A PRACTICAL ACTIVITY REPORT SUBMITTED FOR ENGINEERING DESIGN PROJECT-II (UTA014)

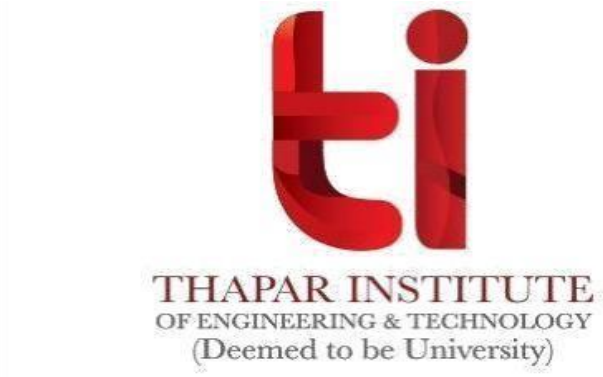
SUBMITTED BY

Arkajyoti Basak

[101808212]

SUBMITTED TO

Sandeep Saharan



CSE DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY

PATIALA-147004, PUNJAB

INDIA

July 2019 – December 2019

INDEX

Experiment: 1

Objective:

To demonstrate the use of ultrasonic sensor by integrating line follower robo car with obstacle avoidance capability.

Hardware Used:

Arduino UNO, Ultrasonic sensor

Software Used:

Arduino IDE

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.

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to- use package. It comes complete with ultrasonic transmitter and receiver module. Ultrasound has several characteristics which make it so useful. Firstly, it is inaudible to humans and therefore undetectable by the user. Secondly, ultrasound waves can be produced with high directivity. Thirdly, they are a compression vibration of matter (usually air). Finally, they have a lower propagation speed than light or radio waves.

Code:

```
#include<NewPing.h>
int echo=12;
int trig =13;
int m=0;
NewPing sonar(trig,echo,maxdist); int
h1,h2;

void setup()
{
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(4,INPUT);
pinMode(A0,INPUT);
pinMode(A1,INPUT);
pinMode(echo,INPUT);
pinMode(trig,OUTPUT);
Serial.begin(9600);
}

void loop()
{
   h1=digitalRead(A0);
   h2=digitalRead(A1);
   m=sonar.ping_cm();

   if(m>0 && m<20)
   {
   stopBuggy();
   delay(2000);
   }

   else
   {
      if(h1==1 && h2==1)
      {
         forward();
      }
      else if(h1==0 && h2==0)
      {
         forward();
      }
      else if(h1==0 && h2==1)
      {
         left();
      }
      else if(h1==1 && h2==0)
      {
         right();
      }
}
}

void stopBuggy()
{
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,LOW);
}

void forward()
{
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
}

void right()
{
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,LOW);
digitalWrite(8,HIGH);
}

void left()
{
digitalWrite(5,HIGH);
digitalWrite(6,LOW);
```

```
    digitalWrite(7,LOW);
    digitalWrite(8,LOW);
    }
}
```

Pictorial Representation:

Fig1: Ultrasonic sensors

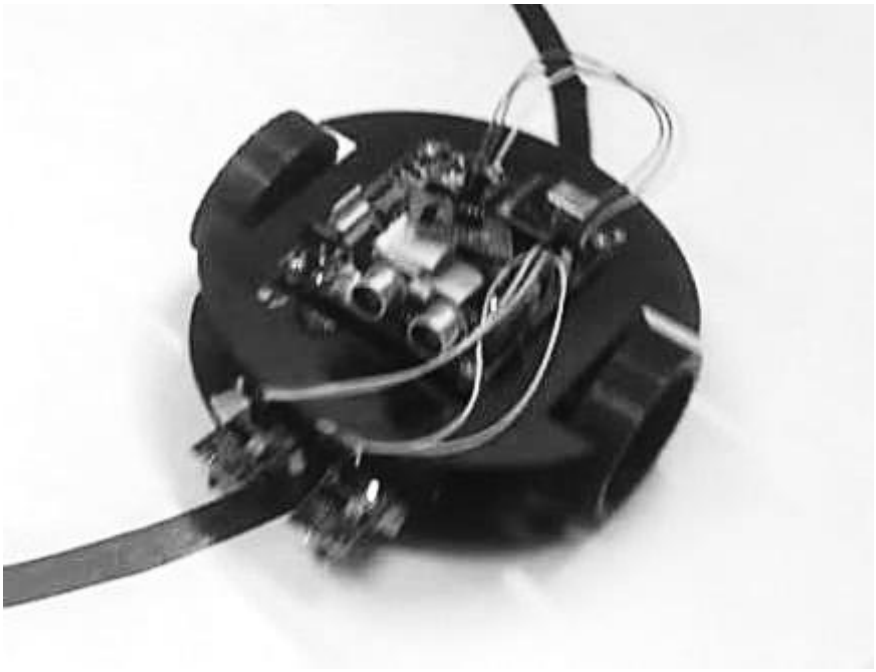


Fig2: Ultrasonic on Buggy

Analysis:

The ultrasonic sensor is connected to the buggy and we moved the buggy. By uploading the code in the buggy, we tested this sensor by placing obstacles in front of the robot car and it worked correctly .

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 Board, Connecting Wires, Robot Car, Receiver Circuit, Ultrasonic Sensor.

Software Used:

Arduino Software (IDE)

Theory:

The gantries are equipped with three pulses of different bandwidth from the transmitter circuit. The range of pulses can be changed by programming the 5 reprogrammable pins of ATtiny85 microcontroller used in the construction of transmitter circuit. The receiver circuit is connected to buggy and programmed to do operations like stop buggy when it passes through the suitable gantry and receives the corresponding pulse. In this way buggy stops after it passes the gantry receiving the pulse from the transmitter connected to the gantry and if the pulse received is in the range according to the code then the required function is performed by Buggy.

Code:

```
int t1=A0;
int t2=A2;
int pin5=5;
int pin6=6;
int pin7=8;
int pin8=7;
int pin=4;
int flag=0;
unsigned long d=0;
static ntgantryCounter=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;
long previousMillisU = mil-
lis(); long intervalU = 500;

#include <NewPing.h>
#define TRIGGER_PIN 13
#define ECHO_PIN 12
#define MAX_DISTANCE 200

NewPing sonar(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("+++");
   delay(1500);
   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 = currentMil- lisU;
detectObstacle();
}
if (flag==1)
{
gantry();
}
if (flag==3)
{
   CurrentTimeG=millis();
   ElapsedTimeG = CurrentTimeG-
   StartTimeG;
   if(ElapsedTimeG<1500)
   {
      flag=3;
      leftBlind();
      }
      if(ElapsedTimeG>1500
&&ElapsedTimeG<3500)
      {
         flag=3; normalLine-
         Follow();
      }
      if(ElapsedTimeG>3500)
      {
         stopBuggy(); Seri-
         al.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);
   }

void stopBuggy()
{
digitalWrite(pin5,LOW);
digitalWrite(pin6,LOW);
digitalWrite(pin7,LOW);
digitalWrite(pin8,LOW);
}

void detectObstacle()
{
   delay(50);
   unsigned int distanceCm; dis-
   tanceCm = 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);
   }
}
```

Pictorial Representation

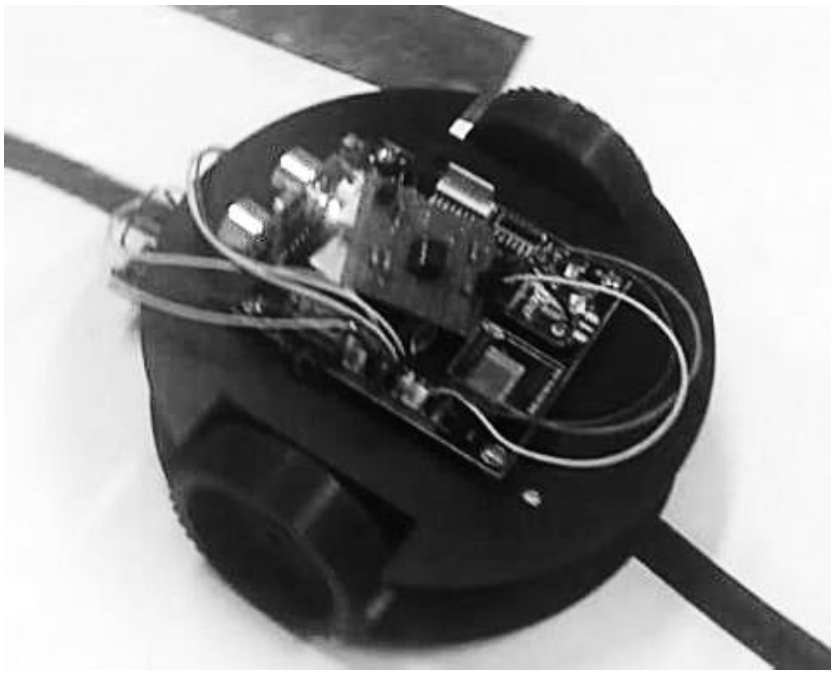


Fig3: Buggy with receiver circuit

Result Analysis:

From this experiment, we get to understand the working of the gantry circuit along receiver circuit. The pulse sends to the computer via receiver circuit which detects the pulse in the form of a Square Wave Function. The time of the high pulse is obtained using pulseIn function along with Serial.begin(9600).

Experiment: 3

Objective:

Write configuration program to demonstrate Xbee module communication between two PC's using XCTU.

Hardware Used:

Arduino Board, Connecting Wires, Robot Car, Zigbee module.

Software Used:

Arduino Software (IDE), XCTU.

Theory:

Zigbee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, Zigbee is a lowpower, low data rate, and close proximity (i.e., personal area) wireless ad hoc network.

The Xbee is the brand name a wireless transceiver device introduced by the Digi international which works on the ZigBee protocol and can form PAN networks. They have an approximate range of 10 to 100 meters and are used in industries, scientific fields, medical fields etc. The Xbee module even though uses complex packet data based Zigbee protocol for communicating with each other; But also they can communicate with other devices using simplest serial communication protocol and hence they are widely used in microcontroller base boards**.**

Code:

```
void setup()
{
   pinMode(5,OUTPUT);
   pinMode(6,OUTPUT);
   pinMode(7,OUTPUT);
   pinMode(8,OUTPUT);
   Serial.begin(9600);
}

void forward()
{
   digitalWrite(5,HIGH);
   digitalWrite(6,LOW);
   digitalWrite(7,LOW);
   digitalWrite(8,HIGH);
}

void loop()
{
   if(Serial.available()>0)
   {
      Char s = Serail.read();
      if(S== ‘G’)
      {
         forward();
      }

      else
      {
         digitalWrite(5,LOW);
         digitalWrite(6,LOW);
         digitalWrite(7,LOW);
         digitalWrite(8,LOW);
      }
      delay(300);
gantryCounter=gantryCounter+1;
   }
   else
   {
      Serial.println("Gantry: Unknown");
      Serial.print("The gantry Counter is:
");
      Serial.println(gantryCounter);
      if (gantryCounter>=2)
      {
         flag=3;
      }
      else
      {
         gantry();
      }
   }
}

void stopBuggy()
{
   digitalWrite(pin5,LOW);
   digitalWrite(pin6,LOW);
   digitalWrite(pin7,LOW);
   digitalWrite(pin8,LOW);
}

void normalLineFollow()
{
   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);
    }
}

void leftBlind()
{
   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);
   }
}

void detectObstacle()
{
   delay(50);
   unsigned int distanceCm; 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);
   }
}
```

Pictorial Representation:

Fig4: Xbee

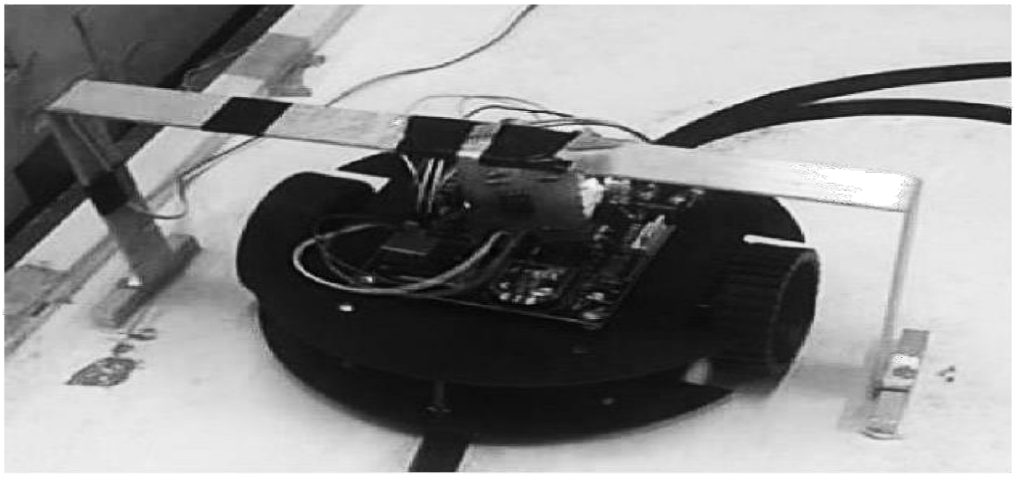


Fig5: Xbee on Buggy

Result Analysis:

From this experiment, we get to understand the working of the Xbee. We have connected the Xbee to buggy and done its configuration on XCTU software. The coordinator will be attached to the computer which will control the buggy and the router will be attached to the buggy**.** Then after doing the configurations, we have controlled the movement of buggy from the computer using Xbee communication.

Experiment: 4

Objective:

Write a Program to demonstrate full bronze challenge.

Hardware Used:

Arduino Board, Connecting Wires, Robot Car, Receiver Circuit, Ultrasonic Sensor, Zigbee Module**.**

Software Used:

Arduino software (IDE), XCTU.

Theory:

Bronze Challenge: Single buggy capable of following main track twice in an clockwise direction under full supervisory control. Buggy must be capable of detecting an obstacle whilst following the track, coming to a halt if it does. The buggy must safely park in the parking bay. It prints the state of the track and buggy at each gantry stop. No external end- user manual control input is permitted once the initial start is signalled.

Code:

```
int t1=A0;
int t2=A2;
int pin5=5;
int pin6=6;
int pin7=8;
int pin8=7;
int irPin=4;
int flag=0;
int inside = 0;
unsigned long d=0;
static int gantryCounter=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;
long previousMillisU = millis();
long intervalU = 500;

#include <NewPing.h>
#define TRIGGER_PIN 13
#define ECHO_PIN 12
#define MAX_DISTANCE 200

NewPing sonar(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 'K':
            {
               flag=1;
            }
         }
      }
   }
   unsigned long currentMillisU = millis();
   if(cur- rentMillisU - previousMillisU >
intervalU)
   {
      previousMillisU = currentMillisU;
detectObstacle();
   }
   if (flag==1)
   {
      gantry();
   }
   if (flag==3)
   {
      int r1=digitalRead(t1);
      int r2=digitalRead(t2);
      if (r1 == LOW && r2 == LOW)
      {
         if (inside == 0)
         {
            digitalWrite(pin5,LOW);
            digitalWrite(pin6,LOW);
```

```
            digitalWrite(pin7,HIGH);
            digitalWrite(pin8,LOW);
            insi- de = 1;
            delay(200);
          }
          else
          {
            stopBuggy();
            delay(100000);
          }
        }
        else
        {
          normalLineFollow();
        }
    }
}

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);
      if(r1==HIGH&&r2==HIGH)
      {
        digitalWrite(pin5,HIGH);
        digitalWrite(pin6,LOW);
        digitalWrite(pin7,HIGH);
        digitalWrite(pin8,LOW);
      }
      delay(300);
      gantryCounter=gantryCounter+1;
```

```
        }
        else if (d>1500 and d< 2700)
        {
           Serial.println(d);
           Serial.println("Gantry: 2");
           stopBuggy(); delay(1000);
           if(r1==HIGH&&r2==HIGH)
           {
              digitalWrite(pin5,HIGH);
              digitalWrite(pin6,LOW);
              digitalWrite(pin7,HIGH);
              digitalWrite(pin8,LOW);
           }

           delay(300);
           gantryCounter=gantryCounter+1;
        }

        else if (d>2500 and d<3500)
        {
           Serial.println(d);
           stopBuggy();
           Serial.println("Gantry: 3");
           delay(1000);
           if(r1==HIGH&&r2==HIGH )
           {
              digitalWrite(pin5,HIGH);
              digitalWrite(pin6,LOW);
              digitalWrite(pin7,HIGH);
           }
        }
     }
}
```

Pictorial Representation:

Fig6: Buggy completing Bronze challenge

Result Analysis:

We have successfully completed the Bronze challenge by using various sensors like IR sensors, ultrasonic sensors, receiver circuit and ZigBee module.