Faculty of Information Technology

IS 1900-Business project

Infectious caring 'bot

Final Report

Group Number: 21(IT&M)

Group Members: **205009E- Chaami.D.H (Group leader)**

205012G-Dahanayake.H.C

205088R Ruchirani B.M.A.D.

205104P Shobisha. S.M

Supervisor Name : Mr.BH Sudantha,
Senior Lecturer,
Dept. Of Information Technology

Co-Supervisor Name : Ms.Kulani Mahadewa,
Senior Lecturer,
Dept. Of Information Technology

Date of Submission : 04/06/2022

Table of Contents

Introduction

There are no clear empirical machines that have been designed to deal with the patients who have been infected with infectious diseases such as (chickenpox, measles, influenza etc.) and treated at their homes. Moreover, the other homies also came to the risk of getting infected with the virus when they came in contact with the patient.

Furthermore, in a pandemic situation like covid, this kind of caring machine is so beneficial to the patients who are residing at their homes.

The specialty of this machine is the ability of the patient to work without colliding with external parties and minimizes the risk of outsiders getting the disease.

By and by a machine that can assist a severely ill patient with a lot of functions such as voice controlling, auto surround sanitizing is so beneficial to the patient and the other homies.

Literature Survey.

Since we use this project as an assistant to a patient the machine was included a lot of functionalities such as voice control, automated floor sanitizing, Food delivering, alarm systems etc.

So Various companies have implemented a massive number of systems that contain those functionalities with various intentions.

As one of those firstly we can indicate the automated vacuum cleaner robot which includes automated moving and obstacles averting technology.

Robotic vacuum cleaners don't use cameras to see the world. Instead, they use various types of sensors to detect and measure the worlds around them and their progress through it. They mainly use obstacles avoiding sensors to avoid collisions while it moves automatically.

So, we've chosen this technology for our infections caring robot. Also, we've used the mopping concept to the automated sanitizing process (To spray sanitizer).

Secondly, we have used the concept of Waiter robot for the sustenance delivery to the patient. Because we expect to minimize the interaction between the patient and a third party.

ROBOT waitress is an innovative functional robot that can act as an interactive assistant for any service sector. Robot waitresses can carry out reception duties, deliver food and drinks to the customers.

So, we have used this technology with slight modifications which added voice control to the machine and the user can move the machine to the paths he wants. also, we've added a plate on the side of the machine, and anyone can keep anything on that plate and deliver that to the patient safely.

Aims and objectives

Aim:

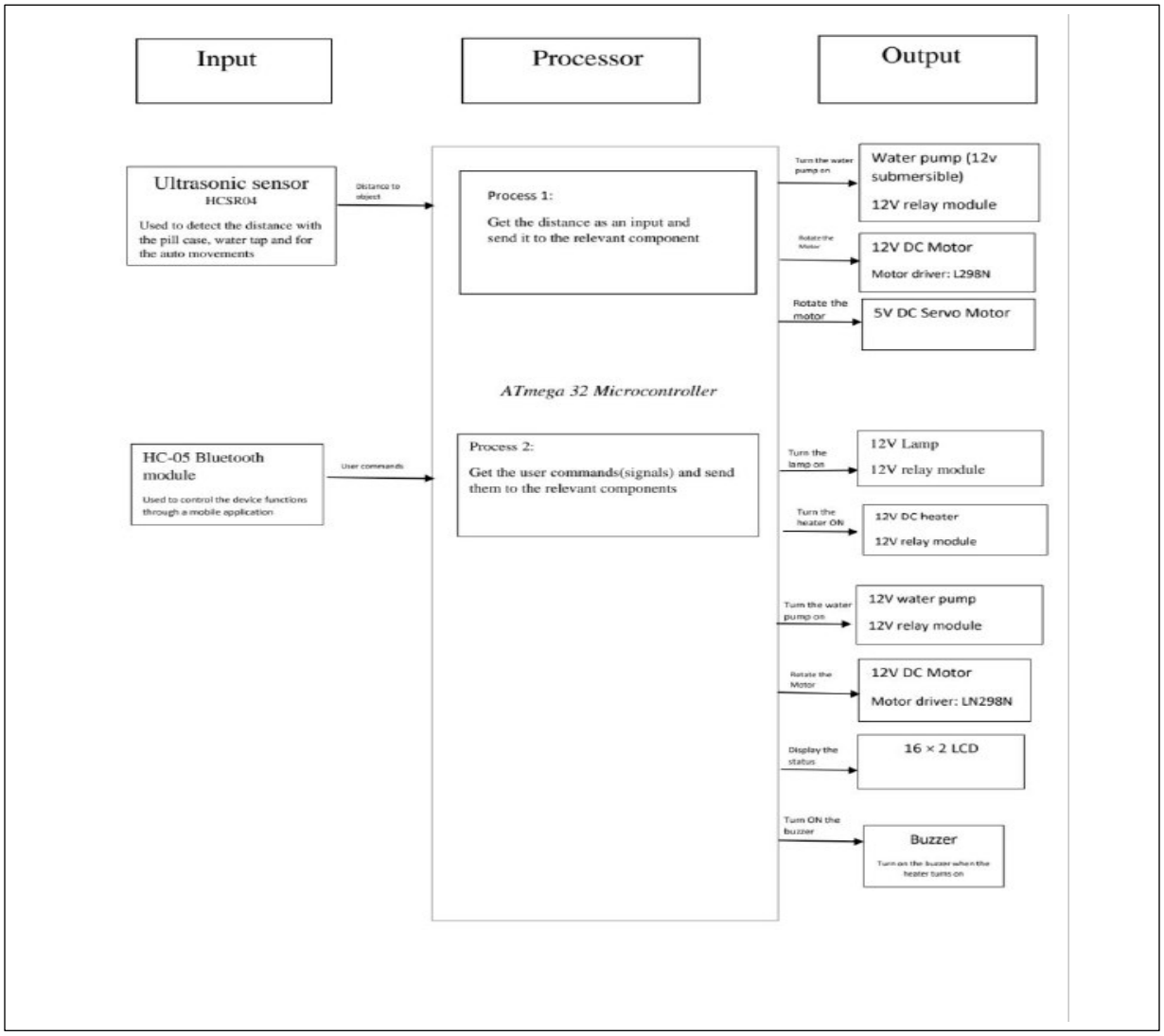
To interweave a machine that can help virus infected patients and diminish the interaction between the other parties and the patient.

Objectives:

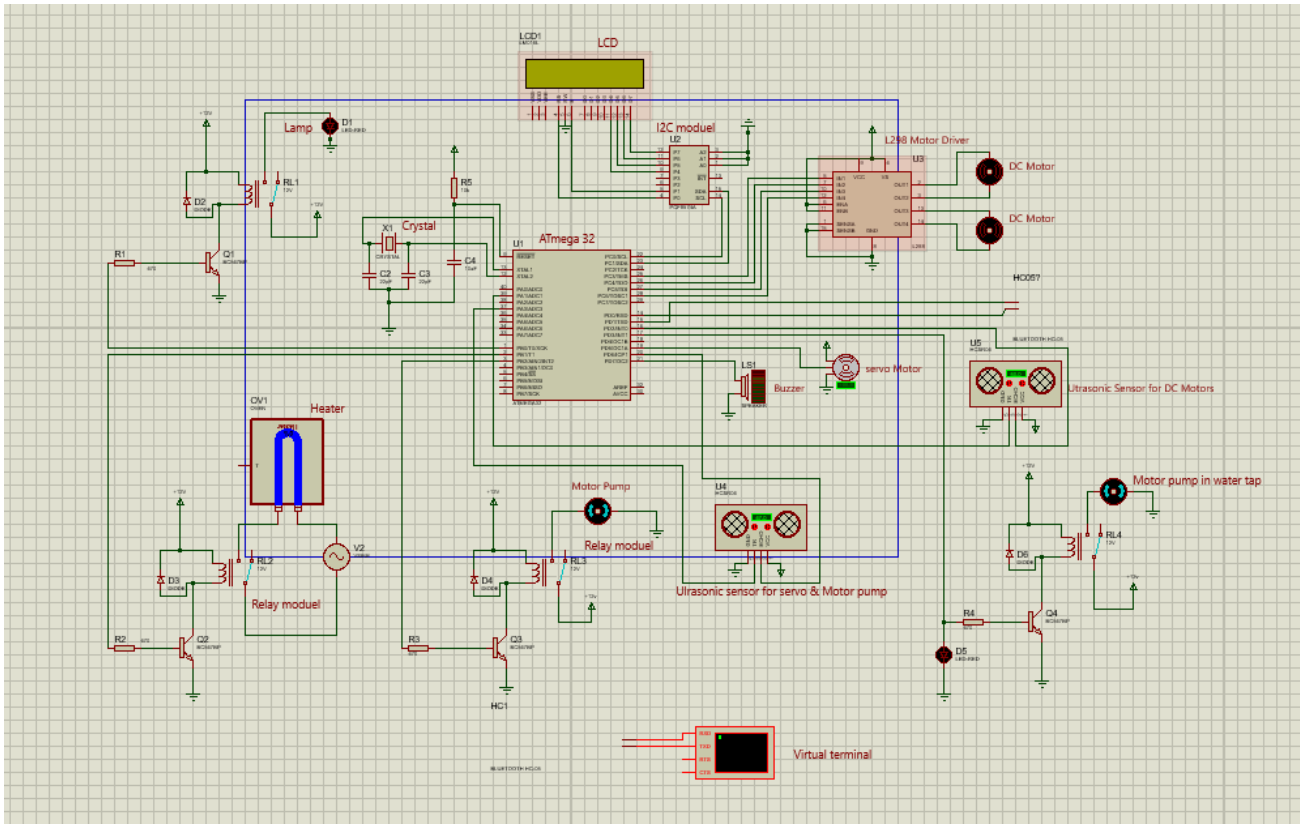
1) Reminds to hydrate in specific time slots to drink 2 liters of water per day.
2) Help patients to take medication properly according to a time frame.
3) Cleans the surrounding of the patient.
4) Diminish the interaction between other parties and the patient.

Analysis and Design

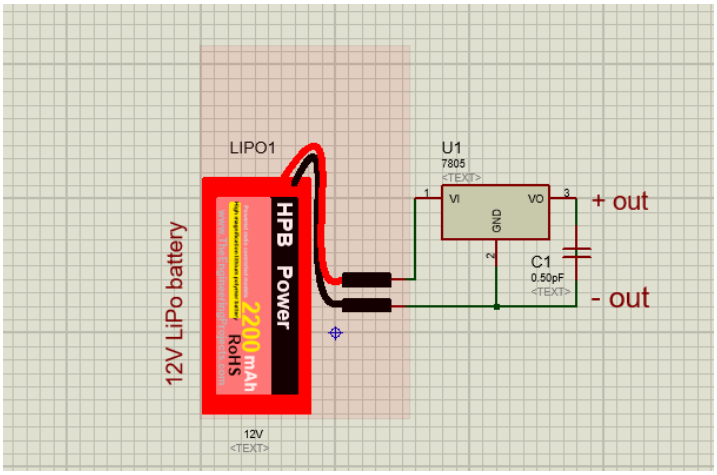
Overall Block



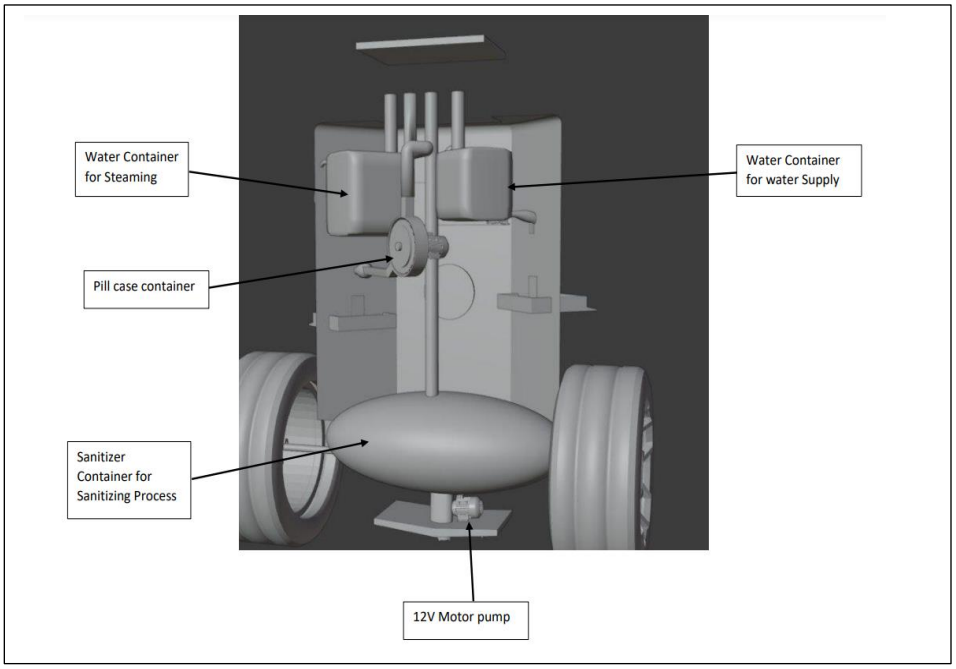
Schematic of the product

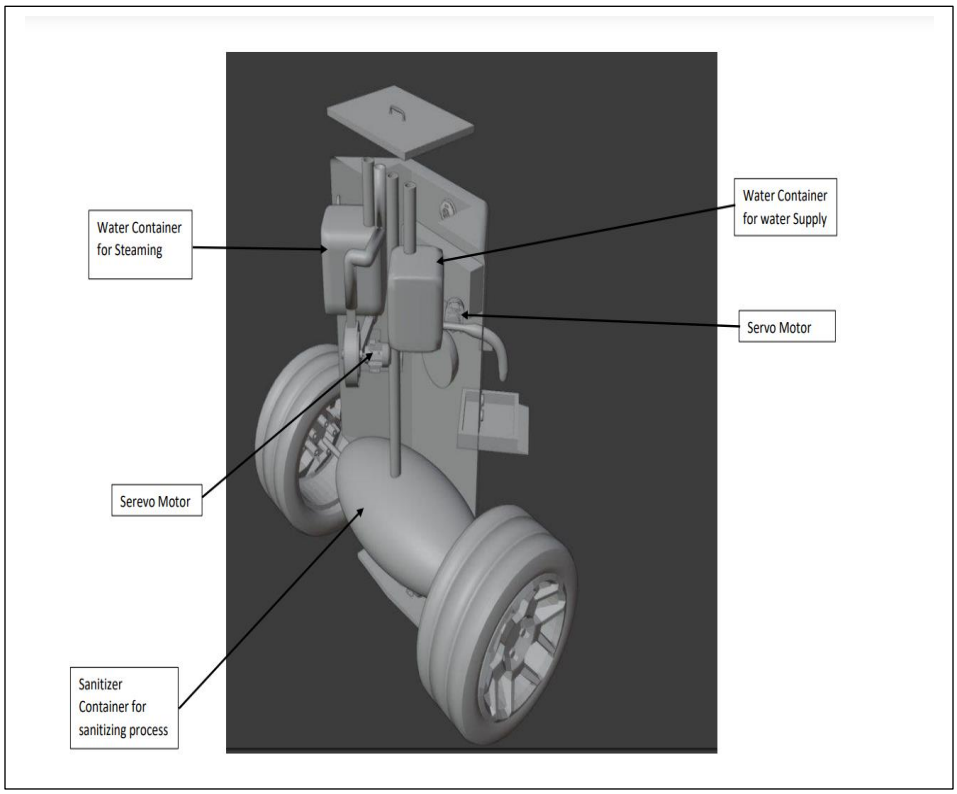


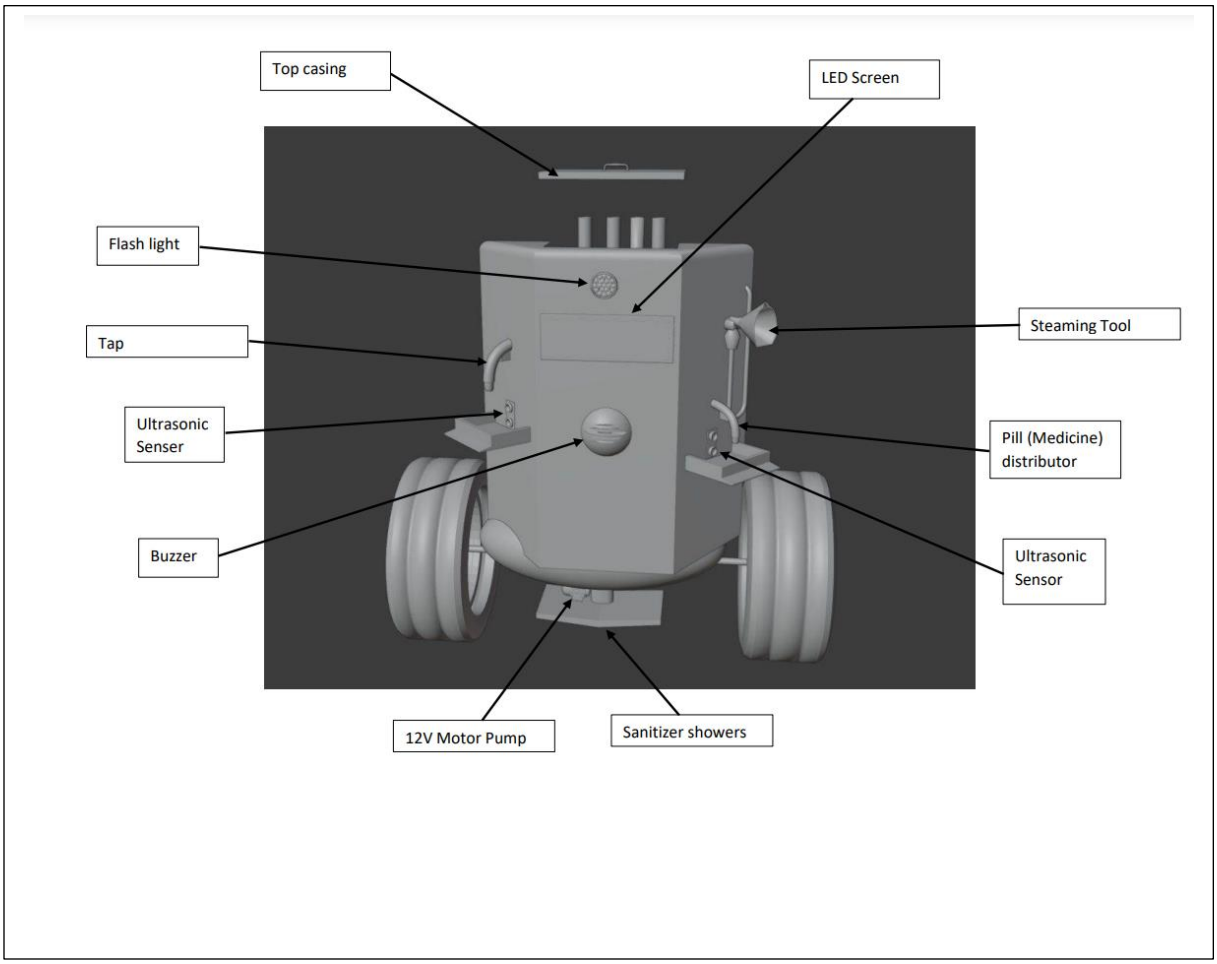
Power supply



3D –view of the product






Top casing
LED Screen
Flash light
Steaming Tool
Tap
Ultrasonic Senser
Pill (Medicine) distributor
Buzzer
Ultrasonic Sensor
12V Motor Pump
Sanitizer showers

Testing and implementation

As for the proposed action plan, we first did a literature survey on similar projects to get an idea about necessary equipment. Then, we studied about each and every component, compared their prices.

Then we studied how to simulate the simple basic circuit using proteus 8.0**.** Then we learnt how to program microcontrollers using Atmel studio 7.0. Then we developed a circuit for different components. And we learnt how to design PCB for our circuit. Finally, we learnt how to connect power supply.

We have used proteus 8.0 to design our circuit diagram.

We have used C language to build programming codes in above implementation.

We have designed PCB for our circuit diagram.

Finally, we have used proteus 8.0 to simulate our schematic diagram.

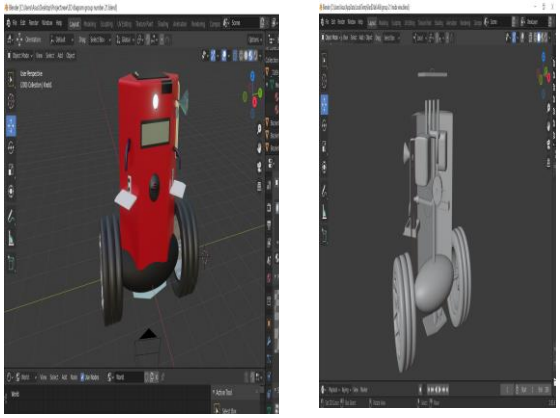


Figure 1

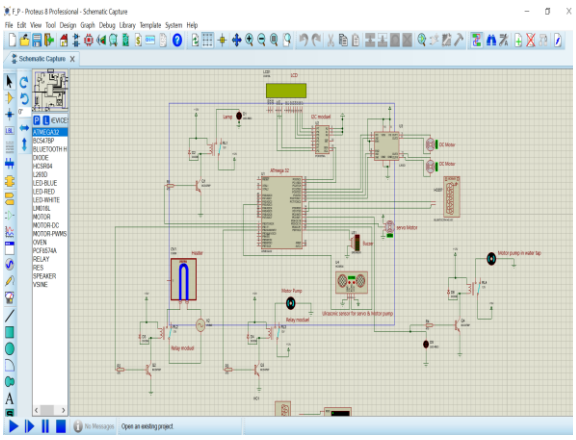


Figure 2

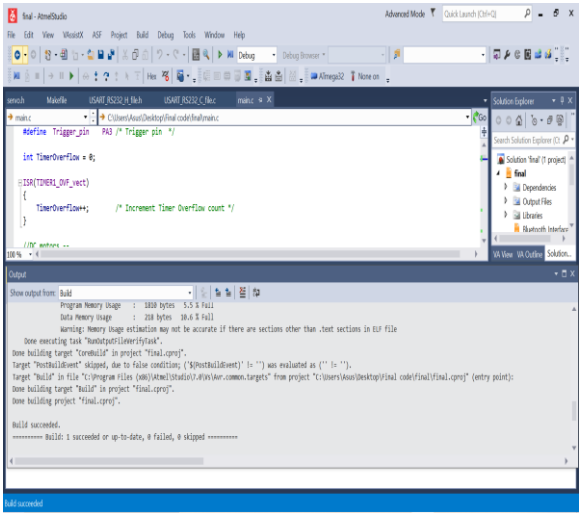


Figure 3

X

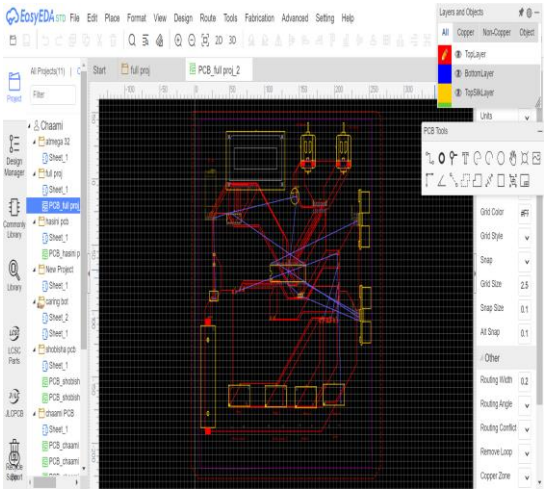


Figure 4

Cost estimation

Name	Unit price(RS)	Quantity	Total(Rs)
Atmel Atmega 32 microcontroller	600	2	1200
Ultrasonic sensor	200	5	1000
I2C module	180	1	180
LiPo battery	7200	1	7200
Servo motor	1690	3	5070
Motor pump submersible 200LH	2780	2	2780
LCD screen	450	1	450
IC 7805	35	1	35
Plastic wheel	76	4	760
DC motor	80	2	160
other	---	---	1000
Total amount	---	---	18445

References

[1]www.sematicweb.org/sources.html

[2]www. easyeda.com/sources.html

[3]www.electronicwings.com/sources.html

[4]www.github.com/sources.html

Individual contributions

Member 1: Chaami D. H (205009E)

Responsible parts

1. Components: - Servo Motor
   Motor pump (Through relay module)
   Ultrasonic sensor (Only for the servo motor and the motor pump)

2. Integration of the final code

3.PCB design of the full project

4.Design of the power supply circuit

Technique and specification

Servo motor

This motor is used in the pill case of our machine. There is a wheel which halved with holes which can hold a pill. Once it is rotated the pill can pan out from the open which is allocated to the pill case. So, whenever a user takes his hand toward the ultrasonic sensor (less than 30m) the servo connected to that wheel is rotated to a 0th-degree position (note here I used 90 the degree as the normal position of the servo motor) and turns back to its normal position again.

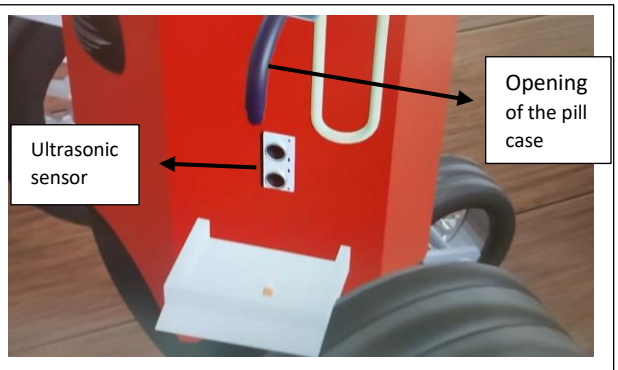


Figure 1

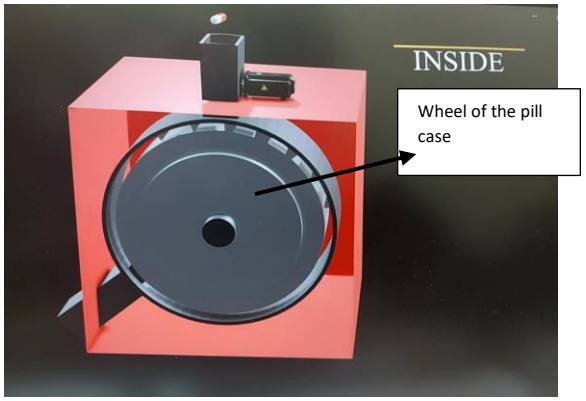


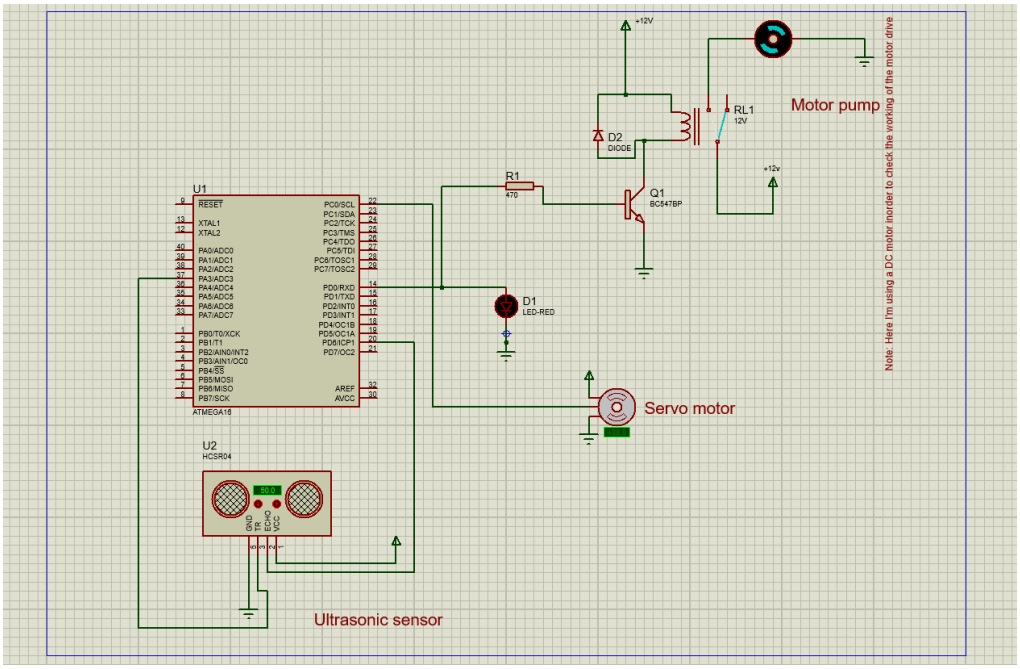
Figure 2

Motor pump

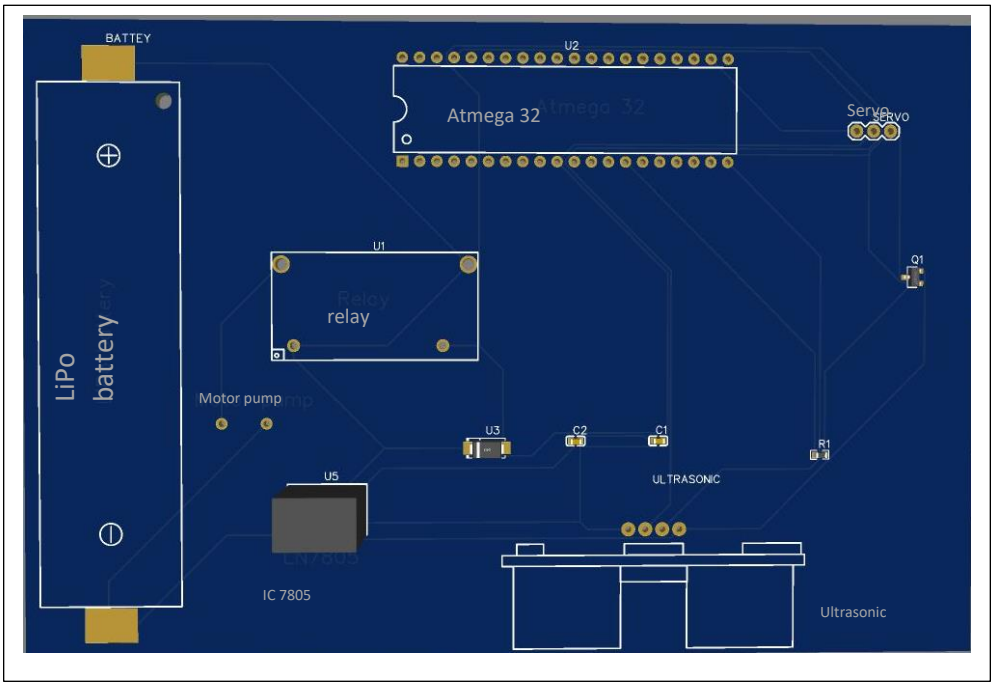
There is a motor pump submersible which is used to push water from the water tank through the water tap. It is connected through a relay module and whenever the user takes his hand toward the ultrasonic less than 45 meters the relay module is switched on and the water comes out through the water tap.

Figure 3

Circuit (Schematic diagram)



PCB design



Code

Servo.h

```
#ifndef _servo_H_
#define _servo_H_

servo_init()
{

	TCCR1A|=(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11);	//NON Inverted PWM
	TCCR1B|=(1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10); //PRESCALER=64 MODE 14(FAST
PWM)

	ICR1=4999;  //fPWM=50Hz (Period = 20ms Standard).

	DDRD|=(1<<PD4)|(1<<PD5);   //PWM Pins as Out
}

float set_degree(int d)
{
	OCR1A =160+((22*d)/9);
}



#endif /* SERVO_H_ */
```

Main.c

```
#ifndef F_CPU
#define F_CPU 8000000UL
#endif
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <string.h>
#include <stdlib.h>



#define  Trigger_pin PA3    /* Trigger pin  */

int TimerOverflow = 0;

ISR(TIMER1_OVF_vect)
{
       TimerOverflow++;            /* Increment Timer Overflow count */
}

int main(void)
{

       char string[10];
       long count;
       double distance;

       DDRA = 0b00001000;
       /* Make trigger pin as output */
       DDRD = 0b00000011;
       PORTD = 0x00;

        DDRC = 0x01; //Makes RC0 output pin
        PORTC = 0x00;

       sei();                              /* Enable global interrupt */
       TIMSK = (1 << TOIE1);       /* Enable Timer1 overflow interrupts */
       TCCR1A = 0;                         /* Set all bit to zero Normal operation */

       while(1)
       {

              PORTA |= (1 << Trigger_pin);/* Give 10us trigger pulse on trig. pin to HC-SR04
*/
              _delay_us(10);
              PORTA &= (~(1 << Trigger_pin));

              TCNT1 = 0;                  /* Clear Timer counter */
              TCCR1B = 0x41;              /* Setting for capture rising edge, No pre-scaler*/
              TIFR = 1<<ICF1;             /* Clear ICP flag (Input Capture flag) */
              TIFR = 1<<TOV1;             /* Clear Timer Overflow flag */

              /*Calculate width of Echo by Input Capture (ICP) on PortD PD6 */

              while ((TIFR & (1 << ICF1)) == 0); /* Wait for rising edge */
              TCNT1 = 0;                  /* Clear Timer counter */
              TCCR1B = 0x01;              /* Setting for capture falling edge, No pre-scaler
*/
              TIFR = 1<<ICF1;             /* Clear ICP flag (Input Capture flag) */
              TIFR = 1<<TOV1;             /* Clear Timer Overflow flag */
              TimerOverflow = 0;   /* Clear Timer overflow count */
```

```
while ((TIFR & (1 << ICF1)) == 0); /* Wait for falling edge */
                count = ICR1 + (65535 * TimerOverflow);  /* Take value of capture register */
                /* 8MHz Timer freq, sound speed =343 m/s, calculation mentioned in doc. */
                distance = (double)count / 466.47;

                dtostrf(distance, 2, 2, string);/* Convert distance into string */
                strcat(string, " cm   ");
                _delay_ms(200);



                if (distance > 35 && distance < 40)
                {
                        PORTD = 0b00000001; //Make the relay pin high
                }
                else if (distance < 35 )
                {
                        PORTD = 0b00000000;
                        //Rotate Motor to 0 degree
                        PORTC = 0x01;
                        _delay_us(1000);
                        PORTC = 0x00;

                        //Rotate Motor to 90 degree
                        PORTC = 0x01;
                        _delay_us(1500);
                        PORTC = 0x00;

                }
                else
                {
                        PORTD = 0b00000000;
                        //Rotate Motor to 0 degree
                        PORTC = 0x01;
                        _delay_us(1000);
                        PORTC = 0x00;
                }
        }
}
```

Member 2: Dahanayake H.C (205012G)

Responsible parts

- HC05 Bluetooth Module
- Buzzer
- Complete 3D Animation Design

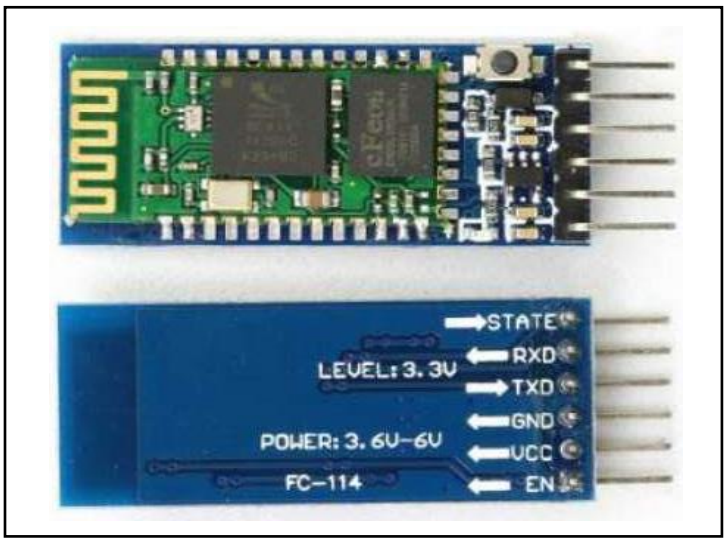
Techniques and Specification

HC05 Bluetooth Module:

Basically, the **Hco5 Bluetooth Module** in our project will operated the bot in general by giving some commands, it'll function according to those commands.

In order to move and also to do some kind of activities we are using the HC05 Bluetooth module. Those functionalities will happen according to the following commands

- When we give command =1 or 2 it'll Turn ON and Turn OFF the LED.
- When we give command=3 or 4 it'll Turn ON and Turn OFF the HEATER as well as the BUZZER.
- When we give command=5 or 6 it'll Turn ON and Turn OFF the MOTOR PUMP.
- When we give Commands like 'W' or 'S' or 'A' or 'd' then will Rotate the DC MOTORS.



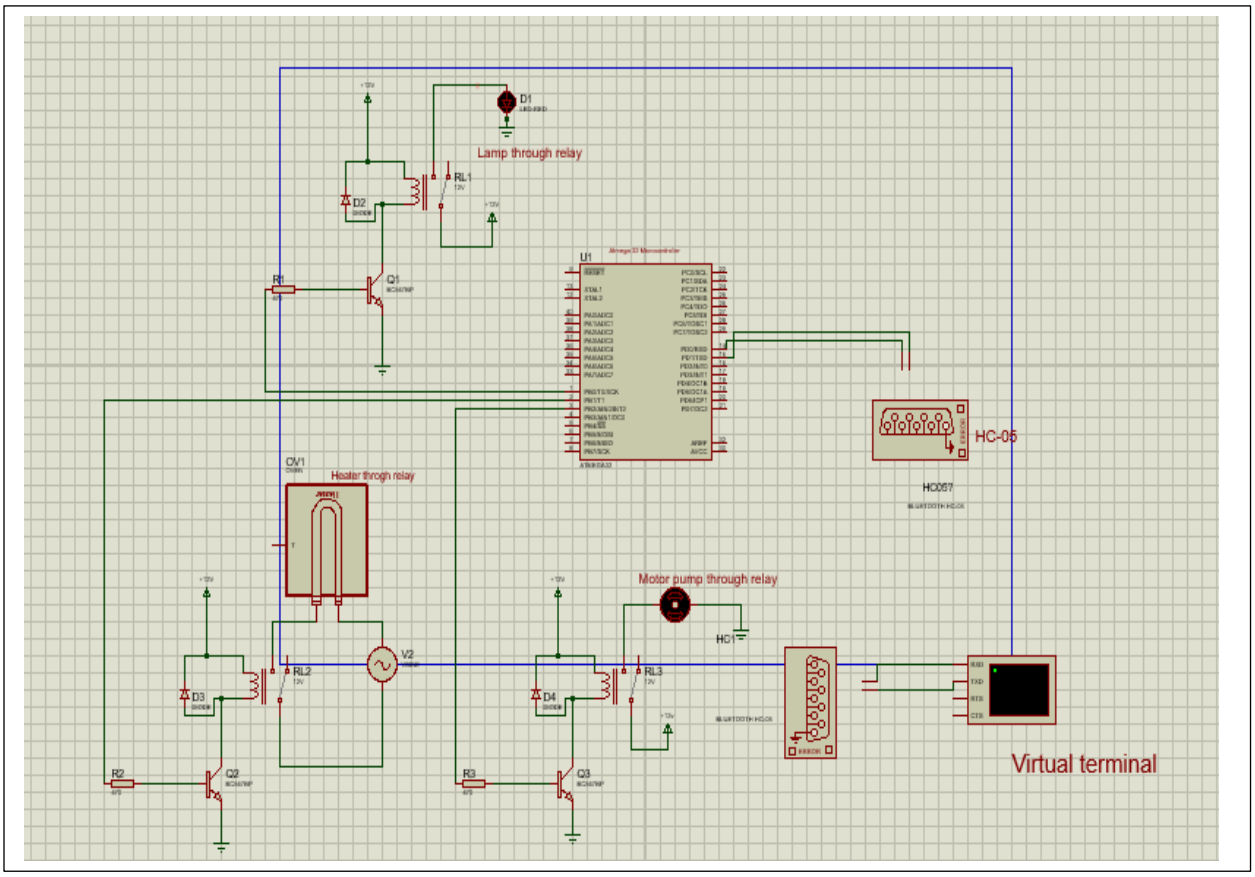
HC05 Bluetooth Module

Buzzer:

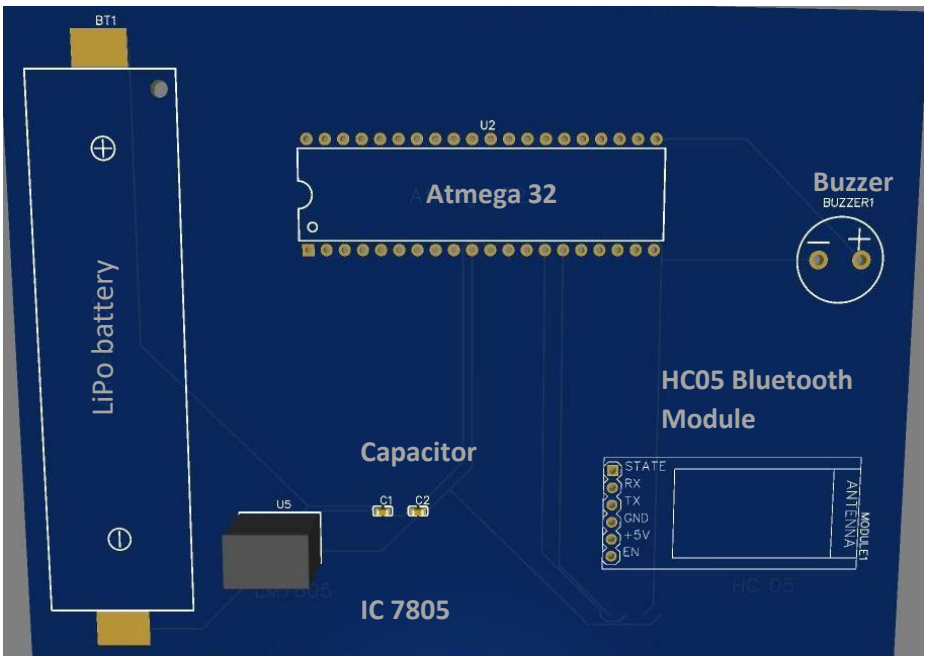
Buzzer, when the user gives the command to On the Heater and when the Heating is done It'll alert the user by the Buzzer with a beep sound.

Note: As madam mentioned since specification of these component have already been entered in the interim report because of the space limitation wont mention it again here in this report.

Circuit (Schematic Diagram)



PCB Design



Code

Buzzer:

```
else if(Data_in =='3')
{
    PORTD = 0b10000000; //buzzer on
    PORTB = 0b00000010;
    USART_SendString("Heater_ON;  ");
    lcd_cmd(0x80);
    lcd_msg("Heater_ON");
    _delay_ms(1000);
    lcd_cmd(0x01); //clear
    PORTB = 0b00000000; //buzzer off
    _delay_ms(1000);
}
```

HC05 Bluetooth Module:

```
#include <avr/io.h>
#include "USART_RS232_H_file.h"    //include USART library
#include <util/delay.h>


int main(void)
{
       char Data_in;
    DDRB = 0xff;            // make PORT as output port
    DDRC = 0xFF;            // PORTB as Output
       USART_Init(9600);    // initialize USART with 9600 baud rate

       while(1)
       {
              Data_in = USART_RxChar(); // receive data from Bluetooth device

      if(Data_in =='1')
              {
                     PORTB = 0b00000001;   // Turn ON LED
                     USART_SendString("LED_ON"); //send status of LED i.e. LED ON

              }
              else if(Data_in =='2')
              {
                     PORTB = 0b00000000;  // Turn OFF LED
                     USART_SendString("LED_OFF"); //send status of LED i.e. LED OFF
              }
              else if(Data_in =='3')
              {
                     PORTB = 0b00000010;   // Turn ON HEATER
                     USART_SendString("H_ON");
              }
              else if(Data_in =='4')
              {
                     PORTB = 0b00000000;   // Turn OFF HEATER
                     USART_SendString("H_OFF");
              }
              else if(Data_in =='5')
              {
                     PORTB = 0b00000100;   // Turn ON MOTOR
                     USART_SendString("M_ON");
              }
              else if(Data_in =='6')
              {
                     PORTB = 0b00000000;    // Turn OFF MOTOR
                     USART_SendString("M_OFF");
              }
              else if(Data_in =='w')                    //Forward
              {
                     USART_SendString("Forward");
                     PORTC = 0b00001010; //00001010
                     _delay_ms(3000);
                     PORTC = 0b00000000;
```

```
                }
                else if(Data_in =='s')                    //Backward
                {
                        USART_SendString("Backward");
                        PORTC = 0b00000101;
                        _delay_ms(3000);
                        PORTC = 0b00000000;
                }
                else if(Data_in =='a')                   //Left
                {
                        USART_SendString("Left");
                        PORTC = 0b00000010; //00001010
                        _delay_ms(1500);
                        PORTC = 0b00000000;
                }
                else if(Data_in =='d')                     //Right
                {
                        USART_SendString("Right");
                        PORTC = 0b00001000;
                        _delay_ms(1500);
                        PORTC = 0b00000000;
                }
                else if(Data_in =='7')                     //Stop
                {
                        USART_SendString("Stop");
                        PORTC = 0b00000000;
                }
                else
                        USART_SendString("Select proper option"); /* send message for
                                                                  selecting proper option */
        }
        return 0;
}
```

Member 3: Ruchirani B.M.A.D. (205088R)

Responsible parts

- HCSR-04 Ultrasonic Sensor
- DC Motor
- L298N Motor Driver
- Compiled C code for robot auto moving part
- Complete schematic circuit diagram

Technique and specification

HCSR-04 Ultrasonic Sensor

In our project, we used ultrasonic sensors to distance measurement. when the sensor sends a pulse, we can measure the distance between the sensor location and object. If the sensor detects an object between 3cm, servo motor rotates in pill case. If the sensor detects an object between 4.5cm, motor pump rotates in water pump. If the sensor detects an object between 5cm, DC motors stop in wheels. If the sensor dose not detect an object between 5cm, DC motors rotate in wheels.

Figure 1

DC Motor

Direct Current motors are used to rotate the wheels. When the ultrasonic sensor detects an object between 5cm, both of the motors stop. Then, one of the motors stop and other motor rotate to turn wheels 90^0 degree left. Otherwise, motors rotate continuously.

According to signal of HC05 Bluetooth module, user can rotate motors.

Figure 2

L298N Motor Driver

This motor driver is used to control the DC motors.

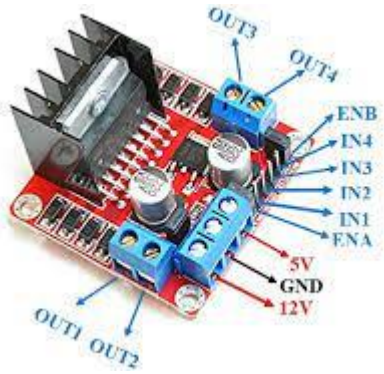
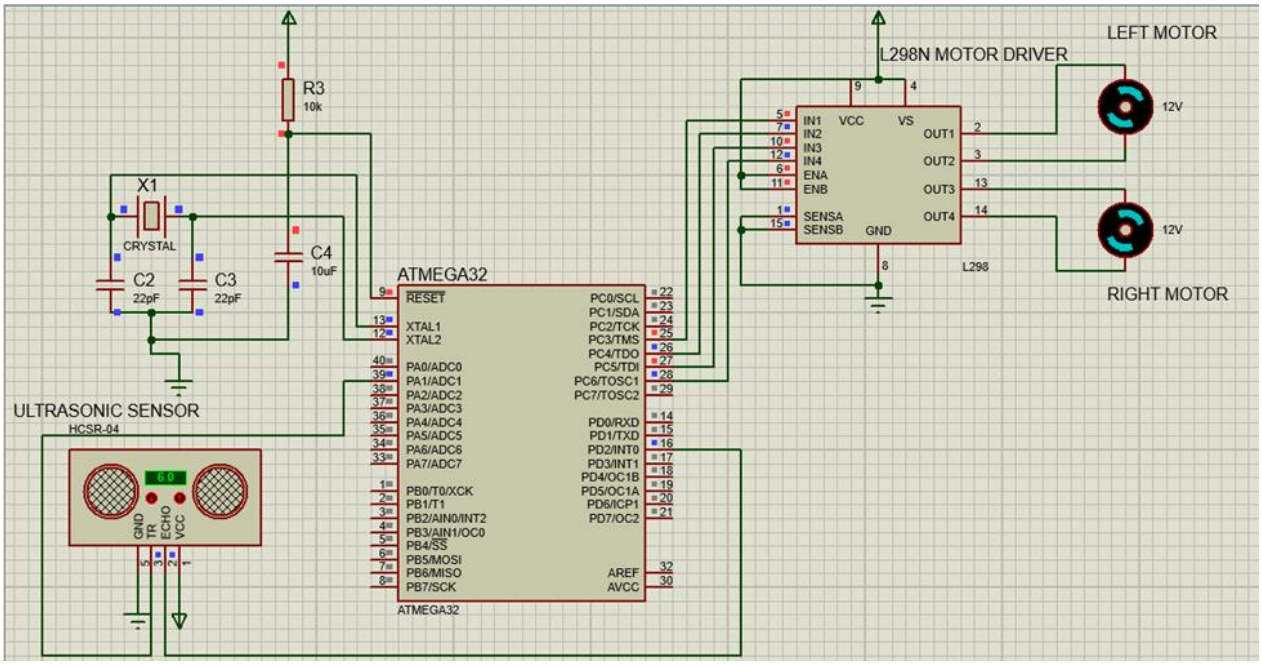
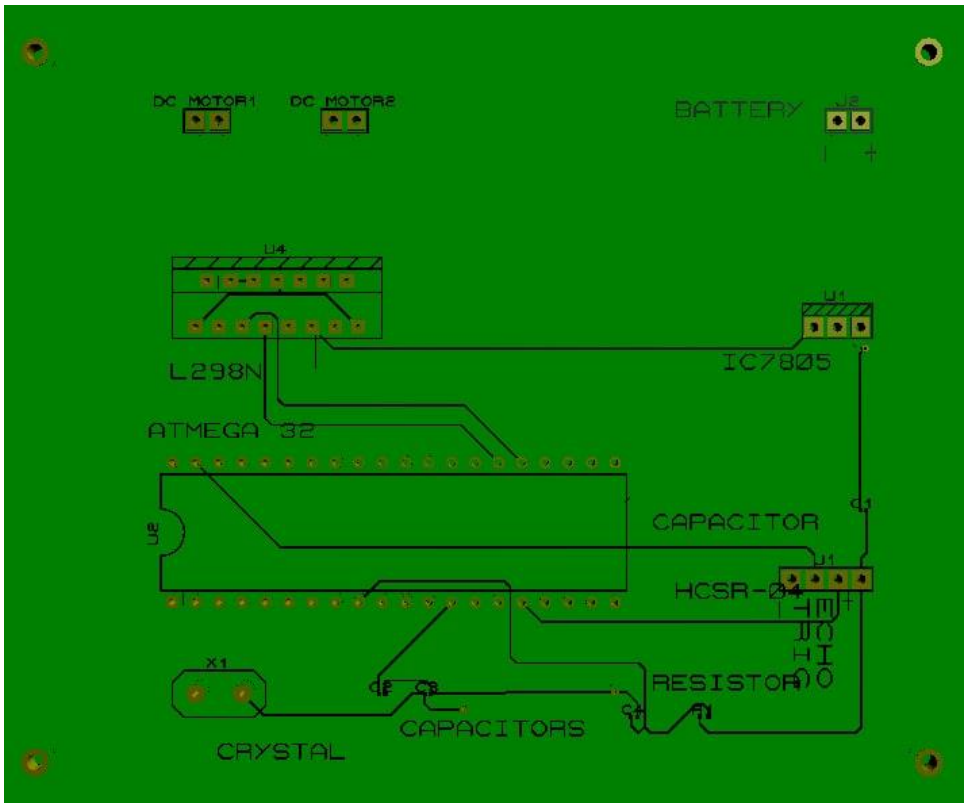


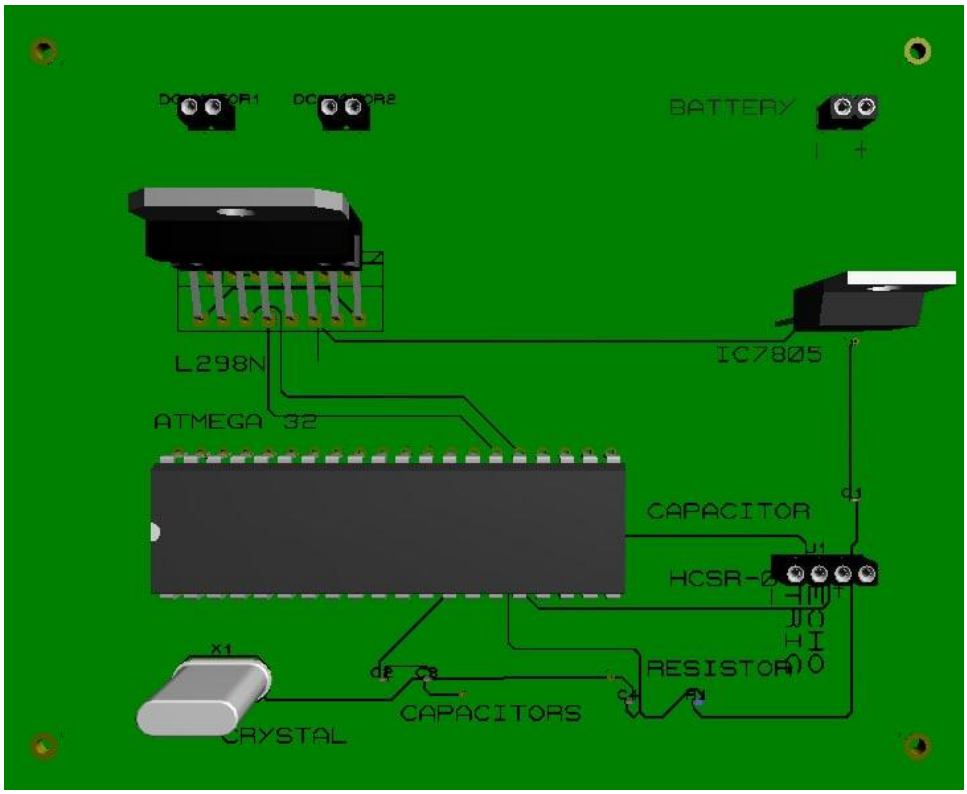
Figure 3

Circuit (Schematic Diagram)



PCB Design


DC MOTOR1
DC MOTOR2
BATTERY
L298N
IC7805
ATMEGA 32
CAPACITOR
HCSR-04
RESISTOR
CAPACITORS
CRYSTAL


BATTERY
L298N
IC7805
ATMEGA 32
CAPACITOR
RESISTOR
CAPACITORS
CRYSTAL

Code

```
/*
* AutoMove.c
*/


#define F_CPU 8000000UL
#include <avr/interrupt.h> //ISR() functions are defined in here
#include <util/delay.h>
#include <stdlib.h>
#include <avr/io.h>

static volatile unsigned int tCount = 0;//to hold the counter value from TCNT1.
static volatile int i = 0; //to indicate the state of ultrasonic sensor.
double distance;//to store the measured distance in cm

void trig(); //to send pulse

void goAhead();
void goBack();
void goLeft();
void stopRobot();
void isBlocked();

int main(void)
{
DDRC = DDRC |(1 << DDC3); // Set pin as an output
DDRC = DDRC |(1 << DDC4);
DDRC = DDRC |(1 << DDC5);
DDRC = DDRC |(1 << DDC6);
DDRA = DDRA |(1 << DDA1); // to set Trigger pin as an output
DDRD = DDRD & ~(1 << DDD2) ; // to set Echo pin(INT0 - PD2 on atmega32) as an input.
PORTD = PORTD & ~(1 << PD2) ;
```

```
_delay_ms(50);
GICR |= (1<<INT0); //to enable interrupt for INT0
MCUCR |= (1<<ISC00);//to make any logical change on INT0 generate an interrupt
TIMSK |= (1 << TOIE1);//to enable Timer/Counter 1, Overflow interrupt
sei();//to enable global interrupt bit
_delay_ms(500);
while(1)
{
isBlocked(); //to call isBlocked function
_delay_ms(200);
}
}

void isBlocked(){
//to control the motors
trig();
if (distance < 5)
{
stopRobot();
_delay_ms(2000);
goLeft();
_delay_ms(2000);
stopRobot();
isBlocked(); // recursively call function
}else{
goAhead();
}
}


ISR(INT0_vect){
// This function will get executed whenever the micro controller is interrupted by INT0.
```

```
if(i== 0)//to get executed when interrupted by a rising edge(for the first time, since i =
0)
{
TCCR1B |=(1<<CS10);//to start counting(No p rescaling). See page number 110
i=1;
}
else//to get executed when interrupted by the falling edge(for the second time, since we
set i = 1 previously)
{
TCCR1B = 0;//to stop counting
tCount = TCNT1;//to take the value of counter to our variable
distance = tCount *1000000.0/F_CPU/58;
TCNT1 = 0;//to reset counter to 0
i=0;
}
}


ISR(TIMER1_OVF_vect)
{// This function will get executed if Timer/counter 1 get overflowed
TCCR1B = 0;//to stop counting
tCount = 65535;//to make the variable ~0 when there is a counter overflow
distance = tCount *1000000.0/F_CPU/58;
TCNT1 = 0;//to reset counter to 0
i = 0;
}


void trig()
{
//Send a 15us pulse to the trigger pin
PORTA = PORTA |(1<<PA1);
_delay_us(15);
PORTA = PORTA &= ~(1<<PA1);
_delay_us(15);
```

```
}



void goAhead(){

//Left Motor
PORTC = PORTC | (1 << DDC3);
PORTC = PORTC & ~(1 << DDC4);
//Right Motor
PORTC = PORTC | (1 << DDC5);
PORTC = PORTC & ~(1 << DDC6);
}

void goBack(){

//Left Motor
PORTC = PORTC & ~(1 << DDC3);
PORTC = PORTC | (1 << DDC4);
//Right Motor
PORTC = PORTC & ~(1 << DDC5);
PORTC = PORTC | (1 << DDC6);
}

void goLeft(){
//Left Motor
PORTC = PORTC & ~(1 << DDC3); //stop
PORTC = PORTC & ~(1 << DDC4);

//Right Motor
PORTC = PORTC | (1 << DDC5); //rotate
```

```
PORTC = PORTC & ~(1 << DDC6);
}



void stopRobot(){
//Left Motor
PORTC = PORTC & ~(1 << DDC3); // stop
PORTC = PORTC & ~(1 << DDC4);
//Right Motor
PORTC = PORTC & ~(1 << DDC5); // stop
PORTC = PORTC & ~(1 << DDC6);
}
```

Member 4: Shobisha S.M (205104P)

Responsible parts

- Lcd screen display
- Heater (Through the relay module)
- Integration of the circuit diagram

Technique and specification

Lcd screen display

The Lcd screen used for display the messages on our machine. It is connected through a i2c module. When we switch on the machine the lcd screen will display the “Hey there!” message.

We give a command likewise “1,2,3,7” the relay module is on or off by the commands (“Heater on”, “Heater on”, “Led on”, “Led off”, “Motor pump on”, “Motor pump off”) at the time lcd screen will display the messages.

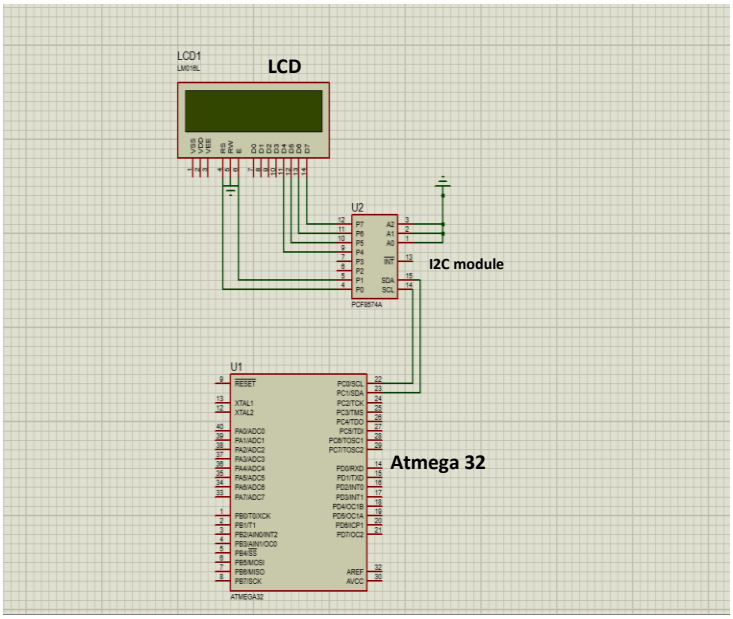
LCD 16×2 Display

Heater

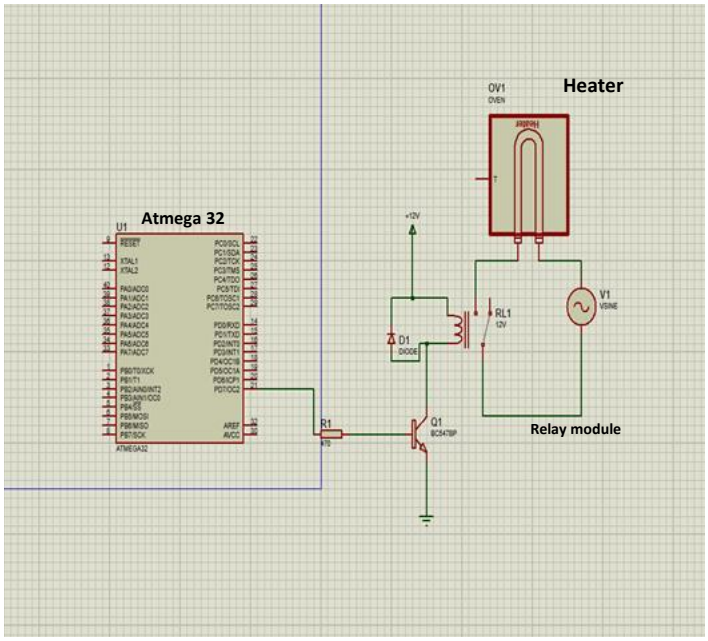
There is a heater which is used to boiling the water in the water tank of our machine. It is connected through a relay module. Whenever the user give command “Heater on”, “Heater off” to the machine through a Bluetooth module, the relay module is on or off and the heater start to boil.

Circuit (Schematic diagram)

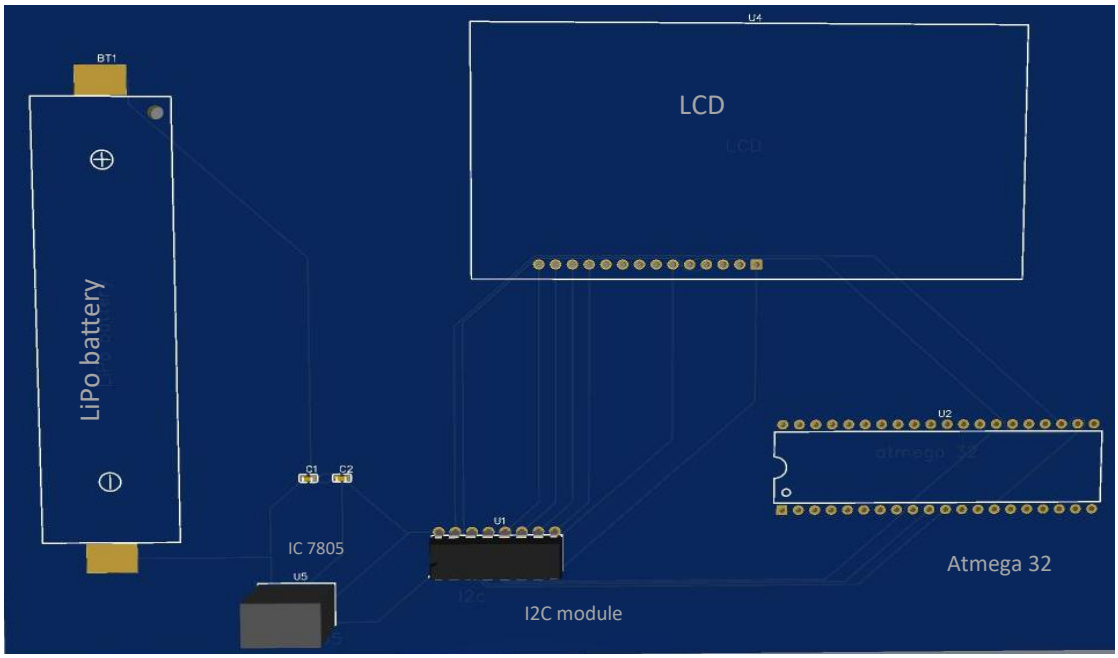
LCD



Heater



PCB Design



Code

LCD screen

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


#define  F_CPU 8000000UL


void i2c_init(){
    TWBR = 0x62;         //  Baud rate is set by calculating
    TWCR = (1<<TWEN);    //Enable I2C
    TWSR = 0x00;         //Prescaler set to 1
```

```
}
//Start condition
void i2c_start(){
    TWCR = (1<<TWINT) | (1<<TWEN) | (1<<TWSTA); //start condition
    while (!(TWCR & (1<<TWINT)));               //check for start condition


}
//I2C stop condition
void i2c_write(char x){             //Cpn esta funcion se escribe en el bus de
TWDR
    TWDR = x;                       //Move value to I2C
    TWCR = (1<<TWINT) | (1<<TWEN);  //Enable I2C and clear interrupt
    while  (!(TWCR &(1<<TWINT)));
}


char i2c_read(){
    TWCR  = (1<<TWEN) | (1<<TWINT); //Enable I2C and clear interrupt
    while (!(TWCR & (1<<TWINT)));   //Read successful with all data received in
TWDR
    return TWDR;
}
```