.46	0.22	37.5	8.25
5740	1.41	2.75	0.19	37.5	7.125
5760	1.44	2.88	0.17	37.9	6.443
5780	1.32	2.61	0.21	37.5	7.875
5800	1.44	2.86	0.22	38.1	8.382
5820	1.68	2.79	0.11	37.4	4.114
5840	1.71	2.98	0.13	37.6	4.888
5860	1.72	3.01	5.34	36.9	197.046
5880	1.56	2.89	1.68	37.1	62.328
5900	1.02	1.61	0.56	37.4	20.944
5920	1.11	1.77	0.12	37.6	4.512
5940	1.33	1.84	0.11	37.7	4.147
5960	1.51	2.13	0.13	37.5	4.875
5980	1.34	2.78	0.65	38.6	25.09
6000	1.89	3.03	0.32	38.5	12.32
6020	1.91	3.13	0.11	37.5	4.125
6040	1.38	2.67	0.13	37.4	4.862
6060	1.21	2.51	0.17	37.5	6.375
6080	1.22	2.53	0.14	37.5	5.25
6100	1.62	2.96	0.21	37.6	7.896
6120	1.91	3.01	0.18	37.6	6.768

6140	2.11	3.18	15.14	36.5	552.61
6160	2.23	3.56	2.17	37.6	81.592
6180	2.05	3.48	0.21	37.5	7.875
6200	2.05	3.48	0.22	38.1	8.382
6220	1.67	2.78	0.31	36.7	11.377
6240	1.74	2.46	0.33	37.7	12.441
6260	1.79	2.75	0.29	37.8	10.962
6280	1.8	2.88	0.26	37.6	9.776
6300	1.23	2.61	0.21	37.5	7.875
6320	2.11	2.88	0.2	37.5	7.5
6340	2.2	2.79	0.31	37.5	11.625
6360	2.33	2.98	0.22	37.4	8.228
6380	2.4	3.01	0.41	37.4	15.334
6400	2.04	2.89	0.17	36.6	6.222
6420	1.07	2.78	0.31	37.8	11.718
6440	1.08	3.03	3.07	36.3	111.441
6460	1.67	3.13	1.09	36.3	39.567
6480	1.74	2.67	0.1	37.4	3.74
6500	1.79	2.51	0.11	37.6	4.136
6520	1.8	2.53	0.15	37.6	5.64
6540	1.23	2.96	0.12	37.1	4.452
6560	2.11	2.81	0.13	37.4	4.862
6580	2.2	2.56	0.14	37.9	5.306
6600	2.33	2.86	0.17	37.8	6.426
6620	2.4	3.04	0.13	37.8	4.914
6640	2.04	2.48	0.14	37.7	5.278
6660	1.07	1.61	7.3	37.9	276.67
6680	1.08	1.64	3.1	37.8	117.18
6700	2.04	2.48	1.1	37.6	41.36
6720	1.07	1.61	0.12	37.8	4.536
6740	1.08	1.64	0.11	37.4	4.114
6760	1.74	2.46	9.41	37.4	351.934
6780	1.79	2.75	2.11	35.5	74.905
6800	1.8	2.88	0.34	37.6	12.784
6820	1.23	2.61	0.31	37.6	11.656
6840	2.11	2.86	0.25	37.8	9.45
6860	2.2	2.79	0.15	37.5	5.625
6880	2.33	2.98	0.18	37.5	6.75
6900	2.4	3.01	0.23	38.5	8.855

6920	2.04	2.89	0.19	36.8	6.992
6940	1.07	2.78	0.14	37.5	5.25
6960	1.08	3.03	2.4	37.5	90
6980	1.67	3.13	0.37	36.9	13.653
7000	1.74	2.67	0.16	37.5	6
7020	1.79	2.51	0.15	37.5	5.625
7040	1.8	2.53	0.06	37.6	2.256
7060	1.23	2.96	0.08	37.7	3.016
7080	2.11	2.81	0.17	38	6.46
7100	2.2	2.56	0.03	38	1.14
7120	2.33	2.86	0.08	37.9	3.032
7140	2.4	3.04	0.45	37.5	16.875
7160	2.04	2.48	0.09	37.5	3.375
7180	1.07	1.56	14.8	37.5	555
7200	1.08	1.58	2.68	35.5	95.14
7220	1.05	1.48	1.31	36.6	47.946
7240	1.03	1.5	0.33	36.4	12.012
7260	1.07	1.61	0.21	37.4	7.854
7280	1.67	2.78	0.22	37.4	8.228
7300	1.41	2.75	0.33	36.9	12.177
7320	1.44	2.88	0.11	37.4	4.114
7340	1.32	2.61	0.15	36.8	5.52
7360	1.44	2.86	0.21	37.6	7.896
7380	1.68	2.79	0.19	37.5	7.125
7400	1.71	2.98	3.42	37.5	128.25
7420	1.72	3.01	0.31	37.7	11.687
7440	1.56	2.89	0.05	37.8	1.89
7460	1.34	2.78	0.78	37.7	29.406
7480	1.89	3.03	0.11	36.8	4.048
7500	1.91	3.13	0.12	37.6	4.512
7520	1.38	2.67	0.07	38.3	2.681
7540	1.21	2.51	0.15	37.9	5.685
7560	1.22	2.53	0.16	37.4	5.984
7580	1.62	2.96	0.17	37.4	6.358
7600	1.91	3.01	0.18	37.4	6.732
7620	2.11	3.18	0.23	37.5	8.625
7640	2.23	3.56	0.33	37.5	12.375
7660	2.05	3.48	0.13	37.5	4.875
7680	2.05	3.48	3.7	37.5	138.75

7700	1.67	2.78	4.1	36.5	149.65
7720	1.74	2.46	2.56	36.2	92.672
7740	1.79	2.75	0.11	35.8	3.938
7760	1.8	2.88	0.34	37.4	12.716
7780	1.23	2.61	0.31	37.2	11.532
7800	2.11	2.86	0.05	37.4	1.87
7820	2.2	2.79	0.78	37.5	29.25
7840	2.33	2.98	0.11	37.5	4.125
7860	2.4	3.01	0.12	37.5	4.5
7880	2.04	2.89	0.07	37.3	2.611
7900	1.07	2.78	0.15	38.4	5.76
7920	1.08	3.03	0.16	38.9	6.224
7940	1.67	3.13	0.17	37.3	6.341
7960	1.74	2.67	0.18	37.2	6.696
7980	1.79	2.51	0.23	37.9	8.717
8000	1.8	2.53	0.33	37.5	12.375
8020	1.23	2.96	0.13	37.3	4.849
8040	2.11	2.81	4.1	37	151.7
8060	2.2	2.56	2.3	36.4	83.72
8080	2.33	2.86	0.31	36.7	11.377
8100	2.4	3.04	0.22	37.5	8.25
8120	2.04	2.48	0.14	36.4	5.096
8140	1.07	1.61	0.15	37.1	5.565
8160	1.08	1.64	0.09	36.4	3.276
8180	2.04	2.48	0.1	37.2	3.72
8200	1.07	1.61	3.11	37.5	116.625
8220	1.08	1.64	1.15	37.6	43.24
8240	1.07	1.56	6.21	37.5	232.875
8260	1.08	1.58	5.25	36.4	191.1
8280	1.05	1.48	0.14	37.2	5.208
8300	1.03	1.5	0.13	36.4	4.732
8320	1.07	1.61	0.36	36.4	13.104
8340	1.67	2.78	0.21	35.4	7.434
8360	1.35	2.46	0.22	37.1	8.162
8380	1.41	2.75	0.19	38	7.22
8400	1.44	2.88	0.17	38.1	6.477
8420	1.32	2.61	0.21	38.1	8.001
8440	1.44	2.86	0.22	37.6	8.272
8460	1.68	2.79	0.11	37.5	4.125

8480	1.71	2.98	0.13	37.4	4.862
8500	1.72	3.01	5.34	37.4	199.716
8520	1.56	2.89	1.68	37.5	63
8540	1.02	1.61	0.56	36	20.16
8560	1.11	1.77	0.12	36.1	4.332
8580	1.33	1.84	0.11	36.1	3.971
8600	1.51	2.13	0.13	36.1	4.693
8620	1.34	2.78	9.65	35.1	338.715
8640	1.89	3.03	2.32	35.2	81.664
8660	1.91	3.13	0.11	34.9	3.839
8680	1.38	2.67	0.13	37.5	4.875
8700	1.21	2.51	0.17	37.6	6.392
8720	1.22	2.53	0.14	37.5	5.25
8740	1.62	2.96	0.21	38	7.98
8760	1.91	3.01	0.18	37.6	6.768
8780	2.11	3.18	12.14	37.4	454.036
8800	2.23	3.56	5.17	35.3	182.501
8820	2.05	3.48	0.21	37.3	7.833
8840	2.05	3.48	0.22	37.2	8.184
8860	1.67	2.78	0.31	37.2	11.532
8880	1.74	2.46	0.33	37.2	12.276
8900	1.79	2.75	0.29	37.2	10.788
8920	1.8	2.88	0.26	36.2	9.412
8940	1.23	2.61	0.21	37.2	7.812
8960	2.11	2.86	6.2	37.2	230.64
8980	2.2	2.79	0.31	36.1	11.191
9000	2.33	2.98	0.22	38.3	8.426
9020	2.4	3.01	0.41	37.3	15.293
9040	2.04	2.89	0.17	37.3	6.341
9060	1.07	2.78	0.31	37.4	11.594
9080	1.08	3.03	0.07	37.5	2.625
9100	1.67	3.13	0.09	37.6	3.384
9120	1.74	2.67	0.1	37.4	3.74
9140	1.79	2.51	0.11	37.5	4.125
9160	1.8	2.53	0.15	37.5	5.625
9180	1.23	2.96	16.12	37.4	602.888
9200	2.11	2.81	0.13	37.4	4.862
9220	2.2	2.56	0.14	37.6	5.264
9240	2.33	2.86	0.17	37.5	6.375

9260	2.4	3.04	0.13	37.4	4.862
9280	2.04	2.48	0.14	37.4	5.236
9300	1.07	1.61	7.3	37.5	273.75
9320	1.08	1.64	3.1	34.6	107.26
9340	2.04	2.48	1.1	35.5	39.05
9360	1.07	1.61	0.12	36.8	4.416
9380	1.08	1.64	0.11	37.8	4.158
9400	1.23	2.61	0.31	36.7	11.377
9420	2.11	2.86	0.25	37.8	9.45
9440	2.2	2.79	0.15	37.7	5.655
9460	2.33	2.98	0.18	37.4	6.732
9480	2.4	3.01	0.23	37.4	8.602
9500	2.04	2.89	0.19	37.5	7.125
9520	1.07	2.78	0.14	37.4	5.236
9540	1.08	3.03	2.4	36.6	87.84
9560	1.67	3.13	0.37	37.9	14.023
9580	1.74	2.67	0.16	38.1	6.096
9600	1.79	2.51	0.15	37.7	5.655
9620	1.8	2.53	0.06	37.5	2.25
9640	1.23	2.96	0.08	37.2	2.976
9660	2.11	2.81	0.17	37.5	6.375
9680	2.2	2.56	0.03	37.4	1.122
9700	2.33	2.86	0.08	37.3	2.984
9720	2.4	3.04	0.45	37.3	16.785
9740	2.04	2.48	0.09	37.3	3.357
9760	1.07	1.56	4.8	37.1	178.08
9780	1.08	1.58	2.68	38	101.84
9800	1.05	1.48	1.31	36.7	48.077
9820	1.03	1.5	0.33	37.2	12.276
9840	1.07	1.61	0.21	37.3	7.833
9860	1.67	2.78	0.22	37.3	8.206
9880	1.35	2.46	0.45	37.3	16.785
9900	1.41	2.75	0.33	37.2	12.276
9920	1.44	2.88	0.11	37.1	4.081
9940	1.32	2.61	0.15	37.1	5.565
9960	1.44	2.86	0.21	38.2	8.022
9980	1.68	2.79	0.19	38	7.22
10000	1.71	2.98	0.42	37.3	15.666
10020	1.72	3.01	0.31	37.2	11.532

10040	1.56	2.89	0.05	37.3	1.865
10060	1.34	2.78	0.78	37.4	29.172
10080	1.89	3.03	5.11	37.4	191.114
10100	1.91	3.13	6.12	36.4	222.768
10120	1.38	2.67	0.07	35.4	2.478
10140	1.21	2.51	0.15	37.3	5.595
10160	1.22	2.53	0.16	37.2	5.952
10180	1.62	2.96	0.17	36.1	6.137
10200	1.91	3.01	0.18	36.4	6.552
10220	2.11	3.18	0.23	37.3	8.579
10240	2.23	3.56	0.33	37.2	12.276
10260	2.05	3.48	0.13	37.3	4.849
10280	2.05	3.48	3.7	37.2	137.64
10300	1.67	2.78	4.1	37.2	152.52
10320	1.74	2.46	2.56	37.2	95.232
10340	1.79	2.75	0.11	35.2	3.872
10360	1.8	2.88	0.34	37.1	12.614
10380	1.23	2.61	0.31	37	11.47
10400	2.11	2.86	0.05	37	1.85
10420	2.2	2.79	0.78	37	28.86
10440	2.33	2.98	0.11	38.4	4.224
10460	2.4	3.01	0.12	37.1	4.452
10480	2.04	2.89	0.07	37.1	2.597
10500	1.07	2.78	0.15	37.1	5.565
10520	1.08	3.03	0.16	37	5.92
10540	1.67	3.13	0.17	37.1	6.307
10560	1.74	2.67	0.18	37	6.66
10580	1.79	2.51	0.23	37	8.51
10600	1.8	2.53	0.33	36.9	12.177
10620	1.23	2.96	0.13	37	4.81
10640	2.11	2.81	4.1	36.8	150.88
10660	2.2	2.56	2.3	36.9	84.87
10680	2.33	2.86	0.31	37	11.47
10700	2.4	3.04	0.22	35.6	7.832
10720	2.04	2.48	0.14	34.9	4.886
10740	1.07	1.61	0.15	36.9	5.535
10760	1.08	1.64	0.09	37	3.33
10780	2.04	2.48	0.1	37	3.7
10800	1.07	1.61	0.11	37	4.07

10820	1.08	1.64	0.15	37.1	5.565
10840	1.07	1.56	0.21	37.4	7.854
10860	1.08	1.58	8.25	37.1	306.075
10880	1.05	1.48	6.14	35	214.9
10900	1.03	1.5	1.13	35.8	40.454
10920	1.07	1.61	0.36	37.1	13.356
10940	1.67	2.78	0.21	36.9	7.749
10960	1.35	2.46	0.22	37	8.14
10980	1.41	2.75	0.19	37	7.03
11000	1.44	2.88	0.17	36.9	6.273
11020	1.32	2.61	0.21	36.8	7.728
11040	1.44	2.86	0.22	37.1	8.162
11060	1.68	2.79	0.11	37	4.07
11080	1.71	2.98	0.13	36.9	4.797
11100	1.72	3.01	5.34	37.1	198.114
11120	1.56	2.89	1.68	35.9	60.312
11140	1.02	1.61	0.56	36.8	20.608
11160	1.11	1.77	0.12	36.9	4.428
11180	1.33	1.84	0.11	36.9	4.059
11200	1.51	2.13	0.13	36.7	4.771
11220	1.34	2.78	0.65	36.9	23.985
11240	1.89	3.03	0.32	36.8	11.776
11260	1.91	3.13	0.11	36.9	4.059
11280	1.38	2.67	0.13	36.9	4.797
11300	1.21	2.51	0.17	36.8	6.256
11320	1.22	2.53	0.14	35.7	4.998
11340	1.62	2.96	0.21	36.8	7.728
11360	1.91	3.01	10.18	36.8	374.624
11380	2.11	3.18	9.14	35.9	328.126
11400	2.23	3.56	4.17	35.1	146.367
11420	1.05	1.48	1.14	36.8	41.952
11440	1.03	1.5	0.13	36.6	4.758
11460	1.07	1.61	0.36	36.9	13.284
11480	1.67	2.78	0.21	36.9	7.749
11500	1.35	2.46	0.22	36.8	8.096
11520	1.41	2.75	0.19	36.9	7.011
11540	1.44	2.88	0.17	35.9	6.103
11560	1.32	2.61	0.21	36.9	7.749
11580	1.44	2.86	0.22	37.2	8.184

11600	1.68	2.79	0.11	37.1	4.081
11620	1.71	2.98	0.13	37	4.81
11640	1.72	3.01	5.34	36	192.24
11660	1.56	2.89	1.68	35.9	60.312
11680	1.02	1.61	0.56	36.8	20.608
11700	1.11	1.77	0.12	36.9	4.428
11720	1.33	1.84	0.11	36.8	4.048
11740	1.51	2.13	0.13	36.9	4.797
11760	1.34	2.78	0.65	36.8	23.92
11780	1.89	3.03	0.32	36.9	11.808
11800	1.91	3.13	0.11	36.9	4.059
11820	1.38	2.67	0.13	35.8	4.654
11840	1.21	2.51	0.17	35.4	6.018
11860	1.22	2.53	0.14	34.8	4.872
11880	1.62	2.96	0.21	37	7.77
11900	1.91	3.01	8.18	36.8	301.024
11920	2.11	3.18	0.14	36.8	5.152
11940	2.23	3.56	0.17	37	6.29
11960	2.05	3.48	0.21	36.9	7.749
11980	2.05	3.48	0.22	36.8	8.096
12000	1.67	2.78	0.31	35.8	11.098
12020	1.74	2.46	3.33	35.1	116.883
12040	1.79	2.75	1.29	35	45.15
12060	1.8	2.88	0.26	36.9	9.594
12080	1.23	2.61	0.21	36.8	7.728
12100	2.11	2.86	0.2	36.8	7.36
12120	2.2	2.79	0.31	36.8	11.408
12140	2.33	2.98	0.22	36.8	8.096
12160	2.4	3.01	0.41	36.8	15.088
12180	2.04	2.89	0.17	36.8	6.256
12200	1.07	2.78	3.31	36.6	121.146
12220	1.08	3.03	0.07	35.7	2.499
12240	1.67	3.13	0.09	35.1	3.159
12260	1.74	2.67	0.1	36.8	3.68
12280	1.79	2.51	0.11	36.9	4.059
12300	1.8	2.53	0.15	36.7	5.505
12320	1.23	2.96	0.12	36.8	4.416
12340	2.11	2.81	0.13	36.8	4.784
12360	2.2	2.56	0.14	36.8	5.152

12380	2.33	2.86	0.17	36.6	6.222
12400	2.4	3.04	0.13	36.7	4.771
12420	2.04	2.48	0.14	36.7	5.138
12440	1.07	1.61	7.3	36.7	267.91
12460	1.08	1.64	3.1	35.5	110.05
12480	2.04	2.48	1.1	36.7	40.37
12500	1.07	1.61	0.12	36.7	4.404
12520	1.08	1.64	0.11	36.6	4.026
12540	1.23	2.61	0.31	36.5	11.315
12560	2.11	2.86	0.25	36.8	9.2
12580	2.2	2.79	0.15	36.7	5.505
12600	2.33	2.98	0.18	36.7	6.606
12620	2.4	3.01	0.23	36.8	8.464
12640	2.04	2.89	0.19	36.7	6.973
12660	1.07	2.78	0.14	36.7	5.138
12680	1.08	3.03	2.4	36.6	87.84
12700	1.67	3.13	4.37	37.7	164.749
12720	1.74	2.67	0.16	36.7	5.872
12740	1.79	2.51	0.15	36.8	5.52
12760	1.8	2.53	0.06	37	2.22
12780	1.23	2.96	0.08	36.8	2.944
12800	2.11	2.81	0.17	36.8	6.256
12820	2.2	2.56	0.03	36.8	1.104
12840	2.33	2.86	0.08	37.3	2.984
12860	2.4	3.04	0.45	37	16.65
12880	2.04	2.48	0.09	36.4	3.276
12900	1.07	1.56	4.8	36.5	175.2
12920	1.08	1.58	2.68	35.8	95.944
12940	1.05	1.48	1.31	36	47.16
12960	1.03	1.5	0.33	36.6	12.078
12980	1.07	1.61	0.21	36.7	7.707
13000	1.67	2.78	0.22	36.7	8.074
13020	1.35	2.46	0.45	36.7	16.515
13040	1.41	2.75	0.33	37.7	12.441
13060	1.44	2.88	0.11	36.8	4.048
13080	1.32	2.61	0.15	37.7	5.655
13100	1.44	2.86	0.21	37.7	7.917
13120	1.68	2.79	0.19	36.6	6.954
13140	1.71	2.98	0.42	36.6	15.372

13160	1.72	3.01	0.31	36.7	11.377
13180	1.56	2.89	0.05	36.7	1.835
13200	1.34	2.78	5.78	35.9	207.502
13220	1.89	3.03	2.11	35.4	74.694
13240	1.91	3.13	0.12	35	4.2
13260	1.38	2.67	0.07	35.3	2.471
13280	1.21	2.51	0.15	36.8	5.52
13300	1.22	2.53	0.16	36.7	5.872
13320	1.62	2.96	3.17	35.6	112.852
13340	1.91	3.01	0.18	35.8	6.444
13360	2.11	3.18	0.23	35.3	8.119
13380	2.23	3.56	0.33	36.6	12.078
13400	2.05	3.48	0.13	36.6	4.758
13420	2.05	3.48	3.7	36.5	135.05
13440	1.67	2.78	4.1	36.5	149.65
13460	1.74	2.46	2.56	35.7	91.392
13480	1.79	2.75	0.11	36.6	4.026
13500	1.8	2.88	0.34	36.6	12.444
13520	1.23	2.61	0.31	36.5	11.315
13540	2.11	2.86	0.05	36.5	1.825
13560	2.2	2.79	0.78	36.4	28.392
13580	2.33	2.98	0.11	36	3.96
13600	2.4	3.01	3.12	37.2	116.064
13620	2.04	2.89	0.07	36.2	2.534
13640	1.07	2.78	0.15	36.1	5.415
13660	1.08	3.03	0.16	36.9	5.904
13680	1.67	3.13	0.17	36	6.12
13700	1.74	2.67	0.18	36.3	6.534
13720	1.79	2.51	0.23	36.4	8.372
13740	1.8	2.53	0.33	36.9	12.177
13760	1.23	2.96	0.13	36.8	4.784
13780	2.11	2.81	14.1	36.1	509.01
13800	2.2	2.56	2.3	35	80.5
13820	2.33	2.86	0.31	35.8	11.098
13840	2.4	3.04	0.22	36.5	8.03
13860	2.04	2.48	0.14	36.5	5.11
13860	1.07	1.61	0.15	36.5	5.475
13900	1.08	1.64	0.09	36.6	3.294
13920	2.04	2.48	3.1	36.6	113.46

13940	1.07	1.61	1.11	36.6	40.626
13960	1.08	1.64	0.15	36.6	5.49
13980	1.07	1.56	0.21	36.5	7.665
14000	1.08	1.58	0.25	36.6	9.15
14020	1.05	1.48	0.14	36.6	5.124
14040	1.03	1.5	4.13	36.4	150.332
14060	1.07	1.61	0.36	37.5	13.5
14080	1.67	2.78	0.21	36.5	7.665
14100	1.35	2.46	8.22	35.5	291.81
14120	1.41	2.75	4.19	35.5	148.745
14140	1.44	2.88	0.07	35.5	2.485
14160	1.32	2.61	0.11	35.5	3.905
14180	1.44	2.86	0.02	36.5	0.73
14200	1.68	2.79	0.01	36.4	0.364
14220	1.71	2.98	11.13	34.3	381.759
14240	1.72	3.01	5.34	34.4	183.696
14260	1.56	2.89	1.68	35.9	60.312
14280	1.02	1.61	0.56	36.5	20.44
14300	1.11	1.77	0.12	36.3	4.356
14320	1.33	1.84	0.11	36.2	3.982
14340	1.51	2.13	3.13	36.3	113.619
14360	1.34	2.78	1.65	35.4	58.41
14380	1.89	3.03	0.32	35.6	11.392
14400	1.91	3.13	0.11	34.2	3.762
14420	1.38	2.67	0.13	36.4	4.732
14440	1.21	2.51	0.17	36.4	6.188
14460	1.22	2.53	0.14	36.4	5.096
14480	1.62	2.96	0.21	36.4	7.644
14500	1.67	3.13	4.09	36.5	149.285
14520	1.74	2.67	2.1	36.6	76.86
14540	1.79	2.51	0.11	36.5	4.015
14560	1.8	2.53	0.15	36.5	5.475
14580	1.23	2.96	0.12	36.5	4.38
14600	2.11	2.81	0.13	36.4	4.732
14620	2.2	2.56	0.14	36.4	5.096
14640	2.33	2.86	0.17	36.5	6.205
14660	2.4	3.04	0.13	36.4	4.732
14680	2.04	2.48	0.14	36.4	5.096
14700	1.07	1.61	7.3	36.6	267.18

14720	1.08	1.64	3.1	35.1	108.81
14740	2.04	2.48	1.1	35.6	39.16
14760	1.07	1.61	0.12	36.3	4.356
14780	1.67	3.13	0.37	36.4	13.468
14800	1.74	2.67	0.16	36.4	5.824
14820	1.79	2.51	0.15	36.5	5.475
14840	1.8	2.53	0.06	36.5	2.19
14860	1.23	2.96	0.08	36.6	2.928
14880	2.11	2.81	0.17	36.4	6.188
14900	2.2	2.56	0.03	36.5	1.095
14920	2.33	2.86	0.08	36.4	2.912
14940	2.04	2.48	0.14	35.4	4.956
14960	1.07	1.61	7.3	35.3	257.69
14980	1.08	1.64	3.1	36.4	112.84
15000	2.04	2.48	1.1	36.3	39.93
15020	1.07	1.61	0.12	36.3	4.356
15040	1.67	3.13	0.37	36.2	13.394
15060	1.74	2.67	0.16	36.4	5.824
15080	2.2	2.56	0.03	36.3	1.089

APPENDIX C

THE CODE IMPLEMENTED IN THE DIGITAL CIRCUIT

```
*************************************************************
//    System Clock                                    : 1MHz
//    Software                                        : Atmel
Studio 6.0
//    LCD Data Interfacing                       : 8-Bit
*************************************************************


#include<avr/io.h>
#include<avr/interrupt.h>
#define          F_CPU          1000000
#include<util/delay.h>
#define          LCD_DATA_DR         DDRB
#define          LCD_DATA_PORT         PORTB
#define          LCD_CONT_DR           DDRD
#define          LCD_CONT_PORT         PORTD
#define          LCD_RS                     PD3
#define          LCD_RW                     PD2
#define          LCD_EN                     PD1

#include "lcd.h"

float torquesensorinput;
float output;
float val;
float gain;
float gainADC;
float gainLCD;

int main(void)
{


      //    DDRB=0xff;
      //    DDRD=0x07;

      lcd_init();
      //lcd_string_write(first_row_lcd_display);
      //lcd_command_write(0xc0);
      //lcd_string_write(second_row_lcd_display);
      //lcd_number_write(1000,2);

      //setting OCR1A as output
      DDRD |= 1<<PINB5;

      //Waveform generation mode as fast PWM
      //no prescaler CS10
      //COM1A1 n COM1A0 inverted mode
```

```
	TCCR1A |= 1<<WGM11 | 1<<COM1A1 ;//| 1<<COM1A0;
	TCCR1B |= 1<<WGM12 | 1<<WGM13 | 1<<CS10;
	ICR1 = 3333;

	//Skeleton code
	//1. Configure the ADC
	//2. Enable interrupts function in ADC
	//3. 8-bit or 10-bit results
	//4. Enable a prescaler - determined by the internal/external
clock
	//5. Turn on the ADC feature
	//6. Start the first conversion
	//7. Enable the global interrupts which is sei();


	//prescaler 16
	ADCSRA |= (1<<ADPS2);

	//AVCC with external capacitor at AREF pin
	ADMUX |= (1<<REFS0) | (1<<REFS1);

	//8-bit or 10-bit results ADLAR = 0 for 10 bits, 1 for 8 bits.
	//ADMUX |= 0<<ADLAR;

	//Enable interrupts function in ADC
	ADCSRA |= 1<<ADIE;

	//Turn on the ADC feature
	ADCSRA |= 1<<ADEN;

	//Enable global interrupts
	sei();

	//Start the first conversion
	ADCSRA |= 1<<ADSC;

	while(1)
	{
		// gain is calculated here from ADC1
		// algorithm for LCD display gain numerical
		gain = (gainADC/103);
		gainLCD = gain*100;

		lcd_string_write("Vin:");
		lcd_number_write(torquesensorinput,10);
		//lcd_command_write(0xc0);
		lcd_string_write(" Vout:");
		lcd_number_write(output,10);
		lcd_command_write(0xc0);
		lcd_string_write("Gain: ");
		lcd_number_write(gainLCD,10);
		lcd_string_write("%");
		_delay_ms(100);
		lcd_command_write(LCD_CLEARDISPLAY);
```

```
            if(torquesensorinput<103) //operating below lower
threshold region
            {
                  val = 0;
                  OCR1A = val;
                  output = 0;
                  _delay_ms(100);
            }
            else if((torquesensorinput>=103) &&
(torquesensorinput<=(737/gain)))
            {

                  output = gain*torquesensorinput;
                  val = (3333*(output/1023));;    //?? Max pulse width
is 20000, check it??
                  OCR1A = val;
                  _delay_ms(100);
            }
            else
            {
                  output = 737;
                  val = 2340;
                  OCR1A = val;
                  _delay_ms(100);
            }
      }
}

ISR(ADC_vect)
{
      uint8_t theLowADC = ADCL;
      uint16_t theTenBitResults = ADCH<<8 | theLowADC;


switch (ADMUX)
{
      case 0xC0:
      torquesensorinput= theTenBitResults;
      ADMUX = 0xC1;
      break;
      case 0xC1:
      gainADC=theTenBitResults;
      ADMUX = 0xC0;
      break;
      default:
      //Default code
      break;
}


      ADCSRA |= 1<<ADSC;
}
```

APPENDIX D

PCB SCHEMATIC

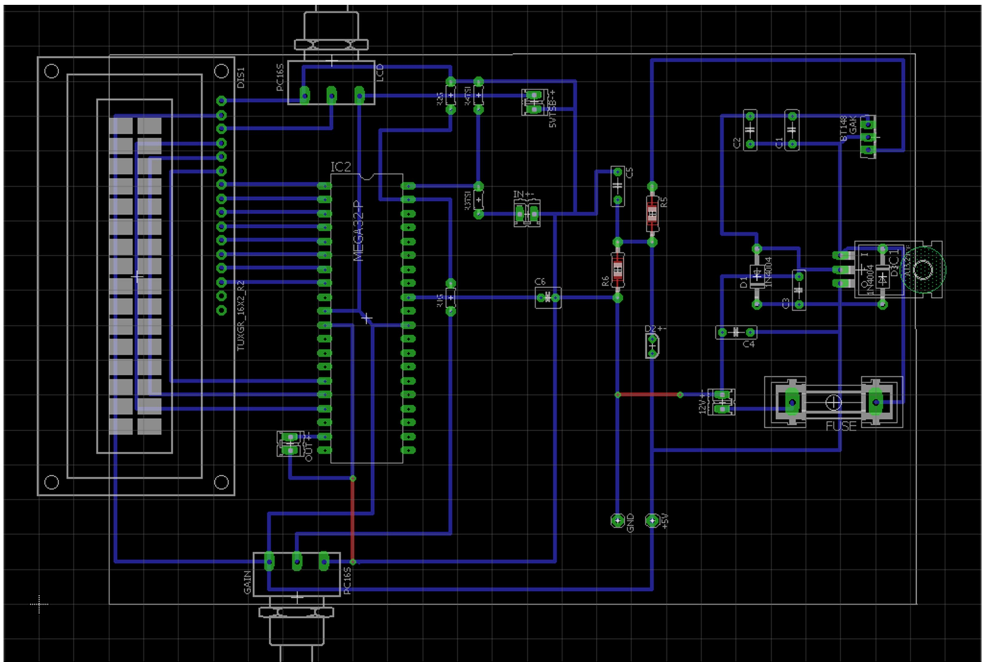


Fig 1: Top View of the PBC design

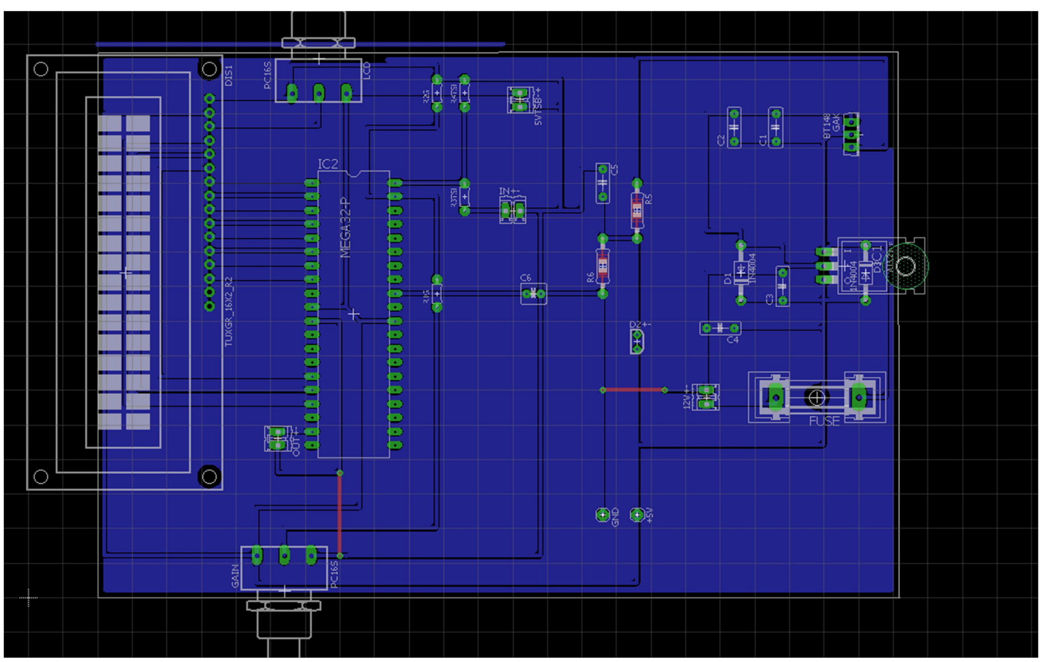


Fig 2: Top view with masking

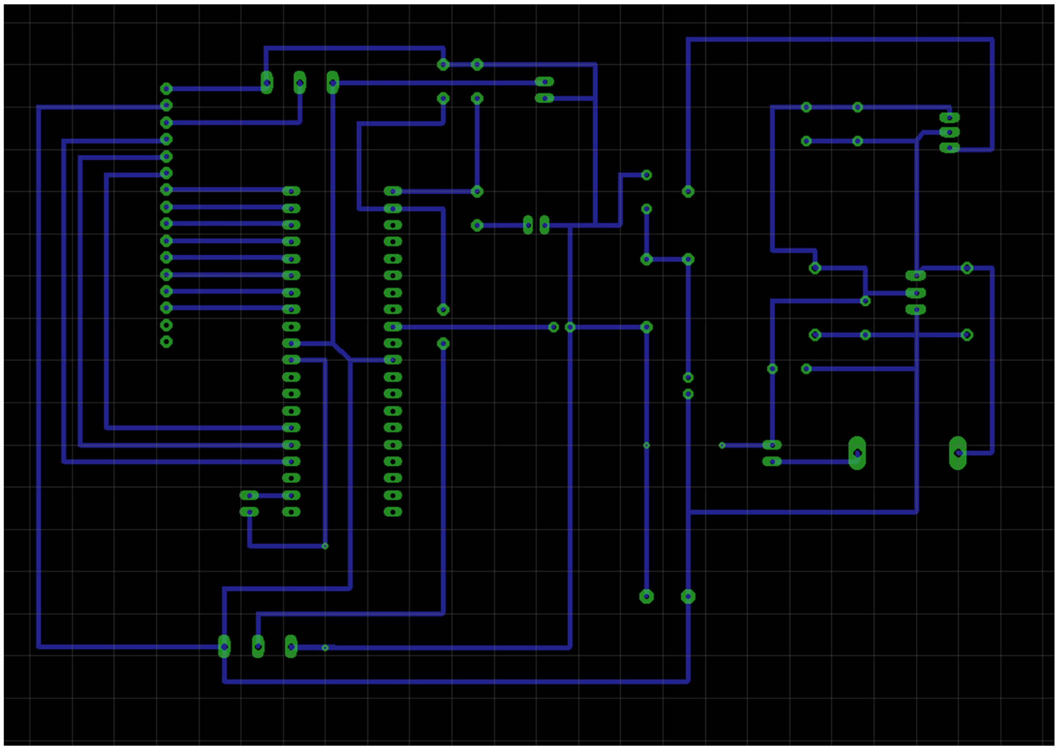


Fig 3: Bottom view of the PCB routing

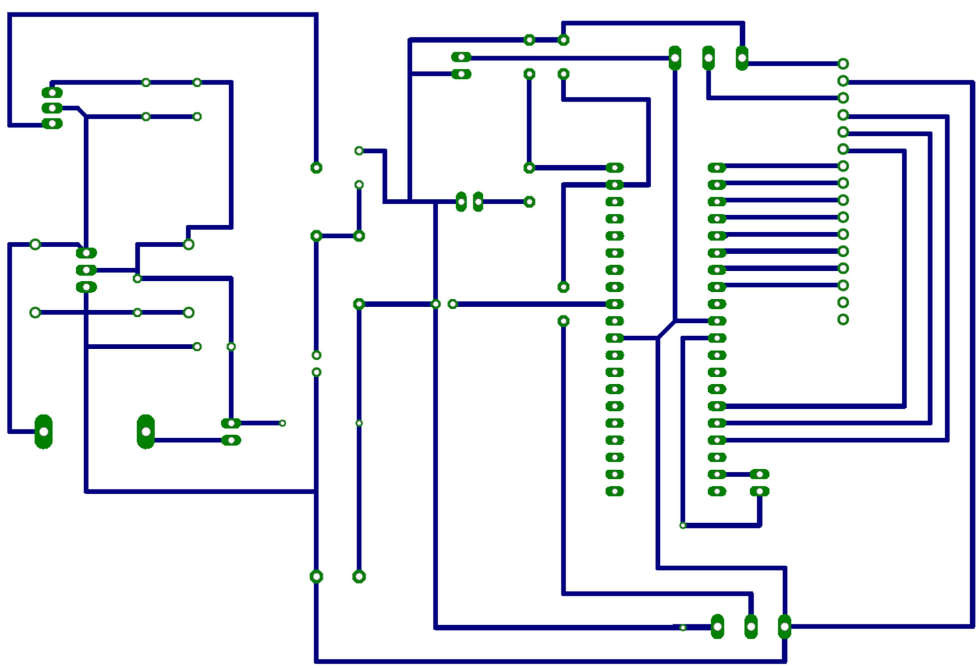


Fig 4: Bottom view of the PCB routing mirrored