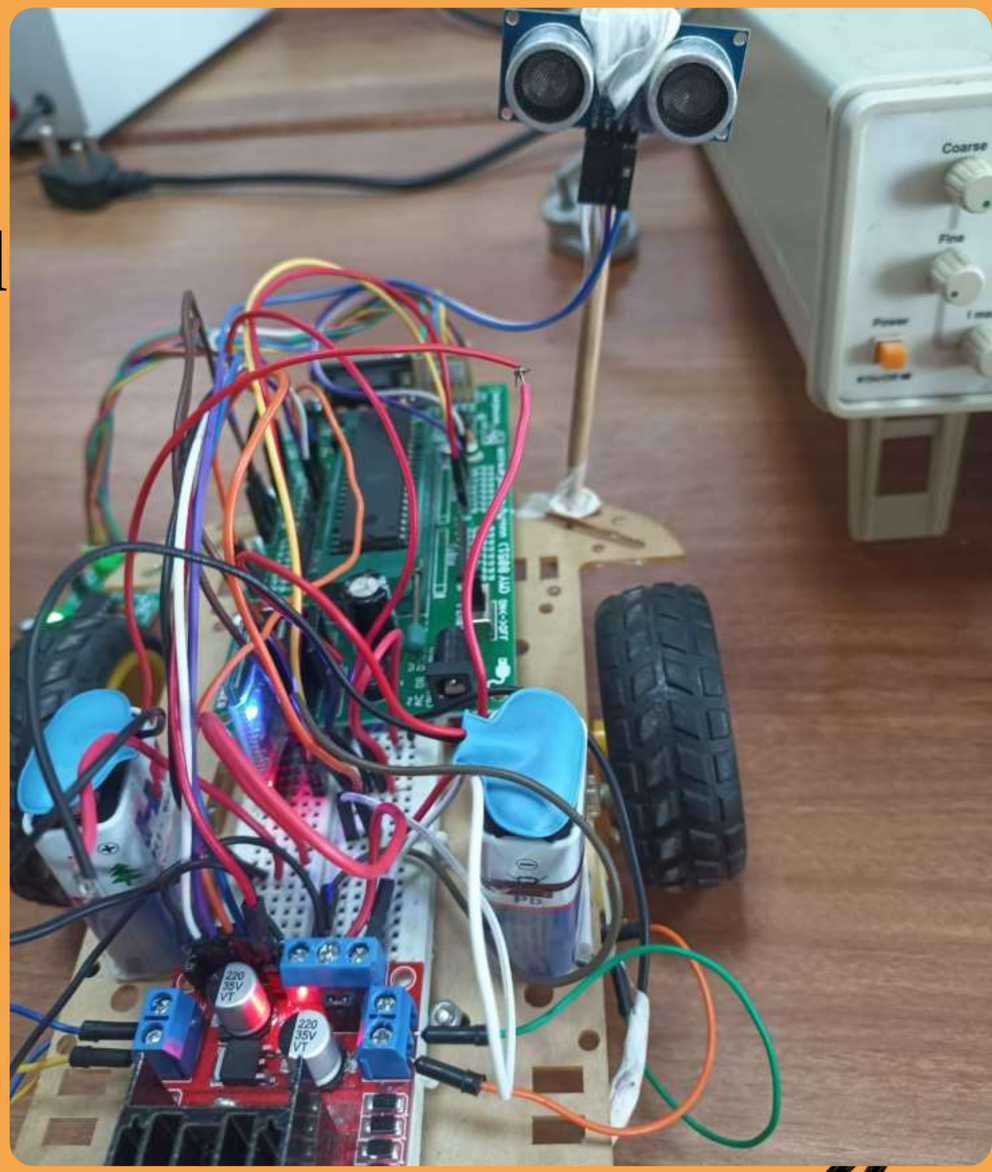
OBSTACLE AVOIDANCE AND BLUETOOTH CONTROLLED CAR

MPMC PROJECT PRESENTATION

INTRODUCTION:

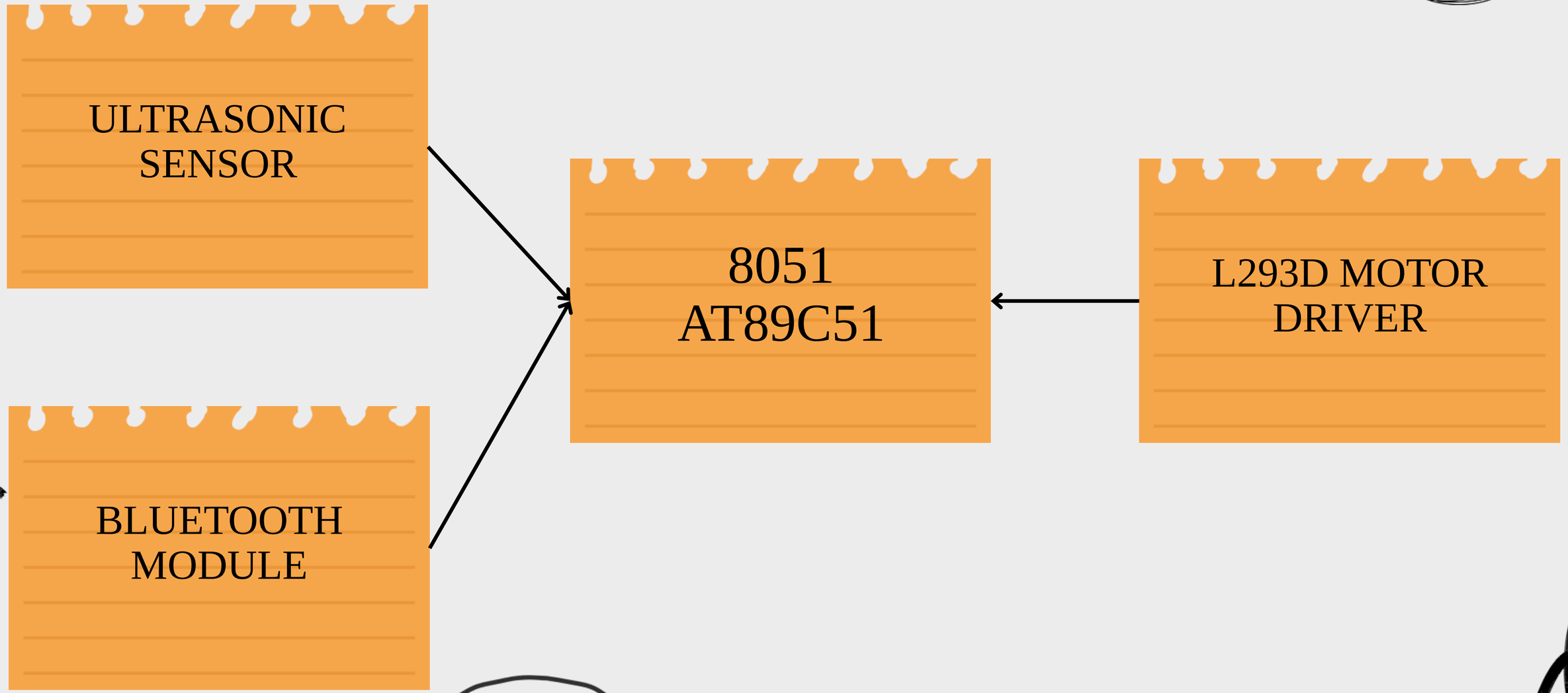
This project aims to design and implement a robot car that can be wirelessly controlled via Bluetooth communication. In addition to remote control functionality, the project aims to enhance the robot's autonomy by incorporating an obstacle detection system. This system will allow the robot to detect obstacles and autonomously navigate around them, ensuring a safe and efficient operation.



COMPONENTS:

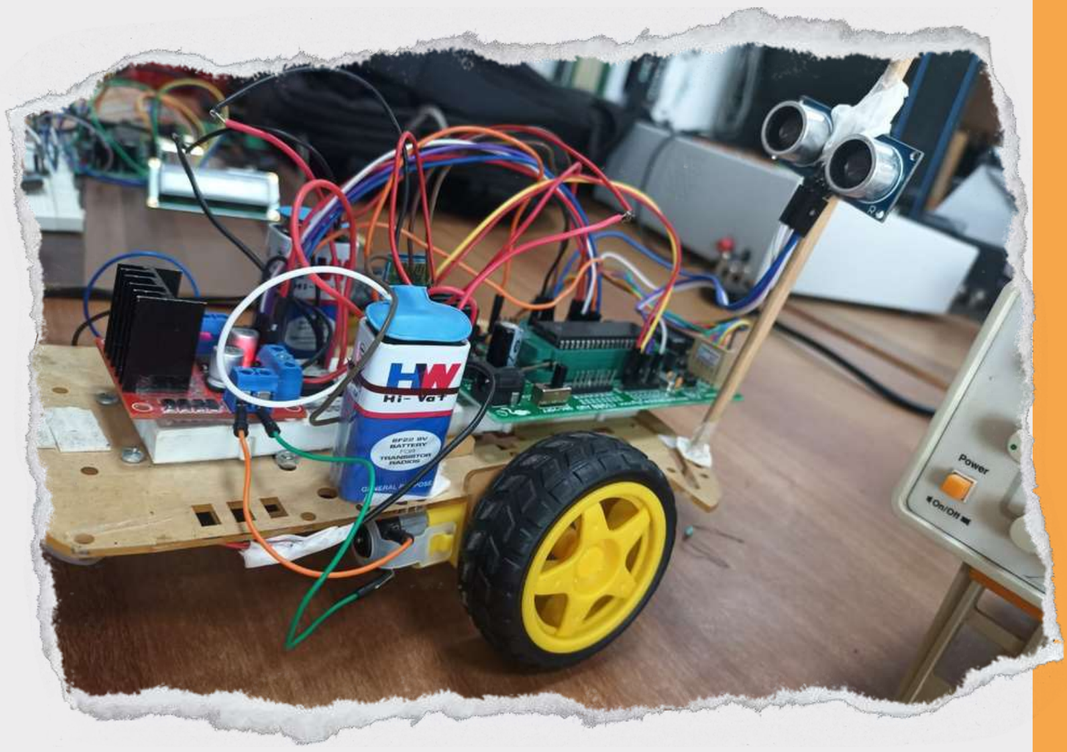
- MICROPROCESSOR 8051
- BLUETOOTH MODULE(HC-O5)
- ULTRASONIC SENSOR
- L293D MOTOR DRIVER
- ROBOT CAR
- BATTERY

BLOCK DIAGRAM



OBJECTIVE

- Implement a robust Bluetooth communication system for remote control of the robot car.
- Integrate sensors for obstacle detection, enabling the car to detect and react to obstacles in its path.
- Develop a user-friendly interface for controlling the car through a Bluetooth-enabled device.
- Test and optimize the system to ensure reliable performance in various environments.



CODE:

```
#include <reg51.h>
#include <intrins.h>
void sendser_char(unsigned char b);
void sendser_str(unsigned char *str);
void delayMicroseconds(unsigned int us);
void measureDistance();
void right();
void left();
void forward();
void backward();
void stop();
// Assuming these are your motor control GPIO pins
sbit m11 = P2^2;
sbit m12 = P2^3;
sbit m21 = P2^4;
sbit m22 = P2^5;

// Assuming these are your sensor trigger and echo pins
sbit trigPin = P3^0;
sbit echoPin = P3^1;
```

CODE:

```
void main() {
  unsigned char x;
   TMOD = 0x20; // Timer 1 mode2 reload mode
   TH1 = 0xFD;  // 9600 baud rate
   SCON = 0x50; // 8-bit data, 1 start bit, 1 stop bit
   TR1 = 1;
  sendser_str("Bluetooth controlled car");
 while (1) {
       while (RI == 0); // F, B, L, R, S
       RI = 0;
       x = SBUF;
       if (x == 'F') {
          forward();
    measureDistance();
       } else if (x == 'B') {
           backward();
     measureDistance();
       } else if (x == 'L') {
    left();
     measureDistance();
       } else if (x == 'R') {
    right();
    measureDistance();
       } else if (x == 'S') {
          stop();
   }
   }
}
```

CODE:

```
void measureDistance(){
  unsigned int time;
  unsigned int distance;

  trigPin=1;
  delayMicroseconds(10);
  trigPin=0;
 while(!echoPin){

  TL1=0;
  TH1=0;
  while(echoPin){
   if(TF1){
    break;
   }
  }
  time=TL1 |(TH1 << 8);

  distance=(time*343)/2000;

  sendser_str("Distance:");
  sendser_char(distance / 100 + '0');
  sendser_char((distance/10)%10+'0');
  sendser_str("cm\r\n");

  if (distance < 101) {
        sendser_str("Turning Right!\r\n");
         right();
    }

  delayMicroseconds(5000);
 }
}
```

```
void forward(){
          m11 = 1;
           m12 = 0;
           m21 = 1;
           m22 = 0;
}
void backward(){
          m11 = 0;
           m12 = 1;
           m21 = 0;
           m22 = 1;
}
void right(){
          m11 = 1;
           m12 = 0;
           m21 = 0;
           m22 = 1;
}
void left(){
          m11 = 0;
           m12 = 1;
           m21 = 1;
           m22 = 0;
}
void stop(){
          m11 = 1;
           m12 = 1;
           m21 = 1;
           m22 = 1;
}
```

```
void delayMicroseconds(unsigned int us) {
unsigned int i, j;
for (i = 0; i < us; i++) {
for (j = 0; j < 3; j++) {
// Adjust this loop for the required delay
}
}
}
```

```
void sendser_char(unsigned char b) {
SBUF = b;
while (TI == 0);
TI = 0;
}

void sendser_str(unsigned char *str) {
while (*str) {
sendser_char(*str++);
}
}
```

CODE (ULTRASONIC CODE IN ASSEMBLY)

```
trig  EQU  P3.1   ;
echo  EQU  P3.0   ;
enable equ p2.2
rs equ p2.0
rw equ p2.1
LCD_dat equ p1
ORG  0000
setb echo
clr trig
mov tmod, #02h
mov th0, #202
acall LCD_init
acall delay_2s
acall LCD_clear
loop1:
       ACALL get_level
       ACALL CONVERT
       acall cursr_home
  ACALL display
       SJMP Loop1

LCD_init: mov dptr, #syntax
  clr rs
  clr rw
loop: clr a
  movc a, @a+dptr
  jz LCD_logo
  setb enable
  mov LCD_dat, a
  clr enable
  acall delay1ms
  inc dptr
  sjmp loop

syntax: db 38h,0fh,01h,10h,00h

LCD_logo: mov dptr, #syntax1
  setb rs
  clr rw
loop4: clr a
  movc a, @a+dptr
  jz new_command
  setb enable
  mov LCD_dat, a
  clr enable
  acall delay1ms
  inc dptr
new_command: mov dptr, #syntax2
  clr rs
  clr rw
loop5: clr a
  movc a, @a+dptr
  jz LCD_logo_2
  setb enable
  mov LCD_dat, a
  clr enable
  acall delay1ms
  inc dptr
  sjmp loop5
syntax2: db 0c0h,14h,14h,14h,00h
LCD_logo_2: mov dptr, #syntax3
  setb rs
  clr rw
loop6: clr a
  movc a, @a+dptr
  jz return
  setb enable
  mov LCD_dat, a
  clr enable
  acall delay1ms
  inc dptr
  sjmp loop6
syntax3: db ".com",0
return:ret
cursr_home:
clr rs
setb enable
mov LCD_dat,#80h
clr enable
acall delay10ms
setb enable
mov LCD_dat,#0Ch
clr enable
ret
LCD_clear:
clr rs
setb enable
mov LCD_dat,#01h
clr enable
acall delay10ms
ret
get_level:
  clr a
  setb trig
  acall delay_10us
  clr trig
wait5:  jnb echo, wait5
  setb tr0
wait6:  jnb tf0, wait6
  inc A
  clr tf0
  jz return

  jb echo, wait6
  clr tr0
  ret
delay_10us:
  mov r7, #18
stay:  djnz r7, stay
  ret
CONVERT:
      MOV    B,#10
      DIV    AB
      MOV    41,B   ; SAVE LOW(ONES) DIGIT IN 41 RAM ADDRESS
      MOV    B,#10
      DIV    AB
      MOV  42,B    ; save tenth place digit in 42 RAM ADDRESS
 MOV  43,A    ; SAVE HUNDREDTH PLACE DIGIT IN 43 RAM ADDRESS
 CALL LOKUP
 MOV  43,A
 MOV  A,42
 CALL LOKUP
 MOV  42,A
 MOV  A,41
 CALL LOKUP
 MOV  41,A
 RET
LOKUP:
      CJNE A,#00H,ONE
      MOV    A,#'0'
      RET
ONE:   CJNE A,#01H,TWO
      MOV    A,#'1'
      RET
TWO:   CJNE A,#02H,THREE
      MOV    A,#'2'
      RET
THREE:  CJNE A,#03H,FOUR
      MOV    A,#'3'
      RET
FOUR:  CJNE A,#04H,FIVE
      MOV    A,#'4'
      RET
FIVE:  CJNE A,#05H,SIX
      MOV    A,#'5'
      RET
SIX:   CJNE A,#06H,SEVEN
      MOV    A,#'6'
      RET
SEVEN:  CJNE A,#07H,EIGHT
      MOV    A,#'7'
      RET
EIGHT:  CJNE A,#08H,NINE
      MOV    A,#'8'
      RET
display:
      clr rw
 setb rs
 acall delay1ms

 SETB   enable
   MOV    LCD_dat,#' '
      clr enable
 acall delay1ms

 SETB   enable
   MOV    LCD_dat,43
      clr enable
 acall delay1ms

 SETB   enable
   MOV    LCD_dat,42
      clr enable
 acall delay1ms

 SETB   enable
   MOV    LCD_dat,41
      clr enable
 acall delay1ms

 SETB   enable
      MOV    LCD_dat,#'c'
      clr enable
 acall delay1ms

 SETB   enable
      MOV    LCD_dat,#'m'
      clr enable
 acall delay1ms

      RET
delay10ms:  MOV   R3,#1
    MOV   R2,#1
      MOV   R1,#19
TT1:   DJNZ  R1,TT1
      DJNZ  R2,TT1
      DJNZ  R3,TT1
      RET
delay1ms:  MOV   R2,#04
      MOV   R1,#18
TT2:   DJNZ  R1,TT2
      DJNZ  R2,TT2
      RET
delay_2s:MOV   R3,#50
    MOV   R2,#10
      MOV   R1,#250
TT3:   DJNZ  R1,TT1
      DJNZ  R2,TT1
      DJNZ  R3,TT3
      RET

    END
```

***we have wriiten code inluding lcd

CHALLENGES FACED:

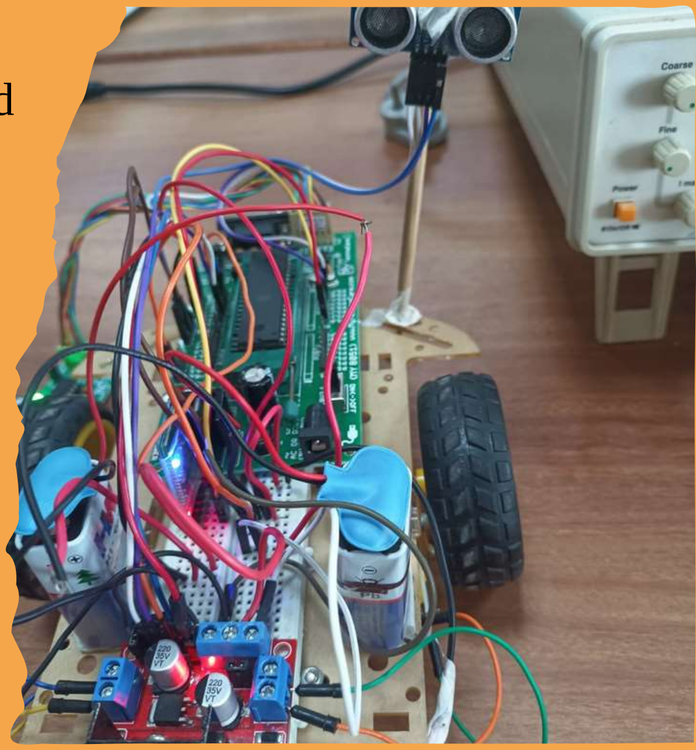
- Limited Input/Output (I/O) Pins on Microcontroller for connecting lcd display.
- Balancing power consumption to ensure an adequate operational lifespan of the robot car can be challenging, especially with the added load of Bluetooth communication and sensor usage.
- Calibrating and ensuring the accuracy of obstacle detection sensors can be complex. Factors such as varying ambient light conditions and the type of obstacles encountered may affect sensor readings.
- Achieving real-time responsiveness in both Bluetooth control and obstacle detection is crucial for the effective operation of the robot car.

OUTPUT VIDEO:

https://drive.google.com/drive/folders/1qN-3zVgb1taeHuDDqLgxZnDg3XY8s2o-

CONCLUSIONS

- The Bluetooth-controlled car with obstacle detection represents a synergy of modern technologies, offering a glimpse into the future of robotics and automation.
  - The integration of Bluetooth control and obstacle detection adds practicality and real-world applicability, making it an engaging and educational project for enthusiasts in the field of embedded systems and robotics.
- The project's key features, including Bluetooth control for remote operation and the implementation of an obstacle detection and avoidance system, address real-world challenges in robotics, making the robot car more user-friendly and safe.



THANK YOU