

# Implementing Behaviour based hybrid control for Intelligent Vehicles

Ms.Gayathiri.J<sup>1</sup>, Mr.Ganesha Perumal.D<sup>2</sup>Department of EEE, Kalasalingam University, Tamilnadu, India <sup>1,2</sup>

**Abstract:** PEV (Plug-in Electric Vehicle) are penetrating the vehicle market due to increasing energy costs, and carbon foot-print. PEV that can manoeuvre automatically are referred to be autonomous, and require automation. Four major operations in autonomous PEV are- cruise control, automated steering control, braking, and path-planning. Each of these operations in-turn is composed of minor tasks and these tasks are called behaviour. As an example, steering control is composed of behaviour such as left-turn, right-turn, lane changing, obstacle avoidance etc. These behaviours are dependent on the lane conditions, and path. Therefore, vehicle needs to switch among these behaviours leading to hybrid behaviour wherein there is discrete switching among this continuous behaviour. Control design for such hybrid behaviour is complex, and implementing such control actions in dedicated embedded platform is challenging. In this paper first a layered architecture for implementing hybrid controller for autonomous vehicle is presented. There are three layers in this architecture, the first layer is the higher-level control, second layer is called the hybrid control layer, and the third layer is called lower level control. Further, this paper discusses the implementation of lower-level control in an embedded platform such as 8051. In order to implement the lower-level controller, first the performance of the controller is verified in MATLAB using simulation, and then the code is hard-coded in embedded C and ported into the microcontroller. This provides a methodology to perform hardware-in-loop simulation for the autonomous PEV. Simulation results in MATLAB are used to illustrate the working of the hybrid controller, while verification is used to ascertain the implementation.

**Keywords:** Plug-in-electric vehicle (PEV), behaviour based control, hybrid control, layered control architecture, autonomous vehicle

## I. INTRODUCTION

Dwindling fossil fuels, climate change, and increasing fuel prices have necessitated researchers to search for alternatives. Plug-in-Electric Vehicle (PEV) has emerged as a promising solution in this scenario. Interest in PEV is increasing both in academia and industry with each passing day. Further, there has been significant development in recent years in autonomous driving. However, these developments have not been used in building PEV. In particular, use of technologies for building autonomous PEV for physically challenged can enhance and assist disabled people. This has not been investigated by researchers extensively mainly due to the challenge in implementing controllers for autonomous operations. Motivated by this intelligent vehicle for speech and hearing impaired people (IVSHIP) [<http://ivship.weebly.com/>] project has taken up the task of designing an autonomous PEV for such special students. This paper discusses the implementation aspects of this vehicle, and presents the current results of the project.

Wirasingha et al. [3] presented a review on available control methods for PEV and classified controllers based on the models they employed as- i) rule-based and ii) model based controllers. But, these controllers are motivated only to reduce fuel (electricity) consumption. Assist systems have been traditionally used in vehicle design (see, [10] and references therein), but the use of assist systems for physically challenged has been investigated minimally. In particular, controller design to aid assist systems has not been investigated. Lygeros et al. [5] designed verified hybrid controllers for IVHS with switching between modes. A verification platform TIGA

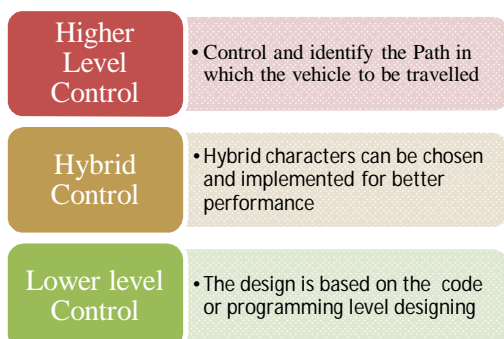
has been used to verify safety. TIGA is a combination of UPPAAL time automated model and Game theoretic based approach. The approach has been used for manual driven vehicles. However, the role of hybrid controllers has not been investigated. In autonomic robotics behaviour based control has been studied for navigation problems by considering obstacles and other objectives in [R. Arkin]. Results obtained in behaviour based robotics have not been used in vehicle control. Our objectives is to combine hybrid controller design with results in behaviour based robotics, and this investigation is motivated to propose control architecture and lower level implementation of such controller.

Main contributions of this paper are: (i) Layered architecture for implementing hybrid controllers for autonomous PEV, and (ii) present the implementation of such controllers in dedicated embedded platform.

This paper is organized into 4 sections including the introduction. Section II, presents the basic operations of the vehicle. Section III presents the autonomous behaviours of the vehicle. Section IV presents layered based architecture and its factors. Section V presents the function of unicycle dynamics. Section VI presents the simulation results that have been obtained by unicycle dynamics.

## II. LAYERED ARCHITECTURE

Layered architecture may have three states. In the case of behaviour based hybrid control for intelligent vehicle two basic states can be considered. The essential states those can be accomplished with intelligent vehicle layered architecture are higher level control and lower level control.



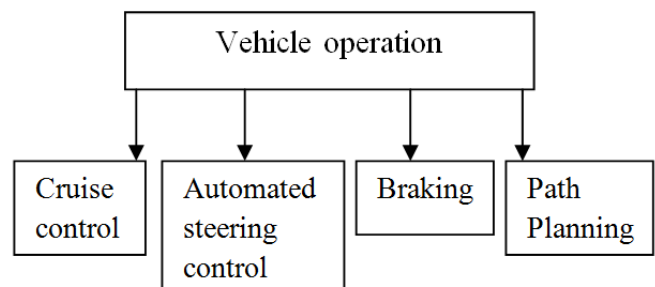
### i. HIGHER LEVEL CONTROL

Higher level control needs to control the characteristics of the vehicle physically. Path planning involves the strategy of framing the way to attain the target. There are many constraints can be associated with the path planning mechanism. If there is no obstacle through the path of the vehicle then it will take a straight and framed outline motion. If there is any obstacle found through the path then the vehicle needs to have some analysis beyond the framed work layout. It can be described as the secure approach with the static and dynamic obstacles. So, the path planning mechanism does not only involved with the waypoint strategy but also with the line following and range detecting mechanism. The path planning leads to follow a framed path at offline situations. The vehicle should undergo the steering commands at path planning condition. Path planning can also be applied when the vehicle needs to travel in an unknown environment. The smooth paths can be finding out by applying efficient algorithms. In order to drive the autonomous vehicle in an unknown environment, it should have the capability of obstacle mapping and adequate sensing.

### ii. LOWER LEVEL CONTROL:

In lower level control design the MATLAB platform has been employed in the case of behaviour based hybrid control for intelligent vehicles. In order to port it to hardware the c code has been generated from the MATLAB code. The developing environment is highly reliable for control mechanisms while working with the MATLAB. The appropriate implementation of code level design decides the efficiency of the code level mechanism.

## III. OPERATIONS OF THE VEHICLE



*i. CRUISE CONTROL*

Cruise control has been employed in automobiles in order to maintain the constant speed. Autonomous vehicle need to maintain a safe distance from the leader vehicle as well as from the follower vehicle. So, cruise control is used in the vehicle dynamics.

*ii. AUTOMATED STEERING CONTROL*

Automated steering control is highly preferred in autonomous vehicles since it leads to have a simplified mode of controlling the steering position.

*iii. BRAKING*

Braking system is an essential factor for the protective control of the vehicle. There are various techniques can be applicable for braking purpose based on the specific application.

*iv. PATH PLANNING*

Planning the path of an autonomous, sprightly vehicle in a dynamic atmosphere is a face up to, especially when the vehicle is required to use its full manoeuvring capabilities. Recent hard work aimed at using randomized algorithms for planning the path of kinematic and dynamic vehicles have established substantial prospective for implementation on future autonomous platforms. The essential efforts can be attained by proposing a randomized path planning architecture for dynamical systems in the attendance of fixed and moving obstacles. The path planning algorithm retains the convergence properties of its kinematic counterparts.

**IV. BEHAVIOUR BASED HYBRID CONTROLLER**

The safety of the system is also addressed in the face of finite computation times, by analyzing the behaviour of the algorithm when the available on-board computation resources are limited the planning is need to be performed in real-time. The basic autonomous behaviours are i) Go-to-goal, ii) Obstacle avoidance.

*i) GO TO GOAL*

Go to goal establishes the travelling behaviour of the autonomous vehicle from a waypoint to the final destination. The proposed algorithm can be applied to vehicles whose dynamics are described either by ordinary differential equations or by higher-level, hybrid representations (fig2.1). This function is designed to provide attractive “forces” towards the goal, and repulsive “forces” which push the vehicles away from obstacles. Notice that this class of methods is based on the definition of a feedback control policy (i.e. the control is computed at each instant in time as a function of the current state), as opposed to the open-loop approach of the previous two classes. A inadequacy of this formulation is the potential subsistence of local minima, in which the vehicles might become trapped. An artificial potential function which does not have local minima is said a routing function, but computing such a function is in the general case as difficult as solving the motion planning problem for all initial conditions.

```
function[dg,ag]=get_goal_value(x);
Xgoal=[dist,dist]';
dg=norm(Xgoal-x);
ag=atan2(Xgoal(2)-x(2),Xgoal(1)-x(1));
```

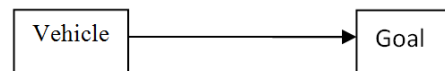


Fig: 2.1 go to goal behaviour

*ii) OBSTACLE AVOIDANCE*

For an autonomous vehicle to succeed at advanced manoeuvres, a solid baseline of Obstacle avoidance is mandatory. Such a ability needs an accurate world model, utilizing maps and sensors, and a robust technique to avoid those obstacles in the model. The avoidance technique used here is a two-tiered approach consisting of a deliberative or far-field model and a reactive or near-field model. The safety for the obstacle avoidance and verify passive safety and passive friendly safety. One of the conceptual difficulties in work on obstacle avoidance which is essential for an autonomous vehicle. Our need is to use hybrid models and verification techniques that illustrate it to be collision-free that usually requires other vehicles and verify the vehicles’ discrete control choices along with its to be

sensible, e. g., not actively try to run into the vehicles when continuous, physical motion (fig:2.2). In summary, our contributions are it is just stopped in a corner. Hybrid models of navigation and obstacle avoidance control assume stringent constraints on the behaviour of obstacles they guarantee. Our need to refrain from doing and passive friendly safety in the presence of stationary and so and allow arbitrary obstacles with arbitrary continuous moving obstacles despite sensor and actuator uncertainty.

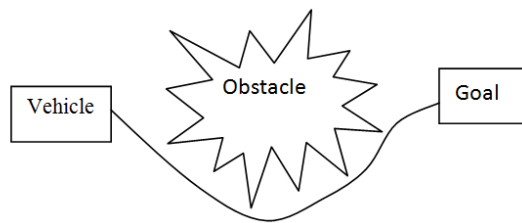


Fig:2.2 obstacle avoidance behavior

```
function [ds,as]=get_obstacle(x,k);
XOB=[a b c d];
if (k==1);
sOB=size(XOB);
ds=1e10; as=0;
for i=1:sOB(1);
dst=min(ds,norm(XOB(i,:) - x'));
if (dst<ds);
ds=dst;
as=atan2(XOB(i,2)-x(2),XOB(i,1)-x(1));
end;
end;
else;
ds=XOB; as=0;
end;
```

The obstacle avoidance is the basic characteristic of an autonomous vehicle. Various kinds of sensors can be employed for obstacle detecting and avoiding behaviour. In order to find the range or relative distance of the leading vehicle from the follower vehicle it is needed to find the range or distance between these two vehicles. Range detection can be achieved by using the ultrasonic sensor. The sensors used in this application should have strong requirements and highly reliable at various circumstances. The entire structure of the system need to be fulfils the mechanical and electrical needs by its performance. The ultrasonic sensor DYP-ME007 can be used for range detection purpose. DYP-ME007 is an IC

that works by sending an ultrasound pulse at around 40 KHz. It then waits and listens for the pulse to echo back, calculating the time taken in microseconds (1 microsecond =  $1.0 \times 10^{-6}$  seconds). You can trigger a pulse as fast as 20 times a second and it can determine objects up to 3 meters away and as near as 3cm. It needs a 5V power supply to run. Obstacle avoidance requires a vehicle can sense or calculate the distance from the objective in front or in the track. A robot or micro controller can perform manoeuvres to avoid obstacles according to the measured signal from those sensors. Therefore it is easy to say that better sensor gives better outputs. The range detection can be achieved by using ultrasonic sensor. It has 4 pins named as vcc, echo, trig, and ground. This sensor can be held at the front end in order to find the distance between the follower vehicle and leader vehicle. The relative distance should be maintained as constant for better performance. To add autonomous behaviour the GPS can be interfaced with the controller. The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather, anywhere on or near the Earth, where there is an unobstructed line of sight to four or more GPS satellites.

## V. UNICYCLE DYNAMICS

A virtual strategy can be used in hybrid autonomous vehicles. Unicycle dynamics provide a simplified design of an autonomous vehicle. The characteristics of unicycle dynamics may vary in accordance with the given inputs. The equations used in the unicycle dynamics are

$$dx/dt= f(x,u)$$

$$X_k=x(k \text{ it}+it)$$

$$X_{k+1}=X_k+it f(X_k,U_k)$$

$$X_1=X_0+it f(X_0,U_0)$$

$$X_2=X_1+it f(X_1,U_1)$$

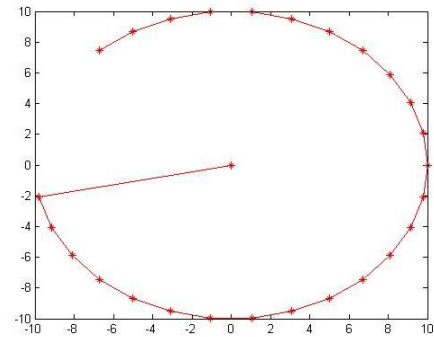
$$X_3=X_2+it f(X_2,U_2)$$

$$X_{k+1}=X_0+(it) v \cos pi$$

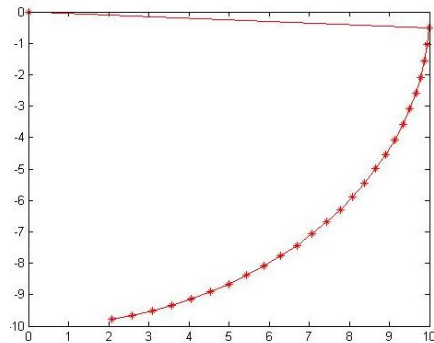
$$Y_{k+1}=Y_0+(it) v \sin pi$$

NO/CONTENT	INPUT V	INPUT W	VEHICLE MOVEMENT
[I] 1	10	0	FORWARD
2 to 5		30,60,90,120	LEFT
6		150 to 360	LEFT+CIRCLE
[II] 7	-10	0	REVERSE
8 to 11		30,60,90,120	RIGHT
12		150 to 360	RIGHT+CIRCLE
[III] 13	10	-0	FORWARD
14 to 17		-30,-60,-90,-120	RIGHT
18		-150 to -360	RIGHT+CIRCLE
[IV] 19	-10	-0	REVERSE
20 to 23		-30,-60,-90,-120	LEFT
24		-150 to -360	RIGHT+CICLE

3. Input V= -10, Input W=120;

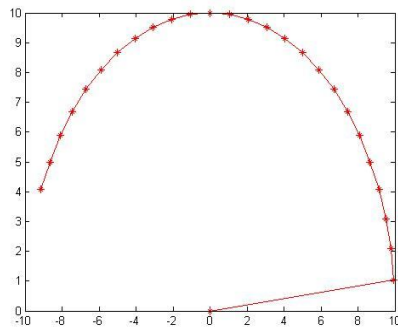


4. Input V=10, Input W=-30;

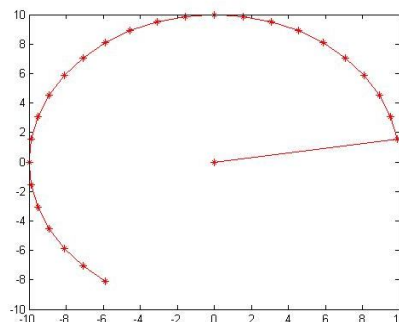


V. SIMULATION RESULTS:

1. Input V=10, Input W=60;



2. Input V=10, Input W=90;



VI.CONCLUSION

In layered architecture the path planning algorithm can be implemented for the appropriate autonomous vehicle. The code level design has been employed in behaviour based control for intelligent vehicles. The Hybrid automated design is to be designed and developed for better performance.

VII.ACKNOWLEDGEMENT

We acknowledge Dr.Seshadhri Srinivasan leader, International Research Centre, Kalasalingam University for his inputs and valuable guidance.

## International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization,

Volume 3, Special Issue 1, February 2014

### International Conference on Engineering Technology and Science-(ICETS'14) On 10<sup>th</sup> & 11<sup>th</sup> February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHANICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

#### REFERENCES

- [1] <http://ivship.weebly.com/>
- [2] Wirasingha, Sanjaka G., and Ali Emadi. "Classification and review of control strategies for plug-in hybrid electric vehicles." *vehicular Technology, IEEE Transactions on* 60, no. 1 (2011): 111-122.
- [3] B. Breimer, "Design of adaptive cruise control model for hybrid systems fault diagnosis", MSc thesis, Dept. of computing and software, McMaster University, 2013.
- [4] Corona, Daniele, and Bart De Schutter. "Comparison of a Linear and a Hybrid Adaptive Cruise Controller for a SMART." In *Decision and Control, 2007 46th IEEE Conference on*, pp. 4779-4784. IEEE, 2007.
- [5] Lygeros, John, Datta N. God bole, and Shankar Sastry. "Verified hybrid controllers for automated vehicles." *Automatic Control, IEEE Transactions on* 43, no. 4 (1998): 522-539.
- [6] Egerstedt, Magnus. "Control of Autonomous Mobile Robots." In *Handbook of Networked and Embedded Control Systems*, pp. 767-778. Birkhäuser Boston, 2005.
- [7] R.A.P.M. van den Bleek, Design of a hybrid adaptive cruise control Stop and Go system, 2013, Technical Report TNO Gmbh, 2012.
- [8] Yimin Gao and Mehrdad Ehsani, "Design and Control Methodology of Plug-in Hybrid Electric Vehicles" IEEE transactions on industrial electronics, vol. 57, no. 2, february 2010 633.
- [9] Dmitri Dolgoy, Sebastian Thrun, Michael Montemerlo, James Diebel "Path planning for autonomous driving in unknown environments" Springer Tracts in advanced Robotics Volume 54, 2009, pp55-64.
- [10] Srinivasan, Seshadhri, and Ramakalyan Ayyagari. "Advanced driver assistance system for AHS over communication links with random packet dropouts." *Mechanical Systems and Signal Processing* (2012).

#### APPENDIX

```
#include "AT89x52.h"
#include "stdio.h"

#define RS P1_0
#define EN P0_7

#define D4 P2_4
#define D5 P2_5
#define D6 P2_6
#define D7 P2_7

#define TRIG P3_3
#define ECHO P3_2

#define MOT0 P1_1
#define MOT1 P1_2
#define MOT2 P1_3
#define MOT3 P1_4

unsigned long timer_count = 0;

#define F_OSC 11059200

void Delay(unsigned int time)
{
    while(time--);
}

void LCD_DAT(unsigned char dat)
{
    unsigned char dat1 = 0;
    RS=1;
    EN=1;
    dat1 = ((dat & 0xF0));
    if(dat1&0x10) D4 = 1;
    else D4 = 0;
    if(dat1&0x20) D5 = 1;
    else D5 = 0;
    if(dat1&0x40) D6 = 1;
    else D6 = 0;
    if(dat1&0x80) D7 = 1;
    else D7 = 0;
    EN=0;
    Delay(0x6FF);

    EN=1;
    dat1 = (dat & 0x0F)<<4;
```

```

    if(dat1&0x10) D4 = 1;
    else D4 = 0;
    if(dat1&0x20) D5 = 1;
    else D5 = 0;
    if(dat1&0x40) D6 = 1;
    else D6 = 0;
    if(dat1&0x80) D7 = 1;
    else D7 = 0;
    EN=0;
    Delay(0x6FF);
}

void LCD_CMD(unsigned char cmd)
{
    unsigned char cmd1 = 0;
    RS=0;
    EN=1;
    cmd1 = (cmd & 0xF0);
    if(cmd1&0x10) D4 = 1;
    else D4 = 0;
    if(cmd1&0x20) D5 = 1;
    else D5 = 0;
    if(cmd1&0x40) D6 = 1;
    else D6 = 0;
    if(cmd1&0x80) D7 = 1;
    else D7 = 0;
    EN=0;
    Delay(0x6FF);

    EN=1;
    cmd1 = (cmd & 0x0F)<<4;
    if(cmd1&0x10) D4 = 1;
    else D4 = 0;
    if(cmd1&0x20) D5 = 1;
    else D5 = 0;
    if(cmd1&0x40) D6 = 1;
    else D6 = 0;
    if(cmd1&0x80) D7 = 1;
    else D7 = 0;
    EN=0;
    Delay(0x6FF);
}

void LCD_Goto(unsigned char row,unsigned char col)
{
    if(row==1) LCD_CMD(0x80+col);
    else if(row==2) LCD_CMD(0xc0+col);
    Delay(0xFF);
}

void LCD_CLR(void)
{
    LCD_CMD(0X01);
    Delay(50);
}

void LCD_INIT(void)
{
    LCD_CMD(0X02);
    Delay(0xFF);
    LCD_CMD(0X28);
    Delay(0xFF);
    LCD_CMD(0X28);
    Delay(0xFF);
    LCD_CMD(0X0E);
    Delay(0xFF);
    LCD_CLR();
    Delay(0xFF);
}

void LCD_Display(unsigned char *ptr,unsigned char
row,unsigned char col)
{
    if(row==1) LCD_CMD(0x01);
    while(*ptr!='\0')
    {
        LCD_Goto(row,col);
        LCD_DAT(*ptr);
        ptr++; col++;
    }
}

void timer0_intr(void) interrupt 1
{
    EA = 0;
    if(TF0)
    {
        TF0 = TR0 = 0;
        ++timer_count;
        TH0 = TL0 = 0;
        TR0 = 1;
    }
    EA = 1;
}

void LCD_Var_Dis(unsigned long t,unsigned char row,
unsigned char col)
{
    LCD_Goto(row,col);
    LCD_DAT((t/1000)+'0');
}

```

Department of CIVIL, CSE, ECE, EEE, MECHANICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

```

t = t - ((t/1000)*1000);
LCD_Goto(row,col+1);
LCD_DAT((t/100)+'0');
t = t - ((t/100)*100);

LCD_Goto(row,col+2);
LCD_DAT((t/10)+'0');
LCD_Goto(row,col+3);
LCD_DAT((t%10)+'0');
}

void main(void)
{
    unsigned char temp = 0, i=0, dat = 0, status = 0;
    unsigned long distance = 0;

    LCD_INIT();
    Delay(0xFFFF);

    LCD_Display("Image Based Visual",1,0);
    LCD_Display("Speech Animation",2,0);

    for(temp=0;temp<4;temp++)
    {
        P2_0 = 0; P2_1 = 1;
        Delay(0xFFFF);
        P2_0 = 1; P2_1 = 0;
        Delay(0xFFFF);
    }
    P2_0 = P2_1 = 1;

    while(1)
    {
        TRIG = 0;

        for(i=0;i<60;i++);

        TMOD = 0x01;
        TH0 = TL0 = 0x00;

        while(ECHO==0);
        EA = ET0 = 1; TR0 = 1;
        if(ECHO)
        {
            while(ECHO);
            TR0 = 0; EA = ET0 = 0;

            distance = (((TH0*256)+TL0))/58;
            if(distance<400)
            {
                LCD_Display("Distance",1,4);
                LCD_Var_Dis(distance,2,4);

                if(distance<30)
                {
                    dat = 1; MOT0 = 1; MOT1 = 1;
                    MOT2 = 1; MOT3 = 1;
                    Delay(0xFFFF);
                }
                if((distance>A)&(distance<B))
                {
                    for(i=0;i<8;i++)
                    {
                        MOT0 = 0; MOT1 = 1; MOT2 = 0; MOT3 = 1;
                        Delay(0xFFF);
                        MOT0 = 1; MOT1 = 1; MOT2 = 1; MOT3 = 1;
                        Delay(0xFFFF);
                    }
                }
                if((distance>C)&(distance<D))
                {
                    for(i=0;i<8;i++)
                    {
                        MOT0 = 0; MOT1 = 1; MOT2 = 0; MOT3 = 1;
                        Delay(0xFFFF);
                        MOT0 = 1; MOT1 = 1; MOT2 = 1; MOT3 = 1;
                        Delay(0xFFFF);
                    }
                }
                temp = P1 & 0xF0;
                if(temp == 0xE0)
                {
                    if(dat)
                    {
                        LCD_Display("Obustacle Detect",1,0);
                        LCD_Display("Unable to Move",2,1);
                    }
                    else
                    {
                        LCD_Display("Moves Forward",1,1);
                        MOT0 = 0; MOT1 = 1; MOT2 = 0; MOT3 = 1;
                        Delay(0xFFFF);
                        MOT0 = 1; MOT1 = 1; MOT2 = 1; MOT3 = 1;
                    }
                }
            }
        }
    }
}

```



**International Journal of Innovative Research in Science, Engineering and Technology***An ISO 3297: 2007 Certified Organization,**Volume 3, Special Issue 1, February 2014***International Conference on Engineering Technology and Science-(ICETS'14)  
On 10<sup>th</sup> & 11<sup>th</sup> February Organized by**

Department of CIVIL, CSE, ECE, EEE, MECHANICAL Engg. and S&amp;H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

```
}  
else if(temp == 0xD0)  
{  
LCD_Display("Moves Reverse",1,1);  
MOT0 = 1; MOT1 = 0; MOT2 = 1; MOT3 = 0;  
Delay(0xFFFF);  
MOT0 = 1; MOT1 = 1; MOT2 = 1; MOT3 = 1;  
}  
else if(temp == 0xB0)  
{  
LCD_Display("Turns Right",1,3);  
MOT0 = 0; MOT1 = 1; MOT2 = 0; MOT3 = 0;  
Delay(0xFFFF);  
MOT0 = 1; MOT1 = 1; MOT2 = 1; MOT3 = 1;  
}  
else if(temp == 0x70)  
{  
LCD_Display("Turns Left",1,3);  
MOT0 = 0; MOT1 = 0; MOT2 = 0; MOT3 = 1;  
Delay(0xFFFF);  
MOT0 = 1; MOT1 = 1; MOT2 = 1; MOT3 = 1;  
}  
else ;  
distance = timer_count = temp = i = dat = 0;  
Delay(0xFFFF);  
}  
}
```