A PROJECT REPORT ON

NAVIGATION ENABLED RECONAAISANCE DEVICE

GUJARAT TECHNOLOGICAL UNIVERSITY

Dr, S. & S. S. GHANDHY COLLEGE OF ENGINEERING AND TECHNOLOGY
MAJURA GATE, SURAT

A PROJECT REPORT ON

"NAVIGATION ENABLED RECONNAISANCE DEVICE"

Submitted by

NAME	***ENROLLMENT NO.***
1. PAWAR YASH A.	186120309045
2. PATEL NACHIKET N.	186120309035
3. GANDHI PARTH P.	186120309026
4. PATRA ASHISH S.	186120309044
5. SHAIKH SAHEBAZ S.	186120309048
6. PATEL BRIJESH N.	186120309028

UNDER THE GUIDANCE OF:

Prof. Dharmvir Prasad

Submitted to

Gujarat Technological University

In Partial Fulfillment of the Requirement for Project-1 (3350908)

The Diploma of Engineering

in

ELECTRICAL ENGINEERING

2020-2021

Dr. S. & S.S. Ghandhy College of Engineering & Technology, Surat

CERTIFICATE

This is to certify that Final Project embodied in this report entitled **"Navigation Enabled RECONNAISANCE Device "** was carried out by,

Sr. No.	Name of The Student	Enrollment No.
1	PAWAR YASH A.	186120309045
2	PATEL NACHIKET N.	186120309035
3	GANDHI PARTH P.	186120309026
4	PATRA ASHISH S.	186120309044
5	SHAIKH SAHEBAZ S.	186120309048
6	PATEL BRIJESH N.	186120309028

at Dr. S & S.S. Ghandhy College of Engineering & Technology**,** Surat for partial fulfillment of Project-1 (3350908) Diploma in Electrical Engineering, to be awarded by Gujarat Technological University. This project work has been carried out under our supervision and is to the satisfaction of department. The student work has been accepted during the academic year 2020-2021.

DATE:

PLACE:

Prof. Dharmvir Prasad
FACULTY GUIDE

Prof. Krishna Mehta
HEAD OF DEPARTMENT
ELECTRICAL ENGG.

ACKNOWLEDGEMENT

It is a matter of extreme honor and privilege for us to offer our grateful acknowledgement to our guides **Prof. Dharmvir Prasad , Dr. S & S.S. Ghandhy College of Engineering & Technology, Surat** for providing us a chance to work under their guidance and supervision.

We express our sincere appreciation and thanks to **Prof. N. A. Sangani (Principal, S & S.S. Ghandhy College of Engineering & Technology, Surat SURAT)** and **Prof. K. B. Mehta (HOD, Electrical Engg. Dept.)** for constant encouragement and needful help during various stages of work. We are thankful to the institute for providing all kind of required resources.

We appreciate all our colleagues whose direct and indirect contribution helped us a lot to accomplish this report and who made the period of project more pleasant and fruitful. We would also like to thank all the teaching and non-teaching staff for cooperating with us and providing valuable advice which helped us in the completion of this report.

Finally, we would like to express our deepest thanks to all the members of our family, who gives us strength and support to aspire for this level of education.

Sr. No.	**Name of The Student**	**Enrollment No.**	**SIGN**
1	PAWAR YASH A.	186120309045	
2	PATEL NACHIKET N.	186120309035	
3	GANDHI PARTH P.	186120309026	
4	PATRA ASHISH S.	186120309044	
5	SHAIKH SAHEBAZ S.	186120309048	
6.	PATEL BRIJESH N.	186120309028	

INDEX

ABSTRACT

Now a days countries are advancing on a great scale in defense sector to safeguard themselves from incoming threats. Reconnaissance plays a very crucial role in any mission.It is proves vital in determining threats and blind-spots on hostile territories.

Military personnel's risk their extremely valuable lives while carrying out reconnaissance on foot. To rectify this the Navigation Enabled Reconnaissance Device (N.E.R.D.)comes into the picture. This U.G.V. (Unmanned Ground Vehicle) can cruise on rough hostile terrains and can be operated from a safe distance at a base station. This device will greatly help the military personnel in safe guarding the nation.

CHAPTER 1 : INTRODUCTION

CHAPTER 1: INTRODUCTION

1.1 DESCRIPTION

Defense sector is evolving day by day. We can see many advancements in defense equipment like precision guided missiles, autonomous drones and advanced nuclear warheads. This equipment will help in warfare but before the warfare starts the militarily agencies have to collect intel on hostile enemy territories for reconnaissance of the field becomes crucial.

To carry out these reconnaissance and surveillance missions Unmanned aerial Vehicles (UAV) and Unmanned Ground Vehicles (UGV) are becoming the first choice of emerging superpowers like India. The UAV’s are good for aerial reconnaissance where as an UGV is a vehicle that operates while keeping ground contact and without any on-board operator presence. UGVs are used for applications where it is inconvenient or perilous to have a human operator on-board. UGVs are perfect for ground applications. Countries are investing huge capital to develop these technologies.

The purpose of this project is to develop a UGV which can be manipulated by the military personnel for their gain. The personnel can use this this device to carry out reconnaissance missions to collect intel during war time to collect intel and for patrolling & surveillance purpose.

1.2 FEATURES

- Navigation capability
- Payload delivery
- Equipped with camera & microphone

CHAPTER 2: COMPONENTS WITH DESCRIPTION

CHAPTER 2: COMPONENTS

2.1 OUTER COMPONENT DETAILS

The NERD uses the following external components :-

- **THE CHASSIS**

The chassis of the UGV will be made from alluminum as it is light in weight and extremely durable in harsh environments.

- **WHEELS**

90.2 diameter rubber wheels will be used for manuevering over rough terrain.

2.2 THE HARDWARE

2.2.1 THE CENTRALIZED CONTROL SYSTEM

The Centralized Control System consists of the microcontrollers and other various sensors & actuators. All thesse are discussed in upcoming topics.

- **MICROCINTROLLER PLATFORM**

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 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 a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

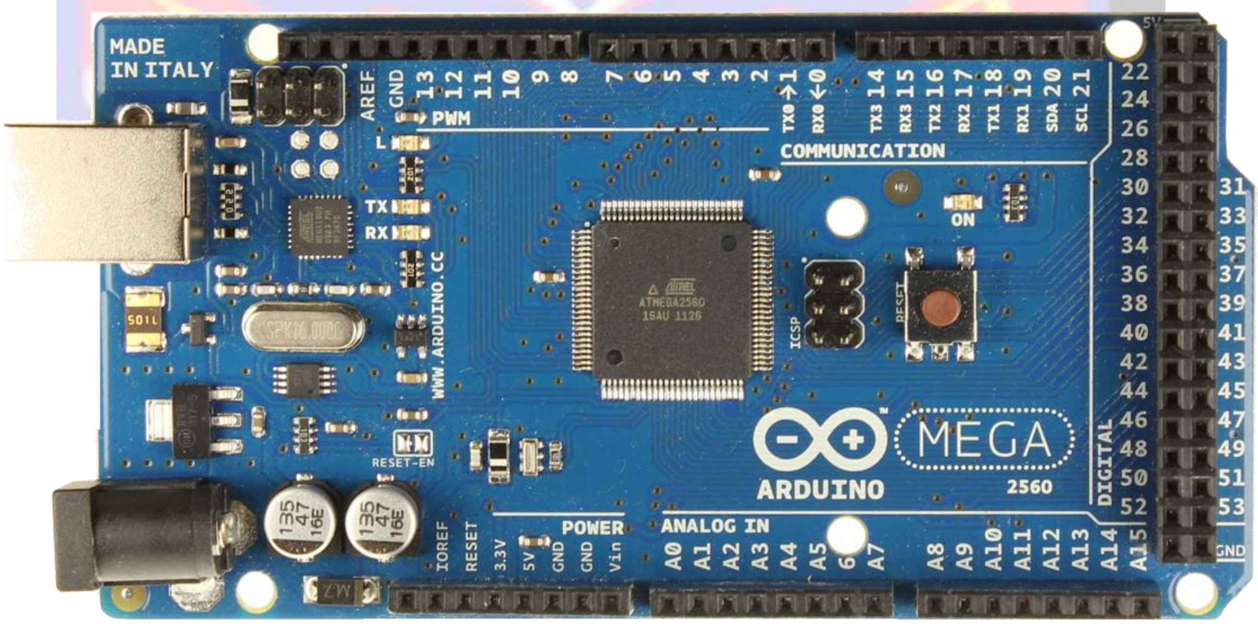

Fig-1 ARDUINO MEGA 2560

I. **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

II. **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

III. **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 20-50 kΩ. 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. See the attachInterrupt() function for details.

• 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. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.

• **LED: 13**. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

• **I 2C: 20 (SDA) and 21 (SCL). Support I2C (TWI)** communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove.

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.

IV. 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 (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Mega's digital pins. The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

V. TECHNICAL SPECIFICATIONS

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins 54	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	8KB
EEPROM	4 KB
Clock Speed	16 MHz

Table-1 Technical specifications for Arduino Mega 2560

2.2.2 VIDEO FEEDING SYSTEM FOR SURVELLINCE

- **ESP32 CAM**

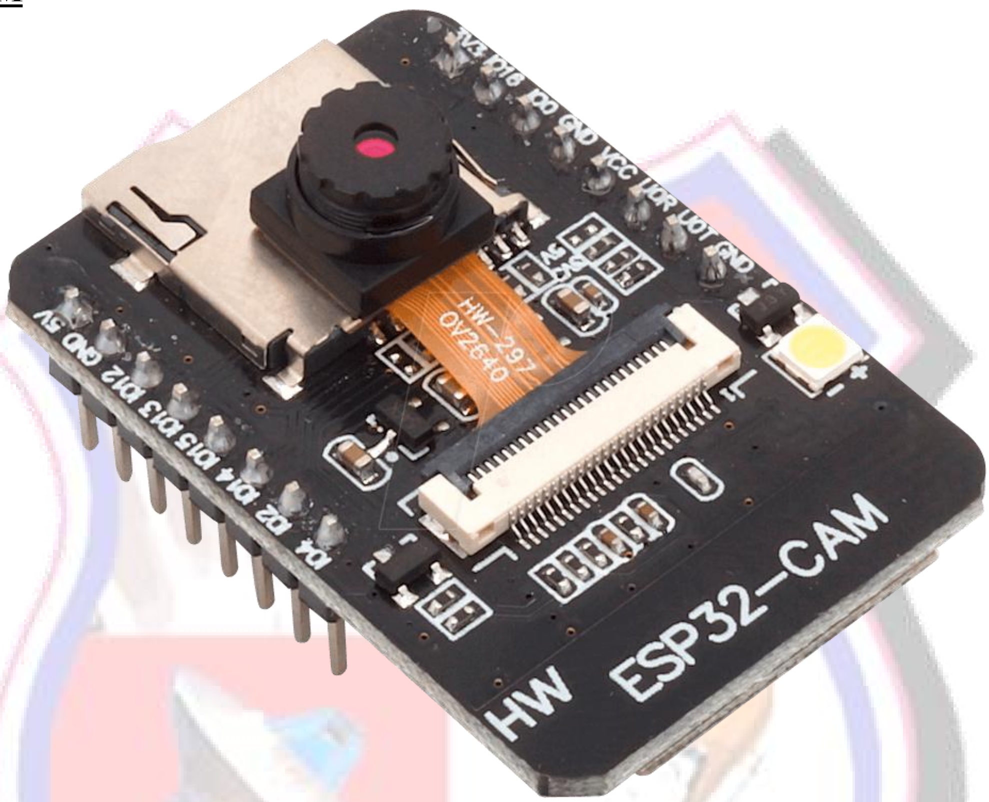

Fig-2 ESP32-CAM

Fig-1 ESP32-CAM Pin layout

ESP32-CAM is a low cost development board with WiFi camera. It allows creating IP camera projects for video streaming with different resolutions. ESP32-CAM has build in PCB antenna. This camera module can feed live surveillance footage to the base station the base station. It has virtues compact form factor, low power consumption camera module based on ESP32. It comes with an OV2640 camera and provides onboard TF card slot.

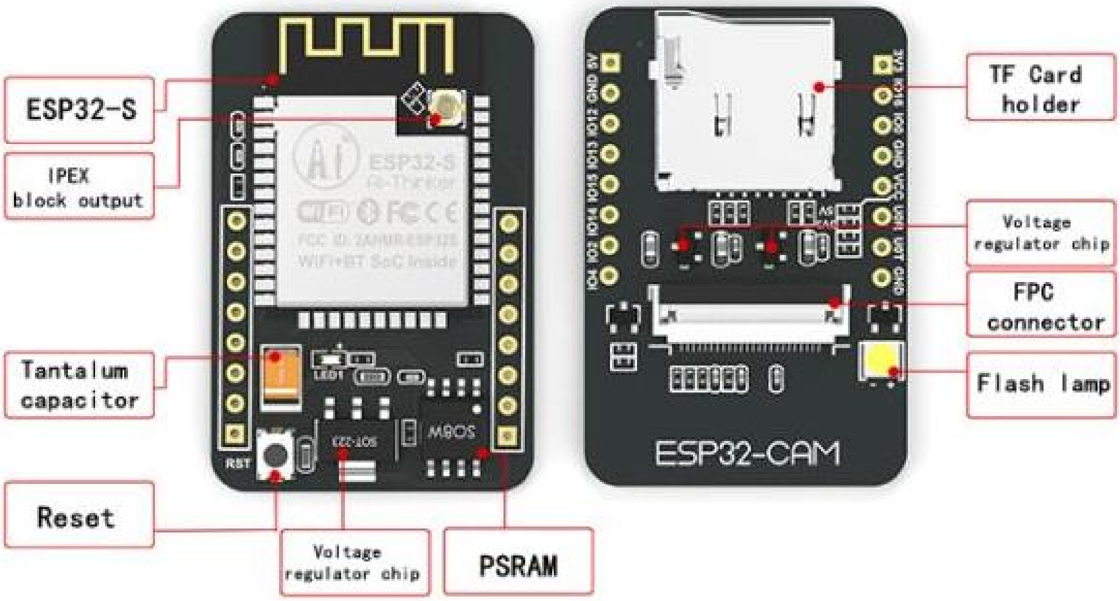

Fig-4 ESP32-CAM Parts

TECHNICAL SPECIFICATIONS:-

WIFI module: ESP-32S
Processor: ESP32-D0WD
Built-in Flash: 32Mbit
RAM: Internal 512KB + External 4M
PSRAM Antenna: On-board PCB antenna
WiFi protocol: IEEE 802.11 b/g/n/e/i
Bluetooth: Bluetooth 4.2 BR/EDR and BLE
WIFI mode: Station / SoftAP / SoftAP+Station
Security: WPA/WPA2/WPA2-Enterprise/WPS
Output image format: JPEG (OV2640 support only), BMP, GRAYSCALE
Supported TF card: up to 4G
Peripheral interface: UART/SPI/I2C/PWM
IO port: 9 UART
baudrate rate: default 115200bps
Power supply: 5V
Transmitting power:
802.11b: 17 ±2dBm(@11Mbps)
802.11g: 14 ±2dBm(@54Mbps)
802.11n: 13 ±2dBm(@HT20,MCS7)
Receiving sensitivity: CCK,1Mbps: -90 dBm
CCK,11Mbps: -85 dBm
6Mbps(1/2 BPSK): -88 dBm
54Mbps(3/4 64-QAM): -70 dBm

HT20,MCS7(65Mbps, 72.2Mbps): -67 dBm

Power consumption:

Flash off: 180mA@5V Flash on and brightness max: 310mA@5V
Deep-Sleep: as low as 6mA@5V
Modern-Sleep: as low as 20mA@5V
Light-Sleep: as low as 6.7mA@5V
Operating temperature: -20 ℃ ~ 85 ℃
Storage environment: -40 ℃ ~ 90 ℃

Dimensions: 40.5mm x 27mm x 4.5mm

- **BATTERIES**

Fig-5 Li-Po Battery

The UGV will possess a 2200 mAh Lithium Polymer battery to power the Centralized Control System(CMS). The main reason for using a Lithium Polymer (Li-Po) battery cells is that they have about four times the energy density than that of nickel cadmium or nickel metal hydride batteries. Li-Po batteries are very lightweight and pliable, and they are available in a very small & compact form factor.

TECHNICAL SPECIFICATIONS

Model No:	ORANGE 2200/3S-30C
Weight:	175.0g
Voltage :	11.1V
Dimensions :	23x34x106(mm)

Table-2 Technical specifications for Li-Po battery

➢ **GPS MODULE**

The UGV uses a GPS module to feed latitudinal and longitudinal coordinates to the base station with high precision and accuracy.

2.3 TRACTION & TRACTION CONTROL

❖ **TRACTION**

➢ **MOTORS**

As the UGV requires high torque for propelling on ground, the characteristics of High Torque D.C. Motor fits perfectly in this criteria. So for traction purpose High torque D.C. motors are selected. These motors produce a holding torque of around 13.59 kgcm.

The UGV uses the Four Wheel Drive Technology (4 WD). In this technology all the four wheels of the vehicles are driven by four individual motors for better and efficient traction and its control.

The chart below shows on Y axis the speed, efficiency, output power, motor current vs the X axis which is the motor torque load with no load on the left and max load on the right.

2.2.3 D.C. MOTOR CHARACTERISTICS

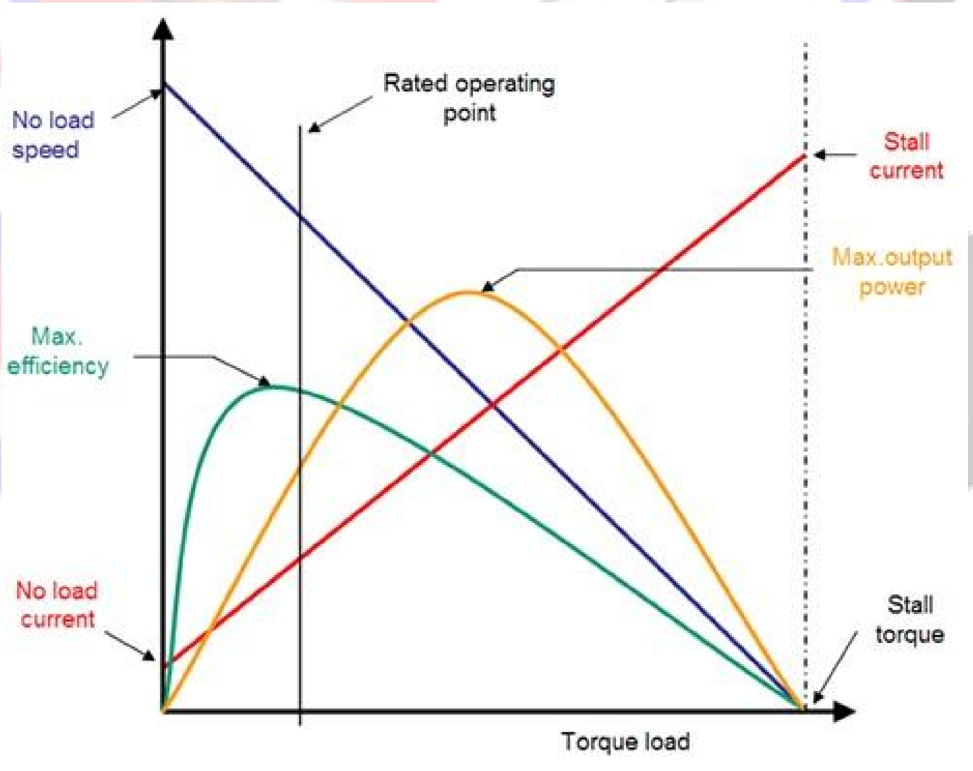

Fig-6 D.C. Motor Characteristics

The maximum power (i.e. horsepower) out is at approximately half the no load rpm whereas the maximum efficiency is at nearly 90% of the no load speed. Motors and typical rated for the maximum output power and operated at 50-70% of that for reliable performance.

One characteristic of a 12v DC motor is the operating voltage. When a motor is powered by batteries, low operating voltages are typically preferred since fewer cells are required to obtain the specified voltage.

Fig-7 High Torque D.C. Motor

TECHNICAL SPECIFICATIONS

Motor RPM	**200 RPM**
Voltage	**12 V D.C.**
Gearbox Diameter	**37 mm**
Motor Diameter	**28.5 mm**
Length	**63 mm (without shaft)**
Shaft Length	**30 mm**
Overall Weight	**200 gms**
Holding Torque	**13.59 kgcm**
Rated Torque	**5.2 kgcm**
No-load current	**800 mA**
Load Current	**upto 7.5 Amp (Max)**
Gear Ratio	**1:90**

Table-2 Technical specifications for High Torque D.C, Motor

❖ TRACTION CONTROL

➢ MOTOR DRIVER MODULE

A BTS-7960B 43A H - bridge motor driver module is used for controlling the motor speed using Pulse Width Modulation(PWM) Technique.

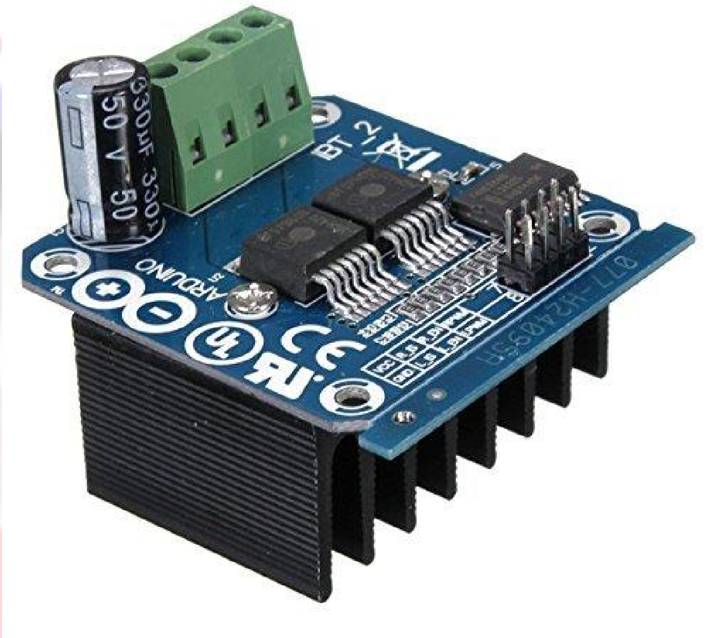

Fig-8 BTS-7960B Motor Driver Module

Pulse-width modulation (PWM), as it applies to motor control, is a way of delivering energy through a succession of pulses rather than a continuously varying (analog) signal. By increasing or decreasing pulse width, the controller regulates energy flow to the motor shaft. The motor's own inductance acts like a filter, storing energy during the "ON" cycle while releasing it at a rate corresponding to the input or reference signal. In other words, energy flows into the load not so much the switching frequency, but at the reference frequency.

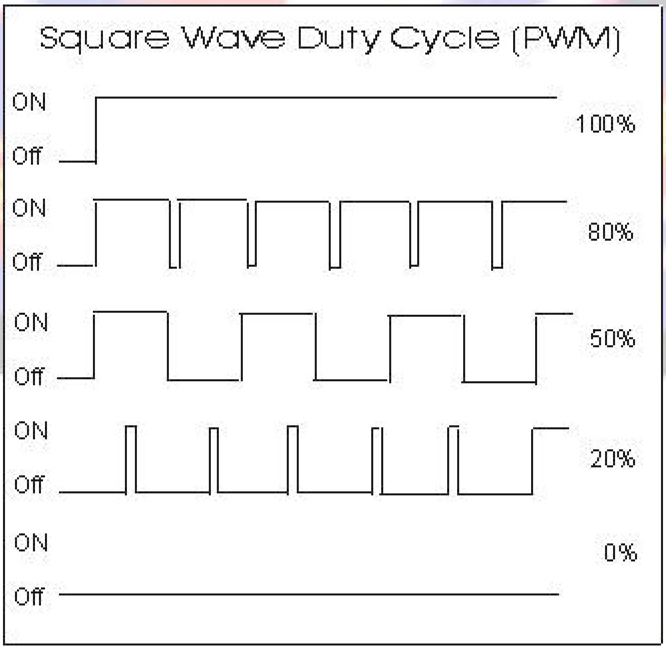

Fig-9 Square Wave Duty Cycle (PWM)

The BTS 7960B is a fully integrated high current half bridge for motor drive applications. It is part of the NovalithICTM family containing one p-channel high side MOSFET and one n-channel low side MOSFET with an integrated driver IC in one package. The power switches utilize vertical MOS technologies to ensure optimum on state resistance. Due to the p-channel highside switch the need for a charge pump is eliminated thus minimizing EMI. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against over-temperature, over-voltage, under-voltage, over-current and short circuit. The BTS 7960B provides a cost optimized solution for protected high current PWM motor drives with very low board space consumption.

TECHNICAL SPECIFICATIONS

Operating Supply Voltage:	5.5 V to 27.5 V
Operating Supply Current:	3 mA
Minimum Operating Temperature:	- 40 C
Maximum Operating Temperature:	+ 150 C
Mounting Style:	SMD/SMT
Package/Case:	TO-263-7
Packaging:	Reel
Brand:	Infineon Technologies
Product Type:	Motor / Motion / Ignition Controllers & Drivers

Table-1 Technical specifications for
BTS - 7960B Motor driver Module

❖ **VELICLE MANIPULATION**

- This vehicle can be manipulated by using a long distance radio transmitter and receiver. The transmitter can control the directions of movements. The following is a 6 channel 2.4Ghz Transmitter and Receiver kit.

Fig-10 2.4 GHz 6CH Transmitter

Fig-11 GHz 6CH Receiver

CHAPTER 3: CIRCUIT DIAGRAMS & LAYOUTS

CHAPTER 3: CIRCUIT DIAGRAM & LAYOUTS

3.1 BLOCK DIAGRAM

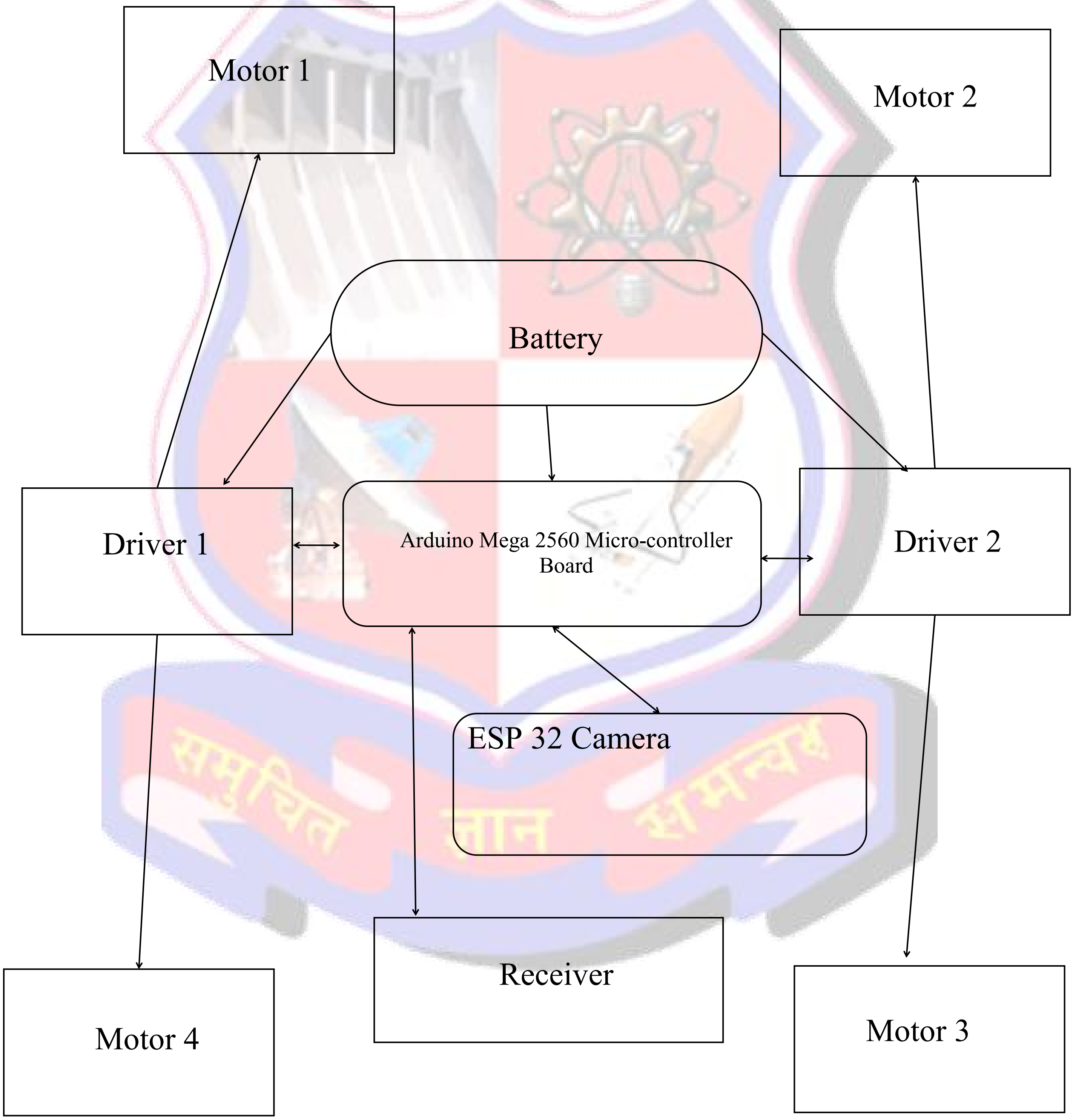

Fig-10 Block Diagram

3.1 CONNECTION LAYOUT

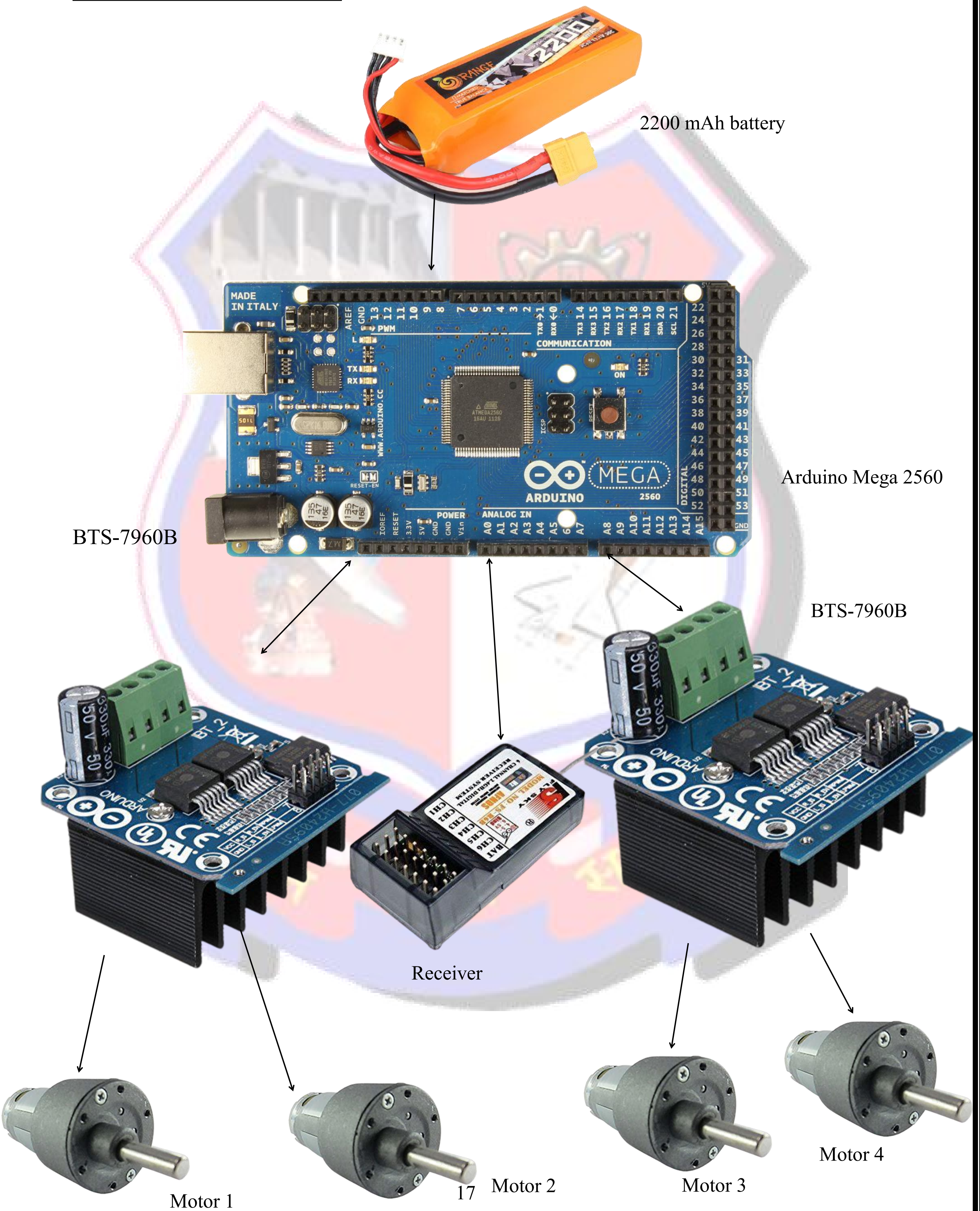

Fig-10 Connection Layout

3.1 CIRCUIT DIAGRAM

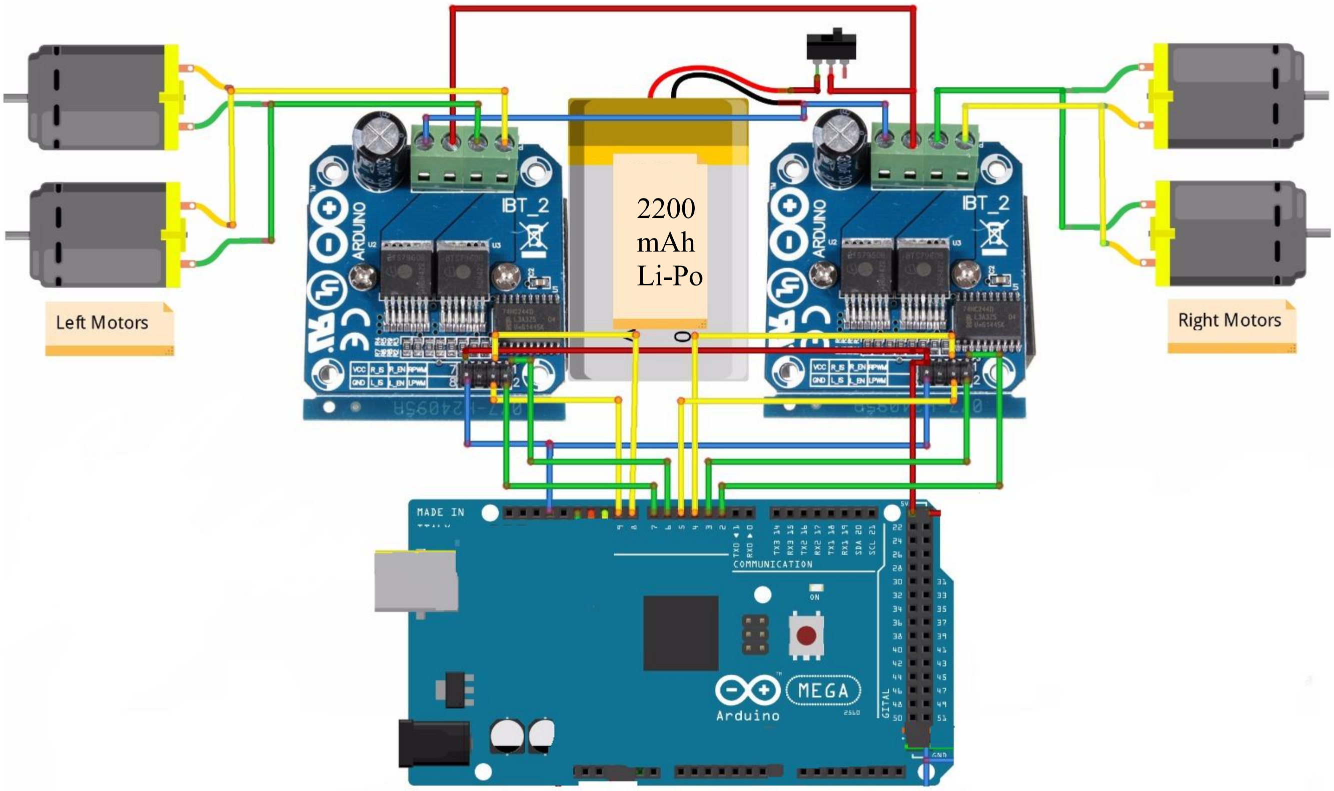

Fig-10 Connection Layout

CHAPTER 4: THE SOURCE CODE

CHAPTER 5: PROGRAM OF PROJECT

4.1 PROGRAM :-

```
/*

        Connections:
        BTS7960 -> Arduino Mega 2560
        MotorRight_R_EN - 22
        MotorRight_L_EN - 23
        MotorLeft_R_EN - 26
        MotorLeft_L_EN - 27
        Rpwm1 - 2
        Lpwm1 - 3
        Rpwm2 - 4
        Lpwm2 - 5

          FlySky FS 2.4GHz Receiver -> Arduino Mega 2560
        ch2 - 7 // Aileron
        ch3 - 8 // Elevator
        */
        #define LOWER_STOP_RANGE_MOVE -20
        #define UPPER_STOP_RANGE_MOVE 20
        #define LOWER_STOP_RANGE_TURN -20
        #define UPPER_STOP_RANGE_TURN 20

        /*BTS7960 Motor Driver Carrier*/
        const int MotorRight_R_EN = 22;
        const int MotorRight_L_EN = 23;

        const int MotorLeft_R_EN = 26;
        const int MotorLeft_L_EN = 27;

        const int Rpwm1 = 2;
        const int Lpwm1 = 3;
        const int Rpwm2 = 4;
        const int Lpwm2 = 5;
        long pwmLvalue = 255;
        long pwmRvalue = 255;
        byte pwmChannel;
        int robotControlState;
        int last_mspeed;
```

```
boolean stop_state = true;

// MODE2
int ch1; // Throttle
int ch2; // Aileron
int ch3; // Elevator
int ch4; // Rudder

int moveValue;
int turnValue;

void setup(){
//pinMode(6, INPUT);
pinMode(7, INPUT);
pinMode(8, INPUT);
//pinMode(9, INPUT);
Serial.begin(9600);

//Setup Right Motors
pinMode(MotorRight_R_EN, OUTPUT); //Initiates Motor Channel A1 pin
pinMode(MotorRight_L_EN, OUTPUT); //Initiates Motor Channel A2 pin

//Setup Left Motors
pinMode(MotorLeft_R_EN, OUTPUT); //Initiates Motor Channel B1 pin
pinMode(MotorLeft_L_EN, OUTPUT); //Initiates Motor Channel B2 pin

//Setup PWM pins as Outputs
pinMode(Rpwm1, OUTPUT);
pinMode(Lpwm1, OUTPUT);
pinMode(Rpwm2, OUTPUT);
pinMode(Lpwm2, OUTPUT);

stop_Robot();
}// void setup()

void loop() {
//ch1 = pulseIn(6, HIGH, 25000); // Read the pulse width of each channel
ch2 = pulseIn(7, HIGH, 25000);
ch3 = pulseIn(8, HIGH, 25000);
//ch4 = pulseIn(9, HIGH, 25000);
```

```
moveValue = map(ch3, 980, 1999, -255, 255);
moveValue = constrain(moveValue, -255, 255);
turnValue = map(ch2, 980, 1999, -255, 255);
turnValue = constrain(turnValue, -255, 255);

//Serial.println("moveValue: "+String(moveValue)+ ", turnValue: "+String(turnValue));
if (moveValue>LOWER_STOP_RANGE_MOVE && moveValue<UPPER_STOP_RANGE_MOVE && turnValue>LOWER_ST
if(stop_state == false){
stop_Robot();
stop_state = true;
Serial.println("Stop Robot");
}
}
//GO FORWARD & BACKWARD
else if(turnValue>LOWER_STOP_RANGE_TURN && turnValue<UPPER_STOP_RANGE_TURN){
if(moveValue>UPPER_STOP_RANGE_MOVE){
go_Forward(moveValue);
stop_state = false;
Serial.println("Go Forward");
}
else if(moveValue<LOWER_STOP_RANGE_MOVE){
go_Backwad(abs(moveValue));
stop_state = false;
Serial.println("Go Backward");
}
}
//TURN RIGHT & LEFT
else if(moveValue>LOWER_STOP_RANGE_MOVE && moveValue<UPPER_STOP_RANGE_MOVE){
if(turnValue>UPPER_STOP_RANGE_TURN){
turn_Right(turnValue);
stop_state = false;
Serial.println("Turn Right");
}
else if(turnValue<LOWER_STOP_RANGE_TURN){
turn_Left(abs(turnValue));
stop_state = false;
Serial.println("Turn Left");
}
}
delay(200);
}
```

```
void stop_Robot(){ // robotControlState = 0
if(robotControlState!=0){
//SetMotors(2);
analogWrite(Rpwm1, 0);
analogWrite(Lpwm1, 0);
analogWrite(Rpwm2, 0);
analogWrite(Lpwm2, 0);
robotControlState = 0;
}
}// void stopRobot()

void turn_Right(int mspeed){ // robotControlState = 1
if(robotControlState!=1 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, 0);
analogWrite(Lpwm1, mspeed);
analogWrite(Rpwm2, mspeed);
analogWrite(Lpwm2, 0);
robotControlState=1;
last_mspeed=mspeed;
}
}// void turn_Right(int mspeed)

void turn_Left(int mspeed){ // robotControlState = 2
if(robotControlState!=2 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, mspeed);
analogWrite(Lpwm1, 0);
analogWrite(Rpwm2, 0);
analogWrite(Lpwm2, mspeed);
robotControlState=2;
last_mspeed=mspeed;
}
}// void turn_Left(int mspeed)

void go_Forward(int mspeed){ // robotControlState = 3
if(robotControlState!=3 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, mspeed);
analogWrite(Lpwm1, 0);
```

```
analogWrite(Rpwm2, mspeed);
analogWrite(Lpwm2, 0);
robotControlState=3;
last_mspeed=mspeed;
}
}// void goForward(int mspeed)

void go_Backwad(int mspeed){ // robotControlState = 4
if(robotControlState!=4 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, 0);
analogWrite(Lpwm1, mspeed);
analogWrite(Rpwm2, 0);
analogWrite(Lpwm2, mspeed);
robotControlState=4;
last_mspeed=mspeed;
}
}// void goBackwad(int mspeed)

void move_RightForward(int mspeed){ // robotControlState = 5
if(robotControlState!=5 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, mspeed*0.4);
analogWrite(Lpwm1, 0);
analogWrite(Rpwm2, mspeed);
analogWrite(Lpwm2, 0);
robotControlState=5;
last_mspeed=mspeed;
}
}// void move_RightForward(int mspeed)

void move_LeftForward(int mspeed){ // robotControlState = 6
if(robotControlState!=6 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, mspeed);
analogWrite(Lpwm1, 0);
analogWrite(Rpwm2, mspeed*0.4);
analogWrite(Lpwm2, 0);
robotControlState=6;
last_mspeed=mspeed;
}
```

```
}// move_LeftForward(int mspeed)

void move_RightBackward(int mspeed){ // robotControlState = 7
if(robotControlState!=7 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, 0);
analogWrite(Lpwm1, mspeed*0.4);
analogWrite(Rpwm2, 0);
analogWrite(Lpwm2, mspeed);
robotControlState=7;
last_mspeed=mspeed;
}
}// void move_RightBackward(int mspeed)

void move_LeftBackward(int mspeed){ // robotControlState = 8
if(robotControlState!=8 || last_mspeed!=mspeed){
SetMotors(1);
analogWrite(Rpwm1, 0);
analogWrite(Lpwm1, mspeed);
analogWrite(Rpwm2, 0);
analogWrite(Lpwm2, mspeed*0.4);
robotControlState=8;
last_mspeed=mspeed;
}
}// void move_LeftBackward(int mspeed)

void stopRobot(int delay_ms){
SetMotors(2);
analogWrite(Rpwm1, 0);
analogWrite(Lpwm1, 0);
analogWrite(Rpwm2, 0);
analogWrite(Lpwm2, 0);
delay(delay_ms);
}// void stopRobot(int delay_ms)

void SetPWM(const long pwm_num, byte pwm_channel){
if(pwm_channel==1){ // DRIVE MOTOR
analogWrite(Rpwm1, 0);
analogWrite(Rpwm2, 0);
analogWrite(Lpwm1, pwm_num);
analogWrite(Lpwm2, pwm_num);
```

```
pwmRvalue = pwm_num;
}
else if(pwm_channel==2){ // STEERING MOTOR
analogWrite(Lpwm1, 0);
analogWrite(Lpwm2, 0);
analogWrite(Rpwm1, pwm_num);
analogWrite(Rpwm2, pwm_num);
pwmLvalue = pwm_num;
}
}// void SetPWM (const long pwm_num, byte pwm_channel)

void SetMotors(int controlCase){
switch(controlCase){
case 1:
digitalWrite(MotorRight_R_EN, HIGH);
digitalWrite(MotorRight_L_EN, HIGH);
digitalWrite(MotorLeft_R_EN, HIGH);
digitalWrite(MotorLeft_L_EN, HIGH);
break;
case 2:
digitalWrite(MotorRight_R_EN, LOW);
digitalWrite(MotorRight_L_EN, LOW);
digitalWrite(MotorLeft_R_EN, LOW);
digitalWrite(MotorLeft_L_EN, LOW);
break;
}
}// void SetMotors(int controlCase)
```

CHAPTER 5: COMPONENT COST

CHAPTER 5: COMPONENT COST

5.1 COST OF COMPONENTS

SR.NO.	COMPONENT NAME	SPECIFICATION	QUANTITY	COST
1.	CHASSIS	ALUMINIUM	1	2000
2.	WHEELS	90.2 cm	4	800
2.	HIGH TORQUE D.C.MOTOR	12V,200RPM	4	2400
3.	ARDUINO MEGA 2560	--------	1	800
4.	BATTERY	12V, 2200 mAh	1	4500
5.	BTS - 7960B	--------	2	1400
6.	CONNECTERS	--------	AS PER USE	AS PER USE
7.	ESP 32 Camera	--------	1	700
Total:-			12600	

CHAPTER 6: ADVANTAGES, DISADVANTAGES, CONCLUSION AND FUTURESCOPE

CHAPTER 6: ADVANTAGES, DISADVANTAGES, CONCLUSION AND FUTURE SCOPE

6.1 What it can do ?

6.1.1 In Defense Sector:

- It can go through enemy terrains and search for blind-spots.
- Can carry out reconnaissance missions..
- Land Surveying .
- Payload delivery to remote locations.
- Live camera feed to the base station.

6.1.2 In Health Sector:

- It can serve food and medicines to patients having contagious diseases like COVID-19, Ebola, Flu, etc.
- The care givers (nurses & doctors) can operate it remotely from a safe distance.

6.1.3 CONCLUSION:

It becomes very difficult for military personnel to carry out reconnaissance missions on foot as it proposes a great risk to their lives. Using The Navigation Enabled Reconnaissance Device the military personnel can easily carry out reconnaissance and surveillance missions without risking their valuable lives.

After a lot of research and discussion we have finalized this title. Our vision was to help those guarding our nation's frontier. The Navigation Enabled reconnaissance Device will prove of great help to them.

6.1.4 FUTURE SCOPE:

In future the N.E.R.D. can be equipped with Light Detection and Ranging (LiDAR) Scanners to map territories.

By using efficient Machine learning Algorithms the N.E.R.D. can be equipped with face detection and track prediction.

The N.E.R.D. can also be equipped with the Navigation with Indian Constellation (NavIC) receivers developed by the Indian Space Research Organization for accurate navigation in the Indian subcontinent and surrounding countries. This NavIC will also help in securing the national borders effectively.

REFERANCES

- https://www.arduino.cc/
- https://www.wikipedia.org/
- https://www.isro.gov.in/
- https://www.drdo.gov.in/home