A

Technical Report

on

**INDUSTRIAL LOAD CARRIER**

*Submitted to CMR Institute of Technology in the partial fulfillment of the requirement of*

**Social Innovation Lab**

Of

**II B.Tech I- Semester**

in

**ECE DEPARTMENT**

Submitted by

**B. Vishwas Goud 20R01A04C5**

**B. Vaishnavi 20R01A04C7**

**B. Shalini sumathi 20R01A04D1**

**G. Praneeth kumar 20R01A04D9**

**P. Harsha 20R01A04G7**

**T. Mounika 21R05A0413**

***Under the esteemed guidance of***

*Mr.T.Nagaraju*

*Assistant Professor*



**CMR INSTITUTE OF TECHNOLOGY**

**(UGC-AUTONOMOUS)**

(Approved by AICTE, Permanently Affiliated to JNTU Hyderabad, Accredited by NBA, Accredited by NAAC with ‘A’ Grade)

Kandlakoya (V), Medchal Road, Hyderabad – 501 401

**2021-2022**

***Department of ECE***

**Certificate**

This is to certify that the technical report entitled “***INDUSTRIAL LOAD CARRIER***” is the bonafide work done and submitted by

**B. Vishwas Goud 20R01A04C5**

**B. Vaishnavi 20R01A04C7**

**B. Shalini sumathi 20R01A04D1**

**G. Praneeth kumar 20R01A04D9**

**P. Harsha 20R01A04G7**

**T. Mounika 21R05A0413**

towards the partial fulfillment of the requirement of Social Innovation (SIL) Laboratory of **II B. Tech I-Semester** in **ECE** is a record of bonafide work carried out by them during the period **Sep 2021 to Jan 2022.**

**Guide Co- Ordinator Head of Department**

**Mr. T. Nagaraju Mr. Veeresh Dr .K. NiranjanReddy**

**INDEX**

**Topics Page No**

**CHAPTER-I INTRODUCTION 1-2**

**CHAPTER -II Empathize 3**

**CHAPTER -III Define**  **4-5**

**CHAPTER -IV Ideate 6-7**

**CHAPTER -V Prototype 8-15**

**CHAPTER -VI Test 16**

**CHAPTER -VII References 17**

1. **INTRODUCTION**

* **WHAT IS SOCIAL INNOVATION?**

The term ‘social innovation ’once rarely heard is ,now often used to describe a whole variety of things that fall into general categories of being both new and good.It’s understandable that the phrase has become popular-we get excited and hopeful when it seems possible for real change to happen in the world.

Social innovation refers to the Design and implementation of new solutions that imply conceptual ,process ,product or organisational change which ultimately aim to improve the welfare and wellbeing of individual communities

Social innovation is not a new concept and should not be considered similar to other definitions, such as social entrepreneurship, creativity or invention, improvement or change. 'As with innovation in technology or business, social innovation is distinct from ‘improvement’ or ‘change’ and from ‘creativity’ and ‘invention’. These last two are both crucial to innovation but overlook the important stages of implementation and diffusion which make new ideas useful.

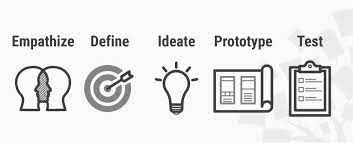
* **What is design thinking process?**

Design Thinking is a design methodology that provides a solution-based approach to solving problems. It’s extremely useful in tackling complex problems that are ill-defined or unknown, by understanding the human needs involved, by re-framing the problem in human-centric ways, by creating many ideas in brainstorming sessions, and by adopting a hands-on approach in prototyping and testing. Understanding these five stages of Design Thinking will empower anyone to apply the Design Thinking methods in order to solve complex problems that occur around us — in our companies, in our countries, and even on the scale of our planet.

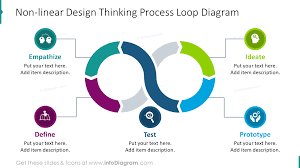
Design thinking originally came about as a way of teaching engineers how to approach problems creatively, like designers do. One of the first people to write about design thinking was John E. Arnold, professor of mechanical engineering at Stanford University.

**1**

**The five stages of design thinking:**



1. Empathize-The Design Thinking process starts with empathy. In order to create desirable products and services, you need to understand who your users are and what they need.
2. Define- In the second stage of the Design Thinking process, you’ll define the user problem that you want to solve.
3. Ideate.-The third stage in the Design Thinking process consists of ideation or generating ideas. ...
4. Prototype- In the fourth stage of the Design Thinking process, you’ll turn your ideas from stage three into prototypes.
5. Test -The fifth step in the Design Thinking process is dedicated to testing: putting your prototypes in front of real users and seeing how they get on.

****

**2**

1. **Empathize**

The first stage of the Design Thinking process is to gain an empathic understanding of the problem you are trying to solve. This involves consulting experts to find out more about the area of concern through observing, engaging and empathizing with people to understand their experiences and motivations, as well as immersing yourself in the physical environment so you can gain a deeper personal understanding of the issues

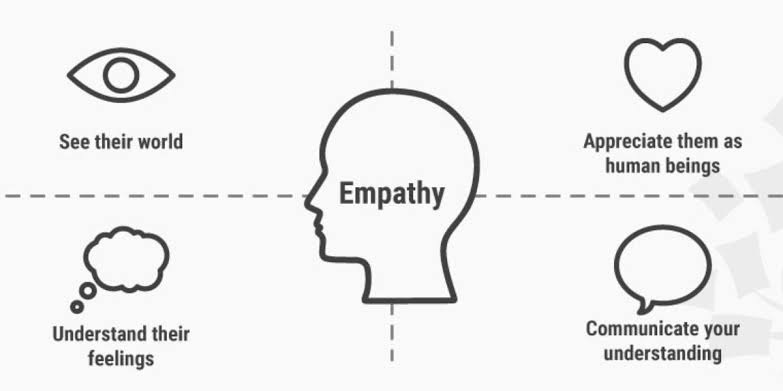
involved. Empathy is crucial to a human-centered design process such as Design Thinking, and empathy allows design thinkers to set aside their own assumptions about the world in order to gain insight into users and their needs.

We have collected information from various sources like conducting surveys among the people about their problems as they are facing right now and interviewing people, reading novels from various books ,collecting information from the internet.

As our team has conducted a survey among the people at the current problems they are facing we have got many problems to be listed .In those information we have found many valid problems as they are facing in the day to day life and the collected information have been segregated accordingly.

We have shortlisted few problems which are being affected by the most people in the society.

So, have chosen one of the problem that is the heavy loads carried by labourers in industries is the major problem from the shortlisted problems that many people are facing in industry



**3**

**3.Define**

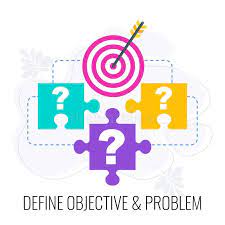
In this define stage, we have defined the problem statement accordingly to our problem. According to the scenario as we have collected information in the empathy stage we have defined the problem statement as “INDUSTRIAL LOAD CARRIERS”

**3.1. Problem Statement**

**We came to a conclusion of a problem statement of helping the labours in industries like minning industries etc., They are facing many problems in industries.They are unable to carry the heavy loads. In order to avoid such problems we came up with an idea called** “INDUSTRIAL LOAD CARRIERS”.

**3.2 Objective :-**

The main objective is to carry the loads with robot using firebird V Atmega 2560 microcontroller to control the robot intellectually and white line sensor used to detect the white line, obstacal avoider is used to detect the obstacal and automatically change it’s path



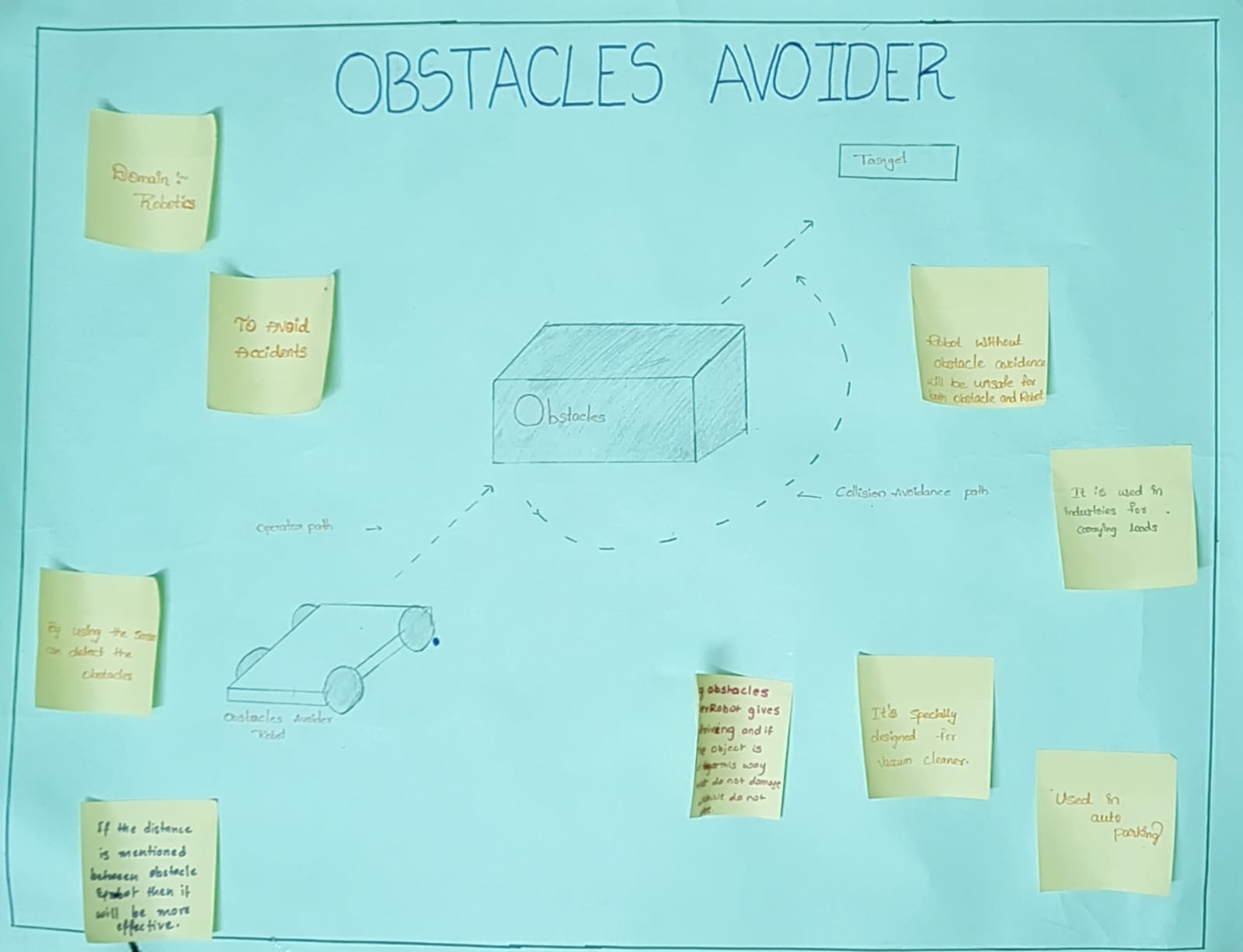
**4.Ideate**

In this design thinking process we have ideate as the next stage and we have come up with a solution according to the above problem statement as we have mentioned. After reviewing our problem statement, we got an idea of preparing a model. Firstly we drew our ideas, thoughts in the form of diagram on chart. Then this chart was passed to each and every team of the class. Through this chart was passed to teams by this we have got some more ideas and new inventions regarding the project. We have made a model of new inventions regarding the project. We have made a model of load carrier using firebird V.

The load carrier robot project is based on firebird V Atmega 2560 microcontroller. project is to sence the white line and avoid any obstacle or object in front of robat and automatically change it's path.

Line follower is a machine that can follow a path. The path can be visible like a black line on a white surface (vice versa).

Obstacal avoider is automatically changing the direction of robot whenever any obstacle comes on it's way.

**** 

**5.Prototype**

The next step is making a prototype , that is for making a prototype we require components like

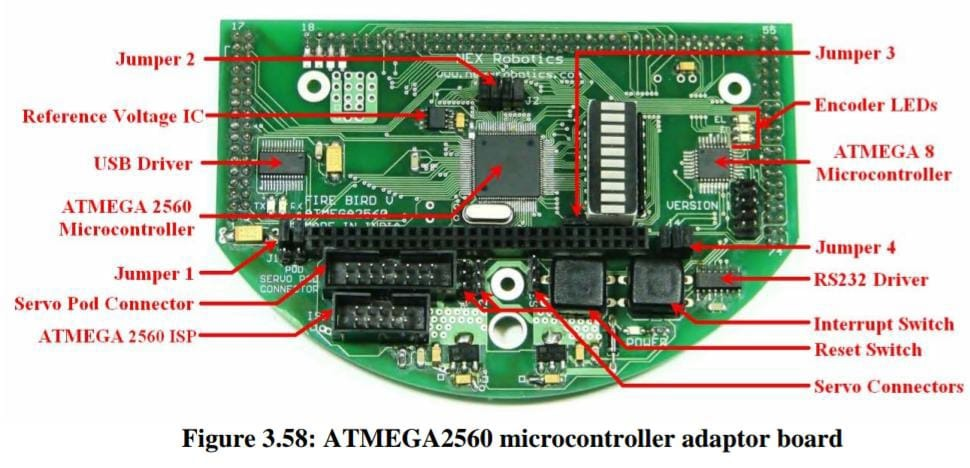
**COMPONENTS REQUIREMENTS :-**

* **Microcontroller:**
* Atmel ATMEGA2560 as Master microcontroller
* Atmel ATMEGA8 as Slave microcontroller
* **Sensors:**
* Three white line sensors
* Five Sharp IR range sensor
* Eight analog IR proximity sensors
* Two position encoders
* Battery voltage sensing
* **LCD:**
* 2 x 16 Characters of LCD
* Indicator LEDs
* Buzzer
* **Power:**
* 9.6V, 2100mAh Nickel Metal Hydride (NiMH) battery pack and external Auxiliary power from battery charger.
* **Software support**
* AVR studio ,Atmel studio

**TOTAL COST OF OUR PROJECT IS =Rs**

**Microcontroller:**

* An integrated circuit that contains a microprocessor along with memory and associated circuits and that controls some or all of the functions of an electronic device or system.
* ATMEGA2560 is a featur rich microcontroller with lots of available I/O ports. Many of the ports are available on the uC Expansion Socket**.**

****

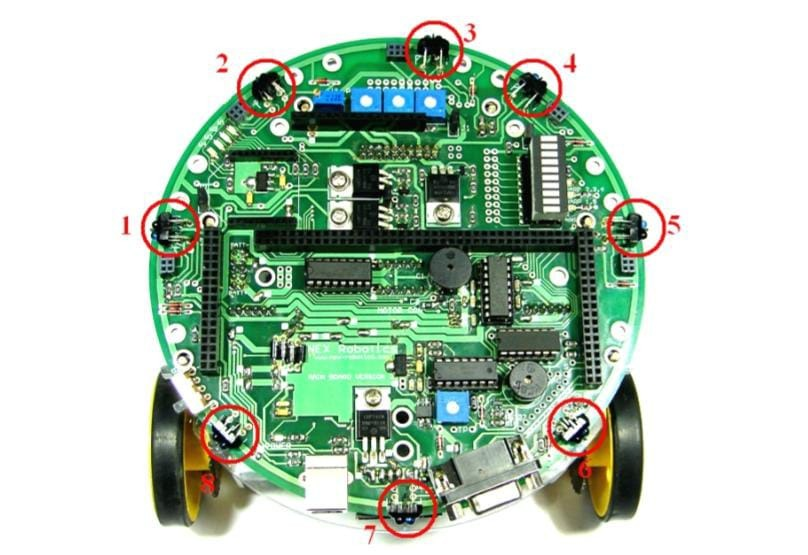
**White line sensor**

* White line sensors are used for detecting white line on the ground surface. White lines are used to give robot sense of localization.
* White line sensor consists of a highly directional photo transistor for line sensing and bright red LED for the illumination.
* Due to the directional nature of the photodiode it does not get affected with ambient light unless it is very bright**.**

****

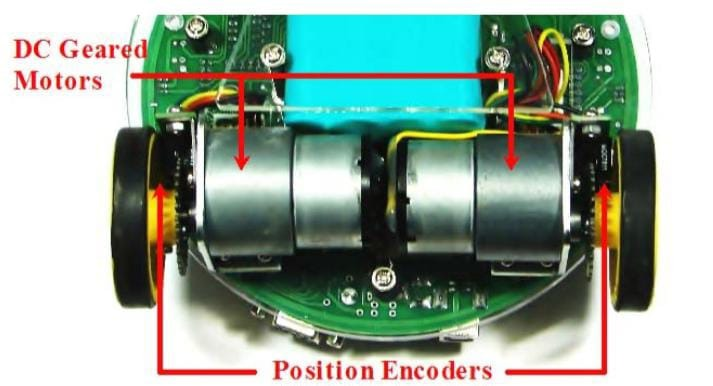
**INFRA RARED SENSORS**

* Infrared proximity sensors are used to detect proximity of any obstacles in the short range.
* IR proximity sensors have about 10cm sensing range. These sensors sense the presence of the obstacles in the blind spot region of the Sharp range sensors.
* Sensors are numbered as 1 to 8 from left to right in clockwise direction. In all the manuals this numbering convention will be used for addressing the particular IR sensor**.**

****

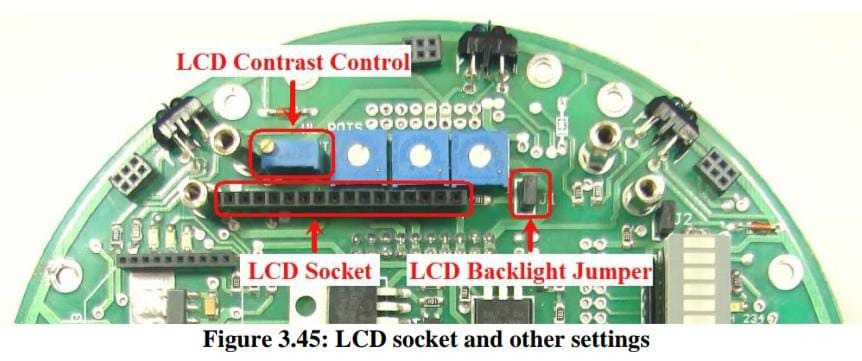
**POSITION ENCODERS**

* Position encoders give position / velocity feedback to the robot. It is used in closed loop to control robot’s position and velocity. Position encoder consists of slotted disc which rotates between optical encoder.
* It consists of IR LED and the photo transistor mounted in front of each other separated by a slot in black opaque casing with small slot shaped window facing each other**.**

****

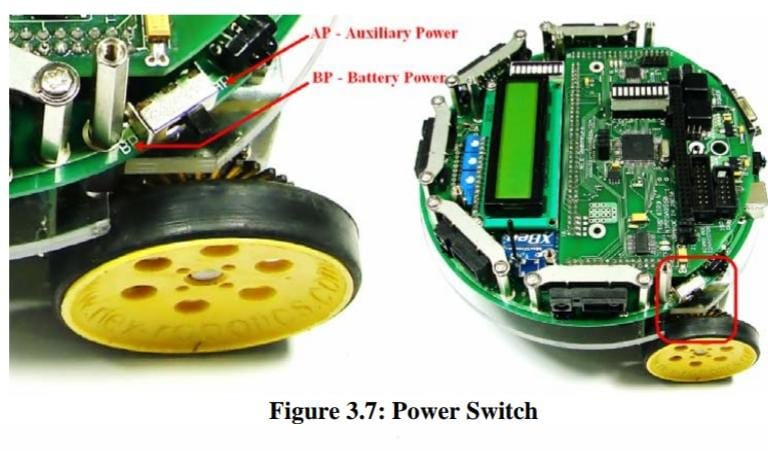
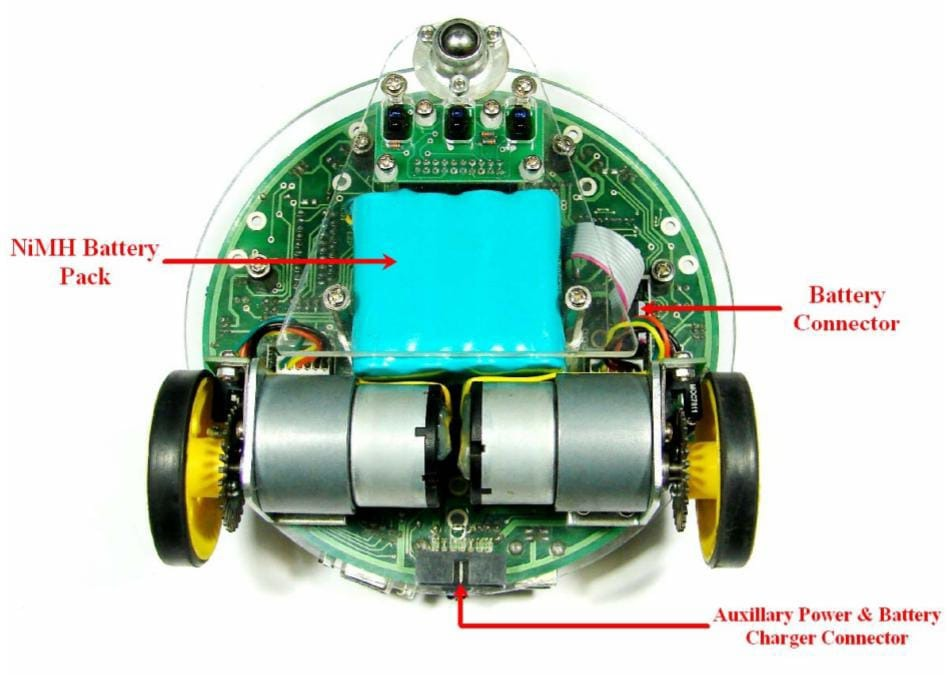
**LCD (LIQUID CRYSTAL DISPLAY)**

* To interface LCD with the microcontroller in default configuration requires 3 control signals and 8 data lines.
* This is known as 8 bit interfacing mode which requires total 11 I/O lines.
* The three control lines are referred to as EN, RS, and RW.
* The EN line is called "Enable“. This control line is used to tell the LCD that microcontroller has sent data to it or microcontroller is ready to receive data from LCD
* The RS line is the "Register Select“. When RS is low (0), the data is treated as a command (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is treated as text data which should be displayed on the screen.
* The RW line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively reading from the LCD.



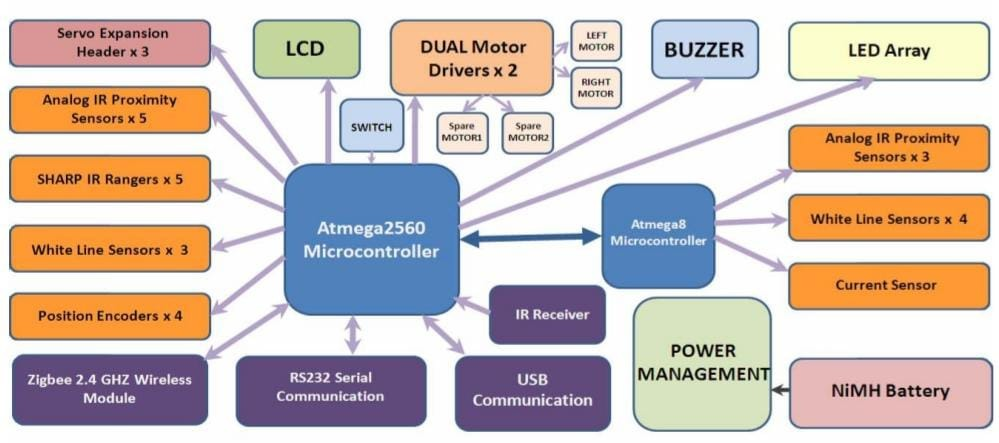
**POWER**

* Robot has onboard rechargeable 9.6V, 2.1Ah Nickel Metal Hydride battery which can power the robot for approximately 2 hours.
* In case experiments are to be performed for an extended period robot can also be powered by external, auxiliary power supply.
* On the two sides of the power switch “BP” and “AP” is marked. “BP” stands for Battery Power while “AP” stands for Auxiliary power.
* When robot is powered by battery, it will turn on when power switch is moved towards “BP” and will get turned of if switch is moved towards “AP”.

  
**SOFTWARE SUPPORT**

**Atmel Studio:**

* The Atmel Studio 7 IDP gives you a seamless and easy-to-use environment to write, build, and debug your applications written in C/C++ or assembly code. It also connects seamlessly to the debuggers, programmers, and development kits that support AVR and SAM devices.



**Prototype model:**

****

**Source code:**

**#define F\_CPU 14745600**

**#include <avr/io.h>**

**#include <avr/interrupt.h>**

**#include <util/delay.h>**

**#include <math.h> //included to support power function**

**#include "lcd.h"**

**void port\_init();**

**void timer5\_init();**

**void velocity(unsigned char, unsigned char);**

**void motors\_delay();**

**unsigned char ADC\_Conversion(unsigned char);**

**unsigned char ADC\_Value;**

**unsigned char flag1 = 0;**

**unsigned char flag2 = 0;**

**unsigned char Left\_white\_line = 0;**

**unsigned char Center\_white\_line = 0;**

**unsigned char Right\_white\_line = 0;**

**unsigned char Front\_Sharp\_Sensor=0;**

**unsigned char Front\_IR\_Sensor=0;**

**//Function to configure LCD port**

**void lcd\_port\_config (void)**

**{**

**DDRC = DDRC | 0xF7; //all the LCD pin's direction set as output**

**PORTC = PORTC & 0x80; // all the LCD pins are set to logic 0 except PORTC 7**

**}**

**//ADC pin configuration**

**void adc\_pin\_config (void)**

**{**

**DDRF = 0x00;**

**PORTF = 0x00;**

**DDRK = 0x00;**

**PORTK = 0x00;**

**}**

**//Function to configure ports to enable robot's motion**

**void motion\_pin\_config (void)**

**{**

**DDRA = DDRA | 0x0F;**

**PORTA = PORTA & 0xF0;**

**DDRL = DDRL | 0x18; //Setting PL3 and PL4 pins as output for PWM generation**

**PORTL = PORTL | 0x18; //PL3 and PL4 pins are for velocity control using PWM.**

**}**

**//Function to initialize Buzzer**

**void buzzer\_pin\_config (void)**

**{**

**DDRC = DDRC | 0x08; //Setting PORTC 3 as outpt**

**PORTC = PORTC & 0xF7; //Setting PORTC 3 logic low to turnoff buzzer**

**}**

**//Function to Initialize PORTS**

**void port\_init()**

**{**

**lcd\_port\_config();**

**adc\_pin\_config();**

**motion\_pin\_config();**

**buzzer\_pin\_config();**

**}**

**// Timer 5 initialized in PWM mode for velocity control**

**// Prescale:256**

**// PWM 8bit fast, TOP=0x00FF**

**// Timer Frequency:225.000Hz**

**void timer5\_init()**

**{**

**TCCR5B = 0x00; //Stop**

**TCNT5H = 0xFF; //Counter higher 8-bit value to which OCR5xH value is compared with**

**TCNT5L = 0x01; //Counter lower 8-bit value to which OCR5xH value is compared with**

**OCR5AH = 0x00; //Output compare register high value for Left Motor**

**OCR5AL = 0xFF; //Output compare register low value for Left Motor**

**OCR5BH = 0x00; //Output compare register high value for Right Motor**

**OCR5BL = 0xFF; //Output compare register low value for Right Motor**

**OCR5CH = 0x00; //Output compare register high value for Motor C1**

**OCR5CL = 0xFF; //Output compare register low value for Motor C1**

**TCCR5A = 0xA9; /\*{COM5A1=1, COM5A0=0; COM5B1=1, COM5B0=0; COM5C1=1 COM5C0=0}**

**For Overriding normal port functionality to OCRnA outputs.**

**{WGM51=0, WGM50=1} Along With WGM52 in TCCR5B for Selecting FAST PWM 8-bit Mode\*/**

**TCCR5B = 0x0B; //WGM12=1; CS12=0, CS11=1, CS10=1 (Prescaler=64)**

**}**

**void buzzer\_on (void)**

**{**

**unsigned char port\_restore = 0;**

**port\_restore = PINC;**

**port\_restore = port\_restore | 0x08;**

**PORTC = port\_restore;**

**}**

**void buzzer\_off (void)**

**{**

**unsigned char port\_restore = 0;**

**port\_restore = PINC;**

**port\_restore = port\_restore & 0xF7;**

**PORTC = port\_restore;**

**}**

**void adc\_init()**

**{**

**ADCSRA = 0x00;**

**ADCSRB = 0x00; //MUX5 = 0**

**ADMUX = 0x20; //Vref=5V external --- ADLAR=1 --- MUX4:0 = 0000**

**ACSR = 0x80;**

**ADCSRA = 0x86; //ADEN=1 --- ADIE=1 --- ADPS2:0 = 1 1 0**

**}**

**//Function For ADC Conversion**

**unsigned char ADC\_Conversion(unsigned char Ch)**

**{**

**unsigned char a;**

**if(Ch>7)**

**{**

**ADCSRB = 0x08;**

**}**

**Ch = Ch & 0x07;**

**ADMUX= 0x20| Ch;**

**ADCSRA = ADCSRA | 0x40; //Set start conversion bit**

**while((ADCSRA&0x10)==0); //Wait for conversion to complete**

**a=ADCH;**

**ADCSRA = ADCSRA|0x10; //clear ADIF (ADC Interrupt Flag) by writing 1 to it**

**ADCSRB = 0x00;**

**return a;**

**}**

**//Function To Print Sesor Values At Desired Row And Coloumn Location on LCD**

**void print\_sensor(char row, char coloumn,unsigned char channel)**

**{**

**ADC\_Value = ADC\_Conversion(channel);**

**lcd\_print(row, coloumn, ADC\_Value, 3);**

**}**

**//Function for velocity control**

**void velocity (unsigned char left\_motor, unsigned char right\_motor)**

**{**

**OCR5AL = (unsigned char)left\_motor;**

**OCR5BL = (unsigned char)right\_motor;**

**}**

**//Function used for setting motor's direction**

**void motion\_set (unsigned char Direction)**

**{**

**unsigned char PortARestore = 0;**

**Direction &= 0x0F; // removing upper nibbel for the protection**

**PortARestore = PORTA; // reading the PORTA original status**

**PortARestore &= 0xF0; // making lower direction nibbel to 0**

**PortARestore |= Direction; // adding lower nibbel for forward command and restoring the PORTA status**

**PORTA = PortARestore; // executing the command**

**}**

**void forward (void)**

**{**

**motion\_set (0x06);**

**}**

**void right(void)**

**{**

**motion\_set (0x0A);**

**}**

**void stop (void)**

**{**

**motion\_set (0x00);**

**}**

**void init\_devices (void)**

**{**

**cli(); //Clears the global interrupts**

**port\_init();**

**adc\_init();**

**timer5\_init();**

**sei(); //Enables the global interrupts**

**}**

**//Main Function**

**int main()**

**{**

**init\_devices();**

**lcd\_set\_4bit();**

**lcd\_init();**

**while(1)**

**{**

**//Left\_white\_line = ADC\_Conversion(3); //Getting data of Left WL Sensor**

**//Center\_white\_line = ADC\_Conversion(2); //Getting data of Center WL Sensor**

**//Right\_white\_line = ADC\_Conversion(1); //Getting data of Right WL Sensor**

**Front\_Sharp\_Sensor = ADC\_Conversion(11);**

**//Front\_IR\_Sensor = ADC\_Conversion(6);**

**//flag1=0;**

**//flag2=0;**

**//print\_sensor(1,1,3); //Prints value of White Line Sensor1**

**//print\_sensor(1,5,2); //Prints Value of White Line Sensor2**

**//print\_sensor(1,9,1); //Prints Value of White Line Sensor3**

**//print\_sensor(2,4,11); //Prints Value of Front Sharp Sensor**

**//print\_sensor(2,8,6); //Prints Value of Front IR Sensor**

**//lcd\_string("IIT Bombay e-yantra ERTS");**

**if(Front\_Sharp\_Sensor>0x52)**

**{**

**//flag2=1;**

**//stop();**

**//buzzer\_on();**

**//\_delay\_ms(200);**

**//buzzer\_off();**

**velocity(180,180);**

**right();**

**\_delay\_ms(850);**

**}**

**if((Center\_white\_line<0x28) && (flag2==0))**

**{**

**flag1=1;**

**buzzer\_off();**

**forward();**

**velocity(150,150);**

**}**

**if((Left\_white\_line>0x28) && (flag1==0) && (flag2==0))**

**{**

**flag1=1;**

**buzzer\_off();**

**forward();**

**velocity(150,50);**

**}**

**if((Right\_white\_line>0x28) && (flag1==0) && (flag2==0))**

**{**

**flag1=1;**

**buzzer\_off();**

**forward();**

**velocity(50,150);**

**}**

**if((Center\_white\_line>0x28) && (Left\_white\_line>0x28) && (Right\_white\_line>0x28) && (flag2==0))**

**{**

**buzzer\_off();**

**forward();**

**velocity(0,0);**

**}**

**forward();**

**velocity(200,200);**

**}**

**}**

**6.Test**

We found code for this project through references and build in atmel studio .so when we run this program .we found some errors in code like there are some undefined variables .so, we have gone through these errors and have been trying to build the solution .Tested the prototype by dumping the program into firebird robot. There are no errors in program and we changed the code to our convience. Like LED interfacing, motor control simple etc., We have seen some tutorial videos about how to dump the program into robot and successfully dumped the program. The program contains obstacles avoiding and white line following program. Then we have tried to change the program to stop when it detects white line. For that we replaced forward with stop but it doesn’t worked and then we decreased the velocity to zero then it worked after doing many modifications in code and dumped in firebird v .At last robot is working

**REFERENCES:-**

* ***<https://youtu.be/Rr6Rzrwfrgw>***
* ***<http://www.nex-robotics.com/products/fire-bird-v-robots/fire-bird-v-atmega2560-robotic-research-platform.html>***
* ***<https://youtu.be/aSVe2Fsw7n0>***
* ***<https://github.com/eyantra/Experiment/tree/master/Experiment>***
* ***<http://www.nex-robotics.com/products/fire-bird-v-robots/fire-bird-v-atmega2560-robotic-research-platform.html>***