A Practical Activity Report

Submitted for

ENGINEERING DESIGN-II (UTA014)

by:

Name of student	**Roll number**
Vivek Arora	101715178
Yash Gupta	101715179
Yash Mittal	101715180
Yashik Anand	101715181
Yogesh Goyal	101715183

Submitted to:

Mrs. Navneet Kaur

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA-147004, PUNJAB
INDIA

Jan-June 2019

INDEX

CERTIFICATE

This is to certify that the project entitled "**ENGINEERING DESIGN (UTA014) Buggy car project**" embodies the group effort of the candidates along with guidance from Mrs. Navneet Kaur and has been submitted for the fulfilment of the project report of Engineering Design Project – II.

Signature of Faculty

Experiment:1

Objective: To demonstrate the use of ultrasonic sensor by integrating line follower robocar with obstacle avoidance capability

Theory:

An **Ultrasonic Sensor** is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

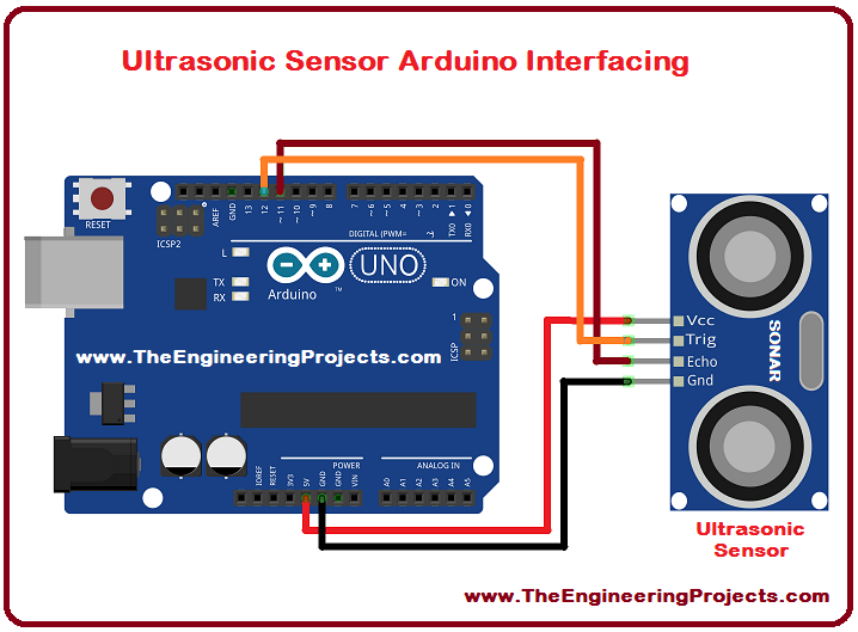

Figure 1/Ultrasonic Sensor

Hardware Used: Arduino/Genuino board, connecting wires, Robocar, Receiver Circuit, Ultrasonic sensor.

Software Used: Arduino software (IDE)

CODE:

```
#include <NewPing.h>
#define TRIGGER_PIN 13
#define ECHO_PIN 12
#define MAX_DISTANCE 200
NewPingsonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
void setup() {
Serial.begin(9600);
}
void loop() {
delay(50);
unsignedintuS = sonar.ping();
pinMode(ECHO_PIN,OUTPUT);
digitalWrite(ECHO_PIN,LOW);
pinMode(ECHO_PIN,INPUT);
Serial.print("Ping: ");
Serial.print(uS / US_ROUNDTRIP_CM);
Serial.println("cm");
Serial.print("uS: ");
Serial.println(uS);
Serial.print("US_ROUNDTRIP_CM: ");
Serial.println(US_ROUNDTRIP_CM);
}
```

Result Analysis:

We attached the ultrasonic sensor to the buggy and moved the buggy. We tested the code of this sensor by placing obstacles in front of the robocar, and it worked correctly.

Signature of Faculty member

Experiment: 2

Objective: Write a program to read the pulse width of gantry transmitter and trigger stop_buggy function by detecting individual gantry.

Hardware Used: Arduino/Genuino board, connecting wires, Robocar, Receiver Circuit, Ultrasonic sensor.

Software Used: Arduino software (IDE)

CODE:

```
int t1=A0;
int t2=A2;
int pin5=5;
int pin6=6;
int pin7=8;

int pin8=7;
intirPin=4;
int flag=0;
unsigned long d=0;
staticintgantryCounter=0;
static long StartTime=0;
static long CurrentTime = 0;
unsigned long ElapsedTime = 0;
static long StartTimeG=millis();;
static long CurrentTimeG = 0;
unsigned long ElapsedTimeG = 0;
longpreviousMillisU = millis();
longintervalU = 500;
#include <NewPing.h>
#define TRIGGER_PIN 13
#define ECHO_PIN 12
#define MAX_DISTANCE 200
NewPingsonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
void setup() {
pinMode(pin5,OUTPUT);
pinMode(pin6,OUTPUT);
pinMode(pin7,OUTPUT);
pinMode(pin8,OUTPUT);
pinMode(t1,INPUT);
pinMode(t2,INPUT);
Serial.begin(9600);
Serial.print("+++");  // Enter xbee AT command mode, NB no carriage return here
delay(1500);          // Guard time
Serial.println("ATID 3333, CH C, CN");
}
```

```
void loop() {
if(flag==0)
{
if (Serial.available() > 0)
{
char s = Serial.read();
switch (s) {
case 'G':
     {
flag=1;
        }
}
}
}
unsigned long currentMillisU = millis();

if(currentMillisU - previousMillisU>intervalU)
      {

previousMillisU = currentMillisU;
detectObstacle();
      }

if (flag==1)
{
gantry();


}

if (flag==3)
  {
    //gantryParking();
CurrentTimeG=millis();
ElapsedTimeG = CurrentTimeG-StartTimeG;

if(ElapsedTimeG<1500)
    {
flag=3;
leftBlind();
    }
if(ElapsedTimeG>1500 &&ElapsedTimeG<3500)
    {
flag=3;
normalLineFollow();
    }
if(ElapsedTimeG>35
00)
```

```
    {
stopBuggy();
Serial.print("Buggy:1 Parked");
Serial.println(ElapsedTimeG);
delay(2000);
flag=-1;
    }

   }
}

void gantry()
{
int r1=digitalRead(t1);
int r2=digitalRead(t2);
if(r1==LOW&&r2==LOW)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);

}
if(r1==LOW&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,LOW);
digitalWrite(pin8,LOW);

} if(r1==HIGH&&r2==LOW)
{
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
 }
if(r1==HIGH&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);

}
voidstopBuggy()
  {
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,LOW);
```

```
digitalWrite(pin8,LOW);
  }
voidnormalLineFollow()
{
int r1=digitalRead(t1);
int r2=digitalRead(t2);
if(r1==LOW&&r2==LOW)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);

}
if(r1==HIGH&&r2==LOW)
{
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);

} if(r1==LOW&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,LOW);
digitalWrite(pin8,LOW);
 }
if(r1==HIGH&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);

}
}
voidleftBlind()
{

int r2=digitalRead(t2);

if(r2==LOW)
{
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);

}
```

```
if(r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);

}
```

Result Analysis:

From this experiment we understood the working of the gantry circuit along with the receiver circuit, which sends the pulse to the computer via receiver circuit which detects the pulse via MBD311D diode and feeds it to the computer , in the form of a Square Wave Function from which the time of the high pulse is obtained using pulseIn function along with Serial.begin(9600) (which gives communication baud rate). And from this time along with its condition statements we operate the Working of the Buggy.

Signature of Faculty member

Experiment: 3

Objective: Write a program to control buggy into full supervisory mode using serial communication.

Hardware Used: Arduino/Genuino board, connecting wires, Robocar, Receiver Circuit, Ultrasonic sensor, Zigbee Module**.**

Software Used: Arduino software (IDE), XCTU

CODE:

```
void setup()
{
Serial.begin(9600);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode (7, OUTPUT);
pinMode(8, OUTPUT);
}
void forward()
{
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
 }
void Backward()
{
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
digitalWrite(8,LOW);
```

```
}
void Anti_Rotation()
{
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
digitalWrite(8,LOW);
}
void Clock_Rotation()
{
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
}
void stops()
{
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
}
void loop()
{
if(Serial.available()>0)
{
s=Serialread();
```

```
if(s=='F')
{ forward();
}
else if(s=='B')
{ Backward();
}
else if(s=='R')
{
Clock_Rotation();
}
else if(s=='L')
{
Anti_Rotation();
}
}
}
```

Result Analysis:

We established a connection between the coordinator which will be attached to the computer which will control the buggy and the router which will be attached to the buggy**.**

Signature of Faculty member

Experiment: 4

Objective: Write a Program to demonstrate full bronze challenge.

Hardware Used: Arduino/Genuino board, connecting wires, Robocar, Receiver Circuit, Ultrasonic sensor, Zigbee Module**.**

Software Used: Arduino software (IDE), XCTU, Microsoft Visual Studio

CODE:

```
int t1=A0;
int t2=A2;
int pin5=5;
int pin6=6;
int pin7=8;

int pin8=7;
intirPin=4;
int flag=0;
unsigned long d=0;
staticintgantryCounter=0;
static long StartTime=0;
static long CurrentTime = 0;
unsigned long ElapsedTime = 0;
static long StartTimeG=millis();;
static long CurrentTimeG = 0;
unsigned long ElapsedTimeG = 0;
longpreviousMillisU = millis();
longintervalU = 500;
#include <NewPing.h>
#define TRIGGER_PIN 13
#define ECHO_PIN 12
#define MAX_DISTANCE 200
```

```
NewPingsonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
void setup() {
pinMode(pin5,OUTPUT);
pinMode(pin6,OUTPUT);
pinMode(pin7,OUTPUT);
pinMode(pin8,OUTPUT);
pinMode(t1,INPUT);
pinMode(t2,INPUT);
Serial.begin(9600);
Serial.print("+++");  // Enter xbee AT command mode, NB no carriage return here
delay(1500);          // Guard time
Serial.println("ATID 3333, CH C, CN");
}

void loop() {
if(flag==0)
{
if (Serial.available() > 0)
{
char s = Serial.read();
switch (s) {
case 'G':
     {
flag=1;
        }
}
}
}
unsigned long currentMillisU = millis();
```

```
if(currentMillisU - previousMillisU>intervalU)
      {
previousMillisU = currentMillisU;
detectObstacle();
      }

if (flag==1)
{
gantry();
}
if (flag==3)
  {
    //gantryParking();
CurrentTimeG=millis();
ElapsedTimeG = CurrentTimeG-StartTimeG;

if(ElapsedTimeG<1500)
    {
flag=3;
leftBlind();
    }
if(ElapsedTimeG>1500 &&ElapsedTimeG<3500)
    {
flag=3;
normalLineFollow();
    }
if(ElapsedTimeG>35
00)
    {
stopBuggy();
```

```
Serial.print("Buggy:1 Parked");
Serial.println(ElapsedTimeG);
delay(2000);
flag=-1;
    }
  }
}
void gantry()
{
int r1=digitalRead(t1);
int r2=digitalRead(t2);
if(r1==LOW&&r2==LOW)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
}
if(r1==LOW&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,LOW);
digitalWrite(pin8,LOW);
} if(r1==HIGH&&r2==LOW)
{
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
```

```
digitalWrite(pin8,LOW);
 }
if(r1==HIGH&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
}
if (digitalRead(irPin)==HIGH)
{
StartTime = millis();
d = pulseIn(irPin,HIGH);
if(d > 500 and d < 1500)
        {
           //Serial.println(d);
Serial.println("Gantry: 1");
stopBuggy();
delay(1000);
        }
else if (d> 1500 and d < 2500)
        {
          //  Serial.println(d);
Serial.println("Gantry: 2");
stopBuggy();
gantryCounter=gantryCounter+1;
Serial.print("The gantry Counter is: ");
Serial.println(gantryCounter);
stopBuggy();
```

```
delay(1000);
        }
else if (d > 2500 and d < 3500)
        {
          // Serial.println(d);
Serial.println("Gantry: 3");
delay(1000);
   }
else {
        //Serial.println(d);
Serial.println("Gantry: Unknown");
}
if (gantryCounter>=2)
{
StartTimeG = millis();
flag=3;
}
else
{
gantry();
}
}
}
voidstopBuggy()
  {
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,LOW);
digitalWrite(pin8,LOW);
```

```
  }
voidnormalLineFollow()
{
int r1=digitalRead(t1);
int r2=digitalRead(t2);
if(r1==LOW&&r2==LOW)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
}
if(r1==HIGH&&r2==LOW)
{
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
} if(r1==LOW&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,LOW);
digitalWrite(pin8,LOW);
 }
if(r1==HIGH&&r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
```

```
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
}
}
voidleftBlind()
{
int r2=digitalRead(t2);
if(r2==LOW)
{
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
}
if(r2==HIGH)
{
digitalWrite(pin5,HIGH);
digitalWrite(pin6,LOW);
digitalWrite(pin7,HIGH);
digitalWrite(pin8,LOW);
}
}
voiddetectObstacle()
{
delay(50);
unsignedintdistanceCm;
distanceCm = sonar.ping_cm();
pinMode(ECHO_PIN,OUTPUT);
digitalWrite(ECHO_PIN,LOW);
```

```
pinMode(ECHO_PIN,INPUT);
//Serial.print("Ping: ");
//Serial.println(distanceCm);
//Serial.println("cm");
if((distanceCm<15) && (distanceCm>0))
{
stopBuggy();
delay(1000);
}
}
```

Result Analysis:

We completed the Bronze challenge using the above code and the circuits designed in the ECED department.

Signature of Faculty member