

Ahsanullah University of Science & Technology
Department of Computer Science & Engineering
Semester Fall 2019



CSE 3216
Microcontroller Based System Design Lab

Project Final Report

Project Name: *MULTIPURPOSE ROBOTIC CAR*

Submitted To

Afsana Ahmed Munia	Anik Chowdhury
Assistant Professor	Lecturer
CSE, AUST	CSE, AUST

Submitted By

Ahosan Rahman Siam	170104089
Shifat Islam	170104100
Nazmus Sakib	170104104

Date of Submission: September 24, 2020

Objective

This **Multipurpose robotic Car** is a four-wheeled Car/Robot that moves in a surface smoothly, detect the smoke, measures Humidity, Temperature, longitude and latitude, shows time and date through a digital clock and also works as a security device of any place. This car is a self-controlled car which can move by taking command from keypad and switching to car mode and every feature can be selected by a menu. The menu will be displayed on the LCD display and the user will be able to access the menu using keypad. Our main goal is to reduce the accident caused by fire and make the home more secured when user is not around. This is an opportunity for us to put up a smart device which can help people to their daily life and make their life easier.

Social Values

Wheeled cars have a significant advantage over humanoid type robots, in that they are faster and can move and this makes them very useful for a number of applications. Among wheeled robots, smoke detecting robots have become popular to minimize the accident caused by fire in buildings and also become popular by making the household more secured. It is also popular for measuring Humidity, Temperature of the room, measuring longitude and latitude and also for Digital clock with calendar. People are using this robot in their daily life.

Required Components

These following parts and tools are required for building this project

- **1 x Ardunio Mega 2560:** Arduino is used for running all the components and sensors. In this project Arduino mega is used for running all components and sensors.
- **1 x Lcd Display LM016L:** Lcd is used to display something in the 16x2 screen. In this project LCD is used to show output of all sensors and showing the menu to the user.
- **1 x keypad 3 x 4:** Keypad is used for taking input from the user. In this project Keypad is used to take input from user for operating menu and Car.

- **4 x Motor:** Motor is usually used for moving some objects. In this project motor is used to move the car by taking input from keypad.
- **2 x Fan(Motor):** we are using 2 motor as fan in this project to with DHT22 sensor.
- **3 x Motor Driver L293D:** Motor driver is usually used for operating and setting up the motor connect with the Arduino. In this project we doing the same thing.
- **1 x MQ-2 Gas Sensor:** Gas Sensor is used to detect Smoke/gas of the surroundings. We are using gas sensor for detecting smoke.
- **1x DHT22 Sensor:** DHT22 is used for measuring temperature and humidity of the surroundings. we are using this to for the same thing.
- **2 x Led:** Led is generally used as a part of alert system. we are using 2 led for Gas and Pir Sensor to alert the user.
- **1 x Buzzer:** Usually Buzzer is used to tone a sound for alerting. We are using Buzzer to alert the user.
- **1 x Gps Module:** Gps module is used to get the longitude and latitude for any location. we are using it for the same purpose.
- **1x DS3232RTC Module:** Generally SD3232RTC module is used as master slave with arduino to save some data on the module and update and return it to the arduino. We are using it for a digital clock and calendar.
- **1 x PirSensor:** Pir Sensor is used to for detecting IR-ray. We are using it for security mode.
- **1 x Servo Motor:** Servo motor is generally used for moving something in a angle. we are using it to locking/unlocking door.
- **2 x Button:** Button er used to take input from user. We are using it to modify/update the digital clock and calendar data.

Design

The circuit diagram is given below.

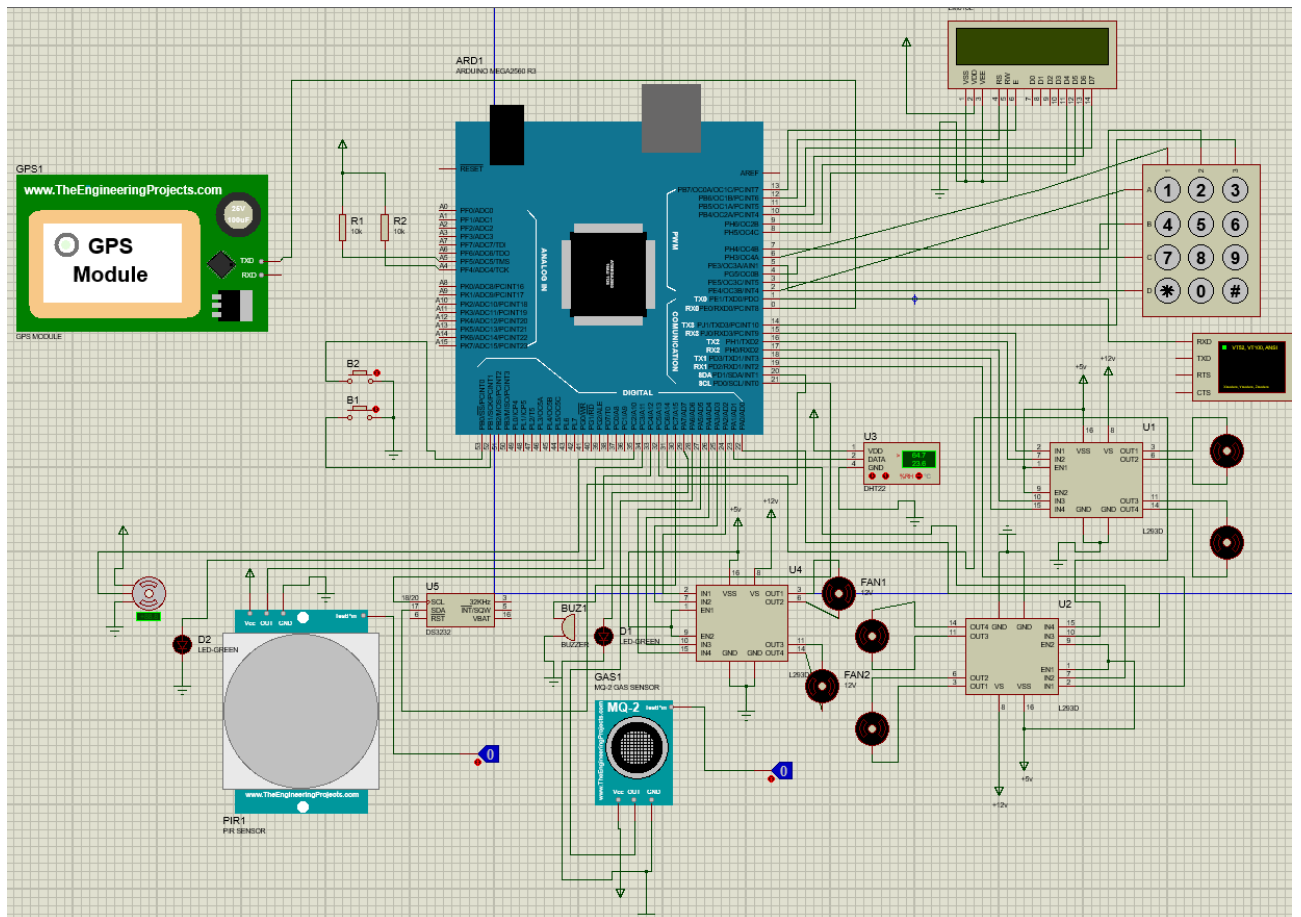


Figure 1: Diagram for MULTIPURPOSE ROBOTIC CAR

Working Procedure

The basic components that react to the input are

- 4 Wheel Motor
- LED
- Fan Motor
- Buzzer
- Servo Motor
- LCD Display
- DS3232RTC Module

The components that take stimulation from the environment are

- DHT22 Sensor
- MQ-2 Gas Sensor
- Gps Module
- Pir Sensor

The components that receive commands

- Keypad
- Button

configure the keypad buttons.

Here the keypad buttons should be configured like this:

- 9 Car Mode
- * Temp Hum Mode
- 0 Gas Mode
- Hash Gps Mode
- 7 Clock Mode
- 3 Security Mode
- 1 To start any sensor in a specific mode
- 2 To stop any specific sensor and return to main menu

First when the system starts a menu will be shown to the user by a LCD Display. The user can chose any feature by using keypad. The chosen feature will be executed. User can also return to the main menu by using Keypad. There are 6 modes in this project. These are Car Mode , Smoke Mode , Humidity Mode , Gps Mode, Clock Mode and Security Mode.

Car Mode:

- "5" is command for going forward. When we receives it via keypad both of the wheel motors start to move forward at a constant speed.
- "8" is the command for going backward. When the device receives it via Keypad both the of the wheel motors start to move Backward at a constant speed which results in the moving backward of the device.

- "6" is the command for going Right. When the device receives it via Keypad the left wheel motor starts going forward and the the Right Wheel motor starts going backward which results in the device turning right.
- "4" is the command for going Left. When the device receives it via keypad the left wheel motor starts going backward and the the Right Wheel motor starts going forward which results in the device turning Left.

Gas Mode:

- From the menu when the smoke mode is activated,then if smoke sensor detects smoke in the air it will start a alarm by a Buzzer and a led will start to blink.

Temp Humidity Mode:

- From the menu when the Temp Humidity mode is activated,then the DHT22 measured the humidity and temperature of the room. If the room temperature is more than 25 degree Celsius then a fan will start spinning forward. If the temperature is less than 25 then the fan will spin backward. For humidity if the humidity is more then 65.8 then the second fan will start spinning forward. The temperature and humidity will also be shown in the LCD display

Gps Mode:

- From the menu when the GPS mode is activated,then the GPS module will take input and show the output to the User through LCD.

Clock Mode:

- When the Clock mode is activated a Digital Clock Will be shown in the LCD Display with time and calendar. Then We and update the time and calendar by using 2 button(Button 1,Button 2).

Security Mode:

- In the Security mode the Pir sensor will be activated and the if it detects any moving objects the servo motor will lock move to a

constant degree and lock the door. user will be able to unlock the door by exiting the security mode.

Budget

Equipment	Quantity	Budget(Tk)
Arduino Mega 2560	1	700
Mini Fan	2	100
LCD Display	1	120
Keypad	1	80
Wheel Motor	4	1200
DHT22 Sensor	1	140
L293D Motor Driver Shield	3	650
MQ-2 Gas Sensor	1	120
Gps Module	1	950
Lipo Nano 1100mAh	1	1200
Lipo Battery Charger	1	500
LED	10	10
BUZZER	1	30
Pir Sensor	1	90
Servo Motor	1	300
DS3231 RTC Module	1	200
Push Button	4	20
Male to Male, Female to Female and Male to Female wire	As required	100
Male/Female headers	As required	40
Card Board	3	250
Cutter	1	50
Soldering Iron	1	150
Total		7000

Code

```

1 #include <LiquidCrystal.h>
2 #include <Keypad.h>
3 #include <TinyGPS.h>
4 #include <Wire.h>

```

```

5 #include <Servo.h>
6 Servo myservo;
7 TinyGPS gps; //Creates a new instance of the
   TinyGPS object
8
9 #include "DHT.h"
10 #define DHTPIN 23
11 float comparetemperature = 25.00, comparehumidity
   =65.80;
12 const int rs = 12, en = 13, d4 = 8, d5 = 9, d6 = 10,
   d7 = 11;
13 const byte ROWS = 4;
14 const byte COLS = 3;
15 boolean start1,start2,start3,start4,start5,start6=
   false;
16 int flag =0;
17 const int n=1;
18 bool newData = false;
19 bool newData_servo= false;
20 int GasSensor = 28;
21 int PirSensor = 33;
22 int Led = 29;
23 int Led_pir= 34;
24 int pos = 0;
25 int Buzzer = 30;
26 //int val,val_1=0;
27 char customKey,mode;
28 char hexaKeys[ROWS][COLS] = {
29     {'1','2','3'},
30     {'4','5','6'},
31     {'7','8','9'},
32     {'*','0','#'}
33 };
34 byte rowPins[ROWS] = {2, 3, 4, 5}; //connect to the
   row pinouts of the keypad
35 byte colPins[COLS] = {6, 7, 14}; //connect to the
   column pinouts of the keypad
36 Keypad customKeypad = Keypad( makeKeymap(hexaKeys),
   rowPins, colPins, ROWS, COLS);
37 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
38 int LeftMotorForward = 15;

```



```

39 int LeftMotorReverse = 16;
40 int RightMotorForward = 17;
41 int RightMotorReverse = 18;
42
43 int LeftMotorForward_2 = 19;
44 int LeftMotorReverse_2 = 31;
45 int RightMotorForward_2 = 32;
46 int RightMotorReverse_2 = 22;
47
48 int FanMotorForward = 24;
49 int FanMotorReverse = 25;
50
51 int FanMotorForward_2 = 26;
52 int FanMotorReverse_2 = 27;
53
54 #define DHTTYPE DHT22          // DHT22 sensor is used
55 DHT dht(DHTPIN, DHTTYPE);     // Initialize DHT
    library
56
57 char temperature[] = "Temp_=_00.0_C";
58 char humidity[]    = "RH_=_00.0_%";
59
60 char Time[]        = "TIME:__:__:__";
61 char Calendar[]    = "DATE:__/__/20__";
62 byte i, second, minute, hour, date, month, year;
63 void setup() {
64     // put your setup code here, to run once:
65     Serial.begin(9600);
66     pinMode(RightMotorForward, OUTPUT);
67     pinMode(RightMotorReverse, OUTPUT);
68     pinMode(LeftMotorForward, OUTPUT);
69     pinMode(LeftMotorReverse, OUTPUT);
70
71     pinMode(RightMotorForward_2, OUTPUT);
72     pinMode(RightMotorReverse_2, OUTPUT);
73     pinMode(LeftMotorForward_2, OUTPUT);
74     pinMode(LeftMotorReverse_2, OUTPUT);
75
76     pinMode(FanMotorForward, OUTPUT);
77     pinMode(FanMotorReverse, OUTPUT);
78     pinMode(FanMotorForward_2, OUTPUT);

```

```

79  pinMode(FanMotorReverse_2, OUTPUT);
80
81  pinMode(GasSensor, INPUT);
82  pinMode(Led, OUTPUT);
83
84  pinMode(Buzzer, OUTPUT);
85
86  pinMode(PirSensor, INPUT);
87  pinMode(Led_pir, OUTPUT);
88
89  pinMode(52, INPUT_PULLUP);
90  pinMode(53, INPUT_PULLUP);
91
92  myservo.attach(35);
93
94  dht.begin();
95  Wire.begin();
96  lcd.begin(16, 2);
97  lcd.setCursor(0, 0);
98  lcd.clear();
99  lcd.print("Initializing..");
100 delay(500);
101 lcd.clear();
102 lcd.setCursor(5, 0);
103 delay(500);
104 lcd.print("Hello");
105 delay(2000);
106 lcd.clear();
107 lcd.setCursor(1, 0);
108 lcd.print("Welcome_to_the");
109 lcd.setCursor(5, 1);
110 lcd.print("System");
111 delay(500);
112 lcd.clear();
113 for(int i=0; i<n; i++){
114   Menu_Display();
115 }
116 }
117 void loop() {
118   // put your main code here, to run repeatedly:
119   //Menu_Display();

```

```

120  if(flag==0) {
121      Menu_Operate();
122  }
123  if(start1==true)
124  { carmode();
125  }
126  else if(start2==true)
127  {
128      temp_hummode();
129  }
130  else if(start3==true)
131  {
132      gasmode();
133  }
134  else if(start4==true)
135  {
136      gpsmode();
137  }
138  else if(start5==true)
139  {
140      clockmode();
141  }
142  else if(start6==true)
143  {
144      securitymode();
145  }
146 }
147 void Menu_Display()
148 {
149     lcd.setCursor(4,0);
150     lcd.print("Press_9_for");
151     lcd.setCursor(5,1);
152     lcd.print("car_mode");
153     Serial.println("Press_9_for_car_mode");
154     delay(500);
155     lcd.clear();
156
157     lcd.setCursor(4,0);
158     lcd.print("Press_*_for");
159     lcd.setCursor(2,1);
160     lcd.print("Temp_hum_mode");

```

```

161  Serial.println("Press_*_for_Temp_hum_mode");
162  delay(500);
163  lcd.clear();
164
165  lcd.setCursor(4,0);
166  lcd.print("Press_0_for");
167  lcd.setCursor(5,1);
168  lcd.print("Gas_mode");
169  Serial.println("Press_0_for_gas_mode");
170  delay(500);
171  lcd.clear();
172
173  lcd.setCursor(4,0);
174  lcd.print("Press_#_for");
175  lcd.setCursor(5,1);
176  lcd.print("Gps_mode");
177  Serial.println("Press_#_for_gps_mode");
178  delay(500);
179  lcd.clear();
180
181  lcd.setCursor(4,0);
182  lcd.print("Press_7_for");
183  lcd.setCursor(5,1);
184  lcd.print("Clock_mode");
185  Serial.println("Press_7_for_Clock_mode");
186  delay(500);
187  lcd.clear();
188
189  lcd.setCursor(4,0);
190  lcd.print("Press_3_for");
191  lcd.setCursor(2,1);
192  lcd.print("Security_mode");
193  Serial.println("Press_3_for_Security_mode");
194  delay(500);
195  lcd.clear();
196 }
197 void Menu_Operate()
198 {
199     customKey = customKeypad.getKey();
200     Serial.println(customKey);
201     switch(customKey)

```

```

202 {
203     case '9':
204         lcd.clear();
205         lcd.setCursor(4,0);
206         lcd.print("Car_mode");
207         lcd.setCursor(5,1);
208         lcd.print("activated");
209         Serial.println("car_mode_activated");
210         delay(500);
211         lcd.clear();
212         start1=true;
213         break;
214
215     case '*':
216         lcd.clear();
217         lcd.setCursor(1,0);
218         lcd.print("Temp_hum_mode");
219         lcd.setCursor(5,1);
220         lcd.print("activated");
221         Serial.println("Temp_hum_mode_activated");
222         delay(500);
223         lcd.clear();
224         start2=true;
225         break;
226
227     case '0':
228         lcd.clear();
229         lcd.setCursor(4,0);
230         lcd.print("Gas_mode");
231         lcd.setCursor(5,1);
232         lcd.print("activated");
233         Serial.println("Gas_mode_activated");
234         delay(500);
235         lcd.clear();
236         start3=true;
237         break;
238
239     case '#':
240         lcd.clear();
241         lcd.setCursor(4,0);
242         lcd.print("GPS_mode");

```

```

243     lcd.setCursor(5,1);
244     lcd.print("activated");
245     Serial.println("Gps_mode_activated");
246     delay(500);
247     lcd.clear();
248     start4=true;
249     newData=false;
250     break;
251
252     case '7':
253     lcd.clear();
254     lcd.setCursor(4,0);
255     lcd.print("Clock_mode");
256     lcd.setCursor(5,1);
257     lcd.print("activated");
258     Serial.println("Clock_mode_activated");
259     delay(500);
260     lcd.clear();
261     start5=true;
262     break;
263
264     case '3':
265     lcd.clear();
266     lcd.setCursor(1,0);
267     lcd.print("Security_mode");
268     lcd.setCursor(5,1);
269     lcd.print("activated");
270     Serial.println("Security_mode_activated");
271     delay(500);
272     lcd.clear();
273     start6=true;
274     newData_servo= false;
275     break;
276 }
277 }
278 void carmode()
279 {
280     lcd.setCursor(0,0);
281     lcd.print("Press_1_to_start");
282     Serial.println("press_1_to_start");
283     //lcd.setCursor(0,1);

```

```

284 //lcd.print("press 2 to stop");
285 delay(500);
286 flag=1;
287 while(flag==1){
288 customKey = customKeypad.getKey();
289 Serial.println(customKey);
290 if(customKey=='1')
291 {
292     lcd.clear();
293     while(flag==1){
294         lcd.setCursor(0,0);
295         lcd.print("Car_started");
296         lcd.setCursor(0,1);
297         lcd.print("press_2_to_stop");
298         customKey = customKeypad.getKey();
299         Serial.println(customKey);
300         switch(customKey)
301         { case '4':
302             //Left Turn
303             digitalWrite(RightMotorForward, LOW);
304             digitalWrite(RightMotorReverse, HIGH);
305             digitalWrite(LeftMotorForward, LOW);
306             digitalWrite(LeftMotorReverse, LOW);
307
308             digitalWrite(RightMotorForward_2, LOW);
309             digitalWrite(LeftMotorForward_2, LOW);
310             digitalWrite(RightMotorReverse_2, HIGH);
311             digitalWrite(LeftMotorReverse_2, HIGH);
312
313             break;
314
315             case '6':
316                 //Right Turn
317                 digitalWrite(LeftMotorForward, HIGH);
318                 digitalWrite(LeftMotorReverse, LOW);
319                 digitalWrite(RightMotorForward, LOW);
320                 digitalWrite(RightMotorReverse, LOW);
321
322                 digitalWrite(RightMotorForward_2, LOW);
323                 digitalWrite(LeftMotorForward_2, LOW);
324                 digitalWrite(RightMotorReverse_2, HIGH);

```

```

325     digitalWrite(LeftMotorReverse_2, HIGH);
326
327     break;
328
329     case '5':
330         // Forward
331         digitalWrite(RightMotorForward, HIGH);
332         digitalWrite(LeftMotorForward, HIGH);
333         digitalWrite(RightMotorReverse, LOW);
334         digitalWrite(LeftMotorReverse, LOW);
335
336         digitalWrite(RightMotorForward_2, LOW);
337         digitalWrite(LeftMotorForward_2, LOW);
338         digitalWrite(RightMotorReverse_2, HIGH);
339         digitalWrite(LeftMotorReverse_2, HIGH);
340         break;
341
342     case '8':
343         //Reverse
344         digitalWrite(RightMotorReverse, HIGH);
345         digitalWrite(LeftMotorReverse, HIGH);
346         digitalWrite(RightMotorForward, LOW);
347         digitalWrite(LeftMotorForward, LOW);
348
349         digitalWrite(RightMotorReverse_2, LOW);
350         digitalWrite(LeftMotorReverse_2, LOW);
351         digitalWrite(RightMotorForward_2, HIGH);
352         digitalWrite(LeftMotorForward_2, HIGH);
353
354         break;
355     case '2':
356         lcd.clear();
357         digitalWrite(RightMotorReverse, LOW);
358         digitalWrite(LeftMotorReverse, LOW);
359         digitalWrite(RightMotorForward, LOW);
360         digitalWrite(LeftMotorForward, LOW);
361
362         digitalWrite(RightMotorReverse_2, LOW);
363         digitalWrite(LeftMotorReverse_2, LOW);
364         digitalWrite(RightMotorForward_2, LOW);
365         digitalWrite(LeftMotorForward_2, LOW);

```



```

366         flag=0;
367         start1=false;
368         for(int i=0;i<n;i++){
369             Menu_Display();
370         }
371         break;
372     }
373 }
374 }
375 }
376 }
377 void temp_hummode()
378 {
379     lcd.setCursor(0,0);
380     lcd.print("Press_1_to_start");
381     Serial.println("press_1_to_start");
382     lcd.setCursor(0,1);
383     lcd.print("press_2_to_stop");
384     Serial.println("press_2_to_stop");
385     delay(500);
386     flag=1;
387     while(flag==1){
388         customKey = customKeypad.getKey();
389         Serial.println(customKey);
390         if(customKey=='1')
391         {
392             lcd.clear();
393             while(flag==1){
394                 lcd.setCursor(0,0);
395
396                 // Read humidity
397                 int RH = dht.readHumidity() * 10;
398                 //Read temperature in degree Celsius
399                 int Temp = dht.readTemperature() * 10;
400
401                 // Check if any reads failed and exit early
402                 // (to try again)
403                 if (isnan(RH) || isnan(Temp)) {
404                     lcd.clear();
405                     lcd.setCursor(5, 0);
406                     lcd.print("Error");

```

```

406         return;
407     }
408
409     if(Temp < 0){
410         temperature[6] = '-';
411         Temp = abs(Temp);
412     }
413
414     else
415         {temperature[6] = '┐';}
416     temperature[7] = (Temp / 100) % 10 + 48;
417     temperature[8] = (Temp / 10) % 10 + 48;
418     temperature[10] = Temp % 10 + 48;
419     temperature[11] = 223; // Degree
420     symbol ( )
421
422     if(RH >= 1000){
423         humidity[6] = '1';}
424     else{
425         humidity[6] = '┐';}
426     humidity[7] = (RH / 100) % 10 + 48;
427     humidity[8] = (RH / 10) % 10 + 48;
428     humidity[10] = RH % 10 + 48;
429
430     lcd.setCursor(0, 0);
431     lcd.print(temperature);
432     lcd.setCursor(0, 1);
433     lcd.print(humidity);
434
435     float tempc = dht.readTemperature();
436     float rhc = dht.readHumidity();
437
438     if (tempc > comparetemperature ) {
439         digitalWrite(FanMotorForward, HIGH);
440         digitalWrite(FanMotorReverse, LOW);
441     }
442     if (tempc < comparetemperature) {
443         digitalWrite(FanMotorForward, LOW);
444         digitalWrite(FanMotorReverse, HIGH);
445     }
446     if (rhc > comparehumidity ) {
447         digitalWrite(FanMotorForward_2, HIGH);

```

```

446         digitalWrite(FanMotorReverse_2, LOW);
447     }
448     if (rhc < comparehumidity) {
449         digitalWrite(FanMotorForward_2, LOW);
450         digitalWrite(FanMotorReverse_2, LOW);
451     }
452     customKey = customKeypad.getKey();
453     Serial.println(customKey);
454     switch(customKey)
455     { case '2':
456         lcd.clear();
457         digitalWrite(FanMotorForward, LOW);
458         digitalWrite(FanMotorReverse, LOW);
459         digitalWrite(FanMotorForward_2, LOW);
460         digitalWrite(FanMotorReverse_2, LOW);
461         flag=0;
462         start2=false;
463         for(int i=0;i<n;i++){
464             Menu_Display();
465         }
466         break;
467     }
468 }
469 }
470 }
471 }
472 void gasmode()
473 {
474     lcd.setCursor(0,0);
475     lcd.print("Press_1_to_start");
476     delay(500);
477     flag=1;
478     while(flag==1){
479         customKey = customKeypad.getKey();
480         Serial.println(customKey);
481         if(customKey=='1')
482         {
483             lcd.clear();
484             while(flag==1){
485                 lcd.setCursor(0,0);
486                 lcd.print("Sensor_started");

```

```

487     lcd.setCursor(0,1);
488     lcd.print("press_2_to_stop");
489     if(digitalRead(GasSensor)==HIGH)
490     {
491         digitalWrite(Led,HIGH);
492         tone(Buzzer,1000);
493         delay(1000);
494         noTone(Buzzer);
495     }
496     else
497     {
498         digitalWrite(Led,LOW);
499     }
500     customKey = customKeypad.getKey();
501     Serial.println(customKey);
502     switch(customKey)
503     { case '2':
504         lcd.clear();
505         flag=0;
506         start3=false;
507         for(int i=0;i<n;i++){
508             Menu_Display();
509         }
510         break;
511     }
512 }
513 }
514 }
515 }
516 void gpsmode()
517 {
518     lcd.setCursor(0,0);
519     lcd.print("Press_1_to_start");
520     Serial.println("press_1_to_start");
521     lcd.setCursor(0,1);
522     lcd.print("press_2_to_stop");
523     Serial.println("press_2_to_stop");
524     delay(500);
525     flag=1;
526     while(flag==1){
527         customKey = customKeypad.getKey();

```

```

528     Serial.println(customKey);
529     if(customKey=='1')
530     {
531         lcd.clear();
532         while(flag==1){
533             lcd.setCursor(0,0);
534             /*lcd.print("Sensor started");
535             lcd.setCursor(0,1);
536             lcd.print("press 2 to stop");*/
537             unsigned long chars;
538             unsigned short sentences, failed;
539
540             // For one second we parse GPS data and report
             some key values
541             if(newData==false){
542                 for (unsigned long start = millis(); millis()
                    - start < 1000;)
543                 {
544                     while (Serial.available())
545                     {
546                         char c = Serial.read();
547                         gps.encode(c);
548                     }
549                 }
550
551                 float flat, flon;
552                 unsigned long age;
553                 gps.f_get_position(&flat, &flon, &age);
554                 Serial.print("Latitude=_");
555                 Serial.print(flat == TinyGPS::
                    GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
556                 Serial.print("_Longitude=_");
557                 Serial.print(flton == TinyGPS::
                    GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
558                 lcd.setCursor(0, 0);
559                 lcd.print("Latitude=");
560                 lcd.setCursor(10, 0);
561                 lcd.print(flat);
562                 lcd.setCursor(0, 1);
563                 lcd.print("Longitude=");
564                 lcd.setCursor(11, 1);

```

```

565         lcd.print(flon);
566
567         Serial.println(failed);
568         newData =true;
569     }
570     customKey = customKeypad.getKey();
571     Serial.println(customKey);
572     switch(customKey)
573     { case '2':
574         lcd.clear();
575         flag=0;
576         start4=false;
577         newData=false;
578         for(int i=0;i<n;i++){
579             Menu_Display();
580         }
581         break;
582     }
583 }
584 }
585 }
586 }
587 void blink_parameter(){
588     byte j = 0;
589     while(j < 10 && digitalRead(52) && digitalRead(53)
590         ){
591         j++;
592         delay(25);
593     }
594 }
595 byte edit(byte x, byte y, byte parameter){
596     char text[3];
597     while(!digitalRead(52)); //
598     // Wait until button (pin #52) released
599     while(true){
600         while(!digitalRead(53)){ //
601             // If button (pin #53) is pressed
602             parameter++;
603             if(i == 0 && parameter > 23) //
604                 // If hours > 23 ==> hours = 0
605                 parameter = 0;

```

```

602     if(i == 1 && parameter > 59)                                //
        If minutes > 59 ==> minutes = 0
603     parameter = 0;
604     if(i == 2 && parameter > 31)                                //
        If date > 31 ==> date = 1
605     parameter = 1;
606     if(i == 3 && parameter > 12)                                //
        If month > 12 ==> month = 1
607     parameter = 1;
608     if(i == 4 && parameter > 99)                                //
        If year > 99 ==> year = 0
609     parameter = 0;
610     sprintf(text, "%02u", parameter);
611     lcd.setCursor(x, y);
612     lcd.print(text);
613     delay(50);                                                  //
        Wait 50ms
614 }
615 lcd.setCursor(x, y);
616 lcd.print("__");                                                //
        Display two spaces
617 blink_parameter();
618 sprintf(text, "%02u", parameter);
619 lcd.setCursor(x, y);
620 lcd.print(text);
621 blink_parameter();
622 if(!digitalRead(52)) {                                         //
        If button (pin #52) is pressed
623     i++;                                                        //
        Increament 'i' for the next parameter
624     return parameter;                                          //
        Return parameter value and exit
625 }
626 }
627 }
628 void DS3231_display() {
629     // Convert BCD to decimal
630     second = (second >> 4) * 10 + (second & 0x0F);
631     minute = (minute >> 4) * 10 + (minute & 0x0F);
632     hour    = (hour >> 4) * 10 + (hour & 0x0F);
633     date    = (date >> 4) * 10 + (date & 0x0F);

```

```

634 month = (month >> 4) * 10 + (month & 0x0F);
635 year = (year >> 4) * 10 + (year & 0x0F);
636 // End conversion
637 Time[12] = second % 10 + 48;
638 Time[11] = second / 10 + 48;
639 Time[9] = minute % 10 + 48;
640 Time[8] = minute / 10 + 48;
641 Time[6] = hour % 10 + 48;
642 Time[5] = hour / 10 + 48;
643 Calendar[14] = year % 10 + 48;
644 Calendar[13] = year / 10 + 48;
645 Calendar[9] = month % 10 + 48;
646 Calendar[8] = month / 10 + 48;
647 Calendar[6] = date % 10 + 48;
648 Calendar[5] = date / 10 + 48;
649 lcd.setCursor(0, 0);
650 lcd.print(Time);
    // Display time
651 lcd.setCursor(0, 1);
652 lcd.print(Calendar);
    // Display calendar
653 }
654 void clockmode()
655 {
656     lcd.setCursor(0, 0);
657     lcd.print("Press_1_to_start");
658     Serial.println("press_1_to_start");
659     lcd.setCursor(0, 1);
660     lcd.print("press_2_to_stop");
661     Serial.println("press_2_to_stop");
662     delay(500);
663     flag=1;
664     lcd.clear();
665     while(flag==1) {
666         customKey = customKeypad.getKey();
667         Serial.println(customKey);
668         if(customKey=='1')
669             {
670                 while(flag==1) {
671                     //lcd.clear();
672                     if(!digitalRead(52)) {

```



```

673     i = 0;
674     hour   = edit(5, 0, hour);
675     minute = edit(8, 0, minute);
676     date   = edit(5, 1, date);
677     month  = edit(8, 1, month);
678     year   = edit(13, 1, year);
679     // Convert decimal to BCD
680     minute = ((minute / 10) << 4) + (minute %
        10);
681     hour = ((hour / 10) << 4) + (hour % 10);
682     date = ((date / 10) << 4) + (date % 10);
683     month = ((month / 10) << 4) + (month % 10);
684     year = ((year / 10) << 4) + (year % 10);
685     // End conversion
686     // Write data to DS3231 RTC
687     Wire.beginTransmission(0x68);
        // Start I2C protocol with DS3231 address
688     Wire.write(0);
        // Send register address
689     Wire.write(0);
        // Reset seconds and start oscillator
690     Wire.write(minute);
        // Write minute
691     Wire.write(hour);
        // Write hour
692     Wire.write(1);
        // Write day (not used)
693     Wire.write(date);
        // Write date
694     Wire.write(month);
        // Write month
695     Wire.write(year);
        // Write year
696     Wire.endTransmission();
        // Stop transmission and release the I2C
        bus
697     delay(75);
        // Wait 200ms
698 }
699     Wire.beginTransmission(0x68);
        // Start I2C protocol

```

```

700         with DS3231 address
Wire.write(0);
// Send
        register address
701 Wire.endTransmission(false);
// I2C restart
702 Wire.requestFrom(0x68, 7);
// Request 7 bytes
        from DS3231 and release I2C bus at end of
        reading
703 second = Wire.read();
// Read seconds
        from register 0
704 minute = Wire.read();
// Read minuts
        from register 1
705 hour = Wire.read();
// Read hour from
        register 2
706 Wire.read();
// Read
        day from register 3 (not used)
707 date = Wire.read();
// Read date
        from register 4
708 month = Wire.read();
// Read month
        from register 5
709 year = Wire.read();
// Read year
        from register 6
710 DS3231_display();
// Diaplay
        time & calendar
711 delay(50);
712 customKey = customKeypad.getKey();
713 Serial.println(customKey);
714 if(customKey=='2')
715 {
716     flag=0;
717     start5=false;

```

```

718         lcd.clear();
719         for(int i=0;i<n;i++){
720             Menu_Display();
721         }
722     }
723     else
724     {
725         continue;
726     }
727 }
728 }
729 }
730 }
731 void securitymode()
732 {
733     lcd.setCursor(0,0);
734     lcd.print("Press_1_to_start");
735     //lcd.setCursor(0,1);
736     //lcd.print("press 2 to stop");
737     delay(500);
738     flag=1;
739     while(flag==1){
740         customKey = customKeypad.getKey();
741         Serial.println(customKey);
742         if(customKey=='1')
743         {
744             lcd.clear();
745             while(flag==1){
746                 lcd.setCursor(0,0);
747                 lcd.print("Sensor_started");
748                 lcd.setCursor(0,1);
749                 lcd.print("press_2_to_stop");
750                 if(digitalRead(PirSensor)==HIGH)
751                 {
752                     digitalWrite(Led_pir,HIGH);
753                     if(newData_servo==false){
754
755                         myservo.write(180);
756                         delay(20);
757                         newData_servo==true;
758                     }

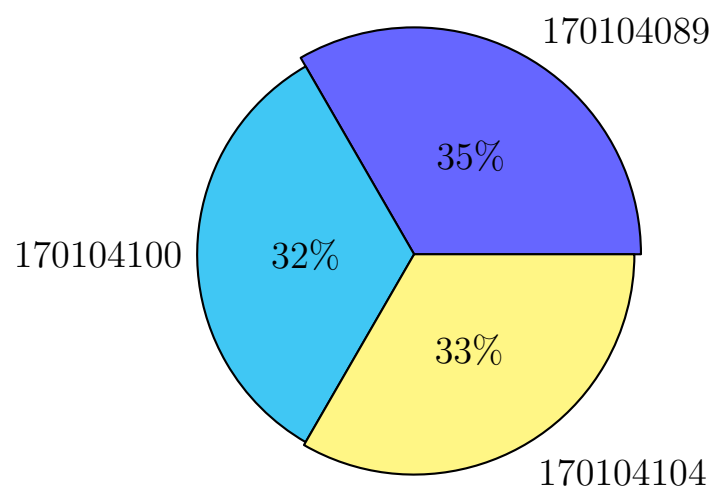
```

```

759
760     }
761 else
762 {
763     digitalWrite(Led_pir, LOW);
764 }
765 customKey = customKeypad.getKey();
766 Serial.println(customKey);
767 switch(customKey)
768 { case '2':
769     lcd.clear();
770     digitalWrite(Led_pir, LOW);
771     myservo.write(90);
772     delay(20);
773     //newData_servo==false;
774     flag=0;
775     start6=false;
776     for(int i=0;i<n;i++){
777         Menu_Display();
778     }
779     break;
780 }
781 }
782 }
783 }
784 }

```

Members Contribution



Difficulties

We faced some difficulties while doing this project. We faced some problem while adding gps module And RTC module. The most difficult part of our project was to merge all the sensor in a single menu and executing them by a single menu.

Future Work

In Future we want to add some extra feature this project like speaker, Image showing Display Etc and fix the bugs(If needed).

Conclusion

Taking the problems of our loved ones and the people of our society in mind we will make an effort of improving our lives by making a problem go away. Although humble in the scale and usability of the device we hope that the impact it will create will be greater than the size of the instrument.