SMART WHEELCHAIR

Dissertation submitted in the partial fulfilment of the requirement for the degree of

BACHELOR OF ENGINEERING IN ELECTRICAL AND ELECTRONICS

By

Saurabh Patil (P.R NO.: 201610614)

Kaushal Sawant (P.R NO.: 201711962)

Prajyot Naik (P.R NO.: 201610598)

Darshan Sangekar (P.R NO.: 201704889)

Under the guidance

Of

Dr. Anant Naik

Professor

GOA COLLEGE OF ENGINEERING

FARMAGUDI PONDA-GOA 2019-2020

Goa College of Engineering

Department of Electrical and Electronics Engineering

Farmagudi, Ponda-Goa

CERTIFICATE

This is to certify that the following students of Goa College of Engineering, Department of Electrical and Electronics Engineering have successfully completed the project work titled “**SMART WHEELCHAIR**” during the academic year 2019-2020 :

Saurabh Patil (P.R NO.: 201610614)

Kaushal Sawant (P.R NO.: 201711962)

Prajyot Naik (P.R NO.: 201610598)

Darshan Sangekar (P.R NO.: 201704889)

Internal examiner

Dr. Anant Naik

Professor

Electrical & Electronics Dept.
Engineering

Head of Department

Dr. G.R. Kunkolienkar

Electrical & Electronics Dept.

Goa College of Goa College
of Engineering

APPROVAL SHEET

This is to certify that

Saurabh Patil (P.R NO.: 201610614)

Kaushal Sawant (P.R NO.: 201711962)

Prajyot Naik (P.R NO.: 201610598)

Darshan Sangekar (P.R NO.: 201704889)

Have been admitted to the candidacy of the degree (Electrical and Electronics) in June 2020 and they have undertaken the dissertation entitled “**SMART WHEELCHAIR**” which is approved for the degree of Bachelor of Engineering in Electrical and Electronics at Goa College of Engineering, Farmagudi under Goa University as it is found satisfactory.

Examiner 1:

Examiner2

ACKNOWLEDGEMENT

We are deeply indebted to our guide Dr. Anant Naik for allowing us to carry out this project under his supervision. He has given us confidence to take up this project and is a pillar of support that we have banked in times of difficulty.

We express our deep sense of gratitude towards Dr. G.R. Kunkolienkar, Head of Department, and Electrical & Electronics Engineering Department to make this Bachelor course worthwhile through his diligent efforts.

Sincere thanks to our Principal, Dr. Krupashankara M.S, for facilitating the resources and providing all the encouragement.

We would also like to thank our Project coordinator Prof .Suneeta Raykar for the timely help as and when required.

We also thank our parents, family members and friends along with the almighty without whom this work would not have seen the light of the day.

ABSTRACT

Smart Wheel Chair is mechanically controlled device designed to have self-mobility with the help of the user command. This reduces the user's human effort and force to drive the wheels for wheelchair .Furthermore it also provides an opportunity for visually or physically impaired persons to move from one place to another. The wheelchair is also provided with obstacle detection system which reduces the chance of collision while on the journey.

Smart wheelchair has gained a lot of interests in the recent times. These devices are useful especially in transportation from one place to another. The machines can also be used in old age homes where the old age persons have difficulty in their movements. The devices serve as a boon for those who have lost their mobility.

Different types of smart wheelchair have been developed in the past but the new generations of wheelchairs are being developed and used which features the use of artificial intelligence and hence leaves a little to tinker about to the user who uses the wheel chair. The project also aims to build a similar wheel chair prototype which would have a features that would help the user on his/her movement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background

Several studies have shown that both children and adults benefit substantially from access to a means of independent mobility, including power wheelchairs, manual wheelchairs, scooters, and walkers. Independent mobility increases vocational and educational opportunities, reduces dependence on caregivers and family members, and promotes feelings of self-reliance. For young children, independent mobility serves as the foundation for much early learning. No ambulatory children lack access to the wealth of stimuli afforded self-ambulating children. This lack of exploration and control often produces a cycle of deprivation and reduced motivation that leads to learned helplessness. For adults, independent mobility is an important aspect of self-esteem and plays a pivotal role in “aging in place.” For example, if older people find it increasingly difficult to walk or wheel themselves to the commode, they may do so less often or they may drink less fluid to reduce the frequency of urination. If they become unable to walk or wheel themselves to the commode and help is not routinely available in the home when needed, a move to a more enabling environment (e.g., assisted living) may be necessary. Mobility limitations are the leading cause of functional limitations among adults, with an estimated prevalence of 40 per 1,000 persons age 18 to 44 and 188 per 1,000 at age 85 and older.

Mobility difficulties are also strong predictors of activities of daily living (ADL) and instrumental ADL disabilities because of the need to move to accomplish many of these activities. In addition, impaired mobility often results in decreased opportunities to socialize, which leads to social isolation, anxiety, and depression. For example, 31 percent of persons with major mobility difficulties reported being frequently depressed or anxious, compared with only 4 percent of persons without mobility difficulties. While the needs of many individuals with disabilities can be satisfied with traditional manual or powered wheelchairs, a segment of the disabled community finds it difficult or impossible to use wheelchairs independently. This population includes, but is not limited to, individuals with low vision, visual field reduction, spasticity, tremors, or cognitive deficits. These individuals often lack independent mobility and rely on a caregiver to push them in a manual wheelchair.

To accommodate this population, several researchers have used technologies originally developed for mobile robots to create “smart wheelchairs.” A smart wheelchair typically consists of either a standard power wheelchair to which a computer and a collection of sensors have been added or a mobile robot base to which a seat has been attached. Smart wheelchairs have been designed that provide navigation assistance to the user in a number of different ways, such as assuring collision-free travel, aiding the performance of specific tasks (e.g., passing through doorways), and autonomously transporting the user between locations. A recent survey indicated that clinicians have a strong desire for the services that a smart wheelchair can offer. Significant survey results included:

• Clinicians indicated that 9 to 10 percent of patients who receive power wheelchair training find it extremely difficult or impossible to use the wheelchair for ADL.

• When asked specifically about steering and manoeuvring tasks, the percentage of patients who reported these tasks difficult or impossible jumped to 40 percent.

• Eighty-five percent of responding clinicians reported seeing some number of patients each year who cannot use a power wheelchair because they lack the requisite motor skills, strength, or visual acuity. Of these clinicians, 32 percent (27% of all respondents) reported seeing at least as many patients who cannot use a power wheelchair as who can.

• Nearly half of patients unable to control a power wheelchair by conventional methods would benefit from an automated navigation system according to the clinicians who treat them. Smart wheelchairs have been the subject of research since the early 1980s and have been developed on four continents.

1.2 Usefulness of the project

1. Report of world health organization states that about 10% of the global population, i.e. about 650 million people, have disabilities of these some 10% require a wheelchair for their mobility. In 2003, it was estimated that 20 million people of those requiring wheelchair did not have one.
2. The old age people also require a wheelchair for their mobility.
3. The wheelchair will also help the differently-abled (adults and children), and the senior citizens in mobility, become independent, comfort, enjoy activities and increase socialization.

CHAPTER 2: PROJECT OBJECTIVES

2.1 Objective

Aim of this project is to design a wheelchair prototype that will include features such as

- Head/hand movement control (accelerometer) control.
- It will include obstacle avoider.
- Mobile controlled (camera) input in case you are away from the wheelchair.
- A panic button that will send messages to your caretakers that you need help.

An important objective of the project will be to implement the above-mentioned features in the cheapest way possible.

Fig 2.1 smart wheelchair

2.2 Project Overview

Recent development promises scope in making intelligent wheelchairs for handicapped one. In earlier days, Automatic wireless gesture control wheelchair has gained popularity due to advanced technology. And also there is need of advanced wheelchair due to today's fast working world, no one has free time for the handicapped person, every person is busy in their personal life. After paralysis or physical disability person require hand recognition and the wheelchair is most common need of gesture recognition. A disable individual which is physically partial paralysis feels more convenient to survive in world also to move anywhere using with the help of chair. This wheelchair gives valuable physical support to disabled person. In this project accelerometer sensor is used which is main component of system and used to measure static as well as dynamic acceleration. Accelerometer sensor measures gravity force and it is related to the displacement.

CHAPTER 3: LITERATURE SURVEY

3.1 Accelerometer (MPU6050)

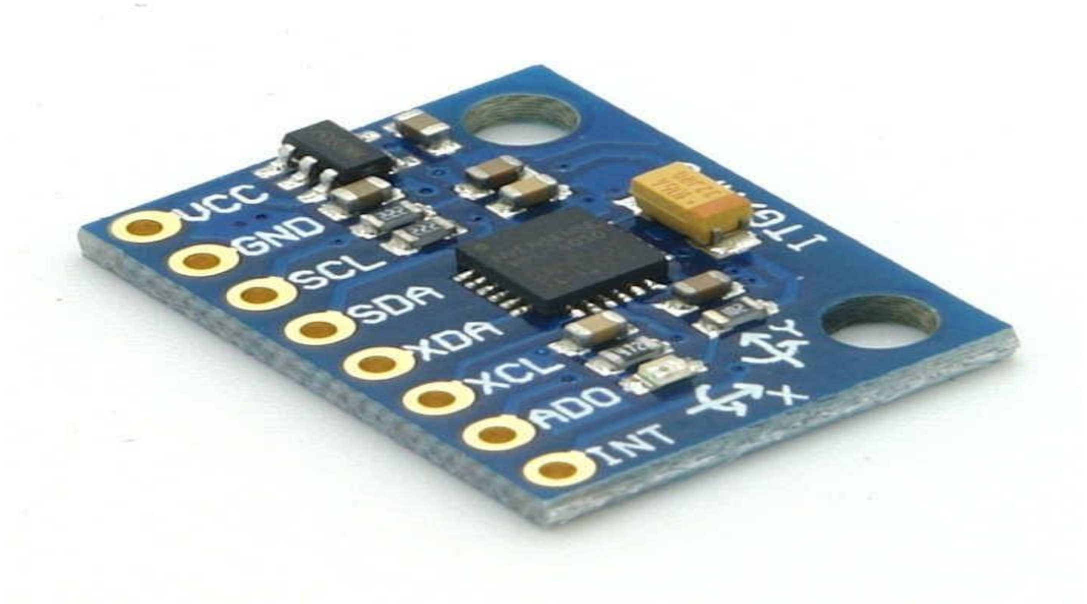

Fig 3.1 MPU6050

3.1.1 Introduction

MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package. Also, it has additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers.

It has Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc. If 3-axis Magnetometer is connected to auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.

3-Axis Gyroscope

The MPU6050 consist of 3-axis Gyroscope with Micro Electro Mechanical System (MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.

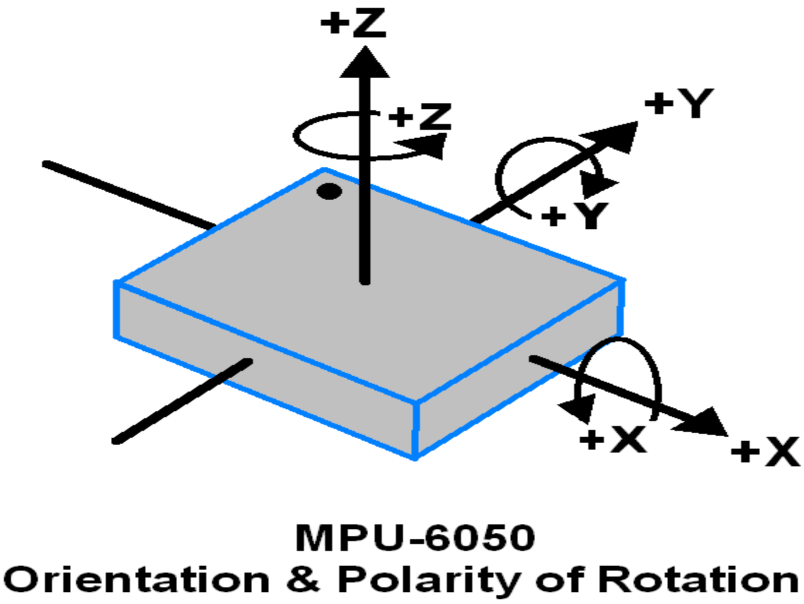

Fig 3.2 Orientation and Polarity of rotation of MPU6050

3.1.2 Working Principle

- When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.

- The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate.

- This voltage is digitized using 16-bit ADC to sample each axis.

- The full-scale range of output are +/- 250, +/- 500, +/- 1000, +/- 2000.

- It measures the angular velocity along each axis in degree per second unit.

3-Axis Accelerometer

The MPU6050 consist 3-axis Accelerometer with Micro Electro Mechanical (MEMs) technology. It used to detect angle of tilt or inclination along the X, Y and Z axes as shown in below figure.

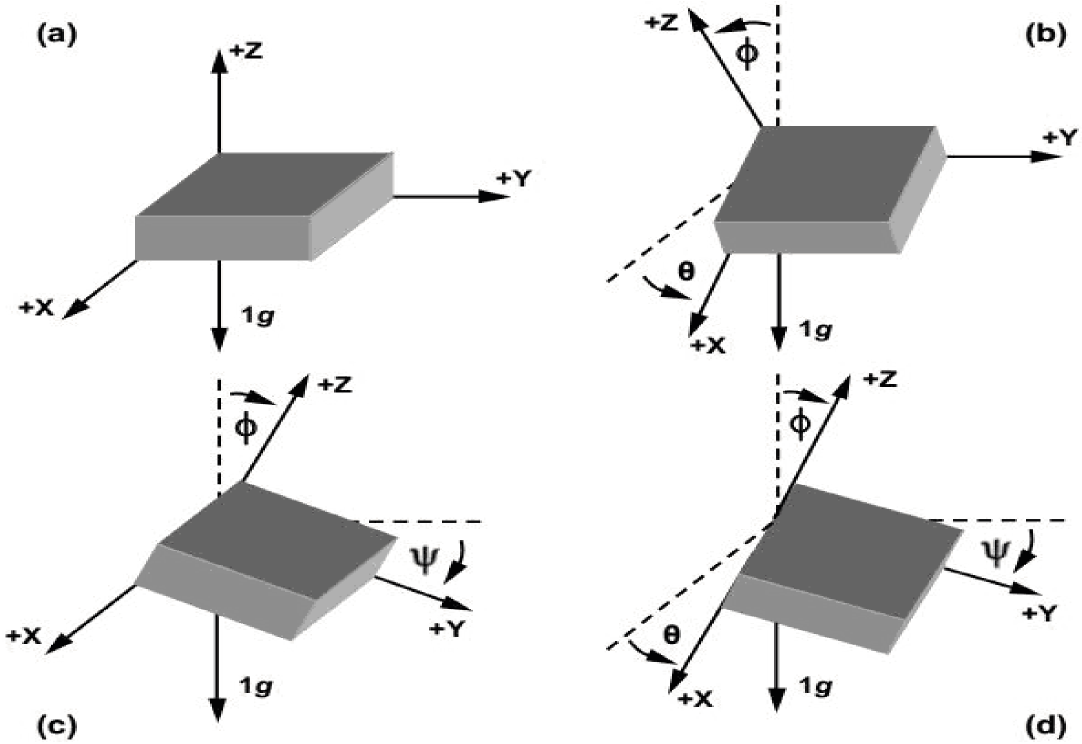

Fig 3.3 Displacement of accelerometer in different axis

- Acceleration along the axes deflects the movable mass.

- This displacement of moving plate (mass) unbalances the differential capacitor which results in sensor output. Output amplitude is proportional to acceleration.

- 16-bit ADC is used to get digitized output.

- The full-scale range of acceleration are +/- 2g, +/- 4g, +/- 8g, +/- 16g.

- It measured in g (gravity force) unit.

- When device placed on flat surface it will measure 0g on X and Y axis and +1g on Z axis.

DMP (Digital Motion Processor)

The embedded Digital Motion Processor (DMP) is used to compute motion processing algorithms. It takes data from gyroscope, accelerometer and additional 3rd party sensor such as magnetometer and processes the data. It provides motion data like roll, pitch, yaw angles, landscape and portrait sense etc. It minimizes the processes of host in computing motion data. The resulting data can be read from DMP registers.

On-chip Temperature Sensor

On-chip temperature sensor output is digitized using ADC. The reading from temperature sensor can be read from sensor data register.

3.1.3 Pin Diagram:

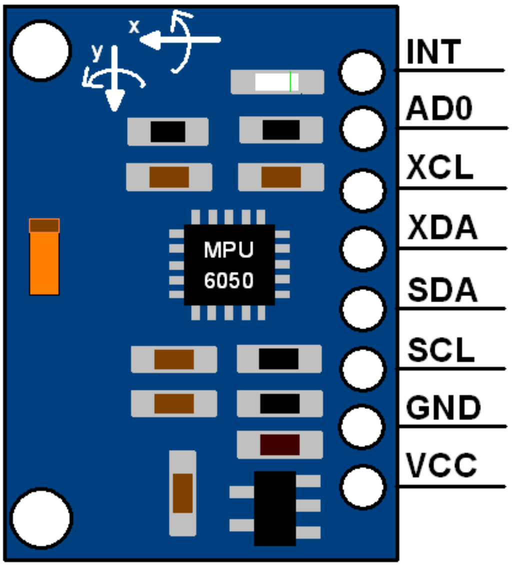

Fig 3.4 Pin diagram of MPU6050

The MPU-6050 module has 8 pins,

INT: Interrupt digital output pin.

AD0: I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.

XCL: Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.

XDA: Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.

SCL: Serial Clock pin. Connect this pin to microcontrollers SCL pin.

SDA: Serial Data pin. Connect this pin to microcontrollers SDA pin.

GND: Ground pin. Connect this pin to ground connection.

VCC: Power supply pin. Connect this pin to +5V DC supply.

3.1.4 Calculations:

Gyroscope and accelerometer sensor data of MPU6050 module consists of 16-bit raw data in 2's complement form.

Temperature sensor data of MPU6050 module consists of 16-bit data (not in 2's complement form).

Now suppose we have selected,

- Accelerometer full scale range of +/- 2g with Sensitivity Scale Factor of 16,384 LSB (Count)/g.

- Gyroscope full scale range of +/- 250 °/s with Sensitivity Scale Factor of 131 LSB (Count)/°/s.

Then,

To get sensor raw data, we need to first perform 2's complement on sensor data of Accelerometer and gyroscope.

After getting sensor raw data we can calculate acceleration and angular velocity by dividing sensor raw data with their sensitivity scale factor as follows,

Accelerometer values in g (g force)

Acceleration along the X axis = (Accelerometer X axis raw data/16384) g.

Acceleration along the Y axis = (Accelerometer Y axis raw data/16384) g.

Acceleration along the Z axis = (Accelerometer Z axis raw data/16384) g.

Gyroscope values in °/s (degree per second)

Angular velocity along the X axis = (Gyroscope X axis raw data/131) °/s.

Angular velocity along the Y axis = (Gyroscope Y axis raw data/131) °/s.

Angular velocity along the Z axis = (Gyroscope Z axis raw data/131) °/s.

Temperature value in °/c (degree per Celsius)

Temperature in degrees C = ((temperature sensor data)/340 + 36.53) °/c.

For example,

Suppose, after 2’ complement we get accelerometer X axes raw value = +15454

Then Ax = +15454/16384 = 0.94 g.

3.2 GSM Module (SIM900A)

3.2.1 Description:

SIM900A Modem is built with Dual Band GSM/GPRS based SIM900A modem from SIMCOM. It works on frequencies 900/ 1800 MHz SIM900A can search these two bands automatically. The frequency bands can also be set by AT Commands. The baud rate is configurable from 1200-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. SIM900A is an ultra-compact and reliable wireless module. This is a complete GSM/GPRS module in a SMT type and designed with a very powerful single-chip processor integrating AMR926EJ-S core, allowing you to benefit from small dimensions and cost-effective solutions.

A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.

2. Read, add, search phonebook entries of the SIM.

3. Make, Receive, or reject a voice call.

The MODEM needs AT commands, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM and GPRS cellular network.

3.2.2 Specification:

1. Dual-Band 900/ 1800 MHz

2. GPRS multi-slot class 10/8GPRS mobile station class B

3. Control via AT commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands)

4. Supply voltage range: 5V

5. Operation temperature: -40°C to +85 °

6. Current Range: Standby: 8mA |Ideal:60mA| Transmission: 180mA

   Reception: 320mA|Peak:2000Ma

7, Baud rate: 9600b/s

Fig 3.5 SIM 900A

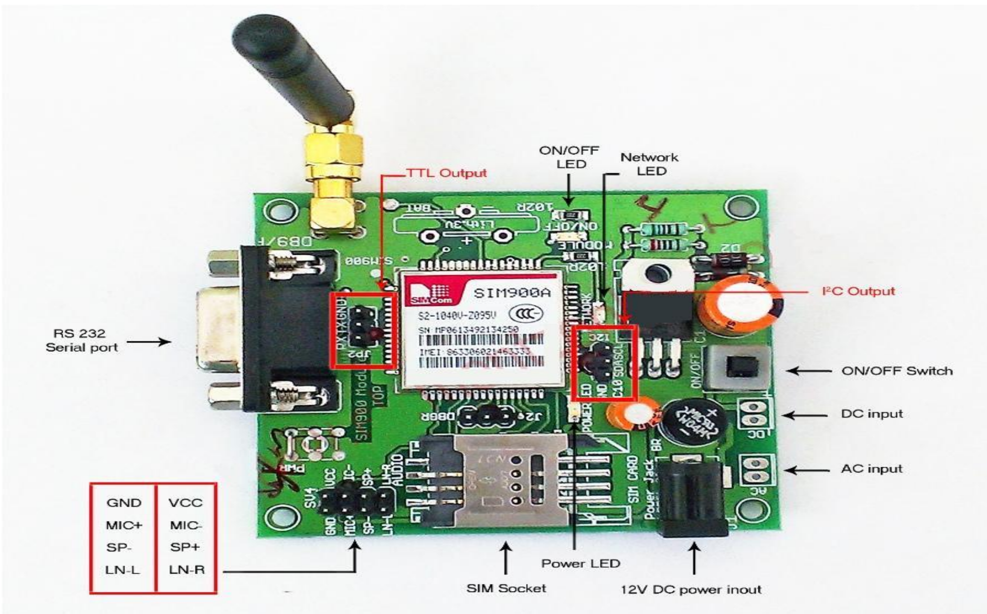

Figure 3.6 Pin Diagram Global system Mobile communication

3.2.3 Features:

Improved spectrum efficiency

International roaming

Compatibility with integrated services digital network (ISDN)

Support for new services.

SIM phonebook management

Fixed dialling number (FDN)

Real time clock with alarm management

High-quality speech

Uses encryption to make phone calls more secure

Short message service (SMS)

3.3 Wireless RF Module (NRF24L01)

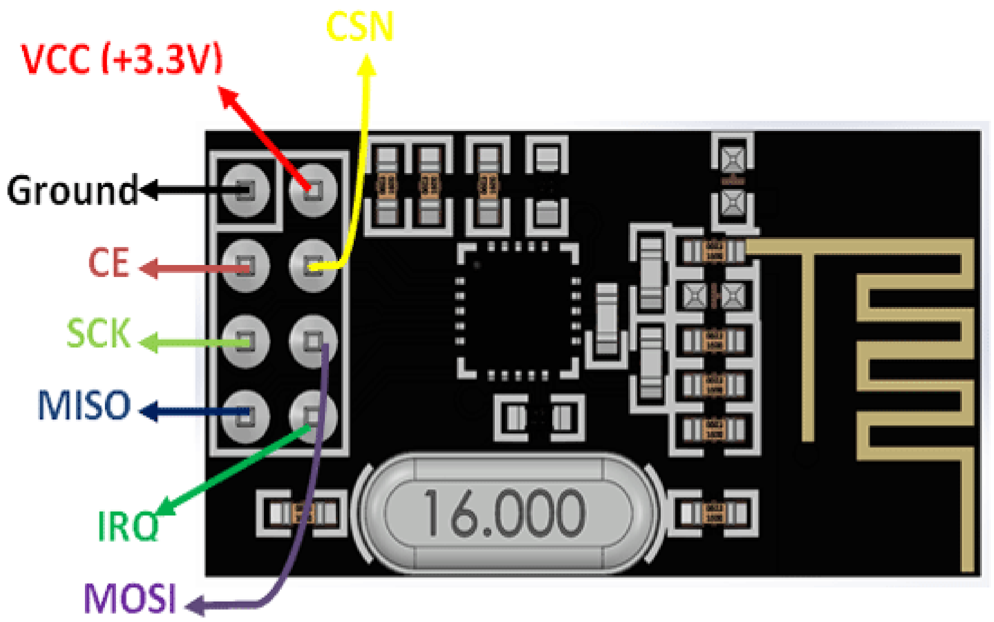

Fig 3.7 NRF24L01

3.3.1 Description:

The nRF24L01 is a wireless transceiver module, meaning each module can both send as well as receive data. They operate in the frequency of 2.4GHz, which falls under the ISM band and hence it is legal to use in almost all countries for engineering applications. The modules when operated efficiently can cover a distance of 100 meters (200 feet) which makes it a great choice for all wireless remote controlled projects.

The module operates at 3.3V hence can be easily used with 3.2V systems or 5V systems. Each module has an address range of 125 and each module can communicate with 6 other modules hence it is possible to have multiple wireless units communicating with each other in a particular area. Hence mesh networks or other types of networks are possible using this module.

3.3.2 Pin Configuration:

Pin Number	Pin Name	Abbreviation	Function
1	Ground	Ground	Connected to the Ground of the system
2	VCC	Power	Powers the module using 3.3V
3	CE	Chip Enable	Used to enable SPI communication
4	CSN	Chip Select Not	This pin has to be kept high always, else it will disable the SPI
5	SCK	Serial Clock	Provides the clock pulse using which the SPI communication works
6	MOSI	Master Out Slave In	Connected to MOSI pin of MCU, for the module to receive data from the MCU
7	MISO	Master In Slave Out	Connected to MISO pin of MCU, for the module to send data from the MCU

8	IRQ	Interrupt	It is an active low pin and is used only if interrupt is required

Table 3.1 pin configuration of NRF24L01

3.3.3 Features:

2.4GHz RF transceiver Module

Operating Voltage: 3.3V

Nominal current: 50mA

Range: 50 – 200 feet

Operating current: 250mA (maximum)

Communication Protocol: SPI

Baud Rate: 250 kbps - 2 Mbps.

Channel Range: 125

Maximum Pipelines/node: 6

Low cost wireless solution

3.3.4 Applications:

Wireless Control application

Mesh Networks

RF Remote Controllers

Connected devices

3.3.5 Working Principle:

The nRF24L01+ transceiver module transmits and receives data on a certain frequency called Channel. Also in order for two or more transceiver modules to communicate with each other, they need to be on the same channel. This channel could be any frequency in the 2.4 GHz ISM band or to be more precise, it could be between 2.400 to 2.525 GHz (2400 to 2525 MHz).

Each channel occupies a bandwidth of less than 1MHz. This gives us 125 possible channels with 1MHz spacing. So, the module can use 125 different channels which give a possibility to have a network of 125 independently working modems in one place.

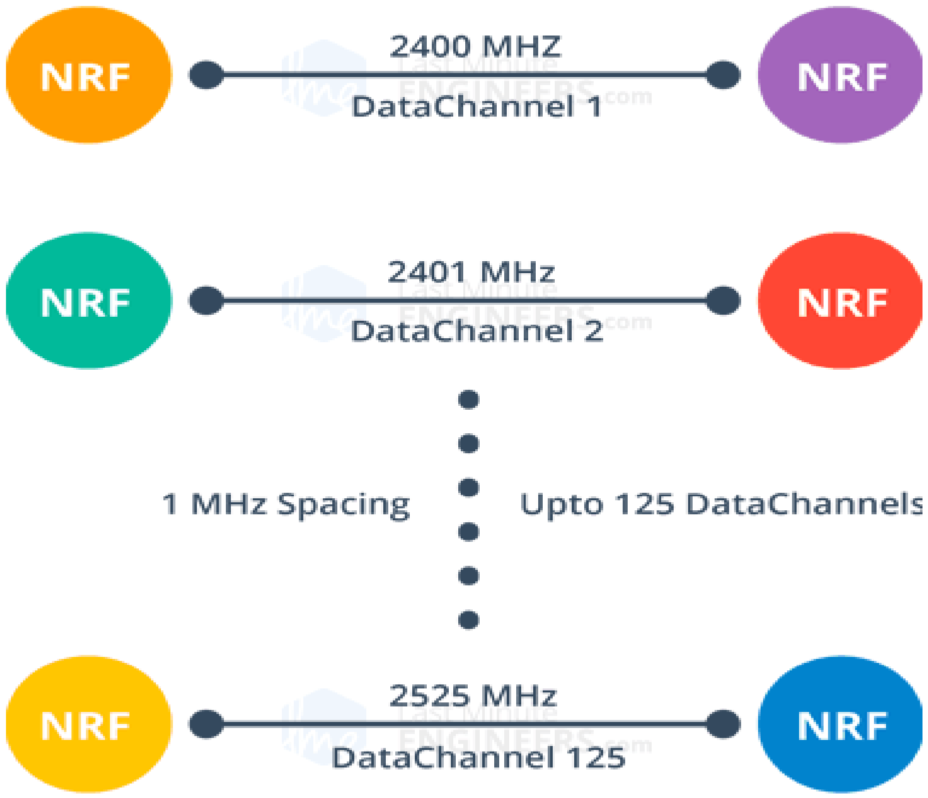

Fig 3.8 Communication between two NRF24L01 modules

The channel occupies a bandwidth of less than 1MHz at 250kbps and 1Mbps air data rate. However at 2Mbps air data rate, 2MHz bandwidth is occupied (wider than the resolution of RF channel frequency setting). So, to ensure non-overlapping channels and reduce cross-talk in 2Mbps mode, you need to keep 2MHz spacing between two channels.

RF channel frequency of your selected channel is set according to the following formula:

Freq (Selected) = 2400 + CH (Selected)

For example, if you select 108 as your channel for data transmission, the RF channel frequency of your channel would be 2508MHz (2400 + 108)

3.4 Ultrasonic Module (HC-SR04)

3.4.1 Introduction:

The ultrasonic sensor works on the principle of SONAR and RADAR system which is used to determine the distance to an object.

An ultrasonic sensor generates the high-frequency sound (ultrasound) waves. When this ultrasound hits the object, it reflects as echo which is sensed by the receiver. By measuring the time required for the echo to reach to the receiver, we can calculate the distance. This is the basic working principle of Ultrasonic module to measure distance.

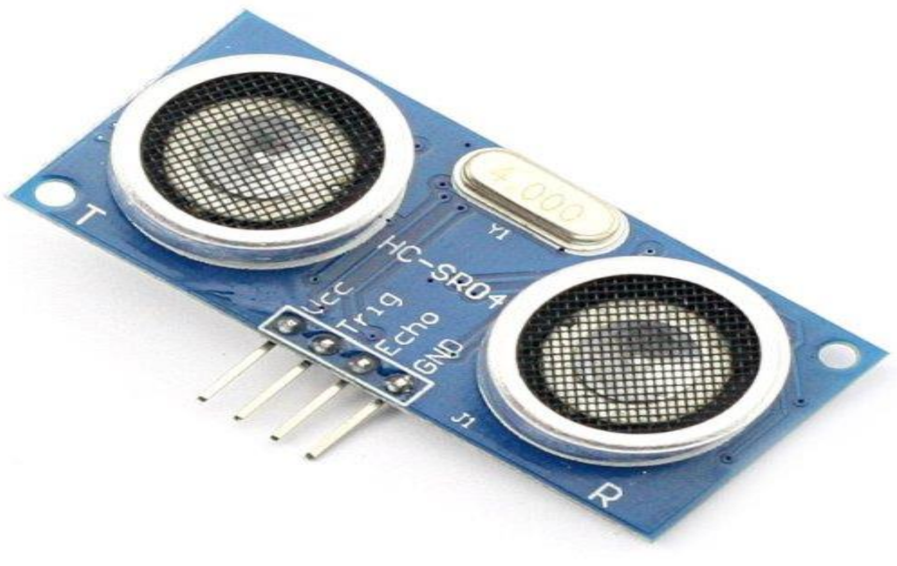

Fig 3.9 HC-SR04

HC-SR-04 has an ultrasonic transmitter, receiver and control circuit. In ultrasonic module HCSR04, we have to give trigger pulse, so that it will generate ultrasound of frequency 40 kHz. After generating ultrasound i.e. 8 pulses of 40 kHz, it makes echo pin high. Echo pin remains high until it does not get the echo sound back. So the width of echo pin will be the time for sound to travel to the object and return back. Once we get the time we can calculate distance, as we know the speed of sound.HC-SR04 can measure up to range from 2 cm - 400 cm.

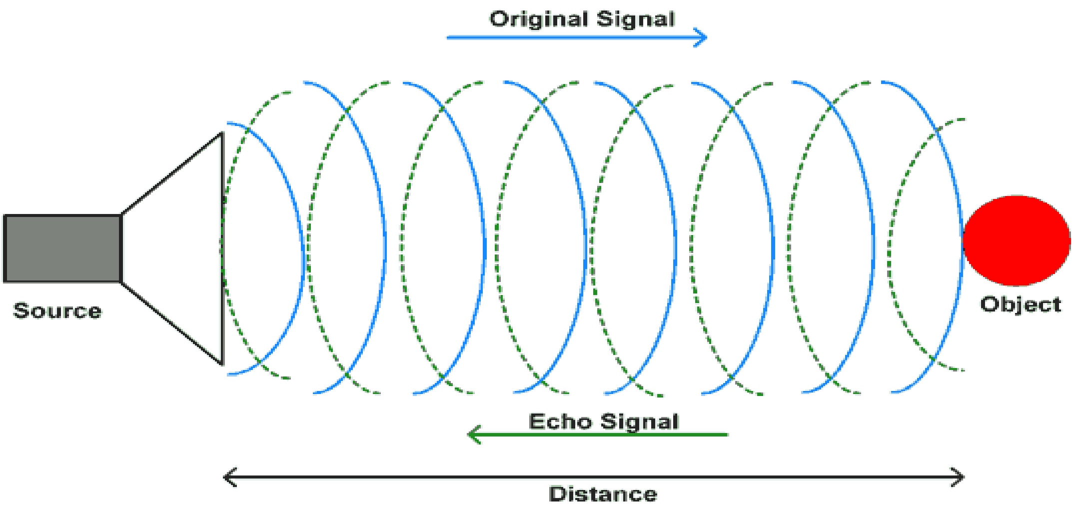

Fig 3.10 Technique for distance measurement

3.4.2 Pin Configuration:

Fig 3.11 Pin diagram for HC-SR04 module

VCC - +5 V supply

TRIG – Trigger input of sensor. Microcontroller applies 10 us trigger pulse to the HC-SR04 ultrasonic module.

ECHO–Echo output of sensor. Microcontroller reads/monitors this pin to detect the obstacle or to find the distance.

GND – Ground

3.4.3 Working Principle:

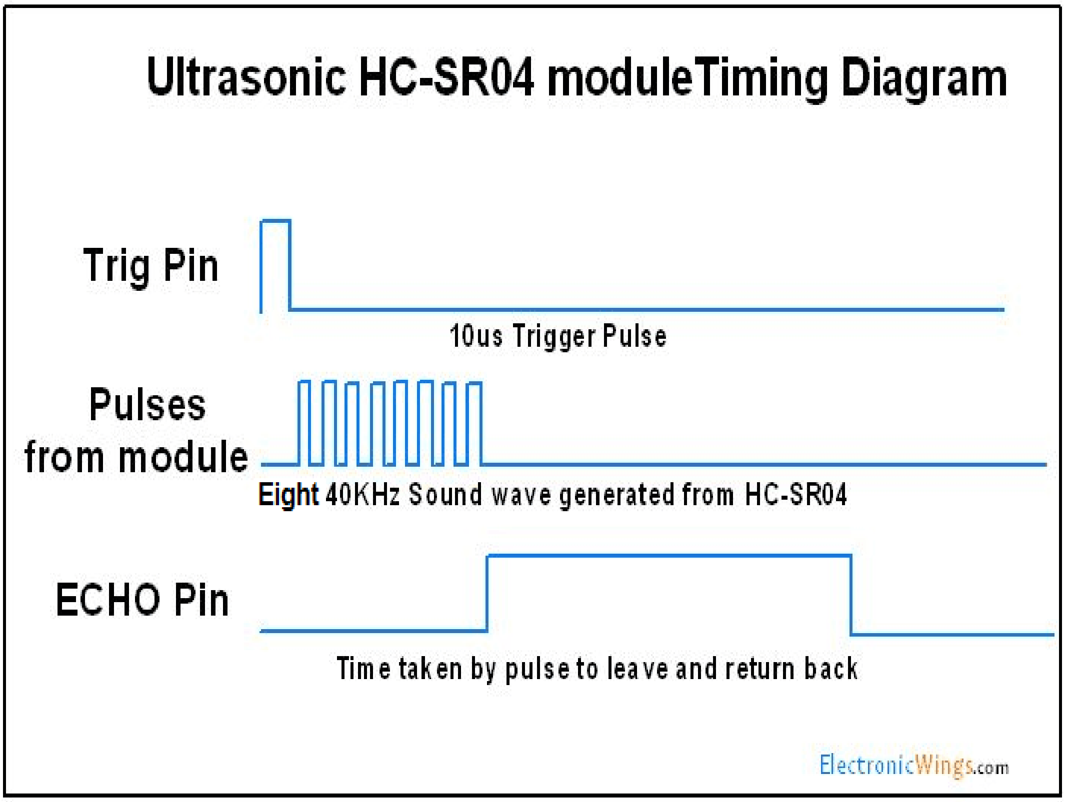

Fig 3.12 HC-SR04 Ultrasonic Module Timing Diagram

We need to transmit trigger pulse of at least 10 us to the HC-SR04 Trig Pin.

Then the HC-SR04 automatically sends Eight 40 kHz sound wave and wait for rising edge output at Echo pin.

When the rising edge capture occurs at Echo pin, start the Timer and wait for falling edge on Echo pin.

As soon as the falling edge is captured at the Echo pin, read the count of the Timer. This time count is the time required by the sensor to detect an object and return back from an object.

$$\text{Distance} = \text{Speed x Time}$$

The speed of sound waves is 343 m/s.

So,

$$\text{Total Distance} = \frac{\text{343 x Time of High(Echo) Pulse}}{2}$$

Total distance is divided by 2 because signal travels from HC-SR04 to object and returns to the module HC-SR-04.

3.4.4 Applications:

Ultrasonic sensors can be used in variety of applications. Most common applications include:

Measuring distance

Detecting an obstacle

People detection for counting

Liquid level control

Vehicle detection in car wash, automotive assembly and parking garage

Bottle Counting on Drink Filling Machines

3.4.5 Technical Specifications:

Working Voltage	DC 5V
Working Current	15 mA
Working Frequency	40 Hz
Max Range	4m
Min Range	2cm
Measuring Angel	15 degree
Trigger Input Signal	10uS TTL Pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45x20x15mm

Table 3.2 Technical specifications of ultrasonic module

3.5 L298N Motor Driver Module

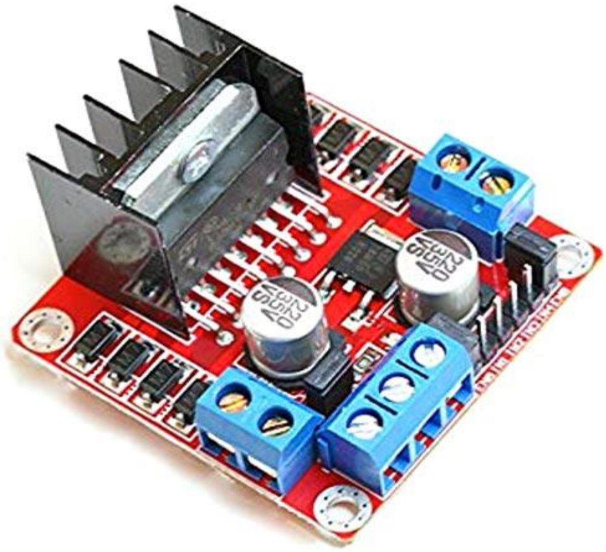

Fig 3.13 L298N

3.5.1 Introduction:

L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

3.5.2 Pin Configuration:

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of motor B
ENB A	Enables PWM signal for Motor A
ENB B	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B

12V	12V input from DC power source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground Pin

Table 3.3 Pin configuration of motor driver

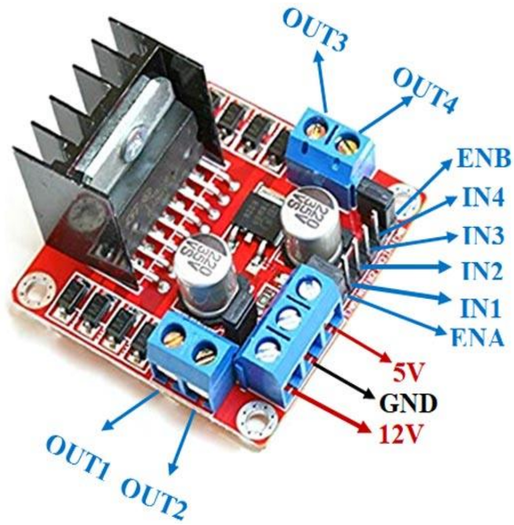

Fig 3.14 Pin diagram for L298N

3.5.3 Specifications:

Driver Model: L298N 2A

Driver Chip: Double H Bridge L298N

Motor Supply Voltage (Maximum): 46V

Motor Supply Current (Maximum): 2A

Logic Voltage: 5V

Driver Voltage: 5-35V

Driver Current: 2A

Logical Current: 0-36mA

Maximum Power (W): 25W

Current Sense for each motor

Heatsink for better performance

Power-On LED indicator

3.5.4 **Brief about L298N Module:**

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.

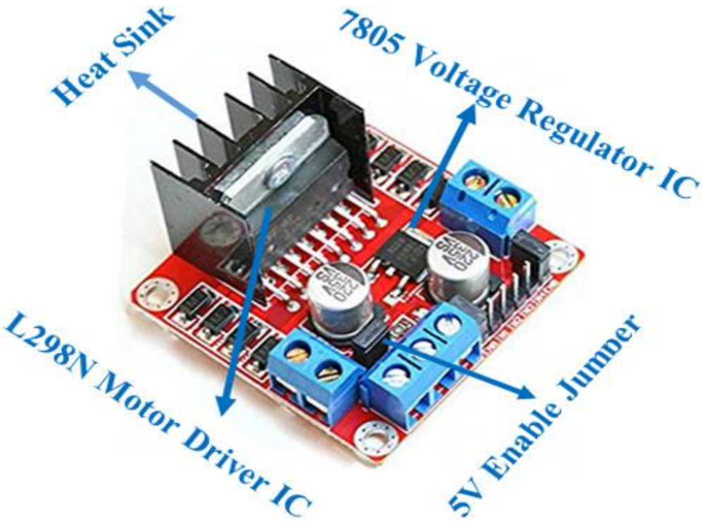

Fig 3.15 Features of L298N

78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.

ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

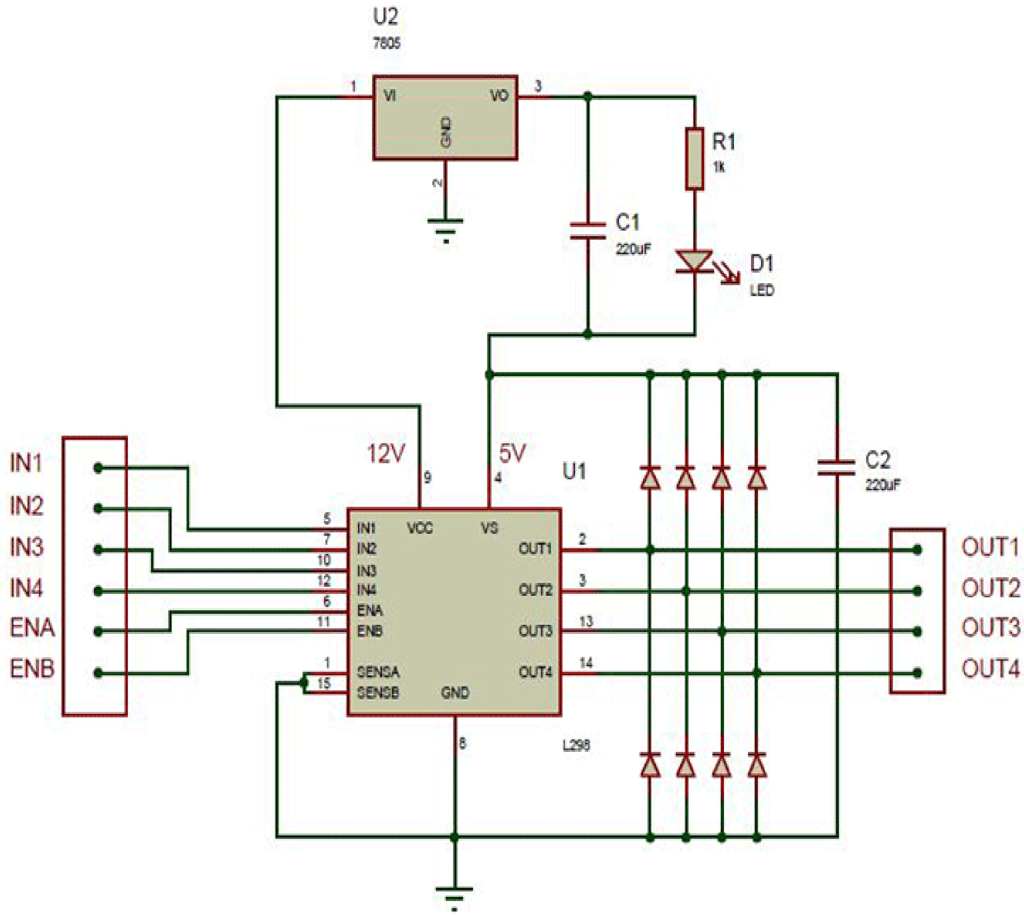

Fig 3.16 Internal circuit diagram of L298N Motor Driver module

3.5.5 Applications

1. Drive DC motors.
2. Drive stepping motors
3. Robotics

3.6 ARDUINO MEGA \ ARDUINO NANO:

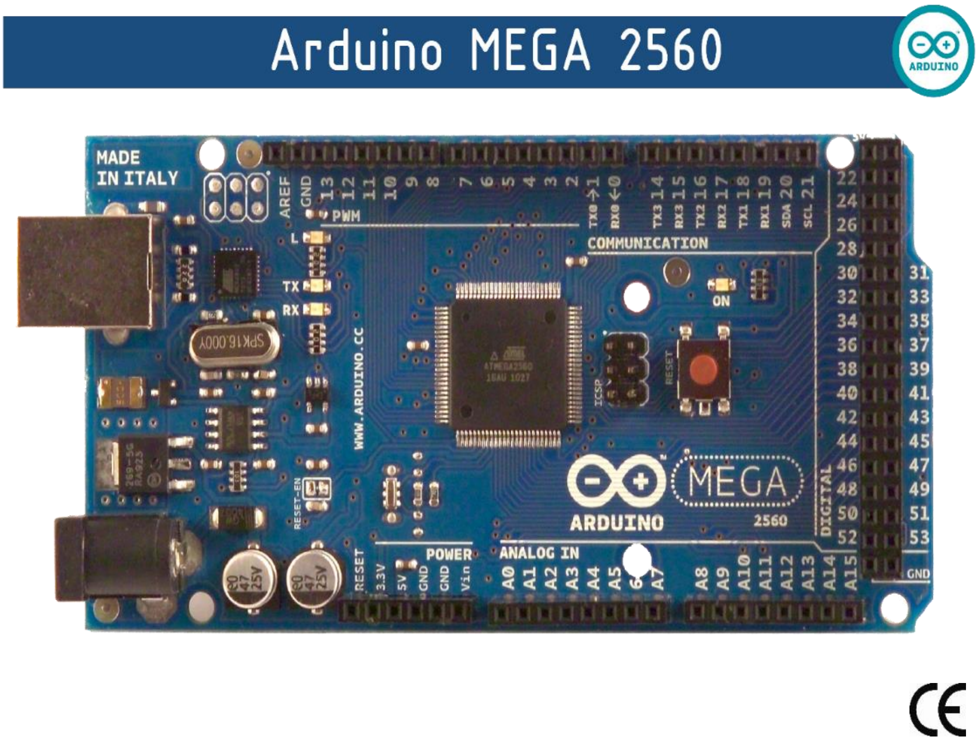

Fig 3.17 Arduino Mega

3.6.1 Introduction:

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

3.6.2 Technical Specification:

Microcontroller	ATmega2560
Operating Voltage	5V Input Voltage 7-12V(limit)
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA

DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

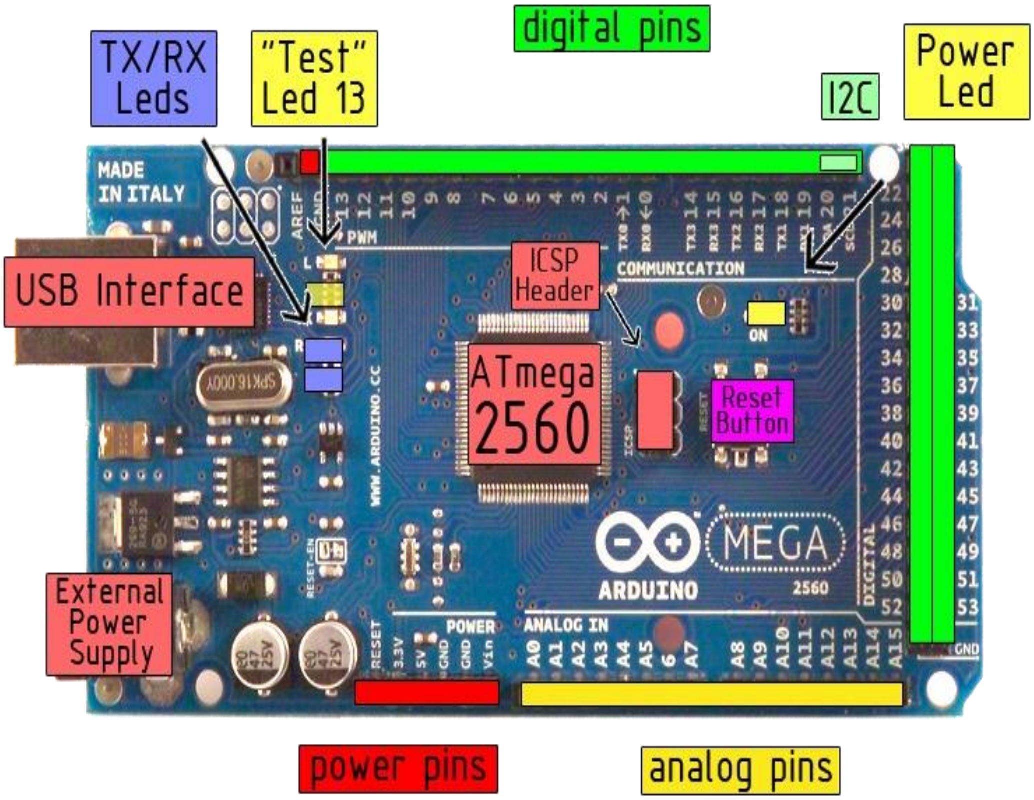

Fig 3.18 Arduino mega pin diagram

3.6.3 Pins Discription:

Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and VIN pin headers of the POWER connector.

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

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** the regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM.

Input and Output :

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

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. The attachInterrupt() function is used for this purpose.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().

- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication:

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

Programing

The Arduino Mega2560 can be programmed with the Arduino software.

The Atmega2560 on the Arduino Mega comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

3.6.4 Arduino Nano Technical Specifications:

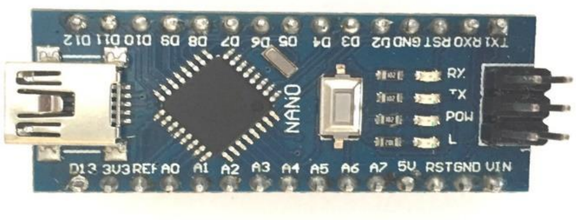

Fig 3.19 Arduino Nano

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage for Vin pin	7-12V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (2 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz
Communication	I2C, SPI, USART

Table 3.4 Technical Specifications of Arduino Nano

3.7 Raspberry Pi

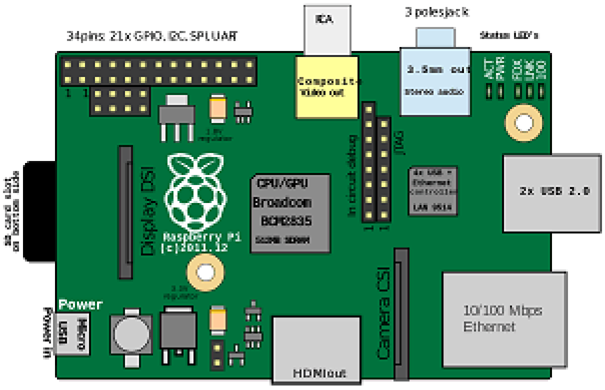

Fig 3.20 Raspberry pi Board

3.7.1 Introduction:

The Raspberry pi is a single computer board with credit card size that can be used for many tasks that your computer does, like games, word processing, spreadsheets and also to play HD video. It was established by the Raspberry pi foundation from the UK. It has been ready for public consumption since 2012 with the idea of making a low-cost educational microcomputer for students and children. The main purpose of designing the raspberry pi board is, to encourage learning, experimentation and innovation for school level students. The raspberry pi board is a portable and low cost. Maximum of the raspberry pi computers is used in mobile phones. In the 20th century, the growth of mobile computing technologies is very high, a huge segment of this being driven by the mobile industries. The 98% of the mobile phones were using ARM technology.

3.7.2 Technical Specifications:

The Raspberry Pi 3 Model B is the earliest model of the third-generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016.

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board

- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

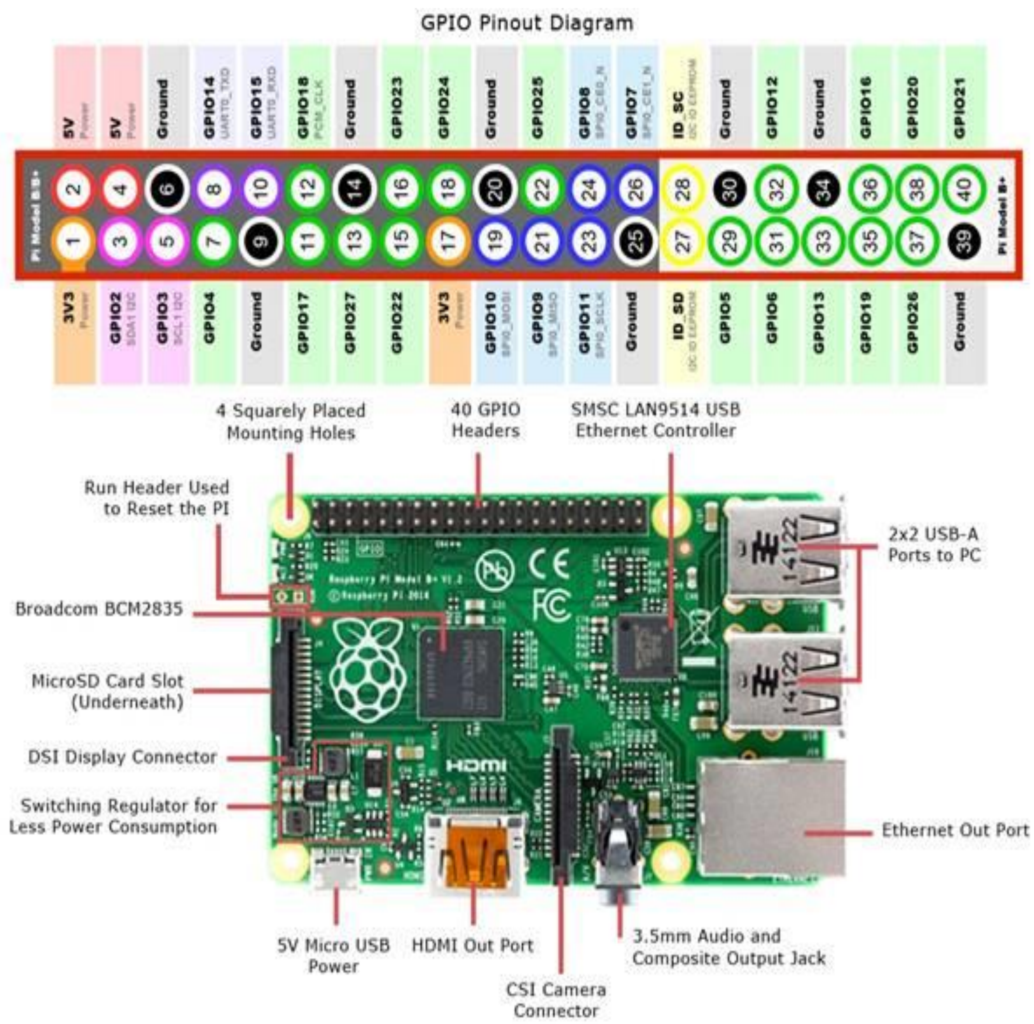

Fig 3.21 Hardware Specifications of Raspberry pi

3.8 Camera Module

Fig 3.22 Raspberry pi camera

3.8.1 Introduction:

The Raspberry Pi Camera Module v2 replaced the original Camera Module in April 2016. The v2 Camera Module has a Sony IMX219 8-megapixel sensor (compared to the 5-megapixel OmniVision OV5647 sensor of the original camera).

The Camera Module can be used to take high-definition video, as well as stills photographs. It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi.

The camera works with all models of Raspberry Pi 1, 2, 3 and 4. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Picamera Python library. The camera module is very popular in home security applications, and in wildlife camera traps.

CHAPTER 4 : DESIGN

4.1 Introduction:

The Smartwheelchair will consist of the following features:

1. Hand movement/gesture control (Drive the wheelchair).
2. Panic Button (Alert the caretaker).
3. Obstacle Avoider.
4. Camera Input on Smartphone to assist the user when away from wheelchair.

The above features are implemented by carelfully designing each and every system as per the user’s convenience. The following block diagram is designed and exlpained to show how all the features are implemented.

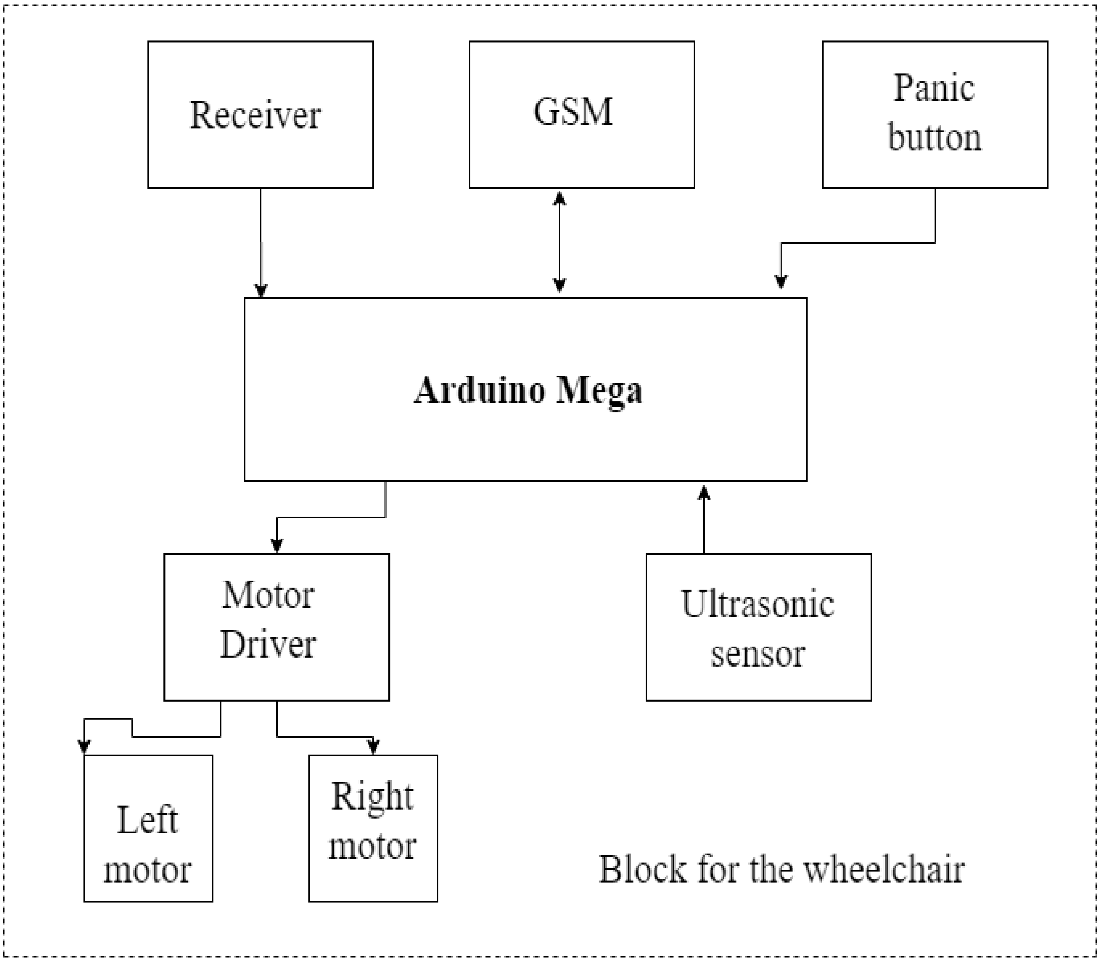

Fig 4.1 Block diagram for the wheelchair (Receiver)

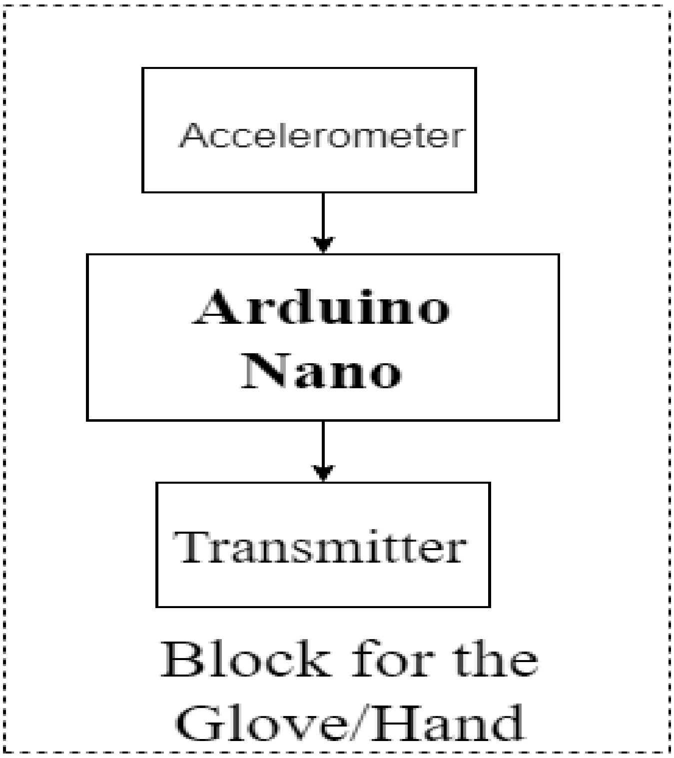

Fig 4.2 Block diagram for the Gloves(Transmitter)

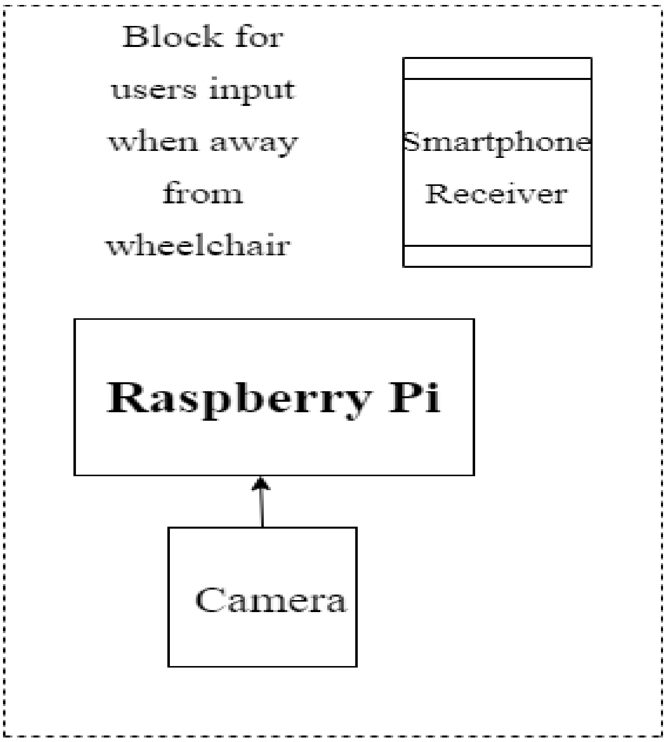

Fig 4.3 Block diagram for the camera input

4.2 Block Diagram Description

Accelerometer: Accelerometer is one of the main inputs for this project. The accelerometer sensors are mounted on headphone/cap/hand and they transducer change in acceleration of head movement to voltage signal which is sent to ADC input of microcontroller. Module to used MPU6050.

Ultrasonic Ranging Module: It is used to detect the obstacle. It is use to halt the wheel chair.

Microcontroller: It is the main computational and processing part of the Smart Wheel Chair that takes input from the sensor, processes it and gives output. Arduino mega/Nano and raspberry pi 3 model B is used in this project.

Transmitter/Receiver: The accelerometer is mounted on the hand/glove of the user which is interfaced with Arduino Nano. Accelerometer records the hand movement and sends the input to the microcontroller in our case Arduino Nano. After processing the raw data the microcontroller will transmit the processed data through a Trans receiver. The module used in this project is nRF24L01. The same module is present on the wheelchair interfaced to an Arduino Mega which will receive the data send by the microcontroller mounted on the hand of the user. This data will be used to drive the wheelchair in the direction the user wants.

GSM (Global System for Mobile communication):

A GSM module is a chip or circuit that will be used to establish communication between a mobile device and a computing machine.it is used to send messages to the caretaker that he/she need help.

A gsm digitizes and reduces the data then sends it down through a channel with two different streams of client data. It requires sim (**Subscriber Identity Module)** card to activate communication with the network.

Module to be used SIM900A.

Panic button: Once the button is pressed it will sent the messages to caretakers that he/she need help

Camera: the camera will capture the live image of the surrounding and feedback to the raspberry pie where it will transmit the image to the smartphone. This will help the person away from the wheelchair to navigate the wheelchair towards him/her. Raspberry camera module will be used. User's mobile is used to stream the live video of the wheelchairs environment.

Motor Drivers: Motor Drivers amplifies the output of the microcontroller such that it can drive the respective actuators. L298 module /L293D IC can be used to drive the motors. It is dual H-Bridge IC.

Actuators: Actuators are the motors that drive Smart Chair. They actually change the electric signals of the microcontroller into the rotational motion and provide desired functionality. Motors used: DC supply: 4 to 12V RPM: 100 at 12V.

Fig 4.4 Dc Motor

4.3 Flowcharts :

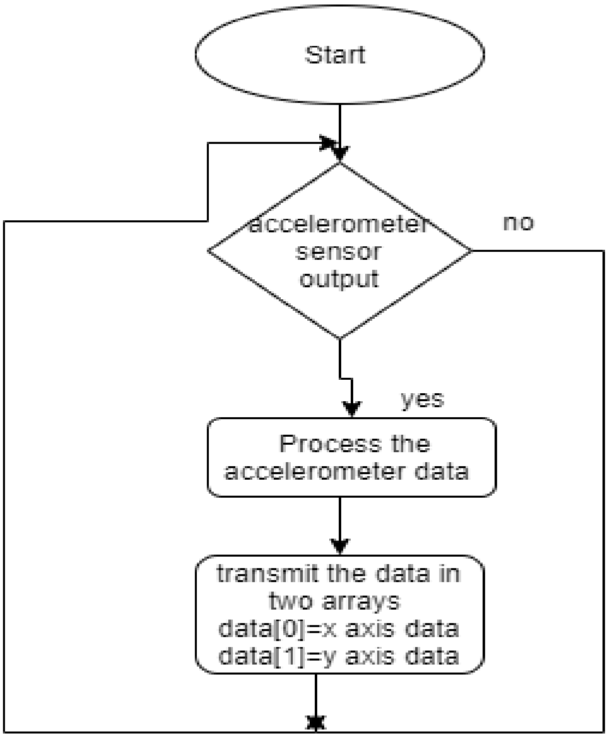

Fig 4.4 Flowchart for the Glove (Transmitter)

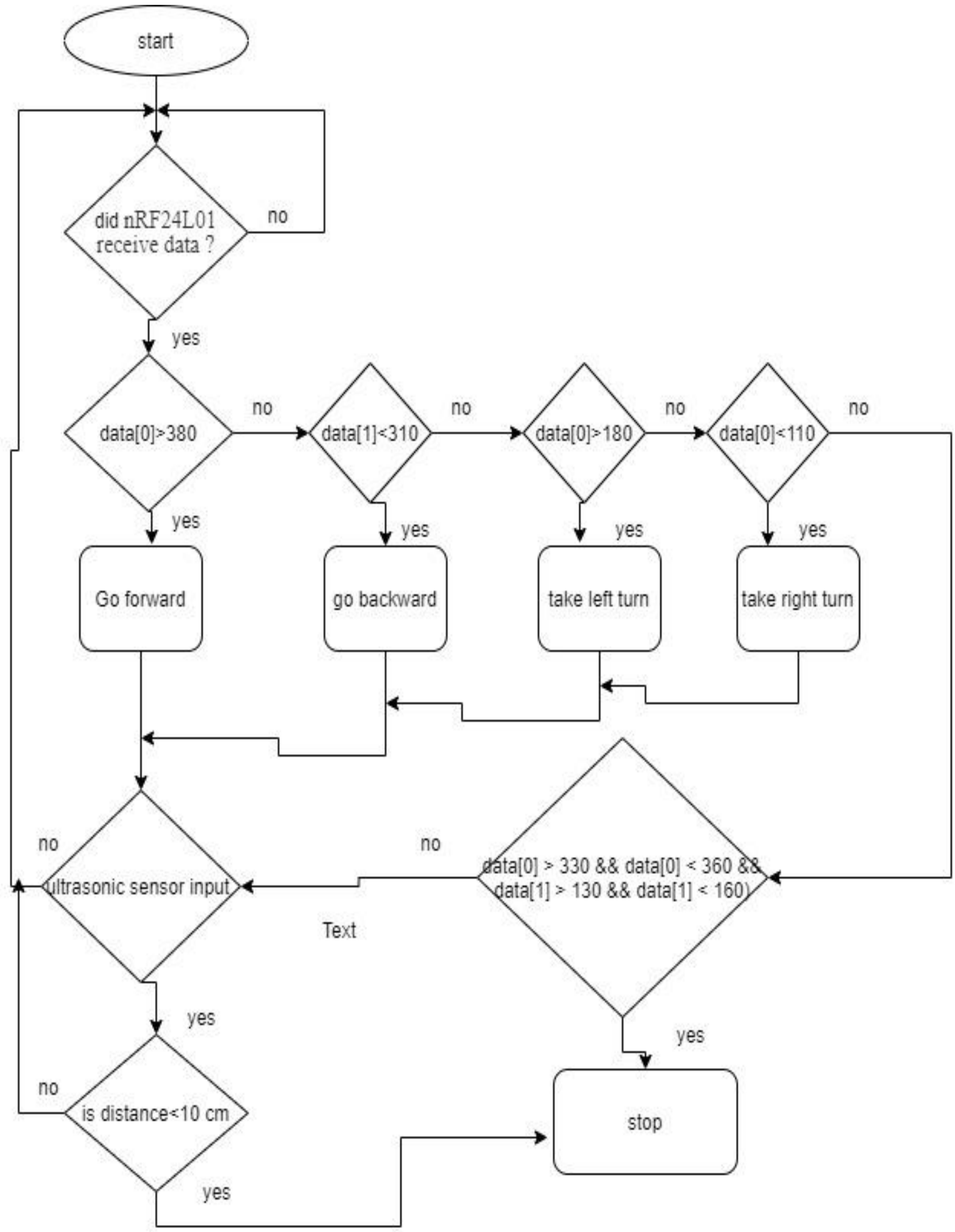

Fig 4.6 Flowchart for the wheelchair (Receiver)

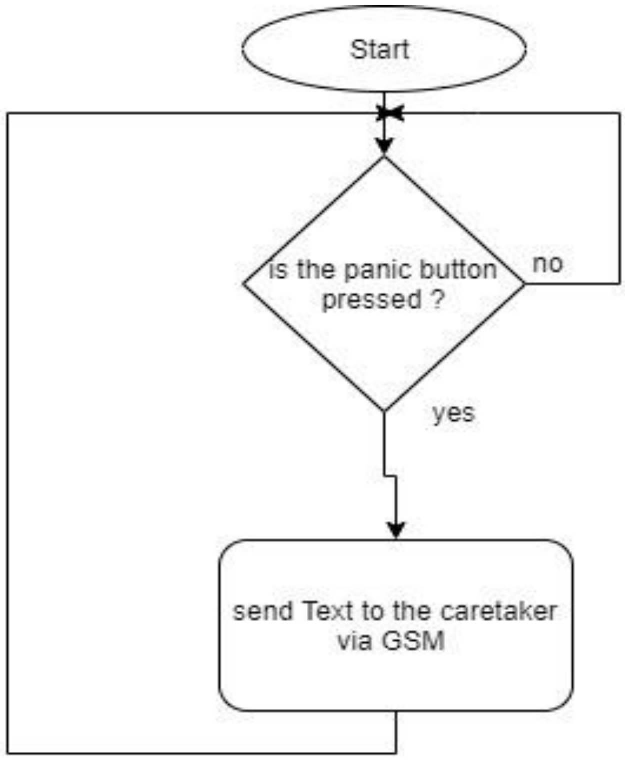

Fig 4.7 Flowchart for the panic button

CHAPTER 5: EXPERIMENTS CONDUCTED

5.1 Hand movement/gesture control (Drive the wheelchair)

5.1.1 Introduction:

The gesture-controlled part of the wheelchair uses Arduino Mega /Nano, MPU6050 Accelerometer, nRF24L01 Transceiver pair, and L298N motor driver module. This part of the smart wheelchair can be divided into parts, one is the Transmitter, and the other is the Receiver. Transmitter section consists of an Arduino Nano, MPU6050 Accelerometer and Gyroscope, and nRF24L01 while the Receiver section consists of an Arduino Mega, nRF24L01, two DC motors and L293D motor driver. The transmitter will act as remote to control the Robot where the robot will move according to the gestures.

5.1.2 Working of Hand Gesture controlled wheelchair using Arduino:

The first part is the transmitter part (remote) in which the MPU6050 Accelerometer sensor continuously sends signals to the receiver (wheelchair) through Arduino and nRF transmitter.

The second part is the Receiver part (wheelchair) in which the nRF receiver receives the transmitted data and sends it to Arduino, which further processes them and move the wheelchair accordingly.

The MPU6050 Accelerometer sensor reads the X Y Z coordinates and sends the coordinates to the Arduino. For this project, we need only X and Y coordinates. Arduino then checks the values of coordinates and sends the data to the nRF Transmitter. The transmitted data is received by the nRF Receiver. The receiver sends the data to the receiver side's Arduino. Arduino passes the data to the Motor Driver IC and the motor driver turns the motors in the required direction.

5.1.3 Circuit Diagram:

This Hand Gesture controlled Wheelchair using Arduino hardware is divided into two sections

1. Transmitter
2. Receiver

Transmitter Circuit for Arduino Gesture Controlled wheelchair:

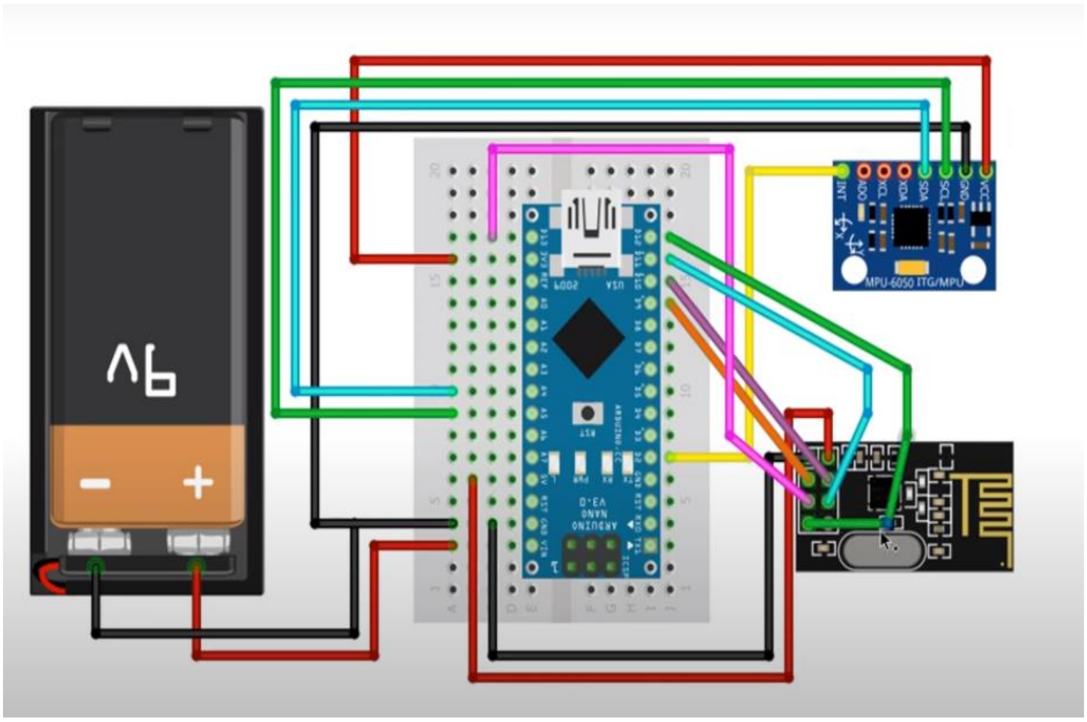

Fig 5.1 Interfacing diagram for the transmitter

The transmitter section of this project consists of MPU6050 Accelerometer and Gyroscope, nRF24L01Transceiver, and Arduino Uno. The Arduino continuously gets data from the MPU6050 and sends this data to the nRF Transmitter. RF transmitter transmits the data into the environment.

Pin connections for Transmitter:

Fig 5.2 Pin positions for NRF24L01

nRF24L01	Arduino Nano
VCC	5V
GND	GND
CE	PIN 9
CSN	PIN 10
SCK	PIN 13
MOSI	PIN 11
MISO	PIN 12

Table 5.1 Pin connections of RF module to Arduino Nano

Pin Connections for accelerometer:

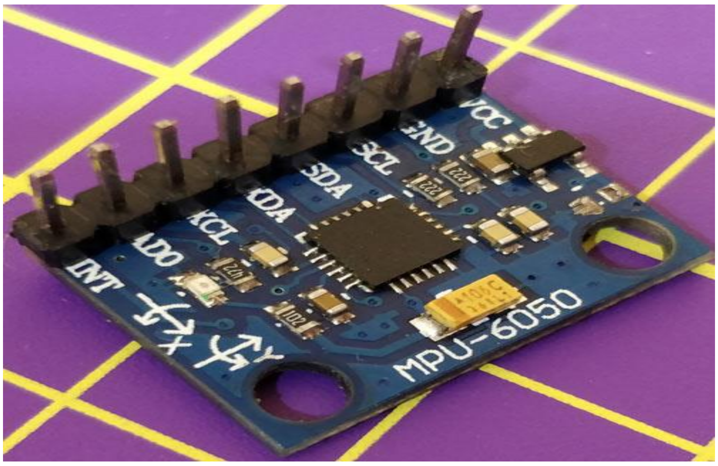

Fig 5.3 Pin positions for MPU6050

MPU 6050	Arduino Nano
VCC	3.3V
GND	GND
SCL	PIN 5
SDA	PIN 4
INT	PIN 2

Table 5.2 Pin connections of Accelerometer to Arduino Nano

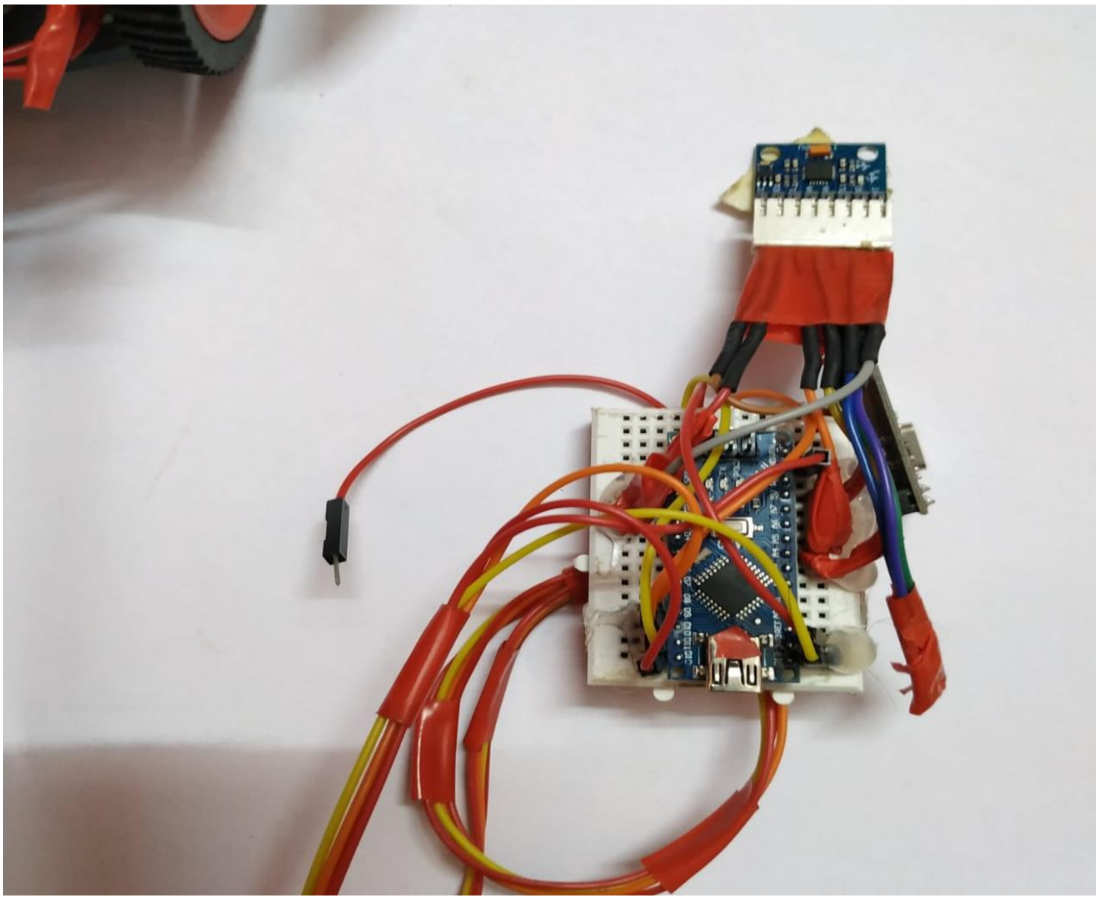

Fig 5.4 Implemented Module for transmission

Receiver Circuit for Arduino Gesture Controlled wheelchair:

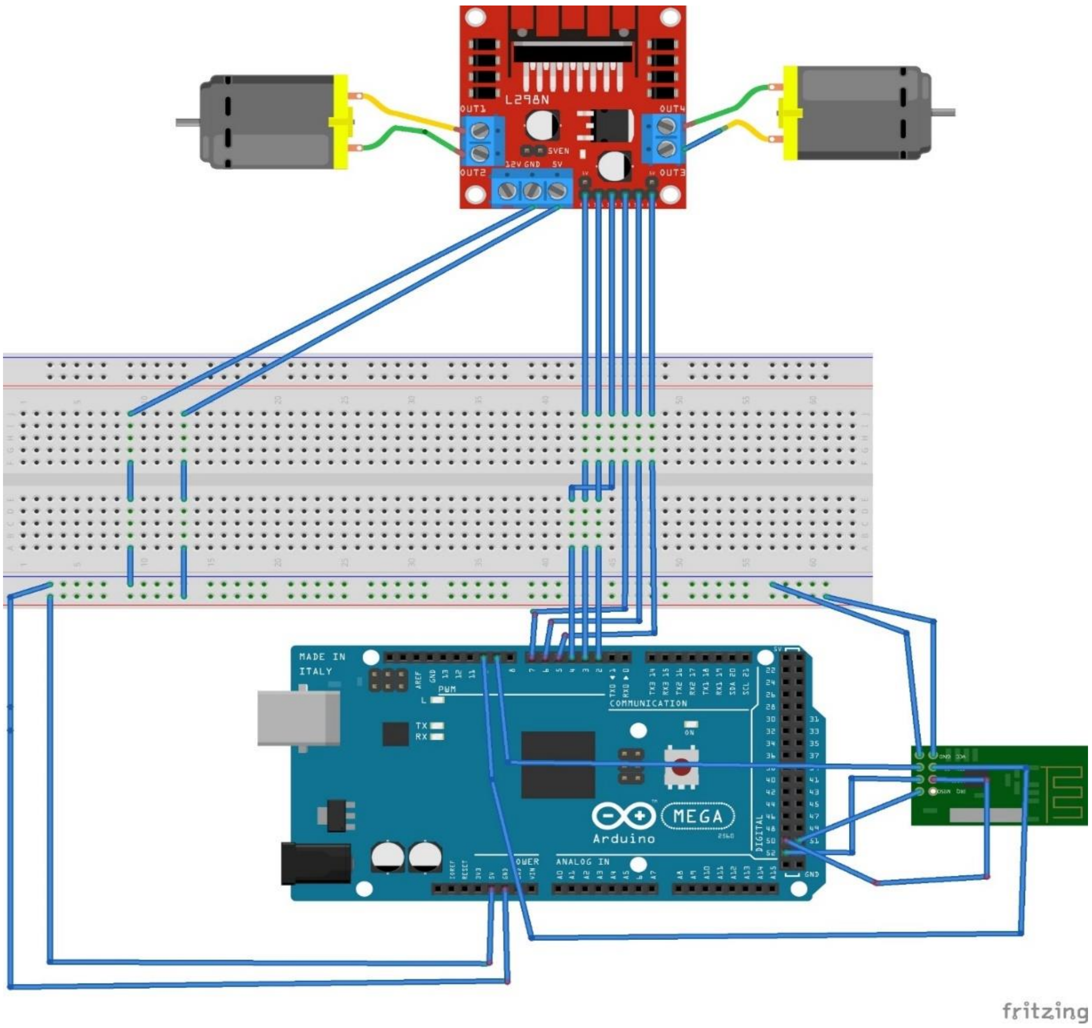

Fig 5.5 Interfacing diagram for the receiver

The receiver section of this gesture controlled wheelchair consists of Arduino Mega, nRF24L01 Transceiver, 2 DC motors, and Motor driver module. NRF24L01 receiver receives the data from the transmitter and sends it to Arduino. Then according to received signals, Arduino moves the DC motors.

Pin connections for the receiver:

nRF24L01	Arduino Mega
VCC	5V
GND	GND
CE	PIN 9

CSN	PIN 10
SCK	PIN 52
MOSI	PIN 51
MISO	PIN 50

Table 5.3 Pin connections of RF module to Arduino Mega

Pin connections for the motor driver L298N:

L298N	Arduino Mega
5V VCC	5V
GND	GND
ENB A	PIN 3
INT 1	PIN 2
INT 2	PIN 4
ENB B	PIN 5
INT 3	PIN 7
INT 4	PIN 6

Table 5.4 Pin connections of Motor driver module to Arduino Mega

Connect the 2 DC motors to the available 4 output ports of L298N driver. Attach wheels to the dc motors to drive the wheelchair.

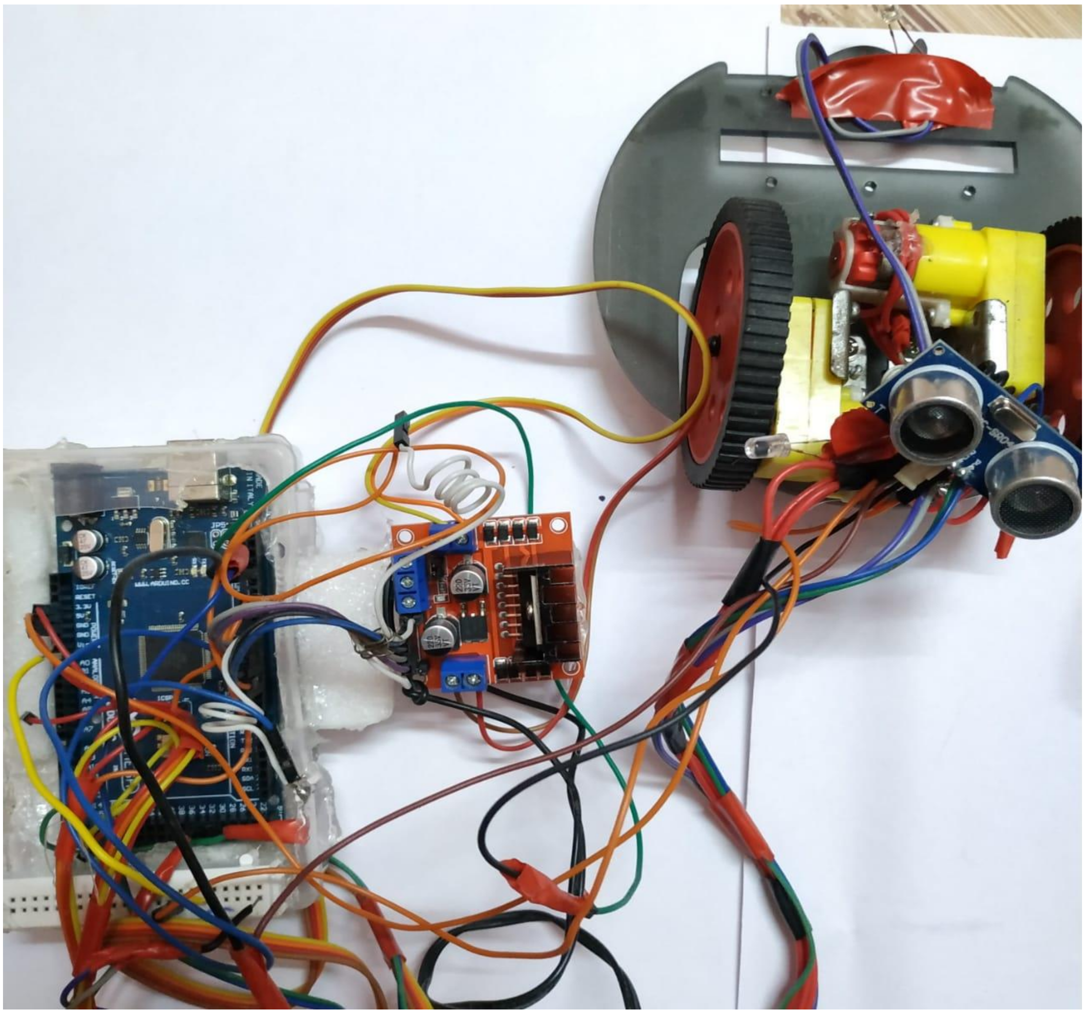

Fig 5.6 Implemented module for receiving

5.2 Panic Button (Alert the caretaker):

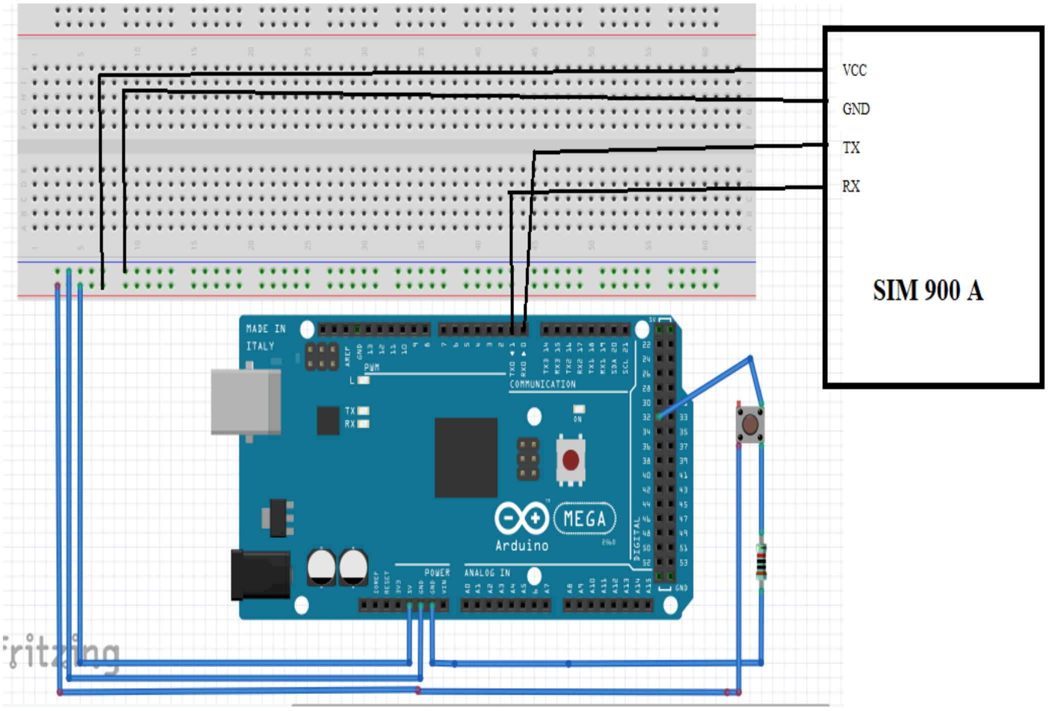

Fig 5.7 Interfacing diagram for panic button

5.2.1 Introduction:

This system is used to alert the caretaker that the user of the wheelchair needs help. If the person using the wheelchair is in any kind of trouble or need help he/she just has to press button on the wheelchair the caretaker will be notified via a GSM message. When the button is pressed the message will go to a preprogramed number of the caretaker via a GSM module SIM900A interfaced with the Arduino Mega. While designing the wheelchair only indoor condition were considered there a need for GPS module for the exact location was not considered. If the care taker happens to be outside the house he/she being notified can rush for the users help.

5.2.2 Connections and Working:

Setting up the GSM module

1. Insert SIM card to GSM module and lock it.
2. Power up your gsm by connecting it to Arduino's 5V and GND.
3. Connect the Antenna
4. Now wait for some time (say 1 minute) and see the blinking rate of ‘status LED’

or ‘network LED’ (GSM module will take some time to establish connection with mobile network)

5.Once the connection is established successfully, the status/network LED will blink continuously every 3 seconds. Try making a call to the mobile number of the sim card inside GSM module. If you hear a ring back, the gsm module has successfully established network connection.

After the GSM module and the push button is interfaced to the Arduino the working is simple. When the push button is pressed it send a high signal to Pin 32 of the Arduino. Once the high signal is received the Arduino is programmed to send a message via GSM module.

Pin connections for the GSM module:

SIM 900A	Arduino Mega
VCC	5V
GND	GND
TX	RX
RX	TX

Table 5.5 Pin connections of GSM module to Arduino Mega

Pin connections for the push button with resister value of 10K ohms:

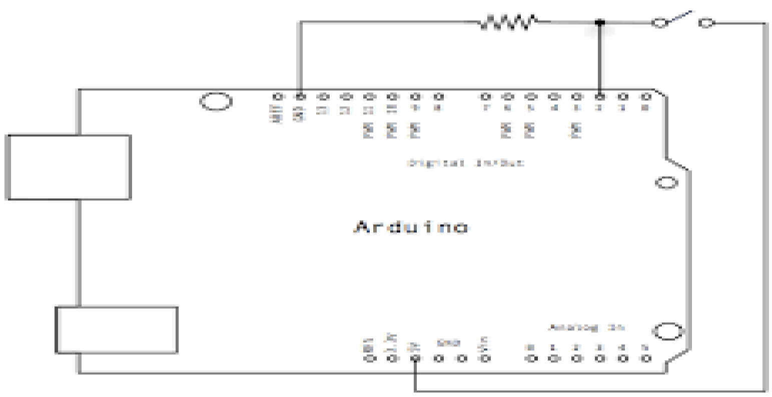

Fig 5.8 Pin connections of push button

One leg of the pushbutton is connected to 5V and the other is connected to GND via a 10k ohms resister and also connected to PIM 32 of the Arduino Mega. While programming bouncing condition must be taken care of. Bouncing is the effect of the mechanical contacts of a switch ‘bouncing’ in between the on/off state when toggled. A Microcontroller can be operating at millions of cycles per second, and these bounce frequencies are occurring right inside of that time domain. The input might try to read switch’s state and instead of seeing a single, rising edge, from 0-5V; it sees multiple rising and falling edges when you press the button. This is solved by using software debounce .A simpler method of polling our buttons state throughout our loop function. If it has changed for a period greater than the expected debounce time we will trigger the action we desire i.e. send the message.

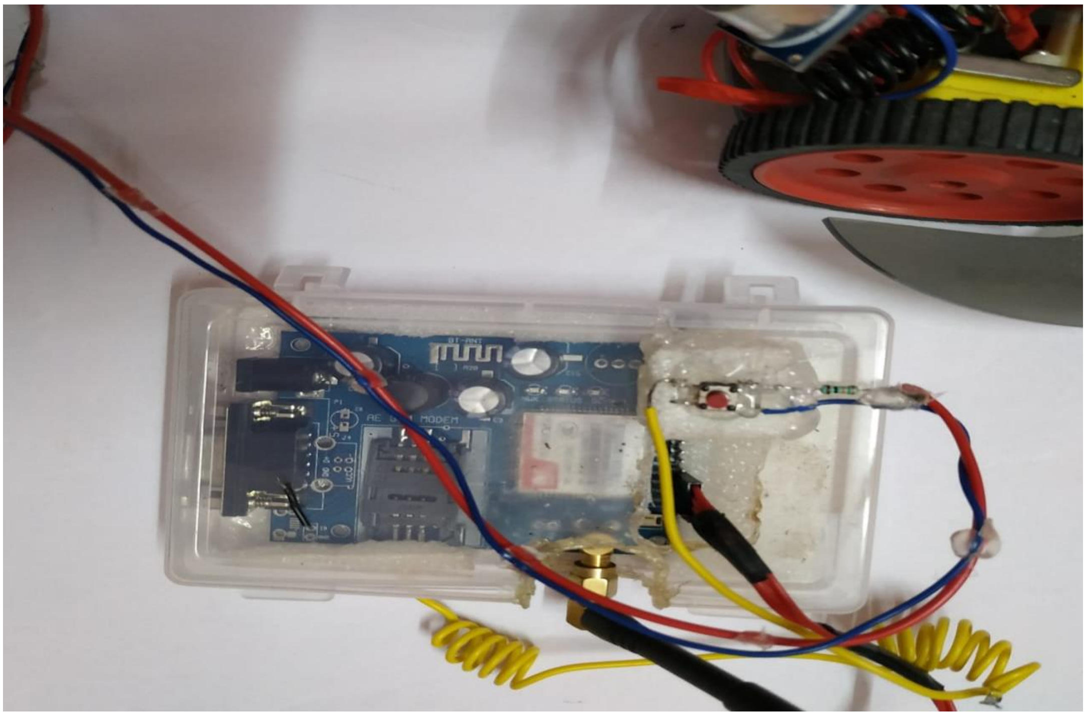

Fig 5.9 Implemented module for panic button

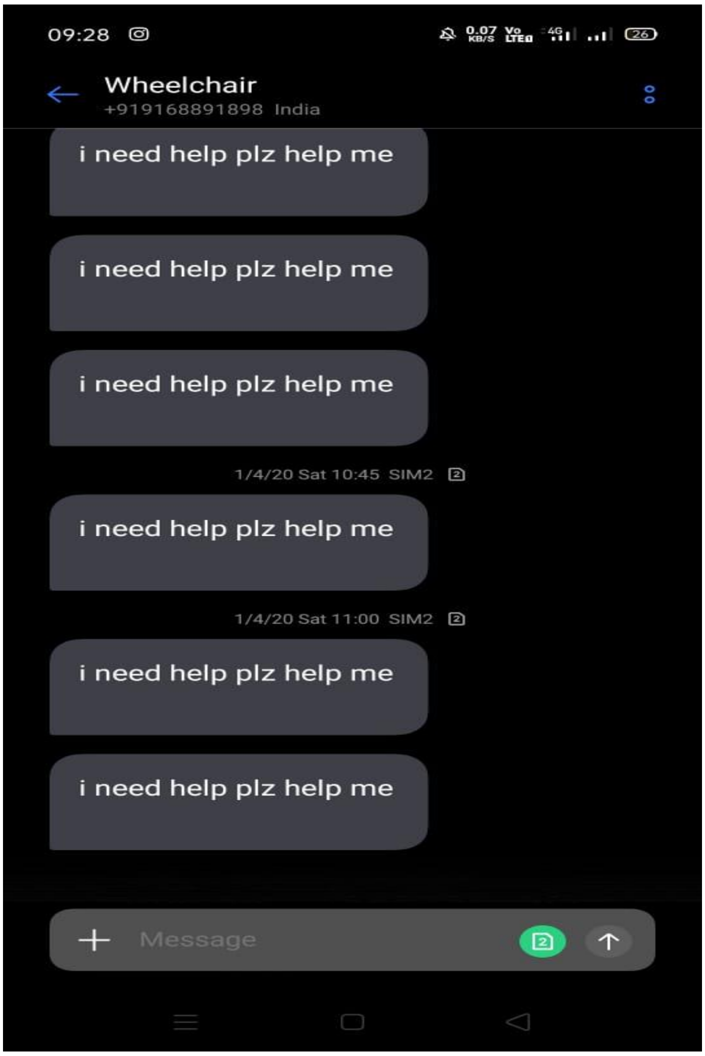

Fig 5.10 Messages received by the caretaker

5.3 Obstacle Avoider:

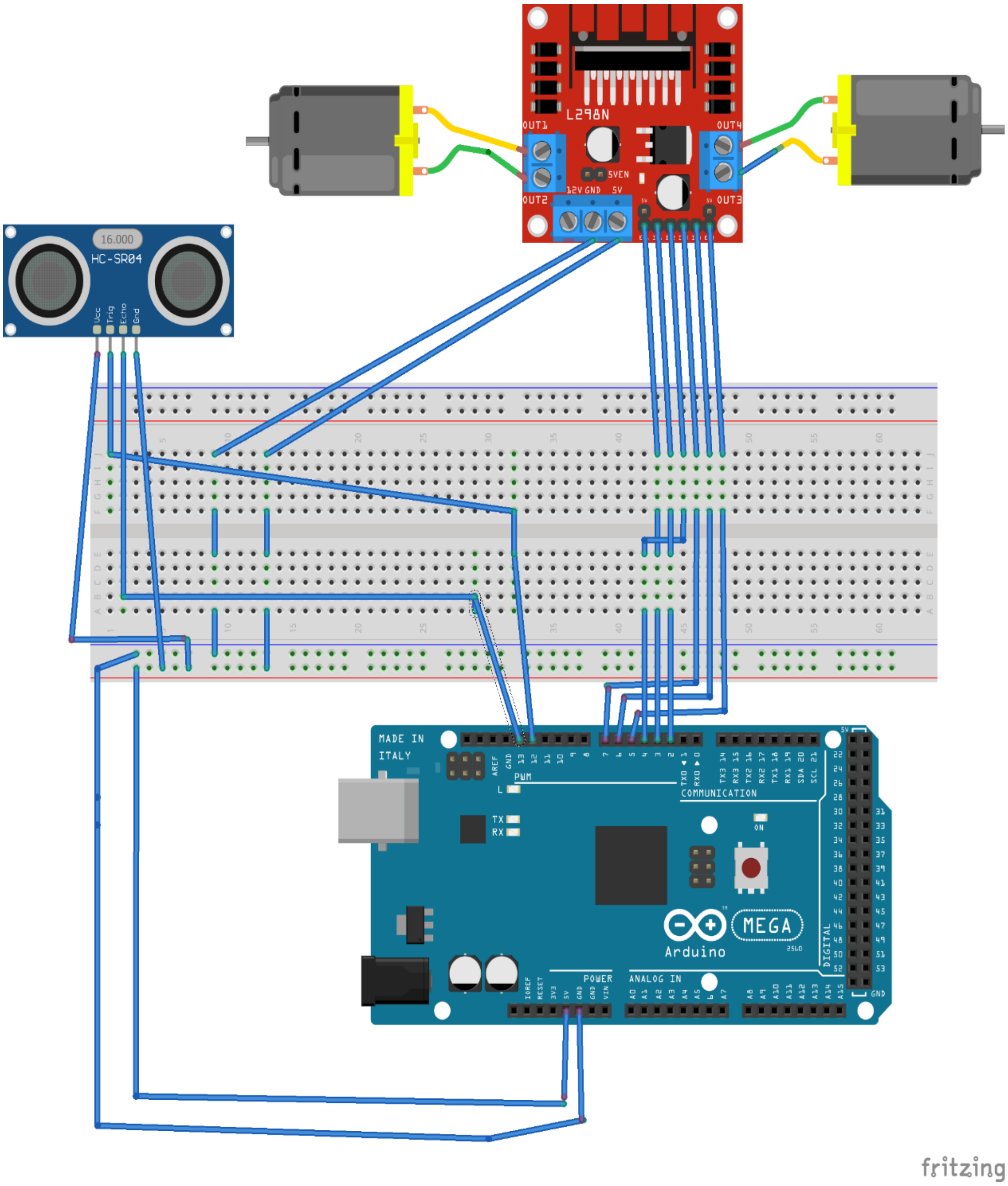

Fig 5.11 Interfacing diagram obstacle avoider

5.3.1 Introduction:

Obstacle Avoiding is an intelligent feature for the wheelchair which can automatically sense the obstacle in front of it and avoid them by turning itself in another direction or stopping the wheelchair. This projects will stop the wheelchair if any obstacle is detected. The reason behind the detection of object can be human negligence/human error. This design allows the wheelchair to navigate in unknown environment by avoiding collisions, which is a primary requirement for the safety of the person on the wheelchair. The application of Obstacle Avoiding is not limited and it is used in most of the military organization now which helps carry out many risky jobs that cannot be done by any soldiers.

Ultrasonic sensor is used to sense the obstacles in the path by calculating the distance between the wheelchair and obstacle. If wheelchair finds any obstacle it will stop moving.

5.3.2 Connections and Working:

The ultrasonic sensor sends the input to the microcontroller (Arduino Mega). The processed distance of the obstacle is stored in a variable D. if the value of D goes less than 10 cm the wheelchair will stop avoiding collision with the obstacle.

Pin diagram for connection of the Ultrasonic module:

HC-SR04 (Ultrasonic)	Arduino Mega
VCC	5V
GND	GND
Trig PIN	PIN 12
Echo PIN	PIN 13

Table 5.6 Pin connections of ultrasonic module to Arduino Mega

Pin connections for the motor driver L298N:

L298N	Arduino Mega
5V VCC	5V
GND	GND
ENB A	PIN 3
INT 1	PIN 2
INT 2	PIN 4
ENB B	PIN 5
INT 3	PIN 7
INT 4	PIN 6

Table 5.7 Pin connections of L298N module to Arduino Mega

Connect the 2 DC motors to the available 4 output ports of L298N driver. Attach wheels to the dc motors to drive the wheelchair.

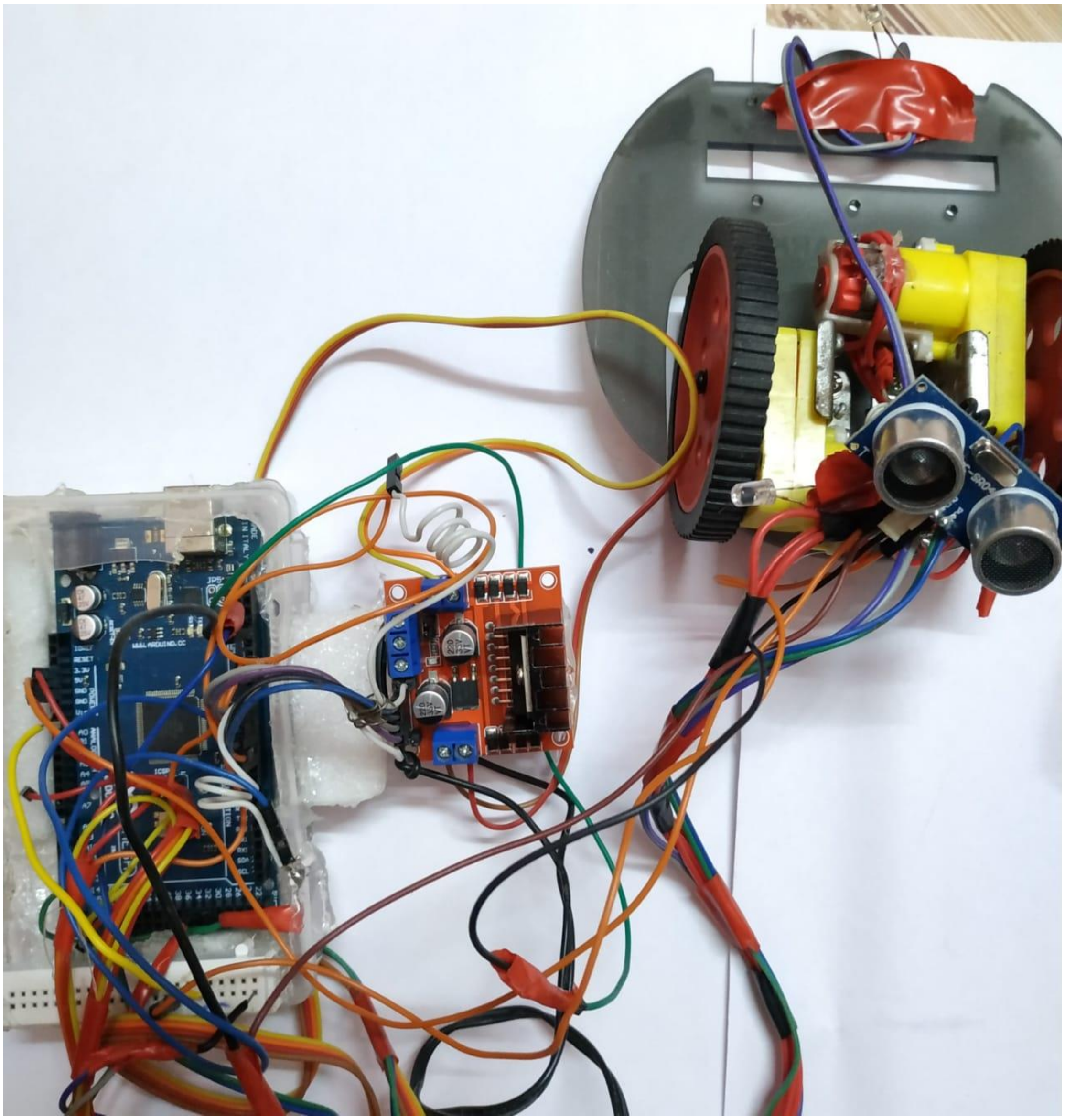

Fig 5.12 Implemented module for obstacle avoiding

5.4 Camera Input on Smartphone to assist the user when away from wheelchair.

5.4.1 Introduction:

This feature is helpful for the user when he is away from the wheelchair. The user receives live video stream of the environment the wheelchair is in on his/her smartphone and through gesture control he can call back/drive the wheelchair. This feature uses a raspberry pi, raspicam (the camera module) and VNC viewer software for Android Operating systems. VNC viewer enables android smartphone to access the Graphical user interface of the Pi.

Fig 5.13 Camera interfaced to raspberry pi 3

5.4.2 Connection of the Camera Module

Ensure Raspberry Pi is turned off.

1. Locate the Camera Module port
2. Gently pull up on the edges of the port’s plastic clip
3. Insert the Camera Module ribbon cable; make sure the cable is the right way round
4. Push the plastic clip back into place

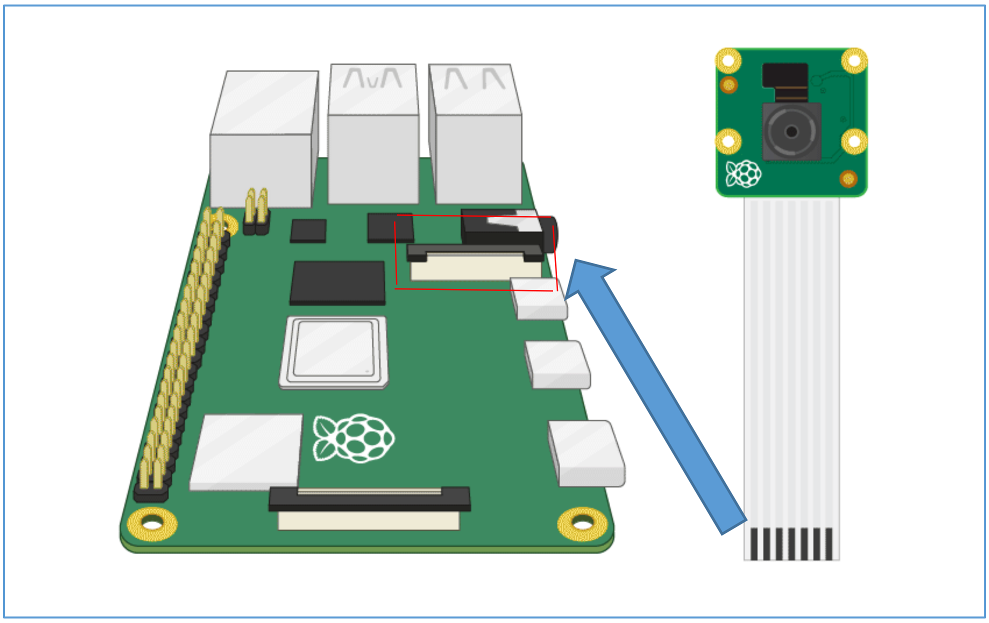

Fig 5.14 Interfacing diagram for connecting camera to raspberry pi

- Start up your Raspberry Pi.

- Go to the main menu and open the Raspberry Pi Configuration tool.

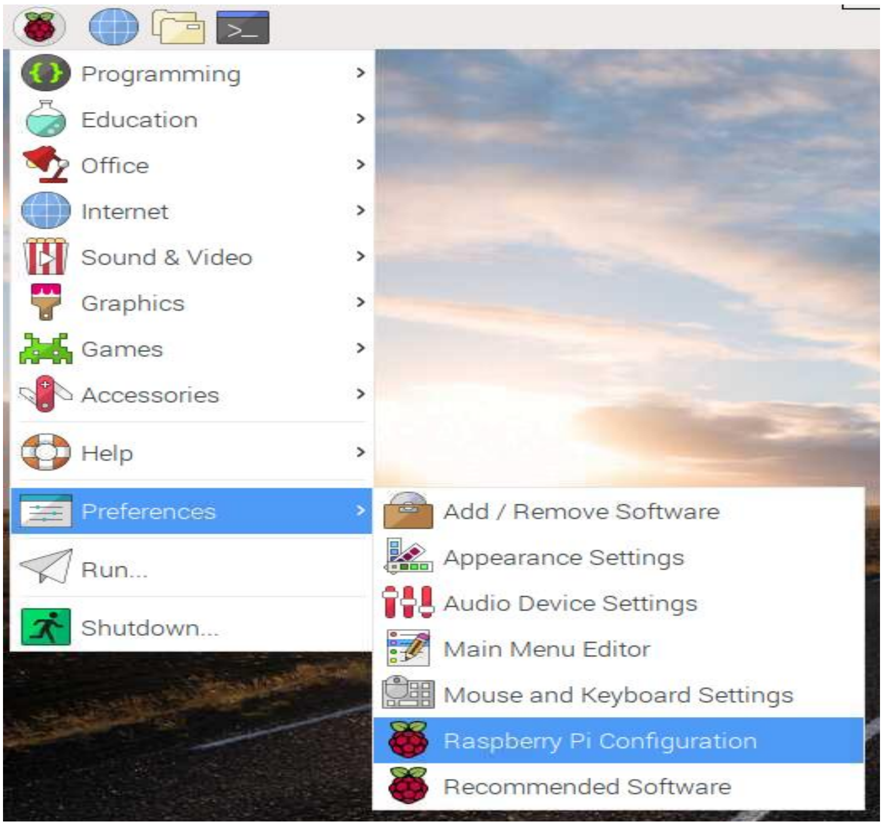

Fig 5.15 Opening of raspberry pi configuration

- Select the Interfaces tab and ensure that the camera is enabled

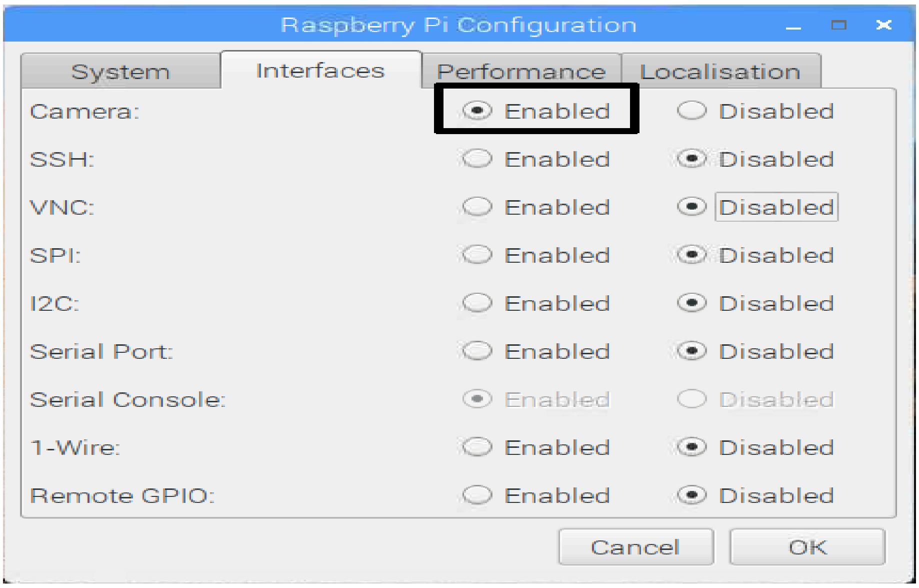

Fig 5.16 Enabling camera option

- Reboot Raspberry Pi.
- Access the pi through VNC viewer on android and run the C++ Program written to view the video stream. Program shows a video of 400 X 240.

Fig 5.17 Raspberry pi inside its caseing

CHAPTER 6 RESULTS

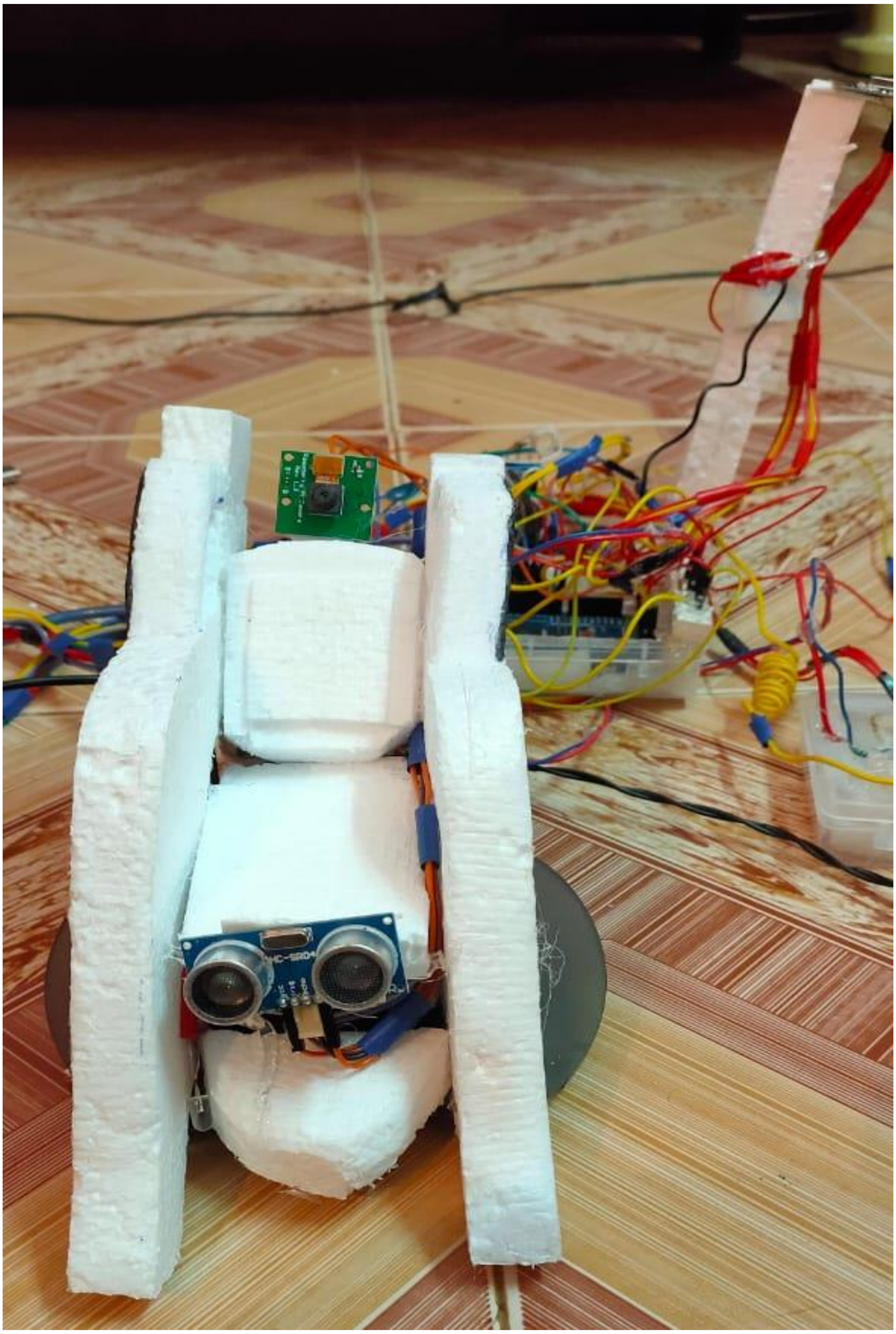

Fig 6.1 Front view of the wheelchair

Fig 6.2 Side view of the wheelchair

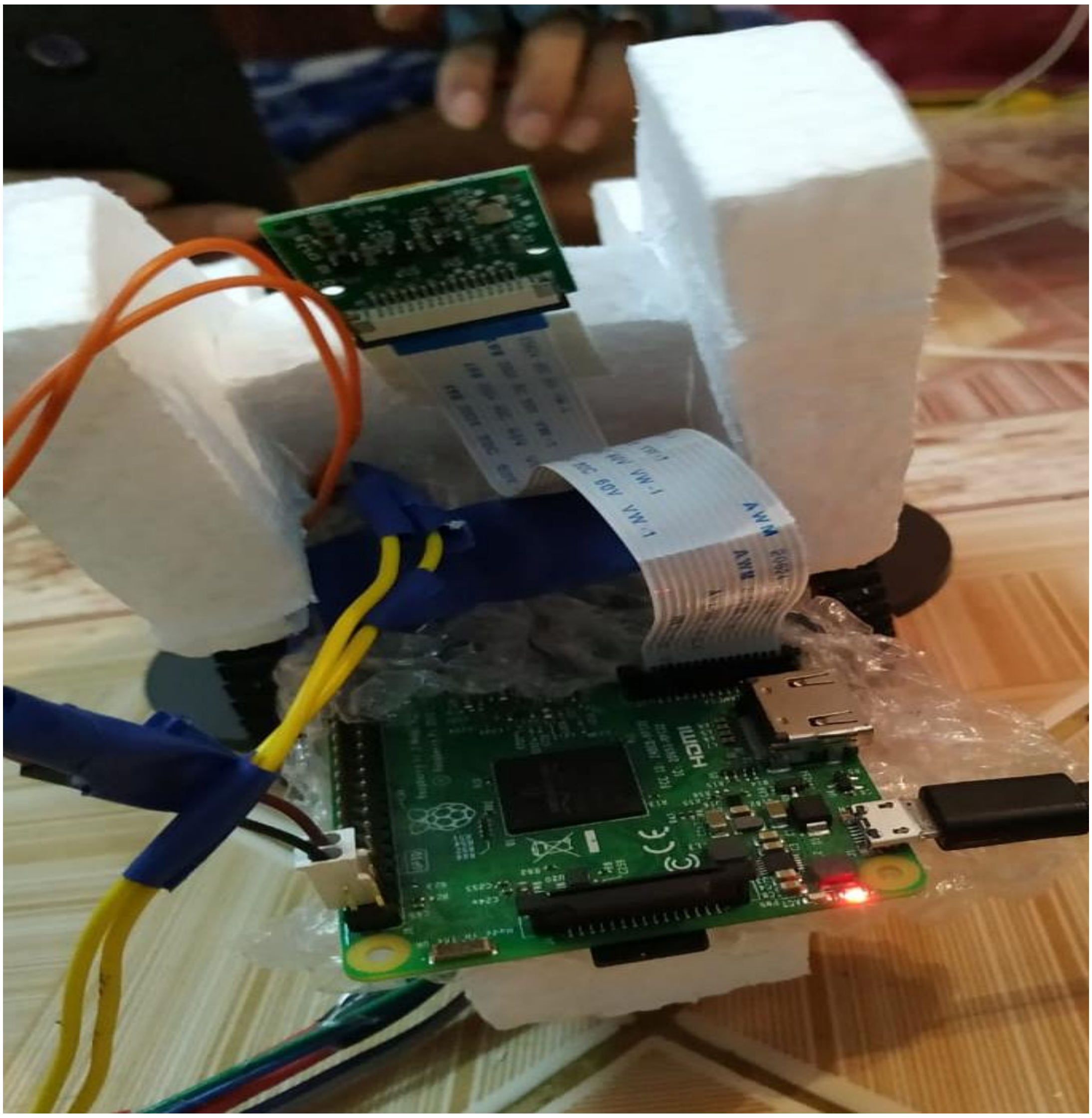

Fig 6.3 Back view of the wheelchair

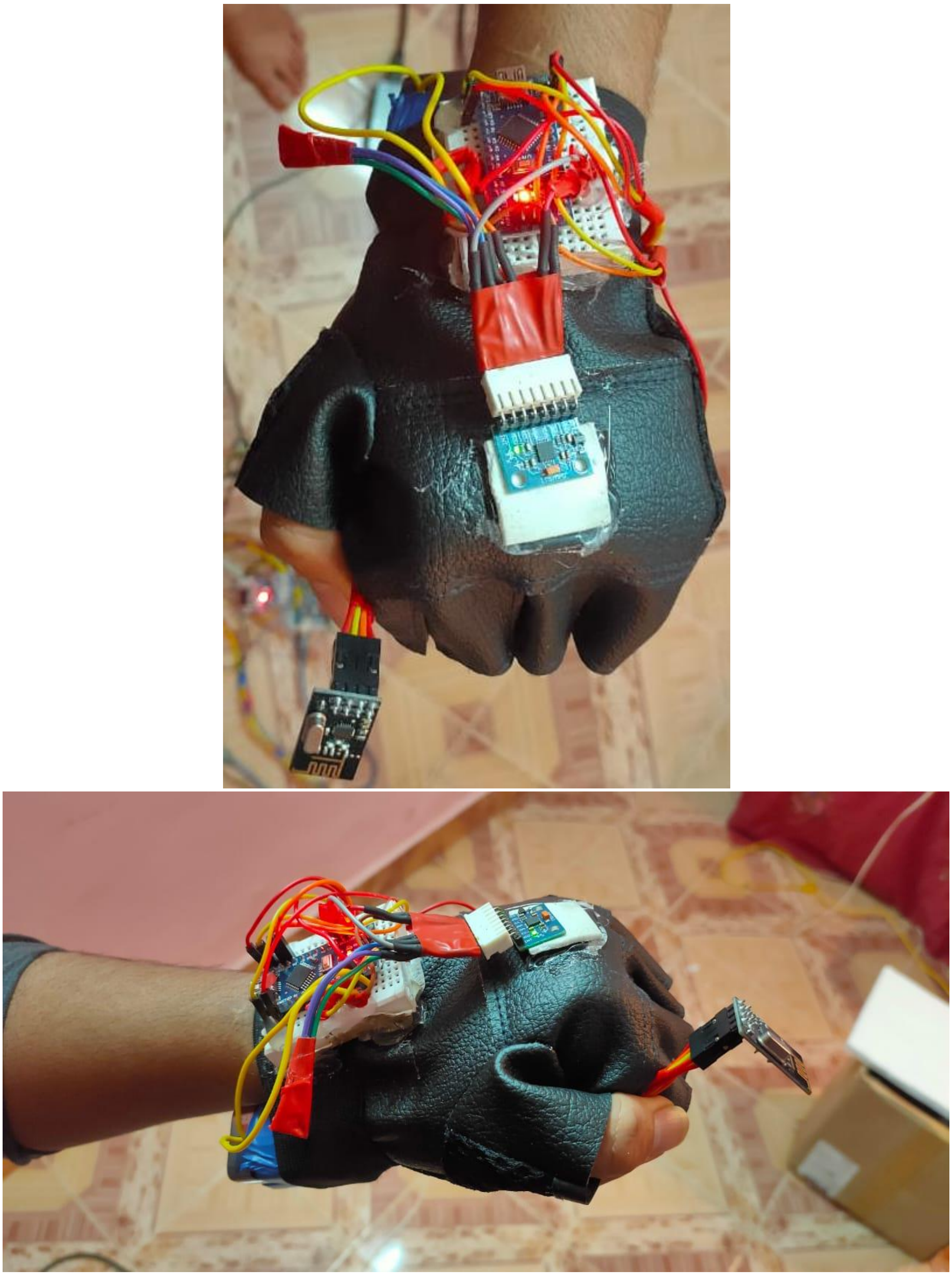

Fig 6.4 Transmitting section on the glove

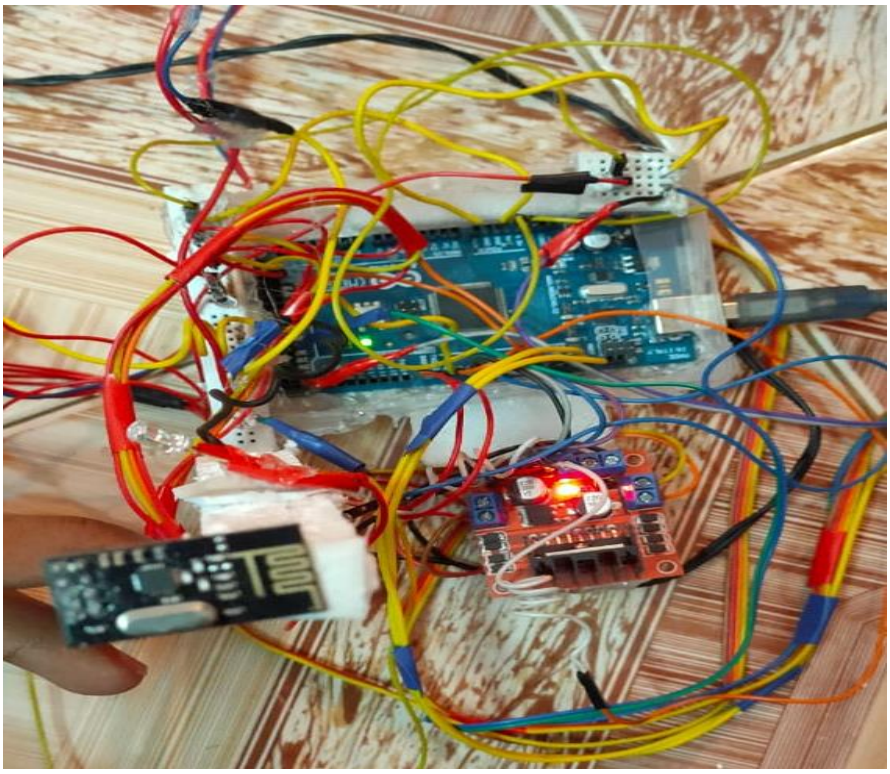

Fig 6.5 Reciever section of the wheelchair

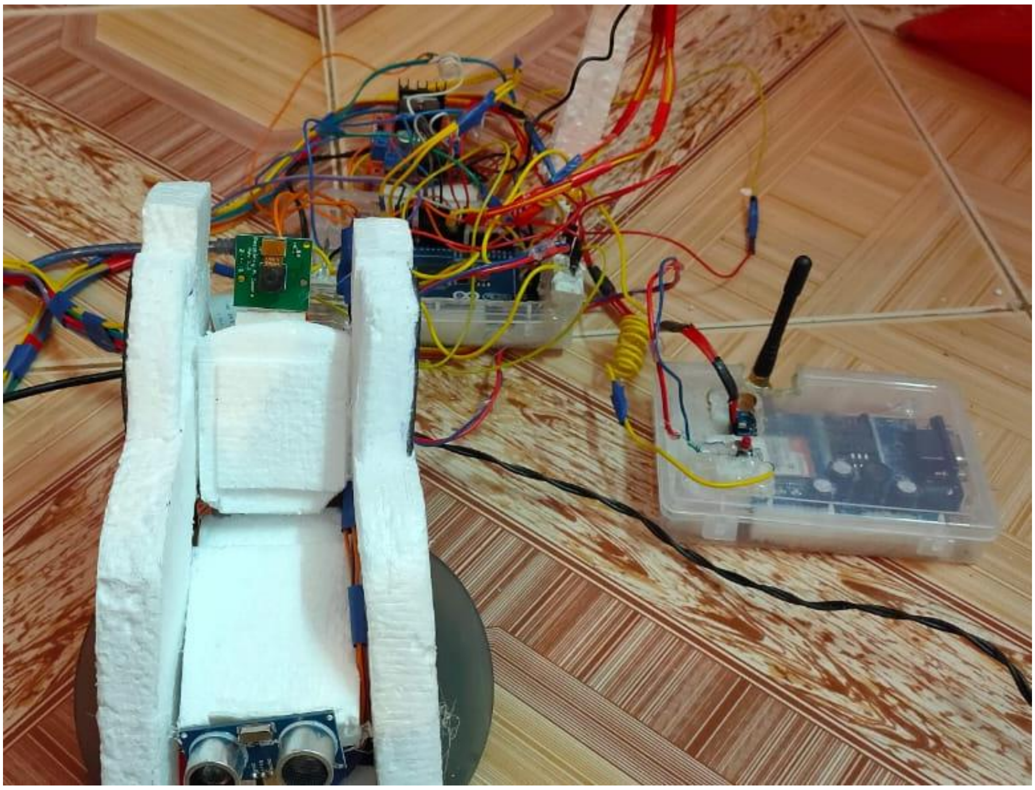

Fig 6.6 Smart wheelchair setup

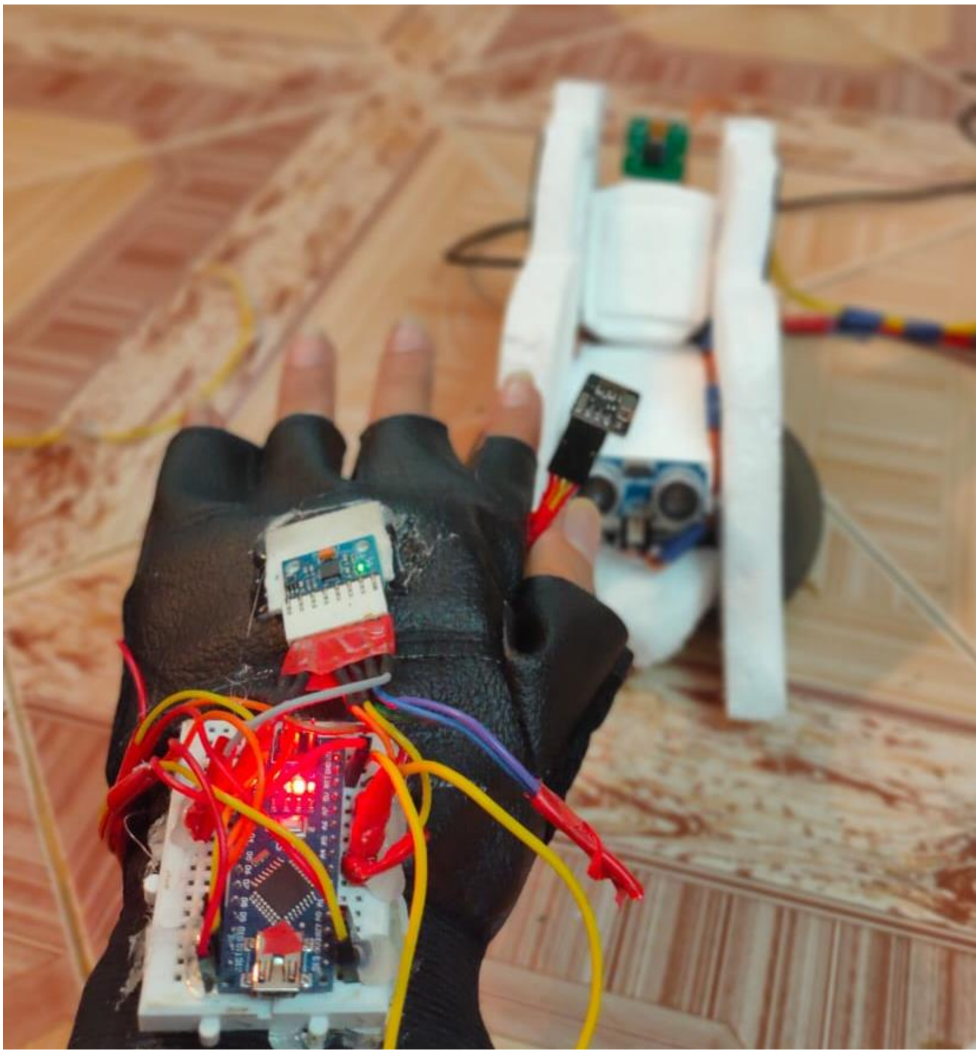

Fig 6.7 Hand movement controlling the wheelchair

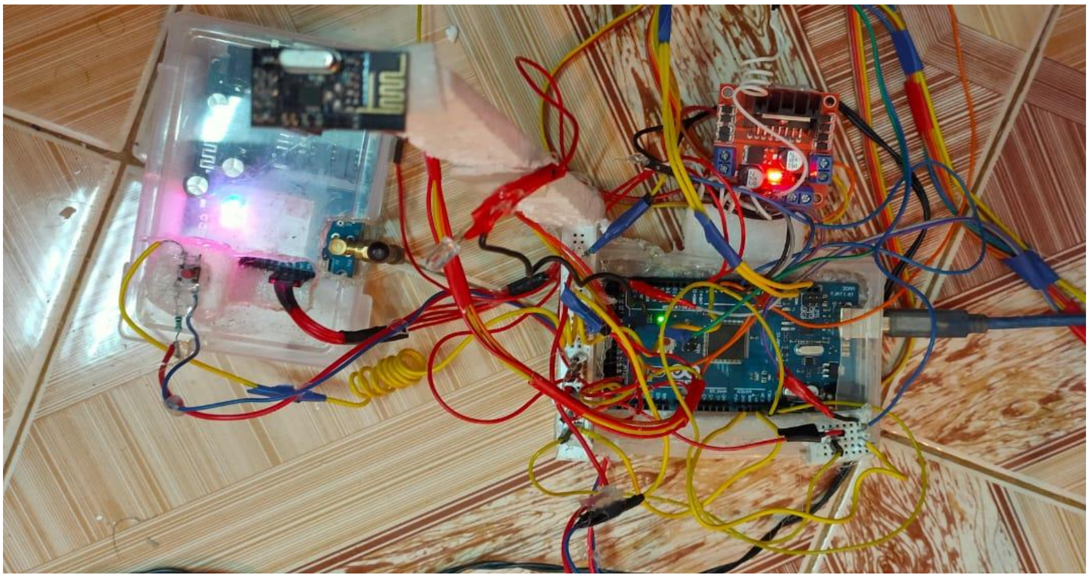

Fig 6.8 Panic button system

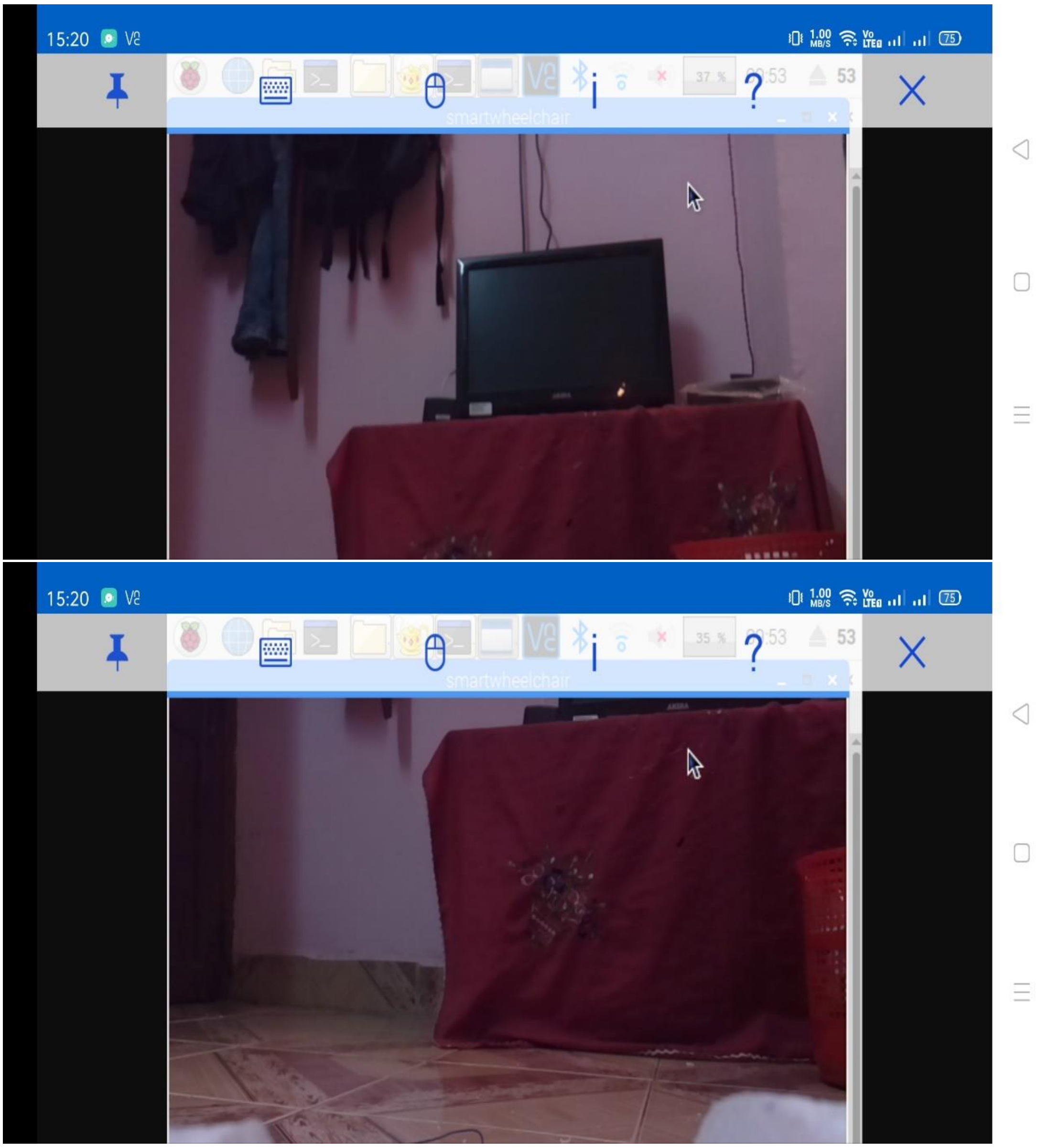

Fig 6.9 Camera input to the mobile for assisting the user

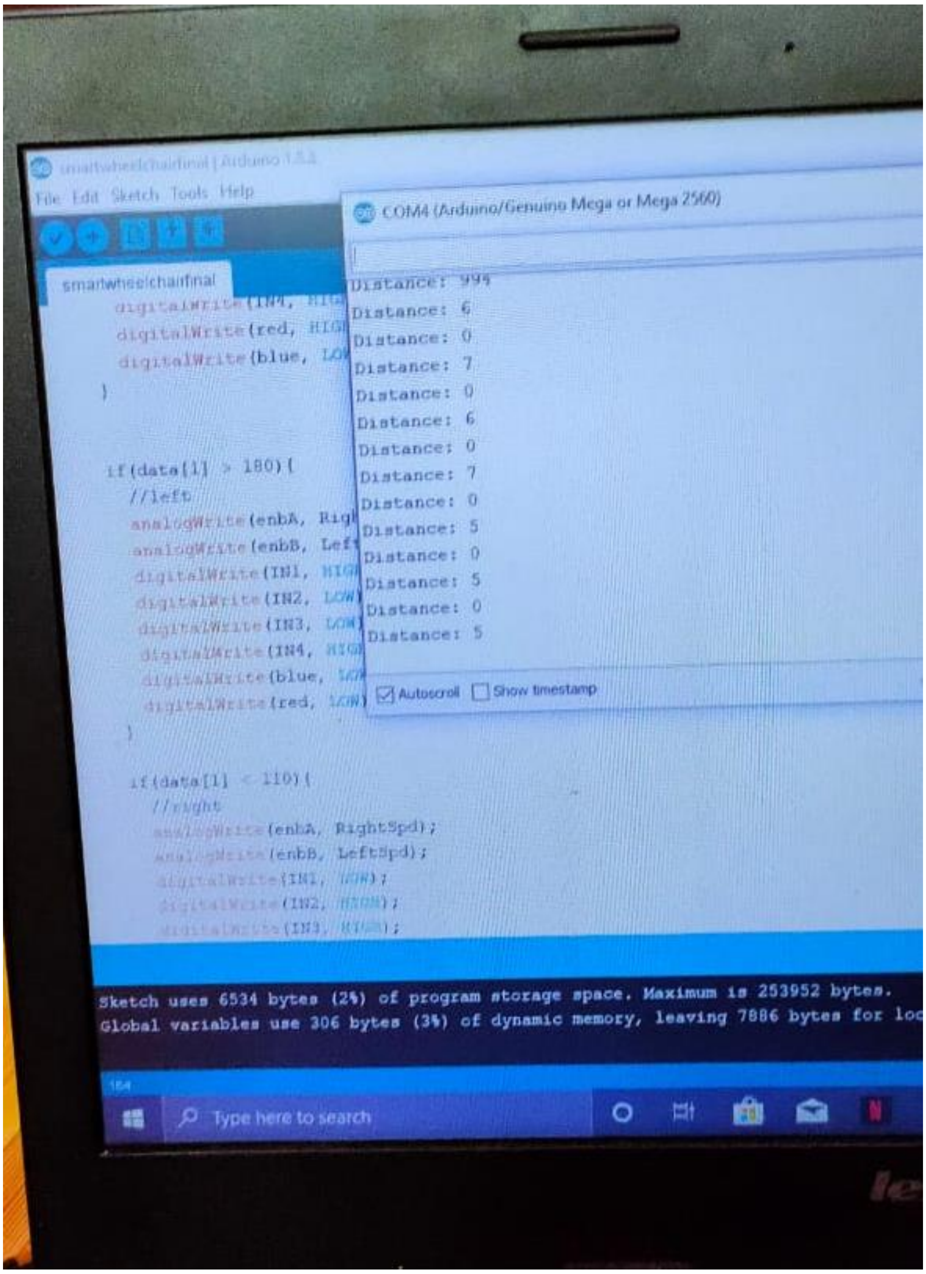

Fig 6.10 Screen showing Distance is less than 10 cm; the wheelchair is stopped

CHAPTER 7: CONCLUSIONS AND FUTURE SCOPE

This project presents different smart technologies for wheelchairs. It focuses the human - machine interface and the navigation methods and devices. Gesture control /hand movement control brings out a great Human machine interface which was implemented successfully. The other features that were implemented include panic button, obstacle avoider and mobile input of the environment when the user is away from the wheelchair.

From the review of many published papers, it is concluded that researchers are continuously trying to build a powerful and helpful wheelchairs to ease the daily life activities and to give more independent mobility for people with different types of disabilities. Unfortunately there are very few commercialized wheelchairs with the smart technology available. One reason is because the robustness and safety of the technology is not 100% guaranteed yet in many researches. Using high-tech smart wheel chairs depends on the severity of the disability, the individual’s overall morale and attitude towards his or her condition and the most important is the price of the technology. Moreover, smart wheelchairs are complicated for many users.

Therefore a simpler approach was tried to build the prototype of the smart wheelchair which was cost efficient and durable.

Future work should maybe focus more on the add-on approach which gives flexibility in configurations of sensors, interface, and input devices based on each individual user’s need s and budget. Other options for making the wheelchair friendlier can be added in future also. For example, entertainment and social communication facilities might be added to the wheelchair. Health monitoring, first aid, muscle relaxing and rehabilitation tools might be considered as useful add-ons too.

Alternate power source Solar panel roof can be used as alternative power source and also it can be a protective layer from rain and sun.

Artificial intelligence and image processing. Artificial intelligence (AI) is technology and a branch of computer science that studies and develops intelligent machines and software. Major AI researchers and textbooks define the field as "the study and design of intelligent agents”, where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success.

APPENDIX

Code for Arduino Mega on the wheelchair:

```
// smartwhellchair GEC ENE


#include <SPI.h>
#include "RF24.h"

const int enbA = 3;
const int enbB = 5;
const int IN1 = 11;    //Right Motor (-)
const int IN2 = 4;    //Right Motor (+)
const int IN3 = 7;    //Left Motor (+)
const int IN4 = 6;    //Right Motor (-)
const int blue = 30;
const int red =  31;
const int trigPin = 12;
const int echoPin = 13;

long duration;
int distance;

const byte ledPin = 32;
volatile byte state = LOW;

int RightSpd = 50;
int LeftSpd = 70;

//Define packet for the direction (X axis and Y axis)
int data[2];

//Define object from RF24 library - 9 and 10 are a digital pin numbers to which signals CE and
CSN are connected
RF24 radio(9,10);

//Create a pipe addresses for the communicate
const uint64_t pipe = 0xE8E8F0F0E1LL;

void setup(){
  setupultra();
  setupwheel();
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
  attachInterrupt(0 , blink, RISING);
}
```

```
void setupultra()
{
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication
}

void setupwheel(){
 //Define the motor pins as OUTPUT
 pinMode(enbA, OUTPUT);
 pinMode(enbB, OUTPUT);
 pinMode(IN1, OUTPUT);
 pinMode(IN2, OUTPUT);
 pinMode(IN3, OUTPUT);
 pinMode(IN4, OUTPUT);
 pinMode(blue,OUTPUT);
 pinMode(red,OUTPUT);

 Serial.begin(9600);
 radio.begin();                //Start the nRF24 communicate
 radio.openReadingPipe(1, pipe);   //Sets the address of the transmitter to which the
program will receive data.
 radio.startListening();
 }

void loop(){
 loopultra();
 loopwheel();
 digitalWrite(ledPin,state);
 if ( state == HIGH)
 {
  state =LOW;
  SendSMS();
 }
}


void blink() {
 state = HIGH;
}
void SendSMS()
{
 Serial.println("AT+CMGF=1");    //To send SMS in Text Mode
 delay(1000);
```

```
 Serial.println("AT+CMGS=\"+918668707467\"\r"); //Change to destination phone number
 delay(1000);
 Serial.println("i need help plz help me ");//the content of the message
 delay(200);
 Serial.println((char)26); //the stopping character Ctrl+Z
 delay(1000);

 }

void loopultra() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}

void loopwheel(){
 if (radio.available()){
  radio.read(data, sizeof(data));

  if(data[0] > 380){
   //forward
   analogWrite(enbA, RightSpd);
   analogWrite(enbB, LeftSpd);
   digitalWrite(IN1, HIGH);
   digitalWrite(IN2, LOW);
   digitalWrite(IN3, HIGH);
   digitalWrite(IN4, LOW);
   digitalWrite(blue, HIGH);
   digitalWrite(red, LOW);
  }

  if(data[0] < 310 ){
   //backward
   analogWrite(enbA, RightSpd);
   analogWrite(enbB, LeftSpd);
   digitalWrite(IN1, LOW);
   digitalWrite(IN2, HIGH);
   digitalWrite(IN3, LOW);
```

```
    digitalWrite(IN4, HIGH);
    digitalWrite(red, HIGH);
    digitalWrite(blue, LOW);
  }


  if(data[1] > 180){
    //left
    analogWrite(enbA, RightSpd);
    analogWrite(enbB, LeftSpd);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    digitalWrite(blue, LOW);
    digitalWrite(red, LOW);
  }

  if(data[1] < 110){
    //right
    analogWrite(enbA, RightSpd);
    analogWrite(enbB, LeftSpd);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    digitalWrite(blue, LOW);
    digitalWrite(red, LOW);
  }

  if((data[0] > 330 && data[0] < 360 && data[1] > 130 && data[1] < 160)|| distance < 10 ){
    //stop car
    analogWrite(enbA, 0);
    analogWrite(enbB, 0);
    digitalWrite(blue, LOW);
    digitalWrite(red, LOW);
  }
 }
}
```

Code for Arduino Nano for the glove:

```
#include <SPI.h>        //SPI library for communicate with the nRF24L01+
#include "RF24.h"       //The main library of the nRF24L01+
#include "Wire.h"       //For communicate
#include "I2Cdev.h"     //For communicate with MPU6050
#include "MPU6050.h"    //The main library of the MPU6050
```

```
//Define the object to access and cotrol the Gyro and Accelerometer (We don't use the Gyro data)
MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;

//Define packet for the direction (X axis and Y axis)
int data[2];

//Define object from RF24 library - 9 and 10 are a digital pin numbers to which signals CE and CSN are connected.
RF24 radio(9,10);

//Create a pipe addresses for the communicate
const uint64_t pipe = 0xE8E8F0F0E1LL;

void setup(void){
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();              //Initialize the MPU object
  radio.begin();                 //Start the nRF24 communicate
  radio.openWritingPipe(pipe);   //Sets the address of the receiver to which the program will send data.
}

void loop(void){

  //With this function, the acceleration and gyro values of the axes are taken.
  //If you want to control the car axis differently, you can change the axis name in the map command.
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  //In two-way control, the X axis (data [0]) of the MPU6050 allows the robot to move forward and backward.
  //Y axis (data [0]) allows the robot to right and left turn.
  data[0] = map(ax, -17000, 17000, 300, 400 ); //Send X axis data
  data[1] = map(ay, -17000, 17000, 100, 200);  //Send Y axis data
  radio.write(data, sizeof(data));
}
```

C++ Code for Raspberry Pi:

```
#include <opencv2/opencv.hpp>
#include <raspicam_cv.h>
#include <iostream>
```

```
#include <chrono>
#include <ctime>
#include <opencv2/imgproc/imgproc.hpp>

using namespace std;
using namespace cv;
using namespace raspicam;

Mat frame;
RaspiCam_Cv Camera;

void Setup ( int argc,char **argv, RaspiCam_Cv &Camera )
 {
   Camera.set ( CAP_PROP_FRAME_WIDTH,  ( "-w",argc,argv,640 ) );
   Camera.set ( CAP_PROP_FRAME_HEIGHT,  ( "-h",argc,argv,480) );
   Camera.set ( CAP_PROP_BRIGHTNESS, ( "-br",argc,argv,50 ) );
   Camera.set ( CAP_PROP_CONTRAST ,( "-co",argc,argv,50 ) );
   Camera.set ( CAP_PROP_SATURATION,  ( "-sa",argc,argv,50 ) );
   Camera.set ( CAP_PROP_GAIN,  ( "-g",argc,argv ,50 ) );
   Camera.set ( CAP_PROP_FPS,  ( "-fps",argc,argv,0));

}


int main(int argc,char **argv)
{



   Setup(argc, argv, Camera);
   cout<<"Connecting to camera"<<endl;
   if (!Camera.open())
   {

          cout<<"Failed to Connect"<<endl;
   }

          cout<<"Camera Id = "<<Camera.getId()<<endl;


          while(1)
          {
                    Camera.grab();
                    Camera.retrieve(frame);
                    imshow("smartwheelchair",frame);
                    waitKey(1);
          }
          return 0 ;
}
```

REFERENCES

Smart wheelchairs: A literature review Richard C. Simpson, PhD, ATP* Department of Rehabilitation Science and Technology, University of Pittsburgh, Pittsburgh, PA

SMART WHEEL CHAIR by Rohit Singh, Ramu Raut and Sabin Adhikari Kathmandu, Nepal.

G. Bourhis, K. Moumen, P. Pino, S. Rohmer and A. Pruski, "Assisted navigation for a powered wheelchair. Systems Engineering in the Service of Humans", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, France, pp. 553–558, 1993.

https://www.instructables.com/id/Interfacing-SIM900A-GSM-Modem-with-Arduino/

https://mechatrofice.com/arduino/send-gps-location-via-sms

https://electrosome.com/controlling-motors-mpu-6050/

https://how2electronics.com/dc-motor-control-using-mpu6050-gyro-accelerometer/